

**NORTH SOUTH UNIVERSITY**

**Project Report**

**Microprocessor Interfacing & Embedded Systems**

**CSE-331**

**Section: 01**

**Submitted by:**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Email** |
| **Abrar Yasir** | **1530069042** | **abrar.yasir@northsouth.edu** |
| **Muhtasim Riffat** | **1511043042** | **muhtasim.riffat@northsouth.edu** |
| **Saifa Mohsin** | **1510960642** | **saifa.nmo12@gmail.com** |

**Submitted to:**

**Mr. Rishad Arfin**

**Lecturer, Department of Electrical and Computer Engineering**

**Date of Submission: 03.09.2019**

**Project Title: Adjustable blinking LED’s with PIC Microcontroller**

**Objective:**

We are building a circuit using PIC16F877A Microcontroller which will be used to control the variation of the blinking of LED lights. The variation of the lights will be controlled by the computer code that will be entered into the microcontroller. Through the adjustment of the computer code the microcontroller can vary the rates of the blinking of the lights individually or it can be programmed for all to blink at the same rate.

**Probable Application:**

* Strobe lights
* LED indicators

**Block Diagram:**

*PIC16F877A Microcontroller*

LED

Crystal Oscillator

9V Battery

**Components:**

1. PIC 16F877A microcontroller
2. LED’s
3. Resistors (330 ohm, 10kohm)
4. Breadboard
5. LM 7805 transistor
6. Crystal Oscillator (3.57 MHz)
7. Jumping wires
8. 9V battery
9. PICKIT 3
10. ZIF socket adapter
11. Battery connector

