DELTA TECHNOLOGIES

Proof of Concept

Table of Contents

1.1 Executive Summary

Post-CDR1, Delta Technologies received authorization from The CARPA Authority to follow through with the design at the time. However, in consideration of feedback received from The CARPA Authority and various design critiques provided by its employees, Delta Technologies concluded that a complete redesign would ultimately provide the best product possible.

After various design iterations were proposed, Delta Technologies came to a consensus on a final revision. The new Aerial Reconnaissance Unit no longer detaches from the main robotic system. The detachment mechanism has been eliminated in order to do away with unnecessary flight module docking risks and connectivity issues between each system. Mercury's floatation capabilities have been eliminated. In addition, Mercury's flight configuration has been changed from that of a quad-copter to a tri-copter.

Using approximately half of the allowed budget, Delta Technologies went about constructing Mercury's individual robotic systems in order to execute tests necessary to demonstrate their functionality.

We conducted tests to examine the Aerial Reconnaissance Unit (ARU) motors' thrust output. We also conducted tests to examine the Hydrological Analysis System's (HAS) ability to determine the levels of contamination in water by parts per million. Lastly, we conducted tests to examine the Geological Analysis System's (GAS) mine sensing capabilities.

As of now, Delta Technologies has completed systems testing for Project Mercury and is on schedule for manufacturing the final Project Mercury.

1.2 Mission Overview

In September 2012, the CARPA Authority commissioned Delta Technologies to design, build, and test a fully autonomous, polymorphic robotic system in what the CARPA Authority dubbed "Project Morpheus."

The CARPA Authority required Project Morpheus to have the following capabilities:

- Autonomous reconfiguration through three morphs and four states
- Autonomous performance of a valued task in each state
- Capability to perform the series of morphs and tasks at least twice
- Transportable within a protective case, of which Morpheus will be removed from and returned to the case by an operator

Delta Technologies is creating a fully autonomous polymorphic robot in order to supplement military personnel with real time information. Project Mercury will perform the following tasks:

- Free aerial movement
- Three-dimensional mapping of a room
- Water testing
- Mine sensing
- Manipulation of C4 explosive

Mercury will not hinder a soldier's performance on the field, making it the paragon of non-combat support equipment.

1.3 Approach Summary

Delta Technologies' approach to Project Morpheus includes one polymorphic robot with four distinct states, and three morphs. In order to avoid putting form before function, each state has unique capabilities. Each morph leaves, the robot, dubbed Mercury, in one of four states: Dormant State, Flight State, Two-Wheel Drive State, and Three-Wheel Drive State.

In Dormant State, Mercury will be in a robust, compact configuration, allowing for ease of transportation. Mercury is composed of a central body and three shrouds. The central shroud will be retracted within the robot's body and the two outer shrouds will lie flush against Mercury's outer walls. Mercury is deployed by simply placing it on the ground and activating it.

Mercury's second state, Flight State, allows the Aerial Reconnaissance Unit (ARU) to perform its three-dimensional room mapping and surveillance tasks. Mercury will operate within a wireless network in order for it to relay back any information it gathers to the end user.

After flight, Mercury has the option of either morphing into a two-wheeled or three-wheeled drive state by landing on its base. All three shrouds will return to their dormant arrangement prior to further morphs. To enter Two-Wheel Drive State, the outer shrouds will lower simultaneously via an elevator system to which they are both attached. As they touch the ground, the robot will ascend yet remain upright, capable of navigation similar to that of a Segway due to its relatively low center of gravity. When a more stable drive state is optimal (example: in rough terrain), the central shroud will extend outward from within the body and act as a third wheel. This is the Three-Wheel Drive State.

In either drive states, an array of systems can be deployed: the Geological Analysis System (GAS), the Hydrological Analysis System (HAS) and the C4 Manipulator.

The GAS performs mine sensing by means of an array of three geophones in a triangular pattern around a speaker. The assembly is in turn attached to the end of an arm that extends outward from Mercury's core. The C4 manipulator, as well as the HAS unit's sensor will be similarly mounted on arms. The deployment of Mercury's motion detection sensor can all occur within either state as well.

2.1 Aerial Reconnaissance Unit (ARU) Tests

Mercury is designed to be able to fly in order to complete aerial reconnaissance, which is advantageous over ground reconnaissance because a complete, three-dimensional coverage of a room or area is more informative than two-dimensional feedback. The robot is designed to lift the projected six kilograms through the use of six motors.

Sub-Tests

- High-Load Test
- Flight Simulation Test
- Point Cloud Viewer Test
- Radar Test for Movement

Required Equipment

- Standard Weights
- Computer
- Electronic Balance
- Polycarbonate ARU Shroud System
  - Structure
  - Skyway X 4108S-11 600KV Brushless Motor
  - APC 11X4.7 Propeller
- Shroud Docking Station
  - Base Plate
  - Guiding Rails
- Electrical System
  - Turnigy Plush 18A Electronic Speed Controller
  - Turnigy 1000mAh 30-40C Discharge Li-Po Battery
  - Arduino Micro
- String
- Polycarbonate Testing Container

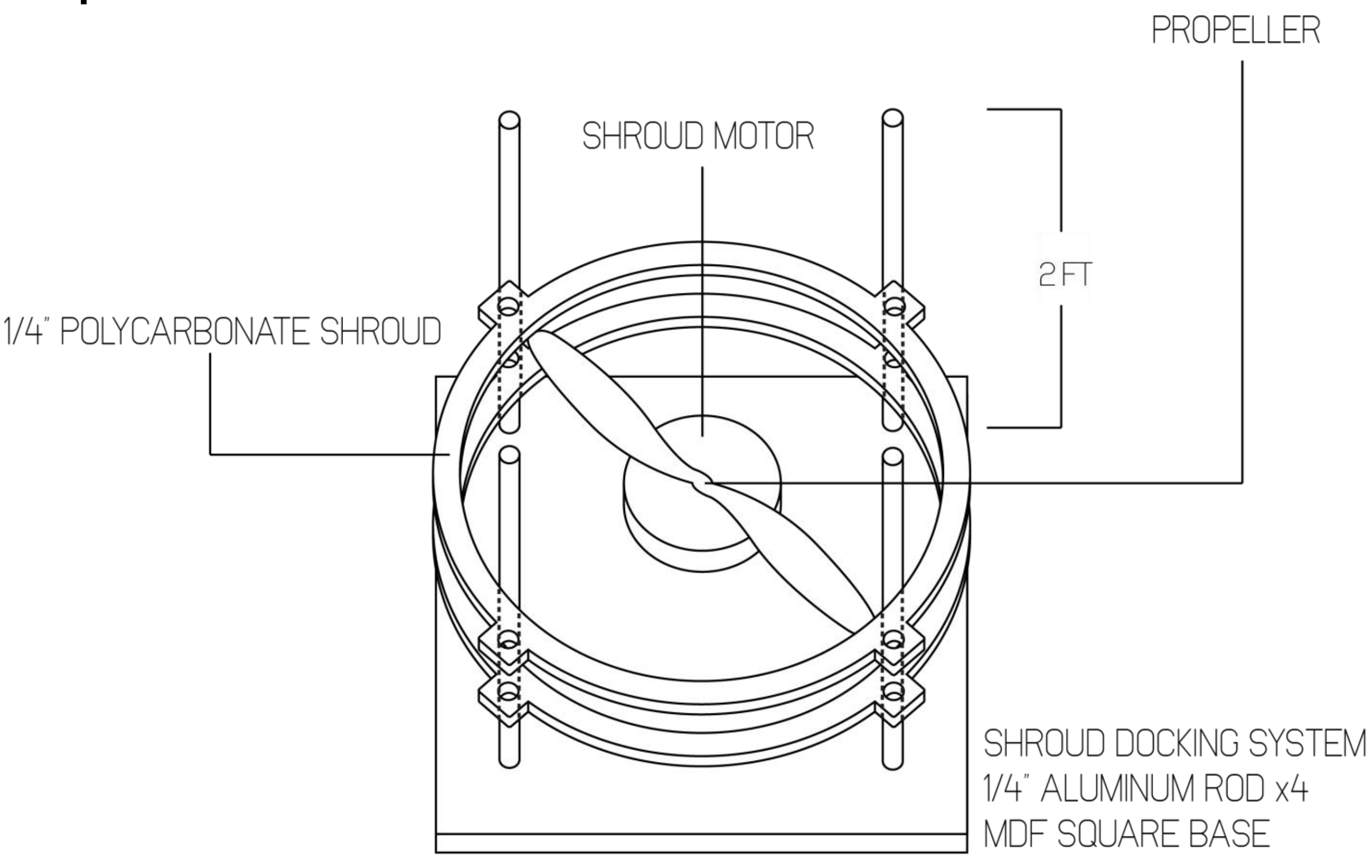
Special Notes

- The testing must be kept within the Polycarbonate Testing Container at all times to avoid injury.
- Test area should be kept clear in order to avoid contact with propellers spinning at high speeds.

ARU –High-Load Test

The ARU High-Load test demonstrates the maximum amount of weight that the ARU can lift.

Setup



Procedure

1. Clear a 1.5′ x 1.5′ space.
2. Place the pre-assembled Shroud Docking System in that space.
3. Position the Electronic Balance in the center of the Shroud Docking System and power it on.
4. Place the Standard Weight on the Electronic Balance.
5. Station the Polycarbonate ARU Shroud System on the Shroud Docking System.
6. Use the string to tie the Standard Weight to the mounted Polycarbonate ARU Shroud Structure. Ensure there is some slack on the string to guarantee accurate Electronic Scale measurements.
7. Allow the balance completes its weight analysis. Click the "ZERO" button to complete scale calibration.
8. Connect the Electrical System to the Polycarbonate ARU Shroud System as shown above.
9. Connect the Arduino Micro to the Computer through a Micro-USB to Mini-USB cable
10. Upload the code onto the Ardunio Micro (Refer to Appendix F for the code).
11. Place the Polycarbonate Testing Container over the Polycarbonate ARU Shroud System.
12. Turn on the Polycarbonate ARU Shroud Motor and allow it to run at full throttle.
13. Record the measurements shown on the Electronic Balance.

