

昇華型カードプリンタ CX-120

ソフト開発用技術資料

Rev.006 発行日 平成 26 年 4 月 15 日

株式会社 JVC ケンウッド

- 本書のご利用には、CX-120 スタートアップガイドに記載の使用許諾契約書への同意が必要です。
- 本資料を元に開発したものに對し、当社は如何なる責任も負いません。
- 本資料は改善のために、予告なく変更する場合があります。
- MicrosoftおよびWindowsは米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。Microsoft® Windows® 8 operating system 日本語版、Microsoft® Windows® 7 operating system 日本語版、Microsoft® Windows Vista™ operating system 日本語版、Microsoft® Windows® XP operating system 日本語版、Microsoft® Windows® 2000 operating system 日本語版の表記について本書中では、上記各オペレーティングシステムをそれぞれ、Windows 8、Windows 7、Windows Vista、Windows XP、Windows 2000 と表記しています。

【改訂履歴】

2006.2.15 初版	
2007.6.28 第 2 版	(1) 関数CXCMD_LoadCard()及びCXCMD_MoveCard()の引数iDestの値に誤りがあったため修正しました。
2007.7.17 第 3 版	(1) 「5.11 プリンタの設定値取得/変更」にLog Select関数、Mode Select関数を追加しました。
2007.12.28 第 4 版	(1) 「2. 動作環境」にWindows Vistaを追加しました。 (2) 「6.3 ICエンコーディングの組込み方法」で、ICエンコードDLLの返り値の処理を変更しました。
2010.6.8 第 5 版	(1) 「2. 動作環境」に、Windows 7 32bit/64bitを追加しました。 (2) 「3. システム構成」「5.1 制御関数の形態」に、64bit版用DLLのファイル名を追加しました。
2014.4.15 第 6 版	(1) 「2. 動作環境」にWindows 8 32bit/64bitを追加しました。Windows 7、Vista、XPにService Packのバージョンを追加しました。

<目次>

1. 概要	4
2. 動作環境	4
3. システム構成	5
4. カードプリンタについて	6
4. 1 コマンド体系	6
4. 2 カード位置	6
5. プリンタ制御関数	7
5. 1 制御関数の形態	7
5. 2 返り値について	7
5. 3 カード発行処理の流れ	8
5. 4 プリンタの検索	9
5. 5 プリンタのステータス取得	9
5. 6 印刷	10
5. 7 カード搬送と排出	13
5. 8 磁気エンコーディング	15
5. 9 IC エンコーディング	16
5. 10 プリンタの初期化	17
5. 11 プリンタの設定値取得/変更	18
6. プリンタドライバでのエンコーディング方法	29
6. 1 インラインエンコーディング	29
6. 2 ExtEscape()によるエンコーディング	30
6. 3 IC エンコーディングの組み込み方法	31
7. プリンタドライバの設定方法	33
7. 1 ExtEscape()の使用方法	33
7. 2 設定の変更	34
7. 3 設定の参照	34
7. 4 プログラミングサンプル	35
7. 5 パラメーター一覧	37
7. 6 エラーコード一覧	39

付録1 エラーコード表

付録2 磁気データコード表

1. 概要

この資料はプリンタ用ソフトウェアを技術的な側面から説明します。プリンタドライバをインストールすると、プリンタドライバが使用するプリンタ制御用の DLL(以降「プリンタ制御関数」と記述します)も同時にインストールされます。通常、カードへの印刷やエンコーディングはプリンタドライバを使用して行いますが、プリンタドライバを使用せずプリンタ制御関数を直接使用して印刷やエンコーディングを行うこともできます。

この資料ではプリンタ制御関数の使用方法や、ドライバを使用した特殊なプログラミング方法(IC エンコーディングや磁気エンコーディングなど)について説明します。

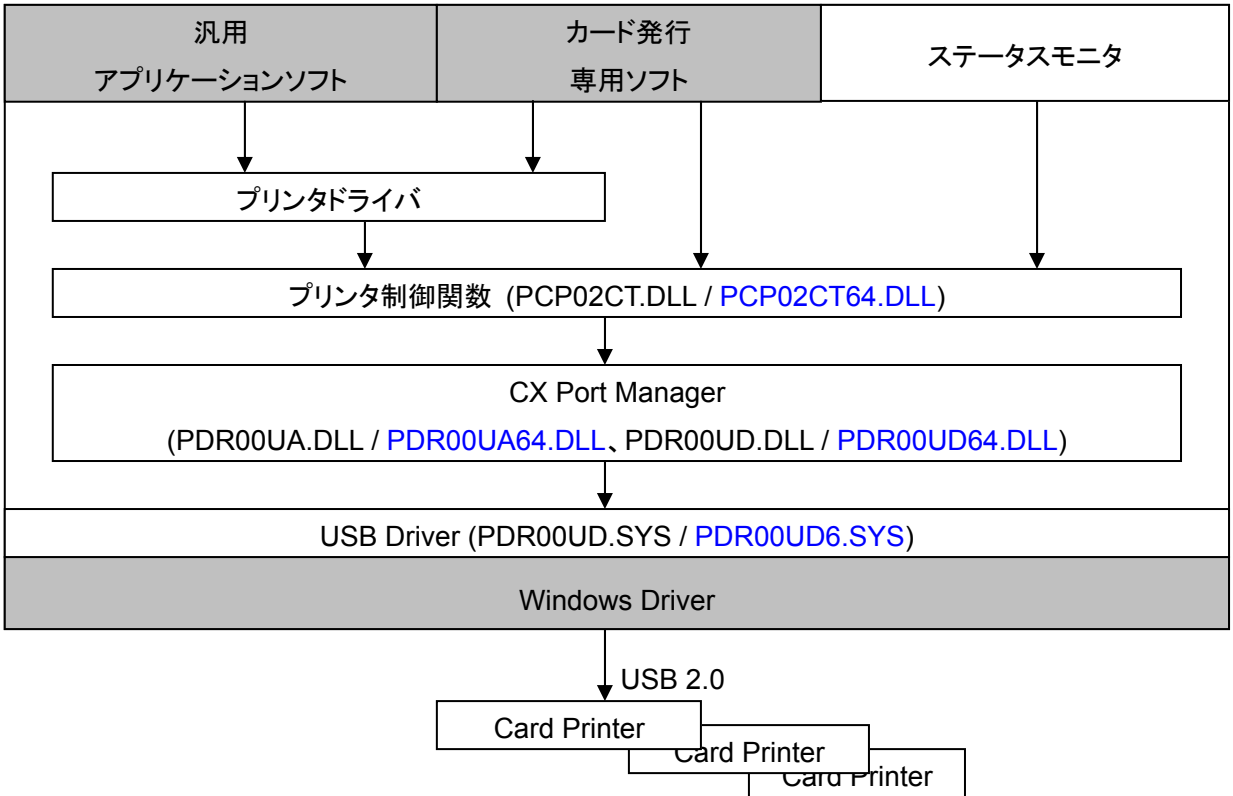
2. 動作環境

プリンタ用ソフトウェアは以下の環境で動作します。

項目	内容	備考
OS	Windows 8 32bit 版/64bit 版 Windows 7 (初版、Service Pack 1) 32bit 版/64bit 版 Windows Vista (Service Pack 1/2) 32bit 版 Windows XP (Service Pack 2/3) 32bit 版 Windows 2000 Professional (Service Pack 4) 32bit 版	
周辺機器	カードプリンタ CX-120 シリーズ	USB 接続

3. システム構成

以下の図はプリンタ用ソフトウェアの構成です。CD-ROM のソフトウェアを全てインストールすると、白枠部のソフトウェアがインストールされます。汎用アプリケーションソフトはプリンタドライバを使用しますが、カード発行専用ソフトであればプリンタ制御関数を使用して、印刷やエンコーディングを行うこともできます。



注:ファイル名は、(32bit 版 / 64bit 版)の順番で記載しています。

4. カードプリンタについて

4. 1 コマンド体系

カードプリンタのコマンド体系は SCSI 準拠で、スロット番号と ID でカードプリンタを特定します。この機能で、最大7台までのカードプリンタを制御することができます。カードプリンタに命令を発行する場合にはスロット番号と ID を指定します。スロット番号はシステムの環境に依存する値で、システムが動作中は固定です。ID はカードプリンタで設定するユニット番号から1を引いた値です。

4. 2 カード位置

カードが適切な位置にないと、カードプリンタへの命令が失敗する場合があります。例えば、磁気エンコーディング命令はカードが磁気エンコーダ位置にあることが必要条件です。カード位置を取得するために、「リード・ポジション関数」が準備されています。

