**Project Design Phase-3**

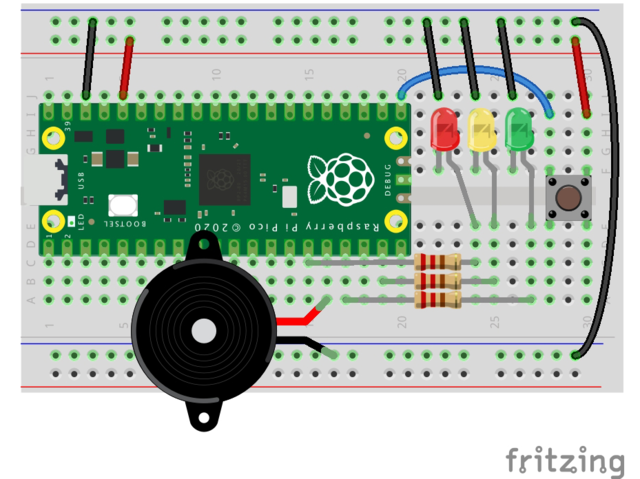
|  |  |
| --- | --- |
| Date | 18 October 2023 |
| Team ID | 448 |
| Project Name | 4123-Traffic Management System |
| Team Name | Proj\_227233\_Team\_1 |

**Develop a Python script on the IoT devices to send real-time traffic data to the traffic information platform.**

A traffic management plan involves planning and controlling the movement of people and goods within an area. This can include stationary and moving traffic, pedestrians, cyclists, and vehicles. The goal of traffic management is to keep this movement orderly and efficiently to minimise risk at the workplace.

These sensors use the ultrasonic Doppler effect. They detect vehicles travelling in a particular direction using a change in frequency (the Doppler effect) according to the speed of the vehicle.  
They are installed on side roads with low traffic volume, and are used for recall control to change the traffic light on the side road to green only when a vehicle is detected.

**Traffic Management Block Diagram:**



**Program code for Traffic Management:**

import machine

import time

# Define the LED pins

led\_red = machine.Pin(11, machine.Pin.OUT)

led\_red = machine.Pin(11, machine.Pin.OUT)  
led\_green = machine.Pin(5, machine.Pin.OUT)

def handle\_red\_state():

led\_red.value(1)

led\_yellow.value(0)

led\_green.value(0)

def handle\_yellow\_state():

led\_red.value(0)

led\_yellow.value(1)

led\_green.value(0)

def handle\_green\_state():

led\_red.value(0)

led\_yellow.value(0)

led\_green.value(1)

def handle\_yellow\_state\_short():

led\_red.value(0)

led\_yellow.value(1)

led\_green.value(0)

# State handlers list

state\_handlers = [

# (state function, time in milliseconds)

(handle\_red\_state, 5000), # Red LED, on for 5 seconds

(handle\_yellow\_state, 3000), # Yellow LED, on for 3 seconds

(handle\_green\_state, 5000), # Green LED, on for 5 seconds

(handle\_yellow\_state\_short, 2000) # Short Yellow LED, on for 2 seconds

]

def traffic\_light():

state = 0

while True:

# Get the current state tuple (handler function and sleep time)

current\_handler\_and\_time = state\_handlers[state]

handler\_func = current\_handler\_and\_time[0]

sleep\_duration\_ms = current\_handler\_and\_time[1]

# Execute the handler function and sleep for the specified time handler\_func()

time.sleep\_ms(sleep\_duration\_ms)

# Update the state index

state = (state + 1) % len(state\_handlers)

# Run the traffic light sequence

traffic\_light()

**OUTPUT:**

