Final Project Report

Wildlife Deterrent Test Device

Report Authors: Garrett Tietz gtietz@calpoly.edu

Adam Webb alwebb@calpoly.edu

Dane Knutson dwknutso@calpoly.edu

Advisor: Dr. Peter Schuster

Sponsor: Kevin Charles, JumpSport

Date: December 10, 2014

Statement of Disclaimer

Table of Contents

List of Figures

List of Tables

Executive Summary

This document presents a complete documentation for the design and development of the Wildlife Deterrence Method Test Device senior project thus far. The project was commissioned and sponsored by JumpSport, Inc. as a device which will aid JumpSport employees in evaluating the effectiveness of multiple methods of deterring wildlife, most notably deer.

The system will consist of a modular system of multiple components controlled by a central electronics unit. This system will allow the user to specify which methods to test and the manner that they will be tested. There will be methods for notifying the user that an action was taken and the capability of recording the performance of each sub-system for later evaluation and analysis.

The methods deemed most probably to be successful are a method of physically approaching the animal, a method of simulating an approach with strategically placed speakers and an industry standard impact sprinkler. Additionally, a method of determining the location of the animal using an ultrasonic rangefinder will be tested. The information from the rangefinder will not be used to determine the location of the animal; it will simply be relayed to the user as a proof of concept. These functions will be controlled by Arduino microcontroller boards with Atmel microcontrollers.

Ch 1. Introduction

JumpSport Inc. develops and manufactures trampolines and safety enclosures for recreational uses and exercising. JumpSport is looking to expand its company from trampolines to the animal repellant field, specifically orientated towards deer that live near or in suburban areas. There are many types and brands of repelling devices on the market but none are effective in the long term. The goal of this project is to design, build, and test a machine or system that will be used to determine successful methods for deterring deer.

The objective of the Deer Busters team is to design and build a device or system of devices that will be used to determine which method, or methods, are most effective at deer deterrence. JumpSport suspects that a method which gives the appearance of approach to the deer in an aggressive or startling manner but also changes the way it attacks so that the deer do not get used to the device will be most effective. Deer Busters is committed to the completion of the deer deterrent testing device by the end of the fall quarter of 2014.

Objectives

Deer and other wildlife pests frequently enter properties and damage or destroy many types of plants. Homeowners and farmers fight a constant battle with these pests to protect their crops and flowers usually employing scented deterrents or motion sensing “scare devices.” These scare devices often work for a short amount of time, after which the animal becomes accustomed or habituated to the stimulus and ignore it. In the case of scented deterrents, they must be reapplied frequently and are less effective in the rain. Effective solutions include fencing and other clever landscaping techniques, but some users may be unable or unwilling to implement this solution.

Deer, while foraging for food, will often feed on (and damage) plants which a person may want to keep intact. Unfortunately, it is not well known how to effectively prevent this activity. A device or system is needed which can be employed, in a suburban to rural setting, to determine an effective method of preventing deer and other wildlife from damaging plants in the desired area. JumpSport has commissioned the design of such a product.

We have developed a method which will effectively test various deterrent methods. We focused on addressing the issue of habituation by attempting to vary the stimulus such that the animal will continue to be startled, even after continuous use. We will also prototype a device

or system to apply these methods with the goal of providing JumpSport with a product to aide in the testing of methodology for future development into marketable products.

Customer Requirements

The product will attempt to consider all of the requirements from the various customers, where customers consist of eventual customers, the engineers conducting tests, the installer, and purchaser of the device. The customer requirements are categorized into Needs and Wants, and are listed below.

Needs

- Methods are Viable for Deterring Deer
- Results are Verified
- Durable and Weatherproof
- Safe for Humans
- Safe for Animals

Wants

- Easy to Test
- Marketable Components
- Notification of Malfunction
- Uses Minimal Resources
- Not Activated by Humans
- Minimal Maintenance
- Reliable
- Low Cost
- Not Obnoxious

Engineering Specifications

In order to satisfy the customers, we will consider the Engineering Specifications, detailed in Table 3. These specifications will be measured against the target by testing, inspection, analysis and/or similarity to existing designs.

Table 1. Deer Scaring Device Engineering Requirements.

	Parameter Description	Target	Tolerance	Risk	Compliance Method
1	Reduce Deer Invasions	90% Reduction	Min	M	Test
2	Reduce Non-Deer Invasions	60% Reduction	Min	L	Test
3	Effective Area	1,000 sq. ft	Min	L	Inspection
4	Initial Cost	$1,000	Max	L	Analysis, Inspection
5	Results Documented	99% of Time	Min	M	Test
6	Time Between Interactions	4 Weeks	Min	H	Test
7	Is Not Activated by Humans	95% of Time	Min	H	Test
8	Ongoing Costs	$5 per Month	Max	M	Analysis
9	Voltage Requirement	120 V	Max	M	Inspection
10	Projectile Energy	½ Joules	Max	M	Analysis
11	Interactions Required	None	Min	H	Inspection
12	Shock Exposure	5 mA	Max	M	Analysis, Inspection
13	Operation Temperature	32-120°F	Min	M	Analysis
14	Sound Requirements	65 dB at 15 ft	Max	M	Inspection
15	Notification of Maintenance Need	99% of time	Min	H	Inspection

These engineering specifications were developed using the QFD method or "House of Quality." The House of Quality is constructed by first creating a list of customers who will be affected by the product. This can include end-users, manufacturers, installers, etc. Next the customer requirements are listed and given certain importance ratings for each of the customers. Next the current situation is analyzed on the far right of the QFD. Here several other options for customers are listed and ratings are awarded based on how well each competitor meets each customer requirement. This helps us to visualize gaps in the marketplace which can be exploited. Next the engineering specifications are developed. Each engineering specification must be a measurable criterion to determine if the product is successful at meeting one or more customer requirements. Now the customer requirements are compared to the engineering specifications in the middle section allowing the engineer to ensure that all customer requirements are addressed. Finally, each engineering specification is compared to the others to determine correlations. This can be useful in determining redundancies or predict inherent constraints. The completed QFD chart is provided as Appendix 2.

Ch 2. Background

Deer Behavior

The underlying problem is that deer enter yards in an effort to find food, generally causing damage to plants and property. These intrusions are often countered with a number of existing products; however, deer tend to habituate to the different stimuli caused by these products and simply learn to ignore them.

Understanding deer behavior is an essential step toward minimizing yard intrusions. Specific knowledge such as eating habits, deer territory range, level of activity, and deer sensory information are vital to the success of the project.

Sensory Abilities

Deer have a very well-developed sense of smell. It is arguably the most sensitive of the deer's senses. Deer repellents have been made to exploit their keen sense of smell. Repellents using rotten egg odors seem to outperform other types of repellents. (Mattern)

A deer's eyesight is similar to other prey animals. Since deer's eyes are located on the side of their head, they have poor depth perception but may see nearly all the way around them. Deer eyes contain a higher proportion of rods to cones than humans, enabling them to see better at night, but at the expense of poor receptivity of color and daylight. Even with fewer cones deer are able to distinguish between shorter wavelengths of light such as blue, violet, and perhaps ultraviolet. However, deer lack a type of cone which is "sensitive to long wavelength colors such as red and orange" (Murphy). Since deer lack sensitivity to longer wavelengths of light, they will be unable to detect infrared radiation. This ensures that devices using infrared wavelengths, such as cameras or sensors, will not have an effect on deer activity. (Murphy)

Deer's sense of taste is comparable to other herbivores. They are drawn to certain types of plants that are easy to digest, such as shoots, and tend to avoid plants that are bitter. This has also been exploited in repellents. Furthermore, deer will pursue nearly any plant, regardless of taste or smell, if pressed by hunger. (DeNicola)

A deer's sense of hearing is similar to that of humans. Deer are able to pinpoint the source of a noise by rotating their ears individually. They can hear at slightly higher frequencies than humans, which has been exploited by ultrasonic scaring devices. However, deer are able to

habituate to simple noise generation easily. A deer's hearing ranges from 0.25 kHz to 30 kHz, with the best hearing range from 4 kHz to 8 kHz (Hearing).

Activity

Deer tend to be most active at dusk and dawn, especially in more urban environments, when human activity is low. However, deer may be active at all times of the day (Factors).

Deer are generally found on most continents and climates, however, "temperature and snow depth exerted the strongest effects" (Factors). Activity decreases greatly below temperatures of 0° Celsius. According to a study performed by Paul Beier and Dale R. McCullough, a given deer population exhibited inactivity for 40% of the day when under -10° Celsius and under 50% of the day for under 0° Celsius. Deer spent nearly 65% of the day inactive with a snow depth of 24 centimeters and exhibited activity inversely proportional to snow depth (Factors).

Deer tend to be territorial animals and prefer to live in regions around 1 square mile in area. This indicates that a specific yard will generally be dealing with the same deer populations, likely explaining the fact that deer will habituate quickly to stimuli in a given yard (Factors).

Using "less palatable herbaceous and woody plants" will help minimize deer intrusions, however, deer will feed on most plants if their survival is threatened by hunger (DeNicola). Therefore, few plants are completely 'deer-proof'.

Detecting Deer

Detecting deer upon an intrusion is vital to the success of the system. Furthermore, differentiating between a human, pet, and pest is a difficult but very important ability of a pest deterrent system. Current systems mostly use contact, infrared, and motion sensors. A variety of sensing technologies are summarized in the table below (Robot Shop):

Table 2. List of deer detection devices with details

Title	Likely Effectiveness	Range	Relative Cost	Complexity
Cameras (motion)	High	100 ft	$75	High
Contact sensors	Low	Contact	<$10	Low
Infrared sensors and light sensors	High	50 ft	$20	Low
Laser scanners and rangefinders	High	200 ft	$700	High
Ultrasonic rangefinders	Medium	15 ft	$15	Medium
Sound sensors	Medium	20 ft	$15	Medium

Less expensive options are not able to explicitly discern the difference between a deer and a human. Complicated solutions using image recognition software may be able to specifically target deer, however they are difficult to use and expensive to make. At the moment, no conclusion can be made regarding the superiority of any one sensing technology over another due to the mentioned variables in both cost and performance.

It is important to be able to detect deer when entering and while inside the guarded yard. Deterring deer at all locations within a yard is essential for protecting plants and altering deer feeding habits. Therefore, the deterrent system must be able to detect deer anywhere within the yard.

Existing Products

Many products and devices exist that effectively fit the needs of specific yards but are not versatile or convenient enough to be applied to most yards. This includes tall fencing, dogs or other pets, electric fencing, and other solutions.

Since the focus of this project is to develop a device that is able to work for the majority of yards and is convenient for the user, the top products and devices that offer this versatility were researched.

Repellents

The effectiveness of repellents depends on several unpredictable variables including alternate foraging options, deer nutrition, deer density, deer population size, and yard size. They are costly and work-intensive to apply and also must be reapplied every 30 days. Repellents are categorized into four types: fear, conditioned aversion, pain, and taste (DeNicola). Scent-based repellents, such as rotten eggs, tend to yield superior results when compared to taste-based repellents. A study of different commercial deer repellent effectiveness reported that the repellent Bobbex had the highest ratings at 93% (Mattern).

Frightening techniques

Many frightening techniques give temporary relief before the deer habituate to the stimulus. Lack of positive punishment allows deer to habituate to a particular stimulus without fear of consequence. Frightening is most effective when instigated prior to an invasion. Deer "behavioral patterns are difficult to modify [after establishment]" (DeNicola).

Several deer deterring devices that are able to be easily adaptable into urban, suburban, and rural yards have been researched below:

- Ultrasound Emitters (e.g. YardGard Ultrasound Device): Ineffective, as deer habituate very quickly because they do not perceive the stimuli as a true danger. (Belant).

Figure 1. Yard Gaurd motion sensing speaker system with strobe light.

- Flashing Strobes: Ineffective. When added to other devices, no change in deer invasions was noted. (Belant).
- Modulating Sirens, (e.g. Electronic Guards): Ineffective, due to quick habituation. (Belant).
- Motion Activated Sprinklers (e.g. ScareCrow): Moderate effectiveness. Current systems do initially scare deer; however, have not been proven to continually keep them away. Deer seem to habituate to these systems and will ignore the stimulus especially when food is scarce. Motion sprinklers seem to be one of the most popular and effective solutions out on the market.

Figure 2. ScareCrow brand motion sensing sprinkler showing sensitivity adjustment

- Shock Sticks, (e.g. Havahart Electronic Deer Repellent): Low Effectiveness. Shock sticks lure deer towards the device with scent and then apply a shock to deter deer. Deer will often learn to avoid the shock sticks while still pursuing other plants in the area.

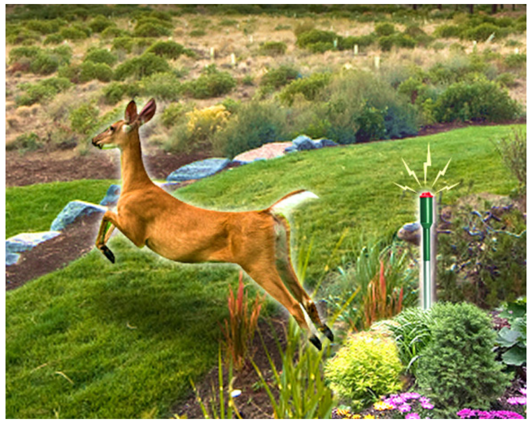


Figure 3. Illustration of Havahart Electronic Deer Repellent deterring deer with static electric shock.

Patents

There are a variety of patents for deer repelling tactics. The majority of patents are recipes for various deer repellents, which are discussed under existing products above. Many other patents have been issued for shock stick devices, which are also discussed above. Below is a summary of relevant patents found.

Table 3. List of some relevant patents.

Title	Patent Number	Details
Electrode design for deer repellent device	US 20130008390 A1	A device consisting of bait, mount, power supply, and an electrode.
Weather protected deer and animal repellent container	CA2538581 C	A container that protects and distributes animal repellent.
Deer repellent composition, process for making same, and a treatment method for applying the same	CA2134058 C	3-7% eggs 6-19% cayenne pepper 3-7% carrier, 68-91 % diluent
Deer repellent formulation	CA2259799 A1	Coyote urine containing 0.2% C8 fatty acid.
Deer Repellent	US6641830 B1	60.87% milk 30.43% deshelled eggs 4.35% corn oil 4.39% of 29% aqueous sodium lauryl sulfate.

JumpSport Inc. has filed a patent regarding a pest deterrent system that consists of several modules. These modules emit a variety of sounds or deterrents, depending on the intruder, in a manner that confuses the pest into thinking the source of the deterrent is moving toward itself. This patent may be used in the project solution subject to approval from JumpSport Inc.

Summary

Motion Activated Sprinklers and Deer Repellent have been the most positively reviewed existing products, yet they are still not proven to be effective in various climates and situations. A combination of deterring methods will likely yield the best results. Targeting the most sensitive of deer senses, hearing and smell, would also maximize the effectiveness of the solution. Applying positive punishment and negative reinforcement should maximized behavioral change. Deer must learn to associate the given stimulus with danger in order for the frightening technique to perform well. Motion activated devices tend to reduce habituation greatly when compared to devices that emit stimuli at fixed or dynamic intervals. In order to continually protect a given area, the system must be able to detect and deter an animal at all points. This will ensure that deer do not habituate to the system and learn to treat certain parts of the yard as ‘safe’.

To minimize intrusions, deer repelling systems should be active at all times, even though most of deer activity is at dawn and dusk. Deer activity is variable depending on many unpredictable factors such as temperature, weather, food scarcity, and deer population density. Deterrent system should continue to work as low as 32 degrees Fahrenheit and up to 5 inches of snow depth. Deer activity in these conditions reduces dramatically as discussed above in Activity.

Few patents exist that inhibit any particular solution other than repellent recipes. JumpSport Inc.’s filed patent may be used upon approval.

Ch 3. Design Development

To create a product or system which achieves these specifications most completely, we will implement a variation of the iterative design process. Specific dates and deadlines will be discussed in the Management Plan section later. Ideation, idea evaluation, and top concepts selection will be discussed.

Ideation

The team focused on creating as many ideas as possible in a creative and sociable environment. Referring evaluative judgment for a later time, hundreds of ideas were conceived and recorded using a variety of different ideating techniques such as brainwriting, SCAMPER, and others. These ideas were sorted under the four main functions of the project. No ideas were discarded, allowing as much idea building and open-mindedness as possible. Several rounds of ideation occurred, addressing each of the main functions multiple times, ensuring an exhaustion of ideas.

To explore further details of promising concepts, various concept models were constructed using foam board, hot glue, and wooden sticks to experiment with and visualize. These models helped introduce more concepts, which were further refined and recorded.

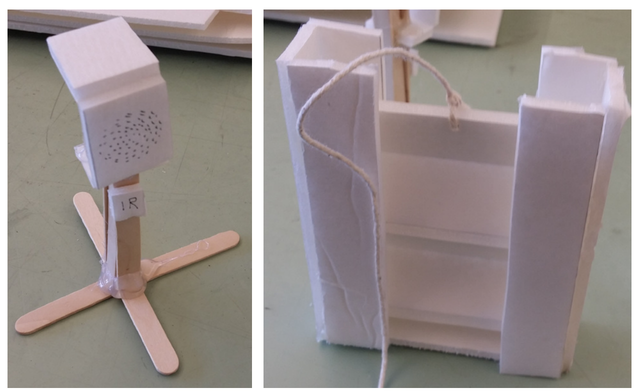


Figure 4. Foam core models built during ideation.

Top Concepts

Below are short explanations and basic conceptual layout drawings (if applicable) of a large variety of top concepts for the four project functions: Protection Methods, Communication of Maintenance Needs, Documentation, and Detection Methods. Concepts chosen to be included in the 'Top Concept' are bolded and descriptions of why each concept was approved or not approved are written below:

Protection Methods:

Air Cannon- Compressed air fired through a nozzle intended to hit deer as well as create a loud noise. This method would require a compressor and would have to target the location of the deer, requiring a highly complex system and a very short effective range. This method was not chosen because of its questionable potential to protect plants as well as it requiring compressed air to operate.

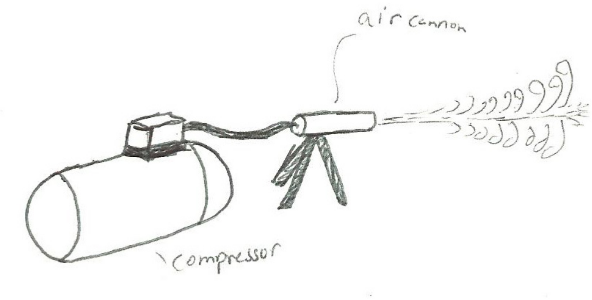


Figure 5. Compressed Air Cannon concept sketch.

Air Bazooka-Column of air projected with a mechanical mechanism. System requires a mechanism to pull back a diaphragm to project column of air. No air compressor is required. This method was not chosen because it requires a likely complex mechanical system while delivering questionable abilities to protect plants.

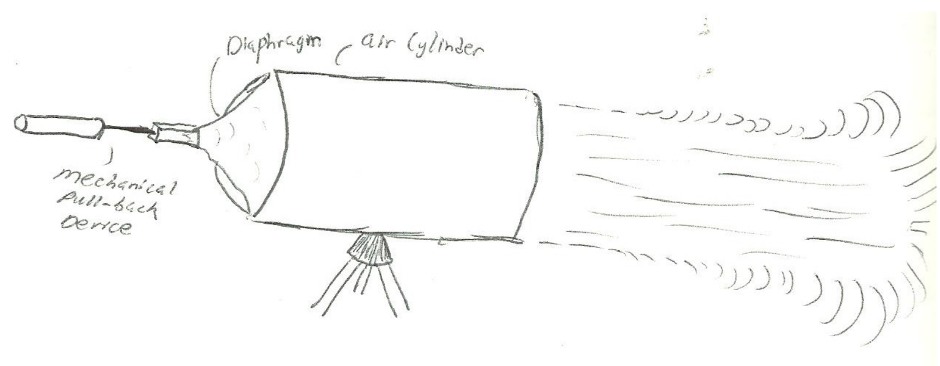


Figure 6. Air Bazooka concept sketch.

Water Wall-A ‘barrier’ protection method consisting of a tube placed around the perimeter of the area to be protected. When a deer invasion is sensed, pressurized water is sent through the tube. The water exits through holes or small nozzles throughout the length of the tube creating a ‘wall of water’. This method was not selected because of its lack of future marketability as well as its inability to deter deer within its perimeter which is vital to prevent deer from habituation to stimuli.

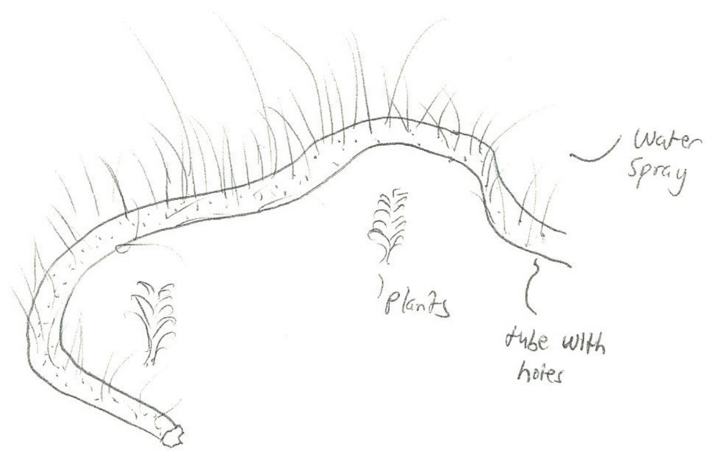


Figure 7. Water Wall concept sketch.

360° Full Area Spray-Similar to a standard yard sprinkler. Water is sprayed in a full 360° area. This system is simple, does not need a targeting system, covers a large area, but does not have a high scaring potential as deer have previously habituated to standard yard sprinklers. This system was not chosen because of its questionable abilities to protect plants.

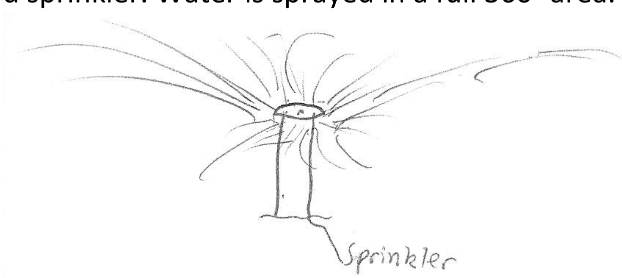


Figure 8. Full Area Spray concept sketch.

120° Area Stream-A motion activated sprinkler that focuses a heavy spray in a 120° cone. This device is currently used by some deer scaring motion activated sprinklers already on the market. This device was chosen as a part of the top concept because of its high potential of marketability, good abilities to protect large areas, as well as its simplicity.

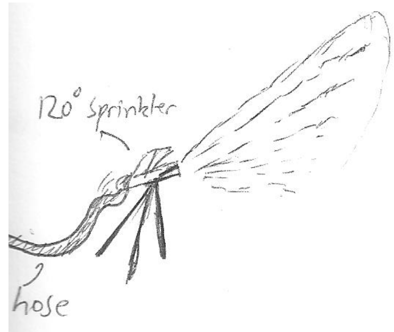


Figure 9. Directed area stream concept sketch.

Targeted Stream-Motion activated system that targets deer's location in yard and shoots water through a nozzle. This system would be highly complex and would need to hit deer directly. This system was not chosen due to it extremely high complexity.

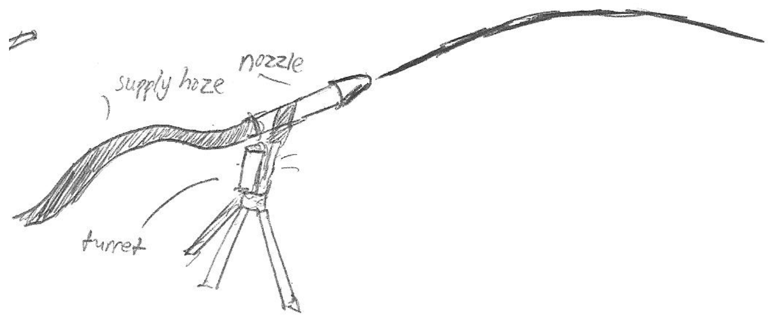


Figure 10. Targeted stream concept sketch.

Air-Crow-Uses a high powered fan to inflate and randomly move a large shape made out of a light material. This device is intended to provide simple yet random motion that is difficult for the deer to habituate to. This method was chosen as part of the top concept because of its high degree of random movement that will be key in preventing deer habituation.

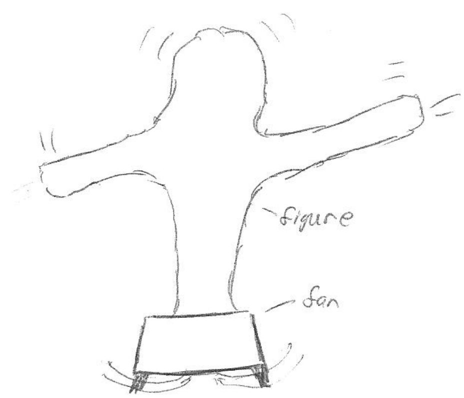


Figure 11. "Air-Crow" concept sketch.

Waving Flags-Motion activated mechanism to move flags in a particular pattern to scare away deer. This pattern would be a continuous motion that would continue until dear are deterred. This method was not chosen because of its simple and not random motion that has a low chance of permanently deterring deer.

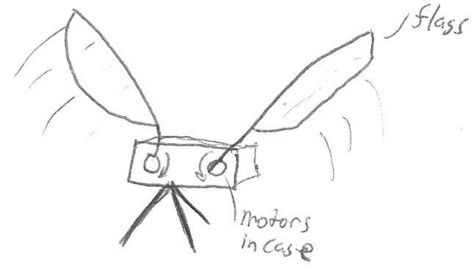


Figure 12. Waving flags concept sketch.

Pop-up Object-A sudden motion object designed to scare deer upon activation. This would need to be reset mechanically after each activation. This device was not chosen due to its complex nature as well as its inability to generate random movement.

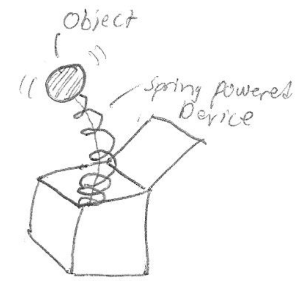


Figure 13. Pop-Up object concept sketch.

Continuous Movement Object- Object that would move in a continuous or cyclic motion once activated. The device would not cease until dear have been deterred. The motion of the object would likely create a noise to help deter deer. This device was not chosen due to its complex nature as well as its inability to generate random movement.

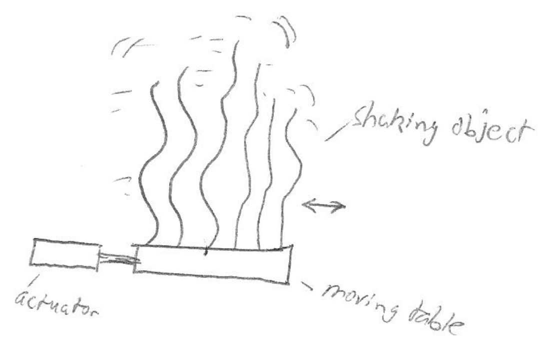


Figure 14. Continuous movement concept sketch.

Noisemakers- Uses mechanical devices to create a continuous and sudden burst of noise to deter deer. This method was not chosen due to its likely poor ability to deter deer.

Figure 15. Mechanical noise-maker concept sketch.

Stationary Speakers-Speakers would be placed in a fixed position and target deer in front of the system. The system would be able to test high and low frequencies and different volumes and types of sound to deter deer. This method was chosen because it will be the most dependable and low cost way to test sound as a deer deterrent.

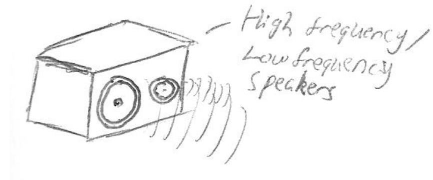


Figure 16. Stationary speaker concept sketch.

Targeted Speakers- Speakers would be rotated to face in the general direction of deer. This system is complex but allows sound to be specifically targeted toward deer, which decreases the amount of sound distributed to other areas outside of the protection zone. This method was not chosen due to its high complexity. The system complexity may interfere with habituation and may not provide accurate results for testing the effectiveness of sound as a deer deterrent. For example, the movement of the system may cause the deer to associate between it and the incoming sound, which may allow them to habituate.

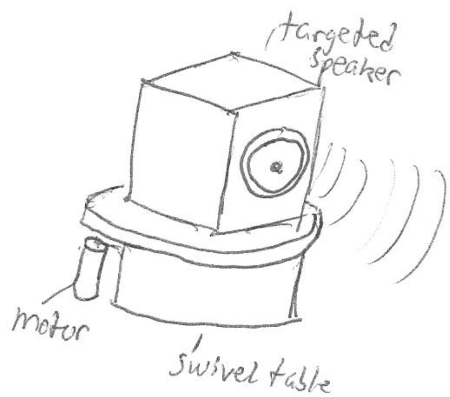


Figure 17. Targeted speakers concept sketch

Simulated Approach Speakers-Speakers are placed in a configuration that would create an illusion of a moving source of sound. This method was selected to be part of the top concept because of its uniqueness and high potential of deer deterrence.

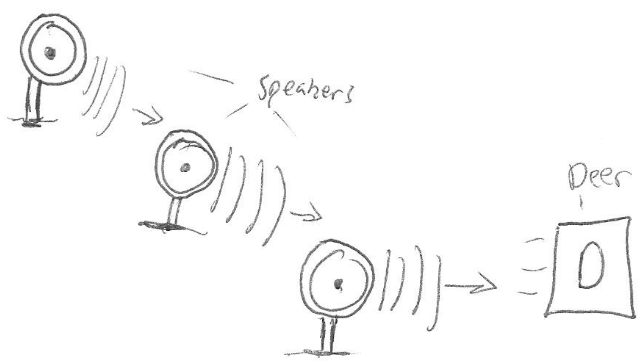


Figure 18. Simulated approach speaker concept sketch.

Simulated Approach Mechanical Devices- Mechanical noise-making devices are placed in a configuration that would create an illusion of a moving source of sound. This method was not selected because it is much more unnecessarily complex than using simulated approach speakers.

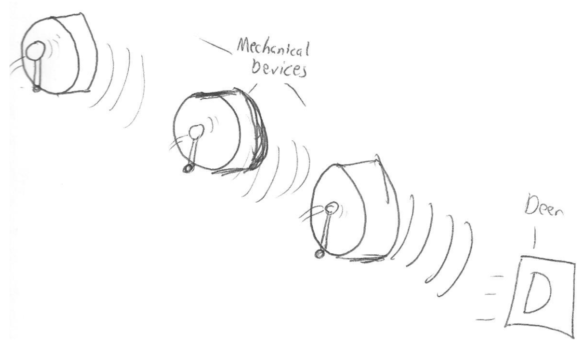


Figure 19. Simulated approach with mechanical devices concept sketch.

Actual Approach- A vehicle placed on a rail or track will rapidly approach the invading deer likely under electrical power. This method was selected because of its highly unique method of deterrence and is unlike any other concept. This method will ensure a more physical scaring method is incorporated into testing.

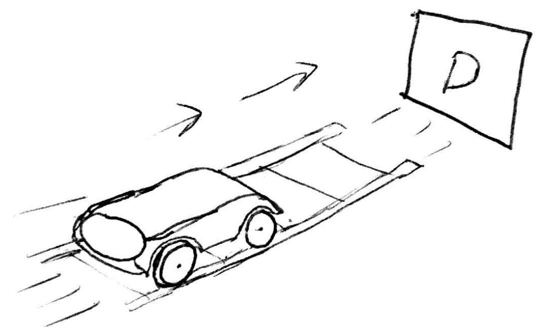


Figure 20. Actual Approach concept sketch.

Communication of Maintenance Needs:

Notify Directly through Electronics-System would send an email, text message, or other electronic report when the system is in need of maintenance. This system was chosen as part of the top concept since the system will be monitored through a PC based user interface.

Figure 21. Direct notification with electronics concept sketch.

Sounds-A beep or similar sound would be emitted from the system when maintenance was needed. This method was not selected due to its inability to be noticed indoors and potential obnoxiousness when outdoors.

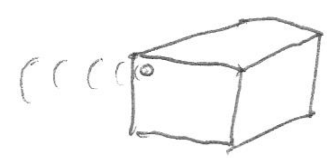


Figure 22. Electronic sounds concept sketch.

Light-A light would be emitted from the system when maintenance was needed. This method was selected to be part of the top concept. Since this method is simple and not obnoxious, it will be used in conjunction with the PC based user interface and electronic notification.

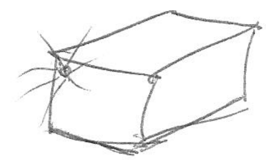
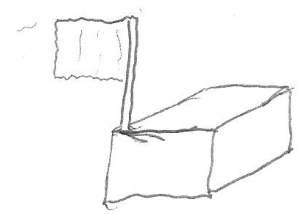


Figure 23. Emitted notification light concept sketch.

Backup Mechanism-If one system is unable to be triggered due to maintenance needs, a backup method would be used to deter deer. This would be used conjunction with another indicator of maintenance needs but would ensure that deer are deterred and a different method would be tested.



Physical Indicator (flag)-A physical indicator, likely a flag, would be deployed when the system was in need of maintenance. This system was not selected due to the relative simplicity of other methods.

Figure 24. Physical indicator concept sketch.

Documentation:

Motion Activated Camera-When movement is detected, camera will turn on and record area. This method was selected to be part of the top concept because it allows important data to be captured and stored while disregarding data in which nothing is occurring.

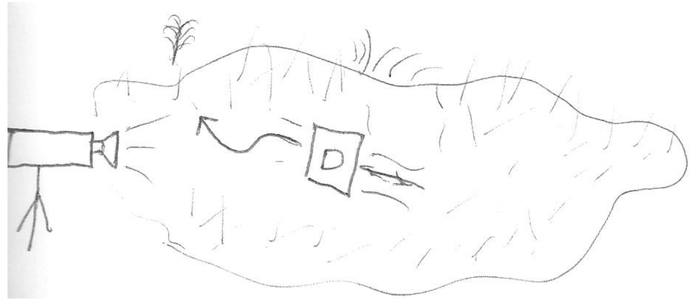


Figure 25. Motion activated camera concept sketch.

Trigger Counter-When device is triggered by motion sensors, the result is electronically counted and a report is generated. This does not give any data other than a count of how many times the system was triggered. This method was not selected because it does not offer enough data to the user.

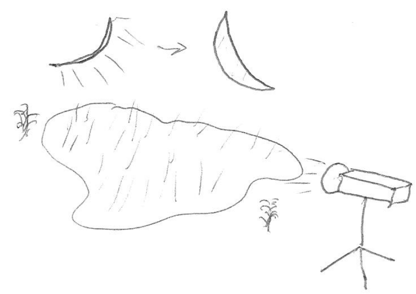


Figure 26. 24-Hour camera concept sketch.

24/7 Camera- A camera would continuously record the area. This would require a large data storage device. This method was not selected because it is impractical for the user to sift through the large amounts of data created.

Detection Methods:

IR Motion Sensor- Sensors that indicate when an object has moved through an area. These sensors are cheap, durable, weatherproof, and have a wide field of view, but only detect motion. This method was selected as part of the top concept because of its low cost, effectiveness, and high practicality.

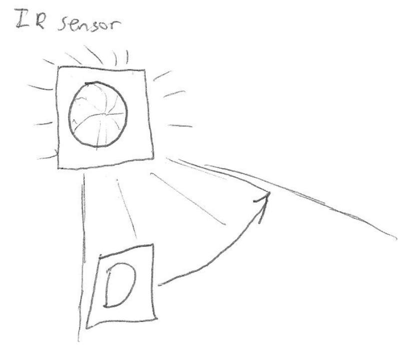


Figure 27. IR motion sensor concept sketch.

Heat Sensor-Of the top selected concepts, only the heat sensor would be able to discern the difference between a human and an animal by reading heat signatures given off by the moving object. This would be a highly complex operation and would be costly. This method was not selected due to its high complexity and cost.

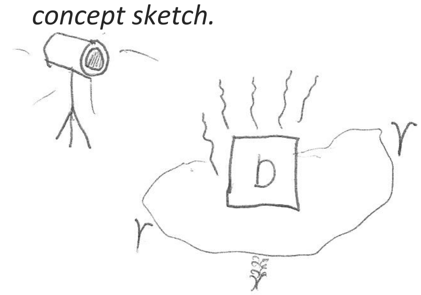


Figure 28. Heat sensor concept sketch.

Laser Rangefinder- Give range of target hit by laser. In order to scan a large area for motion, a scanning rangefinder would be required. These systems are extremely expensive and complex. This method was not selected due to its high complexity and cost.

Figure 29. Laser rangefinder concept sketch.

Sonic Rangefinder- Sensors use ultrasound to determine range of object. They are not as weatherproof as other sensors and have a limited range and field of view but are relatively inexpensive. This method was not selected because it is impractical to have a sensor with such a limited field of view.

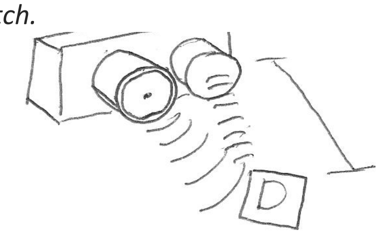


Figure 30. Sonic rangefinder concept sketch.

Summary of Concepts

Out of all of the concept ideas, the few selected to be a part of the final design were chosen from several runs through Pugh and weighted decision matrices comparing concepts to design specifications. The top concepts selected are as follows: 120° area spray sprinkler, the Air-Crow, stationary speakers, speakers set up for simulated approach, an actual approach vehicle, infrared motion sensors, notification through lights and electronic messages, and motion detection cameras. These concepts will likely offer reduced deer invasion over an adequate specified area, while documenting research data 99% of the time. When in need of maintenance the device will communicate with lights and electronic messages to alert testers. It will likely be unable to differentiate between humans and other animals, however will maintain high levels of safety to avoid potential injury to both humans and animals.

Overall our final system concept design met the most engineering specifications and customer requirements and seemed to be the most effective and efficient means to test deer deterrence methods. The selected concepts are the most cost effective and practical devices that can be combined into one cohesive testing machine.

Evaluation

After ideation was completed, the team focused on reducing or eliminating all impractical ideas that were clearly out of the scope of the project. All ideas were evaluated in a Go/No-Go list. All ideas receiving a "Go" were sorted into their respective project functions.

All four project functions were evaluated in their respective Pugh Matrices (see Appendix 3). In each Pugh Matrix, one idea was selected as a datum which to compare all other ideas to. For each project function, certain customer requirements were selected as the evaluation criteria. Every idea was then compared to its associated datum by determining whether it was better, worse, or the same at satisfying the different customer requirement criteria. Since the criteria of the Pugh Matrices were not weighted, all ideas scoring a "better" counted as +1 to their total score, -1 for "worse", and 0 for "same". All scores were tallied and added. All low-scoring concepts were reviewed by the team and then eliminated if deemed an implausible solution. To maintain fair balance in the Pugh Matrices, the ideas selected as a datum were not eliminated and continued into the next round of evaluation.

Project functions still containing a large amount of ideas were evaluated again. To further narrow the selection, all ideas passing the first set of Pugh Matrices were assigned into a second set, using the same customer requirement criterion and scoring system. A different concept was selected as a datum to ensure the validity of the evaluations. Again, low scoring concepts were reviewed and eliminated if deemed implausible, and ideas selected as a datum were not eliminated. The remaining concepts were not eliminated.

After two rounds of Pugh Matrices, enough ideas were eliminated to continue into weighted decision matrices. Although similar to Pugh Matrices, weighted decision matrices assign different weighted values to the different customer requirement criteria. These weighted values, based on determined importance of the criteria, are all assigned a certain percentage (out of 100%) of importance. All of the weighted values of the criteria must add up to 100% in order to ensure that realistic values are assigned to all criteria.

Weighted decision matrices (see Appendix 3) were generated for all four separate project functions. Customer requirement criteria were updated and assigned specific weights, highlighting the more important criteria. The team evaluated each concept's ability to every weighted criterion by assigning a value from 1 (representing a complete inability of the concept to meet the specific criteria) to 10 (representing a strong ability of the concept to meet the specific criteria). These scores were multiplied by the weight of their assigned criterion and then all multiplied scores belonging to an individual concept were added together to form a

composite score. From these composite scores, all ideas reflected potential viability as solutions to the project. Low-scoring concepts were eliminated while high scoring concepts were considered for implication in the top concept.

To form the final top concept design, high value concepts, determined by high scores in the weighted matrices or by team evaluation, were discussed and considered in detail by the team. These top concepts were sketched and evaluated as described below. Additionally, functions that needed further evaluation or that contained a large number of concepts were evaluated with a final set of weighted decision matrices.

Testing and Validation

A certain ‘protection zone’ will be designated by JumpSport. The dimensions of this zone will be replicated on a flat area and the device will be set at the appropriate location within the zone.

Night vision, motion-activated camera(s) will be placed in the designated protection area and will record the entire protection area. When a moving object is sensed by the system, the cameras must start recording a number of seconds before the scaring device activates. The data must be able to be saved for later access.

The device must trigger the appropriate response or responses in the direction of the test subject. The test subject will perform a number of tests to validate the system. The test subject will walk, run, and crawl 5ft., 15ft., and 24ft. from the device. The system must also be able to switch between different scaring mechanisms by user input during testing. If the system is able to perform all of these tasks, it is validated.

Top Concept Details

The design consists of several individual components which will be developed as individual solutions that can be operated in concert with other components or as the sole component active. They will be constructed to be modular so that the users will have more flexibility when positioning the deterrent methods. This may allow for additional testing of the location and orientation of the deer prevention methods.

Control Unit

The system will be a central control unit. It will act as the nerve center of the system by supplying each of the other components with instructions and power. The control unit will house the controlled by processor which will interpret sensing information and user inputs to determine the proper course of action. This unit will be the point of connection for the user, the sensors and the other components.

Figure 31. Control unit concept model.

Sensing Unit

The sensing unit will simply be a stand for the passive infrared sensor. It will lift the sensor above the ground one to five feet to avoid any ground activity or obstacles. It will be enabled by the control unit and communicate directly with it notifying of the presence of motion. Additional sensor may be found to be necessary and will be added to this unit as appropriate. Some possible additions may be a temperature, light and/or proximity sensors.

Stationary Speaker

The stationary speaker will consist of a single speaker that will be enabled to project a number of predefined sounds at the instruction of the control unit. The user will be able to specify the sound to play or allow the controller to randomly determine from a selection. This will allow the user to test the effectiveness of individual sounds as well as the effectiveness of variation to prevent habituation. The speaker will be mounted on a stand

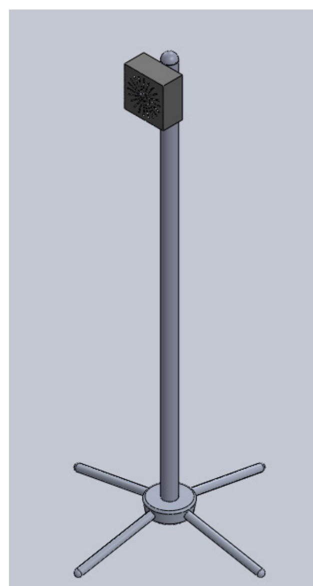


Figure 32. Stationary speaker concept model.

which will raise it to the level of the deer, likely causing the deer to consider the sound to originate from a large, dangerous animal.

Sprinkler

The sprinkler component will consist of an off-the-shelf impact sprinkler with an electrically controlled valve to control on and off functionality. The valve will be instructed to be either on or off by the control unit. The range and angle of spray must be predetermined by the user and set manually. This sprinkler unit will act as a sort of control condition for the tests as it will allow the user to compare the results of other components to the most effective method currently available. The sprinkler head will be raised one to two feet off of the ground to help increase the range of spray slightly and overcome any small obstacles which may obstruct the water stream path.

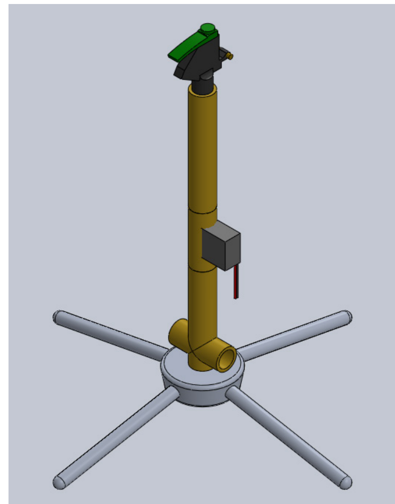


Figure 33. Sprinkler system concept model.

"Air-Crow"

The "Air-Crow" unit will test the presence of a humanoid analog by affixing a hollow, cylindrical cloth tube to an axial fan. The fan will be activated by the control unit which will blow a column of air up the tube causing it to stand near-upright and move sporadically, presenting the deer with a potential threat. A similar technique is often employed to attract attention to businesses, sometimes referred to as "Airdancers." The figure will stand five to ten feet tall in order to present an imposing figure. It may be necessary to vary the fan speed drastically to achieve the sporadic motion desired. This will be determined further in the development of the design.

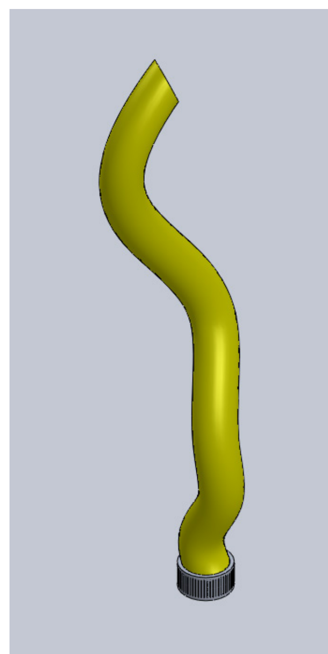


Figure 34. "Air-crow" concept model.

Simulated Approach

In order to test the method of simulating a threatening approach, an array of speakers, likely four, will be set up in a linear series. The speakers will be activated by the control unit and will project a pre-defined sound through the speaker at the end of the row. Then the speakers will successively take up the sound so that the source of the sound will move closer to the animal. This will be done in such a way to simulate the approach of a threat. The speaker array will be arranged in a line with each speaker held the appropriate distance away from the next one. It is critical to fix this spacing because the timing of the speakers will be calibrated to this distance and rearranging them may cause the system to not function properly. However, the set of speakers will be moveable so that numerous positions and directions can be tested. The speakers will be raised four to six feet off of the ground, not necessarily at a consistent height.

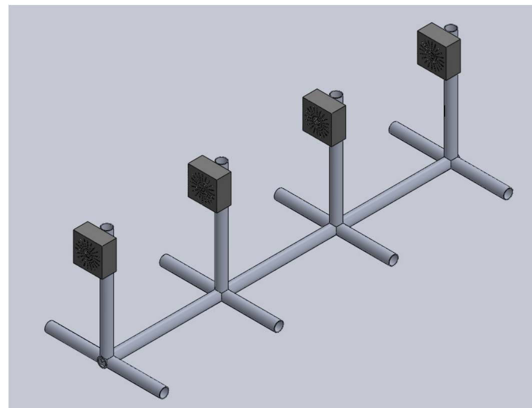


Figure 35. Simulated approach concept model.

Actual Approach

The actual approach unit will feature a vehicle on a guide rail. When activated by the control unit, the vehicle will travel along the rail, towards the offending deer. It will be fitted with a lightweight but large body so that the deer do not dismiss the thread as minor. The mode of motion has not yet been decided but could be a belt-drive, spring-loaded or piston type systems.

Construction

Construction of the testing units will be undertaken in the manner appropriate for each unit. The control unit will likely consist of a large electronics box which will ensure that the electrical components remain protected. Each electronic component will be fastened to the support structure and connected with solder in most cases. While the structures of each component will vary by function, the general support structure will likely be consistent; being comprised of aluminum or steel tubing bent and welded. Many of the components are lightweight and require only to be lifted to the appropriate level. The connections between components will be made with a standardized terminal plug so that several cords could be purchased or manufactured and utilized on any component. These connections, and all

components, must be completely waterproof as the device will be outdoors for long periods of time.

Sub-System Changes

Throughout the course of the design process, Kevin Charles of JumpSport expressed interest in the development of a system which could determine the approximate location of the deer in the yard, rather than simply that there is a deer in the yard. This suggestion was developed into the Rangefinder sub-system discussed later in this section.

In order to accommodate this new request with the limited time and budget of this project, we elected to stop pursuing the least practical solution proposed. After thorough discussion and deep considerations, it was determined that the "Air-Crow" sub-system would be eliminated from the project due to questions regarding the practicality and concerns over the weatherproofing the fan required for the design. This decision was approved by Mr. Charles of JumpSport on April 7, 2014.

Complete System Proposal

Following extensive development, the previously discussed designs were developed into a complete design. This section will detail the detail of these designs thoroughly and completely. Due to the modular nature of our design, we will discuss each of the sub-systems individually and then summarize at the end.

The following system was proposed to the sponsor for approval but was rejected, thus this design does not accurately represent the finished product pursued.

Actual Approach Deterrence

Overview

The Actual Approach system will consist of a motorized car that moves between to fixed points. The car will feature an onboard motor that will be externally powered by the main control unit. The car will not have any steering mechanism and will instead rely on a taut polyester rope running through its frame to guide it between the fixed points. Because of this, the motion of the car will be strictly linear. Once the car receives an input to go from the main controller, it will accelerate as quickly as possible until it reaches the end of its path. At the end, it will contact a switch which will turn off the motor and then move the car slowly backwards until it reaches its start location. At the start location, the car will contact another switch the will stop

the car until further input from the main controller. An overview of the system is shown in Figure 36.

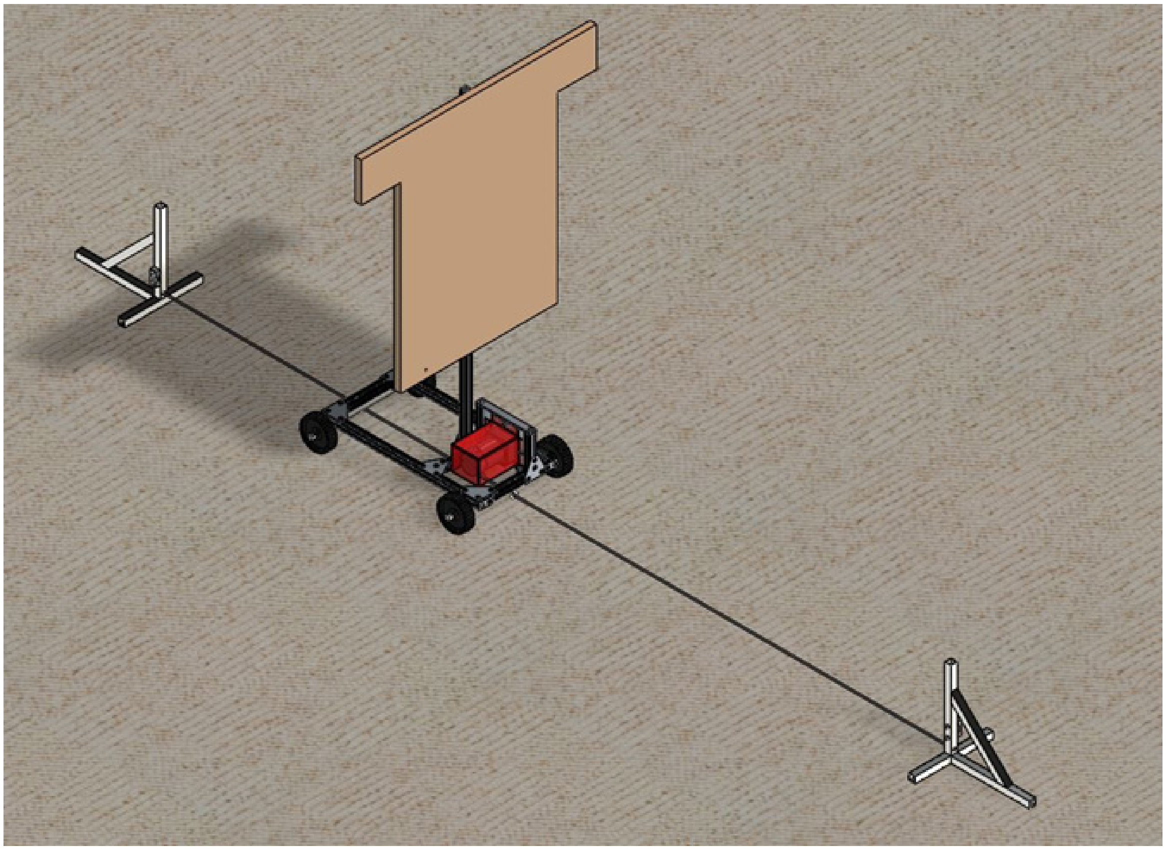


Figure 36. System overview of Actual Approach deterrence method.

Frame

The car frame will be made out of one inch aluminum T-slotted extrusions. This material gives both stability and rigidity while making the design adjustable for potential future additions. Additionally, the aluminum will offer good weather and corrosion resistance. The frame footprint will cover a two foot long by one foot wide area. It will feature extrusions used to mount the motor assembly as well as a mounting tower; a four foot tall extrusion attached in the center of the frame. The mounting tower is used to attach additional objects near the head height of deer to aid deterrence. Because the tower is much taller than the rest of the vehicle, calculations were made to ensure that the car would not be tipped over due to strong winds. Calculations only addressed the car tipping over laterally, as the rope constrains the vehicle from tipping about any axis other than the rope itself. Assuming an effective area of .5ft^2 located at the top of the tower, the car was calculated to withstand up to a 50 mph gust. This

easily surpassed the set requirement of withstanding at least a 35 mph wind (see Appendix 6 for calculation details). The framing will be bolted together by custom manufactured aluminum brackets (see Appendix 5 for manufacturing drawings) and purchased special T-slotted extrusion fasteners. Refer to appendix 5 for frame assembly and bracket attachment. See Figure 37 for an overview of the frame. Two single eyebolts are mounted to the bottom of the front and back frame pieces (the front frame eyebolt is shown in figure 38). The car will be guided along its predetermined path by a taut polyester rope passing through these eyebolts. Additionally, foam padding will be placed on the front and back sides of the frame to reduce impact force and improve the safety of the system.

Figure 37. Isometric view of Actual Approach assembled frame showing mounting brackets.

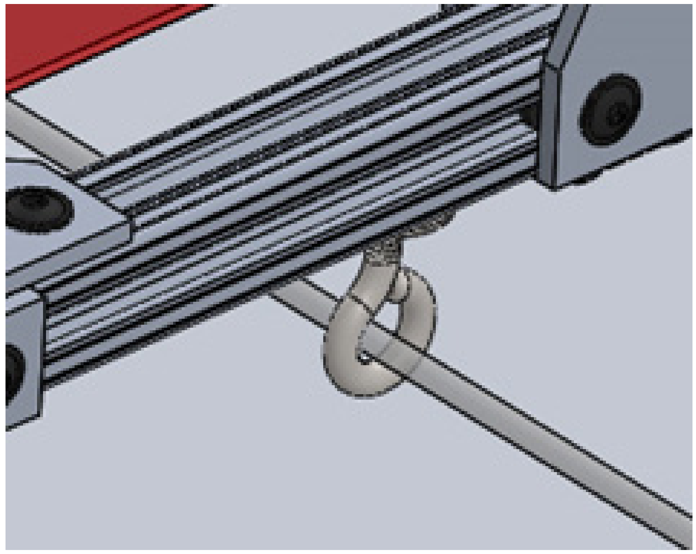


Figure 38. Guide rope interface for Actual Approach.

Motor

Certain performance parameters were created to help guide the selection process of potential motor candidates. Primarily, the car needed to travel the required 30ft. distance in roughly 3.5 seconds. This parameter ensured that the car would be able to mimic a serious physical threat to invading deer. Additionally, the car needed to accelerate quickly enough to maintain the threatening image, so the car was required to reach its top speed within about half of the distance traveled or 1.5 seconds. With these parameters, the car was modeled as a mass on an ideal (frictionless) surface. Kinematics and Newton's second law were used in order to determine both rotational speed (rpm) and torque (oz-in). A correction factor of 1.2 was applied to the torque calculation to account for losses. A summary of the calculation results are shown below in Table 4; refer to Appendix 6 for the full calculation.

Table 4. Summary of motor requirement calculations and selected motor specifications.

	Calculated Requirements	Selected Motor Specifications
Rated Motor Speed	500 RPM	480 RPM
Rated Motor Torque	206 oz-in	194 oz-in

The motor selected for the actual approach system is the BDSG-71-110-24V-3000-R5 Brushed DC Gearmotor from Anaheim Automation. This motor does not actually achieve the calculated specifications but it was selected because it was able to provide rated torque and speed to

within 10% of calculated specifications while having minimal impact on the project budget relative to other options. A list of important motor specifications is shown below in Table 5.

Table 5. List of specifications for BDSG-71-110-24V-R5 Brushed DC Gearmotor from Anaheim Automation.

Rated Torque	Rated Speed	Nominal Voltage	Rated Current	Weight	Price
(oz-in)	(RPM)	(V)	(A)	(lbs)	
194	480	24	3.5	3.48	$174.00

Motor Assembly

The motor assembly will consist of three parts: the acrylic motor case, a motor mounting plate, and the motor as seen in Figure 39. See Appendix 5 for assembly and installation drawings.

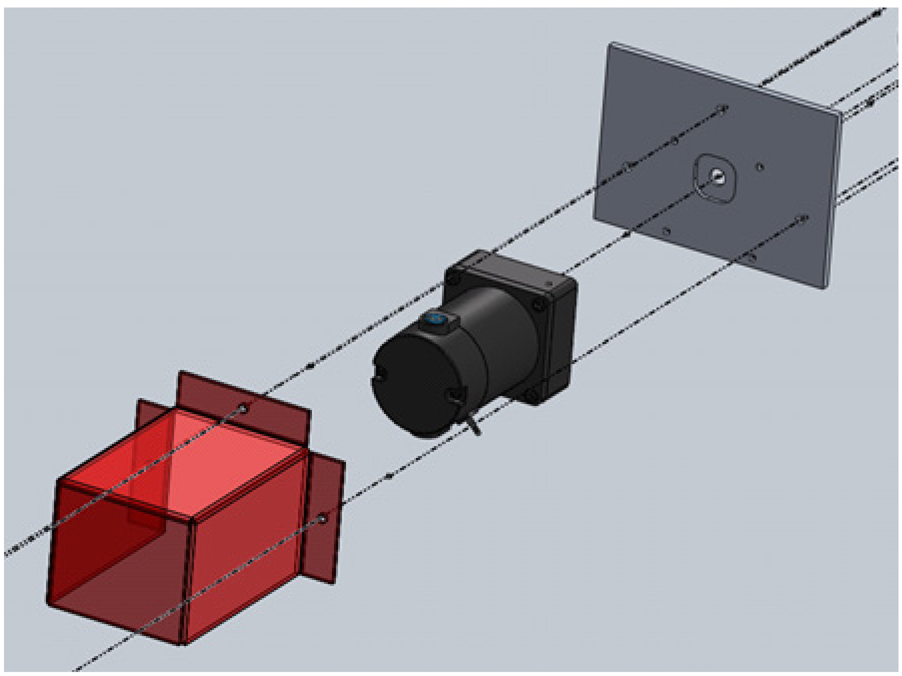


Figure 39. Exploded view of motor assembly.

The motor mounting plate is designed to secure the motor in place by mounting directly to four secure holes in the motor face with machine screws. The plate will be custom machined to match the contours on the face of the motor and precisely locate the mounting holes. (see Appendix 5 for manufacturing drawings). It is made out of a quarter-inch thick aluminum plate, which will be corrosion resistant and sturdy.

The acrylic motor case is designed to protect the motor from water damage due to rain. It will be custom fabricated from a flat piece of acrylic (see Appendix 5 for manufacturing drawings). The motor covers the top, back and two sides of the motor. The bottom of the motor will remain exposed; however this will allow the motor to prevent overheating. The motor case will be attached to the aluminum mounting plate via three machine screws. Any gaps from fabrication or assembly to the mounting plate will be sealed with caulk. The case is made out of UV-resistant acrylic which is a cost effective and weather resistant solution for protecting the motor.

The motor assembly will be installed onto the car frame by inserting the mounting plate into the slots of the aluminum extrusion framing as seen in figure 40. See Appendix Once the plate is secure between the extrusions; a specially fabricated plate (see Motor Top Plate manufacturing drawings in Appendix 5 for details) will be placed over the mounting plate and secured to the mounting extrusions with machine screws. The motor was placed at an equal distance between the two sides of the car in order to help maintain the center of gravity in the middle of the vehicle. See Appendix 5 for details on how the motor is installed onto the car.

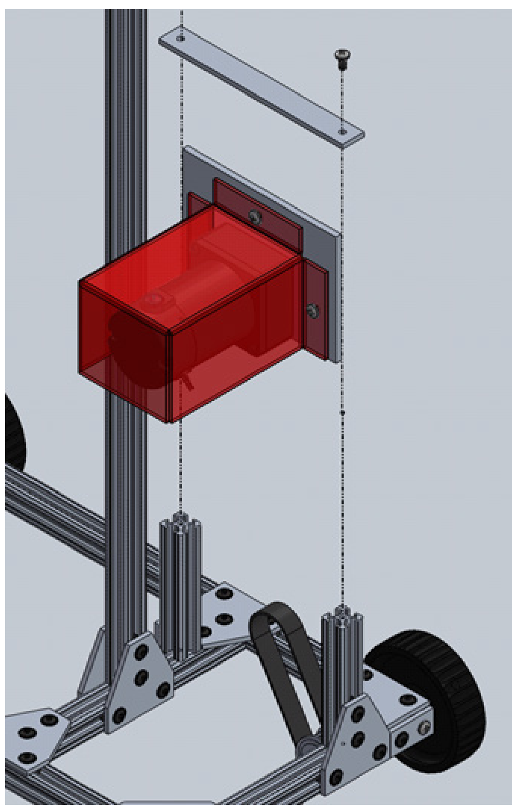


Figure 40. Exploded view showing installation method of motor assembly onto frame.

Power wires will be run from the main control unit to the motor of the car. To ensure that the car does not roll over the wires, they will be routed to the side of the car, perpendicular to the travel path. The wires will be tethered, elevated, and run directly to the motor assembly so that they will never come in contact with the wheels.

Power Transmission

The power from the motor will be transferred to the front shaft of the car through the use of a timing belt and pulley system. One timing pulley will be placed on the shaft of the motor. The selected pulley features a four millimeter keyway that matches with the motor shaft for torque transfer and will use a set screw to maintain its axial position. The timing pulley placed on the front shaft will maintain position and transfer torque through a set screw secured to a flat custom machined onto the front shaft (see Appendix 5 for manufacturing drawings). Both pulleys are made of black oxide coated steel which will be strong and weather resistant. See Figure 41 of the timing pulleys.

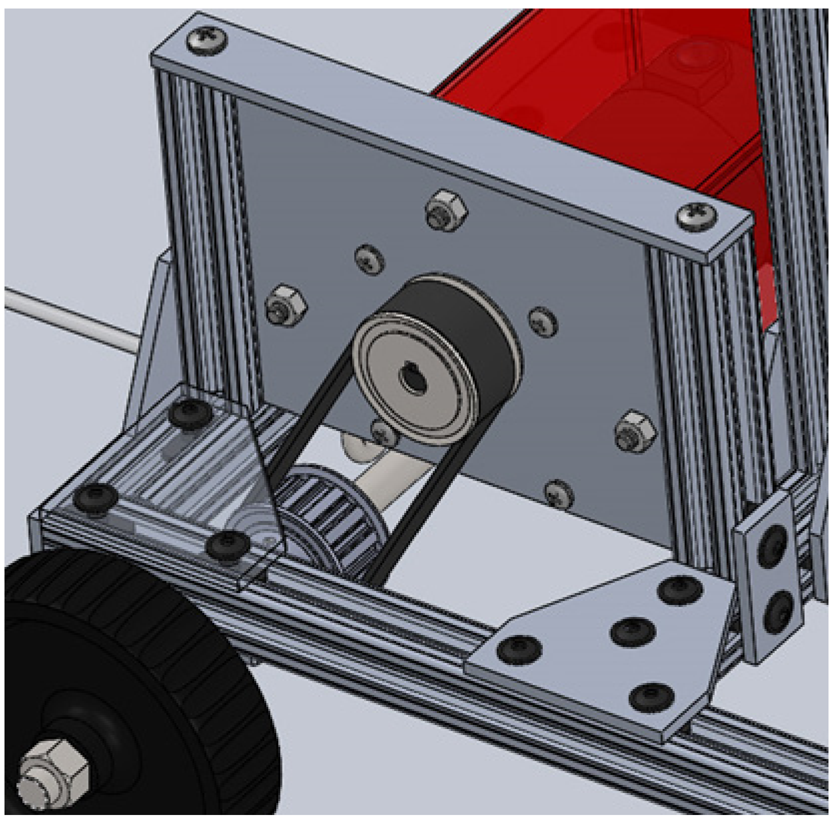


Figure 41. Model view of power transmission system for Actual Approach method.

Shafts and Wheels

The front (drive) shaft is supported by corrosion resistant pillow block bearings mounted just under the front of the frame. This shaft will be driven by the timing pulley system which in turn will drive the front wheels. Both ends of the shaft will be threaded two inches (see Appendix 5 for manufacturing drawings) and the wheels will be placed in between two jam nuts. The nuts will be tightened together against the wheel hubs, securing the wheels to the axle. Figure 42 shows an overview of the drive shaft. See appendix 5 for drive shaft assembly details.

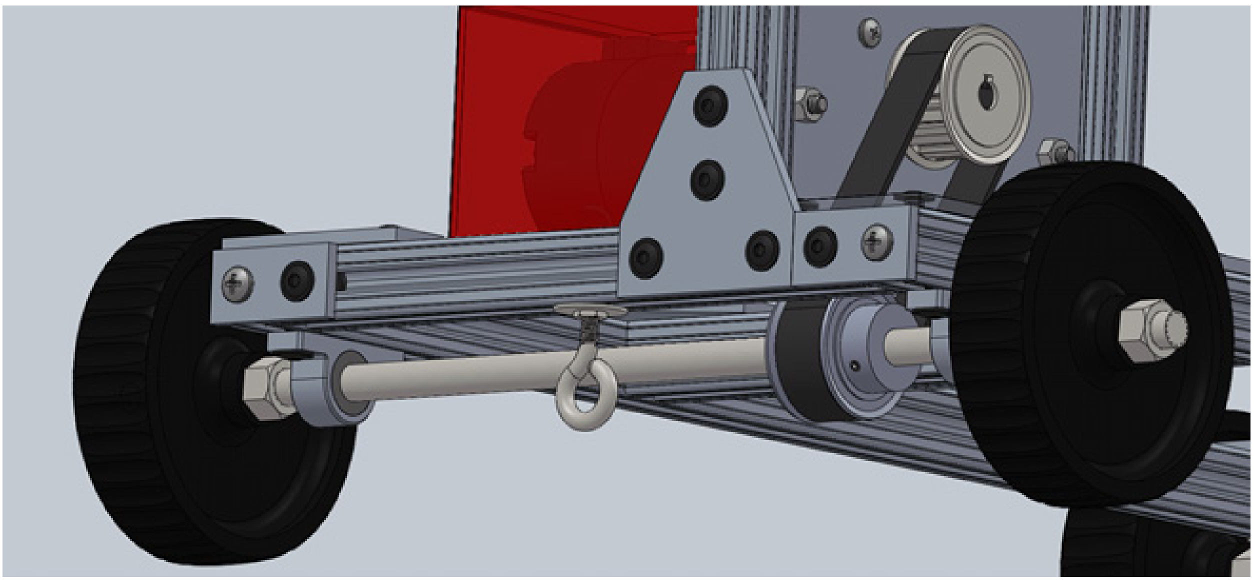


Figure 42. View of drive shaft showing power transmission and locking nut wheel attachment method.

The rear shaft will be completely stationary and will only serve to hold and align the rear wheels. The rear wheels will feature built in bearings that will have minimal rolling friction and will rotate on the rear shaft. These wheels will be held in place on the shaft with shaft collars secured with set screws to the rear shaft.

All of the wheels are made out of polypropylene which will be lightweight and weather resistant. The shafts are made out of general low carbon steel, but will be painted with corrosion resistant paint.

Shaft deflection calculations were performed to ensure that the front drive shaft displacement was not excessive enough to lead to alignment problems and potential shaft damage. After defining a .015in deflection limit, the highest deflection along the length of the shaft was .0001625in. at the left wheel. This displacement is well within the .015in. displacement limit. See Appendix 6 for details on calculation. Stress and fatigue analysis were not performed on the

front drive shaft due to low loading conditions found during the deflection analysis. Furthermore, the system is expected to have a low duty cycle, operating at an estimated frequency of 3 operations per day. Each operation is expected to have 56 shaft cycles, meaning roughly 168 shaft cycles per day, and 61,320 shaft cycles per year. Since loading is low, a shaft life of at least one million cycles is expected. Thus the shaft is expected to last 16 years of daily operation before failing from fatigue. Therefore, fatigue failure analysis was considered not necessary.

Car Stops

Car stops (see Figure 43) will be placed at the fixed points at each of the ends of the actual approach path. These stops will be fabricated (see Appendix 5 for manufacturing drawings) using sturdy rectangular steel tubing. They will be staked into the ground and will secure the polyester guide rope. Additionally, they will feature weatherproof limit switches that will send signals to the main control unit to cut power to the motor when contacted by the car. The limit switches will be attached to the stops by a custom machined polyethylene bracket (see Appendix 5 for manufacturing drawings). The switches are recessed enough so that the body of the switch will never make contact with the car, preventing damage. Furthermore, due to its positioning relative to the guide rope and the car body, the switch is assured to be triggered when the car reaches the end of its path.

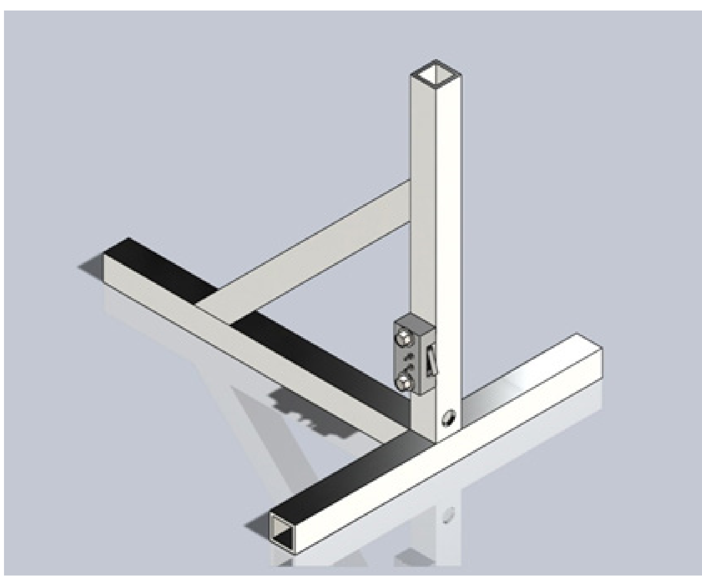


Figure 43. Model view of actual approach car stops with travel limiting switch attached.

Cost Analysis

Table 6 shows the bill of materials for the top level Actual Approach system. The actual approach system accounts for nearly half of the allotted project budget, however, this system is flexible and durable.

Table 6. Cost summary of Actual Approach deterrence system.

Item	Supplier	Unit Price	Quantity	Total Cost
2 ft. Aluminum Extrusion	McMaster	$8.35	2	$16.70
4 ft. Aluminum Extrusion	McMaster	$14.20	2	$28.40
12"x12" - 0.19"thick Aluminum Plate	OnlineMetals	$23.15	1	$23.15
3ftx1inx3/16in Aluminum Strip	McMaster	$5.07	1	$5.07
Aluminum Extrusion Fasteners (4 Pk.)	McMaster	$2.30	14	$32.20
1/2" SAE 840 Pillow Block	McMaster	$11.11	2	$22.22
1/2" Rear Plastic Insert Blocks	McMaster	$1.52	2	$3.04
Timing Belt	Misumi	$13.30	1	$13.30
Timing Pulley A (Motor)	Misumi	$25.13	1	$25.13
Timing Pulley B (Front Shaft)	Misumi	$25.64	1	$25.64
7"x5"x.25" Aluminum Plate (custom cut)	OnlineMetals	$7.70	1	$7.70
1/2"x18" 1018 Steel Rod (custom cut)	OnlineMetals	$5.58	2	$11.16
5" OD, 0.5" ID Rear Wheels Econolite	McMaster	$8.30	2	$16.60
1/4"-20x1"Eyebolts (10 Pk.)	McMaster	$2.99	1	$2.99
1/2" Shaft Collar Black Oxide	McMaster	$1.27	4	$5.08
Motor: BDSG-71-110-24V-3000-R5	AnaheimA	$174.00	1	$174.00
12"x12"x1/8" Acrylic Sheet	McMaster	$6.74	1	$6.74
7/32" Polyester Rope (100ft.)	McMaster	$9.00	1	$9.00
Monster Pool Noodle	Amazon	$10.00	1	$10.00
1"x1"x0.06" Sqr Stl Tube (8ft)	OnlineMetals	$12.18	1	$12.18
18" Concrete Form Board Stakes	McMaster	$3.50	6	$21.00
3/4"x1" Polyethylene Bar (1ft)	McMaster	$2.33	1	$2.33
Misc. Hardware		$5.00	1	$5.00
Total Hardware				**$478.63**

Please note that these figures do not consider shipping costs or relevant taxes. Additional cost may be applied by the supplier at the time of ordering.

The costs provided include only the cost of the physical components; specific electrical hardware is required to operate this system and will be discussed in the Control Sub-System section.

Maintenance and Repair

The actual approach system does not require any regular maintenance schedule. However, it is recommended that the actual approach system be inspected on a weekly basis to ensure that all components are working correctly and that the path of the car is clear of any sizable debris. The following checklist should be used to regularly check the system:

- The front wheels do not slip on the drive shaft
- The pulley belt is tight and rotates with both timing pulleys
- The timing pulleys are secure
- The rear wheels spin freely
- The eyebolts are secure under the frame
- There are no loose or missing fasteners on the frame
- The motor assembly is secure
- Check for any signs of corrosion

The actual approach system has been designed to be modular. There are no permanent fastenings on this system so that every component will be able to be replaced if necessary. Additionally most components are off the shelf so that they will be able to be replaced with minimal effort and cost. Special attention has been given to the material choices in this design to prevent corrosion since the system will be exposed to the elements over a long period of time. All materials that are not corrosion resistant will be painted with corrosion resistant paint in order to maximize the longevity of the system.

Safety Considerations

Since the actual approach system is a physical deterrence method, it presents the biggest safety hazard. To ensure that accidents do not occur, the path of the actual approach system will be clearly marked. Additionally, the system will be able to be turned off during potential high traffic times (such as during the day). Foam padding will also be placed on both the front and back of the unit to ensure that impact is mitigated. All electrical wires and connections will be clearly marked and insulated to minimize shock hazard.

Sprinkler Deterrence System/Motion Sensor

Overview

The sprinkler system will consist entirely of the Yard Enforcer motion activated sprinkler by Orbit. However, the electrical connection (visible in Figure 44) will be severed so that the signal from the motion sensor travels to the system control unit rather than directly to the solenoid. This signal will alert the whole system to the presence of an animal. Similarly, the solenoid will be triggered from the control system when appropriate, utilizing the same portion of wire which currently exists.

Figure 44. Orbit Yard Enforcer motion activated sprinkler system.

We chose to use a pre-engineered sprinkler system due to its sturdy corrosion resistant plastic body which is predesigned to be water resistant.

The range of the Passive Infrared Sensor (PIR) will be utilized directly from the Yard Enforcer system. It features a 120° field of view with manually adjustable sensitivity and range. It also allows the user to set the time of day which the sensor will be active: Day, Night, Off or Always. The sensor module, with settings, is shown in Figure 45, below.

Figure 45. Motion sensor with manual controls from Orbit Yard Enforcer.

Cost Analysis

As we are repurposing an existing system, the cost analysis is very simple. Simply purchase the product. However, the product was purchased previously during the research phase of this project and will be repurposed so that additional purchases are not required. Table 7 shows the bill of materials for this sub-system.

Table 7. Cost summary for Sprinkler and PIR sensor systems.

Item	Supplier	Unit Price	Quantity	Total Cost
Orbit Yard Enforcer	Amazon.com	$69.97	1	$69.97

Maintenance & Repair

The Sprinkler sub-system does not require any regular maintenance. The product will be modified to receive power from the main control unit and thus battery changes are not required. The system was designed to resist the elements and continue to operate but regular inspections are advised to determine that the sensor is still triggering as desired. This simple test could be conducted by simply placing a hand in front of the sensor momentarily while the system is active understanding that it may trigger a deterrence method response. It is also advisable to ensure that the supply of water is maintained.

Safety

The sub-system poses no significant safety threat to people or animals.

Test Results Documentation

Overview

The Bushnell X-8 Trail Camera was selected as our data verification sub-system. It has a 6 megapixel resolution, a day and night auto sensor, an adjustable passive infrared sensor, and programmable photo and video controls. It was the ideal system that satisfies our design parameters because it is a weatherproof camera that can capture both photos and video and it can be activated by movement with its own passive infrared sensor. The camera itself was on the lower end of the price range in the market and as seen below in Figure 46 has a nice brown painted case that helps to camouflage it from wildlife. The images and videos taken are stored on a removable SD card which can be removed to view test results. It comes with a set of straps to attach the camera to a nearby tree.

Figure 46. Image of Bushnell X-8 Trail Camera.

We elected to choose a camera system with an independent motion sensor to enable the detection of instances in which the main sensor fails to trigger. This will help the user place the sensor in the optimal positions for the specific yard conditions.

Cost Analysis

The camera is a pre-engineered product repurposed for our needs, thus the cost summary (provided below in Table 8) is limited. It is only required to simply purchase the item.

Table 8. Cost summary of test documentation method.

Item	Supplier	Unit Price	Quantity	Total Cost
Bushnell X-8 Trail Camera	Amazon.com	$117.20	1	$117.20

Please note that the cost provided does not include shipping costs or applicable taxes which may be imposed by the supplier at the time of purchase.

The camera is powered by 8 AA batteries and stores information on a removable SD storage card. These items may need to be purchased depending on local availability.

Maintenance & Repair

The camera will require regular maintenance to ensure it remains in working order. First the SD card must be removed periodically so that the data stored on it can be copied and erased. If the SD card fills, the camera will no longer record information. Finally, the cameras batteries must be replaced periodically. Bushnell claims up to 9 months of battery life but that will likely depend on the frequency of triggering.

Simulated Approach Deterrence Method

Overview

The Simulated Approach sub-system will consist of a four speaker array that will play sound in sequential order when triggered by the system control hub. Each speaker will be able to be independently placed throughout the yard, adding flexibility to the system. By moving the sound closer to the target, we hope to simulate an approaching predator. Figure 47 shows this array of speakers.

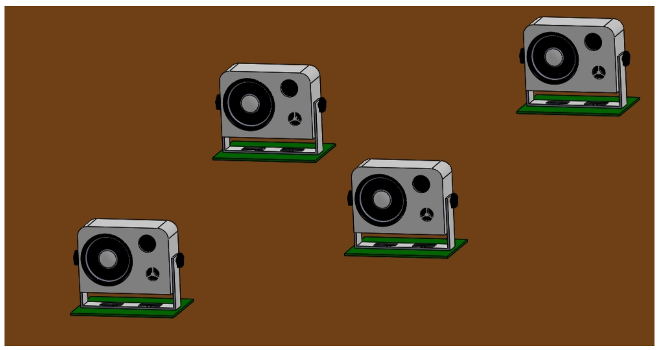


Figure 47. Simulated Approach layout of four independent speakers.

We chose to use the Pyle PLMR24 3.5" 200W 3-Way Weatherproof Mini-Box Speakers (shown in Figure 48). These speakers were chosen because they were the lowest cost solution which are designed to be fully waterproof. These 4 ohm, 100 Watt speakers are constructed of heavy-duty ABS plastic and feature three separate cones enabling frequency range.

Figure 48. Image of Pyle PLMR24 weatherproof speakers.

The speakers will bolted onto a custom machined ⅛" steel sheet with two ¼"-20 tapped holes (see Appendix 5 for manufacturing drawings). The purpose of this plate is to attach to the mounting bracket that comes on the speaker. It will ensure that the speakers always stay upright and can be easily moved into different test configurations desired by the end user. The key benefit of this mounting method is that it is low cost and extremely simple, while still providing stability to the part.

Cost Analysis

A summary of the cost of the project is provided in Table 9, below. The items listed include the speakers, the mounting plates, some camouflaging flowers and the speaker wire necessary to transmit the signal from the control unit to each individual speaker.

Table 9. Cost summary for Simulated Approach speaker deterrence system.

Item	Supplier	Unit Price	Quantity	Total Cost
1/8" Steel Plate	McMaster	$12.49	4	$49.96
5-Stem Artificial Flowers	Dollartree	$1.00	4	$4.00
Pyle PLMR24 Hydra Speaker	Staples	$17.49	4	$69.96
16 Gauge Outdoor Speaker Wire (150 ft)	Amazon	$43.99	1	$43.99
Total Hardware Cost				**$167.91**

Please note that the cost provided does not include shipping costs or applicable taxes which may be imposed by the supplier at the time of purchase.

