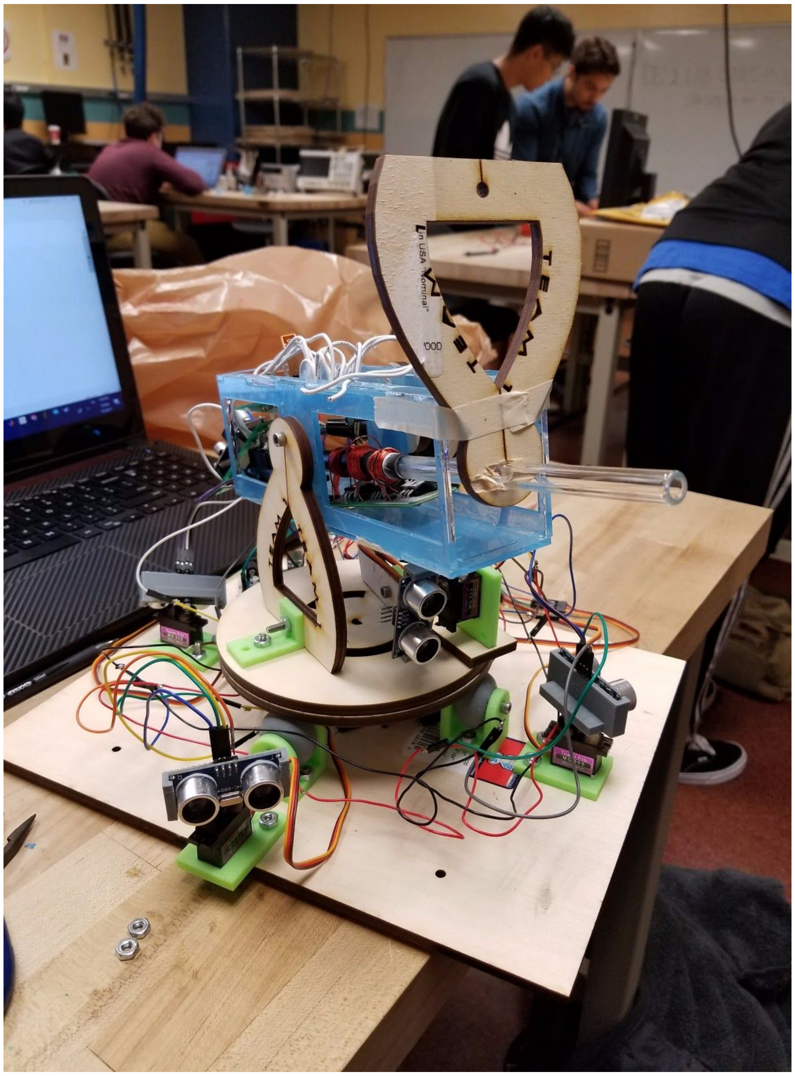
ElectroMagnetic Cannon



DEVELOPERS: Qiyuan Wu, Alberto Avalos, Edwin Molina, Jingpei Lu, Brenden Diep

ADVISOR: Professor Truong Nguyen, ECE 196

REFERENCES: Refer to the end of the report

LAST UPDATED:12/09/17

TABLE OF CONTENTS:

REQUIRED DOWNLOADS / INSTALLATION:

- Latest version of Arduino -Arduino
- SolidWorks -Solidworks
- Inkscape -Inkscape

A NOTE FROM THE DEVELOPERS:

Dear Students,
Our team would like to introduce to you all the ElectroMagnetic Cannon! This project will teach you a variety of skills as you will need to implement various distinct capabilities in order to complete the successful construction of this device. We would like to give a huge thanks to the TAs of not only our section but of the following section as well for helping us out through the many obstacles we've encountered, to Phuong for organizing the course and putting in the work to do everything she could to help us achieve our goals, and to professor Nguyen for making the class itself possible and for helping us acquire the necessary funding for our projects.

PROJECT DESCRIPTION:

In this project we'll be using our knowledge with circuit building and laser cutting/3D printing in order to construct an electromagnetic cannon using coils, transistors, diodes, and programmed by Arduino. It would be able to launch ferromagnetic objects, so we will choose small lightweight material for safety. We would also determine power of the cannon firing by choosing the number of acceleration stages, to avoid any hazard. The power to the cannon is determined by the electromagnetic force acting on the projectile over the length of the barrel through the stages. Our goal is to demonstrate on a small-scale model a functional cannon/ "coilgun" which has real worldwide use and application. Our added creative design is on the implementation with a rotating and stable base structure to add aiming feature and mechanical design. The steps to completing this project include, programming the arduino to implement the logic of our project and fire the electromagnetic cannon, laser cutting, 3D printing in order to build the structure of the cannon, soldering, and circuitry working in order to wire everything.

