

DP-DS820/DP-DS820 (A)
プリンターステータスAPI
(DS820Stat.dll, DS820Stat64.dll)

関数リファレンス

Ver. 1.01_D

大日本印刷株式会社

2017.09.29

改訂履歴管理表		制定日 2016/02/29			頁 1/1	
改訂頁	改 訂 内 容	改訂番号	ドキュメント 番号	改定年月日	承認	作成
	・ 新規作成	0.1.0.0	0.10	2016/02/29		
P5	・ メディア種別の取得がスタンバイ状態を解除しない関数に変更。	0.1.0.0	0.10_1	2016/03/18		
P7 P43, 44	・ 以下のAPI関数を追加 -ガンマテーブルの設定 -ガンマテーブルチェックサム取得 -データテーブルのクリア	0.2.0.0	0.20	2016/05/27		
P23 P15 P42 P25 P30, 32	以下の項が更新されました。 ・ メディア種別の取得 メディア種別文字列の表を追加。 ・ カウンタ値の取得 カウンタ カウントアップ動作の表を更新。 ・ オーバーコート仕上げ制御コマンド 文字定数の誤記を訂正。 ・ USB iSerialNumberの使用可否設定コマンド 初期値変更 “使用しない(0)” -> “使用する(1)” 以下の表記が変更されました。 ・ RXメディア -> SDメディア ・ HDMメディア -> PPメディア	0.2.0.0	0.20_1	2016/06/24		
P41, 42 P7, 43	・ パノラマ印画開始可能チェックの戻り値を追加 ・ 連続パノラマ印画設定関数を追加	1.0.0.0	1.00_D	2016/09/13		
P5	・ BiDi通信処理の排他制御を追加しました。 (Windows10 更新プログラム不具合対策。内部処理変更。)	1.0.1.0	1.01_D	2017/09/29		

目次

はじめに	5
API 関数一覧	6
DLL の初期化	8
バージョン情報の取得	8
センサ情報の取得	9
解像度の取得	10
メディアコードの取得	11
プリンターステータスの取得	12
カウンタ値の取得	14
カウンタ値のクリア	16
カウンタ値 (P) の設定	16
空きバッファ数の取得	17
残り発行枚数の取得	17
メディア残り発行枚数の取得	18
ハーフサイズ換算メディア残り発行枚数の取得	19
メディア初期枚数の取得	20
メディアロット間色バラツキ補正值の取得	21
メディアタグ情報の取得	22
メディア種別の取得	23
RF-ID タグ予備データの取得	23
プリンターシリアルナンバーの取得	24
USB iSerialNumber の使用可否設定コマンド	25
USB iSerialNumber の使用可否確認コマンド	25
ファームウェア書き換えモードへの切り替えコマンド	26
ファームウェア書き換えデータの送信コマンド	26
色制御データクリアコマンド	27
色制御データの送信コマンド	27
色制御データバージョン設定コマンド	28
色制御データバージョン取得コマンド	29
色制御データバージョン取得コマンド(種別指定)	29
色制御データバージョン取得コマンド(種別、メディア指定)	30
色制御データチェックサム取得コマンド	30
色制御データチェックサム取得コマンド(種別指定)	31
色制御データチェックサム取得コマンド(種別、メディア指定)	32
カッター制御コマンド	33
フルカッターセットアップコマンド	34
フルカッターセットアップ拡張コマンド	36
プリント速度指定コマンド	38
スタンバイモード移行時間設定	39
スタンバイモード移行時間設定値の取得	39
メディアエンド保持モード設定	40
メディアエンド保持モード設定値の取得	40
パノラマ印画開始可能チェック	41
連続パノラマ印画設定	43
オーバーコート仕上げ制御コマンド	44
ガンマテーブルの設定	45
ガンマテーブルチェックサム取得	46
データテーブルのクリア	46

汎用設定系コマンド	47
汎用取得系コマンド	47
サンプルプログラム説明	48

はじめに



本書の著作権は権利者にあります。一部または全部の無断転載・無断複製を禁止します。



本書の内容は予告なく変更することがあります。



Microsoft, Windows は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。



仕向けを表す (A) の付かない DP-DS820 を米国へ転売したり、米国で使用したりすることはできません。

適用の範囲

本書は、DP-DS820/DP-DS820 (A) プリンターステータス API 関数の取り扱いに関して記述したものです。

対応 OS、動作環境

本 API は Windows 7、Windows 8、および Windows 10 で動作します。
Windows10 環境では、ver. 1.0.1.0 以降の DLL を使用してください。

API 関数一覧

API 関数の一覧を以下に示します。

またプリンターがスタンバイ中に API 関数を使用した場合にスタンバイ状態が「解除される/解除されない」についても記載します。

■ 注意点

- ・ C 言語で使用する場合、“DS820Stat.h”をインクルードし、リンクするライブラリに DS820Stat.lib を追加してください。
実行時は参照できるパスに“DS820Stat.dll”(32bitOS 環境)を配置してください。(64bitOS 環境は“DS820Stat64.dll”)
- ・ VB.NET で使用する場合、VB から DLL 関数の引数にポインタを直接渡せないため、DLL と同一名の VB.NET 用ラッパー関数を用意しました。定義は DS820Stat.vb ファイルに記述をして有りますので、DS820Stat.vb を VB プロジェクトに追加して使用してください。
DS820Stat.vb は、32bitOS と 64bitOS 環境を切り替えて使用出来るように定義をして有ります。
32bitOS の場合は DS820Stat.vb の先頭行で #Const x64=False とし、64bitOS の場合は #Const x64=True として使用してください。

	API 関数名		スタンバイ 状態	備考
DLL の初期化	CvInitialize()	PortInitialize()	解除しない	
バージョン情報の取得	CvGetVersion()	GetFirmwVersion()	解除しない	
センサ情報の取得	CvGetSensorInfo()	GetSensorInfo()	解除する	
解像度の取得	CvGetResolutionH() CvGetResolutionV()	GetResolutionH() GetResolutionV()	解除する	
メディアコードの取得	CvGetMedia()	GetMedia()	解除しない	
プリンターステータスの取得	CvGetStatus()	GetStatus()	解除しない	
カウンタ値の取得	CvGetCounterL() CvGetCounterA() CvGetCounterB()	GetCounterL() GetCounterA() GetCounterB() GetCounterP() GetCounterMatte() GetCounterM()	解除する	
カウンタ値のクリア	CvSetClearCounterA() CvSetClearCounterB()	SetClearCounterA() SetClearCounterB() SetClearCounterM()	解除する	
カウンタ値(P)の設定		SetCounterP()	解除する	
空きバッファ数の取得	CvGetFreeBuffer()	GetFreeBuffer()	解除する	
残り発行枚数の取得	CvGetPQTY()	GetPQTY()	解除する	
メディア残り発行枚数の取得	CvGetMediaCounter()	GetMediaCounter()	解除しない	
ハーフサイズメディア残り発行枚数の取得		GetMediaCounterH()	解除しない	
メディア初期枚数の取得		GetInitialMediaCount()	解除しない	
メディアロット間色バラツキ補正値の取得	CvGetMediaColorOffset()	GetMediaColorOffset()	解除する	
メディアタグ情報の取得	CvGetMediaLotNo()	GetMediaLotNo()	解除しない	
メディア種別の取得		GetRfidMediaClass()	解除しない	
RF-ID 予備データの取得		GetRfidReserveData()	解除する	
プリンターシリアルナンバーの取得	CvGetSerialNo();	GetSerialNo()	解除しない	
USB iSerialNumber の使用可否の設定		SetUSBSerialEnable()	解除する	
USB iSerialNumber 使用可否設定の取得コマンド		GetUSBSerialEnable()	解除する	
ファームウェア書き換えモードへの切り替えコマンド	CvSetFirmwUpdateMode()	SetFirmwUpdateMode()	解除する	
ファームウェア書き換えデータ送信コマンド	CvSetFirmwDataWrite()	SetFirmwDataWrite()	解除する	VB.NET ラッパー関数

	API 関数名		スタンバイ 状態	備考
色制御データのクリア	CvSetColorDataClear()	SetColorDataClear()	解除する	
色制御データの送信コマンド	CvSetColorDataWrite()	SetColorDataWrite()	解除する	VB.NET ラッパ関数
色制御データのバージョンの設定	CvSetColorDataVersion()	SetColorDataVersion()	解除する	VB.NET ラッパ関数
色制御データのバージョンの取得	CvGetColorDataVersion()	GetColorDataVersion()	解除する	
色制御データのバージョンの取得 (種別指定)		GetColorDataVersionRes()	解除する	
色制御データのバージョンの取得 (種別、メディア指定)		GetColorDataVersionResEx()	解除する	
色制御データのチェックサムの取得	CvGetColorDataChecksum()	GetColorDataChecksum()	解除する	
色制御データのチェックサムの取得 (種別指定)		GetColorDataChecksumRes()	解除する	
色制御データチェックサム取得 (種別、メディア指定)		GetColorDataChecksumResEx()	解除する	
カッター制御の設定		SetCutterMode()	解除する	
フルカッターセットアップコマンド		SetFullCutterSetup()	解除する	
フルカッターセットアップ拡張コマンド		SetFullCutterSetupEX()	解除する	
プリント速度指定		SetPrintSpeed()	解除する	
スタンバイモード移行時間の設定		SetStandbyTime()	解除する	
スタンバイモード移行時間の取得		GetStandbyTime()	解除する	
メディアエンド保持モードの設定		SetEndKeepMode()	解除する	
メディアエンド保持モード設定値の取得		GetEndKeepMode()	解除する	
パノラマ印画開始可能チェック		GetPanoramaPrintable()	解除する	
連続パノラマ印画設定		SetContPanorama()	解除する	
オーバーコート制御コマンド		SetOvercoatFinish()	解除する	
ガンマテーブルの設定		SetGammaTable()	解除する	
ガンマテーブルチェックサム取得		GetGammaTableChecksum()	解除する	
データテーブルのクリア		SetDataTableClear()	解除する	
汎用設定形コマンド	CvSetCommand()	SetCommand()		VB.NET ラッパ関数
汎用取得系コマンド	CvGetCommandEX()	GetCommandEX()		VB.NET ラッパ関数

