JAI SHRIRAM ENGINEERING COLLEGE
TIRUPPUR – 638 660
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
Recognized by UGC & Accredited by NAACandNBA (CSE and ECE)

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

IBM - Naan Mudhalvan

Internet of Things

Group 3

Phase 3 - Development part 1

Tittle: Traffic Management

NAME :E.MONISHA

NM ID :AU711221106017

YEAR : III

INTRODUCTION:

A Traffic Management System (TMS) through the Internet of Things (IOT) is a sophisticated system designed to monitor, control, and optimize traffic flow and transportation infrastructure using IOT technologies. It involves the integration of various IOT devices, sensors, and communication networks to collect real-time data from roadways, intersections, and vehicles. This data is then processed and analyzed to make informed decisions, improve traffic flow, enhance safety ,And reduce congestion on roads.

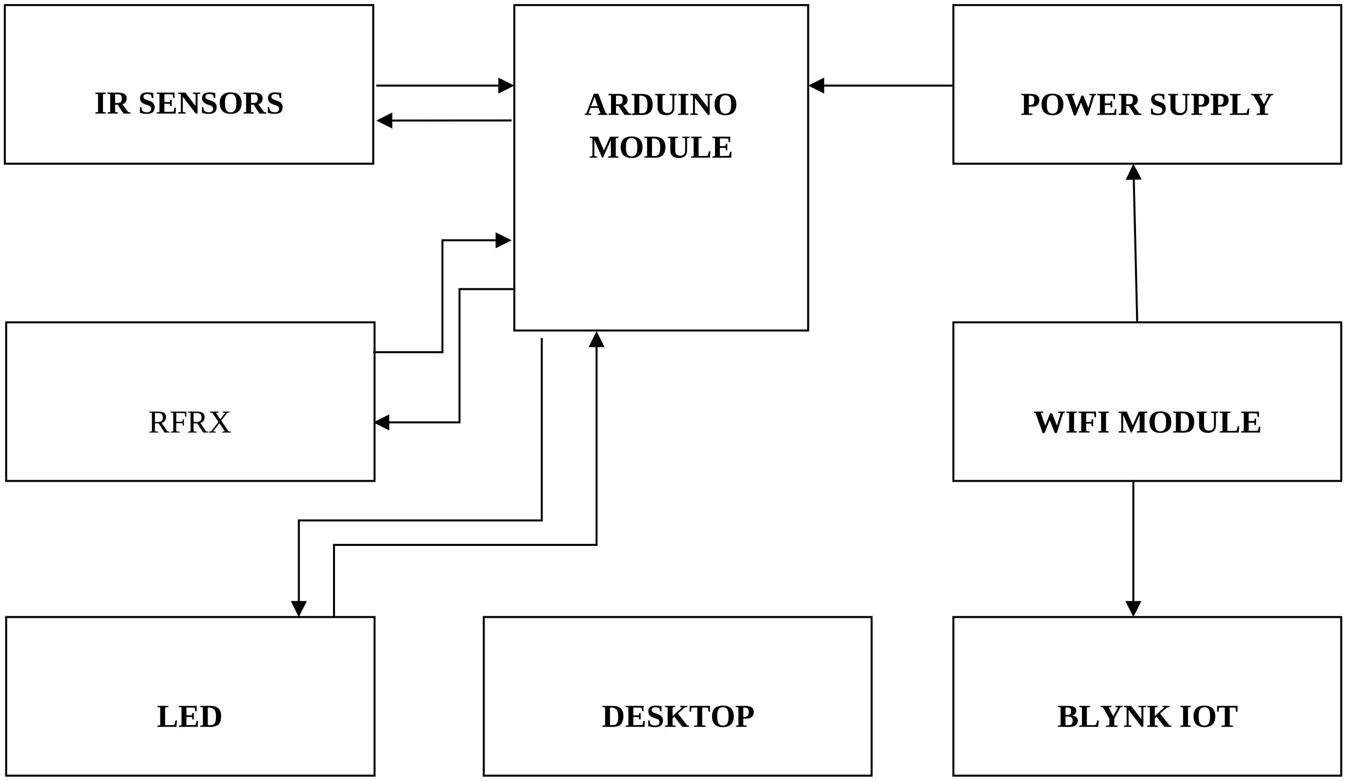
EXISTING SYSTEM:

The existing System is generally controlled by traffic police. The main drawback of our system which is controlled by traffic police is that the system is not smart enough to deal with the traffic congestion. The traffic police official can either block a road for more amount of time or let the vehicle on the other road pass by i.e. the decision making may not be smart enough and it entirely depends on the official's decision.

