Hacettepe University

BBM432 Course Project

19 January

Name and Surname: ZÜBEYDE CİVELEK

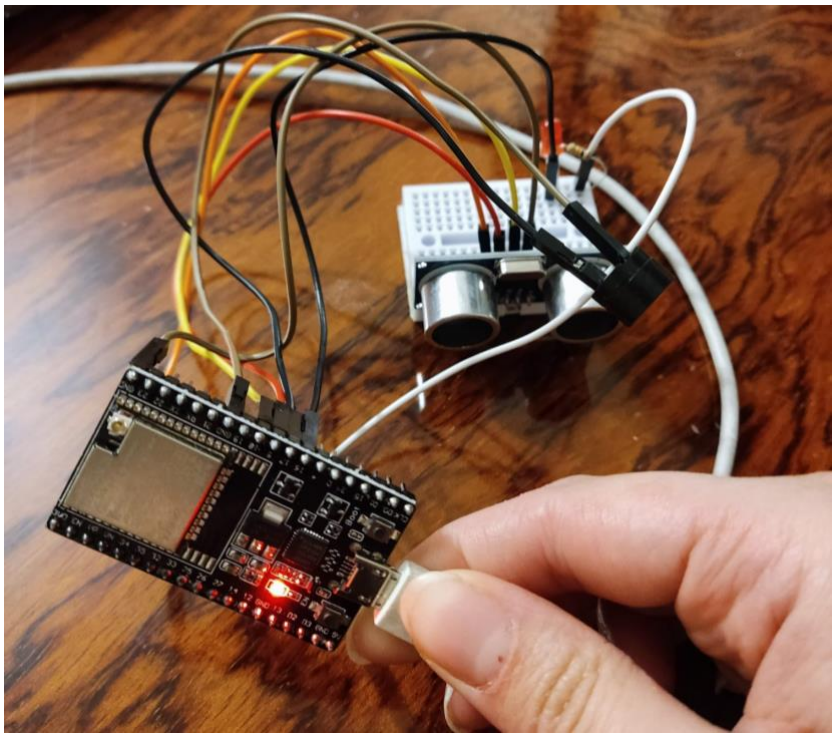
Identity Number: 2200356814

E-Mail: b2200356814@cs.hacettepe.edu.tr

1. Introduction

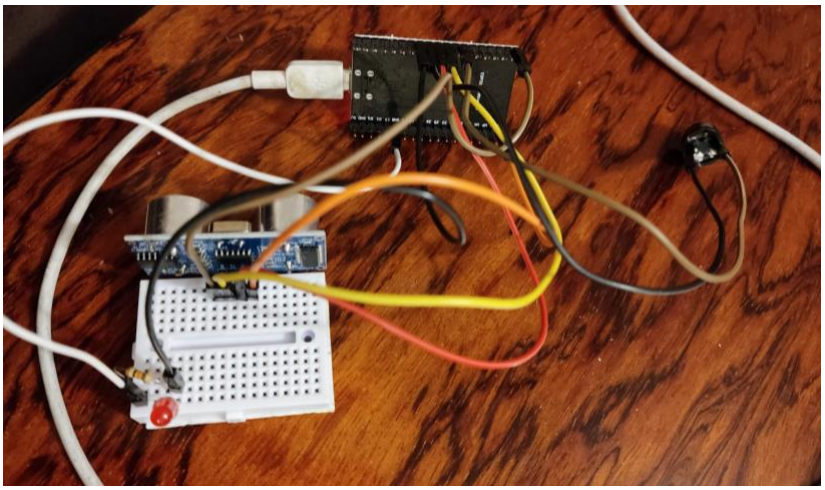
The concept of a smart home has become increasingly prevalent in our daily lives, aiming to provide convenience, efficiency, and enhanced security. One critical aspect of this smart home ecosystem is the security system. The need for a reliable, easily manageable, and technologically advanced home security system is more significant than ever due to the rising concerns over property safety and the desire for remote monitoring capabilities.

