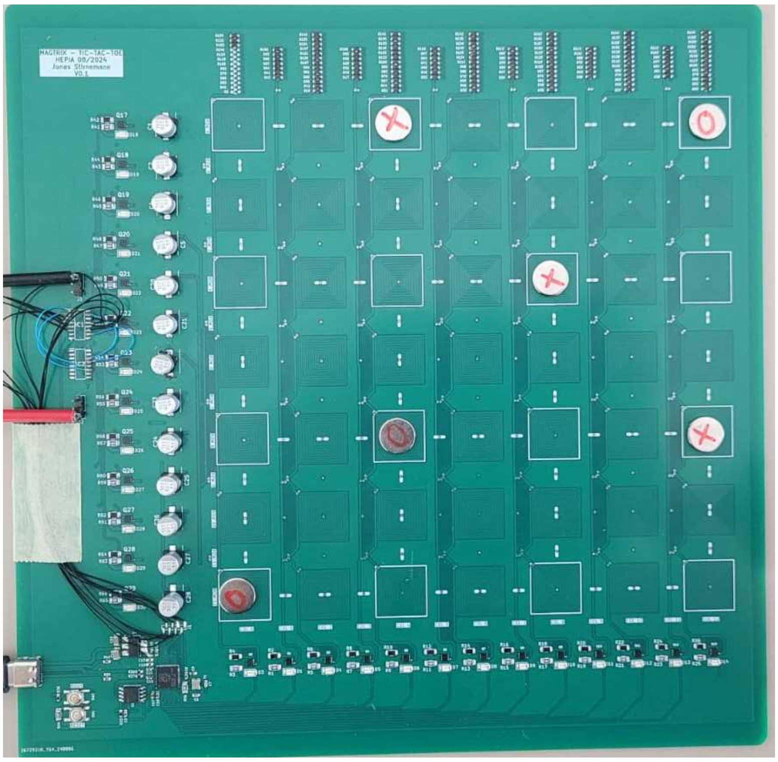
hepia
Haute école du paysage, d'ingénierie et d'architecture de Genève

Hes·so // GENÈVE
Haute Ecole Spécialisée de Suisse occidentale

Magnetic Tic-Tac-Toe



Bachelor Thesis presented by

Jonas STIRNEMANN

Computing and Communication Systems
Embedded systems

September, 2024

Under the guidance of

Fabien Vannel

Mandant

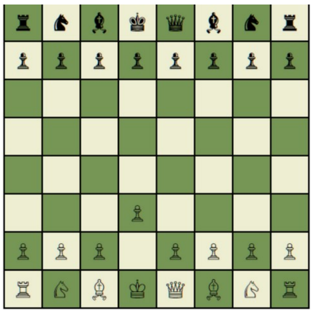
HEPIA

TABLE OF CONTENTS

JEU D'ECHEC

ORIENTATION : SYSTÈMES INFORMATIQUES EMBARQUÉS

Descriptif :
Nous souhaitons réaliser un jeu d'échec physique où les pions se déplacent sur l'échiquier. La technologie à mettre au point se base sur des PCB avec des boucles de courant à même de faire mouvoir des pions aimantés.
L'utilisateur pourra jouer, au travers d'une interface (web, tablette, etc..). L'échiquier, protégé de toute interactions humaines, devra positionner les pions selon la partie en cours



Travail demandé :
Pour ce projet, il est attendu de l'étudiant les étapes suivantes :

- Prendre en main le concept théorique qui permettrait de déplacer un aimant au-dessus d'un PCB plat. Simuler les effets magnétiques
- Imaginer un concept d'échiquier disposant des 64 cases du jeu + les cases nécessaires pour placer les pions « mangé »
- Réaliser l'électronique de contrôle
- Réaliser le logiciel de contrôle avec l'interface humaine
- Tester et valider le fonctionnement
- Documenter

Candidat :

STIRNEMANN JONAS

Filière d'études : ISC

Professeur responsable :

