



SII SDK for Windows  
アプリケーションプログラマーズガイド

Rev.01

[対象製品]

RP-F10シリーズ

RP-G10シリーズ

RP-E10シリーズ

セイコーインスツル株式会社

Rev.01

2024年 11月

©セイコーインスツル株式会社 2024

無断転載を禁じます。

Microsoft®及びWindows®は米国Microsoft Corporationの米国、日本及びその他の国における登録商標です。

Bluetooth はBluetooth SIG, Inc.の登録商標です。

本書の内容は、断りなく変更することがあります。

本書及び本書に記載された製品の利用によって発生した  
損害及びその回復に要する費用に対し、当社は一切の責任を負いかねます。

本書を第三者に無断で頒布することを禁じます。

---

# はじめに

---

本書は、セイコーインスツル株式会社（以降：SII）が提供する「SII SDK for Windows」（以降：SDK）について説明します。  
SDKは「対象製品」に記載のプリンタードライバー上で動作します。

## 対象製品

---

本書の対象製品について下記に記載します。

プリンター	プリンタードライバー
RP-F10シリーズ RP-G10シリーズ RP-E10シリーズ	SII Printer Driver for Windows

# 本書の表記

本書の表記について説明します。

## 用語一覧

本書で使用する用語について下記の通りに定義します。

用語	内容
技術説明書	下記の技術説明書。 ・RP-F10シリーズ サーマルプリンタ 技術説明書 ・RP-G10シリーズ サーマルプリンタ 技術説明書 ・RP-E10シリーズ サーマルプリンタ 技術説明書
プリンターコマンド	「技術説明書」に記載されている、プリンターを制御するための命令。
ASB設定コマンド (ASB: Automatic Status Back)	プリンターコマンド「自動ステータス送信の有効・無効」。 プリンターコマンドの詳細は、「技術説明書」の「コマンドの機能」を参照してください。
プリンターステータス	SDKで取得できるプリンターのステータス情報。 プリンターコマンド「自動ステータス送信の有効・無効」に応答するステータスに、拡張ステータスを付加した内容です。プリンターステータスの内容は、「6.1 プリンターステータス一覧」を参照してください。

<b>1章 概要</b>	<b>1-1</b>
1.1 動作条件	1-1
<b>2章 インストール</b>	<b>2-1</b>
<b>3章 Win32 API</b>	<b>3-1</b>
3.1 開発対象言語	3-1
3.2 ライブラリファイル	3-1
3.3 API一覧	3-2
3.4 API詳細	3-4
OpenMonPrinter	3-4
CloseMonPrinter	3-5
LockPrinter	3-6
UnlockPrinter	3-7
DirectIO	3-8
DirectIOEx	3-9
Reset	3-11
GetStatus	3-12
SetStatusBackFunction	3-13
SetStatusBackWnd	3-14
CancelStatusBack	3-15
SetBarcodeDataBackFunction	3-16
CancelBarcodeDataBack	3-17
PowerOff	3-18
GetCounter	3-19
ResetCounter	3-20
GetType	3-21
GetPrnCapability	3-22
OpenDrawer	3-23
ControlFeature	3-24
SendDataFile	3-25
DirectSendRead	3-26
GetProperty	3-27
SetProperty	3-28
<b>4章 .NET API</b>	<b>4-1</b>
4.1 開発対象言語	4-1
4.2 ライブラリファイル	4-1
4.3 プリンター	4-2
4.3.1 API一覧	4-2
共通 API	4-2
専用 API	4-2
4.3.2 共通プロパティ	4-4
LastError	4-4
IsValid	4-4

4.3.3	共通メソッド .....	4-6
	OpenMonPrinter .....	4-6
	CloseMonPrinter .....	4-7
4.3.4	専用プロパティ .....	4-8
	Status .....	4-8
4.3.5	専用メソッド .....	4-9
	LockPrinter .....	4-9
	UnlockPrinter .....	4-10
	DirectIOEx .....	4-10
	ResetPrinter .....	4-12
	SetStatusBack .....	4-13
	CancelStatusBack .....	4-13
	PowerOff .....	4-14
	GetCounter .....	4-14
	ResetCounter .....	4-15
	GetType .....	4-16
	GetPrnCapability .....	4-17
	OpenDrawer .....	4-18
	ControlFeature .....	4-19
	SendDataFile .....	4-20
	DirectSendRead .....	4-20
	GetProperty .....	4-22
	SetProperty .....	4-23
4.3.6	イベント .....	4-25
	StatusCallback .....	4-25
4.4	ディスプレイ .....	4-26
4.5	バーコードスキャナー .....	4-27
4.5.1	API一覧 .....	4-27
	共通 API .....	4-27
	専用 API .....	4-27
4.5.2	共通プロパティ .....	4-28
	LastError .....	4-28
	IsValid .....	4-28
4.5.3	共通メソッド .....	4-30
	OpenMonPrinter .....	4-30
	CloseMonPrinter .....	4-31
4.5.4	専用プロパティ .....	4-32
	SetBarcodeDataBack .....	4-32
	CancelBarcodeDataBack .....	4-33
4.5.5	イベント .....	4-34
	BarcodeDataCallback .....	4-34

---

## 5章 エラーコード一覧 5-1

---

5.1	エラーコード一覧 .....	5-1
-----	----------------	-----

---

## 6章 引数情報 6-1

---

6.1	プリンタステータス一覧 .....	6-1
6.2	カウンターID .....	6-2
6.3	プリンターID .....	6-3

6.4	タイプID	6-3
6.5	対象ドロワ	6-4
6.6	ドロワ駆動時間	6-4
6.7	プリンターの機能	6-5
6.8	プロパティID	6-5
6.9	バーコードデータ構造体	6-11
6.9.1	Win32 API	6-11
6.9.2	.NET API	6-12

---

<b>7章</b>	<b>コマンド定義ファイル</b>	<b>7-1</b>
7.1	概要	7-1
7.2	フォーマット	7-1
7.3	利用方法	7-3

---

---

# 1章 概要

---

本章ではSDKの概要について説明します。

SDKには、開発者向けに提供されるプリンターを直接制御するためのライブラリファイルが含まれます。また、プリンタードライバーに付属し、プリンタードライバーを利用して動作します。

SDKの利用により、アプリケーション開発においてプリンターを直接制御し、ポートの種類に依存しない設計ができます。また、DEVMODEのプライベート領域の設定項目の一部の値を取得及び変更することが可能です。

SDKは下記のライブラリファイルを内包しています。

- Win32開発環境用SDK(以降: Win32 API)
- .NET開発環境用SDK(以降: .NET API)

SDKの利用例として、各開発対象言語別のサンプルプログラムを提供していますので参照してください。

## 1.1 動作条件

---

SDKの動作条件は、プリンタードライバーの動作環境、プリンターの設定及び制限事項に準じます。プリンタードライバーの動作環境などの詳細は、「SII Printer Driver for Windows 取扱説明書」を参照してください。

その他に、下記の動作条件を満たす必要があります。

- .NET APIを利用する場合は、.NET Framework 2.0以降が必要です。コンピューターから.NET Frameworkをアンインストールすると.NET APIを利用できなくなります。
- SDKのすべての機能は双方向サポートの機能が有効、及びプリンタスプールの機能が無効である必要があります。
- 以下のすべてに該当する場合は、共有プリンターを利用してください。(共有プリンターを利用しない場合、送信データが割り込む可能性があります。)
  - RP-E10を利用している場合
  - TCP/IP接続で複数のホストから1台のプリンターを利用する場合
  - 送信データを複数回に分割して送信する場合



---

## 2章 インストール

---

インストールについては、「SII ソフトウェアパッケージ for Windows インストールガイド」を参照してください。

---

## 3章 Win32 API

---

本章ではWin32 APIについて説明します。

### 3.1 開発対象言語

---

開発対象言語は下記の通りです。

- Visual C++

### 3.2 ライブラリファイル

---

Win32 APIのライブラリファイルはダイナミックリンクライブラリ形式です。

ライブラリファイルのファイル名は下記の通りです。

- SPSWL\_API.dll

ライブラリファイルの保管場所は、Windowsのシステムフォルダーです。

特殊な場合を除き、ライブラリファイルが保存されているフォルダーへパスを通す必要はありません。フォルダーから移動せずにそのまま使用してください。

ライブラリファイルを移動するとプリンタードライバーのバージョンアップの際にライブラリファイルが正しく更新されません。

## 3.3 API一覧

Win32 APIに実装されているAPIは下記の通りです。

API	機能概略
OpenMonPrinterA <sup>※1</sup> OpenMonPrinterW <sup>※1</sup>	Win32 APIの使用を開始し、APIハンドルを返します。
CloseMonPrinter	使用中のWin32 APIを終了します。
LockPrinter	他のプロセスからのプリンターへのデータ送信及びハードウェアリセット要求を全て禁止(ロック)します。
UnlockPrinter	LockPrinterによる他プロセスからのアクセス禁止(ロック)を解除します。
DirectIO	バイナリデータを送信及び受信します。 (受信データにはASB設定コマンドの応答は含まれません。)
DirectIOEx	バイナリデータを送信及び受信します。
Reset	プリンターをハードウェアリセットします。
GetStatus	最新のプリンターステータスを取得します。
SetStatusBackFunction	プリンターステータスの変化を検出すると呼び出されるコールバック関数を登録します。
SetStatusBackWnd	プリンターステータスの変化を検出した際、クリックイベントが呼び出されるボタンのウィンドウハンドルと、プリンターステータスをセットする変数を登録します。
CancelStatusBack	SetStatusBackFunctionまたはSetStatusBackWndで実行されたコールバック関数の登録を解除します。
SetBarcodeDataBackFunction	バーコードスキャナーのコールバック関数を登録します。
CancelBarcodeDataBack	SetBarcodeDataBackFunctionで実行されたコールバック関数の登録を解除します。
PowerOff	プリンターを電源オフ状態にします。
GetCounter	メンテナンスカウンターを取得します。
ResetCounter	メンテナンスカウンターを初期化します。
GetType	プリンターの各種IDを取得します。
GetPrnCapability	プリンター情報を取得します。
OpenDrawer	指定したドロワを駆動します。
ControlFeature	印刷以外のプリンターの機能を制御します。
SendDataFileA <sup>※1</sup> SendDataFileW <sup>※1</sup>	コマンド定義ファイルの内容を、SDK内部のメモリに登録します。
DirectSendReadA <sup>※1</sup> DirectSendReadW <sup>※1</sup>	SendDataFileで登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信します。
GetProperty	印刷設定内容(一部)を取得します。
SetProperty	印刷設定内容(一部)を変更します。

※1: 文字列の引数はMBCS(マルチバイトコード)、またはUNICODE(ユニコード)で指定します。MBCS(マルチバイトコード)を使用する場合は末尾に'A'、UNICODE(ユニコード)を使用する場合は末尾に'W'のついたAPIを呼び出してください。以降の説明ではAPI末尾の'A'、または'W'を省略して説明します。

## 注意

- ◆ 推奨バーコードスキャナー、及びバーコードスキャナーの設定については、「RP-F10シリーズ サーマルプリンタ 取扱説明書」を参照してください。

## 参考

- ディスプレイのAPIについては、DSP-A01用「SII SDK for Windows アプリケーションプログラマーズガイド」を参照してください。
- RP-G10、RP-E10はバーコードスキャナー及びディスプレイはサポートしていません。

