TRAFFIC MANAGEMENT SYSTEM

PRESENTED BY

DESIYA E

ANITHA T

SANGETHA B

REVATHI S

SURYA A

VIJAYACHANDIRAN M

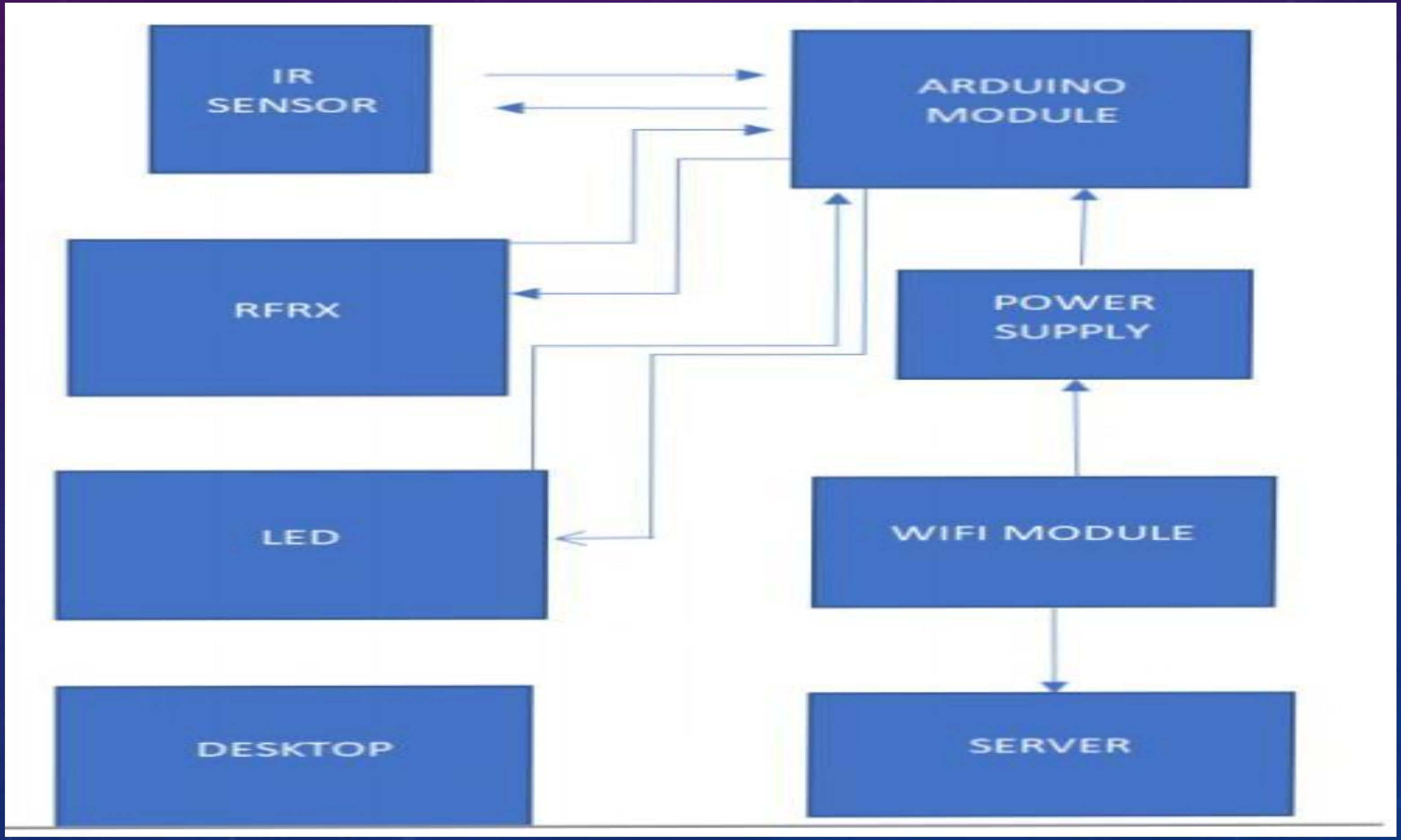
ABSTRACT

- Traffic management system is one of the major proportions of a smart city.
- With the rapid growth of proportions and rapid increase of vehicles across the whole country which further leads to the traffic congestion which is usually seen on roads.
- Nowadays traffic congestion is a difficult issue to deal with as number of vehicles is increasing day by day.
- A simple, effective and less costly method is used to optimize traffic flow on roads and an algorithm is devised to manage various traffic situations efficiently and automatically.

INTRODUCTION

- Traffic congestion on road networks creates many problems such as increased fuel consumption, increase air pollution, increased queuing of the vehicles and many more.
- In every cities of India traffic congestion is a major problem which we are facing nowadays.
- There are also even a severe many security problems in traffic system in our country due to many elements which also leads to the congestion of traffic at one place.
- India is the 2nd most populated country after China in the world, this with increase in population, the number of vehicles also increasing day by day.
- The economic growth has certainly had an impact on country traffic.

