

シグマ光機 Mark-102 用 gcc ライブラリ

中野武雄 nakano@apm.seikei.ac.jp

2004年3月7日 version 1.5

概要

シグマ光機のステージコントローラ Mark-102 を Linux から制御するための gcc 用ライブラリ関数群の仕様書。

1 概要

この文書はシグマ光機製のステージコントローラ Mark-102 を、シリアルインターフェースを通して Linux の gcc から制御するためのライブラリ関数の仕様書である。

Mark-102 と Linux PC の接続は、コントローラ付属のシリアルケーブル (ストレートケーブル) を用いる。PC の COM1 に接続した場合は Linux でのデバイスファイルは /dev/tty0、COM2 に接続した場合は /dev/tty1 となる。

ライブラリの実体は m102.h と m102.c の 2 つのファイルである。m102.h で各種マクロや関数プロトタイプを宣言し、実際の関数の実装を m102.c で行っている。

通信の際には、コントローラ側の設定は

- 通信速度 9600bps
- デリミタは CR

とすること。ディップスイッチで変更できる。詳細は [1] を見よ。

以下のセクションで出てくる関数群を C のプログラムから使うためには、m102.h をインクルードした上で m102.c を一緒にコンパイルすること。例えば foo.c というプログラムから使う場合は、m102.h と m102.c とを foo.c と同じディレクトリに置き、プログラムの先頭付近で

```
#include "m102.h"
```

と宣言して

```
% gcc -o foo foo.c m102.c
```

の様にコンパイルを実行すればよい。詳細は cvs リポジトリ内の Makefile を参照のこと。

プログラム作成に際しては、シグマ光機の資料 [1] を参考にした。以下で記述されていない詳細については、こちらをあたってほしい。なお以下の記述では、第一軸を X、第二軸を Y と記述している。現在バキュームプロダクツから購入したステージは 1 軸仕様なので、X の方しか使わないし、Y 軸関数の大部分は実装していない。

2 m102.h で定義されているマクロ

m102.h で定義されているマクロを説明する。これらは Q: コマンドや !: コマンドによって返される、ステージの状態を反映したものである。

2.1 ACK1 用マクロ

ACK1_NG 直前のコマンド実行に失敗した。

ACK1_OK 直前のコマンド実行に成功した。

2.2 ACK2 用マクロ

ACK2_NORMAL ステージは通常に停止した。

ACK2_LIMIT1 1 軸 (X 軸) がリミットによって停止した。

ACK2_LIMIT2 2 軸 (Y 軸) がリミットによって停止した。

後述する関数 m102_q が返す ack2 は、後者二つの bitwise-OR を取ることがある。

2.3 ACK3 用マクロ

ACK3_BUSY busy 状態で、移動系のコマンドを受けつけない。

ACK3_READY すべてのコマンドを受付ける状態。

3 C 関数

C プログラムから利用できる関数を示す。

3.1 m102_init

書式 `int m102_init(const char* device)`

説明 シリアルデバイスをオープンし、通信条件を設定する。以降で説明されている全ての関数を使う前に実行しておかなければならない。

引数 *device* オープンするシリアルデバイスのファイル名。

戻り値 オープンに成功したら 0。失敗すると `exit(-1)` する。

3.2 m102_close

書式 `int m102_close()`

説明 通信条件を元に戻し、デバイスをクローズする。全ての作業が終了したらこの関数を呼ぶ。

引数 なし

戻り値 成功したら 0、失敗したら 1。

3.3 m102_q

書式 `int m102_q(long *xpos, long *ypos, char *ack1, char *ack2, char *ack3);`

説明 Q: コマンドを発行し、コントローラの状態を問合わせる。

引数 *xpos* X 軸の位置を返す (パルス単位)

ypos Y 軸の位置を返す (パルス単位)

