

10チャンネルデータロガー(PIC18F2320)

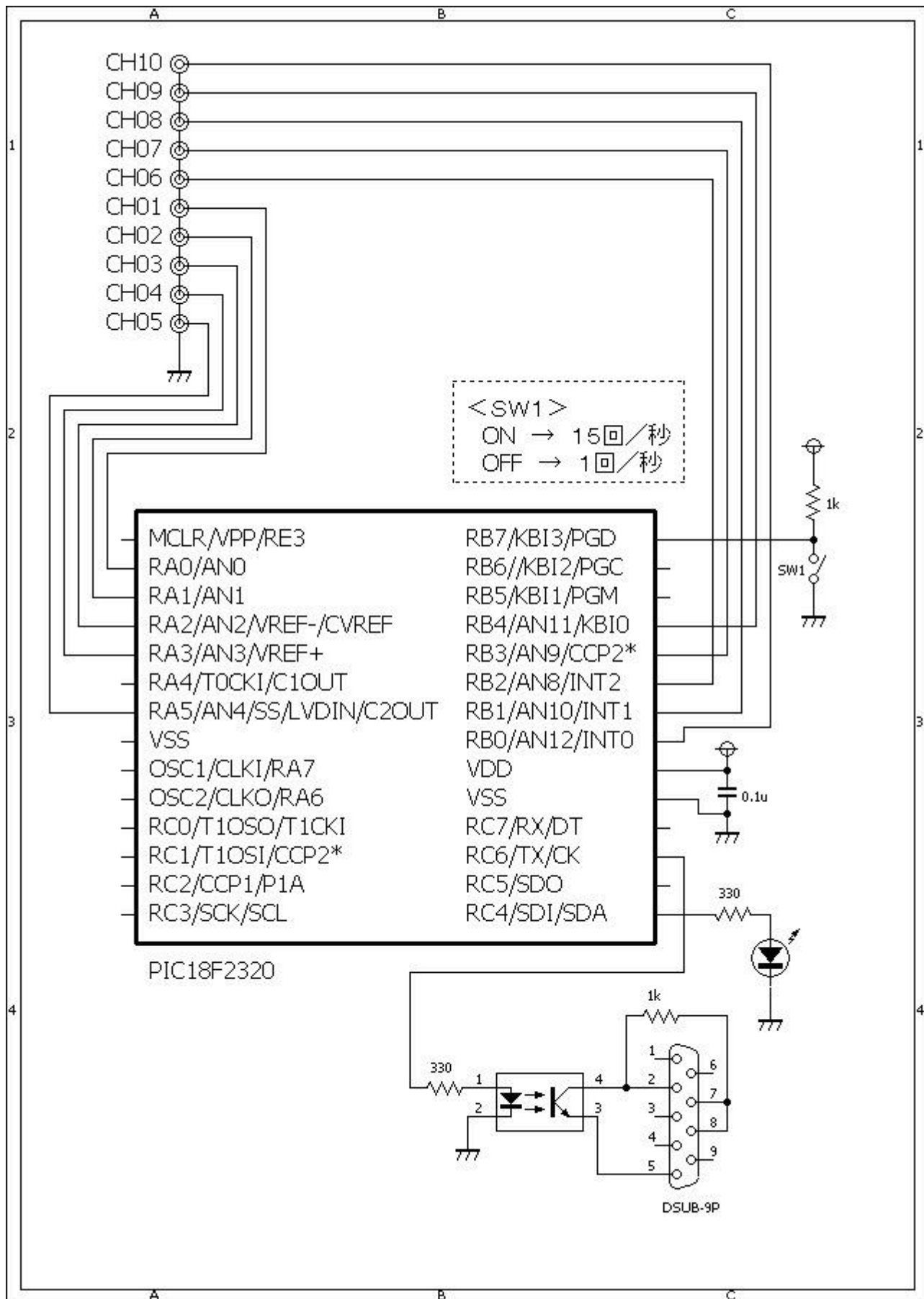
概要

初めてPIC18シリーズを使用しましたPIC18F2320を使用して、単純にAD変換したデータをRS232C経由でPCに送るだけのものを作成しました。尚、計測モードは次の2種類としました。

- 1秒間に1回、AD変換(10チャンネル分)してPCにデータを送る。
- 1秒間に15回、AD変換(10チャンネル分)してPCにデータを送る。

AD変換の精度は10ビットです。なので最小単位は約5mvとなります。精度を高めるには、電源電圧VCCを $1023 \times 5\text{mv} = 5115\text{mv}$ と安定化する必要があります。

