RGB LED(Common Cathode)

```
#include <Arduino.h>

/*RGB LED pins (Common Cathode)*/
#define Red_Pin 12
#define Green_Pin 11
#define Blue_Pin 13
/*Delay  (in milliseconds)*/
#define COLOR_CHANGE_DELAY 1000

void setup() {
  /*Initialize the RGB LED pins as outputs*/
  pinMode(Red_Pin, OUTPUT);
  pinMode(Green_Pin, OUTPUT);
  pinMode(Blue_Pin, OUTPUT);
}

void loop() {
  /*Turn on Red, turn off Green and Blue*/
  digitalWrite(Red_Pin, HIGH);
  digitalWrite(Green_Pin, LOW);
  digitalWrite(Blue_Pin, LOW);
  delay(COLOR_CHANGE_DELAY);

  /*Turn off Red, turn on Green, turn off Blue*/
  digitalWrite(Red_Pin, LOW);
  digitalWrite(Green_Pin, HIGH);
  digitalWrite(Blue_Pin, LOW);
  delay(COLOR_CHANGE_DELAY);

  /*Turn off Red, turn off Green, turn on Blue*/
  digitalWrite(Red_Pin, LOW);
  digitalWrite(Green_Pin, LOW);
  digitalWrite(Blue_Pin, HIGH);
  delay(COLOR_CHANGE_DELAY);

  /*Turn ON Red,Green,Blue*/
  digitalWrite(Red_Pin, HIGH);
  digitalWrite(Green_Pin, HIGH);
  digitalWrite(Blue_Pin, HIGH);
  delay(COLOR_CHANGE_DELAY);
}
```

RGB LED (Common Anode)

```
#include <Arduino.h>

/*RGB LED pins (Common Anode)*/
#define Red_Pin 9
#define Green_Pin 10
#define Blue_Pin 11

/*Delay  (in milliseconds)*/
#define COLOR_CHANGE_DELAY 1000

void setup() {
  // Initialize the RGB LED pins as outputs
  pinMode(Red_Pin, OUTPUT);
  pinMode(Green_Pin, OUTPUT);
  pinMode(Blue_Pin, OUTPUT);
}

void loop() {
  /* Pattern 1: Red */
  digitalWrite(Red_Pin, LOW);   // Turn on Red (Common Anode)
  digitalWrite(Green_Pin, HIGH); // Turn off Green
  digitalWrite(Blue_Pin, HIGH);  // Turn off Blue
  delay(COLOR_CHANGE_DELAY);

  /* Pattern 2: Green */
  digitalWrite(Red_Pin, HIGH);   // Turn off Red
  digitalWrite(Green_Pin, LOW);  // Turn on Green (Common Anode)
  digitalWrite(Blue_Pin, HIGH);  // Turn off Blue
  delay(COLOR_CHANGE_DELAY);

  /* Pattern 3: Blue */
  digitalWrite(Red_Pin, HIGH);   // Turn off Red
  digitalWrite(Green_Pin, HIGH); // Turn off Green
  digitalWrite(Blue_Pin, LOW);   // Turn on Blue (Common Anode)
  delay(COLOR_CHANGE_DELAY);

  /*Pattern 4: white (Red + Blue+ Green)*/
  digitalWrite(Red_Pin, LOW);    // Turn on Red (Common Anode)
  digitalWrite(Green_Pin, HIGH); // Turn off Green
  digitalWrite(Blue_Pin, LOW);   // Turn on Blue (Common Anode)
  delay(COLOR_CHANGE_DELAY);
}
```

PULL-UP Button

```
#include <Arduino.h>

#define buttonPin  2

void setup() {
  // Initialize the button pin as an input
  pinMode(buttonPin, INPUT);
  // Initialize serial communication
  Serial.begin(9600);
  Serial.println("Active-High Push Button Status:");
}

void loop() {
  /*Read the status of the button*/
  bool buttonState = digitalRead(buttonPin);

  // Check for valid button state
  if (buttonState == HIGH || buttonState == LOW) {
    // Print the button status
    if (buttonState == LOW) {
      Serial.println("Button is pressed (Active-High)");
    } else {
      Serial.println("Button is not pressed");
    }
  } else {
    // Handle invalid button state
    Serial.println("Error: Invalid button state");
  }

  delay(1000);
}
```

