

簡易電圧計(高分解能22ビット)

概要

PICに搭載されているA/D変換モジュールの殆どが、10ビットの精度(分解能)を持っています。通常、製作する電子工作物では、これで十分です。

しかし、少し精度の高い測定器などを製作する場合には、10ビットの分解能では不十分な場合が発生します。

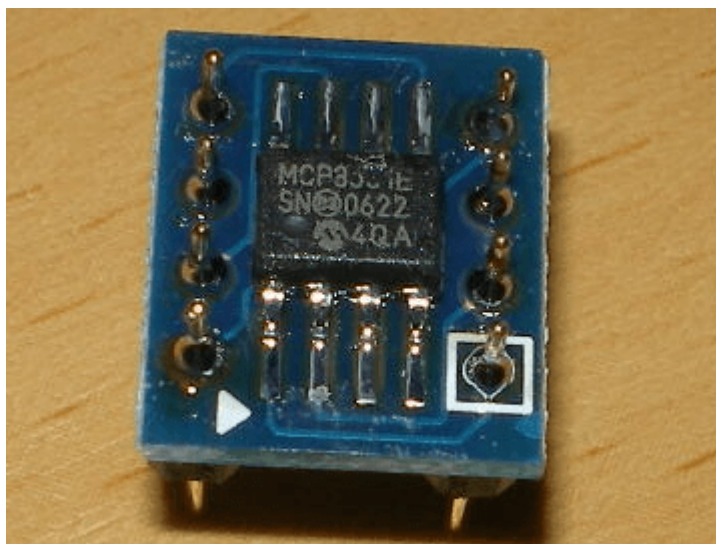
そこで今回は、22ビットの精度(分解能 $1\mu\text{V}$)を持つ、ローコストなA/D変換モジュール(MCP3551)を使用して電圧計を製作してみました。

<仕様> ◎MCP3551使用時

- 分解能= $0.9765625\mu\text{V} \dots 4096\text{mV} \div 22\text{ビット}(4194304)$
- 変換速度=約100msec
- 参照電圧=4096mV
- 入力=1チャンネル(CH1)

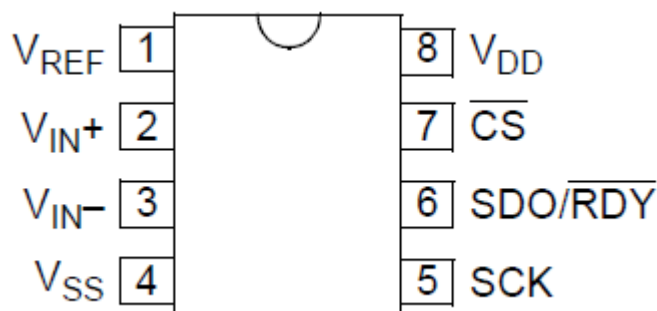
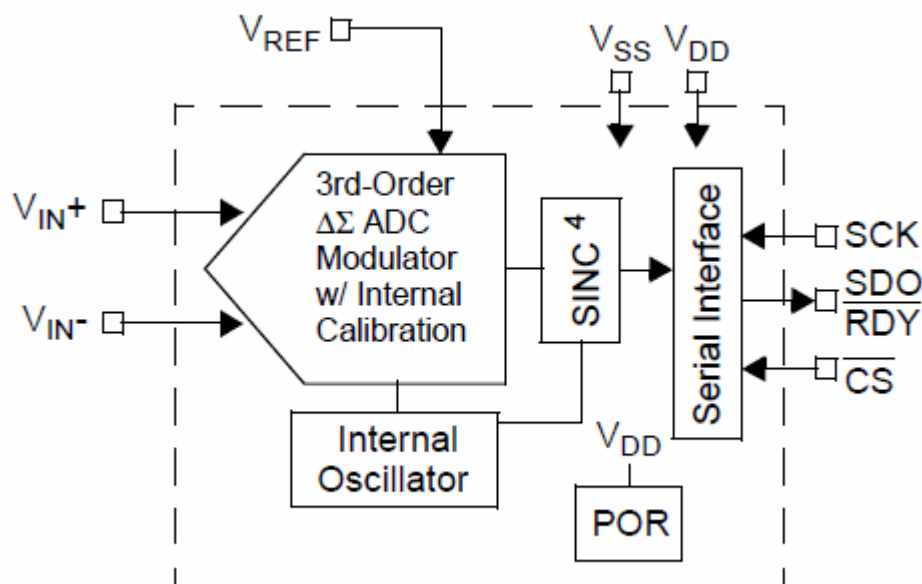
◎PIC内蔵ADC使用時