ack1 直前のコマンドの結果を返す。2.1 も参照。

ack2 ステージの停止位置に関する情報を返す。2.2 も参照。

ack3 ステージの移動状態に関する情報を返す。2.3 も参照。

戻り値 コマンドの発行と結果取得に成功したら 0、失敗したら 1。

3.4 m102_q2

書式 `int m102_q2(char *ack3);`

説明 m102_q の簡易版。ack3 のみの情報を返す。

引数 *ack3* ステージの移動状態に関する情報を返す。2.3 も参照。

戻り値 コマンドの発行と結果取得に成功したら 0、失敗したら 1。

3.5 m102_mx

書式 `int m102_mx(const long xmove);`

説明 X 軸の移動量 (パルス単位) を伝える M: コマンドをコントローラに送る。実際に移動するにはこの後 m102_g 関数を呼ぶ必要がある。

引数 *xmove* X 軸方向の移動量を返す。パキュームプロダクツ製ステージの場合は正が右方向、負が左方向。

戻り値 コマンドの発行に成功したら 0、失敗したら 1。

3.6 m102_g

書式 `int m102_g();`

説明 ステージ移動を開始する G: コマンドをコントローラに送る。移動条件は、この前に `m102_m` コマンドで設定しておく必要がある。

引数 なし。

戻り値 コマンドの発行に成功したら 0、失敗したら 1。

注意 この関数から返っても、ステージの移動が終了したわけではない。 `m102_q` 関数や `m102_q2` 関数の返すパラメータ `ack3` が `ACK3_READY` になるのを確認する必要がある。

3.7 m102_movex

書式 `int m102_movex(const long xmove);`

説明 `m102_mx` と `m102_g` を実行し、 `m102_q2` 関数をパラメータ `ack3` が `ACK3_READY` になるまで繰り返し実行する。このコマンドが返ったときには、ステージ移動は終了していると考えてよい。

引数 *xmove* X 軸方向の移動量を返す。バキュームプロダクツ製ステージの場合は正が右方向、負が左方向。

戻り値 コマンドの発行に成功したら 0、失敗したら 1、リミットセンサで停止したら 2。

注意 バックラッシュの処理はしていない。

3.8 m102_holdx

書式 `int m102_holdx();`

説明 C:11 コマンドを発行する。X 軸のモーターを励磁状態にし、ステージが手で移動できないようにする。

引数 なし。

戻り値 コマンドの発行に成功したら 0、失敗したら 1。

3.9 m102_releasex

書式 `int m102_releasex();`

説明 C:10 コマンドを発行する。X 軸のモーターを解放状態にし、ステージが手で移動できるようにする。

引数 なし。

戻り値 コマンドの発行に成功したら 0、失敗したら 1。

3.10 m102_hzerox

書式 `int m102_hzerox();`

説明 H:1- コマンドを発行する。負側 (左側) ハードウェアリミット+500 の位置を原点にする。

引数 なし。

戻り値 コマンドの発行に成功したら 0、失敗したら 1。

注意 ディップスイッチ (SW2-5) の設定によって、この機能自体の有効/無効を切り替えできる。

3.11 m102_origx

書式 `int m102_origx();`

説明 R:1 コマンドを発行する。現在のステージ位置を X 軸の原点にする。

引数 なし。

戻り値 コマンドの発行に成功したら 0、失敗したら 1。

注意 電源投入時には、各軸の現在位置が原点になる。m102_hzerox コマンドによって原点は再現性のあるかたちで定義できる。

4 例

以上の関数の利用例を示す。これはコマンドライン引数の分だけステージを移動するプログラムになる。cvs リポジトリに `test_m102.c` として保存されている。

```
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include "m102.h"

int main(int argc, char * argv[])
{
    long xpos, ypos, xmove;
    char ack1, ack2, ack3;
```

```

char *endptr;

if (argc != 2) {
    printf("Usage: 'm102 <xmove>'\n");
    exit(1);
}
xmove = strtol(argv[1], &endptr, 10);
if (endptr[0] != '\0' || errno == ERANGE || errno == EINVAL){
    fprintf(stderr, "arg error\n");
    exit(1);
}

m102init("/dev/ttyS0");

m102_q(&xpos, &ypos, &ack1, &ack2, &ack3);
printf("X: %ld, Y: %ld\n", xpos, ypos);

m102_mx(xmove);
m102_g();

m102_q2(&ack3);
while(ack3 == ACK3_BUSY) {
    m102_q2(&ack3);
}

m102_q(&xpos, &ypos, &ack1, &ack2, &ack3);
printf("X: %ld, Y: %ld\n", xpos, ypos);

m102close();
return 0;
}

```

参考文献

- [1] シグマ光機「2軸ステージコントローラ Mark-102 取り扱い説明書」
- [2] Peter H. Baumann 著、藤原輝嘉 訳 “The Linux Serial Programming HOWTO”
<http://www.linux.or.jp/JF/JFdocs/Serial-Programming-HOWTO.html>