Mina Rizk
CU1901039
12/13/2021

Report Writing (CW2)

Lab Report

Assignment 2: **Human Machine Interface**

Mina Rizk/CU1901039

Embedded System Design and Development

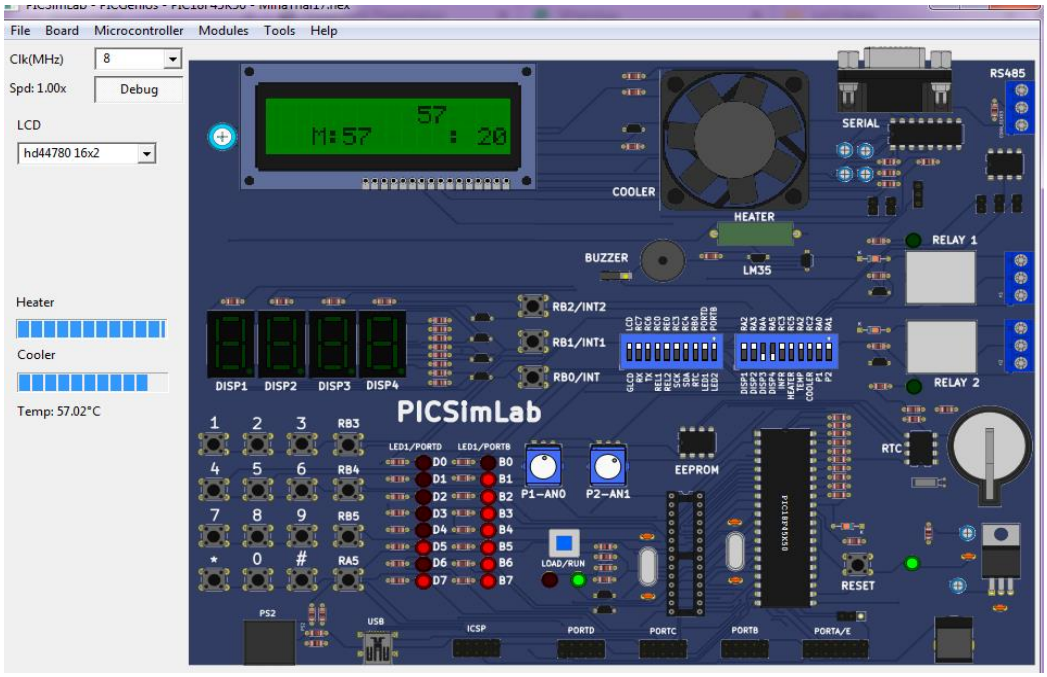
Mina Rizk
CU1901039
12/13/2021

Table of contents

Mina Rizk
CU1901039
12/13/2021

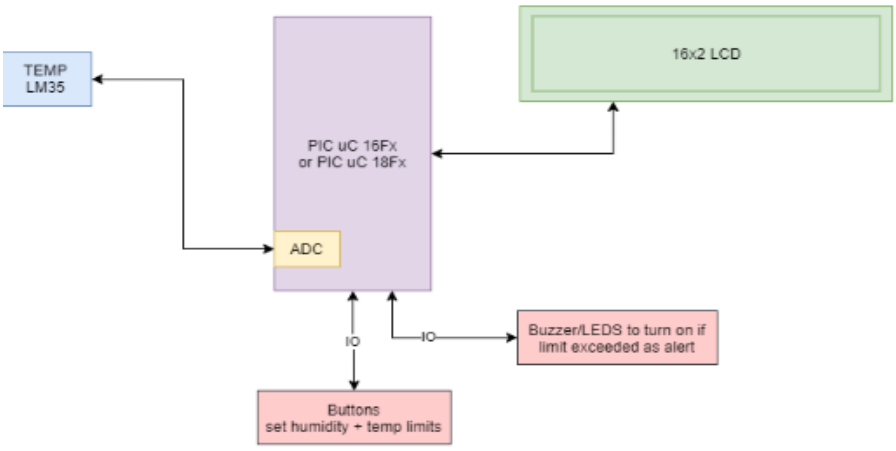
Introduction

PICSimLab is a realtime emulator of development boards with integrated MPLABX/avr-gdb debugger. PICSimLab supports some picsim microcontrollers and some simavr microcontrollers. PICSimLab have integration with MPLABX/Arduino IDE for programming the boards microcontrollers. As the purpose of PICSimLab is to emulate real hardware it does not have any source code editing support. For code editing and debugging the same tools used for a real board should be used with PICSimLab, such as MPLABX or Arduino IDE



