BlindCrown

Bucur Raul-Paraschiv, 10LF221
Andrei-Cristian Ilies, 10LF222
Macovei Alexandru-Fabian, 10LF222

Hardware Scheme

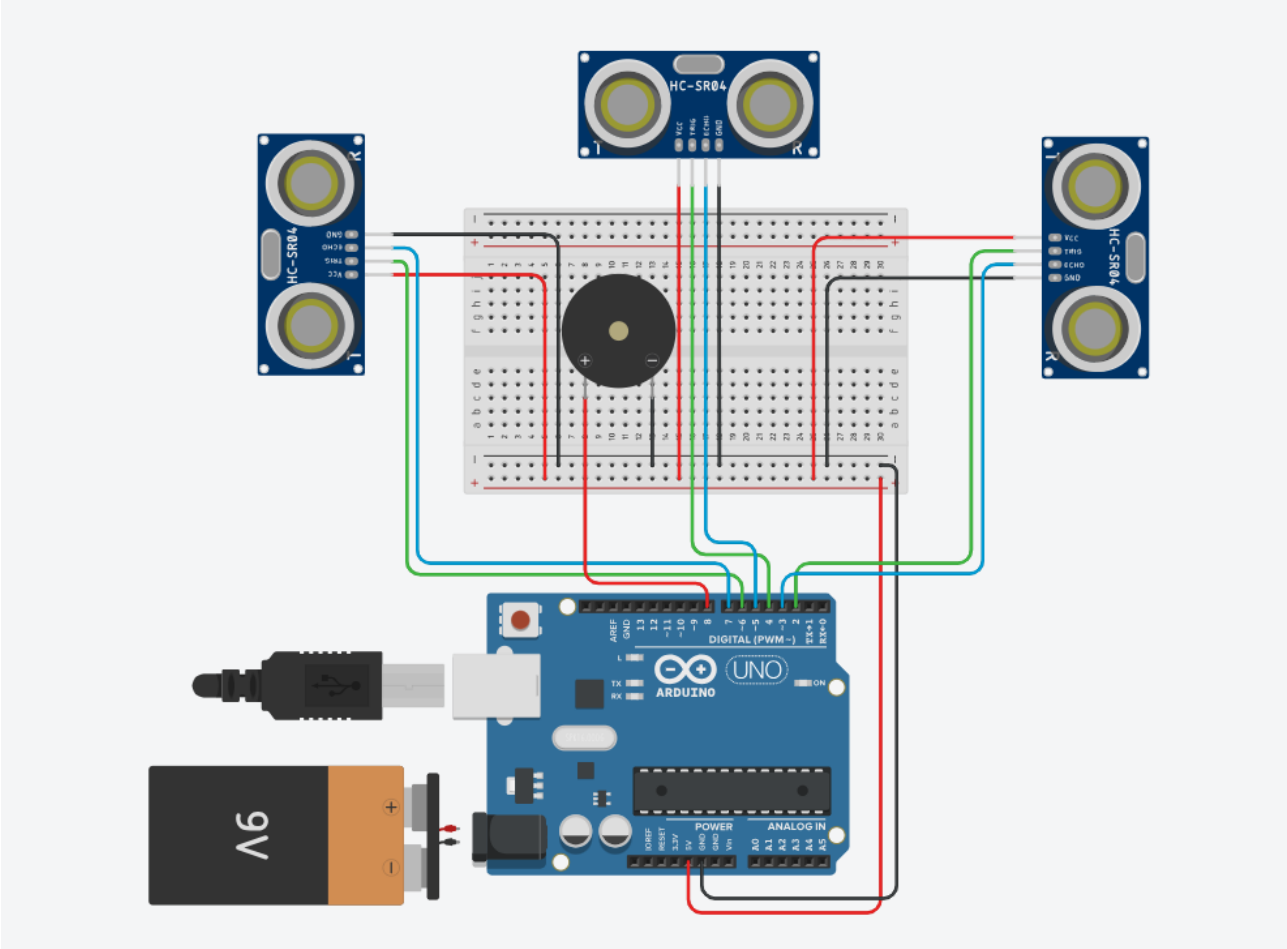

Figure 1: Hardware Scheme

The hardware scheme shows an Arduino Uno connected to a breadboard, which distributes power and ground to multiple components. We used a 9V battery to supply power to the Arduino through the DC jack. The 5V pin is connected to the breadboard's positive power rail and its GND pin is connected to the ground rail.

We used 3 HC-SR04 ultrasonic sensors and connected them to the breadboard, with their VCC pins linked to the positive rail and their GND pins linked to the ground rail. Each sensor's TRIG and ECHO pins are connected to digital pins on the Arduino for control and data transfer.

The buzzer is placed on the breadboard, with one pin connected to a digital pin on the Arduino and the other pin connected to the ground.

The wiring ensures that power is distributed from the Arduino to all components, and communication between the sensors and the Arduino is handled through dedicated digital pins.

Software Scheme

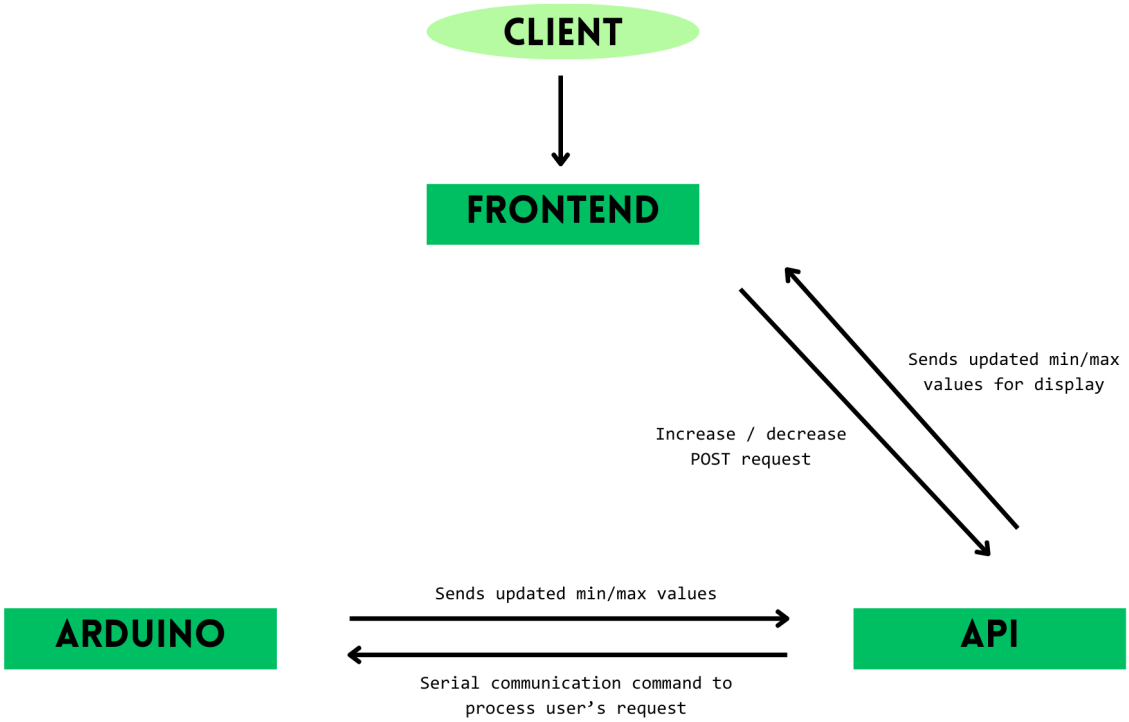

Figure 2: Software Scheme

The software architecture involves interaction between a client, frontend, API, and Arduino UNO device. The client communicates directly with the frontend, which acts as an intermediary for sending and receiving data.

Frontend sends POST requests to the API to adjust the parameters by increasing or decreasing the minimum and maximum values.

The API interfaces with the Arduino UNO, sending serial communication commands to implement the user's requests. Once the Arduino processes these commands, it sends back the updated minimum and maximum values to the API, which relays them to Frontend for display.

Concrete, specific functionalities of the project

- **Obstacle Detection:** Using the 3 HC-SR04 ultrasonic sensors, the system can detect nearby obstacles and measure their distance.
- **Auditory Feedback:** The buzzer provides real-time sound alerts to indicate proximity to obstacles, enhancing navigation for visually impaired users.
- **Customizable Range Settings:** The software allows users to adjust the minimum and maximum detection range via the frontend interface.
- **Serial Communication:** Communication between the Arduino and the software ensures smooth transmission of sensor data and user commands.
- **Dynamic Updates:** The API processes user requests to modify detection parameters and reflects changes in real-time on the frontend.

Concrete, specific utility elements

- **Battery-Powered Operation:** The 9V battery enables portability and independent usage without reliance on external power sources.
- **User-Friendly Frontend:** A client-facing interface allows seamless interaction for adjusting settings and monitoring system status.
- **Compact and Lightweight Setup:** The minimal hardware components make the system convenient for everyday use.

Video

`https://drive.google.com/file/d/1h4pR9THbZHvez_wzsyN64VEmXYrhJPdm/view?usp=sharing`