This project, "Smart Home Security System," addresses these needs by integrating modern technology such as the ESP32 Wi-Fi module to create a system that is not only responsive and reliable but also accessible remotely through a web interface. The primary problem it aims to solve is the lack of integration of affordable and easily accessible technology in the traditional home security systems, making them either too simplistic or prohibitively expensive and complex for the average user. This project bridges that gap, providing an efficient, cost-effective solution.

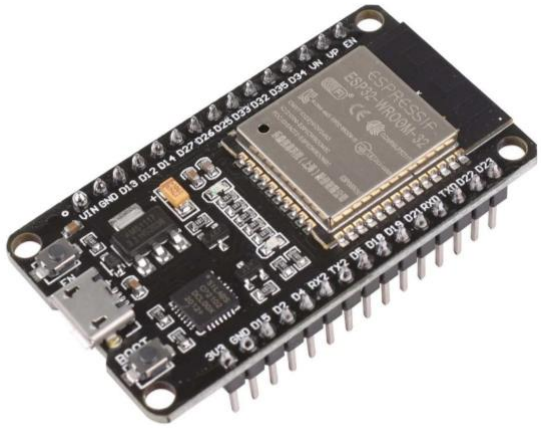


2. Hardware

The hardware components of the "Smart Home Security System" include an ESP32 Wi-Fi module, an ultrasonic distance sensor (HC-SR04), a buzzer, LEDs, and necessary connecting elements like a breadboard and jumper wires.



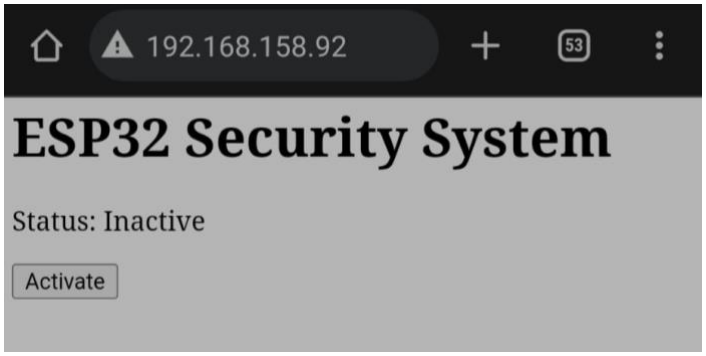
- **ESP32 Wi-Fi Module:** At the heart of the system is the ESP32 module, known for its dual-mode capabilities (Wi-Fi and Bluetooth). However, for this project, only its Wi-Fi functionality is utilized. The ESP32 acts as a microcontroller and connects to the home Wi-Fi network, enabling remote access and control. It is the central unit that communicates with other components and the web server.



- **Ultrasonic Distance Sensor (HC-SR04):** This sensor detects the presence of an object or intruder based on distance measurements. Its integration with the ESP32 allows the system to determine when an object is within a predefined range, triggering the alarm if necessary.

- **Buzzer and LEDs:** These components serve as the alert system. The buzzer emits a sound, and the LEDs light up when the system is triggered, providing a visual and auditory indication of a security breach.

- **Communication with Web Server:** The ESP32 module continuously communicates with the web server, sending sensor data and receiving commands. This setup allows real-time monitoring and control of the system via a web interface, making the security system interactive and user-friendly.



3. Software

The development of the software for this application involved programming the ESP32 to function as a web server and respond to specific conditions sensed by the ultrasonic sensor. The following key aspects were addressed in the software development:

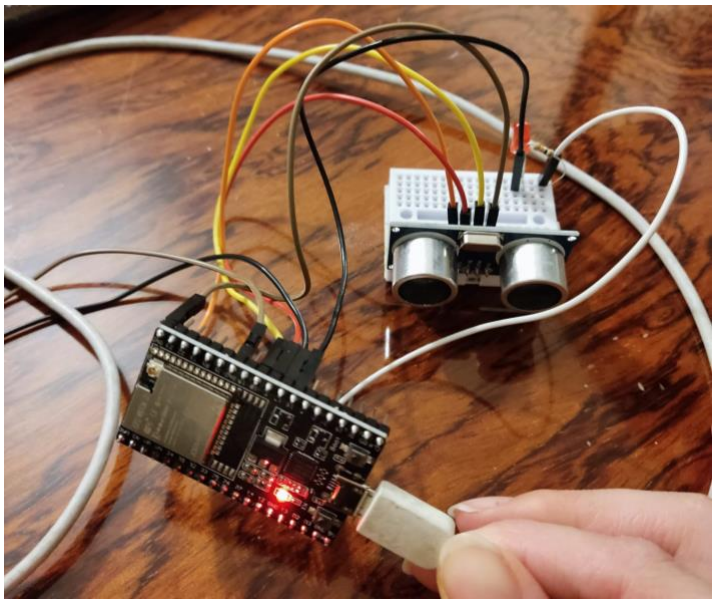
- **Web Server Implementation:** Utilizing the Wi-Fi capabilities of the ESP32, a web server was set up to enable remote access. This server presents a user interface where one can activate or deactivate the security system.

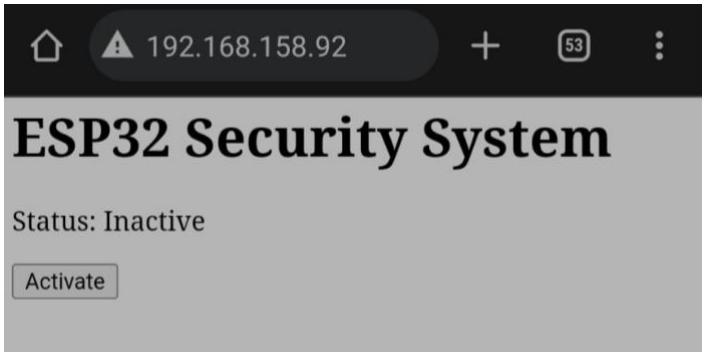
- **Sensor Data Processing:** The ESP32 is programmed to continuously read data from the ultrasonic sensor and determine if the distance measurement falls below a predetermined threshold, indicating a possible intrusion.

- **Alarm System Activation:** Upon detecting a potential security breach, the system activates the buzzer and LEDs. This code segment also ensures that the alarm is deactivated when the system is turned off or when no threat is detected.

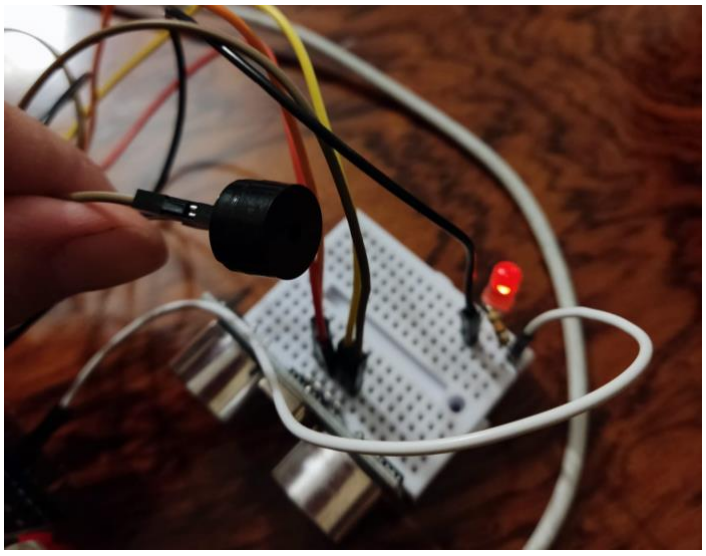
- **User Interface:** The software includes a simple, intuitive web-based interface allowing users to easily control the system's status and view real-time updates on the system's activity.

