

```

88|cm_lcd_main
*****
LC meter with franclin oscilator circuit
1602 display by 4040B counter with PIC16F88
By nobcha all right reserved

```

Ver 0.1 09/20/2010 V0.2 09/28/2010
Hitech C & MPLAB

```

PIC16F88 + LCD via TC4040B
PIN Assign #1 RA1:CAL RELAY
#2 RA0:c/CAL SW status
#3 RA4:counter input
#6 RB0:pulse out for 4040
#7 RB1:EN

```

TMRO is counter with 1/2 prescaler
TMR1 is gate time controller as set (65536-1000)

4040B pin

```

#10 clock in
#11 rst (70uS delayed #10 high state )
#9 seg-a
#7 seg-b
#5 seg-c
#4 seg-d
#6 seg-e
#13 seg-f
#12 seg-g
#14 seg-dp

```

OSC INTERNAL 4MHz

Development Circumstance
MPLAB IDE V8.56 HiTECH C V9.71a

Counter data is put on to RA4 which ia TMRO input.
TMR1 is worked for 10mS gate. TMRO has 1:2 prescaler.
Overflow of TMRO is counted on count_1.

```
*****
```

```

#define pic_clk 4000000

#define MHz 000000
#define _XTAL_FREQ 4MHz // Internal Clock
#define cal_low RA2 // Calibration ON PORT
#define cal_sw RA3 // Calibration status switch
#define chk_led RB4 // Operation check LED

#define c_cal 1000.0 // 1000pF is calibration capacitor

#include <htc.h>
#include <pic.h>
#include <stdio.h>
#include "88_lcd.h"
#include "delay.h"

```

```
__CONFIG(BORDIS & UNPROTECT & PWRREN & WDTON & LVPDIS & MCLREN & INTCLK & DEBUGEN );
```

```

unsigned short freq3, freq2, freq1;
unsigned short ci=0, li=0, test_value, freqcal;

float l_inv, l_int, c_int, f_sq;

unsigned char timeup, count_l, mmm=0 ;

void cnt_setup(void){
    TRISA = 0b00101000; // RA4 disable RA3:CAL:H, L:low RA5:MCR
    TMRO = 0 ; // TMRO clear
    TMR1L = 0; // Clear Low Byte of TMR1
    TMR1H = 216; // Set 216*256 55296
    TMR1L = 240; // Set 240+55296 = 55536 = 65536-10000
}

```

```

88|cm_lcd_main
TMROIF=0;           // TMRO flag off
TMR1IF=0;           // TMR1 flag off
TMR1IE=1;           // TMR1 INT ENABLE
TMROIE=1;           // TMRO INT ENABLE

timeup=0;            // Reset timeup flag
count_=0;            // Reset overtime flag

INTCON=0b11100000;  // GIE=0, PEIE=0, TMROIE=1, INTOIE=0, RBIE=0, TMROIF=0, INTOIF=0, RBIF=0
T1CON = 0b00000001; // T1RUN=1, T1CKPS=00, T1OSCDIS, T1SYNC=1, TMR1CS=0, TMR1ON=1
TRISA = 0b00111000; // RA4 enable
}

void interrupt cnt_int(void){ // Interrupt process routine
    GIE=0;
    if(TMROIF){
        count_++;           // Counter0 overflow occurred
        TMROIF=0;           // INT FLAG RESET
        GIE=1;
    }
    if(TMR1IF){
        TRISA = 0b00101000; // Stop count RA4 disable
        timeup=1;            // Gate time over
        TMR1IF=0;
    }
}

short freq_count(void){
    cnt_setup();           // Frequency counting
    // Counter start set up
    // Waiting gate off 10ms
    GIE=1;                // Interrupt enable
    while(timeup==0){     // Gate time up waiting
        timeup=0;           // Reset flag
        GIE=0;                // Interrupt disable
        // Return frequency
        return((short)((TMRO+count_1*256)*2));
        // (tmr0+(overflow count_1*256))*prescaler
    }
}

void freq_display(unsigned short freq){ // Display freq on LCD
    char k=0, disp_data;          // nnnn.n kHz
    short decimal = 10000;         // Get modulo by deviding
    char zero_sup=1;              // Zero suppress flag
    if(freq>30000){
        freq=99999;
    }
    while(k<5){                  // 5 digits
        disp_data = ((char)((freq/decimal)%10)) | 0x30;
        // Get digit data
        if((disp_data==0x30)&zero_sup & k!=3){
            disp_data=0x20;          // zero suppress
        }
        else{ zero_sup=0; }        // zero suppress release
        lcd_putch(disp_data);      // Display digit
        __delay_ms(5);
        if(k==3){lcd_putch(0xe);}   // Display DP
        __delay_ms(2);
    }
    decimal = decimal / 10;        // 10->1
    k++;
}

void test_display(unsigned short value, char dp){ // Test data display
    char j=0, disp_data;          // nnnnn.uH or pF
    short decimal = 10000;         // Get modulo by deviding
}

