

電光掲示板

石川翔飛、市坂直人
森野雄也

1. 研究概要

私たちは双葉電子工業株式会社の「蛍光表示ディスプレイモジュール」を使用し電光掲示板を製作する研究に取り組んだ。RS-232C ケーブルの作成やプログラム等、ハードウェアとソフトウェアの相互の構造を理解し、今後活かせるような知識や技術を身につけることを目的とした。

2. 研究の具体的内容

今回使用するモジュールは以下のものである。

メーカー

- ・双葉電子工業株式会社

型番

- ・GP1058A02A
14文字*(336*24ドット)タイプ



図 1

蛍光表示ディスプレイモジュール

特徴

- ・独自の方式で高輝度表示が可能
- ・通信形態は RS-232C
- ・漢字 ROM 搭載
- ・外部から任意の表示パターンを表示できるユーザフォントに対応

今回は、PC からモジュールに文字コードを送信してそれを表示させるため、RS-232C ケーブルを作成する。シリアル通信用のピンは番号が合うピン同士を配線し、D-SUB コネクタ側の 4 番ピン（データ端末レディ）と 6 番ピン（データセットレディ）を繋いで作成した。

ピン No.	端子名
1	NC
2	RXD
3	TXD
4	RTS
5	CTS
6	GND

図 3

モジュールのピン番と端子の対応表



図 4 モジュール側の RS-232C

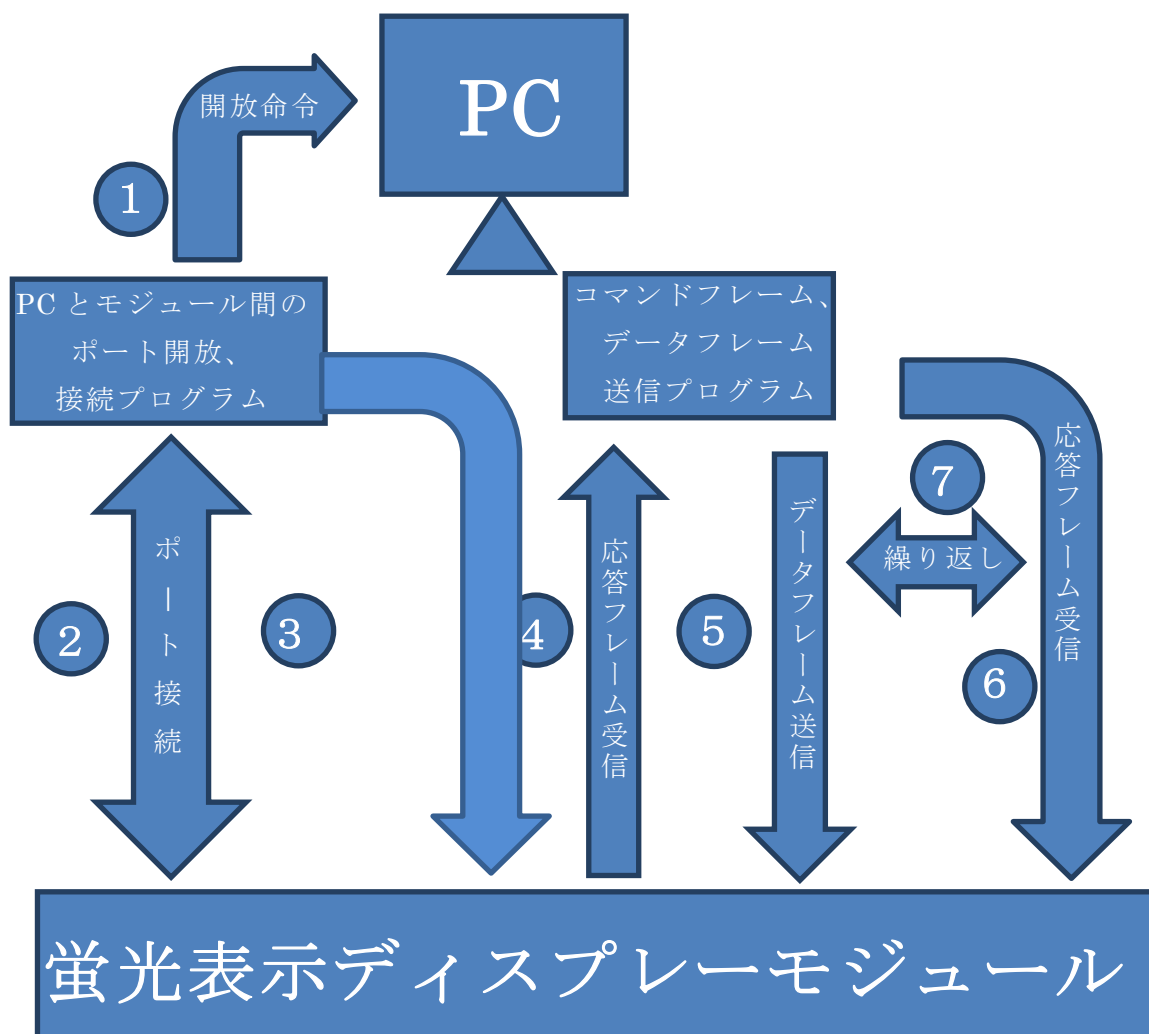


図5 PC、モジュール間の構造

今回は文字送信用のプログラム作成にあたって、

<http://www.kosaka-lab.com/tips/2010/06/vfd-1.php> このホームページで配布されているC#で作られた秋月VFDというソフトをベースに表示OFF機能やスクロール機能を加えた。まずはPCのポート開放し、ケーブル接続後の通信が行えるようにした。その後ポートに接続し、文字を表示させるために必要なデータをそれぞれ送受信していく。上記の図のコマンドフレームには、アドレス、フレーム番号、コマンド、フレーム数、SUM部、

END部が含まれている。コマンド部には即時表示用のコマンド04Hが入力される。応答フレームにはコマンドフレームのコマンド部の代わりに応答部、データフレームにはメッセージ部データフレームに対しての応答フレームには命令終了を示す終了コードが含まれている表示データが256バイトより多い場合はデータフレームと応答フレームのやり取りを繰り返す。それぞれの命令の詳細を以下記述していく。

