

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

拝啓 時下ますますご清栄のこととお喜び申し上げます。平素は格別のご高配を賜り厚く御礼申し上げます。

さて、2017年5月1日を以ってルネサス セミコンダクタ パッケージ&テスト ソリューションズ株式会社の半導体製造装置をはじめとする各種産業用制御ボードの受託開発・製造および画像認識システム開発・製造・販売事業を日立マクセル株式会社へ譲渡したことにより、当該事業は日立マクセル株式会社の子会社として新設されるマクセルシステムテック株式会社に承継されております。

従いまして、ドキュメント等資料中には、旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

敬具

2017年5月1日

マクセルシステムテック株式会社

【発行】 マクセルシステムテック (<http://www.systemtech.maxell.co.jp/>)

【お問い合わせ先】 [denki-support@maxell.co.jp](mailto:denki-support@maxell.co.jp)

---

**maxell**  
マクセルシステムテック株式会社

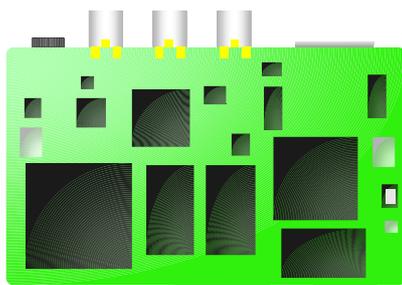
超小型  
画像処理ボード

**SVP-330**

*Network Vision Processor*

# Software Kit Development

コマンドリファレンス



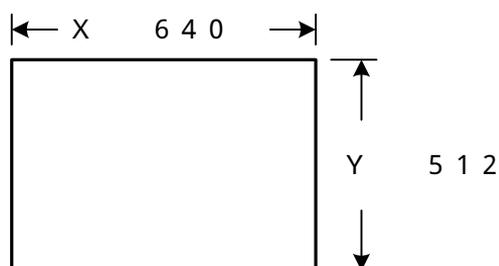
# 画像処理の制限事項

SVP - 330では、画像処理プロセッサの制限により画像処理を行う際、以下の制限があります。

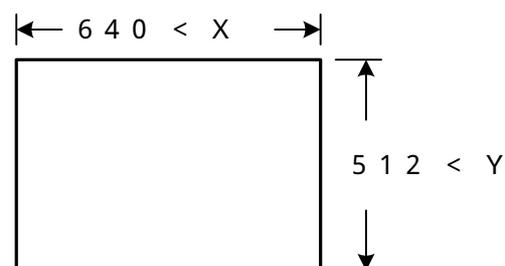
## 1. 画面サイズの制限事項

画面の大きさが640(X)×512(Y)より大きいサイズで画像処理を行うことはできません。

処理可能な画面サイズ

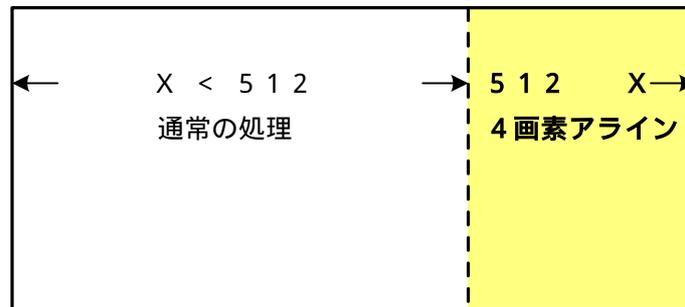


処理できない画面サイズ



## 2. ウィンドウサイズの制限事項

処理ウィンドウのX方向の大きさが512以上が指定された場合、X方向のサイズが4画素アラインサイズに切り捨てられて処理されます。対象のコマンドは下表のデスティネーション画面に処理結果が出力されるコマンドです。



### X方向4バイトアラインコマンド

	対象コマンド	備考
1	画像クリアコマンド	
2	画像転送・アフィン変換コマンド	IP_Zoom(), IP_ZoomExt(), IP_Rotate() コマンド以外
3	2値化コマンド	
4	画素変換コマンド	
5	画像間算術演算コマンド	
6	画像間論理演算コマンド	
7	2値画像形状変換コマンド	
8	コンボリューションコマンド	
9	ミニ/マックスフィルタコマンド	
10	ランクフィルタコマンド	
11	ラベリングコマンド	ラベル毎特徴量抽出以外
12	拡張コンボリューションコマンド	
13	拡張画像処理コマンド	
14	擬似カラーコマンド	

## 3. 画像処理機能の制限事項

SVP-330では、ハードウェアの制約により以下のコマンドは使用できません。

### 使用できないコマンド

	対象コマンド	備考
1	2値パイプラインフィルタ	
2	2値マッチングフィルタ	
3	RGBLUT変換コマンド	

# 画像処理コマンド一覧

## 1. PCドライバセットアップコマンド

コマンド名	機能	ページ	備考
OpenIPDevExt	PCドライバのオープンと初期化	Ref1-1	
OpenIPDev	PCドライバのオープン	Ref1-3	
CloseIPDev	PCドライバのクローズ	Ref1-4	
ResetIPSys	画像処理ボードのリセット	Ref1-5	
LoadIPSys	画像処理コマンド実行ファイルのダウンロード	Ref1-6	
BootIPSys	画像処理コマンドのブート	Ref1-8	
GetIPDevNumber	ボード枚数の読み出し	Ref1-9	
GetIPDevCount	オープンボード枚数の読み出し	Ref1-10	
GetOpenIPDevCount	オープン回数の読み出し	Ref1-11	
EnumAttachIPDev	装着ボードの列挙	Ref1-12	
QueryAttachIPDev	装着ボードチェック	Ref1-13	

## 2. コマンドエラー制御コマンド

コマンド名	機能	ページ	備考
ClearIPError	コマンドエラーのクリア	Ref2-1	
CheckIPError	コマンドエラーのチェック	Ref2-2	
ReadIPErrorTable	コマンドエラー情報の取得	Ref2-3	
EnableIPErrorMessage	エラーメッセージの出力の許可	Ref2-4	
DisableIPErrorMessage	エラーメッセージの出力の禁止	Ref2-5	

### 3. システム制御コマンド

コマンド名	機能	ページ	備考
InitIP	画像処理システムの初期化	Ref3-1	
InitIPExt	画像処理システムの初期化（ハードウェアテスト付き）	Ref3-2	
GetIPVersionInfo	バージョン情報の取得	Ref3-3	
CheckIPVersion	バージョンチェック	Ref3-5	
SetIPDataType	画像処理システムデータタイプの設定	Ref3-6	
GetDispImgID	ディスプレイ画面の画面番号の取得	Ref3-7	
GetBitmapImgID	ビットマップ画面の画面番号の取得	Ref3-8	
ISP_BusyWait	画像処理プロセッサのビジーウェイト	Ref3-9	
ImgBusyWait	処理画面に対するビジーウェイト	Ref3-10	
Img2chBusyWait	2ch処理画面に対するビジーウェイト	Ref3-11	
ISP_BusyCheck	ビジーステータスの読み出し	Ref3-12	

### 4. 画像メモリ領域管理コマンド

コマンド名	機能	ページ	備考
AllocImg	画像メモリ領域確保	Ref4-1	
AllocLockImg	画像メモリ領域確保（固定領域）	Ref4-3	
FreeImg	画像メモリ領域開放（指定画面）	Ref4-6	
FreeAllImg	画像メモリ領域開放（全画面）	Ref4-7	
ReadImgTable	画像メモリ管理テーブル読出し	Ref4-8	
ChangeImgDataType	画面データタイプ変更	Ref4-10	
SetWindow	ウィンドウ設定	Ref4-11	
SetAllWindows	ウィンドウ設定（全種）	Ref4-13	
ResetAllWindows	ウィンドウリセット	Ref4-15	
EnableIPWindow	ウィンドウ有効化	Ref4-16	
DisableIPWindow	ウィンドウ無効化	Ref4-17	
ReadWindow	ウィンドウ設定座標読出し	Ref4-18	
AllocDispImg	ディスプレイ画面の確保	Ref4-19	
FreeDispImg	ディスプレイ画面の開放	Ref4-21	

## 5. 映像入力コマンド

コマンド名	機能	ページ	備考
SetVideoFrame	映像入力画面設定	Ref5-1	
SelectCamera	カメラ番号とカメラタイプの選択	Ref5-3	
GetCamera	カメラ映像入力	Ref5-5	
GetCameraWithSelectPort	カメラ映像入力(カメラ番号とカメラタイプの選択付き)	Ref5-7	
SetCameraSync	カメラ同期信号の設定	Ref5-8	
SetVFDelay	映像入力画面の遅延サイズ設定	Ref5-9	
SetShutterSpeed	カメラ映像入力シャッタースピードの設定	Ref5-11	
Get2Camera	2カメラ映像の同時入力	Ref5-12	
SetTriggerMode	カメラ映像入力トリガモードの設定	Ref5-13	
GetCameraSts	カメラ接続のチェック	Ref5-14	
Get4Camera	4カメラ映像の同時入力	Ref5-15	
ResetCamera	リセットカメラ	Ref5-16	

## 6. 映像表示コマンド

コマンド名	機能	ページ	備考
DispCamera	カメラ映像表示	Ref6-1	
DispImg	画像メモリ表示	Ref6-2	
NoDisp	表示終了	Ref6-4	
BitmapOverlap	オーバーレイ表示制御	Ref6-5	
DispOverlap	画像メモリのオーバーレイ表示	Ref6-6	
SetDFDelay	映像表示画面の遅延サイズ設定	Ref6-7	
SetDispFrame	映像表示画面設定	Ref6-8	
SelectDisp	映像出力形式の選択	Ref6-10	
SetDispWindow	高精細映像表示のウィンドウ設定	Ref6-11	
EnableHIREZDisp	高精細映像表示の有効化	Ref6-12	
DisableHIREZDisp	高精細映像表示の無効化	Ref6-13	

## 7. 画像クリアコマンド

コマンド名	機能	ページ	備考
IP_ClearAllImg	画像メモリクリア(全チャンネル)	Ref7-1	
IP_ClearCHImg	画像メモリクリア(指定チャンネル)	Ref7-2	
IP_ClearImg	画像メモリクリア(指定画面)	Ref7-3	
IP_Const	定数発生	Ref7-4	

## 8. 画像転送・アフィン変換コマンド

コマンド名	機能	ページ	備考
IP_Copy	転送	Ref8-1	
IP_Zoom	ズーム	Ref8-2	
IP_ZoomExt	ズーム（縦横任意倍率）	Ref8-4	
IP_ZoomOut	ズームアウト	Ref8-6	
IP_ZoomOutExt	ズームアウト（縦横任意倍率）	Ref8-7	
IP_Shift	シフト	Ref8-8	
IP_ZoomS	シフト付きズーム	Ref8-10	
IP_Rotate	回転	Ref8-12	
IP_ZoomIn	ズームイン	Ref8-14	
IP_ZoomInExt	ズームイン（縦横任意倍率）	Ref8-15	

## 9. 2値化コマンド

コマンド名	機能	ページ	備考
IP_Binarize	2 値化	Ref9-1	
IP_BinarizeExt	範囲・反転付き 2 値化	Ref9-2	

## 10. 画素変換コマンド

コマンド名	機能	ページ	備考
IP_Invert	論理反転	Ref10-1	
IP_Minus	符号反転	Ref10-2	
IP_Abs	絶対値	Ref10-3	
IP_AddConst	定数加算	Ref10-4	
IP_SubConst	定数減算	Ref10-5	
IP_SubConstAbs	定数減算絶対値	Ref10-6	
IP_MultConst	定数乗算	Ref10-7	
IP_MinConst	定数比較 Min	Ref10-9	
IP_MaxConst	定数比較 Max	Ref10-10	
WriteConvertLUT	濃度変換データ設定	Ref10-11	
IP_ConvertLUT	濃度変換	Ref10-12	
IP_ShiftDown	定数シフトダウン	Ref10-13	
IP_ShiftUp	定数シフトアップ	Ref10-14	
IP_AndConst	定数論理積	Ref10-15	

## 11. 画像間算術演算コマンド

コマンド名	機能	ページ	備考
IP_Add	加算	Ref11-1	
IP_Sub	減算	Ref11-2	
IP_SubAbs	減算絶対値	Ref11-3	
IP_Comb	係数付加算	Ref11-4	
IP_CombAbs	係数付加算絶対値	Ref11-6	
IP_Mult	乗算	Ref11-8	
IP_Average	平均	Ref11-10	
IP_Min	比較 M i n	Ref11-11	
IP_Max	比較 M a x	Ref11-12	
IP_SubConstAbsAdd	定数減算絶対値和	Ref11-13	
IP_SubConstMultAdd	定数減算自乗和	Ref11-15	
IP_SubConstMult	定数減算積	Ref11-17	
IP_CombDrop	係数付加算（切り捨て）	Ref11-19	

## 12. 画像間論理演算コマンド

コマンド名	機能	ページ	備考
IP_And	論理積	Ref12-1	
IP_Or	論理和	Ref12-2	
IP_Xor	排他的論理和	Ref12-3	
IP_InvertAnd	否定論理積	Ref12-4	
IP_InvertOr	否定論理和	Ref12-5	
IP_Xnor	排他的否定論理和	Ref12-6	

## 13. 2値画像形状変換コマンド

コマンド名	機能	ページ	備考
IP_PickNoise4	2値画像ノイズ除去（4連結）	Ref13-1	
IP_PickNoise8	2値画像ノイズ除去（8連結）	Ref13-2	
IP_Outline4	2値画像輪郭抽出（4連結）	Ref13-4	
IP_Outline8	2値画像輪郭抽出（8連結）	Ref13-5	
IP_Dilation4	2値画像膨張（4連結）	Ref13-7	
IP_Dilation8	2値画像膨張（8連結）	Ref13-8	
IP_Erosion4	2値画像収縮（4連結）	Ref13-10	
IP_Erosion8	2値画像収縮（8連結）	Ref13-11	
IP_Thin4	2値画像細線化（4連結）	Ref13-13	
IP_Thin8	2値画像細線化（8連結）	Ref13-14	
IP_Shrink4	2値画像縮退化（4連結）	Ref13-16	
IP_Shrink8	2値画像縮退化（8連結）	Ref13-17	

## 14. コンボリレーションコマンド

コマンド名	機能	ページ	備考
IP_SmoothFLT	平滑化	Ref14-1	
IP_EdgeFLT	濃淡画像輪郭強調	Ref14-3	
IP_EdgeFLTAbs	濃淡画像輪郭強調（絶対値）	Ref14-5	
IP_Lap14FLT	ラプラシアン（4連結）	Ref14-7	
IP_Lap18FLT	ラプラシアン（8連結）	Ref14-8	
IP_Lap14FLTAbs	ラプラシアン（4連結・絶対値）	Ref14-9	
IP_Lap18FLTAbs	ラプラシアン（8連結・絶対値）	Ref14-10	
IP_LineFLT	ラインフィルタ	Ref14-11	
IP_LineFLTAbs	ラインフィルタ（絶対値）	Ref14-13	

## 15. ミニ/マックスフィルタコマンド

コマンド名	機能	ページ	備考
IP_MinFLT	局所最小値フィルタ	Ref15-1	
IP_MinFLT4	局所最小値フィルタ（4連結）	Ref15-3	
IP_MinFLT8	局所最小値フィルタ（8連結）	Ref15-4	
IP_MaxFLT	局所最大値フィルタ	Ref15-5	
IP_MaxFLT4	局所最大値フィルタ（4連結）	Ref15-7	
IP_MaxFLT8	局所最大値フィルタ（8連結）	Ref15-8	
IP_LineMinFLT	ライン局所最小値フィルタ	Ref15-9	
IP_LineMaxFLT	ライン局所最大値フィルタ	Ref15-11	

## 16. ランクフィルタコマンド

コマンド名	機能	ページ	備考
IP_RankFLT	局所ランクフィルタ	Ref16-1	
IP_Rank4FLT	局所ランクフィルタ（4連結）	Ref16-3	
IP_Rank8FLT	局所ランクフィルタ（8連結）	Ref16-4	
IP_MedFLT	局所メディアンフィルタ	Ref16-5	
IP_Med4FLT	局所メディアンフィルタ（4連結）	Ref16-7	
IP_Med8FLT	局所メディアンフィルタ（8連結）	Ref16-8	

## 17. ラベリングコマンド

コマンド名	機能	ページ	備考
IP_Label4	ラベリング ( 4 連結 )	Ref17-1	
IP_Label8	ラベリング ( 8 連結 )	Ref17-2	
IP_Label4withAreaFLT	面積フィルタ付ラベリング ( 4 連結 )	Ref17-3	
IP_Label8withAreaFLT	面積フィルタ付ラベリング ( 8 連結 )	Ref17-5	
IP_Label4withAreaFLTSort	面積フィルタ付ラベリング ( 面積ソート 4 連結 )	Ref17-7	
IP_Label8withAreaFLTSort	面積フィルタ付ラベリング ( 面積ソート 8 連結 )	Ref17-9	
IP_ExtractLORegionX	ラベル毎領域 X 座標抽出 ( Min / Max X 座標 )	Ref17-11	
IP_ExtractLORegionY	ラベル毎領域 Y 座標抽出 ( Min / Max Y 座標 )	Ref17-13	
IP_ExtractLOArea	ラベル毎面積抽出	Ref17-15	
IP_ExtractLOGravity	ラベル毎重心座標抽出	Ref17-17	

## 18. 濃淡画像特徴量抽出コマンド

コマンド名	機能	ページ	備考
IP_ExtractGOFeatures	濃淡画像基本特徴量抽出	Ref18-1	
IP_Histogram	濃度ヒストグラム ( 32 ビット )	Ref18-3	
IP_ProjectGO	X/Y軸への濃度累積値投影	Ref18-5	
IP_ProjectGOonX	X軸への濃淡累積値投影	Ref18-7	
IP_ProjectGOonY	Y軸への濃淡累積値投影	Ref18-9	
IP_ProjectGOMaxValue	X/Y軸への最大濃度投影	Ref18-11	
IP_ProjectGOMinValue	X/Y軸への最小濃度投影	Ref18-13	
IP_HistogramShort	濃度ヒストグラム ( 16 ビット )	Ref18-15	
IP_ProjectBlockGO	領域毎濃淡画像濃度累積	Ref18-17	
IP_ProjectBlockGOMinMaxValue	領域毎濃淡画像最大/最小濃度抽出	Ref18-19	
IP_ProjectLabelGO	ラベル毎濃淡画像濃度累積	Ref18-21	
IP_ProjectLabelGOMinMaxValue	ラベル毎濃淡画像最小/最大濃度値抽出	Ref18-23	
EnableRotateProject	斜方投影処理有効	Ref18-25	
DisableRotateProject	斜方投影処理無効	Ref18-26	
IP_HistogramFeatures	濃度ヒストグラム特徴量抽出	Ref18-27	

## 19. 2値画像特徴量抽出コマンド

コマンド名	機能	ページ	備考
IP_ExtractB0Features	2 値画像基本特徴量抽出	Ref19-1	
IP_ProjectB0	X / Y 軸への投影	Ref19-3	
IP_ProjectB0RegionX	領域 X 座標抽出 ( Y 軸への Min&Max X 投影 )	Ref19-5	
IP_ProjectB0RegionY	領域 Y 座標抽出 ( X 軸への Min&Max Y 投影 )	Ref19-7	
IP_ExtractB0Area	2 値画像累積値抽出	Ref19-9	
IP_ProjectBlockB0	領域毎 2 値画像濃度累積	Ref19-10	

## 20. テンプレートデータ領域管理コマンド

コマンド名	機能	ページ	備考
vpxAllocCorrTemplate	テンプレート特徴量データ領域確保	Ref20-1	
vpxFreeCorrTemplate	テンプレート特徴量データ領域開放	Ref20-2	
vpxReadCorrTemplate	テンプレート特徴量データ読出し	Ref20-3	
vpxWriteCorrTemplate	テンプレート特徴量データ書込み	Ref20-5	

## 21. トレーニングコマンド

コマンド名	機能	ページ	備考
vpxEnableCorrMask	テンプレートマスクの有効化	Ref21-1	
vpxDisableCorrMask	テンプレートマスクの無効化	Ref21-2	
vpxSetCorrTemplate	トレーニング	Ref21-3	
vpxSetCorrTemplateExt	トレーニング	Ref21-5	

## 22. サーチコマンド

コマンド名	機能	ページ	備考
vpxIP_CorrStep	ラスタースーチ	Ref22-1	
vpxIP_CorrPoint	近傍サーチ	Ref22-3	
vpxIP_CorrPrecise	サブピクセルサーチ	Ref22-5	
vpxEnableCorrBreak	相関演算途中打ち切りの有効化	Ref22-6	
vpxDisableCorrBreak	相関演算途中打ち切りの無効化	Ref22-7	
vpxSetCorrBreakThr	相関演算途中打ち切り閾値設定	Ref22-8	
vpxSetSearchDistance	サーチ除外領域の設定	Ref22-9	
vpxSetCorrMode	相関サーチモード設定	Ref22-10	
vpxSetCorrPrecise	サブピクセル算出桁数設定	Ref22-11	
vpxIP_CorrMap	相関値分布の抽出	Ref22-12	
vpxGetCorrMapSize	相関値分布格納テーブルサイズ取得	Ref22-14	

## 23. 2値化閾値算出支援コマンド

コマンド名	機能	ページ	備考
HistAnalyze	判別分析法	Ref23-1	

## 24. 直線抽出コマンド

コマンド名	機能	ページ	備考
GetHoughLine	離散データ座標からの直線抽出	Ref24-1	
GetHoughLineRow	離散データ座標からの2直線抽出	Ref24-5	
GetHoughLineRowExt	離散データ座標からの2直線抽出(拡張)	Ref24-7	
GetSideHoughLine	離散データ座標からの直線抽出(角度固定)	Ref24-9	
GetCrossPoint	ハフ変換直線の交点座標算出	Ref24-11	
GetRectPoint	ハフ変換直線の矩形交点座標の算出	Ref24-13	
GetRectCenter	矩形頂点中心座標の算出	Ref24-15	
GetAnglePoint4	矩形頂点間中心座標4ポイントからの角度算出	Ref24-17	
GetAnglePoint2	矩形頂点間中心座標2ポイントからの角度算出	Ref24-19	

## 25. イメージキャリパコマンド

コマンド名	機能	ページ	備考
ProjectLine	傾斜矩形領域の投影	Ref25-1	
LineEdgeFilter	エッジ抽出	Ref25-4	
LineEdgeFilterExt	フィルター付きエッジ抽出	Ref25-5	
LineCaliper	ラインキャリパ	Ref25-7	
CaliperLPtoSP	ライン座標から平面座標への変換	Ref25-9	
GetCaliperScore	エッジ判定閾値の算出	Ref25-14	
SetCaliperWidth	投影幅の算出	Ref25-15	

## 26. エッジファインダコマンド

コマンド名	機能	ページ	備考
LineEdgeFinder	ラインエッジファインダ	Ref26-1	
EdgeFinderLPtoSP	ライン座標から平面座標への変換	Ref26-5	

## 27. I/O入出力コマンド

コマンド名	機能	ページ	備考
InIPPort	I/Oポートからのデータ入力	Ref27-1	
OutIPPort	I/Oポートへのデータ出力	Ref27-2	

## 28. 画像メモリアクセスコマンド

### 画像メモリアクセスコマンド使用手順

コマンド名	機能	ページ	備考
OpenImg	画像メモリアクセス許可	Ref28-2	
CloseImg	画像メモリアクセス禁止	Ref28-3	
ReadImg	画像メモリ読出し(矩形領域)	Ref28-4	
WriteImg	画像メモリ書込み(矩形領域)	Ref28-5	
ReadImgReverse	画像メモリ読出し(ビットマップ矩形領域)	Ref28-6	
WriteImgReverse	画像メモリ書込み(ビットマップ矩形領域)	Ref28-7	
SetPixelPointer	画像メモリポインタ設定	Ref28-8	
ReadPixel	画像メモリ1画素読出し	Ref28-9	
WritePixel	画像メモリ1画素書込み	Ref28-10	
ReadPixelContinue	画像メモリ連続1画素読出し	Ref28-11	
WritePixelContinue	画像メモリ連続1画素書込み	Ref28-12	
ReadImgLine	画像メモリ1ライン読出し	Ref28-13	
WriteImgLine	画像メモリ1ライン書込み	Ref28-15	
GetImVirtualAddress	画像メモリのアドレス取得	Ref28-16	

## 29. 図形描画コマンド

コマンド名	機能	ページ	備考
PutLine	線・矩形描画	Ref29-1	
PutCross	十字カーソル描画	Ref29-3	
PutLineWindow	ラインウィンドウ描画	Ref29-5	
PutPolygon	多角形描画	Ref29-7	
PutCircle	円描画	Ref29-9	
PutArc	円弧描画	Ref29-10	
PutRectangle	矩形描画	Ref29-12	
PutTriangle	三角形描画	Ref29-14	
PutDiamond	菱形描画	Ref29-16	
PutCrossRect	十字形描画	Ref29-18	
PutTwinRectangle	2連矩形描画	Ref29-20	
PutArcExt	円弧描画(拡張)	Ref29-22	
RefreshImg	画像メモリのリフレッシュ	Ref29-24	
PutEllipse	楕円描画	Ref29-25	

### 30. 文字描画コマンド

コマンド名	機能	ページ	備考
PutAnkString	全角文字列の描画（英数文字）	Ref30-1	
PutHalfString	半角文字列の描画（英数文字）	Ref30-3	
PutAnkChar	全角文字の描画（英数文字）	Ref30-5	
PutHalfChar	半角文字の描画（英数文字）	Ref30-7	
PutString	文字列描画（半角、全角、漢字）	Ref30-9	

### 31. 画面描画コマンド

コマンド名	機能	ページ	備考
ClearBitmap	画面描画（範囲・色指定）	Ref31-1	
ClearScreen	画面クリア	Ref31-2	

### 32. タイマ/RTCコマンド

コマンド名	機能	ページ	備考
startGetIPTime	タイマスタート	Ref32-1	
timeGetIPTime	タイマ時間読出し	Ref32-2	
stopGetIPTime	タイマストップ	Ref32-3	
IPSleep	タイマウェイト	Ref32-4	
SetIPTimeMode	時間計測モード設定	Ref32-5	
SetIPTime	時刻の登録	Ref32-6	
GetIPTime	時刻の取得	Ref32-7	

### 33. プリフェッチ処理コマンド

コマンド名	機能	ページ	備考
vpxEnableCameraPrefetch	プリフェッチ処理の有効化	Ref33-1	
vpxGetCameraPrefetch	プリフェッチ映像入力	Ref33-3	
vpxDisableCameraPrefetch	プリフェッチ処理の終了	Ref33-4	

### 34. インテリジェントモジュール制御コマンド

コマンド名	機能	ページ	備考
RegistIntelIIPModule	インテリジェントモジュールの登録	Ref34-1	
ExecuteIntelIIPModule	インテリジェントモジュールの実行	Ref34-2	

### 35. 割込モジュール制御コマンド(PC側)

コマンド名	機能	ページ	備考
CreateIPTask	ユーザタスクの生成	Ref35-1	
CreateInterruptLink	割り込みリンクオブジェクトの生成	Ref35-3	
DeleteInterruptLink	割り込みリンクオブジェクトの削除	Ref35-5	
EnableInterruptObject	割り込み動作の有効化	Ref35-6	
DisableInterruptObject	割り込みオブジェクトの無効化	Ref35-7	
RegistIPTask	割込モジュールの登録	Ref35-8	
StartIPTaskwithParam	割込モジュールのサービス開始	Ref35-10	
WakeupIPTaskwithParam	割込モジュールの実行	Ref35-11	
WaitforIPTaskSignal	割込モジュールからの終了シグナルウェイト	Ref35-12	
ReadyforWaitIPTaskSignal	割込モジュールからの終了シグナルウェイト準備	Ref35-14	
WakeupIPTaskExt	割り込みモジュールの実行2	Ref35-15	
CancelIPTaskWait	終了シグナルウェイトの解除	Ref35-16	
StartIPTask	割込モジュールのサービス開始	Ref35-17	
WakeupIPTask	割込モジュールの実行	Ref35-18	
TerminateIPTask	割込モジュールのサービス終了	Ref35-19	
ResumeIPTask	割込モジュールサービスの再開	Ref35-20	
SuspendIPTask	割込モジュールサービスの一時的停止	Ref35-21	
DeleteIPTask	割込モジュールの削除	Ref35-22	

### 36. 割込モジュールI/Fコマンド(ボードCPU側)

コマンド名	機能	ページ	備考
SendSignaltoPC	P I O 割込モジュールからのシグナル送信	Ref36-1	

## 37. モジュールサポートコマンド

コマンド名	機能	ページ	備考
InitIPParamTable	モジュールのパラメータテーブルの初期化	Ref37-1	
SetIntegerParam	整数型パラメータのセット	Ref37-3	
SetFloatParam	実数型パラメータのセット	Ref37-4	
SetParamTable	配列データ（入力）パラメータのセット	Ref37-5	
AllocParamTable	配列データ（出力）パラメータ領域の確保	Ref37-7	
GetParamTable	配列データ（出力）パラメータの読み出し	Ref37-9	
SetParamTableExt	配列データ（入力）パラメータのセット	Ref37-11	
AllocParamTableExt	配列データ（出力）パラメータ領域の確保	Ref37-13	
ReadIPMemDatabyLong	ボード側システムメモリデータリード（4バイト）	Ref37-15	
ReadIPMemDatabyWord	ボード側システムメモリデータリード（2バイト）	Ref37-16	
ReadIPMemDatabyByte	ボード側システムメモリデータリード（1バイト）	Ref37-17	
WriteIPMemDatabyLong	ボード側システムメモリデータライト（4バイト）	Ref37-18	
WriteIPMemDatabyWord	ボード側システムメモリデータライト（2バイト）	Ref37-19	
WriteIPMemDatabyByte	ボード側システムメモリデータライト（1バイト）	Ref37-20	
ExitIPTask	自タスクの終了	Ref37-21	
GetIPTaskID	実行状態の自タスクIDの参照	Ref37-22	

### 38. SCIコントロールコマンド

コマンド名	機能	ページ	備考
InitSCI	SCIデバイスの初期化	Ref38-1	
SetSCIBuffer	SCIバッファの設定	Ref38-3	
FlushSCIBuffer	SCIバッファのフラッシュ	Ref38-4	
ReadSCIBuffer	SCIバッファからのデータリード	Ref38-5	
ReadSCIBufferWithTimeout	SCIバッファからのデータリード (タイムアウト付き)	Ref38-6	
WriteSCIBuffer	SCIへのデータ出力	Ref38-7	
IsSCIBuffer	SCIバッファデータ数の取得	Ref38-8	
IsSCIErrror	SCI通信エラーの取得	Ref38-9	
IsSCIBufferFull	SCIバッファフルエラーの取得	Ref38-10	

### 39. 2値パイプラインフィルタコマンド

コマンド名	機能	ページ	備考
SetTrsPipelineFLTMode	パイプラインフィルタのモード設定	Ref39-1	未サポート
IP_TrspipelineFLT	2値パイプラインフィルタの実行	Ref39-4	未サポート

### 40. パイプライン制御コマンド

コマンド名	機能	ページ	備考
EnablePipeline	パイプラインモードの有効化	Ref40-1	
DisablePipeline	パイプラインモードの無効化	Ref40-2	

### 41. オンボードデバッグサポートコマンド

コマンド名	機能	ページ	備考
printf	書式付き出力	Ref41-1	
scanf	書式付き入力	Ref41-2	
charget	1文字入力	Ref41-3	
charput	1文字出力	Ref41-4	

## 42. タスクコントロールコマンド

コマンド名	機能	ページ	備考
ChangeIPTaskPriority	タスクプライオリティの変更	Ref42-1	
EnablePIOInterrupt	P I O 割込マスクの解除	Ref42-3	
DisablePIOInterrupt	P I O 割込のマスク	Ref42-4	
SleepIPTask	モジュールのスリープ	Ref42-5	
WaitIPTask	モジュールの時間指定スリープ	Ref42-6	
GetIPTaskStatus	タスクステータスの取得	Ref42-7	
CancelWakeup	ウェイクアップ要求無効化	Ref42-9	
WaitSemaphore	セマフォウェイト	Ref42-10	
SignalSemaphore	セマフォシグナル	Ref42-11	
SemStatus	セマフォステータスの取得	Ref42-12	
SignalIPSemaphore	P C からのセマフォシグナル	Ref42-13	
SuspendSystemTimer	システムタイマの一時停止	Ref42-14	
ResumeSystemTimer	システムタイマの一時停止解除	Ref42-15	
CreateIPSemaphore	セマフォの生成	Ref42-16	
DeleteIPSemaphore	セマフォの削除	Ref42-17	
ReadSemaphore	セマフォの状態参照	Ref42-18	
WaitSemaphoreWithTimeOut	タイムアウト付きセマフォウェイト	Ref42-19	
CreateEventFlag	イベントフラグの生成	Ref42-20	
DeleteEventFlag	イベントフラグの削除	Ref42-21	
SetEventFlag	イベントフラグのセット	Ref42-22	
ClearEventFlag	イベントフラグのクリア	Ref42-23	
WaitEventFlag	イベントフラグウェイト	Ref42-24	
WaitEventFlagWithTimeOut	タイムアウト付きイベントフラグウェイト	Ref42-26	
ReadEventFlag	イベントフラグの状態参照	Ref42-27	
DelayIPTask	タスク遅延	Ref42-28	
ReleaseWait	待ち状態の強制解除	Ref42-29	

## 43. フラッシュメモリアクセスコマンド

コマンド名	機能	ページ	備考
fmOpen	ファイルオープン	Ref43-1	
fmClose	ファイルクローズ	Ref43-2	
fmRead	ファイル読み込み	Ref43-3	
fmWrite	ファイル書き込み	Ref43-4	
fmGetFileSize	ファイルサイズ取得	Ref43-5	
fmDelete	ファイル削除	Ref43-6	
fmRename	ファイル名変更	Ref43-7	
fmFileList	ファイルリスト取得	Ref43-8	
fmFormat	フラッシュメモリフォーマット	Ref43-9	
fmDiskSize	ディスクサイズ取得	Ref43-10	
fmDiskFree	未使用領域サイズ取得	Ref43-11	
fmFindFile	ファイル検索	Ref43-12	
fmChgAttr	ファイル属性の変更	Ref43-13	
fmSetAttr	ファイル属性の設定	Ref43-14	
fmFileCopy	PC-フラッシュメモリ間のファイルコピー	Ref43-15	

## 44. VP810互換コマンド

コマンド名	機能	ページ	備考
vpxInitIP	画像システムの初期化	Ref44-1	
vpxEnableLoopCamera	ループ映像入力開始	Ref44-3	
vpxDisableLoopCamera	ループ映像入力終了	Ref44-4	
vpxSuspendLoopCamera	ループ映像入力一時停止	Ref44-5	
vpxResumeLoopCamera	ループ映像入力再開	Ref44-6	
EnableLoopDisp	ループ表示開始	Ref44-7	
DisableLoopDisp	ループ表示終了	Ref44-8	
SuspendLoopDisp	ループ表示一時停止	Ref44-9	
ResumeLoopDisp	ループ表示再開	Ref44-10	
vpxIP_Rotate	回転	Ref44-11	
vpxWriteConvertLUT	濃度変換データ設定	Ref44-13	
vpxIP_SmoothFLT	平滑化	Ref44-14	
vpxIP_EdgeFLT	濃淡画像輪郭強調	Ref44-16	
vpxIP_EdgeFLTAbs	濃淡画像輪郭強調 (絶対値)	Ref44-18	
vpxIP_LineFLT	ラインフィルタ	Ref44-20	
vpxIP_LineFLTAbs	ラインフィルタ (絶対値)	Ref44-22	
vpxIP_SmoothFLTF	平滑化 (周辺部処理なし)	Ref44-24	
vpxIP_ExtractLOArea	ラベル毎面積抽出	Ref44-25	
vpxIP_ExtractLORegionX	ラベル毎領域 X 座標抽出 (Min/Max X 座標)	Ref44-27	
vpxIP_ExtractLORegionY	ラベル毎領域 Y 座標抽出 (Min/Max Y 座標)	Ref44-29	
vpxIP_ProjectG0	X / Y 軸への濃度累積値投影 (16ビット)	Ref44-31	
vpxIP_ProjectG0MaxValue	X / Y 軸への最大濃度値投影	Ref44-33	
vpxIP_ProjectG0MinValue	X / Y 軸への最小濃度値投影	Ref44-35	
vpxIP_ProjectB0	X / Y 軸への投影	Ref44-37	
vpxIP_ProjectB0RegionX	領域 X 座標抽出 (Y 軸へのMin&Max X 投影)	Ref44-39	
vpxIP_ProjectB0RegionY	領域 Y 座標抽出 (X 軸へのMin&Max Y 投影)	Ref44-41	

## 45. マルチポート映像入出力コマンド

コマンド名	機能	ページ	備考
SetCameraPortConfig	カメラポート配置の設定	Ref45-1	
ActiveVideoPort	カレントビデオポートの設定	Ref45-3	
SetVideoFrameMltPort	マルチポート対応映像入力画面設定	Ref45-4	
SelectCameraMltPort	マルチポート対応カメラ番号選択	Ref45-5	
GetCameraMltPort	マルチポート対応映像入力	Ref45-6	
DispCameraMltPort	マルチポート対応カメラ映像表示	Ref45-7	
SetTriggerModeMltPort	マルチポート対応トリガモードの設定	Ref45-8	
SetShutterSpeedMltPort	マルチポート対応シャッタースピードの設定	Ref45-9	
Get2CameraMltPort	マルチポート対応2 カメラ映像の同時入力	Ref45-10	
EnableLoopCameraMltPort	マルチポート対応ループ映像入力開始	Ref45-11	
SuspendLoopCameraMltPort	マルチポート対応ループ映像入力停止	Ref45-12	

## 46. YUVカラー処理コマンド

コマンド名	機能	ページ	備考
GetCamera	YUVカメラ映像入力	Ref46-1	
DispCamera	カメラ映像表示	Ref46-3	
IP_ClearImg	画像メモリクリア (指定画面)	Ref46-4	
IP_Copy	画像転送	Ref46-5	
ReadImg	画像メモリ読み出し (矩形領域)	Ref46-6	
WriteImg	画像メモリ書き込み (矩形領域)	Ref46-8	
AllocYUVImg	カラー映像用画像メモリ領域確保	Ref46-10	
GetUVImgID	UV 画面番号取り出し	Ref46-12	
ReadYUVImgTable	カラー画面管理テーブル読み出し	Ref46-13	
IP_Mask	マスク処理	Ref46-15	
IP_ClearColor	カラー画面の色情報クリア (指定画面)	Ref46-17	
IP_ExtractColor	カラー抽出処理	Ref46-18	
IP_ExtractColorRhoTheta	カラー抽出処理 - 色彩距離・色相変換	Ref46-20	
IP_ConvertRho	色彩距離変換	Ref46-23	
IP_ConvertTheta	色相変換	Ref46-25	
IP_ConvertRhoTheta	色彩距離・色相変換	Ref46-27	
IP_ConvertYUVtoRGB	カラー疑似変換 (YUV RGB)	Ref46-29	
IP_ConvertYUVtoRGBfast	カラー疑似変換 (YUV RGB)	Ref46-30	
OpenMultiColor	マルチカラー抽出処理可能	Ref46-31	
CloseMultiColor	マルチカラー処理不可	Ref46-33	
ClearMultiColorYRT	Y マルチカラーテーブルクリア	Ref46-34	
SetMultiColorYRT	Y マルチカラー抽出データ設定	Ref46-35	
IP_ExtractMultiColorRhoTheta	Y マルチカラー抽出	Ref46-37	
ClearMultiColor	マルチカラーテーブルクリア	Ref46-40	
SetMultiColor	マルチカラー抽出データ設定	Ref46-41	
IP_ExtractMultiColor	マルチカラー抽出	Ref46-43	

## 47. RGBカラー処理コマンド

コマンド名	機能	ページ	備考
GetCamera	R G Bカメラ映像入力	Ref47-1	
AttachRGBWorkImg	R G B映像入力ワークフレームの設定	Ref47-3	
DetachRGBWorkImg	R G B映像入力ワークフレームの解除	Ref47-4	
AllocRGBImg	画像メモリ領域確保 (RGB画面)	Ref47-5	
ReadRGBImgTable	RGB画像メモリ管理テーブル読み出し	Ref47-7	
GetGImgID	G画面番号抽出	Ref47-9	
GetBImgID	B画面番号抽出	Ref47-10	
IP_ConvertHue	色相変換	Ref47-11	
IP_ConvertSaturation	彩度変換	Ref47-13	
IP_ConvertIntensity	明度変換	Ref47-15	
IP_ConvertRGBtoYUV	カラー疑似変換 (RGB YUV)	Ref47-17	
IP_ConvertRGBtoYUVfast	カラー疑似変換 (RGB YUV)	Ref47-18	
IP_ExtractColorRGB	RGBカラー抽出処理	Ref47-19	
IP_ExtractColorHSI	カラー抽出処理 - 色相・彩度・明度変換	Ref47-21	
OpenMultiColor	マルチカラー抽出処理可能	Ref47-23	
CloseMultiColor	マルチカラー処理不可	Ref47-25	
ClearMultiColorHSI	HSIマルチカラーテーブルクリア	Ref47-26	
SetMultiColorHSI	HSIマルチカラー抽出データ設定	Ref47-27	
IP_ExtractMultiColorHSI	HSIマルチカラー抽出	Ref47-29	
ClearMultiColorRGB	RGBマルチカラーテーブルクリア	Ref47-31	
SetMultiColorRGB	RGBマルチカラー抽出データ設定	Ref47-32	
IP_ExtractMultiColorRGB	RGBマルチカラー抽出	Ref47-34	

## 48. 2値マッチングフィルタ

コマンド名	機能	ページ	備考
SetBinMatchTemplate	2値画像マッチングフィルタ登録	Ref48-1	未サポート
IP_BinMatchFLT	2値画像マッチングフィルタ実行	Ref48-3	未サポート
IP_BinarizePTM3x3	2値画像マッチングフィルタ (3 × 3)	Ref48-5	未サポート
IP_BinarizePTM5x5	2値画像マッチングフィルタ (5 × 5)	Ref48-6	未サポート
IP_BinarizePTM7x7	2値画像マッチングフィルタ (7 × 7)	Ref48-7	未サポート
IP_BinarizePTM9x9	2値画像マッチングフィルタ (9 × 9)	Ref48-8	未サポート
IP_BinarizePTM3x5	2値画像マッチングフィルタ (3 × 5)	Ref48-9	未サポート
IP_BinarizePTM5x7	2値画像マッチングフィルタ (5 × 7)	Ref48-10	未サポート
IP_BinarizePTM7x9	2値画像マッチングフィルタ (7 × 9)	Ref48-11	未サポート

## 49. 画像ファイリングコマンド

コマンド名	機能	ページ	備考
LoadBMPFile	BMPファイルのロード	Ref49-1	
SaveBMPFile	BMPファイルのセーブ	Ref49-3	

## 50. 拡張コンボリレーションコマンド

コマンド名	機能	ページ	備考
IP_SmoothFLT5x5	平滑化 ( 5 × 5 )	Ref50-1	
IP_SmoothFLT7x7	平滑化 ( 7 × 7 )	Ref50-3	
IP_EdgeFLT5x5	濃淡画像輪郭強調 ( 5 × 5 )	Ref50-5	
IP_EdgeFLT7x7	濃淡画像輪郭強調 ( 7 × 7 )	Ref50-7	
IP_MinFLT5x5	局所最小値フィルタ ( 5 × 5 )	Ref50-9	
IP_MinFLT44	局所最小値フィルタ ( 5 × 5 , 4 4 連結 )	Ref50-11	
IP_MinFLT48	局所最小値フィルタ ( 5 × 5 , 4 8 連結 )	Ref50-12	
IP_MinFLT88	局所最小値フィルタ ( 5 × 5 , 8 8 連結 )	Ref50-13	
IP_MaxFLT5x5	局所最大値フィルタ ( 5 × 5 )	Ref50-14	
IP_MaxFLT44	局所最大値フィルタ ( 5 × 5 , 4 4 連結 )	Ref50-16	
IP_MaxFLT48	局所最大値フィルタ ( 5 × 5 , 4 8 連結 )	Ref50-17	
IP_MaxFLT88	局所最大値フィルタ ( 5 × 5 , 8 8 連結 )	Ref50-18	
IP_SmoothFLT5x5Ext	平滑化 ( 5 × 5 )	Ref50-19	
IP_SmoothFLT7x7Ext	平滑化 ( 7 × 7 )	Ref50-22	
IP_EdgeFLT5x5Ext	画像輪郭強調 ( 5 × 5 )	Ref50-25	
IP_EdgeFLT7x7Ext	画像輪郭強調 ( 7 × 7 )	Ref50-28	

## 51. 正規化相関(IP5000互換)コマンド

### 正規化相関コマンド使用手順

コマンド名	機能	ページ	備考
SetCorrMode	相関サーチモード設定	Ref51-3	
DisableCorrMask	テンプレートマスク無効	Ref51-5	
EnableCorrMask	テンプレートマスク有効	Ref51-6	
SetCorrBreakThr	打切り閾値設定	Ref51-7	
DisableCorrBreak	相関演算途中打切りの無効化	Ref51-9	
EnableCorrBreak	相関演算途中打切りの有効化	Ref51-10	
SetCorrControl	正規化相関制御	Ref51-11	
SetCorrTemplate	正規化相関テンプレート登録	Ref51-13	
SetCorrTemplateExt	正規化相関テンプレート登録(間引き付き)	Ref51-15	
IP_Corr	正規化相関	Ref51-17	
IP_CorrPecise	正規化相関(高精度位置決め)	Ref51-19	
SetOptFlowMode	オプティカルフローモード設定	Ref51-22	
IP_OptFlow	オプティカルフロー	Ref51-25	

## 52. グラフィックス(IP5000互換)コマンド

### グラフィックスコマンド使用手順

コマンド名	機能	ページ	備考
SetDrawMode	描画面面設定	Ref52-2	
SetStringAttributes	文字の属性設定	Ref52-3	
RefreshGraphics	パターン作成終了	Ref52-4	
DrawString	文字列描画	Ref52-5	
DrawLine	直線描画	Ref52-6	
DrawSegments	直線描画(複数)	Ref52-7	
DrawLines	折れ線描画	Ref52-8	
DrawRectangle	矩形描画	Ref52-9	
DrawPolygon	多角形描画	Ref52-10	
DrawArc	円弧描画	Ref52-11	
FillRectangle	矩形描画(塗りつぶし)	Ref52-13	
FillPolygon	多角形描画(塗りつぶし)	Ref52-14	

### 53. 拡張画像処理コマンド

コマンド名	機能	ページ	備考
IP_SmoothFLTExt	平滑化フィルタ (拡張版)	Ref53-1	
IP_EdgeFLTAbsExt	輪郭強調フィルタ・絶対値 (拡張版)	Ref53-3	
IP_Sobel	エッジソーベル	Ref53-10	
IP_SobelBinarize	エッジソーベル (2 値化)	Ref53-12	
IP_RegisterLUT	濃度変換テーブルの登録	Ref53-14	
IP_Prewitt	輪郭強調 (プレビット)	Ref53-15	
IP_PrewittBinarize	輪郭強調 2 値化 (プレビット 2 値化)	Ref53-17	

### 54. ランゲンス・ラベリングコマンド

コマンド名	機能	ページ	備考
IP_Label4byRL	ラング・ス・ラ・リグ 処理 (4 連結)	Ref54-1	
IP_Label8byRL	ラング・ス・ラ・リグ 処理 (8 連結)	Ref54-3	
IP_Label4byRLwithAreaFLT	面積フィルタ付きラング・ス・ラ・リグ 処理 (4 連結)	Ref54-5	
IP_Label8byRLwithAreaFLT	面積フィルタ付きラング・ス・ラ・リグ 処理 (8 連結)	Ref54-7	
IP_Label4byRLwithAreaFLTSort	面積フィルタ付きラング・ス・ラ・リグ 処理 (ソート 4 連結)	Ref54-9	
IP_Label8byRLwithAreaFLTSort	面積フィルタ付きラング・ス・ラ・リグ 処理 (ソート 8 連結)	Ref54-11	
IP_Label4byRLExt	ラング・ス・ラ・リグ 処理 (4 連結・拡張)	Ref54-13	
IP_Label8byRLExt	ラング・ス・ラ・リグ 処理 (8 連結・拡張)	Ref54-15	
IP_Label4byRLwithAreaFLTEExt	面積フィルタ付きラング・ス・ラ・リグ 処理 (4 連結・拡張)	Ref54-17	
IP_Label8byRLwithAreaFLTEExt	面積フィルタ付きラング・ス・ラ・リグ 処理 (8 連結・拡張)	Ref54-19	
IP_Label4byRLwithAreaFLTSortExt	面積フィルタ付きラング・ス・ラ・リグ 処理 (ソート 4 連結・拡張)	Ref54-21	
IP_Label8byRLwithAreaFLTSortExt	面積フィルタ付きラング・ス・ラ・リグ 処理 (ソート 8 連結・拡張)	Ref54-23	
IP_LabelCombine	統合ラベリング	Ref54-25	

## 55. BSDソケットコマンド

コマンド名	機能	ページ	備考
socket	ソケット生成	Ref55-1	
bind	通信アドレス情報の指定	Ref55-2	
listen	受動モード設定	Ref55-4	
accept	ソケットに対するコネクションの受入れ	Ref55-6	
connect	ソケットの接続の開始	Ref55-8	
getpeername	相手側通信アドレス情報取得	Ref55-11	
getsockname	自通信アドレス情報取得	Ref55-12	
recv	ソケットからの受信データ取得	Ref55-13	
recvfrom	受信データと送信者アドレス取得	Ref55-16	
send	ソケットへの送信データ設定	Ref55-18	
sendto	送信データと送信先アドレス設定	Ref55-21	
closesocket	ソケットのクローズ	Ref55-24	
shutdown	コネクション閉鎖	Ref55-25	
getsockopt	プロトコルのオプション取得	Ref55-27	
setsockopt	プロトコルのオプション変更	Ref55-31	
selectsocket	ソケットの利用可能待ち	Ref55-32	
set_blocking_socket	ブロッキングモード設定	Ref55-34	
get_errno	エラーコード取得	Ref55-35	
get_thread_errno	スレッドエラーコード取得	Ref55-36	
set_sock_timewait	TIMEWAIT 時間の変更	Ref55-37	
get_sock_recvlen	受信データ長の取得	Ref55-38	
set_sock_keepalive	キープアライブ制御情報の変更	Ref55-39	

## 56. ビデオ拡張コマンド

コマンド名	機能	ページ	備考
WriteVideoLUT	ビデオ入力LUTへのデータ書き込み	Ref56-1	
SetVideoOpt	ビデオオプションの設定	Ref56-2	
GetVideoOpt	ビデオオプションの参照	Ref56-5	

## 57. 線分化コマンド

コマンド名	機能	ページ	備考
ExtractPolyline	線分列抽出	Ref57-1	
PolyArea	線分列から面積抽出	Ref57-3	
PolyPerim	線分列からの周囲長抽出	Ref57-4	
PolyGrav	線分列から重心抽出	Ref57-5	
PolyFeatures	線分列から形状特徴量抽出	Ref57-6	

## 58. 2値画像の穴埋めコマンド

コマンド名	機能	ページ	備考
IP_FillHole	2値画像穴埋め	Ref58-1	
IP_FillHoleExt	2値画像穴埋め(拡張)	Ref58-3	

## 59. RGLUT変換コマンド

コマンド名	機能	ページ	備考
CreateRGLUT	RGLUTオブジェクトの生成	Ref59-1	未サポート
DeleteRGLUT	RGLUTオブジェクトの削除	Ref59-3	未サポート
SetRGLUT	RGLUTテーブルへ値設定	Ref59-4	未サポート
GetRGLUT	RGLUTテーブルから値取得	Ref59-5	未サポート
GetRGLUTAddr	RGLUTテーブルからLUTテーブル先頭アドレス取得	Ref59-6	未サポート
GetRGLUTSize	RGLUTテーブルからLUTテーブルサイズ取得	Ref59-7	未サポート
WriteRGLUT	RGLUTテーブルのLUTテーブル書き込み	Ref59-8	未サポート
ReadRGLUT	RGLUTテーブルのLUTテーブル読み出し	Ref59-9	未サポート
IP_ConvertRGLUT	RGLUT変換の実行	Ref59-10	未サポート
IP_ConvertRGLUTEx	RGLUT変換の実行(拡張)	Ref59-12	未サポート

## 60. 擬似カラーコマンド

コマンド名	機能	ページ	備考
IP_PseudoColor	RGLUTオブジェクトの生成	Ref60-1	
SetPseudoColor	RGLUTオブジェクトの削除	Ref60-2	
SetPseudoColorExt	RGLUTテーブルへ値設定	Ref60-4	

## 61. WDTコマンド

コマンド名	機能	ページ	備考
StartWDT	WDTスタート	Ref61-1	
StopWDT	WDTストップ	Ref61-2	
ResetWDT	WDTリセット	Ref61-3	
GetWDTStatus	WDTステータスの読み出し	Ref61-4	

## 付録

# OpenIPDevExt

## PCドライバのオープンと初期化

### 機能

画像処理ボードのデバイスドライバをオープンし、その画像処理ボードに対する制御情報を識別するためのデバイスID ( devID ) を取得します。その後、リセット、コマンドのダウンロード、ブートを実行し、全ての処理が成功するとデバイスIDを返します。

SVP - 330を同じネットワーク上で複数台使用する場合、SVP - 330の「BOOT.INI」及びPC側のレジストリ設定でのIPアドレス設定が必要です。詳細は環境設定ガイドを参照して下さい。

### コーディング

```
DEVID
OpenIPDevExt (
    int          BoardNo ,
    OPEN_IP_INFO *info
)
```

### コメント

ボード番号  
オプション情報

### パラメータ

#### BoardNo

使用するSVP-330のボード番号。複数台のSVP-330を同じネットワークにつないで制御するシステムを構成する場合、レジストリ設定(VPSETREG.EXE)で設定したIPアドレス、ポート番号のボード番号を指定して下さい。1台のみの使用の場合は、「0」を指定して下さい。なお、環境設定の詳細は、環境設定ガイドの「レジストリの設定」を参照して下さい。

ボード番号	設定内容
0	ボード番号 # 0 (ディップスイッチの設定値が「0」)
1	ボード番号 # 1 (ディップスイッチの設定値が「1」)
15	ボード番号 # 15 (ディップスイッチの設定値が「15」)

#### info

オプション情報テーブル。「NULL」を指定するとデフォルト動作で処理を実行します。リザーブの領域を含めて使用しないオプションは「0」クリアして下さい。

```
typedef struct {
    int    OpenMode;    // オープンモード
    int    OpenOpt;    // オープンオプション
    int    ResetOpt;   // リセットオプション
    int    BootOpt;    // ブートオプション
    int    Reserved[4]; // リザーブ
} OPEN_IP_INFO;
```

オプション情報テーブル	説明
OpenMode	オープンモード。「0」を指定して下さい。
OpenOpt	オープンオプション。「0」を指定して下さい。
ResetOpt	リセットオプション。「0」を指定して下さい。
BootOpt	ブートオプション。「0」を指定して下さい。
Reserved[4]	リザーブ。全て「0」にして下さい。

**リターン値**

リターン値はデバイスIDで DEVID 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	デバイスID
ISPX_NULL	異常終了

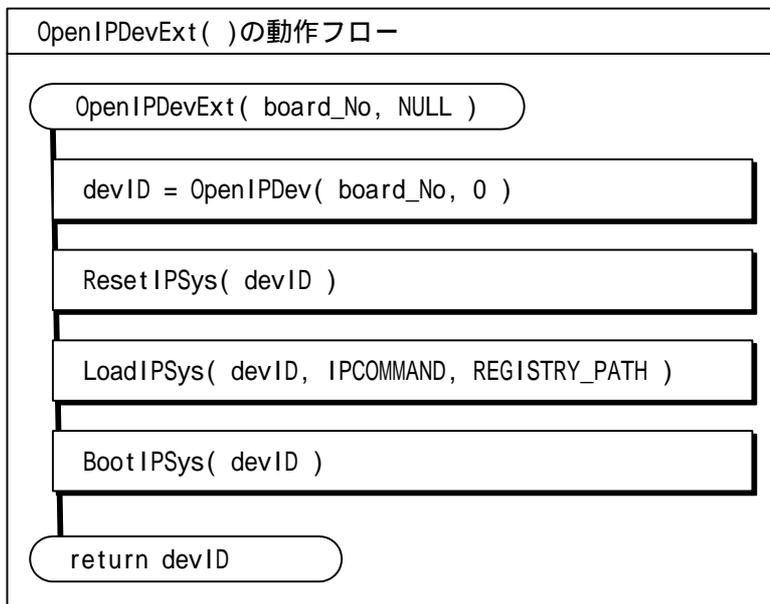
異常終了時、下記の原因が考えられます。

エラー番号	エラー原因
1	画像処理ボード未実装エラー
2	Windowsシステムエラー
3	画像処理ボードハードウェアエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

OpenIPDevExt( )はコマンド内で、以下のコマンドを実行します。



以下にOpenIPDevExt( )の使用例を示します。

## 例 1

```

OPEN_IP_INFO info;
memset(&info,0,sizeof(info));
devID = OpenIPDevExt(0,&info);

InitIP(devID);
. . .
  
```

## 例 2

```

devID = OpenIPDevExt(0,NULL);
InitIP(devID);
. . .
  
```

## OpenIPDev

## PCドライバのオープン

### 機能

画像処理ボードのデバイスドライバをオープンし、その画像処理ボードに対する制御情報を識別するためのデバイスID ( devID ) を取得します。

SVP - 330を同じネットワーク上で複数台使用する場合、SVP - 330の「B00T.INI」及びPC側のレジストリ設定でのIPアドレス設定が必要です。詳細は環境設定ガイドを参照して下さい。

### コーディング

```
DEVID
OpenIPDev(
    int          BoardNo ,
    int          opt
)
```

BoardNo ,  
opt

### コメント

ボード番号  
オプション

### パラメータ

#### BoardNo

使用するSVP-330のボード番号。複数台のSVP-330を同じネットワークにつないで制御するシステムを構成する場合、レジストリ設定(VPSETREG.EXE)で設定したIPアドレス、ポート番号のボード番号を指定して下さい。1台のみの使用の場合は、「0」を指定して下さい。なお、環境設定の詳細は、環境設定ガイドの「レジストリの設定」を参照して下さい。

ボード番号	設定内容
0	「Board0」で設定されたIPアドレス、ポート番号
1	「Board1」で設定されたIPアドレス、ポート番号
15	「Board15」で設定されたIPアドレス、ポート番号

#### opt

オプション。SVP - 330では0を設定して下さい。

### リターン値

リターン値はデバイスIDで DEVID 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	デバイスID
ISPX_NULL	異常終了

異常終了時、下記の原因が考えられます。

エラー番号	エラー原因
1	画像処理ボード未実装エラー
2	Windowsシステムエラー
3	画像処理ボードハードウェアエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンド内でPCのシステム時刻を取得し、SVP - 330の時刻設定を行っています。また、SVP - 330の時刻設定は、SVP - 330の現在時刻が2002以前の場合のみ行われます。

## CloseIPDev

## PCドライバのクローズ

### 機能

画像処理ボードのクリーンアップを行い、デバイスドライバをクローズします。

### コーディング

```
int  
CloseIPDev(  
    DEVID    devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了。デバイスIDの値が不適當。

### 詳細情報

## Reset IPSys

## 画像処理ボードのリセット

### 機能

本コマンドはV P - 9 1 0 との互換性のためにダミーで用意されていますので実際にはリセット処理はございません。

### コーディング

```
int
Reset IPSys(
    DEVID    devID
)
```

### コメント

デバイス I D

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
3	ハードウェアエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画像処理ボードをリセットし、ROMのイニシャライズプログラムを起動するには、以下の\_ResetIPSys( ) コマンドを使用して下さい。\_ResetIPSys( ) コマンドでリセットを行った場合、modeが0x00の場合、次にダウンロードやブートコマンドを実行するためにSVP - 330に対しネットワークの再接続を実行します。その必要がなければ、modeを0x10を指定して下さい。「BOOT.INI」でIPアドレスを変更し、リセットを実行すると再接続の実行でエラーが発生します。

また、\_ResetIPSys( ) コマンドのリセットはWDT (Watch Dog Timer) 機能を使用しています。modeが0x00または0x10で\_ResetIPSys( ) コマンドを実行した場合には、GetWDTStatus( ) コマンドでWDTステータスを読み出すと『ソフトウェアリセットからの起動』になります。

```
int _ResetIPSys(DEVID devID, int mode)
mode :
    0x00 : パワーオンリセット(再接続あり)
    0x01 : ノーオペレーション

    0x10 : パワーオンリセット(再接続なし)
    0x11 : ノーオペレーション
```

# LoadIPSys

## 画像処理コマンドの実行ファイルのダウンロード

### 機能

画像処理コマンドの実行モジュールやインテリジェントモジュール、割込み起動モジュールの実行ファイルをダウンロードします。

ユーザー領域(0x8C80:0000 ~ 0x8CFF:FFFF)は、システムにより管理されています。

プログラムのローダーは、実行ファイル(.abs)からプログラムがロードされる領域を計算し、その領域をユーザ領域から確保し、その領域に実行コードをダウンロードします。実行コードがダウンロードされる領域がすでに確保されている場合、ローダーは実行ファイルの実行コードをダウンロードできません。その場合は、すでに実行しているアプリケーションを終了し、『UNMAP\_APLBITS』オプションでダウンロードする前に、その領域を開放するか、アドレスのマッピングを変更してください。また、領域が重複する場合は『NOMAP\_APLBITS』を指定して下さい。

また、この領域内にmalloc()等のヒープ領域やRAMディスク領域も含まれています。特に、RAMディスクを確保した場合は、0x8C80:0000番地からマッピングできない場合がありますので、注意してください。

### コーディング

```
int
LoadIPSys(
    DEVID          devID ,
    char           *filename ,
    enum IPFileOption opt
)

```

### コメント

デバイスID  
ファイル名  
ファイルオプション

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*filename

ロードモジュールのファイル名。画像処理コマンドをダウンロードする場合は IPCOMMAND マクロを使用して下さい。

opt

ファイルオプションの指定。下記のファイルパスの指定方法とアプリケーションの領域確保の方法を指定します。ファイルパスの指定方法『DIRECT\_PATH』と『REGISTRY\_PATH』はどちらかを指定し、アプリケーションの領域確保の方法『UNMAP\_APLBITS』と『NOMAP\_APLBITS』はファイルパスの指定方法とORで指定して下さい。

例

```
LoadIPSys( devID, "appli.abs", REGISTRY_PATH | UNMAP_APLBITS )
```

画像データタイプ	対応定数	設定内容
DIRECT_PATH	0x00000000	ロードモジュールのファイルパスを絶対パスまたは、カレントパスで指定
REGISTRY_PATH	0x00000010	レジストリに設定されているロードモジュールのファイルパスでロードモジュールを検索
UNMAP_APLBITS	0x00000100	一度確保したアプリケーションの領域を開放し、アプリケーションの領域を再確保してからダウンロードする
NOMAP_APLBITS	0x00000400	アプリケーションの領域を確保しないでダウンロードする

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0 0	Windowsレジストリアクセスエラー
2 0 1	ファイルオープンエラー
2 0 2	ファイルアクセスエラー
2 0 3	Windowsメモリエラー
2 0 4	Windowsメモリ確保エラー
2 0 5	ロードベリファイエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

画像処理コマンドをダウンロードする場合の例を示します。

```
rtncd = LoadIPSys( devID , IPCOMMAND , REGISTRY_PATH );  
if(rtncd){  
    エラー処理  
}
```

SVP330SDKのインストーラは、画像処理コマンドの実行モジュール

**ipcmd.sys**

をデフォルトで以下の場所にインストールします（インストーラで特にコピーディレクトリを変更しない場合）。そして、レジストリにそのディレクトリのパスをセーブします。

**c:%\$VP900SYS\$SVP330**

本コマンドでファイルオプションに REGISTRY\_PATH を指定するとレジストリにセーブしてあるパスを読んできてそのパスでファイルを検索します。

# Boot IPSys

## 画像処理コマンドのブート

### 機能

LoadIPSys()コマンドでロードしたRAM上の画像処理コマンドを起動し、そのRAM上にロードされた画像処理コマンドに制御を移します。すでにRAM上のコマンドが起動されている場合には、ブートの動作をおこないません。

### コーディング

```
int
BootIPSys(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
3	ハードウェアエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## GetIPDevNumber

## ボード枚数の読み出し

### 機能

インストールされているSVP - 330のボード枚数を返します。

### コーディング

```
int  
GetIPDevNumber(void)
```

### コメント

### パラメータ

なし

### リターン値

インストールされているボード枚数をリターン値として返します。

リターン値	内容
1	インストールされているボード枚数
0	ボードがインストールされていないか、または、ドライバが動作していません。

### 詳細情報

## GetIPDevCount

### オープンボード枚数の読み出し

#### 機能

すでにオープンされているSVP - 330のボード枚数を返します。

#### コーディング

```
int  
GetIPDevCount(void)
```

#### コメント

#### パラメータ

なし

#### リターン値

オープンされているボード枚数をリターン値として返します。

リターン値	内容
1	オープンされているボード枚数
0	オープンされているボードがありません

#### 詳細情報

## GetOpenIPDevCount

## オープン回数の読み出し

### 機能

指定されたボード番号の SVP - 330 がオープンされているかどうか読み出します。そして、すでにオープンされている場合、そのオープンされた回数を返します。

### コーディング

```
int  
GetOpenIPDevCount(  
    DEVID    devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

インストールされているボード枚数をリターン値として返します。

リターン値	内容
1	オープンされている回数
0	まだオープンされていません

### 詳細情報

## EnumAttachIPDev

### 装着ボードの列挙

#### 機能

PCに装着されているボードの枚数とボード番号を列挙します。本コマンドは、OpenIPDev(), OpenIPDevExt() コマンド発行前にも実行可能です。

#### コーディング

```
int  
EnumAttachIPDev(  
    int    *board  
)
```

#### コメント

ボード番号出力

#### パラメータ

##### \*board

装着されているボード枚数分のボード番号が出力されます。装着される可能性があるボード枚数以上の領域を確保して下さい。本コマンドのリターンコードが装着されているボード枚数で、その枚数分のデータ以外は無効です。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	装着されているボード枚数。
0	ボードが装着されていない。

#### 詳細情報

## QueryAttachIPDev

### 装着ボードチェック

#### 機能

パラメータで指定したボードがPCに装着されているかどうかチェックします。本コマンドは、OpenIPDev(), OpenIPDevExt() コマンド発行前にも実行可能です。

#### コーディング

```
int  
QueryAttachIPDev(  
    int    board  
)
```

#### コメント

ボード番号

#### パラメータ

board  
ボード番号 ( 0 ~ 15 )。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	指定ボードが装着されていない。
1	指定ボードが装着されている。

#### 詳細情報

# ClearIPError

## コマンドエラーのクリア

### 機能

コマンドエラーをクリアします。コマンドエラーから復帰するには、必ずコマンドエラーのクリアを行う必要があります。  
尚、本コマンドは画像処理コマンドがブートしてから実行してください。

### コーディング

```
int  
ClearIPError(  
    DEVID    devid  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

# CheckIPError

## コマンドエラーのチェック

### 機能

画像処理コマンドでコマンドエラーが発生したかどうかをチェックします。本コマンドはIPエラーのチェックを行い、リターン値として画像処理コマンドでのコマンドエラーのエラー番号を返します。尚、本コマンドは画像処理コマンドがブートしてから実行してください。

### コーディング

```
int  
CheckIPError(  
    DEVID    devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は画像処理コマンドのエラー番号が int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
それ以外	画像処理コマンドで発生したエラー番号

### 詳細情報

# ReadIPErrorTable

## コマンドエラー情報の取得

### 機能

コマンドエラーの情報を取得します。本コマンドはオンボードCPU上のプログラムで使用するもので、オンボードCPU例外を検出することはできません。

### コーディング

```
int
ReadIPErrorTable(
    DEVID      devID ,
    IPErrorTbl *Tbl
)
```

### コメント

デバイスID  
エラー情報格納テーブル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*Tbl

エラー情報を格納する領域へのポインタ。

### リターン値

リターン値は画像処理コマンドのエラー番号が int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

IPErrorTblのフォーマット及び内容を、下記に示します。

vpxsys.hで宣言しています

```
typedef struct {
    int      ErrorCode;
    int      ErrorINFO;
    char     ErrorRoutin[32];
} IPErrorTbl;
```

エラー情報テーブル	説明
ErrorCode	エラー番号 (コマンドエラーのエラー番号)
ErrorINFO	エラー詳細情報 (現在未使用)
ErrorRoutin	エラー発生関数名 (関数名の文字列がセットされます)

## EnableIPErrorMessage

### エラーメッセージの出力許可

#### 機能

エラーメッセージの出力を許可します。

#### コーディング

```
int  
EnableIPErrorMessage(  
    DEVID    devID  
)
```

#### コメント

デバイスID

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

#### 詳細情報

本コマンドを実行すると、コマンドエラー発生時にメッセージが出力されます。（デフォルト設定）

## DisableIPErrorMessage

### エラーメッセージの出力の禁止

#### 機能

エラーメッセージの出力を禁止します。

#### コーディング

```
int  
DisableIPErrorMessage(  
    DEVID    devID  
)
```

#### コメント

デバイスID

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

#### 詳細情報

本コマンドを実行すると、コマンドエラー発生時のエラーメッセージが出力されなくなります。

# InitIP

## 画像処理システムの初期化

### 機能

画像処理システムを初期化します。また、vpxInitIP( )コマンドは、種々のオプションを指定して画像処理システムの初期化を行うことが可能です。vpxInitIP( )コマンドの詳細は、VP810互換コマンドのvpxInitIP( ) (Ref44-1)を参照して下さい。なお、NTSC標準、YUVカラーカメラ以外を接続して本コマンドを実行した場合、「カメラ選択エラー」が発生する場合があります。その場合は、vpxInitIP( )コマンドを使用しカメラ設定を無効状態にしてシステムを初期化して下さい。

### コーディング

```
int
InitIP(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2	システム異常。ボード未実装。ボード認識不可。レジストリデータ破壊。
3	ハードウェアエラー。カメラ設定エラー。

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画像処理ボードをリセットし、システムを下記状態にイニシャライズします。

	設定項目	設定内容
映像入力表示関連	走査方式	インタレース
	映像入力サイズ	512H×480V
	カメラポート番号	ポート#0
	カメラタイプ	NTSCカラーカメラ
	システムデータタイプ	符号なし8ビット
	カメラトリガ出力	OFF
	表示/非表示	カメラ入力映像の表示
画像制御関連	ウィンドウ	有効(512(X)×480(Y))
	文字描画属性(サイズ)	標準サイズ(16×16)
	正規化関連マスク	無効
	パイプラインモード	解除
	エラーメッセージ表示	ON

システムエラーおよびウィンドウは、リセット状態になります。

# InitIPExt

## 画像処理システムの初期化

### 機能

ハードウェアのテストを行い、画像処理システムを初期化します。また、vpxInitIP( )コマンドは、種々のオプションを指定して画像処理システムの初期化を行うことが可能です。vpxInitIP( )コマンドの詳細は、V P 8 1 0 互換コマンドのvpxInitIP( ) (Ref44-1) を参照して下さい。なお、NTSC標準、YUVカラーカメラ以外を接続して本コマンドを実行した場合、「カメラ選択エラー」が発生する場合があります。その場合は、vpxInitIP( )コマンドを使用しカメラ設定を無効状態にしてシステムを初期化して下さい。

### コーディング

```
int
InitIPExt(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2	システム異常 ( ボード未実装、ボード認識不可、レジスタデータ破壊 )
3	ハードウェアエラー。カメラ設定エラー。
4	レジスタテストエラー
5	コマンドテストエラー
6	画像メモリテストエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## GetIPVersionInfo

## バージョン情報の取得

### 機能

PCドライバ、画像処理コマンド、ボード、ROM等のバージョンを読み出します。

### コーディング

```
int
GetIPVersionInfo(
    DEVID
    VERSIONINFO
)
```

```
devID ,
*VersionInfo
```

### コメント

デバイスID  
バージョン情報構造体

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*VersionInfo

バージョン情報格納領域の先頭アドレス。あらかじめ、sizeof(VERSIONINFO)の領域を確保して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

VERSIONINFOのフォーマットを、下記に示します。

```
typedef struct{
    unsigned long    IPCMD_Env;
    unsigned long    BOARD_Info;
    unsigned long    IPCMD_Version;
    unsigned long    IPCMD_Update;
    unsigned long    ROM_Version;
    unsigned long    ROM_Update;
    unsigned long    PCDRV_Version;
    unsigned long    PCDRV_Update;
    unsigned long    DEVDRV_Version;
    unsigned long    DEVDRV_OS;
    unsigned long    Reserve[6];
} VERSIONINFO;
```

以下にVERSIONINFO構造体のメンバ変数の詳細を示します。

メンバ名	内容																																	
IPCMD_Env	画像処理コマンドの動作環境 <table border="1"> <thead> <tr> <th>対応数</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>ROM上の画像処理コマンドが動作している</td> </tr> <tr> <td>1</td> <td>RAMにダウンロードされた画像処理コマンドが動作している</td> </tr> </tbody> </table>	対応数	内容	0	ROM上の画像処理コマンドが動作している	1	RAMにダウンロードされた画像処理コマンドが動作している																											
対応数	内容																																	
0	ROM上の画像処理コマンドが動作している																																	
1	RAMにダウンロードされた画像処理コマンドが動作している																																	
BOARD_Info	画像処理ボードの情報 <table border="1"> <thead> <tr> <th>予約</th> <th>品種</th> <th>ボード情報</th> </tr> </thead> <tbody> <tr> <td>31</td> <td>24</td> <td>23</td> </tr> <tr> <td>12</td> <td>11</td> <td>0</td> </tr> <tr> <td>品種</td> <td>0x00 : VP910</td> <td>0x20 : VP950C</td> </tr> <tr> <td></td> <td>0x10 : VP910</td> <td>0x30 : VP950S</td> </tr> <tr> <td></td> <td>0x11 : VP910A</td> <td>0x40 : NVP930N</td> </tr> <tr> <td></td> <td></td> <td>0x44 : NVP930CL</td> </tr> <tr> <td></td> <td>0x50 : カスタム</td> <td>0x80 : NVP935N</td> </tr> <tr> <td></td> <td></td> <td>0x84 : NVP935CL</td> </tr> <tr> <td></td> <td>0x8C : NVP935P</td> <td></td> </tr> <tr> <td></td> <td>0x90 : SVP330</td> <td></td> </tr> </tbody> </table>	予約	品種	ボード情報	31	24	23	12	11	0	品種	0x00 : VP910	0x20 : VP950C		0x10 : VP910	0x30 : VP950S		0x11 : VP910A	0x40 : NVP930N			0x44 : NVP930CL		0x50 : カスタム	0x80 : NVP935N			0x84 : NVP935CL		0x8C : NVP935P			0x90 : SVP330	
予約	品種	ボード情報																																
31	24	23																																
12	11	0																																
品種	0x00 : VP910	0x20 : VP950C																																
	0x10 : VP910	0x30 : VP950S																																
	0x11 : VP910A	0x40 : NVP930N																																
		0x44 : NVP930CL																																
	0x50 : カスタム	0x80 : NVP935N																																
		0x84 : NVP935CL																																
	0x8C : NVP935P																																	
	0x90 : SVP330																																	
IPCMD_Version	画像処理コマンド(ボードCPU側)のバージョン情報 <table border="1"> <thead> <tr> <th>Major Version</th> <th>Minor Version</th> <th>Build</th> </tr> </thead> <tbody> <tr> <td>31</td> <td>24</td> <td>23</td> </tr> <tr> <td>16</td> <td>15</td> <td>0</td> </tr> </tbody> </table>	Major Version	Minor Version	Build	31	24	23	16	15	0																								
Major Version	Minor Version	Build																																
31	24	23																																
16	15	0																																
IPCMD_Update	画像処理コマンド(ボードCPU側)の最終更新日 <table border="1"> <thead> <tr> <th>Year</th> <th>Month</th> <th>Day</th> <th>Build</th> </tr> </thead> <tbody> <tr> <td>31</td> <td>24</td> <td>23</td> <td>16</td> </tr> <tr> <td>15</td> <td>8</td> <td>7</td> <td>0</td> </tr> </tbody> </table>	Year	Month	Day	Build	31	24	23	16	15	8	7	0																					
Year	Month	Day	Build																															
31	24	23	16																															
15	8	7	0																															
ROM_Version	ROMのバージョン情報 <table border="1"> <thead> <tr> <th>Major Version</th> <th>Minor Version</th> <th>Build</th> </tr> </thead> <tbody> <tr> <td>31</td> <td>24</td> <td>23</td> </tr> <tr> <td>16</td> <td>15</td> <td>0</td> </tr> </tbody> </table>	Major Version	Minor Version	Build	31	24	23	16	15	0																								
Major Version	Minor Version	Build																																
31	24	23																																
16	15	0																																
ROM_Update	ROMの最終更新日 <table border="1"> <thead> <tr> <th>Year</th> <th>Month</th> <th>Day</th> <th>Build</th> </tr> </thead> <tbody> <tr> <td>31</td> <td>24</td> <td>23</td> <td>16</td> </tr> <tr> <td>15</td> <td>8</td> <td>7</td> <td>0</td> </tr> </tbody> </table>	Year	Month	Day	Build	31	24	23	16	15	8	7	0																					
Year	Month	Day	Build																															
31	24	23	16																															
15	8	7	0																															
PCDRV_Version	PCドライバ(DLL)のバージョン情報 <table border="1"> <thead> <tr> <th>Major Version</th> <th>Minor Version</th> <th>Build</th> </tr> </thead> <tbody> <tr> <td>31</td> <td>24</td> <td>23</td> </tr> <tr> <td>16</td> <td>15</td> <td>0</td> </tr> </tbody> </table>	Major Version	Minor Version	Build	31	24	23	16	15	0																								
Major Version	Minor Version	Build																																
31	24	23																																
16	15	0																																
PCDRV_Update	PCドライバ(DLL)の最終更新日 <table border="1"> <thead> <tr> <th>Year</th> <th>Month</th> <th>Day</th> <th>Build</th> </tr> </thead> <tbody> <tr> <td>31</td> <td>24</td> <td>23</td> <td>16</td> </tr> <tr> <td>15</td> <td>8</td> <td>7</td> <td>0</td> </tr> </tbody> </table>	Year	Month	Day	Build	31	24	23	16	15	8	7	0																					
Year	Month	Day	Build																															
31	24	23	16																															
15	8	7	0																															
DEVDRV_Version	デバイスドライバのバージョン情報 <table border="1"> <thead> <tr> <th>Major Version</th> <th>Minor Version</th> <th>Build</th> </tr> </thead> <tbody> <tr> <td>31</td> <td>24</td> <td>23</td> </tr> <tr> <td>16</td> <td>15</td> <td>0</td> </tr> </tbody> </table>	Major Version	Minor Version	Build	31	24	23	16	15	0																								
Major Version	Minor Version	Build																																
31	24	23																																
16	15	0																																
DEVDRV_OS	デバイスドライバの種類 <table border="1"> <thead> <tr> <th>対応数</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Windows NT デバイスドライバ</td> </tr> <tr> <td>2</td> <td>Windows VxD (仮想デバイスドライバ)</td> </tr> <tr> <td>3</td> <td>Windows WDM (デバイスドライバ)</td> </tr> <tr> <td>4</td> <td>TCP DLL(LAN通信インタフェース)</td> </tr> </tbody> </table>	対応数	内容	1	Windows NT デバイスドライバ	2	Windows VxD (仮想デバイスドライバ)	3	Windows WDM (デバイスドライバ)	4	TCP DLL(LAN通信インタフェース)																							
対応数	内容																																	
1	Windows NT デバイスドライバ																																	
2	Windows VxD (仮想デバイスドライバ)																																	
3	Windows WDM (デバイスドライバ)																																	
4	TCP DLL(LAN通信インタフェース)																																	
Reserve	予約(24バイト)																																	

## CheckIPVersion

## バージョンチェック

### 機能

PCドライバと画像処理コマンドのバージョンをチェックします。PC側にインストールされているDLL (VP900DRV.DLL)のバージョンと現在、画像処理ボードのダウンロードされている実行プログラム (IPCMD.SYS)のバージョンを読み出し、正常に実行できるかどうかチェックします。このコマンドのリターンコードがエラーの場合、PCドライバと画像処理コマンドが正しくインストールされているかどうか確認して下さい。

### コーディング

```
int
CheckIPVersion(
    DEVID
    int
)
```

```
devID ,
mode
```

### コメント

デバイスID  
モード

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### mode

モード (現在未使用)。「0」を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	チェック正常終了。PCドライバと画像処理コマンドのバージョンが合っています。
0	チェックエラー。PCドライバと画像処理コマンドのバージョンが違います。PC側にインストールされているDLLで画像処理ボードのダウンロードされている実行プログラムを正常に実行できません。PCドライバ (VP900DRV.DLL) と画像処理コマンド (IPCMD.SYS) が正しくインストールされているかどうか確認して下さい。

### 詳細情報

# SetIPDataType

## 画像処理システムデータタイプの設定

### 機能

画像処理、カメラ入力で扱う画像のデータのタイプを指定します。

### コーディング

```
int
SetIPDataType(
    DEVID
    enum DataType
)
```

devID ,  
Type

### コメント

デバイスID  
画面データタイプ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

Type

画像データタイプ

画像データタイプ	対応定数	設定内容
SIGN8_DATA	0	符号付8ビット(-128~127)
UNSIGN8_DATA	1	符号なし8ビット(0~255)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	システムデータタイプ設定値範囲外
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

システムデータタイプとは、画像処理、カメラ入力で扱う画像のデータタイプのことを示し、符号付8ビットと符号なし8ビットの2種類が設定できます。

InitIP/InitIPExt実行直後は、符号なし8ビットとなっています。

指定されたデータタイプにより、画像データが次のような値をとるとみなして処理します。

符号なし8ビットの場合

カメラ映像のAD変換取り込みデータ精度は8ビットです。

画像処理のデータ入力は、0~255の数値を取ります。

2値化データの背景画像は0、物体画像は255の値をとります。

符号付8ビットの場合

カメラ映像のAD変換取り込みデータ精度は7ビットです。

画像処理のデータ入出力は、-128~127の数値を取ります。

2値化データの背景画像は0、物体画像は-1の値をとります。

## GetDisplmgID

## ディスプレイ画面の画面番号の取得

### 機能

ディスプレイ画面の画面番号を取得します。

### コーディング

```
int  
GetDisplmgID(  
    DEVID devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

ディスプレイ画面番号は、リターン値で返されます。詳細を下表に示します。

リターン値	内容
1	ディスプレイ画面番号
0	ディスプレイ画面が割り当てられていません
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## GetBitmapImgID

## ビットマップ画面の画面番号の取得

### 機能

ビットマップ画面の画面番号を取得します。

### コーディング

```
int  
GetBitmapImgID(  
    DEVID    devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

ビットマップ画面番号は、リターン値で返されます。詳細を下表に示します。

リターン値	内容
1	ビットマップ画面番号
0	ビットマップ画面が割り当てられていません
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ISP\_BusyWait

## 画像処理プロセッサのビジーウェイト

### 機能

画像処理プロセッサの指定したビジーステータスの処理が終了するまでビジーウェイトを行います。  
 画像処理コマンドを組み合わせる場合、画像処理コマンド内でビジーウェイトを行っていますので、このコマンドでウェイトする必要はありません。ただし、ループバック処理を行っている場合は、適切な位置でビジーウェイトを行って下さい。

### コーディング

```
int
ISP_BusyWait(
    DEVID
    unsigned long
)
```

```
devID ,
sts
```

### コメント

```
デバイスID
ビジーステータス
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**sts**

ビジーステータス。ここで指定したビジーステータスの処理が終了するまでビジーウェイトします。

ビジーステータス	対応数	内容
NOBUSY	0x00000000	ImgID0, ImgID1で指定した画面のビジーフラグのみ
BSY_BUSY	0x00003313	DMA   VP   IP 全てのウェイト
IP_BUSY	0x00000001	画像処理
VP_BUSY	0x00003300	映像入力
VP2_BUSY	0x00003000	映像入力(VP#2)
VP1_BUSY	0x00000300	映像入力(VP#1)
DP_BUSY	0x00030000	映像表示

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ImgBusyWait

## 処理画面に対するビジーウェイト

### 機能

ImgIDで指定した処理画面に対して、画像処理プロセッサの指定したビジーステータスの処理が終了するまでビジーウェイトします。画面の管理テーブルには、以前に処理した画像処理のステータスがあり、そのビジーステータスの処理と指定したビジーステータスの処理が終了するまでビジーウェイトします。そして、ビジーウェイトが終了すると画面の管理テーブルの画像処理のステータスをクリアします。

画像処理コマンドを組み合わせる場合、画像処理コマンド内でビジーウェイトを行っていますので、このコマンドでウェイトする必要はありません。

### コーディング

```
int
ImgBusyWait(
    DEVID
    unsigned long
    int
)
```

```
devID ,
sts ,
ImgID
```

### コメント

```
デバイスID
ビジーステータス
画面番号
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**sts**

ビジーステータス。ImgIDで指定した画面のビジーフラグとここで指定したビジーステータスの処理が終了するまでビジーウェイトします。

ビジーステータス	対応数	内容
NOBUSY	0x00000000	ImgID0, ImgID1で指定した画面のビジーフラグのみ
BSY_BUSY	0x00003313	DMA   VP   IP 全てのウェイト
IP_BUSY	0x00000001	画像処理
VP_BUSY	0x00003300	映像入力
VP2_BUSY	0x00003000	映像入力(VP#2)
VP1_BUSY	0x00000300	映像入力(VP#1)
DP_BUSY	0x00030000	映像表示

**ImgID**

画面番号。ビジーウェイトの対象画面を指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# Img2chBusyWait

## 2ch処理画面に対するビジーウェイト

### 機能

それぞれ別のチャンネルのImgID0, ImgID1で指定した処理画面に対して、画像処理プロセッサの指定したビジーステータスの処理が終了するまでビジーウェイトします。画面の管理テーブルには、以前に処理した画像処理のステータスがあり、そのビジーステータスの処理と指定したビジーステータスの処理が終了するまでビジーウェイトします。そして、ビジーウェイトが終了すると画面の管理テーブルの画像処理のステータスをクリアします。

画像処理コマンドを組み合わせる場合、画像処理コマンド内でビジーウェイトを行っていますので、このコマンドでウェイトする必要はありません。

### コーディング

```
int
Img2chBusyWait(
    DEVID
    unsigned long
    int
    int
)
```

```
devID ,
sts ,
ImgID0 ,
ImgID1
```

### コメント

```
デバイスID
ビジーステータス
画面番号#0
画面番号#1
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**sts**

ビジーステータス。ImgID0, ImgID1で指定した画面のビジーフラグとここで指定したビジーステータスの処理が終了するまでビジーウェイトします。

ビジーステータス	対応数	内容
NOBUSY	0x00000000	ImgID0, ImgID1で指定した画面のビジーフラグのみ
BSY_BUSY	0x00003313	DMA   VP   IP 全てのウェイト
IP_BUSY	0x00000001	画像処理
VP_BUSY	0x00003300	映像入力
VP2_BUSY	0x00003000	映像入力(VP#2)
VP1_BUSY	0x00000300	映像入力(VP#1)
DP_BUSY	0x00030000	映像表示

**ImgID0**

画面番号。ビジーウェイトの対象画面を指定して下さい。

**ImgID1**

画面番号。ビジーウェイトの対象画面を指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ISP\_BusyCheck

## ビジーステータスの読み出し

### 機能

画像処理プロセッサのビジーステータスを読み出します。

### コーディング

```
int
ISP_BusyCheck(
    DEVID
    unsigned long
)
```

devID ,  
\*sts

### コメント

デバイスID  
ビジーステータス出力

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*sts

ビジーステータス。以下のビジーステータスが読み出されます。

ビジーステータス	対応数	内容
NOBUSY	0x00000000	ImgID0, ImgID1で指定した画面のビジーフラグのみ
BSY_BUSY	0x00003313	DMA   VP   IP 全てのウェイト
IP_BUSY	0x00000001	画像処理
VP_BUSY	0x00003300	映像入力
VP2_BUSY	0x00003000	映像入力(VP#2)
VP1_BUSY	0x00000300	映像入力(VP#1)
DP_BUSY	0x00030000	映像表示

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# AllocImg

## 画像メモリ領域確保

### 機能

画像メモリに、パラメータで指定されたサイズの領域を確保します。

SVP-330では、ハードウェアの制約により640(X)×512(Y)より大きい画面サイズでの処理は、サポートされていないので注意して下さい。

### コーディング

```
int
AllocImg(
    DEVID
    enum ImageFrameSize size
)
```

devID ,

### コメント

デバイスID  
画像メモリ画面サイズ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

size

画面サイズ

画面サイズ	対応定数	X方向	Y方向
IMG_FS_256H_256V	0	256	256
IMG_FS_512H_256V	1	512	256
IMG_FS_256H_512V	2	256	512
IMG_FS_512H_512V	3	512	512
IMG_FS_640H_256V	4	640	256
IMG_FS_640H_512V	5	640	512
IMG_FS_1024H_512V	6	1024	512
IMG_FS_1024H_1024V	7	1024	1024
IMG_FS_1024H_256V	8	1024	256
IMG_FS_1280H_1024V	9	1280	1024
IMG_FS_1280H_256V	10	1280	256
IMG_FS_1280H_512V	11	1280	512
IMG_FS_1024H_768V	12	1024	768
IMG_FS_2048H_2048V	16	2048	2048
IMG_FS_2048H_1024V	17	2048	1024
IMG_FS_1920H_1024V	18	1920	768

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	確保した画面番号
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	画面サイズ設定値範囲外
4 6	画面領域空きなし
-	エラーリセットコマンド (ClearIPErrors) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

本コマンドで画像領域を確保した場合の画像設定状態を、下記に示します。

項 目	設 定 値
チャンネル	不定
座標	不定
領域サイズ	パラメータで指定
データ属性	システムデータタイプによる
システムプロセッサからの画像メモリアクセス	アクセス不可

# AllocLockImg

## 画像メモリ領域確保(固定領域)

### 機能

画像メモリに、パラメータで指定されたチャンネル・ページ(256単位の開始座標)・サイズの領域を確保します。本コマンドはVP-810専用のコマンドです。VP-810以外では、チャンネルNo、ページNoは無効になり、AllocImg( )コマンドと同じ動作になります。

SVP-330では、ハードウェアの制約により640(X)×512(Y)より大きい画面サイズでの処理は、サポートされていませんので注意して下さい。

### コーディング

```
int
AllocLockImg(
    DEVID          devID ,
    enum ImageFrameSize size ,
    enum CHID      chid ,
    enum PageX_Y   sx ,
    enum PageX_Y   sy
)
```

### コメント

デバイスID  
画像メモリ画面サイズ  
画像メモリチャンネル指定  
画像配置位置設定  
画像配置位置設定

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### size

画面サイズ

画面サイズ	対応定数	X方向	Y方向
IMG_FS_256H_256V	0	256	256
IMG_FS_512H_256V	1	512	256
IMG_FS_256H_512V	2	256	512
IMG_FS_512H_512V	3	512	512
IMG_FS_640H_256V	4	640	256
IMG_FS_640H_512V	5	640	512
IMG_FS_1024H_512V	6	1024	512
IMG_FS_1024H_1024V	7	1024	1024
IMG_FS_1024H_256V	8	1024	256
IMG_FS_1280H_1024V	9	1280	1024
IMG_FS_1280H_256V	10	1280	256
IMG_FS_1280H_512V	11	1280	512
IMG_FS_1024H_768V	12	1024	768
IMG_FS_2048H_2048V	16	2048	2048
IMG_FS_2048H_1024V	17	2048	1024
IMG_FS_1920H_1024V	18	1920	768

chid

チャンネルNo.

チャンネルNo.	対応定数	内容
IMG_CH0	0	画像メモリチャンネル#0
IMG_CH1	1	画像メモリチャンネル#1
IMG_CH2	2	画像メモリチャンネル#2
IMG_CH3	3	画像メモリチャンネル#3
IMG_CH4	4	画像メモリチャンネル#4

sx

X方向ページNo.

sy

Y方向ページNo.

ページNo.	対応定数	内容
PAGE_0	0	0~
PAGE_1	1	2 5 6 ~
PAGE_2	2	5 1 2 ~
PAGE_3	3	7 6 8 ~
PAGE_4	4	1 0 2 4 ~
PAGE_5	5	1 2 8 0 ~
PAGE_6	6	1 5 3 6 ~
PAGE_7	7	1 7 9 2 ~

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	確保した画面番号
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	画面サイズ設定値範囲外
2 1	チャンネルNo.設定値範囲外
2 2	X方向ページNo.設定値範囲外
2 3	Y方向ページNo.設定値範囲外
4 6	画像領域空きなし
4 7	使用中の画像メモリ領域を指定した(画像メモリ重複)
-	エラーリセットコマンド(ClearLError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

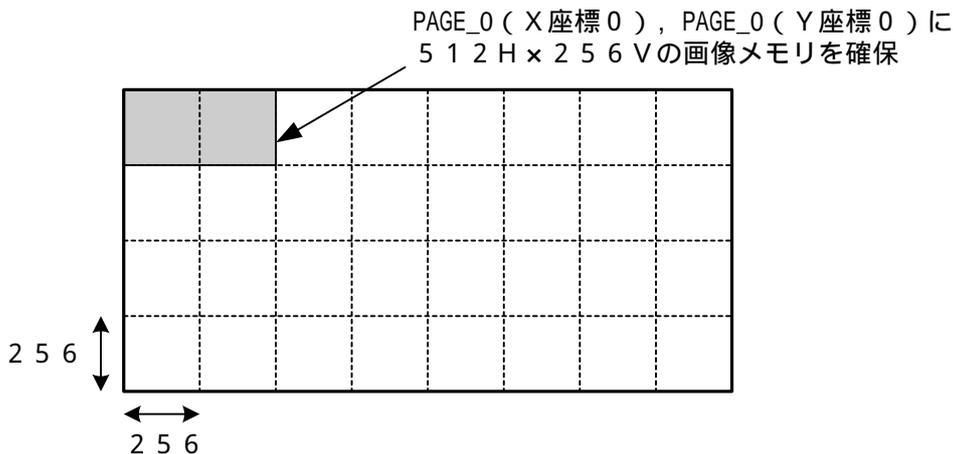
本コマンドで画面領域を確保した場合の画面設定状態を、下記に示します。

項 目	設 定 値
チャンネル	パラメータで指定
座標	パラメータで指定
領域サイズ	パラメータで指定
データ属性	システムデータタイプによる
システムプロセッサからの画像メモリアクセス	アクセス不可

画像処理ボードの画像メモリサイズは2048H×1024Vであり、これが5チャンネル(5枚)存在します。本コマンドでは、この画像メモリの任意の位置を固定的に確保します。

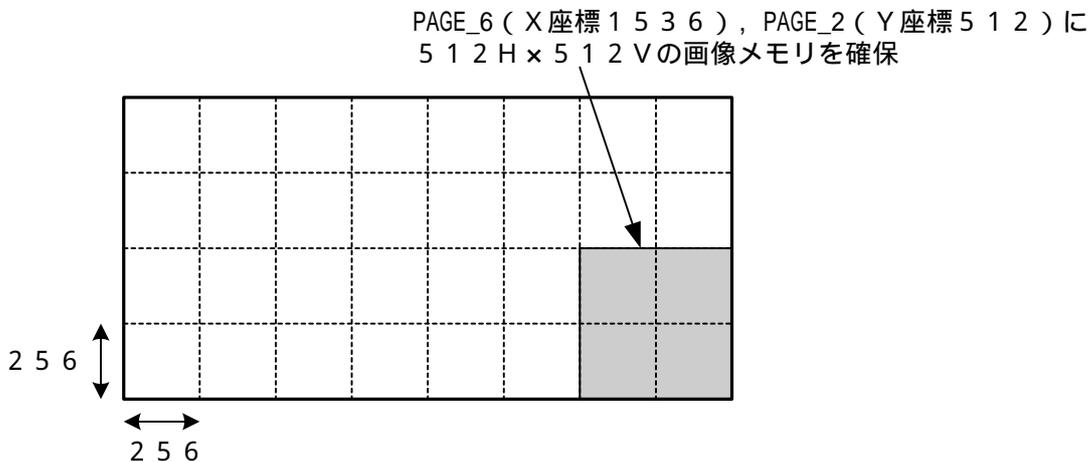
例1) チャンネル0に512H×256Vを確保

```
id = AllocLockImg(IMG_FS_512H_256V, IMG_CH0, PAGE_0, PAGE_0);
```



例2) チャンネル4に512H×512Vを確保

```
id = AllocLockImg(IMG_FS_512H_512V, IMG_CH4, PAGE_6, PAGE_2);
```



# FreeImg

## 画像メモリ領域確保(指定画面)

### 機能

パラメータで指定された画面No.の画像メモリ領域を開放します。

### コーディング

```
int
FreeImg(
    DEVID    devID ,
    int      ImgID
)
```

### コメント

デバイスID  
画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID

解放する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外(不当画面番号エラー)
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# FreeAllImg

## 画像メモリ領域確保(全画面)

### 機能

全ての画面の画像メモリ領域を開放します。

### コーディング

```
int
FreeAllImg(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ReadImgTable

## 画像メモリ管理テーブル読み出し

### 機能

パラメータで指定された画面No.の画像メモリ管理テーブルを読み出し、ユーザテーブルに格納します。

### コーディング

```
int
ReadImgTable(
    DEVID    devID ,
    int      ImgID ,
    IMG_TBL  *tbl
)
```

### コメント

デバイスID  
画面番号  
画像メモリ管理テーブル

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
対象画面の画面番号

**\*tbl**  
画像メモリ管理テーブルを格納する領域のポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

IMG\_TBLのフォーマットは、次のとおりです。

また、内容は次紙を参考にして下さい。

vpxsys.hで宣言しています

```
typedef struct {
    int      ch;
    int      sx;
    int      sy;
    int      size;
    int      xlng;
    int      ylng;
    int      dtyp;
    int      lock;
    int      open;
    int      dirty;
    int      color;
    int      arrange;
    int      next_id;
} IMG_TBL;
```

名称	意味
ch	画像メモリチャンネル番号 0: チャンネル0 1: チャンネル1 2: チャンネル2 3: チャンネル3 4: チャンネル4
sx	画面開始X座標(物理アドレス)
sy	画面開始Y座標(物理アドレス)
size	画面サイズ番号(確保した画像メモリ領域サイズ(enum ImageFrameSize)に対応)
xlng	画面のX方向の長さ
ylng	画面のY方向の長さ
dtype	画面のデータタイプ 0: 符号付8ビット(-128~127) 1: 符号なし8ビット(0~255) 2: 2値(0または255) -1: 不定画面
lock	画面ロックフラグ 0: AllocImgで確保したため再配置可 1: AllocLockImgで確保したため再配置不可
open	オープン制御フラグ リザーブ
dirty	画面更新制御フラグ リザーブ
color	リザーブ
arrange	リザーブ
next_id	リザーブ

# ChangeImgDataType

## 画像データタイプ変更

### 機能

各画面のもつデータタイプの変更を行います。

### コーディング

```
int
ChangeImgDataType(
    DEVID
    int
    enum DataType
)
```

```
devID ,
ImgID ,
type
```

### コメント

```
デバイスID
画面番号
画像データタイプ
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID

変更対象の画面番号

type

画像データタイプ

画像データタイプ	対応定数	設定内容
SIGN8_DATA	0	符号付8ビット(-128~127)
UNSIGN8_DATA	1	符号なし8ビット(0~255)
BINARY_DATA	2	2値(0または255)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10	指定画面空き領域、または範囲外(不当画面番号エラー)
21	データタイプ設定値範囲外
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetWindow

## ウィンドウ設定

### 機能

パラメータで指定された種類・サイズに従い、ウィンドウを設定します。

### コーディング

```
int
SetWindow(
    DEVID
    enum WindowType
    int
    int
    int
    int
    int
)
```

```
devID ,
type ,
sx ,
sy ,
ex ,
ey
```

### コメント

デバイス I D  
画面処理のウィンドウタイプ  
始点 X 座標  
始点 Y 座標  
終点 X 座標  
終点 Y 座標

### パラメータ

**devID**

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**type**

ウィンドウの種類

ウィンドウ種類	対応定数	説明
SRC0_WIN	0	画像処理のソース # 0 画面ウィンドウ
SRC1_WIN	1	画像処理のソース # 1 画面ウィンドウ
DST_WIN	3	画像処理のデスティネーション画面ウィンドウ
SYS_WIN	4	システムプロセッサアクセス画面ウィンドウ

**sx**

始点 X 座標

**sy**

始点 Y 座標

**ex**

終点 X 座標

**ey**

終点 Y 座標

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	ウィンドウ種類設定値範囲外
2 1	始点X座標範囲外
2 2	始点Y座標範囲外
2 3	終点X座標範囲外、始点X座標 > 終点X座標と設定
2 4	終点Y座標範囲外、始点Y座標 > 終点Y座標と設定
-	エラーリセットコマンド (ClearIPError) 未発行

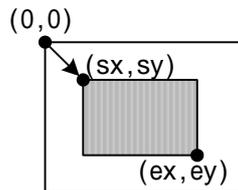
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

ウィンドウとは、画像処理領域のことです。ウィンドウを設定することにより、画像処理領域が小さくなるため、処理時間が短縮されます。

ウィンドウサイズに上限はありませんが、画像領域を超えるウィンドウサイズを設定した場合、画像領域とウィンドウが重なりあった領域のみ処理対象領域となります。下限は2として下さい。2未満でウィンドウを設定した場合は、正しい画像処理結果が得られない場合があります。

ウィンドウ座標は画面原点相対で設定して下さい。



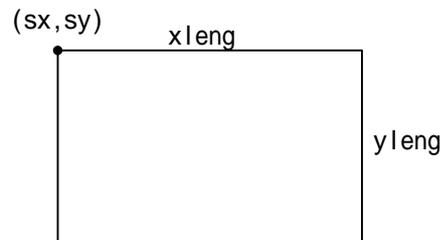
sx, sy, ex, eyで指定される領域は、

sx	region x	ex
sy	region y	ey

です。つまり、sx, syの座標とex, eyの座標を含むということです。

(sx, sy)を始点にxleng, ylengの大きさの範囲で処理を行う場合の(ex, ey)の算出は、

$ex = sx + xleng - 1$
$ey = sy + yleng - 1$



になります。

# SetAllWindows

## ウィンドウ設定(全種)

### 機能

パラメータで指定されたサイズに従い、全てのウィンドウを一括設定します。

### コーディング

```
int
SetAllWindows(
    DEVID    devID ,
    int      sx ,
    int      sy ,
    int      ex ,
    int      ey
)
```

### コメント

デバイス I D  
始点 X 座標  
始点 Y 座標  
終点 X 座標  
終点 Y 座標

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**sx**  
始点 X 座標

**sy**  
始点 Y 座標

**ex**  
終点 X 座標

**ey**  
終点 Y 座標

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	始点 X 座標範囲外
2 1	始点 Y 座標範囲外
2 2	終点 X 座標範囲外、始点 X 座標 > 終点 X 座標と設定
2 3	終点 Y 座標範囲外、始点 Y 座標 > 終点 Y 座標と設定
-	エラーリセットコマンド (ClearIPError) 未発行

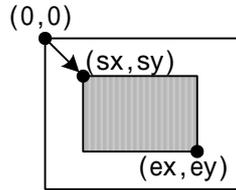
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ウィンドウとは、画像処理領域のことです。ウィンドウを設定することにより、画像処理領域が小さくなるため、処理時間が短縮されます。

ウィンドウサイズに上限はありませんが、画像領域を超えるウィンドウサイズを設定した場合、画像領域とウィンドウが重なりあった領域のみ処理対象領域となります。下限は2として下さい。2未満でウィンドウを設定した場合は、正しい画像処理結果が得られない場合があります。

ウィンドウ座標は画面原点相対で設定して下さい。



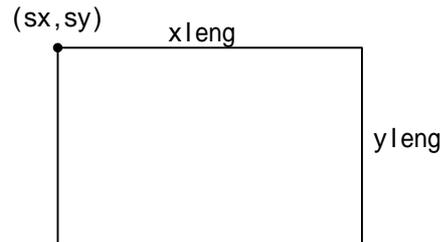
sx, sy, ex, eyで指定される領域は、

sx	region x	ex
sy	region y	ey

です。つまり、sx, syの座標とex, eyの座標を含むということです。

(sx, sy)を始点にxleng, ylengの大きさの範囲で処理を行う場合の(ex, ey)の算出は、

$ex = sx + xleng - 1$
$ey = sy + yleng - 1$



になります。

## ResetAllWindow

## ウィンドウリセット

### 機能

全てのウィンドウ設定を解除し、現在のビデオフレームサイズに従い設定します。

### コーディング

```
int  
ResetAllWindow(  
    DEVID    devid  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## EnableIPWindow

## ウィンドウ有効化

### 機能

全ウィンドウを有効にします。本コマンド発行後の処理（表示系を除く）は、ウィンドウサイズで実行されます。

### コーディング

```
int
EnableIPWindow(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## DisableIPWindow

## ウィンドウ無効化

### 機能

全ウィンドウを無効にします。本コマンド発行後の処理（表示系を除く）は、確保したビデオフレームサイズで実行されます。

### コーディング

```
int  
DisableIPWindow(  
    DEVID    devid  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ReadWindow

## ウィンドウ設定座標読み出し

### 機能

パラメータで設定されたウィンドウ種類のウィンドウサイズを読み出します。

### コーディング

```
int
ReadWindow(
  DEVID
  enum WindowType
  int
  int
  int
  int
)
```

```
devID ,
type ,
*sx ,
*sy ,
*ex ,
*ey
```

### コメント

デバイスID  
画面処理のウィンドウタイプ  
始点X座標  
始点Y座標  
終点X座標  
終点Y座標

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**type**

ウィンドウの種類

ウィンドウ種類	対応定数	説明
SRCO_WIN	0	画像処理のソース# 0画面ウィンドウ
SRC1_WIN	1	画像処理のソース# 1画面ウィンドウ
DST_WIN	3	画像処理のデスティネーション画面ウィンドウ
SYS_WIN	4	システムプロセッサアクセス画面ウィンドウ
DST_EXT_WIN	6	画像処理のデスティネーション拡張画面ウィンドウ

**\*sx**

始点X座標を格納する領域のポインタ

**\*sy**

始点Y座標を格納する領域のポインタ

**\*ex**

終点X座標を格納する領域のポインタ

**\*ey**

終点Y座標を格納する領域のポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	ウィンドウ種類設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# AllocDispImg

## ディスプレイ画面の確保

### 機能

カメラ映像表示のディスプレイの画面を確保します。画像処理ボードでは、カメラ映像をダイレクトに表示することはできません。そのため、カメラ映像を表示する場合は一旦、画像メモリにカメラ映像を入力しその入力した映像を表示する必要があります。本コマンドは、DispCamera()コマンドでそのカメラ映像をダイレクトに表示する為の画像メモリを確保します。また、「BMP\_DISPLAY」のディスプレイモードを選択することにより、オーバーレイ表示の画面を自動的に確保し画像処理コマンド内部で自動的に管理します。また、映像画面サイズを変更する場合は、FreeDispImg()コマンドでカメラ映像表示画面を解放し、その後AllocDispImg()コマンドで任意のサイズで映像画面を再度確保して下さい。なお、確保した画面番号は、GetDispImg(),GetBitmapImgID()コマンドで参照できます。

InitIP( )コマンドを発行すると、画面サイズ「IMG\_FS\_512H\_512V」、ディスプレイモード「YUV\_DISPLAY」でAllocDispImg()が実行された状態になります。

### コーディング

```
int
AllocDispImg(
    DEVID
    enum ImageFrameSize
    enum DispType
)
```

```
devID ,
size ,
type
```

### コメント

```
デバイスID
画像メモリ画面サイズ
ディスプレイモード
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

size

画面サイズ

画面サイズ	対応定数	X方向	Y方向
IMG_FS_256H_256V	0	2 5 6	2 5 6
IMG_FS_512H_256V	1	5 1 2	2 5 6
IMG_FS_256H_512V	2	2 5 6	5 1 2
IMG_FS_512H_512V	3	5 1 2	5 1 2
IMG_FS_640H_256V	4	6 4 0	2 5 6
IMG_FS_640H_512V	5	6 4 0	5 1 2
IMG_FS_1024H_512V	6	1 0 2 4	5 1 2
IMG_FS_1024H_1024V	7	1 0 2 4	1 0 2 4
IMG_FS_1024H_256V	8	1 0 2 4	2 5 6
IMG_FS_1280H_1024V	9	1 2 8 0	1 0 2 4
IMG_FS_1280H_256V	1 0	1 2 8 0	2 5 6
IMG_FS_1280H_512V	1 1	1 2 8 0	5 1 2
IMG_FS_1024H_768V	1 2	1 0 2 4	7 6 8
IMG_FS_2048H_2048V	1 6	2 0 4 8	2 0 4 8
IMG_FS_2048H_1024V	1 7	2 0 4 8	1 0 2 4
IMG_FS_1920H_1024V	1 8	1 9 2 0	7 6 8

type

ディスプレイモード

ディスプレイモード	説明
BW_DISPLAY	モノクロディスプレイ
YUV_DISPLAY	カラーディスプレイ
BMP_DISPLAY	オーバーレイ付きモノクロディスプレイ

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	確保した画面番号
0	既にディスプレイ画面が確保されている
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	画面サイズ設定値範囲外
2 1	ディスプレイモード設定値範囲外
4 6	画面領域空きなし
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

映像画面の変更は以下の要領で行います。以下に、映像画面を 640 × 480 に切り換える例を示します。

```

/* 表示画面の解放 */
FreeDispImg(_devID);
/* 表示画面(640×480)の確保 */
AllocDispImg(_devID, IMG_FS_640H_512V, BMP_DISPLAY);
/* 表示画面(640×480)のビデオモード設定 */
SetDispFrame(_devID, INTERLACE, DISP_FS_640H_480V);

/* 映像入力フレーム(640×480)の設定 */
SetVideoFrame(_devID, INTERLACE, VIDEO_FS_640H_480V);

/* 映像入力 / 処理画面の確保 */
ImgID0 = AllocLockImg(_devID, IMG_FS_640H_512V, IMG_CH0, PAGE_0, PAGE_0);
ImgID1 = .....

```

# FreeDisplmg

## ディスプレイ画面の開放

### 機能

カメラ映像表示ディスプレイの画面を開放します。映像画面サイズを変更する場合は、FreeDisplmg() コマンドでカメラ映像表示画面を解放し、その後AllocDisplmg() コマンドで任意のサイズで映像画面を確保して下さい。なお、本コマンドでカメラ映像表示画面を解放した状態ではDispCamera() コマンドは実行できませんので注意して下さい。

### コーディング

```
int
FreeDisplmg(
    DEVID    devid
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## SetVideoFrame

## 映像入力画面設定

## 機能

カメラ、映像入力での走査方式、映像入力サイズ（ビデオフレームサイズ）を、パラメータに従い設定します。

カメラに対応した走査方式と画面サイズについては、付録1をご覧ください。

## コーディング

```
int
SetVideoFrame(
    DEVID
    enum Interlace
    enum VideoFrameSize
)
```

```
devID ,
mode ,
size
```

```
デバイスID
ビデオ画面規定（インタレース）
ビデオ画面規定（ビデオフレームサイズ）
```

## コメント

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

走査方式

走査方式	対応定数	内容
NONINTERLACE	0	ノンインタレース
INTERLACE	1	インタレース
NONINTERLACE_FAST	2	ノンインタレース&任意フィールド
NONINTERLACE_1N	3	プログレッシブスキャン

size

映像入出力サイズ（ビデオフレームサイズ）

映像入出力サイズ	対応定数	映像入力サイズ		備考
		H	V	
VIDEO_FS_256H_220V	0	256	220	
VIDEO_FS_512H_220V	1	512	220	
VIDEO_FS_256H_440V	2	256	440	
VIDEO_FS_512H_440V	3	512	440	
VIDEO_FS_256H_240V	4	256	240	
VIDEO_FS_320H_240V	5	320	240	
VIDEO_FS_512H_240V	6	512	240	
VIDEO_FS_640H_240V	7	640	240	
VIDEO_FS_256H_480V	8	256	480	
VIDEO_FS_320H_480V	9	320	480	
VIDEO_FS_512H_480V	1 0	512	480	デフォルト
VIDEO_FS_640H_480V	1 1	640	480	
VIDEO_FS_512H_512V	1 2	512	512	
VIDEO_FS_1008H_470V	1 3	1008	470	高精細カメラ
VIDEO_FS_1024H_1024V	1 4	1024	1024	高精細カメラ
VIDEO_FS_1024H_240V	1 5	1024	240	高精細カメラ
VIDEO_FS_1024H_480V	1 6	1024	480	高精細カメラ
VIDEO_FS_1280H_1024V	1 7	1280	1024	高精細カメラ
VIDEO_FS_1280H_240V	1 8	1280	240	高精細カメラ
VIDEO_FS_1280H_480V	1 9	1280	480	高精細カメラ
VIDEO_FS_1024H_768V	2 0	1280	768	高精細カメラ
VIDEO_FS_1920H_1024V	2 1	1920	1024	高精細カメラ
VIDEO_FS_2048H_1024V	2 2	2048	1024	高精細カメラ
VIDEO_FS_1280H_256V	2 3	1280	256	高精細カメラ

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	走査方式設定値範囲外
2 1	映像入出力サイズ規定値判定外
-	エラーリセットコマンド (ClearLError) 未発行

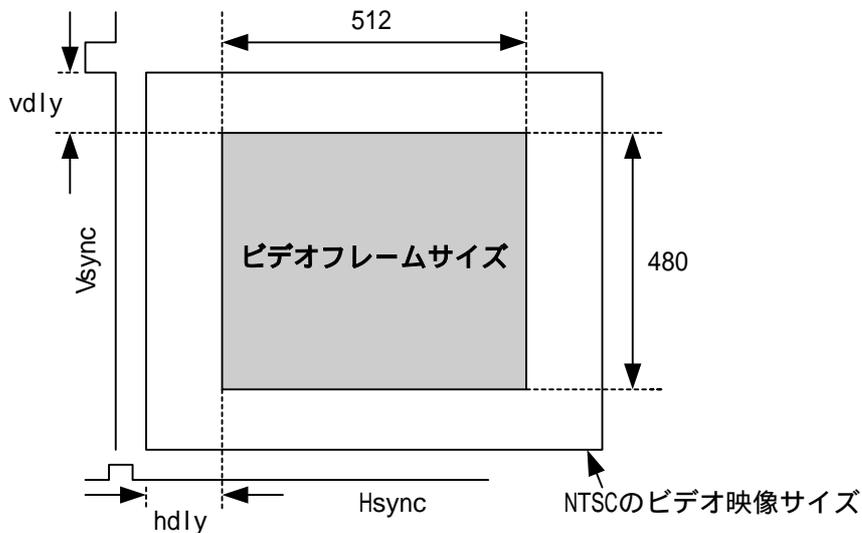
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

映像入力、あるいは表示の対象となる画像メモリの画面サイズは、ビデオフレームサイズと一致している必要があります。

NTSC準拠のカメラ (インタレース) を接続してノンインタレースを指定すると、

- ・カメラ映像入力では、カメラ映像の第1フィールドのみ取り込まれ、縦方向に1/2圧縮した映像となります。
- ・画像メモリの映像表示では、逆に縦方向に2倍に拡大されて表示されます。  
(第1フィールドの映像を第2フィールドにも出力するため)
- ・画面の四個サイズに256Hを指定すると
  - 1) カメラ映像入力では、カメラ映像が縦方向に1/2に圧縮されて取り込まれます。
  - 2) 画像メモリの映像表示では、逆に横方向に2倍にされて表示されます。



映像画面の変更は以下の要領で行います。以下に、映像画面を640×480に切り換える例を示します。

```

/* 表示画面の解放 */
FreeDispImg(_devID);
/* 表示画面(640×480)の確保 */
AllocDispImg(_devID, IMG_FS_640H_512V, BMP_DISPLAY);
/* 表示画面(640×480)のビデオモード設定 */
SetDispFrame(_devID, INTERLACE, DISP_FS_640H_480V);

/* 映像入力フレーム(640×480)の設定 */
SetVideoFrame(_devID, INTERLACE, VIDEO_FS_640H_480V);

/* 映像入力 / 処理画面の確保 */
ImgID0 = AllocLockImg(_devID, IMG_FS_640H_512V, IMG_CH0, PAGE_0, PAGE_0);
ImgID1 = .....

```

# SelectCamera

## カメラ番号とカメラタイプの選択

### 機能

設定されたカメラポート番号にカメラ入力を切り替えます。コマンド内部で SetVideoFrame コマンドと SetDispFrame が指定されたカメラタイプにマッチしたパラメータで実行されます。

### コーディング

```
int
SelectCamera(
    DEVID
    enum CameraID
    enum CameraType
)
```

```
devID ,
id ,
type
```

### コメント

```
デバイスID
カメラポート番号
カメラタイプ
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

id

カメラポート番号

カメラポートNo.	対応定数	内容
CAMERA_PORT0	0	カメラポート#0 (フロントパネルCH#1に対応しています)
CAMERA_PORT1	1	カメラポート#1 (フロントパネルCH#2に対応しています)

type

カメラタイプ

(1) SVP-330(NTSC対応版)

カメラタイプNo.	対応定数	内容	備考
BW_CAMERA	0	モノクロカメラ	
YUV_CAMERA	1	CVBSカラーカメラ	デフォルト
BW_FRESET_CAMERA	3 1	モノカメラ : 外部同期フレームリセットモード	

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
3	ハードウェアエラー。カメラ設定エラー。
2 0	カメラポートNo.
2 1	カメラタイプ設定値範囲外
-	エラーリセットコマンド (ClearIPErrors) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

- ・本コマンドによりカメラタイプを変更すると、その動作により、同期信号が外部同期からカメラ同期に切り替わり、ビットマップ画面がランダムなノイズにより描画される場合があります。その場合は、画面クリアコマンド等でビットマップをクリアするか、BitmapOverlap()コマンドでビットマップ画面をOFF（非表示）にして下さい。
- ・本コマンド発行時のパラメータ（カメラポート番号、カメラタイプ）と現状のパラメータに変更がない場合は、画像処理ボードのカメラ設定を行いません。再設定を行う必要がある場合は、SetCameraPortConfig()コマンド発行後にSelectCamera()を行って下さい。

# GetCamera

## カメラ映像入力

### 機能

カメラ映像を指定された画面に入力します。フレームシャッターカメラでのトリガ入力の場合は、SetTrigerMode コマンドと合わせて使用します。

### コーディング

```
int
GetCamera(
    DEVID    devID ,
    int      ImgID
)
```

### コメント

デバイス I D  
画面番号

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

ImgID

カメラ映像を入力する画面の画面番号。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
4 2	カメラ映像入力画面サイズと、ビデオフレームサイズミスマッチ
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドで映像を入力する際に、画面サイズと映像入力サイズ (ビデオフレームサイズ) を一致させておく必要があります。なお、不一致の場合はエラーとなり、本処理は行われません。

対応画面サイズ一覧

映像入力 フレームサイズ	映像入力サイズ		画面サイズ	画像メモリ	
	H	V		X方向	Y方向
VIDEO_FS_256H_220V	256	220	IMG_FS_256H_256V	256	256
VIDEO_FS_256H_240V	256	240			
VIDEO_FS_512H_220V	512	220	IMG_FS_512H_256V	512	256
VIDEO_FS_512H_240V	512	240			
VIDEO_FS_320H_240V	320	240			
VIDEO_FS_256H_440V	256	440	IMG_FS_256H_512V	256	512
VIDEO_FS_256H_480V	256	480			
VIDEO_FS_512H_440V	512	440	IMG_FS_512H_512V	512	512
VIDEO_FS_512H_480V	512	480			
VIDEO_FS_320H_480V	320	480			
VIDEO_FS_512H_512V	512	512			
VIDEO_FS_640H_240V	640	240	IMG_FS_640H_256V	640	256
VIDEO_FS_640H_480V	640	480	IMG_FS_640H_512V	640	512
VIDEO_FS_1024H_240V	1024	240	IMG_FS_1024H_256V	1024	256
VIDEO_FS_1024H_480V	1024	480	IMG_FS_1024H_512V	1024	512
VIDEO_FS_1008H_470V	1008	470			
VIDEO_FS_1024H_1024V	1024	1024	IMG_FS_1024H_1024V	1024	1024
VIDEO_FS_1280H_240V	1280	240	IMG_FS_1280H_256V	1280	256
VIDEO_FS_1280H_480V	1280	480	IMG_FS_1280H_512V	1280	512
VIDEO_FS_1280H_1024V	1280	1024	IMG_FS_1280H_1024V	1280	1024
VIDEO_FS_1024H_768V	1024	768	IMG_FS_1024H_768V	1024	768

## GetCameraWithSelectPort

カメラ映像入力(カメラ番号とカメラタイプの選択付き)

### 機能

カメラ映像を指定された画面に入力します。オンボードCPU上で、SelectCamera コマンドと GetCamera コマンドを組み合わせたコマンドで、SelectCamera コマンドと GetCamera コマンド間のPCドライバによるタイムラグをなくしたコマンドです。詳細については、SelectCamera、GetCameraコマンドを参照して下さい。

### コーディング

```
int
GetCameraWithSelectPort(
    DEVID          devID ,
    enum CameraID  id ,
    enum CameraType type ,
    int            imgID
)
```

### コメント

デバイスID  
カメラポート番号  
カメラタイプ  
画面番号

### パラメータ

- devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。
- id**  
カメラポート番号。詳細は SelectCamera コマンドを参照して下さい。
- type**  
カメラタイプ。詳細は SelectCamera コマンドを参照して下さい。
- imgID**  
カメラ映像を入力する画面の画面番号。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 2	カメラ映像入力画面サイズと、ビデオフレームサイズミスマッチ
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドで映像を入力する際に、画面サイズと映像入力サイズ（ビデオフレームサイズ）を一致させておく必要があります。なお、不一致の場合はエラーとなり、本処理は行われません。

本コマンドでカメラタイプを切り換えた場合やカメラによっては、同期がずれる場合がありますので注意して下さい。

## SetCameraSync

### カメラ同期信号の設定

#### 機能

N T S C 標準カメラの同期信号の設定を行います。EXTERNAL\_SYNC/CAMERA\_SYNCモードでは、同期信号の切り換えはハードにより自動的に行われるため、実際には動作しません。

#### コーディング

```
int
SetCameraSync(
    DEVID
    enum CameraSync
)
```

devID ,  
sync

#### コメント

デバイス I D  
カメラの同期タイプ

#### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

sync

カメラの同期タイプ

カメラの同期タイプ	説明
EXTERNAL_SYNC	画像処理ボードの同期信号による同期
CAMERA_SYNC	カメラの同期信号による同期
SEPARATE_SYNC	CH2 VP#2同期
VP_SYNC	CH2 VP#1同期

注 1 ) EXTERNAL\_SYNC,CAMERA\_SYNCモードは、SVP-330では動作しません。

注 2 ) SEPARATE\_SYNC,VP\_SYNCモードは、SVP-330では動作しません。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

# SetVFDelay

## カメラ入力画面の遅延サイズ設定

### 機能

映像画面の遅延サイズを設定します。

### コーディング

```
int
SetVFDelay(
    DEVID    devID ,
    int      hdly ,
    int      vdly
)
```

### コメント

デバイスID  
遅延サイズ  
遅延サイズ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

hdly

遅延横サイズ (Hsync信号からの遅延サイズ)

vdly

遅延縦サイズ (Vsync信号からの遅延サイズ)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

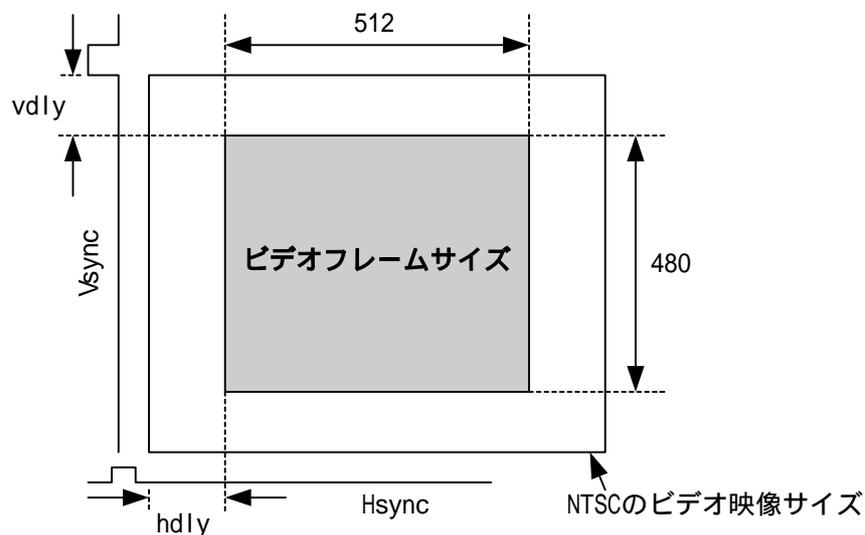
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	遅延横サイズ設定値範囲外
2 1	遅延縦サイズ設定値範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

本コマンドは、ビデオ信号のHsync信号（水平同期信号）及びVsync信号（垂直同期信号）を基準として、カメラ / 表示映像での有効映像領域を設定します。



- vdly : Vsync信号のはじめから設定された値のHsync信号後、カメラ映像  
が取り込まれます。  
映像出力のときは、Vsync信号の始めから設定された値のHsync  
信号後、映像情報がビデオ信号にのせられます。
- hdly : Hsync信号の始めから取り込みを行う画素までの数の2倍。  
映像出力のときは、Hsync信号の始めから映像データをのせる画素  
までの数の2倍。

# SetShutterSpeed

## カメラ映像入力シャッタスピードの設定

### 機能

フレームシャッタカメラのシャッタスピードを設定します。

SVP - 330では、本機能はサポートされていません。

### コーディング

```
int
SetShutterSpeed(
    DEVID          devID ,
    enum ShutterSpeed mode
)
```

### コメント

デバイスID  
シャッタスピード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

シャッタスピード

シャッタスピード	対応定数	説明
SHUT_4000	0	シャッタスピード 1/4000(0.25mS)
SHUT_2000	1	シャッタスピード 1/2000( 0.5mS)
SHUT_1000	2	シャッタスピード 1/1000( 1.0mS)
SHUT_500	3	シャッタスピード 1/500 ( 2.0mS)
SHUT_250	4	シャッタスピード 1/250 ( 4.0mS)
SHUT_125	5	シャッタスピード 1/125 ( 8.0mS)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

以下にコーディング例を示します。

```
SelectCamera(devID,CAMERA_PORT0,YUV_CAMERA);
SetShutterSpeed(devID,SHUT_1000);

GetCamera(devID,ImgID);
```

# Get2Camera

## 2カメラ映像の同時入力

### 機能

2台のカメラ映像を同時に指定された画面に入力します。SVP-330では、2カメラを完全に同期して入力する場合、同じカメラを2台用意し、SVP-330から出力される同期信号(VD/HD)で2つのカメラをハード的に同期させる必要があります。詳細は、「SVP-330ハードウェアマニュアル」2章の2.3(2)項を参照して下さい。なお、SVP-330では、同期信号で2つのカメラを同期させなくても2カメラ入力可能ですがその場合、最大で17ms程度の入力タイミングのずれが発生します。

### コーディング

```
int
Get2Camera(
    DEVID    devID,
    int      imgID1,
    int      imgID2
)
```

### コメント

デバイスID  
画面番号#1  
画面番号#2

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgID1**  
カメラポート#0のカメラ映像を入力する画面の画面番号

**imgID2**  
カメラポート#1のカメラ映像を入力する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10	画面空き領域、または範囲外(不当画面番号エラー)
42	カメラ入力での画面サイズミスマッチ
53	画面再配置エラー(画面ロック)
54	画面再配置エラー(空領域無し)
55	Get2Cameraでサポートしているカメラ以外のカメラ設定
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

SVP - 330での2カメラ同時入力は、2つのビデオプロセッサを同時に起動することにより行っています。そのため、Get2Camera()コマンドを使用する前にカメラ選択をSelectCameraMI tPort()を使用してビデオポート# 0とビデオポート# 1の両方に対して実行する必要があります。以下にその例を示します。

```
SelectCameraMI tPort(devID, VIDEO_PORT0, CAMERA_PORT0, YUV_CAMERA);  
SelectCameraMI tPort(devID, VIDEO_PORT1, CAMERA_PORT0, YUV_CAMERA);  
  
Get2Camera(devID, ImgID1, ImgID2);
```

また、ビデオフレーム設定やトリガー設定も同様にビデオポート# 0とビデオポート# 1の両方に対して行って下さい。

## SetTrigerMode

### カメラ映像入力トリガモードの設定

#### 機能

フレームシャッターカメラのトリガモードを設定します。トリガモードで映像を入力する場合は、本コマンドで TRIGGERモードに設定してから GetCamera( )コマンドで画像入力を行います。SelectCamera( )コマンドを発行するとTRIGGERモードが解除されますので、再設定して下さい。

**SVP - 330では、本機能はサポートされていません。**

#### コーディング

```
int
SetTrigerMode(
    DEVID          devID ,
    enum TrigerMode mode
)
```

#### コメント

デバイスID  
トリガモード

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

トリガモード

トリガモード	対応定数	説明
NOTRIGGER	0	トリガ無し（スルー映像出力）
RR_TRIGGER	1	V Iトリガモード （RR_TRIGGERとTRIGGERは同じ）
TRIGGER	2	
EXT_TRIGGER	3	外部トリガ（使用できません）

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

以下にコーディング例を示します。

```
SelectCamera(devID, CAMERA_PORT0, SONY_XC55);
SetTrigerMode(devID, TRIGGER);
SetShutterSpeed(devID, SHUT_1000);

GetCamera(devID, ImgID);
```

# GetCameraSts

## カメラ接続のチェック

### 機能

カメラが接続されているかどうかのチェックを行います。本コマンドは、SelectCamera()コマンド等でカメラが選択された後に有効になります。また、本コマンドでは同期信号の有無でカメラ接続のチェックを行っておりますので、トリガモード時に同期信号が停止するカメラではカメラ接続のチェックはできませんので注意して下さい。

### コーディング

```
int
GetCameraSts(
    DEVID devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

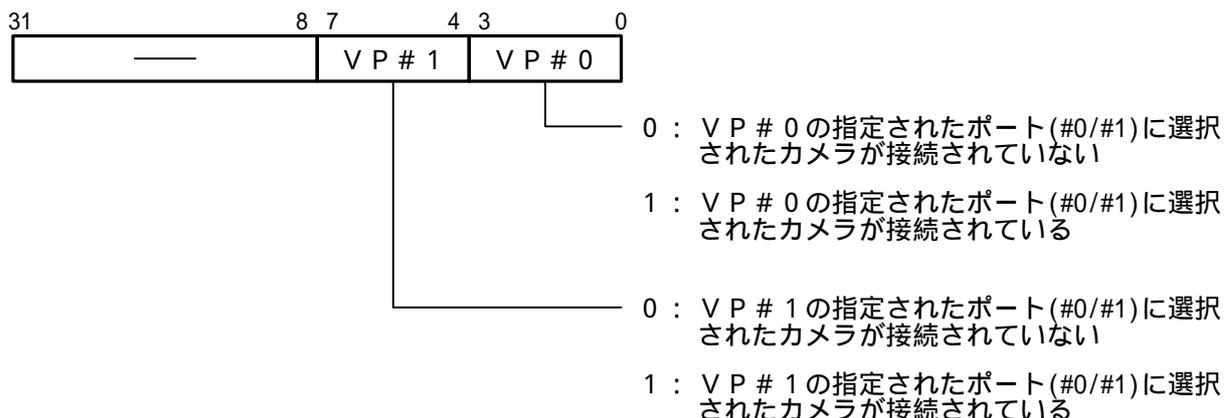
リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	カメラが接続されていない。
1	カメラが接続されている。以下の詳細情報を参照して下さい。
- 1	異常終了

異常終了時の詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

カメラが接続されている場合のリターン値を以下に示します。



カメラポートに対するVPの対応を以下に示します。

カメラポート	モノクロカメラ	CVBSカラーカメラ	備考
CAMERA_PORT0	VP # 0	VP # 0	
CAMERA_PORT1	VP # 1	VP # 1	

## Get4Camera

## 4カメラ映像の同時入力

### 機能

4台のカメラ映像を同時に指定された画面に入力します。

SVP-330では、本機能はサポートされていません。

### コーディング

```
int
Get4Camera(
    devID    devID ,
    int      imgID1 ,
    int      imgID2 ,
    int      imgID3 ,
    int      imgID4
)
```

### コメント

デバイスID  
画面番号#1  
画面番号#2  
画面番号#3  
画面番号#4

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgID1

カメラポート#0のカメラ映像を入力する画面の画面番号

imgID2

カメラポート#1のカメラ映像を入力する画面の画面番号

imgID3

カメラポート#2のカメラ映像を入力する画面の画面番号

imgID4

カメラポート#3のカメラ映像を入力する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10	画面空き領域、または範囲外（不当画面番号エラー）
42	カメラ入力での画面サイズミスマッチ
53	画面再配置エラー（画面ロック）
54	画面再配置エラー（空領域無し）
55	Get4Cameraでサポートしているカメラ以外のカメラ設定
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

4カメラ映像の同時入力を行う場合は、同じカメラを4台用意して下さい。同一のカメラ以外の組み合わせでは、動作の保証はありません。NTSC標準カメラの場合、外部同期信号を入力できるタイプのカメラ以外は使用できません。対応カメラについては、弊社までお問い合わせ下さい。

# ResetCamera

## カメラのリセット

### 機能

トリガカメラのリセット処理を行い、画像処理ボードでトリガカメラを使用できるようにします。現在、本コマンドが必要なケースはトリガモードで立ち上がるカメラリンクのカメラを使用する場合です。

### コーディング

```
int
ResetCamera(
    DEVID
    enum CameraID
    enum CameraType
    int
)
```

```
devID ,
id ,
type ,
opt
```

### コメント

```
デバイスID
カメラポート番号
カメラタイプ
オプション
```

### パラメータ

- devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。
- id**  
カメラポート番号
- type**  
入力カメラタイプ
- opt**  
オプション。現在未使用「0」を指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ResetCamera( )コマンドは、画像処理ボードの電源投入後、InitIP( )の後に行ってください。また、ビデオ系の設定コマンド(SelectCamera(),SetTrigerMode()等)は、ResetCamera( )コマンド発行後行って下さい。

```
InitIP( devID );
ResetCamera( devID, CAMERA_PORT0, CIS_VCC8350CL, 0);
.
.
.
SelectCamera( devID, CAMERA_PORT0, CIS_VCC8350CL);
SetTrigerMode( devID, TRIGGER );
```

# DispCamera

## カメラ映像表示

### 機能

カメラから入力した映像を表示します。

### コーディング

```
int  
DispCamera(  
    DEVID    devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# Displmg

## 画像メモリ表示

### 機能

パラメータで指定された画像を表示します。

### コーディング

```
int
Displmg(
    DEVID    devID ,
    int      imgID
)
```

### コメント

デバイスID  
画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgID

表示する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
4 3	表示画面サイズと、ビデオフレームサイズミスマッチ
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドで、画面を表示する際に、画像メモリの画面サイズと、映像表示サイズを一致させておく必要があります。なお、不一致の場合はエラーとなり、本処理は行われません。

対応画面サイズ一覧

画面サイズ	画像メモリサイズ		表示フレームサイズ	映像入力サイズ	
	X方向	Y方向		H	V
IMG_FS_256H_256V	256	256	DISP_FS_256H_220V	256	220
			DISP_FS_256H_240V	256	240
IMG_FS_512H_256V	512	256	DISP_FS_512H_220V	512	220
			DISP_FS_512H_240V	512	240
			DISP_FS_320H_240V	320	240
IMG_FS_256H_512V	256	512	DISP_FS_256H_440V	256	440
			DISP_FS_256H_480V	256	480
IMG_FS_512H_512V	512	512	DISP_FS_512H_440V	512	440
			DISP_FS_512H_480V	512	480
IMG_FS_640H_256V	640	256	DISP_FS_640H_240V	640	240
IMG_FS_640H_512V	640	512	DISP_FS_640H_480V	640	480
IMG_FS_1024H_256V	1024	256	DISP_FS_1024H_240V	1024	240
IMG_FS_1024H_512V	1024	512	DISP_FS_1024H_480V	1024	480
IMG_FS_1024H_1024V	1024	1024	DISP_FS_1024H_1024V	1024	1024
IMG_FS_1280H_256V	1280	256	DISP_FS_1280H_240V	1280	240
IMG_FS_1280H_512V	1280	512	DISP_FS_1280H_480V	1280	480
IMG_FS_1280H_1024V	1280	1024	DISP_FS_1280H_1024V	1280	1024
IMG_FS_1024H_768V	1024	768	DISP_FS_1024H_768V	1024	768

# NoDisp

表示終了

## 機能

ループ入力とループ表示を中止して表示を中止します。

## コーディング

```
int  
NoDisp(  
    DEVID    devID  
)
```

## コメント

デバイスID

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

NoDisp()コマンドで表示中止状態に設定すると、表示処理と画像処理の競合している場合、オーバーヘッドが削減され画像処理が高速になります。処理の高速化を求める場合は、表示処理と画像処理の競合をさせないようにするか、NoDisp()コマンドを実行して表示中止状態にすることを推奨します。

# BitmapOverlap

## オーバーレイ表示制御

### 機能

ビットマップ（オーバーラップ）画面のオーバーラップ表示の有効／無効の切り換えを行います。AllocDispImg()をディスプレイモード「BMP\_DISPLAY」で実行すると、自動的にオーバーレイ画面が確保され、本コマンドでそのオーバーラップ表示の有効／無効を制御します。SelectCamera()コマンドでカメラタイプが変更されると、自動的にオーバーラップ表示が解除されますので、オーバーラップ表示を続ける場合は再度本コマンドでオーバーラップ表示を有効にして下さい。また、カメラタイプがYUVカラーカメラが選択された場合や表示画面モードがYUVカラー画面に設定されている場合は、オーバーラップ表示はできません。その場合も自動的にオーバーラップ表示が解除されます。オーバーラップ表示の詳細は「5.10 映像入出力の基礎」を参照して下さい。

InitIP()コマンドを発行すると、AllocDispImg()が「BMP\_DISPLAY」モードで実行され、オーバーレイ画面が有効に設定された状態になります。

### コーディング

```
int
BitmapOverlap(
    DEVID      devID ,
    int        mode
)
```

### コメント

デバイスID  
ビットマップオーバーラップモード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

ビットマップオーバーラップモード

ビットマップオーバーラップモード	説明
BITMAP_OFF	ビットマップオーバーラップ表示しない
BITMAP_ON	ビットマップオーバーラップ表示する

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## DispOverlap

## 画像メモリのオーバーレイ表示

### 機能

指定された画面とオーバーラップ表示を行います。InitIP( )コマンドを発行すると、AllocDispImg()が「BMP\_DISPLAY」モードで実行され、オーバーレイ画面が自動的に確保され画像処理コマンド内部で管理される状態になります。本コマンドを発行すると、オーバーレイ画面が画像処理コマンド内部で自動的に管理される動作が解除されます。従ってオーバーラップ画面は、ユーザアプリケーションで管理する必要があります。また、オーバーラップ表示の有効/無効は、BitmapOverlap()コマンドで制御します。オーバーラップ表示の詳細は「5.10 映像入出力の基礎」を参照して下さい。

### コーディング

```
int
DispOverlap(
    DEVID          devID ,
    int            ImgID ,
    enum OverlapMode mode
)
```

### コメント

デバイスID  
画面番号  
重ね合わせ表示属性

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
オーバーレイ（重ね合わせ）表示をおこなう画面の画面番号

**mode**  
重ね合わせ表示属性。OVERLAP\_MAX、OVERLAP\_ADD は同じ表示属性になります。

重ね合わせ表示属性	対応定数	説明
OVERLAP_MAX	0	指定したオーバーレイ画面の画素値が「0」の部分は映像表示画面の画素値が表示され、「0」以外の部分はオーバーレイ画面の画素値が出力される。
OVERLAP_ADD	4	

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	不当画面番号エラー
2 1	重ね合わせ表示属性設定値エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetDFDelay

## 映像表示画面の遅延サイズ設定

### 機能

映像表示画面の遅延サイズを設定します。

### コーディング

```
int
SetDFDelay(
    DEVID    devID ,
    int      hdly ,
    int      vdly
)
```

### コメント

デバイスID  
遅延サイズ  
遅延サイズ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

hdly

遅延横サイズ (Hsync信号からの遅延サイズ)

vdly

遅延縦サイズ (Vsync信号からの遅延サイズ)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	遅延横サイズ設定値範囲外
2 1	遅延縦サイズ設定値範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetDispFrame

## 映像表示画像設定

### 機能

映像表示での走査方式、表示サイズ（ビデオフレームサイズ）を、パラメータに従い設定します。通常は SelectCamera コマンドでカメラタイプによって最適な値に設定されますので、特別に設定が必要な場合以外は再設定しないで下さい。

### コーディング

```
int
SetDispFrame(
    DEVID
    enum Interlace
    enum DispFrameSize
)
```

```
devID ,
mode ,
size
```

```
デバイスID
ビデオ画面規定（インタレース）
映像表示画面サイズ
```

### コメント

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

ビデオ画面規定（インタレース）

走査方式	対応定数	内容
NONINTERLACE	0	ノンインタレース
INTERLACE	1	インタレース

size

映像表示画面サイズ

映像表示画面サイズ	対応定数	映像入力サイズ		表示サイズ	
		H	V	H	V
DISP_FS_256H_220V	0	256	220	256	220
DISP_FS_512H_220V	1	512	220	512	220
DISP_FS_256H_440V	2	256	440	256	440
DISP_FS_512H_440V	3	512	440	512	440
DISP_FS_256H_240V	4	256	240	256	240
DISP_FS_320H_240V	5	320	240	320	240
DISP_FS_512H_240V	6	512	240	512	240
DISP_FS_640H_240V	7	640	240	640	240
DISP_FS_256H_480V	8	256	480	256	480
DISP_FS_320H_480V	9	320	480	320	480
DISP_FS_512H_480V	10	512	480	512	480
DISP_FS_640H_480V	11	640	480	640	480
DISP_FS_512H_512V	12	512	512	512	480
DISP_FS_1008H_470V	13	1008	470	512	470
DISP_FS_1024H_1024V	14	1024	1024	512	480
DISP_FS_1024H_240V	15	1024	240	512	240
DISP_FS_1024H_480V	16	1024	480	512	480
DISP_FS_1280H_1024V	17	1280	1024	640	480
DISP_FS_1280H_240V	18	1280	240	640	240
DISP_FS_1280H_480V	19	1280	480	640	480
DISP_FS_1024H_768V	20	1024	768	512	480
DISP_FS_1920H_1024V	21	1920	1024	640	480
DISP_FS_2048H_1024V	22	2048	1024	640	480
DISP_FS_1280H_256V	23	1280	256	640	256

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20, 21	パラメータエラー
-	エラーリセットコマンド (ClearLError) 未発行

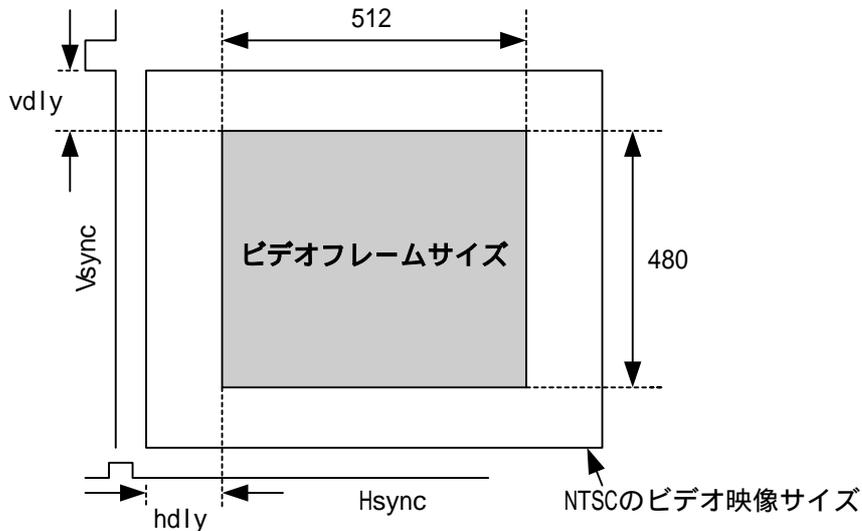
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

映像入力、あるいは表示の対象となる画像メモリの画面サイズは、ビデオフレームサイズと一致している必要があります。

NTSC準拠のカメラ（インタレース）を接続してノンインタレースを指定すると、

- ・カメラ映像入力では、カメラ映像の第1フィールドのみ取り込まれ、縦方向に1/2圧縮した映像となります。
- ・画像メモリの映像表示では、逆に縦方向に2倍に拡大されて表示されます。  
(第1フィールドの映像を第2フィールドにも出力するため)
- ・画面の四角サイズに256Hを指定すると
  - 1) カメラ映像入力では、カメラ映像が縦方向に1/2に圧縮されて取り込まれます。
  - 2) 画像メモリの映像表示では、逆に横方向に2倍にされて表示されます。



# SelectDisp

## 映像出力形式の選択

### 機能

映像出力形式の選択を行います。SelectCamera( )コマンドでモノクロカメラを選択した場合には、BMP\_DISPLAY、YUVカラーカメラやRGBカメラを選択した場合には、YUV\_DISPLAYが自動的に選択されます。SelectCamera( )でカラーカメラを選択せずに、YUVカラー映像を表示する場合や、SelectCamera( )でYUVカメラを指定した場合のモノクロ映像表示に使用します。なお、SelectCamera( )コマンドを実行すると、本コマンドで設定した値が変更されますので注意して下さい。

### コーディング

```
int
SelectDisp(
    DEVID          devID ,
    enum DispType  type
)
```

### コメント

デバイスID  
映像出力形式

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

type

映像出力形式

映像出力形式	対応定数	内容
BW_DISPLAY	0	モノクロ出力
YUV_DISPLAY	1	YUVカラー映像出力
BMP_DISPLAY	2	オーバーレイ表示付きモノクロ映像出力

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# SetDispWindow

## 高精細映像表示のウィンドウ設定

### 機能

高精細映像表示 ( 1 0 2 4 × 1 0 2 4 ) のウィンドウを設定します。

本コマンドで設定した値は、EnableHIREZDisp()コマンド発行後に有効になり、DisableHIREZDisp()コマンド発行後に無効になります。EnableHIREZDisp()コマンド発行により有効になったウィンドウは、その次に発行されるDispCamera(),DispImg()コマンドに反映されます。また、DispCamera()コマンドが有効でかつ高精細映像表示が有効の場合、本コマンドの発行で表示ウィンドウが変更されます。

なお、オーバーラップ表示が有効な場合、X方向間引き間隔(xmag)が「1」以外での表示はできませんので注意して下さい。

### コーディング

```
int
SetDispWindow(
    DEVID    devid ,
    int      sx ,
    int      sy ,
    int      xmag ,
    int      ymag
)
```

### コメント

デバイスID  
映像表示開始X座標  
映像表示開始Y座標  
X方向間引き間隔  
Y方向間引き間隔

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### sx

映像表示開始X座標

#### sy

映像表示開始Y座標

#### xmag

X方向間引き間隔 ( 1 ~ 2 ) 。オーバーラップ表示が有効な場合、「1」以外での表示はできません。

1 : 間引き無し  
2 : 1 / 2 表示

#### ymag

Y方向間引き間隔 ( 1 ~ 2 )

1 : 間引き無し  
2 : 1 / 2 表示

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20 ~ 23	パラメータエラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## EnableHIREZDisp

## 高精細映像表示の有効化

### 機能

SetDispWindow()コマンドで設定した高精細映像表示(1024×1024)のウィンドウを有効にします。本コマンド発行により有効になったウィンドウは、その次に発行されるDispCamera()、DispImg()コマンドに反映されます。

### コーディング

```
int  
EnableHIREZDisp(  
    DEVID    devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## DisableHIREZDisp

### 高精細映像表示の無効化

#### 機能

SetDispWindow()コマンドで設定した高精細映像表示(1024×1024)のウィンドウを無効にします。本コマンド発行により無効になったウィンドウは、その次に発行されるDispCamera()、DispImg()コマンドに反映されます。

#### コーディング

```
int  
DisableHIREZDisp(  
    DEVID    devID  
)
```

#### コメント

デバイスID

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## IP\_ClearAllImg

### 画像メモリクリア(全チャンネル)

#### 機能

全チャンネルの画像メモリを0クリアします。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int  
IP_ClearAllImg(  
    DEVID    devID  
)
```

#### コメント

デバイスID

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

# IP\_ClearCHImg

## 画像メモリクリア(指定チャンネル)

### 機能

パラメータで指定されたチャンネルの画像メモリを0クリアします。  
本コマンドはVP - 810専用のコマンドです。VP - 810以外では、チャンネルNo、ページNoは無効となり、使用できません。

### コーディング

```
int
IP_ClearCHImg(
    DEVID      devID ,
    enum CHID  ch
)
```

### コメント

デバイスID  
画像メモリチャンネル指定

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ch

クリア対象チャンネルNo.