```

```

88|cm_lcd_main
char zero_sup=1; // Zero suppress flag
while(j<5){ // 5 digits
    disp_data = ((char)((value/decimal)%10)) | 0x30;
    // Get digit data
    if((disp_data==0x30)&zero_sup & j!=(dp)){
        disp_data=0x20; // except dp digit
    }
    else{ zero_sup=0; } // zero suppress release
    lcd_putch(disp_data); // Display digit
    if (j==dp & dp!=4) { lcd_putch(0x2e); } // Display DP
    decimal = decimal / 10; // 10->1
    j++;
}
}

float freq_cal(short f1, short f2){ // Coefficient calculating
    return((float)(f1)/(float)(f2)*(float)(f1)/(float)(f2)-1);
}

void main(){
    unsigned char i, l_power, l_digi, l_unit, c_power, c_digi, c_unit;

/* processor hardware depend */
    TRISA = 0b00111000; // RA1,4 input the others output
    ANSEL = 0b00000000; // No AD port
    TRISB = 0b00000000; // RB2(RX) input

    PORTA = 0b00000100; // PORTA clear RA2:CAL C ON
    PORTB = 0b00000000; // PORTB CLEAR
    OPTION = 0b00110000; // PORTB pullup, INTEDG 0, TOCS TOCKI 1, ToSE1, PSA TIMER0, 1/4

    GIE=0; // INT off
    for(i=0;i<2;i++){
        __delay_ms(20);
    }
    chk_led=1;
    OSCCON = 0b01101110; // 4MHz Internal
    for(i=0;i<5;i++){
        __delay_ms(20);
    }

    T1CON = 0; // Timer1 off
    INTCON = 0b01100000; // GIE 0 PEIE 1, TMROIE 1, INTOIE 0, RBIE 0, TMROIF 0, INTOIF 0, RBIF 0
    PIE1 = 0b00000001; // ADIE 0, RCIE 0, TXIE 0, SSPIE 0, CCP1IE 0, TMR2IE 0, TMR1IE 1
    PIR1 = 0b00000000; // ADIF 0, RCIF 0, TXIF 0, SSPIF 0, CCP1IF 0, TMR2IF 0, TMR1IF 1
    PIE2 = 0x0; // No use
    PIR2 = 0x0; // No use
/* Processor hardware depend */

/* Display initial and put on initial messages */
    __delay_ms(2);

while(1){
    lcd_init();
    __delay_ms(2);

    lcd_goto(0); // select first line
    __delay_ms(2);
    lcd_puts("LCM F1:");
    lcd_puts("SET CALSW");
    for(i=0;i<50;i++){
        __delay_ms(20);
    }
}

while(cal_sw==0){ // CAL switch check
    __delay_ms(20);
}

```

```

88|cm_lcd_main
while(cal_sw==0){           // Check again
    __delay_ms(20);
/* Get frequency 1 and display          */
    lcd_goto(7);                  // Set position
    __delay_ms(20);

    freq3=freq_count();           // F1 get
    freq1=freq3;
    freq_display(freq3);         // display F1
    lcd_puts("kHz");            //

/* Turn on CAL relay and get frequency 2      */
    cal_low=0;                   // Calibration capacitor on
    __delay_ms(30);              // Wait 30ms

    freq2=freq_count();           // Get F2
    if(freq2<200){               // Too low freq2 be error
        lcd_puts("ER");
    }

    cal_low=1;                   // Capacitor off

/* F2 displaying                      */
    lcd_goto(40);                // Move cursor to top of 2nd line
    __delay_ms(10);              // Wait 20ms
    freq_display(freq2);         // display F2

/* Calculating C and L             */
    c_int=(float)(c_cal)/freq_cal(freq1,freq2);
    f_sq=(float)(freq1)*(float)(freq1)/+1.000E+2;
    l_inv=(+3.9438E+1)*f_sq*c_int;
    l_int=(+1.000E+14)/l_inv;

/* for debugging                    */
    test_display((short)(c_int),5); // Display c_int value
    __delay_ms(20);              // Wait 20ms
    test_display((short)(l_int),2); // Display l_int value
    for(i=0;i<50;i++){          // Wait 1 second
        __delay_ms(20);
    }
    while(cal_sw==1){ }          // Check sw status again
    for(i=0;i<10;i++){          // Wait 0.2 second
        __delay_ms(20);
    }
/* Measuring starting message       */
    lcd_puts("SET TEST");        // Ask setting L or C

    while(1){                   // Repeat forever
        for(i=0;i<50;i++){      // Wait 1 second
            __delay_ms(20);
        }

        freq3=freq_count();      // Get F3
        chk_led=chk_led+1;       // LED on/off

/* displaying value                 */
        lcd_goto(40);            // Move cursor to top of line
        __delay_ms(1);

/* Metric unit changing           */
        l_power=100;
        l_digi=1;
        l_unit='m';

```

```

88|cm_lcd_main
if(freq3<30){lcd_puts("error");} // Under 2kHz error
else if(freq3<400){
}
else{
    l_power=1;
    l_digi=0;
    l_unit='u';
    c_power=1;
    c_digi=0;
    c_unit='p';
}

/* Calculating value */
if(cal_sw){                                */
    test_value= (unsigned short)((freq_ca(freq1,freq3))*l_int/10/l_power);
    test_display(test_value,3-l_digi);        // 5digits displaying
    __delay_ms(10);
    lcd_putch(l_unit);
    lcd_puts("Hz");
}

else{
    test_value= (unsigned short)((freq_ca(freq1,freq3))*c_int/c_power);
    test_display(test_value,4-c_digi);        // 5digits displaying
    __delay_ms(10);
    lcd_putch(c_unit);                     // If calibration_switch==0 C measuring
}

__delay_ms(20);
freq_display(freq3);                         // display F3
lcd_puts("kHz ");
}
}
}

```