MULTISTAGE COIL GUN

By

Jon Dagdagan

Yohan Ko

Shashvat Nanavati

Final Report for ECE 445, Senior Design, Spring 2013

TA: Ryan May

01 May 2013

Project No. 3

Abstract

The Multistage Coil Gun is a three stage device which ejects a small projectile at great speeds solely through the use of electromagnetics. During demonstration, we consistently achieved projectile launch speeds of 14-17$\frac{m}{s}$ while ensuring the safety of the users and the coil gun itself. Though the basic principles are simple, the coil gun requires many components to efficiently coordinate in order for the projectile to be ejected.

In the following sections, we will first cover the functionality of our coil gun and then each of the components involved in detail. We will then discuss the main design requirements and how they were achieved and verified. Safety modifications and isolation of high voltage components will also be detailed throughout various sections. Finally, we will review the economics of our project before giving concluding remarks on the coil gun's successes.

Contents

1. Introduction

1.1 Theory

The basic theory behind the coil gun is simple. Figure 1 shows a single coil winding with a conductive projectile passing through it. As current is discharged through the winding, a magnetic field is created. This field passes through the projectile and the change in the flux induces a current to flow within the projectile [1]. Finally, a force in the launch direction is produced by the field and the induced current.

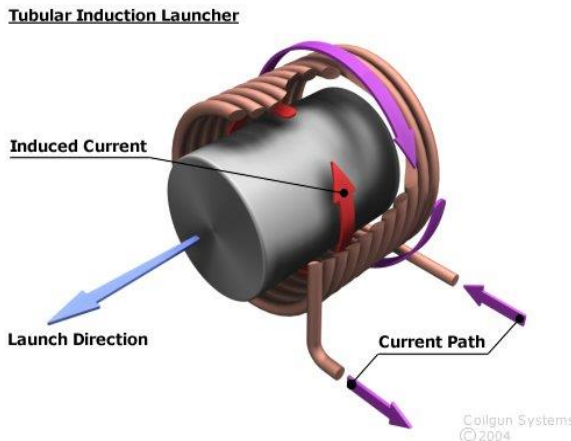

Figure 1 Coil gun theory overview [2].

Our Multistage Coil Gun uses this principle to eject an aluminum cylinder of 1 inch diameter at speeds of 12-17 $^{m}/_{s}$. A novelty of our system is that the projectile is further accelerated at two extra stages after the initial propulsion.

The system we used employs three separate capacitor banks that are charged to a certain voltage to provide the current discharge. The three banks are connected to three different coils that are wrapped consecutively around the barrel. In order to discharge the latter capacitor bank stages at optimal timings, we also used optical sensors and a microcontroller, the details of which will be provided later.

1.2 Purpose

The purpose of our project is to create an educational exhibit to show how electromagnetic force, if harnessed properly, can be used for many applications. We aim to accomplish this by making our project a permanent exhibit for Engineering Open House (EOH). In fact, we demonstrated our coil gun during EOH 2013 and we generated great interest from visitors of all ages. Young students, in particular, could easily see and understand the general scientific principles interacting within our device.

1.3 Overview

There are seven main components to our coil gun design:

- High Voltage Power Supply
- Charge Controller
- Low Voltage Power Supply
- Sensor Circuit
- Microcontroller
- Trigger Circuit
- Coil Windings

The functionality of our coil gun is best described in two separate phases. The first phase consists of all of the mechanisms leading to the discharge of the first capacitor bank and the initial propelling of the projectile towards the second coil winding. The second phase consists of all the mechanisms controlling the discharge of the second and third capacitor banks leading to the eventual ejection of the projectile from the barrel.

1.3.1 Phase 1

The high voltage power supply (HV power supply) is used to charge the capacitor banks through the use of the charge controller as shown in figure 2. Hence, the source of the force which propels the projectile originates from the HV power supply.

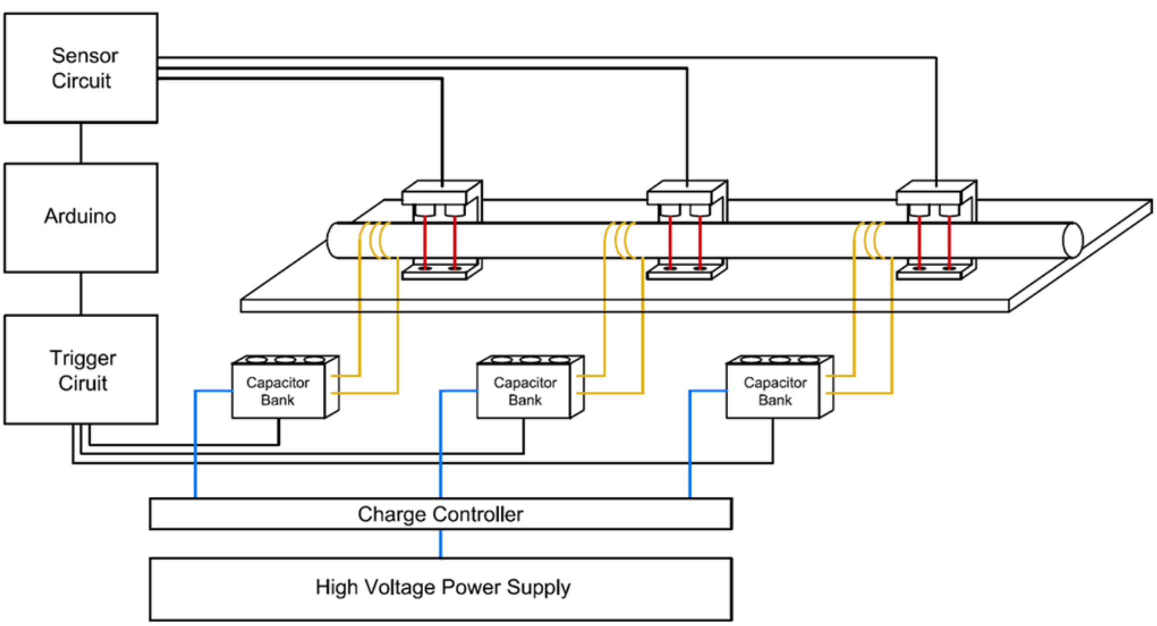

Figure 2 Physical component diagram.

The charge controller consists of three individual switches which can be pressed to charge each capacitor bank. The launch button is also a part of the charge controller and when it is activated, the first capacitor bank discharges which initiates the projectile's movement towards the second coil winding. This concludes the first phase.

1.3.2 Phase 2

The low voltage power supply (LV power supply) is used to power the sensor circuit, microcontroller, and trigger circuit. We have lasers continuously shining on photodiodes after each coil winding. The first set of optical sensors provide a drop in voltage when the projectile, passing through the barrel, comes in between the laser and photodiode. The voltage drop signal is sent to the microcontroller.