BLOCK DIAGRAM



DESCRIPTION

ARDUINO NANO

- An 8 bit microchip AVR which is small, complete and bread board friendly board based on the Atmega328.
- It is the main CPU of our Project, in which we all the program will run.

POWER SUPPLY MODULE

- A power supply is a hardware component that provides power to any electrical device.

IR SENIOR

- These sensors are used to detect the object through infrared rays.
- These sensors are put sideways for giving us the density of vehicles in the specific lane.

WIFI MODULE

- It is used to give microcontroller access to your wife network.

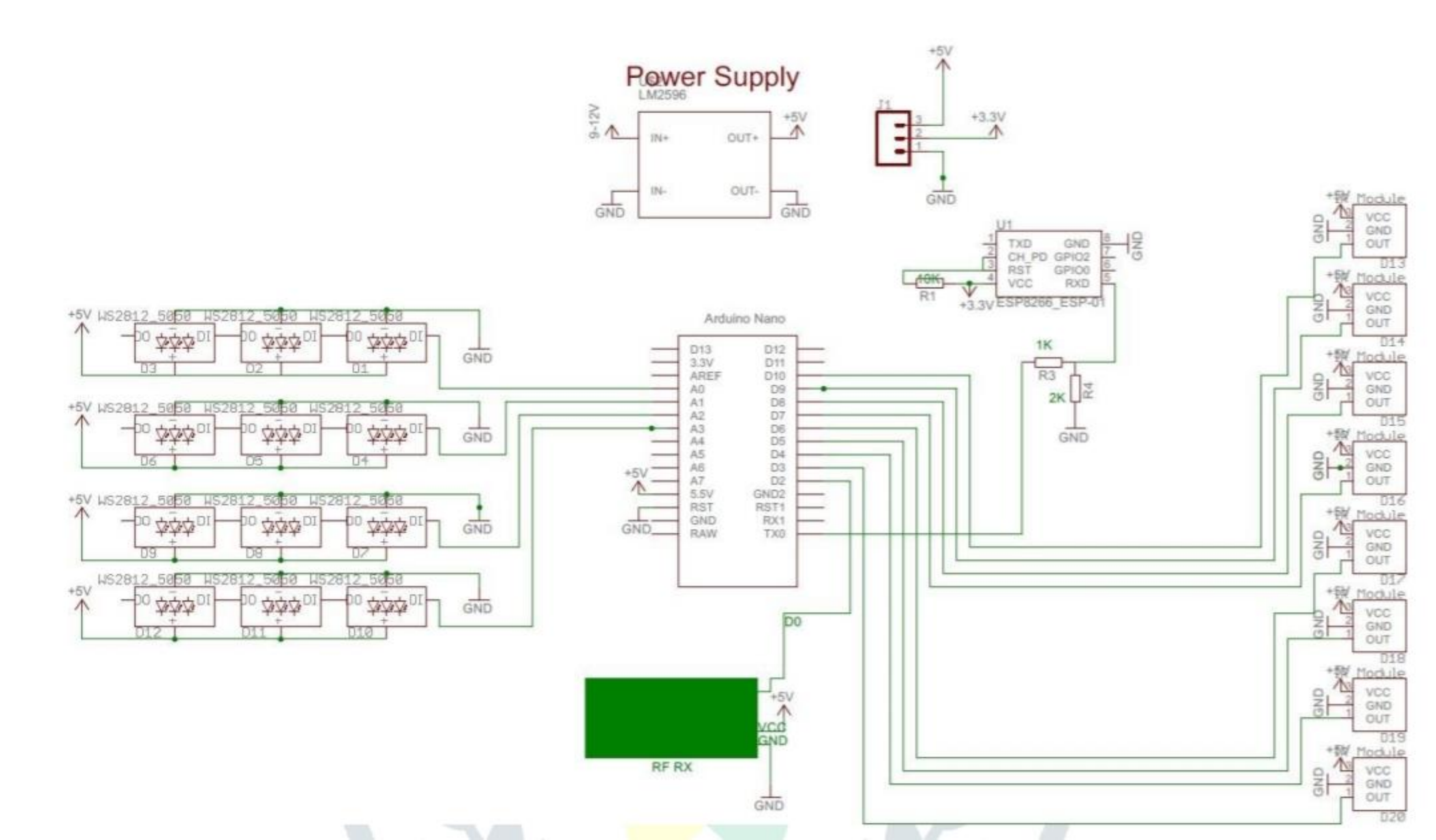
RFRX MODULE

- It consists of RF receiver; it is used for transmitting and receiving data.

BLYNK APP

- It is a mobile application for output and verification for real time data collected.

CIRCUIT DIAGRAM



WORKING PRINCIPLES

- 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, the C lane and at last D lane.
- But in case if any of the lane gets more cars or gets densor then the other 3 lanes then that specific lane will open then the other with the second highest densor, 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.