PULL-Down Button

```
#include <Arduino.h>

#define buttonPin  2

void setup() {
  // Initialize the button pin as an input
  pinMode(buttonPin, INPUT);
  // Initialize serial communication
  Serial.begin(9600);
  Serial.println("Active-High Push Button Status:");
}

void loop() {
  /*Read the status of the button*/
  bool buttonState = digitalRead(buttonPin);

  // Check for valid button state
  if (buttonState == HIGH || buttonState == LOW) {
    // Print the button status
    if (buttonState == HIGH) {
      Serial.println("Button is pressed (Active-High)");
    } else {
      Serial.println("Button is not pressed");
    }
  } else {
    // Handle invalid button state
    Serial.println("Error: Invalid button state");
  }

  delay(1000);
}
```

Buzzer

```
#include <Arduino.h>
// Define the buzzer pin
const int buzzerPin = 8; // You can change this to the pin where your buzzer is connected
// Define the duration of the beeping sound in milliseconds
const int beepDuration = 200; // Change this value to adjust the beep duration
// Define the number of beeps
const int numBeeps = 5; // Change this value to adjust the number of beeps
void beep(void) ;
void setup() {
  // Initialize the buzzer pin as an output
  pinMode(buzzerPin, OUTPUT);
}
void loop() {
  // Generate a sequence of beeps
  for (int i = 0; i < numBeeps; i++) {
    beep();
    delay(500); // Delay between beeps (adjust as needed)
  }
   // Wait for a moment before repeating the sequence
  delay(2000); // Adjust this delay as needed
}
// Function to generate a beep
void beep(void) {
  digitalWrite(buzzerPin, HIGH); // Turn on the buzzer
  delay(beepDuration); // Beep duration
  digitalWrite(buzzerPin, LOW); // Turn off the buzzer
  delay(beepDuration); // Delay between beeps (adjust as needed)
}
```

Sliding Switch

```
#include <Arduino.h>

// Define the slide switch pin
const int switchPin = 2; // Change to the pin where your slide switch is connected
void setup() {
  // Initialize the slide switch pin as an input
  pinMode(switchPin, INPUT_PULLUP);
  // Initialize serial communication
  Serial.begin(9600);
  Serial.println("Slide Switch Status:");
}

void loop() {
  // Read the status of the slide switch (LOW when switched, HIGH when not switched)
  int switchState = digitalRead(switchPin);

  // Check if the slide switch is in the ON position (switched)
  if (switchState == LOW) {
    Serial.println("Switch ON");
  } else {
    Serial.println("Switch OFF");
  }

  delay(500); // Add a small delay to avoid rapid toggling of status
}
```

Relay Control

```
#include <Arduino.h>

// Define the relay control pin
const int relayPin = 13; // Change to the pin where your relay module's control pin is connected

void setup() {
  // Initialize the relay control pin as an output
  pinMode(relayPin, OUTPUT);

  // Initialize serial communication
  Serial.begin(9600);
  Serial.println("Relay Control for 230V Appliance:");

  // Turn off the relay initially for safety
  digitalWrite(relayPin, LOW);
}

void loop() {
  // Prompt the user to control the appliance
  Serial.println("Enter '1' to turn ON, '0' to turn OFF:");

  while (Serial.available() <= 0);
  char command = Serial.read();

  if (command == '1') {
    Serial.println("Turning ON the Relay...");
    digitalWrite(relayPin, HIGH); // Turn ON the relay
  } else if (command == '0') {
    Serial.println("Turning OFF the Relay...");
    digitalWrite(relayPin, LOW); // Turn OFF the relay
  } else {
    Serial.println("Enter '1' to turn ON, '0' to turn OFF.");
  }

  // Wait for a moment
  delay(1000);
}
```

Potentiometer

```
// Include necessary libraries
#include <Arduino.h>  // The Arduino core library

// Define pin constants
const int POTENTIOMETER_PIN = A1;  // Analog input pin

void setup() {
  // Initialize serial communication
  Serial.begin(9600);
}

void loop() {
  // Read the voltage from the potentiometer
  uint16_t sensorValue = analogRead(POTENTIOMETER_PIN);

  Serial.print("RAWDATA:");
  Serial.print(sensorValue);

  // Convert the sensor value to voltage (0-5V)
  float voltage = (sensorValue / 1023.0) * 5.0;

  // Print the voltage to the serial monitor
  Serial.print("Voltage: ");
  Serial.print(voltage, 2);  // Print with 2 decimal places
  Serial.println(" V");

  // Add a delay if needed
  delay(500);  // 500 milliseconds
}
```

IR Sensor

```
#include <Arduino.h>  // The Arduino core library

// Define pin constants
#define  IR_SENSOR_PIN  11  // Digital input pin for the IR sensor

void setup() {
  // Initialize serial communication
  Serial.begin(9600);

  // Define IR sensor pin as an input
  pinMode(IR_SENSOR_PIN, INPUT);
}

void loop() {
  // Read the digital output from the IR sensor
  bool sensorValue = digitalRead(IR_SENSOR_PIN);

  // Print the sensor value to the serial monitor
  Serial.print("IR Sensor Output: ");
  Serial.println(sensorValue);

  // Add a delay if needed
  delay(1000);  // 1000 milliseconds
}
```