The costs provided include only the cost of the physical components; specific electrical hardware is required to operate this system and will be discussed in the Control Sub-System section.

Maintenance & Repair

The Simulated Approach sub-system does not require any regular maintenance. All of the parts are powered from the main control unit and thus do not require changing batteries or any other regular interaction. All exposed metals come painted and fully weatherproofed to prevent corrosion ensuring a long life.

The wires may come loose from someone tripping over them or from being moved. In this case, we recommend checking the connections of each of them at least once every couple weeks or after every time they are moved to a new location. The likelihood of failure is low and is a relatively simple fix.

Safety

The simulated approach system is an audible deterrence method and thus presents minimal injury risks. The speakers will not be provided power significant enough to damage hearing or cause pain to humans or animals. However, if it is deemed to be too loud by the users, a manual volume control knob is available in the electronics control box and will be discussed later in this report.

Rangefinder Targeting System

Overview

The Rangefinder system will scan the environment and return the relative distance of the nearest “large” object for a range of directions from 0º to 180º. With this data, a device could determine the location of an animal relative to the sensor. The scope of this project includes only the collection of data and presentation to the user; any data analysis will be left to others.

The system will consist of an ultrasonic rangefinder mounted, via a custom connection bracket, to a standard hobby servo. The hobby servo will be mounted to a second bracket which is fixed to a post staked into the ground. This configuration is shown in Figure 49, below.

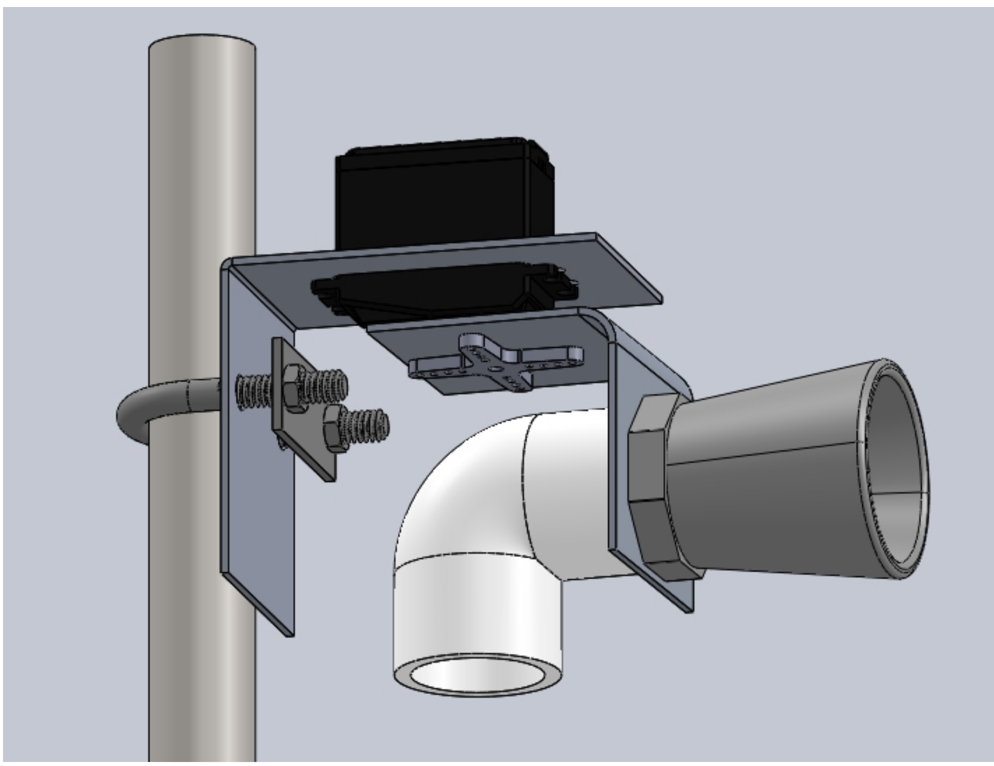


Figure 49. Model view of Rangefinder Targeting system assembly.

The sensor has been chosen to be an XL-MaxSonar-WRMA1 ultrasonic rangefinder from Maxbotix. The sensor is IP67 rated protected against short-term water immersion and thus is suitable for outdoor use. The sensor features real-time background calibration, real-time waveform signature analysis and noise rejection algorithms for very accurate readings. We expect these features to enable this sensor to provide reliable data even in light rain or wind. Additionally, the sensor is calibrated with narrow sensor beams to enable up to 25 feet of range. Detailed information regarding the ultrasonic sensor can be found in Appendix 7.

The sensor housing is fitted with a ¾" NPT pipe thread on the rear of the device so that dry electronic connections can be made. This feature is taken advantage for mounting; a thin piece of aluminum plate will be sandwiched between the threaded connections providing a secure attachment. A 90 degree bend fitting will be used so that the opening for the wires will point down, preventing rainwater from damaging the sensitive electronics. The aluminum plate will be bent such that it will interface with the servo motor as shown in Figure 49. The plate will have the necessary holes drilled for each interface and then be bent to the approximate angle manually.

The servo will be a simple, standard sized hobby servo consisting of a small brushed DC motor, a gear ratio of 180:1 and circuitry allowing for controlled angular positioning all within an inexpensive and simple package. It is important to note that hobby servos are not typically water-tight and will require sealing with silicone sealant. However, the rotational power transmission opening will not be sealed to allow for free rotation. This opening is why the servo will be positioned upside-down, preventing water from climbing up, past the gears, to the motor and electronics.

The servo will be mounted to a plate with the standard mounting screws included (not shown) to a second aluminum plate. This plate is bent at 90 degrees and permits the use of a u-clamp to attach to the post. Technical drawings detailing the dimensions and hole configurations of the plates are provided in Appendix 5. This interface enables the effective height of the sensor to be raised or lowered as seen fit by the user; up to 5 feet from the ground.

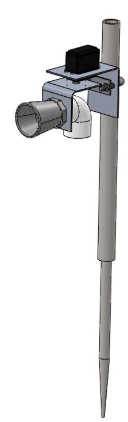


Figure 50. Model view of Rangefinder Targeting system showing pole and stake.

The post will consist of a standard, ½” trade size steel pipe (0.840” OD) with a steel stake partially inserted inside of it. The stake will consist of a plain rod, machined to a point. The pipe and stake will be welded together at the interface allowing the entire assembly to be stuck into the ground allowing for versatility and minimal cost. Figure 50 shows the full assembly.

Cost Analysis

The sub-system was designed with the goal of minimizing cost while not sacrificing quality. For this reason, an advanced sensor was chosen to provide quality data reliably. The remaining components were chosen, primarily with cost in mind as their functions are non-critical and often trivial. Table 10 details the components required and their associated costs.

Table 10. Cost summary for Ultrasonic Rangefinder Targeting system.

Item	Supplier	Unit Price	Quantity	Total Cost
Maxbotix XL-MaxSonar-WRMA1 Ultrasonic Rangefinder	Adafruit	$84.95	1	$84.95
Towerpro SG-5010 Servo w/ Hardware	Adafruit	$12.00	1	$12.00
Rangefinder Aluminum Mounting Plates (2)	McMaster	$1.96	1	$1.96
90deg threaded PVC Elbow	McMaster	$0.44	1	$0.44
½” Stand Pipe	McMaster	$20.52	1	$20.52
½” Diameter Ground Stake	McMaster	$3.06	1	$3.06
U-Bolt Clamp for ½” Pipe	McMaster	$1.02	1	$1.02
Silicone Servo Sealant	Amazon	$4.28	1	$4.28
Total Hardware				**$128.23**

Please note that the cost provided does not include shipping costs or applicable taxes which may be imposed by the supplier at the time of purchase.

Maintenance & Repair

The Rangefinder sub-system should not require any regular maintenance. All of the parts are powered from the main control unit and thus do not require changing batteries or any other regular interaction. All exposed metals are either galvanized (U-Bolt) or will be painted to prevent corrosion ensuring a long life of service.

The servo motor does rely on plastic gears which may become worn or broken due to extreme circumstances. In this case, we recommend purchasing a replacement motor and replacing the broken one. While this is not an ideal solution, the likelihood of failure is low and the component is relatively inexpensive, however the servo casing must be sealed prior to outdoor use to prevent rain damage.

Safety

The sub-system poses no significant safety threat as there are no exposed wires or high voltages and there is not sufficient torque produced by the servo. Though there are potential pinch points, we expect the motor to stall easily if its path is obstructed.

The sensor frequency is 42 kHz which, while outside of the range of human hearing, may be detectible by household animals such as dogs and cats (not deer). This sound will likely not be damaging to the animal but could be bothersome over long periods of time, so prolonged exposure to the emitted sound should be avoided for household pets.

Control System

Overview

The control system will serve as the main "thinking" hub of the device. It will coordinate the actions of all other sub-systems, except for the camera, based on inputs from the user and from the passive infrared sensor. Figure 51 shows a representation of the electrical connections of the components. The circuit for each sub-system will be discussed in more detail later. As mentioned before, the camera sub-system is completely independent from the main system and thus is not controlled.

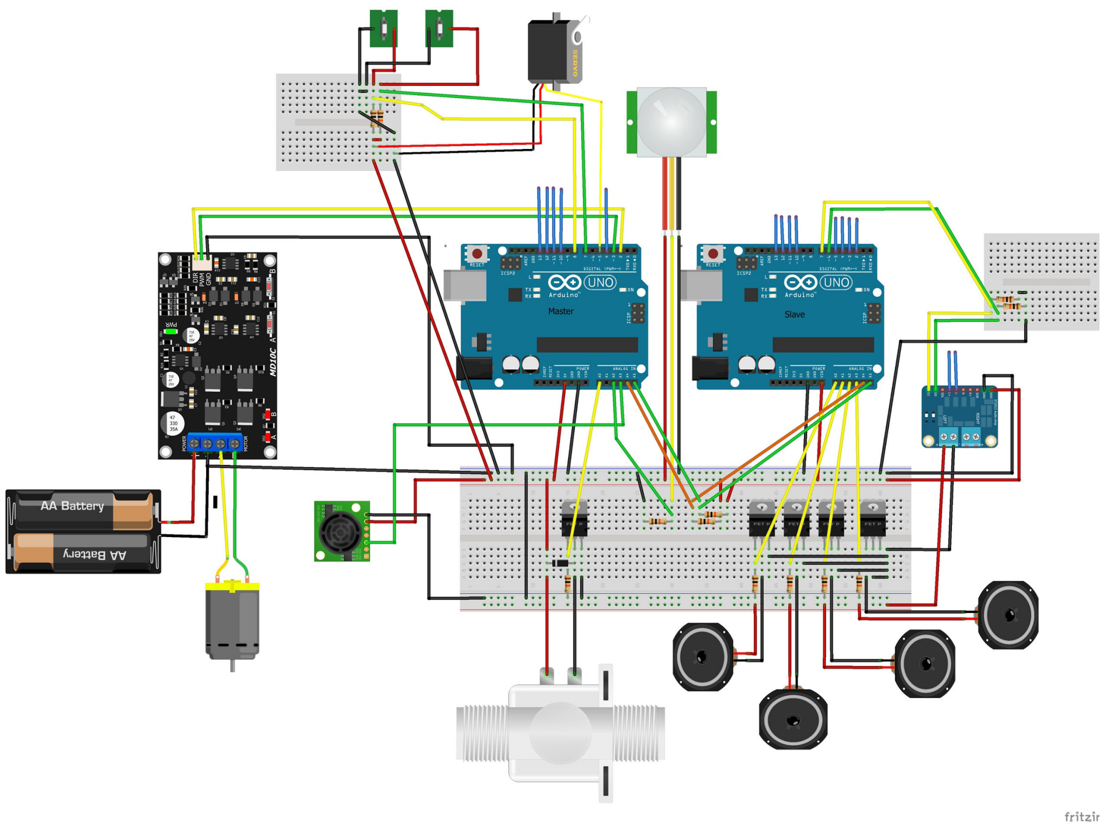


Figure 51. Representation of complete system electronic connections.

Microcontrollers

The system will consist of two Arduino Uno microcontroller boards communicating with each other via the I2C communication protocol. One of the microcontrollers will be defined as the master and the other as the slave. With this protocol the master program will issue instructions to the slave. The slave will carry out the instructions and notify the master that the tasks assigned have been completed. The slave will be constrained to carry out no tasks except on the master's request and only carry out those requested. This will prevent miscommunication or confusion between the two programs. Figure 52 shows the wiring diagram required for I2C communication between Arduinos. Note that pin A4 on the Master board is directly connected to pin A5 on the Slave board, and similarly with pin A5. It is also important that the boards be on the same circuit meaning that the slave must be powered by the output from the master and the ground must be common. Also note that the communication lines must be connected to the 5V connection with a resistor.

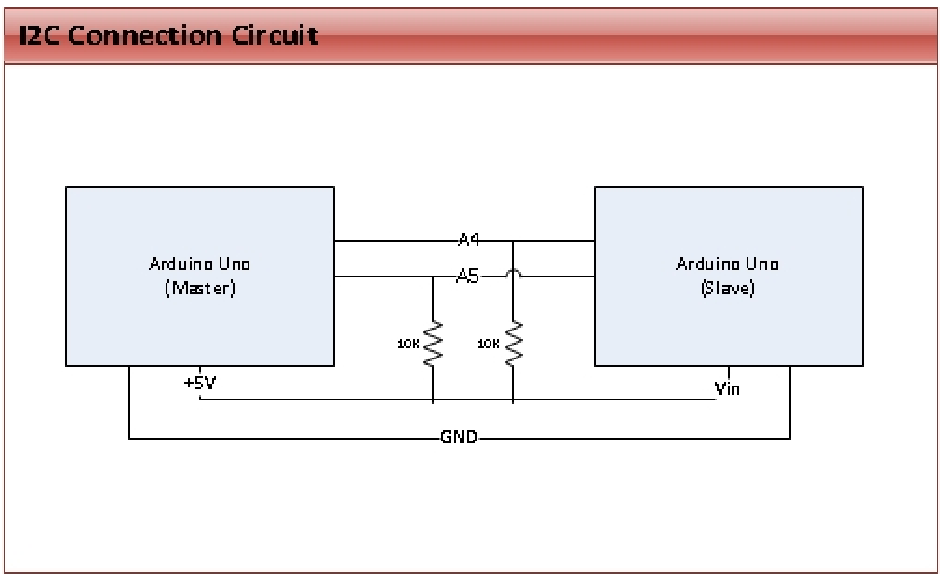


Figure 52. Circuit diagram required for I2C communication between Arduino boards.

Dual microcontrollers are required for two reasons. The first reason that the functions of the slave will require significant memory usage and risk "crowding out" other functions of the system, therefore these computationally intensive tasks will be isolated to the single microcontroller, allowing the other to manage the rest of the system. The second reason for two microcontrollers is that there are simply insufficient input/output pins available for all of the functions required, with specialized pins required multiple times. Additionally, it is generally

more simple to generate two smaller programs that a single large one. These reasons justify the use of two microcontrollers.

The Arduino Uno was chosen to serve as the nerve center of the system because it provides useful features at a relatively low cost. The Uno features a 16 MHz clock speed with 20 Input/Output (I/O) pins, 6 of which can be configured as pulse width modulation (PWM) outputs and 6 of which can be configured as analog inputs. The PWM functionality will be employed to control the motor and the analog input capability will be utilized by the ultrasonic rangefinder (discussed in a previous section). It requires only 7-12V input and is I2C and USB ready. Additionally, the Arduino is programmed in C, though extensive libraries are provided greatly simplifying software development. The Arduino Uno R3 is shown in Figure 53 and detailed specs sheets and circuit diagrams are provided for the board and the Atmega328 IC chip used in Appendix 7.

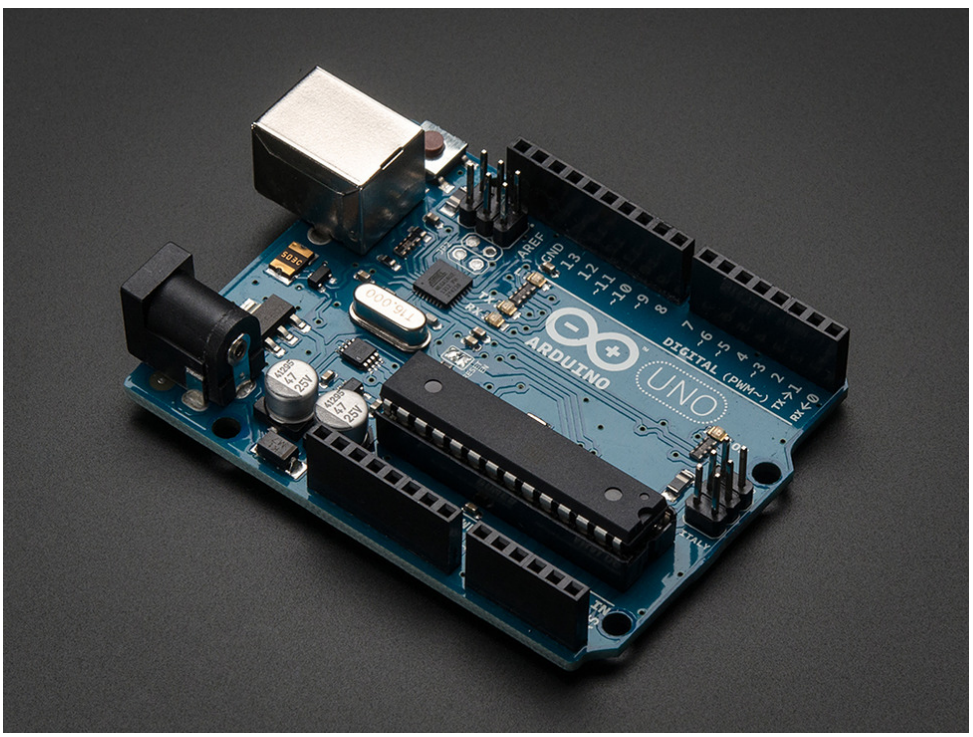


Figure 53. Image of Arduino Uno R3 microcontroller board.

Actual Approach Control

For all of their strengths Arduino boards are poorly suited for controlling DC Motors directly as the back electromotive force, caused by a rotating DC motor, will damage the sensitive electronics. Additionally, microcontrollers are incapable of providing the power required to drive a significant motor. For these reasons, a motor driver is required.

We have chosen to use the MD10C motor driver board to power the DC motor used in the Actual Approach method discussed previously. The MD10C driver board is shown in Figure 54. This board is designed for controlling a single motor on 5-25V with up to 30A for up to 10 seconds or 13A continuously; both of which are safely within our requirements. The driver features a fully NMOS H-Bridge which requires no heat sink and allows for full control of the motor with amplitude control forward and reverse plus free spin and brake functionality.

Figure 54. Image of Cytron MD10C Single DC Motor Controller.

To control the MD10C motor driver, we will use the sign-magnitude method in which the direction of the motor as well as the magnitude of the power is specified separately. This will provide a more stable platform at startup compared to the locked-antiphase mode available. This mode requires input from both a digital pin to specify the direction and a PWM input to specify the magnitude. As shown in Figure 55, these pins will be provided by the

Master Arduino board, pins 2 and 3, respectively. The board will also require the 24V motor power source input. The two outputs to the motor are labeled A and B.

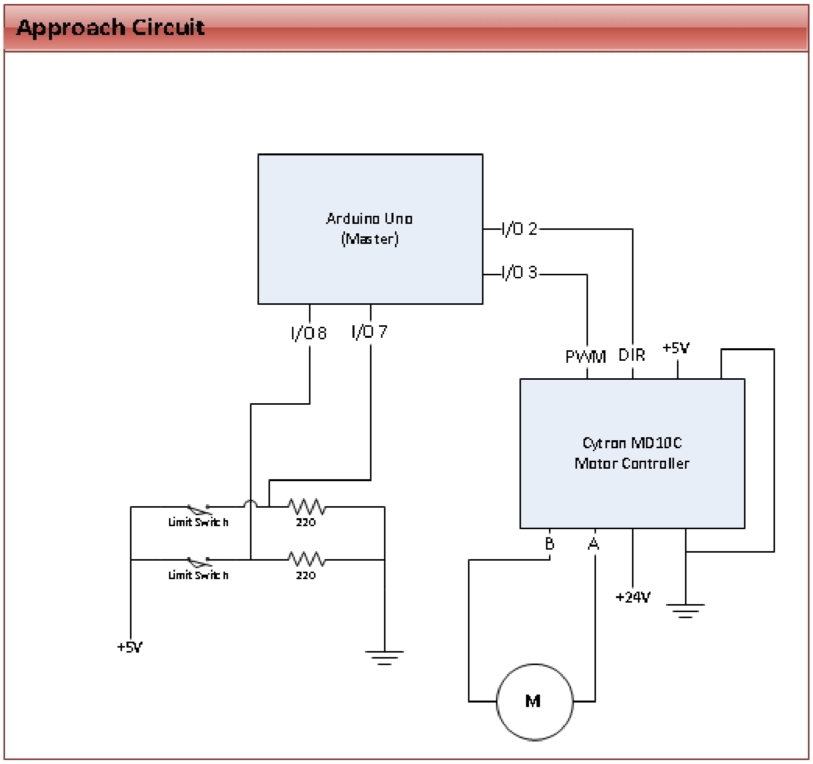


Figure 55. Circuit diagram for Actual Approach deterrence method utilizing Cytron MD10C motor controller and dual travel limit switches.

Figure 55 also shows the use of the travel limit switches employed in the Actual Approach method. These switches will be used to determine when the approach car has reach the end of travel and thus, the motor must be stopped and returned back to the reset position.

Sprinkler Control

Similar considerations to the back EMF must be considered with the sprinkler deterrence method because the solenoid controlling the water flow will also produce this damaging effect. However, unlike the motor, the solenoid only requires on/off control, thus a specific controller is not required. Figure 56 shows the circuitry required to safely control the sprinkler solenoid. The circuit uses a power MOSFET used as a switch along with a diode to protect the microcontroller.

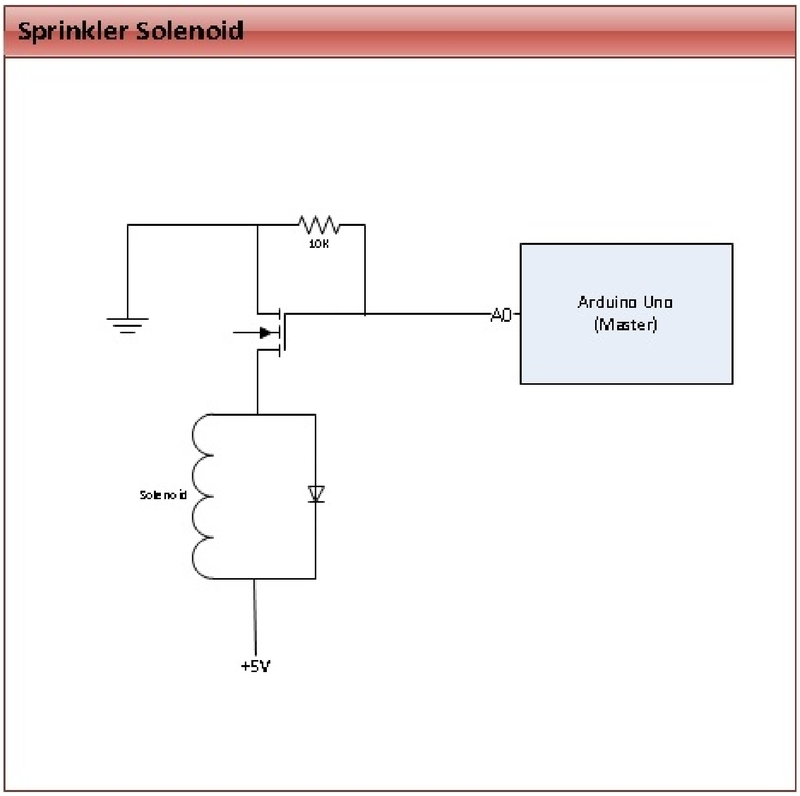


Figure 56. Circuit diagram for sprinkler control with back EMF protection.

Speaker System Control

The control for the speaker system (both simulated approach and stationary) will be controlled with the Slave Arduino Uno fitted with the Adafruit Wave Shield, shown in Figure 57. The Wave Shield is a specialized board that is specifically designed to fit directly into the headers of the Arduino Uno, thus making the necessary connections without wiring of any kind. The shield method also requires less space because the shield is mounted directly on top of the microcontroller and does not require any additional space or hardware. The Wave Shield is designed to play moderate quality audio files off of an SD memory card within a low cost package. It plays any uncompressed 22 kHz, 16-bit mono Wave (.wav) file. See Appendix 7 for more detailed information regarding the Wave Shield.

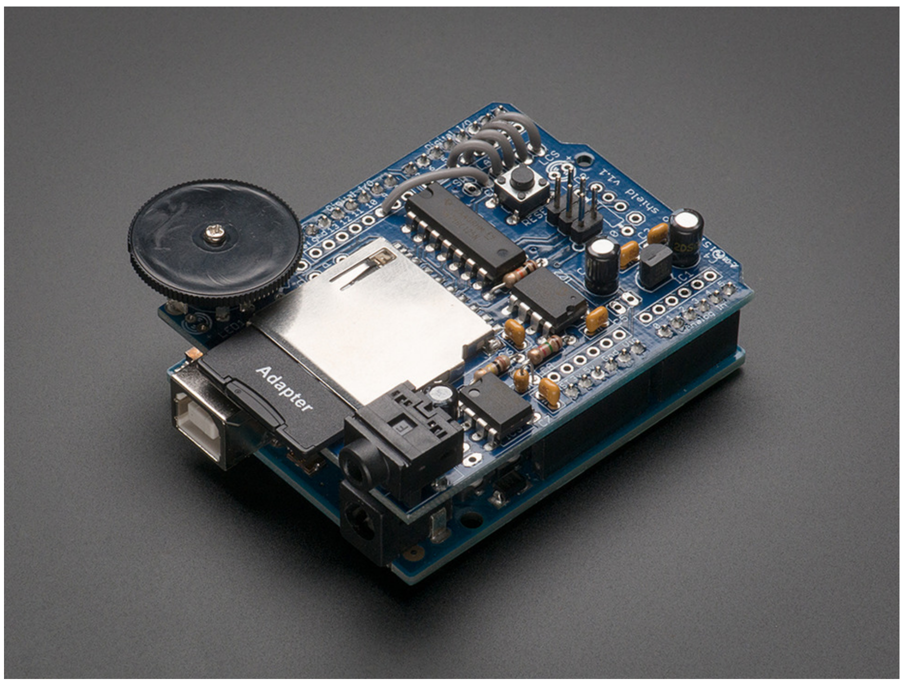


Figure 57. Arduino Uno with Adafruit Wave Shield Mounted on top.

Additionally, the audio signal outputted by the Wave Shield must be directed towards the appropriate speaker. This task will be accomplished using a series of Power N-Type MOSFETS to allow the Slave microcontroller to enable signal to the speakers using the I/O pins. The circuit diagram for this system is provided in Figure 58.

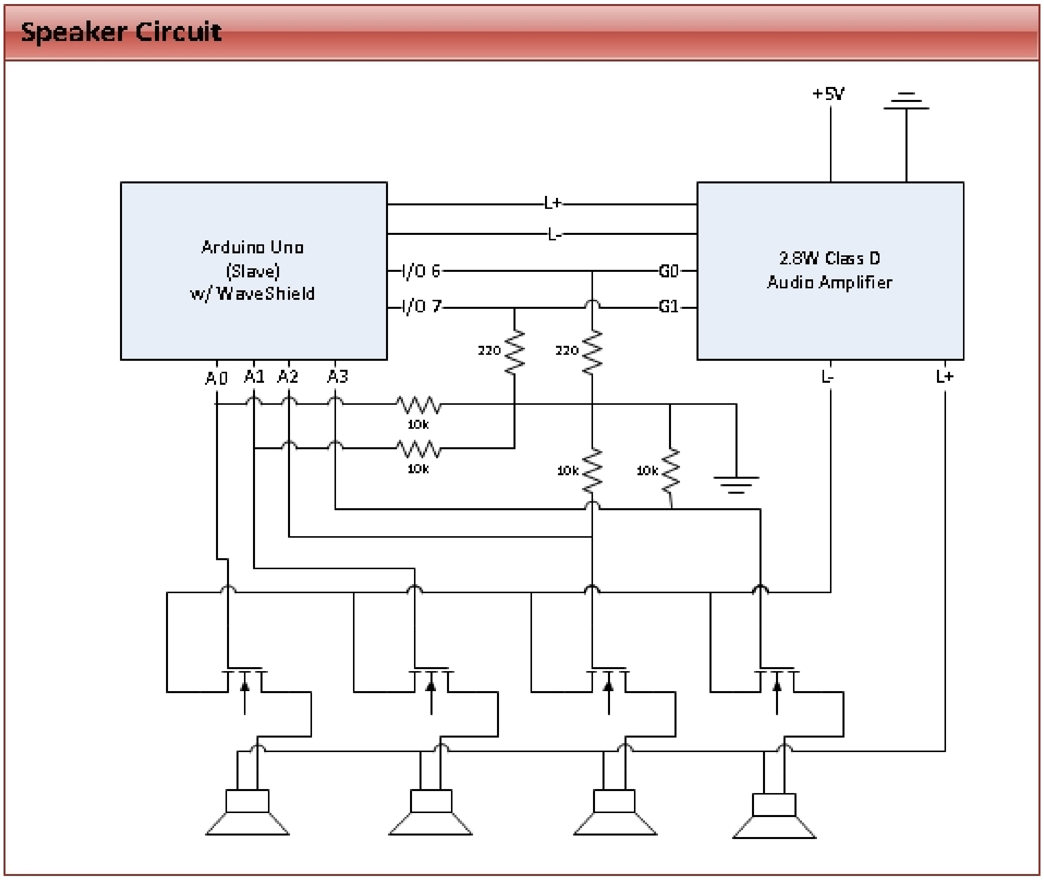


Figure 58. Circuit diagram for the speaker control system showing MOSFET control and Audio Amplifier.

Figure 58 shows an audio amplifier included in the circuit. This will be a simple board purchased from Adafruit which will take an audio input and amplify the signal to increase the sound as needed. It is designed for 4 Ohm speakers. Though the amplifier is built for stereo sound, we will simply leave one "side" of the stereo sound unused because the Wave Shield only outputs mono type sound. It is worth noting that the 2.8W class D amplifier board allows for programmable volume control such that would enable the software to automatically increase the volume of the sound if needed. This may be used to test the effectiveness of alternating the volume of sound as well as the location.

Rangefinder Control

The Rangefinder Targeting system will be controlled with the master microcontroller. As mentioned before the servo motor and ultrasonic rangefinder both interface directly with the microcontroller and thus the connections are greatly simplified. The circuit diagram for the Rangefinder control is provided in Figure 59.

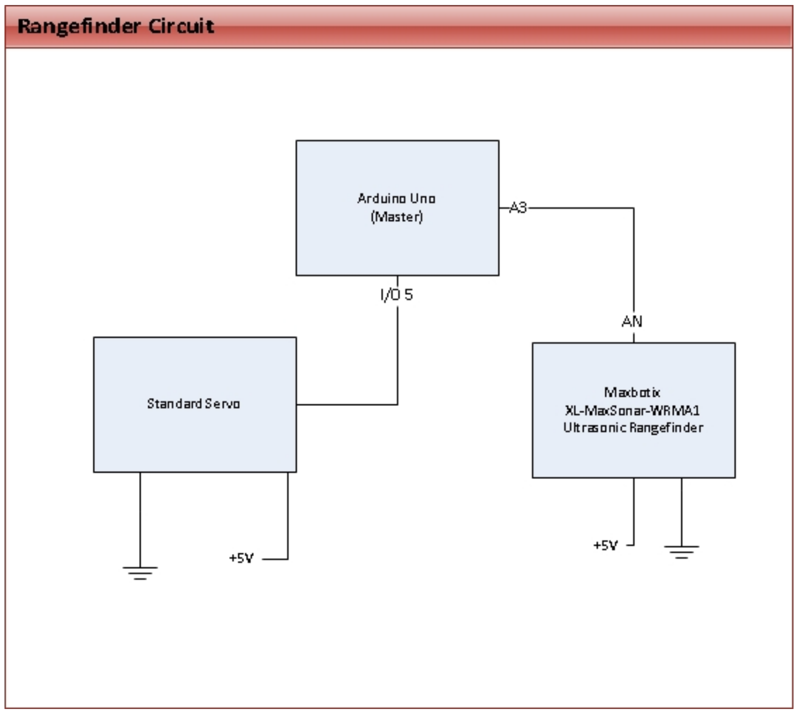


Figure 59. Circuit diagram of Rangefinder Targeting system control.

PIR Sensor Control

As in the Rangefinder circuit, the Passive Infrared sensor requires nearly direct connections to the microcontroller. The circuit diagram for this system is provided in Figure 60.

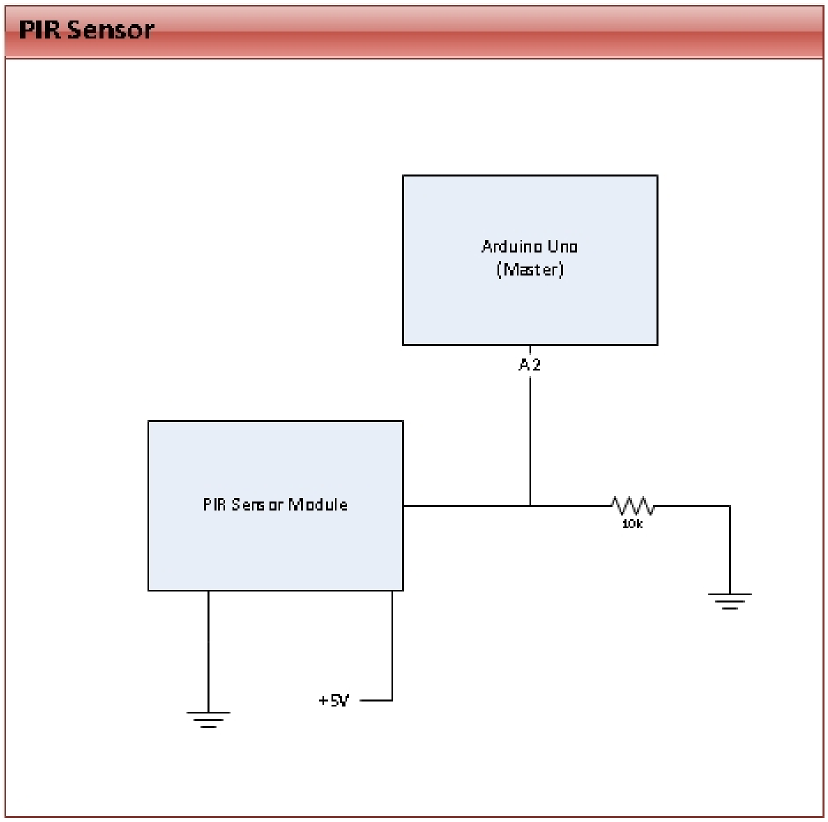


Figure 60. Circuit diagram for PIR sensor control.

User Notification

One of the customer requirements for this project is to notify the user of needs or interesting information electronically. To accomplish this, internet connectivity is required therefore we have chosen to employ the CC3000 WiFi shield from Adafruit, shown in Figure 61. This attachable shield was specifically designed for use with the Arduino Uno and features a removable SD card for storage, though this feature will likely not be used.

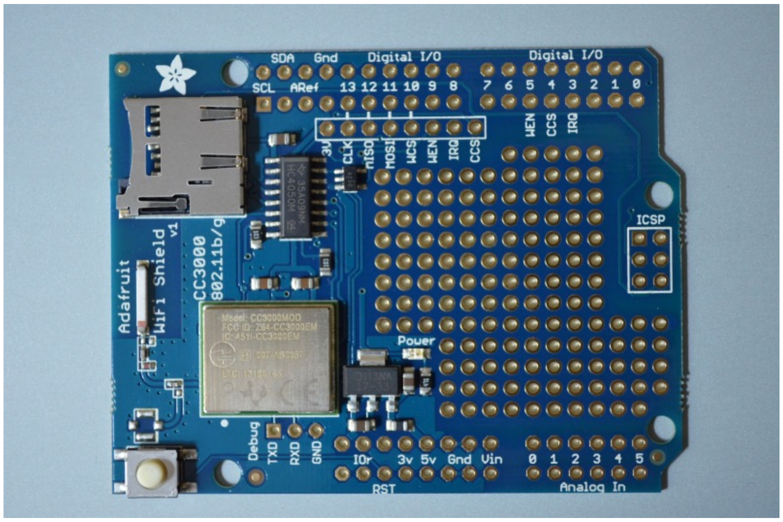


Figure 61. Image of Adafruit CC3000 WiFi shield for Arduino Uno.

Like the Wave Shield, the CC3000 WiFi Shield is intended to be placed directly on top of the microcontroller board and thus no additional wiring or circuit diagrams are required. Detailed information regarding the CC3000 WiFi Shield is provided in Appendix 7.

Cost Analysis

Table 11 lists the components not previously mentioned in other sections. It is the result of significant research into the most effective and cost efficient solutions available.

Table 11. Cost summary for electronic control system.

Item	Supplier	Unit Price	Quantity	Total Cost
General Control				**$156.81**
Arduino Uno R3	Amazon	$27.95	2	$55.90
Electronics Enclosure Case	Amazon	$23.52	1	$23.52
Power Strip	Amazon	$11.98	1	$11.98
Arduino Power Supply	Amazon	$6.99	1	$6.99
Anti-Static Foam	McMaster	$2.74	1	$2.74
10 kOhm Resistor (pack of 10)	Tayda	$0.10	1	$0.10
220 Ohm Resistor (pack of 10)	Tayda	$0.10	1	$0.10
4 Connector Wire (100 ft)	Amazon	$32.99	1	$32.99
2 Connector Wire (100 ft)	Amazon	$14.99	1	$14.99
PCB Board	Adafruit	$4.50	1	$4.50
Misc. Mounting Hardware	-	-	-	$3.00
User Notification				**$30.06**
Arduino Ethernet Shield	Adafruit	$30.06	1	$30.06
Actual Approach				**$56.23**
Cytron Motor Controller	Robot Shop	$16.33	1	$16.33
Limit Switches	Digikey	$2.45	2	$4.90
Motor Power Supply	Amazon	$35.00	1	$35.00
Simulated Approach				**$34.00**
Adafruit Wave Shield	Adafruit	$22.00	1	$22.00
MOSFETS	Tayda	$0.41	5	$2.05
Stereo Amplifier Breakout Board	Adafruit	$9.95	1	$9.95
Total Electronics				**$277.10**

Please note that the cost provided does not include shipping costs or applicable taxes which may be imposed by the supplier at the time of purchase.

Maintenance & Repair

While the system components are fragile, steps have been taken to protect the expensive components by housing them within an electrical component box. Detailed information regarding the enclosure is provided in Appendix 7 with technical drawings detailing the modifications required in Appendix 5.

In the event of electronic component damage, the item should be replaced as available.

There is no regular maintenance routine required for the control system as it is externally powered though the functionality of the overall system should be tested periodically. If a sub-system or component is not functioning correctly, it should be fixed by whatever means required with the system power off by unplugging the system power. While the electrical currents are not significantly high in most cases, a sudden surge of electricity in an unexpected way can damage the sensitive electronics.

Any technician working with the electronic components should take care to adequately discharge any static electrical charge present before coming approaching, especially in the case of the microcontrollers as a significant static shock can simply erase all or part of the instructions programmed.

Safety

While most of the system operates on 5V, the motor for the Actual Approach deterrence system is supplied with more than 3A at 24V. This is enough electrical power to seriously injure or even kill a person who may come in direct contact. Careful consideration will be given to ensuring that all exposed leads are adequately sealed and components are grounded properly.

Regular safety inspections should be performed to ensure that there are no loose wires or exposed leads, especially related to the Actual Approach motor. If a dangerous wire is discovered, disconnect the main power immediately and address the problem promptly.

Note that the system is controlled automatically and thus may move suddenly and without warning. Exercise caution when operating near any component of the device.

Cost Summary

While each sub-system presented a detailed list of the exact components required to make the project a reality, Table 12 presents a summary of the cost to implement each sub-system.

Table 12. Summary of sub-system costs.

Sub-System	Estimated Cost
Simulated Approach	$243
Targeting Rangefinder	$128
Actual Approach	$521
Control System	$124
Notification System	$30
PIR/Sprinkler	$69
Motion Activated Camera	$117
Total	**$1,232**

Please note that these costs are rounded estimations of cost. Many of the figures include estimated allowances for miscellaneous parts. The figures presented do not include relevant taxes or shipping charges which may be applied at the time of purchase.

While we are over our prescribed budget of $1000, we feel that the solution presented in this document will effectively accomplish the goals proposed by JumpSport.

Ch 4. Final Design

The initially proposed system was proposed to the sponsor on May 5, 2014 and portions of the design were rejected while others were accepted. Specifically, the actual approach system was eliminated and the need for a sub-system to simulate eyes was added. Throughout additional design and development the system design evolved to the one described in the next section. The details of the system evolution will be discussed later in this report.

The following section details the final system design prior to manufacturing. Design changes were made during the process, both by sponsor recommendation and practical necessity. The changes and their reasons will be detailed in a later section.

System Overview

The final Wildlife Deterrence system is a modular scaring system design to stop deer from entering suburban yards and gardens. The entire system consists of several subsystems, as in Figure 62 below, all of which incorporate different approaches at scaring deer. The main components of the system include four sets of predator eyes, four simulated approach platforms and an ultrasonic rangefinder.

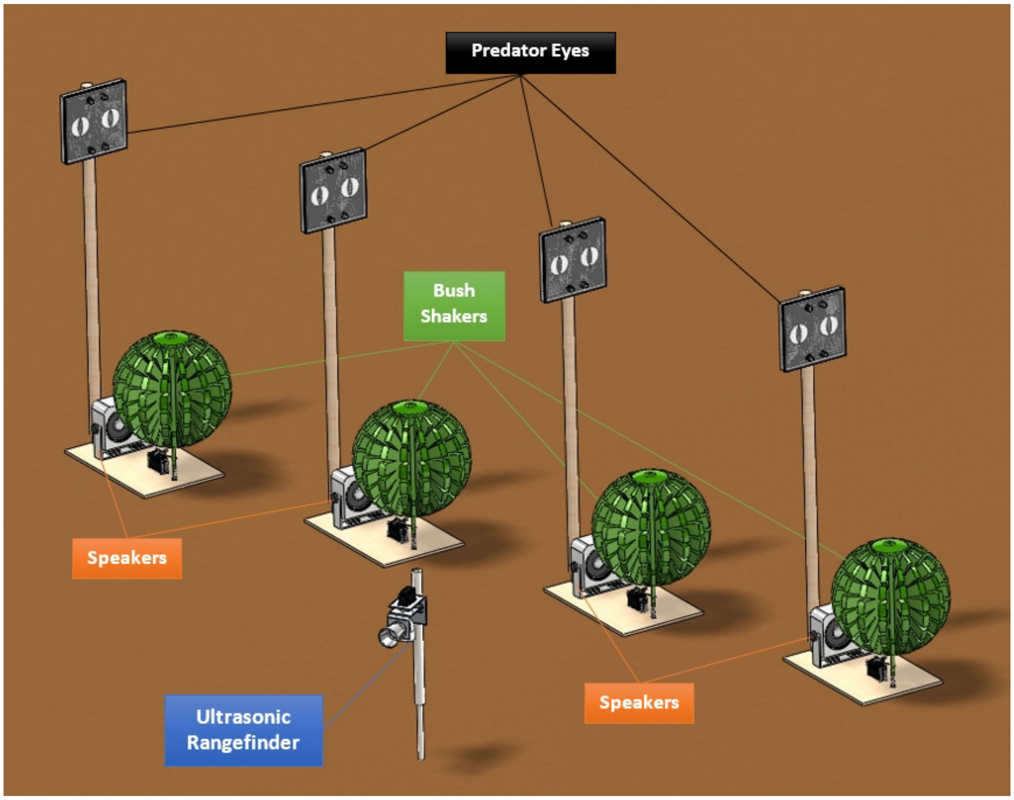


Figure 62. Overall system layout showing Rangefinder and simulated approach sub-systems.

All of the components of the system are triggered by the ultrasonic rangefinder once it picks up motion after scanning the area for discrepancies. Once triggered, the system will receive signals from the main control unit to turn on. Depending on which subsystems the operator has selected, the farthest unit from the triggered area shall go off and will be followed by the next closest unit. This method of motion amongst the components is also known as simulated approach. Its primary goal is to make it appear as though a predator is moving in the direction of the deer and happens in many ways. The first way is by the predator eyes that imitate the green glowing eyes of a mountain lion or big cat at night. The second is by the shakers which create the illusion of an animal in the bushes and it is hastily approaching for the attack. The last method to inspire fear is the speaker system that will play various animal sounds or any other noises the operator wishes to test. In order to make sure the system is working properly and triggering, a passive infrared trail camera shall also be placed in the area of the system so as to record all interactions the machine has. Lastly, the only way we know for sure the system is affective is to compare it with another system out on the market that has worked before. That is where the motion activated sprinkler comes in. It shall be separately be set up along with the whole system as a control for the experiment.

Simulated Approach (Movement)

Overview

The simulated approach system's purpose is to simulate the motion and sound of a predator moving in some nearby shrubs. It ultimately will induce anxiety and fear of attack into the deer that enter the target area. The bushes shall shake in series starting from the farthest bush to the closest, as well as speakers shall simultaneously play a predator noise or any noise preferred by the operator. As of now, our system consists of four separate units that can be placed anywhere around the desired area in any configuration. The subsystem itself was designed to be outside, therefore each of the speakers is waterproof and all of the servo motors have been weatherproofed.

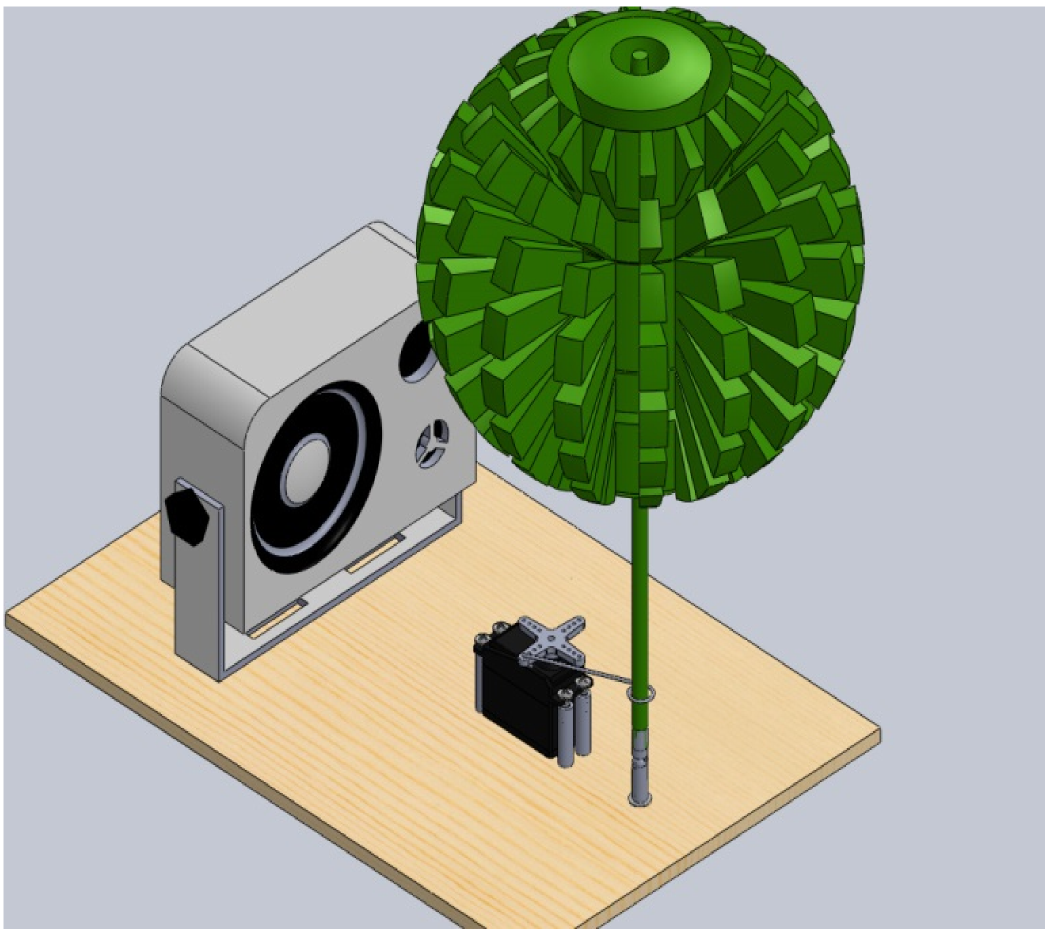


Figure 63. Image showing layout of motion aspect of simulated approach platform design.

A single shaker system will consist of a hobby servo, three fake plants, a metal attachment rod, a pivoting hinge, and a weatherproof speaker as seen in Figure 63 above. The servo is a standard sized hobby servo consisting of a small brushed DC motor, which is the same motor used on the rangefinder subsystem. In order to prevent water damage to the servos, white lithium grease was applied to vulnerable servo surfaces. For the speakers, we chose to use the Pyle PLMR24 Hydra 200 W Speaker because it is inexpensive, has good reviews, and is waterproof.

Cost Analysis

The sub-system was designed with the goal of minimizing cost and having as few parts as possible. Table 13 details the components required and their associated costs.

Table 13. Cost summary of components involved in the Simulated Approach (Motion) sub-system.

Item	Supplier	Unit Price	Quantity	Total Cost
Fake Plants	Dollar tree	$1.00	12	$12.00
Towerpro SG-5010 Servo w/ Hardware	Adafruit	$12.00	4	$48.00
Hinged Threaded Standoff	McMaster	$3.18	4	$12.72
Straight Threaded Standoff	McMaster	$1.29	16	$20.64
#8-32 x 1/2" Machine Screw (4 pack)	Home Depot	$1.50	2	$3.00
Wire Hook (1/16" Diam x 3'L)	McMaster	$1.83	1	$1.83
Pyle PLMR24 Hydra 200 W Speaker	Staples	$17.49	4	$69.96
16 Gauge Outdoor Speaker Wire(150ft)	Amazon	$43.99	1	$43.99
4 GB SD Card	Amazon	$4.37	2	$8.74
Total Hardware				**$302.28**

Maintenance & Repair

The simulated approach sub-system should not require any regular maintenance. All of the parts are powered from the main control unit and thus do not require changing batteries or any other regular interaction. The shaker hinges may come unscrewed some from all of the shaking but that is a simple fix and should not harm the system.

Safety

This sub-system poses no threat due to the low voltages the motor and speaker draw and the fact that the rest of the system consists of fake plastic plants. The only potential safety hazard would be the potential to trip over the units if someone were to walk near them and not pay attention.

Simulated Approach (Eyes)

Overview

The predator eye system is a part of the simulated approach platform. It consists of a set of four 'eyes' that blink in series, simulating the ‘glow’ of the eyes of a predator coming toward the animal.

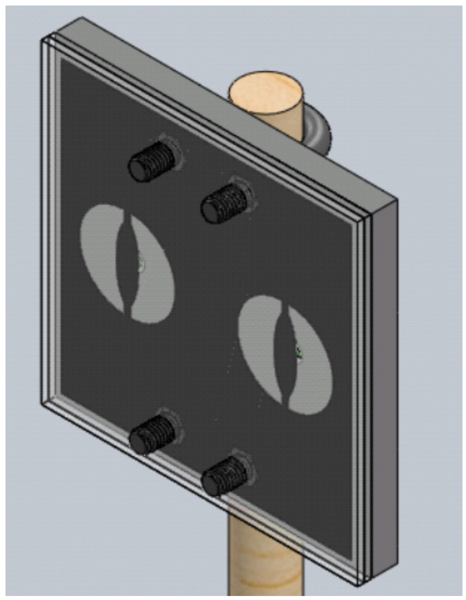


Figure 64. Front view of assembled Eye System

The system consists of a UHMW plate with holes for the placement of two LEDs and two U-bolt connections. Two acrylic plates hold a plastic ‘eye’ sheet and will be placed onto the same two U-bolt connections. The eye sheets help diffuse the LEDs so that they do not emit direct, intense light, which is not representative of the ‘glow’ of a predator eye. The two U-bolts are used to mount to a three foot wooden stake. The stake is sanded into a point to allow it to be placed into earth securely. This configuration is shown in Figure 64, above.

10mm LEDs are press fit into the eye holes of the UHMW plate as seen in Figure 65. Once press fit, the LEDs are wired and soldered to a two wire connector. All exposed wires are covered in Plasti-dip to help insulate the electrical system and increase safety. Green LEDs were selected because deer and many other wildlife are able to see green wavelengths well, unlike other colors such as red. Additionally, green LEDs are very common and are low cost.

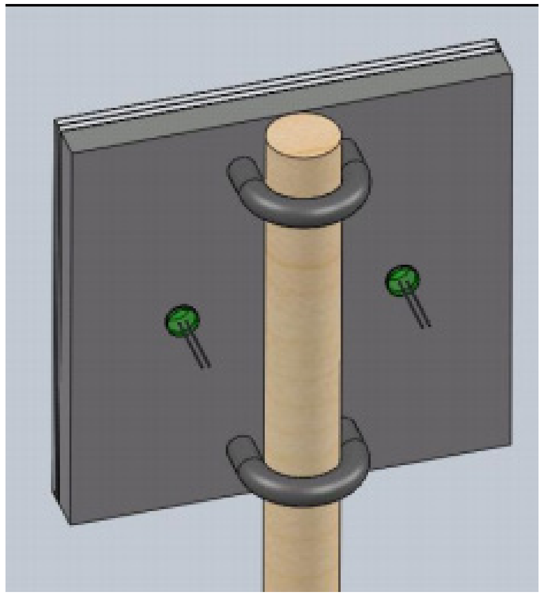


Figure 65. Back view of assembled Eye System.

Cost Analysis

The sub-system was designed for simplicity and low cost while effectively delivering the stimulus as naturally as possible. The total cost of the system was greater than expected because of the multiple units being constructed. Table 14 details all ordered components and costs.

Table 14. Cost summary for Eye system.

Item	Supplier	Unit Price	Quantity	Total Cost
10 mm Green Diffuse LEDs (10 Pack)	Adafruit	$9.95	1	$9.95
4-Wire Connector	Adafruit	$2.50	9	$22.50
1/8” 6”x6” UV Acrylic	McMaster	$2.14	8	$17.12
½” 6”x6” UHMW	McMaster	$5.69	4	$22.76
U-bolts	McMaster	$1.31	8	$10.48
6ft. 1” D. Wooden Pole	Amazon	$9.00	4	$36.00
Total Hardware				**$118.81**

Please note that the cost provided does not include shipping costs or applicable taxes which may be imposed by the supplier at the time of purchase.

Maintenance & Repair

The eye system should not require any regular maintenance. All parts are powered from the main control unit do not need any supplemental utilities. The system is generally built to handle corrosion. Occasionally an LED may fail or a solder joint could falter. This may be repaired by a replacing the LED or resoldering the connection. It is important that the polarities of the LED be wired correctly when replacing LEDs.

Safety

This sub-system is generally very safe. All wires are insulated and have very low currents running through them. There is a potential tripping hazard due to the number of wires needed to power the system. These should be marked appropriately when the system is operating.

Rangefinder Targeting System

Overview

The Rangefinder system will scan the environment and return the relative distance of the nearest “large” object for a range of directions from 0º to 95º. With this data, a device could determine the location of an animal relative to the sensor. The scope of this project includes only the collection of data and presentation to the user; any data analysis will be left to others.

The system will consist of an ultrasonic rangefinder mounted, via a custom connection bracket, to a standard hobby servo. The hobby servo will be mounted to a second bracket which is fixed to a post staked into the ground. This configuration is shown in Figure 66, below.

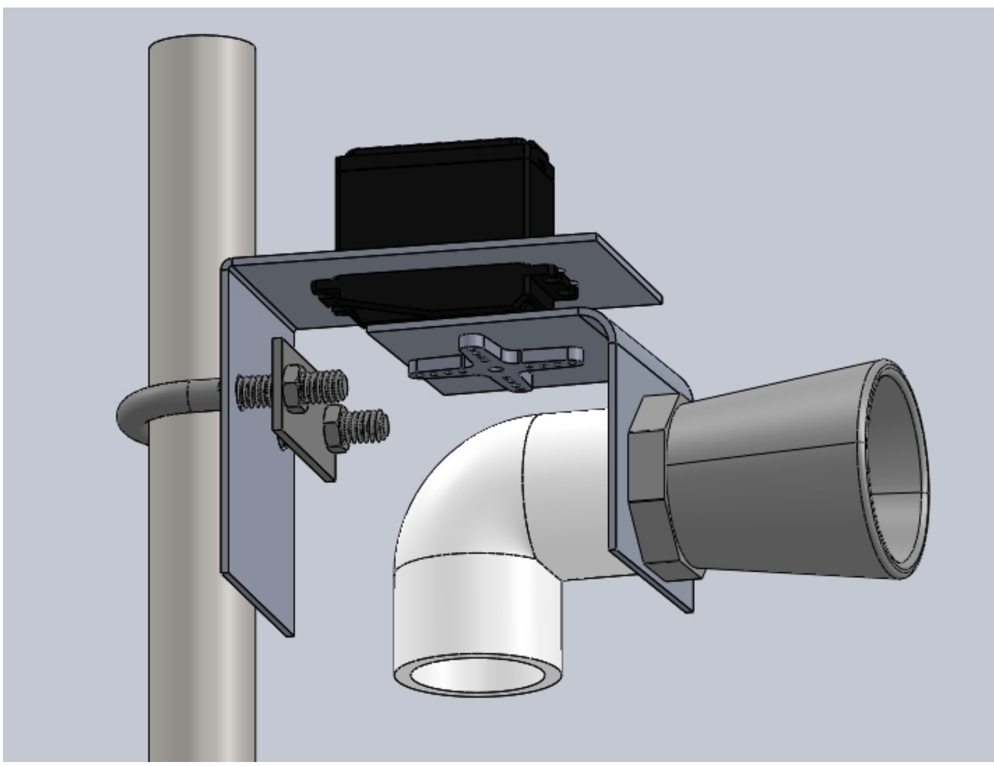


Figure 66. Model view of Rangefinder Targeting system assembly.

The sensor has been chosen to be an XL-MaxSonar-WRMA1 ultrasonic rangefinder from Maxbotix. The sensor is IP67 rated protected against short-term water immersion and thus is suitable for outdoor use. The sensor features real-time background calibration, real-time waveform signature analysis and noise rejection algorithms for very accurate readings. We expect these features to enable this sensor to provide reliable data even in light rain or

wind. Additionally, the sensor is calibrated with narrow sensor beams to enable up to 25 feet of range. Detailed information regarding the ultrasonic sensor can be found in Appendix 7.

The sensor housing is fitted with a ¾" NPT pipe thread on the rear of the device so that dry electronic connections can be made. This feature is taken advantage for mounting; a thin piece of aluminum plate is sandwiched between the threaded connections providing a secure attachment. A 90 degree bend fitting was used so that the opening for the wires points down, preventing rainwater from damaging the sensitive electronics. The aluminum plate is bent such that it will interface with the servo motor as shown in Figure 49. The plate has the necessary holes drilled for each interface and is bent to the approximate angle manually.

The servo is a simple, standard sized hobby servo consisting of a small brushed DC motor, a gear ratio of 180:1 and circuitry allowing for controlled angular positioning all within an inexpensive and simple package. It is important to note that hobby servos are not typically water-tight. To prevent water damage, white lithium grease was applied to vulnerable servo surfaces. However, the rotational power transmission opening will not be sealed to allow for free rotation. This opening is why the servo is positioned upside-down, preventing water from climbing up, past the gears, to the motor and electronics.

The servo is mounted to a plate with the standard mounting screws included (not shown) to a second aluminum plate. This plate is bent at 90 degrees and permits the use of a U-bolt to attach to a pole. Technical drawings detailing the dimensions and hole configurations of the plates are provided in Appendix 5. This interface enables the effective height of the sensor to be raised or lowered as seen fit by the user up to 3 feet from the ground.

Figure 67. Model view of Rangefinder.

The pole is a three foot, one inch diameter wooden pole. The wooden pole is sanded into a point for ease of placement into the ground and low cost. The wooden pole used will be identical to those used for the 'eyes' subsystem. Figure 67 shows the full assembly.

Cost Analysis

The sub-system was designed with the goal of minimizing cost while not sacrificing quality. For this reason, an advanced sensor was chosen to provide quality data reliably. The remaining components were chosen, primarily with cost in mind as their functions are non-critical and often trivial. Table 15 details the components required and their associated costs.

Table 15. Cost summary for Ultrasonic Rangefinder Targeting system.

Item	Supplier	Unit Price	Quantity	Total Cost
Maxbotix XL-MaxSonar-WRMA1 Ultrasonic Rangefinder	Adafruit	$84.95	1	$84.95
Towerpro SG-5010 Servo w/ Hardware	Adafruit	$12.00	1	$12.00
Rangefinder Aluminum Mounting Plates (2)	McMaster	$1.96	1	$1.96
90deg threaded PVC Elbow	McMaster	$0.44	1	$0.44
1" Wooden Pole	Amazon	$9.00	1	$9.00
U-Bolt Clamp for ½" Pipe	McMaster	$1.02	1	$1.02
#4 SS Screws (8 Pack)	Home Depot	$1.50	1	$1.50
White Grease	Amazon	$4.28	1	$4.28
Total Hardware				**$114.15**

Please note that the cost provided does not include shipping costs or applicable taxes which may be imposed by the supplier at the time of purchase.

Maintenance & Repair

The Rangefinder sub-system should not require any regular maintenance. All of the parts are powered from the main control unit and thus do not require changing batteries or any other regular interaction. All exposed metals have basic corrosion protection (aluminum and galvanized steels) and are under very low stresses. Any corrosion that does occur to exposed surfaces will not have an effect on the performance of the system.

The servo motor does rely on plastic gears which may become worn or broken due to extreme circumstances. In this case, we recommend purchasing a replacement motor and replacing the broken one. While this is not an ideal solution, the likelihood of failure is low and the

component is relatively inexpensive, however the servo casing must be sealed prior to outdoor use to prevent rain damage.

Safety

The sub-system poses no significant safety threat as there are no exposed wires or high voltages and there is not sufficient torque produced by the servo. Though there are potential pinch points, we expect the motor to stall easily if its path is obstructed.

The sensor frequency is 42 kHz which, while outside of the range of human hearing, may be detectible by household animals such as dogs and cats (not deer). This sound will likely not be damaging to the animal but could be bothersome over long periods of time, so prolonged exposure to the emitted sound should be avoided for household pets.

Electronics

Overview

The electronics system requires several individual modules which control the operation of each sub-system individually. Two Arduino Uno microcontrollers were chose to control the system because of their cost effectiveness and ease of use. More information about the Arduino Uno can be found in Attachment 10. For this system, the circuit shown in Figure 68 was designed.

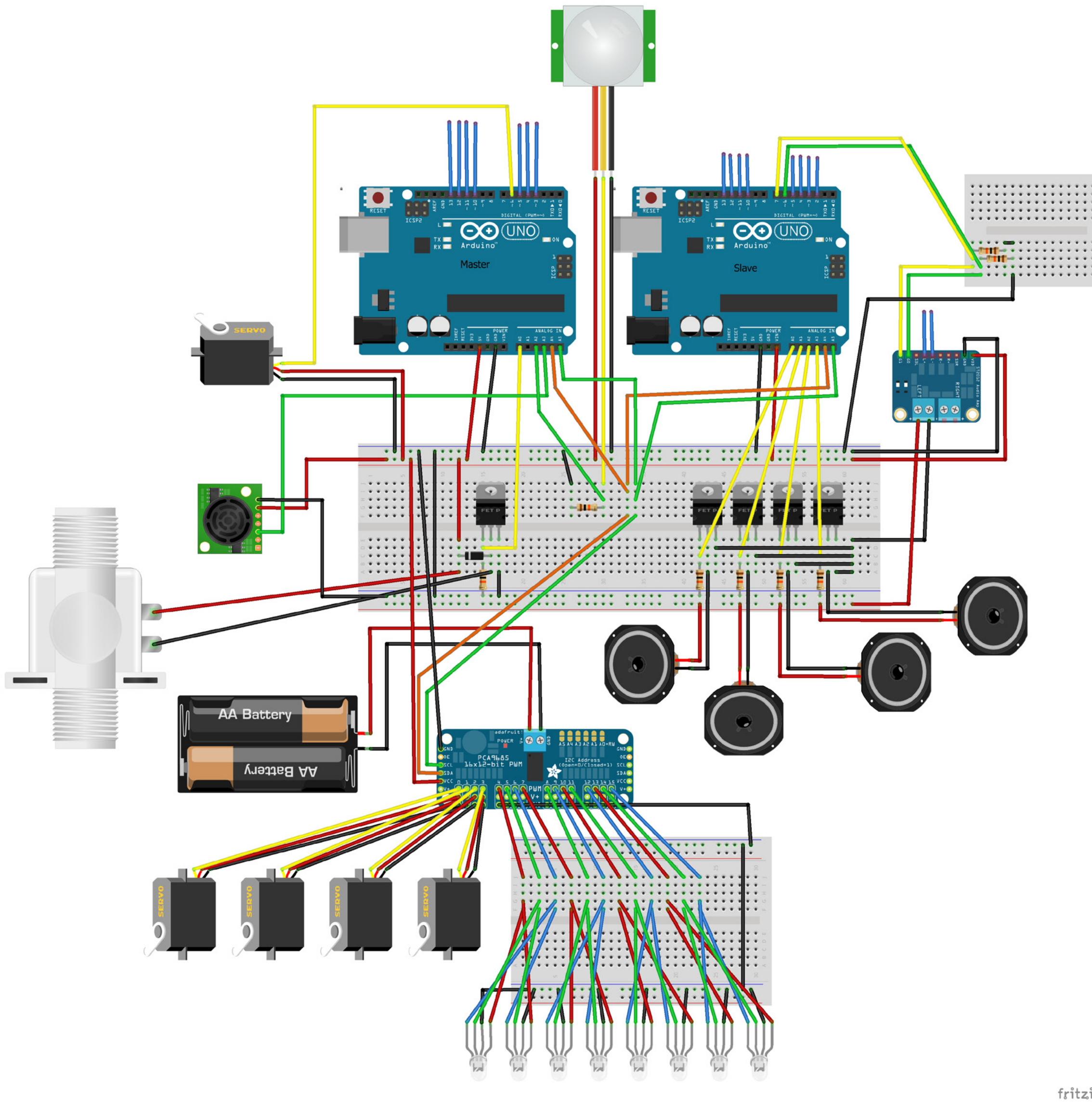


Figure 68. Proposed electronics layout diagram.

The main "Master" microcontroller is designated to control most of the functions of the system. It directly actuates the Rangefinder servo with a PWM output and the sprinkler solenoid using a blowback protection circuit, seen in Figure 69 in order to prevent damage to the microcontroller due to back EMF resultant from inductive loads. Note that this same circuit is used, for the same purpose, for the speaker system. The Master microcontroller also receives the direct inputs from the PIR motion sensor and the rangefinder distance.

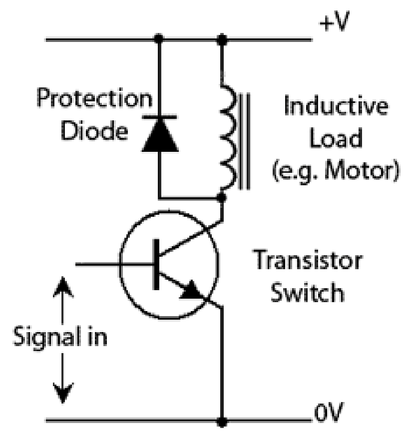


Figure 69. Back EMF protection circuit implemented to prevent harmful back EMF.

Additionally, the Master controls both eyes and motion aspects of the simulated approach using the 12-bit PWM Driver from Adafruit. This small board accepts signals from the microcontroller over the I2C communication protocol and uses an external power source to power the connected devices as desired. See Appendix 10 for more information on this component. The Slave microcontroller is also controlled over the I2C protocol allowing the Master to send commands and the Slave to accept them and act accordingly.

The Slave Microcontroller controls all of the aspects of the sound sub-system using the Adafruit Wave shield, shown in Figure 70, to generate the audio desired. The Wave Shield fits snuggly on top of the Arduino Uno so is not shown in the system diagram. It is capable of playing uncompressed .wav files as a mono output.

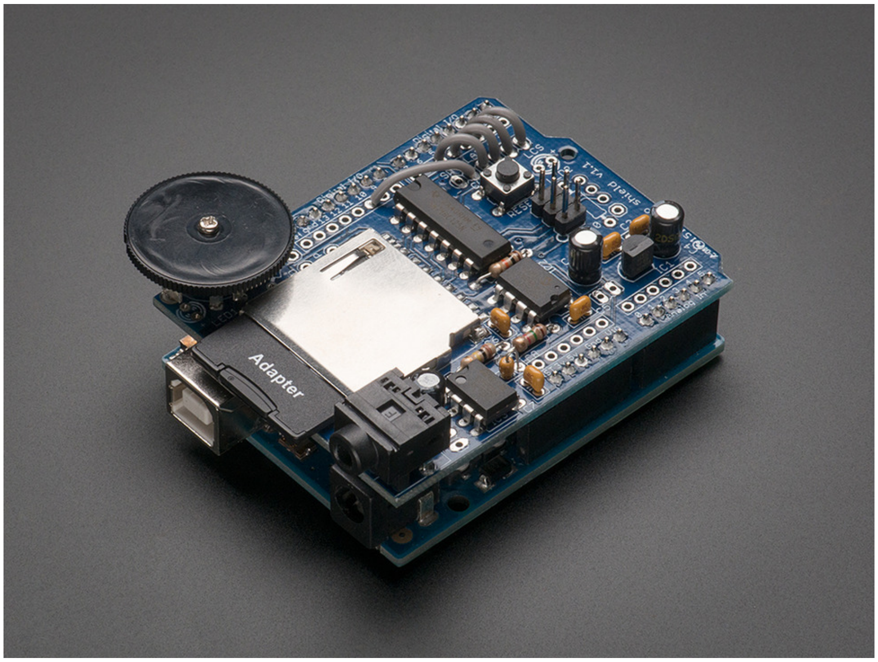


Figure 70. Image of Adafruit Wave Shield installed onto an Arduino Uno.

To enhance the available volume, the 2.8W Class D audio amplifier was chosen from Adafruit. The signal from the amplifier is then provided to the speakers through MOSFETs. The MOSFET array allows the slave microcontroller to selectively allow the sound to reach any speaker desired. Note that the electrical system design allows for multiple speakers to be enabled at once but the software is written in a way which prohibits this feature.

The entire electronics system would be housed inside of a sheet metal case to protect the sensitive electronics from damage during handling. The electronics are intended to be stored indoors so waterproofing is not necessary.

Cost Analysis

The costs of the system include the cost for the physical components and some consumable supplies required to assemble the components and system.

Table 16. Cost summary for electronics sub-system.

Item	Supplier	Unit Price	Quantity	Total Cost
Arduino Uno	Amazon.com	$27.95	2	$55.90
Wave Shield for Arduino	Adafruit	$22.00	1	$22.00
Stereo Amplifier Breakout Board	Adafruit	$9.95	1	$9.95
PCB Prototype Board	Adafruit	$4.50	2	$9.00
4 GB SD Card	Amazon.com	$4.37	2	$8.74
MOSFET	Tayda Electronics	$0.41	5	$2.05
Rectifier Diodes	Tayda Electronics	$0.02	5	$0.10
16-Channel PWM Driver	Adafruit	$14.95	1	$14.95
Enclosure Case	Amazon.com	$23.52	1	$23.52
Power Strip	Amazon.com	$11.98	1	$11.98
5V Switching Power Supply	Adafruit	$9.95	1	$9.95
9V Switching Power Supply	Adafruit	$6.95	2	$13.90
10kOhm Resistors (pack of 10)	Tayda Electronics	$0.10	1	$0.10
220 Ohm Resistors (pack of 10)	Tayda Electronics	$0.10	1	$0.10
Adafruit CC3000 Wifi Shield	Adafruit	$39.95	1	$39.95
4-Wire Connectors (Waterproof)	Adafruit	$2.50	12	$30.00
2-Wire Connectors (Waterproof)	Adafruit	$2.50	16	$40.00
Hookup Wire Set	Adafruit	$15.95	1	$15.95
Misc Consumables & Hardware (est.)	Various	$5.00	1	$5.00
Total Cost				**$313.14**

Maintenance & Repair

The electronics sub-system should be periodically checked to ensure that all wires are properly connected. Many wires are not permanently attached to the microcontroller relying on friction connections; these wires have the correct pin clearly labeled on them. If a wire comes loose from its proper location, it should be promptly plugged in and the system reset.

When handling or working near electronic components, care should be taken to reduce the risk of static shock as such a shock could irreparably damage many of the chips in use on this board. Any technician should ensure that all static electricity is discharged before beginning work.

Avoid storing or handling liquids near electronic components as certain liquids can bridge connections and cause severe damage to electronic components when spilled over them.

Safety

Electricity always poses some risk of danger and care must always be taken to prevent electric shock. This system deals in generally low voltage (5-9V) and thus, the danger is minimized. Despite this low risk, always disconnect power from the system before interacting with any unshielded wire or lead.

Software

Overview

The software portion of the project is written in a version of C++ specifically for Arduino microcontrollers. This language was chosen because of its simplicity and extensive online community to provide support if needed.

The program is structured using the method of cooperative multi-tasking. In this structure, the processor simulates the control of multiple tasks by completing small portions of the specific task in succession so that each task is allowed to perform a short task before moving on to the next one. The effect of this is that all of the tasks appear to run simultaneously due to the rapid computation speed of the microcontroller. This method is successfully implemented in many simple applications but its effectiveness is limited in larger systems.

The alternative to cooperative multitasking is that of pre-emptive multitasking in which each task is given a priority and an operating system allocates computation time based on importance. This method is often implemented in systems which perform a few important functions with some non-critical or slower functions included. This method is implemented in more complex and time-critical situations but is more complicated to implement and more resource intensive due to the required operating system.

Task Organization

The program is organized into 5 tasks with each task being responsible for a certain aspect of the system. Within each of these tasks exist a series of states which the task can remain in until it is instructed to change. Each of these states performs the proper actions required for the situation until it is changed. The State Transition Diagram is useful to show the states in each task and the conditions which cause a state transition (represented by arrows). The tasks are as follows:

MasterMind

MasterMind is responsible for coordinating all of the other systems. Implementing this single coordinating task helps to prevent a confusing web of interactions between tasks. With this method, all tasks interface with MasterMind only and not each other. This allows the tasks to quietly perform their own tasks as needed, and without regard to the status of the others, greatly simplifying each task. Figure 71 shows the State Transition Diagram for the MasterMind Task.

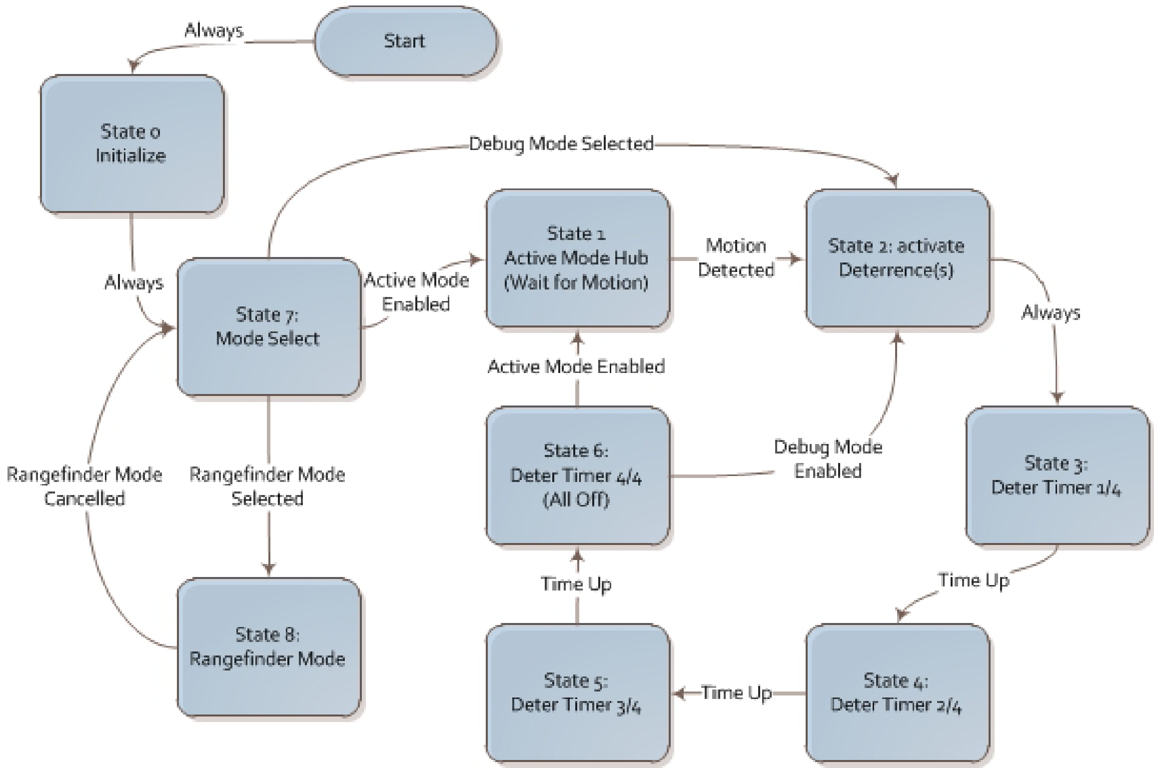


Figure 71. State Transition Diagram for MasterMind Task.

User Interface

User Interface is responsible for all interactions with the user. It prints messages through the serial port to the terminal on the computer and reads the user's inputs. These inputs are confirmed and relayed to MasterMind so that the proper action can be taken. Figure 72 shows the State Transition Diagram for the User Interface Task.

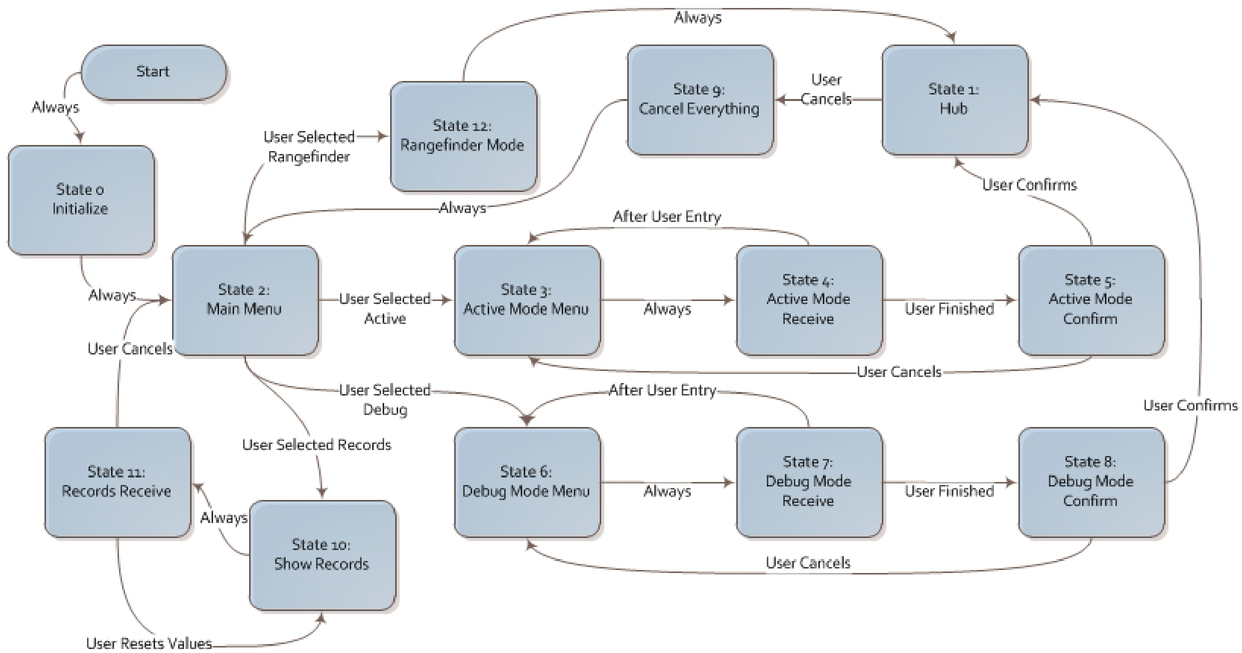


Figure 72. State Transition Diagram for User Interface Task

There are several menus which are displayed by the controller. The first being the Main Menu, provides the user with several choices which correspond to numbers which the user can enter to make a selection. The Main Menu is shown below.

```
Main Menu
----------
a: Active Mode
d: Debug Mode
r: Range Finder Mode
s: Show Records
i: Information
h: Help
```

Figure 73. Main Menu as displayed to the user.

