Sonar Object Detector Project

Project Team Members-Group 2

Melike Tepeli 20COMP1012

Emin Kaan Eren 20COMP1032

Büşra Ablak 218CS2087

The Arduino Sonar Object Detection System is designed to detect objects within a 180-degree range using ultrasonic sensors. This project utilizes Arduino microcontrollers and ultrasonic sensors to provide a cost-effective and efficient solution for object detection. Main parts of the project are an arduino board, a servo motor, an ultrasonic sensor (HC-SR04), a Bluetooth module (HC-06),an LED(lights up if an object is detected), two buttons that control interrupt and timer and a bluetooth enabled device for controlling the servo motor.

Features:

Scanning: The sonar will rotate with the help of the control from the mobile device to scan an area, spanning 180° in front of the detector.

Feedback Mechanism: The system provides real-time feedback to the user through LED and User interface. Depending on the detected object's proximity, the system may activate a buzzer.

Distance Measurement: The ultrasonic sensor will measure the distance to the nearest object and display graphically.

Interrupt: To ensure that the LED lights up at certain intervals.

EEPROM: Keeps count of how many times the interrupt was performed

User Interface: A user interface to show the where the object is graphically.

Timer: Timer2 was used to enable the LED to turn on at certain intervals and also with the tone function.

Project Summary:

In our radar project, we utilized an Arduino Uno microcontroller along with various electronic components to create a simple yet effective radar system. The primary objective of our project was to scan a defined area and measure the distance to objects within that space.

This radar project successfully demonstrated the capabilities of Arduino Uno in creating a functional radar system using accessible electronic components. By combining hardware integration, coding skills, and problem-solving abilities, we were able to develop a practical solution for detecting objects within a designated area.

Hardware Requirements:

1. Arduino Uno: The main microcontroller board.

2. HC-SR04 Ultrasonic Sensor: To measure distances.

3. Micro Servo Motor: To rotate the sensor for scanning.

4. Buzzer: For auditory feedback.

5. Breadboard: To mount and connect the components.

6. Jumper Wires: For making electrical connections.

7. Power Source: Either USB connection to a computer or an external power supply for the Arduino.

8.HC-06 Bluetooth module: To connect to an android app.

Software Requirements:

1. Arduino IDE: For writing, compiling, and uploading code to the Arduino Uno.

2. Arduino Libraries: Libraries for controlling the servo motor and interfacing with the ultrasonic sensor.

Skills and Knowledge:

1. Basic Electronics: Understanding of electronic components and circuits.

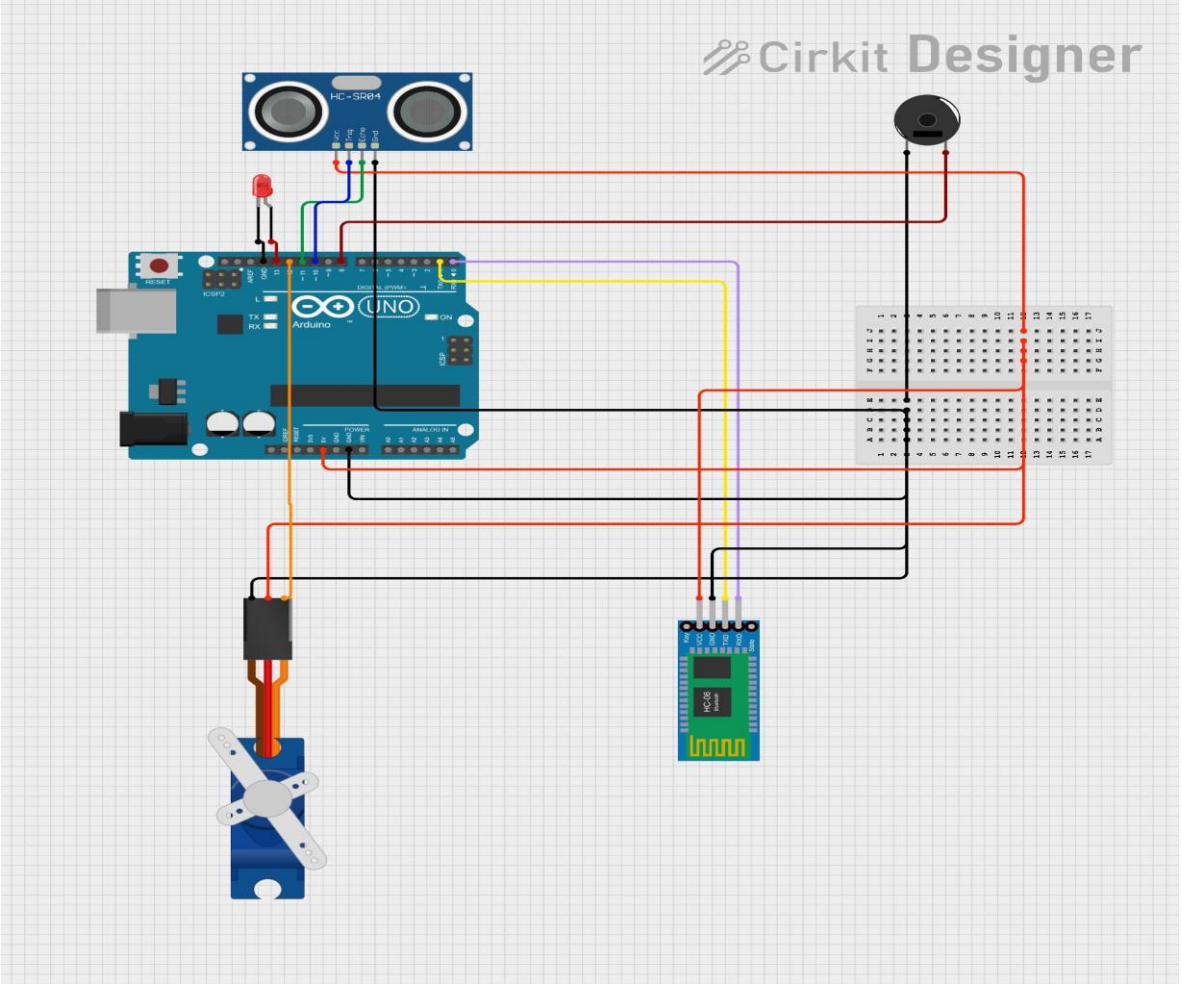
2. Programming: Proficiency in Arduino programming language (based on C/C++).

3. Sensor Integration: Ability to connect and interface with sensors like ultrasonic sensors.

4. Motor Control: Understanding servo motor control and PWM (Pulse Width Modulation).

5. Data Processing: Processing sensor data and implementing logic for distance measurements and alerts.

6. Troubleshooting: Skills to debug hardware and software issues.



1. **Preparation:**

Ensure all components are securely connected and the Arduino sketch is correctly uploaded to the board.

Have all necessary materials ready, including a laptop with the Arduino IDE, USB cable, and any additional presentation tools.

2. **Introduction:**

Briefly explain the purpose and goals of the radar project.

Describe the components used: Arduino Uno, HC-SR04 Ultrasonic Sensor, Micro Servo Motor, Buzzer, Breadboard,HC-06 Bluetooth Module and Jumper Wires.

3. **Initial Power-On Test:**

Connect the Arduino Uno to a power source (USB or external power supply).

Ask the class to observe the power LED on the Arduino board to confirm it's powered on.

Connect Arduino to Android phone via bluetooth module

Expected Outcome: The power LED should be lit, indicating the board is receiving power and phone shows its connected

4. **Integration Test:**

Upload the complete radar project code , integrating the servo motor and ultrasonic sensor.

Demonstrate how the servo motor rotates the ultrasonic sensor, taking distance measurements at different angles and will give some tones about measurements.

Test the buttons to see if the Bluetooth is working.

Expected Outcome: When the touch the button in app servo motor should rotate, and the ultrasonic sensor should measure distances at different angles, displaying the results on the Serial Monitor and gives sound from buzzer.

Project Workflow:

1. Hardware Setup:

- We carefully connected the ultrasonic sensor, servo motor, buzzer, and Arduino Uno on a breadboard, ensuring all connections were secure and properly configured.