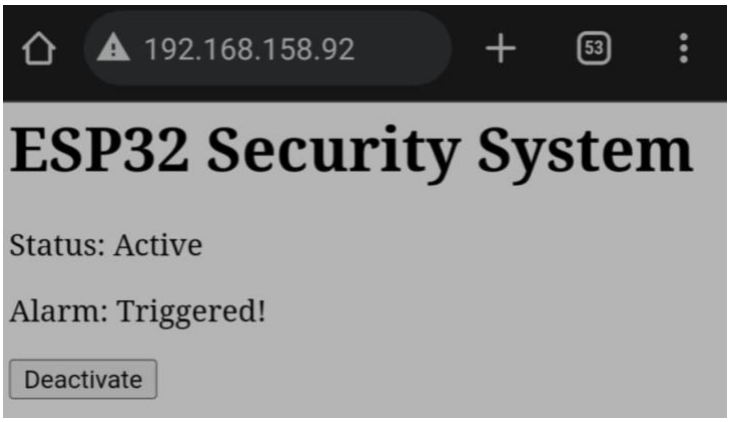
When system is inactive:





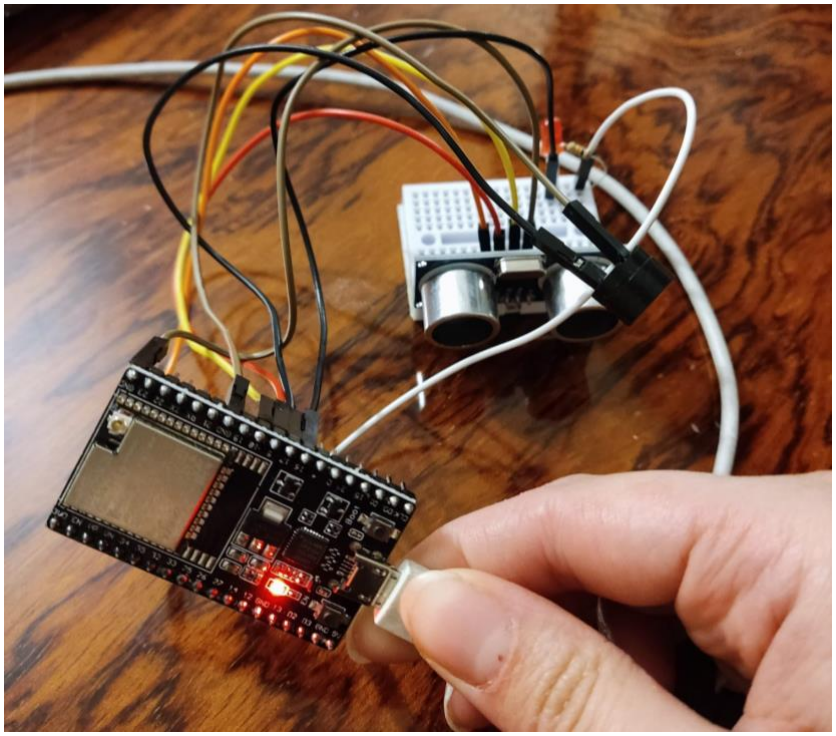
When system is active and alarm triggered:





```
Distance: 7
Distance: 10
Distance: 7
Distance: 10
Distance: 7
Distance: 10
```

When system is active and alarm not triggered:




```
Distance: 112
Distance: 76
```