Here the user may choose one of several modes to enter into, each of cause a relay to MasterMind, or the user can choose to show the number of times that the system perceives to have seen an intruder in the field of view with the "Show Records" selection. This window will also allow the user the option of resetting the counter to zero or to return back to the Main

Menu. The information option will display a short message regarding the purpose of the machine and the designers and then return to the Main Menu. The help selection will display a short message to the user providing instructions on how to use the menu.

In both the "Active Mode" and "Debug Mode" options, the user is presented with another menu in which they are able to choose the deterrence methods which they would like to enable. These menus are displayed in Figure 74 and Figure 75.

```
Active Mode Menu
----------
a: Shakers
e: Eyes
d: Done Selecting
h: Help
x: Stop and Go Back
```

Figure 74. Active Mode Menu.

```
Debug Mode Menu
----------
a: Shakers
e: Eyes
d: Done Selecting
h: Help
x: Stop and Go Back
```

Figure 75. Debug Mode Menu.

In each of these menus, the user selects the deterrence menu to enable. Selecting a method once will activate a static method in which the there is no simulated approach. Selecting the same method a second time will enable the sequential activation of the type of method simulating an approach towards the motion in the area of effect. Selecting the same method a third time will cancel the selection. Any combination of all deterrence methods can be selected at any time. Once the user has finished selecting, the user selects "d" and then confirms based on a prompt. If Active Mode was selected, the system will be enabled when motion is detected. Conversely, if Debug Mode was selected, the deterrence methods will be activated immediately and continuously until the cycle is cancelled by the user by entering "x".

The last mode available to the user is the Rangefinder mode. In this mode, the rangefinder is activated and the results are displayed on the screen rather than used to trigger any deterrence method. The distance at each angle is displayed in a single row for each pass along the range such that each column corresponds to a single angle over time. The magnitude of the number displayed corresponds to the linear distance from the detector. If the system detects motion, an exclamation point (!) will be displayed to the right of the number which caused the detection. An example of this display is shown below in Figure 76.

102	290	273	290	270	155	228	299	339	179
101	294	275	293	273	167	222	301	342	182
110	287	265	290	277	139	227	310	327	172
99	292	276	292	260	144	243!	387	342	183
180!	291	280	296	279	160	222	305	339	175
102	200!	278	290	265	150				

Figure 76. Example of Rangefinder Mode display showing distances read at each position with motion detection marked.

Rangefinder

The Ranger Task handles the operation of the rangefinder sub-system. It is enabled and disabled by a command from Mastermind and controls both the servo and ultrasonic rangefinder module. The rangefinder compares the distance reading with the one taken previously at the same location. This difference is then compared to a sensitivity allowance and used to determine whether the environment has changed, implying an intruder. The State Transition Diagram for the Rangefinder Task is provided in Figure 77.

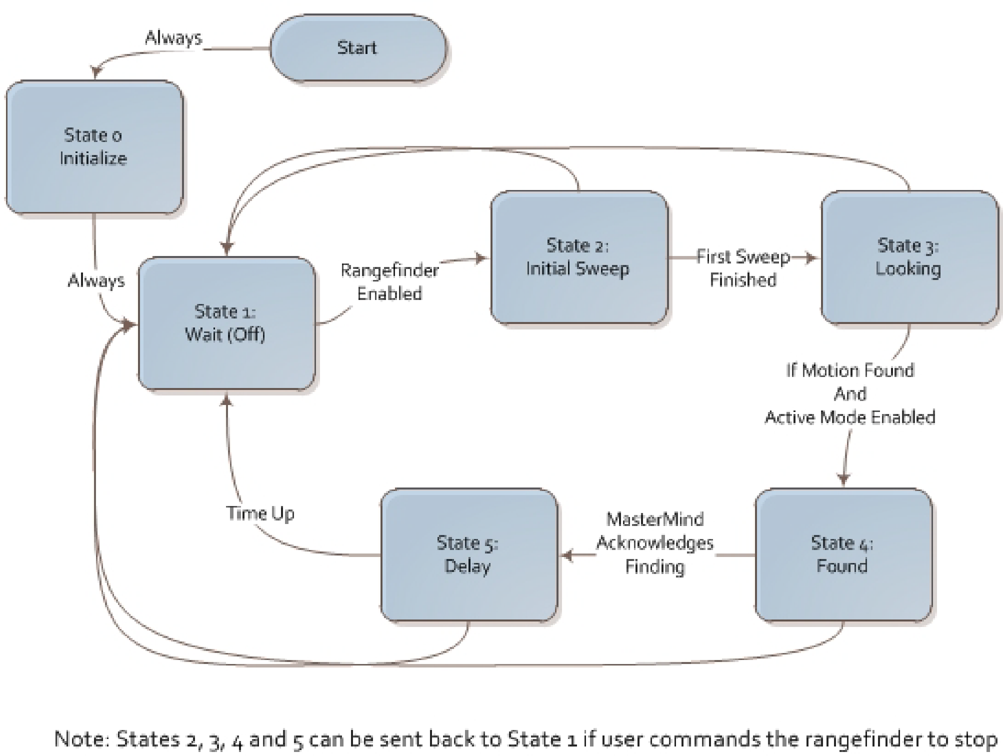


Figure 77. State Transition Diagram for Rangefinder Task.

Eyes

The Eyes task is responsible for controlling the Simulated Approach (Eyes) sub-system by lighting the LEDs in accordance with commands from Mastermind. The control of LEDs is very simple with Arduino and thus the State Transition Diagram is also very simple; it is provided in Figure 78.

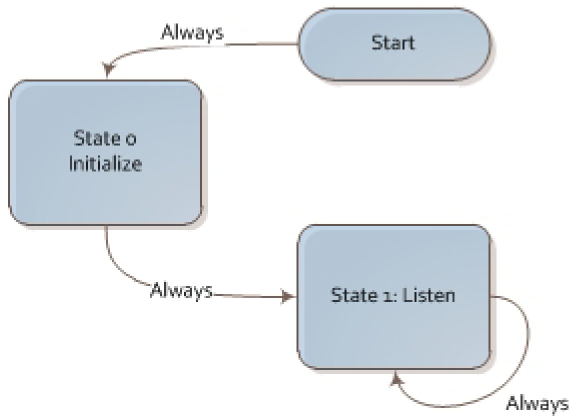


Figure 78. State Transition Diagram for Eyes Task.

Shaker

The Shaker Task controls the servos which operate the shaking mechanism for the Simulated Approach (Motion) sub-system. The system commands the servo to move to a certain location and after a short interval commands it to move to another one. This process is repeated over and over resulting in a shaking motion as desired. The State Transition Diagram is provided in Figure 79.

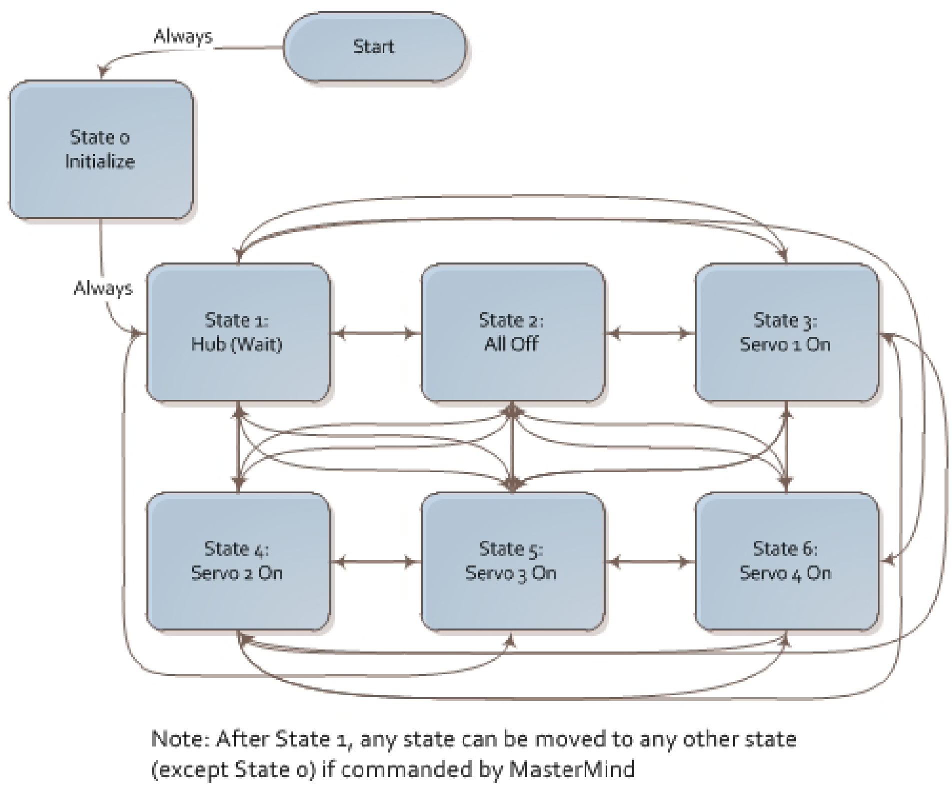


Figure 79. State Transition Diagram for Shaker task.

Ch 5. Product Realization

This section details the final product produced as a result of the project. Some aspects are modified from the planned design as is to be expected during implementation. Changes were made throughout the process at the request of the sponsor and out of practical need.

System Overview

Overall, our final system prototype, in Figure 80 below, turned out to be an impressive machine and has great potential of actually scaring deer. The system itself requires very little input from the user and will only need maintenance if someone or something trips over once of the devices or wires. Although we are very proud of our final product we do believe that there can be several improvements to the design so that it has a much better success rate when operating.

Figure 80. Complete simulated approach prototype as demonstrated.

Simulated Approach (Movement)

Manufacturing

A single simulated approach unit is mounted to a 20x12x¾ inch piece of plywood. As for the speaker and the servo, they are mounted to the platform via screws. All holes were measured, marked, and created using an electric power drill and high speed steel drill bits. The hinge however, needed a predrilled hole so that the threaded hinge would screw into the wood securely. In order to attach the fake plants to the platform, a ½ inch drill bit was used to make 2 holes in the board that were a tight fit. The plants were then permanently attached to the wood by hot glue. The servo itself was attached by two stainless steel screws on opposites corners so that it remains level and stabilized. The entire constructed product can be viewed in Figure 81 below.

Figure 81. Photo showing simple construction methods used in Simulated Approach (Motion) sub-system.

In the end, our final design was put together much differently than previously thought. The new features of the final design were the addition of 2 extra fake bushes to further camouflage the system and the rod attached to the shaking plant was originally planned to loop around but it seemed unnecessary and impractical so the newly refined L shaped bend made the entire system much simpler.

As far as future design iterations, we would probably devise an even simpler and more effective attachment method that would need less space to operate. Also, we would add more plants to make the entire unit appear to be a bush. On top of that, a much more secure and watertight cover would have to be implemented on the servo. One of the last ideas for this subsystem would be to expand the number of units for a target area so as to increase its overall effective and impact on the deer.

Simulated Approach (Eyes)

Manufacturing

A single eye simulator was manufactured to check for any unforeseen difficulties in manufacturing or assembly. No issues were encountered upon either of these tasks, so the remaining three eye simulators were completed at the same time. The mounting holes for the UHMW plate were drilled precisely per manufacturing drawings, as accuracy was needed to ensure that the mounting U-bolts would fit. The LED mounting holes on the UHMW plate were created with loose tolerances as location of the LED's was not strictly important. The acrylic plates were manufactured carefully per manufacturing drawings to ensure that they would be concentric with the UHMW plate mounting holes. All drilled holes used a standard drill press and high speed steel drill bits. The wooden stakes were sanded down using a belt sander to form an arbitrary point capable of being hammered into soft earth.

To assemble the eye simulators, the acrylic and UHMW plates were stacked and the mounting holes aligned. The mounting U-bolts were inserted through the mounting holes, with the threaded parts of the U-bolts closest to the acrylic panels. The U-bolt nuts were then placed on the U-bolts and then tightened loosely. The stake was then placed through each U-bolt opening. The panels were then positioned near the top of the stake, and the U-bolt nuts were secured until tight. 10mm LEDs were press fit into the back of the UHMW plates.

Once assembled, each eye system was wired and soldered to a standard two-wire connector. After soldering, any excess lead length on the LEDs were removed. Plasti-dip was then used to coat the entire system to improve weather resistance, camouflage the system, and prevent electrical shorting between the LED leads.

Modifications

Upon assembly, it was noted that the acrylic panels may not be necessary to create the desired 'eye glowing' effect. Instead, diffused LEDs, which create an indirect glow, were used in the system, greatly simplifying the design and negating the need of acrylic panels.

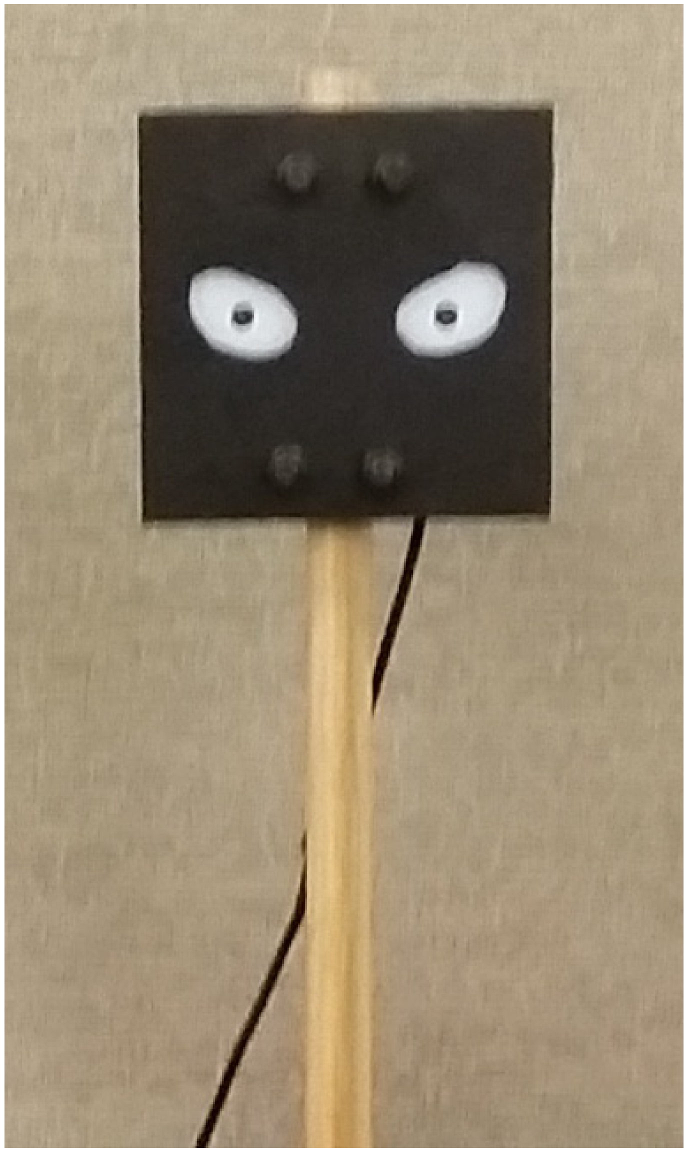


Figure 82: A completed predator 'eye' without acrylic plates

Recommendations

After manufacturing this system, we believe that using less material would help simplify the design and lower the costs. Since the majority of the UHMW plate is not actually used, its dimensions (including thickness) could be reduced, lowering material costs without sacrificing the effect of the system. Additionally, HDPE could replace the UHMW plate for a lower cost and nearly equally weather resistant system.

Rangefinder Targeting System

Manufacturing

The rangefinder mounting system consists of two sheet metal parts, a U-bolt, servo, a stake, and the rangefinder unit. The sheet metal parts were manufactured from one larger piece of sheet metal. All holes were measured, marked, and located with a spring punch, which were then drilled using a standard drill press and high speed steel drill bits. The larger hole on the rangefinder holder part was drilled using a 1 - 1/8" hole saw. The square servo slot was cut using a pneumatic air tool with an abrasive cut-off wheel attachment. Due to the nature of the tool and the shape of the servo slot, the metal outside of the cut area was affected. However, this will only have minor aesthetic effects on the part. The large sheet metal piece was then cut on the shear, removing each independent sheet metal part from the large parent sheet. Each part was then bent to a right angle per manufacturing drawings on the sheet metal brake. See appendix 5 for manufacturing drawings detailing dimensions and configurations.

The range finder unit was then assembled. The servo was inserted into the servo holder sheet metal part. Initially, the servo was to be secured using small screws and nuts, however, it was discovered that zip-ties were very effective. A U-bolt was attached to the servo holder sheet and a stake was slid through the U-bolt.

Figure 83: Completed Rangefinder Holder showing servo mounting holes

The rangefinder holder was secured to the servo head wheel with two #4 304 stainless screws. The large hole of the rangefinder holder was sandwiched between the 90 degree PVC elbow and the rangefinder unit. The rangefinder unit and the PVC elbow were then threaded together. Hot glue was then placed around the connection to ensure that it was sealed and to help prevent rotation of the rangefinder unit.

Recommendations

A more precise fit between the rangefinder holder and rangefinder unit would allow the system to be more easily assembled. A larger amount of hot glue than anticipated was used to help stabilize the rangefinder unit. Using a more precisely threaded elbow and using rubber washers between the mating surfaces would allow the system to be more stable and dampen vibration from the servo motor. This would make the rangefinder system more accurate.

Electronics

Manufacturing

The installation of the electronic components proved to be more tedious than originally expected. Most of the custom boards purchased required assembly and intricate soldering which we were not expecting. After a few trials, however, they were assembled and verified to work. Figure 84 shows the testing of several of the electronic boards by generating sound with sample code.

Figure 84. Verification setup to ensure proper assembly of the Wave Shield and Amplifier board.

Modifications

At some point, however, during the final assembly process, the Wave Shield was damaged beyond repair and ceased to function. This may have been due to faulty wiring or overpowering the output. As a result of the failure, the audio portion of the Actual Approach concept had to be eliminated from the final design project.

Additionally, the software was encountering numerous problems during testing. These issues were suspected to be due to insufficient RAM memory available on the Arduino Uno microcontroller boards. In order to provide sufficient memory, the microcontroller was upgraded to the Arduino Mega – an increase from 2kb of RAM to 8kb. This change would prove to solve the issues experienced. This upgrade also increased the available PWM output ports, enabling the elimination of the PWM driver planned for the eyes and Shaker servos.

The PIR sensor also proved to be difficult to implement as the microcontroller had a difficult time reading the signal from the sensor and thus would miss motion sensed. This trouble and the success of the Rangefinder system lead us to shift motion sensing to the rangefinder and eliminate the PIR sensor and Sprinkler sub-systems that had been planned.

These changes resulted in drastic wiring simplifications. The updated wiring diagram is provided in Figure 85.

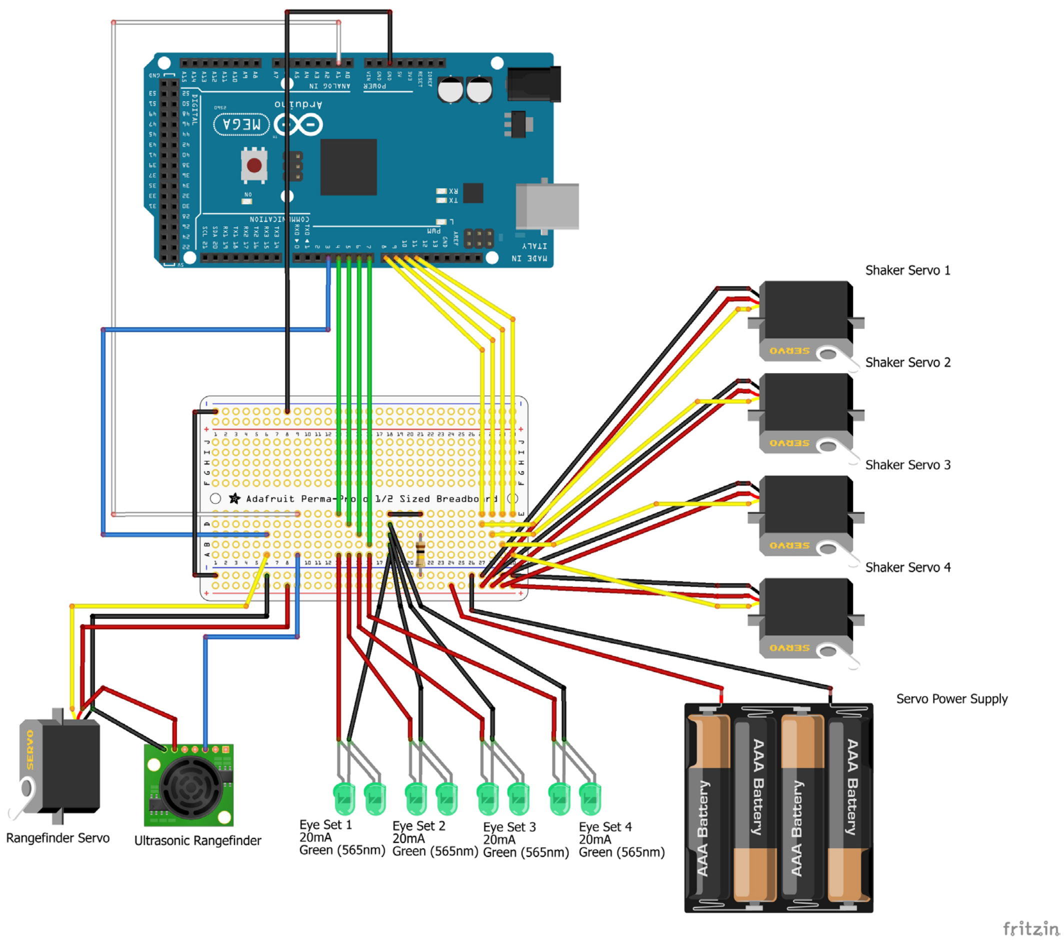


Figure 85. Final wiring diagram for complete system incorporated required changes.

Recommendations

Future manufacturing should include a custom designed microcontroller board including all of the required hardware. This will limit the unnecessary components and hardware included in the Arduino system to provide versatility. Once the system is finalized, a customized board will be practical. This will also allow for the components to be permanently attached to the board to minimize the risk of wires coming loose.

Ch 6. Design Verification

The testing of the system was conducted in a large, open room to eliminate environmental noise and provide more accurate test results. The tests conducted are as follows:

Detection Range (Distance)

Method

The maximum range of the rangefinder was measured by moving a large whiteboard, positioned perpendicular to the measurement angle of the stationary rangefinder so that the whiteboard appears very large to the rangefinder. The whiteboard was started close to the sensor and periodically moved back. The last distance to register a change in the distance reading was marked and measured.

Results

The max observable distance by the rangefinder was found to be 24 feet, 10 inches.

Detection Range (Angle)

Method

The angle of view of the rangefinder was measured by moving the large whiteboard, oriented similarly, towards to center of view of the rangefinder at a distance of 10 feet away, starting from outside of the field of view. The object was moved inwards until the rangefinder registered its presence at which point, the location was marked and the angle was calculated using trigonometry.

Results

The max half-angle was measured to be 49 degrees providing a full angle of 98 degrees.

Detection Area

Method

The detection area was calculated using the measurement from the two Detection Range tests.

Results

The effective detection area was found to be 527 square feet.

Shaker Amplitude

Method

The shaker amplitude was measured by placing a ruler adjacent to the connection point where the plant meets the wire. The plant was activated and the distance traveled was observed.

Results

The amplitude of the shaker was measured to be 3/16” at the connection point of the plant and the wire.

Shaker Frequency

Method

The frequency of the shaker was measured by counting the number of cycles performed over ten seconds. One team member counted the cycles while another timed using a stopwatch and yelling “Stop” when the time was up. This test was performed three times to ensure accurate readings.

Results

A total of 18 cycles were observed during a ten second period implying a shaking frequency of 1.8 Hz.

Angle of Visibility of Eyes

Method

The angle of visibility of the yes sub-system was performed by a team-member standing ten feet away from the eye along the center line and slowly walking sideways until he could no longer discern whether the LEDs were on or not. Once he could no longer see the lit LED, the location was marked and the angle was measured using trigonometry. The same distance was assumed on the other side.

Result

The angle of visibility of the Eyes sub-system was found to be 150 degrees.

Distance of Visibility of Eyes

Method

The distance of visibility of the eyes system was measured by a team member starting far away from the eyes, along the centerline and moving towards the lights until he could see that the lights were lit. Once the lights could be seen, the location was marked and the distance was measured. This test was performed by each of the team members and during the daylight.

Result

The distance of visibility of the eyes were found to be 16 feet.

Area of Visibility of Eyes

Method

The area of visibility of the eyes was calculated using the results from both of the visibility tests performed on the eyes.

Result

The area of visibility of the eyes were found to be 335 square feet.

Sound Level of Shakers

Method

The sound level of the shaker was measured using an app downloaded from the Google Play store and installed on a team-member's smart phone. The decibel level was measured at three distances away from the shaker system: 1 ft, 5 ft, and 15 ft.

Results

The sound level was measured to be 75 dB at 1 foot away, 65 dB at 5 feet away and 57 dB at 15 feet away.

Ch 7. Conclusion

Simulated Approach (Movement)

During our testing phase of the project, our system had it bugs such as faulty soldering and minor software glitches; however the simulated approach turned out to be a big surprise. The only downside is that due to unknown complications we were unable to get the speaker system operational. In our early testing phases we were able to get the speakers to emit sound, but when the whole system connected, we believe the wave amplifier board shorted out due to improper wiring or circuit design. One potential fix could be to buy a new board because the one we had could have been faulty. Although, we believe the best solution would be to redesign the circuit so that the speakers and board are receiving the correct amount of power.

On the bright side, the shakers turned out much better than expected because we were unsure if the servos could support the weight of the fake plants. They very closely mimic the presence of a predator or something close by. The only potential problem with the shaker system could be that when triggered, the servos are almost as loud as the bushes shaking themselves. Therefore, the tester would not really know if it was the shaking bush sound or servo motor sound that scared the deer away.

Simulated Approach (Eyes)

After testing, the team noticed that the LED leads were surprisingly fragile and could fail easily, especially when bent multiple times. We advise to move LEDs as little as possible and to make sure that the connecting wires do not stress the leads and lead connections. Additional LEDs have been supplied in case they are needed for replacement. Future designs of the system should include a smaller plate as well as a more robust LED system. A box mounted onto the back of the plate would help increase the weather resistance of the system and allow for the LED leads to be connected to dedicated connectors on the box. This would help ensure that the LED leads do not break.

Rangefinder Targeting System

After testing, the team noticed that the hobby servo used for the rangefinder targeting system would occasionally not operate. By manually rotating the servo, the servo would begin to automatically start functioning properly again. This 'sticking' of the servo system was investigated but the team was unable to figure out what the cause of this issue was. A possible cause is that the center of gravity of the rangefinder unit is off center (not directly in line with the servo shaft), causing an uneven moment load on the servo shaft. This uneven loading may be causing friction that the internal gearing of the servo is unable to overcome. A solution to this problem is redesigning the rangefinder holder unit to place the center of gravity of the system directly under the servo shaft. Another solution would be to obtain a more heavy duty servo motor designed for higher loading scenarios. However, the electrical specifications of the new motor should be similar to the existing servo.

Another minor issue noticed with the rangefinder system is that it tends to shake as the servo rotates the rangefinder unit, likely decreasing the accuracy of the system. Ensuring that the stake is planted in firm earth is an easy way to help mitigate this issue. Future ways to solve this issue would be to redesign the rangefinder holder to use a second u-bolt to help secure it to the stake. A more complex way of solving this issue would be to buy a servo motor capable of speed control, and program controlled accelerations of the unit to prevent many sudden stops and starts.

The team sees the ultrasonic rangefinder system as a very plausible method for tracking invading wildlife for future advanced deterrence systems. Being able to track the movement of wildlife opens the door to many new and promising solutions to help combat habituation.

Electronics

The electronics system is adequate for the tasks performed and offers some capacity for additional components. The available RAM and I/O pins from the Arduino Mega make this possible.

Additional methods for switching sound to multiple speakers should be attempted, possibly using relays to switch the individual speakers on or off. Alternatively, small, low powered microcontrollers could be positioned at each speaker to play the sound directly to the speaker upon the instruction of a larger controller, though this option would be more costly.

References

Beier, Paul, and Dale R. McCullough. *Factors Influencing White-Tailed Deer Activity Patterns and Habitat Use*. Wiley on Behalf of the Wildlife Society, n.d. Web. 1990. http://www.jstor.org/stable/3830629

Belant, Jerrold L.; Seamans, Thomas W.; and Tyson, Laura A. *Evaluation of Electronic Frightening Devices as White-Tailed Deer Deterrents*. Proceedings of the Eighteenth Vertebrate Pest Conference (1998). Paper 3, Web. http://digitalcommons.unl.edu/vpc18/3

D'angelo, Gino J., Albert R. De Chicchis, David A. Osborn, George R. Gallagher, Robert J. Warren, and Karl V. Miller. *Hearing Range of White-Tailed Deer as Determined by Auditory Brainstem Response*. Journal of Wildlife Management 71.4 (2007): 1238-242. Wiley on Behalf of the Wildlife Society. Web. http://www.jstor.org/stable/4496185

DeNicola, Anthony J. *Managing White-tailed Deer in Suburban Environments: A Technical Guide*. Ithaca, NY: Cornell Cooperative Extension, 2000. Web. http://www.ci.carrboro.nc.us/pd/PDFs/suburbandeer.pdf

Mattern, Vikki. "Ask Our Experts." *Mother Earth News*. Mother Earth News, July 2009. Web. Feb. 2014. http://www.motherearthnews.com/organic-gardening/deer-repellent-zb0z1208zmat.aspx

Murphy, Brian. *What Do Deer See?* Quality Deer Management Association. QDMA, 2012. Web. http://www.qdma.com/corporate/what-do-deer-see

Robot Shop. N.p., n.d. Web. http://www.robotshop.com/en/sensors.html

Appendices

1. Gantt Chart
2. QFD
3. Pugh Matrices
4. Decision Matrices
5. Technical Drawings
6. Calculations
7. Final Cost Ledger
8. Program Source Code
9. Operators Manual
10. Product Specification Sheets

ID	WBS	Task Name	Duration	Start
1	1	Final Design Report	0 days	Fri 5/2/14
2	2	Critical Design Review w/ Sponsor	0 days	Mon 5/12/14
3	3	Polish Critical Design Review Presentation	5 days	Fri 5/2/14
4	**4**	**Construction**	**48 days**	**Mon 5/12/14**
5	4.1	Order & Receive Parts	9 days	Mon 5/12/14
6	4.2	Construct Mechanical Components	25 days	Tue 5/13/14
7	4.3	Construct Structure	15 days	Tue 5/13/14
8	4.4	Construct Electrical Components	15 days	Tue 5/13/14
9	4.5	Develop Software	21 days	Fri 5/16/14
10	4.6	Assemble System	6 days	Mon 9/22/14
11	4.7	Troubleshoot & Fix Problems	18 days	Mon 9/29/14
12	5	Hardware Demo	0 days	Thu 10/23/14
13	**6**	**Testing**	**17 days**	**Fri 10/24/14**
14	6.1	Test Simulated Approach System	2 days	Wed 10/29/14
15	6.2	Test Actual Approach System	3 days	Fri 10/24/14
16	6.3	Test Sprinkler System	1 day	Fri 10/31/14
17	6.4	Test Documenting Cameras	2 days	Mon 11/3/14
18	6.5	Test PC Interface	1 day	Wed 11/5/14
19	6.6	Test Weatherproofing	1 day	Thu 11/6/14
20	6.7	Field Testing	7 days	Sat 11/8/14
21	7	Prepare for Expo	20 days	Thu 10/23/14
22	8	Senior Design Expo	0 days	Thu 11/20/14
23	9	Prepare Final Report	13 days	Tue 11/18/14
24	10	Final Project Report	0 days	Fri 12/5/14

Project: Deer Busters Gantt Chart
Date: Sat 4/26/14

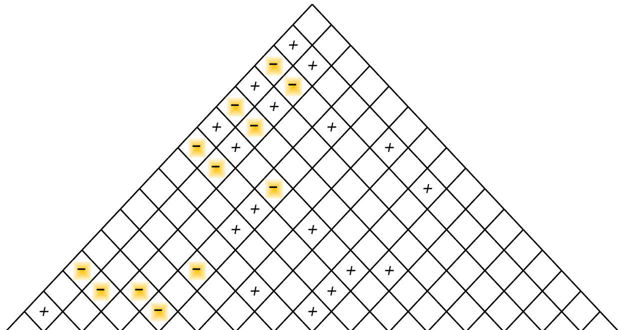
Task	Project Summary	Inactive Milestone	Manual Summary Rollup	Deadline
Split	External Tasks	Inactive Summary	Manual Summary	Progress
Milestone	External Milestone	Manual Task	Start-only	
Summary	Inactive Task	Duration-only	Finish-only	

QFD: House of Quality
Proje Deer Busters
Revision:
Date: 1/23/2014

Correlations	
Positive	+
Negative	−
No Correlation	

Relationships	
Strong	●
Moderate	○
Weak	▽

Direction of Improvemen	
Maximize	▲
Target	◇
Minimize	▼



WHO: Customers

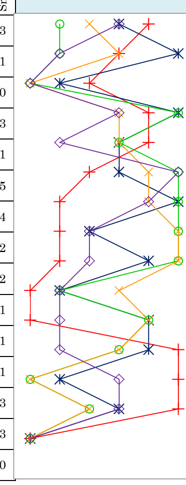
Row #	Weight Chart	Relative Weight	Eventual Customer	Test Engineer	Installer	Purchaser	Maximum Relationship	WHAT: Customer Requirements (explicit & implicit)
1		5%	8	1	1	3	9	Scare Deer Away
2		5%	2	9	1	1	9	Easy to test
3		7%	6	9	1	3	9	Verification of Results
4		7%	7	8	1	3	9	Durable and Weatherproof
5		11%	9	7	4	4	9	Safe for Humans
6		6%	7	5	1	3	9	Safe for Animals
7		6%	2	1	1	8	9	Marketable Components
8		7%	8	8	1	1	9	Notification of Malfunction
9		5%	5	1	1	5	9	Use Minimal Resources
10		8%	6	3	5	1	9	Device Not Activated by Huma
11		6%	7	7	1	2	9	Minimal Maintanence
12		6%	7	5	1	3	9	Reliable
13		7%	6	1	1	8	9	Low Cost
14		8%	8	6	3	2	9	Not Obnoxious
15		5%	1	1	2	5		Utilize JumpSport Patent

HOW: Engineering Specifications

Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Direction of Improvement																
Row #	Reduce deer invasions	Reduce non deer invasions	Effective Area	Initial Cost	Results Documented	Time between interactions	Is not activated by humans	Ongoing Costs	Max voltage: 120 V	No high energy projectiles	Interaction required for operation	Shock exposure	Operation Temperature	Sound Requirement	User is notified of maintenance need	
1	●	○	●			▽					▽		▽		▽	
2							▽				●				▽	
3					●											
4	▽	▽				○							●			
5					▽		●		○	●	○	○		▽		
6									○	●		●		▽		
7	●	○	○	▽		●	○	○	▽		●	▽			○	
8															●	
9			▽			○		●	▽		○					
10							●									
11						●		○							●	
12	○	▽				●							○			
13				●	▽	▽		○	▽		▽				▽	
14						○	○				○	▽	▽	●	▽	
15																

NOW: Current Product Assesment - Customer Requirements

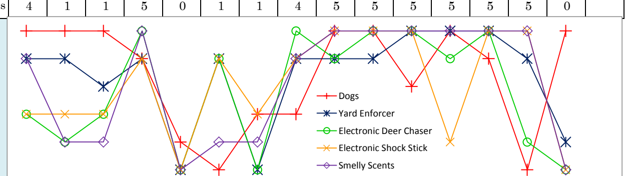
Row #	Dogs	Yard Enforcer	Electronic Deer Chaser	Electric Shock Stick	Smelly Scents
1	4	3	1	2	3
2	3	5	1	3	1
3	2	1	0	0	0
4	4	5	5	3	3
5	4	3	3	3	1
6	2	3	5	4	5
7	1	5	5	4	4
8	1	2	5	5	2
9	1	4	5	5	2
10	0	1	1	3	1
11	0	4	4	4	1
12	5	4	3	3	1
13	5	1	0	0	3
14	5	3	2	2	3
15	0	0	0	0	0



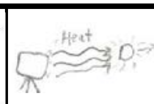
Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
HOW MUCH: Target	90% Reduction	60% Reduction	1000 square feet	$1,000	95% of time	4 Weeks	95% of time	$5 per month	120 V	.5 J	Fully Autonomis	5 mA	32-120°F	65 dB at 15 feet	User notified	
Max Relationship	9	3	9	9	9	9	9	9	3	9	9	9	9	9	9	
Technical Importance Rating	122.7	45.71	67.75	67.66	83.03	238.1	219.1	103.4	68.87	153.2	183.2	102	96.44	91.65	158.4	0
Relative Weight	7%	3%	4%	4%	5%	13%	12%	6%	4%	9%	10%	6%	5%	5%	9%	0%
Weight Chart																

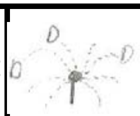
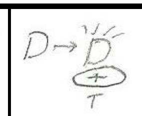
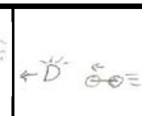
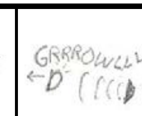
Current Product Assesment - Engineering Specifications

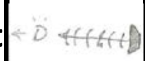
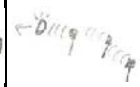
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Dogs	5	5	5	4	1	0	2	2	5	5	3	5	4	0	5	
Yard Enforcer	4	4	3	4	0	4	0	4	4	4	5	5	5	4	1	
Electronic Deer Chaser	2	1	2	5	0	4	0	5	4	5	5	4	5	1	0	
Electric Shock Stick	2	2	2	4	0	4	2	4	5	5	5	1	5	5	0	
Smelly Scents	4	1	1	5	0	1	1	4	5	5	5	5	5	5	0	



Template Revision: 0.9 Date: 4/23/2010
Christopher Battles

Criteria \ Concept	 1	 2	 3	 4	 5	 6	 7	 8	 9	 10
Scare Deer Away	-	n/a	-	-	-	-	-	-	S	S
Easy to Test	-	n/a	S	S	S	-	S	S	S	S
Durable and Weatherproof	S	n/a	S	S	+	+	+	+	S	+
Safe For Humans	+	n/a	+	+	+	+	+	+	S	+
Safe For Animals	+	n/a	+	+	+	+	+	+	S	+
Marketability	+	n/a	+	+	-	-	-	-	-	-
Minimal resources & Maintenance	S	n/a	+	+	+	+	+	+	+	S
Reliability	-	n/a	S	S	-	+	+	+	-	+
Low Cost	S	n/a	-	S	+	+	-	-	-	+
Not Obnoxious	+	n/a	+	S	S	+	S	+	+	S
Σ+	4	0	5	4	5	7	5	6	2	5
Σ-	3	0	2	1	3	3	3	3	3	1
Σ	1	0	3	3	2	4	2	3	-1	4

Concept / Criteria	 11	 12	 13	 14	 15	 16	 17	 18	 19	 20
Scare Deer Away	S	S	S	S	-	+	S	+	-	-
Easy to Test	S	S	S	-	+	-	-	S	+	+
Durable and Weatherproof	S	S	S	+	S	S	S	S	S	S
Safe For Humans	+	+	S	S	+	S	+	-	+	+
Safe For Animals	+	+	S	-	+	+	+	-	+	+
Marketability	+	+	S	-	+	S	S	S	+	+
Minimal resources & Maintenance	S	S	+	+	+	S	S	+	+	+
Reliability	S	S	-	-	S	S	S	S	S	S
Low Cost	S	S	S	+	S	-	-	S	S	+
Not Obnoxious	S	S	+	+	S	S	S	S	-	-
Σ+	3	3	2	4	5	2	2	2	5	6
Σ-	0	0	1	4	1	2	2	2	2	2
Σ	3	3	1	0	4	0	0	0	3	4

Concept / **Criteria**	 21	 22	 23							
Scare Deer Away	-	-	-							
Easy to Test	+	S	-							
Durable and Weatherproof	S	S	S							
Safe For Humans	+	+	+							
Safe For Animals	+	+	+							
Marketability	+	+	-							
Minimal resources & Maintenance	+	+	-							
Reliability	S	S	S							
Low Cost	S	S	S							
Not Obnoxious	S	S	-							
Σ+	**5**	**4**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Σ-	**1**	**1**	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Σ	**4**	**3**	**-3**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Concept / Criteria	Trip Wire 1	Motion Sensor 2	Heat Sensor 3	Contact Sensor 4	Rangefinder 5	Camera 6	Noise Traps 7	Kinect Sensor 8	Smell Sensor 9	Food Scale 10
Not Obnoxious	n/a	S	S	S	S	S	-	S	S	-
Durable	n/a	+	+	-	+	S	+	-	S	S
Weatherproof	n/a	-	-	S	-	-	S	-	-	S
Safe	n/a	+	+	+	+	+	S	+	+	+
Marketable	n/a	+	+	S	+	+	-	+	+	-
Minimal Resource Usage	n/a	-	-	-	-	-	S	-	-	-
Not Activated by Humans	n/a	-	-	+	-	+	+	+	+	+
Minimal Maintenance	n/a	+	+	S	+	+	S	+	+	S
Reliable	n/a	+	+	S	+	+	S	+	-	S
Low Cost	n/a	+	S	+	+	-	S	-	-	-
Σ+	**0**	**6**	**5**	**3**	**6**	**5**	**2**	**5**	**4**	**2**
Σ-	**0**	**3**	**3**	**2**	**3**	**3**	**2**	**4**	**4**	**4**
Σ	**0**	**3**	**2**	**1**	**3**	**2**	**0**	**1**	**0**	**-2**

Concept / Criteria	Camera w/ sensor 1	Counter 2	Computer database 3	24/7 camera 4	Measure food eaten 5	Measure power use 6	Notify directly 7	8	9	10
Verification of results	n/a	-	-	S	-	-	+			
Reliable	n/a	+	S	S	-	-	+			
Durable and weatherproof	n/a	+	-	S	+	+	+			
Low cost	n/a	+	-	+	+	+	-			
Σ+	0	3	0	1	2	2	3	0	0	0
Σ-	0	1	3	0	2	2	1	0	0	0
Σ	0	2	-3	1	0	0	2	0	0	0

Concept / Criteria	Notify directly 1	Sounds 2	Light 3	Backup Function 4	5	6	7	8	9	10
Durable and weatherproof	n/a	S	S	S						
Marketable componets	n/a	-	-	-						
Notifation of malfunction	n/a	-	-	-						
Uses minimal resources	n/a	-	-	-						
Minimal maintenance	n/a	S	S	S						
Reliable	n/a	+	-	S						
Low cost	n/a	+	+	-						
Not obnoxious	n/a	-	S	-						
Σ+	0	2	1	0	0	0	0	0	0	0
Σ-	0	4	4	5	0	0	0	0	0	0
Σ	0	-2	-3	-5	0	0	0	0	0	0

Protection Methods

	Concepts	Weight	Air Cannon		Shield		Move Plant		Water Wall		Non-Targeted Spray		Targeted Spray		Stationary Robot		Actual Approach		Specific Sounds (Raaaar)		Flashing Lights		Directed Sound		Simulated Approach	
Criteria	Protect Plants	0.15	6	0.9	5	0.75	3	0.45	5	0.75	7	1.05	9	1.35	5	0.75	9	1.35	7	1.05	2	0.3	6	0.9	7	1.05
	Easy to Test	0.05	8	0.4	7	0.35	3	0.15	7	0.35	8	0.4	8	0.4	8	0.4	4	0.2	8	0.4	8	0.4	8	0.4	5	0.25
	Safe	0.2	9	1.8	7	1.4	5	1	9	1.8	9	1.8	9	1.8	7	1.4	5	1	9	1.8	9	1.8	8	1.6	8	1.6
	Marketable	0.15	7	1.05	4	0.6	1	0.15	7	1.05	8	1.2	9	1.35	6	0.9	7	1.05	9	1.35	6	0.9	6	0.9	7	1.05
	Minimal Resources Used	0.05	8	0.4	9	0.45	6	0.3	3	0.15	6	0.3	7	0.35	7	0.35	8	0.4	8	0.4	7	0.35	7	0.35	8	0.4
	Minimal Maintenance	0.1	7	0.7	6	0.6	4	0.4	8	0.8	7	0.7	6	0.6	5	0.5	5	0.5	7	0.7	8	0.8	8	0.8	7	0.7
	Reliable	0.15	7	1.05	8	1.2	3	0.45	8	1.2	7	1.05	6	0.9	5	0.75	5	0.75	8	1.2	8	1.2	8	1.2	8	1.2
	Low Cost	0.05	7	0.35	8	0.4	3	0.15	7	0.35	7	0.35	6	0.3	4	0.2	3	0.15	7	0.35	8	0.4	7	0.35	6	0.3
	Not Obnoxious	0.1	6	0.6	2	0.2	2	0.2	6	0.6	6	0.6	6	0.6	6	0.6	5	0.5	3	0.3	2	0.2	4	0.4	4	0.4
	Sum	1	7.25		5.95		3.25		7.05		7.45		7.65		5.85		5.90		7.55		6.35		6.90		6.95	
	%		9.28%		7.62%		4.16%		9.03%		9.54%		9.80%		7.49%		7.55%		9.67%		8.13%		8.83%		8.90%	

Protection Methods Revised

	Concepts	Weight	Air Cannon (Targeted)		Water Wall		Non-Targeted Spray		Targeted Spray		Stationary Robot		Specific Sounds (Raaaar)		Simulated Approach	
Criteria	Protect Plants	0.15	6	0.9	5	0.75	7	1.05	9	1.35	5	0.75	7	1.05	7	1.05
	Easy to Test	0.05	8	0.4	7	0.35	8	0.4	8	0.4	8	0.4	8	0.4	5	0.25
	Safe	0.2	9	1.8	9	1.8	9	1.8	9	1.8	7	1.4	9	1.8	8	1.6
	Marketable	0.15	7	1.05	7	1.05	8	1.2	9	1.35	6	0.9	9	1.35	7	1.05
	Minimal Resources Used	0.05	8	0.4	3	0.15	6	0.3	7	0.35	7	0.35	8	0.4	8	0.4
	Minimal MaintenanceRequired	0.1	7	0.7	8	0.8	7	0.7	6	0.6	5	0.5	7	0.7	7	0.7
	Reliable	0.15	7	1.05	8	1.2	7	1.05	6	0.9	5	0.75	8	1.2	8	1.2
	Low Cost	0.05	7	0.35	7	0.35	7	0.35	6	0.3	4	0.2	7	0.35	6	0.3
	Not Obnoxious	0.1	6	0.6	6	0.6	6	0.6	6	0.6	6	0.6	3	0.3	4	0.4
	Sum	1	7.25		7.05		7.45		7.65		5.85		7.55		6.95	
	%		14.57%		14.17%		14.97%		15.38%		11.76%		15.18%		13.97%	

Detailed Protection Methods

Criteria

Concepts	Weight	Air Cannon		Air Bazooka		Water Wall		360° Full Area Spray		120° Area Stream	
Protect Plants	0.25	3	0.75	1	0.25	6	1.5	3	0.75	8	2
Marketability	0.2	6	1.2	8	1.6	5	1	6	1.2	8	1.6
Simplicity	0.2	7	1.4	4	0.8	8	1.6	8	1.6	6	1.2
Low Cost	0.05	5	0.25	5	0.25	6	0.3	8	0.4	5	0.25
Reliability	0.2	7	1.4	5	1	8	1.6	6	1.2	6	1.2
Minimal Resources	0.1	9	0.9	9	0.9	3	0.3	5	0.5	5	0.5
Sum	1	5.90		4.80		6.30		5.65		6.75	
%		6.41%		5.21%		6.84%		6.14%		7.33%	

Concepts	Weight	Targeted Stream		AirCrow		Waving Flags		Pop-up Object		Rattling Wavy Object	
Protect Plants	0.25	8	2	4	1	4	1	6	1.5	5	1.25
Marketability	0.2	8	1.6	7	1.4	7	1.4	6	1.2	6	1.2
Simplicity	0.2	5	1	7	1.4	6	1.2	8	1.6	8	1.6
Low Cost	0.05	3	0.15	8	0.4	5	0.25	7	0.35	6	0.3
Reliability	0.2	5	1	8	1.6	5	1	6	1.2	7	1.4
Minimal Resources	0.1	5	0.5	8	0.8	8	0.8	6	0.6	7	0.7
Sum	1	6.25		6.60		5.65		6.45		6.45	
%		6.79%		7.17%		6.14%		7.01%		7.01%	

Concepts	Weight	Noisemakers		Targeted Speakers		Stationary Speakers		Simulated Approach Speakers		Simulated Approach Mechanical Devices	
Protect Plants	0.25	5	1.25	7	1.75	6	1.5	8	2	8	2
Marketability	0.2	5	1	7	1.4	7	1.4	7	1.4	7	1.4
Simplicity	0.2	8	1.6	4	0.8	7	1.4	4	0.8	3	0.6
Low Cost	0.05	8	0.4	4	0.2	4	0.2	3	0.15	2	0.1
Reliability	0.2	6	1.2	6	1.2	7	1.4	7	1.4	4	0.8
Minimal Resources	0.1	7	0.7	8	0.8	8	0.8	8	0.8	8	0.8
Sum	1	6.15		6.15		6.70		6.55		5.70	
%		6.68%		6.68%		7.28%		7.12%		6.19%	

Detection Methods

	Concepts	Weight	Trip Wire		IR Motion Sensor		Heat Sensor		Contact Sensor		Laser Rangefinder		Sonic Rangefinder		Camera		Kinect	
	Not Obnoxious	0.1	3	0.3	9	0.9	9	0.9	5	0.5	9	0.9	9	0.9	8	0.8	8	0.8
	Durable & Weatherproof	0.1	8	0.8	7	0.7	7	0.7	7	0.7	8	0.8	4	0.4	6	0.6	6	0.6
	Safe	0.2	4	0.8	9	1.8	9	1.8	6	1.2	8	1.6	9	1.8	9	1.8	9	1.8
Criteria	Marketable	0.2	3	0.6	7	1.4	8	1.6	4	0.8	8	1.6	7	1.4	7	1.4	7	1.4
	Minimal Resources Used	0.05	9	0.45	6	0.3	4	0.2	9	0.45	6	0.3	7	0.35	5	0.25	5	0.25
	Not Activated by Humans	0.05	7	0.35	1	0.05	5	0.25	8	0.4	2	0.1	2	0.1	6	0.3	6	0.3
	Minimal Maintenance	0.05	5	0.25	9	0.45	8	0.4	8	0.4	7	0.35	7	0.35	7	0.35	7	0.35
	Reliable	0.2	3	0.6	8	1.6	7	1.4	3	0.6	7	1.4	7	1.4	6	1.2	8	1.6
	Low Cost	0.05	4	0.2	8	0.4	1	0.05	5	0.25	4	0.2	8	0.4	4	0.2	2	0.1
				0		0		0		0		0		0		0		0
	Sum	1	4.35		7.60		7.30		5.30		7.25		7.10		6.90		7.20	
	%		8.21%		14.34%		13.77%		10.00%		13.68%		13.40%		13.02%		13.58%	

Detection Methods Revised

	Concepts	Weight	IR Motion Sensor		Heat Sensor		Laser Rangefinder		Sonic Rangefinder	
Criteria	Not Obnoxious	0.1	9	0.9	9	0.9	9	0.9	9	0.9
	Durable & Weatherproof	0.1	7	0.7	7	0.7	8	0.8	4	0.4
	Safe	0.2	9	1.8	9	1.8	8	1.6	9	1.8
	Marketable	0.2	7	1.4	8	1.6	8	1.6	7	1.4
	Minimal Resources Used	0.05	6	0.3	4	0.2	6	0.3	7	0.35
	Not Activated by Humans	0.05	1	0.05	5	0.25	2	0.1	2	0.1
	Minimal Maintenance	0.05	9	0.45	8	0.4	7	0.35	7	0.35
	Reliable	0.2	8	1.6	7	1.4	3	0.6	5	1
	Low Cost	0.05	8	0.4	1	0.05	3	0.15	8	0.4
	Sum	1	7.60		7.30		6.40		6.70	
	%		27.14%		26.07%		22.86%		23.93%	

Document Results

	Concepts	Weight	Activated Camera		Trigger Counter		24/7 Camera	
Criteria	Verification of Results	0.35	7	2.45	4	1.4	10	3.5
	Reliable	0.3	7	2.1	9	2.7	8	2.4
	Durable & Weatherproof	0.2	8	1.6	9	1.8	8	1.6
	Ease of Use	0.1	9	0.9	9	0.9	4	0.4
	Low Cost	0.05	4	0.2	9	0.45	4	0.2
	Sum	1	7.25		7.25		8.10	
	%		32.08%		32.08%		35.84%	

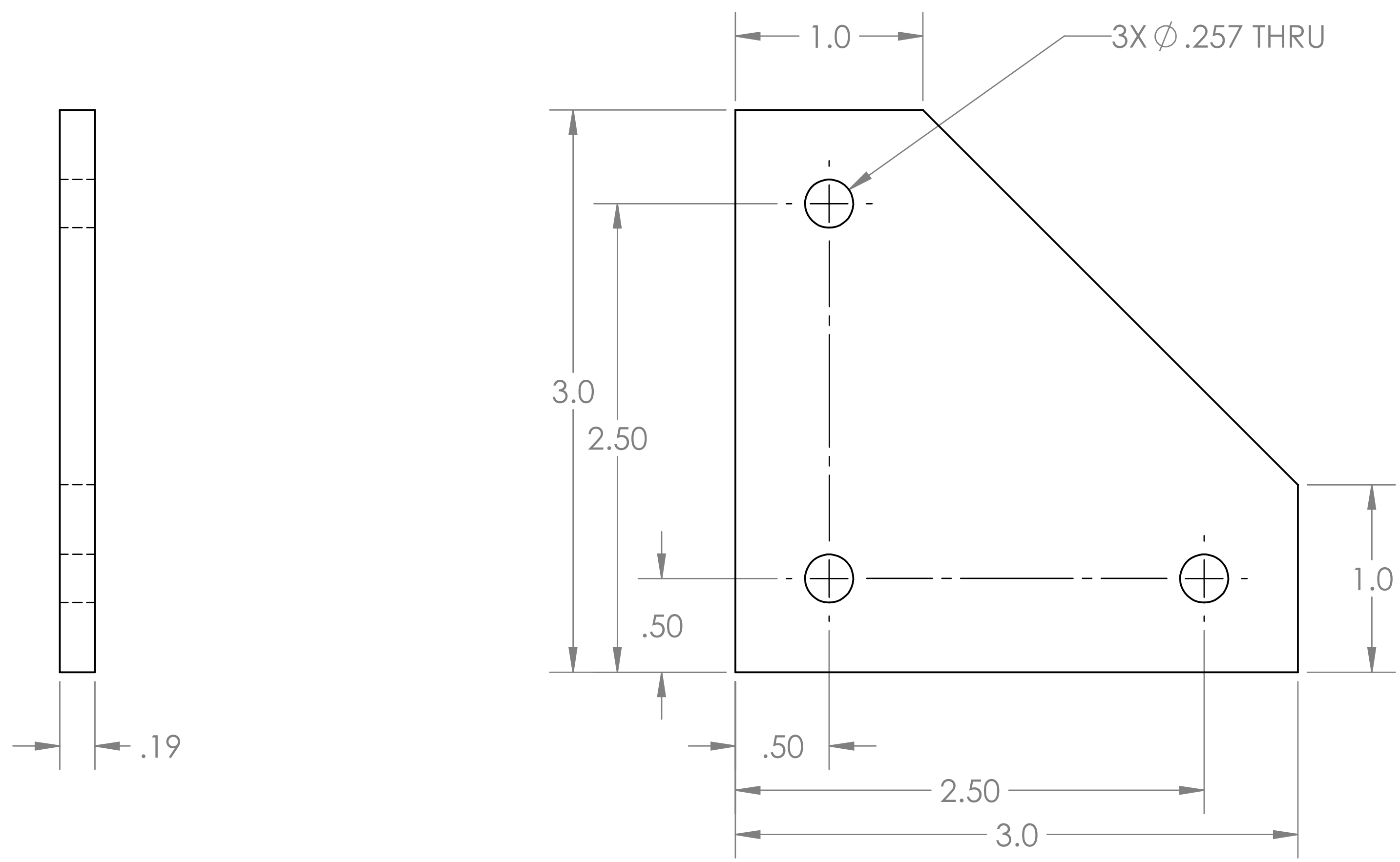
Communicates Maintenance Needs

	Concepts	Weight	Notify Directly		Sounds		Light		Backup Mechanism		Physical Indicator	
Criteria	Durable and Weatherproof	0.1	6	0.6	8	0.8	7	0.7	9	0.9	8	0.8
	Marketable	0.1	7	0.7	6	0.6	6	0.6	5	0.5	6	0.6
	Notification of Malfunction	0.25	9	2.25	5	1.25	5	1.25	3	0.75	7	1.75
	Uses Minimal Resources	0.1	6	0.6	8	0.8	8	0.8	9	0.9	7	0.7
	Reliable	0.3	7	2.1	9	2.7	9	2.7	6	1.8	8	2.4
	Low Cost	0.05	4	0.2	9	0.45	9	0.45	9	0.45	6	0.3
	Not Obnoxious	0.1	4	0.4	5	0.5	8	0.8	9	0.9	9	0.9
				0		0		0		0		0
				0		0		0		0		0
				0		0		0		0		0
	Sum	1	6.85		7.10		7.30		6.20		7.45	
	%		24.95%		25.87%		26.59%		22.59%		27.14%	

Attachment 5

Technical Drawings

1. 90 Fab
2. Approach Stop Frame
3. Approach Stop
4. Control Box Assembly
5. Electronics Box
6. Motor Case (Sheet 1)
7. Motor Case (Sheet 2)
8. Motor Mount Plate
9. Motor Top Plate
10. Rangefinder Assembly
11. Rangefinder Mount Bracket
12. Rangefinder Mount Bracket2
13. Servo Mount
14. Short Bracket
15. Speaker Plate
16. Stake
17. Switch Mount
18. T Fab



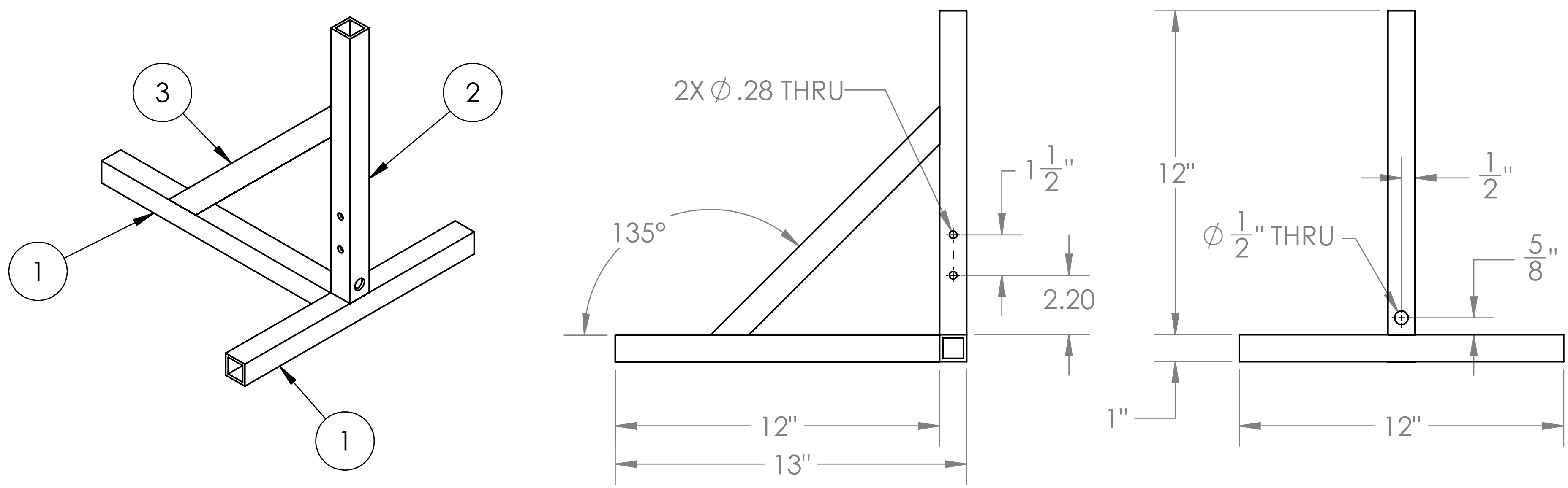
		UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	GWT	4/26/14	
			CHECKED			TITLE:
			ENG APPR.			90-BRACKET
			MFG APPR.			
		INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
		MATERIAL 6061 AL	COMMENTS:			SIZE A / DWG. NO. / REV
NEXT ASSY	USED ON	FINISH				
APPLICATION		DO NOT SCALE DRAWING				SCALE: 1:1 / WEIGHT: / SHEET 1 OF 1

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

5 4 3 2 1

ITEM NO.	DESCRIPTION	MATERIAL	CUT LENGTH	QTY.
1	BASE	1"X1"X0.06" STEEL TUBING	12"	2
2	UPRIGHT	1"X1"X0.06" STEEL TUBING	12"	1
3	SUPPORT BRACE	1"X1"X0.06" STEEL TUBING	12"	1



NOTE: MEMBERS ARE TO BE SECURELY WELDED

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL
STEEL

FINISH

NEXT ASSY | USED ON

APPLICATION

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	GWT	4/27/14
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

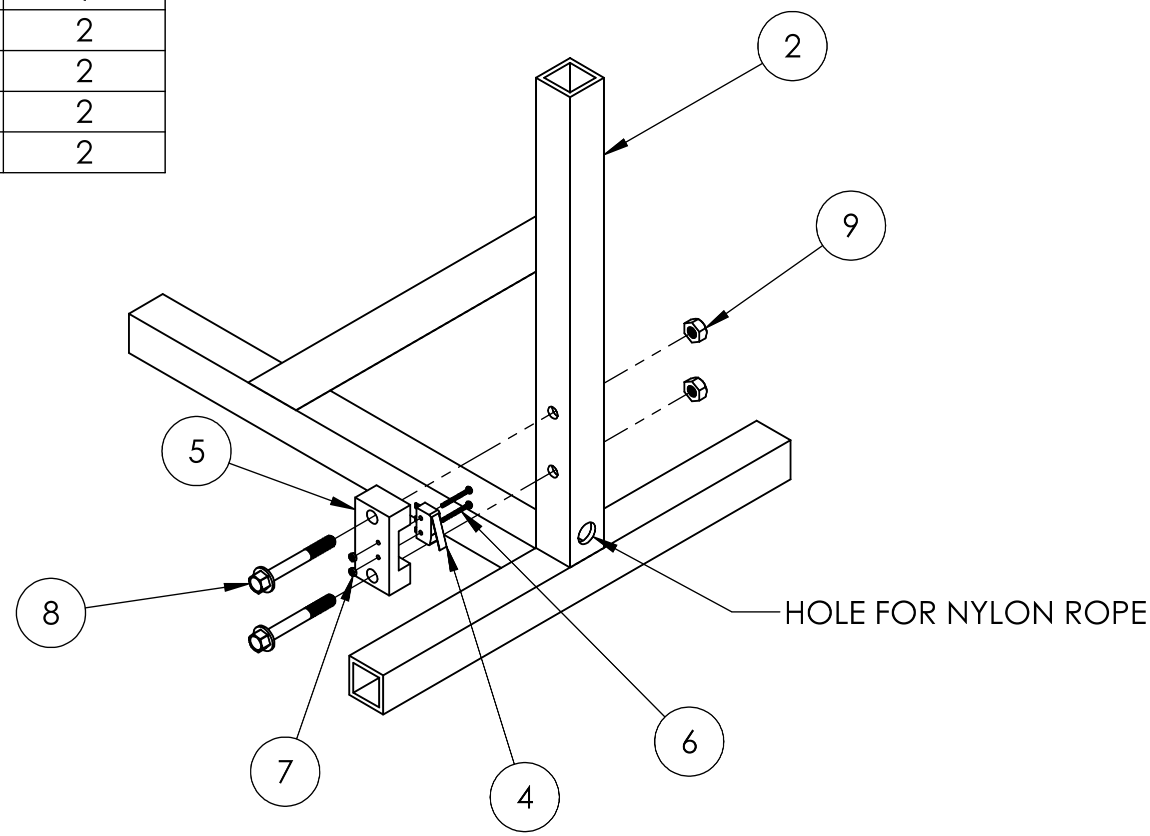
TITLE:
APPROACH STOP FRAME

SIZE A | DWG. NO. | REV

SCALE: 1:6 | WEIGHT: | SHEET 1 OF 1

5 4 3 2 1

ITEM NO.	PART NUMBER	Default/ QTY.
2	APPROACH STOP FRAME	1
4	D2SW-P01L1 LIMIT SWITCH	1
5	Switch Mount	1
6	M2X20MML MACHINE SCREW	2
7	M2 NUT	2
8	1/4"-20X2"L FLANGED NUT	2
9	1/4"-20 NUT	2



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	GWT	4/27/14
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

APPROACH STOP

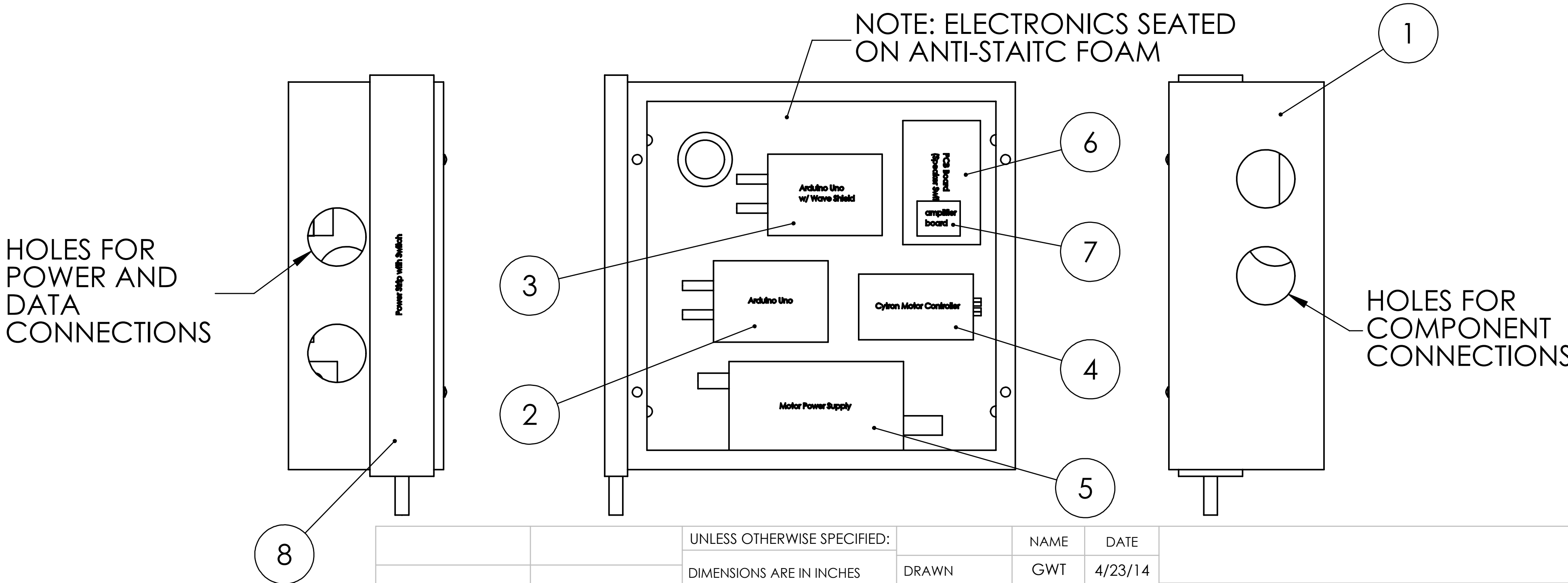
SIZE	DWG. NO.	REV
A		

SCALE: 1:4 WEIGHT: SHEET 1 OF 1

5 4 3 2 1

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Electronics Box	10"x10"x4" Sheet Metal Junction Box	1
2	Arduino Uno Microcontroller	Arduino UNO R3 board with DIP ATmega328P	1
3	Arduino Uno Microcontroller with Wave Shield	Arduino UNO R3 board with DIP ATmega328P + 22kHz 12bit audio output	1
4	Motor Controller	Cytron 5-25VDC 13A Motor Controller	1
5	Motor Power Supply	2.4VDC 3.3A Class 2 Power Supply	1
6	PCB Board	Solderable Electronics Prototyping Board	1
7	Amplifier Board	3.7W Class D Audio Amplifier	1
8	Power Strip	6 Outlet Power Strip with on/off Switch	1

NOTE:
ELECTRONICS
MOUNTING
HARDWARE
NOT SHOWN



UNLESS OTHERWISE SPECIFIED:	NAME	DATE
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN GWT	4/23/14
	CHECKED	
	ENG APPR.	
	MFG APPR.	
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.	
MATERIAL	COMMENTS:	
FINISH		
DO NOT SCALE DRAWING		

NEXT ASSY | USED ON

APPLICATION

TITLE:
CONTROL BOX ASSEMBLY

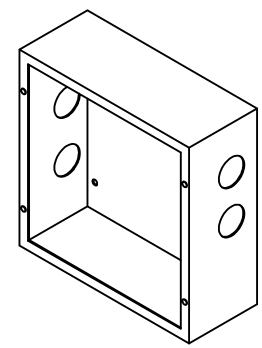
SIZE A | DWG. NO. | REV

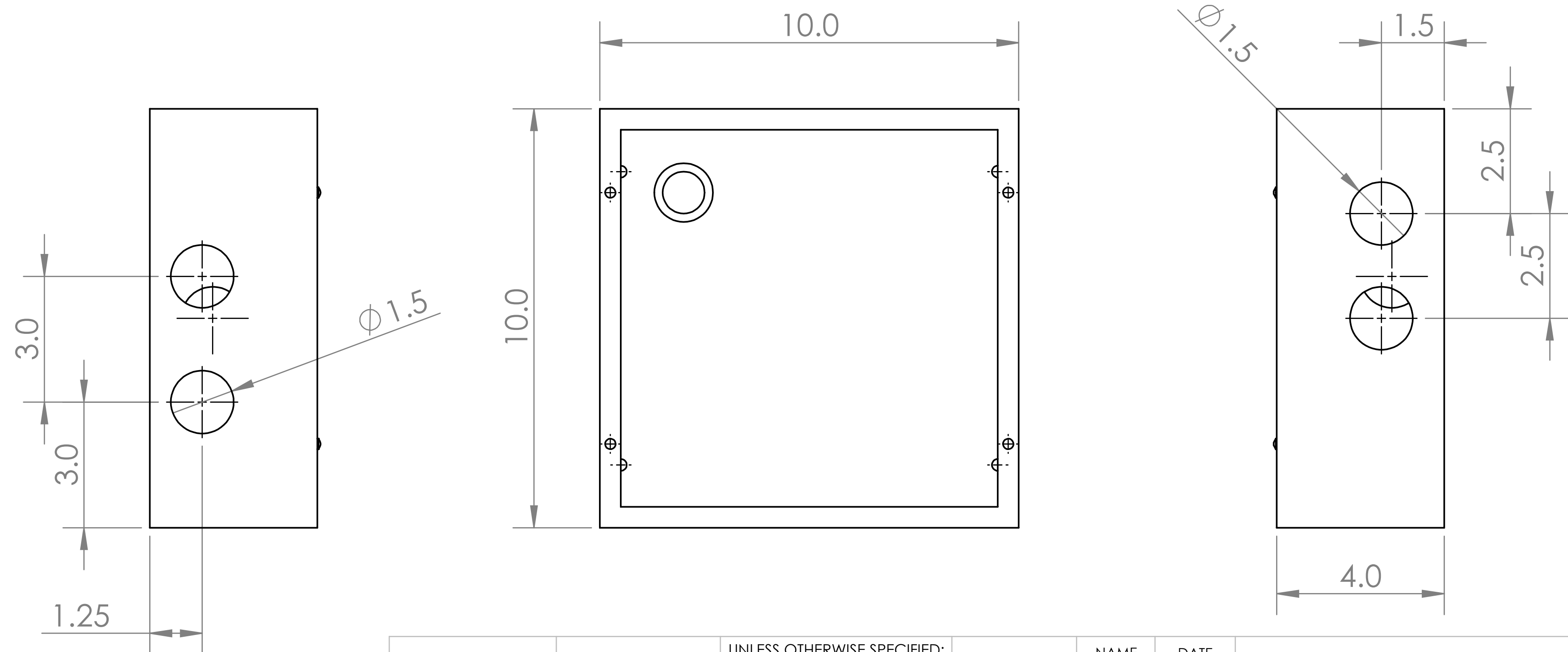
SCALE: 1:4 | WEIGHT: | SHEET 1 OF 1

5 4 3 2 1

NOTE: ADDITIONAL MOUNTING HOLES MAY
BE REQUIRED FOR INTERNAL COMPONENTS

NOTE: ALL WALL THICKNESS IS 16 GA = 0.06"



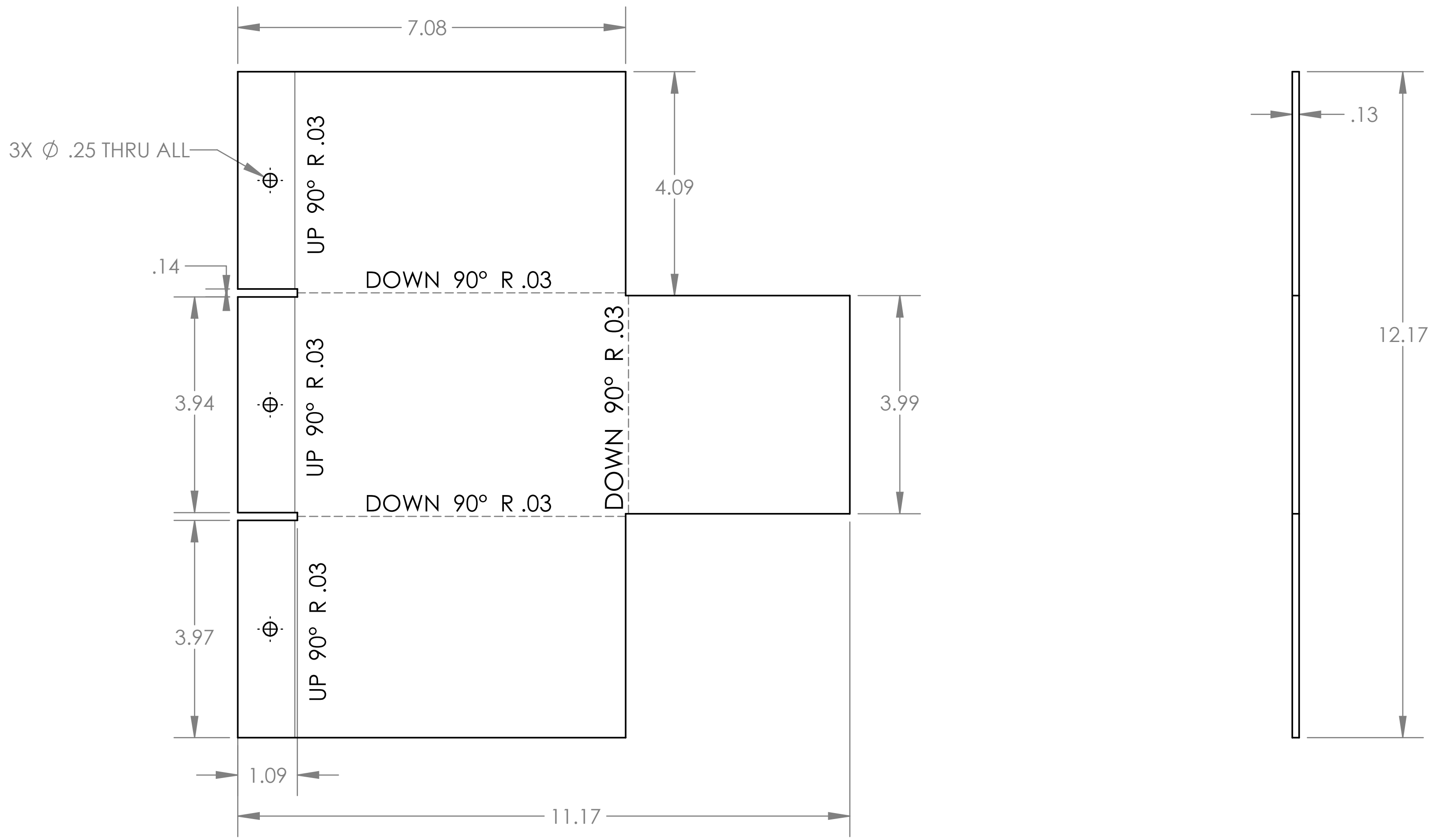


		UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ±.01 THREE PLACE DECIMAL ±.005	DRAWN	GWT	4/23/14	
			CHECKED			TITLE: ELECTRONICS BOX
			ENG APPR.			
			MFG APPR.			
		INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
		MATERIAL STEEL	COMMENTS:			SIZE A DWG. NO. REV
NEXT ASSY	USED ON	FINISH POWDER COATED				
APPLICATION		DO NOT SCALE DRAWING				SCALE: 1:4 WEIGHT: SHEET 1 OF 1

PROPRIETARY AND CONFIDENTIAL

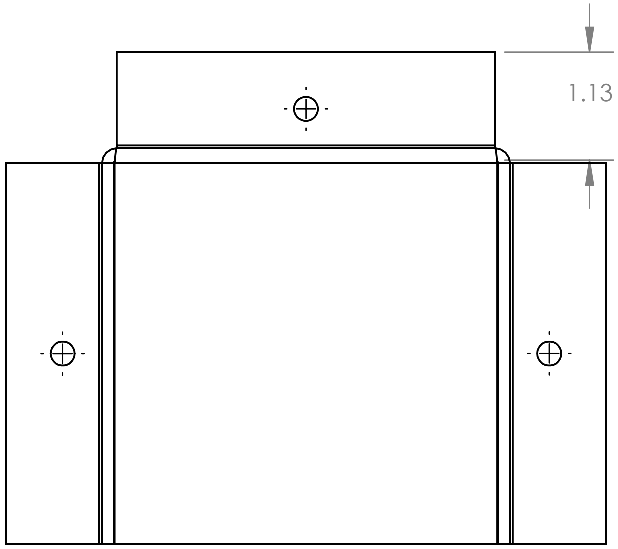
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

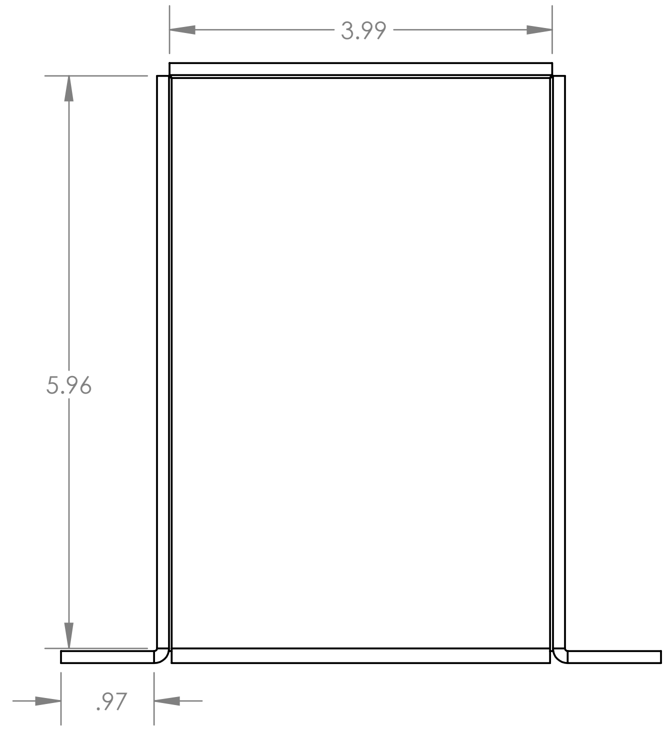
5 4 3 2 1

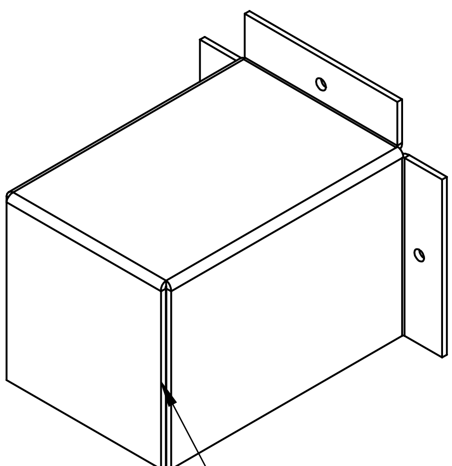


UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN			TITLE:
	CHECKED			MOTOR CASE
	ENG APPR.			
	MFG APPR.			
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
MATERIAL ACRYLIC	COMMENTS:			SIZE A / DWG. NO. / REV
FINISH				
DO NOT SCALE DRAWING				SCALE: 1:3 / WEIGHT: / SHEET 1 OF 2