2. Coding:

- Developing the code was a crucial aspect of our project. We wrote Arduino code to control the servo motor's movement, trigger distance measurements with the ultrasonic sensor, and interpret the collected data.

- Additionally, we implemented logic to process the distance measurements and trigger alerts or notifications through the buzzer when predefined conditions were met.

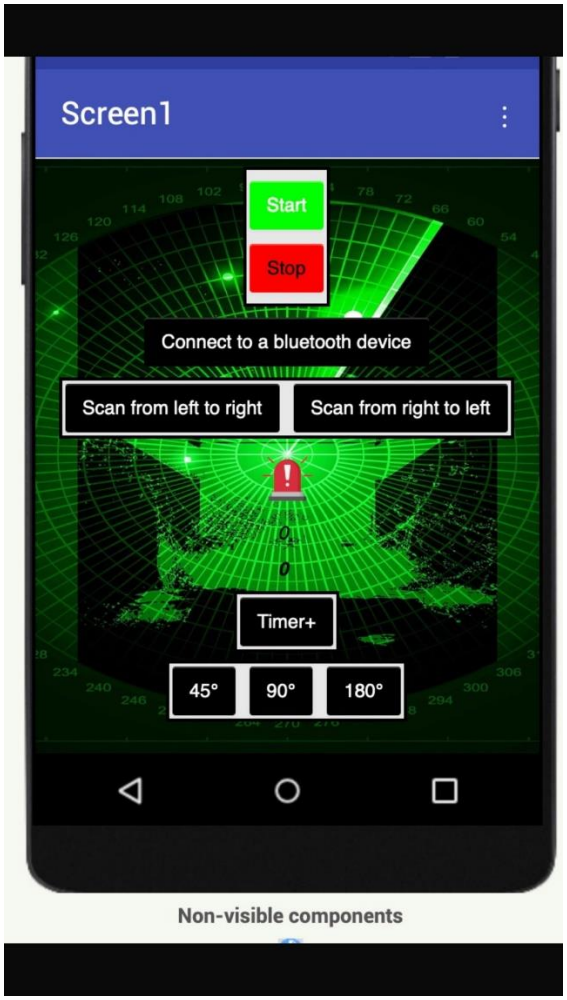
- Also we use Processing for show the results of radar.

3. Testing and Refinement:

- After completing the hardware setup and coding, we extensively tested our radar system to ensure its functionality and accuracy.

- We iteratively refined our code and made adjustments to the hardware setup as needed, addressing any issues or discrepancies encountered during testing.

App Dizayn Part



Aplication Features;

- Start and stop the servo motor.

- Make the motor turn to specific degrees.