## 3.4 API詳細

### 注意

◆ Bluetooth接続では、プリンターと未接続中の応答データの取得はできません。

## OpenMonPrinter

Win32 APIの使用を開始し、APIハンドルを返します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
INT OpenMonPrinter(  
    INT i_type,  
    LPCTSTR i_prt)
```

### パラメーター

*i\_type*

オープンタイプ

プリンター単体で使用する場合 : 2

バーコードスキャナーを接続したプリンターを使用する場合 : 4

*i\_prt*

Win32 APIを使用するプリンター名

プリンター名(フレンドリ名)を指定します。

### 戻り値

成功 : プリンターを識別するAPIハンドルを返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- ・1プロセスで同時に取得できるAPIハンドルの数は最大8個です。全プロセスでは最大126個になります。
- ・本APIで取得したAPIハンドルを使用しなくなった場合、必ず**CloseMonPrinter**を使って無効にしてください。

- プリンタードライバーの接続先は、USB、シリアル（BluetoothデバイスのCOMポートを含む）、TCP/IPのいずれかに設定してください。
- **SetBarcodeDataBackFunction**、または**CancelBarcodeDataBack**を使用する場合は、*i\_type*は4を指定してください。
- プリンターが未接続、または電源オフ状態でも本APIは成功します。
- バーコードスキャナーが未接続でも本APIは成功します。

## CloseMonPrinter

使用中のWin32 APIを終了します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
INT CloseMonPrinter(
    INT i_hdl)
```

### パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。

### 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- 本APIで指定したAPIハンドルが他のAPIで使用中の場合は、その処理が完了するまで本APIの制御は戻りません。
- APIハンドルに関連付けられたすべての設定及びデータは、本APIで破棄されます。

# LockPrinter

他のプロセスからのプリンターへのデータ送信及びハードウェアリセット要求を全て禁止(ロック)します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT LockPrinter(  
    INT i_hdl,  
    DWORD i_timeout)
```

## パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*i\_timeout*

タイムアウト時間

本APIの成功を待つ時間を、ミリ秒(ms)で指定します。

有効範囲は3000～90000です。

3000未満の値を指定した場合は3000msに設定されます。

90000よりも大きい値を指定した場合は90000msに設定されます。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 本APIを呼び出してから、UnlockPrinterが呼び出されるまでの間に、他のプロセスからプリンターに直接アクセスするようなAPIを呼び出した場合、それらのAPIは失敗します。
- 本APIによりロックされるのは他のプロセスのみ有効です。自身のプロセスからの呼び出しはAPIハンドルやスレッドに関わらずロックの対象とはなりません。
- 本APIにより、同一のAPIハンドルで最大99回まで多重のロックが可能です。ロックを解除するためには本APIを呼び出した回数と同じ回数のUnlockPrinterを呼び出してください。
- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- TCP/IP接続の場合、送信をしない時間がTCP/IPの受信タイムアウト時間を超える前にロックを解除してください。ロックを解除しない場合は、データ抜けの発生や他のホストからの送信データがプリンターに割り込む可能性があります。  
受信タイムアウト時間については、「SII Communication Setting Utility for Windows 取扱説明書」に記載の「3.2.3 各接続方式の設定画面」の「LAN通信の設定」を参照してください。

# UnlockPrinter

LockPrinterによる他プロセスからのアクセス禁止(ロック)を解除します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT UnlockPrinter(  
    INT i_hdl)
```

## パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- LockPrinterが多重に呼び出されている場合、ロック解除には同じ回数の本APIの呼び出しが必要です。



# DirectIO

バイナリデータを送信及び受信します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT DirectIO(  
    INT i_hdl,  
    BYTE i_wlen,  
    LPBYTE i_wcmd,  
    LPBYTE io_rlen,  
    LPBYTE o_rbuf,  
    DWORD i_timeout,  
    BOOL i_flag)
```

## パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*i\_wlen*

送信データサイズ

送信するデータのサイズを指定します。

*i\_wcmd*

送信データバッファ

送信するデータが格納されたバッファを指定します。

*io\_rlen*

受信データサイズ

プリンターから受信するデータの最大長を指定します。

データ取得が必要ない場合は、0バイトを指定してください。

APIから制御が戻ると取得した受信データサイズが格納されます。

*o\_rbuf*

受信データバッファ

取得するデータを格納するバッファを指定します。

*i\_timeout*

タイムアウト時間

本APIの成功を待つ時間を、ミリ秒(ms)で指定します。

有効範囲は3000～90000です。

3000未満の値を指定した場合は3000msに設定されます。

90000よりも大きい値を指定した場合は90000msに設定されます。

*i\_flag*

受信動作フラグ

受信動作を指定するフラグを下記から指定します。

TRUE : 何らかのデータを受信するか、タイムアウトになるまで受信を続けます。

FALSE : 受信データサイズ分を受信するか、タイムアウトになるまで受信を続けます。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 別データの割り込みが許されないコマンドやデータ、イメージデータの送信の場合は、**LockPrinter**を呼び出した後に本APIを呼び出してください。**LockPrinter**を呼び出さない場合、別プロセスからのデータが割り込む恐れがあります。
- 送信するデータに、応答を無効にするASB設定コマンドを含まないでください。プリンタステータスを取得するAPIが正常に動作しなくなります。
- 受信データにはASB設定コマンドの応答は含まれません。ASB設定コマンドの応答を含むデータの取得を行う場合は**DirectIOEx**を呼び出してください。
- 本APIは**Reset**で中断できます。
- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。
- Bluetooth接続の場合は、送信データにプリンターコマンド「プリンタの初期化」以外のプリンターを初期化するプリンターコマンドを含めないでください。プリンターの初期化については、「技術説明書」を参照してください。ハードウェアリセットする場合は、**Reset**を実行してください。
- 本APIはバーコードスキャナーからの応答データを取得しません。バーコードスキャナーからの応答データの取得は**SetBarcodeDataBackFunction**で登録するコールバック関数を使用してください。

## DirectIOEx

バイナリデータを送信及び受信します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT DirectIOEx(  
    INT i_hdl,  
    DWORD i_wlen,  
    LPBYTE i_wcmd,  
    LPDWORD io_rlen,  
    LPBYTE o_rbuf,  
    DWORD i_timeout,  
    BOOL i_flag,  
    BYTE i_op)
```

## パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*i\_wlen*

送信データサイズ

送信するデータのサイズを指定します。

*i\_wcmd*

送信データバッファ

送信するデータが格納されたバッファを指定します。

*io\_rlen*

受信データサイズ

プリンターから受信するデータの最大長を指定します。

受信データサイズは最大4096バイトです。

4096よりも大きい値を指定した場合は4096バイトに設定されます。

データ取得が必要ない場合は、0を指定してください。

APIから制御が戻ると受信されたデータサイズが格納されます。

*o\_rbuf*

受信データバッファ

取得するデータを格納するバッファを指定します。

*i\_timeout*

タイムアウト時間

本APIの成功を待つ時間を、ミリ秒(ms)で指定します。

有効範囲は3000～90000です。

3000未満の値を指定した場合は3000msに設定されます。

90000よりも大きい値を指定した場合は90000msに設定されます。

*i\_flag*

受信動作フラグ

受信動作を指定するフラグを下記から指定します。

TRUE : 何らかのデータを受信するか、タイムアウトになるまで受信を続けます。

FALSE : 受信データサイズ分を受信するか、タイムアウトになるまで受信を続けます。

*i\_op*

受信対象オプション

受信対象のデータを下記から指定します。

0 : ASB設定コマンドの応答を除いたデータを取得します。

1 : ASB設定コマンドの応答を含めたデータを取得します。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 別データの割り込みが許されないコマンドやデータ、イメージデータの送信の場合は、LockPrinterを呼び出した後に本APIを使用してください。LockPrinterを呼び出さない場合、別プロセスからのデータが割り込む恐れがあります。
- 送信するデータに、応答を無効にするASB設定コマンドを含めないでください。プリンタステータスを取得するAPIが正常に動作しなくなります。
- 本APIはResetで中断できます。

- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。
- Bluetooth接続の場合は、送信データにプリンターコマンド「プリンタの初期化」以外のプリンターを初期化するプリンターコマンドを含めないでください。プリンターの初期化については、「技術説明書」を参照してください。ハードウェアリセットする場合は、**Reset**を実行してください。
- 本APIはバーコードスキャナーからの応答データを取得しません。バーコードスキャナーからの応答データの取得は**SetBarcodeDataBackFunction**で登録するコールバック関数を使用してください。

## Reset

プリンターをハードウェアリセットします。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
INT Reset(
    INT i_hdl)
```

### パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。

### 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- 通信プロトコルを使用した(プリンターコマンドを使用しない)ハードウェアリセットを行います。
- 本APIを呼び出した後、数秒経過してからデータを送信してください。本API呼び出し直後にデータ送信を行うとデータ抜けの原因になります。
- 本APIを呼び出した場合、下記のAPIは中断されます。
  - **DirectIO**
  - **DirectIOEx**
  - **DirectSendRead**
- 本API呼び出し中のプリンターステータスは、「応答なし」の応答になります。プリンターステータスの内容は、「6.1 プリンターステータス一覧」を参照してください。
- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。

- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。
- Bluetooth接続で、プリンターがデータを受け付けられない状態で本APIを実行した場合、本APIは成功しますが、リセットはプリンターが印刷可能状態となるまで実行されません。また、それまでの間データ送信は行えません。

## GetStatus

最新のプリンターステータスを取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
INT GetStatus(
    INT i_hdl,
    LPDWORD o_status )
```

### パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*o\_status*

プリンターステータス変数

プリンターステータスを格納する変数を指定します。

### 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- プリンターの再接続を検出した場合のプリンターステータスは、その時点で最後に受信しているプリンターステータスになります。
- プリンターステータスの内容は、「6.1 プリンターステータス一覧」を参照してください。

## SetStatusBackFunction

プリンタステータスの変化を検出すると呼び出されるコールバック関数を登録します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
INT SetStatusBackFunction(  
    INT i_hdl,  
    INT ( CALLBACK EXPORT *lpStatusCB ) ( DWORD o_st ) )
```

### パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*lpStatusCB*

コールバック関数アドレス

プリンタステータスを受け取るアプリケーション定義のコールバック関数のアドレスを指定します。

NULLを指定した場合、プリンタステータスの監視を中断します。

*o\_st*

プリンタステータス変数

プリンタステータスが格納される変数を指定します。

### 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- ・本APIによりコールバック関数を登録すると、現在のプリンタステータスでコールバック関数を呼び出します。
- ・プリンタステータスを受信した場合でも、直前に受信したプリンタステータスと比べ変化がない場合、コールバック関数は呼び出されません。
- ・プリンターの再接続を検出した場合のプリンタステータスは、その時点で最後に受信しているプリンタステータスになります。
- ・コールバック関数を登録した状態で本APIを呼び出した場合、登録済みの関数は無効となり、新しいコールバック関数が登録されます。
- ・すでに登録済みで有効なコールバック関数を再度指定して本APIを呼び出した場合も、直後のプリンタステータス応答は行われます。
- ・本APIで登録したコールバック関数は、下記のAPIで解除されます。
  - CloseMonPrinter
  - CancelStatusBack

- 登録されたコールバック関数内から同一のAPIハンドルで下記のAPIを呼び出すことはできません。
  - **CloseMonPrinter**
  - **SetStatusBackFunction**
  - **SetStatusBackWnd**
  - **CancelStatusBack**
  - **SetBarcodeDataBackFunction**
  - **CancelBarcodeDataBack**
- コールバック関数の戻り値は無視されます。
- プリンターステータスの受信からコールバック関数の呼び出しまでの時間は保証されません。
- プリンターステータスの内容は、「6.1 プリンターステータス一覧」を参照してください。

## SetStatusBackWnd

プリンターステータスの変化を検出した際、クリックイベントが呼び出されるボタンのウィンドウハンドルと、プリンターステータスをセットする変数を登録します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
INT SetStatusBackWnd(
    INT i_hdl,
    HANDLE i_Wnd,
    LPDWORD o_status )
```

### パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。

*i\_Wnd*

ウィンドウハンドル

クリックイベントを送るボタンのウィンドウハンドルを指定します。  
NULLを指定した場合、本APIを解除します。

*o\_status*

プリンターステータス変数

プリンターステータスを格納する変数を指定します。

### 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 本APIによりボタンのウィンドウハンドルを登録すると、現在のプリンタステータスでクリックイベントを呼び出します。
- プリンタステータスを受信した場合でも、直前に受信したプリンタステータスと比べ変化がない場合、クリックイベントは呼び出されません。
- プリンターの再接続を検出した場合のプリンタステータスは、その時点で最後に受信しているプリンタステータスになります。
- ボタンのウィンドウハンドルを登録した状態で本APIを呼び出した場合、登録済みのボタンのウィンドウハンドルは無効となり、新しいボタンのウィンドウハンドルが登録されます。
- すでに登録済みで有効なボタンのウィンドウハンドルを再度指定して本APIを呼び出した場合も、直後のプリンタステータス応答は行われます。
- 本APIは下記のAPIで解除されます。
  - **CloseMonPrinter**
  - **CancelStatusBack**
- 呼び出されたクリックイベントの処理ルーチン内から、同一のAPIハンドルで下記のAPIを呼び出すことはできません。
  - **CloseMonPrinter**
  - **SetStatusBackFunction**
  - **SetStatusBackWnd**
  - **CancelStatusBack**
  - **SetBarcodeDataBackFunction**
  - **CancelBarcodeDataBack**
- クリックイベントの戻り値は無視されます。
- プリンタステータスの受信からクリックイベントの呼び出しまでの時間は保証されません。
- プリンタステータスの内容は、「6.1 プリンタステータス一覧」を参照してください。

## CancelStatusBack

**SetStatusBackFunction**または**SetStatusBackWnd**で呼び出されたコールバック関数の登録を解除します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT CancelStatusBack(  
    INT i_hdl)
```

## パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。



## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- **SetStatusBackFunction**または**SetStatusBackWnd**のいずれの登録も行われていない場合でも本APIは成功します。

# SetBarcodeDataBackFunction

バーコードスキャナーのコールバック関数を登録します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	—	—

## 構文

```
INT SetBarcodeDataBackFunction(  
    INT i_hdl,  
    INT ( CALLBACK EXPORT *lpDataCB ) ( ST_BARDATA lpData ) )
```

## パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。

*lpDataCB*

コールバック関数アドレス

バーコードスキャナーからの応答データを受け取るアプリケーション定義のコールバック関数のアドレスを指定します。

NULLを指定した場合、バーコードスキャナーの監視を中断します。

*lpData*

バーコードデータ構造体

バーコードスキャナーからの応答データ、応答サイズが格納される構造体のアドレスを指定します。

バーコードデータ構造体については、「6.9 バーコードデータ構造体」を参照してください。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- コールバック関数の対象は下記の通りです。
  - バーコードデータの受信
  - バーコードスキャナーの接続
  - バーコードスキャナーの切断
- 本APIを使用する場合は、ユーザーのアプリケーションにバーコードデータ構造体を定義する必要があります。バーコードデータ構造体については、「6.9 バーコードデータ構造体」を参照してください。
- 受信データはバイナリの形式で応答します。受信データサイズは最大64KBです。最大サイズを超えた場合は、古い応答から削除されます。
- 本APIによりコールバック関数を登録すると、現在のバーコードスキャナーの接続状態でコールバック関数を呼び出します。
- バーコードスキャナーの接続状態を受信した場合でも、直前に受信したバーコードスキャナーの接続状態と比べ変化がない場合、コールバック関数は呼び出されません。
- コールバック関数を登録した状態で本APIを呼び出した場合、登録済みの関数は無効となり、新しいコールバック関数が登録されます。
- すでに登録済みで有効なコールバック関数を再度指定して本APIを呼び出した場合も、直後のバーコードスキャナーの接続状態応答は行われます。
- 本APIで登録したコールバック関数は、下記のAPIで解除されます。
  - **CloseMonPrinter**
  - **CancelBarcodeDataBack**
- 登録されたコールバック関数内から同一のAPIハンドルで下記のAPIを呼び出すことはできません。
  - **CloseMonPrinter**
  - **SetStatusBackFunction**
  - **SetStatusBackWnd**
  - **CancelStatusBack**
  - **SetBarcodeDataBackFunction**
  - **CancelBarcodeDataBack**
- コールバック関数の戻り値は無視されます。
- バーコードデータの受信からコールバック関数の呼出しまでの時間は保証されません。

## CancelBarcodeDataBack

SetBarcodeDataBackFunctionで呼び出されたコールバック関数の登録を解除します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	-	-

## 構文

```
INT CancelBarcodeDataBack(  
    INT i_hdl)
```

## パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- **SetBarcodeDataBackFunction**の登録が行われていない場合でも本APIは成功します。

# PowerOff

プリンターを電源オフ状態にします。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT PowerOff(  
    INT i_hdl)
```

## パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## GetCounter

メンテナンスカウンタを取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
INT GetCounter(  
    INT i_hdl,  
    WORD i_readno,  
    LPDWORD o_readcounter)
```

### パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*i\_readno*

カウンタID

取得するメンテナンスカウンタのカウンタIDを指定します。  
指定可能な内容は、「6.2 カウンタID」を参照してください。

*o\_readcounter*

カウンタ変数

取得したカウンタ値を格納する変数を指定します。

### 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。

# ResetCounter

メンテナンスカウンタを初期化します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT ResetCounter(  
    INT i_hdl,  
    WORD i_readno)
```

## パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。

*i\_readno*

カウンタID

カウンタ値を初期化するメンテナンスカウンタのカウンタIDを指定します。  
指定可能な内容は、「6.2 カウンタID」を参照してください。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 本APIで指定したカウンタIDのカウンタ値が初期化されたことを**GetCounter**で確認してください。
- 他のプロセスで**LockPrinter**が呼び出されている場合、本APIを呼び出すと失敗します。
- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。

# GetType

プリンターの各種IDを取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT GetType(  
    INT i_hdl,  
    LPBYTE o_typeID  
    LPBYTE o_fontID  
    LPBYTE o_exrom  
    LPBYTE o_special)
```

## パラメーター

*i\_hdl*  
APIハンドル  
OpenMonPrinterで取得したAPIハンドルを指定します。

*o\_typeID*  
ID1  
3が格納されます。

*o\_fontID*  
ID2  
2が格納されます。

*o\_exrom*  
予約  
NULLを指定してください。

*o\_special*  
予約  
NULLを指定してください。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。  
エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。

# GetPrnCapability

プリンター情報を取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT GetPrnCapability(  
    INT i_hdl,  
    BYTE i_id,  
    LPBYTE io_datsize,  
    LPBYTE o_dat)
```

## パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*i\_id*

プリンターID

取得するプリンター情報のプリンターIDを指定します。

指定可能な内容は、「6.3 プリンターID」を参照してください。

*io\_datsize*

受信データサイズ

取得するプリンター情報を格納するバッファサイズを指定します。

APIから制御が戻ると取得したプリンター情報のデータサイズが格納されます。

指定したバッファサイズが取得するプリンター情報の応答サイズよりも小さい場合、本

APIは失敗し、指定したプリンター情報の応答サイズが格納されます。

*o\_dat*

受信データバッファ

取得したプリンター情報を格納するバッファを指定します。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。

# OpenDrawer

指定したドロワを駆動します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT OpenDrawer(  
    INT i_hdl,  
    BYTE i_drawer,  
    BYTE i_pulse )
```

## パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*i\_drawer*

対象ドロワ

駆動する対象ドロワを指定します。

指定可能な内容は、「6.5 対象ドロワ」を参照してください。

*i\_pulse*

ドロワ駆動時間

対象ドロワの駆動パルスのON/OFF時間を指定します。ON時間とOFF時間は同じ時間が指定されます。

指定可能な内容は、「6.6 ドロワ駆動時間」を参照してください。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。
- ドロワの制御時間については、お客様がご利用されているドロワの仕様に従ってください。



## ControlFeature

印刷以外のプリンターの機能を制御します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	–

### 構文

```
INT ControlFeature(  
    INT i_hdl,  
    BYTE i_feature,  
    INT i_arg1,  
    INT i_arg2)
```

### パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*i\_feature*

プリンターの機能

制御するプリンターの機能を指定します。

指定可能な内容は、「6.7 プリンターの機能」を参照してください。

*i\_arg1*

引数1

指定したプリンターの機能に対する引数を指定します。

指定可能な内容は、「6.7 プリンターの機能」を参照してください。

*i\_arg2*

引数2

指定したプリンターの機能に対する引数を指定します。

指定可能な内容は、「6.7 プリンターの機能」を参照してください。

### 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。

## SendDataFile

コマンド定義ファイルの内容を、SDK内部のメモリに登録します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
INT SendDataFile(  
    INT i_hdl,  
    LPCTSTR i_fname)
```

### パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。

*i\_fname*

コマンド定義ファイル名

決められた書式で作成されたコマンド定義ファイル名を指定します。

### 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- 本APIによって登録されたコマンド定義は、**CloseMonPrinter**で破棄されます。
- コマンド定義ファイルの詳細は、「7章 コマンド定義ファイル」を参照してください。

## DirectSendRead

SendDataFileで登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
INT DirectSendRead(  
    INT i_hdl,  
    LPCTSTR i_cname,  
    LPCTSTR i_rtype,  
    LPDWORD io_rlen,  
    LPBYTE o_rbuf,  
    DWORD i_timeout,  
    BOOL i_flag)
```

### パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*i\_cname*

コマンド名

SendDataFileで登録したコマンド名を指定します。

*i\_rtype*

受信データタイプ名

受信するデータのタイプを下記から指定します。

ASB : ASB設定コマンドの応答のみを受信データに保存します。

Other : 上記以外のプリンターからの応答を受信データに保存します。

*io\_rlen*

受信データサイズ

プリンターから受信するデータの最大長を指定します。

受信データサイズは最大4096バイトです。

4096よりも大きい値を指定した場合は4096バイトに設定されます。

データ取得が必要ない場合は、0を指定してください。

APIから制御が戻ると取得した受信データサイズが格納されます。

*o\_rbuf*

受信データバッファ

取得するデータを格納するバッファを指定します。

*i\_timeout*

タイムアウト時間

本APIの成功を待つ時間を、ミリ秒(ms)で指定します。

有効範囲は3000～90000です。

3000未満の値を指定した場合は3000msに設定されます。

90000よりも大きい値を指定した場合は90000msに設定されます。

*i\_flag*

#### 受信動作フラグ

受信動作を指定するフラグを下記から指定します。

TRUE : 何らかのデータを受信するか、タイムアウトになるまで受信を続けます。

FALSE : 受信データサイズ分を受信するか、タイムアウトになるまで受信を続けます。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 本APIはResetで中断できます。
- 他のプロセスでLockPrinterが呼び出されている場合、本APIを呼び出すと失敗します。
- プリンターと未接続、または通信不能状態の場合は、本APIは失敗します。
- Bluetooth接続の場合は、送信データにプリンターコマンド「プリンタの初期化」以外のプリンターを初期化するプリンターコマンドを含めないでください。プリンターの初期化については、「技術説明書」を参照してください。ハードウェアリセットする場合は、Resetを実行してください。

# GetProperty

印刷設定内容(一部)を取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT GetProperty(  
    INT i_hdl,  
    LPDEVMODE i_devmode,  
    BYTE i_pid,  
    LPBYTE o_dat,  
    LPDWORD io_size )
```

## パラメーター

*i\_hdl*

APIハンドル

OpenMonPrinterで取得したAPIハンドルを指定します。

*i\_devmode*

Devmodeアドレス

Devmodeアドレスを指定します。

*i\_pid*

プロパティID

取得する印刷設定内容のプロパティIDを指定します。  
指定可能な内容は、「6.8 プロパティID」を参照してください。

*o\_dat*

取得データバッファ

取得した印刷設定内容を格納するバッファを指定します。  
NULLを指定した場合、本APIは失敗し、指定した印刷設定内容のデータサイズが*io\_size*に格納されます。