DLL の初期化

【書式】	long PortInitialize(LPWSTR p); long CvInitialize(LPWSTR p);
【引数】	p : ポート名WSTR へのポインタ (2 バイト文字)
【戻り】	成功 : ポート番号 失敗 : -1
【説明】	API 内を初期化し、ポート番号を返します。 複数ポートを使用する場合、繰り返し実行し各々の番号を取得してください。
【文例】	<pre>< Visual C > long DS820; if((DS820 = PortInitialize(L"USB001")) < 0){ // error } < VB.NET > Dim DS820 As Integer DS820 = PortInitialize("USB001"); If DS820 < 0 Then GoTo Error</pre>

バージョン情報の取得

【書式】	long GetFirmwVersion(long lPortNum, LPSTR p); long CvGetVersion(long lPortNum, LPSTR p);
【引数】	lPortNum : ポート番号 p : 受信バッファへのポインタ
【戻り】	成功 : バッファ p へ受信した文字数 失敗 : -1
【説明】	<p>プリンターのバージョン情報を文字列バッファへ受信します。</p> <p>[バージョン情報]</p> <ul style="list-style-type: none"> DS820A xx.xx (米国向け) DS820 xx.xx (その他の国向け) <p>xx.xx はバージョンを表します。</p>
【文例】	<pre>< Visual C > char rbuf[256]; if(GetFirmwVersion(DS820, (LPSTR)rbuf) > 0){ // 処理 } < VB.NET > Dim s As String = New String("", 255) Dim i As Integer i = GetFirmwVersion(DS820, s) If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"</pre>

センサ情報の取得

【書式】 long GetSensorInfo(long lPortNum, LPSTR p);
 long CvGetSensorInfo(long lPortNum, LPSTR p);

【引数】 lPortNum : ポート番号
 p : 受信バッファへのポインタ

【戻り】 成功 : バッファ p へ受信した文字数
 失敗 : -1

【説明】 各種センサの情報（AD 値）を文字列バッファへ受信します。

【文例】 < Visual C >
 char rbuf[256];
 if(GetSensorInfo(DS820, (LPSTR)rbuf) > 0){
 // 処理
 }

 < VB.NET >
 Dim s As String = New String("", 255)
 Dim i As Integer
 i = GetSensorInfo(DS820, s)
 If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"

【センサ情報】

HDT-***;	ヘッド温度検出センサ	
MDT-***;	メディア温度検出センサ	
PMK-***;	用紙マーク検出センサ	
RML-***;	左リボンマーク検出センサ	(未使用、常に 000 が返ります)
RMC-***;	中央リボンマーク検出センサ	(未使用)
RMR-***;	右リボンマーク検出センサ	(未使用)
PSZ-***;	用紙サイズ検出センサ	(未使用、常に 000 が返ります)
PNT-***;	用紙ノッチ検出センサ	(未使用、常に 000 が返ります)
PJM-***;	用紙ジャム検出センサ	(未使用、常に 000 が返ります)
PED-***;	ペーパーエンド検出センサ	
PET-***;	ペーパーエンブティ検出センサ	(未使用、常に 000 が返ります)
HDV-***;	ヘッド電圧検出センサ	
HMD-***;	湿度センサ	
GSR-***;	カラーセンサ(赤)	
GSG-***;	カラーセンサ(緑)	
GSB-***;	カラーセンサ(青)	

解像度の取得

【書式】	<pre> long GetResolutionH(long IPortNum); long CvGetResolutionH(long IPortNum); long GetResolutionV(long IPortNum); long CvGetResolutionV(long IPortNum); </pre>
【引数】	IPortNum : ポート番号
【戻り】	成功 : 水平、垂直解像度(dpi) 失敗 : -1
【説明】	GetResolutionH(), CvGetResolutionH() は水平解像度(dpi)を返します。 GetResolutionV(), CvGetResolutionV() は垂直解像度(dpi)を返します。
【文例】	<pre> < Visual C > long rh; if((rh = GetResolutionH(DS820)) >= 0){ // rh に水平解像度値 } < VB.NET > Dim i As Integer i = GetResolutionH(DS820) If i >= 0 Then Text1.Text = Str(i) Else Text1.Text = "ERROR!" </pre>

メディアコードの取得

- 【書式】 long GetMedia(long lPortNum, LPSTR p);
 long CvGetMedia(long lPortNum, LPSTR p);
- 【引数】 lPortNum : ポート番号
 p : 受信バッファへのポインタ
- 【戻り】 成功 : バッファ p へ受信した文字数
 失敗 : -1

【説明】 メディアコードを文字列バッファへ受信します。

【文例】 < Visual C >
 char rbuf[256];
 if(GetMedia(DS820, (LPSTR)rbuf) > 0){
 // 処理
 }

 < VB.NET >
 Dim s As String = New String("", 255)
 Dim i As Integer
 i = GetMedia(DS820, s)
 If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"

【メディアコード】 本プリンターでは、使用するメディアを全て 5 桁の 10 進数で表します。
 メディアコード取得コマンドにより返信するデータは、各メディアに割り当てた 5 桁のコード（ASCII 数字）になります。

4 桁目 (0n000)	用紙タイプ	3, 2 桁目、 (00nn0)	用紙サイズ	1 桁目 (0000n)	位置検出マーク
00000 番	標準紙	00500 番	8x10	00000 番	マーク無し
01000 番	シール紙	00510 番	8x12	00001 番	マーク有り
		00600 番	A4		

コード例

用紙サイズ	用紙タイプ	(サイズ: 幅 x 長さ)	Media Code
8x10	標準紙	(207.0 x 257.0 mm)	00500
8x12	標準紙	(207.0 x 308.0 mm)	00510
A4	標準紙	(214.0 x 300.0 mm)	00600

プリンターステータスの取得

【書式】

DWORD GetStatus(long lPortNum);

DWORD CvGetStatus(long lPortNum);

【引数】

lPortNum : ポート番号

【戻り】

成功 : ステータス

失敗 : STATUS_ERROR

【説明】

プリンターステータスを返します。

ステータスのビット位置は DS820Stat.h にマクロ定義されています (VB は DS820Stat.vb)。

シンボルの意味は下記の通りです。

GROUP_USUALLY	[通常のステータス]	グループを示すビット
GROUP_SETTING	[ユーザー設定エラー]	グループを示すビット
GROUP_HARDWARE	[ハードエラー]	グループを示すビット
GROUP_SYSTEM	[内部エラー]	グループを示すビット
GROUP_FLSHPROG	[書換えモード]	グループを示すビット

STATUS_ERROR	ステータス受信エラー
--------------	------------

STATUS_USUALLY_IDLE	アイドル
STATUS_USUALLY_PRINTING	印画中
STATUS_USUALLY_PAPER_END	ペーパーエンド
STATUS_USUALLY_RIBBON_END	リボンエンド
STATUS_USUALLY_COOLING	ヘッド冷却中
STATUS_USUALLY_MOTCOOLING	モーター冷却中
STATUS_USUALLY_STANDBY_MODE	スタンバイモード

STATUS_SETTING_COVER_OPEN	カバーオープン
STATUS_SETTING_PAPER_JAM	ペーパージャム
STATUS_SETTING_RIBBON_ERR	リボンエラー (検出ミス、リボン切れ)
STATUS_SETTING_PAPER_ERR	用紙設定ミス
STATUS_SETTING_DATA_ERR	データエラー (不正コマンド)
STATUS_SETTING_SCRAPBOX_ERR	クズ BOX 無しエラー

STATUS_HARDWARE_ERR01	ヘッド電圧異常
STATUS_HARDWARE_ERR02	ヘッド位置エラー
STATUS_HARDWARE_ERR03	電源ファン停止
STATUS_HARDWARE_ERR04	カッターエラー (噛み込み等)
STATUS_HARDWARE_ERR05	ピンチローラー位置エラー
STATUS_HARDWARE_ERR06	ヘッド温度異常
STATUS_HARDWARE_ERR07	メディア温度異常
STATUS_HARDWARE_ERR08	リボンテンション異常
STATUS_HARDWARE_ERR09	RF-ID モジュール異常
STATUS_HARDWARE_ERR10	モーター温度異常

STATUS_SYSTEM_ERR01	システムエラー
---------------------	---------

STATUS_FLSHPROG_IDLE	書換えデータ受信待ち
STATUS_FLSHPROG_DATA_ERR1	転送データ関連エラー
STATUS_FLSHPROG_DEVICE_ERR1	デバイス関連エラー
STATUS_FLSHPROG_OTHERS_ERR1	その他エラー

