

Acoustic Trilateration Search and Rescue Using Swarm Robotics

A Major Qualifying Project
Submitted to the Faculty of
Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree in Bachelor of Science
in
Computer Science and Robotics Engineering
By

Corey Aday
Nidhi Diwakar
Dan Murray

Date: April 28, 2016
Project Advisor:
Professor Michael Ciaraldi

Abstract

The goal of the project is to design and build a robotic system able to locate a sound source via trilateration between multiple mobile robots on the same field, overseen by a master controller. As they navigate the test environment, these robots will demonstrate time difference of arrival (TDOA)-based localization and communication over a wireless network. Although previous research has proven that sound localization is possible on a mobile platform, this concept has not yet been shown for multiple mobile units that must communicate between each other. The intended application of this system is to model robot-aided search-and-rescue or underwater sound mapping.

Contents

- 1. Introduction** **1**
 - 1.1. Project Background 1
 - 1.2. Application 3
 - 1.3. Trilateration Concept 3

- 2. Methodology** **5**
 - 2.1. Mechanical Design 5
 - 2.2. Parts Used 6
 - 2.2.1. Motors 6
 - 2.2.2. Wheels and Casters 6
 - 2.2.3. Power Sources 7
 - 2.2.4. Microphone 7
 - 2.2.5. CPUs 8
 - 2.2.6. Chassis Components 8
 - 2.3. Electronic Design 8
 - 2.4. Software Design 10
 - 2.4.1. Program Flow 11
 - 2.5. Costs Summary 12

- 3. Results** **14**
 - 3.1. Assessment Criteria 14

- 4. Discussion** **15**

- 5. Conclusions** **17**

- 6. Future Works** **18**

- A. Appendix A** **20**
 - A.1. ADC Datasheet 20
 - A.2. Digital Pot Datasheet 21
 - A.3. Microphone Datasheet 22
 - A.4. Thin Speaker Datasheet 23

List of Figures

- 1. Microphone Array and Robot [1] 1
- 2. Ultrasonic Sensor Configuration [2] 2
- 3. Sound Direction Localization Using Binaural Sensing [3] 2
- 4. Trilateration Diagram 3
- 5. One Field Robot 5
- 6. MCP3008 Connected to Pi Cobbler [4] 9
- 7. Electronics Configuration between Master and Swarm Unit 10
- 8. Beacon Amplifier Circuit (Circuit Basics, Fritzing) 10
- 9. Program Flow Graphic 11
- 10. Final Chassis Design 15

List of Tables

1. Cost Evaluation 13

1. Introduction

This robotic system was designed to locate a sound source via trilateration between multiple mobile robots on the same field. As they navigated the test environment, these robots aimed to demonstrate time difference of arrival (TDOA)- based localization and mapping over a wireless network. Each robot was equipped with a single microphone and processing units, as well as wheels and encoders for navigation. Upon hearing the sound source, each robot was designed to report wirelessly to the master, which was tasked with making calculations to determine where the source was located. With this information, the master was designed to dispatch the robots to find the source, allowing the robots to navigate to their assigned locations.

1.1. Project Background

Several projects and experiments regarding acoustic robotic localization have been conducted with varying sensor and field configurations. The first of these to be discussed is a mobile robot capable of self-motion triangulation, completed by Sasaki, et al (2006). In this project, a mobile robot equipped with a 32-microphone array used sound direction localization to perform two-dimensional multiple sound source mapping. The robot is shown below in Figure 1.

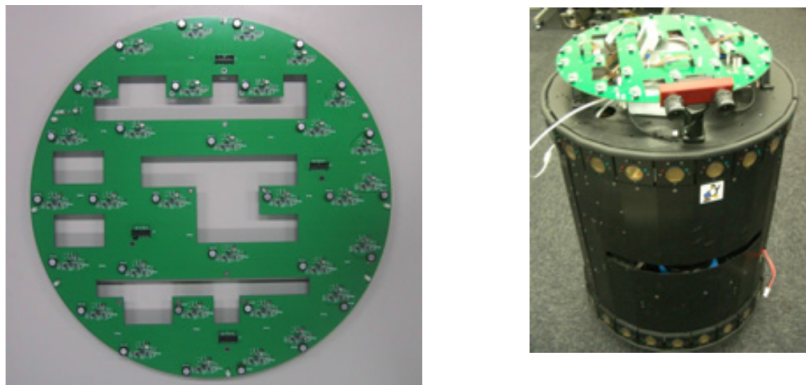


Figure 1: Microphone Array and Robot [1]

Researchers used Delay and Sum Beam Forming with Frequency Band Selection to discern multiple mobile sound sources, and simultaneously sampled 32 channels of analog-to-digital data. Once the robot has sampled, it uses an onboard 1GHz Pentium III dual CPU to run the RANdom SAmple Consensus (RANSAC) algorithm to develop an accurate model of the environment.

Also in 2006, Lewinger, et al from Case Western Reserve University developed an untethered mobile robot that used binaural ultrasonic sensing to navigate its environment and avoid obstacles. A small legged robot was built with two ultrasonic sensors located above an emitter, shown in Figure 2.

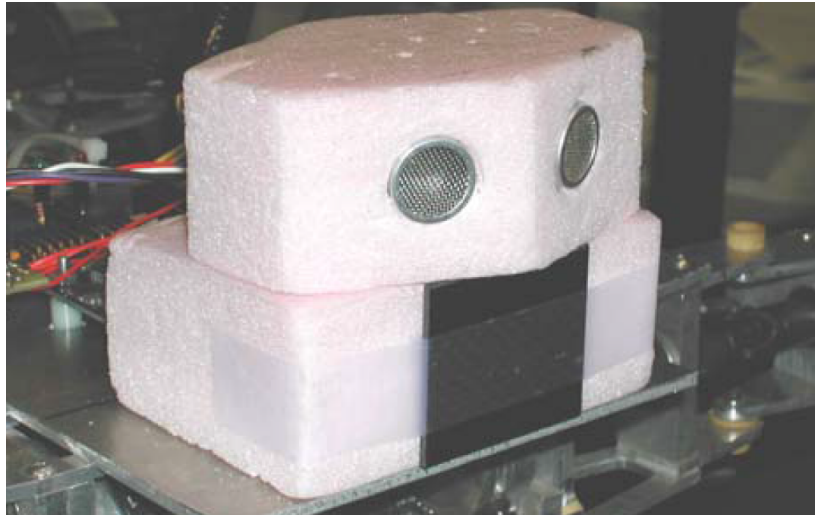


Figure 2: Ultrasonic Sensor Configuration [2]

By analyzing the differences of sound intensity received at each sensor, the team was able to execute obstacle detection and avoidance behaviors in an experimental field. This project used an Acroname Brainstem microcontroller to process signals from the ultrasonic sensors as well as R/C commands sent from operators on the field.