5 4 3 2 1

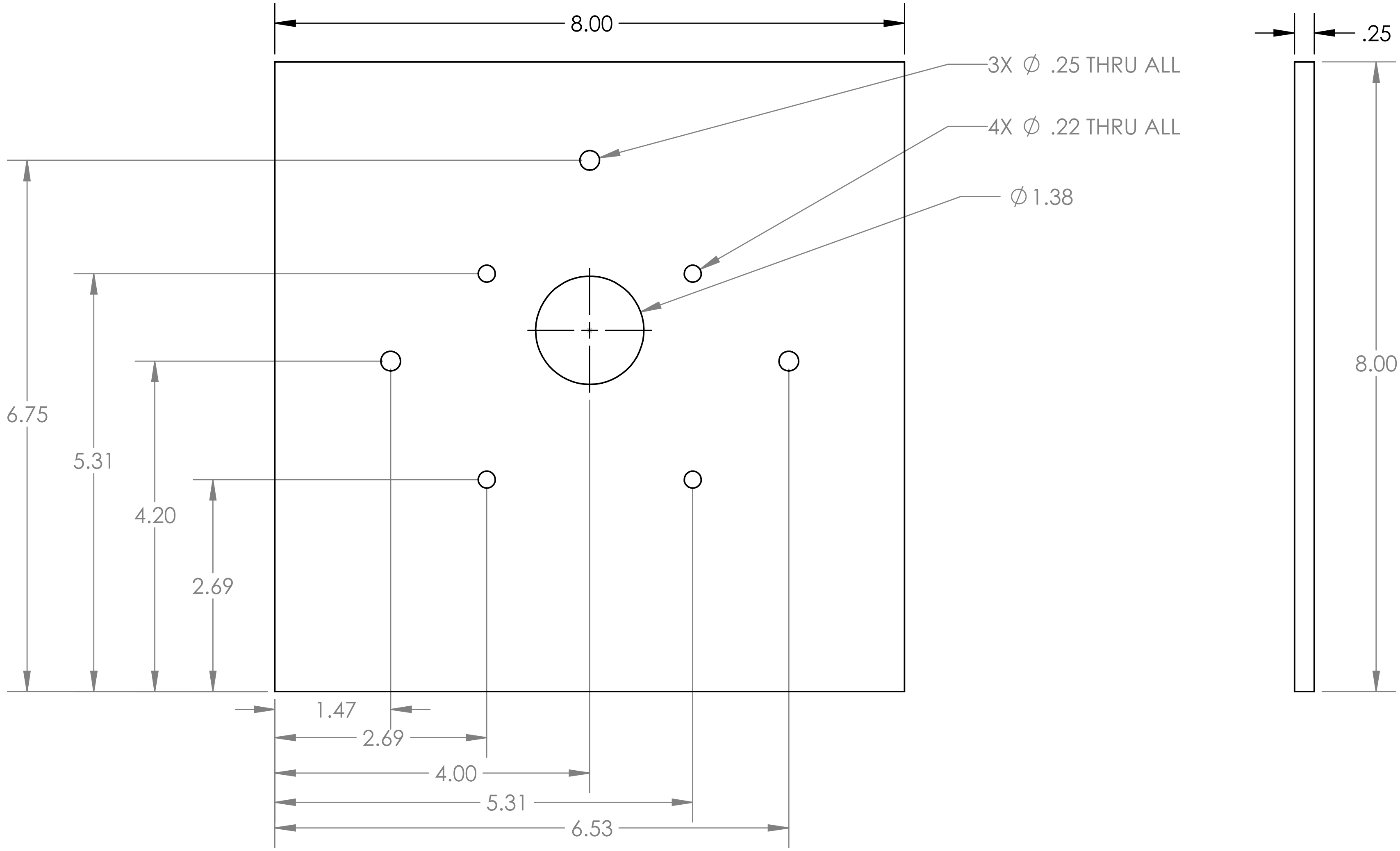






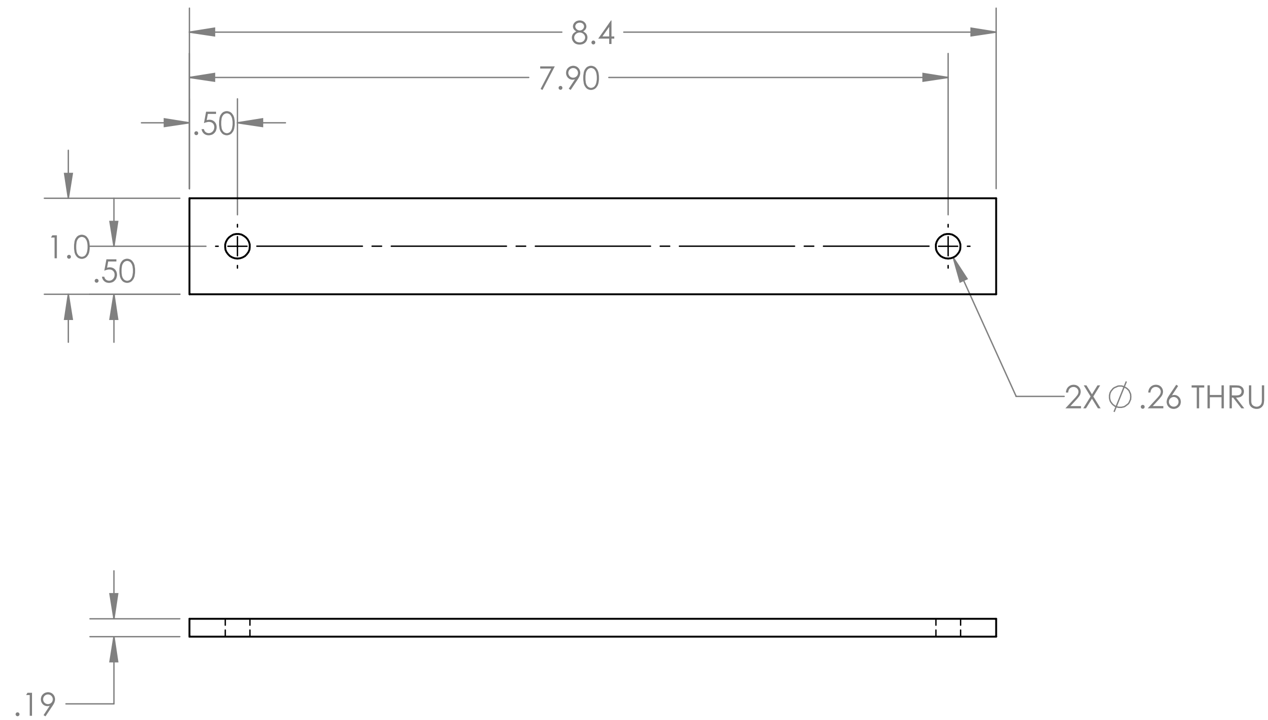
UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	ADAM		TITLE:
	CHECKED			MOTOR CASE
	ENG APPR.			
	MFG APPR.			
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
MATERIAL ACRYLIC	COMMENTS:			SIZE **A** — DWG. NO. — REV
FINISH				
DO NOT SCALE DRAWING				SCALE: 1:2 — WEIGHT: — SHEET 2 OF 2

5 4 3 2 1



UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	Adam		TITLE:
	CHECKED			MOTOR MOUNT PLATE
	ENG APPR.			
	MFG APPR.			
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
MATERIAL ALUMINUM	COMMENTS:			SIZE A	DWG. NO.	REV
FINISH				
DO NOT SCALE DRAWING				SCALE: 1:2	WEIGHT:	SHEET 1 OF 1

5 4 3 2 1



		UNLESS OTHERWISE SPECIFIED:		NAME	DATE
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	GWT	4/26/14
			CHECKED		
			ENG APPR.		
			MFG APPR.		
		INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
		MATERIAL 6061 AL	COMMENTS:		
NEXT ASSY	USED ON	FINISH			
APPLICATION		DO NOT SCALE DRAWING			

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

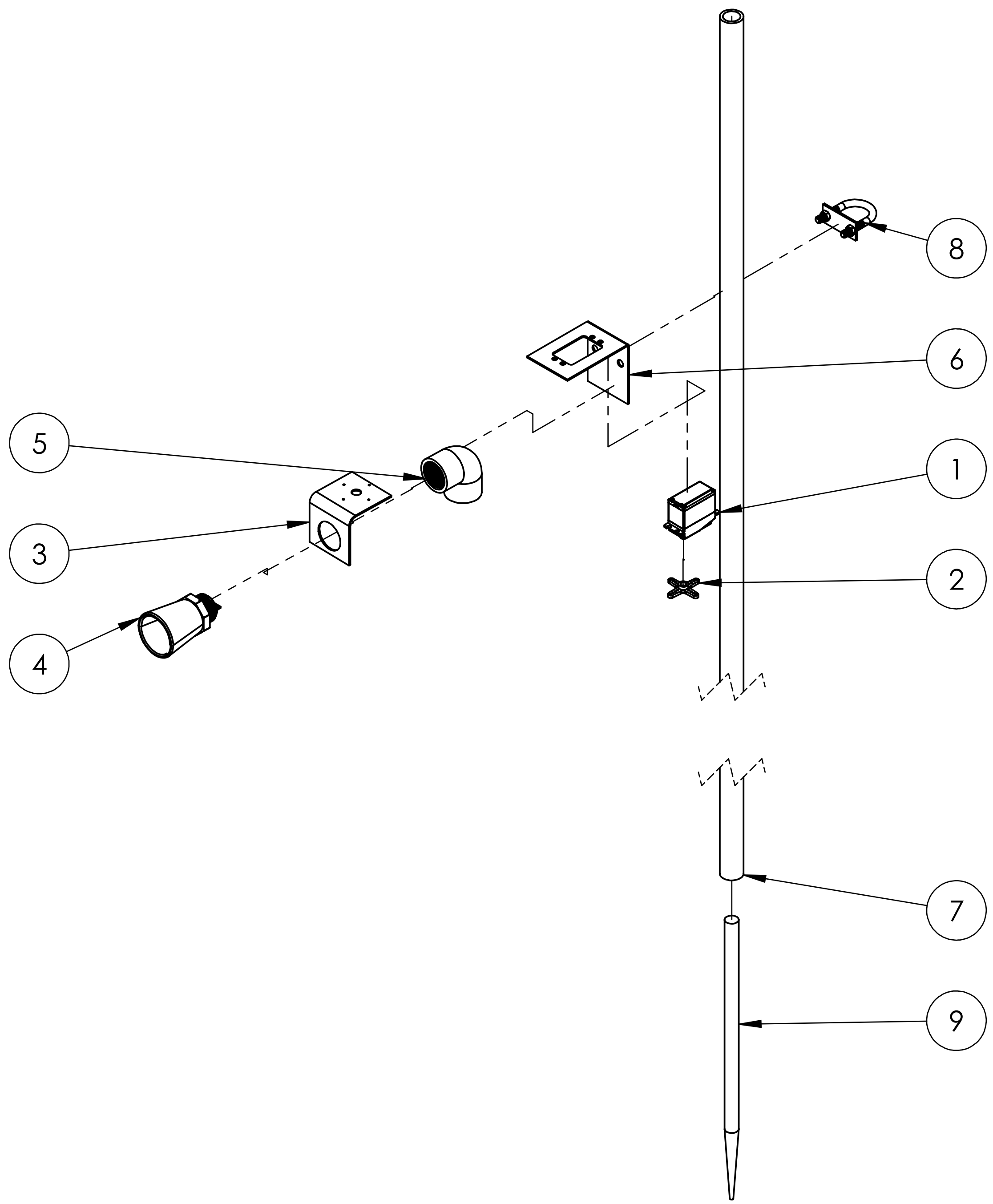
TITLE: **MOTOR TOP PLATE**

SIZE	DWG. NO.	REV
A		

SCALE: 1:2	WEIGHT:	SHEET 1 OF 1

5 4 3 2 1

ITEM NO.	PART NUMBER	Default/ QTY.
1	TowerPro SG-5010 Servo Motor	1
2	TowerPro SG-5010 Servo Motor	1
3	Rangefinder Mount Bracket	1
4	XL-MAX SONAR-WR	1
5	3/4" NPT PVC FITTING	1
6	Servo Mount	1
7	Steel Pipe	1
8	U-BOLT	1
9	Stake	1



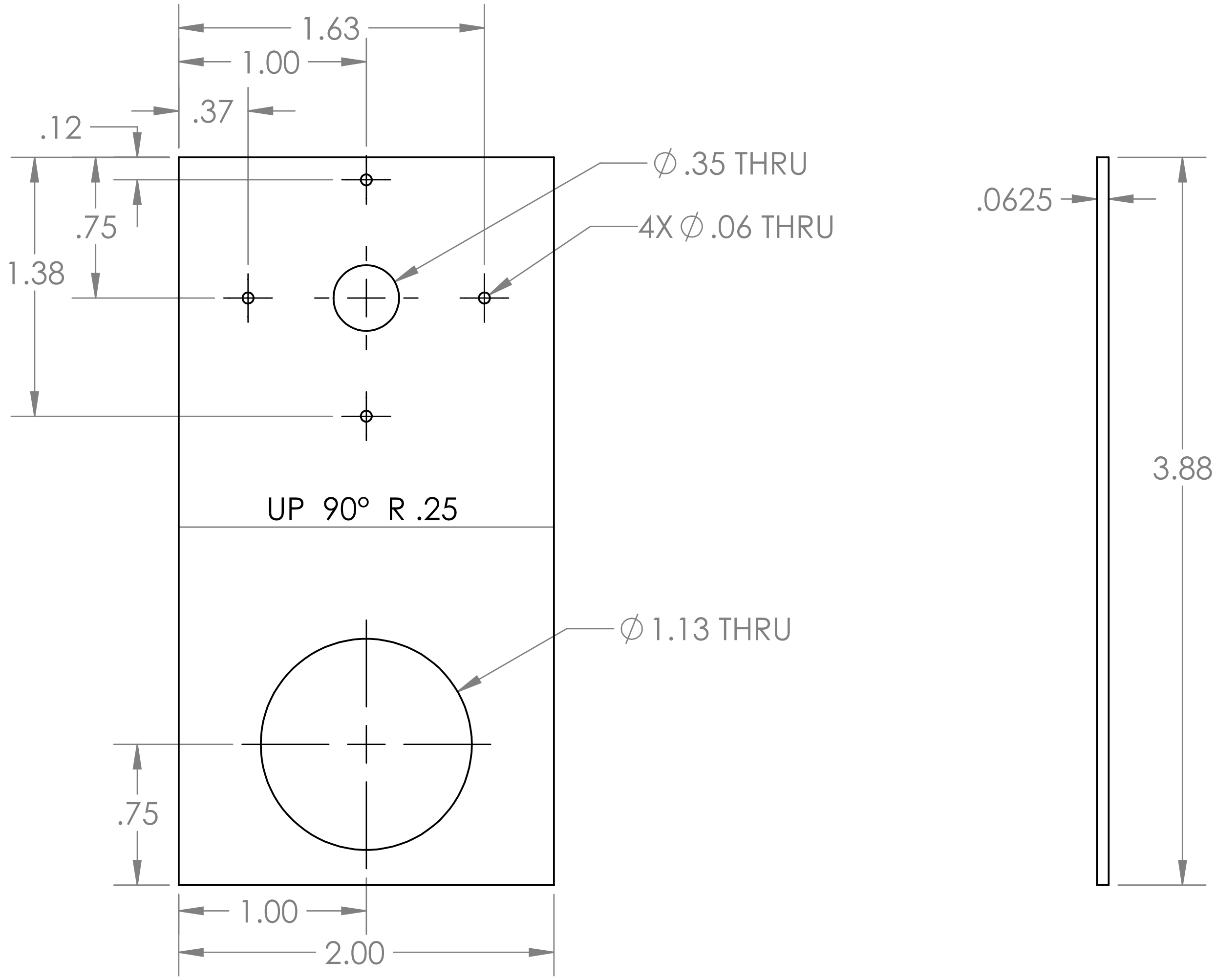
UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	ADAM	
	CHECKED		
	ENG APPR.		
	MFG APPR.		
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
MATERIAL	COMMENTS:		
FINISH			
DO NOT SCALE DRAWING			

TITLE:

Rangefinder Assembly

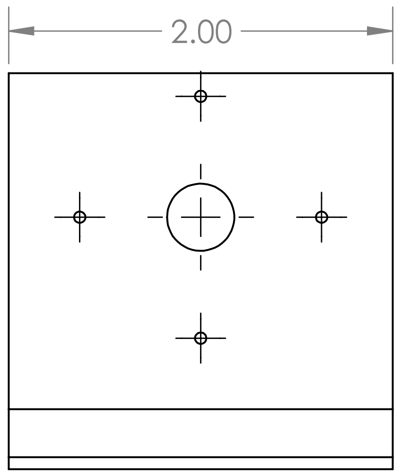
SIZE	DWG. NO.	REV
B		

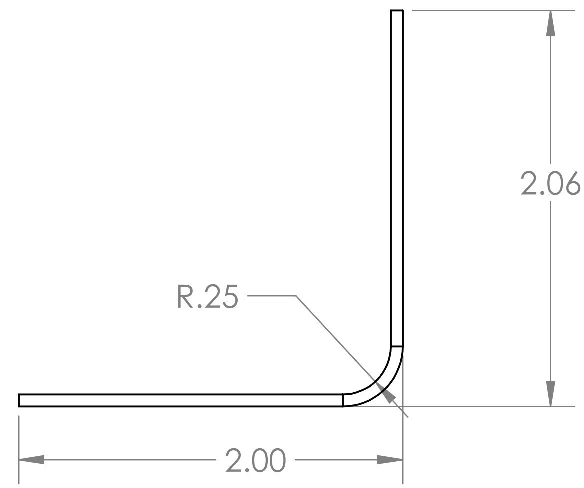
SCALE: 1:6 WEIGHT: SHEET 1 OF 1

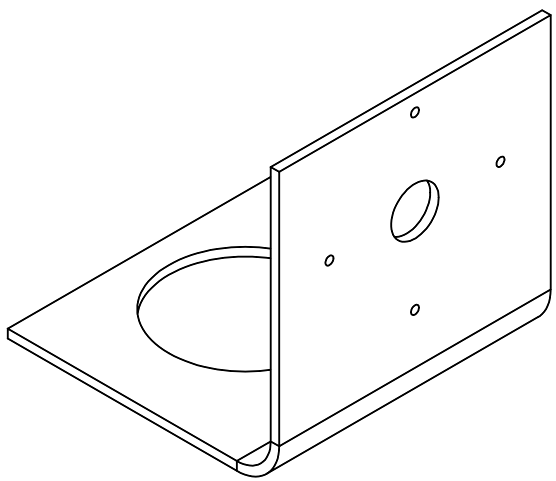


UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	ADAM		TITLE:
	CHECKED			RANGEFINDER MOUNT BRACKET
	ENG APPR.			
	MFG APPR.			
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
MATERIAL ALUMINUM	COMMENTS:			SIZE A — DWG. NO. — REV
FINISH				
DO NOT SCALE DRAWING				SCALE: 1:1 — WEIGHT: — SHEET 1 OF 2

5 4 3 2 1

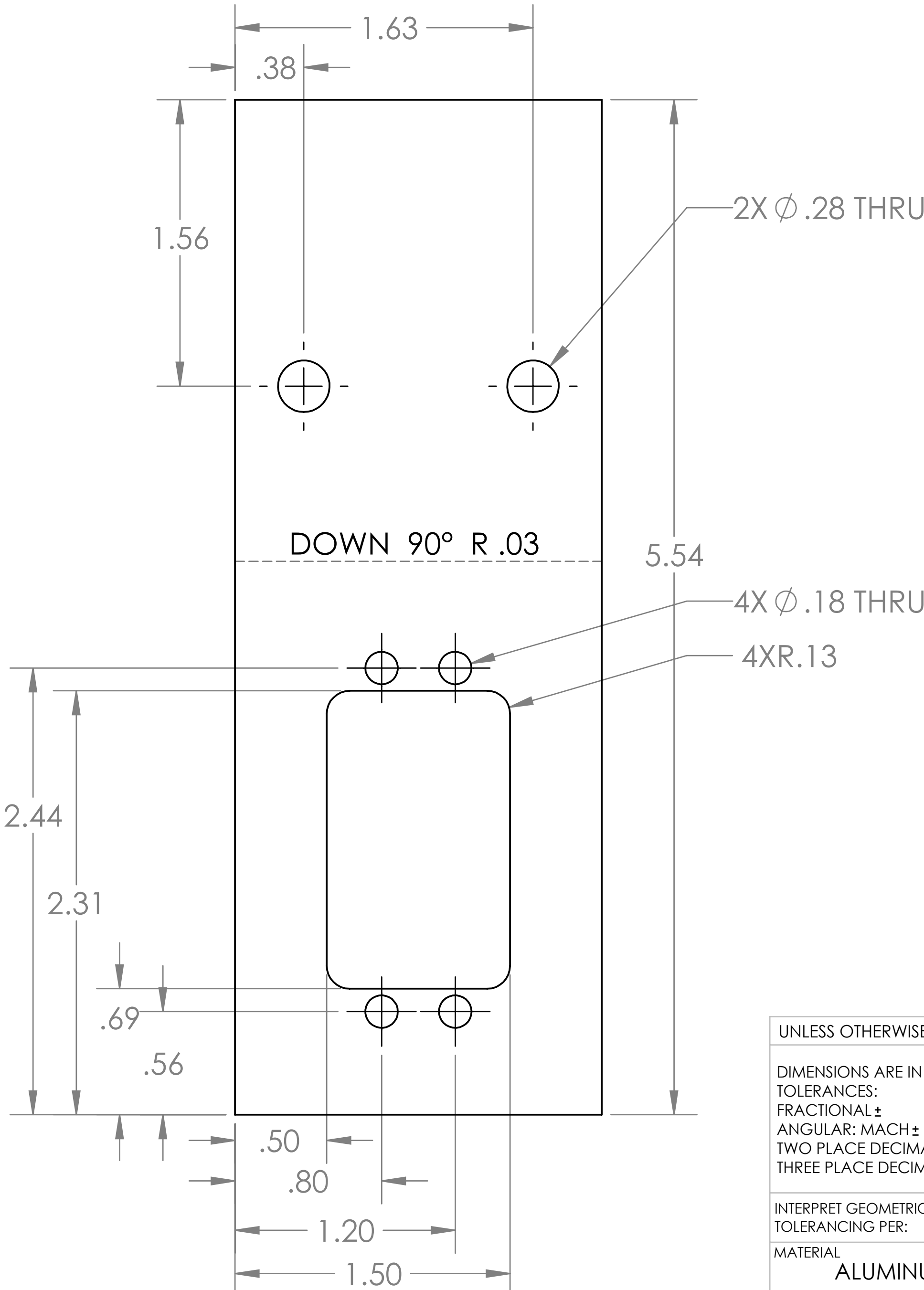


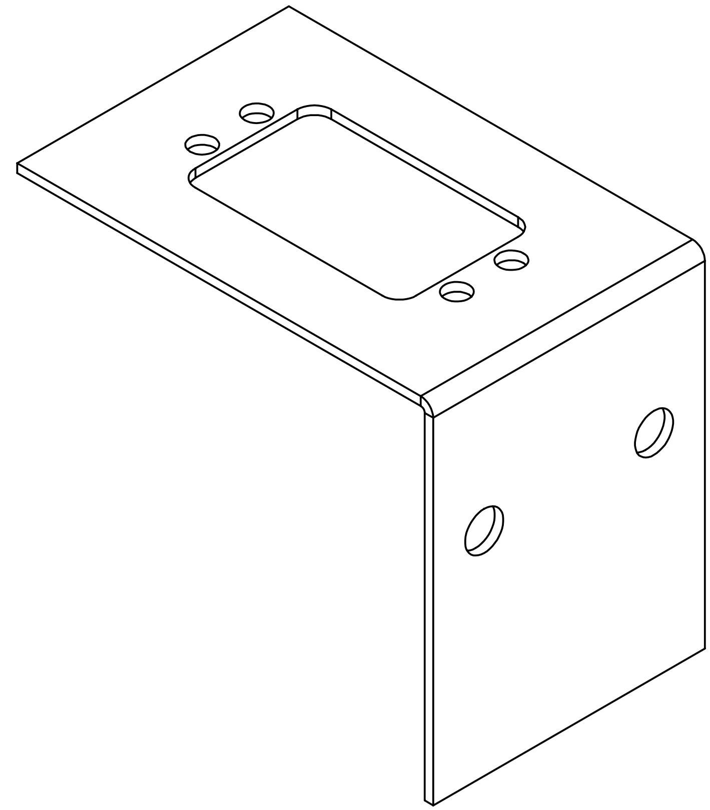




UNLESS OTHERWISE SPECIFIED:		NAME	DATE			
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	ADAM		TITLE:		
	CHECKED			RANGEFINDER MOUNT BRACKET		
	ENG APPR.					
	MFG APPR.					
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.					
MATERIAL ALUMINUM	COMMENTS:			SIZE A	DWG. NO.	REV
FINISH						
DO NOT SCALE DRAWING				SCALE: 1:1	WEIGHT:	SHEET 2 OF 2

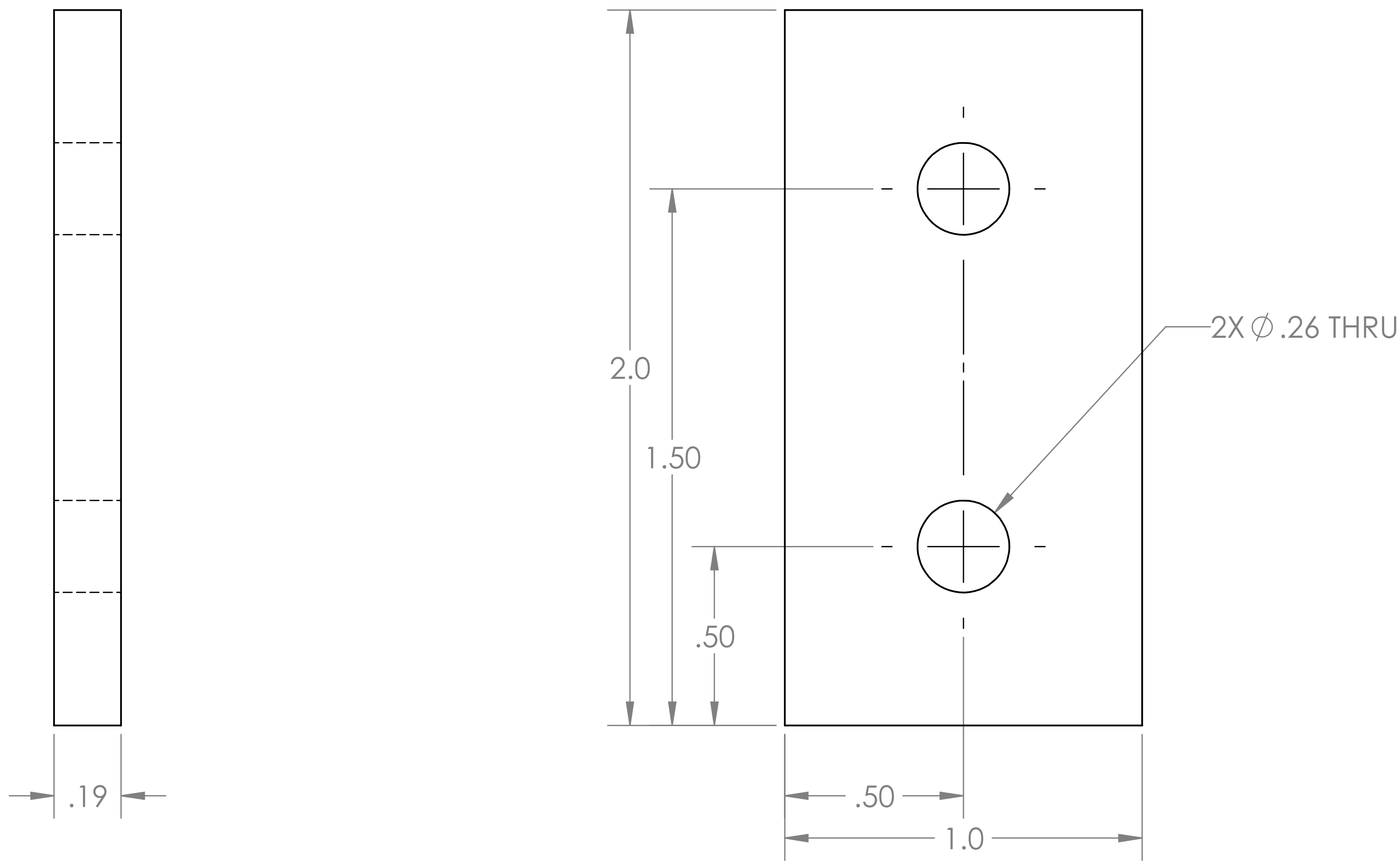
5 4 3 2 1





UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	ADAM	
	CHECKED		
	ENG APPR.		
	MFG APPR.		
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
MATERIAL ALUMINUM	COMMENTS:		
FINISH			
DO NOT SCALE DRAWING			

TITLE: Servo Mount		
SIZE A	DWG. NO.	REV
SCALE: 1:1	WEIGHT:	SHEET 1 OF 1



	NAME	DATE	
DRAWN	GWT	4/26/14	
CHECKED			TITLE:
ENG APPR.			180-BRACKET
MFG APPR.			
Q.A.			
COMMENTS:			

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL
6061 AL

FINISH

DO NOT SCALE DRAWING

NEXT ASSY	USED ON
APPLICATION	

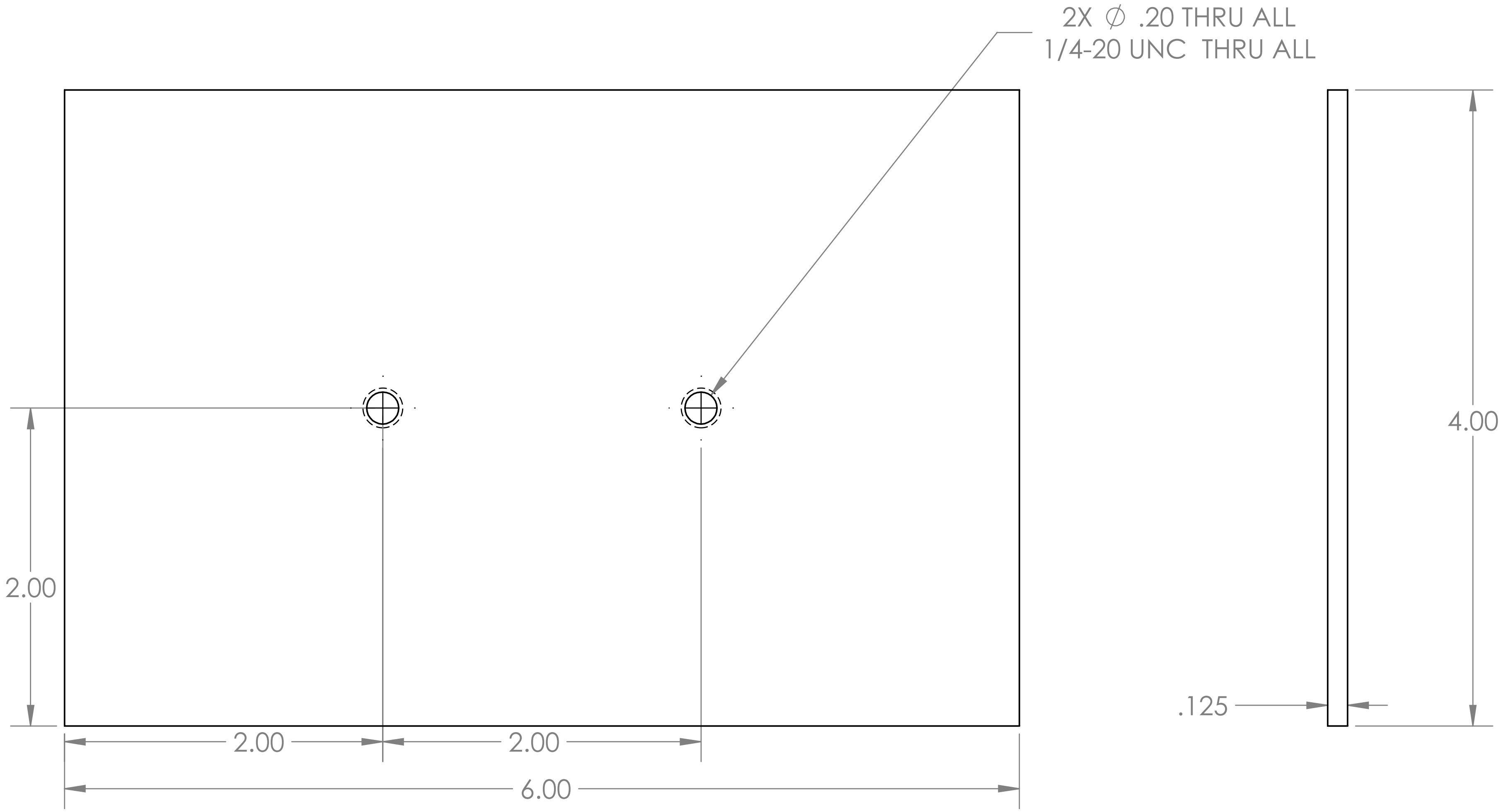
PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

SIZE	DWG. NO.	REV
A		

SCALE: 2:1 WEIGHT: SHEET 1 OF 1

5 4 3 2 1



UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	ADAM	
	CHECKED		
	ENG APPR.		
	MFG APPR.		
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
MATERIAL STEEL	COMMENTS:		
FINISH			
DO NOT SCALE DRAWING			

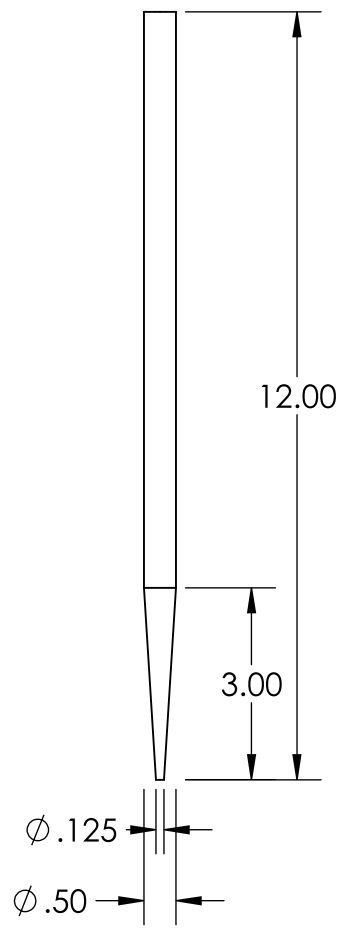
TITLE:

SPEAKER PLATE

SIZE	DWG. NO.	REV
A		

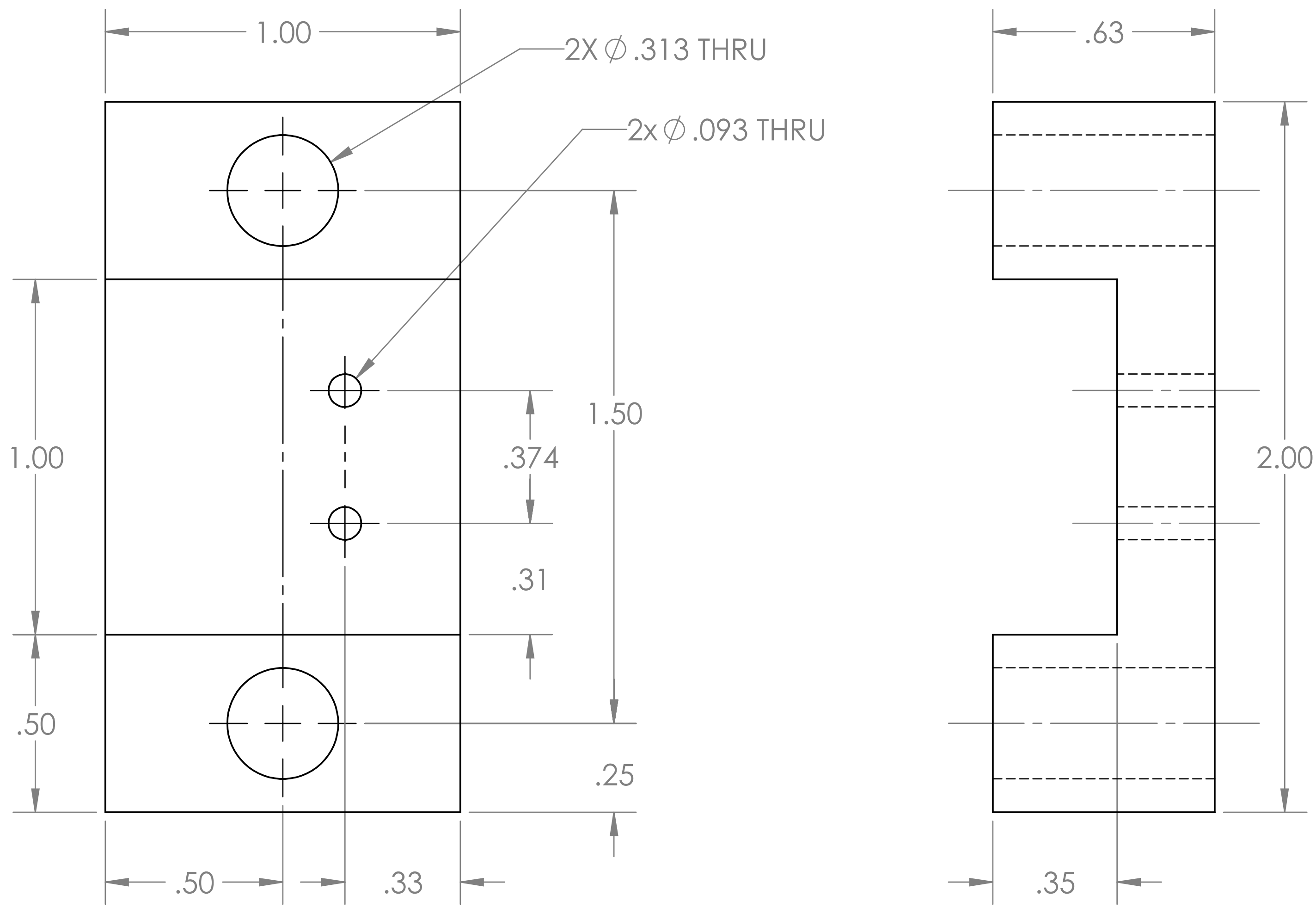
SCALE: 1:1	WEIGHT:	SHEET 1 OF 1

5 4 3 2 1



UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN			TITLE:
	CHECKED			Stake
	ENG APPR.			
	MFG APPR.			
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
MATERIAL STEEL	COMMENTS: STAKE WILL BE WELDED TO 4 FT PIPE			SIZE A DWG. NO. REV
FINISH				
DO NOT SCALE DRAWING				SCALE: 1:3 WEIGHT: SHEET 1 OF 1

5 4 3 2 1



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±.01
THREE PLACE DECIMAL ±.005

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL
POLYETHELYNE

FINISH

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	GWT	4/27/14
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

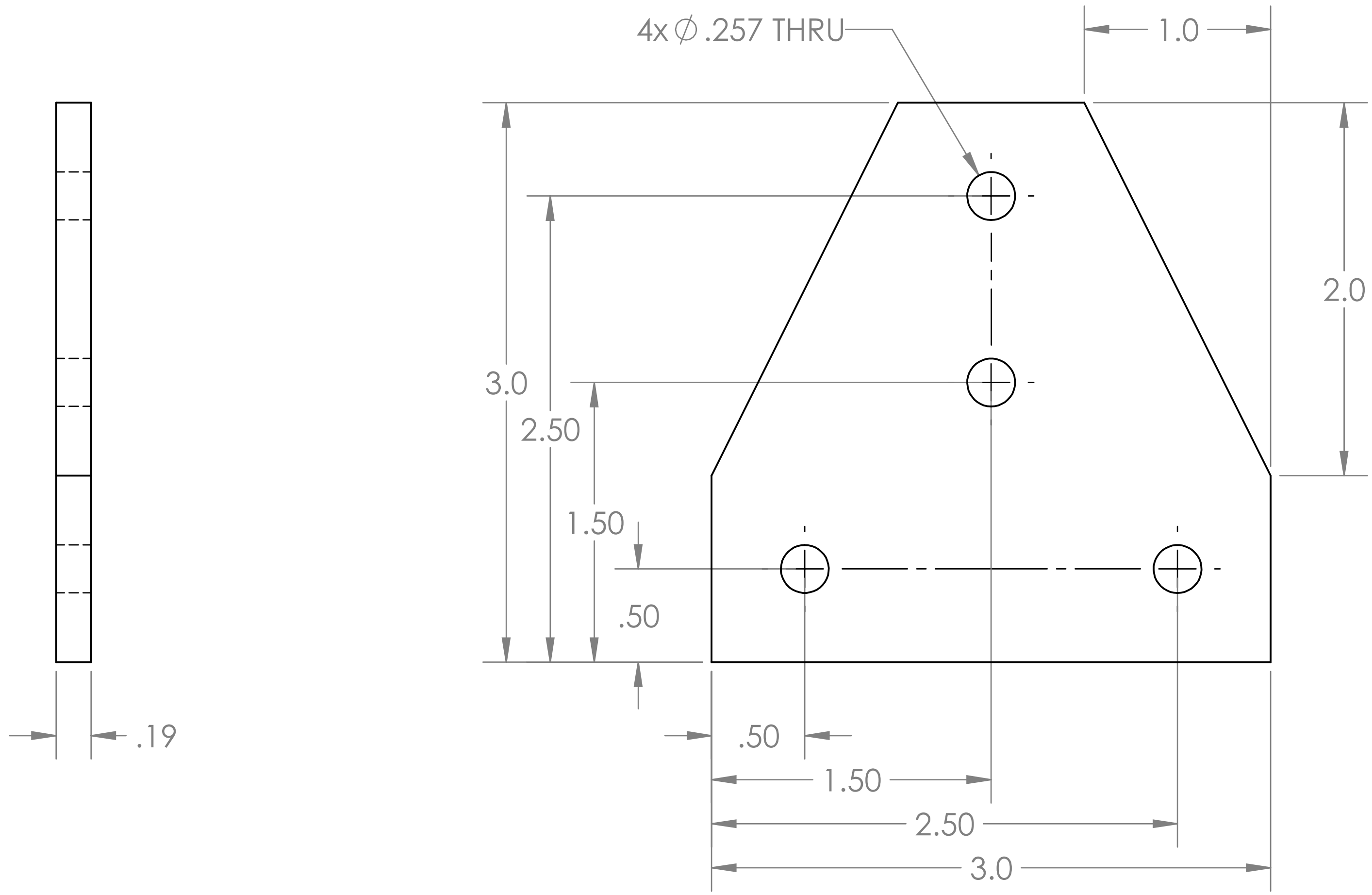
TITLE:

SWITCH MOUNT

SIZE	DWG. NO.	REV
A		

SCALE: 2:1	WEIGHT:	SHEET 1 OF 1

5 4 3 2 1



		UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	GWT	4/26/14	
			CHECKED			TITLE:
			ENG APPR.			T-BRACKET
			MFG APPR.			
		INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
		MATERIAL 6061 AL	COMMENTS:			SIZE A DWG. NO. REV
NEXT ASSY	USED ON	FINISH				
APPLICATION		DO NOT SCALE DRAWING				SCALE: 1:1 WEIGHT: SHEET 1 OF 1

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

5 4 3 2 1

Calculations

1. Motor Torque/Speed
2. Wind Tipping
3. Driving Shaft Bending

Motor Torque/Speed

Requirements:

- Car travels 30 feet in 3.5 seconds from rest.
- Car accelerates to full speed in 1.5 seconds from rest.
- Modify required torque results with a correction factor of 1.25 to account for losses and assumptions.

Model Car as mass on frictionless surface:

X direction:

$$t_{TS} = 1.5s, \qquad t_F = 3s, \quad x_F = 30ft$$

$$x = x_0 + v_0 t + \frac{1}{2}at^2 \rightarrow \; v_0 = 0, x_0 = 0 \; \rightarrow x = \frac{1}{2}at^2$$

$$x_{TS} = \frac{1}{2}at_{TS}{}^2$$

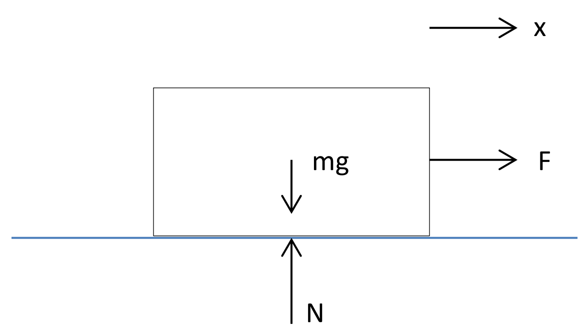


Figure A1: FBD of Car on Ideal Surface

At top speed:

$$x = vt, \qquad v = const, \quad a = 0$$

$$v_{TS} = at_{TS}$$

Total distance travelled, x_F:

$$x_F = x_{TS} + v_{TS}(t_F - t_{TS}) = \frac{1}{2}at_{TS}{}^2 + at_{TS}(t_F - t_{TS})$$

$$a = \frac{x_F}{[\frac{1}{2}(t_{TS}^2 + t_{TS}(t_F - t_{TS})]} = \frac{30ft}{\left[\frac{1}{2}(1.5s^2) + (1.5s)(3.5s)\right]} = 7.27\frac{ft}{s^2}$$

$$v_{TS} = \left(7.27\frac{ft}{s^2}\right)(1.5s) = 10.91\frac{ft}{s}$$

Sum of Forces in X-direction:

Car Weight (SolidWorks): Weight=18.21lbf

$$m = \frac{18.21lbf}{32.174\ \frac{ft}{s^2}} = .566lbm$$

Find required force to push block:

$$F = ma = .566lbm\left(7.27\frac{ft}{s^2}\right) = 4.12lbf$$

Car Wheel: See Figure A2 for Free-Body Diagram

Wheel Diameter=5in.

Max Rotational Speed:

$$v = r\omega \rightarrow \omega = \frac{v}{r} \rightarrow w_{max} = \frac{10.91ft/s}{\frac{5in}{2}\frac{1ft}{12in}} = 52.32rad/s$$

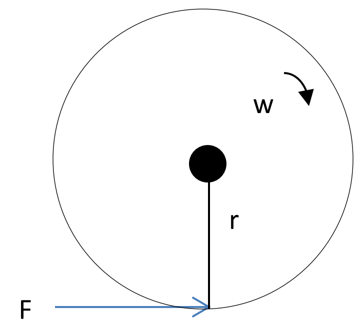


Figure A2: FBD of Car Wheel

Convert to RPM to obtain required motor speed:

$$52.32\frac{rad}{s}\frac{60s}{1min}\frac{1rev}{2\pi rad} = \boldsymbol{500rpm}$$

Find required torque:

$$\tau = rF = \frac{5in}{2}\frac{1ft}{12in}4.12lbf = .86ftlbf$$

Convert to oz*in:

$$.86ftlbf\frac{192ozin}{1ftlbf} = 164.8ozin$$

Apply correction of 1.25 for final torque:

$$164.8ozin * 1.25 = \boldsymbol{206ozin}$$

Conclusion:

Requirements Obtained for the motor selection process are shown below:

Loaded Motor Speed: **500 RPM**

Loaded Motor Torque: **206ozin**

The parameters used to determine the calculated values are not requirements but are a set of estimated values that would contribute towards making the actual approach system an effective deterrent. Therefore, it is not mandatory for the selected motors to have these calculated values.

Wind Tipping:

The goal of this calculation is to find a critical wind speed at which the car will tip. The most likely way for the car to tip is laterally (around the axis of the guidance rope. The rope will prevent the car from being able to tip forward and backward, so these scenarios will not be considered. A wind pressure will be placed at the top of the mounting pole creating a moment at the center of mass. The calculation will be done at the moment of tipping, when the reaction forces must be on one side of the car only and equal the weight of the car. The weight and center of mass location were calculated using SolidWorks. Additionally, the area that the wind pressure acts on will be considered as a rigid 0.5ft. by 1 ft. surface perpendicular to the flow. This assumption will be made to represent the side profile of any mounted object. The car must be able to withstand at least a 35 mph gust in order to be considered stable. See figure B1 below for a simplified free-body diagram.

$$F_W = P_{stagnation} A$$

$$P_{stagnation} = \frac{1}{2}\rho v^2 + \cancel{P_{static}}$$

$$A = (0.5ft)(1ft) = 0.5ft^2$$

$$R_a = 18lbf$$

$$\rho_{air@70°F} = .002329\frac{slugs}{ft^3}$$

$$\Sigma M_O = R_a(7.5in) - F_W(48in) = 0$$

$$F_W = \frac{18lbf(7.5in)}{48in} = 2.8125lbf = \frac{1}{2}\rho v^2 A$$

$$v = \sqrt{\frac{2(2.8125lbf)}{\left(002329\frac{slugs}{ft^3}\right)(0.5ft^2)}} = 69.5\frac{ft}{s}$$

$$= \boldsymbol{50mph}$$

Conclusion:

Since the car is able to withstand a lateral side wind of up to 50 mph, the lateral stability of the car is exceptional. The car as a whole will be considered stable because the car is only able to rotate laterally about the axis of the rope.

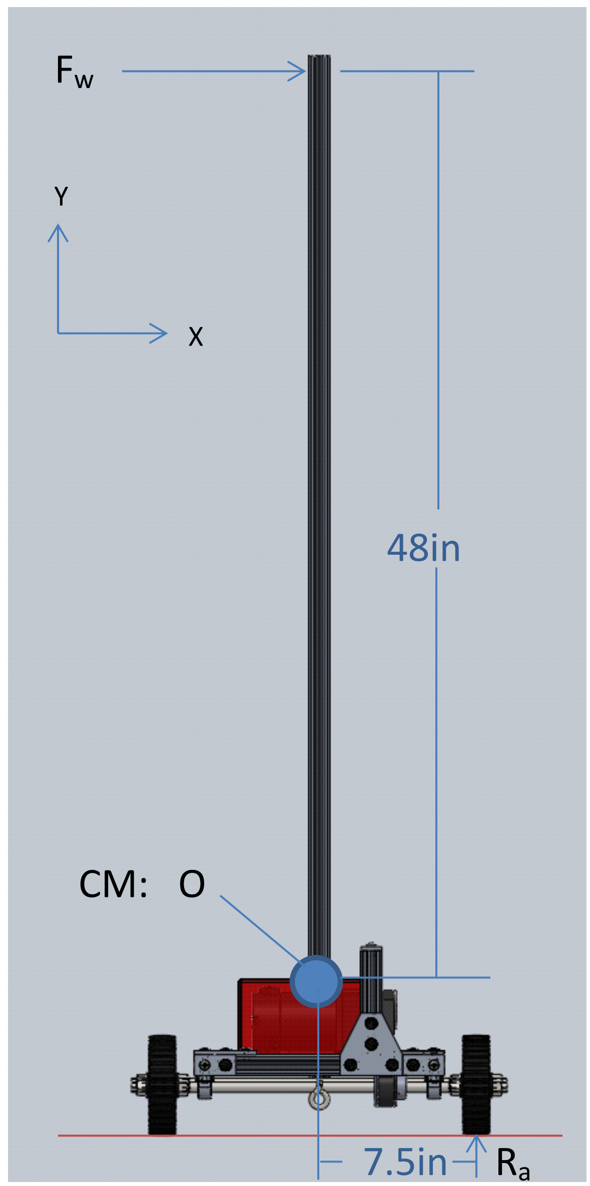


Figure B1: Simplified Car FBD for Lateral Wind

Shaft Bending

The front drive shaft is under the most complex loads of the entire actual approach system. This calculation aims to prove that the maximum deflection of the shaft will be less than .015 inches, so that tension is kept on the timing pulley and so that the wheels remain aligned. Figures C1 and C2 below show free-body diagrams of the front drive shaft with and without components assembled respectively. Dimensions have been excluded from these drawings for simplicity but will be listed below. Furthermore, axial forces on the X-direction will be excluded from this calculation as bending will mostly occur in the X-Z and Y-X planes. The shaft flat located under the timing pulley will be neglected. A factor of safety of 5 will be used to account for assumptions.

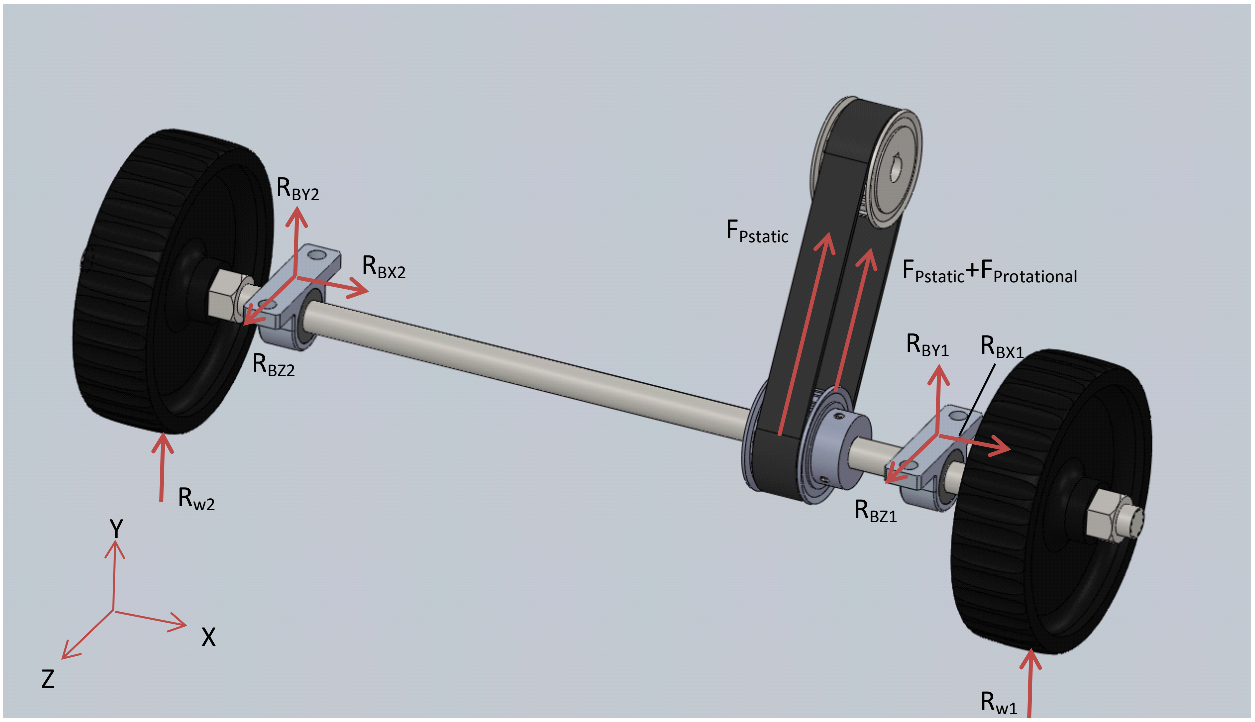


Figure C1: FBD of Front Shaft with Components Assembled

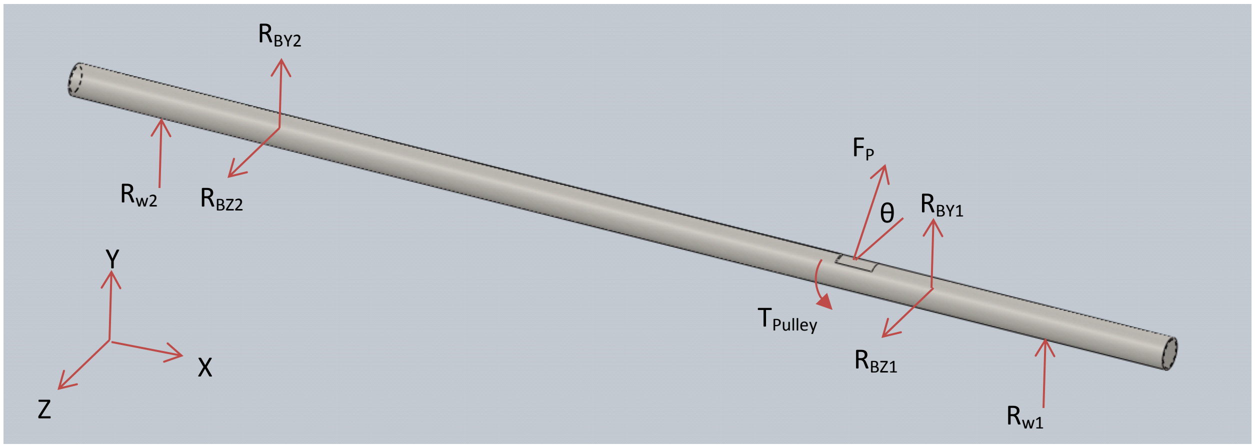


Figure C2: Simplified FBD of Front Shaft. Note: X dimensions removed.

Assumptions:

Dimensions (from the left end of the shaft) to associated force point along X-axis:

$Wheel\ 1\ (R_{W1})$: $16.5in$

$Bearing\ 1\ (R_{B1})$: $14.5in$

$Pulley\ \ (F_P)$: $12.25in$

$Bearing\ 2\ (R_{B2})$: $3.5in$

$Wheel\ 2\ (R_{W2})$: $1.5in$

Forces (calculated from SolidWorks model):

$R_{W2} \cong R_{W1} = 6.7lbf$

$F_P = 5lbf\ @\ \theta = 58°$

Calculations of Reaction Forces:

$\Sigma F_y = R_{BY1} + R_{BY2} + F_p sin\theta + R_{W2} + R_{W1} = 0$

$\Sigma M_{left\ end} = R_{BY1}(14.5in) + R_{BY2}(3.5in) + F_p sin\theta(12.25in) + R_{W2}(1.5in) + R_{W1}(16.5in) = 0$

ΣF_y: $\quad R_{BY1} = -R_{BY2} - F_p sin\theta - R_{W2} - R_{W1}$

$$\Sigma M_{left\ end}\!: (-R_{BY2} - F_p sin\theta - R_{W2} - R_{W1})(14.5in) + R_{BY2}(3.5in) + F_p sin\theta(12.25in) + R_{W2}(1.5in) + R_{W1}(16.5in) = 0$$

$\Sigma M_{left\ end}$:

$(-11in)R_{BY2} = F_p sin\theta(2.25in) + R_{W2}(13.5in) - R_{W1}(2in)$

$$R_{BY2} = \frac{F_p sin\theta(2.25in) + R_{W2}(13.5in) - R_{W1}(2in)}{-11in} = \frac{(5lbf)(sin58°)(2.25in) + (6.7lbf)(13.5in) - (6.7lbf)(2in)}{-11in} = \mathbf{-7.87lbf}$$

$$\Sigma F_y\!: \quad R_{BY1} = -R_{BY2} - F_p sin\theta - R_{W2} - R_{W1} = -(-7.87lbf) - (5lbf)(sin58°) - (6.7lbf) - (6.7lbf) = \mathbf{-9.77lbf}$$

$$\Sigma F_z = R_{BZ1} + R_{BZ2} - F_p cos\theta = 0$$

$$\Sigma M_{left\ end} = R_{BZ1}(14.5in) + R_{BZ2}(3.5in) - F_p cos\theta(12.25in) = 0$$

$$\Sigma F_z: \quad R_{BZ1} = -R_{BZ2} + F_p cos\theta = 0$$

$$\Sigma M_{left\ end}: \quad (-R_{BZ2} + F_p cos\theta)(14.5in) + R_{BZ2}(3.5in) - F_p cos\theta(12.25in) = 0$$

$$\Sigma M_{left\ end}: \quad -R_{BZ2}(14.5in) + F_p cos\theta(14.5in) + R_{BZ2}(3.5in) - F_p cos\theta(12.25in) = 0$$

$$R_{BZ2} = \frac{F_p cos\theta(2.25in)}{11in} = \frac{5lbf cos58°(2.25in)}{11in} = \mathbf{.54lbf}$$

$$\Sigma F_z: \quad R_{BZ1} = -R_{BZ2} + F_p cos\theta = -(.54lbf) + 5lbf(cos58°) = \mathbf{2.1lbf}$$

Summary of Reaction Force Calculations:

$R_{BY2} = -7.87lbf$

$R_{BY1} = -9.77lbf$

$R_{BZ2} = .54lbf$

$R_{BZ1} = 2.1lbf$

Bending:

Bending will be calculated at the point on the front shaft that has the largest bending moment determined by shear and moment diagrams in the X-Y and X-Z planes.

Shear Diagrams:

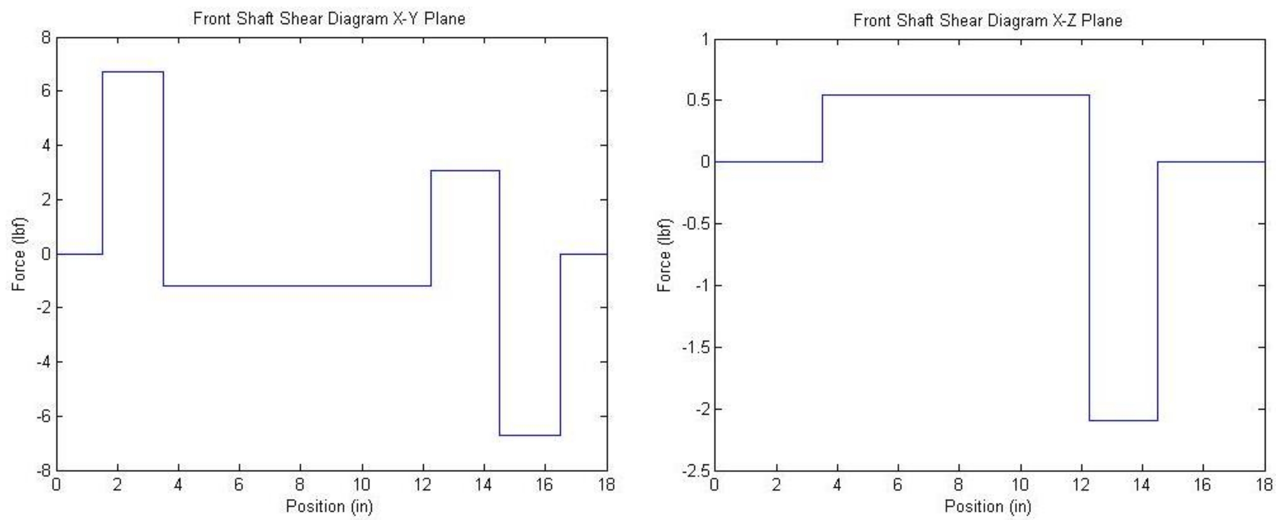


Figure C3: Front Shaft Shear Diagrams for X-Y and X-Z Planes

Moment Diagrams:

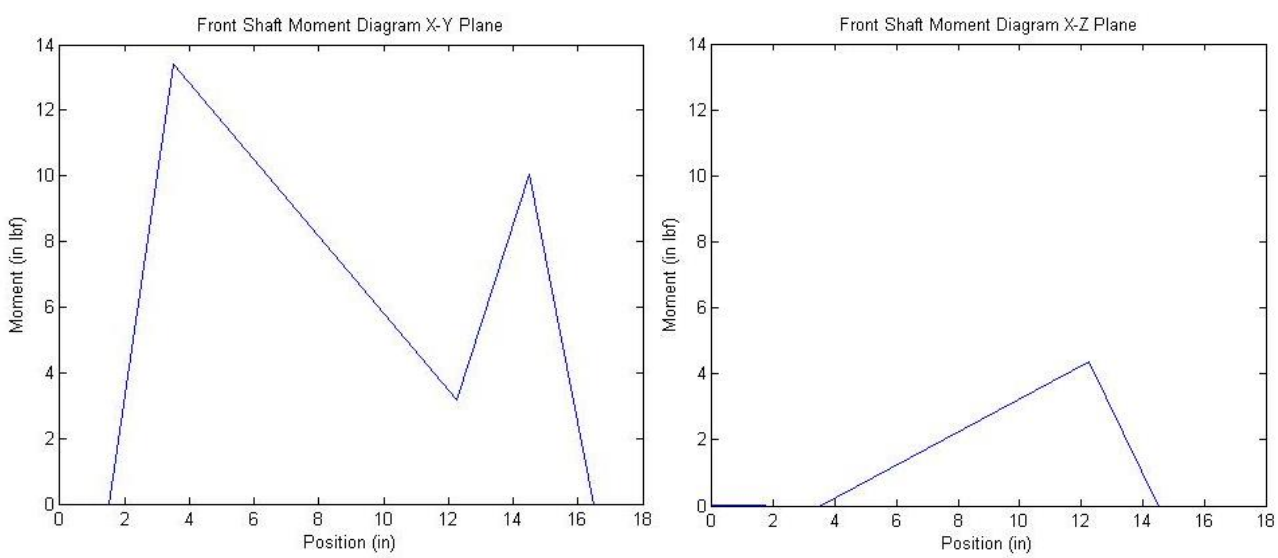


Figure C4: Front Shaft Moment Diagrams for X-Y and X-Z Planes

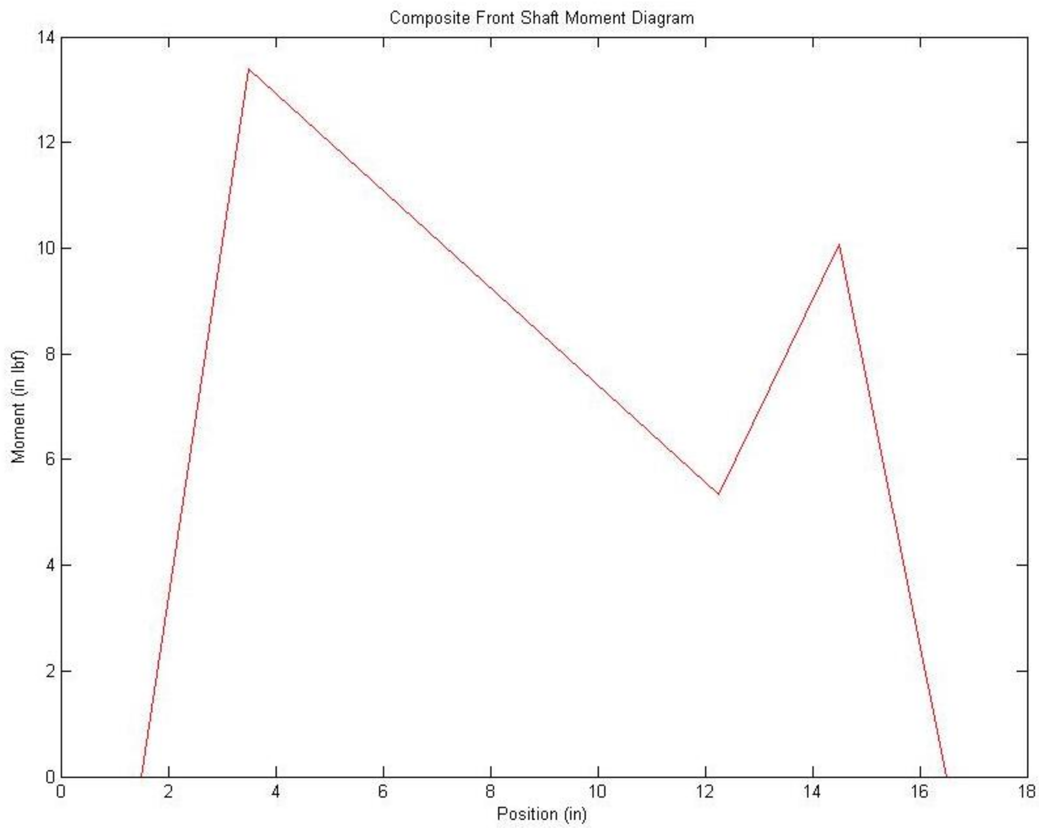


Figure C5: Front Shaft Composite Moment Diagram

As seen in the shear diagrams generated from Matlab (figure C3), the shear in the front shaft is most near the wheels and bearings. Furthermore, figure C4 shows a similar pattern in the moments along the front shaft. As seen in both figure C3 and C4, the X-Y plane forces dominate the X-Z plane forces and the composite moment diagram (figure C5) reflects this as well. The composite moment diagram will be used to calculate deflection as shown below.

Deflection:

Known:

$$D_{shaft} = .5in$$

$$E_{steel} = 29 \times 10^6 \frac{lbf}{in^2}$$

$$I_{shaft} = \frac{\pi}{4} r^4 = \frac{\pi}{4} \left(\frac{.5in}{2}\right)^4 = .00307in^4$$

Calculations:

$$\frac{d^2y}{dx^2} = \frac{M_{(x)}}{E_{steel}I_{shaft}}$$

$$\frac{dy}{dx} = \frac{M_{(x)}x}{E_{steel}I_{shaft}} + c_1$$

$$y = \frac{M_{(x)}x^2}{2E_{steel}I_{shaft}} + c_1x + c_2$$

Initial Conditions: Deflection is zero at bearing mounts (y=0 @ x=3.5in, 14.5in).

Calculate integration constants:

$$y = \frac{M_{(3.5in)}(3.5in)^2}{2E_{steel}I_{shaft}} + c_1(3.5in) + c_2 = 0 \;\; \rightarrow \;\; c_2 = -c_1(3.5in) - \frac{M_{(3.5in)}(3.5in)^2}{2E_{steel}I_{shaft}}$$

$$y = \frac{M_{(14.5in)}(14.5in)^2}{2E_{steel}I_{shaft}} + c_1(14.5in) + -c_1(3.5in) - \frac{M_{(3.5in)}(3.5in)^2}{2E_{steel}I_{shaft}} = 0$$

$$y = \frac{(-3.33inlbf)198in^2}{2E_{steel}I_{shaft}} + c_1(11in) = 0 \rightarrow c_1 = \frac{(-3.33inlbf)198in^2}{(11in)2\left(229 \times 10^6 \frac{lbf}{in^2}\right).00307in^4}$$

$$= -.\mathbf{00004263}$$

$$c_2 = -c_1(3.5in) - \frac{M_{(3.5in)}(3.5in)^2}{2E_{steel}I_{shaft}} = .00004263(3.5in) - \frac{(13.4\ inlbf)(3.5in)^2}{2\left(229 \times 10^6 \frac{lbf}{in^2}\right).00307in^4}$$

$$=.\mathbf{00003246}\boldsymbol{in}$$

Therefore, the equation of the front shaft deflection in terms of position from the left side of the shaft (in inches) is:

$$y = \frac{M_{(x)}x^2}{2E_{steel}I_{shaft}} - .00004263x + .00003246in$$

Matlab will be used to plot the deflection using this governing equation along the shaft:

Figure C6 shows the total front shaft displacement

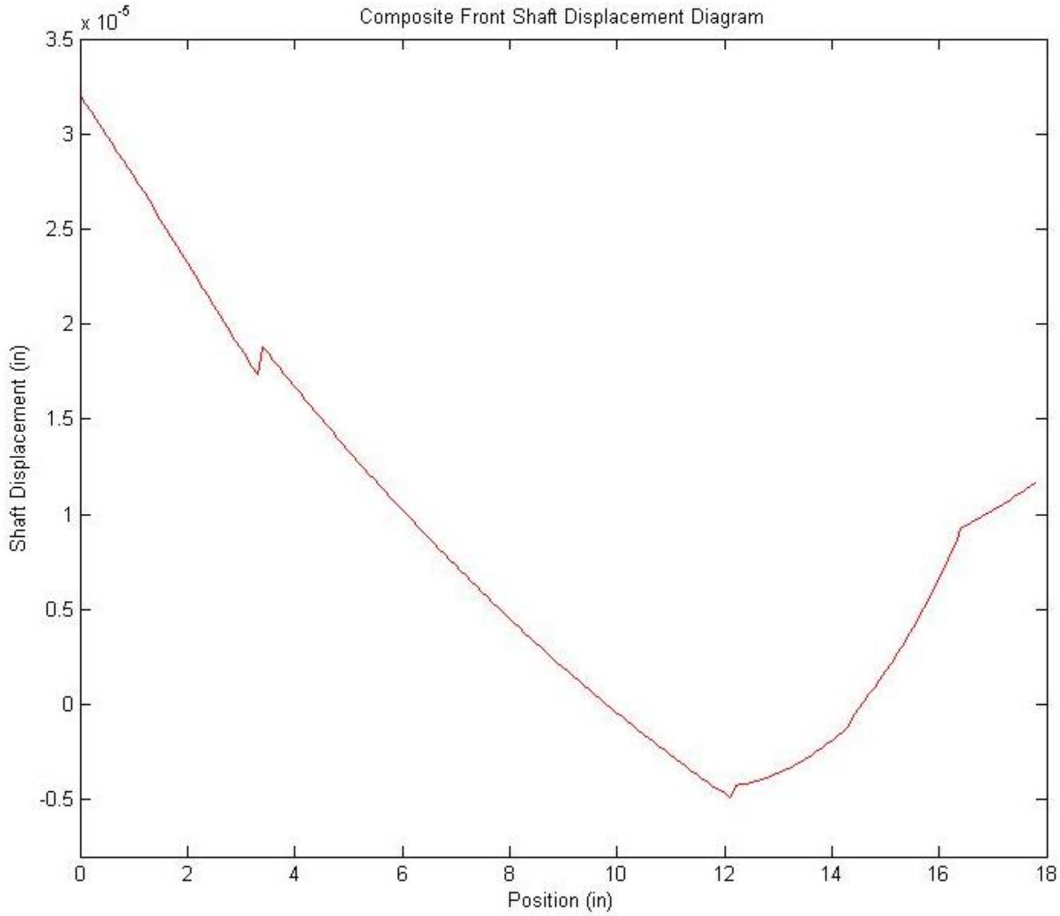


Figure C6: Total Displacement of Front Shaft

Conclusion:

As seen in figure C6, the largest front shaft displacement is roughly $3.25 \times 10^{-5} in$ located at the far left end of the shaft just past the left wheel. After including a Factor of Safety of 5 as specified in the calculation statement, the displacement is equal to:

$$(3.25 \times 10^{-5} in)5 = \mathbf{.0001625} \boldsymbol{in}$$

This displacement is well within the limit of .015in deflection. Therefore, the front shaft is able to withstand all loading while deflecting less than .015in. See below for the Matlab code used to generate displacement diagram.

