

# ステレオレベルメータV3(片チャンネル80ドット)

## 概要

以前にも□LED10ドット表示の簡易レベルメータ□LCD16ドット表示のステレオレベルメータ(LCD)等を製作しました。今回は、更に分解能を80ドットまで上げてみました。

<仕様>

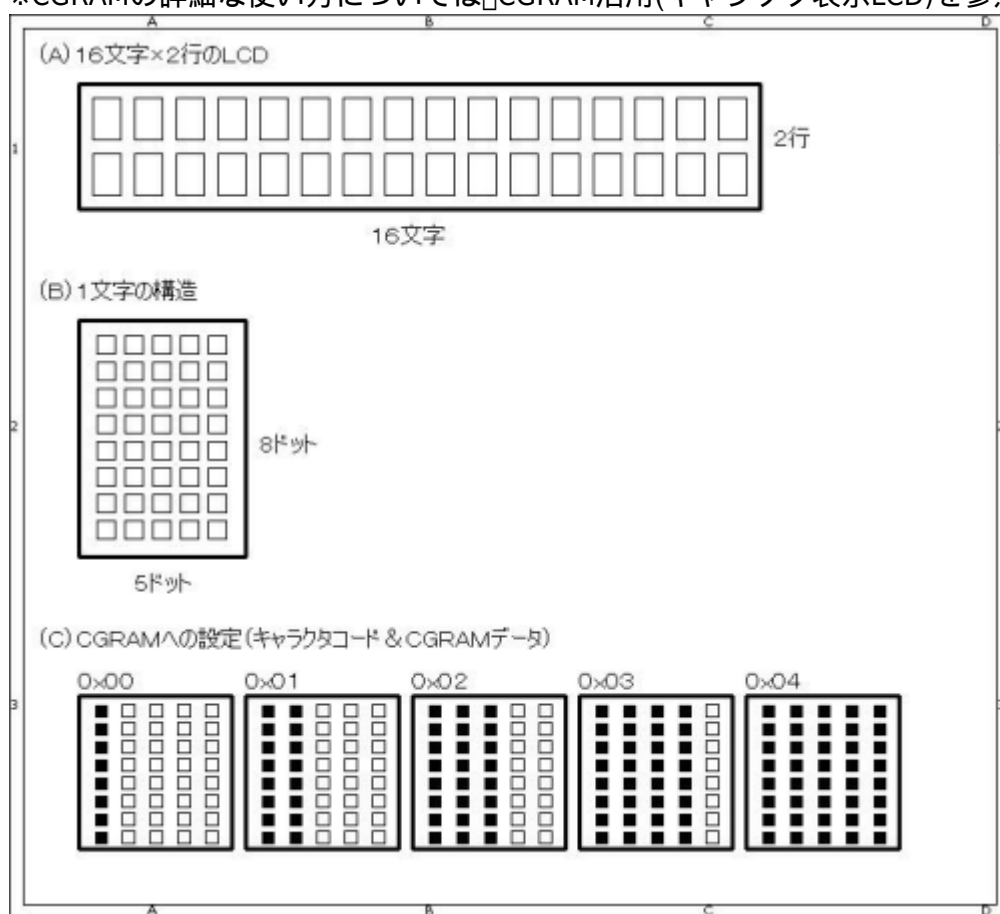
- ステレオ対応(Lチャンネル□Rチャンネル)
- 分解能は、80ドット
- 表示周期は、約0.1秒(1秒間に10回)
- 周波数帯域は、約10Hz~100kHz

