

Faculty of Engineering & technology

Electrical & computer systems Eng department

Simulation laboratory

Report for exp#4

**Introduction to MicroC program**

Student name :Mahdi alsalem

Student ID: 1121598

Instructors : dr.Jamal seyam & Eng.Mutasem omar

Date : 12th mar2016

**Table of contents**

|  |
| --- |
| Abstract ………………………………………………………………………………..2 |
| Introduction ………………………………………………………………………….2-3 |
| Methods and materials ……………………………………………………….3 |
| Procedure and results ……………………………………………………….4-12  |
| Discussion …………………………………………………………………………12-13 |
| Conclusion ……………………………………………………………………………13 |
| References ……………………………………………………………………………13 |
| Appendix A …………………………………………………………………………14 |

**Abstract**

The main objective of this experiment is to become familiar with MicroC Pro and be able to develop PIC based programs

|  |
| --- |
| Pic microcontrollers , are electronic circuits that can be programmed to carry out a vast range of tasks . timers , control a production line and more . and they are found in most electronic devices .  |
|  |

MicroC program for PIC is a full featured ANSI for PIC devices . precisely , its used for developing code using C language for pic microcontrollers.

Proteus is a program that combines mixed mode spice circuit simulation , animated components and microprocessor models to facilitate simulation of microcontroller based design .

In this experiment we use the microC program to build codes for some operations on PICS and then we interface this codes with a schematic captures designed on Proteus program to apply some the operations on the related PICS .

**Introduction**

**PIC Microcontrollers**

Microchip PIC microcontrollers are a family of chips of different sizes, shapes and abilities, each PIC mainly consists of a processor, memory and I/O ports, where some PIC’s may contain analog to digital conversion modules, serial communication modules, pulse width modulation modules and other features

PIC microcontrollers are somehow easy to program and use compared to other controllers The fact that they’ve a built in memory and I/O ports makes it easier for the user to use since there is no need to interface any memory or I/O ports externally like the case of the Intel 8086 chips. Like other controllers, the PIC has its own instruction set and can be programmed using an assembly language of its own or using higher level languages like the C language where special compilers are needed

In our experiment we’ll be using two different controllers for two different applications.



Fig2.1 PIC played by a computer software .

**MicroC program**

MicroC Pro is a computer software that is used to develop applications that run on Microchip PIC microcontrollers, the software runs on windows operating systems. The MicroC Pro with its graphical interface can be considered as an editor that enables the user to write his code using C language, then, compiles it using the proper compiling tools and generates a hexadecimal file containing the machine language that can be downloaded and run on the microcontroller.

**Methods and materials**

• computer

• microC program version 6.4.0

• Proteus 8 program

• microship PIC microcontrollers

**Procedure and results**

**A. Interfacing a PIC microcontroller with an LCD**

1.the code below was written in the microC program

// Lcd pinout settings

sbit LCD\_RS at RB4\_bit;

sbit LCD\_EN at RB5\_bit;

sbit LCD\_D7 at RB3\_bit;

sbit LCD\_D6 at RB2\_bit;

sbit LCD\_D5 at RB1\_bit;

sbit LCD\_D4 at RB0\_bit;

// Pin direction

sbit LCD\_RS\_Direction at TRISB4\_bit;

sbit LCD\_EN\_Direction at TRISB5\_bit;

sbit LCD\_D7\_Direction at TRISB3\_bit;

sbit LCD\_D6\_Direction at TRISB2\_bit;

sbit LCD\_D5\_Direction at TRISB1\_bit;

sbit LCD\_D4\_Direction at TRISB0\_bit;

void main()

{

TRISA = 0x00; // set all pins of port A as output

TRISB = 0x00; // set all pins of port B as output

while(1)

{ lcd\_init(); // initialize the lcd

lcd\_out(1,1,"ENEE413 EXP#3"); // print this message

Lcd\_Cmd(\_LCD\_CURSOR\_OFF;)

}

}

2.the code was built in the microC pro using the build menu and then was saved as hex file for the purposes of interfacing between microC and Proteus .

3.the circuit in figure 3.1 was drawn on the Proteus :



Fig3.1 the circuit in Proteus

4.the hex file that saved for the code in the microC was loaded on the pic 16F84A and the frequency was set to be 4 MHz which is the same as frequency in the code on microC and then the simulation was runned in Proteus and the statement ( ENEE413 EXP#3) was displayed on the LCD .

