デバイスアートツールキット

制御モジュール

MCD05

ソフトウェア マニュアル Rev. 1.0

2015 年 8 月

アークデバイス

目次

1. 概要

DATK（Device Art Tool Kit）の制御モジュール MCD05 にはブートローダプログラム MCD05_Boot，ユーザプログラムとして制御プログラム MCD05_Ctrl が出荷時に書きこまれています．ブートローダは自己書き換え機能を持ちユーザプログラムの書き換えが可能です．MCD05 ではまずブートローダが起動します．自己書き換えの指令がなければ，次にユーザプログラムである制御プログラム MCD05_Ctrl が呼び出されます．

MCD05 はシリアル通信モジュール（SC02 等）と接続し，パソコンからはシリアル通信モジュールを経由して MCD05 を制御します．

MCD05 をパソコンから通信制御するソフトウェアを制作するには，制御プログラム MCD05_Ctrl に対応したクラス MCD05 を用います．クラス MCD05 は Windows の Visual C++(Visual Studio 2008)のコンソールアプリ用です（Unicode には対応しておりません）．基本的にファームウェアの制御プログラム MCD05_Ctrl とクラス MCD05 は同じバージョンのものを使用します．本マニュアルの対象は Ver.2.1 です．

MCD05 のプログラム（ファームウェア・パソコン側ソフトウェア）はオープンソースになっておりますので，用途・環境に応じて適宜書き換えてご利用ください．

1.1 プログラム作成に必要なファイル

MCD05 を SC02 に接続してプログラムを作成する場合，下記のファイルが必要となります．

MCD05.h	クラス MCD05 のヘッダファイル
MCD05.cpp	クラス MCD05 のソースファイル
SC02.h	クラス SC02 のヘッダファイル
SC02.cpp	クラス SC02 のソースファイル
ControlModule.h	クラス ControlModule のヘッダファイル
ftd2xx.h	FTDI 社 D2xx 用のヘッダファイル
ftd2xx.lib	FTDI 社 D2xx 用のライブラリ
ftd2xx.dll	FTDI 社 D2xx 用のダイナミックリンクライブラリ

SC02 は FTDI 社の USB チップ FT245RL を使用しているため，USB のデバイスドライバと専用ライブラリが必要となります．FTDI 社のホームページより D2XX ドライバをダウンロード・インストトールしてください．ダウンロードした D2XX ファイルにはドライバ本体のほかに，fed2xx.h，fed2xx.dll，fed2xx.lib が一括して含まれています．

DATK の各モジュール用のクラスは基底クラス ControlModule を継承しているため．クラス ControlModule 用のファイルが必要となります．

1.2 プログラム開始から終了まで

もっとも簡単な動作例として MCD05 を SC02 と接続して使う例を示します．

ID＝2 のモジュールをオープンして，PWM 出力値の設定とエンコーダカウント値を入力するプログラムです．エラー処理等は省略しています．詳しくはサンプルプログラムのソースコードをご覧ください．

```
MCD05 mcd;                  // クラスの生成
SC02 sc;                    // クラスの生成

sc.Open( 1 );               // ID = 1 の SC02 をオープン
mcd.Open( &sc, 0, 2 );      // ID = 2 の MCD05 をオープン
制御ループ{
        mcd.PwmBoard.sPwm1 = 100; // PWM デューティー比の設定
        mcd.Exchange(TRUE);       // データ送受信
        printf("enc %d¥n", mcd.EncBoard.lEncoder); // エンコーダ値の表示
}
mcd.Close();            // クローズ
sc.Close();             // クローズ
```

また，MCD05 を複数台使用する場合，送受信をまとめて行うことで通信間隔を高速化できます．例えば MCD05 を２台（mcd1,mcd2）使用する場合，

```
mcd1.Exchange(FALSE);       // この時点では送信しない
mcd.Exchange(TRUE);         // ここでまとめて送受信
```

とすることで，送受信の効率を向上できます．

ただしまとめて送信できる台数には限度があります．詳細は次節で解説いたします．

1.3 送受信データサイズ

MCD05 は複数枚連結して使用することを前提としていますが，一度に送受信できるデータ量には限界があります．この限界はシリアル通信モジュール（SC02）のバッファサイズによって規定されます（SC02 は送・受信ともに 256 バイト）．

