

簡易湿度計

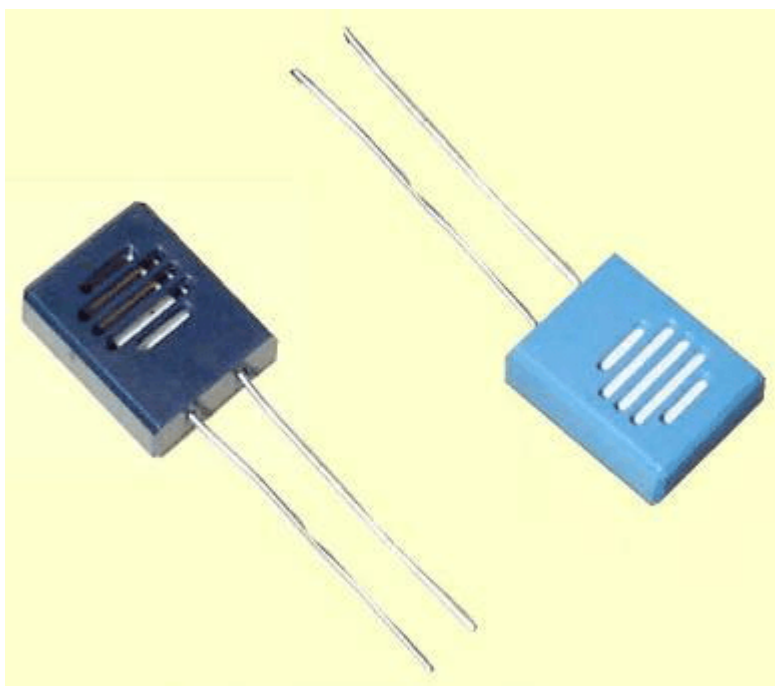
概要

梅雨の時期に入り、じめじめした日が続きます。今回は、湿度センサを使用して湿度を測定しLCDに表示させて見ました。行く行くは、これを応用して、家の床下の湿度制御システムを構築してみたいと考えています。(白蟻防御)

