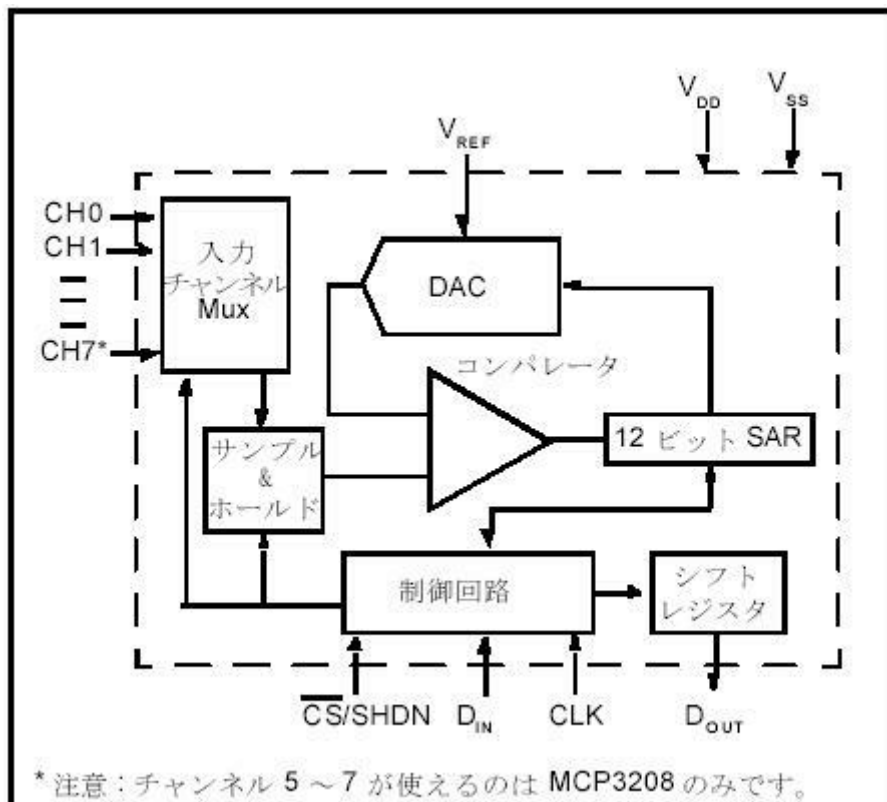


8チャンネルデータロガー(MCP3208)(PIC18F1320)

概要

秋月電子通商で購入した、12ビット精度の8chのADコンバータ [MCP3208]を使用してみました。現時点(2007年6月)で、400円で購入できます。下図は、機能ブロック図です。このICの制御は3線式なのでと