The microcontroller uses the information given by the sensor circuit to calculate the speed of the projectile. This is done by using the difference in the times logged for the voltage drop from a set of two optical sensors and the known distance between the sensors. The microcontroller then uses the speed to calculate the optimal discharge time for the next capacitor bank. The detailed explanations of these calculations are given in the design section. Finally, the microcontroller sends a signal to the trigger circuit to activate the gate of the SCRs which discharges the second capacitor bank. This entire phase 2 process is repeated once more for the second set of sensors and third capacitor bank after which the projectile is ejected from the barrel.

Specific details of all components will be further outlined in chapter 2. The testing procedures and verification of our design will be covered in chapter 3. Lastly, chapters 4 and 5 will summarize the costs and accomplishments of our project.

2 Design

2.1 Design Procedure and Details

2.1.1 Low Voltage Power Supply

The LV Power Supply module supplies power to the sensor circuit, microcontroller, and trigger circuit as mentioned before. It also supplies power to computer fans we are using to keep the temperature of each of the coil windings low. The LV power supply must supply 7.5V ± 10% with a current of 1A ± 10% to the microcontroller and 5 V ± 10% with a current of 3 A ± 10% to the other components. The voltage must also be in reference to earth ground.

2.1.2 High Voltage Power Supply

The HV power supply module consists of the three capacitor banks and the high voltage DC power supply. Two of the capacitor banks have three 10,000µF capacitors in parallel and the other has four 10,000µF capacitors in parallel. Since capacitors supplement when they are in parallel, as described by equation 1, we effectively have two 30,000µF capacitor banks and one 40,000µF capacitor bank.

$$C_{eq} = C_1 + C_2 + \cdots + C_n \tag{1}$$

The high voltage DC power was provided by the lab bench in the power lab. The capacitors were rated at 450 V DC but we only charged them up to 120 V since this was adequate to meet our requirements. The charging was regulated by the charge controller module as shown in figure 3. Once the capacitors are charged, they remain charged until the SCRs are triggered. The SCRs then remain activated until the current stops flowing through them. At that point, the SCR's turn off until their gate is triggered again.

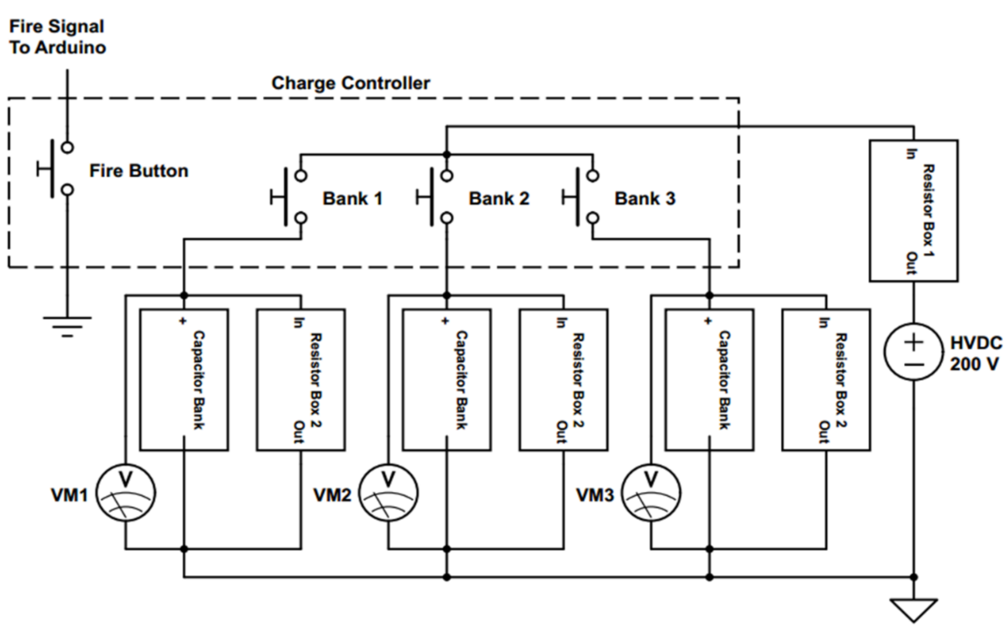

Figure 3 Circuit schematic depicting the high voltage power supply and charge controller circuit.

An alternative to the design shown in figure 3 would be to incorporate a push button on each capacitor bank for charging. The internal connection within the capacitor bank would make the design cleaner and would make it obvious which button charges which capacitor bank.

Next, the capacitors were also modified to include many safety features. These features include:

- Secure covers with latches to limit exposure to all high voltage leads
- Real time voltage monitoring across capacitors using digital multimeters (DMM)
- Shorting rods for automatic discharge of capacitors when opening cover
- Large/small banana plugs for modularity

Even though most of these safety enhancements are mechanical, they were still extremely important to the overall welfare of the users.

2.1.3 Charge Controller

The charge controller allows each of the three capacitor banks to be charged individually. Figure 3 shows the charge controller in relation to the HV power supply.

The design of the charge controller is straight forward. Notice from figure 3 how there are three separate buttons that each complete a circuit between the HVDC supply and the capacitor banks when held down. In addition, a fire button is also encased in the charge controller. One terminal of the fire button is connected to ground while the other is connected to the microcontroller. The microcontroller logic allows the first stage to be triggered when the Arduino signal is grounded. Hence the charge controller not only charges the capacitors, but it is also responsible for discharging the first stage.

2.1.4 Sensor Circuit

The sensor circuit module included low power lasers, photodiodes, potentiometers, and amplifiers. The way the sensor circuit works is that the lasers shining on the photodiodes will be interrupted when the projectile passes through. This interruption will result in a voltage change of approximately 100mV.

Next, the operational amplifier will take the photodiode voltage output and compare it to a reference voltage created by potentiometer. The reference voltage, which is the threshold voltage in which the photodiode is excited, will fluctuate from 150 mV to 200mV. This range of the reference voltage arises from the variation of the ambient lighting of the room. The amplifier then outputs a signal of 4 V ± 15% (HIGH) when the laser is shining on the photodiode ($V_{photodiode}>V_{reference}$) and 0.5 V ± 20% (LOW) when the laser is blocked by the projectile when it is passing through ($V_{photodiode}<V_{reference}$). The schematic of the sensor circuit is shown in figure 4 and it highlights the fact the output has to be amplified before being sent to the microcontroller.

There are three sets of two photodiodes and lasers in the coil gun system. The first set of photodiodes and lasers will be placed after the first coil winding, the second set after the second winding, and the third set after the third windings. The last set of detectors will be used to calculate the final speed of the projectile leaving the barrel. This speed will be displayed on the serial monitor of the microcontroller.

