

# ワットメータ(WattMeater)

## 概要

いろいろな家電製品の消費電力を測り、電気代の節約の契機にしたいと考え、ワットメータを作成してみました。

## 動作原理

消費電力を測るにはAC100Vのラインに流れる電流を測ることにより電力換算することが出来ます。  
電力(W)=商用電圧(100V)×電流(I) 正確な電力を測るには、本来、次のような条件が必要となります。

- 電流の精度を高めるにはAC100Vのラインの間に入れる抵抗器の高精度のものが必要
- 商用電圧の測定が必要(商用電圧は、必ずしも100V丁度とは限らない)

今回は、抵抗の両端に発生する電圧が消費電力に応じて比例することに着目し、既に消費電力の分かっている製品(例えば裸電球)を測定し、1Wあたりの校正電圧を得ることにより簡易的に消費電力を測定しました。

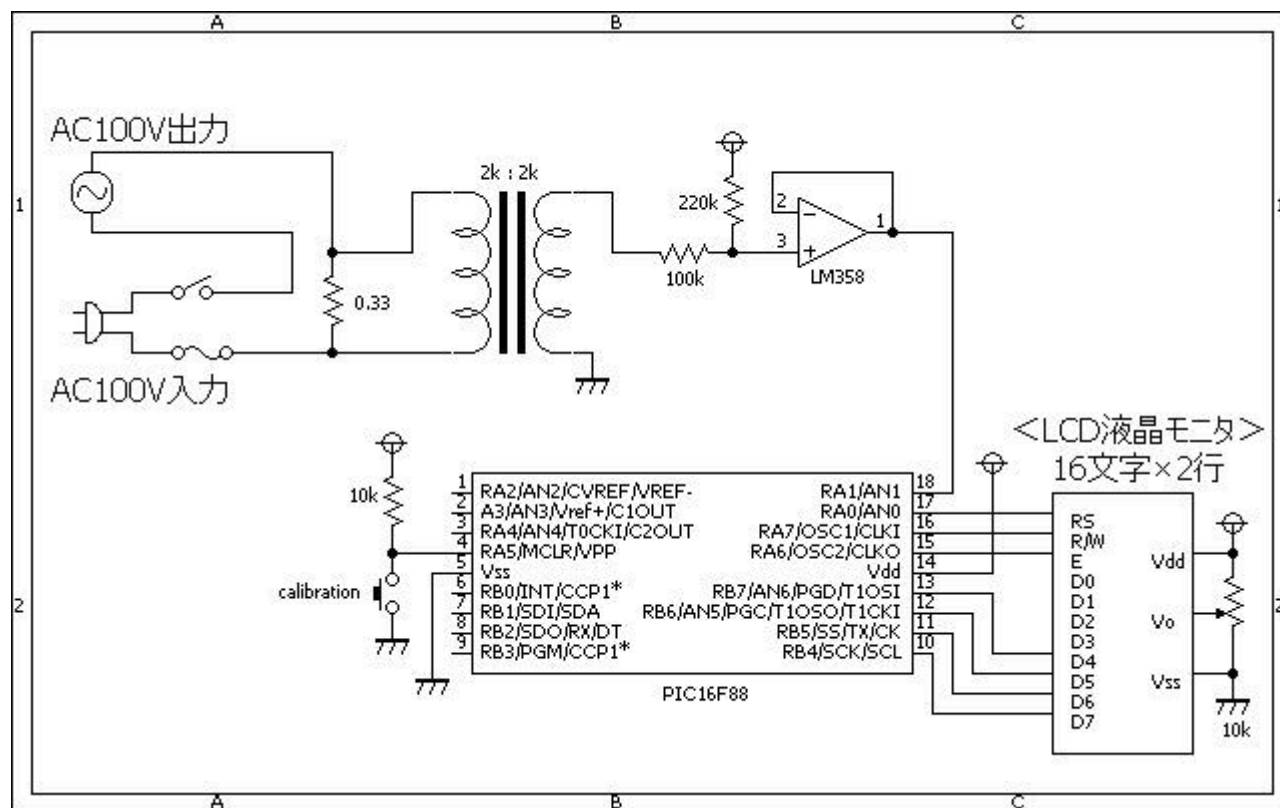
### <測定>

- AC100Vのラインに微少の抵抗(今回は手持ちの0.33Ωのセメント抵抗を使用)を入れます。
- 消費電力に応じて抵抗の両端に電圧が発生します。
- この電圧をトランスを通して、アイソレーションします。
- アイソレーションした電圧をオペアンプでインピーダンス変換します。
- インピーダンス変換した電圧をPICのA/D変換で取り込みます。
- 取り込んだ電圧(V1)を校正電圧(V2)で割ることにより消費電力(W1)が得られます。
- $V1 \div V2 \times W1$ をLCD液晶モニタに表示させます。