てもシンプルな構成です。

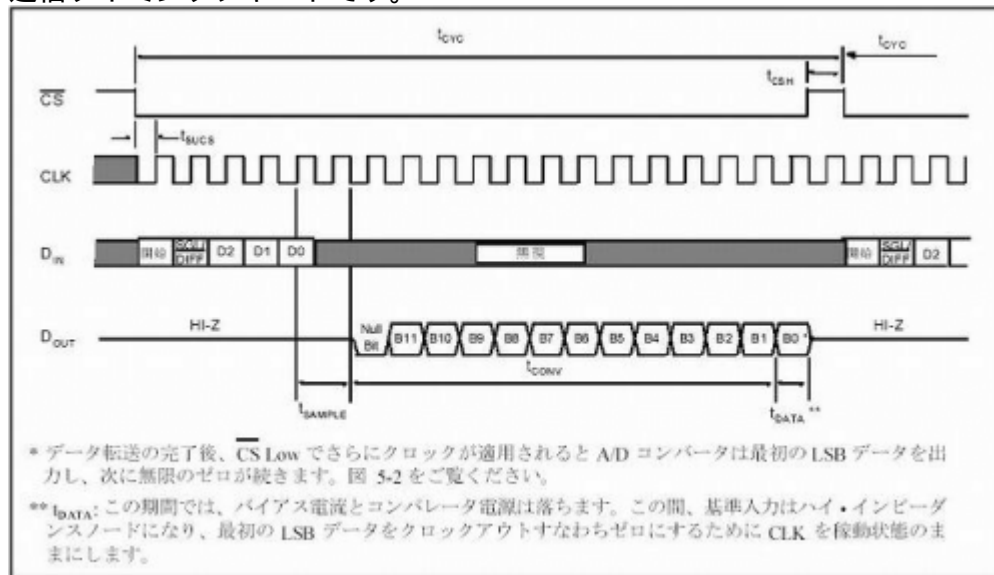
【特徴】

- 12 ビットの分解能
- DNL 最大 ± 1 LSB
- INL 最大 ± 1 LSB (MCP3204/3208-B)
- INL 最大 ± 2 LSB (MCP3204/3208-C)
- 4 入力チャンネル(MCP3204) または 8 入力チャンネル(MCP3208)
- アナログ入力はシングルエンドあるいは疑似差動入力ペアとしてプログラム可能
- オンチップのサンプル & ホールド
- SPI/R シリアル・インターフェース(モード0,0および1,1)
- 単一電源動作: 2.7 ~ 5.5V
- 100ksps max サンプルング速度(VDD = 5V 時)
- 50ksps max サンプルング速度(VDD = 2.7V 時)
- 低電力CMOS 技術
 - 待機電流500 nA typ (最高2 μ A)
 - 動作電流400 μ A max (5V 時)
- 広い温度範囲: -40 ~ +85
- PDIP/SOIC およびTSSOP パッケージで利用可能

【用途】

- センサー・インターフェース
- プロセス制御
- データ収集
- バッテリー駆動システム

通信タイミングチャートです。

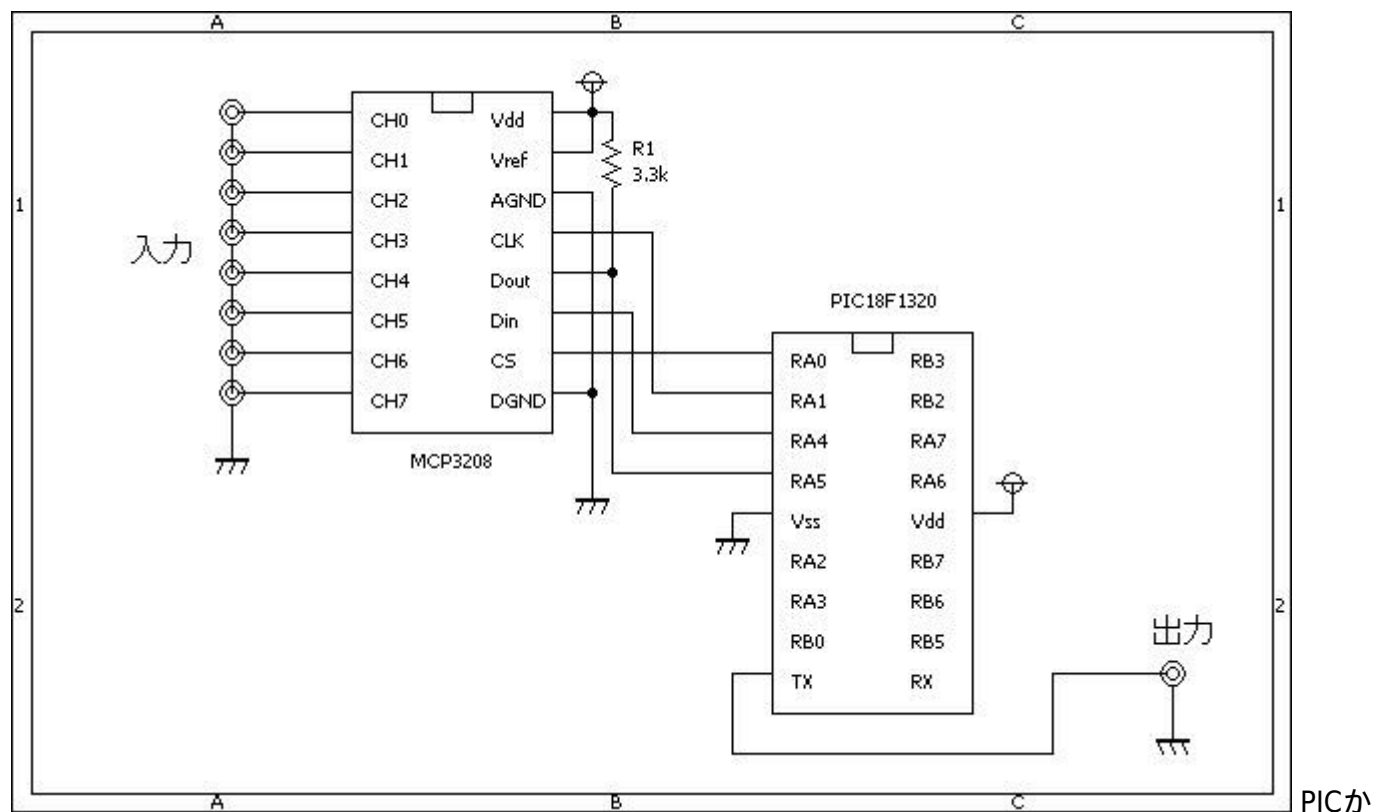


動作原理

特に難しい箇所はありません。PICとMCP3208を4本(CLK、Dout、Din、CS)で接続し、タイミングチャートに従って制御するだけです。そのAD変換したデータを文字列に変換してRS232CでPCへ送信します。

