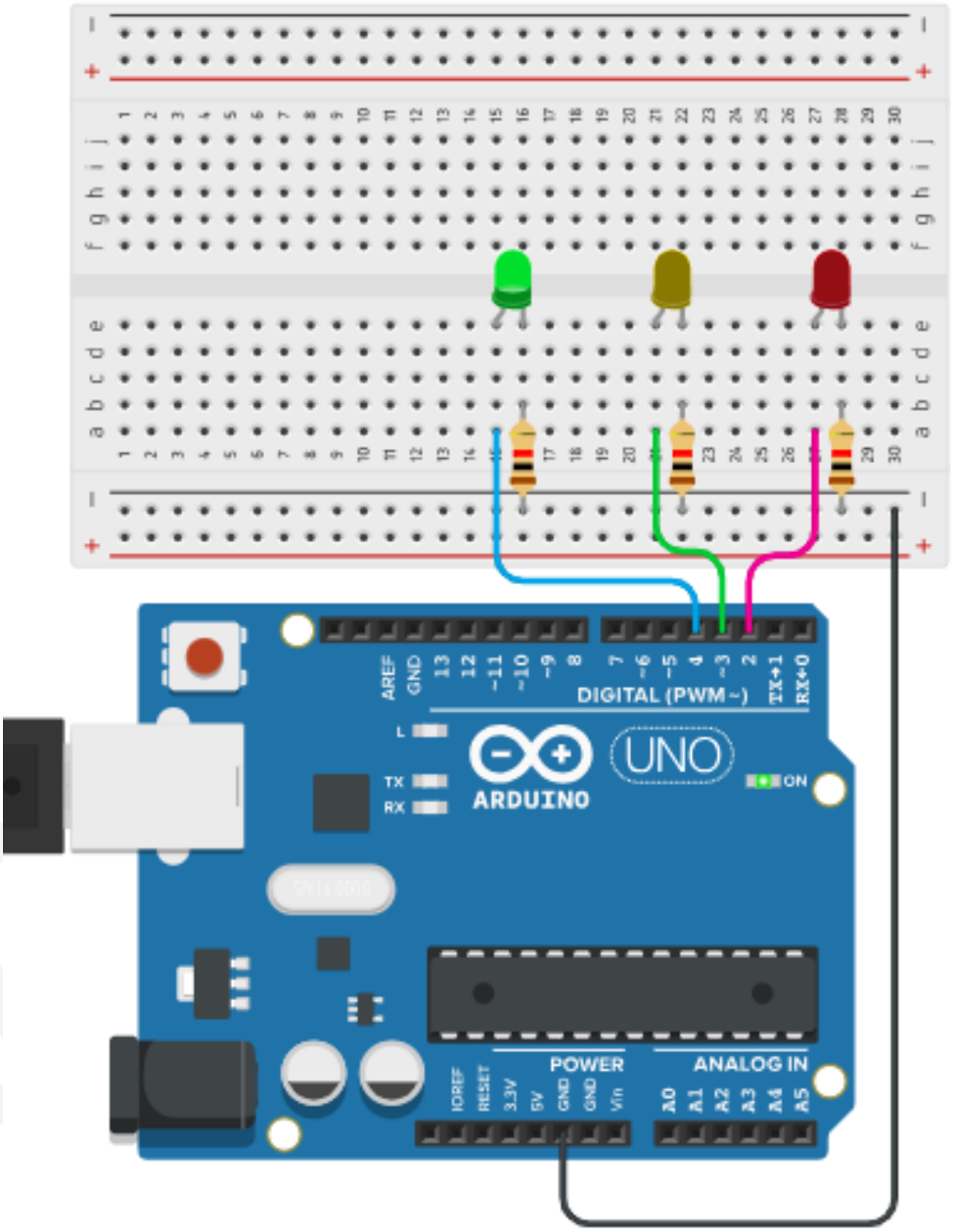
Activity: Make an Automatic Traffic Light System

Objective:

Traffic lights play a crucial role in regulating vehicle and pedestrian traffic at intersections to ensure safety and smooth traffic flow. Simulating traffic lights using LEDs and Arduino allows you to create a miniature version of these traffic control systems.