*io\_size*

取得データサイズ

印刷設定内容を格納するバッファサイズを指定します。  
APIから制御が戻ると取得した印刷設定内容のデータサイズが格納されます。  
指定したバッファサイズが取得する印刷設定内容のデータサイズよりも小さい場合、本APIは失敗し、指定した印刷設定内容のデータサイズが格納されます。

## 戻り値

成功 : 0を返します。

失敗 : エラーコードを返します。  
エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## SetProperty

印刷設定内容(一部)を変更します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

```
INT SetProperty(  
    INT i_hdl,  
    LPDEVMODE i_devmode,  
    BYTE i_pid,  
    LPBYTE i_dat,  
    LPDWORD i_size )
```

## パラメーター

*i\_hdl*

APIハンドル

**OpenMonPrinter**で取得したAPIハンドルを指定します。

*i\_devmode*

Devmodeアドレス

Devmodeアドレスを指定します。

*i\_pid*

プロパティID

変更する印刷設定内容のプロパティIDを指定します。

指定可能な内容は、「6.8 プロパティID」を参照してください。

*i\_dat*

設定データバッファ

変更する印刷設定内容が格納されたバッファを指定します。

*i\_size*

設定データサイズ

変更する印刷設定内容が格納されたバッファサイズを指定します。

## **戻り値**

成功 : 0を返します。

失敗 : エラーコードを返します。

エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

---

## 4章 .NET API

---

本章では.NET APIについて説明します。

### 4.1 開発対象言語

---

開発対象言語は下記の通りです。

- Visual Basic .NET
- Visual C#

### 4.2 ライブラリファイル

---

.NET APIのライブラリファイルはクラスライブラリ形式です。

ライブラリファイルのファイル名は下記の通りです。

- SPSWL\_Cls.dll

ライブラリファイルの保管場所は、グローバル・アセンブリ・キャッシュ(GAC)フォルダーです。

## 4.3 プリンター

### 4.3.1 API 一覧

.NET APIに実装されているAPIは下記の通りです。

- ネームスペース : SII.SDK.PrinterDevice
- クラス名 : StatusAPI

#### 共通 API

種別	API	機能概略
プロパティ	LastError	最後に実行したAPIのエラー値を取得します。
プロパティ	IsValid	OpenMonPrinterの呼び出し状態を取得します。
メソッド	OpenMonPrinter	.NET APIの使用を開始します。
メソッド	CloseMonPrinter	使用中の.NET APIを終了します。

#### 専用 API

種別	API	機能概略
プロパティ	Status	最新のプリンターステータスを取得します。
メソッド	LockPrinter	他のプロセスからのプリンターへのデータ送信及びハードウェアリセット要求を全て禁止(ロック)します。
メソッド	UnlockPrinter	LockPrinterによる他プロセスからのアクセス禁止(ロック)を解除します。
メソッド	DirectIOEx	バイナリデータを送信及び受信します。
メソッド	ResetPrinter	プリンターをハードウェアリセットします。
メソッド	SetStatusBack	StatusCallbackイベントによるプリンターステータスの通知を開始します。
メソッド	CancelStatusBack	StatusCallbackイベントによるプリンターステータスの通知を停止します。
メソッド	PowerOff	プリンターを電源オフ状態にします。
メソッド	GetCounter	メンテナンスカウンタを取得します。
メソッド	ResetCounter	メンテナンスカウンタを初期化します。
メソッド	GetType	プリンターの各種IDを取得します。
メソッド	GetPrnCapability	プリンター情報を取得します。
メソッド	OpenDrawer	指定したドロワを駆動します。
メソッド	ControlFeature	印刷以外のプリンターの機能を制御します。
メソッド	SendDataFile	コマンド定義ファイルの内容を、SDK内部のメモリに登録します。
メソッド	DirectSendRead	SendDataFileで登録したコマンド定義を実行します。
メソッド	GetProperty	印刷設定内容(一部)を取得します。
メソッド	SetProperty	印刷設定内容(一部)を変更します。
イベント	StatusCallback	応答されたプリンターステータスを通知します。



## 注意

- ◆ Bluetooth接続では、プリンターと未接続中の応答データの取得はできません。

## 4.3.2 共通プロパティ

### LastError

最後に呼び出したAPIのエラー値を取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

#### 構文

```
ErrorCode LastError { get; }
```

#### 初期値

SUCCESS

#### 備考

- エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### IsValid

OpenMonPrinterの呼び出し状態を取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

#### 構文

```
bool IsValid { get; }
```

#### 初期値

FALSE

## **備考**

- TRUE : **OpenMonPrinter**に成功した状態です。
- FALSE : **OpenMonPrinter**に成功していない状態です。

### 4.3.3 共通メソッド

## OpenMonPrinter

.NET APIの使用を開始します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
ErrorCode OpenMonPrinter(  
    OpenType type,  
    string name )
```

### パラメーター

*type*  
オープンタイプ  
OpenType.TYPE\_PRINTER (固定)

*name*  
.NET APIを使用するプリンター名  
プリンター名(フレンドリ名)を指定します。

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- 1プロセスで.NET APIの各クラス合わせて最大8インスタンスまで同時にオープンできます。
- .NET APIを使用しなくなった場合は、必ず**CloseMonPrinter**を呼び出してください。
- プリンタードライバーの接続先は、USB、シリアル (BluetoothデバイスのCOMポートを含む)、TCP/IPのいずれかに設定してください。
- プリンターが未接続、または電源オフ状態でも本APIは成功します。

## CloseMonPrinter

使用中の.NET APIを終了します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

ErrorCode **CloseMonPrinter()**

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- APIに関連付けられたすべての設定及びデータは、本APIで破棄されます。

## 4.3.4 専用プロパティ

### Status

最新のプリンターステータスを取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

#### 構文

```
ASB Status { get; }
```

#### 初期値

```
ASB_NO_RESPONSE
```

#### 備考

- プリンターの再接続を検出した場合のプリンターステータスは、その時点で最後に受信しているプリンターステータスになります。
- `IsValid`が`FALSE`の時は、正しい値が取得できません。
- プリンターステータスの内容は、「6.1 プリンターステータス一覧」を参照してください。

## 4.3.5 専用メソッド

### LockPrinter

他のプロセスからのプリンターへのデータ送信及びハードウェアリセット要求を全て禁止(ロック)します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

#### 構文

