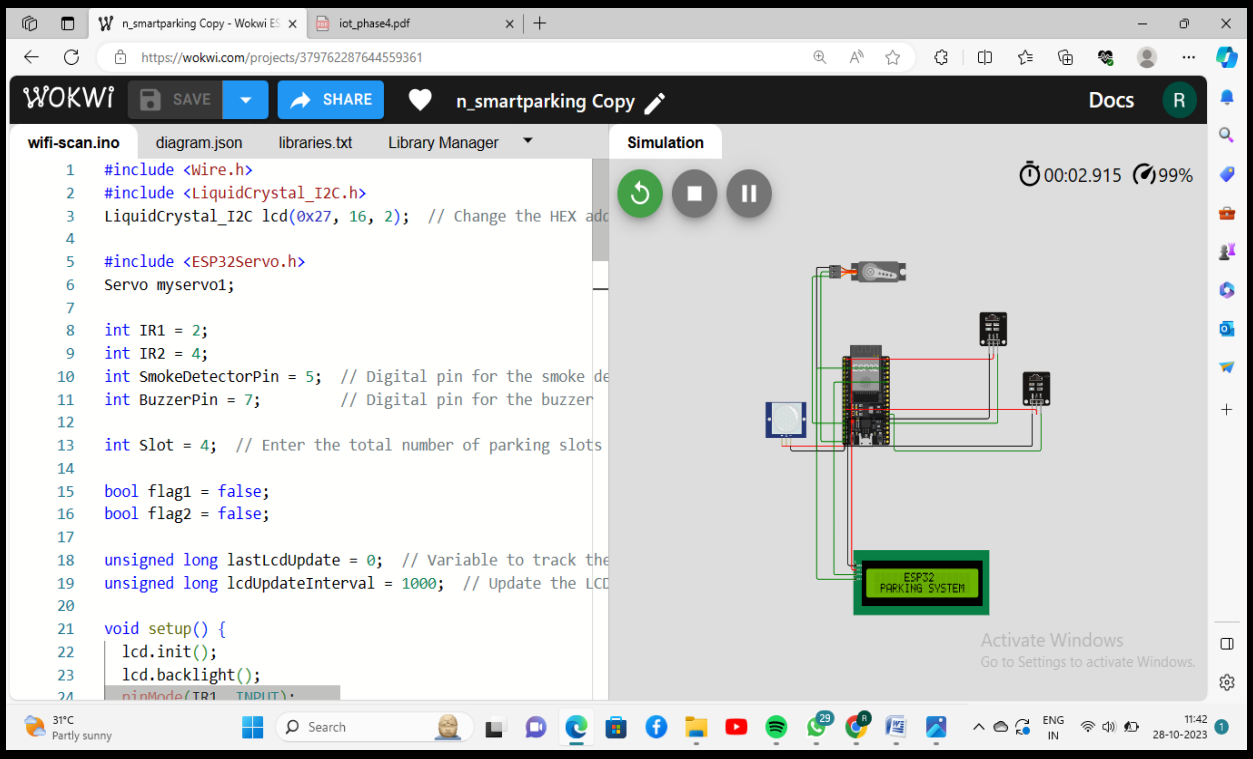
SMART PARKING SYSTEM

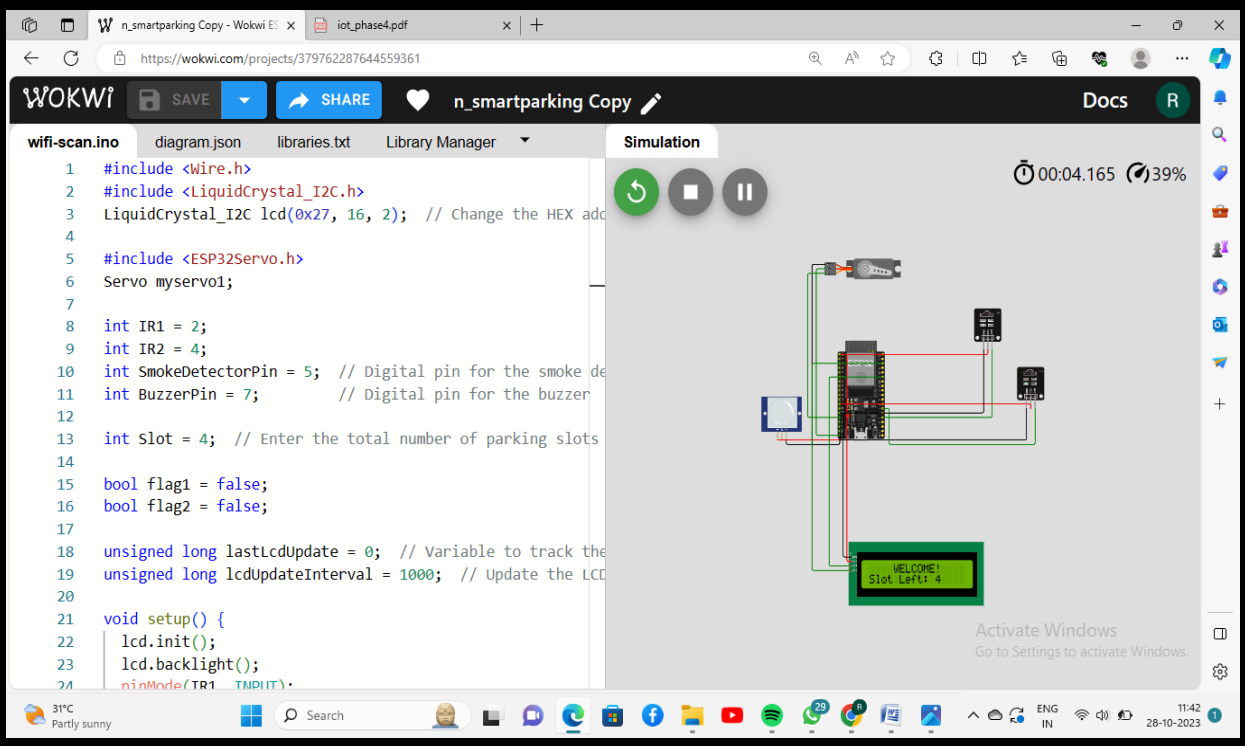
CODE FOR CONNECTING WOKWI WITH BLYNK:

```
#define BLYNK_TEMPLATE_ID "TMPL30KUsGLrZ"
#define BLYNK_TEMPLATE_NAME "SmartParking"
#define BLYNK_AUTH_TOKEN "K4-5fhkkbqgqELD0iTN3VVsBCVAeT6E0"
#include <Wire.h>
#include <BlynkSimpleEsp32.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <ESP32Servo.h>
 // Use Servo library for ESP32
Servo myservo1;

int IR1 = 2;
int IR2 = 4;
int SmokeDetectorPin = 5;  // Digital pin for the smoke detector
int BuzzerPin = 7;        // Digital pin for the buzzer

int Slot = 4;  // Enter the total number of parking slots

bool flag1 = false;
bool flag2 = false;

unsigned long lastLcdUpdate = 0;  // Variable to track the time of the last LCD
update
unsigned long lcdUpdateInterval = 1000;  // Update the LCD every 1000
milliseconds (1 second)
char auth[] = "K4-5fhkkbqgqELD0iTN3VVsBCVAeT6E0";
char ssid[] = "";
char pass[] = "";

void setup() {
  lcd.init();
  lcd.begin(16, 2);  // Initialize LCD with 16 columns and 2 rows
```

```
  lcd.backlight();
  pinMode(IR1, INPUT);
  pinMode(IR2, INPUT);
  pinMode(SmokeDetectorPin, INPUT);
  pinMode(BuzzerPin, OUTPUT);

  myservo1.attach(13);
  myservo1.write(100);

  lcd.setCursor(0, 0);
  lcd.print("     ARDUINO    ");
  lcd.setCursor(0, 1);
  lcd.print(" PARKING SYSTEM ");
  delay(2000);
  lcd.clear();