A third group of researchers Murray, et al, explored the concept of interaural time difference analysis on a mobile robot equipped with two microphones, similar to the configuration of mammalian ears (2004). This robot was designed as a waiter listening for customers in a busy restaurant, so the experimental field included multiple, distinct sound sources. This concept is displayed in Figure 3.

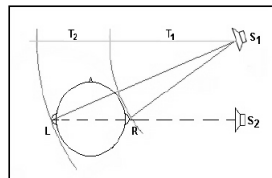


Figure 3: Sound Direction Localization Using Binaural Sensing [3]

To properly detect and localize sound, these researchers used principles present in the mammalian auditory cortex, such as interaural time difference and interaural phase difference. The robot used several high-speed processing units to gather and compute acoustic data with respect to azimuth.

While all of this research proves that sound localization is possible on a mobile platform, this concept has not yet been shown for multiple individually mobile platforms that must communicate between each other. The intended application of this system is to model robot-aided search-and-rescue. In this circumstance, the target that must be sought and rescued

would have a sound-emitting device which would be activated as necessary. Adjustments to the propagation characteristics and communication would be made to allow for environmental shifts.

1.2. Application

The swarm search and rescue can be implemented in a setting where a hiker is lost and requires assistance. The units would patrol well-known hiking areas, and each hiker would carry a sound-emitting device. If the hiker was lost or injured, he or she would activate their beacon. Upon hearing the beacon, the swarm units would begin the cycle of listening and searching for the source.

The concept in this project can also be applied to the real world issue of aquatic animal tracking, where whales and dolphins make loud, distinguishable sounds. Each swarm unit would be dispatched in the water, and would either anchor to the seafloor or dwell at a constant depth while sensing. They would surface to determine their locations via GPS, and this process would be used to find creatures underwater. Naturally, it would be necessary to include altitude detection in this context.

1.3. Trilateration Concept

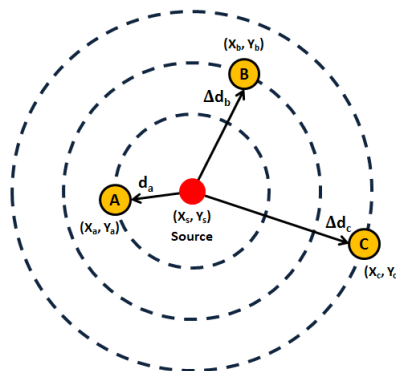


Figure 4: Trilateration Diagram

The basis for an acoustic trilateration method is found in time difference of arrival analysis. In Figure 4, each yellow circle represents a robot with a known location. The red circle represents the sound beacon at an unknown location. Given the absolute positions of each robot and the time delay of reception between each robot's microphone, it is possible to find the relative distances between each element.

The system can be modeled using Equations 1-3, where a_s is the absolute time that robot A received the sound. In this case, the time of flight is unknown, as are the coordinates of the sound source. In order to find the distances between the robots, the time delay is multiplied by the speed of sound v_s .

$$d_a = a_s v_s \quad (1)$$

$$\Delta d_b = b_s v_s \quad (2)$$

$$\Delta d_c = c_s v_s \quad (3)$$

Where b_s and c_s are measured relative to a_s . Using simple distance formulas, we find Equations 4-6, a nonlinear system of equations which can then be condensed into a solvable, linear system.

$$(x_s - a_x)^2 + (y_s - a_y)^2 = d_a^2 \quad (4)$$

$$(x_s - b_x)^2 + (y_s - b_y)^2 = (d_a + \Delta d_b)^2 \quad (5)$$

$$(x_s - c_x)^2 + (y_s - c_y)^2 = (d_a + \Delta d_c)^2 \quad (6)$$

2. Methodology

2.1. Mechanical Design

The robot is designed with three main levels and a smaller fourth level, as seen in Figure 5 below.

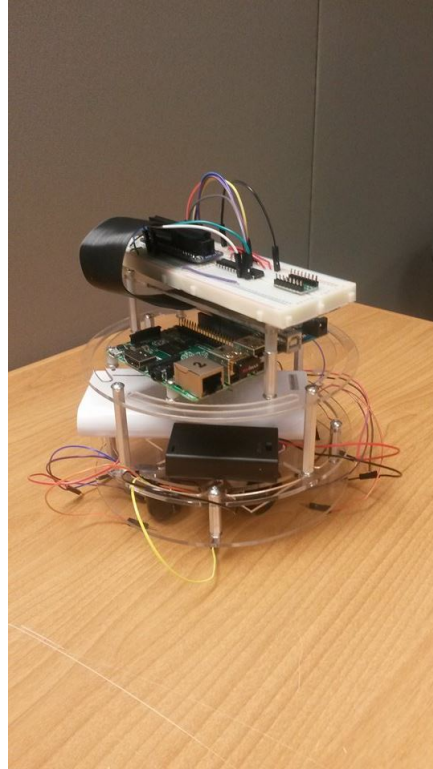


Figure 5: One Field Robot

Starting at the bottom, the first layer contains the components necessary for movement. The motors, wheels, and casters are all attached to this level. The second level holds the power banks, putting them in the middle of the robot. This keeps the center of gravity relatively low, as well as putting the power sources in a central location that is easy to reach for all components.

The third level holds both microcontrollers, situated right next to the small fourth level, which holds the circuitry and the sound detector. Having the sound detector at the very top of the robot minimized the chances of noise interference from the robot's own components, including the motors at the very base of the robot.

