

## ビルバリデータ、コインメックインターフェース BVCM-USB-01

本インターフェースは一般的な直列伝送方式のビルバリデータ、コインメックデバイス（以後、単にデバイスと呼びます）とUSBとを接続します。ユーザは専用DLLによるAPI関数を用いて簡単にデバイスを制御することができます。

インターフェース1台につきビルバリデータ1台、コインメック1台まで接続可能です。さらに多くのデバイスを接続したい場合はインターフェースを複数接続することにより対応できます。

### 1. 交信の種類

本インターフェースを用いてPCからデバイスと通信する場合、常に交信の主導権はPC側にあり、デバイスがPCからのコマンドに答えることで交信が成立します。

直列伝送方式における交信形式には次にあげる3種類があり、すべてのコマンドはこのうちの適合する交信形式を用いて発行されます。また、交信中に起こった伝送エラーなどに関して、データの再送を促す取り決めなどもありますが、本インターフェースを利用する場合、ユーザはこれを意識する必要はありません。これらタイミング的に厳しく、複雑な部分はインターフェース上のファームウェアが処理するため、ユーザは次章に述べるAPIを通してコマンドを送り、結果を受け取るだけの処理となります。

#### 指令交信

デバイスに対して何か特定の操作をする場合の交信（コマンドのみでデータがない場合）で、成功か失敗の情報が返されます。

#### 送信データ交信

デバイスに対してデータを送信する場合の交信で、成功か失敗の情報が返されます。

#### 受信データ交信

デバイスからデータを取得する場合の交信で、コマンドを送信するとデータおよび、成功、失敗の情報が返されます。

## 2 . A P I関数

インターフェースには専用D L L (bvcm.dll)が添付され、ユーザはダイナミックロードあるいは静的リンクすることによってA P I関数を使用することができます。D L Lの構成ファイルとして以下の3ファイルが供給されます。

- ・ bvcm.dll (D L L本体)
- ・ bvcm.h (ヘッダーファイル)
- ・ bvcm.lib (静的リンク用ライブラリ)

A P I関数を用いて複数のプログラムから同一デバイスにアクセスした場合、片方の実行が終了するまでは、もう片方のプログラムに対してはデバイス使用中である旨のステータスが返ります。したがってユーザはこのステータスをハンドルしてプロセス間の同期をとる必要があります。

最初にユーザが行うことはインターフェースをオープンすることです。正常にオープンされれば、以後、そのインターフェースに対して、通信を行うことができるようになります。

## 2 - 1 .関数リファレンス

### 2 - 1 - 1 .BVCM\_ScanInterface

実装されているインターフェースをスキャンし、その台数を返します。

#### 宣言

```
BVCM_STATUS BVCM_ScanInterface (unsigned int* puiNumber)
```

#### 引数

puiNumber インターフェース数を受け取る変数へのポインタ

#### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

#### 説明

本関数が正常に終了した場合、BVCM\_OK が返り、puiNumber で示される変数にはインターフェース数が代入されています。

## 2 - 1 - 2 . BVCM\_OpenInterface

インターフェースをオープンし、使用可能な状態にします。

### 宣言

```
BVCM_STATUS BVCM_OpenInterface (unsigned int uiInterface, BVCM_HANDLE *bvcmHandle)
```

### 引数

uiInterface    インターフェース番号  
bvcmHandle    インターフェースハンドルへのポインタ

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

uiInterface はオープンすべきインターフェースの番号を指定します。0 から始まりインターフェース台数未満の整数です。

bvcmHandle は BVCM\_HANDLE 型変数へのポインタで、本関数が正常に終了すると BVCM\_OK が返り、bvcmHandle で示されるインターフェースハンドルにハンドル値が代入されています。

## 2 - 1 - 3 . BVCN\_CloseInterface

インターフェースをクローズします。

### 宣言

```
BVCN_STATUS BVCN_CloseInterface (BVCN_HANDLE bvcnHandle)
```

### 引数

bvcnHandle インターフェースハンドル

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

bvcnHandle で指定されたインターフェースをクローズします。

本関数が正常に終了すると BVCN\_OK が返ってきます。

## 2 - 1 - 4 . BVCM\_Standby

デバイスを待機状態にします。

### 宣言

```
BVCM_STATUS BVCM_Standby (BVCM_HANDLE bvcmHandle, int iTarget)
```

### 引数

bvcmHandle インターフェースハンドル  
iTarget ターゲット

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

iTarget はビルバリデータとコインメックのどちらにコマンドを送るか、選択します。iTarget に BVCM\_BILL を設定すればビルバリデータに、BVCM\_COIN を設定すればコインメックに対し動作します。

