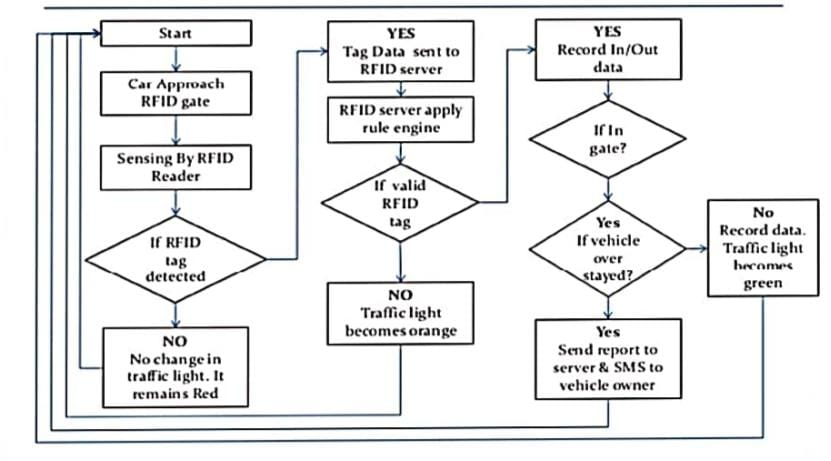
|  |
| --- |
| **Submitted by,** |
|  |
| DHANALAKSHMI S  au812921106007  karthigaqueen85@gmail.com |
|  |
|  |
|  |
|  |
|  |

**SMART TRAFFIC MANAGEMENT**

Introduction

There are various problems of traditional vehicle monitoring system in different conditions, such as license plate is blocked. Therefore, it is not easy for prosecuted the offender by the police force due to the different traffic situations. Thus, a new method is that combining those to achieve the traffic violation detection system, and RFID technology is used to aid some main functions. In this study, it is assumed that all the noise factor which affected RFID system is ignored and all the data get from the RFID system should be appropriated. It is aimed that recording the RFID tag's sensitivity signal within a small range for the use of getting the data of the vehicles when it violates from a simulated real road situation. Since a real-time transportation situation has been simulated, RFID tags as an electronic license plate, a fixed camera is being set for vehicles monitoring and RFID reader place above the road for receiving the vehicle's data through the tags and using OpenCV for motion detection of vehicles in the traffic violation detection. Thus, it can be prosecuted the offender by the police force due to the different traffic situation. The reader used in this study is RF-CODE 433 MHz M250. RF Code tags sent the radio frequency messages to the reader. The maximum range is 45 meters when the tags are being sensed, more than 100 tags are read per second.

Block diagram



Explanation

There are various problems of traditional vehicle monitoring system in different conditions, such as license plate is blocked.

Therefore, it is not easy for prosecuted the offender by the police force due to the different traffic situations. Thus, a new method isthat combining those to achieve the traffic violation detection system, and RFID technology is used to aid some main functions. Inthis study, it is assumed that all the noise factor which affected RFID system is ignored and all the data get from the RFID systemshould be appropriated. It is aimed that recording the RFID tag’s sensitivity signal within a small range for the use of getting thedata of the vehicles when it violates from a simulated real road situation. Since a real-time transportation situation has beensimulated, RFID tags as an electronic license plate, a fixed camera is being set for vehicles monitoring and RFID reader place abovethe road for receiving the vehicle’s data through the tags and using OpenCV for motion detection of vehicles in the traffic violationdetection. Thus, it can be prosecuted the offender by the police force due to the different traffic situation. The reader used in thisstudy is RF-CODE 433 MHz M250. RF Code tags sent the radio frequency messages to the reader. The maximum range is 45meters when the tags are being sensed, more than 100 tags are read per second.

**Working:** Place the automatic self-control model car with RFID tags is attached on the top in the corridor of Industrial Laboratoryof the University of Macau, as shown in Fig. 3. For more accurate data, it is necessary to place the model car on the road on thesame starting point. Once the vehicles reach the end of the road model and then repeat the same operations of getting the RFIDtag’s data. Then the RFID tags sensitivity range between the start and the end of the road model can be obtained.

Program:

import RPi.GPIO as GPIO

from time import sleep

hallpin1=8

#LED1=8

hallpin2=10

hallpin3=12

#hallpin4=24

hallpin11=22

hallpin12=24

hallpin13=26

hallpin21=38

hallpin22=40

hallpin23=37

hallpin31=31

hallpin32=29

hallpin33=23

LED1=16

LED2=18

LED11=32

LED12=36

LED21=35

LED22=33

LED31=21

LED32=19

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

GPIO.setup(LED1, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(LED2, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(hallpin1, GPIO.IN)

#GPIO.setup(LED2, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(hallpin2, GPIO.IN)

GPIO.setup(hallpin3, GPIO.IN)

GPIO.setup(LED11, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(LED12, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(hallpin11, GPIO.IN)

GPIO.setup(hallpin12, GPIO.IN)

GPIO.setup(hallpin13, GPIO.IN)

GPIO.setup(LED21, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(LED22, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(hallpin21, GPIO.IN)

GPIO.setup(hallpin22, GPIO.IN)

GPIO.setup(hallpin23, GPIO.IN)

GPIO.setup(LED31, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(LED32, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(hallpin31, GPIO.IN)

GPIO.setup(hallpin32, GPIO.IN)

