

# RS232Cテストデータ送信ユニットV2(SDカード対応)(PIC18F2550)

## 概要

前回製作したRS232Cテストデータ送信ユニットは、固定データを送信するものでしたが、今回は、SDカードに送信データを事前に登録しておくことにより、よりフレキシブルなテストデータ送信ユニットとしました。

## 動作原理

1. テキストエディタ(例えば、メモ帳、秀丸など)で、送信データファイルを作成します。(ファイル名は、“data.txt” 固定)

### <通信速度の設定>

RS232Cの通信速度を指定します。速度は、4800bps□9600bps□19200bpsの何れかを指定します。省略時は9600bpsとなります。

設定例

```
$BPS=9600
```

### <遅延時間の設定>

次のデータ送信までの遅延時間をmsec単位で設定します。

設定例

```
$SLEEP=1000
```

### <改行コードの設定>

改行コード(CR,LF)を設定します。

設定例

```
$CRLF
```

### <データ送信開始の設定>

以降の行から、データ送信を開始します。

設定例

```
$START
```

### <データ送信停止の設定>

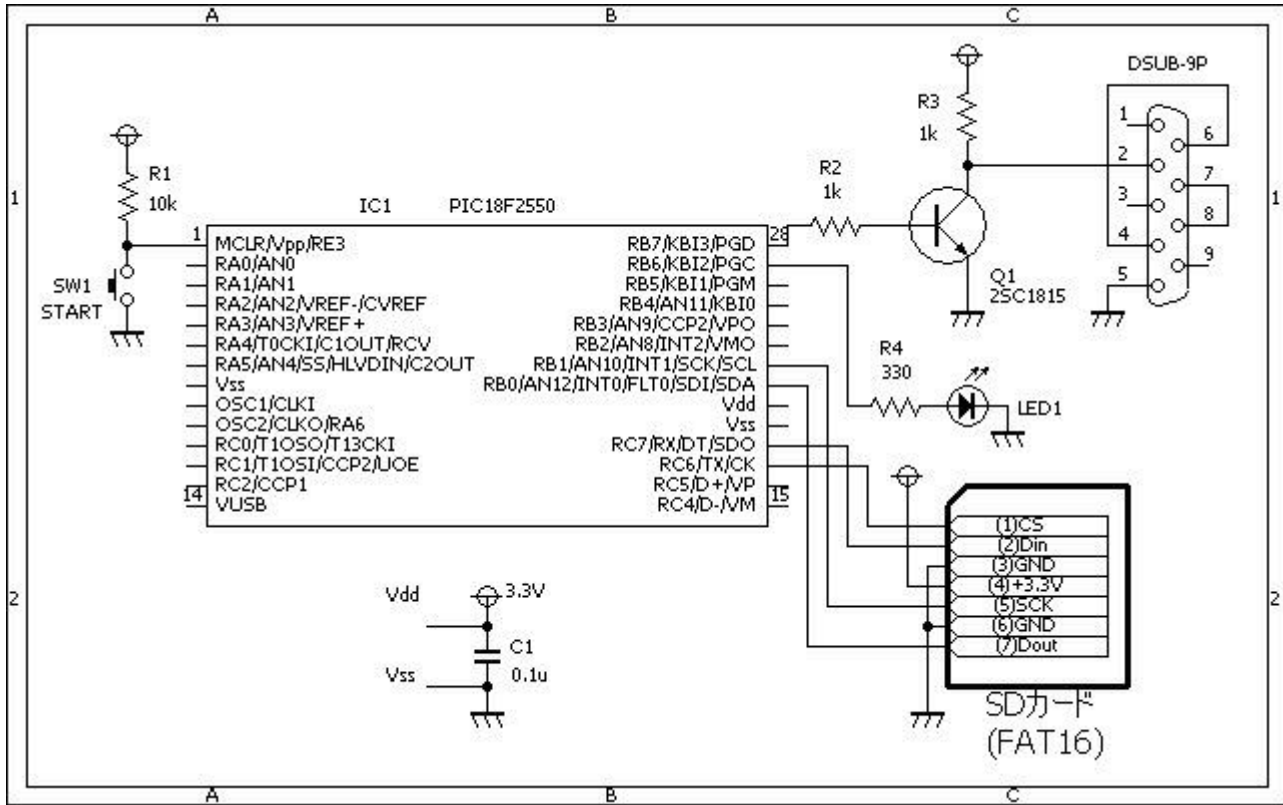
データ送信を停止します。

設定例

```
$STOP
```

2. SW1を押下すると、テストデータの送信を開始します□(LED1点灯)
3. \$STOPで、テストデータの送信を停止します□(LED1消灯)

## 回路図



# ソースコード

[rs232cTestV4.c](#)

```

//*****
*
/*
□□□□□□□□ テストデータ送信用□□□□対応)
*/
//*****
*

#define SW PORTE.F3
#define LED PORTC.F2

#define CR 0x0d
#define LF 0x0a

#define BPS4800 0
#define BPS9600 1
#define BPS19200 2

#define BPS4800_DELAY 190
#define BPS9600_DELAY 90
#define BPS19200_DELAY 35

//*****

