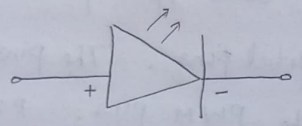
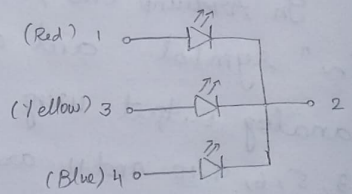
LED:



RGB LED:



EX. NO : 02
DATE :

PRE-LAB QUESTIONS:

1) Draw the electronic symbol for LED and RGB LED

An LED emits only one colour whereas a RGB LED can emit three different colours, namely red, green and blue, which in together gives 8 possible colours in general.

2) State the difference between a pull up resistor and a pull down resistor in a push button.

PULL UP RESISTOR	PULL DOWN RESISTOR
• Connected between input pin and +ve supply • When button is not pressed, pulls the voltage level to high state	• Connected between input pin and ground. • When button is not pressed, pulls the voltage level to low state.

3) What happens if we do not use a pull up resistor and a pull down resistor in a push button?

In it absence, the input will be in a floating state when the button is not pressed which can lead to unpredicted behaviours like erratic readings.

4) What is switch debounce? Explain it in brief.

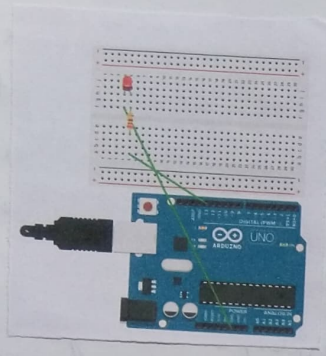
It is a phenomenon that occurs when a switch is toggled frequently. The noise that occurs while transitioning from one state to another can cause errors in the circuit. Switch debouncing in an electronic design ensures that the device that is sampling the switch waveform does not misinterpret a single button press as many.

5) Is it necessary to put a resistor to a LED? What will happen if a low value and a high value resistors are used with a LED?

It is necessary to put a resistor to a LED to prevent excessive current flow that could damage the LED.

USE OF LOW VALUE RESISTOR	USE OF HIGH VALUE RESISTOR
More current flows through the LED, potentially reducing the lifespan.	Less current flows through the LED, causing it to be dim or not light up at all.

1) CIRCUIT DIAGRAM:



PROGRAM:

```
#define LED 13
void setup() {
    pinMode (LED, OUTPUT);
}
void loop() {
    digitalWrite (LED, HIGH);
    delay (1000);
    digitalWrite (LED, LOW);
    delay (1000);
}
```


EXERCISE :

1) Blink of an external LED

AIM:
To blink an LED that is connected externally.

COMPONENTS REQUIRED:
Arduino UNO, USB-B Cable, 1 LED, Breadboard, 1 220 Ω resistor, wires

CONCLUSION:
Thus, an LED is made to blink is externally connected.

2) Blink of RGB LED

AIM:
To blink a RGB LED

COMPONENTS REQUIRED:
Arduino UNO, USB-B cable, 1 RGB LED, Breadboard, 3 220 Ω resistors, wires

CONCLUSION:
Thus, an LED of RGB type is made to blink

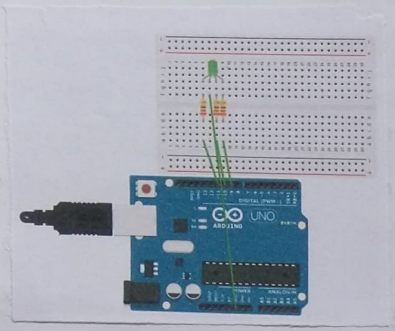
3) Push button + LED (When button is pressed the LED must glow. When button is not pressed the LED must remain LOW)

AIM:
To implement a circuit using push button and LED that toggles state each time.

COMPONENTS REQUIRED:
Arduino UNO, USB-B Cable, 1 LED, 1 push button, Breadboard, wires, 1 220 Ω resistor

CONCLUSION:
Thus, toggling of circuit using push button and LED is implemented.

