

TRAFFIC MANAGEMENT SYSTEM

➤ LITERATURE REVIEW:

The traffic management system fulfills its duty by enabling the smooth movement of vehicle and it also has a fail-safe system which will prove useful in unexpected circumstances, deploying IoT devices, such as traffic flow sensors and cameras, to monitor traffic conditions requires careful planning and execution.

1. TMS HARDWARE TOOLS:

(a) ARDUINO

- ✓ By using Arduino Uno as the main controller for data collection and processing

(b) CAMERA

- ✓ If you want to capture visual data, connect a camera to the Arduino Uno

(c) GPS MODULE

- ✓ Attach a GPS module to the Arduino to track the location of the monitoring device

(d) INTERNET CONNECTIVITY

- ✓ Ensure the Arduino Uno has internet access.

2. TMS SOFTWARE TOOLS:

(a) PYTHON

- ✓ Develop Python scripts to collect, process, and send traffic data. We can use libraries like 'requests', 'gpsd', and 'picamera'.

(b) DATA COLLECTION

- ✓ Use the GPS module to retrieve location data. Capture the images or videos from the camera if needed. Collect other relevant data such as speed, timestamp, and environmental conditions.

3. PROCESSING AND MONITORING:

Test the system thoroughly in a controlled environment. Deploy the monitoring device in a real-world location. Monitor the system performance and troubleshoot any issues that arise.

set up a system for remote monitoring and maintenance of the devices. This can include remote diagnostics and ability to update the device firmware.

4.COMPLIANCE AND REGULATIONS:

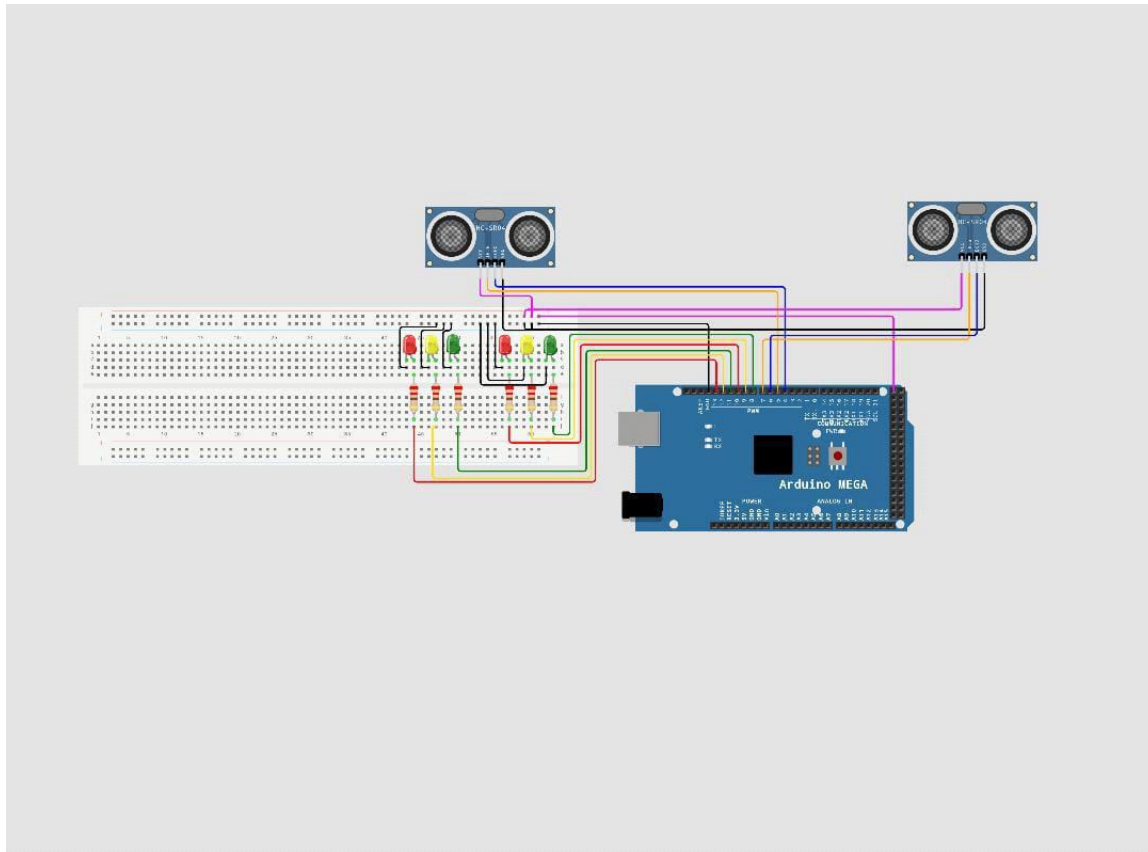
Ensure the deployment complies with local regulations, privacy laws, and data protection requirements.

