

## LAB TASKS

**Q1: Develop the program to read the state of a push button. Watch serial monitor and take screenshot of both states. Add them in your lab report as our regular exercise.**

```
#define buttonPin 7 // Defining global variable

bool buttonstate; // Defining variable to store buttonstate value
int Status = LOW; // Defining variable to store Status value

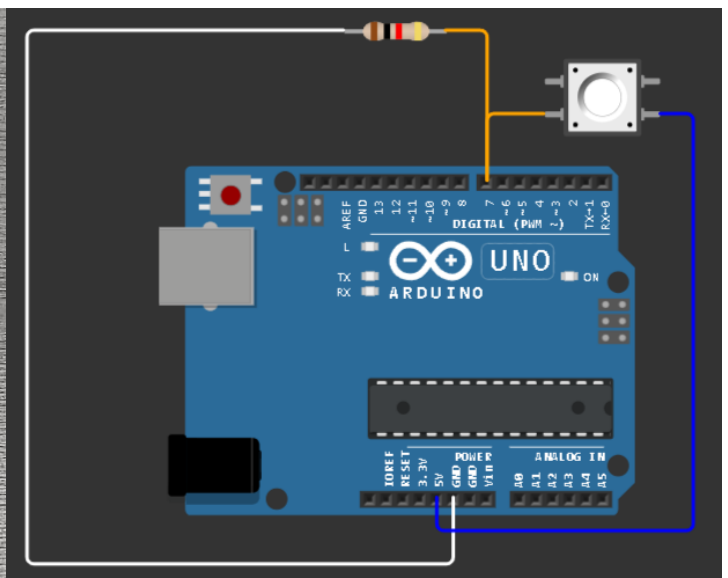
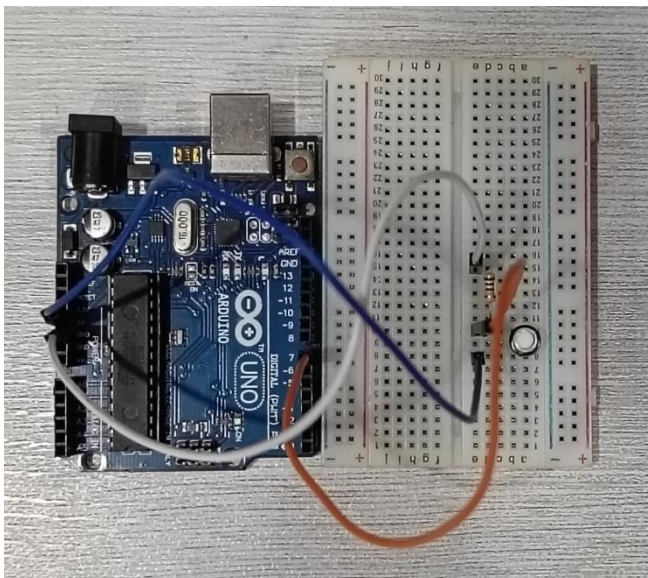
void setup() {
  Serial.begin(9600); // Calling serial monitor for data display
  pinMode(buttonPin, INPUT); // Port location for button
}

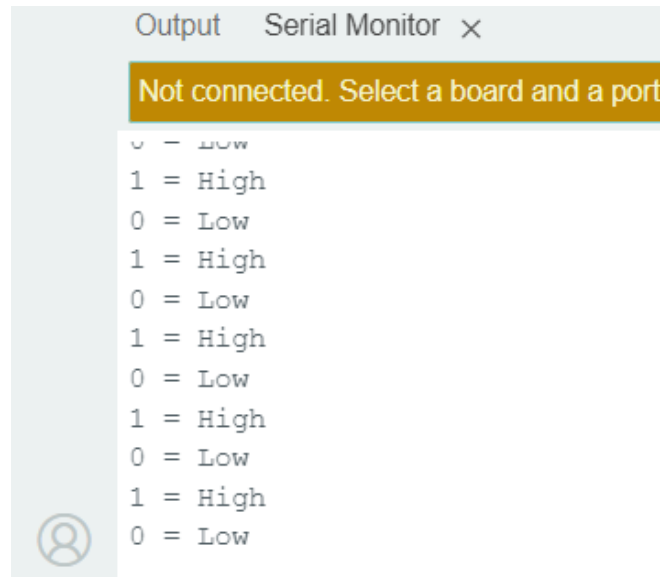
void loop() {
  buttonstate = digitalRead(buttonPin); // Reading the state of button and storing into the
  variable

  if (buttonstate == HIGH) { // Defining an if statement for the code to run when button is
  pressed
    Status = !Status; // Reversing the state of button state variable
    Serial.print(Status); // Printing value of Status to serial monitor

    if (Status == 1) { // Using additional if statements to print more appropriate text to
    serial monitor
      Serial.println(" = High"); // Printing text to serial monitor
    }
    if (Status == 0) { // Using additional if statements to print more appropriate text to
    serial monitor
      Serial.println(" = Low"); // Printing text to serial monitor
    }
  }

  delay(500); // Adding a time delay
}
```