We would integrate ultrasonic sensors to detect nearby objects and once a threshold is reached, the arduino will send a signal to a servo. Once the ultrasonic sensor detects an object the goal would be to rotate the cannon to face the object and fire to hit the object of interest. The end goal would be incorporate the electromagnetic cannon using a mount on a rotating disk. This rotating disk will move via servo motor that will be programmed by arduino to rotate the cannon. Once this design is successful the next step would be to implement the aiming feature incorporated with accelerometer and gyro built into arduino.

PROJECT ORIGINALITY:

Our project incorporates the electromagnetic cannon and the robotic disk, each of the components rather ordinary but combining them together renders innovation. Its originality constitutes the fact that it combines parts from other projects we have seen, for example, servo motion from the robotic arm, ultrasonic sensors which are used in the slackbot, and electromagnetism which is used in a variety of projects. Rather than focusing on just one of these mechanics, our project combines all of them and works to synchronize all of their functions so that they could work in unison in order to achieve the desired result. This is very useful mainly due to the fact that that it teaches us and anyone who decides to replicate the project in the future how to incorporate different working components into a project in order to attain one main function/goal. Also, no one has ever constructed an automatic rail gun in UCSD for security purposes.

OVERALL LEARNING OBJECTIVES:

- How to design a electromagnetic cannon circuit with enough driving voltage as well as satisfactory stability
- How to communicate between 2 arduinos
- How to implement coil wrapping, circuit wiring, soldering, and model design of a stable base structure
- How to use arduino to move cannon using ultrasonic sensor, control movement, and detection
- How to learn the function of a various different components and how to incorporate the distinct functions to work together
- Improvement of 3D designing and printing skills
- Improvement of laser cutting skills
- How to implement the codes of different individual parts into a single, efficient code, and the debugging of such (improving coding skills)
- Confidence in circuit building skills

REQUIRED PROJECT PARTS (ALL):

Note: some items when bought have more than necessary but unavoidable

Arduino nano	$4.00	

Blank PCB boards	$7.00	
Red Enameled copper wire coil	$8.76	The coil
Driver chips (UCC37322P)	~$4.00/each	Use 1 for each stage (note: highly recommended to have extra in case some are ruined)
Mosfets (IRF3205)	$4.00/10pack	1 for each stage
High Voltage diodes IN4007	$7.00/100pack	1 for each stage
250V/15A glass fuses + fuse clips	$7.00/pack	(recommended to have extra fuses, must at least be 250V/8A+ minimum but may burn)
LM2596 DC-DC module power supply	$5.50	Used to power components
90C 1300mah turnigy Lipo battery	$19.00	To plug in the battery need XT60 male+female
XT60 connector	N/A	Needed to attach battery, may not come with battery purchase
Glass straw (for barrel)	$3.00	We used a glass smoothie straw.
1 Toggle Switch + 1 Button	N/A	Flip switch and button to control circuit and give firing signal
Male+female header pins		For components

8-pin dip sockets	$6.30	Used for UCC37322 chips on PCB
projectiles	N/A	We borrowed small cylindrical metallic projectiles (can also use a screw)
Pcb copper wire	$6.00	Used to solder and create wire connections
Ultrasonic sensors	$8.75	
Motor Driver - Dual TB6612FNG	$6.00	

Project Build Steps

Part #1: Networking the components (Fredo)

Parts Required

- 5 Ultrasonic Sensors
- Jumper wires
- Breadboard
- Arduino
- Motor
- Motor Driver - Dual TB6612FNG

1. **Testing the Individual Parts**
    a. Test the Ultrasonic sensors to make sure they are working properly and accurately and to acquire knowledge regarding how the sensors work in the process - A useful video explaining how the ultrasonic sensors work and how to code them can be found here; Refer to figure 1 to see the circuitry for the ultrasonic sensors (NOTE: The Echo pins can be connected to the arduino's analog pins to achieve the same goal as if they were connected to the PWM pins, doing this can help keep the wiring more organized)
    b. Now test the motor to make sure it is in working order, you can find out how to use the motor here, note that the diagram and tutorial in posted website is for using 2 motors with the driver, this however can easily be simplified to use just