本関数が成功すると通常はACKコードがステータスとして返ってきます。

## 2 - 1 - 5 . BVCM\_GetAllData

デバイスから全データを取り出します。

### 宣言

```
BVCM_STATUS BVCM_GetAllData (BVCM_HANDLE bvcmHandle, int iTarget, PCHAR pcBuffer)
```

### 引数

bvcmHandle インターフェースハンドル  
iTarget ターゲット  
pcBuffer データを受け取るためのバッファへのポインタ

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

iTarget はビルバリデータとコインメックのどちらにコマンドを送るか、選択します。iTarget に BVCM\_BILL を設定すればビルバリデータに、BVCM\_COIN を設定すればコインメックに対し動作します。

本関数実行後、pcBuffer で示されるバッファにはデバイスから取得したデータが入ります。データフォーマットについては、デバイスからのデータがそのまま入りますので、それぞれの機器メーカーの通信仕様書を参照して下さい。

本関数が成功すると通常はACKコードがステータスとして返ってきます。

## 2 - 1 - 6 . BVCN\_GetData

デバイスから前回のデータ取得以降、変化のあったデータを取り出します。

### 宣言

```
BVCN_STATUS BVCN_GetData (BVCN_HANDLE bvcmHandle, int iTarge, PCHAR pcBuffer)
```

### 引数

bvcmHandle インターフェースハンドル  
iTarge ターゲット  
pcBuffer データを受け取るためのバッファへのポインタ

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

iTarge はビルバリデータとコインメックのどちらにコマンドを送るか、選択します。iTarge に BVCN\_BILL を設定すればビルバリデータに、BVCN\_COIN を設定すればコインメックに対し動作します。

本関数実行後、pcBuffer で示されるバッファにはデバイスから取得したデータが入ります。データフォーマットについては、デバイスからのデータがそのまま入りますので、それぞれの機器メーカーの通信仕様書を参照して下さい。

本関数が成功すると通常はACKコードがステータスとして返ってきます。



## 2 - 1 - 7 . BVCM\_PutData

デバイスにデータを送信します。

### 宣言

```
BVCM_STATUS BVCM_PutData (BVCM_HANDLE bvcmHandle, int iTarget, PCHAR pcBuffer)
```

### 引数

bvcmHandle インターフェースハンドル  
iTarget ターゲット  
pcBuffer 送信データを格納したバッファへのポインタ

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

iTarget はビルバリデータとコインメックのどちらにコマンドを送るか、選択します。iTarget に BVCM\_BILL を設定すればビルバリデータに、BVCM\_COIN を設定すればコインメックに対し動作します。

本関数実行前に、pcBuffer で示されるバッファに正しいデータフォーマットでデータフレームを作成しておく必要があります。データフォーマットについてはデバイスによって異なる場合がありますので、それぞれの機器メーカーの通信仕様書を参照して下さい。

本関数が成功すると通常はACKコードがステータスとして返ってきます。

## 2 - 1 - 8 . BVCN\_LineReset

インターフェースに接続されているデバイスをリセットします。

### 宣言

```
BVCN_STATUS BVCN_Standby (BVCN_HANDLE bvcnHandle)
```

### 引数

bvcnHandle インターフェースハンドル

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

本関数実行後、すぐに制御は戻ってきますが実際のリセットには数秒間かかります。その間は他のAPIによりアクセスすることはできません。

本関数が正常に終了すると、BVCN\_OK が返ってきます。

## 2 - 2 . 関数ステータス

A P I関数の戻り値である B V C M \_ S T S T U S は以下の通り定義されています。

### **BVCM\_OK**

デバイスに対するコマンド発行以外の関数の場合、正常に関数が終了するとこの値が返ってきます。

### **BVCM\_INVALID\_HANDLE**

インターフェースハンドルが間違っています。

### **BVCM\_IO\_ERROR**

インターフェースとの通信でエラーが発生しました。

### **BVCM\_INVALID\_PARAMETER**

引数が間違っています。

### **BVCM\_NO\_RESPONSE**

デバイスが応答しません。

### **BVCM\_OTHER\_ERROR**

その他のエラー

### **BVCM\_NAK (ACKコード)**

デバイスと正常に交信できましたが、N A K 応答が返ってきました。

### **BVCM\_ACK1 (ACKコード)**

デバイスと正常に交信でき、A C K 1 応答が返ってきました。

### **BVCM\_ACK2 (ACKコード)**

デバイスと正常に交信でき、A C K 2 応答が返ってきました。

### **BVCM\_ACK3 (ACKコード)**

デバイスと正常に交信でき、A C K 3 応答が返ってきました。

### **BVCM\_ACK4 (ACKコード)**

デバイスと正常に交信でき、A C K 4 応答が返ってきました。

### **BVCM\_BUSY**

デバイスは現在、コマンドを受けられません。