2) CIRCUIT DIAGRAM:



2) PROGRAM:

```
#define RED 11
#define GREEN 10
#define BLUE 9
void setup() {
  pinMode (RED, OUTPUT);
  pinMode (GREEN, OUTPUT);
  pinMode (BLUE, OUTPUT);
}
void loop() {
  digitalWrite (RED, HIGH);
  digital Write (GREEN, LOW);
  digital Write (BLUE, LOW);
  delay (1000);
  digital Write (RED, LOW);
  digital Write (GREEN, HIGH);
  digital Write (BLUE, LOW);
  delay (1000);
  digital Write (RED, LOW);
  digitalWrite (GREEN, LOW);
  digital Write (BLUE, HIGH);
} delay (1000);
```


4) Push button + LED (without debouncing) (when button is pressed 1st time the LED must glow, when the button is pressed the 2nd time LED must remain in LOW state)

AIM:

To implement a circuit using push button and LED that toggles each time the state, the button is pressed. (without debouncing)

COMPONENTS REQUIRED:

Arduino UNO, USB-B Cable, 1 LED, 1 push button, Breadboard, wires, 1 220Ω resistor

CONCLUSION:

Thus, the circuit for toggling states without debouncing using push button and LED is implemented.

5) Push button + LED (with debouncing)

AIM:

To implement a circuit using push button and LED that toggles each time the state, the button is pressed (with debouncing)

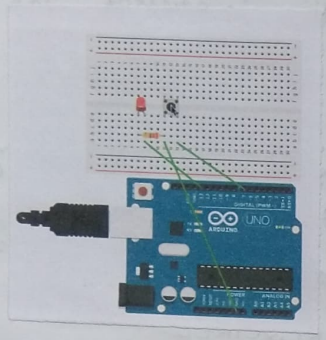
COMPONENTS REQUIRED:

Arduino UNO, USB-B cable, 1 LED, 1 push button, Breadboard, wires, 1 220Ω resistor

CONCLUSION:

Thus, the circuit for toggling states with debouncing using push button and LED is implemented.

3) CIRCUIT DIAGRAM:



PROGRAM:

```
# define LED 13
# define BUTTON 7
void setup() {
    pinMode (LED, OUTPUT);
    pinMode (BUTTON, INPUT_PULLUP);
}
void loop() {
    int digitalVal = digitalRead (BUTTON);
    if (HIGH == digitalVal) {
        digitalWrite (LED, LOW);
    }
    else {
        digitalWrite (LED, HIGH);
    }
}
```


6) Push button + RGB LED (with debouncing)

AIM:

To implement a circuit using push button and RGB LED that toggles to red, green and blue each time the button is pressed.

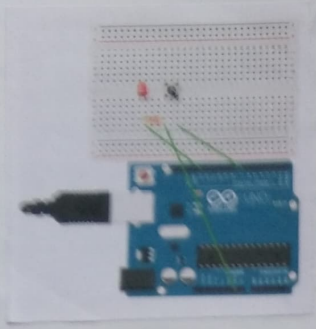
COMPONENTS REQUIRED:

Arduino UNO, USB-B cable, 1 RGB LED, 1 push button, Breadboard, wires, 1 220Ω resistor

CONCLUSION:

Thus, the circuit for toggling to red, green and blue with debouncing using push button and RGB LED is implemented.

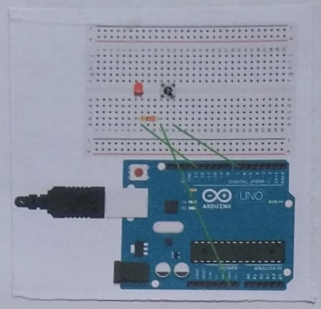
4) ~~PRO~~ CIRCUIT DIAGRAM:



PROGRAM:

```
#define LED 13
#define BUTTON 7

int buttonState =0;
int lastButtonState = HIGH;
int ledState =LOW;

void setup() {
   pinMode (LED, OUTPUT);
   pinMode (BUTTON, INPUT_PULLUP);

}

void loop() {
   buttonState = digitalRead (BUTTON);
   if (buttonState == HIGH && lastButtonState == LOW){
      ledState = !ledState;
      digitalWrite (LED, ledState);
   }
   lastButtonState = buttonState;

}
```

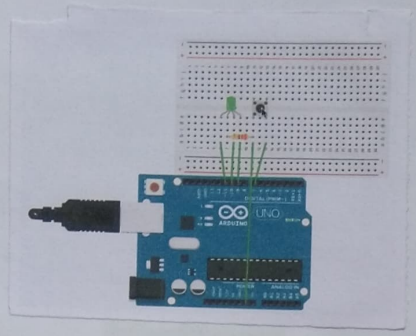
5) CIRCUIT DIAGRAM:



PROGRAM:

```
#define BUTTON 7
#define LED 13

int buttonState = 0;
int lastButtonState = LOW;
int ledState = HIGH;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;
void setup() {
    pinMode (LED, OUTPUT);
    pinMode (BUTTON, INPUT_PULLUP);
}
void loop() {
    int reading = digitalRead (BUTTON);
    if (reading != lastButtonState) {
        lastDebounceTime = millis ();
    }
    if ((millis() - lastDebounceTime) > debounceDelay) {
        if (reading != buttonState) {
            buttonState = reading;
        }
        if (buttonState == HIGH) {
            ledState = !ledState; digitalWrite (LED, ledState);
lastButtonState = reading;
}}}
```