Results

Throttle (%)	Lift (grams)
80	600
100	1250

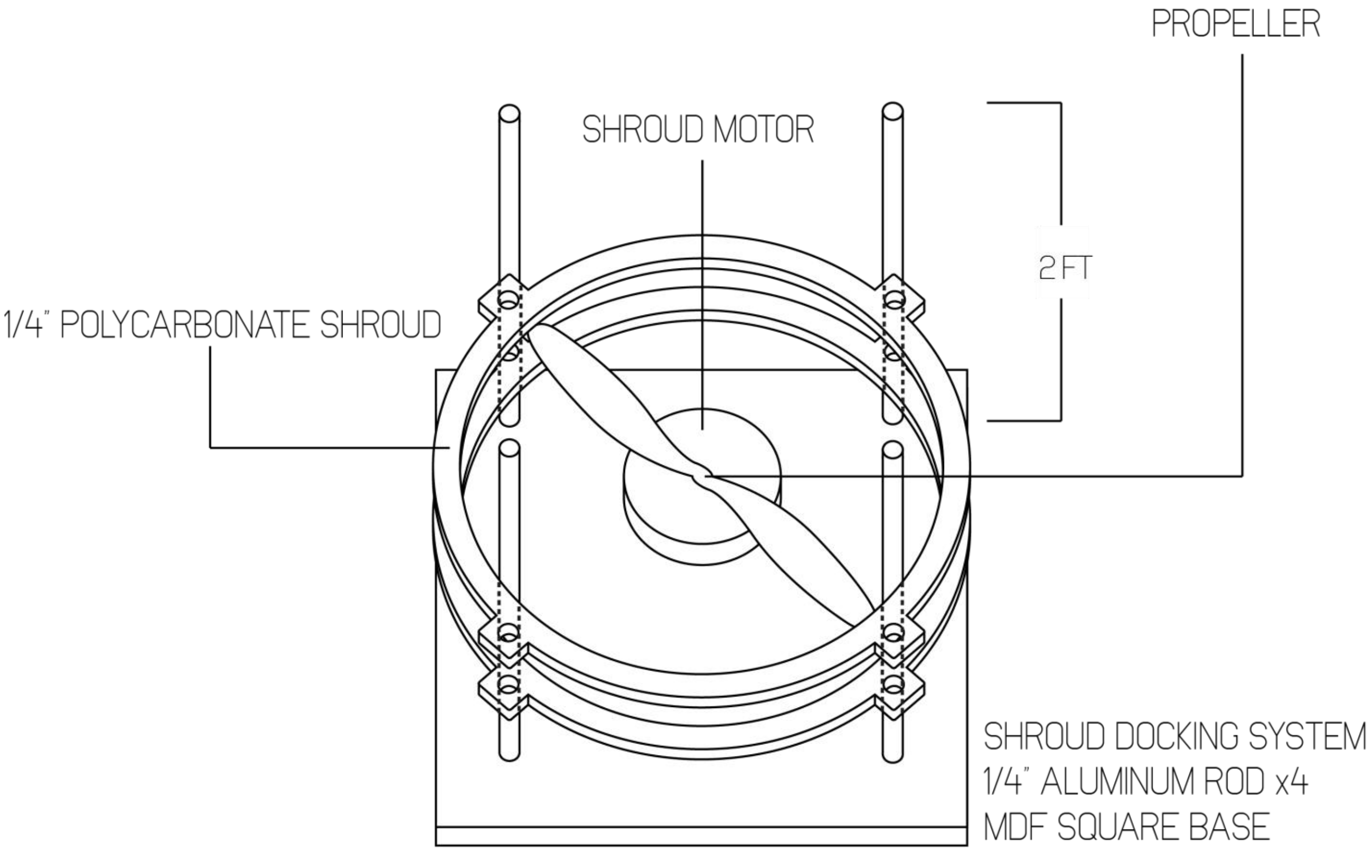
Conclusion

Through the completion of this test, the results show that a total of 1250 grams can be lifted with the selected motor and propeller. If the planned total of six motors are to be integrated onto the Mercury, this would be sufficient to lift the entire robot with a 25% over thrust for banking and turning. (Refer to Appendix D for the data specifications of the motor.)

ARU –Flight Simulation Test

The ARU Flight Simulation Test will determine the ARU's flight capabilities. This test will assess the ARU's flight performance while subjected to a load.

Setup



Procedure

1. Clear a 1.5′ x 1.5′ space.
2. Place the pre-assembled Shroud Docking System in that space.
3. Position the Electronic Balance in the center of the Shroud Docking System and power it on.
4. Place the Standard Weight on the Electronic Balance.
5. Station the Polycarbonate ARU Shroud System on the Shroud Docking System.

6. Use the string to tie the Standard Weight to the mounted Polycarbonate ARU Shroud Structure. Ensure there is some slack on the string to guarantee accurate Electronic Scale measurements.
7. Allow the balance completes its weight analysis. Click the "ZERO" button to complete scale calibration.
8. Connect the Electrical System to the Polycarbonate ARU Shroud System as shown above.
9. Connect the Arduino Micro to the Computer through a Micro-USB to Mini-USB cable
10. Upload the code onto the Ardunio Micro (Refer to Appendix F for the code).
11. Place the Polycarbonate Testing Container over the Polycarbonate ARU Shroud System.
12. Turn on the Polycarbonate ARU Shroud Motor and allow it to run at full throttle.
13. Record the measurements shown on the Electronic Balance.

Results and Conclusions

With the weight of approximately 300 grams, the robot can actually lift. Through the completion of this test, the results show that the mechanism can be lifted off the ground with the selected motor and propeller.

ARU – Point Cloud Viewer Test

Mercury has the feature to enter a room and produce a three dimensional map of the room's interior. In order to accomplish that, a capable camera is required. One such camera is the Asus XTION Pro Live, and that is the camera that will be used on Mercury. To aid in gathering three dimensional data of any given room, a software library referred to as the Point Cloud Library will be utilized. This library allows Mercury access to a point cloud from the camera. With that point cloud, along with other data from the camera, a map of the room can be created by surveying the entirety of the area. This procedure sets up and tests the Asus XTION for receiving point cloud data. This will provide evidence that generating and viewing a three dimensional map is possible.

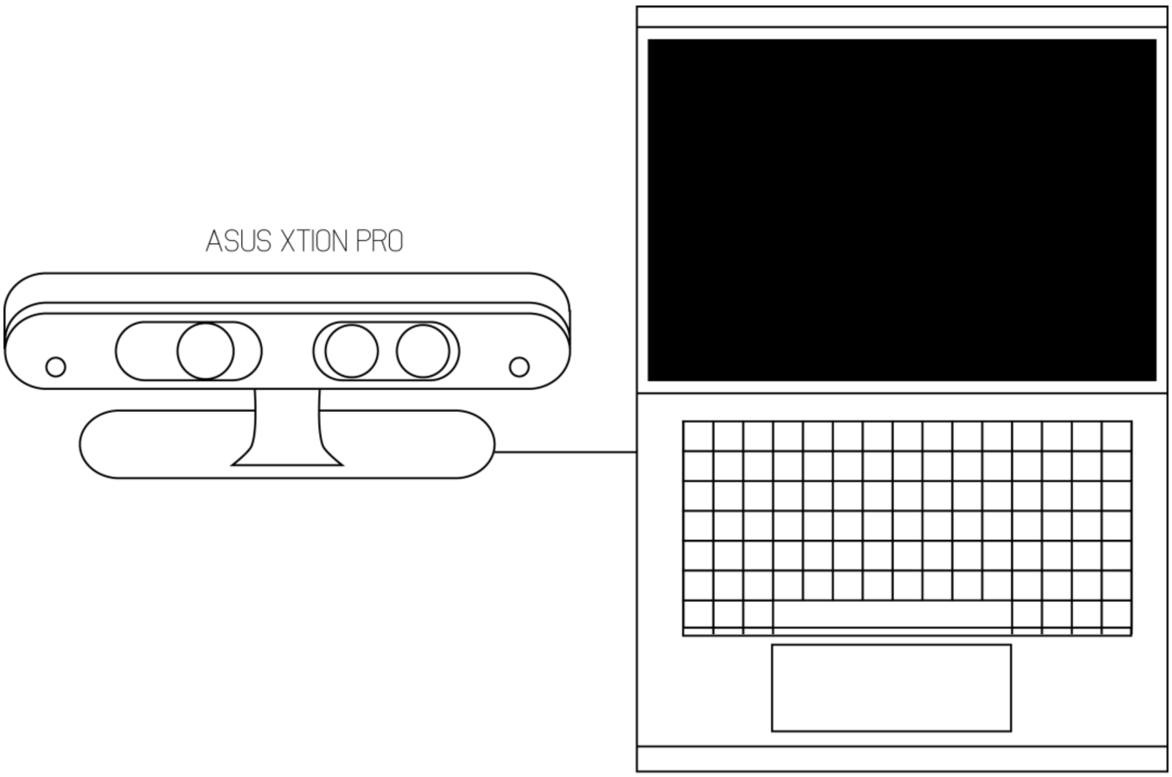
Required Equipment

- Computer (running Linux, in this test)
- Asus XTION Pro Live Camera
- Point Cloud Library

Special Notes

- Note the range of the camera as being approximately 2 feet to 12 feet.
- In this test the Linux terminal is utilized along with the commands “mkdir” and “cd” which make a directory and navigate to a directory, respectively.

Setup



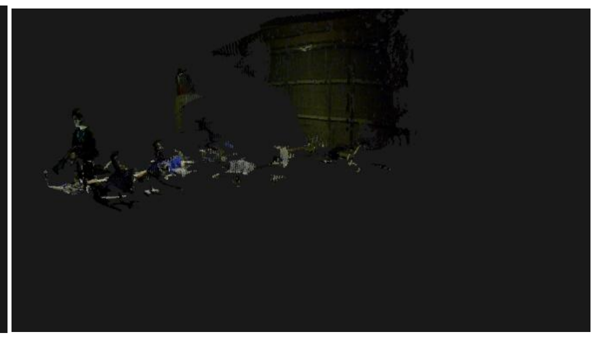
Procedure

1. Install the Point Cloud Library.
2. In our tests, we installed the Point Cloud Library via the terminal.
3. Plug the Asus XTION Pro Live into the computer.
4. Open terminal and cd to a directory you feel comfortable with.
5. Once in an appropriate folder, mkdir a test directory and cd into that directory.
6. Make a new file called “cloud_viewer.cpp”.
7. Type the code for the cloud viewer.

8. Make a new file called “CmakeLists.txt”.
9. Type the contents of the Cmake file.
10. Run “cmake .”
11. Compile the code.
12. Run the newly-made executable.

Results

These are three-dimensional images of rooms taken during team meetings.



ARU – Radar Test for Movement

Mercury is designed to detect movement in a room to alert soldiers of possible hidden enemies. We have designed a rig that uses an ultrasonic range finder to detect changes in position data acquired with the sensor. The goal of this procedure is to create a radar module that detects changes in position in a room.