このモジュールには漢字 ROM が搭載されており、日本語の表示が行えるが、漢字 ROM コードは jis コードをベースにしたオリジナルのコードであるため、シフト JIS コード（1バイト文字と2バイト文字が混合）から jis コード（1バイト文字と2バイト文字の切り替えが必要）に変換した後、さらに漢字 ROM オリジナルコードへ変換する必要がある。テキストボックスに表示させたい文字を入力し、送信ボタンを押した時に、以下の処理が漢字 ROM コードに変換してくれる。それを送信することで文字が表示される。

```

・シフト jis→jis
byte[]byteMessage =
Encoding.GetEncoding(50220).GetBytes(message);
・jis→vfd 漢字ROMコード
byte[] vfdCode = toVFDCODE(byteMessage);

```

(1) 文字表示 *****
漢字 ROM コードにより文字を指定します。(フォントテーブル P17~24 参照)
JIS コードから漢字 ROM コードへの変換を以下に示します。

区分	b7	b6	b5	d8	d7	d6	d5	d4	d3	d2	d1	c8	c7	c6	c5	c4	c3	c2	c1
非漢字	0	1	0				0	a7	a6	b3	b2	b1	0	0	a5	a4	a3	a2	a1
漢字(第1水準)	0	1	1				0	b7	b4	b3	b2	b1	a7	a6	a5	a4	a3	a2	a1
漢字(第2水準)	1	0	0				0	b7	b4	b3	b2	b1	a7	a6	a5	a4	a3	a2	a1
漢字(第1水準)	1	0	1				1	b6	b4	b3	b2	b1	a7	a6	a5	a4	a3	a2	a1
漢字(第2水準)	1	1	0				1	b6	b4	b3	b2	b1	a7	a6	a5	a4	a3	a2	a1
漢字(第2水準)	1	1	1				1	a7	a6	b3	b2	b1	0	0	a5	a4	a3	a2	a1

半角かなフラグ("1"=ひらがな, "0"=カタカナ)
半角文字フラグ("1"=半角, "0"=全角)
非文字フラグ("1"=コマンド, "0"=文字)

注) ・ b7~b1 は JIS コード第1バイト, a7~a1 は JIS コード第2バイトを示します。
・ d8~d1 は漢字 ROM コードの上位バイト, c8~c1 は漢字 ROM コードの下部バイトを示します。
*1: アルファベット, 数字, カタカナ, ひらがな
・ 半角フラグ: P21 のフォントテーブル "アア" を例にすると "1"で"あ" "0"で"ア"を表示します。
・ 非表示フラグ: 文字表示で使用する場合は"0"として下さい。

図6 jis と漢字 ROM コードの対応表

PC とモジュール間を接続した後は、コマンドフレームを送信する必要があります。右上の図のような順番の流れで動作が行われていくので、それに従ってプログラムを作成していく。