Prior to this design, a prototype with a slightly different design was built for testing purposes. The prototype had only two levels, and was half again the diameter of the final design. The prototype also used two omniwheels in place of the casters in the final design. Although

the design was functional, it was too wide for the purposes of this project while still not having enough space for two microcontrollers and all three power banks. In addition, the noise from the motors interfered with the sound detector.

2.2. Parts Used

2.2.1. Motors

The motors were chosen based on three factors: their torque, including their ability to handle the weight of the robot; their compatibility with quadrature encoders, including the availability of the same; and their cost. Their size and difficulty of mounting them to the chassis were also considered. To meet all of these requirements, the Pololu micro metal gearmotors worked exceptionally. A gear ratio of 210:1 (part 3077) was selected as having enough torque to move the robot without issue while spinning fast enough to move the robot at a reasonable pace.

These motors are set up to work with Pololu's magnetic encoders for micro metal gearmotors (part 3081), as well as their micro metal gearmotor brackets (part 1089), which made assembly very simple. Finally, to control the motors, Pololu's DRV8833 dual motor driver carrier (part 2130) was selected for its low price and compatibility with the other Pololu products.

2.2.2. Wheels and Casters

The design choices for robot mobility went through several iterations. The first choice was to use four Mecanum wheels to drive the robot. This would have allowed the robot to stay in the same orientation, yet be able to move easily in any direction. This design was ultimately rejected based on the high price of Mecanum wheels and the four motors that would be needed to drive them.

The second option considered was to use three driving omniwheels, arranged in a triangle. This design still allows the robot to move in any direction while maintaining its orientation. Three motors would be required for this design, a design that also held the potential for a fair amount of slippage in the wheels. As odometry was being used to track the robots' positions, it was decided that this design was neither as cost-effective as it could be nor precise enough for this project's needs.

The third choice for mobility was a set of two powered drive wheels combined with two omniwheels. Vex omniwheels and 70mm Pololu wheels were chosen and used for the prototype design. This design used essentially the same drive programming as the final design.

In the end, however, it was decided that casters would be a better choice than omniwheels for three reasons. First, they were much cheaper per unit; second, a compatible size could be found very easily; and finally, they would allow the robot to have a lower profile, putting the base level very close to the ground. Pololu metal 3/4" ball casters (part 955) were selected

for their height and load-bearing capacity. To complement them, a set of two 32mm Pololu wheels (part 1087), designed to be used with the micro gearmotors mentioned above, were used as drive wheels.

2.2.3. Power Sources

In the original design, the robot would have been run off of a single power source, around 9volts with a preferably high amperage and capacity. The first choice was a LiPo battery, for their low cost and small size; this idea was discarded based on the difficulty of charging LiPo batteries, as well as the availability of large capacities. However, as the designs became more final, it was clear that a single power source would not meet the needs of each separate section. The motors would require their own power source due to the heavy current draw, as would the sound-sensing chip, which needed a more specific voltage. The microcontrollers would need yet another power source to ensure that they both received a steady flow of power at the correct voltage. A steady flow of power was most important for the microcontrollers, so the most steady power source would need to be used for them.

With this in mind, a cell phone battery charger pack was selected to power the microcontrollers. This choice was made for several reasons, including the relative steadiness of a power source meant to be used as a charger. It was also compact, high capacity, and low-cost. A two-port 10AH phone charger pack was selected so that both microcontrollers could be connected at once without bringing in yet another power source.

For the motors, two 3.7V 18650 batteries were placed in series, giving the motors the seven volts required for movement. As 18650 batteries are commonly used in flashlights, headlamps, and similar light sources, they were readily available, and were inexpensive when compared to dedicated motor batteries. For the sound sensing chip, three AAA batteries were placed in series. This satisfied the power requirements in a very low-cost and easily replaceable way.

2.2.4. Microphone

The first choice of design was to use a regular omnidirectional microphone and filter the noises both digitally and mechanically. This proved to add a level of complexity that the design could not accommodate. Instead, the SparkFun sound detector (part 12642) was selected. The sound detector was an audio sensing board that provides three different outputs: audio, a binary signal of the presence or non-presence of sound, and an analog signal representing its amplitude. Although a greater cost than a small omnidirectional microphone alone, this chip allowed for greater flexibility in filtering the sounds to find the signal the robot was searching for.

2.2.5. CPUs

The microcontroller used for this project had a very strict set of requirements to fulfill. First, it needed a processor speed of at least 100MHz to be able to detect the sound signal with enough precision to make an accurate TDOA calculation. Second, it needed to be able to control the motors; third, it needed to be able to read the encoders. It also needed to support a Wi-Fi connection and handle multi-threaded processing. Finally, it needed to be compatible with the sound detector.

With these six requirements in mind, the Beaglebone Black was selected. In theory, this board should have had the capacity to meet each requirement. However, the board was unreliable, poorly documented, and difficult to use. It is still unknown whether the individual unit the team had access to for the prototype was simply faulty or if all iterations of this board would have had the same issue. Unfortunately, a second unit was not available to test this, and a new microcontroller had to be selected in the end.

However, no other low-cost microcontroller was able to meet the specific requirements of the project alone; therefore, it was decided that two microcontrollers would be used in tandem. The Raspberry Pi 2 B, combined with a Pi Cobbler (an extension to the Pi that allows for extra functions), had enough processing speed and power to handle the sound detector as well as Wi-Fi, while the Arduino Uno could easily control the motors and read the encoders. The two microcontrollers could be connected and communicate via USB.

2.2.6. Chassis Components

Each level of each robot was laser-cut from .125" cast acrylic sheets. Wood of a similar thickness was also considered, as it would also be easy to laser-cut and is actually less expensive than the acrylic. However, acrylic was chosen for several reasons. The surface of the acrylic is smoother, making it easier to attach things like Velcro. Acrylic is tough enough to handle the amount of strain that these robots are under—they are fairly light-weight machines, and no single sheet of acrylic is any longer than 8 inches across, not long enough to buckle for lack of support. Finally, the acrylic is transparent, which makes it easier to trace wires and other components through the design while building and redesigning.