カードプリンタのカード位置

カード位置	説明
印刷待機位置	印刷を開始するときのカード位置です。
接触 IC エンコーダ位置	接触 IC エンコーダの位置です
非接触 IC エンコーダ位置	非接触 IC エンコーダの位置です
磁気エンコーダ位置	磁気エンコーダの位置です

5. プリンタ制御関数

5. 1 制御関数の形態

プリンタ制御関数は Windows の DLL 形式で提供します。ドライバをインストールしたとき、プリンタ制御関数は同時にインストールされます。

名称	対応 OS	ファイル名
カードプリンタ制御 DLL	32bit OS	PCP02CT.DLL
	64bit OS	PCP02CT64.DLL

注： 64bit OS で 32bit アプリケーションを利用する場合は、32bit 版の PCP02CT.DLL を使用してください。

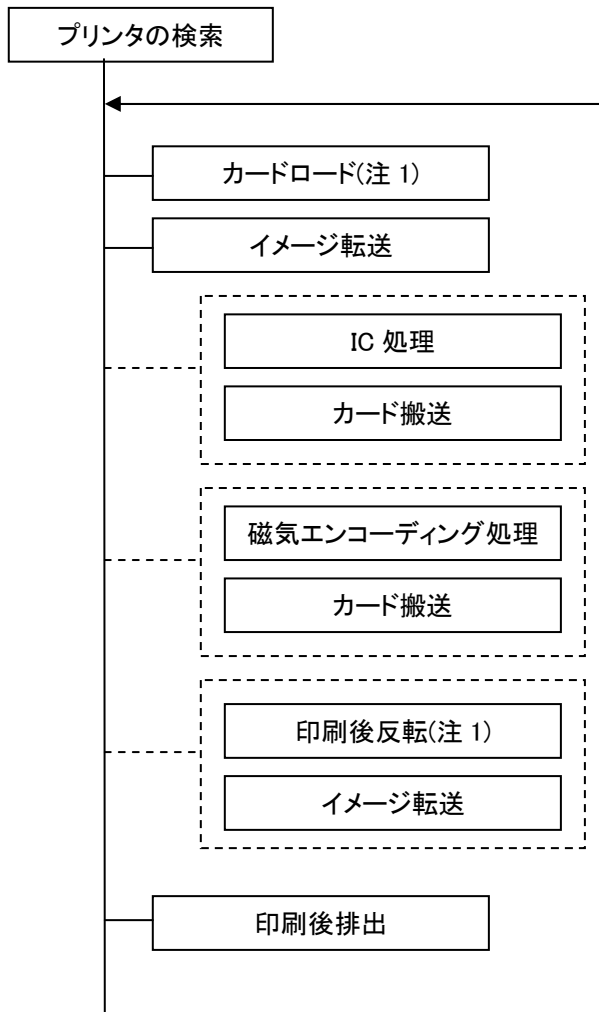
5. 2 返り値について

返り値について特に明示がない場合、以下の規則で返します。

- ① 0: 処理が成功したことを意味します
- ② 正值: プリンタ制御関数、あるいはプリンタが一時的に命令を受け付けることができなかったことを意味します。一定時間後に再試行してください。
- ③ 負値: 障害発生を示します。値の意味は「付録1 エラーコード表」を参照してください。

5.3 カード発行処理の流れ

プリンタ制御関数は、スロット番号と ID でプリンタを特定します。このため、最初にプリンタを検索し、プリンタを特定するためのスロット番号と ID を取得します。プリンタの設定を変えなければ、この ID が変わることはありません。プリンタのコマンドは、SCSI に準拠しています。以下は、標準的なカード発行処理の流れを示しています。



<注意>

- ① イメージ転送とメカ動作を並行させるため、これらの命令を発行する場合は Immediate フラグをセット(1)にしてください。
- ② プリンタは、カードが適切な位置にあることを前提に処理します。このため、「カードロード命令」や「カード搬送命令」で、正しい位置にカードを搬送する必要があります。

ヒント: プリンタドライバはカードをロードして印刷やエンコーディング処理を行います。しかし、カードがローディングされている場合、このカードに対して引き続き印刷やエンコーディング処理を行います。この機能を利用して、プリンタ制御関数でエンコーディングしたカードに、プリンタドライバで印刷することもできます。

注意: このようにプリンタ制御関数とドライバを併用する場合、Windows のスプーリング機能を使用していると、エンコーディングデータと印刷データにズレが生じる可能性があります。運用には注意してください。

5. 4 プリンタの検索

5. 4. 1 プリンタ検索関数

No	関数名	機能
1	int CXCMD_ScanCX120 (int *piSlot, int *piID)	プリンタと通信するためのスロット番号と ID を取得します。スロット番号「0」、ID「0」からプリンタを検索し、最初に見つかったプリンタのスロット番号と ID を、piSlot、piID に返します。
2	int CXCMD_ScanCX120Next (int *piSlot, int *piID)	プリンタと通信するためのスロット番号と ID を取得します。piSlot、piID 設定値の次から検索を開始し、最初に見つかったプリンタのスロット番号と ID を、piSlot、piID に返します。

5. 5 プリンタのステータス取得

5. 5. 1 テスト・ユニットレディ関数

No	関数名	機能
1	int CXCMD_TestUnitReady (int iSlot, int iID)	プリンタに Test Unit Ready 命令を発行して、プリンタの状態を調べます。0 を返したとき、プリンタがカード搬送命令やエンコーディング命令を受け付け可能なことを示します。

5. 5. 2 リード・ポジション関数

No	関数名	機能
1	int CXCMD_ReadPosition (int iSlot, int iID, BYTE *pbyBuffer)	Read Position 命令を発行し、プリンタ本体内のカード位置情報を取得します。 ◆ pbyBuffer: Read Position データを収納するメモリへのアドレス。メモリの大きさは 8 バイト以上必要です。

1) Read Position Data format

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved					PU	Reserved	
1~6	Reserved							
7	Position							

PU (Position Unknown):

- 0: カードがプリンタ内にあり
- 1: カードはプリンタ内にない

注意: カードがプリンタ本体内に無い場合、カードトレイにカードがあっても PU に1を返します。また、カードが移動中の場合、プリンタは Read Position 命令を受け付けることができません。

Position (印刷位置。この値は PU が 0 の場合のみ有効です)

- 0: 印刷待機位置 1: 磁気エンコーダ 2: 接触ICエンコーダ
- 3: 非接触ICエンコーダ

5. 6 印刷

5. 6. 1 イメージアウト関数

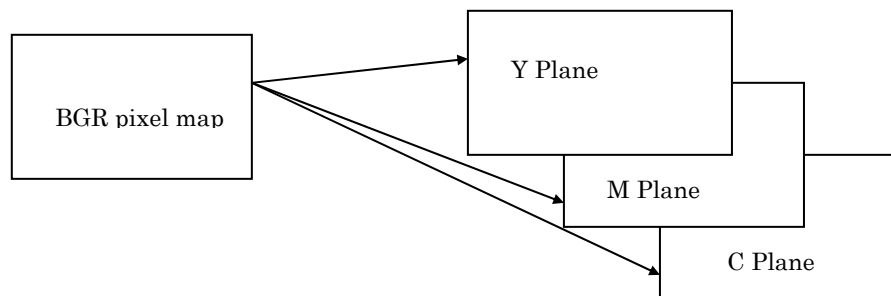
No	関数名	機能
1	int CXCMD_ImageOut (int iSlot, int iID, BYTE * pbyPlane, int iLength, int iColor, int iBuffer)	イメージデータをプリンタに送信します。YMCK-OPのそれぞれの色を別々に送ります。送信しないデータを印刷した場合、プリンタ内に元々あったデータを印刷します。 ◆ pbyPlane: 印刷データを収納したメモリのアドレスを指定します。印刷データの大きさプリンタの最大印刷サイズの 1006x633 バイトです。 ◆ iLength: 印刷データの長さです。1006x633 固定です。 ◆ iColor: 印刷データの種類を指定します。 0: Y インク用データ 1: M インク用データ 2: C インク用データ 3: K インク用データ 5: OP インク用データ ◆ iBuffer: 出力するプリンタの印刷データバッファを指定します。 0: Buffer-0 1: Buffer-1

1) RGB から YMC への変換方法.