チャンネルNo.	対応定数	内容
IMG_CH0	0	画像メモリチャンネル#0
IMG_CH1	1	画像メモリチャンネル#1
IMG_CH2	2	画像メモリチャンネル#2
IMG_CH3	3	画像メモリチャンネル#3
IMG_CH4	4	画像メモリチャンネル#4

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	チャンネルNo.範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_ClearImg

## 画像メモリクリア(指定画面)

### 機能

パラメータで指定された画面の画像メモリを0クリアします。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_ClearImg(
    DEVID  devID ,
    int    ImgID
)
```

### コメント

デバイスID  
画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID

処理画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_Const

## 定数発生

### 機能

パラメータで指定された画面に、定数を書き込みます。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Const(
    DEVID    devID ,
    int      ImgID ,
    int      constant
)
```

### コメント

デバイスID  
画面番号  
定数

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**

処理画面の画面番号

**constant**

定数（下位8ビットのみ有効）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 3	指定画面空き領域、または範囲外（不当画面番号エラー）
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_Copy

## 画像転送

### 機能

指定画面間で画像のコピーを行います。

なお、本コマンドを実行したときの範囲は、ソース画面、デスティネーション画面サイズ、SRCO\_WIN, DST\_WINで指定された範囲に従います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Copy(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
コピー元の画面番号  
コピー先の画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

コピー元の画面番号

ImgDst

コピー先の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## IP\_Zoom

## ズーム

## 機能

パラメータで指定した変換倍率に従い、拡大します。本機能はソフトウェアにより処理しています。  
 なお、本コマンドを実行したときの出力画面サイズは、デスティネーション画面サイズ、またはデスティネーション・ウィンドウサイズ (DST\_WIN) となります。

## コーディング

```
int
IP_Zoom(
    DEVID      devID ,
    int        ImgSrc ,
    int        ImgDst ,
    float      mag
)
```

## コメント

デバイスID  
 ソース画面番号  
 デスティネーション画面番号  
 変換倍率

## パラメータ

**devID**  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
 ソース画面番号

**ImgDst**  
 デスティネーション画面番号

**mag**  
 X、Y方向への変換倍率 ( 0 . 0 4 ~ 1 2 8 )

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2	変換倍率範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ソース画面の画像をパラメータで指定した変換倍率に従い拡大し、その結果をデスティネーション画面へ転送します。

拡大の画像サイズがDST\_WINサイズより大きい場合DST\_WINでクリッピングされます。

拡大の画像サイズがDST\_WINサイズより小さい場合転送対象外の部分は変化しません。

[実行例]

IP\_Zoom ( devID , ImgSrc , ImgDst , 2.0 )



拡大後のサイズがDST\_WINサイズより小さい場合、転送対象以外の部分は変化されません。

[実行例]

IP\_Zoom ( devID , ImgSrc , ImgDst , 0.5 )

変化しない



#### [注] YUV画面を処理する場合

カラー画面では、UVデータはUV画面にUとVの順番で1画素おきに格納されています。

よって、ウィンドウの始点座標を奇数や、ウィンドウの横方向サイズを奇数にして画像処理を実行した場合に、UとVの順番が入れ替わり、カラーの色情報がくずれます。ウィンドウの始点座標は偶数に、ウィンドウの横方向サイズも偶数にしてください。

また、YUV画面での縮小倍率は2の乗数の逆数となります。

(詳細はユーザーズマニュアルを参照してください)

## IP\_ZoomExt

## ズーム(縦横任意倍率)

### 機能

パラメータで指定した変換倍率に従い、拡大します。本機能はソフトウェアにより処理しています。  
 なお、本コマンドを実行したときの出力画面サイズは、デスティネーション画面サイズ、またはデスティネーション・ウィンドウサイズ(DST\_WIN)となります。

### コーディング

```
int
IP_ZoomExt(
    DEVID    devID ,
    int      imgSrc ,
    int      imgDst ,
    float    xmag ,
    float    ymag
)
```

### コメント

デバイスID  
 ソース画面番号  
 デスティネーション画面番号  
 X方向への変換倍率  
 Y方向への変換倍率

### パラメータ

**devID**  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgSrc**  
 ソース画面番号

**imgDst**  
 デスティネーション画面番号

**xmag**  
 X方向への変換倍率(0.04 ~ 128)

**ymag**  
 Y方向への変換倍率(0.04 ~ 128)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	X方向の変換倍率範囲外
23	Y方向の変換倍率範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

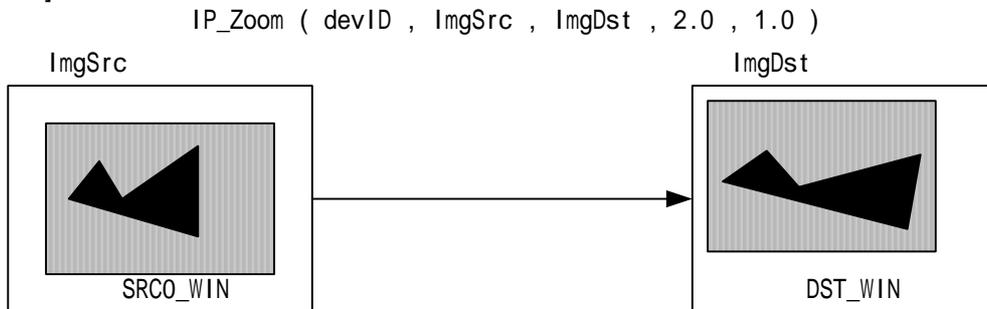
### 詳細情報

ソース画面の画像をパラメータで指定した変換倍率に従い拡大し、その結果をデスティネーション画面へ転送します。

拡大の画像サイズがDST\_WINサイズより大きい場合DST\_WINでクリッピングされます。

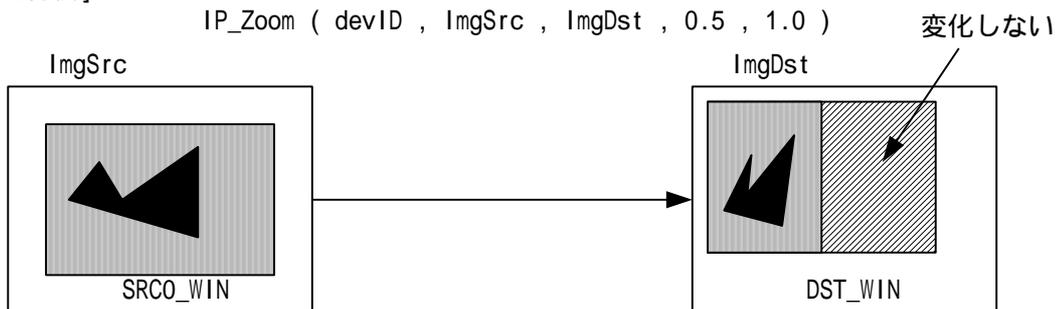
拡大の画像サイズがDST\_WINサイズより小さい場合転送対象外の部分は変化しません。

[実行例]



拡大後のサイズがDST\_WINサイズより小さい場合、転送対象以外の部分は変化されません。

[実行例]



#### [注] YUV画面を処理する場合

カラー画面では、UVデータはUV画面にUとVの順番で1画素おきに格納されています。

よって、ウィンドウの始点座標を奇数や、ウィンドウの横方向サイズを奇数にして画像処理を実行した場合に、UとVの順番が入れ替わり、カラーの色情報がくずれます。ウィンドウの始点座標は偶数に、ウィンドウの横方向サイズも偶数にしてください。

また、YUV画面での縮小倍率は2の乗数の逆数となります。

(詳細はユーザーズマニュアルを参照してください)

## IP\_ZoomOut

## ズームアウト

## 機能

パラメータで指定した変換倍率に従い、縮小します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_ZoomOut(
    DEVID      devID ,
    int        ImgSrc ,
    int        ImgDst ,
    int        mag
)
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
変換倍率

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面番号

ImgDst

デスティネーション画面番号

mag

X、Y方向への変換倍率（1～8）

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	変換倍率範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ソース画面の画像をパラメータで指定した変換倍率に従い縮小し、その結果をデスティネーション画面へ転送します。

縮小後の画像サイズがDST\_WINサイズより大きい場合DST\_WINでクリッピングされます。

縮小後の画像サイズがDST\_WINサイズより小さい場合転送対象外の部分は変化しません。

## IP\_ZoomOutExt

## ズームアウト(縦横任意倍率)

### 機能

パラメータで指定した変換倍率に従い、縮小します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_ZoomOutExt(
    DEVID      devID ,
    int        ImgSrc ,
    int        ImgDst ,
    int        xmag ,
    int        ymag
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
X方向への変換倍率  
Y方向への変換倍率

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面番号

**ImgDst**

デスティネーション画面番号

**xmag**

X方向への変換倍率(1~8)

**ymag**

Y方向への変換倍率(1~8)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	X方向の変換倍率範囲外
23	Y方向の変換倍率範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ソース画面の画像をパラメータで指定した変換倍率に従い縮小し、その結果をデスティネーション画面へ転送します。

縮小後の画像サイズがDST\_WINサイズより大きい場合DST\_WINでクリッピングされます。

縮小後の画像サイズがDST\_WINサイズより小さい場合転送対象外の部分は変化しません。

# IP\_Shift

## シフト

### 機能

ソース画像を移動（シフト）し、デスティネーションウィンドウの左上（ $s_x, s_y$ ）から $X, Y$ 方向に移動した結果をデスティネーション画面に転送します。

IP\_Copy()コマンドとの違いは、オプションで移動先の周辺部の処理を指定できることです。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Shift(
    DEVID          devID ,
    int            ImgSrc ,
    int            ImgDst ,
    int            dx ,
    int            dy ,
    enum SPACE_OPT opt
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
X方向移動量  
Y方向移動量  
周辺部クリアオプション

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面番号

**ImgDst**  
デスティネーション画面番号

**dx**  
X方向移動量

**dy**  
Y方向移動量

**opt**  
周辺部のクリアオプション

クリアオプション	対応数	内容
SPACE_OCLEAR	0	周辺を0クリアする
SPACE_NOTOUCH	1	周辺は変化させない
SPACE_128CLEAR	2	周辺を128クリアする

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

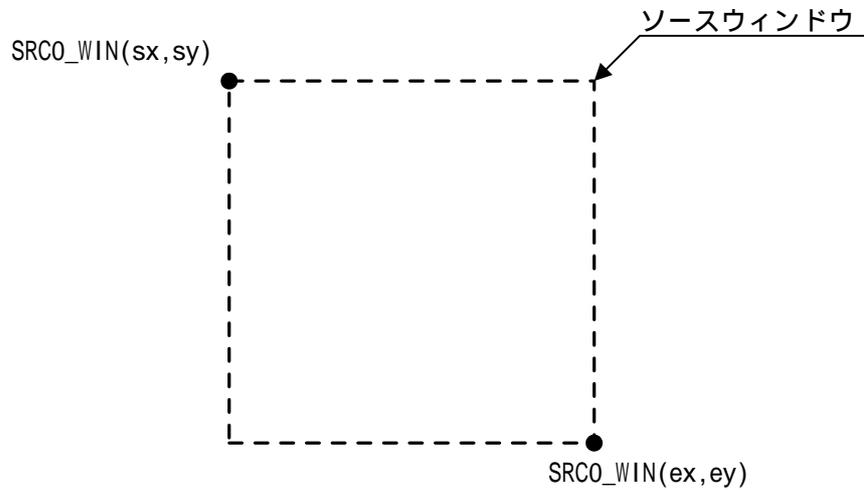
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
22 ~ 27	パラメータエラー
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

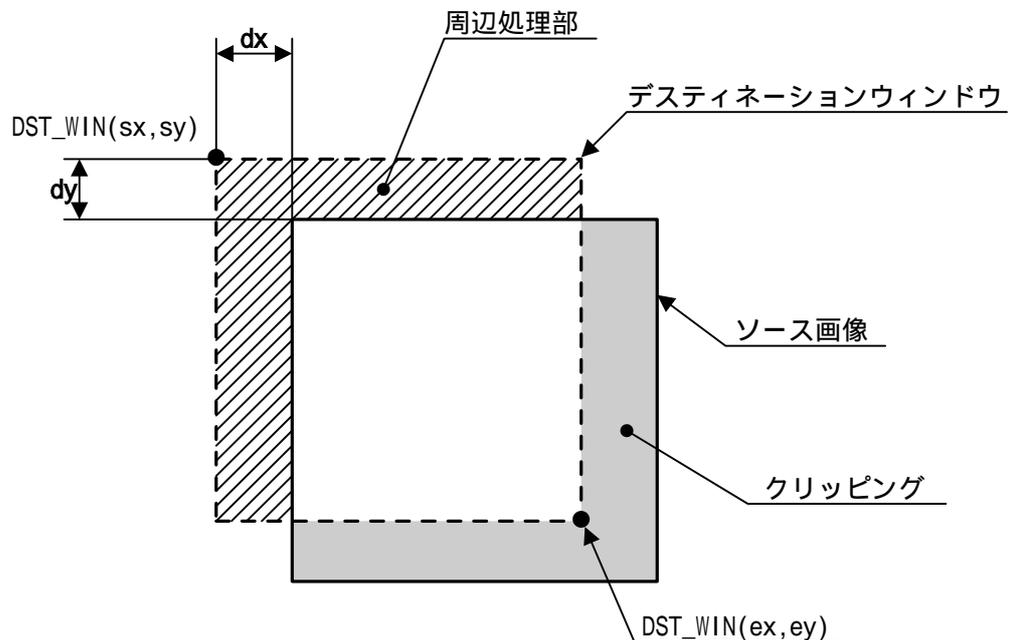
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

SRCO\_WINで指定された範囲を $(sx, sy)$ から $(ex, ey)$ が移動対象領域です。



そして、DST\_WINで指定された $(sx, sy)$ から $(dx, dy)$ 画素だけ移動した位置にソース画像を転送します。また、移動した画像のDST\_WINの範囲を超えた部分はクリッピングされ、周辺部はオプション指定に従い、処理されます。



# IP\_ZoomS

## シフト付きズーム

### 機能

ソース画像を移動（シフト）しズームし、デスティネーションウィンドウの左上（ $s_x, s_y$ ）から $X, Y$ 方向に移動した結果をデスティネーション画面に転送します。本機能はソフトウェアにより処理しています。IP\_Zoom()コマンドとの違いは、オプションで移動先の周辺部の処理を指定できることです。

### コーディング

```
int
IP_ZoomS(
    DEVID
    int
    int
    float
    int
    int
    enum SPACE_OPT
)
```

```
devID ,
ImgSrc ,
ImgDst ,
mag ,
dx ,
dy ,
opt
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
変換倍率
X方向移動量
Y方向移動量
周辺部クリアオプション
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面番号

**ImgDst**

デスティネーション画面番号

**mag**

変換倍率（0.04～128）

**dx**

X方向移動量

**dy**

Y方向移動量

**opt**

周辺部クリアオプション

クリアオプション	対応数	内容
SPACE_OCLEAR	0	周辺を0クリアする
SPACE_NOTOUCH	1	周辺は変化させない
SPACE_128CLEAR	2	周辺を128クリアする

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

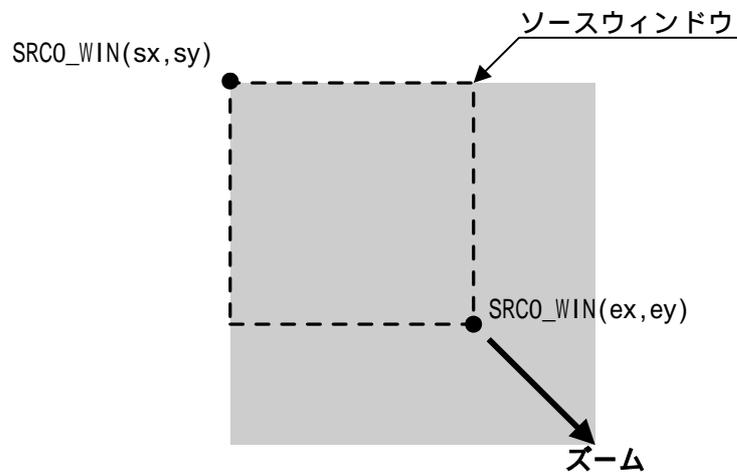
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
22 ~ 25	パラメータエラー
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

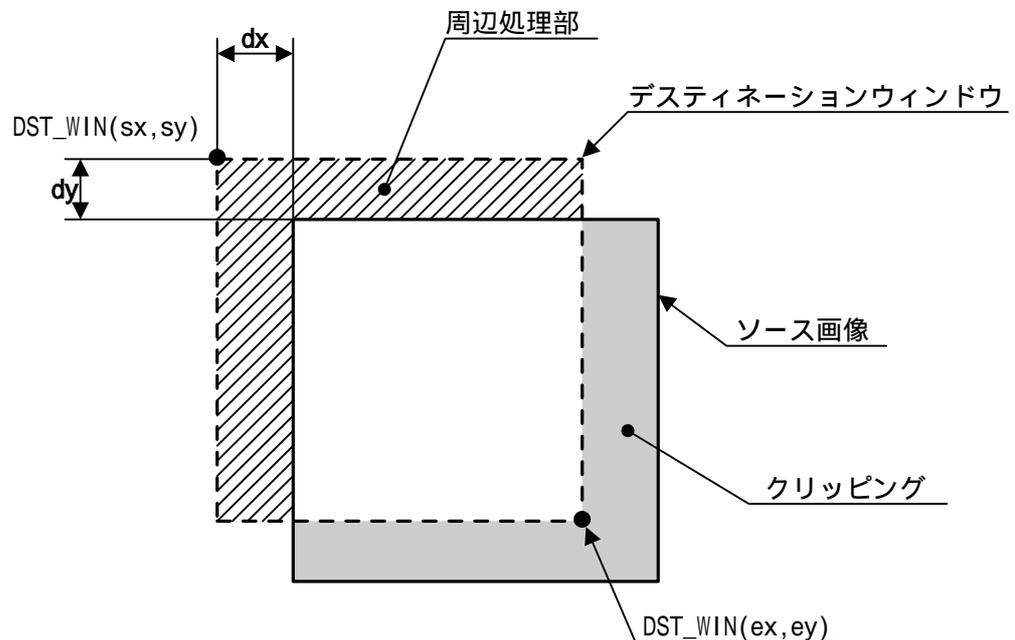
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

SRCO\_WINで指定された範囲を拡大（ズーム）します。



そして、DST\_WINで指定された  $(sx, sy)$  から  $(dx, dy)$  画素だけ移動した位置にズーム画像を転送します。また、移動した画像のDST\_WINの範囲を超えた部分はクリッピングされ、周辺部はオプション指定に従い、処理されます。



# IP\_Rotate

## 回転

### 機能

ソース画面の画像を回転中心座標に対して時計回りに回転させ、デスティネーションウィンドウの左上 (  $s_x, s_y$  ) から  $X, Y$  方向に移動した結果をデスティネーション画面に転送します。本機能はソフトウェアにより処理しています。

### コーディング

```
int
IP_Rotate(
    DEVID
    int
    int
    float
    float
    int
    int
    int
    int
    int
    enum SPACE_OPT
)
```

```
devID ,
ImgSrc ,
ImgDst ,
mag ,
theta ,
xc ,
yc ,
dx ,
dy ,
opt
```

### コメント

デバイス I D  
ソース画面番号  
デスティネーション画面番号  
 $X, Y$  方向への変換倍率 ( 0 . 0 4 ~ 1 2 8 )  
回転角度  
回転中心  $X$  座標  
回転中心  $Y$  座標  
 $X$  方向移動量  
 $Y$  方向移動量  
周辺部クリアオプション

### パラメータ

**devID**

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**ImgSrc**

ソース画面番号

**ImgDst**

デスティネーション画面番号

**mag**

$X, Y$  方向への変換倍率 ( 0 . 0 4 ~ 1 2 8 )

**theta**

回転角度

**xc**

回転中心  $X$  座標 ( SRCO\_WIN 相対 )

**yc**

回転中心  $Y$  座標 ( SRCO\_WIN 相対 )

**dx**

$X$  方向移動量 ( DST\_WIN 相対 )

**dy**

$Y$  方向移動量 ( DST\_WIN 相対 )

**opt**

周辺部クリアオプション。

クリアオプション	対応数	内容	備考
SPACE_OCLEAR	0	周辺を 0 クリアする	
SPACE_NOTOUCH	1	周辺は変化させない	
SPACE_128CLEAR	2	周辺を 1 2 8 クリアする	

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

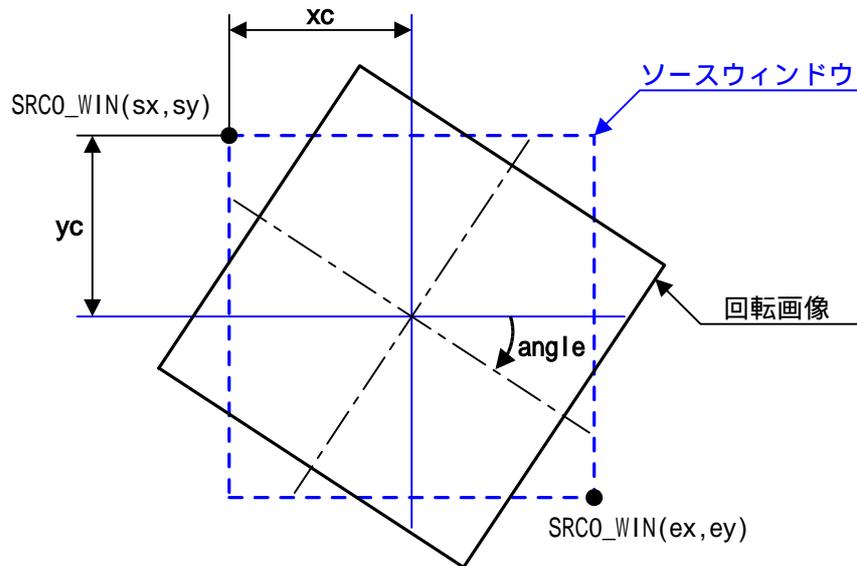
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外 (不当画面番号エラー)
22 ~ 27	パラメータエラー
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

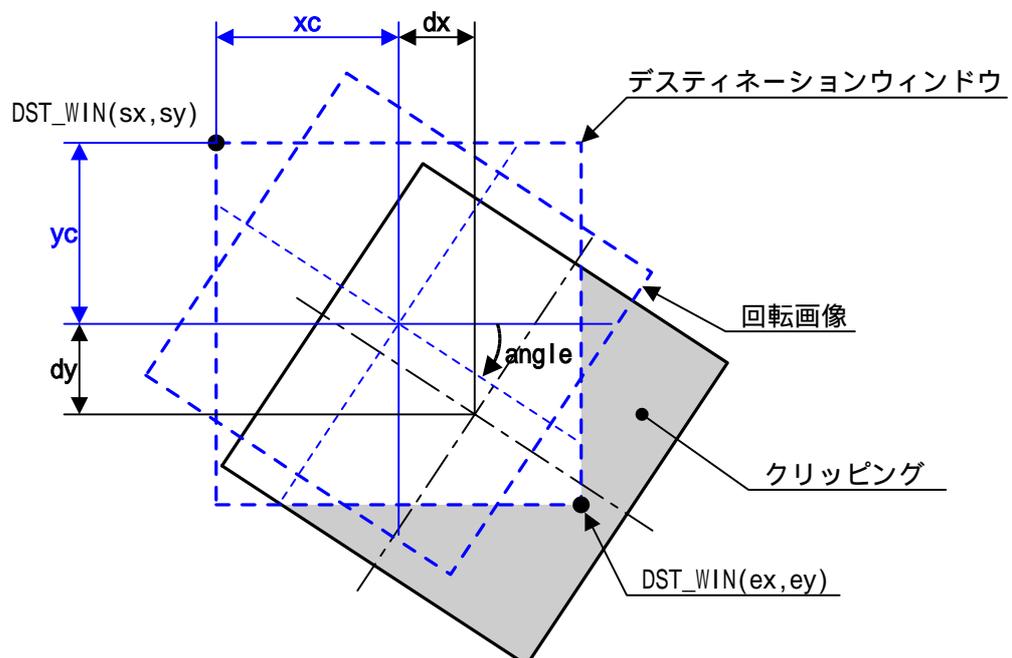
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

SRCO\_WINで指定された範囲を $(sx, sy)$ からの相対位置 $(xc, yc)$ を回転中心として $angle$ 角度回転させます。



そして、DST\_WINで指定された $(sx, sy)$ から $(dx, dy)$ 画素だけ移動した位置に回転したソース画像を転送します。また、回転画像のDST\_WINの範囲を超えた部分はクリッピングされます。



## IP\_ZoomIn

## ズームイン

## 機能

パラメータで指定した変換倍率に従い、拡大します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_ZoomIn(
    DEVID      devID ,
    int        ImgSrc ,
    int        ImgDst ,
    int        mag
)
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
変換倍率

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面番号

## ImgDst

デスティネーション画面番号

## mag

X、Y方向への変換倍率（1～8）

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	変換倍率範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ソース画面の画像をパラメータで指定した変換倍率に従い拡大し、その結果をデスティネーション画面へ転送します。

拡大の画像サイズがDST\_WINサイズより大きい場合DST\_WINでクリッピングされます。

拡大の画像サイズがDST\_WINサイズより小さい場合転送対象外の部分は変化しません。

# IP\_ZoomInExt

## ズームイン(縦横任意倍率)

### 機能

パラメータで指定した変換倍率に従い、拡大します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_ZoomInExt(
    DEVID      devID ,
    int        ImgSrc ,
    int        ImgDst ,
    int        xmag ,
    int        ymag
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
X方向への変換倍率  
Y方向への変換倍率

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面番号

**ImgDst**  
デスティネーション画面番号

**xmag**  
X方向への変換倍率(1~8)

**ymag**  
Y方向への変換倍率(1~8)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	X方向の変換倍率範囲外
23	Y方向の変換倍率範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ソース画面の画像をパラメータで指定した変換倍率に従い拡大し、その結果をデスティネーション画面へ転送します。

拡大の画像サイズがDST\_WINサイズより大きい場合DST\_WINでクリッピングされます。

拡大の画像サイズがDST\_WINサイズより小さい場合転送対象外の部分は変化しません。

# IP\_Binarize

## 2値化

### 機能

ソース画面の画像を、パラメータ指定のしきい値で2値化し、その結果をデスティネーション画面へ転送します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Binarize(
    DEVID  devID ,
    int    imgsrc ,
    int    imgdst ,
    int    thr
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
しきい値

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgSrc

ソース画面の画面番号

imgDst

デスティネーション画面の画面番号

thr

2値化するしきい値（下位8ビットのみ有効）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
22	しきい値設定範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

パラメータthrで指定するしきい値以上の値を白、それより小さい値を黒で2値化します。



# IP\_BinarizeExt

範囲・反転付き2値化

## 機能

ソース画面の画像を、パラメータ指定のしきい値（thrmin, thrmaxにより範囲指定可能）で2値化し、その結果をデスティネーション画面へ転送します。オプションとして反転指定が可能です。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_BinarizeExt(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      thrmin ,
    int      thrmax ,
    int      opt
)
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
しきい値最小値  
しきい値最大値  
拡張2値化オプション

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

thrmin

2値化する最小しきい値（下位8ビットのみ有効）

thrmax

2値化する最大しきい値（下位8ビットのみ有効）

opt

拡張2値化オプション設定

オプション	内容
0	2値化
1	反転2値化

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

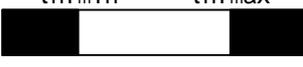
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	しきい値最小値範囲設定エラー
2 3	しきい値最大値範囲設定エラー
2 4	オプション設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

処理結果を下記に示します。

オプション	処理結果	しきい値の色
0	符号付8ビット $-128$ $127$ 符号なし8ビット $0$ $255$ $thrmin$ $thrmax$ 	白
1		黒
0	$thrmin$ $thrmax$ 	白
1		黒

# IP\_Invert

## 論理反転

### 機能

ソース画面の論理反転を行い、結果をデスティネーション画面に出力します。

演算式	$F = \text{NOT} ( f )$
-----	------------------------

F : デスティネーション画面

f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Invert(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_Minus

## 符号反転

### 機能

ソース画面の符号反転を行い、結果をデスティネーション画面に出力します。

演算式	$F = 0 - f$
-----	-------------

F : デスティネーション画面

f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Minus(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_Abs

## 絶対値

### 機能

ソース画面の絶対値化を行い、結果をデスティネーション画面に出力します。

演算式

$$F = | f |$$

F : デスティネーション画面

f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Abs(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_AddConst

## 定数加算

### 機能

ソース画面に定数を加算し、結果をデスティネーション画面に出力します。

演算式	$F = f + \text{constant}$
-----	---------------------------

F : デスティネーション画面

f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_AddConst(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      constant
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
定数

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**constant**

定数（下位8ビットのみ有効）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	定数設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_SubConst

## 定数減算

### 機能

ソース画面から定数を減算し、結果をデスティネーション画面に出力します。

演算式	$F = f - \text{constant}$
-----	---------------------------

F : デスティネーション画面

f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_SubConst(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      constant
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
定数

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**constant**

定数（下位8ビットのみ有効）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	定数設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## IP\_SubConstAbs

## 定数減算絶対値

### 機能

ソース画面から定数を減算し、絶対値をデスティネーション画面に出力します。

演算式

$$F = | f - \text{constant} |$$

F : デスティネーション画面

f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_SubConstAbs(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      constant
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
定数

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

constant

定数（下位8ビットのみ有効）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	定数設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## IP\_MultConst

## 定数乗算

### 機能

ソース画面と定数を乗算し、結果をデスティネーション画面に出力します。

演算式	$F = f \times \text{constant}$
-----	--------------------------------

F : デスティネーション画面

f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MultConst(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      constant ,
    int      scale
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
定数  
ダウンシフト量

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**constant**

定数（下位8ビットのみ有効）

**scale**

ダウンシフト量（有効範囲：0～15）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
22	定数設定値範囲外
23	ダウンシフト量範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

シフトダウン量を下表に示します。

ダウンシフト量	処理内容
0	演算結果ダウンシフトなし
1	演算結果 1 / 2
2	演算結果 1 / 4
3	演算結果 1 / 8
⋮	⋮
⋮	⋮
⋮	⋮
15	演算結果 1 / 3 2 7 6 8

# IP\_MinConst

## 定数比較 Min

### 機能

ソース画面と定数を比較し、値の小さい方をデスティネーション画面に出力します。

演算式	$F = \text{Min} ( f , \text{constant} )$
-----	--

F : デスティネーション画面  
f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MinConst(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      constant
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
定数

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

**constant**  
定数（下位8ビットのみ有効）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	定数設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_MaxConst

## 定数比較 Max

### 機能

ソース画面と定数を比較し、値の大きい方をデスティネーション画面に出力します。

演算式	$F = \text{Max} ( f , \text{constant} )$
-----	--

F : デスティネーション画面  
f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MaxConst(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      constant
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
定数

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

**constant**  
定数（下位8ビットのみ有効）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	定数設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# WriteConvertLUT

## 濃度変換データの設定

### 機能

濃度変換 (IP\_ConvertLUT) コマンドの濃度変換データを設定します。濃度変換の詳細については、ユーザズガイド第5章を参照して下さい。

また、ここで設定したLUTのデータは、ラベリングコマンドを実行すると書き換えられますので注意して下さい。

### コーディング

```
int
WriteConvertLUT(
    DEVID    devID ,
    int      *lut
)
```

### コメント

デバイスID  
濃度変換データ格納テーブル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*lut

濃度変換データ格納テーブル (32bit x 256) のポインタ。このテーブルに濃度変換のデータを設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

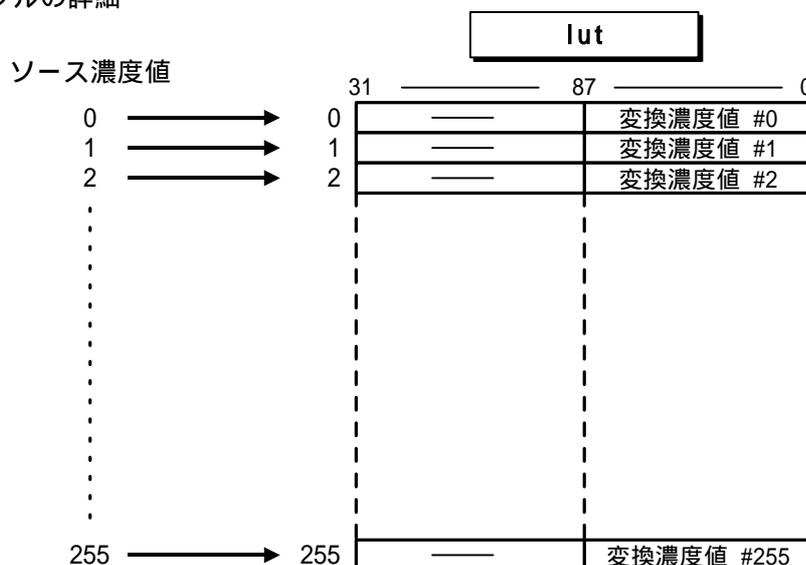
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

濃度変換テーブルの詳細



# IP\_ConvertLUT

## 濃度変換

### 機能

濃度変換テーブルの内容で、画面データを書き換えます。

演算式	$F = g[ f ]$
-----	--------------

F : デスティネーション画面  
f : ソース画面  
g : 濃度変換テーブル

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_ConvertLUT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドを発行する前には、必ず濃度変換テーブル書き込みコマンド ( WriteConvertLUT() ) にて、濃度変換テーブルへの書き込みを行って下さい。

# IP\_ShiftDown

## 定数シフトダウン

### 機能

ソース画面の画素データをシフトダウンし、結果をデスティネーション画面に出力します。

演算式

$$F = f \gg \text{constant}$$

F : デスティネーション画面

f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_ShiftDown(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      n
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
シフトダウン量

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

n

シフトダウン量

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外 (不当画面番号エラー)
22	パラメータエラー
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_ShiftUp

## 定数シフトアップ

### 機能

ソース画面の画素データをシフトアップし、結果をデスティネーション画面に出力します。

演算式

$$F = f \ll \text{constant}$$

F : デスティネーション画面

f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_ShiftUp(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      n
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
シフトアップ量

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

n

シフトアップ量

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外 (不当画面番号エラー)
22	パラメータエラー
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## IP\_AndConst

## 定数論理積

## 機能

ソース画面と定数との論理積をとり、結果をデスティネーション画面に出力します。

演算式	$F = f \& \text{constant}$
-----	----------------------------

F : デスティネーション画面  
f : ソース画面

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_AndConst(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      constant
)
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
定数

## パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

**constant**  
定数（下位8ビットのみ有効）

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
22	パラメータエラー
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

# IP\_Add

## 加算

### 機能

ソース画面 0 とソース画面 1 を加算し、結果をデスティネーション画面に出力します。

演算式	$F = f + g$
-----	-------------

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Add(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

### コメント

デバイスID

ソース0画面番号

ソース1画面番号

デスティネーション画面番号

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc0**

ソース0画面の画面番号

**ImgSrc1**

ソース1画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_Sub

## 減算

### 機能

ソース画面 0 とソース画面 1 を減算し、結果をデスティネーション画面に出力します。

演算式	$F = f - g$
-----	-------------

F: デスティネーション画面

f: ソース画面 0

g: ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Sub(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

### コメント

デバイスID

ソース0画面番号

ソース1画面番号

デスティネーション画面番号

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc0**

ソース0画面の画面番号

**ImgSrc1**

ソース1画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_SubAbs

## 減算絶対値

### 機能

ソース画面 0 とソース画面 1 の減算絶対値の結果を、デスティネーション画面に出力します。

演算式	$F =   f - g  $
-----	-----------------

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_SubAbs(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

### コメント

デバイス ID

ソース 0 画面番号

ソース 1 画面番号

デスティネーション画面番号

### パラメータ

devID

デバイス ID。OpenIPDevコマンドで得られた対応するボード番号のデバイス ID を指定して下さい。

ImgSrc0

ソース 0 画面の画面番号

ImgSrc1

ソース 1 画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

**IP\_Comb****係数付加算****機能**

ソース画面 0 と係数 a を乗算したものと、ソース画面 1 と係数 b を乗算したものを加算し、結果をデスティネーション画面に出力します。

演算式

$$F = f \times a + g \times b$$

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

**コーディング**

```
int
IP_Comb(
    DEVID      devID ,
    int        ImgSrc0 ,
    int        ImgSrc1 ,
    int        ImgDst ,
    int        scale ,
    int        a ,
    int        b
)
```

**コメント**

```
デバイスID
ソース0画面番号
ソース1画面番号
デスティネーション画面番号
ダウンシフト量
係数
係数
```

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc0**

ソース0画面の画面番号

**ImgSrc1**

ソース1画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量 ( 0 ~ 15 )

**a**

係数 a

**b**

係数 b

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 3	ダウンシフト量設定値範囲外
2 4	係数 a 設定値範囲外
2 5	係数 b 設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPErr) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

演算結果が 8 ビットを超える場合は、scaleにてダウンシフト (右ビットシフト) することが可能です。

ダウンシフト量	処理内容
0	演算結果ダウンシフトなし
1	演算結果 1 / 2
2	演算結果 1 / 4
3	演算結果 1 / 8
⋮	⋮
⋮	⋮
1 5	演算結果 1 / 3 2 7 6 8

**IP\_CombAbs****係数付加算絶対値****機能**

ソース画面 0 と係数 a を乗算したものと、ソース画面 1 と係数 b を乗算したものの加算絶対値の結果を、デスティネーション画面に出力します。

演算式

$$F = | f \times a + g \times b |$$

F: デスティネーション画面

f: ソース画面 0

g: ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

**コーディング**

```
int
IP_CombAbs(
    DEVID      devID ,
    int        ImgSrc0 ,
    int        ImgSrc1 ,
    int        ImgDst ,
    int        scale ,
    int        a ,
    int        b
)
```

**コメント**

```
デバイスID
ソース0画面番号
ソース1画面番号
デスティネーション画面番号
ダウンシフト量
係数
係数
```

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc0**

ソース0画面の画面番号

**ImgSrc1**

ソース1画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量 ( 0 ~ 15 )

**a**

係数 a

**b**

係数 b

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 3	ダウンシフト量設定値範囲外
2 4	係数 a 設定値範囲外
2 5	係数 b 設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

演算結果が 8 ビットを超える場合は、scaleにてダウンシフト (右ビットシフト) することが可能です。

ダウンシフト量	処理内容
0	演算結果ダウンシフトなし
1	演算結果 1 / 2
2	演算結果 1 / 4
3	演算結果 1 / 8
⋮	⋮
⋮	⋮
⋮	⋮
1 5	演算結果 1 / 3 2 7 6 8

# IP\_Mult

## 乗算

### 機能

ソース画面 0 とソース画面 1 を乗算し、結果をデスティネーション画面に出力します。

演算式	$F = f \times g$
-----	------------------

F: デスティネーション画面

f: ソース画面 0

g: ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Mult(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst ,
    int      scale
)
```

### コメント

デバイスID  
 ソース0画面番号  
 ソース1画面番号  
 デスティネーション画面番号  
 ダウンシフト量

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc0**

ソース0画面の画面番号

**ImgSrc1**

ソース1画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量 ( 0 ~ 15 )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 3	ダウンシフト量設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

演算結果が8ビットを超える場合は、scaleにてダウンシフト（右ビットシフト）することが可能です。

ダウンシフト量	処理内容
0	演算結果ダウンシフトなし
1	演算結果 1 / 2
2	演算結果 1 / 4
3	演算結果 1 / 8
⋮	⋮
⋮	⋮
⋮	⋮
15	演算結果 1 / 32768

# IP\_Average

平均

## 機能

ソース画面 0 とソース画面 1 の平均を、結果をデスティネーション画面に出力します。

演算式

$$F = ( f + g ) / 2$$

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_Average(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

## コメント

デバイス ID

ソース 0 画面番号

ソース 1 画面番号

デスティネーション画面番号

## パラメータ

**devID**

デバイス ID。OpenIPDev コマンドで得られた対応するボード番号のデバイス ID を指定して下さい。

**ImgSrc0**

ソース 0 画面の画面番号

**ImgSrc1**

ソース 1 画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

# IP\_Min

## 比較 Min

### 機能

ソース画面 0 とソース画面 1 を比較し、値の小さい方をデスティネーション画面に出力します。

演算式	$F = \text{Min} ( f , g )$
-----	----------------------------

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Min(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

### コメント

デバイスID

ソース0画面番号

ソース1画面番号

デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc0

ソース0画面の画面番号

ImgSrc1

ソース1画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_Max

## 比較 Max

### 機能

ソース画面 0 とソース画面 1 を比較し、値の大きい方をデスティネーション画面に出力します。

演算式	$F = \text{Max} ( f , g )$
-----	----------------------------

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Max(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

### コメント

デバイスID

ソース0画面番号

ソース1画面番号

デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc0

ソース0画面の画面番号

ImgSrc1

ソース1画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

**IP\_SubConstAbsAdd****定数減算絶対値和****機能**

ソース画面 0 とソース画面 1 の定数減算絶対値和の結果をデスティネーション画面に出力します。

演算式

$$F = | f - a | + | g - b |$$

F: デスティネーション画面

f: ソース画面 0

g: ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

**コーディング**

```
int
IP_SubConstAbsAdd(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst ,
    int      scale ,
    int      a ,
    int      b
)
```

**コメント**

デバイス ID  
 ソース 0 画面番号  
 ソース 1 画面番号  
 デスティネーション画面番号  
 ダウンシフト量  
 係数  
 係数

**パラメータ****devID**

デバイス ID。OpenIPDevコマンドで得られた対応するボード番号のデバイス ID を指定して下さい。

**ImgSrc0**

ソース 0 画面の画面番号

**ImgSrc1**

ソース 1 画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量 ( 0 ~ 15 )

**a**

係数 a

**b**

係数 b

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 3	ダウンシフト量設定値範囲外
2 4	係数 a 設定値範囲外
2 5	係数 b 設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPErr) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

演算結果が 8 ビットを超える場合は、scaleにてダウンシフト (右ビットシフト) することが可能です。

ダウンシフト量	処理内容
0	演算結果ダウンシフトなし
1	演算結果 1 / 2
2	演算結果 1 / 4
3	演算結果 1 / 8
⋮	⋮
⋮	⋮
⋮	⋮
1 5	演算結果 1 / 3 2 7 6 8

**IP\_SubConstMultAdd****定数減算自乗和****機能**

ソース画面 0 とソース画面 1 の定数減算自乗和の結果をデスティネーション画面に出力します。

演算式

$$F = (f - a) * (f - a) + (g - b) * (g - b)$$

F: デスティネーション画面

f: ソース画面 0

g: ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

**コーディング**

```
int
IP_SubConstMultAdd(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst ,
    int      scale ,
    int      a ,
    int      b
)
```

**コメント**

デバイス ID  
 ソース 0 画面番号  
 ソース 1 画面番号  
 デスティネーション画面番号  
 ダウンシフト量  
 係数  
 係数

**パラメータ****devID**

デバイス ID。OpenIPDevコマンドで得られた対応するボード番号のデバイス ID を指定して下さい。

**ImgSrc0**

ソース 0 画面の画面番号

**ImgSrc1**

ソース 1 画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量 ( 0 ~ 15 )

**a**

係数 a

**b**

係数 b

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 3	ダウンシフト量設定値範囲外
2 4	係数 a 設定値範囲外
2 5	係数 b 設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPErr) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

演算結果が 8 ビットを超える場合は、scaleにてダウンシフト (右ビットシフト) することが可能です。

ダウンシフト量	処理内容
0	演算結果ダウンシフトなし
1	演算結果 1 / 2
2	演算結果 1 / 4
3	演算結果 1 / 8
⋮	⋮
⋮	⋮
1 5	演算結果 1 / 3 2 7 6 8

**IP\_SubConstMult****定数減算積****機能**

ソース画面 0 とソース画面 1 の定数減算積の結果をデスティネーション画面に出力します。

演算式

$$F = ( f - a ) * ( g - b )$$

F: デスティネーション画面

f: ソース画面 0

g: ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

**コーディング**

```
int
IP_SubConstMult(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst ,
    int      scale ,
    int      a ,
    int      b
)
```

**コメント**

デバイスID  
 ソース0画面番号  
 ソース1画面番号  
 デスティネーション画面番号  
 ダウンシフト量  
 係数  
 係数

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc0**

ソース0画面の画面番号

**ImgSrc1**

ソース1画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量 ( 0 ~ 15 )

**a**

係数 a

**b**

係数 b

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 3	ダウンシフト量設定値範囲外
2 4	係数 a 設定値範囲外
2 5	係数 b 設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPErr) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

演算結果が8ビットを超える場合は、scaleにてダウンシフト(右ビットシフト)することが可能です。

ダウンシフト量	処理内容
0	演算結果ダウンシフトなし
1	演算結果 1 / 2
2	演算結果 1 / 4
3	演算結果 1 / 8
⋮	⋮
⋮	⋮
⋮	⋮
1 5	演算結果 1 / 3 2 7 6 8

**IP\_CombDrop****係数付加算(切り捨て)****機能**

四捨五入を行わない切り捨てバージョンで、係数付加算を切り捨てで行う。

係数付加算：  $a * f + b * g$   
 $DST(i, j) = a * SRC0(i, j) + b * SRC1(i, j)$

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

**コーディング**

```
int
IP_CombDrop(
  DEVID
  int
  int
  int
  int
  int
  int
  int
)
```

```
devID ,
ImgSrc0 ,
ImgSrc1 ,
ImgDst ,
scale ,
a ,
b
```

**コメント**

```
デバイスID
ソース0画面番号
ソース1画面番号
デスティネーション画面番号
スケールダウン量
定数a
定数b
```

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc0**

ソース0画面番号

**ImgSrc1**

ソース1画面番号

**ImgDst**

デスティネーション画面番号

**scale**

スケールダウン量 ( 0 ~ 15 )

**a**

定数 a ( 有効範囲 -256 ~ 255 )

**b**

定数 b ( 有効範囲 -256 ~ 255 )

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 1 , 1 2 , 1 3	テンプレート画面の不当画面番号エラー
2 3	スケールダウン設定範囲外
2 4 , 2 5	パラメータ設定範囲外
-	エラーリセットコマンド (ClearIPErrror) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# IP\_And

## 論理積

### 機能

ソース画面 0 とソース画面 1 の論理積を、デスティネーション画面に出力します。

演算式	F = f and g
-----	-------------

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_And(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

### コメント

デバイスID

ソース0画面番号

ソース1画面番号

デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc0

ソース0画面の画面番号

ImgSrc1

ソース1画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## IP\_Or

## 論理和

## 機能

ソース画面 0 とソース画面 1 の論理和を、デスティネーション画面に出力します。

演算式

F = f or g

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_Or(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

## コメント

デバイスID  
 ソース0画面番号  
 ソース1画面番号  
 デスティネーション画面番号

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc0

ソース0画面の画面番号

ImgSrc1

ソース1画面の画面番号

ImgDst

デスティネーション画面の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

# IP\_Xor

## 排他的論理和

### 機能

ソース画面 0 とソース画面 1 の排他的論理和を、デスティネーション画面に出力します。

演算式	$F = f \text{ xor } g$
-----	------------------------

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Xor(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

### コメント

デバイスID

ソース0画面番号

ソース1画面番号

デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc0

ソース0画面の画面番号

ImgSrc1

ソース1画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_InvertAnd

## 否定論理積

### 機能

ソース画面0の否定画面とソース画面1の論理積を、デスティネーション画面に出力します。

演算式	$F = \sim f \text{ and } g$
-----	-----------------------------

F : デスティネーション画面

f : ソース画面0

g : ソース画面1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_InvertAnd(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

### コメント

デバイスID

ソース0画面番号 (否定画面)

ソース1画面番号

デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc0

ソース0画面の画面番号 (否定画面)

ImgSrc1

ソース1画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 (不当画面番号エラー)
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_InvertOr

## 否定論理和

### 機能

ソース画面 0 の否定画面とソース画面 1 の論理和を、デスティネーション画面に出力します。

演算式	$F = \sim f \text{ or } g$
-----	----------------------------

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_InvertOr(
    DEVID    devID ,
    int      imgSrc0 ,
    int      imgSrc1 ,
    int      imgDst
)
```

### コメント

デバイスID

ソース0画面番号 (否定画面)

ソース1画面番号

デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgSrc0

ソース0画面の画面番号 (否定画面)

imgSrc1

ソース1画面の画面番号

imgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 1、1 3	画面空き領域、または範囲外 (不当画面番号エラー)
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IP\_Xnor

## 排他的否定論理和

### 機能

ソース画面 0 の否定画面とソース画面 1 の排他的論理和を、デスティネーション画面に出力します。

演算式	$F = \sim f \text{ xor } g$
-----	-----------------------------

F : デスティネーション画面

f : ソース画面 0

g : ソース画面 1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Xnor(
    DEVID    devID ,
    int      ImgSrc0 ,
    int      ImgSrc1 ,
    int      ImgDst
)
```

### コメント

デバイスID

ソース0画面番号(否定画面)

ソース1画面番号

デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc0

ソース0画面の画面番号(否定画面)

ImgSrc1

ソース1画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 11, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## IP\_PickNoise4

### 2値画像ノイズ除去(4連結)

#### 機能

2値画像中の独立した1画素(黒で囲まれた白または、白で囲まれた黒)を除去します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_PickNoise4(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

処理内容は、IP\_PickNoise8()の詳細情報の欄を参照して下さい。

## IP\_PickNoise8

### 2値画像ノイズ除去(8連結)

#### 機能

2値画像中の独立した1画素(黒で囲まれた白または、白で囲まれた黒)を除去します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_PickNoise8(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

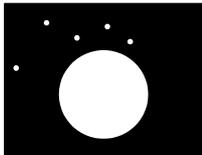
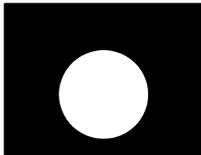
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10 , 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

4連結と8連結の処理内容を以下に示します。

	4 連結		8 連結																			
処 理 内 容	処理前	→	処理後																			
	<table border="1"><tr><td></td><td>1</td><td></td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td></td><td>0</td><td></td></tr></table>		1		0	1	0		0			<table border="1"><tr><td></td><td>1</td><td></td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td></td><td>0</td><td></td></tr></table>		1		0	1	0		0		
		1																				
0	1	0																				
	0																					
	1																					
0	1	0																				
	0																					
<table border="1"><tr><td></td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td></td><td>0</td><td></td></tr></table>		0		0	1	0		0			<table border="1"><tr><td></td><td>0</td><td></td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td></td><td>0</td><td></td></tr></table>		0		0	0	0		0			
	0																					
0	1	0																				
	0																					
	0																					
0	0	0																				
	0																					
<table border="1"><tr><td></td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td></td><td>1</td><td></td></tr></table>		1		1	0	1		1			<table border="1"><tr><td></td><td>1</td><td></td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td></td><td>1</td><td></td></tr></table>		1		1	1	1		1			
	1																					
1	0	1																				
	1																					
	1																					
1	1	1																				
	1																					
処 理 例		→																				
																						
	白の独立点除去																					

## IP\_Outline4

## 2値画像輪郭抽出(4連結)

### 機能

2値画像の白の部分の輪郭を抽出します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Outline4(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

処理内容は、IP\_Outline8()の詳細情報の欄を参照して下さい。

## IP\_Outline8

## 2値画像輪郭抽出(8連結)

### 機能

2値画像の白の部分の輪郭を抽出します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Outline8(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

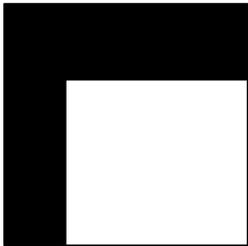
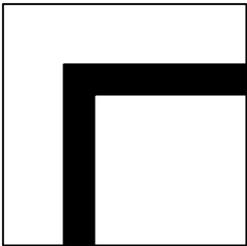
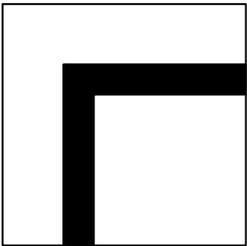
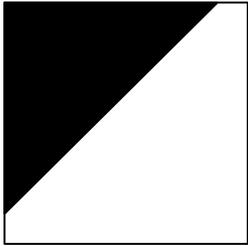
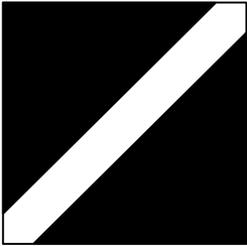
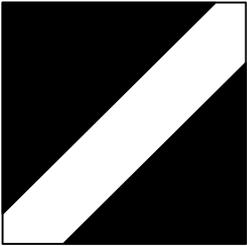
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

4連結と8連結の処理内容を以下に示します。

	処理前	4連結処理後	8連結処理後
処 理 例 1	<pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1                     </pre> 	<pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0                     </pre> 	<pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0                     </pre> 
処 理 例 2	<pre> 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1                     </pre> 	<pre> 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0                     </pre> 	<pre> 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0                     </pre> 

## IP\_Dilation4

### 2値画像膨張(4連結)

#### 機能

2値画像の白の部分を外側へ1画素膨張させます。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_Dilation4(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10 , 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

処理内容は、IP\_Dilation8()の詳細情報の欄を参照して下さい。

## IP\_Dilation8

### 2値画像膨張(8連結)

#### 機能

2値画像の白の部分を外側へ1画素膨張させます。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_Dilation8(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

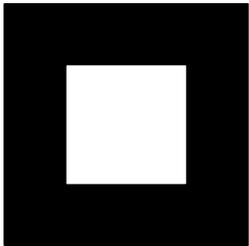
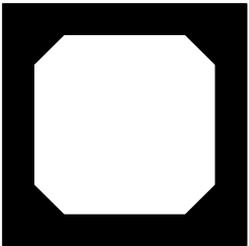
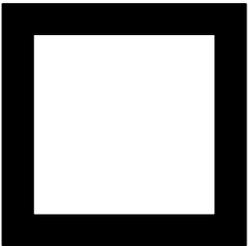
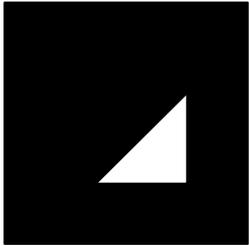
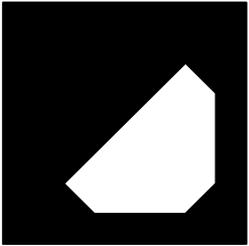
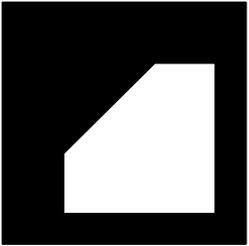
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

4連結と8連結の処理内容を以下に示します。

	処理前	4連結処理後	8連結処理後
処 理 例 1	<pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0                     </pre> 	<pre> 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0                     </pre> 	<pre> 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0                     </pre> 
	<pre> 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0                     </pre> 	<pre> 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0                     </pre> 	<pre> 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0                     </pre> 

## IP\_Erosion4

### 2値画像収縮(4連結)

#### 機能

2値画像の白の部分を内側へ1画素収縮させます。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_Erosion4(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

#### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

処理内容は、IP\_Erosion8()の詳細情報の欄を参照して下さい。

## IP\_Erosion8

## 2値画像収縮(8連結)

### 機能

2値画像の白の部分を内側へ1画素収縮させます。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Erosion8(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

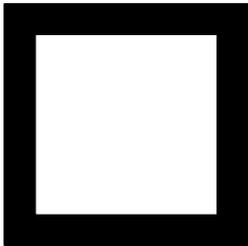
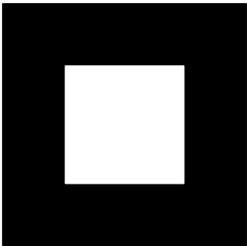
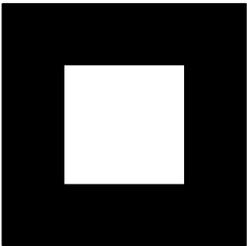
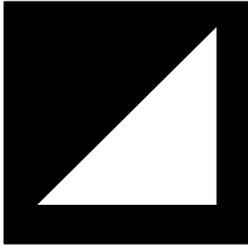
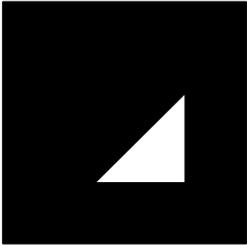
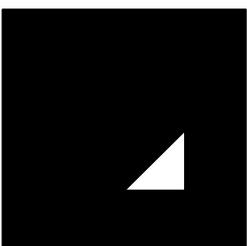
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10 , 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

4連結と8連結の処理内容を以下に示します。

	処理前	4連結処理後	8連結処理後
処 理 例 1	<pre> 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0                     </pre> 	<pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0                     </pre> 	<pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0                     </pre> 
処 理 例 2	<pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0                     </pre> 	<pre> 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0                     </pre> 	<pre> 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0                     </pre> 

## IP\_Thin4

### 2値画像細線化(4連結)

#### 機能

ソース画面（2値画像）に対し、4連結で細線化処理を行い、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_Thin4(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

#### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

処理内容は、IP\_Thin8()の詳細情報の欄を参照して下さい。

## IP\_Thin8

## 2値画像細線化(8連結)

### 機能

ソース画面（2値画像）に対し、8連結で細線化処理を行い、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Thin8(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

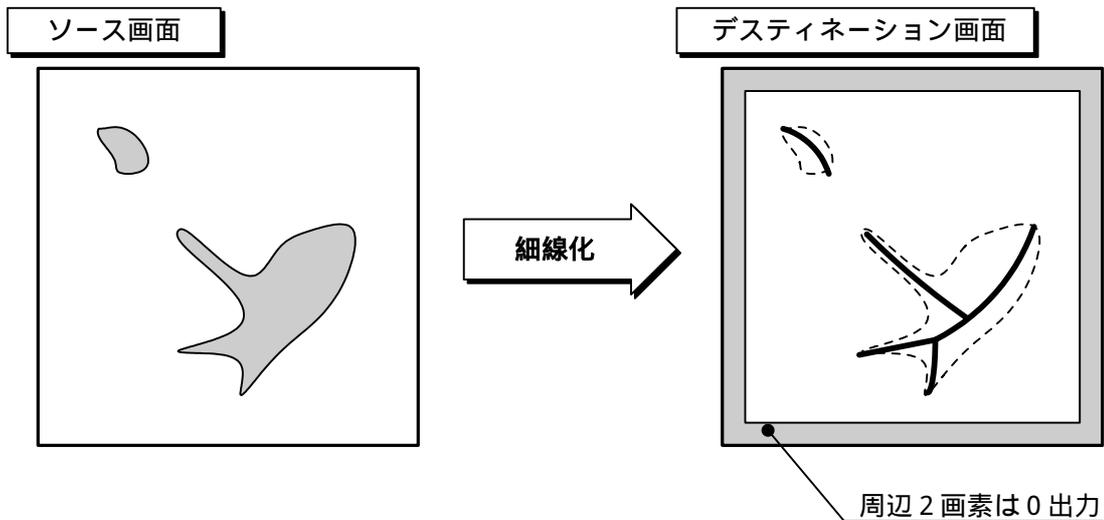
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

細線化処理では、入力画面（2値画像）の指定領域内の全画素に対して細線化処理を行い結果を出力します。また、指定領域の周辺では、0を出力します。



1回の処理で、 $5 \times 5$ 画素領域で局所的な細線化処理を行います。そのため、完全に細線化処理を行うためには、複数回この処理を行う必要があります。

以下に、細線化処理を完全に行う場合のプロセス例を示します。

#### 細線化処理を完全に行う場合のプロセス例

2値画像の管理を、物体：255，背景：0で管理している場合、物体（255）の面積の変化がなくなったことをヒストグラム処理を用いて検知します。つまり、ヒストグラムの255の累積値の変化がなくなるまで処理を繰り返すわけです。

```
int ImgSrc,ImgDst;
long Tbl[256];
IPG0FeatureTbl FtrTbl;
int count;

count = 0;
for(;;){
    IP_Thin4(devID, ImgSrc, ImgDst);

    ChangeImgDataType(_devID, ImgDst, UNSIGN8_DATA);
    IP_Histogram(devID, ImgDst, Tbl, &FtrTbl, 0);
    if(count == Tbl[255]) break;

    count = Tbl[255];
    IP_Copy(devID, ImgDst, ImgSrc);
}
```

## IP\_Shrink4

### 2値画像縮退化(4連結)

#### 機能

ソース画面（2値画像）に対し、4連結で縮退化処理を行い、デスティネーション画面に出力します。また、縮退化処理は、細線化処理の端点を縮退する処理です。物体数の解析等に使用できます。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_Shrink4(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

処理内容は、IP\_Shrink8()の詳細情報の欄を参照して下さい。

## IP\_Shrink8

### 2値画像縮退化(8連結)

#### 機能

ソース画面（2値画像）に対し、8連結で縮退化処理を行い、デスティネーション画面に出力します。また、縮退化処理は、細線化処理の端点を縮退する処理です。物体数の解析等に使用できます。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_Shrink8(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

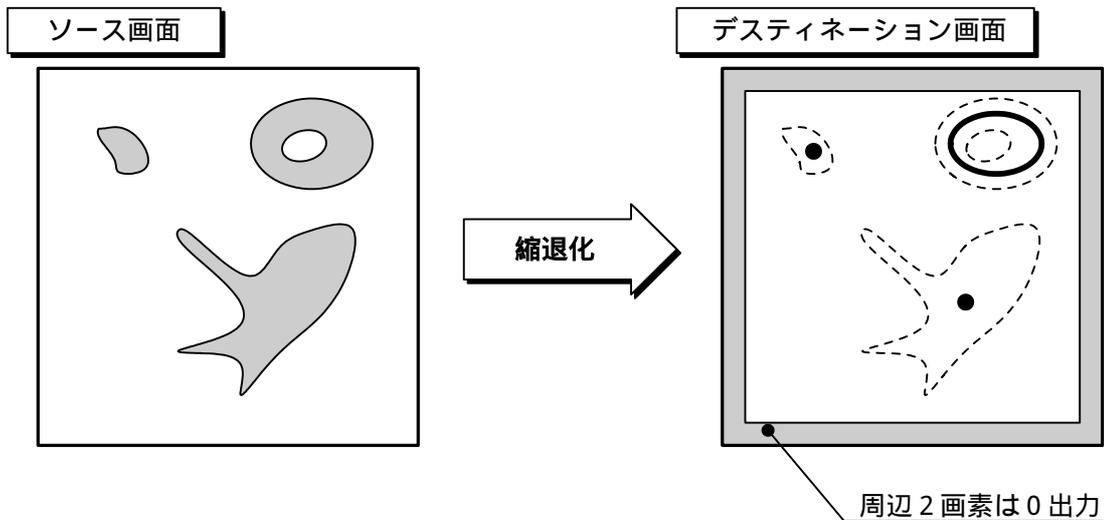
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10 , 13	画面空き領域、または範囲外（不当画面番号エラー）
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

縮退化処理では、入力画面（2値画像）の指定領域内の全画素に対して縮退化処理を行い結果を出力します。また、指定領域の周辺では、0を出力します。



1回の処理で、 $5 \times 5$ 画素領域で局所的な縮退化処理を行います。そのため、完全に縮退化処理を行うためには、複数回この処理を行う必要があります。

以下に、縮退化処理を完全に行う場合のプロセス例を示します。

#### 縮退化処理を完全に行う場合のプロセス例

2値画像の管理を、物体：255、背景：0で管理している場合、物体（255）の面積の変化がなくなったことをヒストグラム処理を用いて検知します。つまり、ヒストグラムの255の累積値の変化がなくなるまで処理を繰り返すわけです。

```
int ImgSrc,ImgDst;
long Tbl[256];
IPGOFeatureTbl FtrTbl;
int count;

count = 0;
for(;;){
    IP_Shrink4(devID, ImgSrc, ImgDst);

    ChangeImgDataType(devID, ImgDst, UNSIGN8_DATA);
    IP_Histogram(devID, ImgDst, Tbl, &FtrTbl, 0);
    if(count == Tbl[255]) break;

    count = Tbl[255];
    IP_Copy(devID, ImgDst, ImgSrc);
}
```

# IP\_SmoothFLT

## 平滑化

### 機能

濃淡画面の平滑化を行います。IP\_EdgeFLT() コマンドとの違いは処理ウィンドウ内の周辺処理にあります。IP\_SmoothFLT() コマンドは、周辺部をソース画像の注目画素の画素データで描画します。

また、IP\_SmoothFLT() コマンドでは、フィルタリング自体はハード的に行われますが、周辺部の処理はソフト的に行われるため、パイプライン処理を実行することはできません。パイプライン処理を行う場合は、本コマンドの代わりにIP\_EdgeFLT() コマンドを使用して下さい。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_SmoothFLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      scale ,
    int      *COEFF
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

**scale**  
ダウンシフト量 ( 0 ~ 15 )

**\*COEFF**  
荷重係数テーブルのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

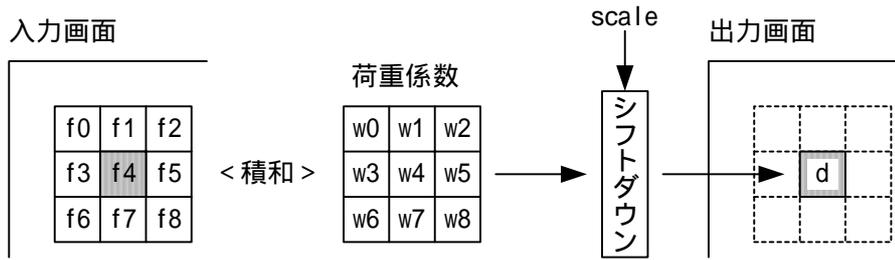
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2	ダウンシフト量範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

平滑化では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 3 × 3 近傍の局所領域で与えられた荷重係数との積和演算を行います。



$$d = ( f_0 \times w_0 + f_1 \times w_1 + f_2 \times w_2 + f_3 \times w_3 + f_4 \times w_4 + f_5 \times w_5 + f_6 \times w_6 + f_7 \times w_7 + f_8 \times w_8 ) \gg scale$$

係数テーブルのフォーマット及び荷重係数の例を、下記に示します。

荷重係数

#0	#1	#2
#3	#4	#5
#6	#7	#8

荷重係数の例

例 1

16	16	0
16	16	0
0	0	0

例 2

7	7	7
7	8	7
7	7	7

	31	—————	87	—————	0
COEFF [0]		—————	係数 # 0		
[1]		—————	# 1		
[2]		—————	# 2		
[3]		—————	# 3		
[4]		—————	# 4		
[5]		—————	# 5		
[6]		—————	# 6		
[7]		—————	▼ # 7		
[8]		—————	係数 # 8		

例 3

5	8	5
8	12	8
5	8	5

例 4

5	5	5
5	20	5
5	5	5

本例の場合、ダウンシフト量は 6 です。

## IP\_EdgeFLT

### 濃淡画像輪郭強調

#### 機能

濃淡画面の輪郭強調を行います。IP\_SmoothFLT()コマンドとの違いは処理ウィンドウ内の周辺処理にあります。IP\_EdgeFLT()コマンドは、周辺部を「0」で描画します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_EdgeFLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      scale ,
    int      *COEFF
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

#### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量(0 ~ 15)

**\*COEFF**

荷重係数テーブルのポインタ。COEFFの詳細については、IP\_SmoothFLT()コマンドの詳細情報を参照して下さい。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

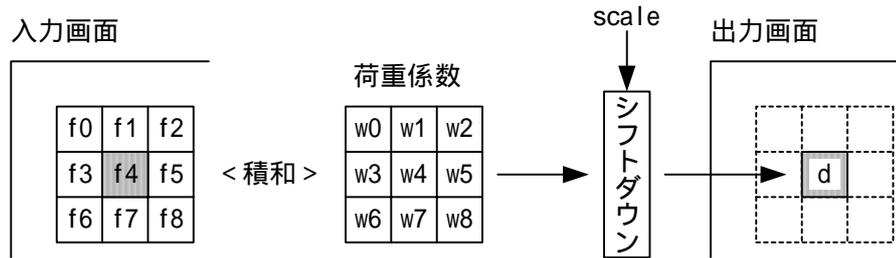
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10 , 13	画面空き領域、または範囲外(不当画面番号エラー)
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

輪郭強調では、入力画面の指定領域内の全画素に対して、注目画素を中心とした  $3 \times 3$  近傍の局所領域で与えられた荷重係数との積和演算を行います。  
 また、対象領域の周辺 ( $3 \times 3$  近傍に領域外が含まれる場合) 処理の場合は 0 を出力します。



$$d = ( f_0 \times w_0 + f_1 \times w_1 + f_2 \times w_2 + f_3 \times w_3 + f_4 \times w_4 + f_5 \times w_5 + f_6 \times w_6 + f_7 \times w_7 + f_8 \times w_8 ) \gg \text{scale}$$

荷重係数の例を、下記に示します。また、係数テーブルのフォーマットは、IP\_SmoothFLTの詳細情報の欄を参照して下さい。

## 荷重係数の例

例 1	<table style="border-collapse: collapse; text-align: center;"> <tr><td>-1</td><td>0</td><td>-1</td></tr> <tr><td>-1</td><td>0</td><td>-1</td></tr> <tr><td>-1</td><td>0</td><td>-1</td></tr> </table>	-1	0	-1	-1	0	-1	-1	0	-1	例 2	<table style="border-collapse: collapse; text-align: center;"> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table>	-1	-1	-1	0	0	0	-1	-1	-1
-1	0	-1																			
-1	0	-1																			
-1	0	-1																			
-1	-1	-1																			
0	0	0																			
-1	-1	-1																			

本例の場合、ダウンシフト量は 0 です。

## IP\_EdgeFLTAbs

### 濃淡画像輪郭強調(絶対値)

#### 機能

濃淡画面の輪郭強調（絶対値）を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_EdgeFLTAbs(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      scale ,
    int      *COEFF
)
```

#### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

#### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量（0～15）

**\*COEFF**

荷重係数テーブルのポインタ。COEFFの詳細については、IP\_SmoothFLT()コマンドの詳細情報を参照して下さい。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

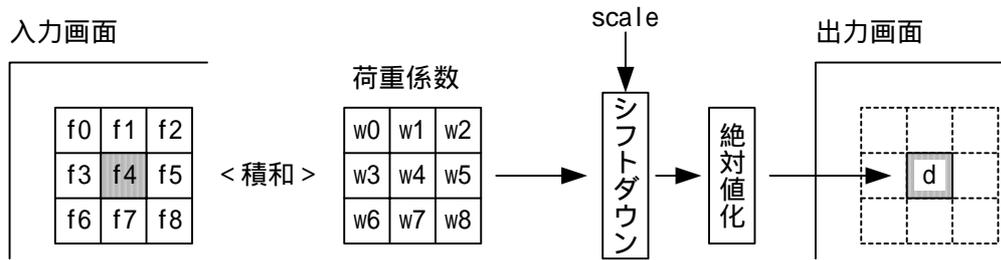
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

輪郭強調（絶対値）では、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域で与えられた荷重係数との積和演算を行い、結果を絶対値化します。また、対象領域の周辺（3×3近傍に領域外が含まれる場合）処理の場合は0を出力します。



$$d = | ( f_0 \times w_0 + f_1 \times w_1 + f_2 \times w_2 + f_3 \times w_3 + f_4 \times w_4 + f_5 \times w_5 + f_6 \times w_6 + f_7 \times w_7 + f_8 \times w_8 ) \gg \text{scale} |$$

係数テーブルのフォーマットは、IP\_SmoothFLTの詳細情報の欄を、荷重係数の例は、IP\_EdgeFLTの詳細情報の欄を参照して下さい。

# IP\_LapI4FLT

## ラブラシアン(4連結)

### 機能

濃淡画像において、下記に示すフィルタで2次微分を行います。

0	-1	0
-1	4	-1
0	-1	0

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_LapI4FLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPErr ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

対象領域の周辺 ( 3 × 3 近傍に領域外が含まれる部分 ) 処理の場合は 0 を出力します。

# IP\_LapI8FLT

## ラプラシアン(8連結)

### 機能

濃淡画像において、下記に示すフィルタで2次微分を行います。

-1	-1	-1
-1	8	-1
-1	-1	-1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_LapI8FLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

対象領域の周辺 ( 3 × 3 近傍に領域外が含まれる部分 ) 処理の場合は 0 を出力します。

## IP\_LapI4FLTAbs

## ラプラシアン(4連結・絶対値)

## 機能

濃淡画像において、下記に示すフィルタで2次微分(絶対値)を行います。

0	-1	0
-1	4	-1
0	-1	0

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_LapI4FLTAbs(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

## パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

対象領域の周辺(3×3近傍に領域外が含まれる部分)処理の場合は0を出力します。

## IP\_LapI8FLTAbs

## ラプラシアン(8連結・絶対値)

## 機能

濃淡画像において、下記に示すフィルタで2次微分(絶対値)を行います。

-1	-1	-1
-1	8	-1
-1	-1	-1

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_LapI8FLTAbs(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

## パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

対象領域の周辺(3×3近傍に領域外が含まれる部分)処理の場合は0を出力します。

# IP\_LineFLT

## ラインフィルタ

### 機能

ソース画面に対し、ユーザの登録パターンでラインフィルタ処理を行い、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_LineFLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      scale ,
    int      *COEFF
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

**scale**  
ダウンシフト量 ( 0 ~ 15 )

**\*COEFF**  
荷重係数テーブルのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

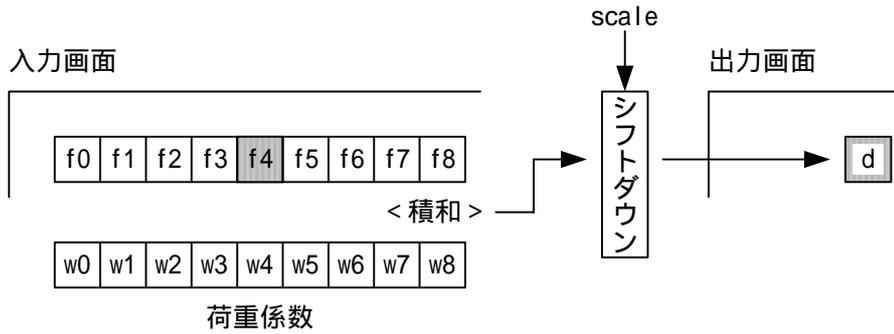
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10 , 13	画面空き領域、または範囲外 ( 不当画面番号エラー )
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPErrorTable ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

ラインフィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした 1 × 9 近傍の局所領域で与えられた荷重係数との積和演算を行います。また、対象領域の周辺 ( 1 × 9 近傍に領域外が含まれる部分 ) 処理の場合は不定になります。



$$d = ( f_0 \times w_0 + f_1 \times w_1 + f_2 \times w_2 + f_3 \times w_3 + f_4 \times w_4 + f_5 \times w_5 + f_6 \times w_6 + f_7 \times w_7 + f_8 \times w_8 ) \gg scale$$

係数テーブルのフォーマット及び荷重係数の例を、下記に示します。

荷重係数

#0	#1	#2	#3	#4	#5	#6	#7	#8
----	----	----	----	----	----	----	----	----

荷重係数の例 ( 横方向の平滑化 )

7	7	7	7	7	7	7	7	7
---	---	---	---	---	---	---	---	---

本例の場合、ダウンシフト量は 6 です。

	31	87	0
COEFF [0]	_____ 係数 # 0		
[1]	_____ # 1		
[2]	_____ # 2		
[3]	_____ # 3		
[4]	_____ # 4		
[5]	_____ # 5		
[6]	_____ # 6		
[7]	_____ ▼ # 7		
[8]	_____ 係数 # 8		

## IP\_LineFLTAbs

## ラインフィルタ(絶対値)

### 機能

ソース画面に対し、ユーザの登録パターンでラインフィルタ処理（絶対値）を行い、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_LineFLTAbs(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      scale ,
    int      *COEFF
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

**scale**  
ダウンシフト量（0～15）

**\*COEFF**  
荷重係数テーブルのポインタ。COEFFの詳細については、IP\_LineFLT()コマンドの詳細情報を参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

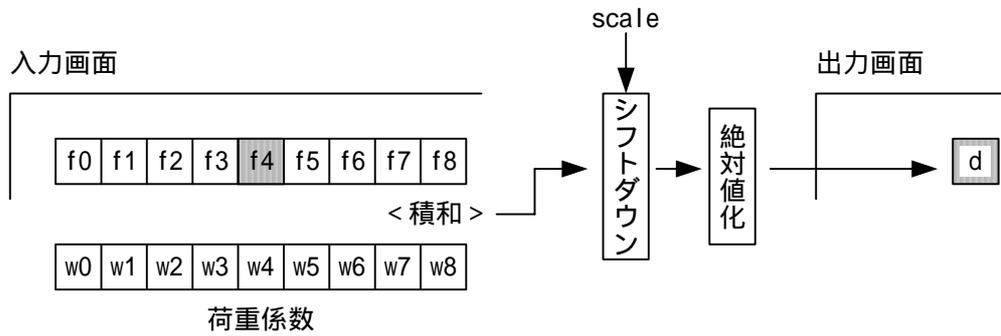
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	ダウンシフト量範囲外
4 1	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ラインフィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした1×9近傍の局所領域で与えられた荷重係数との積和演算を行い、絶対値化します。また、対象領域の周辺(1×9近傍に領域外が含まれる部分)処理の場合は不定になります。



$$d = | ( f_0 \times w_0 + f_1 \times w_1 + f_2 \times w_2 + f_3 \times w_3 + f_4 \times w_4 + f_5 \times w_5 + f_6 \times w_6 + f_7 \times w_7 + f_8 \times w_8 ) \gg \text{scale} |$$

係数テーブルのフォーマット及び荷重係数の例は、IP\_LineFLTの詳細情報の欄を参照して下さい。

## IP\_MinFLT

## 局所最小値フィルタ

### 機能

ソース画面に対し、パラメータで指定されたパターンで最小濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MinFLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      calptn
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
演算パターン

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**calptn**

演算パターン（下位9ビット有効）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

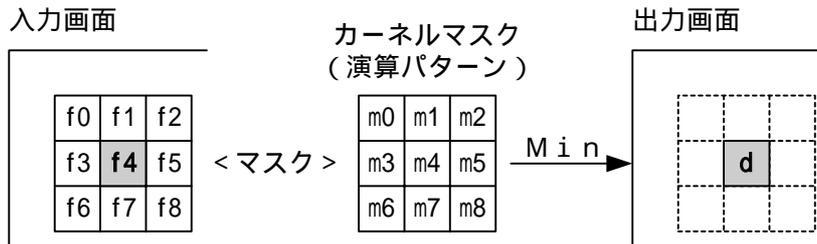
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

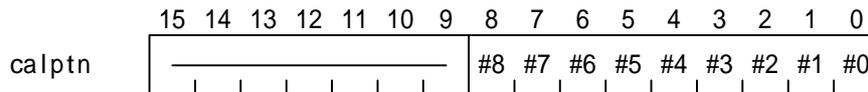
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

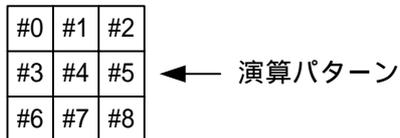
局所最小値フィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域で与えられた演算パターンでマスク処理し、有効となった画素の中から最小の画素を出力します。周辺では中央画素をそのまま出力します。



パラメータcalptnのフォーマットを下記に示します。



0 : 演算マスク  
1 : 演算イネーブル



# IP\_MinFLT4

## 局所最小値フィルタ(4連結)

### 機能

ソース画面に対し、4連結で最小濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MinFLT4(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

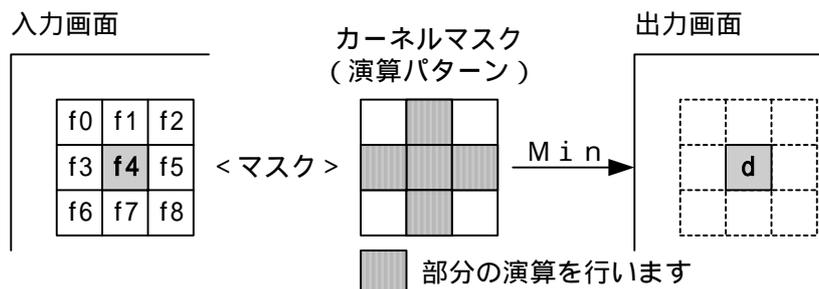
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

局所最小値フィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域を4連結の演算パターンでマスク処理し、有効となった画素の中から最小の画素を出力します。周辺では中央画素をそのまま出力します。



# IP\_MinFLT8

## 局所最小値フィルタ(8連結)

### 機能

ソース画面に対し、8連結で最小濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MinFLT8(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

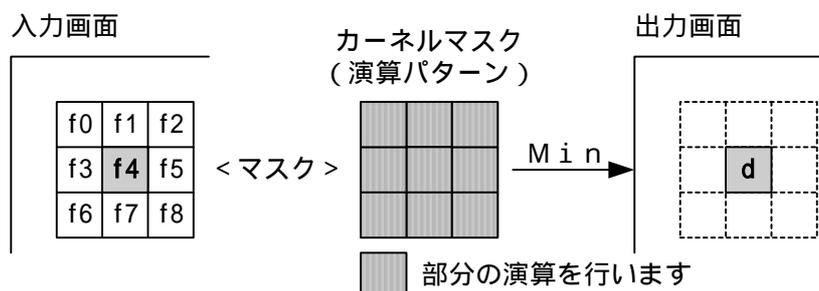
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

局所最小値フィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域を8連結の演算パターンでマスク処理し、有効となった画素の中から最小の画素を出力します。周辺では中央画素をそのまま出力します。



## IP\_MaxFLT

## 局所最大値フィルタ

### 機能

ソース画面に対し、パラメータで指定されたパターンで最大濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MaxFLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      calptn
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
演算パターン

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面の画面番号

#### ImgDst

デスティネーション画面の画面番号

#### calptn

演算パターン（下位9ビット有効）。詳細については、IP\_MinFLT()コマンドの詳細情報を参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

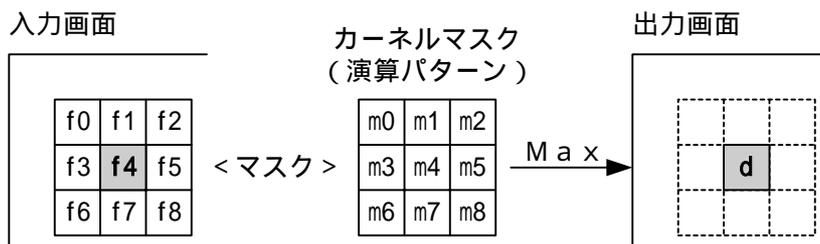
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

局所最大値フィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域で与えられた演算パターンでマスク処理し、有効となった画素の中から最大の画素を出力します。周辺では中央画素をそのまま出力します。



パラメータcalptnのフォーマットは、IP\_MinFLTを参照して下さい。。

# IP\_MaxFLT4

## 局所最大値フィルタ(4連結)

### 機能

ソース画面に対し、4連結で最大濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MaxFLT4(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

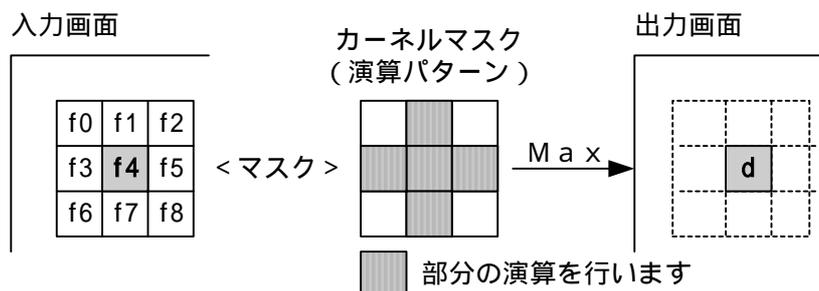
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

局所最大値フィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域を4連結の演算パターンでマスク処理し、有効となった画素の中から最大の画素を出力します。周辺では中央画素をそのまま出力します。



# IP\_MaxFLT8

## 局所最大値フィルタ(8連結)

### 機能

ソース画面に対し、8連結で最大濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MaxFLT8(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

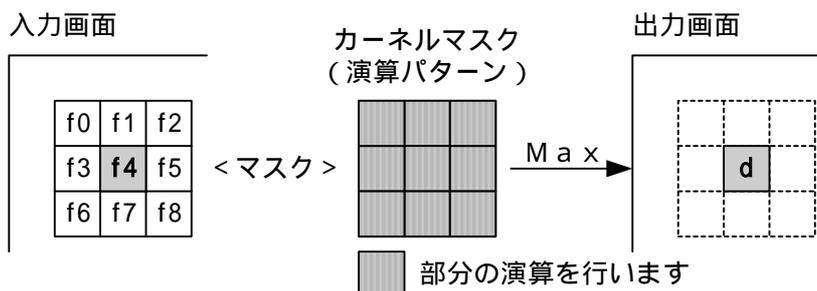
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

局所最大値フィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域を8連結の演算パターンでマスク処理し、有効となった画素の中から最大の画素を出力します。周辺では中央画素をそのまま出力します。



## IP\_LineMinFLT

## ライン局所最小値フィルタ

### 機能

ソース画面に対し、パラメータで指定されたパターンで最小濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_LineMinFLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      calptn
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
演算パターン

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**calptn**

演算パターン（下位9ビット有効）

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

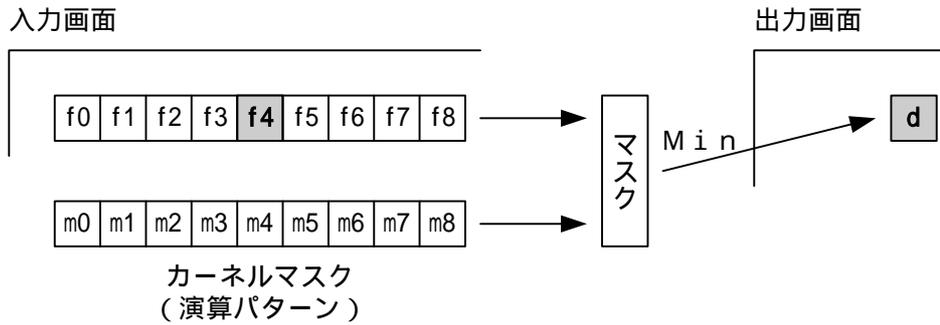
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

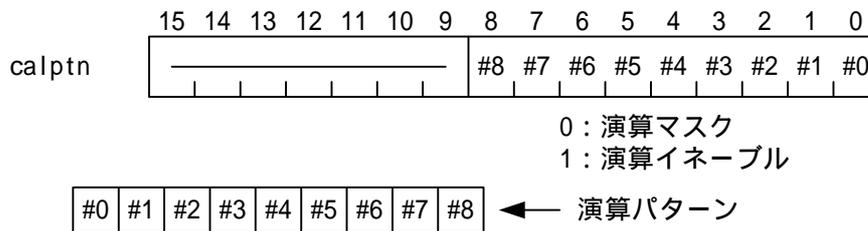
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ライン局所最小値フィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした  $1 \times 9$  近傍の局所領域で与えられた演算パターンでマスク処理し、有効となった画素の中から最小の画素を出力します。周辺では不定となります。



パラメータcalptnのフォーマットを下記に示します。



## IP\_LineMaxFLT

## ライン局所最大値フィルタ

### 機能

ソース画面に対し、パラメータで指定されたパターンで最大濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_LineMaxFLT(
    DEVID  devID ,
    int    ImgSrc ,
    int    ImgDst ,
    int    calptn
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
演算パターン

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面の画面番号

#### ImgDst

デスティネーション画面の画面番号

#### calptn

演算パターン（下位9ビット有効）。詳細については、IP\_LineMinFLT()コマンドの詳細情報を参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

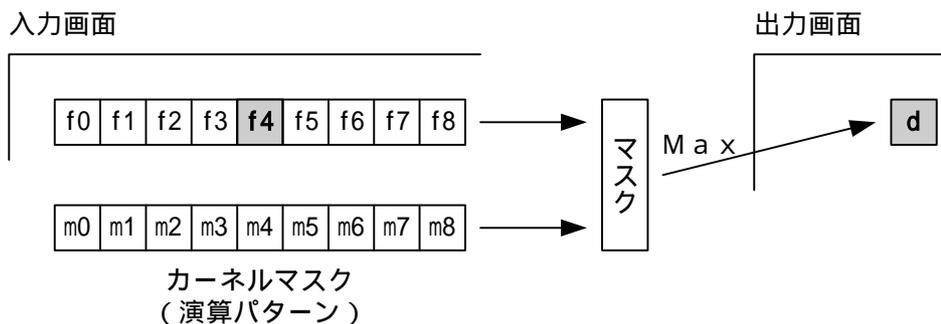
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

ライン局所最大値フィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした  $1 \times 9$  近傍の局所領域で与えられた演算パターンでマスク処理し、有効となった画素の中から最大の画素を出力します。周辺では不定となります。



パラメータcalptnのフォーマットは、IP\_LineMinFLTを参照して下さい。

# IP\_RankFLT

## 局所ランクフィルタ

### 機能

ソース画面に対し、パラメータで指定されたパターンで指定されたランクの濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_RankFLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      calptn ,
    int      rank
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
演算パターン  
ランク

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**calptn**

演算パターン

**rank**

ランク値

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

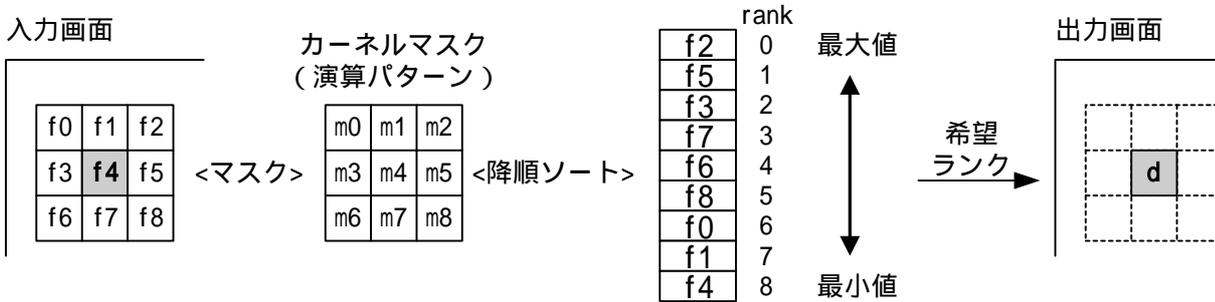
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 3	ランク指定範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

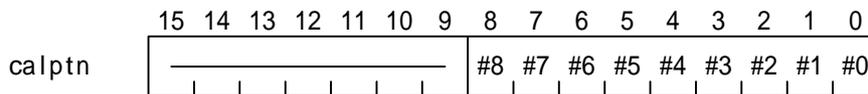
局所ランクフィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした3 × 3 近傍の局所領域を与えられた演算パターンでマスク処理し、有効となった画素の値を大きい順にソートして、希望のランクの数値を抽出し出力します。指定領域の周辺では中央画素をそのまま出力します。



注) マスクされた画素の値は、- 2 5 5 (最小値) として扱われます。従って、ランク値を指定するときは、最大値 (ランク 0) からの順番をランクによって指定して下さい。

- |          |  |
|----------|--|
| 最大値出力の場合 | 常にランク 0 (rank = 0)                                   |
| 中間値の場合   | 有効にした画素数の半分の数値<br>例: 有効画素数 7 画素であれば、ランク 3 (rank = 3) |
| 最低値の場合   | 有効にした画素数<br>例: 有効画素数 7 画素であれば、ランク 6 (rank = 6)       |

パラメータ calptn のフォーマットを下記に示します。



- 0: 演算無効 (- 2 5 5 を代入)
- 1: 演算イネーブル (処理対象)

#0	#1	#2
#3	#4	#5
#6	#7	#8

カーネルマスク (画面との対応)

# IP\_Rank4FLT

## 局所ランクフィルタ(4連結)

### 機能

ソース画面に対し、4連結で指定されたランクの濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Rank4FLT(
    devID ,
    int    imgSrc ,
    int    imgDst ,
    int    rank
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ランク

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgSrc**  
ソース画面の画面番号

**imgDst**  
デスティネーション画面の画面番号

**rank**  
ランク値

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2	ランク指定範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

4連結のマスクパターン

f0	f1	f2
f3	f4	f5
f6	f7	f8

 演算対象画素

# IP\_Rank8FLT

## 局所ランクフィルタ(8連結)

### 機能

ソース画面に対し、8連結で指定されたランクの濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Rank8FLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      rank
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ランク

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**ImgDst**  
デスティネーション画面の画面番号

**rank**  
ランク値

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 3	ランク指定範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPErrror ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

8連結のマスクパターン

f0	f1	f2
f3	f4	f5
f6	f7	f8

 演算対象画素

## IP\_MedFLT

## 局所メディアンフィルタ

### 機能

ソース画面に対し、パラメータで指定されたパターンで中間濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MedFLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      calptn
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
演算パターン

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**calptn**

演算パターン

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

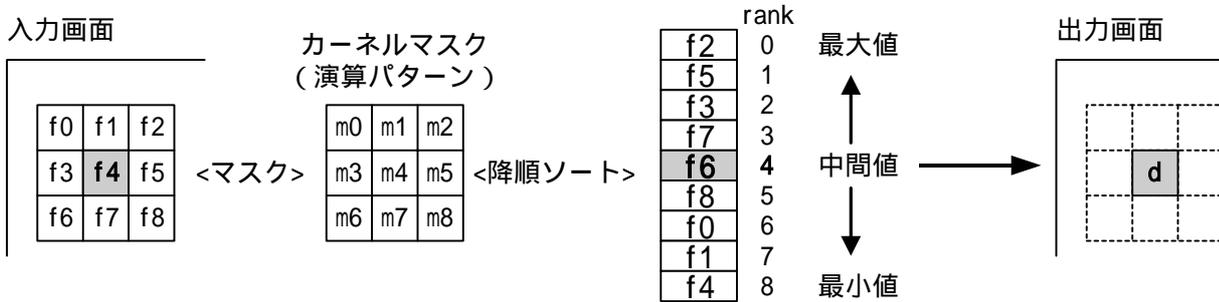
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

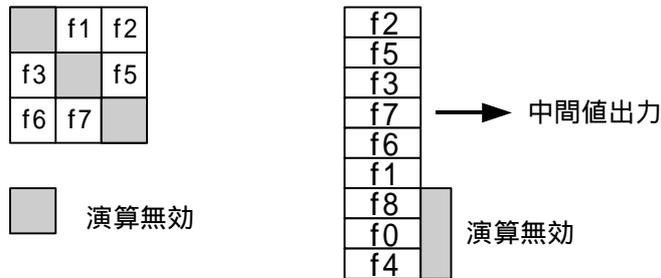
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

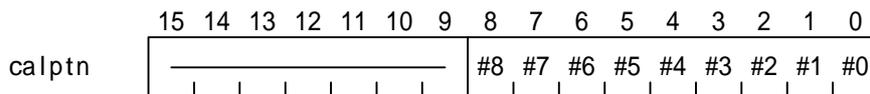
局所メディアンフィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域を与えられた演算パターンでマスク処理し、有効となった画素の値の中間値を抽出し出力します。指定領域の周辺では中央画素をそのまま出力します。



注) 局所メディアンフィルタでは、有効画素の中間値を出力します。



パラメータcalptnのフォーマットを下記に示します。



0: 演算無効 (- 2 5 5 を代入)  
1: 演算イネーブル (処理対象)



カーネルマスク  
(画面との対応)

# IP\_Med4FLT

## 局所メディアンフィルタ(4連結)

### 機能

ソース画面に対し、4連結で中間濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Med4FLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

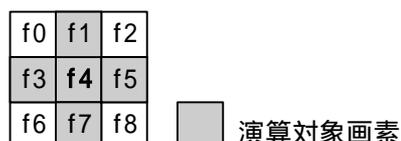
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10 , 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPErr) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

4連結のマスクパターン



# IP\_Med8FLT

## 局所メディアンフィルタ(8連結)

### 機能

ソース画面に対し、8連結で中間濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Med8FLT(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

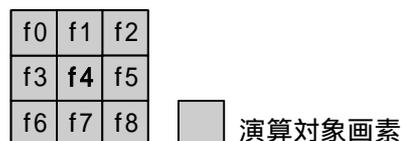
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10 , 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

8連結のマスクパターン



## IP\_Label4

## ラベリング(4連結)

### 機能

ソース画面（2値）に対して4連結ラベリング処理を行い、デスティネーション画面（濃淡）に出力します。ラベル付けされる番号は、1～255です。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Label4(
    DEVID
    int
    int
    enum IP_Label_opt
)
```

```
devID ,
ImgSrc ,
ImgDst ,
opt
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
ラベリングオプション
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

opt

ラベリングオプション

ラベリングオプション	対応定数	内容
LABEL_OBJ	0	白（画素1）を物体としてラベリング
LABEL_BKG	1	黒（画素0）を物体としてラベリング

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 255	正常終了。ラベル数。
- 1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外（不当画面番号エラー）
2 2	オプション設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ラベル付けの最大数は255個です。ラベルオーバーフローは、ラベル数が256個以上、仮ラベル数が1023個以上、合流対が1023以上のいずれかで発生します。

## IP\_Label8

## ラベリング(8連結)

### 機能

ソース画面（2値）に対して8連結ラベリング処理を行い、デスティネーション画面（濃淡）に出力します。ラベル付けされる番号は、1～255です。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Label8(
    DEVID
    int
    int
    enum IP_Label_opt
)
```

```
devID ,
ImgSrc ,
ImgDst ;
opt
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
ラベリングオプション
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

opt

ラベリングオプション

ラベリングオプション	対応定数	内容
LABEL_OBJ	0	白（画素1）を物体としてラベリング
LABEL_BKG	1	黒（画素0）を物体としてラベリング

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 255	正常終了。ラベル数。
- 1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外（不当画面番号エラー）
2 2	オプション設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ラベル付けの最大数は255個です。ラベルオーバーフローは、ラベル数が256個以上、仮ラベル数が1023個以上、合流対が1023以上のいずれかで発生します。

## IP\_Label4withAreaFLT

## 面積フィルタ付きラベリング(4連結)

### 機能

ソース画面（2値）に対して4連結ラベリング処理を行い、面積フィルタ処理後、デスティネーション画面（濃淡）に出力します。ラベル付けされる番号は、1～255です。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Label4withAreaFLT(
    DEVID
    int
    int
    long
    long
    enum IP_Label_opt
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thrmin ,
thrmax ,
opt
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
面積フィルタしきい値
面積フィルタしきい値
ラベリングオプション
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**thrmin**

面積フィルタしきい値（最小値）

**thrmax**

面積フィルタしきい値（最大値）

**opt**

ラベリングオプション

ラベリングオプション	対応定数	内容
LABEL_OBJ	0	白（画素1）を物体としてラベリング
LABEL_BKG	1	黒（画素0）を物体としてラベリング

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 255	正常終了。ラベル数。
- 1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2 , 2 3	面積フィルタしきい値範囲設定エラー
2 4	オプション設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

面積フィルタリングでは、thrmin以上thrmax以下の面積の物体にラベル付けを行います。

ラベル付けの最大数は255個です。ラベルオーバーフローは、ラベル数が256個以上、仮ラベル数が1023個以上、合流対が1023以上のいずれかで発生します。

## IP\_Label8withAreaFLT

## 面積フィルタ付きラベリング(8連結)

### 機能

ソース画面（2値）に対して8連結ラベリング処理を行い、面積フィルタ処理後、デスティネーション画面（濃淡）に出力します。ラベル付けされる番号は、1～255です。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Label8withAreaFLT(
    DEVID
    int
    int
    long
    long
    enum IP_Label_opt
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thrmin ,
thrmax ,
opt
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
面積フィルタしきい値
面積フィルタしきい値
ラベリングオプション
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**thrmin**

面積フィルタしきい値（最小値）

**thrmax**

面積フィルタしきい値（最大値）

**opt**

ラベリングオプション

ラベリングオプション	対応定数	内容
LABEL_OBJ	0	白（画素1）を物体としてラベリング
LABEL_BKG	1	黒（画素0）を物体としてラベリング

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 255	正常終了。ラベル数。
- 1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2 , 2 3	面積フィルタしきい値範囲設定エラー
2 4	オプション設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

面積フィルタリングでは、thrmin以上thrmax以下の面積の物体にラベル付けを行います。

ラベル付けの最大数は255個です。ラベルオーバーフローは、ラベル数が256個以上、仮ラベル数が1023個以上、合流対が1023以上のいずれかで発生します。

# IP\_Label4withAreaFLTSort

## 面積フィルタ付きラベリング (面積ソート4連結)

### 機能

ソース画面（2値）に対して4連結ラベリング処理を行い、面積フィルタ処理後、面積によるラベルのソートを行いデスティネーション画面（濃淡）に出力します。ラベル付けされる番号は、1～255です。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Label4withAreaFLTSort(
    DEVID
    int
    int
    long
    long
    enum IP_Label_opt
    enum IP_Label_Sort_opt
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thrmin ,
thrmax ,
opt ,
opt2
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
面積フィルタしきい値
面積フィルタしきい値
ラベリングオプション
ラベリングソートオプション
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面の画面番号

#### ImgDst

デスティネーション画面の画面番号

#### thrmin

面積フィルタしきい値（最小値）

#### thrmax

面積フィルタしきい値（最大値）

#### opt

ラベリングオプション

ラベリングオプション	対応定数	内容
LABEL_OBJ	0	白（画素1）を物体としてラベリング
LABEL_BKG	1	黒（画素0）を物体としてラベリング

#### opt2

ソートオプション

ラベリングソートオプション	対応定数	内容
LABEL_SORT_MAX	0	面積でのソートを大きい順に行う
LABEL_SORT_MIN	1	面積でのソートを小さい順に行う

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 255	正常終了。ラベル数。
- 1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2 , 2 3	面積フィルタしきい値範囲設定エラー
2 4、2 5	オプション設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPErrror ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

面積フィルタリングでは、thrmin以上thrmax以下の面積の物体にラベル付けを行います。

ラベル付けの最大数は255個です。ラベルオーバーフローは、ラベル数が256個以上、仮ラベル数が1023個以上、合流対が1023以上のいずれかで発生します。

# IP\_Label8withAreaFLSort

## 面積フィルタ付きラベリング (面積ソート8連結)

### 機能

ソース画面（2値）に対して8連結ラベリング処理を行い、面積フィルタ処理後、面積によるラベルのソートを行いデスティネーション画面（濃淡）に出力します。ラベル付けされる番号は、1～255です。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Label8withAreaFLSort(
    DEVID
    int
    int
    long
    long
    enum IP_Label_opt
    enum IP_Label_Sort_opt
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thrmin ,
thrmax ,
opt ,
opt2
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
面積フィルタしきい値
面積フィルタしきい値
ラベリングオプション
ラベリングソートオプション
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**thrmin**

面積フィルタしきい値（最小値）

**thrmax**

面積フィルタしきい値（最大値）

**opt**

ラベリングオプション

ラベリングオプション	対応定数	内容
LABEL_OBJ	0	白（画素1）を物体としてラベリング
LABEL_BKG	1	黒（画素0）を物体としてラベリング

**opt2**

ソートオプション

ラベリングソートオプション	対応定数	内容
LABEL_SORT_MAX	0	面積でのソートを大きい順に行う
LABEL_SORT_MIN	1	面積でのソートを小さい順に行う

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 255	正常終了。ラベル数。
- 1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2 , 2 3	面積フィルタしきい値範囲設定エラー
2 4、2 5	オプション設定値範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPErrror ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

面積フィルタリングでは、thrmin以上thrmax以下の面積の物体にラベル付けを行います。

ラベル付けの最大数は255個です。ラベルオーバーフローは、ラベル数が256個以上、仮ラベル数が1023個以上、合流対が1023以上のいずれかで発生します。

## IP\_ExtractLORegionX

ラベル毎領域X座標抽出  
(Min & Max X座標)

### 機能

ラベル毎に X 座標の最小値 (MinX) と最大値 (MaxX) を抽出します。

### コーディング

```
int
IP_ExtractLORegionX(
    DEVID    devID ,
    int      ImgSrc ,
    int      *TbIMinX ,
    int      *TbIMaxX
)
```

### コメント

デバイス I D  
 ソース画面番号  
 Min X 座標格納テーブルへのポインタ  
 Max X 座標格納テーブルへのポインタ

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

ImgSrc

ソース画面の画面番号。ラベリング後のデスティネーション画面を指定して下さい。

\*TbIMinX

Min X 座標格納テーブルへのポインタ。テーブルは 4 バイト × 2 5 6 の容量を確保して下さい。

\*TbIMaxX

Max X 座標格納テーブルへのポインタ。テーブルは 4 バイト × 2 5 6 の容量を確保して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

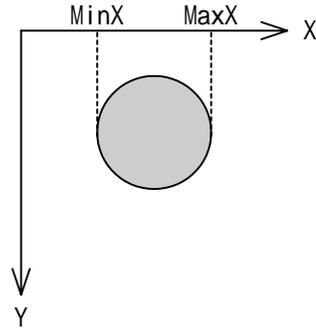
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報



TbIMinX、TbIMaxXのフォーマットを下記に示します。  
 ラベルが存在しない場合には、MinX座標に32767 (0x7FFF) を、MaxX座標に0を出力します。

		TbIMinX		TbIMaxX	
		31	0	31	0
ラベル		不定		不定	
1	→ 1	物体 1 座標		物体 1 座標	
⋮	⋮	⋮		⋮	
⋮	⋮	⋮		⋮	
⋮	⋮	⋮		⋮	
⋮	⋮	⋮		⋮	
⋮	⋮	⋮		⋮	
⋮	⋮	⋮		⋮	
2 5 3	→ 2 5 3	物体 2 5 3 座標		物体 2 5 3 座標	
2 5 4	→ 2 5 4	物体 2 5 4 座標		物体 2 5 4 座標	
2 5 5	→ 2 5 5	物体 2 5 5 座標		物体 2 5 5 座標	

ラベル個数よりも大きい部分は不定です

本コマンドで得られるMinX、MaxXの座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。  
 画面上の座標位置はそれぞれ

$\text{MinX} = \text{TbIMinX}[\text{ラベル番号}] + \text{sx}$ $\text{MaxX} = \text{TbIMaxX}[\text{ラベル番号}] + \text{sx}$	MinX : MinX座標 MaxX : MaxX座標 sx : SRC0_WIN(sx)
---	---

となります。

## IP\_ExtractLORegionY

ラベル毎領域Y座標抽出  
(Min & Max Y座標)

### 機能

ラベル毎にY座標の最小値(MinY)と最大値(MaxY)を抽出します。

### コーディング

```
int
IP_ExtractLORegionY(
    DEVID    devid ,
    int      ImgSrc ,
    int      *TbIMinY ,
    int      *TbIMaxY
)
```

### コメント

デバイスID  
ソース画面番号  
Min Y座標格納テーブルへのポインタ  
Max Y座標格納テーブルへのポインタ

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号。ラベリング後のデスティネーション画面を指定して下さい。

**\*TbIMinY**

Min Y座標格納テーブルへのポインタ。テーブルは4バイト×256の容量を確保して下さい。

**\*TbIMaxY**

Max Y座標格納テーブルへのポインタ。テーブルは4バイト×256の容量を確保して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

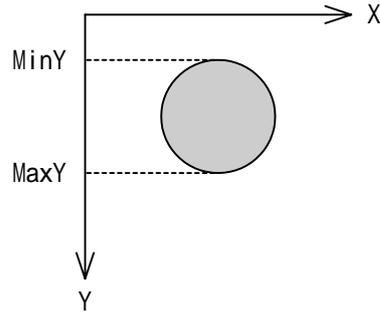
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**



TbIMinY、TbIMaxYのフォーマットを下記に示します。  
 ラベルが存在しない場合には、MinY座標に32767 (0x7FFF) を、MaxY座標に0を出力します。

		TbIMinY		TbIMaxY	
		31	0	31	0
ラベル		0	不定	0	不定
1	→	1	物体 1 座標	1	物体 1 座標
⋮		⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮
⋮		⋮	⋮	⋮	⋮
2 5 3	→	2 5 3	物体 2 5 3 座標	2 5 3	物体 2 5 3 座標
2 5 4	→	2 5 4	物体 2 5 4 座標	2 5 4	物体 2 5 4 座標
2 5 5	→	2 5 5	物体 2 5 5 座標	2 5 5	物体 2 5 5 座標

ラベル個数よりも大きい  
部分は不定です

本コマンドで得られるMinY、MaxYの座標はSRCO\_WINで設定されたウィンドウからの相対座標になります。  
 画面上の座標位置はそれぞれ

$\text{MinY} = \text{TbIMinY}[\text{ラベル番号}] + \text{sy}$ $\text{MaxY} = \text{TbIMaxY}[\text{ラベル番号}] + \text{sy}$	MinY : MinY座標 MaxY : MaxY座標 sy : SRCO_WIN(sy)
---	---

となります。

## IP\_ExtractLOArea

## ラベル毎面積抽出

### 機能

ソース画面に対しヒストグラム処理を行い、ラベル毎面積の抽出を行います。

### コーディング

```
int
IP_ExtractLOArea(
    DEVID    devID ,
    int      ImgSrc ,
    int      *Tbl
)
```

### コメント

デバイスID  
ソース画面番号  
ラベル面積データ格納テーブルへのポインタ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号。ラベリング後のデスティネーション画面を指定して下さい。

\*Tbl

ラベル面積データ格納テーブルへのポインタ。テーブルは4バイト×256の容量を確保して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

Tblのフォーマットを下記に示します。

ラベル		Tbl	
		31	0
			不定
1	→	1	物体 1 面積
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
2 5 3	→	2 5 3	物体 2 5 3 面積
2 5 4	→	2 5 4	物体 2 5 4 面積
2 5 5	→	2 5 5	物体 2 5 5 面積

ラベル個数よりも大きい部分は不定です

# IP\_ExtractLOGravity

## ラベル毎重心座標抽出

### 機能

ラベル画面に対し、ラベル毎の重心座標とラベル毎面積を抽出し、テーブルに格納します。

### コーディング

```
int
IP_ExtractLOGravity(
    DEVID          devID ,
    int            ImgSrc ,
    IPLLOGravityTbl *Tbl
)
```

### コメント

デバイスID  
ソース画面番号  
ラベル毎重心、面積格納テーブル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号。ラベリング後のデスティネーション画面を指定して下さい。

\*Tbl

ラベル毎の重心座標とラベル毎の面積を格納するテーブル。  
テーブルは、sizeof(IPLLOGravityTbl) × 256の容量を確保して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 255	正常終了(ラベル数)
- 1	異常終了

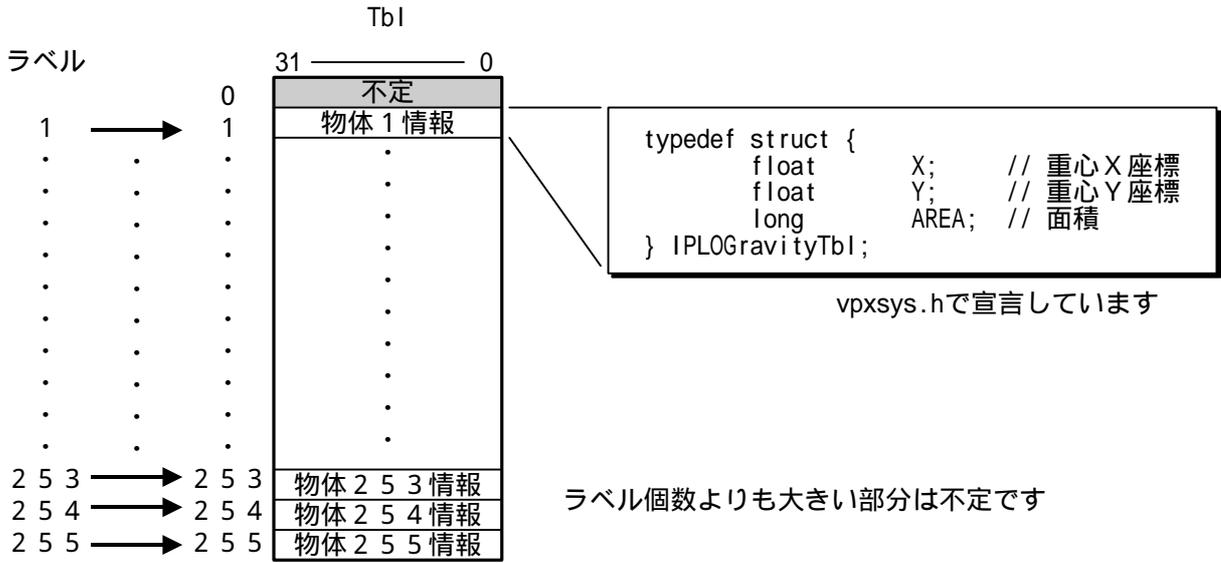
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外(不当画面番号エラー)
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

IPLOGravityTblテーブルのフォーマットを、下記に示します。



本コマンドで得られる重心 X , Y の座標はSRCO\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

$GX = Tbl[ \text{ラベル番号} ].X + sx$ $GY = Tbl[ \text{ラベル番号} ].Y + sy$	GX : 重心 X 座標 GY : 重心 Y 座標 sx : SRCO_WIN(sx) sy : SRCO_WIN(sy)
--	--

となります。

# IP\_ExtractGOFeatures

## 濃淡画像基本特徴量抽出

### 機能

ソース画面（濃淡）に対し、基本特徴量抽出を行います。

### コーディング

```
int
IP_ExtractGOFeatures(
    DEVID
    int
    IPGOFeatureTbl
    enum IPGOFeatureOpt
)
```

```
devID ,
ImgSrc ,
*RegTbl ,
opt
```

### コメント

デバイスID  
ソース画面番号  
基本特徴量テーブル  
濃淡画像基本特徴量抽出オプション

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*RegTbl**

基本特徴量を格納する領域へのポインタ

**opt**

リザーブ。0を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

IPGFeatureTblのフォーマット及び内容を、下記に示します。

また、濃度累積値 (ACC\_LEVEL) が 0 以外のときのみ、テーブル内データが有効です。

ここで、最小 / 最大濃度値をとる点とはラスタスキャンした際に最初に出現した最小 / 最大濃度値をとる点を指します。

vpxsys.hで宣言しています

```
typedef struct {
    int    MIN_LEVEL_X;
    int    MIN_LEVEL_Y;
    int    MAX_LEVEL_X;
    int    MAX_LEVEL_Y;
    long   ACC_LEVEL;
    int    MIN_LEVEL;
    int    MAX_LEVEL;
    int    TYPICAL_LEVEL;
} IPGFeatureTbl;
```

基本特徴テーブル	説明
MIN_LEVEL_X	最小濃度値をとる点のX座標
MIN_LEVEL_Y	最小濃度値をとる点のY座標
MAX_LEVEL_X	最大濃度値をとる点のX座標
MAX_LEVEL_Y	最大濃度値をとる点のY座標
ACC_LEVEL	濃度累積
MIN_LEVEL	最小濃度値
MAX_LEVEL	最大濃度値
TYPICAL_LEVEL	予備

# IP\_Histogram

## 濃度ヒストグラム

### 機能

ソース画面に対しヒストグラム処理を行い、濃度頻度累積及び濃度画像基本特徴量抽出を行います。

### コーディング

```
int
IP_Histogram(
    DEVID
    int
    long
    IPG0FeatureTbl
    int
)
```

```
devID ,
ImgSrc ,
*Tbl ,
*RegTbl ,
opt
```

### コメント

デバイスID  
ソース画面番号  
濃度頻度分布テーブル  
基本特徴量テーブル  
オプション

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*Tbl**

濃度頻度分布を格納する領域へのポインタ。4バイト×256の容量を確保して下さい。

**\*RegTbl**

基本特徴量を格納する領域へのポインタ

**opt**

オプション。0を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

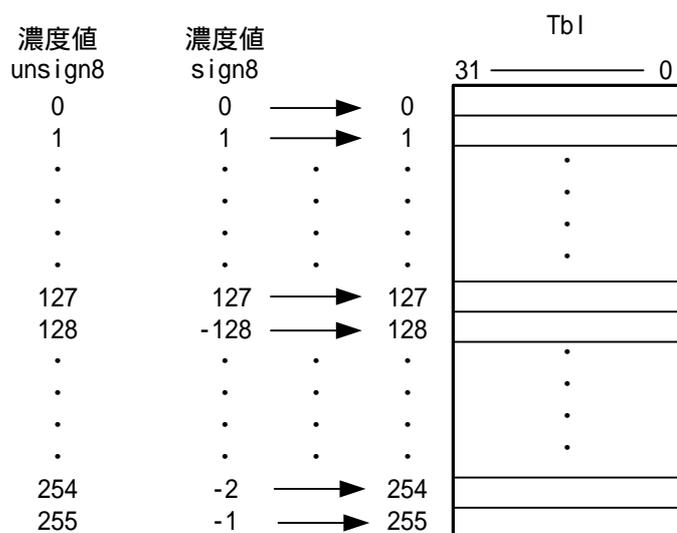
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

Tblのフォーマットを下記に示します。また、IPG0FeatureTblのフォーマット及び内容は、IP\_ExtractG0Features()の詳細情報の欄を参照して下さい。



# IP\_ProjectGO

## X/Y軸への濃度累積値投影 (16ビット)

### 機能

濃淡画像でのX/Y軸への投影を行い、濃度累積(16ビット)及び濃淡画像基本特徴量抽出を行います。本コマンドは、X/Y軸への投影を同時に行いますが、濃度累積値が16ビット分までしか積算出来ませんので注意して下さい。

### コーディング

```
int
IP_ProjectGO(
    DEVID
    int
    int
    int
    IPGOFeatureTbl
)
```

```
devID ,
ImgSrc ,
*TbIX ,
*TbIY ,
*RegTbl
```

### コメント

```
デバイスID
ソース画面番号
濃度累積値テーブル
濃度累積値テーブル
基本特徴量テーブル
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*TbIX**

濃度累積値(X軸への投影)を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲でX/Y軸で長い方に合わせて下さい。

**\*TbIY**

濃度累積値(Y軸への投影)を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲でX/Y軸で長い方に合わせて下さい。

**\*RegTbl**

基本特徴量を格納する領域へのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

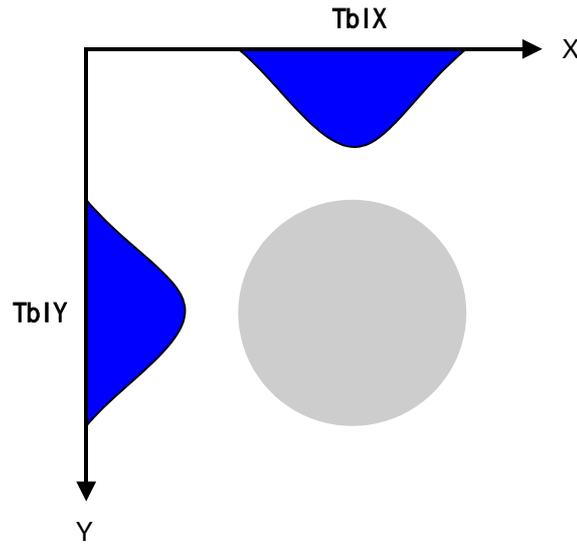
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外(不当画面番号エラー)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報



TbIX、TbIYのフォーマットを下記に示します。また、IPG0FeatureTblのフォーマット及び内容は、IP\_ExtractGOFeatures()の詳細情報の欄を参照して下さい。

TbIX			TbIY				
X 座標		31 ————— 0	Y 座標		31 ————— 0		
0	→	0	濃度累積値	0	→	0	濃度累積値
1	→	1	濃度累積値	1	→	1	濃度累積値
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
1022	→	1022	濃度累積値	1022	→	1022	濃度累積値
1023	→	1023	濃度累積値	1023	→	1023	濃度累積値

書き込まれるテーブルのサイズ(範囲)はSRC0\_WINで指定した範囲になります。X方向、Y方向の大きい方の範囲で書き込まれるサイズが決まります。

本コマンドで得られるTbIX、TbIYの投影値はSRC0\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

$$prjX = TbIX[ X - sx ]$$

$$prjY = TbIY[ Y - sy ]$$

- prjX : X軸投影値
- prjY : Y軸投影値
- X : 画面X座標
- Y : 画面Y座標
- sx : SRC0\_WIN(sx)
- sy : SRC0\_WIN(sy)

となります。

## IP\_ProjectG0onX

## X軸への濃度累積値投影

## 機能

濃淡画像でのX軸への投影を行い、濃度累積及び濃淡画像基本特徴量抽出を行います。

## コーディング

```
int
IP_ProjectG0onX(
    DEVID          devID ,
    int            ImgSrc ,
    long          *Tbl ,
    IPG0FeatTbl   *RegTbl
)
```

## コメント

デバイスID  
ソース画面番号  
濃度累積値テーブル  
基本特徴量テーブル

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面の画面番号

## \*Tbl

濃度累積値（X方向への投影）を格納する領域へのポインタ。テーブルサイズは、SRC0\_WINで指定したX方向の範囲になります。

## \*RegTbl

基本特徴量を格納する領域へのポインタ

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

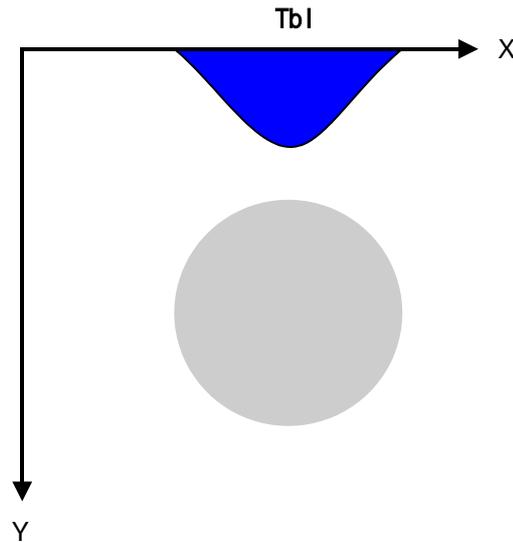
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報



Tblのフォーマットを下記に示します。また、IPG0FeatureTblのフォーマット及び内容は、IP\_ExtractG0Features()の詳細情報の欄を参照して下さい。

X 座標		Tbl	
		31	0
0	→	0	濃度累積値
1	→	1	濃度累積値
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
1022	→	1022	濃度累積値
1023	→	1023	濃度累積値

書き込まれるテーブルのサイズ（範囲）はSRCO\_WINで指定したX方向の範囲になります。  
 本コマンドで得られるTblの投影値はSRCO\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

$$prjX = Tbl[ X - sx ]$$

- prjX : X軸投影値
- X : 画面X座標
- sx : SRCO\_WIN(sx)

となります。

# IP\_ProjectG0onY

## Y軸への濃度累積値投影

### 機能

濃淡画像でのY軸への投影を行い、濃度累積及び濃淡画像基本特徴量抽出を行います。

### コーディング

```
int
IP_ProjectG0onY(
    DEVID          devID ,
    int            ImgSrc ,
    long          *Tbl ,
    IPG0FeatureTbl *RegTbl
)
```

### コメント

デバイスID  
ソース画面番号  
濃度累積値テーブル  
基本特徴量テーブル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

\*Tbl

濃度累積値（Y方向への投影）を格納する領域へのポインタ。テーブルサイズは、SRC0\_WINで指定したY方向の範囲になります。

\*RegTbl

基本特徴量を格納する領域へのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

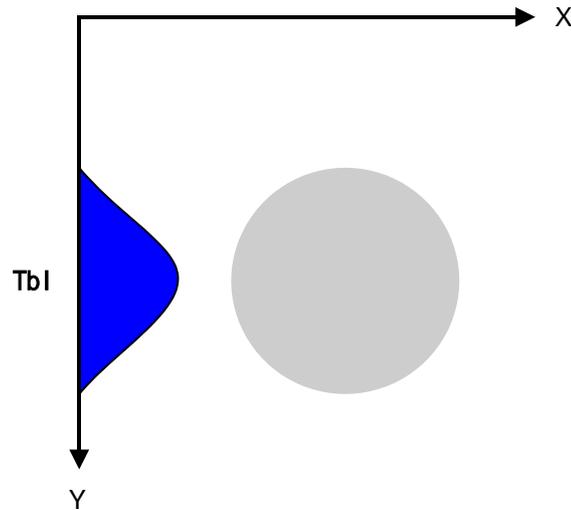
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報



TbIのフォーマットを下記に示します。また、IPG0FeatureTbIのフォーマット及び内容は、IP\_ExtractG0Features()の詳細情報の欄を参照して下さい。

		TbI	
Y 座標		31	0
0	→	0	濃度累積値
1	→	1	濃度累積値
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
1022	→	1022	濃度累積値
1023	→	1023	濃度累積値

書き込まれるテーブルのサイズ（範囲）はSRCO\_WINで指定したY方向の範囲になります。  
本コマンドで得られるTbIの投影値はSRCO\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

$$\text{prjY} = \text{TbI}[Y - \text{sy}]$$

prjY : Y軸投影値  
Y : 画面Y座標  
sy : SRCO\_WIN(sy)

となります。

# IP\_ProjectGOMaxValue

## X / Y軸への最大濃度値投影

### 機能

濃淡画像での X / Y 軸への最大濃度値投影を行い、その結果と濃淡画像基本特徴量抽出を行います。

### コーディング

```
int
IP_ProjectGOMaxValue(
    DEVID
    int
    int
    int
    IPGOFeatureTbl
)
```

```
devID ,
ImgSrc ,
*TbIX ,
*TbIY ,
*RegTbl
```

### コメント

デバイス I D  
ソース画面番号  
最大濃度値 ( X 軸への投影 ) テーブル  
最大濃度値 ( Y 軸への投影 ) テーブル  
基本特徴量テーブル

### パラメータ

**devID**

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*TbIX**

最大濃度値 ( X 軸への投影 ) を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲で X / Y 軸で長い方に合わせて下さい。

**\*TbIY**

最大濃度値 ( Y 軸への投影 ) を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲で X / Y 軸で長い方に合わせて下さい。

**\*RegTbl**

基本特徴量を格納する領域へのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

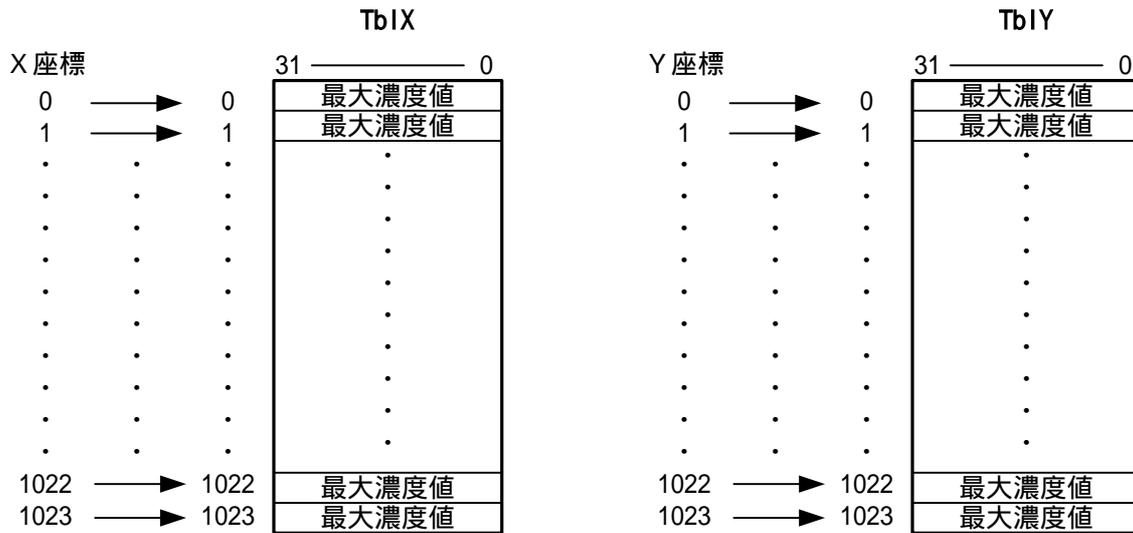
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

TbIX、TbIYのフォーマットを下記に示します。また、IPG0FeatureTblのフォーマット及び内容は、IP\_ExtractG0Features()の詳細情報の欄を参照して下さい。



書き込まれるテーブルのサイズ（範囲）はSRCO\_WINで指定した範囲になります。X方向、Y方向の大きい方の範囲で書き込まれるサイズが決まります。

本コマンドで得られるTbIX、TbIYの最大濃度値はSRCO\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

$\text{prjX} = \text{TbIX}[ X - \text{sx} ]$ $\text{prjY} = \text{TbIY}[ Y - \text{sy} ]$
---

prjX : X軸最大濃度値  
 prjY : Y軸最大濃度値  
 X : 画面X座標  
 Y : 画面Y座標  
 sx : SRCO\_WIN(sx)  
 sy : SRCO\_WIN(sy)

となります。

# IP\_ProjectGOMinValue

## X / Y軸への最小濃度値投影

### 機能

濃淡画像での X / Y 軸への最小濃度値投影を行い、その結果と濃淡画像基本特徴量抽出を行います。

### コーディング

```
int
IP_ProjectGOMinValue(
    DEVID
    int
    int
    int
    IPCOFeatureTbl
)
```

```
devID ,
ImgSrc ,
*TbIX ,
*TbIY ,
*RegTbl
```

### コメント

デバイス I D  
ソース画面番号  
最小濃度値 ( X 軸への投影 ) テーブル  
最小濃度値 ( Y 軸への投影 ) テーブル  
基本特徴量テーブル

### パラメータ

**devID**

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*TbIX**

最小濃度値 ( X 軸への投影 ) を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲で X / Y 軸で長い方に合わせて下さい。

**\*TbIY**

最小濃度値 ( Y 軸への投影 ) を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲で X / Y 軸で長い方に合わせて下さい。

**\*RegTbl**

基本特徴量を格納する領域へのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。



# IP\_HistogramShort

## 濃度ヒストグラム(16ビット)

### 機能

ソース画面に対しヒストグラム処理を行い、濃度頻度分布および濃淡画像基本特徴量抽出を行います。

### コーディング

```
int
IP_HistogramShort(
    DEVID
    int
    int
    IPGOFeatureTbl
    int
)
```

```
devID ,
ImgSrc ,
*Tbl ,
*FtrTbl,
opt
```

### コメント

```
デバイスID
ソース画面番号
濃度頻度累積データを格納するアドレス
基本特徴量を格納するアドレス
リザーブ
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面番号

**\*Tbl**

濃度頻度累積データを格納するアドレス(256個以上確保して下さい)

**\*FtrTbl**

基本特徴量を格納するアドレス

**opt**

リザーブ。0を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

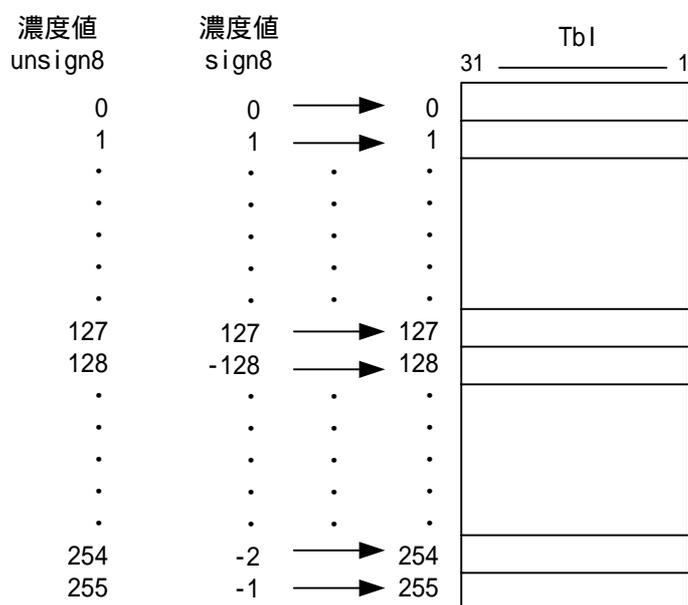
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外(不当画面番号エラー)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

Tbl のフォーマットを、以下に示します。また、FtrTbl のフォーマットおよび内容は、IP\_ExtractG0Features()の備考を参照してください。



(注) 有効ビット幅は16ビットです。

# IP\_ProjectBlockG0

## 領域毎濃淡画像濃度累積

### 機能

ソース画像(濃淡)の画面領域を設定したパラメータで分割し、各領域毎の濃度累積値及び濃淡画像基本特徴量抽出を行います。

### コーディング

```
int
IP_ProjectBlockG0(
    DEVID
    int
    long
    IPG0FeatureTbl
    IPDivideTbl
)
```

```
devID ,
ImgSrc ,
*Tbl ,
*FtrTbl ,
*DivideTbl
```

### コメント

```
デバイスID
ソース画面番号
領域毎濃度累積値データを格納するアドレス
基本特徴量を格納するアドレス
領域分割数を格納するアドレス
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面番号

#### \*Tbl

領域毎濃度累積値データを格納するアドレス

注) テーブルサイズは領域分割数(X軸分割数×Y軸分割数)用意してください。(最大768)

#### \*FtrTbl

基本特徴量を格納するアドレス

#### \*DivideTbl

領域分割数を格納するアドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外(不当画面番号エラー)
2 3	領域分割数が不正
4 9	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

画面分割数と処理領域の間に以下の関係が成立しないと特徴量抽出で誤差が発生します。

- 条件(1) (画面分割数) \* 1024 / (処理領域の長さ) が割り切れること
- (2) (処理領域の長さ) / (画面分割数) が割り切れること

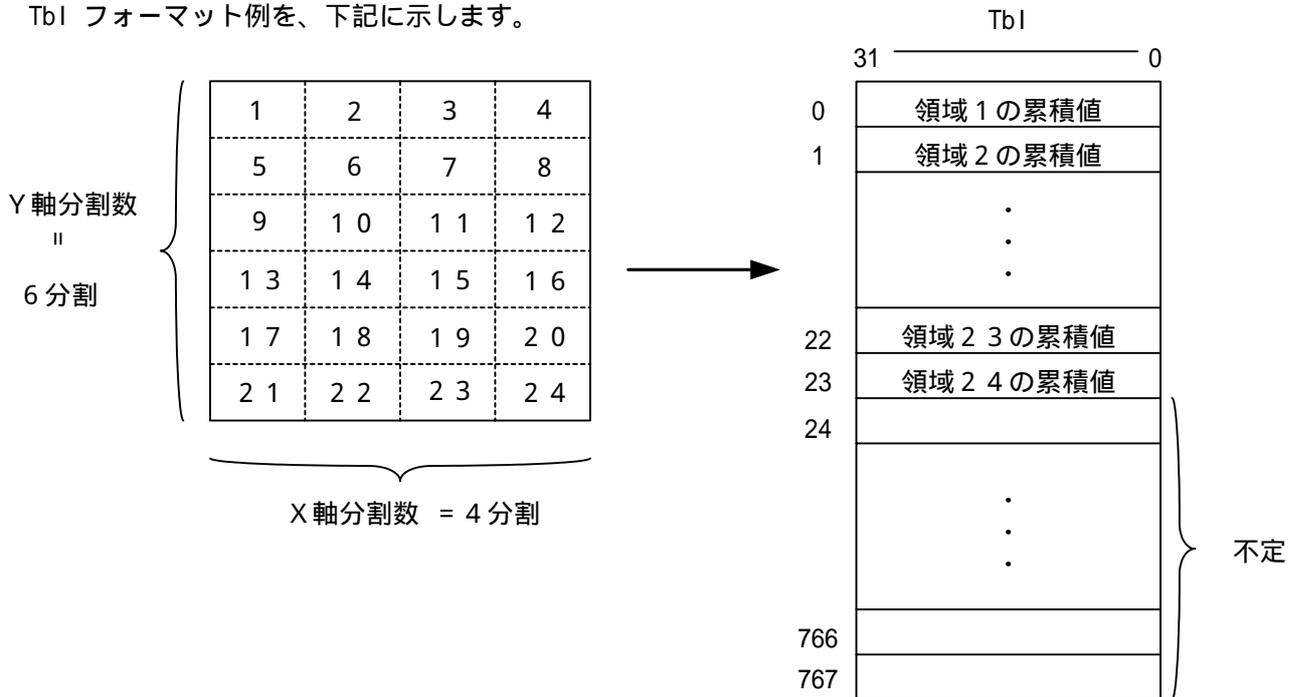
つまり、(画面分割数)と(処理領域の長さ)が2のN乗以外は、特徴量抽出に誤差が発生します。これは、IP\_ProjectBlockB0()及びIP\_ProjectBlockG0()を行う場合のヒストグラムメモリ(HM)への画像処理LSI内部のアドレス発生回路が固定小数点演算で行っており、上記(1)、(2)の条件を満たさないとHMのアドレスに誤差が生じる為です。なお、IP\_ProjectBlockB0()、IP\_ProjectBlockGOMinMaxValue()、及びIP\_ProjectBlockG0()以外では、このアドレッシングを使用しないので、他のコマンドには影響ありません。