Vex brand 8x32 hex standoffs combined with 3/8" 8x32 screws separate the layers, leaving space in each layer to move the components if necessary. Velcro strips are used to attach the power banks and breadboards to the chassis, making them easy to detach or move around. The chassis altogether is relatively small and lightweight, considering the number of components and potential battery life.

2.3. Electronic Design

For this application, the team originally chose the BeagleBone Black microcontroller and upon finding multiple functional defects, switched to a Raspberry Pi/Arduino combination.

With the latter configuration, each swarm unit used a Raspberry Pi to sense sound and communicate wirelessly with the master computer. Each unit also contained an Arduino, which received motor commands from the Raspberry Pi.

The Raspberry Pi offers WiFi capabilities, forty input/output pins, an HDMI port, four USB ports, an Ethernet port, and a clock speed of 900 MHz. While it does not offer an onboard analog to digital converter (ADC), there exists a Pi Cobbler breakout board, which provides access to all of the pins on the Pi board, and easily connects to the MCP3008. The MCP3008 is an 8-channel ADC with 10-bit resolution and a maximum sampling capacity of 200 kilosamples per second. Figure 6 shows the MCP3008 connected to the Cobbler with a potentiometer for testing purposes.

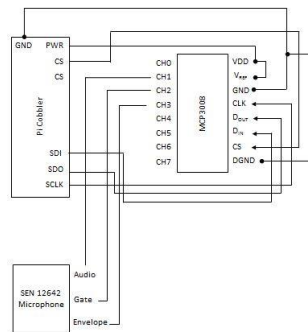


Figure 6: MCP3008 Connected to Pi Cobbler [4]

To sense the beacon, the team chose the SEN 12642 sound detector from Sparkfun. This particular module was chosen for its three outputs: a binary representation of the presence of sound, an amplitude representation of the wave, and an analog representation of the full wave form. This varied functionality gave each robot more flexibility when processing the signal and this affected the method of software analysis as well. To control sensitivity at varying distances and to properly account for the variation in ambient noise level, the team utilized an MCP4131 digital potentiometer. This allowed the capability to control the gain of the microphone through software via I2C. A top-level view of this configuration can be seen in Figure 7.

The team used a WiFi adapter with the Raspberry Pi to communicate with the master computer. To control motion, an Arduino Uno was connected to the Pi via USB. This board is equipped with six PWM-enabled I/O pins, multiple ADC pins, and a USB port. Throughout the trilateration process, the master would send motor commands wirelessly to the Pi, which would pass on the information to the Arduino over USB. The Arduino sent PWM signals to a DRV8833 dual motor driver, which controlled two DC motors equipped with two encoders. Each encoder fed into an I/O interrupt pin of the Arduino.

The beacon for this project was a thin speaker attached to an amplifier. An Arduino Mini was designed to provide a square wave at 5 KHz. Figure 8 shows the amplifier set up.

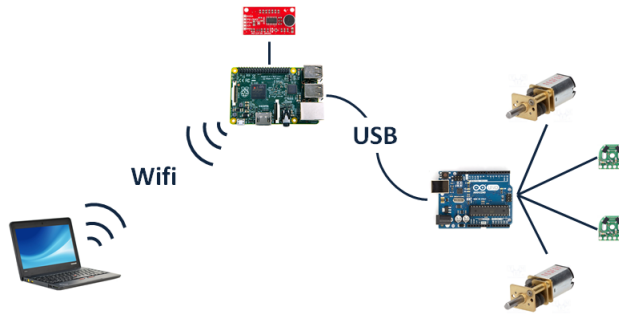
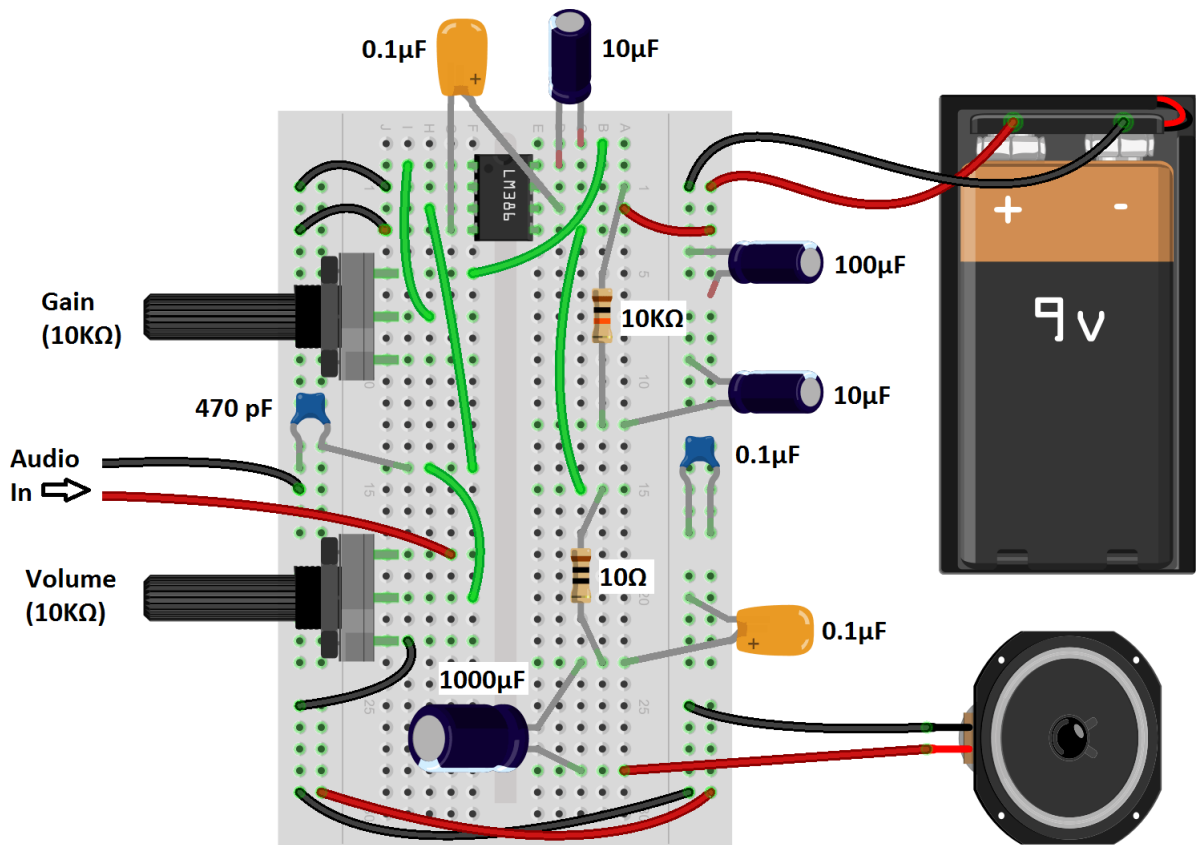