VANNEL FABIEN

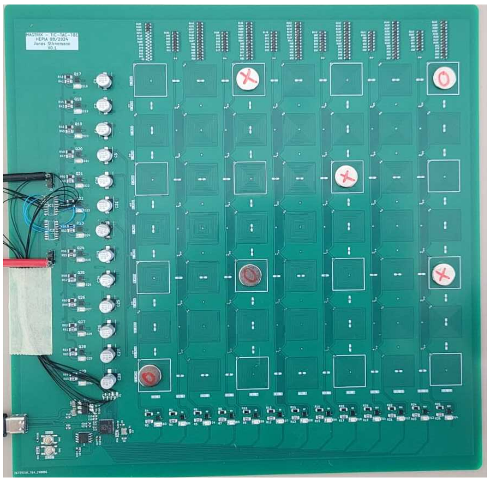
En collaboration avec :

Travail de bachelor soumis à une convention de stage en entreprise : non

Travail de bachelor soumis à un contrat de confidentialité : non

ABSTRACT

When chess enthusiasts want to watch grandmaster games live, they often rely on computer screens. This project aims to bring board games back to the physical world by developing a board that can automatically move pieces. Controlled by a computer, it can be connected to any API, the board can be used for real-time game tracking or play. While this thesis focuses on a Tic-Tac-Toe game, the core technology is adaptable to any board game, with potential for scaling the board size as needed. The project involved designing a PCB, developing firmware, and creating a simple web app. The key innovation is using coils etched onto the PCB to attract pieces embedded with magnets. The process included testing various coil designs to find the most effective one and ensuring scalable control options. The final product is a Tic-Tac-Toe board with a 3x3 playable zone and storage for pieces that have not yet been placed. It also comes with a full web-based interface to control the game via a backend communicating with the board via serial. Despite some minor issues that required manual adjustments, the project demonstrates significant potential for future expansion and application. The actual concept works with a single magnet but when trying to play with multiple bare magnets, the attraction between them is too strong and make the game unplayable since they attract each other. The issue could probably be resolved by using a different pieces design or by making the inter magnet distance bigger.



Candidate:

JONAS STIRNEMANN

Branch : ISC

Professor:

FABIEN VANNEL

In collaboration with:

Thesis subject to an internship agreement: No

Work subject to confidentiality agreement: No

Glossary

GPIO General Purpose Input Output. 24, 31

HEPIA Haute École du Paysage, d'Ingénierie et d'Architecture de Genève. 1, 27

PCB Printed Circuit Board. ii, vii, viii, 1, 2, 8, 9, 10, 11, 12, 13, 14, 15, 24, 25, 26, 27, 28, 29, 30, 31, 33, 36, 37, 38, 39, 40, 41, 51, 53

PWM Pulse Width Modulation. vii, 24, 31, 58

LIST OF FIGURES

URL references

- URL01 https://en.wikipedia.org/wiki/Field_line#/media/File:VFPt_cylindermagnet_-

field-representations.svg

- URL02 https://www.atomic14.com/2022/11/20/simulation-coils.html
- URL03 https://pcbstator.com/wp-content/uploads/2023/09/Stator-Only-5-e1695660921510-1024x1024.png
- URL04 https://pcbstator.com/wp-content/uploads/2023/09/thumbnail-1.png
- URL05 https://hackaday.io/project/154496-2d-actuator-move-micro-robot-in-xy-2d-space
- URL06 https://hackaday.io/project/154496/gallery
- URL07 https://hackaday.io/project/158017-linear-pcb-motor
- URL07 https://www.youtube.com/watch?v=EJZX66JzVDo
- URL08 https://thumbs.dreamstime.com/b/curl-right-hand-rule-vector-illustration-example-diagram-detecting-direction-induced-current-flow-magnetic-field-physics-173736679.jpg
- URL09 https://www.supermagnete.ch/

LIST OF APPENDICES

INTRODUCTION

As the final step in our Bachelor's program in Computing and Communication Systems at Haute École du Paysage, d'Ingénierie et d'Architecture de Genève (HEPIA), we undertake a Bachelor project that spans 14 weeks immediately following our last semester. It extends for a period of 14 weeks during which a workload of 450 hours is expected. This project is the culmination of our studies and is intended to demonstrate our ability to apply the knowledge and skills acquired during our studies to a concrete problem.