【文例】

< Visual C >

```

include "DS820Stat.h"
long stat;

stat = GetStatus( DS820 );
if( stat & GROUP_USUALLY ){ // 通常のステータスグループ
    switch( stat ){
        case STATUS_USUALLY_IDLE: ;
        case STATUS_USUALLY_PRINTING: ;
        case STATUS_USUALLY_PAPER_END: ;
        :
    }
}
if( stat & GROUP_SETTING ){ // 設定エラーステータスグループ
    switch( stat ){
        case STATUS_SETTING_COVER_OPEN: ;
        :
        :
    }
}
}

```

< VB.NET >

```

Dim stat as Integer

stat = GetStatus(DS820)
If stat And GROUP_USUALLY Then ' 通常のステータスグループ
    Select Case stat
        Case STATUS_USUALLY_IDLE: Text1.Text = "IDLE"
        Case STATUS_USUALLY_PRINTING: Text1.Text = "PRINTING"
        Case STATUS_USUALLY_PAPER_END: Text1.Text = "PAPER_END"
        Case STATUS_USUALLY_RIBBON_END: Text1.Text = "RIBBON_END"
        Case STATUS_USUALLY_COOLING: Text1.Text = "COOLING"
    End Select
ElseIf stat And GROUP_SETTING Then ' 設定エラーステータスグループ
    ' 設定エラー (省略)
ElseIf stat And GROUP_HARDWARE Then ' ハードエラーステータスグループ
    ' ハードエラー (省略)
ElseIf stat And GROUP_SYSTEM Then ' システムエラーステータスグループ
    ' システムエラー (省略)
End If

```

カウンタ値の取得

【書式】 long GetCounterL(long IPortNum);
 long CvGetCounterL(long IPortNum);

 long GetCounterA(long IPortNum);
 long CvGetCounterA(long IPortNum);

 long GetCounterB(long IPortNum);
 long CvGetCounterB(long IPortNum);

 long GetCounterP(long IPortNum);

 long GetCounterMatte(long IPortNum);
 long GetCounterM(long IPortNum);

【引数】 IPortNum : ポート番号

【戻り】 成功 : カウンタの値
 失敗 : -1

【説明】 GetCounterL(), CvGetCounterL() は、ライフカウンタの値を返します。
 GetCounterA(), CvGetCounterA() は、カウンタ A の値を返します。
 GetCounterB(), CvGetCounterB() は、カウンタ B の値を返します。
 GetCounterP() は、カウンタ P の値を返します。
 GetCounterMatte() は、マットカウンタの値を返します。
 GetCounterM() は、カウンタ M の値を返します。

各カウンタは、1 枚発行する毎に 1 カウントアップします。多画面割付プリントの場合は 2 枚目の発行時に 2 ウントアップします。

カウンタ P は電源投入時に 0 に初期化されます。SetCounterP() 関数で任意の値に設定出来ます。
 カウンタ P は 1 画像排出毎に 1 ずつカウントアップします。

マットカウンタとカウンタ M は、オーバーコート仕上げをマットで印画したときにカウントアップします（ライフカウンタ、カウンタ A/B もカウントアップします）。

カウンタ M は SetClearCounterM() 関数でクリア可能です。
 カウントアップ仕様は、上記ライフカウンタ、カウンタ A/B と同様です。

【文例】 < Visual C >
 long counter;
 if((counter = GetCounterL(DS820)) >= 0){
 // counter にライフカウンタ値
 }

 < VB.NET >
 Dim c As Integer
 c = GetCounterL(DS820)
 If c >= 0 Then Text1.Text = Str(c) Else Text1.Text = "ERROR!"

カウンタ カウントアップの動作

カウントアップするタイミングは、プリント画像が正常にカットされた後です。
エラーが発生したときはカウントアップされません。

	プリントサイズ		Counter L/A/B・ Matte/M	Counter P
シングルカット	8x10		+1	+1
	8x12		+1	+1
	8x4		+1	+1
	8x5		+1	+1
	8x6		+1	+1
	8x8		+1	+1
	A5 Format		+1	+1
	A4 Format		+1	+1
マルチカット	8x4x2	1 面目	---	+1
		2 面目	+1	+1
	8x5x2	1 面目	---	+1
		2 面目	+1	+1
	8x6x2	1 面目	---	+1
		2 面目	+1	+1
	A5x2	1 面目	---	+1
		2 面目	+1	+1
	8x4x3	1 面目	---	+1
		2 面目	---	+1
		3 面目	+1	+1
2inch カット	8x4	1 枚目	---	+1
		2 枚目	+1	+1
	8x6	1 枚目	---	+1
		2 枚目	---	+1
		3 枚目	+1	+1
	8x8	1 枚目	---	+1
		2 枚目	---	+1
		3 枚目	---	+1
		4 枚目	+1	+1
	8x10	1 枚目	---	+1
		2 枚目	---	+1
		3 枚目	---	+1
		4 枚目	---	+1
		5 枚目	+1	+1
	8x12	1 枚目	---	+1
		2 枚目	---	+1
		3 枚目	---	+1
		4 枚目	---	+1
		5 枚目	---	+1
		6 枚目	+1	+1

カウンタ値のクリア

【書式】	<pre> BOOL SetClearCounterA(long IPortNum); BOOL CvSetClearCounterA(long IPortNum); BOOL SetClearCounterB(long IPortNum); BOOL CvSetClearCounterB(long IPortNum); BOOL SetClearCounterM(long IPortNum); </pre>
【引数】	<pre> IPortNum : ポート番号 </pre>
【戻り】	<pre> 成功 : True 失敗 : False </pre>
【説明】	<p>SetClearCounterA(), CvSetClearCounterA()は、カウンタ A の値をクリアします。 SetClearCounterB(), CvSetClearCounterB()は、カウンタ B の値をクリアします。 SetClearCounterM()は、カウンタ M の値をクリアします。</p>
【文例】	<pre> < Visual C > if(SetClearCounterA(DS820)){ // カウンタ A クリア成功 } < VB.NET > If SetClearCounterA(DS820) <> FalseThen ' カウンタ A クリア成功 End If </pre>

カウンタ値 (P) の設定

【書式】	<pre> BOOL SetCounterP(long IPortNum, long ICounter); </pre>
【引数】	<pre> IPortNum : ポート番号 ICounter : 設定するカウンタ値 </pre>
【戻り】	<pre> 成功 : True 失敗 : False </pre>
【説明】	<p>Counter P に任意の値をセットします。プリンターの電源切ると 0 に初期化されます。</p>
【文例】	<pre> < Visual C > if(SetCounterP(DS820, 100)){ // カウンタ値 P を設定 } < VB.NET > If SetCounterP(DS820, 100) <> False Then ' カウンタ値 P を設定 End If </pre>

空きバッファ数の取得

【書式】	long GetFreeBuffer(long IPortNum); long CvGetFreeBuffer(long IPortNum);
【引数】	IPortNum : ポート番号
【戻り】	成功 : 印画バッファの空き数 失敗 : -1
【説明】	印画バッファの空き数を返します。
【文例】	<div>< Visual C ></div> <pre>long bn; if((bn = GetFreeBuffer(DS820)) >= 0){ // bn に空きバッファ数 }</pre> <div>< VB.NET ></div> <pre>Dim bn As Integer bn = GetFreeBuffer(DS820)</pre>

残り発行枚数の取得

【書式】	long GetPQTY(long IPortNum); long CvGetPQTY(long IPortNum);
【引数】	IPortNum : ポート番号
【戻り】	成功 : 残り発行枚数 失敗 : -1
【説明】	残り発行枚数を返します。
【文例】	<div>< Visual C ></div> <pre>long number; if((number = GetPQTY(DS820)) >= 0){ // number に残り発行枚数 }</pre> <div>< VB.NET ></div> <pre>Dim number As Integer number = GetPQTY(DS820)</pre>

メディア残り発行枚数の取得

【書式】 long GetMediaCounter(long lPortNum);
 long CvGetMediaCounter(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : メディア残り発行枚数
 失敗 : -1

【説明】 メディア残り発行枚数を返します。

メディアの使用条件により、実際の印画可能枚数とメディアタグ枚数が異なる場合があります。
メディアエンドの検出は、GetStatus() 関数のリボンエンド、ペーパーエンドと併用してください。

■メディア毎の枚数初期値

メディアサイズ	メディア発行枚数 初期値	印画可能枚数
8x10	130	130
8x12	110	110
A4	110	110

【文例】 < Visual C >
 long number;
 if((number = GetMediaCounter(DS820)) >= 0){
 // number にメディア残り発行枚数
 }

 < VB.NET >
 Dim number As Integer
 number = GetMediaCounter(DS820)

ハーフサイズ換算メディア残り発行枚数の取得

【書式】 long GetMediaCounterH(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : ハーフサイズ換算メディア残り発行枚数
失敗 : -1

【説明】 8×10 サイズのメディアを使用した場合は 8×5 サイズ、8×12 の場合は 8×6 サイズ、A4 サイズの場合は A5 サイズの残り発行枚数を返します。

メディアの使用条件により、実際の印画可能枚数とメディアタグ枚数が異なる場合があります。メディアエンドの検出は、GetStatus() 関数のリボンエンド、ペーパーエンドと併用してください。