回路図



ソースコード

DataLogger_10channel.c

```
//*****
*
/*
「10チャンネルデータロガー」

・10チャンネルのアナログデータを10bit変換し、10進数で送信し
ます。（秒間約5回のサンプリング）

*/
//*****
*

#define LED          PORTC.F4
#define SW           PORTB.F7

//*****
*

static unsigned short startCount;

//*****
*

void interrupt(){
    if (PIR1.TMR1IF == 1) {
        PIR1.TMR1IF = 0;
        //
        LED = ~LED;
        //
        if (startCount > 0)
            startCount--;
    }
}

//*****
*

void Usart_Write_String(char *buf)
{
    static int len, i;
    len = strlen(buf);
    for (i = 0; i < len; i++) {
        Usart_Write(buf[i]);
    }
}

//*****
*
```

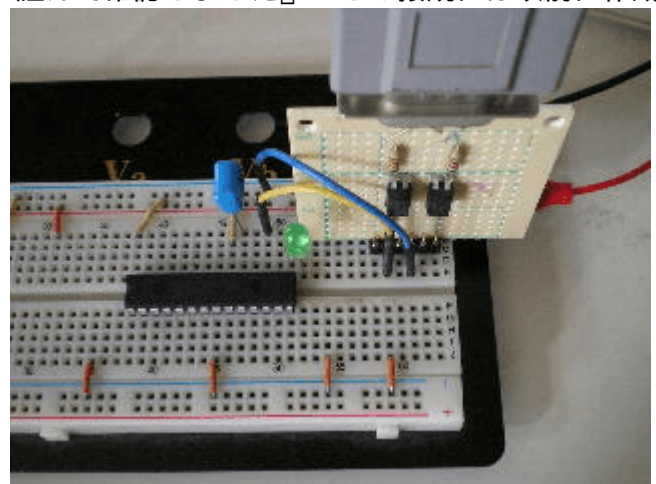
```
void main()
{
    static unsigned int    ad0, ad1, ad2, ad3, ad4, ad8, ad9,
ad10, ad11, ad12, cnt;
    static unsigned char    buf[10];
    //
    OSCCON.IRCF2=1;
    OSCCON.IRCF1=1;
    OSCCON.IRCF0=1;
    //
    TRISA = 0b11111111;
    TRISB = 0b11111111;
    TRISC = 0b00000000;
    //
    // TIMER1の設定
    PIE1.TMR1IE = 1;
    PIR1.TMR1IF = 0;
    TMR1L = 0;
    TMR1H = 0;
    T1CON.T1CKPS0 = 1;
    T1CON.T1CKPS1 = 1;
    T1CON.TMR1ON = 1;
    //
    startCount = 4;
    cnt = 0;
    //
    Usart_Init(9600);
    Delay_ms(100);
    Usart_Write_String("DataLogger(10channel)\r\n");
    // 割り込み(全体)の設定
    INTCON.PEIE = 1;
    INTCON.GIE = 1;
    //
    while(1) {
        if ((SW == 1) && (startCount != 0))
            continue;
        //
        cnt++;
        WordToStr(cnt, buf);
        Usart_Write_String(buf);
        Usart_Write_String(": ");
        ad0 = Adc_Read(0);
        ad1 = Adc_Read(1);
        ad2 = Adc_Read(2);
        ad3 = Adc_Read(3);
        ad4 = Adc_Read(4);
        ad8 = Adc_Read(8);
        ad9 = Adc_Read(9);
        ad10 = Adc_Read(10);
        ad11 = Adc_Read(11);
        ad12 = Adc_Read(12);
    }
```

```
WordToStr(ad0 * 5, buf);
Usart_Write_String(buf);
WordToStr(ad1 * 5, buf);
Usart_Write_String(buf);
WordToStr(ad2 * 5, buf);
Usart_Write_String(buf);
WordToStr(ad3 * 5, buf);
Usart_Write_String(buf);
WordToStr(ad4 * 5, buf);
Usart_Write_String(buf);
WordToStr(ad8 * 5, buf);
Usart_Write_String(buf);
WordToStr(ad9 * 5, buf);
Usart_Write_String(buf);
WordToStr(ad10 * 5, buf);
Usart_Write_String(buf);
WordToStr(ad11 * 5, buf);
Usart_Write_String(buf);
WordToStr(ad12 * 5, buf);
Usart_Write_String(buf);
Usart_Write_String("\r\n");
startCount = 4;
}
}
```