Our initial project concept was to design and implement a PCB-based chessboard, where chess pieces are moved using magnetic fields generated by electromagnets directly etched onto the PCB. However, due to the complexity involved in implementing, producing, shipping, and using a chessboard of this nature, we decided to shift our focus to another board game—Tic-Tac-Toe—that shares similar concepts but offers a more manageable form factor. Should there be a need, the project can be scaled up to a chessboard with minimal adjustments, extending the board and reprogramming the game rules.

Tic-Tac-Toe is a simple game played on a 3x3 grid, where two players take turns placing their pieces on the board. The objective is to be the first to align three pieces in a row, column, or diagonal. Our implementation of this game eliminates the need for physical interaction with the board. The pieces, embedded with magnets, are moved by electromagnets that are etched onto the PCB.

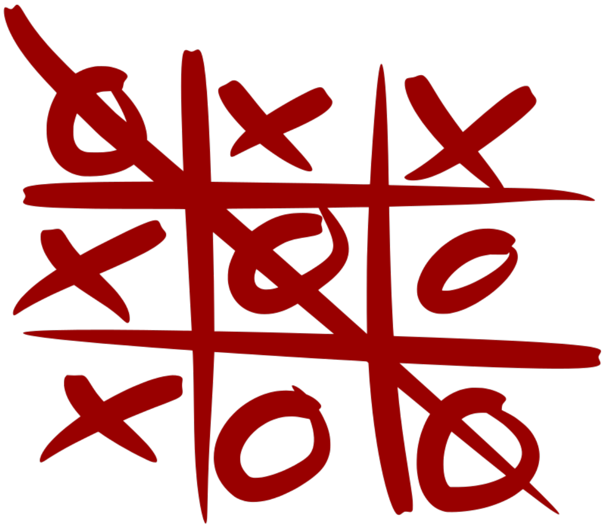


FIGURE 1: Tic Tac Toe example

Players interact solely through a web application, which displays the game board and enables them to make their moves. This application communicates with a backend, which in turn connects to a microcontroller responsible for controlling the electromagnets. The microcontroller activates the coils in a specific sequence to move the pieces on the board.

The project was chosen because it combines several aspects of our studies, such as electronics, software development and microcontroller programming. It also allows us to work on a project from start to finish, from the design phase to the implementation and testing phase. The project is also interesting because it involves the use of different technologies as PCB design, microcontroller programming, and web development.

The primary objective of this project is to design and implement a functional, automated Tic-Tac-Toe board that utilizes electromagnetism to move the pieces, completely eliminating the need for manual interaction. While the project focuses on a simple game like Tic-Tac-Toe, it serves as a proof of concept for more complex games such as chess. In a chess setting, this technology could enable spectators to watch a grandmaster's game with the moves being replicated in real-time on a physical board, without anyone needing to physically move the pieces. Or it could be used for educational purposes, allowing students to play against a computer that moves the pieces for them.

This thesis is organized into three main sections, each addressing a distinct aspect of the project. The first section explores the theoretical concepts of magnetism and electromagnetism, along with simulations conducted to validate the concept. The second section focuses on the iterative design process of the hardware to identify the optimal coil design. The third section details the final hardware design and the software components of the project.

Chapter 1 : Technical concepts

1.1. Magnetism

Magnetism[1] is a physical phenomenon produced by the motion of electric charge, resulting in attractive and repulsive forces between objects. The motion of electric charge creates a magnetic field, which exerts a force on other moving charges.

