

```

/*****

```

```

Simple frequency counter by PIC12F1822
I2C LCD display and internal clock
By nobcha all right reserved

```

```

Ver 0.1 08/25/2010 for PIC12F629
Ver 0.2 02/15/2011 for PIC12F683 OSCCAL, ANSEL
Ver 2.0 06/11/2011 for PIC12F683 INT clock and I2C LCD ORG CODE
Ver 5.0 07/26/2011 for PIC16F1822 INT clock and I2C LCD ORG CODE

```

```

PIC12F683 + LCD via I2C
PIN Assign

```

```

#6 RA1:Monitor LED
#5 RA2:FREQ IN/TMRO
#3 RA4/SCL I2C clock
#2 RA5/SDA I2C data

```

```

TMRO is counter with 1/2 prescaler
TMR1 is gate time controller as set (65536-20000)
10mS Time gate 6.5MHz max count

```

```

OSC INT 4MHz

```

```

Development Circumstance
MPLAB IDE V8.73 HiTECH C V9.81

```

```

Counter data is put on to RA2 which is TMRO input.
TMR1 is worked for 10mS gate. TMRO has 1:2 prescaler.
Overflow of TMRO is counted on count_1.

```

```

*****/

```

```

#define _XTAL_FREQ 4000000

```

```

#include <htc.h>
#include <pic.h>
#include <stdio.h>
#include "delay.h"
#include "lcd_i2c_func.h"
unsigned char Msg1[17] = "Frequency Gate";
unsigned char Msg2[5] = "10mS";
unsigned char Msg3[4] = "";
unsigned char Msg4[4] = "kHz";
unsigned char Msg5[4] = "OF ";

```

```

__CONFIG(
    FOSC_INTOSC & WDTE_OFF & PWRTE_ON & MCLRE_ON & CP_OFF
    & CPD_OFF & BOREN_OFF & CLKOUTEN_OFF & IESO_OFF & FCMEN_OFF
);

```

```

__CONFIG(
    WRT_OFF & PLLEN_OFF & STVREN_ON & LVP_OFF
);

```

```

unsigned char timeup;
long read_data, count_1;

```

```

void cnt_setup(void) {

```

```

    PORTA = 0b00110100; // bit4,5:high
    TRISA = 0b00111000; // io2:output, io3:in, io4:SCL, io5:SDA
    TMRO = 0; // TMRO clear
    TMR1L = 0; // Clear Low Byte of TMR1
    TMR1H = 177; // Set 177*256 + 224
    TMR1L = 224; // Set (177*256)+208=45536=65536-20000

    TOIF=0; // TMRO flag off
    TMR1IF=0; // TMR1 flag off
    TMR1IE=1; // TMR1 INT ENABLE
    TOIE=1; // TMRO INT ENABLE
    PIE1=0b00000001; // ADIE 0, RCIE 0, TX1E 0, SSP1E 0, CCP1IE 0, TMR2IE 0, TMR1IE 1

    timeup=0; // Reset timeup flag
    count_1=0; // Reset overtime flag

```

```

        INTCON=0b01100000;           // 1822_cnt_lcd_main
        // GIE 0,PEIE 1,TOIE 1,INTE 0,GPIE 0,TOIF 0,INTF 0,GPIF 0
        T1CON = 0b00000001;         // T1RUN 1,T1CKPS 00,T1OSCDIS,T1SYNC 1,TMR1CS 0,TMR1ON 1
        TRISA = 0b00111100;         // io2:input,io3:in,io4:SCL,io5:SDA
    }

void interrupt cnt_int(void){
    GIE=0;
    if(TOIF){
        count_1++;                 // Timer 0 overflow occurred
        TOIF=0;                     // FLAG CLEAR
        GIE=1;                       // INT ENABLE
    }
    if(TMR1IF){
        TRISA=0b00111100;         // STOP GPIO2->output
        // RA2->output
        TMR1ON=0;                 // TMR1 STOP
        timeup=1;                 // Gate time over
        TMR1IF =0;
    }
}

void main(){
    unsigned char i, zero_sup,disp_data ;
    short decimal;

    PORTA = 0b00110100;           // PORT clear
    ANSELA = 0;                   // GPIO all digital
    TRISA = 0b00111100;           // io2:data in,io3:reset,io4:SCL,io5:SDA
    OSCCON = 0b01101000;         // OSTS 4MHz

    OPTION_REG = 0b00110001;     // W pullup,INTEDG 0,TOCS TOCKI 1,ToSE1,PSA TIMERO,1/4

    INTCON=0;                       // INT off
    T1CON = 0;                     // Timer1 FOSC/4, T1CKPS:00, T1OSC DIS,T1SYNC, TMR1 off
    T1GCON = 0b00000000;
    PIR1 = 0b00000000;           // ADIF 0,RCIF 0,TXIF 0,SSPIF 0,CCP1IF 0,TMR2IF 0,TMR1IF 0

    PORTA = 0b00110110;           // LED set GPIO1
    __delay_ms(200);
    __delay_ms(200);
    Tcd_init();

    lcd_goto(0x00);                 // select first line
    lcd_str(Msg1);                 // "Frequency Gate";

    lcd_goto(0x4C);                 // 2nd line 12th char
    lcd_str(Msg2);                 // "10mS";

    lcd_goto(0x40);                 // select second line
    __delay_us(200);
    cnt_setup();                   // counter initial set up
    __delay_us(10);

    __delay_ms(200);
    __delay_ms(200);

    while(1){
        GIE=1;                       // INT ENABLE
        if(timeup){
            timeup=0;               // When time up, display counter value
            GIE=0;
            read_data=((long)(TMR0)+(count_1)*256)*2; // (tmr0+(overflow
count_1*256))*prescaler
            if(count_1>100){lcd_str(Msg5);} // more than 64 could not be covered
        }
        else{lcd_str(Msg3); }
        i=0;
    }
}

```

```

                                1822_cnt_lcd_main
decimal = 10000;
zero_sup=1;                      // Zero suppress flag
while(i<5){                      // 5digit
    disp_data = ((char)((read_data/decimal)%10)) | 0x30; // Get digit data
    if((disp_data==0x30)&zero_sup & i!=3 ){
        disp_data=0x20;        // zero suppress
    }
    else{ zero_sup=0;}         // zero suppress release
    lcd_data(disp_data );      // Display digit
    decimal = decimal / 10;    // 10->1
    i++;
    if(i==4){lcd_data( 0x2e);} // Display DP
}
lcd_str(Msg4);                 // "kHz"
PORTA = 0b00110110;          // LED set GPI01
__delay_ms(200);
Tcd_goto(0x40);               // select second line
cnt_setup();                  // Gate open Counter start
}
}
}

```