(1) コマンドフレーム 図-1

アドレス 1バイト	フレーム番号 1バイト "00H"固定	コマンド 1バイト	フレーム数 1バイト	SUM 2バイト	END 1バイト "EFH"固定
アドレス:	Don't care				
フレーム番号:	"00H"固定				
コマンド:	コマンドコード 表-8, 9 参照				
フレーム数:	コマンドで 03H, 04H, 05H を指定した場合は、この後に送信するデータの量に応じたフレームの数。それ以外の場合は、"00H"。				
SUM:	アドレスからフレーム数までをバイト単位で加算した値 上位バイト、下位バイトの順に送信する。				
END:	"EFH"固定				

図7 実際に送信される
コマンドフレームの内容

アドレスは Dont' care なので、何を入れても動作に違いはない。フレーム番号はコマンドフレームの場合、常に 0x00 を送信する。対応した文字コードをプログラム内の `this.send_commandFrame(0x**, (byte)nFrame)` というソースの ** に入力することで、電光掲示板の動作をコントロールする。例えば、表示を OFF にしたい場合、OFF やクリアなどといったボタンを作りそこに上記の関数を格納し、下記の表示を OFF にするコマンド 11 を入すればボタンを押せば電光掲示板に消えている文字が消えるといった仕様である。メッセージを送信する場合は、まずコマンドフレームのコマンド部にコマンドを送信し、フレーム部にデータフレームによりデータを送信する。コマンド部に使用できる表示コントロールコードには以下のものがある。

03H	送信したデータを表示中のデータの最後に続けて表示する。
04H	送信したデータを即表示する。
05H	送信したデータを RAM に格納するが表示は切り換えない。 別途、表示画面切り替えコマンドにより表示を切り替える。
10H	表示を ON にする。
11H	表示を OFF にする。
12H	スクロールを開始する。

13H	スクロールを停止する。
14H	表示データをすべてクリアする。
15H	表示設定をデフォルトにする。
17H	表示画面を切り替える。
18H	表示データの頭出しをする。
1CH	表示動作状態の確認をする。 スクロール中"C2"、スクロール停止状態"C0"を応答する。
25H	画素数を 339×24 に設定する。
30H	点滅コマンドの表示状態を点灯と消灯の切り替えにする。
31H	点滅コマンドの表示状態を点灯と反転表示の切り替えにする。
41H	タイムアウト時間 0.5 秒設定。
42H	タイムアウト時間 1.0 秒設定。
43H	タイムアウト時間 2.0 秒設定。
44H	タイムアウト時間 3.0 秒設定。
45H	タイムアウト時間 5.0 秒設定。

図 8 コマンド部に使用できる表示コントロールコード

フレーム数にはこれから送信するデータフレームの総数を送信する。データフレーム内のメッセージ部に入れられるデータは 256 バイトなので、漢字 ROM コードへ変換されたバイト配列の総数からフレーム数を割り出す。

SUM にはアドレスからフレーム数までのデータを合計した値を入れる。

END は 0xEF で固定。

コマンドフレームを送信すると応答フレームがかえってきて、その後にデータフレームを送信していく。

(2) データフレーム

アドレス 1 バイト	フレーム番号 1 バイト	コマンド 1 バイト "00H"固定	メッセージ 256 バイト	SUM 2 バイト	END 1 バイト "EFH"固定
アドレス:	Don't care				
フレーム番号:	01H から表示データフレーム毎にインクリメント				
コマンド:	"00H"固定				
フレーム数:	データ 256 バイト固定 空きは FFH にする。				
SUM:	アドレスからメッセージまでをバイト単位で加算した値 上位バイト、下位バイトの順に送信する。				
END:	"EFH"固定				

図 9 データフレームの構成

コマンドフレームの時と同じような手順でプログラムを作成し、送信する。メッセージに、漢字 ROM コードへ変換されたバイト配列を 256 バイトずつ入れて送信していく。フレーム番号はそのたびにカウントしていく。コマンドは 00H で固定。フレーム数は 256 バイトで固定だが、空きは FFH で埋める必要があるため 0xff を代入していく。データフレームのプログラムを例にすると

```
//アドレス
send[0] = 0x00;
//フレーム番号
send[1] = frame;
//コマンド
send[2] = 0x00;
```

```
//メッセージ
for (int i = 3; i < 256 + 3; i++)
{
    send[i] = message[i - 3];
}
```

```
//サム計算
int nSum = 0;
for (int i = 0; i < 256 + 3; i++)
{
    nSum += send[i];
}
```

byteSum =

BitConverter.GetBytes(nSum);

```
//SUM
send[259] = byteSum[1];
send[260] = byteSum[0];
```

```
//END
send[261] = 0xEF;
```