■メディア毎の枚数初期値

メディアサイズ	メディア発行枚数 初期値	印画可能枚数
8x10	260	260
8x12	220	220
A4	220	220

【文例】

< Visual C >

```
long number;
if(( number = GetMediaCounterH( DS820 )) >= 0 ){
    // number にメディア残り発行枚数
}
```

< VB.NET >

```
Dim number As Integer
number = GetMediaCounterH(DS820)
```

メディア初期枚数の取得

【書式】	<code>long GetInitialMediaCount(long lPortNum);</code>
【引数】	<code>lPortNum</code> : ポート番号
【戻り】	成功 : メディア初期枚数 失敗 : -1
【説明】	メディアの初期枚数を返します。
【文例】	<div>< Visual C ></div> <pre>long number; if((number = GetInitialMediaCount (DS820)) >= 0){ // number にメディア初期枚数 }</pre> <div>< VB.NET ></div> <pre>Dim number As Integer number = GetInitialMediaCount (DS820)</pre>

メディアロット間色バラツキ補正値の取得

- 【書式】 long GetMediaColorOffset(long lPortNum);
 long CvGetMediaColorOffset(long lPortNum);
- 【引数】 lPortNum : ポート番号
- 【戻り】 成功 : メディアロット間色バラツキ補正値
 失敗 : -1
- 【説明】 リボンのロット間での色バラツキ補正値を返します。
- 【文例】 < Visual C >
 long offset;
 if((offset = GetMediaColorOffset(DS820)) >= 0){
 // offset に色補正値
 }

 < VB.NET >
 Dim offset As Integer
 offset = GetMediaColorOffset(DS820)
- 【例】 各色の補正値を2バイト単位で表現します。
 offset=169082895 (10進) の場合、16進表記で 0x0A14000F となり以下の内容になります。

色	補正値
Yellow	10 (0x0A)
Magenta	20 (0x14)
Cyan	0 (0x00)
Op	15 (0x0F)

メディアタグ情報の取得

- 【書式】 long GetMediaLotNo(long lPortNum, LPSTR p);
 long CvGetMediaLotNo(long lPortNum, LPSTR p);
- 【引数】 lPortNum : ポート番号
 p : 受信バッファへのポインタ
- 【戻り】 成功 : バッファ p へ受信した文字数
 失敗 : -1
- 【説明】 RF-ID タグに記録されているメディアロット情報を返します。
- 【文例】 < Visual C >
 char rbuf[256];
 if(GetMediaLotNo(DS820, (LPSTR)rbuf) > 0){
 // 処理
 }

 < VB.NET >
 Dim s As String = New String("", 255)
 Dim i As Integer
 i = GetMediaLotNo(DS820, s)
 If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"
- 【メディアタグユーザー情報】
 ML***** ユーザー専用情報<16Byte>

メディア種別の取得

- 【書式】 `long GetRfidMediaClass(long lPortNum, LPSTR p);`
- 【引数】 `lPortNum`: ポート番号
`p`: 受信バッファへのポインタ
- 【戻り】 成功: バッファ `p` へ受信した文字数
 失敗: -1
- 【説明】 RF-ID タグに記録されているメディア種別データを返します。(4 桁の ASCII 文字)

・メディア種別情報

取得文字列	詳細
0001	Digital (SD) メディア
0003	Pure Premium (PP) メディア

- 【文例】
- ```
< Visual C >
char rbuf[256];
if(GetRfidMediaClass(DS820, (LPSTR)rbuf) > 0){
 // 処理
}

< VB.NET >
Dim s As String = New String("", 255)
Dim i As Integer
i = GetRfidMediaClass(DS820, s)
If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"
```

## RF-ID タグ予備データの取得

---

- 【書式】 `long GetRfidReserveData ( long lPortNum, LPSTR p, DWORD dwPage );`
- 【引数】 `lPortNum`: ポート番号  
`p`: 受信バッファへのポインタ  
`dwPage`: 予備データのページ番号
- 【戻り】 成功: バッファ `p` へ受信した文字数  
 失敗: -1
- 【説明】 RF-ID タグに記録されている予備データを返します。(4 桁の ASCII 文字)
- 【文例】
- ```
< Visual C >
char rbuf[256];
if(GetRfidReserveData ( DS820, (LPSTR)rbuf , 1 ) > 0 ){ // ページ 1 を指定
    // 処理
}

< VB.NET >
Dim s As String = New String("", 255)
Dim i As Integer
i = GetRfidReserveData (DS820, s, 2 ) 'ページ 2 を指定
If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"
```

プリンターシリアルナンバーの取得

【書式】	long GetSerialNo(long lPortNum, LPSTR p); long CvGetSerialNo(long lPortNum, LPSTR p);
【引数】	lPortNum : ポート番号 p : 受信バッファへのポインタ
【戻り】	成功 : バッファ p へ受信した文字数 失敗 : -1
【説明】	プリンターのシリアル No（製造番号）を取得します。
【文例】	<div>< Visual C ></div> <pre>char rbuf[256]; if(GetSerialNo(DS820, (LPSTR)rbuf) > 0){ // 処理 }</pre> <div>< VB.NET ></div> <pre>Dim s As String = New String("", 255) Dim i As Integer i = GetSerialNo(DS820, s) If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"</pre>

USB iSerialNumber の使用可否設定コマンド

【書式】	B00L SetUSBSerialEnable(long lPortNum, DWORD dwEnable);
【引数】	lPortNum : ポート番号 dwEnable : USB iSerialNumber を使用する(1) USB iSerialNumber を使用しない(0)
【戻り】	成功 : True 失敗 : False
【説明】	<p>本プリンターを USB 接続する時の USB iSerialNumber 使用可否を設定します。USB iSerialNumber を使用する場合はシリアル番号は、製品固有情報である製品シリアル番号となります。</p> <p>本設定の初期値は、“USB iSerialNumber を使用する(1)” です。設定はプリンターに記憶され、電源を OFF しても保持されます。</p>
【注意】	設定変更は、プリンターの電源が再投入された際に有効となります。
【文例】	<pre>< Visual C > if(SetUSBSerialEnable(DS820, 1)){ // USB iSerialNumber を使用する設定 // 設定成功 } < VB.NET > if SetUSBSerialEnable(DS820, 0) <> False Then ' USB iSerialNumber を使用しない設定 ' 設定成功 End if</pre>

USB iSerialNumber の使用可否確認コマンド

【書式】	long GetUSBSerialEnable(long lPortNum);
【引数】	lPortNum : ポート番号
【戻り】	成功 : USB iSerialNumber を使用する(1) USB iSerialNumber を使用しない(0) 失敗 : -1
【説明】	本プリンターを USB 接続する時の USB iSerialNumber 使用可否を取得します。
【文例】	<pre>< Visual C > if(GetUSBSerialEnable(DS820) == 1){ // 使用する } < VB.NET > if GetUSBSerialEnable (DS820) = 1 Then ' 使用する End If</pre>

ファームウェア書き換えモードへの切り替えコマンド

【書式】	<pre> BOOL SetFirmwUpdateMode(long lPortNum); BOOL CvSetFirmwUpdateMode(long lPortNum); </pre>
【引数】	<pre> lPortNum : ポート番号 </pre>
【戻り】	<pre> 成功 : True 失敗 : False </pre>
【説明】	プリンターをファームウェア書き換えモードに切り替えます。
【文例】	<pre> < Visual C > if(SetFirmwUpdataMode(DS820) == True){ // 切り替え成功 } < VB.NET > if (SetFirmwUpdateMode(DS820) <> False) Then ' 切り替え成功 End If </pre>

ファームウェア書き換えデータの送信コマンド

【書式】	<pre> BOOL SetFirmwDataWrite(long lPortNum, LPSTR lpData, DWORD dwDataLen); BOOL CvSetFirmwDataWrite(long lPortNum, LPSTR lpData, DWORD dwDataLen); </pre>
【引数】	<pre> lPortNum : ポート番号 lpData : 書き換えデータが格納されたバッファのポインタ dwDataLen : データの文字数 </pre>
【戻り】	<pre> 成功 : True 失敗 : False </pre>
【説明】	<p>プリンターのファームウェア書き換えデータを転送します。</p> <p>書き換えデータはファイルで提供されるモトローラ S フォーマットです。本コマンド使用時はファイルを読み込み、書き換えデータを一時格納するバッファ（5Mbyte 程度）が必要です。</p> <p>書き換えが完了すると、プリンターは自動的に再起動して、書き換えモードは終了します。</p> <p><VB.NET ラッパー関数について></p> <p>VB.NET を使用する場合、DS820Stat.vb ファイルにラッパー関数として記述して有ります。</p> <p>VB プロジェクトに DS820Stat.vb ファイルを追加してください。</p>
【文例】	<pre> < VB .Net > Dim fd(5000000) Dim c As Long, n As Long c=0 FileOpen(1, fname, OpenMode.Binary, OpenAccess.Read) FileLength = LOF(1) For n = 0 To FileLength - 1 FileGet(1, fd(c)) c = c + 1 Next n FileClose(1) SetFirmwDataWrite(DS820, fd, c) </pre>

