University of Connecticut

Development and Implementation of a 2D Ultrasonic Sensor System Using HC-SR04

Group 4: Luke Pepin, XXX, Antoine El Hani, XXX
NIUVT - Research 101
Professor Kevin Zmetra
May 1, 2024

Abstract

This report investigates the Development and Implementation of a 2D Ultrasonic Sensor System Using HC-SR04 Technology: Hardware Setup, Software Programming, and Performance Analysis

I. Introduction

The HC-SR04 ultrasonic distance sensor module is widely recognized for its applications in proximity sensing projects, particularly in one dimension. By emitting ultrasonic waves and measuring their reflection time, the HC-SR04 accurately calculates distances without requiring direct contact, making it an indispensable component for projects demanding precise distance measurements within a range of 2 centimeters to 4 meters.

This report delves into the comprehensive development and deployment process of a 2D Ultrasonic Sensor utilizing the HC-SR04 module with Arduino. It encompasses hardware and software configuration, calibration, testing procedures, and thorough data analysis. By meticulously detailing each step, this report provides an in-depth overview of the procedures essential for the successful creation of a 2D Ultrasonic sensor, as part of our work for NIUVT Research 101.

II. Hardware Setup

The hardware comprises four essential components: one Arduino Mega 2560 Rev3 microcontroller board, one 830 Tie-Points Breadboard 1PC, and two HC-SR04 Ultrasonic sensors to enable precise 2D distance measurement. These components are interconnected using a multitude of wires, ensuring seamless integration and efficient data transmission.

The Arduino Mega 2560 Rev3 microcontroller board serves as the central processing unit in a 2D ultrasonic sensor setup, orchestrating the entire operation. It manages the acquisition of ultrasonic data from the HC-SR04 sensors, executes the programmed algorithm for distance calculation, and facilitates communication with external devices, allowing the received sensor data to be exported for comparison or further analysis.

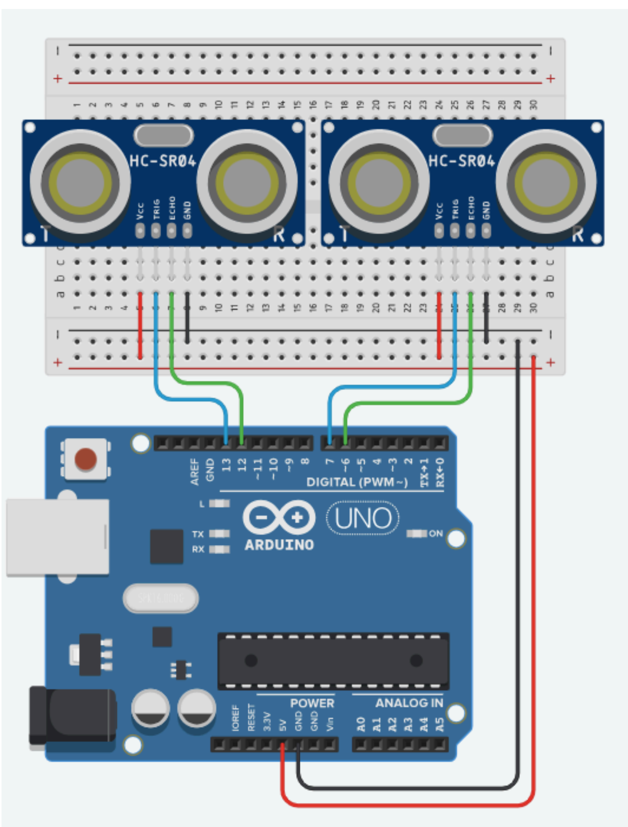
The 830 Tie-Points Breadboard 1PC serves as a platform for assembling the electronic circuit without the need for soldering. It allows components to be easily connected and rearranged, facilitating rapid prototyping and experimentation in electronic projects. With its grid of tie-points, it provides a convenient layout for organizing, testing and connecting the Arduino microcontroller and ultrasonic sensors.

The HC-SR04 sensor is an ultrasonic distance measuring device commonly utilized in robotics, automation, and proximity sensing applications. It is composed of two ultrasonic transducers, one which emits ultrasonic pulses and another which is a receiver which listens for reflected waves. It measures the time it takes for the pulses to bounce off an object and return to the sensor. With its high accuracy and a detection range of 2 centimeters to 4 meters, the HC-SR04 enables precise distance measurement in both indoor and outdoor environments.