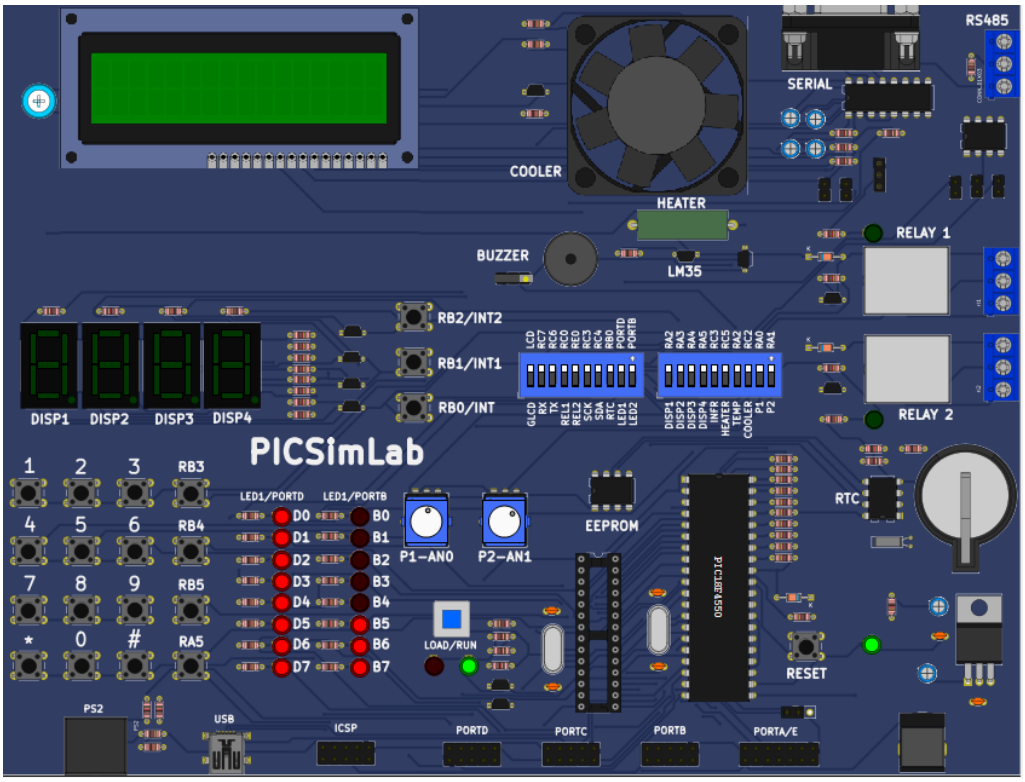
- ## Purpose of the lab

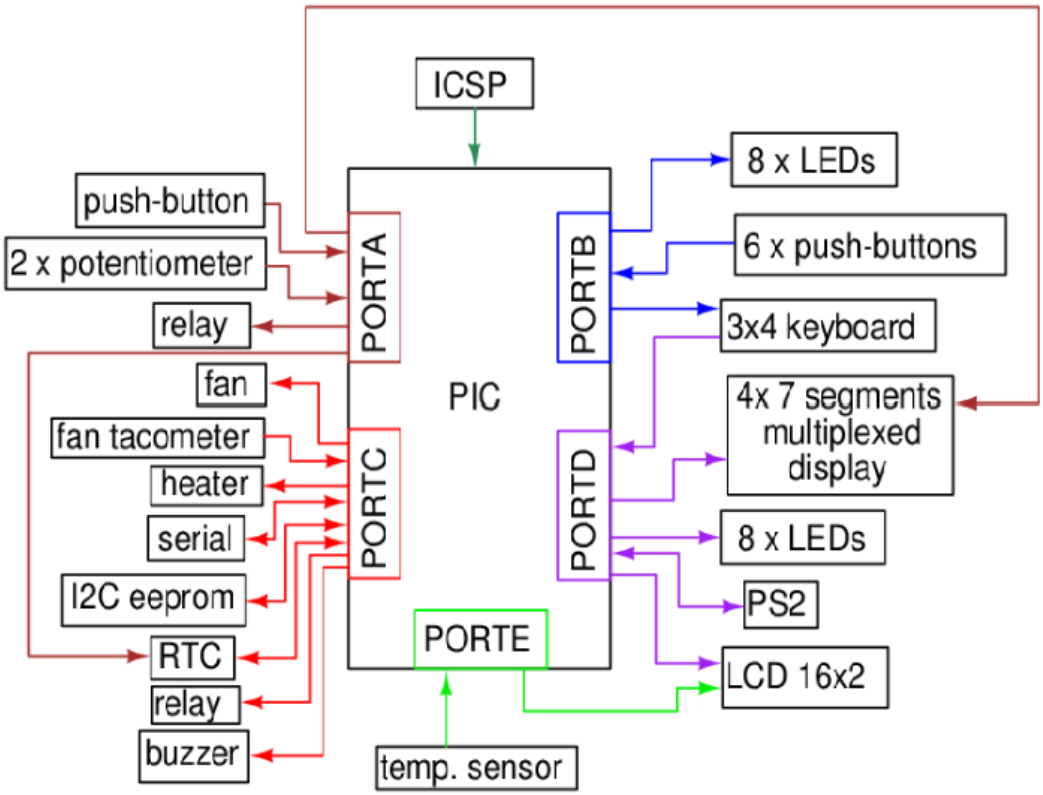
  - Construct LCD 16x2 Interface that displays humidity and temperature sensor controlled sensors via I2C Interface .



Mina Rizk
CU1901039
12/13/2021

2.0 System design architecture





Mina Rizk
CU1901039
12/13/2021

I. ***CODE:mickro c***

```
// LCD module connections
sbit LCD_RS at RE2_bit;
sbit LCD_EN at RE1_bit;
sbit LCD_D0 at RD0_bit;
sbit LCD_D1 at RD1_bit;
sbit LCD_D2 at RD2_bit;
sbit LCD_D3 at RD3_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D7 at RD7_bit;

sbit LCD_RS_Direction at TRISE2_bit;
sbit LCD_EN_Direction at TRISE1_bit;
sbit LCD_D0_Direction at TRISD0_bit;
sbit LCD_D1_Direction at TRISD1_bit;
sbit LCD_D2_Direction at TRISD2_bit;
sbit LCD_D3_Direction at TRISD3_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;
// End LCD module connections

unsigned char high_temperature_limit=57;
unsigned char low_temperature_limit=20;
void InitLCD()
{
```

```
ANSELD=0;
TRISD=0;
PORTD=0;
TRISE=0;
PORTE=0;
ANSELE=0;

Lcd_Init();                    // Initialize LCD module
Lcd_Cmd(_LCD_CLEAR);           // clear LCD
Lcd_Cmd(_LCD_CURSOR_OFF);      //cursor off
Lcd_Out(1,2,"Embedded systems");
Lcd_Out(3,2,"Assement");
Delay_ms(2500);
Lcd_Cmd(_LCD_CLEAR);
}
void InitAdc()                 //Initialize ANOLOG to DIGITAL
void
{
ANSELA=0xff;                   // set all pins in PORT A as analog
pins
TRISA-0xff;                   //set all pins in PORT A as input pins