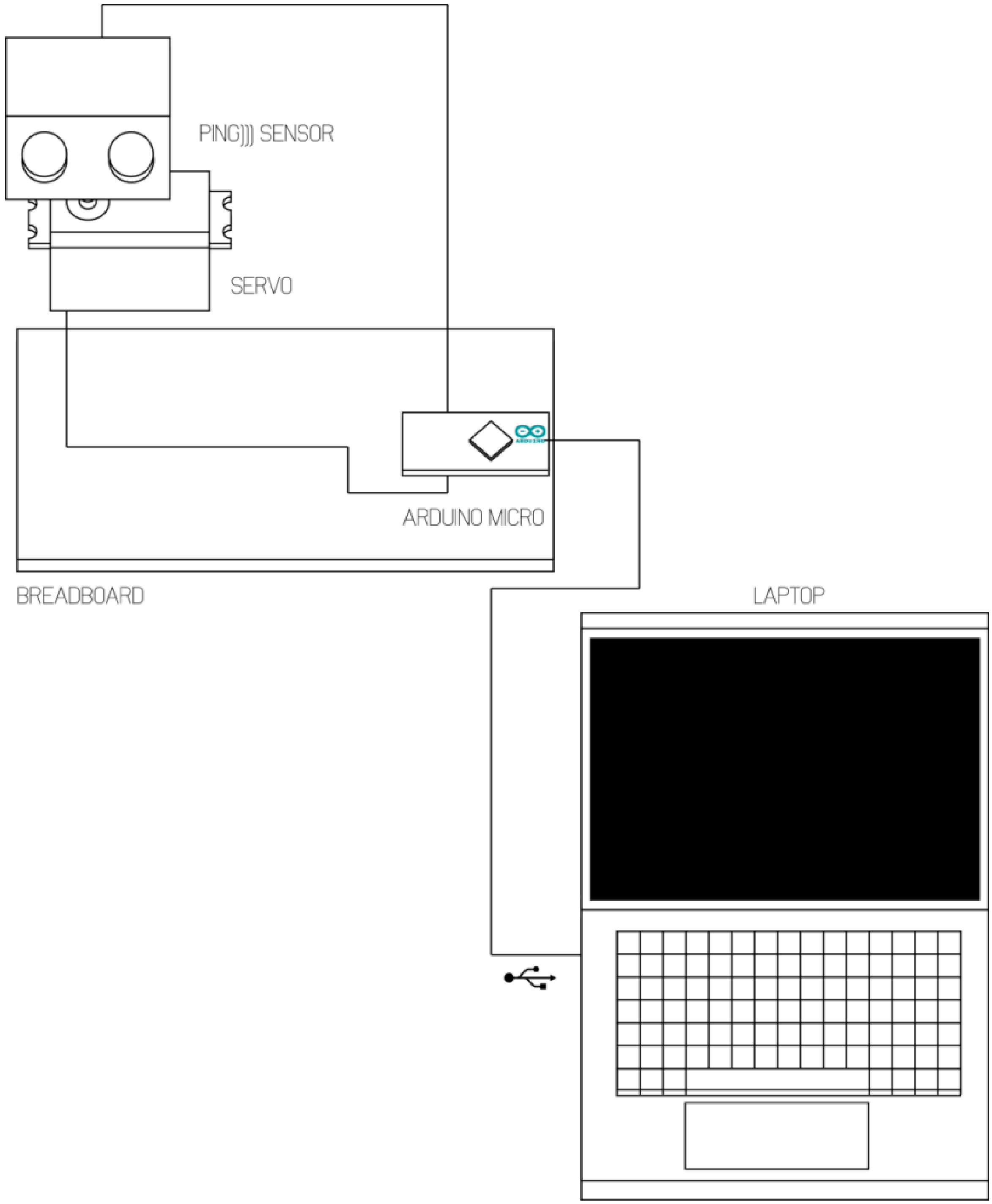
Sub-Tests

- Rotation Test
- Ping Sensor Test
- Data Storage Test / Motion Detecting Test

Required Equipment

- Bread Board
- Arduino Micro
- PWM Cables
- Servo Motor
- Zip Ties
- 3-D Printed Mount
- Ping Sensor

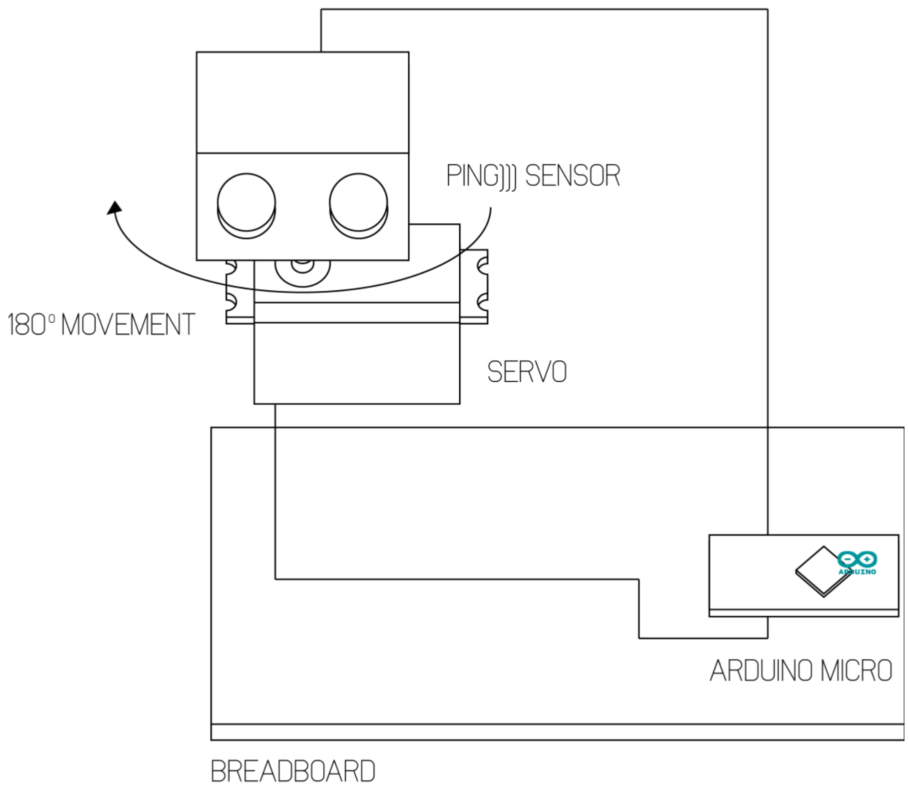
Setup:



ARU – Radar Test for Movement – Rotation Test

The rotation test determines if the servo motor rotates.

Setup



Procedure

1. Write algorithm for rotation of servo motor. (Refer toAppendix E.)
2. Wire servo motor to Bread Board.
3. Wire Arduino Micro to Bread Board.
4. Provide power to bread board through 9V battery.
5. Interface corresponding pins in to allow Arduino to control servo motor.
6. Confirm servo motor rotates 180 degrees.

Results

The continuous rotation servo is able to rotate 180 degrees while instructed by the Arduino Micro.

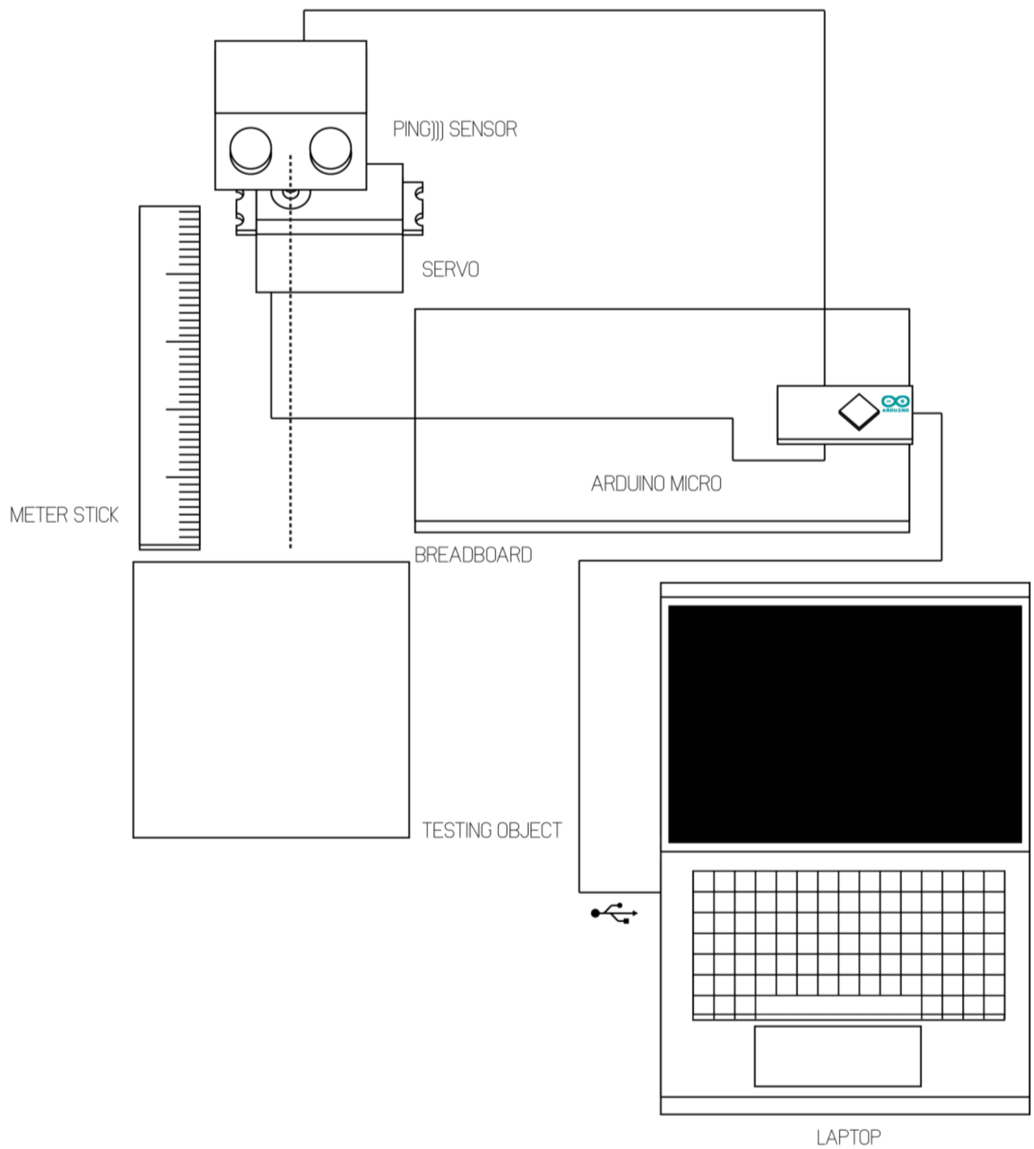
Conclusion:

Our servo motor will be able to rotate during the proof of concept. It will also be able to rotate during the performance of the task.

ARU – Radar Test for Movement – Ping Sensor Test

The Ping Sensor Test determines if the ultrasonic range finder functions properly.

Setup



Procedure

1. Wire Ping Sensor with Bread Board.
2. Wire Arduino Micro with Bread Board.
3. Provide power to Bread Board through 9V battery.
4. Interface corresponding pins to allow Arduino micro to control Ping Sensor.
5. Place object set distance from front of Ping Sensor.
6. Allow Ping Sensor to read distance.
7. Record results.
8. Repeat steps 5-7 with other set distances.