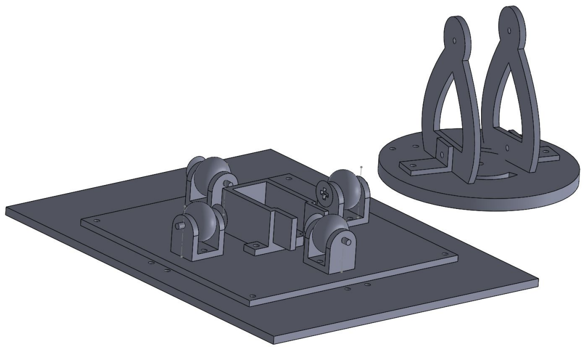
one motor by reducing the code to that of just one motor and connecting the single motor to the outputs indicated in the code. Also, the diagram shows that you need to connect an external power source to the chip to power the 2 motors, but because we will simply be using one motor the arduino can produce enough energy to power the single motor, therefore you would have to connect the arduino's 5V pin in the place where the diagram tells you to connect the + wire from an external power source.

2. **<u>Communicating the components with one another</u>**
    a. Now use your knowledge regarding how the ultrasonic sensors and motor work and get the 2 different components, for this you will need to use all 5 sensors and the motor, to better learn how to get the sensors to communicate with the motor you may want to start off by using one sensor and getting it to actuate the motor whenever it detects any distance less than a threshold distance, for example less than 10cm
    b. Now implement the concept but with all 5 ultrasonic sensors and the motor, code the components in such a way so that if any of the first 4 sensors detects anything under the threshold distance, then it will trigger the motor to begin rotating, and then while the motor is rotating the 5th sensor will begin scanning, (this sensor will be placed on the rotating disk with the cannon so it will be scanning as the cannon rotates), and the cannon will only stop rotating once the 5th sensor detects something within the threshold distance, when the 5th sensor detects something it is to have a 3s delay or so, (in order to fire the cannon), before rotating back to its original position. This is done to improve the accuracy of the cannon and in order to cancel out any false alarms. If the 5th detector doesn't detect anything, then the motor is to stop rotating after 5 seconds, or after it's done one full revolution, (you may have to adjust your time window differently for that), and then it is to return to its original position by rotating in the opposite direction than it was initially rotating for the same amount of time that it spent rotating in its initial direction. This is done to untangle any wires that begin wrapping around the cannon's base as it rotates.
    c. **OPTIONAL:** The 4 servos that will be the initial detectors will be placed around the base of the cannon 90 degrees from each other. So what you may do is have the cannon rotate clockwise if either of the 2 sensors to its right detect anything, and rotate counter-clockwise if either of the 2 sensors to its left detect anything. Although not entirely necessary, this is more efficient than having the cannon do a full 360 degree clockwise rotation to detect a target that is to its left or vice versa.
    d. The final code should look something like this although it may vary.

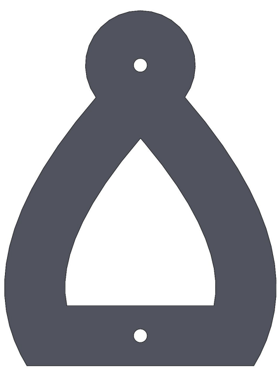
Part #2: Designing and assembling the cannon base & framework **(Edwin)**

Parts Required

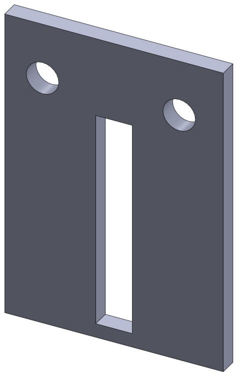
- Solidworks
- Computer
- Mouse (recommended)



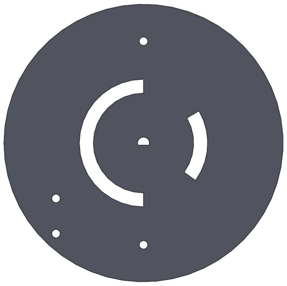
There is no one specific method to construct the individual parts since there is multiple ways to create the same part:



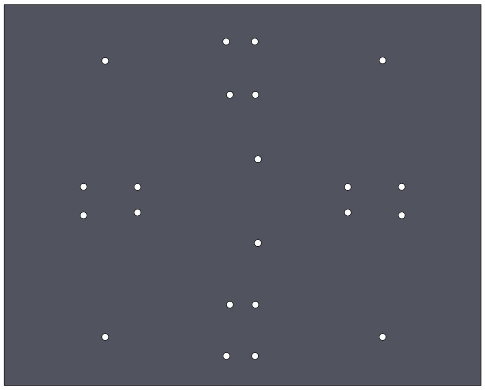
Gun Mount: Need to use the spline tool to create the curvatures as shown. Create circles and trim any unnecessary lines that may interfere with the extrude feature. Then use the offset tool to create a parallel spline. Trim any unnecessary lines and lastly extrude. Diameter of circle for bolts are 4.5mm.