**Q2: Develop a simple counter which displays the “number of times the push button is pressed”. Using sniffing tool, add the screenshot in your lab report when the counter value is 10.**

```
#define buttonPin 7 // Defining global variable

int count = 0; // Defining variable to store count value
bool buttonstate; // Defining variable to store buttonstate value

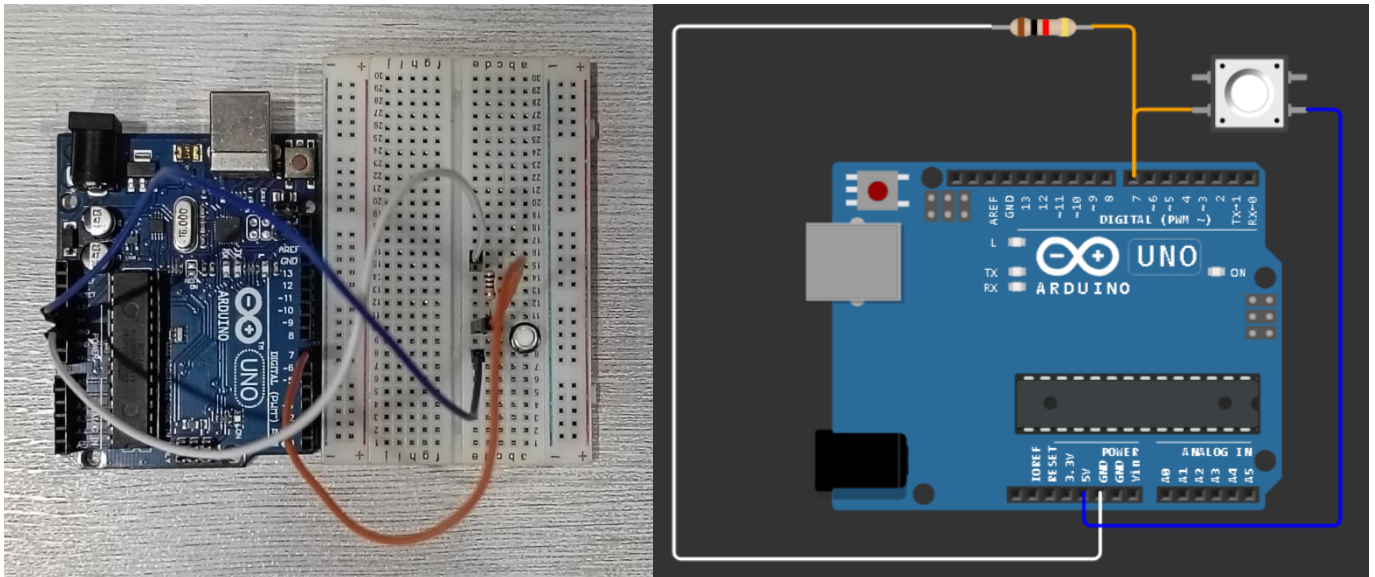
void setup() {
  Serial.begin(9600); // Calling serial monitor for data display
  pinMode(buttonPin, INPUT); // Port location for button
}

void loop() {
  buttonstate = digitalRead(buttonPin); // Reading the state of button and storing into the
  variable

  if (buttonstate == HIGH) { // Defining an if statement for the code to run when button is
  pressed
    count++; // Incrementing the count value by 1

    Serial.print("Number of button presses: "); // Printing text to the serial monitor
    Serial.print(count); // Printing value of count to the serial monitor
    Serial.println(" times"); // Printing text to the serial monitor

    delay(500); // Adding a delay
  }
}
```