Results

Actual Distance (m)	Sensor Distance (m)
0.5	0.6
1.0	1.1
1.5	1.6
2.0	1.8
2.5	2.3
3.0	2.6
3.5	3.0

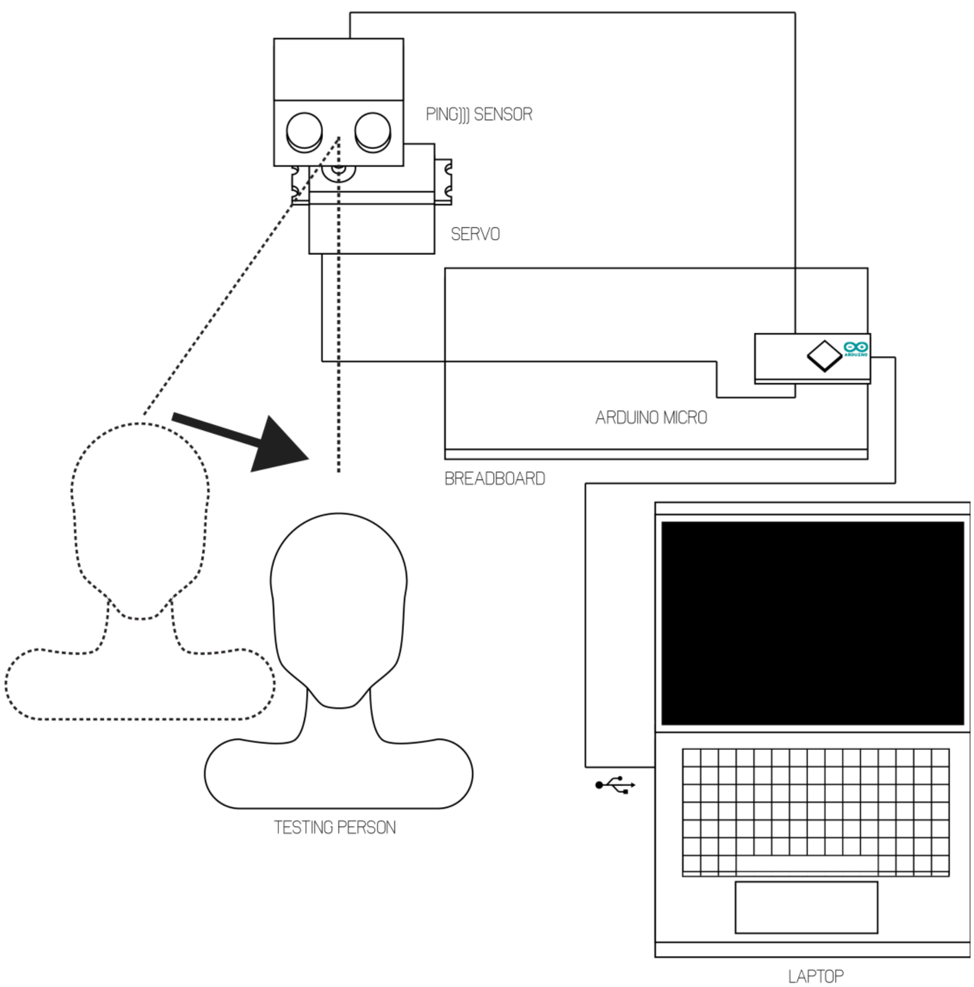
Conclusion

The Ping sensor is able to measure distance within a half meter tolerance. Although the exact distance is not important to the function of the test, it can be extra information used by the military in determining the location of unknown personnel.

ARU – Radar Test for Movement – Data Storage Test/Motion Detecting Test

This test will determine if it is possible to create an array of distance data stored as integers over 180 degrees. It will also determine whether it is possible to take four of these readings and compare them to each other to determine a change in position, and therefore movement, somewhere in the environment.

Setup



Procedure

1. Write algorithm to store data sets for every 180 degrees. The algorithm will store four sets of data, two clockwise and two counterclockwise. The algorithm will compare clockwise data sets to clockwise data sets, and counterclockwise data sets to counterclockwise data sets, in order to determine if there was a change in the position of anything in the environment between the two readings.
2. Wire Servo Motor to Bread Board.
3. Wire Ping Sensor to Bread Board.
4. Wire Arduino Micro to Bread Board.
5. Provide power to Bread Board through 9V battery.
6. Interface corresponding pins to allow Arduino Micro to control Ping Sensor and Servo Motor.
7. Place Bread Board set up in environment without motion.
8. Start Servo Motor rotation with Ping Sensor readings.
9. Record results.
10. Repeat test with someone moving in front of Ping Sensor.
11. Record results.

Results

The algorithm correctly compares data sets. In a stationary room, the sensor did not detect a change in distances while comparing data sets. The sensor did detect movement while someone was moving in front of it.

Conclusion

This radar module will be able to successfully function during the proof of concept. It will also be able to correctly detect movement during the performance of the task.

2.2 Geological Analysis System (GAS) Test

Mercury is designed to autonomously detect and travel toward buried mines. This capability is possible through the use of BG Micro Geophones and a Pyle Mini Speaker. The speaker generates a tone that causes the simulated mine to reverberate. This small movement is then picked up by the Geophone. The goal of this test is to determine if the Geophones can detect the simulated mine in a sand environment.

Sub Tests

- Frequency Test
- Decibel Test
- Distance versus LED Response Test (Geophone on Mine)
- Distance versus LED Response Test (Geophone next to Mine)
- Sand Test

Required Equipment

- Pyle Mini Speaker
- BG Micro Geophone
- BG Micro Electronic Meter Display
- Plastic Container (used interchangeably with Simulated Mine)
- Decibel Meter
- Frequency Generator
- Measuring Tape
- 9 Volt Battery

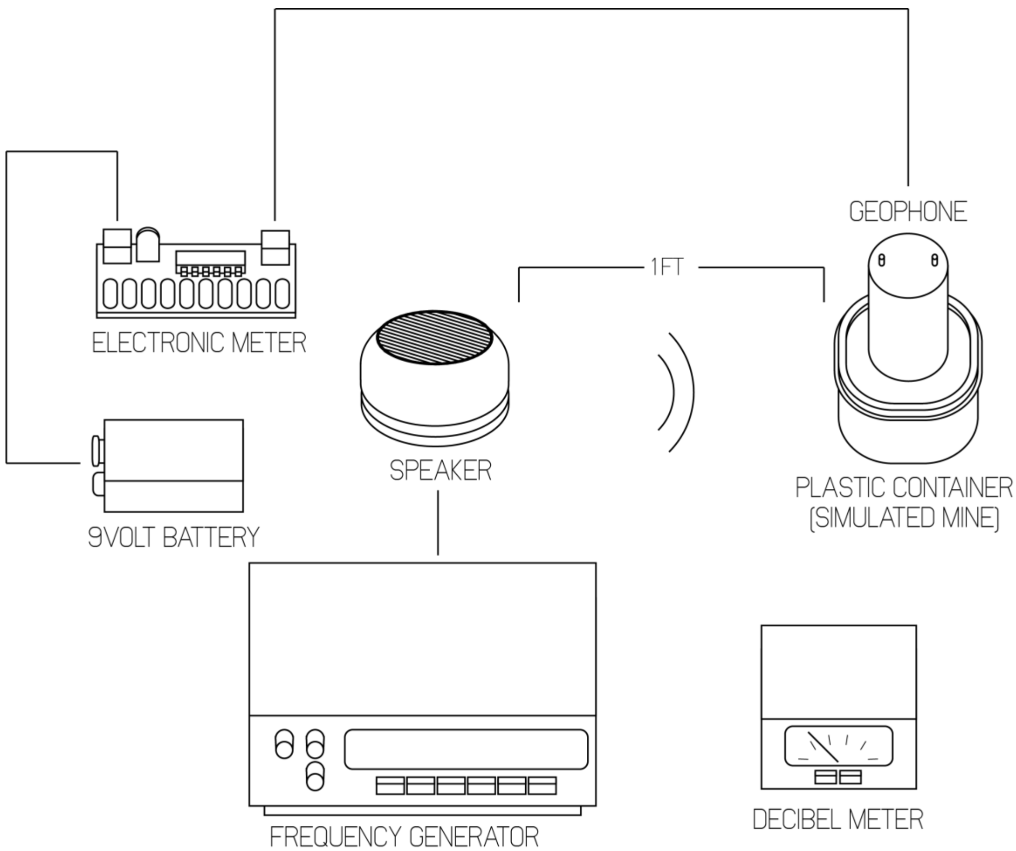
Special Notes

- Test areas should be free of extraneous movement and noise.
- The Pyle Mini Speaker configuration is always closed unless otherwise noted.

GAS –Frequency Test

The GAS Frequency Test determines the frequency emitted at which the simulated mine resonates most.

Setup



Procedure

1. Mark out at one foot intervals.
2. Place the Plastic Container at one end.
3. Place the Geophone on top of the Plastic Container.
4. Place the Speaker, open, at the other end.
5. Use the Frequency Generator to give off a constant 100 Hz sound at 84 dB.
6. Record the number of lighted LEDs on the Electric Meter.
7. Repeat steps 5-6, increasing the frequency by 25 Hz each time until 300 Hz is reached.
8. After finding the frequency range at which the most LEDs lights up, repeat steps 5-7, changing the frequency by 1 Hz each time.
9. After determining the frequency at which the most LEDs light up, close the Speaker and retest.

Results

*0.5 indicates flickering readings

Speaker Setting	**Frequency (Hz)**	**Lit LEDs**
Open	**85***	**3.5**
Open	**100**	**4.0**
Open	125	3.0
Open	150	2.0
Open	175	2.5
Open	200	2.5
Open	225	2.5
Open	250	2.0
Open	275	2.5
Open	300	2.5

* The frequency range of the Pyle Mini Speaker is 100Hz to 300 Hz. 85 Hz was the lowest setting at which we felt comfortable testing without damaging the Speaker.

Speaker Setting	**Frequency (Hz)**	**Lit LEDs**
Open	85	3.0
Open	86	3.5
Open	87	4.0
Open	88	4.5
Open	89	5.5
Open	**90**	**6.5**
Open	**91**	**6.5**
Open	92	5.0
Open	93	4.5
Open	94	4.5
Open	95	4.0

Speaker Setting	**Frequency (Hz)**	**Lit LEDs**
Closed	85	4.0
Closed	86	4.5
Closed	87	5.5
Closed	88	6.0
Closed	89	6.5
Closed	**90**	**7.0**
Closed	**91**	**7.0**
Closed	92	6.0
Closed	93	5.0
Closed	94	4.5
Closed	95	4.5

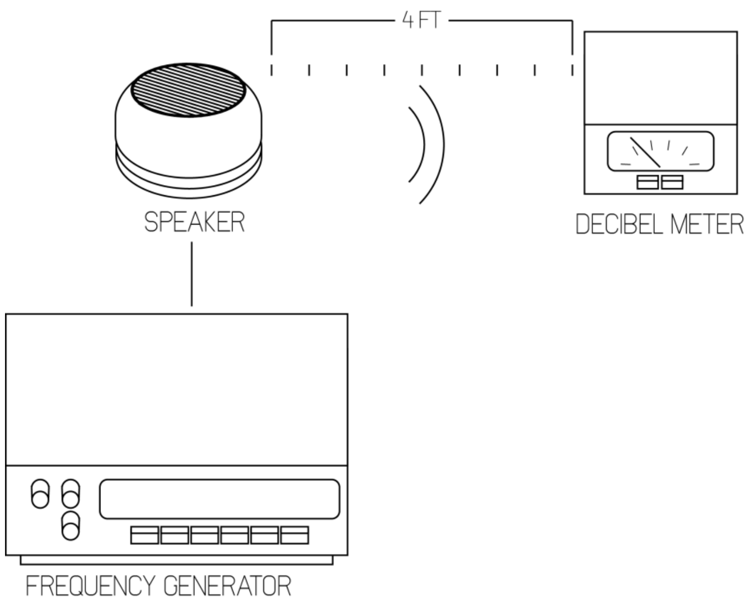
Conclusion

The Plastic Container resonates the most is at 90 and 91 Hz when the Pyle Mini Speaker is both open and closed. The BG Micro Electronic Meter displays 6.5 and 7.0 LEDS when the Pyle Mini Speaker is open and closed, respectively. Therefore, for optimal mine detection results, the Speaker should be emitting 90 Hz in the closed position.

