Ultra-Sonic UDP terminal feed

Table of Contents

[1. Introduction 3](#_Toc149761772)

[2. Let’s Get to Work! 3](#_Toc149761773)

[2.1 Project-setup: 3](#_Toc149761774)

[2.1 Here's how the system works: 3](#_Toc149761775)

[2.1.1 Hardware Setup: 3](#_Toc149761776)

[2.1.2 Data Transmission: 4](#_Toc149761777)

[2.1.3 Data Reception: 4](#_Toc149761778)

[2.1.4 Data Display: 4](#_Toc149761779)

[3. What Do We Need? 4](#_Toc149761780)

[3.1 Hardware 4](#_Toc149761781)

[3.2 Software 4](#_Toc149761782)

[4. What Do We Achieve? 4](#_Toc149761783)

[5. Skills Needed: 4](#_Toc149761784)

[6. Circuit diagram: 5](#_Toc149761785)

[7. Tasks in hand 6](#_Toc149761786)

[8. Source code 6](#_Toc149761787)

[9. Source code explanation 6](#_Toc149761788)

[10. Use cases. 6](#_Toc149761789)

[11. Pros/cons 6](#_Toc149761790)

[12. Conclusion 6](#_Toc149761791)

# Introduction

Welcome to the "Ultra-Sonic UDP Terminal Feed" project, where we embark on a journey to seamlessly transfer ultrasonic sensor data wirelessly to any Linux terminal via a Wi-Fi network. This cutting-edge solution leverages the power of UDP (User Datagram Protocol) to ensure swift and efficient data streaming. Say farewell to the constraints of wired connections and embrace a new era of flexibility and convenience in acquiring, analyzing, and utilizing ultrasonic data.

# Let’s Get to Work!

## Project-setup:

A screenshot of a computer

Description automatically generated

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.

* 1. Here's how the system works:

### 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.

### 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.

### 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.

### 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.

# What Do We Need?

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

# What Do We Achieve?

This project provides a simple yet effective hardware-software fusion, making it accessible and engaging for beginners interested in the embedded field.

# Skills Needed:

C-coding

Hardware understandings

UDP networking

**Amateur Basic Intermediate Advanced**

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

# Circuit diagram:

A computer chip with wires connected to it

Description automatically generated with medium confidence

Figure : Circuit diagram

# 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 filed. | Subjective | Subjective |  |

# Source code

# Source code explanation

# Use cases.

# Pros/cons

# Conclusion

In conclusion, the "Ultra-Sonic UDP Terminal Feed" 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.