****

**Electrical Engineering Department**

**ENEE4104**

**ENGINEERING SIMULATION LAB**

**Experiment #3 & #4 Report**

**PCB Design & Implementation and *Introduction to MicroC Program***

**Prepared By: Ahmad Zahran**

**Student Number: 1142583**

**Instructor: Dr. Jamal Seyam**

**Date: 2-11-2017**

**Section: 2**

**Abstract:**

These experiments were performed on Proteus software and MicroC Program. The aims of these experiments were to introduce the Printed Circuit Board (PCB) technology, learn how using Proteus software, be familiar with the Gerber files, implement the designed board using the acid etching method and to become familiar with MicroC Pro and be able to develop PIC based programs.

**Contents**

[Introduction: 1](#_Toc496455842)

[Methods and Materials: 2](#_Toc496455843)

[Procedure: 3](#_Toc496455844)

[3.1: PCB Design & Implementation 3](#_Toc496455845)

[3.2: Introduction to MicroC Program 4](#_Toc496455846)

[Results: 6](#_Toc496455847)

[4.1: PCB Design & Implementation 6](#_Toc496455848)

[4.2: Introduction to MicroC Program 9](#_Toc496455849)

[Discussion: 12](#_Toc496455850)

[Conclusion: 13](#_Toc496455851)

[References: 14](#_Toc496455852)

[Appendices: 15](#_Toc496455853)

**Table of Figures:**

[Figure 1: the voltage regulator circuit 3](#_Toc496455789)

[Figure 2:circuit 1 16X2 LCD with 16F84A PIC 4](#_Toc496455790)

[Figure 3: circuit 2 the 16F877A PIC microcontroller (analog to digital converter) 5](#_Toc496455791)

[Figure 4: the voltage regulator circuit in PCB 6](#_Toc496455792)

[Figure 5:The routed board 6](#_Toc496455793)

[Figure 6: The 3-D view of the circuit board 7](#_Toc496455794)

[Figure 7:checked if there were any errors in the designed PCB layout 7](#_Toc496455795)

[Figure 8:The required board by the CNC machine 8](#_Toc496455796)

[Figure 9:The output after loaded CODE1 and simulated the circuit in figure 2 9](#_Toc496455797)

[Figure 10:The output after loaded CODE2 and simulated the circuit in figure 2 9](#_Toc496455798)

[Figure 11:The output of circuit 2 (analog to digital converter) at 36% of RV1 10](#_Toc496455799)

[Figure 12:The output of circuit 2 (analog to digital converter) at 48% of RV1 10](#_Toc496455800)

[Figure 13:The output of circuit 2 (analog to digital converter) at 82% of RV1. 11](#_Toc496455801)

[Figure 14:The output of circuit 2 (analog to digital converter) at 99% of RV1. 11](#_Toc496455802)

# **Introduction:**

The aims of these experiments, first, to learn how to use Proteus software and be familiar with the Printed Circuit Board (PCB) technology to generate Gerber file and install it into the CNC machine to implement voltage regulator circuit on a board and print it by CNC machine. Second, to become familiar with MicroC Program to build codes and create hex file for some operations on PICS, and to simulate the PIC microcontroller in Proteus to test codes and verify PIC operation before downloading it on a real PIC.

**Theoretical Background**

**Proteus software**

Proteus combines ease of use with powerful features to help you design, test and layout professional PCBs like never before. With nearly 800 microcontroller variants ready for simulation straight from the schematic, one of the most intuitive professional PCB layout packages on the market and a world class shape based autoroute included as standard, Proteus Design delivers the complete software package for engineers.[1]

**PCB technology**

Proteus PCB design seamlessly combines Schematic Capture and PCB layout to provide a powerful, integrated and easy to use suite of tools for professional PCB Design. [1]

**PIC Microcontrollers**

PIC microcontrollers (Programmable Interface Controllers), are electronic circuits that can be programmed to carry out a vast range of tasks. They can be programmed to be timers or to control a production line and much more. They are found in most electronic devices such as alarm systems, computer control systems, phones, in fact almost any electronic device. Many types of PIC microcontrollers exist, although the best is probably found in the GENIE range of programmable microcontrollers. These are programmed and simulated by Circuit Wizard software. [2]

**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:**

* Proteus 8 Professional program
* CNC machine
* MicroC Program

.

# **Procedure:**

## **3.1: PCB Design & Implementation**

* **Part 1:****Starting Proteus & Drawing the circuit**
1. Proteus professional software was started and schematic page and PCP layout page were created.
2. The circuit shown in figure 1 was drawn. The components that had a PCB footprint were added from libraries.



Figure 1: the voltage regulator circuit

* **Part 2: Place the Parts**
1. From the PCB layout page, the parts were placed.
2. To routing the PCB, the routing layers was determined, the single layer PCB was made from (Technology *>> Design Rule Manager >> Net Classes).*
3. from *Track Mode,* the width was adjusted to75th.
4. The board was routed from (Tools *>> Auto Router >> Begin Routing). Then* the routed board was appeared.
5. The 3-D view of the circuit board was created.
* **Part 3:****Gerber files & CNC machine**
1. The design was checked if there were any errors associated with the designed PCB layout from (Output >> Pre-Production Check).
2. The Gerber file was generated from (Output >> Gerber/Excellon Output).
3. the Gerber file was loaded into the program of the CNC machine, then the required board was drilled and cut by the CNC machine.

## **3.2: Introduction to MicroC Program**

* **Part 1: Interfacing a PIC microcontroller with an LCD**
1. The MicroC Pro was started and the code in Appendix A.1 that controls a 16X2 LCD using a 16F84A PIC was written and built, then the hex file was created.
2. The Protues 8 Professional program was opened and the circuit given was drawn as shown in figure 2.