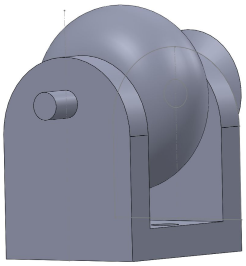
Ultrasonic base: Use the square and circle tools and extrude. Diameter of circle for bolts are 4.5mm. Diameter of circle for bolts are 4.5mm.



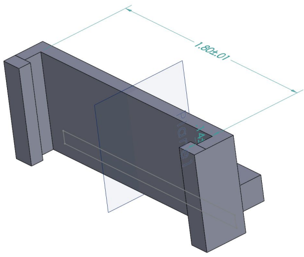
Disk: Use the circle tool and the trim tool to design the disk as shown above. Diameter of circle for bolts are 4.5mm.



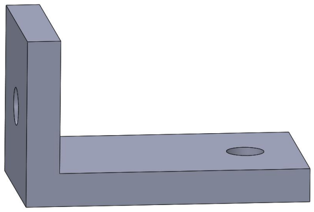
Disk: Use the circle tool to design the base as shown above. Note that this design may vary depending on how the caster, driver and ultrasonic mounts are designed.Diameter of circle for bolts are 4.5mm.



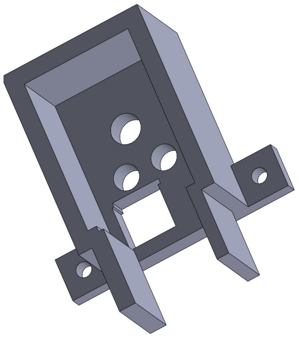
Caster: Use the circle tool and the trim tool to design the disk as shown above.Diameter of circle for bolts are 4.5mm.



Ultrasonic Head: Used to hold on to the ultrasonic sensor. Use the rectangle, line and extrude tool to design as shown above. Diameter of circle for bolts are 4.5mm.



Gun mount support: Use the rectangle, line and extrude tool to design as shown above. Diameter of circle for bolts are 4.5mm.



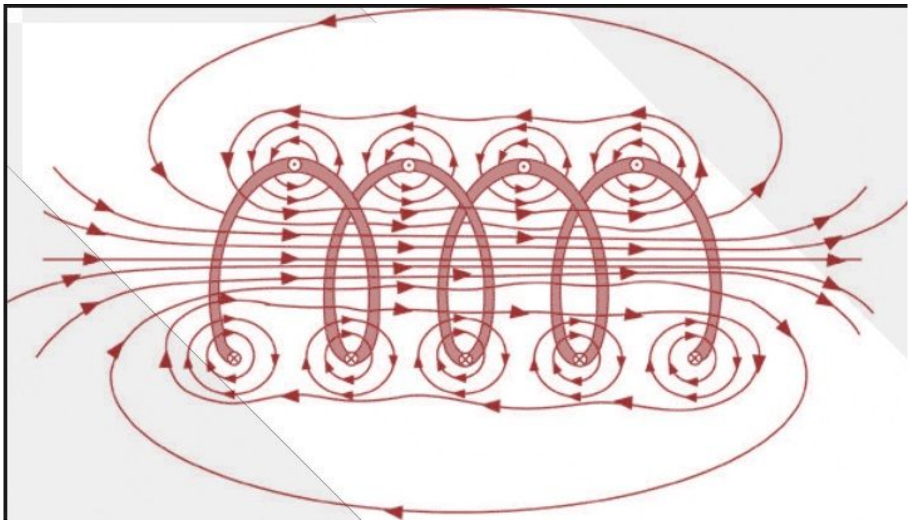
Driver Mount: Use the circle tool and the trim tool to design the mount shown above. An important thought is to measure the motor carefully and all its asymmetric parts to avoid tilting. Diameter of circle for bolts are 4.5mm.