色制御データクリアコマンド

【書式】	<pre> BOOL SetColorDataClear(long lPortNum); BOOL CvSetColorDataClear(long lPortNum); </pre>
【引数】	<pre> lPortNum : ポート番号 </pre>
【戻り】	<pre> 成功 : True 失敗 : False </pre>
【説明】	<p>プリンターに書き込まれている色制御データをクリアします。</p> <p>色制御データを書き換える場合には、最初にこのコマンドで既存の色制御データをクリアしてください。</p>
【文例】	<pre> < Visual C > if(SetColorDataClear(DS820) == True){ //クリア成功 } < VB.NET > If SetColorDataClear(DS820) <> False Then ' クリア成功 End If </pre>

色制御データの送信コマンド

【書式】	<pre> BOOL SetColorDataWrite(long lPortNum, LPSTR lpData, DWORD dwDataLen); BOOL CvSetColorDataWrite(long lPortNum, LPSTR lpData, DWORD dwDataLen); </pre>
【引数】	<pre> lPortNum : ポート番号 lpData : 色制御データが格納されたバッファのポインタ dwDataLen : データの文字数 </pre>
【戻り】	<pre> 成功 : True 失敗 : False </pre>
【説明】	<p>色制御データを転送します。</p> <p>このコマンドで新しい色制御データを転送する前に、色制御データクリアコマンドで既存の色制御データをクリアしてください。</p> <p>色制御データは独自フォーマットのバイナリファイルで提供されます。本コマンド使用時はファイルを読み込み、書き換えデータを一時格納するバッファ（100kbyte 程度）が必要です。</p> <p><VB.NET ラッパー関数について></p> <p>VB.NET を使用する場合、DS820Stat.vb ファイルにラッパー関数として記述して有ります。</p> <p>VB プロジェクトに DS820Stat.vb ファイルを追加してください。</p>
【文例】	<pre> < VB.NET > Dim fd(100000) Dim c As Long, n As Long c=0 FileOpen(1, fname, OpenMode.Binary, OpenAccess.Read) FileLength = LOF(1) For n = 0 To FileLength - 1 FileGet(1, fd(c)) c = c + 1 Next n FileClose(1) SetColorDataWrite(DS820, fd, c) </pre>

色制御データバージョン設定コマンド

【書式】	<pre> BOOL SetColorDataVersion(long lPortNum, LPSTR lpData, DWORD dwDataLen); BOOL CvSetColorDataVersion(long lPortNum, LPSTR lpData, DWORD dwDataLen); </pre>						
【引数】	<table> <tr> <td>lPortNum :</td><td>ポート番号</td></tr> <tr> <td>lpData :</td><td>色制御データバージョンが格納されたバッファのポインタ</td></tr> <tr> <td>dwDataLen :</td><td>データの文字数</td></tr> </table>	lPortNum :	ポート番号	lpData :	色制御データバージョンが格納されたバッファのポインタ	dwDataLen :	データの文字数
lPortNum :	ポート番号						
lpData :	色制御データバージョンが格納されたバッファのポインタ						
dwDataLen :	データの文字数						
【戻り】	<table> <tr> <td>成功 :</td><td>True</td></tr> <tr> <td>失敗 :</td><td>False</td></tr> </table>	成功 :	True	失敗 :	False		
成功 :	True						
失敗 :	False						
【説明】	<p>色制御データのバージョンを設定します。 色制御データ送信コマンドで新しい色制御データを転送した後、このコマンドで色制御データのバージョンを設定してください。 設定する色制御データのバージョンは、提供される色制御データファイルのファイル名としてください。</p> <p><VB.NET ラッパー関数について> VB.NET を使用する場合、DS820Stat.vb ファイルにラッパー関数として記述して有ります。 VB プロジェクトに DS820Stat.vb ファイルを追加してください。</p>						
【文例】	<pre> < Visual C > char fname[256]; ' fname ヘバージョン情報文字列を格納 SetColorDataVersion (DS820, (LPSTR) fname, strlen(fname)); < VB.NET > Dim fname As String ' fname ヘバージョン情報文字列を格納 SetColorDataVersion(DS820, fname, Len(fname)) </pre>						

色制御データバージョン取得コマンド

【書式】	long GetColorDataVersion(long lPortNum, LPSTR p); long CvGetColorDataVersion(long lPortNum, LPSTR p);
【引数】	lPortNum : ポート番号 p : 受信バッファへのポインタ
【戻り】	成功 : バッファ p へ受信した文字数 失敗 : -1
【説明】	色制御データのバージョン情報を文字列バッファへ受信します。
【文例】	<div>< Visual C ></div> <pre>char rbuf[256]; if(GetColorDataVersion(DS820, (LPSTR)rbuf) > 0){ // 処理 }</pre> <div>< VB.NET ></div> <pre>Dim s As String = New String("", 255) Dim i As Integer i = GetColorDataVersion(DS820, s) If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"</pre>

色制御データバージョン取得コマンド(種別指定)

【書式】	long GetColorDataVersionRes(long lPortNum, LPSTR p , DWORD dwType);
【引数】	lPortNum : ポート番号 p : 受信バッファへのポインタ dwType : 300dpi 色制御バージョンデータを取得(0) 600dpi 色制御バージョンデータを取得(1) 低速印画色制御バージョンデータを取得(2) 高濃度印画色制御バージョンデータを取得(3)
	色制御バージョンデータは、装着されたメディア種別により異なります。
【戻り】	成功 : バッファ p へ受信した文字数 失敗 : -1
【説明】	色制御データのバージョン情報を文字列バッファへ受信します。
【文例】	<div>< Visual C ></div> <pre>char rbuf[256]; if(GetColorDataVersionRes (DS820, (LPSTR)rbuf, 1) > 0){ // 600dpi のバージョンを取得 // 処理 }</pre> <div>< VB.NET ></div> <pre>Dim s As String = New String("", 255) Dim i As Integer i = GetColorDataVersionRes(DS820, s, 0) ' 300dpi のバージョンを取得 If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"</pre>

色制御データバージョン取得コマンド(種別、メディア指定)

【書式】	<code>long GetColorDataVersionResEX(long IPortNum, LPSTR p, DWORD dwType, DWORD dwMedia);</code>
【引数】	<p><code>IPortNum</code> : ポート番号</p> <p><code>p</code> : 受信バッファへのポインタ</p> <p><code>dwType</code> : 300dpi 色制御バージョンデータを取得 (0) 600dpi 色制御バージョンデータを取得 (1) 低速印画色制御バージョンデータを取得 (2) 高濃度印画色制御バージョンデータを取得 (3)</p> <p><code>dwMedia</code> : SD メディア (1) PP メディア (3)</p>
【戻り】	<p>成功 : バッファ p へ受信した文字数</p> <p>失敗 : -1</p>
【説明】	色制御データのバージョン情報を文字列バッファへ受信します。
【文例】	<pre>< Visual C > char rbuf[256]; if(GetColorDataVersionResEX(DS820, (LPSTR)rbuf, 1, 3) > 0){ // 600dpi のバージョンを取得 // 処理 } < VB.NET > Dim s As String = New String("", 255) Dim i As Integer i = GetColorDataVersionResEX(DS820, s, 0, 3) ' 300dpi のバージョンを取得 If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"</pre>

色制御データチェックサム取得コマンド

【書式】	<code>long GetColorDataChecksum(long IPortNum, LPSTR p);</code> <code>long CvGetColorDataChecksum(long IPortNum, LPSTR p);</code>
【引数】	<p><code>IPortNum</code> : ポート番号</p> <p><code>p</code> : 受信バッファへのポインタ</p>
【戻り】	<p>成功 : バッファ p へ受信した文字数</p> <p>失敗 : -1</p>
【説明】	色制御データのチェックサムを文字列バッファへ受信します。
【文例】	<pre>< Visual C > char rbuf[256]; if(GetColorDataChecksum(DS820, (LPSTR)rbuf) > 0){ // 処理 } < VB.NET > Dim s As String = New String("", 255) Dim i As Integer i = GetColorDataChecksum(DS820, s) If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"</pre>

色制御データチェックサム取得コマンド(種別指定)

【書式】 `long GetColorDataChecksumRes(long lPortNum, LPSTR p , DWORD dwType);`

【引数】 `lPortNum` : ポート番号
`p` : 受信バッファへのポインタ
`dwType` : 300dpi 色制御データチェックサムを取得 (0)
 600dpi 色制御データチェックサムを取得 (1)
 低速印画色制御データチェックサムを取得 (2)
 高濃度印画色制御データチェックサムを取得 (3)

色制御データチェックサムは、装着されたメディア種別により異なります。

【戻り】 成功 : バッファ `p` へ受信した文字数
 失敗 : -1

【説明】 色制御データのチェックサムを文字列バッファへ受信します。

【文例】 `< Visual C >`
`char rbuf[256];`
`if(GetColorDataChecksumRes (DS820, (LPSTR)rbuf, 0) > 0){ // 300dpi のチェックサムを取得`
`// 処理`
`}`

`< VB.NET >`
`Dim s As String = New String("", 255)`
`Dim i As Integer`
`i = GetColorDataChecksumRes(DS820, s, 1) ' 600dpi のチェックサムを取得`
`If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"`

色制御データチェックサム取得コマンド(種別、メディア指定)

【書式】 `long GetColorDataChecksumResEx(long lPortNum, LPSTR p , DWORD dwType、DWORD dwMedia);`