Figure 7: Electronics Configuration between Master and Swarm Unit



fritzing

Figure 8: Beacon Amplifier Circuit (Circuit Basics, Fritzing)

2.4. Software Design

The general layout of the software underlying the project was designed to be modular and straightforward. There was a section of the program to be run on the master, a section to be run on each swarm robot, and a section to be run on the beacon. The master was in charge of

making the trilateration calculations and telling the swarm robots where to go. The swarm robots were designed to listen for the beacon, report their time information to the master, and drive to the beacon, following instructions from the master. The communication and listening functionality were handled on the swarm robots by the Raspberry Pi, and the movement was handled by the Arduino. The beacon was simply in charge of making sound pulses that the swarm robots could hear.

Most of the code was written in Python. The master code was written in Python, and all of the code run by the Raspberry Pi was written in Python. The movement code handled by the Arduino, however, was written in the Arduino Programming Language, which is a combination of C and C++. The beacon code was also written in the Arduino Programming language, as the beacon was also controlled by an Arduino.

2.4.1. Program Flow

A graphical representation of the program flow can be found in Figure 9.

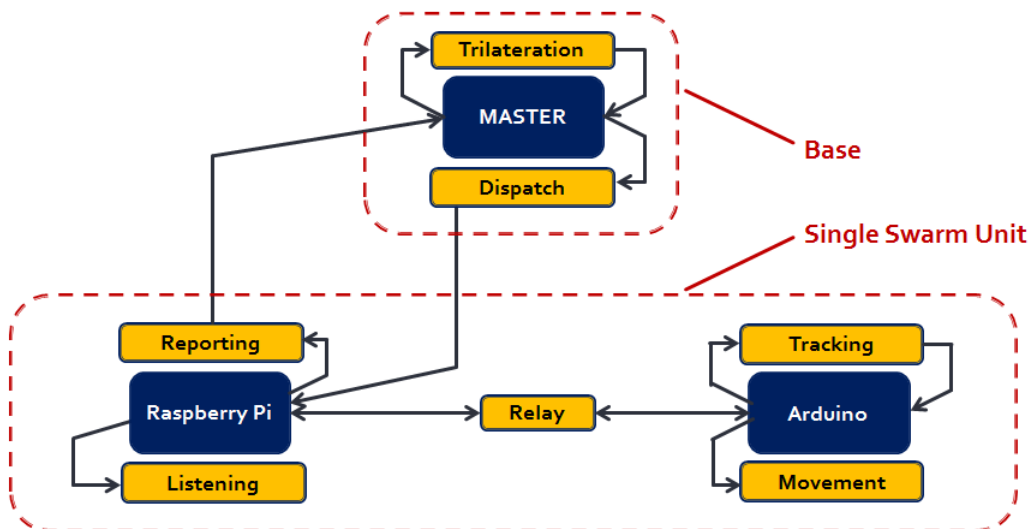


Figure 9: Program Flow Graphic

First, the beacon starts making noise. The first thing the swarm robots do is send their time information to the master. Using this information, the master performs a pseudo-synchronization where it keeps track of how far off each swarm robot is from real time.

After this, the swarm robots listen for the beacon, and send information on when they heard it to the master. Once the master has this information, it performs the trilateration calculations found in section 1.3.

Once the master has determined the position of the beacon with respect to the positions of the robots, it sends movement commands to each of the swarm robots.

The robots receive these commands and execute them, using the encoders on their motors to determine how far they have moved in a given direction. Once the master has finished giving movement commands to the swarm robots, it sends a message signifying the end of the program.

Threading was a key component in this project. The swarm robots each had two threads in their program: one thread to communicate with the master and receive movement commands, and one thread to execute the commands using the motors and encoders. The master had one thread for each of the robots, so that the movements of the robots could occur simultaneously instead of one-at-a-time.

2.5. Costs Summary

Table 1 below contains a detailed breakdown of costs for each robot.

According to Table 1, the total cost per robot is approximately \$249.43. Comparably sized robots cost between \$250 and \$350, and do not usually have sound-sensing systems or the versatility of two microcontrollers that this design has.

Part	Cost	Number/Robot	Total Cost/Robot
Barrel Jack to USB Connector	1.88	1	1.88
USB A to USB B Connector	5.99	1	5.99
Raspberry Pi	39.95	1	39.95
Raspberry Pi Cobbler	6.95	1	6.95
Arduino Uno	24.95	1	24.95
ADC	3.75	1	3.75
Micro SD Card (for Raspberry Pi)	11.99	1	11.99
AAA Battery Case	1.95	1	1.95
AAA Batteries	2.50	1	2.50
Micro Gearmotor	18.95	2	37.90
Magnetic Motor Quadrature Encoder	8.95	1	8.95
Dual Motor Driver	4.95	1	4.95
Motor Brackets	4.99	1	4.99
1/8x12x24" Acrylic Sheet	12.99	1	12.99
Cell Phone Battery/Charger	9.99	1	9.99
WiFi Adaptor	9.99	1	9.99
18650 Batteries	9.50	1	9.50
18650 Battery Case	1.50	1	1.50
Sound Detector	10.95	1	10.95
Solderless Bread Boards	7.69	1	7.69
Jumper Wires	5.99	1	5.99
Vex Standoffs	12.50	1	12.50
Digital Potentiometer	0.67	1	0.67
Wheels	4.98	1	4.98
Casters	2.99	2	5.98

Table 1: Cost Evaluation

3. Results

3.1. Assessment Criteria

There are several criteria the project was judged by. The robots themselves were judged based on their cost, abilities, and connectivity. The robots needed to be low-cost, be able to identify sounds and record their times of arrival, and communicate with the master.