GAS – Decibel Test

The GAS Decibel Test determines the frequency emitted at which the simulated mine resonates most.

Setup



Procedure

1. Mark out 6 inch intervals up to 4 feet. Set one end of the markings as the origin.
2. Place Pyle Mini Speaker at the origin.
3. Place the decibel meter at 6 inches away from the Pyle Mini Speaker
4. Use the frequency generator to give off a constant 90Hz sound at 84 dB.
5. Record the value on the decibel meter.
6. Increase the distance of the decibel meter from the Pyle Mini Speaker by 6 inches.
7. Repeat steps 5-6 until there are decibel values recorded for each interval.

Results

Distance (ft)	Decibel Value (dB)
0.5	69
1.0	63
1.5	61
2.0	61
2.5	60
3.0	56
3.5	53
4.0	50

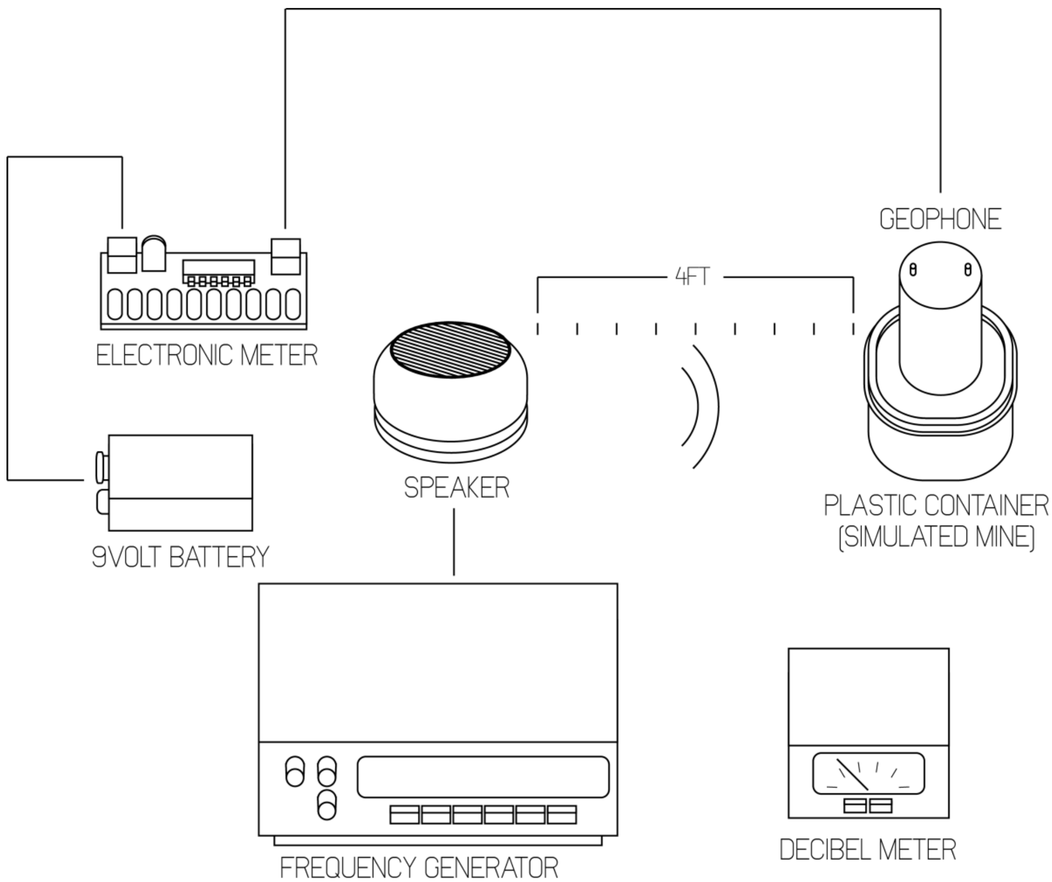
Conclusion

This test proves that there is an inverse relationship between the Pyle Mini Speaker distance and the decibel reading at the Decibel Meter.

GAS – Distance vs LED Response Test (Geophone on Mine)

The Distance vs LED Response, Geophone on Mine Test determines how much the vibration is detected when the Geophone and Simulated Mine are stacked while the Speaker distance from the aforementioned pair varies.

Setup



Procedure

1. Mark out 6 in intervals up to 4 ft. Set one end of the markings as the origin.
2. Place the Geophone and Plastic Container next to each other at the origin.
3. Place the Speaker at the 6 in mark.
4. Use the Frequency Generator to give off a constant 90 Hz sound at 84 dB.
5. Record the number of lighted LEDs on the Electronic Meter.
6. Repeat steps 4-5 until there are LED values recorded for each interval.

Results

Distance (ft)	Lit LEDs
0.5	4.5
1.0	4.5
1.5	4.0
2.0	4.0
2.5	4.5
3.0	4.5
3.5	4.5
4.0	5.0

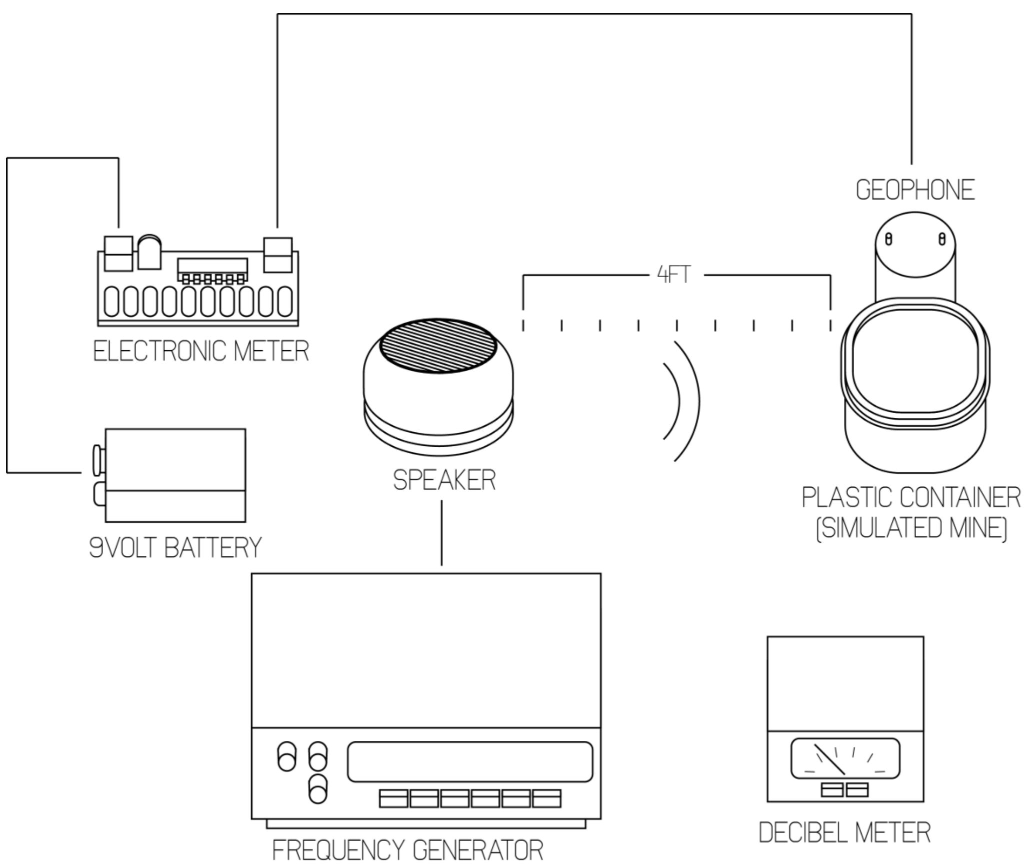
Conclusion

With the results of the test, it can be inferred that as the speaker moves farther away, the amount of LEDs activated decreases as well.

GAS – Distance vs LED Response Test (Geophone next to Mine)

The Distance vs LED Response, Geophone Next to Mine Test determines how much the vibration is detected when the Geophone and Simulated Mine are adjacent while the Speaker distance from the aforementioned pair varies.

Setup



Procedure

1. Mark out 6 inch intervals up to 4 feet. Set one end of the markings as the origin
2. Place the Geophone on top of the Plastic Container on top of each other at the origin
3. Place the Speaker at the 6 in mark
4. Use the Frequency Generator to give off a constant 90 Hz sound at 84 dB
5. Record the number of lighted LEDs on the Electronic Meter
6. Repeat steps 4-5 until there are LED values recorded for each interval

Results

Distance (ft)	Lit LEDs
0.5	7.5
1.0	7.0
1.5	3.0
2.0	6.5
2.5	6.0
3.0	6.5
3.5	1.5
4.0	6.5