DEVELOPMENT IDEA:

- The first and primary element of this system is the wireless sensor nodes consists of sensors. The sensors interact with the physical environment means vehicles presence or absence while the local server sends the sensors data to the central micro controller.
- The micro controller receives the signal sent from the sensors and computes which road or which lane has to be chosen based on the density of traffic or no of car.

BLOCK DIAGRAM:



Implementation of project :

There will 8 sensors across the 4 lanes with each lane having 2 sensors each, to give the data how much dense the lane is. If in case the entire lanes have less traffic then the system will work normally means the lanes sequence will be first A lane then B lane, then C lane and at last D lane. But in case if any of the lane gets more cars or gets denser then the other 3 lanes then that specific lane will open then the other with the second highest denser, then the same order continued to the other 2 lanes. If in case all the lanes have same number of vehicles then all the lanes will open in same order i.e. A, B, C, D. If in case there is an emergency vehicle that can be ambulance, fire brigade or an emergency scout team then we have a separate Wi-Fi module through which we can open can lane in which the emergency vehicle is arrived.

PROGRAM:

```
ledA1 = 27
ledA2 = 25
ledA3 = 23
ledB1 = 51
ledB2 = 49
ledB3 = 47
ledC1 = 11
ledC2 = 12
ledC3 = 13
ledD1 = 8
ledD2 = 9
ledD3 = 10

a1 = 0
a2 = 0
b1 = 0
b2 = 0
c1 = 0
c2 = 0
d1 = 0
d2 = 0



def setup():
    global a1, a2, b1, b2, c1, c2, d1, d2
    a1 = 0
    a2 = 0
    b1 = 0
    b2 = 0
    c1 = 0
    c2 = 0
    d1 = 0
    d2 = 0

def loop():
    readSensor()
    if b1 == 1 and b2 == 1:
        roadBopen()
    elif d1 == 1 and d2 == 1 and (b1 == 0 or b2 == 0):
        roadDopen()
        if b1 == 1 and b2 == 1:
            roadBopen()
    elif a1 == 1 and a2 == 1 and ((d2 == 0 or b2 == 0) or (d1 == 0 or b1 == 0)):
        roadAopen()
        if b1 == 1 and b2 == 1:
            roadBopen()
        elif d1 == 1 and d2 == 1 and (b1 == 0 or b2 == 0):
            roadDopen()
    elif c1 == 1 and c2 == 1 and ((d2 == 0 or b2 == 0 or a2 == 0) or (d1 == 0 or b1 == 0 or a1
== 0)):
        roadCopen()
        if b1 == 1 and b2 == 1:
            roadBopen()
        elif d1 == 1 and d2 == 1 and (b1 == 0 or b2 == 0):
            roadDopen()

def readSensor():
    global a1, a2, b1, b2, c1, c2, d1, d2
    # Read sensor values and update variables a1, a2, b1, b2, c1, c2, d1, d2

def roadBopen():
    # Code for opening road B

def roadDopen():
    # Code for opening road D

def roadAopen():
    # Code for opening road A

def roadCopen():
    # Code for opening road C
if (b1 == 1 and b2 == 0) and (c1 == 1 or d1 == 1 or b1 == 1) and (c2 == 1 and d2 == 1 and b2
== 1):
  roadBopen()
elif (d1 == 1 and d2 == 0) and (c1 == 1 or a1 == 1) and (b1 == 0 and b2 == 0) and (c2 == 0 and
a2 == 0):
  roadDopen()
elif (a1 == 1 and a2 == 0) and (c1 == 1 and c2 == 0) and (d1 == 0 and d2 == 0) and (b1 == 0
