

# タコメータ(Tachometer)

## 概要

車のエンジンの回転数を、車に特別な細工をせずに測定し表示してみました。

## 動作原理

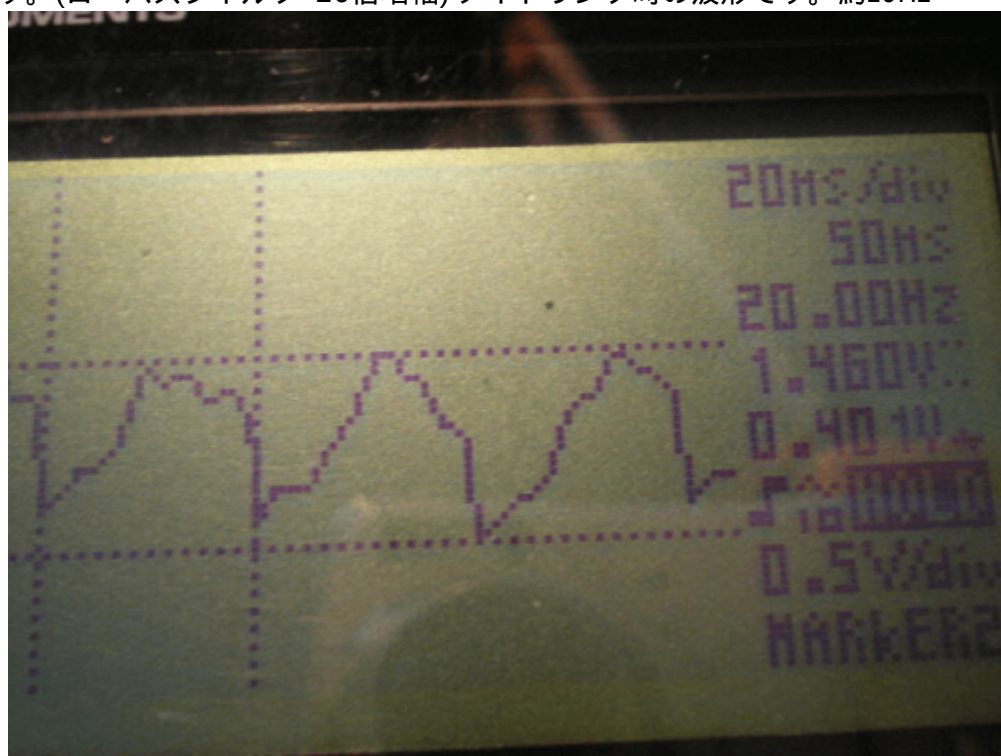
本ホームページをご覧の方からご指摘(感謝)がありましたので修正させていただきます。 <誤> シガーライターソケット(cigarette lighter socket)から、電源を取り、その電源に含まれるスパークノイズより回転数を求めます。 <正> シガーライターソケット(cigarette lighter socket)から、電源を取り、その電源に含まれるオルタネータの微小信号波形(三相交流)より、エンジンの回転数を求めます。

【オルタネータ】 オルタネータは、単相交流に対し多くのすぐれた特性をもつ三相交流を発電するようになっています。 エンジン回転数は、車両の走行状況によって刻々と変化するのでオルタネータの回転数も変化します。

オルタネータのしくみについての詳細は、デンソー東京のホームページをご覧ください。

<http://www.denso-tokyo.com/idea/alternator/>

ハンドヘルドのオシロスコープでシガーライターの電圧の波形を観測してみました。 正確には、回路図のLM386の出力波形です。(ローパスフィルタ+20倍増幅) アイドリング時の波形です。 約20Hz



で1.46Vp-pあります。

### 【プログラム上の仕組み】

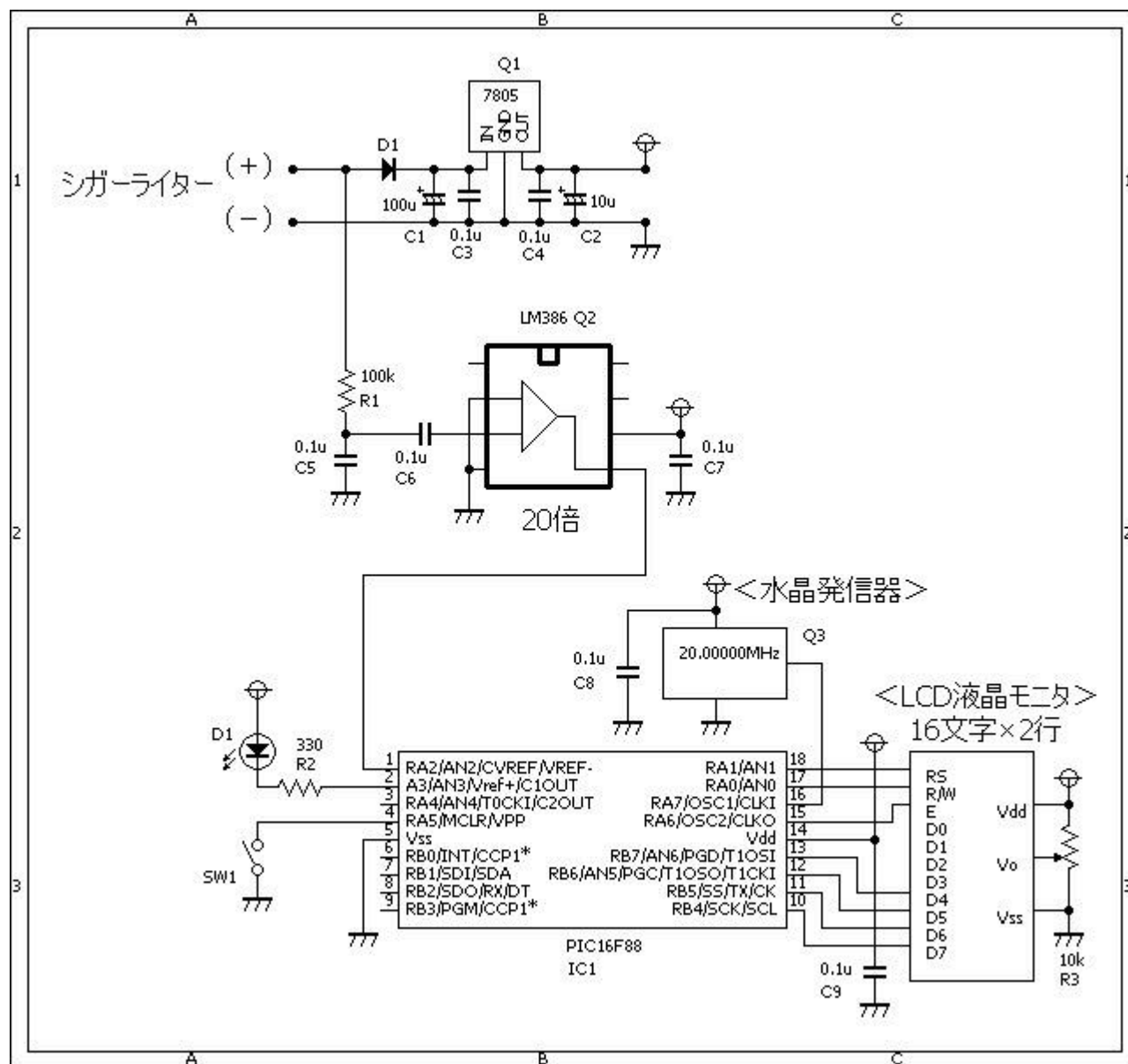
#### 基準電圧を求める

- 1msec周期で電圧データ(A/D変換)を取り込む。
- 1秒間分(1000個の電圧データ)を累積し、平均電圧(基準電圧)を求める。

## 回転数を求める

- 1msec周期で取り込んだ電圧データが、基準電圧よりも<高い、低い、等しい>かを判断する。
- この時にはノイズ誤差を減らすために2回連続してデータを取り込み判断する。
- 判断した結果、<高い、低い>場合には、回転数をカウントアップする。
- 1秒間(1000回)これらの処理を繰り返す。
- この回転数を更に換算し、エンジン回転数(RPM)とする。
  - 換算式 (回転数 ÷ 3相交流) × 60秒

