PROJECT-PHASE 1

SMART PARKING

Creating a smart parking project for ESP32 on the Wokwi platform involves using the ESP32 microcontroller to detect and manage parking spaces, and then visualizing the data on a virtual interface provided by wokwi.here's a step-by-step guide how to create such a project:

Components needed:

1.ESP32 development board

2.Ultrasonic distance sensors(HC-SR04) for each parking space

3.breadboard and jumper wires

4.Wokwi virtual simulator(http://wokwi.com/)

PROJECTS STEPS:

1.Hardware setup

a. connect the HC-SR04 ultrasonic sensors to your ESP32 board. You will need one sensor per parking space.

b.wire the HC-SR04 sensors as follows

1.VCC to 5V on ESP32

2.GND to GND on ESP32

3.Trig to a digital G PIO pin on ESP32

4.Echo to another digital G PIO pin on ESP32

c. Connect all the sensors in same way,one for each parking space you want to monitor

2.Programming

a. Write an Arduino sketch for the ESP32 that reads the distance data from ultrasonic sensors

```
‘’’cpp
#include<ultrasonic.h>
Ultrasonic sensor1(GPIO_TRIGGER1,GPIO_ECHO1);
Ultrasonic sensor2(GPIO_TRIGGER2,GPIO_ECHO2);
//add more sensors if needed

Void setup(){
   Serial.begin(115200);
}
Void loop(){
Long distance1=sensor1.read();
Long distance2=sensor2.read();
//read distances from more sensors if needed
//process distance data manage parking spaces here
Delay(1000);
//delay for better readability
}
…
```

b. In the loop function,process the distance data from each sensors to determine whether a parking space is occupied or vacant.you can set a threshold distance to decide when a space is occupied

c. You may want to use data structure to keep track of parking space status

3.Visualization

a. Go to the wokwi platform (http://wowki.com/) and create an account if you haven’t already.
b. Create a new project and select the ESP32 as your target board.
c. Import the Arduino sketch you created earlier into wowki editor.
d. Use the virtual interface provided by wowki to display the parking space status.you can use LEDs or any other graphical elements to represents the parking space.

4. Testing

a. Simulate the project on wowki and observe how the parking space status changes based on simulated distance measurements

b. Fine-tune your code and interface as needed to ensure it works correctly

5. Deployment

Once your smart parking projects works as expected in virtual machine simulator, you can deploy it to a physical ESP32 board

6.Enhancements

Depending on your project requirements, you can add extra features such as mobile app integration for real-time parking updates,data logging and alerts when parking spaces are full or vacant