X、Y軸の分割数はウィンドウの画素数を越えて設定できません。また以下の制約がありますのでご注意ください。

X 分割数	Y 分割数
1	1 ~ 7 6 8
2	1 ~ 3 8 4
3, 4	1 ~ 1 9 2
5 ~ 8	1 ~ 9 6
9 ~ 1 6	1 ~ 4 8

X 分割数	Y 分割数
1 7 ~ 3 2	1 ~ 2 4
3 3 ~ 6 4	1 ~ 1 2
6 5 ~ 1 2 8	1 ~ 6
1 2 9 ~ 2 5 6	1 ~ 3
2 5 7 ~ 7 6 8	1

Tbl フォーマット例を、下記に示します。



DivideTbl のフォーマットを下記に示します。

領域分割数設定テーブル

```

typedef struct {
    int Divide_X;      // 領域毎抽出 X 方向分割数
    int Divide_Y;      // 領域毎抽出 Y 方向分割数
} IPDivideTbl;
    
```

本テーブルはipxsys.h で宣言しています。

# IP\_ProjectBlockGOMinMaxValue

## 領域毎濃淡画像最大/最小濃度抽出

### 機能

ソース画像（濃淡）の画面領域を設定したパラメータで分割し、各領域毎の最小 / 最大濃度値抽出を行い、結果をユーザのテーブルへ返します。

### コーディング

```
int
IP_ProjectBlockGOMinMaxValue(
    DEVID          devID ,
    int            ImgSrc ,
    IPGOMinMaxTbl *Tbl ,
    IPDivideTbl    *DivideTbl
)
```

### コメント

デバイスID  
ソース画面番号  
累積値データを格納するアドレス  
領域分割数を格納するアドレス

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面番号

**\*Tbl**  
領域毎最小 / 最大累積値データを格納するアドレス  
注) テーブルサイズは領域分割数 ( X軸分割数 × Y軸分割数 ) 用意してください。(最大768 )

**\*DivideTbl**  
領域分割数を格納するアドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2	領域分割数が不正
4 9	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画面分割数と処理領域の間に以下の関係が成立しないと特徴量抽出で誤差が発生します。

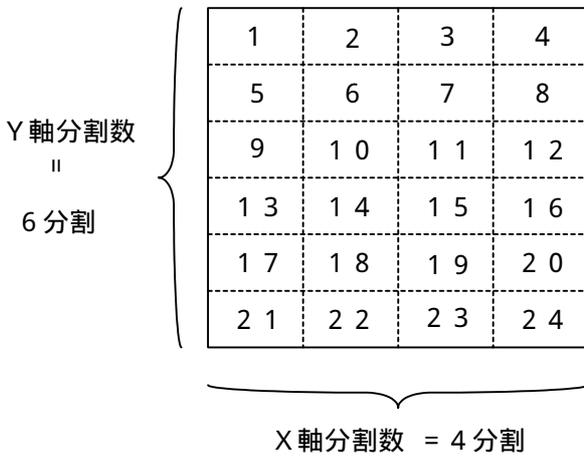
- 条件(1) (画面分割数) \* 1024 / (処理領域の長さ) が割り切れること
- (2) (処理領域の長さ) / (画面分割数) が割り切れること

つまり、(画面分割数) と (処理領域の長さ) が2のN乗以外は、特徴量抽出に誤差が発生します。これは、IP\_ProjectBlockB0() 及び IP\_ProjectBlockG0() を行う場合のヒストグラムメモリ ( HM ) への画像処理 L S I 内部のアドレス発生回路が固定小数点演算で行っており、上記(1)、(2)の条件を満たさないと HM のアドレスに誤差が生じる為です。なお、IP\_ProjectBlockB0()、IP\_ProjectBlockGOMinMaxValue()、及び IP\_ProjectBlockG0() 以外では、このアドレッシングを使用しないので、他のコマンドには影響ありません。

X、Y軸の分割数はウィンドウの画素数を越えて設定できません。また以下の制約がありますのでご注意ください。

X 分割数	Y 分割数	X 分割数	Y 分割数
1	1 ~ 7 6 8	1 7 ~ 3 2	1 ~ 2 4
2	1 ~ 3 8 4	3 3 ~ 6 4	1 ~ 1 2
3, 4	1 ~ 1 9 2	6 5 ~ 1 2 8	1 ~ 6
5 ~ 8	1 ~ 9 6	1 2 9 ~ 2 5 6	1 ~ 3
9 ~ 1 6	1 ~ 4 8	2 5 7 ~ 7 6 8	1

Tbl フォーマット例を、下記に示します。

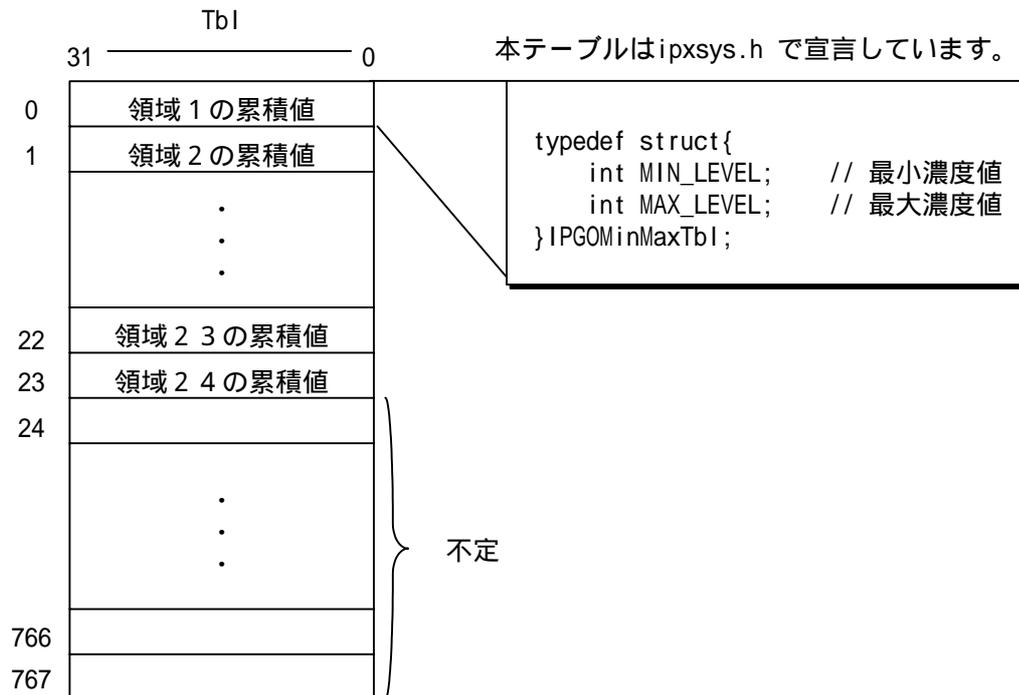


DivideTbl のフォーマットを下記に示します。

領域分割数設定テーブル

```
typedef struct {
    int Divide_X; // 領域毎抽出 X 方向分割数
    int Divide_Y; // 領域毎抽出 Y 方向分割数
} IPDivideTbl;
```

本テーブルはipxsys.h で宣言しています。



## IP\_ProjectLabelGO

## ラベル毎濃淡画像濃度累積

## 機能

ソース画面（濃淡画像）をラベリング処理したラベル画面により、ラベル毎にソース画面の濃度累積を行い、結果をユーザのテーブルへ返します。

## コーディング

```
int
IP_ProjectLabelGO(
    DEVID
    int
    int
    long
)
```

```
devID ,
ImgSrc ,
ImgLabel ,
*Tbl
```

## コメント

デバイスID  
ソース画面番号  
ラベル画面番号  
累積値データを格納するアドレス

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面番号

## ImgLabel

ラベル画面番号

## \*Tbl

領域毎最小 / 最大累積値データを格納するアドレス  
注) テーブルは必ず256確保してください。

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

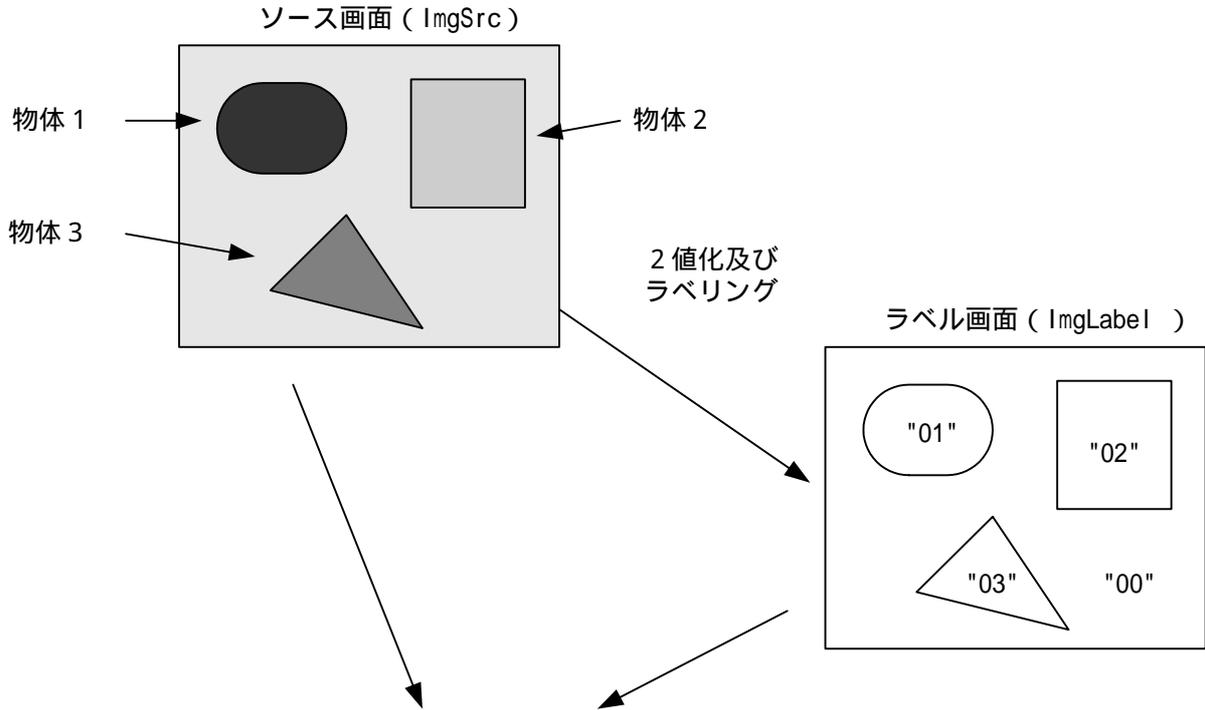
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	ソース画面空き領域、または範囲外（不当画面番号エラー）
1 1	ラベル画面空き領域、または範囲外（不当画面番号エラー）
2 2	テーブルのアドレスが不正
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPErrorTable）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

TbI の参考例を記載します。



TbI

ラベル	31	0
0	0	背景の濃度累積値
1	1	物体 1 の濃度累積値
2	2	物体 2 の濃度累積値
3	3	物体 3 の濃度累積値
4	4	0
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
254	254	0
255	255	0

ラベル画面の背景（0の部分）の濃度累積値が入ります

ラベル数を超える部分は0が入ります。

## IP\_ProjectLabelGOMinMaxValue

ラベル毎濃淡画像最小 / 最大濃度値抽出

### 機能

ソース画面(濃淡画像)をラベリング処理したラベル画面により、ラベル毎にソース画面の最小 / 最大濃度値抽出を行い、結果をユーザのテーブルへ返します。

### コーディング

```
int
IP_ProjectLabelGOMinMaxValue(
    DEVID          devID ,
    int            ImgSrc ,
    int            ImgLabel ,
    IPGOMinMaxTbl *Tbl
)
```

### コメント

デバイスID  
ソース画面番号  
ラベル画面番号  
ラベル毎最小 / 最大値格納テーブル

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面番号

**ImgLabel**  
ラベル画面番号

**\*Tbl**  
ラベル毎最小 / 最大値格納テーブル  
注) テーブルは必ず 2 5 6 確保してください。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

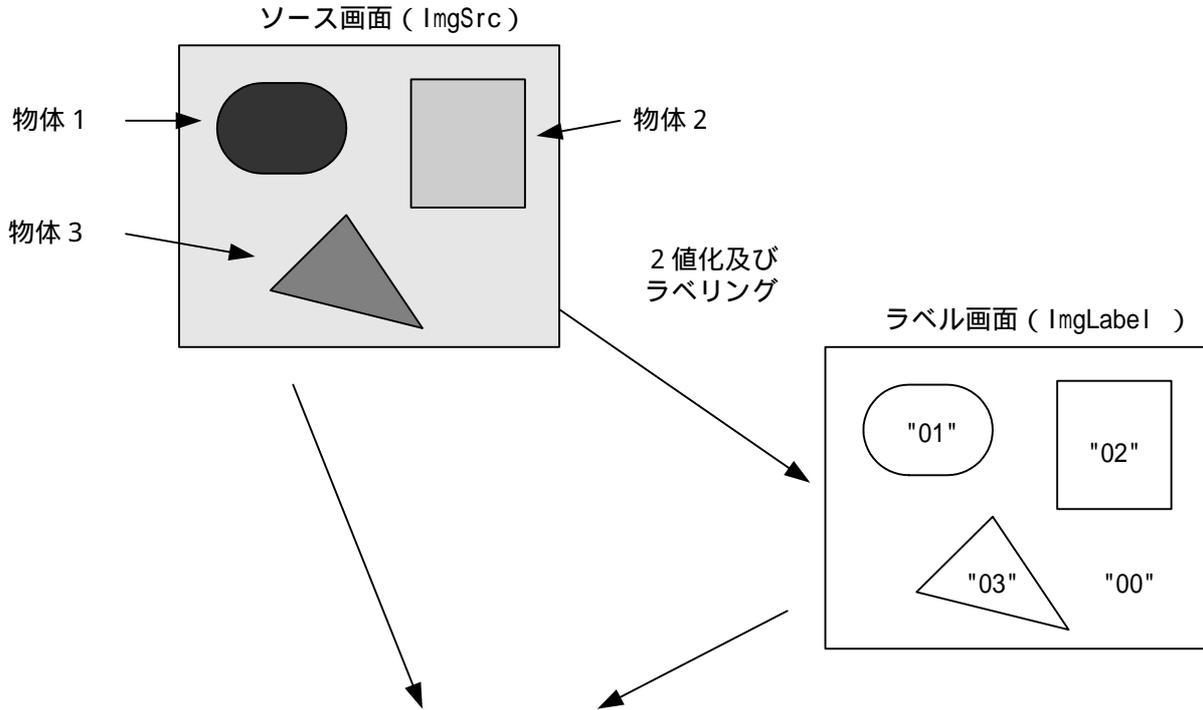
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	ソース画面空き領域、または範囲外 (不当画面番号エラー)
1 1	ラベル画面空き領域、または範囲外 (不当画面番号エラー)
2 2	テーブルのアドレスが不正
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

Tbl の参考例を記載します。



Tbl

ラベル	31	0
0	0	背景の濃度抽出値
1	1	物体 1 の濃度抽出値
2	2	物体 2 の濃度抽出値
3	3	物体 3 の濃度抽出値
4	4	0
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
254	254	0
255	255	0

本テーブルはipxsys.h で宣言しています。

```
typedef struct{
    int MIN_LEVEL;    // 最小濃度値
    int MAX_LEVEL;    // 最大濃度値
}IPGMinMaxTbl;
```

ラベル数を超える部分は  
MIN\_LEVEL = 32767  
MAX\_LEVEL = 0  
が入ります

ラベル画面の背景 ( 0 の部分 )  
の濃度抽出値が入ります

# EnableRotateProject

## 斜方投影処理有効

### 機能

ヒストグラム処理での座標軸の回転処理を有効にします。  
本コマンドは、ヒストグラム処理にしか有効ではありません。また、回転角45度付近では量子化誤差が非常に大きくなります。

### コーディング

```
int
EnableRotateProject(
    DEVID
    float
)
```

devID ,  
theta

### コメント

デバイス I D  
回転角

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**theta**  
回転角

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

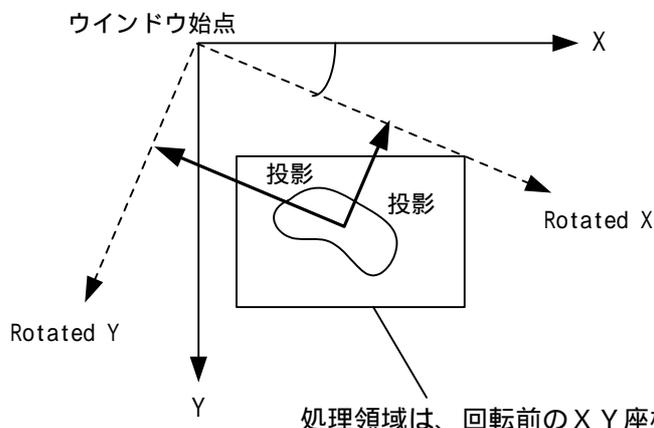
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 2	定数設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

回転角theta ( ) を設定することにより投影軸を回転して投影を行います。



変換式：  

$$\begin{aligned} \text{Rotated X} &= X \cos \theta + Y \sin \theta \\ \text{Rotated Y} &= -X \sin \theta + Y \cos \theta \end{aligned}$$

(注)本コマンドを実行し斜方投影を有効にしている場合、濃淡 / 2 値特徴量抽出の結果格納テーブルに出力される個数が必ず 1 0 2 4 個分になります。結果格納テーブルは 1 0 2 4 データ分確保して下さい。

## DisableRotateProject

斜方投影処理無効

### 機能

ヒストグラム処理での座標軸の回転処理を無効にします。  
(本コマンドは、ヒストグラム処理にしか有効ではありません。)

### コーディング

```
int  
DisableRotateProject(  
    DEVID          devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# IP\_HistogramFeatures

## 濃度ヒストグラム特徴量抽出

### 機能

ソース画面に対しヒストグラム処理を行い、濃度頻度累積及び濃度画像ヒストグラム特徴量抽出を行います。

### コーディング

```
int
IP_HistogramFeatures(
    DEVID
    int
    long
    HISTOGRAM_FEATURE
    int
)
```

```
devID ,
ImgSrc ,
*Tbl ,
*RegTbl ,
opt
```

### コメント

```
デバイスID
ソース画面番号
濃度頻度分布テーブル
ヒストグラム特徴量テーブル
オプション
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*Tbl**

濃度頻度分布を格納する領域へのポインタ。4バイト×256の容量を確保して下さい。

「NULL」を指定すると濃度頻度分布は出力されません。

**\*RegTbl**

ヒストグラム特徴量を格納する領域へのポインタ

**opt**

オプション。0を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

HISTOGRAM\_FEATUREのフォーマット及び内容を、下記に示します。

また、濃度累積値 (ACC\_LEVEL) が 0 以外のときのみ、テーブル内データが有効です。

ここで、最小 / 最大濃度値をとる点とはラスタスキャンした際に最初に出現した最小 / 最大濃度値をとる点を指します。

vpxsys.hで宣言しています

```
typedef struct {
    int    MIN_LEVEL_X;
    int    MIN_LEVEL_Y;
    int    MAX_LEVEL_X;
    int    MAX_LEVEL_Y;
    long   ACC_LEVEL;
    int    MIN_LEVEL;
    int    MAX_LEVEL;
    int    TYPICAL_LEVEL;
    long   N;
    float  MEAN;
    float  VARIANCE;
    float  DEVIATION;
} HISTOGRAM_FEATURE;
```

基本特徴テーブル	説明
MIN_LEVEL_X	最小濃度値をとる点のX座標
MIN_LEVEL_Y	最小濃度値をとる点のY座標
MAX_LEVEL_X	最大濃度値をとる点のX座標
MAX_LEVEL_Y	最大濃度値をとる点のY座標
ACC_LEVEL	濃度累積
MIN_LEVEL	最小濃度値
MAX_LEVEL	最大濃度値
TYPICAL_LEVEL	予備
N	総画素数
MEAN	平均濃度値
VARIANCE	分散値
DEVIATION	標準偏差

# IP\_ExtractB0Features

## 2値画像基本特徴量抽出

### 機能

ソース画面（2値）に対し、基本特徴量抽出を行います。

### コーディング

```
int
IP_ExtractB0Features(
    DEVID
    int
    IPB0FeatureTbl
    enum IPB0FeatureOpt
)
```

```
devID ,
ImgSrc ,
*RegTbl ,
opt
```

### コメント

デバイスID  
ソース画面番号  
基本特徴量テーブル  
2値画像基本特徴量抽出オプション

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*RegTbl**

基本特徴量を格納する領域へのポインタ

**opt**

2値画像基本特徴量抽出オプション

2値画像基本特徴量抽出オプション	対応定数	内容
BINARY_REGISTER	0	H P 並列特徴量のみ
BINARY_CENTER_of_GRAVITY	1	+ 重心
BINARY_MOMENT	2	+ 角度 + 2次モーメント

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外（不当画面番号エラー）
2 2	オプション設定値範囲外
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

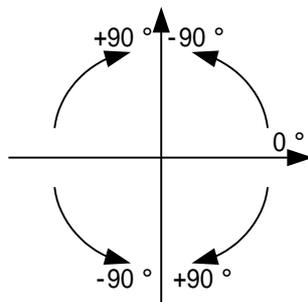
IPBOFeatureTblのフォーマット及び内容を、下記に示します。  
 2値物体（白）の面積が0以外のおきのみ、テーブル内データが有効です。  
 また、重心及び各種モーメントは、2値物体が1つのときのみ有効です。

vpxsys.hで宣言しています

```
typedef struct {
    int    LEFT_UP_POINT_X;
    int    LEFT_UP_POINT_Y;
    int    RIGHT_DOWN_POINT_X;
    int    RIGHT_DOWN_POINT_Y;
    long   AREA;
    int    REGION_X0;
    int    REGION_Y0;
    int    REGION_X1;
    int    REGION_Y1;
    float  CENTER_of_GRAVITY_X;
    float  CENTER_of_GRAVITY_Y;
    float  ANGLE;
    float  MOMENT_SIGMA_X;
    float  MOMENT_SIGMA_Y;
    float  MOMENT_SIGMA_XX;
    float  MOMENT_SIGMA_XY;
    float  MOMENT_SIGMA_YY;
} IPBOFeatureTbl;
```

基本特徴テーブル	説明
LEFT_UP_POINT_X	2値物体（白）が最初に出現する点のX座標
LEFT_UP_POINT_Y	2値物体（白）が最初に出現する点のY座標
RIGHT_DOWN_POINT_X	2値物体（白）が最後に出現する点のX座標
RIGHT_DOWN_POINT_Y	2値物体（白）が最後に出現する点のY座標
AREA	2値物体（白）の面積
REGION_X0	2値物体（白）が存在する最小X座標
REGION_Y0	2値物体（白）が存在する最小Y座標
REGION_X1	2値物体（白）が存在する最大X座標
REGION_Y1	2値物体（白）が存在する最大Y座標
CENTER_of_GRAVITY_X	重心X座標
CENTER_of_GRAVITY_Y	重心Y座標
ANGLE	慣性主軸とX軸とのなす角度（degree）（注）
MOMENT_SIGMA_X	X軸への1次モーメント
MOMENT_SIGMA_Y	Y軸への1次モーメント
MOMENT_SIGMA_XX	X軸への2次モーメント
MOMENT_SIGMA_XY	共役モーメント
MOMENT_SIGMA_YY	Y軸への2次モーメント

（注）



# IP\_ProjectB0

## X/Yへの投影

### 機能

2値画像でのX/Y軸への投影を行い、物体（白）の個数を算出します。

### コーディング

```
int
IP_ProjectB0(
    DEVID
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
*TbIX ,
*TbIY
```

### コメント

```
デバイスID
ソース画面番号
物体の個数格納テーブル
物体の個数格納テーブル
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*TbIX**

物体の個数（X軸への投影）を格納する領域へのポインタ。テーブルサイズは、SRC0\_WINで指定した範囲でX/Y軸で長い方に合わせて下さい。

**\*TbIY**

物体の個数（Y軸への投影）を格納する領域へのポインタ。テーブルサイズは、SRC0\_WINで指定した範囲でX/Y軸で長い方に合わせて下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

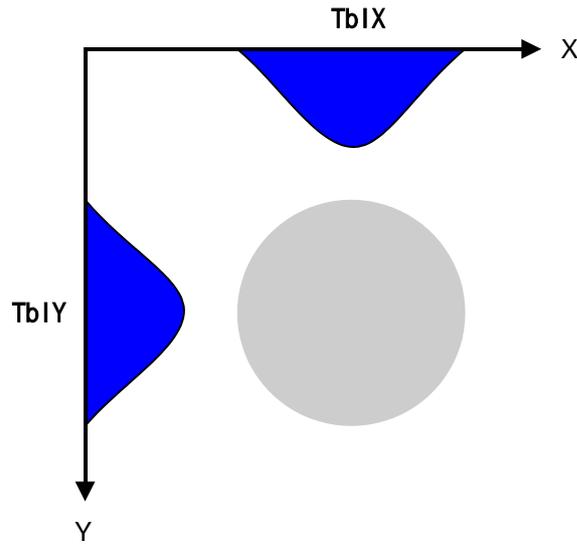
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

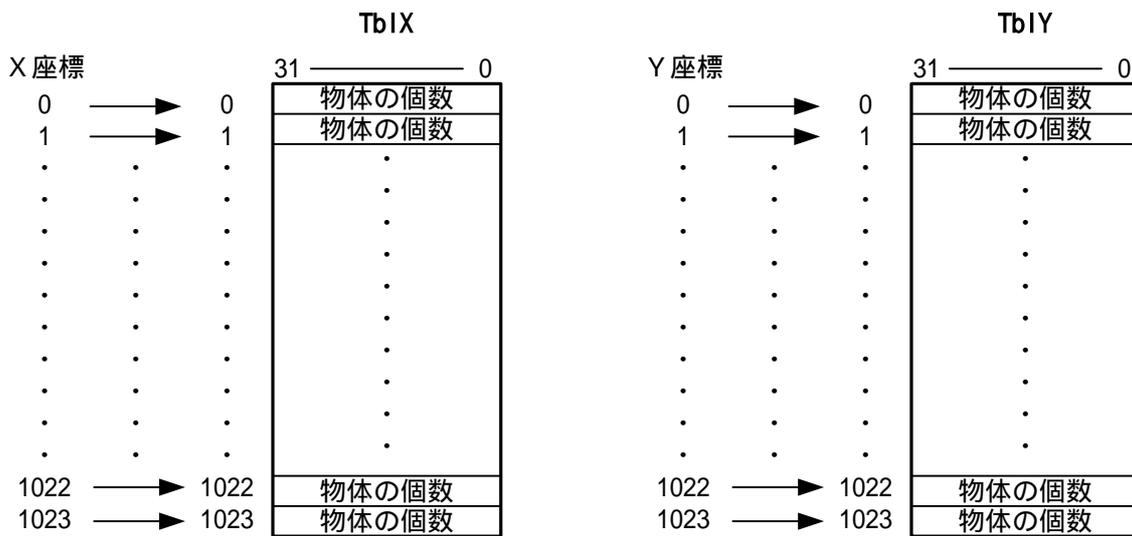
エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報



TbIX、TbIYのフォーマットを下記に示します。



本コマンドで得られるTbIX、TbIYの投影値はSRC0\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

$$\begin{aligned} \text{prjX} &= \text{TbIX}[ X - \text{sx} ] \\ \text{prjY} &= \text{TbIY}[ Y - \text{sy} ] \end{aligned}$$

- prjX : X軸投影値
- prjY : Y軸投影値
- X : 画面X座標
- Y : 画面Y座標
- sx : SRC0\_WIN(sx)
- sy : SRC0\_WIN(sy)

となります。

# IP\_ProjectB0RegionX

## 領域X座標抽出 (Y軸へのMin&MaxX投影)

### 機能

2値画像でのY軸への投影を行い、物体（白）が存在するMin/MaxのX座標抽出を行います。

### コーディング

```
int
IP_ProjectB0RegionX(
    DEVID
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
*TbIMinX ,
*TbIMaxX
```

### コメント

デバイスID  
ソース画面番号  
物体が存在するMinX座標格納テーブル  
物体が存在するMaxX座標格納テーブル

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*TbIMinX**

物体が存在するMinX座標を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定したX方向の範囲になります。

**\*TbIMaxX**

物体が存在するMaxX座標を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定したX方向の範囲になります。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

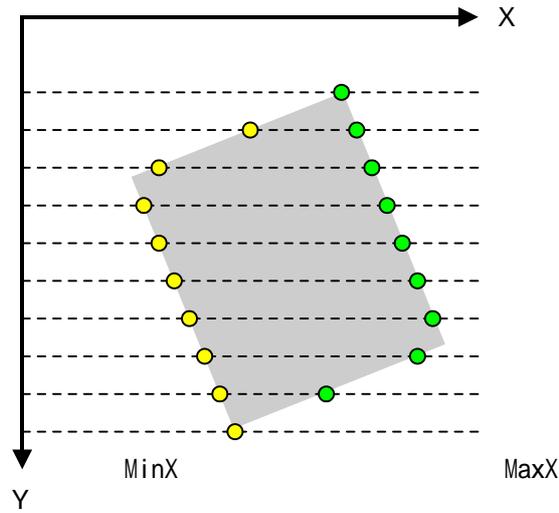
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報



TbIMinX、TbIMaxXのフォーマットを下記に示します。  
 また、物体（白）が存在しないY座標上のMinX座標には32767（0x7FFF）を、MaxX座標には0を出力します。

Y座標	TbIMinX		TbIMaxX	
	31	0	31	0
0	0	MinX座標	0	MaxX座標
1	1	MinX座標	1	MaxX座標
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
1022	1022	MinX座標	1022	MaxX座標
1023	1023	MinX座標	1023	MaxX座標

本コマンドで得られるMinX、MaxXの投影値はSRC0\_WINで設定されたウィンドウからの相対座標になります。  
 画面上の座標位置はそれぞれ

$$\begin{aligned} \text{MinX} &= \text{TbIMinX}[ Y - \text{sy} ] + \text{sx} \\ \text{MaxX} &= \text{TbIMaxX}[ Y - \text{sy} ] + \text{sx} \end{aligned}$$

- MinX : MinX座標
- MaxX : MaxX座標
- Y : 画面Y座標
- sx : SRC0\_WIN(sx)
- sy : SRC0\_WIN(sy)

となります。

# IP\_ProjectB0RegionY

## 領域Y座標抽出 (X軸へのMin&MaxY投影)

### 機能

2値画像でのX軸への投影を行い、物体（白）が存在するMin/MaxのY座標抽出を行います。

### コーディング

```
int
IP_ProjectB0RegionY(
    DEVID
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
*TbIMinY ,
*TbIMaxY
```

### コメント

```
デバイスID
ソース画面番号
物体が存在するMinY座標格納テーブル
物体が存在するMaxY座標格納テーブル
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面の画面番号

#### \*TbIMinY

物体が存在するMinY座標を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定したY方向の範囲になります。

#### \*TbIMaxY

物体が存在するMaxY座標を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定したY方向の範囲になります。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

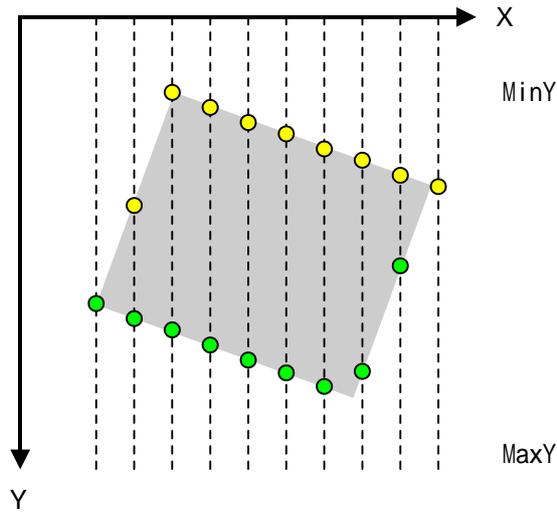
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

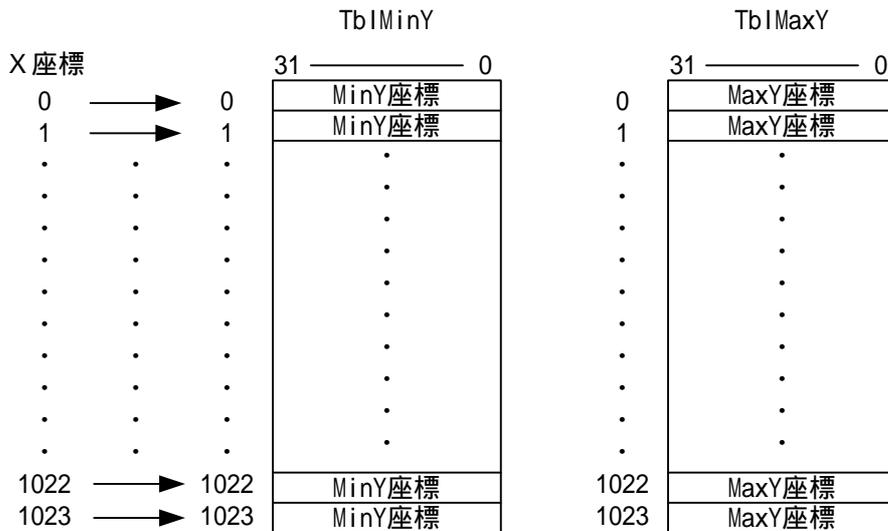
エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報



TbIMinY、TbIMaxYのフォーマットを下記に示します。  
 また、物体（白）が存在しないX座標上のMinY座標には32767（0x7FFF）を、MaxY座標には0を出力します。



本コマンドで得られるMinY、MaxYの投影値はSRC0\_WINで設定されたウィンドウからの相対座標になります。  
 画面上の座標位置はそれぞれ

$$\begin{aligned} \text{MinY} &= \text{TbIMinX}[ X - \text{sx} ] + \text{sy} \\ \text{MaxY} &= \text{TbIMaxX}[ X - \text{sx} ] + \text{sy} \end{aligned}$$

- MinY : MinY座標
- MaxY : MaxY座標
- X : 画面X座標
- sx : SRC0\_WIN(sx)
- sy : SRC0\_WIN(sy)

となります。

## IP\_ExtractBOArea

## 2値画像累積値抽出

### 機能

ソース画像(2値)に対し、2値の面積累積値を抽出する。

### コーディング

```
int
IP_ExtractBOArea(
    DEVID
    int
    long
)
```

```
devID ,
ImgSrc ,
*Area
```

### コメント

デバイスID  
ソース画面番号  
2値画像データの面積格納先アドレス

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面番号

#### \*Area

2値画像データの面積格納先アドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
-	エラーリセットコマンド (ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# IP\_ProjectBlockB0

## 領域毎2値画像濃度累積

### 機能

ソース画像(2値)の画面領域を設定したパラメータで分割し、各領域毎の濃度累積値をユーザのテーブルへ返します。

### コーディング

```
int
IP_ProjectBlockB0(
    DEVID
    int
    long
    IPDivideTbl
)
```

```
devID ,
ImgSrc ,
*Tbl ,
*DivideTbl
```

### コメント

デバイスID  
ソース画面番号  
領域毎濃度累積値データを格納するアドレス  
領域分割数を格納するアドレス

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面番号

#### \*Tbl

領域毎濃度累積値データを格納するアドレス

注) テーブルサイズは領域分割数(X軸分割数×Y軸分割数)用意してください。(最大768)

#### \*DivideTbl

領域分割数を格納するアドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外(不当画面番号エラー)
2 2	領域分割数が不正
4 9	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画面分割数と処理領域の間に以下の関係が成立しないと特徴量抽出で誤差が発生します。

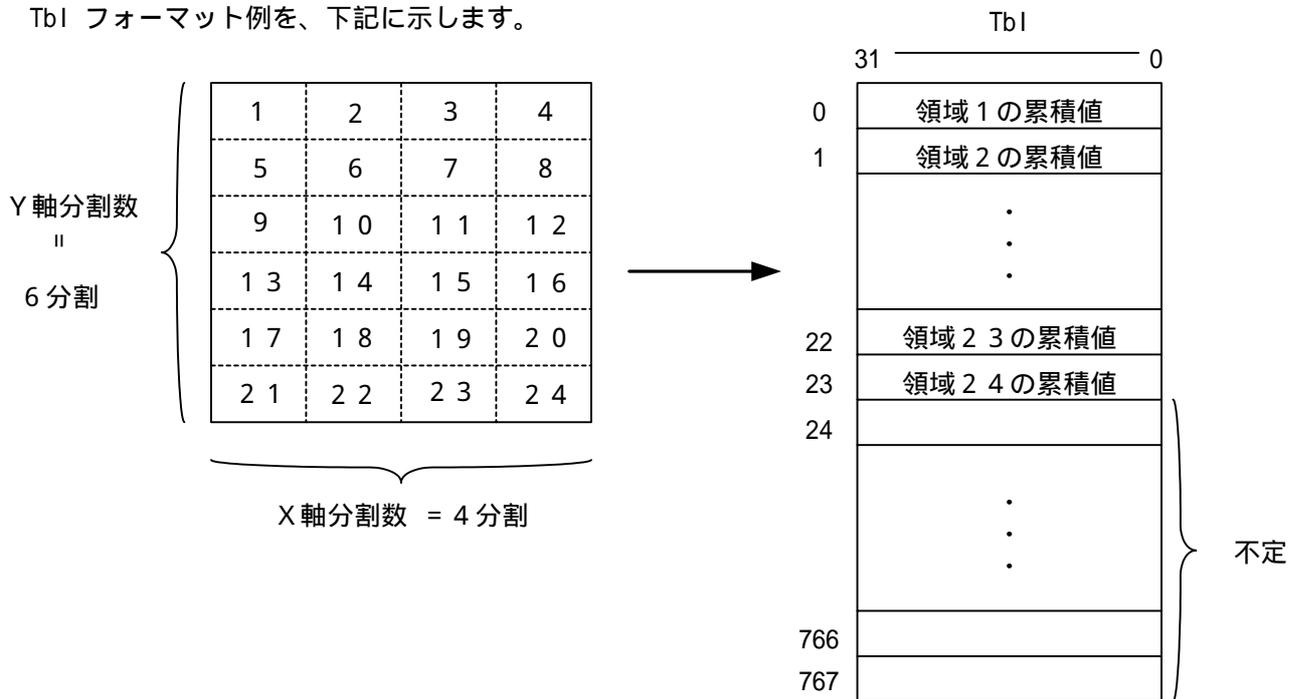
- 条件(1) (画面分割数) \* 1024 / (処理領域の長さ) が割り切れること
- (2) (処理領域の長さ) / (画面分割数) が割り切れること

つまり、(画面分割数)と(処理領域の長さ)が2のN乗以外は、特徴量抽出に誤差が発生します。これは、IP\_ProjectBlockB0()及びIP\_ProjectBlockG0()を行う場合のヒストグラムメモリ(HM)への画像処理LSI内部のアドレス発生回路が固定小数点演算で行っており、上記(1)、(2)の条件を満たさないとHMのアドレスに誤差が生じる為です。なお、IP\_ProjectBlockB0()、IP\_ProjectBlockG0MinMaxValue()、及びIP\_ProjectBlockG0()以外では、このアドレッシングを使用しないので、他のコマンドには影響ありません。

X、Y軸の分割数はウィンドウの画素数を越えて設定できません。また以下の制約がありますのでご注意ください。

X分割数	Y分割数	X分割数	Y分割数
1	1 ~ 7 6 8	1 7 ~ 3 2	1 ~ 2 4
2	1 ~ 3 8 4	3 3 ~ 6 4	1 ~ 1 2
3, 4	1 ~ 1 9 2	6 5 ~ 1 2 8	1 ~ 6
5 ~ 8	1 ~ 9 6	1 2 9 ~ 2 5 6	1 ~ 3
9 ~ 1 6	1 ~ 4 8	2 5 7 ~ 7 6 8	1

Tbl フォーマット例を、下記に示します。



DivideTbl のフォーマットを下記に示します。

領域分割数設定テーブル

```
typedef struct {
    int Divide_X;      // 領域毎抽出X方向分割数
    int Divide_Y;      // 領域毎抽出Y方向分割数
} IPDivideTbl;
```

本テーブルはipxsys.h で宣言しています。

# vpxAllocCorrTemplate

## テンプレート特徴量データ領域確保

### 機能

正規化関連サーチのテンプレート特徴量データ領域の確保を行います。

### コーディング

```

int
vpxAllocCorrTemplate(
    DEVID    devID,
    int      Tempsize
)

```

### コメント

デバイスID  
テンプレートサイズ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

Tempsize

テンプレートサイズ(テンプレートの個数)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	テンプレートが確保できない
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## vpxFreeCorrTemplate

### テンプレート特徴量データ領域開放

#### 機能

正規化関連サーチのテンプレート特徴量データ領域の開放を行います。

#### コーディング

```
void  
vpxFreeCorrTemplate(  
    DEVID    devID  
)
```

#### コメント

デバイスID

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### リターン値

なし。

#### 詳細情報

# vpxReadCorrTemplate

## テンプレート特徴量データ読み出し

### 機能

正規化関連サーチのテンプレート特徴量データを読み出します。

### コーディング

```
int
vpxReadCorrTemplate(
    DEVID      devID ,
    int        TmpID ,
    CORRTEMPLATE *CorrTmp
)
```

### コメント

デバイスID  
テンプレートID  
テンプレートデータテーブル

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**TmpID**  
テンプレートID

**\*CorrTmp**  
テンプレートデータテーブルへのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
8 1	テンプレートIDが不当
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

CORRTEMPLATEのフォーマット及び内容を、下記に示します。

vpxsys.hで宣言しています

```
typedef struct {
    RECTWINDOW      Win;
    short           xleng;
    short           yleng;
    short           v;
    short           lmgID;
    short           filter;
    short           opt;
    short           affsrc;
    short           afftmp;
    int             xmag;
    int             ymag;
    float           dx;
    float           dy;
    unsigned long   tn;
    unsigned long   sum;
    unsigned long   sqsum;
} CORRTEMPLATE;
```

```
typedef struct {
    short          sx;
    short          sy;
    short          ex;
    short          ey;
    short          xleng;
    short          yleng;
} RECTWINDOW;
```

テンプレートデータテーブル	説明
Win	テンプレートのウィンドウ情報
xleng	テンプレートのアフェイン変換前のX方向の長さ
yleng	テンプレートのアフェイン変換前のY方向の長さ
v	テンプレートの登録フラグ
lmgID	テンプレートのイメージID
filter	フィルタの種類
opt	オプション
affsrc	ソース画面のアフェイン変換
afftmp	テンプレートのアフェイン変換
xmag	X方向間引き間隔
ymag	Y方向間引き間隔
dx	テンプレート中心X方向オフセット
dy	テンプレート中心Y方向オフセット
tn	テンプレート画素数
sum	テンプレートの濃度総和
sqsum	テンプレートの濃度二乗総和

RECTWINDOW	説明
sx	テンプレートの開始点X座標
sy	テンプレートの開始点Y座標
ex	テンプレートの終了点X座標
ey	テンプレートの終了点Y座標
xleng	テンプレートのX方向長さ
yleng	テンプレートのY方向長さ

# vpXWriteCorrTemplate

## テンプレート特徴量データ書き込み

### 機能

正規化関連サーチのテンプレート特徴量データを書き込みます。

### コーディング

```
int
vpXWriteCorrTemplate(
    DEVID      devID ,
    int        TmpID ,
    CORRTEMPLATE *CorrTmp
)
```

### コメント

デバイスID  
テンプレートID  
テンプレートデータテーブル

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**TmpID**  
テンプレートID

**\*CorrTmp**  
テンプレートデータテーブルへのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
8 1	テンプレートIDが不当
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

CORRTEMPLATEのフォーマット及び内容は、vpXReadCorrTemplate()の詳細情報の欄を参照して下さい。

# vpxEnableCorrMask

## テンプレートマスクの有効化

### 機能

正規化関連サーチのテンプレートパターンのマスクを有効にします。

### コーディング

```
int
vpxEnableCorrMask(
    DEVID    devid
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

正規化関連マスクは、テンプレート内で濃度0の画素を処理対象外にします。よって、矩形でない任意の形をテンプレートとしたい場合に有効です。矩形のテンプレート内で必要物体部以外を0で塗りつぶしておき、vpxIP\_Corr()を実行する前に、vpxEnableCorrMask()を実行すれば、0画素部分は処理されないため、任意形のテンプレートで処理したことと同一となります。vpxEnableCorrMask()コマンドは、テンプレート登録 (vpxSetCorrTemplate()) を行う前に実行して下さい。

# vpxDisableCorrMask

## テンプレートマスクの無効化

### 機能

正規化関連サーチのテンプレートパターンのマスクを無効にします。

### コーディング

```
int
vpxDisableCorrMask(
    DEVID    devid
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

正規化関連マスクは、テンプレート内で濃度0の画素を処理対象外にします。よって、矩形でない任意の形をテンプレートとしたい場合に有効です。矩形のテンプレート内で必要物体部以外を0で塗りつぶしておき、vpxIP\_Corr()を実行する前に、vpxEnableCorrMask()を実行すれば、0画素部分は処理されないため、任意形のテンプレートで処理したことと同一となります。

vpxEnableCorrMask()コマンドで有効になっているマスクを無効にする場合、vpxDisableCorrMask()コマンドをテンプレート登録 (vpxSetCorrTemplate()) を行う前に実行して下さい。

# vpxSetCorrTemplate

## トレーニング

### 機能

正規化サーチのテンプレートパターンのテンプレート特徴量データの登録を行います。この関数でテンプレートの登録を行った場合は、正規化相関サーチの際にソース画像だけがxmag、ymagで示される間隔で間引きされます。つまり、テンプレートの大きさをxmag、ymagで指定した値で小さくできるということです。この場合、情報量はテンプレートの大きさと同じになります。

### コーディング

```
int
vpxSetCorrTemplate(
    DEVID devID,
    int    imgTmp,
    int    tmpID,
    int    xmag,
    int    ymag
)
```

### コメント

デバイスID  
テンプレート登録画面番号  
テンプレート番号  
X方向の間引き率  
Y方向の間引き率

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgTmp**  
テンプレートに登録する画面の画面番号。

**tmpID**  
テンプレートID。vpxAllocCorrTemplate()コマンドで確保したテンプレートの範囲で設定して下さい。

**xmag**  
X方向の間引き率(1~8)

**ymag**  
Y方向の間引き率(1~8)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
INVALID_CORR_AREA(-4)	テンプレートの面積エラー
INVALID_CORR_DATA(-5)	テンプレート画像の分散値エラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外(不当画面番号エラー)
2 2	X方向の間引き率設定値範囲外
2 3	Y方向の間引き率設定値範囲外
4 9	アクセス画面が不定画面
8 0	テンプレートサイズエラー
8 1	テンプレート画面の不当画面番号エラー
8 3	テンプレート内同一画素エラー
8 6	マスク処理でのテンプレート登録エラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

テンプレートの位置及びサイズは、ウィンドウ (SRCO\_WIN) によって決まります。本コマンド設定前に必ず、ウィンドウ設定コマンドによりSRCO\_WINウィンドウを設定して下さい。

また、テンプレートのサイズは2~65,536画素 (256×256相当) までとし、テンプレート内が同一画素値の時はエラーとなります。

正規化テンプレート登録では、テンプレート画像の存在する画面番号 (ImgTmp) と座標情報のみ登録します。よって、テンプレート登録画面は、開放 (FreeImg) したり、画像処理によるデータ変更を行わないで下さい。

## vpxSetCorrTemplateExt

## トレーニング

## 機能

正規化サーチのテンプレートパターンのテンプレート特徴量データの登録を行います。この関数でテンプレートの登録を行った場合は、正規化相関サーチの際にソース画像とテンプレート画像の両方がxmag、ymagで示される間隔で間引きされます。つまり、テンプレートの大きさを変更せずに情報量を制限できるということです。

## コーディング

```
int
vpxSetCorrTemplateExt(
    DEVID devID ,
    int    ImgTmp ,
    int    TmpID ,
    int    xmag ,
    int    ymag
)
```

## コメント

デバイスID  
テンプレート登録画面番号  
テンプレート番号  
X方向の間引き率  
Y方向の間引き率

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgTmp

テンプレートに登録する画面の画面番号。

TmpID

テンプレートID。vpxAllocCorrTemplate()コマンドで確保したテンプレートの範囲で設定して下さい。

xmag

X方向の間引き率(1~8)

ymag

Y方向の間引き率(1~8)

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
INVALID_CORR_AREA(-4)	テンプレートの面積エラー
INVALID_CORR_DATA(-5)	テンプレート画像の分散値エラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外(不当画面番号エラー)
2 2	X方向の間引き率設定値範囲外
2 3	Y方向の間引き率設定値範囲外
4 9	アクセス画面が不定画面
8 0	テンプレートサイズエラー
8 1	テンプレート画面の不当画面番号エラー
8 3	テンプレート内同一画素エラー
8 6	マスク処理でのテンプレート登録エラー
-	エラーリセットコマンド(ResetIPErrorTable)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

テンプレートの位置及びサイズは、ウィンドウ (SRCO\_WIN) によって決まります。本コマンド設定前に必ず、ウィンドウ設定コマンドによりSRCO\_WINウィンドウを設定して下さい。

また、テンプレートのサイズは2~65,536画素 (256×256相当) までとし、テンプレート内が同一画素値の時はエラーとなります。

正規化テンプレート登録 (間引き付き) では、テンプレート画像の存在する画面番号 (ImgTmp) と座標情報と間引き率を登録します。よって、テンプレート登録画面は、開放 (FreeImg) したり、画像処理によるデータ変更を行わないで下さい。

間引き率を2とすると、テンプレート内データは1画素おきで処理されるため、データ量は半分になります。間引きテンプレートは、処理が高速ですが、精度がフルテンプレートより落ちます。処理対象物体によって、フルテンプレート/間引きテンプレートどちらが有効か、また、間引きテンプレートによる間引き率は異なるため、実際に評価を行い、使用して下さい。

精度

フルテンプレート &gt; 間引きテンプレート

処理速度

フルテンプレート &lt; 間引きテンプレート

## vpxIP\_CorrStep

## ラスターサーチ

## 機能

矩形領域の正規化相関サーチを行います。

サーチで得られるスコア(score)は、実際の相関値(r)の2乗を10000倍した値です。

$$\text{score} = r * r * 10000$$

## コーディング

```
int
vpxIP_CorrStep(
    DEVID
    int
    int
    CORRTBL
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
TmpID ,
*tbl ,
cNum ,
StepX ,
StepY
```

## コメント

```
デバイスID
ソース画面番号
テンプレート番号
正規化相関サーチ結果格納テーブル
サーチ個数設定
X方向サーチ間隔
Y方向サーチ間隔
```

## パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**TmpID**

テンプレート番号

**\*tbl**

正規化相関サーチ結果格納テーブルへのポインタ

**cNum**

サーチ個数指定

**StepX**

X方向サーチ間隔

**StepY**

Y方向サーチ間隔

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。サーチ個数。
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

SRCO\_WINで指定された矩形領域をStepX、StepYで指定された間隔でラスタースーチします。そしてcnumで指定された個数の正規化相関データを相関係数の大きい順にソートしてtblに格納します。この関数で得られるサーチ結果の座標は、テンプレートの左上のポイントです。

正規化相関サーチ結果格納テーブルのフォーマットを、下記に示します。

本テーブルはvpxsys.hで宣言しています。

```
typedef struct {  
    long          score;  
    long          x;  
    long          y;  
} CORRTBL;
```



**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。サーチ個数。
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

入力ポイントテーブルinTblに入力された座標から、StepX、StepYで指定した間隔でLengX、LengYで指定した幅の正規化相関サーチをinNum回数行い、outNumで指定された個数の正規化相関データを相関係数の大きい順にソートしてoutTblに格納します。この関数で得られるサーチ結果の座標は、テンプレートの左上のポイントです。

CORRTBLのフォーマットは、vpxIP\_CorrStep()の詳細情報の欄を参照して下さい。

## vpxIP\_CorrPrecise

## サブピクセルサーチ

### 機能

正規化相関サーチから得られたポイントから近傍領域で正規化相関サーチを行い、その相関値からサブピクセルの位置計算を行います。

サーチで得られるスコア(score)は、実際の相関値(r)の2乗を10000倍した値です。

$$\text{score} = r * r * 10000$$

### コーディング

```
int
vpxIP_CorrPrecise(
    DEVID
    int
    int
    CORRTBL
    FCORRTBL
)
```

```
devID ,
ImgSrc ,
TmpID ,
*inTbl ,
*outTbl
```

### コメント

デバイスID  
ソース画面番号  
テンプレート番号  
入力処理ポイント  
サブピクセル計算結果

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

TmpID

テンプレート番号

\*inTbl

入力処理ポイント

\*outTbl

サブピクセル計算結果

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。サーチ個数。
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

CORRTBLのフォーマットは、IP\_CorrStep()の詳細情報の欄を参照して下さい。

FCORRTBLはvpxsys.hで宣言しています。

```
typedef struct {
    long        score;
    float       x;
    float       y;
} FCORRTBL;
```

## vpxEnableCorrBreak

### 相関演算途中打ち切りの有効化

#### 機能

正規化相関サーチの相関演算のブレイク機能（相関演算途中打ち切り）を許可します。相関演算の値とテンプレートデータの値の差がvpxSetCorrBreakThr()コマンドで設定した値よりも大きい場合相関演算を途中で打ち切ります。

なお、この機能を使用すると、明るさの変動に影響されやすくなりますので注意して下さい。

#### コーディング

```
int  
vpxEnableCorrBreak(  
    DEVID          devID  
)
```

#### コメント

デバイスID

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

# vpxDisableCorrBreak

## 相関演算途中打ち切りの無効化

### 機能

vpxEnableCorrBreak() コマンドで有効化した正規化相関サーチの相関演算のブレイク機能を無効にします。

### コーディング

```
int
vpxDisableCorrBreak(
    DEVID          devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## vpxSetCorrBreakThr

### 相関演算途中打ち切り閾値設定

#### 機能

正規化相関値サーチの相関演算のブレイク機能での相関演算途中打ち切りを行うための差分累積値のしきい値を設定します。相関演算途中の総和値とテンプレートデータの総和値の差が `thr` よりも大きい場合、ハード的に相関演算を途中で打ち切り、サーチを高速化します。

なお、この機能を使用すると、明るさの変動に影響されやすくなりますので注意して下さい。

#### コーディング

```
int
vpxSetCorrBreakThr(
    DEVID
    int
    int
)
```

```
devID ,
thr ,
opt
```

#### コメント

デバイスID  
しきい値  
オプション

#### パラメータ

`devID`

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

`thr`

相関演算途中打ち切りを行うための差分累積値のしきい値。ここでは、1画素のレベル0～255の値を設定して下さい。ここで指定された`thr`はテンプレートの画素数換算されたものが実際の差分累積しきい値になります。

$$\text{差分累積しきい値} = \text{thr} \times (\text{テンプレートの画素数})$$

`opt`

オプション。処理打ち切り方法

オプション	内容
0	常に <code>thr</code> で指定したしきい値で演算をうち切る
1	残差逐次検定演算 (SSDA) 法により演算をうち切る

#### リターン値

リターン値は `int` 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20, 21	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

しきい値 (`thr`) が小さいほど高速になりますが、明るさの変動に影響されやすくなります。現実的なしきい値は、64程度かそれ以上と考えて下さい。

なお、`thr` に「0」を指定すると最大値が設定されます。

# vpxSetSearchDistance

## サーチ除外領域の設定

### 機能

正規化関連サーチのサーチ領域の除外領域を設定します。正規化関連サーチのパラメータのサーチ個数が2以上の場合、抽出座標の間隔を設定し、近傍座標の抽出を防止します。尚、除外領域X間隔、除外領域Y間隔の1ステップは、正規化関連サーチのパラメータのX方向サーチ間隔、Y方向サーチ間隔になります。

### コーディング

```
int
vpxSetSearchDistance(
    DEVID
    int
    int
    int
)
```

```
devID ,
mode ,
xleng ,
yleng
```

### コメント

```
デバイスID
サーチモード
除外領域X間隔
除外領域Y間隔
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### mode

サーチモード

サーチモード	内容
0	ノーマルモード。サーチ除外領域モード解除。
1	サーチ除外領域モード。xleng,ylengが有効になります。

#### xleng

除外領域X間隔。間隔の1ステップは、正規化関連サーチのパラメータのX方向サーチ間隔になります。

#### yleng

除外領域Y間隔。間隔の1ステップは、正規化関連サーチのパラメータのY方向サーチ間隔になります。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20 ~ 22	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# vpxSetCorrMode

## 関連サーチモード設定

### 機能

正規化関連サーチでの処理モードの設定を行います。

### コーディング

```
int
vpxSetCorrMode(
    DEVID          devID ,
    enum IPCorrMode mode
)
```

### コメント

デバイス I D  
正規化関連処理モード

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

mode

正規化関連処理モード

正規化関連サーチモード	対応数	説明
CORR_NORMAL	0	ノーマルモード
CORR_SINGLE_KERNEL	1	1 カーネルモード
CORR_PARALLEL_KERNEL	2	4 × 4 カーネルモード

相関値分布抽出モード	内容
0x00	符号なし抽出モード
0x20	符号付き抽出モード

相関値分布抽出コマンドの処理モードは、正規化関連処理モードと相関値分布抽出モードの論理和を設定してください。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# vpxSetCorrPrecise

## サブピクセル算出桁数設定

### 機能

vpxIP\_CorrPrecise() コマンドで出力される X , Y 座標の小数点の桁数を指定します。

### コーディング

```
int
vpxSetCorrPrecise(
    DEVID
    enum IPCorrPrecise
)
```

devID ,  
precise

### コメント

デバイス I D  
正規化相関小数点出力桁数

### パラメータ

devID

デバイス I D。OpenIPDev コマンドで得られた対応するボード番号のデバイス I D を指定して下さい。

precise

正規化相関小数点出力桁数

正規化相関小数点出力桁数	対応数	説明
CORR_PRECISE_1	0	小数点 1 桁
CORR_PRECISE_2	1	小数点 2 桁
CORR_PRECISE_3	2	小数点 3 桁
CORR_PRECISE_4	3	小数点 4 桁

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## vpxIP\_CorrMap

## 相関値分布の抽出

## 機能

SRCO\_WINで指定された矩形領域の相関値をすべて計算しテーブルに格納します。本コマンドを実行する場合、vpxGetCorrMapSize( )コマンドで抽出サイズ(xIng,yIng)、格納テーブルサイズ(size)を取得し、処理を行って下さい。

## コーディング

```
int
vpxIP_CorrMap(
    DEVID
    int
    int
    float
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
TmpID ,
*tbl ,
count ,
StepX ,
StepY
```

## コメント

```
デバイスID
ソース画面番号
テンプレート番号
相関値分布抽出データ格納テーブル
相関値分布抽出データ数
X方向サーチ間隔
Y方向サーチ間隔
```

## パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**TmpID**

テンプレート番号

**\*tbl**

相関値分布抽出データ格納テーブルへのポインタ。このテーブルには、データをDMA転送するため、転送単位が512バイト単位になります。つまり、確保する領域を512バイトでアラインする必要があります。そこでvpxGetCorrMapSize( )コマンドでその値を計算しますので、vpxGetCorrMapSize( )で得られたsize値以上の大きさの領域を確保して下さい。

なお、出力される相関値は、vpxSetCorrMode( )コマンドで正值抽出モードを指定した場合は0.0~1.0、正負値抽出モードを指定した場合は-1.0~1.0です。

**count**

相関値分布抽出データ数。vpxGetCorrMapSize( )で得られた(xIng \* yIng)を設定して下さい。

**StepX**

X方向サーチ間隔

**StepY**

Y方向サーチ間隔

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。相関値分布抽出データ数。
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

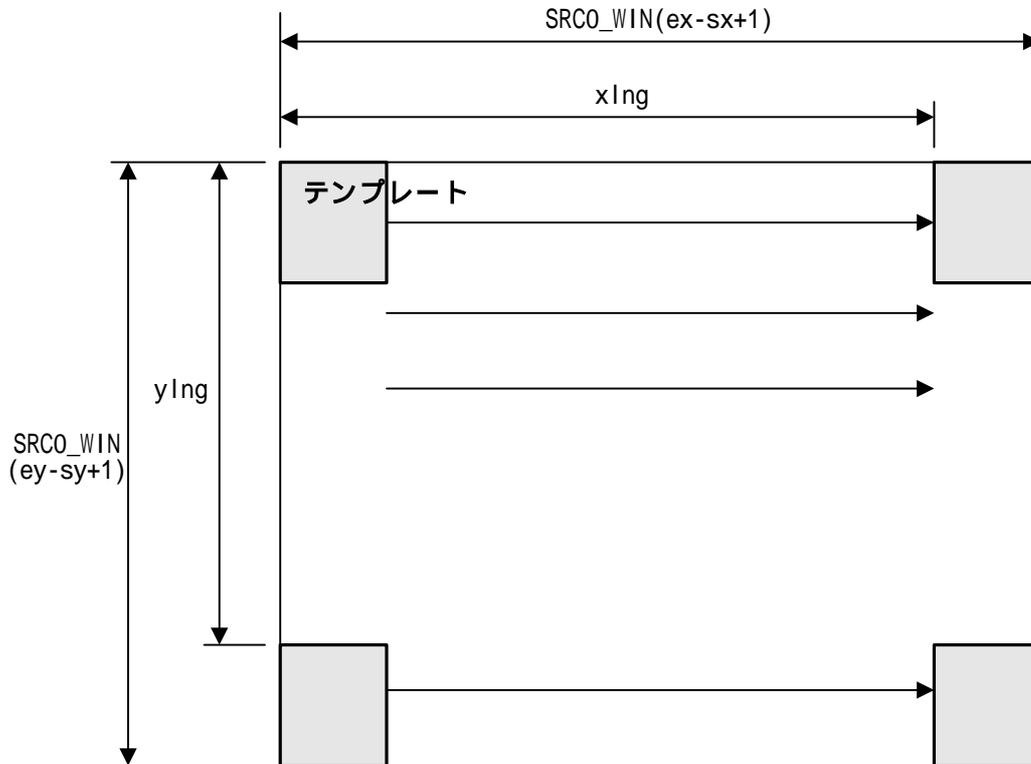
エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

SRCO\_WINで指定された矩形領域をStepX、StepYで指定された間隔でラスタースキャンして相関値を計算し相関値格納テーブルに相関値データをすべて格納します。

本コマンドを実行する前にvpxGetCorrMapSize()コマンドで出力相関値データのX方向のデータ長(xLng)、Y方向のデータ長(yLng)、相関値格納テーブル確保サイズ(size)を取得し、それに合わせて処理を行ってください。



以下に、プログラム例を示します。

```
int  xLng,yLng,size,count,num,i;
float *rtbl;
char *itbl;

.....
.....

SetWindow(devID,SRCO_WIN,0,0,511,479);
vpxGetCorrMapSize(devID,ImgSrc,TmpID,1,1,&xLng,&yLng,&size);

rtbl= (float*)malloc(size);
count= xLng * yLng;

num= vpxIP_CorrMap(devID,ImgSrc,TmpID,rtbl,count,1,1);

itbl= (char*)malloc(count);
for(i=0; i< count; i++){
    itbl[i]= (char)(rtbl[i]* 255.0f);
}

SetWindow(devID,SYS_WIN,0,0,xLng-1,yLng-1);
WriteImg(devID,ImgDisp,itbl,count);
Displmg(devID,ImgDisp);

free(itbl);
free(rtbl);
```

## vpxGetCorrMapSize

## 相関値分布格納テーブルサイズ取得

### 機能

SRCO\_WINで指定された画面の矩形領域と指定されたテンプレートデータから、抽出サイズ(xIng,yIng)、格納テーブルサイズ(size)を計算しそれぞれの値を出力します。

### コーディング

```
int
vpxGetCorrMapSize(
    DEVID    devID ,
    int      ImgSrc ,
    int      TmpID ,
    int      StepX ,
    int      StepY ,
    int      *xIng ,
    int      *yIng ,
    int      *size
)
```

### コメント

デバイスID  
ソース画面番号  
テンプレート番号  
X方向サーチ間隔  
Y方向サーチ間隔  
X方向データ長(出力)  
Y方向データ長(出力)  
相関値分布抽出データ格納テーブル確保サイズ(出力)

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**TmpID**  
テンプレート番号

**StepX**  
X方向サーチ間隔

**StepY**  
Y方向サーチ間隔

**xIng**  
X方向データ長(出力)

**yIng**  
Y方向データ長(出力)

**size**  
相関値分布抽出データ格納テーブル確保サイズ(出力)。vpxIP\_CorrMap()コマンドでは512バイト単位の転送を行う為、512バイトにアラインされたサイズが出力されます。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# HistAnalyze

## 判別分析法

### 機能

tblに設定されたヒストグラムテーブルの判別分析を行い、その特徴量を求めます。

### コーディング

```
int
HistAnalyze(
    DEVID      devID ,
    HISTANATBL *tbl ,
    int        lower ,
    int        upper
)
```

### コメント

デバイスID  
判別分析テーブル  
下位レベル  
上位レベル

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*tbl**  
判別分析テーブル

**lower**  
下位レベル

**upper**  
上位レベル

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	抽出された2値化しきい値
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
21 ~ 22	レベル設定値範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

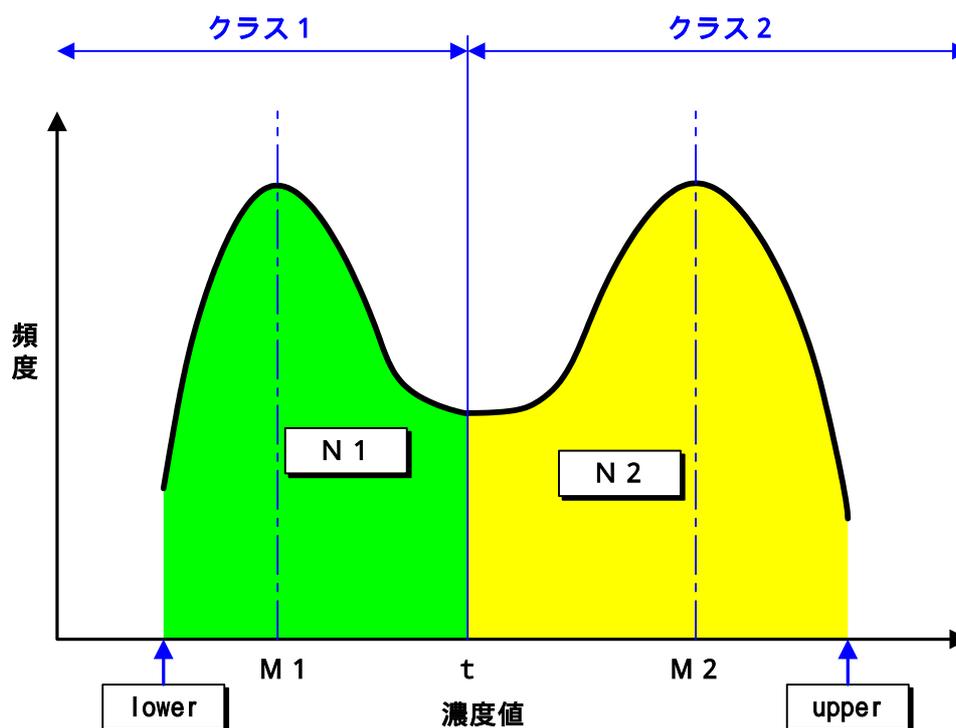
### 詳細情報

HISTANATBLのフォーマット及び内容を、下記に示します。

vpxsys.hで宣言しています

```
typedef struct {
    long      HistTbl[256];
    long      n1;
    long      n2;
    short     m1;
    short     m2;
    short     thr;
    short     ct;
} HISTANATBL;
```

判別分析テーブル	説明
HistTbl[256] (入力)	ヒストグラムテーブル。Histogram()コマンドで得られるヒストグラムテーブルをコピーするか、または、このポインタをHistogram()コマンドのパラメータに設定して下さい。
n1 (出力)	クラス1の総画素数
n2 (出力)	クラス2の総画素数
m1 (出力)	クラス1の平均濃度値
m2 (出力)	クラス2の平均濃度値
thr (出力)	2値化しきい値
ct (出力)	m2 - m1 (簡易コントラスト)



上図において

N 1	: クラス1の総画素数
N 2	: クラス2の総画素数
M 1	: クラス1の平均濃度値
M 2	: クラス2の平均濃度値
t	: レベル

のとき2つのクラス間分散は、

$$s^2 = N 1 \cdot N 2 \cdot (M 1 - M 2)^2$$

により求められます。そして、「t」の値を「lower」から「upper」まで変化させ、 $s^2$ を最大にするtの値を高速に計算しそのときのレベルをしきい値「thr」とするN 1, N 2, M 1, M 2を求めます。

# GetHoughLine

## 離散データ座標からの直線抽出

### 機能

PointTblに格納された点列をLINERANGEで設定された角度の範囲でハフ変換を行い、ハフ変換直線をサブピクセルの単位まで求めます。

### コーディング

```
int
GetHoughLine(
    DEVID      devid ,
    POINTTBL   *PointTbl ,
    LINERANGE  *AngleRange ,
    HOUGHLINE  *HoughLine
)
```

### コメント

デバイスID  
X, Y座標配列  
検索角度範囲  
ハフ変換直線算出

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*PointTbl**

X, Y座標配列(入力)

**\*AngleRange**

検索角度範囲(入力)

**\*HoughLine**

ハフ変換直線算出(出力)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。抽出点列数。
-1	異常終了
INVALID_LINE_DATA(-3)	直線のポイント数不足による直線抽出エラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	パラメータエラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

POINTTBL、LINERANGE、HOUGHLINEのフォーマットを、下記に示します。  
vpxsys.hで宣言しています。

```
typedef struct {
    short      num;
    short      opt;
    WPOINT     point[512];
} POINTTBL;
```

```
typedef struct {
    short  x;
    short  y;
} WPOINT;
```

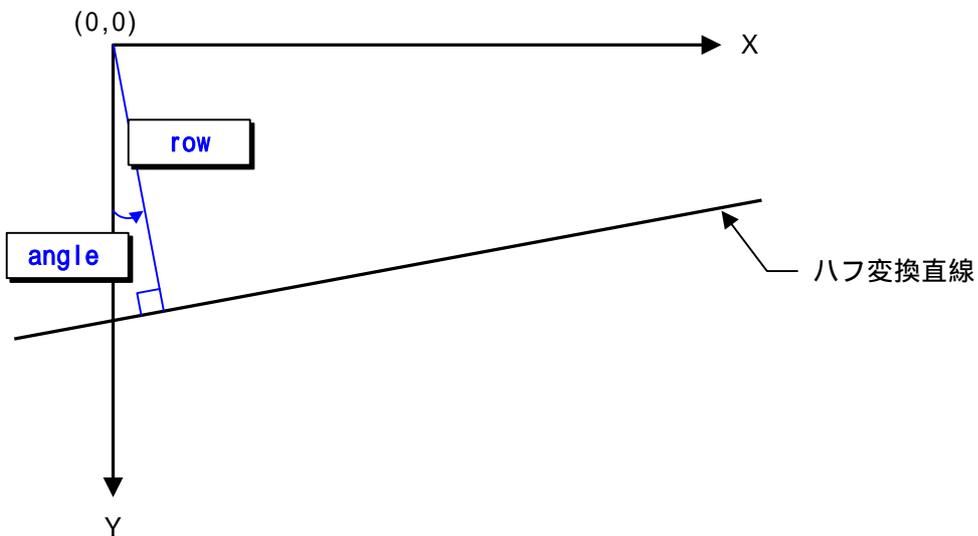
POINTTBL	説明
num	有効データ数
opt	未使用
point[512]	X , Y座標配列

```
typedef struct {
    short      lower;
    short      upper;
} LINERANGE;
```

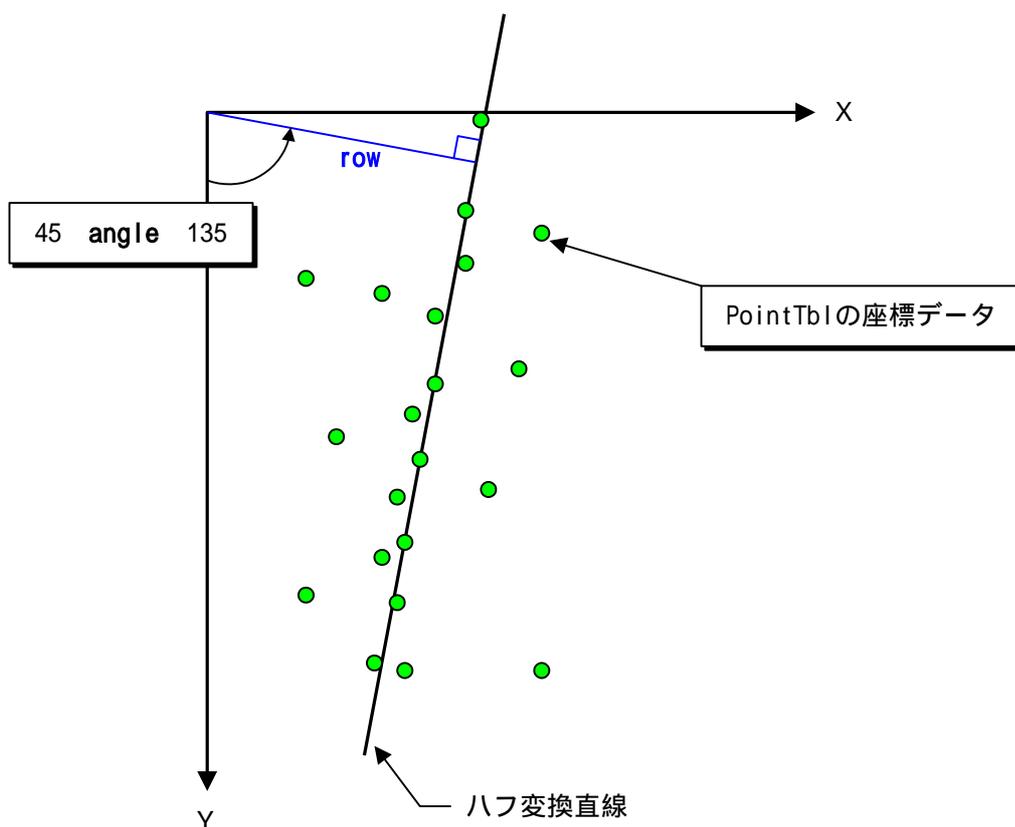
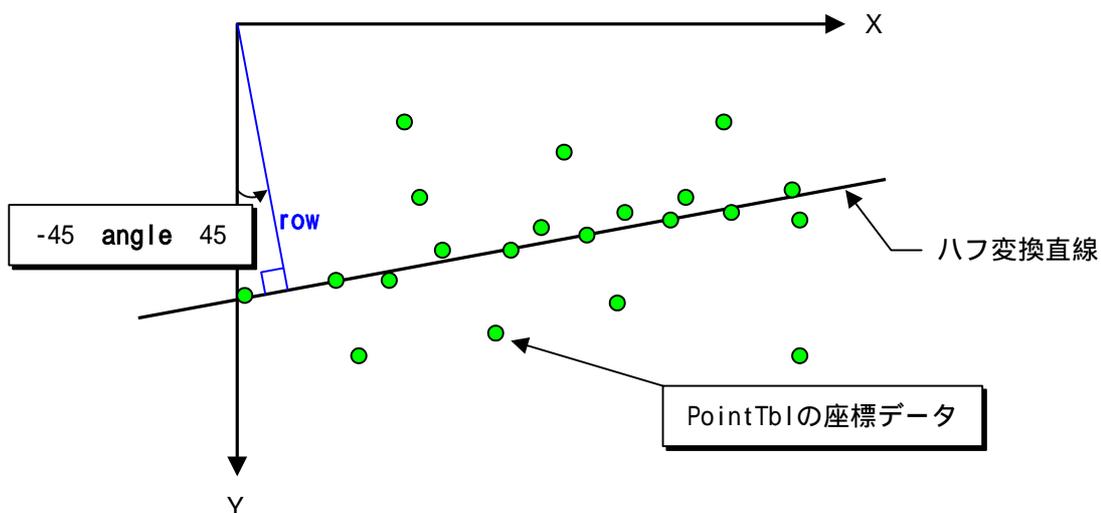
LINERANGE	説明
lower	検索範囲 下限
upper	検索範囲 上限

```
typedef struct {
    float      row;
    float      angle;
} HOUGHLINE;
```

HOUGHLINE	説明
row	ハフ変換直線 row
angle	ハフ変換直線 angle



本コマンドは、常に変換抽出数量が最大となる直線を抽出します。



ハフ変換のコマンドは、高速化のため、関数内部でPointTblに格納されたデータのバウンダリチェックを行いません。そのため、ユーザ側でバウンダリチェック及び領域の絞り込みを行う必要があります。

以下にその例を示しました。例では、領域X座標抽出(IP\_ProjectB0RegionX())、領域Y座標抽出(IP\_ProjectB0RegionY())での抽出座標のハフ変換を行っています。リストで「sx,sy,ex,ey」は、SetIPWindow()で指定した処理ウィンドウ領域です。またポイント座標(point.x,point.y)として指定できる範囲は0～511です。この範囲を超えるとシステム例外が発生する可能性がありますので注意して下さい。

```

int sx; /* 処理領域開始X座標 */
int sy; /* 処理領域開始Y座標 */
int ex; /* 処理領域終了X座標 */
int ey; /* 処理領域終了Y座標 */

POINTTBL RegionYMin,RegionYMax,RegionXMin,RegionXMax;
POINTTBL *RegionTbl;
int *LineTbl;
int TblMin[512],TblMax[512];
int TblX[512],TblY[512];
int lower,upper;
int i,j,x,y,n,val;

/** 領域YのMax,Min抽出 */
IP_ProjectB0RegionY(_devID,BinID,TblMin,TblMax);
LineTbl= TblMin; RegionTbl= &RegionYMin;
lower= sy; upper= ey;
for(n=0,i=0; i<ex-sx+1; i++){
    val= LineTbl[i]+ sy;
    if(lower < val && val < upper){
        RegionTbl->point[n].x= i+ sx;
        RegionTbl->point[n].y= val;
        n++;
    }
}
RegionTbl->num= n;
LineTbl= TblMax; RegionTbl= &RegionYMax;
lower= sy; upper= ey;
for(n=0,i=0; i<ex-sx+1; i++){
    val= LineTbl[i]+ sy;
    if(lower < val && val < upper){
        RegionTbl->point[n].x= i+ sx;
        RegionTbl->point[n].y= val;
        n++;
    }
}
RegionTbl->num= n;
/** 領域XのMax,Min抽出 */
IP_ProjectB0RegionX(_devID,BinID,TblMin,TblMax);
LineTbl= TblMin; RegionTbl= &RegionXMin;
lower= sx; upper= ex;
for(n=0,i=0; i<ey-sy+1; i++){
    val= LineTbl[i]+ sx;
    if(lower < val && val < upper){
        RegionTbl->point[n].x= val;
        RegionTbl->point[n].y= i+ sy;
        n++;
    }
}
RegionTbl->num= n;
LineTbl= TblMax; RegionTbl= &RegionXMax;
lower= sx; upper= ex;
for(n=0,i=0; i<ey-sy+1; i++){
    val= LineTbl[i]+ sx;
    if(lower < val && val < upper){
        RegionTbl->point[n].x= val;
        RegionTbl->point[n].y= i+ sy;
        n++;
    }
}
RegionTbl->num= n;

```

# GetHoughLineRow

## 離散データ座標からの2直線抽出

### 機能

PointTblに格納された点列をLINERANGEで設定された角度の範囲でハフ変換を行い、同一角度の2直線を求め、modeにより、2直線のrowのうち最小値又は最大値を検出し、ハフ変換直線をサブピクセルの単位まで求めます。

なお、本コマンドを使用する場合のバウンダリチェック及び領域の絞り込みについては、GetHoughLine()コマンドの詳細情報を参照して下さい。

### コーディング

```
int
GetHoughLineRow(
    DEVID      devid,
    POINTTBL   *PointTbl,
    LINERANGE  *AngleRange,
    HOUGHLINE  *HoughLine,
    int        mode
)
```

### コメント

デバイスID  
X, Y座標配列  
検索角度範囲  
ハフ変換直線算出  
ライン検出モード

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*PointTbl**  
X, Y座標配列(入力)

**\*AngleRange**  
検索角度範囲(入力)

**\*HoughLine**  
ハフ変換直線算出(出力)

**mode**  
ライン検出モード(入力)

検出モード	対応数	内容
MIN_LINE	0	rowの最小値を検出
MAX_LINE	1	rowの最大値を検出

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。抽出点列数。
-1	異常終了
INVALID_LINE_DATA(-3)	直線のポイント数不足による直線抽出エラー

異常終了時、下記の原因のコマンドエラーが考えられます。

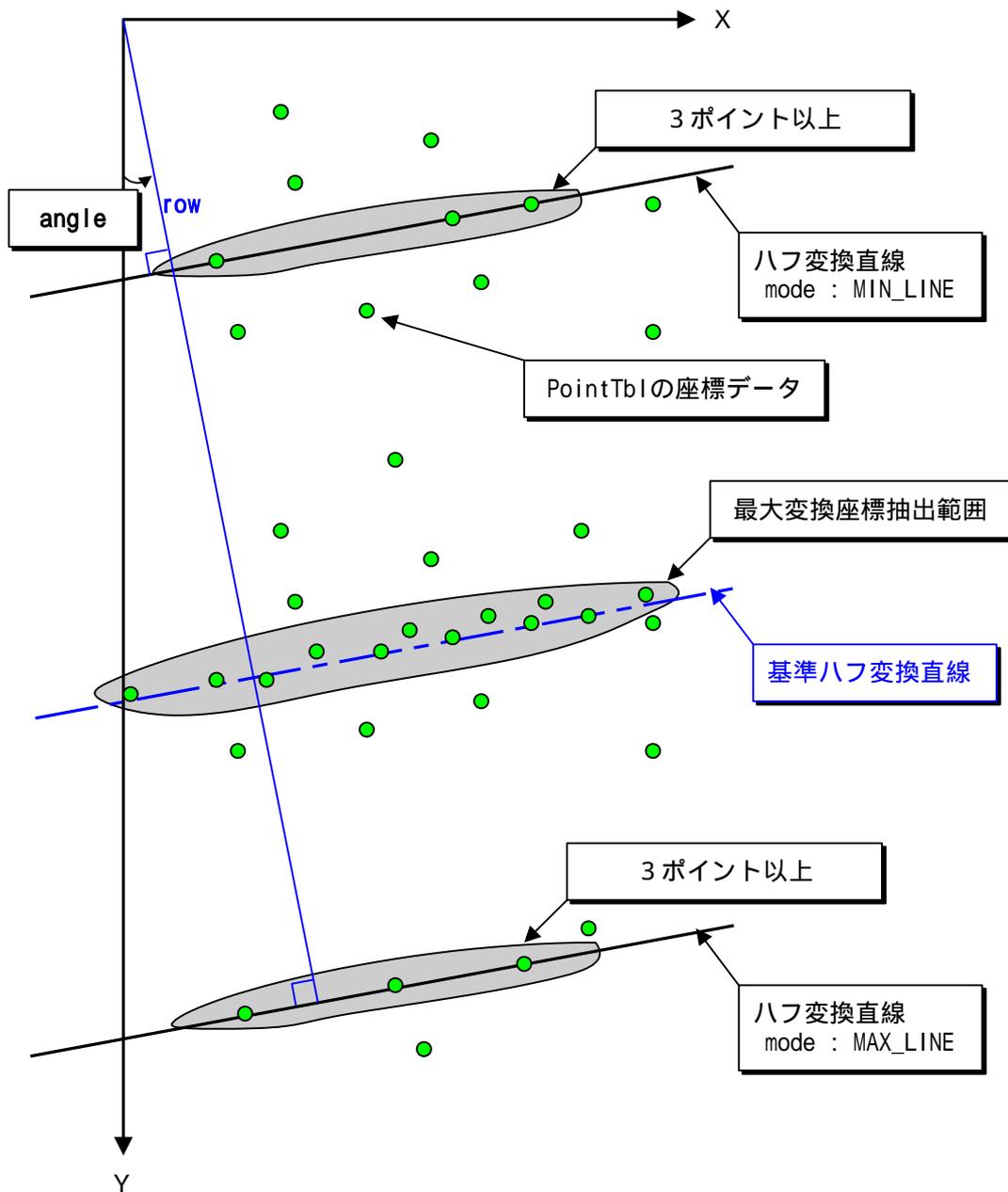
エラー番号	エラー原因
20	パラメータエラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

POINTTBL、LINERANGE、HOUGHLINEのフォーマットは、GetHoughLine()の詳細情報の欄を参照して下さい。

本コマンドは、まず変換抽出数量が最大となる直線を抽出し、その抽出した直線を基準ハフ変換直線にします。そして、基準ハフ変換直線の角度で変換抽出数のが3ポイント以上の直線を検索します。



# GetHoughLineRowExt

## 離散データ座標からの2直線抽出(拡張)

### 機能

PointTblに格納された点列をLINERANGEで設定された角度の範囲でハフ変換を行い、thrで設定された変換抽出数量しきい値以上の同一角度の2直線を求め、modeにより、2直線のrowのうち最小値又は最大値を検出し、ハフ変換直線をサブピクセルの単位まで求めます。

なお、本コマンドを使用する場合のバウンダリチェック及び領域の絞り込みについては、GetHoughLine()コマンドの詳細情報を参照して下さい。

### コーディング

```
int
GetHoughLineRowExt(
    DEVID      devID ,
    POINTTBL   *PointTbl ,
    LINERANGE  *AngleRange ,
    HOUGHLINE *HoughLine ,
    int        thr ,
    int        mode
)
```

### コメント

デバイスID  
X, Y座標配列  
検索角度範囲  
ハフ変換直線算出  
変換抽出数量しきい値  
ライン検出モード

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*PointTbl**  
X, Y座標配列(入力)

**\*AngleRange**  
検索角度範囲(入力)

**\*HoughLine**  
ハフ変換直線算出(出力)

**thr**  
変換抽出数量しきい値(入力)

**mode**  
ライン検出モード(入力)

検出モード	対応数	内容
MIN_LINE	0	rowの最小値を検出
MAX_LINE	1	rowの最大値を検出

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。抽出点列数。
-1	異常終了
INVALID_LINE_DATA(-3)	直線のポイント数不足による直線抽出エラー

異常終了時、下記の原因のコマンドエラーが考えられます。

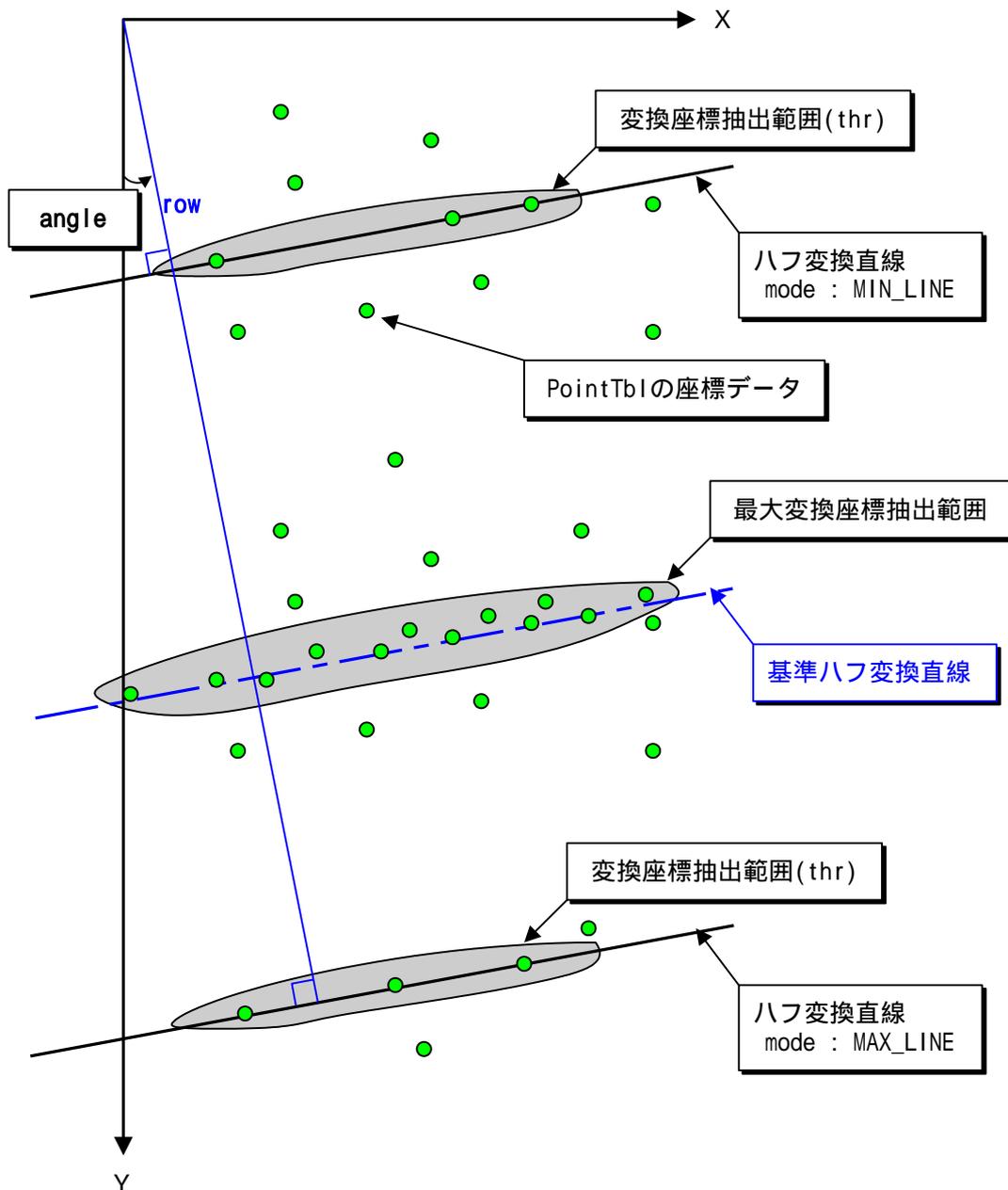
エラー番号	エラー原因
20	パラメータエラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

POINTTBL、LINERANGE、HOUGHLINEのフォーマットは、GetHoughLineの詳細情報の欄を参照して下さい。

本コマンドは、まず変換抽出数量が最大となる直線を抽出し、その抽出した直線を基準ハフ変換直線にします。そして、基準ハフ変換直線の角度で変換抽出数のしきい値(thr)以上の直線を検索します。



## GetSideHoughLine

### 離散データ座標からの直線抽出(角度固定)

#### 機能

PointTblに格納された点列をangleで設定された角度でハフ変換を行い、thrで設定された変換抽出数量しきい値以上のポイントの直線を求めます。また、modeにより直線のrowの最小値または、最大値を検出します。なお、本コマンドを使用する場合のバウンダリチェック及び領域の絞り込みについては、GetHoughLine()コマンドの詳細情報を参照して下さい。

#### コーディング

```
int
GetSideHoughLine(
    DEVID      devid ,
    POINTTBL   *PointTbl ,
    float      angle ,
    HOUGHLINE  *HoughLine ,
    int        thr ,
    int        mode
)
```

#### コメント

デバイスID  
X, Y座標配列  
検索角度  
ハフ変換直線算出  
変換抽出数量しきい値  
ライン検出モード

#### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*PointTbl**  
X, Y座標配列(入力)

**angle**  
検索角度(°)

**\*HoughLine**  
ハフ変換直線算出(出力)

**thr**  
しきい値

**mode**  
ライン検出モード(入力)

検出モード	対応数	内容
MIN_LINE	0	rowの最小値を検出
MAX_LINE	1	rowの最大値を検出

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。抽出点列数。
-1	異常終了
INVALID_LINE_DATA(-3)	直線のポイント数不足による直線抽出エラー

異常終了時、下記の原因のコマンドエラーが考えられます。

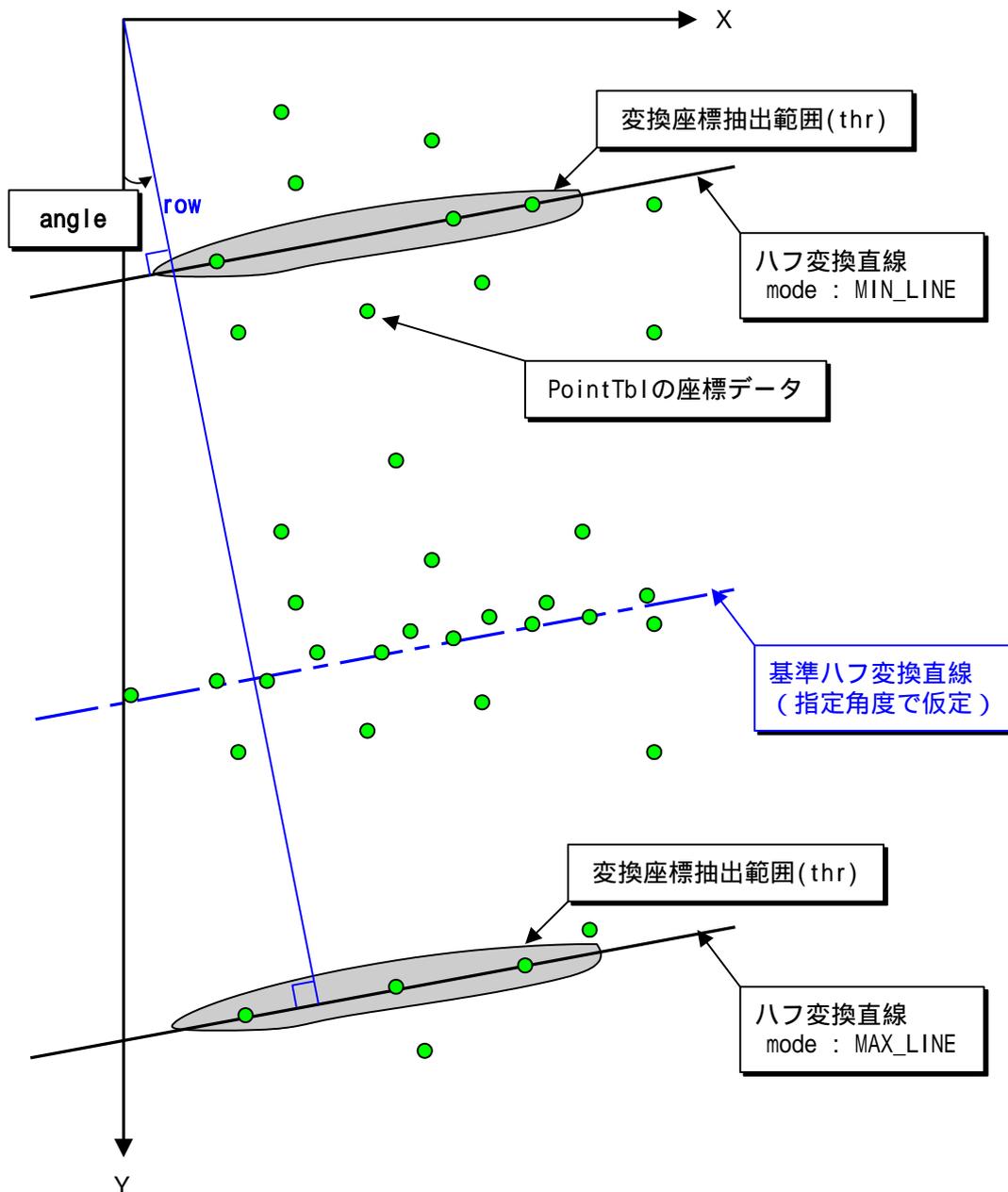
エラー番号	エラー原因
20	パラメータエラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

POINTTBL、HOUGHLINEのフォーマットは、GetHoughLineの詳細情報の欄を参照して下さい。

本コマンドは、angleで指定された角度で直線を基準ハフ変換直線を仮定します。そして、基準ハフ変換直線の角度で変換抽出数のしきい値(thr)以上の直線を検索します。



# GetCrossPoint

## ハフ変換直線の交点座標算出

### 機能

ハフ変換 2 直線の交点座標の算出を行います。

### コーディング

```
int
GetCrossPoint(
    DEVID
    HOUGHLINE
    HOUGHLINE
    FPOINT
)
```

```
devID ,
*xline ,
*yline ,
*point
```

### コメント

デバイス ID  
ハフ変換直線  
ハフ変換直線  
直線の交点座標

### パラメータ

**devID**

デバイス ID。OpenIPDevコマンドで得られた対応するボード番号のデバイス ID を指定して下さい。

**\*xline**

-45° angle 45° までのハフ変換直線

**\*yline**

45° angle 135° までのハフ変換直線

**\*point**

直線の交点座標

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

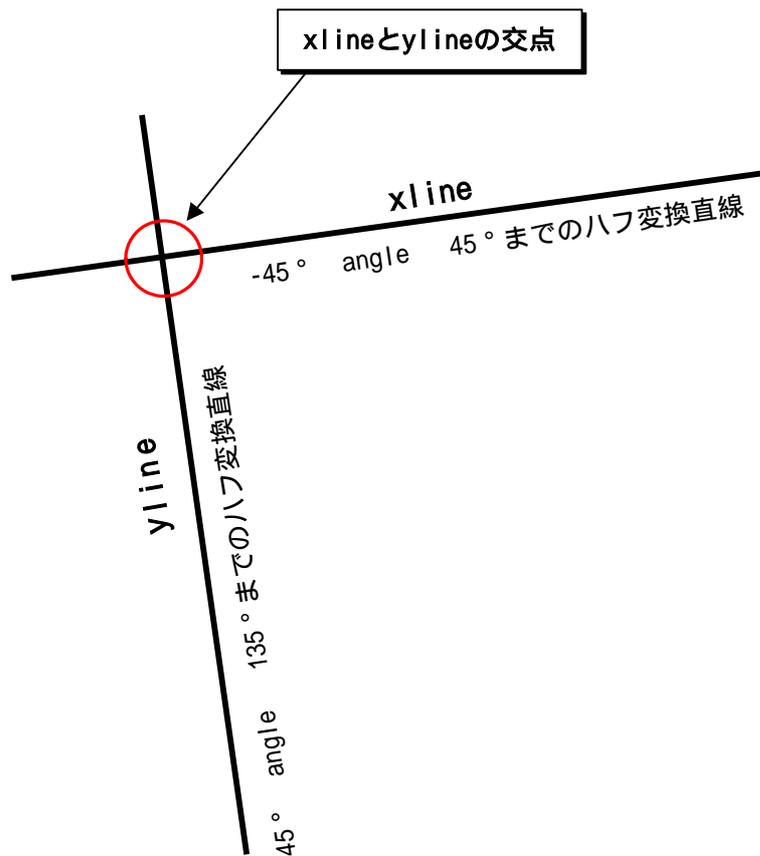
エラー番号	エラー原因
2 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

HOUGHLINEのフォーマットは、GetHoughLineの詳細情報の欄を参照して下さい。  
FPOINTのフォーマットを下記に示します。  
vpxsys.hで宣言しています。

```
typedef struct {  
    float      x;  
    float      y;  
} FPOINT;
```



# GetRectPoint

## ハフ変換直線の矩形交点座標の算出

### 機能

ブロック毎のハフ変換 4 直線から矩形交点座標の算出を行います。

### コーディング

```
int
GetRectPoint(
    DEVID          devID ,
    HOUGHLINE     *block0 ,
    HOUGHLINE     *block1 ,
    HOUGHLINE     *block2 ,
    HOUGHLINE     *block3 ,
    FRECTPOINT    *RectPoint
)

```

### コメント

デバイス I D  
ハフ変換直線  
ハフ変換直線  
ハフ変換直線  
ハフ変換直線  
直線の交点座標

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**\*block0**  
-45° angle 45°までのブロック 0 のハフ変換直線

**\*block1**  
-45° angle 45°までのブロック 1 のハフ変換直線

**\*block2**  
45° angle 135°までのブロック 2 のハフ変換直線

**\*block3**  
45° angle 135°までのブロック 3 のハフ変換直線

**\*RectPoint**  
直線の交点座標

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 0 0	0 除算エラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

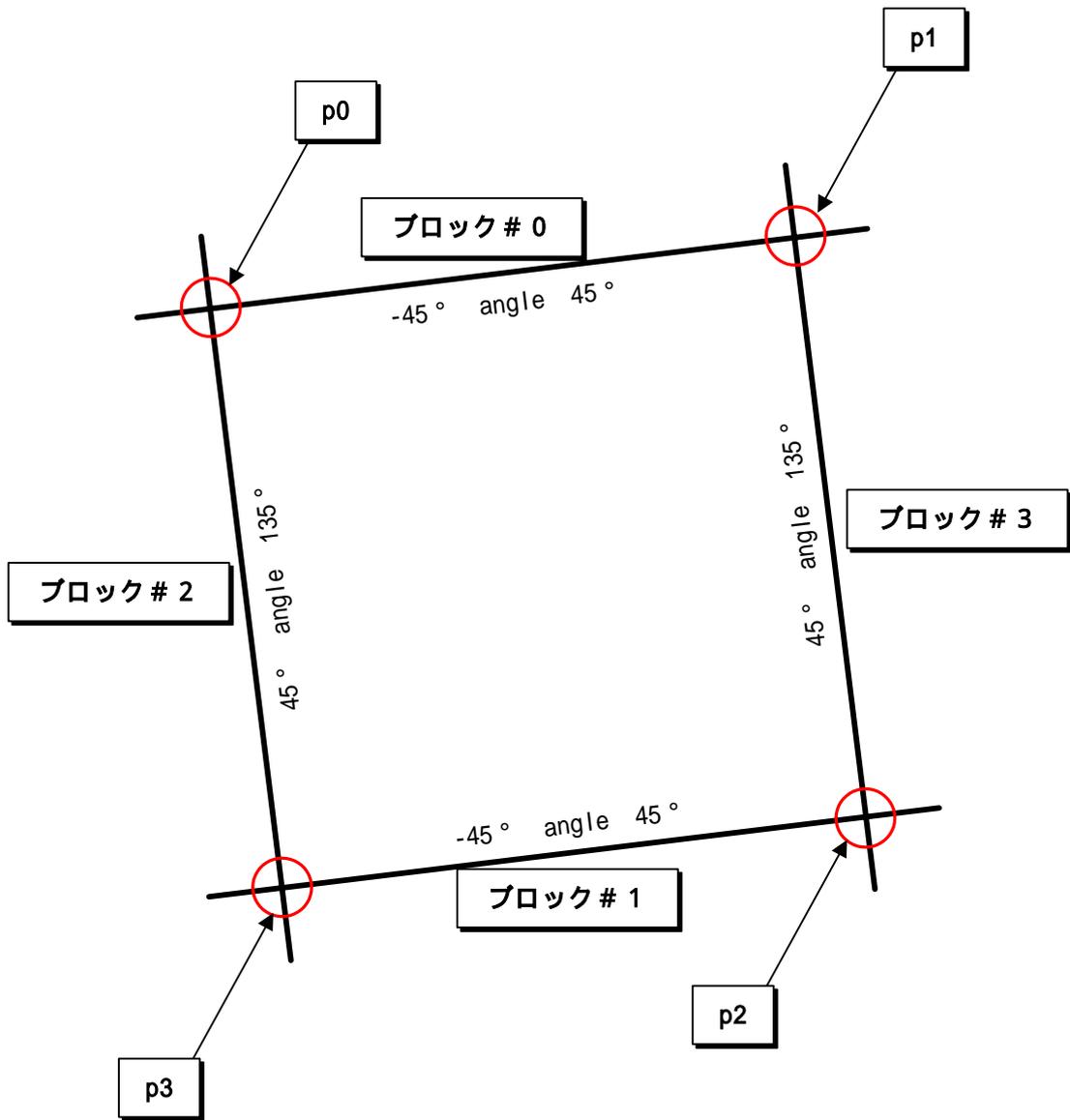
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

HOUGHLINEのフォーマットは、GetHoughLine()の詳細情報の欄を参照して下さい。  
 FRECTPOINTのフォーマットを下記に示します。  
 vpxsys.hで宣言しています。

```
typedef struct {
    FPOINT      p0;
    FPOINT      p1;
    FPOINT      p2;
    FPOINT      p3;
} FRECTPOINT;
```

FPOINTのフォーマットは、GetCrossPoint()の詳細情報の欄を参照して下さい。



# GetRectCenter

## 矩形頂点中心座標の算出

### 機能

ブロック毎の頂点中心座標から矩形中心座標の算出を行います。

### コーディング

```
int
GetRectCenter(
    DEVID  devID ,
    FPOINT *block0 ,
    FPOINT *block1 ,
    FPOINT *block2 ,
    FPOINT *block3 ,
    FPOINT *CenterPoint
)
```

### コメント

デバイス I D  
頂点中心座標  
頂点中心座標  
頂点中心座標  
頂点中心座標  
矩形中心座標

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**\*block0**  
ブロック 0 の頂点中心座標

**\*block1**  
ブロック 1 の頂点中心座標

**\*block2**  
ブロック 2 の頂点中心座標

**\*block3**  
ブロック 3 の頂点中心座標

**\*CenterPoint**  
矩形中心座標

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

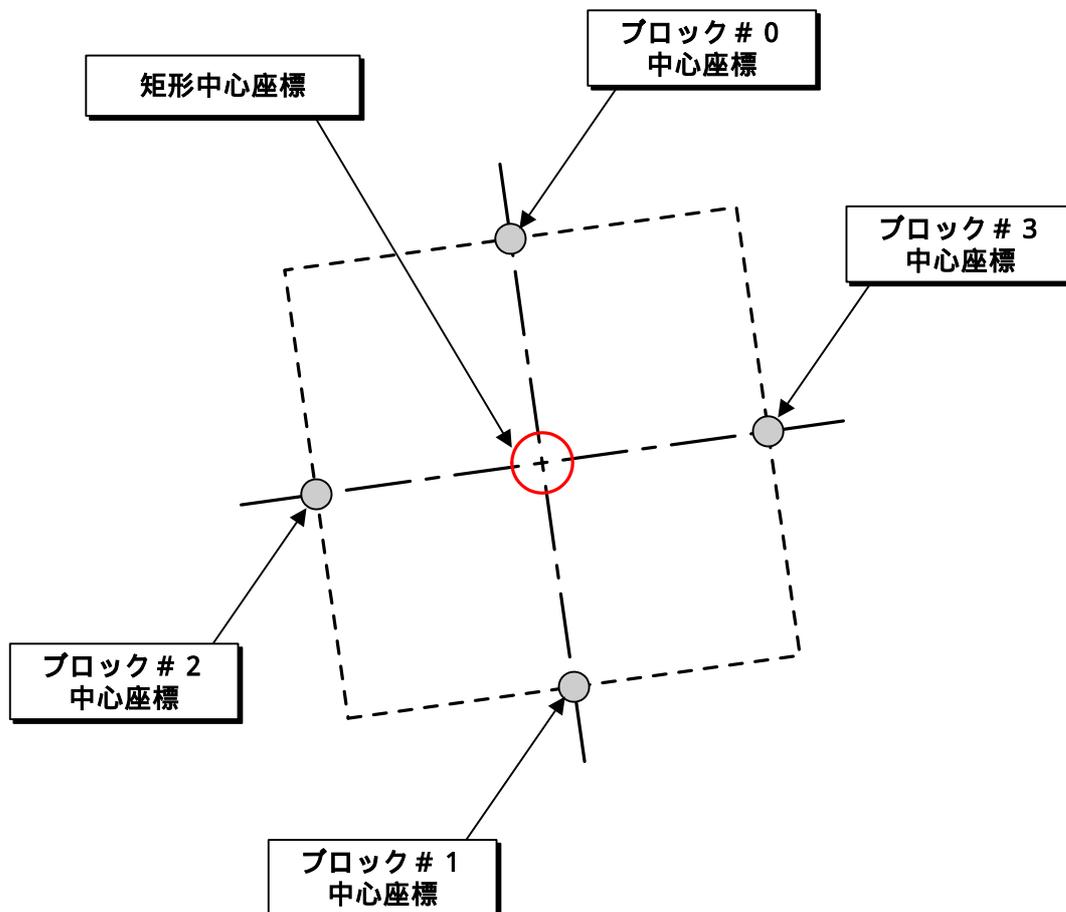
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

FPOINTのフォーマットは、GetCrossPointの詳細情報の欄を参照して下さい。



## GetAnglePoint4

## 矩形頂点間中心座標4ポイントからの角度算出

### 機能

ブロック毎の頂点中心座標 4 ポイントから矩形の傾き角度の算出を行います。

### コーディング

```
int
GetAnglePoint4(
    DEVID  devID ,
    FPOINT *block0 ,
    FPOINT *block1 ,
    FPOINT *block2 ,
    FPOINT *block3 ,
    float  *angle
)
```

### コメント

デバイス I D  
頂点中心座標  
頂点中心座標  
頂点中心座標  
頂点中心座標  
矩形の傾き角度

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**\*block0**  
ブロック 0 の頂点中心座標

**\*block1**  
ブロック 1 の頂点中心座標

**\*block2**  
ブロック 2 の頂点中心座標

**\*block3**  
ブロック 3 の頂点中心座標

**\*angle**  
矩形の傾き角度

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

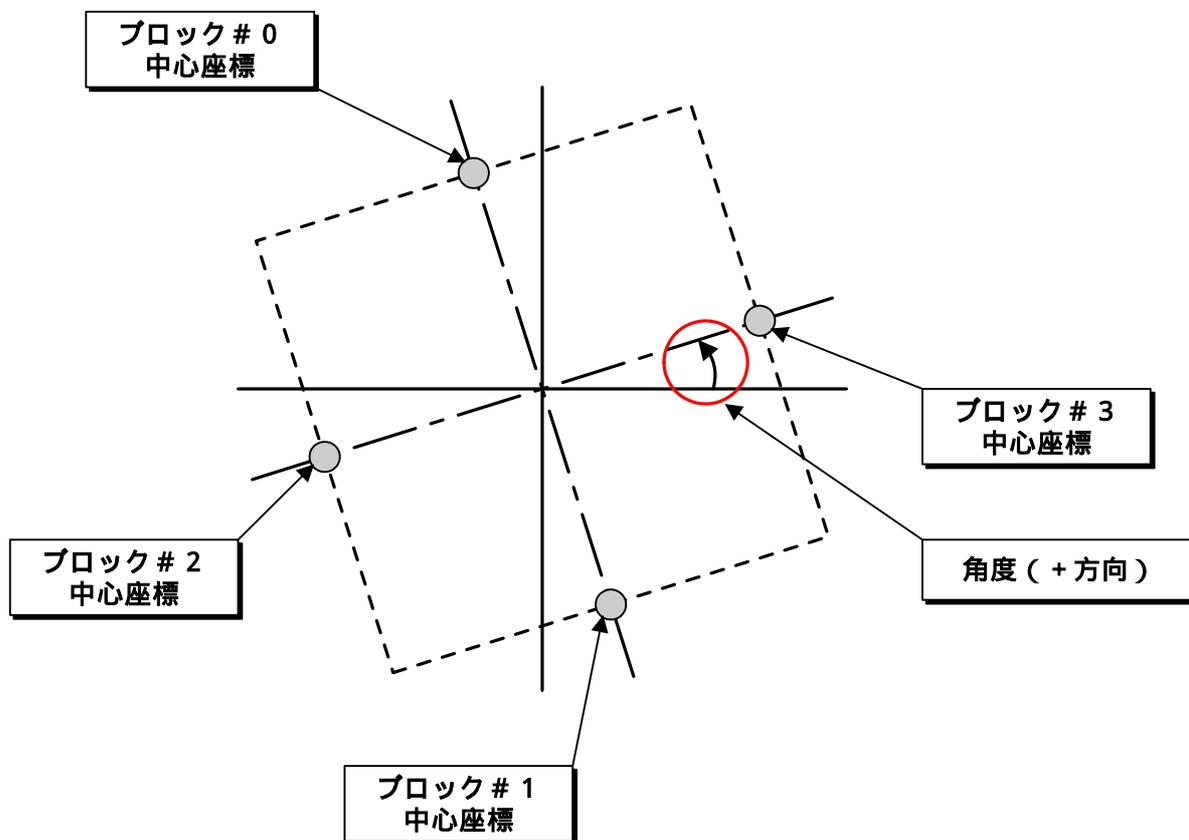
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

FPOINTのフォーマットは、GetCrossPoint()の詳細情報の欄を参照して下さい。



## GetAnglePoint2

### 矩形頂点間中心座標2ポイントからの角度算出

#### 機能

ブロック毎の頂点中心座標 2 ポイントから矩形の傾き角度の算出を行います。

#### コーディング

```
int
GetAnglePoint2(
    DEVID    devID ,
    FPOINT   *block0 ,
    FPOINT   *block1 ,
    float    *angle
)
```

#### コメント

デバイス ID  
 ブロック # 0 , # 2 の頂点中心座標  
 ブロック # 1 , # 3 の頂点中心座標  
 矩形の傾き角度

#### パラメータ

**devID**  
 デバイス ID。OpenIPDevコマンドで得られた対応するボード番号のデバイス ID を指定して下さい。

**\*block0**  
 ブロック # 0 , # 2 の頂点中心座標

**\*block1**  
 ブロック # 1 , # 3 の頂点中心座標

**\*angle**  
 矩形の傾き角度

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

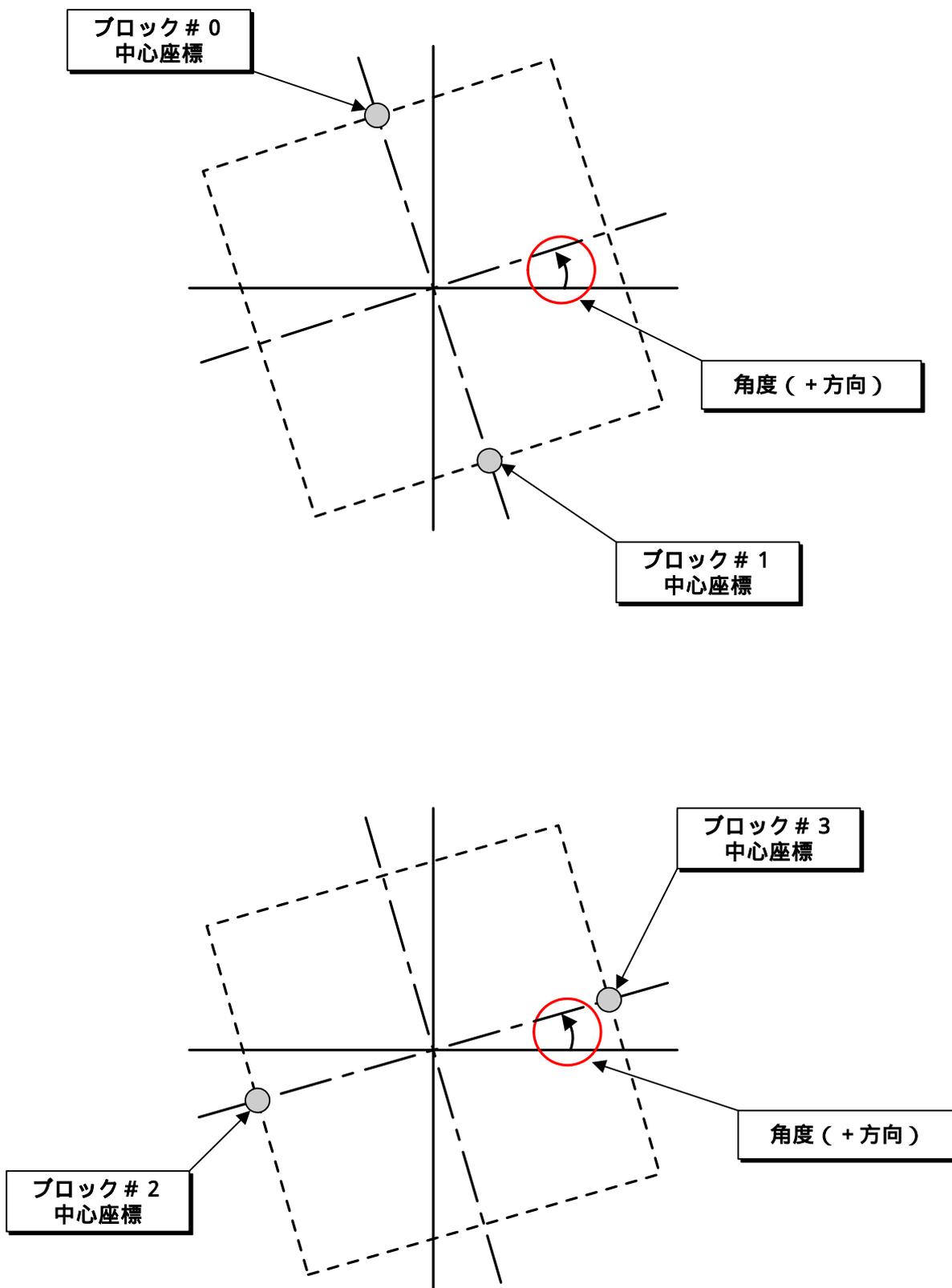
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

FPOINTのフォーマットは、GetCrossPoint()の詳細情報の欄を参照して下さい。



# ProjectLine

## 傾斜矩形領域の投影

### 機能

ImgIDで指定された画面番号の画像メモリから、投影領域 (sx、sy) を開始点座標、(ex、ey) を終了点座標とする直線上をleng幅で投影したデータを、データテーブルに読み出します。

### コーディング

```
int
ProjectLine(
    DEVID      devid ,
    int        imgID ,
    LINETBL    *LineTbl ,
    int        mode ,
    LINEWINDOW *LineWindow
)
```

### コメント

デバイスID  
画面番号  
読み出しデータテーブル  
モード  
ラインウィンドウ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
処理画面の画面番号

**\*LineTbl**  
読み出しデータテーブル

**mode**  
モード (現在未使用)。「0」を設定して下さい。

**\*LineWindow**  
投影領域 (ラインウィンドウ)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 2	パラメータエラー
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

LINETBL、LINEWINDOWのフォーマットおよび内容を下記に示します。  
vpxsys.hで宣言しています。

```
typedef struct {
    short    lower;
    short    num;
    short    data[512];
} LINETBL;
```

LINETBL	説明
lower	開始点座標 ( sx / sy )
num	有効データ数 ( ex - sx + 1 / ey - sy + 1 )
data[512]	データ

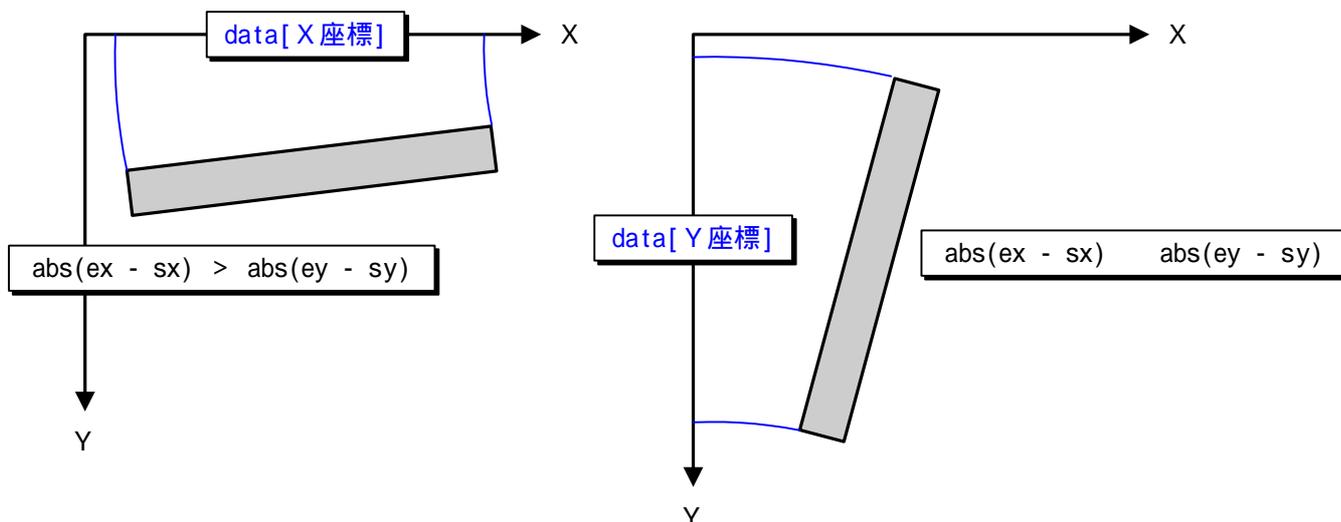
```
typedef struct {
    short    sx;
    short    sy;
    short    ex;
    short    ey;
    short    leng;
    short    opt;
} LINEWINDOW;
```

LINEWINDOW	説明
sx	開始点座標 X
sy	開始点座標 Y
ex	終了点座標 X
ey	終了点座標 Y
leng	投影幅
opt	オプション ( 「 0 」 を設定して下さい ) 。 CaliperLPtoSP(), EdgeFinderLPtoSP() コマンドで意味を持ちます。詳細はそちらを参照して下さい。

読み出しデータは、data[0]からdata[num-1]に書き込まれます。そのときのdata[0]の投影座標がlowerになります。また、設定されたdata[]配列のポインタ ( 添字 ) は

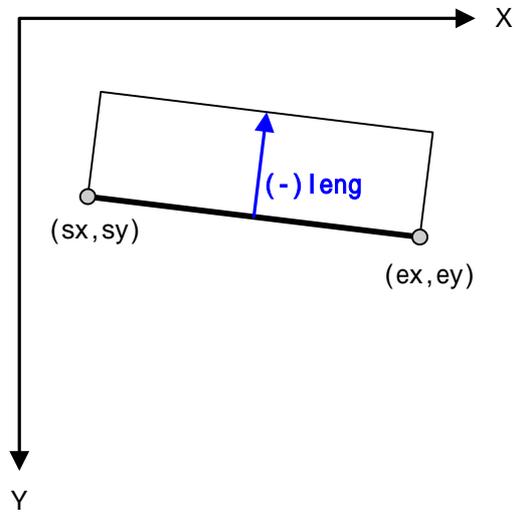
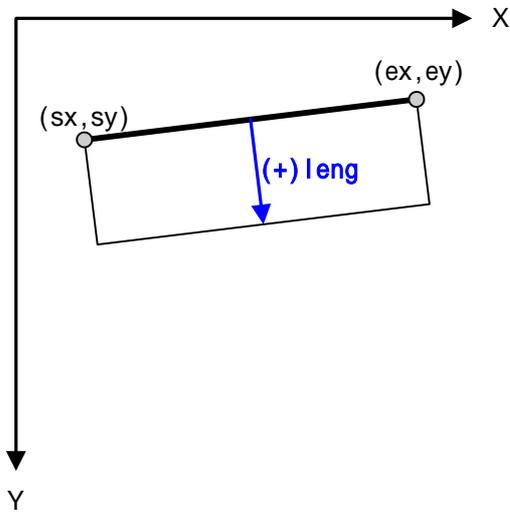
```
abs(ex - sx) > abs(ey - sy) : data[ X座標 ]
abs(ex - sx)  abs(ey - sy) : data[ Y座標 ]
```

になります。



投影幅  $leng$  は符号により以下のように投影する方向が変わります。

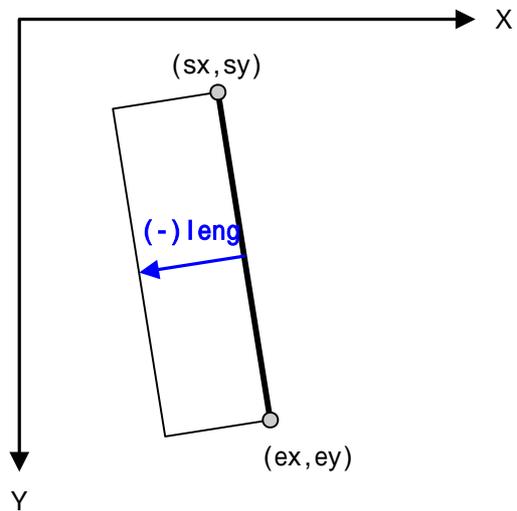
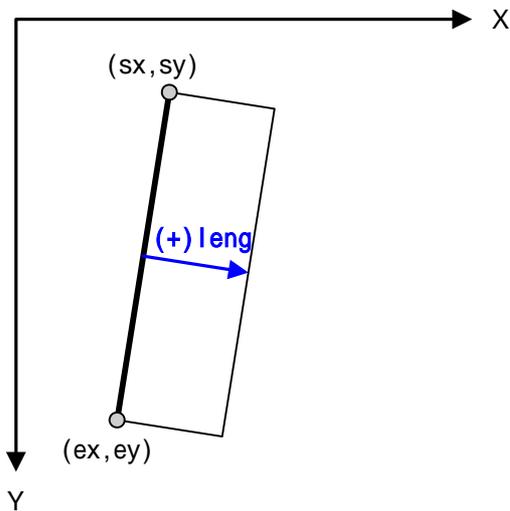
$$\text{abs}(ex - sx) > \text{abs}(ey - sy)$$



#### ラインテーブルに出力される値

LineTbl.lower = sx  
 LineTbl.num = ex - sx + 1  
 LineTbl.data[] = X軸上に投影されるデータ

$$\text{abs}(ex - sx) < \text{abs}(ey - sy)$$



#### ラインテーブルに出力される値

LineTbl.lower = sy  
 LineTbl.num = ey - sy + 1  
 LineTbl.data[] = Y軸上に投影されるデータ

# LineEdgeFilter

## エッジ抽出

### 機能

ProjectTblに設定された画像データ及び投影データの微分を行い、そのエッジ成分を抽出し、EdgeTblに出力します。

### コーディング

```
int
LineEdgeFilter(
    DEVID      devid ,
    LINETBL    *ProjectTbl ,
    LINETBL    *EdgeTbl
)
```

### コメント

デバイスID  
投影データテーブル  
出力エッジテーブル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*ProjectTbl

投影データテーブル

\*EdgeTbl

出力エッジテーブル

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

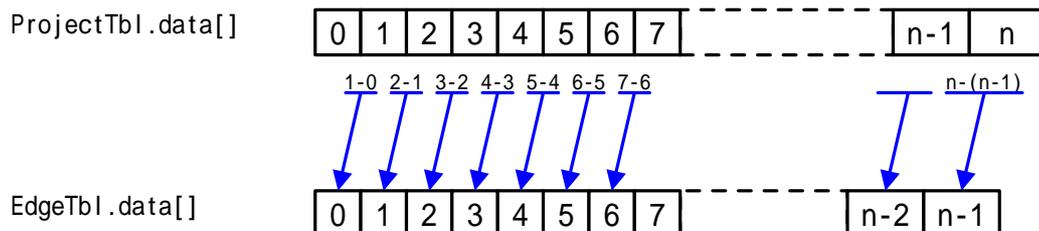
エラー番号	エラー原因
2 0	パラメータエラー
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

LineTblのフォーマット及び内容は、ProjectLine()の詳細情報の欄を参照して下さい。

本コマンドは、以下の演算を行いEdgeTblに出力します。



エッジテーブルの有効データ数

EdgeTbl.num → ProjectTbl.num - 1

エッジテーブルの先頭アドレス

EdgeTbl.lower → ProjectTbl.lower

# LineEdgeFilterExt

## フィルタ付きエッジ抽出

### 機能

ProjectTblに設定された画像データ及び投影データをCOEFFテーブルの空間オペレータでフィルタリンクし、EdgeTblに出力します。

### コーディング

```
int
LineEdgeFilterExt(
    DEVID      devid ,
    LINETBL    *ProjectTbl ,
    LINETBL    *EdgeTbl ,
    int        offset ,
    int        scale ,
    int        size ,
    short      *COEFF
)
```

### コメント

デバイスID  
 投影データテーブル  
 出力エッジテーブル  
 オフセット  
 スケールダウン量  
 荷重係数のサイズ  
 荷重係数テーブルへのポインタ

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*ProjectTbl**

投影データテーブル

**\*EdgeTbl**

出力エッジテーブル

**offset**

オフセット。出力エッジテーブルの有効データが始まる位置を指定して下さい。また、先頭から指定した分だけ「0」が設定されます。通常は、 $\text{INT}(\text{size} / 2 - 0.5)$ を設定して下さい。

**scale**

スケールダウン量。荷重係数との積和演算後の除算の除数ですので「0」以外の値を設定して下さい。

**size**

荷重係数のサイズ

**\*COEFF**

荷重係数テーブルへのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

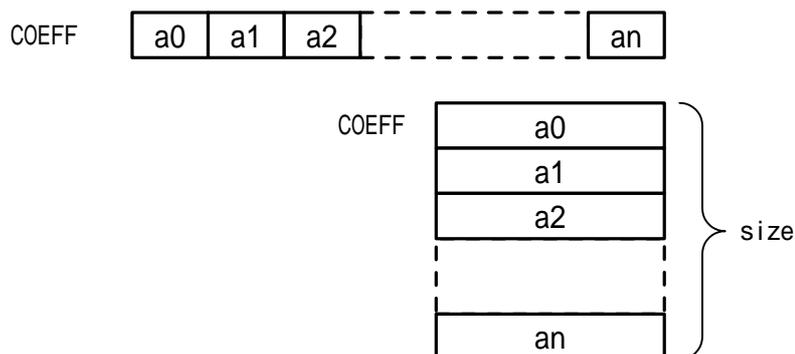
エラー番号	エラー原因
2 0	パラメータエラー
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

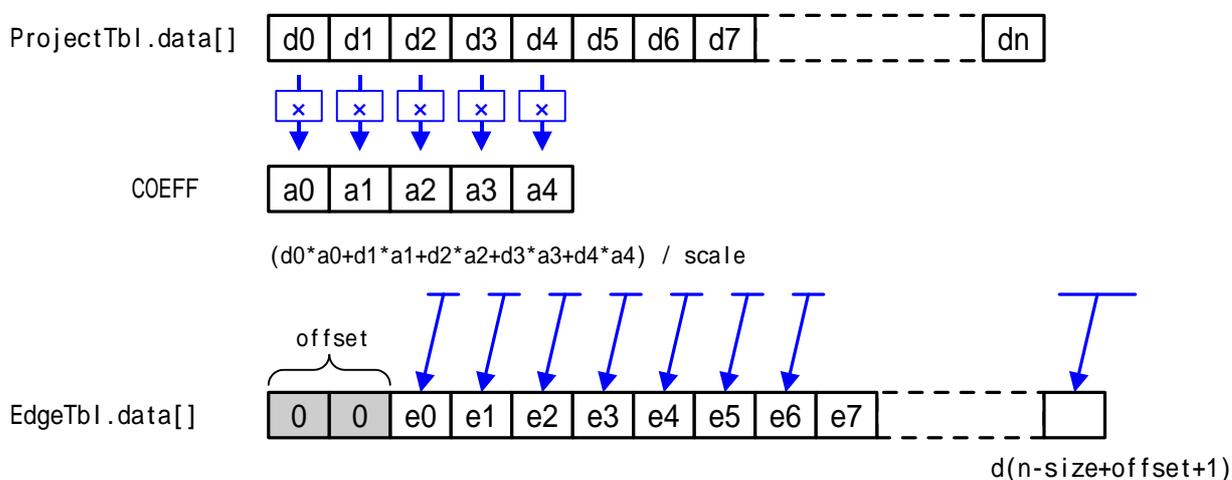
LineTblのフォーマット及び内容は、ProjectLine()の詳細情報の欄を参照して下さい。

荷重係数 COEFF のフォーマットを以下に示します。



本コマンドは、以下の演算を行いEdgeTblに出力します。

荷重係数サイズ size = 5 , offset = 2 の時



エッジテーブルの有効データ数

EdgeTbl.num → ProjectTbl.num - size + offset + 1

エッジテーブルの先頭アドレス

EdgeTbl.lower → ProjectTbl.lower

offset の設定値の目安は、以下のようになります。

$$\text{offset} = \text{INT}(\text{size} / 2 - 0.5)$$

#### 設定例

COEFF    

-1	-1	0	1	1
----	----	---	---	---

size     5

scale    1

offset   2

# LineCaliper

## ラインキャリパ

### 機能

EdgeTblのエッジ成分からscoreをしきい値としてアップエッジとダウンエッジを検出し、抽出モードmodeに従いオブジェクトの開始点、終了点、中心点、幅を一次元論理座標で抽出し、LineCaliperTblに出力します。

### コーディング

```
int
LineCaliper(
    DEVID
    LINETBL
    LINECALIPERTBL
    enum IPCaliperMode
    int
)
```

```
devID ,
*LineTbl ,
*CaliperTbl ,
mode ,
score
```

### コメント

```
デバイスID
エッジ成分ラインテーブル
一次元論理座標のキャリパテーブル
抽出モード
エッジ判定しきい値
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### \*LineTbl (入力)

エッジ成分ラインテーブル

#### \*CaliperTbl (出力)

一次元論理座標のキャリパテーブル

#### mode (入力)

抽出モード

描画モード	対応数	説明
CALIPER_BRIGHT	0	明るいオブジェクトの抽出
CALIPER_DARK	1	暗いオブジェクトの抽出

#### score (入力)

エッジ判定しきい値

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

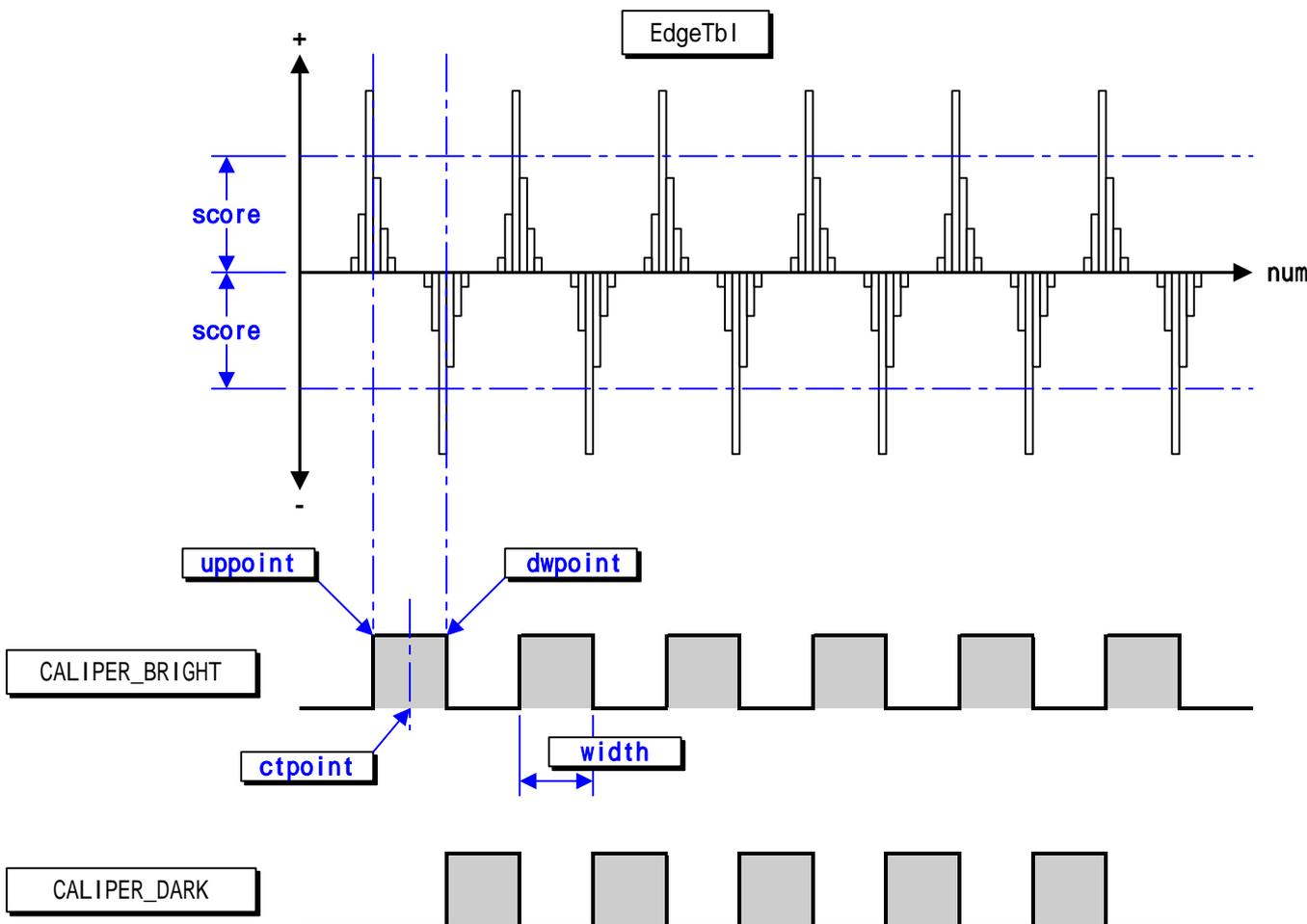
**詳細情報**

LINETBLのフォーマット及び内容は、ProjectLine()の詳細情報の欄を参照して下さい。  
 LINECALIPERTBLのフォーマット及び内容を、下記に示します。  
 vpxsys.hで宣言しています。

```
typedef struct {
    short      num;
    short      opt;
    short      uppoint[256];
    short      dwpoint[256];
    short      ctpoint[256];
    short      width[256];
} LINECALIPERTBL;
```

LINEWINDOW	説明
num	抽出オブジェクトの個数
opt	未使用
uppoint[256]	オブジェクトの開始位置（一次元論理座標）
dwpoint[256]	オブジェクトの終了位置（一次元論理座標）
ctpoint[256]	オブジェクトの中心位置（一次元論理座標）
width[256]	オブジェクトの幅（論理幅）

一次元論理座標キャリパテーブル（LINECALIPERTBL）の座標の詳細を以下に示します。



# CaliperLPtoSP

## ライン座標から平面座標への変換

### 機能

LineCaliperコマンドで得られた一次元論理座標のLineCaliperTblのデータをCalibテーブルに設定されたキャリブレーションデータとLineWindowテーブルに設定されたウィンドウテーブルから二次元論理座標に変換し、CaliperTblに設定します。

### コーディング

```
int
CaliperLPtoSP(
    DEVID          ,   devID ,
    LINECALIPERTBL,   *LineCaliperTbl ,
    CALIPERTBL     ,   *CaliperTbl ,
    LINEWINDOW     ,   *LineWindow ,
    CALIBTBL       ,   *Calib
)

```

### コメント

デバイスID  
一次元論理座標のラインキャリパテーブル  
二次元論理座標のキャリパテーブル  
投影を行ったラインウィンドウ  
キャリブレーションデータテーブル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*LineCaliperTbl (入力)

一次元論理座標のラインキャリパテーブル。LineCaliper()コマンドで得られたラインキャリパテーブルをそのまま設定します。

\*CaliperTbl (出力)

二次元論理座標のキャリパテーブル。一次元論理座標のラインキャリパテーブルが二次元論理座標に変換され出力されます。

\*LineWindow (入力)

投影を行ったラインウィンドウ。ProjectLine()コマンドで指定したラインウィンドウを設定して下さい。

\*Calib (入力)

キャリブレーションデータテーブル。画素サイズを設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

LINECALIPERTBLのフォーマット及び内容は、LineCaliper()の詳細情報の欄を参照して下さい。  
CALIPERTBL、CALIBTBLのフォーマットおよび内容を下記に示します。  
vpxsys.hで宣言しています。

```
typedef struct {
    LINEWINDOW      LineWindow;
    CFACTOR         calib;
    DFACTOR         disp;
    short           num;
    short           mode;
    FPOINT          uppoint[256];
    FPOINT          dwpoint[256];
    FPOINT          ctpoint[256];
    float           width[256];
    char            flag[256];
} CALIPERTBL;
```

### 二次元論理キャリパテーブル

CALIPERTBL	説明
LineWindow	ラインウィンドウ (LineWindowパラメータからコピーされます)
calib	物理座標変換係数
disp	論理座標変換係数
num	抽出ポイント数
mode	0 : abs(dx) > abs(dy) 1 : abs(dx) <= abs(dy)
uppoint[256]	オブジェクトの開始ポイント (二次元論理座標)
dwpoint[256]	オブジェクトの終了ポイント (二次元論理座標)
ctpoint[256]	オブジェクトの中心ポイント (二次元論理座標)
width[256]	オブジェクトの幅 (物理幅 単位: mm)
flag[256]	オブジェクトの有効フラグ = 1 : 有効 = 0 : 無効