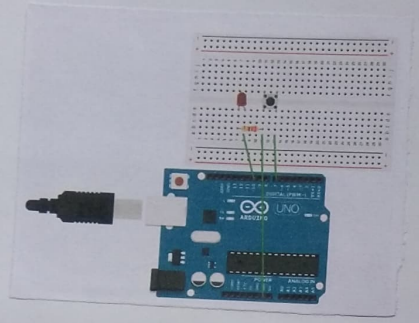
6) CIRCUIT DIAGRAM :



PROGRAM:

```
#define BUTTON 7
#define RED 11
#define GREEN 10
#define BLUE 9
int buttonState = 0, lastButtonState = LOW, colourIndex = 0;
unsigned long lastDebounceTime = 0, debounceDelay = 50;
void setup() {
   pinMode(RED, OUTPUT); pinMode(GREEN, OUTPUT);
   pinMode(BLUE, OUTPUT); pinMode(BUTTON, INPUT_PULLUP);
}
void loop() {
   int reading = digitalRead(BUTTON);
   if (reading != lastButtonState) {
      lastDebounceTime = millis();
   }
   if ((millis() - lastDebounceTime) > debounceDelay) {
      if (reading != buttonState) {
         buttonState = reading;
         if (buttonState == HIGH) {
            digitalWrite(RED, LOW); digitalWrite(GREEN, LOW);
            digitalWrite(BLUE, LOW);
            if (colourIndex == 0) { digitalWrite(RED, HIGH); }
            else if (colourIndex == 1) { digitalWrite(GREEN, HIGH); }
            else { digitalWrite(BLUE, HIGH); }
            colourIndex = (colourIndex + 1) % 3; } } }
   lastButtonState = reading;
}
```

CIRCUIT DIAGRAM:



PROGRAM:

```
#define BUTTON 7
#define LED 10
int buttonState = 0, lastButtonState = LOW, count = 0;
unsigned long lastDebounceTime = 0, debounceDelay = 50;
void setup() {
   pinMode (LED, OUTPUT);
   pinMode (BUTTON, INPUT_PULLUP);
}
void loop() {
  int reading = digitalRead(BUTTON);
  if (reading != lastButtonState) {
     lastDebounceTime = millis();
  }
  if ((millis() - lastDebounceTime) > debounceDelay) {
     if (reading != buttonState) {
        buttonState = reading;
        if (buttonState == HIGH) {
           count++;
           glowLED (count);
        }
     }
  }
  lastButtonState = reading;
}
```

```
void glowLED (int count) {
   for (int i=0; i<count; i++) {
      digitalWrite(LED, HIGH);
      delay (500);
      digitalWrite(LED, LOW);
   } delay (500);
}
```

POST-LAB QUESTIONS:

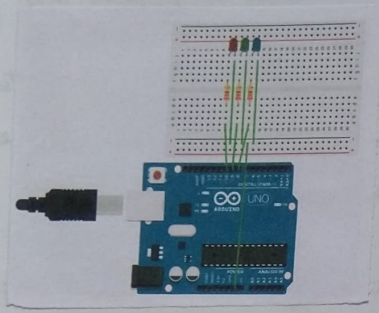
1) Write an Arduino sketch to perform the following

a) Count the # of times the push button is pressed.

b) Based on the count value the external LED connected must glow (e.g) when the push button is pressed for the 1st time LED must glow twice (use the debounce logic)

c) Draw the necessary circuit diagram.

CIRCUIT DIAGRAM:



PROGRAM:

```
#define LED1 8
#define LED2 9
#define LED3 10
void setup() {
  pinMode (LED1, OUTPUT);
  pinMode (LED2, OUTPUT);
  pinMode (LED3, OUTPUT);
}
void loop() {
  for (int i=0; i<=7; i++) {
    displayBinary (i);
    delay (1000);
  }
}
void displayBinary (int i) {
  digitalWrite (LED1, i & 0b001);
  digitalWrite (LED2, (i >>1) & 0b001);
  digitalWrite (LED3, (i>>2) & 0b001);
}
```


2) Write an Arduino sketch to perform the following

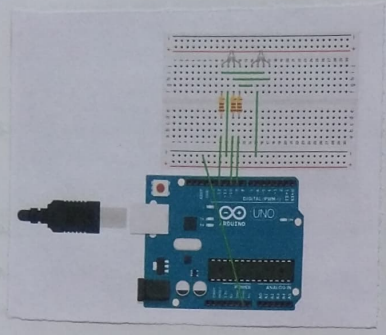
a) Connect an array of 3 LEDs.

b) Design a counter to display the binary equivalent of numbers from 0 to 7 using three LEDs

c) When the number 7 is reached, the value must again start with 0 (e.g.) 0 for all LED in LOW state. 1 - LED in HIGH State and others in LOW state.

d) Draw the necessary circuit diagram.