The primary technical challenge we faced in implementing the sensor circuit was in regards to output signal generation. Initially, our design did not incorporate an amplifier and we were feeding the output signal directly to the microcontroller. We realized, however, that a low signal, which is only around 100mV, was not powerful enough to be detected by the microcontroller. Hence, the capacitor banks would never discharge. To resolve this issue, we included an amplifier to strengthen the signal before being outputted to the microcontroller.

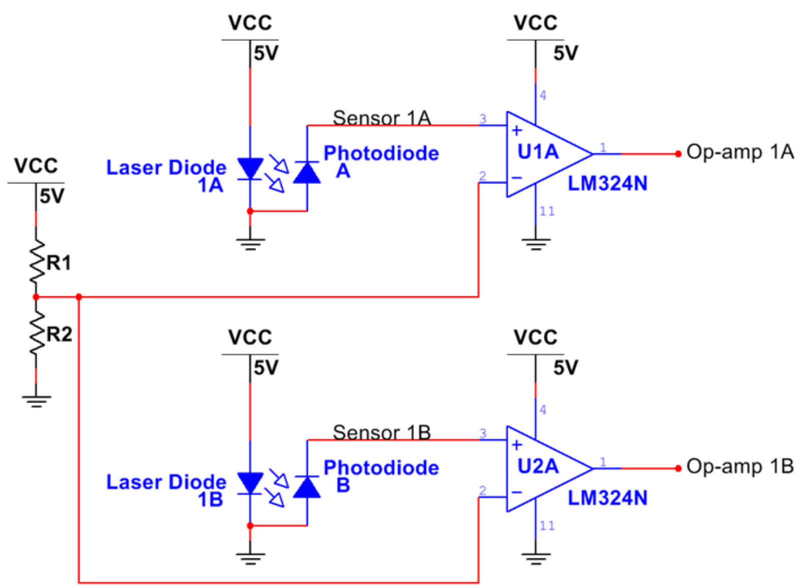

Figure 4 Sensor circuit schematic. The laser diodes will always be on. The voltage across the photodiodes depends on whether the projectile is blocking the light from the lasers.

2.1.5 Microcontroller

This microcontroller is a vital component of our project. The purpose of the microcontroller is to automate the firing of the latter two stages of the coil gun. It logs the times when the projectile passes through the two sensors and using the known distance between sensors, it calculates the speed of the projectile. The relation to conduct this calculation is just the trivial v=d/t equation from basic physics. Then, using the speed and the known distance between the sensors and the next coil, it will wait for a certain period of time before triggering the coils for the next stage.

For our purposes, we used an Arduino Uno since it was easy to use and readily accessible. The Arduino is able to run at 115,200 baud, which is quick enough to complete the velocity calculations while the projectile is moving. The Arduino was connected as seen in figure 5. It reads in 6 digital inputs from the sensor circuit and outputs 3 digital outputs to the trigger circuit.

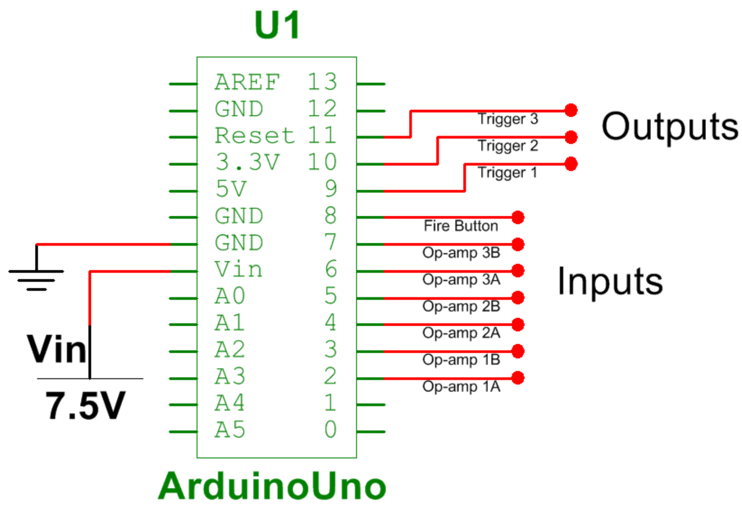

Figure 5 Microcontroller connections.

Looking at equation 4 we can see that the Arduino simply needs to multiply the time difference between the two diodes by a distance constant, α. It then waits that amount of time before sending the signal to the trigger circuit to fire the next coil. The constant α is simply a ratio of the distance between a set of lasers and coils, d, divided by the distance between the set of two lasers, x, as seen in figure 6. It is important to mention that α can be easily adjusted within the Arduino code to achieve better velocities after experimental analysis. Appendix B holds the Arduino code itself.

$$v = \frac{x}{t_f - t_i} \tag{2}$$

$$t_{delay} = \frac{d}{v} \tag{3}$$

$$t_{delay} = \frac{d}{x}(t_f - t_i) \qquad \alpha = \frac{d}{x} \tag{4}$$

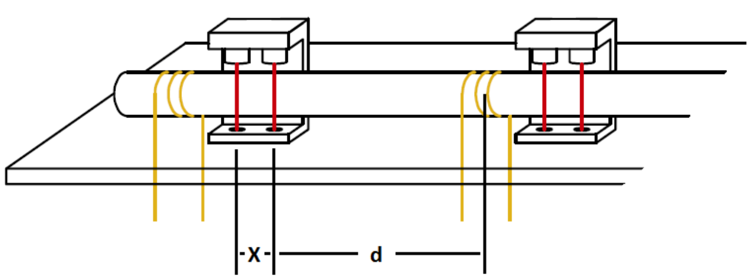

Figure 6 Coil gun spacing.

2.1.6 Trigger Circuit

This module consisted of optocoupler relays and SCRs. When an optocoupler receives a HIGH signal from the microcontroller, it relays the signal in the range of 1 V ± 20% and 50mA ± 20% to the SCR's and activates them (the optocoupler will provide electrical isolation to the microcontroller from the HV power supply). For our project, a component with an optocoupler matched with relay was used.

Given the high power nature of the project, it was essential to estimate what sort of current was expected to be discharged through the coils. Initially, a simulation was conducted on the current dissipation through each set of coils. As seen in figure 7, at 120Vdc, it was estimated that 2.8kA of peak current would be flowing through each set of coils. Knowing this information we estimated speed capabilities of the system and more importantly selected parts which adequately met those voltage and current demands.