```
typedef struct {
    float           cx;
    float           cy;
    float           angle;
    float           pixelx;
    float           pixely;
} CALIBTBL;
```

### キャリブレーションテーブル

CALIBTBL	説明
cx	カメラ中心X座標
cy	カメラ中心Y座標
angle	カメラ傾き
pixelx	画素サイズX (mm)
pixely	画素サイズY (mm)

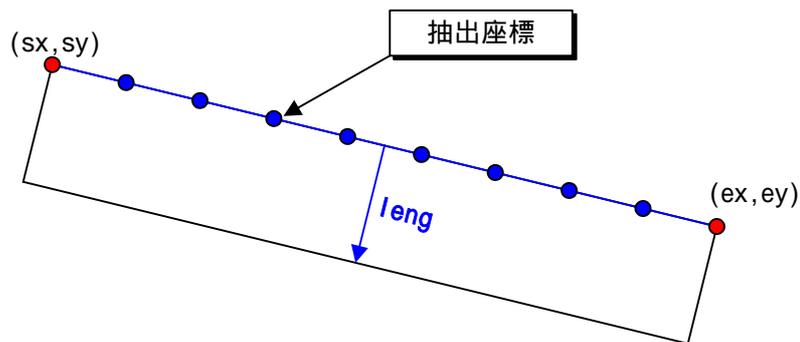
```
typedef struct {
    short    sx;
    short    sy;
    short    ex;
    short    ey;
    short    leng;
    short    opt;
} LINEWINDOW;
```

## ラインウィンドウテーブル

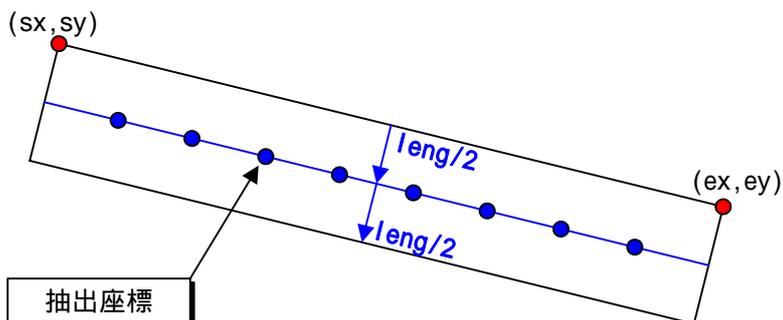
LINEWINDOW	説明								
sx	開始点座標 X								
sy	開始点座標 Y								
ex	終了点座標 X								
ey	終了点座標 Y								
leng	投影幅								
opt	オプション <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>オプション</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>ラインウィンドウ直線上の座標を抽出</td> </tr> <tr> <td>1</td> <td>ラインウィンドウ直線と leng 幅との中心座標を抽出</td> </tr> <tr> <td>2</td> <td>ラインウィンドウ直線と leng 幅間隔の座標を抽出</td> </tr> </tbody> </table> 詳細を以下に示します。	オプション	内容	0	ラインウィンドウ直線上の座標を抽出	1	ラインウィンドウ直線と leng 幅との中心座標を抽出	2	ラインウィンドウ直線と leng 幅間隔の座標を抽出
オプション	内容								
0	ラインウィンドウ直線上の座標を抽出								
1	ラインウィンドウ直線と leng 幅との中心座標を抽出								
2	ラインウィンドウ直線と leng 幅間隔の座標を抽出								

## LINEWINDOWのメンバ変数 opt の詳細

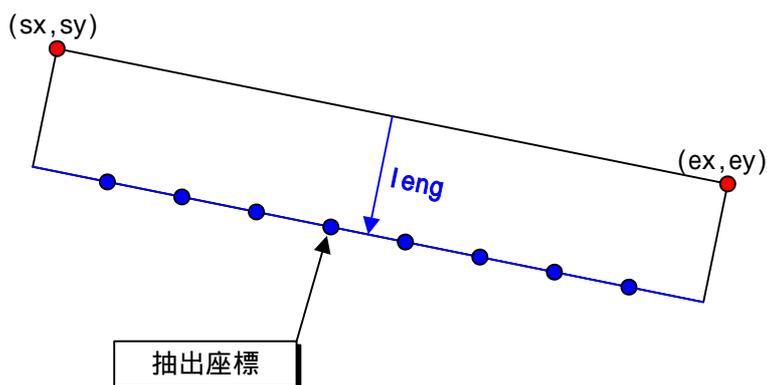
opt : 0



opt : 1



opt : 2



DFACTOR、FPOINTのフォーマットを下記に示します。  
vpxsys.hで宣言しています。

```
typedef struct {
    float    sin;
    float    cos;
    float    Factor;
} DFACTOR;
```

```
typedef struct {
    float    x;
    float    y;
} FPOINT;
```

## 論理座標変換係数

DFACTOR	説明
sin	mode(0): $\sin(\text{atan}(dy/dx))$ // mode(1): $\sin(\text{atan}(dx/dy))$
cos	mode(0): $\cos(\text{atan}(dy/dx))$ // mode(1): $\cos(\text{atan}(dx/dy))$
Factor	mode(0): $\cos(\text{atan}(dy/dx))$ // mode(1): $\sin(\text{atan}(dx/dy))$

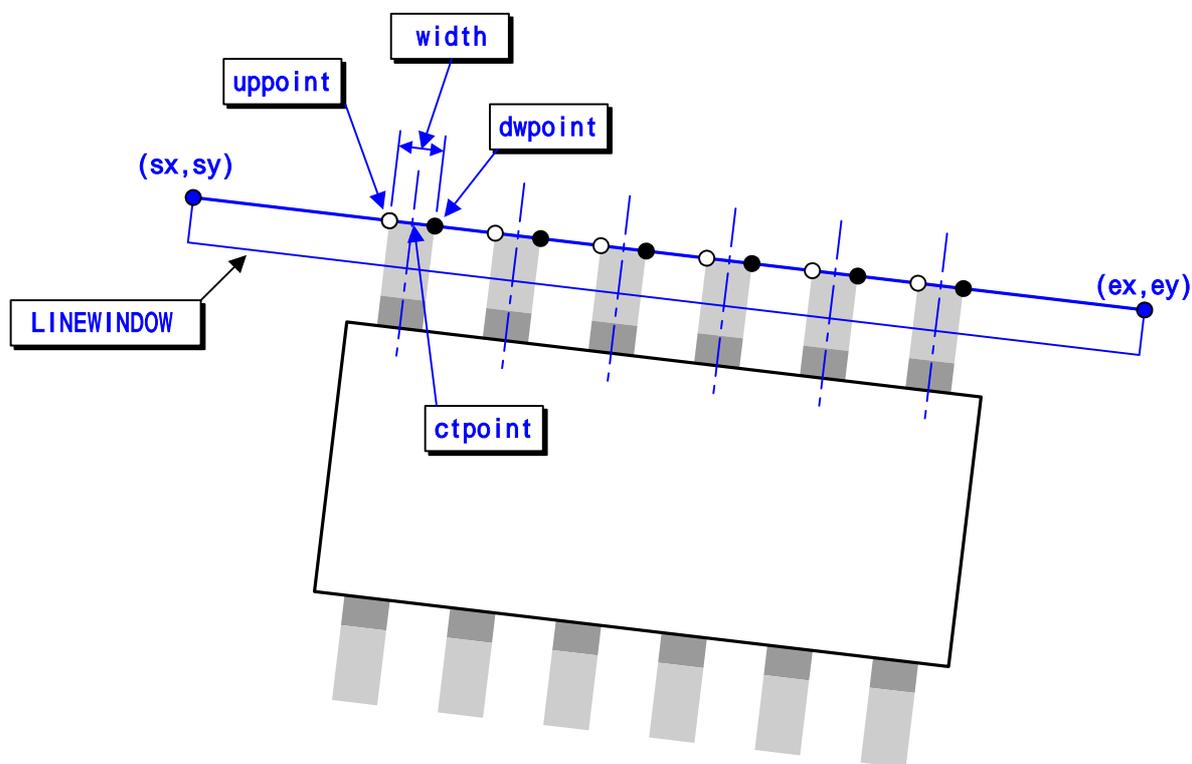
CFACTORのフォーマットを下記に示します。  
vpxsys.hで宣言しています。

```
typedef struct {
    float    pixelx;
    float    pixely;
    float    angle;
    float    sinFactor;
    float    cosFactor;
    float    divFactor;
    float    divSinFactor;
    float    divCosFactor;
} CFACTOR;
```

物理座標変換係数

CFACTOR	説明
pixelx	Calib.pixelxのコピー
pixely	Calib.pixelyのコピー
angle	Calib.angleのコピー
sinFactor	mode(0): pixelx * calibSin // mode(1): pixely * calibSin
cosFactor	mode(0): pixelx * calibCos // mode(1): pixely * calibCos
divFactor	mode(0): pixelx / calibCos // mode(1): pixely / calibSin
divSinFactor	mode(0): calibSin / pixelx // mode(1): calibSin / pixely
divCosFactor	mode(0): calibCos / pixelx // mode(1): calibCos / pixely

二次元論理座標キャリパテーブル (CALIPERTBL) の座標の詳細を以下に示します。



# GetCaliperScore

## エッジ判定しきい値の算出

### 機能

EdgeTblからLineCaliper()に使用するエッジ判定しきい値を算出します。

### コーディング

```
int
GetCaliperScore(
    DEVID      devID ,
    LINETBL    *LineTbl
)
```

### コメント

デバイスID  
エッジ成分ラインテーブル

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*LineTbl**  
エッジ成分ラインテーブル

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	正常終了。エッジ判定しきい値。
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

LINETBLのフォーマット及び内容は、ProjectLine()の詳細情報の欄を参照して下さい。

# SetCaliperWidth

## 投影幅の算出

### 機能

LineWindowの投影幅LineWindow.lengを設定します。LineWindowテーブルのsx、sy、ex、eyとキャリブレーションテーブルCalib、方向mode、物理座標の投影幅widthから論理座標投影幅を算出します。

### コーディング

```
int
SetCaliperWidth(
    DEVID      devid ,
    CALIBTBL   *Calib ,
    LINEWINDOW *LineWindow ,
    float      width ,
    int        mode
)
```

### コメント

デバイスID  
 キャリブレーションデータテーブル  
 ラインウィンドウ  
 投影幅  
 ラインウィンドウ幅の方向

### パラメータ

**devID**  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*Calib**  
 キャリブレーションデータテーブル

**\*LineWindow**  
 ラインウィンドウテーブル

**width**  
 投影幅

**mode**  
 ラインウィンドウ幅の方向  
 0 : 投影幅 +  
 1 : 投影幅 -

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

LINETBLのフォーマット及び内容は、ProjectLine()の詳細情報の欄を、CALIBTBLのフォーマット及び内容は、CaliperLPtoSP()の詳細情報の欄を参照して下さい。

# LineEdgeFinder

## ラインエッジファインダ

### 機能

EdgeTblのエッジ成分から、scoreをしきい値としてエッジを検出し、そのエッジの位置を一次元論理座標で抽出し、LineEdgeFinderTblに出力します。

本コマンドでは、抽出したエッジの強さや極性を個別に抽出できます。

### コーディング

```
int
LineEdgeFinder(
  DEVID          devID ,
  LINETBL        *EdgeTbl ,
  LINEEDGEFINDERTBL *LineEdgeFinderTbl ,
  int            mode ,
  int            minthr ,
  int            minscore ,
  int            score
)
```

### コメント

デバイスID  
エッジ成分ラインテーブル  
一次元論理座標のエッジ抽出テーブル  
抽出モード  
エッジ抽出領域検出2次微分しきい値  
エッジ抽出領域検出微分しきい値  
エッジ判定しきい値

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### \*EdgeTbl (入力)

エッジ成分ラインテーブル

#### \*LineEdgeFinderTbl (出力)

一次元論理座標のラインエッジ抽出テーブル

#### mode (入力)

抽出モード。「0」を設定して下さい。

#### minthr (入力)

エッジ抽出領域を検出するための2次微分しきい値。エッジ抽出の領域は minthr と minscore どちらかの条件が一致した部分になります。

#### minscore (入力)

エッジ抽出領域を検出するための微分しきい値。エッジ抽出の領域は minthr と minscore どちらかの条件が一致した部分になります。

#### score (入力)

エッジ判定しきい値

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

LINETBL, LINEEDGEFINDERTBLのフォーマットおよび内容を下記に示します。  
vpxsys.hで宣言しています。

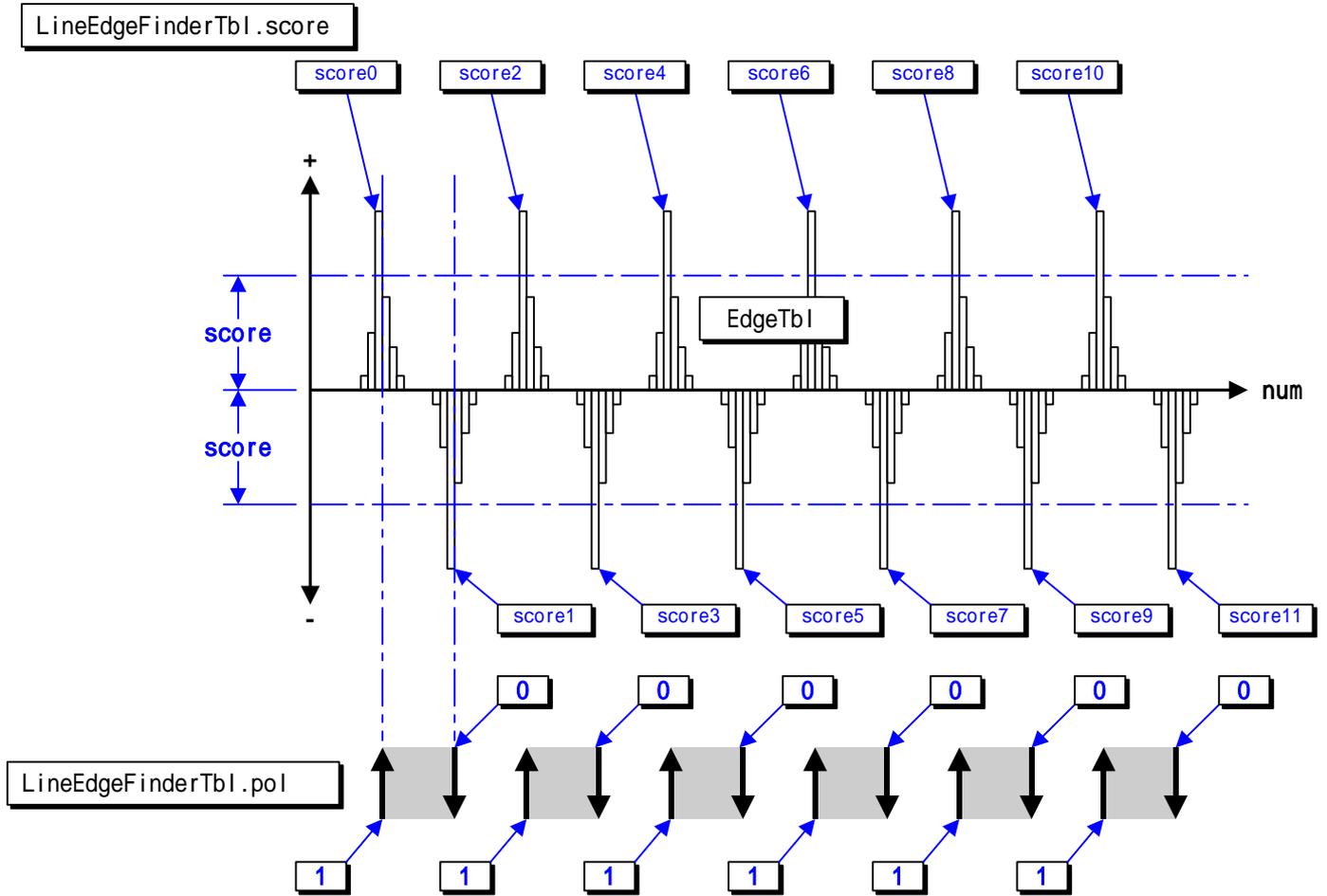
```
typedef struct {
    short    lower;
    short    num;
    short    data[512];
} LINETBL;
```

LINETBL	説明
lower	開始点座標
num	有効データ数
data[512]	エッジデータ

```
typedef struct {
    short    num;
    short    opt;
    float    pos[256];
    short    pol[256];
    short    score[256];
} LINEEDGEFINDERTBL;
```

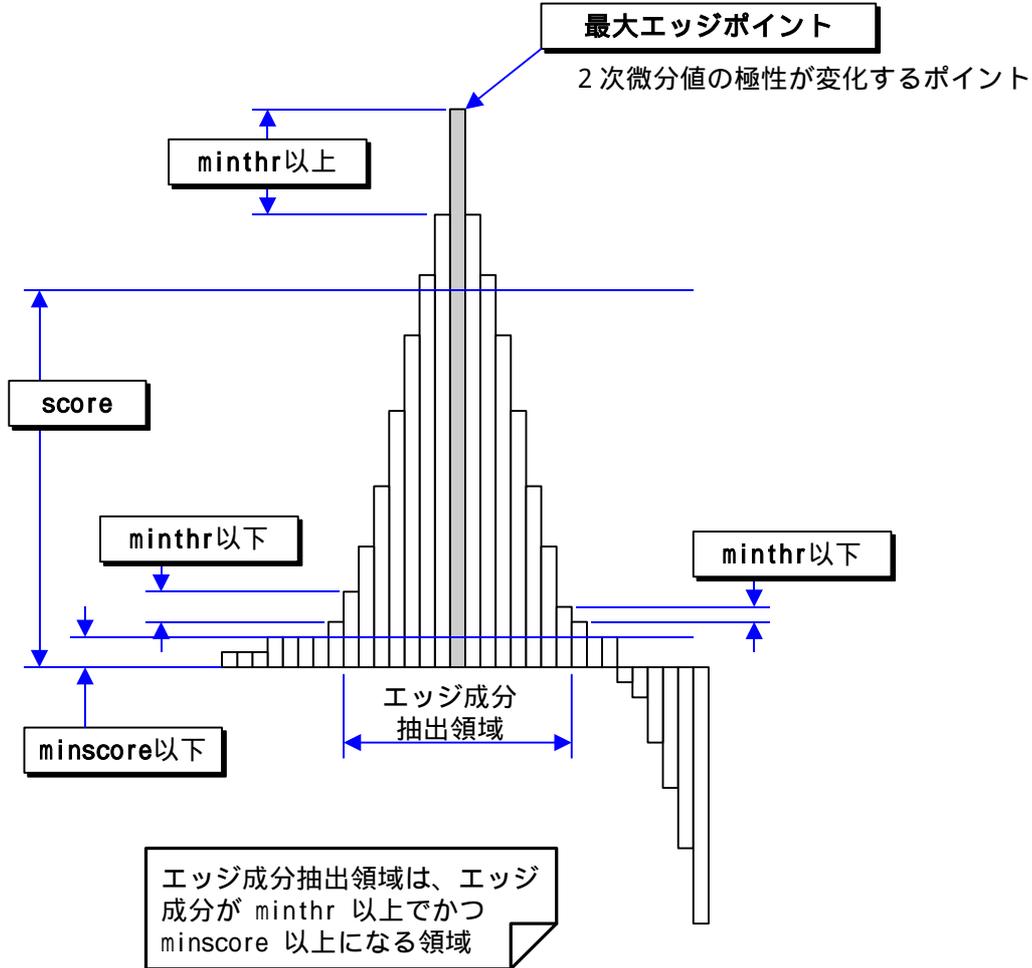
LINEEDGEFINDERTBL	説明						
num	抽出エッジの個数						
opt	未使用						
pos[256]	抽出エッジの位置（一次元論理座標）						
pol[256]	抽出エッジの属性 <table border="1" data-bbox="544 1211 1428 1350"> <thead> <tr> <th>属性</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>明 暗</td> </tr> <tr> <td>1</td> <td>暗 明</td> </tr> </tbody> </table>	属性	内容	0	明 暗	1	暗 明
属性	内容						
0	明 暗						
1	暗 明						
score[256]	抽出エッジのスコア						

一次元論理座標ラインエッジ抽出テーブル (LINEEDGEFINDER\_TBL) の詳細を以下に示します。



エッジ抽出領域検出 2 次微分しきい値 : minthr  
 エッジ抽出領域検出微分しきい値 : minscore

エッジの抽出は、最大エッジポイントから左右にエッジ成分の範囲をサーチし、その範囲内でエッジの中心を求めることにより行います。その場合のエッジの終了点は、エッジ抽出領域検出 2 次微分しきい値 : minthr 以下または、エッジ抽出領域検出微分しきい値 : minscore 以下になるポイントです。



入力されたエッジテーブル（1 次微分値）内のデータをサーチし、score 以上のポイントを探し、次にエッジテーブル内の 2 次微分値を探し、2 次微分値の極性が変化したポイントを探し、最大のエッジポイントとして座標を探します。そのとき、同時に 2 次微分値が minthr 以下になるかどうかをチェックしており、minthr 以下のポイントが出現した時点でそのポイントを探し、最大のエッジポイントとします。次に、最大のエッジポイント座標を基点にエッジテーブル内のデータをサーチし、エッジ成分抽出領域範囲の座標を決定します。このときの条件は 1 次微分値が minscore 以下になるポイントか、または 2 次微分値が minthr 以下になるポイントです。

エッジ抽出領域検出 2 次微分しきい値 : minthr とエッジ抽出領域検出微分しきい値 : minscore に設定する値は、エッジ成分抽出領域を決める際に 2 次微分で行うか 1 次微分で行うかを選択できます。1 次微分でエッジ成分抽出領域を決める場合は、minthr を比較的大きな値にする必要があります。逆に、2 次微分で行う場合は、minscore を大きな値にします。通常は、1 次微分のみで十分です。

エッジ成分抽出領域条件	minthr	minscore	備考
1 次微分のみ	score 以上	0 ~ 10 程度	エッジの立ち上がりが急な場合
2 次微分のみ	0 ~ 10 程度	score 以上	エッジの立ち上がりが平坦な場合
1 次微分、2 次微分両方	0 ~ 10 程度	0 ~ 10 程度	

# EdgeFinderLPtoSP

## ライン座標から平面座標への変換

### 機能

LineEdgeFinder() コマンドで得られた一次元論理座標のLineEdgeFinderTblのデータをLineWindowテーブルに設定されたウィンドウデータから二次元論理座標に変換し、EdgeFinderTblに設定します。

### コーディング

```
int
EdgeFinderLPtoSP(
    DEVID
    LINEEDGEFINDERTBL
    EDGEFINDERTBL
    LINEWINDOW
)
```

```
devID ,
*LineEdgeFinderTbl ,
*EdgeFinderTbl ,
*LineWindow
```

### コメント

デバイスID  
一次元論理座標ラインエッジ抽出テーブル  
二次元論理座標エッジ抽出テーブル  
投影を行ったラインウィンドウ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### \*LineEdgeFinderTbl (入力)

一次元論理座標ラインエッジ抽出テーブル。LineEdgeFinder() コマンドで得られたラインエッジ抽出テーブルをそのまま設定します。

#### \*EdgeFinderTbl (出力)

二次元論理座標エッジ抽出テーブル。一次元論理座標のラインエッジ抽出テーブルが二次元論理座標に変換され出力されます。

#### \*LineWindow (入力)

投影を行ったラインウィンドウ。ProjectLine() コマンドで指定したラインウィンドウを設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
2 0 0	0 除算エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

EDGEFINDERTBL, LINEWINDOWのフォーマットおよび内容を下記に示します。  
vpxsys.hで宣言しています。

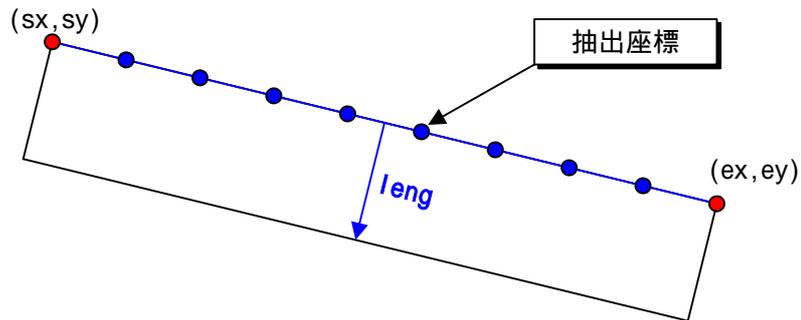
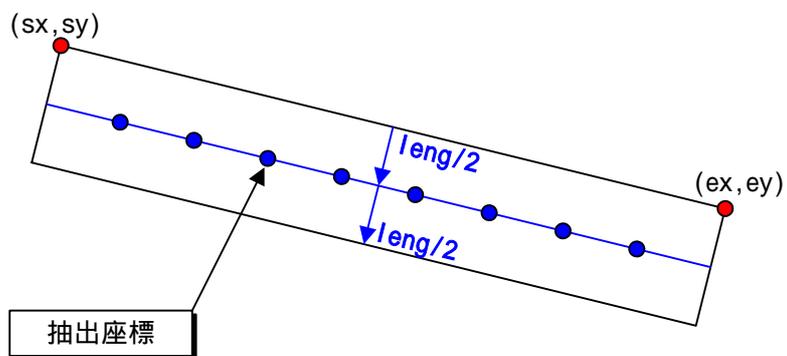
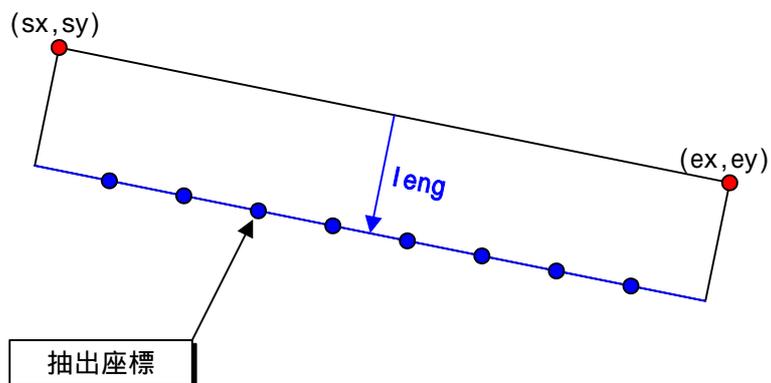
```
typedef struct {
    short      num;
    short      opt;
    FPOINT     pos[256];
    short      pol[256];
    short      score[256];
} EDGEFINDERTBL;
```

```
typedef struct {
    float      x;
    float      y;
} FPOINT;
```

EDGEFINDERTBL	説明						
num	抽出エッジの個数						
opt	未使用						
pos[256]	抽出エッジの位置（二次元論理座標）						
pol[256]	抽出エッジの属性						
	<table border="1"> <thead> <tr> <th>属性</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>明 暗</td> </tr> <tr> <td>1</td> <td>暗 明</td> </tr> </tbody> </table>	属性	内容	0	明 暗	1	暗 明
	属性	内容					
	0	明 暗					
1	暗 明						
score[256]	抽出エッジのスコア						

```
typedef struct {
    short      sx;
    short      sy;
    short      ex;
    short      ey;
    short      leng;
    short      opt;
} LINEWINDOW;
```

LINEWINDOW	説明								
sx	開始点座標 X								
sy	開始点座標 Y								
ex	終了点座標 X								
ey	終了点座標 Y								
leng	投影幅								
opt	オプション								
	<table border="1"> <thead> <tr> <th>オプション</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>ラインウィンドウ直線上の座標を抽出</td> </tr> <tr> <td>1</td> <td>ラインウィンドウ直線とleng幅との中心座標を抽出</td> </tr> <tr> <td>2</td> <td>ラインウィンドウ直線とleng幅間隔の座標を抽出</td> </tr> </tbody> </table>	オプション	内容	0	ラインウィンドウ直線上の座標を抽出	1	ラインウィンドウ直線とleng幅との中心座標を抽出	2	ラインウィンドウ直線とleng幅間隔の座標を抽出
	オプション	内容							
	0	ラインウィンドウ直線上の座標を抽出							
	1	ラインウィンドウ直線とleng幅との中心座標を抽出							
2	ラインウィンドウ直線とleng幅間隔の座標を抽出								
	詳細を次ページに示します。								

LINEWINDOWのメンバ変数 `opt` の詳細`opt : 0``opt : 1``opt : 2`

# InIPPort

## I/Oポートからのデータ入力

### 機能

I/Oポートからデータ入力を行います。

### コーディング

```
int  
InIPPort(  
    DEVID    devID ,  
    int      opt  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

opt

オプション。0を設定して下さい。

### リターン値

リターン値はI/Oポートの値で int 型で返されます。詳細を下表に示します。

リターン値	内容
0 ~ 255	I/Oポートの値(8ビット)
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	パラメータエラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## OutIPPort

## I/Oポートへのデータ出力

### 機能

I/Oポートにデータを出力します。また、オプションでマスクを指定することにより、指定ビットだけを変更することができます。

### コーディング

```
int
OutIPPort(
    DEVID    devID ,
    int      val ,
    int      mask
)
```

### コメント

デバイスID  
ポートの設定値  
マスクオプション

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

val

I/Oポートに設定する値(0~255)

mask

マスクオプションの指定を行います。

オプション	内容
0	全ビット書き換えます。
0	タスク間で排他制御を行い有効な8ビットのうち「1」に対応するビットだけを書き換えます。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

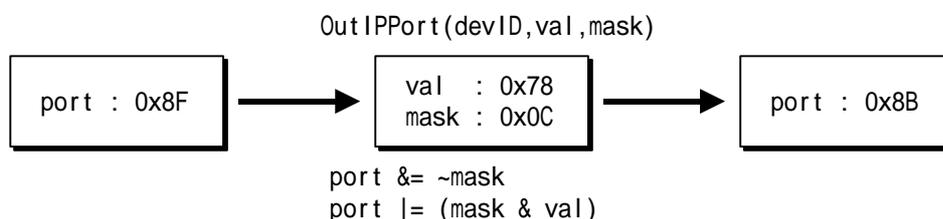
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	パラメータエラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

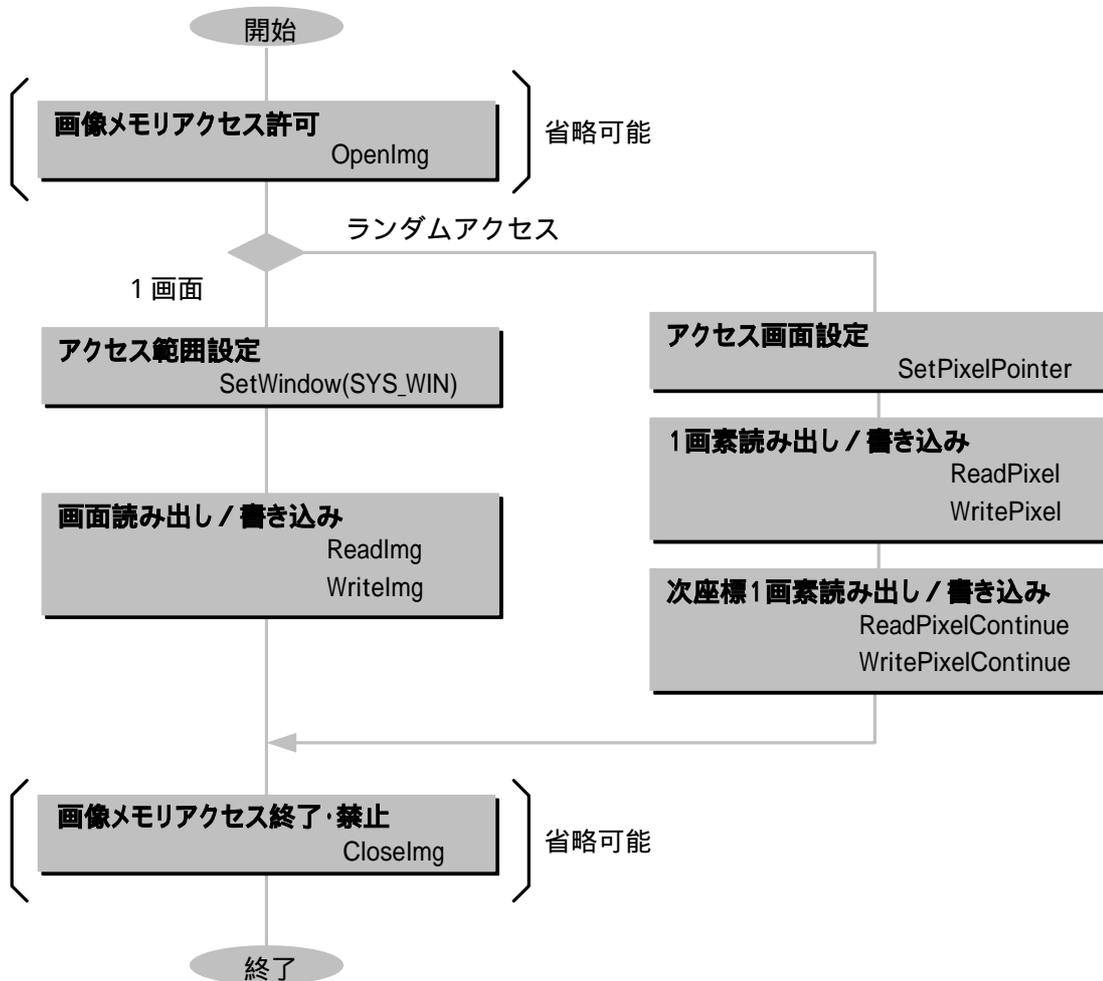
マスクオプションの例



## 28. 画像メモリアクセスコマンド

画像メモリアクセスコマンドでは、画像メモリに対するアクセスが可能です。また、画像メモリにアクセスを行う場合は、下記の手順に従って下さい。

### 画像メモリアクセス作成手順



SVP - 330の画像メモリはUMA方式でオンボードCPUからダイレクトに画像メモリをアクセスできます。詳細は、GetImVirtualAddress()コマンドを参照して下さい。ただし、PC側ではこの方法ではアクセスできないので注意して下さい。

# OpenImg

## 画像メモリアクセス許可

### 機能

画像メモリをアクセス可能状態（オープン状態）にします。

### コーディング

```
int
OpenImg(
    DEVID          devID ,
    enum WaitMode mode
)
```

### コメント

デバイスID  
ウェイトモード

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**mode**  
ウェイトモード

ウェイトモード	対応定数	内容
WAIT_FOREVER	0	オープン処理実行が終了するまでウェイトする
NOWAIT	1	オープン処理実行終了を待たずに処理終了

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	画像メモリアクセス可能
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	ウェイトモード設定値範囲外
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドは、IP5000コマンドとの互換性のために残されているコマンドです。SVP330画像処理コマンドでは、OpenImg()/CloseImg()コマンドを使用しなくても、ReadImg(),WriteImg(),SetPixelPointer(),ReadPixel(),WritePixel(),ReadPixelContinue(),WritePixelContinue()コマンドが使用できます。

# CloseImg

## 画像メモリアクセス禁止

### 機能

画像メモリをアクセス不可能状態（クローズ状態）にします。

### コーディング

```
int
CloseImg(
    DEVID          devID
)
```

### コメント

デバイスID

### パラメータ

devID  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドは、IP5000コマンドとの互換性のために残されているコマンドです。SVP330画像処理コマンドでは、OpenImg()/CloseImg()コマンドを使用しなくても、ReadImg(),WriteImg(),SetPixelPointer(),ReadPixel(),WritePixel(),ReadPixelContinue(),WritePixelContinue()コマンドが使用できます。

# ReadImg

## 画像メモリ読み出し(矩形領域)

### 機能

パラメータで指定された画面番号の画像メモリを、指定画素数分読み出します。また、指定画素数が画面（ウィンドウ）を超える場合は、超える直前までのデータを読み出します。

なお、読み出しウィンドウはSetWindow()コマンドのSYS\_WINで指定して下さい。

### コーディング

```
long
ReadImg(
    DEVID    devID ,
    int      imgID ,
    char     *imgTbl ,
    long     count
)
```

### コメント

デバイスID  
画面番号  
読み出しデータ格納アドレス  
画素数

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgID

処理画面の画面番号

\*imgTbl

読み出しデータ格納アドレス

count

読み出し画素数

### リターン値

リターン値は long 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	正常終了。読み出しを行ったデータ数。
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 0	画像メモリアクセス禁止状態（画像メモリオープンエラー）
4 9	指定画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

(1) 画像メモリからの読み出しサイズは、ウィンドウが有効のときはウィンドウサイズ、無効のときはビデオフレームサイズに従います。

(2) 画像メモリからの読み出しバイト数は、処理画面の属性により動作が異なります。処理画面がYUVカラー画面の場合は、ウィンドウサイズの2倍になります。詳細は、YUVカラー処理のReadImg( )コマンドを参照して下さい。

# WriteImg

## 画像メモリ書き込み(矩形領域)

### 機能

パラメータで指定された画面番号の画像メモリに、指定画素数分書き込みます。また、指定画素数が画面（ウィンドウ）を超える場合は、超える直前までのデータが書き込まれます。  
なお、書き込みウィンドウはSetWindow() コマンドのSYS\_WINで指定して下さい。

### コーディング

```
long
WriteImg(
    DEVID    devID ,
    int      imgID ,
    char     *imgTbl ,
    long     count
)
```

### コメント

デバイスID  
画面番号  
書き込みデータアドレス  
画素数

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgID**  
処理画面の画面番号

**\*imgTbl**  
書き込みデータアドレス

**count**  
書き込み画素数

### リターン値

リターン値は long 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	正常終了。書き込みを行ったデータ数。
0	異常終了

異常終了時、下記の原因が考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 0	画像メモリアクセス禁止状態（画像メモリオープンエラー）
-	エラーリセットコマンド（ClearIPError）未発行

付録のエラー番号一覧も参照して下さい。

### 詳細情報

(1) 画像メモリへの書き込みサイズは、ウィンドウが有効のときはウィンドウサイズ、無効のときはビデオフレームサイズに従います。

(2) 画像メモリからの書き込みバイト数は、処理画面の属性により動作が異なります。処理画面がYUVカラー画面の場合は、ウィンドウサイズの2倍になります。詳細は、YUVカラー処理のWriteImg( ) コマンドを参照して下さい。

# ReadImgReverse

## 画像メモリ読み出し(ビットマップ矩形領域)

### 機能

パラメータで指定された画面番号の画像メモリを、指定画素数分読み出します。読み出される画像データの原点は左下隅になります。本コマンドはWindowsのDIB形式のビットマップを作成する際に使用します。なお、読み出しウィンドウはSetWindow()コマンドのSYS\_WINで指定して下さい。

### コーディング

```
long
ReadImgReverse(
    DEVID    devID ,
    int      imgID ,
    char     *ImgTbl ,
    long     count
)
```

### コメント

デバイスID  
画面番号  
読み出しデータ格納アドレス  
画素数

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgID**  
処理画面の画面番号

**\*ImgTbl**  
読み出しデータ格納アドレス

**count**  
読み出し画素数

### リターン値

リターン値は long 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。読み出しを行ったデータ数。
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
4 0	画像メモリアクセス禁止状態 (画像メモリオープンエラー)
4 9	指定画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画像メモリからの読み出しサイズは、SYS\_WINの設定に従います。Windowsのビットマップファイルを作成する場合、SYS\_WINの矩形領域から

$biWidth = ex - sx + 1$

$biHeight = ey - sy + 1$

でBITMAPINFOHEADER構造体のbiWidth、biHeightを計算して下さい。countにはSYS\_WINで指定した領域のサイズ

$count = (ex - sx + 1) * (ey - sy + 1)$

を設定しますが、X方向の長さが4の倍数にパディングされるため、WindowsのBITMAPINFOHEADER構造体のbiSizeImageの値以上のデータ格納領域を確保してください。パディングされた領域は0で埋められます。

# WriteImgReverse

## 画像メモリ書き込み(ビットマップ矩形領域)

### 機能

パラメータで指定された画面番号の画像メモリに、指定画素数分書き込みます。書き込まれる画像データの原点は左下隅になります。本コマンドはWindowsのDIB形式のビットマップから画像データを書き込む際に使用します。

なお、読み出しウィンドウはSetWindow()コマンドのSYS\_WINで指定して下さい。

### コーディング

```
long
WriteImgReverse(
    DEVID    devID ,
    int      imgID ,
    char     *imgTbl ,
    long     count
)
```

### コメント

デバイスID  
画面番号  
書き込みデータアドレス  
画素数

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgID

処理画面の画面番号

\*imgTbl

書き込みデータアドレス

count

書き込み画素数

### リターン値

リターン値は long 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。書き込みを行ったデータ数。
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
4 0	画像メモリアクセス禁止状態 (画像メモリオープンエラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画像メモリへの書き込みサイズは、SYS\_WINの設定に従います。Windowsのビットマップファイルからダウンロードを行なう場合、BITMAPINFOHEADER構造体のbiWidth、biHeightと始点座標(sx, sy)から

$ex = sx + biWidth$

$ey = sy + biHeight$

を計算し、SYS\_WINを設定して下さい。count にはSYS\_WINで指定した領域のサイズ

$count = (ex - sx + 1) * (ey - sy + 1)$

を設定しますが、X方向の長さが4の倍数にパディングされるため、WindowsのBITMAPINFOHEADER構造体のbiSizeImageの値以上のデータ格納領域を確保してください。パディングされた領域は0で埋められます。

# SetPixelPointer

## 画像メモリポインタ設定

### 機能

画像メモリアクセスするための、初期設定を行います。  
 なお、ウィンドウサイズはSetWindow()コマンドのSYS\_WINで指定して下さい。

### コーディング

```
int
SetPixelPointer(
    DEVID    devID ,
    int      ImgID
)
```

### コメント

デバイスID  
 画面番号

### パラメータ

devID  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID  
 処理画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
4 0	画像メモリアクセス禁止状態 (画像メモリオープンエラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ReadPixel

## 画像メモリ1画素読み出し

### 機能

パラメータで指定された座標から、1画素読み出します。

### コーディング

```
int
ReadPixel(
    DEVID    devID ,
    int      x ,
    int      y ,
    char     *val
)
```

### コメント

デバイスID  
X座標  
Y座標  
読み出しデータ格納アドレス

### パラメータ

- devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。
- x**  
X座標（画面又はウィンドウ相対値）
- y**  
Y座標（画面又はウィンドウ相対値）
- \*val**  
読み出しデータ格納アドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	X座標設定値範囲外
2 1	Y座標設定値範囲外
4 0	画像メモリアクセス禁止状態（画像メモリオープンエラー）
4 1	使用手続き不正
4 9	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画像メモリ制御コマンドでの座標系はウィンドウ相対であり、ウィンドウの始点が座標（0，0）になります。

設定されているウィンドウ外の座標が指定された場合はエラーとなります。

# WritePixel

## 画像メモリ1画素書き込み

### 機能

パラメータで指定された座標に、1画素書き込みます。

### コーディング

```
int
WritePixel(
    DEVID    devID ,
    int      x ,
    int      y ,
    char     val
)
```

### コメント

デバイスID  
X座標  
Y座標  
書き込みデータ

### パラメータ

- devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。
- x**  
X座標（画面又はウィンドウ相対値）
- y**  
Y座標（画面又はウィンドウ相対値）
- val**  
書き込みデータ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	X座標設定値範囲外
2 1	Y座標設定値範囲外
4 0	画像メモリアクセス禁止状態（画像メモリオープンエラー）
4 1	使用手続き不正
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画像メモリ制御コマンドでの座標系はウィンドウ相対であり、ウィンドウの始点が座標（0，0）になります。

設定されているウィンドウ外の座標が指定された場合はエラーとなります。

# ReadPixelContinue

## 画像メモリ連続読み出し

### 機能

前回読み出しを行った次座標から、1画素読み出します。

### コーディング

```
int
ReadPixelContinue(
    DEVID    devID ,
    char     *val
)
```

### コメント

デバイスID  
読み出しデータ格納アドレス

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*val

読み出しデータ格納アドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
4 0	画像メモリアクセス禁止状態 (画像メモリオープンエラー)
4 1	使用手続き不正
4 4	座標が画面 (ウィンドウ) 外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ReadPixelContinue()は、ReadPixel()を最初に実行し、座標を設定してから実行して下さい。Readpixel()を実行せずに実行すると、読み出し座標が不定のままであるため、処理結果は不定です。

ReadPixelContinue()とWritePixelContinue()を混在して使用しないで下さい。読み出し座標のポインタは、ReadPixelContinue()とWritePixelContinue()で共通なため、WritePixelContinue()を実行すると、座標ポインタがインクリメントされ、次にReadPixelContinueを実行すると、座標ポインタは、WritePixelContinue()でインクリメントされた座標になります。

本コマンドは画面サイズ (ウィンドウサイズ) によって座標の更新を行いますが、座標が画面サイズ (ウィンドウサイズ) を超えた場合はエラーになります。

# WritePixelContinue

## 画像メモリ連続1画素書き込み

### 機能

前回書き込みを行った次座標から、1画素書き込みます。

### コーディング

```
int
WritePixelContinue(
    DEVID    devID ,
    char     val
)
```

### コメント

デバイスID  
書き込みデータ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

val

書き込みデータ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
4 0	画像メモリアクセス禁止状態 (画像メモリオープンエラー)
4 1	使用手続き不正
4 4	座標が画面 (ウィンドウ) 外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

WritePixelContinue()は、WritePixel()を最初に実行し、座標を設定してから実行して下さい。WritePixel()を実行せずに実行すると、書き込み座標が不定のままであるため、処理結果は不定です。

WritePixelContinue()とReadPixelContinue()を混在して使用しないで下さい。書き込み座標のポインタ

は、WritePixelContinue()とReadPixelContinue()で共通なため、ReadPixelContinue()を実行すると、座標ポインタがインクリメントされ、次にWritePixelContinue()を実行すると、座標ポインタは、ReadPixelContinue()でインクリメントされた座標になります。

本コマンドは画面サイズ (ウィンドウサイズ) によって座標の更新を行いますが、座標が画面サイズ (ウィンドウサイズ) を超えた場合はエラーになります。

# ReadImgLine

## 画像メモリ1ライン読み出し

### 機能

パラメータで指定された画面番号の画像メモリから、(sx,sy)を開始座標、(ex,ey)を終了座標とする直線上のデータをデータテーブルに読み出します。

### コーディング

```
int
ReadImgLine(
    DEVID      devID ,
    int        imgID ,
    LINETBL    *LineTbl ,
    int        sx ,
    int        sy ,
    int        ex ,
    int        ey
)
```

### コメント

デバイスID  
画面番号  
読み出しデータテーブル  
読み出し開始点X座標  
読み出し開始点Y座標  
読み出し終了点X座標  
読み出し終了点Y座標

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgID**  
処理画面の画面番号

**\*LineTbl**  
読み出しデータテーブル

**sx**  
読み出し開始点X座標

**sy**  
読み出し開始点Y座標

**ex**  
読み出し終了点X座標

**ey**  
読み出し終了点Y座標

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

LINETBLのフォーマットおよび内容を下記に示します。  
vpxsys.hで宣言しています。

```
typedef struct {  
    short    lower;  
    short    num;  
    short    data[512];  
} LINETBL;
```

LINETBL	説明
lower	読み出し開始点座標
num	読み出し有効データ数
data[512]	読み出しデータ

# WriteImgLine

## 画像メモリ1ライン書き込み

### 機能

パラメータで指定された画面番号の画像メモリに、データテーブルに設定されたデータを、(sx,sy)を開始座標、(ex,ey)を終了座標とする直線上に書き込みます。

### コーディング

```
int
WriteImgLine(
    DEVID      devID ,
    int        imgID ,
    LINETBL    *LineTbl ,
    int        sx ,
    int        sy ,
    int        ex ,
    int        ey
)
```

### コメント

デバイスID  
画面番号  
書き込みデータテーブル  
書き込み開始点X座標  
書き込み開始点Y座標  
書き込み終了点X座標  
書き込み終了点Y座標

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgID**  
処理画面の画面番号

**\*LineTbl**  
書き込みデータテーブル

**sx**  
書き込み開始点X座標

**sy**  
書き込み開始点Y座標

**ex**  
書き込み終了点X座標

**ey**  
書き込み終了点Y座標

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

LINETBLのフォーマット及び内容は、ReadImgLine()の詳細情報の欄を参照して下さい。

# GetImVirtualAddress

## 画像メモリのアドレス取得

### 機能

指定画面の画像メモリの先頭アドレスを取得します。画像メモリはUMA方式でオンボードCPUからダイレクトに画像メモリをアクセスできます。ただし、アドレス計算やキャッシュパージの操作 (RefreshImg()) はすべてユーザアプリケーション側で行う必要があります。  
PC側ではこのアドレスを取得してもアクセスできないので注意して下さい。

### コーディング

```
unsigned long
GetImVirtualAddress(
    DEVID      devID ,
    int        ImgID ;
    int        mode
)
```

### コメント

デバイスID  
画面番号  
モード

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
画面番号

**mode**  
現在未使用。「0」を設定して下さい。

### リターン値

リターン値は unsigned long 型で返されます。詳細を下表に示します。

リターン値	内容
1	画像メモリアクセス可能
0xFFFFFFFF	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	ウェイトモード設定値範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

(X,Y)座標値に対するアドレス(addr)計算は、以下のようになります。

$$\text{addr} = \text{GetImVirtualAddress}(\text{devID}, \text{ImgID}, 0) + \text{WIDTH} * Y + X ;$$

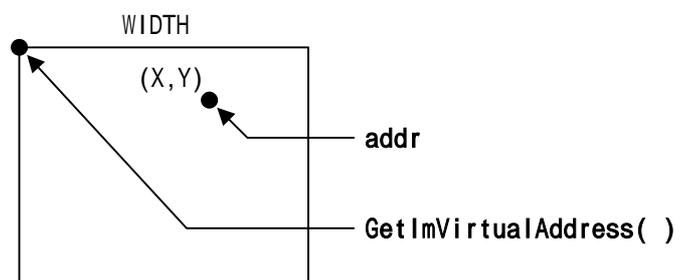
WIDTH : 画面(ImgID)のX方向のサイズ

画像メモリへのアクセス終了時のキャッシュパージが必要な場合は

RefreshImg( )

コマンドを使用して下さい。

なお、画面領域を超えてアクセスするとシステムダウンになりますので十分注意が必要です。



## PutLine

## 線・矩形描画

## 機能

colで指定されたデータを、ImgIDで指定された画面番号の画像メモリの(sx,sy)を開始点座標、(ex,ey)を終了座標とする直線上に書き込みます。

図形描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、Displmg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg()コマンドは必要ありません。

## コーディング

```
int
PutLine(
    DEVID          devID ,
    int            ImgID ,
    enum IPDrawLineMode mode ,
    int            sx ,
    int            sy ,
    int            ex ,
    int            ey ,
    int            col
)
```

## コメント

デバイスID  
画面番号  
モード  
書き込み開始点X座標  
書き込み開始点Y座標  
書き込み終了点X座標  
書き込み終了点Y座標  
書き込みデータ

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID

描画面の画面番号

mode

描画モード

描画モード	対応数	説明
DRAW_LINE	0	線描画 
DRAW_RECT	1	矩形描画 
DRAW_FILL	2	フィル 
DRAW_CROSS_RECT	3	中心線付矩形 
DRAW_LEFT_TOP_RECT	4	左上ポイント付矩形 
DRAW_RIGHT_BOTTOM_RECT	5	右下ポイント付矩形 

sx

書き込み開始点X座標

sy

書き込み開始点Y座標

ex

書き込み終了点X座標

ey

書き込み終了点Y座標

col

書き込みデータ(0~255)

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

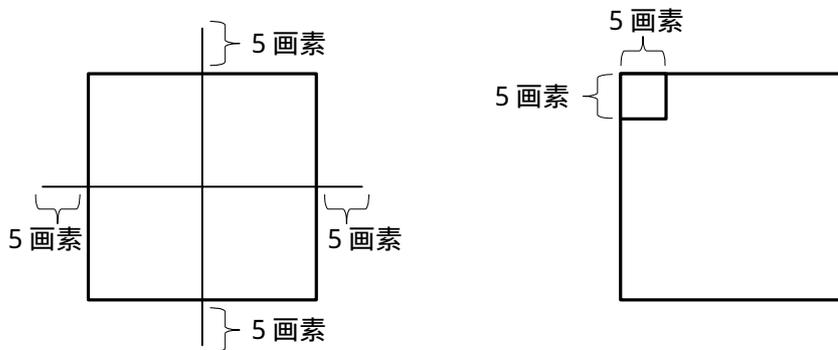
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

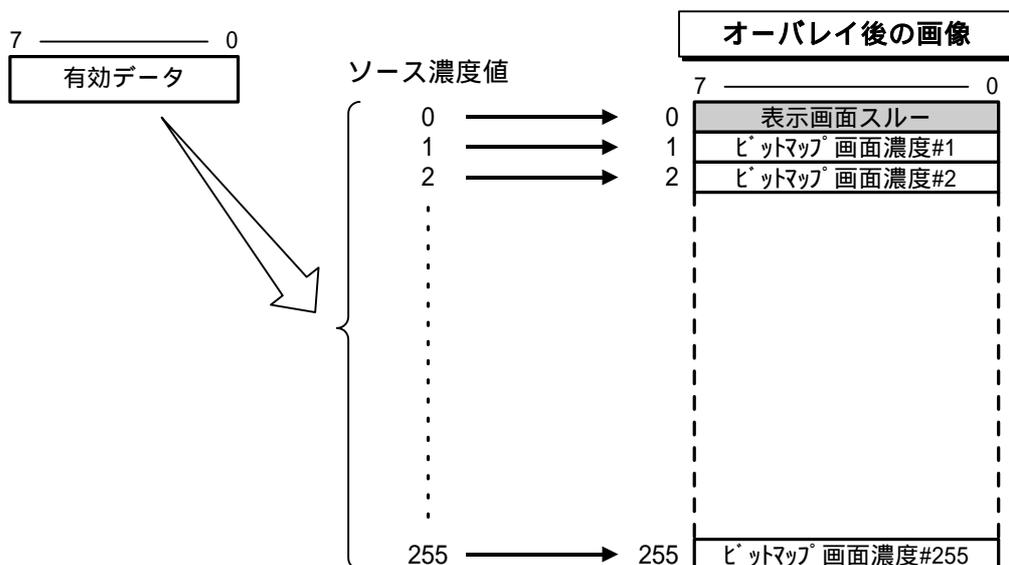
## 詳細情報

十字およびポイントのサイズは図のようになっています。



ビットマップ画面での表示について

ビットマップ画面 (オーバーレイ画面) のオーバーレイ表示はビットマップ画面のデータを変換し、表示します。ビットマップ画面データが「0」の部分は表示画面、「1 ~ 255」の部分はビットマップ画面の映像を表示します。



# PutCross

## 十字カーソル描画

### 機能

十字カーソルを描画します。

図形描画コマンドの描画は、ローカルCPU (SH) で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ (IM) には書き込まれていない部分が出てきます。RefreshImg( ) コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg( ) やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg( ) コマンドは必要ありません。

### コーディング

```
int
PutCross(
    DEVID      devID ,
    int        ImgID ,
    int        mode ,
    int        xp ,
    int        yp ,
    int        col
)
```

### コメント

デバイスID  
画面番号  
モード  
中心X座標  
中心Y座標  
書き込みデータ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID

描画面の画面番号

mode

描画モード

xp

中心X座標

yp

中心Y座標

col

書き込みデータ

ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

描画モード	説明
0	十字 
1	x印 
2	点 
3	点十字 
4	点x印 
5	点印 

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

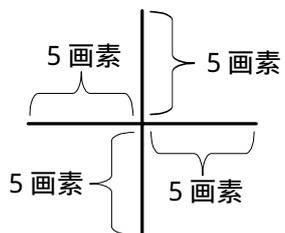
エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

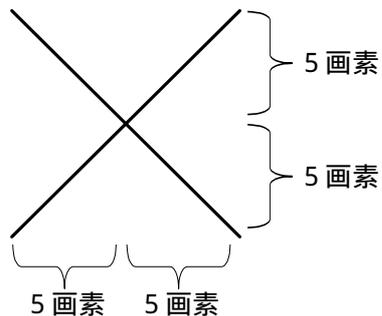
## 詳細情報

十字および×印のサイズは図のようになっています。

十字(mode : 0)



×印(mode : 1)



点(mode : 2)



点 + 印(mode : 3)



点 × 印(mode : 4)



点 印(mode : 5)



## PutLineWindow

## ラインウィンドウ描画

### 機能

指定されたラインウィンドウの外形を線描画します。

図形描画コマンドの描画は、ローカルCPU (SH) で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ (IM) には書き込まれていない部分が出てきます。RefreshImg( ) コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg( ) やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg( ) コマンドは必要ありません。

### コーディング

```
int
PutLineWindow(
    DEVID      devID ,
    int        ImgID ,
    LINEWINDOW *LineWindow ,
    int        col
)
```

### コメント

デバイスID  
画面番号  
ラインウィンドウ  
書き込みデータ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画画面の画面番号

**\*LineWindow**  
描画を行うラインウィンドウ

**col**  
書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

LINETBL、LINEWINDOWのフォーマットおよび内容を下記に示します。  
vpxsys.hで宣言しています。

```
typedef struct {  
    short      sx;  
    short      sy;  
    short      ex;  
    short      ey;  
    short      leng;  
    short      opt;  
} LINEWINDOW;
```

LINEWINDOW	説明
sx	開始点座標 X
sy	開始点座標 Y
ex	終了点座標 X
ey	終了点座標 Y
leng	投影幅
opt	オプション (未使用)

# PutPolygon

## 多角形描画

### 機能

colで指定されたデータで、ImgIDで指定された画面番号の画像メモリの(xp,yp)を頂点とする多角形を描画モードに従い描画します。

図形描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutPolygon(
    DEVID          devid ,
    int            ImgID ,
    enum IPDrawLineMode mode ,
    WPOINT        *point ,
    int            num ,
    int            col
)
```

### コメント

デバイスID  
画面番号  
モード  
頂点X, Y座標列  
頂点数  
書き込みデータ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画面の画面番号

**mode**  
描画モード

描画モード	対応数	説明
DRAW_LINE	0	線描画
DRAW_FILL	2	フィル

**\*point**  
頂点X、Y座標列

**num**  
頂点数

**col**  
書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外(不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

WPOINTのフォーマットを下記に示します。  
vpxsys.hで宣言しています。

```
typedef struct {  
    short      x;  
    short      y;  
} WPOINT;
```

# PutCircle

## 円描画

### 機能

colで指定されたデータで、ImgIDで指定された画面番号の画像メモリの(xp,yp)を中心とする半径rの円形を描画し、中を塗りつぶします。

図形描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutCircle(
    DEVID
    int
    enum IPDrawLineMode
    int
    int
    int
    int
    int
)
```

```
devID ,
ImgID ,
mode ,
xp ,
yp ,
r ,
col
```

```
デバイスID
画面番号
描画モード
中心X座標
中心Y座標
半径
書き込みデータ
```

### コメント

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID

描画面の画面番号

mode

描画モード

xp

中心X座標

yp

中心Y座標

r

半径

col

書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外(不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# PutArc

## 円弧描画

### 機能

colで指定されたデータを、ImgIDで指定された画面番号の画像メモリの(xp,yp)を中心、(sx,sy)を始点とする角angleの円弧に書き込みます。

図形描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutArc(
    DEVID          devid ,
    int            ImgID ,
    enum IPDrawLineMode mode ,
    int            xp ,
    int            yp ,
    int            sx ,
    int            sy ,
    float          angle ,
    int            col
)
```

### コメント

デバイスID  
画面番号  
描画モード  
中心X座標  
中心Y座標  
始点X座標  
始点Y座標  
円弧の角度  
書き込みデータ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画面の画面番号

**mode**  
描画モード

描画モード	対応数	説明
DRAW_LINE	0	線描画
DRAW_FILL	2	フィル

**xp**  
中心X座標

**yp**  
中心Y座標

**sx**  
始点X座標

**sy**  
始点Y座標

**angle**  
円弧の角度 (degree)

**col**  
書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

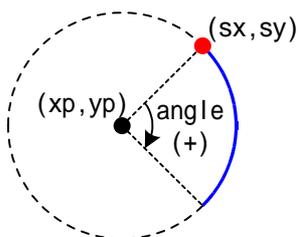
エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

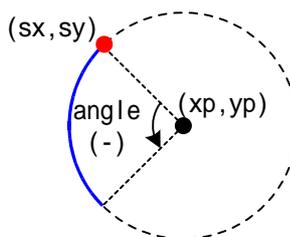
**詳細情報**

円弧の中心と始点、描画角度の関係を示します。

angle  $\geq$  0  
(時計回り描画)



angle < 0  
(反時計回り描画)



なお、描画モードが DRAW\_FILL のとき angle が 180° を超えると正しく描画できません。

# PutRectangle

## 矩形描画

### 機能

colで指定されたデータを、ImgIDで指定された画面番号の画像メモリの(xp,yp)を中心座標、横幅(lengh)、縦幅(lengv)、角の円弧の半径(round)、傾き(angle)とする四角形に書き込みます。

図形描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutRectangle(
    DEVID          devid ,      デバイスID
    int            ImgID ,      画面番号
    enum IPDrawLineMode mode ,  描画モード
    int            xp ,         中心X座標
    int            yp ,         中心Y座標
    int            lengh ,      横幅
    int            lengv ,      縦幅
    int            round ,      角の円弧の半径
    float          angle ,      傾き
    int            col          書き込みデータ
)
```

### コメント

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画面の画面番号

**mode**  
描画モード

描画モード	対応数	説明
DRAW_LINE	0	線描画
DRAW_FILL	2	フィル

**xp**  
中心X座標

**yp**  
中心Y座標

**lengh**  
横幅

**lengv**  
縦幅

**round**  
角の円弧の半径

**angle**  
傾き (degree)

**col**  
書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

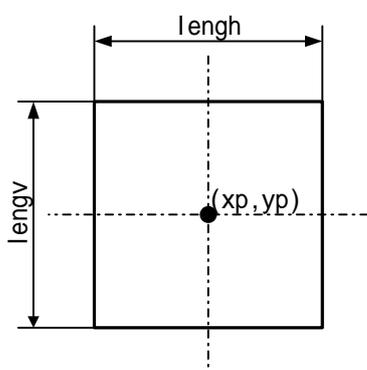
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (Clear IPError) 未発行

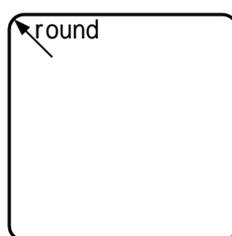
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

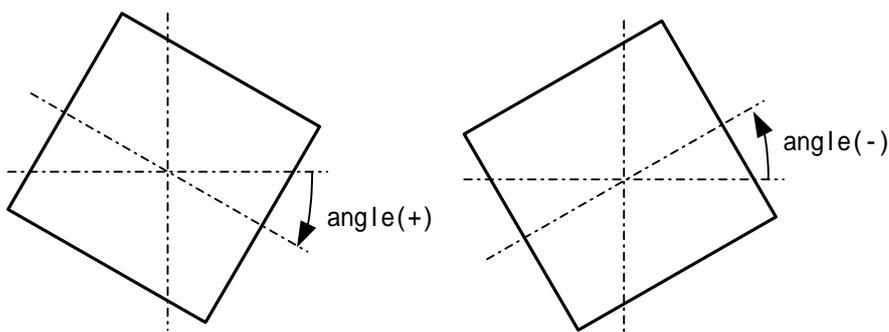
四角形と各パラメータの関係を示します。



<コーナー部>



<傾き>



# PutTriangle

## 三角形描画

### 機能

colで指定されたデータを、ImgIDで指定された画面番号の画像メモリの(xp,yp)を中心座標、横幅(lengh)、縦幅(lengv)、角の円弧の半径(round)、傾き(angle)とする三角形に書き込みます。

図形描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、Displng()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutTriangle(
    DEVID          devid ,      デバイスID
    int            ImgID ,      画面番号
    enum IPDrawLineMode mode ,  描画モード
    int            xp ,         中心X座標
    int            yp ,         中心Y座標
    int            lengh ,      横幅
    int            lengv ,      縦幅
    int            round ,      角の円弧の半径
    float          angle ,      傾き
    int            col          書き込みデータ
)
```

### コメント

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgID

描画面の画面番号

#### mode

描画モード

描画モード	対応数	説明
DRAW_LINE	0	線描画
DRAW_FILL	2	フィル

#### xp

中心X座標

#### yp

中心Y座標

#### lengh

横幅

#### lengv

縦幅

#### round

角の円弧の半径

#### angle

傾き (degree)

#### col

書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

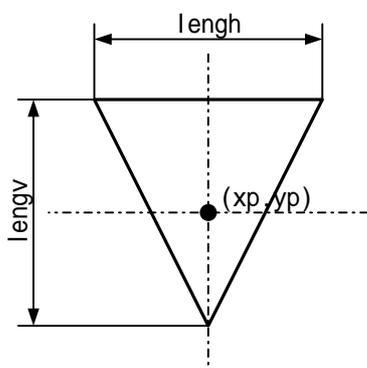
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

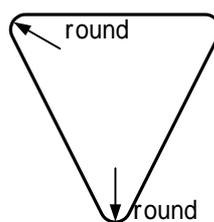
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

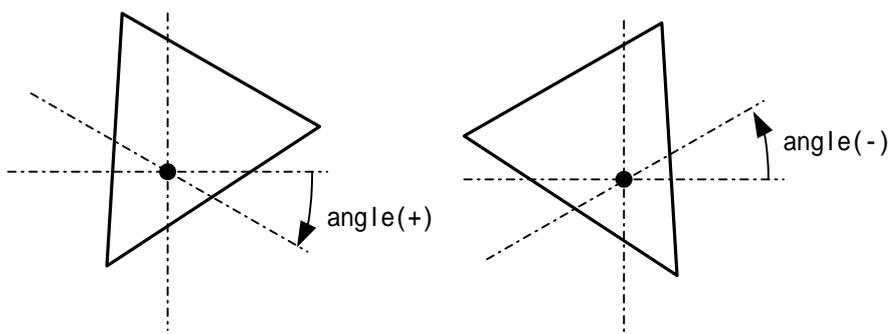
三角形と各パラメータの関係を示します。



<コーナー部>



<傾き>



# PutDiamond

## 菱形描画

### 機能

colで指定されたデータを、ImgIDで指定された画面番号の画像メモリの(xp,yp)を中心座標、横幅(lengh)、縦幅(lengv)、角の円弧の半径(round)、傾き(angle)とするひし形に書き込みます。

図形描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutDiamond(
    DEVID          devID ,
    int            ImgID ,
    enum IPDrawLineMode mode ,
    int            xp ,
    int            yp ,
    int            lengh ,
    int            lengv ,
    int            round ,
    float          angle ,
    int            col
)
```

### コメント

デバイスID  
画面番号  
描画モード  
中心X座標  
中心Y座標  
横幅  
縦幅  
角の円弧の半径  
傾き  
書き込みデータ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画面の画面番号

**mode**  
描画モード

描画モード	対応数	説明
DRAW_LINE	0	線描画
DRAW_FILL	2	フィル

**xp**  
中心X座標

**yp**  
中心Y座標

**lengh**  
横幅

**lengv**  
縦幅

**round**  
角の円弧の半径

**angle**  
傾き (degree)

**col**  
書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

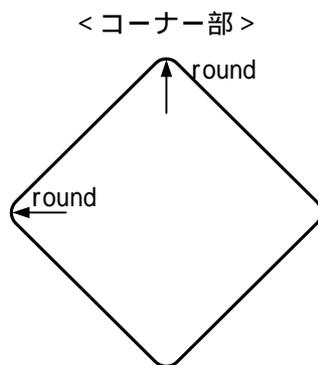
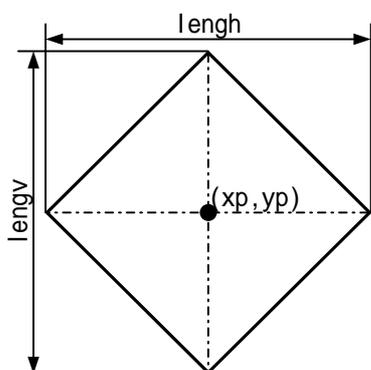
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

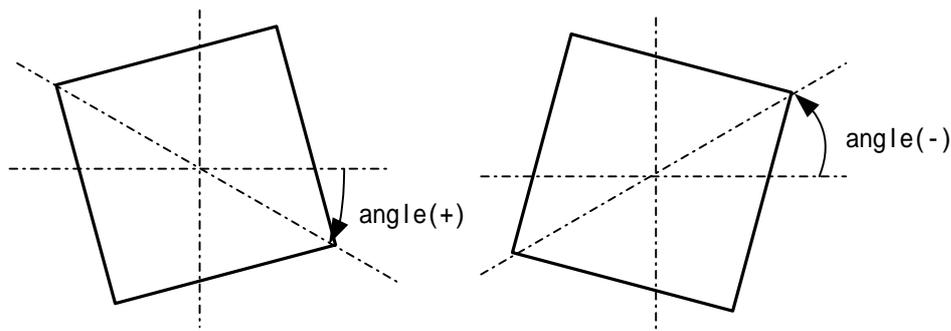
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ひし形と各パラメータの関係を示します。



<傾き>



# PutCrossRect

## 十字形描画

### 機能

colで指定されたデータを、ImgIDで指定された画面番号の画像メモリの(xp,yp)を中心座標、横幅(lengh)、縦幅(lengv)、腕幅(width)、角の円弧の半径(round)、傾き(angle)とする十字形に書き込みます。

### コーディング

```
int
PutCrossRect(
    DEVID          devID ,
    int            ImgID ,
    enum IPDrawLineMode mode ,
    int           xp ,
    int           yp ,
    int           lengh ,
    int           lengv ,
    int           width ,
    int           round ,
    int           angle ,
    float         col
)

```

### コメント

デバイスID  
画面番号  
描画モード  
中心X座標  
中心Y座標  
横幅  
縦幅  
腕幅  
角の円弧の半径  
傾き  
書き込みデータ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画面の画面番号

**mode**  
描画モード

描画モード	対応数	説明
DRAW_LINE	0	線描画
DRAW_FILL	2	フィル
DRAW_LINE_R	6	内側ラウンド付線描画
DRAW_FILL_R	7	内側ラウンド付フィル

**xp**  
中心X座標

**yp**  
中心Y座標

**lengh**  
横幅

**lengv**  
縦幅

**width**  
腕幅

**round**  
角の円弧の半径

**angle**  
傾き (degree)

**col**  
書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

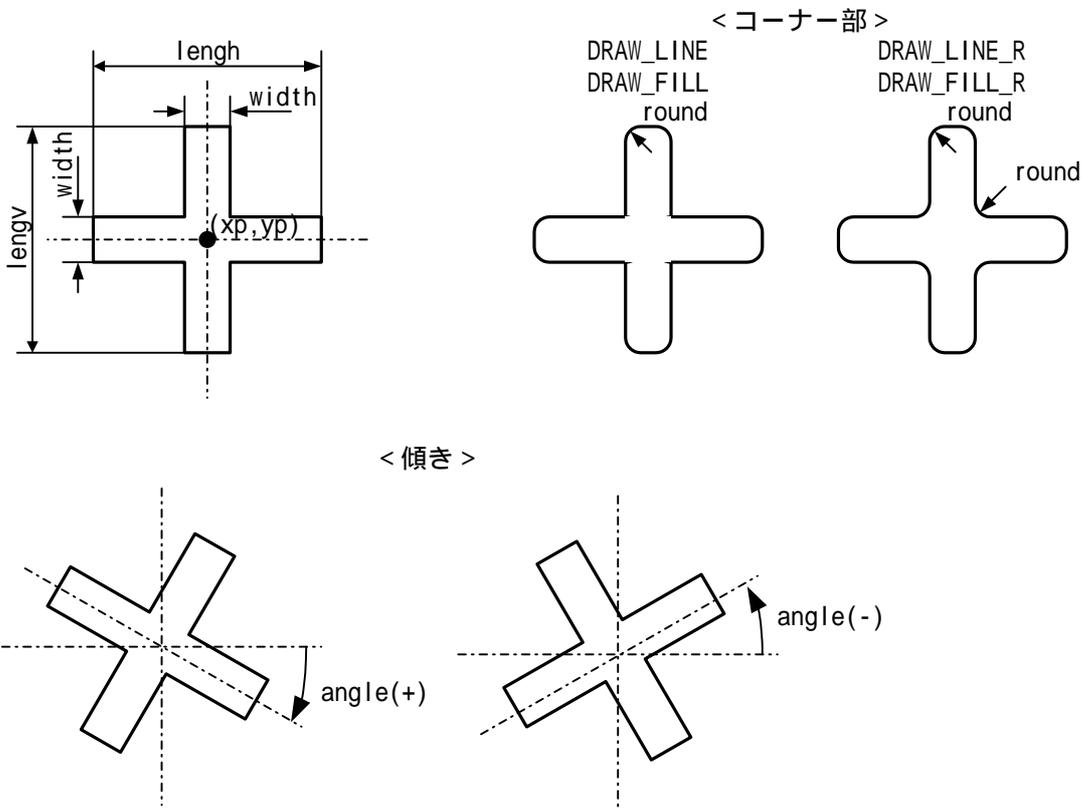
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPErr) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

十字形と各パラメータの関係を示します。



図形描画コマンドの描画は、ローカルCPU (SH) で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ (IM) には書き込まれていない部分が出てきます。RefreshImg( ) コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、Displmg( ) やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg( ) コマンドは必要ありません。

# PutTwinRectangle

## 2連矩形描画

### 機能

colで指定されたデータを、ImgIDで指定された画面番号の画像メモリの(xp,yp)を中心座標、横幅(lengh)、縦幅(lengv)、腕幅(width)、角の円弧の半径(round)、傾き(angle)とする2四角形に書き込みます。

### コーディング

```
int
PutTwinRectangle(
    DEVID          ,      devID ,
    int            ,      ImgID ,
    enum IPDrawLineMode mode ,
    int           ,      xp ,
    int           ,      yp ,
    int           ,      lengh ,
    int           ,      lengv ,
    int           ,      width ,
    int           ,      round ,
    float         ,      angle ,
    int           ,      col
)
```

### コメント

デバイスID  
画面番号  
描画モード  
中心X座標  
中心Y座標  
横幅  
縦幅  
2四角形の重なり幅  
角の円弧の半径  
傾き  
書き込みデータ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画面の画面番号

**mode**  
描画モード

描画モード	対応数	説明
DRAW_LINE	0	線描画
DRAW_FILL	2	フィル
DRAW_LINE_R	6	内側ラウンド付線描画
DRAW_FILL_R	7	内側ラウンド付フィル

**xp**  
中心X座標

**yp**  
中心Y座標

**lengh**  
横幅

**lengv**  
縦幅

**width**  
腕幅

**round**  
角の円弧の半径

**angle**  
傾き (degree)

**col**  
書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

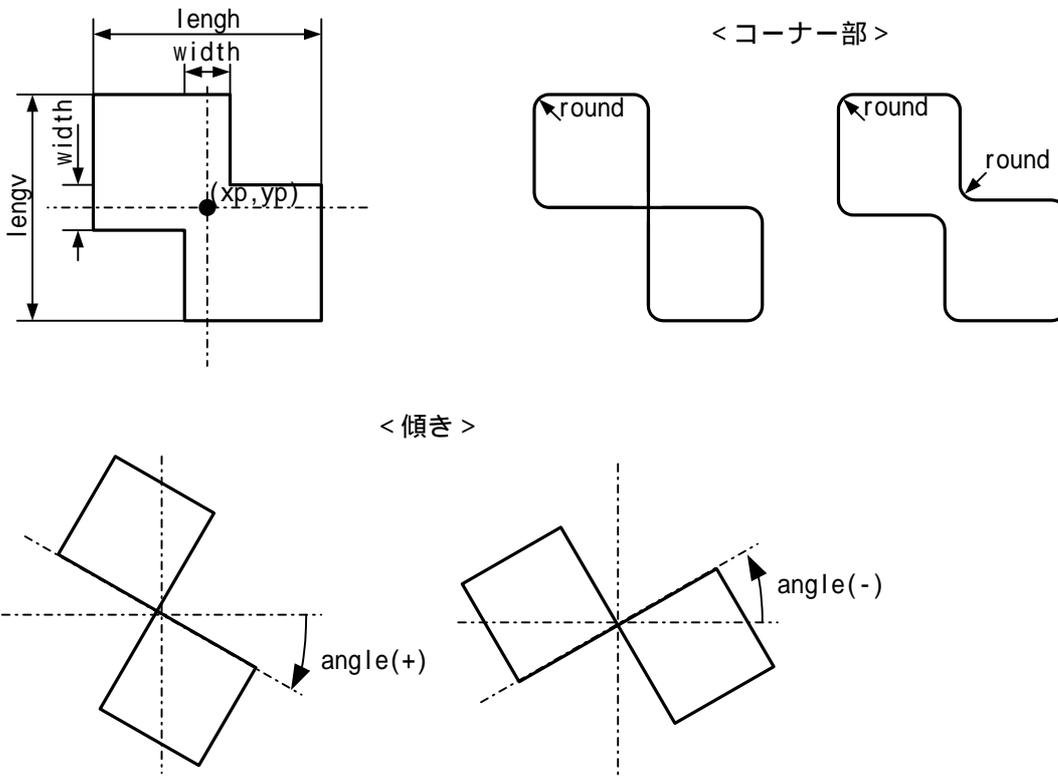
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPErr) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

2 四角形と各パラメータの関係を示します。



図形描画コマンドの描画は、ローカルCPU (SH) で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ (IM) には書き込まれていない部分が出てきます。RefreshImg( ) コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、Displmg( ) やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg( ) コマンドは必要ありません。

# PutArcExt

## 円弧描画(拡張)

### 機能

colで指定されたデータを、ImgIDで指定された画面番号の画像メモリの(xp,yp)を中心、(sx,sy)を始点とする角angleの円弧に書き込みます。なお、パラメータがフロート型である以外はPutArc()と同じです。

図形描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutArcExt(
    DEVID          devID ,      デバイスID
    int            ImgID ,      画面番号
    enum IPDrawLineMode mode ,  描画モード
    float         xp ,         中心X座標
    float         yp ,         中心Y座標
    float         sx ,         始点X座標
    float         sy ,         始点Y座標
    float         angle ,      円弧の角度
    int           col          書き込みデータ
)
```

### コメント

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画面の画面番号

**mode**  
描画モード

描画モード	対応数	説明
DRAW_LINE	0	線描画
DRAW_FILL	2	フィル

**xp**  
中心X座標

**yp**  
中心Y座標

**sx**  
始点X座標

**sy**  
始点Y座標

**angle**  
円弧の角度 (degree)

**col**  
書き込みデータ。ビットマップ画面への書き込みデータについては、PutLine()コマンドの詳細情報を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

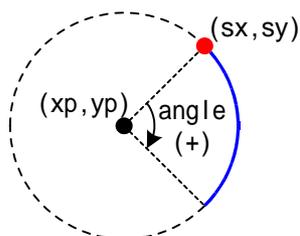
エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

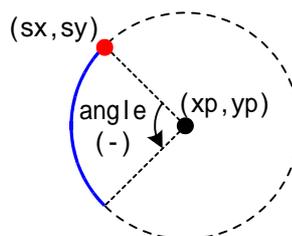
**詳細情報**

円弧の中心と始点、描画角度の関係を示します。

angle  $\geq$  0  
(時計回り描画)



angle < 0  
(反時計回り描画)



なお、描画モードが DRAW\_FILL のとき angle が 180° を超えると正しく描画できません。

# RefreshImg

## 画像メモリのリフレッシュ

### 機能

図形描画コマンド、文字描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。そのため、描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出す処理が必要になります。本コマンドは、リフレッシュ対象画面の最新画像データがSHのキャッシュ上に存在する場合、最新画像データを画像メモリへ書き出します。

### コーディング

```
int
RefreshImg(
    DEVID          ,   devid ,
    int            ,   ImgID  ,
)                   デバイスID
                   画面番号
```

### コメント

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

以下に、コーディング例を示します。

```
/* 図形描画 */
PutLine(devID,ImgID,DRAW_LINE,50,60,100,60,255);
PutLine(devID,ImgID,DRAW_LINE,100,60,100,150,255);
PutLine(devID,ImgID,DRAW_LINE,50,150,100,150,255);
PutLine(devID,ImgID,DRAW_LINE,50,150,50,60,255);

/* 画像メモリのリフレッシュ */
RefreshImg(devID,ImgID);
```

# PutEllipse

## 楕円描画処理

### 機能

楕円描画処理を行います。

### コーディング

```

int
PutEllipse(
    DEVID          devID ,
    int            ImgID ,
    enum IPDrawLineMode mode ,
    float          cx ,
    float          cy ,
    float          width ,
    float          height ,
    float          angle ,
    int            col
)

```

### コメント

デバイスID  
 画面番号  
 描画モード  
 楕円の中心X座標  
 楕円の中心Y座標  
 楕円を内接する長方形の幅(5~)  
 楕円を内接する長方形の高さ(5~)  
 回転角度(°)  
 書き込みデータ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
画面番号。

**mode**  
描画モード。

描画モード	対応定数	
DRAW_RECT	1	線描画
DRAW_FILL	2	フィル

**cx**  
楕円の中心X座標。

**cy**  
楕円の中心Y座標。

**width**  
楕円を内接する長方形の幅(5~)。

**height**  
楕円を内接する長方形の高さ(5~)。

**angle**  
回転角度(°)。

**col**  
書き込みデータ。

**リターン値**

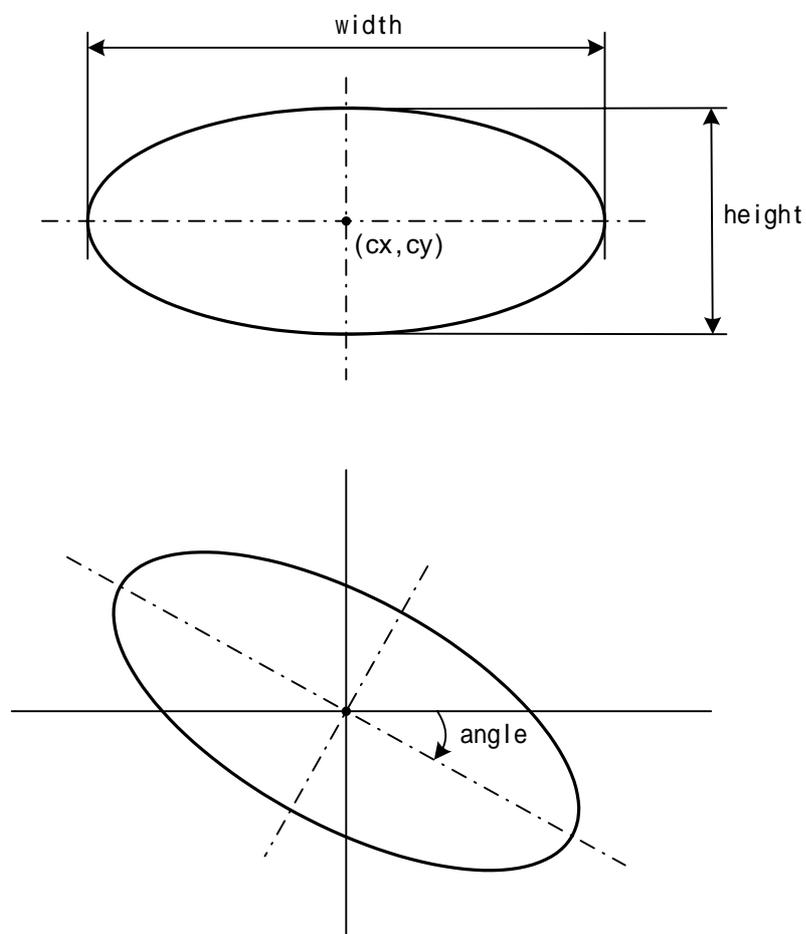
リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10	画面空き領域、または範囲外 (不当画面番号エラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# PutAnkString

## 全角文字列の描画(英数文字)

### 機能

指定位置に全角文字列を描画します。

文字描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われませんので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutAnkString(
    DEVID    devID ,
    int lmgID ,
    char     *string ,
    int mode ,
    int sx ,
    int sy ,
    int col
)
```

### コメント

デバイスID  
画面番号  
文字列テーブルへのポインタ  
モード  
開始X座標  
開始Y座標  
色

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

lmgID

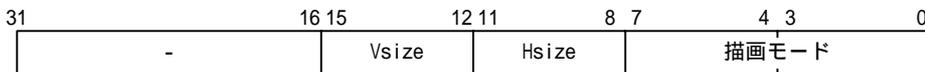
描画面の画面番号

\*string

文字列テーブルへのポインタ

mode

文字描画モード



Hsize、VsizeはそれぞれX方向、Y方向の文字倍率です。

Hsize,Vsize	対応数	内容
STR_SIZE_1	0	X(Y)方向を倍率1倍で描画
STR_SIZE_2	1	X(Y)方向を倍率2倍で描画
STR_SIZE_3	2	X(Y)方向を倍率3倍で描画
STR_SIZE_4	3	X(Y)方向を倍率4倍で描画

文字描画モード	対応数	内容
BACKCOL_NOWRITE	0	文字部分のみcolF色で描画
BACKCOL_NOWRITE_REVERSE	1	文字背景のみcolF色で描画
BACKCOL_WRITE	2	文字部分をcolF色、背景をcolBで描画
BACKCOL_WRITE_REVERSE	3	文字部分をcolB色、背景をcolFで描画

sx

開始X座標

sy

開始Y座標

col

書き込みデータ(詳細説明を参照)

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

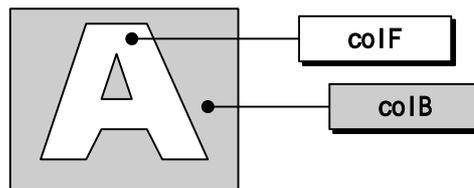
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPErr) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

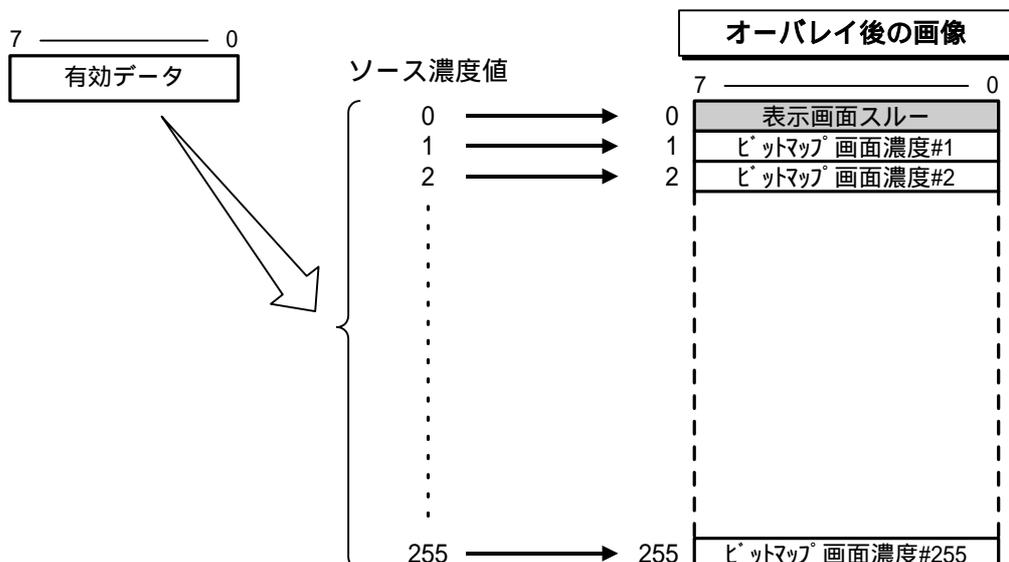
## 詳細情報

書き込みデータ col のビット構成



ビットマップ画面での表示について

ビットマップ画面 (オーバーレイ画面) のオーバーレイ表示はビットマップ画面のデータを変換し、表示します。ビットマップ画面データが「0」の部分は表示画面、「1 ~ 255」の部分はビットマップ画面の映像を表示します。



# PutHalfString

## 半角文字列の描画(英数文字)

### 機能

指定位置に半角文字列を描画します。

文字描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われませんので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutHalfString(
    DEVID    devID ,
    int lmgID ,
    char     *string ,
    int mode ,
    int sx ,
    int sy ,
    int col
)
```

### コメント

デバイスID  
画面番号  
文字列テーブルへのポインタ  
モード  
開始X座標  
開始Y座標  
色

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

lmgID

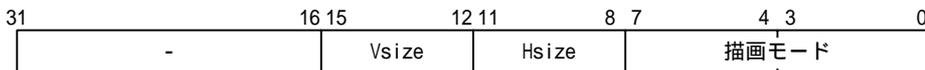
描画面の画面番号。

\*string

文字列テーブルへのポインタ

mode

文字描画モード



Hsize、VsizeはそれぞれX方向、Y方向の文字倍率です。

Hsize,Vsize	対応数	内容
STR_SIZE_1	0	X(Y)方向を倍率1倍で描画
STR_SIZE_2	1	X(Y)方向を倍率2倍で描画
STR_SIZE_3	2	X(Y)方向を倍率3倍で描画
STR_SIZE_4	3	X(Y)方向を倍率4倍で描画

文字描画モード	対応数	内容
BACKCOL_NOWRITE	0	文字部分のみcolF色で描画
BACKCOL_NOWRITE_REVERSE	1	文字背景のみcolF色で描画
BACKCOL_WRITE	2	文字部分をcolF色、背景をcolBで描画
BACKCOL_WRITE_REVERSE	3	文字部分をcolB色、背景をcolFで描画

sx

開始X座標

sy

開始Y座標

col

書き込みデータ。PutAnkString()コマンドの詳細説明を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPErrror) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## PutAnkChar

## 全角文字の描画(英数文字)

## 機能

指定位置に全角文字を描画します。

文字描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg( )コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg( )やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われませんので、RefreshImg( )コマンドは必要ありません。

## コーディング

```
int
PutAnkChar(
    DEVID    devID ,
    int lmgID ,
    char     code ,
    int mode ,
    int sx ,
    int sy ,
    int lx ,
    int ly ,
    int col
)
```

## コメント

デバイスID  
画面番号  
文字コード  
モード  
開始X座標  
開始Y座標  
フォントのX方向画素数  
フォントのY方向画素数  
色

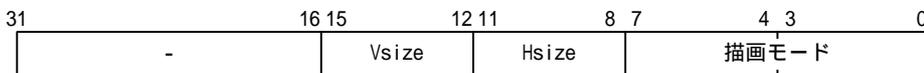
## パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**lmgID**  
描画面の画面番号

**code**  
文字コード

**mode**  
文字描画モード



Hsize、VsizeはそれぞれX方向、Y方向の文字倍率です。

Hsize,Vsize	対応数	内容
STR_SIZE_1	0	X(Y)方向を倍率1倍で描画
STR_SIZE_2	1	X(Y)方向を倍率2倍で描画
STR_SIZE_3	2	X(Y)方向を倍率3倍で描画
STR_SIZE_4	3	X(Y)方向を倍率4倍で描画

文字描画モード	対応数	内容
BACKCOL_NOWRITE	0	文字部分のみcolF色で描画
BACKCOL_NOWRITE_REVERSE	1	文字背景のみcolF色で描画
BACKCOL_WRITE	2	文字部分をcolF色、背景をcolBで描画
BACKCOL_WRITE_REVERSE	3	文字部分をcolB色、背景をcolFで描画

**sx**  
開始X座標

**sy**  
開始Y座標

**lx**  
フォントのX方向画素数

**ly**  
フォントのY方向画素数

**col**  
書き込みデータ。PutAnkString()コマンドの詳細説明を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

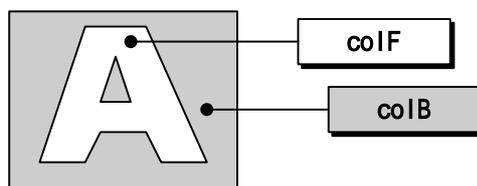
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

書き込みデータ col のビット構成



## PutHalfChar

## 半角文字の描画(英数文字)

## 機能

指定位置に半角文字を描画します。

文字描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg( )コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg( )やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われませんので、RefreshImg( )コマンドは必要ありません。

## コーディング

```
int
PutHalfChar(
    DEVID    devID ,
    int lmgID ,
    char     code ,
    int mode ,
    int sx ,
    int sy ,
    int lx ,
    int ly ,
    int col
)
```

## コメント

デバイスID  
画面番号  
文字コード  
モード  
開始X座標  
開始Y座標  
フォントのX方向画素数  
フォントのY方向画素数  
色

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

lmgID

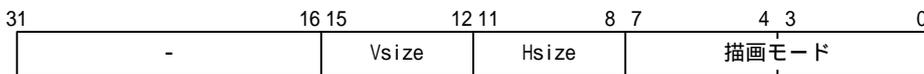
描画面の画面番号

code

文字コード

mode

文字描画モード



Hsize、VsizeはそれぞれX方向、Y方向の文字倍率です。

Hsize,Vsize	対応数	内容
STR_SIZE_1	0	X(Y)方向を倍率1倍で描画
STR_SIZE_2	1	X(Y)方向を倍率2倍で描画
STR_SIZE_3	2	X(Y)方向を倍率3倍で描画
STR_SIZE_4	3	X(Y)方向を倍率4倍で描画

文字描画モード	対応数	内容
BACKCOL_NOWRITE	0	文字部分のみcolF色で描画
BACKCOL_NOWRITE_REVERSE	1	文字背景のみcolF色で描画
BACKCOL_WRITE	2	文字部分をcolF色、背景をcolBで描画
BACKCOL_WRITE_REVERSE	3	文字部分をcolB色、背景をcolFで描画

sx

開始X座標

sy

開始Y座標

lx

フォントのX方向画素数

ly

フォントのY方向画素数

col

書き込みデータ。PutAnkString()コマンドの詳細説明を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

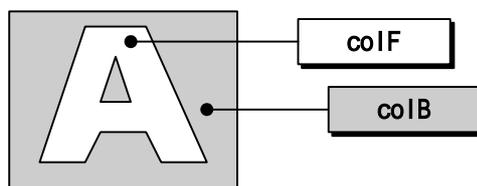
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

書き込みデータ col のビット構成



# PutString

## カナ漢字文字列の描画(シフトJIS)

### 機能

指定位置にカナ漢字文字列を描画します。カナ漢字文字列はシフトJISに対応しています。

文字描画コマンドの描画は、ローカルCPU(SH)で実行します。コマンド実行中はSHのキャッシュに一時的に蓄えられるため、実際の画像メモリ(IM)には書き込まれていない部分が出てきます。RefreshImg()コマンドにより描画が全て終了した時点で、そのキャッシュメモリから画像メモリに強制的にデータを書き出して下さい。なお、図形描画コマンドの後に、DispImg()やその他の画像処理コマンドを実行すると、その中でキャッシュの書き出しが行われますので、RefreshImg()コマンドは必要ありません。

### コーディング

```
int
PutString(
    DEVID    devID ,
    int lmgID ,
    char    *string ,
    int mode ,
    int sx ,
    int sy ,
    int col
)
```

### コメント

デバイスID  
画面番号  
文字列テーブルへのポインタ  
モード  
開始X座標  
開始Y座標  
色

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

lmgID

描画面の画面番号

\*string

文字列テーブルへのポインタ

mode

文字描画モード

31	16 15	12 11	8 7	4 3	0
-	Vsize	Hsize	描画モード		

Hsize、VsizeはそれぞれX方向、Y方向の文字倍率です。

Hsize,Vsize	対応数	内容
STR_SIZE_1	0	X(Y)方向を倍率1倍で描画
STR_SIZE_2	1	X(Y)方向を倍率2倍で描画
STR_SIZE_3	2	X(Y)方向を倍率3倍で描画
STR_SIZE_4	3	X(Y)方向を倍率4倍で描画

文字描画モード	対応数	内容
BACKCOL_NOWRITE	0	文字部分のみcolF色で描画
BACKCOL_NOWRITE_REVERSE	1	文字背景のみcolF色で描画
BACKCOL_WRITE	2	文字部分をcolF色、背景をcolBで描画
BACKCOL_WRITE_REVERSE	3	文字部分をcolB色、背景をcolFで描画

sx

開始X座標

sy

開始Y座標

col

書き込みデータ。PutAnkString()コマンドの詳細説明を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

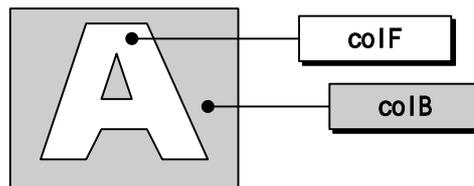
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

書き込みデータ col のビット構成



```
PutString(devID, ImgID, "漢字が使用できません", BACKCOL_NOWRITE, 128, 64, 15)
```

```
PutString(devID, ImgID, "半角かも使用できません", BACKCOL_NOWRITE, 128, 80, 15)
```

# ClearBitmap

## 画像描画(範囲・色指定)

### 機能

クリア領域を指定してビットマップ画面やディスプレイ画面などの表示系画面をクリアします。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
ClearBitmap(
    DEVID    devID ,
    int      imgID ,
    int      sx ,
    int      sy ,
    int      xleng ,
    int      yleng ,
    int      col
)
```

### コメント

デバイスID  
画面番号  
開始X座標  
開始Y座標  
X方向長さ  
Y方向長さ  
クリアデータ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgID**  
描画画面の画面番号

**sx**  
開始X座標

**sy**  
開始Y座標

**xleng**  
X方向の長さ

**yleng**  
Y方向の長さ

**col**  
クリアするデータ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ClearScreen

## 画面クリア

### 機能

ビットマップ画面やディスプレイ画面などの表示系画面を全画面クリアします。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
ClearScreen(
    DEVID  devID ,
    int    ImgID
)
```

### コメント

デバイスID  
画面番号

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgID**  
描画画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外（不当画面番号エラー）
2 0	パラメータエラー
4 1	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## startGetIPTime

## タイマスタート

### 機能

時間計測のためのタイマを起動します。

### コーディング

```
int
startGetIPTime(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

以下に処理時間の計測例を示します。

```
unsigned long time,stime,etime;

startGetIPTime(_devID);           // タイマスタート

stime = timeGetIPTime(_devID);    // 開始時間
. . . . .
処理
. . . . .

etime = timeGetIPTime(_devID);    // 処理終了時間

stopGetIPTime(_devID);           // タイマ停止

time = etime - stime;             // 処理時間算出
printf("Total Time : %d mS\n",time);
```

## timeGetIPTime

## タイマ時間読み出し

### 機能

時間計測のためのタイマの現在の経過時間を読み出します。読み出される時間は、デフォルトでmS単位ですが、SetIPTimeMode()コマンドにより、 $\mu$ S, mS, S単位での計測が可能です。

### コーディング

```
unsigned long
timeGetIPTime(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は unsigned long 型で返される現在の経過時間です。詳細を下表に示します。

リターン値	内容
0xFFFFFFFF以外	経過時間 (デフォルト : mS Max値 : 2,061,586[ms])
0xFFFFFFFF	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

以下に処理時間の計測例を示します。

```
unsigned long time,stime,etime;

startGetIPTime(_devID);           // タイマスタート

stime = timeGetIPTime(_devID); // 開始時間

. . . . .
処理
. . . . .

etime = timeGetIPTime(_devID); // 処理終了時間

stopGetIPTime(_devID);           // タイマ停止

time = etime - stime;           // 処理時間算出
printf("Total Time : %d mS¥n",time);
```

# stopGetIPTime

## タイマストップ

### 機能

時間計測のためのタイマを停止します。

### コーディング

```
int
stopGetIPTime(
    DEVID devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラーの解析」の章を参照して下さい。

### 詳細情報

以下に処理時間の計測例を示します。

```
unsigned long time,stime,etime;

startGetIPTime(_devID);           // タイマスタート
stime = timeGetIPTime(_devID);    // 開始時間
. . . . .
処理
. . . . .
etime = timeGetIPTime(_devID);    // 処理終了時間
stopGetIPTime(_devID);           // タイマ停止
time = etime - stime;             // 処理時間算出
printf("Total Time : %d mS¥n",time);
```

# IPSleep

## タイマウェイト

### 機能

任意時間ウェイトします。

### コーディング

```
int
IPSleep(
    DEVID          devID ,
    unsigned long  wait
)
```

### コメント

デバイスID  
ウェイト時間

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

wait

ウェイト時間。mS単位で設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## SetIPTimeMode

## 時間計測モード設定

### 機能

timeGetIPTime()コマンドによる時間計測のモード設定を行います。なお、デフォルトは mS 単位に設定されています。

### コーディング

```
int
SetIPTimeMode(
    DEVID
    enum IPTimeMode
)
```

devID ,  
mode

### コメント

デバイスID  
時間計測モード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

時間計測モード

時間計測モード	対応数	説明
U_SEC	0	μS 単位の計測
M_SEC	1	mS 単位の計測
S_SEC	2	秒 単位の計測

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

以下に処理時間の計測例を示します。

```
unsigned long time,stime,etime;

SetIPTimeMode(_devID,U_SEC);           // μS 計測設定

startGetIPTime(_devID);                 // タイマスタート

stime = timeGetIPTime(_devID);          // 開始時間

. . . . .
処理
. . . . .

etime = timeGetIPTime(_devID);          // 処理終了時間

stopGetIPTime(_devID);                  // タイマ停止

time = etime - stime;                    // 処理時間算出
printf("Total Time : %d uS¥n",time);
```

# SetIPTime

## 時刻の登録

### 機能

μITRON上の仮想リアルタイムクロックに対して、時刻を登録します。GetIPTime()コマンドでは、本コマンドで登録した時刻から積算された時刻が読み出されます。

### コーディング

```
int
SetIPTime(
  DEVID
  IP_SYSTEM_TIME
)
```

devID ,  
\*time

### コメント

デバイスID  
時刻構造体

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*time

仮想リアルタイムクロックに登録する時刻（西暦/月/日/時/分/秒）を表す構造体。曜日のデータは登録されません。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了。RTCに登録する時刻（西暦/月/日/時/分/秒）が不適切。

### 詳細情報

IP\_SYSTEM\_TIMEのフォーマットを、下記に示します。

```
typedef struct {
  short  year;          /* 西暦          */
  short  month;         /* 月            */
  short  day_of_week;  /* 曜日          */
  short  day;          /* 日            */
  short  hour;         /* 時(24時間)    */
  short  minute;       /* 分            */
  short  second;       /* 秒            */
  short  milliseconds; /* mS            */
} IP_SYSTEM_TIME;
```

メンバ名	説明														
year	年（西暦）														
month	月（1-1月，2-2月・・・）														
day_of_week	曜日 <table border="1" style="margin-left: 20px;"> <tr> <td>日</td><td>月</td><td>火</td><td>水</td><td>木</td><td>金</td><td>土</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td> </tr> </table>	日	月	火	水	木	金	土	0	1	2	3	4	5	6
日	月	火	水	木	金	土									
0	1	2	3	4	5	6									
day	日（1～31）														
hour	時（24時間制：0～23）														
minute	分（0～59）														
second	秒（0～59）														
milliseconds	m sec（10m sec単位：0～990）。設定できません。														

# GetIPTime

## 時刻の取得

### 機能

μITRON上の仮想リアルタイムクロックの現在時刻を取得します。正確には、SetIPTime()コマンドで登録した時刻から積算された時刻が読み出されます。従って、SetIPTime()コマンドで時刻をあらかじめ登録する必要があります。システムがイニシャライズされると 2000/01/01 0:0:0:0 にリセットされます。

### コーディング

```
int
GetIPTime(
    DEVID          devID ,
    IP_SYSTEM_TIME *time
)

```

### コメント

デバイスID  
時刻構造体

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*time**  
仮想リアルタイムクロックに登録する時刻（西暦/月/日/曜日/時/分/秒）を表す構造体。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

IP\_SYSTEM\_TIMEのフォーマットを、下記に示します。

```
typedef struct {
    short   year;           /* 西暦           */
    short   month;         /* 月             */
    short   day_of_week;   /* 曜日           */
    short   day;           /* 日             */
    short   hour;          /* 時(24時間)     */
    short   minute;        /* 分             */
    short   second;        /* 秒             */
    short   milliseconds; /* mS             */
} IP_SYSTEM_TIME;
```

メンバ名	説明														
year	年（西暦）														
month	月（1-1月，2-2月・・・）														
day_of_week	曜日 <table border="1"> <tr> <td>日</td> <td>月</td> <td>火</td> <td>水</td> <td>木</td> <td>金</td> <td>土</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> </tr> </table>	日	月	火	水	木	金	土	0	1	2	3	4	5	6
日	月	火	水	木	金	土									
0	1	2	3	4	5	6									
day	日（1～31）														
hour	時（24時間制：0～23）														
minute	分（0～59）														
second	秒（0～59）														
milliseconds	m sec（10m sec単位：0～990）														

## vpxEnableCameraPrefetch

## プリフェッチ処理の有効化

### 機能

プリフェッチ処理の画面を登録し、プリフェッチ処理を有効にします。

### コーディング

```
int
vpxEnableCameraPrefetch(
    DEVID    devID ,
    int      ImgID0 ,
    int      ImgID1
)
```

### コメント

デバイスID  
画面番号#0  
画面番号#1

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID0

画面番号#0。必ず、ImgID1の画面と別のチャンネルの画面を指定して下さい。

ImgID1

画面番号#1。必ず、ImgID0の画面と別のチャンネルの画面を指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	画像メモリチャンネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

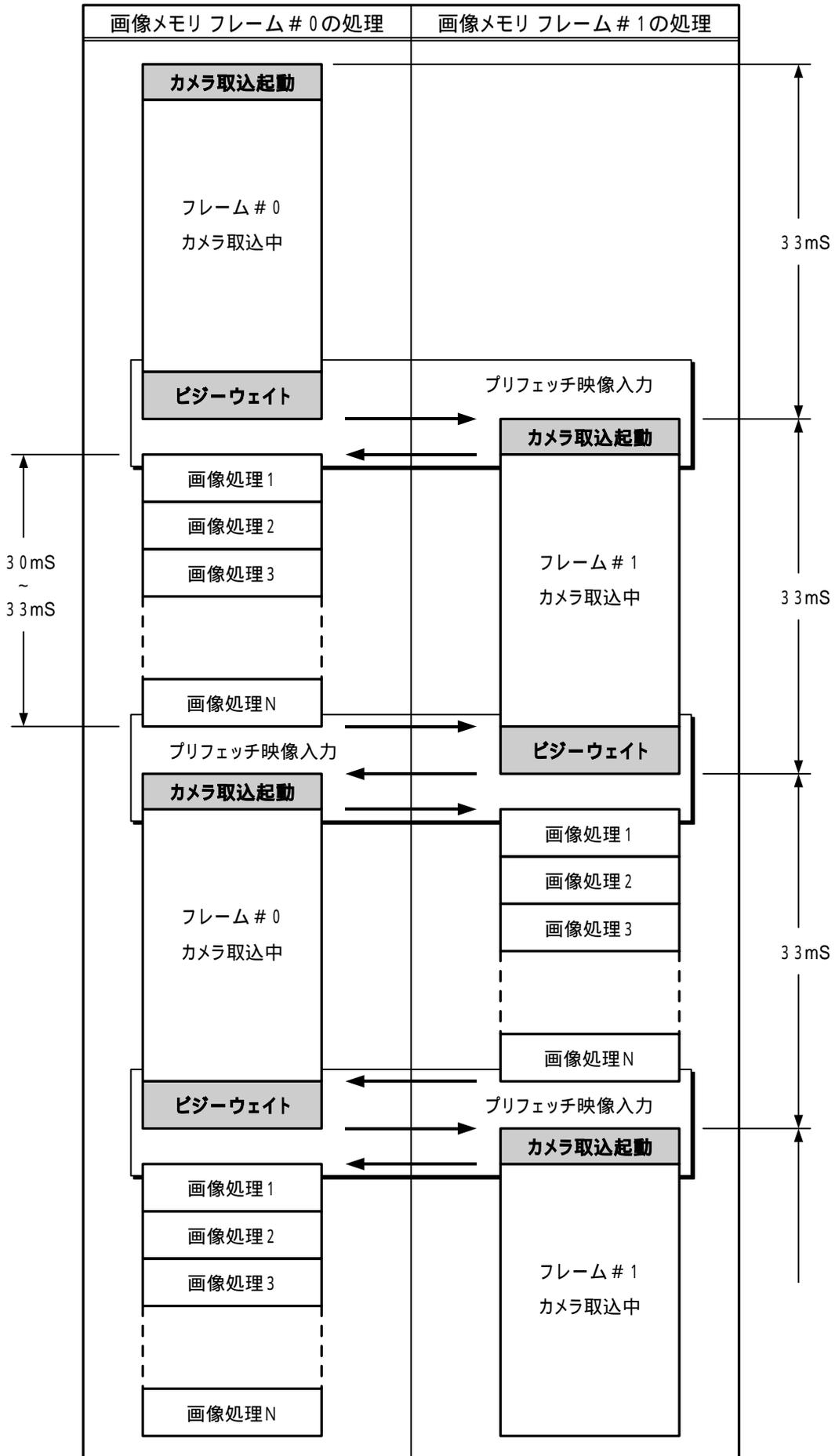
映像を入力すると同時に別のチャンネルの画像メモリの画像処理を行うことで、ビデオレートの処理を実現することができます。それらの処理をプリフェッチ処理といいます。

画像処理コマンドでは、プリフェッチ処理を簡単に実行するために、

```
vpxEnableCameraPrefetch()
vpxGetCameraPrefetch()
vpxDisableCameraPrefetch()
```

コマンドを用意しています。

vpxEnableCameraPrefetch()コマンドで設定した画面#0と画面#1は、vpxGetCameraPrefetch()コマンドが実行されるたびに交互に切換ながら映像が入力されます。



## vpxGetCameraPrefetch

## プリフェッチ映像入力

### 機能

次のカメラ映像入力を起動し、すでに入力終了した画面の画面番号 (ImgID) を出力します。  
vpxEnableCameraPrefetch() コマンドで設定した画面 # 0 と画面 # 1 を交互に切替ながら入力します。画像処理は、vpxGetCameraPrefetch() コマンドで出力される画面番号の画面に対して処理を行って下さい。

### コーディング

```
int
vpxGetCameraPrefetch(
    DEVID      devID ,
    int        *ImgID
)
```

### コメント

デバイスID  
画面番号 (出力)

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*ImgID**  
すでに入力が終了した画面の画面番号が出力されます。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	プリフェッチ処理が無効
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

#### プリフェッチ処理のプログラミング例

```
// 画面確保 (別々のチャンネルになるように確保して下さい)
_ImgID0= AllocImg(_devID, IMG_FS_512H_512V); // フレーム # 0
_ImgID1= AllocImg(_devID, IMG_FS_512H_512V); // フレーム # 1

_ImgID2= AllocImg(_devID, IMG_FS_512H_512V);

vpxEnableCameraPrefetch(_devID, _ImgID0, _ImgID1); // プリフェッチ処理の画面設定と有効化

for(;;){
    vpxGetCameraPrefetch(_devID, &ImgID); // プリフェッチ映像入力
                                           // ImgIDに入力が終了した画面番号が出力されます

    // この部分以降に画像処理を入れて下さい。
    // 例として2値化処理を示します。
    IP_Binarize(_devID, ImgID, _ImgID2); // 2値化の場合
}

vpxDisableCameraPrefetch(_devID); // プリフェッチ処理の終了
```

## vpxDisableCameraPrefetch

## プリフェッチ処理の終了

### 機能

プリフェッチ処理を終了します。

### コーディング

```
int  
vpxDisableCameraPrefetch(  
    DEVID          devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# RegistIntelIIPModule

## インテリジェントモジュールの登録

### 機能

オンボードCPUで実行されるインテリジェントモジュールを登録します。登録したモジュールは、ExecuteIntelIIPModule()コマンドで実行します。  
インテリジェントモジュールは256個まで登録できます。

### コーディング

```
int
RegistIntelIIPModule(
    DEVID
    int
    unsigned long
    int
)
```

```
devID ,
module ,
module_addr ,
opt
```

### コメント

デバイスID  
モジュール番号  
モジュールのアドレス  
オプション

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**module**

登録するインテリジェントモジュールのモジュール番号

モジュール番号	説明
0	モジュール番号 # 0
1	モジュール番号 # 1
2	モジュール番号 # 2
255	モジュール番号 # 255

**module\_addr**

インテリジェントモジュールのアドレス。オンボードCPU上のインテリジェントモジュールのプログラムセクションのアドレスを指定して下さい。

**opt**

オプション。「0」を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	モジュールID設定範囲外

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## ExecuteIntelIPModule

## インテリジェントモジュールの実行

### 機能

RegistIntelIPModule() コマンドで登録したインテリジェントモジュールの実行を行います。  
 インテリジェントモジュールは画像処理コマンドの一部として動作するため、60秒以上(デフォルト)の処理ではタイムアウトが発生してしまいますので注意が必要です。  
 このコマンドではインテリジェントモジュールにC言語の関数形式でパラメータを渡すことができます。

### コーディング

```
int
ExecuteIntelIPModule(
    DEVID
    int
    MODULE_PARAM_TBL
    void
)
```

```
devID ,
module ,
*prmtbl ,
*ovl
```

### コメント

デバイスID  
 モジュール番号  
 モジュールパラメータテーブル  
 オーラップI/O構造体

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### module

実行するインテリジェントモジュールのモジュール番号

モジュール番号	説明
0	モジュール番号 # 0
1	モジュール番号 # 1
2	モジュール番号 # 2
255	モジュール番号 # 255

#### \*prmtbl

モジュールのパラメータテーブルのポインタ。初期化及び設定についての詳細は、InitIPParamTable(), SetIntegerParam(), SetFloatParam() コマンドなどのモジュールサポートコマンドを参照して下さい。

#### \*ovl

NULLを設定して下さい。

### リターン値

このコマンドのリターンコードは登録モジュールから返されるリターンコードです。型は int 型のみで、登録モジュールから返されるリターンコードがそれ以外の場合、正しくデータが渡されないので注意してください。

### 詳細情報

# CreateIPTask

## ユーザタスクの生成

### 機能

ユーザタスクの生成を行います。タスクの生成に成功するとタスクIDをリターン値として返します。

### コーディング

```
int
CreateIPTask(
    DEVID
    CREATE_TASK_TBL
)
```

devID ,  
\*tbl

### コメント

デバイスID  
タスク生成情報テーブル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*tbl

タスク生成情報テーブル。この情報テーブルに設定された情報をもとにタスクを生成します。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	タスクID
0	異常終了
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	スタックサイズ, プライオリティ設定範囲外
4 8	パラメータバッファメモリ確保失敗
2 6 0	タスク生成失敗
2 6 1	登録できるタスク数を超えている
2 6 2	登録済みのタスクIDが生成された

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

CREATE\_TASK\_TBLのフォーマット及び内容を、下記に示します。

vpxsys.hで宣言しています

```
typedef struct {
    unsigned long task_addr;
    int priority;
    unsigned long pbuff_size;
    unsigned long stack_size;
    int param_opt;
    int task_opt;
} CREATE_TASK_TBL;
```

タスク情報生成テーブル	説明																			
task_addr	<p>タスクの先頭アドレス</p> <table border="1"> <thead> <tr> <th>task_addr</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>割込起動モジュールに対してパラメータを引き渡す機能のあるフレームを使用します。この機能を選択した場合は、RegistIPTask( )コマンドでモジュールのアドレスを設定して下さい。</td> </tr> <tr> <td>0以外</td> <td>指定したモジュールを直接起動します。割込起動モジュールに対してパラメータを引き渡すことができませんが、RegistIPTask( )コマンドでのモジュールのアドレス指定も必要ありません。</td> </tr> </tbody> </table>	task_addr	内容	0	割込起動モジュールに対してパラメータを引き渡す機能のあるフレームを使用します。この機能を選択した場合は、RegistIPTask( )コマンドでモジュールのアドレスを設定して下さい。	0以外	指定したモジュールを直接起動します。割込起動モジュールに対してパラメータを引き渡すことができませんが、RegistIPTask( )コマンドでのモジュールのアドレス指定も必要ありません。													
task_addr	内容																			
0	割込起動モジュールに対してパラメータを引き渡す機能のあるフレームを使用します。この機能を選択した場合は、RegistIPTask( )コマンドでモジュールのアドレスを設定して下さい。																			
0以外	指定したモジュールを直接起動します。割込起動モジュールに対してパラメータを引き渡すことができませんが、RegistIPTask( )コマンドでのモジュールのアドレス指定も必要ありません。																			
priority	<p>タスクのプライオリティ</p> <table border="1"> <thead> <tr> <th>プライオリティ</th> <th>対応数</th> <th>説明</th> <th rowspan="6">                     高い                        低い                 </th> </tr> </thead> <tbody> <tr> <td>TASK_PRI_HIGHEST</td> <td>-2</td> <td>ノーマル - 2</td> </tr> <tr> <td>TASK_PRI_ABOVE_NORMAL</td> <td>-1</td> <td>ノーマル - 1</td> </tr> <tr> <td>TASK_PRI_NORMAL</td> <td>0</td> <td>ノーマル (初期値)</td> </tr> <tr> <td>TASK_PRI_BELOW_NORMAL</td> <td>1</td> <td>ノーマル + 1</td> </tr> <tr> <td>TASK_PRI_LOWEST</td> <td>2</td> <td>ノーマル + 2</td> </tr> </tbody> </table>	プライオリティ	対応数	説明	高い  低い	TASK_PRI_HIGHEST	-2	ノーマル - 2	TASK_PRI_ABOVE_NORMAL	-1	ノーマル - 1	TASK_PRI_NORMAL	0	ノーマル (初期値)	TASK_PRI_BELOW_NORMAL	1	ノーマル + 1	TASK_PRI_LOWEST	2	ノーマル + 2
プライオリティ	対応数	説明	高い  低い																	
TASK_PRI_HIGHEST	-2	ノーマル - 2																		
TASK_PRI_ABOVE_NORMAL	-1	ノーマル - 1																		
TASK_PRI_NORMAL	0	ノーマル (初期値)																		
TASK_PRI_BELOW_NORMAL	1	ノーマル + 1																		
TASK_PRI_LOWEST	2	ノーマル + 2																		
pbuff_size	パラメータバッファのサイズ(バイト)。必要以外は、「0」を指定して下さい																			
stack_size	スタックサイズ(バイト)																			
param_opt	<p>パラメータオプション</p> <p>task_addr 情報に「0」を設定した場合、割込起動モジュールに対してパラメータを引き渡す機能を使用するかどうか指定します。task_addr 情報が「0」以外の場合は、「0」を指定して下さい。</p> <table border="1"> <thead> <tr> <th>param_opt</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>パラメータを渡さない</td> </tr> <tr> <td>1</td> <td>パラメータを渡す</td> </tr> </tbody> </table>	param_opt	内容	0	パラメータを渡さない	1	パラメータを渡す													
param_opt	内容																			
0	パラメータを渡さない																			
1	パラメータを渡す																			
task_opt	<p>タスクオプション</p> <p>未使用。0を設定して下さい。</p>																			

# CreateInterruptLink

## 割込オブジェクトの生成

### 機能

割込リンクオブジェクトの生成。本コマンドにより、P I Oなどの割込オブジェクトからの割込によりウェイクアップ等の動作を行うように設定します。

### コーディング

```
int
CreateInterruptLink(
  DEVID          devID ,
  INT_DEVICE_OBJ *obj ,
  int            taskID ,
  int           event ,
  int           opt
)
```

### コメント

デバイス I D  
割込デバイスオブジェクト  
タスク I D  
割込時の動作  
オプション

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**\*obj**  
割込デバイスオブジェクト情報。リンクする割込デバイスの情報を設定します。

**taskID**  
割込デバイスオブジェクトとリンクするタスクのタスク I D。

**event**  
割込時の動作を指定します。

割込イベント	対応数	説明
INTEVENT_NOP	0	イベントなし
INTEVENT_WAKEUP	1	ウェイクアップ

**opt**  
現在、未使用。「0」を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	割込ビット設定範囲外
2 1	タスク I D設定範囲外
2 6 2	すでに登録されている割込リンク先に登録しようとしている
2 6 3	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

INT\_DEVICE\_OBJ のフォーマット及び内容を、下記に示します。

vpxsys.hで宣言しています

```
typedef struct {
    int          intdev;
    int          intbit;
    int          evtpri;
    int          opt;
} INT_DEVICE_OBJ;
```

割込デバイスオブジェクト	説明												
intdev	<p>割込デバイス番号</p> <table border="1"> <thead> <tr> <th>デバイス番号</th> <th>対応数</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>INTDEV_UNLINK</td> <td>0</td> <td>割込デバイスなし(現在指定不可)</td> </tr> <tr> <td>INTDEV_PCI</td> <td>1</td> <td>P C I - C O M</td> </tr> <tr> <td>INTDEV_PIO</td> <td>2</td> <td>P I O割込</td> </tr> </tbody> </table>	デバイス番号	対応数	説明	INTDEV_UNLINK	0	割込デバイスなし(現在指定不可)	INTDEV_PCI	1	P C I - C O M	INTDEV_PIO	2	P I O割込
デバイス番号	対応数	説明											
INTDEV_UNLINK	0	割込デバイスなし(現在指定不可)											
INTDEV_PCI	1	P C I - C O M											
INTDEV_PIO	2	P I O割込											
intbit	<p>割込デバイスの対象ビット。以下にデバイス番号に対する対応ビットを示します。</p> <table border="1"> <thead> <tr> <th>デバイス番号</th> <th>対応数</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>INTDEV_PCI</td> <td>16-23 (0-31)</td> <td>P C I - C O M割込ビットに対応しています。ユーザに開放されているのは、16～23までです。それ以外は設定しないで下さい。</td> </tr> <tr> <td rowspan="2">INTDEV_PIO</td> <td></td> <td>P I Oのビットに対応しています。</td> </tr> <tr> <td>0-3</td> <td>SVP-330</td> </tr> </tbody> </table>	デバイス番号	対応数	説明	INTDEV_PCI	16-23 (0-31)	P C I - C O M割込ビットに対応しています。ユーザに開放されているのは、16～23までです。それ以外は設定しないで下さい。	INTDEV_PIO		P I Oのビットに対応しています。	0-3	SVP-330	
デバイス番号	対応数	説明											
INTDEV_PCI	16-23 (0-31)	P C I - C O M割込ビットに対応しています。ユーザに開放されているのは、16～23までです。それ以外は設定しないで下さい。											
INTDEV_PIO		P I Oのビットに対応しています。											
	0-3	SVP-330											
evtpri	<p>割込動作のプライオリティ。以下の4種類の設定が可能です。現在未使用なため、「0」を設定して下さい。</p> <table border="1"> <thead> <tr> <th>プライオリティ</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>0</td> <td rowspan="4"> <p style="text-align: center;">高い</p>  <p style="text-align: center;">低い</p> </td> </tr> <tr> <td>1</td> </tr> <tr> <td>2</td> </tr> <tr> <td>3</td> </tr> </tbody> </table>	プライオリティ	説明	0	<p style="text-align: center;">高い</p>  <p style="text-align: center;">低い</p>	1	2	3					
プライオリティ	説明												
0	<p style="text-align: center;">高い</p>  <p style="text-align: center;">低い</p>												
1													
2													
3													
opt	オプション(「0」を指定)												

## DeleteInterruptLink

## 割込オブジェクトの削除

### 機能

CreateInterruptLink()により生成した割込リンクオブジェクトを削除します。本コマンドにより、P I Oなどの割込オブジェクトからの割込による動作を解除します。

### コーディング

```
int
DeleteInterruptLink(
    DEVID      devID ,
    INT_DEVICE_OBJ *obj
)
```

### コメント

デバイス I D  
割込デバイスオブジェクト

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

\*obj

割込デバイスオブジェクト情報。リンクを解除する割込デバイスの情報を設定します。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	割込ビット設定範囲外
2 6 0	割込ビット無効化失敗

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# EnableInterruptObject

## 割込オブジェクトの有効化

### 機能

本コマンドにより、P I Oなどの割込オブジェクトからの割込動作を有効にします。

### コーディング

```
int
EnableInterruptObject(
    DEVID      devID ,
    INT_DEVICE_OBJ *obj ,
    int        opt
)
```

### コメント

デバイスID  
割込デバイスオブジェクト  
オプション

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*obj

割込デバイスオブジェクト情報。リンクする割込デバイスの情報を設定します。

opt

現在、未使用。「0」を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	割込ビット設定範囲外
2 6 0	割込ビット有効化不能

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

「INT\_DEVICE\_OBJ」構造体の詳細は、CreateInterruptLink()コマンドの詳細情報を参照して下さい。

# DisableInterruptObject

## 割込オブジェクトの無効化

### 機能

本コマンドにより、P I Oなどの割込オブジェクトからの割込による動作を無効にします。

### コーディング

```
int
DisableInterruptObject(
    DEVID      devID ,
    INT_DEVICE_OBJ *obj
)
```

### コメント

デバイス I D  
割込デバイスオブジェクト

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

\*obj

割込デバイスオブジェクト情報。リンクを解除する割込デバイスの情報を設定します。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	割込ビット設定範囲外
2 6 0	割込ビット無効化不能

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

「INT\_DEVICE\_OBJ」構造体の詳細は、CreateInterruptLink()コマンドの詳細情報を参照して下さい。

# RegistIPTask

## 割込モジュールの登録

### 機能

オンボードCPUで実行される割込起動モジュールを登録します。登録するモジュールは、初期化部とPIOなどのデバイスからの割込で起動される部分の2つのペアです。

### コーディング

```
int
RegistIPTask(
    DEVID
    int
    unsigned long
    unsigned long
    int
)
```

```
devID ,
taskID ,
init_task ,
pioint_task ,
opt
```

### コメント

```
デバイスID
タスクID
初期化モジュールアドレス
PIO割込起動モジュールアドレス
オプション
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### taskID

登録する割込モジュールのタスクID。CreateIPTask()コマンドで生成したタスクIDを設定して下さい。設定範囲は1～64です。

#### init\_task

初期化モジュールのアドレス。オンボードCPU上のプログラムセクションのアドレスを指定して下さい。初期化モジュールが不必要な場合、0を設定して下さい。

#### pioint\_task

PIOからの割込で起動されるモジュールのアドレス。オンボードCPU上のプログラムセクションのアドレスを指定して下さい。0を設定するとNOPモジュールが設定されます。

#### opt

オプション。通常は「0」を指定して下さい。

タスクオプション	対応数	説明
TASK_NO_OPTION	0x00000000	・ウェイクアップ要求をキューイングする。 ・スタートタスク実行後、ハード的に割込マスクを解除し、PIOからの割込によりタスクがウェイクアップされるようにします。
CANCEL_WAKEUP	0x00000001	タスク実行中にキューイングされたウェイクアップ要求をキャンセルします。
DISABLE_INTERRUPT	0x00000100	PIOからの割込をハード的にマスクし、割込によりタスクがウェイクアップされないようにします。

複数のオプションを指定する場合は、論理和をとって指定して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	タスクID設定範囲外
2 6 0	割込の有効化失敗
2 6 3	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

## Start IPTaskwithParam

## 割込モジュールのサービス開始

### 機能

RegistIPTask()コマンドで登録された割込初期化モジュール(init\_task)を実行し、割込起動モジュール(pioint\_task)に対するサービスを開始します。RegistIPTask()コマンドで登録したtaskIDに対応するinit\_taskが実行された後、対応する割込のハード的なマスクが解除され、ウェイクアップコマンドでpioint\_taskが起動される状態になります。

また、このコマンドではinit\_taskにC言語の関数形式でパラメータをわたすことができます。

### コーディング

```
int
Start IPTaskwithParam(
    DEVID
    int
    MODULE_PARAM_TBL
)
```

```
devID ,
taskID ,
*prmtbl
```

### コメント

```
デバイスID
タスクID
モジュールパラメータテーブル
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

taskID

割込モジュールのタスクID。CreateIPTask()コマンドで生成したタスクIDを設定して下さい。設定範囲は1～64です。

\*prmtbl

モジュールのパラメータテーブルのポインタアドレス。初期化及び設定についての詳細は、InitIPParamTable(),SetIntegerParam(),SetFloatParam()コマンドなどのモジュールサポートコマンドを参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了
-51～-9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	タスクID設定範囲外
263	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## WakeupIPTaskwithParam

## 割込モジュールの実行

### 機能

本コマンドの起動オプション(opt)がSELF\_WAKEUPの場合、RegistIPTask()コマンドで登録された割込起動モジュール(pioint\_task)を自己起動します。また、本コマンドの起動オプション(opt)がPIO\_WAKEUPの場合、RegistIPTask()コマンドで登録された割込起動モジュール(pioint\_task)をtaskIDで示されるP I Oの割込により起動します。

また、このコマンドではpioint\_taskにC言語の関数形式でパラメータをわたすことができます。

### コーディング

```
int
WakeupIPTaskwithParam(
    DEVID
    int
    MODULE_PARAM_TBL
    int
)
```

```
devID ,
taskID ,
*prmtbl ,
opt
```

### コメント

```
デバイスID
タスクID
モジュールパラメータテーブル
起動オプション
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

taskID

割込モジュールのタスクID。CreateIPTask()コマンドで生成したタスクIDを設定して下さい。設定範囲は1～64です。

\*prmtbl

モジュールのパラメータテーブルのポケットアドレス。初期化及び設定についての詳細は、InitIPParamTable(),SetIntegerParam(),SetFloatParam()コマンドなどのモジュールサポートコマンドを参照して下さい。

opt

起動オプション

起動オプション	説明
SELF_WAKEUP	自己起動します
PIO_WAKEUP	P I O入力の割込で起動します

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	タスクID設定範囲外
2 6 3	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# Wait for IPTaskSignal

## 割込モジュールからの終了シグナルウェイト

### 機能

オンボードCPUの割込モジュールが終了シグナルを発信するまでウェイトします。オンボードCPUの割込モジュールは、SendSignaltoPC()コマンドで終了シグナルを発信し、割込モジュールの処理が終了したことをPCに伝えます。

しかし、SELF\_WAKEUPモードのWakeupIPTaskwithParam()コマンドとWaitforIPTaskSignal()コマンドにより、タスクの制御を行うとWakeup動作とWait動作を単一のドライバ内で行うことができないため、WakeupIPTaskwithParam()コマンドとWaitforIPTaskSignal()コマンドの間に発生したオンボードCPUからのシグナルに対して応答する事ができず、そのスレッドは、別のスレッドからCancelIPTaskWait()コマンドを発行するまでデッドロック状態になります。そのような状態になることが予想されるアプリケーションではWakeupIPTaskExt()コマンドを使用して下さい。

### コーディング

```
int
WaitforIPTaskSignal(
    DEVID
    int
    int
    int
    void
)
```

```
devID ,
signal ,
mode ,
opt ,
*ovl
```

### コメント

```
デバイスID
ユーザシグナル番号
モード
オプション
ウェイト情報
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**signal**

ユーザシグナル番号。ローカルCPUからPCへのシグナルを送信する場合、ローカルCPU側からのSendSignaltoPC()コマンドのsignalパラメータに対応するユーザシグナル番号に応答して本コマンドのウェイトが解除されます。

ユーザシグナル番号	対応数	説明
PCI_SIGNAL_USER_0	16	ユーザシグナル# 0
PCI_SIGNAL_USER_1	17	ユーザシグナル# 1
PCI_SIGNAL_USER_2	18	ユーザシグナル# 2
PCI_SIGNAL_USER_7	23	ユーザシグナル# 7

**mode**

モード（現在未使用）。「0」を設定して下さい。

**opt**

オプション（現在未使用）。「0」を設定して下さい。

**\*ovl**

WAIT\_SIGNAL\_INFOテーブルのアドレスを指定して下さい。NULLを指定するとシグナルが入力されるまでウェイトします。WAIT\_SIGNAL\_INFOテーブルの詳細は、詳細情報を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	ユーザシグナル番号設定範囲外
2 6 3	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

WAIT\_SIGNAL\_INFO のフォーマット及び内容を、下記に示します。

vpxsys.hで宣言しています

```
typedef struct {
    long    timeout;
    long    reserv[15];
} WAIT_SIGNAL_INFO;
```

ウェイトシグナル情報	説明
timeout	タイムアウト時間 ( m s )
reserve	予約。全て 0 クリアして下さい。

## ReadyforWaitIPTaskSignal

### 割込モジュールからの終了シグナルウェイト準備

#### 機能

本コマンドはV P - 9 1 0 コマンドでの互換性のためのダミーとしてインプリメントされています。本コマンドを発行しても何も実行しません。

#### コーディング

```
int
ReadyforWaitIPTaskSignal (
    DEVID          devID ,
    int            signal
)
```

#### コメント

デバイスID  
ユーザシグナル番号

#### パラメータ

##### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

##### signal

ユーザシグナル番号。ローカルCPUからPCへのシグナルを送信する場合、ローカルCPU側からのSendSignaltoPC()コマンドのsignalパラメータに対応するユーザシグナル番号に回答して本コマンドのウェイトが解除されます。

ユーザシグナル番号	対応数	説明
PCI_SIGNAL_USER_0	16	ユーザシグナル# 0
PCI_SIGNAL_USER_1	17	ユーザシグナル# 1
PCI_SIGNAL_USER_2	18	ユーザシグナル# 2
PCI_SIGNAL_USER_7	23	ユーザシグナル# 7

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	ユーザシグナル番号設定範囲外
2 6 3	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

## WakeupIPTaskExt

## 割込モジュールの実行2

### 機能

機能的には、WakeupIPTaskwithParam() コマンドを起動オプション(opt)がSELF\_WAKEUPで実行し、WaitforIPTaskSignal() コマンドで登録されたP I O 割込起動モジュール(pioint\_task)からのシグナルをウェイトする動作と同じです。しかし、本コマンドでは、Wakeup動作とWait動作を単一のドライバ内で行うため、WakeupIPTaskwithParam() コマンドとWaitforIPTaskSignal() コマンドの間に発生したオンボードC P Uからのシグナルに対しても確実に応答する事ができます。

### コーディング

```
int
WakeupIPTaskExt(
    DEVID          devID ,
    int            taskID ,
    MODULE_PARAM_TBL *prmtbl ,
    int           opt ,
    void          *ovl
)
```

### コメント

デバイス I D  
タスク I D  
モジュールパラメータテーブル  
起動オプション  
未使用

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**taskID**  
割込モジュールのタスク I D。CreateIPTask() コマンドで生成したタスク I Dを設定して下さい。設定範囲は1 ~ 64です。

**\*prmtbl**  
モジュールのパラメータテーブルの packets アドレス。初期化及び設定についての詳細は、InitIPParamTable(), SetIntegerParam(), SetFloatParam() コマンドなどのモジュールサポートコマンドを参照して下さい。

**opt**  
起動オプション

起動オプション	説明
SELF_WAKEUP	自己起動します
PIO_WAKEUP	P I O 入力の割込で起動します

**\*ovl**  
N U L L を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μ I TORN サービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	タスク I D 設定範囲外
2 6 3	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドでウェイトを解除する場合、ローカルC P U側からの SendSignaltoPC() コマンドの signal パラメータに対応するユーザシグナル番号を PCI\_SIGNAL\_USER\_0 にして下さい。

## CancelIPTaskWait

## 終了シグナルウェイトの解除

### 機能

本コマンドはV P - 9 1 0 コマンドでの互換性のためのダミーとしてインプリメントされています。本コマンドを発行しても何も実行しません。

### コーディング

```
int
CancelIPTaskWait(
    DEVID
    int
    int
)
```

```
devID ,
signal ,
mode
```

### コメント

デバイスID  
ユーザシグナル番号  
モード

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**signal**

ローカルCPUからPCへのシグナルを送信する場合の番号。

ユーザシグナル番号	対応数	説明
PCI_SIGNAL_USER_0	16	ユーザシグナル# 0
PCI_SIGNAL_USER_1	17	ユーザシグナル# 1
PCI_SIGNAL_USER_2	18	ユーザシグナル# 2
PCI_SIGNAL_USER_7	23	ユーザシグナル# 7

**mode**

モード（現在未使用）。「0」を設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	ユーザシグナル番号設定範囲外
2 6 3	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# StartIPTask

## 割込モジュールのサービス開始

### 機能

StartIPTaskwithParam() コマンドと同等の処理を行いますが、本コマンドでは、パラメータを設定することはできません。また、本コマンドを使用する場合は、必ずWakeupIPTask() コマンドをペアで使用して下さい。WakeupIPTaskwithParam() コマンドとペアで使用しないで下さい。

### コーディング

```
int
StartIPTask(
    DEVID      devID ,
    int        taskID
)
```

### コメント

デバイスID  
タスクID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

taskID

割込モジュールのタスクID。CreateIPTask() コマンドで生成したタスクIDを設定して下さい。設定範囲は1～64です。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了
-51～-9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	タスクID設定範囲外
263	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# WakeupIPTask

## 割込モジュールの起動

### 機能

WakeupIPTaskwithParam()コマンドと同等の処理を行います。本コマンドでは、パラメータを設定することはできません。また、本コマンドを使用する場合は、必ずStartIPTask()コマンドをペアで使用して下さい。StartIPTaskwithParam()コマンドとペアで使用しないで下さい。

### コーディング

```
int
WakeupIPTask(
    DEVID      devID ,
    int        taskID
)
```

### コメント

デバイスID  
タスクID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

taskID

割込モジュールのタスクID。CreateIPTask()コマンドで生成したタスクIDを設定して下さい。設定範囲は1～64です。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了
-51～-9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	タスクID設定範囲外
263	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## TerminateIPTask

## 割込モジュールのサービス終了

### 機能

StartIPTaskwithParam()及びStartIPTask()コマンドでサービスが開始されているP I O割込モジュールのサービスを終了します。再びサービスを開始する場合は、StartIPTaskwithParam()及びStartIPTask()コマンドを実行して下さい。

### コーディング

```
int
TerminateIPTask(
    DEVID      devID ,
    int        taskID
)
```

### コメント

デバイスID  
タスクID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

taskID

割込モジュールのタスクID。CreateIPTask()コマンドで生成したタスクIDを設定して下さい。設定範囲は1～64です。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了
-51～-9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	タスクID設定範囲外
263	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ResumeIPTask

## 割込モジュールサービスの再開

### 機能

SuspendIPTask()コマンドで一時停止したP I O割込モジュールのサービスを再開します。

### コーディング

```
int
ResumeIPTask(
    DEVID      devID ,
    int        taskID
)
```

### コメント

デバイスID  
タスクID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

taskID

割込モジュールのタスクID。CreateIPTask()コマンドで生成したタスクIDを設定して下さい。設定範囲は0～64です。TASK\_SELF(0)を指定することにより、自タスクも指定できます。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了
-51～-9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	タスクID設定範囲外
263	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SuspendIPTask

## 割込モジュールサービスの一時停止

### 機能

任意の P I O 割込モジュールのサービスを一時停止します。ResumeIPTask() コマンドでサービスを再開できます。

### コーディング

```
int
SuspendIPTask(
    DEVID      devID,
    int        taskID
)
```

### コメント

デバイス I D  
タスク I D

### パラメータ

devID

デバイス I D。OpenIPDev コマンドで得られた対応するボード番号のデバイス I D を指定して下さい。

taskID

割込モジュールのタスク I D。CreateIPTask() コマンドで生成したタスク I D を設定して下さい。設定範囲は 0 ~ 6 4 です。TASK\_SELF(0) を指定することにより、自タスクも指定できます。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μ ITORN サービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	タスク I D 設定範囲外
2 6 3	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# DeleteIPTask

## 割込モジュールの削除

### 機能

RegistIPTask()コマンドで登録したP I O割込モジュールを削除します。

### コーディング

```
int
DeleteIPTask(
    DEVID      devID ,
    int        taskID
)
```

### コメント

デバイスID  
タスクID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

taskID

割込モジュールのタスクID。CreateIPTask()コマンドで生成したタスクIDを設定して下さい。設定範囲は1～64です。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了
-51～-9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	タスクID設定範囲外
263	未登録タスクが指定されている

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## SendSignal toPC

### PIO割込モジュールからのシグナル送信

#### 機能

オンボードCPU側のPIO割込モジュールからPC側に終了シグナルを送信する場合に使用します。PC側は、WaitforIPTaskSignal()コマンド等でオンボードCPU側からの終了シグナルをウェイトしていますが、本コマンドをオンボードCPU側から実行することにより、ウェイト状態が解除されます。本コマンドは、オンボードCPU側専用のコマンドですが、オンボードCPU側のプログラムをPC側でシミュレートできるようにPC側のコマンドとしてもダミーで実装しています。

#### コーディング

```
int
SendSignal toPC(
    int
    int
    int
)
```

```
signal ,
code ,
opt
```

#### コメント

PCへのシグナル番号  
終了コード  
オプション

#### パラメータ

##### signal

ローカルCPUからPCへのシグナルを送信する場合の番号。

ユーザシグナル番号	対応数	説明
PCI_SIGNAL_USER_0	16	ユーザシグナル# 0
PCI_SIGNAL_USER_1	17	ユーザシグナル# 1
PCI_SIGNAL_USER_2	18	ユーザシグナル# 2
PCI_SIGNAL_USER_7	23	ユーザシグナル# 7

##### code

終了コード

終了コード	対応数	説明
TASK_COMPLETE_REQUEST	0	PCに終了シグナルを送信します。
TASK_COMPLETE_PENDING	1	現在未使用。設定しないで下さい。

##### opt

現在、未使用。「0」を設定して下さい。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	ユーザシグナル番号設定範囲外
上記以外	μITRONシステム異常終了

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

# InitIPParamTable

## モジュールのパラメータテーブルの初期化

### 機能

インテリジェントモジュール実行コマンド (ExecuteInterIPModule()) 及び P I O 割込モジュール実行コマンド (StartIPTaskwithParam(), WakeupIPTaskwithParam(), WakeupIPTaskExt()) で使用するモジュールパラメータテーブルの初期化を行います。モジュールパラメータテーブルとは、オンボード C P U 側のプログラムに C 言語関数の仕様でわたされるパラメータを設定するためのテーブルです。

### コーディング

```
int
InitIPParamTable(
    DEVID
    MODULE_PARAM_TBL
    int
    enum IPModuleType
    int
    int
)
```

```
devID ,
*tbl ,
ParamCount ,
type ,
taskID ,
opt
```

### コメント

```
デバイス I D
モジュールパラメータテーブル
パラメータ数
モジュールタイプ
タスク I D ( オプション )
オプション
```

### パラメータ

#### devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

#### \*tbl

モジュールパラメータテーブル構造体。MODULE\_PARAM\_TBLテーブルの領域をあらかじめ確保して下さい。

#### ParamCount

パラメータ数。0 ~ 3 2 個のパラメータをオンボード C P Uに C 言語のパラメータとしてわたすことができます。

#### type

ターゲットのモジュールタイプを指定します。

モジュールタイプ	対応数	説明
INTELI_MODULE	0	インテリジェントモジュール
PIOINT_MODULE	1	P I O 割込起動モジュール

#### taskID

割込モジュールのタスク I D。CreateIPTask() コマンドで生成したタスク I Dを設定して下さい。設定範囲は 1 ~ 6 4 です。インテリジェントモジュールの場合は常に「0」を設定して下さい。

#### opt

オプション ( 現在未使用 )。「0」を設定して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	タスクID設定範囲外
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# Set IntegerParam

## 整数型パラメータのセット

### 機能

整数型のパラメータをモジュールパラメータテーブルにセットします。

### コーディング

```
int
SetIntegerParam(
    DEVID
    MODULE_PARAM_TBL
    enum IPModuleParamNo
    int
)
```

