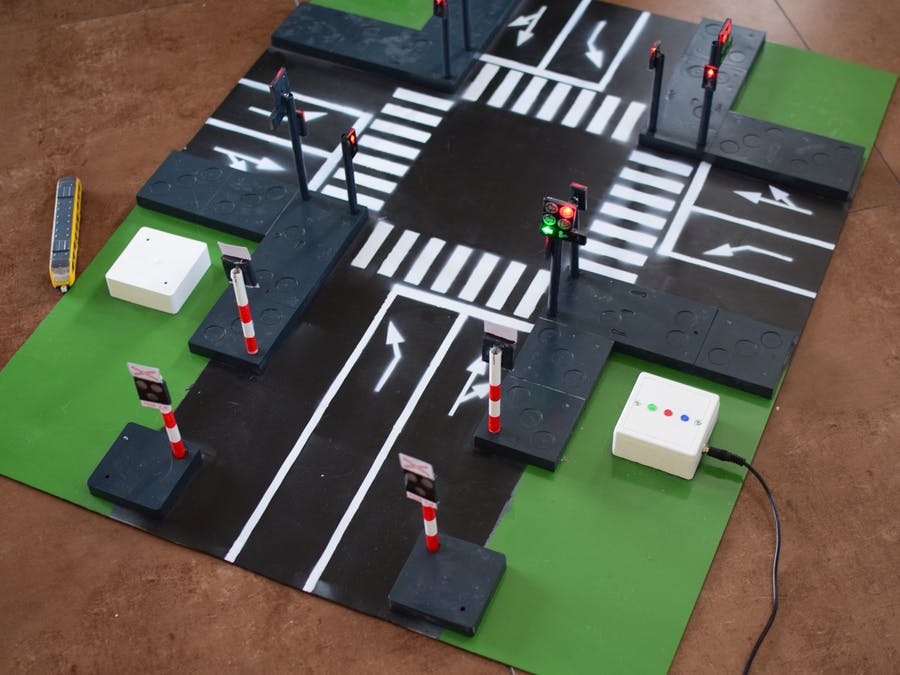
**Arduino control traffic light**

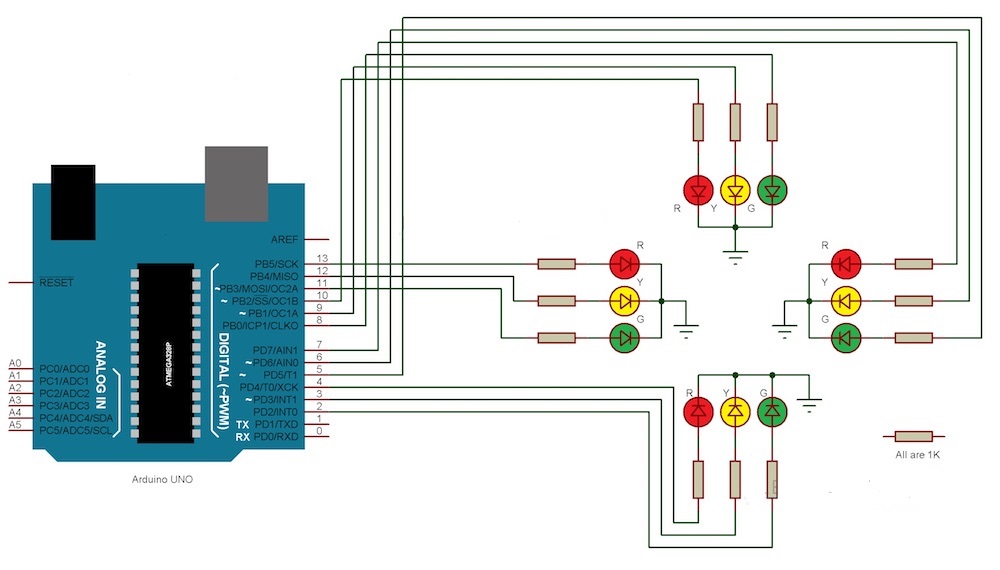
****

* **Introduction:** The use of personal vehicles is very common now a days and a result, the number of vehicles on the roads are exponentially increasing. Roads without any supervision or guidance can lead in to traffic congestions and accidents. Traffic Lights or Traffic Signals are signaling devices that are used to control the flow of traffic. Generally, they are positioned at junctions, intersections, ‘X’ roads, pedestrian crossings etc. and alternate the priority of who has to wait and who has to go. The traffic lights will provide instructions to the users (drivers and pedestrians) by displaying lights of standard color. The three colors used in traffic lights are Red, Yellow and Green. The system must be used to control the traffic lights for smooth and safe movement of traffic. These control systems consists of electro mechanical controllers with clockwork mechanisms or modern solid state computerized systems with easy setup and maintenance.