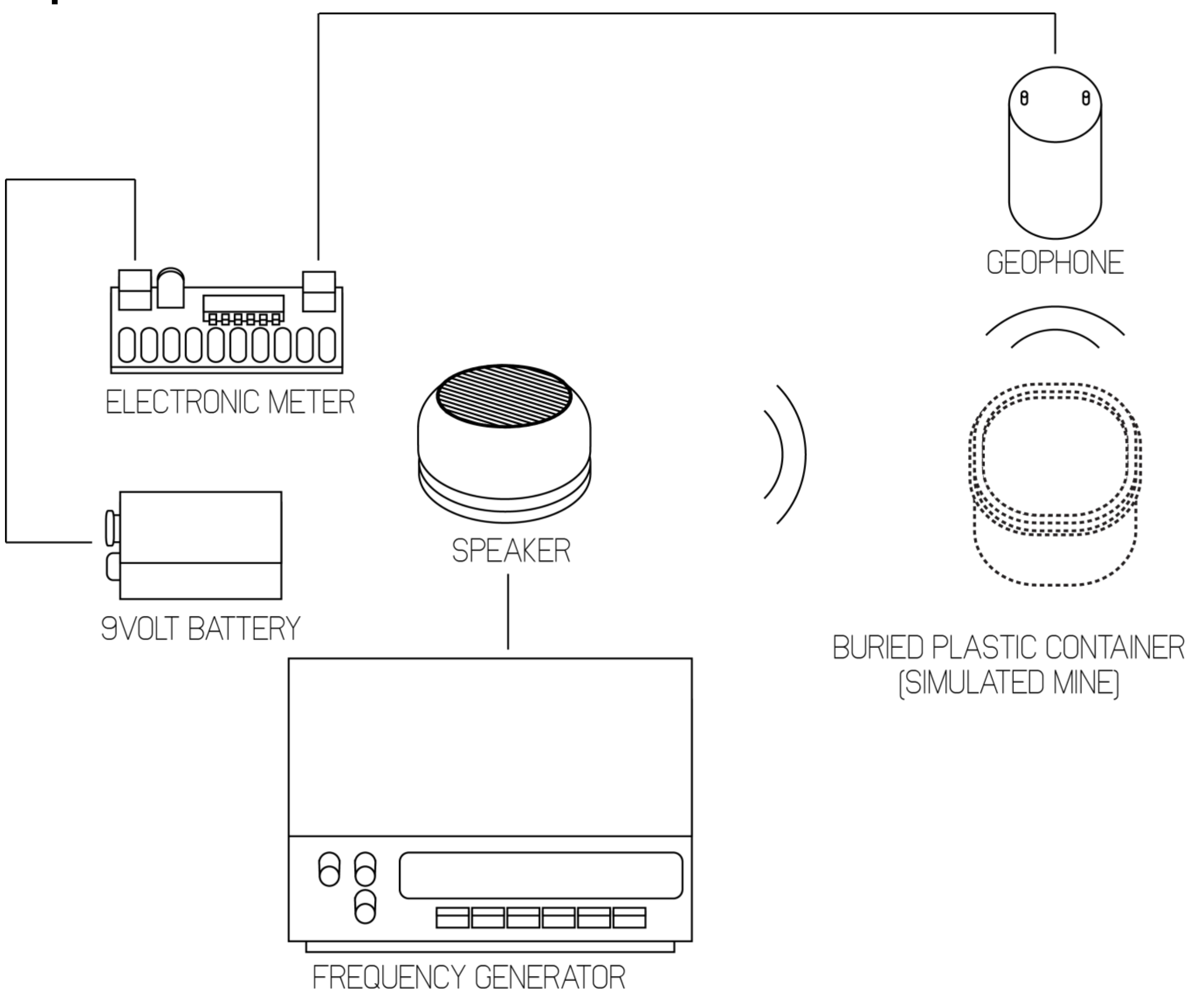
Conclusion

As the distance from the speaker increases, LED generally decreases. This test proves that the geophone picks up the vibrations from the plastic container, while the geophone is the previous test (GAS – Distance vs LED Response Test (Geophone next to Mine)) picks up the vibration from the table.

GAS – Sand Test

The goal of this test is to determine if the Geophones can detect the simulated mine in a sand environment.

Setup



Procedure

1. Mark out at one foot intervals.
2. Place the Plastic Container at one end.
3. Place the Geophone on top of the Plastic Container.
4. Place the Speaker, open, at the other end.
5. Use the Frequency Generator to give off a constant 100 Hz sound at 84 dB.
6. Record the number of lighted LEDs on the Electric Meter.
7. Repeat steps 5-6, increasing the frequency by 25 Hz each time until 300 Hz is reached.
8. After finding the frequency range at which the most LEDs lights up, repeat steps 5-7, changing the frequency by 1 Hz each time.
9. After determining the frequency at which the most LEDs light up, close the Speaker and retest.

Results

*0.5 indicates flickering readings

0.25" Depth

Speaker Position	Lit LEDs	Geophone Position	Lit LEDs
0.0	9.0	0.0	9.0
0.5	6.0	0.5	5.5
1.0	5.0	1.0	6.0
1.5	3.5	1.5	4.0
2.0	3.0	2.0	0.0
2.5	3.0	2.5	3.0
3.0	3.0	3.0	3.0
3.5	3.0	3.5	3.0
4.0	2.5	4.0	2.5

0.50" Depth

Speaker Position	Lit LEDs	Geophone Position	Lit LEDs
0.0	9.0	0.0	10.0
0.5	7.5	0.5	7.0
1.0	6.0	1.0	7.0
1.5	4.0	1.5	4.0
2.0	3.0	2.0	0.0
2.5	0.0	2.5	3.0
3.0	3.0	3.0	3.0
3.5	3.0	3.5	3.0
4.0	3.0	4.0	3.0

0.75" Depth

Speaker Position	Lit LEDs	Geophone Position	Lit LEDs
0.0	10.0	0.0	9.0
0.5	6.0	0.5	6.5
1.0	5.5	1.0	5.0
1.5	3.5	1.5	4.0
2.0	3.0	2.0	0.0
2.5	3.0	2.5	3.0
3.0	0.0	3.0	3.0
3.5	0.0	3.5	3.0
4.0	0.0	4.0	3.0

1.00" Depth

Speaker Position	Lit LEDs	Geophone Position	Lit LEDs
0.0	10.0	0.0	9.0
0.5	5.5	0.5	6.5
1.0	4.5	1.0	4.5
1.5	3.5	1.5	3.0
2.0	3.0	2.0	0.0
2.5	3.0	2.5	3.0
3.0	3.0	3.0	3.0
3.5	-	3.5	-
4.0	-	4.0	-

Conclusion

The test conducted in the larger sand pit followed the trends from the previous tests. This proves that the testing rig used in the previous sections was a viable testing arena.

2.3 Hydrological Analysis System (HAS) – Total Dissolved Solids Test for Water Drinkability

Mercury is designed to determine if soldiers can drink an unknown liquid based on its total dissolved solids. We are using a total dissolved solids (TDS) sensor to determine the TDS of varying saline solutions. The goal of this procedure is to differentiate between salt water and fresh water using the TDS sensor.

Sub-Tests

- Calibration Test
- Salinity Test
- Temperature Test

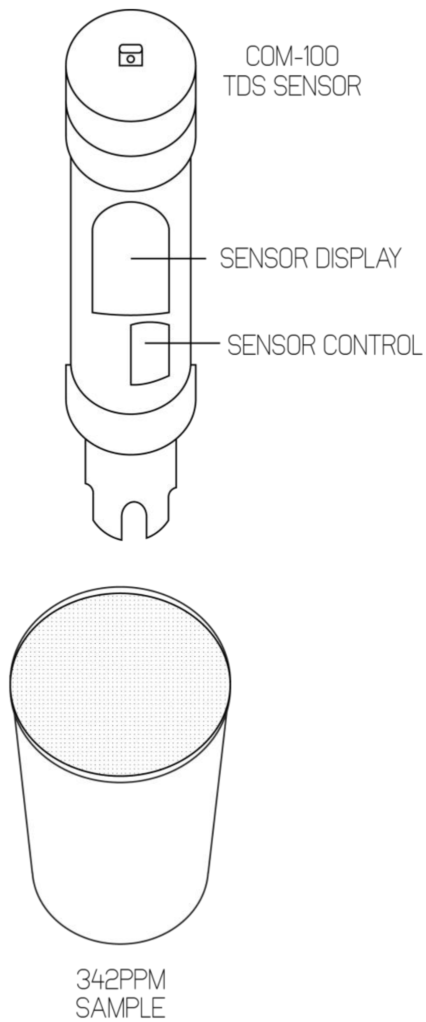
Required Equipment

- PPM Sensor
- Cups
- Sodium Chloride
- Fresh Water Sample
- 342 PPM NaCl Solution
- Digital Scale
- 100mL Graduated Cylinder
- Thermometer

HAS – Calibration Test

The calibration test calibrates the sensor, so it takes accurate total dissolved solids readings.

Setup



Procedure

1. Prepare NaCl 342 Parts Per Million solution
2. Put COM-100 TDS sensor in solution
3. Hold the "Mode" button until the sensor displays "NaCl"
4. Hold the "CAL" button after putting the sensor in the solution

Results

It is calibrated. This test was not explicitly performed because the sensor comes calibrated from the factory.

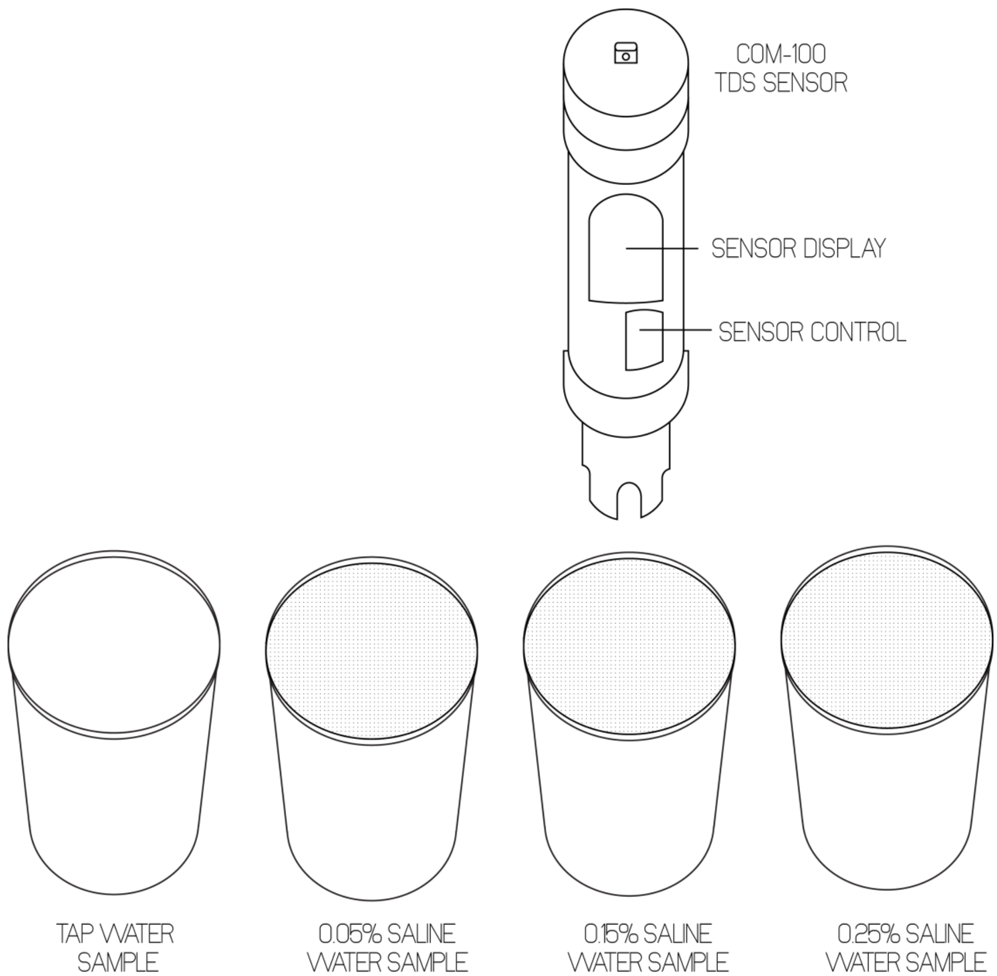
Conclusion

The COM- 100 sensor comes out of the package factory calibrated. Calibrating the instrument is important so that the results gotten from the sensor can be reliable and used to determine the temperature and salinity of water.

HAS – Salinity Test

The salinity test determines if there is sodium chloride in a water sample.