5.TMS TOOLS OBJECTIVES:

- arduino uno,
- ultrasonic sensors,
- GPS module,
- camera.

6.SYSTEM METHODOLOGY:

Test system architecture involves the Arduino Uno as the central controller, which interfaces with the camera and ultrasonic sensors. The Arduino Uno collects the data from these sensors, processes it, and sends the data to the ThingSpeak platform over Wi-Fi. ThingSpeak displays the output in graphical representation.



7.SOURCE CODE

```
#include <WiFi.h>

#include <DHTesp.h>

#include <Ultrasonic.h>

#include <ThingSpeak.h>

const char*ssid="wokwi-GUEST";

const char*password="";

const unsigned long channel ID=2315218;

const char*writeAPIkey="WJLKFCP438EX4NG2";

python

import time

import request

#configure ultrasonic sensor pins

TRIG_PIN=23

ECHO_PIN=24

#simulated ultrasonic sensor data

def get_simulated distance():

return 60

#ThingSpeak configuration

THINGSPEAK_API_KEY='your_api_key'

THINGSPEAK_URL='https://api.thingspeak.com/update'

try:

while true:

#distance=get_distance()

#distance=get_simulate_distance()
```

```

#send data to thingspeak

payload={'api_key':THINGSPEAK_API_KEY,'field1':distance}

response=request.post(THINGSPEAK_URL,parms=payload)

#intialize GPIO settings

GPIO.setmode(GPIO.BCM)

GPIO.setup(TRIG_PIN,GPIO.OUT)

GPIO.setup(ECHO_PIN,GPIO.IN)

def get_distance()

GPIO.output(TRIG_PIN,True)

time.sleep(0.00001)

GPIO.output(TRIG_PIN,False)

while GPIO.input(ECHO_PIN)==1:

pulse_end=time.time()

pulse_duration=pulse_end-pulse_start

distance=(pulse_duration*3400)/2

return distance

try:

while true:

distance=get_distance()

if response.status_code==200:

print("DISTANCE:{distance}cm-data send to thingspeak")

else:

print("failed to send data to thingspeak")

time.sleep(60)

except KeyboardInterrupt:

```

GPIO.clean up()

GPIO on exit

8.OPERATION:

Import required libraries.define thingspeak parameters.intialize GPIO settings.

(a)LOOP

- ✓ The program enters a infinite loop,contineously checking for the presence of vehcile using motion sensors