The following Matlab code was used to generate the total shaft displacement:

```
% Calculates Displacement of Front Drive shaft
% Governing Equation:
% y=(M(x)(x^2))/(2EI)-.00004263x+.00003246;
clear
% Properties of 1/2" Steel Shaft
E=229000000;
I=.00307;
%Begin Loop
x=1;
while x<180
    if x<15
        Mx1=0;
        y1=(Mx1*((x/100)^2))/(2*E*I)-.00004263*(x/100)+.00003246;
    elseif (x>=15)&(x<35)
        Mx1=6.7*(x/100)-10.05;
        y1=(Mx1*((x/100)^2))/(2*E*I)-.00004263*(x/100)+.00003246;
    elseif (x>=35)&(x<122.5)
        Mx1=-.698*(x/100)+14.693;
        y1=(Mx1*((x/100)^2))/(2*E*I)-.00004263*(x/100)+.00003246;
    elseif (x>=122.5)&(x<145)
        Mx1=20.056*(x/100)-10.02;
        y1=(Mx1*((x/100)^2))/(2*E*I)-.00004263*(x/100)+.00003246;
    elseif (x>=145)&(x<165)
        Mx1=25.15*(x/100)-17.08;
        y1=(Mx1*((x/100)^2))/(2*E*I)-.00004263*(x/100)+.00003246;
    elseif (x>=165)&(x<180)
        Mx1=.206*(x/100)+24.15;
        y1=(Mx1*((x/100)^2))/(2*E*I)-.00004263*(x/100)+.00003246;
    else
    end
a(x)=y1;
x=x+1;
clear y1;
end
% Create Time Vector, b:
b=[0:.1:17.8];
% Plot
plot (b,a)
```

Attachment 7: Final Cost Ledger

Component Name	Sub-System	Qty	Cost Each	Total Cost
Speakers				**$189.14**
Arduino Uno (Slave)	Speakers	1	$27.95	$27.95
Adafruit Wave Shield	Speakers	1	$22.00	$22.00
MOSFETS	Speakers	5	$0.41	$2.05
Stereo Amplifier Breakout Board	Speakers	1	$9.95	$9.95
PCB Board	Speakers	1	$4.50	$4.50
Pyle PLMR24 Hydra 200 W Speaker	Speakers	4	$17.49	$69.96
16 G Outdoor Speaker Wire(150ft)	Speakers	1	$43.99	$43.99
4 GB SD Card	Speakers/Camera	2	$4.37	$8.74
Rangefinder				**115.15**
Standard Servo w/ Hardware	Rangefinder	1	$12.00	$12.00
Weatherproof Ultra-Sonic Rangefinder	Rangefinder	1	$84.95	$84.95
Rangefinder Servo Mount	Rangefinder	1	$1.96	$1.96
PVC Elbow	Rangefinder	1	$0.44	$0.44
White Grease	Rangefinder/Shakers	1	$4.28	$4.28
1" Wooden Pole	Rangefinder	1	$9.00	$9.00
Servo Mount Clamp	Rangefinder	1	$1.02	$1.02
#4 SS Screws (8 Pack)	Rangefinder	1	$1.50	$1.50
Eyes				**118.81**
10mm Green Diffuse LEDs (10 Pack)	Eyes	1	$9.95	$9.95
4-Wire Connector	Eyes/Shake/Sprinkler	9	$2.50	$22.50
1/8" 6"x6" UV Acrylic	Eyes	8	$2.14	$17.12
1/2" 6"x6" UHMW	Eyes	4	$5.69	$22.76
U Bolts	Eyes	8	$1.31	$10.48
6 ft. 1"D. Wooden Pole	Eyes	4	$9.00	$36.00
Shakers				**113.14**
Adafruit 16 Channel PWM Driver	Shake/Eyes	1	$14.95	$14.95
Fake Plants	Shake	12	$1.00	$12.00
Towerpro SG-5010 Servo w/ Hardware	Shake	4	$12.00	$48.00
Hinged Threaded Standoff	Shake	4	$3.18	$12.72
Straight Threaded Standoff	Shake	16	$1.29	$20.64
#8-32 x 1/2" Machine Screw (4 pack)	Shake	2	$1.50	$3.00
Wire Hook (1/16" Diam x 3'L)	Shake	1	$1.83	$1.83
Control				**$162.10**
Arduino Uno	Control	1	$27.95	$27.95
Arduino Mega	Control	1	$37.74	$37.74
Enclosure Case	Control	1	$23.52	$23.52
Power Strip	Control	1	$11.98	$11.98

Arduino Power Supply	Control	1	$6.99	$6.99
Anti-static foam	Control	1	$2.74	$2.74
Mounting Hardware	Control	1	$3.00	$3.00
10kOhm Resistor (pack of 10)	Control	1	$0.10	$0.10
220 Ohm Resistor (pack of 10)	control	1	$0.10	$0.10
4 Connector Wire (100ft)	Control	1	$32.99	$32.99
2 Connector Wire (50ft)	Control	1	$14.99	$14.99
Notification of Errors				**$30.06**
Arduino Ethernet Shield	Control	1	$30.06	$30.06
Data Verification				**$117.20**
Bushnell X-8 6MP Trail Camera	Camera	1	$117.20	$117.20
Sprinkler/PIR Sensor				**$69.97**
Yard Enforcer	Sprinkler	1	$69.97	$69.97
				915.57

```
/* This program is the source code for the Deer Busters Senior Project from
the mechanical engineering department at California Polytechnic State
Unviersity.  It was written by student Garrett Tietz to be used by hardware
constructed by students Dane Knutson and Adam Webb in 2014.  The project was
sponsored by coorporate sponsor JumpSport. This code and the hardware
are property of JumpSport.


Intended for Arduino Mega

Hardware
------------------
Rangefinder
  -PWM Output for Servo
      The IC will sweep the servo position over 180 degrees so allow for
      distance readings at each angle.
  -Analog Input for distance data
      The IC will read the distance value from the rangefinder over the
      analog input line each specified PWM location
Shaker/Eyes
  -Direct Wiring to Servos and Eyes from PWM outputs
      The IC will specify which servo/LED will be lit at specified times
      to simulate the the approach of a predator.  The servos will simulate
      a shaking motion and the LEDs will be turned on to simulate eyes.

See product documentation for more information


Tasks & States
------------------
Task: Mastermind (MM)
    [In progress]






Pinouts
------------------
Output Pins:
      3: Rangefinder Servo Output (PWM)
      4: Eye 1 Output (PWM)
      5: Eye 2 Output (PWM)
      6: Eye 3 Output (PWM)
      7: Eye 4 Output (PWM)
      8: Shaker 1 Output (PWM)
      9: Shaker 2 Output (PWM)
     10: Shaker 3 Output (PWM)
     11: Shaker 4 Output (PWM)

Input Pins:
     A1: Rangefinder Distance Input (Analog)
```

```
*/

//Includes
#include <Servo.h>

//Create Objects
Servo shakeservo1;      //create objects to control servos (max of 8)
Servo shakeservo2;
Servo shakeservo3;
Servo shakeservo4;
Servo rangeservo;

//Set Constants
const byte servoMin = 20;
const byte servoMax = 40;
const byte shakeTime = 2;
const byte eye1Pin = 4;         //4
const byte eye2Pin = 5;         //5
const byte eye3Pin = 6;         //6
const byte eye4Pin = 7;         //7
const byte shake1Pin = 8;       //8
const byte shake2Pin = 9;       //9
const byte shake3Pin = 10;      //10
const byte shake4Pin = 11;      //11
const byte rangeServoPin = 3;   //3
const byte rangeDistPin = A1;   //A1
const byte rangeMin = 45;
const byte rangeMax = 135;
const byte rangeSensitivity = 75;       //bigger number means less sensitive
const byte rangeIncs = 10;
const byte rangeTime = 1000;                //delay time between rangefinder data points

//Initialize Shared Variables
unsigned int deterTime;         //stores duration of on time in loops
byte rangerFlag;
byte modeFlag = 0;
byte uiShakeOn = 0;     //stores mode of shaker (on/off/single)
byte uiEyeOn = 0;           //stores mode of eyes (on/off/single)
byte eyeFlag = 0;
byte shakeFlag = 0;
byte motionRecord = 0;      //storage for number of times motion is detected

//Initialize Local Variables
byte mmState = 0;
byte rangerState = 0;
byte uiState = 0;
byte eyeState = 0;
byte shakeState = 0;
byte shakeCount = 0;
byte shakeDir = 0;
byte rangePos = 0;
int oldRanges[100];
int newRanges[100];
```

```
int difRange;
int rangerDelay = 0;

char inByte;
int cancelCount = 0;
int deterCount = 0;


void setup()
{
    //Initialize Serial communication
    Serial.begin(9600);     //set baud rate
    Serial.flush();
    Serial.println("Starting");
    Serial.flush();

    //set deter duration
    deterTime = 100;
}

void loop()
{
    //Serial.print(F("Free RAM: "));
    //Serial.println(freeRam());

    mmTask();
    uiTask();
    rangerTask();
    eyeTask();
    shakeTask();

    //Serial.println(F("Working"));
    //Serial.flush();
    //Serial.println(mmState);
    //Serial.println(rangerState);
    //Serial.println(uiState);
    //Serial.println(eyeState);
    //Serial.println(shakeState);
    //Serial.println(rangerFlag);
    delay(50);
}

int freeRam () {
  extern int __heap_start, *__brkval;
  int v;
  return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);
}


/*
====================
Tasks
====================
```

```
*/


/*
--------------------
Task: Mastermind
This task handles all of the "thinking" in the program.  It coordinates
with all other tasks giving instructions and interpreting information.
*/

void mmTask()
{
    //MM State 0: Initialize
    if (mmState == 0)
    {
        //No setup required

        mmState = 7;    //set next state mode select
    }

    //MM State 1: Active Mode Hub
    else if (mmState == 1)
    {
        //check for motion from rangefinder
        if (rangerFlag == 1)    //motion detected
        {
            rangerFlag = 3; //tell rangefinder to stop looking
            mmState = 2;    //set next state to deter on
        }
    }

    //MM State 2: Activate Deterrence(s)
    else if (mmState == 2)
    {
        if (modeFlag == 1)
        {
            motionRecord = motionRecord + 1;                //increment counter
        }

        //Check if shakers are active
        if (uiShakeOn == 1) //single shaker is active
        {
            shakeFlag = 1;  //activate closest
        }
        else if (uiShakeOn == 2)    //multi shaker is active
        {
            shakeFlag = 4;  //activate farthest away
        }
        else
        {
            shakeFlag = 0;
        }
```

```
        //Check if eyes are active
        if (uiEyeOn == 1)       //single eyes are active
        {
            eyeFlag = 1;    //activate closest
        }
        else if (uiEyeOn == 2)  //multi eyes are active
        {
            eyeFlag = 4;    //activate farthest away
        }
        else
        {
            eyeFlag = 0;
        }

        mmState = 3;    //set next state to deter timer
    }

    //MM State 3: Deter timer 1/4
    else if (mmState == 3)
    {
        if (uiShakeOn == 0) shakeFlag = 0;
        if (uiEyeOn == 0) eyeFlag = 0;
        if (modeFlag == 0) rangerFlag = 3;

        deterCount++;
        if (deterCount > (0.25*deterTime))
        {
            deterCount = 0;     //reset counter
            mmState = 4;        //set next state to timer 2/4

            //Check if shakers are active
            if (uiShakeOn == 2) //shaker is active
            {
                shakeFlag = 3;
            }
            //no effect on single shaker

            //Check if eyes are active
            if (uiEyeOn == 2)       //eyes are active
            {
                eyeFlag = 3;    //activate second eye set
            }
            //no effect on single eyes
        }
    }

    //MM State 4: Deter timer 2/4
    else if (mmState == 4)
    {
        if (uiShakeOn == 0) shakeFlag = 0;
        if (uiEyeOn == 0) eyeFlag = 0;
        if (modeFlag == 0) rangerFlag = 3;
```

```
        deterCount++;
        if (deterCount > (0.25*deterTime))
        {
            deterCount = 0;      //reset counter
            mmState = 5;         //set next state to timer 3/4

            //Check if shakers are active
            if (uiShakeOn == 2) //shaker is active
            {
                shakeFlag = 2;
            }
            //no effect on single shaker

            //Check if eyes are active
            if (uiEyeOn == 2)        //eyes are active
            {
                eyeFlag = 2;
            }
            //no effect on single eyes

        }
    }

    //MM State 5: Deter timer 3/4
    else if (mmState == 5)
    {
        if (uiShakeOn == 0) shakeFlag = 0;
        if (uiEyeOn == 0) eyeFlag = 0;
        if (modeFlag == 0) rangerFlag = 3;

        deterCount++;
        if (deterCount > (0.25*deterTime))
        {
            deterCount = 0;      //reset counter
            mmState = 6;         //set next state timer 4/4

            //Check if shakers are active
            if (uiShakeOn == 2) //shaker is active
            {
                shakeFlag = 1;
            }
            //no effect on single shaker

            //Check if eyes are active
            if (uiEyeOn == 2)        //eyes are active
            {
                eyeFlag = 1;
            }
            //no effect on single eyes
        }
    }

    //MM State 6: Deter Timer 4/4 (all off)
```

```
    else if (mmState == 6)
    {
        if (uiShakeOn == 0) shakeFlag = 0;
        if (uiEyeOn == 0) eyeFlag = 0;
        if (modeFlag == 0) rangerFlag = 3;

        deterCount++;
        if (deterCount > (0.25*deterTime))
        {
            deterCount = 0;     //reset counter

            if (modeFlag == 1)      //if active mode
            {
                mmState = 1;        //set next state to hub
                rangerFlag = 0;     //Tell rangefinder to start looking
            }
            else if (modeFlag == 2) //if debug mode
            {
                mmState = 2;        //set next state to activate
            }
            eyeFlag = 0;        //Tell eyes to turn all off
            shakeFlag = 0;      //Tell shakers to turn all off
        }
    }

    //MM State 7: Mode Select
    else if (mmState == 7)
    {
        if (modeFlag == 1)  //active mode
        {
            mmState = 1;    //set next state to active mode hub
            rangerFlag = 0; //command rangefinder on
        }
        else if (modeFlag == 2) //Debugg mode
        {
            mmState = 2;    //set next state to activate
            rangerFlag = 3; //do not turn on rangefinder
        }
        else if (modeFlag == 3) //Rangefinder mode
        {
            mmState = 8;    //set next state to rangefinder
            rangerFlag = 0; //turn on rangefinder
        }
        //otherwise just wait for a mode to be set by UI
    }

    //MM State 8: Rangefinder mode
    else if (mmState == 8)
    {
        if (modeFlag != 3)          //if no longer in rangefinder mode
            {
                mmState = 7;
                rangerFlag = 3;
```

```
            }
    }


    //MM Fall Through State
    else
    {
        Serial.println("Error, bad Mastermind State");
        while(1){}      //Stop execution
    }
}


/*
--------------------
Task: User Interface
This task handles all communication with the user.  It both sends
information through the serial port on the instruction from Mastermind
and receives input and relays the information back to Mastermind.

The user interface task does not strictly follow the rules of cooperative
multitasking as it waits for inputs in a while l
*/


void uiTask()
{
    //ensure that the enter key is not counted as an entry
    if (Serial.peek() == 13)    //see if enter is next in the buffer
    {
        Serial.read();      //read from buffer to eliminate
    }

    //UI State 0: Initialize
    if (uiState == 0)
    {
        uiState = 2;    //start with printing the main menu
        modeFlag = 0;   //ensure that no mode is selected

        //start with all methods off
        uiShakeOn = 0;
        uiEyeOn = 0;
    }

    //UI State 1: Hub (look for inputs)
    else if (uiState == 1)
    {
        inByte = Serial.read();
        if (inByte == 'x')
        {
            Serial.println(F("\nStopping."));
            uiState = 9;    //return to initialize state
```

```
            //turn off everything
            uiShakeOn = 0;
            uiEyeOn = 0;
            modeFlag = 0;
        }
    }

    //UI State 2: Main Menu
    else if (uiState == 2)
    {
        Serial.println("\nMain Menu");
        Serial.flush();
        Serial.println("---------");
        Serial.flush();
        Serial.println("a: Active Mode");
        Serial.flush();
        Serial.println("d: Debug Mode");
        Serial.flush();
        Serial.println("r: Range Finder Mode");
        Serial.flush();
        Serial.println("s: Show Records");
        Serial.flush();
        Serial.println("i: Information");
        Serial.flush();
        Serial.println("h: Help");
        Serial.flush();         //wait for transmission
        Serial.println(" ");
        Serial.flush();

        modeFlag = 0;       //ensure that no mode is set

        //flush the buffer
        while (Serial.available() > 0)
        {
            Serial.read();
        }

        //wait for input
        while (Serial.available() == 0)
        {
            //do nothing
        }

        //do something with the input
        inByte = Serial.read();
        switch (inByte)
        {
            case 'a':   //active mode selected
                uiState = 3;    //set next state to active mode display
            break;

            case 'd':   //debug mode selected
                uiState = 6;    //set next state to debug mode display
```

```
        break;

        case 'r':   //rangefinder mode selected
            uiState = 12;   //set next state to rangefinder mode
        break;

        case 's':   //show records
            uiState = 10;   //set next state to show records mode display
        break;

        case 'i':   //information selected
            //Display info message
            Serial.println("\nThis project was created by Mechanical Engineering");
            Serial.flush();
            Serial.println("students Garrett Tietz, Dane Knutson and Adam Webb");
            Serial.flush();
            Serial.println("from California Polytechnic State University");
            Serial.flush();
            Serial.println("as a Senior Project in Jan-Nov 2014 for JumpSport.");
            Serial.flush();
            Serial.println("\nGO MUSTANGS!");
            Serial.flush();
            uiState = 2;        //remain in main menu state
            delay(2000);
        break;

        case 'h':       //help selected
            //display help message
            Serial.println("\nSelect desired operation mode by entering the");
            Serial.flush();
            Serial.println("key that corresponds to the desired selection");
            Serial.flush();
            uiState = 2;        //remain in main menu state
            delay(2000);
        break;

        default:
            Serial.println("\nInvalid. Try agian.");
            Serial.flush();
    }
}   //end ui state 2

//UI State 3: Active mode menu
else if (uiState == 3)
{
    Serial.println("\nActive Mode Menu");
    Serial.flush();
    Serial.println("---------------------------");
    Serial.flush();
    Serial.println("Select Deterrence Methods");
    Serial.flush();
    Serial.println("a: Shakers");
    Serial.flush();
```

```
        Serial.println("e: Eyes");
        Serial.flush();
        Serial.println("d: Done Selecting");
        Serial.flush();
        Serial.println("h: Help");
        Serial.flush();
        Serial.println("x: Stop and Go Back");
        Serial.flush();
        Serial.println(" ");
        Serial.flush();

        uiState = 4;    //set next state to active mode receive
    }

    //UI State 4: Active Mode receive
    else if (uiState == 4)
    {
        //flush the buffer
        while (Serial.available() > 0)
        {
            Serial.read();
        }

        //wait for input
        while (Serial.available() == 0)
        {
            //do nothing
        }


        //do something with the input
        inByte = Serial.read();
        switch (inByte)
            {
                case 'x':   //back selected
                    uiState = 2;

                    //reset all methods  to off
                    uiShakeOn = 0;
                    uiEyeOn = 0;

                    uiState = 2;
                break;

                case 'h':
                    //Display Help
                    Serial.println("\nSelect the Deterrence methods to enable");
                    Serial.flush();
                    Serial.println("and press 'd' when finished.");
                    Serial.flush();
                    delay(2000);
                    uiState = 3;
                break;
```

```
        case 'd':
            //Done selecting

            //Print confirmation message
            Serial.println("\nActive Deterrence Methods:");
            if (uiShakeOn == 1)
            {
                Serial.println("\tSingle Shaker");
                Serial.flush();
            }
            else if (uiShakeOn == 2)
            {
                Serial.println("\tMultiple Shakers");
                Serial.flush();
            }

            if (uiEyeOn == 1)
            {
                Serial.println("\tSingle Eye Set");
                Serial.flush();
            }
            else if (uiEyeOn == 2)
            {
                Serial.println("\tMultiple Eye Sets");
                Serial.flush();
            }

            if (uiEyeOn == 0 && uiShakeOn == 0)
            {
                Serial.println("\tNone");
                Serial.flush();
            }

            Serial.println("Confirm and enable? (y/n)");
            Serial.flush();
            uiState = 5;    //set next state to active confirm

        break;

        case 'a':
            //Shaker
            if (uiShakeOn == 0)     //if shaker is off
            {
                uiShakeOn = 1;      //set one shaker on
                Serial.println("Single Shaker Enabled");
                Serial.flush();
                Serial.println("\tSelect Again for Multiple");
                Serial.flush();
            }
            else if (uiShakeOn == 1)    //if one shaker is on
            {
                uiShakeOn = 2;      //set multiple shakers on
```

```
                    Serial.println("Multiple Shakers Enabled.");
                    Serial.flush();
                    Serial.println("\tSelect Again to Disable");
                    Serial.flush();
                }
                else if (uiShakeOn == 2)     //if all shakers on
                {
                    uiShakeOn = 0;       //set to off
                    Serial.println("All Shakers Disabled");
                    Serial.flush();
                }
                uiState = 3;
            break;

            case 'e':
                //Eyes
                if (uiEyeOn == 0)        //if eye is off
                {
                    uiEyeOn = 1;         //set one eye set on
                    Serial.println("Single Eye Set Enabled");
                    Serial.flush();
                    Serial.println("\tSelect Again for Multiple");
                    Serial.flush();
                }
                else if (uiEyeOn == 1)  //if one eye is on
                {
                    uiEyeOn = 2;         //set multiple eyes on
                    Serial.println("Multiple Eye Sets Enabled.");
                    Serial.flush();
                    Serial.println("\tSelect Again to Disable");
                    Serial.flush();
                }
                else if (uiEyeOn == 2)      //if all eyes on
                {
                    uiEyeOn = 0;         //set to off
                    Serial.println("All Eye Sets Disabled");
                    Serial.flush();
                }
                uiState = 3;
            break;

            default:
                Serial.println("\nInvalid. Try again.");
                Serial.flush();
        }
    } //end ui State 4

    //UI State 5: Active mode confirm
    else if (uiState == 5)
    {
        //flush the buffer
        while (Serial.available() > 0)
        {
```

```
            Serial.read();
        }

        //wait for input
        while (Serial.available() == 0)
        {
            //do nothing
        }
        inByte = Serial.read();
        switch (inByte)
        {
            case 'y':
                //Confirmed
                modeFlag = 1;       //Enable active mode
                Serial.println("\nActive Mode enabled.");
                Serial.flush();
                Serial.println("Enter 'x' at any time to cancel");
                Serial.flush();
                uiState = 1;        //go to hub
            break;

            case 'n':
                //Rejected
                uiState = 3;            //redisplay active mode menu
            break;

            default:
                //invalid
                Serial.println("\nInvalid. Try again.");
                Serial.flush();
        }
    } //end uiState 5

    //UI State 6: Debug Mode Menu
    else if (uiState == 6)
    {
        Serial.println("\nDebug Mode Menu");
        Serial.flush();
        Serial.println("---------------------------");
        Serial.flush();
        Serial.println("Select Deterrence Methods");
        Serial.flush();
        Serial.println("a: Shakers");
        Serial.flush();
        Serial.println("e: Eyes");
        Serial.flush();
        Serial.println("d: Done Selecting");
        Serial.flush();
        Serial.println("h: Help");
        Serial.flush();
        Serial.println("x: Go Back");
        Serial.flush();
        Serial.println(" ");
```

```
        Serial.flush();

        uiState = 7;    //set next state to debug mode receive
    }

    //UI State 7: Debug Mode Receive
    else if (uiState == 7)
    {
        //flush the buffer
        while (Serial.available() > 0)
        {
            Serial.read();
        }

        //wait for input
        while (Serial.available() == 0)
        {
            //do nothing
        }

        //do something with the input
        inByte = Serial.read();
        switch (inByte)
            {
                case 'x':   //back selected
                    uiState = 2;

                    //reset all methods  to off
                    uiShakeOn = 0;
                    uiEyeOn = 0;
                break;

                case 'h':
                    //Display Help
                    Serial.println("\nSelect the Deterrence methods to enable");
                    Serial.flush();
                    Serial.println("and press 'd' when finished.");
                    Serial.flush();
                    delay(2000);
                    uiState = 6;
                break;

                case 'd':
                    //Done selecting

                    //Print confirmation message
                    Serial.println("\nActive Deterrence Methods:");
                    if (uiShakeOn == 1)
                    {
                        Serial.println("\tSingle Shaker");
                        Serial.flush();
                    }
                    else if (uiShakeOn == 2)
```

```
        {
            Serial.println("\tMultiple Shakers");
            Serial.flush();
        }

        if (uiEyeOn == 1)
        {
            Serial.println("\tSingle Eye Set");
            Serial.flush();
        }
        else if (uiEyeOn == 2)
        {
            Serial.println("\tMultiple Eye Sets");
            Serial.flush();
        }

        if (uiEyeOn == 0 && uiShakeOn == 0)
        {
            Serial.println("\tNone");
            Serial.flush();
        }

        Serial.println("Confirm and enable? (y/n)");
        Serial.flush();
        uiState = 8;    //set next state to debug confirm

    break;

    case 'a':
        //Shaker
        if (uiShakeOn == 0)      //if shaker is off
        {
            uiShakeOn = 1;       //set one shaker on
            Serial.println("Single Shaker Enabled");
            Serial.flush();
            Serial.println("\tSelect Again for Multiple");
            Serial.flush();
        }
        else if (uiShakeOn == 1)     //if one shaker is on
        {
            uiShakeOn = 2;       //set multiple shakers on
            Serial.println("Multiple Shakers Enabled.");
            Serial.flush();
            Serial.println("\tSelect Again to Disable");
            Serial.flush();
        }
        else if (uiShakeOn == 2)     //if all shakers on
        {
            uiShakeOn = 0;       //set to off
            Serial.println("All Shakers Disabled");
            Serial.flush();
        }
        uiState = 6;
```

```
            break;

            case 'e':
                //Eyes
                if (uiEyeOn == 0)       //if eye is off
                {
                    uiEyeOn = 1;        //set one eye set on
                    Serial.println("Single Eye Set Enabled");
                    Serial.flush();
                    Serial.println("\tSelect Again for Multiple");
                    Serial.flush();
                }
                else if (uiEyeOn == 1)  //if one eye is on
                {
                    uiEyeOn = 2;        //set multiple eyes on
                    Serial.println("Multiple Eye Sets Enabled.");
                    Serial.flush();
                    Serial.println("\tSelect Again to Disable");
                    Serial.flush();
                }
                else if (uiEyeOn == 2)      //if all eyes on
                {
                    uiEyeOn = 0;        //set to off
                    Serial.println("All Eye Sets Disabled");
                    Serial.flush();
                }
                uiState = 6;
            break;

            default:
                Serial.println("\nInvalid. Try again.");
                Serial.flush();
        }
    } //end ui State 7

    //UI State 8: Debug Confirm
    else if (uiState == 8)
    {
    //flush the buffer
        while (Serial.available() > 0)
        {
            Serial.read();
        }

        //wait for input
        while (Serial.available() == 0)
        {
            //do nothing
        }
        inByte = Serial.read();
        switch (inByte)
        {
            case 'y':
```

```
                //Confirmed
                modeFlag = 2;       //Enable active mode
                Serial.println("\nDebug Mode enabled.");
                Serial.flush();
                Serial.println("Enter 'x' at any time to cancel");
                Serial.flush();
                uiState = 1;        //go to hub
            break;

            case 'n':
                //Rejected
                uiState = 6;            //redisplay debug mode menu
            break;

            default:
                //invalid
                Serial.println("\nInvalid. Try again.");
                Serial.flush();
        }
    } //end uiState 8


    //UI State 9: Cancel Everything
    else if (uiState == 9)
    {
            //turn off everything
            uiShakeOn = 0;
            uiEyeOn = 0;
            modeFlag = 0;

        //allow some time for the other tasks to turn everything off
        //necessary because of while loops in main menu
        if (cancelCount > 10)
        {
            uiState = 2;    //return to initialize state
            cancelCount = 0;
            Serial.println("Everything Off");
            Serial.flush();
        }
        cancelCount++;
    }

    //UI State 10: Show Records
    else if (uiState == 10)
    {
        Serial.println("\nRecords Display");
        Serial.flush();
        Serial.println("---------------------------");
        Serial.flush();
        Serial.print("Motions Sensed: ");
        Serial.println(motionRecord);
        Serial.flush();
        Serial.println("\nr: Reset count");
```

```
        Serial.flush();
        Serial.println("x: Go back to Main Menu");
        Serial.flush();
        Serial.println(" ");
        Serial.flush();

        uiState = 11;       //Set next state to Records receive
    }

    //UI State 11: Receive input for records
    else if (uiState == 11)
    {
        //flush the buffer
        while (Serial.available() > 0)
        {
            Serial.read();
        }

        //wait for input
        while (Serial.available() == 0)
        {
            //do nothing
        }

        //do something with the input
        inByte = Serial.read();
        switch (inByte)
        {
            case 'r':       //reset all values
                motionRecord = 0;
                uiState = 10;   //redisplay menu as confirmation
            break;

            case 'x':       //Go back to main menu
                uiState = 2;
            break;

            default:        //invalid
                Serial.println("\nInvalid. Try again.");
                Serial.flush();
        }
    }

    //UI State 12: Rangefinder Mode
    else if (uiState == 12)
    {
        modeFlag = 3;       //Enable rangefinder mode
        Serial.println("Rangefinder Mode enabled.");
        Serial.flush();
        Serial.println("Enter 'x' at any time to cancel");
        Serial.flush();
        delay(500);
        uiState = 1;        //go to hub
```

```
    }

    //UI State Fall-through
    else
    {
        Serial.println("Error, bad UI State");
        while(1){}      //Stop execution
    }
}   //end of uiTask()



/*
--------------------
Task: Ranger
This task handles the rangefinder motion
sensor using a PWM output pin to control
the sweeping servo and an analog input pin
to read the distance inputs.
input.
Requires:
States:     0 - Initialize
            1 - Wait (Off)
            2 - Initial Sweep
            3 - Looking
            4 - Found
            5 - Delay
Vars:       rangerState
            rangerFlag
            rangerDelay
            rangeMin
            rangeMax
            rangeServoPin
            rangeDistPin
            rangeSensitivity
            difRange
            oldRanges[10]
            newRanges[10]
            rangePos


*/

void rangerTask()
{
    //NOT A STATE
    //always check for stop command
    if (rangerFlag == 3)    //if MM says to stop
        {
            rangerState = 1;        //stop
        }

    //Ranger State 0: Initialize
```

```
    if (rangerState == 0)
    {
        rangeservo.attach(3);       //attach ranger servo object to pin 3
        rangeservo.write(rangeMin); //keep servo at zero location

        rangerFlag = 4;             //say I'm not looking
        rangerState = 1;            //set next state to wait
    }   //end of Ranger State 0

    //Ranger State 1: Wait (Off)
    else if (rangerState == 1)
    {
        rangeservo.write(rangeMin); //keep servo at zero location

        //check for instructions from MM
        if (rangerFlag == 0)        //MM wants to look
        {
            rangerState = 2;        //set next state to initial sweep
            rangePos = 0;           //reset ranger position counter
        }
    }   //end of Ranger State 1

    //Ranger State 2: Initial Sweep
    else if (rangerState == 2)
    {
        //move servo to new location
        rangeservo.write(rangePos*rangeIncs + rangeMin);

        delay(1000);        //wait for the servo to arrive
        //it needs a bit of extra time to come all the way across
        if (rangePos == 0)
            delay(2000);

        //take reading and store
        oldRanges[rangePos] = analogRead(rangeDistPin);

        rangePos++;     //increment position

        //check for end of range
        if (rangePos*rangeIncs+rangeMin > rangeMax)
        {
            rangePos = 0;
            rangerState = 3;        //start looking for changes next
        }
    }   //end of Ranger State 2

    //Ranger State 3: Looking
    else if (rangerState == 3)
    {
        //move servo to new location
        rangeservo.write(rangePos*rangeIncs + rangeMin);

        delay(1000);        //wait for the servo to arrive
```

```
        //it needs a bit of extra time to come all the way across
        if (rangePos == 0)
            delay(2000);

        //take reading
        newRanges[rangePos] = analogRead(rangeDistPin);
        //calculate difference from old reading
        difRange = newRanges[rangePos] - oldRanges[rangePos];
        //replace old reading with new for next time
        oldRanges[rangePos] = newRanges[rangePos];

        if (modeFlag == 1)      //active mode
        {
            if (abs(difRange) > rangeSensitivity)   //if motion
            {
                rangerFlag = 1;
                rangerState = 4;
            }
        }
        else if (modeFlag == 0) //no mode
            {
                //do nothing
            }
        else
        {
            //print everything out
            Serial.print(newRanges[rangePos]);

            if (abs(difRange) > rangeSensitivity)   //change found
                Serial.print(F("! "));      //signify with "!"
            else
                Serial.print(F("  "));      //don't signify

            //formatting help for varying number of digits
            if (newRanges[rangePos] < 100)
                Serial.print(F(" "));
            if (newRanges[rangePos] < 10)
                Serial.print(F(" "));
        }

        rangePos++;     //increment position

        if (rangePos*rangeIncs+rangeMin > rangeMax)
        {
            rangePos = 0;
            Serial.println(F(" "));
        }
    }   //end of Ranger State 3

    //Ranger State 4: Found
    else if (rangerState == 4)
    {
        //wait for MM to acknowledge
```

```
        if (rangerFlag = 5)
            rangerState = 5;
        else
            rangerFlag = 1; //be persistent
    }   //end of RangerState 4

    //Ranger State 5: Delay
    else if (rangerState == 5)
    {
        //wait for a while before turning back on
        rangerDelay++;
        if (rangerDelay > 10000)    //wait for a number of program loops
        {
            rangerDelay = 0;        //reset counter for next time
            rangerState = 1;        //set next state to perform initial sweep
        }
    }   //end of Ranger State 4

    else
    {
        Serial.println("Error, bad Ranger State");
        while(1){}      //Stop execution
    }

}//end of rangerTask()


/*
--------------------
Task: Eyes
This task controls the eyes sub-system LEDs using
digital I/O pins
Requires:
States:     0 - Initialize
            1 - Listen
Vars:       eyeState
            eyeFlag
            eye1Pin
            eye2Pin
            eye3Pin
            eye4Pin
*/

void eyeTask()
{
    //Eye state 0: Initialize
    if (eyeState == 0)
    {
        //set pins to outputs
        pinMode(eye1Pin, OUTPUT);
        pinMode(eye2Pin, OUTPUT);
        pinMode(eye3Pin, OUTPUT);
        pinMode(eye4Pin, OUTPUT);
```

```
        eyeFuncAllOff();        //start with all eyes off
        eyeState = 1;           //set next state to Listen
    }

    //Eye State 1: Listen
    if (eyeState == 1)
    {
        //stay in state 1 forever

        //wait for input from Mastermind
        if (eyeFlag == 0)    //MM wants all off
        {
            eyeFuncAllOff();
            eyeFlag = 5;     //done
        }
        else if (eyeFlag == 1)  //MM wants #1 on
        {
            eyeFuncAllOff();
            digitalWrite(eye1Pin, HIGH);
            eyeFlag = 5;     //done
        }
        else if (eyeFlag == 2)  //MM wants #2 on
        {
            eyeFuncAllOff();
            digitalWrite(eye2Pin, HIGH);
            eyeFlag = 5;     //done
        }
        else if (eyeFlag == 3)  //MM wants #3 on
        {
            eyeFuncAllOff();
            digitalWrite(eye3Pin, HIGH);
            eyeFlag = 5;     //done
        }
        else if (eyeFlag == 4)  //MM wants #4 on
        {
            eyeFuncAllOff();
            digitalWrite(eye4Pin, HIGH);
            eyeFlag = 5;     //done
        }
        //else do nothing

    }
    //Eye Fall through state
    else
    {
        Serial.println("Error, bad Eye State");
        while(1){}      //Stop execution
    }
}   //end of eyeTask()

void eyeFuncAllOff()
{
```

```
    digitalWrite(eye1Pin, LOW);
    digitalWrite(eye2Pin, LOW);
    digitalWrite(eye3Pin, LOW);
    digitalWrite(eye4Pin, LOW);
}   //end of eyeFuncAllOff()


/*
--------------------
Task: Shaker
This task controls the shaker sub-system servos
through the PWM output pins using the native
Servo library.
Requires:   #include <Servo.h>
States:     0 - Initialize
            1 - Hub (wait)
            2 - All off
            3 - Shaker 1 On
            4 - Shaker 2 On
            5 - Shaker 3 On
            6 - Shaker 4 On
Vars:       shakeState
            shakeFlag
            shakeCount
            shakeDir
            shakeTime    (const)
            servoMin     (const)
            servoMax     (const)
            servo1Pin
            servo2Pin
            servo3Pin
            servo4Pin

*/

void shakeTask()
{
    //Shaker State 0: Initialize
    if (shakeState == 0)
    {
        //Attach servo objects to pinouts
        shakeservo1.attach(shake1Pin);
        shakeservo2.attach(shake2Pin);
        shakeservo3.attach(shake3Pin);
        shakeservo4.attach(shake4Pin);

        shakeState = 1;
        }

    //Shaker State 1: Hub (wait)
    else if (shakeState == 1)
    {
        //do nothing but fall through to mastermind check
```

```
    }

    //Shaker State 2: All off
    else if (shakeState == 2)
    {
        shakeCount = 0;     //reset shake count to 0
        shakeState = 1;     //go back to hub and wait for instructions
    }

    //Shaker State 3: Servo 1 On
    else if (shakeState == 3)
    {
        shakeCount++;       //increment shakeCount

        if (shakeCount == shakeTime)
        {
            shakeCount = 0;         //reset counter to zero

            //change direction
            if (shakeDir > 0)
            {
                shakeDir = 0;
                shakeservo1.write(servoMax);        //go towards max
            }
            else
            {
                shakeDir = 1;
                shakeservo1.write(servoMin);    //go towards min
            }
        }
    }

    //Shaker State 4: Servo 2 On
    else if (shakeState == 4)
    {

        shakeCount++;       //increment shakeCount

        if (shakeCount == shakeTime)
        {
            shakeCount = 0;         //reset counter to zero

            //change direction
            if (shakeDir > 0)
            {
                shakeDir = 0;
                shakeservo2.write(servoMax);        //go towards max
            }
            else
            {
                shakeDir = 1;
                shakeservo2.write(servoMin);    //go towards min
            }
```

```
        }
    }

    //Shaker State 5: Servo 3 On
    else if (shakeState == 5)
    {

        shakeCount++;       //increment shakeCount

        if (shakeCount == shakeTime)
        {
            shakeCount = 0;         //reset counter to zero

            //change direction
            if (shakeDir > 0)
            {
                shakeDir = 0;
                shakeservo3.write(servoMax);        //go towards max
            }
            else
            {
                shakeDir = 1;
                shakeservo3.write(servoMin);    //go towards min
            }
        }
    }

    //Shaker State 6: Servo 4 on
    else if (shakeState == 6)
    {

        shakeCount++;       //increment shakeCount

        if (shakeCount == shakeTime)
        {
            shakeCount = 0;         //reset counter to zero

            //change direction
            if (shakeDir > 0)
            {
                shakeDir = 0;
                shakeservo4.write(servoMax);        //go towards max
            }
            else
            {
                shakeDir = 1;
                shakeservo4.write(servoMin);    //go towards min
            }
        }
    }
    //Shaker Fall through state
    else
    {
```

```
        Serial.println("Error, bad Shaker State");
        while(1){}      //Stop execution
    }

    //fall through to check for mastermind flags for all states

    //look for input from MM
    //Act accordingly
    if (shakeFlag == 0)     //MM says turn everything off
    {
        shakeState =  2;    //set next state to all off
    }
    else if (shakeFlag == 1)    //MM says activate #1
    {
        shakeState = 3;     //set next state to #1 on
    }
    else if (shakeFlag == 2)    //MM says activate #2
    {
        shakeState = 4;     //set next state to #2 on
    }
    else if (shakeFlag == 3)    //MM says activate #3
    {
        shakeState = 5;     //set next state to #3 on
    }
    else if (shakeFlag == 4)    //MM says activate #4
    {
        shakeState = 6;     //set next state to #4 on
    }
}   //end of shakerTask()
```

User Manual

Wildlife Deterrent Test Device

1) Installation

Setting up The Yard

When implementing the test system ensure that sufficient space is available for the test which will have minimal interference. The yard should be relatively level, though smooth scenery is not required.

Place each of the four platforms in the yard at distances less than 40, 50, 60 and 70 feet away from the control unit. The platforms should be arranged in a way that are successively getting nearer to the main test area. The most common configuration will be in a line away from the control unit.

Once the platforms are in position, place the Predator Eye units into the ground in a similar configuration as the platforms ensuring no more than 40, 50, 60 and 70 feet away from the control unit. Generally, the predator eye systems will be placed directly behind the shaker platforms, though this is not required. Hammer the stakes into the ground securely.

Next Place the rangefinder module into a position that such that the field of view is consistent with the location that the other components are oriented towards. Hammer the stake into the ground securely.

Wire the System

Ensure the control system is disconnected from the power source. Next find the wires which will reach each component and plug them into the corresponding component. Note that there are two types of extension wires: 2-Strand and 4-Strand. Any two and four strand wires are acceptable for use as long as the wires are long enough to reach all of the components desired. Plug in the other end of the wire to the lead coming out of the control box that is labeled to correspond with the correct component.

It is recommended to connect both ends of each wire sequentially to avoid confusion. Note that module 1 is intended to be the one closest to the animal with module 4 being the farthest away.

Once the system is connected as desired, plug in all components and turn on the power. Also plug the USB connector to the paired PC. This will provide power to the microcontroller. There may be a jerk from the servos when the system is first connected.

Prepare the Computer

Once the computer is connected to the microcontroller via the USB connector, open the Arduino IDE by clicking on the icon on the desktop. Once the program opens, click on the magnifying glass near the upper right corner to open the Serial monitor. This window is the window that will provide the user interface for the system. It is important to note that the microcontroller will reset whenever the serial window is opened.

After a few moments, the window will display the messages from the microcontroller. Once the Main Menu is displayed, the microcontroller is ready to receive inputs.

Enter information to the microcontroller by selecting the entry field towards the top of the window and entering the keys on the keyboard and pressing enter to transmit the data. Note that the microcontroller will only recognize the first character entered so that an entry of “turn on” will be interpreted as “t”.

2) Testing Components

Testing the components can be accomplished using the debug mode provided. Simply setup the system as described in Section 1 and select “Debug Mode” from the main menu. Next the specific deterrence methods can be chosen by entering the character that corresponds to the method desired. Selecting any method will instruct the system to activate that particular system in static mode, such that the number 1 module will be activated continuously. To enable the simulated approach method, select the same method again, instructing the system to migrate the action from module number 4 to module number 1. Select the same deterrence method again to disable it. Any combination of deterrence methods in either static or dynamic modes may be selected. Once the appropriate choices have been made, enter “d” and confirm by entering “y”. Alternatively, entering “n” at the confirmation will return to the debug menu.

Upon activation, the selected deterrence methods will operate continually until the user enters “x” to stop. At this time, the system can be inspected to ensure proper operation.

3) Activating System

Activating the system in Active Mode can be accomplished by selecting Active Mode from the Main Menu. Here the deterrence methods can be selected as in Debug Mode by entering the character which corresponds to the method desired and selecting “Done” to begin.

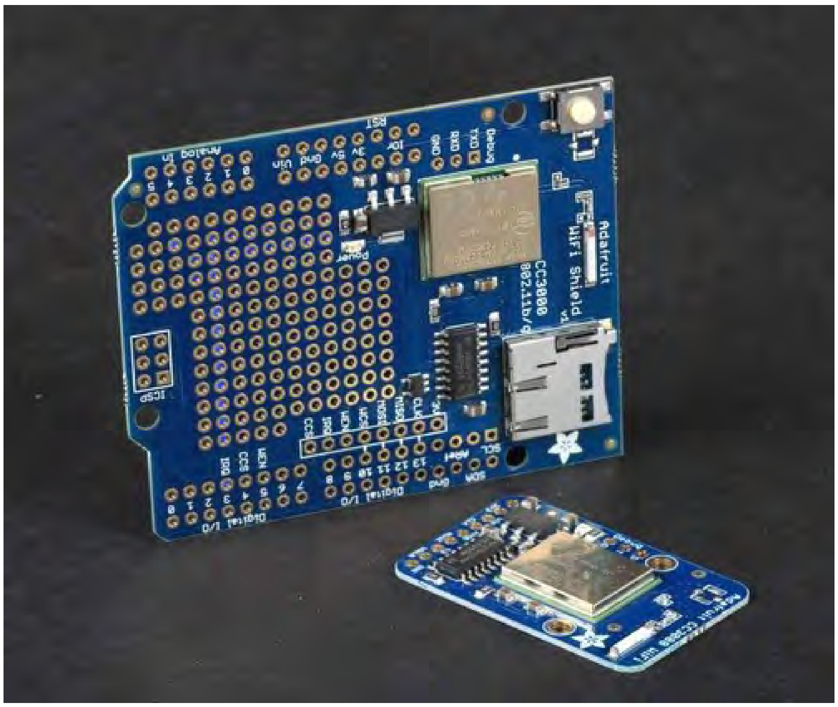
Once activated, the deterrence methods will not immediately move. Instead, the rangefinder will begin operation by taking an initial sweep of the area. After this first sweep the system will begin to detect motion by comparing the readings to the previous ones. If motion is detected, the rangefinder will be reset to the start position and the deterrence methods selected will be activated.

After a certain period of time, the deterrence method will stop and the rangefinder will perform another initial sweep of the yard. This process will repeat indefinitely until the user enters “x” to cancel or power is disconnected.

The number of times motion is detected is counted during the tests and can be displayed using the “Show Records” option in the Main Menu. This will display the number of activations performed and allow the user to reset the number to zero.

Adafruit CC3000 WiFi

Created by Rick Lesniak



Last updated on 2014-11-16 07:15:11 PM EST

Guide Contents

Overview

The CC3000 WiFi module from Texas Instruments is a small silver package which finally brings easy-to-use, affordable WiFi functionality to your Arduino projects.

It uses SPI for communication (not UART!) so you can push data as fast as you want or as slow as you want. It has a proper interrupt system with IRQ pin so you can have asynchronous connections. It supports 802.11b/g, open/WEP/WPA/WPA2 security, TKIP & AES. A built in TCP/IP stack with a "BSD socket" interface supports TCP and UDP in both client and server mode, with up to 4 concurrent socket connections.

The CC3000 does not support "AP" mode, it can connect to an access point but it cannot be an access point.

The CC3000 is available from Adafruit As a Breakout Board, and as an Arduino Shield.

Both the shield and the breakout board have an onboard 3.3V regulator that can handle the 350mA peak current, and a level shifter to allow 3 or 5V logic level. The antenna layout is identical to TI's suggested layout and we're using the same components, trace arrangement, and antenna so the board maintains its FCC emitter compliance (you'll still need to perform FCC validation for a finished product, but the WiFi part is taken care of). Even though it's got an onboard antenna we were pretty surprised at the range, as good as a smartphone's.

The shield also features a MicroSD socket, and a reset button.

AND, the shield supports the Arduino SPI passthrough header pins, so it's compatible with the Mega & Leonardo, right out of the box - no rewiring necessary!

Note the CC3000 library right now only supports the 1.0 series Arduino IDE (like 1.0.6) and not the 1.5.7 or above beta Arduino IDE!

Assembly and Wiring

Check out the next couple of pages for detailed instructions for setting up your CC3000 shield or breakout board!

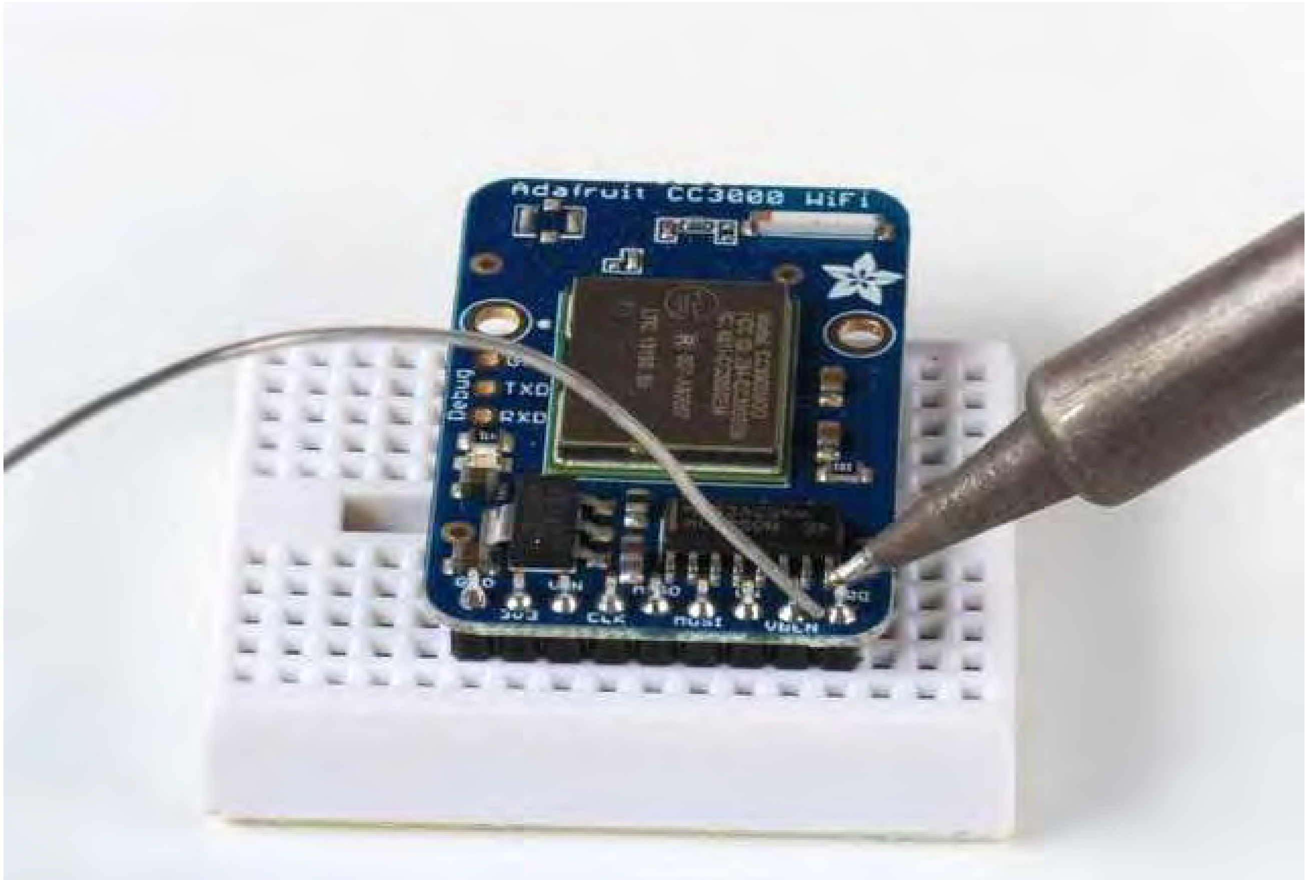
CC3000 Breakout

Assembly

The CC3000 breakout board ships with a strip of header pins. Snip off a 9-pin section and solder it to the 9 holes on the side of the board.

The easiest way to do this is to first insert the header pins into a breadboard, to hold them securely while you solder.

Set the breakout board over the pins, and carefully solder each pin (see our Guide To Excellent Soldering (http://adafru.it/aTk) for instructions and tips on getting the best results).

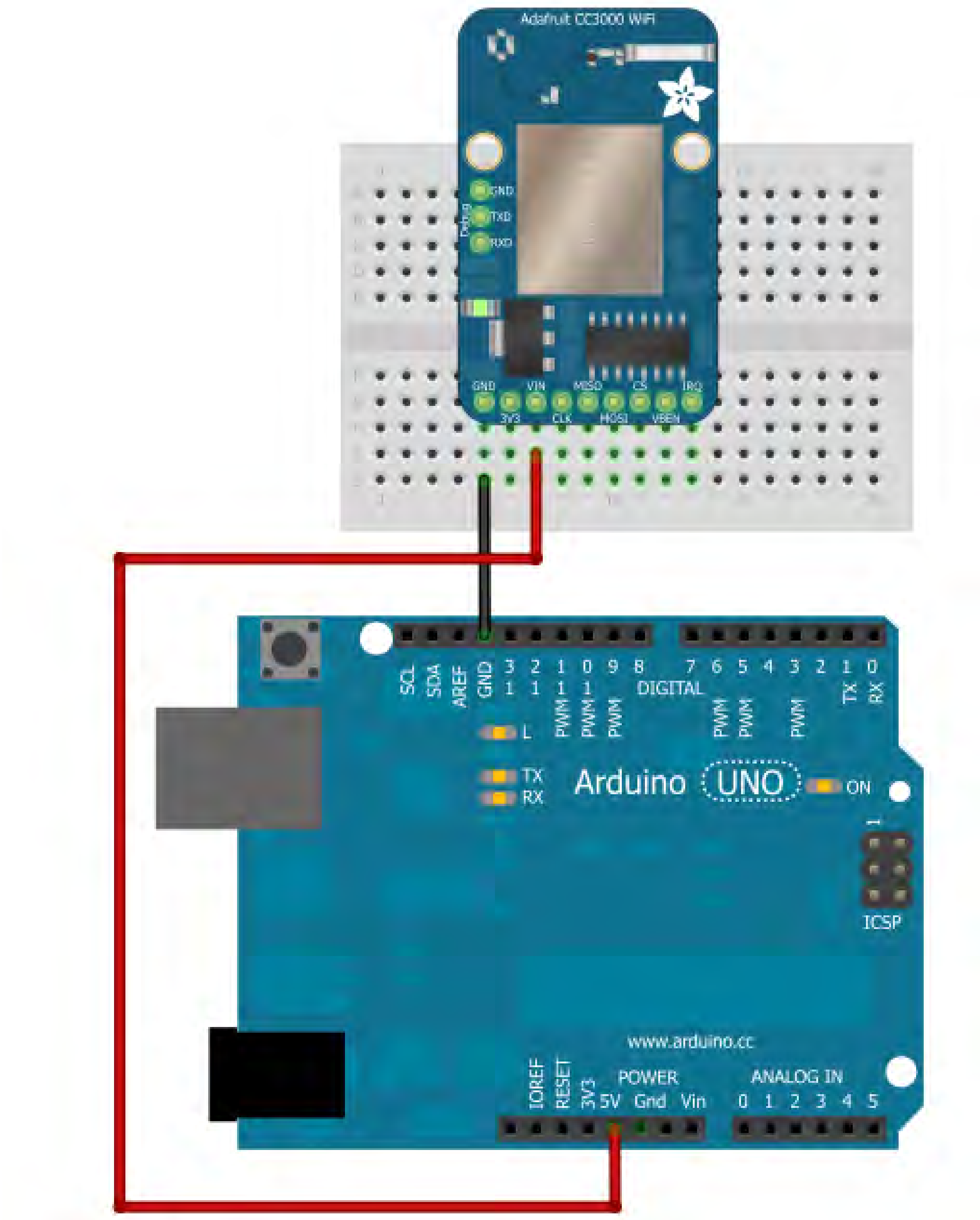


Wiring

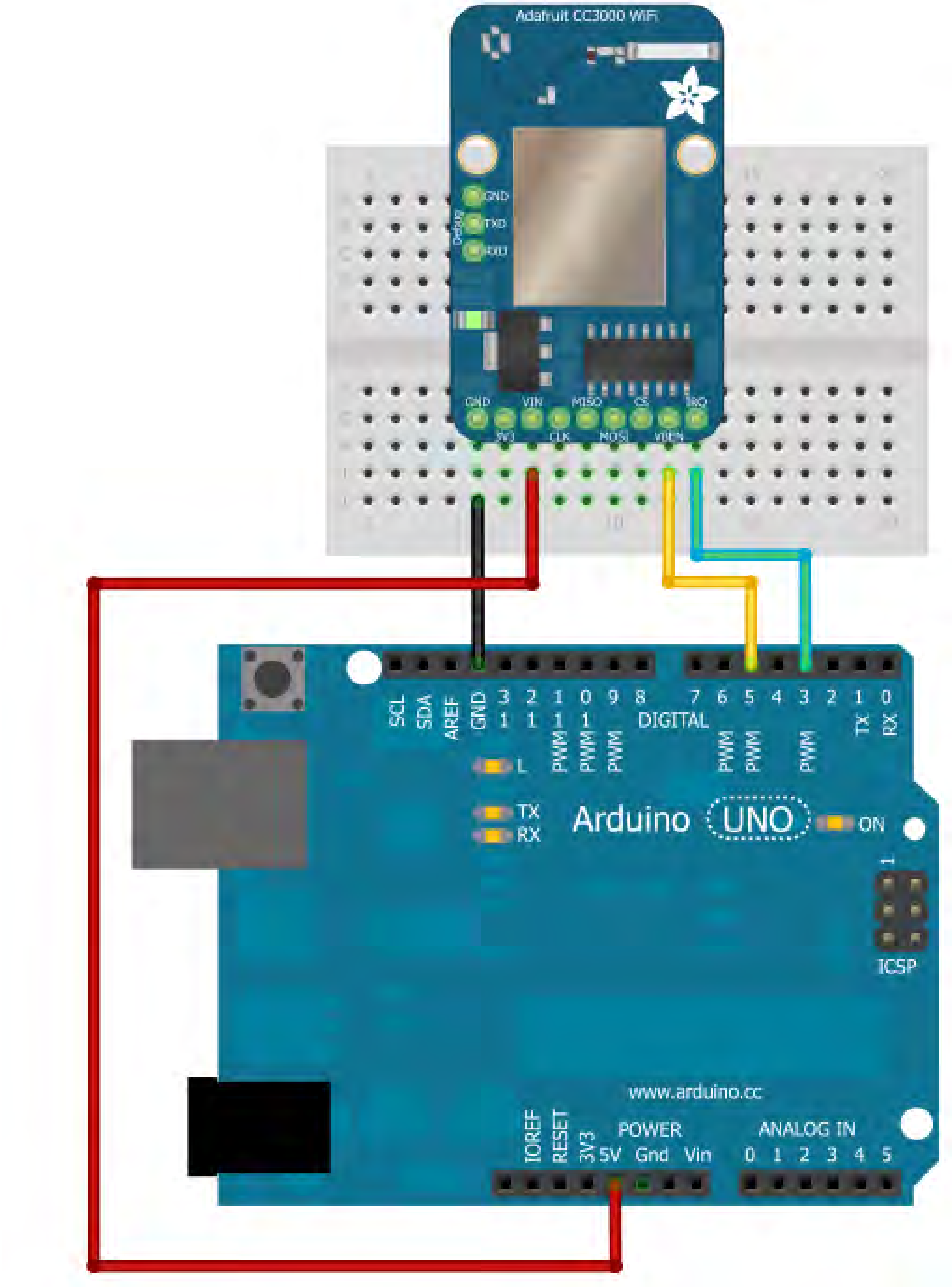
Use jumper wires to attach the CC3000 breakout board to your arduino:

Connect GND to one of the Arduino GND pins:
Connect Vin to Arduino +5V

NOTE: If using an Arduino Due, which is not tolerant of 5V on its input pins, you must instead connect the CC3000 3V3 pin to the Due's 3.3V power pin. Don't connect Vin to the Due's +5V!



Next, connect the enable and interrupt lines:
VBEN to Digital 5
IRQ to Digital 3



Now, connect SPI:

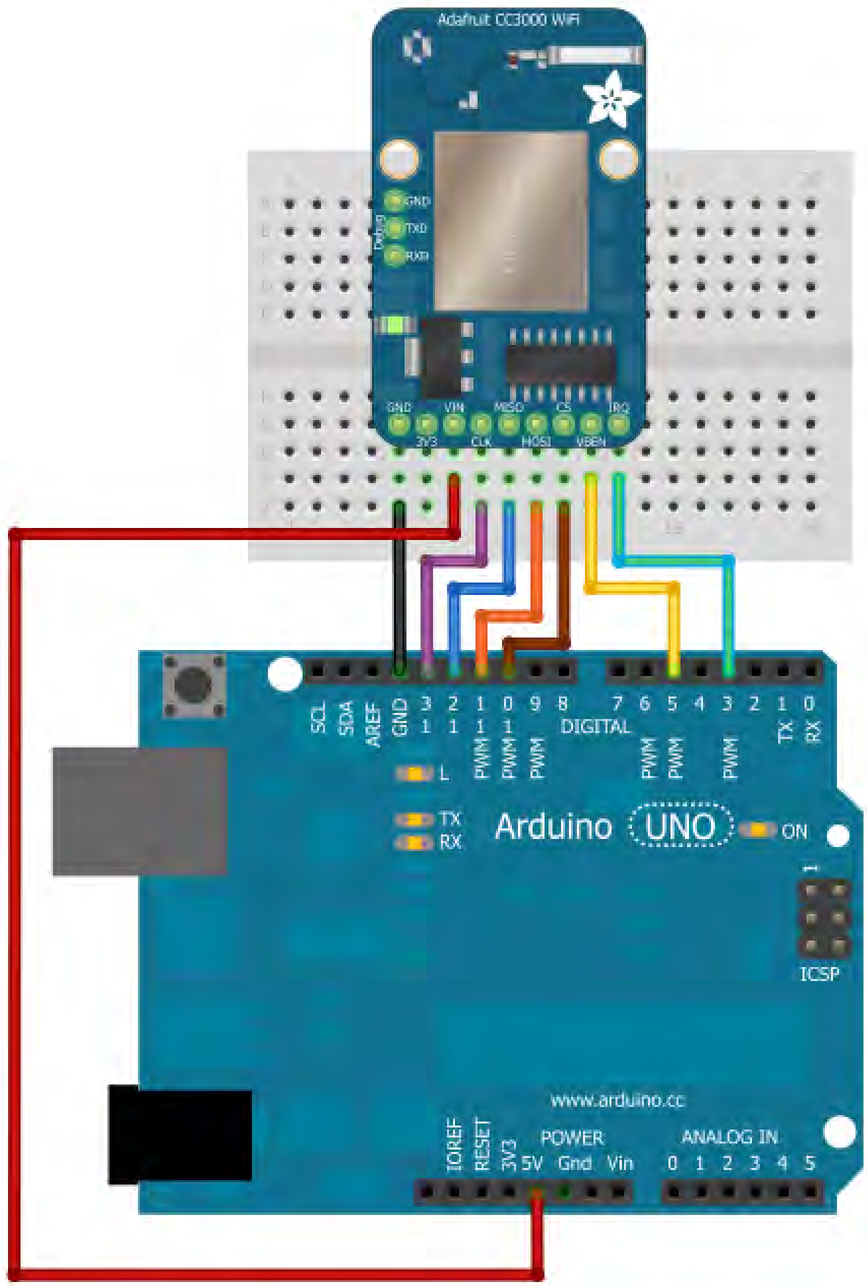
- CLK to Digital 13
- MISO to Digital 12
- MOSI to Digital 11
- CS to Digital 10

If you're using a Mega, you'll need to connect to the hardware SPI pins:

- CLK to Digital 52
- MISO to Digital 50
- MOSI to Digital 51
- CS to Digital 10

If you're using an Arduino Due, you'll need to connect to the hardware SPI pins. See the excellent diagram in this forum post (http://adafru.it/dVz) if you aren't sure where the hardware SPI pins are located on the Due. You want to connect to the SCK, MISO, and MOSI pins on the small 6 pin male header next to the Due's SAM3X8E processor:

- CLK to SPI SCK
- MISO to SPI MISO
- MOSI to SPI MOSI
- CS to Digital 10



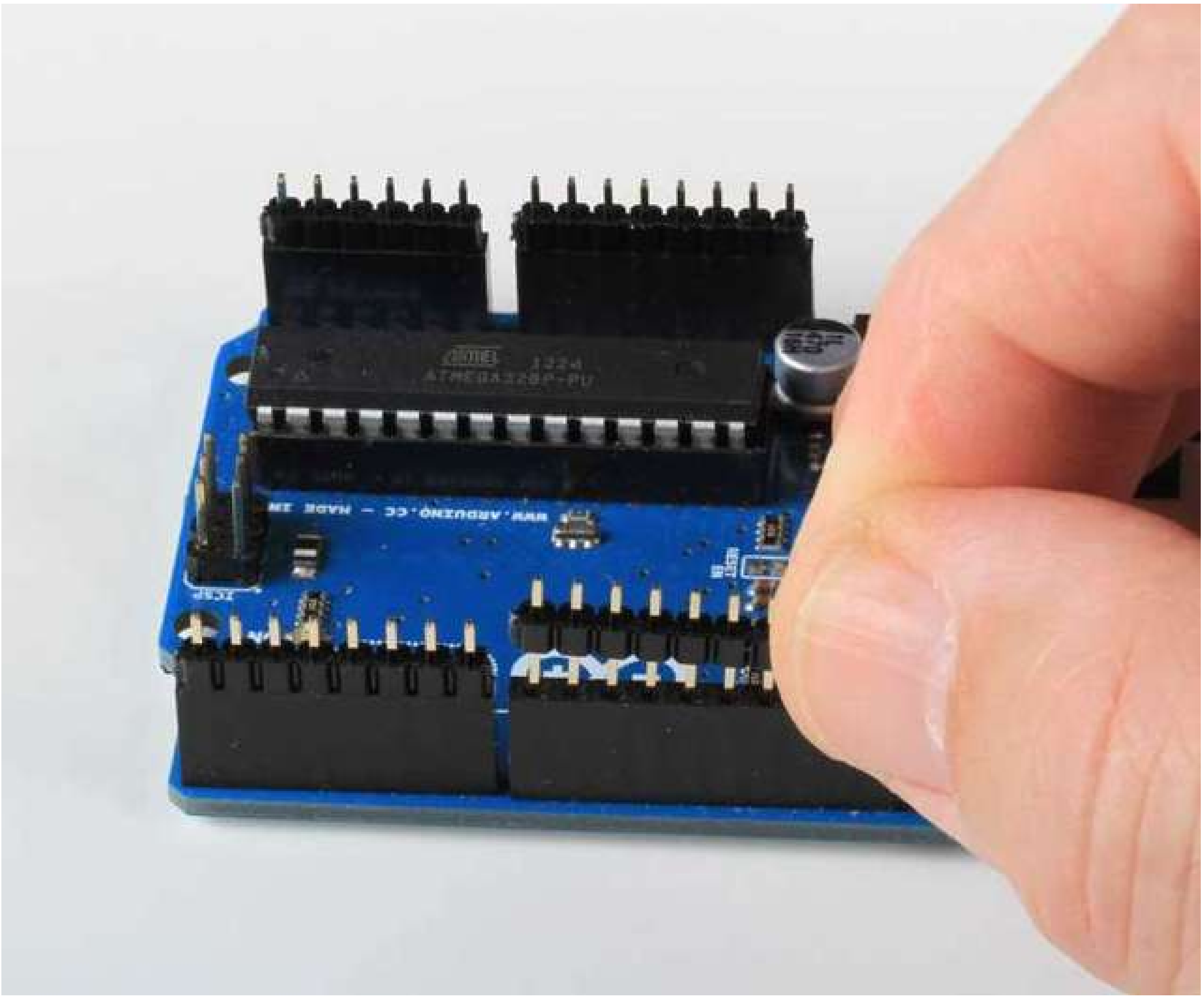
3.3v is an output from the breakout board's voltage regulator - we won't be connecting anything to it in this tutorial.

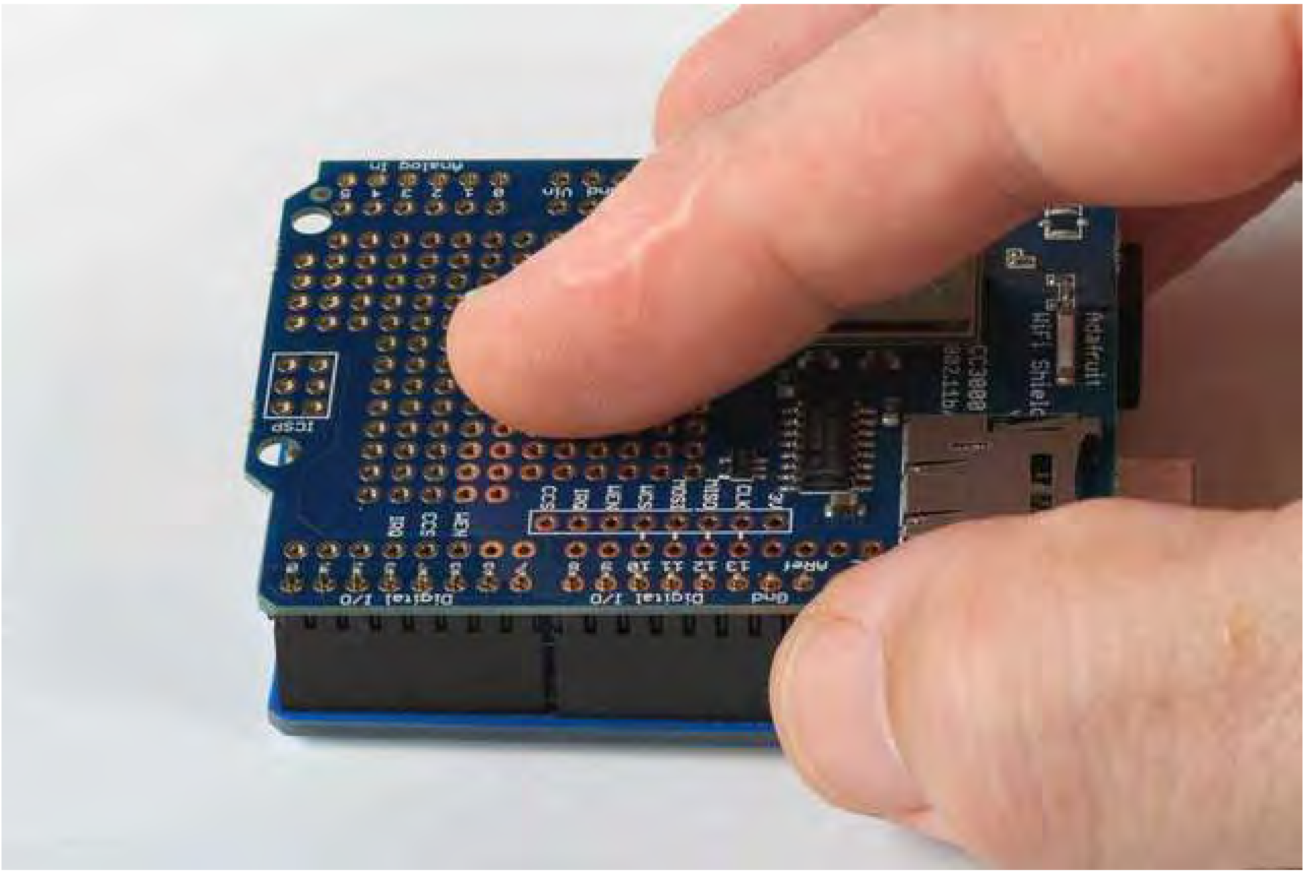
CC3000 Shield

Assembly

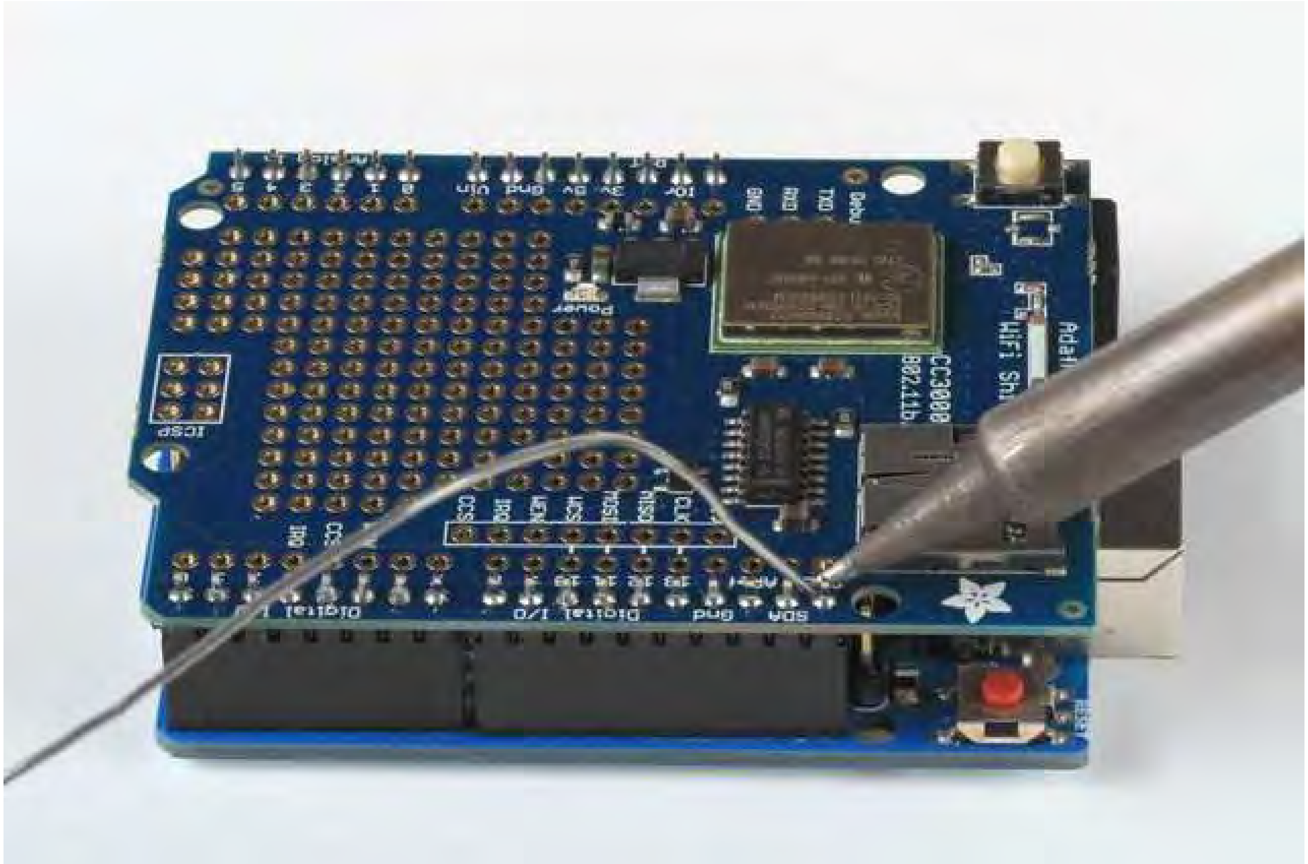
The CC3000 Shield ships with a strip of header pins.

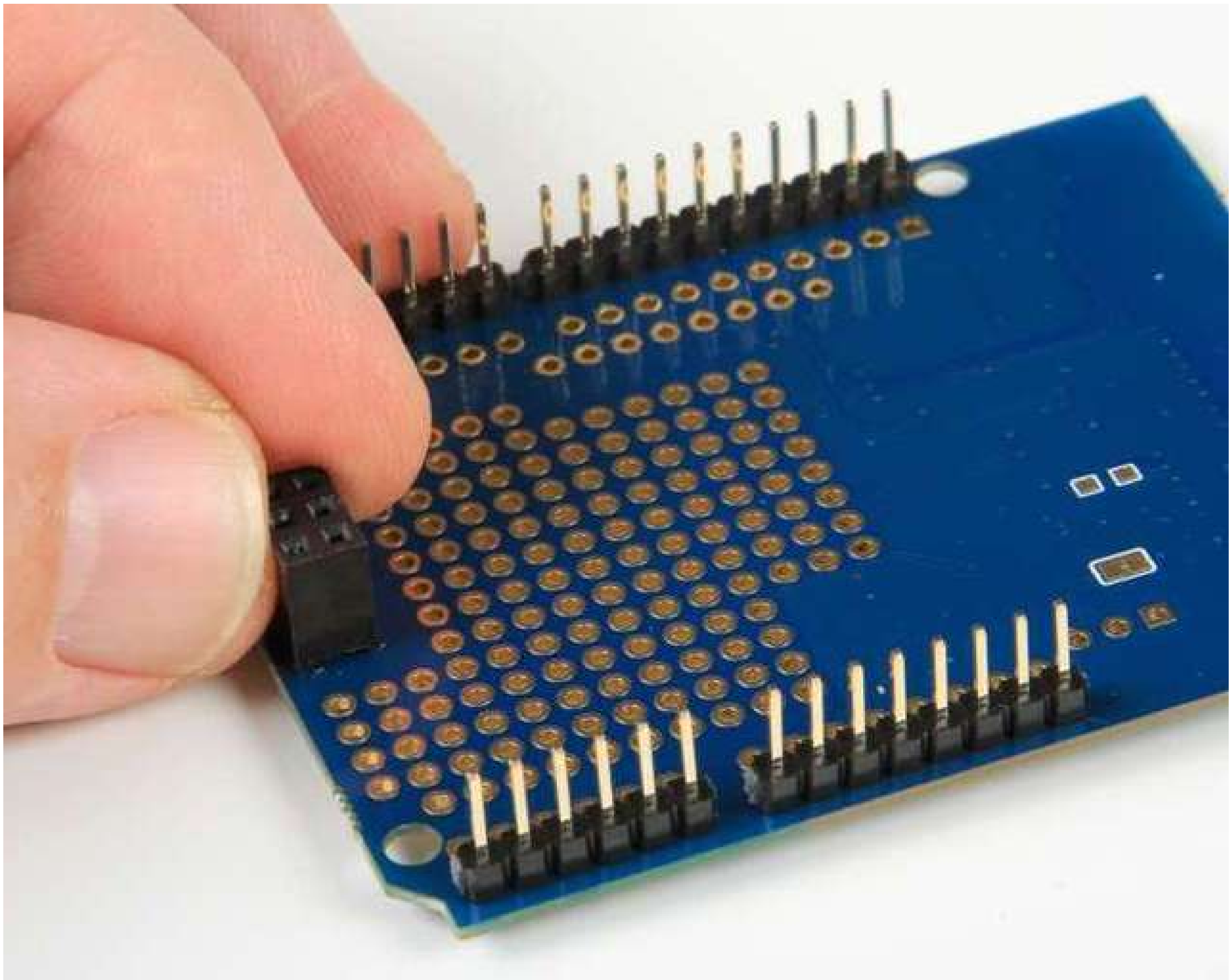
Break off 6, 8, or 10-pin sections and insert them into the header sockets of your Arduino





Place the shield over the pins, and carefully solder each one in place (see our Guide To Excellent Soldering (http://adafru.it/aTk) for instructions and tips on getting the best results).





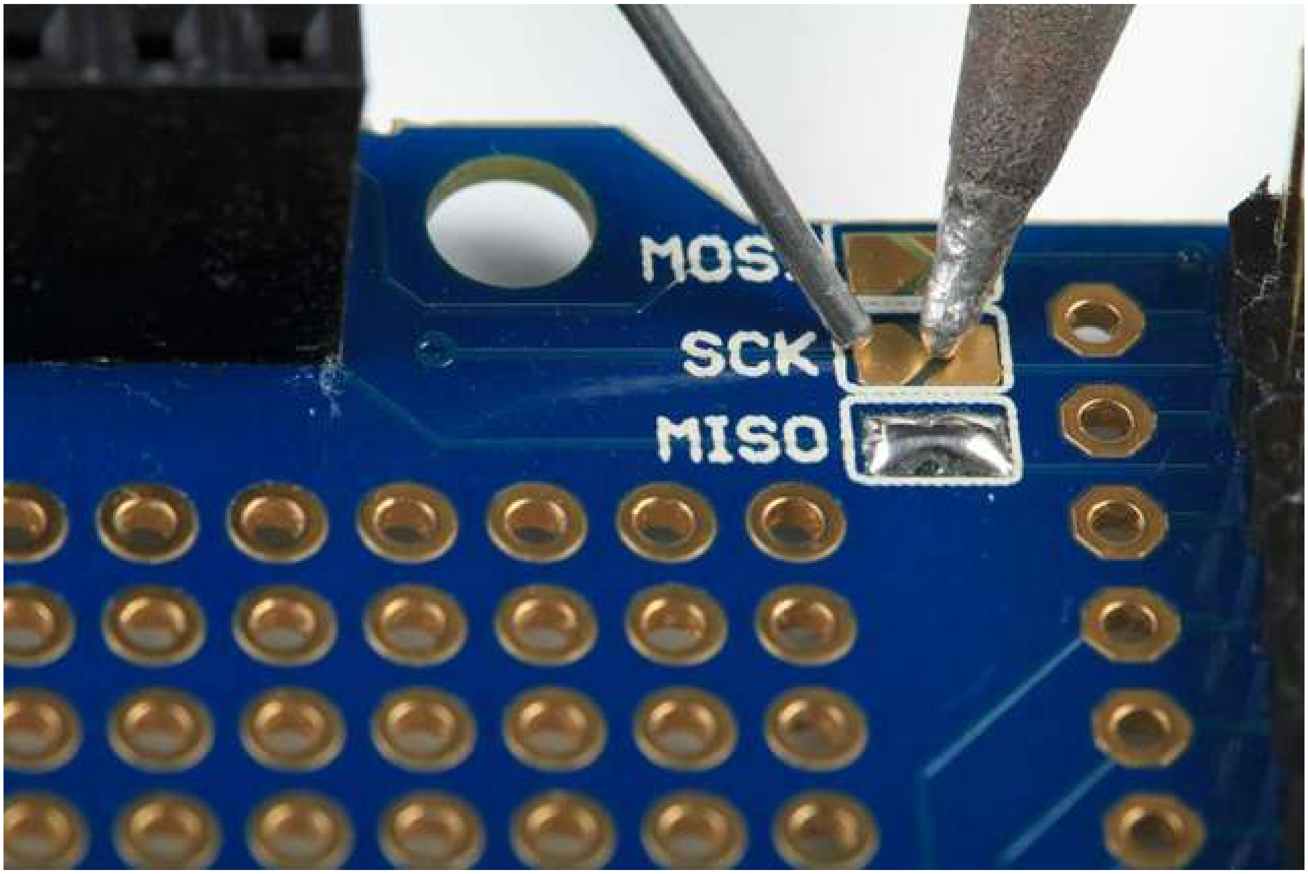
The shield also comes with a 2X3 pin female header socket. This will plug into the 2X3 ICSP pin header on your Arduino, to bring SPI up to the shield. This allows you to use the shield with an Arduino Mega, Leonardo, or Due without having to cut traces or solder jumper wires for SPI.

Set the socket header into the holes on the shield, then flip the shield over and solder the pins. The height of the header block matches the height of the rest of the shield header pins, so the block should be perfectly positioned for soldering!

By default, SPI through the ICSP header is not connected. To enable the ICSP header connections, you'll have to connect three solder jumpers on the bottom of the CC3000 shield.

> Soldering these jumpers is required if you are using the shield with a Mega, Leonardo, or Due, but is not required for use with an UNO!

Simply melt a blob of solder, connecting the pads, on each of the three solder jumpers (keep your solder inside the white boxes - don't let the solder cross between boxes!)



And that's it! Your CC3000 Shield is ready to use. Move on to the next tutorial page to get started!


WiFi Shield
CC3000
MISO
MOSI
WEN
IRQ
CCS

Connections

Pinouts

The CC3000 is (electrically) fairly simple to use. The module requires an SPI connection, including a clock (CLK), data in from a microcontroller (MOSI) and data out to the microcontroller (MISO). It also uses a chip-select line (CS) for SPI to indicate when a data transfer as started

Along with the SPI interface, there is a power-enable type pin called **VBAT_EN** which we use to start the module properly and also an **IRQ** pin, which is the interrupt from the CC3000. The IRQ pin is required to communicate and must be tied to an interrupt-in pin on the Arduino. On the Mega/UNO, we suggest #2 or #3

On the CC3000 shield, we use the following pin connections

- SCK - #13
- MISO #12
- MOSI #11
- CS for CC3000 #10
- VBAT_EN #5
- CS for SD Card #4
- IRQ #3

On the breakout, be aware that the MISO (data out from module) pin does not go 'high impedance' when CS is driven high. Check the shield for how we use a 74AHC125 to manually tri-state this pin when it's shared with an SD card.

Optional Antenna

If you have a shield or breakout with a uFL connector (instead of an on-board ceramic antenna) you can use a uFL to RP-SMA (http://adafru.it/852) or uFL to SMA (http://adafru.it/851) (less common) adapter and then connect to any 2.4 GHz antenna (http://adafru.it/945). This is handy when you want to place the module in a box but have the antenna on the outside, or when you need a signal boost

Please note that when using an external antenna, the module is no longer FCC-compliant, so if you want to sell the product with FCC certification, it must be retested.

Using the CC3000

Make sure your Arduino is powered by a 1 amp or higher rated external power supply when using with the CC3000! Powering an Arduino + CC3000 from a computer/laptop USB port will lead to unstable behavior and lockups because the USB port can't supply enough power!

Use the 1.0 series Arduino IDE like 1.0.6 with the CC3000 library. The 1.5.7 beta and above versions are known to have some issues with the library. If you're using a Due, grab the older 1.5.6-r2 version from: http://arduino.cc/en/Main/OldSoftwareReleases

Download the Library

We will start by downloading the Adafruit CC3000 Library, available from our GitHub repository (http://adafru.it/cFn).

You can download the latest ZIP file by clicking the button below.

Download the latest Adafruit_CC3000 library
http://adafru.it/cHe

Rename the uncompressed folder **Adafruit_CC3000**. Check that the **Adafruit_CC3000** folder contains **Adafruit_CC3000.cpp** and **Adafruit_CC3000.h;** also **ccspi.cpp, ccspi.h,** an **examples** folder, and a **utility** folder

Place the **Adafruit_CC3000** library folder your ***sketchbookfolder*/libraries/** folder. You may need to create the libraries subfolder if its your first library. Restart the IDE. You can figure out your ***sketchbookfolder*** by opening up the Preferences tab in the Arduino IDE.

If you're not familiar with installing Arduino libraries, please visit our tutorial: All About Arduino Libraries (http://adafru.it/aYM)!

Sample Sketches

The Adafruit CC3000 Library contains several example sketches, demonstrating different capabilities of the CC3000 along with some useful programming techniques.

To run the sample sketches, you'll have to edit them to include the SSID and password of your access point.

```
#define WLAN_SSID       "myNetwork"        // cannot be longer than 32 characters!
#define WLAN_PASS       "myPassword"
```

Also, make sure that the right wireless security scheme is selected (unsecured, WEP, WPA, or WPA2)

```
// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or WLAN_SEC_WPA2
#define WLAN_SECURITY   WLAN_SEC_WPA2
```

WEP with HEX Passphrases

If you are using WEP security, and your passphrase is a series of HEX digits, you can't simply enter it as a literal string. Instead you have to define is as an actual binary sequence.
For example, if your passphrase is 8899aabbccdd, you would define it as follows:

```
//  #define WLAN_PASS       "8899aabbccdd"  //don't do it this way!
//do it this way:
const char WLAN_PASS[] = {0x88, 0x99, 0xaa, 0xbb, 0xcc, 0xdd, 0x00};
```

Remember to append 0x00 to the declaration, after the passphrase, as shown in the example!

Be aware the library does not currently support WEP passphrases with 0x00 null characters! See this bug for more details: https://github.com/adafruit/Adafruit_CC3000_Library/issues/97

buildtest

buildtest

The **buildtest** sketch does a full test of core WiFi connectivity:

- Initialization
- SSID Scan
- Access Point connection
- DHCP address assignment
- DNS lookup of www.adafruit.com (http://adafru.it/aK0)
- Ping www.adafruit.com (http://adafru.it/aK0)
- Disconnect

It's a good idea to run this sketch when first setting up the module. It will let you know that everything is working correctly.