Setup



Procedure

1. Measure .05g, .15g, and .25g of Sodium Chloride using the digital scale and set the amounts aside separately.
2. Measure 99.95mL, 99.85mL, and 99.75mL of tap water using a graduated cylinder.
3. Measure 100mL of tap water for the tap water sample and put it in the cup.
4. Add .05g of Sodium Chloride to 99.95mL of water in a cup and mix thoroughly.
5. Add .15g of Sodium Chloride to 99.85mL of water in a cup and mix thoroughly.
6. Add .25g of Sodium Chloride to 99.75mL of water in a cup and mix thoroughly.
7. Test ppt of each saline solution using the COM-100 sensor
8. Record results of each solution.
9. Test ppm of 100mL of pure tap water.
10. Record results of the tap water.

Results

Solution	Total Dissolved Solids (PPT)
Tap Water	0.420
0.05% Saline Solution	0.700
0.15% Saline Solution	1.70
0.25% Saline Solution	2.63

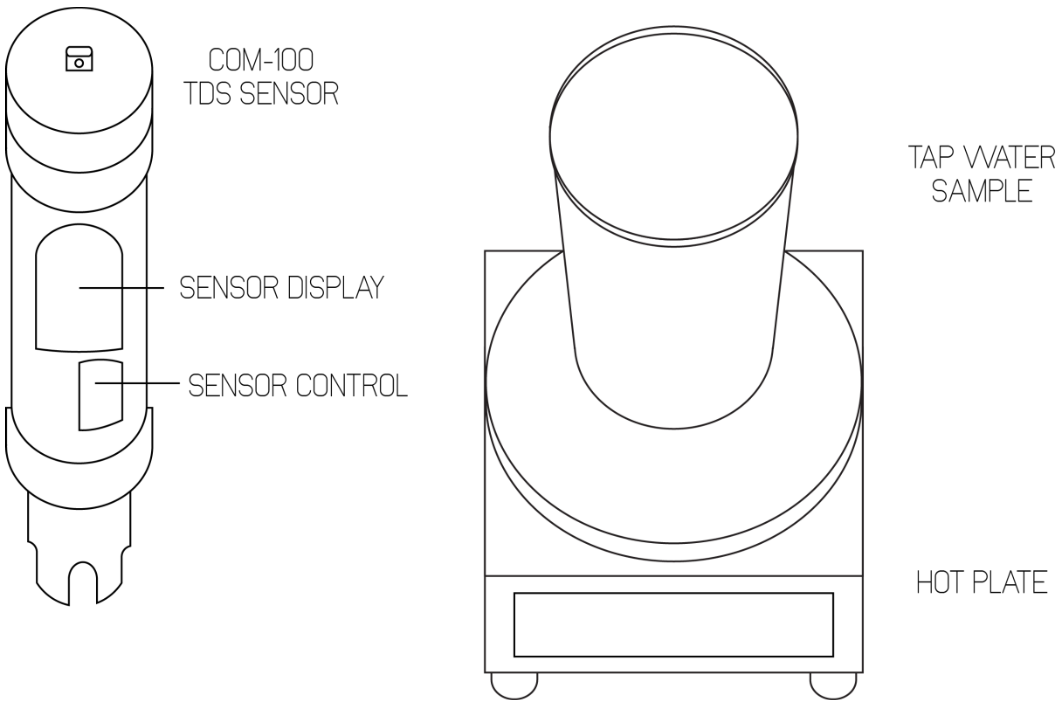
Conclusion

The sensor can correctly differentiate between salt water and fresh water. The concentration (parts per trillion (PPT) detected) increased as the percent of salt in the saline solution increased. The tap water was used as a control group and had a different PPT from the saline solutions. Determining if there is saline in the water will help soldiers determine if they can drink water that is around them.

HAS – Temperature Test

The temperature test determines the temperature of a water sample.

Setup



Procedure

1. Measure 100mL of tap water.
2. Pour 100mL of water into three separate cups totaling 300mL of water combined.
3. Alter one set of 100 mL of water using a hot plate and a flask until it is 40 ℃.
4. Measure the temperature of the water when heating it up using a thermometer.
5. Pour the water back into the cup when the thermometer reads the correct temperature.
6. Measure the temperature of the water using the COM-100 sensor.
7. Record results.
8. Repeat the above steps for a temperature of 60 ℃.

Results

Temperature	Sensor Temperature
25 ℃	24 ℃
40 ℃	38 ℃
60 ℃	59 ℃

Conclusion

The sensor can correctly tell the temperature of a solution. The sensor temperature might also differ slightly because the water that was being tested could've cooled down from the initial measurement. Measuring and knowing the temperature of a liquid will help soldiers prevent themselves from being burned when dealing with the aforementioned liquids.

3.1 Budget

For Project Morpheus, the CARPA Initiative instated a budget limit of $1,500. As of the 31st of January, Delta Technologies' expenditures sit at a total of $972.37. Of that total, $382.32 have been spent on the manufacturing and testing of Mercury's flight system, $100.59 have been spent on procuring and testing the components necessary for Mercury's Geological Analysis System, $67.65 have been spent on procuring and testing the components necessary for Mercury's Hydrological Analysis System, and $375.14 have been spent on the materials necessary to program Mercury.

Delta Technologies has $527.64 remaining in its active budget and is on track for successfully manufacturing and testing its soon-to-be-built final iteration of Mercury.

Total Budget	**$ 1,500.00**
Expenditures	**$ 972.37**
Remaining	**$ 527.64**

ARU Bill of Materials

Vendor	Part #	System	Description	QTY	Unit Price	Total Price
OMGfly		ARU	SunnySky X4108S 600KV motor(NEW VERSION)	2	$ 34.00	$ 79.50
Hobby King	--	ARU	CW prop	2	$ 3.79	$ 7.58
Hobby King	--	ARU	CCW prop	2	$ 3.85	$ 7.70
Hobby King	HXT4MM-PAR	ARU	HXT4mm Battery Harness 14AWG for 2 Packs in Parallel	2	$ 2.83	$ 5.66
Hobby King	258000038	ARU	HXT 4mm to 4 X 3.5mm bullet Multistar ESC Power Breakout Cable	2	$ 3.65	$ 7.30
Hobby King	--	ARU	Turnigy 1000mAh 4S 30C Lipo Pack	1	$ 12.18	$ 12.18
Hobby King	TR_PC	ARU	TURNIGY ESC Programming Card	1	$ 6.95	$ 6.95
Hobby King	TR_P18A	ARU	TURNIGY Plush 18amp Speed Controller	2	$ 11.90	$ 23.80
Hobby King	AM1001A	ARU	PolyMax 3.5mm Gold Connectors 10 PAIRS (20PC)	2	$ 1.46	$ 2.92
Hobby King	--	--	Hobby King order shipping costs	--	--	$ 17.95
HobbyPartz	--	ARU	Banana Plug With Protect Pipe 34MM	5	$ 3.60	$ 18.00
HobbyPartz	--	ARU	Thunder AC6 Smart LiPo Charger/Discharger	1	$ 43.95	$ 43.95
HobbyPartz	--	ARU	Li-Po GUARD Safety Battery Charging/Storing Bag	1	$ 5.75	$ 8.44
HobbyPartz	--	--	Hobby Partz order shipping costs			$ 13.08
McMaster	93330A483	ARU	Aluminum Female Threaded Round Standoff 1/4" OD, 2" Length, 8-32 Screw Size	6	$ 1.36	$ 8.16

McMaster	89015K18	ARU	Multipurpose aluminum alloy 6061 .125"thick, 12" x12"	1	$ 28.02	$ 32.04
McMaster	93330A463	ARU	Aluminum Female Threaded Round Standoff 1/4" OD, 2" Length, 4-40	6	$ 1.36	$ 8.16
McMaster	91253A113	ARU	Alloy Steel Flat Head Socket Cap Screw 4-40 Thread, 3/4" Length (Pk. of 100)	1	$ 11.61	$ 11.61
McMaster	92949A108	ARU	18-8 SS Button Head Socket Cap Screw 4-40 Thread, 3/8" Length (Pk. of 100)	1	$ 3.49	$ 3.49
McMaster	90633A005	ARU	Grade 2 Steel Nylon-Insert Thin Hex Locknut Zinc-Plated, 4-40 Pk. 100)	1	$ 2.85	$ 2.85
McMaster	92395A315	ARU / Machining	Brass Press-Fit Expansion Insert, W/Flange, 4-40 (Pk. of 25)	1	$ 6.50	$ 6.50
McMaster	8974K32	ARU /Test	Multipurpose Aluminum (Alloy 6061) 3/8" Diameter X 6' Length	1	$ 11.85	$ 11.85
McMaster	94738A300	ARU /Test	Plain Steel Knock-in Insert for Wood 1/4"-20 Internal Thread, .414" Length	1	$ 4.88	$ 4.88
McMaster	6338K414	ARU /Test	SAE 841 Bronze Flanged-Sleeve Bearing for 3/8" Shaft Diameter, 1/2" OD, 1/4" Length	8	$ 0.70	$ 5.60
Plastics Depot	--	ARU /Test	Polycarbonate Stock	1	$ 32.17	$ 32.17
					ARU Total	$ 321.32
					ARU / Machining Total	$ 6.50
					ARU /Test Total	$ 22.33
					Total	$382.32

GAS Bill of Materials

Vendor	Part #	System	Description	QTY	Unit Price	Total Price
Adorama	PYPMS2B	GAS	Speaker	1	$ 12.00	$ 15.35
BGMicro	KIT1019	GAS	Geophone sensor kit	1	$ 39.95	$ 35.96
BGMicro	ACS1582	GAS	Geophones	2	$ 22.20	$ 39.96
BGMicro	--	--	BG Micro order shipping costs	--	--	$ 9.32
					Total	$ 100.59

HAS Bill of Materials

Vendor	Part #	System	Description	QTY	Unit Price	Total Price
Amazon	--	HAS	HM COM100 Waterproof EC/TDS And Temperature Combo Meter	1	$ 49.31	$ 53.75
ECrater	--	HAS	MQ-7 Carbon Monoxide CO Gas Sensor module Detecters f Arduino	1	$ 13.90	$ 13.90
					Total	$ 67.65

Programming Bill of Materials

Vendor	Part #	System	Description	QTY	Unit Price	Total Price
Ebay	--	Recon	9DOF Sensor	1	$ 18.55	$ 18.55
DigiKey	Pandaboard ES	Recon	Pandaboard ES	1	$ 161.64	$ 167.44
Ebay	Asus Xtion	Recon	ASUS XTION Pro Live	1	$ 159.99	$ 159.99
Adafruit	1086	Recon	arduino micro	1	$ 24.95	$ 29.16
					Total	$ 375.14

3.2 Yearlong Schedule

Task	% Complete	Sept.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.	Apr.	May	Jun
SPS Document	100%										
SPS Presentation	100%										
ARM Concept	100%										
GAS Concept	100%										
DM Concept	100%										
HAS Concept	100%										
TS Concept	100%										
Proof of Concept Document	0%										
Proof of Concept Presentation	0%										
ARM System Test	0%										
GAS System Test	0%										
HAS System Test	0%										
Programming	0%										
Prototype Testing	0%										
Robot Manufacturing	0%										
Robot Assembly	0%										
Source Control Drawings	0%										
Test and Evaluation/Modification	0%										
IDP Expo	0%										

	In Progress		Complete		Not Yet Started

4.1 Appendix A: Subsystem Acronyms

GAS - Geological Analytical System

HAS - Hydrological Analytical System

ARU – Aerial Reconnaissance Unit

4.2Appendix B: Water Ratings for HAS

Water Rating Table:

Total Dissolved Solids (PPM)	**Description**
0-50	Ideal drinking water, distillation
50-100	Carbon filters, mt. springs, aquifers
100-200	Hard water
200-300	Marginally acceptable
300-400	Average tap water
400-500	High TDS from tap or mineral springs
500+	U.S. EPA's max contaminant level
1000 - 5,000	Brackish water, Mildly
5000 - 15,000	Brackish water, Moderately
15,000 - 35,000	Brackish water, Heavily
30,000 - 50,000	Sea water

This Water Rating Table was obtained from "Water Filters Online" <*http://www.waterfiltersonline.com/tds-sources.asp*>.