回路図

とてもシンプルな回路になりました。



らのTX出力は、RS232Cレベル変換ユニットを経由して、パソコンへRS232C経由で出力します。

ソースコード

MCP3208の制御部分(Read部分)は関数化しました。

[ad12bit.c](#)

```
//*****
*

#define MCP3208_CS      PORTA.F0    // output
#define MCP3208_CLK     PORTA.F1    // output
#define MCP3208_DIN     PORTA.F4    // output
#define MCP3208_DOUT    PORTA.F5    // input

//*****
*

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

```
//*****
*

unsigned int    Adc_Read_mcp3208(unsigned char ch)
{
    unsigned char    cnt;
    unsigned int      data;
    //
    MCP3208_CS = 0;
    //
    MCP3208_DIN = 1;    // start
    MCP3208_CLK = 1;
    asm    nop;
    MCP3208_CLK = 0;
    //
    MCP3208_DIN = 1;    // single
    MCP3208_CLK = 1;
    asm    nop;
    MCP3208_CLK = 0;
    //
    MCP3208_DIN = (ch >> 2) & 0x01;    // D2
    MCP3208_CLK = 1;
    asm    nop;
    MCP3208_CLK = 0;
    //
    MCP3208_DIN = (ch >> 1) & 0x01;    // D1
    MCP3208_CLK = 1;
    asm    nop;
    MCP3208_CLK = 0;
    //
    MCP3208_DIN = ch & 0x01;    // D0
    MCP3208_CLK = 1;
    asm    nop;
    MCP3208_CLK = 0;
    //
    MCP3208_DIN = 0;    // sample
    MCP3208_CLK = 1;
    asm    nop;
    MCP3208_CLK = 0;
    //
    MCP3208_DIN = 0;    // null
    MCP3208_CLK = 1;
    asm    nop;
    MCP3208_CLK = 0;
    //
    data = 0;    // data
    for (cnt = 0; cnt < 12; cnt++) {
        MCP3208_DIN = 0;
        MCP3208_CLK = 1;
        if (MCP3208_DOUT == 1) {
            data |= 0x01;
        }
    }
}
```

```
        } else {
            data |= 0x00;
        }
        data = data << 1;
        MCP3208_CLK = 0;
    }
    //
    MCP3208_CS = 1;
    MCP3208_CLK = 1;
    asm    nop;
    MCP3208_CLK = 0;
    //
    data = data >> 1;
    //
    return(data);
}