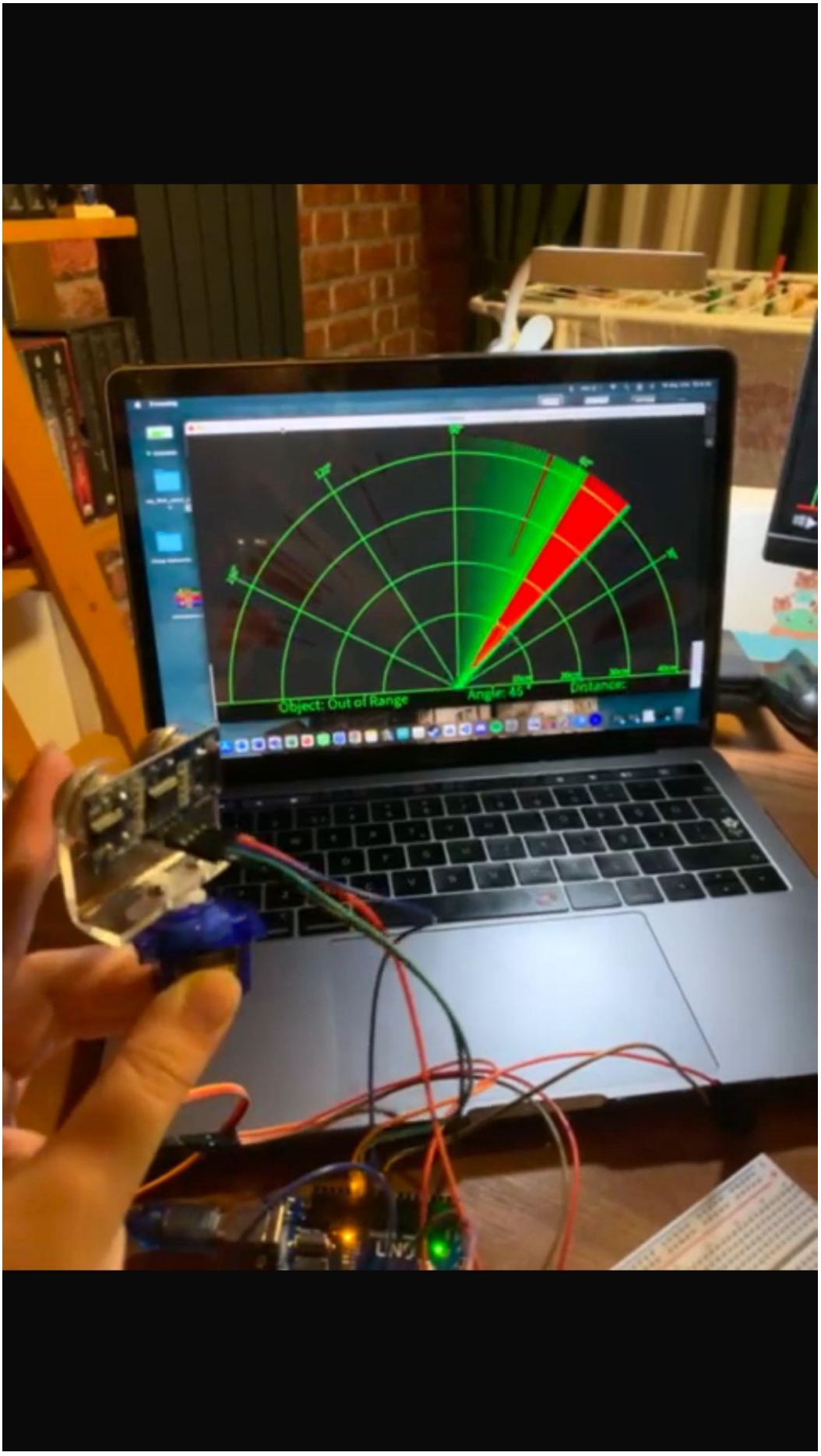
- Checking the operation of the LED at certain time intervals.

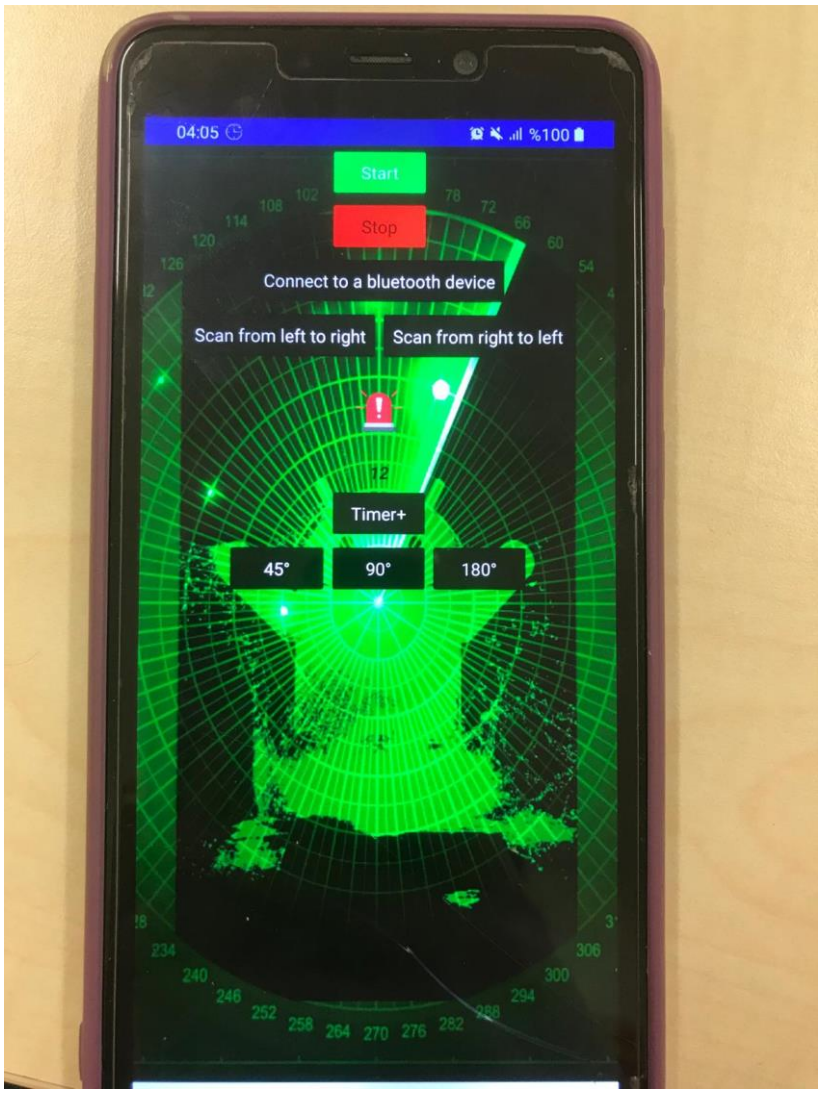
- Display nearby threats on the screen.