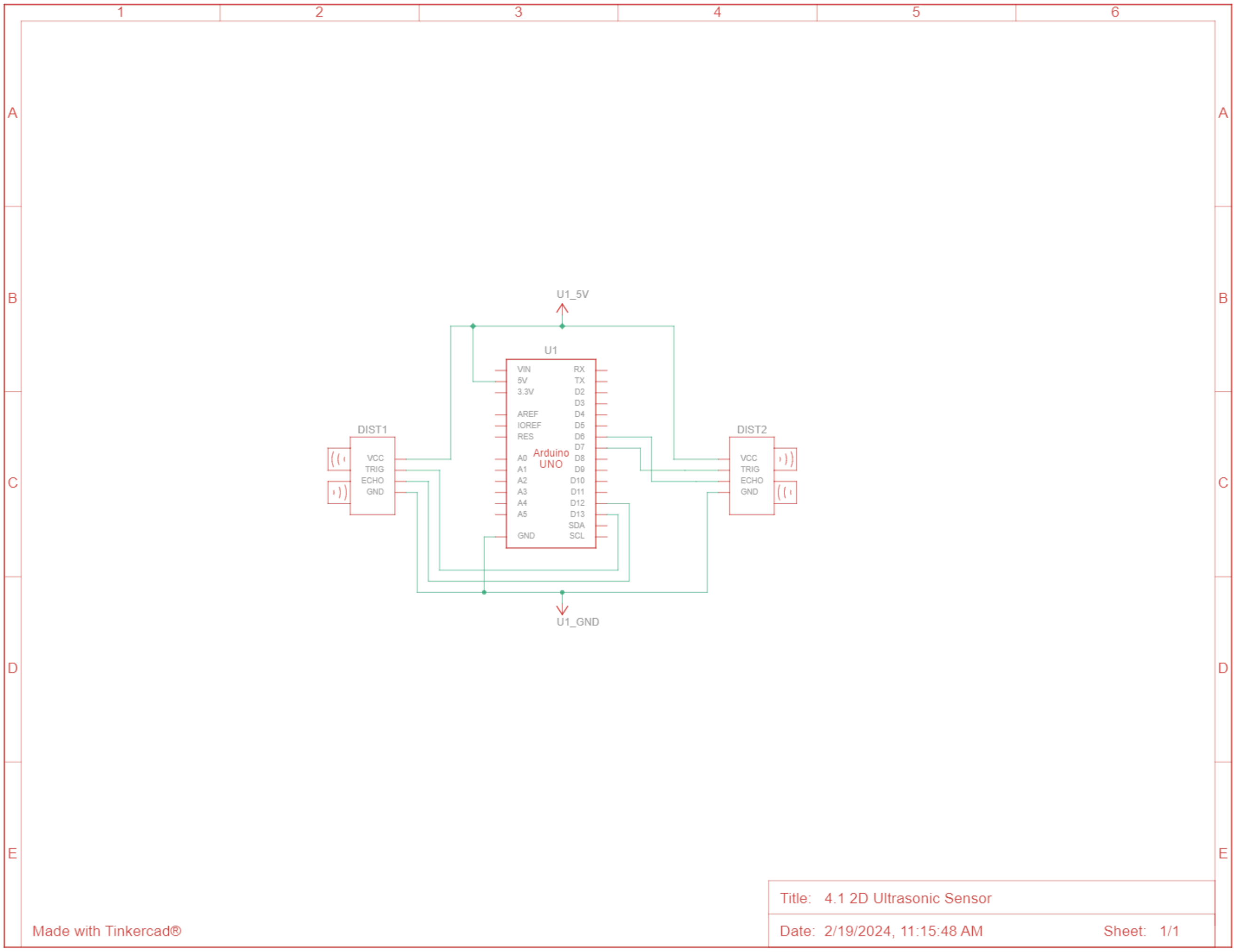
In the system, two separate HC-SR04 sensors are employed to establish a grid for determining both X and Y coordinates of the target. The integration of multiple HC-SR04 sensors allows for triangulation, thereby enhancing the precision and reliability of the target's position determination. By merging data from both sensors, the system can calculate not only the distance but also the direction of the target relative to the sensor array, thereby facilitating advanced tracking and navigation functionalities.

