**Interfacing button with MSP430 microcontroller and polling based Night rider  
Lab03**

****

**Spring 2025**

Submitted by: **Mohsin Sajjad**

Registration No: **22pwsce2149**

Class Section: **A**

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



Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Faheem Jan**

Month Day, Year (02 03, 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

# LAB No 3

**TASKS:**

**TASK1:**

**write a program which monitor a switch if it is pressed then toggle LED attach with P1.0**

**CODE:**  
int main(void) {

WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

PM5CTL0 &= ~LOCKLPM5; // Enable GPIOs

P1DIR |= 0x01; // Set P1.0 as output (LED)

P1DIR &= ~0x04; // Set P1.2 as input (Switch)

P1OUT |= 0x04; // Pull-up resistor on P1.2

P1REN |= 0x04; // Enable resistor on P1.2

while(1) {

if (!(P1IN & 0x04)) { // If switch is pressed mean condition P1.2 goes low and condtion will true

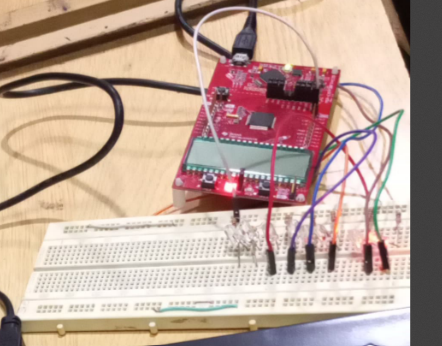
P1OUT ^= 0x01; // Toggle LED on P1.0

\_\_delay\_cycles(100000);

}

}

}

**OUTPUT:  
**

**TASK2: write a program which monitor a switch if it is not pressed then toggle LED attach with P1.0 if it is pressed then stop toggling.**

**#include <msp430.h>**

**CODE:**#include <msp430.h>

int main(void) {

WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

PM5CTL0 &= ~LOCKLPM5; // Enable GPIOs

P1DIR |= 0x01; // Set P1.0 as output (LED)

P1DIR &= ~0x04; // Set P1.2 as input (Switch)

P1OUT |= 0x04; // Pull-up resistor on P1.2

P1REN |= 0x04; // Enable resistor on P1.2

while(1) {

if (P1IN & 0x04) { // If switch is NOT pressed mean P1.2 will high

P1OUT ^= 0x01; // Toggle LED on P1.0

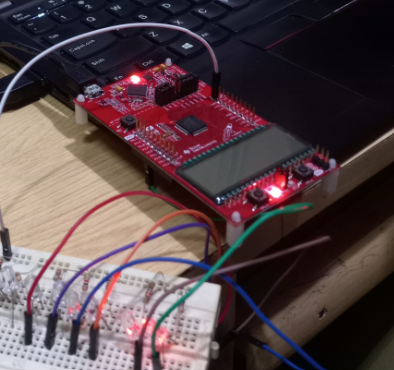
\_\_delay\_cycles(100000); // Toggle speed

}

}

}

**OUTPUT:**

**  
TASK3: write a program which monitor a switch if the switch is pressed the LED if on should off and if off should ON.**

**#include <msp430.h>**

**CODE:**int main(void) {

WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

PM5CTL0 &= ~LOCKLPM5; // Enable GPIOs

P1DIR |= 0x01; // Set P1.0 as output (LED)

P1DIR &= ~0x04; // Set P1.2 as input (Switch)

