PROJECT COILGUN

The Art and Science of PCB Design 6.S092

Track 2 Write-Up

By Alonso Vela

IAP 2023

Table of Contents

Goal

Last semester, for the Fall of 2022, I started to investigate ways to learn more about electronics and design, so I decided to build a coilgun. A coilgun uses several stages of sequentially activated coils to launch ferromagnetic projectiles in a particular direction. Specifically, it generates a gradient in the magnetic field strength along the projectile's path such that it applies a force to propel the projectile. A target final velocity for when the projectile leaves the last coil was chosen to be around 100 feet per second, compared to the muzzle velocity of an average Nerf Gun of 70 feet per second.

Initial Design Hardware

The first design uses four stages, each of which provides an extra boost sequentially to the projectile as it moves along a polycarbonate barrel. Each stage consists of the coil being powered along the barrel, a diode to block inductive kickback, a MOSFET to power the coil, an IR beam/sensor to detect the projectile and activate the coil at the appropriate time, and a circuit to amplify the signal from the sensor and use a push-pull amplifier to drive the MOSFETs quickly. The coils are all powered by a supercapacitor bank consisting of 18 2.7V-350F supercapacitors, organized in 3 parallel columns of 6 in series. A 3S Lipo is used to power the MOSFET driving circuit and sensors and to charge the supercapacitor bank when away from a power supply.

Trials with this configuration ended up working with magnets and ferromagnetic projectiles. The exit speeds were in a very safe range, launching the projectiles across rooms with a small but noticeable arc.

