SPARSH GLOBAL SCHOOL, GREATER NOIDA

Project Title:

Traffic Light System Using Arduino

NAME:

CLASS:

Introduction

Traffic lights are an essential part of modern road infrastructure, regulating the flow of vehicles and pedestrians at intersections. They ensure safety, reduce traffic congestion, and provide a systematic way to manage road usage. In this project, we will design a basic traffic light system using an Arduino microcontroller, which will control three LEDs to simulate the red, yellow, and green signals of a traffic light.

This project serves as an excellent introduction to working with microcontrollers, basic electronics, and programming. By building this system, you will gain practical experience in controlling outputs (LEDs) with a microcontroller and understanding timing sequences that mimic real-world applications.

Traffic light systems are not just limited to simple on/off operations; they are designed to manage complex scenarios such as pedestrian crossings, emergency vehicle detection, and adaptive control based on traffic conditions. While this project focuses on a basic implementation, it lays the groundwork for more advanced traffic management systems.

By the end of this project, you will have a solid understanding of how traffic lights operate, how to interface electronic components with an Arduino, and how to write code to control them. This knowledge is fundamental to many areas of embedded systems, automation, and robotics.

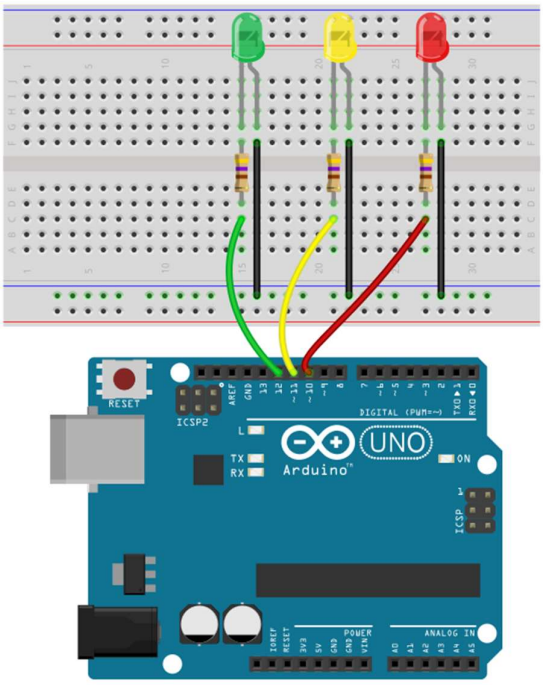
Components Required

- Arduino Uno board
- Red, Yellow, and Green LEDs
- 220-ohm resistors (3 pieces)
- Breadboard
- Jumper wires
- USB cable (for programming the Arduino)

Circuit Diagram

The circuit consists of connecting the red, yellow, and green LEDs to the Arduino board. Each LED will be connected in series with a 220-ohm resistor to limit the current. The anode (longer leg) of each LED will be connected to digital pins on the Arduino, and the cathode (shorter leg) will be connected to the ground (GND).

- Red LED: Digital Pin 10
- Yellow LED: Digital Pin 11
- Green LED: Digital Pin 12



All of these LEDs connect to these digital pins through a 470Ω resistor.

The cathodes, or negative terminals of the LEDs, connect to ground.

These completes all the hardware connections to the circuit.

How the circuit works is that we can control which LED is on because each LED is connected to a separate digital pin on the arduino. If we make a digital pin HIGH, that LED is on, while the rest are written LOW and so are off. With the software code, we can decide how long each LED is on for. This is explained in detail below.

Code

The code to simulate a traffic light circuit is shown below.

```
const int greenLED= 12;
const int yellowLED= 11;
const int redLED= 10;
void setup() {
pinMode (greenLED, OUTPUT);
pinMode (yellowLED, OUTPUT);
pinMode (redLED, OUTPUT);
}
void loop(){
digitalWrite (greenLED, HIGH);
digitalWrite (yellowLED, LOW);
digitalWrite (redLED, LOW);
delay(15000);
digitalWrite (greenLED, LOW);
digitalWrite (yellowLED, HIGH);
digitalWrite (redLED, LOW);
delay(2000);
digitalWrite (greenLED, LOW);
digitalWrite (yellowLED, LOW);
digitalWrite (redLED, HIGH);
delay(15000);

}
```

Here in this code, we first define the pin connections. Since the green LED is connected to digital pin 12, we set it equal to 12. Since the yellow LED is connected to digital pin 11, we set it equal to 11. Since the red LED is connected to digital pin 10, we set it equal to 10.

The next block of lines, the setup() function, declare all the LEDs as outputs. We are not reading any value from the LED. We are writing either a HIGH or LOW value to the LED. Therefore, the LEDs must be outputs, not inputs.