### <校正>

- 裸電球(54W)を接続します。  
たまたま手持ちが54Wでしたので、これをcalibrationのために用いました。  
プログラム上で変更が可能です。もしも100Wを消費する何か(裸電球でも抵抗でも結構です)があれば  
プログラム上の54の数値を100に変更してください。
- 校正用のスイッチ(calibration)が押されると1Wあたりの換算値を設定します。  
校正電圧=A/D変換した電圧÷54W
- 校正電圧は、PIC内のEEPROMに記憶されますので、毎回calibrationする必要はありません。

## 回路図



## ソースコード

### WattMeter2.c

```
//*****
*
/*
『ワットメータ』
*/
//*****
*

unsigned int measurement()
{
    unsigned int ad, max, min, cnt;
    //
    ad = 0;
    max = 0;
    min = 1024;
    for (cnt = 0; cnt < 1000; cnt++) {
        ad = Adc_Read(1);
        max = ad > max ? ad : max;
        min = ad < min ? ad : min;
    }
    return (max - min);
}
```

```
//*****
*

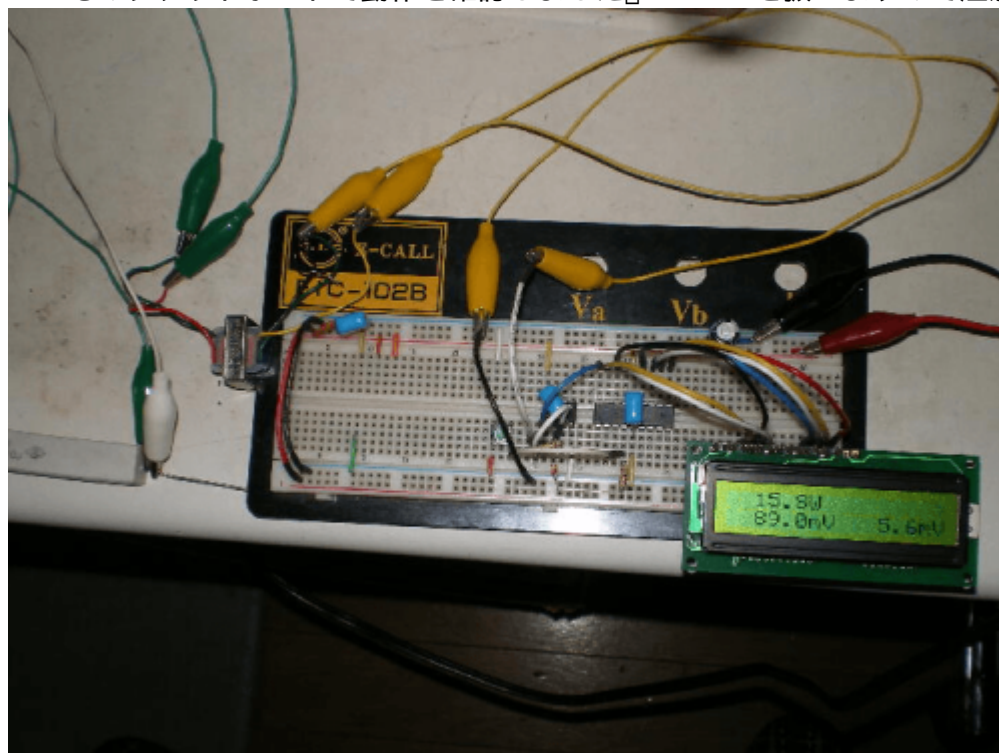
void main()
{
    static unsigned char buf[8], cnt, tmp1, tmp2;
    static unsigned int ad, wattOffset;
    //
    OSCCON = 0b01110000; // クロックは8Mhz
    CMCON = 0b00000111; // コンパレータは使用しない。
    // 変換を使用する。
    ANSEL = 0b00000010;
    // ポートを初期化する。
    TRISA = 0b00111010;
    TRISB = 0b00001111;
    // を初期化する。
    Lcd_Custom_Config(&PORTB, 4, 5, 6, 7, &PORTA, 0, 7, 6);
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);
    Lcd_Custom_Cmd(LCD_CLEAR);
    Lcd_Custom_Out(1, 1, "WattMeter2 V1.0");
    Delay_ms(500);
    Lcd_Custom_Cmd(LCD_CLEAR);
    // 保存されたデータを読み込む
    tmp1 = Eeprom_Read(0);
    Delay_ms(20);
    tmp2 = Eeprom_Read(1);
    Delay_ms(20);
    wattOffset = (tmp2 << 8) | tmp1;
    wattOffset = (wattOffset > 10000) ? 10 : wattOffset;
    //
    while (1) {
        ad = 0;
        for (cnt = 0; cnt < 10; cnt++) {
            ad += measurement();
        }
        ad = ad * 5;
        //
        WordToStr(ad, buf);
        buf[6] = 0x00;
        buf[5] = buf[4];
        buf[4] = '.';
        Lcd_Custom_Out(2, 1, buf);
        Lcd_Custom_Out(2, 7, "mV");
        //
        WordToStr((ad * 10) / wattOffset, buf);
        buf[6] = 0x00;
        buf[5] = buf[4];
        buf[4] = '.';
        Lcd_Custom_Out(1, 1, buf);
        Lcd_Custom_Out(1, 7, "W");
        //
    }
}
```

```
WordToStr(wattOffset, buf);
buf[6] = 0x00;
buf[5] = buf[4];
buf[4] = '.';
Lcd_Custom_Out(2, 9, buf);
Lcd_Custom_Out(2, 15, "mV");
//
if (PORTA.F5 == 0) {
    wattOffset = ad / 54;    // 校正用です。
    // データを保存する。
    Eeprom_Write(0, wattOffset & 0xFF);
    Delay_ms(20);
    Eeprom_Write(1, (wattOffset >> 8) & 0xFF);
    Delay_ms(20);
}
}

//*****
*
```