## 動作原理

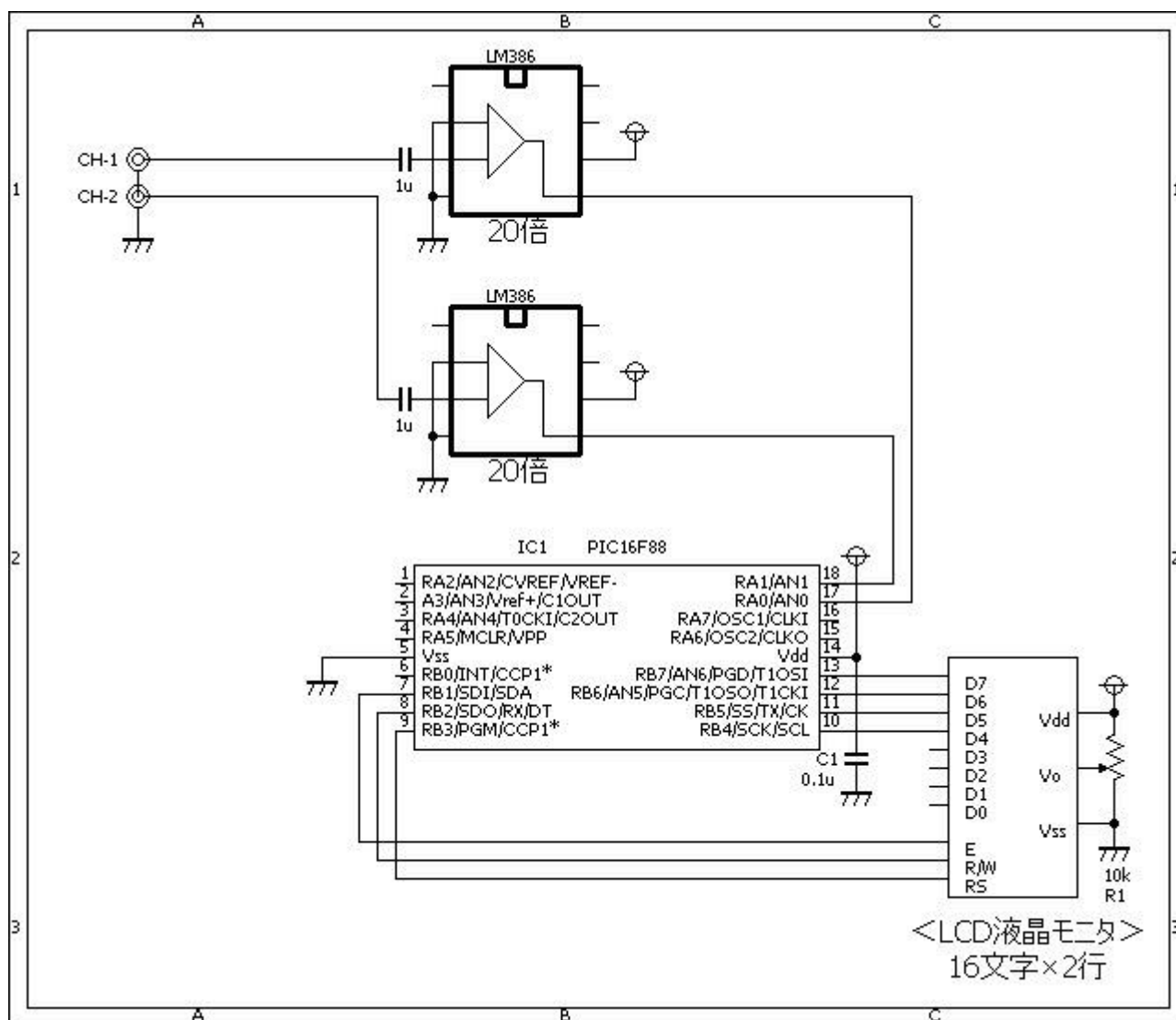
通常、よく使われLCDは、16文字×2行、1文字の構造は、5ドット×8ドットなので、工夫すれば、片チャンネル80ドットのレベルメータが出来ます。