Figure 2:circuit 1 16X2 LCD with 16F84A PIC

1. The Hex file for CODE1 was loaded to the 16F84A PIC, and the clock frequency was changed to 4MHz, then the circuit was simulated and the output was seen.
* **Part 2: Class Exercise**

The previous code was changed to display my first name and ID number simultaneously with each on separate lines and the output was seen. the code shown in Appendix A.2.

* **Part 3: Using the A/D module**

In this part the 16F877A PIC microcontroller was used to build analog to digital converter.

1. New MicroC project was created and the code in Appendix A.3 was written and built, then the hex file was created.
2. The circuit given was drawn in Protues as shown in figure 3.



Figure 3: circuit 2 the 16F877A PIC microcontroller (analog to digital converter)

1. The Hex file for CODE3 was loaded to the 16F877A PIC, and the clock frequency was changed to 4MHz, then the circuit was simulated and the output was seen on variable value of RV1.

# **Results:**

## **4.1: PCB Design & Implementation**

* **Part 1: Place the Parts**

The voltage regulator circuit in PCB:



Figure 4: the voltage regulator circuit in PCB

The routed board:



Figure 5:The routed board

The 3-D view of the circuit board:



Figure 6: The 3-D view of the circuit board

* **Part 2:****Gerber files & CNC machine**

The design was checkedand there were no errors associated with the designed PCB layout as shown in the figure.



Figure 7:checked if there were any errors in the designed PCB layout

The required board by the CNC machine:



Figure 8:The required board by the CNC machine

## **4.2:** **Introduction to MicroC Program**

* **Part 1: Interfacing a PIC microcontroller with an LCD**

The output after loaded CODE1 and simulated the circuit in figure 2:



Figure 9:The output after loaded CODE1 and simulated the circuit in figure 2

* **Part 2: Class Exercise**

The output after loaded CODE2 and simulated the circuit in figure 2:



Figure 10:The output after loaded CODE2 and simulated the circuit in figure 2

* **Part 3: Using the A/D module**

The output of circuit 2 (analog to digital converter) at 36% of RV1.



Figure 11:The output of circuit 2 (analog to digital converter) at 36% of RV1

The output of circuit 2 (analog to digital converter) at 48% of RV1.



Figure 12:The output of circuit 2 (analog to digital converter) at 48% of RV1

The output of circuit 2 (analog to digital converter) at 82% of RV1.



Figure 13:The output of circuit 2 (analog to digital converter) at 82% of RV1.

The output of circuit 2 (analog to digital converter) at 99% of RV1.



Figure 14:The output of circuit 2 (analog to digital converter) at 99% of RV1.

# **Discussion:**

* **Part 1: PCB Design & Implementation**

In this part, voltage regulator circuit as shown in figure1 was design and implemented. The results obtained fairly acceptable. After the PCB design was done and the output was seen, there were gaps between the components, where we had to reduce the distances between them and exploit the smallest possible space, because larger area means more copper required, so the cost will be increase.

 Also, from the 3-D view of the circuit board as shown in figure 6 we can sure that’s the components distributed well and use the board space in the best possible way. Then after the Gerber file built, it was loaded into the program of the CNC machine, then we were seen the required board was drilled and cut by the CNC machine shown in figure 8.

* **Part 2: Introduction to MicroC Program**

In this part, the protues and microC were used to control 16X2 LCD using 16F84A PIC and to build analog to digital converter using 16F877A PIC microcontroller. As we see in figure 9 &10, the output was seen on the LCD, where in the part1, the code1 was used directly but, the command (delay\_ms(1000); ) was used to make time delay of 1000ms to the circuit to see the output of LCD statically. In part2, the previous code was change to display my name and ID number simultaneously with each on separate lines, and to do this the commands (lcd\_out(2,1,"1142583"); ) & ( lcd\_out(1,1,"Ahmad Zahran"); ) just added that made my name on the upper line and the ID number on the lower line.

In the part 3, the output the analog to digital converter of was seen in variable value of RV. This converter read input voltage in analog and convert it to a binary number which limited by 10 bits that represented by the diodes. The potentiometer(RV) used for changing the input voltage. And as we see in figure 12,13&14, when RV in the lowest point, all LEDs lighted off, and when it in the highest point, all LEDs lighted on, and at any point between them, some LEDs lighted on and some lighted off depending on value of the input voltage.

# **Conclusion:**

The aims of these experiments were done successfully, where in the first experiment we learned how to use Proteus program in plots circuits and simulation it, and in design a printed circuit board(PCB). Also, we shown how to use CNC machine to implement the design on the real world. But in my design, there was a large space between the components in the board. In the second experiment the microC program was used for programming PIC microcontrollers and interfaced between them. We see how microC easy and powerful for PIC.

# **References:**

[1] <https://www.labcenter.com/>

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

[3] Lab Manual.

# **Appendices:**

**Appendix A.1: Code to control 16X2 LCD using 16F84A PIC**

**CODE 1:**

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);

}

}

**Appendix A.2: Code for Class Exercise**

**CODE 2:**

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,"Ahmad Zahran"); // print this message

lcd\_out(2,1,"1142583");

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

delay\_ms(1000);

}

}

**Appendix A.3: Code for Class Exercise**

**CODE 3:**

unsigned int a;

void main() {

TRISA = 0xFF; // PORTA is input

TRISC = 0; // PORTC is output

TRISB = 0; // PORTB is output

do {

a = ADC\_Read(0); // Get 10-bit results of AD conversion

 PORTB = a; // Send lower 8 bits to PORTB

PORTC = a >> 8; // Send 2 most significant bits to RC1, RC0

}

while(1);

}