- 分解能= $4\text{mV} \dots 4096\text{mV} \div 10\text{ビット}(1024)$
- 変換速度=約0.1msec
- 参照電圧=4096mV
- 入力=1チャンネル(CH2)



<MCP3551の概観(IC変換基板に実装)>

<MCP3551のブロックダイアグラム>



<MCP3551のピン配置>

動作原理

外付けのA/D変換モジュール(MCP3551)とPIC内蔵のA/D変換モジュールで、入力信号をA/D変換で取り込み、LCDに表示させます。CH1(MCP3551使用)は、“**μV単位**”で電圧値を表示します。CH2(PIC内蔵モジュール使用)は、“**mV単位**”で電圧値を表示します。

動作原理(ハードウェア)

◎A/D変換

- 22ビット精度にはMCP3551を使用します。
- 10ビット精度にはPIC16F88の内蔵モジュールを使用します。

◎A/D変換参照電圧

- 多回転ボリューム(VR1)を使用して4096mVの参照電圧を得ます。

精度を高める場合には、専用のIC(例えばMCP1541等)を使用してください。

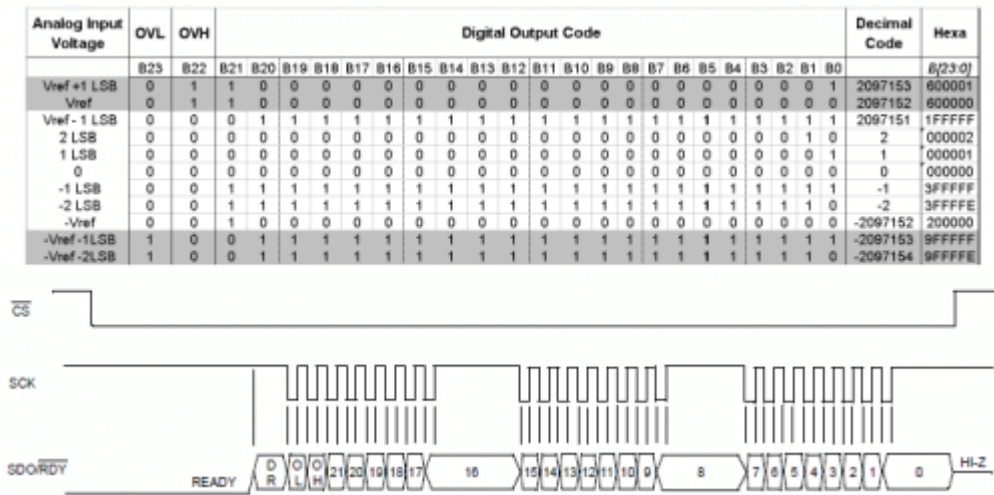
動作原理(ソフトウェア)

