

Report on the Experiment

No. 4

Subject マイコン実習(4):シリアル通信

Date 2020. 07. 02

Weather 晴れ Temp °C Wet %

Class E4
Group 2
Chief
Partner

No 14
Name 小畠 一泰

Kure National College of Technology

1 目的

組み込みマイコン dsPIC を用いてシリアル通信を行うことで, PC やセンサ等からのデータのやり取りに習熟することを目的とする.

2 実習

2.1 PC と通信

実習キットは USB により PC と接続され, USB-シリアル変換 IC(FT232RL) によって UART 通信が PC と行えるようになっている. また, マイコンの受信結果を確認するため, LCD ユニット (SC1602BS-B) が接続されている.

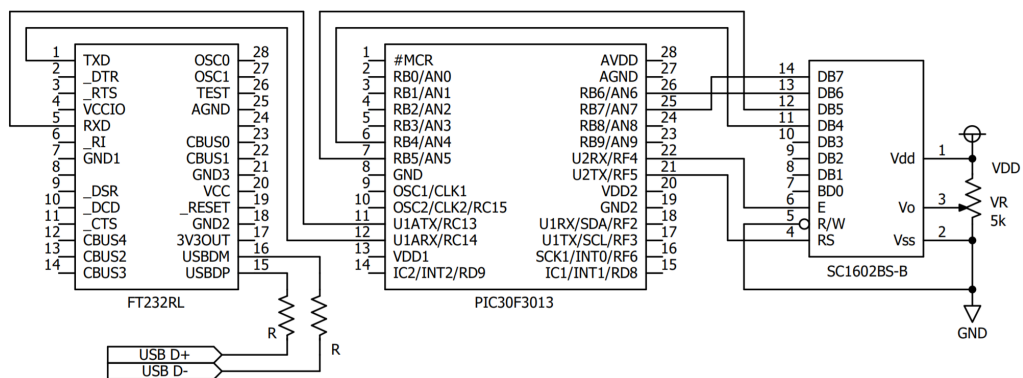


図 1: PC との通信回路

2.1.1 課題 1

PC から 1 文字送信し, マイコンで受信して LCD にそれを表示するプログラムを作成せよ.

コード 1 課題 1 のソースコード全文

```
1 #include <p30F3013.h>
2 #include "c:\work\e4exp.h"
3
4 int main(void) {
5     char x;
6     U1MODE = 0x8400;
7     U1BRG = 95;
8     U1STA = 0x0400;
9     init_LCD();
10    while (1) {
11        while (U1STAbits.URXDA != 1);
12        x = U1RXREG;
13        put_char(x);
14        while (U1STAbits.UTXBF != 0);
15        U1TXREG = x;
16    }
17 }
```

2.1.2 課題 2

マイコンから文字列”Hello PC”を送信し, LCD と PC にそれぞれ表示するプログラムを作成せよ.

コード 2 課題 2 のソースコード全文

```
1 #include <p30F3013.h>
2 #include "c:\work\e4exp.h"
3
4 int main(void) {
5     U1MODE = 0x8400;
6     U1BRG = 95;
7     U1STA = 0x0400;
8     init_LCD();
9     char* greet = "Hello PC";
10    while(*greet != '\0') {
11        U1TXREG = *greet;
12        put_char(*greet);
13        greet++;
14    }
15    while(1);
16 }
```

2.1.3 課題 3

マイコンから 10 個の数値を CSV に変換して送信し, PC の通信ソフトで受信した後, バッファの中身をファイルに保存後, それを Excel で開いて表示せよ.

図 2 にログのスクリーンショットを示す.

コード 3 課題 3 のソースコード全文

```
1  #include <p30F3013.h>
2  #include "c:\work\e4exp.h"
3
4  int main(void) {
5      U1MODE = 0x8400;
6      U1BRG = 95;
7      U1STA = 0x0400;
8
9      init_LCD();
10
11     char data[] = {'0','1','2','3','4','5','6','7','8','9'};
12     char split = ',';
13
14     int i = 0;
15     for(;i < 10; i++) {
16         U1TXREG = data[i];
17         put_char(data[i]);
18         U1TXREG = split;
19         put_char(split);
20     }
21
22     while(1);
23 }
```

