**Traffic Light System**

Select a few LEDs (red, yellow, and green), resistors, and some jumper wires. This project simulates a basic traffic light system.

**Components:**

* Arduino Mega
* Red LED
* Yellow LED
* Green LED
* 3x 220-ohm resistors
* Jumper wires
* Breadboard

**Circuit Setup:**

1. Connect the anode (longer leg) of the Red LED to digital pin 8 on the Arduino Mega through a 220-ohm resistor.
2. Connect the anode of the Yellow LED to digital pin 9 on the Arduino Mega through a 220-ohm resistor.
3. Connect the anode of the Green LED to digital pin 10 on the Arduino Mega through a 220-ohm resistor.
4. Connect all the cathodes (shorter legs) of the LEDs to the ground (GND) on the Arduino Mega.

**TRAFFIC LIGHT SYSTEM ARDUINO CODE**

**const int redLED = 11;     // Pin for the red LED**

**const int yellowLED = 12;  // Pin for the yellow LED**

**const int greenLED = 13;   // Pin for the green LED**

**void setup() {**

**// Set the LED pins as outputs**

**pinMode(redLED, OUTPUT);**

**pinMode(yellowLED, OUTPUT);**

**pinMode(greenLED, OUTPUT);**

**}**

**void loop() {**

**// Green light**

**digitalWrite(greenLED, HIGH);  // Turn on the green LED**

**delay(7000);                   // Wait for 7 seconds**

**digitalWrite(greenLED, LOW);   // Turn off the green LED**

**// Yellow light**

**digitalWrite(yellowLED, HIGH); // Turn on the yellow LED**

**delay(4000);                   // Wait for 4 seconds**

**digitalWrite(yellowLED, LOW);  // Turn off the yellow LED**

**TO MAKE IT BLINK**

// Blink each LED

  blinkLED(redLED);

  blinkLED(yellowLED);

  blinkLED(greenLED);

}

// Function to blink the LED

void blinkLED(int ledPin) {

  for (int i = 0; i < 3; i++) {   // Blink 3 times

    digitalWrite(ledPin, HIGH);   // Turn on the LED

    delay(500);                   // Wait for 0.5 seconds

    digitalWrite(ledPin, LOW);    // Turn off the LED

    delay(500);                   // Wait for 0.5 seconds

  }

}

**TASK 2: SIMULATING REAL TRAFFIC LIGHT**

const int redLED = 11; // Pin for the red LED

const int yellowLED = 12; // Pin for the yellow LED

const int greenLED = 13; // Pin for the green LED

void setup() {

// Set the LED pins as outputs

pinMode(redLED, OUTPUT);

pinMode(yellowLED, OUTPUT);

pinMode(greenLED, OUTPUT);

// Turn off all LEDs initially

digitalWrite(redLED, LOW);

digitalWrite(yellowLED, LOW);

digitalWrite(greenLED, LOW);

}

void loop() {

// Green light

digitalWrite(greenLED, HIGH); // Turn on the green LED

delay(7000); // Wait for 7 seconds

digitalWrite(greenLED, LOW); // Turn off the green LED

// Yellow light

digitalWrite(yellowLED, HIGH); // Turn on the yellow LED

delay(2000); // Wait for 2 seconds

digitalWrite(yellowLED, LOW); // Turn off the yellow LED

// Red light

digitalWrite(redLED, HIGH); // Turn on the red LED

delay(7000); // Wait for 7 seconds

digitalWrite(redLED, LOW); // Turn off the red LED

// Turn off all LEDs

digitalWrite(redLED, LOW);

digitalWrite(yellowLED, LOW);

digitalWrite(greenLED, LOW);

}

**TASK 3: BUTTON TO CONTROL THE LIGHTS**

**Components:**

* **Arduino Mega**
* **Red LED**
* **Yellow LED**
* **Green LED**
* **3x 220-ohm resistors**
* **Jumper wires**
* **Breadboard**
* **Push button**
* **10k-ohm resistor**

**Circuit Setup:**

1. **Connect the anode (longer leg) of the Red LED to digital pin 8 on the Arduino Mega through a 220-ohm resistor.**
2. **Connect the anode of the Yellow LED to digital pin 9 on the Arduino Mega through a 220-ohm resistor.**
3. **Connect the anode of the Green LED to digital pin 10 on the Arduino Mega through a 220-ohm resistor.**
4. **Connect all the cathodes (shorter legs) of the LEDs to the ground (GND) on the Arduino Mega.**
5. **Connect one leg of the push button to digital pin 2 on the Arduino Mega.**
6. **Connect the other leg of the push button to GND.**
7. **Connect a 10k-ohm resistor between digital pin 2 and 5V (pull-up resistor).**

**Arduino Code:**

**const int redLED = 8;**

**const int yellowLED = 9;**

**const int greenLED = 10;**

**const int buttonPin = 2;**

**int lightState = 0;**

**int buttonState = 0;**

**int lastButtonState = 0;**

**void setup() {**

**pinMode(redLED, OUTPUT);**

**pinMode(yellowLED, OUTPUT);**

**pinMode(greenLED, OUTPUT);**

**pinMode(buttonPin, INPUT\_PULLUP);**

**}**

**void loop() {**

**buttonState = digitalRead(buttonPin);**

**if (buttonState == LOW && lastButtonState == HIGH) {**

**lightState++;**

**if (lightState > 2) {**

**lightState = 0;**

**}**

**delay(200); // debounce delay**

**}**

**lastButtonState = buttonState;**

**switch (lightState) {**

**case 0:**

**digitalWrite(greenLED, HIGH);**

**digitalWrite(yellowLED, LOW);**

**digitalWrite(redLED, LOW);**

**break;**

**case 1:**

**digitalWrite(greenLED, LOW);**

**digitalWrite(yellowLED, HIGH);**

**digitalWrite(redLED, LOW);**

**break;**

**case 2:**

**digitalWrite(greenLED, LOW);**

**digitalWrite(yellowLED, LOW);**

**digitalWrite(redLED, HIGH);**

**break;**

**}**

**}**

**TASK 4: TO DISPLAY WHICH LIGHT IS ON OR OFF – STATUS OF YOUR LIGHT**

**To display which light is on and off on an LCD screen, you'll need an LCD display and a few additional connections. Here's how you can modify your code to achieve this:**

**Components Needed:**

1. Arduino Board: Any compatible board (e.g., Arduino Uno).
2. LEDs: Red, yellow, and green LEDs.
3. Resistors: 220 ohm or 330 ohm for each LED.
4. Breadboard and Jumper Wires: For connections.
5. 16x2 LCD Display: With I2C module for easy interfacing.
6. I2C Module: Allows for simple connection to the LCD.

**Wiring Instructions:**

1. LEDs: Connect as per your original setup.
   * Red LED: Pin 11
   * Yellow LED: Pin 12
   * Green LED: Pin 13
2. I2C LCD Display:
   * Connect the I2C module to the LCD.
   * SDA (Data) to Arduino A4
   * SCL (Clock) to Arduino A5
   * VCC to 5V
   * GND to GND