For the trilateration, another set of criteria was used. These include accuracy, execution time, rate of success, and feasibility. Accuracy was judged by how close the robots could calculate the location of the sound beacon compared to its actual location. The calculated error was approximately 7cm, assuming that the processors had error of approximately 200 nanoseconds, with a field size of approximately six meters square. Execution time was judged based on how long the robots took to find the beacon (or to find the spot they assumed the beacon was located), and rate of success was based on how many times the robots found the beacon (within acceptable error) out of the times the trials were run.

The final criteria, feasibility, includes the potential for future work. Some possible factors for usefulness include the length of the battery life, the accuracy of the program (as discussed above), and what changes might be made to adapt it to a new environment.

4. Discussion

During the course of the project, the team encountered many technical challenges, some of which hindered progress. The first of these was the choice of microcontroller. To properly sense and trilaterate, it was essential to have a microcontroller with an ADC built in or modular with a sufficient clock speed. Since the robots are also mobile, the team initially chose the BeagleBone Black. This board has features of both an Arduino and RaspberryPi. However, documentation on its use is heavily lacking, and frequently, the team was unable to communicate with its GPIO pins, connect to the internet, or even upload any code to the board. It was after much deliberation and effort that the team switched to the Raspberry Pi/Arduino combination. This choice was much more functional, as both boards are well known and come with technical support. The interface between the user and the board is also considerably more intuitive and well adapted to various applications.

In addition to this, another hardware challenge was the organization of all necessary parts on board a single swarm unit. Each robot was no more than ten inches in diameter, with three tiers to support motors, three battery packs, two microcontrollers, and a breadboard containing the microphone, motor driver, ADC, and digital potentiometer. Several configurations had to be tested in order to minimize size while accommodating all pieces of hardware. The team found the most successful design placed the motors at the base, three battery packs on the next tier, two boards on the third tier, and the microphone-containing breadboard at the top to prevent sound from the rest of the robot from creating additional noise. This is shown in Figure 10.

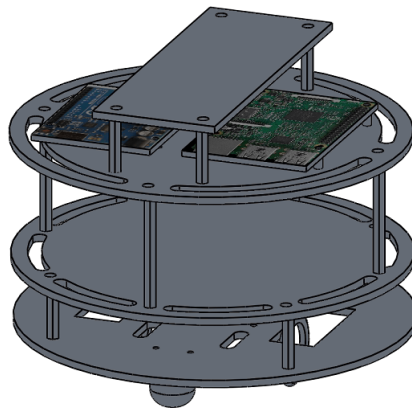


Figure 10: Final Chassis Design

The software for the project had multiple functioning parts, but ultimately, the team was unable to integrate all of the pieces into one system. Functionality that was accomplished was: WiFi communication between the Raspberry Pi and master, USB communication between the Pi and the Arduino, motor driving from the Arduino/motor driver to the motors, microphone sensing using Cobbler and ADC, software control of the digital potentiometer, and encoder reading from the encoders to the Arduino. The team encountered difficulty in the

process of multithreading with Python and debugging through the layer of abstraction that Python presents as a language. It is also uncertain how robust the microphone sensing is with respect to the environment, as the maximum sensitivity with the software-controlled potentiometer was half of the original sensitivity on the manufactured microphone. Due to this, the sound from the beacon had to be much louder than our design could support.

The final aspect of our design that required improvement was the beacon. It was designed to amplify a signal from an Arduino Uno or Mini board. However, the team found several issues with the wave generation feature of the Arduino. It was only possible to output a square wave from the Arduino, which directly conflicted with the specifications on most speakers, which are easily capable of producing a clean 15kHz when given sine waves. In addition, due to the slow standard clock speed available on standard PWM pins, the Arduino was unable to match any high frequency as the PWM operational frequency is only roughly 500 Hz. When the team attempted to use the Tone library, which is capable of 65.535 kHz, ringing and rattling could be heard from the speaker despite the documentation clearly stating that it should not pose a problem. The team also implemented a single transistor between the Arduino and speaker, and while this provided the same intensity of sound, it was not possible to increase the amplitude for the distance that was required.

5. Conclusions

From this project, the team gained a great understanding of the complexity of signal processing with sound in the audible range. Many factors must be considered, including scale, accuracy of hardware, and robustness of detection. It was found that it is theoretically possible to use swarm units to trilaterate a sound in an environment. However, this work requires high processing power and statistical analysis, as well as adjustments in mechanical design for real world environments. It is also necessary to repeatedly test the reliability of this system and carefully calculate the error, as this changes with the nature of the environment and the motion capabilities of the robots themselves. Overall, this project has strong potential to be implemented underwater, and the dynamics of sound would be worthwhile to study.

6. Future Works

The initial design of this project included multiple phases. Through the course of the project, work was only done in phase 1. Upon completion of phase 1, the robots would be able to hear the beacon, send their information to the master, receive commands to move to the beacon, and monitor their motion using only their own encoders. For the sake of future work, the following is a high-level layout of the remaining phases.

Phase 2 would see the addition of a second master computer, as well as each swarm robot receiving its own beacon with its own unique sound signature. This way, the robots would still monitor their motion using their own encoders, but after they believe they have reached the beacon, they would then sound off their own beacons. The two master computers would then perform a modified trilateration process to locate each of the robots, the robots would listen for the goal beacon again, and the whole process would repeat until the robots reached the goal beacon. Phase 2 accounts for any slipping that may occur, which the motor encoders wouldn't report.

After phase 2 was complete, phase 3 would institute a system where the robots were continuously sounding off their beacons, and their motion would be exclusively monitored by the two masters' modified trilateration technique. This way, the program would only have to be run through a single time.

In phase 4, obstacles in the field would be introduced. The field would be known to the master computers, and they would be in charge of path planning for the swarm robots.