The next function, the loop() function, is the heart of our code. This block of code loops over and over and over again, repeatedly, without end.

In the first section, we write the greenLED HIGH while all the others are LOW. We call a delay of 15000 which is equal to 15 seconds. This causes the green LED to be on for 15 seconds, while all the other LEDs are off.

In the second section, we write the yellowLED HIGH while all the others are LOW. We call a delay of 2000, which is equal to 2 seconds. This causes the yellow LED to be on for 2 seconds, while the other LEDs are off.

In the third section, we write the redLED HIGH while all the others are LOW. We call a delay of 15000, which is equal to 15 seconds. This causes the red LED to be on for 15 seconds, while all the other LEDs are off.

Again, you can change the time the LEDs are on for by changing the value in the delay() function. Since the parameter in the delay() function is measured in milliseconds (ms), delay(1000) is 1 second. So if you want to have the LED on for 10 seconds, for example, the parameter would 1000 * 10= 10000. So the delay function would be delay(10000). If you want the LED on for 30 seconds, the parameter would be 1000 * 30= 30000. So the delay function would be delay(30000).

And this is how a traffic light can be simulated with an arduino microcontroller.

Applications

The Traffic Light System project using Arduino is a foundational experiment that demonstrates how basic electronic components and programming concepts can be applied to real-world scenarios. This project can be adapted and expanded into various applications, ranging from educational tools to more advanced traffic management systems. Below are some of the key applications:

1. Educational Tools

- **STEM Education:** This project serves as an excellent introduction to STEM (Science, Technology, Engineering, and Mathematics) education. It is particularly useful for teaching students about the principles of electronics, coding, and automation. The simplicity of the project makes it ideal for beginners, while its potential for expansion makes it suitable for more advanced learners as well.
- **Classroom Demonstrations:** Teachers can use this project as a demonstration to explain concepts like sequencing, timing, and control systems. It can also be integrated into curricula focused on microcontrollers, embedded systems, or automation technologies.

2. Prototype Development

- **Traffic Management Simulation:** The basic traffic light system can be used as a prototype for simulating traffic management in small-scale urban planning projects. Urban planners and engineers can use it to visualize how traffic lights would work in a new intersection or a redesigned road layout.

- **Smart Traffic Lights:** This project can be expanded to include sensors and communication modules to create a smart traffic light system. Such systems can adjust the timing of the lights based on real-time traffic conditions, reducing congestion and improving traffic flow efficiency.

3. Pedestrian Safety Systems

- **Pedestrian Crossing Lights:** By adding buttons or sensors, this project can be modified to simulate pedestrian crossing lights. The system would allow pedestrians to request a green light for safe crossing, and the Arduino would manage the sequence to ensure both vehicles and pedestrians are safely managed.
- **School Zone Safety:** A more advanced version of this project could be implemented in school zones where traffic lights need to be controlled based on specific times of the day, ensuring the safety of children during school hours.

4. Home Automation and Robotics

- **Automation Projects:** Traffic light systems are a great way to introduce concepts of automation in home projects. For example, the sequencing logic used in traffic lights can be applied to home automation tasks, such as controlling lights, fans, or other appliances in a sequence.
- **Robotics:** This project's principles can be extended to robotics, where similar sequencing logic might be required to control different components of a robot, such as actuators, sensors, or even communication between different robots in a swarm.

5. Public Awareness and Training

- **Driver Education:** Traffic light simulators based on this project can be used in driver education programs to teach new drivers about traffic signals and the importance of following them. Interactive versions could allow users to test their reactions to traffic lights in various simulated traffic conditions.
- **Public Safety Campaigns:** Traffic light models can be used in public safety campaigns to educate the public about the importance of obeying traffic signals, particularly in areas with high pedestrian traffic or where accidents frequently occur.

6. Research and Development

- **Urban Mobility Studies:** Researchers can use this traffic light system as part of larger studies on urban mobility, traffic flow optimization, and the impact of traffic signal timing on congestion and air quality.
- **IoT and Smart Cities:** With the integration of Internet of Things (IoT) technologies, this basic traffic light system can be developed into a smart traffic management system that communicates with other IoT devices in a smart city framework. This could include vehicle-to-infrastructure communication, adaptive signal control, and data collection for urban planning.

7. Entertainment and DIY Projects

- **DIY Hobby Projects:** Hobbyists and DIY enthusiasts can use the traffic light system as a starting point for more creative

projects, such as integrating it with model railroads, slot car tracks, or as part of a home-made arcade game.

- **Interactive Exhibits:** This project can be used to create interactive exhibits in museums or science fairs, where visitors can learn about traffic lights and control the system themselves through buttons or sensors.

THANK YOU