【引数】 `lPortNum` : ポート番号
`p` : 受信バッファへのポインタ
`dwType` : 300dpi 色制御データチェックサムを取得 (0)
 600dpi 色制御データチェックサムを取得 (1)
 低速印画色制御データチェックサムを取得 (2)
 高濃度印画色制御データチェックサムを取得 (3)

`dwMedia` : SD メディア (1)
 PP メディア (3)

【戻り】 成功 : バッファ `p` へ受信した文字数
 失敗 : -1

【説明】 色制御データのチェックサムを文字列バッファへ受信します。

【文例】

```
< Visual C >
char rbuf[256];
if(GetColorDataChecksumResEX( DS820, (LPSTR)rbuf, 0, 3 ) > 0 ){
    // 処理
}

< VB.NET >
Dim s As String = New String("", 255)
Dim i As Integer
i = GetColorDataChecksumResEX( DS820, s, 1, 3 )
If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"
```


カッター制御コマンド

【書式】 BOOL SetCutterMode(long lPortNum, DWORD dwMode);

【引数】 lPortNum : ポート番号
dwMode : カッターのモード選択

【戻り】 成功 : True
失敗 : False

【説明】 カッター動作を指定します。
カッターの動作指定は DS820Stat.h (VB は DS820Stat.vb) にマクロ定義されています。
シンボルの意味は下記の通りです。

CUTTER_MODE_STANDARD	通常のカット動作
CUTTER_MODE_NONSCRAP	カットくず無し指定
CUTTER_MODE_2INCHCUT	2 インチカット指定
CUTTER_MODE_PANORAMA	パノラマ印画動作 (画像間余白あり)

本設定の初期値は、“CUTTER_MODE_STANDARD” です。
設定はプリンターの電源を OFF すると初期値に戻ります。

【注意】 カッター制御コマンドは、画像データの送信前に動作指定します。コマンドは 1 画像に対して
1 回有効です。プリンターは印画⇒指定カット動作後、通常のカット動作に戻ります。
2 インチカット指定は用紙サイズ (8x4)、(8x6)、(8x8)、(8x10) または (8x12) に有効です。
パノラマ印画は以下のサイズに対応しています。
8x10 ⇒ 8x20 (2 枚印画)、8x30 (3 枚印画)
8x12 ⇒ 8x24 (2 枚印画)、8x36 (3 枚印画)
A4 ⇒ A4x2 (2 枚印画)、A4x3 (3 枚印画)

以下の場合には本コマンドによる指定は無効となります。
・フルカッターセットアップコマンド又はフルカッターセットアップ拡張コマンドが使用された
場合。

【文例】 < Visual C >
if (SetCutterMode(DS820, (DWORD)CUTTER_MODE_NONSCRAP) == True) {
 // 設定成功
}

 < VB.NET >
if SetCutterMode(DS820, CUTTER_MODE_NONSCRAP) <> False
 ' 設定成功
End If

フルカッターセットアップコマンド

【書式】 B00L SetFullCutterSetup(long lPortNum, long lSize1, long lSize2,
long lSize3, long lSize4, long lSize5, long lSize6);

【引数】 lPortNum : ポート番号
lSize1 : 画像 1 カットサイズ (値域 : 0, 20～)
lSize2 : 画像 2 カットサイズ (値域 : 0, 20～)
lSize3 : 画像 3 カットサイズ (値域 : 0, 20～)
lSize4 : 画像 4 カットサイズ (値域 : 0, 20～)
lSize5 : 画像 5 カットサイズ (値域 : 0, 20～)
lSize6 : 画像 6 カットサイズ (値域 : 0, 20～)

【戻り】 成功 : True
失敗 : False

【説明】 下表の用紙サイズのときに、1 枚から最大カット枚数まで、指定された各サイズにカットします。カットサイズは、合計したサイズが最大サイズを超えない範囲で、2 インチから最大サイズまで 0.1 インチ単位で設定できます。

画像 1 カットサイズ設定を '0' に設定された場合は、通常サイズで印画します。その場合、画像 2～4 のカットサイズ設定は無視します。

カット枚数が 1 枚の場合は、画像 2～6 カットサイズ設定に '0' を設定してください。
カット枚数が 2 枚の場合は、画像 3～6 カットサイズ設定に '0' を設定してください。
カット枚数が 3 枚の場合は、画像 4～6 カットサイズ設定に '0' を設定してください。
カット枚数が 4 枚の場合は、画像 5～6 カットサイズ設定に '0' を設定してください。
カット枚数が 5 枚の場合は、画像 6 カットサイズ設定に '0' を設定してください。

各カットサイズ設定で、上記設定範囲外の値が設定された場合は False を返します。

■用紙サイズとカットサイズ設定範囲

用紙サイズ	最大 カット枚数	最大 サイズ	画像	カットサイズ設定範囲
8x10	5 枚	10 インチ	1	20～100
			2	20～80
			3	20～60
			4	20～40
			5	20
8x12	6 枚	12 インチ	1	20～120
			2	20～100
			3	20～80
			4	20～60
			5	20～40
A4	5 枚	11.7 インチ	6	20
			1	20～117
			2	20～97
			3	20～77
			4	20～57
			5	20～37

【注意】 本コマンドは印画開始コマンド送信前に送信してください。

本コマンドは 1 画像に対して 1 回有効です。プリンターは 1 画像毎に通常サイズ印画動作に戻ります。

本コマンドで設定したカットサイズの合計が、プリンターに転送された画像サイズより大きい場合、本コマンドは無効となり、プリンターに転送した用紙サイズ(通常サイズ)で印画します。

本コマンドを使用した場合以下の指定は無効となり、本コマンドを優先します。

- ・カッター制御コマンド

フルカッターセットアップ拡張コマンドと本コマンドを併用した場合は後に使用した設定を優先します。

【文例】

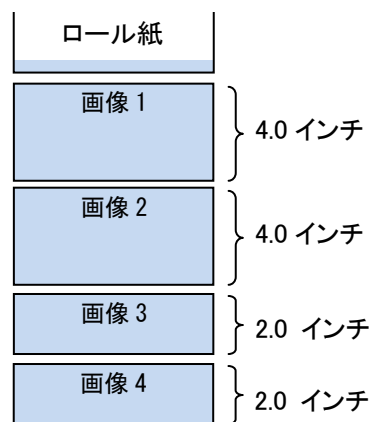
< Visual C >

```
if((SetFullCutterSetup( DS820, 40, 40, 20, 20, 0, 0 )) != False){
    // 設定成功
}
```

< Visual Basic >

```
If SetFullCutterSetup(DS820, 40, 40, 20, 20, 0, 0) <> False Then
    ' 設定成功
End If
```

* 上記の文例を使用した場合の印刷結果は以下になります。



フルカッターセットアップ拡張コマンド

- 【書式】 B00L SetFullCutterSetupEX(long lPortNum, long lCutSize1, long lCutSize2, long lCutSize3, long lCutSize4, long lCutSize5, long lCutSize6, long lDustSize);
- 【引数】
- | | | |
|------------|---------------|---------------|
| lPortNum: | ポート番号 | |
| lCutSize1: | 画像 1 カットサイズ設定 | 設定値 : 0、20～ |
| lCutSize2: | 画像 2 カットサイズ設定 | 設定値 : 0、20～ |
| lCutSize3: | 画像 3 カットサイズ設定 | 設定値 : 0、20～ |
| lCutSize4: | 画像 4 カットサイズ設定 | 設定値 : 0、20～ |
| lCutSize5: | 画像 5 カットサイズ設定 | 設定値 : 0、20～ |
| lCutSize6: | 画像 6 カットサイズ設定 | 設定値 : 0、20 |
| lDustSize: | 中間層カットサイズ設定 | 設定値 : 0、12～22 |
- 【戻り】
- | | |
|-----|-------|
| 成功: | True |
| 失敗: | False |
- 【説明】
- 中間層カットサイズの設定が可能なフルカッターセットアップの拡張コマンドです。
 中間層カットサイズは 0.12～0.22 インチの範囲で、0.01 インチ単位で設定できます。
 中間層カットサイズを 0 以外の有効な範囲で設定する場合、下記の式のカットサイズ合計が用紙の最大サイズを超えないように設定してください。
- $$\text{カットサイズ合計} = (\text{各画像のカットサイズ合計}) + ((\text{中間層カットサイズ}/10) \times (\text{画像枚数}-1))$$
- 例) 用紙サイズ 8x12 インチ (120)、画像 (2 インチ (設定値 20) で 5 枚、中間層を 0.22 インチ (設定値 22) に設定した場合の計算
- $$(20+20+20+20+20+0) + ((22/10) \times (5-1)) = 108.8 \leq 120$$
- 中間層カットサイズを 0 にした場合の各画像の設定可能範囲は、「フルカッターセットアップ」の表、用紙サイズとカットサイズ設定範囲を参照してください。
- 各カットサイズ設定で、上記設定範囲外の値を設定した場合、“False” が戻ります。
- 【注意】
- 本コマンドで設定したカットサイズの合計が、プリンターに転送された画像サイズより大きい場合、本コマンドは無効となり、プリンターに転送した用紙サイズ (通常サイズ) で印画します。
- 本コマンドを使用した場合以下の指定は無効となり、本コマンドを優先します。
- ・カッター制御コマンド
- フルカッターセットアップコマンドと本コマンドを併用した場合は後に使用した設定を優先します。