SOFTWARE

- We are using the python software for the traffic management system.
- It was developed by the Guido Van Rossum.
- Python is a interpreted, object oriented and high level language. It was developed in the year of 1991.
- The system we proposed uses a python algorithm that can easily work in real-time and helps in determining traffic light whilst scheduling the time phase that results in increasing traffic regularity and efficiency by predicting a perfect route, controlling traffic jams, reducing, average waiting time , improving traffic.

PROGRAM

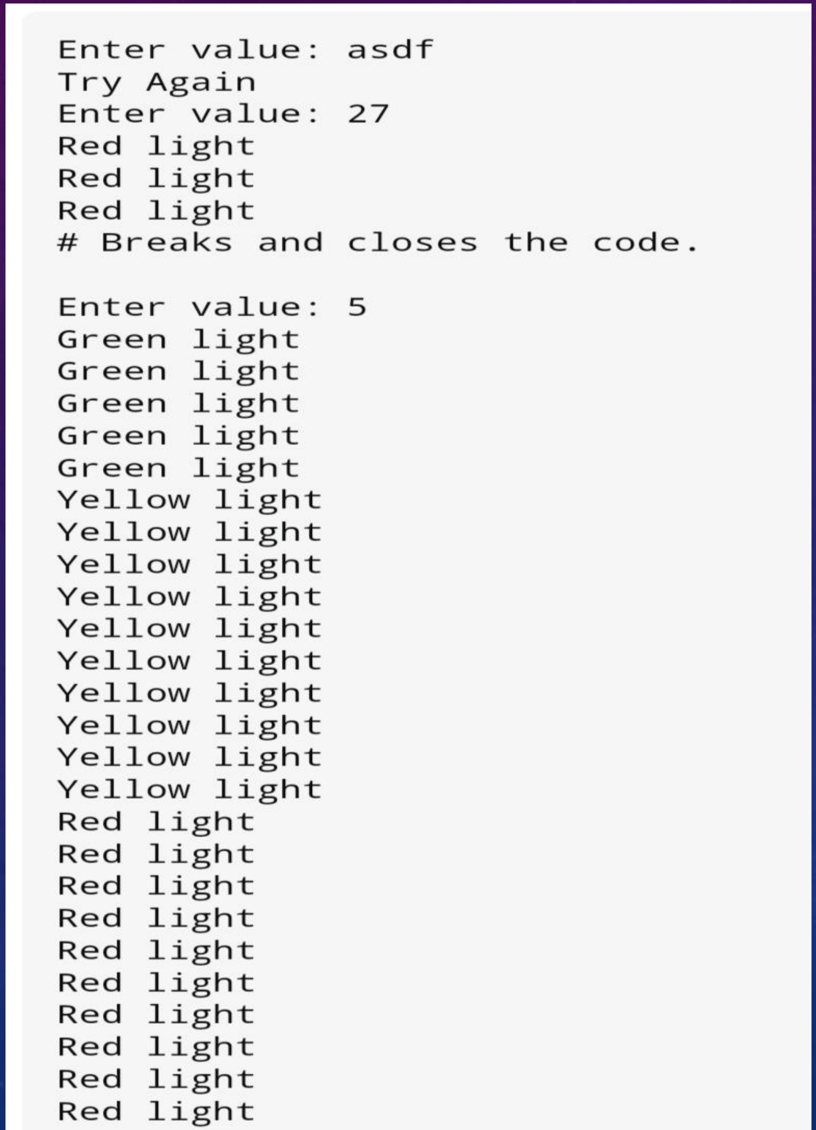
```
`x = input(“Enter value: “)
stop_light = int(x)
while True:
    if stop_light >= 1 and stop_light < 10:
        print(‘Green light’)
        stop_light += 1
    elif stop_light < 20:
        print(‘Yellow light’)
        stop_light += 1
    elif stop_light < 30:
        print(“Red light”)
        stop_light += 1
    else:
        stop_light = 0
    break`
```

```
While True:
    x = input(“Enter value: “)
    stop_light = int(x)
    if stop_light == 30:
        break
    elif stop_light >= 1 and stop_light < 10:
        print(‘Green light’)
        stop_light += 1
    elif stop_light < 20:
        print(‘Yellow light’)
        stop_light += 1
    elif stop_light < 30:
        print(“Red light”)
        stop_light += 1
    else:
        stop_light = 0
```

```
While True:
    try:
        x = input(“Enter value: “)
        stop_light = int(x)
    except ValueError:
        print(“Try Again”)
    else:
        break


while stop_light <= 30:
    if stop_light >= 1 and stop_light < 10:
        print(‘Green light’)
    elif stop_light < 20:
        print(‘Yellow light’)
    elif stop_light < 30:
        print(“Red light”)
    stop_light += 1
```

OUTPUT



```
Enter value: asdf
Try Again
Enter value: 27
Red light
Red light
Red light
# Breaks and closes the code.

Enter value: 5
Green light
Green light
Green light
Green light
Green light
Yellow light
Yellow light
Yellow light
Yellow light
Yellow light
Yellow light
Yellow light
Yellow light
Yellow light
Yellow light
Red light
Red light
Red light
Red light
Red light
Red light
Red light
Red light
Red light
Red light
```

THANKING YOU