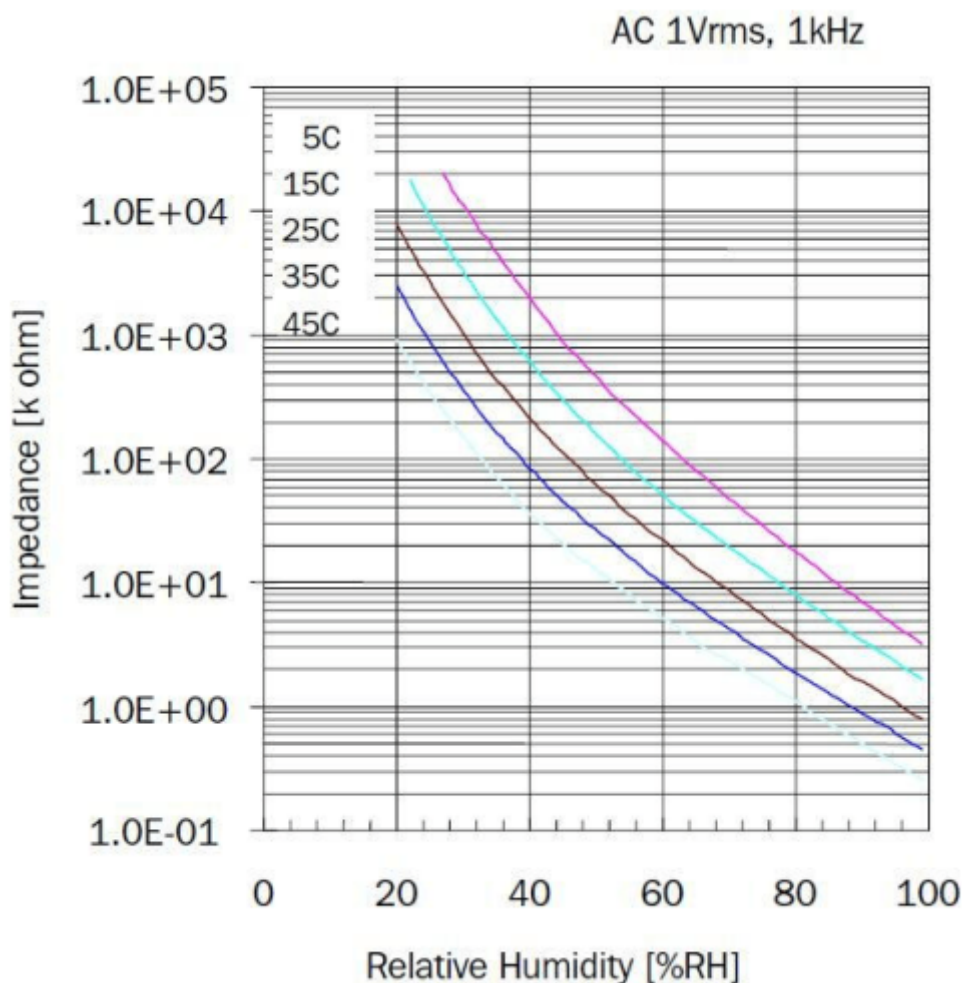
動作原理

湿度センサにはHS-15Pを使用しました。このセンサは、空調機器などの湿度制御システム用として開発されたセラミック湿度センサです。

<特徴> 使用湿度範囲:10%RH~90%RH ◆使用温度範囲:0~60 使用周波数:50Hz~1kHz ◆応答特性:(30⇔90%RH)≤2分 センサは直流駆動できません。交流信号でセンサーをドライブします。出力は湿度に対して指数関数的に出ます。(湿度が上がるとインピーダンスが下がる) 正確に計るには、温度補正が必要となりますが、今回は考慮していません。