その工夫とは□CGRAMを使用し、1文字で、5段階を表示させるようにすることにより、5段階×16文字で80ドットを表示させることです。

※CGRAMの詳細な使い方については□CGRAM活用(キャラクタ表示LCD)を参照してください。



## 回路図



## ソースコード

### LevelMeterV3.c

```
//*****
*
/*
『ステレオレベルメータ□□□□ドット』
*/
//*****
*

#define LED PORTB.F0
#define ON 1
#define OFF 0
```

```
//*****  
*  
const char character1[] = {16,16,16,16,16,16,16,16};  
const char character2[] = {24,24,24,24,24,24,24,24};  
const char character3[] = {28,28,28,28,28,28,28,28};  
const char character4[] = {30,30,30,30,30,30,30,30};  
const char character5[] = {31,31,31,31,31,31,31,31};  
  
void RegistCustomChar()  
{  
    char i;  
    //  
    LCD_Cmd(64);  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character1[i]);  
    }  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character2[i]);  
    }  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character3[i]);  
    }  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character4[i]);  
    }  
    for (i = 0; i<=7; i++) {  
        LCD_Chr_Cp(character5[i]);  
    }  
    LCD_Cmd(LCD_RETURN_HOME);  
}  
  
//*****  
*  
void BarDisp(char index, unsigned int dat)  
{  
    short    i, j, k, cnt;  
    //  
    i = (dat * 10) / 128;  
    j = i / 5;  
    k = i - (j * 5);  
    //  
    if (index == 1)  
        Lcd_Cmd(LCD_FIRST_ROW);  
    else  
        Lcd_Cmd(LCD_SECOND_ROW);  
    //  