(b)TIMING

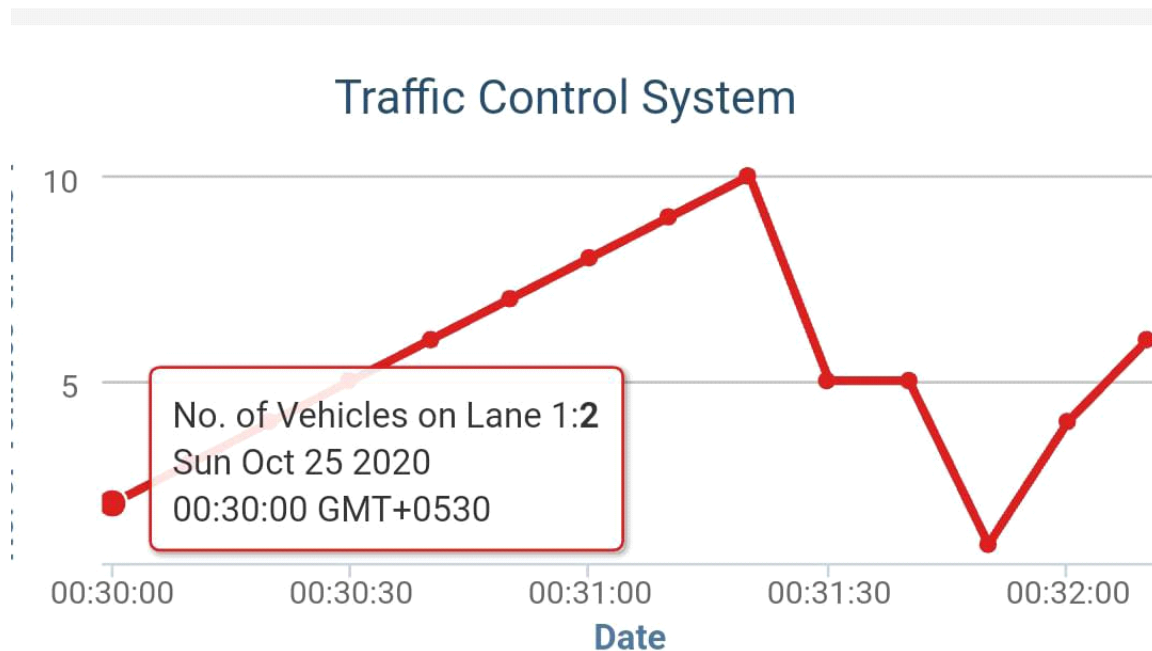
- ✓ The program then waits forn 2 seconds before repeating the process,checking for vehicle presence again.

(c)TERMINATION

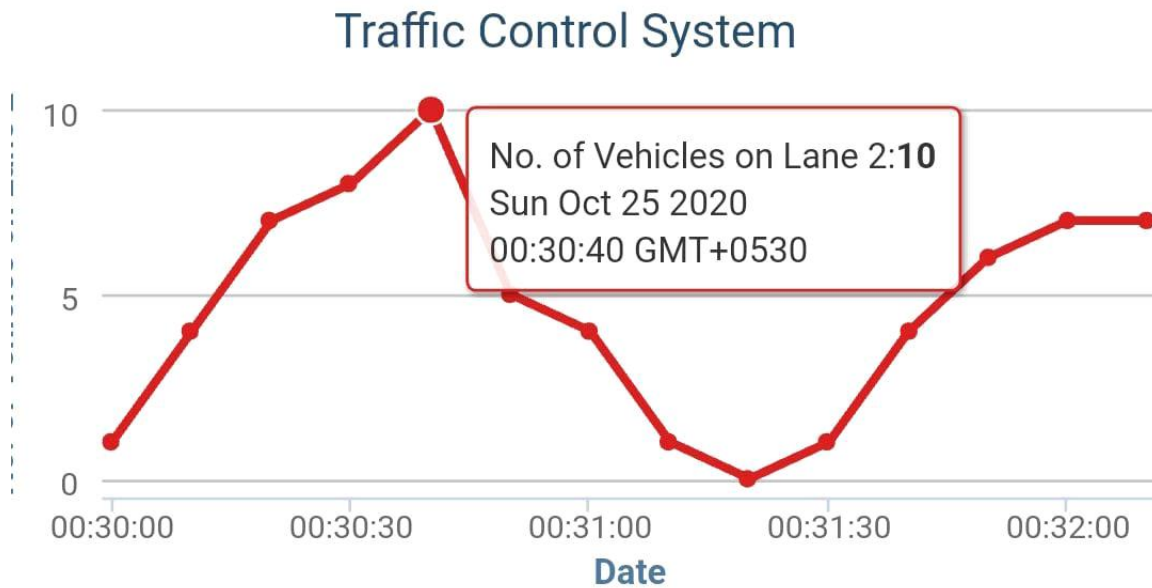
- ✓ To stop the program,you can press ctrl+c,which trigger a keyboard interrupt,allowing for GPIO cleanup and existing the program.

