

アナログデータロガーV4(SDカード対応)(PIC18F2550)

概要

前回製作した、アナログデータロガーV3(SDカード対応)は、SDカード制御の実験の意味もあって、かなりハードウェアを簡略化しました。

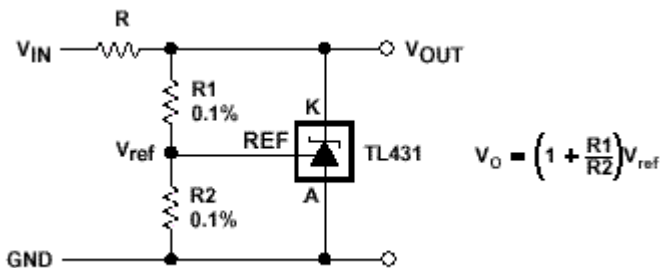
今回は、次の点を考慮し製作しました。

- データロガーだけの機能なので、28ピンのPIC18F2550を使用する。
- A/D変換(4チャンネル分)で取り込んだデータを、LCDに表示させる。
- PICとLCDの電圧は5V、SDカードの電圧は3.3Vとする。

動作原理

SDカード用の電圧を得るために、手持ちのTL431(可変シャントレギュレータ)を使いました。出力電圧は、次の式で求めることができます。

$$3.3V \approx (1 + (1k\Omega \div 3k\Omega)) \times 2.495V$$

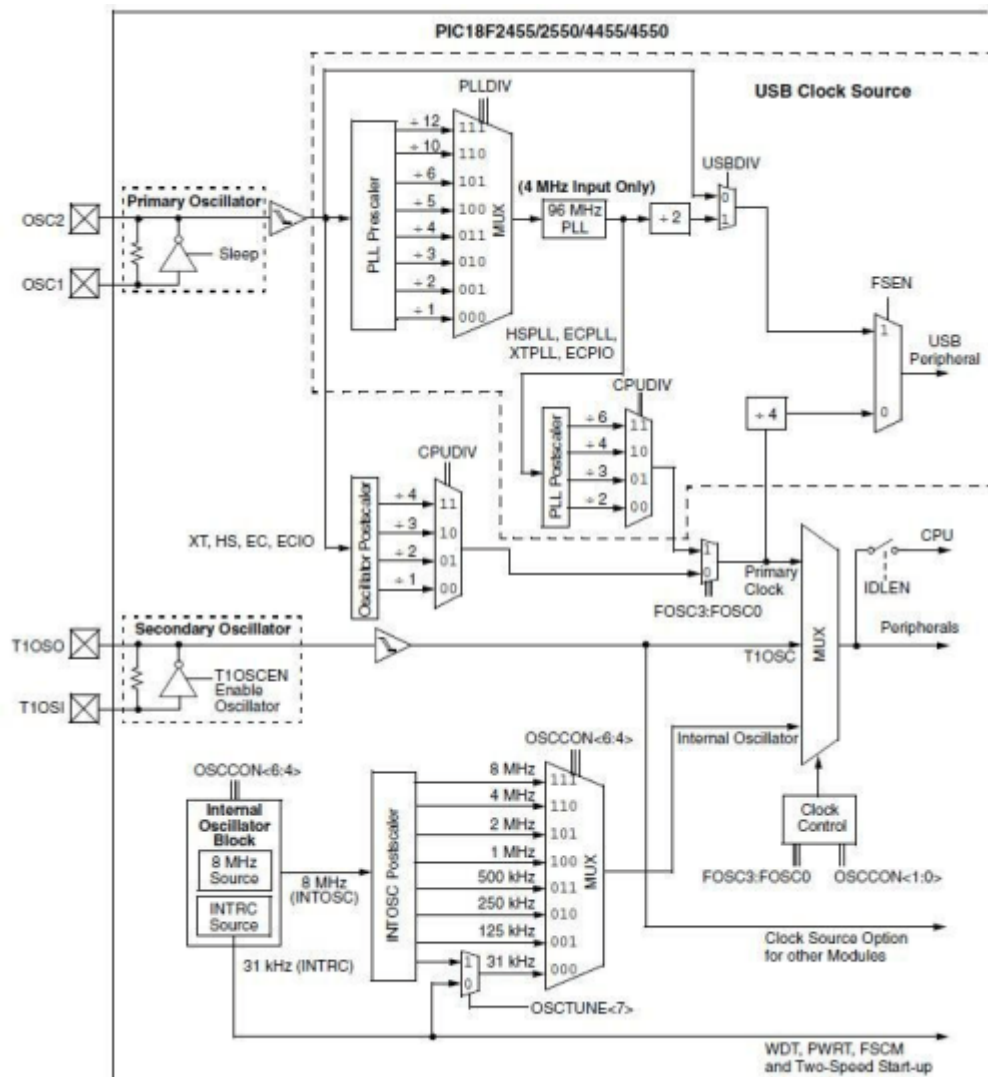


※3.3Vの3端子レギュレータがあればそれを使って

ください。

PIC18F2550は、最高クロック48MHzで動作させることが出来るので、外付けに20MHzのセラミック発振子(セラロック)を使用した場合には、次の項目を設定します。