    for (cnt = 1; cnt <= j; cnt++) {  
        Lcd_Chr_Cp(4);  
    }  
}
```

```
Lcd_Chr_Cp(k);
for (cnt++; cnt <= 16; cnt++) {
    Lcd_Chr_Cp(' ');
}
}

//*****
*

unsigned int      adR, adL;

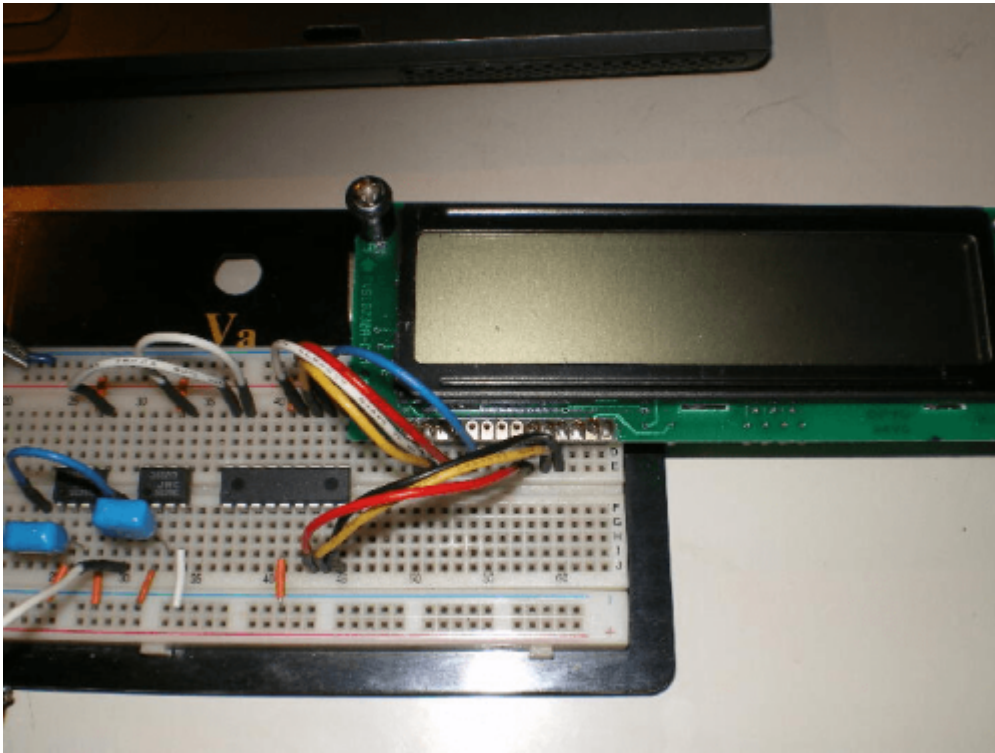
void measurement()
{
    unsigned int      maxR, minR, maxL, minL;
    unsigned int      cnt;
    //
    maxR = 0;
    minR = 1024;
    maxL = 0;
    minL = 1024;
    for (cnt = 0; cnt < 300; cnt++) {
        adR = Adc_Read(0);
        adL = Adc_Read(1);
        maxR = maxR < adR ? adR : maxR;
        minR = minR > adR ? adR : minR;
        maxL = maxL < adL ? adL : maxL;
        minL = minL > adL ? adL : minL;
    }
    adR = maxR - minR;
    adL = maxL - minL;
}

//*****
*

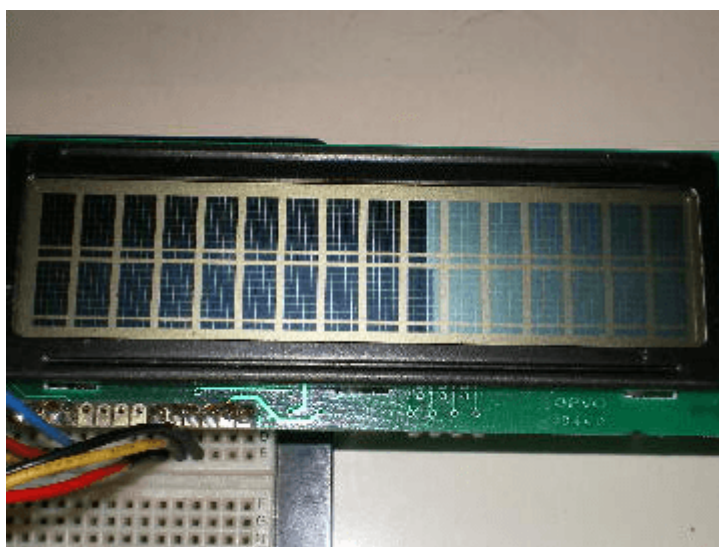
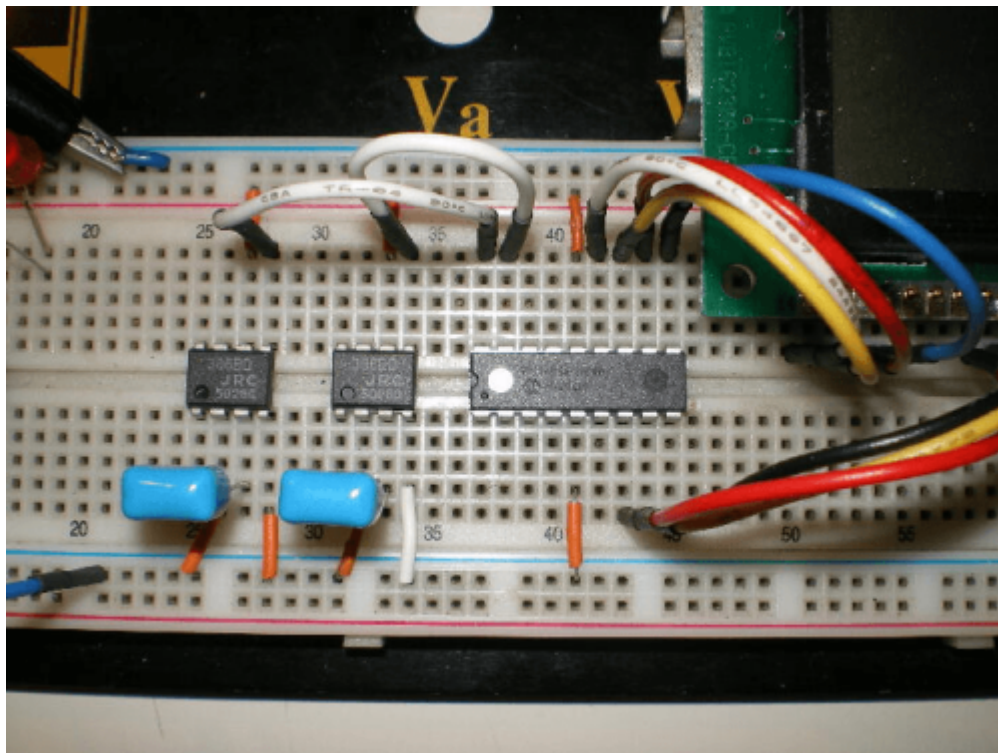
void main()
{
    char    buf[6];
    //
    OSCCON = 0b01110000;           // クロックは8Mhz
    CMCON  = 0b00000111;           // コンパレータは使用しない。
    ANSEL  = 0b00000011;           // □□□変換を使用する。
    TRISA  = 0b11111111;
    TRISB  = 0b00000000;
    //□□□の初期化
    Lcd_Config(&PORTB, 3, 1, 2, 7, 6, 5, 4);
    RegistCustomChar();
    Lcd_Cmd(LCD_CURSOR_OFF);
    Lcd_Cmd(LCD_CLEAR);
}
```

```
//  
while (1) {  
    LED = ON;  
    measurement();  
    LED = OFF;  
    //  
    BarDisp(1, adR);  
    BarDisp(2, adL);  
}  
} //~!  
  
//*****  
*
```