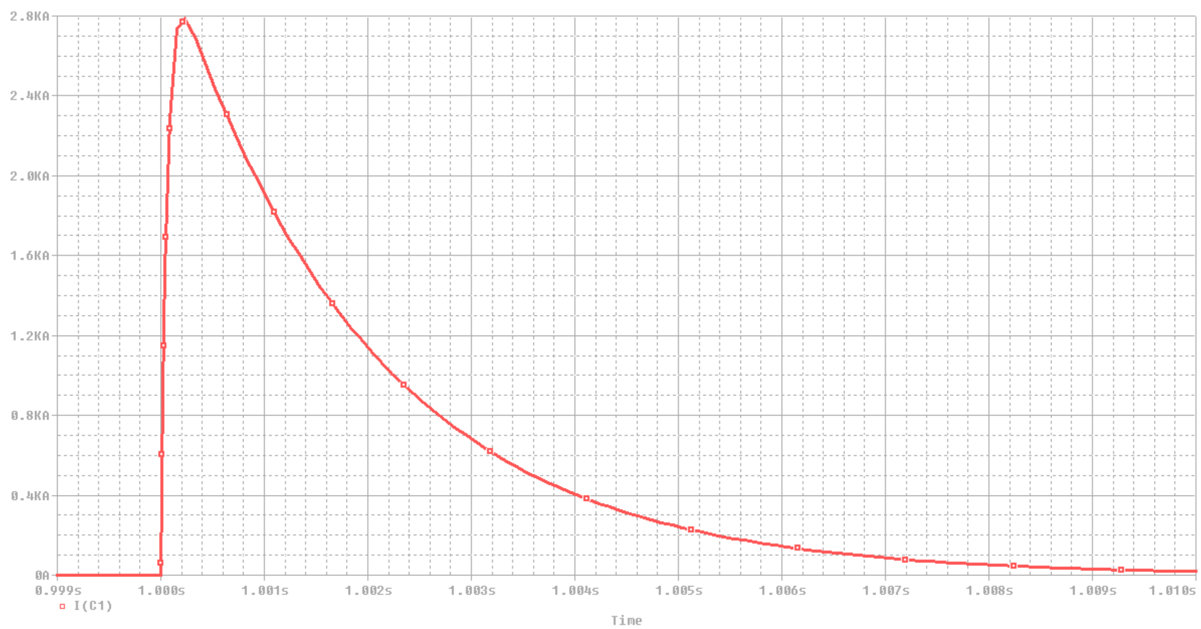

Figure 7 Peak current as a function of time. The current initially peaks at 2.8kA before decaying.

The SCRs we chose can withstand a voltage difference of 400 V and a maximum peak surge current of 2.18 kA [3]. This limit provided us with the absolute maximum voltage we could charge the capacitors with using the HV power supply. In the lab, we charged only up to 120 V as mentioned before.

Regarding the design of the circuit itself, a set of three SCRs were placed in parallel for each capacitor bank to carry the discharged current. This lowered the current stress on each SCR and provided a safety measure that guaranteed a path for the current even if one of the SCRs failed.

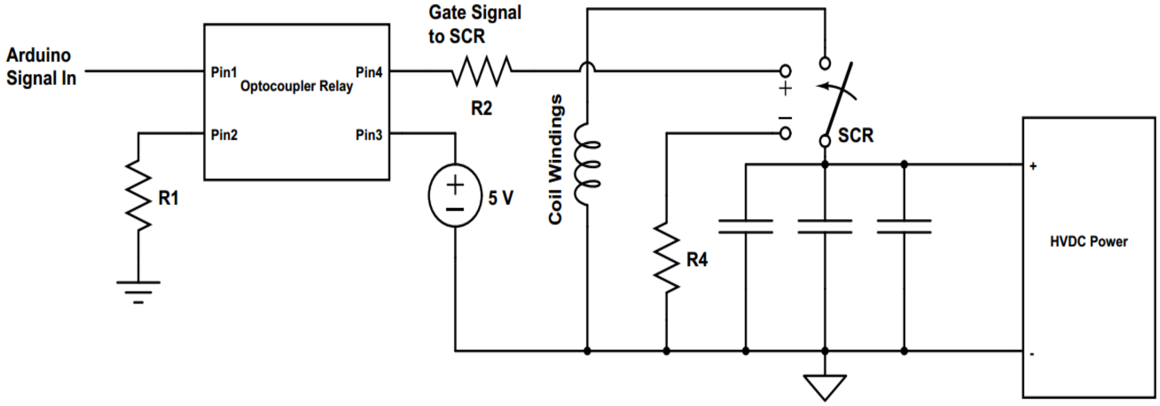

Figure 8 Circuit schematic depicting the trigger circuit, capacitor bank, and coil windings.

Figure 8 shows the 3 SCR's in parallel as a voltage controlled switch. This was done to represent the physical terminals of a power SCR. A typical circuit schematic of an SCR shows a diode with one gate terminal. However, a physical power SCR has four leads. One is the gate, another one is the anode, and the remaining two are cathodes. In order for current to flow through the anode and cathode, a voltage must be supplied to the gate in reference to the cathode. Figure 8 shows this requirement clearly being met.

Next, the resistance value of R4 needed to be very large with a large power rating in order to prevent any current from flowing through it. The resistance values of R1 and R2 were 6 ohms and 0.33 ohms respectively. These resistors limit the amount of current flowing through the optocoupler relay.

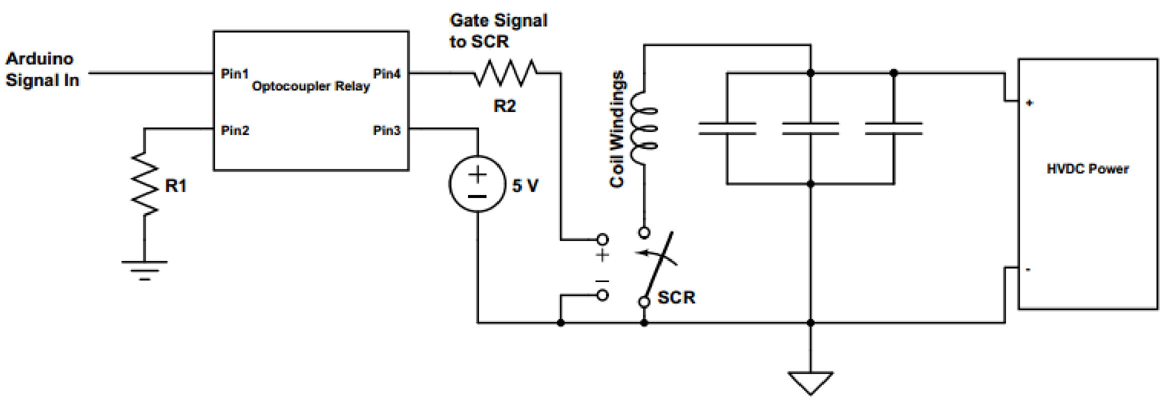

Figure 9 Alternative design of the trigger circuit, capacitor bank, and coil windings.

An alternative design for the trigger circuit is shown in figure 9. It allows the gate signal of the SCR to be in reference to ground. This would be safer as it would take away the need for R4. Unfortunately, we were unable to implement this alternative circuit because of the designs modifications we had already implemented on the capacitor banks.