2 台のモジュールを SC02 に連結した場合の送信パケットの構成を表 1 に受信パケットの構成を表 2 に示します．

表 1　SC02 送信パケット

項目	サイズ（バイト）
コマンド	1
送受信デバイス数	1
モジュール 1 のパケット	
ポート番号	1
ID	1
送信サイズ	2
受信サイズ	2
コマンド	1
データ	X
チェックサム	1
モジュール 2 のパケット	

表 2　SC02 受信パケット

項目	サイズ（バイト）
モジュール 1 のデータ	
ポート番号	1
ID	1
受信サイズ	2
データ	X
ステータス	1
チェックサム	1
モジュール 2 のパケット	
ステータス	1

・計算例

2 台の MCD05 を SC02 に連結し Exchange コマンドを実行した場合の送受信パケットサイズを計算します．表 1，2 より，送信データ以外の要素は 8 バイト，PWM 出力の送信データは 2 バイトであるため，1 台の MCD05 の送信パケットサイズ TxSize は，

TxSize = 8 バイト + 2 バイト = 10 バイト

となります．同様に MCD05 のエンコーダ入力の受信データは 4 バイトとなるので，受信パケットサイズ RxSize は，

RxSize = 6 バイト + 4 バイト = 10 バイト

となります．

SC02 の送信データ以外の要素は 2 バイトなので SC02 の送信パケットサイズ

ScTxSize は，

ScTxSize = 2 バイト+ 10 バイト×2 = 22 バイト

となります．同様に SC02 の受信パケットサイズ ScRxSize は，

ScRxSize = 1 バイト+ 10 バイト×2 = 21 バイト

となります．

2. 入出力データ構造体

2.1 MCD05_INPUT 型構造体

```
typedef struct _MCD05_INPUT_ENC
{
    long lEncoder;          エンコーダのカウント値.
} MCD05_INPUT_ENC;
```

概要

エンコーダデータの構造体.

2.2 MCD05_OUTPUT 型構造体

定義

```
typedef struct _MCD05_OUTPUT
{
    short sPwm;     PWM出力デューティ比.
} MCD05_OUTPUT;
```

概要

PWM 出力基板用のデータ構造体．デューティ比の設定値によって，モータドライバの状態は以下のようになります．

sPwm	状態
0	停止（フリー）
1024	正転 100%
-1024	逆転 100%
1024 以上	電磁ブレーキ

3. クラス MCD05

クラス MCD05 はクラス ControlModule を継承しています．このためクラス ControlModule の変数や関数がメンバとなりますが，実際には使用されず無効となる変数や機能もあります．

3.1 メンバ変数

クラスConrolModuleからの継承

BYTE bPort;	MCD05の接続されているポート（無効）．
BYTE bID;	MCD05のID．
BYTE bStatus;	MCD05本体及びパソコンのステータス．
BYTE bVersion;	ファームウェアバージョン．
char Error[256];	エラーメッセージ．

クラスMCD05独自のもの

MCD05_INPUT InputData;	エンコーダ入力データ構造体
MCD05_OUTPUT OutputData;	PWM出力データ構造体

3.2 MCD05::Open

書式

```
BOOL Open( ControlModule* pParent, BYTE bParentPort, BYTE bModuleID );
```

引数

ControlModule* pParent	親モジュールのポインタ
BYTE bParentPort	接続している親モジュールのポート番号
BYTE bModuleID	MCD05 の ID（0～255）

戻値

成功：TRUE

失敗：FALSE

動作

MCD05 をオープンします．bModuleID に CM_MASTER_ID を指定すると，すべての ID の制御モジュールが応答します．ID が分からない場合や，制御モジュールが 1 枚しか使用しない場合に CM_MASTER_ID を設定します．

オープンの時点で接続されている入出力ボードの情報が bBoardID_PA と bBoardID_PB に入力されます．

3.3 MCD05::Close

書式

BOOL Close(void);

引数

なし

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 をクローズします.

3.4 MCD05::Reset

書式

BOOL Reset(void);

引数

なし

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 をリセットします．MCD05 内部でソフトウェアリセットがかかります.