In this project, an Arduino based Traffic Light Controller system is designed. It is a simple implementation of traffic lights system but can be extended to a real time system with programmable timings, pedestrian lighting etc.

* **Materials :**
* Arduino UNO
* 1KΩ Resistor X 12
* Red LEDs X 4
* Yellow LEDs X 4
* Green LEDs X 4
* Connecting wires
* Prototyping board
* Power adapter
* **How to make :**

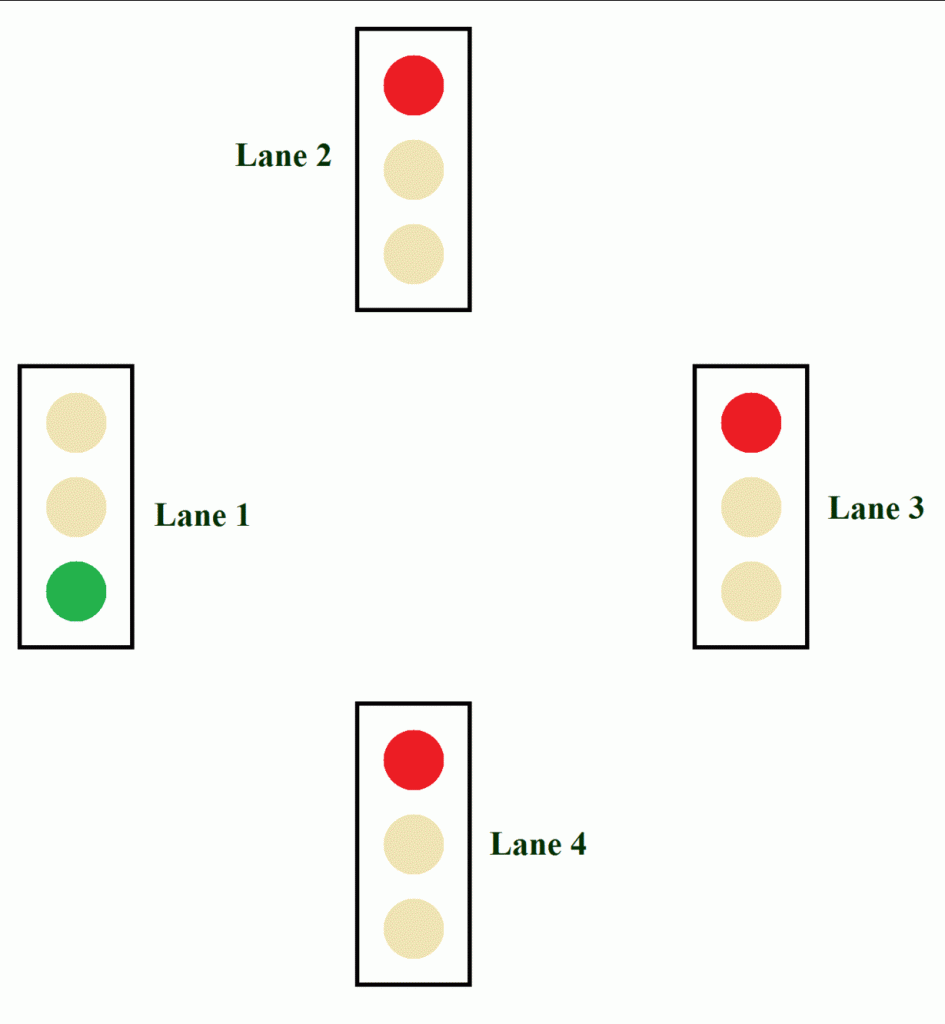
**Circuit Diagram**



* Since the project is a traffic light controller, the circuit consists of many LEDs (12 as a matter of fact) as we are implementing traffic lights at a 4 way intersection. The project is a simple representation of traffic light controller and hence no other extra components are used.
* We need three LEDs of Red, Yellow and Green colors at each intersection. The intersection is divided in to four lanes: Lane1, Lane 2 Lane 3 and Lane 4.
* All the LEDs are connected to the Arduino UNO’s digital I/O pins through respective current limiting resistors of 1KΩ.
* **Note**: In the practical implementation of the project, we did not use the current limiting resistors as the current from each digital I/O pin of the Arduino UNO is only 20mA. This small current will not burn the LED. But it is advised to use the current limiting resistors of at least 220 Ω in series with each LED.