## 回路図



## ソースコード

[TachoMeter.c](#)

```
// *****
```

```

*
/*
< タコメータ >
■ 機能概要
車のエンジンの回転数を測定し、表示する。
シガーライターソケット[cigarette lighter socket]から、電源を取り
その電源に含まれるスパークノイズより回転数を求める。
■ 表示内容
回転数
回転数 (バー表示)
平均電圧
最大電圧
最小電圧
□■□□□□□□の設定
    □LVP_OFF
□□□MCLR_OFF
□□□WDT_OFF
□□□EXTCLK
*/
//*****
*

#define          LED          PORTA.F3
#define          SW          PORTA.F5

//*****
*

static unsigned    int          aveAd, maxAd, maxAdTmp, minAd, minAdTmp;
static unsigned    long        tmpAd;
static unsigned    int          cntAd;

static unsigned    int          tachoCnt, tachoCntTmp;

static unsigned    char        highCnt;
static unsigned    char        lowCnt;

static unsigned    char        secFlag;

void interrupt()
{
    static unsigned    int          ad;
    //□□□□□□の割り込み処理
    if (PIR1.CCP1IF == 1) {
        PIR1.CCP1IF = 0;
        //アナログデータを□□□変換で取り込み累積する。
        ad = Adc_Read(2);
        tmpAd += ad;
        cntAd++;
        // 1 秒毎に回転数、平均電圧、最大電圧、最小電圧を求める。
        if (cntAd == 1000) {
            //平均電圧