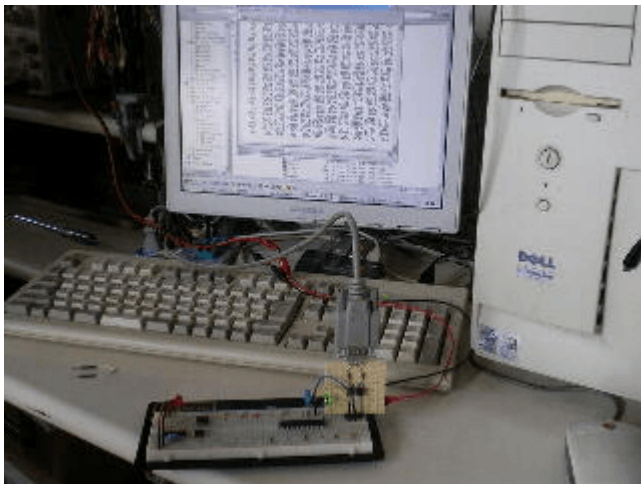
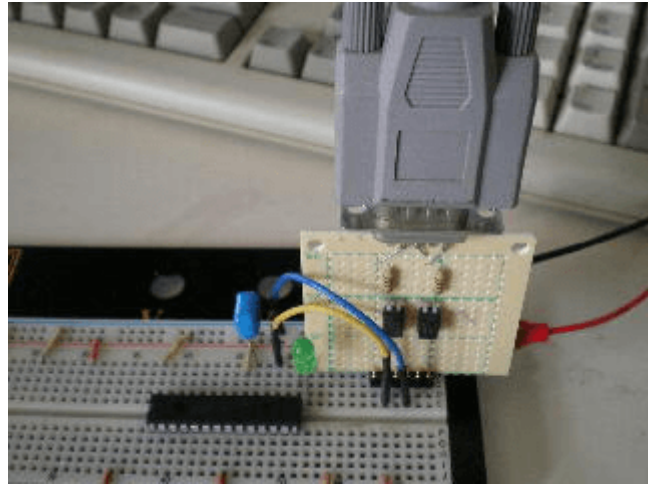
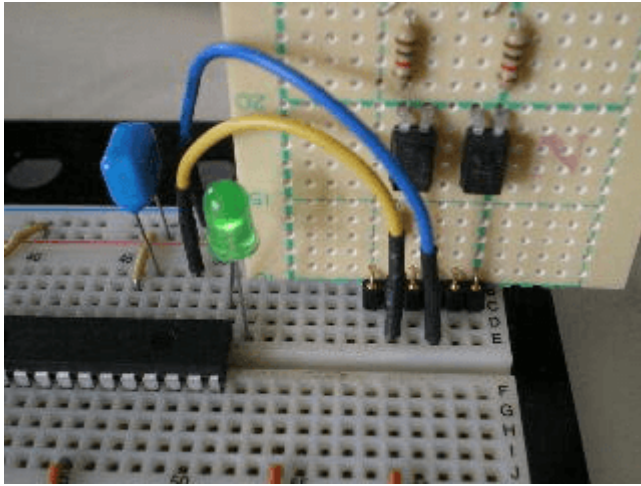
//*****
*