	A	B	C	D	E	F	G	H	I	J
1	0	1	2	3	4	5	6	7	8	9
2										
3										

図 2: ログのスクリーンショット

2.2 アドレス付き UART 通信

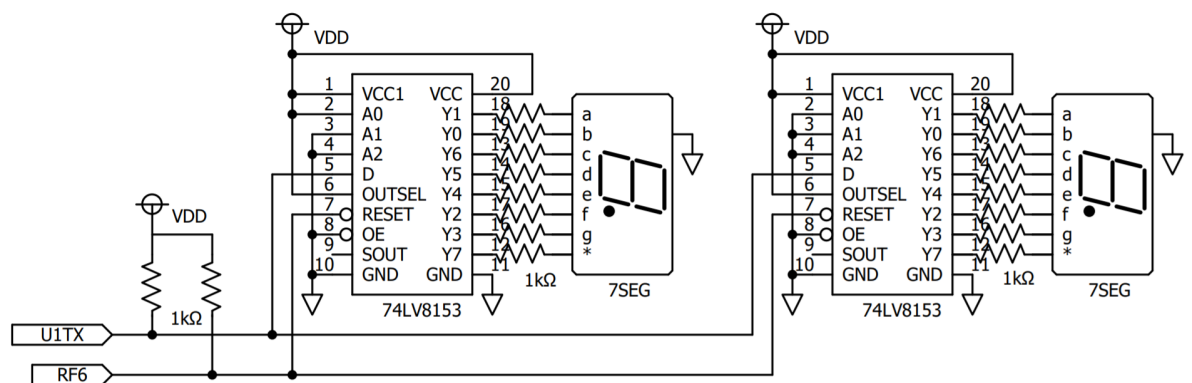


図 3: 2 個の SN74LV8153 に 7 セグメント LED を取り付けた回路

図 3 にあるように、シリアル-パラレル・インタフェースとしてテキサス・インスツルメンツ社の SN74LV8153 が 2 個並列に接続され、アドレス設定ピン (A0~A2, 2~4 番ピン) によりそれぞれアドレスが (000,001) に設定されている。dsPIC から UART 出力 U1TX(17 番ピン) が SN74LV8153 のシリアル・データ入力 D (5 番ピン) に、デジタル I/O の RF6(16 番ピン) が SN74LV8153 のレジスタ・ステータスの初期化 RESET (7 番ピン) に接続されている。CPU ボードからモジュール 3 には電源 VDD とグラウンド GND と上記 2 本が接続されているのみである。SN74LV8153 の出力 (Y0~Y7) は 7 セグメント LED に 1 [kΩ] を通して接続されている。伝送するシリアル・データのフォーマットを表 1 に示す。2 回の送信によりアドレスと 8 ビット分のデータ D0~D7 を送信する。このインタフェース IC のデータ伝送速度は無調整/無設定で 2 [kbps]~24 [kbps] の任意のビットレートに対応している。

表 1: シリアル・データのフォーマット

第 1 フレーム		第 2 フレーム	
スタートビット		スタートビット	
1		1	
A0	アドレス	A0	アドレス
A1		A1	
A2		A2	
D0	データ	D0	データ
D1		D1	
D2		D2	
D3		D3	
ストップビット		ストップビット	

2.2.1 課題 1

表 2 より, hdata[3] を hdata[16] に拡張せよ. 表 2 を埋めて, それを元に配列にデータを追加せよ.

コード 4 課題 1 のソースコード一部

```
1 unsigned char hdata[16]={ 0x77,0x41,0x3b,0x6b,0x4d,0x6e,  
2                          0x7e,0x43,0x7f,0x4f,0x5f,0x7c,  
3                          0x36,0x79,0x3e,0x1e };
```

2.2.2 課題 2

第 1 引数は表示する桁, 第 2 引数は表示する値 (0 から 16) を受け取り, 16 進数 1 桁分を表示する関数を作成せよ.

コード 5 課題 2 のソースコード一部

```
1 void printNumberWithDigit(unsigned digit, unsigned number) {  
2     unsigned char adrs = digit == 1 ? 1 : 3;  
3     unsigned char x = hdata[number];  
4     while(U1STAbits.UTXBF!=0);  
5     U1TXREG=adrs+(x<<4);  
6     while(U1STAbits.UTXBF!=0);  
7     U1TXREG=adrs+((x>>4)<<4);  
8 }
```

2.2.3 課題 3

#0 から 255 の値を受け取り, 2 桁の 16 進数のデータに分けて, 16 進数 2 桁を表示する関数を, 課題 5 の関数を利用して作成せよ.