ADC_Init();
}
void InitHeater()              //Initialize the Heater
{
ANSELC=0;                      // set all pins in PORT C as analog
pins
TRISC=0;                      //set all pins in PORT C as input pins
RC5_bit=1;
}
void InitThresholds()          //Initialze the thresholds
```

```
{
ANSELB=0;                    // set all pins in PORT B as analog pins
TRISB=0xff;                  //set all pins in PORT B as input pins
}
float GetTemp()
{
int adc_of_temp;
float temp_in_celsius;
adc_of_temp=ADC_Read(2);              //ReadADC ON PIN(2)
temp_in_celsius=5*adc_of_temp/10.24;  //Transform the adc temperature to celsius degree temperature
return temp_in_celsius;
}
void DisplayCurrentTemp(float temp_in_celsius)
{
char temp_in_string[10];
WordToStr(temp_in_celsius, temp_in_string);  //Converts input word to a string. The output string has fixed width of 6 characters including null character at the end (string termination). The output string is right justified and the remaining positions on the left (if any) are filled with blanks
Lcd_Out(1,8,temp_in_string);
}

void  SetTempLimits()          //setting the temperature limits
{
char high_temperature_limit_string[2];
char low_temperature_limit_string[2];
if (RB3_bit==1)
{
```

```
high_temperature_limit --;
}
if (RB4_bit==1)
{
high_temperature_limit++;
}
if(RB0_bit==1)
{
low_temperature_limit++;
}
if(RB1_bit==1)
{
low_temperature_limit--;
}
ShortToStr(high_temperature_limit,high_temperature_limit_string);
```

