Lab 7: Basic Introduction to Microcontrollers

Kaysi Gutierrez and Juan Cantu

EECE 3225

04.28.2023

ABSTRACT

This lab aimed to enhance understanding of microcontrollers by exploring their structure, operation, and programming using assembly language. The objectives included observing the effects of various C statements on microcontroller behavior and measuring different microcontroller parameters. The lab was divided into three parts, each focusing on a specific project. Part 1 involved writing a program to generate a pulse train with a specific width and period on a designated pin. This project utilized NOP() instructions to control the timing and achieve the desired pulse characteristics.

Part 2 focused on configuring the microcontroller's ports to handle input and output operations. The program written for this project utilized PORTB as input and PORTD as output. It involved performing arithmetic operations on two 4-bit numbers (X1 and X2) received via specific pins and displaying the sum on a seven-segment display. Additionally, if the sum exceeded the hexadecimal value of F, a dash symbol was displayed. In Part 3, a MOSFET transistor was employed as a switch to control a motor. The circuit utilized Pin RD0 of the microcontroller to drive the MOSFET's gate. Two input pins (RB0 and RB1) denoted as AB determined the microcontroller's behavior. Specifically, four different states were programmed based on AB values: AB = 00 for motor off, AB = 11 for motor on, AB = 01 for the motor to alternate between on and off states every 2 seconds, and AB = 10 for generating a pulse train with a 50% high and 50% low duty cycle at a specific frequency.

Part 1

For this project, we had to write a program that produces a pulse train with the given width and period on the specified pin. Table 1, below shows the projects with their assigned group letter. For this lab, we were assigned group 'B'.

Group	Pulse Width	Period	Pin
A	4us	10us	RD0
B	5us	15us	RC2
C	7us	20us	RD1
D	4us	25us	RC3
E	5us	10us	RD2
F	7us	15us	RC4
G	4us	20us	RD3
H	5us	25us	RC5
I	9us	20us	RD4

Table 1. Project 1 Information

Code 1:

```
#include <xc.h>
#pragma config FOSC = INTOSC
#pragma config WDTE = OFF
#pragma config MCLRE = ON
#pragma config CP = OFF

void main(){
  ANSELC = 0b00000000;
  TRISC = 0b00000000;
  OSCCON = 0b01101010;
  while(1){
    PORTC = 0b00000000;
    NOP();NOP();NOP();NOP();NOP();NOP();NOP();
    PORTC = 0b01010101;
    NOP();
  }
}
```

This code generates a square wave with a pulse width of 5 microseconds and a period of 15 microseconds on pin RC2 of a microcontroller. The code sets registers to configure the microcontroller and enters an infinite loop to generate the square wave by repeatedly setting the appropriate output pins on Port C.