```
devID ,
*tbl ,
paramNo ,
val
```

### コメント

デバイス I D  
モジュールパラメータテーブル  
パラメータの順番  
整数設定値

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

\*tbl

モジュールパラメータテーブル構造体。MODULE\_PARAM\_TBLテーブルの領域をInitIPParamTable()コマンドで初期化して下さい。

paramNo

設定するパラメータの順番

パラメータの順番	対応数	説明
PARAM_1	0	第 1 パラメータ
PARAM_2	1	第 2 パラメータ
PARAM_3	2	第 3 パラメータ
PARAM_32	31	第 3 2 パラメータ

val

モジュールパラメータテーブルにセットする整数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	パラメータ順番範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetFloatParam

## 実数型パラメータのセット

### 機能

実数型のパラメータをモジュールパラメータテーブルにセットします。

### コーディング

```
int
SetFloatParam(
    DEVID
    MODULE_PARAM_TBL
    enum IPModuleParamNo
    float
)
```

```
devID ,
*tbl ,
paramNo ,
val
```

### コメント

デバイス I D  
モジュールパラメータテーブル  
パラメータの順番  
実数型の設定値

### パラメータ

**devID**

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**\*tbl**

モジュールパラメータテーブル構造体。MODULE\_PARAM\_TBLテーブルの領域をInitIPParamTable()コマンドで初期化して下さい。

**paramNo**

設定するパラメータの順番

パラメータの順番	対応数	説明
PARAM_1	0	第 1 パラメータ
PARAM_2	1	第 2 パラメータ
PARAM_3	2	第 3 パラメータ
PARAM_32	31	第 3 2 パラメータ

**val**

モジュールパラメータテーブルにセットする実数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	パラメータ順番範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetParamTable

## 配列データ(入力)パラメータのセット

### 機能

PC上にある配列データをオンボードCPUのプログラムにパラメータとして引き渡します。オンボードCPUのプログラムは、たんなる配列のポインタ渡しで受け取ることができます。

本コマンドにより、配列データ(入力)の領域をオンボードCPUシステムメモリに確保し、その配列データを転送します。そして、確保した領域のポインタをモジュールパラメータテーブルにセットします。

本コマンドでは、オンボードCPUシステムメモリにパラメータ領域としてあらかじめ確保されている領域から確保します。そのため、パラメータ領域の容量を超える領域は確保できません。パラメータ領域の容量は約64Kバイトです。また、SetParamTableExt()コマンドでは、その制限はありません。

### コーディング

```
int
SetParamTable(
    DEVID
    MODULE_PARAM_TBL
    enum IPModuleParamNo
    void
    int
)
```

```
devID ,
*tbl ,
paramNo ,
*pctbl ,
num
```

### コメント

```
デバイスID
モジュールパラメータテーブル
パラメータの順番
PC側データテーブル
データ数
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*tbl**

モジュールパラメータテーブル構造体。MODULE\_PARAM\_TBLテーブルの領域をInitIPParamTable()コマンドで初期化して下さい。

**paramNo**

設定するパラメータの順番

パラメータの順番	対応数	説明
PARAM_1	0	第1パラメータ
PARAM_2	1	第2パラメータ
PARAM_3	2	第3パラメータ
PARAM_32	31	第32パラメータ

**\*pctbl**

PC側のデータが格納されている配列のポインタ。4バイト単位に領域を確保して下さい。

**num**

セットする配列のバイト数。4バイト単位に設定して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	パラメータ順番範囲外
-	確保しようとしている容量がオンボードCPUのパラメータ領域を越えた
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# AllocParamTable

## 配列データ(出力)パラメータ領域の確保

### 機能

配列データ(入力)の領域をオンボードCPUシステムメモリに確保します。本コマンドでは、オンボードCPUシステムメモリにパラメータ領域としてあらかじめ確保されている領域から確保します。そのため、パラメータ領域の容量を超える領域は確保できません。パラメータ領域の容量は約64Kバイトです。また、AllocParamTableExt()コマンドでは、その制限はありません。

GetParamTable()コマンドでは、オンボードCPUのプログラムからPC上に確保した領域にデータを転送します。その場合、オンボードCPUのプログラムは、たんなる配列のポインタ渡しで出力することができます。AllocParamTable()コマンドはその配列データ(入力)の領域をオンボードCPUシステムメモリに確保するわけです。

この機能は、AllocParamTable()コマンドとGetParamTable()コマンドを組み合わせで使用します。通常、本コマンドにより、配列データ(入力)の領域をオンボードCPUシステムメモリに確保し、確保した領域のポインタをモジュールパラメータテーブルにセットします。そして、インテリジェントモジュール及びP I O割込モジュールを実行後、GetParamTable()コマンドでそのオンボードCPUシステムメモリの配列データをPC上のメモリに転送します。

### コーディング

```
int
AllocParamTable(
    DEVID
    MODULE_PARAM_TBL
    enum IPModuleParamNo
    int
)
```

```
devID ,
*tbl ,
paramNo ,
num
```

### コメント

```
デバイスID
モジュールパラメータテーブル
パラメータの順番
データ数
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*tbl**

モジュールパラメータテーブル構造体。MODULE\_PARAM\_TBLテーブルの領域をInitIPParamTable()コマンドで初期化して下さい。

**paramNo**

設定するパラメータの順番

パラメータの順番	対応数	説明
PARAM_1	0	第1パラメータ
PARAM_2	1	第2パラメータ
PARAM_3	2	第3パラメータ
PARAM_32	31	第32パラメータ

**num**

セットする配列のバイト数。4バイト単位に設定して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	パラメータ順番範囲外
-	確保しようとしている容量がオンボードCPUのパラメータ領域を越えた
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# GetParamTable

## 配列データ(出力)パラメータの読み出し

### 機能

オンボードCPUのプログラムからAllocParamTable()及びAllocParamTableExt()コマンドでPC上に確保した領域にデータを転送します。その場合、オンボードCPUのプログラムは、たんなる配列のポインタ渡しで出力することができます。

この機能は、AllocParamTable()コマンドとGetParamTable()コマンドを組み合わせで使用します。通常、AllocParamTable()及びAllocParamTableExt()コマンドにより、配列データ(入力)の領域をオンボードCPUシステムメモリに確保し、確保した領域のポインタをモジュールパラメータテーブルにセットします。そして、インテリジェントモジュール及びPIO割込モジュールを実行後、本コマンドでそのオンボードCPUシステムメモリの配列データをPC上のメモリに転送します。

### コーディング

```

int
GetParamTable(
    DEVID
    MODULE_PARAM_TBL
    enum IPModuleParamNo
    void
    int
)
devID ,
*tbl ,
paramNo ,
*pctbl ,
num
デバイスID
モジュールパラメータテーブル
パラメータの順番
PC側データテーブル
データ数

```

### コメント

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*tbl**  
モジュールパラメータテーブル構造体。MODULE\_PARAM\_TBLテーブルの領域をInitIPParamTable()コマンドで初期化して下さい。

**paramNo**  
設定するパラメータの順番

パラメータの順番	対応数	説明
PARAM_1	0	第1パラメータ
PARAM_2	1	第2パラメータ
PARAM_3	2	第3パラメータ
PARAM_32	31	第32パラメータ

**\*pctbl**  
オンボードCPUシステムメモリからデータを読み出すPC側のデータ領域のポインタ。4バイト単位に領域を確保して下さい。

**num**  
読み出しを行うバイト数。4バイト単位に設定して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	パラメータ順番範囲外
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# SetParamTableExt

## 配列データ(入力)パラメータのセット

### 機能

PC上にある配列データをオンボードCPUのプログラムにパラメータとして引き渡します。オンボードCPUのプログラムは、たんなる配列のポインタ渡しで受け取ることができます。

本コマンドにより、指定されたオンボードCPUシステムメモリアドレスからPC側の配列データを転送します。そして、指定した領域のアドレスをモジュールパラメータテーブルにセットします。

SetParamTable()コマンドでは、オンボードCPUシステムメモリにパラメータ領域としてあらかじめ確保されている領域から確保します。そのため、64Kバイトの容量を超える領域は確保できませんが

SetParamTableExt()コマンドでは、その制限はありません。任意の領域をユーザが確保し管理する必要があります。

### コーディング

```
int
SetParamTableExt(
    DEVID
    MODULE_PARAM_TBL
    enum IPModuleParamNo
    unsigned long
    void
    int
)
    devID ,          デバイスID
    *tbl ,           モジュールパラメータテーブル
    paramNo ,       パラメータの順番
    iptbl_addr ,   ボードCPUシステムメモリアドレス
    *pctbl ,       PC側データテーブル
    num             データ数
```

### コメント

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*tbl**  
モジュールパラメータテーブル構造体。MODULE\_PARAM\_TBLテーブルの領域をInitIPParamTable()コマンドで初期化して下さい。

**paramNo**  
設定するパラメータの順番

パラメータの順番	対応数	説明
PARAM_1	0	第1パラメータ
PARAM_2	1	第2パラメータ
PARAM_3	2	第3パラメータ
PARAM_32	31	第32パラメータ

**iptbl\_addr**  
書き込みを行うオンボードCPUシステムメモリのアドレス。4バイト単位のアドレスして下さい。

**\*pctbl**  
PC側のデータが格納されている配列のポインタ。4バイト単位に領域を確保して下さい。

**num**  
セットする配列のバイト数。4バイト単位に設定して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	パラメータ順番範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# AllocParamTableExt

## 配列データ(出力)パラメータ領域の確保

### 機能

配列データ(入力)の領域をオンボードCPUのプログラムにパラメータとして引き渡します。オンボードCPUのプログラムは、たんなる配列のポインタ渡しで受け取ることができます。

SetParamTableExtコマンドは、指定されたオンボードCPUシステムメモリアドレスからPC側の配列データを転送しますが、本コマンドは転送しないで指定された領域のアドレスをモジュールパラメータテーブルにセットします。

AllocParamTable()コマンドでは、オンボードCPUシステムメモリにパラメータ領域としてあらかじめ確保されている領域から確保します。そのため、64Kバイトの容量を超える領域は確保できませんがAllocParamTableExt()コマンドでは、その制限はありません。任意の領域をユーザが確保し管理する必要があります。

この機能は、AllocParamTableExt()コマンドとGetParamTable()コマンドを組み合わせで使用します。通常、本コマンドにより、配列データ(入力)の領域をオンボードCPUシステムメモリに確保し、確保した領域のポインタをモジュールパラメータテーブルにセットします。そして、インテリジェントモジュール及びP I O割込モジュールを実行後、GetParamTable()コマンドでそのオンボードCPUシステムメモリの配列データをPC上のメモリに転送します。

### コーディング

```
int
AllocParamTableExt(
    DEVID
    MODULE_PARAM_TBL
    enum IPModuleParamNo
    unsigned long
    int
)
```

```
devID ,
*tbl ,
paramNo ,
iptbl_addr ,
num
```

### コメント

```
デバイスID
モジュールパラメータテーブル
パラメータの順番
ボードCPUシステムメモリアドレス
データ数
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*tbl**

モジュールパラメータテーブル構造体。MODULE\_PARAM\_TBLテーブルの領域をInitIPParamTable()コマンドで初期化して下さい。

**paramNo**

設定するパラメータの順番

パラメータの順番	対応数	説明
PARAM_1	0	第1パラメータ
PARAM_2	1	第2パラメータ
PARAM_3	2	第3パラメータ
PARAM_32	31	第32パラメータ

**iptbl\_addr**

書き込みを行うオンボードCPUシステムメモリのアドレス。4バイト単位のアドレスして下さい。

**num**

セットする配列のバイト数。4バイト単位に設定して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	パラメータ順番範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

## ReadIPMemDatabyLong

## ボード側システムメモリのリード(4バイト)

### 機能

オンボードCPUシステムメモリの任意のアドレスからデータを4バイト単位に読み出し、PC側のデータ領域に書き込みます。

### コーディング

```
int
ReadIPMemDatabyLong(
    DEVID
    long
    unsigned long
    unsigned long
)
```

```
devID ,
*pctbl ,
ipaddr ,
num
```

### コメント

デバイスID  
PC側データ領域  
ボードCPUシステムメモリアドレス  
データ数(ロングワード数)

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*pctbl**

PC側データ領域。この領域にボードCPUシステムメモリの読み出されたデータが書き込まれます。領域は4バイト単位で確保して下さい。

**ipaddr**

読み出しを行うオンボードCPUシステムメモリのアドレス。4バイト単位のアドレスを指定して下さい。

**num**

読み出しを行うロングワード数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## ReadIPMemDatabyWord

## ボード側システムメモリのリード(2バイト)

### 機能

オンボードCPUシステムメモリの任意のアドレスからデータを2バイト単位に読み出し、PC側のデータ領域に書き込みます。

### コーディング

```
int
ReadIPMemDatabyWord(
    DEVID
    short
    unsigned long
    unsigned long
)
```

```
devID ,
*pctbl ,
ipaddr ,
num
```

### コメント

デバイスID  
PC側データ領域  
ボードCPUシステムメモリアドレス  
データ数(ワード数)

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*pctbl

PC側データ領域。この領域にボードCPUシステムメモリの読み出されたデータが書き込まれます。領域は2バイト単位で確保して下さい。

ipaddr

読み出しを行うオンボードCPUシステムメモリのアドレス。4バイト単位のアドレスを指定して下さい。

num

読み出しを行うワード数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## ReadIPMemDatabyByte

## ボード側システムメモリのリード(1バイト)

### 機能

オンボードCPUシステムメモリの任意のアドレスからデータを1バイト単位に読み出し、PC側のデータ領域に書き込みます。

### コーディング

```
int
ReadIPMemDatabyByte(
    DEVID
    char
    unsigned long
    unsigned long
)
```

```
devID ,
*pctbl ,
ipaddr ,
num
```

### コメント

デバイスID  
PC側データ領域  
ボードCPUシステムメモリアドレス  
データ数(バイト数)

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*pctbl

PC側データ領域。この領域にボードCPUシステムメモリの読み出されたデータが書き込まれます。領域は1バイト単位で確保して下さい。

ipaddr

読み出しを行うオンボードCPUシステムメモリのアドレス。4バイト単位のアドレスを指定して下さい。

num

読み出しを行うバイト数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# WriteIPMemDatabyLong

## ボード側システムメモリへのライト(4バイト)

### 機能

PC側のデータを、オンボードCPUシステムメモリの任意のアドレスから4バイト単位に書き込みます。

### コーディング

```
int
WriteIPMemDatabyLong(
    DEVID
    long
    unsigned long
    unsigned long
)
```

```
devID ,
*pctbl ,
ipaddr ,
num
```

### コメント

デバイスID  
PC側データ領域  
ボードCPUシステムメモリアドレス  
データ数(ロングワード数)

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*pctbl**

PC側データ領域。この領域のデータがボードCPUシステムメモリに書き込まれます。領域は4バイト単位で確保して下さい。

**ipaddr**

書き込みを行うオンボードCPUシステムメモリのアドレス。4バイト単位のアドレスを指定して下さい。

**num**

書き込みを行うロングワード数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## WriteIPMemDatabyWord

### ボード側システムメモリへのライト(2バイト)

#### 機能

PC側のデータを、オンボードCPUシステムメモリの任意のアドレスから2バイト単位に書き込みます。

#### コーディング

```
int
WriteIPMemDatabyWord(
    DEVID
    short
    unsigned long
    unsigned long
)
```

```
devID ,
*pctbl ,
ipaddr ,
num
```

#### コメント

デバイスID  
PC側データ領域  
ボードCPUシステムメモリアドレス  
データ数(ワード数)

#### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*pctbl**

PC側データ領域。この領域のデータがボードCPUシステムメモリに書き込まれます。領域は2バイト単位で確保して下さい。

**ipaddr**

書き込みを行うオンボードCPUシステムメモリのアドレス。4バイト単位のアドレスを指定して下さい。

**num**

書き込みを行うワード数

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

# WriteIPMemDatabyByte

## ボード側システムメモリへのライト(1バイト)

### 機能

PC側のデータを、オンボードCPUシステムメモリの任意のアドレスから1バイト単位に書き込みます。

### コーディング

```
int
WriteIPMemDatabyByte(
    DEVID
    char
    unsigned long
    unsigned long
)
```

```
devID ,
*pctbl ,
ipaddr ,
num
```

### コメント

デバイスID  
PC側データ領域  
ボードCPUシステムメモリアドレス  
データ数(バイト数)

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*pctbl

PC側データ領域。この領域のデータがボードCPUシステムメモリに書き込まれます。領域は1バイト単位で確保して下さい。

ipaddr

書き込みを行うオンボードCPUシステムメモリのアドレス。4バイト単位のアドレスを指定して下さい。

num

書き込みを行うバイト数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ExitIPTask

自タスクの終了

## 機能

自タスクを正常終了させます。

本コマンドは、ボード側のみ有効です。P C 側には存在しません。

## コーディング

```
void  
ExitIPTask(void)
```

## コメント

## パラメータ

なし

## リターン値

なし

## 詳細情報

## Get IPTaskID

### 実行状態の自タスクIDの参照

#### 機能

実行状態の自タスクIDを参照します。

本コマンドは、ボード側のみ有効です。PC側には存在しません。

#### コーディング

```

int
GetIPTaskID(
    int *tskid
)

```

#### コメント

タスクIDへのポインタ(出力)

#### パラメータ

\*tskid

参照したタスクIDが出力されます。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-51 ~ -9	μITORNサービスコールエラー

#### 詳細情報

# InitSCI

## SCIデバイスの初期化

### 機能

SCI (Serial Communication Interface) デバイスに対して、ビットレート、キャラクタ長、パリティ、ストップビットの設定を行います。

### コーディング

```
int
InitSCI(
    DEVID          devid ,
    enum SCIch     ch ,
    enum SCIBitRate rate ,
    enum SCIchrlng lng ,
    enum SCIParity par ,
    enum SCIStopBit stop
)
```

### コメント

デバイスID  
SCIチャンネル  
ビットレート  
キャラクタ長  
パリティ  
ストップビット

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ch

SCIのチャンネルを指定します。

SCIチャンネル	対応数	説明
SCI_CH0	0	SCIチャンネル# 0
SCI_CH1	1	SCIチャンネル# 1

#### rate

ビットレートを指定します。

ビットレート	対応数	説明
SCI_4800	0	4800bit/sec
SCI_9600	1	9600bit/sec(デフォルト設定)
SCI_19200	2	19200bit/sec
SCI_38400	3	38400bit/sec
SCI_57600	4	57600bit/sec

#### lng

キャラクタ長を指定します。

キャラクタ長	対応数	説明
SCI_8Bit	0	8bit(デフォルト設定)
SCI_7Bit	1	7bit

**par**

パリティを指定します。

パリティ	対応数	説明
SCI_NonParity	0	パリティなし(デフォルト設定)
SCI_NonParity1	1	パリティなし
SCI_EvenParity	2	偶数パリティ
SCI_OddParity	3	奇数パリティ

**【注】**

偶数パリティに設定すると送信時には、パリティビットと送信キャラクタをあわせて、その中の1の数の合計が偶数になるようにパリティビットを付加して送信します。受信時には、パリティビットと受信キャラクタを合わせて、その中の1の数の合計が偶数であるかどうかのチェックを行います。

奇数パリティに設定すると送信時には、パリティビットと送信キャラクタをあわせて、その中の1の数の合計が奇数になるようにパリティビットを付加して送信します。受信時には、パリティビットと受信キャラクタを合わせて、その中の1の数の合計が奇数であるかどうかのチェックを行います。

**stop**

ストップビット長を指定します。

ストップビット長	対応数	説明
SCI_StopBit1	0	1ストップビット(デフォルト設定)
SCI_StopBit2	1	2ストップビット

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0 ~ 2 4	パラメータ設定エラー
- 5 0 ~ - 9	μITRONサービスコールエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# SetSCIBuffer

## SCIバッファの設定

### 機能

SCIの入力バッファの設定を行います。デフォルトで128バイト分のバッファ領域が割り当てられています。それ以上のバッファ領域が必要な場合、本コマンドでオンボードCPUのシステムメモリ上のユーザー領域から割り当てて下さい。

### コーディング

```
int
SetSCIBuffer(
    DEVID
    enum SCIch
    unsigned long
    int
)
```

```
devID ,
ch ,
buff ,
n
```

### コメント

```
デバイスID
SCIチャンネル
バッファアドレス
バッファ容量(バイト)
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ch

SCIのチャンネルを設定します。

SCIチャンネル	対応数	説明
SCI_CH0_WRITE	0	SCIチャンネル#0 ライトバッファ
SCI_CH0_READ	1	SCIチャンネル#0 リードバッファ
SCI_CH1_WRITE	2	SCIチャンネル#1 ライトバッファ
SCI_CH1_READ	3	SCIチャンネル#1 リードバッファ

buff

バッファアドレス。オンボードCPUのシステムメモリ上のユーザー領域から割り当てて下さい。また、「0」を指定することでデフォルトのバッファ(16バイト)領域が設定されます。

n

バッファ容量。バイト数で指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20 ~ 22	パラメータ設定エラー
-51 ~ -9	μITRONサービスコールエラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# FlushSCIBuffer

## SCIバッファのフラッシュ

### 機能

SCIバッファのフラッシュを行います。本コマンドにより、バッファにプールされている入力データが全て捨てられ、バッファが初期化されます。

### コーディング

```
int
FlushSCIBuffer(
    DEVID
    enum SCIch
)
```

devID ,  
ch

### コメント

デバイスID  
SCIチャンネル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ch

SCIのチャンネルを設定します。

SCIチャンネル	対応数	説明
SCI_CH0_WRITE	0	SCIチャンネル#0 ライトバッファ
SCI_CH0_READ	1	SCIチャンネル#0 リードバッファ
SCI_CH1_WRITE	2	SCIチャンネル#1 ライトバッファ
SCI_CH1_READ	3	SCIチャンネル#1 リードバッファ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	パラメータ設定エラー
- 5 1 ~ - 9	μITRONサービスコールエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## ReadSCIBuffer

## SCIバッファからのデータリード

### 機能

SCIバッファから指定されたバイト数のデータを読み出します。バッファがエンプティ状態になると入力待ちになります。入力待ちにたくない場合は、IsSCIBuffer()コマンドでバッファに入力されているデータ数をチェックして下さい。

### コーディング

```
int
ReadSCIBuffer(
    DEVID
    enum SCIch
    char
    int
)
```

```
devID ,
ch ,
*buff ,
n
```

### コメント

```
デバイスID
SCIチャンネル
データバッファ
リードバイト数
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ch

SCIのチャンネルを設定します。

SCIチャンネル	対応数	説明
SCI_CH0	0	SCIチャンネル# 0
SCI_CH1	1	SCIチャンネル# 1

\*buff

データバッファ。ここで指定したアドレスの領域に n バイト数のデータをロードします。

n

リードデータ数。バイト数で指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20 ~ 22	パラメータ設定エラー
- 51 ~ - 9	μITRONサービスコールエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## ReadSCIBufferWithTimeout

## SCIバッファからのデータリード(タイムアウト付き)

### 機能

SCIバッファから指定されたバイト数のデータをタイムアウト付きで読み出します。バッファがエンブティ状態になると入力待ちになります。入力待ちにたくない場合は、IsSCIBuffer()コマンドでバッファに入力されているデータ数をチェックして下さい。

### コーディング

```
int
ReadSCIBufferWithTimeout(
    DEVID          devID ,
    enum SCICH     ch ,
    char           *buff ,
    int            n ,
    long           timeout
)
```

### コメント

デバイスID  
SCIチャンネル  
データバッファ  
リードバイト数  
タイムアウト時間

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ch**  
SCIのチャンネルを設定します。

SCIチャンネル	対応数	説明
SCI_CH0	0	SCIチャンネル# 0
SCI_CH1	1	SCIチャンネル# 1

**\*buff**  
データバッファ。ここで指定したアドレスの領域に n バイト数のデータをロードします。

**n**  
リードデータ数。バイト数で指定して下さい。

**timeout**  
タイムアウト時間 [ms]。  
「- 1」を指定するとタイムアウト処理は行わず、ReadSCIBufferコマンドと同じ処理を行います。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20 ~ 22	パラメータ設定エラー
- 51 ~ - 9	μITRONサービスコールエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# WriteSCIBuffer

## SCIへのデータ出力

### 機能

指定したバッファから指定されたバイト数のデータを S C I に出力します。

### コーディング

```
int
WriteSCIBuffer(
    DEVID
    enum SCIch
    char
    int
)
```

```
devID ,
ch ,
*buff ,
n
```

### コメント

デバイス I D  
S C I チャンネル  
データバッファ  
出力バイト数

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

ch

S C I のチャンネルを設定します。

S C I チャンネル	対応数	説明
SCI_CH0	0	S C I チャンネル # 0
SCI_CH1	1	S C I チャンネル # 1

\*buff

データバッファ。ここで指定したアドレスの領域から n バイト数のデータを出力します。

n

出力データ数。バイト数で指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0 ~ 2 2	パラメータ設定エラー
- 5 1 ~ - 9	μ ITRONサービスコールエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# IsSCIBuffer

## SCIバッファデータ数の取得

### 機能

SCIバッファにプールされているデータ数の取得を行います。

### コーディング

```
int
IsSCIBuffer(
    DEVID
    enum SCIch
)
```

devID ,  
ch

### コメント

デバイスID  
SCIチャンネル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ch

SCIのチャンネルを設定します。

SCIチャンネル	対応数	説明
SCI_CH0_WRITE	0	SCIチャンネル#0 ライトバッファ
SCI_CH0_READ	1	SCIチャンネル#0 リードバッファ
SCI_CH1_WRITE	2	SCIチャンネル#1 ライトバッファ
SCI_CH1_READ	3	SCIチャンネル#1 リードバッファ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	プールされているデータ数
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	パラメータ設定エラー
- 5 1 ~ - 9	μITRONサービスコールエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

SCI通信は、LAN通信と重なるとデータを取りこぼす可能性がありますので、注意して下さい。

# IsSCIError

## SCI通信エラーの取得

### 機能

SCI通信エラー（オーバランエラー，フレーミングエラー，パリティエラー）の取得を行います。  
IsSCIError()コマンドでエラーを読み出すことによりエラーがクリアされます。

### コーディング

```
int
IsSCIError(
    DEVID
    enum SCIch
)
```

```
devID ,
ch
```

### コメント

デバイスID  
SCIチャンネル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ch

SCIのチャンネルを設定します。

SCIチャンネル	対応数	説明
SCI_CH0	0	SCIチャンネル# 0
SCI_CH1	1	SCIチャンネル# 1

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	通信エラー
0	エラーなし
- 1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

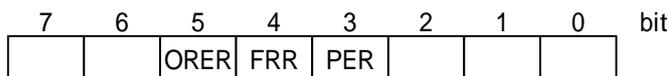
エラーコード	エラー原因
2 0	パラメータ設定エラー
- 5 1 ~ - 9	μITRONサービスコールエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

通信エラーの詳細

リターン値で得られるエラーステータスを以下に示します。



1 : 受信時のパリティエラー  
パリティチェックでのエラー

1 : 受信時のフレーミングエラー  
ストップビットのチェックでのエラー

1 : 受信時のオーバランエラー  
SCI受信データレジスタに未リードデータがあり、そこに次のデータが受信され、未リードデータが破壊された場合に発生します。

## IsSCIBufferFull

## SCIバッファフルエラーの取得

### 機能

SCIのバッファフルエラー（バッファオーバーフロー）の取得を行います。バッファオーバーフローが起きますとFlushSCIBuffer()コマンドでバッファをフラッシュするまで通信できません。

### コーディング

```
int
IsSCIBufferFull(
    DEVID          devID ,
    enum SCIch     ch
)
```

### コメント

デバイスID  
SCIチャンネル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ch

SCIのチャンネルを設定します。

SCIチャンネル	対応数	説明
SCI_CH0_WRITE	0	SCIチャンネル#0 ライトバッファ
SCI_CH0_READ	1	SCIチャンネル#0 リードバッファ
SCI_CH1_WRITE	2	SCIチャンネル#1 ライトバッファ
SCI_CH1_READ	3	SCIチャンネル#1 リードバッファ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	バッファオーバーフロー
0	エラーなし
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	パラメータ設定エラー
- 5 1 ~ - 9	μITRONサービスコールエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetCOMMode

## カメラリンク用シリアルポート切替え

### 機能

カメラリンク通信用のシリアルポート(SCI\_CH1)の接続を切替えます。初期状態で「PC用外部シリアルポートを選択する」に設定されています。

### コーディング

```
int
SetCOMMode(
    DEVID      devID ,
    int        mode
)
```

### コメント

デバイスID  
シリアルポート選択モード

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**mode**  
シリアルポート選択モード

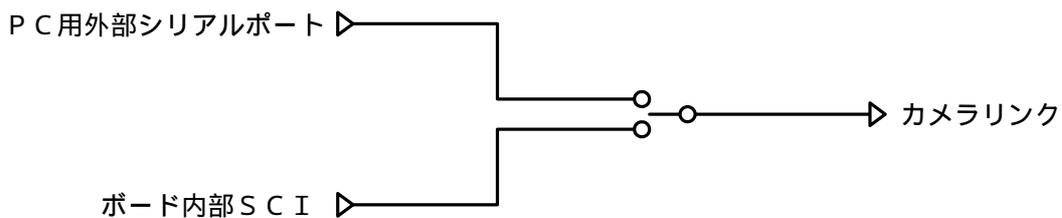
シリアルポート選択モード	説明
0	PC用外部シリアルポートを選択する(デフォルト)
1	ボード内部シリアルポート(SCI_CH1)を選択する

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了

### 詳細情報



**\*fmode**

パイプライン処理のステージごとの処理指定構造体を指定して下さい。

```
typedef struct {
    int    Stage0;    // ステージ 0
    int    Stage1;    // ステージ 1
    int    Stage2;    // ステージ 2
    int    Stage3;    // ステージ 3
    int    Stage4;    // ステージ 4
    int    Stage5;    // ステージ 5
    int    Stage6;    // ステージ 6
    int    Stage7;    // ステージ 7
} PipelineStageMode;
```

ステージ0～ステージ7に以下の処理を指定できます。

出力オプション	対応数	説明
STAGE_NOOP	0	処理なし
STAGE_NOOP1	1	処理なし (STAGE_NOOPと同じ)
DELETENOISE4	2	ノイズ除去 4 連結
DELETENOISE8	3	ノイズ除去 8 連結
DILATION4	4	4 連結膨張
DILATION8	5	8 連結膨張
EROSION4	6	4 連結収縮
EROSION8	7	8 連結収縮
OUTLINE4	8	4 連結輪郭抽出
OUTLINE8	9	8 連結輪郭抽出
DILATION4X	10	4 連結 X 方向膨張
DILATION8X	11	8 連結 X 方向膨張
EROSION4X	12	4 連結 X 方向収縮
EROSION8X	13	8 連結 X 方向収縮
DILATION4Y	14	4 連結 Y 方向膨張
DILATION8Y	15	8 連結 Y 方向膨張
EROSION4Y	16	4 連結 Y 方向収縮
EROSION8Y	17	8 連結 Y 方向収縮
DILATION4RU	18	4 連結右上がり方向膨張
DILATION8RU	19	8 連結右上がり方向膨張
EROSION4RU	20	4 連結右上がり方向収縮
EROSION8RU	21	8 連結右上がり方向収縮
DILATION4RD	22	4 連結右下がり方向膨張
DILATION8RD	23	8 連結右下がり方向膨張
EROSION4RD	24	4 連結右下がり方向収縮
EROSION8RD	25	8 連結右下がり方向収縮

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20, 21	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

フィルタ処理の注目画素の詳細

各ステージ毎に指定できる画像処理の注目画素を以下に示します。

	方向指定なし	X方向	Y方向	右上り方向	右下り方向
4 連結					
8 連結					

注目画素

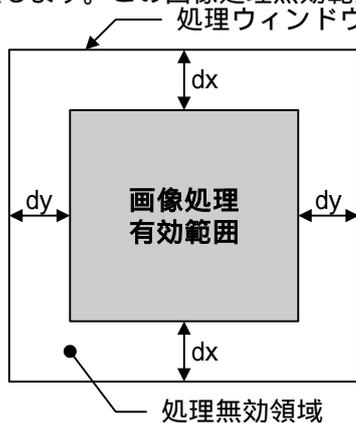
フィルタ処理の処理内容の詳細

各ステージ毎に指定できる画像処理の内容を以下に示します。

処理名	内容
ノイズ除去	注目画素が全て0の時、中心画素に0を出力
膨張	注目画素の少なくとも1つが1の時、中心画素に0を出力
収縮	注目画素の少なくとも1つが0の時、中心画素に0を出力
輪郭抽出	注目画素が全て1の時、中心画素に0を出力

画像処理対象エリアの境界部分の処理

3 × 3 カーネルを用いて画像処理を行うため、画像処理対象エリアの境界部分に画像処理結果無効の範囲が生じます。この画像処理無効範囲は、パイプラインの段数に依存します。



パイプライン段数	d x	d y
1 段	1	1
2 段	2	2
3 段	3	3
4 段	4	4
5 段	5	5
6 段	6	6
7 段	7	7
8 段	8	8

注) 4段論理演算モードは4段の値になります

## IP\_TrspipelineFLT

## 2値パイプラインフィルタの実行

### 機能

1画面の2値画像に対し、 $3 \times 3$ カーネルを用いてノイズ除去、膨張、収縮、輪郭の4つの画像処理を組み合わせ、最大8段までのフィルタ処理のパイプライン処理を行います。

本コマンドの前にSetTrspipelineFLTMode()コマンドで2値パイプラインフィルタのパイプライン処理出力モードとパイプライン処理ステージの設定を行なって下さい。

SVP-330では、ハードウェアの制約により「2値パイプラインフィルタコマンド」は、サポートされていません。

### コーディング

```
int
IP_TrspipelineFLT(
    DEVID
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst
```

### コメント

```
デバイスID
ソース画面
デスティネーション画面
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面番号を指定して下さい。

ImgDst

デスティネーション画面番号を指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPErr)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## EnablePipeline

## パイプラインモードの有効化

### 機能

パイプラインモードを有効化します。パイプライン処理についてはユーザズガイド第9章を参照して下さい。

### コーディング

```
int
EnablePipeline(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画面番号の付け方の例

```
IP_Add(devID, ImgSrc0, ImgSrc1, ImgID2);
IP_Binarize(devID, ImgID2, ImgDst, thr);
IP_ExtractB0Features(devID, ImgDst, &RegTbl, opt);
```

このとき、ImgID2の画面に加算した結果は入りません。また、パイプラインの処理終了時には、必ずパイプライン解除(DisablePipeline)を発行して下さい。

## DisablePipeline

## パイプラインモードの無効化

### 機能

パイプラインモードを無効化します。パイプライン処理についてはユーザーズガイド第9章を参照して下さい。

### コーディング

```
int  
DisablePipeline(  
    DEVID    devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# printf

## 書式付き出力

### 機能

標準出力ストリーム (RS232C) にフォーマット付きで出力します。C 言語の標準関数 printf() と同じです。

### コーディング

```
int  
printf(  
    const char    *format  
                [,argument]  
                .....  
)
```

### コメント

フォーマット  
オプションの引数

### パラメータ

\*format  
フォーマット

argument  
オプションの引数。フォーマットに合わせて指定して下さい。

### リターン値

表示した文字数を返します。エラーが発生すると負の値を返します。

### 詳細情報

これらの標準入出力関数を使用する場合は、インクルードファイル「stdio.h」をCのソースファイルにインクルードして下さい。

## scanf

### 書式付き入力

#### 機能

標準入力ストリーム (RS232C) からフォーマット付きで入力します。C 言語の標準関数 scanf() と同じです。

#### コーディング

```
int
scanf(
    const char    *format
                [, *argument]
                .....
)
```

#### コメント

フォーマット  
オプションの引数

#### パラメータ

\*format  
フォーマット

\*argument  
オプションの引数。フォーマットに合わせて指定して下さい。

#### リターン値

正しく読み出され、変換されたフィールド数を返します。

#### 詳細情報

これらの標準入出力関数を使用する場合は、インクルードファイル「stdio.h」をCのソースファイルにインクルードして下さい。

## *charget*

1文字入力

### 機能

標準入力ストリーム (RS232C) から1文字入力します。文字が入力されていない場合、入力されるまでウェイトします。本コマンドは、正確には文字ではなく0～255のデータをそのまま入力します。

### コーディング

```
char  
charget(void)
```

### コメント

引数なし

### パラメータ

なし

### リターン値

読み込んだデータを返します。

### 詳細情報

これらの標準入出力関数を使用する場合は、インクルードファイル「vpxcnv.h」をCのソースファイルにインクルードして下さい。

## charput

### 1文字出力

#### 機能

標準入力ストリーム (RS232C) に 1 文字出力します。本コマンドは、正確には文字ではなく 0 ~ 255 のデータをそのまま出力します。

#### コーディング

```
void  
charput(  
    char    character  
)
```

#### コメント

データ

#### パラメータ

character  
出力するデータ

#### リターン値

なし

#### 詳細情報

これらの標準入出力関数を使用する場合は、インクルードファイル「vpxcnv.h」をCのソースファイルにインクルードして下さい。

# ChangeIPTaskPriority

## タスクプライオリティの変更

### 機能

RegistIPTask()コマンドで登録した割込モジュールのタスクプライオリティを変更します。このコマンドは、StartIPTask(),StartIPTaskwithParam()コマンドでタスクをスタートさせてから発行して下さい。

また、変更したタスクのプライオリティはTerminateIPTask(),DeleteIPTask()コマンドを行っても初期状態に戻りません。初期状態に戻すには、再び本コマンドを発行してプライオリティを再設定して下さい。

### コーディング

```
int
ChangeIPTaskPriority(
    DEVID          devID ,
    int            taskID ,
    enum IPTaskPriority priority
)
```

### コメント

デバイスID  
タスクID  
プライオリティ

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### taskID

割込モジュールのタスクID。CreateIPTask()コマンドで生成したタスクIDを設定して下さい。設定範囲は0～64です。TASK\_SELF(0)を指定することにより、自タスクも指定できます。

#### priority

タスクプライオリティ

プライオリティ	対応数	説明
TASK_PRI_HIGHEST	-2	ノーマル - 2
TASK_PRI_ABOVE_NORMAL	-1	ノーマル - 1
TASK_PRI_NORMAL	0	ノーマル (初期値)
TASK_PRI_BELOW_NORMAL	1	ノーマル + 1
TASK_PRI_LOWEST	2	ノーマル + 2

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	タスクID設定範囲外

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# EnablePIOInterrupt

## PIO割込のマスク解除

### 機能

DisablePIOInterrupt() コマンドで無効にされた P I O の入力割込をハード的に有効にします。  
StartIPTask(), StartIPTaskwithParam() コマンドでタスクをスタートさせた時に自動的に解除されますので、  
通常はこのコマンドでハード的な割込マスクの解除は必要ありません。

### コーディング

```
int
EnablePIOInterrupt(
    DEVID      devID ,
    int        intbit
)
```

### コメント

デバイス I D  
割込のビット番号

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

intbit

割込のビット番号。SVP-330はDI0~DI3の4ビットが有効です。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	タスク I D設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# DisablePIOInterrupt

## PIO割込のマスク

### 機能

PIOの入力割込をハード的に無効にします。このコマンドを実行すると以降、割込による起動ができなくなりますので注意して下さい。また、StartIPTask(), StartIPTaskwithParam()コマンドでタスクをスタートさせた時に自動的に解除されますので、タスクスタート後に発行して下さい。

### コーディング

```
int
DisablePIOInterrupt(
    DEVID      devid ,
    int        intbit
)
```

### コメント

デバイスID  
割込のビット番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

intbit

割込のビット番号。SVP-330はDI0~DI3の4ビットが有効です。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	タスクID設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SleepIPTask

## モジュールのスリープ

### 機能

P I O 割込モジュール ( 自タスク ) をスリープ状態にします。このスリープ状態は、他のタスクや割込ハンドラからウェイクアップされるまで継続されます。このコマンドは、ボード側のプログラムでのみ有効です。

このコマンドを実行するとタスクチェンジが発生しますので、それによる不具合が発生する可能性がありますので注意して下さい。

本コマンドは、ボード側のみの有効です。P C 側ではダミーコマンドとして存在します。

### コーディング

```
int  
SleepIPTask(void)
```

### コメント

スリープ

### パラメータ

なし

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
0	異常終了
-51 ~ -9	μITORNサービスコールエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## WaitIPTask

## モジュールの時間指定スリープ

### 機能

PIO割込モジュール（自タスク）を指定時間スリープ状態にします。このスリープ状態は、他のタスクや割込ハンドラからウェイクアップされるかまたは、指定した時間になるまで継続されます。このコマンドは、ボード側のプログラムでのみ有効です。

このコマンドを実行するとタスクチェンジが発生しますので、それによる不具合が発生する可能性がありますので注意して下さい。

本コマンドは、ボード側のみ有効です。PC側ではダミーコマンドとして存在します。

### コーディング

```
int
WaitIPTask(
    long    time
)
```

### コメント

ウェイト時間

### パラメータ

**time**  
スリープ時間。設定単位は ( m S ) で、分解能は 1 m S です。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	他タスクからのウェイクアップ要求により起床
WAIT_TASK_TIME_OUT ( -50 )	タイムアウト
-51 ~ -9	μITORNサービスコールエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# Get IPTaskStatus

## タスクステータスの取得

### 機能

RegistIPTask()コマンドで登録したP I O割込モジュールの現在の状態を取得します。

本コマンドは、ボード側のみ有効です。P C側ではダミーコマンドとして存在します。

### コーディング

```
int
GetIPTaskStatus(
    int
    RTSK_INFO
)
```

```
taskID ,
*info
```

### コメント

タスク I D  
タスクステータス情報

### パラメータ

#### taskID

割込モジュールのタスク I D。CreateIPTask() コマンドで生成したタスク I Dを設定して下さい。設定範囲は 0 ~ 6 4 です。TASK\_SELF(0)を指定することにより、自タスクも指定できます。

#### \*info

タスクステータス情報テーブル。taskIDで指定したタスクのステータス情報を出力します。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了。タスク I D設定範囲外
-51 ~ -9	μ ITORNサービスコールエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

vpxsys.hで宣言しています

```
typedef struct {
    unsigned long    tskstat;
    short           tskpri;
    short           tskbpri;
    unsigned long    tstwait;
    short           wobjid;
    long            lefttmo;
    unsigned long    actcnt;
    unsigned long    wupcnt;
    unsigned long    suscnt;
    unsigned long    tskmode;
    unsigned long    tflptn;
} RTSK_INFO;
```

タスクステータス情報	説明																																							
tskstat	現在のタスクの状態。 <table border="1"> <thead> <tr> <th>タスク状態</th> <th>対応数</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>TASK_STS_RUN</td> <td>0x00000001</td> <td>実行状態</td> </tr> <tr> <td>TASK_STS_RDY</td> <td>0x00000002</td> <td>実行可能状態</td> </tr> <tr> <td>TASK_STS_WAI</td> <td>0x00000004</td> <td>待ち状態</td> </tr> <tr> <td>TASK_STS_SUS</td> <td>0x00000008</td> <td>強制待ち状態</td> </tr> <tr> <td>TASK_STS_WAS</td> <td>0x0000000C</td> <td>2重待ち状態</td> </tr> <tr> <td>TASK_STS_DMT</td> <td>0x00000010</td> <td>休止状態</td> </tr> </tbody> </table>	タスク状態	対応数	説明	TASK_STS_RUN	0x00000001	実行状態	TASK_STS_RDY	0x00000002	実行可能状態	TASK_STS_WAI	0x00000004	待ち状態	TASK_STS_SUS	0x00000008	強制待ち状態	TASK_STS_WAS	0x0000000C	2重待ち状態	TASK_STS_DMT	0x00000010	休止状態																		
タスク状態	対応数	説明																																						
TASK_STS_RUN	0x00000001	実行状態																																						
TASK_STS_RDY	0x00000002	実行可能状態																																						
TASK_STS_WAI	0x00000004	待ち状態																																						
TASK_STS_SUS	0x00000008	強制待ち状態																																						
TASK_STS_WAS	0x0000000C	2重待ち状態																																						
TASK_STS_DMT	0x00000010	休止状態																																						
tskpri	タスクの現在優先度。 <table border="1"> <thead> <tr> <th>プライオリティ</th> <th>対応数</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>TASK_PRI_HIGHEST</td> <td>-2</td> <td>ノーマル - 2</td> </tr> <tr> <td>TASK_PRI_ABOVE_NORMAL</td> <td>-1</td> <td>ノーマル - 1</td> </tr> <tr> <td>TASK_PRI_NORMAL</td> <td>0</td> <td>ノーマル (初期値)</td> </tr> <tr> <td>TASK_PRI_BELOW_NORMAL</td> <td>1</td> <td>ノーマル + 1</td> </tr> <tr> <td>TASK_PRI_LOWEST</td> <td>2</td> <td>ノーマル + 2</td> </tr> </tbody> </table>	プライオリティ	対応数	説明	TASK_PRI_HIGHEST	-2	ノーマル - 2	TASK_PRI_ABOVE_NORMAL	-1	ノーマル - 1	TASK_PRI_NORMAL	0	ノーマル (初期値)	TASK_PRI_BELOW_NORMAL	1	ノーマル + 1	TASK_PRI_LOWEST	2	ノーマル + 2																					
プライオリティ	対応数	説明																																						
TASK_PRI_HIGHEST	-2	ノーマル - 2																																						
TASK_PRI_ABOVE_NORMAL	-1	ノーマル - 1																																						
TASK_PRI_NORMAL	0	ノーマル (初期値)																																						
TASK_PRI_BELOW_NORMAL	1	ノーマル + 1																																						
TASK_PRI_LOWEST	2	ノーマル + 2																																						
tskbpri	タスクのベース優先度。																																							
tstwait	tskstat が TASK_STS_WAI , TASK_STS_WAS のときに有効で、次の値を返します。 <table border="1"> <thead> <tr> <th>タスク状態</th> <th>対応数</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>TASK_WAIT_SLP</td> <td>0x00000001</td> <td>スリープタスクによる待ち</td> </tr> <tr> <td>TASK_WAIT_DLY</td> <td>0x00000002</td> <td>ディレイタスクによる待ち</td> </tr> <tr> <td>TASK_WAIT_SEM</td> <td>0x00000004</td> <td>ウェイトセマフォによる待ち</td> </tr> <tr> <td>TASK_WAIT_FLG</td> <td>0x00000008</td> <td>ウェイトフラグによる待ち</td> </tr> <tr> <td>TASK_WAIT_SDTQ</td> <td>0x00000010</td> <td>SNDDTQによる待ち</td> </tr> <tr> <td>TASK_WAIT_RDTQ</td> <td>0x00000020</td> <td>RCVDTQによる待ち</td> </tr> <tr> <td>TASK_WAIT_MBX</td> <td>0x00000040</td> <td>レシーブメッセージによる待ち</td> </tr> <tr> <td>TASK_WAIT_MTX</td> <td>0x00000080</td> <td>ミューテックスによる待ち</td> </tr> <tr> <td>TASK_WAIT_SMBF</td> <td>0x00000100</td> <td>SNDMBFによる待ち</td> </tr> <tr> <td>TASK_WAIT_RMBF</td> <td>0x00000200</td> <td>RCVMBFによる待ち</td> </tr> <tr> <td>TASK_WAIT_MPF</td> <td>0x00002000</td> <td>ゲットブロックによる待ち</td> </tr> <tr> <td>TASK_WAIT_MPL</td> <td>0x00004000</td> <td>ゲットブロックによる待ち</td> </tr> </tbody> </table>	タスク状態	対応数	説明	TASK_WAIT_SLP	0x00000001	スリープタスクによる待ち	TASK_WAIT_DLY	0x00000002	ディレイタスクによる待ち	TASK_WAIT_SEM	0x00000004	ウェイトセマフォによる待ち	TASK_WAIT_FLG	0x00000008	ウェイトフラグによる待ち	TASK_WAIT_SDTQ	0x00000010	SNDDTQによる待ち	TASK_WAIT_RDTQ	0x00000020	RCVDTQによる待ち	TASK_WAIT_MBX	0x00000040	レシーブメッセージによる待ち	TASK_WAIT_MTX	0x00000080	ミューテックスによる待ち	TASK_WAIT_SMBF	0x00000100	SNDMBFによる待ち	TASK_WAIT_RMBF	0x00000200	RCVMBFによる待ち	TASK_WAIT_MPF	0x00002000	ゲットブロックによる待ち	TASK_WAIT_MPL	0x00004000	ゲットブロックによる待ち
タスク状態	対応数	説明																																						
TASK_WAIT_SLP	0x00000001	スリープタスクによる待ち																																						
TASK_WAIT_DLY	0x00000002	ディレイタスクによる待ち																																						
TASK_WAIT_SEM	0x00000004	ウェイトセマフォによる待ち																																						
TASK_WAIT_FLG	0x00000008	ウェイトフラグによる待ち																																						
TASK_WAIT_SDTQ	0x00000010	SNDDTQによる待ち																																						
TASK_WAIT_RDTQ	0x00000020	RCVDTQによる待ち																																						
TASK_WAIT_MBX	0x00000040	レシーブメッセージによる待ち																																						
TASK_WAIT_MTX	0x00000080	ミューテックスによる待ち																																						
TASK_WAIT_SMBF	0x00000100	SNDMBFによる待ち																																						
TASK_WAIT_RMBF	0x00000200	RCVMBFによる待ち																																						
TASK_WAIT_MPF	0x00002000	ゲットブロックによる待ち																																						
TASK_WAIT_MPL	0x00004000	ゲットブロックによる待ち																																						
wobjid	tskstat が TASK_STS_WAI , TASK_STS_WAS のときに有効で、待ち対象のオブジェクト ID を返します。																																							
lefttmo	対象タスクがタイムアウトするまでの時間を返します。																																							
actcnt	現在の起動要求キューイング数を返します。																																							
wupcnt	現在の起床要求キューイング数を返します。																																							
suscnt	現在の強制待ち要求ネスト数を返します。																																							
tskmode	未使用																																							
tflptn	未使用																																							

## CancelWakeup

## ウェイクアップ要求無効化

### 機能

キューイングされているウェイクアップ要求を無効にします。

本コマンドは、ボード側のみ有効です。PC側ではダミーコマンドとして存在します。

### コーディング

```
int  
CancelWakeup(  
    int taskID  
)
```

### コメント

タスクID

### パラメータ

#### taskID

割込モジュールのタスクID。CreateIPTask()コマンドで生成したタスクIDを設定して下さい。設定範囲は0～64です。TASK\_SELF(0)を指定することにより、自タスクも指定できます。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了。タスクID設定範囲外
-51～-9	μITORNサービスコールエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## WaitSemaphore

## セマフォウェイト

## 機能

semidで示されたセマフォのカウンタ値が1以上の場合、セマフォのカウンタ値を1減らして発行タスクは実行を継続します。セマフォのカウンタ値が0の場合は、セマフォのカウンタ値を変更せず、発行タスクはそのセマフォに対する待ち行列につなわれ、ウェイト状態へ移行します。

ウェイト状態に移行したタスクは、同じsemidでSignalSemaphore()コマンドが別のタスクから発行されるとレディ(実行可能)状態に移行します。

なお、セマフォのカウンタ値の初期値は、1です。

本コマンドは、ボード側のみ有効です。PC側ではダミーコマンドとして存在します。

## コーディング

```
int
WaitSemaphore(
    int    semid
)
```

## コメント

セマフォID

## パラメータ

semid

セマフォID。CreateIPSempre()コマンドで得られたIDを指定して下さい。

## リターン値

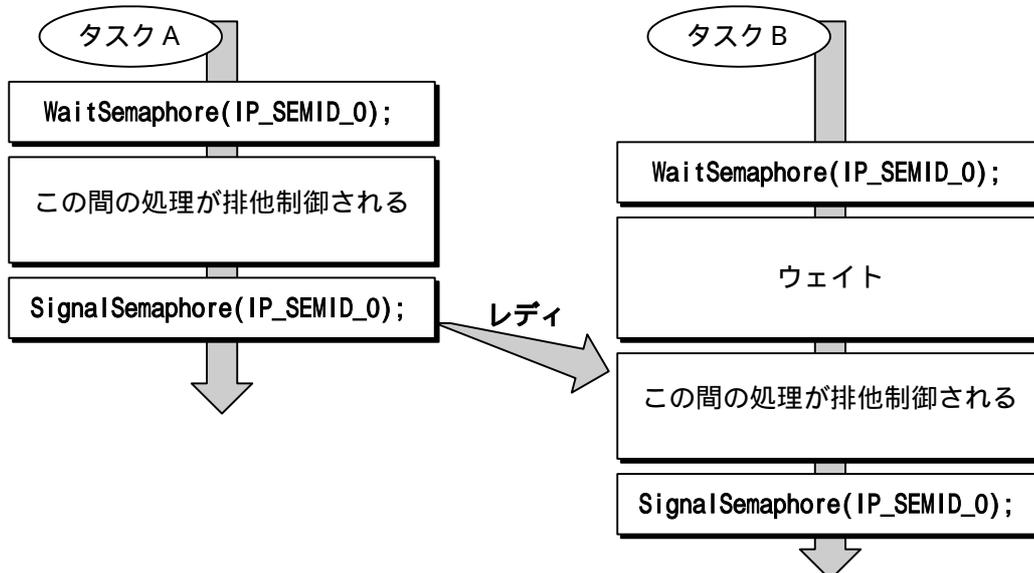
リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了。セマフォID設定範囲外
-51 ~ -9	μITORNサービスコールエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

セマフォは、システムリソースの排他制御を行う場合に使用します。以下にその例を示します。



# SignalSemaphore

## セマフォシグナル

### 機能

semidで示されたセマフォに対してウェイトしているタスクがあれば、セマフォの待ち行列の先頭のタスクをレディ（実行可能）状態にします。セマフォに対してウェイトしているタスクがなければ、そのセマフォのカウンタ値を1増やします。

本コマンドは、ボード側のみ有効です。PC側ではダミーコマンドとして存在します。

### コーディング

```
int
SignalSemaphore(
    int    semid
)
```

### コメント

デバイスID  
セマフォID

### パラメータ

**semid**  
セマフォID。CreateIPSemapre()コマンドで得られたIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了。セマフォID設定範囲外
-51 ~ -9	μITORNサービスコールエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

詳細については、WaitSemaphore()コマンドを参照して下さい。

## SemStatus

## セマフォステータスの取得

### 機能

semidで指定されたセマフォの状態を参照し、対象セマフォの現在のカウンタ値と待ち行列の先頭のタスクIDを求めます。

本コマンドは、ボード側のみ有効です。PC側ではダミーコマンドとして存在します。

### コーディング

```
int
SemStatus(
    int          semid ,
    long         *semcnt ,
    short        *wtskid
)
```

### コメント

セマフォID  
セマフォカウンタ値  
待ちタスクID

### パラメータ

#### semid

セマフォID。CreateIPSemapre()コマンドで得られたIDを指定して下さい。

#### \*semcnt

セマフォのカウンタ値。この値が0の時、semidのセマフォに対してウェイトしているタスクが存在しています。また、ウェイト状態になっている先頭のタスクがwtskidで返されます。

#### \*wtskid

semcntが0の場合、semidのセマフォに対してウェイトしている割込モジュールのタスクIDを返します。対象セマフォのウェイトタスクがない場合は、「0」を返します。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了。セマフォID設定範囲外
-51 ~ -9	μITORNサービスコールエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

セマフォの動作については、WaitSemaphore()コマンドを参照して下さい。

# SignalIPSemaphore

## PCからのセマフォシグナル

### 機能

semidで示されたセマフォに対してウェイトしているタスクがあれば、セマフォの待ち行列の先頭のタスクをレディ（実行可能）状態にします。セマフォに対してウェイトしているタスクがなければ、そのセマフォのカウンタ値を1増やします。

本コマンドは、SignalIPSemaphore()コマンドと同じですが、PC（パソコン）側からボード側でウェイトしているセマフォに対してシグナルを送信することができます。

### コーディング

```
int
SignalIPSemaphore(
    DEVID    devid ,
    int      semid
)
```

### コメント

デバイスID  
セマフォID

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**semid**  
セマフォID。CreateIPSemapre()コマンドで得られたIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了。セマフォID設定範囲外
-51 ~ -9	μITORNサービスコールエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

セマフォの動作については、WaitSemaphore()コマンドを参照して下さい。

## SuspendSystemTimer

## システムタイマの一時停止

### 機能

画像処理コマンドで使用しているタイマをハード的に一時停止させます。復帰するには、ResumeSystemTimer()コマンドを使用して下さい。

### コーディング

```
int
SuspendSystemTimer(
    DEVID
    int
)
```

```
devID ,
timerid
```

### コメント

```
デバイスID
タイマID
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

timerid

タイマID

タイマID	対応数	説明
TIMERID_0	0	μITRONで使用しているインターバルタイマ
TIMERID_1	1	スリープタイマ
TIMERID_2	2	時間計測タイマ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドでは、timeridに「TIMERID\_0」を設定することにより、μITRONのインターバルタイマ割込による処理時間の変動を抑えることができます。しかし、μITRONのインターバルタイマ割込を強制的に停止することにより、PCコマンド実行時のタイムアウト検出や、WaitIPTask()コマンドが機能しなくなりますので注意して下さい。

# ResumeSystemTimer

## システムタイマの一時停止解除

### 機能

SuspendSystemTimer() コマンドで一時停止させたタイマを元の状態に復帰させます。

### コーディング

```
int
ResumeSystemTimer(
    DEVID
    int
)
```

devID ,  
timerid

### コメント

デバイスID  
タイマID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

timerid

タイマID

タイマID	対応数	説明
TIMERID_0	0	μITRONで使用しているインターバルタイマ
TIMERID_1	1	スリープタイマ
TIMERID_2	2	時間計測タイマ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# CreateIPSemaphore

## セマフォの生成

### 機能

セマフォの生成を行います。セマフォの生成に成功するとセマフォIDをリターン値として返します。

### コーディング

```
int
CreateIPSemaphre(
    DEVID      devID ,
    CSEM_INFO  *info
)
```

### コメント

デバイスID  
セマフォ生成情報

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*info**  
セマフォ生成情報

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	セマフォID
= 0	セマフォ生成失敗
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
260	タスク管理機能システムでエラーが発生しました。

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

CSEM\_INFO のフォーマット及び内容を、下記に示します。  
vpxsys.hで宣言しています

```
typedef struct {
    unsigned long    attr;
    unsigned long    count;
    unsigned long    maxsem;
} CSEM_INFO;
```

セマフォ生成情報	説明									
attr	セマフォ属性 <table border="1"> <thead> <tr> <th>デバイス番号</th> <th>対応数</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>TASK_ATTR_TFIFO</td> <td>0x0000</td> <td>待ちタスクのキューイングはFIFO</td> </tr> <tr> <td>TASK_ATTR_TPRI</td> <td>0x0001</td> <td>待ちタスクのキューイングは優先度順</td> </tr> </tbody> </table>	デバイス番号	対応数	説明	TASK_ATTR_TFIFO	0x0000	待ちタスクのキューイングはFIFO	TASK_ATTR_TPRI	0x0001	待ちタスクのキューイングは優先度順
デバイス番号	対応数	説明								
TASK_ATTR_TFIFO	0x0000	待ちタスクのキューイングはFIFO								
TASK_ATTR_TPRI	0x0001	待ちタスクのキューイングは優先度順								
count	セマフォ資源数の初期値									
maxsem	セマフォの最大資源数									

# DeleteIPSemaphore

## セマフォの削除

### 機能

セマフォの削除を行います。

### コーディング

```

int
DeleteIPSemaphre(
    DEVID          devID ,
    int            semid
)

```

### コメント

デバイスID  
セマフォID

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**semid**  
セマフォID

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μITRONサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ReadSemaphore

## セマフォの状態参照

### 機能

セマフォの状態参照を行います。

### コーディング

```
int
ReadSemaphore(
    DEVID          devID ,
    int            semid ,
    RSEM_INFO     *info
)
```

### コメント

デバイス I D  
セマフォ I D  
セマフォの状態情報

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

semid

セマフォ I D

\*info

セマフォの状態情報

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

RSEM\_INFO のフォーマット及び内容を、下記に示します。

vpxsys.hで宣言しています

```
typedef struct {
    int            wtskid;
    unsigned long semcnt;
} RSEM_INFO;
```

セマフォ生成情報	説明
wtskid	待ちタスク I D
semcnt	現在のセマフォカウント値

## WaitSemaphoreWithTimeOut

## タイムアウト付きセマフォウェイト

### 機能

タイムアウト機能付きのセマフォウェイトです。待ち解除の条件が満たされないままtmout時間が経過するとエラーコードを返し、処理が終了します。セマフォウェイトの詳細は、WaitSemaphore()コマンドを参照して下さい。

本コマンドは、ボード側のみ有効です。P C側ではダミーコマンドとして存在します。

### コーディング

```
int
WaitSemaphoreWithTimeOut(
    DEVID      devid ,
    int        semid ,
    int        tmout
)
```

### コメント

デバイスID  
セマフォID  
タイムアウト時間

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**semid**  
セマフォID

**tmout**  
タイムアウト時間 (ms)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了。タイムアウト。
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# CreateEventFlag

## イベントフラグの生成

### 機能

イベントフラグの生成を行います。イベントフラグの生成に成功するとイベントフラグIDをリターン値として返します。

### コーディング

```
int
CreateEventFlag(
    DEVID          devID ,
    CFLG_INFO     *info
)
```

### コメント

デバイスID  
イベントフラグ生成情報

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*info**  
イベントフラグ生成情報

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	イベントフラグID
= 0	イベントフラグ生成失敗
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
260	タスク管理機能システムでエラーが発生しました。

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

CFLG\_INFO のフォーマット及び内容を、下記に示します。  
vpxsys.hで宣言しています

```
typedef struct {
    unsigned long   attr;
    unsigned long   flgptn;
} CFLG_INFO;
```

イベントフラグ生成情報	説明																		
attr	イベントフラグ属性																		
	<table border="1"> <thead> <tr> <th>デバイス番号</th> <th>対応数</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>TASK_ATTR_TFIFO</td> <td>0x0000</td> <td>待ちタスクのキューイングはFIFO</td> </tr> <tr> <td>TASK_ATTR_TPRI</td> <td>0x0001</td> <td>待ちタスクのキューイングは優先度順</td> </tr> <tr> <td>TASK_ATTR_WSGL</td> <td>0x0000</td> <td>複数タスクの待ちを許さない</td> </tr> <tr> <td>TASK_ATTR_WMUL</td> <td>0x0002</td> <td>複数タスクの待ちを許す</td> </tr> <tr> <td>TASK_ATTR_CLR</td> <td>0x0004</td> <td>クリア指定</td> </tr> </tbody> </table>	デバイス番号	対応数	説明	TASK_ATTR_TFIFO	0x0000	待ちタスクのキューイングはFIFO	TASK_ATTR_TPRI	0x0001	待ちタスクのキューイングは優先度順	TASK_ATTR_WSGL	0x0000	複数タスクの待ちを許さない	TASK_ATTR_WMUL	0x0002	複数タスクの待ちを許す	TASK_ATTR_CLR	0x0004	クリア指定
	デバイス番号	対応数	説明																
	TASK_ATTR_TFIFO	0x0000	待ちタスクのキューイングはFIFO																
	TASK_ATTR_TPRI	0x0001	待ちタスクのキューイングは優先度順																
	TASK_ATTR_WSGL	0x0000	複数タスクの待ちを許さない																
TASK_ATTR_WMUL	0x0002	複数タスクの待ちを許す																	
TASK_ATTR_CLR	0x0004	クリア指定																	
flgptn	イベントフラグの初期値																		

## DeleteEventFlag

## イベントフラグの削除

### 機能

イベントフラグの削除を行います。

### コーディング

```
int
DeleteEventFlag(
    DEVID          devid ,
    int            flgid
)
```

### コメント

デバイスID  
イベントフラグID

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**flgid**  
イベントフラグID

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetEventFlag

## イベントフラグのセット

### 機能

flgidで示されたイベントフラグを、setptnで示された値との論理和（OR）に更新します。

イベントフラグ値の更新の結果、そのイベントフラグで待っているタスクの待ち解除条件を満たせば、そのタスクの待ち状態を解除します。なお、待ち解除条件は待ち行列の順に評価されます。この時、対象のイベントフラグ属性に TASK\_ATTR\_CLR 属性が指定されている場合には、イベントフラグのビットパターンのすべてのビットをクリアし、サービスコールの処理を終了します。イベントフラグにTASK\_ATTR\_WMUL属性が指定されており、TASK\_ATTR\_CLR属性が指定されていない場合、SetEventFlagの1回の呼び出しで複数のタスクが待ち解除される可能性があります。待ち解除されるタスクが複数ある場合には、イベントフラグの待ち行列につながれていた順序で待ち解除されます。

### コーディング

```
int
SetEventFlag(
    DEVID          devID ,
    int            flgid ,
    int            setptn
)
```

### コメント

デバイスID  
イベントフラグID  
セットするビットパターン

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**flgid**  
イベントフラグID

**setptn**  
セットするビットパターン

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ClearEventFlag

## イベントフラグのクリア

### 機能

flgidで示されたイベントフラグをclrptnで示された値との論理積 (AND) に更新します。

### コーディング

```
int  
ClearEventFlag(  
    DEVID      devID ,  
    int        flgid ,  
    int        clrptn  
)
```

### コメント

デバイスID  
イベントフラグID  
クリアするビットパターン

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**flgid**  
イベントフラグID

**clrptn**  
クリアするビットパターン

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# WaitEventFlag

## イベントフラグウェイト

### 機能

flgidで指定されるイベントフラグのビットパターンが、waiptnとmodeで指定される待ち解除条件を満たすのを待ちます。\*flgptnの領域には、待ち解除される時のイベントフラグのビットパターンを返します。

対象イベントフラグがTASK\_ATTR\_WSGL属性で、すでに他のタスクが待っている場合は、本コマンドはエラーになります。

本コマンドを発行した時すでに待ち解除条件が成立している場合は、本コマンドはただちに終了します。待ち解除条件が成立していない場合は、イベント待ち行列につながれます。

### コーディング

```
int
WaitEventFlag(
    DEVID          devID ,
    int            flgid ,
    unsigned long  waiptn ,
    int            mode ,
    unsigned long  *flgptn
)
```

### コメント

デバイスID  
イベントフラグID  
ウェイトするビットパターン  
ウェイトモード  
ウェイト解除時のビットパターン

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**flgid**  
イベントフラグID

**waiptn**  
ウェイトするビットパターン

**mode**  
ウェイトモード

デバイス番号	対応数	説明
WFLG_AND_WAIT	0x0000	AND待ち waiptnで指定したビットの全てがセットされるのを待ちます。
WFLG_OR_WAIT	0x0001	OR待ち flgidで示されたイベントフラグのうちwaiptnで指定したビットのいずれかがセットされるのを待ちます。

**\*flgptn**  
ウェイト解除時のビットパターンを返す領域のポインタ

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μITORNサーブিসコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# WaitEventFlagWithTimeOut

## タイムアウト付きイベントフラグウェイト

### 機能

タイムアウト機能付きのイベントフラグウェイトです。待ち解除の条件が満たされないまま tmout 時間が経過するとエラーコードを返し、処理が終了します。イベントフラグウェイトの詳細は、WaitEventFlag() コマンドを参照して下さい。

本コマンドは、ボード側のみ有効です。P C 側ではダミーコマンドとして存在します。

### コーディング

```
int
WaitEventFlagWithTimeOut(
    DEVID          devid ,
    int            flgid ,
    unsigned long  waiptn ,
    int            mode ,
    unsigned long  *flgptn ,
    int            tmout
)
```

### コメント

デバイス I D  
イベントフラグ I D  
ウェイトするビットパターン  
ウェイトモード  
ウェイト解除時のビットパターン  
タイムアウト時間

### パラメータ

devID

デバイス I D。OpenIPDev コマンドで得られた対応するボード番号のデバイス I D を指定して下さい。

flgid

イベントフラグ I D

waiptn

ウェイトするビットパターン

mode

ウェイトモード

デバイス番号	対応数	説明
WFLG_AND_WAIT	0x0000	A N D 待ち
WFLG_OR_WAIT	0x0001	O R 待ち

\*flgptn

ウェイト解除時のビットパターンを返す領域のポインタ

tmout

タイムアウト時間 ( m s )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了
-51 ~ -9	μ ITORN サービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# ReadEventFlag

## イベントフラグの状態参照

### 機能

イベントフラグの状態参照を行います。

### コーディング

```
int
ReadEventFlag(
    DEVID          devID ,
    int            flgid ,
    RFLG_INFO     *info
)
```

### コメント

デバイスID  
 イベントフラグID  
 イベントフラグの状態情報

### パラメータ

**devID**  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**flgid**  
 イベントフラグID

**\*info**  
 イベントフラグの状態情報

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了
-51 ~ -9	μITORNサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

RFLG\_INFO のフォーマット及び内容を、下記に示します。

vpxsys.hで宣言しています

```
typedef struct {
    int            wtskid;
    unsigned long flgptn;
} RFLG_INFO;
```

セマフォ生成情報	説明
wtskid	待ちタスクID
flgptn	イベントフラグのビットパターン

# DelayIPTask

## タスク遅延

### 機能

自タスクの状態を実行状態から時間経過待ち状態へ移行し、dlytim で指定された時間が経過するのを待ちます。dlytim で指定された時間が経過した時点で、自タスクの状態を実行可能状態に移行します。dlytim = 0 を指定した場合にも、自タスクを待ち状態に移行させます。

### コーディング

```
int
DelayIPTask(
    int      dlytim
)
```

### コメント

遅延時間

### パラメータ

dlytim  
遅延時間(ms)。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-5 1 ~ -9	μITRONサービスコールエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本サービスコールは、dlytim 時間だけ実行を遅延して終了した場合に正常終了します。また、遅延時間が経過する前に待ち状態を解除するには、ReleaseWait() を実行してください。

# ReleaseWait

## 待ち状態の強制解除

### 機能

taskIDで示されるタスクが何らかの待ち状態（強制待ち状態は含まれません）の場合、それを強制的に解除します。本コマンドにより待ち状態を解除したタスクには、エラーコードとして「-49」が返ります。

### コーディング

```
int
ReleaseWait(
  int taskID
)
```

### コメント

タスクの番号(1-X)

### パラメータ

taskID  
タスクの番号(1-X)。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了
-51 ~ -9	μITRONサービスコールエラー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	タスクID設定範囲外
263	登録されていないタスクが指定された

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

二重待ち状態のタスクに対して本コマンドを実行すると、対象タスクは強制待ち状態へ移行します。

**fmOpen****ファイルオープン****機能**

指定されたファイルをアクセスモード(mode)に従いオープンします。通常はアクセスモードを新規作成や書き込みの場合は「FM\_RDWR | FM\_CREATE」、読み出しのみの場合は「FM\_RDONLY」を指定して下さい。

**コーディング**

```
FM_HFILE
fmOpen(
    DEVID  devID ,
    char   *filename ,
    int    mode
)
```

**コメント**

デバイスID  
ファイル名  
モード

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*filename**

ファイル名(最大128文字(NULL含む))

**mode**

アクセスモード

モード	対応数	説明
FM_DEFAULT	0x00	指定されたファイルが存在している場合、そのファイルを開きます。指定されたファイルが存在していない場合、新しいファイルを作成します。なお、指定されたファイルが存在している場合、FM_RDWR属性で開くため、オープン時に既存のデータは削除されませんので、既存ファイルよりも小さいサイズを上書きする場合は、「FM_CREATE   FM_RDWR」属性を指定して下さい。
FM_RDONLY	0x01	読み出しモードで開きます。ファイルがない場合は、エラーになります。データの書き換えはできません。
FM_WRONLY	0x02	書き込みモードで開きます。ファイルがない場合は、エラーになります。オープン時に既存のデータは削除されません。
FM_RDWR	0x04	読み出し、書き込みモードで開きます。ファイルがない場合は、エラーになります。オープン時に既存のデータは削除されません。
FM_CREATE	0x08	指定されたファイルが存在していない場合、新しいファイルを作成します。指定されたファイルが存在している場合、そのファイルのデータを削除します。
FM_APPEND	0x20	指定されたファイルが存在している場合、追記モードで開きます。オープン時にファイルポインタをファイルの終端へ移動します。

各モードはビットフィールドに対応していますので、複数組み合わせで使用可能です。

**リターン値**

リターン値は FM\_HFILE 型で返されます。詳細を下表に示します。

リターン値	内容
0 以外	ファイルハンドル
0	異常終了

**詳細情報**

## fmClose

## ファイルクローズ

### 機能

オープンしたファイルをクローズします。

### コーディング

```
int  
fmClose(  
    DEVID      devID ,  
    FM_HFILE   fh  
)
```

### コメント

デバイスID  
ファイルハンドル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

fh

クローズするファイルのファイルハンドル

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

### 詳細情報

ファイル作業領域の内容が更新されていた場合、その内容をフラッシュメモリに書き込みます。  
また、ファイル作業領域の解放を行います。

## fmRead

## ファイル読み込み

### 機能

オープンしたファイルの内容を読み込みます。

### コーディング

```
int
fmRead(
    DEVID
    FM_HFILE
    char
    unsigned long
)
```

```
devID ,
fh ,
*buff ,
size
```

### コメント

```
デバイスID
ファイルハンドル
データバッファ
データサイズ
```

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**fh**  
読み込むファイルのファイルハンドル

**\*buff**  
読み込むデータを保存するバッファ

**size**  
読み込むデータサイズ(バイト)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

### 詳細情報

本関数は、ファイル作業領域から読み込みを行います。

## fmWrite

## ファイル書き込み

### 機能

オープンしたファイルにデータを書き込みます。

### コーディング

```
int
fmWrite(
    DEVID          devID ,
    FM_HFILE       fh ,
    char           *buff ,
    unsigned long  size
)
```

### コメント

デバイスID  
ファイルハンドル  
データバッファ  
データサイズ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**fh**  
書き込むファイルのファイルハンドル

**\*buff**  
書き込むデータが保存されているバッファ

**size**  
書き込むデータサイズ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

### 詳細情報

本関数は、ファイル作業領域に対して書き込みを行います。  
フラッシュメモリへの書き込みタイミングは、ファイル作業領域サイズを超える場合に行います。

## fmGetFileSize

## ファイルサイズ取得

### 機能

オープンしたファイルのサイズを取得します。

### コーディング

```
unsigned long  
fmGetFileSize(  
    DEVID      devID ,  
    FM_HFILE   fh  
)
```

### コメント

デバイスID  
ファイルハンドル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

fh

ファイルサイズを取得するファイルのファイルハンドル

### リターン値

リターン値は unsigned long 型で返されます。詳細を下表に示します。

リターン値	内容
> = 0	ファイルサイズ(byte)
-1	異常終了

### 詳細情報

## fmDelete

## ファイル削除

### 機能

指定ファイルを削除します。

### コーディング

```
int  
fmDelete(  
    DEVID  devID ,  
    char   *filename  
)
```

### コメント

デバイスID  
ファイル名

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*filename**  
ファイル名

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

### 詳細情報

## fmRename

## ファイル名変更

### 機能

指定ファイル名を変更します。

### コーディング

```
int
fmRename(
    DEVID    devid,
    char     *srcfilename,
    char     *dstfilename
)
```

### コメント

デバイスID  
変更前ファイル名  
変更後ファイル名

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*srcfilename**  
変更前ファイル名

**\*dstfilename**  
変更後ファイル名

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

### 詳細情報

**fmFileList****ファイルリスト取得****機能**

フラッシュメモリ内のファイル一覧を取得します。

**コーディング**

```
int
fmFileList(
    DEVID      devid,
    FM_FLIST   *filelist,
    unsigned int count
)
```

**コメント**

デバイスID  
ファイルリスト  
ファイルリスト登録上限数

**パラメータ**

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*filelist**  
ファイルリスト

**count**  
ファイルリスト登録上限数

**リターン値**

ファイル数。リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	ファイル数
-1	異常終了

**詳細情報**

ファイル名の取得は、若番セクタから順に行います。以下に、ファイルリスト構造体 (FM\_FLIST) の詳細を示します。

```
typedef struct {
    char          name[128];
    unsigned long size;
    unsigned long attr;
    unsigned long reserve[2];
} FM_FLIST;
```

メンバ名	型	内容
name[]	char	ファイル名
size	unsigned long	ファイルサイズ
attr	unsigned long	ファイル属性
reserve[]	unsigned long	リザーブ

## fmFormat

## フラッシュメモリフォーマット

### 機能

フラッシュメモリをフォーマットします。

### コーディング

```
int  
fmFormat(  
    DEVID devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

### 詳細情報

ブートセクタを除いたセクタのみを行います。

## fmDiskSize

## ディスクサイズ取得

### 機能

フラッシュメモリの物理サイズを取得します。

### コーディング

```
unsigned long  
fmDiskSize(  
    DEVID devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は unsigned long 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	物理ディスクサイズ(byte)
0	異常終了

### 詳細情報

**fmDiskFree****未使用領域サイズ取得****機能**

フラッシュメモリの未使用領域を取得します。

**コーディング**

```
unsigned long
fmDiskFree(
    DEVID devID
)
```

**コメント**

デバイス I D

**パラメータ**

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**リターン値**

リターン値は unsigned long 型で返されます。詳細を下表に示します。

リターン値	内容
0	未使用ディスクサイズ(byte)

**詳細情報**

## fmFindFile

## ファイル検索

### 機能

指定されたファイルが存在するかどうかをチェックします。

### コーディング

```
int  
fmFindFile(  
    DEVID  devid ,  
    char   *filename  
)
```

### コメント

デバイスID  
ファイル名

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*filename**  
ファイル名

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	ファイル有り
0	ファイル無し
-1	異常終了

### 詳細情報

**fmChgAttr****ファイル属性の変更****機能**

指定されたファイルの属性を変更します。

**コーディング**

```
int
fmChgAttr(
    DEVID  devID ,
    char   *filename ,
    int    attr ,
    int    flg
)
```

**コメント**

デバイスID  
ファイル名  
属性  
フラグ

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*filename**

ファイル名

**attr**

属性

属性	対応数	説明
FM_ATTR_DEFAULT	0x00	デフォルト
FM_ATTR_RDONLY	0x01	リードオンリ
FM_ATTR_WRONLY	0x02	ライトオンリ
FM_ATTR_RDWR	0x04	リードライト
FM_ATTR_SECRET	0x08	シークレット

**flg**

attr属性ビットを有効 / 無効にするフラグ

フラグ	対応数	説明
FM_RESET	0	リセット (無効)
FM_SET	1	セット (有効)

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

**詳細情報**

## fmSetAttr

## ファイル属性の設定

### 機能

指定されたファイルの属性を設定します。

### コーディング

```
int
fmSetAttr(
    DEVID  devID ,
    FM_HFILE fh ,
    int    attr ,
    int    flg
)
```

### コメント

デバイスID  
ファイルハンドル  
属性  
フラグ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**fh**  
ファイルハンドル

**attr**  
属性

属性	対応数	説明
FM_ATTR_DEFAULT	0x00	デフォルト
FM_ATTR_RDONLY	0x01	リードオンリ
FM_ATTR_WRONLY	0x02	ライトオンリ
FM_ATTR_RDWR	0x04	リードライト
FM_ATTR_SECRET	0x08	シークレット

**flg**  
attr属性ビットを有効 / 無効にするフラグ

フラグ	対応数	説明
FM_RESET	0	リセット (無効)
FM_SET	1	セット (有効)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

### 詳細情報

## fmFileCopy

## PC-フラッシュメモリ間のファイルコピー

### 機能

PC フラッシュメモリ/フラッシュメモリ PCへファイルをコピーします。PC側にのみ実装されていますのでSH側からの呼び出しはできません。

### コーディング

```
int
fmFileCopy(
    DEVID    devID ,
    int      mode ,
    char     *srcfilename ,
    char     *dstfilename
)
```

### コメント

デバイスID  
コピーモード  
ソースファイル名  
デスティネーションファイル名

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### mode

コピーモード

モード	対応数	説明
HDD_TO_FLASH	0	HDD ->フラッシュメモリ
FLASH_TO_HDD	1	フラッシュメモリ -> HDD

#### \*srcfilename

ソース(コピー元)ファイル名

#### \*dstfilename

デスティネーション(コピー先)ファイル名

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

### 詳細情報

## vpxInitIP

## 画像処理システムの初期化

## 機能

画像処理システムを初期化します。本コマンドでは、初期化のオプションを設定可能です。尚、InitIP( )コマンドは、vpxInitIP( devID, VP910\_CONFIG, 0, INIT\_ONLY )と等価です。

## コーディング

```
int
vpxInitIP(
    DEVID devID ,
    int    cfg ,
    int    mode ,
    int    opt
)
```

## コメント

デバイスID  
初期化環境  
初期化モード  
初期化オプション

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## cfg

初期化環境

初期化環境	対応定数	説明
IP5K_CONFIG	0	IP5000互換の初期化
VP810_CONFIG	1	VP810モードでの初期化
VP910_CONFIG	2	VP910モードでの初期化

## mode

初期化モード。vpxInitIP( )コマンド内での下記動作を制御します。

初期化環境	対応定数	説明
DISPCAMERA_DISABLE	1	スルー画像表示無効
ALLOCDISPIMG_DISABLE	2	表示画面確保無効
DISPOVERLAP_DISABLE	4	オーバーラップ表示無効
CAMERA_DISABLE	8	SelectCamera( ), DispCamera( )無効 NTSC標準, YUVカラーカメラ以外を接続してコマンドを実行した場合、「カメラ選択エラー」が発生する場合があります。その場合は、このオプションを設定して下さい。

## opt

初期化オプション

初期化環境	対応数	説明
INIT_ONLY	0	初期化のみ実行
SELF_TEST	1	初期化とシステムテストを実行

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2	システム異常（ボード未実装、ボード認識不可、レジスタデータ破壊）
3	ハードウェアエラー。カメラ設定エラー。
4	レジスタテストエラー
5	コマンドテストエラー
6	画像メモリテストエラー

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

## vpxEnableLoopCamera

## ループ映像入力開始

### 機能

映像のループ入力を開始します。

### コーディング

```
int
vpxEnableLoopCamera(
    DEVID    devID ,
    int      imgID
)
```

### コメント

デバイスID  
画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgID

カメラ映像を入力する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外（不当画面番号エラー）
4 1	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## vpxDisableLoopCamera

## ループ映像入力終了

### 機能

映像のループ入力を終了します。

### コーディング

```
int
vpxDisableLoopCamera(
    DEVID          devID ,
    enum WaitMode mode
)
```

### コメント

デバイスID  
ウェイトモード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

ウェイトモード

ウェイトモード	対応定数	説明
NOWAIT	0	現在の状態を返して戻る
WAIT_FOREVER	1	オープン状態になるまでロック

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## vpxSuspendLoopCamera

## ループ映像入カ一時停止

### 機能

映像のループ入カを一時的に停止します。

### コーディング

```
int
vpxSuspendLoopCamera(
    DEVID          devID ,
    enum WaitMode  mode
)
```

### コメント

デバイスID  
ウェイトモード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

ウェイトモード

ウェイトモード	対応定数	説明
NOWAIT	0	現在の状態を返して戻る
WAIT_FOREVER	1	オープン状態になるまでロック

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## vpxResumeLoopCamera

## ループ映像入力再開

### 機能

一時停止した映像のループ入力を再開します。

### コーディング

```
int
vpxResumeLoopCamera(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## EnableLoopDisp

## ループ表示開始

### 機能

ループ表示を開始します。

### コーディング

```
int
EnableLoopDisp(
    DEVID    devID ,
    int      ImgID
)
```

### コメント

デバイスID  
画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID

表示する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# DisableLoopDisp

## ループ表示終了

### 機能

ループ表示を終了します。

### コーディング

```
int
DisableLoopDisp(
    DEVID
    enum WaitMode
)
```

devID ,  
mode

### コメント

デバイスID  
ウェイトモード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

ウェイトモード

ウェイトモード	対応定数	説明
NOWAIT	0	現在の状態を返して戻る
WAIT_FOREVER	1	オープン状態になるまでロック

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## SuspendLoopDisp

## ループ表示一時停止

### 機能

ループ表示を一時停止します。

### コーディング

```
int
SuspendLoopDisp(
    DEVID
    enum WaitMode
)
```

devID ,  
mode

### コメント

デバイスID  
ウェイトモード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

ウェイトモード

ウェイトモード	対応定数	説明
NOWAIT	0	現在の状態を返して戻る
WAIT_FOREVER	1	オープン状態になるまでロック

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## ResumeLoopDisp

## ループ表示再開

### 機能

一時停止したループ表示を再開します。

### コーディング

```
int
ResumeLoopDisp(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# vpxIP\_Rotate

## 回転

### 機能

ソース画面の画像を回転中心座標に対して時計回りに回転させ、デスティネーションウィンドウの左上 (  $s_x, s_y$  ) から X, Y 方向に移動した結果をデスティネーション画面に転送します。

### コーディング

```
int
vpxIP_Rotate(
    DEVID
    int
    int
    float
    int
    int
    int
    int
    int
    enum SPACE_OPT
)
```

```
devID ,
ImgSrc ,
ImgDst ,
angle ,
xc ,
yc ,
dx ,
dy ,
opt
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
回転角度
回転中心X座標
回転中心Y座標
X方向移動量
Y方向移動量
周辺部クリアオプション
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面番号

**ImgDst**

デスティネーション画面番号

**angle**

回転角度

**xc**

回転中心X座標 ( SRCO\_WIN相対 )

**yc**

回転中心Y座標 ( SRCO\_WIN相対 )

**dx**

X方向移動量 ( DST\_WIN相対 )

**dy**

Y方向移動量 ( DST\_WIN相対 )

**opt**

周辺部クリアオプション。

クリアオプション	対応数	内容	備考
SPACE_OCLEAR	0	周辺を0クリアする	
SPACE_NOTOUCH	1	周辺は変化させない	
SPACE_128CLEAR	2	周辺を128クリアする	

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

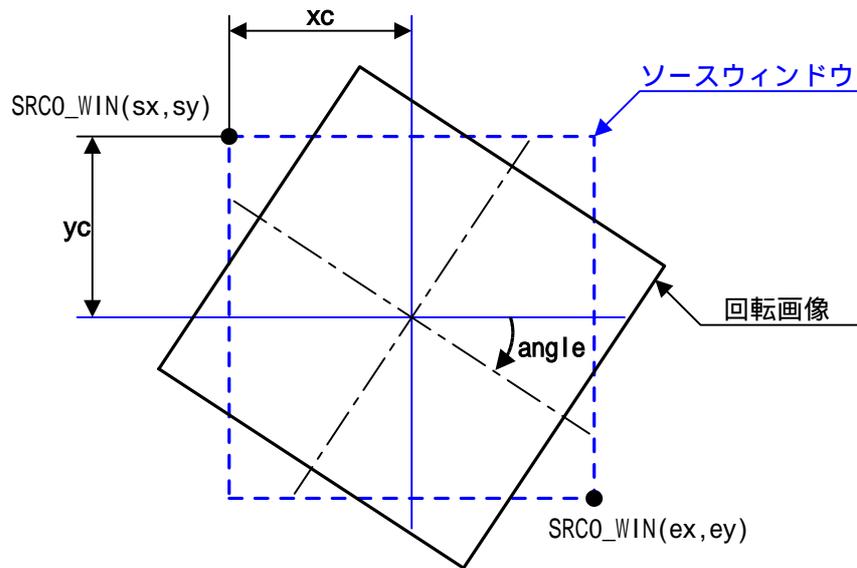
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外 (不当画面番号エラー)
23 ~ 28	パラメータエラー
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

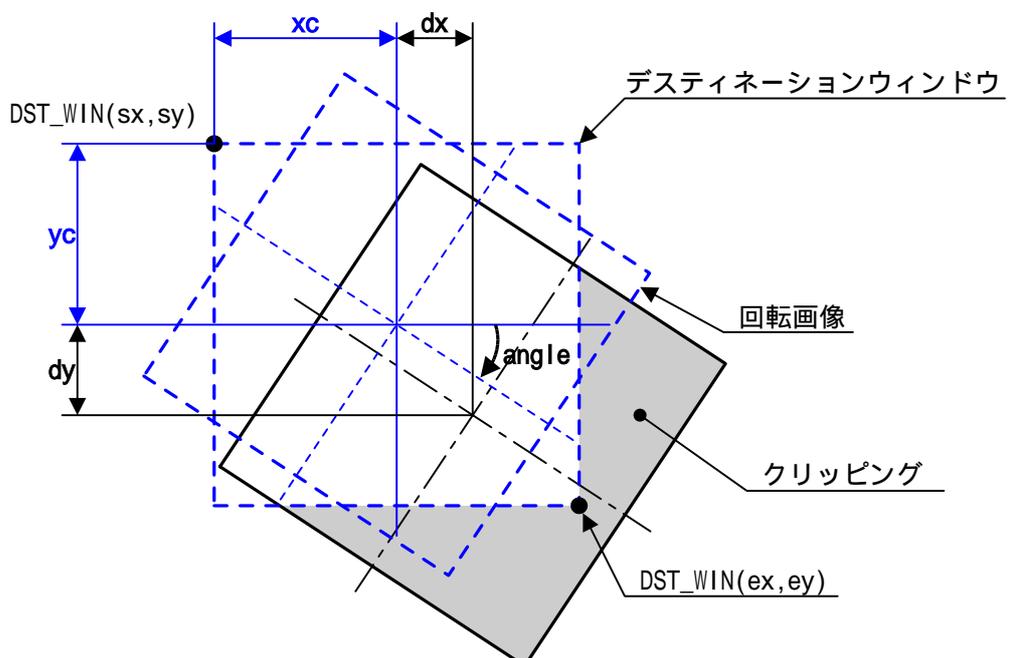
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

SRCO\_WINで指定された範囲を $(sx, sy)$ からの相対位置 $(xc, yc)$ を回転中心として $angle$ 角度回転させます。



そして、DST\_WINで指定された $(sx, sy)$ から $(dx, dy)$ 画素だけ移動した位置に回転したソース画像を転送します。また、回転画像のDST\_WINの範囲を超えた部分はクリッピングされます。



## vpxWriteConvertLUT

## 濃度変換データの設定

## 機能

濃度変換 (IP\_ConvertLUT) コマンドの濃度変換データを設定します。濃度変換の詳細については、ユーザズガイド第5章を参照して下さい。

また、ここで設定したLUTのデータは、ラベリングコマンドを実行すると書き換えられますので注意して下さい。

## コーディング

```
int
vpxWriteConvertLUT(
    DEVID    devID,
    short    *lut
)

```

## コメント

デバイスID  
濃度変換データ格納テーブル

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*lut

濃度変換データ格納テーブル (16bit × 256) のポインタ。このテーブルに濃度変換のデータを設定して下さい。

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

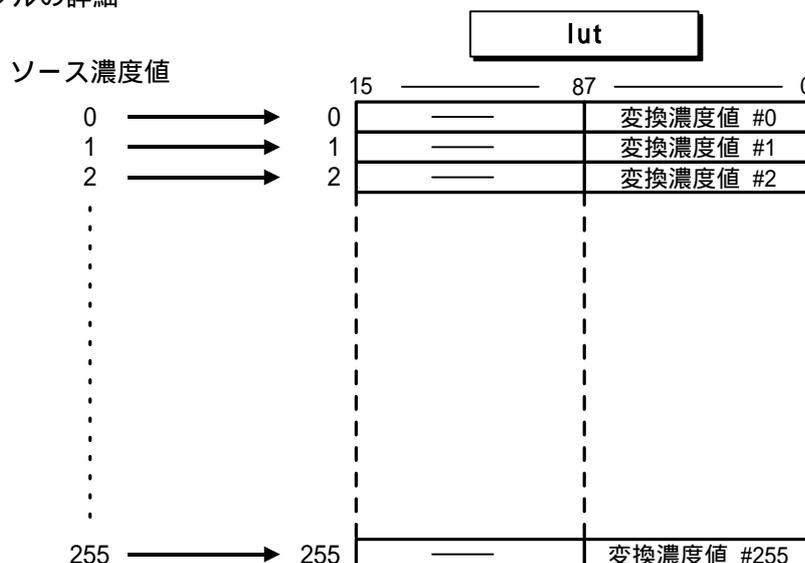
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

濃度変換テーブルの詳細



## vpxIP\_SmoothFLT

## 平滑化

### 機能

濃淡画面の平滑化を行います。vpxIP\_EdgeFLT() コマンドとの違いは処理ウィンドウ内の周辺処理にあります。vpxIP\_SmoothFLT() コマンドは、周辺部をソース画像の注目画素の画素データで描画します。

また、vpxIP\_SmoothFLT() コマンドでは、フィルタリング自体はハード的に行われますが、周辺部の処理はソフト的に行われるため、パイプライン処理を実行することはできません。パイプライン処理を行う場合は、本コマンドの代わりにvpxIP\_EdgeFLT() コマンドまたは、vpxIP\_SmoothFLT() コマンドを使用して下さい。

### コーディング

```
int
vpxIP_SmoothFLT(
    DEVID    devID ,
    int      imgSrc ,
    int      imgDst ,
    int      scale ,
    short    *COEFF
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgSrc**  
ソース画面の画面番号

**imgDst**  
デスティネーション画面の画面番号

**scale**  
ダウンシフト量 ( 0 ~ 15 )

**\*COEFF**  
荷重係数テーブルのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

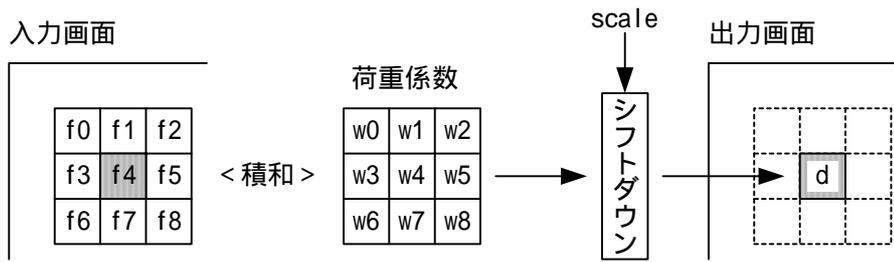
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外 ( 不当画面番号エラー )
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

平滑化では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 3 × 3 近傍の局所領域で与えられた荷重係数との積和演算を行います。



$$d = ( f_0 \times w_0 + f_1 \times w_1 + f_2 \times w_2 + f_3 \times w_3 + f_4 \times w_4 + f_5 \times w_5 + f_6 \times w_6 + f_7 \times w_7 + f_8 \times w_8 ) \gg scale$$

係数テーブルのフォーマット及び荷重係数の例を、下記に示します。

荷重係数

#0	#1	#2
#3	#4	#5
#6	#7	#8

荷重係数の例

例 1	16 16 0	例 2	7 7 7
	16 16 0		7 8 7
	0 0 0		7 7 7

	15	87	0
COEFF [0]	_____ 係数 # 0		
[1]	_____ # 1		
[2]	_____ # 2		
[3]	_____ # 3		
[4]	_____ # 4		
[5]	_____ # 5		
[6]	_____ # 6		
[7]	_____ ▼ # 7		
[8]	_____ 係数 # 8		

例 3	5 8 5	例 4	5 5 5
	8 12 8		5 20 5
	5 8 5		5 5 5

本例の場合、ダウンシフト量は 6 です。

## vpxIP\_EdgeFLT

## 濃淡画像輪郭強調

### 機能

濃淡画面の輪郭強調を行います。IP\_SmoothFLT()コマンドとの違いは処理ウィンドウ内の周辺処理にあります。IP\_EdgeFLT()コマンドは、周辺部を「0」で描画します。

### コーディング

```
int
vpxIP_EdgeFLT(
    DEVID    devID ,
    int      imgSrc ,
    int      imgDst ,
    int      scale ,
    short    *COEFF
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgSrc**  
ソース画面の画面番号

**imgDst**  
デスティネーション画面の画面番号

**scale**  
ダウンシフト量(0~15)

**\*COEFF**  
荷重係数テーブルのポインタ。COEFFの詳細については、vpxIP\_SmoothFLT()コマンドの詳細情報を参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

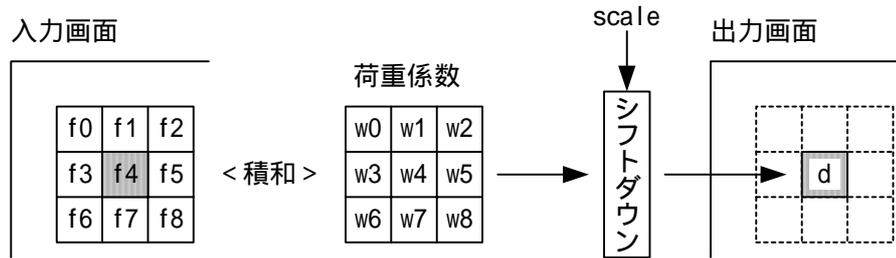
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

輪郭強調では、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域で与えられた荷重係数との積和演算を行います。  
また、対象領域の周辺（3×3近傍に領域外が含まれる場合）処理の場合は0を出力します。



$$d = ( f_0 \times w_0 + f_1 \times w_1 + f_2 \times w_2 + f_3 \times w_3 + f_4 \times w_4 + f_5 \times w_5 + f_6 \times w_6 + f_7 \times w_7 + f_8 \times w_8 ) \gg \text{scale}$$

荷重係数の例を、下記に示します。また、係数テーブルのフォーマットは、vpxIP\_SmoothFLTの詳細情報の欄を参照して下さい。

## 荷重係数の例

例 1	-1	0	1	例 2	-1	-1	-1
	-1	0	1		0	0	0
	-1	0	1		-1	-1	-1

本例の場合、ダウンシフト量は0です。

## vpxIP\_EdgeFLTAbs

## 濃淡画像輪郭強調(絶対値)

### 機能

濃淡画面の輪郭強調(絶対値)を行います。

### コーディング

```
int
vpxIP_EdgeFLTAbs(
    DEVID    devID ,
    int      imgSrc ,
    int      imgDst ,
    int      scale ,
    short    *COEFF
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgSrc**

ソース画面の画面番号

**imgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量(0~15)

**\*COEFF**

荷重係数テーブルのポインタ。COEFFの詳細については、vpxIP\_SmoothFLT()コマンドの詳細情報を参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

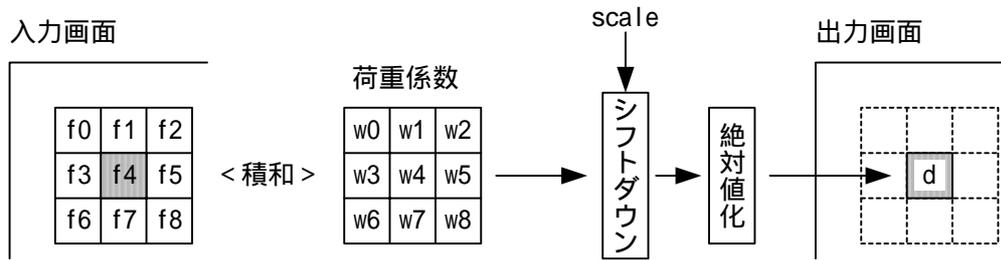
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

輪郭強調（絶対値）では、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域で与えられた荷重係数との積和演算を行い、結果を絶対値化します。また、対象領域の周辺（3×3近傍に領域外が含まれる場合）処理の場合は0を出力します。



$$d = | ( f_0 \times w_0 + f_1 \times w_1 + f_2 \times w_2 + f_3 \times w_3 + f_4 \times w_4 + f_5 \times w_5 + f_6 \times w_6 + f_7 \times w_7 + f_8 \times w_8 ) \gg \text{scale} |$$

係数テーブルのフォーマットは、vpxIP\_SmoothFLTの詳細情報の欄を、荷重係数の例は、vpxIP\_EdgeFLTの詳細情報の欄を参照して下さい。

**vpxIP\_LineFLT****ラインフィルタ****機能**

ソース画面に対し、ユーザの登録パターンでラインフィルタ処理を行い、デスティネーション画面に出力します。

**コーディング**

```
int
vpxIP_LineFLT(
    DEVID  devID ,
    int    imgSrc ,
    int    imgDst ,
    int    scale ,
    short  *COEFF
)
```

**コメント**

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

**パラメータ**

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgSrc**  
ソース画面の画面番号

**imgDst**  
デスティネーション画面の画面番号

**scale**  
ダウンシフト量(0~15)

**\*COEFF**  
荷重係数テーブルのポインタ

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

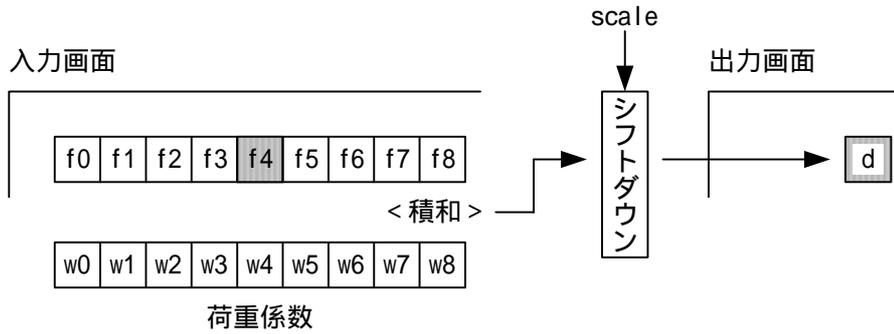
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラインフィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした1×9近傍の局所領域で与えられた荷重係数との積和演算を行います。また、対象領域の周辺(1×9近傍に領域外が含まれる部分)処理の場合は不定になります。



$$d = ( f0 \times w0 + f1 \times w1 + f2 \times w2 + f3 \times w3 + f4 \times w4 + f5 \times w5 + f6 \times w6 + f7 \times w7 + f8 \times w8 ) \gg scale$$

係数テーブルのフォーマット及び荷重係数の例を、下記に示します。

## 荷重係数

#0	#1	#2	#3	#4	#5	#6	#7	#8
----	----	----	----	----	----	----	----	----

荷重係数の例(横方向の平滑化)

7	7	7	7	7	7	7	7	7
---	---	---	---	---	---	---	---	---

本例の場合、ダウンシフト量は6です。

	15	87	0
COEFF [0]	係数 # 0		
[1]	# 1		
[2]	# 2		
[3]	# 3		
[4]	# 4		
[5]	# 5		
[6]	# 6		
[7]	▼ # 7		
[8]	係数 # 8		

**vpxIP\_LineFLTAbs****ラインフィルタ(絶対値)****機能**

ソース画面に対し、ユーザの登録パターンでラインフィルタ処理（絶対値）を行い、デスティネーション画面に出力します。

**コーディング**

```
int
vpxIP_LineFLTAbs(
    DEVID  devID ,
    int    imgSrc ,
    int    imgDst ,
    int    scale ,
    short  *COEFF
)
```

**コメント**

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgSrc**

ソース画面の画面番号

**imgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量（0～15）

**\*COEFF**

荷重係数テーブルのポインタ。COEFFの詳細については、vpxIP\_LineFLT()コマンドの詳細情報を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

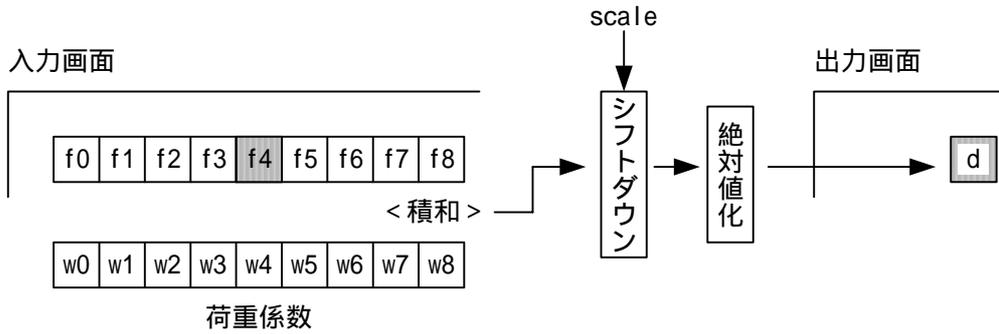
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
22	ダウンシフト量範囲外
41	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ラインフィルタでは、入力画面の指定領域内の全画素に対して、注目画素を中心とした1×9近傍の局所領域で与えられた荷重係数との積和演算を行い、絶対値化します。また、対象領域の周辺(1×9近傍に領域外が含まれる部分)処理の場合は不定になります。



$$d = | ( f_0 \times w_0 + f_1 \times w_1 + f_2 \times w_2 + f_3 \times w_3 + f_4 \times w_4 + f_5 \times w_5 + f_6 \times w_6 + f_7 \times w_7 + f_8 \times w_8 ) \gg \text{scale} |$$

係数テーブルのフォーマット及び荷重係数の例は、vpxIP\_LineFLTの詳細情報の欄を参照して下さい。

**vpxIP\_SmoothFLTF****平滑化(周辺部処理なし)****機能**

濃淡画面の平滑化を行います。vpxIP\_SmoothFLT()コマンドでは、フィルタリング自体はハード的に行われますが、周辺部の処理はソフト的に行われるため、パイプライン処理を実行することはできません。そこで、パイプライン処理を行う場合に、vpxIP\_SmoothFLT()コマンドの代わりに本コマンドを使用します。機能的には、vpxIP\_EdgeFLT()コマンドと同じです。

**コーディング**

```
int
vpxIP_SmoothFLTF(
    DEVID    devID ,
    int      imgSrc ,
    int      imgDst ,
    int      scale ,
    short    *COEFF
)
```

**コメント**

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
係数テーブルのポインタ

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgSrc**

ソース画面の画面番号

**imgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量(0~15)

**\*COEFF**

荷重係数テーブルのポインタ。COEFFの詳細については、vpxIP\_SmoothFLT()コマンドの詳細情報を参照して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	ダウンシフト量範囲外
41	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

## vpxIP\_ExtractLOArea

## ラベル毎面積抽出

### 機能

ラベル画面に対しヒストグラム処理を行い、ラベル毎面積の抽出を行います。

### コーディング

```
int
vpxIP_ExtractLOArea(
    DEVID
    int
    long
)
```

```
devID ,
ImgSrc ,
*Tbl
```

### コメント

デバイスID  
ソース画面番号  
ラベル面積データ格納テーブルへのポインタ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号。ラベリング後のデスティネーション画面を指定して下さい。

\*Tbl

ラベル面積データ格納テーブルへのポインタ。テーブルは4バイト×256の容量を確保して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外（不当画面番号エラー）
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

Tblのフォーマットを下記に示します。

ラベル		Tbl	
		31	0
			不定
1	→	1	物体 1 面積
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
2 5 3	→	2 5 3	物体 2 5 3 面積
2 5 4	→	2 5 4	物体 2 5 4 面積
2 5 5	→	2 5 5	物体 2 5 5 面積

ラベル個数よりも大きい部分は不定です

## vpxIP\_ExtractLORegionX

ラベル毎領域X座標抽出  
(Min & Max X座標)

### 機能

ラベル毎に X 座標の最小値 (MinX) と最大値 (MaxX) を抽出します。

### コーディング

```

int
vpxIP_ExtractLORegionX(
    DEVID
    int
    short
    short
)
devID ,
ImgSrc ,
*TbIMinX ,
*TbIMaxX

```

### コメント

デバイス ID  
ソース画面番号  
Min X 座標格納テーブルへのポインタ  
Max X 座標格納テーブルへのポインタ

### パラメータ

**devID**  
デバイス ID。OpenIPDevコマンドで得られた対応するボード番号のデバイス ID を指定して下さい。

**ImgSrc**  
ソース画面の画面番号。ラベリング後のデスティネーション画面を指定して下さい。

**\*TbIMinX**  
Min X 座標格納テーブルへのポインタ。テーブルは 2 バイト × 2 5 6 の容量を確保して下さい。

**\*TbIMaxX**  
Max X 座標格納テーブルへのポインタ。テーブルは 2 バイト × 2 5 6 の容量を確保して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

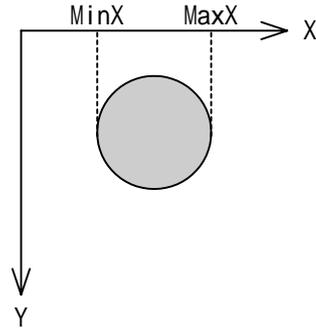
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、オンボード CPU 例外と下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報



TbIMinX、TbIMaxXのフォーマットを下記に示します。

		TbIMinX		TbIMaxX		
		15	0	15	0	
ラベル		0	不定	0	不定	
1	→	1	物体 1 座標	1	物体 1 座標	
.	.	.	.	.	.	
.	.	.	.	.	.	
.	.	.	.	.	.	
.	.	.	.	.	.	
.	.	.	.	.	.	
.	.	.	.	.	.	
2 5 3	→	2 5 3	物体 2 5 3 座標	2 5 3	物体 2 5 3 座標	ラベル個数よりも大きい 部分は不定です
2 5 4	→	2 5 4	物体 2 5 4 座標	2 5 4	物体 2 5 4 座標	
2 5 5	→	2 5 5	物体 2 5 5 座標	2 5 5	物体 2 5 5 座標	

本コマンドで得られるMinX、MaxXの座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。  
画面上の座標位置はそれぞれ

$$\begin{aligned} \text{MinX} &= \text{TbIMinX}[\text{ラベル番号}] + \text{sx} \\ \text{MaxX} &= \text{TbIMaxX}[\text{ラベル番号}] + \text{sx} \end{aligned}$$

MinX : MinX座標  
MaxX : MaxX座標  
sx : SRC0\_WIN(sx)

となります。

**vpXIP\_ExtractLORegionY**ラベル毎領域Y座標抽出  
(Min & Max Y座標)**機能**

ラベル毎にY座標の最小値(MinY)と最大値(MaxY)を抽出します。

**コーディング**

```
int
vpXIP_ExtractLORegionY(
    DEVID
    int
    short
    short
)
    devID ,
    ImgSrc ,
    *TbIMinY ,
    *TbIMaxY
```

**コメント**

デバイスID  
ソース画面番号  
Min Y座標格納テーブルへのポインタ  
Max Y座標格納テーブルへのポインタ

**パラメータ**

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号。ラベリング後のデスティネーション画面を指定して下さい。

**\*TbIMinY**  
Min Y座標格納テーブルへのポインタ。テーブルは2バイト×256の容量を確保して下さい。

**\*TbIMaxY**  
Max Y座標格納テーブルへのポインタ。テーブルは2バイト×256の容量を確保して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

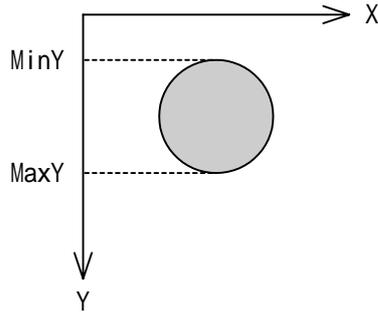
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**



TbIMinY、TbIMaxYのフォーマットを下記に示します。

		TbIMinY		TbIMaxY		
ラベル		15	0	15	0	
	0		不定	0	不定	
1	→ 1		物体 1 座標	1	物体 1 座標	
⋮	⋮		⋮	⋮	⋮	
⋮	⋮		⋮	⋮	⋮	
⋮	⋮		⋮	⋮	⋮	
⋮	⋮		⋮	⋮	⋮	
⋮	⋮		⋮	⋮	⋮	
⋮	⋮		⋮	⋮	⋮	
⋮	⋮		⋮	⋮	⋮	
2 5 3	→ 2 5 3		物体 2 5 3 座標	2 5 3	物体 2 5 3 座標	ラベル個数よりも大きい 部分は不定です
2 5 4	→ 2 5 4		物体 2 5 4 座標	2 5 4	物体 2 5 4 座標	
2 5 5	→ 2 5 5		物体 2 5 5 座標	2 5 5	物体 2 5 5 座標	

本コマンドで得られるMinY、MaxYの座標はSRCO\_WINで設定されたウィンドウからの相対座標になります。  
画面上の座標位置はそれぞれ

$\text{MinY} = \text{TbIMinY}[\text{ラベル番号}] + \text{sy}$ $\text{MaxY} = \text{TbIMaxY}[\text{ラベル番号}] + \text{sy}$	MinY : MinY座標 MaxY : MaxY座標 sy : SRCO_WIN(sy)
---	---

となります。

**vpxIP\_ProjectGO****X/Y軸への濃度累積値投影  
(16ビット)****機能**

濃淡画像でのX/Y軸への投影を行い、濃度累積(16ビット)及び濃淡画像基本特徴量抽出を行います。本コマンドは、X/Y軸への投影を同時に行いますが、濃度累積値が16ビット分までしか積算出来ませんので注意して下さい。

**コーディング**

```
int
vpxIP_ProjectGO(
    DEVID
    int
    unsigned short
    unsigned short
    IPGOFeatureTbl
)
```

```
devID ,
ImgSrc ,
*TbIX ,
*TbIY ,
*RegTbl
```

**コメント**

```
デバイスID
ソース画面番号
濃度累積値テーブル
濃度累積値テーブル
基本特徴量テーブル
```

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*TbIX**

濃度累積値(X軸への投影)を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲でX/Y軸で長い方に合わせて下さい(最大2048)。

**\*TbIY**

濃度累積値(Y軸への投影)を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲でX/Y軸で長い方に合わせて下さい(最大2048)。

**\*RegTbl**

基本特徴量を格納する領域へのポインタ

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

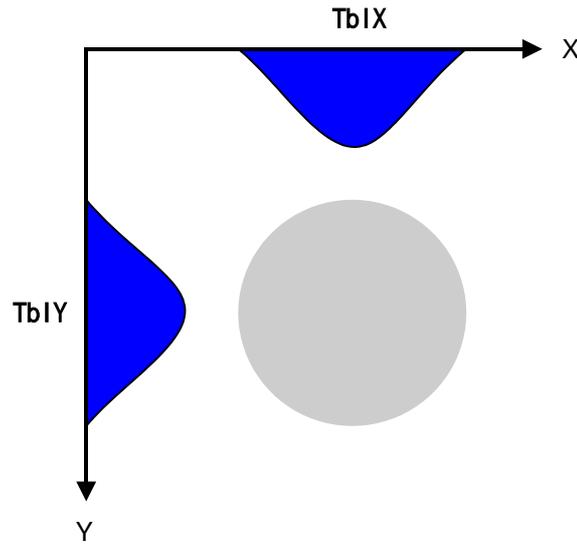
リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10	指定画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報



TbIX、TbIYのフォーマットを下記に示します。  
 また、IPGOFeatureTbIのフォーマット及び内容は、IP\_ExtractG0Features()の詳細情報の欄を参照して下さい。

TbIX			TbIY				
X座標		15 ————— 0	Y座標		15 ————— 0		
0	→	0	濃度累積値	0	→	0	濃度累積値
1	→	1	濃度累積値	1	→	1	濃度累積値
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
⋮		⋮	⋮	⋮		⋮	⋮
2046	→	2046	濃度累積値	2046	→	2046	濃度累積値
2047	→	2047	濃度累積値	2047	→	2047	濃度累積値

書き込まれるテーブルのサイズ(範囲)はSRCO\_WINで指定した範囲になります。X方向、Y方向の大きい方の範囲で書き込まれるサイズが決まります。また、その範囲が2048を越える処理はできませんので注意して下さい。

本コマンドで得られるTbIX、TbIYの投影値はSRCO\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

$$\begin{aligned}
 \text{prjX} &= \text{TbIX}[ X - \text{sx} ] \\
 \text{prjY} &= \text{TbIY}[ Y - \text{sy} ]
 \end{aligned}$$

- prjX : X軸投影値
- prjY : Y軸投影値
- X : 画面X座標
- Y : 画面Y座標
- sx : SRCO\_WIN(sx)
- sy : SRCO\_WIN(sy)

となります。

## vpxIP\_ProjectGOMaxValue

## X / Y軸への最大濃度値投影

### 機能

濃淡画像での X / Y 軸への最大濃度値投影を行い、その結果と濃淡画像基本特徴量抽出を行います。

### コーディング

```
int
vpxIP_ProjectGOMaxValue(
    DEVID
    int
    short
    short
    IPGOFeatureTbl
    devID ,
    ImgSrc ,
    *TbIX ,
    *TbIY ,
    *RegTbl
)
```

### コメント

デバイスID  
ソース画面番号  
最大濃度値 ( X 軸への投影 ) テーブル  
最大濃度値 ( Y 軸への投影 ) テーブル  
基本特徴量テーブル

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**\*TbIX**  
最大濃度値 ( X 軸への投影 ) を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲で X / Y 軸で長い方に合わせて下さい ( 最大 2 0 4 8 ) 。

**\*TbIY**  
最大濃度値 ( Y 軸への投影 ) を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲で X / Y 軸で長い方に合わせて下さい ( 最大 2 0 4 8 ) 。

**\*RegTbl**  
基本特徴量を格納する領域へのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

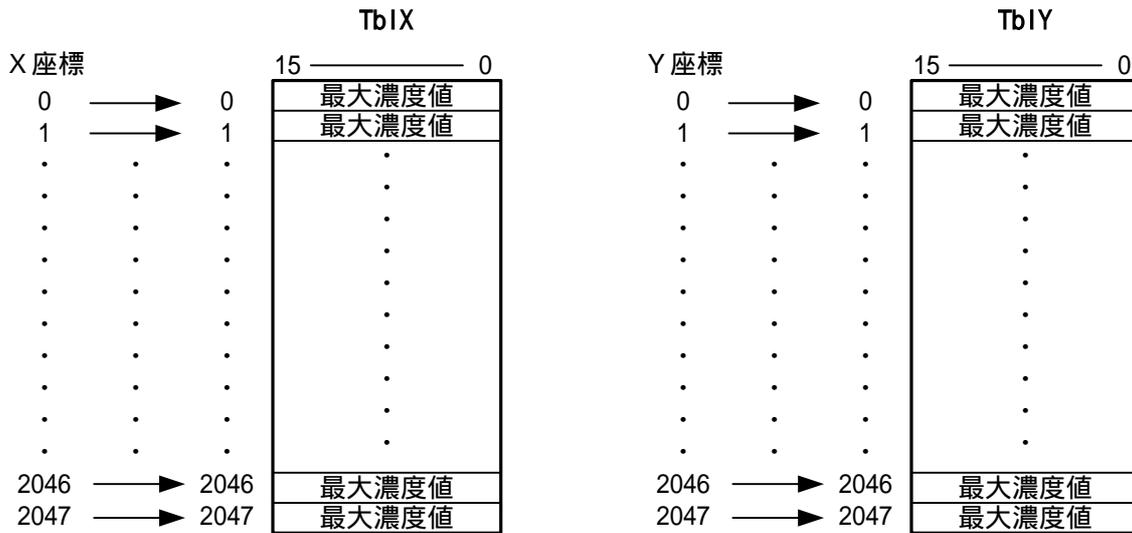
エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

TbIX、TbIYのフォーマットを下記に示します。

また、IPGFeatureTblのフォーマット及び内容は、IP\_ExtractG0Features()の詳細情報の欄を参照して下さい。



書き込まれるテーブルのサイズ（範囲）はSRCO\_WINで指定した範囲になります。X方向、Y方向の大きい方の範囲で書き込まれるサイズが決まります。また、その範囲が2048を越える処理はできませんので注意して下さい。

本コマンドで得られるTbIX、TbIYの最大濃度値はSRCO\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

$\text{prjX} = \text{TbIX}[ X - \text{sx} ]$ $\text{prjY} = \text{TbIY}[ Y - \text{sy} ]$
---

prjX : X軸最大濃度値  
 prjY : Y軸最大濃度値  
 X : 画面X座標  
 Y : 画面Y座標  
 sx : SRCO\_WIN(sx)  
 sy : SRCO\_WIN(sy)

となります。

**vpxIP\_ProjectGOMinValue****X / Y軸への最小濃度値投影****機能**

濃淡画像での X / Y 軸への最小濃度値投影を行い、その結果と濃淡画像基本特徴量抽出を行います。

**コーディング**

```
int
vpxIP_ProjectGOMinValue(
    DEVID
    int
    short
    short
    IPCOFeatureTbl
)
    devID ,
    ImgSrc ,
    *TbIX ,
    *TbIY ,
    *RegTbl
```

**コメント**

デバイス I D  
ソース画面番号  
最小濃度値 ( X 軸への投影 ) テーブル  
最小濃度値 ( Y 軸への投影 ) テーブル  
基本特徴量テーブル

**パラメータ****devID**

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*TbIX**

最小濃度値 ( X 軸への投影 ) を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲で X / Y 軸で長い方に合わせて下さい ( 最大 2 0 4 8 )。

**\*TbIY**

最小濃度値 ( Y 軸への投影 ) を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲で X / Y 軸で長い方に合わせて下さい ( 最大 2 0 4 8 )。

**\*RegTbl**

基本特徴量を格納する領域へのポインタ

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

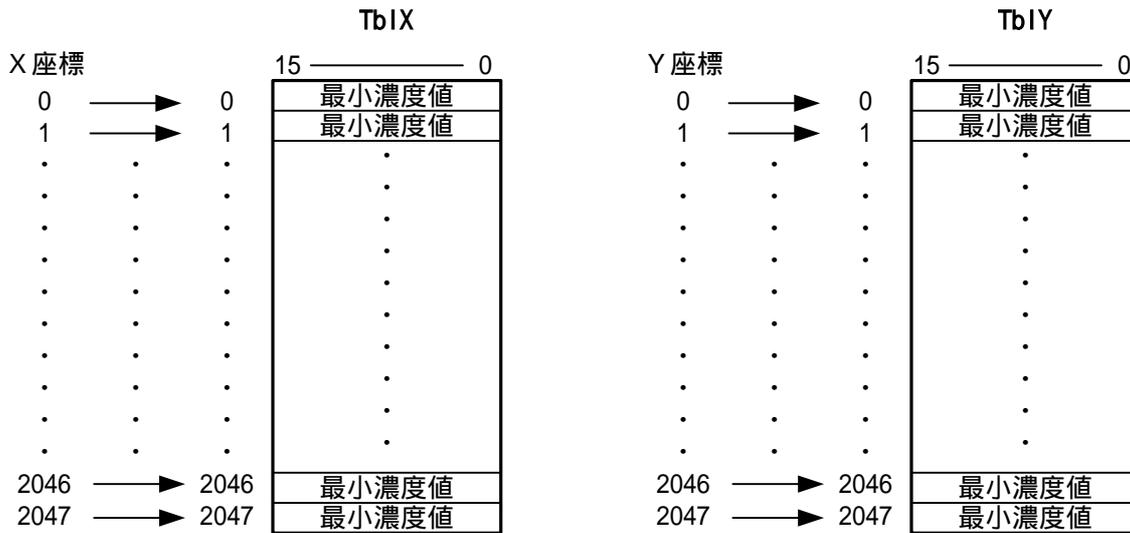
エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

TbIX、TbIYのフォーマットを下記に示します。

また、IPG0FeatureTblのフォーマット及び内容は、IP\_ExtractG0Features()の詳細情報の欄を参照して下さい。



**vpxIP\_ProjectB0****X / Yへの投影****機能**

2値画像でのX / Y軸への投影を行い、物体（白）の個数を算出します。

**コーディング**

```
int
vpxIP_ProjectB0(
    DEVID
    int
    short
    short
)
```

```
devID ,
ImgSrc ,
*TbIX ,
*TbIY
```

**コメント**

デバイスID  
ソース画面番号  
物体の個数格納テーブル  
物体の個数格納テーブル

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**\*TbIX**

物体の個数（X軸への投影）を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲でX / Y軸で長い方に合わせて下さい（最大2048）。

**\*TbIY**

物体の個数（Y軸への投影）を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定した範囲でX / Y軸で長い方に合わせて下さい（最大2048）。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10	指定画面空き領域、または範囲外（不当画面番号エラー）
49	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。



**vpxIP\_ProjectB0RegionX****領域X座標抽出  
(Y軸へのMin&MaxX投影)****機能**

2値画像でのY軸への投影を行い、物体（白）が存在するMin/MaxのX座標抽出を行います。

**コーディング**

```
int
vpxIP_ProjectB0RegionX(
    DEVID
    int
    short
    short
)
    devID ,
    ImgSrc ,
    *TbIMinX ,
    *TbIMaxX
```

**コメント**

デバイスID  
ソース画面番号  
物体が存在するMinX座標格納テーブル  
物体が存在するMaxX座標格納テーブル

**パラメータ**

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面の画面番号

**\*TbIMinX**  
物体が存在するMinX座標を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定したX方向の範囲になります（最大2048）。

**\*TbIMaxX**  
物体が存在するMaxX座標を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定したX方向の範囲になります（最大2048）。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

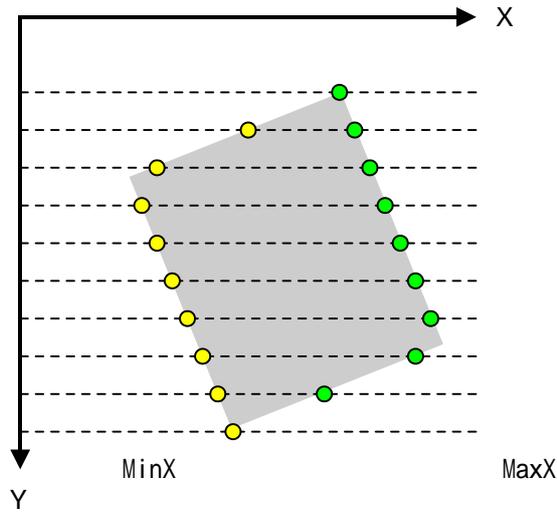
リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10	指定画面空き領域、または範囲外（不当画面番号エラー）
49	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報



TbIMinX、TbIMaxXのフォーマットを下記に示します。  
 また、物体（白）が存在しないY座標上のMinX座標には32767（0x7FFF）を、MaxX座標には0を出力します。

Y座標	TbIMinX		TbIMaxX	
	15	0	15	0
0	0	MinX座標	0	MaxX座標
1	1	MinX座標	1	MaxX座標
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
2046	2046	MinX座標	2046	MaxX座標
2047	2047	MinX座標	2047	MaxX座標

本コマンドで得られるMinX、MaxXの投影値はSRC0\_WINで設定されたウィンドウからの相対座標になります。  
 画面上の座標位置はそれぞれ

$$\begin{aligned} \text{MinX} &= \text{TbIMinX}[ Y - \text{sy} ] + \text{sx} \\ \text{MaxX} &= \text{TbIMaxX}[ Y - \text{sy} ] + \text{sx} \end{aligned}$$

- MinX : MinX座標
- MaxX : MaxX座標
- Y : 画面Y座標
- sx : SRC0\_WIN(sx)
- sy : SRC0\_WIN(sy)

となります。

## vpxIP\_ProjectB0RegionY

### 領域Y座標抽出 (X軸へのMin&MaxY投影)

#### 機能

2値画像でのX軸への投影を行い、物体（白）が存在するMin/MaxのY座標抽出を行います。

#### コーディング

```
int
vpxIP_ProjectB0RegionY(
    DEVID
    int
    short
    short
)
    devID ,
    ImgSrc ,
    *TbIMinY ,
    *TbIMaxY
```

#### コメント

デバイスID  
ソース画面番号  
物体が存在するMinY座標格納テーブル  
物体が存在するMaxY座標格納テーブル

#### パラメータ

##### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

##### ImgSrc

ソース画面の画面番号

##### \*TbIMinY

物体が存在するMinY座標を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定したY方向の範囲になります（最大2048）。

##### \*TbIMaxY

物体が存在するMaxY座標を格納する領域へのポインタ。テーブルサイズは、SRCO\_WINで指定したY方向の範囲になります（最大2048）。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

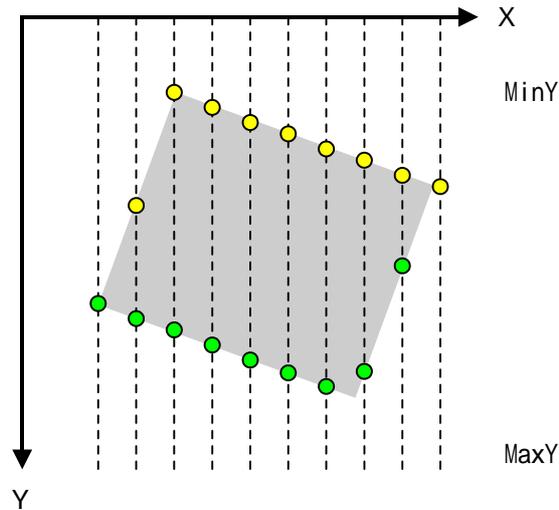
リターン値	内容
0	正常終了
-1	異常終了

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10	指定画面空き領域、または範囲外（不当画面番号エラー）
49	アクセス画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報



TbIMinY、TbIMaxYのフォーマットを下記に示します。

また、物体（白）が存在しないX座標上のMinY座標には32767（0x7FFF）を、MaxY座標には0を出力します。

X座標	TbIMinY		TbIMaxY	
	15	0	15	0
0	0	MinY座標	0	MaxY座標
1	1	MinY座標	1	MaxY座標
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
2046	2046	MinY座標	2046	MaxY座標
2047	2047	MinY座標	2047	MaxY座標

本コマンドで得られるMinY、MaxYの投影値はSRC0\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

$$\begin{aligned} \text{MinY} &= \text{TbIMinY}[X - \text{sx}] + \text{sy} \\ \text{MaxY} &= \text{TbIMaxY}[X - \text{sx}] + \text{sy} \end{aligned}$$

MinY : MinY座標  
 MaxY : MaxY座標  
 X : 画面X座標  
 sx : SRC0\_WIN(sx)  
 sy : SRC0\_WIN(sy)

となります。

# SetCameraPortConfig

## カメラポート配置の設定

### 機能

カメラポート配置の設定を行います。InitIP( )後は、「PORTCFG\_NORMAL」に設定されます。

### コーディング

```
int
SetCameraPortConfig(
    DEVID    devID ,
    int      type ,
    int      mode ,
    int      opt
)
```

### コメント

デバイスID  
カメラポート配置のタイプ  
カメラ切り換えモード  
オプション

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

type

カメラポート配置タイプの設定。(NVP-935Nでは「0」のみ使用できます)

カメラポート配置	対応定数	説明
PORTCFG_NORMAL	0	ノーマルモード
PORTCFG_CROSS	1	VP#0とVP#1をモノクロカメラとし独立して使用する。但し、トリガはPORT#0にしか出力できない。
PORTCFG_YUV_ONLY	2	カメラポート(4ポート)全てYUVカメラとして使用する。
PORTCFG_BOTH_BW_YUV	3	VP#0にモノクロカメラ、VP#1をYUVカメラとして使用する。
PORTCFG_BOTH_POBW_P1YUV	4	PORT#0(VP#0)にモノクロカメラ、PORT#1(VP#1)をYUVカメラとして使用する。
PORTCFG_BOTH_POYUV_P1BW	5	PORT#1(VP#0)にモノクロカメラ、PORT#0(VP#1)をYUVカメラとして使用する。
PORTCFG_RGB	6	コンポーネントRGBカメラを使用する。
PORTCFG_4CH	7	モノクロカメラ4CH同時入力。
PORTCFG_ACTIVE_VIDEO_PORT	8	ActiveVideoPort( )コマンドによるビデオポートの切替を行う。

mode

カメラ切り換えモードの設定。現在未使用。「0」を設定して下さい。

opt

オプション。現在未使用。「0」を設定して下さい。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

# ActiveVideoPort

## カレントビデオポートの設定

### 機能

カレントのビデオポートを設定します。

### コーディング

```
int
ActiveVideoPort(
    DEVID    devID ,
    int      vpid
)
```

### コメント

デバイスID  
ビデオポート番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

vpid

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドにより、以下のコマンドでカレントのビデオポート設定が有効になります。

- SetVideoFrame( )
- GetCamera( )
- SetVFDelay( )
- SetShutterSpeed( )
- SetTrigerMode( )
- Get2Camera( )

# SetVideoFrameMltPort

## マルチポート対応映像入力画面設定

### 機能

ビデオポート毎に映像入力画面を設定します。カメラ、映像入力での走査方式、映像入力サイズ（ビデオフレームサイズ）を、パラメータに従い設定します。通常は SelectCamera() コマンドでカメラタイプによって最適な値に設定されますが、ビデオサイズが 512 × 480 以外の場合は設定が必要です。

### コーディング

```
int
SetVideoFrameMltPort(
    DEVID
    int
    enum Interlace
    enum VideoFrameSize
)
```

```
devID ,
vpID ,
mode ,
size
```

### コメント

```
デバイスID
ビデオポート番号
ビデオ画面規定（インタレース）
ビデオ画面規定（ビデオフレームサイズ）
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

vpID

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

mode

走査方式。詳細は、SetVideoFrame( )コマンドを参照して下さい。

size

映像入出力サイズ（ビデオフレームサイズ）。詳細は、SetVideoFrame( )コマンドを参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	第一引数設定エラー
21	第二引数設定エラー
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## SelectCameraMltPort

## マルチポート対応カメラ番号選択

### 機能

ビデオポート毎にカメラ選択を行います。設定されたカメラポート番号にカメラ入力を切り替えます。コマンド内部で SetVideoFrame() コマンドと SetDispFrame() が指定されたカメラタイプにマッチしたパラメータで実行されます。また、カメラタイプが切替わる場合、自動的にディレイが挿入されます。

### コーディング

```
int
SelectCameraMltPort(
    DEVID          devid ,
    int            vpid ,
    enum CameraID  id ,
    enum CameraType type
)
```

### コメント

デバイスID  
ビデオポート番号  
カメラポート番号  
カメラタイプ

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### vpid

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

#### id

カメラポート番号

カメラポートNo.	対応定数	内容
CAMERA_PORT0	0	カメラポート#0

#### type

カメラタイプ。SelectCamera()コマンドを参照して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
3	ハードウェアエラー。カメラ設定エラー。
2 0	第一引数設定エラー
2 1	第二引数設定エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンド発行時のパラメータ（カメラポート番号、カメラタイプ）と現状のパラメータに変更がない場合は、画像処理ボードのカメラ設定を行いません。再設定を行う必要がある場合は、SetCameraPortConfig() コマンド発行後にSelectCameraMltPort() を行って下さい。

# GetCameraMltPort

## マルチポート対応映像入力

### 機能

ビデオポート毎の映像入力を行います。

### コーディング

```
int
GetCameraMltPort(
    DEVID    devid ,
    int      vpid ,
    int      imgID
)
```

### コメント

デバイスID  
ビデオポート番号  
画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

vpid

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

imgID

カメラ映像を入力する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
4 1	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# DispCameraMltPort

## マルチポート対応カメラ映像表示

### 機能

ビデオポート毎のカメラ映像表示を行います。

### コーディング

```

int
DispCameraMltPort(
    DEVID    devID ,
    int      vpid ,
    int      imgID
)

```

### コメント

デバイスID  
ビデオポート番号  
画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

vpid

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

imgID

カメラ映像を表示する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
3	ハードウェアエラー
10	ImgSrc0の不当画面番号エラー
20	第一引数の設定エラー
42	映像表示での画面サイズミスマッチエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetTriggerModeMltPort

## マルチポート対応トリガモードの設定

### 機能

ビデオポート毎にフレームシャッタカメラのトリガモードを設定します。トリガモードで映像を入力する場合は、本コマンドで TRIGGER モードに設定してから GetCameraMltPort( )コマンドで画像入力を行います。

SVP - 330では、トリガカメラの入力は、サポートされていません。

### コーディング

```
int
SetTriggerModeMltPort(
    DEVID          devid ,
    int            vpid ,
    enum TrigerMode mode
)
```

### コメント

デバイスID  
ビデオポート番号  
トリガモード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

vpid

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

mode

トリガモード

トリガモード	対応定数	説明
NOTRIGGER	0	トリガ無し
TRIGGER	1	トリガ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	第一引数の設定エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetShutterSpeedMitPort

## マルチポート対応シャッタースピードの設定

### 機能

ビデオポート毎にフレームシャッタカメラのシャッタースピードを設定します。

SVP - 330では、トリガカメラの入力は、サポートされていません。

### コーディング

```
int
SetShutterSpeedMitPort(
  DEVID
  int
  enum ShutterSpeed
)
```

```
devID ,
vpid ,
mode
```

### コメント

デバイスID  
ビデオポート番号  
シャッタースピード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

vpid

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

mode

シャッタースピード

シャッタースピード	対応定数	説明
SHUT_4000	0	シャッタースピード 1/4000(0.25mS)
SHUT_2000	1	シャッタースピード 1/2000( 0.5mS)
SHUT_1000	2	シャッタースピード 1/1000( 1.0mS)
SHUT_500	3	シャッタースピード 1/500 ( 2.0mS)
SHUT_250	4	シャッタースピード 1/250 ( 4.0mS)
SHUT_125	5	シャッタースピード 1/125 ( 8.0mS)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# Get2CameraMltPort

## マルチポート対応2カメラ映像の同時入力

### 機能

ビデオポート毎に2台のカメラ映像を同時に指定された画面に入力します。

SVP-330では、本機能は、サポートされていません。

### コーディング

```
int
Get2CameraMltPort(
    DEVID    devid ,
    int      vpid ,
    int      imgID1 ,
    int      imgID2
)
```

### コメント

デバイスID  
ビデオポート番号  
画面番号#1  
画面番号#2

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

vpid

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

imgID1

カメラポート#0のカメラ映像を入力する画面の画面番号

imgID2

カメラポート#1のカメラ映像を入力する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10	画面空き領域、または範囲外（不当画面番号エラー）
42	カメラ入力での画面サイズミスマッチ
53	画面再配置エラー（画面ロック）
54	画面再配置エラー（空領域無し）
55	Get2Cameraでサポートしているカメラ以外のカメラ設定
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

2カメラ映像の同時入力を行う場合は、同じカメラを2台用意して下さい。同一のカメラ以外の組み合わせでは、動作の保証はありません。NTSC標準カメラの場合、外部同期信号を入力できるタイプのカメラ以外は使用できません。対応カメラについては、弊社までお問い合わせ下さい。

## EnableLoopCameraMltPort

## マルチポート対応ループ映像入力開始

### 機能

ビデオポート毎の映像のループ入力を開始します。

### コーディング

```
int
EnableLoopCameraMltPort(
    DEVID    devID ,
    int      vpid ,
    int      ImgID
)
```

### コメント

デバイスID  
ビデオポート番号  
画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

vpid

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

ImgID

カメラ映像を入力する画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
3	ハードウェアエラー
10	ImgIDの不当画面番号エラー 画面空き領域、または範囲外（不当画面番号エラー）
20	第一引数の設定エラー
42	映像表示での画面サイズミスマッチエラー
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SuspendLoopCameraMltPort

## マルチポート対応ループ映像入力停止

### 機能

ビデオポート毎に映像のループ入力を一時停止します。

### コーディング

```
int
SuspendLoopCameraMltPort(
    DEVID          devid ,
    int            vpid ,
    enum WaitMode mode
)
```

### コメント

デバイスID  
ビデオポート番号  
ウェイトモード

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

vpid

ビデオポート番号

ビデオポートNo.	対応定数	内容
VIDEO_PORT0	0	ビデオポート#0
VIDEO_PORT1	1	ビデオポート#1

mode

ウェイトモード

ウェイトモード	対応数	説明
NOWAIT	0	現在の状態を返して戻る
WAIT_FOREVER	1	オープン状態になるまでロック

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
4 1	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# GetCamera

## YUVカメラ映像入力

### 機能

カメラ映像を指定された画面に入力します。映像入力画面には、AllocYUVImg( )コマンドで取得した画面を指定して下さい。

### コーディング

```
int
GetCamera(
    DEVID    devID ,
    int      imgID
)
```

### コメント

デバイスID  
画面番号

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgID**  
カメラ映像を入力する画面の画面番号。AllocYUVImg( )コマンドで取得した画面を指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
4 2	カメラ映像入力画面サイズと、ビデオフレームサイズミスマッチ
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

GetCamera()では、カメラ映像入力対象画面の種類と、SelectCamera()で設定されるカメラタイプの条件により、カラー映像やモノクロ映像を画像メモリに取り込みます。  
条件を下表に示します。

ソース画面 \ * カメラタイプ	モノクロ (BW_CAMERA)	カラ -
モノクロ (AllocImg/AllocLockImg で確保した画面)	Y画像を取り込む	Y画像のみ取り込む
カラ - (AllocYUVImg で確保した画面)	Y画像をY画面に取り込む UV画面は128 でクリアする	Y画像をY画面に取り込む UV画像をUV画面に取り込む

\* SelectCamera() で選択しているカメラタイプ

本コマンドで映像を入力する際に、画面サイズと映像入力サイズ（ビデオフレームサイズ）を一致させておく必要があります。なお、不一致の場合はエラーとなり、本処理は行われません。

対応画面サイズ一覧

映像入力 フレームサイズ	映像入力サイズ		画面サイズ	画像メモリ	
	H	V		X方向	Y方向
VIDEO_FS_256H_220V	256	220	IMG_FS_256H_256V	256	256
VIDEO_FS_256H_240V	256	240			
VIDEO_FS_512H_220V	512	220	IMG_FS_512H_256V	512	256
VIDEO_FS_512H_240V	512	240			
VIDEO_FS_320H_240V	320	240			
VIDEO_FS_256H_440V	256	440	IMG_FS_256H_512V	256	512
VIDEO_FS_256H_480V	256	480			
VIDEO_FS_512H_440V	512	440	IMG_FS_512H_512V	512	512
VIDEO_FS_512H_480V	512	480			
VIDEO_FS_320H_480V	320	480			
VIDEO_FS_512H_512V	512	512			
VIDEO_FS_640H_240V	640	240	IMG_FS_640H_256V	640	256
VIDEO_FS_640H_480V	640	480	IMG_FS_640H_512V	640	512
VIDEO_FS_1024H_240V	1024	240	IMG_FS_1024H_256V	1024	256
VIDEO_FS_1024H_480V	1024	480	IMG_FS_1024H_512V	1024	512
VIDEO_FS_1008H_470V	1008	470			
VIDEO_FS_1024H_1024V	1024	1024	IMG_FS_1024H_1024V	1024	1024
VIDEO_FS_1280H_240V	1280	240	IMG_FS_1280H_256V	1280	256
VIDEO_FS_1280H_480V	1280	480	IMG_FS_1280H_512V	1280	512
VIDEO_FS_1280H_1024V	1280	1024	IMG_FS_1280H_1024V	1280	1024
VIDEO_FS_1024H_768V	1024	768	IMG_FS_1024H_768V	1024	768

# DispCamera

## カメラ映像表示

### 機能

カメラから入力した映像を表示します。SelectCamera( )コマンドでカメラタイプがYUV\_CAMERAを指定した場合、YUVカラー映像を表示します。

### コーディング

```
int
DispCamera(
    DEVID    devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

