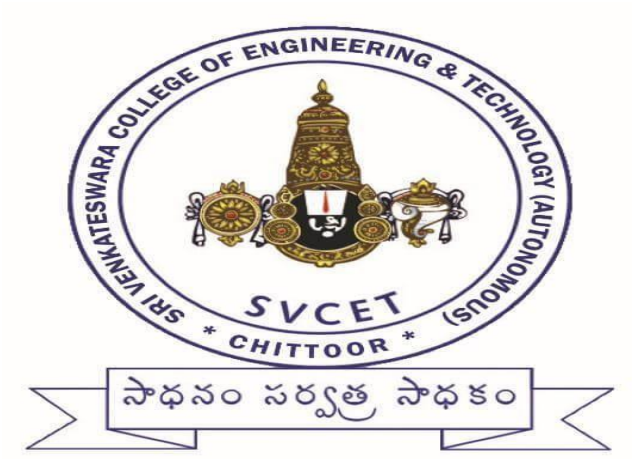


SRI VENKATESWARA COLLEGE OF ENGINEERING& TECHNOLOGY

(AUTONOMOUS)

R.V.S NAGAR, CHITTOOR – 517 127. (A.P)

(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapuramu) (Accredited

By NBA, New Delhi & NAAC, Bengaluru)

(An ISO 9001:2000 Certified Institution)

2023-2024

PROJECT DETAILS: (Mid-Course Project)

CLASS : Third year ECE(second semester)

COURSE NAME: Applied industrial IOT

PROJECT NAME : GARAGE DOOR CONTROOL SYSYEM

BATCH: Batch 54(4)

1.G.V. PAVAN KUMAR (22785A0411)

2.GURUGINZAKUNTA GANGADHAR (22785A0412)

3.M.VENKATA GANESH (22785A0417)

4.BODDUPOINA SOWMYA (22785A0404)

PROJECT

CONFIGURATION OF ADDRESS RESOLUTION PROTOCOL(ARP)

AIM :To design and develop a garage door control system using ESP8266 with its wifi capability to provides an opportunity for creating a smart garage door opener that enhances security and offers remote control futures.

PROBLEM STATEMENT: Traditional garage door openers lack smart features and remote control capabilities, which can compromise security and convenience for homeowners. Additionally, existing smart garage door openers in the market may be expensive or lack customizable features. Therefore, there is a need for a cost-effective and customizable smart garage door opener solution that enhances security and offers remote control features.

SCOPE OF THE SOLUTION:

Software Development: Develop firmware for the ESP8266 microcontroller to handle Wi-Fi connectivity, sensor data processing, and actuation of the garage door mechanism.

Implement secure authentication mechanisms to prevent unauthorized access to the garage door control system.

Create a user-friendly web interface or mobile application for remote control of the garage door, allowing users to open, close, and monitor the status of the garage door from anywhere with an internet connection.

Incorporate features such as scheduling for automatic opening/closing of the garage door at specific times or geofencing for automatic actions based on the user's location.

REQUIRED COMPONENTS TO DEVELOP SOLUTIONS:

1. Tinkercad(software)
2. PCs
3. ESP8266 microcontroller

Concept:

An automatic garage door opener involves integrating various components like a motor for opening/closing the door, sensors for detecting obstacles or the door's position, and a controller to manage the operation.

STEPS INVOLVED IN SMART GARAGE DOOR OPENER:

Designing a garage door opener using ESP8266 WiFi on Tinkercad can be a fun project. Below is a step-by-step guide to create a simple simulation of such a system:

Step 1: Set Up Your Tinkercad Account

If you haven't already, sign up for an account on Tinkercad.

Step 2: Create a New Circuit

Log in to Tinkercad and navigate to the Circuits section.

Click on "Create New Circuit" to start a new project.

Step 3: Add Components

Drag and drop an ESP8266 module from the Components panel onto the workspace. This will be the brain of your project.

Add a servo motor. This will simulate the garage door mechanism.

Include a push-button switch. This will act as your virtual garage door opener button.

You might also want to add some LEDs for status indication, although this step is optional.

Step 4: Wire Components

Connect the VCC (power) and GND (ground) pins of the ESP8266 and servo motor to the 3.3V and GND rails respectively.

Connect the signal pin of the servo motor to one of the digital pins on the ESP8266, for example, GPIO2.

Connect one terminal of the push-button switch to a digital pin on the ESP8266, for example, GPIO0.

Connect the other terminal of the push-button switch to GND.