- PLL-DIV $\div 5$
- クロックモード HS-PLL
- CPU-DIV $\div 2$

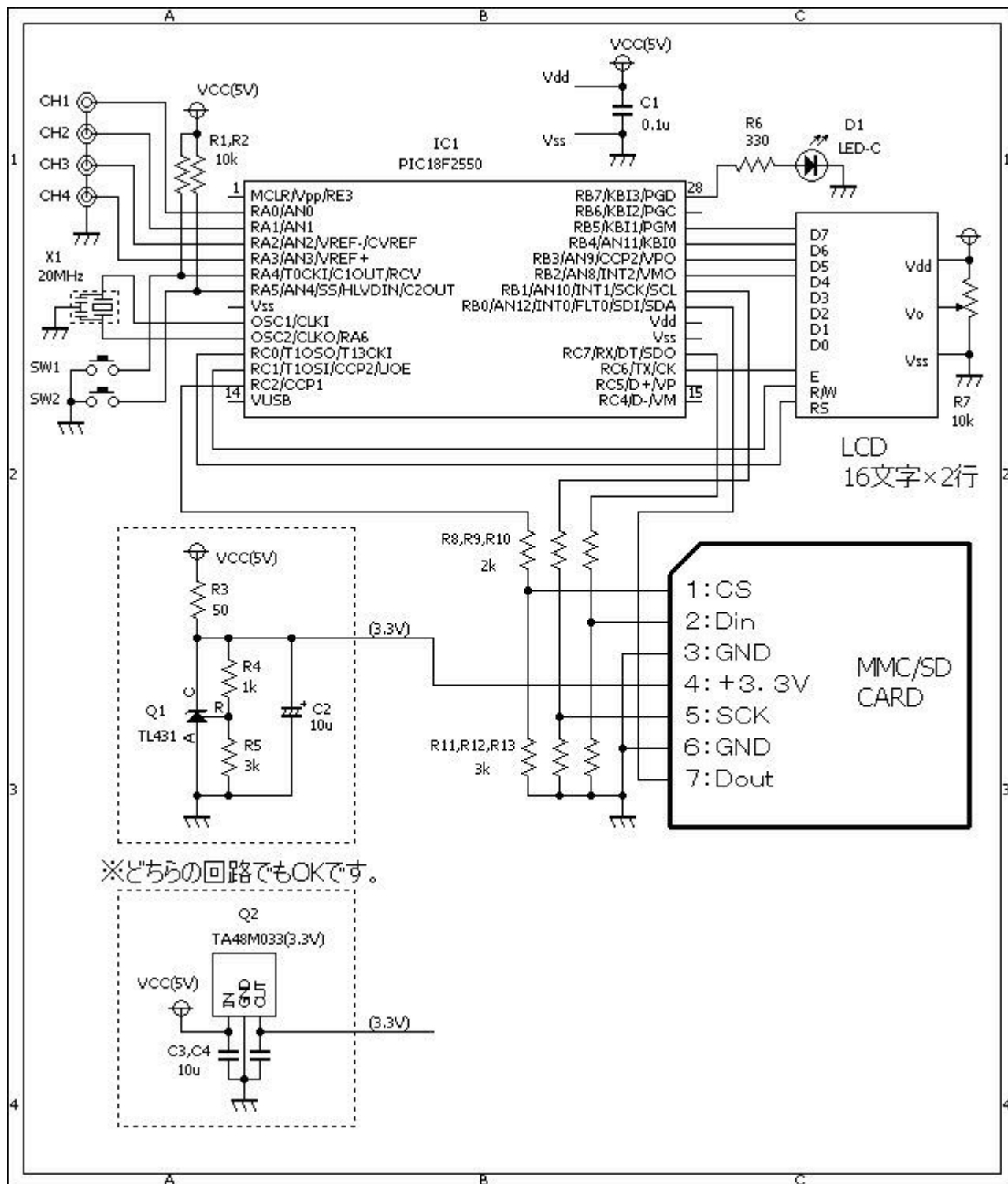


Input Oscillator Frequency	PLL Division (PLLDIV2:PLLDIV0)	Clock Mode (FOSC3:FOSC0)	MCU Clock Division (CPUDIV1:CPUDIV0)	Microcontroller Clock Frequency
20 MHz	-5 (100)	HS, EC, ECIO	None (00)	20 MHz
			+2 (01)	10 MHz
			+3 (10)	6.67 MHz
			+4 (11)	5 MHz
		HSPLL, ECPLL, ECPIO	+2 (00)	48 MHz
			+3 (01)	32 MHz
			+4 (10)	24 MHz
		+6 (11)	16 MHz	

<処理の流れ>

1. LCDを制御するための初期処理を行う。
2. 記録開始スイッチが押されるのを待つ。
3. 押されるとSDカードを制御するための初期処理を行う。
4. A/D変換で4CH分全てを取り込む。
5. 取り込んだデータを、LCDに表示する。
6. 取り込んだデータを、文字列に変換する(CSV形式)
7. 文字列を、SDカードに書き込む。(ファイル名は、LOG.CSVとする)
8. 記録停止スイッチが押されるまで4~7.を繰り返す。
9. 記録停止スイッチが押されるとSDカードに書き込んだ件数をLCDに表示する。
10. 2.に戻る。

回路図



ソースコード

[AnalogDataLoggerSDC.c](http://www.deepsky.jp/wiki/AnalogDataLoggerSDC.c)

```

//*****
*
/*