2.1.7 Coil Windings

The coil winding module consisted of the three coils which were connected to each capacitor bank. When a set of SCRs are triggered, current is discharged from the capacitor bank and flows through the coil. The current through the coil creates a magnetic field that induces a current onto the projectile. This generates a force between the induced current and the magnetic field of the coil that propels the projectile.

We used 12 gauge copper wire with a thin film of insulation for each of the windings. We minimized resistance by placing two windings in parallel for each stage. To maximize the mutual inductance, we made the length of the winding to be equal to the length of the projectile. The energy expended for a fixed coil length of 20 mm as a function of projectile length is shown in figure 10.

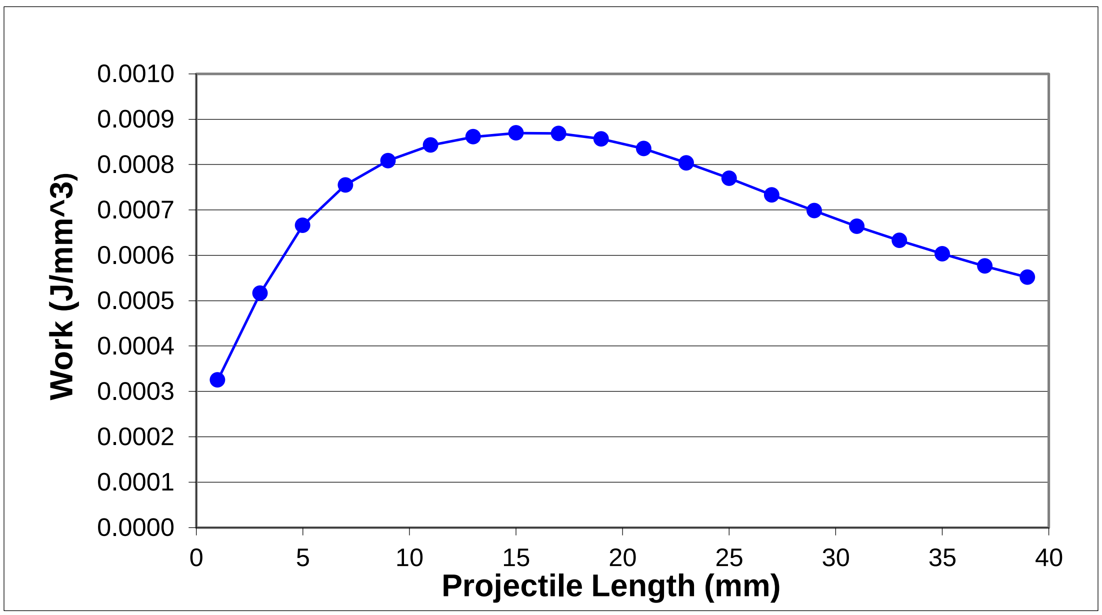

Figure 10 Energy per volume (20mm coil) vs. projectile length [4].

One can see from the graph how matching the projectile length and coil winding length will roughly maximize the work done on the projectile. Next, we also made the diameter of the projectile as close as possible to the coil winding diameter. This is done to maximize the area of the projectile which allows a larger current to be induced on the projectile’s surface. The only limitation which we had to take into consideration was that we had to leave a small gap between the projectile and the barrel. If we excluded this gap, then friction loss would become dominant and would counteract any of the additional force generated by the larger projectile size.

3. Design Verification

3.1 Requirements and Verification

We were able to verify the requirements for each subsystem module. The requirements and verification table is included in Appendix A and we will next cover the main requirements of each component.

The LV power supply requirements were easy to verify using the voltage and current display of the Kenwood power supply. The supply also allowed us to detect if any shorting occurred in our circuit. The sensor circuit lasers were also easy to verify since the lasers were visibly on if the connections were made properly. During testing, we verified requirement 3.3 by using a wooden stick to block the laser from shining on the photodiode and probing the output of the op-amp. We measured a voltage reading of 4V± 15% (depending on the ambient lighting) when the laser was not blocked and a reading of 0.5V± 20% when the laser was blocked. We performed this test for each set of lasers/photodiodes.

The microcontroller requirement 3.5 was easily verified by plugging the Arduino to the computer using a USB cable. With the requirement met, we were able to upload our code into the Arduino. Next, requirement 3.7 was verified by probing the output of the Arduino going to the gate of each of the capacitor banks. Figure 11 shows the delayed triggering of the consecutive stages.

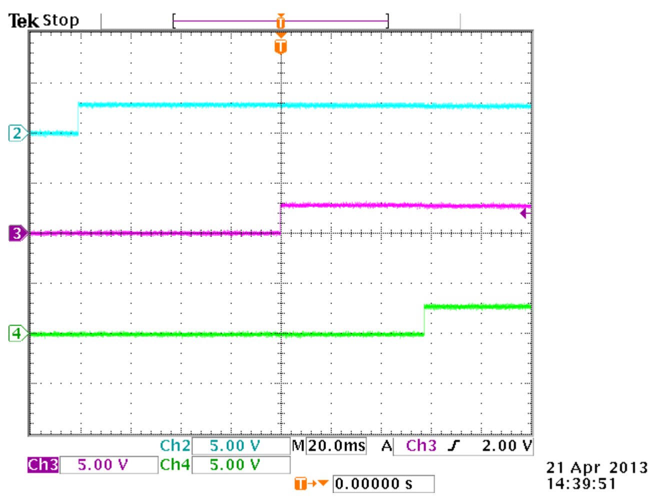

Figure 11 Arduino output to gate. Notice the delayed triggering of consecutive stages.

The plot above was an unofficial high level verification for all of the subsystems that came before the Arduino. It proved to us that the sensors were working and that the microcontroller was able to record the times that the each sensor was triggered. The difference in the delay of the second to third stage and the delay of the first to second stage was also proof that the microcontroller was calculating the speed of the projectile correctly. Notice from figure 11 how it took the second stage longer to trigger compared to the third stage due to the fact that the projectile had been accelerated one more time before passing through the third stage.

The trigger circuit requirements were verified by seeing if the SCRs triggered when the Arduino sent a signal to the optocoupler. This was accomplished by placing each SCR in the test circuit shown in figure 12. The voltmeter would initially read a value of 0 V until a signal is supplied to the gate of the SCR. The voltage would then read 24 V even if the gate signal is removed. The voltage would read 0 V only once the 24 V power supply is reset.

