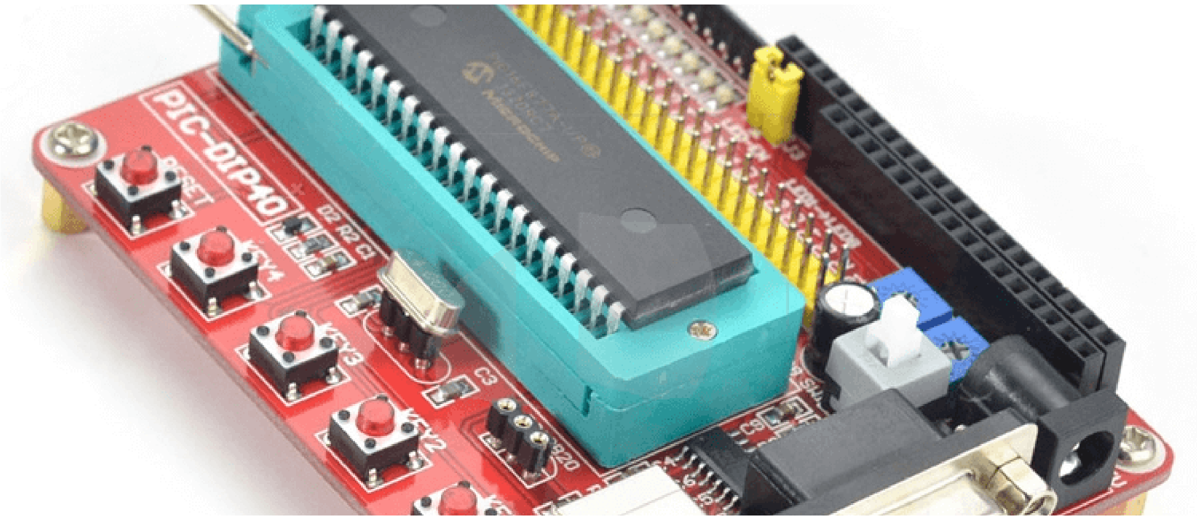


EE302 Project - LCD Rotation Display

Name	**Zhuoran Wang**
FZU ID	**831902126**
MU ID	**19105339**

🎉Part 1 - System Design

What module did I use? (A `checklist`)