```
ErrorCode LockPrinter(  
    int timeout )
```

#### パラメーター

*timeout*

タイムアウト時間

本APIの成功を待つ時間を、ミリ秒(ms)で指定します。

有効範囲は3000～90000です。

3000未満の値を指定した場合は3000msに設定されます。

90000よりも大きい値を指定した場合は90000msに設定されます。

#### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

#### 備考

- 本APIにより、最大99回まで多重のロックが可能です。ロックを解除するためには本APIを呼び出した回数と同じ回数の**UnlockPrinter**を呼び出してください。
- TCP/IP接続の場合、送信をしない時間がTCP/IPの受信タイムアウト時間を超える前にロックを解除してください。ロックを解除しない場合は、データ抜けの発生や他のホストからの送信データがプリンターに割り込む可能性があります。  
受信タイムアウト時間については、「SII Communication Setting Utility for Windows 取扱説明書」に記載の「3.2.3 各接続方式の設定画面」の「LAN通信の設定」を参照してください。

# UnlockPrinter

LockPrinterによる他プロセスからのアクセス禁止(ロック)を解除します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

ErrorCode **UnlockPrinter**()

## 戻り値

エラーコードを返します。エラーコードの内容は、5.1 エラーコード一覧」を参照してください。

## 備考

- LockPrinterが多重に呼び出されている場合、ロック解除には同じ回数の本APIの呼び出しが必要です。

# DirectIOEx

バイナリデータを送信及び受信します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

- (a)の構文は、バイナリデータを送信後、受信データをプリンターからバイナリデータとして取得します。  
(b)の構文は、バイナリデータを送信後、受信データをプリンターから文字列データとして取得します。  
(c)の構文は、バイナリデータを送信します。

## 構文

(a) ErrorCode **DirectIOEx**(  
    byte[] *cmd*,  
    ref byte[] *data*,  
    int *timeout*,  
    bool *readFlag*,  
    byte *option* )



(b) ErrorCode **DirectIOEx**(  
    byte[] *cmd*,  
    out string *data*,  
    int *timeout*,  
    byte *option* )

(c) ErrorCode **DirectIOEx**(  
    byte[] *cmd*,  
    int *timeout* )

## パラメーター

*cmd*

送信データバッファー

送信するデータが格納されたバッファーを指定します。

*data*

受信データバッファー

取得するデータを格納するバッファーを指定します。

受信データサイズは最大4096バイトです。

4096よりも大きい値を指定した場合は4096バイトに設定されます。

データ取得が必要ない場合は、0を指定してください。

APIから制御が戻ると受信されたデータサイズが格納されます。

*timeout*

タイムアウト時間

本APIの成功を待つ時間を、ミリ秒(ms)で指定します。

有効範囲は3000～90000です。

3000未満の値を指定した場合は3000msに設定されます。

90000よりも大きい値を指定した場合は90000msに設定されます。

*readFlag*

受信動作フラグ

受信動作を指定するフラグを下記から指定します。

TRUE : 何らかのデータを受信するか、タイムアウトになるまで受信を続けます。

FALSE : 受信データサイズ分を受信するか、タイムアウトになるまで受信を続けます。

*option*

受信対象オプション

受信対象のデータを下記から指定します。

0 : ASB設定コマンドの応答を除いたデータを取得します。

1 : ASB設定コマンドの応答を含めたデータを取得します。

## 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 本APIが成功した場合、構文(a)の*data*は*data*で指定した受信データサイズを限度に、実際の受信データのサイズにリサイズされます。
- 送信するデータに、応答を無効にするASB設定コマンドを含めないでください。プリンタステータスを取得するAPIが正常に動作しなくなります。
- 本APIは**ResetPrinter**で中断できます。
- 受信データに0x02が含まれる場合、0x02は0x5fに変換されます。

- Bluetooth接続の場合は、送信データにプリンターコマンド「プリンタの初期化」以外のプリンターを初期化するプリンターコマンドを含めないでください。プリンターの初期化については、「技術説明書」を参照してください。ハードウェアリセットする場合は、**ResetPrinter**を実行してください。

## ResetPrinter

プリンターをハードウェアリセットします。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

ErrorCode **ResetPrinter**()

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- 通信プロトコルを使用した(プリンターコマンドを使用しない)ハードウェアリセットを行います。
- 本APIを呼び出した後、数秒経過してからデータを送信してください。本API呼び出し直後にデータ送信を行うとデータ抜けの原因になります。
- 本APIを呼び出した場合、下記のAPIは中断されます。
  - **DirectIOEx**
  - **DirectSendRead**
- 本API呼び出し中のプリンターステータスは、「応答なし」の応答になります。プリンターステータスの内容は、「6.1 プリンターステータス一覧」を参照してください。

## SetStatusBack

StatusCallbackイベントによるプリンタステータスの通知を開始します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

ErrorCode **SetStatusBack**()

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- 本APIによりプリンタステータスの通知を開始すると、現在のプリンタステータスで **StatusCallback** イベントが発生します。
- **StatusCallback** イベントにイベントハンドラーを登録していない場合、本APIは失敗します。
- すでに**StatusCallback** イベントを開始済みで再度本APIを呼び出した場合も、直後のプリンタステータス応答は行われます。
- 本APIで開始した**StatusCallback** イベントは、下記のAPIで停止されます。
  - **CloseMonPrinter**
  - **CancelStatusBack**
- プリンタステータスの受信からイベント発生までの時間は保証されません。
- プリンタステータスの内容は、「6.1 プリンタステータス一覧」を参照してください。

## CancelStatusBack

StatusCallbackイベントによるプリンタステータスの通知を停止します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

ErrorCode **CancelStatusBack**()

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- **SetStatusBack**によりプリンタステータスの通知が開始されていない場合でも本APIは成功します。

## PowerOff

プリンターを電源オフ状態にします。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

ErrorCode **PowerOff**()

## 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 本APIを実行すると、プリンターで電源オフに伴う処理が実行されます。

## GetCounter

メンテナンスカウンターを取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

メンテナンスカウンターの取得には、CounterIndexに定義されたカウンターIDを用いる構文と、数値で指定する構文の2種類があります。

(a)の構文は、CounterIndexに定義されたカウンターIDを使用し、カウンター値を取得します。

(b)の構文は、カウンターIDを数値で指定し、カウンター値を取得します。

## 構文

- (a) ErrorCode **GetCounter**(  
CounterIndex *index*,  
bool *type*,  
out int *data* )
- (b) ErrorCode **GetCounter**(  
byte *index*,  
out int *data* )

## パラメーター

*index*

カウンターID

取得するメンテナンスカウンターのカウンターIDを指定します。  
CounterIndexに定義されたカウンターIDを使用する場合は、(a)を指定してください。  
カウンターIDを数値で指定する場合は、(a)を指定してください。  
指定可能な内容は、「6.2 カウンターID」を参照してください。

*type*

メンテナンスカウンターのタイプ

取得するメンテナンスカウンターのタイプを下記から指定します。  
TRUE : 積算カウンター  
FALSE : 初期化可能カウンター

*data*

カウンター変数

取得したカウンター値を格納する変数を指定します。

## 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## ResetCounter

メンテナンスカウンターを初期化します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

メンテナンスカウンターの初期化には、CounterIndexに定義されたカウンターIDを用いる構文と、数値で指定する構文の2種類があります。

(a)の構文は、CounterIndexに定義されたカウンターIDを使用し、カウンター値を初期化します。

(b)の構文は、カウンターIDを数値で指定し、カウンター値を初期化します。

## 構文

- (a) ErrorCode **ResetCounter**(  
CounterIndex *index*)
- (b) ErrorCode **ResetCounter**(  
byte *index*)

## パラメーター

*index*

カウンターID

カウンター値を初期化するメンテナンスカウンターのカウンターIDを指定します。  
CounterIndexに定義されたカウンターIDを使用する場合は、(a)を指定してください。  
カウンターIDを数値で指定する場合は、(b)を指定してください。  
指定可能な内容は、「6.2 カウンターID」を参照してください。

## 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 本APIで指定したカウンターIDの値が初期化されたことを**GetCounter**で確認してください。

# GetType

プリンターの各種IDを取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

ErrorCode **GetType**(  
out byte *typeId*,  
out byte *fontID*,  
out byte *exrom*,  
out byte *special*)

## パラメーター

*typeId*

ID1

3が格納されます。

*fontID*  
ID2  
2が格納されます。

*exrom*  
予約  
nullを指定してください。

*special*  
予約  
nullを指定してください。

## 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## GetPrnCapability

プリンター情報を取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

プリンター情報の取得には、PrinterIdに定義されたプリンターIDを用いる構文と、数値で指定する構文の2種類があります。

- (a) の構文は、PrinterIdに定義されたプリンターIDを使用し、プリンター情報をバイナリデータで取得します。
- (b) の構文は、PrinterIdに定義されたプリンターIDを使用し、プリンター情報を文字列データで取得します。
- (c) の構文は、プリンターIDを数値で指定し、プリンター情報をバイナリデータで取得します。
- (d) の構文は、プリンターIDを数値で指定し、プリンター情報を文字列データで取得します。

## 構文

- (a) ErrorCode **GetPrnCapability**(  
PrinterId *id*,  
out byte[] *data* )
- (b) ErrorCode **GetPrnCapability**(  
PrinterId *id*,  
out string *data* )
- (c) ErrorCode **GetPrnCapability**(  
byte *id*,  
out byte[] *data* )

(d) ErrorCode **GetPrnCapability**(  
byte *id*,  
out string *data* )

## パラメーター

*id*

プリンターID

取得するプリンター情報のプリンターIDを指定します。  
PrinterIdに定義されたプリンターIDを使用する場合は、(a)、(b)を指定してください。(a)はプリンター情報の応答形式をバイナリコードで取得し、(b)はASCII文字列で取得します。プリンターIDを数値で指定する場合は、(c)、(d)を指定してください。(c)はプリンター情報の応答形式をバイナリコードで取得し、(d)はASCII文字列で取得します。指定可能な内容は、「6.3 プリンターID」を参照してください。

*data*

受信データバッファー

取得したプリンター情報を格納するバッファーを指定します。

## 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

# OpenDrawer

指定したドロワを駆動します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

## 構文

ErrorCode **OpenDrawer**(  
Drawer *drawer*,  
Pulse *pulse* )

## パラメーター

*drawer*

対象ドロワ

駆動する対象ドロワを指定します。  
指定可能な内容は、「6.5 対象ドロワ」を参照してください。



*pulse*

#### ドロワ駆動時間

対象ドロワの駆動パルスのON/OFF時間を指定します。ON時間とOFF時間は同じ時間が指定されます。

指定可能な内容は、「6.6 ドロワ駆動時間」を参照してください。

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- ・ドロワの制御時間については、お客様がご利用されているドロワの仕様に従ってください。

## ControlFeature

印刷以外のプリンターの機能を制御します。

#### サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	—

### 構文

```
ErrorCode ControlFeature(  
    Feature feature,  
    int arg1,  
    int arg2)
```

### パラメーター

*feature*

#### プリンターの機能

制御するプリンターの機能を指定します。

指定可能な内容は、「6.7 プリンターの機能」を参照してください。

*arg1*

#### 引数1

指定したプリンターの機能に対する引数を指定します。

指定可能な内容は、「6.7 プリンターの機能」を参照してください。

*arg2*

#### 引数2

指定したプリンターの機能に対する引数を指定します。

指定可能な内容は、「6.7 プリンターの機能」を参照してください。

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## SendDataFile

コマンド定義ファイルの内容を、SDK内部のメモリに登録します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

### 構文

```
ErrorCode SendDataFile(  
    string fileName )
```

### パラメーター

*fileName*

コマンド定義ファイル名

決められた書式で作成されたコマンド定義ファイル名を指定します。

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- 本APIによって登録されたコマンド定義は、**CloseMonPrinter**で破棄されます。
- コマンド定義ファイルの詳細は、「7章 コマンド定義ファイル」を参照してください。

## DirectSendRead

**SendDataFile**で登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

- (a)の構文は、**SendDataFile**で登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信後、受信データをプリンターからバイナリデータとして取得します。
- (b)の構文は、**SendDataFile**で登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信後、受信データをプリンターから文字列として取得します。
- (c)の構文は、**SendDataFile**で登録したコマンド定義ファイルについて、指定されたコマンド名に対応する送信データを送信します。

## 構文

(a) ErrorCode **DirectSendRead**(  
    string *cmdName*,  
    string *readType*,  
    ref byte[] *data*,  
    int *timeout*,  
    bool *readFlag*)

(b) ErrorCode **DirectSendRead**(  
    string *cmdName*,  
    string *readType*,  
    out string *data*,  
    int *timeout*)

(c) ErrorCode **DirectSendRead**(  
    string *cmdName*,  
    string *readType*,  
    int *timeout*)

## パラメーター

*cmdName*

コマンド名

**SendDataFile**で定義したコマンド名を指定します。

*readType*

受信データタイプ名

受信するデータのタイプを下記から指定します。

ASB: ASB設定コマンドの応答のみを受信データに保存します。

Other: 上記以外のプリンターからの応答を受信データに保存します。

*data*

受信データバッファ

取得するデータを格納するバッファを指定します。

*timeout*

タイムアウト時間

本APIの成功を待つ時間を、ミリ秒(ms)で指定します。

有効範囲は3000～90000です。

3000未満の値を指定した場合は3000msに設定されます。

90000よりも大きい値を指定した場合は90000msに設定されます。

*readFlag*

受信動作フラグ

受信動作を指定するフラグを下記から指定します。

TRUE : 何らかのデータを受信するか、タイムアウトになるまで受信を続けます。

FALSE : 受信データサイズ分を受信するか、タイムアウトになるまで受信を続けます。

## 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 備考

- 本APIは**ResetPrinter**で中断できます。

- 受信データに0x02が含まれる場合、0x02は0x5fに変換されます。

## GetProperty

印刷設定内容(一部)を取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

印刷設定内容の取得には、PropertyIdに定義されたプロパティIDを用いる構文と、数値で指定する構文の2種類があります。

(a)の構文は、PropertyIdに定義されたプロパティIDを使用し、プロパティの印刷設定内容(一部)を取得します。

(b)の構文は、プロパティIDを数値で指定し、プロパティの印刷設定内容(一部)を取得します。

### 構文

```
(a) ErrorCode GetProperty(
    IntPtr devmode,
    PropertyId id,
    byte[] data,
    ref uint size )
```