【文例】

< Visual C >

```

if( SetFullCutterSetupEX( DS820, 20, 20, 20, 20, 20, 0, 12 ) == True) {
    //設定成功
}

```

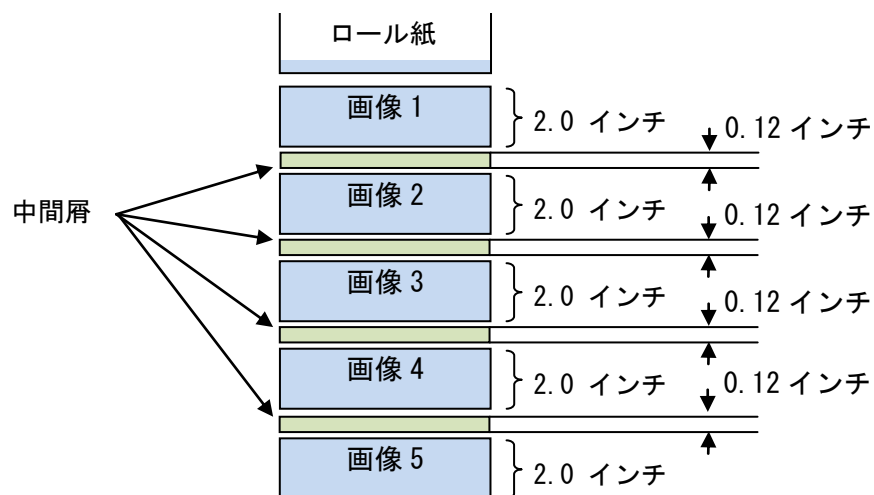
< VB.NET >

```

Dim Result As Long
Result = SetFullCutterSetupEX (DS820, 20, 20, 20, 20, 20, 0, 12)
If Result <> False Then
    ' 設定成功
End If

```

* 上記の文例を使用した場合の印刷結果は以下になります。



プリント速度指定コマンド

【書式】	B00L SetPrintSpeed(long lPortNum, long lPrintSpeed);
【引数】	lPortNum: ポート番号 lPrintSpeed: 印画速度 0: 通常速度 (デフォルト) 1: 予約 2: 予約 3: 高濃度印画速度
【戻り】	成功: True 失敗: False
【説明】	指定したプリント速度で印刷を行います。 高濃度の印画を行いたい場合に指定します。
【注意】	本コマンドは1画像毎に指定します。 プリンターは1画像印刷後、デフォルト設定に戻ります。
【文例】	<pre> < Visual C > if(SetPrintSpeed (DS820, 3)){ // 高濃度印画速度 // 設定成功 } < VB.NET > if SetPrintSpeed (DS820, 0) Then // 通常印画速度 ' 設定成功 End if </pre>

スタンバイモード移行時間設定

【書式】	B00L	SetStandbyTime(long lPortNum, long lTime);
【引数】	lPortNum : lTime :	ポート番号 スタンバイモード移行時間 設定値 : 0、1~99
【戻り】	成功 : 失敗 :	True False
【説明】	<p>スタンバイモードに移行するまでの時間を1分単位で設定します。 設定されている時間だけアイドル状態が継続すると、スタンバイモードに移行します。</p> <p>移行時間に '0' が設定された場合は、スタンバイ機能は無効となりスタンバイモードに入りません。 移行時間に上記設定範囲外の値を設定した場合、False を返します。</p> <p>本設定の初期値は、10 分 です。 設定はプリンターに記憶され、電源を OFF しても保持されます。</p>	
【注意】	10分より短い時間に変更する場合は、プリンター製品仕様書の『消費電力』仕様に記載されている内容を確認の上、ご使用ください。	
【文例】	<p>< Visual C > <pre>if((SetStandbyTime(DS820, 0)) == FALSE) { // 設定失敗 }</pre> </p> <p>< VB.NET > <pre>Dim Result As Long Result = SetStandbyTime(DS820, 99) If Result = FALSE Then GoTo Error</pre> </p>	

スタンバイモード移行時間設定値の取得

【書式】	long	GetStandbyTime(long lPortNum);
【引数】	lPortNum :	ポート番号
【戻り】	成功 : 失敗 :	スタンバイモード移行時間 -1
【説明】	スタンバイモード移行時間(分単位)を返します。	
【文例】	<p>< Visual C > <pre>long time; if((time = GetStandbyTime(DS820)) >= 0) { // time にスタンバイモード移行時間 }</pre> </p> <p>< VB.NET > <pre>Dim time As Integer time = GetStandbyTime(DS820)</pre> </p>	

メディアエンド保持モード設定

【書式】	B00L	SetEndKeepMode(long lPortNum, long lMode);
【引数】	lPortNum: lMode:	ポート番号 保持モード設定 設定値 : 0、1
【戻り】	成功 : 失敗 :	True False
【説明】	<p>メディアエンド発生後、そのままの状態、カバー開閉や電源を再投入したときに、「メディアエンドを保持する/しない」の動作モードを設定します。</p> <p>メディアエンド（リボンエンド、ペーパーエンド）とメディアエラー（リボンエラー、ペーパージャム）が対象となります。</p> <p>モード設定に '0' が設定された場合は、メディアエンド非保持モードになります。 モード設定に '1' が設定された場合は、メディアエンド保持モードになります。 モード設定に上記設定範囲外の値が設定された場合、False を返します。</p> <p>本設定の工場出荷時設定値は、“メディアエンドを保持する(1)” です。 設定はプリンターに記憶され、電源を OFF しても保持されます。</p>	
【文例】	<pre>< Visual C > if((SetEndKeepMode(DS820, 0)) == FALSE){ // 設定失敗 } < VB.NET > Dim Result As Integer Result = SetEndKeepMode (DS820, 1) If Result = FALSE Then GoTo Error</pre>	

メディアエンド保持モード設定値の取得

【書式】	long	GetEndKeepMode (long lPortNum);
【引数】	lPortNum :	ポート番号
【戻り】	成功 : 失敗 :	保持モード設定 -1
【説明】	メディアエンド保持モード設定値を 0(保持モード無効)か 1(保持モード有効)で返します。	
【文例】	<pre>< Visual C > long mode; if((mode = GetEndKeepMode(DS820)) >= 0){ // mode にモード設定 } < VB.NET > Dim mode As Integer mode = GetEndKeepMode(DS820)</pre>	

パノラマ印画開始可能チェック

- 【書式】 long GetPanoramaPrintable(long lPortNum);
- 【引数】 lPortNum : ポート番号
- 【戻り】 成功 : パノラマ印画開始可能かどうかのプリンターステータス
 失敗 : STATUS_COMM_ERROR
- 【説明】 パノラマ印画開始可能かどうか、プリンターの状態を返します。
 各ステータスの値は、DS820Stat.h (VB は DS820Stat.vb) にマクロ定義されています。
 シンボルの意味は以下の通りです。

戻り値	値[HEX]	状態
STATUS_PRINTABLE	0x00000000	パノラマ印画開始可能
STATUS_HIGHTEMP_HEAD	0x00000001	ヘッド高温
STATUS_HIGHHUMIDITY *1	0x00000100	高湿
STATUS_LOWTEMP_MEDIA *2	0x00000020	メディア低温
STATUS_OTHER_STATE	0x00010000	その他の状態 (印刷中、クーリング中、エラー等) プリンターステータスの取得コマンドで内容が確認できます。
STATUS_COMM_ERROR	0xFFFFFFFF	通信エラー

*1: DP-DS820 ファームウェアバージョン 01.01 以降対応

*2: DP-DS820 ファームウェアバージョン 01.03 以降対応

- 【注意】
- ・ヘッド高温状態の場合は、パノラマ印画可能状態になるのを待たずに、パノラマ印画データを送信してください。(通常印画と同様にクーリングを開始し、ヘッド温度低下後に印画が開始されます。)
 - ・湿度が高い状況での連続パノラマ印画は、画質が低下する可能性があるため、推奨いたしません。
 - ・メディア温度が低い状況での連続パノラマ印画は、行わないでください。(画像によってはジャム等の不具合が発生する可能性があります。)

(余白ありパノラマ印画の詳細については「Panorama_withSpace_J.doc」を参照して下さい。)

(連続パノラマ印画については連続パノラマ SDK を参照してください。)

【文例】

< Visual C >

```

long    stat;
stat = GetPanoramaPrintable( DS820 );
if(stat == STATUS_LOWTEMP_MEDIA) {
    // 連続パノラマ印画を実行しないでください
}else if(stat == STATUS_HIGHHUMIDITY) {
    // 連続パノラマ印画の実行は推奨いたしません
}else if((stat == STATUS_PRINTABLE) || (stat == STATUS_HIGHTEMP_HEAD)) {
    // パノラマ印画を開始してください
}else{
    // プリンターステータスの確認等
}

```

< VB.NET >

```

Dim stat As Integer
stat = GetPanoramaPrintable( DS820 )
If stat = STATUS_LOWTEMP_MEDIA Then
    ' 連続パノラマ印画を実行しないでください
Else If stat = STATUS_HIGHHUMIDITY Then
    ' 連続パノラマ印画の実行は推奨いたしません
Else If (stat = STATUS_PRINTABLE) Or (stat = STATUS_HIGHTEMP_HEAD) Then
    ' パノラマ印画を開始してください
Else
    ' プリンターステータスの確認等
End If