☑ ~~Basic I/O - LED~~

☑ ~~ADC~~

☑ ~~LCD - display at different speeds~~

☑ ~~UART~~

☑ ~~I2C and EEPROM~~

☑ ~~Interrupt~~

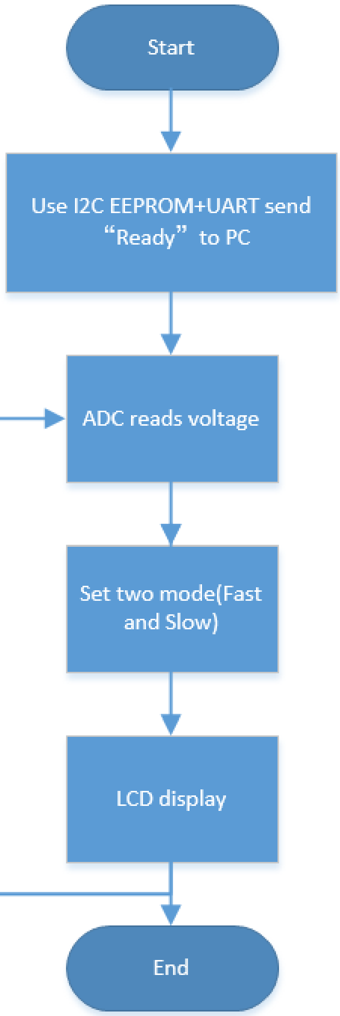
☑ ~~Timer~~

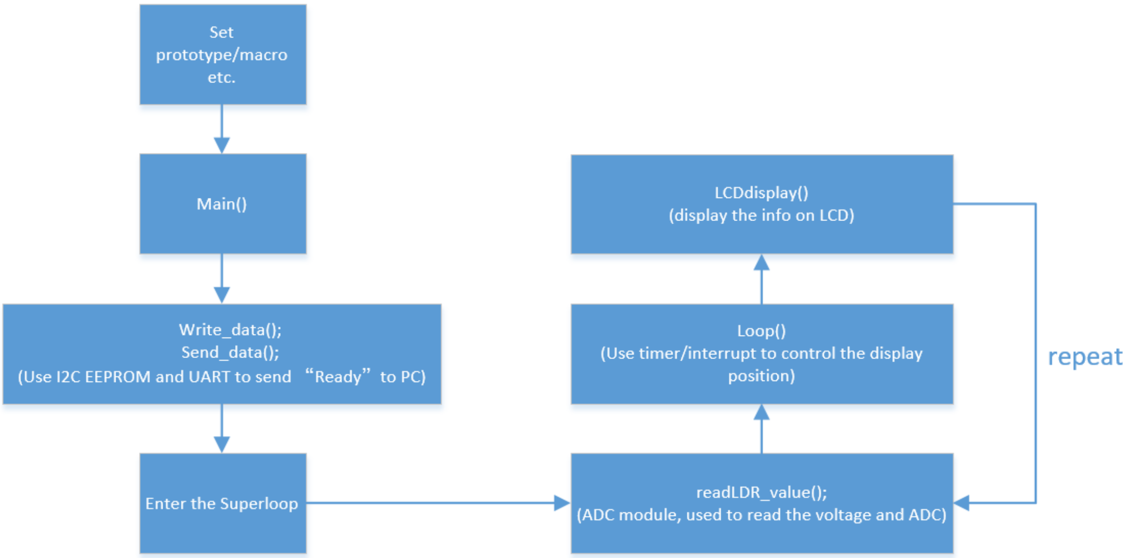
Function

In this project, I designed a **Light Controlled LCD Rotation Display.**

1. First, the MCU uses **I2C** to write “Ready” Information to **EEPROM**
2. Then use **UART** to send the information in EEPROM to PC
3. After that, the **ADC** module converts the Photosensitive resistance controlled voltage into discrete values.
4. According to this value, set the rotation display speed to “Fast” or “Slow”.
5. The **LCD** displays a pre-defined string at different speeds.
6. And the **LED** can be RED or GREEN according to the conditions.
7. Use **Timer** and **Interrupt** to control the speed of rotation display.
8. Then we can see the display string speed changes.
9. When the system is working, use **UART** to send “working” information to the PC.

Block diagram


Start
Use I2C EEPROM+UART send "Ready" to PC
ADC reads voltage
Set two mode(Fast and Slow)
LCD display
End


Set prototype/macro etc.
Main()
Write_data();
Send_data();
(Use I2C EEPROM and UART to send "Ready" to PC)
Enter the Superloop
readLDR_value();
(ADC module, used to read the voltage and ADC)
Loop()
(Use timer/interrupt to control the display position)
LCDdisplay()
(display the info on LCD)
repeat

🕓Part 2 - Real-Time Aspect

The system is a soft real-time system, which means the deadline can be missed.

The first aspect is that the I2C EEPROM work can be done within a specific time.

The second aspect is that the UART sending information can be done at a specific time.

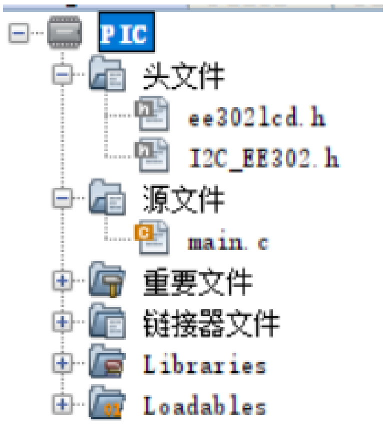
The third aspect is that the Analog to digital conversion is within the desired time.