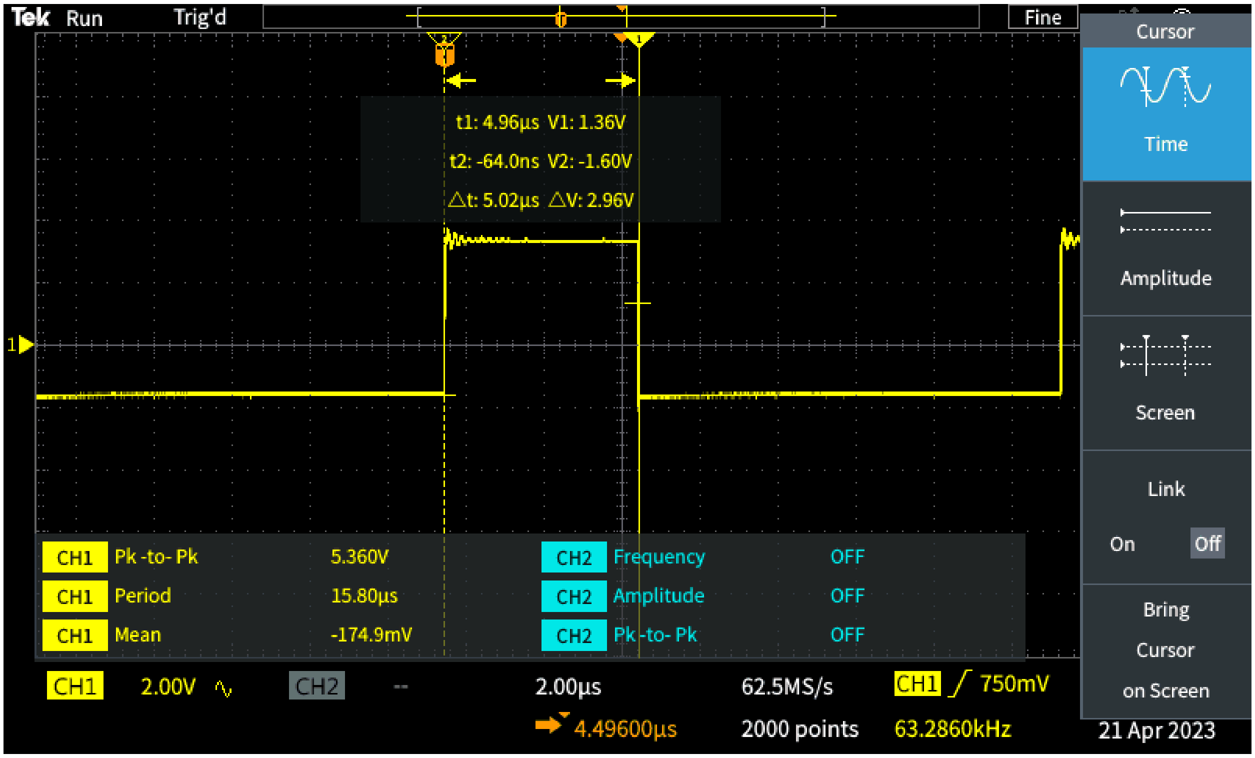


Figure 1. Oscilloscope Shot from Code 1 at Pin RC2

Part 2

For this project, we had to write a program that configures PORTB as an input and PORTD as an output. Pins RB3-RB0 represent a 4-bit number, X1 and pins RB7-RB4 represent a second 4-bit number, X2. The program should result in the sum of X1 and X2 and the result should be displayed on a seven-segment display as a HEX number, driven by PORTD through resistors. If the result is greater than hexadecimal F, a simple dash should be displayed.

Code 2:

```
 #include <xc.h>
#pragma config FOSC = INTOSC
#pragma config WDTE = OFF
#pragma config MCLRE = ON
#pragma config CP = OFF

void main() {
   ANSELD = 0b00000000;
   TRISD = 0b00000000;

   TRISB = 0b11111111;
   ANSELB = 0b00000000;
```

```
while (1) {
  // Read X1 and X2 from PORTB
  unsigned char x1 = PORTB & 0b00001111;
  unsigned char x2 = PORTB & 0b11110000;

  unsigned char sum = x1+(x2/16); // Calculate the sum of X1 and X2

  if (sum > 15) { // If the sum is greater than F, display a dash
    PORTD = 0b01000000;
  } else {
    switch (sum) {  // Convert the sum to a HEX number and display it on the seven-segment display
      case 0:
        PORTD = 0b10111111;
        break;
      case 1:
        PORTD = 0b10000110;
        break;
      case 2:
        PORTD = 0b11011011;
        break;
      case 3:
        PORTD = 0b11001111;
        break;
      case 4:
        PORTD = 0b111100110;
        break;
      case 5:
        PORTD = 0b11101101;
        break;
      case 6:
        PORTD = 0b11111101;
        break;
      case 7:
        PORTD = 0b10000111;
        break;
      case 8:
        PORTD = 0b11111111;
        break;
      case 9:
        PORTD = 0b11100111;
        break;
      case 10:
        PORTD = 0b01110111;
        break;
```

```
        case 11:
          PORTD = 0b01111100;
          break;
        case 12:
          PORTD = 0b10011100;
          break;
        case 13:
          PORTD = 0b01011110;
          break;
        case 14:
          PORTD = 0b01111001;
          break;
        case 15:
          PORTD = 0b01110001;
          break;
      }
    }
  }
}
```

The program configures PORTB as an input and PORTD as an output. It reads two 4-bit numbers (X1 and X2) from pins RB3-RB0 and RB7-RB4 respectively, calculates their sum, and displays the result on a seven-segment display driven by PORTD through resistors. If the sum is greater than hexadecimal F, a dash is displayed. The program uses a switch-case statement to convert the sum to a HEX number and display it on the seven-segment display. The program runs in an infinite loop, continuously reading the input and updating the output.