Part #3: Building the cannon **(Brenden, Qiyuan, Jingpei)**

Parts Required

- All the components in project materials except for sensors or motors
- Highly recommended to have extra UCC37322 drivers + IRF3205 mosfet + 250V/15A glass fuses
- Soldering tools

Concept Refresh

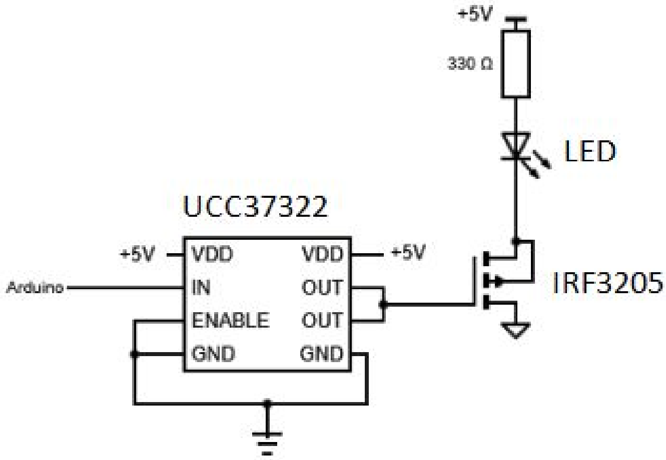


A voltage differential applied to coil segment causes moving current and magnetic field, AKA solenoid.

(Brenden)

1. **Initial circuit planning**

Before showing the coil circuit diagram, we present the prototype circuit that will be built on the breadboard. First **instead of coil** use a **resistor + LED** for testing functionality on breadboard. With this setup, the LED will turn on when arduino sends an input high signal, so the LED will represent successful current flow in the “coil”. Use the arduino to send a high signal to input pin of UCC37322 and you should see the LED turn on.



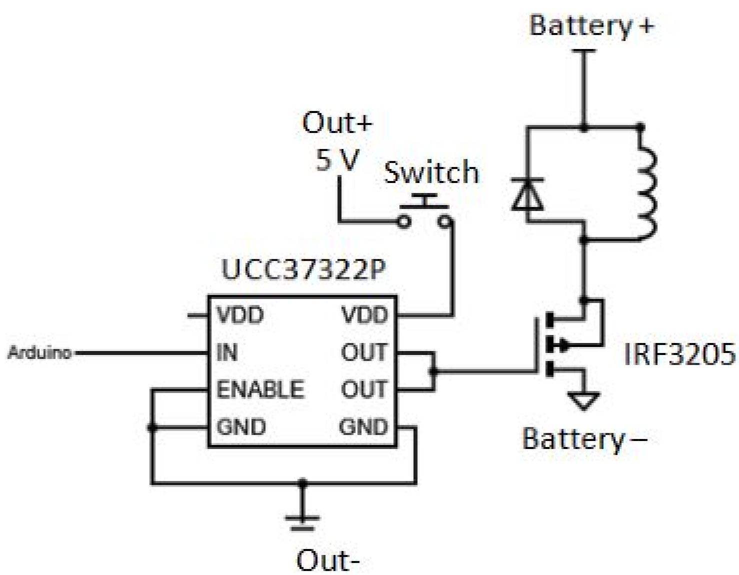
UCC37322 pins are labelled and can be verified in the datasheet. IRF3205 pins are Gate, Drain, Source (left, middle, right pin). Highly recommended to write down pin labels for easy reference when soldering later.

a. **UCC37322 & IRF3205**

What is the purpose of UCC37322 and IRF3205? The IRF3205 mosfet when activated essentially amplifies the current flow in the coil. Thus, its function is easy to see. But we also need a way to turn each IRF3205 on/off and very quickly, which the arduino itself cannot do. The reason we need fast switching on/off: to accelerate the projectile, each stage is turned on/off at very fast and controlled timings. Therefore we use UCC37322 to act as a “driver” to control IRF3205 on/off.

2. **Coil Circuit (1-stage)**

We present the 1-stage acceleration circuit with important changes. While it looks straightforward, the connections/layout can get very complex when multiple stages so see bullet points below.

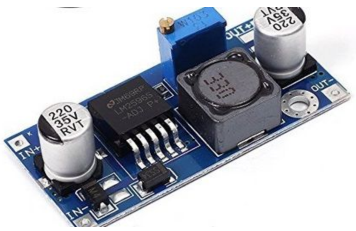


Schematic picture

Important: Cannot be breadboarded as currents involved are very high, use blank PCB board and soldering, so use dip sockets + male header pins to attach components whenever possible.

Important: Every bullet point to explain important functions of the circuit:

- Coil segment + high voltage diode IN4007 (note: direction of diode is reverse parallel) to protect circuit from reverse current
- Power of coil segment will come from the battery+/battery- so cut and use some XT60 thick wire
- Fuses 250V/15+Amp are required so solder 3 fuse clips in parallel to PCB board near DC-DC module. Fuses **must** be between battery and anything else for safety. We put two 15 amp fuses in parallel but may vary based on amps.
- Power/GND of the UCC37322P/Arduino is now Out+ (5V) and Out-, which are on the LM2596 DC-DC module power supply. The LM2596 needs to be adjusted to supply 5V from the battery. Turn the tiny brown dial on top to do so.