Code:

```
#include <WiFi.h>
#include <WebServer.h>
#include <Ultrasonic.h>

// WiFi credentials
const char* ssid = "ssid"; // WiFi network name
const char* password = "pass"; // WiFi network password

// Server and sensor initialization
WebServer server(80);
Ultrasonic ultrasonic(5, 18); // Ultrasonic sensor pins (Trig, Echo)
const int buzzerPin = 17; // GPIO pin number for buzzer
const int ledPin = 16; // GPIO pin number for LED
bool systemActive = false; // Flag to check if the system is active
bool alarmTriggered = false; // Flag to check if the alarm has been triggered
```

```
void setup() {
  Serial.begin(115200);
  // Set pin modes
  pinMode(buzzerPin, OUTPUT); // Set buzzer pin as output
  pinMode(ledPin, OUTPUT); // Set LED pin as output

  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
  // Print the IP address
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP()); // Print the IP address

  // Handle root URL ("/")
  server.on("/", HTTP_GET, []() {
    String html = "<html><head><title>ESP32 Security System</title></head><body>";
    html += "<h1>ESP32 Security System</h1>";

    // Display current status of the system on the web page
    if (systemActive) {
      html += "<p>Status: Active</p>";
      // Show if alarm has been triggered
      if (alarmTriggered) {
        html += "<p>Alarm: Triggered!</p>";
      } else {
        html += "<p>Alarm: Not triggered</p>";
      }
      html += "<button onclick=\"location.href='/deactivate'\">Deactivate</button>";
    } else {
      html += "<p>Status: Inactive</p>";
      html += "<button onclick=\"location.href='/activate'\">Activate</button>";
    }

    html += "</body></html>";
```

```
    server.send(200, "text/html", html);
  });

  // Handle "/activate" URL
  server.on("/activate", HTTP_GET, []() {
    systemActive = true; // Activate the system
    alarmTriggered = false; // Reset alarm trigger
    server.sendHeader("Location", "/"); // Redirect to root
    server.send(303);
  });

  // Handle "/deactivate" URL
  server.on("/deactivate", HTTP_GET, []() {
    systemActive = false; // Deactivate the system
    // Turn off buzzer and LED
    digitalWrite(buzzerPin, LOW); // Turn off buzzer
    digitalWrite(ledPin, LOW); // Turn off LED
    server.sendHeader("Location", "/"); // Redirect to root
    server.send(303);
  });

  // Start the server
  server.begin();
}
```

```
void loop() {
  // Handle client requests
  server.handleClient();

  // Check if system is active
  if (systemActive) {
    long distance = ultrasonic.read(); // Read distance from ultrasonic sensor
    Serial.print("Distance: ");
    Serial.println(distance);
    // If distance is less than or equal to 20 cm, trigger the alarm
    if (distance > 0 && distance <= 20) {
      alarmTriggered = true; // Set alarm trigger
      // Activate buzzer and LED
      digitalWrite(buzzerPin, HIGH); // Activate buzzer
      digitalWrite(ledPin, HIGH); // Activate LED
    } else {
      // If distance is more than 20 cm, reset the alarm
      alarmTriggered = false; // Reset alarm trigger
      // Deactivate buzzer and LED
      digitalWrite(buzzerPin, LOW); // Deactivate buzzer
      digitalWrite(ledPin, LOW); // Deactivate LED
    }
  }
}
```

4. Limitations

- **Dependency on Local Wi-Fi Network:** The current design requires the ESP32 module to be connected to the same Wi-Fi network as the device accessing the web server. This constraint limits remote access capabilities, particularly when the user is outside the home network. In scenarios where the Wi-Fi network is down or the ESP32 is disconnected, the system becomes inaccessible remotely.

- **Limited Range of Ultrasonic Sensor:** The HC-SR04 ultrasonic sensor has a limited sensing range. Objects beyond its maximum range cannot be detected, potentially leaving certain areas unmonitored.

- **No Backup Power Source:** The system currently lacks a backup power solution. In the event of a power outage, the entire security system would become non-operational.

- **Encryption and Security Protocols:** The system's communication over the Wi-Fi network is not encrypted, which poses a risk of interception or unauthorized access.

5. Future Developments

To overcome these limitations and enhance the overall functionality and reliability of the system, the following future developments are proposed:

- **Integration with Cloud Services:** To allow remote access outside the local Wi-Fi network, integrating the system with cloud services can be considered. This would enable users to control and monitor their home security system from any location with internet access.

- **Inclusion of More Sensors and Cameras:** Adding additional types of sensors, like motion detectors, and integrating camera modules would provide comprehensive surveillance capabilities.

- **Implementation of Backup Power Solutions:** Incorporating a battery backup or alternative power sources would ensure that the system remains operational during power outages.

- **Enhanced Security Features:** Implementing encryption and secure communication protocols would protect the system from potential cyber threats. Additionally, incorporating user authentication for access to the web interface would add an extra layer of security.

- **Mobile Application Development:** Creating a dedicated mobile application would offer a more user-friendly and accessible interface for system control and monitoring.