Before you run the sketch, edit it to replace the dummy SSID and password with your own:

```
#define WLAN_SSID       "yourNetwork"        // cannot be longer than 32 characters!
#define WLAN_PASS       "yourPassword"
```

If you're using WEP, the password should look like this:

const char WLAN_PASS[] = {0x1A, 0x2B, 0x3C, 0x4D, 0x5E, 0x00};

Since it's a collection of bytes not 'passphrase' style key

Also, make sure that the right wireless security scheme is selected (unsecured, WEP, WPA, or WPA2)

```
// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or WLAN_SEC_WPA2
#define WLAN_SECURITY   WLAN_SEC_WPA2
```

Here's a sample of the Serial Monitor output of buildtest. You should see something similar:

```
Hello, CC3000!

RX Buffer : 131 bytes
TX Buffer : 131 bytes
Free RAM: 1237

Initialising the CC3000 ...
Firmware V. : 1.19
MAC Address : 0x08 0x00 0x28 0x01 0xA8 0x8A
```

```
Started AP/SSID scan


Networks found: 3
================================================
SSID Name    : Extreme
RSSI         : 58
Security Mode: 3

SSID Name    : Express
RSSI         : 59
Security Mode: 3

SSID Name    : fios63
RSSI         : 57
Security Mode: 3

================================================

Deleting old connection profiles

Attempting to connect to fios63
Started AP/SSID scan




Connecting to fios63...Waiting to connect...Connected!
Request DHCP

IP Addr: 192.168.1.23
Netmask: 255.255.255.0
Gateway: 192.168.1.1
DHCPsrv: 192.168.1.1
DNSserv: 192.168.1.1
www.adafruit.com -> 207.58.139.247

Pinging 207.58.139.247...5 replies
Ping successful!


Closing the connection
```

Make sure you can see and recognize all of the access points around, connect to the access point, get a good connection with DHCP, can do a DNS lookup on www.adafruit.com (http://adafru.it/aK0) and ping it successfully. If all this works, then your

hardware is known good!

WebClient

WebClient

The WebClient sketch does a test of the TCP client capability:

- Initialization
- Optional SSID Scan (uncomment code section to enable)
- Access Point connection
- DHCP address assignment
- DNS lookup of www.adafruit.com (http://adafru.it/aK0)
- Optional Ping of www.adafruit.com (http://adafru.it/aK0) (uncomment code section to enable)
- Connect to website and print out webpage contents
- Disconnect

The sketch connects to www.adafruit.com (http://adafru.it/aK0) and opens a special webpage (http://adafru.it/cFo) we have prepared for this example. It reads the contents of the page and prints that out to the Serial Monitor.

Before you run the sketch, edit it to replace the dummy SSID and password with your own:

```
#define WLAN_SSID       "yourNetwork"        // cannot be longer than 32 characters!
#define WLAN_PASS       "yourPassword"
```

Also, make sure that the right wireless security scheme is selected (unsecured, WEP, WPA, or WPA2)

```
// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or WLAN_SEC_WPA2
#define WLAN_SECURITY   WLAN_SEC_WPA2
```

Here's a sample of the Serial Monitor output of WebClient. You should see something similar:

```
Hello, CC3000!

Free RAM: 1157

Initializing...
Started AP/SSID scan





Connecting to fios63...Waiting to connect...Connected!
```

```
Request DHCP

IP Addr: 192.168.1.23
Netmask: 255.255.255.0
Gateway: 192.168.1.1
DHCPsrv: 192.168.1.1
DNSserv: 192.168.1.1
www.adafruit.com -> 207.58.139.247

Connect to 207.58.139.247:80
-------------------------------------
HTTP/1.1 200 OK
Date: Thu, 12 Sep 2013 11:04:02 GMT
Server: Apache
Access-Control-Allow-Origin: http://learn.adafruit.com
Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept, Accept-Encoding, Au
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 1728000
Last-Modified: Thu, 27 Jun 2013 14:13:27 GMT
Accept-Ranges: bytes
Content-Length: 74
Connection: close
Content-Type: text/html

This is a test of the CC3000 module!
If you can read this, its working :)
-------------------------------------


Disconnecting
```

Once you get this working, you can change the webpage you want to access to any kind of webpage on the Internet

ntpTest

ntpTest

The ntpTest sketch does a test of the library's SNTP (Simple Network Time Protocol) client:

- Initialization
- SSID Scan
- Access Point connection
- DHCP address assignment
- SNTP time synchronization
- Extract and print current time and date

The sntp client performs a time synchronization with servers from us.pool.ntp.org and pool.ntp.org. You can also optionally provide it the addresses of one or two of your own time servers. The ntpTest sketch tries time.nist.gov first, before falling back to one of the pool servers.

The client also breaks out the synchronized network time into a structure containing current date and time fields. The sketch formats and prints this information to the Serial Monitor.

> To avoid unnecessary loading of NTP servers, please perform the time synchronization as infrequently as possible. Once per day or longer should be plenty to maintain reasonably accurate time.

Before you run the sketch, edit it to replace the dummy SSID and password with your own:

```
#define WLAN_SSID       "yourNetwork"        // cannot be longer than 32 characters!
#define WLAN_PASS       "yourPassword"
```

Also, make sure that the right wireless security scheme is selected (unsecured, WEP, WPA, or WPA2)

```
// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or WLAN_SEC_WPA2
#define WLAN_SECURITY   WLAN_SEC_WPA2
```

Here's a sample of the Serial Monitor output of ntpTest. You should see something similar:

```
    Hello, CC3000!

Free RAM: 843

Initialising the CC3000 ...
```

```
Firmware V. : 1.19

Deleting old connection profiles

Attempting to connect to fios63
Started AP/SSID scan




Connecting to fios63...Waiting to connect...Connected!
Request DHCP
UpdateNTPTime
Current local time is:
7:18:52.65445
Thursday, September 12, 2013
Day of year: 255


Closing the connection
```

InternetTime

InternetTime

The InternetTime sketch is a simplifies version of the ntpTest sketch. It does not use the library's SNTP client, but directly queries an NTP time server from pool.ntp.org to get the current "UNIX time" (seconds since 1/1/1970, UTC (GMT)).

The sketch then uses the Arduino's internal timer to keep relative time. The clock is re-synchronized roughly once per day. This minimizes NTP server misuse/abuse.

The RTClib library (a separate download, and not used here) contains functions to convert UNIX time to other formats if needed.

To avoid unnecessary loading of NTP servers, please perform the time synchronization as infrequently as possible. Once per day or longer should be plenty to maintain reasonably accurate time.

Before you run the sketch, edit it to replace the dummy SSID and password with your own:

```
#define WLAN_SSID       "yourNetwork"        // cannot be longer than 32 characters!
#define WLAN_PASS       "yourPassword"
```

Also, make sure that the right wireless security scheme is selected (unsecured, WEP, WPA, or WPA2)

```
// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or WLAN_SEC_WPA2
#define WLAN_SECURITY   WLAN_SEC_WPA2
```

Here's a sample of the Serial Monitor output of InternetTime. You should see something similar:

```
Hello, CC3000!

RX Buffer : 131 bytes
TX Buffer : 131 bytes

Initialising the CC3000 ...
Firmware V. : 1.19
MAC Address : 0x08 0x00 0x28 0x01 0xA8 0x8A
```

```
Deleting old connection profiles

Attempting to connect to fios63
Started AP/SSID scan




Connecting to fios63...Waiting to connect...Connected!
Request DHCP

IP Addr: 192.168.1.23
Netmask: 255.255.255.0
Gateway: 192.168.1.1
DHCPsrv: 192.168.1.1
DNSserv: 192.168.1.1
Locating time server...
Attempting connection...
Connect to 62.116.162.126:123
connected!
Issuing request...
Awaiting response...OK
Current UNIX time: 1378987424 (seconds since 1/1/1970 UTC)
Current UNIX time: 1378987439 (seconds since 1/1/1970 UTC)
Current UNIX time: 1378987454 (seconds since 1/1/1970 UTC)
Current UNIX time: 1378987469 (seconds since 1/1/1970 UTC)
... etc ...
```

GeoLocation

GeoLocation

This example sketch queries the freegeoip.net service to get the local approximate geographic location based on IP address.

Combined with code in the ntpTest or InternetTime sketches, this can give absolute position and time, extremely useful for seasonal calculations like sun position, insolation, day length, etc. One could always add a GPS module or just plug in values from your GPS or phone, but for applications where extreme accuracy isn't required, this has the luxury of coming 'free' with the CC3000 already in use.

> Positional accuracy depends on the freegeoip.net database, in turn based on data collected by maxmind.com. No guarantees this will work for every location. This software is provided as-is.

Position should be polled only once, at startup, or very infrequently if making a mobile network-hopping thing, so as not to overwhelm the kindly-provided free geolocation service.

Before you run the sketch, edit it to replace the dummy SSID and password with your own:

```
#define WLAN_SSID       "yourNetwork"   // cannot be longer than 32 characters!
#define WLAN_PASS       "yourPassword"
```

Also, make sure that the right wireless security scheme is selected (unsecured, WEP, WPA, or WPA2)

```
// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or WLAN_SEC_WPA2
#define WLAN_SECURITY   WLAN_SEC_WPA2
```

Here's a sample of the Serial Monitor output of GeoLocation. You should see something similar:

```
Hello, CC3000!
Free RAM: 837
Initializing...OK.
Connecting to network...Started AP/SSID scan



Connecting to Turlingdrome...Waiting to connect...connected!
```

```
Requesting address from DHCP server...OK

IP Addr: 192.168.0.4
Netmask: 255.255.255.0
Gateway: 192.168.0.1
DHCPsrv: 192.168.0.1
DNSserv: 192.168.0.1

Getting server IP address...192.151.154.154
Connecting to geo server...
Connect to 192.151.154.154:80
connected.
Requesting data...
Reading response...OK

Disconnecting

RESULTS:
  Country: United States
  Region: California
  City: Richmond
  Longitude: -122.35
  Latitude: 37.94
```

SmartConfig

SmartConfig is the special functionality in the CC3000 that allows setting the SSID and password settings without having to type or re-program the module. Any iOS/Android device can be used to set the configuration - solving the annoying deployment problem of how to set the connection details for a new device.

SmartConfigCreate and SmartConfigReconnect

These two SmartConfig sketches should be used together to demonstrate how the SmartConfig app can be used on your smartphone to pass connection details to your CC3000.

SmartConfigCreate

This sketch will initialise the CC3000, erasing any previous connection details stored on the device. It will then enter SmartConfig mode with a 60 second timeout where it waits for configuration data to arrive from the SmartPhone.

If a connection was successfully established, the connection details will be stored in the non-volatile memory of the CC3000, and the module will be configured to automatically reconnect to this network on startup (meaning you don't need to run the SmartConfig app unless your AP details change or you erase the stored connection details on the module).

There's no need to edit the sketch to add your SSID and password - the SmartConfig app does that for you!

SmartConfigReconnect

This sketch shows how to use the CC3000 in 'reconnect' mode, and avoid erasing all stored connection profiles, which is unfortunately necessary with other sketches where manual config data is provided.

- Initializates the CC3000 with a special SmartConfig flag so it doesn't erase the profile data
- Access Point connection (based on saved AP details)
- DHCP address assignment
- Disconnect

SmartConfig is still in beta testing! It might not work on all networks!

Using the SmartConfigCreate Sketch

Step One: Install the SmartConfig App

Before you can use SmartConfig to provide your AP connection details, you need to install the SmartConfig app:

- For **iOS devices** simply search for the TI WiFi SmartConfig app (http://adafru.it/cQ3) from the app store.

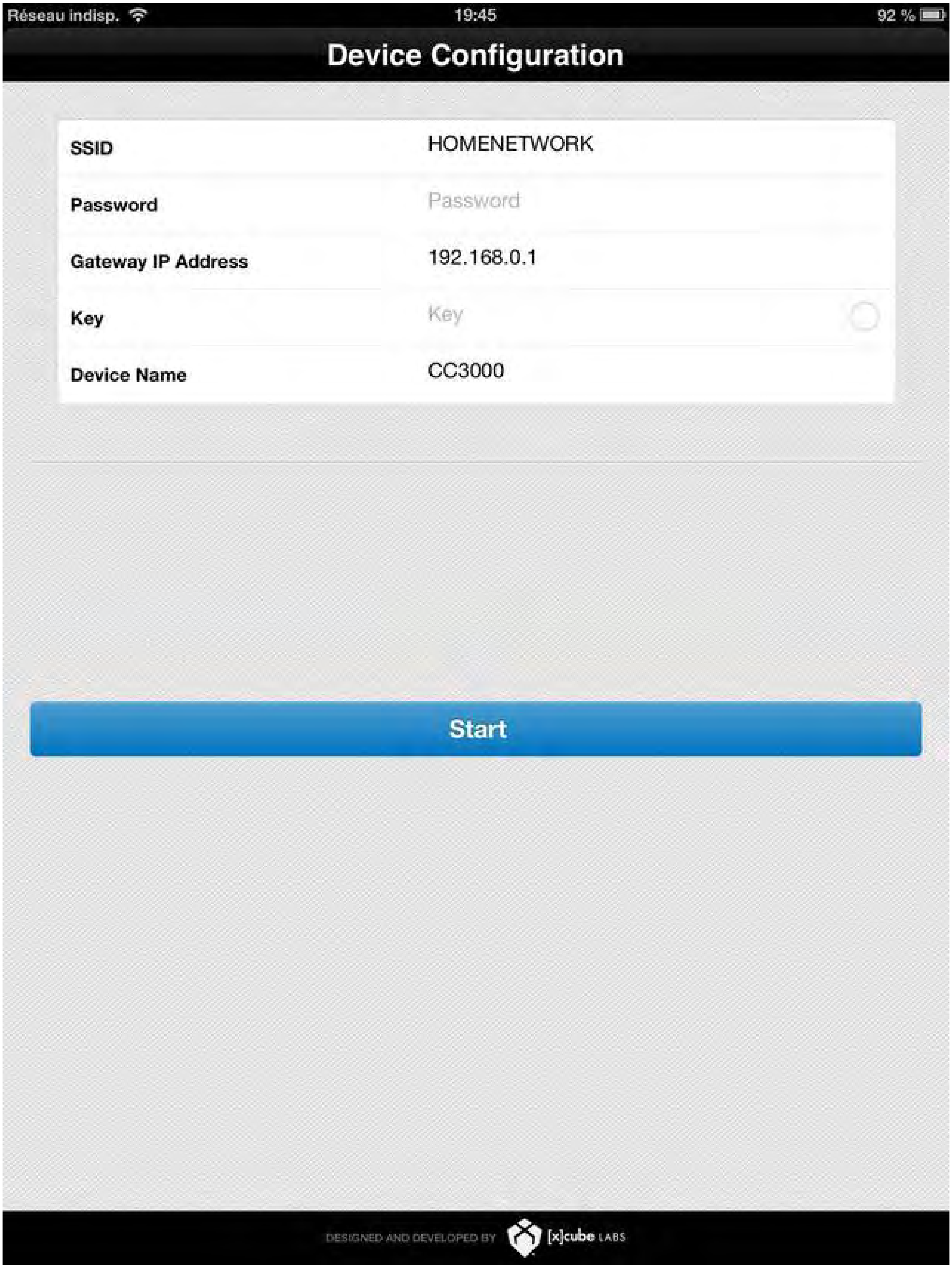
- For **Android devices**, you can download the app directly from TI's CC3000 Wiki (http://adafru.it/cQ4)

Step Two: Configure the SmartConfig App on your Phone

Once you've installed the SmartConfig app, you need to connect to the AP that the CC3000 will be using (HOMENETWORK in the images below), and then load the app.

You should see a screen similar to the following, with the AP's **SSID**, **Gateway IP Address** and **Device Name** fields already populated:

This tutorial will use an iPad to provide the SmartConfig details, but the process is basically the same on Android.



Add the password for your Access Point, but **don't click the START button yet!**

Don't change the Key or DeviceName fields!

Step Three: Open and Run 'SmartConfigCreate'

- In the **File > Examples > Adafruit_CC3000** menu select the **SmartConfigCreate** sketch.
- Run the sketch and open the Serial Monitor via **Tools > Serial Monitor**.
- You should see something similar to the following text:

```
Hello, CC3000!

RX Buffer : 131 bytes
TX Buffer : 131 bytes
Free RAM: 595

Initialising the CC3000 ...
Firmware V. : 1.24
MAC Address : 0x08 0x00 0x28 0x01 0xA8 0x1F
Waiting for a SmartConfig connection (~60s) ...
```

Step Four: Start the SmartConfig app on your Phone

Before the Android sketch times out, click to '**Start**' button in your TI app, and watch the serial monitor window of your sketch. After about 30 seconds you should see something similar to the following:

```
Got smart config data
Saved connection details and connected to AP!
Request DHCP

IP Addr: 192.168.0.103
Netmask: 255.255.255.0
Gateway: 192.168.0.1
DHCPsrv: 192.168.0.1
DNSserv: 192.168.0.1

To use these connection details be sure to use
'.begin(false, true)' with your Adafruit_CC3000
code instead of the default '.begin()' values!

Closing the connection
```

Step Five: Stop the SmartConfig App on the Phone

If everything worked out and you successfully connected to your AP, the connection details were also stored in non-volatile memory on the CC3000 module. You can now use the **SmartConfigReconnect** sketch to test the connection details, specifically paying attention to the extra flags in the **Adafruit_CC3000.begin()** function compared to other sketches..

Did the sketch timeout before connecting?

- Be sure to click '**Stat**' in the SmartPhone app as soon as the '**Waiting for SmartConfig connection (~60s) ...**' message pops up. The SmartConfig device will timeout after 60 seconds, so you may need to run the sketch again and be a bit quicker with your fingers.
- Make sure that the iPad or SmartPhone is connected to the same AP that you want the CC3000 to connect to!
- Check your password in case there is a typo

Using the SmartConfigReconnect Sketch

The SmartConfigReconnect sketch shows how to use (and retain) the connection details that were written to the device in the example above.

The key to using and maintaining the connection details is to pass an optional flag to the Adafruit_CC3000 classes **.begin()** function to tell the driver **NOT** to delete existing connections, and to stay in auto connect mode:

```
/* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
/* !!! Note the additional arguments in .begin that tell the   !!! */
/* !!! app NOT to deleted previously stored connection details !!! */
/* !!! and reconnected using the connection details in memory! !!! */
/* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
if (!cc3000.begin(false, true))
{
  Serial.println(F("Unable to re-connect!? Did you run the SmartConfigCreate"));
  Serial.println(F("sketch to store your connection details?"));
  while(1);
}
```

The first flag should always be **false**, and is used to indicate that we are going to perform a firmware update. The second flag should always be **true** when using SmartConfig data, and puts the CC3000 in an auto-reconnect mode and maintains existing connection details in non-volatile memory.

If you were able to successfully connect using the **SmartConfigCreate** sketch, SmartConfigReconnect should give you something similar to the following output:

```
Hello, CC3000!

Trying to reconnect using SmartConfig values ...
Reconnected!

Requesting DHCP

IP Addr: 192.168.0.103
Netmask: 255.255.255.0
Gateway: 192.168.0.1
DHCPsrv: 192.168.0.1
DNSserv: 192.168.0.1

Closing the connection
```

Any time you don't provide the extra flags to the .begin method, all connection details will be erased and auto-reconnect mode will be disabled! Unfortunately, this is necessary when providing manual connection details, so be careful using non SmartConfig* sketches if you don't want to lose your connection details.

SendTweet

Unfortunately Twitter changed their API and require SSL connections which the CC3000 does not support. This page is only for reference as the SendTweet example does not work anymore and is not included in the latest library.

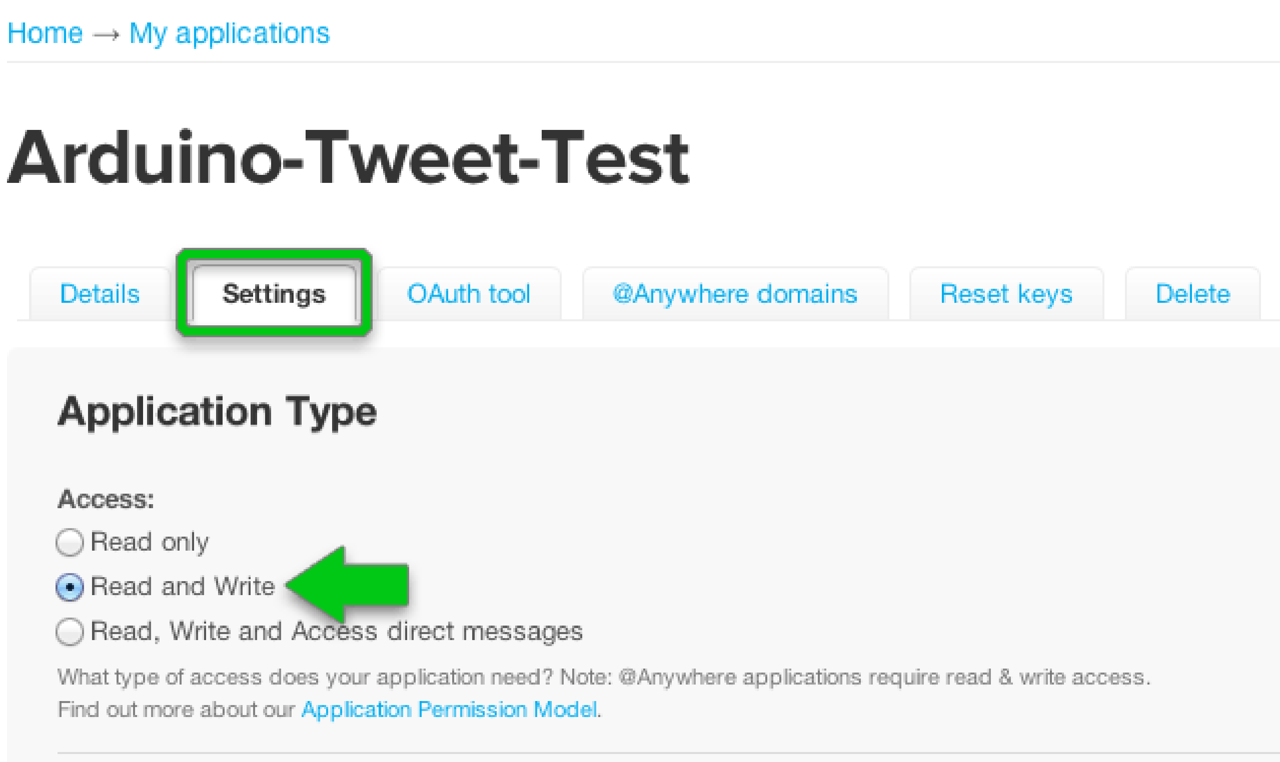
SendTweet

This example sketch sends "tweets" (Twitter messages) from an Arduino with CC3000 WiFi. Usually this requires extra proxy software running on another computer, but this sketch operates directly from the Arduino.

This is a barebones example that issues a single fixed message, but it's easily adapted to send different information such as a periodic sensor reading.

In addition to the WiFi setup explained below, it's necessary to set up a Twitter developer account and complete an application form before this can be used. **That procedure is explained on the Twitter Setup page of the Internet of Things Printer tutorial** (http://adafru.it/cHs)**.**

One additional configuration step is required on the Twitter developer site: from your applications "Settings" tab, set access to "Read and Write." This is necessary so our sketch can <u>send</u> tweets; the printer sketch only <u>reads</u> tweets.



Before you run the sketch, edit it to replace the dummy SSID and password with your own:

```
#define WLAN_SSID     "yourNetwork"  // cannot be longer than 32 characters!
#define WLAN_PASS     "yourPassword"
```

Also, make sure that the right wireless security scheme is selected (unsecured, WEP, WPA, or WPA2)

```
// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or WLAN_SEC_WPA2
#define WLAN_SECURITY   WLAN_SEC_WPA2
```

Here's a sample of the Serial Monitor output of SendTweet. You should see something similar:

```
Hello! Initializing CC3000...Firmware V. : 1.19
OK
Deleting old connection profiles...OK
Connecting to network...Started AP/SSID scan


Connecting to Turlingdrome...Waiting to connect...OK
Requesting address from DHCP server...OK
IP Addr: 192.168.0.4
Netmask: 255.255.255.0
Gateway: 192.168.0.1
DHCPsrv: 192.168.0.1
DNSserv: 192.168.0.1
Locating time server...found
Connecting to time server...
Connect to 155.101.3.115:123
connected!
Issuing request...OK
Awaiting response...success!
Locating Twitter server...OK
Connecting to server...

Connect to 199.59.150.9:80
OK
Issuing HTTP request...OK
Awaiting response...success!
Waiting ~1 hour...
```

Firmware Upgrades

The CC3000 is a complex chip that has its own firmware published by Texas Instruments. You might find it necessary to upgrade the firmware, for example to use the latest version with recent bug fixes. Luckily it's easy to upgrade (and downgrade) firmware on the chip using a few included examples in the Adafruit CC3000 Arduino library.

Currently there are two firmware versions provided in the library:

- **Version 1.12** - This is a minor bug fix release with release notes here (http://adafru.it/dVA).
    - The 1.12 and earlier firmware versions are known to have problems with heavy load and certain network conditions which can cause the CC3000 to lock up. Consider upgrading to the more recent 1.13 release if you run into stability issues.
- **Version 1.13** - This is another bug fix release with release notes here (http://adafru.it/dVA).
    - The 1.13 release has a fix for internal CC3000 issues which cause lock ups under heavy usage and certain network conditions. However note that the 1.13 release also appears to have bugs with UDP traffic (http://adafru.it/dVB). Most internet traffic uses TCP instead of UDP so you likely won't run into problems and should consider upgrading to the latest 1.13 firmware.

Do not power your Arduino & CC3000 from a computer/laptop USB port during a firmware upgrade! You MUST use an external power supply with at least 1 amp of current capacity. This is to ensure the Arduino and CC3000 have enough power to operate during the firmware upgrade.

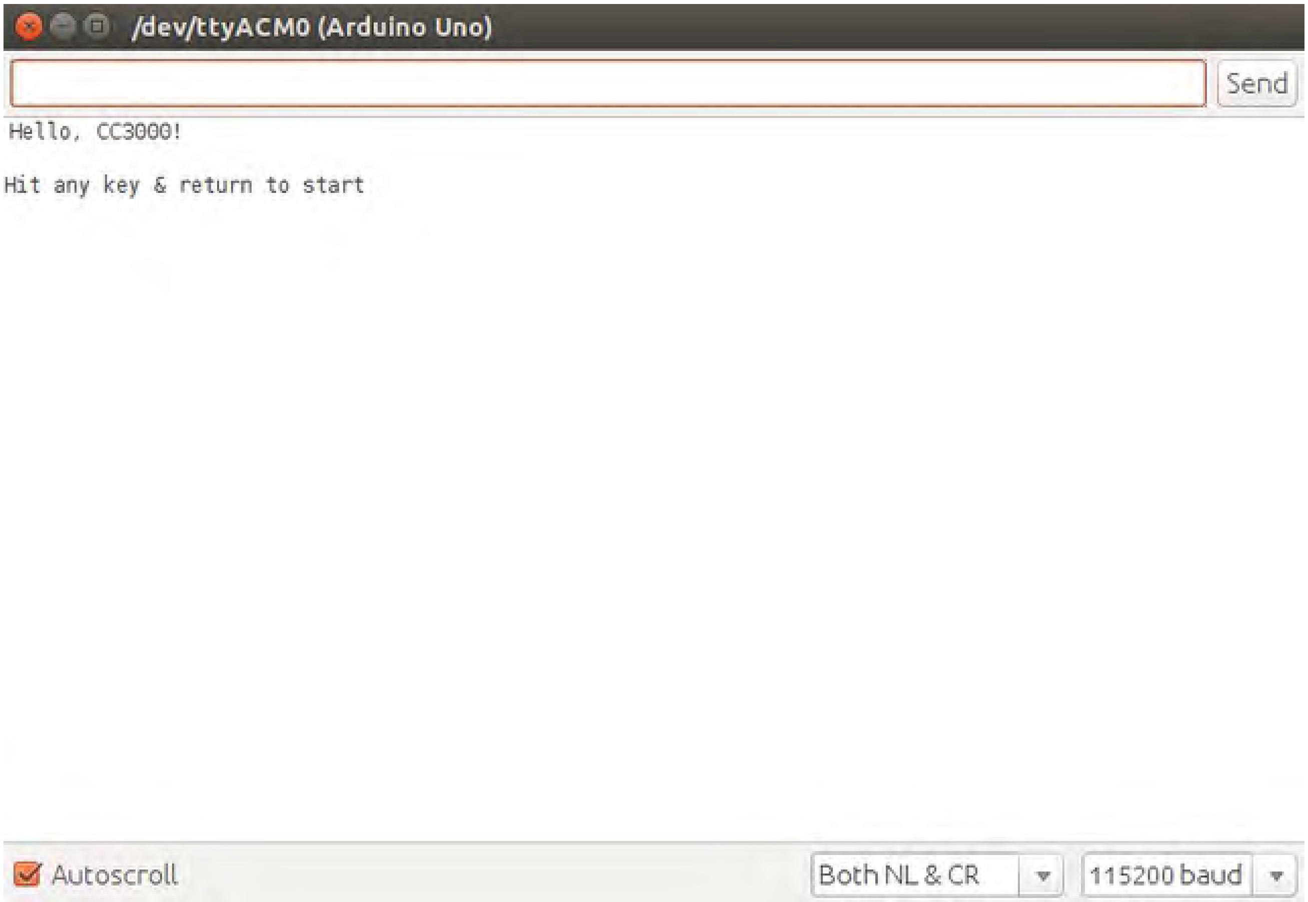
To upgrade to a specific version first make sure you have the most recent Adafruit CC3000 Arduino library (http://adafru.it/cFn) installed. If you installed the CC3000 library some time ago make sure to download and install it again as fixes and new firmware versions are added periodically.

Next make sure your CC3000 is wired to your Arduino and can successfully run CC3000 sketches like buildtest. If there's a problem communicating with the CC3000 you want to find out before your start the firmware upgrade. **Also be sure you're using a good quality 1 amp or more external power supply and not a computer/laptop USB port to power the Arduino & CC3000!**

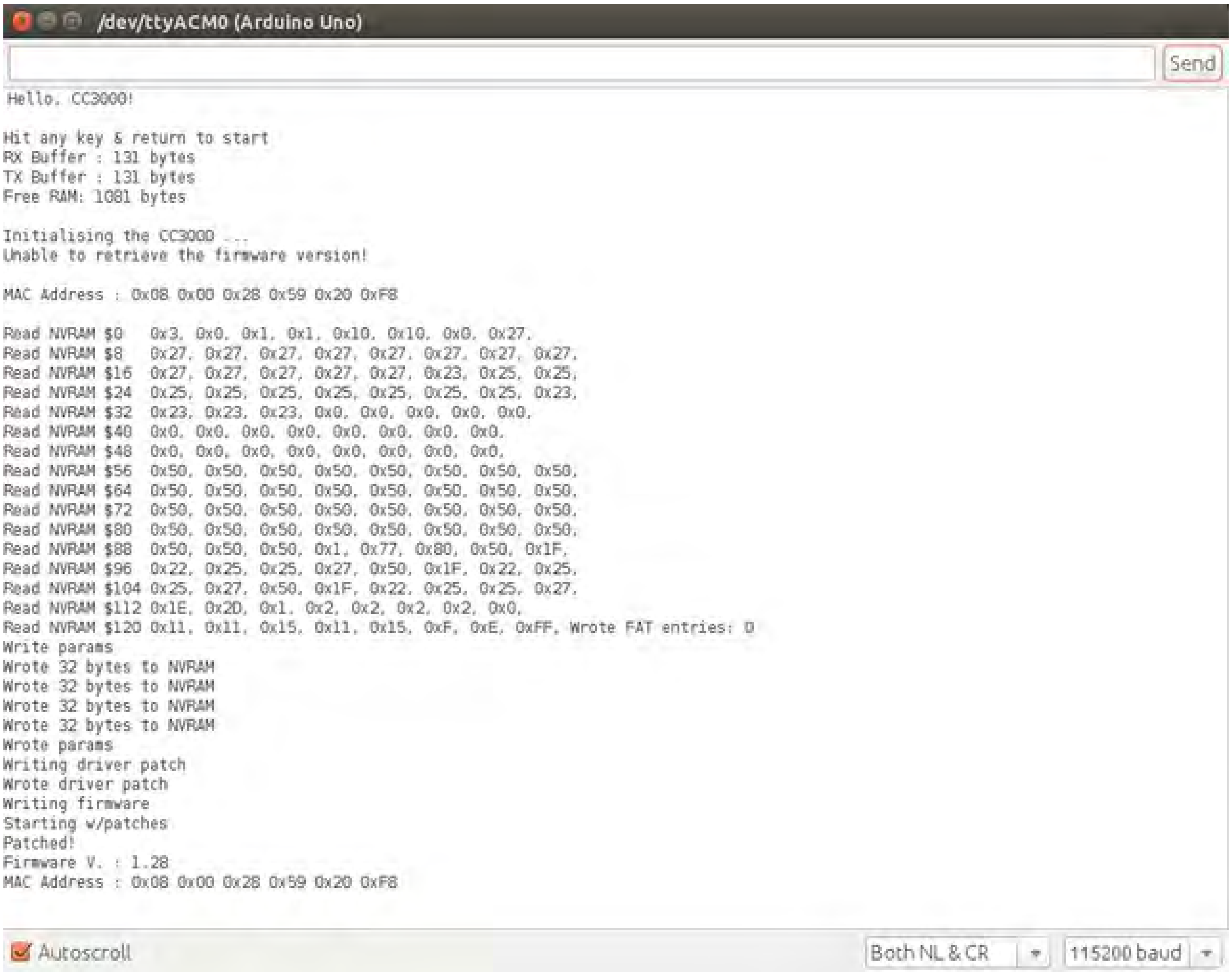
Now in the Arduino IDE load one of the **driverpatch_X_XX** CC3000 examples, where X_XX is the version like **driverpatch_1_13** for version 1.13. Adjust any of the pins to communicate with your Arduino just like you would to run buildtest. Compile and load the

sketch on your Arduino.

Open the serial monitor at 115200 baud and you should see a message such as a the following (try pressing the Arduino's reset button if you see no message):



Enter some text and press send to start the firmware upgrade process. The upgrade will happen fairly quickly and should be done in a minute or two. Here's what you should see during an upgrade:



After the upgrade finishes load the buildtest example and run it again. Confirm that everything works as expected with the connection to your wireless network and ping of adafruit.com. Congratulations you've upgraded the firmware on the CC3000!

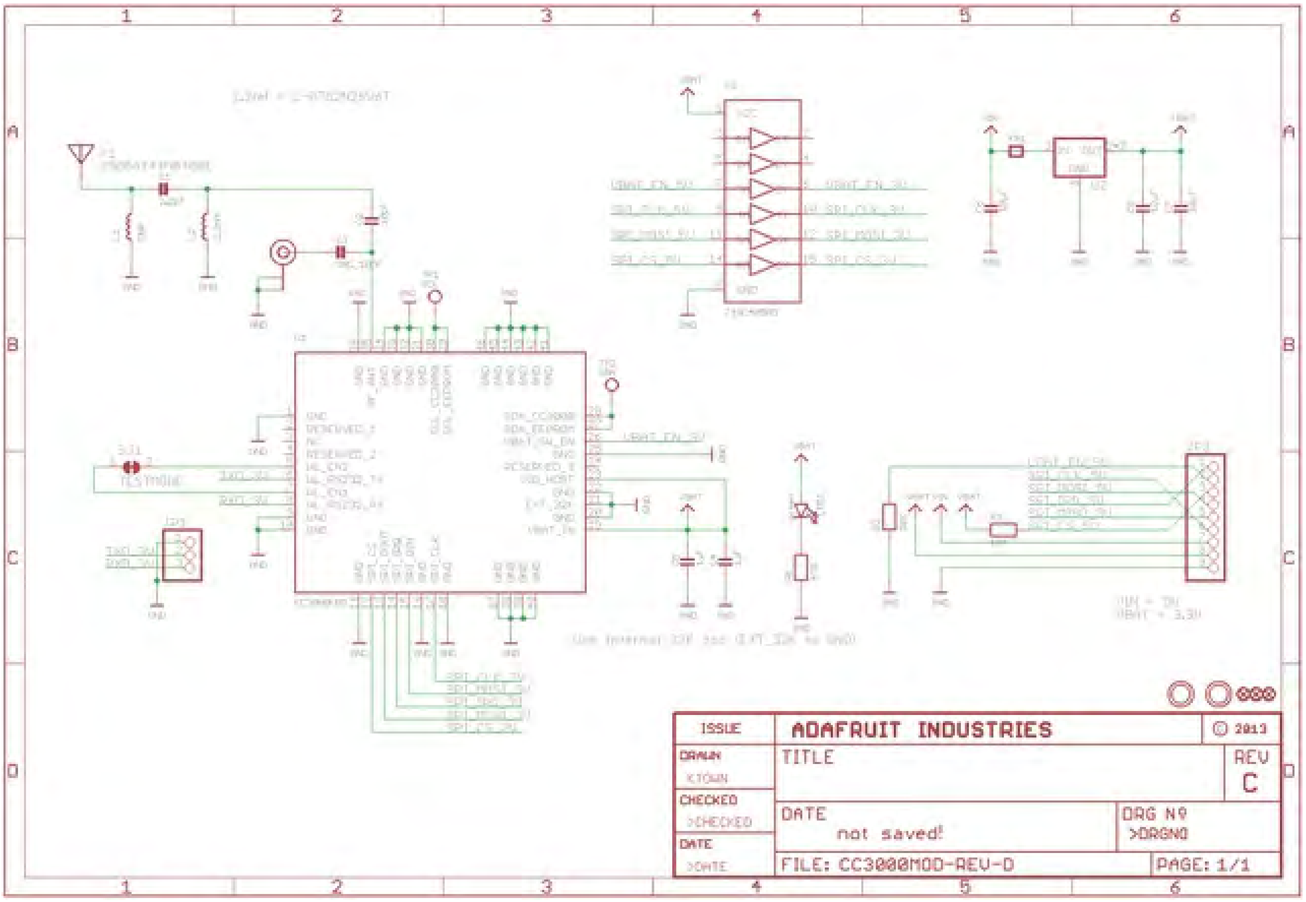
Note: The version number displayed during the firmware upgrade and from sketches like buildtest includes the major, minor, and patch numbers so it doesn't exactly match the version number from TI. Here's a handy conversion between reported version number and CC3000 firmware version:

- Buildtest reports version 1.28 = CC3000 firmware v1.13
- Buildtest reports version 1.26 = CC3000 firmware v1.12
- Buildtest reports version 1.24 = CC3000 firmware v1.11

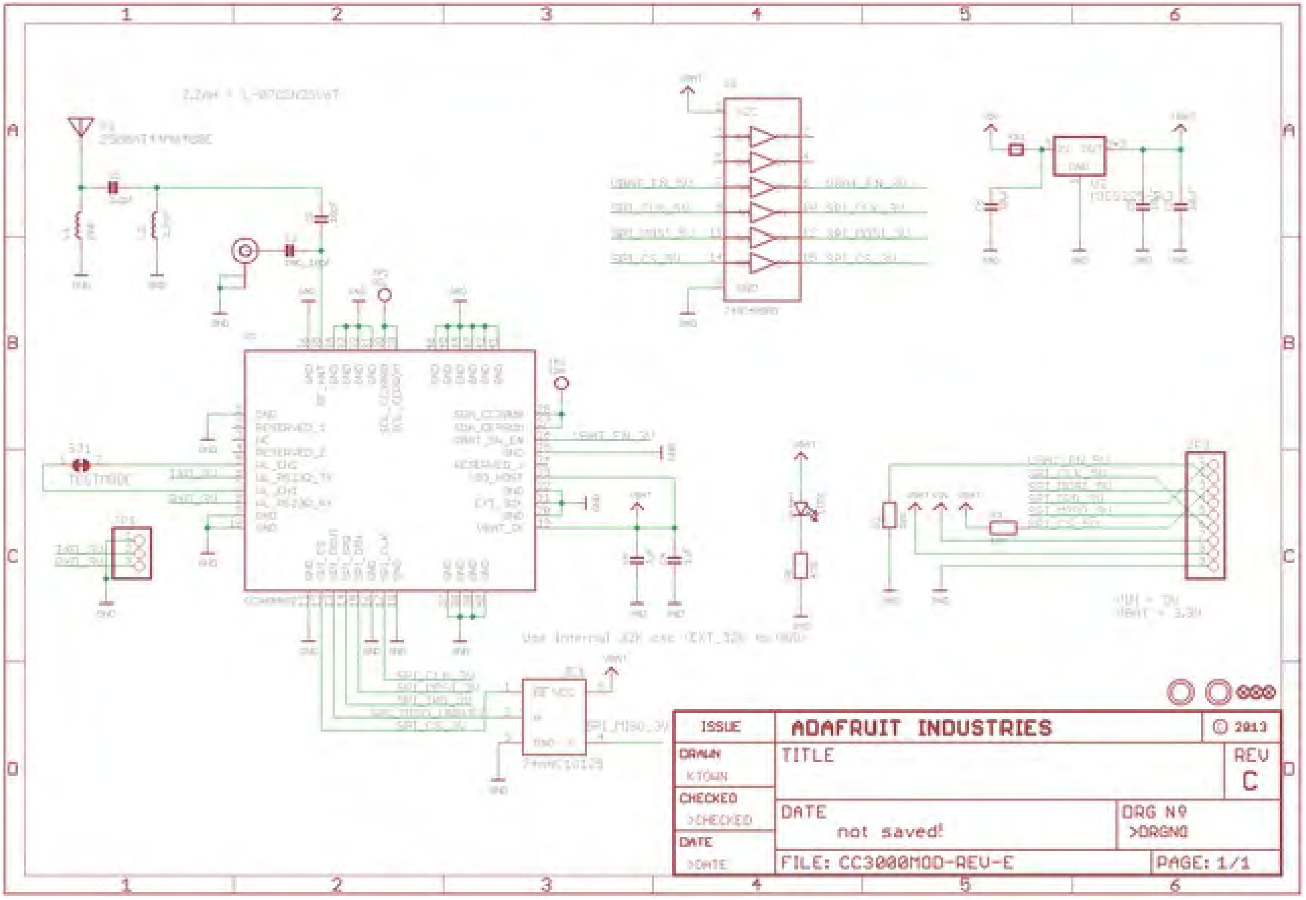
Downloads

For more information on the CC3000, check out TI's product page (http://adafru.it/cHf) and wiki microsite (http://adafru.it/cHg), its got tons and tons of information about their WiFi module

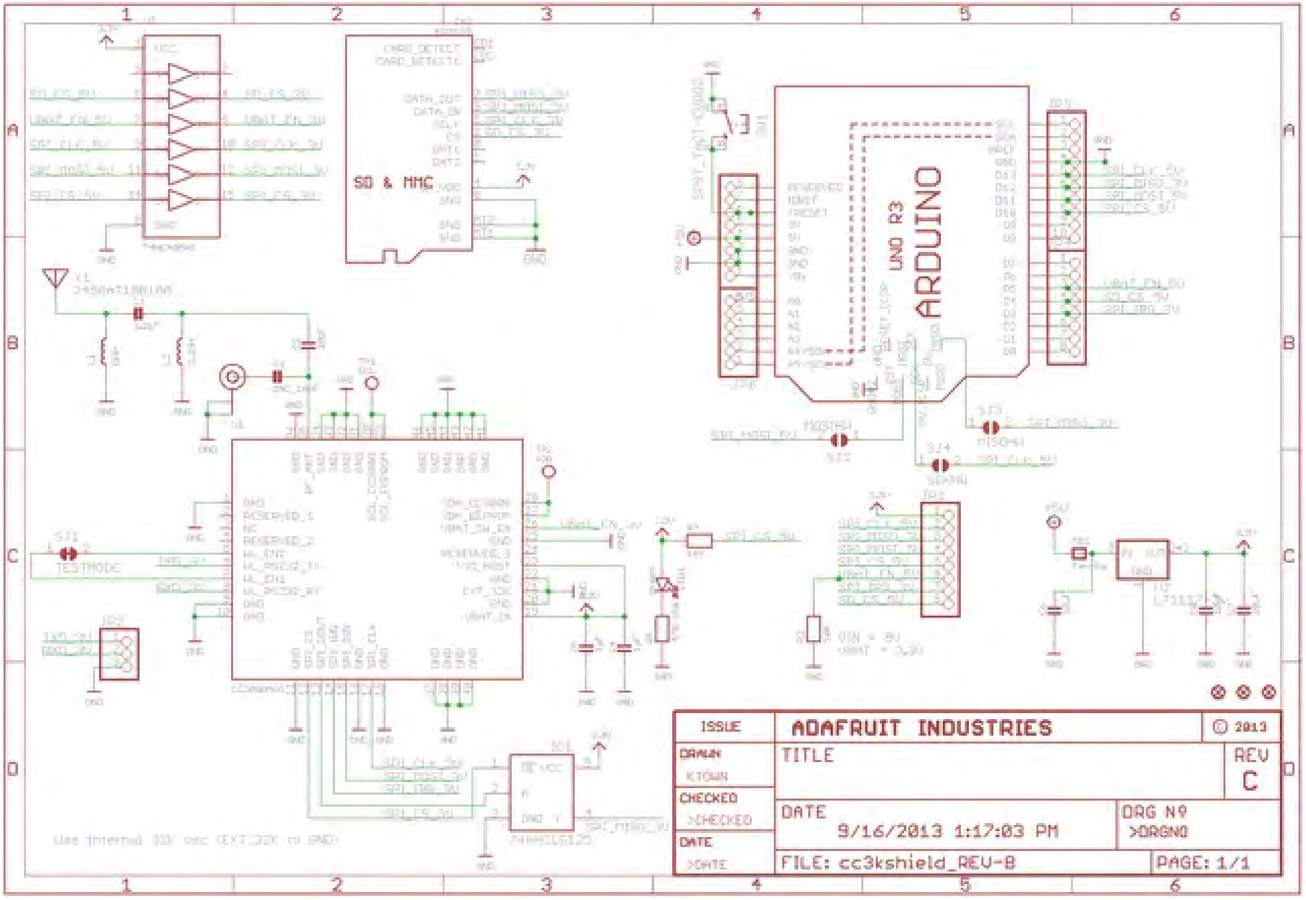
Schematic for the CC3000 breakout board v1.0 (no buffer on the MISO pin)



Schematic for v1.1 with a buffer on MISO



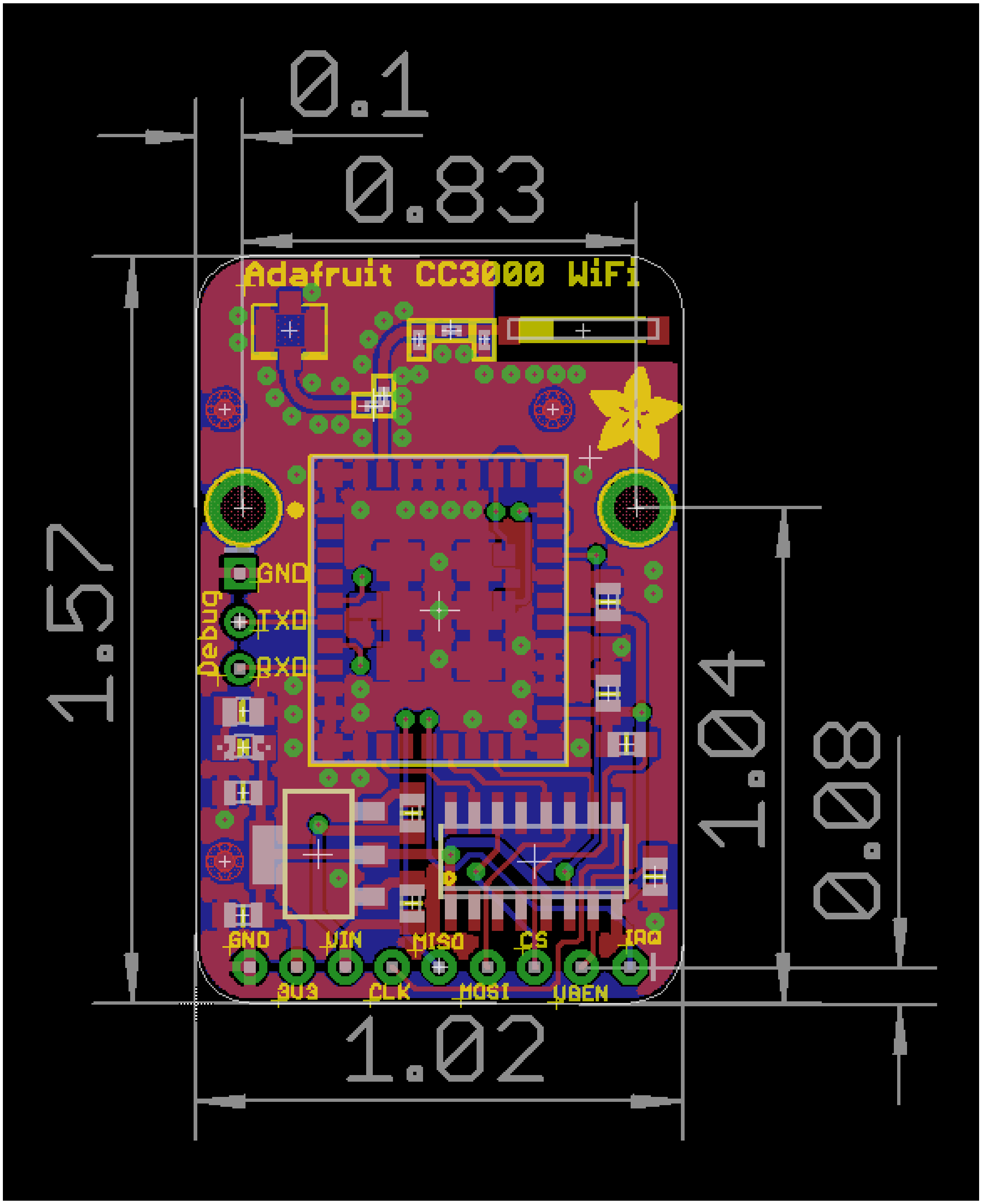
Schematic for the CC3000 shield



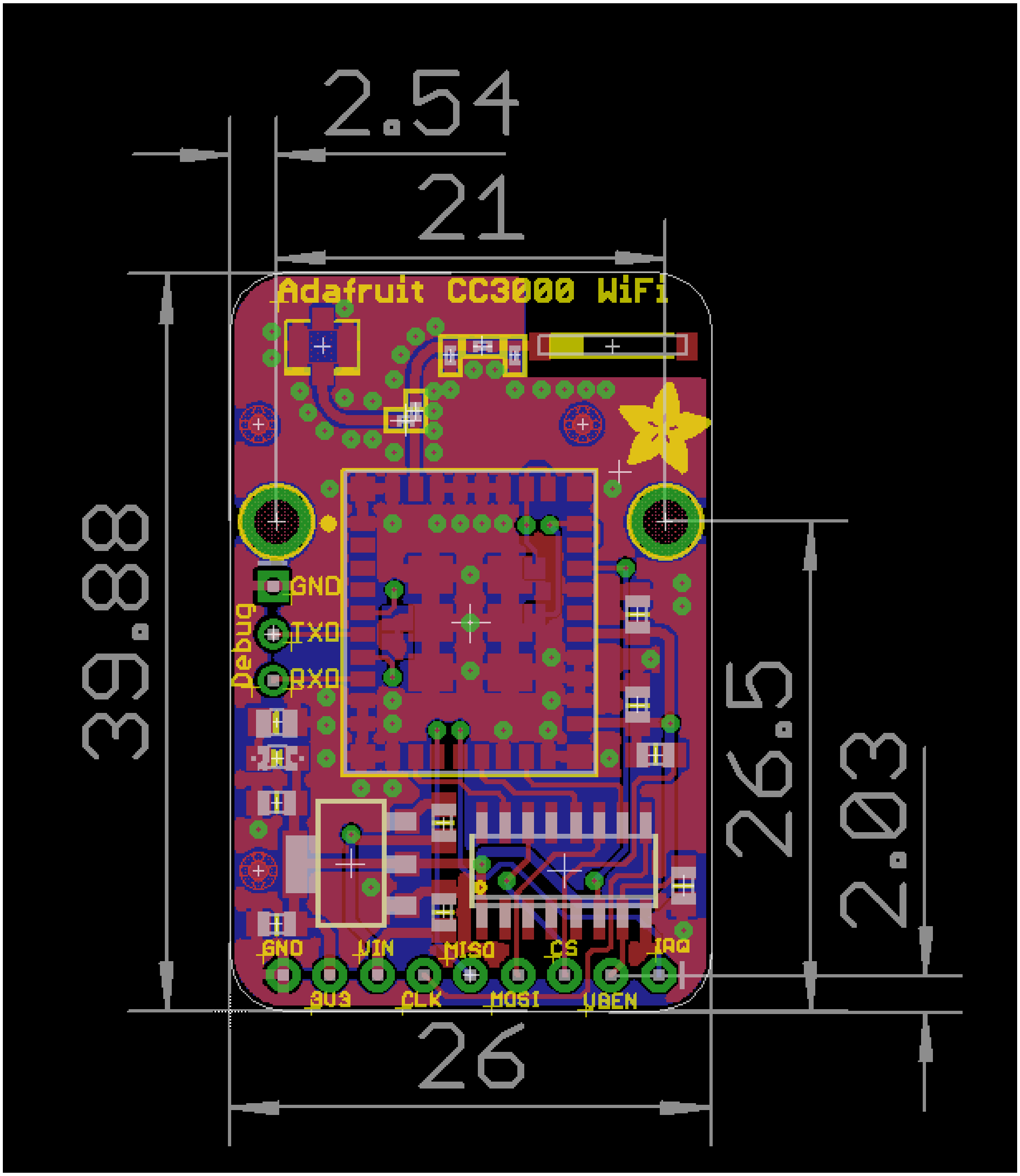
Dimensional diagram for the CC3000 breakout

The dimentions are the same for v1 and v1.1

Inches:


0.1
0.83
1.57
1.04
0.08
1.02
Adafruit CC3000 WiFi
GND
TXD
RXD
Debug
GND
VIN
MISO
CS
IRQ
3V3
CLK
MOSI
VBEN

mm:


2.54
21
Adafruit CC3000 WiFi
39.88
GND
TXD
RXD
Debug
26.5
2.03
GND
VIN
MISO
CS
IRQ
3V3
CLK
MOSI
VBEN
26

FAQ

I'm using WEP - how do I configure my HEX passphrase?

If your passphrase is a series of HEX digits, you can't simply enter it as a literal string. Instead you have to define is as an actual binary sequence.

For example, if your passphrase is 8899aabbccdd, you would define it as follows:

```
//  #define WLAN_PASS       "8899aabbccdd"  //don't do it this way!
//do it this way:
const char WLAN_PASS = {0x88, 0x99, 0xaa, 0xbb, 0xcc, 0xdd, 0x00};
```

I'm using WEP and I tried that but it still doesn't work

Make sure you have WLAN_SECURITY defined as WEP:

```
#define WLAN_SECURITY   WLAN_SEC_WEP
```

What is the gain of the ceramic antenna? How does it compare to the external antennas?

We use the Johannson 2500AT44M0400 (http://adafru.it/cVL) which has 0.5 dBi gain. Compare this to the external antennas with 2 dBi and 5dBi. Since antenna 'range' is not linear with the gain and antenna range has *a lot to do* with what else is transmitting or receiving, physical barriers, noise, etc. We can roughly say that the ceramic antenna has half the range of the 2dBi antenna, and the 5dBi antenna has double the range of the 2dBi antenna *(roughly!)*

There is no way to know the actual range you will get unless you experiment with your setup since there is so many variables, but the ceramic antenna gets about the same range we expect with an every day cellphone

(http://adafru.it/cVL)

How can I use the CC3000 with a static IP?

WiFi device IPs are dynamic 99% of the time, but it is possible to assign a static IP if your router permits it. Check out this forum post for how to go about it (http://adafru.it/clc)

I'm having difficulty seeing/connecting to my network...

If you have an '802.11n only' router please configure it to add 'b or g' support. The CC3000 is 802.11b or g only, it does not do 'n'!

I'm not in the USA and my CC3000 can't see my router

The CC3000 only operates on channels 1 through 11. Outside of the United States, you may be able to configure your router to a channel number outside of that range. Make sure your router is configured for a channel in the range 1 through 11.

My CC3000 sketch locks up when initializing the CC3000...

A lockup during initialization is a very common issue when there isn't enough power to supply both the Arduino and CC3000 board. Make sure you're powering the Arduino from a 1 amp or higher rated external power supply. **Don't try to power the Arduino from a computer/laptop USB port because those ports typically can't provide enough power and will cause lockups!**

If you're using an Arduino UNO you can plug in a 7-12 volt power supply to the barrel jack on the board. A supply like [this 9V 1amp wall wart (http://adafru.it/63)](http://adafru.it/63) is perfect for the Arduino UNO and CC3000.

Also if your Arduino supports changing the voltage of the digital I/O pins (like some 3rd party Arduino clones), make sure the voltage is set to 5 volts and not 3.3 volts.

My CC3000 sketch locks up after running for a while or under heavy load...

Unfortunately there is a well known internal issue with the CC3000 which can cause lock ups and instability over time or under heavy load. [This thread on the Spark Core forums (http://adafru.it/dVC)](http://adafru.it/dVC) dives deeply into the issue and investigation with Texas Instruments. Ultimately the latest firmware version 1.13 was released with a potential fix for the stability problems. If you'd like to upgrade to firmware version 1.13 see the [page on firmware upgrades in this guide (http://adafru.it/dVD)](http://adafru.it/dVD).

How do I set a static IP address?

By default the CC3000 is configured to get an IP address automatically from your router using DHCP. In most cases this works well, however if you run into trouble getting an IP address or DNS server you should consider setting a static IP address and DNS server.

To set a static IP address make sure you have the latest version of the CC3000 library and load the buildtest example. Scroll down to the commented section of code in the setup function which discusses setting a static IP address:

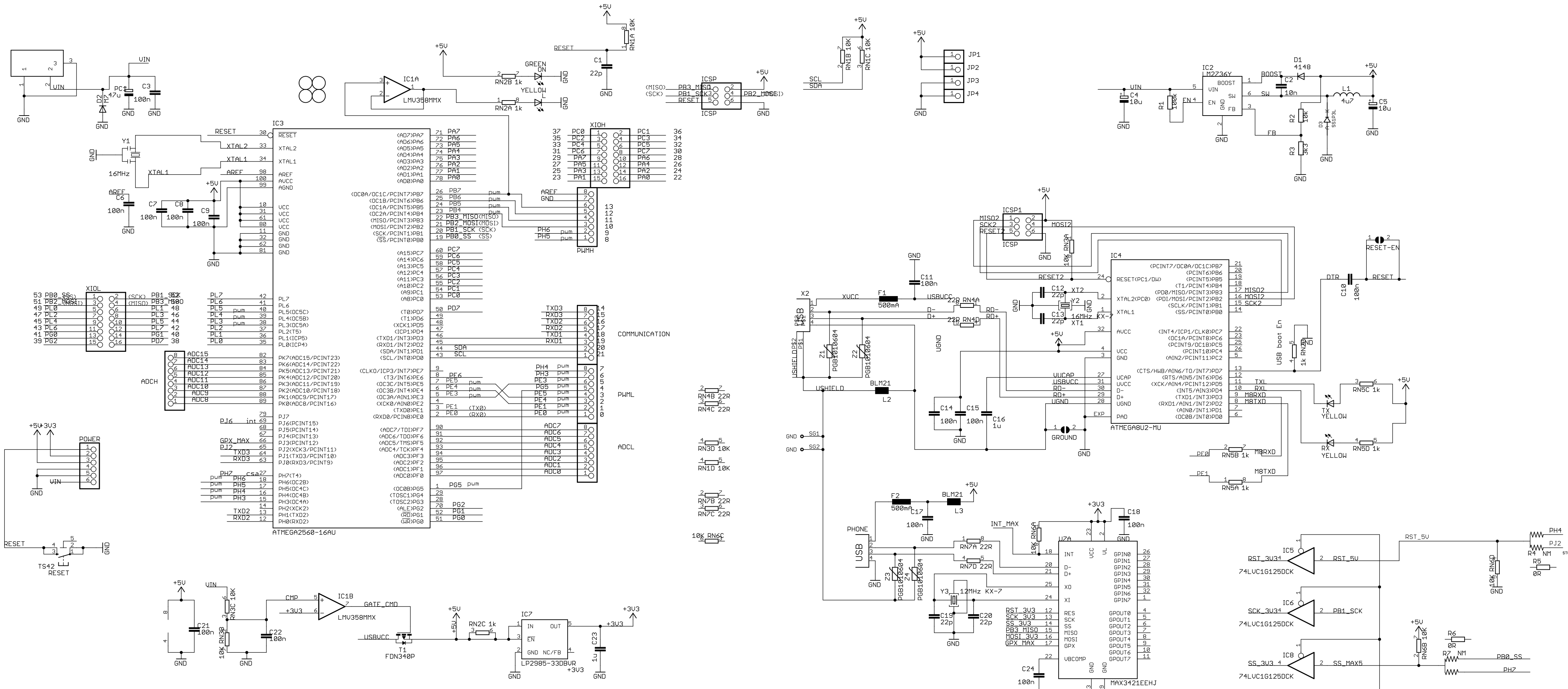
```
/*
uint32_t ipAddress = cc3000.IP2U32(192, 168, 1, 19);
uint32_t netMask = cc3000.IP2U32(255, 255, 255, 0);
uint32_t defaultGateway = cc3000.IP2U32(192, 168, 1, 1);
uint32_t dns = cc3000.IP2U32(8, 8, 4, 4);
if (!cc3000.setStaticIPAddress(ipAddress, netMask, defaultGateway, dns)) {
Serial.println(F("Failed to set static IP!"));
while(1);
}
*/
```

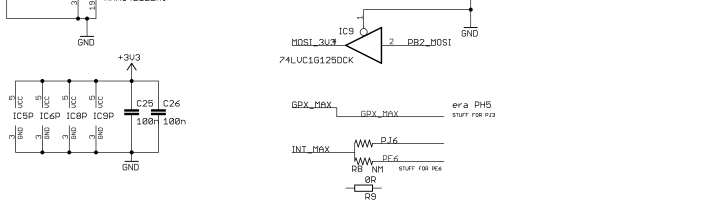
Remove the /* and */ comment delineators and fill in the IP address, net mask, default gateway, and DNS server values for your network. You might need to check your router's configuration page to find these details.

Run the buildtest sketch and the CC3000 should be configured to use the static IP address and configuration you assigned. You can actually remove or comment out the IP assignment code because the CC3000 will remember the configuration in its internal non-volatile storage.

If you'd ever like to enable DHCP again, load buildtest and uncomment the section below the static IP address configuration:

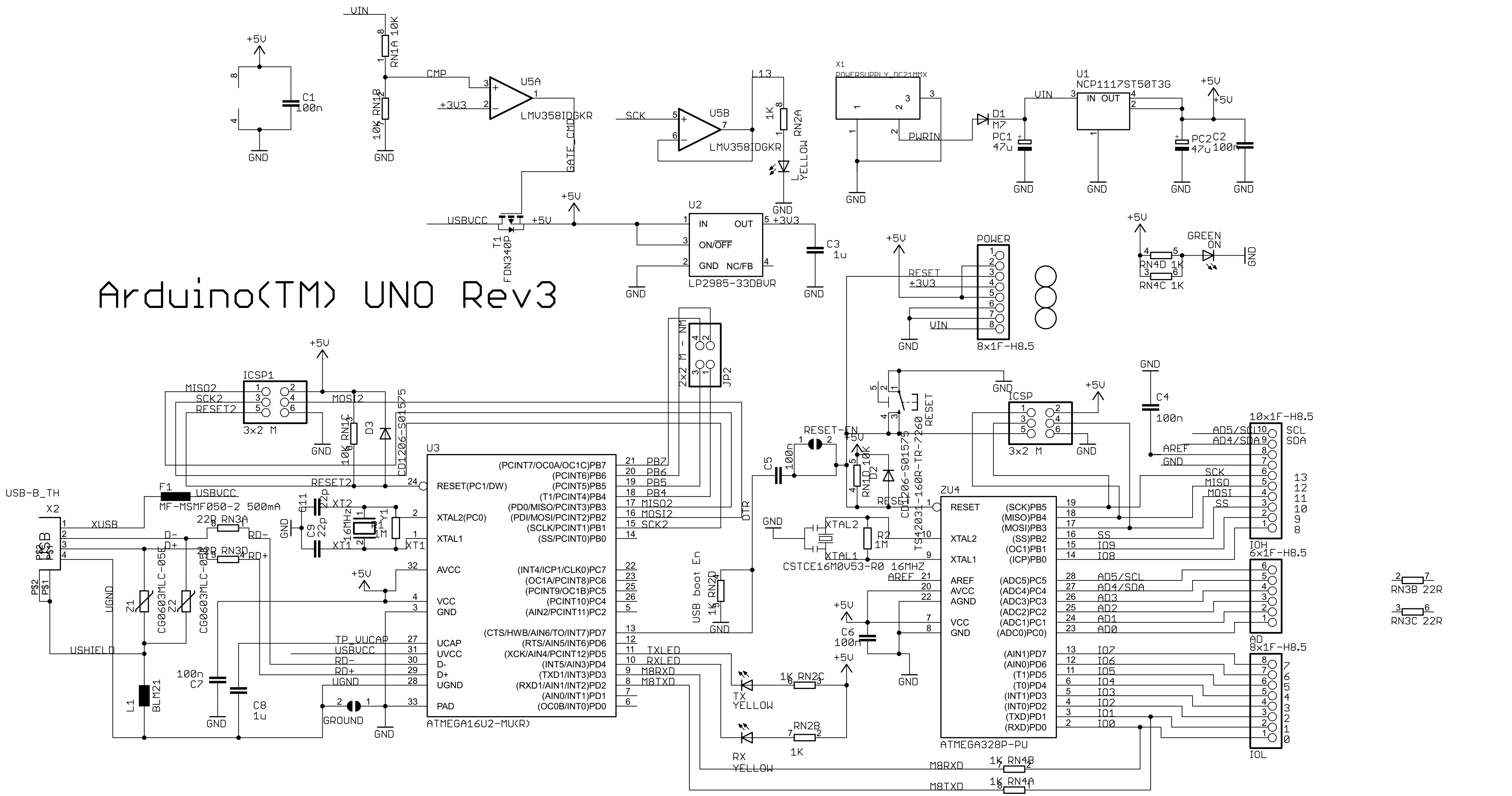
```
/*
if (!cc3000.setDHCP()) {
Serial.println(F("Failed to set DHCP!"));
while(1);
}
*/
```





ARDUINO MEGA2560 ADK (for Android)

Arduino(TM) UNO Rev3



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

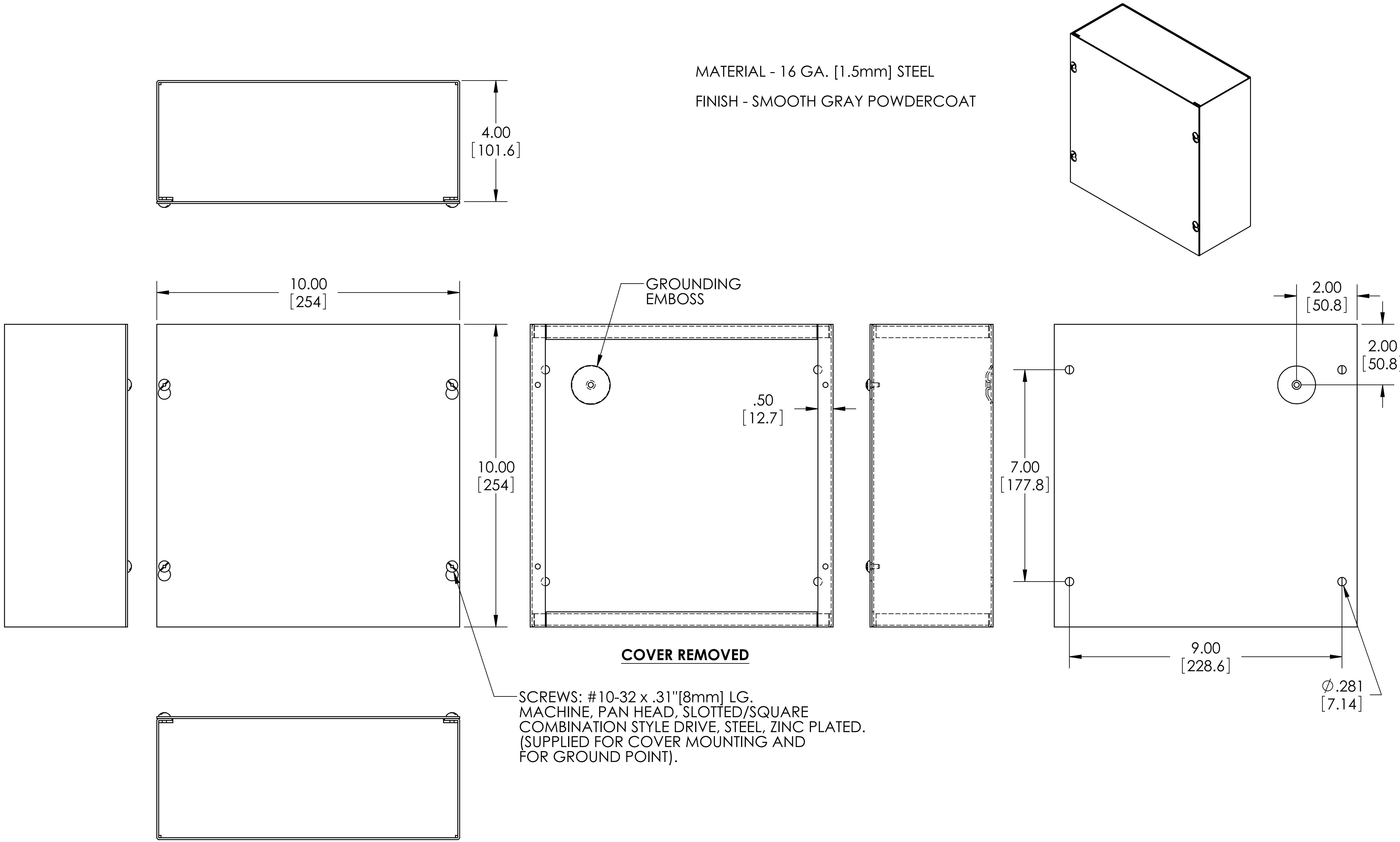
ARDUINO is a registered trademark.

Use of the ARDUINO name must be compliant with http://www.arduino.cc/en/Main/Policy

PART NAME	JUNCTION BOX	PART No.	JB-3960	REVISION DATE	7-18-11

MATERIAL - 16 GA. [1.5mm] STEEL

FINISH - SMOOTH GRAY POWDERCOAT

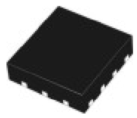


TS2012

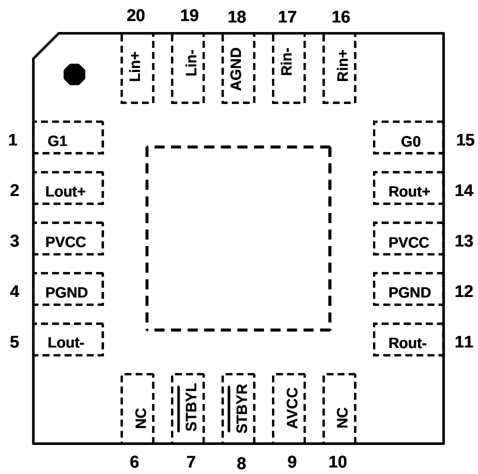
Filter-free stereo 2x2.8 W class D audio power amplifier

Datasheet - production data

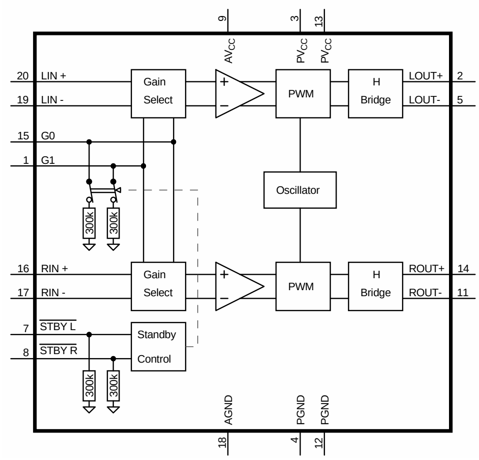
TS2012 - QFN20 (4x4 mm)



Pin connections (top view)



Block diagram



Features

- Operating range from V_{CC} = 2.5 V to 5.5 V
- Standby mode active low
- Output power per channel: 1.35 W @ 5 V or 0.68 W @ 3.6 V into 8 Ω with 1 % THD+N max
- Output power per channel: 2.2 W @ 5 V into 4 Ω with 1 % THD+N max
- Four gains can be selected: 6, 12, 18, 24 dB
- Low current consumption
- PSRR: 70 dB typ @ 217 Hz with 6 dB gain
- Fast startup phase: 1 ms
- Thermal shutdown protection
- QFN20 4x4 mm lead-free package

Applications

- Cellular phone
- PDA
- Flat panel TV

Description

The TS2012 is a fully differential, class D, power amplifier stereo. It is able to drive up to 1.35 W into an 8 Ω load at 5 V per channel. It achieves outstanding efficiency compared to typical class AB audio amps.

The device has four different gain settings utilizing two discrete pins: G0 and G1.

Pop and click reduction circuitry provides low on/off switch noise while allowing the device to start within 1 ms.

Two standby pins (active low) allow each channel to be switched off independently.

The TS2012 is available in a QFN20 4x4 mm package.

Contents

1 Absolute maximum ratings and operating conditions

Table 1. Absolute maximum ratings

Symbol	Parameter	Value	Unit
V_{CC}	Supply voltage [1]	6	V
V_i	Input voltage [2]	GND to V_{CC}	
T_{oper}	Operating free air temperature range	-40 to + 85	°C
T_{stg}	Storage temperature	-65 to +150	
T_j	Maximum junction temperature	150	
R_{thja}	Thermal resistance junction to ambient [3]	100	°C/W
P_d	Power dissipation	Internally limited[4]	
ESD	HBM: human body model[5]	2	kV
	MM: machine model[6]	200	V
Latch-up	Latch-up immunity	200	mA
V_{STBY}	Standby pin voltage maximum voltage	GND to V_{CC}	V
	Lead temperature (soldering, 10 s)	260	°C

1. All voltage values are measured with respect to the ground pin.
2. The magnitude of the input signal must never exceed V_{CC} + 0.3 V / GND - 0.3 V.
3. The device is protected in case of over temperature by a thermal shutdown active @ 150 °C.
4. Exceeding the power derating curves over a long period causes abnormal operation.
5. Human body model: 100 pF discharged through a 1.5 kΩ resistor between two pins of the device, done for all couples of pin combinations with other pins floating.
6. Machine model: a 200 pF cap is charged to the specified voltage, then discharged directly between two pins of the device with no external series resistor (internal resistor < 5 Ω), done for all couples of pin combinations with other pins floating.

Table 2. Operating conditions

Symbol	Parameter	Value	Unit
V_{CC}	Supply voltage	2.5 to 5.5	V
V_I	Input voltage range	GND to V_{CC}	
V_{ic}	Input common mode voltage(1)	GND+0.5V to V_{CC}-0.9V	
V_{STBY}	Standby voltage input (2) Device ON Device in STANDBY(3)	 $1.4 \le V_{STBY} \le V_{CC}$ $GND \le V_{STBY} \le 0.4$	
R_L	Load resistor	≥ 4	Ω
V_{IH}	GO, G1 - high level input voltage(4)	$1.4 \le V_{IH} \le V_{CC}$	V
V_{IL}	GO, G1 - low level input voltage	$GND \le V_{IL} \le 0.4$	
R_{thja}	Thermal resistance junction to ambient(5)	40	°C/W

1. $| V_{oo} | \le 40$ mV max with all differential gains except 24 dB. For 24 dB gain, input decoupling caps are mandatory.
2. Without any signal on V_{STBY}, the device is in standby (internal 300 kΩ ± 20 % pull-down resistor).
3. Minimum current consumption is obtained when V_{STBY} = GND.
4. Between G0, G1pins and GND, there is an internal 300kΩ (±20 %) pull-down resistor. When pins are floating, the gain is 6 dB. In full standby (left and right channels OFF), these resistors are disconnected (HiZ input).
5. With 4-layer PCB.

2 Typical application

Figure 1. Typical application schematics

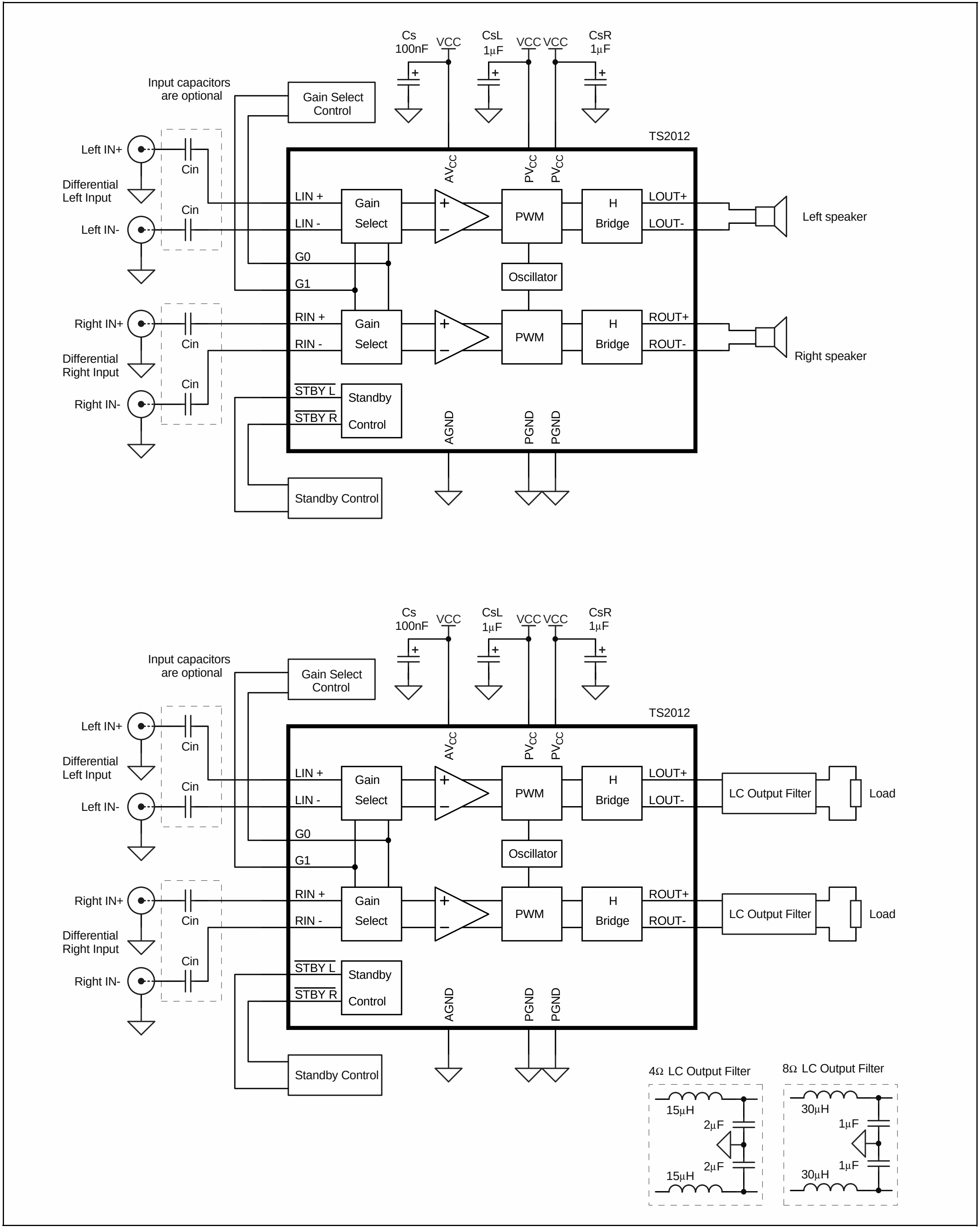


Table 3. External component descriptions

Components	Functional description
C_S, C_{SL}, C_{SR}	Supply capacitor that provides power supply filtering.
C_{in}	Input coupling capacitors (optional) that block the DC voltage at the amplifier input terminal. The capacitors also form a high pass filter with Z_{in} ($F_{cl} = 1 / (2 \times \pi \times Z_{in} \times C_{in})$).