9.DIGITAL OUTPUT:

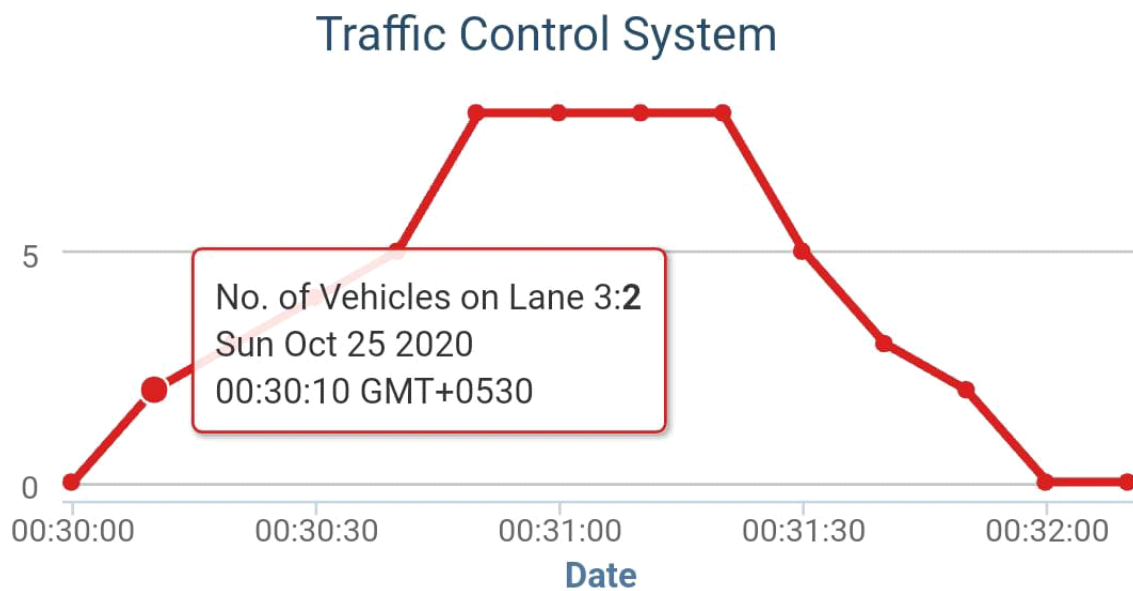
(a)FIELD CHART 1



(b)FIELD CHART 2

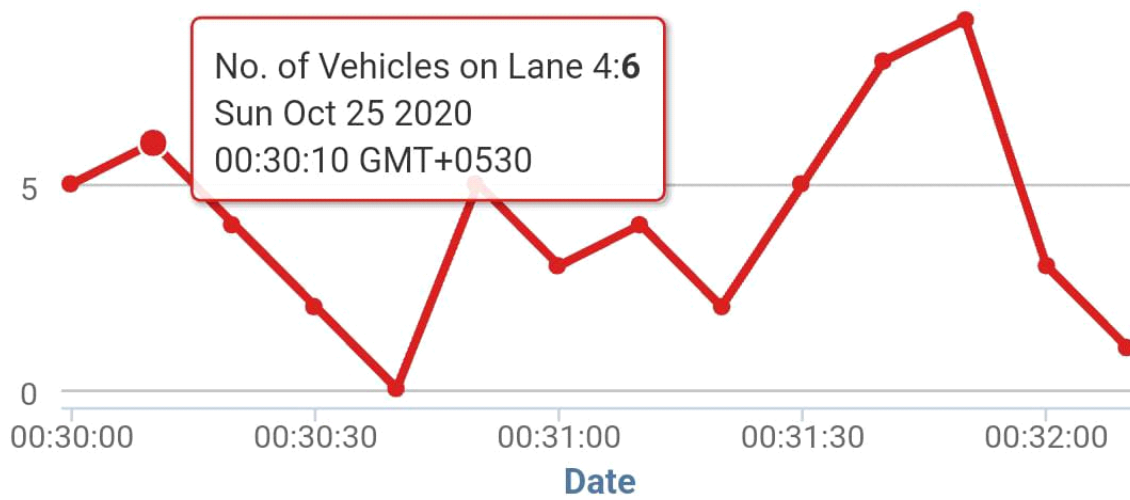


(c)FIELD CHART 3



(d)FIELD CHART4

Traffic Control System



10.WOKWI SIMULATION OUTPUT:

11.CONCLUSION:

In conclusion, the provided python program demonstrates a basic traffic management system that stimulates the operation of traffic lights based on the detection of vehicle presence using a motion sensors.the program also communicate with the TingSpeak platform to record and monitor traffic condition in simple mannner.