3.5 MCD05::Ping

書式

BOOL Ping(void);

引数

なし

戻値

成功：TRUE，失敗：FALSE

動作

動作チェック用のコマンドです. パソコン側から送信した 1 バイトデータと同じデータが返ってくるかを確認します.

3.6 MCD05::SetID

書式

BYTE bNewID　　新しい ID（0～254）

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 に新しい ID を設定します．新しい ID は MCD05 の EEPROM に書き込まれるため，次回起動時も有効になります．ID=255(CM_MASTER_ID)は指定できません．

3.7 MCD05::GetID

書式

BOOL GetID(void);

引数

なし

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 に設定されている ID を取得し bID に書き込みます．

3.8 MCD05::GetVersion

書式

BOOL GetVersion(void);

引数

なし

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 のプログラムバージョンを取得し bVersion に書き込みます．
プログラムバージョンが 2.1 の場合 bVersion = 0x21 となります．

3.9 MCD05::Blink

書式

BOOL Blink(BYTE bNum);

引数

BYTE bNum　　点滅回数

戻値

成功：TRUE，失敗：FALSE

動作

bNum に設定した回数 LED が点滅します．チェック時などに使用します．

3.10 MCD05::SetParam

書式

BOOL SetParam(PCM_PARAM pParam);

引数

PCM_PARAM pParam　　制御パラメータ構造体のポインタ

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 に制御パラメータを設定します．MCD05 では制御タイムアウトは使用されていませんので fCtrlTimeout の値は無効となります．

3.11 MCD05::GetParam

書式

BOOL GetParam(PCM_PARAM pParam);

引数

PCM_PARAM pParam　　制御パラメータ構造体のポインタ

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 に設定されている制御パラメータを取得します．

3.12 MCD05::Exchange

書式 1

```
BOOL Exchange( void* pOutputData, void* pInputData, BOOL RightNow=TRUE );
```

引数

void* pOutputData	出力データバッファのポインタ
void* pInputData	入力データバッファのポインタ
BOOL RightNow	送受信タイミングの指定

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 とデータの送受信を行います．pOutputData に MCD05_OUTPUT 構造体のポインタを，pInputData に MCD05_INPUT 構造体のポインタを設定します．引数 RightNow を省略するか TRUE を指定するとすぐに送受信が行われ，FALSE を指定すると送受信予約だけが行われます．

書式 2

```
BOOL Exchange( BOOL RightNow=TRUE );
```

引数

BOOL RightNow	送受信タイミングの指定

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 とデータの送受信を行います．接続されている入出力基板に合わせたデータが送受信されます．所要の PWM 出力値はメンバ変数 OutputData の sPwm に設定します．エンコーダのカウント値は，メンバ変数 InputData の lEncoder に入力されます．引数 RightNow を省略するか TRUE を指定するとすぐに送受信が行われ，FALSE を指定すると送受信予約だけが行われます．

3.13 MCD05::ClearData

書式

```
BOOL ClearData( void );
```

引数

なし

戻値

成功：TRUE，失敗：FALSE

動作

MCD05 内部のデータをクリアします．エンコーダのカウント値が 0 にクリアされます．

*ファームウェア v2.0 は未対応です．

付録

CM_PARAM 型構造体

メンバ変数

float dt;	制御周期をミリ秒単位で指定します．デフォルトは1.0ミリ秒です．
float fCtrlTimeout;	制御タイムアウトをミリ秒単位で指定します．デフォルトは1000ミリ秒です．0でタイムアウトは無効になります．
float fCommTimeout;	パソコン側，モジュール側双方の送受信通信タイムアウトをミリ秒単位で指定します．デフォルトは1000ミリ秒です．0でタイムアウトは無効になります．

注意事項

・本ソフトウェアは，ソースコードをオープンにしておりますが，商業利用・無断転載はおやめください．また，本ソフトウェアはお客様が自由に改変して御使用いただけますが，お客様のソフトウェア改変に伴う事故や故障等に関して当方では，一切の責任を負いません．
・本ソフトウェアにより生じるお客様の製品に起因して発生したいかなる損害に対しても当方では，一切の責任を負いません．
・本ソフトウェアの仕様は改良のため予告なく変更することがあります．

改版履歴

2015 年 8 月　初版

お問い合わせはメールにてお願いいたします．

アークデバイス

E-mail:info@arcdevice.com

URL: http://www.arcdevice.com/