Step 5: Write Code

You can write code for the ESP8266 using the Arduino IDE.

Step 6: Upload Code

Copy the code into the Arduino IDE.

Select the appropriate board (e.g., NodeMCU 1.0) and COM port.

Upload the code to your ESP8266.

Step 7: Test

Once the code is uploaded, go back to Tinkercad and press the Start Simulation button.

You can now test your garage door opener by clicking on the virtual push-button switch.

That's it! You've created a simulated garage door opener using ESP8266 WiFi on Tinkercad. You can further expand this project by adding features like WiFi connectivity for remote operation or integrating sensors for automation.

REMOTE CONTROL FUTURES:

Remote Access:

Allow users to remotely control the garage door opener from anywhere with an internet connection, using a smartphone app or web interface.

Security:

Implement secure authentication mechanisms such as username/password, biometric authentication, or two-factor authentication to prevent unauthorized access.

Utilize encryption protocols (e.g., SSL/TLS) to ensure secure communication between the user's device and the garage door opener system.

Enable activity logging to track who accessed the garage door and when, providing an audit trail for security purposes.

Real-time Status Monitoring:

Provide real-time updates on the status of the garage door (open/closed) to the user's smartphone app or web interface.

Send push notifications or alerts to the user's device in case of unauthorized access attempts, door left open for an extended period, or sensor malfunctions.

Customizable Access Permissions:

Allow users to set access permissions for different individuals (e.g., family members, friends, delivery personnel) with varying levels of access control.

Implement temporary access codes or time-limited access tokens for guests or service providers, which expire after a specified period.

Automation and Scheduling:

Enable users to schedule automatic opening/closing of the garage door at specific times or intervals.

Implement geofencing capabilities to automatically open/close the garage door when the user's smartphone enters or leaves a predefined geographic area.

Integration with Smart Home Ecosystems:

Ensure compatibility with popular smart home platforms and protocols (e.g., Amazon Alexa, Google Assistant, Apple HomeKit) for seamless integration into existing smart home ecosystems.

Allow users to control the garage door opener using voice commands through smart speakers or virtual assistants.

Obstacle Detection and Safety Features:

Incorporate sensors such as ultrasonic sensors or infrared sensors to detect obstacles in the path of the garage door and prevent accidents.

Implement safety features such as automatic reversal of the door's direction if an obstruction is detected during closing.

Battery Backup and Power Management:

Include a battery backup system to ensure the garage door opener remains functional during power outages.

Implement power-saving features to optimize energy consumption and extend the battery life of the system.

Remote Diagnostics and Troubleshooting:

Provide remote diagnostic capabilities to allow users to troubleshoot common issues and perform maintenance tasks remotely.

Offer access to online support resources, FAQs, and troubleshooting guides through the smartphone app or web interface.

User-friendly Interface and Accessibility:

Design an intuitive and user-friendly interface for the smartphone app or web interface, with clear navigation and instructions.

Ensure accessibility features such as voice commands, screen reader support, and adjustable font sizes for users with disabilities.

Battery Backup and Power Management:

Include a battery backup system to ensure the garage door opener remains functional during power outages.

Implement power-saving features to optimize energy consumption and extend the battery life of the system.

Remote Diagnostics and Troubleshooting:

Provide remote diagnostic capabilities to allow users to troubleshoot common issues and perform maintenance tasks remotely.

Offer access to online support resources, FAQs, and troubleshooting guides through the smartphone app or web interface.

User-friendly Interface and Accessibility:

Design an intuitive and user-friendly interface for the smartphone app or web interface, with clear navigation and instructions.