```

連続パノラマ印画設定

【書式】	BOOL SetContPanorama(long lPortNum, DWORD dwContinuous, DWORD dwOverlap);		
【引数】	lPortNum :	ポート番号	
	dwContinuous :	連続パノラマ印画指定	
	dwOverlap :	重ね合わせ幅	0-200 (推奨値 200)
【戻り】	成功 :	True	
	失敗 :	False	
【説明】	連続パノラマ印画動作を設定します。 (連続パノラマ印画については連続パノラマ SDK を参照してください)		
	連続パノラマ印画指定の値は、DS820Stat.h (VB は DS820Stat.vb) にマクロ定義されています。 シンボルの意味は以下の通りです。		
	CONT_PANORAMA_PRINT :	連続パノラマ印画指定	
	CONT_PANORAMA_LAST :	連続パノラマ印画最終画像、または通常印画 (連続パノラマキャンセル)	
	重ね合わせ幅は、0.00~2.00 インチの範囲で、0.01 単位で設定できます。		
【注意】	連続パノラマ印画指定は 1,2 ページ目のみ設定が有効になります。3 ページ目に連続パノラマ指定を行った場合は、設定が無視され用紙のカットを行います。		
	重ね合わせ幅は、連続パノラマの 1 枚目の印画で設定した値が 2 ページ目以降の印画にも適用され、2 ページ目以降に異なる値を設定した場合は無効となります。 また、重ね合わせ幅を推奨値よりも狭くした場合、重ね合わせ部分の印画品質が低下します。		
【文例】	< Visual C >		
	if(SetContPanorama(DS820, CONT_PANORAMA_MODE, 200)){ // 連続パノラマ印画設定 // 設定成功 }		
	< VB.NET >		
	If SetContPanorama(DS820, CONT_PANORAMA_MODE, 200) Then ' 連続パノラマ印画設定 ' 設定成功 End If		

オーバーコート仕上げ制御コマンド

【書式】 B00L SetOvercoatFinish(long lPortNum, DWORD ovcoat);

【引数】 lPortNum : ポート番号
ovcoat : オーバーコート仕上げの種類選択

【戻り】 成功 : True
失敗 : False

【説明】 プリンターのオーバーコート仕上げを制御します。
オーバーコート仕上げの種類は DS820Stat.h (VB は DS820Stat.vb) にマクロ定義されています。
シンボルの意味は下記の通りです。

OVERCOAT_FINISH_GLOSSY	光沢
OVERCOAT_FINISH_MATTE1	マット 1
OVERCOAT_FINISH_MATTE2	マット 2 (予約)
OVERCOAT_FINISH_MATTE3	マット 3 (予約)
OVERCOAT_FINISH_FINEMATTE	ファインマット
OVERCOAT_FINISH_LUSTER	ラスター
OVERCOAT_FINISH_PMATTE11	部分マット (マット)
OVERCOAT_FINISH_PMATTE12	部分マット (ファインマット)
OVERCOAT_FINISH_PMATTE13	部分マット (ラスター)

本設定の初期値は、“OVERCOAT_FINISH_GLOSSY” です。

【注意】 本コマンドは印画開始コマンド送信前に送信して下さい。
本コマンドは 1 画像に対して 1 回有効です。指定オーバーコートで動作後、光沢仕上げに戻ります。

以下の場合、本コマンドによる指定は無効となります。
・プリンタードライバのオーバーコート仕上げ設定が『光沢』以外の場合。

【文例】

```
< Visual C >
if ( SetOvercoatFinish( DS820, OVERCOAT_FINISH_MATTE ) == True )
{
    // 設定成功
}

< VB.NET >
If SetOvercoatFinish( DS820, OVERCOAT_FINISH_MATTE ) <> False Then
    ' 設定成功
End If
```

ガンマテーブルの設定

【書式】 B00L SetGammaTable(long lPortNum, LPBYTE lpGammaTable, long lDataLen);

【引数】 lPortNum : ポート番号
lpGammaTable : ガンマテーブルデータの先頭アドレス。
lDataLen : ガンマテーブルデータのサイズ。(Byte 単位)

【戻り】 成功 : TRUE
失敗 : FALSE

【説明】 ガンマテーブルデータを設定します。
ガンマテーブルデータの形式については、下表をご参照ください。

種別	項目	説明
ヘッダー (8byte)	Data unit size	0002h 固定
	Row	0100h 固定
	Plane	0004h 固定
	Checksum	xxxxh(拡張データとガンマテーブルデータのサム値)
拡張データ (8byte)	Mode	0000h 固定
	Rev1	0000h 固定
	Rsv2	0000h 固定
	Rsv3	0000h 固定
ガンマテーブルデータ (2048byte)	Y[0]...Y[255] M[0]...M[255] C[0]...C[255] OP[0]...OP[255]	0000h~FFFFh

【注意】 ガンマテーブルデータは、プリンターの状態が電源 OFF、もしくは、スタンバイモードに移行すると失われるため、印刷前に送信する必要があります。

【文例】

```
< Visual C >
BYTE tableData[2064];
...
if(SetGammaTable( DS820, tableData, 2064 ) > 0 ){
    // 成功
}

< VB.NET >
Dim tableData(2064) As Byte
...
if SetGammaTable( DS820, tableData, 2064 ) > 0 Then
    ' 成功
End If
```

ガンマテーブルチェックサム取得

【書式】	<code>long GetGammaTableChecksum(long lPortNum, LPSTR p);</code>
【引数】	lPortNum : ポート番号 lpRcvBuf : 受信バッファのポインタ。
【戻り】	成功 : バッファ p へ受信した文字数 失敗 : -1
【説明】	ガンマテーブルのチェックサム値を取得します。 但し、プリンターにガンマテーブルデータが設定されていない場合は、4文字のスペース文字列 “□□□□” (CR<0Dh>終端、4バイトパディング) を受信します。
【文例】	<pre> < Visual C > char sum[256]; if(GetGammaTableChecksum(DS820, sum) > 0){ // 成功 } < VB.NET > Dim sum As String = New String(" ", 256) if GetGammaTableChecksum(DS820, sum) > 0 Then ' 成功 End If </pre>

データテーブルのクリア

【書式】	<code>BOOL SetDataTableClear(long lPortNum);</code>
【引数】	lPortNum : ポート番号
【戻り】	成功 : TRUE 失敗 : FALSE
【説明】	プリンターに設定したガンマテーブルの値をクリアします。
【文例】	<pre> < Visual C > if(SetDataTableClear(DS820)){ // 成功 } < VB.NET > if SetDataTableClear(DS820) Then ' 成功 End If </pre>

汎用設定系コマンド

【書式】	<pre> BOOL SetCommand(long lPortNum, LPSTR lpCmd, DWORD dwCmdLen); BOOL CvSetCommand(long lPortNum, LPSTR lpCmd, DWORD dwCmdLen); </pre>
【引数】	<pre> lPortNum : ポート番号 lpCmd : コマンドが格納されたバッファへのポインタ dwCmdLen : コマンドの文字数 </pre>
【戻り】	<pre> 成功 : True 失敗 : False </pre>
【説明】	<p>プリンターへコマンドを送信します。</p> <p><VB.NET ラッパー関数について> VB.NET を使用する場合、DS820Stat.vb ファイルにラッパー関数として記述して有ります。 VB プロジェクトに DS820Stat.vb ファイルを追加してください。</p>

汎用取得系コマンド

【書式】	<pre> long GetCommandEX(long lPortNum, LPSTR lpCmd, DWORD dwCmdLen, LPSTR lpRetBuff, DWORD dwRetBuffSize); long CvGetCommandEX(long lPortNum, LPSTR lpCmd, DWORD dwCmdLen, LPSTR lpRetBuff, DWORD dwRetBuffSize); </pre>
【引数】	<pre> lPortNum : ポート番号 lpCmd : コマンドが格納されたバッファへのポインタ dwCmdLen : コマンド文字数 lpRetBuff : 受信データを格納するバッファへのポインタ dwRetBuffSize : 受信バッファの許容サイズ </pre>
【戻り】	<pre> 成功 : 受信バイト数 (RetBuff へ受信データ) 失敗 : -1 </pre>
【説明】	<p>プリンターへコマンドを送信後、受信データをバッファへ格納します。</p> <p><VB.NET ラッパー関数について> VB.NET を使用する場合、DS820Stat.vb ファイルにラッパー関数として記述して有ります。 VB プロジェクトに DS820Stat.vb ファイルを追加してください。</p>

サンプルプログラム説明

ステータス API サンプルプログラムは、プリンタステータス API (DS820Stat.dll、DS820Stat.vb) を使用してプリンター情報を取得する、VB.NET で作成したサンプルプログラムです。以下に使用方法を説明いたします。

サンプルプログラムを実行すると下記のウィンドウが表示されます。各ボタンをクリックすることにより、それぞれの情報を取得することができます。以下に GetCounterL ボタンをクリックした時の例を示します。

GetCounterL ボタンをクリックすると下記プログラムが実行され、ステータス API 関数の GetCounterL() により、プリンターからライフカウンタの値を取得し、ウィンドウ内下部の出力部に表示されます。

```
'***** Get life counter
Private Sub Command7_Click()
    Dim c As Long

    c = GetCounterL(DS820)
    If c >= 0 Then Text1.Text = Str(c) Else Text1.Text = "ERROR!"
End Sub
```