References

- [1] M. H. Sasaki Y, Kagami S, “Multiple sound source mapping for a mobile robot by self-motion triangulation,” 2009, in: IEEE international conference on intelligent robots and systems, pp 380-385.
- [2] R. D. Q. William A Lewinger, Michael S Watson, “Obstacle avoidance behavior for a biologically-inspired mobile robot using binaural ultrasonic sensors,” 2006, iEEE/RSJ International Conference on Intelligent Robots and Systems, p. 5769-5774, Beijing, China.
- [3] W. S. Murray JC, Erwin HR, “Robotic sound-source localization and tracking using interaural time difference and cross-correlation,” 2004, aI Workshop on Neurobotics.
- [4] “Build a great sounding audio amplifier (with bass boost) from the lm386,” 2016, circuit Basics.
- [5] C.-K. W. J.-S. Hu, C.-Y. Chan and C.-C. Wang, “Simultaneous localization of mobile robot and multiple sound sources using microphone array,” 2009, in Proc. IEEE ICRA, Kobe, Japan, pp. 29-34.
- [6] F. Keyrouz, “Efficient binaural sound localization for humanoid robots and telepresence applications,” 2008, (Ph.D. thesis), Technische Universität München.
- [7] F. C. Emmanuel Vincent, Aghilas Sini, “Audio source localization by control of a mobile robot,” 2015, iEEE 2015 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia.

A. Appendix A

A.1. ADC Datasheet



MCP3004/3008

2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface

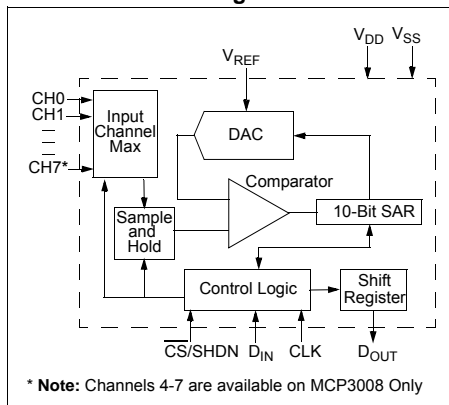
Features

- 10-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL
- 4 (MCP3004) or 8 (MCP3008) input channels
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200 ksp/s max. sampling rate at $V_{DD} = 5V$
- 75 ksp/s max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
- 5 nA typical standby current, 2 μA max.
- 500 μA max. active current at 5V
- Industrial temp range: $-40^{\circ}C$ to $+85^{\circ}C$
- Available in PDIP, SOIC and TSSOP packages

Applications

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

Functional Block Diagram

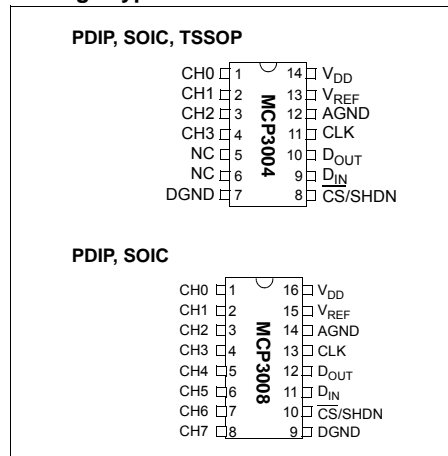


* Note: Channels 4-7 are available on MCP3008 Only

Description

The Microchip Technology Inc. MCP3004/3008 devices are successive approximation 10-bit Analog-to-Digital (A/D) converters with on-board sample and hold circuitry. The MCP3004 is programmable to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3008 is programmable to provide four pseudo-differential input pairs or eight single-ended inputs. Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) are specified at ± 1 LSB. Communication with the devices is accomplished using a simple serial interface compatible with the SPI protocol. The devices are capable of conversion rates of up to 200 ksp/s. The MCP3004/3008 devices operate over a broad voltage range (2.7V - 5.5V). Low-current design permits operation with typical standby currents of only 5 nA and typical active currents of 320 μA . The MCP3004 is offered in 14-pin PDIP, 150 mil SOIC and TSSOP packages, while the MCP3008 is offered in 16-pin PDIP and SOIC packages.