```

```
『アナログデータログ(□□□□□□カード対応)』
□■□□□変換の精度
・10ビット
チャンネル数
・4チャンネル
記録媒体
□□□□□カード
記録形式
□□□□□□形式
記録周期
約0.1秒
*/
//*****
*

#define          LED          PORTB.F7

#define          SW_START     PORTA.F4
#define          SW_STOP      PORTA.F5

#define          ON           1
#define          OFF          0

#define          CR           0x0d
#define          LF           0x0a

//*****
*

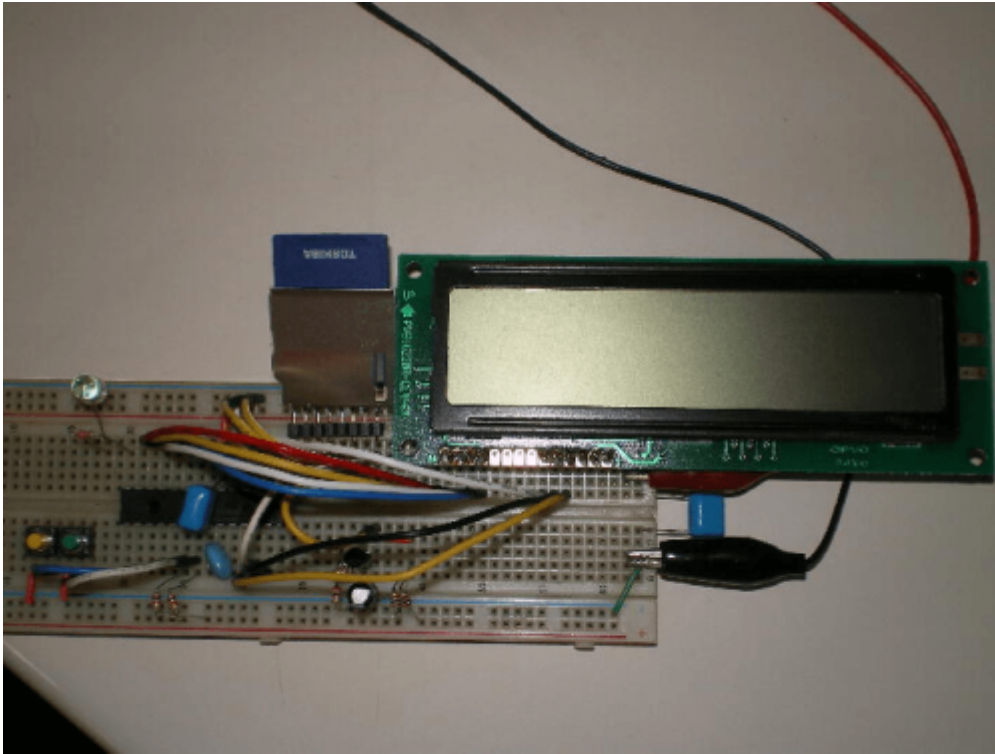
void main()
{
    //変数の定義
    static char    buf[50];
    short          cnt, channel, pnt;
    unsigned       ad;
    long           dataCnt;
    //□□□変換の設定
    ADCON1.PCFG3 = 1;
    ADCON1.PCFG2 = 0;
    ADCON1.PCFG1 = 1;
    ADCON1.PCFG0 = 1;
    //ポートの設定
    TRISA = 0b11111111;
    TRISB = 0b00000000;
    TRISC = 0b00000000;
    //□□□の初期化
    Lcd_Custom_Config(&PORTB,5,4,3,2,&PORTC,0,1,6);
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);
    Lcd_Custom_Cmd(LCD_CLEAR);
    for (cnt = 1; cnt <= 16; cnt++) {
        Lcd_Custom_Chr(1, cnt, 0xFF);
    }
}
```

```
    LED = ON;
    Delay_ms(50);
    LED = OFF;
    Delay_ms(50);
}
for (cnt = 1; cnt <= 16; cnt++) {
    Lcd_Custom_Chrc(2, cnt, 0xFF);
    LED = ON;
    Delay_ms(50);
    LED = OFF;
    Delay_ms(50);
}
Lcd_Custom_Cmd(LCD_CLEAR);
//
while (1) {
    //開始スイッチが押されるのをチェックする。
    while (SW_START == 1) {
        Delay_ms(10);
    }
    dataCnt = 0;
    //初期化
    Spi_Init_Advanced(MASTER_OSC_DIV64, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
    if (!Mmc_Fat_Init(&PORTC, 2)) {
        Spi_Init_Advanced(MASTER_OSC_DIV16, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
        Mmc_Fat_Assign("log.csv", 0xA0);
        Mmc_Fat_Rewrite();
        Mmc_Fat_Write("$START\r\n", 8);
        //停止スイッチが押されるまで処理を繰り返す。
        while (SW_STOP == 1) {
            pnt = 0;
            //4チャンネル分の変換を行い、表示する。
            for (channel = 0; channel < 4; channel++) {
                ad = Adc_Read(channel);
                ad = (double)ad * 4.8828125;
                WordToStr(ad, &buf[pnt]);
                switch (channel) {
                    case 0:
                        Lcd_Custom_Out(1, 1, "1:");
                        Lcd_Custom_Out(1, 3, &buf[pnt + 1]);
                        break;
                    case 1:
                        Lcd_Custom_Out(1, 9, "2:");
                        Lcd_Custom_Out(1, 11, &buf[pnt + 1]);
                        break;
                    case 2:
                        Lcd_Custom_Out(2, 1, "3:");
                        Lcd_Custom_Out(2, 3, &buf[pnt + 1]);
                        break;
                    case 3:
```

