

LESSON 4: LED TRAFFIC LIGHT

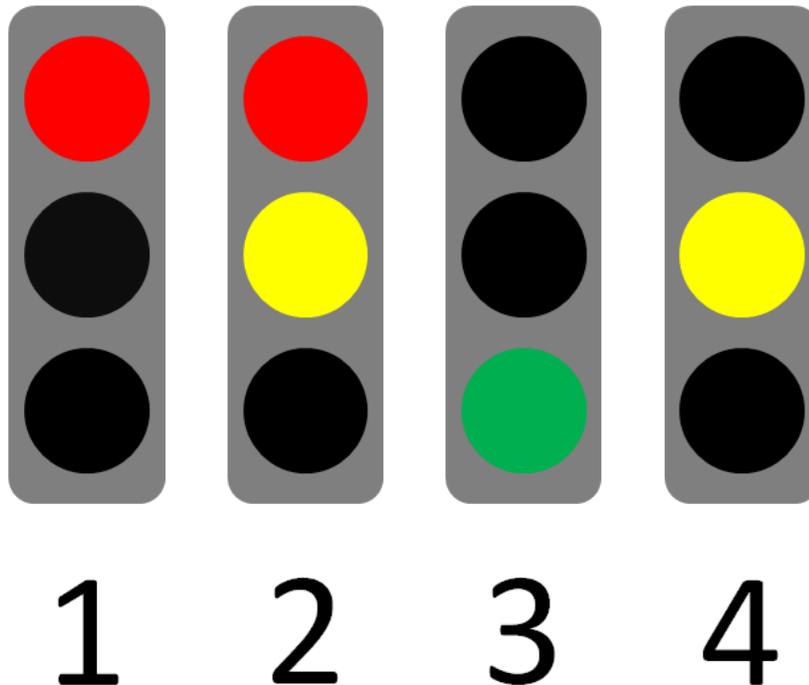
INTRODUCTION

In this tutorial we will take a look at building an Arduino traffic light circuit. This will involve using a little a bit of code and a very simple circuit to put together. This is a great beginner project if you have only just bought your first Arduino uno.

COMPONENTS

- Arduino Uno
- LED Red, Yellow and green
- 3x 330Ω resistors (Colour = Orange Orange Brown)
- Jumper wire
- Breadboard

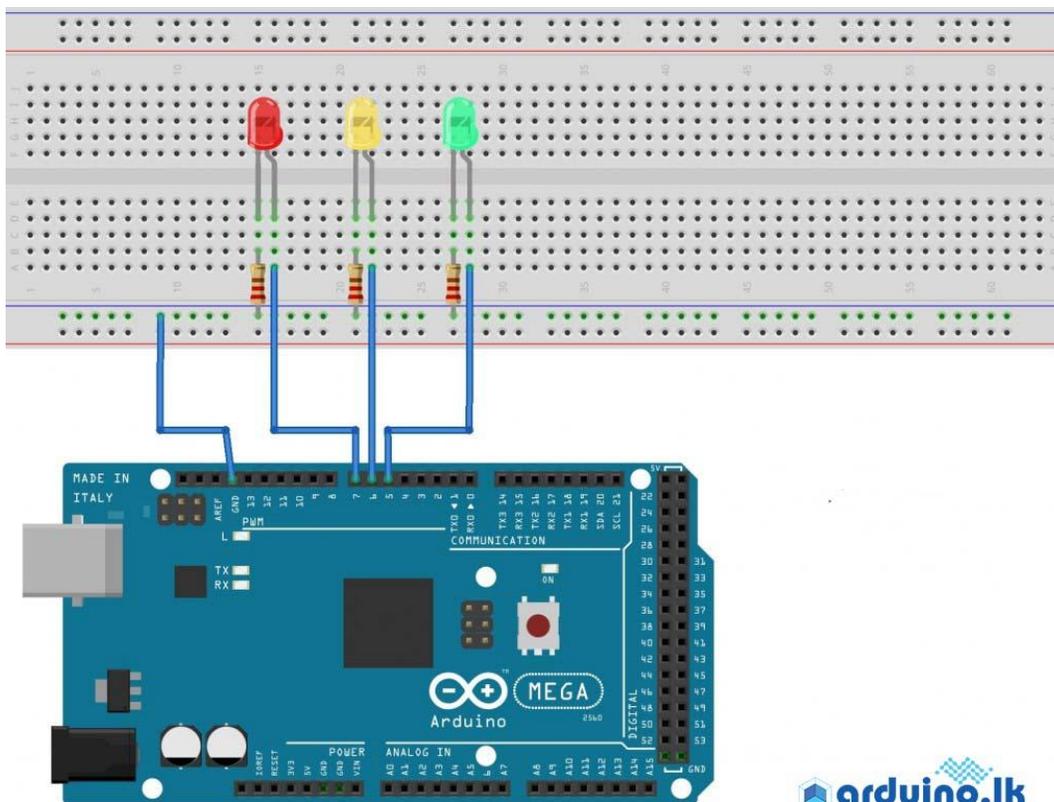
TRAFFIC LIGHTS TO WORK IN THIS WAY:



CONNECTION

STEP 1: Adding the LEDs.

1. Add a red, yellow, and green LED to the breadboard.
2. Connect the smallest lead (short end) of the LEDs to a 330 (Ohm) Resistor crossing the gap to the GND rail in the breadboard.
3. Connect the LED to the Arduino using a Jumper Wire.
 - Connect the Red LED to Pin 7 on the Arduino.
 - Connect the Yellow LED to Pin 6 on the Arduino.
 - Connect the Green LED to Pin 5 on the Arduino.
4. Finally, connect a Jumper Wire from the **GND** rail on the breadboard to the **GND** pin on the Arduino.



STEP 2: Program

```
// Projet 2 - Traffic Lights
int redLedDelay = 10000;
int greenLedDelay = 10000;
int redLed = 7;
int yellowLed = 6;
int greenLed = 5;
void setup(){
  pinMode(redLed, OUTPUT);
  pinMode(yellowLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
}
void loop(){

  digitalWrite(redLed, HIGH);
  delay(redLedDelay);

  digitalWrite(yellowLed, HIGH);
  delay(2000);

  digitalWrite(redLed, LOW);
  digitalWrite(yellowLed, LOW);
  digitalWrite(greenLed, HIGH);
  delay(greenLedDelay);

  digitalWrite(yellowLed, HIGH);
  digitalWrite(greenLed, LOW);
  delay(2000);

  digitalWrite(yellowLed, LOW);

}
```

STEP 3: Compile the code



The screenshot shows the Arduino IDE interface with the sketch 'led_traffic_light' open. The code is as follows:

```
// Projet 2 - Traffic Lights
int redLedDelay = 10000;
int greenLedDelay = 10000;
int redLed = 7;
int yellowLed = 6;
int greenLed = 5;
void setup(){
  pinMode(redLed, OUTPUT);
  pinMode(yellowLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
}
void loop(){

  digitalWrite(redLed, HIGH);
  delay(redLedDelay);

  digitalWrite(yellowLed, HIGH);
  delay(2000);

  digitalWrite(redLed, LOW);
  digitalWrite(yellowLed, LOW);
  digitalWrite(greenLed, HIGH);
  delay(greenLedDelay);

  digitalWrite(yellowLed, HIGH);
  digitalWrite(greenLed, LOW);
  delay(2000);

  digitalWrite(yellowLed, LOW);

}
```

At the bottom of the IDE, a progress bar indicates 'Compiling sketch...' is in progress. The status bar at the bottom shows '32' on the left and 'Arduino/Genuino Uno on COM3' on the right.

STEP 4: Upload the sketch to the Arduino Uno board.



The screenshot shows the Arduino IDE interface with the sketch 'led_traffic_light' being uploaded to an Arduino/Genuino Uno board. The code in the editor is as follows:

```
// Projet 2 - Traffic Lights
int redLedDelay = 10000;
int greenLedDelay = 10000;
int redLed = 7;
int yellowLed = 6;
int greenLed = 5;
void setup(){
  pinMode(redLed, OUTPUT);
  pinMode(yellowLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
}
void loop(){

  digitalWrite(redLed, HIGH);
  delay(redLedDelay);

  digitalWrite(yellowLed, HIGH);
  delay(2000);

  digitalWrite(redLed, LOW);
  digitalWrite(yellowLed, LOW);
  digitalWrite(greenLed, HIGH);
  delay(greenLedDelay);

  digitalWrite(yellowLed, HIGH);
  digitalWrite(greenLed, LOW);
  delay(2000);

  digitalWrite(yellowLed, LOW);

}
```

The status bar at the bottom indicates '32' and 'Arduino/Genuino Uno on COM3'. The console shows the upload progress and memory usage:

```
Uploading...
Sketch uses 1058 bytes (3%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes free.
```

RESULT:

The LEDs function same as the image of traffic light to work.