## 動作確認

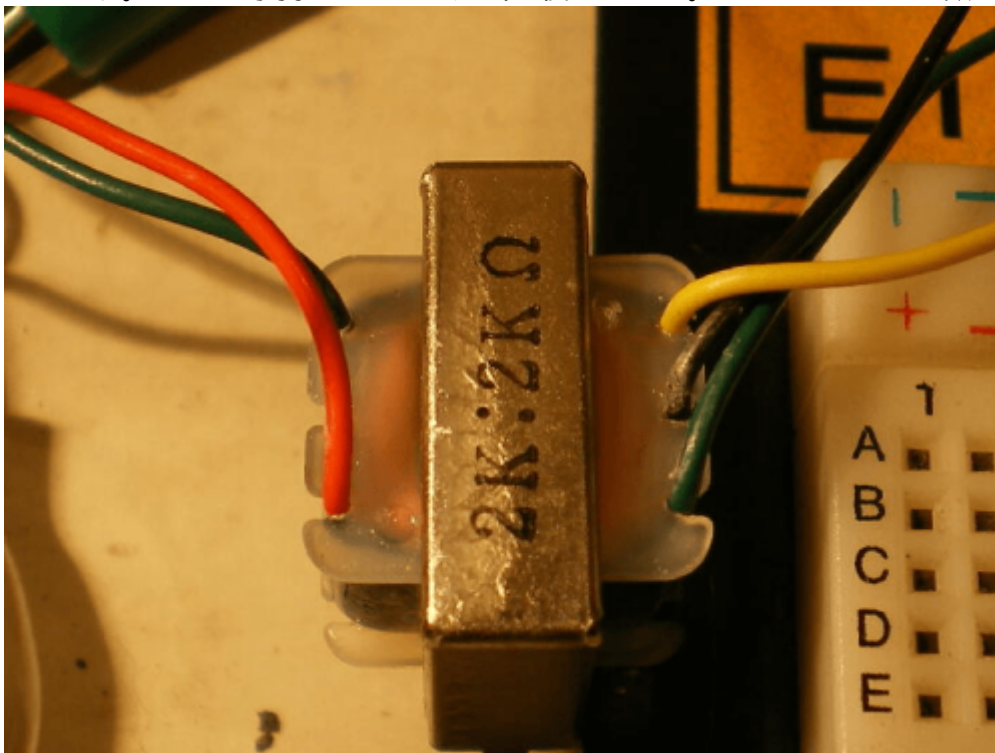
いつものブレッドボードで動作を確認しました。AC100Vを扱いますので注意してください。