DC-DC module.   XT-60 battery plug + thick wire.

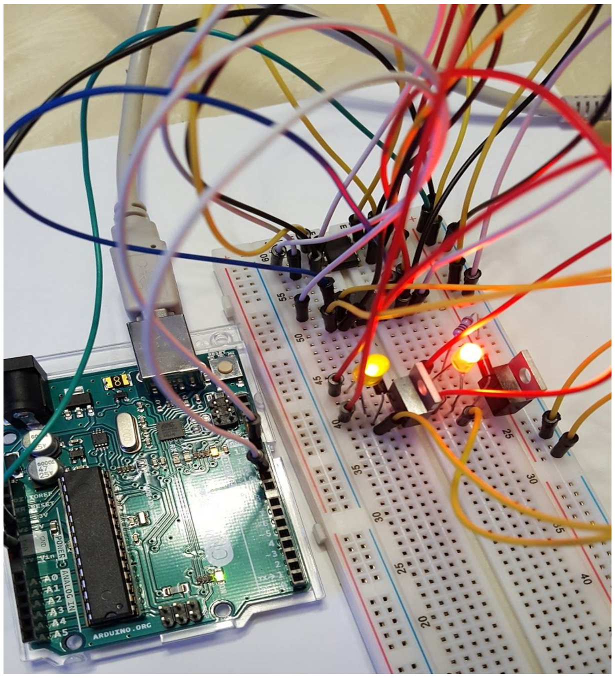
First, solder male pin headers to DC-DC module then solder female pins to board. Next, we want to connect In+/In- to the fuses then XT60 battery plug so solder accordingly. You now have a place where the battery can plug into fuses then to DC-DC module. But in addition, we also want the battery to connect directly to Battery+/Battery-. So cut 2 segments of thick wire (we cut red/black thick wire from XT60) to solder. In summary, battery is being used for 2 purposes: one branch going to the DC-DC module and one branch going directly to Battery+/Battery- location (visualization: like a fork in the road with the starting point being the battery plug + fuses).

- The 1 **switch** we add is used to control power to all the driver UCC37322P's you use; the reason is to only flip the switch on ***after*** battery is plugged in and arduino has time to initialize. Also, turn the switch off after firing to prevent them being active for prolong periods. We also didn't need to use the pin1 VDD on UCC37322P, and only 1 switch is needed to control all.

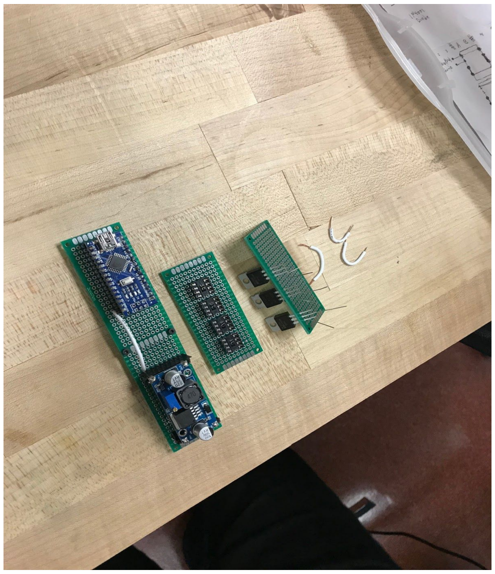
- We use a button connected to arduino to act as the firing signal, when button is pressed a HIGH signal will trigger the firing sequence

3. Build process: soldering and general layout **(Qiyuan, Jingpei)**

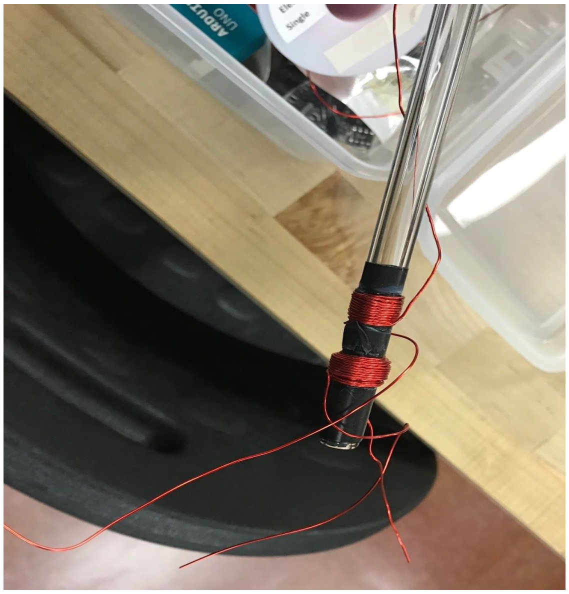
- We first prototype the cannon schematic on the breadboard using LEDs.