```
(b) ErrorCode GetProperty(
    IntPtr devmode,
    byte id,
    byte[] data,
    ref uint size )
```

### パラメーター

*devmode*  
DevModeアドレス  
Devmodeアドレスを指定します。

*id*  
プロパティID  
取得する印刷設定内容のプロパティIDを指定します。  
PropertyIdに定義されたプロパティIDを使用する場合は、(a)を指定してください。  
プロパティIDを数値で指定する場合は、(b)を指定してください。  
指定可能な内容は、「6.8 プロパティID」を参照してください。

*data*  
取得データバッファ  
取得した印刷設定内容を格納するバッファを指定します。

*size*

#### 取得データサイズ

印刷設定内容を格納するバッファサイズを指定します。  
APIから制御が戻ると取得した印刷設定内容のデータサイズが格納されます。

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## SetProperty

印刷設定内容(一部)を変更します。

#### サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

印刷設定内容の変更は、PropertyIdに定義されたプロパティIDを用いる構文と、数値で指定する構文の2種類があります。

- (a) の構文は、PropertyIdに定義されたプロパティIDを使用し、印刷設定内容(一部)を変更します。
- (b) の構文は、プロパティIDを数値で指定し、印刷設定内容(一部)を変更します。

### 構文

(a) ErrorCode SetProperty(  
    IntPtr devmode,  
    PropertyId id,  
    byte[] data,  
    uint size )

(b) ErrorCode SetProperty(  
    IntPtr devmode,  
    byte id,  
    byte[] data,  
    uint size )

### パラメーター

*devmode*

DevModeアドレス

Devmodeアドレスを指定します。

*id*

プロパティID

変更する印刷設定内容のプロパティIDを指定します。  
PropertyIdに定義されたプロパティIDを使用する場合は、(a)を指定してください。  
プロパティIDを数値で指定する場合は、(b)を指定してください。  
指定可能な内容は、「6.8 プロパティID」を参照してください。

*data*

設定データバッファ

変更する印刷設定内容が格納されたバッファを指定します。

*size*

設定データサイズ

変更する印刷設定内容が格納されたバッファサイズを指定します。

## **戻り値**

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

## 4.3.6 イベント

### StatusCallback

応答されたプリンターステータスを通知します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	✓	✓

#### 構文

event StatusCallbackHandler **StatusCallback**

delegate void **StatusCallbackHandler**(  
ASB *status* )

#### パラメーター

*status*

プリンターステータス変数

プリンターステータスが格納されます。

プリンターステータスの内容は、「6.1 プリンターステータス一覧」を参照してください。

#### 備考

- プリンターステータスの通知を開始する場合は**SetStatusBack**を呼び出してください。
- **SetStatusBack**によりプリンターステータスの通知を開始すると、現在のプリンターステータスでイベントが発生します。
- プリンターステータスの通知を開始する前に、応答されたプリンターステータスを処理するイベントハンドラーを登録してください。
- イベントハンドラーを登録していない場合、**SetStatusBack**は失敗します。
- プリンターステータスを受信した場合でも、直前に受信したプリンターステータスと比べ変化がない場合、イベントは発生しません。
- プリンターの再接続を検出した場合のプリンターステータスは、その時点で最後に受信しているプリンターステータスになります。
- 本イベントを停止する場合は下記のAPIを呼び出してください。
  - **CloseMonPrinter**
  - **CancelStatusBack**
- 登録されたイベントハンドラー内から下記のAPIを呼び出すことはできません。
  - **CloseMonPrinter**
  - **SetStatusBack**
  - **CancelStatusBack**
- プリンターステータスの受信からイベント発生までの時間は保証されません。
- プリンターステータスの内容は、「6.1 プリンターステータス一覧」を参照してください。

## 4.4 ディスプレイ

---

.NET APIに実装されているAPIは下記の通りです。

- ネームスペース : SII.SDK.PrinterDevice
- クラス名 : DisplayAPI

ディスプレイのAPIについては、DSP-A01用「SII SDK for Windows アプリケーションプログラマーズガイド」を参照してください。

### 参考

- RP-G10、RP-E10はディスプレイはサポートしていません。



## 4.5 バーコードスキャナー

### 4.5.1 API 一覧

.NET APIに実装されているAPIは下記の通りです。

- ネームスペース : SII.SDK.PrinterDevice
- クラス名 : BarcodeScannerAPI

#### 共通 API

種別	API	機能概略
プロパティ	LastError	最後に実行したAPIのエラー値を取得します。
プロパティ	IsValid	OpenMonPrinterの呼び出し状態を取得します。
メソッド	OpenMonPrinter	.NET APIの使用を開始します。
メソッド	CloseMonPrinter	使用中の.NET APIを終了します。

#### 専用 API

種別	API	機能概略
メソッド	SetBarcodeDataBack	BarcodeDataCallbackイベントによるバーコードスキャナーの状態変化の通知を開始します。
メソッド	CancelBarcodeDataBack	BarcodeDataCallbackイベントによるバーコードスキャナーの状態変化の通知を停止します。
イベント	BarcodeDataCallback	応答されたバーコードスキャナーの状態変化を通知します。

#### 注意

- ◆ Bluetooth接続では、プリンターと未接続中の応答データの取得はできません。
- ◆ 推奨バーコードスキャナー、及びバーコードスキャナーの設定については、「RP-F10シリーズ サーマルプリンタ 取扱説明書」を参照してください。

## 4.5.2 共通プロパティ

### LastError

最後に呼び出したAPIのエラー値を取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	–	–

#### 構文

```
ErrorCode LastError { get; }
```

#### 初期値

SUCCESS

#### 備考

- エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### IsValid

OpenMonPrinterの呼び出し状態を取得します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	–	–

#### 構文

```
bool IsValid { get; }
```

#### 初期値

FALSE

## **備考**

- TRUE : **OpenMonPrinter**に成功した状態です。
- FALSE : **OpenMonPrinter**に成功していない状態です。

## 4.5.3 共通メソッド

### OpenMonPrinter

.NET APIの使用を開始します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	—	—

#### 構文

```
ErrorCode OpenMonPrinter(  
    OpenType type,  
    string name )
```

#### パラメーター

*type*

オープンタイプ

OpenType.TYPE\_BARCODE\_SCANNER (固定)

*name*

.NET APIを使用するプリンター名

プリンター名(フレンドリ名)を指定します。

#### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

#### 備考

- 1プロセスで.NET APIの各クラス合わせて最大8インスタンスまで同時にオープンできます。
- .NET APIを使用しなくなった場合は、必ず**CloseMonPrinter**を呼び出してください。
- プリンタードライバーの接続先は、USB、シリアル(BluetoothデバイスのCOMポートを含む)、TCP/IPのいずれかに設定してください。
- プリンターが未接続、または電源オフ状態でも本APIは成功します。
- バーコードスキャナーが未接続でも本APIは成功します。

## CloseMonPrinter

使用中の.NET APIを終了します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	–	–

### 構文

ErrorCode **CloseMonPrinter()**

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- APIに関連付けられたすべての設定及びデータは、本APIで破棄されます。

## 4.5.4 専用プロパティ

### SetBarcodeDataBack

BarcodeDataCallbackイベントによるバーコードスキャナーの状態変化の通知を開始します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	-	-

#### 構文

ErrorCode **SetBarcodeDataBack**()

#### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

#### 備考

- バーコードスキャナーの状態変化とは下記の通りです。
  - バーコードデータの受信
  - バーコードスキャナーの接続
  - バーコードスキャナーの切断
- 本APIによりバーコードスキャナーの状態変化の通知を開始すると、現在のバーコードスキャナーの接続状態でBarcodeDataCallbackイベントが発生します。
- BarcodeDataCallbackイベントにイベントハンドラーを登録していない場合、本APIは失敗します。
- すでにBarcodeDataCallbackイベントを開始済みで再度本APIを呼び出した場合も、直後のバーコードスキャナーの接続状態応答は行われます。
- 本APIで開始したBarcodeDataCallbackイベントは、下記のAPIで停止されます。
  - CloseMonPrinter
  - CancelBarcodeDataBack
- バーコードデータの受信からイベント発生までの時間は保証されません。

## CancelBarcodeDataBack

BarcodeDataCallbackイベントによるバーコードスキャナーの状態変化の通知を停止します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	-	-

### 構文

ErrorCode **CancelBarcodeDataBack**()

### 戻り値

エラーコードを返します。エラーコードの内容は、「5.1 エラーコード一覧」を参照してください。

### 備考

- **SetBarcodeDataBack**によりバーコードスキャナーの状態変化の通知が開始されていない場合でも本APIは成功します。

## 4.5.5 イベント

### BarcodeDataCallback

応答されたバーコードスキャナーの状態変化を通知します。

サポートプリンター

RP-F10	RP-G10	RP-E10
✓	–	–

#### 構文