The Timer and Interrupt control the speed of rotation display, thus it is real-time because the timer counts time accurately.

🛠️Part 3 - Project Code

The files are configured as follows:

The project has one main code file (`main.c`) and includes three header files (`ee302lcd.h` and `I2C_EE302.h` and `xc.h`).



```
/*
 * File:   main.c
 * Created by: Zhuoran Wang
 *
 * Created on 25th Dec 2021
 */

// CONFIG
#pragma config FOSC = XT   // Oscillator Selection bits (XT oscillator)
#pragma config WDTE = OFF  // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = OFF // Brown-out Reset Enable bit (BOR disabled)
#pragma config LVP = OFF   // Low-Voltage (Single-Supply) In-Circuit Serial
// Programming Enable bit (RB3 is digital I/O, HV
// on MCLR must be used for programming)
#pragma config CPD = OFF // Data EEPROM Memory Code Protection bit
//(Data EEPROM code protection off)
#pragma config WRT = OFF // Flash Program Memory Write Enable bits (Write
// protection off; all program memory may be
// written to by EECON control)
#pragma config CP = OFF // Flash Program Memory Code Protection bit
```

```
//(Code protection off)

#include <xc.h>       // Include standard PIC library
#include "ee302lcd.h" // Include required header file for LCD functions
#include "I2C_EE302.h"

#ifndef _XTAL_FREQ
// Unless already defined assume 4MHz system frequency
// This definition is required to calibrate the delay functions, __delay_us() and __delay_ms()
#define _XTAL_FREQ 4000000
#endif

#define SW1 RB0 // Assign Label SW1 to PortB bit 0 (RB0)
#define CLOSED 0
#define OPEN 1
#define HIGH 1
#define LOW 0
#define hi 0x11
#define lo 0x55
#define G_led RC0
#define R_led RC1
//prototypes
void setup(void);
void loop(void);
char receive(void);
void send_str(char *str);
void LCDTitle(void);
void readLDR_value(void);
void Write_data(void);
void Send_data(void);
void receive_data(void);
void Light();
// Global variables:
unsigned char gMode;
char gSpeed;
unsigned char new_value = 0;
unsigned char ad_value = 0;
int value1 = 0;
int value2 = 0;
int thr = 1;
unsigned char gOutString[16];
int i = 0;
char ch = 0;
int col = 16;
int flag = 1; //是否选择了某个模式

void main()
{
    setup(); // do initialisation

    Write_data();
    Send_data();

    for (;;) // endless loop
    {
        readLDR_value();
        loop();
        LCDTitle();
        send_str("working");
    }
}

void setup(void) // setup stuff
```

```
{
    PORTD = 0b11111111;
    TRISD = 0b00000000;
    //TRISC = 0xC0; //RC6 and RC7 must be set to inputs for USART.
    TXSTA = 0x24; //Set TXEN bit to enable transmit.
                  //Set BRGH bit for Baud Rate table selection.
    RCSTA = 0x90; //Set CREN bit for continuous read.
    //Set SPEN bit to enable serial port.
    SPBRG = 0x19; //Set Baud Rate to 9600

    TRISC = 0xd8;
    TRISD = 0x00;

    T1CON = 0x21;
    INTCON = 0xc0;
    PIE1 = 0x21;
    PIR1 = 0x00;
    Lcd8_Init();  // Required initialisation of LCD to 8-bit mode
    TRISB = 0x07; // Set PORTB bit 0 as input

    TRISC = 0xd8;
    TRISD = 0x00;
    PORTD = 0xff;

    TXSTA = 0x24;
    RCSTA = 0x90;
    SPBRG = 0x19; // set baud rate

    T1CON = 0x21;
    INTCON = 0xc0;
    PIE1 = 0x21;
    PIR1 = 0x00;
    //Set the ACD registers

    TRISA = 0b00000101; // Set PORTA bits 0 and 2 are output
    //TRISC = 0b00000000; // Set PORTC bit 1 and 0 as output
    PORTC = 0b00000010;
    ADCON0 = 0b01010001; // Set FOSC/8,RA2 as analog input and A/D converter module is powered up