プリンタ は YMC のみサポートします。Y、M、C はそれぞれ送信する必要があります。

RGB から YMC への変換は以下のように行います。

RGB と YMC の関係: $Y = 255 - B$; $M = 255 - G$; $C = 255 - R$;



2) K インクについて

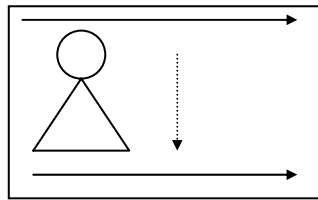
K インクは黒色の文字を印刷するための特別なインクです。文字を K インクで印刷することにより、文字の印刷品位が向上します。値が 0 の場合は印刷せず、0 以外の場合は同一濃度で印刷します。

3) OP について

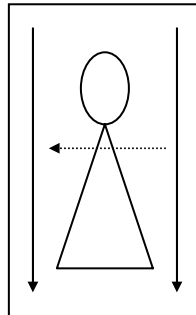
OP は印刷結果を保護するための透明なコーティングです。値が 0 の場合は印刷せず、0 以外の場合は同一濃度で印刷します。

4) イメージ送信順序

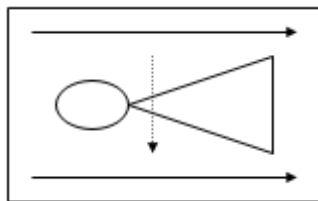
プリンタに送るイメージの順序は常に下図左側の形式 (Landscape 型) になります。右側のようないデザインの場合、ホストシステムで事前に変換して送ってください。



Landscape 形態の場合のデータ出力方向。



Portrait 形態の場合のデータ出力方向



(実際にカードに印刷されるイメージ)

上の図で、

- a. 実線はプリンタへのピクセルの送信順序を示します。
- b. 破線はプリンタへのライン送信順序を示します。Landscape 型の画像と見做し、イメージデータは左端の上端から右端の下端に向かって送信してください。

5. 6. 2 印刷関数

No	関数名	機能
1	int CXCMD_Print (int iSlot, int iID, int iColor, int iBuffer, int bFlip, int bImmed)	<p>カードに印刷後反転し、カードを印刷待機位置に戻します。</p> <ul style="list-style-type: none"> ◆ iColor: 印刷色を Bit 対応で指定します。 bit0: YMC 印刷時 1 bit1: K 印刷時 1 bit2: OP 印刷時 1 bit3: UV 印刷時 1 ◆ iBuffer: 印刷対象の印刷データバッファを指定します。 0: Buffer-0 1: Buffer-1 ◆ bFlip: カード反転の有無を指定します。 TRUE: カード印刷後反転します FALSE: 反転せず 注意) FALSE であっても必ず反転します。 ◆ bImmed: 処理終了時期の指定 TRUE: プリンタが命令を受け付けた後、この関数から戻ります。 FALSE: プリンタが処理を終了した後、この関数から戻ります。
2	int CXCMD_PrintAndEject (int iSlot, int iID, int iColor, int iBuffer, int bFlip, int bImmed)	<p>カードに印刷後、排出します。</p> <ul style="list-style-type: none"> ◆ iColor: 印刷色を Bit 対応で指定します。 bit0: YMC 印刷時 1 bit1: K 印刷時 1 bit2: OP 印刷時 1 bit3: UV 印刷時 1 ◆ iBuffer: 印刷対象の印刷データバッファを指定します。 0: Buffer-0 1: Buffer-1 ◆ bFlip: カード反転の有無を指定します。 TRUE: カード反転後排出します FALSE: 反転せず ◆ bImmed: 処理終了時期の指定 TRUE: プリンタが命令を受け付けた後、この関数から戻ります。 FALSE: プリンタが処理を終了した後、この関数から戻ります。

注意: カードが印刷待機位置に無いときこれらの関数を呼ぶと、エラーを返します。カード搬送中や印刷中にこの関数を呼ぶと、正の値を返します。正の値が返ったとき、一定時間後にリトライしてください。

5. 7 カード搬送と排出

5. 7. 1 カードロード関数

No	関数名	機能
1	int CXCMD_LoadCard (int iSlot, int iID, int iDest, int bFlip, int bImmed)	カードをカードホッパからロードし、指定の場所に搬送します。 ◆ iDest: カード搬送先を指定します。 0: 印刷待機位置 1: 磁気エンコーダ 2: 接触 IC エンコーダ 3: 非接触 IC エンコーダ ◆ bFlip: カード反転の有無を指定します TRUE: カード反転後、搬送先へ FALSE: カード反転せず ◆ bImmed: 処理終了時期を指定します TRUE: プリンタが命令を受け付けた後、この関数から戻ります。 FALSE: プリンタが処理を終了した後、この関数から戻ります。

注意: カードがプリンタ本体にあるとき、この関数を呼ぶと、エラーを返します。カード搬送中や印刷中にこの関数を呼ぶと、正の値を返します。正の値が返ったとき、一定時間後にリトライしてください。

5. 7. 2 カード搬送関数

No	関数名	機能
1	int CXCMD_MoveCard (int iSlot, int iID, int iDest, int bFlip, int bImmed)	カードを指定の場所に搬送します。 ◆ iDest: カード搬送先を指定します。 0: 印刷待機位置 1: 磁気エンコーダ 2: 接触 IC エンコーダ 3: 非接触 IC エンコーダ ◆ bFlip: カード反転の有無を指定します TRUE: カード反転後、搬送先へ FALSE: カード反転せず ◆ bImmed: 処理終了時期を指定します。 TRUE: プリンタが命令を受け付けた後、この関数から戻ります。 FALSE: プリンタが処理を終了した後、この関数から戻ります。

注意: カードがプリンタ本体内に無いときこの関数を呼ぶと、エラーを返します。カード搬送中や印刷中にこの関数を呼ぶと、正の値を返します。正の値が返ったとき、一定時間後にリトライしてください。

5. 7. 3 カード排出関数

No	関数名	機能
1	int CXCMD_EjectCard (int iSlot, int iID, int bFlip, int bImmed)	カードをカード排出口から排出します。 ◆ bFlip: カード反転の有無を指定します TRUE: カード反転後排出 FALSE: カード反転せず ◆ bImmed: 処理終了時期を指定します。 TRUE: プリンタが命令を受け付けた後、この関数から戻ります。 FALSE: プリンタが処理を終了した後、この関数から戻ります。
2	int CXCMD_EjectCardFromNGExt (int iSlot, int iID, int bImmed)	プリンタに Eject 命令を発行し、カードを NG カード排出口から排出します。 ◆ bImmed: 処理終了時期を指定します。 TRUE: プリンタが命令を受け付けた後、この関数から戻ります。 FALSE: プリンタが処理を終了した後、この関数から戻ります。

注意: カードがプリンタ本体内に無いときこの関数を呼ぶと、エラーを返します。カード搬送中や印刷中にこの関数を呼ぶと、正の値を返します。正の値が返ったとき、一定時間後にリトライしてください。

5. 8 磁気エンコーディング

5. 8. 1 書込み関数

No	関数名	機能
1	int CXCMD_WriteMagData (int iSlot, int iID, BYTE *pbyBuff, int iLength, int iMagFormat)	磁気データをカードの磁気ストライプに書き込みます。 ◆ pbyBuff: 書込みデータを収納するメモリのアドレスを指定します。書込みデータは7ビットの ASCII 文字列で設定します。 ◆ iLength: 書込みデータ長を指定します ◆ iMagFormat: 磁気データ種別を指定します 0x07: JIS2 Track1(7単位コード、最大 69 文字) 0x16: ISO Track1(6 単位コード、最大 76 文字) 0x24: ISO Track2(4 単位コード、最大 37 文字) 0x34: ISO Track3(4 単位コード、最大 104 文字)

注意: カードが磁気エンコーダ位置に無いとき、エラーを返します。プリンタがカード搬送中や印刷中のとき、正の値を返します。正の値が返った場合、一定時間後にリトライしてください。

5. 8. 2 読み込み関数

No	関数名	機能
1	int CXCMD_ReadMagData (int iSlot, int iID, BYTE *pbyBuff, int *piLength, int iMagFormat)	カードの磁気ストライプに記録されているデータを読み込みます。 ◆ pbyBuff: 読み込みデータを収納するメモリのアドレスを指定します。データは7ビットの ASCII 文字列で返します。 ◆ piLength: 読込んだデータの長さを返します ◆ iMagFormat: 磁気データ種別を指定します 0x07: JIS2 Track1(7単位コード、最大 69 文字) 0x16: ISO Track1(6 単位コード、最大 76 文字) 0x24: ISO Track2(4 単位コード、最大 37 文字) 0x34: ISO Track3(4 単位コード、最大 104 文字)

