**SMART PARKING**

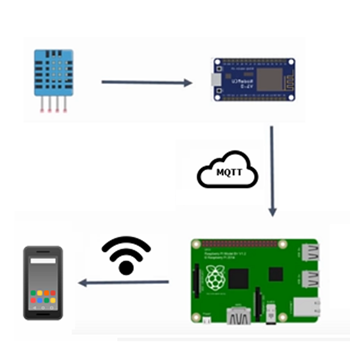
**Phase 3: DEVELOPMENT PART 1**

* **To Start building the IoT sensor system and Raspberry Pi integration.**
* **To Configure IoT sensors (e.g., ultrasonic sensors) to detect parking space occupancy.**
* **To Write Python scripts on Raspberry Pi to collect data from sensors and send it to the cloud or mobile app server.**

**Components Needed:**

* **Raspberry Pi 4**
* **MQTT Server**
* **Pi Camera Module**
* **Fritzing**
* **NEO-6M GPS Sensor**
* **HSC-SR04 Ultrasonic Sensors**

**Workflow:**

****

**Integrating IOT Sensor & Raspberry Pi:**

* **The Main IoT Sensor Used Here is Ultrasonic Sensor**
* **The Ultrasonic Sensor and Raspberry Pi is Integrated Using The Python Script**

**Python Script:**

**pip install RPi.GPIO**

**Configuring Ultrasonic Sensor:**

* **Configuring the Raspberry Pi Pins for The UltraSonic Sensor**
* **Trigger Input Will be given to Specified Pins only**

**Python Script:**

**import RPi.GPIO as GPIO**

**import time**

**# Set the GPIO mode to BCM**

**GPIO.setmode(GPIO.BCM)**

**# Define the GPIO pins for the ultrasonic sensor**

**TRIG = 23 # GPIO pin for the trigger (TX)**

**ECHO = 24 # GPIO pin for the echo (RX)**

**# Set up the GPIO pins**

**GPIO.setup(TRIG, GPIO.OUT)**

**GPIO.setup(ECHO, GPIO.IN)**

**Collecting Sensor Data:**

* **The Ultrasonic Sensor Data is Collected through the Raspberry Pi**
* **The Collected Data will be processed for various applications.**

**Python Script:**

**try:**

**while True:**

**# Send a short pulse to trigger the ultrasonic sensor**

**GPIO.output(TRIG, True)**

**time.sleep(0.00001)**

**GPIO.output(TRIG, False)**

**# Record the start time of the pulse**

**while GPIO.input(ECHO) == 0:**

**pulse\_start = time.time()**

**# Record the end time of the pulse**

**while GPIO.input(ECHO) == 1:**

**pulse\_end = time.time()**

**# Calculate the duration of the pulse**

**pulse\_duration = pulse\_end - pulse\_start**

**# Calculate the distance using the speed of sound (34300 cm/s)**

**distance = (pulse\_duration \* 34300) / 2**

**# Print the distance**

**print("Distance: {:.2f} cm".format(distance))**

**# Sleep for a while to avoid continuous measurements**

**time.sleep(1)**

**except KeyboardInterrupt:**

**pass**

**finally:**

**# Clean up the GPIO pins**

**GPIO.cleanup()**

**Sending to MQTT Cloud Server:**

* **Sending The Collected data from the Ultrasonic Sensor to The MQTT server.**
* **The Received Data Can be Stored in the MQTT Cloud Server.**

**Python Script:**

**import paho.mqtt.client as mqtt**

**# MQTT settings**

**mqtt\_broker = "My\_mqtt\_broker\_address"**

**mqtt\_port = 1883**

**mqtt\_topic = "Smart\_Parking"**

**# Create an MQTT client**

**client = mqtt.Client("UltrasonicSensor")**

**# Connect to the MQTT broker**

**client.connect(mqtt\_broker, mqtt\_port, keepalive=60)**

**# Send the data to the MQTT server**

**client.publish(mqtt\_topic, payload="{:.2f}".format(distance))**

**Prepared By,**

**VISHNUVARTHAN S (Team Leader)**