Also note that Arduino UNO in this project acts as source of current for all the LED i.e. it provides the necessary current to turn ON the LED. Hence, a reliable power supply (like a DC adapter) to power the Arduino UNO must be used.

* **Working :**
* The real time traffic light controller is a complex piece of equipment which consists of power cabinet, main controller or processor, relays, control panel with switches or keys, communication ports etc.
* In this project, a simple traffic light system for a 4 way intersection is implemented using Arduino UNO. Although it is not the ideal implementation for real life scenarios, it gives an idea of the process behind the traffic light control system
* The aim of the project is to implement a simple traffic light controller using Arduino UNO, where the traffic is controlled in a pre-defined timing system. The working of the project is very simple and is explained below.
* Consider the following gif image showing a loop of traffic light operations. The project is also implemented in the same manner.



* In that, first the Lane 1 gets its Green light turned. Hence, in all the other Lanes, their corresponding Red lights are turned on. After a time delay of predefined time say 5 seconds, the Green light in the Lane 3 must be turned on and the Green light in the Lane 1 must be turned off.
* As a warning indicator, the Yellow light in Lane 1 is tuned on indicating that the red light is about to light up. Similarly, the yellow light in the Lane 3 is also turned as an indication that the green light about to be turned on.
* The yellow lights in Lanes 1 and 3 are turned for a small duration say 2 seconds after with the red light in the Lane 1 is turned on and green light in Lane 3 is also turned on.
* The green light in Lane 3 is also turned on for a predefined time and the process moves forward to Lane 4 and finally Lane 2.
* The system then loops back to Lane 1 where the process mentioned above will be repeated all over again.
* **Note:**The project implemented here doesn’t include the pedestrian crossing and pedestrian signaling in to consideration

**Code**

int Lane1[] = {13,12,11}; // Lane 1 Red, Yellow and Green

int Lane2[] = {10,9,8};// Lane 2 Red, Yellow and Green

int Lane3[] = {7,6,5};// Lane 3 Red, Yellow and Green

int Lane4[] = {4,3,2};// Lane 4 Red, Yellow and Green

void setup()

{

for (int i = 0; i < 3; i++)

{

pinMode(Lane1[i], OUTPUT);

pinMode(Lane2[i], OUTPUT);

pinMode(Lane3[i], OUTPUT);

pinMode(Lane4[i], OUTPUT);

}

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

}

void loop()

{

digitalWrite(Lane1[2], HIGH);

digitalWrite(Lane3[0], HIGH);

digitalWrite(Lane4[0], HIGH);

digitalWrite(Lane2[0], HIGH);

delay(7000);

digitalWrite(Lane1[2], LOW);

digitalWrite(Lane3[0], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane3[1], HIGH);

delay(3000);

digitalWrite(Lane1[1], LOW);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane1[0], HIGH);

digitalWrite(Lane3[2], HIGH);

delay(7000);

digitalWrite(Lane3[2], LOW);

digitalWrite(Lane4[0], LOW);

digitalWrite(Lane3[1], HIGH);

digitalWrite(Lane4[1], HIGH);

delay(3000);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane4[1], LOW);

digitalWrite(Lane3[0], HIGH);

digitalWrite(Lane4[2], HIGH);

delay(7000);

digitalWrite(Lane4[2], LOW);

digitalWrite(Lane2[0], LOW);

digitalWrite(Lane4[1], HIGH);

digitalWrite(Lane2[1], HIGH);

delay(3000);

digitalWrite(Lane4[1], LOW);

digitalWrite(Lane2[1], LOW);

digitalWrite(Lane4[0], HIGH);

digitalWrite(Lane2[2], HIGH);

delay(7000);

digitalWrite(Lane1[0], LOW);

digitalWrite(Lane2[2], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane2[1], HIGH);

delay(3000);

digitalWrite(Lane2[1], LOW);

digitalWrite(Lane1[1], LOW);

}

**ALL DONE !**