YUVカラー映像表示する場合は、このコマンド発行前にSelectCamera() コマンドを発行してカメラタイプをカラーカメラに切り替えてください。

## IP\_ClearImg

## 画像メモリクリア(指定画面)

### 機能

パラメータで指定された画面の画像メモリを0クリアします。YUV画面のクリア時は、Y画面を0、UV画面を128でクリアします。

### コーディング

```
int
IP_ClearImg(
    DEVID  devID ,
    int    ImgID
)
```

### コメント

デバイスID  
画面番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgID

処理画面の画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## IP\_Copy

## 画像転送

## 機能

指定画面間で画像のコピーを行います。

なお、本コマンドを実行したときの範囲は、ソース画面、デスティネーション画面サイズ、SRCO\_WIN, DST\_WINで指定された範囲に従います。

## コーディング

```
int
IP_Copy(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst
)
```

## コメント

デバイスID  
コピー元の画面番号  
コピー先の画面番号

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

コピー元の画面番号

ImgDst

コピー先の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

下図に示す条件にてYUV画像を転送します。

ソース画面 \ デスティネーション画面	モノクロ	カラ -
モノクロ	Y画面のみ転送	Y画面のみ転送
YUVカラ -	Y画面のみ転送	Y/U/V画面を転送

モノクロ : AllocImg/AllocLockImg で確保した画面  
YUVカラー : AllocYUVImg で確保した画面

## ReadImg

## 画像メモリ読み出し(矩形領域)

### 機能

パラメータで指定された画面番号の画像メモリを、指定画素数分読み出します。また、指定画素数が画面（ウィンドウ）を超える場合は、超える直前までのデータを読み出します。YUVカラー画面の読み出し時は、Y画面、UV画面の順に読み出すため、読み出しデータ格納配列は2画面分確保してください。count 値は1画面の画素数を設定してください。

なお、読み出しウィンドウはSetWindow()コマンドのSYS\_WINで指定して下さい。

### コーディング

```
long
ReadImg(
    DEVID    devID ,
    int      imgID ,
    char     *ImgTbl ,
    long     count
)
```

### コメント

デバイスID  
画面番号  
読み出しデータ格納アドレス  
画素数

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgID

処理画面の画面番号

\*ImgTbl

読み出しデータ格納アドレス

count

読み出し画素数。1画面の画素数を設定してください。

### リターン値

リターン値は long 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。読み出しを行ったデータ数。
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 0	画像メモリアクセス禁止状態（画像メモリオープンエラー）
4 9	指定画面が不定画面
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

画像メモリからの読み出しサイズは、ウィンドウが有効のときはウィンドウサイズ、無効のときは画面サイズに従います。

また、YUV画面の読み出し時は、Y画面、UV画面の順に読み出すため、読み出しデータ格納配列は2画面分確保してください。count 値は1画面の画素数を設定してください。

## [ YUV画面データ読み出し例 ]

```
int ImgYUV;
char ImgTbl[512 *2];
long count= 512;
int ret;

SetWindow( devID,SYS_WIN,0,0,511,0);
ImgYUV= AllocYUVImg( devID,IMG_FS_512H_512V );
OpenImg( devID, WAIT_FOREVER );
ret= ReadImg( devID, ImgYUV, ImgTbl, count);
CloseImg( devID);
```



画像メモリからのX方向の読み出しサイズが128で割り切れる場合、PCIのDMA転送により高速に転送が行われます。それ以外の場合は、プログラム転送になりますので高速転送できなくなりますので注意して下さい。また、プログラム転送についても、X方向の読み出しサイズが4で割り切れる場合、4バイト転送モードになりますが、それ以外の場合は1バイト転送モードになり、更に転送速度が落ちますので注意して下さい。

# Writelmg

## 画像メモリ書き込み(矩形領域)

### 機能

パラメータで指定された画面番号の画像メモリに、指定画素数分書き込みます。また、指定画素数が画面（ウィンドウ）を超える場合は、超える直前までのデータが書き込まれます。YUVカラー画面の読み出し時は、Y画面、UV画面の順に読み出すため、読み出しデータ格納配列は2画面分確保してください。

count 値は1画面の画素数を設定してください。

なお、書き込みウィンドウはSetWindow()コマンドのSYS\_WINで指定して下さい。

### コーディング

```
long
Writelmg(
    DEVID    devID ,
    int      imgID ,
    char     *imgTbl ,
    long     count
)
```

### コメント

デバイスID  
画面番号  
書き込みデータアドレス  
画素数

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### imgID

処理画面の画面番号

#### \*imgTbl

書き込みデータアドレス

#### count

書き込み画素数。1画面の画素数を設定してください。

### リターン値

リターン値は long 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了。読み出しを行ったデータ数。
- 1	異常終了

異常終了時、下記の原因が考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
4 0	画像メモリアクセス禁止状態（画像メモリオープンエラー）
-	エラーリセットコマンド（ClearIPError）未発行

付録のエラー番号一覧も参照して下さい。

## 詳細情報

画像メモリへの書き込みサイズは、ウィンドウが有効のときはウィンドウサイズ、無効のときは画面サイズに従います。

また、YUV画面の書き込み時は、Y画面、UV画面の順に書き込むため、書き込みデータ格納配列は2画面分確保してください。

## [ YUV画面データ書き込み例 ]

```
int ImgYUV;
char ImgTbl[512 *2];
long count= 512;
int ret;

SetWindow( devID,SYS_WIN,0,0,511,0);
ImgYUV= AllocYUVImg( devID,IMG_FS_512H_512V );
OpenImg( devID, WAIT_FOREVER );
ret= WriteImg( devID, ImgYUV, ImgTbl, count);
CloseImg( devID);
```



画像メモリからのX方向の書き込みサイズが128で割り切れる場合、PCIのDMA転送により高速に転送が行われます。それ以外の場合は、プログラム転送になりますので高速転送できなくなりますので注意して下さい。また、プログラム転送についても、X方向の読み出しサイズが4で割り切れる場合、4バイト転送モードになりますが、それ以外の場合は1バイト転送モードになり、更に転送速度が落ちますので注意して下さい。

# AllocYUVImg

## カラー映像用画像メモリ領域確保

### 機能

画像メモリに、パラメータで指定されたサイズのYUVカラー映像を格納するための画面領域を確保し、その画面番号を返します。

YUVカラー映像用には2画面確保しますが、ユーザには1画面(Y画面)分の番号だけ返して、もう1画面(U画面)はシステムで管理します。

### コーディング

```
int
AllocYUVImg(
    DEVID          devID ,
    enum ImgFrameSize  ImgSize
)
```

### コメント

デバイスID  
画面サイズ

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSize

画面サイズ

画面サイズ	対応定数	X方向	Y方向
IMG_FS_256H_256V	0	2 5 6	2 5 6
IMG_FS_512H_256V	1	5 1 2	2 5 6
IMG_FS_256H_512V	2	2 5 6	5 1 2
IMG_FS_512H_512V	3	5 1 2	5 1 2
IMG_FS_640H_256V	4	6 4 0	2 5 6
IMG_FS_640H_512V	5	6 4 0	5 1 2
IMG_FS_1024H_512V	6	1 0 2 4	5 1 2
IMG_FS_1024H_1024V	7	1 0 2 4	1 0 2 4
IMG_FS_1024H_256V	8	1 0 2 4	2 5 6
IMG_FS_1280H_1024V	9	1 2 8 0	1 0 2 4
IMG_FS_1280H_256V	1 0	1 2 8 0	2 5 6
IMG_FS_1280H_512V	1 1	1 2 8 0	5 1 2
IMG_FS_1024H_768V	1 2	1 0 2 4	7 6 8
IMG_FS_2048H_2048V	1 6	2 0 4 8	2 0 4 8
IMG_FS_2048H_1024V	1 7	2 0 4 8	1 0 2 4
IMG_FS_1920H_1024V	1 8	1 9 2 0	7 6 8

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1 以上	確保画面番号
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	画面サイズ設定値範囲外
4 6	画面領域空きなし
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

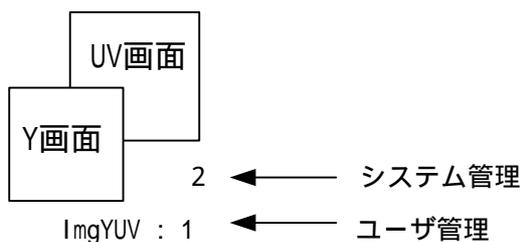
このコマンドで画面領域を確保した場合の画面設定状態を、以下に示します。  
ワーク画面などシステムで使用する空きエリアが必要となるので、そのことを考慮して5画面分空きエリアとしてください。

項目	設定値
チャンネル	不定
座標	不定
領域サイズ	パラメータで指定
データ属性	システムデータタイプによる
システムプロセッサからの画像メモリアクセス	アクセス不可

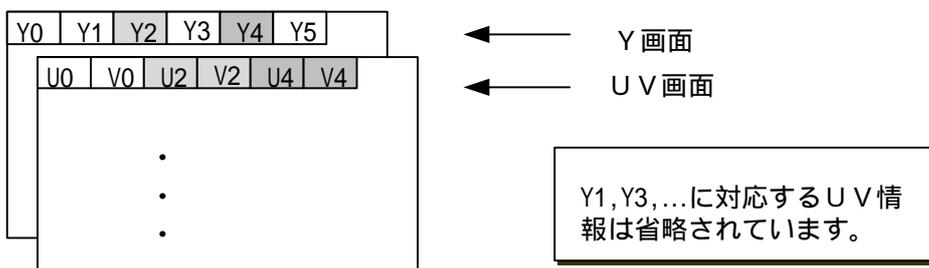
YUVカラー処理用画像メモリの確保は下図のようになります。

[ YUVカラー処理用画像メモリ確保例 ]

```
int ImgYUV;
ImgYUV = AllocYUVImg( devID, IMG_FS_512H_512V )
```



UV画面ではUV信号は交互(U:偶数列 V:奇数列)に配列していて、色情報は、2(X方向)×1(Y方向)ブロックで1セットとなっています。そのため、処理対象ウィンドウの始点X座標によっては色差情報が反転(UV → VU)してしまう可能性があります。カラー処理を行う際、始点X座標とX方向の長さを偶数に設定するように注意してください。



# GetUVImgID

## UV 画面番号取り出し

### 機能

パラメータで指定されたソース画面番号(YUV画面)に対するシステムが管理しているUV画面番号を返します。

### コーディング

```
int
GetUVImgID(
    DEVID    devID ,
    int      ImgYUV
)
```

### コメント

デバイスID  
画面番号

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgYUV**  
ソース画面番号 (YUV画面)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	UV画面の画面番号
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10	指定画面空き領域、または範囲外(不当画面番号エラー)
14	YUV画面番号に対してモノクロ画面番号か、UV画面のないカラー画面番号を指定した
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

共通部の画像処理はIP\_Copy()をのぞいてY画面のみ処理を行います。  
カラー画面の処理が必要な場合、またはUV画面のみ処理を行いたい場合は、UV画面番号をGetUVImgID()で取得しY画面とは別に処理を行います。

# ReadYUVImgTable

## カラー画面管理テーブル読み出し

### 機能

パラメータで指定されたユーザ管理画面（Y画面）およびシステム管理画面（UV画面）の画面管理テーブルを読み出し、ユーザテーブルに格納します。

### コーディング

```
int
ReadYUVImgTable(
    DEVID    devID ,
    int      ImgYUV ,
    IMG_TBL  *YTbl ,
    IMG_TBL  *UVTbl
)
```

### コメント

デバイスID  
画面番号  
Y画面管理テーブル  
UV画面管理テーブル

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgYUV**  
ソース画面番号（YUV画面）

**\*YTbl**  
ユーザ管理画面（Y画面）管理テーブルを格納するアドレス

**\*UVTbl**  
システム管理画面（UV画面）管理テーブルを格納するアドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
1 4	YUV画面番号に対してモノクロ画面番号が、UV画面のないカラー画面番号を指定した
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

画像メモリ管理テーブルのフォーマットは、次のとおりです。  
ipxsys.h で宣言しています。

```
typedef struct {
    int ch ;
    int sx ;
    int sy ;
    int size ;
    int xlng ;
    int ylng ;
    int dtyp ;
    int lock ;
    int open ;
    int dirty ;
    int color ;
    int arrange ;
    int next_id ;
} IMG_TBL ;
```

また、内容は以下に示します。

名称	意味
ch	チャンネルNo. 0 : チャンネル0 1 : チャンネル1 2 : チャンネル2 3 : チャンネル3 4 : チャンネル4
sx	開始X座標
sy	開始Y座標
size	画面サイズ番号 (確保した画像メモリ領域サイズ(enum ImageFrameSize)に対応)
xlng	領域 (X方向) サイズ
ylng	領域 (Y方向) サイズ
dtyp	0 : 符号付8ビット (-128 ~ 127 ) 1 : 符号なし8ビット (0 ~ 255 ) 2 : 2値 (0 または 255 ) - 1 : 不定画面
lock	リザーブ
open	リザーブ
dirty	リザーブ
color	画面タイプ 1 : Y画面 2 : UV画面
arrange	リザーブ
next_id	UV画面番号

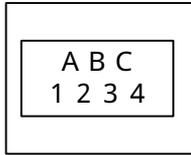
## IP\_Mask

## マスク処理

## 機能

ソース画面と2値のマスク画面をかけ、マスク領域(マスク画面の白の領域)の画像だけを取り出します。

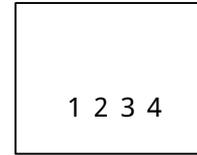
ソース画面  
(濃淡/YUV画面)



マスク画面 (2 値画面)



処理結果  
(濃淡/YUV画面)



## コーディング

```
int
IP_Mask(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgMask ,
    int      ImgDst
)
```

## コメント

デバイスID  
ソース画面番号  
マスク画面番号  
デスティネーション画面番号

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面番号

ImgMask

マスク画面番号 (2 値マスク画面番号)

ImgDst

デスティネーション画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
4 0	動的メモリ確保エラー
5 0	画像メモリチャネル競合エラー
-	エラーリセットコマンド (Clear IPErrror) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

マスク処理とは、濃淡画像を任意の形状で切り出す処理のことで、任意の形状は、2値画像データを使用します。2値画像上に任意形状の図形を発生させ、(パターン作成等で)これをマスク画像として濃淡画像と重ね合せ、2値画像の白の部分のみ濃淡画像を残し、黒の部分は0とします。

YUV画面に対して実行すると、Y画面は、2値画像の白の部分のみ濃淡画像を残し、黒の部分は0とします。UVデータは、2値画像の黒の部分を128にします。

## IP\_ClearColor

### カラー画面の色情報クリア(指定画面)

#### 機能

パラメータで指定されたYUV画面の色情報をクリアし、濃淡画像にします。

#### コーディング

```
int
IP_ClearColor(
    DEVID    devID,
    int      ImgYUV
)
```

#### コメント

デバイスID  
ソース画面番号

#### パラメータ

##### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

##### ImgYUV

ソース画面番号(YUV画面)

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外(不当画面番号エラー)
1 4	YUV画面番号に対してモノクロ画面番号か、UV画面のないカラー画面番号を指定した
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

パラメータで指定されたカラー画面のUV画面を128(色なし状態)で初期化します。

# IP\_ExtractColor

## カラー抽出処理

### 機能

パラメータで指定されたカラー画面から下記条件の色をした部分を抽出し、2値画像としてモノクロ画面に転送します。

( Ythrmin Y Ythrmax ) & ( Uthrmin U Uthrmax ) & ( Vthrmin V Vthrmax )

### コーディング

```
int
IP_ExtractColor(
    DEVID          devID ,
    int            ImgYUV ,
    int            ImgDst ,
    int            Ythrmin ,
    int            Ythrmax ,
    int            Uthrmin ,
    int            Uthrmax ,
    int            Vthrmin ,
    int            Vthrmax ,
    enum ColorOpt  opt
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
Y画像の最小しきい値  
Y画像の最大しきい値  
U画像の最小しきい値  
U画像の最大しきい値  
V画像の最小しきい値  
V画像の最大しきい値  
抽出処理オプション

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgYUV**  
ソース画面No.(YUV画面)

**ImgDst**  
デスティネーション画面番号

**Ythrmin**  
Y画像の最小しきい値(1 ~ 255)

**Ythrmax**  
Y画像の最大しきい値(1 ~ 255)

**Uthrmin**  
U画像の最小しきい値(1 ~ 255)

**Uthrmax**  
U画像の最大しきい値(1 ~ 255)

**Vthrmin**  
V画像の最小しきい値(1 ~ 255)

**Vthrmax**  
V画像の最大しきい値(1 ~ 255)

**opt**  
抽出処理オプション

抽出処理オプション	対応数	説明
NORMAL_EXTRACT	0	抽出結果を圧縮せずに出力します。
XCOMPRESS_EXTRACT	1	抽出結果をX方向へ1/2圧縮して出力します。
XYCOMPRESS_EXTRACT	2	抽出結果をX、Y方向へ1/2圧縮して出力します。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	YUV画面番号に対してモノクロ画面番号か、UV画面のないカラー画面番号を指定した
2 2 ~ 2 7	指定しきい値設定範囲外
2 8	オプション設定範囲外
5 0	画像メモリチャンネル競合エラー
5 1	ワーク用画像メモリ空きなし
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

- ・ デスティネーション画面にカラー画面を指定した場合、UV画面は処理されません。
- ・ IP\_ExtractColor()では、ワーク画面を2枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー ( エラーコード : 5 1 ) となります。  
この場合は、FreeImg()で不要な画面を解放して、空き画面を確保してください。  
抽出処理オプションにより処理速度が異なり、下記のようになります。

**処理速度**

NORMAL\_EXTRACT < XCOMPRESS\_EXTRACT < XYCOMPRESS\_EXTRACT

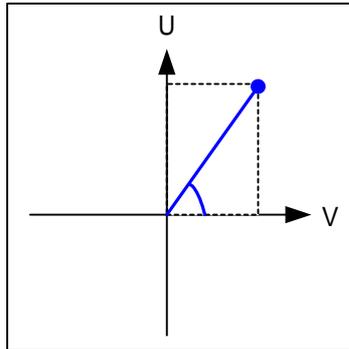
## IP\_ExtractColorRhoTheta

## カラー抽出処理 - 色彩距離・色相変換

## 機能

パラメータで指定されたカラー画面を（色彩距離）画像と（色相）画像に変換します。また下記条件の色をした部分を抽出し、2値画像としてモノクロ画面に転送します。

(Ythrmin Y Ythrmax ) & (Rthrmin Rthrmax ) & (Tthrmin Tthrmax )



演算式	$= \sqrt{\{(U - 128)^2 + (V - 128)^2\}}$
-----	--

演算式	$= \arctan \frac{(U - 128)}{(V - 128)}$
-----	---

## コーディング

```
int
IP_ExtractColorRhoTheta(
    DEVID      devid ,
    int        ImgYUV ,
    int        ImgDst ,
    int        Ythrmin ,
    int        Ythrmax ,
    int        Rthrmin ,
    int        Rthrmax ,
    int        Tthrmin ,
    int        Tthrmax ,
    enum ColorOpt opt
)
```

## コメント

デバイスID  
 ソース画面番号  
 デステネーション画面番号  
 (輝度)画像の最小しきい値  
 (輝度)画像の最大しきい値  
 (色彩距離)画像の最小しきい値  
 (色彩距離)画像の最大しきい値  
 (色相)画像の最小しきい値  
 (色相)画像の最大しきい値  
 抽出処理オプション

## パラメータ

### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### ImgYUV

ソース画面番号(YUV画面)

### ImgDst

デスティネーション画面番号

### Ythrmin

Y (輝度) 画像の最小しきい値 ( 1 ~ 255 )

### Ythrmax

Y (輝度) 画像の最大しきい値 ( 1 ~ 255 )

### Rthrmin

R (色彩距離) 画像の最小しきい値 ( 0 ~ 181 )

### Rthrmax

R (色彩距離) 画像の最大しきい値 ( 0 ~ 181 )

### Tthrmin

T (色相) 画像の最小しきい値 ( 0 ~ 360 )

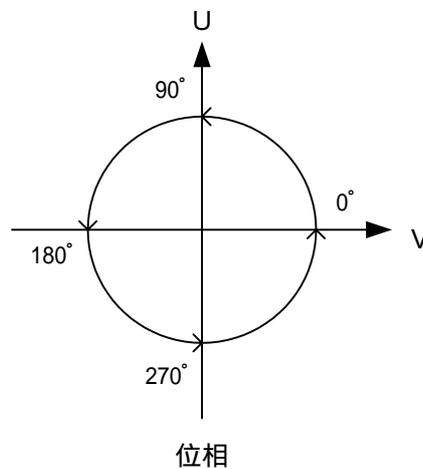
### Tthrmax

T (色相) 画像の最大しきい値 ( 0 ~ 360 )

### opt

抽出処理オプション

抽出処理オプション	対応数	説明
NORMAL_EXTRACT	0	抽出結果を圧縮せずに出力します。
XCOMPRESS_EXTRACT	1	抽出結果をX方向へ1 / 2 圧縮して出力します。
XYCOMPRESS_EXTRACT	2	抽出結果をX , Y方向へ1 / 2 圧縮して出力します。



## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外（不当画面番号エラー）
1 4	YUV画面番号に対してモノクロ画面番号か、UV画面のないカラー画面番号を指定した
2 2 ~ 2 7	指定しきい値設定範囲外
2 8	オプション設定範囲外
5 0	画像メモリチャネル競合エラー
5 1	ワーク用画像メモリ空きなし
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

- ・デスティネーション画面にカラー画面を指定した場合、UV画面は処理されません。
- ・IP\_ExtractColorRhoTheta()では、ワーク画面を5枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー（エラーコード：5 1）となります。この場合は、FreeImg()で不要な画面を解放して、空き画面を確保してください。  
抽出処理オプションにより処理速度が異なり、下記のようになります。

### 処理速度

NORMAL\_EXTRACT < XCOMPRESS\_EXTRACT < XYCOMPRESS\_EXTRACT

- ・各最小しきい値を設定値の最小値（Ythrmin=1、Rthrmin=0、Tthrmin=0）に設定し、各最大しきい値を設定値の最大値（Ythrmax=255、Rthrmax=181、Tthrmax=360）にすると、しきい値処理がなくなり、処理の高速化が計れます。

## IP\_ConvertRho

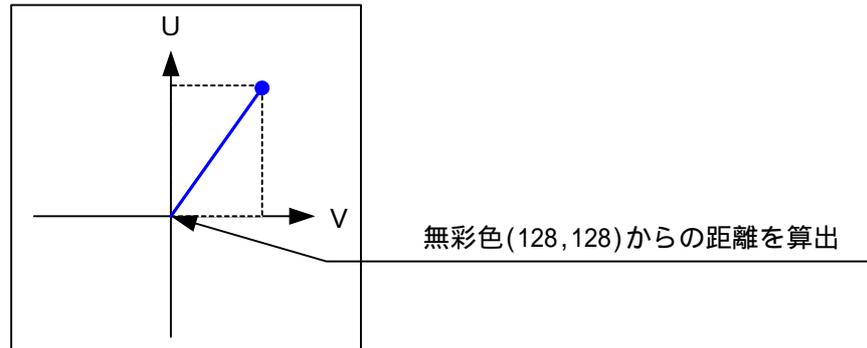
## 色彩距離変換

## 機能

パラメータで指定されたカラー画面を（色彩距離）画像に変換します。

演算式	$2 = 2^{-\text{RhoScale}} \times ((U - 128)^2 + (V - 128)^2)$
-----	---

画像メモリには、0 ~ 255 までの数値しか格納できません。そこでRhoScale を用いて、抽出しようとしている色彩距離の値（の2乗）が0 ~ 255 に入る用に調整してください。



IP\_ExtractRhoTheta()で を使用するには、IP\_ConvertRho()で抽出した の2乗の値から を計算して使用してください。この場合、RhoScale によるダウンシフトを行っていることに十分注意を払ってください。

## コーディング

```
int
IP_ConvertRho(
    DEVID
    int
    int
    int
    enum ColorOpt
)
```

```
devID ,
ImgYUV ,
ImgDstRho ,
RhoScale ,
opt
```

## コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
色彩距離抽出時のダウンシフト量
抽出処理オプション
```

## パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgYUV**  
ソース画面番号(YUV画面)

**ImgDstRho**  
デスティネーション画面番号(色彩距離)

**RhoScale**  
色彩距離抽出時のダウンシフト量(0 ~ 6)

**opt**  
抽出処理オプション

抽出処理オプション	対応数	説明
NORMAL_EXTRACT	0	抽出結果を圧縮せずに出力します。
XCOMPRESS_EXTRACT	1	抽出結果をX方向へ1 / 2 圧縮して出力します。
XYCOMPRESS_EXTRACT	2	抽出結果をX , Y方向へ1 / 2 圧縮して出力します。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	Y U V画面番号に対してモノクロ画面番号か、U V画面のないカラー画面番号を指定した
2 2	ダウンシフト値設定範囲外
2 3	オプション設定範囲外
4 9	指定画面が不定画面
5 0	画像メモリチャネル競合エラー
5 1	ワーク用画像メモリ空きなし
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

- ・デスティネーション画面にカラー画面を指定した場合、U V画面は処理されません。
- ・IP\_ConvertRho()では、ワーク画面を2 枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、異常終了 ( エラーコード : 5 1 ) となります。  
この場合は、FreeImg()で不要な画面を解放して、空き画面を確保してください。
- ・抽出処理オプションにより抽出結果を圧縮した場合、デスティネーション画面の処理対象範囲外 ( X方向1/2 圧縮なら画面右半分 ) は不定となります。これは処理時間の高速化の為、ワーク画面をクリアせずに使用しているためです。
- ・抽出処理オプションにより処理速度が異なり、下記のようになります。

#### 処理速度

NORMAL\_EXTRACT < XCOMPRESS\_EXTRACT < XYCOMPRESS\_EXTRACT

## IP\_ConvertTheta

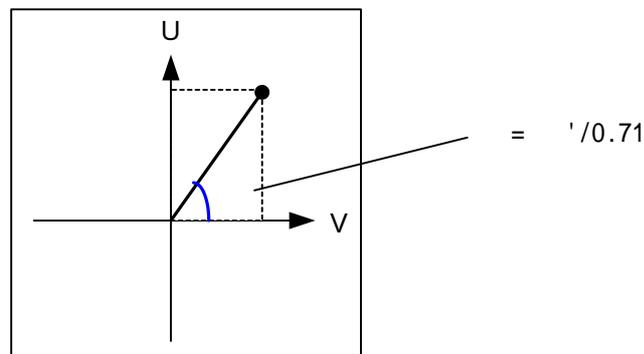
## 色相変換

## 機能

パラメータで指定されたカラー画面を（色相）画像に変換します。

演算式	$\theta = 0.71 \times \arctan \frac{(U - 128)}{(V - 128)}$
-----	--

画像メモリには、0 ~ 255 までしかデータを格納することができません。  
そこで、本コマンドでは0 ° ~ 360 °までの  $\theta$  の値を0.71 倍して画像メモリに格納しています。



IP\_ExtractColorRhoTheta()を使用する場合は、IP\_ConvertTheta()で求められた  $\theta$  の値が、実際の値を0.71倍したものであることに十分注意してください。

## コーディング

```
int
IP_ConvertTheta(
    DEVID          devID ,
    int            ImgYUV ,
    int            ImgDstTheta ,
    enum ColorOpt  opt
)
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
抽出処理オプション

## パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgYUV**  
ソース画面番号(YUV画面)

**ImgDstTheta**  
デスティネーション画面番号(色相)

**opt**  
抽出処理オプション

抽出処理オプション	対応数	説明
NORMAL_EXTRACT	0	抽出結果を圧縮せずに出力します。
XCOMPRESS_EXTRACT	1	抽出結果をX方向へ1 / 2 圧縮して出力します。
XYCOMPRESS_EXTRACT	2	抽出結果をX , Y方向へ1 / 2 圧縮して出力します。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	カラー画面番号に対して、YUV画面以外の画面番号を指定した
2 2	オプション設定範囲外
4 9	指定画面が不定画面
5 0	画像メモリチャネル競合エラー
5 1	ワーク用画像メモリ空きなし
-	エラーリセットコマンド ( ClearIPErr ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

- ・デスティネーション画面にカラー画面を指定した場合、UV画面は処理されません。
- ・IP\_ConvertTheta()では、ワーク画面を2枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、異常終了 ( エラーコード : 5 1 ) となります。  
この場合は、FreeImg()で不要な画面を解放して、空き画面を確保してください。
- ・抽出処理オプションにより抽出結果を圧縮した場合、デスティネーション画面の処理対象範囲外 ( X方向1/2 圧縮なら画面右半分 ) は不定となります。これは処理時間の高速化の為、ワーク画面をクリアせずに使用しているためです。
- ・抽出処理オプションにより処理速度が異なり、下記のようになります。

#### 処理速度

NORMAL\_EXTRACT < XCOMPRESS\_EXTRACT < XYCOMPRESS\_EXTRACT

# IP\_ConvertRhoTheta

## 色彩距離・色相変換

### 機能

パラメータで指定されたカラー画面を（色彩距離）画像と（色相）画像に変換します。

### コーディング

```
int
IP_ConvertRhoTheta(
    DEVID      devID ,
    int        ImgYUV ,
    int        ImgDstRho ,
    int        ImgDstTheta ,
    int        RhoScale ,
    enum ColorOpt opt
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号（色彩距離）  
デスティネーション画面番号（色相）  
色彩距離抽出時のダウンシフト値  
抽出処理オプション

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgYUV**  
ソース画面番号（YUV画面）

**ImgDstRho**  
デスティネーション画面番号（色彩距離）

**ImgDstTheta**  
デスティネーション画面番号（色相）

**RhoScale**  
色彩距離抽出時のダウンシフト値（0 ~ 6）

**opt**  
抽出処理オプション

抽出処理オプション	対応数	説明
NORMAL_EXTRACT	0	抽出結果を圧縮せずに出力します。
XCOMPRESS_EXTRACT	1	抽出結果をX方向へ1 / 2 圧縮して出力します。
XYCOMPRESS_EXTRACT	2	抽出結果をX , Y方向へ1 / 2 圧縮して出力します。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	カラー画面番号に対して、Y U V 画面以外の画面番号を指定した
2 3	ダウンシフト値設定範囲外
2 4	オプション設定範囲外
4 9	指定画面が不定画面
5 0	画像メモリチャネル競合エラー
5 1	ワーク用画像メモリ空きなし
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

- ・ デスティネーション画面に Y U V 画面を指定した場合、U V 画面は処理されません。
- ・ IP\_ConvertRhoTheta() では、ワーク画面を 2 枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、異常終了 ( エラーコード : 5 1 ) となります。  
この場合は、FreeImg() で不要な画面を解放して、空き画面を確保してください。
- ・ 抽出処理オプションにより抽出結果を圧縮した場合、デスティネーション画面の処理対象範囲外 ( X 方向 1/2 圧縮なら画面右半分 ) は不定となります。これは処理時間の高速化の為、ワーク画面をクリアせずに使用しているためです。
- ・ 抽出処理オプションにより処理速度が異なり、下記のようになります。

**処理速度**

NORMAL\_EXTRACT < XCOMPRESS\_EXTRACT < XYCOMPRESS\_EXTRACT

**IP\_ConvertYUVtoRGB****カラー疑似変換 (YUV RGB)****機能**

パラメータで指定されたカラーYUV画像をRGB (R画像、G画像、B画像) に変換します。

YUV RGB 変換式	$R = Y + 1.3707 \times (U - 128)$ $G = Y - 0.6980 \times (U - 128) - 0.3364 \times (V - 128)$ $B = Y + 1.7324 \times (V - 128)$
-------------------	---

**コーディング**

```
int
IP_ConvertYUVtoRGB(
    DEVID    devID ,
    int      ImgYUV ,
    int      ImgDstR ,
    int      ImgDstG ,
    int      ImgDstB
)
```

**コメント**

デバイスID  
 ソース画面番号  
 デスティネーション画面番号 ( R画面 )  
 デスティネーション画面番号 ( G画面 )  
 デスティネーション画面番号 ( B画面 )

**パラメータ**

**devID**  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgYUV**  
 ソース画面番号 ( YUV画面 )

**ImgDstR**  
 デスティネーション画面番号 ( R画面 )

**ImgDstG**  
 デスティネーション画面番号 ( G画面 )

**ImgDstB**  
 デスティネーション画面番号 ( B画面 )

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
14	カラー画面番号に対してYUV 画面以外の画面番号を指定した
49	指定画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

デスティネーション画面にYUV画面を指定した場合、UV画面は処理されません。

## IP\_ConvertYUVtoRGBfast

カラー疑似変換(YUV RGB)

### 機能

YUV画像をRGB画像に変換します。

### コーディング

```
int
IP_ConvertYUVtoRGBfast(
    DEVID    devID ,
    int      ImgYUV ,
    int      ImgRGB
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgYUV**  
ソース画面番号(YUV画面)

**ImgRGB**  
デスティネーション画面番号(RGB画面)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	指定画面空き領域、または範囲外(不当画面番号エラー)
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

IP\_ConvertYUVtoRGBfast()では、処理の高速化を図るために、変換をハードウェアで計算を行っています。そのため、誤差が大きく生じますのでご注意ください。

# OpenMultiColor

マルチカラー抽出処理可能

## 機能

マルチカラー抽出処理を可能状態にします。

## コーディング

```
int
OpenMultiColor(
    DEVID          devID ,
    enum IPMultiMethod method
)
```

## コメント

デバイスID  
マルチカラー抽出手法

## パラメータ

devID  
デバイスID

method  
マルチカラー抽出手法

マルチカラー抽出手法	内容
IPMultiYRT	Y/ / でのマルチカラー抽出オープン処理実行
IPMultiYUV	Y/U/Vでのマルチカラー抽出オープン処理実行
IPMultiHSI	Hue/Saturation/Intensityでのマルチカラー抽出オープン処理実行
IPMultiRGB	Red/Green/Blueでのマルチカラー抽出オープン処理実行

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	抽出手法設定範囲外
-	エラーリセットコマンド (ClearIPErrors) 未発行

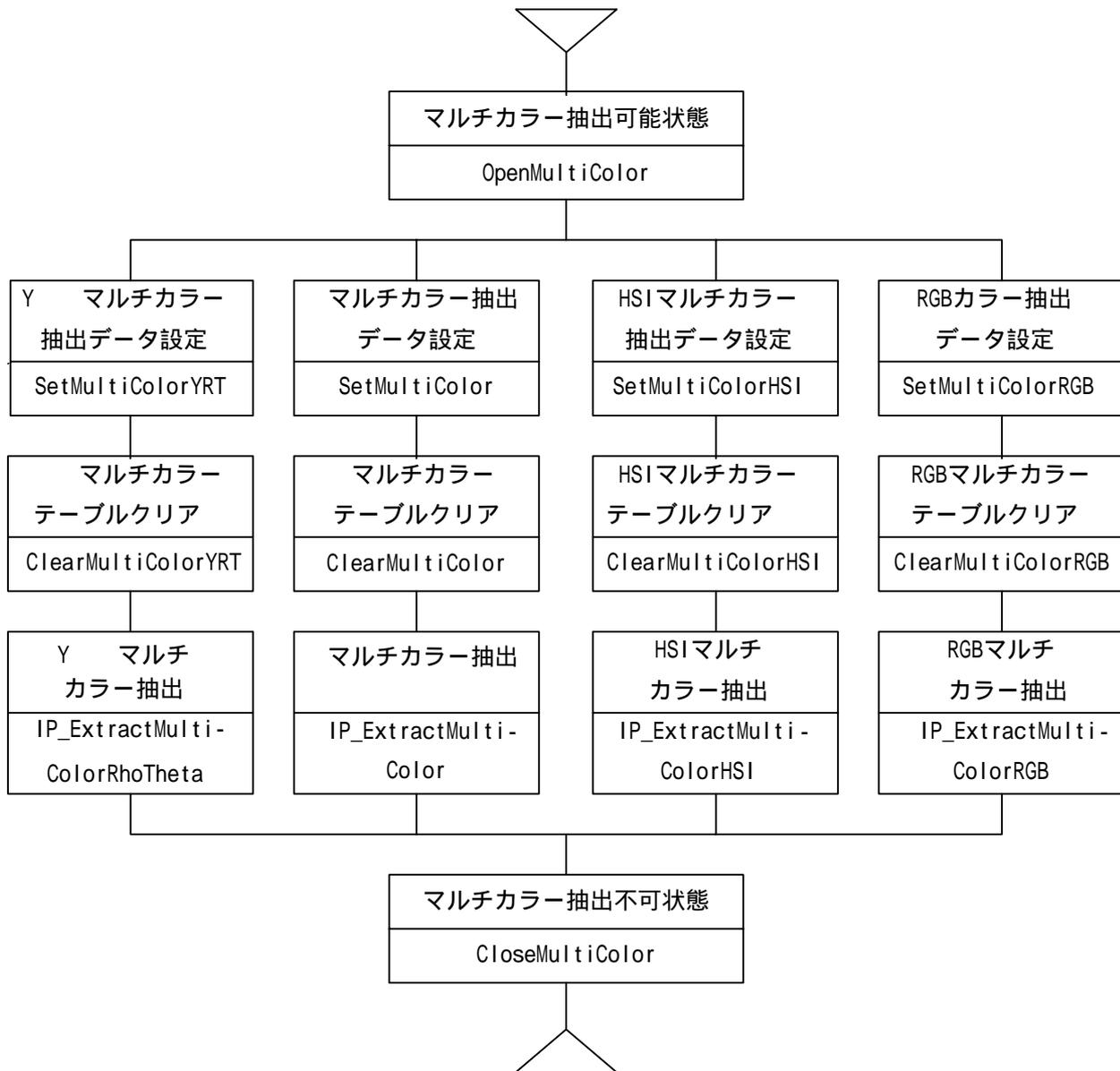
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

マルチカラー抽出処理を可能にします。マルチカラー抽出処理を実行する場合、予め本コマンドを実行して下さい。

なお各マルチカラー抽出は同時に実行できます。それぞれ本コマンドを実行して下さい。

マルチカラー抽出を使用する場合には、下記手順に従ってください。  
各手順中のマルチカラーテーブルクリアは省略可能です。



# CloseMultiColor

マルチカラー処理不可

## 機能

マルチカラー抽出処理を不可状態にします。

## コーディング

```
int
CloseMultiColor(
    DEVID          devID ,
    enum IPMultiMethod method
)
```

## コメント

デバイスID  
マルチカラー抽出手法

## パラメータ

devID  
デバイスID  
method  
マルチカラー抽出手法

マルチカラー抽出手法	内容
IPMultiYRT	Y/ / でのマルチカラー抽出クローズ処理実行
IPMultiYUV	Y/U/Vでのマルチカラー抽出クローズ処理実行
IPMultiHSI	Hue/Saturation/Intensityでのマルチカラー抽出クローズ処理実行
IPMultiRGB	Red/Green/Blueでのマルチカラー抽出クローズ処理実行

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	抽出手法設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

マルチカラー抽出処理の各手法を不可状態にします。マルチカラー抽出処理を実行した後で、本コマンドを実行して下さい。

# ClearMultiColorYRT

## Y マルチカラーテーブルクリア

### 機能

指定のY マルチカラー抽出データをクリアします。

### コーディング

```
int
ClearMultiColorYRT(
    DEVID  devID ,
    int    colorNum
)
```

### コメント

デバイスID  
変換色番号 ( 0 ~ 8 )

### パラメータ

devID  
デバイスID

colorNum  
変換色番号 ( 0 ~ 8 )  
( 0 : 全マルチカラーテーブルをクリアします )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	変換色番号設定範囲外
4 1	使用手続きエラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

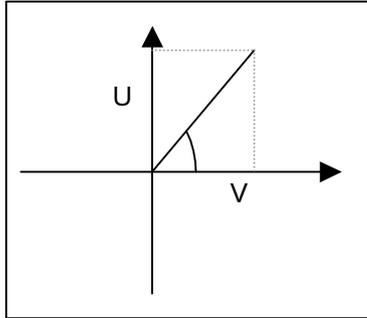
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

Y R T マルチカラー抽出データ設定 - 色彩距離・色相変換 SetMultiColorYRT() コマンドで、Y R T マルチカラー抽出用のデータを設定します。設定したデータの一部、又は全てをクリアする時は本コマンドを実行して下さい。

**SetMultiColorYRT****Y マルチカラー抽出データ設定****機能**

マルチカラー抽出・出力用のデータを設定します。パラメータで指定された範囲のY（輝度）データと（色彩距離）データと（色相）データを設定します。また、色ラベル画像出力時の出力番号を設定します。



演算式	$= \sqrt{\{ (U - 128)^2 + (V - 128)^2 \}}$
-----	--

演算式	$= \tan^{-1} \frac{U - 128}{V - 128}$
-----	---------------------------------------

**コーディング**

```
int
SetMultiColorYRT(
    DEVID  devID ,
    int    colorNum ,
    int    OutNum ,
    int    Ythrmin ,
    int    Ythrmax ,
    int    Rthrmin ,
    int    Rthrmax ,
    int    Tthrmin ,
    int    Tthrmax
)
```

**コメント**

デバイスID  
 変換色番号（1～8）  
 出力番号（1～255）  
 Y（輝度）画像の最小しきい値  
 Y（輝度）画像の最大しきい値  
 （色彩距離）画像の最小しきい値  
 （色彩距離）画像の最大しきい値  
 （色相）画像の最小しきい値  
 （色相）画像の最大しきい値

**パラメータ**

devID  
 デバイスID

colorNum  
 変換色番号（1～8）

OutNum  
 出力番号（1～255）

Ythrmin  
 Y（輝度）画像の最小しきい値（1～255）

Ythrmax  
 Y（輝度）画像の最大しきい値（1～255）

Rthrmin  
 （色彩距離）画像の最小しきい値（0～181）

Rthrmax  
 （色彩距離）画像の最大しきい値（0～181）

Tthrmin  
 （色相）画像の最小しきい値（0～360）

Tthrmax  
 （色相）画像の最大しきい値（0～360）

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	変換色番号設定範囲外
2 1	出力番号設定範囲外
2 2	Y (輝度) の最小しきい値設定範囲外
2 3	Y (輝度) の最大しきい値設定範囲外
2 4	(色彩距離) の最小しきい値設定範囲外
2 5	(色彩距離) の最大しきい値設定範囲外
2 6	(色相) の最小しきい値設定範囲外
2 7	(色相) の最大しきい値設定範囲外
4 1	使用手続きエラー
-	エラーリセットコマンド (ClearIPErrors) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

詳細はマルチカラー抽出 - 色彩距離・色相変換 IP\_ExtractMultiColorRhoTheta() を参照して下さい。

# IP\_ExtractMultiColorRhoTheta

Y マルチカラー抽出

## 機能

パラメータで指定されたYUV画面を（色彩距離）画像と（色相）画像に変換します。またマルチカラーテーブルに設定した色の部分を抽出し、オプションにより2値画像か色ラベル画像としてモノクロ画面に転送します。

## コーディング

```
int
IP_ExtractMultiColorRhoTheta(
    DEVID          devID ,
    int            ImgYUV ,
    int            ImgDst ,
    enum ColorOpt  opt ,
    enum MultiColorOpt MultiOpt
)
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
抽出処理オプション  
マルチカラー出力オプション

## パラメータ

devID  
デバイスID

ImgYUV  
ソース画面番号(YUV画面)

ImgDst  
デスティネーション画面番号(モノクロ画面)

opt  
抽出処理オプション

抽出処理オプション	内容
NORMAL_EXTRACT	抽出結果を圧縮せずに出力します。
XCOMPRESS_EXTRACT	抽出結果をX方向へ1/2圧縮して出力します。
XYCOMPRESS_EXTRACT	抽出結果をX,Y方向へ1/2圧縮して出力します。

MultiOpt  
マルチカラー出力オプション

マルチカラー出力オプション	内容
MULTICOLOR_LABEL	抽出した色毎に番号付けして出力します。デスティネーション画像はUNSIGNED8画像になります。
MULTICOLOR_BIN	抽出した色を2値化して出力します。デスティネーション画像は2値画像になります。

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

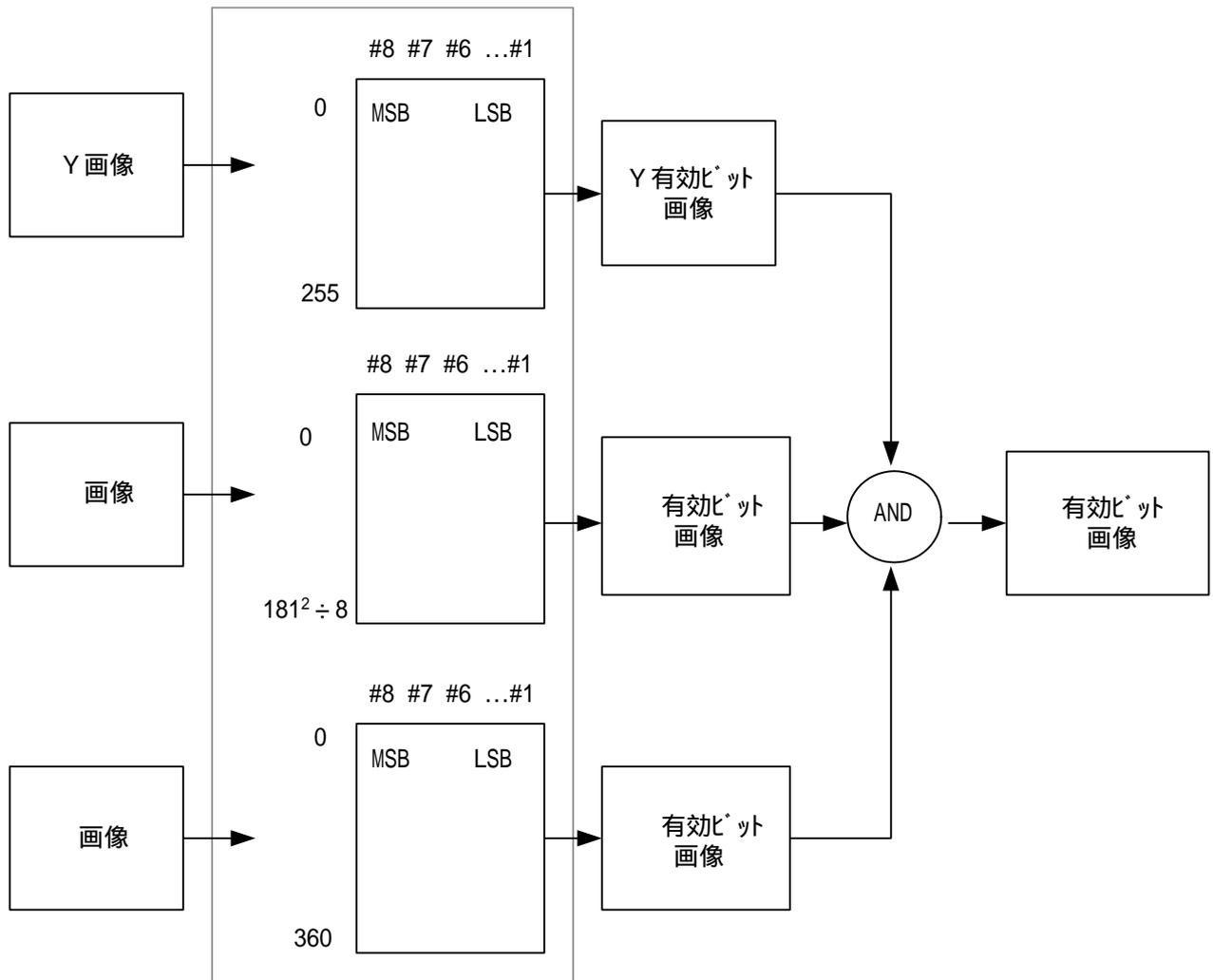
エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	カラー画面No. に対して Y U V 画面以外の画面No. を指定した
2 2	抽出処理オプション設定範囲外
2 3	マルチカラー出力オプション設定範囲外
4 1	使用手続きエラー
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
5 1	ワーク画面確保エラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

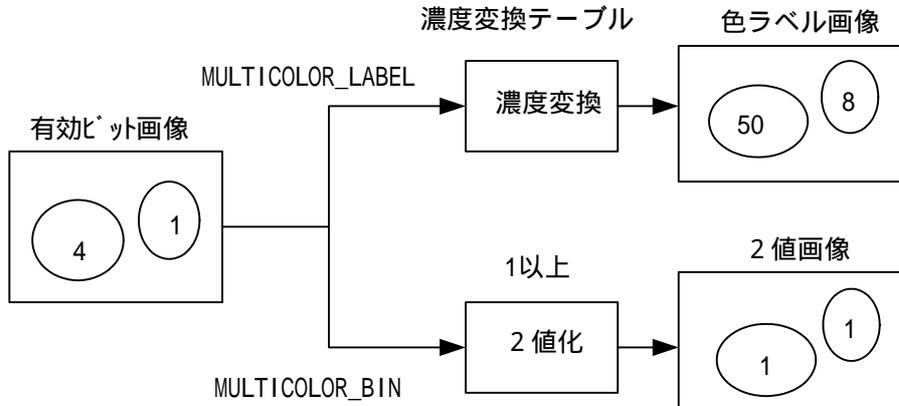
## 詳細情報

- IP\_ExtractMultiColorRhoTheta()では、ワーク画面を2枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー ( エラーコード : 5 1 ) となります。この場合は、FreeImg() で不要な画面を解放して、空き画面を確保してください。
- $$= \{ ( U - 128 )^2 + ( V - 128 )^2 \}^{1/2}$$
 ですが、ハードの制約のため、
$$\{ ( U - 128 )^2 + ( V - 128 )^2 \} \div 8$$
 として計算します。  
このため、IP\_ExtractColorRhoTheta()の抽出結果と若干の差があります。
- マルチカラー出力オプションで MULTICOLOR\_LABEL 設定時、SetMultiColorYRT() で設定した出力番号を抽出した色領域に出力します。これを色ラベル画像と呼びます。Y 範囲が重なる範囲を設定した場合、変換番号の小さい方を優先して出力します。MULTICOLOR\_BIN 設定時は、色ラベル画像を "1" で 2 値化した画像を出力します。

SetMultiColorYRT()で指定範囲  
の各ビット(#1~#2)を設定



SetMultiColorYRT()  
で指定した出力番号  
濃度変換テーブル



# ClearMultiColor

## マルチカラーテーブルクリア

### 機能

指定のマルチカラー抽出データをクリアします。

### コーディング

```
int
ClearMultiColor(
    DEVID  devID ,
    int    colorNum
)
```

### コメント

デバイスID  
変換色番号 ( 0 ~ 8 )

### パラメータ

devID  
デバイスID

colorNum  
変換色番号 ( 0 ~ 8 )  
( 0 : 全マルチカラーテーブルをクリアします )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	変換色番号設定範囲外
4 1	使用手続きエラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

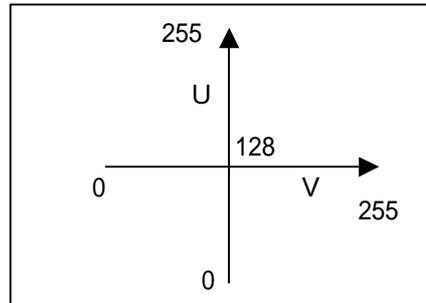
マルチカラー抽出データ設定 SetMultiColor() コマンドで、マルチカラー抽出用のデータを設定します。設定したデータの一部、又は全てをクリアする時は本コマンドを実行して下さい。

# SetMultiColor

## マルチカラー抽出データ設定

### 機能

マルチカラー抽出・出力用のデータを設定します。パラメータで指定された範囲のY（輝度）データとUデータとVデータを設定します。また、色ラベル画像出力時の出力番号を設定します。



### コーディング

```
int
SetMultiColor(
    DEVID    devID ,
    int      colorNum ,
    int      OutNum ,
    int      Ythrmin ,
    int      Ythrmax ,
    int      Uthrmin ,
    int      Uthrmax ,
    int      Vthrmin ,
    int      Vthrmax
)
```

### コメント

デバイスID  
 変換色番号 ( 1 ~ 8 )  
 出力番号 ( 1 ~ 255 )  
 Y ( 輝度 ) 画像の最小しきい値 ( 0 ~ 255 )  
 Y ( 輝度 ) 画像の最大しきい値 ( 0 ~ 255 )  
 U画像の最小しきい値 ( 0 ~ 255 )  
 U画像の最大しきい値 ( 0 ~ 255 )  
 V画像の最小しきい値 ( 0 ~ 255 )  
 V画像の最大しきい値 ( 0 ~ 255 )

### パラメータ

**devID**  
 デバイスID

**colorNum**  
 変換色番号 ( 1 ~ 8 )

**OutNum**  
 出力番号 ( 1 ~ 255 )

**Ythrmin**  
 Y ( 輝度 ) 画像の最小しきい値 ( 0 ~ 255 )

**Ythrmax**  
 Y ( 輝度 ) 画像の最大しきい値 ( 0 ~ 255 )

**Uthrmin**  
 U画像の最小しきい値 ( 0 ~ 255 )

**Uthrmax**  
 U画像の最大しきい値 ( 0 ~ 255 )

**Vthrmin**  
 V画像の最小しきい値 ( 0 ~ 255 )

**Vthrmax**  
 V画像の最大しきい値 ( 0 ~ 255 )

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	変換色番号設定範囲外
2 1	出力番号設定範囲外
2 2	Y (輝度) の最小しきい値設定範囲外
2 3	Y (輝度) の最大しきい値設定範囲外
2 4	U の最小しきい値設定範囲外
2 5	U の最大しきい値設定範囲外
2 6	V の最小しきい値設定範囲外
2 7	V の最大しきい値設定範囲外
4 1	使用手続きエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

詳細はマルチカラー抽出 IP\_ExtractMultiColor() を参照して下さい。

# IP\_ExtractMultiColor

## マルチカラー抽出

### 機能

マルチカラーテーブルに設定した色の部分を抽出し、オプションにより2値画像か色ラベル画像としてモノクロ画面に転送します。

### コーディング

```
int
IP_ExtractMultiColor(
    DEVID          devID ,
    int            ImgYUV ,
    int            ImgDst ,
    enum ColorOpt  opt ,
    enum MultiColorOpt MultiOpt
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
抽出処理オプション  
マルチカラー抽出オプション

### パラメータ

**devID**  
デバイスID

**ImgYUV**  
ソース画面番号(YUV画面)

**ImgDst**  
デスティネーション画面番号

**opt**  
抽出処理オプション

抽出処理オプション	内容
NORMAL_EXTRACT	抽出結果を圧縮せずに出力します。
XCOMPRESS_EXTRACT	抽出結果をX方向へ1/2圧縮して出力します。
XYCOMPRESS_EXTRACT	抽出結果をX,Y方向へ1/2圧縮して出力します。

**MultiOpt**  
マルチカラー出力オプション

マルチカラー出力オプション	内容
MULTICOLOR_LABEL	抽出した色毎に番号付けて出力します。デスティネーション画像はUNSIGNED画像になります。
MULTICOLOR_BIN	抽出した色を2値化して出力します。デスティネーション画像は2値画像になります。

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

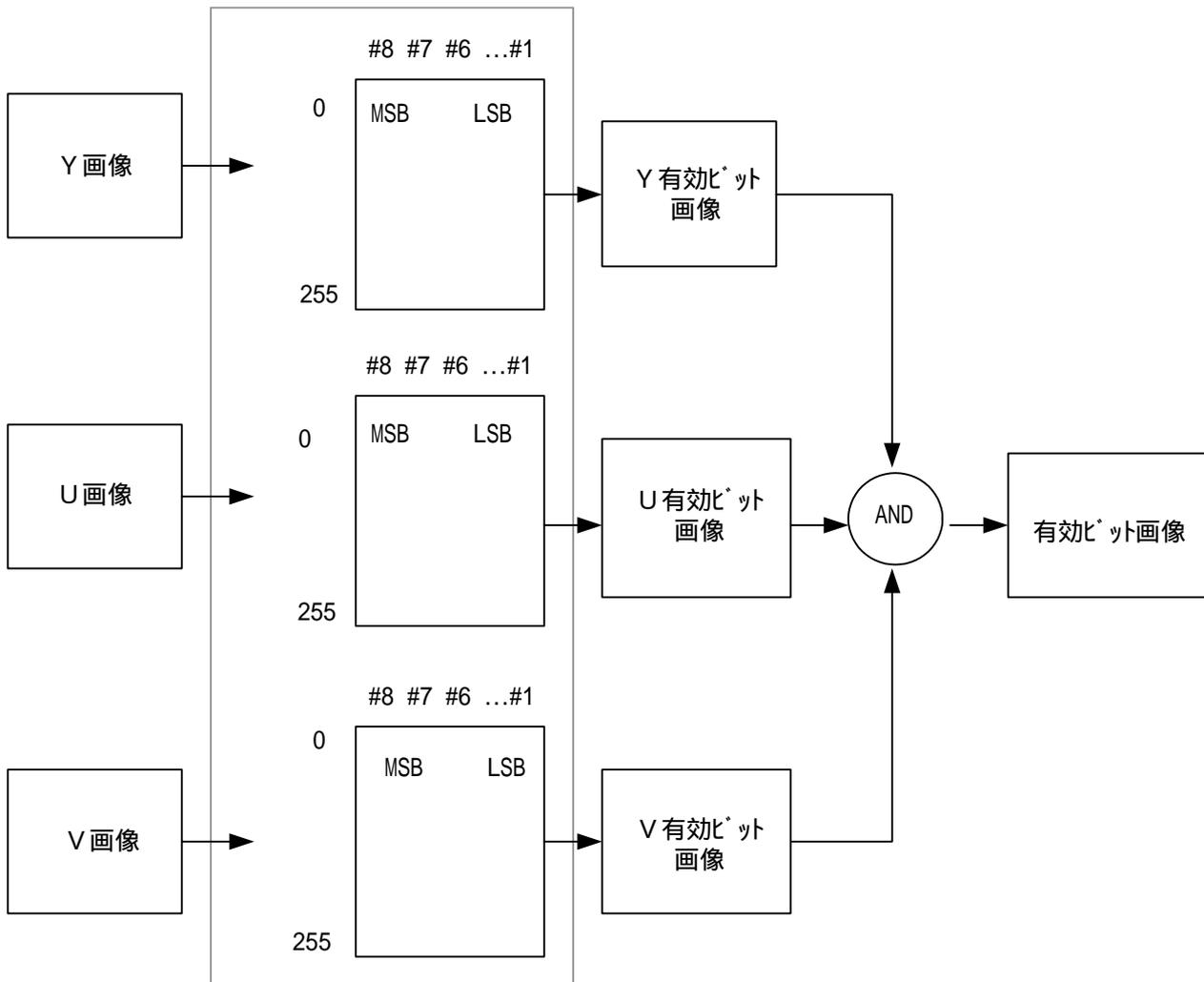
エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	カラー画面No. に対して Y U V 画面以外の画面No. を指定した
2 2	抽出処理オプション設定範囲外
2 3	マルチカラー出力オプション設定範囲外
4 1	使用手続きエラー
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
5 1	ワーク画面確保エラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

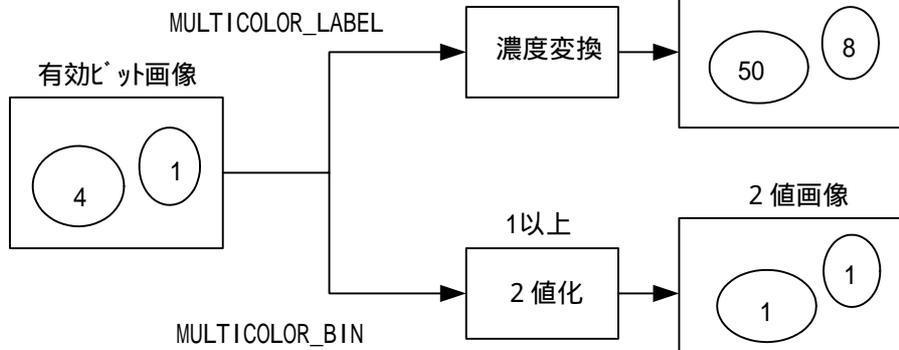
- ・ IP\_ExtractMultiColor() では、ワーク画面を 1 枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー ( エラーコード : 5 1 ) となります。この場合は、FreeImg() で不要な画面を解放して、空き画面を確保してください。
- ・ マルチカラー出力オプションで MULTICOLOR\_LABEL 設定時、SetMultiColor() で設定した出力番号を抽出した色領域に出力します。これを色ラベル画像と呼びます。Y U V 範囲が重なる範囲を設定した場合、変換番号の小さい方を優先して出力します。MULTICOLOR\_BIN 設定時は、色ラベル画像を "1" で 2 値化した画像を出力します。

SetMultiColor()で指定範囲  
の各ビット(#1~#8)を設定



SetMultiColor()で指  
定した出力番号  
濃度変換テーブル

色ラベル画像



# GetCamera

## RGBカメラ映像入力

### 機能

カメラ映像を指定された画面に入力します。本説明では、コンポーネントRGBカメラからの映像入力について説明します。

SVP-330では、RGB映像入力は、サポートされていません。

### コーディング

```
int
GetCamera(
    DEVID    devID ,
    int      imgID
)
```

### コメント

デバイスID  
画面番号 (RGB画面)

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgID

カメラ映像を入力する画面の画面番号。AllocRGBImg( )コマンドで確保した画面を指定します。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
4 2	カメラ映像入力画面サイズと、ビデオフレームサイズミスマッチ
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

コンポーネントRGBカメラからの映像入力は、4カメラ同時入力の機能を使用しています。そのため、RGB映像入力のワークフレームが必要になります。以下にコンポーネントRGBカメラの映像入力のハードウェア構成を示します。詳細は12章を参照して下さい。

カメラポート配置の設定	SetCameraPortConfig
RGBカメラの選択	SelectCamera
RGB映像入力ワーク画面の設定	AttachRGBWorkImg
RGBカメラ映像入力	GetCamera
RGB映像入力ワーク画面の解除 (必要に応じて)	DetachRGBWorkImg

コンポーネントRGBカメラからの映像入力のサンプルプログラムを示します。

```
// 画面確保
// RGB映像入力ワーク画面確保
ImgBW = AllocImg(devID, IMG_FS_512H_512V);
// RGB画面確保
ImgRGB = AllocRGBImg(devID, IMG_FS_512H_512V);
// YUV表示画面確保
ImgYUV = AllocYUVImg(devID, IMG_FS_512H_512V);

// RGB映像入力
// カメラポート配置の設定
SetCameraPortConfig(devID, PORTCFG_RGB, 0, 0);
// RGBカメラの選択
SelectCamera(devID, 0, PULNIX_TMC6700);
// RGB映像入力ワーク画面設定
AttachRGBWorkImg(devID, ImgBW);
// RGBカメラからの映像入力
GetCamera(_devID, ImgRGB);
// RGB映像入力ワーク画面設定解除
DetachRGBWorkImg(devID);

// RGBカラー映像表示
// RGB -> YUV 変換
IP_ConvertRGBtoYUVfast(devID, ImgRGB, ImgYUV);
// YUV カラー映像表示
DispImg(devID, ImgYUV);
```

本コマンドで映像を入力する際に、画面サイズと映像入力サイズ（ビデオフレームサイズ）を一致させておく必要があります。なお、不一致の場合はエラーとなり、本処理は行われません。詳細は、映像入力コマンドのカメラ映像入力を参照して下さい。

# AttachRGBWorkImg

## RGB映像入力ワークフレームの設定

### 機能

R G B 映像入力を行う場合、ワークフレームが必要です。本コマンドでそのワークフレームを設定し、GetCamera()コマンドでR G B 画面にR G B の映像を入力して下さい。

### コーディング

```
int
AttachRGBWorkImg(
    DEVID
    int
)
```

devID ,  
ImgID

### コメント

デバイス I D  
画面番号

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I D を指定して下さい。

ImgID

ワークフレームの画面番号。モノクロ画面の画面番号を指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	不当画面番号
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## DetachRGBWorkImg

### RGB映像入力ワークフレームの解除

#### 機能

AttachRGBWorkImg()コマンドで設定したワークフレームを解除します。

#### コーディング

```
int  
DetachRGBWorkImg(  
    DEVID  
)
```

#### コメント

devID

デバイスID

#### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

# AllocRGBImg

## 画像メモリ領域確保(RGB画面)

### 機能

画像メモリに、パラメータで指定されたサイズのRGB画面領域を確保します。

### コーディング

```
int
AllocRGBImg(
    DEVID          devID ,
    enum ImageFrameSize size
)
```

### コメント

デバイスID  
画像メモリ画面サイズ

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### size

画面サイズ

画面サイズ	対応定数	X方向	Y方向
IMG_FS_256H_256V	0	2 5 6	2 5 6
IMG_FS_512H_256V	1	5 1 2	2 5 6
IMG_FS_256H_512V	2	2 5 6	5 1 2
IMG_FS_512H_512V	3	5 1 2	5 1 2
IMG_FS_640H_256V	4	6 4 0	2 5 6
IMG_FS_640H_512V	5	6 4 0	5 1 2
IMG_FS_1024H_512V	6	1 0 2 4	5 1 2
IMG_FS_1024H_1024V	7	1 0 2 4	1 0 2 4
IMG_FS_1024H_256V	8	1 0 2 4	2 5 6
IMG_FS_1280H_1024V	9	1 2 8 0	1 0 2 4
IMG_FS_1280H_256V	1 0	1 2 8 0	2 5 6
IMG_FS_1280H_512V	1 1	1 2 8 0	5 1 2
IMG_FS_1024H_768V	1 2	1 0 2 4	7 6 8
IMG_FS_2048H_2048V	1 6	2 0 4 8	2 0 4 8
IMG_FS_2048H_1024V	1 7	2 0 4 8	1 0 2 4
IMG_FS_1920H_1024V	1 8	1 9 2 0	7 6 8

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	確保した画面の画面番号
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	画面サイズ設定値範囲外
4 6	画面領域空きなし
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

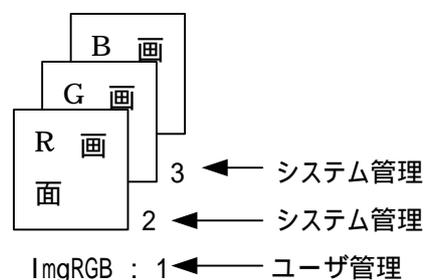
本コマンドで画像領域を確保した場合の画像設定状態を、下記に示します。  
ワーク画面などシステムで使用する空きエリアが必要となるので、そのことを考慮して5画面分空きエリアとすることを推奨します。

項 目	設 定 値
チャンネル	不定
座標	不定
領域サイズ	パラメータで指定
データ属性	システムデータタイプによる
システムプロセッサからの画像メモリアクセス	アクセス不可

RGBカラー処理用画像メモリの確保は下図のようになります。

[ RGBカラー処理用画像メモリ確保例 ]

```
int ImgRGB;
ImgRGB= AllocRGBImg( devID, IMG_FS_512H_512V );
```



RGB画面では色情報は、3画面で1セットとなっています。

# ReadRGBImgTable

## RGB画像メモリ管理テーブル読み出し

### 機能

パラメータで指定されたRGB画面番号の画像メモリ管理テーブルを読み出し、ユーザテーブルに格納します。

### コーディング

```
int
ReadRGBImgTable(
    DEVID    devID ,
    int      imgID ,
    IMG_TBL  *Rtbl ,
    IMG_TBL  *Gtbl ,
    IMG_TBL  *Btbl
)
```

### コメント

デバイスID  
RGB画面番号  
R画像メモリ管理テーブル  
G画像メモリ管理テーブル  
B画像メモリ管理テーブル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

imgID

対象画面の画面番号

\*Rtbl

R画像メモリ管理テーブルを格納する領域のポインタ

\*Gtbl

G画像メモリ管理テーブルを格納する領域のポインタ

\*Btbl

B画像メモリ管理テーブルを格納する領域のポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ipxsys.h で宣言しています。

```
typedef struct {
    int    ch ;
    int    sx ;
    int    sy ;
    int    size ;
    int    xlng ;
    int    ylng ;
    int    dtyp ;
    int    lock ;
    int    open ;
    int    dirty ;
    int    color ;
    int    arrange ;
    int    next_id ;
} IMG_TBL ;
```

名称	意味
ch	チャンネルNo. 0 : チャンネル0 1 : チャンネル1 2 : チャンネル2 3 : チャンネル3 4 : チャンネル4
sx	開始 X 座標
sy	開始 Y 座標
size	画面サイズ番号 (確保した画像メモリ領域サイズ(enum ImageFrameSize)に対応)
xlng	領域 ( X 方向 ) サイズ
ylng	領域 ( Y 方向 ) サイズ
dtyp	0 : 符号 8 ビット ( - 1 2 8 ~ 1 2 7 ) 1 : 符号なし 8 ビット ( 0 ~ 2 5 5 ) 2 : 2 値 ( 0 または 2 5 5 ) - 1 : 不定画面
lock	リザーブ
open	リザーブ
dirty	リザーブ
color	画面タイプ 3 : R 画面 4 : G 画面 5 : B 画面
arrange	リザーブ
next_id	G / B 画面番号

# GetGImgID

## G画面番号抽出

### 機能

R G B 画面の画面番号から G 画面の画面番号を読み出します。

### コーディング

```
int
GetGImgID(
    DEVID    devID,
    int      RGBimgID
)
```

### コメント

デバイス I D  
R G B 画面番号

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

RGBimgID

対象画面の R G B 画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	G画面の画面 I D
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

カラー画面の処理が必要な場合、またはG画面のみ処理を行いたい場合は、G画面番号をGetGImgID()で取得しR画面とは別に処理を行います。

# GetBImgID

## B画面番号抽出

### 機能

R G B画面の画面番号から B画面の画面番号を読み出します。

### コーディング

```
int
GetBImgID(
    DEVID    devID,
    int      RGBimgID
)
```

### コメント

デバイス I D  
R G B画面番号

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

RGBimgID

対象画面の R G B画面番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	B画面の画面 I D
0	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

カラー画面の処理が必要な場合、または B画面のみ処理を行いたい場合は、B画面番号をGetBImgID()で取得し R画面とは別に処理を行います。

## IP\_ConvertHue

## 色相変換

## 機能

パラメータで指定されたRGBカラー画面をHue（色相）画像に変換します。

演算式	$I = \max(R, G, B)$ $I = 0: S = 0: H = 0(\text{不定})$ $I > 0: (i = \min(R, G, B))$ $S = (I - i) * 255 / I$ $R = I: H = (G - B) * 60 / (I - i)$ $G = I: H = (B - R) * 60 / (I - i) + 120$ $B = I: H = (R - G) * 60 / (I - i) + 240$
-----	---

(  $H < 0$  の時は、  $H = H + 360$  とする )

画像メモリには、0 ~ 255 までしかデータを格納することができません。そこで、本コマンドでは0 ~ 360度までのHueの値を0.71倍して画像メモリに格納しています。

## コーディング

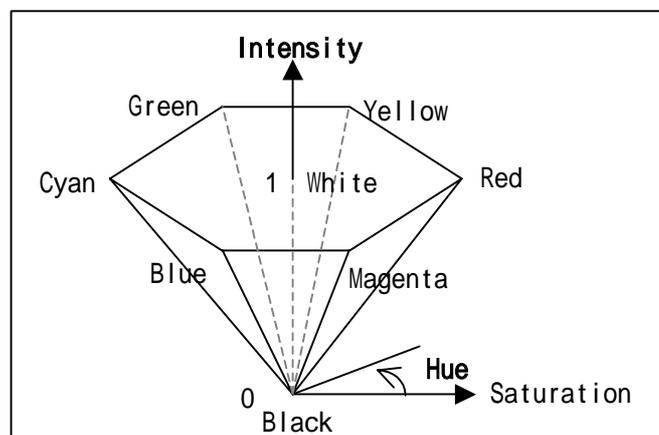
```
int
IP_ConvertHue(
    DEVID  devID ,
    int    ImgRGB ,
    int    ImgDst
)
```

## コメント

デバイスID  
ソース画面番号(RGBカラー画面)  
デスティネーション画面番号

## パラメータ

devID  
デバイスID  
ImgRGB  
ソース画面番号(RGBカラー画面)  
ImgDst  
デスティネーション画面番号



**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	R G B カラー画面番号に対して R G B 画面以外の画面番号を指定した
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
5 1	ワーク画面確保エラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

- ・デスティネーション画面に Y U V / R G B カラー画面を指定した場合、U V / G , B 画面は処理されません。
- ・IP\_ConvertHue()では、ワーク画面を 8 枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー ( エラーコード : 5 1 ) となります。この場合は、FreeImg() で不要な画面を解放して、空き画面を確保してください。

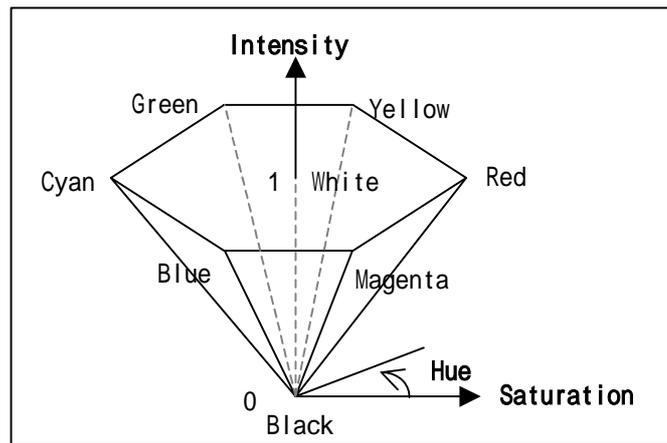
# IP\_ConvertSaturation

彩度変換

## 機能

パラメータで指定されたRGBカラー画面をSaturation(彩度)画像に変換します。

演算式	$I = \max(R, G, B)$ $I = 0: S = 0: H = 0(\text{不定})$ $I \neq 0: (i = \min(R, G, B))$ $S = (I - i) * 255 / I$
-----	---



## コーディング

```
int
IP_ConvertSaturation(
    DEVID  devID ,
    int    ImgRGB ,
    int    ImgDst
)
```

## コメント

デバイスID  
 ソース画面番号(RGBカラー画面)  
 デスティネーション画面番号

## パラメータ

devID  
 デバイスID

ImgRGB  
 ソース画面番号(RGBカラー画面)

ImgDst  
 デスティネーション画面番号

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	R G B カラー画面番号に対して R G B 画面以外の画面番号を指定した
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
5 1	ワーク画面確保エラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

- ・デスティネーション画面に Y U V / R G B カラー画面を指定した場合、U V / G , B 画面は処理されません。
- ・IP\_ConvertSaturation()では、ワーク画面を 5 枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー ( エラーコード : 5 1 ) となります。この場合は、FreeImg() で不要な画面を解放して、空き画面を確保してください。

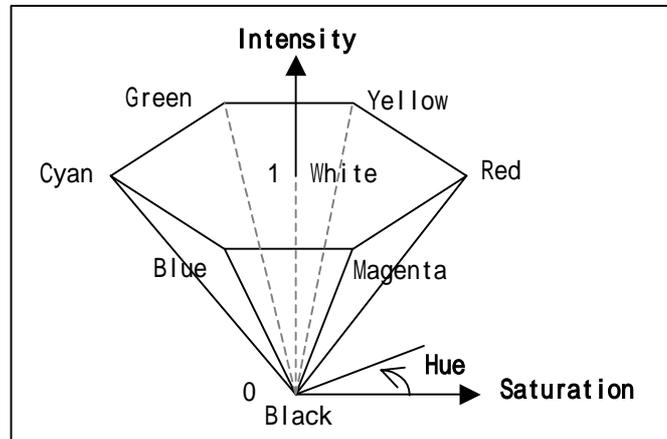
# IP\_ConvertIntensity

明度変換

## 機能

パラメータで指定されたRGBカラー画面をIntensity（明度）画像に変換します。

演算式	$I = \max(R, G, B)$
-----	---------------------



## コーディング

```
int
IP_ConvertIntensity(
    DEVID  devID ,
    int    ImgRGB ,
    int    ImgDst
)
```

## コメント

デバイスID  
 ソース画面番号(RGBカラー画面)  
 デスティネーション画面番号

## パラメータ

devID  
 デバイスID

ImgRGB  
 ソース画面番号(RGBカラー画面)

ImgDst  
 デスティネーション画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	画面空き領域、または範囲外（不当画面番号エラー）
14	RGBカラー画面番号に対してRGB画面以外の画面番号を指定した
49	アクセス画面が不定画面
50	画像メモリチャネル競合
51	ワーク画面確保エラー
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

- ・デスティネーション画面にYUV/RGBカラー画面を指定した場合、UV/G,B画面は処理されません。
- ・IP\_ConvertIntensity()では、ワーク画面を1枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー（エラーコード：51）となります。この場合は、FreeImg()で不要な画面を解放して、空き画面を確保してください。

## IP\_ConvertRGBtoYUV

## カラー疑似変換 (RGB YUV)

### 機能

R G B 画像を Y U V 画像に変換します。

RGB YUV 変換式	$Y = 0.299 * R + 0.587 * G + 0.144 * B$ $U = 0.72955 * (R - Y) + 128$ $V = 0.57722 * (B - Y) + 128$
----------------	---

### コーディング

```
int
IP_ConvertRGBtoYUV(
    DEVID    devID ,
    int      ImgSrcR ,
    int      ImgSrcG ,
    int      ImgSrcB ,
    int      ImgYUV
)
```

### コメント

デバイス I D  
ソース画面番号 ( R 画面 )  
ソース画面番号 ( G 画面 )  
ソース画面番号 ( B 画面 )  
デスティネーション画面番号

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**ImgSrcR**  
ソース画面番号 ( R 画面 )

**ImgSrcG**  
ソース画面番号 ( G 画面 )

**ImgSrcB**  
ソース画面番号 ( B 画面 )

**ImgYUV**  
デスティネーション画面番号 ( Y U V 画面 )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	カラー画面番号に対して Y U V 画面以外の画面番号を指定した
3 0	S R C 0 ウィンドウ設定エラー
3 3	D S T ウィンドウ設定エラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

### 詳細情報

- ・ソース画面にカラー画面を指定した場合、U V 画面は処理されません。
- ・画像メモリへのカラー画像書き込みで、X 方向の終点座標 ( クリッピング後 ) が偶数の場合は、X 方向終点座標を 1 画素内側に移動させます。そのため、ユーザーが指定した X 方向終点座標の 1 ラインは不定となります

## IP\_ConvertRGBtoYUVfast

カラー疑似変換 (RGB YUV)

### 機能

R G B 画像を Y U V 画像に変換します。

### コーディング

```
int
IP_ConvertRGBtoYUVfast(
    DEVID    devID ,
    int      ImgRGB ,
    int      ImgYUV
)
```

### コメント

デバイス I D  
ソース画面番号  
デスティネーション画面番号

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**ImgRGB**  
ソース画面番号 ( R G B 画面 )

**ImgYUV**  
デスティネーション画面番号 ( Y U V 画面 )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 1	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
-	エラーリセットコマンド ( ClearIPError ) 未発行

### 詳細情報

IP\_ConvertRGBtoYUVfast() では、処理の高速化を図るために、変換をハードウェアで計算を行っています。そのため、誤差が大きく生じますのでご注意ください。

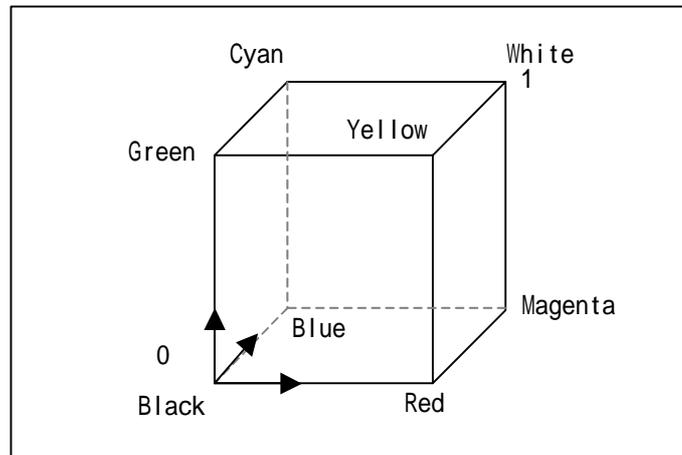
# IP\_ExtractColorRGB

## RGBカラー抽出処理

### 機能

下記条件の色をした部分を抽出し、2値画像としてモノクロ画面に転送します。

(Rthrmin R Rthrmax) & (Gthrmin G Gthrmax) & (Bthrmin B Bthrmax)



### コーディング

```
int
IP_ExtractColorRGB(
    DEVID  devID ,
    int    ImgRGB ,
    int    ImgDst ,
    int    Rthrmin ,
    int    Rthrmax ,
    int    Gthrmin ,
    int    Gthrmax ,
    int    Bthrmin ,
    int    Bthrmax
)
```

### コメント

デバイスID  
 ソース画面番号(RGBカラー画面)  
 デスティネーション画面番号  
 Red(赤)画像の最小しきい値  
 Red(赤)画像の最大しきい値  
 Green(緑)画像の最小しきい値  
 Green(緑)画像の最大しきい値  
 Blue(青)画像の最小しきい値  
 Blue(青)画像の最大しきい値

### パラメータ

**devID**  
 デバイスID

**ImgRGB**  
 ソース画面番号(RGBカラー画面)

**ImgDst**  
 デスティネーション画面番号

**Rthrmin**  
 Red(赤)画像の最小しきい値(0~255)

**Rthrmax**  
 Red(赤)画像の最大しきい値(0~255)

**Gthrmin**  
 Green(緑)画像の最小しきい値(0~255)

**Gthrmax**  
 Green(緑)画像の最大しきい値(0~255)

**Bthrmin**  
 Blue(青)画像の最小しきい値(0~255)

**Bthrmax**  
 Blue(青)画像の最大しきい値(0~255)

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	R G B カラー画面番号に対して R G B 画面以外の画面番号を指定した
2 2	R e d ( 赤 ) の最小閾値設定範囲外
2 3	R e d ( 赤 ) の最大閾値設定範囲外
2 4	G r e e n ( 緑 ) の最小閾値設定範囲外
2 5	G r e e n ( 緑 ) の最大閾値設定範囲外
2 6	B l u e ( 青 ) の最小閾値設定範囲外
2 7	B l u e ( 青 ) の最大閾値設定範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
5 1	ワーク画面確保エラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

- ・デスティネーション画面に Y U V / R G B カラー画面を指定した場合、U V / G , B 画面は処理されません。
- ・IP\_ExtractColorRGB() では、ワーク画面を 1 枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー ( エラーコード : 5 1 ) となります。この場合は、FreeImg() で不要な画面を解放して、空き画面を確保してください。

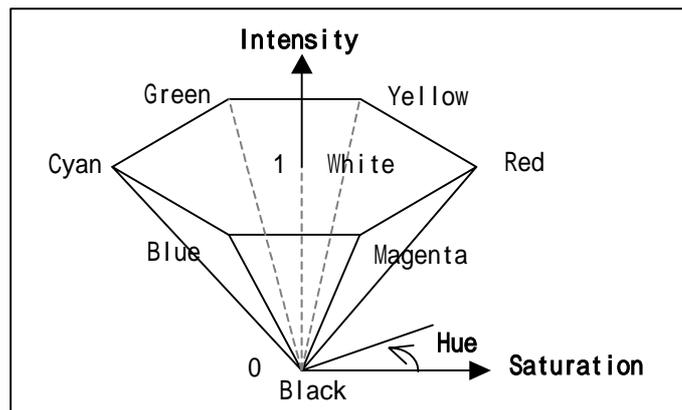
**IP\_ExtractColorHSI****カラー抽出処理 - 色相・彩度・明度変換****機能**

パラメータで指定されたRGBカラー画面をHue（色相）画像とSaturation（彩度）とIntensity（明度）画像に変換します。また下記条件の色をした部分を抽出し、2値画像としてモノクロ画面に転送します。

( Hthrmin H Hthrmax ) & ( Sthrmin S Sthrmax ) & ( Ithrmin I Ithrmax )

演算式	$I = \max( R, G, B )$ $I = 0: S = 0: H = 0(\text{不定})$ $I \neq 0: ( i = \min( R, G, B ) )$ $S = ( I - i ) * 255 / I$ $R \neq I: H = ( G - B ) * 60 / ( I - i )$ $G \neq I: H = ( B - R ) * 60 / ( I - i ) + 120$ $B \neq I: H = ( R - G ) * 60 / ( I - i ) + 240$
-----	---

( H < 0 の時は、H = H + 360 とする )

**コーディング**

```
int
IP_ExtractColorHSI(
    DEVID    devID ,
    int      ImgSrc ,
    int      ImgDst ,
    int      Hthrmin ,
    int      Hthrmax ,
    int      Sthrmin ,
    int      Sthrmax ,
    int      Ithrmin ,
    int      Ithrmax
)
```

**コメント**

デバイスID  
 ソース画面番号(RGBカラー画面)  
 デスティネーション画面番号  
 Hue（色相）画像の最小しきい値  
 Hue（色相）画像の最大しきい値  
 Saturation（彩度）画像の最小しきい値  
 Saturation（彩度）画像の最大しきい値  
 Intensity（明度）画像の最小しきい値  
 Intensity（明度）画像の最大しきい値

### パラメータ

devID	デバイスID
ImgRGB	ソース画面番号(RGBカラー画面)
ImgDst	デスティネーション画面番号
Hthrmin	Hue (色相) 画像の最小しきい値 (0 ~ 359)
Hthrmax	Hue (色相) 画像の最大しきい値 (0 ~ 359)
Sthrmin	Saturation (彩度) 画像の最小しきい値 (0 ~ 255)
Sthrmax	Saturation (彩度) 画像の最大しきい値 (0 ~ 255)
Ithrmin	Intensity (明度) 画像の最小しきい値 (0 ~ 255)
Ithrmax	Intensity (明度) 画像の最大しきい値 (0 ~ 255)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	画面空き領域、または範囲外 (不当画面番号エラー)
14	RGBカラー画面番号に対してRGB画面以外の画面番号を指定した
22	Hue (色相)の最小閾値設定範囲外
22	Hue (色相)の最大閾値設定範囲外
22	Saturation (彩度)の最小閾値設定範囲外
22	Saturation (彩度)の最大閾値設定範囲外
22	Intensity (明度)の最小閾値設定範囲外
22	Intensity (明度)の最大閾値設定範囲外
49	アクセス画面が不定画面
50	画像メモリチャネル競合
51	ワーク画面確保エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

- ・デスティネーション画面にYUV/RGBカラー画面を指定した場合、UV/G,B画面は処理されません。
- ・IP\_ExtractColorHSI()では、ワーク画面を9枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー(エラーコード:51)となります。この場合は、FreeImg()で不要な画面を解放して、空き画面を確保してください。

# OpenMultiColor

マルチカラー抽出処理可能

## 機能

マルチカラー抽出処理を可能状態にします。

## コーディング

```
int
OpenMultiColor(
    DEVID          devID ,
    enum IPMultiMethod method
)
```

## コメント

デバイスID  
マルチカラー抽出手法

## パラメータ

devID  
デバイスID

method  
マルチカラー抽出手法

マルチカラー抽出手法	内容
IPMultiYRT	Y / / でのマルチカラー抽出オープン処理実行
IPMultiYUV	Y / U / Vでのマルチカラー抽出オープン処理実行
IPMultiHSI	H u e / S a t u r a t i o n / I n t e n s i t yでのマルチカラー抽出オープン処理実行
IPMultiRGB	R e d / G r e e n / B l u eでのマルチカラー抽出オープン処理実行

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	抽出手法設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

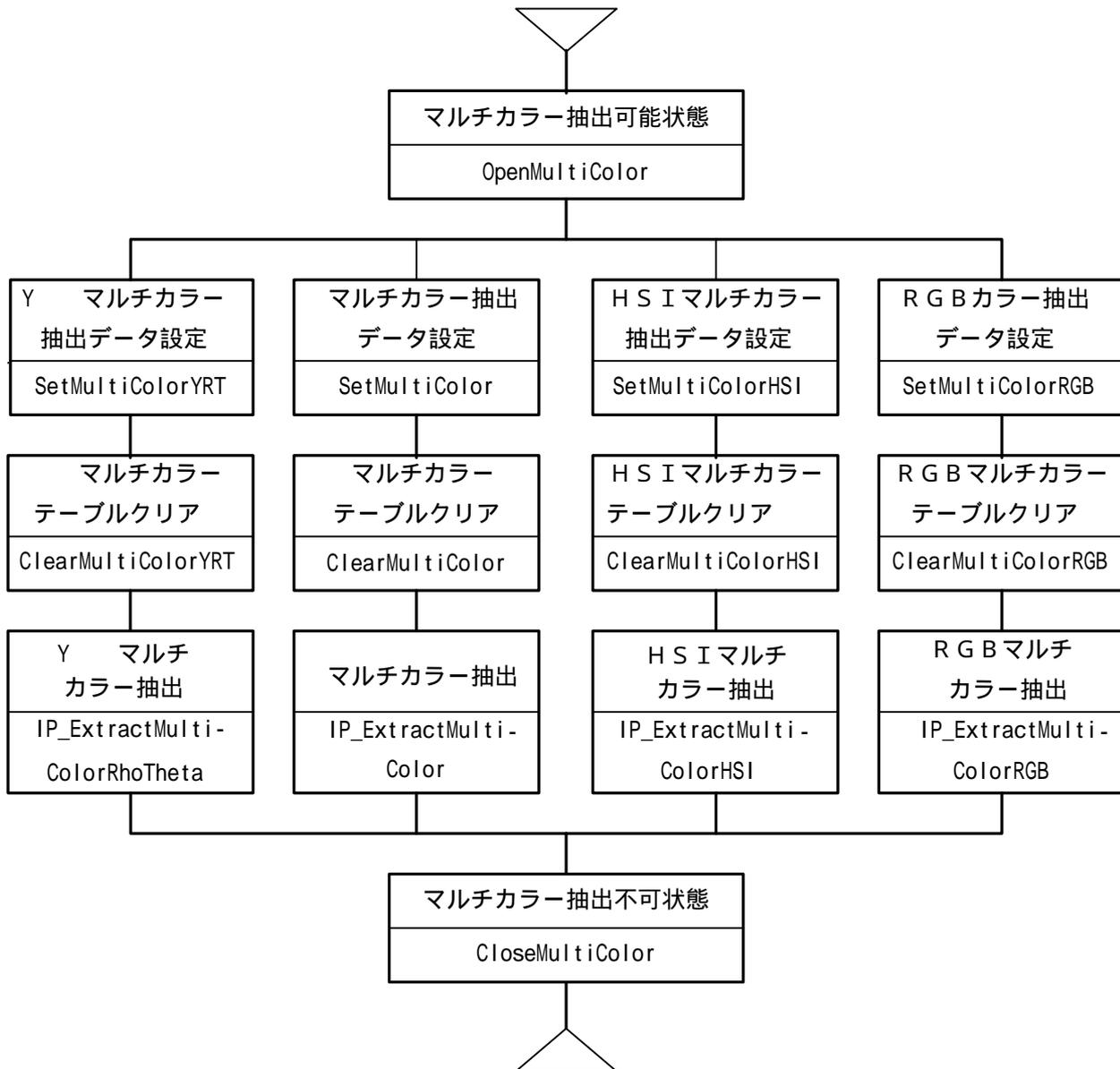
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

マルチカラー抽出処理を可能にします。マルチカラー抽出処理を実行する場合、予め本コマンドを実行して下さい。

尚、各マルチカラー抽出は同時に実行できます。それぞれ本コマンドを実行して下さい。

マルチカラー抽出を使用する場合には、下記手順に従ってください。  
各手順中のマルチカラーテーブルクリアは省略可能です。



# CloseMultiColor

マルチカラー処理不可

## 機能

マルチカラー抽出処理を不可状態にします。

## コーディング

```
int
CloseMultiColor(
    DEVID
    enum IPMultiMethod
)
```

devID ,  
method

## コメント

デバイスID  
マルチカラー抽出手法

## パラメータ

devID  
デバイスID  
method  
マルチカラー抽出手法

マルチカラー抽出手法	内容
IPMultiYRT	Y / / でのマルチカラー抽出クローズ処理実行
IPMultiYUV	Y / U / Vでのマルチカラー抽出クローズ処理実行
IPMultiHSI	H u e / Saturation / Intensityでのマルチカラー抽出クローズ処理実行
IPMultiRGB	R e d / G r e e n / B l u eでのマルチカラー抽出クローズ処理実行

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	抽出手法設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

マルチカラー抽出処理の各手法を不可状態にします。マルチカラー抽出処理を実行した後で、本コマンドを実行して下さい。

## ClearMultiColorHSI

## HSIマルチカラーテーブルクリア

### 機能

指定の H S I マルチカラー抽出データをクリアします。

### コーディング

```
int
ClearMultiColorHSI(
    DEVID  devID ,
    int    colorNum
)
```

### コメント

デバイスID  
変換色番号 ( 0 ~ 8 )

### パラメータ

**devID**  
デバイスID

**colorNum**  
変換色番号 ( 0 ~ 8 )  
( 0 : 全マルチカラーテーブルをクリアします )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	変換色番号設定範囲外
4 1	使用手続きエラー
-	エラーリセットコマンド ( ClearIPErrror ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

H S I マルチカラー抽出データ設定 SetMultiColorHSI() コマンドで、H S I マルチカラー抽出用のデータを設定します。設定したデータの一部、又は全てをクリアする時は本コマンドを実行して下さい。

## SetMultiColorHSI

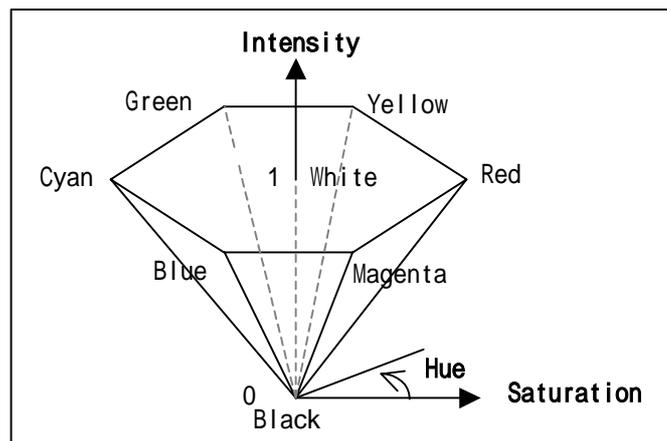
## HSIマルチカラー抽出データ設定

## 機能

HSIマルチカラー抽出・出力用のデータを設定します。パラメータで指定された範囲のHue(色相)データとSaturation(彩度)データとIntensity(明度)データを設定します。また、色ラベル画像出力時の出力番号を設定します。

演算式	$I = \max( R, G, B )$ $I = 0: S = 0: H = 0(\text{不定})$ $I = 0: ( i = \min( R, G, B ) )$ $S = ( I - i ) * 255 / I$ $R = I: H = ( G - B ) * 60 / ( I - i )$ $G = I: H = ( B - R ) * 60 / ( I - i ) + 120$ $B = I: H = ( R - G ) * 60 / ( I - i ) + 240$
-----	---

( $H < 0$  の時は、 $H = H + 360$  とする)



## コーディング

```
int
SetMultiColorHSI(
    DEVID  debID ,
    int    colorNum ,
    int    OutNum ,
    int    Hthrmin ,
    int    Hthrmax ,
    int    Sthrmin ,
    int    Saturation ,
    int    Ithrmin ,
    int    Intensity
)
```

## コメント

デバイスID  
 変換色番号 ( 1 ~ 8 )  
 出力番号 ( 1 ~ 255 )  
 Hue (色相) 画像の最小しきい値  
 Hue (色相) 画像の最大しきい値  
 Saturation (彩度) 画像の最小しきい値  
 Saturation (彩度) 画像の最大しきい値  
 Intensity (明度) 画像の最小しきい値  
 Intensity (明度) 画像の最大しきい値

**パラメータ**

devID	デバイスID
colorNum	変換色番号 ( 1 ~ 8 )
OutNum	出力番号 ( 1 ~ 255 )
Hthrmin	Hue (色相) 画像の最小しきい値 ( 0 ~ 360 )
Hthrmax	Hue (色相) 画像の最大しきい値 ( 0 ~ 360 )
Sthrmin	Saturation (彩度) 画像の最小しきい値 ( 0 ~ 255 )
Sthrmax	Saturation (彩度) 画像の最大しきい値 ( 0 ~ 255 )
Ithrmin	Intensity (明度) 画像の最小しきい値 ( 0 ~ 255 )
Ithrmax	Intensity (明度) 画像の最大しきい値 ( 0 ~ 255 )

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	変換色番号設定範囲外
2 1	出力番号設定範囲外
2 2	Hue (色相) の最小しきい値設定範囲外
2 3	Hue (色相) の最大しきい値設定範囲外
2 4	Saturation (彩度) の最小しきい値設定範囲外
2 5	Saturation (彩度) の最大しきい値設定範囲外
2 6	Intensity (明度) の最小しきい値設定範囲外
2 7	Intensity (明度) の最大しきい値設定範囲外
4 1	使用手続きエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

詳細はマルチカラー抽出 IP\_ExtractMultiColor() を参照して下さい。

# IP\_ExtractMultiColorHSI

## HSIマルチカラー抽出

### 機能

HSIマルチカラーテーブルに設定した色の部分を抽出し、オプションにより2値画像か色ラベル画像としてモノクロ画面に転送します。

### コーディング

```
int
IP_ExtractMultiColorHSI(
    DEVID          devID ,
    int            ImgRGB ,
    int            ImgDst ,
    enum MultiColorOpt MultiOpt
)
```

### コメント

デバイスID  
ソース画面番号(RGBカラー画面)  
デスティネーション画面番号  
マルチカラー出力オプション

### パラメータ

**devID**  
デバイスID

**ImgRGB**  
ソース画面番号(RGBカラー画面)

**ImgDst**  
デスティネーション画面番号

**MultiOpt**  
マルチカラー出力オプション

マルチカラー出力オプション	内容
MULTICOLOR_LABEL	抽出した色毎に番号付けして出力します。デスティネーション画像はUNSIGN8画像になります。
MULTICOLOR_BIN	抽出した色を2値化して出力します。デスティネーション画像は2値画像になります。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

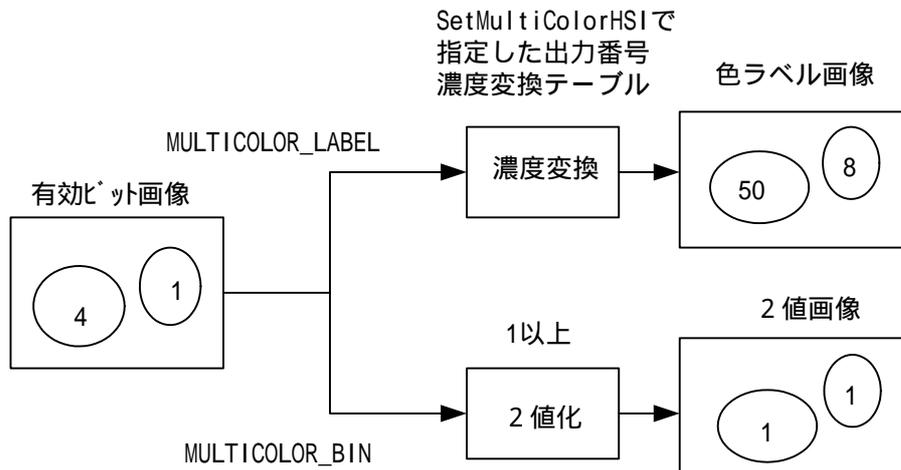
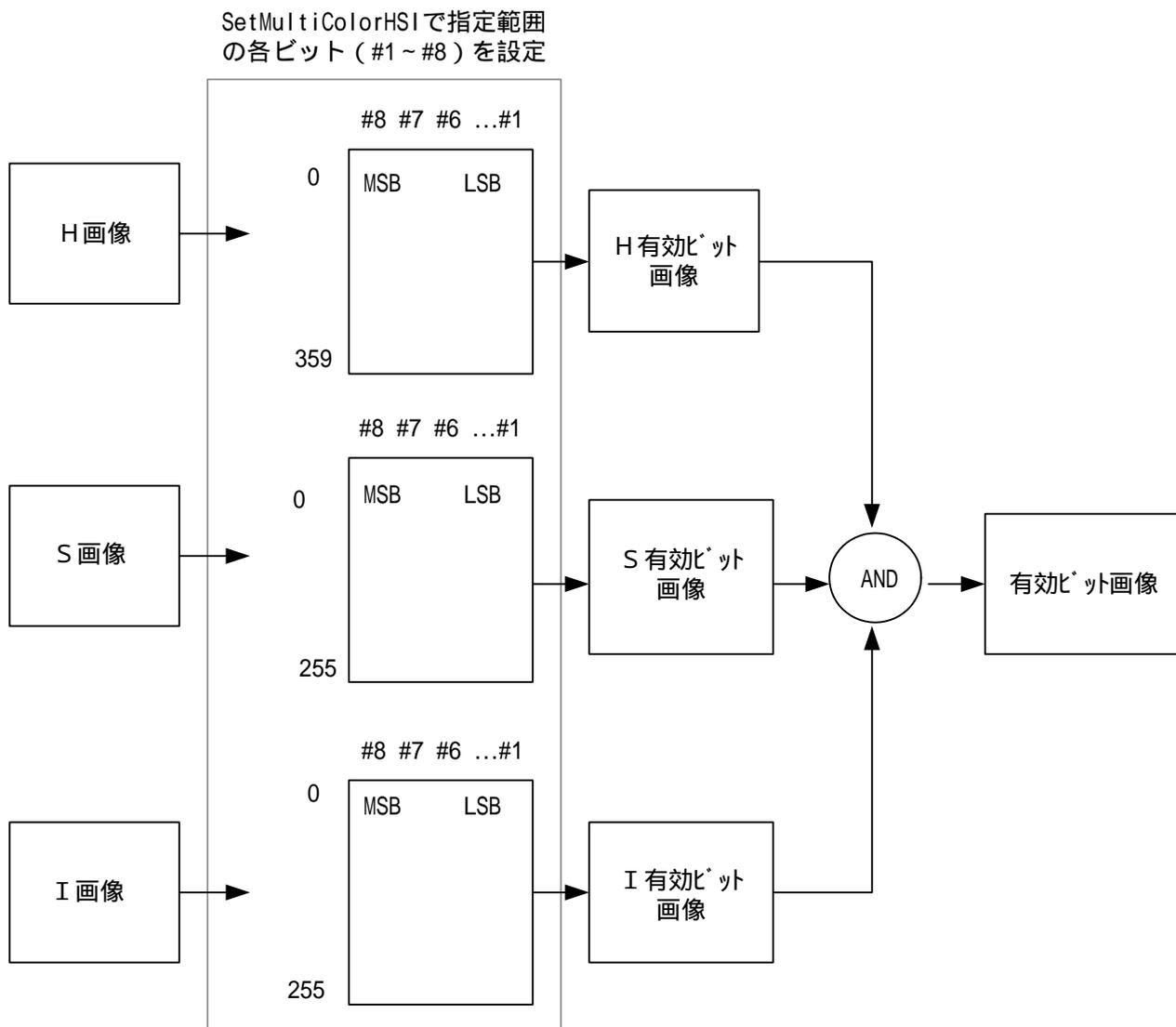
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
14	RGBカラー画面番号に対してRGB画面以外の画面番号を指定した
22	マルチカラー出力オプション設定範囲外
41	使用手続きエラー
49	アクセス画面が不定画面
50	画像メモリチャネル競合
51	ワーク画面確保エラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

- IP\_ExtractMultiColorHSI()では、ワーク画面を9枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー（エラーコード：51）となります。この場合は、FreeImg()で不要な画面を解放して、空き画面を確保してください。
- マルチカラー出力オプションで MULTICOLOR\_LABEL 設定時、SetMultiColorHSI()で設定した出力番号を抽出した色領域に出力します。これを色ラベル画像と呼びます。H S I の範囲が重なる範囲を設定した場合、変換色番号の小さい方を優先して出力します。MULTICOLOR\_BIN 設定時は、色ラベル画像を"1"で2値化した画像を出力します。



## ClearMultiColorRGB

RGBマルチカラーテーブルクリア

### 機能

指定RGBマルチカラー抽出データをクリアします。

### コーディング

```
int
ClearMultiColorRGB(
    DEVID  devID ,
    int    colorNum
)
```

### コメント

デバイスID  
変換色番号(0~8)

### パラメータ

devID  
デバイスID

colorNum  
変換色番号(0~8)  
(0:全マルチカラーテーブルをクリアします)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	変換色番号設定範囲外
41	使用手続きエラー
-	エラーリセットコマンド(ClearIPErr)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

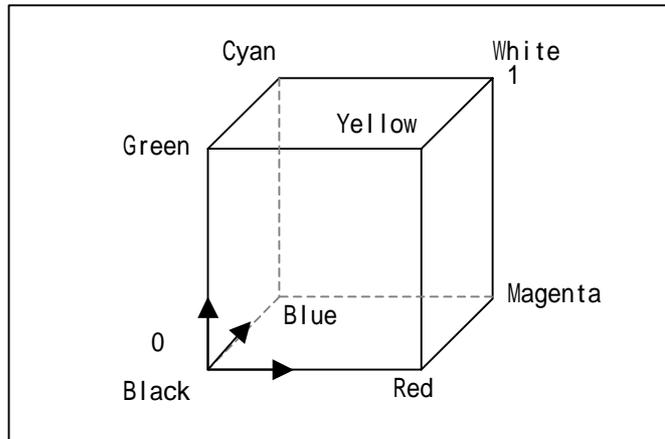
RGBマルチカラー抽出データ設定 - 色彩距離・色相変換 SetMultiColorRGB()コマンドで、RGBマルチカラー抽出用のデータを設定します。設定したデータの一部、又は全てをクリアする時は本コマンドを実行して下さい。

## SetMultiColorRGB

## RGBマルチカラー抽出データ設定

### 機能

RGBマルチカラー抽出・出力用のデータを設定します。パラメータで指定された範囲のRed（赤）データとGreen（緑）データとBlue（青）データを設定します。また、色ラベル画像出力時の出力番号を設定します。



### コーディング

```
int
SetMultiColorRGB(
    DEVID  devID ,
    int    colorNum ,
    int    OutNum ,
    int    Rthrmin ,
    int    Rthrmax ,
    int    Gthrmin ,
    int    Gthrmax ,
    int    Bthrmin ,
    int    Bthrmax
)
```

### コメント

デバイスID  
 変換色番号（1～8）  
 出力番号（1～255）  
 Red（赤）画像の最小しきい値  
 Red（赤）画像の最大しきい値  
 Green（緑）画像の最小しきい値  
 Green（緑）画像の最大しきい値  
 Blue（青）画像の最小しきい値  
 Blue（青）画像の最大しきい値

### パラメータ

devID  
 デバイスID  
 colorNum  
 変換色番号（1～8）  
 OutNum  
 出力番号（1～255）  
 Rthrmin  
 Red（赤）画像の最小しきい値（0～255）  
 Rthrmax  
 Red（赤）画像の最大しきい値（0～255）  
 Gthrmin  
 Green（緑）画像の最小しきい値（0～255）  
 Gthrmax  
 Green（緑）画像の最大しきい値（0～255）  
 Bthrmin  
 Blue（青）画像の最小しきい値（0～255）  
 Bthrmax  
 Blue（青）画像の最大しきい値（0～255）

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	変換番号設定範囲外
2 1	出力番号設定範囲外
2 2	Red (赤) の最小しきい値設定範囲外
2 3	Red (赤) の最大しきい値設定範囲外
2 4	Green (緑) の最小しきい値設定範囲外
2 5	Green (緑) の最大しきい値設定範囲外
2 6	Blue (青) の最小しきい値設定範囲外
2 7	Blue (青) の最大しきい値設定範囲外
4 1	使用手続きエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

詳細はマルチカラー抽出 IP\_ExtractMultiColorRGB() を参照して下さい。

# IP\_ExtractMultiColorRGB

## RGBマルチカラー抽出

### 機能

RGBマルチカラーテーブルに設定した色の部分を抽出し、オプションにより2値画像か色ラベル画像としてモノクロ画面に転送します。

### コーディング

```
int
IP_ExtractMultiColorRGB(
    DEVID          devID ,
    int            ImgRGB ,
    int            ImgDst ,
    enum MultiColorOpt MultiOpt
)
```

### コメント

デバイスID  
ソース画面番号(RGBカラー画面)  
デスティネーション画面番号  
マルチカラー出力オプション

### パラメータ

**devID**  
デバイスID

**ImgRGB**  
ソース画面番号(RGBカラー画面)

**ImgDst**  
デスティネーション画面番号

**MultiOpt**  
マルチカラー出力オプション

マルチカラー出力オプション	内容
MULTICOLOR_LABEL	抽出した色を2値化して出力します。デスティネーション画像はUNSIGN8画像になります。
MULTICOLOR_BIN	抽出した色を2値化して出力します。デスティネーション画像は2値画像になります。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

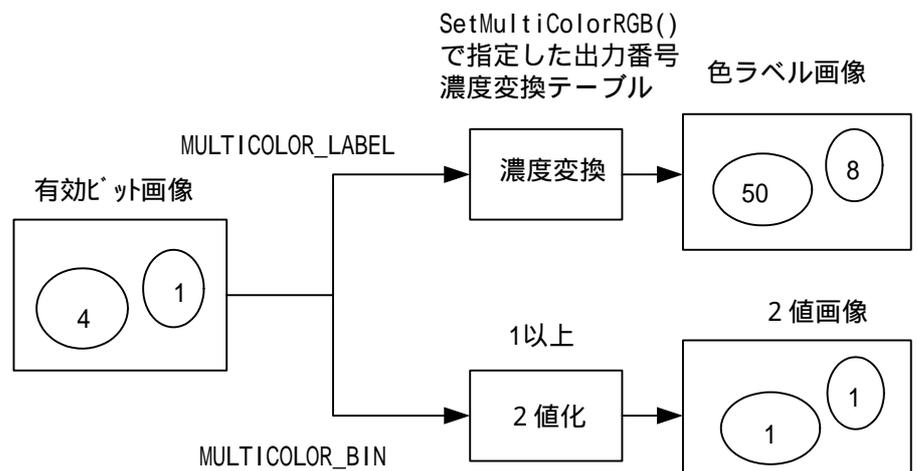
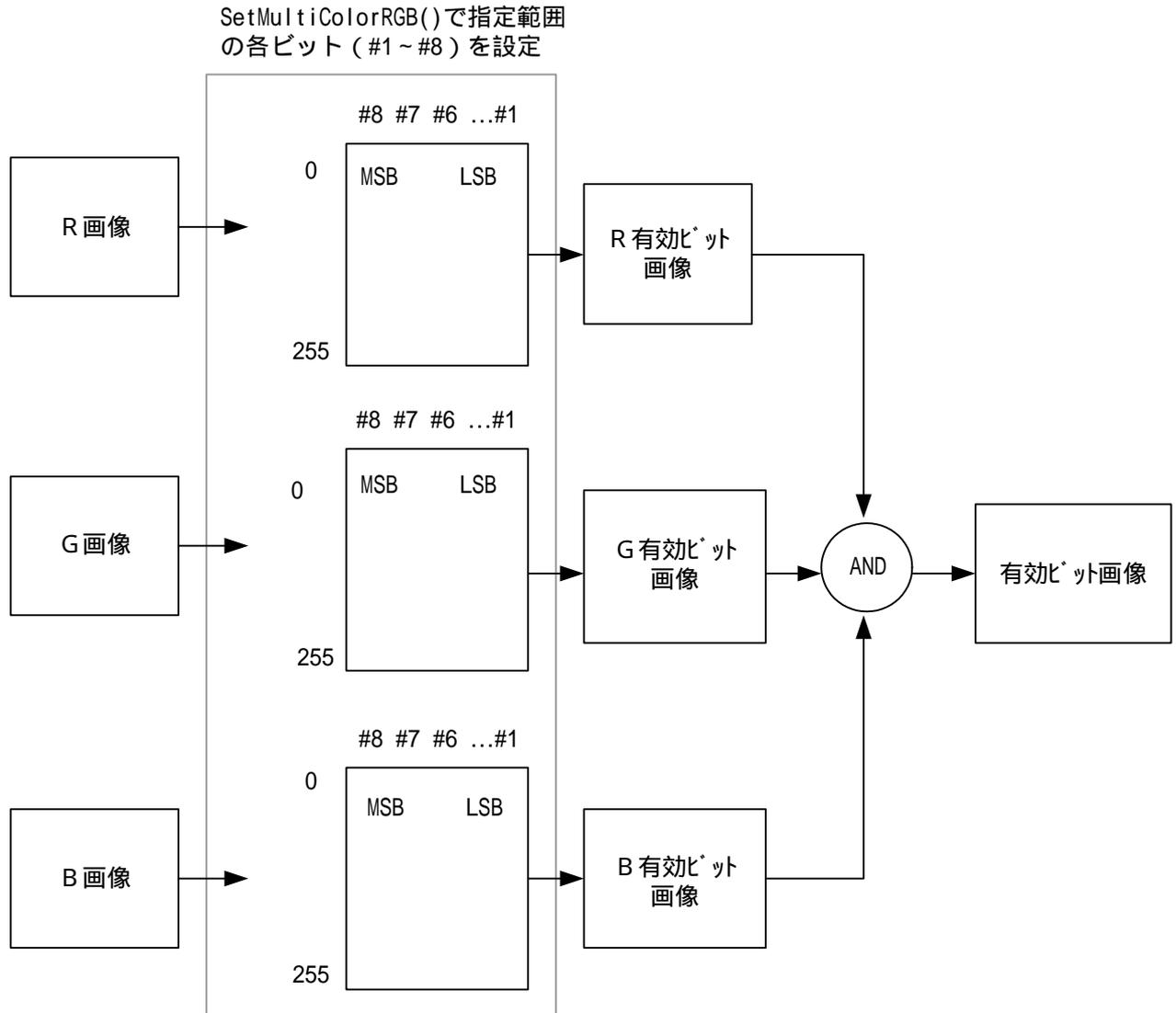
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
14	RGBカラー画面番号に対してRGB画面以外の画面番号を指定した
22	マルチカラー出力オプション設定範囲外
41	使用手続きエラー
49	アクセス画面が不定画面
50	画像メモリチャネル競合
51	ワーク画面確保エラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

- IP\_ExtractMultiColorRGB()では、ワーク画面を1枚使用します。画像メモリの空きがなく、ワーク画面が確保できない場合に、エラー（エラーコード：51）となります。この場合は、FreeImg()で不要な画面を解放して、空き画面を確保してください。
- マルチカラー出力オプションでMULTICOLOR\_LABEL設定時、SetMultiColorRGB()で設定した出力番号を抽出した色領域に出力します。これを色ラベル画像と呼びます。RGBの範囲が重なる範囲を設定した場合、変換番号の小さい方を優先して出力します。MULTICOLOR\_BIN設定時は、色ラベル画像を"1"で2値化した画像を出力します。



# SetBinMatchTemplate

## 2値画像マッチングフィルタテンプレート登録

### 機能

2値画像マッチングフィルタのテンプレート登録を行います。

SVP - 330では、ハードウェアの制約により「2値画像マッチングフィルタコマンド」は、サポートされていません。

### コーディング

```
int
SetBinMatchTemplate(
    DEVID      devID ,
    int        *Temp ,
    int        TmpID
)
```

### コメント

デバイスID  
登録テーブルのアドレス  
テンプレート番号

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*Temp

テンプレート登録テーブルのアドレス

TmpID

テンプレート番号 ( 0 ~ 63 )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
8 1	テンプレート画面の不当画面番号エラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

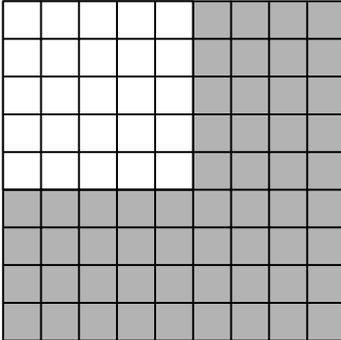
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

テンプレートのサイズは9 × 9画素です。その設定は、Temp[Y][X] (C言語) で記述してください。  
また、テンプレートは9 × 9画素を正しく設定してください。正しく設定しない場合は、動作保証をいたしません。

<記述例>

テンプレート(9 × 9画素)



テンプレート登録テーブル設定

```
Temp[9][9]={ { 0, 0, 0, 0, 0, 1, 1, 1, 1 },
              { 0, 0, 0, 0, 0, 1, 1, 1, 1 },
              { 0, 0, 0, 0, 0, 1, 1, 1, 1 },
              { 0, 0, 0, 0, 0, 1, 1, 1, 1 },
              { 0, 0, 0, 0, 0, 1, 1, 1, 1 },
              { 1, 1, 1, 1, 1, 1, 1, 1, 1 },
              { 1, 1, 1, 1, 1, 1, 1, 1, 1 },
              { 1, 1, 1, 1, 1, 1, 1, 1, 1 },
              { 1, 1, 1, 1, 1, 1, 1, 1, 1 }
            };
```

## IP\_BinMatchFLT

## 2値画像マッチングフィルタ実行

### 機能

2値画像マッチングフィルタの実行を行います。

SVP-330では、ハードウェアの制約により「2値画像マッチングフィルタコマンド」は、サポートされていません。

### コーディング

```
int
IP_BinMatchFLT(
    DEVID
    int
    int
    int
    enum IPBinMatchMode
)
```

```
devID ,
ImgSrc ,
ImgDst ,
TmpID ,
mode
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
テンプレート管理番号
マッチング処理オプション
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面番号

#### ImgDst

デスティネーション画面番号

#### TmpID

テンプレート管理番号

#### mode

マッチング処理オプション

マッチング処理オプション	対応定数	処理内容
MATCH_AND	0	AND処理
MATCH_XNOR	1	XNOR処理

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

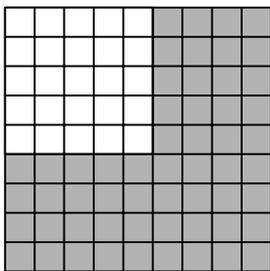
エラーコード	エラー原因
10、13	指定画面空き領域、または範囲外（不当画面番号エラー）
23	オプション設定値範囲外
49	アクセス画面が不定画面
81	テンプレート画面の不当画面番号エラー
87	テンプレート未設定エラー
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

2値画像マッチングフィルタでは9×9画素のテンプレートにより入力画面（2値）のマッチング処理を行い、結果を濃淡画面に出力します。

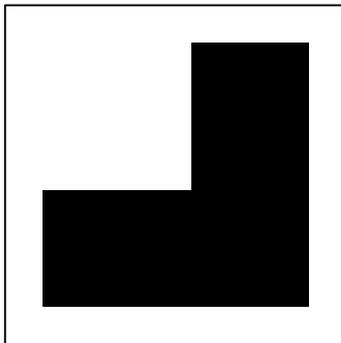
テンプレート(9×9 画素)



テンプレート登録

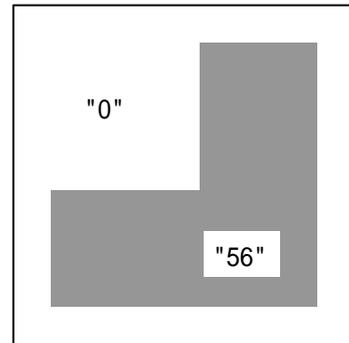
→ SetBinMatchTemplate を参照して下さい。

入力画像 ( ImgSrc )

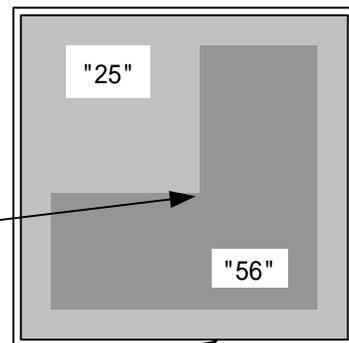


マッチング処理オプション  
MATCH\_AND

出力画像 ( ImgDst )



マッチング処理オプション  
MATCH\_XNOR



周辺4画素は"0"が入ります







# IP\_BinarizePTM9x9

## 2値画像マッチングフィルタ(9×9)

### 機能

2値画像マッチングフィルタ(9×9)を実行し、処理画像に対してパラメータのしきい値で2値化を行います。

SVP-330では、ハードウェアの制約により「2値画像マッチングフィルタコマンド」は、サポートされていません。

### コーディング

```
int
IP_BinarizePTM9x9(
  DEVID
  int
  int
  int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thr
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
2値化しきい値
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面番号

ImgDst

デスティネーション画面番号

thr

2値化しきい値(1~61)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	閾値設定範囲外
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

テンプレートサイズは9×9で、以下のように設定されています。

0	0	0	1	1	1	0	0	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	0	0	0





## IP\_BinarizePTM7x9

## 2値画像マッチングフィルタ(7×9)

### 機能

2値画像マッチングフィルタ(7×9)を実行し、処理画像に対してパラメータのしきい値で2値化を行います。

SVP-330では、ハードウェアの制約により「2値画像マッチングフィルタコマンド」は、サポートされていません。

### コーディング

```
int
IP_BinarizePTM7x9(
    DEVID
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thr
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
2値化しきい値
```

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面番号

ImgDst

デスティネーション画面番号

thr

2値化しきい値(1~47)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	閾値設定範囲外
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

テンプレートサイズは9×9で、以下のように設定されています。

0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	0	0	0

# LoadBMPFile

## BMPファイルのロード

### 機能

BMPファイルの画像データをNVP-935画像データに変換し、ImgIDで示される画面へロードします。

### コーディング

```
int
LoadBMPFile(
    DEVID  devID ,
    int    imgID ;
    char   *filename
)
```

### コメント

デバイスID  
画面番号  
ファイル名へのポインタ

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**imgID**  
転送先の画面番号

**\*filename**  
BMPファイルのファイル名(拡張子まで含む)へのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10	画面空き領域、または範囲外(不当画面番号エラー)
100	画像ファイルオープンエラー
101	不当画像ファイルエラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

下図に示す条件にてビットマップファイルの画像データを画像メモリにロードします。

BMP ファイル 画面	B/W (モノクロ) ビットマップ	YUV ビットマップ	RGB ビットマップ 256色ビットマップファイル 24ビットマップファイル
モノクロ	モノクロ画像をロードする	1面目をY画面のみロードする	YUV画像に変換してから1面目をY画面にロードする
YUVカラー	モノクロ画像をY画面のみロードする。UV画面は128でクリアする	1面目をY画面 / 2面目をUV画面にロードする	YUV画像に変換してから1面目をY画面 / 2面目をUV画面にロードする
RGBカラー	モノクロ画像をR、G、B画面すべてにロードする		RGB画像をRGBそれぞれの画面にロードする

画像メモリのデータをビットマップファイルに読み出す場合は以下のファイル形式を選択できます。

- ・ B/W (モノクロ) ビットマップ (ビットマップファイル準拠)  
Y画像 (8ビット階調) をそのままのイメージで保存する
- ・ YUV ビットマップ (IP-X 専用フォーマット)  
Y画像とUV画像 (各8ビット階調) をそのままのイメージで保存する
- ・ RGB ビットマップ (ビットマップファイル準拠)  
R、G、B成分 (各8ビット階調) に変換し保存する

注1) YUV ビットマップはIP-X専用フォーマットのため、ペイントブラシ等でファイルを開くことができません。

注2) カラー画像をRGB ビットマップで保存する場合、高輝度 / 高濃色部分でデータの階調が足りなくなる場合があります。この部分のデータはクリッピングされますので元の画像イメージと多少異なります。

B/W (モノクロ) ビットマップ、RGB ビットマップはWindows 標準画像ファイルとして使用できます。また、ファイル名の拡張子は.bmp を使用してください。画面内の転送領域はウインドウ (SYS\_WIN) で指定できます。ただし、データ転送領域のX方向始点座標を偶数、終点座標を奇数に設定する必要があります。

# SaveBMPFile

## BMPファイルのセーブ

### 機能

ImgID で示される画面内の画像データを BMP ファイルに変換しセーブします。

### コーディング

```
int
SaveBMPFile(
    DEVID
    int
    char
    enum BITMAP_MODE
)
```

```
devID ,
ImgID ,
*filename ,
mode
```

### コメント

```
デバイス I D
画面番号
ファイル名へのポインタ
BMP ファイル転送モード
```

### パラメータ

#### devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

#### ImgID

転送元の画面番号

#### \*filename

BMPファイルのファイル名（拡張子まで含む）へのポインタ

#### mode

BMPファイル転送モード

・下図に示す条件にて画像メモリの画像データをビットマップファイルにセーブします。

画面 \ mode	BW_BITMAP	RGB_BITMAP	YUV_BITMAP
モノクロ	Y画像のみセーブする	Y画像のみセーブする	Y画像のみセーブする
YUVカラー	Y画像のみセーブする	YUV画像をRGB画像に変換し画像をセーブする	Y画像を1面目 / UV画像を2面目にセーブする
RGBカラー	RGB画像をY画像に変換し画像をセーブする		RGB画像をYUV画像に変換しY画像を1面目 / UV画像を2面目にセーブする

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 2	転送モード指定範囲外
4 9	アクセス画面が不定画面画像
1 0 0	ファイルオープンエラー
-	エラーリセットコマンド (ClearIPErrors) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

画像メモリには以下のファイル形式のデータを書き込むことができます。

- ・ B/W (モノクロ) ビットマップ
- ・ R G B ビットマップ
- ・ Y U V ビットマップ
- ・ Windows 標準の 256色ビットマップファイル / 24ビットビットマップファイル

注1) 白黒 (1ビット) ビットマップ, 16色 (4ビット) ビットマップ画像についてはサポートしていません。

注2) 圧縮画像についてはサポートしていません。

注3) 256色ビットマップファイルをYUV画面に書き込んだ場合、色が正常に表示されません。カラービットマップファイルデータをYUV画面に書き込む場合は24ビットビットマップファイルを使用してください。

B/W (モノクロ) ビットマップ、R G B ビットマップはWindows 標準画像ファイルとして使用できません。また、ファイル名の拡張子は.bmp を使用してください。画面内の転送領域はウインドウ (SYS\_WIN) で指定できます。ただし、データ転送領域のX方向始点座標を偶数、終点座標を奇数に設定する必要があります。

ソース画面のデータタイプは保存されません。このため、画像メモリに書き込まれた映像データのデータタイプは転送先画面のデータタイプに依存します。データタイプの異なる画面を混在して使用する場合は ChangeImgDataType() で任意のデータタイプに設定してください。ファイルの画像データが画像メモリの転送領域より大きい場合はクリッピングされます。

**IP\_SmoothFLT5x5****平滑化(5x5近傍)****機能**

ソース画面（濃淡）に対し、5 × 5 近傍で平滑化を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

**コーディング**

```
int
IP_SmoothFLT5x5(
    DEVID
    int
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
scale ,
*COEFF
```

**コメント**

```
デバイスID
ソース画面番号
デスティネーション画面番号
ダウンシフト量
係数テーブルのポインタ
```

**パラメータ**

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量 ( 0 ~ 15 )

**\*COEFF**

荷重係数テーブルのポインタ

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

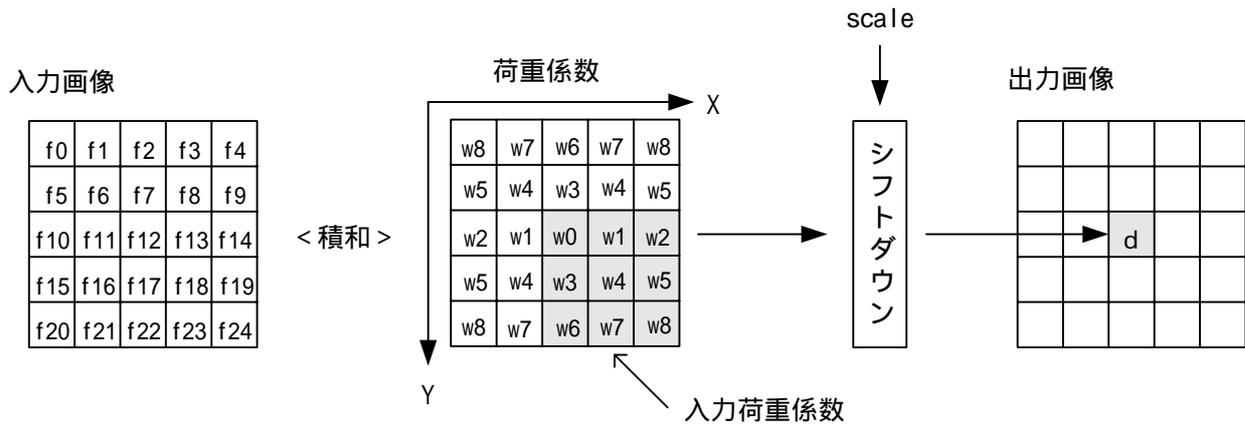
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2	ダウンシフト量範囲外
4 9	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

平滑化では、入力画面の指定領域内の全画素に対して、注目画素を中心とした5 × 5近傍の局所領域で与えられた荷重係数との積和演算を行います。



$$d = ( f_0 \times w_8 + f_1 \times w_7 + f_2 \times w_6 + f_3 \times w_7 + f_4 \times w_8 + f_5 \times w_5 + f_6 \times w_4 + f_7 \times w_3 + f_8 \times w_4 + f_9 \times w_5 + f_{10} \times w_2 + f_{11} \times w_1 + f_{12} \times w_0 + f_{13} \times w_1 + f_{14} \times w_2 + f_{15} \times w_5 + f_{16} \times w_4 + f_{17} \times w_3 + f_{18} \times w_4 + f_{19} \times w_5 + f_{20} \times w_8 + f_{21} \times w_7 + f_{22} \times w_6 + f_{23} \times w_7 + f_{24} \times w_8 ) \gg \text{scale}$$

係数テーブルのフォーマット及び荷重係数の例を、下記に示します。

荷重係数

#0	#1	#2
#3	#4	#5
#6	#7	#8

	31	8	7	0
COEFF [ 0 ]	_____		係数 # 0	
[ 1 ]	_____		# 1	
[ 2 ]	_____		# 2	
[ 3 ]	_____		# 3	
[ 4 ]	_____		# 4	
[ 5 ]	_____		# 5	
[ 6 ]	_____		# 6	
[ 7 ]	_____		▼ # 7	
[ 8 ]	_____		係数 # 8	

< 例1 >

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

< 例2 >

0	0	0	0	0
0	3	3	3	0
0	3	3	3	0
0	3	3	3	0
0	0	0	0	0

この例の場合、ダウンシフト量は5です。

## IP\_SmoothFLT7x7

## 平滑化(7×7近傍)

### 機能

ソース画面（濃淡）に対し、7×7近傍で平滑化を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_SmoothFLT7x7(
    DEVID
    int
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
scale ,
*COEFF
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
ダウンシフト量
係数テーブルのポインタ
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面の画面番号

#### ImgDst

デスティネーション画面の画面番号

#### scale

ダウンシフト量 ( 0 ~ 15 )

#### \*COEFF

荷重係数テーブルのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

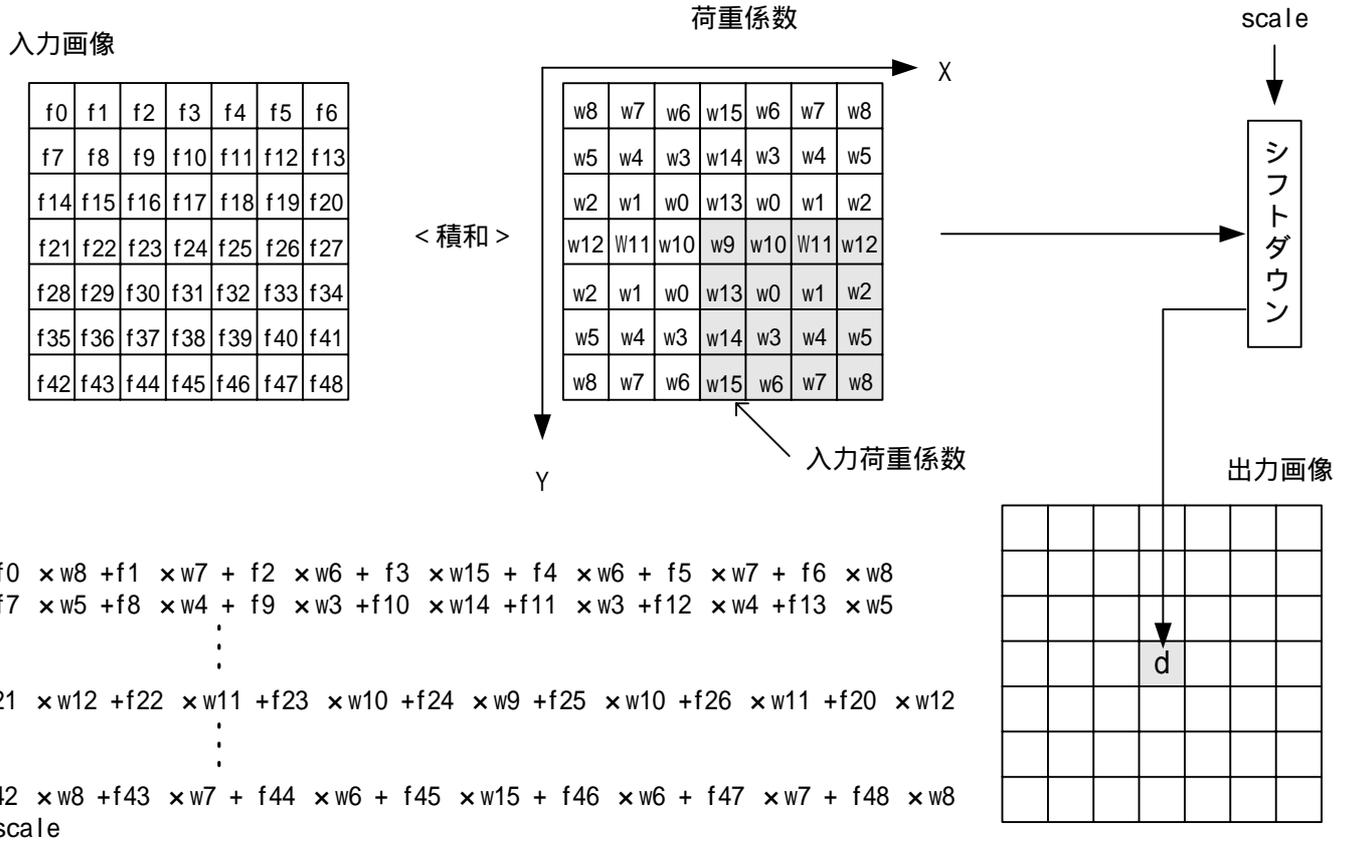
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10 , 13	画面空き領域、または範囲外 ( 不当画面番号エラー )
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

平滑化では、入力画面の指定領域内の全画素に対して、注目画素を中心とした7×7近傍の局所領域で与えられた荷重係数との積和演算を行います。



係数テーブルのフォーマット及び荷重係数の例を、下記に示します。

	31	8	7	0
COEFF [ 0 ]	_____			係数 # 0
[ 1 ]	_____			# 1
[ 2 ]	_____			# 2
[ 3 ]	_____			# 3
[ 4 ]	_____			# 4
[ 5 ]	_____			# 5
[ 6 ]	_____			# 6
[ 7 ]	_____			# 7
[ 8 ]	_____			# 8
[ 9 ]	_____			# 9
[ 10 ]	_____			# 10
[ 11 ]	_____			# 11
[ 12 ]	_____			# 12
[ 13 ]	_____			# 13
[ 14 ]	_____			# 14
[ 15 ]	_____			係数 # 15

荷重係数

#0	#1	#2	#3
#4	#5	#6	#7
#8	#9	#10	#11
#12	#13	#14	#15

< 例1 >

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

< 例2 >

0	0	0	0	0	0	0
0	2	2	2	2	2	0
0	2	2	2	2	2	0
0	2	2	2	2	2	0
0	2	2	2	2	2	0
0	2	2	2	2	2	0
0	0	0	0	0	0	0

この例の場合、ダウンシフト量は6です。

## IP\_EdgeFLT5x5

### 濃淡画像輪郭強調(5×5近傍)

#### 機能

ソース画面（濃淡）に対し、5×5近傍で輪郭強調を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_EdgeFLT5x5(
    DEVID
    int
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
scale ,
*COEFF
```

#### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
ダウンシフト量
係数テーブルのポインタ
```

#### パラメータ

##### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

##### ImgSrc

ソース画面の画面番号

##### ImgDst

デスティネーション画面の画面番号

##### scale

ダウンシフト量(0 ~ 15)

##### \*COEFF

荷重係数テーブルのポインタ

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

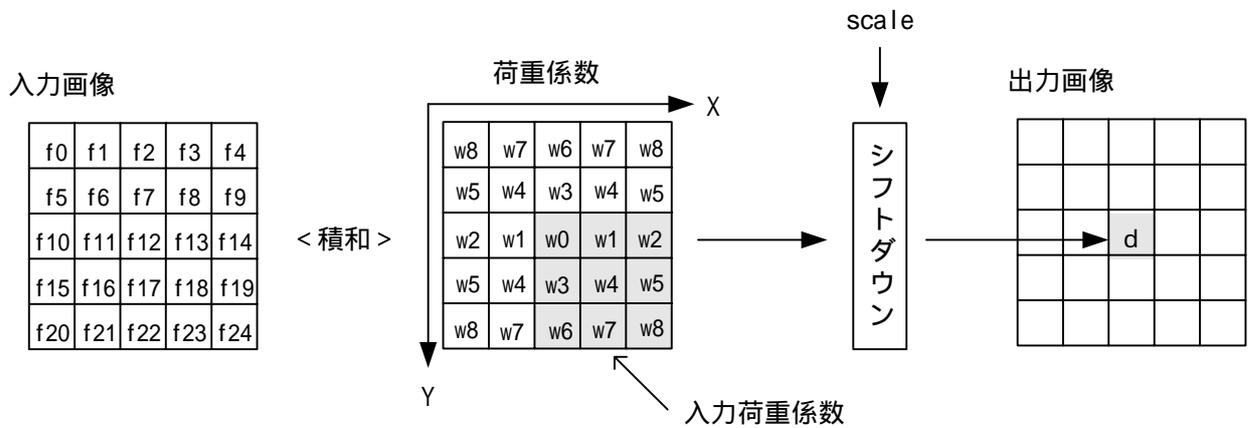
エラーコード	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

輪郭強調では、入力画面の指定領域内の全画素に対して、注目画素を中心とした5×5近傍の局所領域で与えられた荷重係数との積和演算を行います。

また、対象領域の周辺(5×5近傍に領域外が含まれる部分)処理の場合は0を出力します。



$$\begin{aligned}
 d = & ( f0 \times w8 + f1 \times w7 + f2 \times w6 + f3 \times w7 + f4 \times w8 \\
 & + f5 \times w5 + f6 \times w4 + f7 \times w3 + f8 \times w4 + f9 \times w5 \\
 & + f10 \times w2 + f11 \times w1 + f12 \times w0 + f13 \times w1 + f14 \times w2 \\
 & + f15 \times w5 + f16 \times w4 + f17 \times w3 + f18 \times w4 + f19 \times w5 \\
 & + f20 \times w8 + f21 \times w7 + f22 \times w6 + f23 \times w7 + f24 \times w8 \\
 & ) \gg \text{scale}
 \end{aligned}$$

荷重係数の例を、以下に示します。また、係数テーブルのフォーマットは、IP\_SmoothFLT5x5 の備考の欄を参照してください。

荷重係数の例

< 例1 > X方向微分

-1	1	0	1	-1
-1	1	0	1	-1
-1	1	0	1	-1
-1	1	0	1	-1
-1	1	0	1	-1

< 例2 > Y方向微分

-1	-1	-1	-1	-1
1	1	1	1	1
0	0	0	0	0
1	1	1	1	1
-1	-1	-1	-1	-1

この例の場合、ダウンシフト量は0です。

# IP\_EdgeFLT7x7

## 濃淡画像輪郭強調(7×7近傍)

### 機能

ソース画面（濃淡）に対し、7×7近傍で輪郭強調を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_EdgeFLT7x7(
    DEVID
    int
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
scale ,
*COEFF
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
ダウンシフト量
係数テーブルのポインタ
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画面の画面番号

**ImgDst**

デスティネーション画面の画面番号

**scale**

ダウンシフト量(0 ~ 15)

**\*COEFF**

荷重係数テーブルのポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

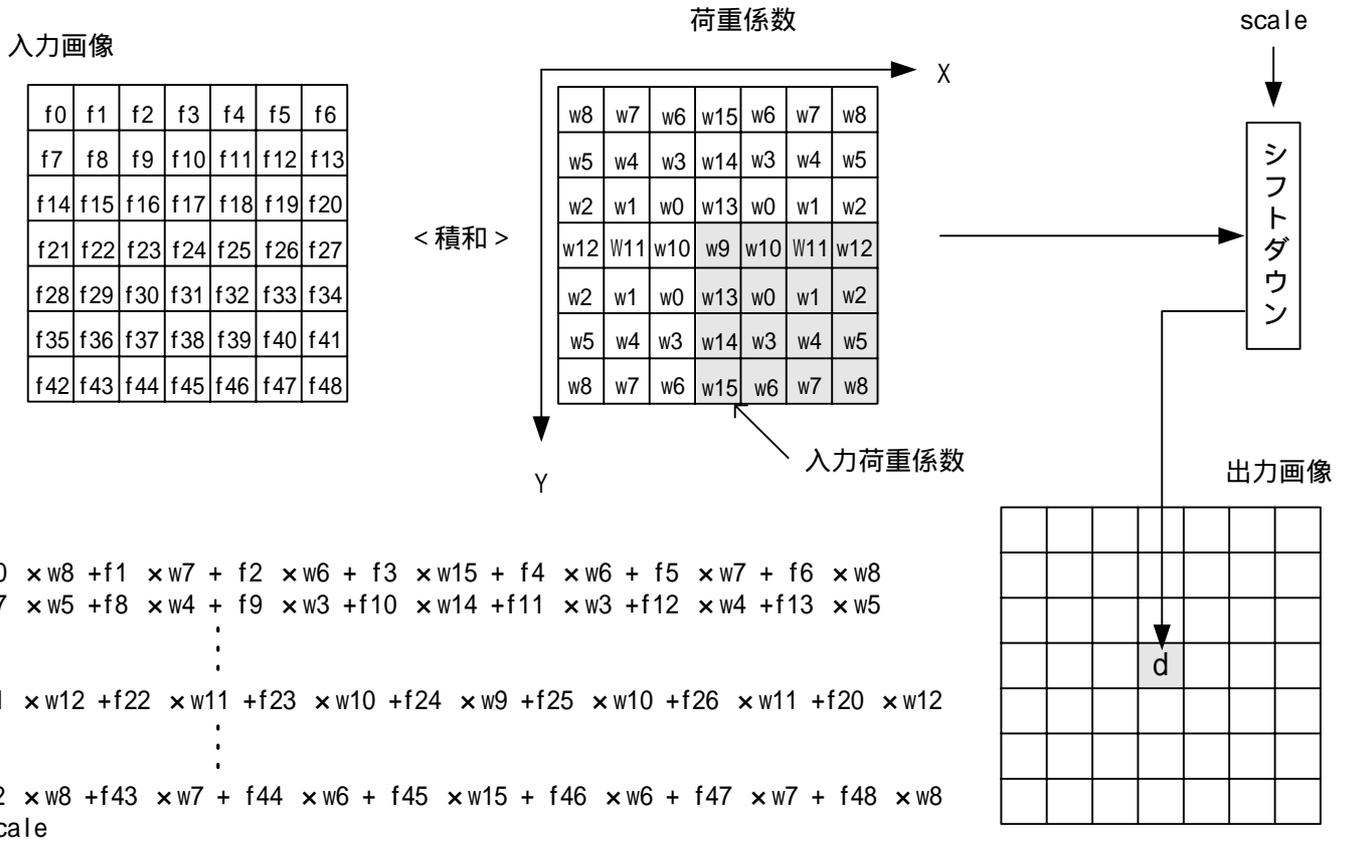
エラーコード	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
22	ダウンシフト量範囲外
49	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

詳細情報

輪郭強調では、入力画面の指定領域内の全画素に対して、注目画素を中心とした7×7近傍の局所領域で与えられた荷重係数との積和演算を行います。

また、対象領域の周辺(7×7近傍に領域外が含まれる部分)処理の場合は0を出力します。



荷重係数の例を、以下に示します。また、係数テーブルのフォーマットは、IP\_SmoothFLT7x7の備考の欄を参照してください。

荷重係数の例

< 例1 > X方向微分

0	-1	1	0	1	-1	0
0	-1	1	0	1	-1	0
0	-1	1	0	1	-1	0
0	-1	1	0	1	-1	0
0	-1	1	0	1	-1	0
0	-1	1	0	1	-1	0
0	-1	1	0	1	-1	0

< 例2 > Y方向微分

0	0	0	0	0	0	0
-1	-1	-1	-1	-1	-1	-1
1	1	1	1	1	1	1
0	0	0	0	0	0	0
1	1	1	1	1	1	1
-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0

この例の場合、ダウンシフト量は0です。

## IP\_MinFLT5x5

## 局所最小値フィルタ(5x5近傍)

### 機能

ソース画面に対し、5 × 5 近傍で、パラメータで指定されたパターンで最小濃度値を見つけ、デステーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_MinFLT5x5(
    DEVID
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
calptn
```