注意: カードが磁気エンコーダ位置に無いとき、エラーを返します。プリンタがカード搬送中や印刷中のとき、正の値を返します。正の値が返った場合、一定時間後にリトライしてください。

5. 9 IC エンコーディング

5. 9. 1 IC 制御関数

No	関数名	機能
1	int CXCMD_ICControl (int iSlot, int iID, int iICType, int iAction)	ホストから IC カードのチップにアクセスするための制御を行います。 ◆ iICType:IC 種別を指定します 0:接触型 IC(接触 IC エンコーダ位置にカードがあること) 1:非接触型 IC(非接触 IC エンコーダ位置にカードがあること) ◆ iAction:動作を指定します 0:IC 接点を圧着する 1:IC 接点を離間する 注意)非接触型 IC は圧着/離間動作をしません、IC アクセス前に 圧着命令、IC アクセス終了後に離間命令を必ず発行してください。

注意1: カードが適切な IC エンコーダ位置に無いとき、エラーを返します。プリンタがカード搬送中や印刷中のとき、正の値を返します。正の値が返った場合、一定時間後にリトライしてください。

注意2:IC 接点圧着後の状態で CXCMD_MoveCard()を発行するとエラーになります。非接触 IC の場合でも、IC 接点圧着後は必ず IC 接点を離間してください。

5. 10 プリンタの初期化

5. 10. 1 リゼロ関数

No	関数名	機能
1	int CXCMD_ RezeroUnit (int iSlot, int iID)	プリンタに Rezero 命令を発行して、プリンタを初期化します。プリンタはカードを排出し、印刷インク位置を初期化します。

5. 11 プリンタの設定値取得/変更

5. 11. 1 インクワイアリー関数

No	関数名	機能
1	int CXCMD_StandardInquiry (int iSlot, int iID, BYTE *pbyBuffer)	プリンタに Inquiry 命令を発行し、Inquiry 応答データを取得します。 ◆ pbyBuffer: データを収納するメモリへのアドレス。メモリの大きさは 56 バイト以上確保してください。 注)データフォーマットについては「1) Standard Inquiry Data format」を参照してください。
2	int CXCMD_VPDInquiry (int iSlot, int iID, int iPage, BYTE *pbyBuffer)	プリンタに Inquiry 命令を発行し、Vital Product Data を取得します。 ◆ iPage: データ種別 0x01: Head Information Data. ◆ pbyBuffer: データを収納するメモリへのアドレス。メモリの大きさは 28 バイト以上確保してください。 注)データフォーマットについては「2) Inquiry Head Information Data format」を参照してください。

1) Standard Inquiry Data format

Byte \ Bit	7	6	5	4	3	2	1	0
0	Peripheral Qualifier				Peripheral Device Type			
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0
3	Reserved				0	0	1	0
4	Additional Length (33H)							
5~6	Reserved							
7	0	0	0	0	0	0	0	0
8~15	Vendor Identification "VDS" (ASCII)							
16~31	Product Identification "CX-120" (ASCII)							
32~35	Product Revision Level "???" (ASCII)							
36	Config Revision Level (Binary)							
37	Table Revision Level (Binary)							
38			LA	ICLI	ICJI	ICII	MGJI	MGII
39~42	Laminator Unit Product Revision Level "???" (ASCII)							
43~55	Reserved							

MGII(MG ISO Unit Installed):

1 の場合、磁気 ISO ユニットが実装されています。0 の場合は実装されていません。

MGJI(MG JIS Unit Installed):

1 の場合、磁気 JIS ユニットが実装されています。0 の場合は実装されていません。

ICII(IC Contact ISO Installed):

1 の場合、接触型 IC ユニット(ISO) が実装されています。0 の場合は実装されていません。

ICJI(IC Contact JIS Installed):

1 の場合接触型 IC ユニット(JIS) が実装されています。0 の場合は実装されていません。

ICLI(IC Contactless Installed):

1 の場合、非接触型 IC ユニットが実装されています。0 の場合は実装されていません。

注意)この値は常に1です。

LA(Laminator Attached):

1 の場合、ラミネータが接続されています。0 の場合は接続されていません。

2) Inquiry Head Information Data format

Byte \ Bit	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	01H (Page Code)							
2	Reserved							
3～7	(N-3)H (Page Length)							
8	Head Information (アスキー文字列で、先頭の8文字だけが有効です。残りは0で埋められています)							
27								

5. 11. 2 ログセンス/ログセレクト関数

No	関数名	機能
1	int CXCMD_ LogSelect (int iSlot, int iID)	プリンタに Log Select コマンドを発行し、カード発行枚数(Free Count)を初期化します。カード発行枚数値は CXCMD_ LogSense ()関数で取得することができます。
2	int CXCMD_ LogSense (int iSlot, int iID, int iPage, BYTE *pbyBuffer)	<p>プリンタに Log Sense コマンドを発行し、Log Sense 情報を取得します。</p> <p>◆ iPage: Log 情報の種別を指定します。 0x06: Non-medium error page 0x38: Counters page 0x3a: Medium Quantity page</p> <p>◆ pbyBuffer: データを収納するメモリへのアドレス。メモリの大きさは各ログ情報を収納するのに十分な大きさを確保してください。</p> <p>注)データフォーマットについては「1) Log Sense Data format」を参照してください。</p>

1) Log Sense Data format

(1) Non-medium error page (Page Code=0x06)

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved		Page Code (06H)					
1	Reserved							
2	(MSB) Page Length (0008H) (LSB)							
3								
4	(MSB) Parameter Code (0000H) (LSB)							
5								
6	0	1	0	0	00b		0	0
7	Parameter Length (0004H)							
8	(MSB) Non-medium error count(00H) (LSB)							
9								
10								
11								

*この項目はサポートしていません。“Non-medium error count”の値には常に0を設定します。

(2) Counter page (Page Code=0x38)

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved		Page Code (38H)					
1	Reserved							
2	(MSB) Page Length (0018H) (LSB)							
3								
4	(MSB) Parameter Code (0000H) (LSB)							
5								
6	0	0	0	0	00b		0	0
7	Parameter Length (0004H)							
8	(MSB) Total count (LSB)							
9								
10								
11								
12	(MSB) Parameter Code (0001H) (LSB)							
13								
14	0	0	0	0	00b		0	0
15	Parameter Length (0004H)							
16	(MSB) Free count (LSB)							
17								
18								
19								
20	(MSB) Parameter Code (0002H) (LSB)							
21								
22	0	0	0	0	00b		0	0
23	Parameter Length (0004H)							
24	(MSB) Head count (LSB)							
25								
26								
27								

(3) Medium quantity page (Page Code=0x3a)

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved		Page Code (3AH)					
1	Reserved							
2	(MSB) Page Length (000AH) (LSB)							
3								
4	(MSB) Parameter Code (0000H) (LSB)							
5								
6	0	0	0	0	00b		0	0
7	Parameter Length (0001H)							
8	Ink quantity							
9	(MSB) Parameter Code (0001H) (LSB)							
10								
11	0	0	0	0	00b		0	0
12	Parameter Length (0001H)							
13	Card quantity							

Ink quantity ;パーセンテージでインク残量を示します

0: 無し ~ 50: 一杯 FFH: 不明なインクが装着されている

Card quantity ;カードの有無を示します

0: カードがホツパにある

1: カードがホツパにない

5. 11. 3 モードセンス/モードセレクト関数

No	関数名	機能
1	int CXCMD_ModeSense (int iSlot, int iID, int iPage, BYTE *pbyBuffer)	<p>プリンタに Mode Sense 命令を発行し、Mode Sense 情報を取得します。</p> <ul style="list-style-type: none"> ◆ iPage: 取得データ種別指定 0x23: Ink Sense Page 0x28: Print Unit Information Data Page 0x2a: MG Encode Unit Information Data page 0x2b: Adjust Print Position Page 0x3a: JIS Mode Page ◆ pbyBuffer: Mode sense データを収納するメモリへのアドレス。各 Mode Sense Data の前には 4 バイトの Mode Parameter Header が付加されているので、メモリサイズは Mode Sense Data のサイズよりも 4 バイト以上大きい領域を確保してください。 <p>注) データフォーマットについては「1) Mode Sense/Mode Select Data format」を参照してください。</p>
2	int CXCMD_ModeSelect (int iSlot, int iID, int iPage, BYTE *pbyData)	<p>プリンタに Mode Select 命令を発行し、プリンタの設定を変更します。変更した設定は、プリンタの不揮発性メモリに保存します。</p> <ul style="list-style-type: none"> ◆ iPage: 変更データ種別指定 0x23: Ink Sense Page 0x28: Print Unit Information Data Page 0x2a: MG Encode Unit Information Data page 0x2b: Adjust Print Position Page 0x3a: JIS Mode Page ◆ pbyData: Mode Select データを収納するメモリへのアドレス。メモリの先頭には 4 バイトの空き領域を確保してください。Mode Select データは 5 バイト目以降に作成します。4 バイトの空き領域は CXCMD_ModeSelect()関数が使用します。 <p>注) データフォーマットについては「1) Mode Sense/Mode Select Data format」を参照してください。</p>