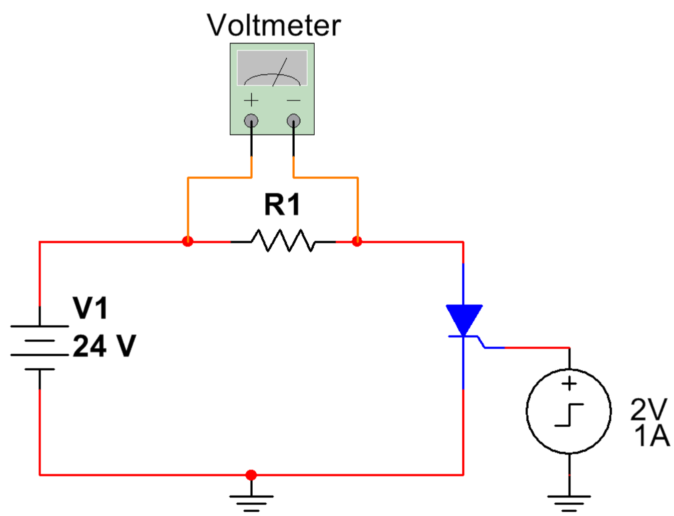

Figure 12 SCR test circuit to verify proper triggering.

The HV power supply requirements were performed by placing three Kenwood power supply in series and measuring the voltage across the power supply. The capacitor leakage specified by requirement 3.14 was verified by charging the capacitor to 50 V and ensuring the voltage did not decrease past 45 V after 5 minutes. Next, the safety requirement outlined in 3.15 was verified visually and by measuring the resistance across the bars with the enclosure opened. The voltage was found to be initially 0 ohms but then started to rise because of the DMM charging the capacitors. This was still a sign that the capacitors were indeed shorted.

The charge controller requirements were verified by seeing that the resistances of switches were infinite when open and 0 ohms when pressed. Connections between the power supply and capacitor banks were verified by seeing that the capacitor bank voltage increased when the charge controller button was pressed.

Finally, the coil windings requirements were met by choosing high temperature rated adhesives and using fans to keep the coil temperatures cool. The requirements were verified by making sure the coils did not slide even when forcefully twisted.

4. Costs

4.1 Parts

Table 1 Item Costs

Item	Part Number	Retail Cost($)	Quantity	Actual Cost($)
Photodiode	BPW34	1.00	15	15.00
Laser	B002IX97NS	5.00	10	50.00
Op-amp	LM324	0.10	2	0.20
Potentiometer	319C W 103	0.25	3	0.75
Arduino	A000006	23.95	1	23.95
Optocoupler	CPC1718J	0.64	3	1.92
Thyristor (110A, 400V)	110RKI40	42.68	6	256.08
Thyristor (175A, 600V)	T610061804BT	40.53	3	121.59
Gate Driver	44OCT 0203	2.03	3	6.09
Capacitor (10000µF, 450V)	DCMC103P450DG2BS	70.00	10	700.00
Capacitor (1000µF, 25 V)	031ME	1.44	1	1.44
Resistor (0.25Ω, 5W)	RW57VR25	0.04	3	0.12
Resistor (6Ω, 0.25W)	-	0.08	3	0.24
Resistors (33Ω, 0.25W)	-	0.07	3	0.21
PVC Pipe	-	1.50	1	1.50
Launch Platform	-	3.00	1	3.00
Barrel Holder	-	20.00	1	20.00
Capacitor Cover	-	6.00	3	18.00
Capacitor Cover Latches	-	1.33	6	7.98
Capacitor Frame	-	10.00	3	30.00
Copper Bar (1.5 ft)	-	54.00	6	324.00
Fiberglass Bar (10 in)	-	2.50	6	15.00
Winding	-	1.00	6	6.00
Wire (1 meter)	-	1.00	1	1.00
Projectile	-	2.00	1	2.00
Screws, blots, other mechanical items	-	5.00	-	5.00
PCB	-	0.00	1	0.00
Total	-	-	-	**1611.07**

4.2 Labor

Table 2 Labor Costs

Name	**Hourly Rate**	**Hours Spent**	**Total Labor Cost** $\mathbf{Hourly\ Rate} \times \mathbf{2.5} \times \mathbf{Hours\ Spent})$
Jon Dagdagan	\$40.00	200	\$20000
Yohan Ko	\$40.00	200	\$20000
Shashvat Nanavati	\$40.00	200	\$20000
Machine Shop	\$40.00	40	\$4000
Total	\$120.00	600	**\$64000**

4.3 Total Costs

Table 3 Total Costs

Parts	\$1611.07
Labor	\$64,000
Grand Total	**\$65,611.07**

5. Conclusion

The project was successful given that we were able to most of our requirements. We were able to demonstrate the project at the Engineering Open House over the course of two days as well as for our final demonstration. We will now cover our accomplishments, any ethical issues we had to take into consideration, and the next steps involved with improving the coil gun.

5.1 Accomplishments

We consistently achieved projectile speeds of 14-17$\frac{m}{s}$ throughout the duration of Engineering Open House. In fact, we were awarded 2nd place in the "Back to School" EOH awards category for our efforts. Further, the project was also awarded the Bitzer-Slottow Creativity Award by the Electrical and Computer Engineering department.

The final status of our project consisted of all components successfully coordinating with each other. All of our safety enhancements functioned properly as well. Modularity was well maintained and this was evident by the fact that each component could be easily separated from the overall coil gun system.

5.2 Ethical considerations

We held each point of the IEEE Code of Ethics to the highest regards and utmost importance. The following, however, are the IEEE ethics which pertained to our project in particular.

1. *To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment.*

   Given that our project involved high voltage, we gave full commitment to ensuring the safety of the public, our colleagues, and ourselves through numerous safety enhancements and protocols. We completed General, Electrical, Power Lab, and EOH safety training and thus knew how to responsibly make decisions which ensured the welfare of all parties involved. Swift corrective action was taken whenever potentially dangerous flaws within the project were identified.

3. *To be honest and realistic in stating claims or estimates based on available data.*

   We conducted our calculations, estimates, and simulations with the highest accuracy possible. In addition, we never obscured project results to benefit us at the expense of others' safety.

7. *To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others.*

   We provided accurate, sincere feedback during all peer review sessions. We also took into consideration the opinions of others and implemented all necessary modifications identified. Since our project was a continuation from previous semesters, we only took credit for the things we did and have credited those who were involved previously wherever necessary.

9. *To avoid injuring others, their property, reputation, or employment by false or malicious action.*

   We never used our coil gun for meaningless purposes and our primary objective was always to create an educational tool for future EOH exhibits. Further, we never conducted actions which caused property damage and safety violations.

5.3 Future work

There is scope for additional features to be implemented on the coil gun. Most of these features are for enhancing either the safety or the visual display of the coil gun. The potential project work includes:

1. Obtain silk-screened PCB
2. Construct enclosure for barrel
3. Construct adjustable tilt for the launch base
4. Use rifling on the inside of the barrel to improve the projectile's aerodynamic stability
5. Audio/visual alert system that notifies user once capacitors have been charged past 300V

Most of these changes are mechanical modifications. Given that we are leaving the PCB files and operation manuals with the project, future teams working on the coil gun should not have a difficult time implementing these changes. However, a deeper analysis may need to be done on whether or not the implementation of the alternate trigger configuration is needed for the longevity of the SCRs.

