TRAFFIC MANAGEMEMT SYSTE­­­­M

* **LITERACTURE REVIEW:**

The traffic management system fulfills its duty by enabling the smooth movement of vechicle and it also has afail-safe system which will prove usefull 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)AURDINO**

* By useing aurdino uno as the main controller for data collection and processing

**(b)CAMERA**

* If you want to capture vissual data,cannect a camera to the aurdino uno

(c)GPS MODULE

* Attach a GPS module to the aurdino to track the location of the monitoring device

(d)INTERNET CONNECTIVITY

* Ensure the aurdino 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 'request','gpsd',and 'picamera'.

(b)DATA COLLECTION

* Use the GPS module to retrive location data.capture the images or videos from the camera if needed.collect other revelent data such as speed,timestamp and environmental conditions.

**3.PROCESSING AND MONITORING:**

Test the system throughly in a controlled environment.deploy the monitoring device in a real world location.monitor the system performance and troubleshoot any issuses that arises.

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

**4.COMPLIANCE AND REGULATIONS:**

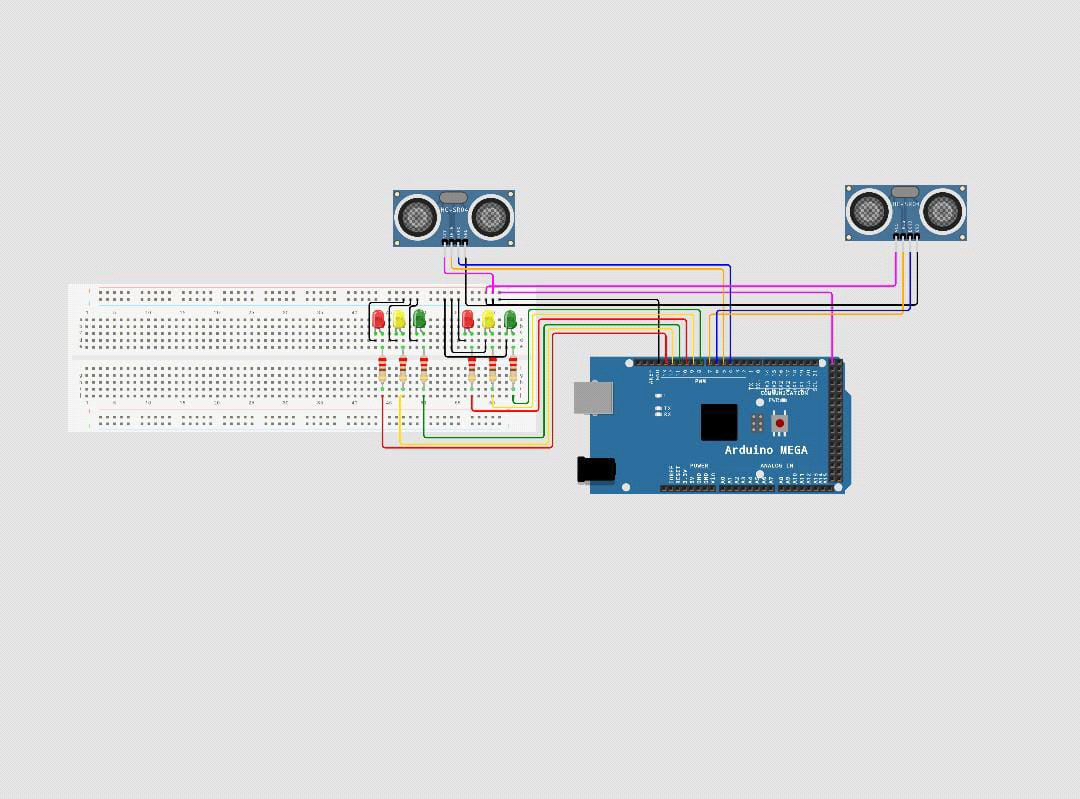
Ensure the deployment complies with local regulations ,privacy laws,and data protection requirements.

**5.TMS TOOLS OBJECTIVES:**

* aurdino uno,
* ultrasonic sensors,
* GPS module,
* camera.

**6.SYSTEM METHODOLOGY:**

Test system architecture involves the aurdino uno as the cental controller.which interface with the camera and ultrasonic sensors.the aurdino uno collects the data from these sensors,processes it,and sends the datas to the ThingSpeak platform over wi-fi,ThingSpeak display the output in graphical repersentation.