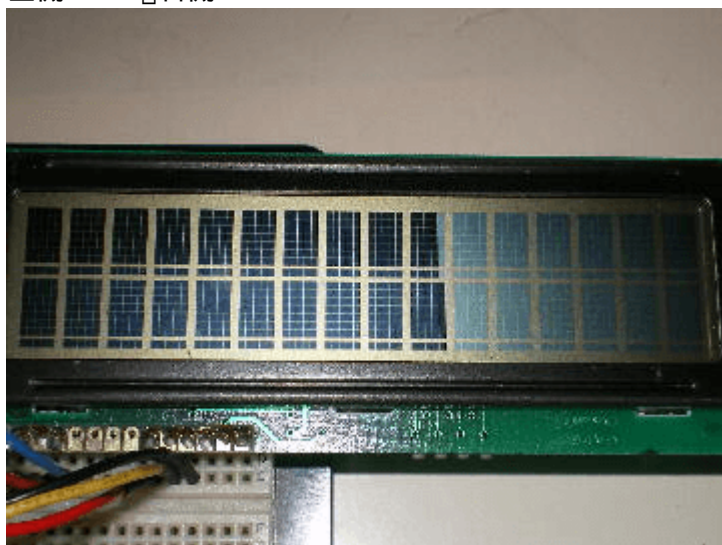
## 動作確認

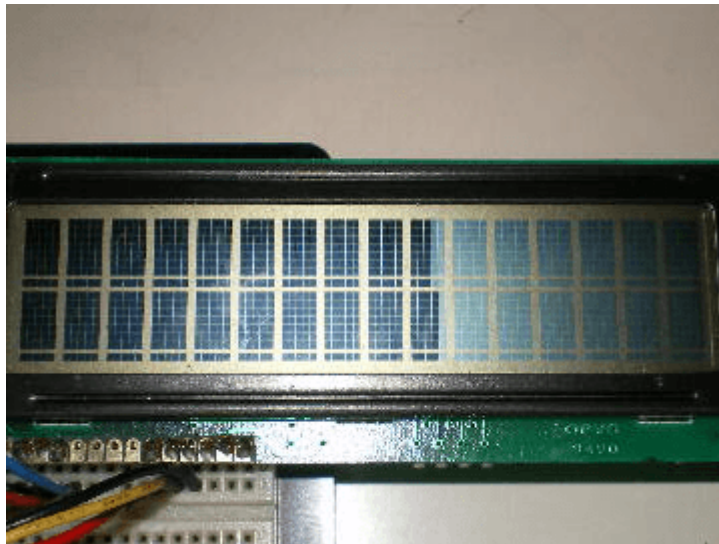


左側から□LM386□LM386□PIC16F88□LCDです。

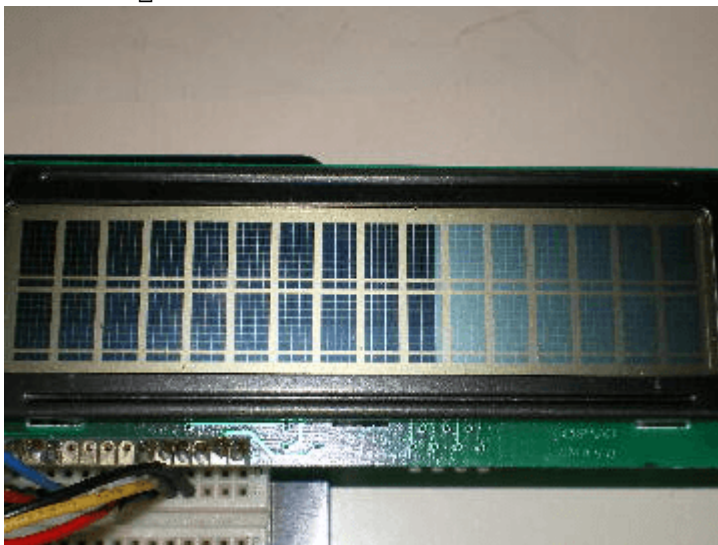