動作確認

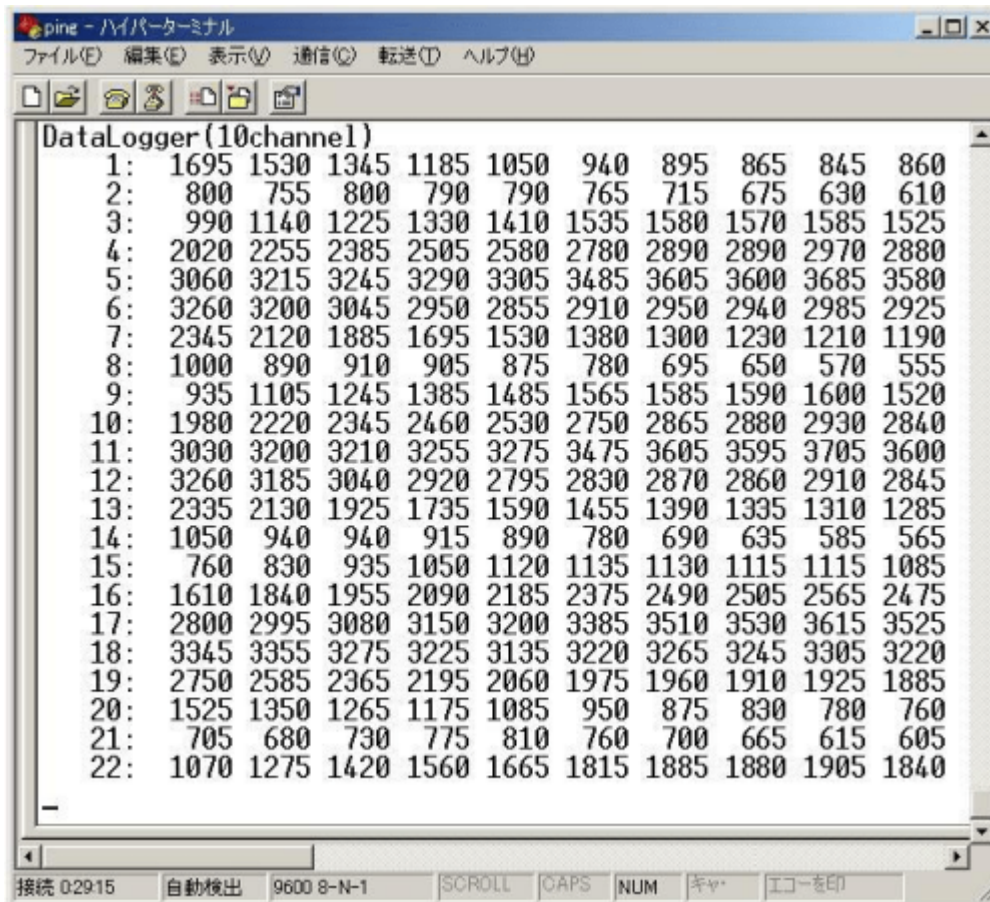
今回は部品点数がとても少ないのでブレッドボードに組んで確認しました。PCとの接続には以前に作成



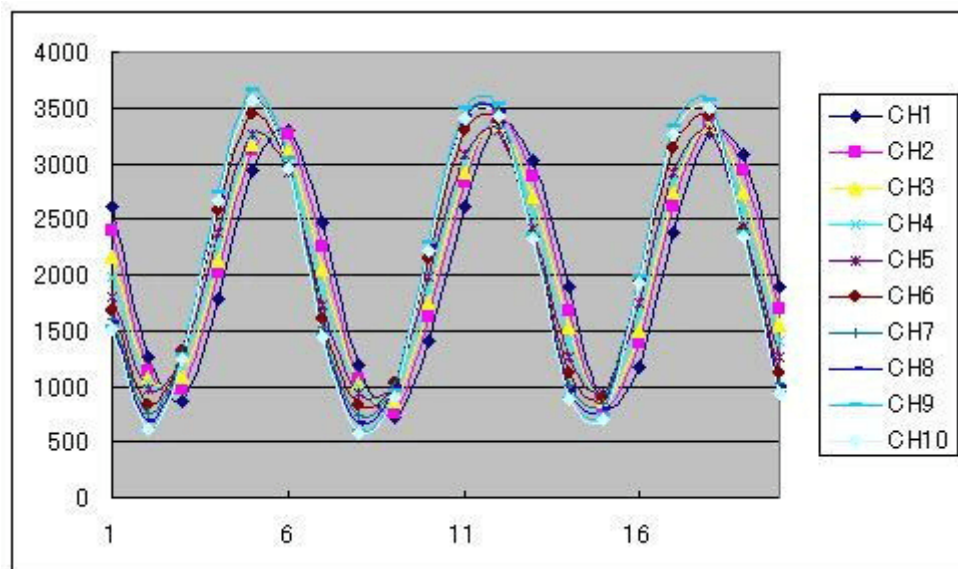
したRS232Cレベル変換ユニットを使用しています。



Windowsに標準で付いているターミナルソフト(ハイパーターミナル)を使用してデータを表示します。一番左のデータは、単なるカウント値です。その右からCH1~CH10の順番です。



取り込んだデータをExcelを使用してグラフ表示させてみました。入力端子をオープン状態にして計測したのでノイズ(商用の60Hz)が乗っています。入力端子を100k Ω 位の抵抗でプルダウンすると良い



と思います。

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://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:otherpic:159>

Last update: 2020/05/01 19:09