```

```
*  
  
void  init_sdc()  
{  
    static  short  cnt;  
    //□□□□□□□□の初期化  
    Spi_Init_Advanced(MASTER_OSC_DIV64, DATA_SAMPLE_MIDDLE,  
CLK_IDLE_LOW, LOW_2_HIGH);  
    if (Mmc_Fat_Init(&PORTC, 6)) {  
        while (1) {  
            LED = 1;  
            Delay_ms(100);  
            LED = 0;  
            Delay_ms(100);  
        }  
    }  
    Spi_Init_Advanced(MASTER_OSC_DIV16, DATA_SAMPLE_MIDDLE,  
CLK_IDLE_LOW, LOW_2_HIGH);  
    for (cnt = 0; cnt < 3; cnt++) {  
        LED = 1;  
        Delay_ms(300);  
        LED = 0;  
        Delay_ms(300);  
    }  
}  
  
//*****  
*  
  
void  SwitchONcheck()  
{  
    while (Button(&PORTE, 3, 1, 0) == 0)  
        ;  
    while (Button(&PORTE, 3, 1, 1) == 0)  
        ;  
}  
  
//*****  
*  
  
static short  bps;  
  
void  Soft_Usart_Send_Chrc(char chr)  //9600bps  
{  
    static  short  cnt;  
    //  
    PORTB.F7 = 0;  //start bit  
    switch (bps) {  
    case BPS4800:  
        Delay_us(BPS4800_DELAY);  
        break;
```

```
    case BPS9600:
        Delay_us(BPS9600_DELAY);
        break;
    case BPS19200:
        Delay_us(BPS19200_DELAY);
        break;
}
//
for (cnt = 0; cnt < 8; cnt++) {
    PORTB.F7 = chr & 0x01;
    chr = chr >> 1;
    switch (bps) {
    case BPS4800:
        Delay_us(BPS4800_DELAY);
        break;
    case BPS9600:
        Delay_us(BPS9600_DELAY);
        break;
    case BPS19200:
        Delay_us(BPS19200_DELAY);
        break;
    }
}
//
PORTB.F7 = 1;    //stop bit
switch (bps) {
case BPS4800:
    Delay_us(BPS4800_DELAY);
    break;
case BPS9600:
    Delay_us(BPS9600_DELAY);
    break;
case BPS19200:
    Delay_us(BPS19200_DELAY);
    break;
}
}

//*****
*

void  Soft_Usart_Send_Str(char *str)
{
    while (*str != 0x00) {
        Soft_Usart_Send_Chr(*str);
        Delay_us(1000);
        str++;
    }
}

//*****
```

```
*  
  
void Sleep_ms(unsigned int tm)  
{  
    while (tm != 0) {  
        Delay_ms(1);  
        tm--;  
    }  
}  
  
//*****  
*  
  
void rs232cProc()  
{  
    //変数の定義  
    static char          buf[80], character, cnt, *pnt, mode;  
    static unsigned long  fsize, length;  
    //[]のファイルのオープン  
    Mmc_Fat_Assign("data.txt", 0);  
    Mmc_Fat_Reset(&fsize);  
    length = 0;  
    cnt = 0;  
    mode = 0;  
    //  
    while (length < fsize) {  
        //設定ファイルから1行分のデータを読み込む  
        for (cnt = 0; cnt < 80; cnt++) {  
            Mmc_Fat_Read(&character);  
            length++;  
            if (character == ' ') {  
                cnt--;  
                continue;  
            }  
            if (character == CR) {  
                cnt--;  
                continue;  
            }  
            if (character == LF) {  
                buf[cnt] = 0x00;  
                break;  
            }  
            buf[cnt] = character;  
        }  