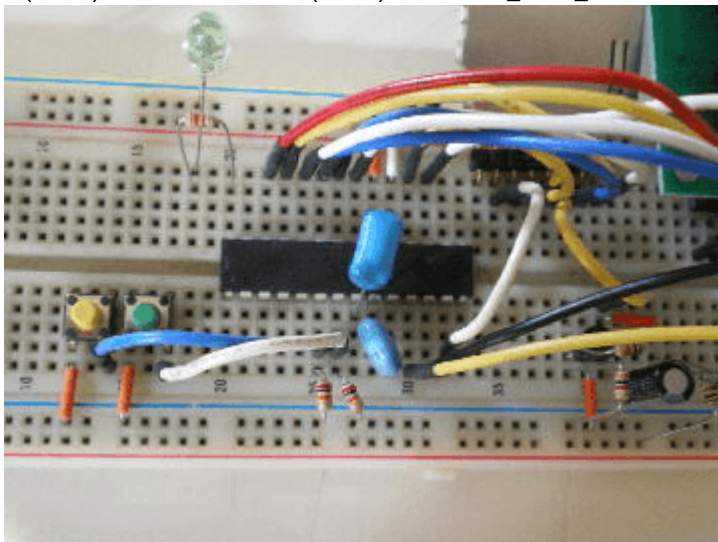
```
        Lcd_Custom_Out(2, 9, "4:");
        Lcd_Custom_Out(2, 11, &buf[pnt + 1]);
        break;
    }
    pnt += 6;
}
buf[5] = ',';
buf[11] = ',';
buf[17] = ',';
buf[23] = CR;
buf[24] = LF;
// 4チャンネル分まとめて[] [] [] []に書き込む。
Mmc_Fat_Write(buf, 25);
//
LED = ON;
Delay_ms(100);
LED = OFF;
//
dataCnt++;
}
Mmc_Fat_Write("$STOP\r\n", 7);
//書き込んだ件数を[] [] [] []に表示する。
LongToStr(dataCnt, buf);
Lcd_Custom_Cmd(LCD_CLEAR);
Lcd_Custom_Out(1, 1, buf);
//
for (cnt = 0; cnt < 10; cnt++) {
    LED = ON;
    Delay_ms(50);
    LED = OFF;
    Delay_ms(50);
}
}
}
} //~!

//*****
*
```

