Smart Parking System – Phase 3

Welcome to Phase 3 of the Smart Parking System project. In this phase, we will guide you through the setup and code implementation for integrating an HC-SR04 ultrasonic sensor with an Arduino Uno and transmitting the collected data to a Raspberry Pi for further processing and integration with cloud and mobile applications.

Arduino Setup

1. Connect HC-SR04 to Arduino Uno:

To set up the HC-SR04 sensor, follow these connections:

Connect the VCC pin of the HC-SR04 sensor to the 5V pin on the Arduino Uno.

Connect the GND pin of the HC-SR04 sensor to the GND pin on the Arduino Uno.

Connect the Trig pin of the HC-SR04 sensor to a digital output pin (e.g., D2 on Arduino).

Connect the Echo pin of the HC-SR04 sensor to another digital input pin (e.g., D3 on Arduino).

Refer to the following image for a visual representation of the Arduino setup:

1. Arduino Code:

Before you proceed, ensure you have the “NewPing” library installed in your Arduino IDE. Then, upload the following Arduino code to read data from the HC-SR04 sensor and send it to the Raspberry Pi via Serial communication:

#include <NewPing.h>

#define TRIG\_PIN 2

#define ECHO\_PIN 3

#define MAX\_DISTANCE 200

NewPing sonar(TRIG\_PIN, ECHO\_PIN, MAX\_DISTANCE);

void setup() {

Serial.begin(9600);

}

void loop() {

delay(1000); // Adjust the delay as needed

unsigned int distance = sonar.ping\_cm();

Serial.println(distance);

}

This Arduino code initializes the HC-SR04 sensor and sends the distance measurements to the Raspberry Pi through a serial connection.

Raspberry Pi Setup

1. Connect Raspberry Pi to Arduino Uno:

Establish a connection between the Raspberry Pi and Arduino Uno using a USB cable for serial communication.

1. Install Serial Communication Libraries:

On your Raspberry Pi, install the necessary libraries for serial communication using the following command:

sudo apt-get install python-serial

1. Python Code for Data Handling:

Create a Python script to read data from the Arduino over the USB serial port and forward it to the cloud or a mobile app server. Below is a basic Python script to get you started:

import serial

ser = serial.Serial(‘/dev/ttyACM0’, 9600)

# Optionally, open a file to log the data

log\_file = open(‘sensor\_data.log’, ‘a’)

while True:

try:

data = ser.readline().decode().strip() # Read data from the Arduino

# Optionally, log the data to a file

log\_file.write(data + ‘\n’)

log\_file.flush()

# Send data to a cloud or mobile app server

# Implement the server communication according to your specific requirements.

print(f”Received data from Arduino: {data}”)

except KeyboardInterrupt:

ser.close()

log\_file.close()

break

Conclusion

This README serves as a solid starting point for implementing Phase 3 of your Smart Parking System. Remember to customize the code and integration process according to your unique project needs and the specifications of your chosen cloud platform or app server.