        //  
        pnt = strstr(buf, "$BPS=");  
        if (pnt != 0x00) {  
            switch (atoi(&buf[5])) {  
                case 4800:  
                    bps = BPS4800;  
                    break;  
            }  
        }  
    }  
}
```

```
        case 9600:
            bps = BPS9600;
            break;
        case 19200:
            bps = BPS19200;
            break;
    }
    continue;
}
//
pnt = strstr(buf, "$SLEEP=");
if (pnt != 0x00) {
    Sleep_ms(atoi(&buf[7]));
    continue;
}
//
pnt = strstr(buf, "$CRLF");
if (pnt != 0x00) {
    Soft_Usart_Send_Chrcr(LF);
    Soft_Usart_Send_Chrcr(CR);
    continue;
}
//
pnt = strstr(buf, "$START");
if (pnt != 0x00) {
    mode = 1;
    continue;
}
//
pnt = strstr(buf, "$STOP");
if (pnt != 0x00) {
    mode = 0;
    continue;
}
//
if (mode == 1) {
    Soft_Usart_Send_Str(buf);
}
}
}

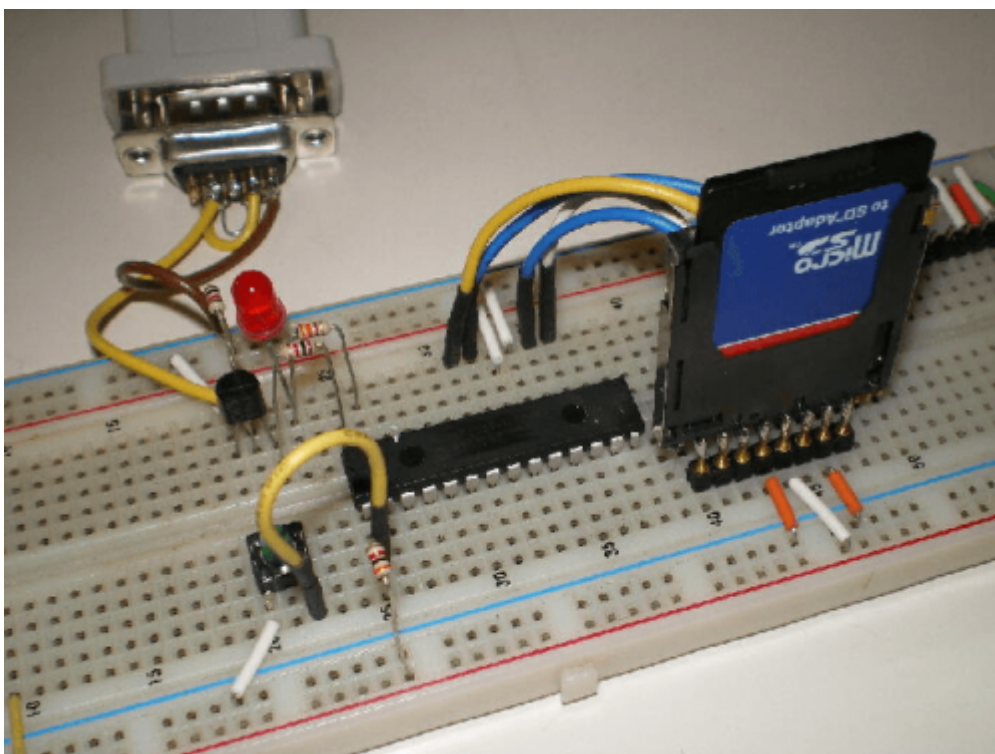
//*****
*

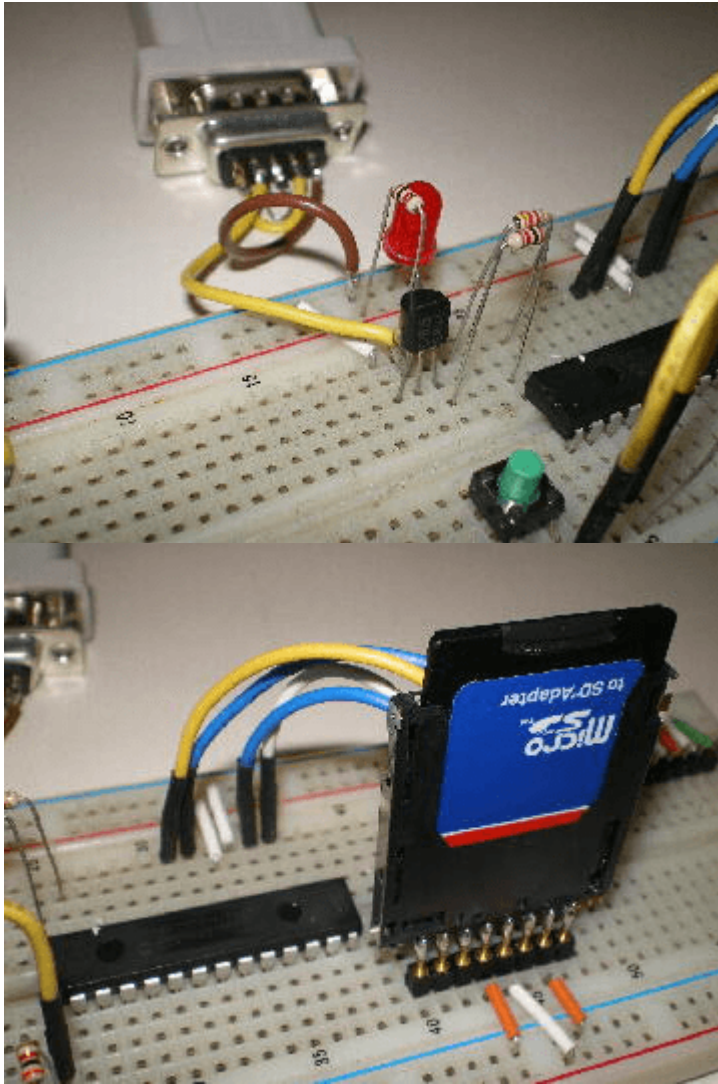
void main()
{
    //コンパレータは使用しない。
    CMCON = 0b00000111;
    //変換の設定
    ADCON1.PCFG3 = 1;
    ADCON1.PCFG2 = 0;
}
```

```
ADCON1.PCFG1 = 1;
ADCON1.PCFG0 = 1;
//クロックの設定
OSCCON.IRCF2 = 1;
OSCCON.IRCF1 = 1;
OSCCON.IRCF0 = 1;
//ポートの設定
TRISA = 0b0000000000;
TRISB = 0b00000001;
TRISC = 0b00000000;
PORTB.F7 = 1;
//□□□□□□□□の初期化
init_sdc();
//
bps = BPS9600;
//
while (1) {
    SwitchONcheck();
    LED = 1;
    rs232cProc();
    LED = 0;
}
}

//*****
*
```