1) Mode Sense / Mode Select Data format

(1) Mode Parameter Header

Byte \ Bit	7	6	5	4	3	2	1	0
0	Mode Parameter Length (Reserved when Mode Select)							
1	Reserved							
2	Reserved	0	0	1	Reserved			
3	Block Descriptor Length (00H)							

注) Mode Parameter Header は CXCMD_ModeSelect()関数が書込みます。

(2) Ink Sense Page Data format (Page Code=0x23)

Byte \ Bit	7	6	5	4	3	2	1	0
*4	0	0	Page Code (23H)					
*5	Page Specific Parameter Length (26H)							
6	Ink Code(Read Only)							
*7	Reserved							
8	(MSB) Number of Set of Ink Pannel(Binary, Read Only) (LSB)							
9								
10-11	Reserved							
12	Lot Number(ASCII, Read Only) Only top of 6 characters are effective. Remainder is filled with NULL							
43								

注) “*”でマークした項目は CXCMD_ModeSelect()関数が書込みます。

Ink Code (Read Only: インク種別):

0: YMC-K-OP

1: K

3: YMC-K-UV

4: YMC-K-OP-K

252: Unknown Ink (TAG 記録内容が正しくない)

253: Unknown Ink (TAG との通信に失敗した)

254: Unknown Ink (TAG が存在しない)

255: Unknown Ink (TAG リーダ・ライターとの通信に失敗した)

* Ink Code が 252 から 255 の場合、Lot Number は NULL で埋められます。

(3) Print Unit Information Data Sense Page Data format(Page Code=0x28)

Bit Byte	7	6	5	4	3	2	1	0
*4	0	0	Page Code (28H)					
*5	Page Specific Parameter Length (30H)							
6-7	Reserved							
8	(MSB)	Basic Resolution X (300, Read Only)						(LSB)
9								
10	(MSB)	Basic Resolution Y (300, Read Only)						(LSB)
11								
12-15	Reserved							
16	(MSB)	Card Size X (1013, Read Only)						(LSB)
17								
18	(MSB)	Card Size Y (639, Read Only)						(LSB)
19								
20	Reserved							
21	MG Mode							
22	Contact IC Mode(Read Only)							
23	Contactless IC Mode(Read Only)							
24	Laminator Mode(Read Only)							
25	Reserveed							
26	Card Code(Read Only)							
27	Reserved							
28	(MSB)	Print Position X(0, Read Only)						(LSB)
29								
30	(MSB)	Print Position Y(0, Read Only)						(LSB)
31								
32	(MSB)	Print Size X (1006, Read Only)						(LSB)
33								
34	(MSB)	Print Size Y (633, Read Only)						(LSB)
35								
36-37	Reserved							
38	YMC Level							
39	YMC Print Mode							
40-44	Reserved							
45	Resin Black Level							

46	Resin Black Print Mode
47	OP Level
48	UV Level
49-53	Reserved

注) “*”でマークした項目は CXCMD_ModeSelect()関数が書込みます。

Basic Resolution X =水平方向のプリンタ解像度を DPI 単位で表す.

Basic Resolution Y = 垂直方向のプリンタ解像度を DPI 単位で表す.

Card Size X =水平方向のカードサイズをプリンタの Pixel 数で表す

Card Size Y =垂直方向のカードサイズをプリンタの Pixel 数で表す

MG Mode;

- 0: MG エンコーダなし 1: ISO MG エンコーダが装着されている
2: JIS MAG MG エンコーダが装着されている

Contact IC Mode;

- 0: 接触 IC エンコーダ装着不可 1: 接触 IC エンコーダ装着可能

Non-Contact IC Mode;

- 0: 非接触 IC エンコーダ装着不可 1: 非接触 IC エンコーダ装着可能

Laminator Mode;

- 0: ラミネータなし 1: ラミネータが接続されている

Card Code (Kind of card)

- 0: 標準サイズ

Print Position X =カード左端からの印刷可能開始位置を、プリンタの Pixel 数で示す。

Print Position Y =カード上端からの印刷可能開始位置を、プリンタの Pixel 数で示す。

Print Size X = カード長手方向の印刷可能サイズを、プリンタの Pixel 数で示す。

Print Size Y = カード短手方向の印刷可能サイズを、プリンタの Pixel 数で示す。

YMC Level(YMC インクの印刷レベル);

- 0:最低 ~ 6:最高 (標準値は 3)

YMC Print Mode;

- 0:標準 1:精細

Resin Black Level(K インクの印刷レベル);

- 0:最低 ~ 6:最高 (標準値は 3)

Resin Black Print Mode;

- 0:標準 1:精細

OP Level(OP インクの印刷レベル);

- 0:最低 ~ 6:最高 (標準値は 3)

UV Level(UV インクの印刷レベル);

- 0:最低 ~ 6:最高 (標準値は 3)

(4) MG Encode Unit Information Data Page Data format(Page Code=0x2A)

Bit Byte	7	6	5	4	3	2	1	0
*4	0	0	Page Code (2AH)					
*5	Page Specific Parameter Length (08H)							
6	ISO Mode							
7-10	Reserved							
11	Retry Count							
12-13	Reserved							

注) “*”でマークした項目は CXCMD_ModeSelect()関数が書込みます。

ISO Mode(ISO 磁気エンコーダタイプ)

0: ISO 磁気エンコーダなし

1: ISO 磁気エンコーダあり(Lo-Co:300 Oe)

2: ISO 磁気エンコーダあり(Hi-Co :2750 Oe)

Retry Count = 磁気書込みエラーが発生した場合の最大リトライ数。最大値は3。

(5) Adjust Print Position Page Data format(Page Code=0x2E)

Byte \ Bit	7	6	5	4	3	2	1	0
*4	0	0	Page Code (2BH)					
*5	Page Specific Parameter Length (06H)							
6	Reserved							
7	Offset X							
8	Reserved							
9	Offset Y							
10-11	Reserved							

注) “*”でマークした項目は CXCMD_ModeSelect()関数が書込みます。

Offset X = 水平方向カード印刷開始位置。設定値は 0～14 の範囲。

Offset Y = 垂直方向カード印刷開始位置。設定値は 0～14 の範囲。

* 標準値は 7。

(6) MAG JIS Mode Page Data format(Page Code=0x3A)

Byte \ Bit	7	6	5	4	3	2	1	0
*4	0	0	Page Code (3AH)					
*5	Page Specific Parameter Length (08H)							
6	Reserved							
7	JIS Mode							
8-13	Reserved							

注) “*”でマークした項目は CXCMD_ModeSelect()関数が書込みます。

JIS Mode(JIS 磁気エンコーダタイプ)

- 0: JIS 磁気エンコーダなし
- 1: JIS 磁気エンコーダあり(Lo-Co)
- 2: JIS 磁気エンコーダあり(Hi-Co)

6. プリンタドライバでのエンコーディング方法

プリンタドライバで磁気エンコーディングや IC エンコーディングを行うことができます。標準的な方法ではエンコーディングデータをプリンタドライバに渡すことはできません。エンコーディングデータをプリンタドライバに渡すために、2つの方法を準備しています。

6.1 インラインエンコーディング

印刷データの一部をエンコーディングデータとして渡す方法です。印刷データ中の特定のプレフィックス文字を先頭に持つテキストをエンコーディングします。プレフィックス文字とエンコーディング対象のテキストは印刷しません。

プレフィックス	最大長	コード体系	エンコーディングテキスト
~?0	69	7 単位符号	プレフィックスに続くデータを JIS-2 磁気データとして処理します。
~?1	76	6 単位符号	プレフィックスに続くデータを ISOトラック1 磁気データとして処理します。
~?2	37	4 単位符号	プレフィックスに続くデータを ISOトラック2 磁気データとして処理します。
~?3	104	4 単位符号	プレフィックスに続くデータを ISOトラック3 磁気データとして処理します。
~?4			プレフィックスに続くデータを接触 IC 用データとして処理します。
~?5			プレフィックスに続くデータを非接触 IC 用データとして処理します。