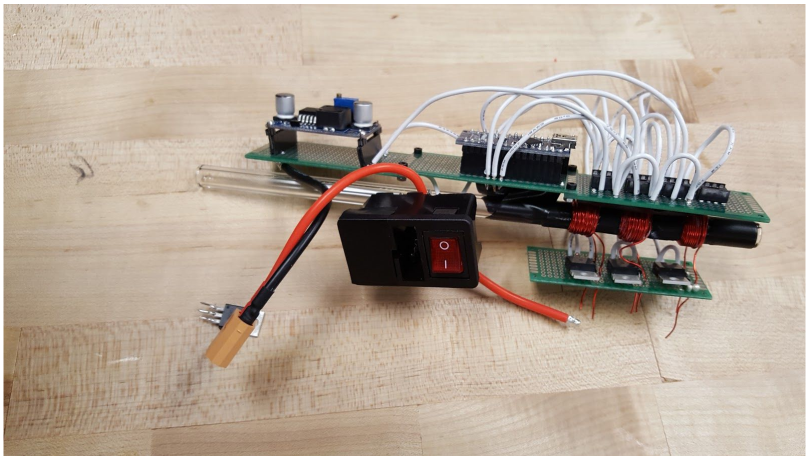
- Then we design the structure of the cannon gun and place all the components we need as we designed. We solder Arduino Nano, DC to DC bock circuit; 4 sockets for IC drivers chip on separate boards and 3 MOSFET and 3 high voltage diodes together on another board.



- We wrapped the coil onto the barrel, which is a glass tube. This is a hard process because we have to wrap 4 layer of coil for each stage and ensure all the wrapping is **in one direction**.To identify the wrapping area for each layer, we use tape in between each stage to identify the area we should wrap the coil. When wrapping each stage, we use superglue after finish wrapping each layer (10 turns) to secure the position of the coil so that we can continue wrapping the next layer. We made 3 segments of coils, each segment consists of 4 layers of coil (40 round of wrapping).



- We connect the fuse into the switch, and connect the button with the live wire.
- We connect and solder the wires according to the schematic shown in the schematic picture. Instead of one MOSFET and one IC chip in the picture, we wire 3 MOSFETs and 3 IC chips in parallel for 3 stages of coils.



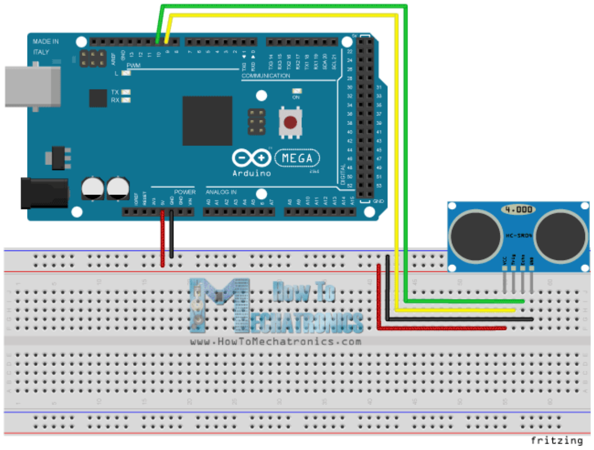
- After all the wiring done, we start to writing the code to test each stages. For testing a stage, we place a ferromagnetic projectile in the barrel close to the coil of the testing stage and use Arduino to send a high signal to the corresponding pin, delay for 0.005 second and send a low signal to that pin. If successfully, we should see the projectile moves.
  Note: when doing the test, don't plug in the battery for a long time. The large current might heat up the coil and damage the circuitry.
- After successfully testing each stage, we write the code to control all 3 stages. We switch the first stage on and off as we testing it. Then delay for a very short time, and switch on and off for second stages. We do the same process for the remaining stages. The delay time between each stage is depend on the distance between the coils. You should switch on the stage just at the time the projectile arrive this stage from the previous stage. The easy way to figure out the delay time is to test several time intervals and find out which one is correct.
- Sometimes there are mistakes of wiring or the hardwares fail. You can use multimeter or LED to diagnose the circuit. Make sure everything is connected correctly and there is not misconnect between the component.
- After the cannon works, we use InkScape to design a case that can be mounted to the base through screws and put the cannon in.