<概観>※HS-15Pは、右側です。

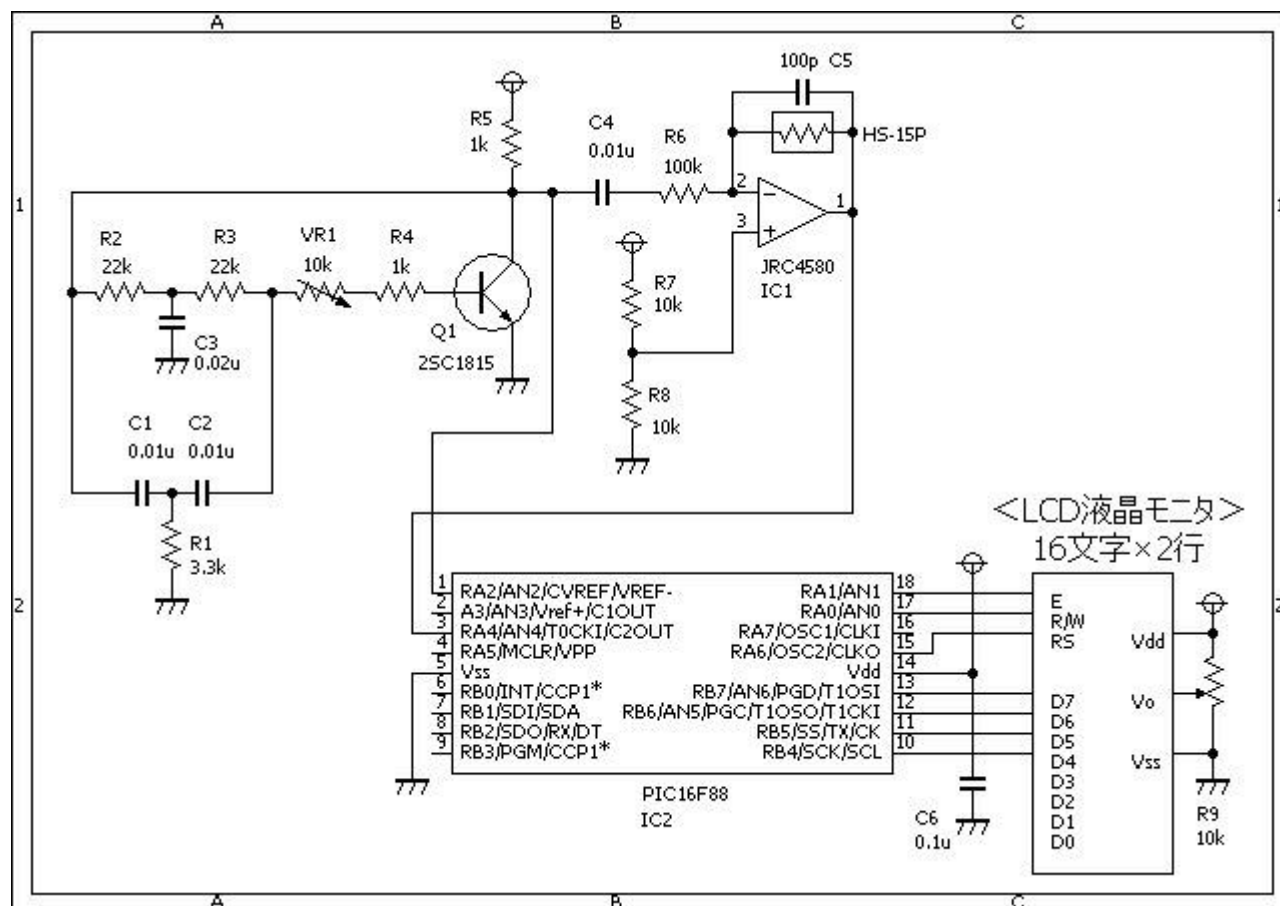


<特性>

<構成> 信号発振部分 簡易な、トランジスタ1石による、ツインT形発振回路で約1kHzの正弦波を発振させます。 信号増幅部分 オペアンプによる反転増幅回路としました□ $V_o = V_i \times (R_6 \div HS-15P \text{の抵抗})$ 信号処理部分 PICで次のような処理を行います。

- 入力信号 V_i の電圧を求める。
 - 出力信号 V_o の電圧を求める。
 - HS-15Pの抵抗 R_x を求める。
 $R_x = (V_o \div V_i) \times 100k$
 - R_x およびHS-15Pの特性値より湿度を求める。
特性値は、次のようにテーブル化しておく。(今回は25 °Cの時の特性値固定としている)
- | | |
|-----|----------|
| 90% | 1.7kΩ |
| 80% | 3.5kΩ |
| 70% | 9.0kΩ |
| 60% | 20.3kΩ |
| 50% | 60.0kΩ |
| 40% | 220.0kΩ |
| 30% | 900.0kΩ |
| 20% | 8000.0kΩ |

回路図



ソースコード

HygroMeter.c

```
//*****
*
*/
<簡易湿度計>
*/
//*****
*

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