```
event BarcodeDataCallbackHandler BarcodeDataCallback
```

```
delegate void BarcodeDataCallbackHandler(  
    BarcodeData barcodeData )
```

#### パラメーター

*barcodeData*

受信データバッファー

バーコードスキャナーからの応答データが格納されます。

格納される応答データの内容は、「6.9 バーコードデータ構造体」を参照してください。

#### 備考

- バーコードスキャナーの状態変化とは下記の通りです。
  - バーコードデータの受信
  - バーコードスキャナーの接続
  - バーコードスキャナーの切断
- バーコードスキャナーの状態変化の通知を開始する場合は**SetBarcodeDataBack**を呼び出してください。
- **SetBarcodeDataBack**によりバーコードスキャナーの状態変化の通知を開始すると、現在のバーコードスキャナーの接続状態でイベントが発生します。
- バーコードスキャナーの状態変化の通知を開始する前に、応答されたバーコードスキャナーの状態変化を処理するイベントハンドラーを登録してください。
- イベントハンドラーを登録していない場合、**SetBarcodeDataBack**は失敗します。
- バーコードスキャナーの接続状態を受信した場合でも、直前に受信したバーコードスキャナーの接続状態と比べ変化がない場合、イベントは発生しません。
- バーコードスキャナーの再接続を検出した場合のバーコードスキャナーの接続状態は、その時点で最後に受信しているバーコードスキャナーの接続状態になります。
- 本イベントを停止する場合は下記のAPIを呼び出してください。
  - **CloseMonPrinter**
  - **CancelBarcodeDataBack**
- 登録されたイベントハンドラー内から下記のAPIを呼び出すことはできません。
  - **CloseMonPrinter**



- **SetBarcodeDataBack**
- **CancelBarcodeDataBack**
- バーコードデータの受信からイベント発生までの時間は保証されません。

# 5章 エラーコード一覧

本章ではエラーコードについて記載します。

## 5.1 エラーコード一覧

主なエラーコードは下記の通りです。

マクロ定義(定数)	値	内容
SUCCESS	0	正常
ERR_TYPE	-10	オープンタイプのパラメーターエラーです。
ERR_OPENED	-20	指定したプリンターはすでにオープン済みです。
ERR_NO_PRINTER	-30	指定したプリンタードライバがありません。
ERR_HANDLE	-60	APIハンドルの値が不正です。
ERR_TIMEOUT	-70	タイムアウト、またはビジー状態です。
ERR_ACCESS	-80	ネットワーク上で到達できない、もしくはアクセス中にサービスが停止されるなどの理由でプリンターにアクセスできません。
ERR_PARAM	-90	パラメーターが不正です。
ERR_NOT_SUPPORT	-100	サポートしていない機能です。
ERR_OFFLINE	-110	切断状態、またはプリンターがオフラインです。
ERR_NON_TARGETTED_DRIVER	-120	指定したプリンタードライバは対象外です。
ERR_DISK_FULL	-170	プリンターがビジー状態です。
ERR_ENTRY_OVER	-190	処理の最大使用数を超過しています。
ERR_EXIST	-210	すでに存在するモジュールを呼び出しています。
ERR_NOT_FOUND	-220	ファイルが見つかりません。または登録されていません。
ERR_WORKAREA_NO_MEMORY	-260	メモリの領域指定が不足しています。
ERR_WORKAREA_FAILED	-280	メモリの確保ができません。

マクロ定義(定数)	値	内容
ERR_EXEC_FUNCTION	-310	別のスレッド、またはプロセスで使用中のため、その機能は使用できません。
ERR_SPL_NOT_EXIST	-350	スプーラーサービス、またはSII Printer Softwareサービスが起動していません。
ERR_LOCKED	-1000	プリンターがロックされています。
ERR_UNLOCKED	-1010	プリンターがロックされていないのに、UnlockPrinterが実行されました。
ERR_INVALID_DATA	-1020	不正なデータが指定されています。
ERR_READ_FAULT	-1030	プリンターからの応答がありません。
ERR_WRITE_FAULT	-1040	プリンターにデータを送信できません。
ERR_CANCELLED	-1050	機能が取り消されました。
ERR_UNKNOWN_PORT	-1070	対象外のポートです。
ERR_INVALID_PRINTER_STATE	-1080	プリンターステータスが正常ではありません。
ERR_BAD_ENVIRONMENT	-1090	プリンタードライバのインストールが正常に行われていない可能性があります。

## 6章 引数情報

本章では引数について記載します。

### 6.1 プリンターステータス一覧

プリンターステータスの応答内容の対応ビットは下記の通りです。サンプルプログラムのプリンターステータス処理も参考にしてください。

プリンターステータス	対応ビット		内容
電圧エラー	ASB_VP_ERR	0x00000000	電圧エラーなし
		0x00000001	電圧エラーあり
RP-F10、RP-G10 ハードウェアエラー	ASB_HARDWARE_ERR	0x00000000	ハードウェアエラーなし
		0x00000002	ハードウェアエラーあり
RP-E10 ヘッドエラー/初期化電圧エラー	ASB_HEAD_ERR	0x00000000	ヘッドエラー/初期過電圧エラーなし
		0x00000002	ヘッドエラー/初期過電圧エラーあり
ヘッド温度エラー	ASB_HEAD_TEMPERATUR_ERR	0x00000000	ヘッド温度エラーなし
		0x00000004	ヘッド温度エラーあり
オートカッターエラー	ASB_AUTOCUTTER_ERR	0x00000000	オートカッターエラーなし
		0x00000008	オートカッターエラーあり
紙無しエラー	ASB_RECEIPT_END	0x00000000	紙無しエラーなし
		0x00000010	紙無しエラーあり
RP-E10 ペーパーニアエンド	ASB_RECEIPT_NEAR_END	0x00000000	ペーパーニアエンドなし
		0x00000020	ペーパーニアエンドあり
RP-E10 マーク検出時紙ジャムエラー	ASB_MARK_PAPER_JAM_ERR	0x00000000	マーク検出時紙ジャムエラーなし
		0x00000040	マーク検出時紙ジャムエラーあり
カバーオープンエラー	ASB_COVER_OPEN	0x00000000	ペーパーカバーが閉じている
		0x00000080	ペーパーカバーが開いている

プリンターステータス	対応ビット		内容
フィードスイッチの状態	ASB_PAPER_FEED	0x00000000	フィードスイッチの状態="Off"
		0x00000100	フィードスイッチの状態="On"
紙送り状態	ASB_NOW_PRINTING	0x00000000	停止
		0x00000400	動作中
復帰待ち状態	ASB_RETURN_WAITING	0x00000000	-
		0x00000800	復帰待ち状態
ドロワセンサーの状態	ASB_DRAWER_KICK	0x00000000	ドロワセンサーの状態="L"
		0x00000800	ドロワセンサーの状態="H"
フラッシュメモリ書き換え中	ASB_FLASH_MEMORY_REWRITING	0x00000000	-
		0x00010000	フラッシュメモリ書き換え中
RP-E10 周辺機器選択	ASB_PERIPHERAL_EQUIPMENT	0x00000000	プリンター
		0x00020000	その他
自動復帰エラー	ASB_AUTORECOVER_ERR※1	0x00000000	自動復帰エラーなし
		0x20000000	自動復帰エラーあり
復帰不可能エラー	ASB_UNRECOVER_ERR※1	0x00000000	復帰不可能エラーなし
		0x40000000	復帰不可能エラーあり
応答なし	ASB_NO_RESPONSE※1	0x00000000	プリンター応答
		0x80000000	未接続、または通信エラー

※1: ASB設定コマンドの応答に対し、拡張されたステータスです。

## 6.2 カウンターID

カウンターIDと、その内容は下記の通りです。

カウンターID		内容	ResetCounterによる 初期化
.NET API	.NET API Win32 API		
ROLL_FEED_LINES	20	紙送りドットライン数 (100ドットライン単位)	可
ROLL_HEAD_CHARGE	21	サーマルヘッド通電回数 (100ドットライン単位)	可
PAPER_CUT	50	オートカッター駆動回数	可
OPERATION_TIME	70	製品の駆動時間(分単位)	可
ROLL_FEED_LINES※1	148	紙送りドットライン数(積算) (100ドットライン単位)	不可
ROLL_HEAD_CHARGE※1	149	サーマルヘッド通電回数(積算) (100ドットライン単位)	不可
PAPER_CUT※1	178	オートカッター駆動回数(積算)	不可
OPERATION_TIME※1	198	製品の駆動時間(積算) (分単位)	不可

※1: .NET APIの場合、積算カウンターのカウンター値の取得はGetCounterのtypeで指定します。APIの詳細は、.NET APIのGetCounterを参照してください。

## 6.3 プリンターID

プリンターIDと、その内容は下記の通りです。

プリンターID		内容	応答形式
.NET API	.NET API Win32 API		
DEVICE	1	機種ID	数値(1バイト) RP-F10、RP-G10:0x25 RP-E10:0x1a
TYPE	2	タイプID	数値(1バイト) 「6.4 タイプID」を参照
ROM_VER	3	ROMバージョンID	数値(1バイト)
FW_VER_MAIN	65	ファームウェアバージョン(main)	ASCII文字列 “x.xx.xx”
MFG	66	メーカー名	ASCII文字列 “Seiko Instruments Inc.”
MDL	67	モデル名	ASCII文字列 RP-F10シリーズ:“SII RP-F10 Series.” RP-G10シリーズ:“SII RP-G10 Series.” RP-E10シリーズ:“SII RP-E10 Series.”
FONT	69	多国語フォントの種類	ASCII文字列 “KANJI JAPANESE”
FW_VER_BOOT	97	ファームウェアバージョン(boot)	ASCII文字列 “x.xx.xx”
FW_SUM_BOOT	98	ファームウェアチェックサム(boot)	数値(2バイト)
FW_SUM_MAIN	99	ファームウェアチェックサム(main)	数値(2バイト)
FW_SUM	100	ファームウェアチェックサム(main + boot)	数値(2バイト)

## 6.4 タイプID

タイプIDの応答内容の対応ビットと、その内容は下記の通りです。

タイプIDとはプリンターコマンド「プリンタIDの送信」によって取得されるプリンター情報の一部です。

対応ビット	内容	
	RP-F10 RP-G10	RP-E10
0x01	0 : 58 mm 1 : 80 mm	0: マルチバイトコード対応なし 1: マルチバイトコード対応あり
0x02	予約済み(1に固定)	0: オートカッターなし 1: オートカッターあり
0x04	未定義(0に固定)	
0x08	未定義(0に固定)	
0x10	予約済み(1に固定)	未定義(0に固定)

対応ビット	内容	
	RP-F10 RP-G10	RP-E10
0x20	予約済み(1に固定)	未定義(0に固定)
0x40	未定義(0に固定)	
0x80	未定義(0に固定)	

取得できる応答内容は上記の値を合計した値です。

例：RP-F10、80mm選択の場合の応答値 = 0x33

## 6.5 対象ドロワ

対象ドロワと、その内容は下記の通りです。

対象ドロワ		内容
.NET API	.NET API Win32 API	
DRAWER_1	1	ドロワ1を駆動
DRAWER_2	2	ドロワ2を駆動

## 6.6 ドロワ駆動時間

ドロワ駆動時間と、その内容は下記の通りです。

ドロワ駆動時間		内容
.NET API	.NET API Win32 API	
TIME_100	1	100 ms ドロワを駆動
TIME_200	2	200 ms ドロワを駆動
TIME_300	3	300 ms ドロワを駆動
TIME_400	4	400 ms ドロワを駆動
TIME_500	5	500 ms ドロワを駆動
TIME_600	6	600 ms ドロワを駆動
TIME_700	7	700 ms ドロワを駆動
TIME_800	8	800 ms ドロワを駆動

ドロワの制御時間については、お客様がご利用されているドロワの仕様に従ってください。

## 6.7 プリンターの機能

プリンターの機能と、その内容は下記の通りです。

プリンターの機能		内容	引数1	引数2
.NET API	.NET API Win32 API			
EXT_BUZZER	1	外部ブザーを鳴動	ブザーパターン※2 0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4	ブザー鳴動回数※2 1 ~ 255(回)

※1: ブザーパターンについては「技術説明書」を参照してください。

※2: 外部ブザーの鳴動は下記のいずれかの条件で停止します。

- ・設定したブザーの回数分の鳴動
- ・カバーオープン
- ・プリンターコマンド「外付けブザーの停止」の実行

## 6.8 プロパティID

### 注意

- ◆ SetPropertyを呼び出す場合、ロゴキーコード及びカスタムコマンド以外はデータサイズを1バイトで指定してください。

プロパティIDと、その内容は下記の通りです。

[ ]カッコ内はオプション設定において当該処理が行われるタイミングを示します。オプション設定については、「SII Printer Driver for Windows 取扱説明書」の「3.3.7 オプションの設定」を参照してください。

プロパティID		内容							
.NET API	.NET API Win32 API								
INIT	1	初期化	0 : する 1 : しない						
SPEED	2	速度	<table><tr><th>プリンター</th><th>内容</th></tr><tr><td>RP-F10 RP-G10</td><td>0 : 高速 1 : 中速(品質) 2 : 中速(静音)</td></tr><tr><td>RP-E10</td><td>0 : 中速(静音)<sup>※1</sup> 1 : 低速 2 : 中速(品質) 3 : 高速</td></tr></table>	プリンター	内容	RP-F10 RP-G10	0 : 高速 1 : 中速(品質) 2 : 中速(静音)	RP-E10	0 : 中速(静音) <sup>※1</sup> 1 : 低速 2 : 中速(品質) 3 : 高速
			プリンター	内容					
			RP-F10 RP-G10	0 : 高速 1 : 中速(品質) 2 : 中速(静音)					
RP-E10	0 : 中速(静音) <sup>※1</sup> 1 : 低速 2 : 中速(品質) 3 : 高速								



プロパティID		内容	
.NET API	.NET API Win32 API		
MARGIN	3	余白	0 : 最小余白 1 : 上余白最小 2 : 下余白最小 3 : 最大余白
DENSITY	4	濃度(パーセント)	70 ~ 130
DIRECTION	5	方向	0 : 順 1 : 逆
REDUCTION	6	縮小	0 : 縮小しない RP-F10、RP-G10 20 ~ 100 : 指定倍率(パーセント) RP-E10 25 ~ 100 : 指定倍率(パーセント)
CUT	7	用紙カット	0 : カットしない 1 : フルカット(ジョブ単位) 2 : パーシャルカット(ジョブ単位) 3 : フルカット(ページ単位) 4 : パーシャルカット(ページ単位) 5 : パーシャルカット(ページ間)
MARK_FEED	8	マーク紙の頭出し※2	0 : しない 1 : ページ単位 2 : ジョブ単位
CUT_FEED	9	カット位置までフィード	0 : する 1 : しない
START_DOC_LOGO	10	[印刷開始] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
START_DOC_LOGO_KEY	11	[印刷開始] ロゴキーコード (2バイト)※3	RP-F10、RP-G10 各バイト 0x20~0x39 RP-E10 各バイト 0x30~0x39
START_DOC_DRAWER	12	[印刷開始] ドロワ※4	0 : 利用しない 1 : ドロワ1を利用 2 : ドロワ2を利用
START_DOC_DRAWER_ON	13	[印刷開始] ON時間(x 2 ms)	1 ~ 255
START_DOC_DRAWER_OFF	14	[印刷開始] OFF時間(x 2 ms)	1 ~ 255
START_DOC_CMD	15	[印刷開始] カスタムコマンド (128バイト)	コマンドデータ
START_DOC_BACK_FEED	16※5	[印刷開始] 紙送り(バックフィード)(dot)	RP-F10、RP-G10 0 ~ -60 RP-E10 0 ~ -74

プロパティID		内容	
.NET API	.NET API Win32 API		
START_DOC_FEED	17※5	[印刷開始] 紙送り(フィード) (dot)	0 ~ 255
START_DOC_BUZZER	18	[印刷開始] ブザー※2	0 : ON 1 : OFF
START_DOC_EXT_BUZZER	110	[印刷開始] 外部ブザー※6	0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4 255: OFF
START_DOC_EXT_BUZZER_RPT	111	[印刷開始] 外部ブザー 鳴動回数※6※7	1 ~ 255
START_PAGE_LOGO	20	[ページ開始] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
START_PAGE_LOGO_KEY	21	[ページ開始] ロゴキーコード (2バイト)※3	RP-F10、RP-G10 各バイト 0x20~0x39 RP-E10 各バイト 0x30~0x39
START_PAGE_CMD	25	[ページ開始] カスタムコマンド (128バイト)	コマンドデータ
START_PAGE_BACK_FEED	26※5	[印刷開始] 紙送り(バックフィード) (dot)	RP-F10、RP-G10 0 ~ -60 RP-E10 0 ~ -74
START_PAGE_FEED	27※5	[ページ開始] 紙送り(フィード) (dot)	0 ~ 255
START_PAGE_BUZZER	28	[ページ開始] ブザー※2	0 : ON 1 : OFF
START_PAGE_EXT_BUZZER	120	[ページ開始] 外部ブザー※6	0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4 255: OFF
START_PAGE_EXT_BUZZER_RPT	121	[ページ開始] 外部ブザー 鳴動回数※6※7	1 ~ 255
END_PAGE_LOGO	30	[ページ終了] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ

プロパティID		内容	
.NET API	.NET API Win32 API		
END_PAGE_LOGO_KEY	31	[ページ終了] ロゴキーコード (2バイト) <sup>※3</sup>	RP-F10、RP-G10 各バイト 0x20～0x39 RP-E10 各バイト 0x30～0x39
END_PAGE_CMD	35	[ページ終了] カスタムコマンド (128バイト)	コマンドデータ
END_PAGE_BACK_FEED	36 <sup>※5</sup>	[印刷開始] 紙送り(バックフィード)(dot)	RP-F10、RP-G10 0 ～ -60 RP-E10 0 ～ -74
END_PAGE_FEED	37 <sup>※5</sup>	[ページ終了] 紙送り(フィード) (dot)	0 ～ 255
END_PAGE_BUZZER	38	[ページ終了] ブザー <sup>※2</sup>	0 : ON 1 : OFF
END_PAGE_EXT_BUZZER	130	[ページ終了] 外部ブザー <sup>※6</sup>	0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4 255: OFF
END_PAGE_EXT_BUZZER_RPT	131	[ページ終了] 外部ブザー 鳴動回数 <sup>※6※7</sup>	1 ～ 255
END_DOC_LOGO	40	[印刷終了] ロゴ	0 : なし 1 : 左寄せ 2 : 中央 3 : 右寄せ
END_DOC_LOGO_KEY	41	[印刷終了] ロゴキーコード (2バイト) <sup>※3</sup>	RP-F10、RP-G10 各バイト 0x20～0x39 RP-E10 各バイト 0x30～0x39
END_DOC_DRAWER	42	[印刷終了] ドロワ <sup>※4</sup>	0 : 利用しない 1 : ドロワ1を利用 2 : ドロワ2を利用
END_DOC_DRAWER_ON	43	[印刷終了] ON時間(x 2 ms)	1 ～ 255
END_DOC_DRAWER_OFF	44	[印刷終了] OFF時間(x 2 ms)	1 ～ 255
END_DOC_CMD	45	[印刷終了] カスタムコマンド (128バイト)	コマンドデータ

プロパティID		内容							
.NET API	.NET API Win32 API								
END_DOC_BACK_FEED	46※5	[印刷開始] 紙送り(バックフィード)(dot)	RP-F10、RP-G10 0 ~ -60 RP-E10 0 ~ -74						
END_DOC_FEED	47※5	[印刷終了] 紙送り(フィード)(dot)	0 ~ 255						
END_DOC_BUZZER	48	[ページ終了] ブザー※2	0 : ON 1 : OFF						
END_DOC_EXT_BUZZER	140	[印刷終了] 外部ブザー※6	0 : パターン1 1 : パターン2 2 : パターン3 3 : パターン4 255: OFF						
END_DOC_EXT_BUZZER_RPT	141	[印刷終了] 外部ブザー 鳴動回数※6※7	1 ~ 255						
PAPER_SIZE	50	用紙サイズ	<table><tr><th>プリンター</th><th>内容</th></tr><tr><td>RP-F10 RP-G10</td><td>0 : Letter 1 : A4 2 : 80 mm (72×3276 mm) 3 : 58 mm (54×3276 mm) 4 ~ 255 : 上記以外の用紙※8</td></tr><tr><td>RP-E10</td><td>0 : Letter 1 : A4 2 : 80 mm 3 : 58 mm 4 ~ 255 : 上記以外の用紙※8</td></tr></table>	プリンター	内容	RP-F10 RP-G10	0 : Letter 1 : A4 2 : 80 mm (72×3276 mm) 3 : 58 mm (54×3276 mm) 4 ~ 255 : 上記以外の用紙※8	RP-E10	0 : Letter 1 : A4 2 : 80 mm 3 : 58 mm 4 ~ 255 : 上記以外の用紙※8
プリンター	内容								
RP-F10 RP-G10	0 : Letter 1 : A4 2 : 80 mm (72×3276 mm) 3 : 58 mm (54×3276 mm) 4 ~ 255 : 上記以外の用紙※8								
RP-E10	0 : Letter 1 : A4 2 : 80 mm 3 : 58 mm 4 ~ 255 : 上記以外の用紙※8								
ORIENTATION	51	印刷の向き	0 : 縦 1 : 横						
COLOR_MODE	52	カラー印刷モード	0 : システム※9 1 : ドライバー※9						
WATERMARK	53	ウォーターマーク	0 : なし 1 : 左上 2 : 中央上 3 : 右上 4 : 左 5 : 中央 6 : 右 7 : 左下 8 : 中央下 9 : 右下						

プロパティID		内容			
.NET API	.NET API Win32 API				
PRESET	60※10	プリセット	プリンター	内容	
			RP-F10 RP-G10	0 : 80 mmレシート設定 1 : 58 mmレシート設定 4 : A4 → 80 mm縮小設定 5 : A4 → 58 mm縮小設定 6 : ユーザー設定※8	
				RP-E10	0 : 80 mmレシート設定 1 : 58 mmレシート設定 2 : 80 mmマーク紙設定 3 : 58 mmマーク紙設定 4 : A4 → 80 mm縮小設定 5 : A4 → 58 mm縮小設定 6 : ユーザー設定※8

※1: プリンターのF/Wバージョンが1.02以前の場合は、「0: 中速(静音)」以外を選択してください。

※2: RP-E10のみサポートします。

※3: 格納順は、上位桁、下位桁の順です。

※4: 印刷開始と印刷終了の両方でドロワを駆動することはできません。

※5: 同じタイミングのフィードは、最後の設定が有効になります。

※6: RP-E10はサポートしていません。

※7: 外部ブザーの鳴動は下記のいずれかの条件で停止します。

- ・設定したブザーの回数分の鳴動
- ・カバーオープン
- ・プリンターコマンド「外付けブザーの停止」の実行

※8: 設定することはできません。取得のみ行えます。

※9: カラー印刷モードの詳細は、「SII Printer Driver for Windows 取扱説明書」の「3.2 用紙・品質」を参照してください。

※10: 本項目を指定した場合、他のプロパティIDの一部の内容が無視されます。プリセットの詳細は、「SII Printer Driver for Windows 取扱説明書」の「3.3.4 プリセットの利用」を参照してください。

## 6.9 バーコードデータ構造体

バーコードデータ構造体と、その内容は下記の通りです。

### 6.9.1 Win32 API