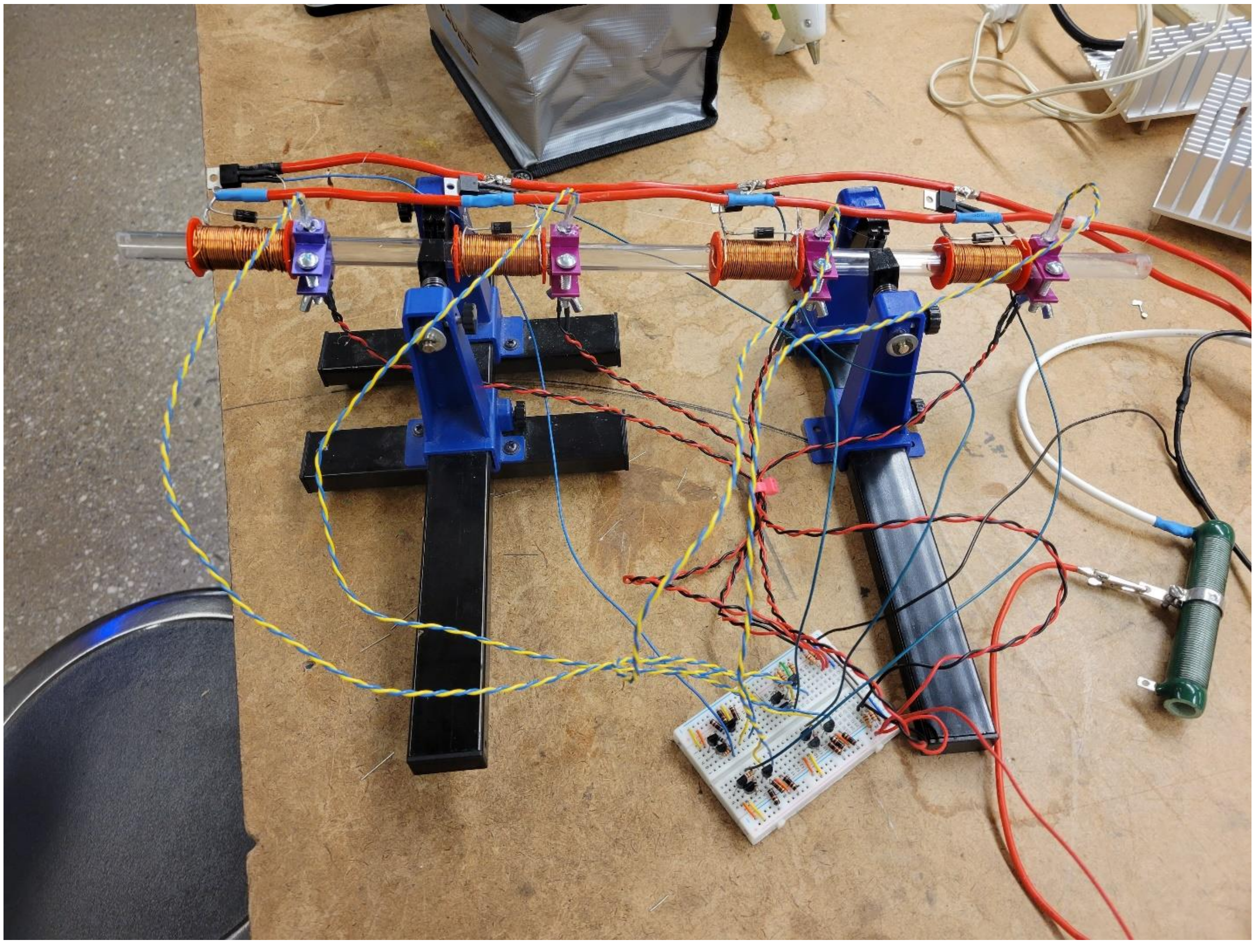

Figure 1: The completed first iteration of the coilgun, including everything but the power supplies, which is out of frame to the right

Initial Circuit

The most complex part of designing a coilgun is the timing circuit and the circuit to power the inductive coils that propel the projectiles. So, we will focus on that part here.

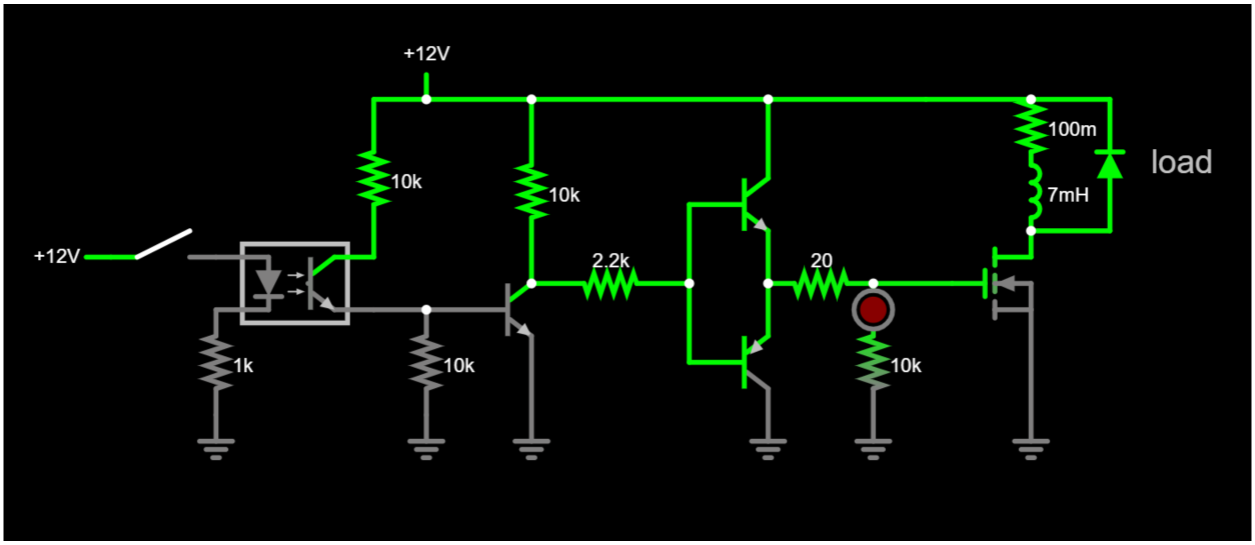

Figure 2: The driving circuit with the MOSFET on

Each stage of the coilgun consists of a circuit controlling the individual coil. The circuit consists of 4 main parts: 1) The infrared beam detection, 2) an amplifier and inverter, 3) a push-pull MOSFET driver, and 4) the MOSFET and load. The first part of the circuit, the infrared beam detection, uses a constantly powered infrared LED pointed through the barrel to an infrared phototransistor. In the idle state, the phototransistor allows some current to flow to the gate of the amplifier transistor, setting the voltage on the push-pull driver to a low state. In the on state, as shown above (with green depicting higher voltages), the beam is blocked (simulated by an off LED), resulting in the amplifier receiving low signal and the driver gate being pulled up, activating the MOSFET. This allows for a simple and faster way to control the coils than using a microcontroller.

In addition, a status led is added connected to the gate of the driver to provide a real-time status of the coil being activated which helps with troubleshooting.