電圧検出用のセメント抵抗です。手持ちのものを適当に使いました。0.1Ω~1.0Ω位のものをお使いください。<calibration>スイッチを押すことにより、プログラムで自動的に校正します。



アイソレーション用のトランスです。これも手持ちのものを適当に使いました。1:1のトランスで数kΩのものをお使いください。

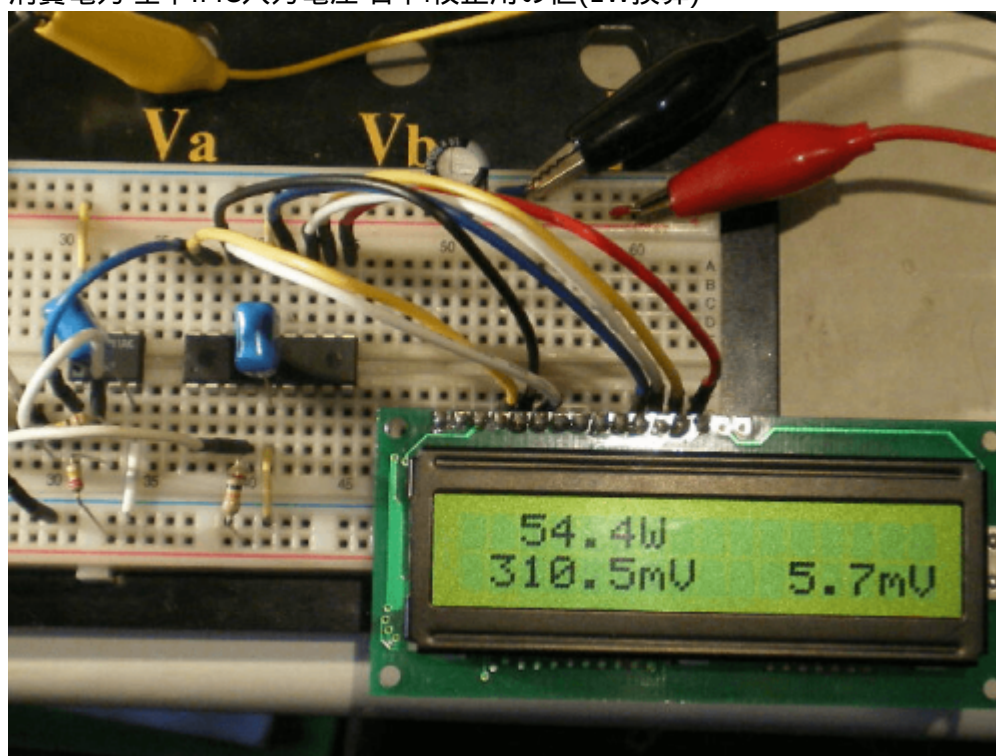


54Wの裸電球で校正しま

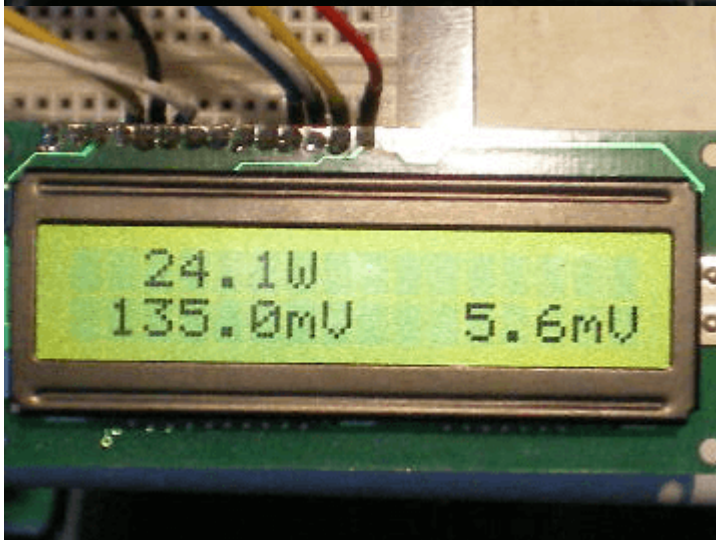
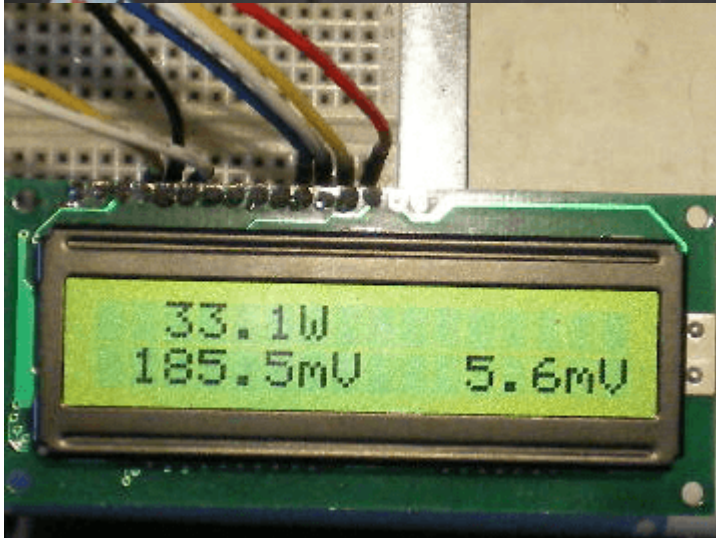