```
ST_BARDATA
{
    INT io_rlen,
    LPBYTE o_rdata,
    DWORD Reserved
}

ST_BARDATA, *PST_BARDATA;
```

#### パラメーター

*io\_rlen*

受信データサイズ

コールバック関数が呼び出されると下記の受信データが格納されます。

<i>io_rlen</i>	内容
0より大きい	受信されたバーコードデータのサイズ
0	バーコードスキャナーが接続している状態
-1	バーコードスキャナーが接続していない状態

*o\_rdata*

受信データバッファー

バーコードスキャナーから応答されたデータ配列のアドレスを返します。

*io\_rlen*が0か-1の場合は、*o\_rdata*のデータ配列に0が入ります。

*o\_rdata*の受信データは、次にユーザーが登録したコールバック関数が呼び出されたタイミングで削除されます。値を保持するために、受信データバッファーのデータをコピーしてください。

*Reserved*

予約

予約です。

## 6.9.2 .NET API

```
class BarcodeData
{
    int length;
    byte[] data;
    uint reserved
}
```

### パラメーター

*length*

受信データサイズ

コールバック関数が呼び出される下記の受信データが格納されます。

<i>length</i>	内容
0より大きい	受信されたバーコードデータのサイズ
0	バーコードスキャナーが接続している状態
-1	バーコードスキャナーが接続していない状態

*data*

受信データバッファー

バーコードスキャナーから応答されたデータ配列を返します。

*length*が0か-1の場合は、*data*のデータ配列は空になります。

*reserved*

予約

予約です。

# 7章 コマンド定義ファイル

## 7.1 概要

コマンド定義ファイルとは、送信データを決められたフォーマットで記述し、ANSI形式及びUNICODE形式で保存されたテキストファイル(\*.txt)です。

## 7.2 フォーマット

コマンド定義ファイルのフォーマットを説明します。

Command definition

[Command name 1] = [Transmission data 1]#[Comment 1]  
[Command name 2] = [Transmission data 2]#[Comment 2]  
[Command name 3] = [Transmission data 3]#[Comment 3]  
:  
:

名称	説明
Command definition コマンド定義	フォーマットに沿って記述したコマンド名単位の水データです。 必ず1コマンド定義に対して1行で記述してください。
Command name コマンド名	<b>DirectSendRead</b> で指定する任意のコマンド名を記述します。 <ul style="list-style-type: none"><li>・ “=”、“#”以外のASCII文字を使用してください。ASCII文字以外を記述した場合、その文字だけ無視されます。</li><li>・ 大文字、小文字は区別されます。</li><li>・ 登録可能なコマンド名は最大33バイトです。34バイト以上記述した場合、33バイトで登録され、34バイト以降の文字は無視されます。</li><li>・ 1個のコマンド定義ファイルの中で同じコマンド名を記述している場合、最初のコマンド名が登録されます。すでに登録されているコマンド名を指定した場合は無視されます。</li><li>・ コマンド名と送信データの間には“=”を記述してください。</li></ul>



名称	説明
Transmission data 送信データ	<p>プリンターに送信する、プリンターコマンド及びデータを記述します。</p> <ul style="list-style-type: none"> <li>送信データに応答を無効にするASB設定コマンドを含めないでください。プリンターステータスを取得するAPIが正常に動作しなくなります。</li> <li>送信データにバイナリデータを指定する場合は、データは2桁の16進数で記述してください。各データの間には半角スペースを挿入してください。 バイナリデータにはプリンターコマンド「プリンタの初期化」以外のプリンターを初期化するプリンターコマンドを含めないでください。プリンターの初期化については、「技術説明書」を参照してください。ハードウェアリセットする場合は、<b>Reset</b>、または<b>ResetPrinter</b>を実行してください。</li> <li>送信データに文字列を指定する場合は、ASCII文字で記述してください。文字列を指定する場合は、""で囲います。</li> <li>送信データのデータサイズは最大10240バイトです。ただし、送信データを文字列で指定した場合のサイズは、指定した文字列をバイナリデータに変換した後のサイズです。 後述のコマンド定義例の送信データサイズは下記の通りです。 <ul style="list-style-type: none"> <li>CmdName_1 : 3バイト</li> <li>CmdName_2 : 4バイト</li> <li>CmdName_3 : 3バイト</li> <li>CmdName_4 : 6バイト</li> </ul> </li> </ul>
Comment コメント	<p>コマンド定義の説明などを記述します。</p> <ul style="list-style-type: none"> <li>コメントの先頭に"#"を記述してください。</li> <li>記述する文字の制限はありません。</li> <li>コメントは省略可能です。</li> </ul>

#### コマンド定義例:

CmdName\_1="SII"#Comments1

CmdName\_2=53 49 49 0A

CmdName\_3=1D 56 00#Feed the paper to cut position.

CmdName\_4=1B 40 "SII" 0A#Initialize the printer, the strings "SII" and line feed.

#### 参考

- コマンド定義ファイルのファイルサイズは、最大4GBです。
- 登録可能なコマンド定義数は、システムの使用可能メモリによって異なります。

## 7.3 利用方法

---

コマンド定義ファイルの利用方法を説明します。

1. コマンド定義ファイルを作成してください。
2. コマンド定義ファイルを**SendDataFile**でSDK内部のメモリに登録してください。
3. コマンド名を**DirectSendRead**で指定すると、送信データの内容が実行されます。

### 注意

- ◆ 登録したコマンド定義は、**CloseMonPrinter**で破棄されます。
- ◆ コマンド定義ファイル登録後に、新たなコマンド定義ファイルを登録すると、登録済みのコマンド定義は上書きされます。