◎MCP3551の初期化処理

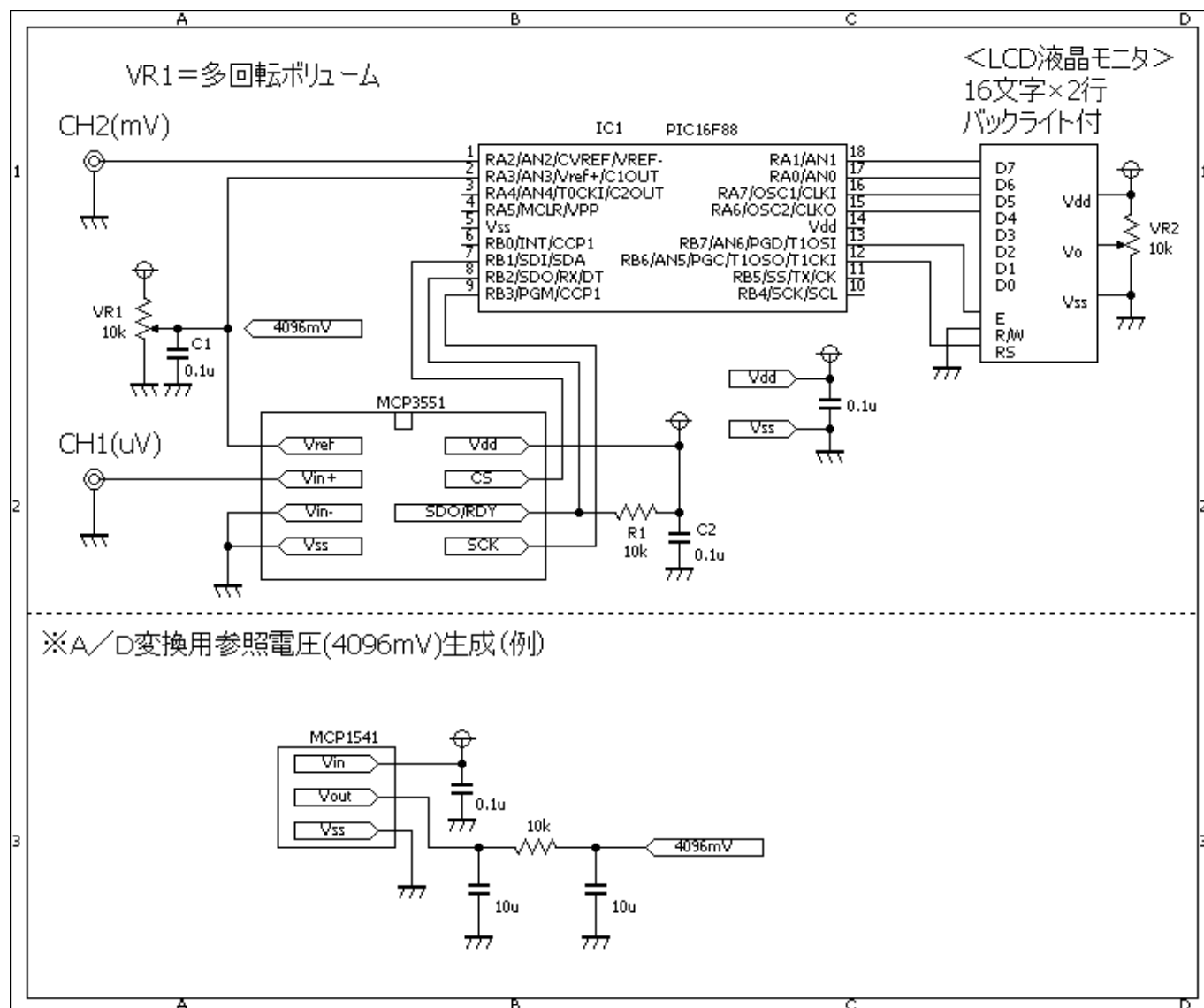
- CS,SDO/RDY,SCKに接続するポートの入出力モードを設定します。
- CS=1(high)□SCK=1(high)にします。

◎MCP3551からのデータの取り込み

- 下図のシーケンスに従って□A/D変換されたデータを取り込みます。



回路図



ソースコード

[volt_meter_mcp3551_22bits.c](#)

```
//*****
*
/*
<簡易電圧計（高分解能22ビット）>
*/
//*****
*
// ■ マクロ定義
#define BYTE      unsigned short
#define WORD      unsigned int
#define DWORD     unsigned long
//LCD
sbit LCD_RS at RB6_bit;
sbit LCD_EN at RB7_bit;
sbit LCD_D7 at RA1_bit;
```

```
sbit LCD_D6 at RA0_bit;
sbit LCD_D5 at RA7_bit;
sbit LCD_D4 at RA6_bit;
sbit LCD_RS_Direction at TRISB6_bit;
sbit LCD_EN_Direction at TRISB7_bit;
sbit LCD_D7_Direction at TRISA1_bit;
sbit LCD_D6_Direction at TRISA0_bit;
sbit LCD_D5_Direction at TRISA7_bit;
sbit LCD_D4_Direction at TRISA6_bit;
//
#define INPUT_MODE 1
#define OUTPUT_MODE 0
//*****
*
// ■ 関数宣言
extern void    main();
extern long    mcp3551_read();
extern void    mcp3551_init();
//*****
*
// ■ メイン関数
void    main()
{
    double    v1;
    WORD      v2;
    short     cnt;
    char      buf1[16], buf2[16];
    char      *space = "          ";
    //
    OSCCON = 0b01110000;
    ANSEL  = 0b00000100;
    TRISA  = 0b11111111;
    TRISB  = 0b00000100;
    //
    mcp3551_init();
    ADCON1.VCFG1 = 1;
    ADCON1.VCFG0 = 0;
    ADC_Init();
    //
    Lcd_Init();
    Lcd_Cmd(_LCD_CURSOR_OFF);
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1, 1, "Voltmeter(22bit)");
    Delay_ms(1000);
    Lcd_Cmd(_LCD_CLEAR);
    //
    while (1) {
        //測定
        v1 = mcp3551_read();
        v2 = ADC_Get_Sample(2);
        //補正
```