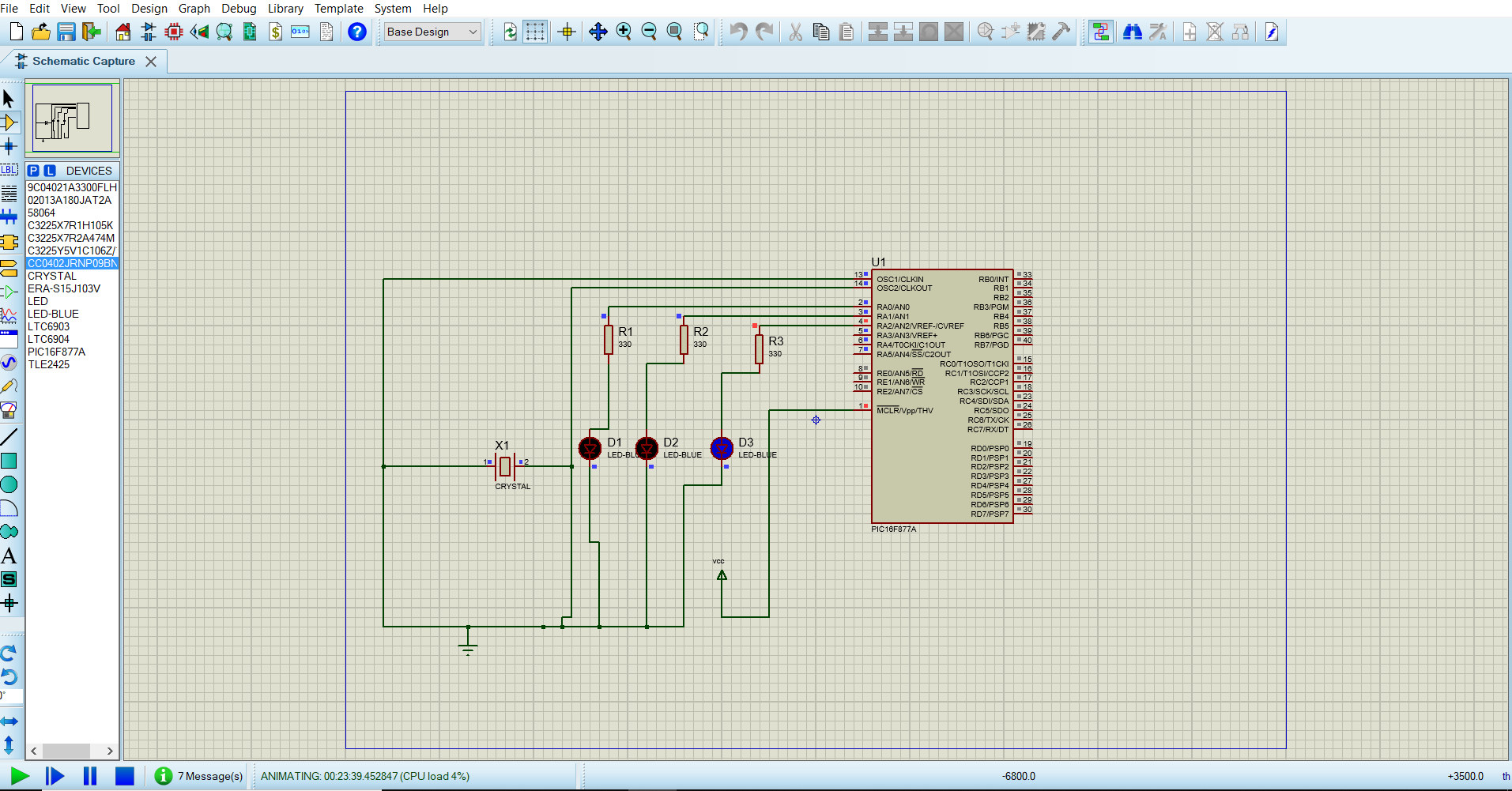
**Estimated Cost:**

|  |  |
| --- | --- |
| Components Name | Cost (BDT) |
| Breadboard | 84.9 |
| 16F877A MCU | 200 |
| Resistor (330 ohm) x3 | 4 |
| Resistor (10kohm) | 1.25 |
| Crystal Oscillator | 7.9 |
| Jumping wires | 20 |
| LM 7805 | 5 |
| 9V Battery | 30 |
| PIC KIT 3 | 1669 |
| Adapter | 269 |
| Battery connector | 15 |
| LED (x3) | 6 |
| Total | 2312 |

**Working Procedure:**

1. First, we wrote our code in C language using microchip programming tool.
2. Then we generated the hex file to install the program in the micro controller.
3. Before implementing with the hardware, we used the virtual circuit building tool “Proteus”, to draw our circuit and simulate it.
4. After drawing the circuit, we made the necessary connections between the components, we imported the hex file to Proteus and ran the simulation to check if the given instructions (code) were working properly.
5. Before hardware implementation, we burned the program into the PIC 16F877A using MCU adapter and pick kit 3.
6. After building the circuit properly, we provided the power using 9V battery, the 7508 transistor converted it to 5V and noticed the LED’s were blinking.

**Figure from Simulation Tool:**



**Interfacing Language from Simulation Tool:**

void main() {

TRISA = 0x00;

while (1)

{

PORTA.F0 = 1;

Delay\_ms(500) ;

PORTA.F0 = 0;

PORTA.F1 = 1;

Delay\_ms(400) ;

PORTA.F1 = 0;

PORTA.F2 = 1;

Delay\_ms(300) ;

PORTA.F2 = 0;

}

**Discussion:**

Our original plan was to create a stopwatch circuit using the 8051 microcontroller. We collected all the necessary components for our planned circuit and we started building the circuit. After completing the basic circuit we started to write the program code in C for the microcontroller. Upon completion of the code we had to install the code into the microcontroller and for that we required the 8051 MCU burner. We searched for the burner but could not obtain it from any local sources. After some time we realized that the 8051 MCU burner was not readily available and we were not aware about this circumstance. After much effort, since we could not obtain the burner we decided to change our project from a stopwatch using 8051 microcontroller to building an adjustable blinking LED circuit using PIC16F877A microcontroller. Due to our time constraints and the late issue of not having found the burner for the 8051 MCU we had to change the project to something simple in order to meet our deadline.

We hastily collected the required components for the PIC16F877A microcontroller project and started building our basic circuit. After completion of the circuit we moved onto the simulation part using Proteus Simulation Tool. We created the circuit in Proteus Simulation Tool then moved onto writing the code for the microcontroller. We wrote the code in C language using the Mikrochip microC programming tool. We built the code to generate to the hex file. We uploaded the hex code to our simulation in Proteus Simulation Tool and ran the simulation. The simulation ran perfectly and we found the output we were looking for. We connected the PICkit 3 to the MCU and connected that with the computer. Using the PICkit 3 software we checked the connection and the power input was 5V. We found the connection to be good and imported the hex file to PICkit 3 software and after importing we wrote the hex code into the MCU. Then we got a message that the program was written successfully to the MCU. We then built the circuit by putting all the components in the breadboard together and connected all the necessary wirings and power sources. We connected the 9V battery and found the circuit to be working as expected from our simulation. We used the 7508 Transistor to modulate the power from 9V to 5V. We found the LED’s to be blinking at the rate that we programmed it to blink in our code. Since the circuit was working as intended we increased the number of LED’s to 3 and programmed different blinking rates for the 3 LED’s in our program code. Then we simulated the new circuit and also found our circuit to be working as we intended.

Throughout building our project we faced many different types of problems. We at first used MPLAB X IDE to generate our hex code. But there was an issue with the compilation process and it would not generate. We’re getting an issue with configuration bits when we tried to burn it into the microcontroller using the PICkit. The first PICkit that we collected was not connecting properly because it was not receiving the necessary voltage. We required 5V but the PICkit was only receiving 4.75V. Since it was not receiving the correct voltage it was also telling us that there was no hex code in the programming tool. Since this PICkit was not working we acquired another PICkit and tried the same process again. This new PICkit was working and it connected properly. But this time there was an issue with the MPLAB X IDE where we still were unable to burn the code using the PICkit. After facing this issue we decided to try another program Mikrochip microC programming tool. We wrote the code again using the new program. We built the code and the build was successful. Then we used the basic PICkit 3 programmer to burn the program in our microcontroller. This time the PICkit successfully connected with the PICkit 3 programmer and it was receiving the correct voltage(5V). We then uploaded the program using the PICkit 3 programmer to the microcontroller. The programming was successful according to the PICkit 3 programmer. We then disconnected the MCU from the burner and proceeded to connect the MCU with our hardware. Finally, we connected the 9V battery and found our circuit to be working according to our simulation in the Proteus Simulation Tool and the LED’s were blinking at the rate that we programmed them.