The concepts of magnetism and electromagnetism are essential to the project. The board's design must take into account the magnetic fields generated by the coils and the magnets' attraction to these fields. For that we need to understand the basics of magnetism and electromagnetism. Ampere's Law for example states that the line integral of the magnetic field **B** around a closed loop is proportional to the total electric current I passing through the loop. Mathematically, it is expressed as:

$$\oint_{\mathscr{C}} \mathbf{B} \cdot d\mathbf{l} = \mu_0 I_{\text{enc}}$$

where:

- $\oint_{\mathscr{C}} \mathbf{B} \cdot d\mathbf{l}$ is the line integral of the magnetic field around a closed path $\mathscr{C}$,
- μ_0 is the permeability of free space ($\mu_0 \approx 4\pi \times 10^{-7}\,\mathrm{T \cdot m/A}$),
- I_{enc} is the total current enclosed by the path (in Amperes) $\mathscr{C}$.

[1] **noauthor_nasa_magnetism**.

a. Direction of the magnetic field

The direction of the current passing through a wire determines the direction of the magnetic field around the wire. The right-hand rule is a common way to determine the direction of the magnetic field around a current-carrying wire. The rule states that if the thumb of the right hand points in the direction of the current, the fingers will curl in the direction of the magnetic field around the wire.

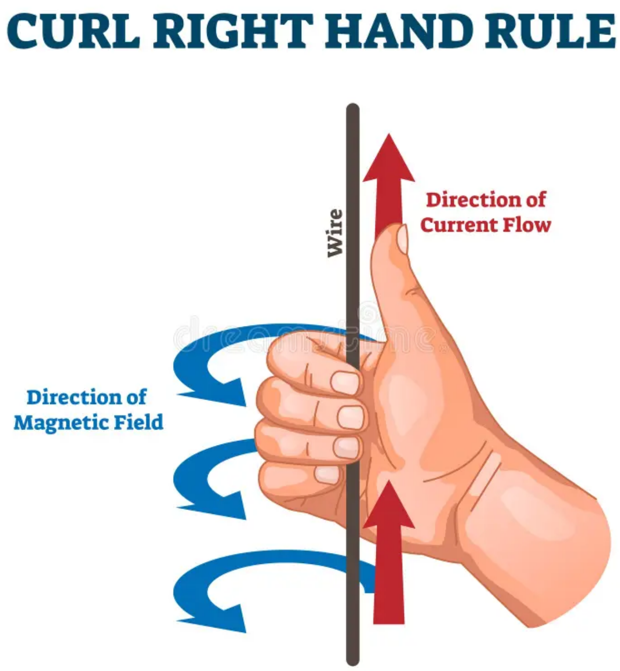


FIGURE 1.1: Right hand rule Source: thumbs.dreamstime.com ref: URL08

1.2. PERMANANT MAGNETS

Permanent magnets are materials that produce a magnetic field without the need for an external magnetic field. They are made of ferromagnetic materials such as iron, nickel, and cobalt. The magnetic field is created by the alignment of the magnetic domains in the material. The domains align in the same direction, creating a net magnetic field. The strength of the magnetic field depends on the material and the alignment of the domains.

The strength of the magnetic field is measured in units of Tesla (T) or Gauss (G). Stacking magnets can increase the magnetic field strength[2]. For example, stacking flat disc magnets

[2] *Factors influencing adhesive force of magnets*. URL: `https://www.supermagnete.ch/eng/faq/What-`

multiplies the adhesive force until the stack reaches a height of half the diameter of a single disc magnet at a maximum.

For testing purposes, we tried multiple magnets of different sizes from Super Magnet Shop[3]. The magnets come with a datasheet that give us the magnetic field strength of the magnets as Residual magnetism Br 13200-13700 G, 1.32-1.37 T. This means for example that the magnetic field strength of this specific magnet is between 1.32 and 1.37 Tesla.

They also give us the polarization pattern of the magnet. For example the disk magnet has its poles on the flat faces of the disk.