  Serial.begin(9600);  // Start serial communication for debugging
  Blynk.begin(auth, ssid, pass);  // Initialize Blynk
}

void loop() {
  Blynk.run();  // Allow Blynk to run

  if (digitalRead(IR1) == LOW && !flag1) {
    if (Slot > 0) {
      flag1 = true;
      if (!flag2) {
        myservo1.write(0);
        Slot--;
      }
    } else {
      displayMessage("    SORRY :(    ", "  Parking Full  ");
    }
  }

  if (digitalRead(IR2) == LOW && !flag2) {
    flag2 = true;
    if (!flag1) {
      myservo1.write(0);
      Slot++;
    }
```

```
  }

  if (flag1 && flag2) {
    delay(1000);
    myservo1.write(100);
    Serial.println("Servo returned to the initial position.");
    flag1 = false;
    flag2 = false;
  }

  // Update the LCD display with a delay
  if (millis() - lastLcdUpdate >= lcdUpdateInterval) {
    updateLcdDisplay();
    lastLcdUpdate = millis();
  }

}

void updateLcdDisplay() {
  if (digitalRead(SmokeDetectorPin) == HIGH) {
    displayMessage("   WARNING!   ", " Smoke Detected ");
    digitalWrite(BuzzerPin, HIGH);  // Turn on the buzzer
  } else {
    displayMessage("    WELCOME!    ", "Slot Left: " + String(Slot));
    digitalWrite(BuzzerPin, LOW);   // Turn off the buzzer
  }
}

void displayMessage(const char *line1, const String &line2) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(line1);
  lcd.setCursor(0, 1);
  lcd.print(line2);
}
```