    ADCON1 = 0b00000010; // Set Left justified
    OPTION_REG &= 0b01111111;



    TRISC = 0b1101100; //RC6 and RC7 must be set to inputs for USART.
    TXSTA = 0x24; //Set TXEN bit to enable transmit.
                    //Set BRGH bit for Baud Rate table selection.
    RCSTA = 0x90; //Set CREN bit for continuous read.
    //Set SPEN bit to enable serial port.
    SPBRG = 0x19; //Set Baud Rate to 9600
    i2c_init();




    PORTD = 0b11111111;
    TRISD = 0b00000000;
    //TRISC = 0xC0; //RC6 and RC7 must be set to inputs for USART.
    TXSTA = 0x24; //Set TXEN bit to enable transmit.
                    //Set BRGH bit for Baud Rate table selection.
    RCSTA = 0x90; //Set CREN bit for continuous read.
    //Set SPEN bit to enable serial port.
    SPBRG = 0x19; //Set Baud Rate to 9600
}
```

```
void loop(void)
{
        //通过光照调节速度
        if (value1 > thr)
        {
            T1CON = 0x01;
            //send_str("Fast Mode");
            //LCDTitle();
            R_led = LOW;
            G_led = HIGH;
        }
        else if (value1 <= thr)
        {
            T1CON = 0x31;
            //LCDTitle();
            //send_str("Normal Mode");
            R_led = LOW;
            G_led = HIGH;
        }
}

void __interrupt() // Interrupt identifier
    isr(void)      // Here be interrupt function - the name is
// unimportant.
{
    if (TMR1IF)
    {
        TMR1IF = 0;
        i++;
        if (i == 2) {
            col--;
            i = 0;
        } else {

        }
    }
    else if (RCIF)
    {
        RCIF = 0;
        ch = RCREG;
    }
}

char receive(void)
{
    RCIF = 0;
    while (!RCIF)
        ;
    return RCREG;
}

void send_str(char *str)
{
    int index = 0;
    char ch = *str;

    while (ch != '\0')
    {
        ch = *(str + index);
        index++;
        while (!TXIF)
            ;
        TXREG = ch;
    }
}


//Display title on LCD
void LCDTitle(void)
{
    //int row = 1, col = 16;
    //分别代表第几行第几列. 最多有16列
    if (col < 0)
    {
        col = 16;
    }
    Lcd8_Clear();
    Lcd8_Set_Cursor(2, col);    // select line 2 of LCD
    Lcd8_Write_String("Hello"); // display "EE302" on second line of LCD
    __delay_ms(300);
}


void readLDR_value(void){
    if(1){
        __delay_ms(150);
        //while(1){
            __delay_us(50);
            GO_nDONE = 1;
            while(GO_nDONE){
                continue;
            }
            if(ADRESH!=new_value){
                ad_value = ADRESH;
                value1 = (ad_value*5/255);
                value2 = (ad_value*10*5/255)%10;

            }

            Light();
            new_value = ad_value;
           __delay_ms(100);
        }
    //}
}

void Write_data(void)
{
    unsigned char address_hi = hi;
    unsigned char address_lo = lo;
    unsigned char data[12] = "Ready      ";
    int i = 0;
    while (i <= 10)
    {
        write_ext_eeprom(address_hi, address_lo, data[i]);
        address_lo++;
        i++;
    }
}
void Send_data(void)
{
    unsigned char address_hi = hi;
    unsigned char address_lo = lo;
    int i = 0;
    while (i <= 10)
    {
        while (!TXIF)
            ;
        TXREG = read_ext_eeprom(address_hi, address_lo);
        address_lo++;
        i++;
        __delay_us(500);
    }
}

void Light(){

    if(value1+value2*0.1<1.5){
        R_led = LOW;
        G_led = HIGH;
    }else{
        R_led = HIGH;
        G_led = LOW;

    }
}
```