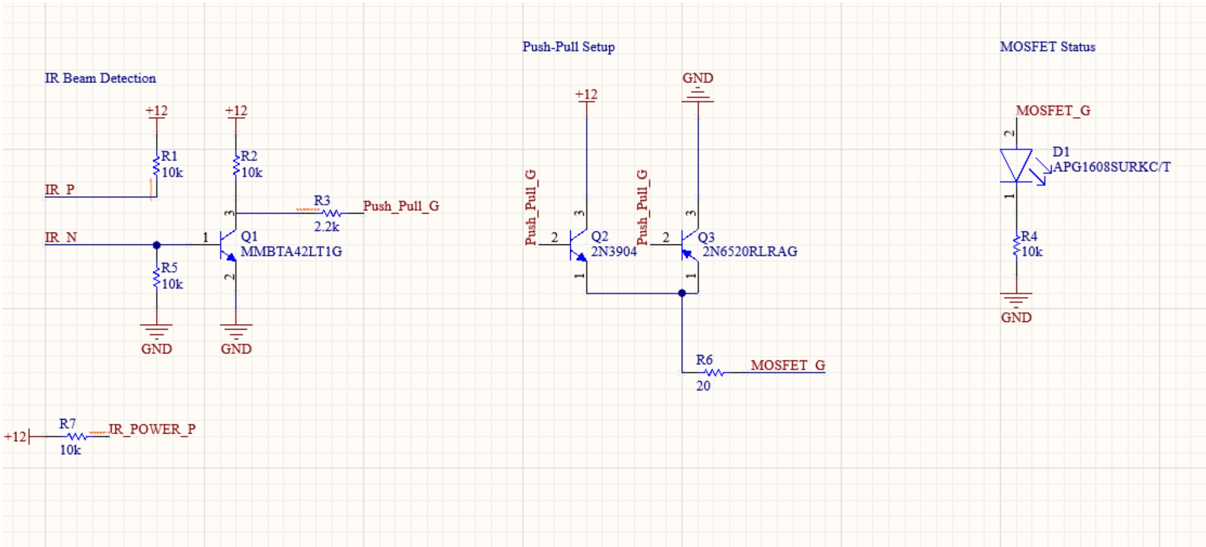

Figure 3: A schematic in Altium depicting the components placed on a PCB for a single stage

Second Idea

During a design review, it was brought up to me by Adi and Fisher that it would remove a lot of the complexity of the MOSFET driver and increase the time constant of current flow through the inductor by using a relay, as when it switches, it closes near instantly, and current flows near-immediately (there is no time constant like the one required to activate the MOSFET GATE). In addition, a MOSFET will no longer need to be mounted on a PCB, which would have been very inefficient due to the amount of current flowing through the relatively small and thin traces. However, with this option, there is one main issue: mainly, there is a small delay (~10 ms) between powering the relay coils and the relay closing for nearly all available consumer relays. This delay becomes significant when the projectile moves at a fast enough speed, covering 1 ft in 10 ms while traveling at 100 feet per second; this must be compensated by firing the coil beforehand. If the delay between the relay coil activation and relay continuity can be measured and is consistent within each relay, then a fixed time to activate the coil can be calculated and deduced by trial and error for the optimal exit velocity of each stage. As a final addition, since this involves fast switching of inductive loads, it is crucial to place a flyback diode across both the relay coil and the coilgun solenoid to prevent damage. It can also be worthwhile to explore solid-state relays, which are very similar to a MOSFET but different in that they include a driver circuit which can allow for more consistency and faster, powerful switching.

Conclusion

The first iteration of the coilgun project was a very worthwhile proof-of-concept, but there are many improvements and optimizations that can be made. I plan to explore the different varieties of relays to see if there is one best suited for fast switching and high peak (and non-constant) current and test it to see if it is viable. In the meantime, to explore PCB design, I plan to finalize the first design by placing the switching logic and amplification components on a printed PCB (a significant upgrade from a breadboard).

As far as optimizations go, one thing to consider is that I believe the force could be dramatically increased by increasing the number of wire loops per coil while still keeping the inductance relatively low. In addition, pairing the infrared phototransistors with an oscilloscope and/or sufficiently fast microcontroller can be helpful to log data from the sensors to get an estimate of the speed of the projectile.

Resources

Thank you Adi and Fisher for the wonderful class and learning experience about the Art and Science of PCB Design, the Edgerton Center for providing resources and materials for the prototype, as well as https://www.falstad.com/circuit/ for providing circuit simulations.