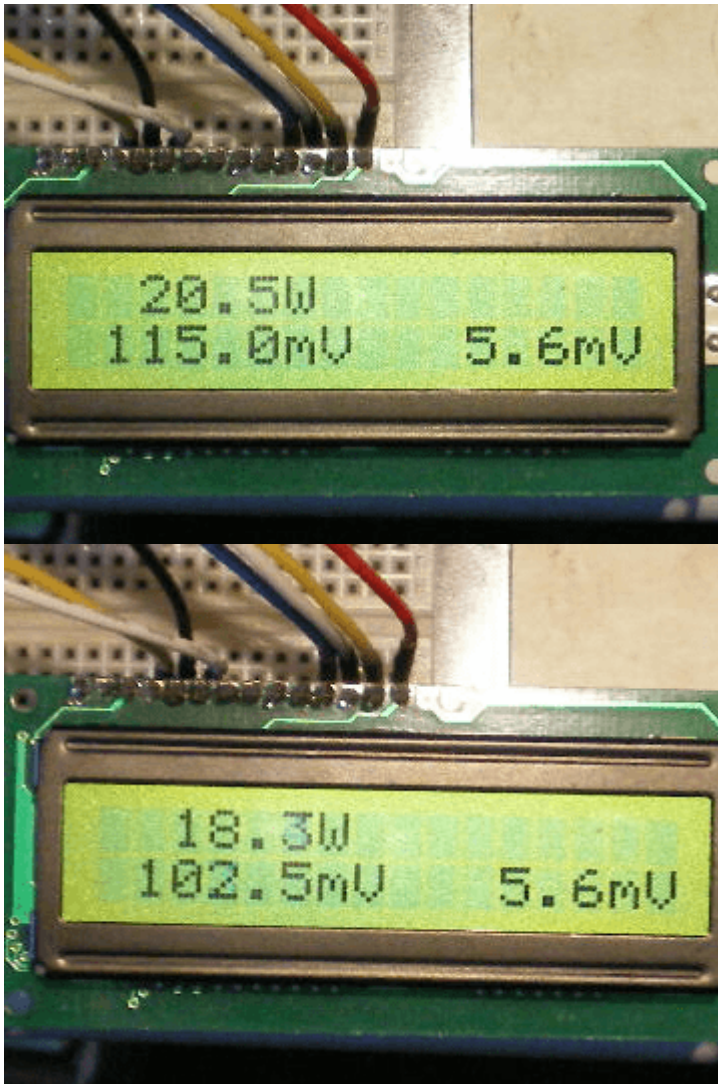
す。  
消費電力 左下:PIC入力電圧 右下:校正用の値(1W換算)

表示の内容は、左上:



手持ちの半田ごて(18W)を測定してみました。電源投入直後は、セラミックヒータの抵抗値が低いので定格値よりも大きな値の消費電力となりますが、温まってくると定格値に近づいてきます。





From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:72&rev=1588153398>

Last update: 2025/10/17 14:28