左側:10Hz □ 右側:100Hz

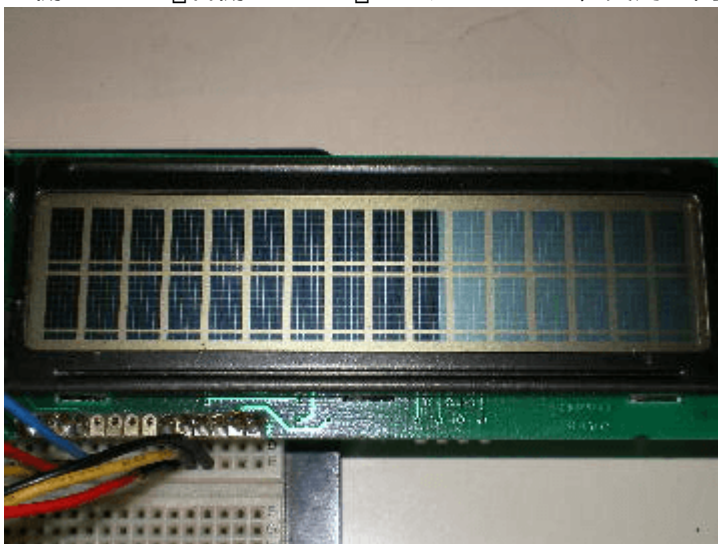


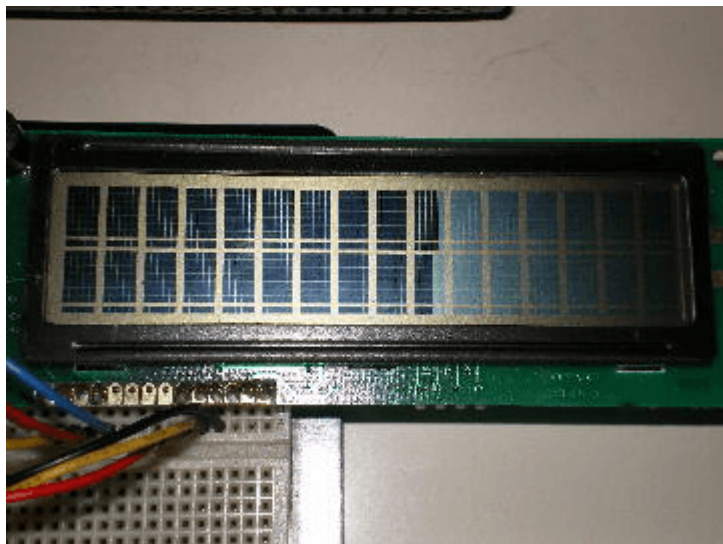


左側:1kHz 右側:10kHz

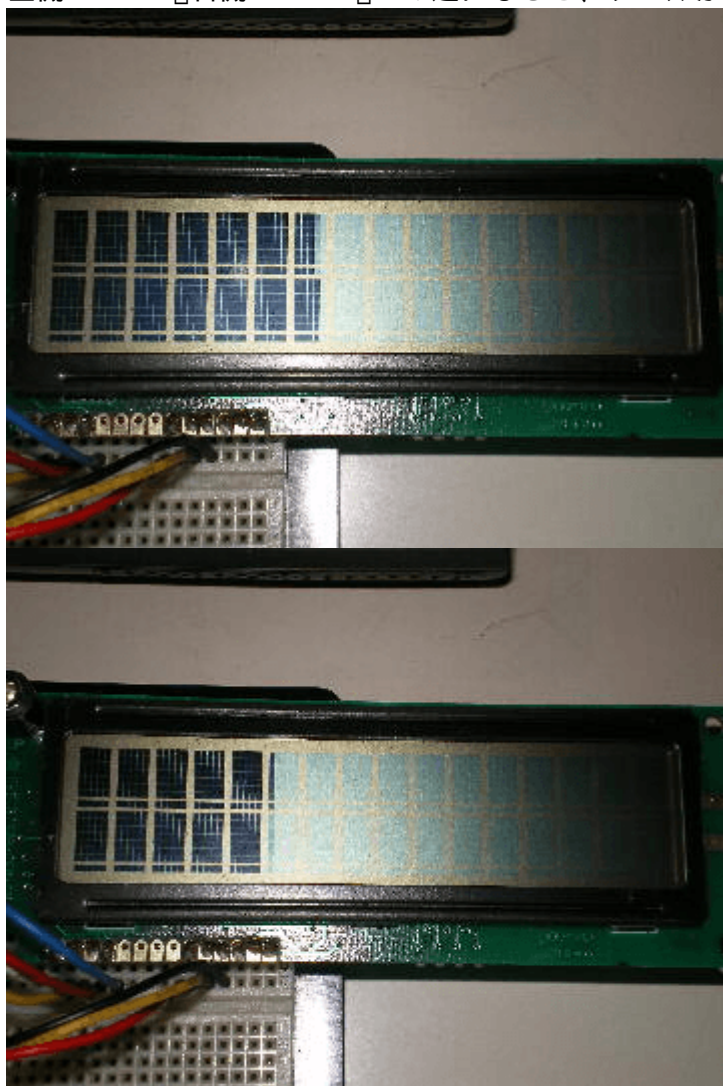