Table 4. Pin descriptions

Pin number	Pin name	Pin description
1	G1	Gain select pin (MSB)
2	Lout+	Left channel positive output
3	PVCC	Power supply
4	PGND	Power ground
5	Lout-	Left channel negative output
6	NC	No internal connection
7	STBYL	Standby pin (active low) for left channel output
8	STBYR	Standby pin (active low) for right channel output
9	AVCC	Analog supply
10	NC	No internal connection
11	Rout-	Right channel negative output
12	PGND	Power ground
13	PVCC	Power supply
14	Rout+	Right channel positive output
15	G0	Gain select pin (LSB)
16	Rin+	Right channel positive differential input
17	Rin-	Right channel negative differential input
18	AGND	Analog ground
19	Lin-	Left channel negative differential input
20	Lin+	Left channel positive differential input
	Thermal pad	Connect the thermal pad of the QFN package to PCB ground

3 Electrical characteristics

3.1 Electrical characteristic tables

Table 5. V_{CC} = +5 V, GND = 0 V, V_{ic} = 2.5 V, T_{amb} = 25 °C (unless otherwise specified)

Symbol	Parameters and test conditions	Min.	Typ.	Max.	Unit
I_{CC}	Supply current No input signal, no load, both channels		5	8	mA
I_{STBY}	Standby current No input signal, V_{STBY} = GND		0.2	2	µA
V_{oo}	Output offset voltage Floating inputs, G = 6dB, R_L = 8Ω			25	mV
P_o	Output power THD + N = 1 % max, f = 1 kHz, R_L = 4 Ω THD + N = 1 % max, f = 1 kHz, R_L = 8 Ω THD + N = 10 % max, f = 1 kHz, R_L = 4 Ω THD + N = 10 % max, f = 1 kHz, R_L = 8 Ω		 2.2 1.35 2.8 1.65		W
THD + N	Total harmonic distortion + noise P_o = 0.8 W, G = 6 dB, f =1 kHz, R_L = 8 Ω		0.07		%
Efficiency	Efficiency per channel P_o = 2.2 W, R_L = 4 Ω +15 µH P_o = 1.25 W, R_L = 8 Ω+15 µH		 81 89		
PSRR	Power supply rejection ratio with inputs grounded C_{in} = 1 µF (1), f = 217 Hz, R_L = 8 Ω, Gain = 6 dB, V_{ripple} = 200 mV_{pp}		70		dB
Crosstalk	Channel separation P_o = 0.9 W, G = 6 dB, f = 1 kHz, R_L = 8 Ω		90		
CMRR	Common mode rejection ratio C_{in} = 1 µF, f = 217 Hz, R_L = 8 Ω, Gain = 6 dB, Δ_{VICM} = 200 mV_{pp}		70		
Gain	Gain value G1 = G0 = V_{IL} G1 = V_{IL} & G0 = V_{IH} G1 = V_{IH} & G0 = V_{IL} G1 = G0 = V_{IH}	 5.5 11.5 17.5 23.5	 6 12 18 24	 6.5 12.5 18.5 24.5	
Z_{in}	Single ended input impedance All gains, referred to ground	24	30	36	kΩ
F_{PWM}	Pulse width modulator base frequency	190	280	370	kHz
SNR	Signal to noise ratio (A-weighting) P_o = 1.3 W, G = 6 dB, R_L = 8 Ω		99		dB
t_{WU}	Wakeup time		1	3	ms
t_{STBY}	Standby time		1		

Table 5. V_{CC} = +5 V, GND = 0 V, V_{ic} = 2.5 V, T_{amb} = 25 °C (unless otherwise specified) (continued)

Symbol	Parameters and test conditions	Min.	Typ.	Max.	Unit
V_N	Output voltage noise f = 20 Hz to 20 kHz, R_L=8 Ω Unweighted (filterless, G = 6 dB) A-weighted (filterless, G = 6 dB) Unweighted (with LC output filter, G = 6 dB) A-weighted (with LC output filter, G = 6 dB) Unweighted (filterless, G = 24 dB) A-weighted (filterless, G = 24 dB) Unweighted (with LC output filter, G = 24 dB) A-weighted (with LC output filter, G = 24 dB)		 63 35 60 35 115 72 109 71		μV_{RMS}

1. Dynamic measurements - 20*log(rms(V_{out})/rms(V_{ripple})). V_{ripple} is the superimposed sinus signal to V_{CC} @ f = 217 Hz.

Table 6. V_{CC} = +3.6 V, GND = 0 V, V_{ic} = 1.8 V, T_{amb} = 25 °C (unless otherwise specified)

Symbol	Parameter	Min.	Typ.	Max.	Unit
I_{CC}	Supply current No input signal, no load, both channels		3.3	6.5	mA
I_{STBY}	Standby current No input signal, V_{STBY} = GND		0.2	2	µA
V_{oo}	Output offset voltage Floating inputs, G = 6dB, R_L = 8Ω			25	mV
P_o	Output power THD + N = 1 % max, f = 1 kHz, R_L = 4 Ω THD + N = 1 % max, f = 1 kHz, R_L = 8 Ω THD + N = 10 % max, f = 1 kHz, R_L = 4 Ω THD + N = 10 % max, f = 1 kHz, R_L = 8 Ω		 1.15 0.68 1.3 0.9		W
THD + N	Total harmonic distortion + noise P_o = 0.4 W, G = 6 dB, f =1 kHz, R_L = 8 Ω		0.05		%
Efficiency	Efficiency per channel P_o = 1.15 W, R_L = 4 Ω +15 µH P_o = 0.68 W, R_L = 8 Ω +15 µH		 80 88		%
PSRR	Power supply rejection ratio with inputs grounded C_{in}= 1 µF [(1)], f = 217 Hz, R_L = 8 Ω, Gain = 6 dB, V_{ripple} = 200 mV_{pp}		70		dB
Crosstalk	Channel separation P_o = 0.5 W, G = 6 dB, f =1 kHz, R_L = 8 Ω		90		dB
CMRR	Common mode rejection ratio C_{in} = 1 µF, f = 217 Hz, R_L = 8 Ω, Gain = 6 dB, Δ_{VICM} = 200 mV_{pp}		70		dB
Gain	Gain value G1 = G0 = V_{IL} G1 = V_{IL} & G0 = V_{IH} G1 = V_{IH} & G0 = V_{IL} G1 = G0 = V_{IH}	 5.5 11.5 17.5 23.5	 6 12 18 24	 6.5 12.5 18.5 24.5	dB
Z_{in}	Single ended input impedance All gains, referred to ground	24	30	36	kΩ
F_{PWM}	Pulse width modulator base frequency	190	280	370	kHz
SNR	Signal to noise ratio (A-weighting) P_o = 0.65 W, G = 6 dB, R_L = 8 Ω		96		dB
t_{WU}	Wakeup time		1	3	ms
t_{STBY}	Standby time		1		ms

Table 6. V_{CC} = +3.6 V, GND = 0 V, V_{ic} = 1.8 V, T_{amb} = 25 °C (unless otherwise specified) (continued)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_N	Output voltage noise f = 20 Hz to 20 kHz, R_L= 4 Ω Unweighted (filterless, G = 6 dB) A-weighted (filterless, G = 6 dB) Unweighted (with LC output filter, G = 6 dB) A-weighted (with LC output filter, G = 6 dB) Unweighted (filterless, G = 24 dB) A-weighted (filterless, G = 24 dB) Unweighted (with LC output filter, G = 24 dB) A-weighted (with LC output filter, G = 24 dB)		 58 34 55 34 111 70 105 69		μV_{RMS}

1. Dynamic measurements - 20*log(rms(V_{out})/rms(V_{ripple})). V_{ripple} is the superimposed sinus signal to V_{CC} @ f = 217 Hz.

Table 7. V_{CC} = +2.5V, GND = 0V, V_{ic}=1.25V, T_{amb} = 25°C (unless otherwise specified)

Symbol	Parameter	Min.	Typ.	Max.	Unit
I_{CC}	Supply current No input signal, no load, both channels		2.8	4	mA
I_{STBY}	Standby current No input signal, V_{STBY} = GND		0.2	2	µA
V_{oo}	Output offset voltage Floating inputs, G = 6dB, R_L = 8Ω			25	mV
P_o	Output power THD + N = 1 % max, f = 1 kHz, R_L = 4 Ω THD + N = 1 % max, f = 1 kHz, R_L = 8 Ω THD + N = 10 % max, f = 1 kHz, R_L = 4 Ω THD + N = 10 % max, f = 1 kHz, R_L = 8 Ω		0.53 0.32 0.75 0.45		W
THD + N	Total harmonic distortion + noise P_o = 0.2 W, G = 6 dB, f =1 kHz, R_L = 8 Ω		0.04		%
Efficiency	Efficiency per channel P_o = 0.53 W, R_L = 4 Ω +15 µH P_o = 0.32 W, R_L = 8 Ω+15 µH		80 88		
PSRR	Power supply rejection ratio with inputs grounded C_{in} = 1 µF (1), f = 217 Hz, R_L = 8 Ω, Gain = 6 dB, V_{ripple} = 200 mV_{pp}		70		dB
Crosstalk	Channel separation P_o = 0.2 W, G = 6 dB, f = 1 kHz, R_L = 8 Ω		90		
CMRR	Common mode rejection ratio C_{in} = 1 µF, f = 217 Hz, R_L = 8 Ω, Gain = 6 dB, Δ_{VICM} = 200 mV_{pp}		70		
Gain	Gain value G1 = G0 = V_{IL} G1 = V_{IL} & G0 = V_{IH} G1 = V_{IH} & G0 = V_{IL} G1 = G0 = V_{IH}	5.5 11.5 17.5 23.5	6 12 18 24	6.5 12.5 18.5 24.5	
Z_{in}	Single ended input impedance All gains, referred to ground	24	30	36	kΩ
F_{PWM}	Pulse width modulator base frequency	190	280	370	kHz
SNR	Signal to noise ratio (A-weighting) P_o = 0.3 W, G = 6 dB, R_L = 8 Ω		93		dB
t_{WU}	Wakeup time		1	3	ms
t_{STBY}	Standby time		1		

Table 7. V_{CC} = +2.5V, GND = 0V, V_{ic}=1.25V, T_{amb} = 25°C (unless otherwise specified) (continued)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_N	Output voltage noise f = 20 Hz to 20 kHz, R_L = 8 Ω Unweighted (filterless, G = 6 dB) A-weighted (filterless, G = 6 dB) Unweighted (with LC output filter, G = 6 dB) A-weighted (with LC output filter, G = 6 dB) Unweighted (filterless, G = 24 dB) A-weighted (filterless, G = 24 dB) Unweighted (with LC output filter, G = 24 dB) A-weighted (with LC output filter, G = 24 dB)		 57 34 54 33 110 71 104 69		μV_{RMS}

1. Dynamic measurements - 20*log(rms(V_{out})/rms(V_{ripple})). V_{ripple} is the superimposed sinus signal to V_{CC} @ f = 217 Hz.

3.2 Electrical characteristic curves

The graphs shown in this section use the following abbreviations:

- R_L+ 15 µH or 30 µH = pure resistor + very low series resistance inductor
- Filter = LC output filter (1 µF + 30 µH for 4 Ω and 0.5 µF + 60 µH for 8 Ω)

All measurements are made with C_{SL} = C_{SR} = 1 µF and C_S = 100 nF (see *Figure 2*), except for the PSRR where $C_{SL,R}$ is removed (see *Figure 3*).

Figure 2. Test diagram for measurements

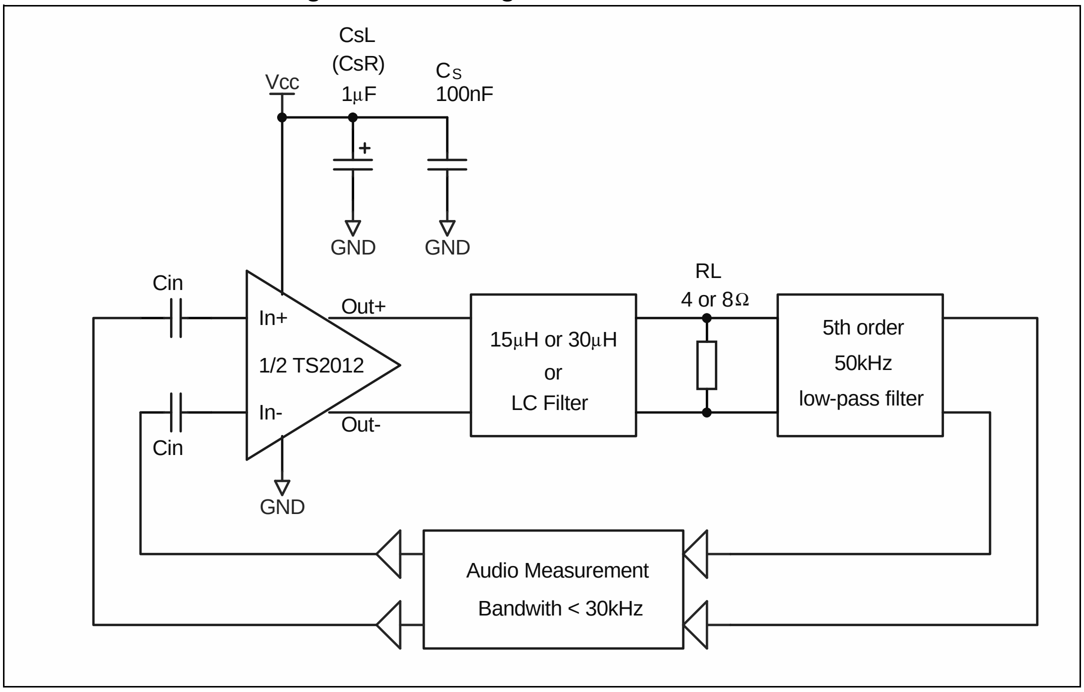


Figure 3. Test diagram for PSRR measurements

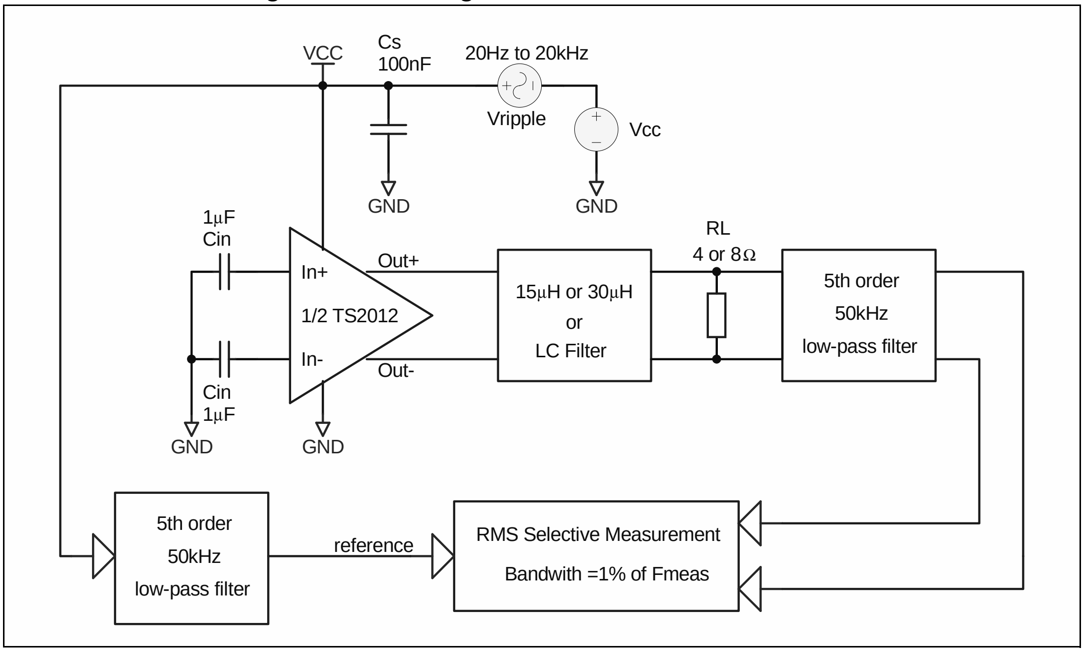


Table 8. Index of graphics

Description	Figure
Current consumption vs. power supply voltage	*Figure 4*
Current consumption vs. standby voltage	*Figure 5*
Efficiency vs. output power	*Figure 6* - *Figure 9*
Output power vs. power supply voltage	*Figure 10*, *Figure 11*
PSRR vs. common mode input voltage	*Figure 12*
PSRR vs. frequency	*Figure 13*
CMRR vs. common mode input voltage	*Figure 14*
CMRR vs. frequency	*Figure 15*
Gain vs. frequency	*Figure 16*, *Figure 17*
THD+N vs. output power	*Figure 18* - *Figure 25*
THD+N vs. frequency	*Figure 26* - *Figure 37*
Crosstalk vs. frequency	*Figure 38* - *Figure 41*
Power derating curves	*Figure 42*
Startup and shutdown time	*Figure 43*, *Figure 44*

Figure 4. Current consumption vs. power supply voltage

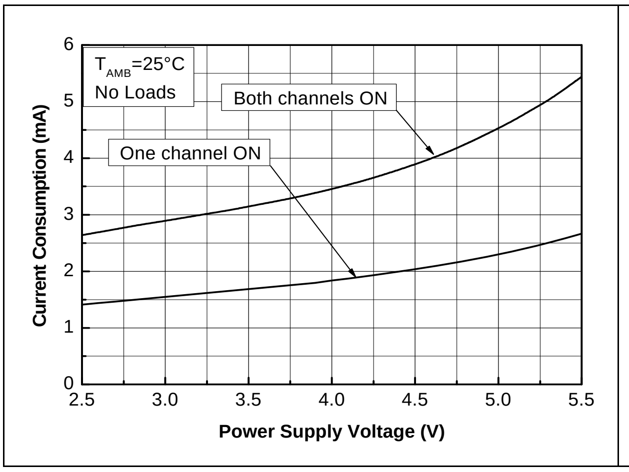


Figure 5. Current consumption vs. standby voltage (one channel)

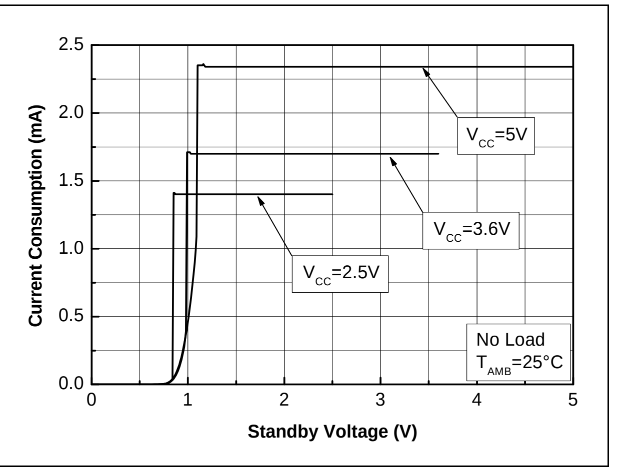


Figure 6. Efficiency vs. output power (i)

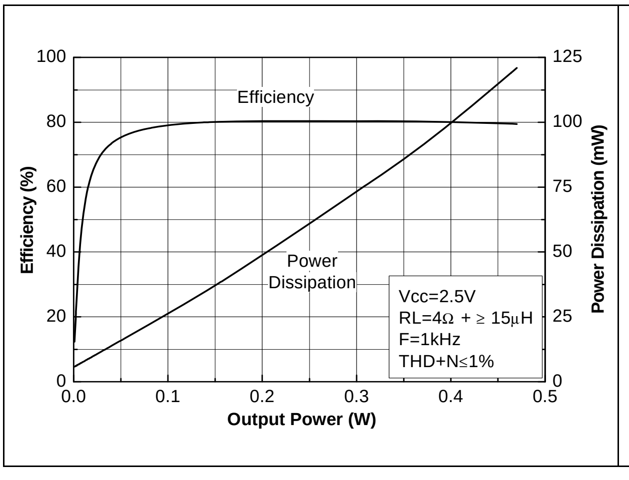


Figure 7. Efficiency vs. output power (ii)

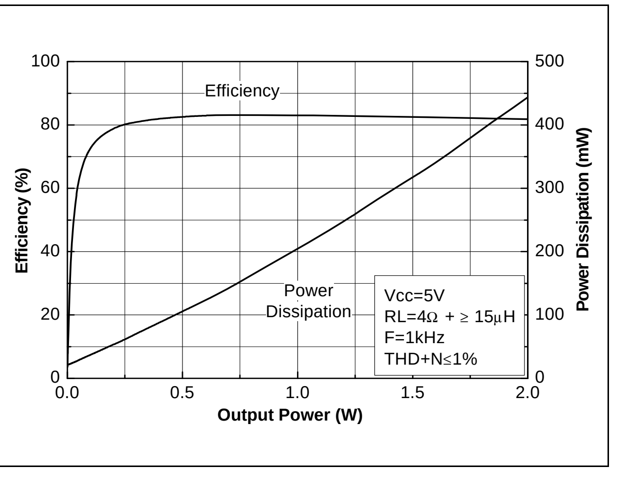


Figure 8. Efficiency vs. output power (iii)

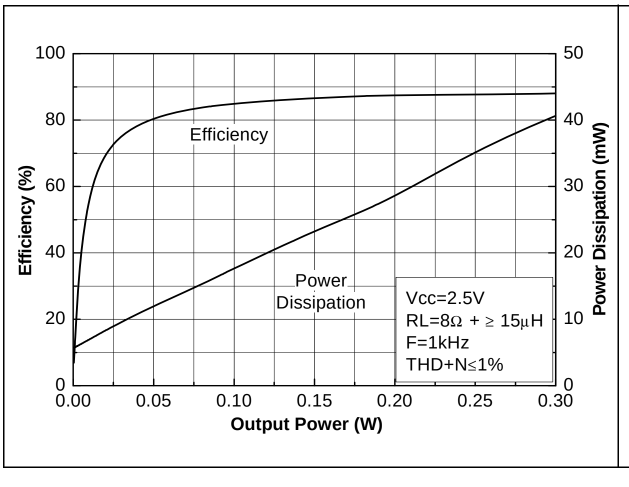


Figure 9. Efficiency vs. output power (iv)

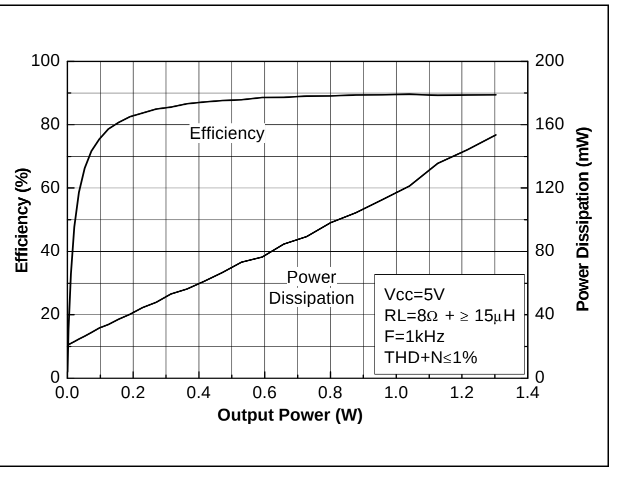


Figure 10. Output power vs. power supply voltage (i)

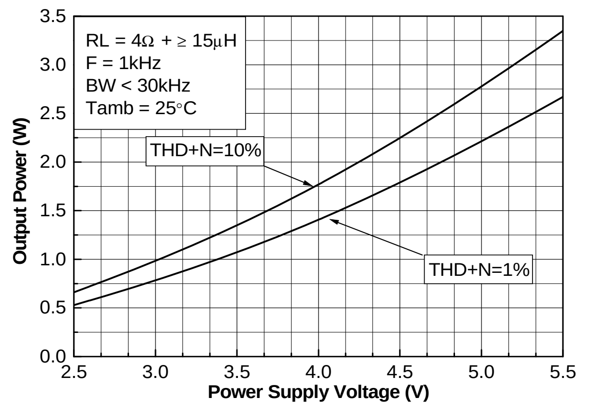


Figure 11. Output power vs. power supply voltage (ii)

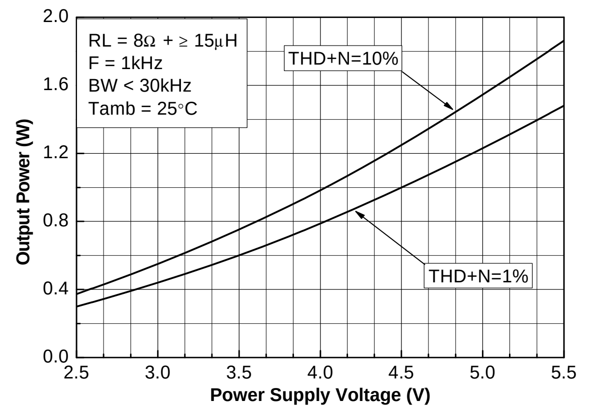


Figure 12. PSRR vs. common mode input voltage

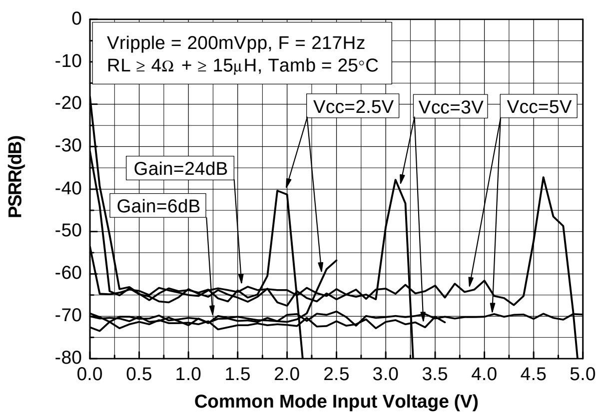


Figure 13. PSRR vs. frequency

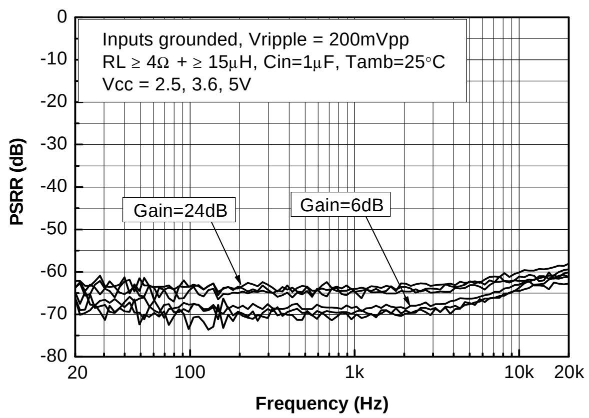


Figure 14. CMRR vs. common mode input voltage

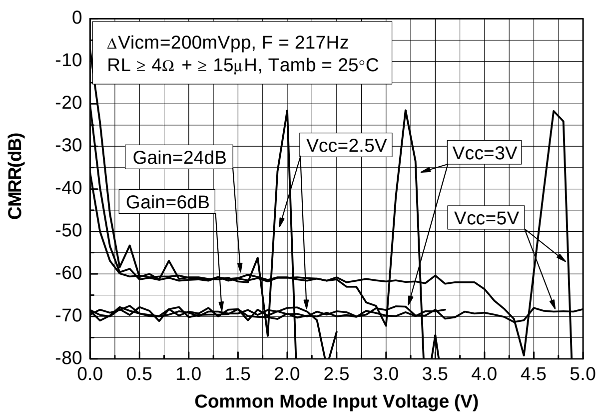


Figure 15. CMRR vs. frequency

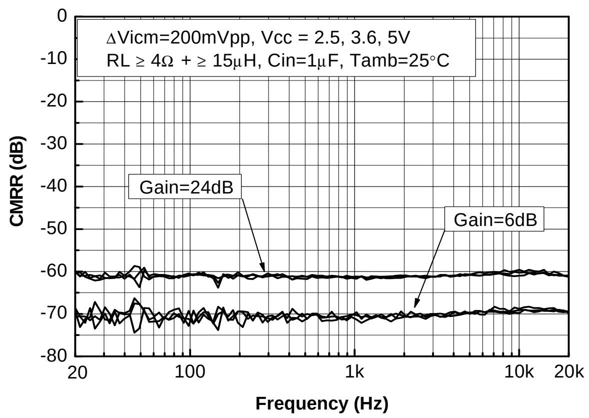


Figure 16. Gain vs. frequency (i)

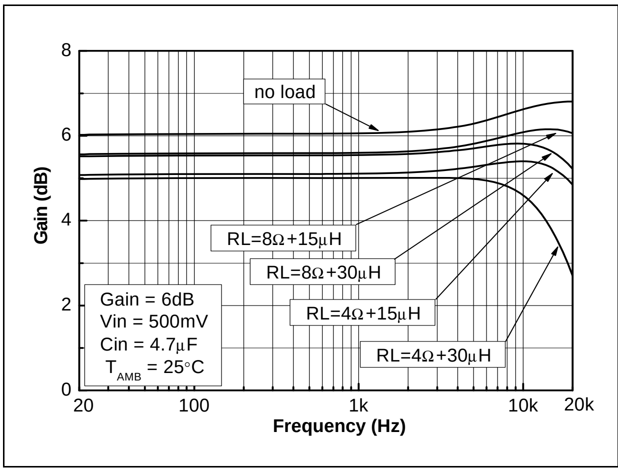


Figure 17. Gain vs. frequency (ii)

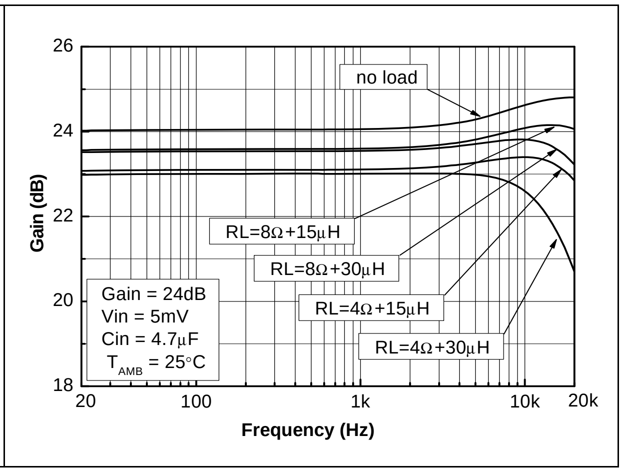


Figure 18. THD+N vs. output power (i)

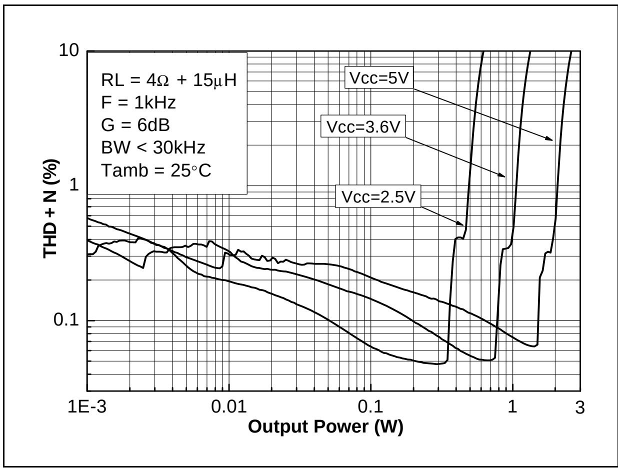


Figure 19. THD+N vs. output power (ii)

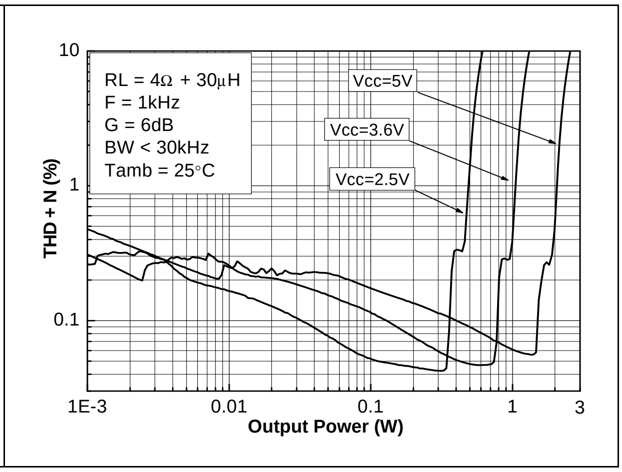


Figure 20. THD+N vs. output power (iii)

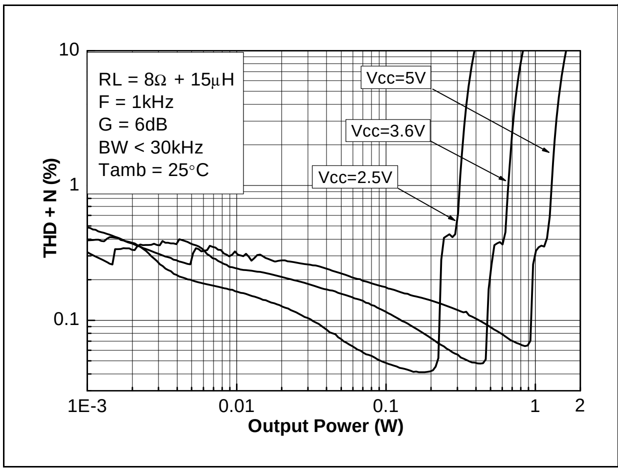


Figure 21. THD+N vs. output power (iv)

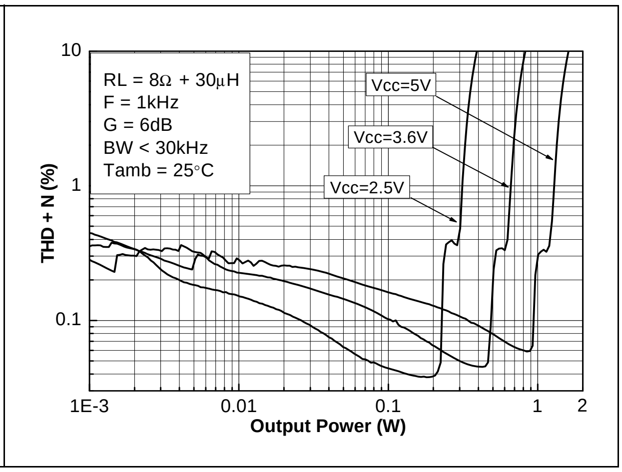


Figure 22. THD+N vs. output power (v)

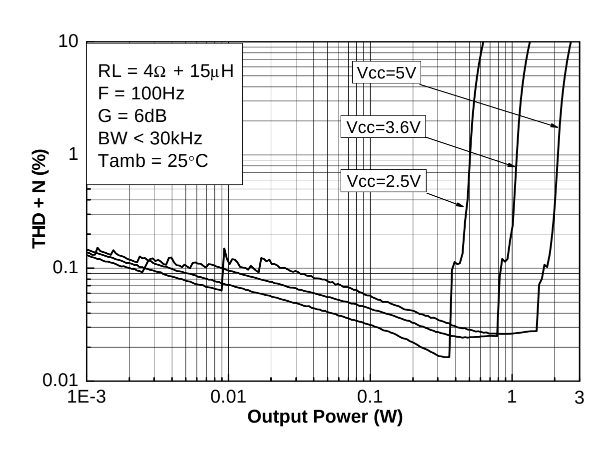


Figure 23. THD+N vs. output power (vi)

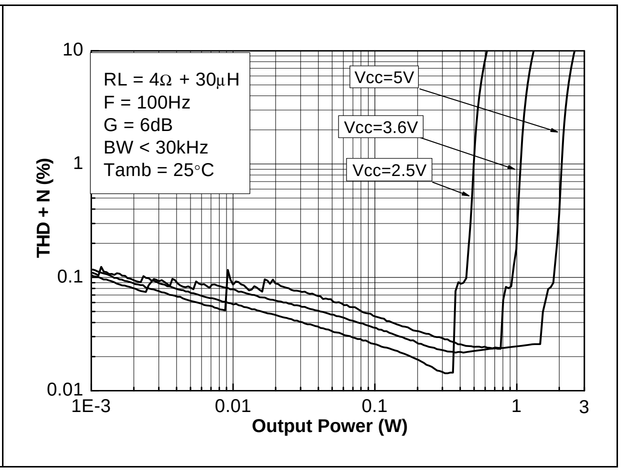


Figure 24. THD+N vs. output power (vii)

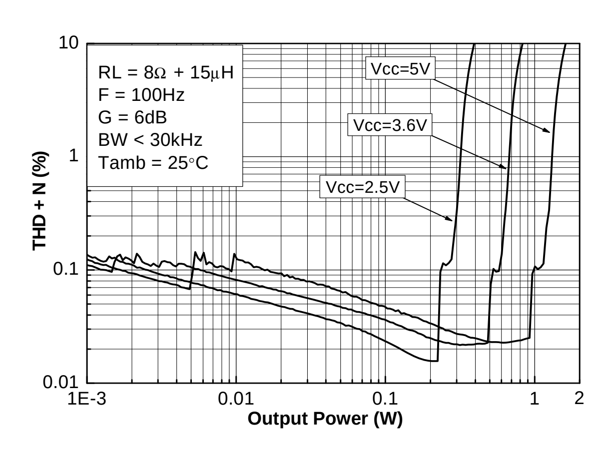


Figure 25. THD+N vs. output power (viii)

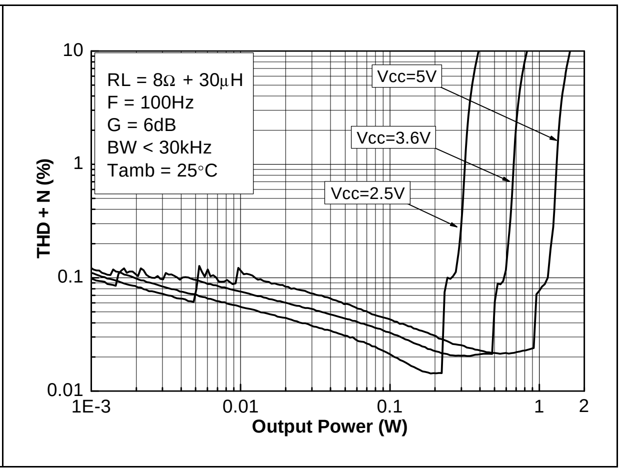


Figure 26. THD+N vs. frequency (i)

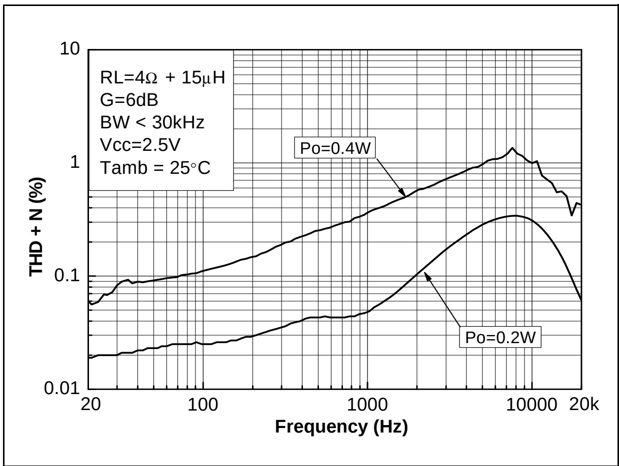


Figure 27. THD+N vs. frequency (ii)

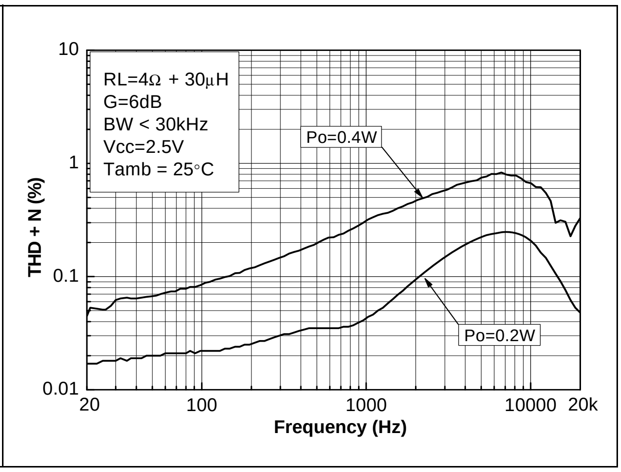


Figure 28. THD+N vs. frequency (iii)

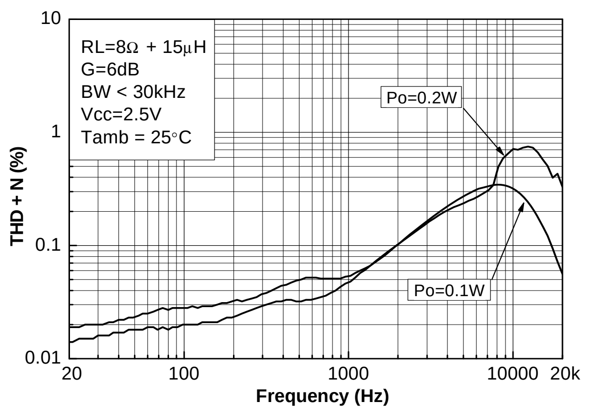


Figure 29. THD+N vs. frequency (iv)

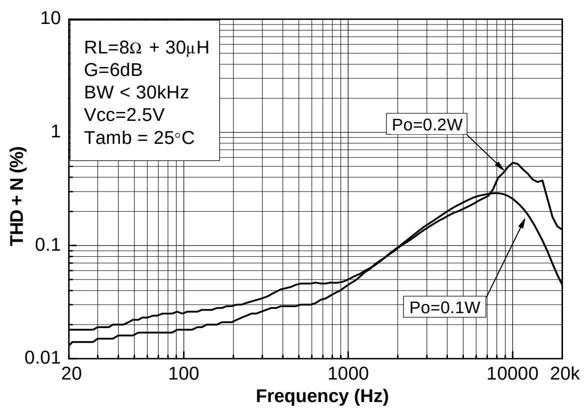


Figure 30. THD+N vs. frequency (v)

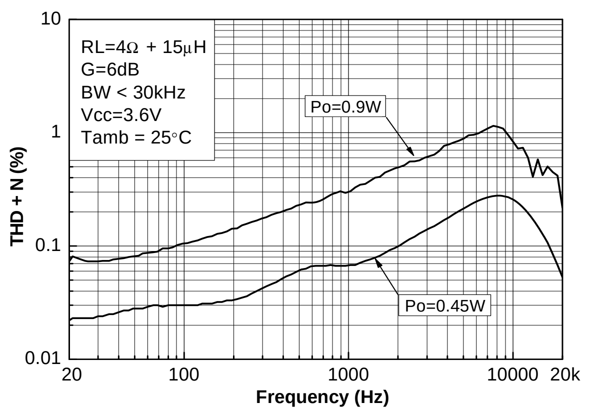


Figure 31. THD+N vs. frequency (vi)

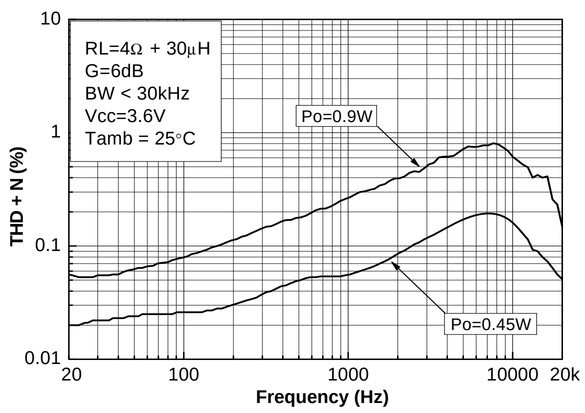


Figure 32. THD+N vs. frequency (vii)

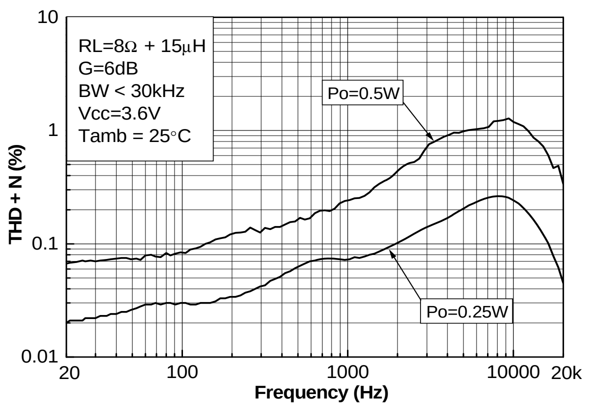


Figure 33. THD+N vs. frequency (viii)

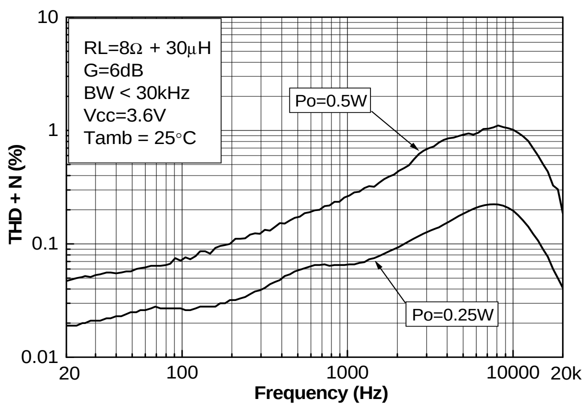


Figure 34. THD+N vs. frequency (ix)

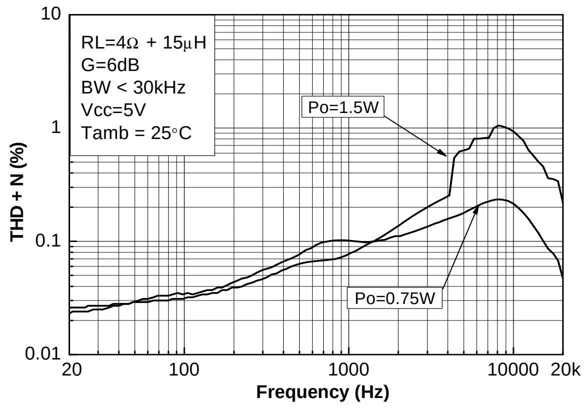


Figure 35. THD+N vs. frequency (x)

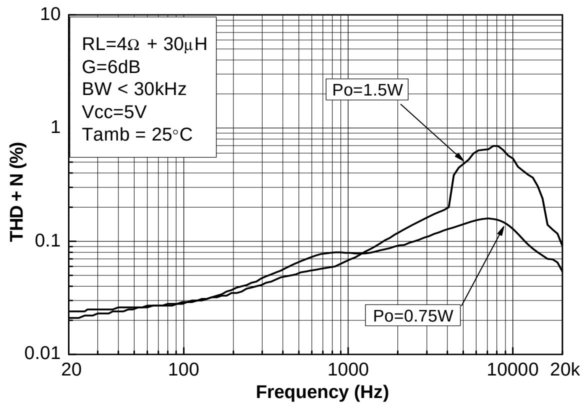


Figure 36. THD+N vs. frequency (xi)

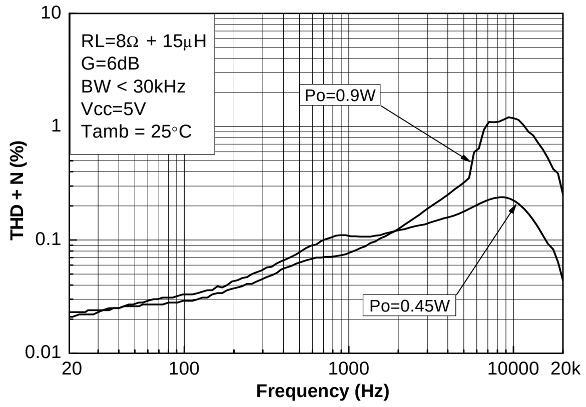


Figure 37. THD+N vs. frequency (xii)

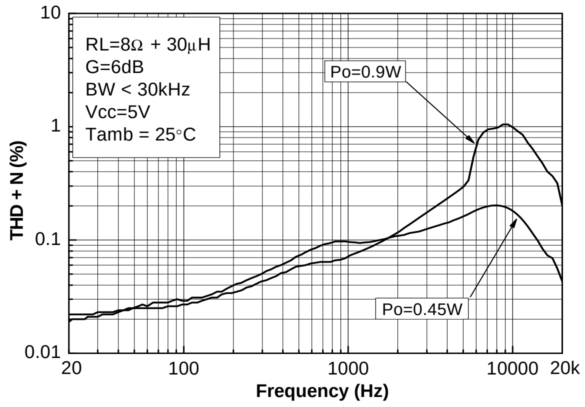


Figure 38. Crosstalk vs. frequency (i)

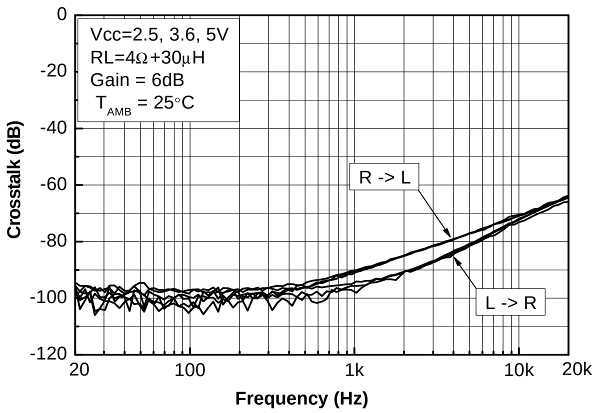


Figure 39. Crosstalk vs. frequency (ii)

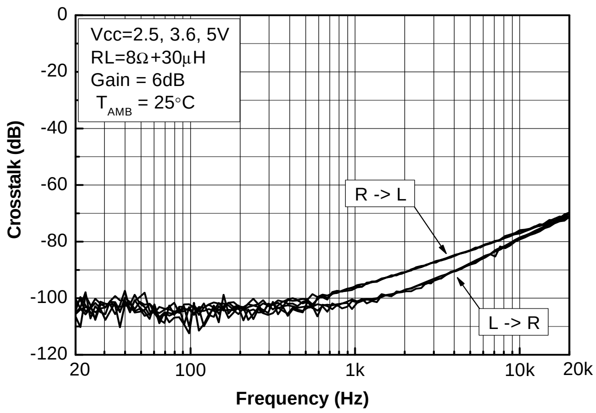


Figure 40. Crosstalk vs. frequency (iii)

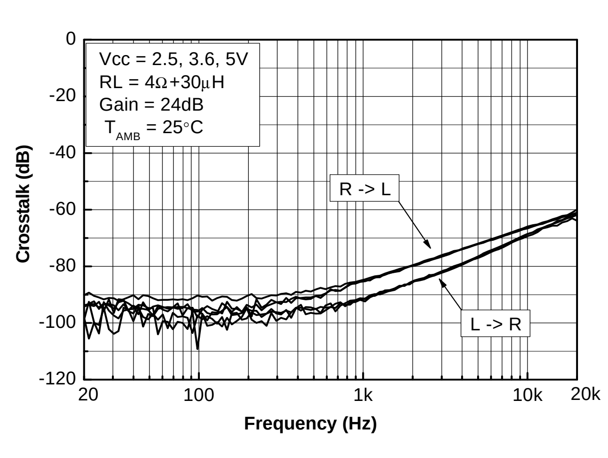


Figure 41. Crosstalk vs. frequency (iv)

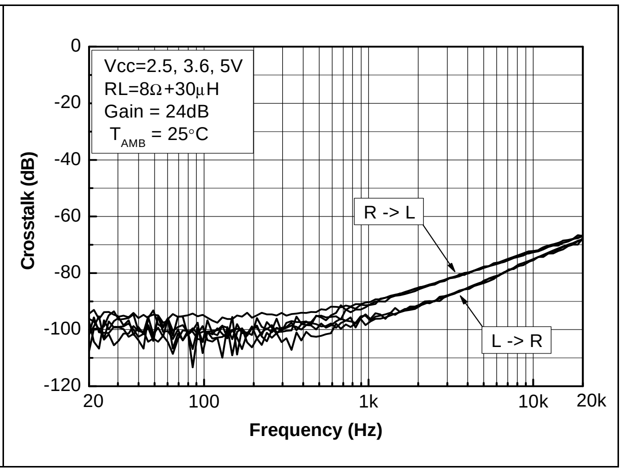


Figure 42. Power derating curves

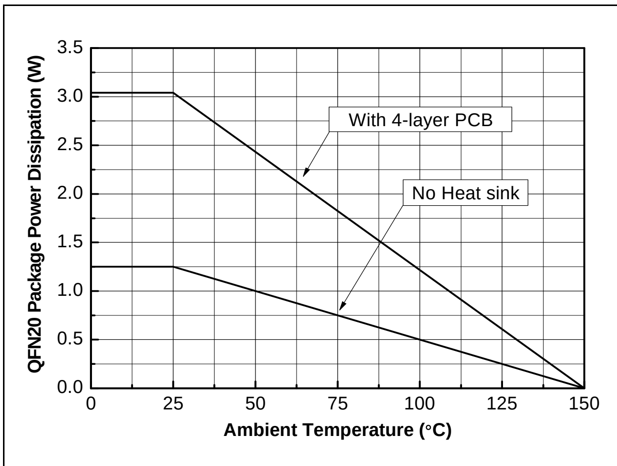


Figure 43. Startup and shutdown phase (i)

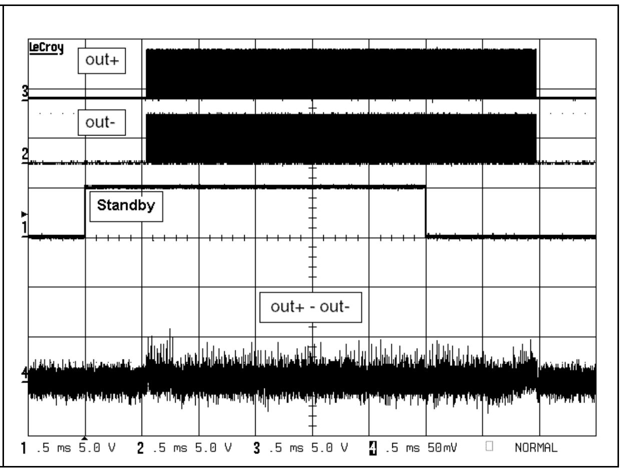
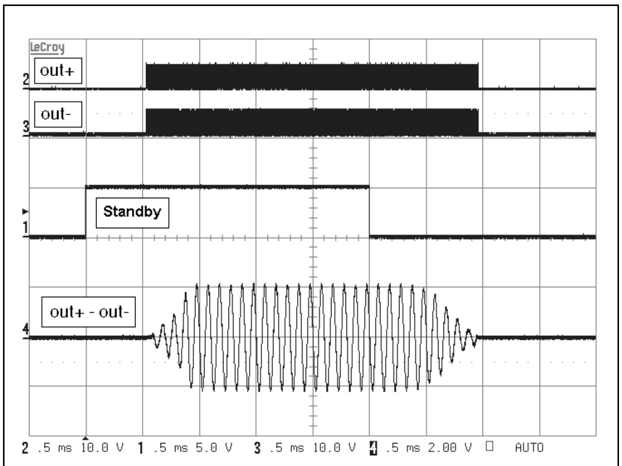


Figure 44. Startup and shutdown phase (ii)



4 Application information

4.1 Differential configuration principle

The TS2012 is a monolithic fully-differential input/output class D power amplifier. The TS2012 also includes a common-mode feedback loop that controls the output bias value to average it at $V_{CC}/2$ for any DC common mode input voltage. This allows the device to always have a maximum output voltage swing, and by consequence, maximize the output power. Moreover, as the load is connected differentially compared with a single-ended topology, the output is four times higher for the same power supply voltage.

The **advantages** of a full-differential amplifier are:

- High PSRR (power supply rejection ratio)
- High common mode noise rejection
- Virtually zero pop without additional circuitry, giving a faster startup time compared with conventional single-ended input amplifiers
- Easier interfacing with differential output audio DAC
- No input coupling capacitors required thanks to common mode feedback loop

4.2 Gain settings

In the flat region of the frequency-response curve (no input coupling capacitor or internal feedback loop + load effect), the differential gain can be set to 6, 12 18, or 24 dB depending on the logic level of the G0 and G1 pins, as shown in *Table 9*.

Table 9. Gain settings with G0 and G1 pins

G1	G0	Gain (dB)	Gain (V/V)
0	0	6	2
0	1	12	4
1	0	18	8
1	1	24	16

Note: Between pins G0, G1 and GND there is an internal 300 kΩ(±20 %) resistor. When the pins are floating, the gain is 6 dB. In full standby (left and right channels OFF), these resistors are disconnected (HiZ input).

4.3 Common mode feedback loop limitations

The common mode feedback loop allows the output DC bias voltage to be averaged at $V_{CC}/2$ for any DC common mode bias input voltage.

Due to the V_{ic} limitation of the input stage (see *Table 2: Operating conditions*), the common mode feedback loop can fulfill its role only within the defined range.

4.4 Low frequency response

If a low frequency bandwidth limitation is required, it is possible to use input coupling capacitors. In the low frequency region, the input coupling capacitor, C_{in}, starts to have an effect. C_{in}, with the input impedance Z_{in}, forms a first order, high-pass filter with a -3 dB cut-off frequency (see *Table 5* to *Table 7*) as shown in *Equation 1*:

Equation 1

$$F_{CL} = \frac{1}{2 \cdot \pi \cdot Z_{in} \cdot C_{in}}$$

So, for a desired cut-off frequency, $F_{CL,}$ C_{in} is calculated as shown in *Equation 2*:

Equation 2

$$C_{in} = \frac{1}{2 \cdot \pi \cdot Z_{in} \cdot F_{CL}}$$

with F_{CL} in Hz, Z_{in} in Ω and C_{in} in F.

The input impedance Z_{in} is typically 30 kΩ for the whole power supply voltage range. There is also a tolerance around the typical value (see *Table 5* to *Table 7*). The maximum and minimum tolerance of the F_{CL} can be calculated using *Equation 3* and *Equation 4* respectively.

Equation 3

$$F_{CLmax} = 1.103 \cdot F_{CL}$$

Equation 4

$$F_{CLmin} = 0.915 \cdot F_{CL}$$

4.5 Decoupling of the circuit

Power supply capacitors, referred to as C_S, C_{SL}, and C_{SR} are needed to correctly bypass the TS2012.

The TS2012 has a typical switching frequency of 280 kHz and an output fall and rise time of about 5 ns. Due to these very fast transients, careful decoupling is mandatory.

A 1 µF ceramic capacitor between each PVCC and PGND and also between AVCC and AGND is enough, but they must be located very close to the TS2012 in order to avoid any extra parasitic inductance created by a long track wire. Parasitic loop inductance, in relation to di/dt, introduces overvoltage that decreases the global efficiency of the device and may also cause a TS2012 breakdown if the parasitic inductance is too high.

In addition, even if a ceramic capacitor has an adequate high frequency ESR value, its current capability is also important. A 0603 size is a good compromise, particularly when a 4 Ω load is used.

Another important parameter is the rated voltage of the capacitor. A 1 µF/6.3 V capacitor used at 5 V, loses about 50 % of its value. With a power supply voltage of 5 V, the decoupling value, instead of 1 µF, could be reduced to 0.5 µF. As C_S has particular influence on the THD+N in the medium to high frequency region, this capacitor variation becomes decisive. In addition, less decoupling means higher overshoots which can be problematic if they reach the power supply AMR value (6 V).

4.6 Wakeup time (t_{wu})

When standby is released to set the device ON, there is typically a delay of 1 ms. The TS2012 has an internal digital delay that mutes the outputs and releases them after this delay time to avoid any pop noise.

Note: The gain increases smoothly (see Figure 44) from the mute to the gain selected by the G1 and G0 pin (Section 4.2).

4.7 Shutdown time

When the standby command is set, the time required to set the output stage to high impedance and to put the internal circuitry in shutdown mode, is typically 1 ms. This time is used to decrease the gain and avoid any pop noise during shutdown.

Note: The gain decreases smoothly until the outputs are muted (see Figure 44).

4.8 Consumption in shutdown mode

Between the shutdown pin and GND there is an internal 300 kΩ (±20 %) resistor. This resistor forces the TS2012 to be in shutdown when the shutdown input is left floating.

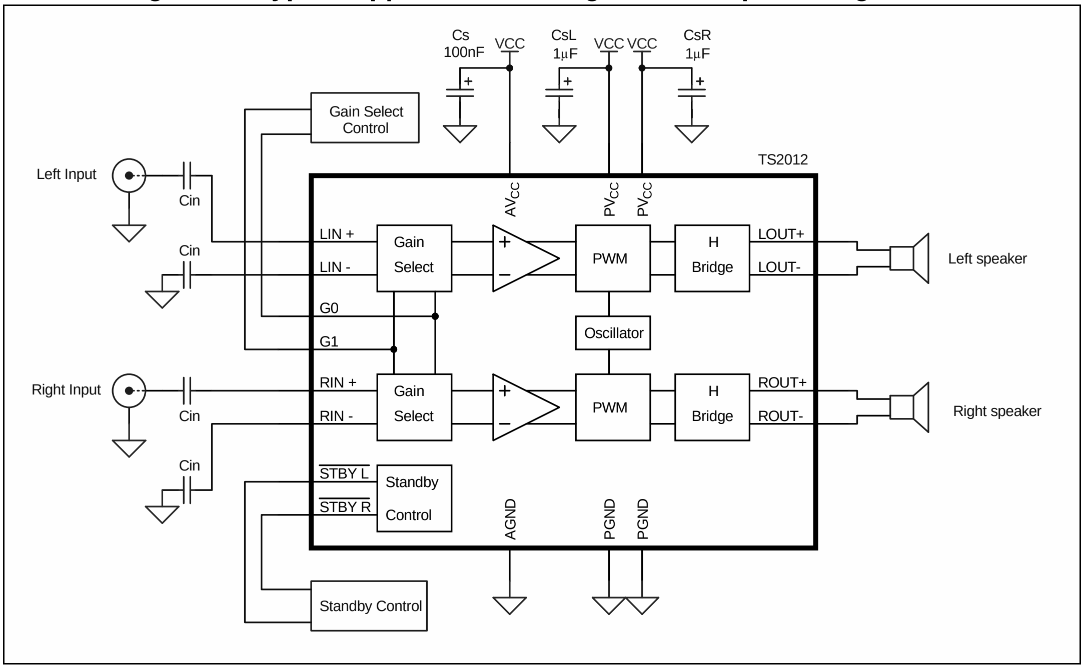
However, this resistor also introduces additional shutdown power consumption if the shutdown pin voltage is not 0 V.

For example, with a 0.4 V shutdown voltage pin, the following typical and maximum values respectively for each shutdown pin must be added to the standby current specified in *Table 5* to *Table 7*: 0.4 V/300 kΩ = 1.3 µA and 0.4 V/240 kΩ = 1.66 µA. This current is provided by the external control device for standby pins.

4.9 Single-ended input configuration

It is possible to use the TS2012 in a single-ended input configuration. However, input coupling capacitors are mandatory in this configuration. The schematic diagram in *Figure 45* shows a typical single-ended input application.

Figure 45. Typical application for single-ended input configuration



4.10 Output filter considerations

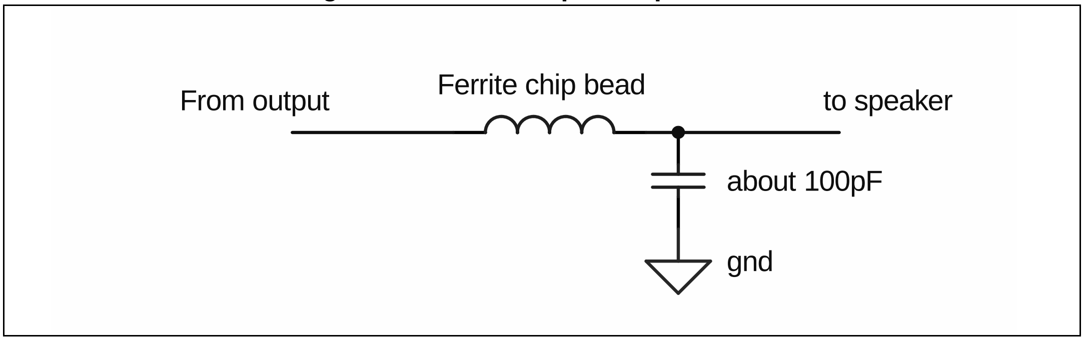
The TS2012 is designed to operate without an output filter. However, due to very sharp transients on the TS2012 output, EMI radiated emissions may cause some standard compliance issues.

These EMI standard compliance issues can appear if the distance between the TS2012 outputs and the loudspeaker terminal are long (typically more than 50 mm or 100 mm, in both directions, to the speaker terminals). As the PCB layout and internal equipment device are different for each configuration, it is difficult to provide a one-size-fits-all solution.

However, to decrease the probability of EMI issues, the following simple rules should be followed:

- Reduce as much as possible the distance between the TS2012 output pins and the speaker terminals.
- Use a ground plane for “shielding” sensitive wires.
- Place, as close as possible to the TS2012 and in series with each output, a ferrite bead with a minimum rated current of 2.5 A and an impedance greater than 50 Ω at frequencies above 30 MHz. If, after testing, these ferrite beads are not necessary, replace them by a short-circuit.
- Allow extra footprint to place, if necessary, a capacitor to short perturbations to ground (see *Figure 46*).

Figure 46. Ferrite chip bead placement



If the distance between the TS2012 output and the speaker terminals is too long, it is possible to have low frequency EMI issues due to the fact that the typical operating frequency is 280 kHz. In this configuration, it is necessary to place the output filter shown in *Figure 1: Typical application schematics* as close as possible to the TS2012.

5 Package information

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: *www.st.com*. ECOPACK® is an ST trademark.

5.1 QFN20 package information

The QFN20 package has an exposed pad E2 x D2. For enhanced thermal performance, the exposed pad must be soldered to a copper area on the PCB, acting as a heatsink. This copper area can be electrically connected to pin 4, 12, 18 (PGND, AGND) or left floating.

Figure 47. QFN20 package mechanical drawing

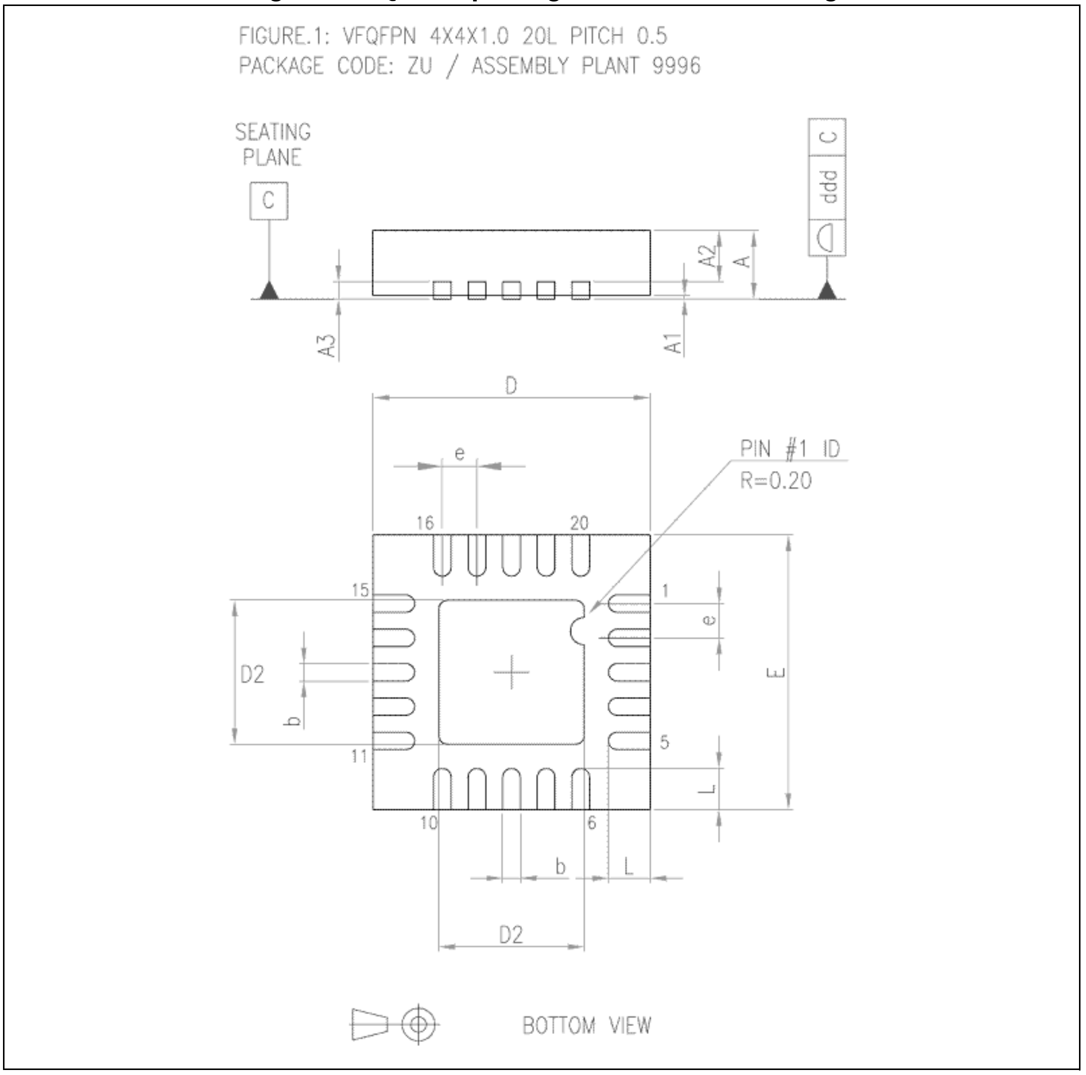


Table 10. QFN20 package mechanical data

Ref	Dimensions in mm			Dimensions in inches		
	Min	Typ	Max	Min	Typ	Max
A	0.8	0.9	1	0.031	0.035	0.039
A1		0.02	0.05		0.001	0.002
A2		0.65	1		0.026	0.039
A3		0.25			0.010	
b	0.18	0.23	0.3	0.007	0.009	0.012
D	3.85	4	4.15	0.152	0.157	0.163
D2		2.6			0.102	
E	3.85	4	4.15	0.152	0.157	0.163
E2		2.6			0.102	
e	0.45	0.5	0.55	0.018	0.020	0.022
L	0.3	0.4	0.5	0.012	0.016	0.020
ddd			0.08			0.003

Figure 48. QFN20 footprint recommendation

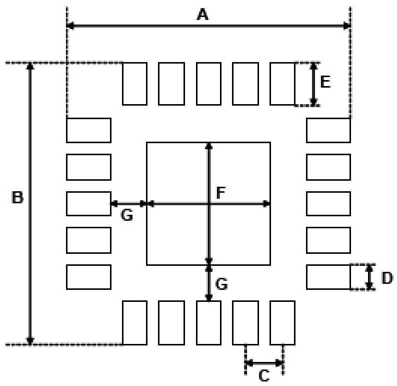


Table 11. QFN20 footprint data

Ref.	Dimensions in mm	Dimensions in inches
A	4.55	0.179
B		
C	0.50	0.020
D	0.35	0.014
E	0.65	0.026
F	2.45	0.096
G	0.40	0.016

6 Ordering information

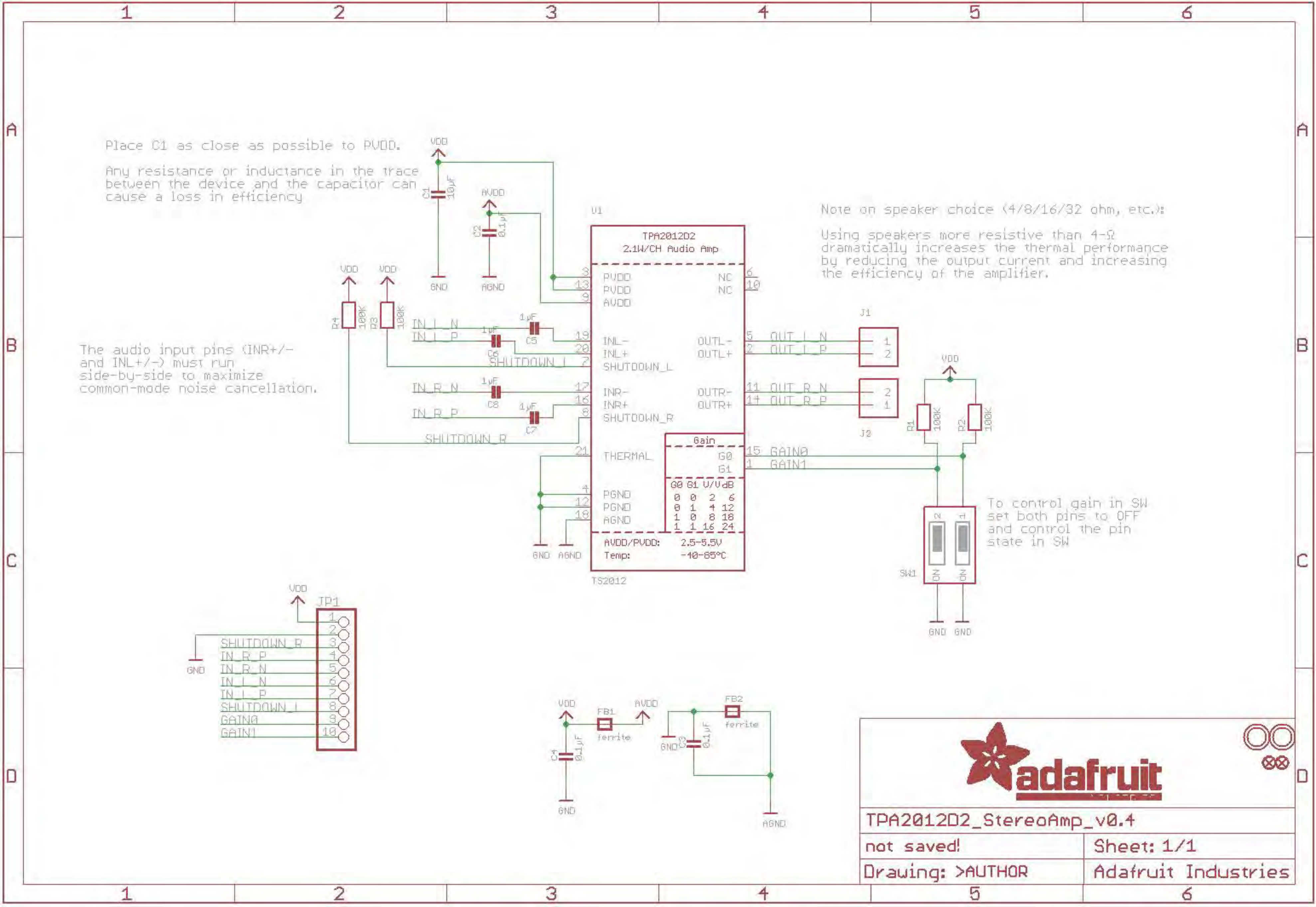
Table 12. Order code

Part number	Temperature range	Package	Packaging	Marking
TS2012IQT	-40 °C to +85 °C	QFN20	Tape and reel	K012

7 Revision history

Table 13. Document revision history

Date	Revision	Changes
17-Dec-2007	1	First release.
17-Jul-2013	2	Small text changes throughout document. Updated titles of *Figure 6* to *Figure 11* and *Figure 16* to *Figure 44* *Table 10: QFN20 package mechanical data*: added package mechanical dimensions in inches. Added *Table 11: QFN20 footprint data* *Table 12: Order code*; updated “Marking”



Place C1 as close as possible to PVDD.
Any resistance or inductance in the trace between the device and the capacitor can cause a loss in efficiency
Note on speaker choice (4/8/16/32 ohm, etc.):
Using speakers more resistive than 4-Ω dramatically increases the thermal performance by reducing the output current and increasing the efficiency of the amplifier.
The audio input pins (INR+/- and INL+/-) must run side-by-side to maximize common-mode noise cancellation.
To control gain in SW set both pins to OFF and control the pin state in SW
TPA2012D2
2.1W/CH Audio Amp
U1
TS2012
PVDD
PVDD
AVDD
NC
NC
INL-
INL+
SHUTDOWN_L
OUTL-
OUTL+
INR-
INR+
SHUTDOWN_R
OUTR-
OUTR+
THERMAL
PGND
PGND
AGND
Gain
G0
G1
G0 G1 V/V dB
0 0 2 6
0 1 4 12
1 0 8 18
1 1 16 24
AVDD/PVDD: 2.5-5.5V
Temp: -40-85°C
VDD
AVDD
GND
AGND
C1 10µF
C2 0.1µF
C3 0.1µF
C4 0.1µF
C5 1µF
C6 1µF
C7 1µF
C8 1µF
R1 100K
R2 100K
R3 100K
R4 100K
FB1 ferrite
FB2 ferrite
J1
J2
SW1
JP1
IN_L_N
IN_L_P
IN_R_N
IN_R_P
SHUTDOWN_L
SHUTDOWN_R
OUT_L_N
OUT_L_P
OUT_R_N
OUT_R_P
GAIN0
GAIN1
ON
adafruit
TPA2012D2_StereoAmp_v0.4
not saved!
Sheet: 1/1
Drawing: >AUTHOR
Adafruit Industries

MD10C

Enhanced 10Amp
DC Motor Driver

User's Manual Rev2.0

V1.0

February 2013

Index

1. INTRODUCTION AND OVERVIEW

MD10C is an enhanced version of the MD10B which is designed to drive high current brushed DC motor up to 13A continuously. It offers several enhancements over the MD10B such as support for both locked-antiphase and sign-magnitude PWM signal as well as using full solid state components which result in faster response time and eliminate the wear and tear of the mechanical relay.

The MD10C has been designed with the capabilities and features of:

- Bi-directional control for 1 brushed DC motor.
- Support motor voltage ranges from 5V to 25V.
- Maximum current up to 13A continuous and 30A peak (10 second).
- 3.3V and 5V logic level input.
- Solid state components provide faster response time and eliminate the wear and tear of mechanical relay.
- Fully NMOS H-Bridge for better efficiency and no heat sink is required.
- Speed control PWM frequency up to 20KHz.
- Support both locked-antiphase and sign-magnitude PWM operation.
- **Dimension:**75mm x 43mm

2. PACKING LIST

Please check the parts and components according to the packing list. If there are any parts missing, please contact us at sales@cytron.com.my immediately.

1. 1 x MD10C Enhanced 10A DC Motor Driver
2. 1x 2510 PCB Connector - 3 Ways (Female)
3. 3 x 2510 Iron Pin
4. User's manual can be downloaded from http://www.cytron.com.my

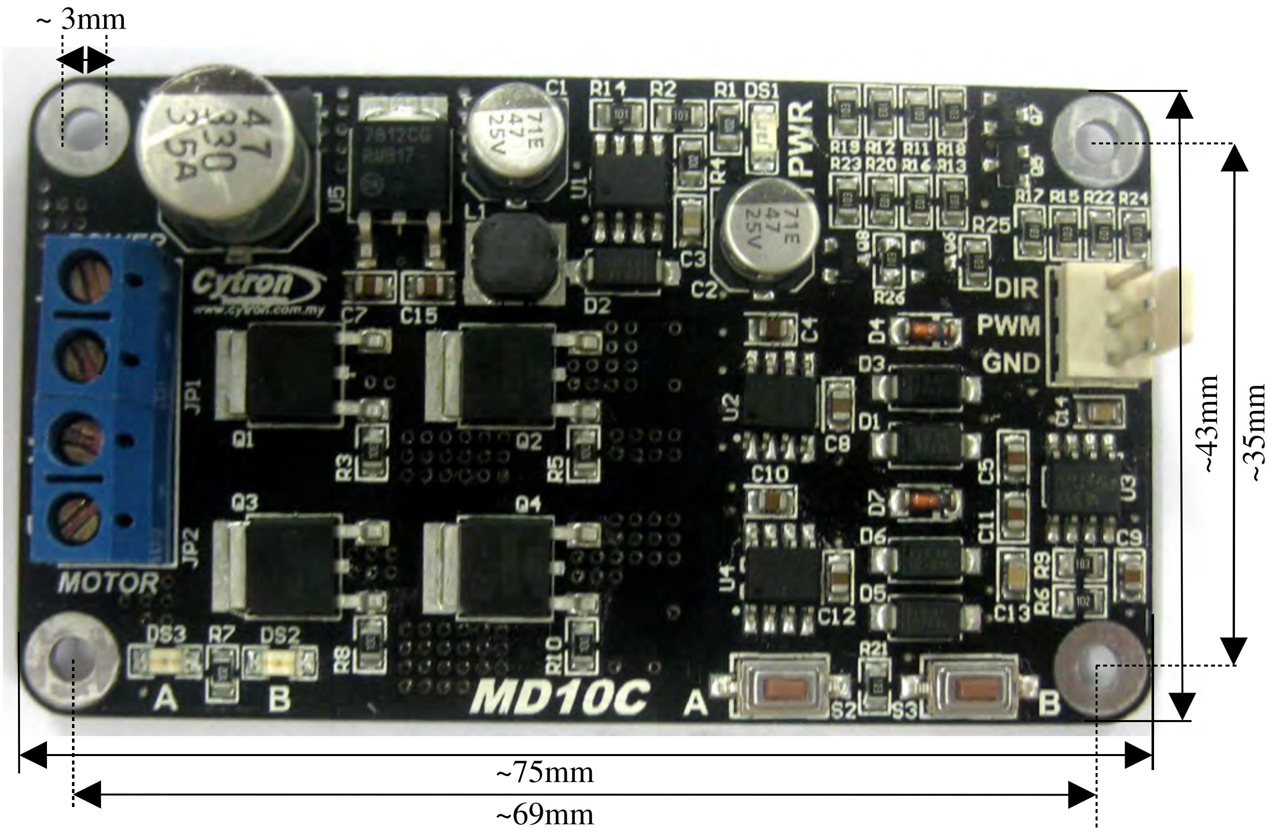
3. PRODUCT SPECIFICATION AND LIMITATIONS

Absolute Maximum Rating

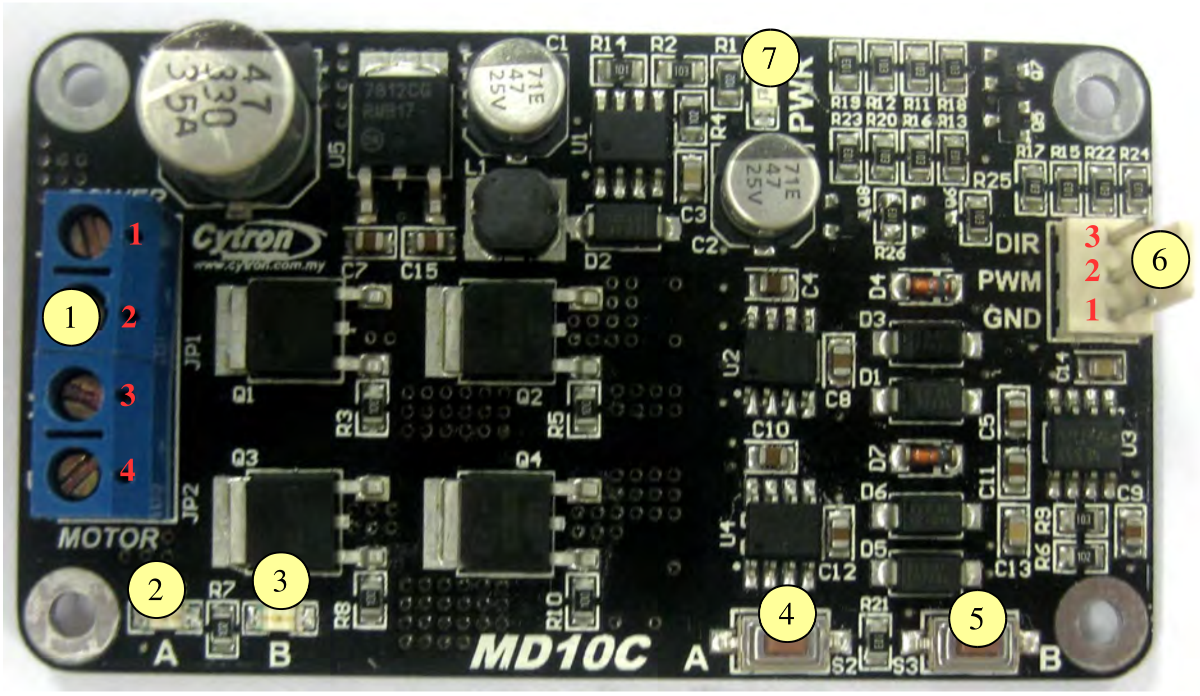
No	Parameters	Min	Typical	Max	Unit
1	Power Input Voltage	5	-	25	V
2	I_{MAX} (Maximum Continuous Motor Current)	-	-	13	A
3	I_{PEAK} – (Peak Motor Current) *	-	-	30	A
4	V_{IOH} (Logic Input – High Level)	3	-	5.5	V
5	V_{IOL} (Logic Input – Low Level)	0	0	0.5	V
6	Maximum PWM Frequency	-	-	20	KHz

**** Must not exceed 10 seconds.***

4. DIMENSION



5. BOARD LAYOUT



1. Terminal Block – Connect to motor and power source.

Pin No.	Pin Name	Description
1	POWER +	Positive supply.
2	POWER -	Negative supply.
3	Motor Output A	Connect to motor terminal A.
4	Motor Output B	Connect to motor terminal B.