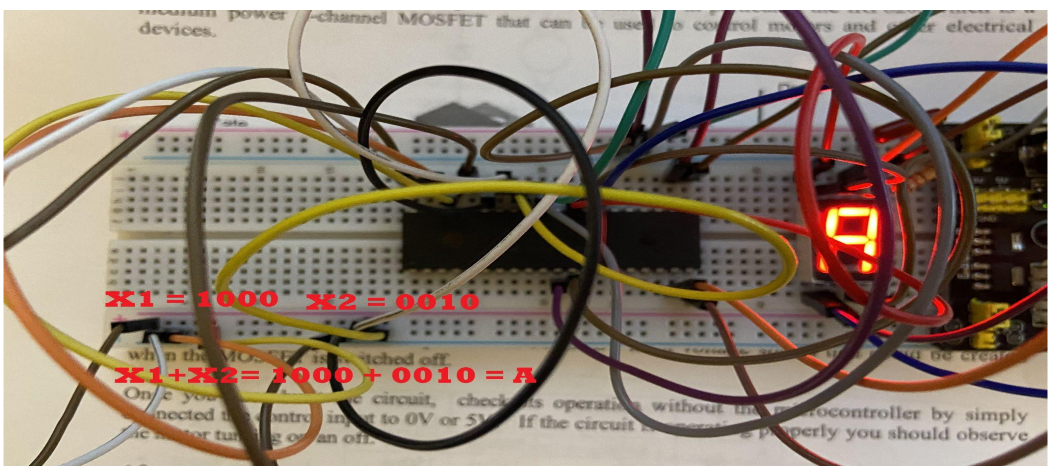


Figure 2. Successful Run of Code 2

In Figure 2, you can see a run of the program where x1=1000 and x2=0010 resulting in the output of 1000+0010 = A.

Part 3

For this project, we used a MOSFET transistor (IRF620) to control a motor. The MOSFET was used as a simple switch. The circuit we used is shown below in Figure 3.

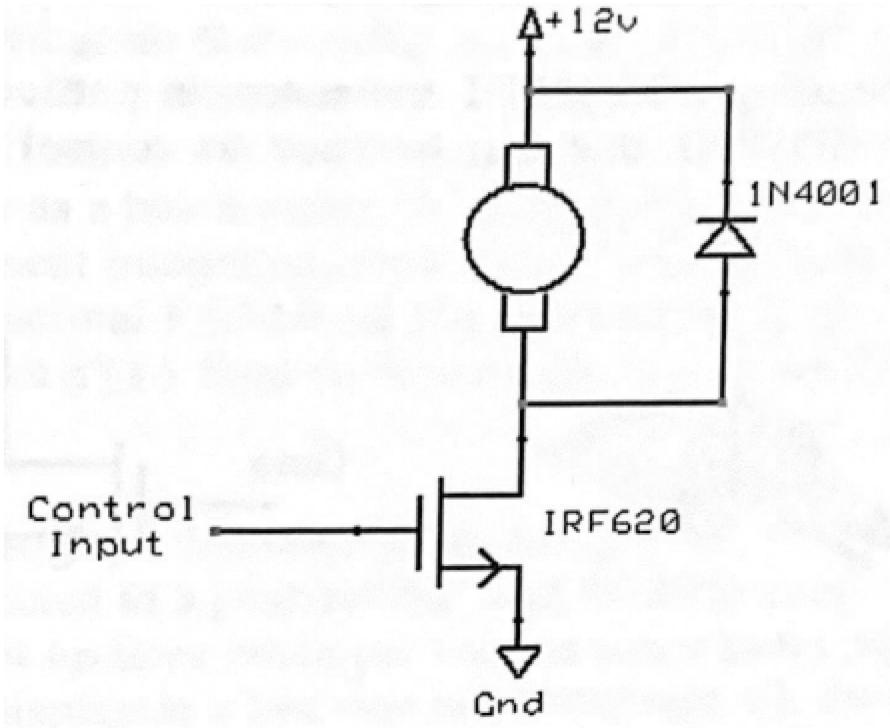


Figure 3. Circuit To Control DC Motor With MOSFET

The gate of the MOSFET was connected to Pin RD0 of the microcontroller. On the Microcontroller pins RB0 and RB1 are used as input pins, AB. Respective of input AB the microcontroller is programmed to have 4 different states:

AB = 00, the motor should be off.