//*****
*

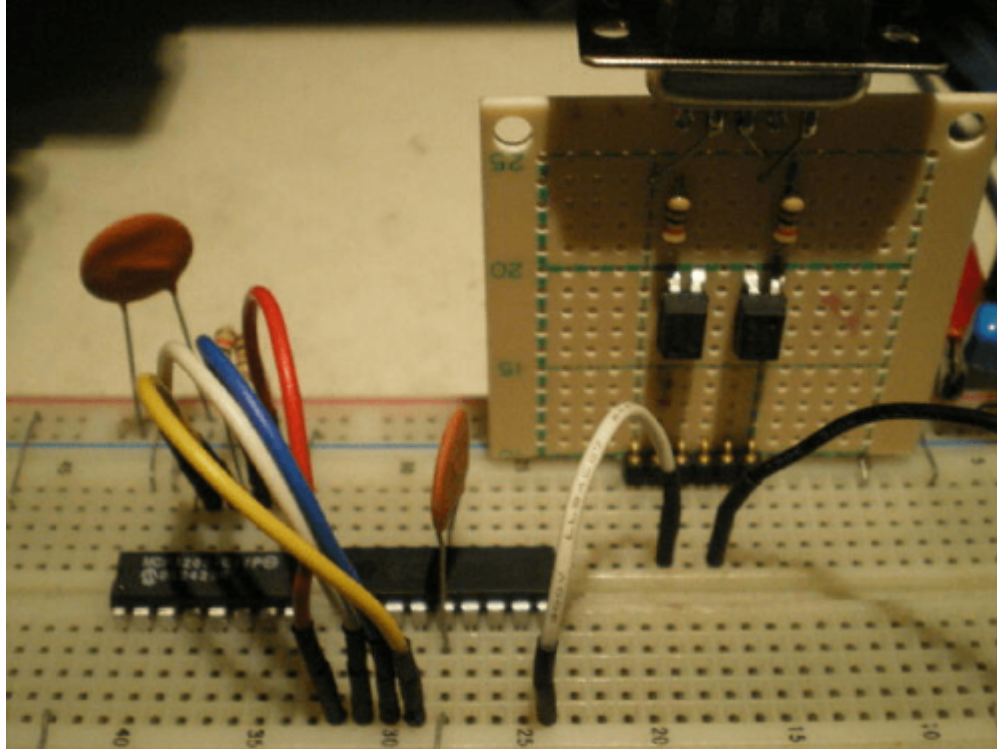
void    main()
{
    unsigned    char    buf[20], cnt;
    unsigned    int     data;
    OSCCON.IRCF2 = 1;
    OSCCON.IRCF1 = 1;
    OSCCON.IRCF0 = 1;
    ADCON1 = 0b11111111;
    TRISA = 0b00100000;
    TRISB.F1 = 0;
    TRISB.F4 = 1;
    T0CON.T0CS = 0;
    //
    Usart_Init(9600);
    Delay_ms(100);
    //
    MCP3208_CS = 1;
    MCP3208_CLK = 0;
    while (1) {
        for (cnt = 0; cnt < 8; cnt++) {
            data = Adc_Read_mcp3208(cnt);
            data = data * 1.2;
            IntToStr(data, buf);
            Usart_Write_String(&buf[1]);
            Usart_Write_String("mv");
        }
        Usart_Write_String("\r\n");
    }
}