5. a time delay of 1000ms was added to the circuit using the command **[delay\_ms(1000)]** for some aims that will discussed in the discussion part of this report

The code with the time delay will be :

// Lcd pinout settings

sbit LCD\_RS at RB4\_bit;

sbit LCD\_EN at RB5\_bit;

sbit LCD\_D7 at RB3\_bit;

sbit LCD\_D6 at RB2\_bit;

sbit LCD\_D5 at RB1\_bit;

sbit LCD\_D4 at RB0\_bit;

// Pin direction

sbit LCD\_RS\_Direction at TRISB4\_bit;

sbit LCD\_EN\_Direction at TRISB5\_bit;

sbit LCD\_D7\_Direction at TRISB3\_bit;

sbit LCD\_D6\_Direction at TRISB2\_bit;

sbit LCD\_D5\_Direction at TRISB1\_bit;

sbit LCD\_D4\_Direction at TRISB0\_bit;

void main()

{

TRISA = 0x00; // set all pins of port A as output

TRISB = 0x00; // set all pins of port B as output

while(1)

{ lcd\_init(); // initialize the lcd

lcd\_out(1,1,"ENEE413 EXP#3"); // print this message

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

Delay\_ms(1000);

}

}

After adding time delay the statement was displayed on the LCD in Proteus again as shown in figure3.2



Fig3.2 circuit with time delay

**B.class work : displaying my name and ID**

1.The code was edited to be as follow :



Fig3.3 code to display name and ID number

2.the code was built again and saved as hex file .

3.the hex file was loaded to the PIC 16F84A on Proteus and the LCD displayed the name and ID as shown in figure 3.4



Fig3.4 name and ID # displayed on the LCD

**C. Using the A/D module**

In this part the PIC 16F877A was used .

1.the following code was written and built in microC



Fig3.5 code of converting analog number to a digital number using the PIC 16F877

2.the circuit in the following figure was drawn in the Proteus program but using the color's LEDs



Fig3.6 A/D converter circuit in Proteus .

3.the hex file that saved for the code in fig3.5 was loaded on the PIC 16F877A , the frequency was changed be 4 MHz , the simulation was runned

4. the potentiometer was used to change the input analog number and the result is changing the diodes was lighted on at the output of the PIC which represents the digital number converted from the analog

a) when the potentiometer is in the highest point (5V) all LEDs will be lighted on



Fig3.7 representation of the highest point at the potentiometer

b) when the potentiometer is in the lowest point (ground) all LEDs will be lighted off



Fig3.8 representation of the lowest point at potentiometer

c)at any point between the above 2 points on the potentiometer (0-5 V) some LEDs will be lighted on and some will be lighted off



Fig3.9 representation of a point on the middle of the potentiometer

**Discussion**

**LCD**

-The potentiometer is used to adjust the brightness of the LCD .

-The LCD in this experiment is a 16-bit so the maximum statement could displayed on it , contains 16 letter

-LCD\_out (x,y,"statement"); this is the code for the statement which will displayed on the LCD , y represents the number of digits before display the first letter (colom) and x represents the line at which the statement will displayed . (Row)

-The command while(1) is used for making an infinite loop until the power is disconnected from the circuit .

-The RW bin is connected to the ground for enabling the write operation . (data sheet )

-The time delay is used because LCDs take a long time to wake up and Proteus does not simulate this properly

**AD converter**

-The A/D converter read an input voltage in analog and convert it to a binary number as an output which is limited by 10 bits by the code .

-The potentiometer is used for changing the input voltage and so the binary representation which is represented by the diodes will also be changed.

-The binary number changes from 0000 ( 0 volt) to 1023 ( 5 volts)

**Conclusion**

In this experiment we begin using the microC program for programming pic microcontrollers and we learn how to make interfacing between Proteus program and the microC . how to apply the code written in microC to the circuit that designed in Proteus .

**References**

<http://www.mikroe.com/mikroc/pic>

<http://www.technologystudent.com/pics/picgen1.html>

lab manual

**appendix** A

A.1 data sheet of the LM016L LCD 16\*2 bit



 A.2 data sheet for PIC 16F84A