コード 6 課題 3 のソースコード一部

```
1 void printNumber(unsigned number) {  
2     unsigned a = number % 16;  
3     unsigned b = number / 16;  
4     printNumberWithDigit(1, a);  
5     printNumberWithDigit(2, b);  
6 }
```

2.2.4 課題 4

LED 表示器に 00 から FF までの 16 進をカウント・アップするプログラムを作成せよ。

コード 7 課題 4 のソースコード全文 (前半)

```
1  #include <p30F3013.h>
2  #include "c:\work\e4exp.h"
3
4  unsigned char hdata[16]={ 0x77,0x41,0x3b,0x6b,0x4d,0x6e,
5                           0x7e,0x43,0x7f,0x4f,0x5f,0x7c,
6                           0x36,0x79,0x3e,0x1e };
7
8  void wait(unsigned x) {
9      unsigned i, j;
10     for (i = 0; i < x; i = i + 1) {
11         for (j = 0; j < 60000; j = j + 1){
12             asm("nop");
13         }
14     }
15 }
16
17 void printNumberWithDigit(unsigned digit, unsigned number) {
18     unsigned char adrs = digit == 1 ? 1 : 3;
19     unsigned char x = hdata[number];
20     while(U1STAbits.UTXBF!=0);
21     U1TXREG=adrs+(x<<4);
22     while(U1STAbits.UTXBF!=0);
23     U1TXREG=adrs+((x>>4)<<4);
24 }
25
26 void printNumber(unsigned number) {
27     unsigned a = number % 16;
28     unsigned b = number / 16;
29     printNumberWithDigit(1, a);
30     printNumberWithDigit(2, b);
31 }
```

コード 8 課題7のソースコード全文 (後半)

```
32 void main(void) {
33     _TRISF6 = 0;          // RF6 ポートを出力設定
34     U1BRG = 95;
35     U1MODE = 0x8000;
36     U1STA = 0x0400;
37
38     _RF6 = 0;            // SN74LV8153 をリセット
39     wait(10);
40     _RF6 = 1;            // SN74LV8153 release reset
41
42     while(1) {
43         int i = 0;
44         for(; i < 256; i++){
45             printNum(i);
46             wait(60);
47         }
48     }
49 }
```

3 検討課題

1. サンプル 1 は U1MODE = 0x8400; , サンプル 2 は U1MODE = 0x8000; となっている。その理由を説明せよ。
UART モードレジスタより 15 ビット目は UART の有効化ビットであり, 10 ビット目が入出力選択ビットの役割を果たしていることが分かる。ALTIO = 1 のときは U1ATX と U1ARX が, ALTIO = 0 のときは U1TX と U1RX が使われる。本実験キットでは PC との通信に USB を介しており, 図 1, 図 3 にそれぞれあるように USB-シリアル変換 IC(FT232RL) によって UART 通信が PC と行えるようになっている。サンプル 1 では PC にシリアル通信で文字情報を送るため ALTIO が 1 となっているが, サンプル 2 では PC と通信しないため ALTIO が 0 になっている。
- 2) マイコンのシリアル通信には UART の他に I2C や SPI がある。これについて調べよ。
 - I2C(Inter-Integrated Circuit)
 - 主に同じ基板内などのように近距離で直結したデバイスと, 100kbps または 400kbps の速度でシリアル通信を行う
 - マスタ/スレーブ間を, SCL と SDA という 2 本の線でパーティーライン状に接続する
 - 個々のスレーブがアドレスを持ちデータ中にアドレスが含まれている
 - 1 バイト転送毎に受信側から ACK 信号の返送をして, 互いに確認を取りながらデータ転送を行っている
 - SPI(Serial Peripheral Interface)
 - シリアル EEPROM や D/A コンバータなどの周辺 IC を専用のシリアルインタフェースで接続し, 高速の同期式通信を可能とする
 - 2 つの SPI のモジュール (マスタ/スレーブ) が互いに 3 本または 4 本 (SS 信号を使う場合) の線で接続する
 - マスタが出力するクロック信号 (SCK) を基準にして, 互いに向かい合わせて接続した SDI と SDO で, 同時に 1 ビット毎のデータの送受信を行う

– SS ピンの HIGH/LOW を切り替えることで通信するスレーブを指定する

4 参考文献

- 山口晶大: PIC マイコン・スタートアップ, CQ 出版社 (2009)
- マイコンボードに関する情報 http://www.marutsu.co.jp/shohin_40258/(2011.9.30 現在)
- 電子工作の実験室 <http://www.picfun.com/f1/f05.html>(2022.07.04 現在)
- 電子工作の実験室 <http://www.picfun.com/f1/f06.html>(2022.07.04 現在)