左側:100kHz 右側:200kHz この辺までなら、実用上問題なさそうですね。

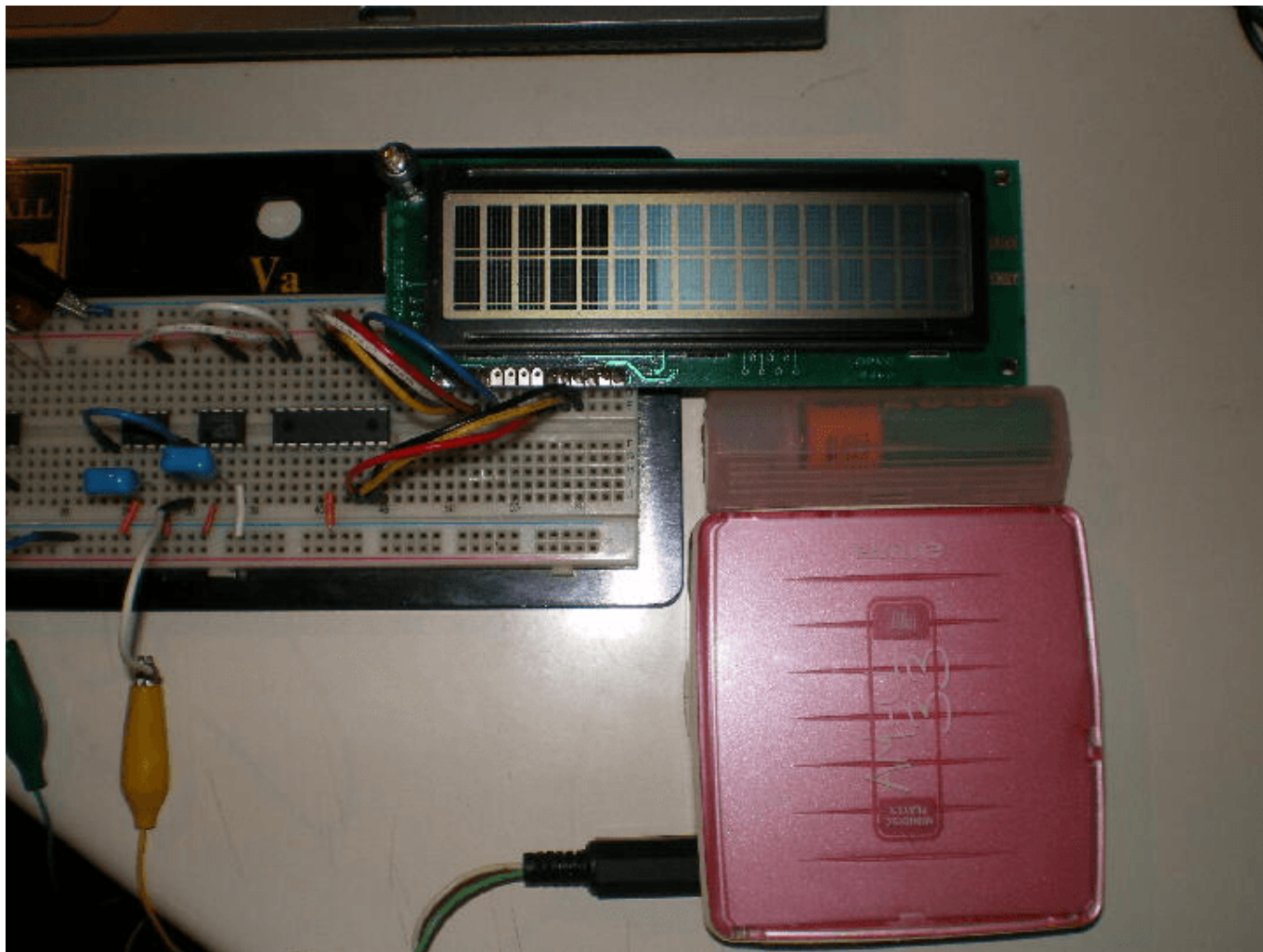




左側:300kHz 右側:400kHz この辺になると、レベルがかなり低下してしまいます。



入力にMDプレーヤを接続してみました。



如何ですか? 80ドットなので、かなり滑らかに変化します。中々宜しいようで。。。{ 😊 }!

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:pic16f88:104&rev=1588205577>

Last update: 2025/10/17 14:27

