**Lab:4**

**Interfacing Push buttons to 8051 using interrupts.**



**MBSD Lab**

**Spring 2023**

**Submitted by:**

**Muhammad Fahad Khan [20 Pwcse1940]**

**Semester: 6**

“On my honor, as a student of University of Engineering and Technology Peshawar, I have neither nor received unauthorized assistance on this academic work”

**Submitted to:**

**Dr:Amad Khalil**

**Task1: Interface Push buttons to 8051 microcontroller and blink odd LEDs using input buttons.**

**Answer:** To interface push buttons to an 8051 microcontroller and blink odd LEDs using the input buttons, you will need to connect the push buttons and LEDs to the appropriate GPIO pins of the microcontroller. Here's a step-by-step guide:

1. Connect the push buttons to the GPIO pins of the 8051 microcontroller. Each push button should be connected to a separate GPIO pin. You can use pull-up or pull-down resistors to ensure a stable logic level when the buttons are not pressed. For example, you can use pull-down resistors connected to Vcc and connect the buttons between the GPIO pins and ground.

2. Connect the LEDs to the GPIO pins of the microcontroller. Connect the anode (longer leg) of each LED to a separate GPIO pin and connect the cathode (shorter leg) to a current-limiting resistor. The other end of the resistor should be connected to ground.

3. Configure the GPIO pins as inputs for the push buttons and outputs for the LEDs in the microcontroller's code. You can refer to the datasheet or the programming manual of the 8051 microcontroller for specific instructions on how to configure the GPIO pins.

4. Write a program in a suitable programming language (such as C) to read the status of the push buttons and control the blinking of the LEDs. Here's a simple example program that blinks the odd-numbered LEDs when the corresponding push buttons are pressed:

```c

#include <reg51.h>

// Define the GPIO pins for push buttons and LEDs

sbit button1 = P1^0;

sbit button2 = P1^1;

sbit led1 = P2^0;

sbit led3 = P2^2;

sbit led5 = P2^4;

void delay(unsigned int time) {

unsigned int i, j;

for (i = 0; i < time; i++)

for (j = 0; j < 1275; j++); // Adjust this value for the desired delay

}

void main() {

// Set push button pins as inputs

button1 = 1;

button2 = 1;

// Set LED pins as outputs

led1 = 0;

led3 = 0;

led5 = 0;

while (1) {

// Check if button1 is pressed

if (!button1) {

led1 = ~led1; // Toggle LED1

delay(1000); // Delay for 1 second

}

// Check if button2 is pressed

if (!button2) {

led3 = ~led3; // Toggle LED3

led5 = ~led5; // Toggle LED5

delay(1000); // Delay for 1 second

}

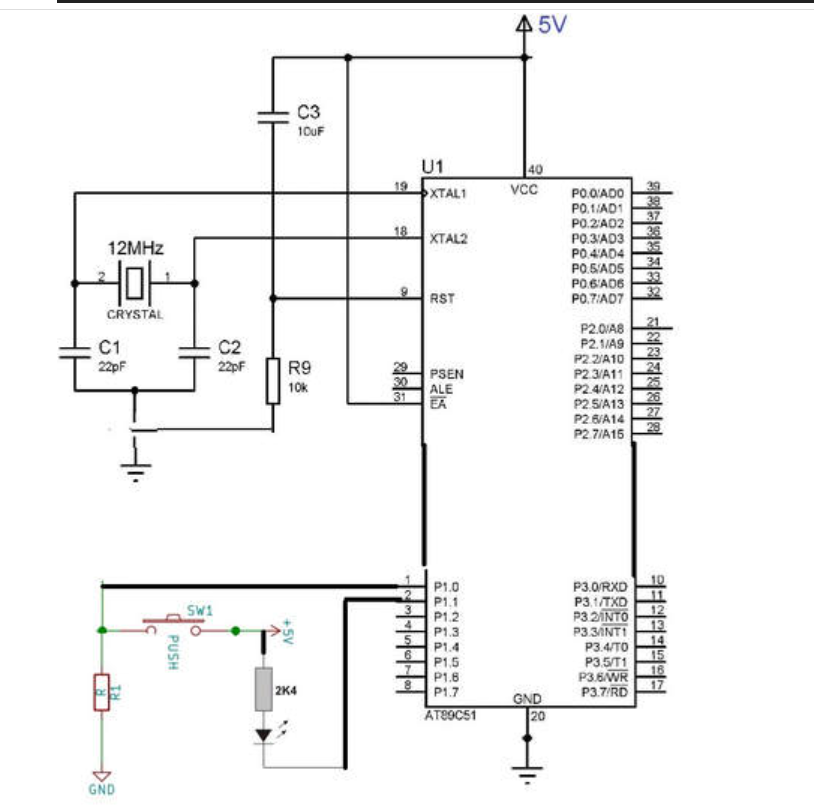
}

}

```

In this example, button1 is connected to P1^0, and button2 is connected to P1^1. LED1, LED3, and LED5 are connected to P2^0, P2^2, and P2^4, respectively. The `delay` function is used to introduce a delay between LED toggling to make the blinking visible. Adjust the delay value according to your requirements.

**Schemetric Diagram:**

****