n_smartparking Copy - Wokwi ES
iot_phase4.pdf
https://wokwi.com/projects/379762287644559361
WOKWI
SAVE
SHARE
n_smartparking Copy
Docs
wifi-scan.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:02.915
99%
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); // Change the HEX add
#include <ESP32Servo.h>
Servo myservo1;
int IR1 = 2;
int IR2 = 4;
int SmokeDetectorPin = 5; // Digital pin for the smoke de
int BuzzerPin = 7; // Digital pin for the buzzer
int Slot = 4; // Enter the total number of parking slots
bool flag1 = false;
bool flag2 = false;
unsigned long lastLcdUpdate = 0; // Variable to track the
unsigned long lcdUpdateInterval = 1000; // Update the LCD
void setup() {
lcd.init();
lcd.backlight();
ESP32
PARKING SYSTEM
Activate Windows
Go to Settings to activate Windows.
31°C
Partly sunny
Search
ENG
IN
11:42
28-10-2023


n_smartparking Copy - Wokwi ES
iot_phase4.pdf
https://wokwi.com/projects/379762287644559361
WOKWI
SAVE
SHARE
n_smartparking Copy
Docs
wifi-scan.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:04.165
39%
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); // Change the HEX add
#include <ESP32Servo.h>
Servo myservo1;
int IR1 = 2;
int IR2 = 4;
int SmokeDetectorPin = 5; // Digital pin for the smoke de
int BuzzerPin = 7; // Digital pin for the buzzer
int Slot = 4; // Enter the total number of parking slots
bool flag1 = false;
bool flag2 = false;
unsigned long lastLcdUpdate = 0; // Variable to track the
unsigned long lcdUpdateInterval = 1000; // Update the LCD
void setup() {
lcd.init();
lcd.backlight();
WELCOME!
Slot Left: 4
Activate Windows
Go to Settings to activate Windows.
31°C
Partly sunny
Search
ENG
IN
11:42
28-10-2023

n_smartparking Copy - Wokwi ES | iot_phase4.pdf

https://wokwi.com/projects/379762287644559361

WOKWI | SAVE | SHARE | n_smartparking Copy | Docs

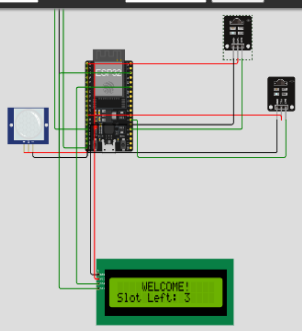
wifi-scan.ino | diagram.json | libraries.txt | Library Manager | Simulation