```
    return (max - min);
}

//*****
*

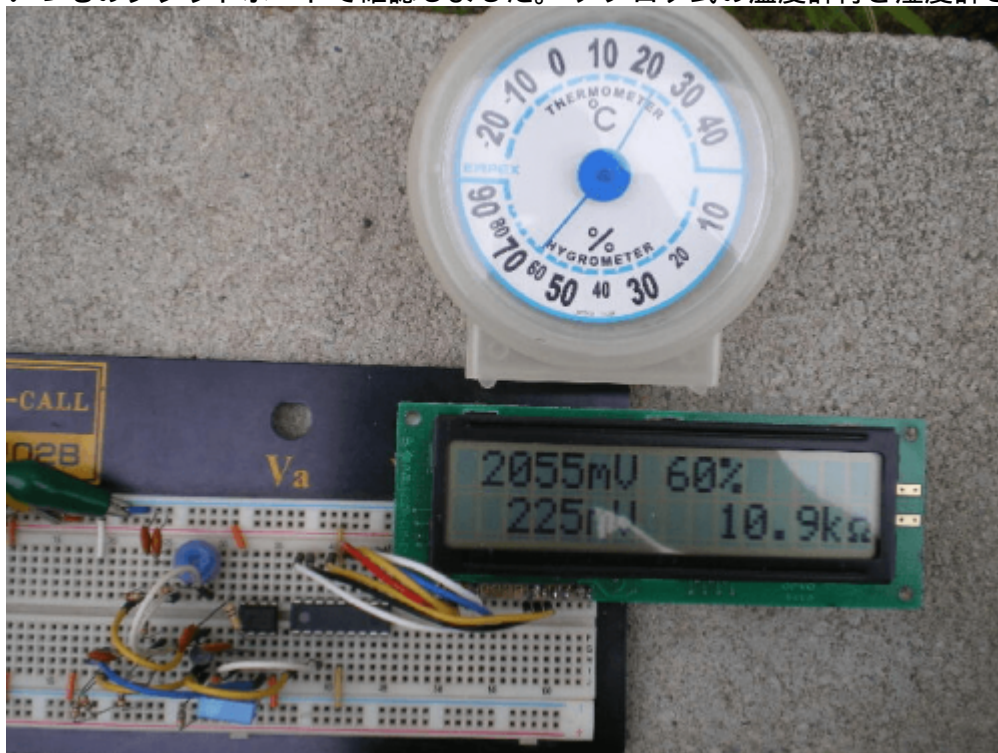
void main()
{
    static unsigned char buf[10], cnt;
    static unsigned int vi, vo;
    static unsigned long tmp;
    //
    OSCCON = 0b01110000; // クロックは8Mhz
    CMCON = 0b00000111; // コンパレータは使用しない。
    // □□□変換を使用する。
    ANSEL = 0b00010100;
    // ポートを初期化する。
    TRISA = 0b10111100;
    TRISB = 0b00000110;
    // □□□を初期化する。
    Lcd_Custom_Config(&PORTB, 7, 6, 5, 4, &PORTA, 6, 0, 1);
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);
    Lcd_Custom_Out(1, 1, "HygroMeter V1");
    Delay_ms(500);
    Lcd_Custom_Cmd(LCD_CLEAR);
    //
    while (1) {
        vi = 0;
        for (cnt = 0; cnt < 5; cnt++) {
            vi += measurement(2);
        }
        //
        vo = 0;
        for (cnt = 0; cnt < 5; cnt++) {
            vo += measurement(4);
        }
        //
        WordToStr(vi, buf);
        Lcd_Custom_Out(1, 1, buf);
        Lcd_Custom_Out(1, 6, "mV");
        //
        WordToStr(vo, buf);
        Lcd_Custom_Out(2, 1, buf);
        Lcd_Custom_Out(2, 6, "mV");
        //
        tmp = vo;
        tmp *= 1000;
        tmp /= vi;
        //
        WordToStr(tmp, buf);
    }
}
```