### コメント

```
デバイスID
ソース画面番号
デステーション画面番号
演算パターン
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面の画面番号

#### ImgDst

デステーション画面の画面番号

#### calptn

演算パターン(下位9ビット有効)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

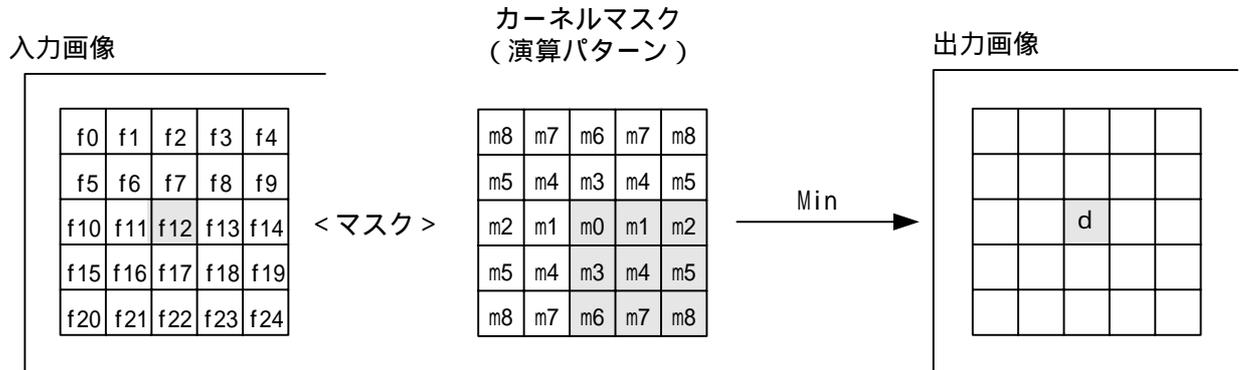
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

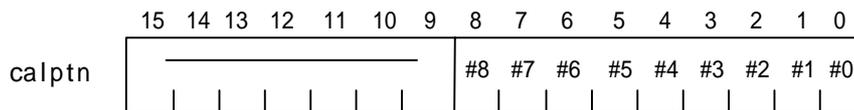
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

局所最小値フィルタ（5 × 5 近傍）では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 5 × 5 近傍の局所領域で与えられた演算パターンでマスク処理し、有効となった画素の中から最小の画素を出力します。周辺では中央画素をそのまま出力します。



パラメータcalptn のフォーマットを下記に示します。



0 : 演算マスク

1 : 演算イネーブル

#0	#1	#2
#3	#4	#5
#6	#7	#8

← 演算パターン

## IP\_MinFLT44

## 局所最小値フィルタ(5x5近傍、44連結)

## 機能

ソース画面に対し、5 × 5 近傍・4 4 連結で、パラメータで指定されたパターンで最小濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_MinFLT44(
  DEVID
  int
  int
)
```

```
devID ,
ImgSrc ,
ImgDst
```

## コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
```

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

ソース画面の画面番号

ImgDst

デスティネーション画面の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

局所最小値フィルタ ( 5 × 5 近傍、4 4 連結 ) では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 5 × 5 近傍の局所領域を 4 4 連結の演算パターンでマスク処理し、有効となった画素の中から最小の画素を出力します。周辺では中央画素をそのまま出力します。

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

カーネルマスク  
( 演算パターン )

0 : 演算マスク  
1 : 演算イネーブル

## IP\_MinFLT48

## 局所最小値フィルタ(5x5近傍、48連結)

## 機能

ソース画面に対し、5 × 5 近傍・48 連結で、パラメータで指定されたパターンで最小濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_MinFLT48(
    DEVID
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面の画面番号

## ImgDst

デスティネーション画面の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	画面空き領域、または範囲外(不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

局所最小値フィルタ(5 × 5 近傍、48 連結)では、入力画面の指定領域内の全画素に対して、注目画素を中心とした5 × 5 近傍の局所領域を48 連結の演算パターンでマスク処理し、有効となった画素の中から最小の画素を出力します。周辺では中央画素をそのまま出力します。

0	1	1	1	0
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	1	0

カーネルマスク  
(演算パターン)

0 : 演算マスク  
1 : 演算イネーブル

## IP\_MinFLT88

## 局所最小値フィルタ(5x5近傍, 88連結)

## 機能

ソース画面に対し、5 × 5 近傍・88 連結で、パラメータで指定されたパターンで最小濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_MinFLT88(
    DEVID
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst
```

## コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
```

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面の画面番号

## ImgDst

デスティネーション画面の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10 , 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

局所最小値フィルタ (5 × 5 近傍、88 連結) では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 5 × 5 近傍の局所領域を 88 連結の演算パターンでマスク処理し、有効となった画素の中から最小の画素を出力します。周辺では中央画素をそのまま出力します。

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

カーネルマスク  
(演算パターン)

0 : 演算マスク  
1 : 演算イネーブル

## IP\_MaxFLT5x5

## 局所最大値フィルタ(5x5近傍)

## 機能

ソース画面に対し、5 × 5 近傍で、パラメータで指定されたパターンで最大濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_MaxFLT5x5(
    DEVID
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
calptn
```

## コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
演算パターン
```

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面の画面番号

## ImgDst

デスティネーション画面の画面番号

## calptn

演算パターン(下位9ビット有効)

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

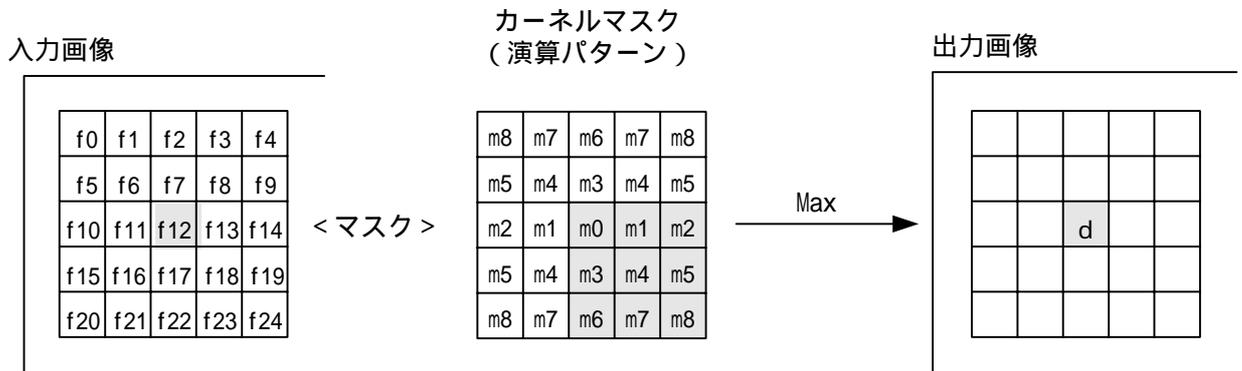
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外(不当画面番号エラー)
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド(ClearIPError)未発行

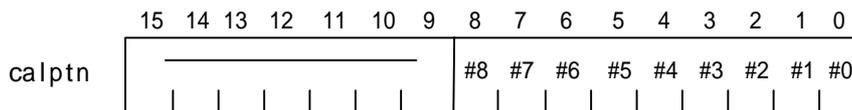
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

局所最大値フィルタ(5×5近傍)では、入力画面の指定領域内の全画素に対して、注目画素を中心とした5×5近傍の局所領域で与えられた演算パターンでマスク処理し、有効となった画素の中から最大の画素を出力します。周辺では中央画素をそのまま出力します。

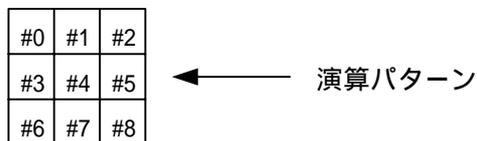


パラメータcalptn のフォーマットを下記に示します。



0 : 演算マスク

1 : 演算イネーブル



## IP\_MaxFLT44

### 局所最大値フィルタ(5x5近傍、44連結)

#### 機能

ソース画面に対し、5 x 5 近傍・4 4 連結で、パラメータで指定されたパターンで最大濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

#### コーディング

```
int
IP_MaxFLT44(
    DEVID
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst
```

#### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
```

#### パラメータ

##### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

##### ImgSrc

ソース画面の画面番号

##### ImgDst

デスティネーション画面の画面番号

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

#### 詳細情報

局所最大値フィルタ (5 x 5 近傍、4 4 連結) では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 5 x 5 近傍の局所領域を 4 4 連結の演算パターンでマスク処理し、有効となった画素の中から最大の画素を出力します。周辺では中央画素をそのまま出力します。

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

カーネルマスク  
(演算パターン)

0 : 演算マスク  
1 : 演算イネーブル

## IP\_MaxFLT48

## 局所最大値フィルタ(5x5近傍、48連結)

## 機能

ソース画面に対し、5 × 5 近傍・48 連結で、パラメータで指定されたパターンで最大濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_MaxFLT48(
    DEVID
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst
```

## コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
```

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面の画面番号

## ImgDst

デスティネーション画面の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10 , 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

局所最大値フィルタ (5 × 5 近傍、48 連結) では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 5 × 5 近傍の局所領域を 48 連結の演算パターンでマスク処理し、有効となった画素の中から最大の画素を出力します。周辺では中央画素をそのまま出力します。

0	1	1	1	0
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	1	0

カーネルマスク  
(演算パターン)

0 : 演算マスク  
1 : 演算イネーブル

## IP\_MaxFLT88

## 局所最大値フィルタ(5x5近傍、88連結)

## 機能

ソース画面に対し、5 × 5 近傍・88 連結で、パラメータで指定されたパターンで最大濃度値を見つけ、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_MaxFLT88(
    DEVID
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst
```

## コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
```

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面の画面番号

## ImgDst

デスティネーション画面の画面番号

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10 , 13	画面空き領域、または範囲外 (不当画面番号エラー)
49	アクセス画面が不定画面
50	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

局所最大値フィルタ (5 × 5 近傍、88 連結) では、入力画面の指定領域内の全画素に対して、注目画素を中心とした5 × 5 近傍の局所領域を 88 連結の演算パターンでマスク処理し、有効となった画素の中から最大の画素を出力します。周辺では中央画素をそのまま出力します。

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

カーネルマスク  
(演算パターン)

0 : 演算マスク  
1 : 演算イネーブル

## IP\_SmoothFLT5x5Ext

## 拡張平滑化(5 × 5)

## 機能

ソース画面（濃淡）に対し、5 × 5 近傍で平滑化を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_SmoothFLT5x5Ext(
    DEVID
    int
    int
    int
    enum IPConvExtOpt
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
scale ,
opt ,
*COEFF
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
オプション  
係数テーブルのポインタ

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面の画面番号

## ImgDst

デスティネーション画面の画面番号

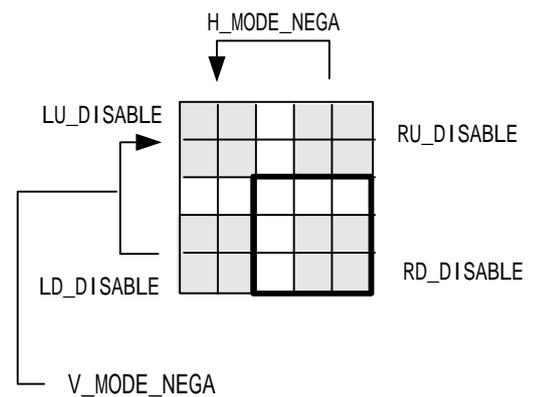
## scale

ダウンシフト量（0 ~ 15）

## opt

拡張コンポリューションオプション

オプション	対応定数	内容
H_MODE_NEGA	3 2	負対称（水平方向）
V_MODE_NEGA	6 4	負対称（垂直方向）
RD_DISABLE	1	右下演算処理無効
LD_DISABLE	2	左下演算処理無効
RU_DISABLE	4	右上演算処理無効
LU_DISABLE	8	左上演算処理無効



## \*COEFF

係数テーブルのポインタ

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

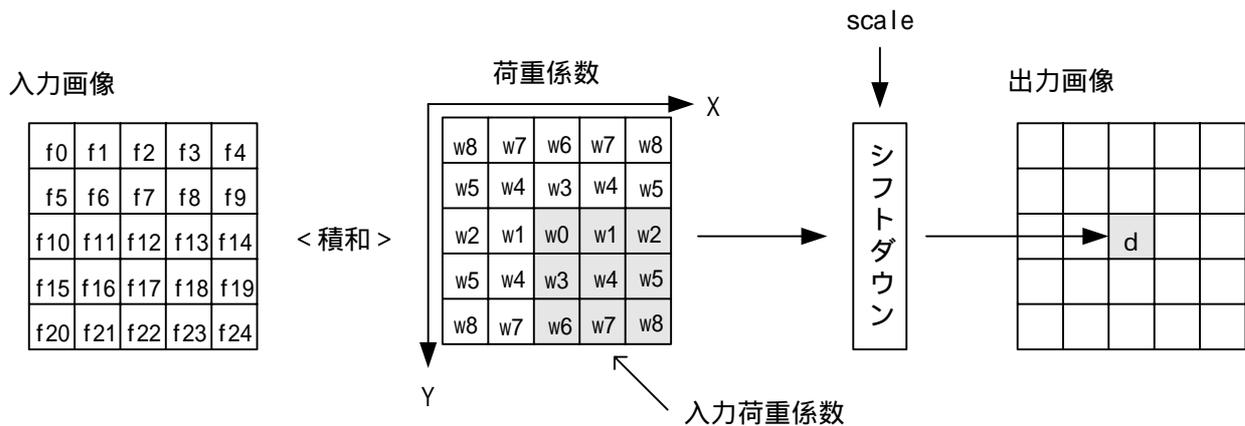
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 1	ダウンシフト量設定範囲外
2 2	オプション設定範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPErrors) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

平滑化では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 5 × 5 近傍の局所領域で与えられた荷重係数との積和演算を行います。



$$\begin{aligned}
 d = & ( f0 \times w8 + f1 \times w7 + f2 \times w6 + f3 \times w7 + f4 \times w8 \\
 & + f5 \times w5 + f6 \times w4 + f7 \times w3 + f8 \times w4 + f9 \times w5 \\
 & + f10 \times w2 + f11 \times w1 + f12 \times w0 + f13 \times w1 + f14 \times w2 \\
 & + f15 \times w5 + f16 \times w4 + f17 \times w3 + f18 \times w4 + f19 \times w5 \\
 & + f20 \times w8 + f21 \times w7 + f22 \times w6 + f23 \times w7 + f24 \times w8 \\
 & ) \gg \text{scale}
 \end{aligned}$$

荷重係数の例を、以下に示します。また、係数テーブルのフォーマットは、IP\_SmoothFLT5x5()の備考の欄を参照してください。

## 荷重係数

&lt; 例1 &gt;

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

opt == 0

&lt; 例2 &gt;

0	0	0	0	0
0	3	3	3	0
0	3	3	3	0
0	3	3	3	0
0	0	0	0	0

opt == 0

この例の場合、ダウンシフト量は5です。

## [ 拡張コンボリュ-ション設定例 ]

全カーネル有効で負対称 ( 垂直方向 )

```
opt = ( enum IPConvExtOpt )( V_MODE_NEGA )
```

右上と左下カーネル無効で負対称 ( 水平方向 )

```
opt = ( enum IPConvExtOpt )( LD_DISABLE | RU_DISABLE | H_MODE_NEGA );
```

( 注 ) Opt=0 のとき、Extなしの拡張コンボリュ-ションに一致

## IP\_SmoothFLT7x7Ext

## 拡張平滑化(7×7)

## 機能

ソース画面（濃淡）に対し、7×7近傍で平滑化を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

## コーディング

```
int
IP_SmoothFLT7x7Ext(
    DEVID
    int
    int
    int
    enum IPConvExtOpt
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
scale ,
opt ,
*COEFF
```

## コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
ダウンシフト量  
オプション  
係数テーブルのポインタ

## パラメータ

## devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

## ImgSrc

ソース画面の画面番号

## ImgDst

デスティネーション画面の画面番号

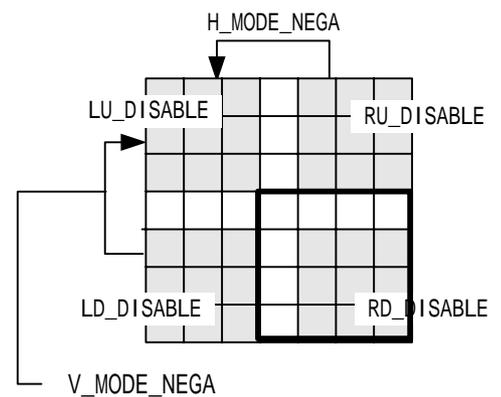
## scale

ダウンシフト量(0～15)

## opt

拡張コンポリューションオプション

オプション	対応定数	内容
H_MODE_NEGA	3 2	負対称(水平方向)
V_MODE_NEGA	6 4	負対称(垂直方向)
RD_DISABLE	1	右下演算処理無効
LD_DISABLE	2	左下演算処理無効
RU_DISABLE	4	右上演算処理無効
LU_DISABLE	8	左上演算処理無効



## \*COEFF

係数テーブルのポインタ

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

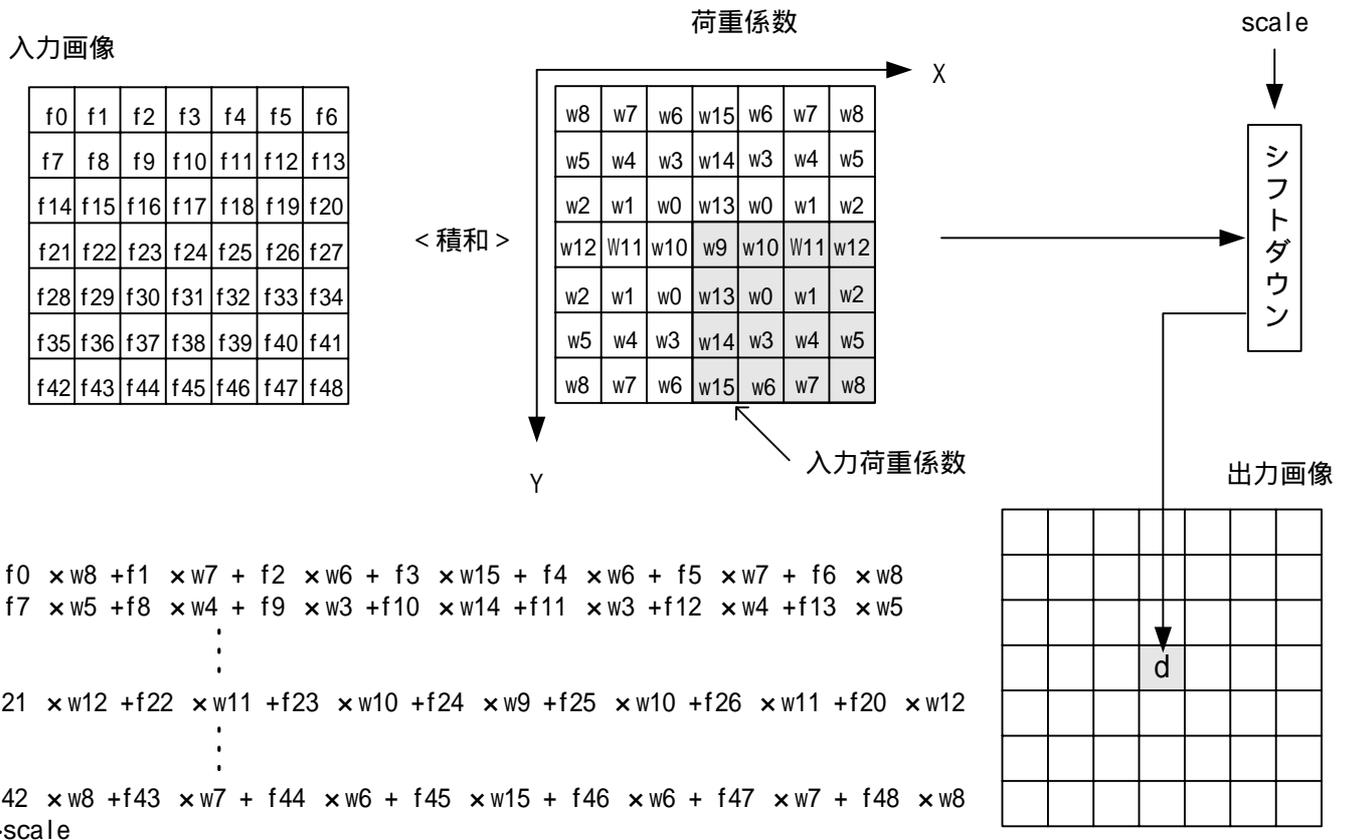
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 1	ダウンシフト量設定範囲外
2 2	オプション設定範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

平滑化では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 7 × 7 近傍の局所領域で与えられた荷重係数との積和演算を行います。



荷重係数の例を、以下に示します。また、係数テーブルのフォーマットは、IP\_SmoothFLT7x7()の備考の欄を参照してください。

#### 荷重係数

< 例1 >

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

opt == 0

この例の場合、ダウンシフト量は6です。

#### [ 拡張コンボリュ-ション設定例 ]

全カーネル有効で負対称（垂直方向）

opt = ( enum IPConvExtOpt )( V\_MODE\_NEGA )

右上と左下カーネル無効で負対称（水平方向）

opt = ( enum IPConvExtOpt )( LD\_DISABLE | RU\_DISABLE | H\_MODE\_NEGA );

（注）Opt=0 のとき、Extなしの拡張コンボリュ-ションに一致

# IP\_EdgeFLT5x5Ext

## 拡張濃淡画像輪郭強調(5x5近傍)

### 機能

ソース画面(濃淡)に対し、5×5近傍で輪郭強調を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_EdgeFLT5x5Ext(
    DEVID
    int
    int
    int
    enum IPConvExtOpt
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
scale ,
opt ,
*COEFF
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
ダウンシフト量
オプション
係数テーブルのポインタ
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面の画面番号

#### ImgDst

デスティネーション画面の画面番号

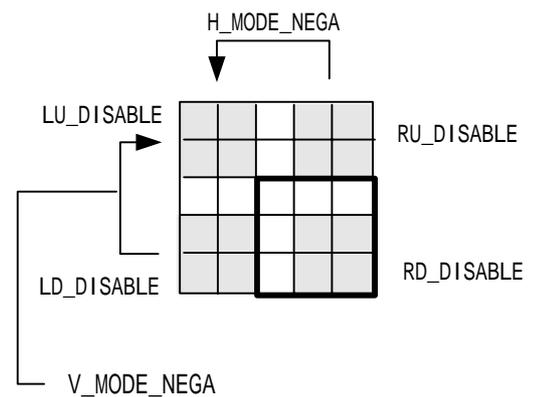
#### scale

ダウンシフト量(0~15)

#### opt

拡張コンボリユ-ションオプション

オプション	対応定数	内容
H_MODE_NEGA	3 2	負対称(水平方向)
V_MODE_NEGA	6 4	負対称(垂直方向)
RD_DISABLE	1	右下演算処理無効
LD_DISABLE	2	左下演算処理無効
RU_DISABLE	4	右上演算処理無効
LU_DISABLE	8	左上演算処理無効



#### \*COEFF

係数テーブルのポインタ

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

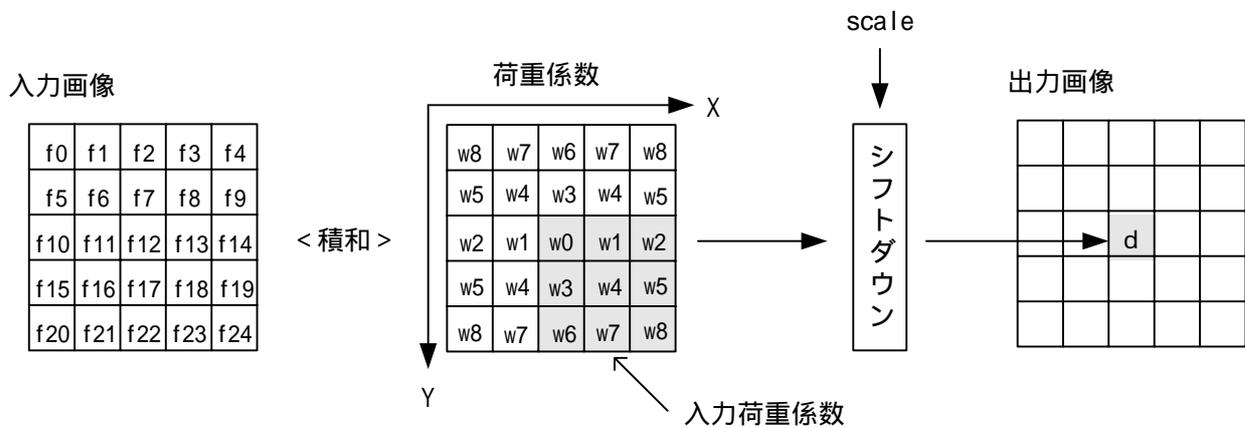
エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 1	ダウンシフト量設定範囲外
2 2	オプション設定範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

輪郭強調では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 5 × 5 近傍の局所領域で与えられた荷重係数との積和演算を行います。

また、対象領域の周辺 ( 5 × 5 近傍に領域外が含まれる部分 ) 処理の場合は 0 を出力します。



$$\begin{aligned}
 d = & ( f0 \times w8 + f1 \times w7 + f2 \times w6 + f3 \times w7 + f4 \times w8 \\
 & + f5 \times w5 + f6 \times w4 + f7 \times w3 + f8 \times w4 + f9 \times w5 \\
 & + f10 \times w2 + f11 \times w1 + f12 \times w0 + f13 \times w1 + f14 \times w2 \\
 & + f15 \times w5 + f16 \times w4 + f17 \times w3 + f18 \times w4 + f19 \times w5 \\
 & + f20 \times w8 + f21 \times w7 + f22 \times w6 + f23 \times w7 + f24 \times w8 \\
 & ) \gg \text{scale}
 \end{aligned}$$

荷重係数の例を、以下に示します。また、係数テーブルのフォーマットは、IP\_EdgeFLT5x5()の備考の欄を参照してください。

### 荷重係数の例

< 例1 > X方向微分

0	-1	0	1	0
0	-1	0	1	0
0	-1	0	1	0
0	-1	0	1	0
0	-1	0	1	0

opt == H\_MODE\_NEGA

< 例2 > Y方向微分

0	0	0	0	0
-1	-1	-1	-1	-1
0	0	0	0	0
1	1	1	1	1
0	0	0	0	0

opt == V\_MODE\_NEGA

この例の場合、ダウンシフト量は0です。

< 例3 > X方向微分

2	1	0	-1	-2
2	1	0	-1	-2
2	1	0	-1	-2
2	1	0	-1	-2
2	1	0	-1	-2

opt == H\_MODE\_NEGA

この例の場合、ダウンシフト量は0です。

# IP\_EdgeFLT7x7Ext

## 拡張濃淡画像輪郭強調(7x7近傍)

### 機能

ソース画面（濃淡）に対し、7 × 7 近傍で輪郭強調を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_EdgeFLT7x7Ext(
    DEVID
    int
    int
    int
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
scale ,
opt ,
*COEFF
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
ダウンシフト量
オプション
係数テーブルのポインタ
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面の画面番号

#### ImgDst

デスティネーション画面の画面番号

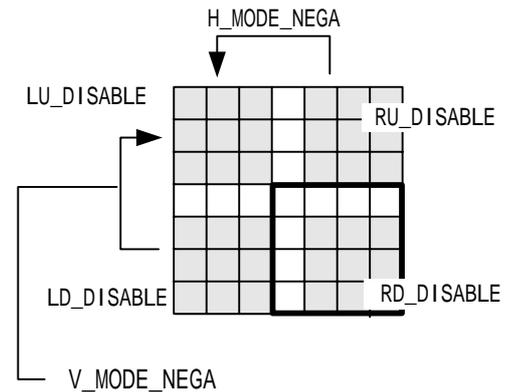
#### scale

ダウンシフト量 ( 0 ~ 15 )

#### opt

拡張コンポリューションオプション

オプション	対応定数	内容
H_MODE_NEGA	3 2	負対称（水平方向）
V_MODE_NEGA	6 4	負対称（垂直方向）
RD_DISABLE	1	右下演算処理無効
LD_DISABLE	2	左下演算処理無効
RU_DISABLE	4	右上演算処理無効
LU_DISABLE	8	左上演算処理無効



#### \*COEFF

係数テーブルのポインタ

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

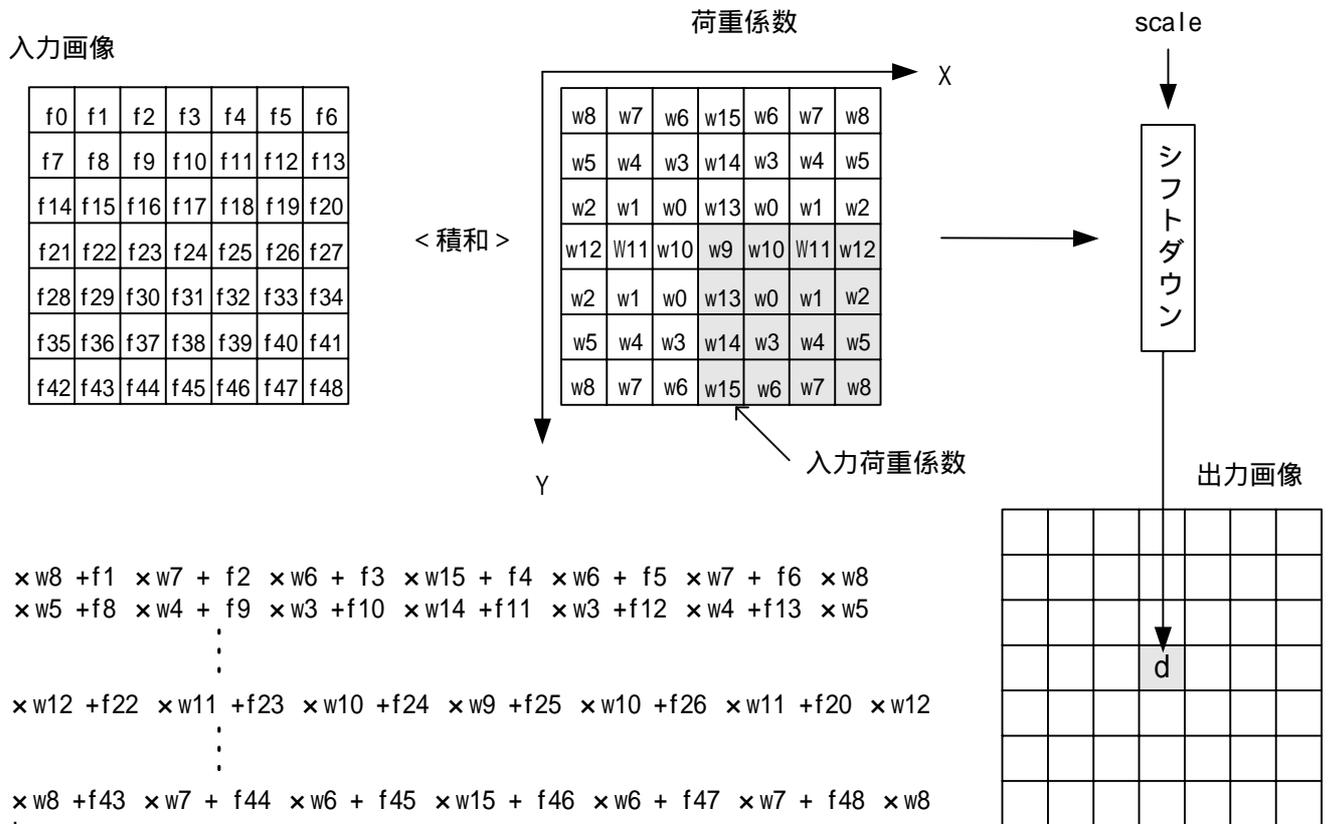
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 1	ダウンシフト量設定範囲外
2 2	オプション設定範囲外
4 9	アクセス画面が不定画面
5 0	画像メモリチャネル競合
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

平滑化では、入力画面の指定領域内の全画素に対して、注目画素を中心とした 7 × 7 近傍の局所領域で与えられた荷重係数との積和演算を行います。



$$\begin{aligned}
 d = & ( f_0 \times w_8 + f_1 \times w_7 + f_2 \times w_6 + f_3 \times w_{15} + f_4 \times w_6 + f_5 \times w_7 + f_6 \times w_8 \\
 & + f_7 \times w_5 + f_8 \times w_4 + f_9 \times w_3 + f_{10} \times w_{14} + f_{11} \times w_3 + f_{12} \times w_4 + f_{13} \times w_5 \\
 & \vdots \\
 & + f_{21} \times w_{12} + f_{22} \times w_{11} + f_{23} \times w_{10} + f_{24} \times w_9 + f_{25} \times w_{10} + f_{26} \times w_{11} + f_{20} \times w_{12} \\
 & \vdots \\
 & + f_{42} \times w_8 + f_{43} \times w_7 + f_{44} \times w_6 + f_{45} \times w_{15} + f_{46} \times w_6 + f_{47} \times w_7 + f_{48} \times w_8 \\
 & ) \gg \text{scale}
 \end{aligned}$$

荷重係数の例を、以下に示します。また、係数テーブルのフォーマットは、IP\_EdgeFLT7x7()の備考の欄を参照してください。

## 荷重係数

&lt; 例1 &gt;

-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3

opt == H\_MODE\_NEGA

&lt; 例2 &gt;

-3	-3	-3	-3	-3	-3	-3
-2	-2	-2	-2	-2	-2	-2
-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3

opt == V\_MODE\_NEGA

この例の場合、ダウンシフト量は0です。

## [ 拡張コンボリュ-ション設定例 ]

全カーネル有効で負対称(垂直方向)

```
opt = ( enum IPConvExtOpt )( V_MODE_NEGA )
```

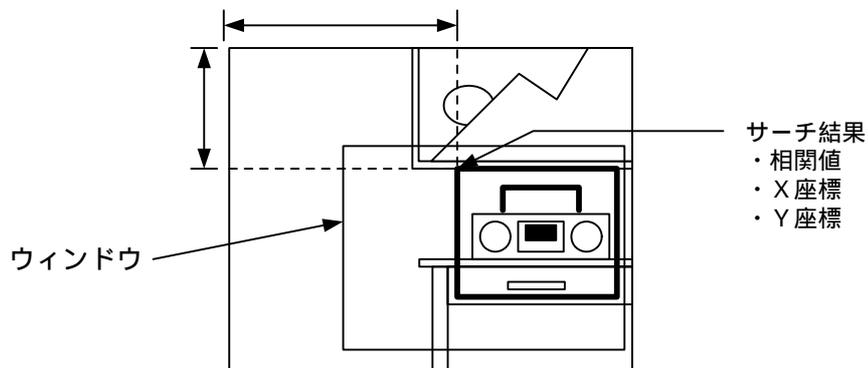
右上と左下カーネル無効で負対称(水平方向)

```
opt = ( enum IPConvExtOpt )( LD_DISABLE | RU_DISABLE | H_MODE_NEGA );
```

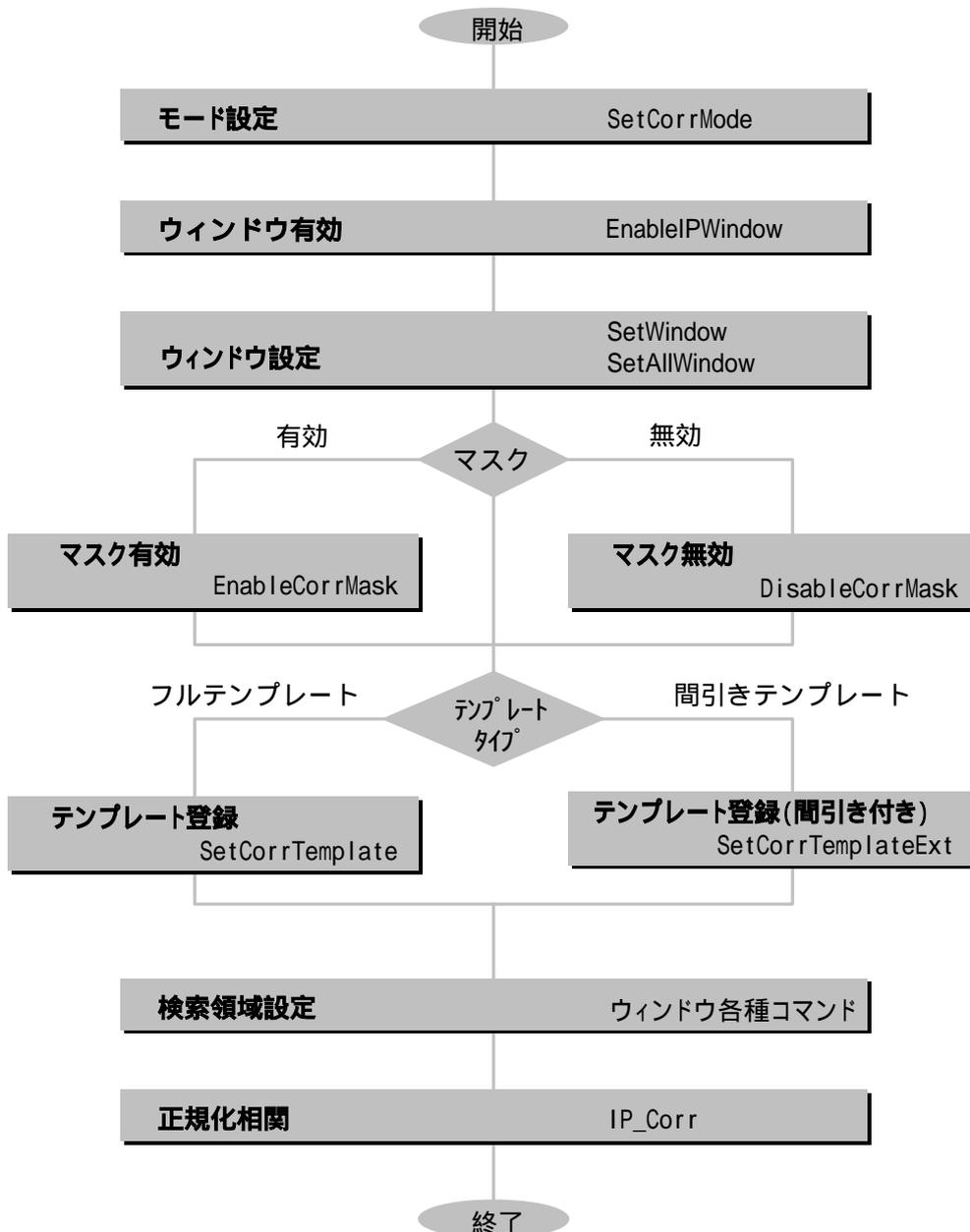
(注) Opt=0 のとき、Extなしの拡張コンボリュ-ションに一致

## 51. 正規化相関(IP5000互換)コマンド

正規化相関コマンドでは、正規化相関演算が実行できます。  
 正規化相関コマンドでの座標系は、画面絶対座標であり、画面始点が座標0になります。  
 よって、正規化相関処理結果の座標は画面始点を基準とした座標です。



(1) 正規化相関実行手順を下図のフローで実行します。



## (2) テンプレートタイプ

正規化相関のテンプレートには、2種類のタイプがあります。

### ・フルテンプレート

SetCorrTemplate()コマンドで登録されたテンプレートです。フルテンプレートは、IP\_Corr()実行時に、処理対象画面内全画素を処理する(フルサーチ)ので、高精度な処理ができます。

### ・間引きテンプレート

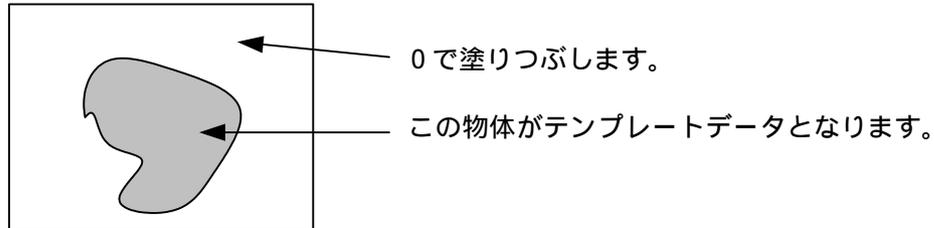
SetCorrTemplateExt()コマンドで登録されたテンプレートです。間引きテンプレートは、X/Y方向の間引き率を登録できます。この間引き率により、テンプレートデータおよび処理対象画面を間引いて演算処理するので、精度はフルテンプレートに対して落ちますが、演算回数が減るため、高速処理ができます。

処理対象画像に応じて、これら2つのテンプレートタイプを評価し使用してください。

## (3) 正規化相関マスク

正規化相関マスクは、テンプレート内で濃度0の画素を処理対象外にします。よって、矩形でない任意の形をテンプレートとしたい場合に有効です。矩形のテンプレート内で必要物体部以外を0で塗りつぶしておき、IP\_Corr()を実行する前に、EnableCorrMask()を実行すれば、0画素部分は処理されないため、任意形のテンプレートで処理したことと同じになります。

正規化相関マスクはEnableCorrMask()コマンドで設定します。



# SetCorrMode

## 正規化相関モード設定

### 機能

正規化相関での処理モードを設定します。

### コーディング

```
int
SetCorrMode(
    DEVID
    int
    int
    float
)
```

```
devID ,
SearchMode ,
count ,
CorrThr
```

### コメント

デバイス I D  
走査方法  
検知物体数  
相関値

### パラメータ

#### devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

#### SearchMode

走査方法

走査方法	対応定数	内容
RASTER	0	ラスタ型に走査
RASTER_STOP	1	同上 (但し、相関値以上の物体を指定個数検知すると停止します。)
SPIRAL	2	スパイラル型に走査
SPIRAL_STOP	3	同上 (但し、相関値以上の物体を指定個数検知すると停止します。)
CORR_FULL_SEARCH	8	全面サーチ

設定例のように全面サーチと併用して設定してください。

#### count

検知物体数 ( 1 以上 )

#### CorrThr

相関値 ( 0 ~ 1 )

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	走査方法設定値範囲外
8 4	検知物体数エラー
8 5	相関値エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

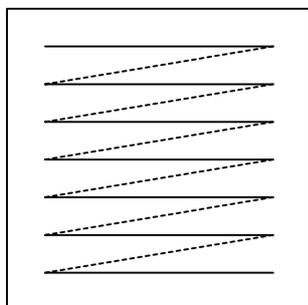
探索モードについて説明します。

探索方法と終了条件指定

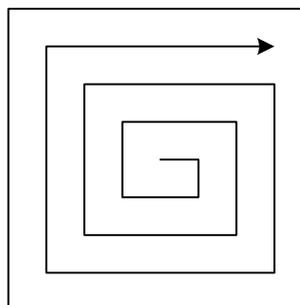
RASTER, RASTER\_STOP, SPIRAL, SPIRAL\_STOP

- ・走査方法は下記に示す通り、ラスタ型は左上から、スパイラル型は中央から走査します。
- ・STOP モードは、探索途中でも指定された個数分閾値を超えた場合、探索を終了します。

ラスタ型



スパイラル型



[ 設定例 ]

ラスタサーチ&フルサーチ方式

```
SetCorrMode(devID, RASTER | CORR_FULL_SEARCH, count, CorrThr );
```

# DisableCorrMask

## 正規化関連マスク無効

### 機能

正規化関連処理での、テンプレート内 0 値のマスク無効を設定します。

### コーディング

```
int  
DisableCorrMask(  
    DEVID  
)
```

### コメント

devID

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## EnableCorrMask

## 正規化相関マスク有効

### 機能

正規化相関処理での、テンプレート内 0 値のマスク有効を設定します。

### コーディング

```
int
EnableCorrMask(
    DEVID
)
```

devID

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

正規化相関マスクは、テンプレート内で濃度 0 の画素を処理対象外にします。よって、矩形でない任意の形をテンプレートとしたい場合に有効です。矩形のテンプレート内で必要物体部以外を 0 で塗りつぶしておき、IP\_Corr()を実行する前に、EnableCorrMask()を実行すれば、0 画素部分は処理されないため、任意形のテンプレートで処理したことと同じになります。

EnableCorrMask()コマンドは、テンプレート登録 (SetCorrTemplate()) をする前に実行してください。

# SetCorrBreakThr

## 打切りしきい値設定

### 機能

システムが管理しているテーブル(システム・テーブル)に累積誤差しきい値計算用パラメータを設定します。

### コーディング

```
int
SetCorrBreakThr(
    DEVID
    int
)
```

```
devID ,
thr
```

### コメント

デバイスID  
累積誤差しきい値

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**thr**

累積誤差しきい値設定(0 ~ 255)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
20	累積誤差しきい値設定パラメータ設定範囲外
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

本コマンドは、指定累積誤差オーバによる相関演算打切り制御に使用します。  
相関演算打切り制御とは、相関演算処理中、累積誤差が設定したしきい値を超えた時点でサーチ対象画像がテンプレート画像と明らかに異なるものと判断し相関演算を打切ります。  
途中で演算を打切ることにより処理の高速化を図ることができます。

## しきい値の設定

実際に演算を途中で打ち切るための累積誤差しきい値は、

$$\text{累積誤差しきい値} = \text{thr} \quad : \quad \text{累積誤差しきい値設定パラメータ} \times \text{テンプレート画素数}$$

で計算されます。また、 $\text{thr} = 0$  を設定した場合、自動しきい値変更モードとなり、常に最小値をしきい値として、演算の打切りを行います (SSDA 法の適用)。この場合、照明変動にはかなり弱くなりますので十分に注意して御使用ください。

また、照明変動がほとんどない場合は  $\text{thr}$  を小さく、照明変動がある場合には、 $\text{thr}$  を大きく設定してください。

## 相関演算打切り処理 許可 / 禁止は

```
EnableCorrBreak( );      // 相関演算打切り処理許可  
DisableCorrBreak( );    // 相関演算打切り処理禁止(デフォルト)
```

を御使用ください。

# DisableCorrBreak

## 相関演算打切り禁止(デフォルト)

### 機能

相関演算打切り処理を禁止します。

### コーディング

```
int
DisableCorrBreak(
    DEVID          devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドは、指定累積誤差オーバーによる相関演算打切り制御に使用します。

相関演算打切り制御とは、サーチ対象画像がテンプレート画像と明らかに異なる場合、相関演算実行中に累積誤差が非常に大きくなると考えられます。そこで、この累積誤差がある閾値を超えた時点で、ミスマッチと判断し、途中で相関演算を中止します。

途中で演算を中止することにより処理の高速化を図ることができます。

しきい値の設定については、SetCorrBreakThr()を参照してください。

# EnableCorrBreak

## 相関演算打切り許可

### 機能

相関演算打切り処理を禁止します。

### コーディング

```
int
EnableCorrBreak(
    DEVID          devID
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

本コマンドは、指定累積誤差オーバによる相関演算打切り制御に使用します。

相関演算打切り制御とは、サーチ対象画像がテンプレート画像と明らかに異なる場合、相関演算実行中に累積誤差が非常に大きくなると考えられます。そこで、この累積誤差がある閾値を超えた時点で、ミスマッチと判断し、途中で相関演算を中止します。

途中で演算を中止することにより処理の高速化を図ることができます。

しきい値の設定については、SetCorrBreakThr()を参照してください。

# SetCorrControl

## 正規化相関制御

### 機能

正規化相関での相関値を符号付の値で扱うか、絶対値で扱うかを制御します。

### コーディング

```
int
SetCorrControl(
    DEVID
    IPCorrControl
)
```

```
devID ,
*cnt
```

### コメント

デバイスID  
制御モード構造体のアドレス

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*cnt

制御モード設定用構造体のアドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

テンプレート登録関数 ( SetCorrTemplate(), SetCorrTemplateExt() ) の戻り値として、テンプレートの分散値を返すように設定できます。

正規化相関での相関値を符号付きか、絶対値のいずれかで扱うかを指定できます。  
絶対値で扱うとポジ/ネガの関係の画像でも一致していると評価されます。

以下の設定のデフォルトを使用する場合には、本コマンドを発行する必要はありません。

IPCorrControl のフォーマットを以下に示します。

```
typedef struct {
    int SignOperation;    /* 0 : 相関値は絶対値で処理 (デフォルト)
                        1 : 相関値を符号付きで処理
                        2 : 相関値演算中は絶対値で処理、戻り値には符号をつける
                        */
    int ReturnValueOfSetCorrTemplate;
                        /* 0 : 正常終了 = 0 、異常終了 = -1 (デフォルト)
                        1 : テンプレートの分散値を返す。異常終了 = -1
                        */
} IPCorrControl;
```

# SetCorrTemplate

## 正規化相関テンプレート登録

### 機能

正規化相関で使用するための、テンプレートを登録します。

### コーディング

```
int
SetCorrTemplate(
    DEVID
    int
    int
)
```

```
devID ,
ImgTmp ,
TmpID
```

### コメント

```
デバイス I D
画面番号
テンプレート番号
```

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

ImgTmp

テンプレート登録画面番号

TmpID

テンプレート番号 ( 0 ~ 2 5 5 )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
4 9	アクセス画面が不定画面
8 0	テンプレートサイズエラー
8 1	テンプレート画面の不当画面番号エラー
8 3	テンプレート内同一画素エラー
8 6	マスク処理でのテンプレート登録エラー
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

テンプレートの位置およびサイズは、ウィンドウ (SRCO\_WIN) によって決まります。このコマンド発行前に必ず、ウィンドウ設定コマンドによりSRCO\_WINウィンドウを設定してください。

また、テンプレートのサイズは 2~65,536画素 (256×256相当) までとし、テンプレート内が同一画素値のときはエラーとなります。

正規化相関テンプレート登録では、テンプレート画像の存在する画面番号 (ImgTmp) と座標情報のみ登録します。よって、テンプレート登録画面は解放 (FreeImg) したり、画像処理によるデータ変更をしないでください。

# SetCorrTemplateExt

## 正規化相関テンプレート登録(間引き付き)

### 機能

正規化相関で使用するための、テンプレート(間引きテンプレート)を登録します。

### コーディング

```
int
SetCorrTemplateExt(
    DEVID
    int
    int
    int
    int
)
```

```
devID ,
ImgTmp ,
TmpID ,
xmag ,
ymag
```

### コメント

```
デバイスID
画面番号
テンプレート番号
X方向の間引き率
Y方向の間引き率
```

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgTmp**  
テンプレートの登録画面番号

**TmpID**  
テンプレート番号(0~255)

**xmag**  
X方向の間引き率(1~8)

**ymag**  
Y方向の間引き率(1~8)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10	指定画面空き領域、または範囲外(不当画面番号エラー)
22	X方向の間引き率設定値範囲外
23	Y方向の間引き率設定値範囲外
49	アクセス画面が不定画面
80	テンプレートサイズエラー
81	テンプレート画面の不当画面番号エラー
83	テンプレート内同一画素エラー
86	マスク処理でのテンプレート登録エラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報**

テンプレートの位置およびサイズは、ウィンドウ (SRC0\_WIN) によって決まります。このコマンド発行前に必ず、ウィンドウ設定コマンドによりSRC0\_WINウィンドウを設定してください。

また、テンプレートのサイズは 2~65,536画素 (256×256 相当) までとし、テンプレート内が同一画素値のときはエラーとなります。

正規化相関テンプレート登録(間引き付き)では、テンプレート画像の存在する画面番号 (ImgTmp) と座標情報と間引き率を登録します。よって、テンプレート登録画面は解放 (FreeImg) したり、画像処理によるデータ変更をしないでください。

間引き率を2とすると、テンプレート内データは1画素おきで処理されるため、データ量は半分になります。間引きテンプレートは、処理が高速ですが、精度がフルテンプレートより落ちます。処理対象物体によって、フルテンプレート/間引きテンプレートどちらが有効か、また、間引きテンプレートによる間引き率は異なるため、実際に評価をしてから、使用してください。

精度

フルテンプレート &gt; 間引きテンプレート

処理速度

フルテンプレート &lt; 間引きテンプレート

## IP\_Corr

## 正規化相関

### 機能

正規化相関を実行し、検知した物体数をリターン値として返します。  
また、そのときの相関値および座標を、ユーザテーブルへセットします。

### コーディング

```
int
IP_Corr(
    DEVID
    int
    int
    IPCorrTbl
)
```

```
devID ,
ImgSrc ,
TmpID ,
*Tbl
```

### コメント

デバイス I D  
画面番号  
テンプレート番号  
相関値および座標テーブルのアドレス

### パラメータ

**devID**

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**ImgSrc**

サーチ対象画面番号

**TmpID**

テンプレート番号 ( 0 ~ 255 )

**\*Tbl**

相関値および座標をセットするテーブルのアドレス。正規化相関モード設定コマンド ( SetCorrMode ) で設定した検知物体数分を確保してください。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
- 1 以外	正常終了 ( 検知した物体の数 )
- 1	異常終了

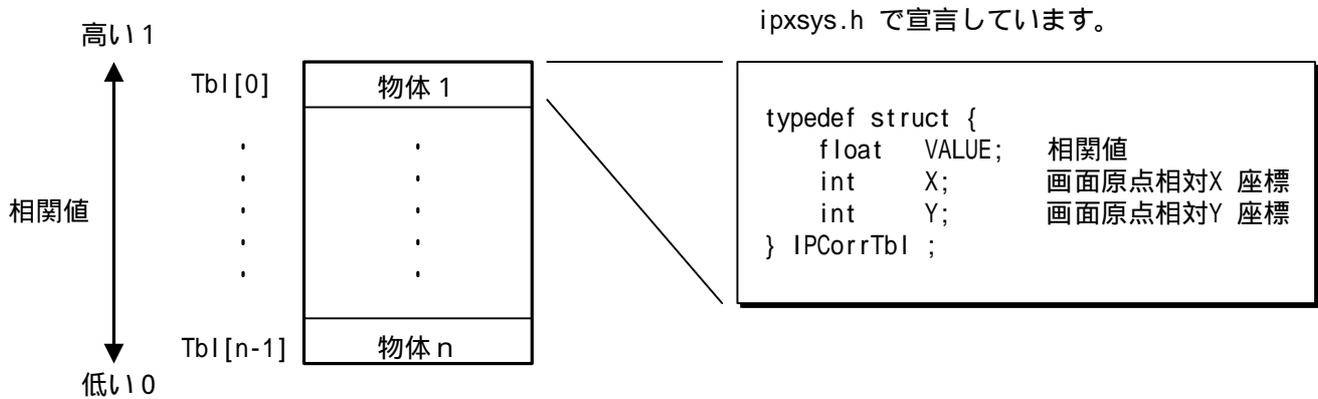
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
8 0	テンプレートサイズエラー
8 1	テンプレート画面の不当画面番号エラー
8 7	テンプレート未設定エラー
8 8	テンプレート登録画面番号エラー
-	エラーリセットコマンド ( ClearIPErrror ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

相関値および座標テーブルのフォーマットを、以下に示します。



IP\_Corr では、使用するテンプレートタイプにより、処理内容が異なります。  
 フルテンプレート (SetCorrTemplate() で登録) の場合は、処理領域内全画素を演算処理します。  
 間引きテンプレート (SetCorrTemplateExt() で登録) の場合は、登録された間引き率で処理領域内を間引いて処理します。

# IP\_CorrPrecise

## 正規化相関(高精度位置決め)

### 機能

正規化相関を (xc,yc) の近傍で実行し、最も相関値の高い位置のサブピクセル単位 (1画素単位以下) の座標値をユーザテーブルへセットします。

### コーディング

```
int
IP_CorrPrecise(
    DEVID
    int
    int
    int
    int
    int
    IPCorrPreciseTbl
    enum IPCorrPreciseOpt
)
```

```
devID ,
ImgSrc ,
TmpID ,
xc ,
yc ,
*Tbl ,
opt
```

### コメント

```
デバイスID
ソース画面番号
テンプレート管理番号
サーチ中心X座標指定
サーチ中心Y座標指定
サブピクセル単位の座標のアドレス
オプション指定
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面番号

#### TmpID

テンプレート管理番号

#### xc, yc

サーチ中心座標指定 (X座標、Y座標)

#### \*Tbl

サブピクセル単位の座標をセットするテーブルのアドレス

#### opt

高精度位置決めオプション

高精度位置決めオプション	対応定数	処理内容
CRP_CENTER	0	xc, yc を中心座標とみなす
CRP_MAXVAL	1	上下左右の4近傍をサーチして相関値が最大となった点を中心とする

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了 (4近傍の範囲以内で高精度位置を計測し処理を終了)
- 1	異常終了
1	4近傍の範囲以外で高精度位置を計測し終了 この場合、4近傍の5点内で最も相関値の高い座標をテーブルにセット

異常終了時、オンボードCPU例外と下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 5	オプション設定値範囲外
4 9	アクセス画面が不定画面
8 0	テンプレートサイズエラー
8 1	テンプレート画面の不当画面番号エラー
8 2	サーチ領域が画面をはみ出す様なサーチポイントがあります
8 7	テンプレート未設定エラー
8 8	テンプレート登録画面番号エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

高精度位置決め座標テーブルのフォーマットを下記に示します。

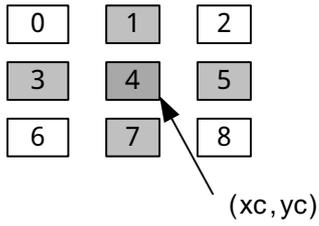
```
typedef struct{
    float X; /* X座標：画面相対アドレス */
    float Y; /* Y座標：画面相対アドレス */
} IPCorrPreciseTbl ;
```

- (xc, yc)は、画面相対座標で指定して下さい。
- 間引き指定付きテンプレートを用いると、4近傍のアドレス計算にも間引き率が反映します。  
(例)間引き率2を指定していると、4近傍のアドレスは中心から2画素離れます。
- ウィンドウ指定、SetCorrMode()は無視されます。
- 本コマンドIP\_CorrPrecise()は近傍の狭い範囲しか見ないため、まず、IP\_Corr()を用いて範囲を絞り込んでおく必要があります。

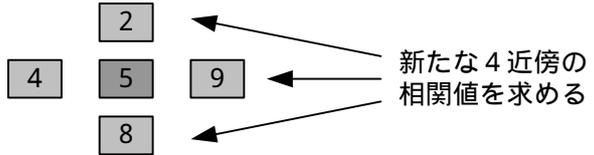
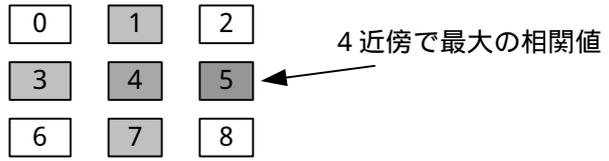
処理仕様：

( A ) オプション指定に従い4近傍の相関値を求める。

opt=CRP\_CENTER

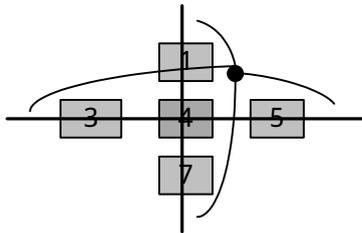


opt=CRP\_MAXVAL

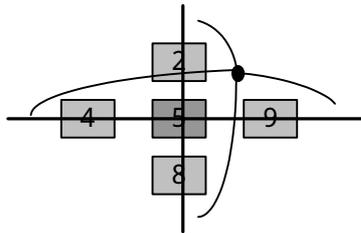


( B ) 求めたX方向とY方向の3点の相関値を用いて、2次近似により最大値の存在するX座標とY座標を得る。

opt=CRP\_CENTER



opt=CRP\_MAXVAL



# SetOptFlowMode

## オプティカルフロー設定

### 機能

オプティカルフローのモードを設定します。

### コーディング

```
int
SetOptFlowMode(
    DEVID
    enum IP_Opt_SearchMode
    IPFlowSize
    int
    int
)
```

### コメント

```
devID ,
SearchMode ,
tempsize ,
variance ,
thr_c_value
```

デバイスID  
 フロー計測探索範囲モード  
 テンプレートサイズ情報の構造体  
 フロー計測実行点を選定する閾値  
 中心での差分累積の評価値に対する閾値

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### SearchMode

フロー計測探索範囲モード

フロー計測探索範囲モード	対応定数	内容
OPT_FULL_SEARCH_3	0	左上2画素、右下3画素
OPT_FULL_SEARCH_6	1	左上5画素、右下6画素
OPT_FULL_SEARCH_9	2	左上8画素、右下9画素
OPT_FULL_SEARCH_12	3	左上11画素、右下12画素
OPT_FULL_SEARCH_15	4	左上14画素、右下15画素

#### tempsize

テンプレートサイズ情報の構造体(40以下で設定)

#### variance

フロー計測実行点を選定する閾値

#### thr\_c\_value

中心での差分累積の評価値に対する閾値(0 ~ 65535)

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了 (検知した物体の数)
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	フロー計測探索範囲モード設定範囲外
2 1	テンプレートサイズ設定範囲外 (41以上)
2 2	選定しきい値設定範囲外
2 3	差分累積の評価しきい値設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

**詳細情報****フロー計測実行点を選定する閾値の設定方法 1 (テンプレートの分散値を使用)**

フロー計測において、計測に使用されるテンプレートがはっきりとしたパターンを持っていない場合は、計測されたフローは信頼できないフローである。本コマンドでは、信頼できないフローに対して始めから計測を行わないこととして処理速度の高速化を図っている。

信頼できるフローが計測されるパターンか否かは、取り扱う画像に大きく依存しているためパラメータで指定できるようにしている。本コマンドでは、テンプレートの分散値を用いている (分散値が大きい-->はっきりしたパターンが存在-->信頼性が高い)。

本コマンドによるフロー計測では、variance 以上の値を持ったテンプレートのみフローの計測を行う。つまり、

$$\text{variance} = < \text{分散値} \times \text{テンプレート画素数}$$

を満たした計測位置のみフロー計測を実行。

variance は、上記の式を参考にして、計測を行う最小分散値  $\times$  テンプレート画素数で算出する。

### フロー計測実行点を選定する閾値の設定方法 2 (計測位置での変化量を使用)

フロー計測を行う場合、計測位置において画像に変化がある場合のみ対象が動いているものと考えて、計測を実行することが望ましい。これは、例え対象が動いていたとしても、計測位置での変化が非常に小さければ計測されたフローの信頼性は低いと考えられるからです(ノイズによって、動いているのか止まっているのか判断できないため)。信頼性の低いフローを計測するのは無駄が多いため、計算量の削減を目的としてある一定量の変化があった場合のみ計測を行うこととします。

全ての場合について計測を行う場合

thr\_c\_value =0

計測位置で thr 以上の変化があったときのみ計測を行う場合

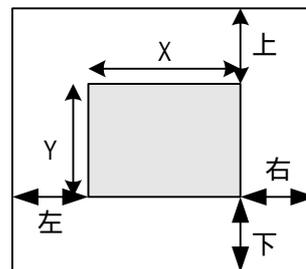
thr\_c\_value =thr

上記の二つの閾値によって、信頼性の低いフローの計測はできるだけ行わないほうが賢明です。計測を行わない箇所でのフローは、0 となります。

IPFlowSize のフォーマットを以下に示します。

```
typedef struct {
    int X ; /* テンプレート領域X成分 */
    int Y ; /* テンプレート領域Y成分 */
}IPFlowSize;
```

フロー計測の探索領域



## IP\_OptFlow

## オプティカルフロー実行

## 機能

フロー計測を行うための参照画面と計測対象であるカレント画面に対し、オプティカルフローを実行します。

## コーディング

```
int
IP_OptFlow(
  DEVID
  int
  int
  int
  IPFlowPoint
)
```

```
devID ,
ImgOld ,
ImgCurrent ,
num ,
*Tbl
```

## コメント

デバイス I D  
参照画面  
カレント画面  
登録点数  
フロー計測結果テーブル

## パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

ImgOld

フロー計測に使用する参照画面（旧画面）

ImgCurrent

フロー計測に使用するカレント画面（新画面）

num

IPFlowPoint 登録点数

\*Tbl

フロー計測結果テーブル  
(但し、結果格納テーブルを十分に取る必要があります Tbl[xx] : xx >= num+1 )

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外（不当画面番号エラー）
2 2	登録点数設定範囲外
-	エラーリセットコマンド（ClearIPError）未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

本コマンドによるオプティカルフロー計測は、ブロックマッチングを使用したフロー計測を行っています。ブロックマッチングの際のテンプレートサイズ、探索領域は SetOptFlowMode() コマンドを用いて設定して下さい。

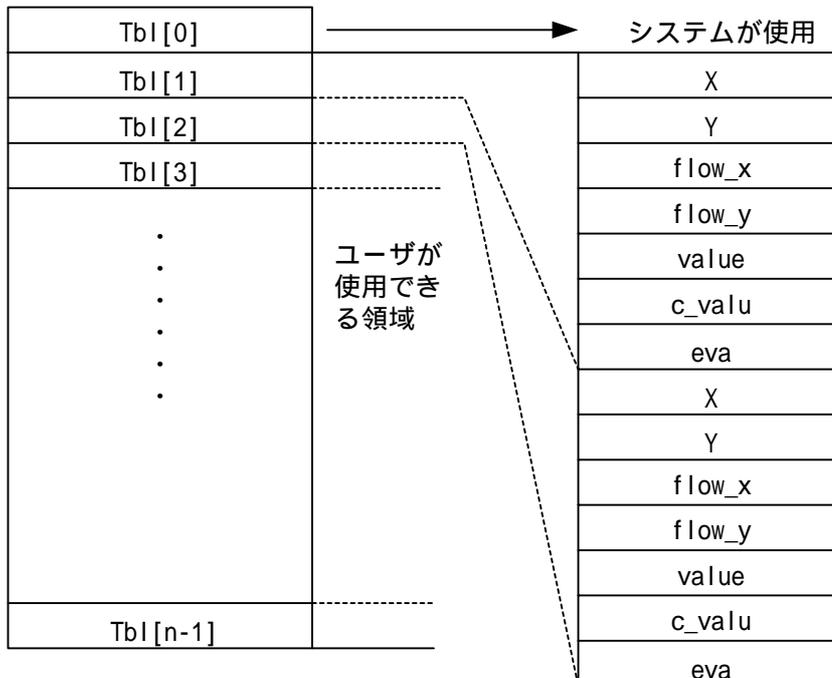
本コマンドでは、参照画面の一部をテンプレートとして、カレント画面のどの位置にテンプレートが移動したかを計測することによってフロー計測を行っています(ブロックマッチング)。以下の手順を参考にして、フロー計測を行って下さい。

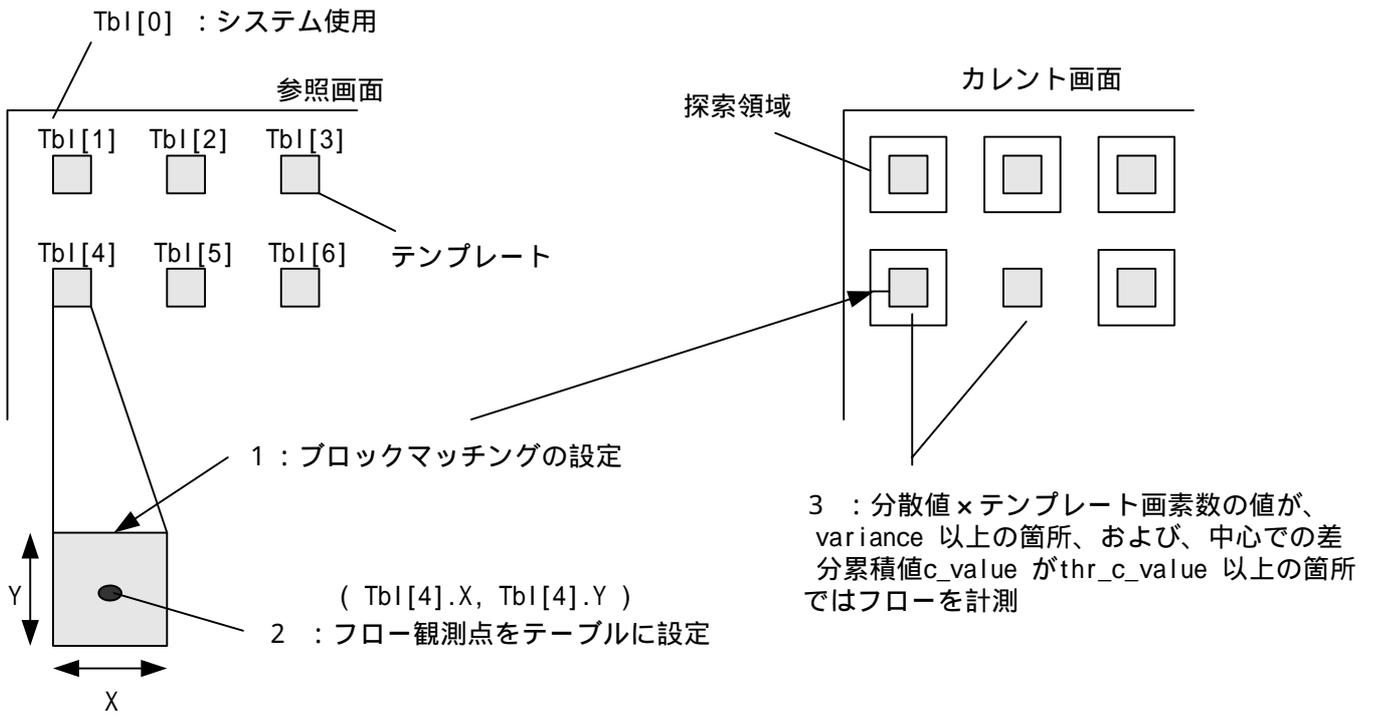
- 1 : まず始めに、SetOptFlowMode() コマンドを使用してモード設定を行います。
- 2 : つぎに、フロー計測を行う点を IPFlowPoint で宣言したテーブルに登録(X、Yのみ登録)します。このとき、X、Y座標は画面相対で指定して下さい。また、配列の先頭は、システムが使用しますので、二番目以降を使用するようにして下さい。
- 3 : 参照画面、カレント画面を指定して IP\_OptFlow() を実行して下さい。

IPFlowPoint のフォーマットを以下に示します。

```
typedef struct {
    int X; /* 第 n 番目の計測位置でのX 座標：画面相対アドレス（登録値）*/
    int Y; /* 第 n 番目の計測位置でのY 座標：画面相対アドレス（登録値）*/
    int flow_x; /* 第 n 番目の計測フローベクトル（X成分） */
    int flow_y; /* 第 n 番目の計測フローベクトル（Y成分） */
    int value; /* 第 n 番目の計測された位置での差分累積値 */
    int c_value; /* 第 n 番目の中心位置での差分累積値 */
    int eva; /* 第 n 番目の計測評価値（分散値×テンプレート画素数） */
} IPFlowPoint;
```

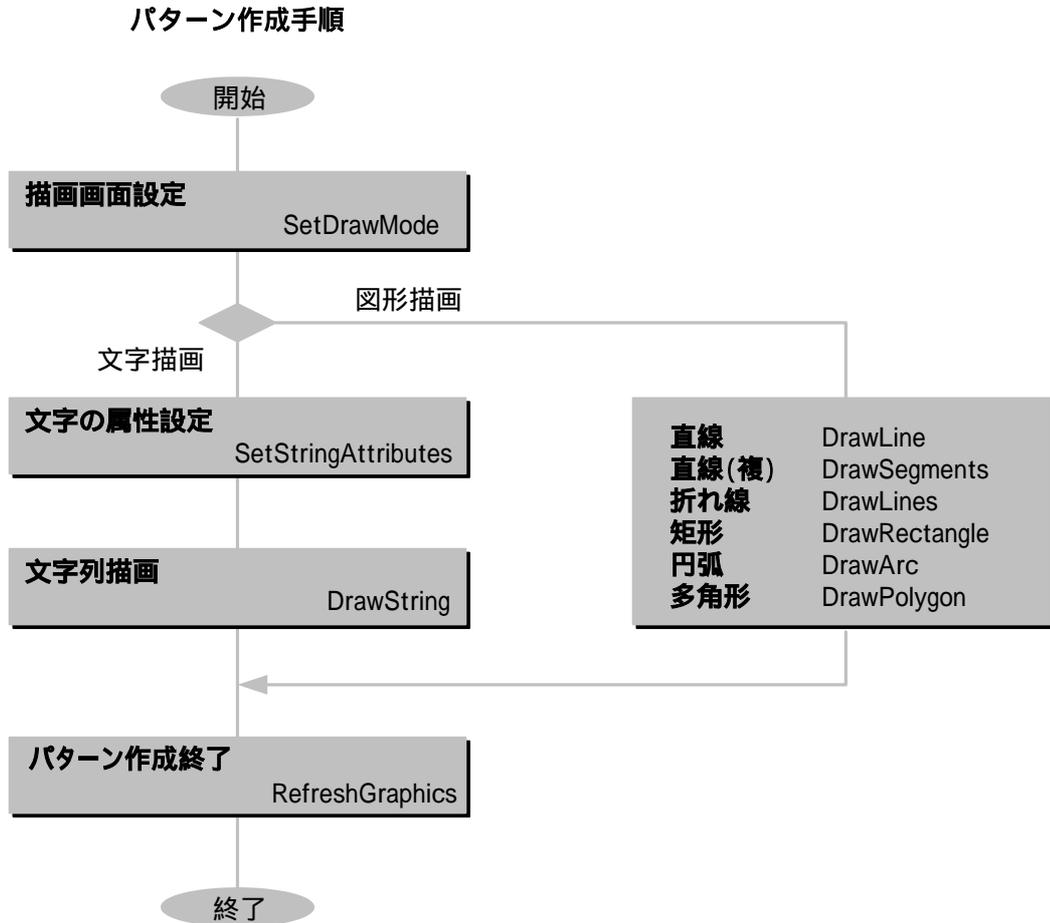
注意：配列の一番始めはシステムが使用しますので、二番目の配列からデータとして使用して下さい。





## 52. グラフィックス(IP5000互換)コマンド

グラフィックスコマンドでは、文字列、図形描画等のパターン作成ができます。グラフィックスは、下記手順のフローに従って実行してください。なお、グラフィックスコマンドでの座標系はウィンドウ相対であり、ウィンドウの始点が座標0になります。



# SetDrawMode

## 描画面面設定

### 機能

パターン作成時の画面番号、描画カラーを設定します。

### コーディング

```
int
SetDrawMode(
    DEVID          devID ,
    int            lmgID ,
    enum IPX_DRAW_MODE mode ,
    enum IPX_DRAW_COLOR color
)

```

### コメント

デバイスID  
画面番号  
描画モード  
描画カラー

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### lmgID

ソース画面の画面番号。

#### mode

描画モード

描画モード	対応定数	内容
DRAW_DIRECT	0	上書き
DRAW_EOR	1	元の画素とEORして描画

#### color

描画カラー

描画カラー	対応定数	内容
COLOR_WHITE	0	白
COLOR_BLACK	1	黒

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0	画面空き領域、または範囲外 (不当画面番号エラー)
2 1	描画モード設定値範囲外
2 2	描画カラー設定値範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# SetStringAttributes

## 文字の属性設定

### 機能

描画する文字のサイズ、文字間隔、行間隔を設定します。

### コーディング

```
int
SetStringAttributes(
    DEVID
    enum IPX_STRING_SIZE
    int
    int
)
```

```
devID ,
size ,
C_Interval ,
L_Interval
```

### コメント

```
デバイスID
文字サイズ
文字間隔
行間隔
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### size

文字サイズ

描画モード	対応定数	処理内容
STRING_16x16	0	標準サイズ(16×16)
STRING_32x32	1	4倍サイズ(32×32)
STRING_48x48	2	9倍サイズ(48×48)
STRING_64x64	3	16倍サイズ(64×64)

#### C\_Interval

文字間隔 ( 0 ~ 128 )

#### L\_Interval

行間隔 ( 0 ~ 128 )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	文字サイズ設定値範囲外
2 1	文字間隔設定値範囲外
2 2	行間隔設定値範囲外
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# RefreshGraphics

## パターン作成終了

### 機能

パターン作成時に、一時停止していた処理を再開します。また、パターン作成した画面を表示している場合は、自動的に再表示します。

### コーディング

```
int  
RefreshGraphics(  
    DEVID          devID  
)
```

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# DrawString

## 文字列描画

### 機能

属性設定コマンドで設定した内容にもとづき、文字列を書き込みます。  
 なお、ユーザ作成の文字列テーブルは、ASCII 文字またはシフトJIS 文字で作成してください。また、文字列終了コードにはNULL を入れてください。

### コーディング

```
int
DrawString(
  DEVID
  int
  int
  char
)
```

```
devID ,
x ,
y ,
*str
```

### コメント

```
デバイスID
始点X座標
始点Y座標
文字列のポインタ
```

### パラメータ

**devID**  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**x**  
 始点X座標

**y**  
 始点Y座標

**\*str**  
 文字列のポインタ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1 ~ n	正常終了。画像メモリへ描画した行数。
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	始点X座標設定エラー
2 1	始点Y座標設定エラー
4 4	座標設定エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# DrawLine

## 直線描画

### 機能

直線を描画します。

### コーディング

```
int
DrawLine(
    DEVID
    int
    int
    int
    int
)
```

```
devID ,
sx ,
sy ,
ex ,
ey
```

### コメント

```
デバイス I D
始点 X 座標
始点 Y 座標
終点 X 座標
終点 Y 座標
```

### パラメータ

devID

デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

sx

始点 X 座標

sy

始点 Y 座標

ex

終点 X 座標

ey

終点 Y 座標

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	始点 X 座標設定エラー
2 1	始点 Y 座標設定エラー
2 2	終点 X 座標設定エラー
2 3	終点 Y 座標設定エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# DrawSegments

## 直線描画(複数)

### 機能

複数の直線を描画します。

### コーディング

```
int
DrawSegments(
    DEVID
    IPX_SEGMENT
    int
)
```

```
devID ,
*segments ,
ns
```

### コメント

デバイスID  
線分設定テーブルのポインタ  
描画する線分数

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*segments

線分設定テーブルのポインタ

ns

描画する線分数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 1	描画線分数が0以下
4 4	座標設定エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

線分設定テーブルのフォーマットおよび内容を、以下に示します。

ipxsys.h で宣言しています。

```
typedef struct {
    int sx, sy ; 線分の始点座標
    int ex, ey ; 線分の終点座標
} IPX_SEGMENT ;
```

# DrawLines

## 折れ線描画

### 機能

折れ線を描画します。

### コーディング

```
int
DrawLines(
    DEVID
    IPX_POINT
    int
)
```

```
devID ,
*points ,
np
```

### コメント

デバイスID  
線分設定テーブルのポインタ  
座標数

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*points

線分設定テーブルのポインタ

np

座標数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 1	設定座標数が1以下
4 4	座標設定エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

線分設定テーブルのフォーマットおよび内容を、以下に示します。

ipxsys.h で宣言しています。

```
typedef struct {
    int x, y ; 線分の始点および終点座標 (ウィンドウ相対)
} IPX_POINT ;
```

# DrawRectangle

## 矩形描画

### 機能

矩形を描画します。

### コーディング

```
int
DrawRectangle(
    DEVID          devID ,
    int            x ,
    int            y ,
    int            width ,
    int            height
)
```

### コメント

デバイス I D  
 始点 X 座標  
 始点 Y 座標  
 幅  
 高さ

### パラメータ

**devID**  
 デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**x**  
 始点 X 座標

**y**  
 始点 Y 座標

**width**  
 幅

**height**  
 高さ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	始点 X 座標設定エラー
2 1	始点 Y 座標設定エラー
2 2	幅設定エラー
2 3	高さ設定エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

幅、高さを 0 に指定した場合は、始点座標へ 1 点を描画します。

## DrawPolygon

## 多角形描画

### 機能

多角形を描画します。そのとき、始点座標と終点座標が一致しないときは自動的に始点座標と終点座標を結びます。

### コーディング

```
int
DrawPolygon(
    DEVID
    IPX_POINT
    int
)
```

```
devID ,
*points ,
np
```

### コメント

デバイスID  
線分設定テーブルのポインタ  
座標数

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*points

線分設定テーブルのポインタ

np

座標数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 1	設定座標数が1以下
4 4	座標設定エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

線分設定テーブルのフォーマットおよび内容は、DrawLines の備考の欄を参照してください。

# DrawArc

## 円弧描画

### 機能

円弧を描画します。

### コーディング

```

int
DrawArc(
    DEVID
    int
    int
    int
    int
    int
    int
)
    devid ,
    x ,
    y ,
    width ,
    height ,
    angle1 ,
    angle2

```

### コメント

デバイス I D  
 始点 X 座標  
 始点 Y 座標  
 幅  
 高さ  
 弧の開始角度  
 弧の存在する角度

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**x**  
始点 X 座標

**y**  
始点 Y 座標

**width**  
幅

**height**  
高さ

**angle1**  
弧の開始角度 (°)

**angle2**  
弧の存在する角度 (°)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

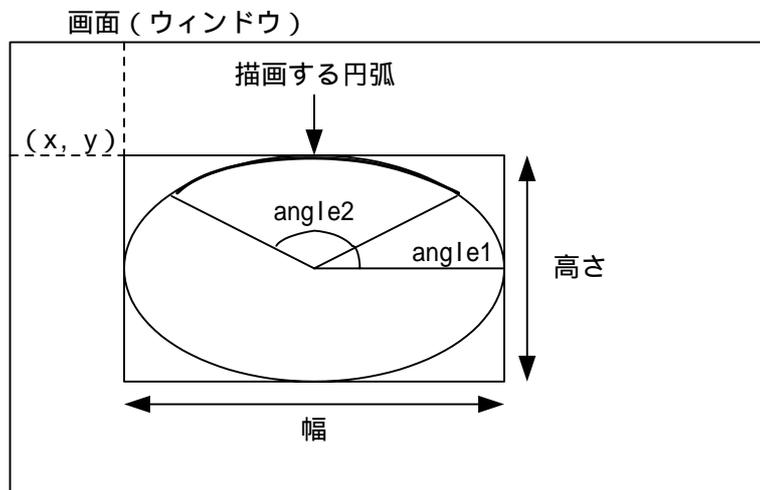
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	X座標設定エラー
2 1	Y座標設定エラー
2 2	幅設定エラー(4以下)
2 3	高さ設定エラー(4以下)
2 4	弧の開始角度設定エラー
2 5	弧の存在する角度設定エラー
4 5	パラメータの設定条件違反
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

パラメータの設定内容を、以下に示します。



# FillRectangle

## 矩形描画(塗りつぶし)

### 機能

矩形(塗りつぶし)を描画します。

### コーディング

```

int
FillRectangle(
    DEVID          devid ,
    int            x ,
    int            y ,
    int            width ,
    int            height
)

```

### コメント

デバイスID  
 始点X座標  
 始点Y座標  
 幅  
 高さ

### パラメータ

**devID**  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**x**  
 始点X座標

**y**  
 始点Y座標

**width**  
 幅

**height**  
 高さ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 0	始点X座標設定エラー
2 1	始点Y座標設定エラー
2 2	幅設定エラー
2 3	高さ設定エラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

幅、高さを0に指定した場合は、始点座標へ1点を描画します。

# FillPolygon

## 多角形描画(塗りつぶし)

### 機能

多角形(塗りつぶし)を描画します。そのとき、始点座標と終点座標が一致しないときは自動的に始点座標と終点座標を結びます。

### コーディング

```
int
FillPolygon(
    DEVID
    IPX_POINT
    int
)
```

```
devID ,
*points ,
np
```

### コメント

デバイスID  
線分設定テーブルのポインタ  
座標数

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

\*points

線分設定テーブルのポインタ

np

座標数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
2 1	設定座標数が1以下
4 4	座標設定エラー
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

線分設定テーブルのフォーマットおよび内容は、DrawLines()の備考の欄を参照してください。

## IP\_SmoothFLText

## 平滑化フィルタ(拡張版)

### 機能

濃淡画像をパラメータ指定の平滑化手法で平滑化します。なお、同一映像に対して最大16回まで連続して実行処理できます。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_SmoothFLText(
    DEVID
    int
    int
    enum IPMSmoothMethod
    int
)
```

```
devID ,
imgSrc ,
imgDst ,
method ,
n
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
平滑化手法
繰り返し回数
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### imgSrc

ソース画面の画面番号

#### imgDst

デスティネーション画面の画面番号

#### method

平滑化手法

平滑化手法	内容
SMOOTHNOOP	コピー
SMOOTH1	平滑化フィルタ1 : 詳細情報参照
SMOOTH2	平滑化フィルタ2 : 詳細情報参照
SMOOTH3	平滑化フィルタ3 : 詳細情報参照
SMOOTH4	平滑化フィルタ4 : 詳細情報参照
MEDIAN8	メディアンフィルタ(8連結)
MEDIAN4	メディアンフィルタ(4連結)
MEDIAN84	メディアンフィルタ(8 4 8 ...連結)
MEDIAN48	メディアンフィルタ(4 8 4 ...連結)
SENSUP2	2倍感度UPフィルタ: 詳細情報参照
SENSUP4	4倍感度UPフィルタ: 詳細情報参照
MAX8	局所最大フィルタ(8連結)
MIN8	局所最小フィルタ(8連結)

#### n

繰り返し回数 ( 1 ~ 16 )

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 2	平滑化手法設定範囲外
2 3	繰り返し回数設定範囲外
4 8	動的メモリ確保エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

スムージングフィルタ・感度UPフィルタの荷重係数

平滑化フィルタ 1

7	7	7
7	8	7
7	7	7

6 ビットシフトダウン

平滑化フィルタ 2

6	7	6
7	12	7
6	7	6

6 ビットシフトダウン

平滑化フィルタ 3

2	4	2
4	8	4
2	4	2

5 ビットシフトダウン

平滑化フィルタ 4

2	3	2
3	12	3
2	3	2

5 ビットシフトダウン

2 倍感度UP フィルタ

1	2	1
2	4	2
1	2	1

3 ビットシフトダウン

4 倍感度UP フィルタ

1	2	1
2	4	2
1	2	1

2 ビットシフトダウン

デスティネーション画面に画像処理の途中結果を格納しているため、デスティネーション画面を表示画面にしていると、映像がちらつく場合があります。

# IP\_EdgeFLTAbsExt

## 輪郭強調フィルタ・絶対値(拡張版)

### 機能

濃淡画像をパラメータ指定の輪郭強調手法で、濃淡画像の輪郭強調を行います。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_EdgeFLTAbsExt (
    DEVID
    int
    int
    enum IPMEdgeMethod
    float
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
method ,
gain ,
lutno
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
輪郭強調手法
輪郭強調ゲイン
拡張濃度変換テーブル登録番号
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面の画面番号

#### ImgDst

デスティネーション画面の画面番号

#### method

輪郭強調手法

輪郭強調手法	内容
EDGENOOP	コピー
EDGEMAXVH	$\max( \text{水平エッジ} ,  \text{垂直エッジ} )$ : 詳細情報(A), (B)参照
EDGEADDVH	$ \text{水平エッジ}  +  \text{垂直エッジ} $ : 詳細情報(A), (B)参照
EDGEV	$ \text{垂直エッジ} $ : 詳細情報(A), (B)参照
EDGEH	$ \text{水平エッジ} $ : 詳細情報(A), (B)参照
EDGE_SOBEL_DASH	ソーベル' : 詳細情報(A), (B)参照
EDGE_ZEROCROSS8_ALL	ゼロクロッシング1 : 詳細情報(C)参照
EDGE_ZEROCROSS4_ALL	ゼロクロッシング2 : 詳細情報(C)参照
EDGE_ZEROCROSS8_SEQ	ゼロクロッシング3 : 詳細情報(D)参照
EDGE_ZEROCROSS4_SEQ	ゼロクロッシング4 : 詳細情報(D)参照
EDGE_MINMAX	Min,Max フィルタによるエッジ抽出 : 詳細情報(B), (E)参照
EDGE_SOBEL	$ \text{水平エッジ}  +  \text{垂直エッジ} $ : 詳細情報(A), (B)参照

**gain**

- エッジ出力時 - 輪郭強調ゲイン (0.5 ~ 0.7)  
 ゼロクロッシング時 - 乗算のダウンシフト量 (0 ~ 8)

**lutno**

拡張濃度変換テーブル登録番号

- 0 : NOOP  
 1 ~ 8 : 濃度適応補正処理

IP\_RegisterLUT(int lutno, CNVLUT \*lut)コマンドで設定された拡張濃度変換テーブル登録番号 (lutno) で、濃度変換を実行します。但し、以下の条件時には、濃度適応は行いません。

- ・拡張濃度変換テーブル登録番号が0の場合
- ・輪郭強調手法が「EDGENOOP」の場合

**リターン値**

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

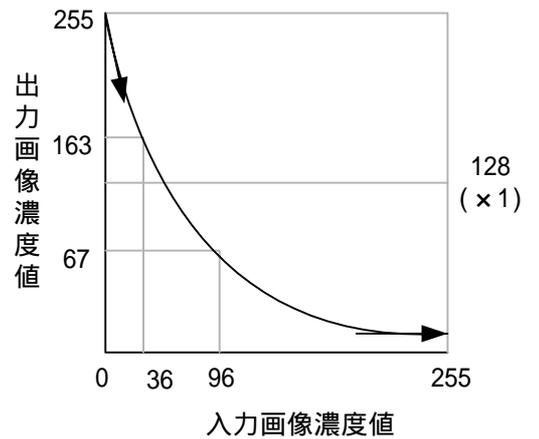
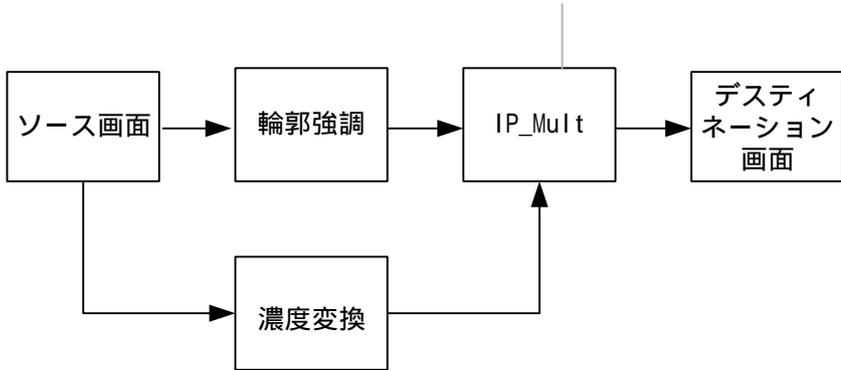
エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 2	平滑化手法設定範囲外
2 3	繰り返し回数設定範囲外
2 4	拡張濃度変換テーブル登録番号設定範囲外
4 8	動的メモリ確保エラー
-	エラーリセットコマンド (ClearIPErrors) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

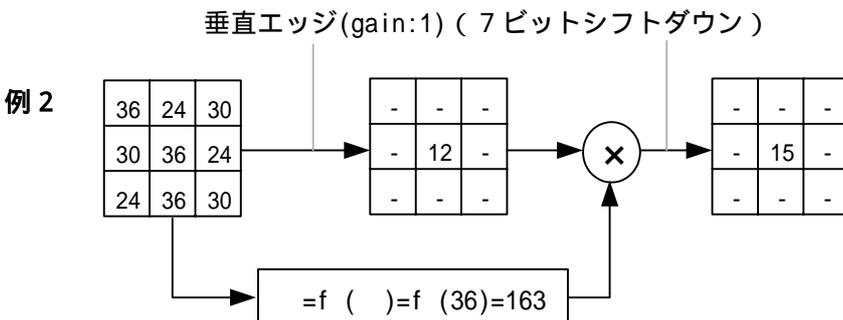
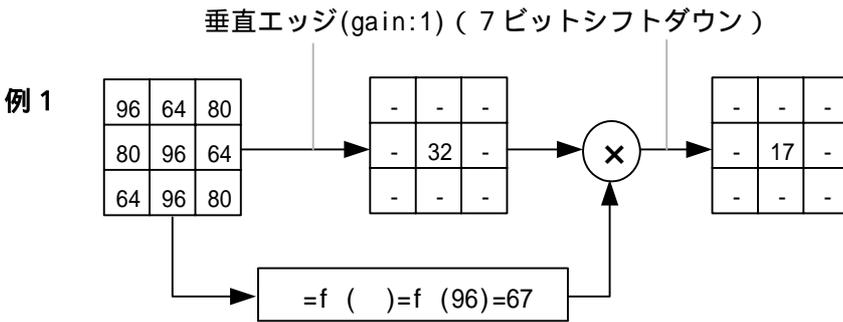
**詳細情報**

[A] 濃度適応処理は、下図左の処理を行います。下図右の濃度変換テーブルを登録・指定して濃度適応処理を実行させた場合、暗い部分のエッジは明るくなり・明るい部分のエッジは暗くなります。

乗算後、7ビットシフトダウン(÷128)し、濃度補正の変換倍率を正規化します。



startvalue:255 startgrad:-3.0  
endvalue :16 endgrad :-0.025

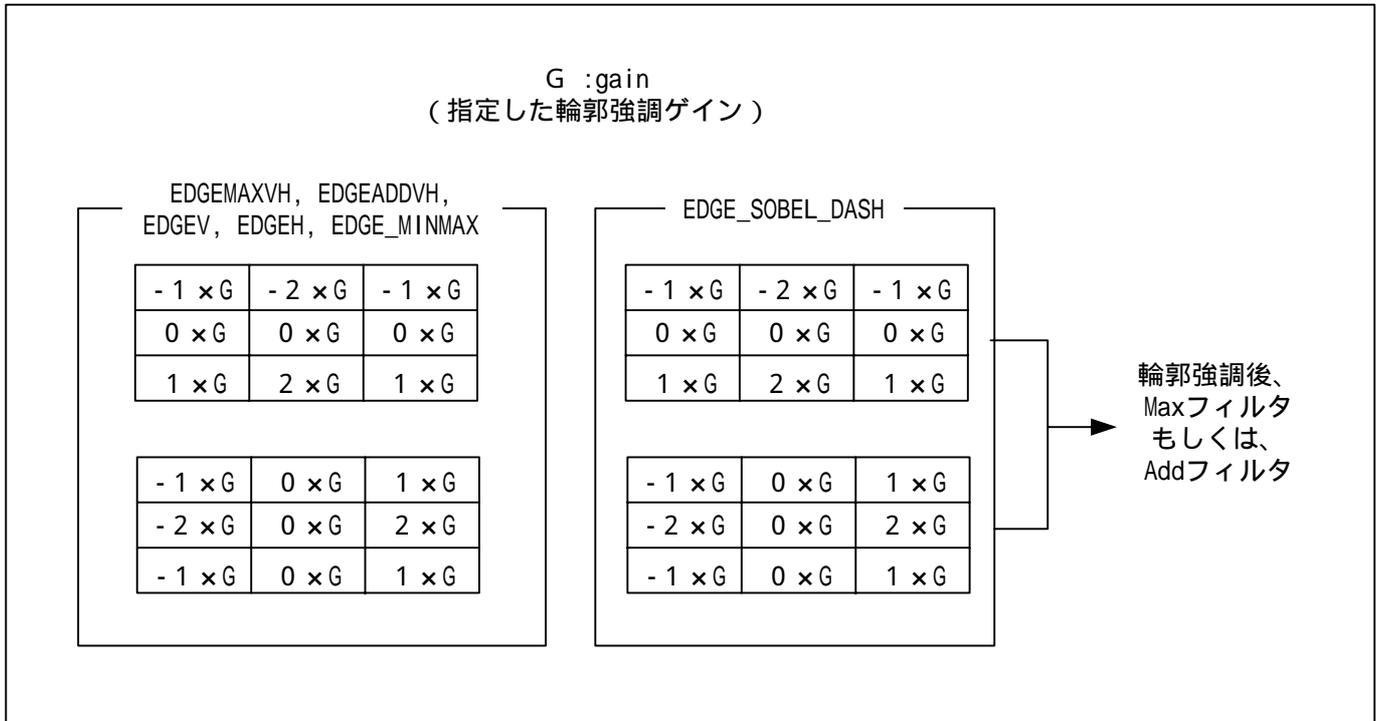


左図の例 1、2 の様に濃度補正を行う事で、同一パターンで出力される値が均一化されます。  
 濃度変換は、上図のテーブルを指定します。  
 ソース画面のサンプル  
 垂直エッジの濃度値  
 濃度変換 =f ( ) (右上図)  
 濃度適応処理結果 (四捨五入されます)

## [B] 係数テーブル

輪郭強調フィルタで、EDGEMAXVH, EDGEADDVH, Sobel, Sobel'フィルタの係数テーブルは次の通りです。

## 係数テーブルの構造



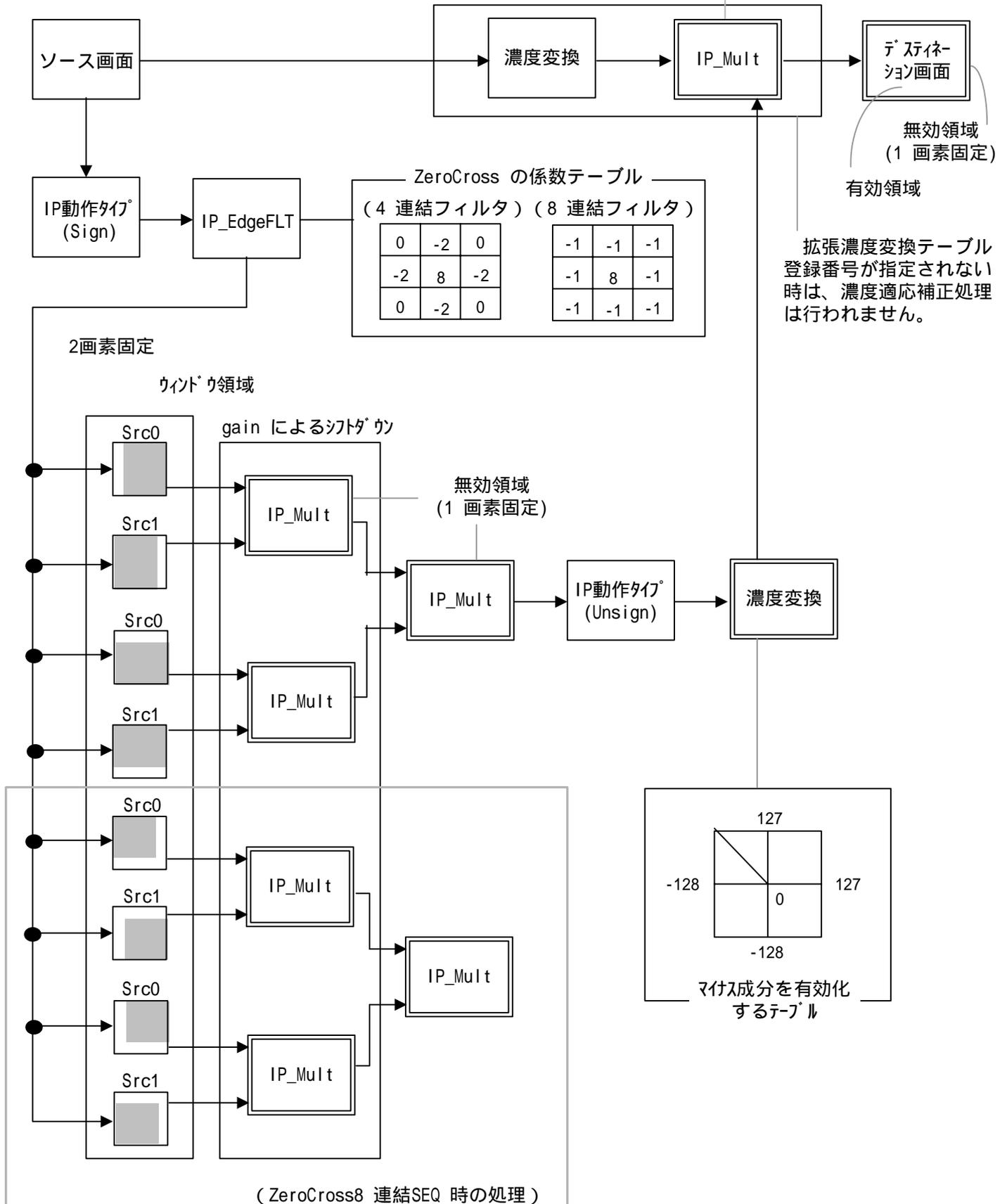


[D] ZeroCross8,4SEQ

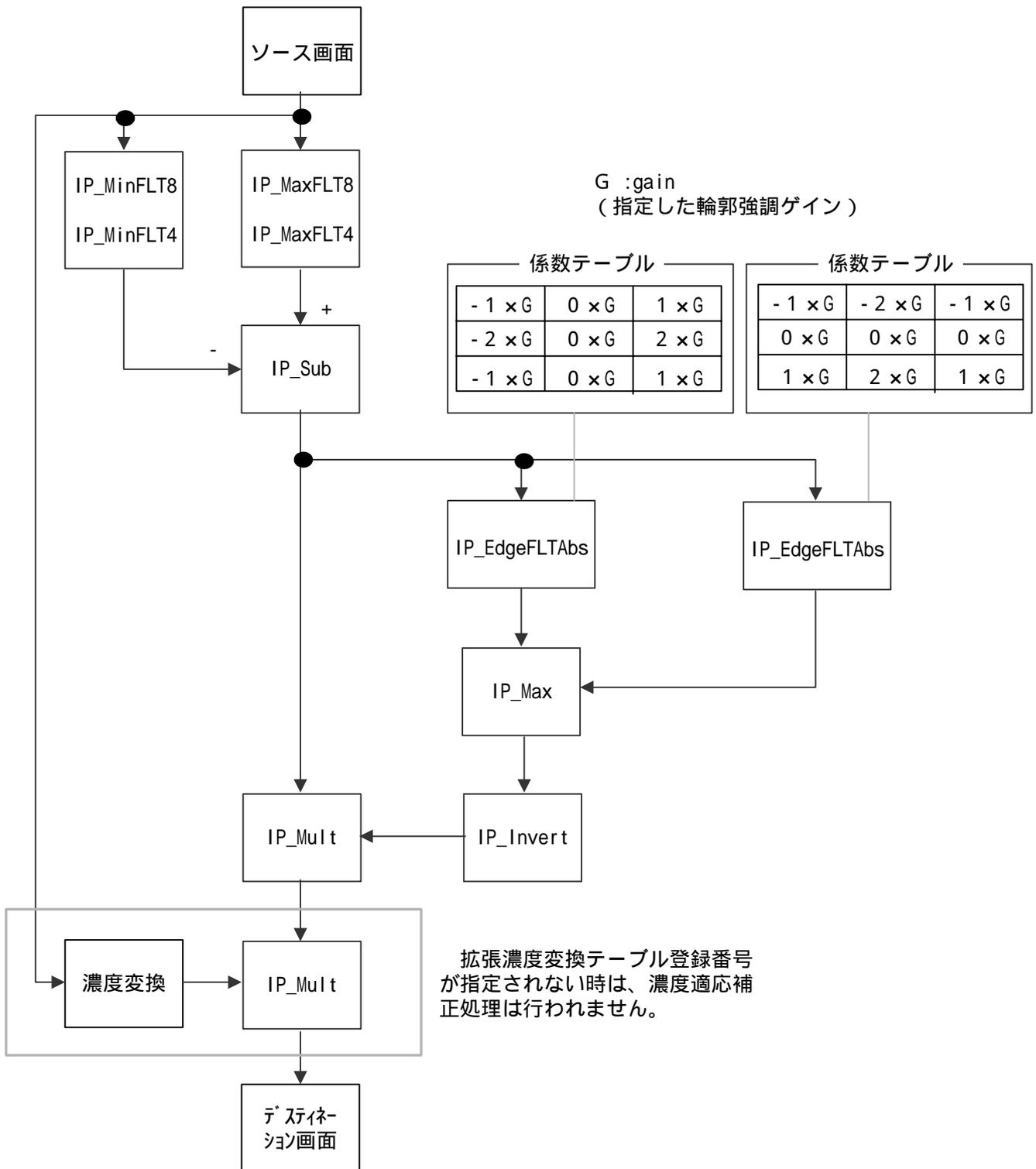
下記の係数テーブルで取得した微分符号付き画像に対し、座標をずらし乗算した画像で、負の値(0をクロスする)を抽出します。乗算結果の値が小さいほど、エッジ濃度は大きくなります。

ウィンドウサイズが、縦、もしくは横が1ラインの時は、NOOP に切り換えます。

乗算後、7ビットシフトダウン(÷128)し、濃度補正の変換倍率を正規化します。



[ E ] Min,Max フィルタによるエッジ抽出



拡張濃度変換テーブル登録番号が指定されない時は、濃度適応補正処理は行われません。

## IP\_Sobel

## 輪郭強調(ソーベル)

### 機能

ソース画面からソーベルによって輪郭強調を行い、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Sobel(
  DEVID
  int
  int
  int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
gain
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
ゲイン
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面番号

#### ImgDst

デスティネーション画面番号

#### gain

ゲイン ( 1 ~ 8 )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

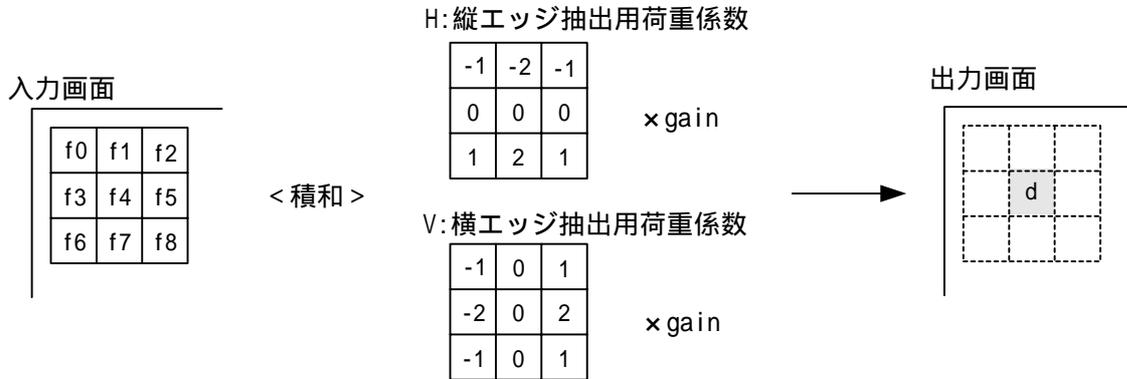
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2	転送モード指定範囲外
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

輪郭強調では、入力画面の指定領域内の全画素に対して、注目画素を中心とした  $3 \times 3$  近傍の局所領域で与えられた荷重係数との積和演算を行います。また、対象領域の周辺 ( $3 \times 3$  近傍に領域外が含まれる部分) 処理の場合は 0 を出力します。



$$d = |f_0 \times (-\text{gain}) + f_1 \times (-2) \times \text{gain} + f_2 \times (-\text{gain}) + f_6 \times \text{gain} + f_7 \times 2 \times \text{gain} + f_8 \times \text{gain}| \\ + |f_0 \times (-\text{gain}) + f_3 \times (-2) \times \text{gain} + f_6 \times (-\text{gain}) + f_2 \times \text{gain} + f_5 \times 2 \times \text{gain} + f_8 \times \text{gain}|$$

## IP\_SobelBinarize

## 輪郭強調2値化(ソーベル2値化)

### 機能

濃淡画像からソーベルによって輪郭強調を行い、その結果をしきい値によって2値化します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_SobelBinarize(
    DEVID
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thr
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
2値化しきい値
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面番号

#### ImgDst

デスティネーション画面番号

#### thr

2値化しきい値

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2	転送モード指定範囲外
-	エラーリセットコマンド ( ClearIPError ) 未発行

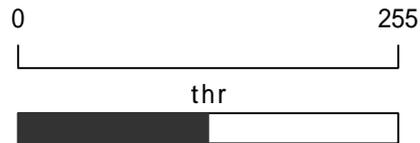
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

輪郭強調では、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域で与えられた荷重係数との積和演算を行います。その結果をパラメータthrで指定するしきい値以上の値を白、それより小さい値を黒で2値化します。また、対象領域の周辺(3×3近傍に領域外が含まれる部分)処理の場合は0を出力します。



$$d = |f_0 \times (-1) + f_1 \times (-2) + f_2 \times (-1) + f_6 + f_7 \times 2 + f_8| \\ + |f_0 \times (-1) + f_3 \times (-2) + f_6 \times (-1) + f_2 + f_5 \times 2 + f_8|$$



# IP\_RegisterLUT

## 濃度変換テーブルの登録

### 機能

IP\_EdgeFLTAbsExt() コマンドの濃度変換データを設定します。  
また、ここで設定したLUTのデータは、ラベリングコマンドを実行すると書き換えられますので注意して下さい。

### コーディング

```
int
IP_RegisterLUT(
  DEVID      devID ,
  int        lutno ,
  int        *lut
)
```

### コメント

デバイスID  
濃度変換テーブル登録番号(1~8)  
濃度変換データ格納テーブル

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**lutno**  
濃度変換テーブル登録番号(1~8)

**\*lut**  
濃度変換データ格納テーブル(32bit x 256)のポインタ。このテーブルに濃度変換のデータを設定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

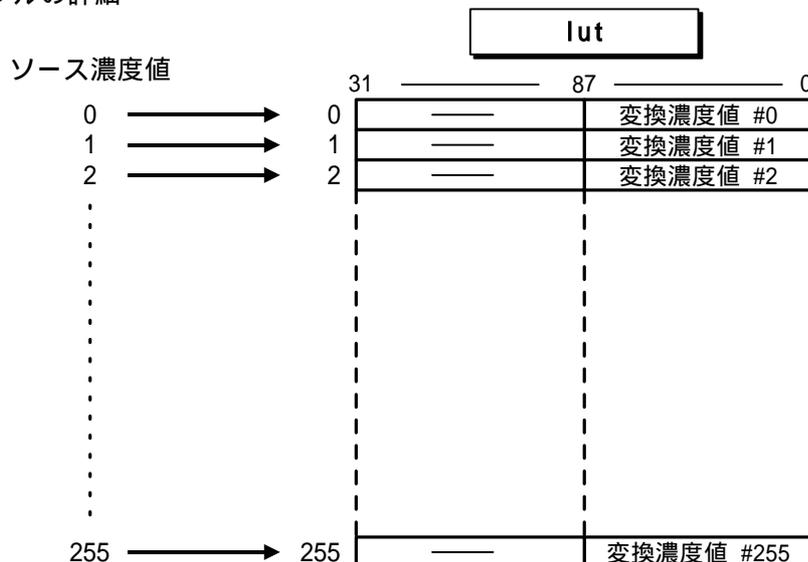
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0	濃度変換テーブル登録番号設定範囲外
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

濃度変換テーブルの詳細



# IP\_Prewitt

## 輪郭強調(プレビット)

### 機能

ソース画面からプレビットにより輪郭強調を行い、デスティネーション画面に出力します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_Prewitt(
    DEVID
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
gain
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
ゲイン
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面番号

#### ImgDst

デスティネーション画面番号

#### gain

ゲイン ( 1 ~ 8 )。通常は「1」を指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

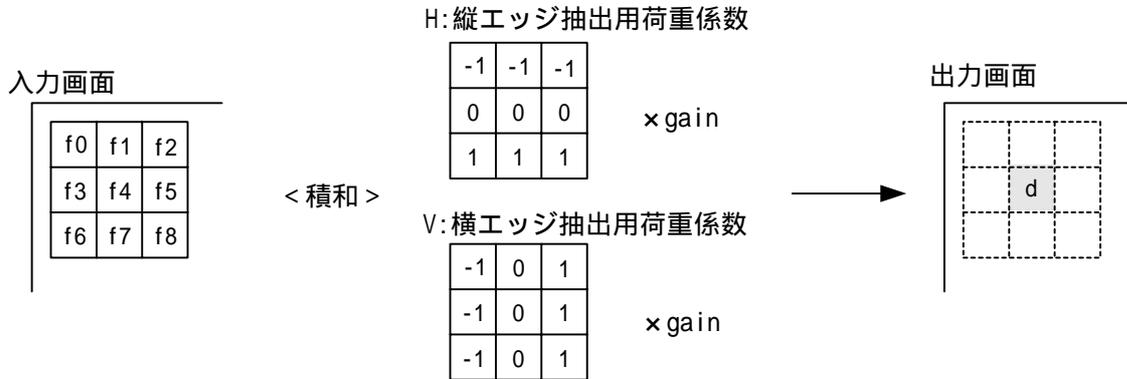
異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 ( 不当画面番号エラー )
2 2	転送モード指定範囲外
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

輪郭強調では、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域で与えられた荷重係数との積和演算を行います。また、対象領域の周辺(3×3近傍に領域外が含まれる部分)処理の場合は0を出力します。



$$d = |f0 \times (-gain) + f1 \times (-2) \times gain + f2 \times (-gain) + f6 \times gain + f7 \times 2 \times gain + f8 \times gain| \\ + |f0 \times (-gain) + f3 \times (-2) \times gain + f6 \times (-gain) + f2 \times gain + f5 \times 2 \times gain + f8 \times gain|$$

## IP\_PrewittBinarize

## 輪郭強調2値化(プレビット2値化)

### 機能

濃淡画像からプレビットにより輪郭強調を行い、その結果を2値化します。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_PrewittBinarize(
    DEVID
    int
    int
    int
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thr
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
2値化しきい値
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画面番号

#### ImgDst

デスティネーション画面番号

#### thr

2値化しきい値

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 2	転送モード指定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

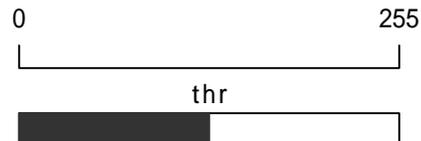
詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

輪郭強調では、入力画面の指定領域内の全画素に対して、注目画素を中心とした3×3近傍の局所領域で与えられた荷重係数との積和演算を行います。その結果をパラメータthrで指定するしきい値以上の値を白、それより小さい値を黒で2値化します。また、対象領域の周辺(3×3近傍に領域外が含まれる部分)処理の場合は0を出力します。



$$d = |f_0 \times (-1) + f_1 \times (-2) + f_2 \times (-1) + f_6 + f_7 \times 2 + f_8| \\ + |f_0 \times (-1) + f_3 \times (-2) + f_6 \times (-1) + f_2 + f_5 \times 2 + f_8|$$



# IP\_Label4byRL

## ランレングス・ラベリング(4連結)

### 機能

ソース画面(2値)に対し、4連結でラベリング処理と基本特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

### コーディング

```
int
IP_Label4byRL(
    DEVID
    int
    int
    enum IP_Label_opt
    IPLabelBasicTbl
    enum IP_Label_opt2
)
```

```
devID ,
ImgSrc ,
ImgDst ,
opt ,
*BasicTbl ,
opt2
```

### コメント

```
デバイスID
ソース画像
デスティネーション画像
ラベリング処理オプション1
基本特徴量テーブル
ラベリング処理オプション2
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画像

#### ImgDst

デスティネーション画像

#### opt

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

#### \*BasicTbl

基本特徴量テーブル。255個以上確保して下さい。

#### opt2

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	指定画面空き領域、または範囲外 (不当画面番号エラー)
22	オプション設定値範囲外
30	SRC0 ウィンドウ幅エラー (ウィンドウ幅は2以上にして下さい)
49	アクセス画面が不定画面
-	エラーリセットコマンド (ClearLError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は254個です。ラベルオーバーフローは、ラベル数が255個以上、仮ラベル数が512個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容を下記に示します。

ipxsys.h で宣言しています。

```
typedef struct {
    long          AREA;          /* 面積 */
    short         LEFT_UP_POINT_X; /* 最小X座標 */
    short         LEFT_UP_POINT_Y; /* 最小Y座標 */
    short         RIGHT_DOWN_POINT_X; /* 最大X座標 */
    short         RIGHT_DOWN_POINT_Y; /* 最大Y座標 */
    float         CENTER_of_GRAVITY_X; /* 重心X座標 */
    float         CENTER_of_GRAVITY_Y; /* 重心Y座標 */
} IPLabelBasicTbl
```

基本、拡張特徴量のテーブルはラベル番号に対応させています。0番目は不定で、1番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数(254)+1 確保しておく必要があります。

本コマンドで得られる各座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

```
MinX = BasicTbl[ラベル番号].LEFT_UP_POINT_X + sx
MinY = BasicTbl[ラベル番号].LEFT_UP_POINT_Y + sy
MaxX = BasicTbl[ラベル番号].RIGHT_DOWN_POINT_X + sx
MaxY = BasicTbl[ラベル番号].RIGHT_DOWN_POINT_Y + sy
Gx   = BasicTbl[ラベル番号].CENTER_of_GRAVITY_X + sx
Gy   = BasicTbl[ラベル番号].CENTER_of_GRAVITY_Y + sy
```

```
MinX   : 最小X座標
MinY   : 最小Y座標
MaxX   : 最大X座標
MaxY   : 最大Y座標
Gx     : 重心X座標
Gy     : 重心Y座標
sx, sy : SRC0_WIN(sx, sy)
```

となります。

# IP\_Label8byRL

## ランレングス・ラベリング(8連結)

### 機能

ソース画面(2値)に対し、8連結でラベリング処理と基本特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

### コーディング

```
int
IP_Label8byRL(
    DEVID
    int
    int
    enum IP_Label_opt
    IPLabelBasicTbl
    enum IP_Label_opt2
)
```

```
devID ,
ImgSrc ,
ImgDst ;
opt ,
*BasicTbl ,
opt2
```

### コメント

```
デバイスID
ソース画像
デスティネーション画像
ラベリング処理オプション1
基本特徴量テーブル
ラベリング処理オプション2
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画像

#### ImgDst

デスティネーション画像

#### opt

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

#### \*BasicTbl

基本特徴量テーブル。255個以上確保して下さい。

#### opt2

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2	オプション設定値範囲外
3 0	S R C 0 ウィンドウ幅エラー (ウィンドウ幅は2以上にして下さい)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は254個です。ラベルオーバーフローは、ラベル数が255個以上、仮ラベル数が512個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRL()コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0番目は不定で、1番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数(254)+1確保しておく必要があります。

本コマンドで得られる各座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。

詳細はIP\_Label4byRL()コマンドを参照して下さい。

**IP\_Label4byRLwithAreaFLT****ランレングス・面積フィルタ付きラベリング(4連結)****機能**

ソース画面(2値)に対し、4連結でラベリング処理を行い、指定しきい値で面積フィルタ処理後に、再度ラベル付けと基本特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

**コーディング**

```
int
IP_Label4byRLwithAreaFLT(
    DEVID
    int
    int
    int
    int
    enum IP_Label_opt
    IPLabelBasicTbl
    enum IP_Label_opt2
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thrmin ,
thrmax ,
opt ,
*BasicTbl ,
opt2
```

**コメント**

```
デバイスID
ソース画像
デスティネーション画像
面積フィルタしきい値
面積フィルタしきい値
ラベリング処理オプション1
基本特徴量テーブル
ラベリング処理オプション2
```

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画像

**ImgDst**

デスティネーション画像

**thrmin**

面積フィルタしきい値(最小値)

**thrmax**

面積フィルタしきい値(最大値)

**opt**

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

**\*BasicTbl**

基本特徴量テーブル。255個以上確保して下さい。

**opt2**

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2	オプション設定値範囲外
3 0	S R C 0 ウィンドウ幅エラー (ウィンドウ幅は 2 以上にして下さい)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は 2 5 4 個です。ラベルオーバーフローは、ラベル数が 2 5 5 個以上、仮ラベル数が 5 1 2 個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRL() コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0 番目は不定で、1 番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数 ( 2 5 4 ) + 1 確保しておく必要があります。

本コマンドで得られる各座標は SRC0\_WIN で設定されたウィンドウからの相対座標になります。

詳細は IP\_Label4byRL() コマンドを参照して下さい。

**IP\_Label18byRLwithAreaFLT****ランレングス・面積フィルタ付きラベリング(8連結)****機能**

ソース画面(2値)に対し、8連結でラベリング処理を行い、指定しきい値で面積フィルタ処理後に、再度ラベル付けと基本特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

**コーディング**

```
int
IP_Label18byRLwithAreaFLT(
    DEVID
    int
    int
    int
    int
    enum IP_Label_opt
    IPLabelBasicTbl
    enum IP_Label_opt2
)
```

```
devID ,
ImgSrc ,
ImgDst ,
thrmin ,
thrmax ,
opt ,
*BasicTbl ,
opt2
```

**コメント**

```
デバイスID
ソース画像
デスティネーション画像
面積フィルタしきい値
面積フィルタしきい値
ラベリング処理オプション1
基本特徴量テーブル
ラベリング処理オプション2
```

**パラメータ****devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**

ソース画像

**ImgDst**

デスティネーション画像

**thrmin**

面積フィルタしきい値(最小値)

**thrmax**

面積フィルタしきい値(最大値)

**opt**

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

**\*BasicTbl**

基本特徴量テーブル。255個以上確保して下さい。

**opt2**

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2	オプション設定値範囲外
3 0	S R C 0 ウィンドウ幅エラー (ウィンドウ幅は 2 以上にして下さい)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は 2 5 4 個です。ラベルオーバーフローは、ラベル数が 2 5 5 個以上、仮ラベル数が 5 1 2 個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRL() コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0 番目は不定で、1 番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数 ( 2 5 4 ) + 1 確保しておく必要があります。

本コマンドで得られる各座標は SRC0\_WIN で設定されたウィンドウからの相対座標になります。

詳細は IP\_Label4byRL() コマンドを参照して下さい。

## IP\_Label4byRLwithAreaFLSort

ランレングス・面積フィルタ付ソートラベリング(4連結)

### 機能

ソース画面(2値)に対し、4連結でラベリング処理を行い、指定しきい値で面積フィルタ処理後に、面積でソートして再度ラベル付けと基本特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

### コーディング

```
int
IP_Label4byRLwithAreaFLSort(
    DEVID          devID ,
    int             ImgSrc ,
    int             ImgDst ,
    int             thrmin ,
    int             thrmax ,
    enum IP_Label_opt opt ,
    enum IP_Label_Sort_opt Sort_opt ,
    IPLabelBasicTbl *BasicTbl ,
    enum IP_Label_opt2 opt2
)
```

### コメント

デバイスID  
ソース画像  
デスティネーション画像  
面積フィルタしきい値  
面積フィルタしきい値  
ラベリング処理オプション1  
ソート処理オプション  
基本特徴量テーブル  
ラベリング処理オプション2

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画像

**ImgDst**  
デスティネーション画像

**thrmin**  
面積フィルタしきい値(最小値)

**thrmax**  
面積フィルタしきい値(最大値)

**opt**  
ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

## Sort\_opt

ソート処理オプション

ソート処理オプション	対応定数	内容
LABEL_SORT_MAX	0	面積でのソートを大きい順に行う
LABEL_SORT_MIN	1	面積でのソートを小さい順に行う

## \*BasicTbl

基本特徴量テーブル。255個以上確保して下さい。

## opt2

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデステーション画面未出力
LABEL_DST	1	結果のデステーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了(ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	指定画面空き領域、または範囲外(不当画面番号エラー)
22	オプション設定値範囲外
30	SRC0 ウィンドウ幅エラー(ウィンドウ幅は2以上にして下さい)
49	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は254個です。ラベルオーバーフローは、ラベル数が255個以上、仮ラベル数が512個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRL()コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0番目は不定で、1番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数(254)+1確保しておく必要があります。

本コマンドで得られる各座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。

詳細はIP\_Label4byRL()コマンドを参照して下さい。

# IP\_Label8byRLwithAreaFLSort

ランレングス・面積フィルタ付ソートラベリング(8連結)

## 機能

ソース画面(2値)に対し、8連結でラベリング処理を行い、指定しきい値で面積フィルタ処理後に、面積でソートして再度ラベル付けと基本特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

## コーディング

```

int
IP_Label8byRLwithAreaFLSort(
    DEVID          devID ,
    int            ImgSrc ,
    int            ImgDst ,
    int            thrmin ,
    int            thrmax ,
    enum IP_Label_opt  opt ,
    enum IP_Label_Sort_opt  Sort_opt ,
    IPLabelBasicTbl *BasicTbl ,
    enum IP_Label_opt2  opt2
)

```

## コメント

デバイスID  
 ソース画像  
 ディスティネーション画像  
 面積フィルタしきい値  
 面積フィルタしきい値  
 ラベリング処理オプション1  
 ソート処理オプション  
 基本特徴量テーブル  
 ラベリング処理オプション2

## パラメータ

**devID**  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
 ソース画像

**ImgDst**  
 ディスティネーション画像

**thrmin**  
 面積フィルタしきい値(最小値)

**thrmax**  
 面積フィルタしきい値(最大値)

**opt**  
 ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

## Sort\_opt

ソート処理オプション

ソート処理オプション	対応定数	内容
LABEL_SORT_MAX	0	面積でのソートを大きい順に行う
LABEL_SORT_MIN	1	面積でのソートを小さい順に行う

## \*BasicTbl

基本特徴量テーブル。255個以上確保して下さい。

## opt2

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデステーション画面未出力
LABEL_DST	1	結果のデステーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了(ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
10, 13	指定画面空き領域、または範囲外(不当画面番号エラー)
22	オプション設定値範囲外
30	SRC0 ウィンドウ幅エラー(ウィンドウ幅は2以上にして下さい)
49	アクセス画面が不定画面
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は254個です。ラベルオーバーフローは、ラベル数が255個以上、仮ラベル数が512個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRL()コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0番目は不定で、1番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数(254)+1確保しておく必要があります。

本コマンドで得られる各座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。

詳細はIP\_Label4byRL()コマンドを参照して下さい。

# IP\_Label4byRLExt

## ランレングス・ラベリング(4連結・拡張)

### 機能

ソース画面(2値)に対し、4連結でラベリング処理と基本特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

### コーディング

```
int
IP_Label4byRLExt(
    DEVID
    int
    int
    enum IP_Label_opt
    ILabelExtTbl
    enum IP_Label_opt2
)
```

```
devID ,
ImgSrc ,
ImgDst ,
opt ,
*ExtTbl ,
opt2
```

### コメント

```
デバイスID
ソース画像
デスティネーション画像
ラベリング処理オプション1
基本特徴量テーブル
ラベリング処理オプション2
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画像

#### ImgDst

デスティネーション画像

#### opt

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

#### \*ExtTbl

基本特徴量テーブル。255個以上確保して下さい。

#### opt2

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2	オプション設定値範囲外
3 0	S R C 0 ウィンドウ幅エラー (ウィンドウ幅は2以上にしてください)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は254個です。ラベルオーバーフローは、ラベル数が255個以上、仮ラベル数が512個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容を下記に示します。

ipxsys.h で宣言しています。

```
typedef struct {
    long AREA; /* 面積 */
    short LEFT_UP_POINT_X; /* 最小X座標 */
    short LEFT_UP_POINT_Y; /* 最小Y座標 */
    short RIGHT_DOWN_POINT_X; /* 最大X座標 */
    short RIGHT_DOWN_POINT_Y; /* 最大Y座標 */
    float CENTER_of_GRAVITY_X; /* 重心X座標 */
    float CENTER_of_GRAVITY_Y; /* 重心Y座標 */
    float ANGLE; /* 慣性主軸とX軸とのなす角度 (degree) */
    float AXIS_LONG; /* 予備 */
    float AXIS_SHORT; /* 予備 */
    long Round_Length; /* 予備 */
} IPELabelExtTbl;
```

基本、拡張特徴量のテーブルはラベル番号に対応させています。0番目は不定で、1番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数(254)+1確保しておく必要があります。

本コマンドで得られる各座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。画面上の座標位置はそれぞれ

MinX = ExtTbl[ラベル番号].LEFT_UP_POINT_X + sx	MinX	: 最小X座標
MinY = ExtTbl[ラベル番号].LEFT_UP_POINT_Y + sy	MinY	: 最小Y座標
MaxX = ExtTbl[ラベル番号].RIGHT_DOWN_POINT_X + sx	MaxX	: 最大X座標
MaxY = ExtTbl[ラベル番号].RIGHT_DOWN_POINT_Y + sy	MaxY	: 最大Y座標
Gx = ExtTbl[ラベル番号].CENTER_of_GRAVITY_X + sx	Gx	: 重心X座標
Gy = ExtTbl[ラベル番号].CENTER_of_GRAVITY_Y + sy	Gy	: 重心Y座標
	sx, sy	: SRC0_WIN(sx, sy)

となります。

# IP\_Label8byRLExt

## ランレングス・ラベリング(8連結・拡張)

### 機能

ソース画面(2値)に対し、8連結でラベリング処理と基本特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

### コーディング

```
int
IP_Label8byRLExt(
    DEVID
    int
    int
    enum IP_Label_opt
    IPLabelExtTbl
    enum IP_Label_opt2
)
```

```
devID ,
ImgSrc ,
ImgDst ,
opt ,
*ExtTbl ,
opt2
```

### コメント

```
デバイスID
ソース画像
デスティネーション画像
ラベリング処理オプション1
基本特徴量テーブル
ラベリング処理オプション2
```

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画像

#### ImgDst

デスティネーション画像

#### opt

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

#### \*ExtTbl

基本特徴量テーブル。255個以上確保して下さい。

#### opt2

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2	オプション設定値範囲外
3 0	S R C 0 ウィンドウ幅エラー (ウィンドウ幅は2以上にして下さい)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は254個です。ラベルオーバーフローは、ラベル数が255個以上、仮ラベル数が512個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRLExt()コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0番目は不定で、1番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数(254)+1確保しておく必要があります。

本コマンドで得られる各座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。

詳細はIP\_Label4byRLExt()コマンドを参照して下さい。

## IP\_Label4byRLwithAreaFLText

### ランレングス・拡張面積フィルタ付きラベリング(4連結)

#### 機能

ソース画面(2値)に対し、4連結でラベリング処理を行い、指定しきい値で面積フィルタ処理後に、再度ラベル付けと拡張特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

#### コーディング

```
int
IP_Label4byRLwithAreaFLText(
    devID ,
    int     imgSrc ,
    int     imgDst ,
    int     thrmin ,
    int     thrmax ,
    enum IP_Label_opt
    IPLabelExtTbl
    enum IP_Label_opt2
    *ExtTbl ,
    opt2
)
```

#### コメント

デバイスID  
 ソース画像  
 デスティネーション画像  
 面積フィルタしきい値(最小値)  
 面積フィルタしきい値(最大値)  
 ラベリング処理オプション1  
 基本特徴量テーブル  
 ラベリング処理オプション2

#### パラメータ

##### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

##### imgSrc

ソース画像

##### imgDst

デスティネーション画像

##### thrmin

面積フィルタしきい値(最小値)

##### thrmax

面積フィルタしきい値(最大値)

##### opt

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

##### \*ExtTbl

基本特徴量テーブル。255個以上確保して下さい。

##### opt2

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2、2 3	面積フィルタしきい値範囲設定エラー
2 4	オプション設定値範囲外
3 0	S R C 0 ウィンドウ幅エラー (ウィンドウ幅は2以上にしてください)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は254個です。ラベルオーバーフローは、ラベル数が255個以上、仮ラベル数が512個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRLExt()コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0番目は不定で、1番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数(254)+1確保しておく必要があります。

本コマンドで得られる各座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。

詳細はIP\_Label4byRLExt()コマンドを参照して下さい。

# IP\_Label8byRLwithAreaFLTEExt

## ランレングス・拡張面積フィルタ付きラベリング(8連結)

### 機能

ソース画面(2値)に対し、8連結でラベリング処理を行い、指定しきい値で面積フィルタ処理後に、再度ラベル付けと拡張特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

### コーディング

```
int
IP_Label8byRLwithAreaFLTEExt(
    DEVID          devID ,
    int            ImgSrc ,
    int            ImgDst ,
    int            thrmin ,
    int            thrmax ,
    enum IP_Label_opt opt ,
    ILabelExtTbl  *ExtTbl ,
    enum IP_Label_opt2 opt2
)
```

### コメント

デバイスID  
ソース画像  
デスティネーション画像  
面積フィルタしきい値(最小値)  
面積フィルタしきい値(最大値)  
ラベリング処理オプション1  
基本特徴量テーブル  
ラベリング処理オプション2

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画像

#### ImgDst

デスティネーション画像

#### thrmin

面積フィルタしきい値(最小値)

#### thrmax

面積フィルタしきい値(最大値)

#### opt

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

#### \*ExtTbl

基本特徴量テーブル。255個以上確保して下さい。

#### opt2

ラベリング処理オプション2

ラベリング処理オプション2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2、2 3	面積フィルタしきい値範囲設定エラー
2 4	オプション設定値範囲外
3 0	S R C 0 ウィンドウ幅エラー (ウィンドウ幅は2以上にしてください)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

ラベル付けの最大数は254個です。ラベルオーバーフローは、ラベル数が255個以上、仮ラベル数が512個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRLExt()コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0番目は不定で、1番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数(254)+1確保しておく必要があります。

本コマンドで得られる各座標はSRC0\_WINで設定されたウィンドウからの相対座標になります。

詳細はIP\_Label4byRLExt()コマンドを参照して下さい。

# IP\_Label4byRLwithAreaFLSortExt

## ランレングス・拡張面積フィルタ付ソートラベリング(4連結)

### 機能

ソース画面(2値)に対し、4連結でラベリング処理を行い、指定しきい値で面積フィルタ処理後に、面積でソートして再度ラベル付けと拡張特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

### コーディング

```
int
IP_Label4byRLwithAreaFLSortExt(
    DEVID ,
    int     ImgSrc ,
    int     ImgDst ,
    int     thrmin ,
    int     thrmax ,
    enum IP_Label_opt
    enum IP_Label_Sort_opt
    ILabelExtTbl
    enum IP_Label_opt2
)
```

### コメント

デバイスID  
 ソース画像  
 デスティネーション画像  
 面積フィルタしきい値(最小値)  
 面積フィルタしきい値(最大値)  
 ラベリング処理オプション1  
 ソート処理オプション  
 基本特徴量テーブル  
 ラベリング処理オプション2

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画像

#### ImgDst

デスティネーション画像

#### thrmin

面積フィルタしきい値(最小値)

#### thrmax

面積フィルタしきい値(最大値)

#### opt

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

#### Sort\_opt

ソート処理オプション

ラベリング処理オプション2	対応定数	内容
LABEL_SORT_MAX	0	面積でのソートを大きい順に行う
LABEL_SORT_MIN	1	面積でのソートを小さい順に行う

#### \*ExtTbl

基本特徴量テーブル。255個以上確保して下さい。

opt2

ラベリング処理オプション 1

ラベリング処理オプション 2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0, 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2、2 3	面積フィルタしきい値範囲設定エラー
2 4、2 5	オプション設定値範囲外
3 0	S R C 0 ウィンドウ幅エラー (ウィンドウ幅は 2 以上にしてください)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ラベル付けの最大数は 2 5 4 個です。ラベルオーバーフローは、ラベル数が 2 5 5 個以上、仮ラベル数が 5 1 2 個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRLExt() コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0 番目は不定で、1 番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数 (2 5 4) + 1 確保しておく必要があります。

本コマンドで得られる各座標は SRC0\_WIN で設定されたウィンドウからの相対座標になります。

詳細は IP\_Label4byRLExt() コマンドを参照して下さい。

# IP\_Label8byRLwithAreaFLSortExt

## ランレングス・拡張面積フィルタ付ソートラベリング(8連結)

### 機能

ソース画面(2値)に対し、8連結でラベリング処理を行い、指定しきい値で面積フィルタ処理後に、面積でソートして再度ラベル付けと拡張特徴量の抽出を行います。本コマンドでは、254個までのラベル特徴量を抽出可能です。それ以上のラベル特徴量の抽出が必要な場合は、IP\_LabelCombine( )コマンドを使用して下さい。

### コーディング

```
int
IP_Label8byRLwithAreaFLSortExt(
    DEVID          devID ,
    int            ImgSrc ,
    int            ImgDst ,
    int            thrmin ,
    int            thrmax ,
    enum IP_Label_opt opt ,
    enum IP_Label_Sort_opt Sort_opt ,
    IPLabelExtTbl *ExtTbl ,
    enum IP_Label_opt2 opt2
)
```

### コメント

デバイスID  
 ソース画像  
 デスティネーション画像  
 面積フィルタしきい値(最小値)  
 面積フィルタしきい値(最大値)  
 ラベリング処理オプション1  
 ソート処理オプション  
 基本特徴量テーブル  
 ラベリング処理オプション2

### パラメータ

#### devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

#### ImgSrc

ソース画像

#### ImgDst

デスティネーション画像

#### thrmin

面積フィルタしきい値(最小値)

#### thrmax

面積フィルタしきい値(最大値)

#### opt

ラベリング処理オプション1

ラベリング処理オプション1	対応定数	内容
LABEL_OBJ	0	白を物体としてラベリング
LABEL_BKG	1	黒を物体としてラベリング

#### Sort\_opt

ソート処理オプション

ラベリング処理オプション2	対応定数	内容
LABEL_SORT_MAX	0	面積でのソートを大きい順に行う
LABEL_SORT_MIN	1	面積でのソートを小さい順に行う

#### \*ExtTbl

基本特徴量テーブル。255個以上確保して下さい。

opt2

ラベリング処理オプション 1

ラベリング処理オプション 2	対応定数	内容
LABEL_NO_DST	0	結果のデスティネーション画面未出力
LABEL_DST	1	結果のデスティネーション画面出力

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 254	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0, 1 3	指定画面空き領域、または範囲外 (不当画面番号エラー)
2 2、2 3	面積フィルタしきい値範囲設定エラー
2 4、2 5	オプション設定値範囲外
3 0	S R C 0 ウィンドウ幅エラー (ウィンドウ幅は 2 以上にしてください)
4 9	アクセス画面が不定画面
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

ラベル付けの最大数は 2 5 4 個です。ラベルオーバーフローは、ラベル数が 2 5 5 個以上、仮ラベル数が 5 1 2 個以上のいずれかで発生します。

基本特徴量テーブルのフォーマット及び内容については、IP\_Label4byRLExt() コマンドを参照して下さい。

基本、拡張特徴量のテーブルはラベル番号に対応させています。0 番目は不定で、1 番目から物体の特徴量が入ります。このため、特徴量テーブルは必ず出力最大数 (2 5 4) + 1 確保しておく必要があります。

本コマンドで得られる各座標は SRC0\_WIN で設定されたウィンドウからの相対座標になります。

詳細は IP\_Label4byRLExt() コマンドを参照して下さい。

## IP\_LabelCombine

## 統合ラベリング

## 機能

ソース画面（多値）を2値化してラベリング処理を行い、統合処理を行ったラベルの特徴量を算出及びラベル画像を出力します。本コマンドでは、4000個（仮ラベル数）までのラベル特徴量を抽出可能です。

最大ラベル数は4000個ですが、あくまでも仮ラベルの最大数です。実際はその半分程度の2000個程度までのラベル数になると考えられます。また、実際ソース画像のラベル数により、処理時間が大幅に変化しますので注意して下さい。

## コーディング

```
int
IP_LabelCombine(
    DEVID          devID ,
    int            ImgSrc ,
    int            ImgDst ,
    enum IPCombineType type ,
    IPBinarizeThr *binarizeThr ,
    IPLabelCtl    *labelCtl ,
    IPCombineCtl  *combineCtl ,
    IPCombineTbl  *combineTbl ,
    IPCombineTblExt *combineTblExt
)
```

## コメント

デバイスID  
ソース画像  
デスティネーション画像  
統合ラベリングオプション  
2値化しきい値  
ラベリング管理テーブル  
統合管理テーブル  
基本特徴量テーブル  
拡張特徴量テーブル

## パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画像

**ImgDst**  
デスティネーション画像

**type**  
統合ラベリングオプション

統合ラベリングオプション	内容
NORMAL_FEATURE	基本特徴量,ラベル画像を出力します
EXTRA_FEATURE	基本特徴量,拡張特徴量,ラベル画像を出力します
NORMAL_FEATURE_NODST	基本特徴量のみ出力します デスティネーション画面にラベリング結果は出力しません
EXTRA_FEATURE_NODST	基本特徴量,拡張特徴量を出力します デスティネーション画面にラベリング結果は出力しません

**\*binarizeThr**  
2値化しきい値

**\*labelCtl**  
ラベリング管理テーブル

**\*combineCtl**  
統合管理テーブル。現在未サポートの機能です。必ず以下のように設定して下さい。  
(下記以外の設定値では正常動作しません)