References

[1] Barros, Sam. PowerLabs Multi Stage Coil Gun Page. Available at: http://www.powerlabs.org/coilguns.htm

[2] Paul, James. (2006) *Coilgun Basics.* Available at: http://www.coilgun.eclipse.co.uk/coilgun_basics_1.html, 2013

[3] POWEREX , PA. Phase Control SCR. Available at: http://www.galco.com/techdoc/prx/t610061804bt_dat.pdf

[4] Hansen, Barry. (2013) *Solving Solid Cylindrical Projectile of various lengths.* Available: http://www.coilgun.info/femm/projectile_length/projectile_length_results.txt

Appendix A Requirement and Verification Table

Component	Requirement	Verification
Low Voltage Power Supply	3.1 Power supply must provide 5 V ± 10% at 3 A ± 5%	3.1 Connect the low voltage system to the power supply. Use a voltage probe connected to an oscilloscope to measure the output voltage and ensure that it is within ± 10% of 5 V. Use a DMM to measure the current output of the power supply and ensure that it is within ± 5% of 3 A
Sensor Circuit	3.2 Photodiodes must output a voltage in the range of 250mV when laser is shining and 150mV when laser is blocked	3.2 Measure the voltage of the photodiode using a voltage probe connected to an oscilloscope. Ensure that the voltage is ▪ No less than 250 mV when laser is shining ▪ No more than 150 mV when laser is blocked
	3.3 Op-Amps must convert photodiode signal to digital signal (4 V ± 15% for HIGH 0.5 V ± 20% for LOW)	3.3 Measure the voltage of the photodiode using a voltage probe connected to an oscilloscope. Ensure that the voltage of the output on the op-amp is ▪ Within ± 15% of 4 V when the laser is shining ▪ Within ± 20% of 0.5 V when the laser is blocked
	3.4 All laser diodes must turn on when connected to power supply	3.4 Connect sensor circuit to power supply. Check to see that the red lasers are on and verify that the lasers are shining on the photodiode directly below laser
Microcontroller	3.5 Arduino communicates with PC	3.5 Plug USB cable into arduino and PC. Run arduino program. Under Sketchbook > Examples > Digital select Blink. Click Upload. Arduino LED should blink at about 1 Hz

Microcontroller	3.6 Detect fault for when photodiodes do not trigger in order	3.6 Connect microcontroller to a computer. Run code and open the serial monitor. Provide a HIGH signal to op-amp 1B. A fault message should appears on the serial monitor and the next stage does not trigger. Repeat for op-amp 2A, 2B, 3A, and 3B.
	3.7 Calculate speed of projectile and using the speed, calculate the time to send signal to trigger circuit	3.7 Connect the microcontroller to a computer. Run code and open the serial monitor. Slowly push the projectile through the barrel. Ensure that the serial monitor displays the logged times and projectile speed
	3.8 Output digital signal(4 V ± 15% for HIGH 0.5 V ± 20% for LOW) to trigger circuit	3.8 Measure the voltage of the output terminal going to the trigger circuit using a voltage probe. Ensure that a signal with a 4V ± 15% is triggered after the projectile has been pushed through the preceding set of sensors
Trigger Circuit	3.9 Provide isolation between HV power supply and microcontroller	3.9 Use a DMM to test the resistance across the input of the microcontroller and the output of the optocoupler. The resistance should be in the 10 MΩ range as specified by [3]
	3.10 Optocoupler must relay microcontroller signal and send output signal to gate driver	3.10 Supply a 5 V ± 10% signal pulse to the input node of the optocoupler. The output voltage of the optocoupler, measured with an oscilloscope, should be 5 V ± 20%
	3.11 Gate driver must output a signal powerful enough to trigger the thyristors	3.11 Supply a 5 V ± 10% signal pulse to the input node of the gate driver. The output voltage and current of the gate driver, measured with an oscilloscope, should be 1 V ± 20% with 50mA ± 20%
	3.12 Thyristors, when triggered must remain latched until all the current is discharged	3.12 Place thyristor in test circuit shown in figure 11. The initial voltage reading across the resistor should be zero. Once the gate is triggered ensure that the voltage across the resistor is 24V ± 10% that it remains latched. Turn the power supply off and back on again and ensure that the thyristor has been reset and that the voltage across the resistor is zero again

High Voltage Power Supply	3.13 Bench power supply must provide 120 V ± 10% voltage with 3 A ± 20% current	3.13 Ensure that the HV power supply is off. Connect the power supply in series with a 500Ω power resistor. Connect a DMM across the resistor to monitor voltage. Turn on the power supply. The voltage reading should be 120 V ± 12%. Turn off power supply after testing.
	3.14 Capacitors must retain 95% of the charged voltage for at least five minutes	3.14 Charge the capacitors to 50V ± 10% while monitoring the voltage across the capacitors using a DMM. After five minutes ensure that the capacitor voltage is within 95% if the initial charged voltage
	3.15 Automatically short the capacitors when enclosure is opened	3.15 Check that the voltage across the capacitors is zero. Take the enclosure off. Check to see that the aluminum bar is making physical contact with both the copper bars attached to the capacitors. Also check to see that the resistance across the copper bars is zero using a DMM
Charge Controller	3.16 Relay a ground signal to the microcontroller when the fire button is pressed	3.16 Use a voltage probe connected to an oscilloscope to measure the resistance between the output fire pin to the microcontroller and ground. Ensure that the resistance is 0 ohms when the fire button is pressed
	3.17 Push button switch must connect bench power supply to the corresponding capacitor bank	3.17 Use a DMM to measure the resistance across the two leads of each push button switch. Ensure that the leads are short circuited if and only if the push button is pressed
Coil Windings	3.18 Coil Windings must be in parallel and be able to carry the current discharged by capacitor	3.18 Using an RLC meter, check the resistance across the four leads of each set of coils to verify that two of the leads are shorted
	3.19 Epoxy adhesive must be able to withstand peak temperatures of 400°F and keep integrity of winding structure	3.19 Check the physical condition of the winding structure. Ensure that no parts are loose and that the coil remains in tight contact with the barrel