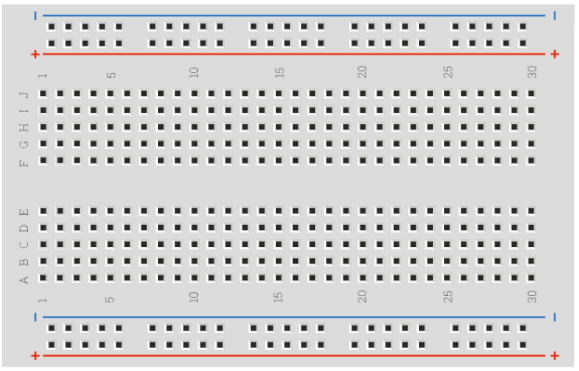
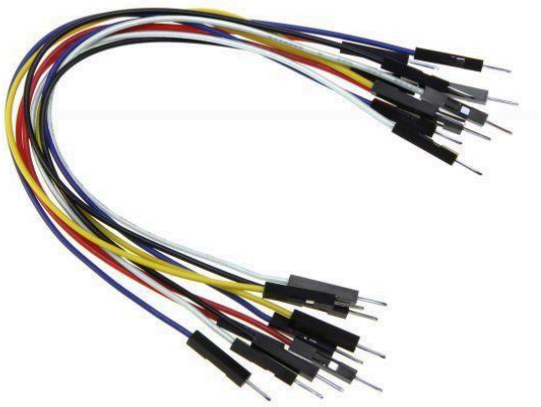


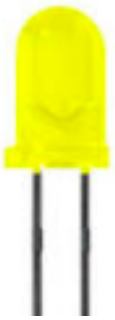
An automatic traffic light model using LEDs and Arduino is a fun and educational project that simulates a simplified version of real-world traffic signals. This project is an excellent way to learn about basic electronics, programming with Arduino, and how traffic lights work.

Building an automatic traffic light model using LEDs and Arduino is an engaging and informative project for both beginners and hobbyists. It provides a valuable learning experience in electronics and programming while allowing you to create a miniature traffic control system that you can proudly display.

So, in today's activity we will learn how to make an Automatic Traffic Light system using an Arduino and some LEDs and resistors.

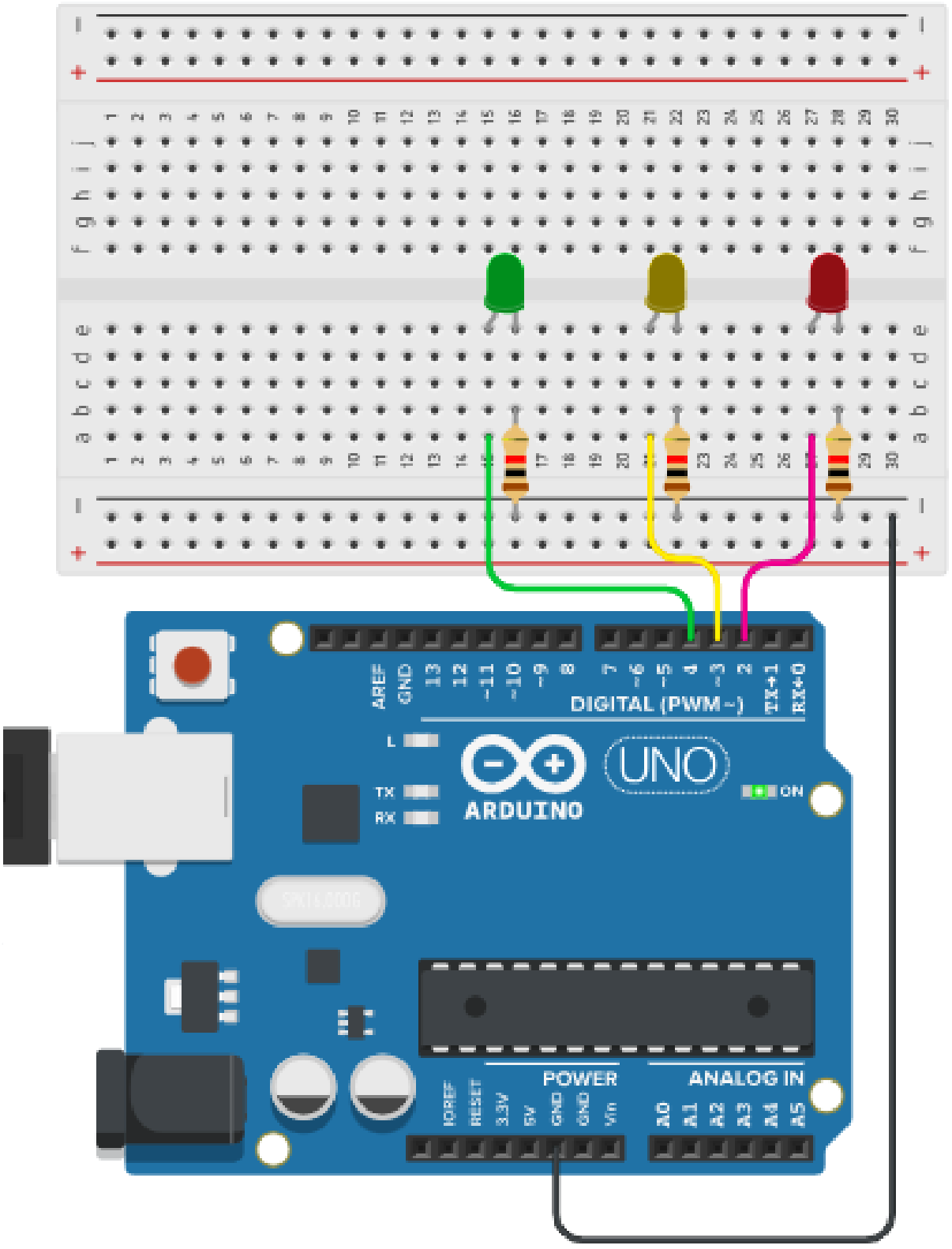
Materials Required:

S.no.	Part	Qty	Image
1	Arduino (Nano / UNO)	1	
2	Breadboard	1	
3	USB cable for Arduino	1	
4	Connection Wires (M - M)	5	

5	5mm Yellow LED	1	
6	5mm Red LED	1	
7	5mm Green LED	1	
8	1k resistor (Brown Black Red Golden)	3	

Connection Diagram:

The connection diagram shown in the image below is based on the Arduino UNO.



Explanation:

The connection diagram shown in the image shows the connections between the Arduino and the Output. The Red LED, the Yellow LED and the Green LED are the Outputs for the Arduino. The color of these lights has been chosen to match with the real life traffic signal colors.

When we carefully observe the connection diagrams, it becomes clear that:

1. The Red LED is connected such that its +ve terminal is connected to Arduino digital pin 2 (D2) and its -ve terminal is connected to one side of the 1k resistor. The other side of the 1k resistor is connected to the -ve of the breadboard. The Arduino - Red LED connection is shown by the **pink / red line** in the diagram above.

2. The Yellow LED is connected such that its +ve terminal is connected to Arduino digital pin 3 (D3) and its -ve terminal is connected to one side of the 1k resistor. The other side of the 1k resistor is connected to the -ve of the breadboard. The Arduino - Yellow LED connection is shown by the **yellow line** in the diagram above.

3. The Green LED is connected such that its +ve terminal is connected to Arduino digital pin 4 (D4) and its -ve terminal is connected to one side of the 1k resistor. The other side of the 1k resistor is connected to the -ve of the breadboard. The Arduino - Green LED connection is shown by the **green line** in the diagram above.

4. Finally, the Gnd pin of the Arduino is connected to the -ve of the breadboard. The Arduino is connected to the computer through the USB cable which serves for providing power, program upload and bi-directional serial data communication with the computer. This is shown by the Black line in the diagrams above.

Arduino Code:

Here is the complete Arduino code for making an Automatic Traffic Light System using some LEDs and an Arduino.

```
#define Red     2
#define Yellow  3
#define Green   4

void setup()
{
 pinMode(Red, OUTPUT);
 pinMode(Green, OUTPUT);
 pinMode(Yellow, OUTPUT);
}

void loop()
{
  digitalWrite(Red,LOW);
  digitalWrite(Green,HIGH);
  digitalWrite(Yellow,LOW);
  delay(6000);

  digitalWrite(Red,LOW);
  digitalWrite(Green,LOW);
  digitalWrite(Yellow,HIGH);
  delay(3000);

  digitalWrite(Red,HIGH);
  digitalWrite(Green,LOW);
  digitalWrite(Yellow,LOW);
  delay(9000);
}
```

Explanation:

```
#define Red     2
#define Yellow  3
#define Green   4
```

Here, we are declaring that:

1. Arduino digital pin 2 (D2) will be referred to as “Red” as it is used for the Red LED.
2. Arduino digital pin 3 (D3) will be referred to as “Yellow” as it is used for the Yellow LED.
3. Arduino digital pin 4 (D4) will be referred to as “Green” as it is used for the Green LED.

```
void setup()
{
 pinMode(Red, OUTPUT);
 pinMode(Green, OUTPUT);
 pinMode(Yellow, OUTPUT);
}
```

Here, inside the setup function, we are declaring that the “Red” pin, the “Yellow” pin and the “Green” pin will be all used as Output pins. This needs to be done just once.

