EC – 310: Microprocessors and Microcontroller based Design

**LAB # 12: Programming PIC microcontroller using MPLAB and PicKit3 in C**

**Course Instructor:** Dr. Sagheer Khan

**Lab Instructor:** Ayesha Khanam

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Students’ Names** | **CMS ID** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LAB # 12: Programming PIC microcontroller using MPLAB and PicKit3 in C**

In this lab, students will explore the programming of the PIC18F452 microcontroller using C language. The lab emphasizes the use of MPLAB IDE for coding, alongside PicKit3 for programming the microcontroller.

**Lab Objectives**

Upon completion of this lab, students will be able to:

* Design microcontroller based circuits using PicKit3

**C Programming Overview**

C programming for PIC microcontrollers involves utilizing specific libraries and functions tailored to interact directly with the microcontroller. This allows for effective control of input and output operations, which is critical for embedded system development. In this lab we will use the C18 Compiler on the MPLAB IDE.

**C18 Compiler:**

The **C18 compiler** is a C language compiler used within the **MPLAB IDE** for programming PIC18 series microcontrollers. It allows writing C code that can be compiled into a hex file, which is then uploaded to the microcontroller. To access the C18 Compiler on the MPLAB IDE, select “C18 Compiler” instead of “MPASM Compiler” while setting up an MPLAB Project.

*(Referring to “MPLAB & Proteus Guide” available on LMS, follow the same procedure as for .asm project file except step 4 and save the source file with a .c extension instead of .asm)*

1. **Header File**

This directive includes the necessary definitions and function prototypes for working with the PIC microcontroller, providing access to built-in functions and register definitions.

1. **Conditional Statements**

Conditional statements in C are characterized by the “if” statement, which executes a block of code when a specified condition evaluates to true. The following example shows how a conditional statement is implemented in the context of programming microcontrollers.

if (PORTD < 100) {

LATB = PORTD + 50; // Modify and output value to LATB (PORTB output latch)

}

else {

LATB = 0x00; // Clear LATB if value is 100 or more

}

1. **Using Delay in C Programming**

Delays are essential for controlling the timing of events such as blinking LEDs, creating animations, or pausing for user input. In C, delays are implemented using loops that create a time delay by repeatedly executing a block of code.

Below is an example of how to create a simple delay using a loop. This code will make the LED blink with a delay.

void delay(unsigned int count) {

while (count > 0) {

count--;

}

}

1. **The Main Function**

Ensure that the main function is declared with a void return type, as C18 does not require a return statement in main().

void main(void) {

TRISB = 0x00; // Set PORTB as output

LATB = 0x00; // Clear LATB (PORTB output latch)

while (1) {

LATB = 0xFF; // Turn ON all LEDs on PORTB

delay(1000); // Delay

LATB = 0x00; // Turn OFF all LEDs on PORTB

delay(1000); // Delay

}

}

1. **Use of LATx**

The LATx register controls the output prompts, which is safer and recommended for writing operations.

#include <p18f452.h>

#define DELAY\_TIME 1000

void delay(unsigned int count) {

while (count > 0) {

count--;

}

}

void main(void) {

TRISB = 0x00; // Set PORTB as output

LATB = 0x00; // Clear PORTB output

while (1) {

LATB = 0xFF; // Turn ON all LEDs on PORTB

delay(DELAY\_TIME); // Delay

LATB = 0x00; // Turn OFF all LEDs on PORTB

delay(DELAY\_TIME); // Delay

}

}

***Note:*** *Refer to Lab # 08 - 11 resources on LMS if reinforcement be needed.*

**Lab Tasks**

1. Write a program to take input on PORTD and perform operations based on its value. If PORTD < 250, add 15 to the input and display the result on PORTB. If PORTD >= 180, clear PORTB (assuming PORTB = 255 initially.) Use LEDs connected to PORTB to visualize the output on breadboard.
2. Create a program to check if an 8-bit binary number (input through PORTD) is a palindrome. The result is to be indicated using LEDs connected to PORTB.

**Additional Task:**

1. **Develop a program that performs mathematical calculations based on user input from PORTD and uses loops and conditionals. The program should do the following:**
2. **Factorial Calculation:** If the input value from **PORTD** is between 1 and 5, calculate and display the factorial of the input on **PORTB** using a loop. For example, if **PORTD** is 4, the program should calculate 4! (4 × 3 × 2 × 1 = 24) and display the result on **PORTB** in binary format.
3. **Sum of Squares:** If the input is between 6 and 15, calculate the sum of squares of all integers from 1 to the value of **PORTD** (e.g., if **PORTD** is 5, calculate 1² + 2² + 3² + 4² + 5² = 55) using a loop and display the result on **PORTB.**
4. **Even Number Check:** If the input value is 16 or greater, check if the number is even. If **PORTD** is an even number, turn on all LEDs connected to **PORTB** (0xFF); if not, display an alternating LED pattern for 10 times with a delay.

**General Guidelines**

* Make the design intuitive.
* Elaborate the errors faced in hardware implementation and debugging.
* Attach appropriate pictures of the hardware results.
* Ensure originality in your documentation. Plagiarized reports get zero credit, whatsoever.
* Make submissions well in time to earn a maximum score.