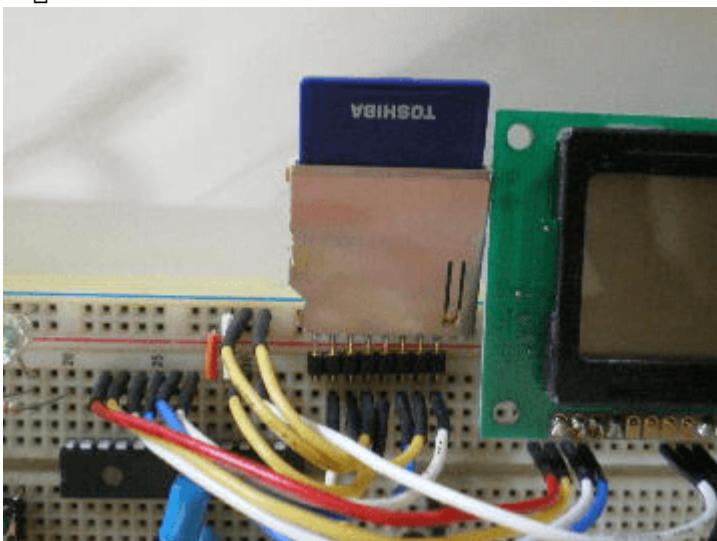
動作確認

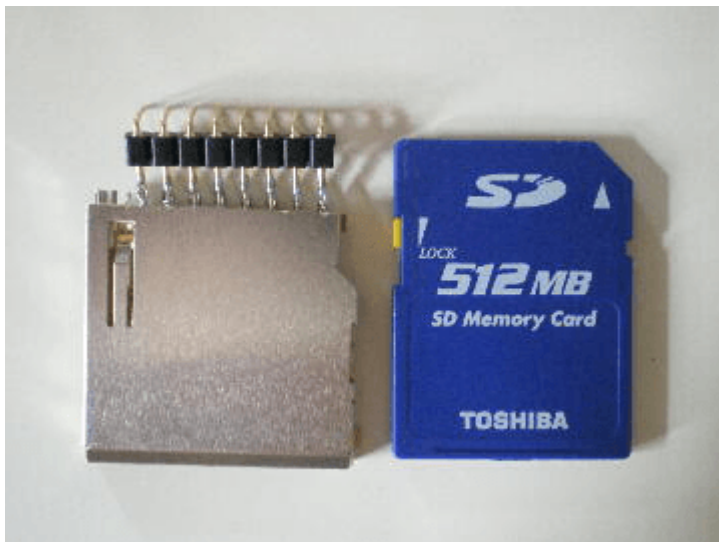


左側:開始(黄色)スイッチ、停止(緑色)スイッチ、LED、PIC18F2550、TL431 右側:PIC18F2550、SDカード



LCD

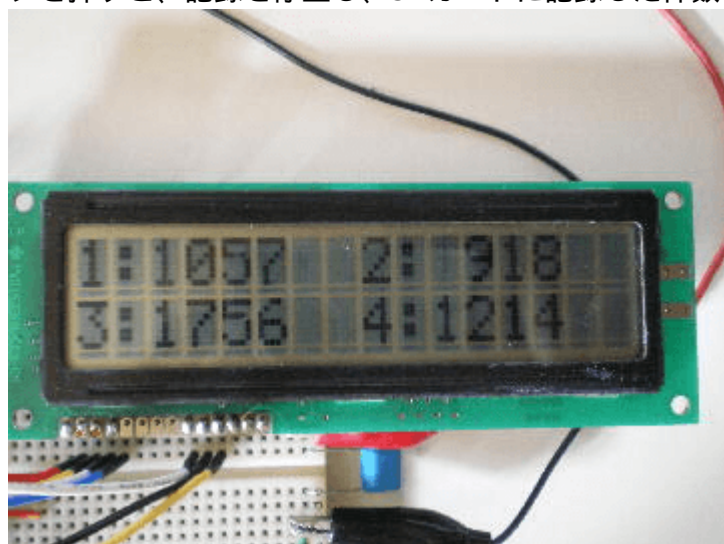