AB = 11, the motor should be on.

AB = 01, the motor turns on and off at 2 second intervals.

AB = 10, pulse train 50% high and 50% low (assigned frequency shown below).

Group	Frequency	Time High	Time Low
A	1k Hz	500us	500us
B	2k Hz	250us	250us
C	2.5k Hz	200us	200us
D	4k Hz	125us	125us
E	5k Hz	100us	100us
F	1k Hz	500us	500us
G	2k Hz	250us	250us
H	2.5k Hz	200us	200us
I	4k Hz	125us	125us

Table 2. Frequency for Pulse Train

Code 3:

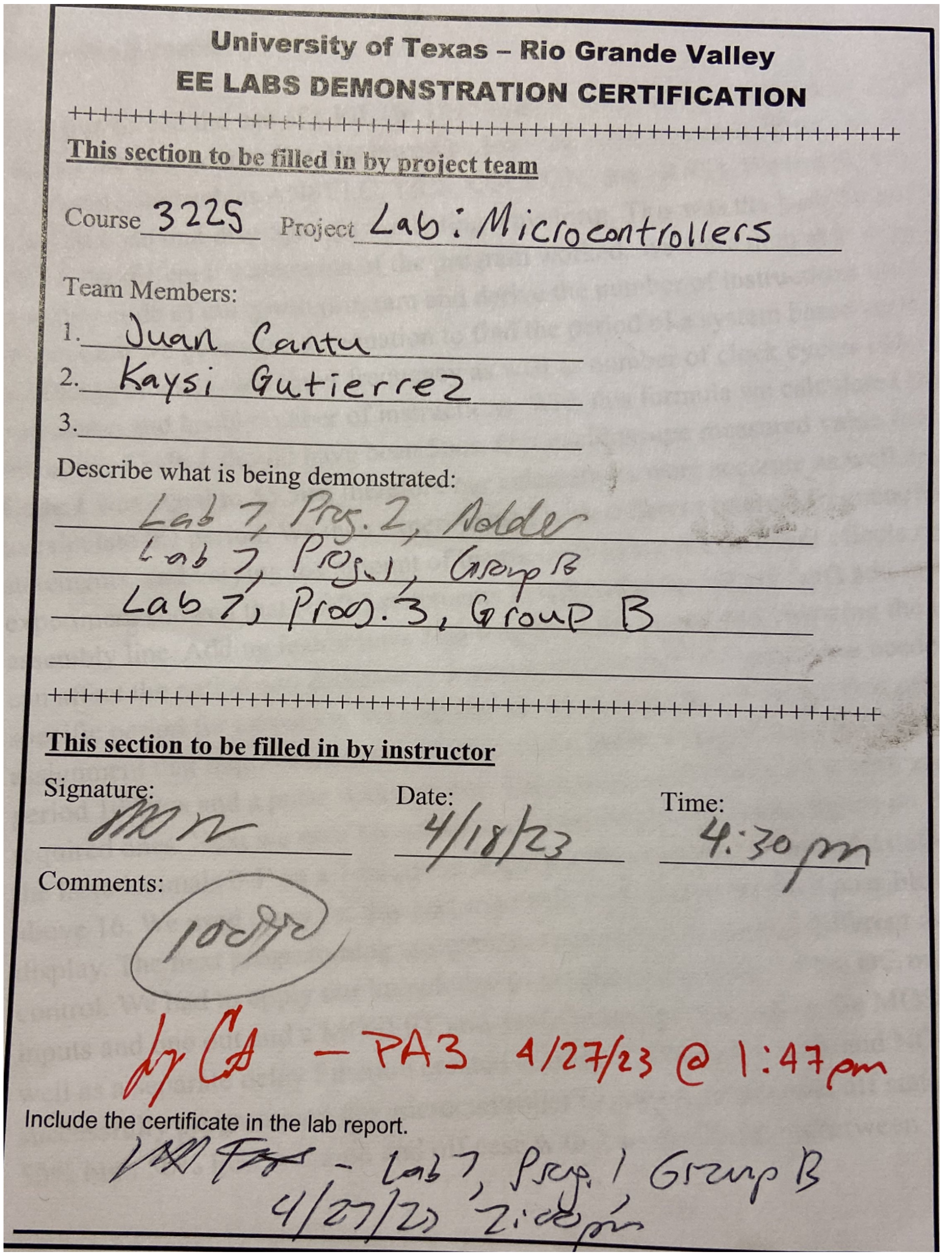
```
#include <xc.h>
#pragma config FOSC = INTOSC
#pragma config WDTE = OFF
#pragma config MCLRE = ON
#pragma config CP = OFF

void delay (int x){
    for (int i=0; i<x; i++){
        _delay(100);
    }
}

void main(){
    ANSELD = 0b00000000;
    TRISD = 0b00000000;

    TRISB = 0b00000011;
    ANSELB = 0b00000000;

    int x1;
    while(1){

        x1 = PORTB & 0b00000011;
        switch(x1){
            case 0: // 00
                PORTD = 0b00000000;
                break;
            case 1: // 01
                OSCCON = 0b00111010;
                PORTD = 0b00000001;
                delay(2000);
                PORTD = 0b00000000;
                delay(2000);
                break;
            case 2: // 10
                OSCCON = 0b00111010;
                PORTD = 0b00000001;
                NOP();NOP();NOP();NOP();
                NOP();NOP();NOP();NOP();
                NOP();NOP();NOP();NOP();
                NOP();NOP();NOP();NOP();
                NOP();NOP();NOP();NOP();
                NOP();NOP();NOP();NOP();
```

```
            NOP();NOP();NOP();NOP();
            NOP();NOP();
            PORTD = 0b00000000;
            break;
        case 3: // 11
            PORTD = 0b0000001;
            break;
        default: // 00
            PORTD = 0b00000000;
    }
  }
}
```