## 動作確認

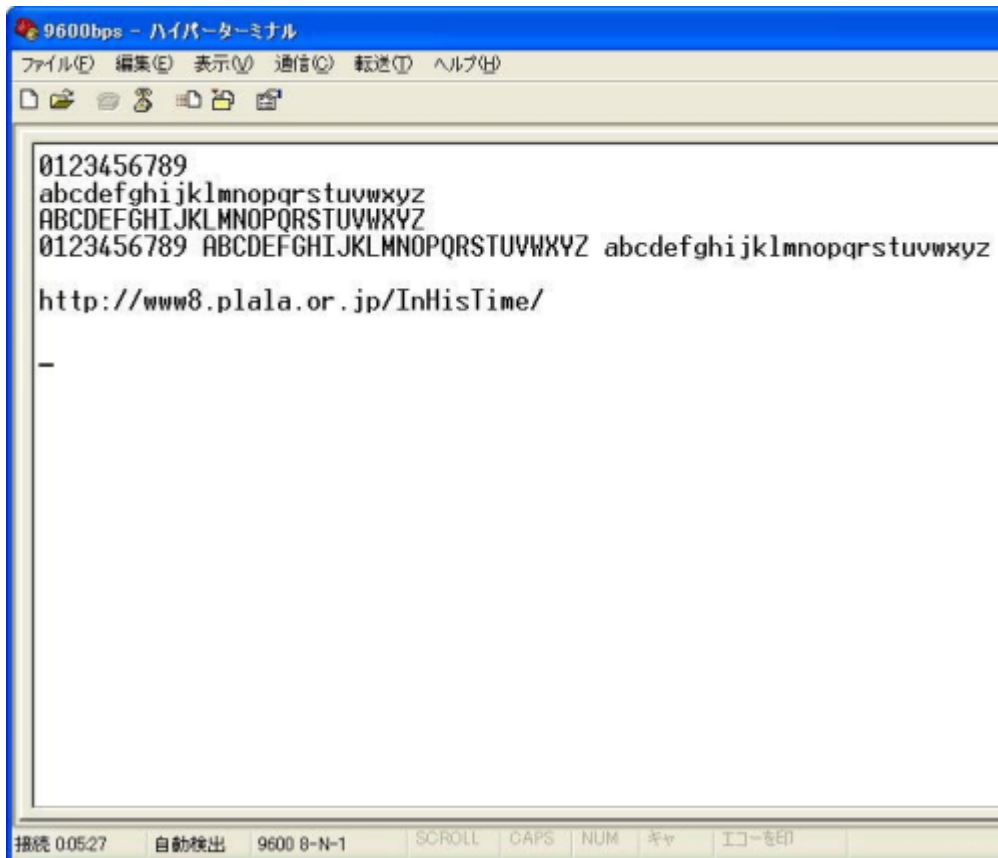




```
data - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
SBPS=9600
SSTART
0123456789
SCRLF
SSLEEP=1000
abcdefghijklmnopqrstuvwxy
SCRLF
SSLEEP=1000
ABCDEFGHIJKLMNPOQRSTUVWXYZ
SCRLF
SSLEEP=1000
0123456789 ABCDEFGHIJKLMNPOQRSTUVWXYZ abcdefghijklmnopqrstuvwxy
SCRLF
SCRLF
SSLEEP=1000
http://www.8.plala.or.jp/InHisTime/
SCRLF
SCRLF
SSLEEP=1000
SSTOP
```

送信データファイルの例

実行結果(ハイパーターミナルで確認)



如何ですか? RS232Cを使用する電子工作では、本ユニットが手元があれば、効率よく作業を進めることが出来ますね。

著作権表示 **copyright notice**

このページは稲崎様の閉鎖したHPのコピーで、著作権は稲崎様にあります。詳細 This page is a copy of Mr. Inasaki's closed website, and the copyright is held by him.[Details](#)

From:

<http://www.deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:otherpic:179>

Last update: **2025/10/17 14:29**