このようになり図 9 の内容に沿ってフレームが送信されていることがわかる。

これらの動作を行っていくことで電光掲示板に文字が表示され、また、スクロールや点滅の制御をしていくことができる。

今回利用させていただいたプログラムには単純に文字を表示させる機能しかなかったためより使用しやすいように以下の改良を加わえた。

- ・スクロール機能の追加
- ・スクロール停止機能の追加
- ・時間の取得、表示機能の追加
- ・文字表示クリア機能の追加

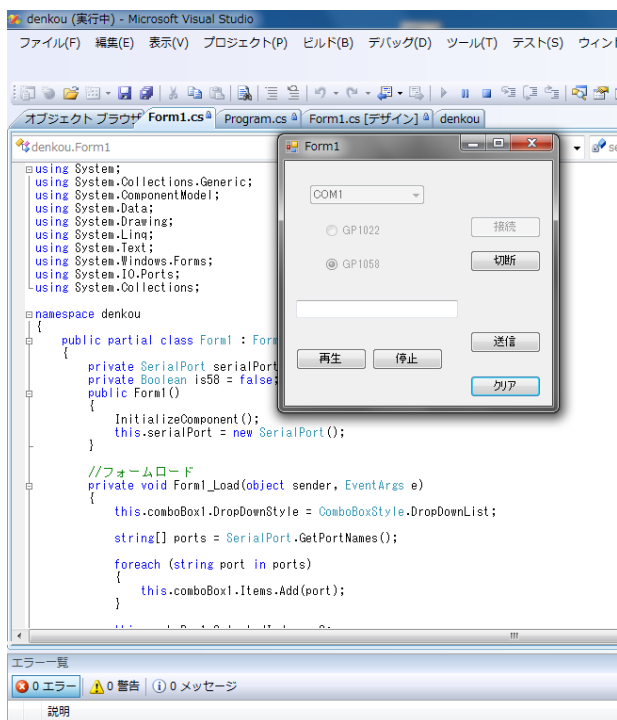


図 10 プログラム実行時

例として文字表示クリア機能の場合コマンドフレームに 14H というコマンドを使ので以下のようなプログラムを追加することによって文字表示のクリアが可能となる

```
private void button4_Click(object sender, EventArgs e)
{
    send_commandFrame(0x14, 0);
}
```

動作手順

- 1 : 電源を入れる。
- 2 : RS-232C をモジュール側と PC 側に繋ぐ。
- 3 : ソフトを実行。
- 4 : GP1022 か GP1058、自分が使用しているモジュールの型番を選択。
- 5 : モジュールが接続されている COM ポートを選び、接続ボタンを押す。
- 6 : テキストボックスに文字を入力して送信ボタンを押す。
- 7 : 切断ボタンを押して、終了。



図 11 動作結果

3. 研究のまとめ

C#はあまり使用したことがなかったため、インターネットを見たりや友人に聞いたりしながらプログラムを改良した。ソフトウェア面だけでなく、RS-232C ケーブルの作成でハードウェア面の経験もできたので、非常に充実した研究になった。また、予定にあったように文化祭でも使用することができ、在学中に役に立たせることができて良かったと思う。この電光掲示板一つで様々な動作に対応しているため、既存の文字コードを応用したりプログラムをさらに改良したりしてもっと便利なものにしていきたい。また、今回は時間の関係で電光掲示板の外装を作成できなかったのが心残りである。



図 12 岡工祭模擬店使用時

森野

この課題研究では初めてのことが多く配線にしてもプログラムにしてもなかなかうまくいかず苦勞した。しかしハード、ソフト両方に通じるものがあったので勉強になった。予定より大分作業が遅れたため当初作る予定だった外装などが作れなかったのは残念だったが文化祭の時模擬店に設置できたのでよかった。

感想

石川

八田先生に薦められて始めた課題研究だったが、予想よりも作業が進まなくて苦勞した。インターネット上にあった文字表示ソフトの中に画像を表示させることができるものがあったので自分たちの力で作成することができなかった。もっと柔軟な発想とそれを実現できる技術力を身につけたいと痛感した。

市坂

電光掲示板という、身近にあるがなかなか作成できる機会がないものを作ることとはとても良い課題研究になったと思う。目標の一つであった、文化祭の模擬店で実際に使用することができてよかったが、外装の作成が間に合わなかったことは残念だった。今回の課題研究で学んだことは将来役立つと思うので、このような貴重な体験をすることができてよかったと思う。

参考

<http://www.kosaka-lab.com/tips/2010/06/vfd.php>

<http://www.kosaka-lab.com/tips/2010/06/vfd-1.php>

<http://e-words.jp/>