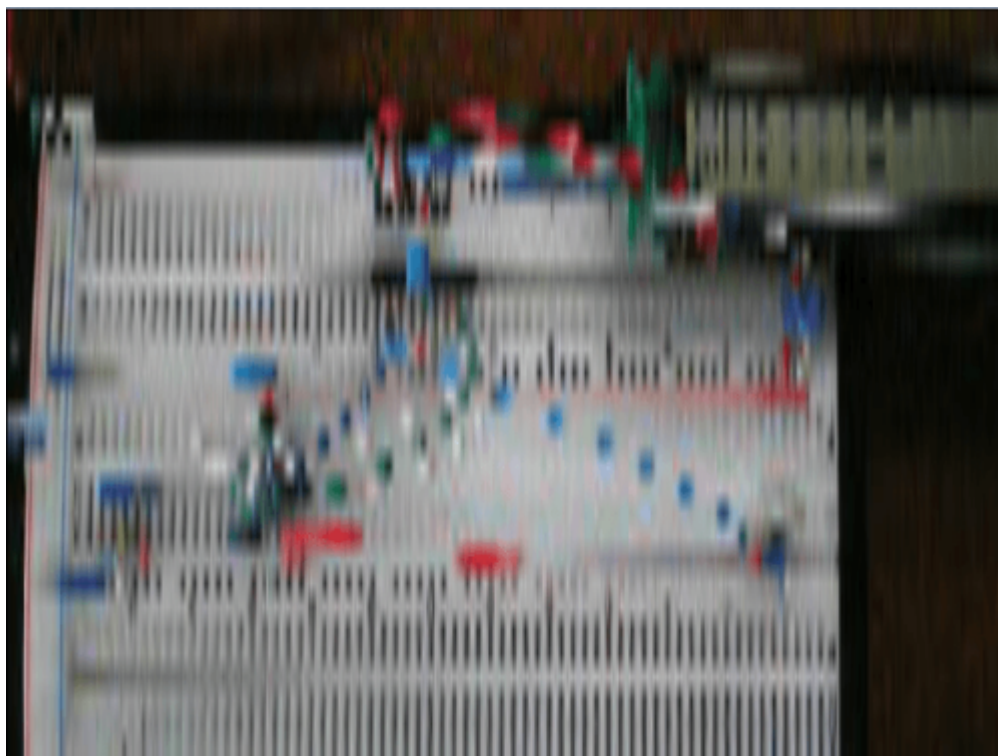
```

    v1 = v1 * 0.9765625; // 分解能 0.9765625uV = 4096000uV ÷
4194304
    v2 = v2 * 4; // 分解能 4mV = 4096mV ÷ 1024
    //表示
    FloatToStr(v1, buf1);
    WordToStr(v2, buf2);
    Lcd_Out(1, 1, "uV:");
    Lcd_Out(2, 1, "mV:");
    Lcd_Out(1, 4, space);
    Lcd_Out(2, 4, space);
    Lcd_Out(1, 4, buf1);
    Lcd_Out(2, 4, &buf2[1]);
    //
    Delay_ms(500);
}

}
//*****
*
//MCP3551
sbit CS at RB1_bit;
sbit SDO at RB2_bit;
sbit SCK at RB3_bit;
sbit CS_Direction at TRISB1_bit;
sbit SDO_Direction at TRISB2_bit;
sbit SCK_Direction at TRISB3_bit;
//
void mcp3551_init()
{
    CS_Direction = OUTPUT_MODE;
    SDO_Direction = INPUT_MODE;
    SCK_Direction = OUTPUT_MODE;
    //
    CS = 1;
    SCK = 1;
}
//*****
*
long mcp3551_read()
{
    long dat;
    short cnt;
    //
    dat = 0;
    CS = 0;
    while (SDO == 1)
    ;
    for (cnt = 0; cnt < 24; cnt++) {
        SCK = 0;
        SCK = 1;
        if (SDO == 1) {
```

```
        dat |= 0x01;
    }
    dat <= 1;
}
CS = 1;
return (dat & 0b001111111111111111111111);
}
//*****
*
```