and b2 == 0):
  roadAopen()
elif (c1 == 1 and c2 == 0) and (b1 == 0 or b2 == 0) and (d1 == 0 or d2 == 0) and (a1 == 0 or
a2 == 0):
  roadCopen()
elif (b1 == 1 and b2 == 0) and (c1 == 0 or c2 == 0) and (d1 == 0 or d2 == 0) and (a1 == 0 or
a2 == 0):
  roadBopen()
elif (d1 == 1 and d2 == 0) and (c1 == 0 or c2 == 0) and (b1 == 0 or b2 == 0) and (a1 == 0 or
a2 == 0):
  roadDopen()
elif (a1 == 1 and a2 == 0) and (c1 == 0 or c2 == 0) and (d1 == 0 or d2 == 0) and (b1 == 0 or
b2 == 0):
  roadAopen()
elif a1 == 0 and b1 == 0 and c1 == 0 and d1 == 0:
  roadBopen()
  if a1 == 0 and b1 == 0 and c1 == 0 and d1 == 0:
    roadDopen()
  if a1 == 0 and b1 == 0 and c1 == 0 and d1 == 0:
    roadAopen()
  if a1 == 0 and b1 == 0 and c1 == 0 and d1 == 0:
    roadCopen()
def readSensor():
    a1 = analogRead(A7)
    a2 = analogRead(A6)
    b1 = analogRead(A4)
    b2 = analogRead(A5)
    c1 = analogRead(A1)
    c2 = analogRead(A0)
    d1 = analogRead(A3)
    d2 = analogRead(A2)
    if a1 < 400:
        a1 = 1
    else:
        a1 = 0
    if a2 < 400:
        a2 = 1
    else:
        a2 = 0
    if b1 < 400:
        b1 = 1
    else:
        b1 = 0
    if b2 < 400:
        b2 = 1
    else:
        b2 = 0
    if c1 < 400:
        c1 = 1
    else:
        c1 = 0
    if c2 < 400:
        c2 = 1
    else:
        c2 = 0
    if d1 < 400:
        d1 = 1
    else:
        d1 = 0
    if d2 < 400:
        d2 = 1
    else:
        d2 = 0
    Serial.print(a1)
    Serial.print("\t")
    Serial.print(a2)
    Serial.print("\t")
    Serial.print(b1)
    Serial.print("\t")
    Serial.print(b2)
    Serial.print("\t")
    Serial.print(c1)
    Serial.print("\t")
    Serial.print(c2)
    Serial.print("\t")
    Serial.print(d1)
    Serial.print("\t")
    Serial.print(d2)
    Serial.println("\t")

def roadAopen():
    digitalWrite(ledA3, LOW)
    digitalWrite(ledA1, HIGH)
    digitalWrite(ledB3, HIGH)
    digitalWrite(ledC3, HIGH)
    digitalWrite(ledD3, HIGH)
    delay(10000)
    digitalWrite(ledA1, LOW)
    digitalWrite(ledA2, HIGH)
    delay(1000)
    digitalWrite(ledA2, LOW)
    readSensor()

def roadBopen():
    digitalWrite(ledB3, LOW)
    digitalWrite(ledA3, HIGH)
    digitalWrite(ledB1, HIGH)
    digitalWrite(ledC3, HIGH)
    digitalWrite(ledD3, HIGH)
    delay(10000)
    digitalWrite(ledB1, LOW)
    digitalWrite(ledB2, HIGH)
    delay(1000)
    digitalWrite(ledB2, LOW)
    readSensor()

def roadCopen():
    digitalWrite(ledC3, LOW)
    digitalWrite(ledA3, HIGH)
    digitalWrite(ledB3, HIGH)
    digitalWrite(ledC1, HIGH)
    digitalWrite(ledD3, HIGH)
    time.sleep(10)
    digitalWrite(ledC1, LOW)
    digitalWrite(ledC2, HIGH)
    time.sleep(1)
    digitalWrite(ledC2, LOW)
    readSensor()

def roadDopen():
    digitalWrite(ledD3, LOW)
    digitalWrite(ledA3, HIGH)
    digitalWrite(ledB3, HIGH)
    digitalWrite(ledC3, HIGH)
    digitalWrite(ledD1, HIGH)
    time.sleep(10)
    digitalWrite(ledD1, LOW)
    digitalWrite(ledD2, HIGH)
    time.sleep(1)
    digitalWrite(ledD2, LOW)
    readSensor()
```