**Pharos University in Alexandria جامعة فاروس بالإسكندرية**

**Faculty of Computer Science كلية علوم الحاسبات و الذكاء الاصطناعي**

Course: Robotics AI 302

application Project

*(*Design of *Smart Gate)*

**Under supervision of**

**Dr. Eng. Hany M. Arnaoot**

**Eng. Esraa Hamdy**

**By:**

**Mahytab Adel Aborisha**

1. **Project brief description:**

The Smart Gate project utilizes an ESP32 microcontroller along with an ultrasonic sensor and a servo motor to create an automated gate system. The system detects approaching objects using the ultrasonic sensor and controls the opening and closing of the gate using the servo motor. This project provides a simple and efficient solution for automating gates, offering convenience and security to users.

1. **Project used component (if any):**

**ESP32 Microcontroller**:

Serves as the central processing unit, responsible for interfacing with the ultrasonic sensor and servo motor, as well as executing the control logic.

**Ultrasonic Sensor**:

Detects the distance of approaching objects by emitting ultrasonic pulses and measuring the time it takes for the pulses to bounce back.

**Servo Motor**:

Controls the movement of the gate, allowing it to open and close based on the input received from the ESP32.

**Breadboard :**

Which used to connect all of them with jumper wires

1. **Project used software (if any):**

**Arduino IDE:**

The Integrated Development Environment (IDE) used for writing, compiling, and uploading code to the ESP32 microcontroller.

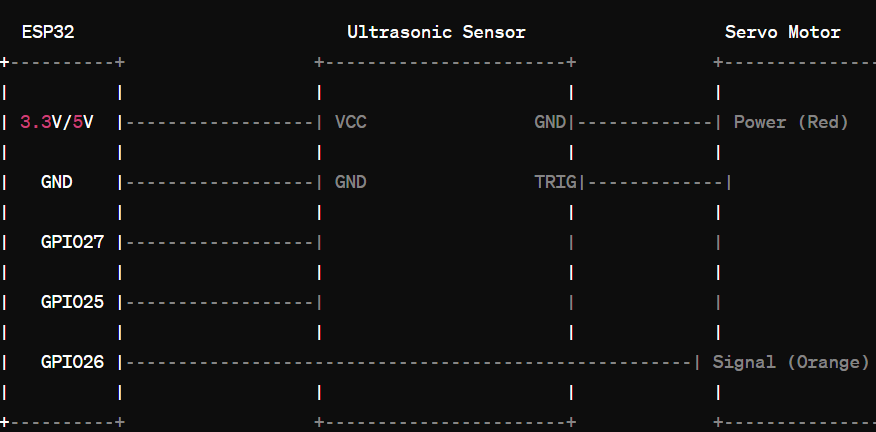
**ESP32Servo Library:**

This library provides functions to control servo motors connected to ESP32 boards. It allows the code to easily interface with the servo motor and control its position.

1. **Project execution Steps:**
2. **Hardware Setup:**
   * Assemble the required hardware components including the ESP32 microcontroller, ultrasonic sensor, servo motor, jumper wires, and breadboard.
   * Ensure all components are securely connected according to the provided wiring diagram, paying close attention to pin assignments and placement on the breadboard.
3. **Software Installation:**
   * Download and install the Arduino IDE from the official website onto your computer.
   * Install the necessary libraries, such as the ESP32Servo library, using the Arduino IDE Library Manager to facilitate code compilation and execution.
4. **Code Implementation:**
   * Open the Arduino IDE and create a new sketch.
   * Write the provided code into the sketch window, ensuring all code segments are accurately transferred.
   * Optionally, customize the code by adjusting parameters such as pin assignments (**TRIG\_PIN**, **ECHO\_PIN**, **SERVO\_PIN**) and distance threshold (**DISTANCE\_THRESHOLD**) to match your specific hardware configuration.
5. **Code Upload:**
   * Connect the ESP32 microcontroller to your computer via USB.
   * Select the appropriate board type and COM port from the Arduino IDE's **Tools** menu.
   * Click the "Upload" button in the Arduino IDE to compile and upload the code to the ESP32 microcontroller.
6. **Verification and Testing:**
   * Open the Serial Monitor in the Arduino IDE to monitor the output from the ESP32 microcontroller.
   * Place objects within the detection range of the ultrasonic sensor and observe the behavior of the servo motor.
   * Verify that the gate opens and closes appropriately based on the distance of the detected objects.
   * Use the serial monitor to analyze distance readings and troubleshoot any operational issues.
7. **Refinement and Optimization:**
   * Fine-tune the code and hardware setup as necessary to improve system performance and reliability.
   * Adjust parameters such as distance thresholds, servo motor angles, and sensor positions to optimize the functionality of the smart gate system.
8. **Deployment:**
   * Once satisfied with the performance and functionality, mount the hardware components in their final location for deployment.
   * Ensure all connections are secure and the system operates as expected before integrating it into the desired gate system configuration
9. **Main problems faced during building**

I had an issue with initializing the servo libraries.

1. **Schematic diagrams (if any)**



1. **Images of the Project**

**A close up of a device

Description automatically generated**

1. **Each student's responsibilities and contribution:**
2. **Student's contribution detailed description**
3. **First Student:**

Wrote c++ code , soldered and assembled the component

Verify connections: Double-check all connections to ensure they are correctly made and there are no loose wires.

Test hardware functionality: Ensure each compon Write code: Develop the code logic to control the smart gate system, including interfacing with the ultrasonic sensor and servo motor.

Upload code: Use the Arduino IDE to compile and upload the code to the ESP32 microcontroller.

Test software functionality: Verify that the code operates correctly by monitoring the system's behavior and response to different scenarios.

Debug and refine: Identify and troubleshoot any issues with the code or hardware setup, making adjustments as needed for optimal performance. ent functions as expected and is responsive to input signals.

1. **Student’s contribution summary table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Students Name** | **Software design** | **Wiring** | **Component selection** | **assembly** | **coding** |
| **Mahytab** | **•** | **•** | **•** | **•** | **•** |