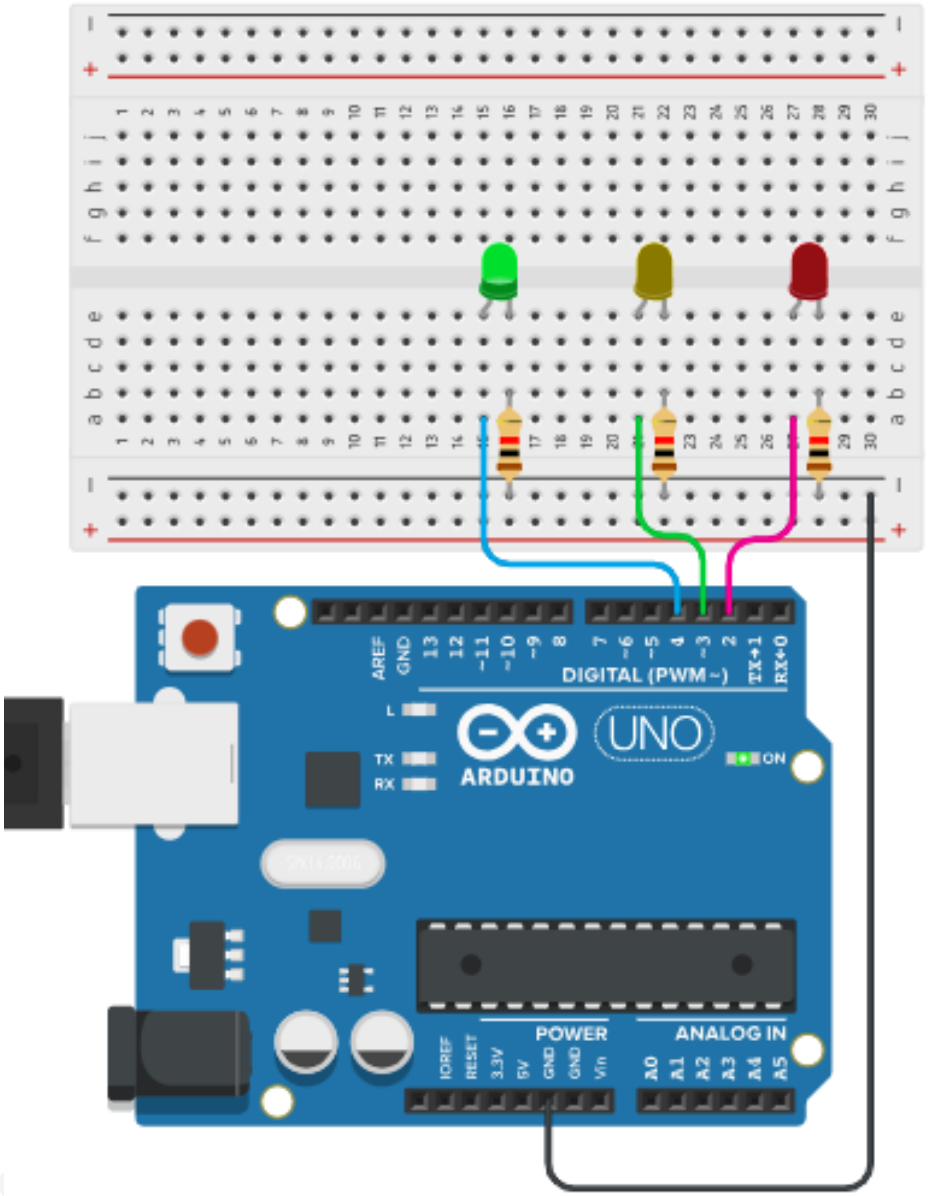
```
void loop()
{
  digitalWrite(Red,LOW);
  digitalWrite(Green,HIGH);
  digitalWrite(Yellow,LOW);
  delay(6000);

  digitalWrite(Red,LOW);
  digitalWrite(Green,LOW);
  digitalWrite(Yellow,HIGH);
  delay(3000);

  digitalWrite(Red,HIGH);
  digitalWrite(Green,LOW);
  digitalWrite(Yellow,LOW);
  delay(9000);
}
```

Here, in the loop function, first the Green LED is turned On and kept On for 6 seconds, then the Yellow LED is turned On and kept On for 3 seconds and finally, the Red LED is turned On and kept On for 9 seconds. Once done, the sequence repeats itself over and over again for as long as there is power.

Outcome and Observations:



1. Once the Arduino code is compiled and uploaded, the sequence starts with the Green LED turning On. Once turned On, the Green LED stays On for 6000 ms or 6 seconds.

2. Next, the Yellow LED turns On. Once turned On, the Yellow LED stays On for 3000 ms or 3 seconds.

3. Lastly, the Red LED turns On. Once turned On, the Red LED stays On for 9000 ms or 9 seconds.

4. The entire sequence repeats itself over and over again as long as power is made available to the setup.