P1OUT |= 0x04; // Pull-up resistor on P1.2

P1REN |= 0x04; // Enable resistor on P1.2

while(1) {

if (!(P1IN & 0x04)) { // If switch is pressed

P1OUT ^= 0x01; // Toggle LED on P1.0

while (!(P1IN & 0x04)); // Wait for switch release

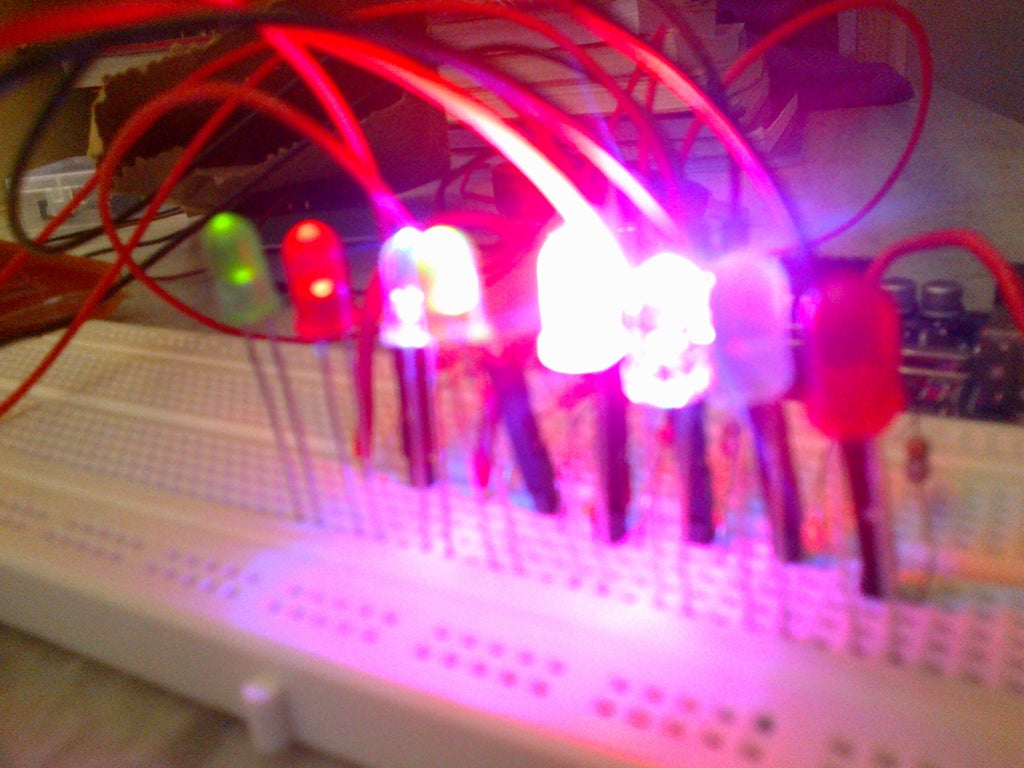
\_\_delay\_cycles(100000); // Debounce delay

}

}

}

**OUTPUT:**



**TASK4: Write a program which monitors a switch. If it is not pressed, then toggle the LED attached with P1.0. If it is pressed, then stop toggling.**

**CODE:**#include <msp430.h>

int main(void) {

WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

PM5CTL0 &= ~LOCKLPM5; // Enable GPIOs

P1DIR |= 0x01; // Set P1.0 as output (LED)

P1DIR &= ~0x04; // Set P1.2 as input (Switch)

P1OUT |= 0x04; // Pull-up resistor on P1.2

P1REN |= 0x04; // Enable resistor on P1.2

while(1) {

if (P1IN & 0x04) { // If switch is NOT pressed

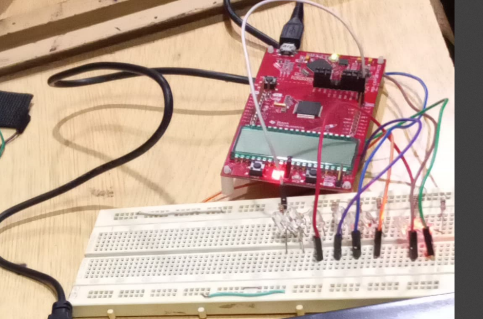
P1OUT ^= 0x01; // Toggle LED on P1.0

\_\_delay\_cycles(100000);

}

// If the switch is pressed, do nothing (stop toggling)

}

}  
**OUTPUT:**  


**Task 05:  
NIGHT RIDER**

**CODE:**

#include <msp430.h>

int main(void) {

WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

PM5CTL0 &= ~LOCKLPM5;

P8DIR |= 0xFF; // Set all 8 pins of Port 8 as output

P1DIR &= ~0x04; // Set P1.2 as input (Switch)

P8OUT = 0x00; // Start with all LEDs off

P1REN |= 0x04; // Enable pull-up resistor on P1.2

P1OUT |= 0x04; // Pull-up resistor active

unsigned char pattern = 0x01; // Start with the first LED

while (1) {

if (!(P1IN & 0x04)) { // If switch is pressed

\_\_delay\_cycles(100000); // Debounce delay

P8OUT = pattern; // Turn on current LED

\_\_delay\_cycles(100000); // Blink delay

P8OUT = 0x00; // Turn off all LEDs

\_\_delay\_cycles(100000); // Off delay

pattern <<= 1; // Shift pattern to the next LED

if (pattern == 0x00) { // If pattern goes beyond P8.7

pattern = 0x01; // Reset to the first LED

}

} else {

P8OUT = 0x00; // Turn off all LEDs when switch is not pressed

}

}

return 0;

}