This program takes two inputs represented by pins RB0 and RB1. The output is on pin RD0. Case 0 is executed when both RB0 and RB1 are set to low voltage, resulting in the motor being off. In Case 1, when RB0 is high and RB1 is low, the motor turns on and off in 2-second intervals with the help of a user-defined function called 'delay' that uses the built-in '_delay()' method provided by MPLAB. Case 2 is achieved when RB0 is low and RB1 is high. It is similar to Case 1, but the motor pulses at a frequency of 2 kHz. The motor produces a 50% duty cycle with a high time of 250 microseconds and a low time of 250 microseconds, achieved using NOP() instructions. Case 3 is executed when both RB0 and RB1 are set to high voltage. This results in the motor being on.

Conclusions

In conclusion, this lab provided valuable insights into the structure, operation, and programming of microcontrollers. Through the three distinct projects, we were able to gain hands-on experience and expand knowledge in various aspects of microcontroller applications. Overall, this lab not only provided practical exposure to microcontroller programming but also emphasized the importance of understanding assembly language, circuit design, and parameter measurement. By observing the effects of different C statements and measuring microcontroller parameters, participants gained valuable insights into the behavior and performance of microcontrollers in different scenarios. By successfully completing these projects, participants developed a solid foundation in microcontroller programming, assembly language, and circuit integration. The skills and knowledge acquired in this lab will prove invaluable in future endeavors involving microcontrollers and embedded systems, enabling participants to design and implement more complex and sophisticated applications.



University of Texas – Rio Grande Valley
EE LABS DEMONSTRATION CERTIFICATION

++

This section to be filled in by project team

Course 3225 Project Lab: Microcontrollers

Team Members:

1. Juan Cantu
2. Kaysi Gutierrez
3.

Describe what is being demonstrated:

Lab 7, Prg. 2, Adder
Lab 7, Prog.1, Group B
Lab 7, Prog. 3, Group B

++

This section to be filled in by instructor

Signature: | Date: 4/18/23 | Time: 4:30pm

Comments:

100%

– PA3 4/27/23 @ 1:47pm

Include the certificate in the lab report.

– Lab 7, Prog. 1 Group B
4/27/23 2:00pm

Demonstration. Demos for 3 projects