〈注意〉

- ① プリンタドライバの対象エンコーディング設定を有効にしてください。
- ② プレフィックスとエンコーディングテキストは連続させ、半角の同一のフォント(種類/大きさ)を使用してください。
- ③ 上表で、最大長に最大の磁気データ数を、コード体系に磁気データのコード体系を示します。エンコーディングできる文字はコード体系により異なります。付録の「磁気データコード表」を参照してください。
- ④ 7 単位符号の場合、半角カタカナは SI, SO コントロール文字を使ってドライバが 7 単位に変換します。
- ⑤ 接触 IC、非接触 IC エンコーディングの場合、エンコーディングテキストは ASCII 文字に限定してください。

次のようなカードを印刷した場合、“12345678”をISO 磁気トラック 1 にエンコードします。そして“~?112345678”は印刷されません。



6. 2 ExtEscape()によるエンコーディング

WIN32 API ExtEscape()関数を使用してエンコーディングデータをドライバに渡す方法です。Escape ID の値でエンコーディングデータの種類を指定します。Escape ID 値は十進です。

(1)ExtEscape()関数へのパラメータ

```
int ExtEscape (  
    HDC          hdc,           // handle to the device context.  
    Int          nEscape,       // Escape ID  
    int          cbInput,       // size of encoding data  
    LPCSTR       lpzInData,    // encoding data  
    int          cbOutput,     // unused.  
    LPSTR        lpzOutData    // unused.  
);
```

Escape ID値		
No	Escape ID 値	データ種別
1	9010	非接触 IC 用データ
2	9011	接触 IC 用データ
3	9020	JIS-2 磁気データ
4	9021	ISO トラック1 磁気データ
5	9022	ISO トラック2 磁気データ
6	9023	ISO トラック3 磁気データ

(2) ExtEscape()関数からの戻り値

成功した場合、0より大きな値を返します。成功はエンコーディングが正しく行われたという意味ではなく、エンコーディングデータがドライバに渡されたという意味です。

<注意>

- ・ プリンタドライバの対象エンコーディング設定を有効にしてください。
- ・ ExtEscape()は、最初に印刷する面の StartPage() と EndPage()の間で呼んでください。

6. 3 IC エンコーディングの組み込み方法

IC エンコーディングの場合、プリンタは実際のエンコーディングを行いません。カードを IC エンコーディング可能な状態にしたあとで、お客様が準備した IC エンコーディング処理(以降「IC エンコーディング DLL」と記述します)を呼び出します。実際の IC エンコーディングは IC エンコーディング DLL が行います。IC エンコーディングを行うためには、お客様が IC エンコーディング DLL を準備し、Windows の system32 フォルダーに置かなければなりません。ここでは IC エンコーディング DLL の仕様について説明します。

(1) IC エンコード DLL のファイル名

PDR01IC0.DLL : 非接触 IC エンコーディング用

PDR01IC1.DLL : 接触 IC エンコーディング用

L

(2) 関数仕様

```
int stdcall Encode (  
    LPCSTR      lpzInData,      // encoding data  
    int         cbInput,         // size of encoding data  
);
```

(3)関数からの戻り値

戻り値0が成功を意味します。成功が関数から返った場合、処理を継続します。エンコーディング処理全てが終了していれば印刷を開始します。0以外が返った場合、カードを排出し、このカードへの印刷を中断します。

<注意>

プリンタドライバ バージョン 3.0.0.7 から、IC エンコード DLL の戻り値の処理が以下のように変わります。

0: 成功。引き続き印刷を行います。

0x1001: 失敗。ダイアログを表示しません。カードを排出後、新しいカードで再試行します。

0x1002: 失敗。ダイアログを表示しません。カードを排出後、印刷ジョブをキャンセルします。

負値及び上記以外の正值: 失敗。ダイアログを表示し、オペレータの指示に従い動作します。

<注意>

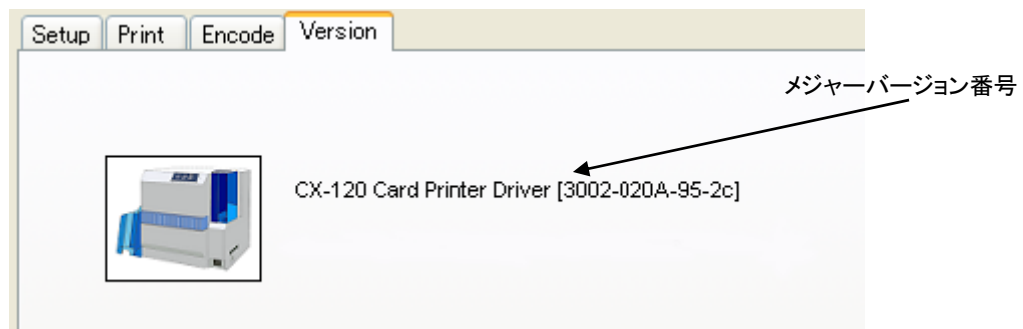
- ・ 同一の PC に複数のプリンタを接続した場合の複数プリンタでの IC エンコーディング動作はサポートしておりません。

7. プリントドライバの設定方法

プリントドライバの設定を WIN32 API ExtEscape()関数を使用して参照したり、変更します。

<注意>

- ⑥ この機能はドライバーのメジャーバージョン番号が3以上で有効です。バージョンはプリントドライバダイアログの Version タグに表示されています。
- ⑦ 設定を変更する場合は、CreateDC()から StartDoc()の間で行ってください。
- ⑧ 変更した設定は、プリンタのハンドルを DeleteDC()で解放するまで有効です。



Versionタグ画面

7. 1 ExtEscape()の使用方法

```
int ExtEscape (  
    HDC          hdc,           // handle to the device context.  
    Int          nEscape, // Escape ID  
    int          cbParameter,   // size of the parameter data  
    LPCSTR       lpzParameter,  // pointer to the parameter data  
    int          cbResult,    // size of the result area  
    LPSTR        lpzResult     // pointer to the result area  
);
```

- nEscape : 9100 を設定する。
- cbParameter : パラメータの長さをバイト長で指定する
- lpzParameter : パラメータを格納するメモリへのポインタ
- cbResult : 結果を格納するメモリの長さをバイト長で指定する
- lpzResult : ドライバからの結果を格納するメモリへのポインタ。

7. 2 設定の変更

1)パラメータの形式

名称	Command Code	ID	Size	Data(設定値)
バイト長	1	2	2	n
値	'S' (0x53)	*1	*2	*1

*1:「7.5 パラメーター一覧」を参照してください。

*2: 設定値のバイト長

注)パラメータ、及び結果の複数バイトの値は全て Big Endian 形式です。

2)結果の形式

(1)正常終了の場合

名称	Error Code	Reserved
バイト長	1	4
値	0x00	不定

(2)異常終了の場合

名称	Error Code	Error Code-A	Error Code-B
バイト長	1	2	2
値	0xff	*1	*1

*1:「7.6 エラーコード一覧」を参照してください。

7. 3 設定の参照

1)パラメータの形式

名称	Command Code	ID
バイト長	1	2
値	'G' (0x47)	*1

*1:「7.5 パラメーター一覧」を参照してください。

2)結果の形式

(1)正常終了の場合

名称	Error Code	ID	Size	Data(参照値)
バイト長	1	2	2	n
値	0x00	*1	*2	*1

*1:「7.5 パラメーター一覧」を参照してください。

*2: 参照値のバイト長

(2)異常終了の場合

名称	Error Code	Error Code-A	Error Code-B
バイト長	1	2	2
値	0xff	*1	*1

*1:「7.6 エラーコード一覧」を参照してください。

7. 4 プログラミングサンプル

1) 設定の変更

```
/*
/* [Number of copies]の値を"100"に変更する
int          escape_id;
unsigned short id, size;
unsigned long data;
unsigned char in[9], out[5];
int          ret;
unsigned short error_code;

escape_id = 9100; id = 257; size = 4; data = 100;

in[0]   = 'S';
in[1]   = (unsigned char)((id >> 8) & 0xFF);
in[2]   = (unsigned char)(id);
in[3]   = (unsigned char)((size >> 8) & 0xFF);
in[4]   = (unsigned char)(size);
in[5]   = (unsigned char)((data >> 24) & 0xFF);
in[6]   = (unsigned char)((data >> 16) & 0xFF);
in[7]   = (unsigned char)((data >> 8) & 0xFF);
in[8]   = (unsigned char)(data);

ret = ExtEscape ( hDC, escape_id, sizeof(in), (const char*)in, sizeof(out), (char*)out );
if (ret > 0) {          // Succeed in calling ExtEscape()
    if (out[0] == 0x00) {    // Succeed
        ;
    } else {                // Error happens in the driver
        // Get error code
        error_code = (unsigned short)((unsigned short)out[1] << 8 | out[2]);
    }
} else {                  //Fail in the ExtEscape()
    ;
}
```