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Change the HEX adc

#include <ESP32Servo.h>
Servo myservo1;

int IR1 = 2;
int IR2 = 4;
int SmokeDetectorPin = 5;  // Digital pin for the smoke de
int BuzzerPin = 7;         // Digital pin for the buzzer

int Slot = 4;  // Enter the total number of parking slots

bool flag1 = false;
bool flag2 = false;

unsigned long lastLcdUpdate = 0;  // Variable to track the
unsigned long lcdUpdateInterval = 1000;  // Update the LCD

void setup() {
  lcd.init();
  lcd.backlight();
  pinMode(IR1, INPUT);
```

00:08.931 45%

IR Receiver

Command: 1 Address: 0 Send



Activate Windows
Go to Settings to activate Windows.

31°C Partly sunny | Search | ENG IN | 11:42 28-10-2023

n_smartparking Copy - Wokwi ES | iot_phase4.pdf

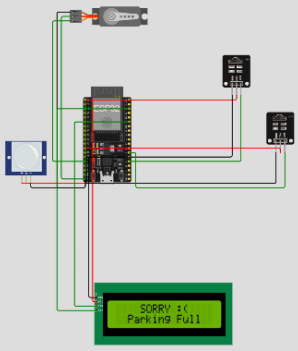
https://wokwi.com/projects/379762287644559361

WOKWI | SAVE | SHARE | n_smartparking Copy | Docs

wifi-scan.ino | diagram.json | libraries.txt | Library Manager | Simulation

```
  lcd.init();
  lcd.backlight();

  lcd.setCursor(0, 0);
  lcd.print("     ESP32     ");
  lcd.setCursor(0, 1);
  lcd.print(" PARKING SYSTEM ");
  delay(2000);
  lcd.clear();

  Serial.begin(9600);
}

void loop() {
  displayMessage("    SORRY :(    ", "  Parking Full  ");
  digitalWrite(BuzzerPin, LOW);
  delay(5000);
}

void displayMessage(const char *line1, const char *line2) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(line1);
```

00:04.645 101%



Activate Windows
Go to Settings to activate Windows.

31°C Partly sunny | Search | ENG IN | 11:53 28-10-2023

MOBILE APPLICATION:

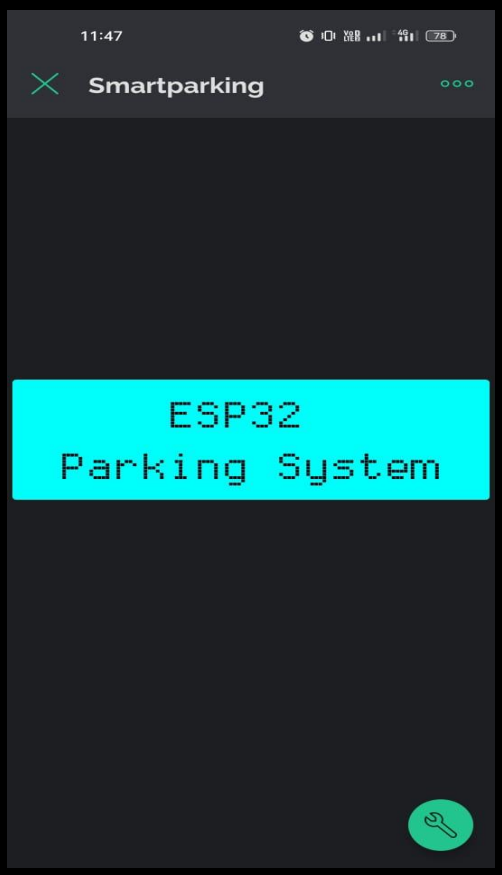
We've developed a mobile application that is designed to be user-friendly and accessible via smartphones, providing a simple interface for users.

The ESP32 microcontroller is equipped with sensors to collect real-time data about parking space occupancy.

It processes the sensor data to determine whether parking spaces are available or occupied.

Then connects to the mobile app through the Blynk platform, enabling real-time data communication.

Users can request and view up-to-the-minute parking availability data in the mobile app ,enhancing the overall parking experience.




22:12
Smartparking
WELCOME!
Slot Left: 3


22:13
Smartparking
SORRY :(
Parking Full!!