Package Types



A.2. Digital Pot Datasheet



MICROCHIP MCP413X/415X/423X/425X

7/8-Bit Single/Dual SPI Digital POT with Volatile Memory

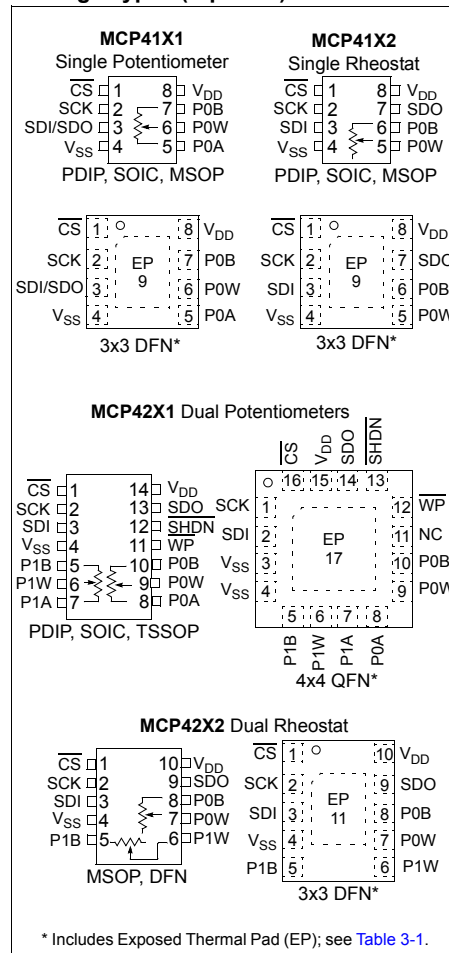
Features

- Single or Dual Resistor Network options
- Potentiometer or Rheostat configuration options
- Resistor Network Resolution
 - 7-bit: 128 Resistors (129 Steps)
 - 8-bit: 256 Resistors (257 Steps)
- R_{AB} Resistances options of:
 - 5 k Ω
 - 10 k Ω
 - 50 k Ω
 - 100 k Ω
- Zero Scale to Full-Scale Wiper operation
- Low Wiper Resistance: 75 Ω (typical)
- Low Tempco:
 - Absolute (Rheostat): 50 ppm typical (0°C to 70°C)
 - Ratiometric (Potentiometer): 15 ppm typical
- SPI Serial Interface (10 MHz, modes 0,0 & 1,1)
 - High-Speed Read/Writes to wiper registers
 - SDI/SDO multiplexing (MCP41X1 only)
- Resistor Network Terminal Disconnect Feature via:
 - Shutdown pin ($\overline{\text{SHDN}}$)
 - Terminal Control (TCON) Register
- Brown-out reset protection (1.5V typical)
- Serial Interface Inactive current (2.5 μ A typical)
- High-Voltage Tolerant Digital Inputs: Up to 12.5V
- Supports Split Rail Applications
- Internal weak pull-up on all digital inputs
- Wide Operating Voltage:
 - 2.7V to 5.5V - Device Characteristics Specified
 - 1.8V to 5.5V - Device Operation
- Wide Bandwidth (-3 dB) Operation:
 - 2 MHz (typical) for 5.0 k Ω device
- Extended temperature range (-40°C to +125°C)

Description

The MCP41XX and MCP42XX devices offer a wide range of product offerings using an SPI interface. This family of devices support 7-bit and 8-bit resistor networks, and Potentiometer and Rheostat pinouts.

Package Types (top view)



A.3. Microphone Datasheet



www.fairchildsemi.com

LMV321, LMV358, LMV324

General Purpose, Low Voltage, Rail-to-Rail Output Amplifiers

Features at +2.7V

- 80 μ A supply current per channel
- 1.2MHz gain bandwidth product
- Output voltage range: 0.01V to 2.69V
- Input voltage range: -0.25V to +1.5V
- 1.5V/ μ s slew rate
- LMV321 directly replaces other industry standard LMV321 amplifiers; available in SC70-5 and SOT23-5 packages
- LMV358 directly replaces other industry standard LMV358 amplifiers; available in MSOP-8 and SOIC-8 packages
- LMV324 directly replaces other industry standard LMV324 amplifiers; available in SOIC-14 package
- Fully specified at +2.7V and +5V supplies
- Operating temperature range: -40°C to +125°C

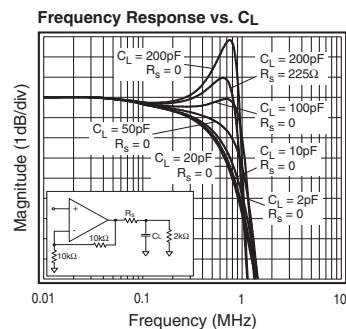
Applications

- Low cost general purpose applications
- Cellular phones
- Personal data assistants
- A/D buffer
- DSP interface
- Smart card readers
- Portable test instruments
- Keyless entry
- Infrared receivers for remote controls
- Telephone systems
- Audio applications
- Digital still cameras
- Hard disk drives
- MP3 players

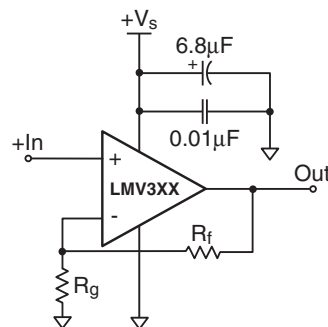
Description

The LMV321 (single), LMV358 (dual), and LMV324 (quad) are a low cost, voltage feedback amplifiers that consume only 80 μ A of supply current per amplifier. The LMV3XX family is designed to operate from 2.7V (± 1.35 V) to 5.5V (± 2.75 V) supplies. The common mode voltage range extends below the negative rail and the output provides rail-to-rail performance.

The LMV3XX family is designed on a CMOS process and provides 1.2MHz of bandwidth and 1.5V/ μ s of slew rate at a low supply voltage of 2.7V. The combination of low power, rail-to-rail performance, low voltage operation, and tiny package options make the LMV3XX family well suited for use in personal electronics equipment such as cellular handsets, pagers, PDAs, and other battery powered applications.



Typical Application



A.4. Thin Speaker Datasheet



Cover Industrial Co., Ltd.

Block 12, 3/F, DongFang JianFu, YuSheng Industrial zone
GuShu Village, XiXiang Town, BaoAn District , Shenzhen
Guangdong Province, China 518126

Attn: Alex Xiao, Tel: 86-755-29173471, Fax: 86-755-29174502

[Http://www.cover-cn.com](http://www.cover-cn.com)

Parameters:

1	Type	Dynamic speaker
2	Dimension	External diameter 40 mm
3	Rated Input Power	0.25 W
4	Impedance	8 ohm \pm 15% at 1500Hz
5	Resonance Frequency (Fo)	440 Hz \pm 20% at Fo, 1V
6	Sensitivity (S.P.L.)	85dB(W/m) \pm 3 dB at AVE 0.6K,0.8K,1.0K,1.2K(Hz). 96dB(0.25W/0.1m) \pm 3 dB
7	Frequency Range	Fo – 20KHz
8	Distortion	Less than 10 % at 1500Hz 0.25W
9	Max. Input Power	Must be normal at 0.4W white noise for 1 minute.
10	Voice Coil	Diameter 10.8 mm
11	Magnet	Rare earth permanent (Nd-Fe-B) magnet Φ 10 x 1.5mm
12	Weight	11g \pm 2g
13	Appearance	Should not exist any obstacle to be harmful to normal operation; damages, cracks, rusts and distortions, etc.
14	Operation Test	Must be normal at program source –0.25W
15	Buzz, Rattle, etc.	Should not be audible at 1.41V sine Wave between Fo to 20KHz
16	Polarity	When positive voltage is applied to the terminal marked (+), diaphragm should move to the front.
17	Terminal Strength	Capable of withstand 1kg load for 30 seconds without resulting in any damage or rejection.
18	Load Test	0.2 W white noise is applied for 96 hours and satisfy the test listed on item 05,06,13,15
19	High Temp. Test	Keep 96 hours at +70°C \pm 3°C and leave 3 hours in normal temperature and then check
20	Low Temp. Test	Keep 96 hours at -20°C \pm 3°C and leave 3 hours in normal temperature and then check
21	Humidity Test	Keep 96 hours at + 60°C \pm 3°C relative humidity 95% and leave 3 hours in normal temperature and then checked.