CIRCUIT DIAGRAM:



PROGRAM:

```
#define LED1 12
#define LED2 10
#define LED3 9
void setup() {
   pinMode (LED1, OUTPUT);
   pinMode (LED2, OUTPUT);
   pinMode (LED3, OUTPUT);
}
void loop() {
  digitalWrite (LED1, HIGH);
  delay (2000);
  digitalWrite (LED1, LOW);

  digitalWrite (LED2, HIGH);
  delay (2000);
  digitalWrite (LED2, LOW);

  digitalWrite (LED3, HIGH);
  delay (2000);
  digitalWrite (LED3, LOW);
}
```

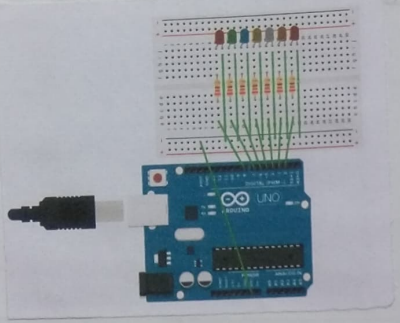

3) Write an Arduino sketch to perform the following

a) Connect 2 RGB LEDs

b) The colour display of both the LEDs must be as follows

LED1	LED2	Delay
Red	Blue	2s
Blue	Green	2s
Green	Red	3s

CIRCUIT DIAGRAM:



PROGRAM:

```
void setup() {
    for(int i=8; i>=2; i--) {
        pinMode(i, OUTPUT);
    }
}
void loop() {
    for(int i=8; i>=2; i--) {
        digitalWrite(i, HIGH);
        delay(1000);
        digitalWrite(i, LOW);
    }
}
```


4) Write an Arduino sketch to perform the following

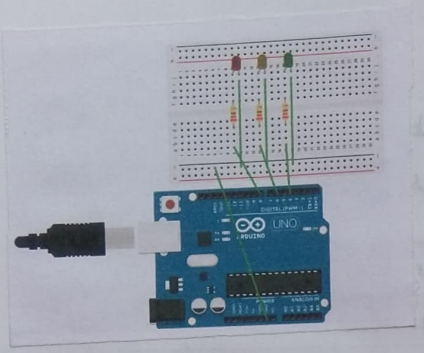
a) Connect an array of 7 LEDs.

b) Perform blink of LEDs as stated. Set the delay as 1S.

LED 1	LED 2	LED 3	LED 4	LED 5	LED 6	LED 7
HIGH	LOW	LOW	LOW	LOW	LOW	LOW
LOW	HIGH	LOW	LOW	LOW	LOW	LOW
LOW	LOW	HIGH	LOW	LOW	LOW	LOW
LOW	LOW	LOW	HIGH	LOW	LOW	LOW
LOW	LOW	LOW	LOW	HIGH	LOW	LOW
LOW	LOW	LOW	LOW	LOW	HIGH	LOW
LOW	LOW	LOW	LOW	LOW	LOW	HIGH

c) Draw the necessary circuit diagram.

CIRCUIT DIAGRAM:



PROGRAM:

```
#define R_LED 8
#define Y_LED 6
#define G_LED 4
void setup(){
    pinMode (R_LED, OUTPUT);
    pinMode (Y_LED, OUTPUT);
    pinMode (G_LED, OUTPUT);
}
void loop(){
    digitalWrite (R_LED, HIGH);
    delay (45 *1000);
    digitalWrite (R_LED, LOW);
    digitalWrite (Y_LED, HIGH);
    delay (5 *1000);
    digitalWrite (Y_LED, LOW);
    digitalWrite (G_LED, HIGH);
    delay (50*1000);
    digitalWrite (G_LED, LOW);
}
```

5) Write an Arduino Sketch to perform the following

a) Connect LEDs of Colors red, yellow and green

b) Simulate a traffic control system in such a way that

i) Start with red light. Make it glow for 45 seconds

ii) After 45 seconds, glow yellow LED for 5 seconds

iii) After 5 seconds, glow green LED for 50 seconds.

c) Draw the necessary circuit diagram.