

- Create a set of traffic lights which change at set intervals
- Add a push button to start the light sequence



30 mins

## 1. Components list

In addition to your Arduino and breadboard, this project requires the following:



1 x momentary push switch



1 x red LEDs



1 x yellow LED



1 x green LEDs

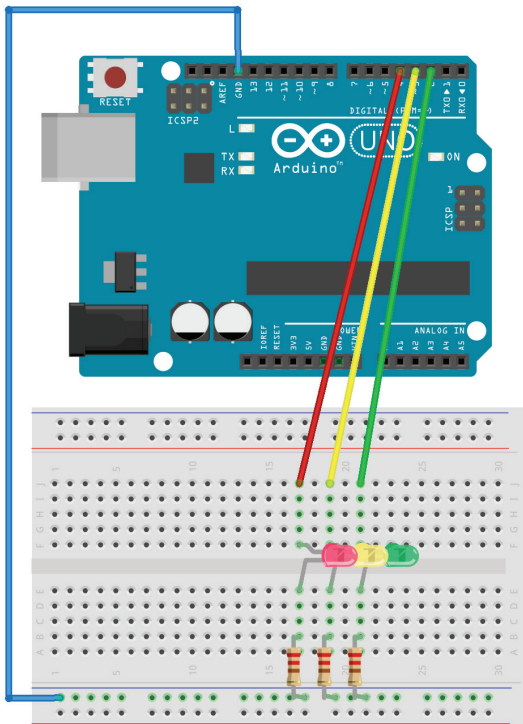


4 x 220 ohm resistors (red, red, brown)



6 x jumpers

## 2. Add the resistors



## 3. Start the code

Previously, we've referred to the LEDs by their pin number. Whilst this keeps our code small, it can be difficult to remember which pin is connected to which component.

To make the code easier for us to follow, we're going to store the pin numbers as *constants* with helpful names. Unlike a variable, constants cannot be changed by the program.

```
const int LED_red = 4; // Red LED, pin 4
const int LED_yellow = 3; // Yellow LED, pin 3
const int LED_green = 2; // Green LED, pin 2
```

```
void setup() {
  // Initialise pins 2 to 4 for output
  pinMode(LED_red, OUTPUT);
  pinMode(LED_yellow, OUTPUT);
  pinMode(LED_green, OUTPUT);
}
```

## 4. Analyse a traffic light sequence

Now that you have created your circuit and initialised your LEDs, it's nearly time to begin programming your light sequence. But do you know the correct order?

View this YouTube video to watch the sequence of most UK traffic lights:

- <http://bit.ly/trafficlightsequence>

Pelican crossings, which are operated by a pedestrian push-button, have a slightly different sequence. However, we're going to follow the usual sequence, but add a push-button control later.

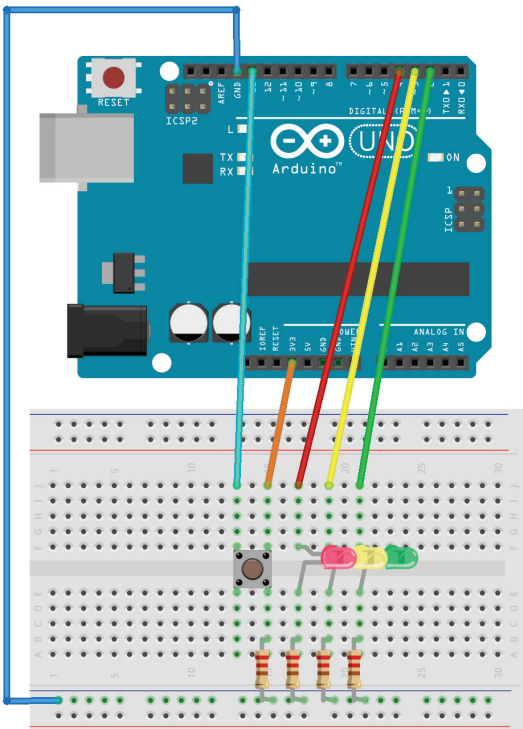
## 5. The loop

```
void loop() {  
  digitalWrite(LED_green, HIGH); // Turn on green LED  
  delay(4000); // Wait 4 seconds  
  digitalWrite(LED_green, LOW); // Turn off green LED  
  digitalWrite(LED_yellow, HIGH); // Turn on yellow LED  
  delay(1000); // Wait 1 second  
  digitalWrite(LED_yellow, LOW); // Turn off yellow LED  
  digitalWrite(LED_red, HIGH); // Turn on red LED  
  delay(4000); // Wait 4 seconds  
  digitalWrite(LED_yellow, HIGH); // Turn on yellow LED  
  delay(1000); // Wait 1 second  
  digitalWrite(LED_red, LOW); // Turn off red LED  
  digitalWrite(LED_yellow, LOW); // Turn off yellow LED  
  // Sequence ends and returns to the beginning of loop()  
}
```

Copy the `loop()` section of the program, as shown, then upload to the Arduino.

- Green LED lights-up for four seconds
- Green LED switches off
- Yellow LED lights-up for one second
- Yellow LED switches off
- Red LED lights-up for four seconds
- Yellow LED lights-up for one second
- Red and yellow LEDs switched off
- The loop begins again

## 6. Add the button



Add the following code with the other constants:

```
// Set up pin 13 for button  
const int buttonPress = 13;
```

Add the following code within the `setup()` function to define the pin mode for input:

```
pinMode(buttonPress, INPUT);
```

## 7. Button code

Modify your `loop()` function so that it is as shown below.

```
void loop() {  
  digitalWrite(LED_green, HIGH); // Turn on green LED  
  bool buttonState = digitalRead(buttonPress);  
  if (buttonState == true) {  
    // Beginning of traffic light sequence  
    digitalWrite(LED_green, LOW); // Turn off green LED  
    digitalWrite(LED_yellow, HIGH); // Turn on yellow  
    delay(1000); // Wait 1 second  
    digitalWrite(LED_yellow, LOW); // Turn off yellow  
    digitalWrite(LED_red, HIGH); // Turn on red LED  
    delay(4000); // Wait 4 seconds  
    digitalWrite(LED_yellow, HIGH); // Turn on yellow  
    delay(1000); // Wait 1 second  
    digitalWrite(LED_red, LOW); // Turn off red LED  
    digitalWrite(LED_yellow, LOW); // Turn off yellow  
    // If condition ends  
  }  
}
```

The traffic light now begins with the green LED lit. It remains lit until the button is pressed, at which point the `buttonState` value changes to `true` and the if condition is true, starting the sequence which changes the traffic light to red.

## 8. Challenge activities

- Add additional LEDs to represent the red and green pedestrian crossing persons
- Add more LEDs to represent traffic lights at a junction. Remember, there is a pause at which point all lights are red