Appendix B Arduino Code

```
int detect1A = 5;
int detect1B = 6;
int detect2A = 7;
int detect2B = 8;
int detect3A = 9;
int detect3B = 10;
int fireButton = 11;
int coil1 = 4;
int coil2 = 3;
int coil3 = 2;

float delay1 = 0;
float delay2 = 0; //initializing delay constants

unsigned long startTime = 0;
unsigned long detect1StartTime = 0;
unsigned long detect1EndTime = 0;
unsigned long detect2StartTime = 0;
unsigned long detect2EndTime = 0;
unsigned long detect3StartTime = 0;
unsigned long detect3EndTime = 0;
unsigned long endTime = 0;

unsigned long coil2FireTime = 0;
unsigned long coil3FireTime = 0;

void setup()
{
  pinMode(detect1A, INPUT);
  pinMode(detect1B, INPUT);
  pinMode(detect2A, INPUT);
  pinMode(detect2B, INPUT);
  pinMode(detect3A, INPUT);
  pinMode(detect3B, INPUT);
  pinMode(fireButton, INPUT);
  pinMode(coil1, OUTPUT);
  pinMode(coil2, OUTPUT);
  pinMode(coil3, OUTPUT);

  digitalWrite(coil1, LOW); //verify initial trigger signals are low
  digitalWrite(coil2, LOW);
  digitalWrite(coil3, LOW);
  digitalWrite(11, HIGH); //pull-up resistor for fire signal, should prevent misfire

  Serial.begin(115200); //set arduino to highest baud rate
  Serial.println("Start!");
  waitForFire();
}

void waitForFire()
{
  Serial.println("WAITING TO FIRE...");
  while (digitalRead(fireButton)) //wait until arduino gets fire signal
  {
    delay(10);
  }

  Serial.println("FIRE!!!");
  digitalWrite(coil1, HIGH); //trigger first stage
  startTime = micros(); //log start time
  delay(2);
}

void loop ()
{
  if (startTime!= 0 && micros() > startTime +1000000) //loop ends after 1 second
  {
    endTime = micros();
```

```
  finish();
}

if (!digitalRead(detect1A)) //detection for first set of detectors
{
  if (detect1StartTime == 0)
  {
    detect1StartTime = micros(); //recording first detection time
  }
}

if (!digitalRead(detect1B))
{
  if (detect1StartTime == 0) //making sure this isn't detected before first photodiode
  {
    Serial.println("FAULT: detect1B triggered before start.");
    fault();
  }
 else
  {
    if (detect1EndTime == 0)
    {
      detect1EndTime = micros(); //recording second detection time
      delayFireCoil(2); //fire second coil
    }
  }
}

if (!digitalRead(detect2A)) //detection for second set of detectors
{
  if (detect1StartTime == 0)
  {
    Serial.println("FAULT: detect2A triggered before start.");
    fault();
  }
  else if (detect2StartTime == 0)
  {
    detect2StartTime = micros(); //recording first detection time
  }
}

if (!digitalRead(detect2B)) //detected before first photodiode
{
  if (detect1StartTime == 0)
  {
    Serial.println("FAULT: detect2B triggered before start.");
    fault();
  }
  else
  {
    if (detect2EndTime == 0)
    {
      detect2EndTime = micros(); //recording second detection time
      delayFireCoil(3); //fire third coil
    }
  }
}

if (!digitalRead(detect3A))//detection for third set of detectors
{
  if (detect1StartTime == 0)
  {
    Serial.println("FAULT: detect3A triggered before start.");
    fault();
  }
  else if (detect3StartTime == 0)
  {
    detect3StartTime = micros(); //recording first detection time
  }
}
```

```
  if (!digitalRead(detect3B))
  {
    if (detect1StartTime == 0)
    {
      Serial.println("FAULT: detect3B triggered before start.");
      fault();
    }
    else
    {
      if (detect3EndTime == 0)
      {
        detect3EndTime = micros(); //recording second detection time
      }
    }
  }
}

void delayFireCoil(int i)
{
  unsigned long time = 0;

  /////////HERE ARE THE DELAY CONSTANTS///////////
  delay1 = 1.5; //can tweak to get better results
  delay2 = 2;

  time = (i == 2) ? delay1 * (detect1EndTime - detect1StartTime)/1000 :
                    delay2 * (detect2EndTime - detect2StartTime)/1000 ;

  delay(time); //waiting for the time specified above before firing

  digitalWrite((i == 2) ? coil2 : coil3, HIGH); //Firing Coil 2 and Coil 3

  if (i == 2)
  coil2FireTime = micros(); //record fire times
  else
  coil3FireTime = micros();
}

void finish() //displaying results
{
  Serial.println("DONE!");

  if (findSpeed(detect1StartTime, detect1EndTime) > 0)
  {
    Serial.print("Start time: ");
    Serial.println(startTime);
    Serial.println();

    Serial.print("Detector 1 start time: ");
    Serial.println(detect1StartTime);

    Serial.print("Detector 1 end time: ");
    Serial.println(detect1EndTime);

    Serial.print("Calculated speed: ");
    Serial.println(findSpeed(detect1StartTime, detect1EndTime));
    Serial.println();

    Serial.print("Fired coil 2 at: ");
    Serial.println(coil2FireTime);
    Serial.println();
  }

  if (findSpeed(detect2StartTime, detect2EndTime) > 0)
  {
    Serial.print("Detector 2 start time: ");
    Serial.println(detect2StartTime);

    Serial.print("Detector 2 end time: ");
    Serial.println(detect2EndTime);
```

```
    Serial.print("Calculated speed: ");
    Serial.println(findSpeed(detect2StartTime, detect2EndTime));
    Serial.println();

    Serial.print("Fired coil 3 at: ");
    Serial.println(coil3FireTime);
    Serial.println();
  }

  if (findSpeed(detect3StartTime, detect3EndTime) > 0)
  {
    Serial.print("Detector 3 start time: ");
    Serial.println(detect3StartTime);

    Serial.print("Detector 3 end time: ");
    Serial.println(detect3EndTime);

    Serial.print("Calculated speed: ");
    Serial.println(findSpeed(detect3StartTime, detect3EndTime));
    Serial.println();
  }
  Serial.print("End time: ");
  Serial.println(endTime);

  Serial.print("Ratio1: "); //velocity ratio between first and second stages
  Serial.println(findSpeed(detect2StartTime, detect2EndTime)/findSpeed(detect1StartTime,
detect1EndTime));
  Serial.println();
  Serial.print("Ratio2: "); //velocity ratio between third and second stages
  Serial.println(findSpeed(detect3StartTime, detect3EndTime)/findSpeed(detect2StartTime,
detect2EndTime));
  Serial.println();

  resetAll();
}

float findSpeed(unsigned long sTime, unsigned long eTime)
{
  return 30000.0 / (eTime - sTime); //distance between diodes divided by time
}
void fault()
{
  Serial.println("FAULT DETECTED, WAITING...");
  delay(1000);
  resetAll();
}
void resetAll() //reset parameters, prepare for another run
{
  startTime = 0;
  detect1StartTime = 0;
  detect1EndTime = 0;
  detect2StartTime = 0;
  detect2EndTime = 0;
  detect3StartTime = 0;
  detect3EndTime = 0;
  endTime = 0;

  coil2FireTime = 0;
  coil3FireTime = 0;

  digitalWrite(coil1, LOW);
  digitalWrite(coil2, LOW);
  digitalWrite(coil3, LOW);

  Serial.println("Ready.");
  while (!digitalRead(fireButton)) //wait for the fire button on charge controller to be unlocked
    delay(5);
  waitForFire();
}
```