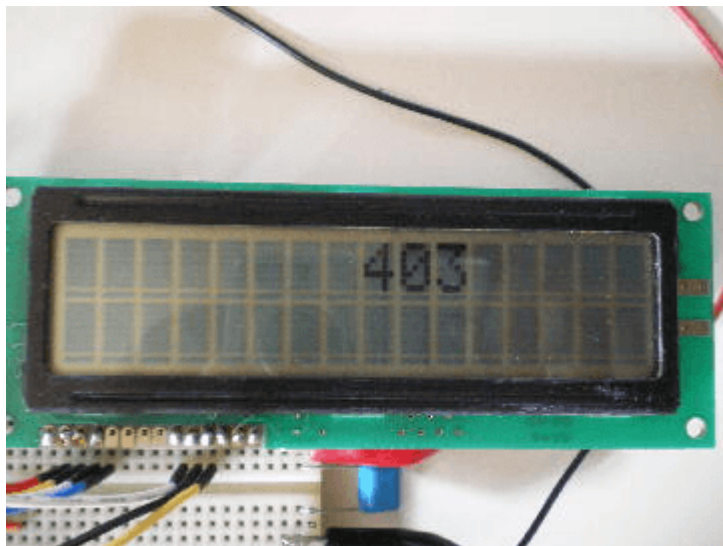


SDカードスロットと512MBのSDカードです。

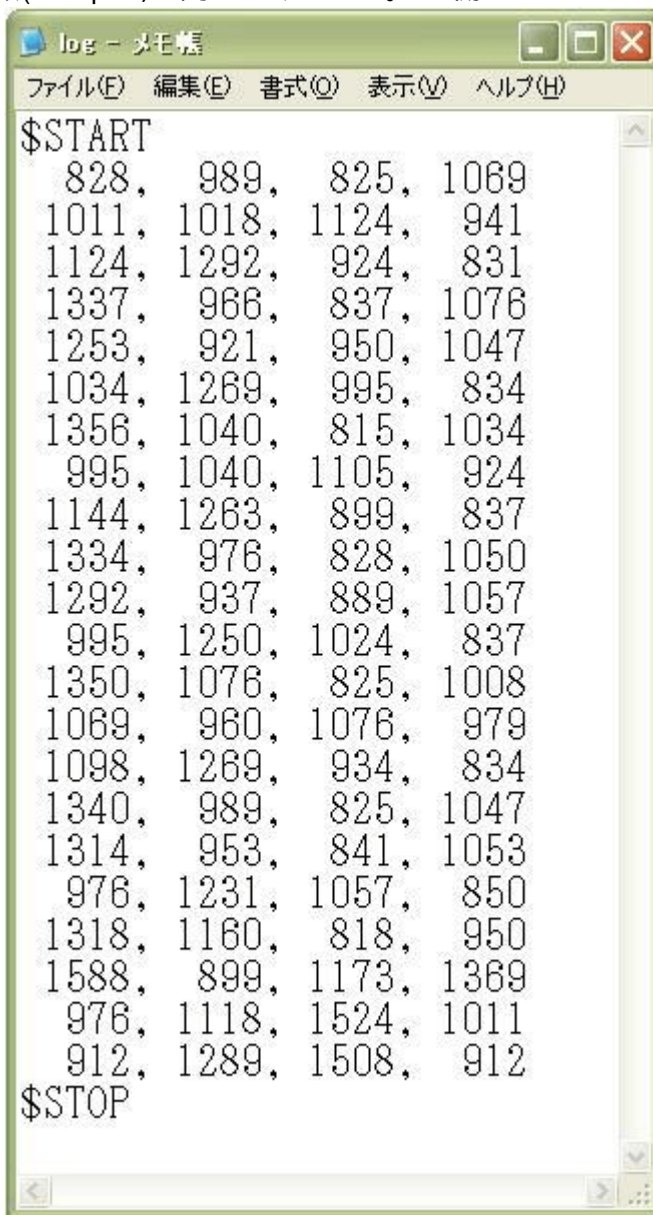


左側:開始スイッチを押すと、4チャンネル分のアナログデータを周期的に記録します。右側:停止スイッチを押すと、記録を停止し、SDカードに記録した件数を表示します。



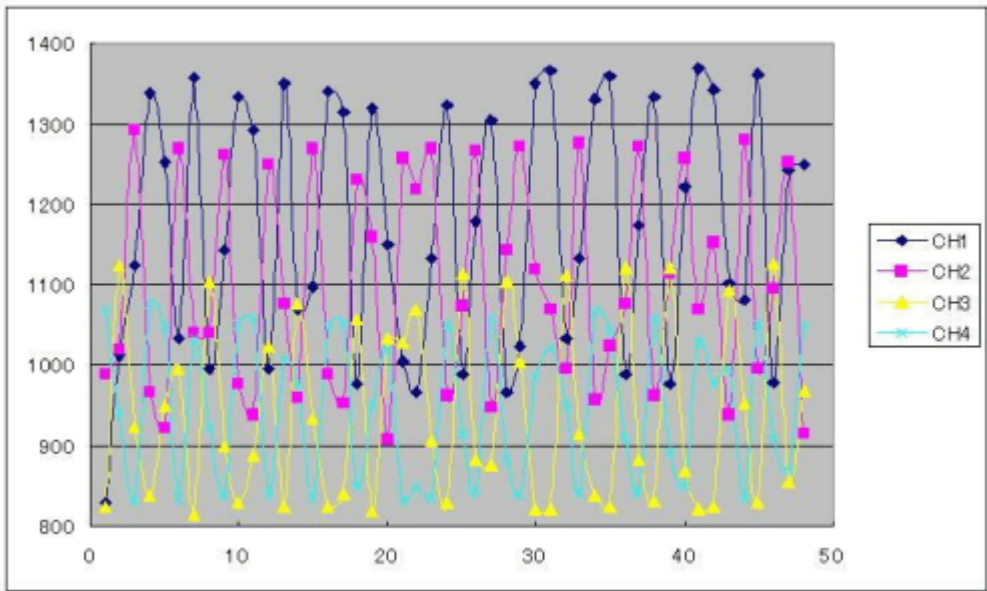


SDカードに保存したデータ(LOG.CSV)を、メモ帳(notepad)で開いてみました。左側か



らCH1CH2CH3CH4です。単位はmVです。

Excelでデータをグラフ表示させてみました。



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:174&rev=1588240140>

Last update: **2025/10/17 14:27**