UltraSonic Sensor

```
#include <Arduino.h>  // The Arduino core library

// Define pin constants
const int TRIGGER_PIN = 8;  // Digital output pin for the sensor's trigger
const int ECHO_PIN = 9;     // Digital input pin for the sensor's echo

// Define constants for speed of sound in air (in meters per second)
const float SPEED_OF_SOUND = 343.0;

void setup() {
  // Initialize serial communication
  Serial.begin(9600);

  // Define trigger pin as an output and echo pin as an input
  pinMode(TRIGGER_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

void loop() {
  // Generate a 10us pulse on the trigger pin to start the measurement
  digitalWrite(TRIGGER_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);

  // Measure the time it takes for the echo pin to go HIGH
  long duration = pulseIn(ECHO_PIN, HIGH);

  // Calculate the distance based on the time and speed of sound
  float distance = (duration * 0.5 * SPEED_OF_SOUND) / 10000.0;  // Convert to centimeters

  // Print the distance to the serial monitor
  Serial.print("Distance: ");
  Serial.print(distance, 2);  // Print with 2 decimal places
  Serial.println(" cm");

  // Add a delay if needed
  delay(1000);  // 100 milliseconds
}
```

DHT Temperature and humidity

```
#include <DFRobot_DHT11.h>
DFRobot_DHT11 DHT;
#define DHT11_PIN 8

void setup() {
  Serial.begin(9600);
}

void loop() {
  DHT.read(DHT11_PIN);
  Serial.print("temp:");
  Serial.print(DHT.temperature);
  Serial.print("  humi:");
  Serial.println(DHT.humidity);
  delay(1000);
}
```

LM35 voltage and temperature

```
// Include necessary libraries
#include <Arduino.h>  // The Arduino core library

// Define pin constants
const int TEMPERATURE_SENSOR_PIN = A0;  // Analog input pin for the LM35 sensor

void setup() {
  // Initialize serial communication
  Serial.begin(9600);
}

void loop() {
  // Read the analog voltage from the LM35 sensor
  int sensorValue = analogRead(TEMPERATURE_SENSOR_PIN);

  // Convert the sensor value to temperature in degrees Celsius
  float voltage = (sensorValue / 1023.0) * 5.0;  // Use 5V reference
  float temperatureCelsius = (voltage) * 100.0;  // LM35 output is 10 mV per degree Celsius

  // Print the temperature to the serial monitor
  Serial.print("Voltage: ");
  Serial.print(voltage, 2); // Print voltage with 2 decimal places
  Serial.println(" V");

  Serial.print("Temperature (Celsius): ");
  Serial.print(temperatureCelsius, 3); // Print temperature with 2 decimal places
  Serial.println(" °C");

  // Add a delay if needed
  delay(1000);  // 1000 milliseconds (1 second)
}
```

LDR

```
#define LDR 3
#define LED 13
void setup() {
  // put your setup code here, to run once:
  pinMode(LDR,INPUT);
  pinMode(LED,OUTPUT);
  Serial.begin(9600);
  Serial.println("Light Lux Sensor Starting to detect...");
}

void loop() {
  // put your main code here, to run repeatedly:
  bool Sensor_Value = digitalRead(LDR);
  Serial.print("Lux_Value :");
  Serial.print(Sensor_Value);
  Serial.println("lux");

 if(Sensor_Value ==0){
  digitalWrite(LED,0);
  Serial.print("Light is detected... it's Morning!!!");
 }
 else{
  digitalWrite(LED,1);
  Serial.print("Dark...Please Turn on the Light!!!");
 }
}
```

Touch Sensor

```
const int SENSOR_PIN = 7; // the Arduino's input pin that connects to the sensor's SIGNAL pin
const int Buzzer = 2;
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // initialize the Arduino's pin as an input
  pinMode(SENSOR_PIN, INPUT);
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(Buzzer, OUTPUT);
}

void loop() {
  // read the state of the the input pin:
  int state = digitalRead(SENSOR_PIN);

  // control Buzzer according to the sensor's state
  digitalWrite(Buzzer, state);
}
```

Gas Sensor

```
#define MQ135 A0
#define LED 13
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
 float Sensor_Value = analogRead(MQ135);
 Serial.println(Sensor_Value);
  if(MQ135 < 300){
    digitalWrite(LED,1);
    Serial.println("Smoke has been Detected...");
  }
  else{
    digitalWrite(LED,0);
    Serial.println("Smoke has been not Detected...");
  }


}
```