```
    buf[8] = 0x00;
    buf[7] = 0xF4;
    buf[6] = 'k';
    buf[5] = buf[4];
    buf[4] = '.';
    Lcd_Custom_Out(2, 9, buf);
    //
    if (tmp <= 17) {
        Lcd_Custom_Out(1, 11, "90%");
        continue;
    }
    if (tmp <= 35) {
        Lcd_Custom_Out(1, 11, "80%");
        continue;
    }
    if (tmp <= 90) {
        Lcd_Custom_Out(1, 11, "70%");
        continue;
    }
    if (tmp <= 230) {
        Lcd_Custom_Out(1, 11, "60%");
        continue;
    }
    if (tmp <= 600) {
        Lcd_Custom_Out(1, 11, "50%");
        continue;
    }
    if (tmp <= 2200) {
        Lcd_Custom_Out(1, 11, "40%");
        continue;
    }
    if (tmp <= 9000) {
        Lcd_Custom_Out(1, 11, "30%");
        continue;
    }
    if (tmp <= 80000) {
        Lcd_Custom_Out(1, 11, "20%");
        continue;
    }
    Lcd_Custom_Out(1, 11, "10%");
}
}
/*
    17,          // 90%
    35,          // 80%
    90,          // 70%
    203,         // 60%
    600,         // 50%
    2200,        // 40%
    9000,        // 30%
    80000,       // 20%
```

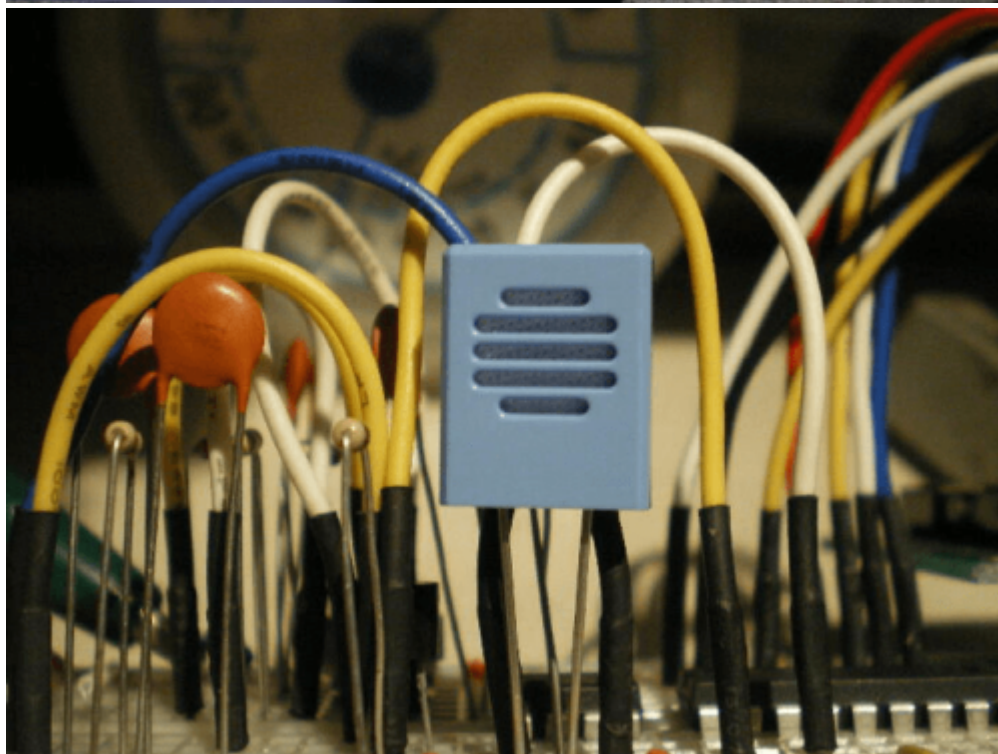
```
* /  
// *****  
*
```

動作確認

いつものブレッドボードで確認しました。アナログ式の温度計付き湿度計と結果を比較しました。



