```
/**************************************************
  Simple frequency counter by PIC12F683
   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

   PIC12F683 + LCD via I2C
   PIN Assign
                #6 GPIO1:Monitor LED
                #5 GPIO2:FREQ IN/TMR0
                #3 GPIO4/SCL I2C clock
                #2 GPIO5/SDA I2C data


   TMR0 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 V8.60   HiTECH C V9.71a

 Counter data is put on to GPIO2 which ia TMR0 input.
 TMR1 is worked for 10mS gate. TMR0 has 1:2 prescaler.
 Overflow of TMR0 is counted on  count_1.
**************************************************/
#define _XTAL_FREQ 8000000

#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(BORDIS & UNPROTECT & PWRTEN & WDTDIS & MCLREN & INTIO  );

unsigned char timeup ;
long read_data, count_1;

void cnt_setup(void){

        GPIO = 0x34;                    // bit4,5:high
        TRISIO = 0x38;                  // io2:output,io3:in,io4:SCL,io5:SDA
        TMR0 = 0 ;                      // TMR0 clear
        TMR1L = 0;                      // Clear Low Byte of TMR1
        TMR1H = 177;                    // Set 177*256 + 224
        TMR1L = 224;                    // Set (177*256)+208=45536=65536-20000

        T0IF=0;                         // TMR0 flag off
        TMR1IF=0;                       // TMR1 flag off
        TMR1IE=1;                       // TMR1 INT ENABLE
        T0IE=1;                         // TMR0 INT ENABLE
        PIE1=0b00000001;                // ADIE 0,RCIE 0,TXIE 0,SSPIE 0,CCP1IE 0,TMR2IE 0,TMR1IE 1

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

        INTCON=0b01100000;              // GIE 0,PEIE 1,T0IE 1,INTE 0,GPIE 0,T0IF 0,INTF 0,GPIF 0

        T1CON = 0b00000001;             // T1RUN 1,T1CKPS 00,T1OSCDIS,T1SYNC 1,TMR1CS 0,TMR1ON 1
        TRISIO = 0x3C;                  // io2:input,io3:in,io4:SCL,io5:SDA
        }

void interrupt cnt_int(void){
        GIE=0;
        if(T0IF){
                count_1++;              // Timer 0 overflow occured
                T0IF=0;                 // FLAG CLEAR
                GIE=1;                  // INT ENABLE
                }
        if(TMR1IF){                     // STOP GPIO2->output
                TRISIO = 0x38;          // io2:output,io3:in,io4:SCL,io5:SDA
                timeup=1;               // Gate time over
                TMR1IF =0;
                }

        }

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


        GPIO = 0b00110100;              // PORT clear without bit 2,4,5
        ANSEL = 0;                      // GPIO all digital
        TRISIO = 0x3C;                  // io2:data in,io3:reset,io4:SCL,io5:SDA
        OSCCON = 0b01111000;            // OSTS 8MHz

        OPTION = 0b10110000;            // PORTB pullup,INTEDG 0,TOCS TOCKI 1,ToSE1,PSA TIMER0,1/4

        INTCON=0;                       // INT off
        T1CON = 0;                      // Timer1 off
        PIR1 = 0b00000000;              // ADIF 0,RCIF 0,TXIF 0,SSPIF 0,CCP1IF 0,TMR2IF 0,TMR1IF 0



        GPIO = 0b00110110;              // LED set GPIO1
        __delay_ms(200);
        __delay_ms(200);
        lcd_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 short
                else{lcd_str(Msg3); }

                i=0;
                decimal = 10000;
                zero_sup=1;                             // Zero suppres 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 suppres
                                }
                        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"
                GPIO = 0b00110110;                      // LED set GPIO1
                __delay_ms(200);
                lcd_goto(0x40);                         // select second line
                cnt_setup();                            // Gate open Counter start
                }
        }
}
```