Part #4: Mounting to the base & system troubleshooting and debugging for final network **(All)**

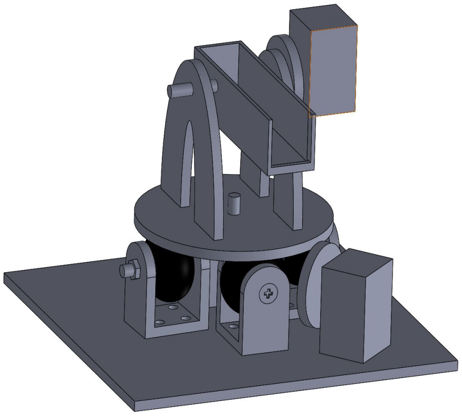
1. **Assembling the cannon framework**
    a. Assemble all the parts that were cut out and 3D printed out as shown in the first schematics for part 2 in preparation for all the other components that will be placed into the cannon to perform their designated functions.
    b. The motor will sit in the center of the base within the structure made to hold it in place, then the rotating disk is to sit on top of the motor so that the motor turns the disk along with it as it turns
    c. Place the 4 stationary ultrasonic sensors around the base of the cannon as they will be the first line of detection that detects any object that passes the threshold distance, the 5th sensor will sit on top of the rotating disk
    d. Fix the cannon in between the framework designed to hold it up on top of the rotating disk
    e. All the wiring is to go around or under the ultrasonic sensors so that they don’t interfere with the signals, and the wiring for the 5th sensor is to go through the disk so that it moves behind the sensors and not in front of them, in such a way so that when the cannon turns back the wires move back to their original position.
    f. Once this is done test that the entire device works, if working correctly, the cannon should begin rotating once one of the 4 stationary sensors detect something and then stop moving once the 5th sensor detects something, or continue moving then move back after one full revolution if nothing is detected by the 5th sensor, if the 5th sensor detects anything then the cannon should stop in place for about 3 seconds (long enough for you to fire the cannon) and then move back to its original spot.

2. **Possible optional improvements (potential project ideas?)**
    a. To make the cannon automatic by sending it a signal to shoot when the 5th detector detects something rather than have it stop in place long enough for it to be manually shot.
    b. Possible to attach servos to the 4 stationary sensors on the base to have the rotated and scan their respective areas in order to further widen their field of detection to allow for better detection of objects within range of the cannon

PROJECT SCHEMATICS:



(Figure 1)



(Figure 2)

PROJECT TIMELINE

Week 1: Test individual parts and learned how to use said parts, also created rough drafts of the cannon and base frameworks.

Week 2: Initial application of the different parts all together, getting them to communicate & work w/ each other. Finished cannon framework.

Week 3: Testing and debugging of the network of all the parts working together in order to achieve the goal of the project. Base framework still a WIP

Week 4: Finished debugging the code for the working of all the different parts of the project and cannon base framework is finished. Assembly of the cannon base begins.

Week 5: Mounting electromagnetic cannon on the disk. Communicating Arduino nano on the cannon and the Arduino uno on the disk. Making fire process smoother.

TEAM MEMBER RESPONSIBILITIES

Edwin Molina - Designed the motor, ultrasonic mount mount, base, cannon mount/assembly via Solidworks

Alberto Avalos - Coding ultrasonic sensors, coding the motor, building sensor and motor circuits, troubleshooting

Qiyuan Wu - Design the cannon circuit; soldering; coding firing; solving firing problems; communicating cannon Arduino Nano to base Arduino Uno

Jingpei Lu - Soldering cannon circuit; wrapping coil layers onto barrel; coding firing; identifying and solving firing problems;

Brenden Diep: Prototyping; soldering cannon circuit; wrapping coil layers; coding firing; organize workflow

REFERENCES

The principle of electromagnetic coil gun:
https://en.wikipedia.org/wiki/Coilgun

Source for solid work license:
www.solidworks.com/sek
https://diyhacking.com/make-coil-gun-without-camera/
http://www.nutsvolts.com/magazine/article/electromagnetic_coil_launcher_project

The principle of the arduino / ultrasonic sensor:
http://www.instructables.com/id/Arduino-and-ultrasonic-sensor-control-servo-and-di/

https://forum.arduino.cc/index.php?topic=125711.0
http://www.themakersworkbench.com/tutorial/triggering-servo-using-hc-sr04-distance-sensor-and-arduino

How to Work the Motor
http://bildr.org/2012/04/tb6612fng-arduino/