Lastly of the components, wires play a crucial role in connecting various components within the circuit, facilitating the transmission of electrical signals and power. Their flexibility and conductivity ensure seamless integration, enabling the functioning of the electronic system with reliability and efficiency.

Circuit View:



Schematic View:



(Please note in the schematic view above the Ultrasonic sensors are not realistically placed, in order to properly function the system requires the sensors to be facing the same direction.)

III. Software Setup

All software utilized within the 2D ultrasonic system is exclusively developed and executed via the Arduino IDE, a specialized platform tailored for programming Arduino microcontrollers. This IDE offers a user-friendly environment for crafting, compiling, and deploying code into Arduino boards, streamlining the creation process for diverse electronic projects. Leveraging its comprehensive library support and intuitive workflow, the Arduino IDE proves highly adept at constructing resilient applications for the system's operations.

This specific project does not incorporate any external libraries. All functionalities are developed and implemented independently within the project framework.

The provided code, "2DUltrasonicTester.ino," serves as the core program for operating the Ultrasonic sensors and collecting data. It comprises three essential functions—readUltrasonicDistance, setup, and loop—each integral to the proper execution of the code.

readUltrasonicDistance - Is both straightforward and intricate. Its purpose is to measure distance using an ultrasonic sensor, requiring the input of a triggerPin and an echoPin. Ultrasonic sensors operate by emitting high-frequency sound waves and then assessing the time it takes for these waves to rebound after encountering an object. Subsequently, the function returns the duration of the sound wave travel time in microseconds.

setup - Is an inherent part of the Arduino IDE, executing only once at the commencement of the code. Its primary role is to initialize various parameters and settings for the program. In this particular code segment, it is responsible for regulating the pace of the code execution. It prompts the user to input the distance between sensors in inches, ensuring that the input is a valid integer. Once a proper integer is provided, the function echoes back the distance to the user and proceeds to execute the main function loop.

loop - Is where the main workload of the algorithm is executed, consisting of two distinct parts:

> Part One, repeatedly executes the readUltrasonicDistance function for both sensors. The output, initially in microseconds, is converted to inches. In case of a bad reading where a non-number is returned, the function skips the reading and retries. If 10 consecutive bad readings occur, the function exits. Utilizing trigonometry, the algorithm calculates the x and y coordinates of the target and stores them in separate arrays. This process is iterated 10 times to ensure accuracy.
>
> Part Two, statistically analyzes the 2 arrays of x and y values (10 sets of coordinates) for user interpretation. It computes the minimum and maximum values of both x and y, the average x and y values, and finally, the standard deviation of both x and y. These statistical metrics provide a visual representation of the data, enabling the user to gain insights into the readings.

Coding practices play a crucial role in ensuring the reliability, maintainability, and efficiency of software systems. In the development of the 2D ultrasonic system, adherence to established coding principles has been paramount.

Firstly, modularity and encapsulation have been prioritized throughout the codebase. Functions such as `readUltrasonicDistance` are designed to perform specific tasks, promoting code reuse and readability. Additionally, the separation of concerns between setup and loop functions enhances clarity and simplifies debugging.

Secondly, error handling mechanisms are implemented to maintain the robustness of the system. The code gracefully handles bad readings from the ultrasonic sensors, ensuring continuous operation even in less-than-ideal conditions. By incorporating retry mechanisms and exit conditions, the system can effectively recover from transient failures.

Moreover, the use of meaningful variable names, comments, and descriptive function documentation fosters comprehensibility and facilitates collaboration among developers. Clear and concise code not only accelerates the development process but also eases future modifications and enhancements.

In conclusion, the adherence to sound coding practices underscores the commitment to delivering a dependable and efficient 2D ultrasonic system. By leveraging the capabilities of the Arduino IDE and adhering to established software engineering principles, the developed codebase exhibits resilience, maintainability, and scalability, laying a solid foundation for future iterations and enhancements.

IV. Calibration and Testing

Methods for calibrating the sensor. Testing procedures to verify accuracy and reliability.
Discussion of potential challenges and troubleshooting steps.
Independent Variables

- Distance between sensors
- Target distance
- Microsecond delays
- Target types

V. Results

Presentation of experimental data.
Comparison of measured distances with expected values.
Analysis of any discrepancies and potential sources of error.

VI. Conclusions and Recommendations

This section will discuss the recommendation and conclusions we make with regard to the results of the experiments that were discussed in the previous section

REFERENCES