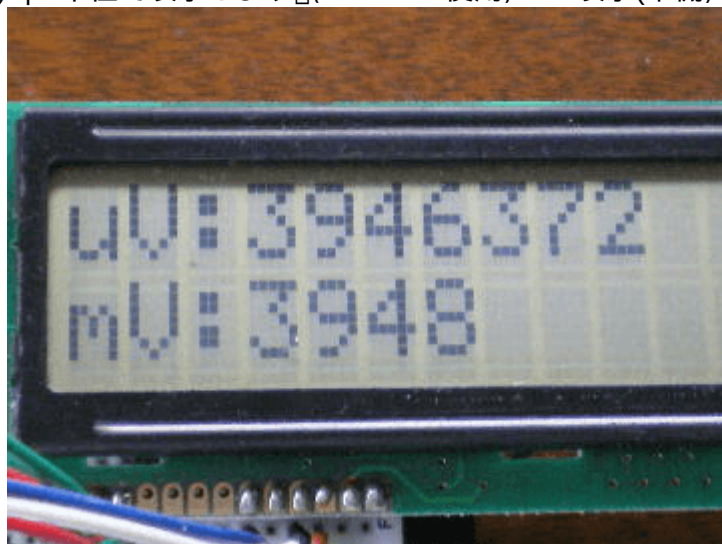
動作確認



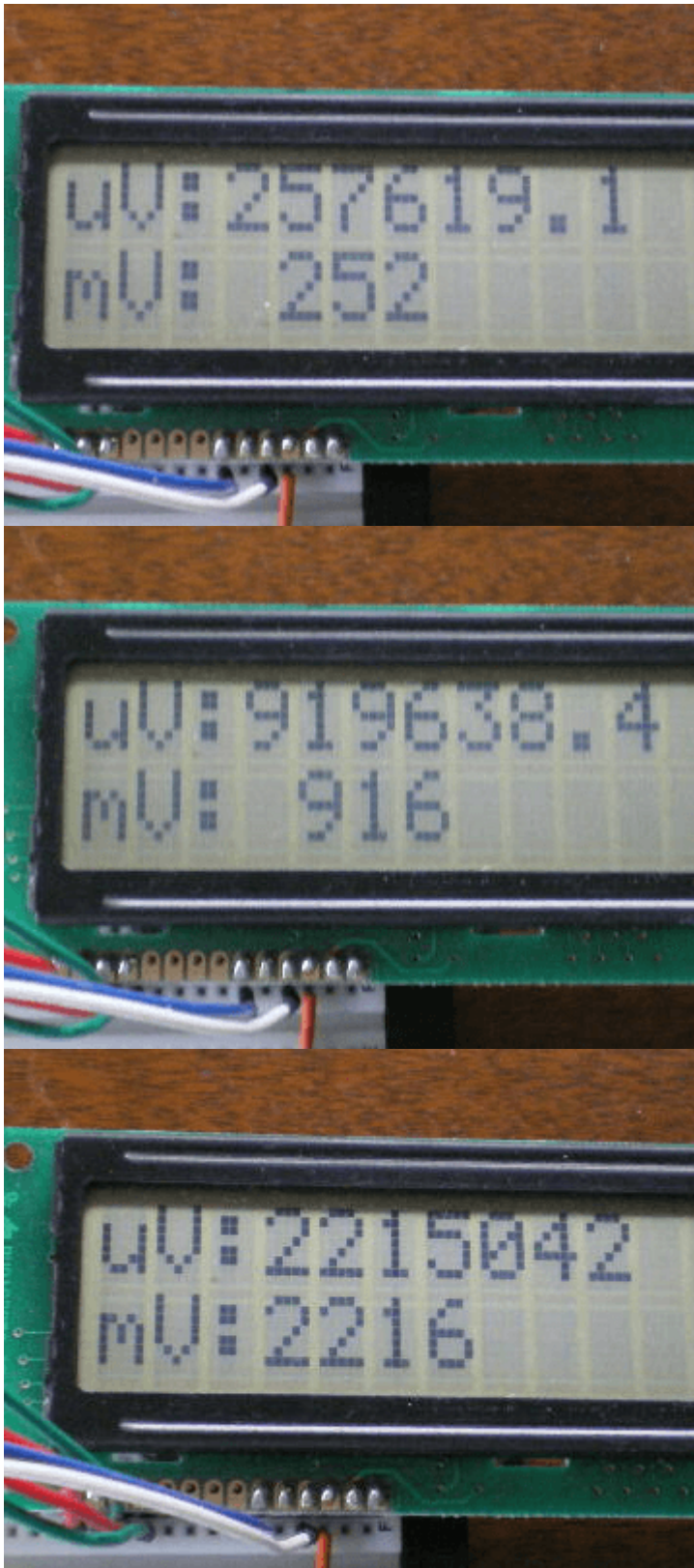
保存されておりました)

(編者注：この画像しか保

LCD表示(上側)= μ V単位で表示します□(MCP3551使用) LCD表示(下側)=mV単位で表示します□(PIC内蔵モ



ジュール使用)



From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:147&rev=1588227308>

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