**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 distace():**

**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 vechicle using motion sensors

**(b)TIMING**

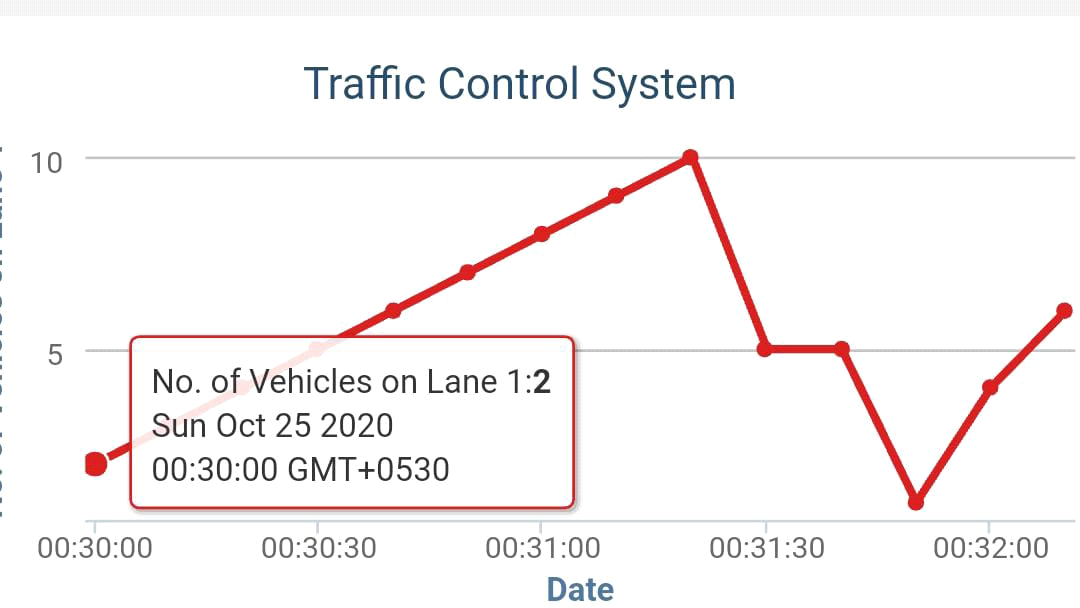
* The program then waits forn 2 seconds before repeating the process,checking for vehicle presence again.

**(c)TERMINATION**

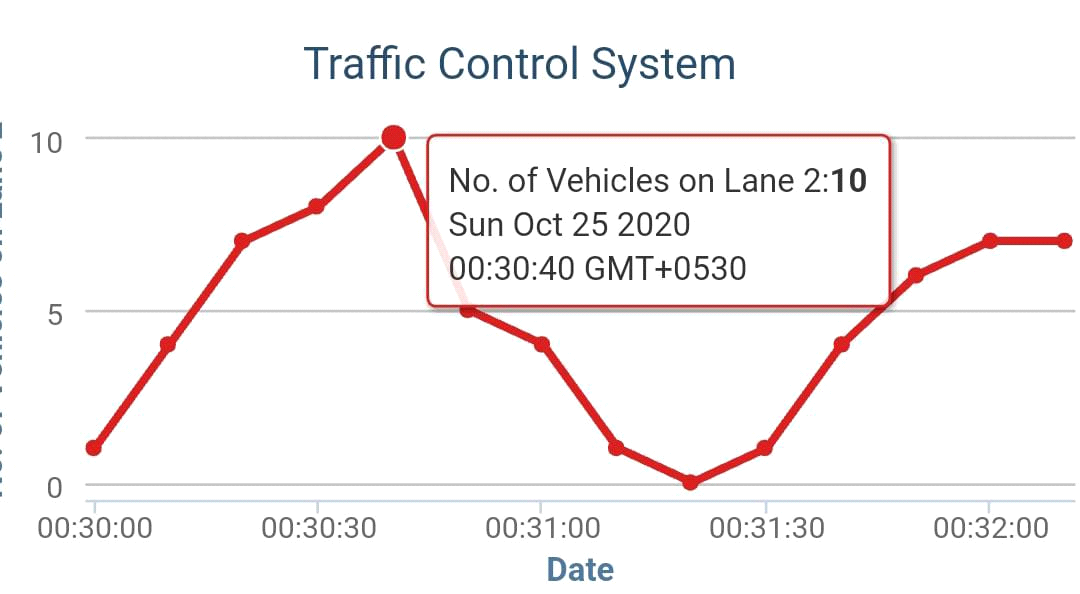
* To stop the program,you can press ctrl+c,which trigger a keyboard interrupt,allowing for GPIO cleanup and existing the program.