Message (Enter to send message to 'Arduino Uno' on 'COM4')

```
Number of button presses: 1 times
Number of button presses: 2 times
Number of button presses: 3 times
Number of button presses: 4 times
Number of button presses: 5 times
Number of button presses: 6 times
Number of button presses: 7 times
Number of button presses: 8 times
Number of button presses: 9 times
Number of button presses: 10 times
```

**Q3: Develop the program which changes state of an LED by using a single reset push button (i.e., when the button is pressed, the LED should turn ON and when the button is pressed again, it should turn OFF).**

```
#define buttonPin 7 // Defining global variable
#define LEDPin 13 // Defining global variable

bool buttonstate; // Defining variable to store buttonstate value
int Status = LOW; // Defining variable to store Status value

void setup() {
    Serial.begin(9600); // Calling serial monitor for data display
    pinMode(buttonPin, INPUT); // Port location for button
    pinMode(LEDPin, OUTPUT); // Port location for red LED
}

void loop() {
    buttonstate = digitalRead(buttonPin); // Reading the state of button and storing into the
    variable
```

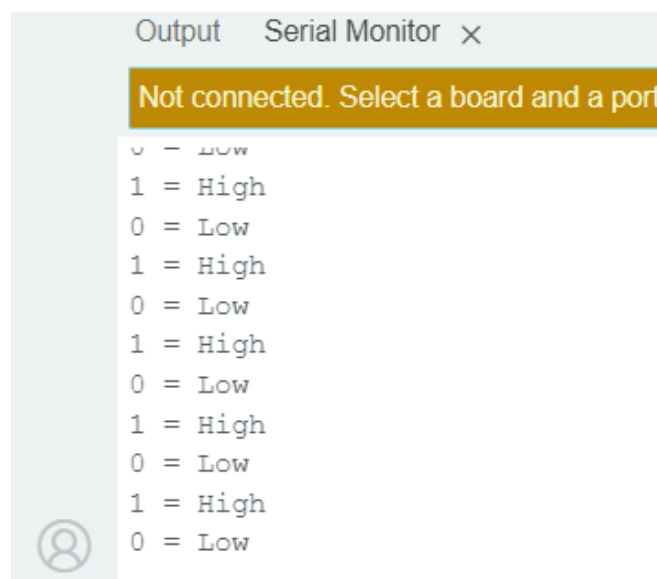
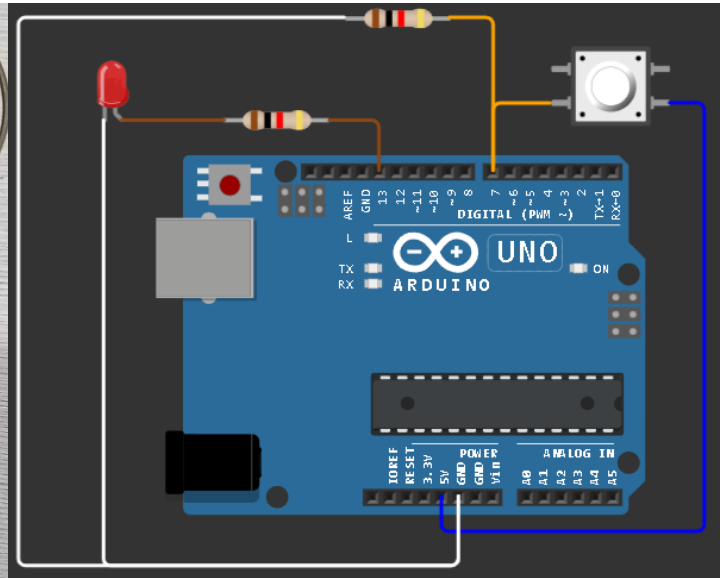
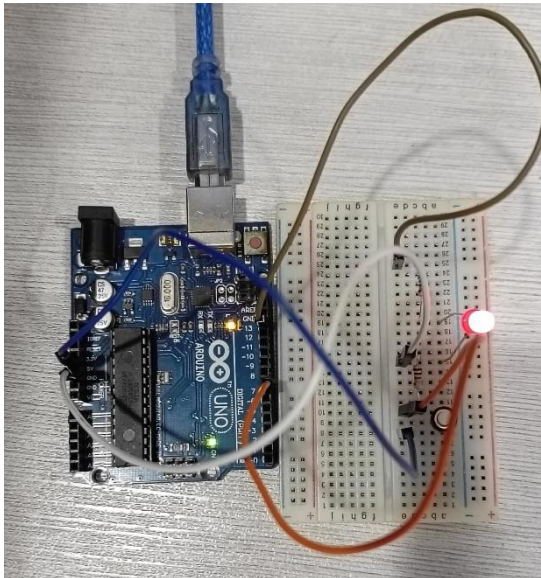
```

if (buttonstate == HIGH) { // Defining an if statement for the code to run when button is
pressed
    Status = !Status; // Reversing the state of button state variable
    digitalWrite(LEDPin,Status); // Defining the LED with a output using Status variable
    Serial.print(Status); // Printing value of Status to serial monitor

    if (Status == 1) { // Using additional if statements to print more appropriate text to
serial monitor
        Serial.println(" = High"); // Printing text to serial monitor
    }
    if (Status == 0) { // Using additional if statements to print more appropriate text to
serial monitor
        Serial.println(" = Low"); // Printing text to serial monitor
    }

    delay(500); // Adding a time delay
}
}