2. Red LED A – Turns on when the output A is high and output B is low. Indicates the current flows from output A to B.

3. Red LED B – Turns on when the output A is low and output B is high. Indicates the current flows from output B to A.

4. Test Button A – When this button is pressed, current flows from output A to B and motor will turn CW (or CCW depending on the connection).

5. Test Button B – When this button is pressed, current flows from output B to A and motor will turn CCW (or CW depending on the connection).

6. Input

Pin No.	Pin Name	Description
1	GND	Logic ground.
2	PWM	PWM input for speed control.
3	DIR	Direction control.

The truth table for the control logic is as follow:

Pin 2 (PWM)	Pin 3 (DIR)	Output A	Output B
Low	X (Don't Care)	Low	Low
High	Low	High	Low
High	High	Low	High

7. Green LED – Power LED. Should be on when the board is powered on.

6. INSTALLATION AND GETTING STARTED

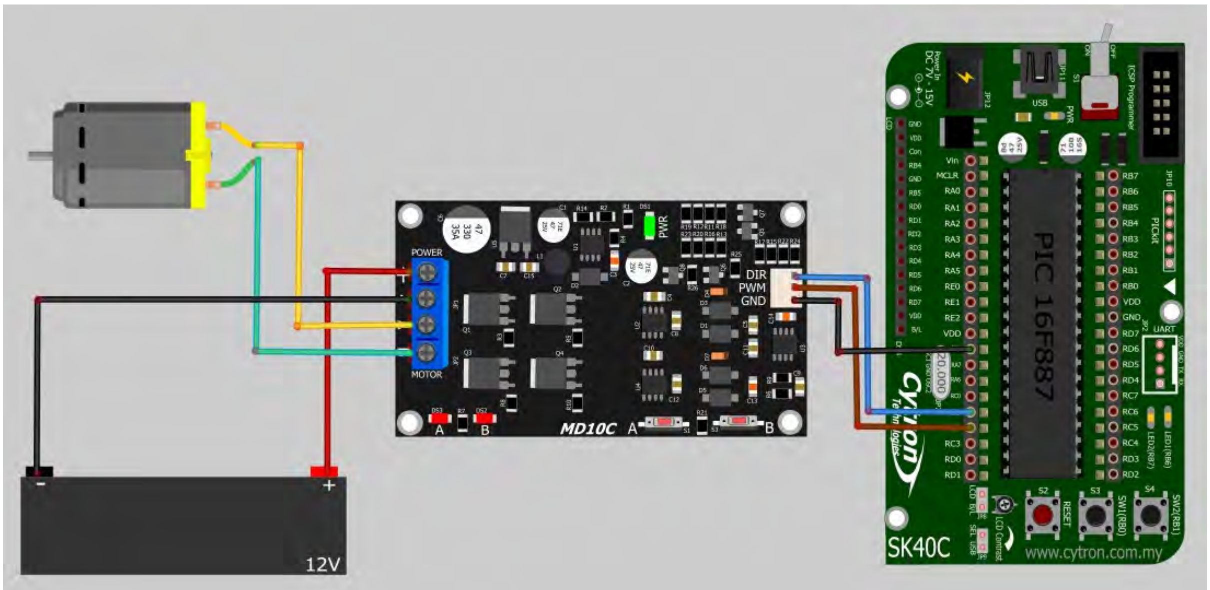
6.1 Getting Started MD10C with SK40C

MD10C is compatible with 2 types of PWM operation, which are:

1. Sign-Magnitude PWM – For sign-magnitude PWM operation, 2 control signals are used to control the speed and direction of the motor. PWM is feed to the PWM pin to control the speed while DIR pin is used to control the direction of the motor.

2. Locked-Antiphase PWM – For locked-antiphase PWM operation, only 1 control signal is needed to control the speed and direction of the motor. PWM pin is connected to logic high while the DIR pin is being feed with the PWM signal. When the PWM signal has 50% duty cycle, the motor stops running. If the PWM has less than 50% duty cycle, the motor will turn CW (or CCW depending on the connection). If the PWM signal has more than 50% duty cycle, motor will turn CCW (or CW depending on the connection).

Sample source code for using PIC16F877A to control the motor with MD10C is provided and is available for download at Cytron's website under the product page. SK40C is used in the demonstration and the connection diagram is as follow:



1. Connect MD10C and SK40C as shown in the schematic above and select the board supply for MD10C.

2. Upload the hex file into SK40C using UIC00A/B. The hex file can be downloaded from Cytron's website under MD10C Sample program. Please refer SK40C or UIC00B User's Manual to upload the hex code into SK40C.

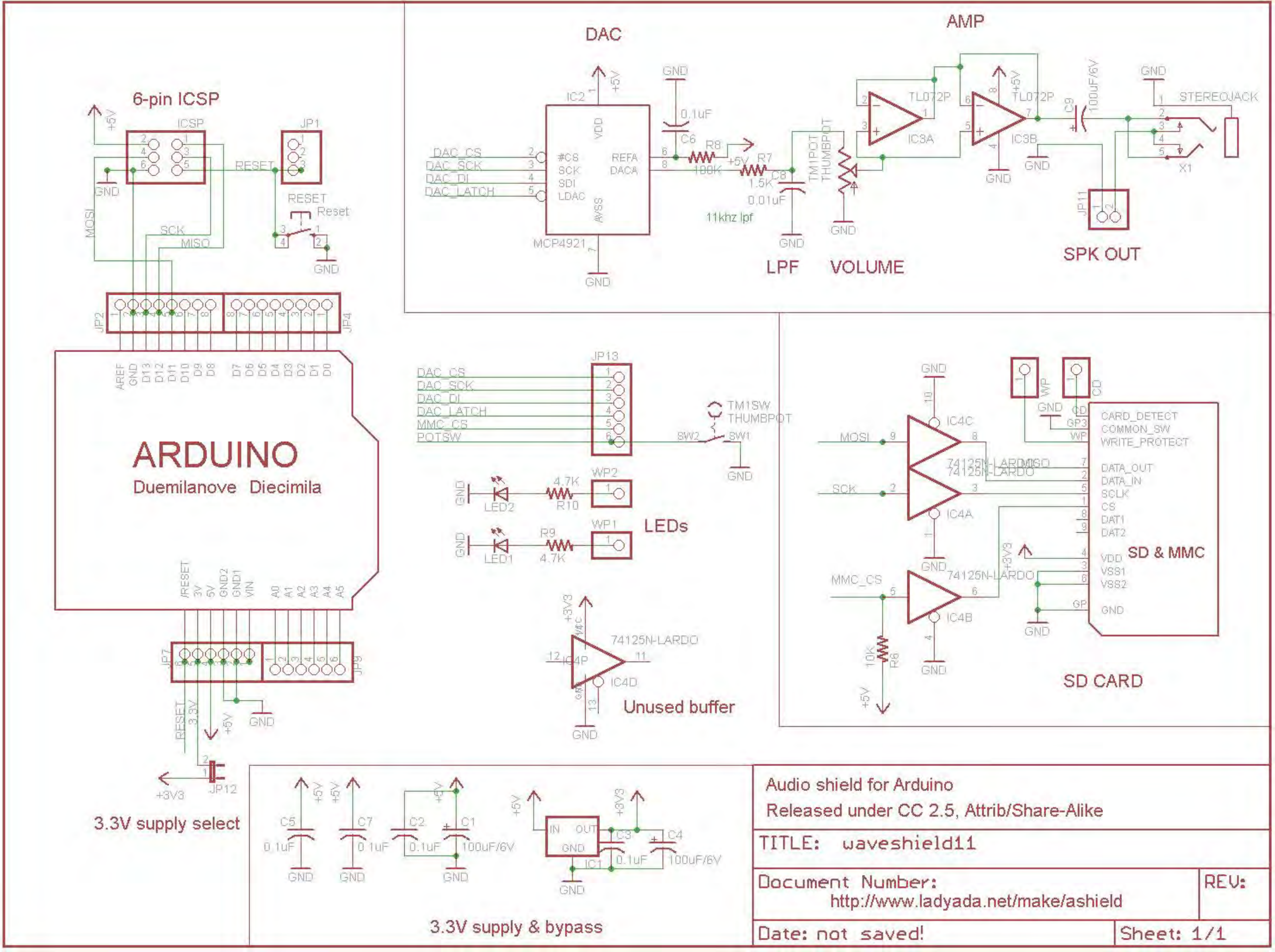
7. WARRANTY

- Product warranty is valid for 6 months.
- Warranty only applies to manufacturing defect.
- Damage caused by mis-use is not covered under warranty.
- Warranty does not cover freight cost for both ways.

Prepared by
Cytron Technologies Sdn. Bhd.
19, Jalan Kebudayaan 1A,
Taman Universiti,
81300 Skudai,
Johor, Malaysia.

Tel: +607-521 3178
Fax: +607-521 1861

URL: www.cytron.com.my
Email: support@cytron.com.my
sales@cytron.com.my



6-pin ICSP
ICSP
JP1
RESET
Reset
MOSI
SCK
MISO
GND
+5V
JP2
JP4
AREF
GND
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
ARDUINO
Duemilanove Diecimila
/RESET
3V
5V
GND2
GND1
VIN
A0
A1
A2
A3
A4
A5
JP7
JP9
RESET
3.3V
+5V
+3V3
JP12
3.3V supply select
DAC
IC2
MCP4921
VDD
#CS
SCK
SDI
LDAC
REFA
DACA
AVSS
DAC_CS
DAC_SCK
DAC_DI
DAC_LATCH
0.1uF
C6
R8
100K
R7
1.5K
C8
0.01uF
11khz lpf
LPF
TM1POT
THUMBPOT
VOLUME
AMP
TL072P
IC3A
IC3B
C9
100uF/6V
STEREOJACK
X1
JP11
SPK OUT
JP13
DAC_CS
DAC_SCK
DAC_DI
DAC_LATCH
MMC_CS
POTSW
TM1SW
THUMBPOT
SW2
SW1
WP2
4.7K
R10
LED2
WP1
R9
4.7K
LED1
LEDs
+3V3
74125N-LARDO
IC4P
IC4D
Unused buffer
IC4C
IC4A
IC4B
MOSI
SCK
MMC_CS
74125N-LARDMISO
74125N-LARDO
10K
R6
WP
CD
CARD_DETECT
COMMON_SW
WRITE_PROTECT
DATA_OUT
DATA_IN
SCLK
CS
DAT1
DAT2
VDD
VSS1
VSS2
GND
SD & MMC
SD CARD
C5
0.1uF
C7
C2
C1
100uF/6V
IN
OUT
IC1
C3
C4
3.3V supply & bypass
Audio shield for Arduino
Released under CC 2.5, Attrib/Share-Alike
TITLE: waveshield11
Document Number:
http://www.ladyada.net/make/ashield
REV:
Date: not saved!
Sheet: 1/1

XL-MaxSonar® - WR/WRC™ Series

High Resolution, IP67 Weather Resistant, Ultra Sonic Range Finder

MB7052, MB7060, MB7062, MB7066, MB7067, MB7068, MB7070, MB7072, MB7076, MB7077, MB7078, MB7092

The XL-MaxSonar-WR and XL-MaxSonar-WRC sensor series provide users with robust range information in air. These sensors also feature high-power acoustic output along with real-time auto calibration for changing conditions (supply voltage sag, acoustic noise, or electrical noise), operation with supply voltage from 3V to 5.5V, object detection from 0-cm to 765-cm (select models) or 1068-cm (select models), and sonar range information from 20-cm out to 765-cm (select models) or 1068-cm (select models) with 1-cm resolution. Objects from 0-cm to 20-cm range as 20-cm or closer. The sensor is housed in a robust PVC housing, designed to meet the IP67 water intrusion standard, and matches standard electrical/water ¾" PCV pipe fittings. The user interface formats included are pulse-width (select models), real-time analog-voltage envelope (select models), analog voltage output, and serial output.

Features

- Real-time auto calibration and noise rejection
- High acoustic power output
- Precise narrow beam
- Object detection includes zero range objects
- 3V to 5.5V supply with very low average current draw
- Free run operation can continually measure and output range information
- Triggered operation provides the range reading as desired
- All interfaces are active simultaneously
- RS232 Serial, 0 to Vcc, 9600 Baud, 81N
- Analog, (Vcc/1024) / cm for standard models
- Analog, (Vcc/1024) / 2cm for 10-meter models (MB7066, MB7076)
- Sensor operates at 42KHz

Benefits

- Acoustic and electrical noise resistance
- Reliable and stable range data
- Sensor dead zone virtually non-existent
- Robust, low cost IP67 standard sensor
- Narrow beam characteristics
- Very low power excellent for battery based systems
- Ranging can be triggered externally or internally
- Sensor reports the range reading directly, frees up user processor
- Easy hole mounting or mating with standard electrical fittings
- Filtering allows very reliable operation in most environments

Applications and Uses

- Tank level measurement
- Bin level measurement
- Proximity zone detection
- Environments with acoustic and electrical noise
- Distance measuring
- Long range object detection
- Industrial sensor
- -40°C to +65°C (limited operation to +85°C)

About Ultrasonic Sensors

Our ultrasonic sensors are desired for use in air, non-contact object detection and ranging sensors that detect objects within a defined area. These sensors are not affected by the color or other visual characteristics of the detected object. Ultrasonic sensors use high frequency sound to detect and localize objects in a variety of environments. Ultrasonic sensors measure the time of flight for sound that has been transmitted to and reflected back from nearby objects. Based upon the time of flight, the sensor then outputs a range reading.

XL-MaxSonar-WR/WRC Pin Out

Pin 1- Leave open (or high) for serial output on the Pin 5 output. When Pin 1 is held low the Pin 5 output sends a pulse (instead of serial data), suitable for low noise chaining.

Pin 2- This pin outputs a pulse-width representation of range. To calculate the distance, use a scale factor of 58uS per cm. (MB7052, MB7060, MB7062, MB7066, MB7067, MB7068)

This pin outputs the analog voltage envelope of the acoustic waveform. For the MB7070 series and MB7092 sensors, this is a real-time always-active output (MB7070, MB7072, MB7076, MB7077, MB7078, MB7092)

Pin 3- AN-This pin outputs analog voltage with a scaling factor of (Vcc/1024) per cm. A supply of 5V yields ~4.9mV/cm., and 3.3V yields ~3.2mV/cm. Hardware limits the maximum reported range on this output to ~700 cm at 5V and ~600 cm at 3.3V. The output is buffered and corresponds to the most recent range data.

For the 10-meter sensors (MB7066, MB7076) Pin 3 outputs an analog voltage with a scaling of (Vcc/1024) per 2cm. A supply of 5V yields ~4.9mV/2cm., and 3.3V yields ~3.2mV/2cm. This Analog Voltage output steps in 2cm increments.

Pin 4- RX- This pin is internally pulled high. If Pin-4 is left unconnected or held high, the sensor will continually measure the range. If Pin-4 is held low the sensor will stop ranging. Bring high 20uS or more to command a range reading.

Pin 5- TX- When Pin 1 is open or held high, the Pin 5 output delivers asynchronous serial data in an RS232 format, except the voltages are 0-Vcc. The output is an ASCII capital "R", followed by ASCII character digits representing the range in centimeters up to a maximum of 765 (select models) or 1068 (select models), followed by a carriage return (ASCII 13). The baud rate is 9600, 8 bits, no parity, with one stop bit. Although the voltages of 0V to Vcc are outside the RS232 standard, most RS232 devices have sufficient margin to read the 0V to Vcc serial data. If standard voltage level RS232 is desired, invert, and connect an RS232 converter such as a MAX232.When Pin 1 is held low, the Pin 5 output sends a single pulse, suitable for low noise chaining (no serial data).

V+ Operates on 3V - 5.5V. The average (and peak) current draw for 3.3V operation is 2.1mA (50mA peak) and 5V operation is 3.4mA (100mA peak) respectively. Peak current is used during sonar pulse transmit.

GND-Return for the DC power supply. GND (& V+) must be ripple and noise free for best operation.

Auto Calibration

Each time before the XL-MaxSonar-WR takes a range reading it auto calibrates. The sensor then uses this data to range objects. If the temperature, humidity, or applied voltage changes during sensor operation, the sensor will continue to function normally. (The sensors do not apply compensation for the speed of sound change verses temperature to any range readings.) If the application requires temperature compensation please look at the HRXL-MaxSonar-WR sensor line.

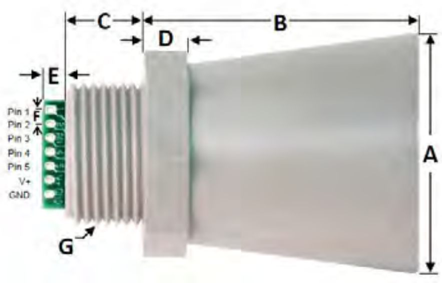
Supply Voltage Compensation

During power up, the XL-MaxSonar-WR sensor line will calibrate itself for the supply voltage. Additionally, the sensor will compensate if the supplied voltage gradually changes.

If the average voltage applied to the sensor changes faster than 0.5V per second, it is best to remove and reapply power to the sensor.

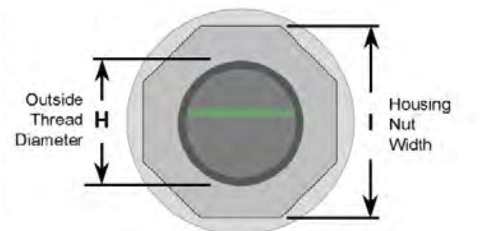
For best operation, the sensor requires noise free power. If the sensor is used with noise on the supplied power or ground, the accuracy of the readings may be affected. Typically, adding a 100uF capacitor at the sensor between the V+ and GND pins will correct most power related electrical noise issues.

XL-MaxSonar-WR Mechanical Dimensions

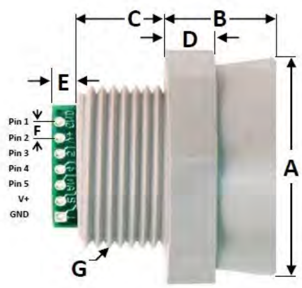


Values Are Nominal

A	1.72" dia.	43.8 mm dia.
B	2.00"	50.7 mm
C	0.58"	14.4 mm
D	0.31"	7.9 mm
E	0.23"	5.8 mm
F	0.1"	2.54 mm
G	3/4"-14 National Pipe Thread Straight	
H	1.032" dia.	26.2 mm dia.
I	1.37"	34.8 mm
Weight		50 grams

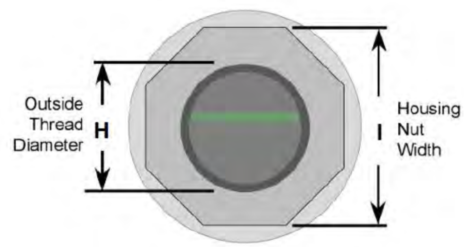


XL-MaxSonar-WRC Mechanical Dimensions



Values Are Nominal

A	34.7 mm dia.	1.37" dia.
B	17.9 mm	0.700"
C	14.4 mm	0.570"
D	7.9 mm	0.310"
E	5.8 mm	0.230"
F	2.54 mm	0.100"
G	3/4"-14 National Pipe Thread Straight	
H	26.2 mm dia.	1.032" dia.
I	34.8 mm	1.370"
Weight		32 grams



Range "0" Location

The XL-MaxSonar-WR and XL-MaxSonar-WRC reports the range to distant targets starting from the front of the transducer as shown in the diagram below.

The range is measured from the front of the transducer.

Base Sensor (MB7060 and MB7070)

The MB7060 and MB7070 are the base models of the XL-MaxSonar-WR sensor line. These sensors are recommended for general purpose usage. All other sensors in this series are based off these sensor models. The additional features are mentioned in their respective sections below.

XL-MaxSonar-WR1 (MB7062 and MB7072)

The XL-MaxSonar-WR1 sensors feature a 3 reading stability filter that ranges to the first detectable target. This filter requires that 3 consecutive range readings are within 10cm of each other to be considered a valid range reading. If the range readings are outside 10cm, the sensor discards the range reading set and reports the last valid range reading. This sensor does not view maximum range as a valid range, and will not report 765 when no target is detected. If this sensor does not detect a target for 1 hour, the sensor will go into fail-safe and report 000.

XL-MaxSonar-WRL (MB7066 and MB7076)

The XL-MaxSonar-WRL will report a maximum distance of 10 meters for large targets.

XL-MaxSonar-WRM (MB7052 and MB7092)

The XL-MaxSonar-WRM sensors are equipped with filtering firmware that allows the sensor to ignore smaller targets and noise, and still report the target that gives the largest acoustic reflection. This sensor will also reject infrequent and random noise, even if the noise has a higher amplitude than the acoustic return from the target. If the largest target is removed from the field of view, the XL-MaxSonar-WRM will switch to the target that gives the next largest detectable return.

The XL-MaxSonar-WR sensors were designed for applications where users are concerned with ranging the distance to flat targets (such as water and fuel tanks). This filtering algorithm stands in contrast to the other XL-MaxSonar-WR sensors that are designed to report the distance to the first detectable target.

In general, the XL-MaxSonar-WR will select the largest target from its field of view and report its range. Even so, objects up close may provide significantly greater returns over distant objects. Users are encouraged to test the sensor in their application to verify usability.

When targets are of similar amplitude reflections, preference is gen to the closest target.

XL-MaxSonar-WRC (MB7067 and MB7077)

The XL-MaxSonar-WRC sensors are the compact version of the MB7060 and MB7070. These sensors have a maximum detection range of 645 cm, and will report 765 cm when there are not targets detectable. If size is a concern in your application, you may want to consider the MB72XX sensors at http://www.maxbotix.com/WRUC.

XL-MaxSonar-WRC1 (MB7068 and MB7078)

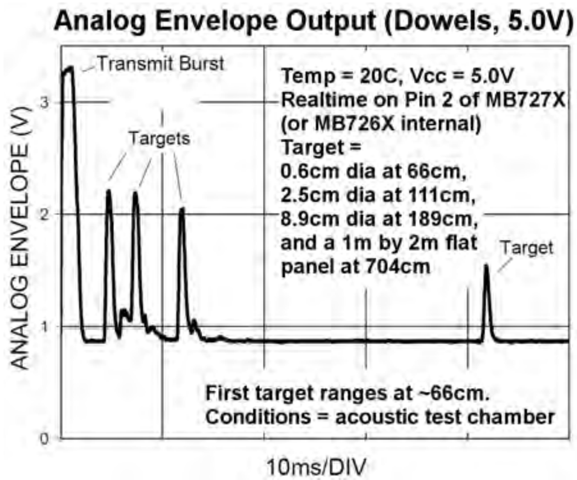
The XL-MaxSonar-WRC1 sensors are the compact version of the MB7062 and MB7072. These sensors have a maximum detection range of 645 cm. If size is a concern in your application, you may want to consider the MB72XX sensors at http://www.maxbotix.com/WRUC.

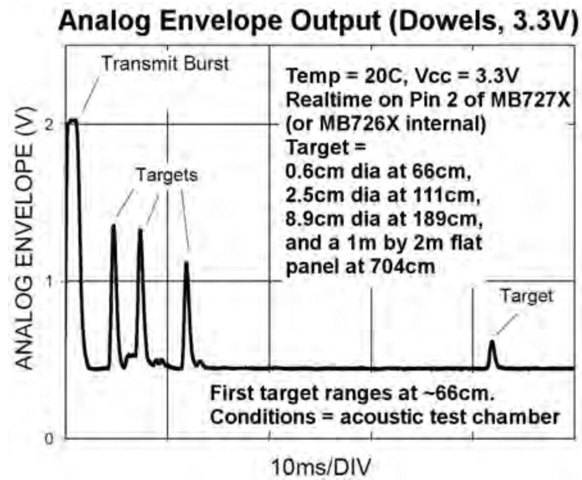
Sensor Minimum Distance - No Sensor Dead Zone

The XL-MaxSonar-WR sensors have a minimum reported distance of 20-cm (7.87 inches). However, the XL-MaxSonar-WR will range and report targets to the front sensor face. Large targets closer than 20-cm will typically range as 20-cm. For the XL-MaxSonar-WRC, objects between 3-cm and 20-cm will typically range as 20-cm.

Typical Performance to Target

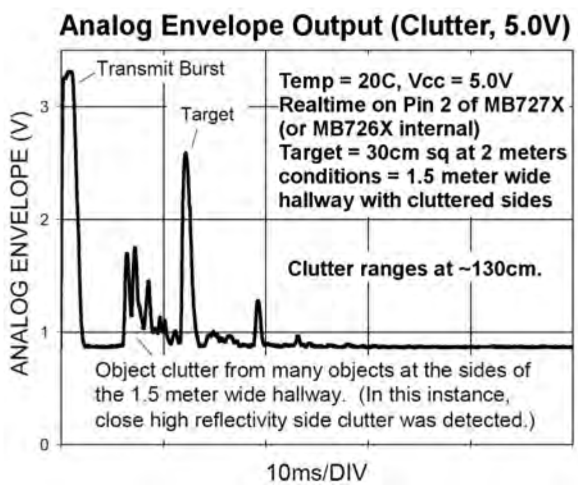
All sensor models

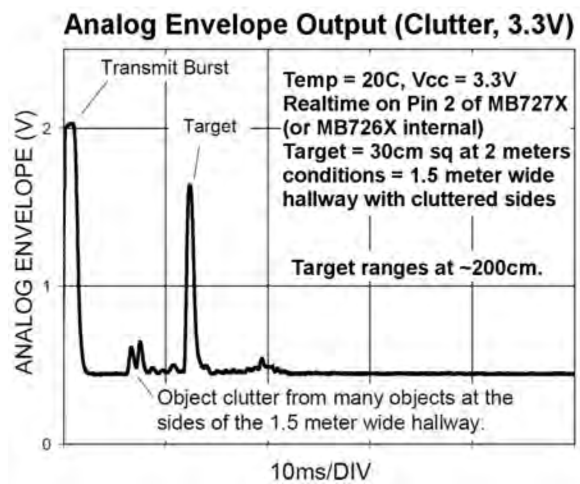




Typical Performance in Clutter

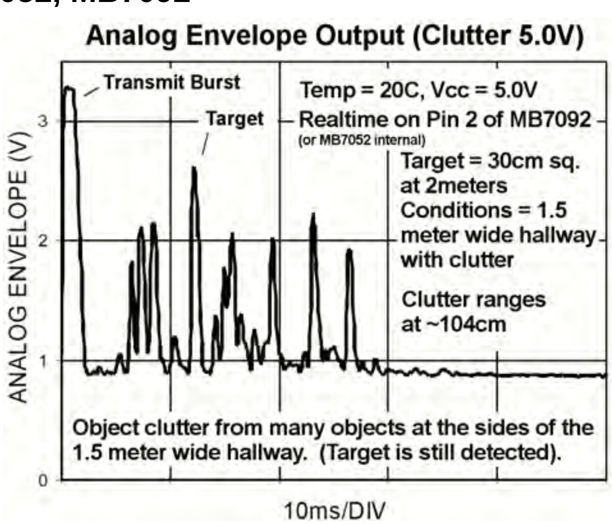
MB7060, MB7062, MB7076, MB7067, MB7068, MB7070, MB7072, MB7076, MB7077, MB7078

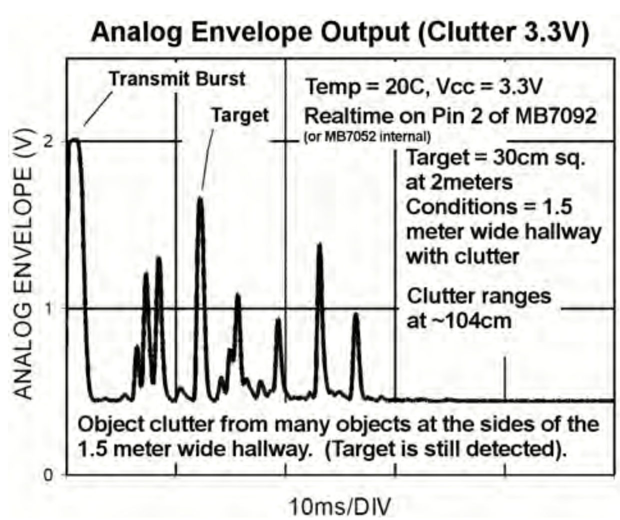




Typical Performance in Clutter

MB7052, MB7092





Device Comparison Chart

Part Number	AN Voltage	Serial Data (0 to Vcc level)	Pulse Width	Analog Envelope	Stability Filter	Most Likely Filter	Compact	7 meter range	10 meter range
MB7052	Yes	RS232	Yes		Yes	Yes		Yes	
MB7060	Yes	RS232	Yes					Yes	
MB7062	Yes	RS232	Yes		Yes			Yes	
MB7066	Yes	RS232	Yes						Yes
MB7067	Yes	RS232	Yes				Yes	Yes	
MB7068	Yes	RS232	Yes		Yes		Yes	Yes	
MB7070	Yes	RS232		Yes				Yes	
MB7072	Yes	RS232		Yes	Yes			Yes	
MB7076	Yes	RS232		Yes					Yes
MB7077	Yes	RS232		Yes			Yes	Yes	
MB7078	Yes	RS232		Yes	Yes		Yes	Yes	
MB7092	Yes	RS232		Yes	Yes	Yes		Yes	

Real-time Auto Calibration

The XL-MaxSonar-WR automatically calibrates prior to each range reading. The sensor then uses this data to range objects. If the temperature, humidity, or applied voltage changes during sensor operation, the sensor will continue to function normally. (The sensors do not apply compensation for the speed of sound change verses temperature to any range readings.) Detection has been characterized in the published sensor beam patterns.

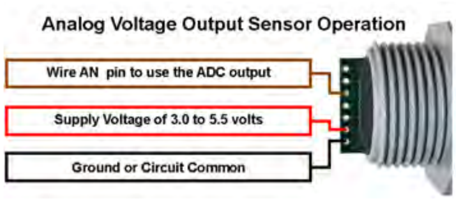
Real-time Noise Rejection

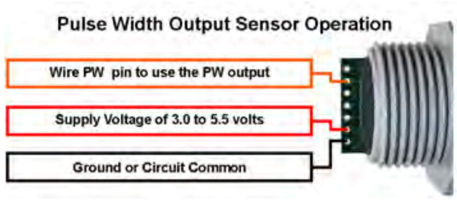
While the XL-MaxSonar-WR is designed to operate in the presence of noise, best operation is obtained when noise strength is low and desired signal strength is high. Hence, the user is encouraged to mount the sensor in such a way that minimizes outside acoustic noise pickup. In addition, keep the DC power to the sensor free of noise. This will let the sensor deal with noise issues outside of the users direct control (Even so, in general, the sensor will still function well even if these things are ignored). Users are encouraged to test the sensor in their application to verify usability.

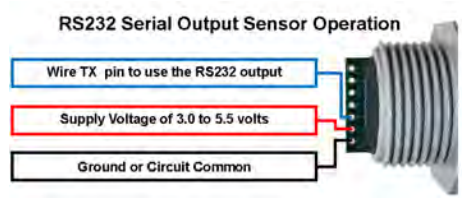
XL-MaxSonar-WR Sensor Operating Modes

Independent Sensor Operation

The XL-MaxSonar-WR sensors are designed to operate in a single sensor environment. Free-run is the default mode of operation for all of the MaxBotix Inc., sensors. The XL-MaxSonar-WR sensors have three separate outputs that update the range data simultaneously: Analog Voltage, Pulse Width[1], and RS232 Serial. Below are diagrams on how to connect the sensor for each of the three outputs. Note 1 - select models output an Analog Envelope for end user processing (MB707X sensors and MB7092)



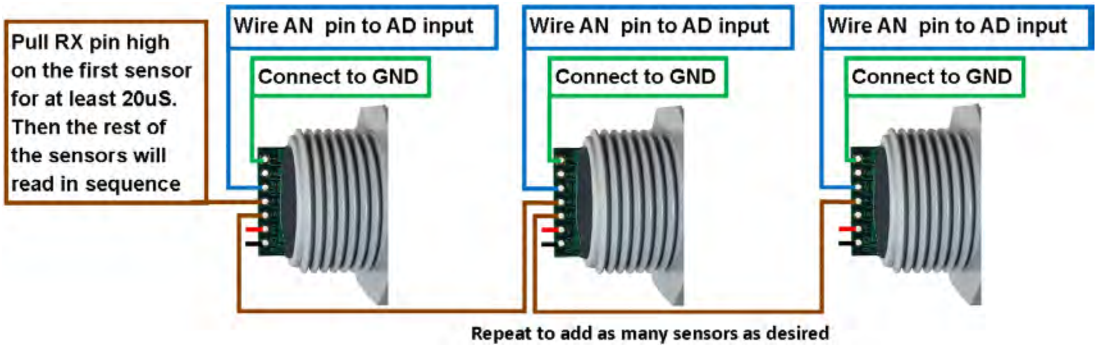




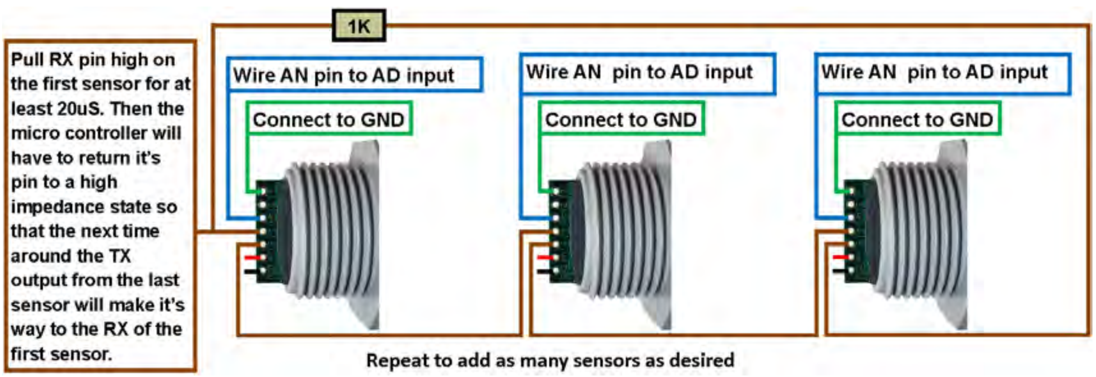
Using Multiple Sensors in a Single System

When using multiple ultrasonic sensors in a single system, there can be interference (cross-talk) from the other sensors. MaxBotix Inc., has engineered a solution to this problem for the XL-MaxSonar-WR sensors. The solution is referred to as chaining. We have 3 methods of chaining that work well to avoid the issue of cross-talk.

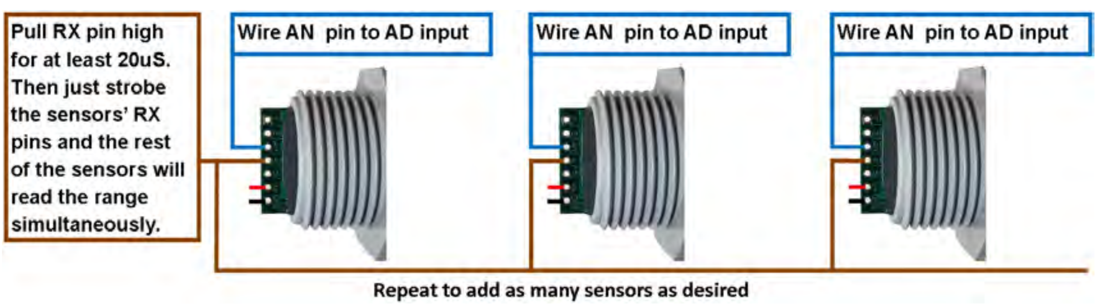
The first method is AN Output Commanded Loop. The first sensor will range, then trigger the next sensor to range and so on for all the sensors in the array. Once the last sensor has ranged, the array stops until the first sensor is triggered to range again. Below is a diagram on how to set this up.



The next method is AN Output Constantly Looping. The first sensor will range, then trigger the next sensor to range and so on for all the sensor in the array. Once the last sensor has ranged, it will trigger the first sensor in the array to range again and will continue this loop indefinitely. Below is a diagram on how to set this up.

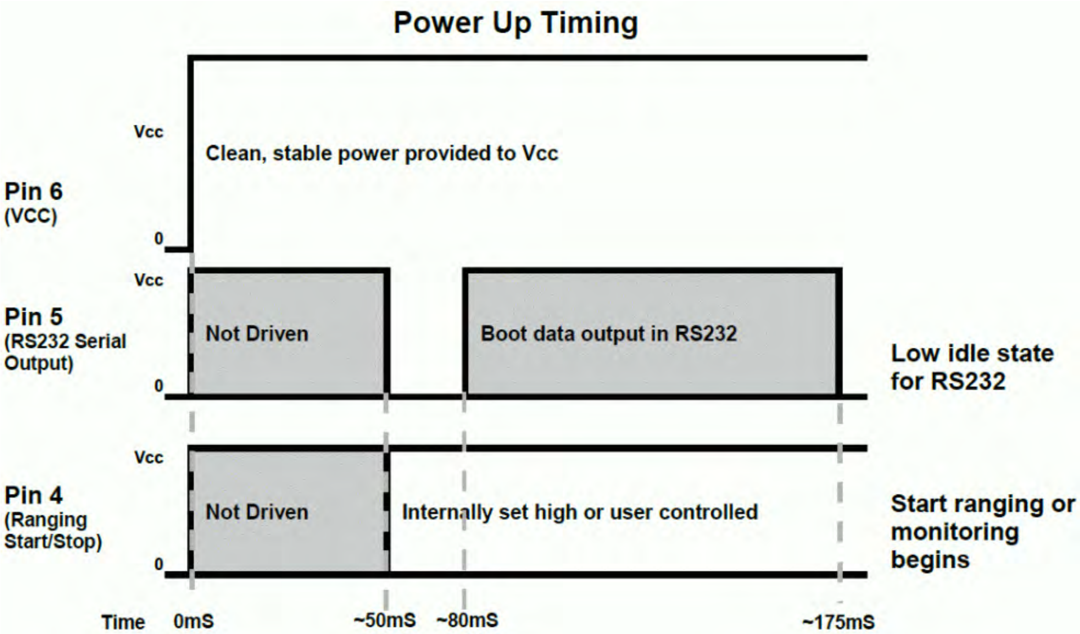


The final method is AN Output Simultaneous Operation. This method does not work in all applications and is sensitive to how the other sensors in the array are physically positioned in comparison to each other. Testing is recommend to verify this method will work for your application. All the sensors RX pins are connected together and triggered at the same time causing all the sensor to take a range reading at the same time. Once the range reading is complete, the sensors stop ranging until triggered next time. Below is a diagram on how to set this up.

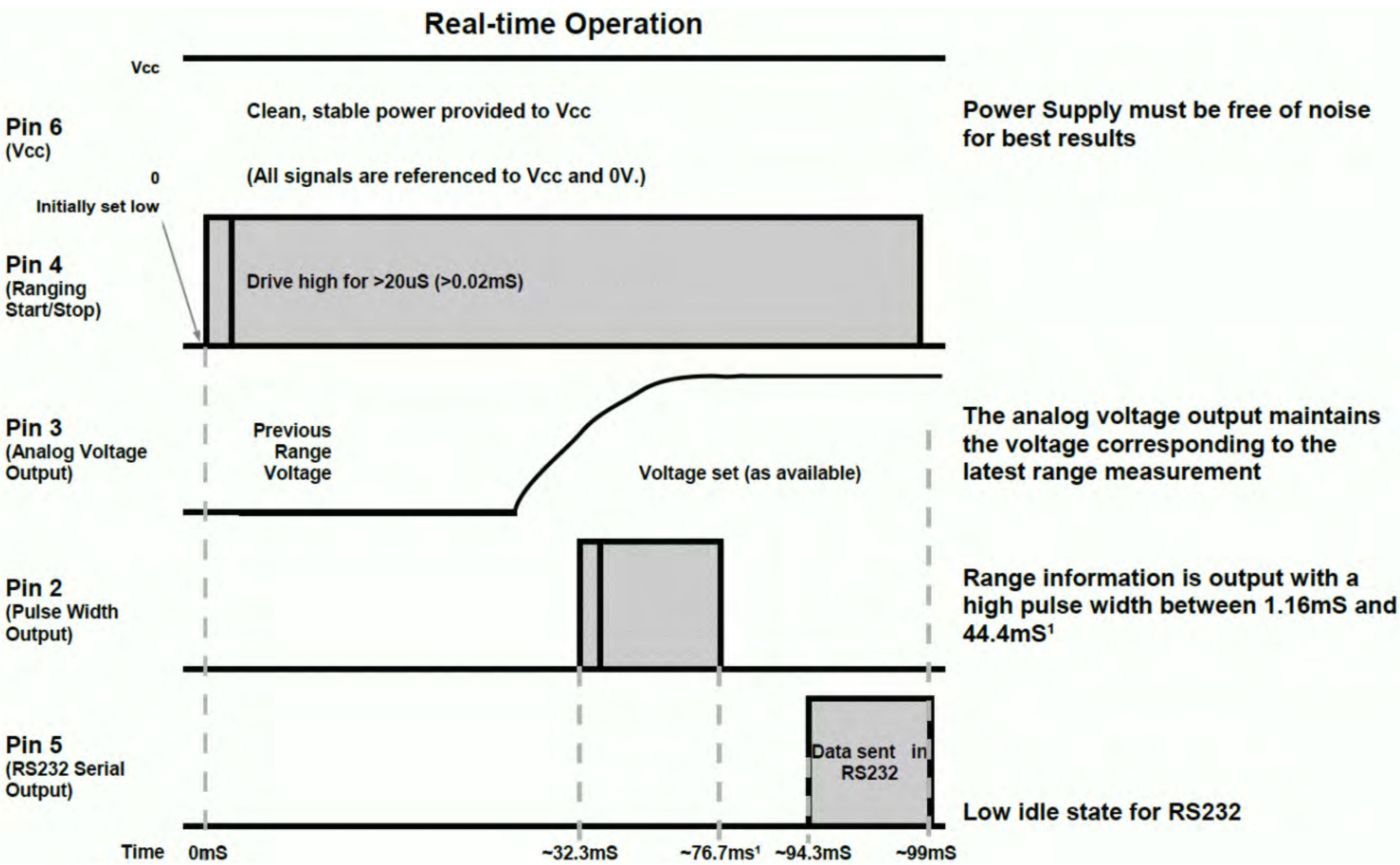


Sensor Timing Diagrams

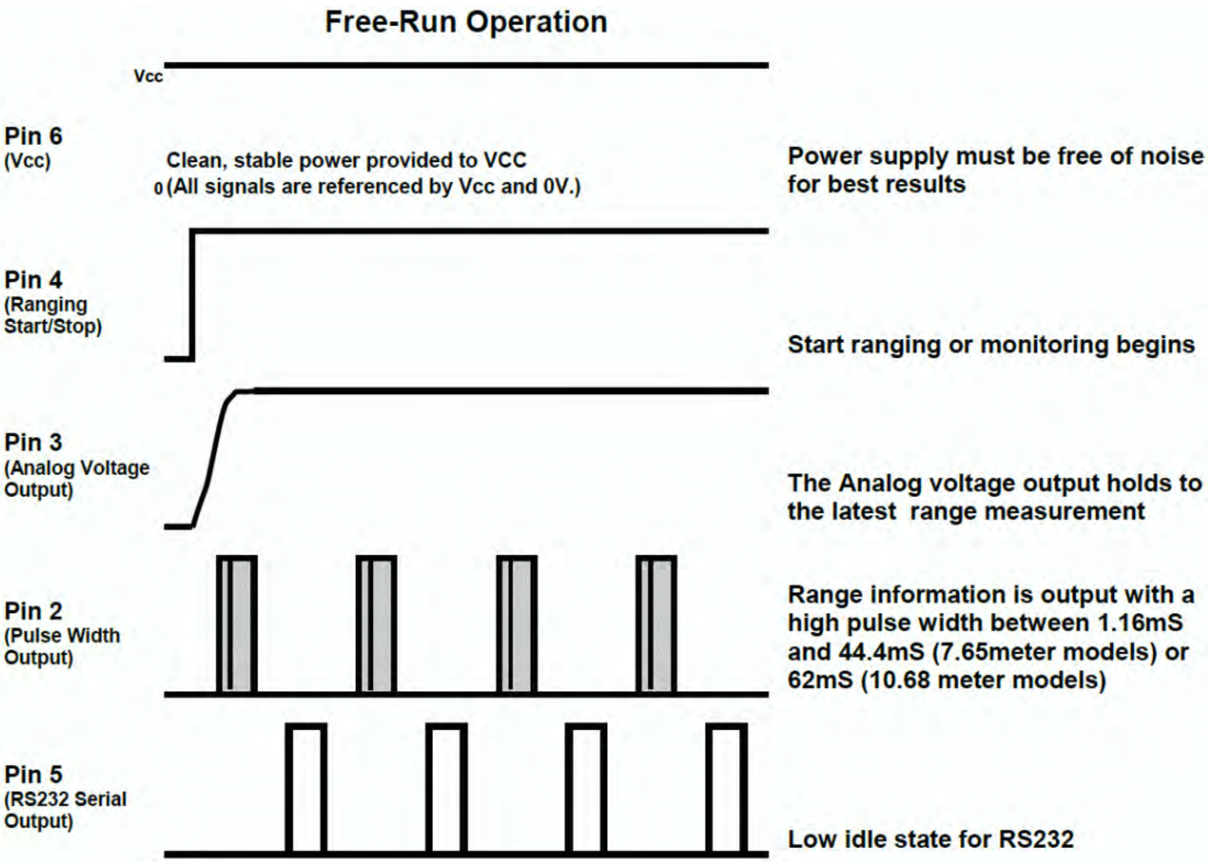
Power-Up Timing



Sensor Free-Run Timing



Real-Time Operation



XL-MaxSonar-WR Timing Description

175mS after power-up, the XL-MaxSonar-WR is ready to begin ranging. If Pin-4 is left open or held high (20uS or greater), the sensor will take a range reading. The XL-MaxSonar-WR checks Pin-4 at the end of every cycle. Range data can be acquired once every 99mS. Each 99mS period starts by Pin-4 being high or open, after which the XL-MaxSonar-WR calibrates and calculates for 20.5mS, and after which, thirteen 42KHz waves are sent. The sensor then determines the range to the target. Next the analog voltage is set. At 32.3mS, the sensors with a pulse width (PW), Pin 2 is set high for a length of time between 1.16mS and 44.4mS[1]. At 94.3mS, for the next 4.7mS the serial data is sent. The most accurate range output on the XL-MaxSonar-WR sensors is the PW output. Note 1: for 10-meter sensors, this time is 61.95mS

Then sensors with the analog envelope output (MB7070 series and MB7092), Pin-2 will show the real-time signal return information of the Analog Waveform.

Background Information Regarding our Beam Patterns

Each XL-MaxSonar-WR sensor has an individually calibrated beam pattern, and is matched to provide the approximate detection pattern shown in this datasheet. This allows end users to select the part number that matches their given sensing application. Each part number has a consistent field of detection so additional units of the same part number will have similar beam patterns. The beam plots are provided to help identify an estimated detection zone for an application based on the acoustic properties of a target versus the plotted beam patterns.

People Sensing: For users that desire to detect people, the detection area to the 1-inch diameter dowel, in general, represents the area that the sensor will reliably detect people.

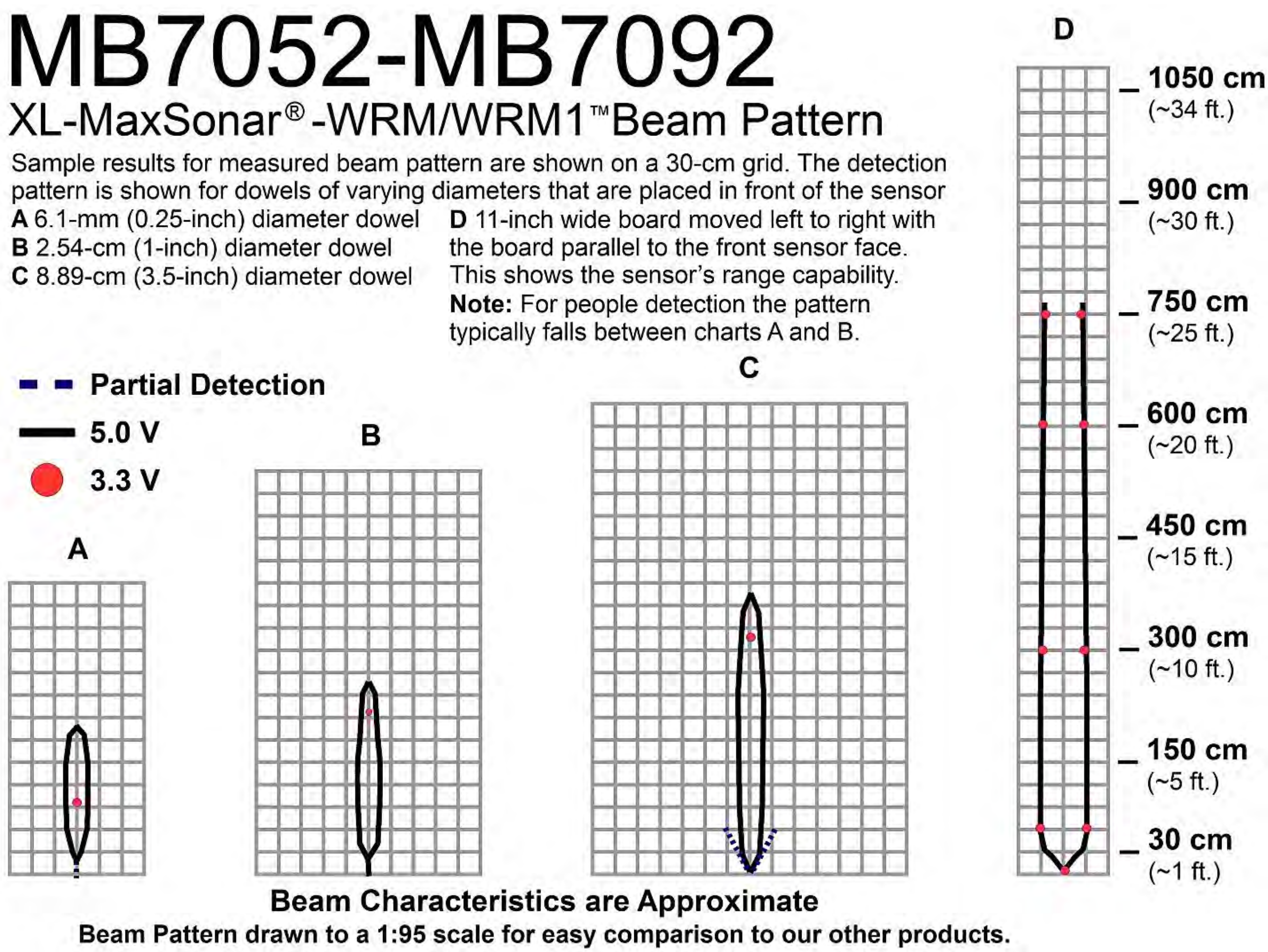
Each beam pattern is a 2D representation of the detection area of the sensor. The beam pattern is actually shaped like a 3D cone (having the same detection pattern both vertically and horizontally). Detection patterns for dowels are used to show the beam pattern of each sensor. Dowels are long cylindered targets of a given diameter. The dowels provide consistent target detection characteristics for a given size target which allows easy comparison of one MaxSonar sensor to another MaxSonar sensor.

For each part number, the four patterns (A, B, C, and D) represent the detection zone for a given target size. Each beam pattern shown is determined by the sensor's part number and target size.

The actual beam angle changes over the full range. Use the beam pattern for a specific target at any given distance to calculate the beam angle for that target at the specific distance. Generally, smaller targets are detected over a narrower beam angle and a shorter distance. Larger targets are detected over a wider beam angle and a longer range.

MB7052-MB7092 XL-MaxSonar®-WRM1/WRMA1™ Beam Pattern and Uses

The XL-MaxSonar-WRM1/WRMA1 ignores smaller targets and only reports the range to the largest acoustic return. The filtering in the MB7052 and MB7092 also rejects moving target clutter such as rain or snow, electrical noise, and outside acoustic noise.



MB7052-MB7092 Features and Benefits

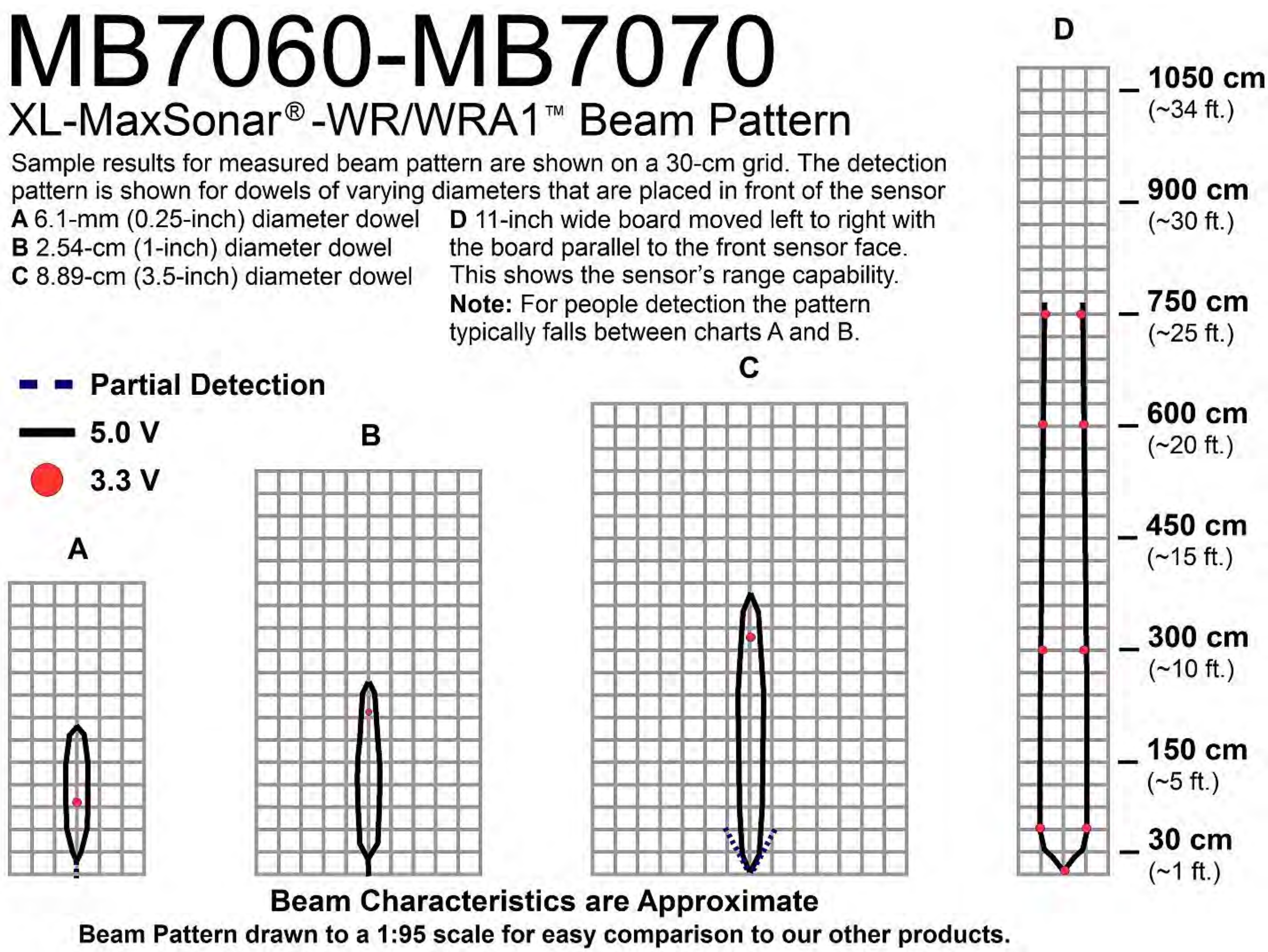
- Clutter rejection provides range to the largest amplitude reflection within the field of view
- Real-time calibration, noise rejection and additional filtering provides stable range information
- Excellent for ranging to large objects in the presence of cluttered or noisy environments
- Excellent for applications that require consistently accurate outputs
- Impressive acoustic and electrical noise resistance
- 10Hz refresh rate for the MB7092
- 7.5Hz refresh rate on the MB7052

MB7052-MB7092 Applications and Uses

- Autonomous Navigation
- Environments with acoustic and electrical noise
- Bin Level Measurement
- Tank Level Measurement

MB7060-MB7070 XL-MaxSonar®-WR/WRA1™ Beam Pattern and Uses

The XL-MaxSonar-WR/WRA1 reports the range to the first detectable target. The MB7060 and MB7070 sensors are the most recommended XL-MaxSonar-WR sensor. This is a good starting place when unsure of which XL-MaxSonar-WR to use.



MB7060-MB7070 Features and Benefits

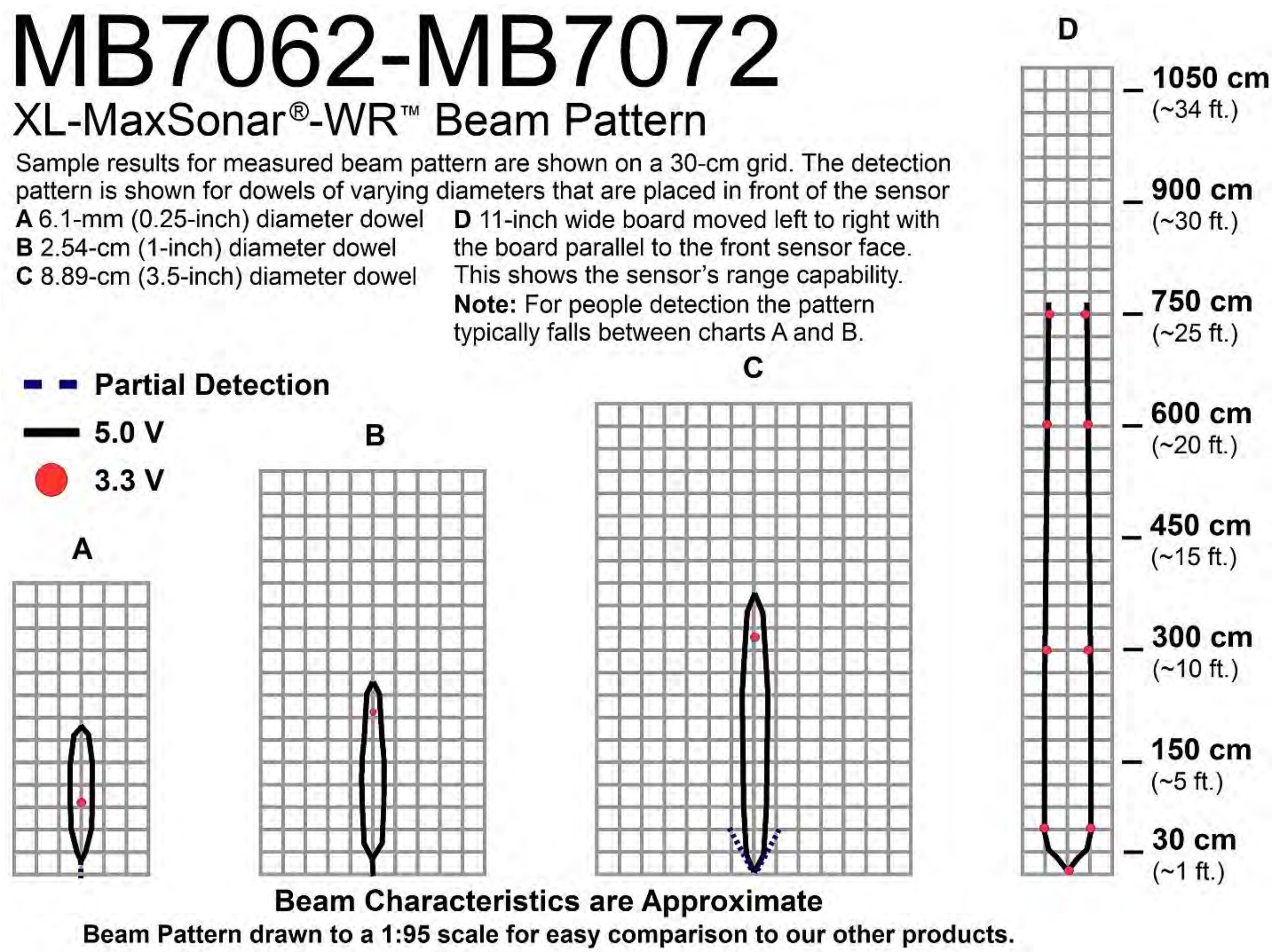
- Real-time calibration, and noise rejection for every ranging cycle
- Readings can occur up to every 100mS (10Hz)
- Analog voltage (Vcc/1024) / cm
- Precise narrow beam
- Continuously variable gain

MB7060-MB7070 Applications and Uses

- Applications where a stability filter is not needed or desired
- Multi-Sensor Arrays
- Distance Measuring
- People Detection

MB7062-MB7072 XL-MaxSonar®-WR/WRA™ Beam Pattern and Uses

The XL-MaxSonar-WR/WRA sensors have a 3 reading stability filter in the firmware. This sensor is well suited for applications requiring stable, accurate range readings. This sensor ranges to the first detectable target.



MB7062-MB7072 Features and Benefits

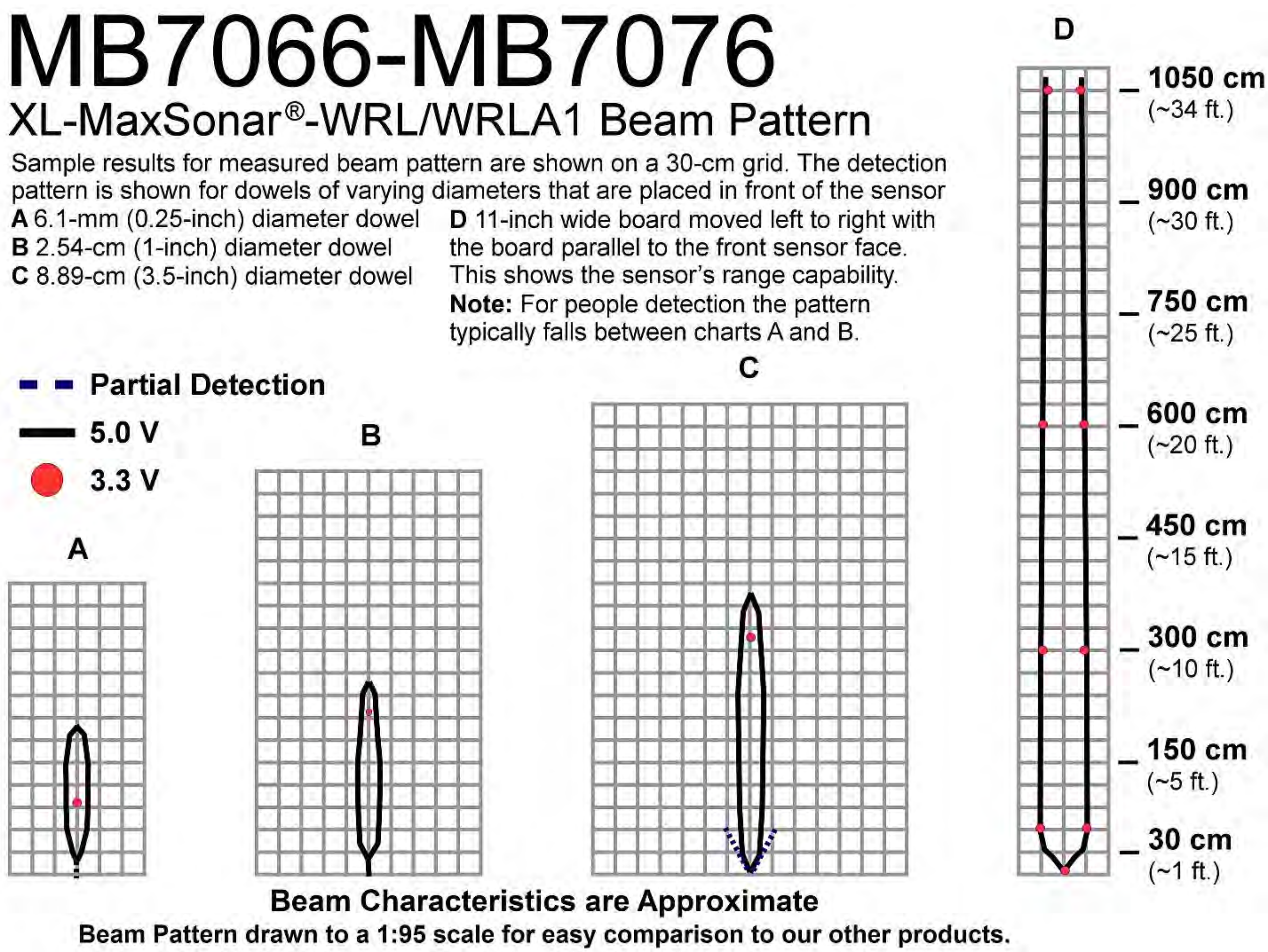
- 1 hour fail-safe built into sensor
- Real-time calibration, noise rejection and additional filtering provides stable range information
- Excellent for ranging to large objects in the presence of cluttered or noisy environments
- Excellent for applications that require consistently accurate outputs
- Advanced acoustic and electrical noise filtered output
- Reports filtered output on serial and analog-voltage outputs
- Reliable stable range data
- No power up calibration is required

MB7062-MB7072 Applications and Uses

- Long range object detection
- Industrial sensor
- Drop in upgrade for MB7060 and MB7070

MB7066-MB7076 XL-MaxSonar®-WRL/WRLA1™ Beam Pattern and Uses

The XL-MaxSonar-WRL/WRLA1 ranges objects from 0-cm to 1068-cm (35 feet) and provides range information from 20-cm to 1068-cm with a 1-cm resolution. This sensor is designed for applications where large object detection is needed to 10 meters.



MB7066-MB7076 Features and Benefits

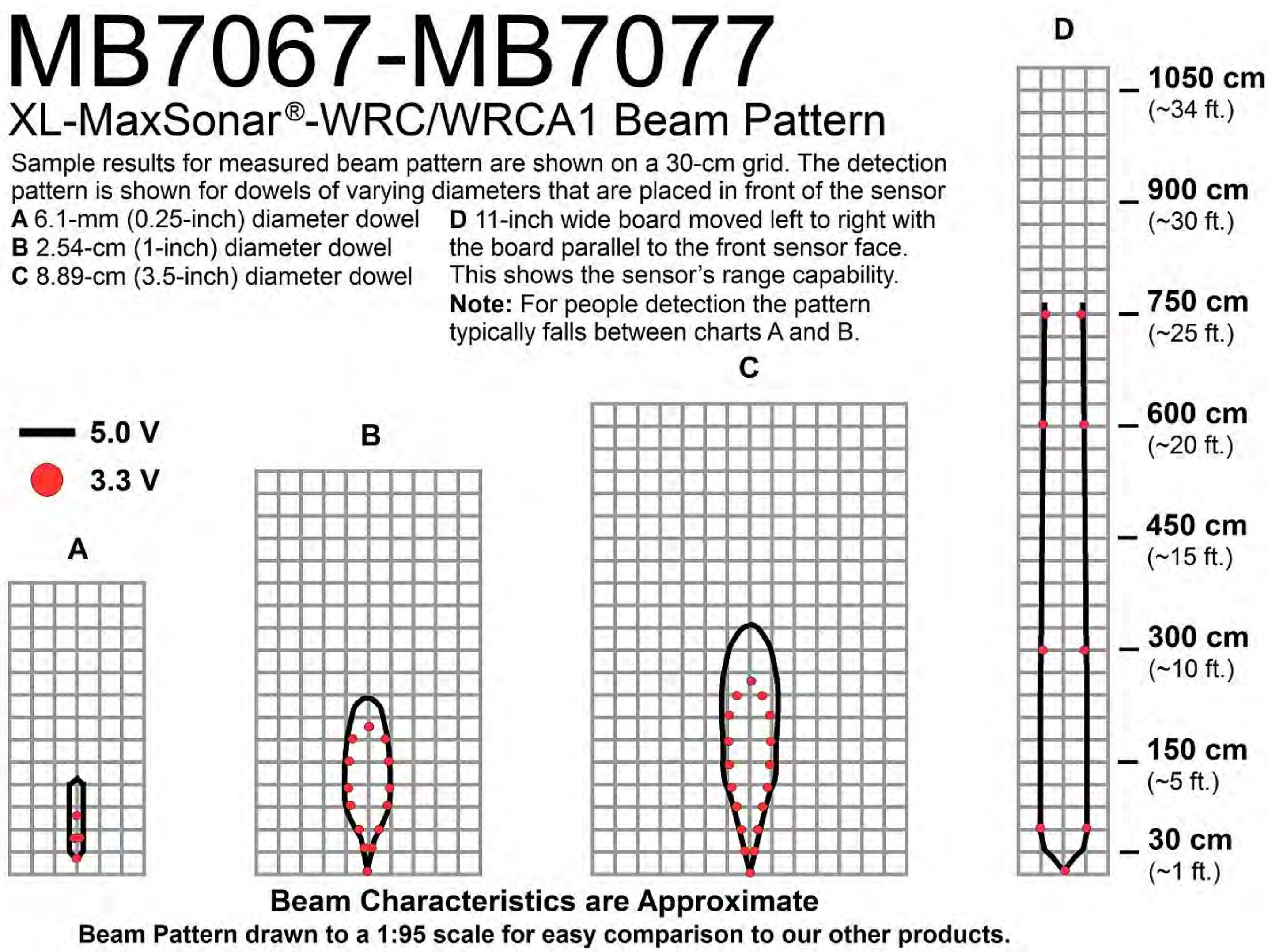
- Extended 10 meter range detection and outputs
- High acoustic power output
- Readings can occur up to every 100ms, 10-Hz rate
- Triggered operation provides the range reading as desired
- Fast measurement cycle
- Quality narrow beam characteristics
- Low cost, long range IP67 sensor

MB7066-MB7076 Applications and Uses

- Robot ranging sensor
- Autonomous navigation
- Distance measuring
- Long range object detection
- Industrial sensor

MB7067-MB7077 XL-MaxSonar®-WRC/WRCA1™ Beam Pattern and Uses

The XL-MaxSonar-WRC/WRCA1 is the compact version of the MB7060 and MB7070. The MB7067 and MB7077 sensors are idea for applications where weight and size are restricted.



MB7067-MB7077 Features and Benefits

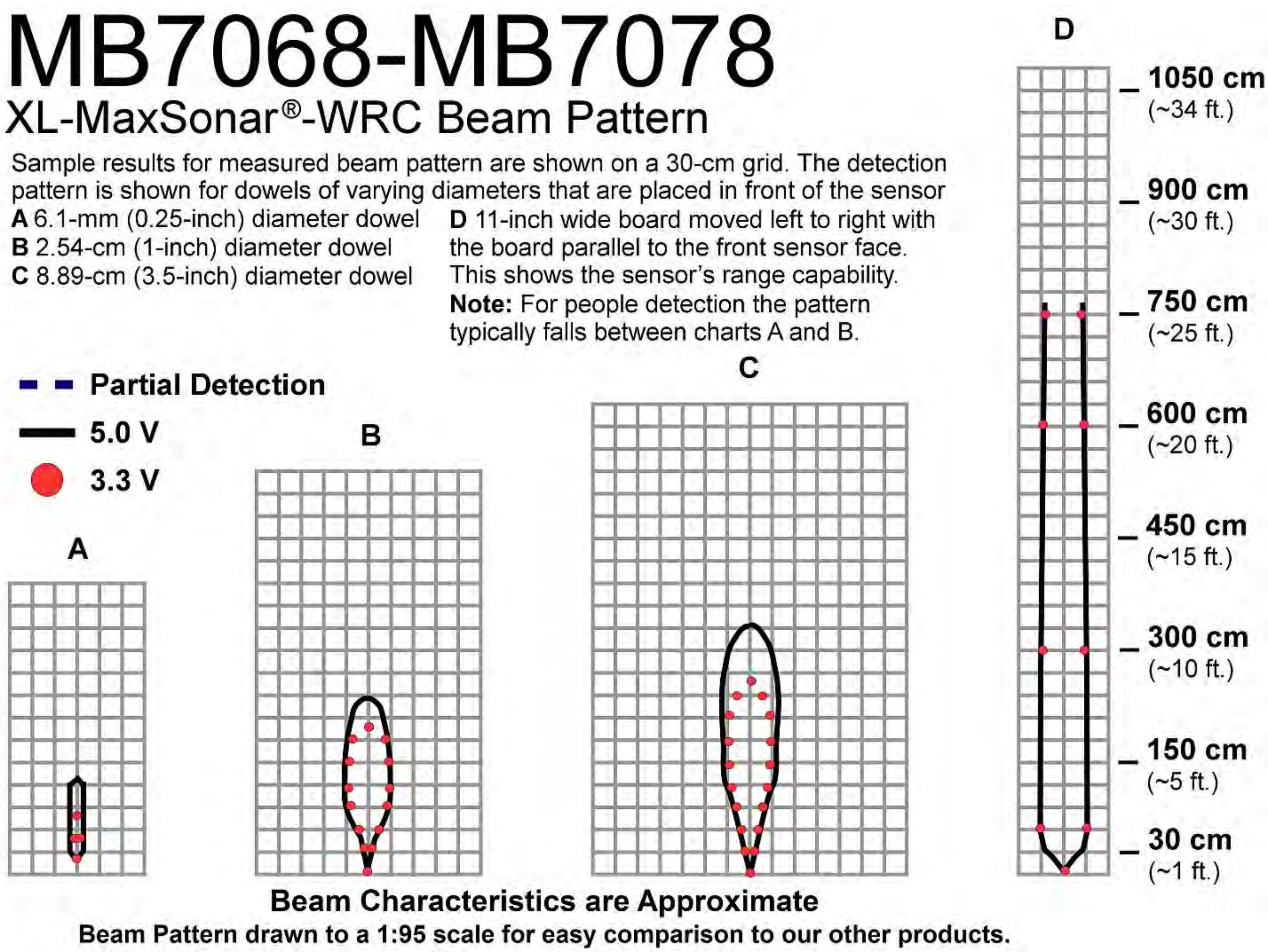
- Extra compact housing
- Designed for outdoor or indoor environments
- Lightweight, compact, weather resistant design
- Sensor dead zone is virtually gone
- Low cost IP67 sensor

MB7067-MB7077 Applications and Uses

- UAV blimps
- Bin level measurement
- Proximity zone detection
- Robot ranging sensor

MB7068-MB7078 XL-MaxSonar®-WRC/WRCA1™ Beam Pattern and Uses

The XL-MaxSonar-WRC/WRCA1 is the compact version of the MB7062 and MB7072. This sensor is well suited for applications where the MB7062 and 7072 are too heavy or too large.



MB7068-MB7078
Features and Benefits

- 1 hour fail-safe built into sensor
- Extra compact housing
- Excellent for applications that require consistently accurate outputs
- Impressive acoustic and electrical noise resistance
- Lightweight, compact, weather resistant design

MB7068-MB7078
Applications and Uses

- Environments with acoustic and electrical noise
- Bin Level Measurement
- Tank Level Measurement

Have the right MaxSonar for your application?

Check out our MaxSonar Product Lines

Indoor Use
(or protected environments)

1 mm Resolution
HRLV-MaxSonar-EZ

1 in Resolution
LV-MaxSonar-EZ
LV-ProxSonar-EZ

1 cm Resolution
XL-MaxSonar-EZ
XL-MaxSonar-AE
XL-MaxSonar-EZL
XL-MaxSonar-AEL
I2CXL-MaxSonar-EZ

1 mm Resolution
HRUSB-MaxSonar-EZ

1in Resolution
USB-ProxSonar-EZ

Outdoor Use
(or rugged environments) IP67

1 mm Resolution
HRXL-MaxSonar-WR
HRXL-MaxSonar-WRS
HRXL-MaxSonar-WRT
HRXL-MaxSonar-WRM
HRXL-MaxSonar-WRMT
HRXL-MaxSonar-WRL
HRXL-MaxSonar-WRLT
HRXL-MaxSonar-WRLS
HRXL-MaxSonar-WRLST

1 cm Resolution
XL-MaxSonar-WR
XL-MaxSonar-WRL
XL-MaxSonar-WRA
XL-MaxSonar-WRLA
I2CXL-MaxSonar-WR

F-Option Available for all WR models. For additional protection when necessary in hazardous

1 mm Resolution
HRXL-MaxSonar-WRC
HRXL-MaxSonar-WRCT

1 cm Resolution
XL-MaxSonar-WRC
XL-MaxSonar-WRCA
I2CXL-MaxSonar-WRC

1 cm Resolution
UCXL-MaxSonar-WR
UCXL-MaxSonar-WRC

Accessories-More information available online

MB7954 - Shielded Cable

The MaxSonar Connection Wire is used to reduce interference caused by electrical noise on the lines. This cable is a great solution to use when running the sensors at a long distance or in an area with a lot of EMI and electrical noise.

MB7950 - XL-MaxSonar-WR Mounting Hardware

The MB7950 Mounting Hardware is selected for use with our outdoor ultrasonic sensors. The mounting hardware includes a steel lock nut and two O-ring (Buna-N and Neoprene) each optimal for different applications.

MB7955 / MB7956 / MB7957 / MB7958 / MB7972 - HR-MaxTemp

The HR-MaxTemp is an optional accessory for the HR-MaxSonar. The HR-MaxTemp connects to the HR-MaxSonar for automatic temperature compensation without self heating.

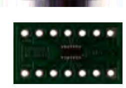
MB7961 - Power Supply Filter

The power supply filter is recommended for applications with unclean power or electrical noise.

MB7962 / MB7963 / MB7964 / MB7965 - Micro-B USB Connection Cable

The MB7962, MB7963, MB7964, and MB7965 Micro-B USB cables are USB2.0 compliant and backwards compatible with USB 1.0 standards. Varying lengths.

MB7973 CE Compliance Widget



The MB7973 adds protection for the CE requirement for Lightning/Surge IEC61000-4-5