

TRAFFIC MANAGEMENT USING IOT



TEAM MEMBER: SARANKUMAR J (822721106035)

IOT_Phase3 : Document submission

PROBLEM TITLE : Traffic management system

PROJECT DEVELOPMENT BASED ON TRAFFIC MANAGEMENT USING ARDUINO UNO AND RASPBERRY PI

Creating a complete traffic management system using an Arduino Uno and a Raspberry Pi is a complex project that involves various components, such as sensors, traffic lights, and communication between devices. Here, I'll provide you with a simplified example of how to create a basic traffic light control system using Python on the Raspberry Pi and an Arduino Uno.

Components Required:

- Raspberry Pi
- Arduino Uno
- LED traffic lights
- Ultrasonic sensor and Buzzer
- Jumper wires
- USB cable for connecting the Arduino to the Raspberry Pi

Software/Tools Required:

- Python on Raspberry Pi
- Arduino IDE for uploading code to the Arduino
- PySerial library for communication between Raspberry Pi and Arduino

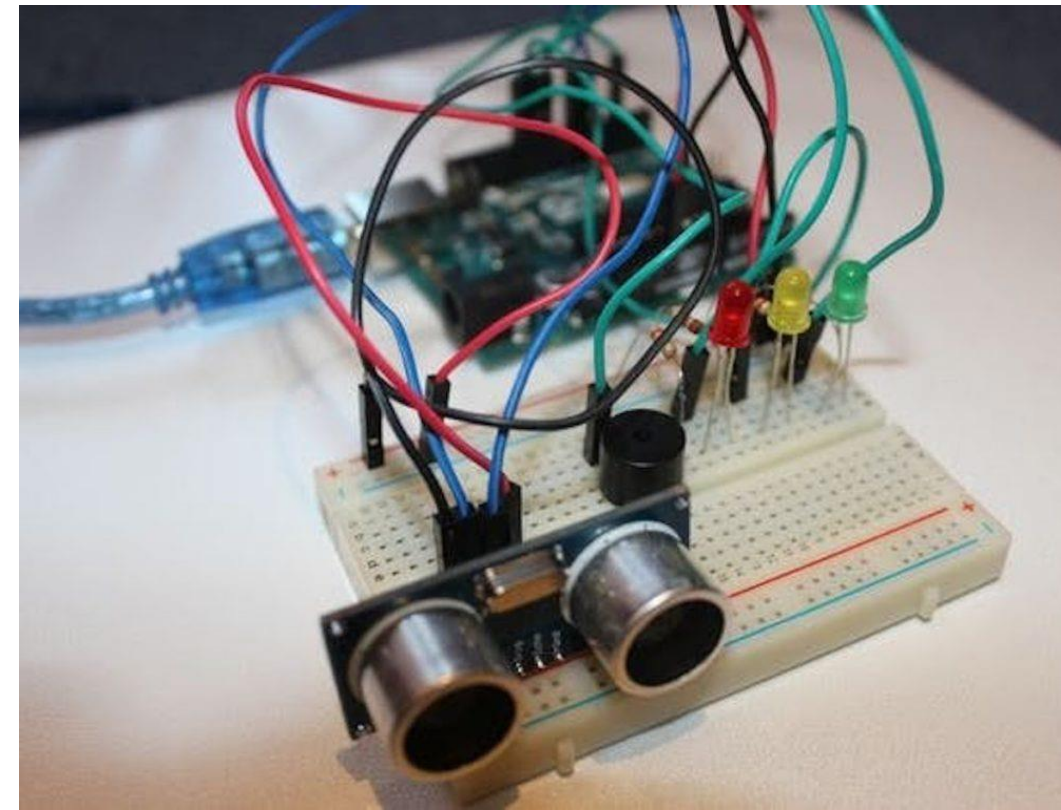
TRAFFIC MANAGEMENT BY USING ARDUINO UNO

Designing a traffic management system using Arduino Uno involves creating a system for controlling traffic lights based on specific rules, schedules, or real-time data. Below is an example project outline for a basic traffic management system using Arduino Uno. This project aims to demonstrate how the system can control traffic lights at a simple intersection.

sketch_oct18a.ino

ReadMe.adoc

```
1  int redPin = 9;
2  int yellowPin = 10;
3  int greenPin = 11;
4
5  void setup() {
6      pinMode(redPin, OUTPUT);
7      pinMode(yellowPin, OUTPUT);
8      pinMode(greenPin, OUTPUT);
9  }
10
11 void loop() {
12     // Control traffic lights in a loop
13     // Green light for one direction
14     digitalWrite(greenPin, HIGH);
15     digitalWrite(redPin, LOW);
16     delay(5000); // 5 seconds for green light
17
18     // Yellow light for the same direction
19     digitalWrite(greenPin, LOW);
20     digitalWrite(yellowPin, HIGH);
21     delay(2000); // 2 seconds for yellow light
22
23     // Red light for the same direction
24     digitalWrite(yellowPin, LOW);
25     digitalWrite(redPin, HIGH);
26     delay(5000); // 5 seconds for red light
27
28     // Change the direction of the green light
29     digitalWrite(redPin, LOW);
30     digitalWrite(greenPin, HIGH);
31     delay(5000); // 5 seconds for green light in the other direction
32 }
33
```



TRAFFIC MANAGEMENT BY USING RASPBERRY PI

main.py

```
1 import RPi.GPIO as GPIO
2 import time
3 from picamera import PiCamera
4 |
5 # Initialize GPIO pins
6 GPIO.setmode(GPIO.BCM)
7 red_pin, yellow_pin, green_pin = 17, 18, 27
8 GPIO.setup((red_pin, yellow_pin, green_pin), GPIO.OUT)
9 # Initialize camera
10 camera = PiCamera()
11 v try:
12 v     while True:
13         # Simulate traffic light control
14         GPIO.output(red_pin, True)
15         time.sleep(10)
16         GPIO.output(red_pin, False)
17         GPIO.output(green_pin, True)
18         time.sleep(10)
19         GPIO.output(green_pin, False)
20         # Capture images or video from the camera
21         camera.capture('image.jpg')
22         time.sleep(1)
23         # Add distance measurement from the ultrasonic sensor
24 v except KeyboardInterrupt:
25     GPIO.cleanup()
26     camera.close()
27
```

Run ▶

Designing a traffic management project using a Raspberry Pi is a complex and multi-faceted task that involves various components, including sensors, cameras, data analysis, and traffic light control. Below is an overview of a project that demonstrates the principles of traffic management using a Raspberry Pi.