GPIO.setup(hallpin33, GPIO.IN)

while True:

print("-----------------------------")

if(GPIO.input(hallpin1)==True):

# GPIO.output(LED1, GPIO.HIGH)

a1=1

print("magnet 1")

print("detected")

if(GPIO.input(hallpin1)==False):

a1=0

print("magnet 1")

print("not detected")

if(GPIO.input(hallpin2)==True):

a2=1

print(" magnet 2")

print(" detected")

if(GPIO.input(hallpin2)==False):

a2=0

print(" magnet 2")

print("not detected")

if(GPIO.input(hallpin3)==True):

a3=1

print(" magnet 3")

print(" detected")

if(GPIO.input(hallpin3)==False):

a3=0

print("magnet 3")

print(" not detected")

print("---------------------------------")

if(GPIO.input(hallpin11)==True):

b1=1

print("magnet 11")

print("detected")

if(GPIO.input(hallpin11)==False):

b1=0

print(" magnet 11")

print(" not detected")

if(GPIO.input(hallpin12)==True):

b2=1

print(" magnet 12")

print(" detected")

if(GPIO.input(hallpin12)==False):

b2=0

print(“magnet 12")

print(" not detected")

if(GPIO.input(hallpin13)==True):

b3=1

print(" magnet 13")

print(" detected")

if(GPIO.input(hallpin13)==False):

b3=0

print(" magnet 13")

print(" not detected")

print("------------------------------

if(GPIO.input(hallpin21)==True):

c1=1

print(" magnet 21")

print(" detected")

if(GPIO.input(hallpin21)==False):

c1=0

print("magnet 21")

print("not detected")

if(GPIO.input(hallpin22)==True):

c2=1

print("magnet 22")

print("detected")

if(GPIO.input(hallpin22)==False):

c2=0

print(" magnet 22")

print("not detected")

if(GPIO.input(hallpin23)==True):

c3=1

print("magnet 23")

print(" detected")

if(GPIO.input(hallpin23)==False):

c3=0

print("magnet 23")

print("not detected")

print("-------------------------------")

if(GPIO.input(hallpin31)==True):

d1=1

print("

magnet 31")

print("

detected")

if(GPIO.input(hallpin31)==False):

d1=0

print("

magnet 31")

print("

not detected")

if(GPIO.input(hallpin32)==True):

d2=1

print("

magnet 32")

print("

detected")

if(GPIO.input(hallpin32)==False):

d2=0

print("

magnet 32")

print("

not detected")

if(GPIO.input(hallpin33)==True):

d3=1

print("

magnet 33")

print("

detected")

if(GPIO.input(hallpin33)==False):

d3=0

print("

magnet 33")

print("

not detected")

sum1=a1+a2+a3

sum2=b1+b2+b3

sum3=c1+c2+c3

sum4=d1+d2+d3

print(sum1)

print(sum2)

print(sum3)

print(sum4)

f1=0

f2=0

f3=0

f4=0

if(f1==1)and(f2==1)and(f3==1)and(f4==1):

f1=0

f2=0

f3=0

f4=0

if(f1==0):

if(sum1>sum2)and(sum1>sum3)and(sum1>sum4):

GPIO.output(LED1, GPIO.HIGH)

GPIO.output(LED12, GPIO.HIGH)

GPIO.output(LED22, GPIO.HIGH)

GPIO.output(LED32, GPIO.HIGH)

sleep(15)

GPIO.output(LED1, GPIO.LOW)

GPIO.output(LED12, GPIO.LOW)

GPIO.output(LED22, GPIO.LOW)

GPIO.output(LED32, GPIO.LOW)

f1=1

if(f2==0):

if(sum2>sum1)and(sum2>sum3)and(sum2>sum4):

GPIO.output(LED11, GPIO.HIGH)

GPIO.output(LED2, GPIO.HIGH)

GPIO.output(LED22, GPIO.HIGH)

GPIO.output(LED32, GPIO.HIGH)

sleep(15)

GPIO.output(LED11, GPIO.LOW)

GPIO.output(LED2, GPIO.LOW)

GPIO.output(LED22, GPIO.LOW)

GPIO.output(LED32, GPIO.LOW)

f2=1

if(f3==0):

if(sum3>sum1)and(sum3>sum2)and(sum3>sum4):

GPIO.output(LED21, GPIO.HIGH)

GPIO.output(LED2, GPIO.HIGH)

GPIO.output(LED12, GPIO.HIGH)

GPIO.output(LED32, GPIO.HIGH)

sleep(15)

GPIO.output(LED21, GPIO.LOW)

GPIO.output(LED2, GPIO.LOW)

GPIO.output(LED12, GPIO.LOW)

GPIO.output(LED32, GPIO.LOW)

f3=1

if(f4==0):

if(sum4>sum1)and(sum4>sum2)and(sum4>sum3):

GPIO.output(LED31, GPIO.HIGH)

GPIO.output(LED2, GPIO.HIGH)

GPIO.output(LED12, GPIO.HIGH)

GPIO.output(LED22, GPIO.HIGH)

sleep(15)

GPIO.output(LED31, GPIO.LOW)

GPIO.output(LED2, GPIO.LOW)

GPIO.output(LED12, GPIO.LOW)

GPIO.output(LED22, GPIO.LOW)

f4=1

sleep(2)