//*****
```

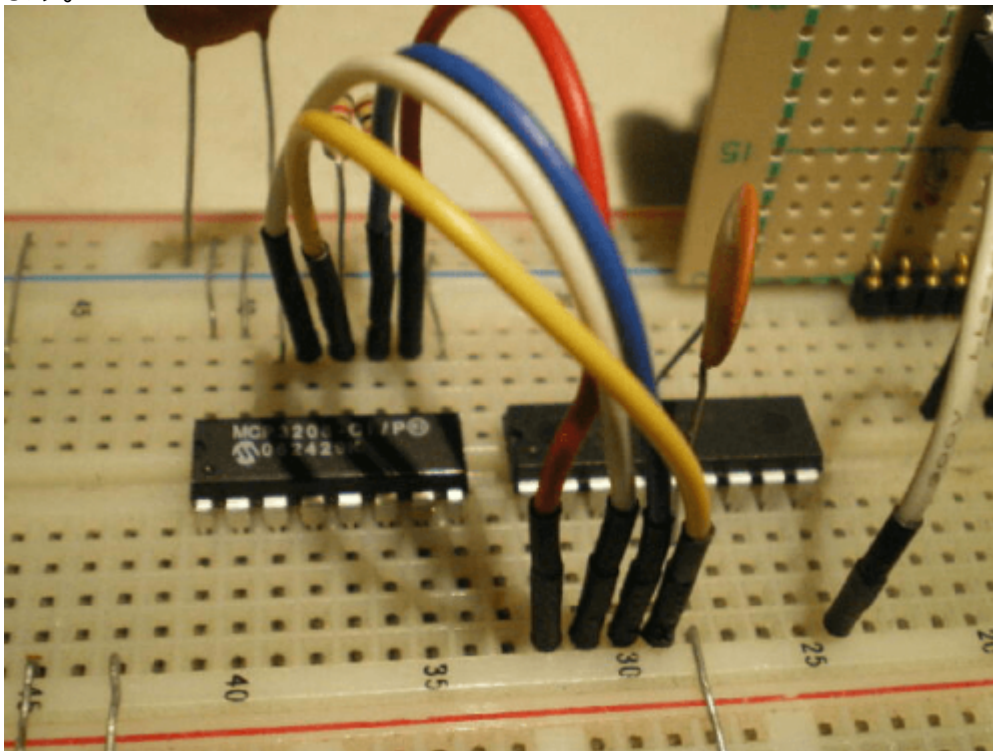
*

動作確認

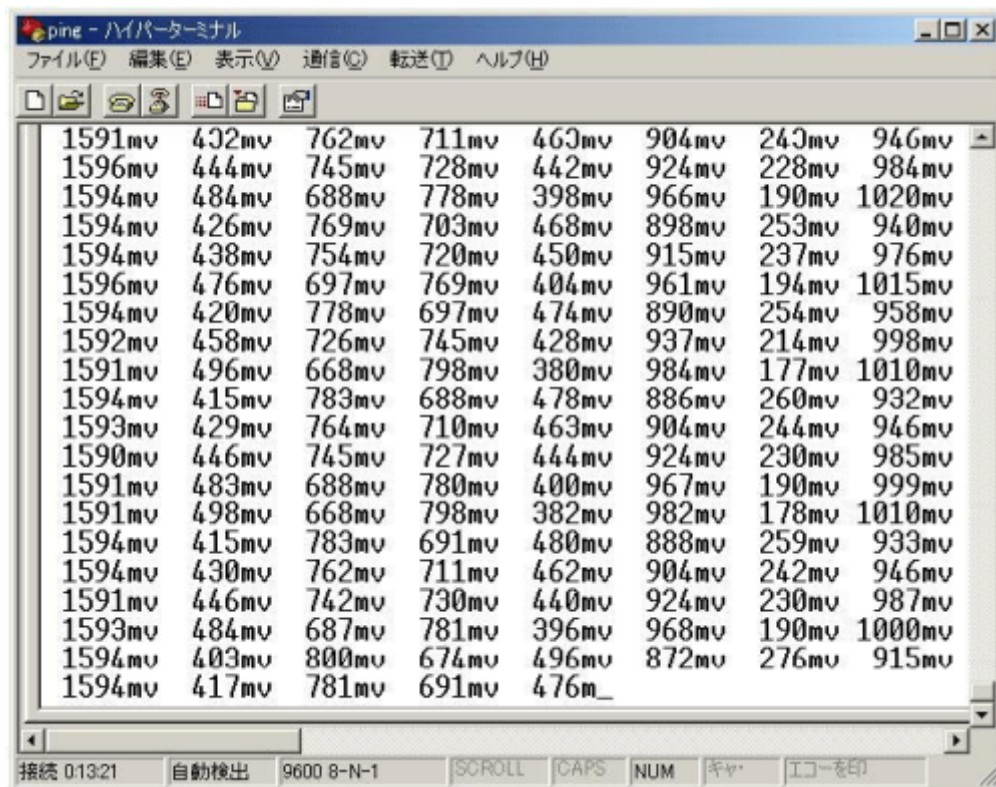
いつものブレッドボードで確認しました。PICとMCP3208の間は、4本線なので、とてもすっきりしてい



ます。



PCのハイパーターミナルでデータを受信し表示しました。1行に左からCH0、CH1、CH2.....CH7の順に表



1591mv	432mv	762mv	711mv	463mv	904mv	243mv	946mv
1596mv	444mv	745mv	728mv	442mv	924mv	228mv	984mv
1594mv	484mv	688mv	778mv	398mv	966mv	190mv	1020mv
1594mv	426mv	769mv	703mv	468mv	898mv	253mv	940mv
1594mv	438mv	754mv	720mv	450mv	915mv	237mv	976mv
1596mv	476mv	697mv	769mv	404mv	961mv	194mv	1015mv
1594mv	420mv	778mv	697mv	474mv	890mv	254mv	958mv
1592mv	458mv	726mv	745mv	428mv	937mv	214mv	998mv
1591mv	496mv	668mv	798mv	380mv	984mv	177mv	1010mv
1594mv	415mv	783mv	688mv	478mv	886mv	260mv	932mv
1593mv	429mv	764mv	710mv	463mv	904mv	244mv	946mv
1590mv	446mv	745mv	727mv	444mv	924mv	230mv	985mv
1591mv	483mv	688mv	780mv	400mv	967mv	190mv	999mv
1591mv	498mv	668mv	798mv	382mv	982mv	178mv	1010mv
1594mv	415mv	783mv	691mv	480mv	888mv	259mv	933mv
1594mv	430mv	762mv	711mv	462mv	904mv	242mv	946mv
1591mv	446mv	742mv	730mv	440mv	924mv	230mv	987mv
1593mv	484mv	687mv	781mv	396mv	968mv	190mv	1000mv
1594mv	403mv	800mv	674mv	496mv	872mv	276mv	915mv
1594mv	417mv	781mv	691mv	476m			

示しています。

高速なAD変換を要する場合は、高速AD変換ユニット(AD7820)を...(但し8ビット分解能) ◎PICの標準では10ビット分解能を... 分解能が更に必要であれば、今回のMCP3208の12ビットを... 如何ですか? 夫々、用途に応じて使い分けをして頂ければ宜しいかと。

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:166>

Last update: **2020/05/01 19:09**