4.3 Appendix C: ARU Motor Specifications

Motor Specifications:

Model: X4108S 600KV

KV (rpm/v): 600

Weight: 113g

Diameter: 46mm

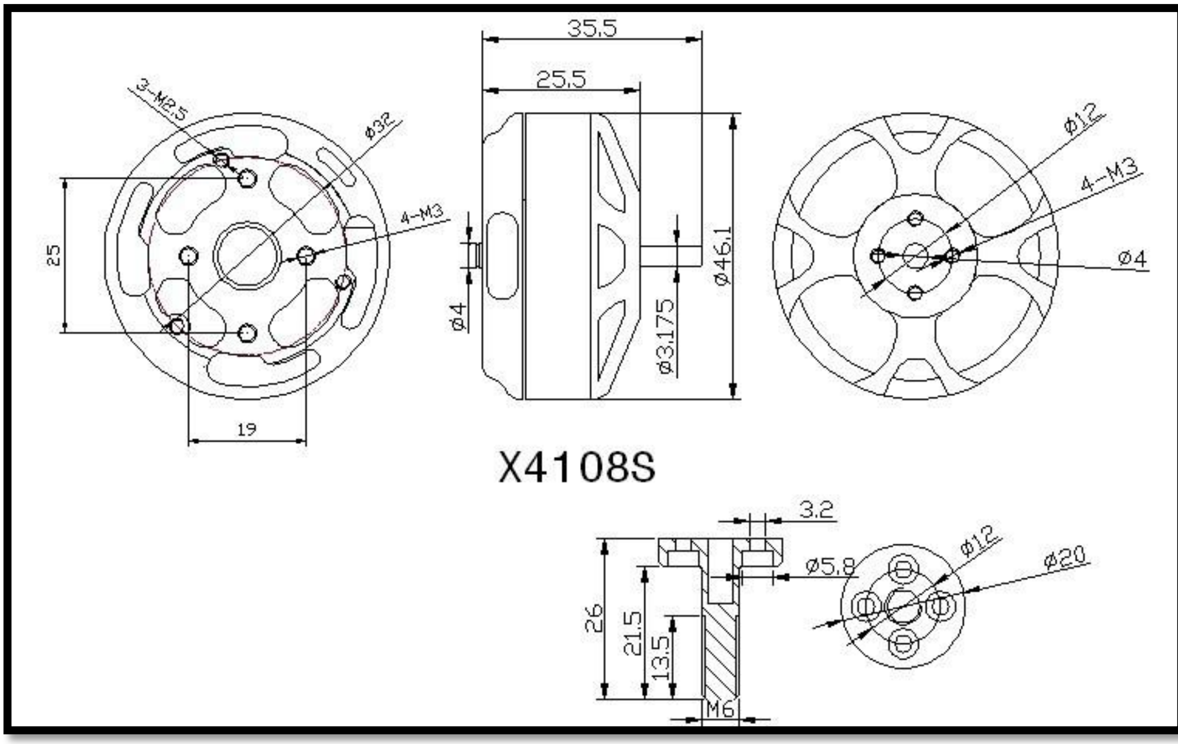
Length: 25.5mm (excluding motor shaft)

Wire length: 60cm

Prop adaptor output shaft diameter: 6mm

Application: specially designed for quad-copter and multi-rotor aircrafts.

Type	Prop size (inch)	Voltage (V)	Throttle	Current (A)	Pull (g)	RPM	Power (W)	g/w ratio
X4108S kv600	APC 11 x 4.7	14.8	50%	2.5	420	4014	37	11.35135
			65%	5	700	5000	74	9.459459
			75%	8.7	970	5930	128.76	7.533395
			85%	11	1130	6330	162.8	6.941032
			100%	13.5	1290	6741	199.8	6.456456



Motor Dimensions

4.4Appendix D: ARU Test Code

The program sets certain keys on the keyboard as throttle up and throttle down for controlling the motor, like on a remote control. This gives the ability to vary the output of the motor by increments of 5-10%. The program will then relay the information to the motor in order to run it at a certain speed.

Commented Code:

```
#include <Servo.h>

//include the Servo library

intmotorPin = 2, _read = 0, val=45;

//set a variable for the motor pin, for serial input,
and the value of the motor

Servo propeller;

//create an object of the Servo class

void setup(){

propeller.attach(motorPin);

 //correlate the propeller object with the pin the
motor controller is plugged into

Serial.begin(9600);

//begin serial communication at 9600 baud

}

void loop(){

propeller.write(val);

 //write a value of the variable val to the propeller
(currently 0 throttle)
```

```
if (Serial.available() > 0) {

//if there is something in the serial stream

    _read = Serial.read();

//set whatever is in the serial stream to the _read
variable

if(_read==110){

//if the character 'o' was entered

val+=5;

//increment the speed of the motor by 5

Serial.println(val);

//display the new value for debugging

}else if(_read==111){

  //or else, if the character 'p' was entered

Serial.println(val++);

      //increment the value, and print it out

}else{

      //or else

Serial.println(_read);

//print the value that was read

    }

  }

}
```

4.5 Appendix E: Ultrasonic Range Code

This program stores data sets for every 180 degrees of rotation. The program will store four sets of data: two clockwise and two counterclockwise. The program will compare clockwise data sets to clockwise data sets, and counterclockwise data sets to counterclockwise data sets, in order to determine if position changes occurred in the environment in the time interval between the two readings. These readings occur from a couple milliseconds to one second.

Commented Code:

```
#include <Servo.h>
//include the Servo library

Servo radar;
//create an object of the servo class

intservoPin = 3, pingPin = 2, turn = 1, threshold
= 10;
//declaring a few variables of the integer type

unsigned long cw1[180], cw2[180], ccw1[180],
ccw2[180];
//declares arrays of type long, one for each
direction, per sweep
//of the ping sensor

void setup()
{
  //the setup loop
radar.attach(servoPin);
  //associates the servo to the data pin the servo
is plugged in to
Serial.begin(9600);
  //begins the Serial stream
```

```
}

void loop()
{
for(int i=0;i<180;i++){
    //a loop that goes from 0 - 179
radar.write(i);
//sends the value that the loop is on, to the
   servo
pingSetup();
    //calls the pingSetup function, which prepares
the ping sensor to
//give back data
if(turn == 1){
//happens if the module is on its first pass
cw1[i] = pulseIn(pingPin, HIGH)/29/2;
//the sensor returns 29 microseconds per
centimeter, roundtrip
        //calculate the centimeter value and add
it to the array
if(cw2[i]){
 //if there is data from the previous pass
if(abs(cw2[i]-cw1[i])<threshold){
   //if the absolute value of the current value
subtracted
   //from the previous value of the same position
is less than the
   //user defined threshold
Serial.println("stayed");
   //consider the result of this position to be
not have moved
}else{
Serial.println("moved");
//if it is greater than the threshold thenconsider
the result
            //of this position to be moved
}
}
}else if(turn == 2){
        //if it is currently the second pass
```

```
cw2[i] = pulseIn(pingPin, HIGH)/29/2;
      //the sensor returns 29 microseconds per
centimeter, roundtrip
        //calculate the centimeter value and add
it to the array
}
Serial.println(pulseIn(pingPin, HIGH)/29/2);
//print the value to the Serial stream for
debugging
}
delay(1000);
    //pause for one second
for(int i=180;i>1;i--){
    //a loop that goes from 180 - 2
radar.write(i);
      //sends the value that the loop is on, to
the servo
pingSetup();
      //calls the pingSetup function, which
prepares the ping sensor to
      //give back data
if(turn == 1){
      //happens if the module is on its first pass
ccw1[i] = pulseIn(pingPin, HIGH)/29/2;
//the sensor returns 29 microseconds per
centimeter, roundtrip
//calculate the centimeter value and add it to the
array
if(ccw2[i]){
//if there is data from the previous pass
if(abs(ccw2[i]-ccw1[i])<threshold){
    //if the absolute value of the current value
subtracted
    //from the previous value of the same position
is less than the
    //user defined threshold
Serial.println("stayed");
   //consider the result of this position to be
not have moved
}else{
```

```
Serial.println("moved");
//if it is greater than the threshold then
consider the result
            //of this position to be moved
}
}
}else if(turn == 2){
        //if it is currently the second pass
ccw2[i] = pulseIn(pingPin, HIGH)/29/2;
//the sensor returns 29 microseconds per
centimeter, roundtrip
        //calculate the centimeter value and add
it to the array
}
Serial.println(pulseIn(pingPin, HIGH)/29/2);
      //print the value to the Serial stream for
debugging
}
delay(1000);
//wait for one second
if(turn == 1){
//if it is currently the first pass
turn++;
      //increment the variable to make it the
second pass
}else{
turn = 1;
//or make it the first pass
}
}

voidpingSetup(){
//the function to setup the sensor to get data
from it
pinMode(pingPin, OUTPUT);
   //sets the pin to output
digitalWrite(pingPin, LOW);
   //write a value of low to the pin
delayMicroseconds(2);
   //waits 2 microseconds
```

```
digitalWrite(pingPin, HIGH);
   //writes a value of high to the pin
delayMicroseconds(5);
   //waits 5 microseconds
digitalWrite(pingPin, LOW);
   //write a value of low to the pin
pinMode(pingPin, INPUT);
   //set the pin to be an input
}
```

5.1 Contact Delta Technologies

Project Manager Liani Lye

Deputy Project Manager David Doan

Chief Scientist Babatunde Alford

Chief Strategist Scott Nakatsu

Chief Financial Officer Arturo Munoz

Media Specialist Jyrrl Kliff Figueroa

Systems Engineer Obaid Miah

Test Engineer Stephanie Leyva

Programming Lead Misbah Khan

Software Engineer Austin DeYoung

Mechanical Lead Thomas Arakaki

Mechanical Engineer Eric Johnson

Electrical Lead Justin Griswold

Electrical Engineer Aaron Rangel

Manufacturing Lead Michael Delorio

Manufacturing Engineer Joshua Brown

6.1 CARPA Approval

Rules, Conditions, and Requirements are subject to change by the CARPA authority. All changes will be discussed in class, and be presented to the company teams in writing.

Program Manager – Delta Technologies	Date
Deputy Program Manager – Delta Technologies	Date
Chief Scientist – Delta Technologies	Date
CARPA Authority	Date