- Move the motor to the right or left.

- A button is available to connect the device to Bluetooth.

RESULTS





Summary & Explanation

This Arduino code is a program that controls various hardware components. The code allows controlling a servo motor based on Bluetooth commands and measures distances using an ultrasonic sensor to control an LED and a piezo buzzer based on specific distance ranges.

Below is a detailed explanation of the code:

Libraries Used:

1. Servo.h:Used to control the servo motor.
2. SoftwareSerial.h: Used for software serial communication, connecting to the Bluetooth module.
3. TimerOne.h: Used for timer operations.

Hardware Connections:

- Bluetooth Module: Connected to RX and TX pins 0 and 1.

- Ultrasonic Sensor (HC-SR04): Trig pin connected to pin 10, Echo pin connected to pin 11.
- LED: Connected to pin 13.
- Piezo Buzzer: Connected to pin 8.
- Servo Motor: Connected to pin 12.

Variables and Constants:

- trigPin, echoPin, ledPin, piezoPin: Define the pin numbers for the connected components.
- duration, distance: Variables used for measuring distance.
- notes: Frequencies that the piezo buzzer will play for different distances.
- myServo: Servo motor object.

setup() Function:

1. Sets pin modes.
2. Initializes serial and Bluetooth communication.
3. Attaches the servo motor to the specified pin.

loop() Function:

- Reads commands from Bluetooth and sends them to the switch part.

Switch part :

Executes actions based on the commands received from Bluetooth:

- `'S'`: Starts the servo motor.
- `'T'`: Stops the servo motor.
- `'R'`: Resets and restarts the servo motor.
- `'Q'`: Moves the servo motor in the reverse direction.
- `'U'`: Moves the servo motor from 0 to 45 degrees.
- `'O'`: Moves the servo motor from 0 to 90 degrees.
- `'P'`: Moves the servo motor from 0 to 180 degrees.
- `'M'`: Stops the servo motor and sets up a timer to run the servo motor every 5 seconds.

calculateDistance() Function:

- Measures the distance using the ultrasonic sensor and returns the result in centimeters.

handleBeepAndLed() Function:

- Controls the piezo buzzer and LED based on the distance:
  - If the distance is greater than 40 cm, the LED turns off and the buzzer does not sound.
  - If the distance is less than 40 cm, the buzzer plays different tones based on the distance and the LED blinks.

ledCallback() Function:

- It inverts the value of the toggle variable and updates the state of the ledPin pin with this new value. So, every time the function is called it changes the on or off state of the LED.

This code controls a servo motor's movement and an LED and piezo buzzer based on distance measurements using commands received via Bluetooth. When the distance is less than 40 cm, the piezo buzzer plays specific tones, and the LED lights up. The servo motor moves at different angles, measuring the distance and responding accordingly.