2) 設定の参照

/*

/* [Number of copies]の値を参照する

```
int            escape_id;
unsigned short id, size;
unsigned long  data;
unsigned char  in[3], out[9];
int           ret;
unsigned short error_code;
```

```
escape_id = 9100; id = 257;
```

```
in[0]  = 'G';
in[1]  = (unsigned char)((id >> 8) & 0xFF);
in[2]  = (unsigned char)(id);
```

```
ret = ExtEscape ( hDC, escape_id, sizeof(in), (const char*)in, sizeof(out), (char*)out );
```

```
if (ret > 0) {           // Succeed in calling ExtEscape()
    if (out[0] == 0x00) { // Succeed
        size = (unsigned short)((unsigned short)out[3] << 8 | out[4]);
        data = (unsigned long)((unsigned long)out[5] << 24 | (unsigned long)out[6] << 16 |
                                (unsigned long)out[7] << 8 | out[8]);
    } else {              // Error happens in the driver
        // Get error code
        error_code = (unsigned short)((unsigned short)out[1] << 8 | out[2]);
    }
} else {                  // Fail in the ExtEscape()
    ;
}
```

7.5 パラメーター一覧

注意) 複数バイトの値は全て Big Endian 形式です。

項目名	属性	ID	Size	Data	説明
Number of copies	R/W	0x0101	0x0004	0x00000001~ 0x000003E7	印刷部数 (1~999)
Card Load/Eject settings	R/W	0x0103	0x0004	0x00000000	カードを反転しない
				0x00000001	排出時にカードを反転
				0x00000002	ロード時にカードを反転
				0x00000003	ロード時と排出時にカードを反転
Print Print on both sides Print the back side first	R/W	0x0201	0x0004	0x00000000	印刷しない
				0x00000010	片面に印刷
				0x00000020	両面に印刷
				0x00000021	両面に印刷する (裏面の印刷イメージを先に印刷する)
Print mode [Front side]	R/W	0x0202	0x0004	0x00000001	表面の印刷に YMC インクを使用
				0x00000002	表面の印刷に K インクを使用
				0x00000003	表面の印刷に YMCK インクを使用
				0x00000011	表面の印刷に YMC+OP インクを使用
				0x00000012	表面の印刷に K+OP インクを使用 (指定できません)
				0x00000013	表面の印刷に YMCK+OP インクを使用
Print mode [Back side]	R/W	0x0203	0x0004	0x00000001	裏面の印刷に YMC インクを使用
				0x00000002	裏面の印刷に K インクを使用
				0x00000003	裏面の印刷に YMCK インクを使用
				0x00000011	裏面の印刷に YMC+OP インクを使用
				0x00000012	裏面の印刷に K+OP インクを使用
				0x00000013	裏面の印刷に YMCK+OP インクを使用

項目名	属性	ID	Size	Data	説明
Using of Resin K ink [Front side]	R/W	0x0204	0x0004	0x00000010	表面の黒色テキストの印刷に K インクを使用
				0x00000011	表面の黒色テキスト印刷に K インク、その背景の印刷に YMC インクを使用.
				0x00000020	表面の黒色要素印刷に K インクを使用
				0x00000021	表面の黒色要素の印刷に K インク、その背景印刷に YMC インクを使用.
				0x00000040	ページ分割機能を使用する
Using of Resin K ink [Back side]	R/W	0x0205	0x0004	0x00000010	裏面の黒色テキスト印刷に K インクを使用
				0x00000011	裏面の黒色テキスト印刷に K インク、その背景印刷に YMC インクを使用
				0x00000020	裏面の黒色要素印刷に K インクを使用
				0x00000021	裏面の黒色要素印刷に K インク、その背景印刷に YMC インクを使用.
				0x00000040	ページ分割機能を使用する
Rotate by 180 [Front side]	R/W	0x0206	0x0004	0x00000000	表面の印刷イメージを回転しない
				0x00000001	表面の印刷イメージを 180 度回転
Rotate by 180 [Back side]	R/W	0x0207	0x0004	0x00000000	裏面の印刷イメージを回転しない
				0x00000001	裏面の印刷イメージを 180 度回転
Magnetic encoding	R/W	0x0301	0x0004	0x00000000	磁気エンコードしない
				0x00000010	磁気エンコードする
				0x00000011	磁気エンコード後、カードを反転
Non-contact/Contact IC encoding	R/W	0x0302	0x0004	0x00000000	IC エンコードしない
				0x00000010	接触 IC エンコードする
				0x00000011	接触 IC エンコード後、カードを反転
				0x00000020	非接触 IC エンコードする
				0x00000021	非接触 IC エンコード後、カードを反転
				0x00000030	接触・非接触 IC エンコードする
				0x00000031	接触・非接触 IC エンコード後、カードを反転

7.6 エラーコード一覧

注意) 複数バイトの値は全てビッグエンディアン形式です。

No	Error code A	Error code B	説明
1	0x0901	0x0000	不正な引数が渡されました.
2	0x0902	*	バッファサイズが不足しています. Error code B には, 必要なバッファサイズが格納されます.

<付録1 エラーコード表>

エラーコードの構成は以下の表のようになっています。表中の値は正值の HEXA コードです。
エラーコードは負値なので、実際のエラーコードは(-1)を乗じた値になります。

Error Code(HEXA)				説明
Bit31-24	Bit23-16	Bit15-8	Bit7-0	
0x01	Sense Key	ASC	ASCQ	カードプリンタの障害です。bit0 から bit23 はカードプリンタからのエラーコードです。
0x02	00	XXX		USB ドライバ CX Port Manager の障害です。XXX がエラー種別を示します。
0x02	01	XXX		USBドライバCX Port Managerの障害です。XXX がエラー種別を示します。XXX は不正 SRB のステータス値です。
0x02	02	XXX		USBドライバCX Port Managerの障害です。XXX は不正HA のステータス値です。
0x02	03	XXX		USBドライバCX Port Managerの障害です。XXX は不正 Target のステータス値です。
0x09	XXX			その他のエラーです。

A) プリンターエラー (エラーコード: 0x01xxxxxx)

S K Y	A S C	A S Q	内容	説明
02	04	00	Not ready because of mechanical busy.	CX-120 のメカが動作中で、メカ動作のコマンドを受付けることができません。このエラーは、制御関数が正値に変換して返します。
02	04	01	Not ready because the printer is on the way of initializing.	プリンタがメカ初期化中です。
02	04	80	Not ready because of door open	
02	04	81	Not ready because of no cleaning roller	
02	04	82	Not ready because of MG head cleaning.	
02	04	83	Not ready because the printer is in the Self-Test mode.	プリンタが自己診断モード中です。
02	04	84	Not ready because the printer is in the down load mode.	プリンタが電源投入後のダウンロードモード中です。電源断まで本モードを継続します。
02	04	85	Not ready because the printer is in the offline mode.	OFF LINE 動作中です。 ①ローラクリーニングモード実行中 ②自己診断実行中
02	3A	00	No card in the hopper.	
02	3A	80	Laminator is Not Ready	ラミネータが使用できる状態ではありません。エラーの検出タイミングは印刷処理が終了したカードをラミネータに渡すときです。ラミネータの障害が解消すればラミネートを自動再開します。
02	FE	00	Certification failure	認証失敗。パスワード機能が有効で認証処理が終了していないとき、あるいはログオンせずにプリンタを使用したときに発生します。