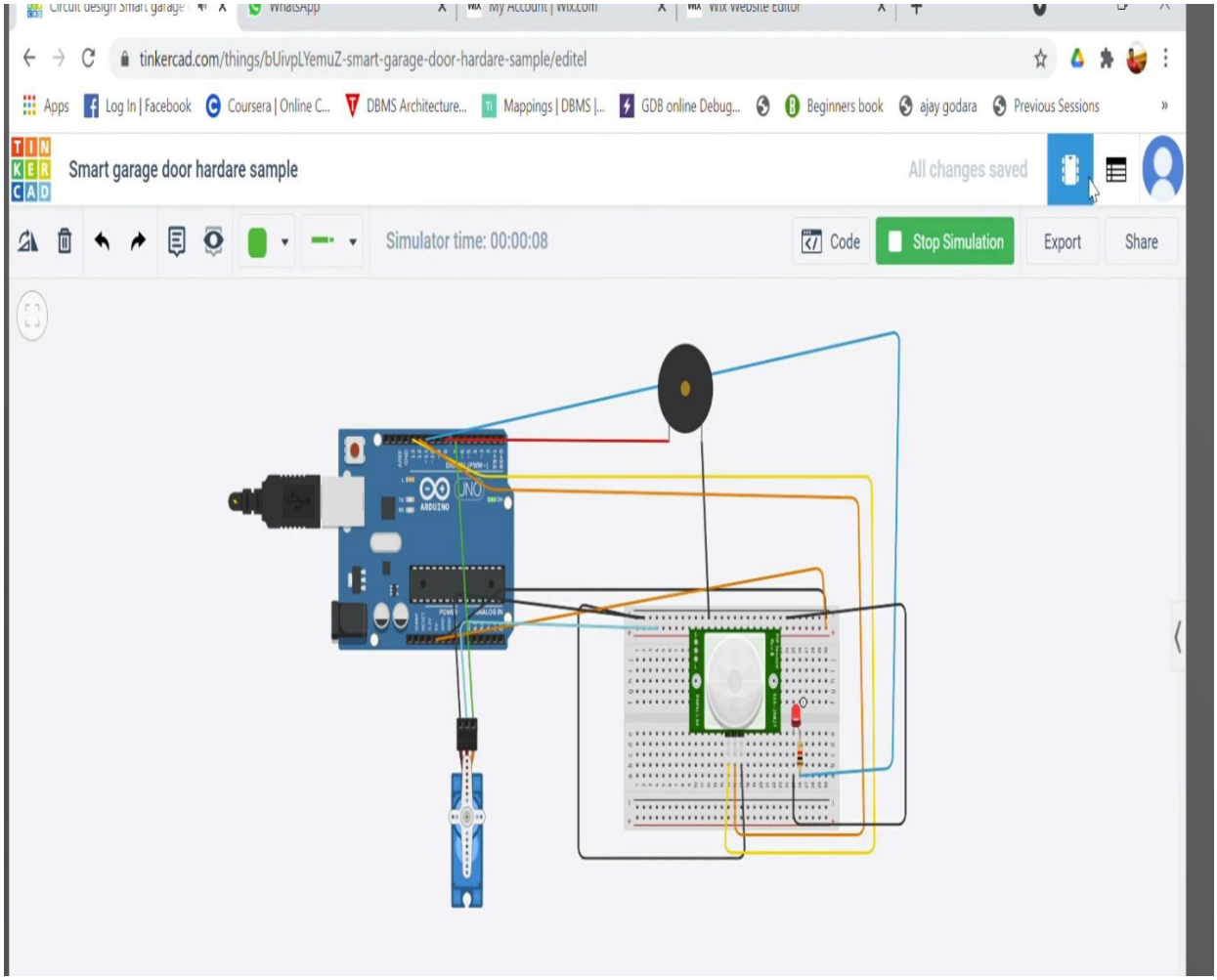
Ensure accessibility features such as voice commands, screen reader support, and adjustable font sizes for users with disabilities.

CODE FOR THE SOLUTION:

```
include <Servo.h>

#define SERVO_PIN 9  // Pin connected to the servo motor

#define BUZZER_PIN 8 // Pin connected to the buzzer

#define PIR_PIN 7    // Pin connected to the PIR sensor

#define LED_PIN 6    // Pin connected to the LED

Servo servo;

int pirState = LOW;

int lastPirState = LOW;

 void setup() {

 servo.attach(SERVO_PIN);

  pinMode(BUZZER_PIN, OUTPUT);

  pinMode(PIR_PIN, INPUT);

  pinMode(LED_PIN, OUTPUT);

}

void loop() {

  // Move the servo motor back and forth

  for (int angle = 0; angle <= 180; angle++) {

    servo.write(angle);

    delay(15);
```

```
  }
  for (int angle = 180; angle >= 0; angle--) {
    servo.write(angle);
    delay(15);
  }
digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(LED_PIN, HIGH);
    delay(1000); // Buzzer and LED on for 1 second
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(LED_PIN, LOW);
  }
  lastPirState = pirState;
}
```

SIMULATED CIRCUIT:



CONCLUTION:

In conclusion, a smart garage door opener offers enhanced security, convenience, and control. Through remote access, scheduling, and integration with smart home systems, users can ensure safety, automate operations, and manage access efficiently, making it a valuable addition to any modern home.