```

```

    aveAd = tmpAd / cntAd;
    tmpAd = 0;
    cntAd = 0;
    //最大電圧
    maxAd = maxAdTmp;
    maxAdTmp = 0;
    //最小電圧
    minAd = minAdTmp;
    minAdTmp = 1024;
    //回転数
    tachoCnt = tachoCntTmp;
    tachoCntTmp = 0;
    secFlag = 1;
    //□□□の□□□□□□
    LED = ~LED;
}
//今回取り込んだ電圧が平均電圧より高いかを判断する。
if (ad >= (aveAd + 10)) {
    highCnt++;
    lowCnt = 0;
    if (highCnt == 2)
        tachoCntTmp++;
}
//今回取り込んだ電圧が平均電圧より低いかを判断する。
if (ad <= (aveAd - 10)) {
    lowCnt++;
    highCnt = 0;
    if (lowCnt == 2)
        tachoCntTmp++;
}
//今回取り込んだ電圧が平均電圧とほぼ同等かを判断する。
if ((ad < (aveAd + 10)) && (ad > (aveAd - 10))) {
    highCnt = 0;
    lowCnt = 0;
}
//最大電圧を求める。
if (ad > maxAdTmp)
    maxAdTmp = ad;
//最小電圧を求める。
if (ad < minAdTmp)
    minAdTmp = ad;
}

//*****
*

void main()
{
    static    unsigned    char    buf[20], cnt;
    static    unsigned    int     ad;

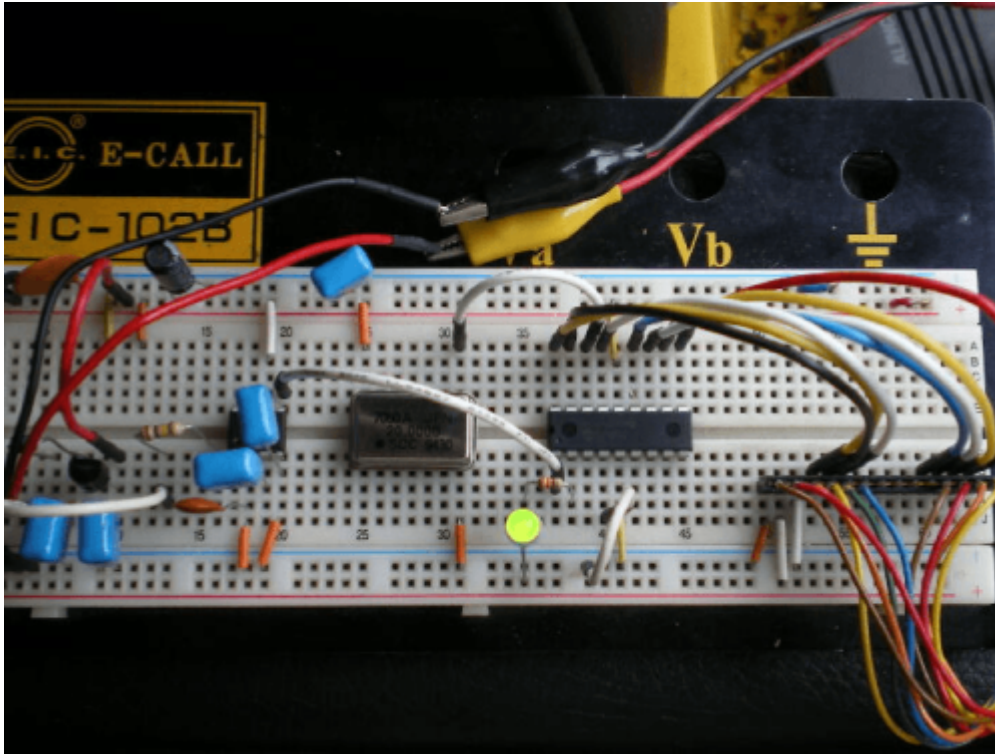
```

```
// アナログの設定
ANSEL = 0b00000100; // 使用する。
// ポートの設定
TRISA = 0b10110100;
TRISB = 0b00001111;
OPTION_REG.F7 = 0; // PORTBをプルアップする。
// CCPの設定
PIE1.CCP1IE = 1;
PIR1.CCP1IF = 0;
CCP1CON = 0b00001011;
CCPR1L = 0x88; // 0.001sec...(1÷20000000)*4*5000
CCPR1H = 0x13;
// TIMER1の設定
PIE1.TMR1IE = 0;
PIR1.TMR1IF = 0;
TMR1L = 0;
TMR1H = 0;
T1CON.T1CKPS0 = 0;
T1CON.T1CKPS1 = 0;
T1CON.TMR1ON = 1;
// 変数の初期化
TMR1L = 0;
TMR1H = 0;
// □□□□液晶モニタ)の初期化
Lcd_Custom_Config(&PORTB,4,5,6,7,&PORTA,1,0,6);
Lcd_Custom_Cmd(LCD_CURSOR_OFF);
Lcd_Custom_Out(1, 1, "TachoMeter V1");
Delay_ms(1000);
Lcd_Custom_Cmd(LCD_CLEAR);
//
LED = 1; // LED off!
aveAd = 0;
maxAdTmp = 0;
minAdTmp = 1024;
tmpAd = 0;
cntAd = 0;
highCnt = 0;
lowCnt = 0;
secFlag = 0;
tachoCnt = 0;
tachoCntTmp = 0;
// 割り込み(全体)の設定
INTCON.PEIE = 1;
INTCON.GIE = 1;
//
while (aveAd == 0)
;
//
while(1) {
    // 1秒周期でデータを表示する。
    if (secFlag == 1) { //回転数を表示する。
```

```
secFlag = 0;
WordToStr(tachoCnt * 20, buf);
Lcd_Custom_Out(1, 1, buf);
Lcd_Custom_Out(1, 6, "rpm");
//
if (SW == 0) {    //最大、最小、平均を表示する。
    WordToStr(maxAd * 5, buf);
    Lcd_Custom_Out(2, 1, buf);
    Lcd_Custom_Out(2, 6, "mV");
    //
    WordToStr(minAd * 5, buf);
    Lcd_Custom_Out(2, 9, buf);
    Lcd_Custom_Out(2, 14, "mV");
    //
    WordToStr(aveAd * 5, buf);
    Lcd_Custom_Out(1, 9, buf);
    Lcd_Custom_Out(1, 14, "mV");
} else {    //回転数をバー表示する。
    for (cnt = 0; cnt < 16; cnt++) {
        if (cnt < ((tachoCnt * 20) / 200))
            Lcd_Custom_Chr(2, cnt + 1, 0xFF);
        else
            Lcd_Custom_Chr(2, cnt + 1, ' ');
    }
    //
    WordToStr((maxAd - minAd) * 5, buf);
    Lcd_Custom_Out(1, 9, buf);
    Lcd_Custom_Out(1, 14, "mV");
}
}
}
}

//*****
*
```

## 動作確認



アイドリングの時の回転数です。



その時の各種データの表示です。 左上:回転数 右上:平均電圧 左下:最大電圧 右下:最小電圧



アクセルを踏んで回転数を上げてみました。



如何ですか?これでタコメータの無い車でも回転数が分かりますね 😊

今回は、自家用車(ダイハツmove)で確認しました。

#### 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://www.deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:82>

Last update: **2020/05/01 18:01**

