TRAFFIC MANAGEMENT SYSTEM

- 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.

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
```

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 ()
```