//onverts input signed short number to a string. The output string has fixed width of 5 characters including null character at the end (string termination). The output string is right justified and remaining positions on the left (if any) are filled with blanks.

```
 Lcd_Out(2,4,high_temperature_limit_string);
 Lcd_Out(2,4,"M:");
 ShortToStr(low_temperature_limit,low_temperature_limit_string);
 Lcd_Out(2,13,low_temperature_limit_string);
 Lcd_Out(2,13,":");
 }
void main() {              //main function
 InitSystem();              //Inilialize the UnitSystem in the main void loop
InitHeater();             //Initialize the Heater
InitLCD();                 //Inilialize the LCD
InitAdc();                //Initialize the analog to digital converter
```

```
InitThresholds();          //Initialize the thresholds
 while(1)                  //open while loop
 {
 float temp_measure=GetTemp();
 DisplayCurrentTemp(temp_measure);              //Display the current measured temperature on the lcd
 SetTempLimits();                        //Set the temperature limits

 Delay_ms(1500);                  // Wait 1.5second

  if                 (((char)temp>=high_temperature_limit)                 || ((char)temp<=low_temperature_limit))
{
RC1_bit=1;
RC2_bit=1;
}
else
{
RC1_bit=0;
RC2_bit=0;
}
}
{
if                 (((char)temp>=high_temperature_limit)                 || ((char)temp<=low_temperature_limit))
{
RC1_bit=1;
RC2_bit=1;
}
else
{
```

```
RC1_bit=0;
RC2_bit=0;
}
}
 }
 }
```

Used libraries

- **ADC_LIBRARY** : ADC (Analog to Digital Converter) module is available with a number of PIC MCU modules. ADC is an electronic circuit that converts continuous signals to discrete digital numbers. ADC Library provides you a comfortable work with the module.

- **Conversion:**The mikroC PRO for PIC Conversions Library provides routines for numerals to strings and BCD/decimal conversions

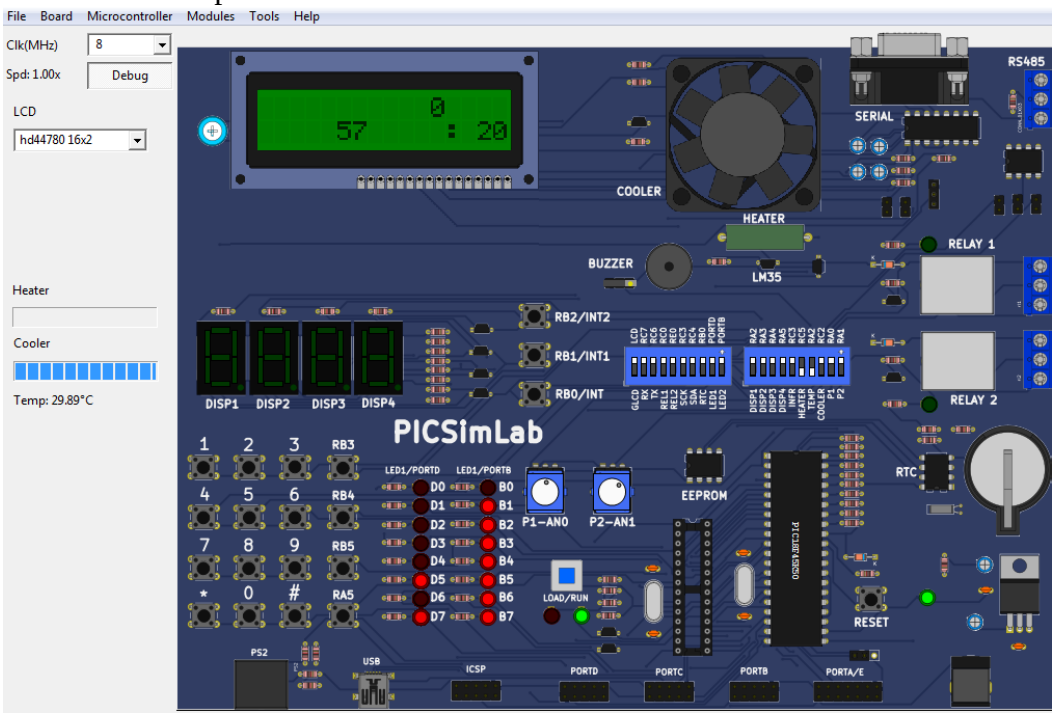
.

- **LCD:** The mikroC PRO for PIC provides a library for communication with Lcds .

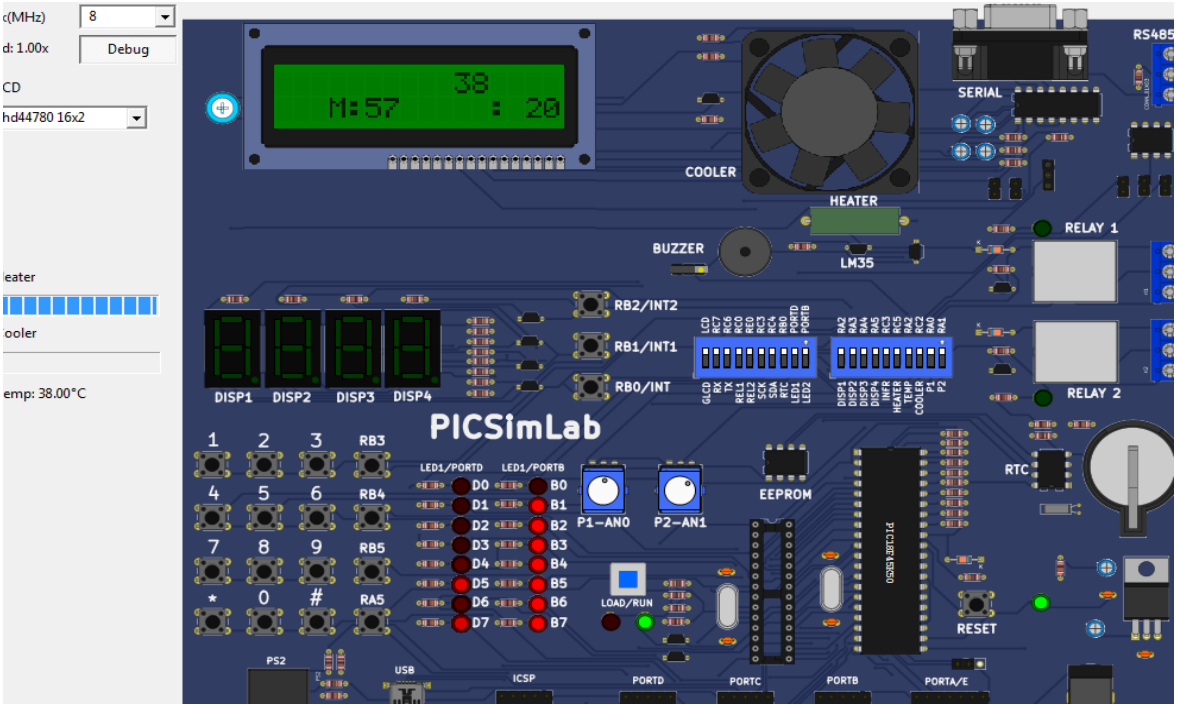
Mina Rizk
CU1901039
12/13/2021

II. Software Test cases

- First case:When the temperature reach the maximal stetted value of (high_temperature_limit), the buzzer functions and the cooler turned on in order to reduce the value of temperature



Second case: When the temperature reach the minima lstetted value of (low_temperature_limit), the buzzer functions and the heater turned on in order to elevate value of temperature.



Mina Rizk
CU1901039
12/13/2021

References

- <https://download.mikroe.com/documents/compilers/mikroc/pic/help/uart_remappable_library.htm>

- <https://www.google.com/search?q=adc+library+easy+pic+definition&sxsrf=AOaemvJg4VCYUtCuOwRo-BvjTEzTXOllPg:1637776390192&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjQn9

- <https://www.verical.com/datasheet/mikroelektronika-d-o-o-embedded-system-development-tools-mikroe-798-3249897.pdf