**9.DIGITAL OUTPUT:**

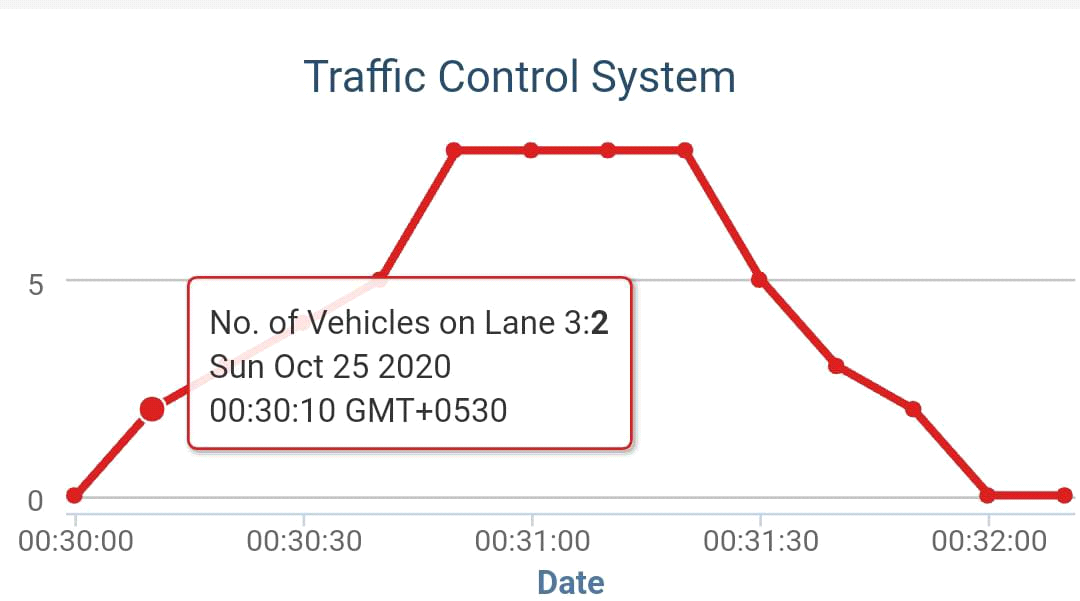
**(a)FIELD CHART 1**

****

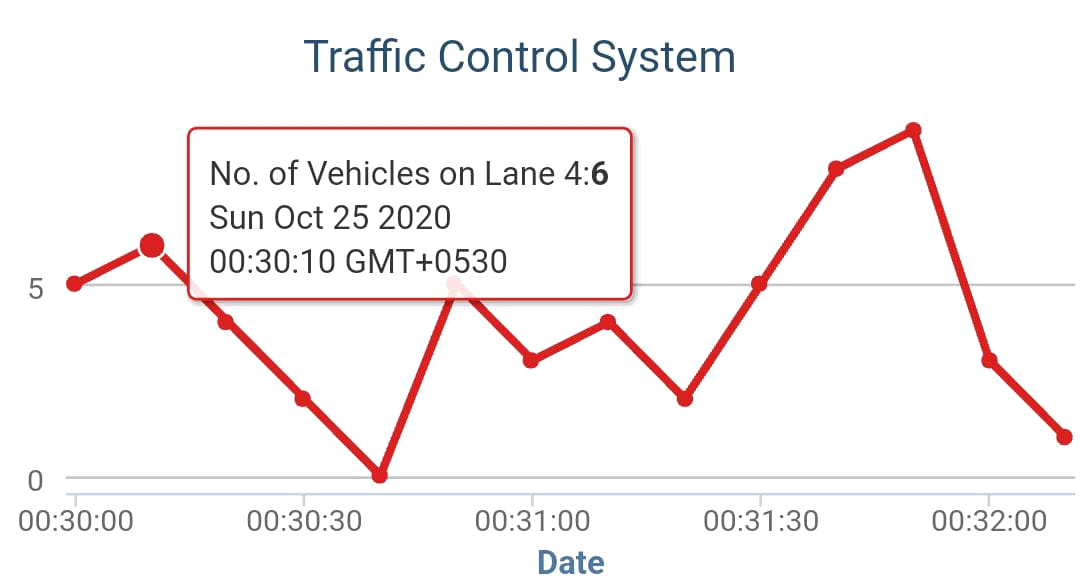
**(b)FIELD CHART 2**

****

**(c)FIELD CHART 3**

****

**(d)FIELD CHART4**

****

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 vechicle presence using a motion sensors.the program also communicate with the TingSpeak platform to record and monitor traffic condition in simple mannner.