S K Y	A S C	A S Q	内容	説明
03	0C	00	MG writing error happens.	
03	11	00	MG reading error happens.	
03	36	00	Ink failure.	CX-120 初期化時、インク貼付 TAG と交信できませんでした。CX-120 の RESET スイッチを押し、再度初期化を行うことで印刷できる状態になりますが、インク種別が不明のため正しく印刷できるかは不明です。
03	36	01	The diameter of the ink ribbon is too large.	(現在、カードプリンタはこのエラーを生成しません)
03	3B	81	Card JAM was detected during card loading.	カードカセット部でカード JAM が発生。
03	3B	82	Card JAM was detected during card turning over.	本体内でカード JAM が発生。ドアを開け、プリンタ内からカードを取り除く必要があります。
03	3B	83	Card JAM was detected during card transporting or printing.	本体内でカード JAM が発生。ドアを開け、プリンタ内からカードを取り除く必要があります。
03	3B	84	Card jam was detected in the MG encoder	本体内でカード JAM が発生。ドアを開け、プリンタ内からカードを取り除く必要があります。
03	3B	85	Card JAM was detected during card unloading.	本体内でカード JAM が発生。ドアを開け、プリンタ内からカードを取り除く必要があります。
03	3B	86	Card JAM was detected in Laminator unit	両面ラミネートで、片面ラミネート時にラミネータ内でカード JAM が発生。ラミネータ内のカードを取り除いてラミネータを初期化したあと、プリンタを初期化する必要があります。
03	3B	88	Card JAM was detected during card unloading to the NG card exit.	本体内でカード JAM が発生。ドアを開け、プリンタ内からカードを取り除く必要があります。

S K Y	A S C	A S Q	内容	説明
04	08	82	Communication error happens between Laminator unit	プリンタ初期化時、ラミネータとノ接続不良かラミネータの障害で初期化処理が正しくできませんでした。ラミネータ間の接続と、ラミネータの状態を確認する必要があります。
04	36	81	Ink mark is not detected	インク不良.インク交換が必要です。
04	40	01	Unit EEPROM failure	解消には電源の再投入が必要です
04	40	02	Head EEPROM failure	解消には電源の再投入が必要です
04	44	00	Hardware error happens.	解消には電源の再投入が必要です
04	44	80	Firmware error happens in mechanical control procedure.	解消には電源の再投入が必要です
04	44	81	Firmware error happens in system control procedure.	解消には電源の再投入が必要です
04	44	82	Firmware error happens in pixel control procedure.	解消には電源の再投入が必要です
04	44	83	Firmware error happens in host interface procedure.	解消には電源の再投入が必要です
04	AB	00	MG unit mechanical error happens.	解消には電源の再投入が必要です
04	C0	00	Flip unit mechanical error happens.	解消には電源の再投入が必要です
04	C1	00	Head mechanical error happens.	解消には電源の再投入が必要です
04	F6	00	Temperature in the printer is too hot.	カードが本体内にあるときには発生しません。解消には電源の再投入が必要です。
04	F6	01	Temperature in the printer is too cool.	カードが本体内にあるときには発生しません。解消には電源の再投入が必要です。
04	F8	00	Temperature of the thermal head is too hot.	カードが本体内にあるときには発生しません。解消には電源の再投入が必要です。
04	F8	01	Temperature around the thermal head is too cool.	カードが本体内にあるときには発生しません。解消には電源の再投入が必要です。

S K Y	A S C	A S Q	内容	説明
05	1A	00	Length of Parameter list is invalid.	
05	20	00	Invalid operation code in CDB	CDB の OPE コードが不正です。
05	20	80	IC encoder is not installed	
05	20	81	MG encoder is not installed	
05	24	00	Invalid field in CDB	
05	24	01	Invalid Color Code in CDB	コマンドタスクが CDB 内のカラーコードデータを評価し、不正値が設定されている場合応答。エラー状態は保持しない。
05	24	80	USB failure	プリンタ内の USB ドライバ関連障害です。
05	25	00	Logical unit not supported	CDB 内の Logical Unit Number が 0 ではありません。
05	26	01	Parameter is not supported	Parameter List の Page Code が不正です
05	26	02	Parameter value is invalid	Parameter List の Page Code 以降の定義領域のデータが不正です
05	26	80	MG encoding data is invalid.	
05	26	81	Download data is invalid	ファームウェアダウンロードのデータが正しくありません。
05	2C	00	Command sequence error	発行コマンドシーケンスに矛盾があります。
06	28	00	CX-120 has been initialized.	Inquiry 以外の命令に対し、1 回のみ応答します。
06	29	00	CX-120 has been powered.	Inquiry 以外の命令に対し、1 回のみ応答します。「CX-120 has been initialized.」と双方が発生した場合、本応答のみを応答します。
42	00	81	Ink end.	消費済みインクを装着した場合、あるいは印刷中にインク上の END マークを検出した後に応答します。カードが本体内にあるときには発生しません。
42	00	82	End of Laminator Film.	プリンタがラミネータにカードを渡すときに検知します。本エラーが発生したとき、本体内のカードは正常に処理を終了しています。ラミネータが動作可能状態になると自動的にラミネータを再開します。

B) Driver エラー (エラーコード: 0x02xxxxxx)

Bit31-24 (0x02)	Bit23-16	Bit15-0	説明
0x02	00	1	メモリ不足
0x02	00	2	ドライバがビジー状態で、コマンドを受け付けることができません。
0x02	00	3	ドライバに出力したコマンドが強制終了しました。
0x02	00	4	ドライバに出力したコマンドが規定時間内に終了しません。
0x02	00	5	プリンタが見つかりません。
0x02	00	6	カードプリンタ制御 DLL が使用するソフトウェアが正しくインストールされていません。
0x02	01	XXXX	不正 SRB。SRB 値は Bit15 から Bit0 に設定されています。
0x02	02	XXXX	不正 HA ステータス。HA ステータスは Bit15 から Bit0 に設定されています。
0x02	03	XXXX	不正ターゲットステータス。ターゲットステータスは Bit15 から Bit0 に設定されています。

C) その他のエラー (エラーコード: 0x09xxxxxx)

Bit31-24 (0x09)	Bit23-0	説明
0x09	1	パラメータが不正です。
0x09	2	プリンタが見つかりません。
0x09	3	メモリ不足。
0x09	4	ファイルへのアクセスに失敗。
0x09	5	イメージファイルの形式が無効。

<付録2. 磁気データコード表>

4ビット符号					
b4	b3	b2	b1		
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	A	:
1	0	1	1	B	;
1	1	0	0	C	<
1	1	0	1	D	=
1	1	1	0	E	>
1	1	1	1	F	?

参考) 特殊コード

No.	用途	コード
1	始め符号	;
2	終わり符号	?
3	分離符号	=
4	ハードウェアコントロール用	:<>

注) 始め符号と終わり符号はデータとして使用してはならない

(続き)

6ビット符号									
						0	0	1	1
						0	1	0	1
b4	b3	b2	b1		0	1	2	3	
0	0	0	0	0		0	@	P	
0	0	0	1	1	!	1	A	Q	
0	0	1	0	2	“	2	B	R	
0	0	1	1	3	#	3	C	S	
0	1	0	0	4	\$	4	D	T	
0	1	0	1	5	%	5	E	U	
0	1	1	0	6	&	6	F	V	
0	1	1	1	7	‘	7	G	W	
1	0	0	0	8	(8	H	X	
1	0	0	1	9)	9	I	Y	
1	0	1	0	A	*	:	J	Z	
1	0	1	1	B	+	;	K	[
1	1	0	0	C	,	<	L	¥	
1	1	0	1	D	－	=	M]	
1	1	1	0	E	.	>	N	^	
1	1	1	1	F	/	?	O	_	

参考) 特殊コード

No.	用途	コード
1	始め符号	%
2	終わり符号	?
3	分離符号	^
4	ハードウェアコントロール用	!“&‘*+,,:;<=>@[¥]_

注) 始め符号と終わり符号はデータとして使用してはならない

(続き)

7ビット符号											
					b7	0	0	0	0	1	1
					b6	0	0	1	1	0	0
					b5	0	1	0	1	0	1
b4	b3	b2	b1			0	1	2	3	4	5
0	0	0	0	0					0	@	P
0	0	0	1	1					!	1	A
0	0	1	0	2					“	2	B
0	0	1	1	3					#	3	C
0	1	0	0	4					\$	4	D
0	1	0	1	5					%	5	E
0	1	1	0	6					&	6	F
0	1	1	1	7					‘	7	G
1	0	0	0	8					(8	H
1	0	0	1	9)	9	I
1	0	1	0	A					*	:	J
1	0	1	1	B					+	;	K
1	1	0	0	C					,	<	L
1	1	0	1	D					-	=	M
1	1	1	0	E					.	>	N
1	1	1	1	F					/	?	O

参考) 特殊コード

No.	用途	コード
1	始め符号	0x7f (DEL)
2	終わり符号	0x7f (DEL)
3	分離符号	^
4	ハードウェアコントロール用	! “ & ‘ * + , ; < = > @ [¥] _

注) 始め符号と終わり符号はデータとして使用してはならない