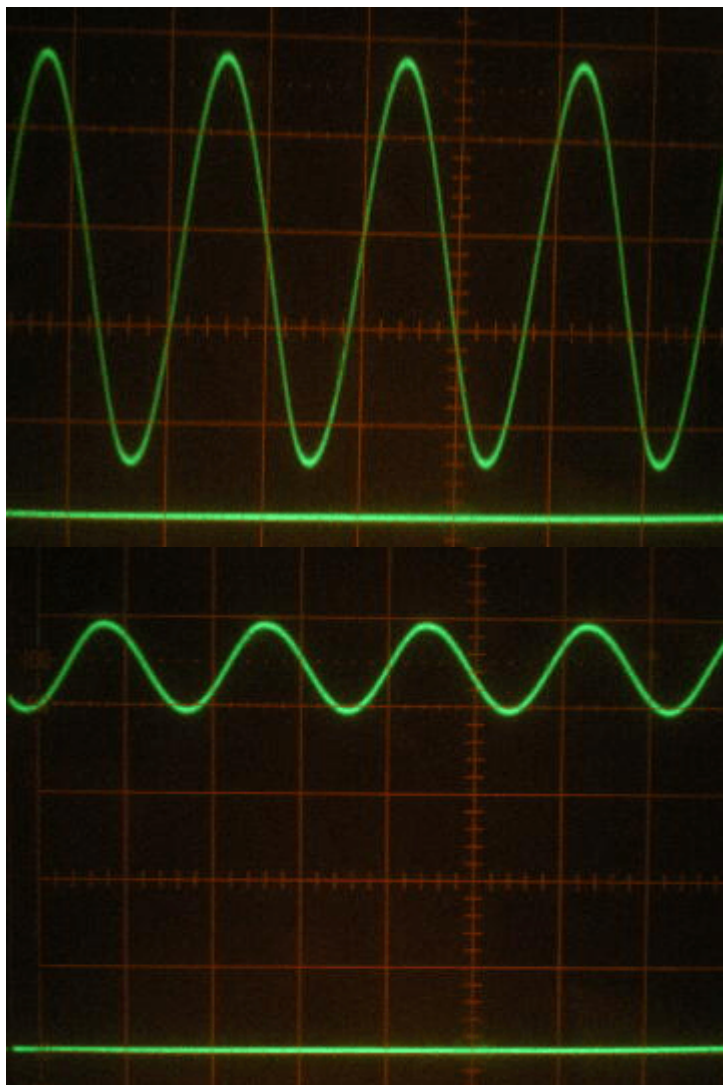
センサー部分です。



- 左側: トランジスタの出力波形(PICのAN2入力信号)約2V-pp(固定電圧)

綺麗な波形になるように10k Ω のボリュームで調整します。

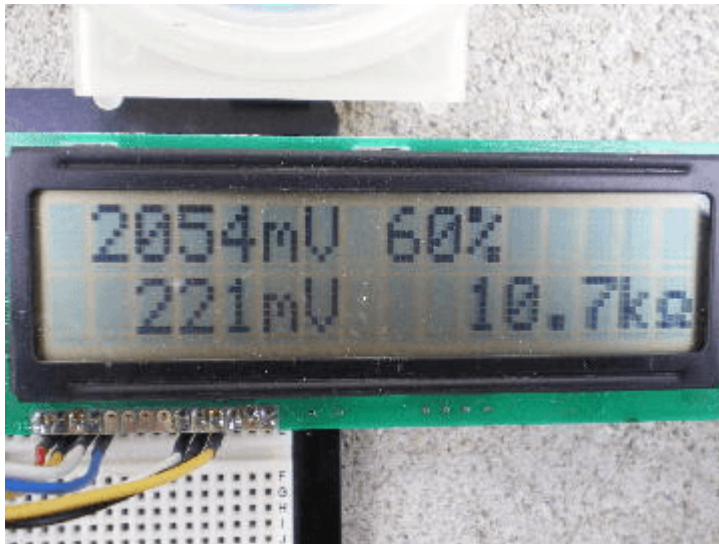
- 右側:オペアンプの出力波形(PICのAN4入力信号)約0.5V-pp(この電圧が湿度によって変化します)



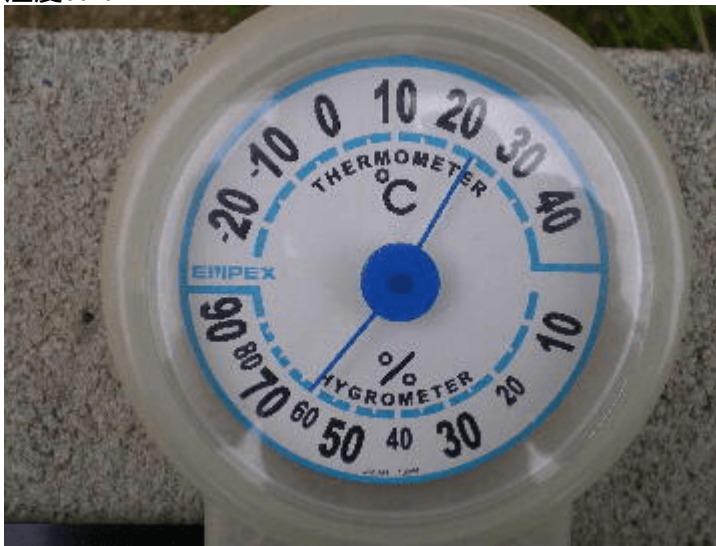
信号の周波数は、約1.1kHzです。回路上のR1,R2,R3,C1,C2,C3を調整して1kHzに下げた方が良いでしょう。



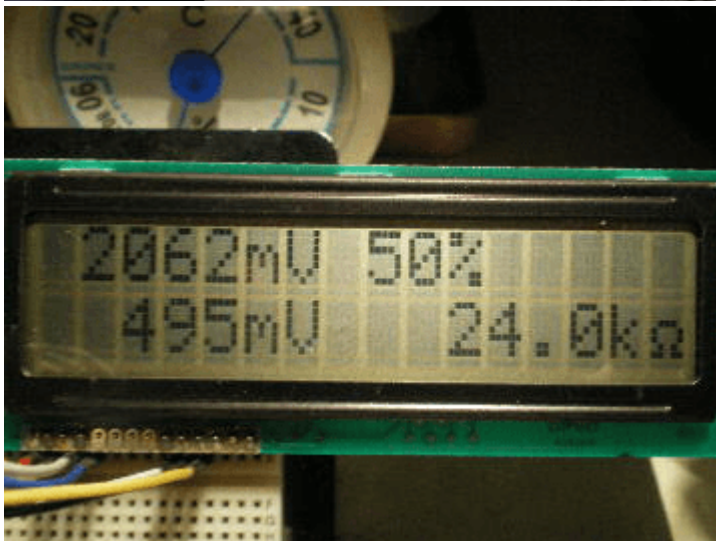
実際に測定してみました。

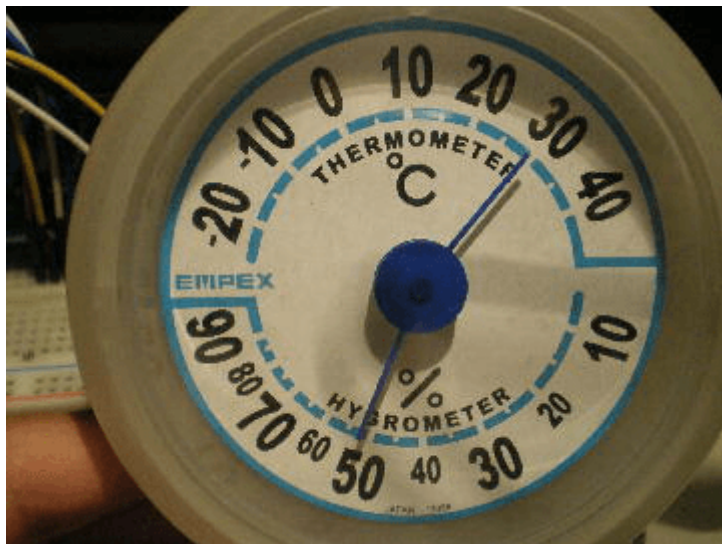


湿度60%



実際に測定してみました。湿度50%





From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:88&rev=1588162986>

Last update: 2025/10/17 14:28