```
combineCtl.including = NO_INCLUDE ; /* 包含無効 */
combineCtl.width     = 0 ;          /* 統合無し */
combineCtl.height    = 0 ;          /* 統合無し */
```

\*combineTbl ( 2 ~ 4 0 0 1 )  
基本特徴量テーブルのポインタ

\*combineTblExt ( 2 ~ 4 0 0 1 )  
拡張特徴量テーブルのポインタ

(注) 特徴量のテーブルサイズは、必ず  
「ラベリング管理テーブルの出力最大数+1」  
確保してください

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0 rtn 4000	正常終了 (ラベル数)
-1	異常終了
LABEL_OVERFLOW(0x7FFF)	ラベルオーバーフロー

異常終了時、下記の原因のコマンドエラーが考えられます。

エラーコード	エラー原因
1 0 , 1 3	画面空き領域、または範囲外 (不当画面番号エラー)
2 2	ランレングス統合オプション設定範囲外
2 3	2 値化管理テーブル設定範囲外
2 4	ラベリング管理テーブル設定範囲外
3 0	S R C 0 ウィンドウ設定エラー
3 3	D S T ウィンドウ設定エラー
4 8	動的メモリ確保エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

2 値化しきい値構造体、ラベリング管理テーブル、統合管理テーブル、基本特徴量テーブル、拡張特徴量テーブルのフォーマット及び内容を下記に示します。  
これらは ipxdef.h, ipxsys.h で宣言しています。

### IPBinarizeThr : 2 値化しきい値

```
typedef struct {
    int thrmin;      /* 最小しきい値 ( 0 ~ thrmax ) */
    int thrmax;     /* 最大しきい値 ( thrmin ~ 255 ) */
} IPBinarizeThr;
```

## IPLabelCtl :ラベリング管理テーブル

連結手法	内容
LABEL4	4 連結
LABEL8	8 連結

ソート手法	内容
LABEL_SORT_MAX	面積でのソートを大きい順に行う
LABEL_SORT_MIN	面積でのソートを小さい順に行う
LABEL_SORT_NON	出現順

```
typedef struct {
    int          max_num;      /* 出力最大数 ( 1 ~ 4000 )          */
    enum IPLabelConnect connect; /* 連結手法                          */
    int          area_min;    /* 最小面積しきい値 ( 1 ~ thrmax )  */
    int          area_max;    /* 最大面積しきい値 ( thrmin ~ INT_MAX ) */
    enum IP_Label_Sort_opt sort; /* ソート手法                          */
} IPLabelCtl;
```

## IPCombineInc :統合管理テーブル

包含による統合	内容
INCLUDE	包含有効
NO_INCLUDE	包含無効

```
typedef struct {
    int          width;      /* 統合幅範囲 ( 0 ~ 横ウインドウAND サイズ          */
                                /*              =0 :統合なし )                          */
    int          height;    /* 統合高さ範囲 ( 0 ~ 縦ウインドウAND サイズ)          */
    enum IPCombineInc including; /* 包含による統合                          */
} IPCombineCtl;
```

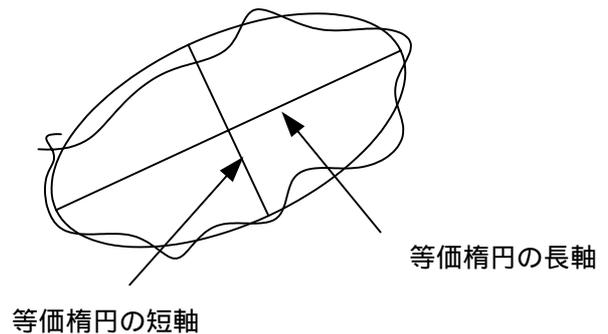
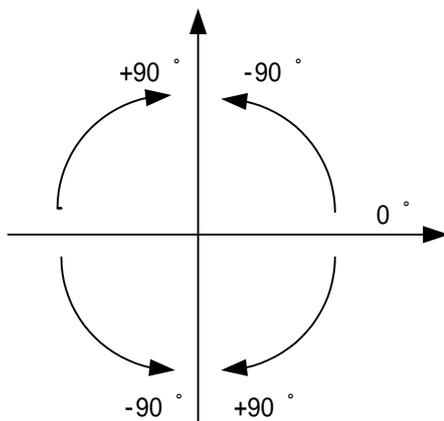
## IPCombineTbl : 基本特徴量テーブルのポインタ

```
typedef struct {
    int    area;          /* 物体面積 */
    int    xmin;         /* 物体を囲む最小X座標 */
    int    xmax;         /* 物体を囲む最大X座標 */
    int    ymin;         /* 物体を囲む最小Y座標 */
    int    ymax;         /* 物体を囲む最大Y座標 */
    int    length;       /* 物体周囲長 */
}IPCombineTbl;
```

## IPCombineTblExt : 拡張特徴量テーブルのポインタ

```
typedef struct {
    float  xmean;        /* 重心X座標 */
    float  ymean;        /* 重心Y座標 */
    float  tilt;         /* 物体傾き ( -90 ° ~ 90 ° ) (注) */
    float  axs_ratio;    /* 短軸の長さ ÷ 長軸の長さ (物体の等価楕円) */
}IPCombineTblExt;
```

(注)



- 基本、拡張特徴量のテーブルはラベル番号に対応させています。0番目は不定で、1番目から物体の特徴量が入ります。このため特徴量テーブルは、必ずラベリング管理テーブルで設定した出力最大数(max\_num)+1 確保しておく必要があります。
- 仮ラベル付けの最大数は4 0 0 0個です。ラベルオーバーフローは、仮ラベル数4 0 0 1個以上で発生します。真ラベル付けの最大数は4 0 0 0個です。
- 現状でのラベル画像のデータタイプはUNSIGN8 なので、2 5 5番目以上のラベル毎の特徴量は出力されますが、ラベル画像には出力されません。出力ラベル数が「0」の時、出力画面は不定になります。

# socket

## ソケット生成

### 機能

ソケットを生成し、ソケット識別子を返します。

本ソケットAPI関数においてサポートしているプロトコルは、TCPとUDPのみです。

familyにはAF\_INET(2)を指定してください(PF\_INET(2)でも可)。

typeには、TCPではSOCK\_STREAM(1)、UDPではSOCK\_DGRAM(2)を指定してください(SOCK\_RAW(3)は指定できません)。protocolには、TCPでは0またはTCPのプロトコル番号(6)を、UDPでは0またはUDPのプロトコル番号(17)を指定してください。

BSDソケットコマンドを使用する場合は、"VPXSOCK.H"をインクルードして下さい。

### コーディング

```
int
socket(
    int      family,
    int      type,
    int      protocol
)
```

### コメント

アドレスファミリ  
サービス型  
使用するプロトコル番号

### パラメータ

family      アドレスファミリ

type        サービス型

protocol    使用するプロトコル番号

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	正常処理 (生成したソケットのソケット識別子)
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
ENFILE	2 3	ファイルテーブル不足 (ソケット数に空きがない)
EPROTONOSUPPORT	4 3	プロトコル未サポート (アドレスファミリまたはサービスまたはプロトコルが不正)

### 詳細情報

空きのソケットがなく、ソケットが生成できない場合は、大域変数errnoにENFILEを設定し-1 (エラー) を返します。

family、type、protocolが表にない組み合わせの場合は、大域変数errnoにEPROTONOSUPPORTを設定し-1 (エラー) を返します。

使用するプロトコルごとのfamily、type、protocolの設定を下の表に示します。

使用するプロトコル	Family	type	protocol
TCPのプロトコル	AF_INET(2) または PF_INET(2)	SOCK_STREAM(1)	0
			TCPのプロトコル番号(6)
UDPのプロトコル	PF_INET(2)	SOCK_DGRAM(2)	0
			UDPのプロトコル番号(17)

# bind

## 通信アドレス情報の指定

### 機能

ソケットに対し自分の通信アドレス情報を割り当てます。  
 socket()を用いて作成された直後のソケットには、通信アドレス情報が割当てられていません。  
 bind()はsockfd が指すソケットに、p\_localaddr で示す領域の通信アドレス情報を割り当てます。  
 通信アドレス情報はstruct sockaddr\_in 型です。この領域のアドレスを関数に渡す際には、  
 struct sockaddr へのポインタ型にキャストしてください。

### コーディング

```
int
bind(
    int                sockfd,
    struct sockaddr    *p_localaddr,
    socklen_t          addrLen
)
```

### コメント

ソケット識別子  
 自分の通信アドレス情報を格納した領域のアドレス  
 自分の通信アドレス情報を格納した領域の長さ

### パラメータ

**sockfd**                    アドレスファミリ

**\*p\_localaddr**            自分の通信アドレス情報を格納した領域のアドレス

**addrLen**                    自分の通信アドレス情報を格納した領域の長さ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	対応定数	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 ( sockfd < 0 )
ENOMEM	1 2	メモリ不足 (UDPソケットを生成するための受信バッファ資源が足りない)
EFAULT	1 4	ポインタアドレス不正 ( p_localaddr が 4 の倍数以外 )
EINVAL	2 2	引数不正 指定したソケットはすでに自通信アドレス情報が決定している addrLen が構造体struct sockaddr_in 型の長さとは異なる
EPIPE	3 2	指定したソケットに対しshutdown()が実行されている
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfd と同じ識別子のソケットが存在しない )
EAFNOSUPPORT	4 7	アドレスファミリが不正 ( アドレスファミリがAF_INT(2)以外 )
EADDRINUSE	4 8	通信アドレス情報使用中 指定した通信アドレス情報において、自動割り当て以外の指定をしたIPアドレスまたはポート番号が、他のソケットにおいて使用中 指定した通信アドレス情報において、自動割り当て指定をしたIPアドレスまたはポート番号に空きが無い
EADDRNOTAVAIL	4 9	IPアドレス利用不可
EPROCLIM	6 7	プロセス超過 ( TCP/IPマネージャのUDP通信端点が足りない )
EUSERS	6 8	重複操作 ( 指定したソケットを他のBSD 関数が使用中 )

指定した通信アドレス情報のうち、IPアドレスまたはポート番号が他のソケットにおいて使用中の場合は、大域変数 `errno` に `EADDRINUSE` \*2 を設定し-1 (エラー) を返します。

`sockfd` が指定したソケットにすでに通信アドレス情報が割り当てられている場合は、通信アドレス情報を割り当てることはできないため、大域変数 `errno` に `EINVAL` を設定し-1 (エラー) を返します。

`addrlen` が構造体 `struct sockaddr_in` 型の長さ異なる場合は、大域変数 `errno` に `EINVAL` を設定し-1 (エラー) を返します。

`sin_port` に 0 が指定された場合には登録されているポート番号の範囲から未使用のポートを検索して割り当てます。未使用のポートが無い場合は、大域変数 `errno` に `EADDRINUSE` を設定し-1 (エラー) を返します。

`sin_addr` で指定したIPアドレスが登録されていない場合は、大域変数 `errno` に `EADDRNOTAVAIL` を設定し-1 (エラー) を返します。

`sin_addr` に 0 が指定された場合には登録されているIPアドレスから未使用のIPアドレスを検索して割り当てます。未使用のIPアドレスが無い場合は、大域変数 `errno` に `EADDRINUSE` を設定し-1 (エラー) を返します。

TCP/IP マネージャで確保したUDP通信端点に空きがなく処理ができない場合は、大域変数 `errno` に `EPROCLIM` を設定し-1 (エラー) を返します。

UDPソケットが送受信に使用するバッファが確保できない場合は、大域変数 `errno` に `ENOMEM` を設定し-1 (エラー) を返します。

\*2: ソケットオプション `SO_REUSEADDR` がOFFの時のものです。  
詳細は、`getsockopt()` の解説を参照してください。

#### パケットの構造

##### < IP アドレスの構造体 >

```
struct in_addr {
    in_addr_t s_addr;
};
```

##### < 通信アドレス情報の構造体 >

```
struct sockaddr_in {
    uint8_t      sin_len;           // 構造体の長さ (16 バイト固定)
    sa_family_t  sin_family;       // アドレスファミリ (AF_INET(2))
    in_port_t    sin_port;         // TCP あるいはUDP のポート番号
    struct in_addr sin_addr;       // IPアドレス (ネットワークバイトオーダー)
    char         sin_zero[8];      // 未使用
};
```

# listen

## 受動モード設定

### 機能

ソケットを受動オープンし、受動モードに設定します。

### コーディング

```
int
listen(
    int          sockfd,
    int          queuelen
)
```

### コメント

ソケット識別子  
受け入れ可能なコネクション要求の数

### パラメータ

**sockfd**           ソケット識別子

**queuelen**       受け入れ可能なコネクション要求の数

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常処理
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 ( sockfd < 0 )
ENOMEM	1 2	メモリ不足 (ソケット用のバッファが不足している。)
EINVAL	2 2	引数不正 queuelen 0 指定したソケットを、受動モードへ変更できない
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfdと同じ識別子のソケットが存在しない)
EOPNOTSUPP	4 5	未サポートソケット (指定したソケットがTCPのソケットでない)
EADDRINUSE	4 8	通信アドレス情報使用中 登録されている通信アドレス情報のポート番号が、他のソケットにおいて使用中 (自動割り当て指定でない場合) 通信アドレス情報に割り当てるポート番号に空きが無い (自動割り当て指定の場合)
EPROCLIM	6 7	プロセス超過 (TCP/IP マネージャのTCP 受付口が足りない)
EUSERS	6 8	重複操作 (指定したソケットを他のBSD 関数が使用中)
EDEADLK	7 8	デッドロック条件 BSDで生成した通信端点またはポートを他のプロセスが使用中 BSDで生成した通信端点か他のプロセスによって削除された BSDで生成した受付口が他のプロセスによって削除された
EPROTO	8 6	プロトコルエラー (使用するIPアドレスが動作していない)

**詳細情報**

queuelenには、このソケットが受け付けることができるコネクション要求の最大数を設定します。queuelenには1以上を指定してください。0以下の値を指定した場合は、大域変数 `errno` に `EINVAL` を設定し - 1 (エラー) を返します。

対象ソケットが下記に示す(1)と(2)の状態以外の場合は、大域変数 `errno` に `EINVAL` を設定し - 1 (エラー) を返します。

- (1) `socket()`で生成した直後のソケット
- (2) `bind()`が成功した直後のソケット

`listen()`は、TCPのソケットが対象です。指定されたソケットがTCPでない場合は、大域変数 `errno` に `EOPNOTSUPP` を設定し - 1 (エラー) を返します。

ソケットに登録されている通信アドレス情報のポート番号が、他のソケットにおいて既に使用中の場合は、大域変数 `errno` に `EADDRINUSE` を設定し - 1 (エラー) を返します。

通信アドレス情報のポート番号を自動割り当てに指定し、割り当てるポート番号に空きが無い場合は、大域変数 `errno` に `EADDRINUSE` を設定し - 1 (エラー) を返します。

ソケットが送受信に使用するバッファが確保できない場合は、大域変数 `errno` に `ENOMEM` を設定し - 1 (エラー) を返します。

TCP/IPマネージャで確保したTCP受付口に空きがなく処理が行えない場合は、大域変数 `errno` に `EPROCLIM` を設定し - 1 (エラー) を返します。

マルチタスク動作において、指定したソケットを他のプロセスから呼び出されたBSD関数が操作中の場合は、大域変数 `errno` に `EUSERS` を設定し - 1 (エラー) を返します。

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API上で動作しています。BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数 `errno` に `EDEADLK` を設定し - 1 (エラー) を返します。

自分の通信アドレス情報として設定されているIPアドレスがTCP/IPマネージャ上で動作していない場合は、大域変数 `errno` に `EPROTO` を設定し - 1 (エラー) を返します。

# accept

## ソケットに対するコネクションの受入れ

### 機能

確立済みコネクションから接続済みのソケットを生成します。  
 確立済みコネクションから接続済みのソケットを生成してデータの送受信を可能にした後、p\_addrの指し示す領域に接続相手の通信アドレス情報を返します。  
 p\_addr が示す領域はstruct sockaddr\_in型です。関数に渡す際にはstruct sockaddr型のポインタにキャストする必要があります。

### コーディング

```
int
accept(
    int
    struct sockaddr
    socklen_t
)
    sockfd,
    *p_addr,
    *addrlen
```

### コメント

ソケット識別子  
 接続相手の通信アドレス情報を格納する領域のアドレス  
 通信アドレス情報の長さを格納した領域のアドレス

### パラメータ

**sockfd** ソケット識別子

**\*p\_addr** 接続相手の通信アドレス情報を格納する領域のアドレス

**\*addrlen** 通信アドレス情報の長さを格納した領域のアドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	新しく生成されたソケット識別子 (正常終了)
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EINTR	4	割り込みエラー (待ち状態が強制的に解除された)
EBADF	9	識別子無効 ( sockfd < 0 )
EFAULT	1 4	ポインタアドレス不正 ( p_addr もしくは addrlen の示す値が4の倍数以外 )
EINVAL	2 2	引数不正 指定したソケットが受動モードでない addrlen が構造体struct sockaddr_in 型の長さとは異なる)
ENFILE	2 3	ファイルテーブル不足 (ソケット数に空きがない)
EWOLDBLOCK	3 5	ノンブロッキングリターン (ソケットがノンブロッキングモードでかつキュー上に待ち状態のコネクションがない)
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfd と同じ識別子のソケットが存在しない)
EOPNOTSUPP	4 5	未サポートソケット (指定したソケットがTCPのソケットではない)
EPROCLIM	6 7	プロセス超過 (TCP/IPマネージャのTCP通信端点が足りない)
EUSERS	6 8	重複操作 (指定したソケットを他のBSD 関数が使用中)
EDEADLK	7 8	デッドロック条件 BSDで生成した通信端点またはポートを他のプロセスが使用中 BSDで生成した通信端点が他のプロセスによって削除された BSDで生成した受付口が他のプロセスによって削除された

## 詳細情報

accept()はコネクション待ちキューにある確立済みコネクションを引き出し、確立済みコネクションから接続済みの能動モードのソケットを作成して、そのソケットに新しいソケット識別子を割り当てリターン値として返します。新しく生成された接続済みソケットは、sockfdの示すソケットとは別のソケットです。

キュー上に待ち状態のコネクションがない場合、コネクションができるまでaccept()からリターンしません(ソケットのノンブロッキングモード\*1がOFFの場合)。ソケットのノンブロッキングモード\*1がONの場合は、大域変数 errno に EWOULDBLOCK を設定しリターン値に - 1 を返します。

accept()により新しく生成された接続済みのソケットは、能動モードであり、データの送受信のために使用されます。

sockfd が示すソケットは、accept()からリターンしたあとも受動モードのまま存在するため、再び accept()を実行することが可能です。

accept()は、TCPのソケットが対象です。指定されたソケットがTCPでない場合は、大域変数 errno に EOPNOTSUPP を設定し - 1 (エラー) を返します。

sockfdの示すソケットは、受動モードかつshutdown()が実行されていない状態である必要があります。この条件を満たしていないソケットが指定された場合は、大域変数 errno に EINVAL を設定し - 1 (エラー) を返します。

addrlen が構造体 struct sockaddr\_in型の長さとは異なる場合は、大域変数 errno に EINVAL を設定し - 1 (エラー) を返します。

ソケットに空きが無く、新しいソケットが生成できない場合は、大域変数 errno に ENFILE を設定し - 1 (エラー) を返します。

他のプロセスによって待ち状態が強制的に解除された場合は、大域変数 errno に EINTR を設定し - 1 (エラー) を返します。

マルチタスク動作において、指定したソケットを他のプロセスから呼び出されたBSD関数が操作中の場合は、大域変数 errno に EUSERS を設定し - 1 (エラー) を返します。

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API上で動作しています。BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数 errno に EDEADLK を設定し - 1 (エラー) を返します。

\*1 : ノンブロッキングモードとそのON/OFFの切り替えについては set\_blocking\_socket()を参照してください。

## パケットの構造体

## &lt; IP アドレスの構造体 &gt;

```
struct in_addr {
    in_addr_t s_addr;
};
```

## &lt; 通信アドレス情報の構造体 &gt;

```
struct sockaddr_in {
    uint8_t      sin_len;           // 構造体の長さ(16バイト固定)
    sa_family_t  sin_family;       // アドレスファミリ(AF_INET(2))
    in_port_t    sin_port;         // TCPあるいはUDPのポート番号
    struct in_addr sin_addr;       // IPアドレス(ネットワークバイトオーダー)
    char         sin_zero[8];      // 未使用
};
```

# connect

## ソケットの接続の開始

### 機能

通信相手とのコネクションを確立します。  
 p\_addr が示す領域はstruct sockaddr\_in型です。関数に渡す際にはstruct sockaddr型のポインタにキャストする必要があります。

### コーディング

```
int
connect(
    int
    struct sockaddr
    socklen_t
)
```

```
sockfd,
*p_addr,
addr len
```

### コメント

ソケット識別子  
 相手通信アドレス情報を格納する領域のアドレス  
 相手通信アドレス情報の長さを格納した領域のアドレス

### パラメータ

sockfd           ソケット識別子

\*p\_addr           相手通信アドレス情報を格納する領域のアドレス

addr len           相手通信アドレス情報の長さを格納した領域のアドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常処理
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EINTR	4	割り込みエラー (待ち状態が強制的に解除された)
EBADF	9	識別子無効 ( sockfd < 0 )
ENOMEM	1 2	メモリ不足 (ソケット用のバッファが不足している)
EFAULT	1 4	ポインタアドレス不正 ( p_addr が0 または4 の倍数以外)
EINVAL	2 2	引数不正 指定したソケットが受動モード 指定したソケットが接続失敗状態 addr lenが構造体struct sockaddr_in型の長さとは異なる
EPIPE	3 2	指定したソケットに対しshutdown()が実行されている
EINPROGRESS	3 6	処理開始 (ノンブロッキング設定で、かつ接続がすぐに完了しない)
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfdと同じ識別子のソケットが存在しない)
EAFNOSUPPORT	4 7	アドレスファミリ未サポート TCPにおいてsin_family AF_INET(2) UDPにおいてsin_family AF_INET(2)かつ sin_family AF_UNSPEC(0)
EADDRINUSE	4 8	通信アドレス情報割り当て不可 (自分のポート番号に空きがない)
EADDRNOTAVAIL	4 9	通信アドレス情報利用不可 (相手のIPアドレスまたはポート番号が0)

大域変数	エラー番号	内容
EISCONN	5 6	接続済みソケット指定
ETIMEDOUT	6 0	タイムアウトエラー（プロトコルがコネクション確立前に75秒経過）
ECONNREFUSED	6 1	接続失敗（コネクションが相手によって拒絶された）
EPROCLIM	6 7	プロセス超過（TCP/IP マネージャのUDP 通信端点が足りない）
EUSERS	6 8	重複操作（指定したソケットを他のBSD 関数が使用中）
EDEADLK	7 8	デッドロック条件 BSDで生成した通信端点またはポートを他のプロセスが使用中 BSDで生成した通信端点が他のプロセスによって削除された
EPROTO	8 6	プロトコルエラー（使用するIPアドレスが動作していない）

## 詳細情報

ソケットが送受信に使用するバッファが確保できない場合は、大域変数 `errno` に `ENOMEM` を設定し - 1（エラー）を返します。

マルチタスク動作において、指定したソケットを他のプロセスから呼び出されたBSD関数が操作中の場合は、大域変数 `errno` に `EUSERS` を設定し - 1（エラー）を返します。

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API上で動作しています。BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数 `errno` に `EDEADLK` を設定し - 1（エラー）を返します。

自分の通信アドレス情報として設定されているIPアドレスがTCP/IPマネージャ上で動作していない場合は、大域変数 `errno` に `EPROTO` を設定し - 1（エラー）を返します。

次にプロトコルごとの説明を行います。

### (1) TCPのソケット

TCPのソケットの場合には、`connect()`は3ウェイハンドシェイクを使って`p_addr`の示す相手に対しコネクションを確立します。コネクションの確立に成功すると、0（正常終了）を返します。

コネクションの確立に失敗した場合、そのソケットは使用不可になり、同じソケットに対し再度`connect()`が呼び出された場合は必ず異常終了となります。

使用可能なIPアドレスやポート番号に空きがない場合は、大域変数`errno`に`EADDRINUSE`を設定し - 1（エラー）を返します。

コネクションが確立しないで一定時間（75秒）経過した場合は、大域変数`errno`に`ETIMEDOUT`を設定し - 1（エラー）を返します。

コネクションが相手によって拒絶された場合は、大域変数`errno`に`ECONNREFUSED`を設定し - 1（エラー）を返します。

ノンブロッキングモードがOFFの場合、コネクションが確立するかエラーが発生するまでは接続中（TCP/IPマネージャによる接続を待つ）の状態となりリターンしません。

他のプロセスによって待ち状態が強制的に解除された場合は、大域変数`errno`に`EINTR`を設定し - 1（エラー）を返します。

ノンブロッキングモードがONの場合は、大域変数`errno`に`EINPROGRESS`を設定し - 1（エラー）を返しますが異常ではありません。対象となっていたソケットはコネクションを開始し、接続中の状態になります。

コネクションが終了する前に再度`connect()`を実行した場合は、大域変数`errno`に`EINVAL`を設定し - 1（エラー）を返します。

コネクションが正常に終了した後に再度`connect()`を実行した場合は、大域変数`errno`に`EISCONN`を設定し - 1（エラー）を返します。

## (2) UDPのソケット

UDPのソケットの場合には相手側の通信アドレス情報を記録し、すぐにリターンします。これにより、UDPソケットを相手設定済みの状態にします。

UDPのソケットの場合、同じソケットに対し何度もconnect()が呼び出せます。このとき、別の相手側通信アドレスを設定すると、登録してある相手側通信アドレスが変更されます。

相手側通信アドレスの p\_addrにNULLを、addrlenに0を設定しconnect()を呼び出した場合は、大域変数 errno に EADDRNOTAVAIL を設定し - 1 (エラー) を返しますが異常ではありません。この場合、ソケットは相手未設定の状態になります。

送信側および受信側が共にshutdown()によって閉じられている場合は、大域変数 errno に EPIPE を設定し - 1 (エラー) を返します。

TCP/IPマネージャで確保したUDPの通信端点に空きがなく処理が行えない場合は、大域変数 errno に EPROCLIM を設定し - 1 (エラー) を返します。

## パケットの構造体

## &lt; IP アドレスの構造体 &gt;

```
struct in_addr {
    in_addr_t s_addr;
};
```

## &lt; 通信アドレス情報の構造体 &gt;

```
struct sockaddr_in {
    uint8_t      sin_len;           // 構造体の長さ (16バイト固定)
    sa_family_t  sin_family;       // アドレスファミリ
    in_port_t    sin_port;         // TCPあるいはUDPのポート番号
    struct in_addr sin_addr;       // IPアドレス (ネットワークバイトオーダー)
    char         sin_zero[8];      // 未使用
};
```

## getpeername

## 相手側通信アドレス情報取得

### 機能

接続されているソケットの相手側通信アドレス情報を取得します。

### コーディング

```
int
getpeername(
    int
    struct sockaddr
    socklen_t
)
```

sockfd,  
\*p\_remaddr,  
\*addrlen

### コメント

ソケット識別子  
相手通信アドレス情報を格納する領域  
相手通信アドレス情報を格納する領域の長さ

### パラメータ

sockfd           ソケット識別子

\*p\_remaddr       相手通信アドレス情報を格納する領域

\*addrlen         相手通信アドレス情報を格納する領域の長さ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常処理
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EINVAL	2 2	引数不正 (*addrlenが構造体struct sockaddr_in型の長さより小さい)
ENOTSOCK	3 8	識別子ソケット未指定 (sockfd と同じ識別子のソケットが存在しない)
ENOTCONN	5 7	ソケット未接続 TCPにおいて通信相手と接続されていない UDPにおいて通信相手が設定されていない
EADDRNOTAVAIL	9	識別子無効 ( sockfd < 0 )
EFAULT	1 4	ポインタアドレス不正 ( p_remaddr, addrlenが 0 または 4 の倍数以外 )
EPERM	1	関数実行不可

### 詳細情報

p\_remaddr が示す領域はstruct sockaddr\_in型です。関数に渡す際にはstruct sockaddr型のポインタにキャストする必要があります。対象ソケットが接続されておらず、相手がいない場合は、大域変数errnoにはENOTCONNを設定し - 1 (エラー) を返します。

パケットの構造は、connect()を参照して下さい。

# getsockname

## 自通信アドレス情報取得

### 機能

ソケット識別子が示すソケットの通信アドレス情報を取得します。

### コーディング

```
int
getsockname(
    int          sockfd,
    struct sockaddr *p_addr,
    socklen_t    *addrlen
)
```

### コメント

ソケット識別子  
自通信アドレス情報を格納する領域  
自通信アドレス情報の長さを格納する領域の先頭アドレス

### パラメータ

**sockfd**                   ソケット識別子

**\*p\_addr**                   相手通信アドレス情報を格納する領域  
(リターン: struct sockaddr\_in p\_addr;   指定ソケットの通信アドレス情報)

**\*addrlen**                  相手通信アドレス情報を格納する領域の長さ  
(リターン: int addrlen;                   指定ソケットの通信アドレス情報の長さ)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常処理
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 ( sockfd < 0 )
EFAULT	1 4	ポインタアドレス不正 ( p_addr または addrlen が 4 の倍数以外 )
EINVAL	2 2	引数不正 ( *addrlen が構造体 struct sockaddr_in 型の長さより小さい )
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfd と同じ識別子のソケットが存在しない )

### 詳細情報

p\_addr が示す領域は struct sockaddr\_in 型です。関数に渡す際には struct sockaddr 型のポインタにキャストする必要があります。指定のソケットに通信アドレス情報が設定されていない場合でも、正常終了になります。ただし、p\_addr に設定するパラメータの値は保証しません。

パケットの構造は、connect() を参照して下さい。

**recv****ソケットからの受信データ取得****機能**

ソケットからデータを受信します。

**コーディング**

```
ssize_t
recv(
    int
    void
    size_t
    int
)
```

```
sockfd,
*buffer,
length,
flags
```

**コメント**

ソケット識別子  
受信データを格納する領域の先頭アドレス  
受信データを格納する領域の長さ  
受信動作を指定するフラグ

**パラメータ**

**sockfd** ソケット識別子

**\*buffer** 受信データを格納する領域の先頭アドレス

**length** 受信データを格納する領域の長さ

**flags** 受信動作を指定するフラグ

**リターン値**

リターン値は `ssize_t` 型で返されます。詳細を下表に示します。

リターン値	内容
0 または正の数	正常処理 (受信データの長さ)
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EINTR	4	割り込みエラー (待ち状態が強制的に解除された)
EBADF	9	識別子無効 ( <code>sockfd &lt; 0</code> )
EINVAL	2 2	引数不正 指定したソケットが受動モード <code>length = 0</code> または <code>length 0x80000000</code> <code>flags</code> が不正
EPIPE	3 2	指定したソケットに対し <code>shutdown()</code> が実行されている
EWOULDBLOCK	3 5	ノンブロッキングリターン (ソケットがノンブロッキングモードでかつ受信済みのデータがない)
ENOTSOCK	3 8	識別子ソケット未指定 ( <code>sockfd</code> と同じ識別子のソケットが存在しない)
ECONNRESET	5 4	コネクションリセット (コネクションがリセットされた)
ENOTCONN	5 7	未接続エラー ( <code>sockfd</code> の示すソケットが未接続)
EUSERS	6 8	重複操作 (指定したソケットを他のBSD関数が使用中)
EDEADLK	7 8	デッドロック条件 BSDで生成した通信端点またはポートを他のプロセスが使用中 BSDで生成した通信端点が他のプロセスによって削除された

## 詳細情報

ソケット内に読み取られていない受信データがありその長さがlengthより小さい場合は、全てのデータを取り出してその取り出したデータの長さを返します。lengthより大きい場合は、length分のデータを取り出してその長さを返します。

ノンブロッキングモードがOFFの場合は、読み取られていない受信データが無いときは、データを受信するかエラーが発生するまでリターンしません。

ノンブロッキングモードがONで受信データがない場合は、大域変数errnoにEWOULDBLOCKを設定し - 1 (エラー)を返します。この場合は異常ではありません。

指定したソケットがTCPで接続されていない場合は、大域変数errnoにENOTCONNを設定し - 1 (エラー)を返します。

指定したソケットの受信側が、shutdown()によって閉じられている場合は、大域変数errnoにEPIPEを設定し - 1 (エラー)を返します。

指定したソケットがTCPで既にFINを受信しており、かつ受信バッファが空の場合は、正常終了として受信データ長0を返します (FINを受信しても受信データがあるうちは正常に受信できます)。

指定したソケットがTCPで既にコネクションがリセットされている場合は、大域変数errnoにECONNRESETを設定し - 1 (エラー)を返します。

TCPソケットの受信待ち状態でFINを受信し、かつ受信データが無い場合は、正常終了として受信データ長0を返します。

TCPソケットの受信待ち状態でコネクションがリセットされた(RST受信またはRST送信)場合は、大域変数errnoにECONNRESETを設定し - 1 (エラー)を返します。

他のプロセスによって待ち状態が強制的に解除された場合は、大域変数errnoにEINTRを設定し - 1 (エラー)を返します。

マルチタスク動作において、指定したソケットを他のプロセスから呼び出されたBSD関数が操作中の場合は、大域変数errnoにEUSERSを設定し - 1 (エラー)を返します。

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API上で動作しています。

BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数errnoにEDEADLKを設定し - 1 (エラー)を返します。

flagsに設定する値に対応した動作を行うことができます。また、値の論理和を設定することで2つ以上の動作を指定することができます。

flagsに不正な値が設定された場合は、大域変数errnoにEINVALを設定し - 1 (エラー)を返します。

recv(), recvfrom() で flags に設定する値

define名	値	内容	TCP	UDP
MSG_PEEK	0x0002	バッファリングデータを保持したままで読み込み実行		
MSG_DONTWAIT	0x0040	1回の操作毎のノンブロッキング指定		
MSG_WAITALL	0x0100	全データ到着待ち		×

注意：MSG\_PEEK とMSG\_WAITALL は同時に設定できません

**MSG\_PEEK** : 受信バッファの状態とデータを変化させずに、受信データの内容を取り出します。  
 MSG\_PEEKが設定されていない場合は、受信したデータ分、受信バッファの内容が変化します。  
 MSG\_PEEKが設定されている場合は、受信バッファの状態とデータを変化させません。  
 (再度、受信を実行したとき、同じデータを読み込みます)  
 T C Pの場合は、再度受信を実行するとデータの長さが増える場合があります。  
 U D Pの場合は、再度受信を実行してもまったく同じデータを取得します。

**MSG\_DONTWAIT** : ノンブロッキングモードON として動作します。  
 MSG\_DONTWAITが設定されていない場合は、ソケット自身のノンブロッキングモードに従って動作します。  
 MSG\_DONTWAITが設定されている場合は、関数内の処理はノンブロッキングモードONとして動作します。

**MSG\_WAITALL** : 要求された長さ分のデータを受信します。  
 MSG\_WAITALLが設定されていない場合は、受信データがあればその長さがlengthより小さい場合でもデータを取り出してリターンします。  
 MSG\_WAITALLが設定されている場合は、要求された長さ分のデータを受信するまでリターンしません。ただし、次の場合は要求より少ないデータでリターンすることがあります。  
 (1) コネクションが切断された場合  
 (2) ソケットにエラーが発生した場合

# recvfrom

## 受信データと送信者アドレス取得

### 機能

受信したデータと、その送信者の通信アドレス情報を取得します。  
recvfrom()は相手未設定のUDPソケットを対象としていますが、相手設定済みのUDPソケットやTCPソケットの場合にも使用できます。

p\_fromが示す領域はstruct sockaddr\_in型です。関数に渡す際にはstruct sockaddr型のポインタにキャストする必要があります。p\_fromの示す領域には送信者の通信アドレス情報を、fromlenの示す領域には送信者の通信アドレス情報の長さを返します。p\_fromにNULLを指定するとrecv()と同じ動作を行います。その場合は、\*p\_from および\*fromlenの領域には設定を行いません。

### コーディング

```
ssize_t
recvfrom(
    int
    void
    size_t
    int
    struct sockaddr
    socklen_t
)
sockfd,
*buffer,
length,
flags,
*p_from,
*fromlen
```

### コメント

ソケット識別子  
受信データを格納した領域の先頭アドレス  
受信データを格納した領域の長さ  
各種の特殊な受信動作を指定するフラグ  
送信者のアドレス情報を格納する領域  
送信者のアドレス情報を格納する領域の長さを格納した領域の先頭アドレス

### パラメータ

sockfd ソケット識別子

\*buffer 受信データを格納する領域の先頭アドレス

length 受信データを格納する領域の長さ

flags 受信動作を指定するフラグ

\*p\_from 受信動作を指定する領域

\*fromlen 送信者のアドレス情報を格納する領域の長さを格納した領域の先頭アドレス

### リターン値

リターン値は ssize\_t 型で返されます。詳細を下表に示します。

リターン値	内容
0	buffer に設定した受信データの長さ (正常終了)
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EINTR	4	割り込みエラー (待ち状態が強制的に解除された)
EBADF	9	識別子無効 (sockfd < 0)
EFAULT	14	ポインタアドレス不正 (p_fromまたはfromlenが4の倍数以外)
EINVAL	22	引数不正 指定したソケットが受動モード fromlenが構造体struct sockaddr_in型の長さとは異なる length=0またはlength 0x80000000 flagsが不正
EPIPE	32	指定したソケットに対しshutdown()が実行されている
EWOULDBLOCK	35	ノンブロッキングリターン (ソケットがノンブロッキングモードでかつ受信済みのデータがない)
ENOTSOCK	38	識別子ソケット未指定 (sockfdと同じ識別子のソケットが存在しない)
ECONNRESET	54	コネクションリセット (コネクションがリセットされた)
ENOTCONN	57	未接続エラー (sockfdの示すソケットが未接続)
EUSERS	68	重複操作 (指定したソケットを他のBSD関数が使用中)
EDEADLK	78	デッドロック条件 BSDで生成した通信端点またはポートを他のプロセスが使用中 BSDで生成した通信端点が他のプロセスによって削除された

## 詳細情報

パケットの構造は、connect()を参照して下さい。

p\_fromがNULL以外でfromlenが構造体struct sockaddr\_in型の長さとは異なる場合は、大域変数errnoにEINVALを設定し - 1 (エラー) を返します。

ソケット内に読み取られていない受信データがありその長さがlengthより小さい場合は、全てのデータを取り出してその取り出したデータの長さを返します。lengthより大きい場合は、length分のデータを取り出してその長さを返します。

ノンブロッキングモードがOFFの場合は、読み取られていない受信データが無いときは、データを受信するかエラーが発生するまでリターンしません。

ノンブロッキングモードがONで受信データがない場合は、大域変数errnoにEWOULDBLOCKを設定し - 1 (エラー) を返します。この場合は異常ではありません。

指定したソケットがTCPで接続されていない場合は、大域変数errnoにENOTCONNを設定し - 1 (エラー) を返します。

指定したソケットの受信側が、shutdown()によって閉じられている場合は、大域変数errnoにEPIPEを設定し - 1 (エラー) を返します。

指定したソケットがTCPで既にFINを受信しており、かつ受信バッファが空の場合は、正常終了として受信データ長0を返します (FINを受信しても受信データがあるうちは正常に受信できます)。

指定したソケットがTCPで既にコネクションがリセットされている場合は、大域変数errnoにECONNRESETを設定し - 1 (エラー) を返します。

TCPソケットの受信待ち状態でFINを受信し、かつ受信データが無い場合は、正常終了として受信データ長0を返します。

TCPソケットの受信待ち状態でコネクションがリセットされた(RST受信またはRST送信)場合は、大域変数errnoにECONNRESETを設定し - 1 (エラー) を返します。

他のプロセスによって待ち状態が強制的に解除された場合は、大域変数errnoにEINTRを設定し - 1 (エラー) を返します。

マルチタスク動作において、指定したソケットを他のプロセスから呼び出されたBSD関数が操作中の場合は、大域変数errnoにEUSERSを設定し - 1 (エラー) を返します。

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API上で動作しています。BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数errnoにEDEADLKを設定し - 1 (エラー) を返します。

flagsに設定する値に対応した動作を行うことができます。詳細については、recv()を参照してください。

# send

## ソケットへの送信データ設定

### 機能

ソケットを用いてデータを送信します。

### コーディング

```
ssize_t
send(
    int
    void
    size_t
    int
)
```

```
sockfd,
*buffer,
len,
flags
```

### コメント

ソケット識別子  
送信データを格納した領域の先頭アドレス  
送信データの長さ  
各種の特殊な送信動作を指定するフラグ

### パラメータ

**sockfd** ソケット識別子

**\*buffer** 送信データを格納した領域の先頭アドレス

**len** 送信データの長さ

**flags** 各種の特殊な送信動作を指定するフラグ

### リターン値

リターン値は `ssize_t` 型で返されます。詳細を下表に示します。

リターン値	内容
0	送信用ウィンドウバッファに書き込んだ送信データの長さ (正常終了)
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EINTR	4	割り込みエラー (待ち状態が強制的に解除された)
EBADF	9	識別子無効 ( <code>sockfd &lt; 0</code> )
ENOMEM	1 2	メモリ不足 (UDPソケットを生成するための受信バッファ資源が足りない)
EACCES	1 3	許可が無い (SO_BROADCASTがOFFで相手アドレスにブロードキャストを指定)
EINVAL	2 2	引数不正 指定したソケットが受動モード <code>len = 0</code> または <code>len &gt; 0x80000000</code> <code>flags</code> が不正
EPIPE	3 2	指定したソケットに対し <code>shutdown()</code> が実行されている
EWOULDBLOCK	3 5	ノンブロッキングリターン (ソケットがノンブロッキングモードでかつ送信バッファに空きがない)
ENOTSOCK	3 8	識別子ソケット未指定 ( <code>sockfd</code> と同じ識別子のソケットが存在しない)
EDESTADDRREQ	3 9	相手側通信アドレス情報未指定 ( <code>sockfd</code> の示すソケットが相手未設定のUDPソケット)
EADDRINUSE	4 8	通信アドレス情報使用中 (通信アドレス情報のポート番号が他のソケットにおいて使用されている)
ECONNRESET	5 4	コネクションリセット (コネクションがリセットされた)
ENOBUFS	5 5	利用可能エリア不足 (UDPの送信バッファが確保できない)
ENOTCONN	5 7	未接続エラー ( <code>sockfd</code> の示すソケットが未接続)

大域変数	エラー番号	内容
EPROCLIM	6 7	プロセス超過 (TCP/IPマネージャのUDP通信端点が足りない)
EUSERS	6 8	重複操作 (指定したソケットを他のBSD関数が使用中)
EDEADLK	7 8	デッドロック条件 BSDで生成した通信端点またはポートを他のプロセスが使用中 BSDで生成した通信端点が他のプロセスによって削除された

## 詳細情報

TCPでノンブロッキングモードがOFFの場合は、送信用ウィンドウバッファに対しlenの長さ分のデータを書き込み、書き込んだ長さを返します。lenの長さ分のデータ全てを書き込む領域がない場合は、lenの長さ分のデータ全てを書き込むかエラーが発生するまで内部で待ち状態となり、リターンしません。

TCPでノンブロッキングモードがONの場合は、送信用ウィンドウバッファに1バイトでも書き込んだ場合は、正常終了として書きこんだバイト数を返します。送信用ウィンドウバッファに空きがなく、1バイトも書き込めなかった場合は、大域変数errnoにEWOULDBLOCKを設定し - 1 (エラー) を返します。送信用ウィンドウバッファに書き込まれたデータはTCP/IPマネージャによって送信されます。

sockfdに相手未設定のUDPソケットを指定した場合は、大域変数errnoにEDESTADDRREQを設定し - 1 (エラー) を返します。

UDPでは送信用バッファにlenの長さ分のデータを書き込み、書き込んだ長さを返します。また、送信データが1回で送信可能なデータサイズより大きくて全てを書き込めない場合は、書き込める分のデータを書き込み、書き込んだ長さを返します(注1)。いずれの場合も書き込んだ分のデータは送信が行います。

UDPのソケットの場合、ノンブロッキングモードOFFでもブロックすることはありません。

UDPでノンブロッキングモードがOFFのときに送信用バッファに空きがなく、送信データを書き込めなかった場合は、大域変数errnoにENOBUFSを設定し - 1 (エラー) を返します。

UDPでノンブロッキングモードがONのときに送信用バッファに空きがなく、送信データを書き込めなかった場合は、大域変数errnoにEWOULDBLOCKを設定し - 1 (エラー) を返します。

指定したソケットがTCPで接続されていない場合は、大域変数errnoにENOTCONNを設定し - 1 (エラー) を返します。

指定したソケットの送信側が、shutdown()によって閉じられている場合は、大域変数errnoにEPIPEを設定し - 1 (エラー) を返します。

指定したソケットがTCPで既にコネクションがリセットされている場合は、大域変数errnoにECONNRESETを設定し - 1 (エラー) を返します。

TCPソケットの送信待ち状態でコネクションがリセットされた(RST受信またはRST送信)場合は、大域変数errnoにECONNRESETを設定し - 1 (エラー) を返します。

他のプロセスによって待ち状態が強制的に解除された場合は、大域変数errnoにEINTRを設定し - 1 (エラー) を返します。

マルチタスク動作において、指定したソケットを他のプロセスから呼び出されたBSD関数が操作中の場合は、大域変数errnoにEUSERSを設定し - 1 (エラー) を返します。

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API で動作しています。BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数errnoにEDEADLKを設定し - 1 (エラー) を返します。

flagsに設定する値に対応した動作を行うことができます。ただし、本BSDソケットAPIではMSG\_DONTWAITのみをサポートしています。

flagsに不正な値が設定された場合は、大域変数errnoにEINVALを設定し - 1 (エラー) を返します。

注1 : EMSGSIZE(40)のエラーは発生しません。また、要求したブロックの全データをIPで分割して送信することはありません。

send(), sendto() で flags に設定する値に

define名	値	内容	TCP	UDP
MSG_DONTWAIT	0x0040	1回操作毎のノンブロッキング指定		

**MSG\_DONTWAIT** : ノンブロッキングモードONとして動作します。

MSG\_DONTWAITが設定されていない場合は、ソケット自身のノンブロッキングモードに従って動作します。

MSG\_DONTWAITが設定されている場合は、関数内の処理はノンブロッキングモードONとして動作します。

# sendto

## 送信データと送信先アドレス設定

### 機能

指定された通信アドレスに対し送信を行います。

### コーディング

```
ssize_t
sendto(
    int
    void
    size_t
    int
    struct sockaddr
    int
)
```

```
sockfd,
*p_msg,
msglen,
flags,
*p_to,
tolen
```

### コメント

ソケット識別子  
送信データを格納した領域のポインタ  
送信データの長さ  
各種の特殊な送信動作を指定するフラグ  
送信先の通信アドレス情報を格納した領域の先頭アドレス  
送信先の通信アドレス情報を格納した領域の長さ

### パラメータ

**sockfd** ソケット識別子

**\*p\_msg** 送信データを格納した領域のポインタ

**msglen** 送信データの長さ

**flags** 各種の特殊な送信動作を指定するフラグ

**\*p\_to** 送信先の通信アドレス情報を格納した領域の先頭アドレス

**tolen** 送信先の通信アドレス情報を格納した領域の長さ

### リターン値

リターン値は `ssize_t` 型で返されます。詳細を下表に示します。

リターン値	内容
0	送信用ウィンドウバッファに書き込んだ送信データの長さ (正常終了)
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EINTR	4	割り込みエラー (待ち状態が強制的に解除された)
EBADF	9	識別子無効 ( <code>sockfd &lt; 0</code> )
ENOMEM	1 2	メモリ不足 (UDPソケットを生成するための受信バッファ資源が足りない)
EACCES	1 3	許可が無い ( <code>SO_BROADCAST</code> がOFFで相手アドレスにブロードキャストを指定)
EINVAL	2 2	引数不正 指定したソケットが受動モード <code>msglen = 0</code> または <code>msglen 0x80000000</code> <code>flags</code> が不正 <code>tolen</code> が構造体 <code>struct sockaddr_in</code> 型の長さとは異なる
EPIPE	3 2	指定したソケットに対し <code>shutdown()</code> が実行されている

大域変数	エラー番号	内容
EWOLDBLOCK	3 5	ノンブロッキングリターン (ソケットがノンブロッキングモードでかつ送信バッファに空きがない)
ENOTSOCK	3 8	識別子ソケット未指定 (sockfdと同じ識別子のソケットが存在しない)
EDESTADDRREQ	3 9	相手側通信アドレス情報未指定 (sockfdの示すソケットが相手未設定のUDPソケット、かつ指定した相手側通信アドレスが不正)
EADDRINUSE	4 8	通信アドレス情報使用中 (通信アドレス情報のポート番号が他のソケットにおいて使用されている)
ECONNRESET	5 4	コネクションリセット (コネクションがリセットされた)
ENOBUFS	5 5	利用可能エリア不足 (UDPの送信バッファが確保できない)
EISCONN	5 6	接続済みソケット指定
ENOTCONN	5 7	未接続エラー (sockfdの示すソケットが未接続)
EPROCLIM	6 7	プロセス超過 (TCP/IPマネージャのUDP通信端点が足りない)
EUSERS	6 8	重複操作 (指定したソケットを他のBSD関数が使用中)
EDEADLK	7 8	デッドロック条件 BSDで生成した通信端点またはポートを他のプロセスが使用中 BSDで生成した通信端点が他のプロセスによって削除された

## 詳細情報

パケットの構造は、connect()を参照して下さい。

送信先を指定する場合、p\_toが示す領域はstruct sockaddr\_in型です。関数に渡す際にはstruct sockaddr型のポインタにキャストする必要があります。

送信先を指定しない場合、p\_toにNULLを指定すると、send()と同じ動作を行います。

p\_toがNULL以外でtolenが構造体struct sockaddr\_in型の長さとは異なる場合は、大域変数errnoにEINVALを設定し-1 (エラー) を返します。

### (1) sockfdが相手未設定のUDPソケットを指している場合

送信バッファにmsglenの長さ分のデータを書き込み、書き込んだ長さを返します。また、送信データが1回で送信可能なデータサイズより大きくて全てを書き込めない場合は、書き込める分のデータを書き込み、書き込んだ長さを返します(注1)。いずれの場合も書き込んだ分のデータは送信を行います。

ノンブロッキングモードがOFFのときに送信バッファに空きがなく、送信データを書き込めなかった場合は、大域変数errnoにENOBUFSを設定し-1 (エラー) を返します。

ノンブロッキングモードがONのときに送信バッファに空きがなく、送信データを書き込めなかった場合は、大域変数errnoにEWOLDBLOCKを設定し-1 (エラー) を返します。

送信先通信アドレス情報 (p\_to) を指定しない (NULL) 場合は、大域変数errnoにEDESTADDRREQを設定し-1 (エラー) を返します。

### (2) sockfdが相手設定済みのUDPソケットまたはTCPソケットを指している場合

送信先通信アドレス情報 (p\_to) を指定しない (NULL) 場合は、send()と同様の動作を行います。

送信先通信アドレス情報 (p\_to) を指定した (NULL以外) 場合は、大域変数errnoにEISCONNを設定し-1 (エラー) を返します。

指定したソケットがTCPで接続されていない場合は、大域変数errnoにENOTCONNを設定し-1 (エラー) を返します。

指定したソケットの送信側が、shutdown()によって閉じられている場合は、大域変数errnoにEPIPEを設定し-1 (エラー) を返します。

指定したソケットがTCPで既にコネクションがリセットされている場合は、大域変数errnoにECONNRESETを設定し-1 (エラー) を返します。

注1 : EMSGSIZE(40)のエラーは発生しません。  
また、要求したブロックの全データをIPで分割して送信することはありません。

TCPソケットの送信待ち状態でコネクションがリセットされた(RST受信またはRST送信)場合は、大域変数errnoにECONNRESETを設定し - 1 (エラー) を返します。  
他のプロセスによって待ち状態が強制的に解除された場合は、大域変数errnoにEINTRを設定し - 1 (エラー) を返します。

マルチタスク動作において、指定したソケットを他のプロセスから呼び出されたBSD関数が操作中の場合は、大域変数errnoにEUSERSを設定し - 1 (エラー) を返します。

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API上で動作しています。BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数errnoにEDEADLKを設定し - 1 (エラー) を返します。

flagsに設定する値に対応した動作を行うことができます。  
詳細については、send()を参照してください。

# closesocket

## ソケットのクローズ

### 機能

通信を終了し、ソケットを閉じます。

### コーディング

```
int
closesocket(
    int          sockfd
)
```

### コメント

ソケット識別子

### パラメータ

sockfd  
ソケット識別子

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 ( sockfd < 0 )
EWOULDBLOCK	3 5	ノンブロッキングリターン (ノンブロッキングモードでかつSO_LINGERがONの場合に、リガ-時間が経過した)
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfdと同じ識別子のソケットが存在しない)

### 詳細情報

送信バッファ内に残っているデータは全て送信し、未読の受信データは破棄します。  
TCPソケットの場合は、送信バッファ内のデータを全て送信した後にFINを送ります。  
送信データ、受信データに対する処理の終了後、ソケットを未生成の状態にします。

SO\_LINGERオプションがOFFの場合は、クローズ処理を開始後直ちに、0 (正常終了) を返します。

SO\_LINGERオプションがONの場合は、リガ-時間が経過するまでclosesocket()からリターンしません。リガ-時間が経過した場合、ノンブロッキングモードがOFFの場合は、0 (正常終了) を返しますが、ノンブロッキングモードがONの場合は、大域変数errnoにEWOULDBLOCK設定し - 1 (エラー) を返します。

TCPソケットではSO\_LINGERオプションがONのとき、リガ-時間に0を指定した場合と、リガ-時間が経過した場合には、送信バッファ内に残っているデータを破棄してRSTを送信します。

# shutdown

## コネクション閉鎖

### 機能

コネクションを部分的に閉鎖します。

### コーディング

```
int
shutdown(
    int
    int
)
```

sockfd,  
direction

### コメント

ソケット識別子  
切断方向

### パラメータ

sockfd           ソケット識別子

direction        切断方向

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 ( sockfd < 0 )
EINVAL	2 2	引数不正 ( directionの値が対象外 )
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfdと同じ識別子のソケットが存在しない )
ENOTCONN	5 7	ソケット未接続
EUSERS	6 8	重複操作 ( 指定したソケットを他のBSD関数が使用中 )
EDEADLK	7 8	デッドロック条件 ( BSDで生成した通信端点が他のプロセスによって削除された )

**詳細情報**

切断方向ごとの動作は以下になります。接続していないソケットに使用した場合は、大域変数errnoにENOTCONNを設定し-1(エラー)を返します。

**SHUT\_RD(0)の場合:**

コネクションの読み出し側をクローズし、受信バッファ内のデータやその後到着したデータを破棄します。TCPソケットの場合、到着データに対しては応答(ACK)を返しますが、データ自体は破棄します。

**SHUT\_WR(1)の場合:**

コネクションの書き込み側をクローズし、それ以後データ送信を行いません。送信バッファに残っているデータがあれば、そのデータは全て送信します。TCPソケットの場合、送信バッファ内データを全て送信した後にFINを送信します。

**SHUT\_RDWR(2)の場合:**

コネクションの読み出し側、書き込み側の双方をクローズし、それ以後、データの送受信を行えなくします。

マルチタスク動作において、指定したソケットを他のプロセスから呼び出されたBSD関数が操作中の場合は、大域変数errnoにEUSERSを設定し-1(エラー)を返します。

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API上で動作しています。BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数errnoにEDEADLKを設定し-1(エラー)を返します。

# getsockopt

## プロトコルのオプション取得

### 機能

プロトコルに対するオプションの値を読み出します。

ソケットから取得したオプションデータをp\_optvalの示す領域に返し、オプションデータの長さをoptlenに返します。

オプションには、特定の機能の有効/無効を設定、取得するオプション（フラグ型オプション）と、特定の値の設定、取得を行うオプション（数値型オプション）の2種類に分けられます。

フラグ型オプションの場合は、オプションデータが0ならオプションが無効になっていることを示し、0以外ならオプションが有効になっていることを示します。

### コーディング

```
int
getsockopt(
    int
    int
    int
    void
    int
)
```

```
sockfd,
level,
opt,
*p_optval,
*optlen
```

### コメント

```
ソケット識別子
プロトコルレベル
オプション
オプションを格納する領域の先頭アドレス
オプションを格納する領域の長さを格納する領域の
先頭アドレス
```

### パラメータ

```
sockfd      ソケット識別子
level       プロトコルレベル
opt         オプション
*p_optval   オプションを格納する領域の先頭アドレス
*optlen     オプションを格納する領域の長さを格納する領域の先頭アドレス
```

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 ( sockfd < 0 )
EFAULT	1 4	ポインタアドレス不正 ( p_optval もしくはoptlenが4の倍数以外 )
EINVAL	2 2	引数不正 ( optlenが短い )
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfdと同じ識別子のソケットが存在しない )
ENOPROTOPT	4 2	プロトコル利用不可 ( level、optの組み合わせが不正 )

## 詳細情報

以下に、本BSDソケットAPIにおいて使用するソケットオプションを示します。

levelとoptの組み合わせが正しくない場合は、大域変数errnoにENOPROTOOPTを設定し-1（エラー）を返します。

level	Opt	使用可能な関数	内容	オプションの型	データ型
SOL_SOCKET	SO_REUSEADDR (0x0004)	getsockopt setsockopt	自通信アドレスの再利用の許可	フラグ型	int型
	SO_KEEPALIVE (0x0008)	getsockopt setsockopt	コネクションが有効か定期的な検査の実行	フラグ型	int型
	SO_LINGER (0x0080)	getsockopt setsockopt	クローズ時のリンガー動作実行	数値型	linger型
	SO_BROADCAST (0x0020)	getsockopt setsockopt	ブロードキャストメッセージの転送を許可（UDPのみ）	フラグ型	int型
	SO_SNDBUF (0x1001)	getsockopt setsockopt	出力バッファの大きさ	数値型	int型
	SO_RCVBUF (0x1002)	getsockopt setsockopt	入力バッファの大きさ	数値型	int型
	SO_TYPE (0x1008)	getsockopt	ソケットの型	数値型	int型
	SO_ERROR (0x1007)	getsockopt	ソケットの最新のエラー（取得によりクリア）	数値型	int型
IPPROTO_TCP	TCP_KEEPALIVE (0x02)	getsockopt setsockopt	TCPキープアライブプローブ送信までの待ち時間（単位は秒）	数値型	int型
	TCP_NODELAY (0x01)	getsockopt setsockopt	Nagle アルゴリズムの禁止	フラグ型	int型

### (1) SO\_REUSEADDR (0x0004) 自通信アドレスの再利用の許可

このオプションは、自分の通信アドレス情報重複許可のON/OFFを示します。

このオプションがOFFの場合は、同一のポートの通信アドレス情報は使用できません。bind()により既に存在するポート番号を指定すると大域変数errnoにEADDRINUSEを設定して-1（エラー）を返します。

このオプションがONの場合、自分の通信アドレス情報の重複を可能にします。TCPとUDPでは働きが異なります。

TCPソケットの場合、能動モードのソケットにおいて、ポート番号、IPアドレスともに同一の自通信アドレスを複数使用できるようにします。受動モードのソケットは、ポート番号、IPアドレスともに同一にすることはできません。自通信アドレスがすでに存在する受動モードソケットと同一の能動モードのソケットを生成することはできますが、この能動モードのソケットに対しlisten()をコールすると大域変数errnoにEADDRINUSEを設定して-1（エラー）を返します。また、能動モードのソケットの場合でも自通信アドレスと相手側通信アドレスの両方を同じにすることはできません。connect()により自通信アドレスと相手側通信アドレスが同一のソケットが生成される場合は、大域変数errnoにEINVALを設定して-1（エラー）を返します。

UDPソケットの場合、複数の自分の通信アドレス情報を同一のポートで使用できるようにします。ただし、通信アドレス情報毎のIPアドレスは異なっている必要があります。bind()によりポート番号、IPアドレスともに同一の自通信アドレスを指定すると大域変数errnoにEADDRINUSEを設定し-1（エラー）を返します。

**( 2 ) SO\_KEEPALIVE ( 0x0008 )      コネクションが有効か定期的な検査の実行**

このオプションは、TCPソケットのコネクション定期的に検査する機能のON/OFFを示します。このオプションがONかつソケットが接続されている場合、そのソケットによる送信受信ともに行われないう状況が2時間以上続いたなら、そのTCPソケットは自動的に相手に向けてキープアライブプローブを送信します。このキープアライブプローブに対しての相手側の対応によって次の処理を行います。

相手がACKで応答

正常な状態であるのでアプリケーションに対しなにも通知しません。送信受信ともないまま再び2時間経過したら再びキープアライブプローブを送信します。

相手からの応答が無い

この場合、75秒経過するごとにキープアライブプローブの再送をおこないます。

8回再送しても応答が無い場合、RSTを送信してコネクションを切断します。

**( 3 ) SO\_LINGER ( 0x0080 )      クローズ時のリンガー動作実行**

このオプションは、TCPソケットにおけるクローズ処理のリンガー動作の内容を示します。このオプションデータは、構造体linger型（以下に示す）のデータです。この型のデータをp\_optvalを通じて参照、設定します。

```
struct linger {
    int    l_onoff;      // 0: リンガーオプションOFF、0以外: リンガーオプションON
    int    l_linger;    // リンガー時間単位は[1/100秒]
};
```

オプションデータの内容によってclosesocket()の動作が異なります。オプションデータ毎のclosesocket()の動作を次に示します。

l\_onoffが0の場合

l\_lingerの値は無視します。

closesocket()からすぐにリターンしますが、送信バッファ中のデータを送信し終わるのを待ってFINを送ります。

l\_onoffが0以外、かつl\_lingerが0の場合

closesocket()からすぐにリターンし、直ちにRSTを送信します。未送信のデータは破棄します。l\_onoffが0以外、かつl\_lingerが0以外の場合closesocket()からすぐにはリターンしません。送信バッファ中のデータを送信し終わるのを待ってFINを送りますが、l\_lingerの指定する時間を経過した場合は、RSTを送信して未送信のデータを破棄します。

**( 4 ) SO\_BROADCAST ( 0x0020 )      ブロードキャストメッセージの転送の許可**

このオプションは、ブロードキャストメッセージの転送能力の有効/無効を示します。ブロードキャストはUDPでのみサポートします。このオプションがONの時にブロードキャストメッセージの転送が有効となります。

このオプションがOFFの状態ではデータの送信を行う(sendto()等の関数を使用する)時、相手側通信アドレス情報としてブロードキャストアドレスが設定されると、大域変数errnoにEACCESを設定して-1(エラー)を返します。

**( 5 ) SO\_SNDBUF ( 0x1001 ) 出力バッファの大きさ**

このオプションは、ソケットの送信バッファサイズを示します。このオプションはint型のデータであり、p\_optvalを通じて参照、設定します。設定する値は2048以上である必要があります。UDPのソケットに対してこのオプションを指定することはできません。

**( 6 ) SO\_RCVBUF ( 0x1002 ) 入力バッファの大きさ**

このオプションは、ソケットの受信バッファサイズを示します。このオプションはint型のデータで、p\_optvalを通じて参照、設定します。設定する値は2048以上である必要があります。UDPソケットの場合、データグラム受信キューの数を示します。

**( 7 ) SO\_TYPE ( 0x1008 ) ソケットの型**

このオプションを指定してgetsockopt()をコールすることで、ソケットの型の参照を行います。このオプションはint型のデータで、SOCK\_STREAM(1) (TCPの場合)、SOCK\_DGRAM(2) (UDPの場合)の値を取ります。このオプションは、setsockopt()では、使用できません。

**( 8 ) SO\_ERROR ( 0x1007 ) ソケットの最近のエラー取得**

このオプションを指定してgetsockopt()をコールすることで、ソケット内の変数so\_errorの値を取得することができます。このとき、getsockopt()は変数so\_errorを0にクリアします。このオプションは、setsockopt()では、使用できません。

**( 9 ) IP\_TOS ( 0x02 ) IPヘッダ内のTOSの値**

このオプションは、IPヘッダ内のTOSの値を参照または設定します。このオプションはint型のデータで、p\_optvalを通じて参照、設定します。有効なデータの範囲は8ビットで0~255です。下位8ビット以外の設定データは無視します。sockfdにより指定したソケットの送信するIPヘッダのTOSだけが対象となります。

**( 1 0 ) IP\_TTL ( 0x03 ) IPヘッダ内のTTLの値**

このオプションは、IPヘッダ内のTTLの値を参照または設定します。このオプションはint型のデータで、p\_optvalを通じて参照、設定します。有効なデータの範囲は8ビットで1~255です。下位8ビット以外の設定データは無視します。sockfdにより指定したソケットの送信するIPヘッダのTTLだけが対象となります。

**( 1 1 ) TCP\_KEEPAALIVE ( 0x02 ) キープアライブプローブ送信までのアイドル時間**

このオプションは、TCPキープアライブプローブ送信までのアイドル時間を秒単位で設定します。このオプションはint型のデータで、p\_optvalを通じて参照、設定します。また、SO\_KEEPAALIVEオプションがOFFのときは、このオプションは無効です。時間の単位は秒です。sockfdにより指定したソケットだけアイドル時間が変更されます。

**( 1 2 ) TCP\_NODELAY ( 0x01 ) Nagleアルゴリズムの禁止**

このオプションは、TCP Nagleアルゴリズムの禁止のON/OFFを示します。sockfdにより指定したソケットだけNagleアルゴリズムがON/OFFされます。

# setsockopt

## プロトコルのオプション変更

### 機能

プロトコルに対するオプションの値を設定します。  
 オプションの設定値をp\_optvalの示す領域から取得しソケットに設定します。  
 フラグ型オプションの場合、オプションの設定値が0なら、オプションを無効にし、0以外ならオプションを有効にします。ソケットオプションについては、getsockopt()を参照してください。

### コーディング

```
int
setsockopt(
    int
    int
    int
    void
    int
)
```

```
sockfd,
level,
opt,
*p_optval,
opt len
```

### コメント

```
ソケット識別子
プロトコル
オプション
オプションデータを格納する領域の先頭アドレス
オプションデータの長さ
```

### パラメータ

```
sockfd      ソケット識別子
level       プロトコル
opt         オプション
*p_optval   オプションデータを格納する領域の先頭アドレス
opt len     オプションデータの長さ
```

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 ( sockfd < 0 )
EFAULT	1 4	ポインタアドレス不正 ( p_optval が 4 の倍数以外 )
EINVAL	2 2	引数不正 ( opt len が短い )
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfd と同じ識別子のソケットが存在しない )
ENOPROTOOPT	4 2	プロトコル利用不可 ( opt が不正 )
ENOBUFS	5 5	利用可能エリア不足 ( バッファが確保できない )
EPROTO	8 6	プロトコルエラー ( 使用中の通信端点に対し実行できない )

### 詳細情報

level と opt 組み合わせが正しくない場合、およびオプションとして SO\_TYPE または SO\_ERROR が指定された場合は、大域変数 errno に ENOPROTOOPT を設定し -1 (エラー) を返します。

BSDソケットAPIが生成したTCP/IPマネージャの通信端点を既に使用中で、指定したオプションの設定ができない状態の場合は、大域変数errnoにEPROTOを設定し-1(エラー)を返します。(SO\_KEEPALIVE、SO\_SNDBUF、SO\_RCVBUF、IP\_TOS、IP\_TTL、TCP\_KEEPALIVE、TCP\_NODELAYの各オプションはlisten()またはconnect()後のTCPソケットには使えません)

# selectsocket

## ソケットの利用可能待ち

### 機能

ソケットに対する複数のイベントのいずれかが発生するのを待ちます。UNIX系で使用されているselect()の  
コマンド名でも使用できます。

### コーディング

```
int
selectsocket (
    int
    fd_set
    fd_set
    fd_set
    struct timeval
)
```

```
maxfdp1,
*readset,
*writeset,
*exceptset,
*timeout
```

### コメント

検査するディスクリプタの範囲  
受信用ディスクリプタを指定した領域のアドレス  
送信用ディスクリプタを指定した領域のアドレス  
例外用ディスクリプタを指定した領域のアドレス  
タイムアウト時間を格納した領域のアドレス

### パラメータ

**maxfdp1** 検査するディスクリプタの範囲

**\*readset** 受信用ディスクリプタを指定した領域のアドレス

**\*writeset** 送信用ディスクリプタを指定した領域のアドレス

**\*exceptset** 例外用ディスクリプタを指定した領域のアドレス

**\*timeout** タイムアウト時間を格納した領域のアドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
1	正常終了
0	タイムアウト
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EINTR	4	割り込みエラー (timeoutで指定したタイムアウト時間を経過した)
EBADF	9	識別子ソケット未指定 (ディスクリプタに対応するソケットが未生成)
EFAULT	1 4	ポインタアドレス不正 (readset、writeset、exceptset、timeoutが4の倍数以外)
EINVAL	2 2	引数不正 (maxfdp1 < 0 またはmaxfdp1 > FD_SETSIZE(97)、0 > tv_sec、 0 tv_usec 999999 以外)

## 詳細情報

イベントの発生が無い場合は、指定時間が経過するまで待ちます。

maxfdp1は、指定するソケット識別子の最大値を指定します（ソケット識別子最大番号プラス1の値）。実際には、maxfdp1の値をソケット識別子の最大値プラス1の値として取得します（ソケット識別子の最大値は、maxfdp1-1の値である）。

readset、writerset、exceptsetの領域内のデータは、それぞれ受信可能、送信可能及び例外状態発生を検査するソケットを複数指定します。本BSDソケットAPIでは、例外状態に対応する状態は次の1つのみです。

### ソケットが緊急データを受信した

readset、writerset、exceptsetの領域内のデータは、unsigned long型であり各ビットがそれぞれのソケット識別子を指定します（例えば、ソケット識別子が5のソケットを指定する場合はデータのbit 5をセットします）。

対象となるビットは、maxfdp1により決まります。maxfdp1の指定するビットより大きいビットは無視します。

selectsocket()がコールされたときには、readset、writerset、exceptsetの領域から検査するソケットの指定を受け取ります。また、リターンする際にはreadset、writerset、exceptsetの領域に対し、受信可能、送信可能及び例外状態発生となったソケットを指定する値を設定します。

リターン値は、readset、writerset、exceptsetの領域に設定されたデータのビットの総計です（例えば、ソケット識別子5のソケットのみ利用可能でかつそのソケットが受信、送信とも可能な場合、リターン値は2となります）。

\*timeoutには、タイムアウト時間を指定します。timeoutがNULLの場合は、利用可能なソケットが発生するまで待ち続けます。

selectsocket()により接続中のTCPソケットを検査すると、受信不可かつ送信不可の状態という結果になります。接続の試みが終了すると（正常、異常問わず）、受信可能や送信可能（あるいは、両方とも可能）になります。そのため、selectsocket()により対象のソケットの受信可能/不可と送信可能/不可を検査することで接続の試みの終了を確認することができます。

接続の確立が成功したかどうかは、ソケットオプションSO\_ERRORによりソケットのエラーを調べることで確認できます。ソケットのエラーが正常終了を示している（errno=0）なら接続の確立に成功しています。

### パケットの構造

#### < ディスクリプタ情報の構造体 >

```
struct{
    fd_mask    fds_bits[4];
} fd_set;
```

#### < タイムアウト時間の構造体 >

```
struct timeval{
    long tv_sec;      // 秒単位の値
    long tv_usec;    // μ秒単位の値
};
```

# set\_blocking\_socket

## ブロッキングモード設定

### 機能

指定したソケットのブロッキングモードを設定します。

### コーディング

```

int
set_blocking_socket(
    int          sockfd,
    int          blocking
)

```

### コメント

ソケット識別子  
設定するブロッキングモード

### パラメータ

**sockfd**           ソケット識別子

**blocking**        設定するブロッキングモード  
( 0 : ブロッキング、 1 : ノンブロッキング )

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 ( sockfd < 0 )
EINVAL	2 2	引数不正 ( blocking が 0 , 1 以外 )
ENOTSOCK	3 8	識別子ソケット未指定 ( sockfdと同じ識別子のソケットが存在しない )
EUSERS	6 8	重複操作 ( 指定したソケットを他のBSD関数が使用中 )

### 詳細情報

blocking が 0 , 1 以外の値の場合は、リターン値としてEINVALを返します。

## get\_errno

## エラーコード取得

### 機能

BSDソケットAPI全体で発生した最新のエラーコードを取得します。

### コーディング

```
int  
get_errno(  
    void  
)
```

### コメント

なし

### パラメータ

なし

### リターン値

errno (大域変数) に設定されていたエラーコード。

### 詳細情報

## get\_thread\_errno

## スレッドエラーコード取得

### 機能

BSDソケットAPIを使用したタスク毎に発生した最新のエラーコードを取得します。  
 threadidに0を指定した場合は、自タスクのエラーコードを取得します。  
 取得したタスクのエラーコード(大域変数)は0にクリアされます。

### コーディング

```
int
get_thread_errno(
  short threadid
)
```

### コメント

スレッド(タスク)ID

### パラメータ

threadid  
 スレッド(タスク)ID

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
正の値	正常終了(タスクの大域変数(エラーコード))
0	正常終了(タスクの大域変数(エラーなし))
-1	エラー(関数実行不可)

### 詳細情報

# set\_sock\_timewait

## TIMEWAIT 時間の変更

### 機能

指定したソケットの TIMEWAIT 時間を変更します。

### コーディング

```
int
set_sock_timewait(
    int
    short
)
```

sockfd,  
socktimewait

### コメント

ソケット識別子  
TIMEWAIT時間 (秒)

### パラメータ

sockfd  
ソケット識別子

socktimewait  
TIMEWAIT時間 (秒)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常処理
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 (sockfd < 0 )
EINVAL	2 2	引数不正 (socktimewait < 1 または socktimewait > 240 )
ENOTSOCK	3 8	識別子ソケット未指定 (sockfdと同じ識別子のソケットが存在しない)
EOPNOTSUPP	4 5	未サポートソケット (指定したソケットがTCPのソケットではない)
EDEADLK	7 8	デッドロック条件 (BSDで生成した通信端点が他のプロセスによって削除された)
EPROTO	8 6	プロトコルエラー (使用中の通信端点に対し実行できない)

### 詳細情報

TIMEWAIT時間として指定できる範囲は1秒から4分です。socktimewaitが指定できる範囲の値でない場合は、大域変数errnoにEINVALを設定し-1(エラー)を返します。

指定されたソケットがTCPでない場合は、大域変数errnoにEOPNOTSUPPを設定し-1(エラー)を返します。

BSDソケットAPIが生成したTCP/

IPマネージャの通信端点を既に使用中で、TIMEWAIT時間の変更ができない状態の場合は、大域変数errnoにEPROTOを設定し-1(エラー)を返します。(listen()またはconnect()後は使えません)

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API上で動作しています。BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数errnoにEDEADLKを設定し-1(エラー)を返します。

# get\_sock\_recvlen

## 受信データ長の取得

### 機能

ソケットが受信しているデータ長を取得します。  
 TCPソケットの場合は、受信用ウィンドバッファ内の有効な受信セグメント長を返します。  
 UDPソケットの場合は、キューイングされている全てのデータグラムを合計した長さを返します。

### コーディング

```
int
get_sock_recvlen(
    int
```

sockfd

### コメント

ソケット識別子

### パラメータ

sockfd

ソケット識別子

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了 (受信データの長さ)
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 (sockfd < 0)
EPIPE	3 2	指定したソケットに対しshutdown()が実行されている
ENOTSOCK	3 8	識別子ソケット未指定 (sockfdと同じ識別子のソケットが存在しない)
ENOTCONN	5 7	未接続エラー (sockfd の示すソケットが未接続)
EDEADLK	7 8	デッドロック条件 (BSDで生成した通信端点が他のプロセスによって削除された)

### 詳細情報

指定したソケットがTCPで接続されていない場合は、大域変数errnoにENOTCONNを設定し - 1 (エラー) を返します。

指定したソケットの受信側が、shutdown()によって閉じられている場合は、大域変数errnoにEPIPEを設定し - 1 (エラー) を返します。

本BSDソケットAPIはTCP/IPマネージャのITRON TCP/IP API上で動作しています。BSDソケットAPIがITRON TCP/IP API上で生成した通信端点や受付口を他のプロセスがITRON TCP/IP APIを用いて直接操作したことにより、BSDソケットAPIの正常な動作が保証できない状態に陥った場合は、大域変数errnoにEDEADLKを設定し - 1 (エラー) を返します。

# set\_sock\_keepalive

## キープアライブ制御情報の変更

### 機能

指定したソケットのキープアライブプローブ機能の設定を変更します。

### コーディング

```
int
set_sock_keepalive(
    int
    T_BSD_KEEPALIVE
)
```

```
sockfd,
*p_ktbl
```

### コメント

ソケット識別子  
キープアライブ制御情報テーブル

### パラメータ

sockfd           ソケット識別子

\*p\_ktbl           キープアライブ制御情報テーブル

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常処理
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。(各エラー番号の前に「BSD\_」を付けた形で定義されています。)

大域変数	エラー番号	内容
EPERM	1	関数実行不可
EBADF	9	識別子無効 (sockfd < 0 )
EINVAL	2 2	引数不正 (keepalive < 1 , kretrytime < 1 , kretrycnt < 1)
ENOTSOCK	3 8	識別子ソケット未指定 (sockfdと同じ識別子のソケットが存在しない)
EOPNOTSUPP	4 5	未サポートソケット (指定したソケットがTCPのソケットではない)
EDEADLK	7 8	デッドロック条件 (BSDで生成した通信端点が他のプロセスによって削除された)
EPROTO	8 6	プロトコルエラー (使用中の通信端点に対し実行できない)

### 詳細情報

以下にパケットの構造を示します。

```
typedef struct {
    short    k_onoff;      // キープアライブ機能スイッチ(0:OFF/1:ON)
    short    keepalive;   // キープアライブパケット開始タイマ(秒)
    short    kretrytime;  // キープアライブパケットのリトライ間隔(秒)
    short    kretrycnt;   // キープアライブパケットのリトライ回数
} T_BSD_KEEPALIVE;
```

keepalive , kretrytime , kretrycnt は1以上を指定して下さい。

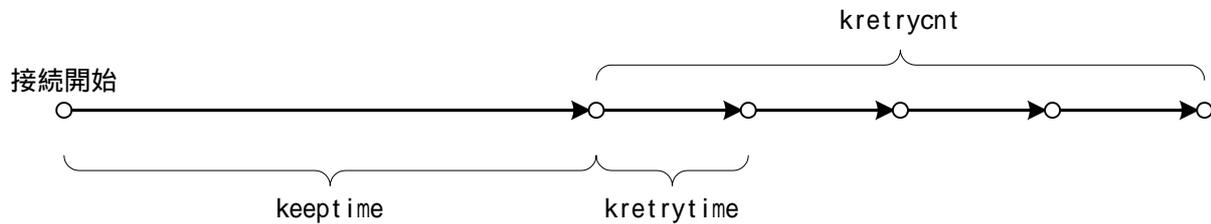
k\_onoff には、キープアライブ機能の有効/無効を設定します。k\_onoff に 0 を指定するとキープアライブ機能は無効になり、keepalive\_time, keepalive\_retry\_time, keepalive\_retry\_cnt に指定した値は無視します。k\_onoff に 1 (0 以外の値) を指定するとキープアライブ機能が有効になります。

k\_onoff に 1 (0 以外の値) を指定した場合は、keepalive\_time, keepalive\_retry\_time, keepalive\_retry\_cnt には 1 以上の値を指定して下さい。keepalive\_time, keepalive\_retry\_time, keepalive\_retry\_cnt に 0 以下の値を指定した場合は、大域変数 errno に EIVAL を設定し「- 1」(エラー)を返します。

指定されたソケットが TCP でない場合は、大域変数 errno に EOPNOTSUPP を設定し「- 1」(エラー)を返します。

BSDソケットAPIが生成したTCP/IPマネージャの通信端点を既に使用中で、キープアライブ機能の設定変更ができない場合は、大域変数 errno に EPROTO を設定し「- 1」(エラー)を返します(listen()またはconnect()後は使用できません)。

キープアライブプローブは、TCP通信の接続が確立されてから開始されます。設定値のデフォルトは、それぞれ keepalive\_time : 7200秒(2時間), keepalive\_retry\_time : 75秒, keepalive\_retry\_cnt : 8回です。



# WriteVideoLUT

## ビデオ入力LUTへのデータ書き込み

### 機能

ビデオ入力濃度変換メモリへのデータ書き込みを行います。

### コーディング

```
int
WriteVideoLUT(
    DEVID  devID ,
    int    *P0lut ,
    int    *P1lut ,
    int    leng,
    int    mode
)
```

### コメント

デバイスID  
P0LUTへ書き込むデータテーブル  
P1LUTへ書き込むデータテーブル  
書き込みデータ数  
モード

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*P0lut**

P0LUTへ書き込むデータテーブル。sizeof(int)\*leng分領域を確保し、データを設定して下さい。

**\*P1lut**

P1LUTへ書き込むデータテーブル。sizeof(int)\*leng分領域を確保し、データを設定して下さい。  
SVP-330は、未サポートです。

**leng**

書き込みデータ数。通常カメラの場合、8ビット入力で256、カメラリンク10ビットの場合は1024を指定して下さい。

**mode**

書き込みを行う濃度変換テーブルのバンク指定

LUTバンク	定数	内容
VIDEO_LUT_BANK0	0	LUTバンク#0
VIDEO_LUT_BANK1	1	LUTバンク#1 (SVP-330は未サポート)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20, 21	パラメータ設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetVideoOpt

## ビデオオプションの設定

### 機能

ビデオオプションの設定を行います。

### コーディング

```

int
SetVideoOpt(
    DEVID  devID ,
    int    mode ,
    int    opt ,
    void   *optval ,
    int    optlen
)

```

### コメント

デバイスID  
 設定モード  
 オプション  
 設定するオプションデータ領域のポインタ  
 オプションデータ領域のサイズ

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**mode**

設定モード。

設定モード	定数	内容
VIDEO_OPT_VP0	0	ビデオポート# 0のオプション指定
VIDEO_OPT_VP1	1	ビデオポート# 1のオプション指定
VIDEO_OPT_DP	2	表示モードのオプション指定

**opt**

オプション。modeにより、設定値が異なります。詳細情報を参照して下さい。

**\*optval**

設定するオプションデータ領域のポインタ。mode,optにより、設定値が異なります。詳細情報を参照して下さい。

**optlen**

オプションデータ領域のサイズ(バイトサイズ)。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20, 21	パラメータ設定範囲外
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

以下に、オプションの設定値を示します。

mode	opt	内容	データ型	
VIDEO_OPT_VPO VIDEO_OPT_VP1	VIDEO_LUT_CTL ( 0 )	濃度変換モード指定	VIDEO_OPT_LUTCTL 型	
	MIRROR_CTL ( 1 )	ミラー映像入力指定		int 型
		ミラーモード	定数	
NO_MIRROR		0	通常取込	
H_MIRROR		1	水平方向ミラー反転	
V_MIRROR	2	垂直方向ミラー反転		
HV_MIRROR	3	全ミラー反転		
	POWER_SAVE ( 0x100 )	サポートされていません。	——	
VIDEO_OPT_DP	DP_SLAVE_CTL ( 0 )	表示モードのオプション指定		int 型
		定数	内容	
	0	DPマスタモード	1	DPスレーブモード カメラスルー映像をダイレクトに表示。本モードを指定することでカラーカメラのスルー映像にオーバーレイすることが可能です。ただし、一度、本モードに設定すると解除できませんので注意して下さい。
	POWER_SAVE ( 0x100 )	サポートされていません。	——	

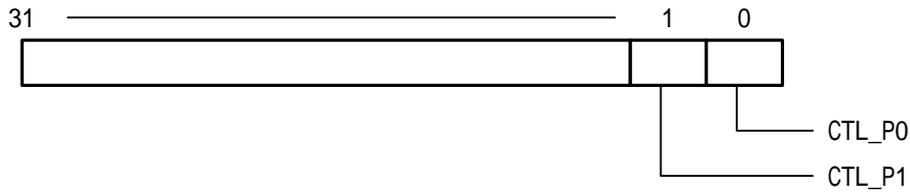
## ( 1 ) VIDEO\_LUT\_CTL ( 0 ) 濃度変換モード指定

ビデオ入力濃度変換機能の制御を行います。制御データは、VIDEO\_OPT\_LUTCTLテーブルにより設定します。以下に、設定の詳細を示します。

```
typedef struct {
    int    port; /* ビデオ入力部濃度変換制御 */
    int    bank; /* 濃度変換テーブルのバンク指定 */
}VIDEO_OPT_LUTCTL;
```

## ビデオ入力部濃度変換制御指定

「port」のビット構成



「port」の設定値

カメラ種別	CTL_P0	CTL_P1
ノーマルモノクロ入力	0 : 下位 8 ビットを入力 1 : 濃度変換実行	無効
YUVカラー入力	0 : UV画素の下位 8 ビットを入力 1 : UV画素の濃度変換実行	0 : Y画素の下位 8 ビットを入力 1 : Y画素の濃度変換実行
2カメラ入力	0 : カメラ 1 の下位 8 ビットを入力 1 : カメラ 1 の濃度変換実行	0 : カメラ 2 の下位 8 ビットを入力 1 : カメラ 2 の濃度変換実行
ODD/EVENカメラ入力	0 : ODDラインの下位 8 ビットを入力 1 : ODDラインの濃度変換実行	0 : EVENラインの下位 8 ビットを入力 1 : EVENラインの濃度変換実行
高精細カメラ入力	0 : 偶数画素の下位 8 ビットを入力 1 : 偶数画素の濃度変換実行	0 : 奇数画素の下位 8 ビットを入力 1 : 奇数画素の濃度変換実行

## 濃度変換を行う濃度変換テーブルのバンク指定

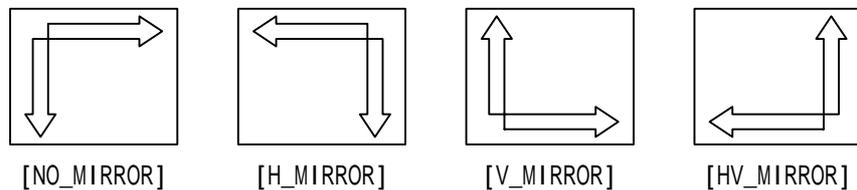
「bank」の設定値

LUTバンク	定数	内容
VIDEO_LUT_BANK0	0	LUTバンク # 0
VIDEO_LUT_BANK1	1	LUTバンク # 1

なお、ビデオ入力濃度変換メモリへのデータ書き込みは、WriteVideoLUT( )コマンドを使用して下さい。

## (2) MIRROR\_CTL ( 1 ) ミラー映像入力指定

ミラー映像入力機能の制御を行います。以下に、設定の詳細を示します。



## GetVideoOpt

## ビデオオプションの参照

## 機能

ビデオオプションの参照を行います。オプションについては、SetVideoOpt() コマンドを参照して下さい。

## コーディング

```
int
GetVideoOpt(
    DEVID    devID ,
    int      mode ,
    int      opt ,
    void     *optval ,
    int      optlen
)
```

## コメント

デバイスID  
設定モード  
オプション  
参照するオプションデータ領域のポインタ  
オプションデータ領域のサイズ

## パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**mode**

設定モード。

設定モード	定数	内容
VIDEO_OPT_VP0	0	ビデオポート# 0のオプション指定
VIDEO_OPT_VP1	1	ビデオポート# 1のオプション指定
VIDEO_OPT_DP	2	表示モードのオプション指定

**opt**

オプション。mode により、設定値が異なります。詳細情報を参照して下さい。

**\*optval**

参照するオプションデータ領域のポインタ。mode,opt により、値が異なります。SetVideoOpt() コマンドの詳細情報を参照して下さい。

**optlen**

オプションデータ領域のサイズ(バイトサイズ)。

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20, 21	パラメータ設定範囲外
-	エラーリセットコマンド(ClearIPError)未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

# ExtractPolyline

## 線分列抽出

### 機能

指定画面の指定した 2 値画像オブジェクトの線分列（外周座標）の抽出を行います。

### コーディング

```
int
ExtractPolyline(
    DEVID      devid ,
    int        imgID ,
    int        sx ,
    int        sy ,
    int        col ,
    int        opt ,
    POLY_TBL  *poly ,
    int        maxnum
)
```

### コメント

デバイス I D  
画面番号  
探索開始 X 座標 (画面絶対座標)  
探索開始 Y 座標 (画面絶対座標)  
探索対象物体の濃度値  
探索オプション  
線分列抽出座標格納テーブル  
抽出線分列最大数

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I D を指定して下さい。

**imgID**  
画面番号

**sx**  
探索開始 X 座標 (画面絶対座標)

**sy**  
探索開始 Y 座標 (画面絶対座標)

**col**  
探索対象物体の濃度値

**opt**  
探索オプション

探索オプション	内容
0	探索した全ての座標を線分列抽出座標格納テーブルに出力します。
1	探索した座標のうち頂点のみを線分列抽出座標格納テーブルに出力します。

**\*poly**  
線分列抽出座標格納テーブル。座標値は画面絶対座標です。

**maxnum**  
抽出線分列最大数

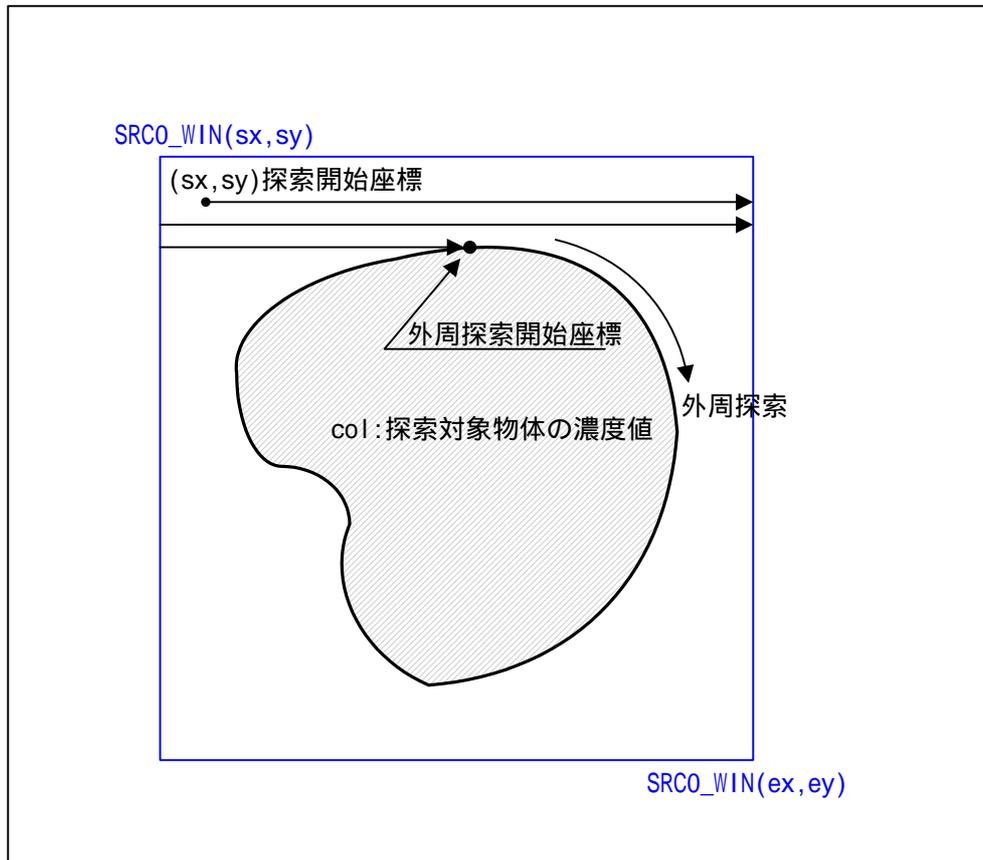
### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
> 0	抽出線分列数
- 1	異常終了

## 詳細情報

画面全体



ExtractPolyline() コマンドは、指定画面の探索開始座標 (sx, sy) からラスタースキャンで探索対象物体濃度値 (col) で指定された探索対象物体の外周探索開始座標を取得します。取得した外周探索開始座標から時計回りに探索対象物体の外周を探索し、再び外周探索開始座標に到達するか、抽出線分列最大数 (maxnum) になるまで探索し、線分列抽出座標格納テーブルに画面絶対座標を出力します。

ExtractPolyline() コマンドの探索開始座標 (sx, sy) が探索対象物体に近いほど処理時間は短くなります。また、探索対象物体が SRCO\_WIN で指定した座標に接しているか、それをまたいでいる場合、処理はエラーで終了します。

POLY\_TBL のフォーマット及び内容を、下記に示します。

vpxsys.h で宣言しています。

```
typedef struct {
    int      num;    // 有効データ数
    WPOINT*  pos;    // X , Y 座標配列
} POLY_TBL;
```

```
typedef struct {
    short    x;
    short    y;
} WPOINT;
```

# PolyArea

## 線分列からの面積抽出

### 機能

線分列から面積値の抽出を行います。

### コーディング

```
int
PolyArea(
    DEVID      devID ,
    POLY_TBL   *poly ,
    double     *area
)
```

### コメント

デバイス I D  
線分列抽出座標  
面積(画素)

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**\*poly**  
線分列抽出座標

**\*area**  
面積(画素)

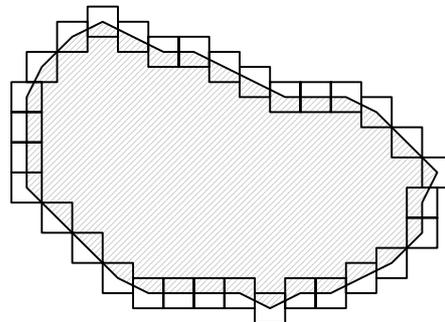
### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

面積は、外周座標値の中間点を結ぶ閉曲面内の面積を計算します。なお、穴があった場合でも穴の面積を除外しません。



POLY\_TBL のフォーマット及び内容を、下記に示します。

vpxsys.h で宣言しています。

```
typedef struct {
    int      num;      // 有効データ数
    WPOINT*  pos;      // X , Y 座標配列
} POLY_TBL;
```

```
typedef struct {
    short    x;
    short    y;
}WPOINT;
```

# PolyPerim

## 線分列からの周囲長抽出

### 機能

線分列から周囲長の抽出を行います。

### コーディング

```
int
PolyPerim(
    DEVID      devID ,
    POLY_TBL   *poly ,
    double     *perim
)
```

### コメント

デバイス I D  
線分列抽出座標  
周囲長(画素)

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**\*poly**  
線分列抽出座標

**\*perim**  
周囲長(画素)

### リターン値

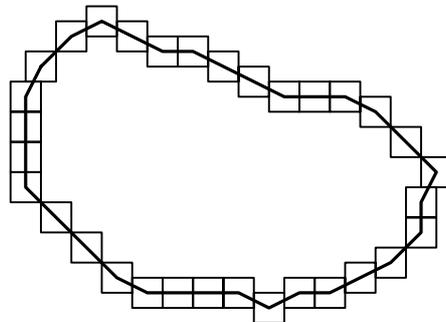
リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

POLY\_TBL のフォーマット及び内容は、PolyArea()を参照して下さい。

周囲長は、外周座標値の中間点を結ぶ線の長さを計算します。



# PolyGrav

## 線分列からの重心座標抽出

### 機能

線分列から重心座標の抽出を行います。

### コーディング

```
int
PolyGrav(
    DEVID          devID ,
    POLY_TBL      *poly ,
    float         *gx ,
    float         *gy
)
```

### コメント

デバイス I D  
線分列抽出座標  
重心 X 座標  
重心 Y 座標

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**\*poly**  
線分列抽出座標

**\*gx**  
重心 X 座標

**\*gy**  
重心 Y 座標

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

POLY\_TBL のフォーマット及び内容は、PolyArea()を参照して下さい。

# PolyFeatures

## 線分列からの形状特徴量抽出

### 機能

線分列から形状特徴量の抽出を行います。

### コーディング

```
int
PolyFeatures(
    DEVID          devID ,
    POLY_TBL       *poly ,
    POLY_FEATURE   *tbl  ,
    int            opt
)
```

### コメント

デバイスID  
線分列抽出座標  
ヒストグラム特徴量テーブル  
オプション

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**\*poly**  
線分列抽出座標

**\*tbl**  
ヒストグラム特徴量テーブル

**opt**  
抽出オプション。抽出する特徴量を指定して下さい。

抽出オプション	定数	内容
POLY_CENTER_of_GRAVITY	0	外接座標 + 周囲長 + 面積 + 重心
POLY_MOMENT	1	上記 + モーメント + 慣性主軸角度
POLY_AXIS LENG	2	上記 + 慣性主軸長 + パターン幅
POLY_MAX LENG	3	上記 + 絶対最大長 + パターン幅

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

**詳細情報**

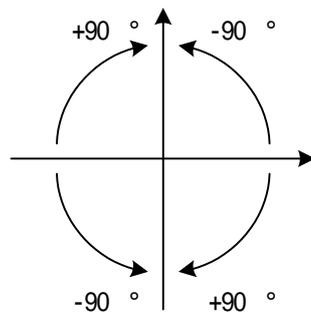
POLY\_TBL のフォーマット及び内容は、PolyArea()を参照して下さい。  
POLY\_FEATURE のフォーマット及び内容を、下記に示します。

vpxsys.h で宣言しています。

```
typedef struct {
    int     LEFT_UP_POINT_X;
    int     LEFT_UP_POINT_Y;
    int     REGION_X0;
    int     REGION_Y0;
    int     REGION_X1;
    int     REGION_Y1;
    double  PERIMETER;
    double  AREA;
    double  CENTER_of_GRAVITY_X;
    double  CENTER_of_GRAVITY_Y;
    double  ANGLE;
    double  MOMENT_SIGMA_X;
    double  MOMENT_SIGMA_Y;
    double  MOMENT_SIGMA_XX;
    double  MOMENT_SIGMA_XY;
    double  MOMENT_SIGMA_YY;
    float   AXIS LENG;
    float   AXIS_WIDTH;
    int     AXIS LENG_X0;
    int     AXIS LENG_Y0;
    int     AXIS LENG_X1;
    int     AXIS LENG_Y1;
    float   MAX LENG;
    float   WIDTH;
    float   ORIENTATION;
    int     MAX LENG_X0;
    int     MAX LENG_Y0;
    int     MAX LENG_X1;
    int     MAX LENG_Y1;
} POLY_FEATURE;
```

ヒストグラム特徴テーブル	内 容
LEFT_UP_POINT_X	2 値物体（白）が最初に出現する点の X 座標
LEFT_UP_POINT_Y	2 値物体（白）が最初に出現する点の Y 座標
REGION_X0	2 値物体（白）が存在する最小 X 座標
REGION_Y0	2 値物体（白）が存在する最小 Y 座標
REGION_X1	2 値物体（白）が存在する最大 X 座標
REGION_Y1	2 値物体（白）が存在する最大 Y 座標
PERIMETER	周囲長
AREA	2 値物体（白）の面積
CENTER_of_GRAVITY_X	重心 X 座標
CENTER_of_GRAVITY_Y	重心 Y 座標
ANGLE	慣性主軸と X 軸とのなす角度（degree）（注）
MOMENT_SIGMA_X	X 軸への 1 次モーメント
MOMENT_SIGMA_Y	Y 軸への 1 次モーメント
MOMENT_SIGMA_XX	X 軸への 2 次モーメント
MOMENT_SIGMA_XY	共役モーメント
MOMENT_SIGMA_YY	Y 軸への 2 次モーメント
AXIS LENG	慣性主軸の軸長
AXIS WIDTH	慣性主軸の軸長を挟む 2 本の直線間の距離
AXIS LENG_X0	慣性主軸の軸長を示す対角線 X 座標
AXIS LENG_Y0	慣性主軸の軸長を示す対角線 Y 座標
AXIS LENG_X1	慣性主軸の軸長を示す対角線 X 座標
AXIS LENG_Y1	慣性主軸の軸長を示す対角線 Y 座標
MAX LENG	絶対最大長
WIDTH	絶対最大長を挟む 2 本の直線間の距離
ORIENTATION	絶対最大長の方向と水平線との角度（degree）（注）
MAX LENG_X0	絶対最大長を示す対角線 X 座標
MAX LENG_Y0	絶対最大長を示す対角線 Y 座標
MAX LENG_X1	絶対最大長を示す対角線 X 座標
MAX LENG_Y1	絶対最大長を示す対角線 Y 座標

(注)



# IP\_FillHole

## 2値画像の穴埋め

### 機能

2値画像の穴埋めを行います。2値のソース画像に対し、パラメータで指定された設定値でラベリング処理を行い、2値画像の穴埋めをします。なお、ラベリング処理を用いているため、255個以上の穴埋めはできません。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_FillHole(
    DEVID          devID ,
    int            ImgSrc,
    int            ImgDst,
    int            opt ,
    int            col ,
    OBJ_AREA_OPT  *hole_opt
)
```

### コメント

デバイスID  
ソース画面番号  
デスティネーション画面番号  
オプション  
穴埋めする穴の色  
穴埋めする物体の抽出面積閾値

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面番号

**ImgDst**  
デスティネーション画面番号

**opt**  
オプション

オプション	内容
0	4連結ラベリングによる処理
1	8連結ラベリングによる処理

**col**  
穴埋めする穴の色

色	内容
0	黒の穴を埋める
1	白の穴を埋める

**hole\_opt**  
穴埋めする穴の面積閾値。面積判定しない場合はNULLを指定してください。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
51	ワーク画面確保エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

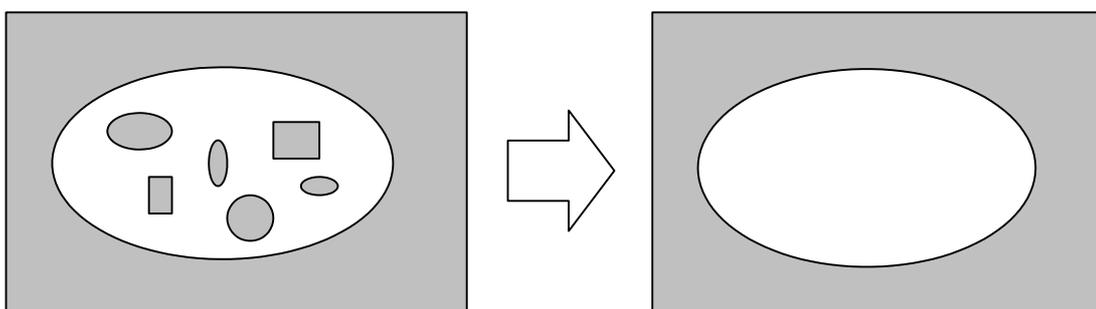
OBJ\_AREA\_OPTのフォーマットを下記に示します。

vpxsys.hで宣言しています。

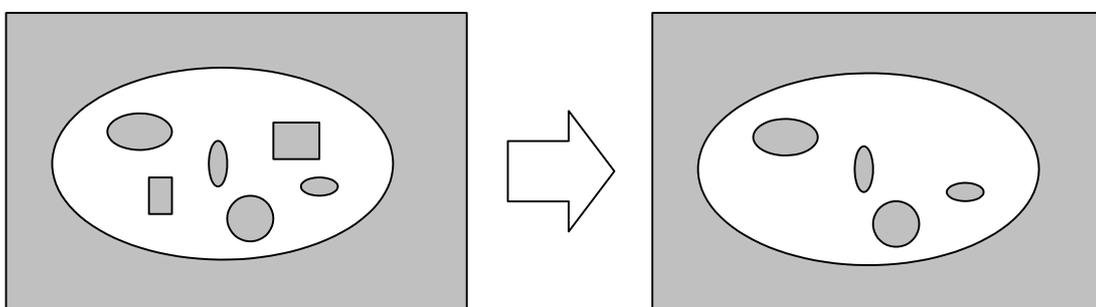
```
typedef struct {  
    int    min;    // 最小面積  
    int    max;    // 最大面積  
} OBJ_AREA_OPT;
```

IP\_FillHole()コマンドは、下図に示すように画面のSRCO\_WINの指定領域中に対象物が1個の場合を想定しています。指定領域内に物体が複数存在する場合は、IP\_FillHoleExt()コマンドを使用してください。

hole\_opt= NULL の場合



hole\_opt 指定の場合



また、画像サイズ2048×2048で面積判定する場合、ウィンドウサイズの横幅を2047に設定してください。2048でコマンドを実行すると、正常に処理できません。

# IP\_FillHoleExt

## 2値画像の穴埋め(拡張)

### 機能

2値画像の対象物体の抽出を行い抽出した対象物の穴埋めを行います。2値のソース画像に対し、指定されたパラメータによりラベリング処理を行い、対象物体の抽出と2値画像の穴埋めを行います。なお、ラベリング処理を用いているため、255個以上の穴埋めはできません。

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_FillHoleExt(
    DEVID
    int
    int
    int
    int
    OBJ_AREA_OPT
    OBJ_AREA_OPT
)
```

```
devID ,
ImgSrc,
ImgDst,
opt ,
col ,
*hole_opt ,
*obj_opt
```

### コメント

```
デバイスID
ソース画面番号
デスティネーション画面番号
オプション
穴埋めする穴の色
穴埋めする穴の面積閾値
穴埋めする対象物体の抽出面積閾値
```

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面番号。

**ImgDst**  
デスティネーション画面番号。

**opt**  
オプション

オプション	内容
0	4 連結ラベリングによる処理
1	8 連結ラベリングによる処理

**col**  
穴埋めする穴の色

色	内容
0	黒の穴を埋める
1	白の穴を埋める

**hole\_opt**  
穴埋めする穴の面積閾値。面積判定しない場合はNULLを指定してください。

**obj\_opt**  
穴埋めする対象物体の抽出面積閾値。面積判定しない場合はNULLを指定してください。NULLを指定すると、指定領域内で最大面積の物体を抽出し、穴埋めを行います。

OBJ\_AREA\_OPTのフォーマットは、IP\_FillHole()コマンドを参照してください。

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

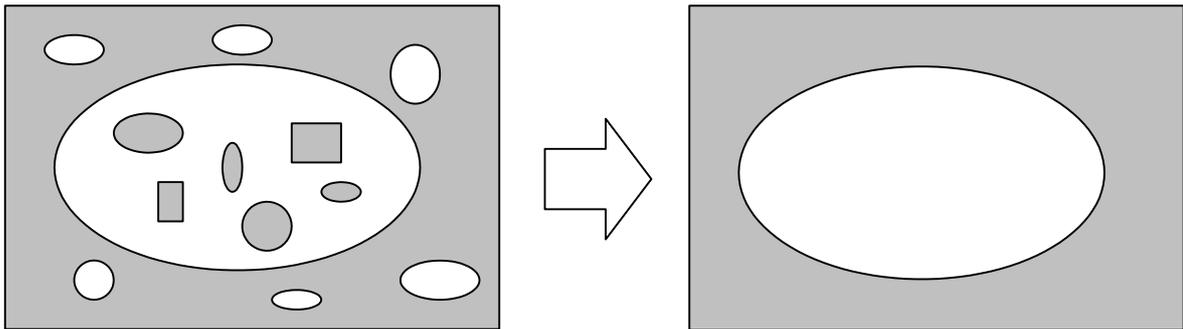
エラー番号	エラー原因
5 1	ワーク画面確保エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

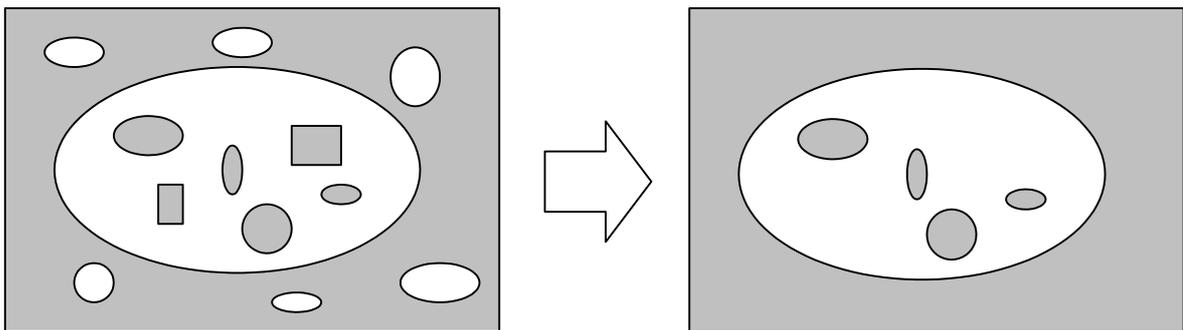
## 詳細情報

IP\_FillHoleExt() コマンドは、画面のSRCO\_WINの指定領域内に物体が複数存在する場合でも、まず、穴埋めする対象物体を抽出し、その抽出した物体に対して穴埋め処理を実行します。

hole\_opt= NULL、obj\_opt 指定の場合



hole\_opt、obj\_opt 指定の場合



また、画像サイズ 2048 × 2048 で面積判定する場合、ウィンドウサイズの横幅を 2047 に設定してください。2048 でコマンドを実行すると、正常に処理できません。

# CreateR G B L U T

## R G B L U Tオブジェクトの生成

### 機能

R G B L U Tオブジェクトの生成を行います。IP\_ConvertR G B L U T()およびIP\_ConvertR G B L U TEx()コマンドは、modeで指定されたL U Tモードに従いR G BのL U T変換を実行します。

S V P - 3 3 0では、ハードウェアの制約により「R G B L U T変換コマンド」は、サポートされていません。

### コーディング

```
HRGBLUT
CreateR G B L U T(
    DEVID
    int
    mode
    int
    opt
)
```

### コメント

デバイスID  
L U Tモード  
オプション

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

mode

L U Tモード

L U Tモード	定数値	内 容
R G B L U T888	0	R:8bit / G:8bit / B:8bit (16,777,216byte)
R G B L U T777	1	R:7bit / G:7bit / B:7bit (2,097,152byte)
R G B L U T666	2	R:6bit / G:6bit / B:6bit (262,144byte)
R G B L U T887	3	R:8bit / G:8bit / B:7bit (8,388,608byte)
R G B L U T878	4	R:8bit / G:7bit / B:8bit (8,388,608byte)
R G B L U T788	5	R:7bit / G:8bit / B:8bit (8,388,608byte)
R G B L U T787	6	R:7bit / G:8bit / B:7bit (4,194,304byte)
R G B L U T2CH88	7	R:8bit / G:8bit (65,536byte)

opt

オプション。現在未使用。「0」を指定してください。

### リターン値

リターン値は HRGBLUT 型で返されます。詳細を下表に示します。

リターン値	内 容
0 以外	R G B L U Tハンドル
0	異常終了

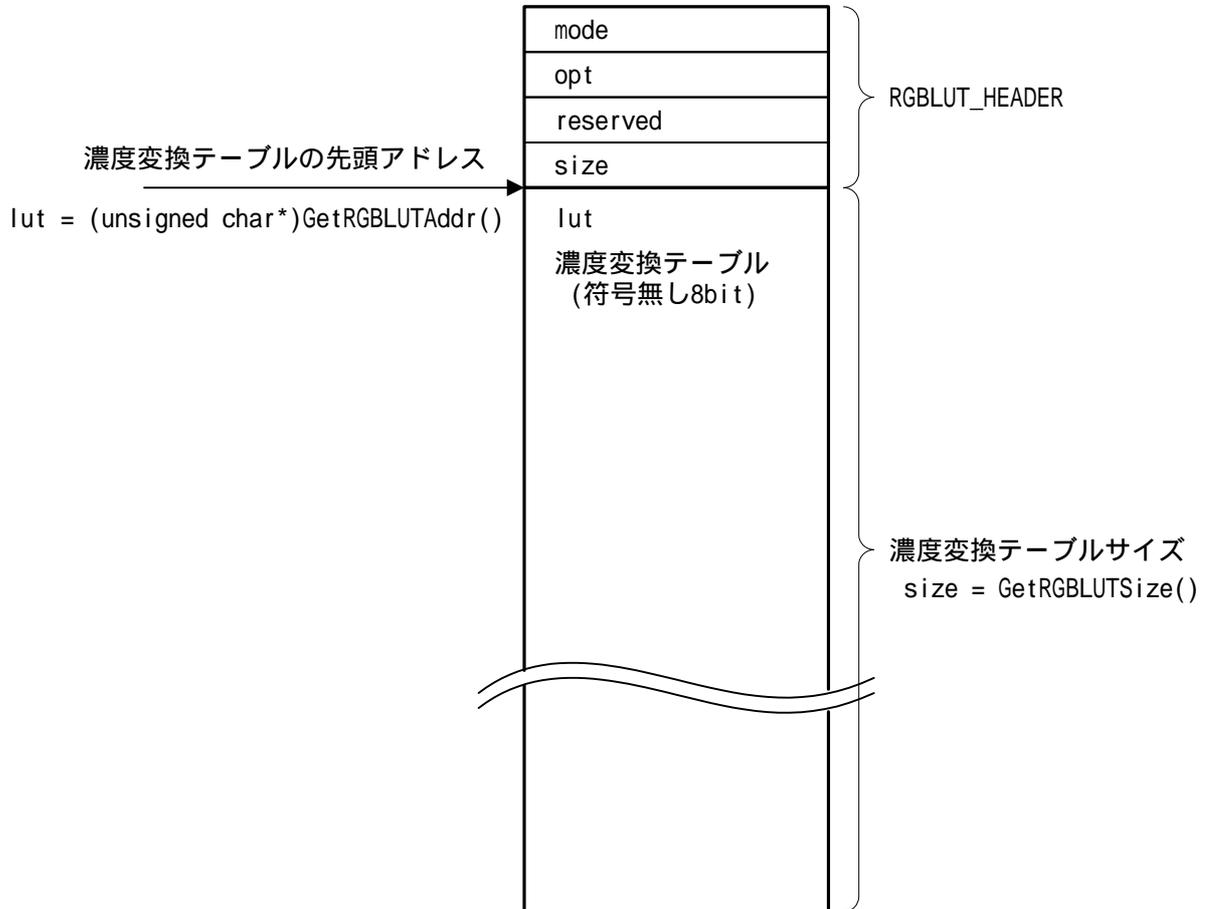
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
2 0 , 2 1	L U Tモード、オプションの設定範囲外
4 6	メモリヒープ領域空きなし
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

本コマンドで生成される R G B オブジェクトを以下に示します。R G B オブジェクトはオンボード C P U 上に生成され、R G B L U T ハンドルで管理されます。オンボード C P U 上からは、濃度変換テーブルの先頭アドレスを取得し直接濃度変換テーブルをアクセスできます。濃度変換テーブルの先頭アドレスおよびサイズは、それぞれ `GetRGLUTAddr()`、`GetRGLUTSize()` コマンドで取得できます。



## DeleteRGLUT

## RGLUTオブジェクトの削除

### 機能

R G B L U Tオブジェクトの削除を行います。

S V P - 3 3 0では、ハードウェアの制約により「R G B L U T変換コマンド」は、サポートされていません。

### コーディング

```
int  
DeleteRGLUT(  
    DEVID  
    HRGLUT  
)
```

### コメント

devID ,  
hLUT

デバイスID  
LUTハンドル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

hLUT

LUTハンドル

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

# SetRGLUT

## RGLUTテーブルに値を設定

### 機能

R G B L U Tテーブルに値を設定します。

### コーディング

```
int
SetRGLUT(
    DEVID          devID ,
    HRGLUT         hLUT ,
    int            r ,
    int            g ,
    int            b ,
    int            v
)
```

### コメント

デバイス I D  
L U Tハンドル  
R成分  
G成分  
B成分  
設定値

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**hLUT**  
L U Tハンドル

**r**  
R成分

**g**  
G成分

**b**  
B成分

**v**  
設定値

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

以下のマクロが「VPXDEF.H」に定義されています。オンボード C P U側では、LUTに直接アクセスすることにより、SetRGLUT()コマンドよりもLUTテーブルに高速にアクセスできます。

```
#define SetRGLUT888(lut,r,g,b,d) lut[r << 16 | g << 8 | b]= (d)
#define SetRGLUT777(lut,r,g,b,d) lut[(r & 0xfe) << 13 | (g & 0xfe) << 6 | b >> 1]= (d)
#define SetRGLUT666(lut,r,g,b,d) lut[(r & 0xfc) << 10 | (g & 0xfc) << 4 | b >> 2]= (d)
#define SetRGLUT887(lut,r,g,b,d) lut[r << 15 | g << 7 | b >> 1]= (d)
#define SetRGLUT878(lut,r,g,b,d) lut[r << 15 | (g & 0xfe) << 7 | b]= (d)
#define SetRGLUT788(lut,r,g,b,d) lut[(r & 0xfe) << 15 | g << 8 | b]= (d)
#define SetRGLUT787(lut,r,g,b,d) lut[(r & 0xfe) << 14 | g << 7 | b >> 1]= (d)
#define SetRGLUT2CH88(lut,r,g,d) lut[r << 8 | g]= (d)
```

「lut」はオンボード C P U側で、以下のように取得してください。

```
unsigned char* lut;
lut = (unsigned char*)GetRGLUTAddr(devID,hLUT);
```

# GetRGLUT

## RGLUTテーブルから値を取得

### 機能

RGLUTテーブルから値を取得します。

### コーディング

```
int
GetRGLUT(
    DEVID          devID ,
    HRGLUT        hLUT ,
    int            r ,
    int            g ,
    int            b ,
    int            *v
)
```

### コメント

デバイスID  
LUTハンドル  
R成分  
G成分  
B成分  
設定値

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**hLUT**  
LUTハンドル

**r**  
R成分

**g**  
G成分

**b**  
B成分

**\*v**  
設定値

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

以下のマクロが「VPXDEF.H」に定義されています。オンボードCPU側では、LUTに直接アクセスすることにより、GetRGLUT()コマンドよりもLUTテーブルに高速にアクセスできます。

```
#define GetRGLUT888(lut,r,g,b) lut[r << 16 | g << 8 | b]
#define GetRGLUT777(lut,r,g,b) lut[(r & 0xfe) << 13 | (g & 0xfe) << 6 | b >> 1]
#define GetRGLUT666(lut,r,g,b) lut[(r & 0xfc) << 10 | (g & 0xfc) << 4 | b >> 2]
#define GetRGLUT887(lut,r,g,b) lut[r << 15 | g << 7 | b >> 1]
#define GetRGLUT878(lut,r,g,b) lut[r << 15 | (g & 0xfe) << 7 | b]
#define GetRGLUT788(lut,r,g,b) lut[(r & 0xfe) << 15 | g << 8 | b]
#define GetRGLUT787(lut,r,g,b) lut[(r & 0xfe) << 14 | g << 7 | b >> 1]
#define GetRGLUT2CH88(lut,r,g) lut[r << 8 | g]
```

「lut」はオンボードCPU側で、以下のように取得してください。

```
unsigned char* lut;
lut = (unsigned char*)GetRGLUTAddr(devID,hLUT);
```

## GetRGLUTAddr

## RGLUTのLUTテーブルの先頭アドレス

### 機能

RGLUTのLUTテーブルの先頭アドレスを取得します。本コマンドはオンボードCPU側のメモリアドレスを返します。オンボードCPU側のアプリケーションからのみ使用できます。

### コーディング

```
unsigned long
GetRGLUTAddr(
    DEVID
    HRGLUT
)
```

devID ,  
hLUT

### コメント

デバイスID  
LUTハンドル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

hLUT

LUTハンドル

### リターン値

リターン値は unsigned long 型で返されます。詳細を下表に示します。

リターン値	内容
0 以外	LUTの先頭アドレス
0	異常終了

### 詳細情報

以下のマクロが「VPXDEF.H」に定義されています。オンボードCPU側では、LUTに直接アクセスすることにより、SetRGLUT(),GetRGLUT()コマンドよりもLUTテーブルに高速にアクセスできます。

```
#define SetRGLUT888(lut,r,g,b,d) lut[r << 16 | g << 8 | b]= (d)
#define SetRGLUT777(lut,r,g,b,d) lut[(r & 0xfe) << 13 | (g & 0xfe) << 6 | b >> 1]= (d)
#define SetRGLUT666(lut,r,g,b,d) lut[(r & 0xfc) << 10 | (g & 0xfc) << 4 | b >> 2]= (d)
#define SetRGLUT887(lut,r,g,b,d) lut[r << 15 | g << 7 | b >> 1]= (d)
#define SetRGLUT878(lut,r,g,b,d) lut[r << 15 | (g & 0xfe) << 7 | b]= (d)
#define SetRGLUT788(lut,r,g,b,d) lut[(r & 0xfe) << 15 | g << 8 | b]= (d)
#define SetRGLUT787(lut,r,g,b,d) lut[(r & 0xfe) << 14 | g << 7 | b >> 1]= (d)
#define SetRGLUT2CH88(lut,r,g,d) lut[r << 8 | g]= (d)

#define GetRGLUT888(lut,r,g,b) lut[r << 16 | g << 8 | b]
#define GetRGLUT777(lut,r,g,b) lut[(r & 0xfe) << 13 | (g & 0xfe) << 6 | b >> 1]
#define GetRGLUT666(lut,r,g,b) lut[(r & 0xfc) << 10 | (g & 0xfc) << 4 | b >> 2]
#define GetRGLUT887(lut,r,g,b) lut[r << 15 | g << 7 | b >> 1]
#define GetRGLUT878(lut,r,g,b) lut[r << 15 | (g & 0xfe) << 7 | b]
#define GetRGLUT788(lut,r,g,b) lut[(r & 0xfe) << 15 | g << 8 | b]
#define GetRGLUT787(lut,r,g,b) lut[(r & 0xfe) << 14 | g << 7 | b >> 1]
#define GetRGLUT2CH88(lut,r,g) lut[r << 8 | g]
```

「lut」はオンボードCPU側で、以下のように取得してください。

```
unsigned char* lut;
lut = (unsigned char*)GetRGLUTAddr(devID,hLUT);
```

## GetRGLUTSize

## R G B L U TのL U Tテーブルのサイズ

### 機能

R G B L U Tのテーブルのサイズを取得します。

### コーディング

```
int  
GetRGLUTSize(  
    DEVID  
    HRGLUT  
)
```

### コメント

devID ,  
hLUT

デバイスID  
LUTハンドル

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

hLUT

LUTハンドル

### リターン値

リターン値は int 型でLUTのサイズが返されます。

### 詳細情報

## WriteRGLUT

### 濃度変換テーブルからRGLUTオブジェクトへのコピー

#### 機能

ローカルの濃度変換テーブルバッファ(rgblut)からR G B L U TオブジェクトのL U Tテーブルへのコピーを行います。

#### コーディング

```
int
WriteRGLUT(
    DEVID
    HRGLUT
    void
    int
)
```

```
devID ,
hLUT ,
*rgblut ,
size
```

#### コメント

```
デバイスID
LUTハンドル
濃度変換テーブルの先頭アドレス
濃度変換テーブルのサイズ
```

#### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**hLUT**

LUTハンドル

**\*rgblut**

ローカルの濃度変換テーブルの先頭アドレス

**size**

ローカルの濃度変換テーブルのサイズ

#### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

#### 詳細情報

## ReadRGLUT

## RGLUTオブジェクトから濃度変換テーブルへのコピー

### 機能

RGLUTオブジェクトのLUTテーブルからローカルの濃度変換テーブルバッファ(rglut)へのコピーを行います。

### コーディング

```
int
ReadRGLUT(
    DEVID
    HRGLUT
    void
    int
)
```

```
devID ,
hLUT ,
*rglut ,
size
```

### コメント

デバイスID  
LUTハンドル  
濃度変換テーブルの先頭アドレス  
濃度変換テーブルのサイズ

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**hLUT**

LUTハンドル

**\*rglut**

ローカルの濃度変換テーブルの先頭アドレス

**size**

ローカルの濃度変換テーブルのサイズ

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

### 詳細情報

## IP\_ConvertRGLUT

## RGLUT変換の実行

## 機能

R G B 画像の L U T 変換を行います。指定された L U T オブジェクトの L U T モードに従い R G B の L U T 変換を実行します（詳細情報を参照してください）。

```
ImgDst = LUT[ ImgSrc(R) << 16 | ImgSrc(G) << 8 | ImgSrc(B) ] (RGLUT888の場合)
```

S V P - 3 3 0 では、ハードウェアの制約により「R G B L U T 変換コマンド」は、サポートされていません。

## コーディング

```
int
IP_ConvertRGLUT(
    DEVID
    int
    int
    int
    HRGLUT
)
```

```
devID ,
ImgSrc ,
ImgDst ,
opt ,
hLUT
```

## コメント

```
デバイスID
R G B ソース画面番号
デスティネーション画面番号
オプション
L U T ハンドル
```

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgSrc

R G B ソース画面番号

ImgDst

デスティネーション画面番号

opt

オプション ("0"を設定してください)

hLUT

L U T ハンドル

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0	R G B ソース画面の不当画面番号エラー
1 3	デスティネーション画面の不当画面番号エラー
2 2	オプションの設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

以下にLUTモードに対する演算内容を示します。

LUTモード	演算内容
RGLUT888	$\text{ImgDst} = \text{LUT}[ \text{ImgSrc}(R) \ll 16 \mid \text{ImgSrc}(G) \ll 8 \mid \text{ImgSrc}(B) ]$
RGLUT777	$\text{ImgDst} = \text{LUT}[ (\text{ImgSrc}(R) \& 0xFE) \ll 13 \mid (\text{ImgSrc}(G) \& 0xFE) \ll 6 \mid \text{ImgSrc}(B) \gg 1 ]$
RGLUT666	$\text{ImgDst} = \text{LUT}[ (\text{ImgSrc}(R) \& 0xFC) \ll 10 \mid (\text{ImgSrc}(G) \& 0xFC) \ll 4 \mid \text{ImgSrc}(B) \gg 2 ]$
RGLUT887	$\text{ImgDst} = \text{LUT}[ \text{ImgSrc}(R) \ll 15 \mid \text{ImgSrc}(G) \ll 7 \mid \text{ImgSrc}(B) \gg 1 ]$
RGLUT878	$\text{ImgDst} = \text{LUT}[ \text{ImgSrc}(R) \ll 15 \mid (\text{ImgSrc}(G) \& 0xFE) \ll 7 \mid \text{ImgSrc}(B) ]$
RGLUT788	$\text{ImgDst} = \text{LUT}[ (\text{ImgSrc}(R) \& 0xFE) \ll 15 \mid \text{ImgSrc}(G) \ll 8 \mid \text{ImgSrc}(B) ]$
RGLUT787	$\text{ImgDst} = \text{LUT}[ (\text{ImgSrc}(R) \& 0xFE) \ll 14 \mid \text{ImgSrc}(G) \ll 7 \mid \text{ImgSrc}(B) \gg 1 ]$
RGLUT2CH88	$\text{ImgDst} = \text{LUT}[ \text{ImgSrc}(R) \ll 8 \mid \text{ImgSrc}(G) ]$

## IP\_ConvertRGLUTEx

## RGLUT変換の実行(RGB分離)

## 機能

分離された R, G, B 画像の LUT 変換を行います。指定された LUT オブジェクトの LUT モードに従い RGB の LUT 変換を実行します (詳細情報を参照してください)。

```
ImgDst = LUT[ ImgR << 16 | ImgG << 8 | ImgB ] (RGLUT888の場合)
```

SVP-330では、ハードウェアの制約により「RGLUT変換コマンド」は、サポートされていません。

## コーディング

```
int
IP_ConvertRGLUTEx(
    DEVID
    int
    int
    int
    int
    int
    int
    HRGLUT
)
```

```
devID ,
ImgR ,
ImgG ,
ImgB ,
ImgDst ,
opt ,
hLUT
```

## コメント

```
デバイスID
Rソース画面番号
Gソース画面番号
Bソース画面番号
デスティネーション画面番号
オプション
LUTハンドル
```

## パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

ImgR

Rソース画面番号

ImgG

Gソース画面番号

ImgB

Bソース画面番号

ImgDst

デスティネーション画面番号

opt

オプション ("0"を設定してください)

hLUT

LUTハンドル

## リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
10	RGBソース画面の不当画面番号エラー
13	デスティネーション画面の不当画面番号エラー
22	オプションの設定範囲外
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

以下にLUTモードに対する演算内容を示します。

LUTモード	演算内容
RGBLUT888	$\text{ImgDst} = \text{LUT}[ \text{ImgR} \ll 16 \mid \text{ImgG} \ll 8 \mid \text{ImgB} ]$
RGBLUT777	$\text{ImgDst} = \text{LUT}[ (\text{ImgR} \& 0xFE) \ll 13 \mid (\text{ImgG} \& 0xFE) \ll 6 \mid \text{ImgB} \gg 1 ]$
RGBLUT666	$\text{ImgDst} = \text{LUT}[ (\text{ImgR} \& 0xFC) \ll 10 \mid (\text{ImgG} \& 0xFC) \ll 4 \mid \text{ImgB} \gg 2 ]$
RGBLUT887	$\text{ImgDst} = \text{LUT}[ \text{ImgR} \ll 15 \mid \text{ImgG} \ll 7 \mid \text{ImgB} \gg 1 ]$
RGBLUT878	$\text{ImgDst} = \text{LUT}[ \text{ImgR} \ll 15 \mid (\text{ImgG} \& 0xFE) \ll 7 \mid \text{ImgB} ]$
RGBLUT788	$\text{ImgDst} = \text{LUT}[ (\text{ImgR} \& 0xFE) \ll 15 \mid \text{ImgG} \ll 8 \mid \text{ImgB} ]$
RGBLUT787	$\text{ImgDst} = \text{LUT}[ (\text{ImgR} \& 0xFE) \ll 14 \mid \text{ImgG} \ll 7 \mid \text{ImgB} \gg 1 ]$
RGBLUT2CH88	$\text{ImgDst} = \text{LUT}[ \text{ImgR} \ll 8 \mid \text{ImgG} ]$

# IP\_PseudoColor

## 擬似カラー変換

### 機能

R, G, B変換テーブルに従ってモノクロ画像を擬似カラー画像に変換します。  
(ウィンドウを無視して画面サイズで変換します)

本コマンドはハードウェアの制約により、いくつかの制限事項があります。詳細は『画像処理の制限事項』の章を参照して下さい。

### コーディング

```
int
IP_PseudoColor(
    DEVID          devID ,
    int            ImgSrc,
    int            ImgYUV,
    unsigned char  Rtbl[255] ,
    unsigned char  Gtbl[255] ;
    unsigned char  Btbl[255] ;
)
```

### コメント

デバイスID  
ソース画面番号  
擬似カラー変換先画面番号  
R画像データ格納先アドレス  
G画像データ格納先アドレス  
B画像データ格納先アドレス

### パラメータ

**devID**  
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**ImgSrc**  
ソース画面番号

**ImgYUV**  
擬似カラー変換先画面番号

**Rtbl**  
R画像データ格納先アドレス

**Gtbl**  
G画像データ格納先アドレス

**Btbl**  
B画像データ格納先アドレス

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	抽出線分列数
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
1 0 , 1 3	指定画面空き領域、または範囲外 ( 不当画面番号エラー )
1 4	カラー画面番号に対してYUV画面以外の画面番号を指定した
-	エラーリセットコマンド ( ClearIPError ) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

# SetPseudoColor

## RGB擬似カラーテーブル作成

### 機能

RGB画像データによる  
 黒色 青色 水色 緑色 黄色 赤色 ピンク色 白色  
 の擬似カラーテーブルを作成します。

### コーディング

```
int
SetPseudoColor(
    DEVID
    unsigned char    Rtbl[255] ,
    unsigned char    Gtbl[255] ;
    unsigned char    Btbl[255] ;
)
```

### コメント

デバイスID  
 R画像データ格納先アドレス  
 G画像データ格納先アドレス  
 B画像データ格納先アドレス

### パラメータ

**devID**  
 デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**Rtbl**  
 R画像データ格納先アドレス(出力)

**Gtbl**  
 G画像データ格納先アドレス(出力)

**Btbl**  
 B画像データ格納先アドレス(出力)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	抽出線分列数
- 1	異常終了

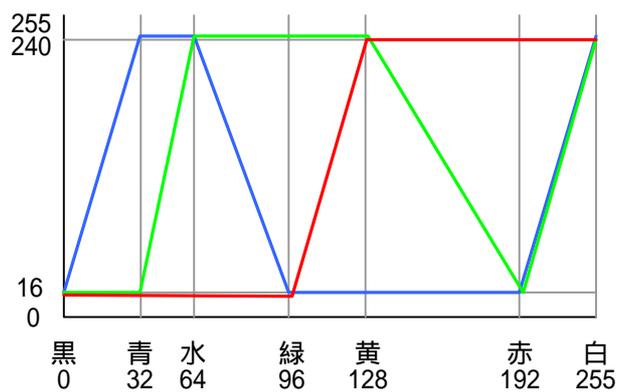
異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
5 1	ワーク画面確保エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

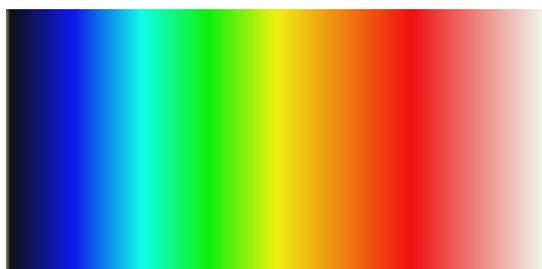
## 詳細情報

R G B 擬似カラー対応表を下記に示します。



注) R G B 画像から Y U V 画像に変換する際のオーバーフローを考慮して濃度値を 16 から 240 に設定しています。

— Rtbl  
— Gtbl  
— Btbl



# SetPseudoColorExt

## R G B 擬似カラーテーブル作成(拡張)

### 機能

R G B 画像での擬似カラー制御テーブル(cnt)による擬似カラーテーブルを作成します。

### コーディング

```
int
SetPseudoColorExt (
    DEVID
    unsigned char
    unsigned char
    unsigned char
    PseudoColorTBL
)
```

```
devID ,
Rtbl [255] ,
Gtbl [255] ,
Btbl [255] ,
cnt
```

### コメント

デバイス I D  
R 画像データ格納先アドレス  
G 画像データ格納先アドレス  
B 画像データ格納先アドレス  
擬似カラー制御テーブル

### パラメータ

**devID**  
デバイス I D。OpenIPDevコマンドで得られた対応するボード番号のデバイス I Dを指定して下さい。

**Rtbl**  
R 画像データ格納先アドレス(出力)

**Gtbl**  
G 画像データ格納先アドレス(出力)

**Btbl**  
B 画像データ格納先アドレス(出力)

**cnt**  
擬似カラー制御テーブル(入力)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	抽出線分列数
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
5 1	ワーク画面確保エラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

## 詳細情報

PseudoColorTBL のフォーマットを下記に示します。

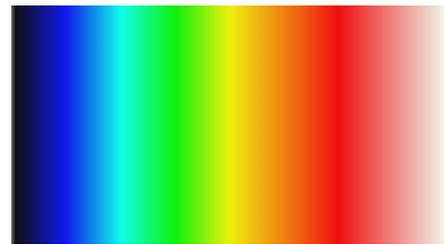
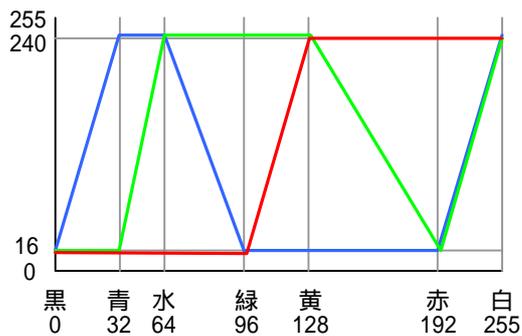
ipxsys.h で宣言しています。

```
typedef struct{
    int    blue
    int    lightblue
    int    green
    int    yellow
    int    red
}PseudoColorTBL;
```

実行例を下記に示します。

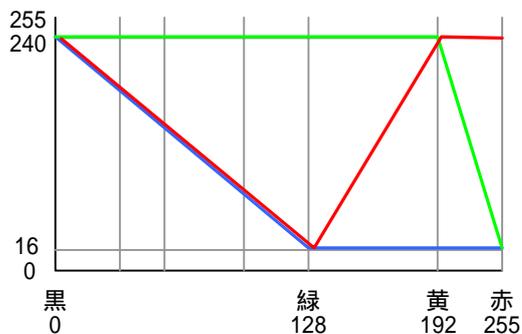
[例 1]

```
cnt.blue      = 32;
cnt.lightblue = 64;
cnt.green     = 96;
cnt.yellow    = 128;
cnt.red       = 192;
SetPseudoColorExt( Rtbl, Gtbl, Btbl, cnt );
```



[例 2]

```
cnt.blue      = 0;
cnt.lightblue = 0;
cnt.green     = 128;
cnt.yellow    = 192;
cnt.red       = 255;
SetPseudoColorExt( Rtbl, Gtbl, Btbl, cnt );
```



# StartWDT

## WDTスタート

### 機能

WDT (Watch Dog Timer) を起動します。

### コーディング

```
int
StartWDT(
  DEVID
  int
  int
)
```

### コメント

```
devID ,
mode ,
timeout
```

```
デバイスID
WDT動作モード
タイムアウト時間
```

### パラメータ

**devID**

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

**mode**

WDT動作モード。現在未使用「0」を指定して下さい。

**timeout**

タイムアウト時間 (ms)

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
-1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
20	パラメータエラー
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

実際のタイムアウト時間は、指定した時間(timeout) + 約600 ~ 1500ms になります。

# StopWDT

## WDTストップ

### 機能

WDT (Watch Dog Timer) を停止します。

### コーディング

```
int  
StopWDT(  
    DEVID  
)
```

### コメント

devID

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	正常終了
- 1	異常終了

異常終了時、下記の原因のコマンドエラーが考えられます。

エラー番号	エラー原因
-	エラーリセットコマンド (ClearIPError) 未発行

詳細は「コマンドエラー発生時の対策と手順」の章を参照して下さい。

### 詳細情報

## ResetWDT

## WDTリセット

### 機能

WDT (Watch Dog Timer) カウンタをリセットします。

### コーディング

```
void  
ResetWDT(  
    DEVID  
)
```

devID

### コメント

デバイスID

### パラメータ

devID

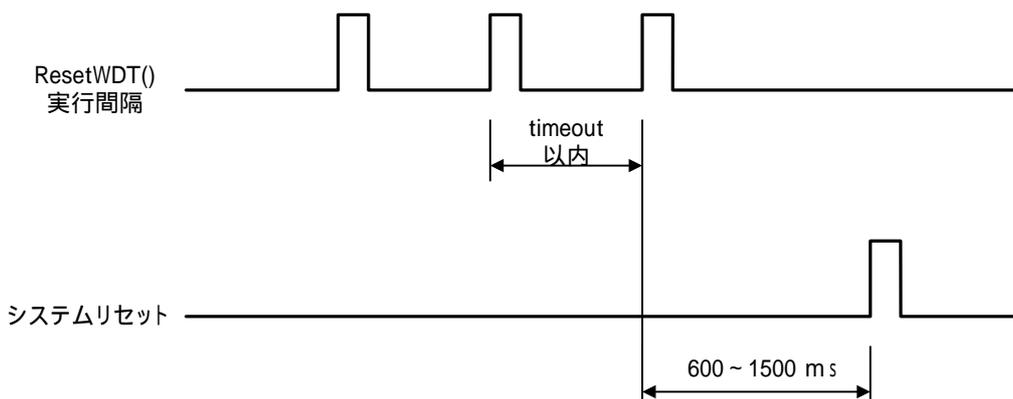
デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

なし。

### 詳細情報

本コマンドは、StartWDT()コマンドで指定したタイムアウト時間(timeout)(ms)以下のサイクルで実行してください。WDTリセットが実施されない場合、約600~1500ms後にシステムリセットされます。



# GetWDTStatus

## WDTステータスの読み出し

### 機能

WDT ( Watch Dog Timer ) のステータス情報を読み出します。

### コーディング

```
int
GetWDTStatus(
    DEVID
)
```

devID

### コメント

デバイスID

### パラメータ

devID

デバイスID。OpenIPDevコマンドで得られた対応するボード番号のデバイスIDを指定して下さい。

### リターン値

リターン値は int 型で返されます。詳細を下表に示します。

リターン値	内容
0	通常のPOWER ONリセットからの起動
1	ソフトウェアリセットからの起動
2	WDTタイムアウトでリセット発生

### 詳細情報

\_ResetIPSys( )コマンドのリセットはWDT機能を使用しているため、\_ResetIPSys( )コマンド実行後、GetWDTStatus( )コマンドでWDTステータスを読み出すと”ソフトウェアリセットからの起動”となります。

## 付録1 カメラと画面サイズ

以下にカメラの走査方式とビデオフレームサイズおよび画像メモリサイズを示します。

カメラ映像を入力する場合には、入力する画像メモリの大きさはそのビデオフレームサイズの大きさ以上である必要があります。表に示すカメラのビデオフレームサイズに合わせて、AllocImgコマンドで画像メモリを確保して下さい。

尚、カメラの走査方式とビデオフレームサイズは、SetVideoFrameコマンドで設定します。デフォルトでは、インタレースモードで512×480に設定されています。

### (1) SVP-330(NTSC対応版)

カメラタイプ	内容	走査方式	ビデオ フレーム サイズ 1	画像メモリ サイズ 1	ビデオ フレーム サイズ 2	画像メモリ サイズ 2	ビデオ フレーム サイズ 3	画像メモリ サイズ 3
BW_CAMERA	モノクロカメラ	INTERLACE	512x480	512x480	640x480	640x480	512x440	512x440
YUV_CAMERA	CVBSカラーカメラ	INTERLACE	512x480	512x480	640x480	640x480	512x440	512x440
BW_FRESET_CAMERA	モノクロカメラ(外部同期フレームリセットモード)	INTERLACE	512x480	512x480	-	-	-	-

各カメラのビデオフレームサイズと画像メモリサイズのデフォルト値です。

---

超小型画像処理ボード SVP - 330  
Software Development Kit  
コマンドリファレンス Version 2.0

(C) 株式会社 ルネサス北日本セミコンダクタ

開発元

株式会社 ルネサス北日本セミコンダクタ

電子機器本部	〒992-0021 山形県米沢市花沢3091-6 TEL 0238-22-7755 FAX 0238-22-6570
電子機器営業部	〒105-0004 東京都港区新橋5-11-3 (新橋住友ビル8階) TEL 03-5733-4550 (代) FAX 03-5733-4660
技術サポート窓口	E-Mail <a href="mailto:vp.support@kitasemi.renesas.com">vp.support@kitasemi.renesas.com</a> URL <a href="http://www.kitasemi.renesas.com">http://www.kitasemi.renesas.com</a>

---