```



## POST LAB TASKS

**Q1: Suggest a real-life example of controlling an LED or any other instrument with the help of a push button.**

Push button has various real-life applications. Generally, push buttons are used where a sudden switch between on and off states is required. Some of these applications include buttons in water or soda dispensers, where you hold the button to fill your glass and release to stop the flow of fluid. Additionally, push button being used as reset buttons for systems. Push buttons are widely used in elevators to select respective floors, in dashboard of vehicles for the usage of various functions, in gaming controls, as click buttons in computer mice, in doorbells, and numerous other applications across industries.

**Q2) If you want to develop a program that consist of 3 LEDs and 1 push button such that when button is pressed, LED 1 turns ON. When the button is pressed again, LED 2 turns ON and LED 1 turns OFF at the same time and so on for the LED 3. This can be easily achieved by using SWITCH CASE structure.**

**Write down the syntax and code for above problem statement by using SWITCH CASE. Only write case structure.**

```
#define buttonPin 7 // Defining global variable
#define LEDPin3 13 // Defining global variable
#define LEDPin2 11 // Defining global variable
#define LEDPin1 9 // Defining global variable

bool buttonstate; // Defining variable to store buttonstate value
int Status = LOW; // Defining variable to store Status value
int count = 0; // Defining variable to store count value

void setup() {
    Serial.begin(9600); // Calling serial monitor for data display
    pinMode(buttonPin, INPUT); // Port location for button
    pinMode(LEDPin1, OUTPUT); // Port location for red LED
    pinMode(LEDPin2, OUTPUT); // Port location for yellow LED
    pinMode(LEDPin3, OUTPUT); // Port location for green LED
}

void loop() {
    buttonstate = digitalRead(buttonPin); // Reading the state of button and storing into the variable

    count = count + 1; // Incrementing the count value by 1
    //Serial.print(count); // Printing value of Status to serial monitor

    switch(count) { // Using switch statement to defining multiple cases for different outputs
        case 1: // Case number 1 defined by the Count variable value
            digitalWrite(LEDPin1,HIGH); // Output for LED defined with respective High
            digitalWrite(LEDPin2,LOW); // Output for LED defined with respective Low
            digitalWrite(LEDPin3,LOW); // Output for LED defined with respective Low
            delay(500); // Adding a time delay
            break; // Using break statement to get out off switch statement and also to not run the ahead code
```

```

case 2: // Case number 2 defined by the Count variable value
    digitalWrite(LEDPin1,LOW); // Output for LED defined with respective Low
    digitalWrite(LEDPin2,HIGH); // Output for LED defined with respective High
    digitalWrite(LEDPin3,LOW); // Output for LED defined with respective Low
    delay(500); // Adding a time delay
    break; // Using break statement to get out off switch statement and also to not run the
ahead code

case 3: // Case number 3 defined by the Count variable value
    digitalWrite(LEDPin1,LOW); // Output for LED defined with respective Low
    digitalWrite(LEDPin2,LOW); // Output for LED defined with respective Low
    digitalWrite(LEDPin3,HIGH); // Output for LED defined with respective High
    delay(500); // Adding a time delay
    break; // Using break statement to get out off switch statement and also to not run the
ahead code
}

if (count == 3) { // Using an if statment to reset the Count value
    count = 0; // Defining the count value again which will overwrite the previous value
}
}

```

