INTERNET OF THINGS – GROUP2

PROJECT 10. TRAFFIC MANAGEMENT SYSTEM

OBJECTIVES:

Creating a traffic control management system using sensors, cameras, Arduino, Raspberry Pi, and a cloud simulator involves multiple components and steps. Below is a high-level overview along with sample code snippets for guidance. This is a simplified example and real-world systems require careful design, testing, and potentially certification depending on the application.

COMPONENTS USED ARE:

1. Raspberry Pi (with Raspbian OS installed)

2. Arduino Board

3. Sensors (e.g., ultrasonic sensors for vehicle detection)

4. Camera module

5. Cloud simulator (for ex: Azure IOT cloud simulator)

STEPS TO BE FOLLOWED:

STEP1.SETTING UP A HARDWARE

- Connect the sensors, camera, and Arduino to the Raspberry Pi as per their datasheets or manuals. Ensure they're powered properly.

STEP2.INSTALL REQUIRED LAIBAERIES AND SOFTWARE:

- If you're using additional sensors, you might need to install libraries for them. For example, for an ultrasonic sensor, you'd use something like the GPIOZERO library on the Raspberry Pi.

STEP3. WRITING AN ARDUINO CODE:

- The Arduino code will handle sensor data and potentially control hardware components like traffic lights.

STEP4. SETTING UP CLOUD SIMULATOR:

- Create an account on a cloud service and set up a project with IoT capabilities.

STEP5. SENDING DATA TO CLOUD FROM RASPBERRY PI:

- Use the cloud service's provided libraries or SDKs to send data from the Raspberry Pi to the cloud. This could be sensor data or images captured by the camera.

STEP6.IMPLEMENTING TRAFFIC CONTROL LOGIC IN CLOUD:

- In the cloud, process the received data (e.g., apply machine learning for object detection). Based on the processed data, decide when to change traffic signals.

STEP7.SENDING CONTROL SIGNALS FROM CLOUD TO ARDUINO:

- Send control signals from the cloud to the Arduino through the cloud service. This could be done using MQTT or other communication protocols.

STEP8.INTEGRATION AND TESTING:

- Integrate all components and thoroughly test the system.

WRITING PYTHON CODE FOR RASPBERRY PI:

#Sample Python Code for Vehicle Detection using Ultrasonic Sensors:

import Distance Sensor

from time import sleep

sensor = Distance Sensor (echo=17, trigger=4) # Example GPIO pins

while True:

distance = sensor. distance \* 100 # Convert to cm

print (f ’Distance: {distance:.2f} cm')

if distance < 30: # Adjust as needed

# Vehicle detected, implement traffic control logic here

pass

sleep (1)

# Sample Python Code for Image Processing using Camera:

python

from pi Camera import Pi Camera

from time import sleep

camera = Pi Camera ()

def capture\_ image (file\_ name):

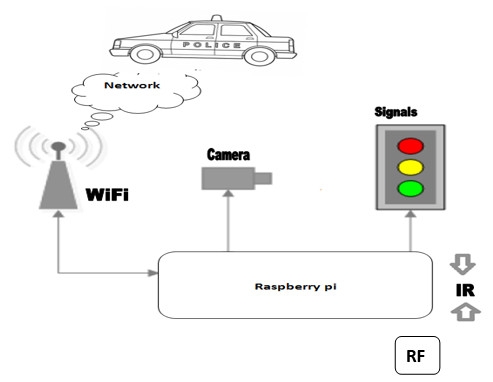
camera. Start \_ preview ()

sleep (2) # Allow time for the camera to adjust

camera capture(filename)

camera. Stop \_ preview ()

capture\_ image ('test\_image.jpg')



COMMON BENEFITS:

Using both Raspberry Pi and Arduino for traffic control management offers a combination of benefits due to their complementary strengths:

1. Scalability: By leveraging both Raspberry Pi and Arduino, you can create a scalable solution. Raspberry Pi can handle the high-level processing and coordination, while Arduinos manage specific hardware control tasks.

2. Redundancy and Reliability: Redundancy can be built into the system by using multiple Arduinos for critical tasks. This enhances system reliability as one board's failure won't affect the entire system.

3. Distributed Control: With Arduino handling hardware-specific tasks, the load on Raspberry Pi is reduced, allowing for a distributed and optimized control system.

4. Customization: The combination allows for highly customized solutions. Raspberry Pi can host a flexible software stack, while Arduino boards can be tailored to specific hardware requirements.

5. Low Latency: The real-time capabilities of Arduino can help reduce latency in critical tasks like traffic light control.

6.Low Power Consumption: Raspberry Pi devices are energy-efficient, making them suitable for long-term deployment

7.Cost-Effectiveness: Arduino boards are typically more affordable than Raspberry Pi, which can be advantageous for deploying multiple units across a traffic management system.

FEATURES:

Traffic control management systems typically include a range of features aimed at optimizing traffic flow, enhancing safety, and improving overall transportation efficiency. Here are some common features:

1. Traffic Signal Control: This includes the ability to adjust traffic signals based on real-time traffic conditions to reduce congestion and improve traffic flow.

2. Traffic Monitoring and Surveillance: Utilizes cameras, sensors, and other monitoring devices to collect data on traffic conditions, enabling real-time analysis and decision-making.

3. Incident Detection and Management: Automatically identifies incidents like accidents or road obstructions and provides notifications to appropriate authorities for quick response.

4. Dynamic Message Signs (DMS): Displays real-time information to drivers about traffic conditions, alternative routes, and important alerts.

6. Traffic Data Analytics: Gathers and analysis data to identify traffic patterns, congestion hotspots, and trends for better planning and decision-making.

7. Emergency Vehicle Priority: Allows emergency vehicles to be granted priority in traffic flow, enabling quicker response times.

BLOCK DIAGRAM:

