

Wi-Fi Enabled Ultrasonic Ranging Device

Table of Contents

1. Introduction.....	3
2. Let's Get to Work!	4
2.1 Project-setup:.....	4
2.1 Here's how the system works:.....	4
2.1.1 Hardware Setup:.....	4
2.1.2 Data Transmission:	4
2.1.3 Data Reception:	4
2.1.4 Data Display:	5
3. What Do We Need?.....	5
3.1 Hardware.....	5
3.2 Software.....	5
4. What Do We Achieve?.....	5
5. Skills Needed:	5
6. Circuit diagram:	5
7. Tasks in hand.....	7
8. Source code	7
9. Source code explanation	7
10. Use cases.	7
11. Pros/cons	7
12. Conclusion	8

1. Introduction

The "Wi-Fi Enabled Ultrasonic Ranging Device" project aims to develop a wireless distance measurement system using an ultrasonic sensor HC-SR04, an ESP-32 chip, and the UDP protocol over Wi-Fi. This project enables real-time transmission of distance data from the sensor to a remote receiver, which can be a client socket code written in C programming. The receiver can be executed on a Linux terminal or an Android phone terminal using the Termux app, as long as they are connected to the same local Wi-Fi network.

By leveraging the capabilities of the ESP-32 chip and its built-in Wi-Fi module, the system establishes a wireless connection and utilizes the UDP protocol for data transmission. The ultrasonic sensor HC-SR04 measures distances by emitting ultrasonic waves and calculating the time it takes for the waves to bounce back. The measured distance is then transmitted wirelessly to the receiver within the local Wi-Fi network, allowing for remote monitoring and analysis.

It's important to note that the data transmission is limited to the local Wi-Fi network and is not connected to the internet. This ensures data privacy and security, as the distance data remains within the confines of the local network.

The receiver, implemented as a client socket code in C programming, can be executed on a Linux terminal or an Android phone terminal using the Termux app. This flexibility allows for easy integration with existing systems and provides a convenient interface for receiving and processing the distance data. The client socket code establishes a connection with the ESP-32 chip acting as the server, enabling seamless communication and data retrieval.

The primary objective of this project is to provide a reliable and efficient means of distance measurement that eliminates the need for physical connections. This wireless solution offers flexibility in terms of sensor placement and enables real-time monitoring of distances in various applications such as robotics, automation, security systems, and more. In this project I will leave it to your imagination on how you can use this.

Throughout this design document, we will outline the system architecture, hardware components, software implementation, and testing procedures required to successfully develop the "Ultrasonic Wi-Fi Distance Transmitter" project, including the setup and execution of the client socket code on a Linux terminal or an Android phone terminal using the Termux app within the local Wi-Fi network..

2. Let's Get to Work!

2.1 Project-setup:

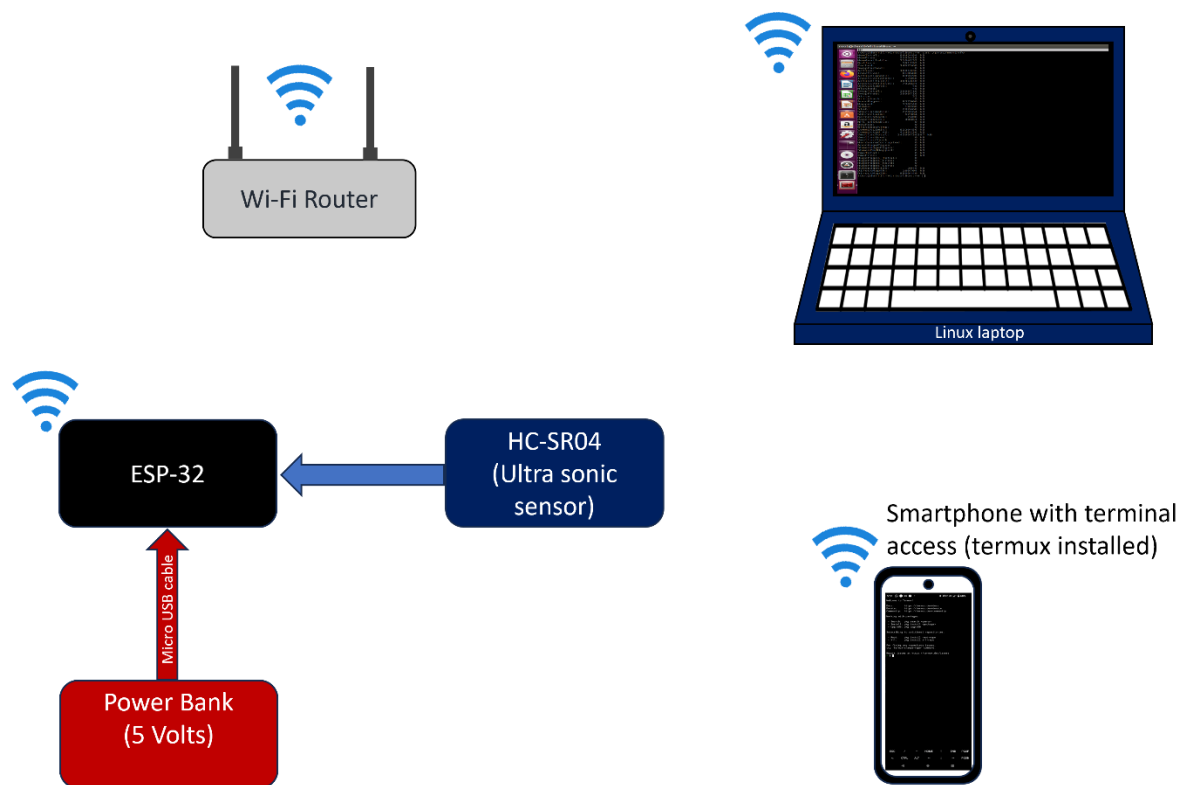


Figure 1: Project Setup

In our project, we establish a seamless connection between our ultrasonic sensor and a Wi-Fi network. This connectivity is facilitated by an ESP-32 microcontroller, renowned for its capability to handle Wi-Fi packets effectively.

2.1 Here's how the system works:

2.1.1 Hardware Setup:

- We connect the HC-SR04 sensor to the ESP-32 microcontroller, creating a robust hardware foundation.
- A dedicated code is crafted to read the ultrasonic data, which is then uploaded onto the microcontroller.

2.1.2 Data Transmission:

- The microcontroller takes on the pivotal role of not only reading but also formatting and transmitting the sensor data into the WiFi network.
- To ensure efficient data transfer, we employ UDP broadcast on a specified port number. This ensures that the data flows seamlessly within the network.

2.1.3 Data Reception:

- On the receiving end, whether it's a Linux laptop or an Android phone, a client code is implemented to register itself to the designated port number.

- The client code is responsible for accepting all incoming data, thus establishing a vital link between the sensor and the terminal.

2.1.4 Data Display:

- In this iteration of the project, our primary objective is to display the received data on the terminal. This foundational step lays the groundwork for further data processing and subsequent actions in future project phases.

3. What Do We Need?

3.1 Hardware

- A Linux laptop or an Android phone with a terminal installed.
- An ESP-32 Microcontroller.
- An HC-SR04 Ultrasonic sensor.
- A couple of DuPont/jumper cables.
- Optional: Power bank, casing, and other accessories.

3.2 Software

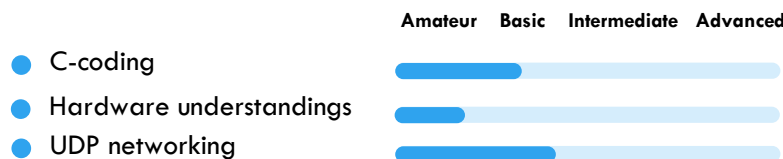
- Arduino IDE
- Gcc/clang or a similar C code compiler for your laptop or phone.

4. What Do We Achieve?

The "Wi-Fi Enabled Ultrasonic Ranging Device" project offers an accessible and engaging platform for beginners to delve into the world of embedded programming. It provides a hands-on learning experience, fosters understanding of wireless communication and sensor integration, and encourages creativity and innovation.

5. Skills Needed:

- Software development in C.
- Basic hardware understanding.
- A grasp of network protocols, specifically UDP in this case.
- Time management to complete the project efficiently.



6. Circuit diagram:

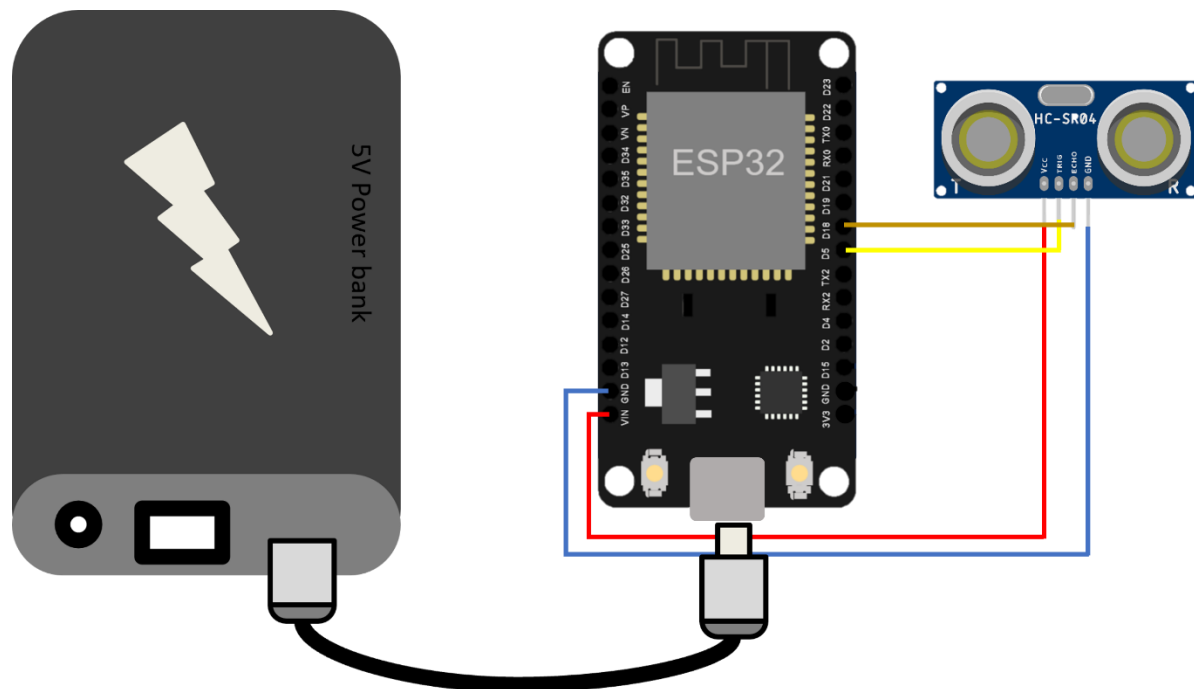


Figure 2: Circuit diagram

7. Tasks in hand

Task	Time to be spent (hrs.)	Efforts	Percentage completion
Get components.	1	Online/Offline shopping	19%
Download codes.	0.25	Browse web/git	24%
Install software.	1	Browse web. install applications. Linux command line application installation	43%
Edit and Compile code.	0.5	Arduino IDE compilation gcc/clang command line compilation	52%
Assemble circuit.	0.5	Connect DuPont cables to US sensor while looking at the circuit diagram present in the document.	62%
Create a test environment.	1	A worktable with all the hardware/software in one place. Upload ESP-32 code through Micro-USB data cable.	82%
Debugging	1 (subjective)	Subjective	99%
Deploy on field.	Subjective	Subjective	

8. Source code

9. Source code explanation

10. Use cases.

11. Pros/cons

12. Conclusion

In conclusion, the " Wi-Fi Enabled Ultrasonic Ranging Device " project introduces a seamless way to harness ultrasonic sensor data wirelessly on Linux terminals. By combining hardware and software expertise, this project offers valuable learning experience for beginners in the embedded systems field. With a clear block diagram, source code explanation, and insights into its use cases and limitations, this project aims to empower enthusiasts in the world of ultrasonic data acquisition and analysis.