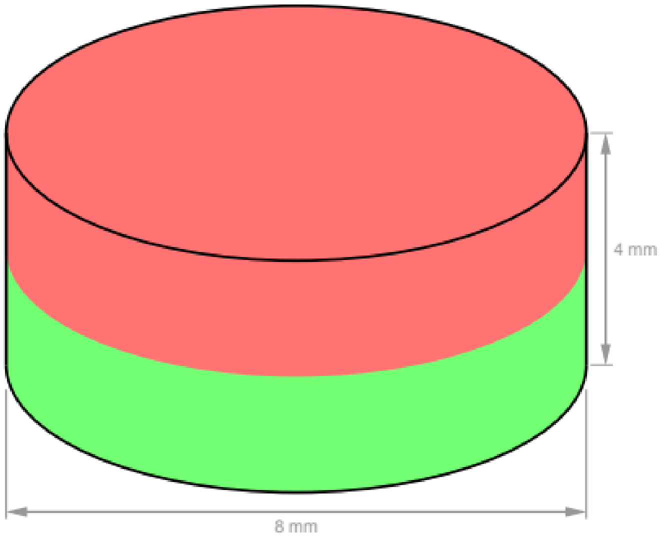


FIGURE 1.2: Disk magnet polarity Source: www.supermagnete.ch ref: URL09

factors-affect-the-adhesive-force-and-strength-of-a-magnet (visited on 08/20/2024).

[3] *Buy strong magnets online - supermagnete.ch.* URL: https://www.supermagnete.ch/eng/ (visited on 08/20/2024).

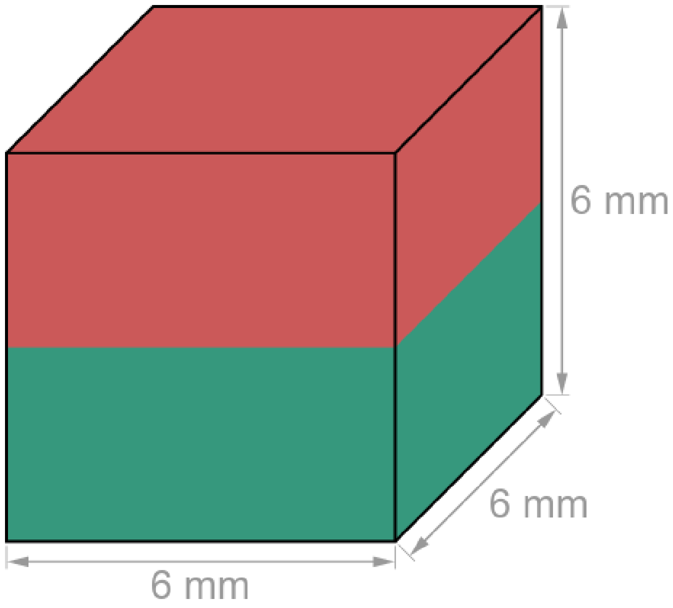


FIGURE 1.3: Cube magnet polarity Source: www.supermagnete.ch ref: URL09

1.3. CALCULATE THE FORCE OF A MAGNETIC FIELD ON A MAGNET

To calculate the force exerted on a magnet by a magnetic field (for example from a pcb coil), taking into account friction and the magnet's mass, we can use the following formulas:

Calculate the Magnetic Force

$$F_{\text{mag}} = m \cdot B \cdot \sin(\theta)$$

Calculate the Friction Force

$$F_{\text{friction}} = \mu \cdot m_{\text{weight}} \cdot g$$

Calculate the Total Force

$$F_{\text{total}} = F_{\text{mag}} - F_{\text{friction}}$$

Chapter 2 : State of the art

We want to use PCB coils to move a magnet directly via the magnetic field generated by the coil. This concept is not new and has been used in some motor applications for example. There also is some people that use this concept for learning or entertainement purposes. This helped us to get some ideas on how to design our own system.

2.1. PCB Motors

There has been a huge interest in the development of PCB motors. These motors are made by etching a coil pattern on a PCB to act as the stator. There is a center shaft attached to magnets. When powering the coils in a specific sequence, the magnets will be attracted to the coils and the motor will rotate.

Figure 2.1: Atomic14 PCB stators Source: https://www.atomic14.com ref: URL02

There also is some commercial products that use this concept. They claim to be more efficient, Lighter with higher torque density. For example[4] sells services and production to design your own PCB motor with specified specs, and they use PCB stators.

[4] *PCB Stator*. en-US. URL: `https://pcbstator.com/` (visited on 08/18/2024).

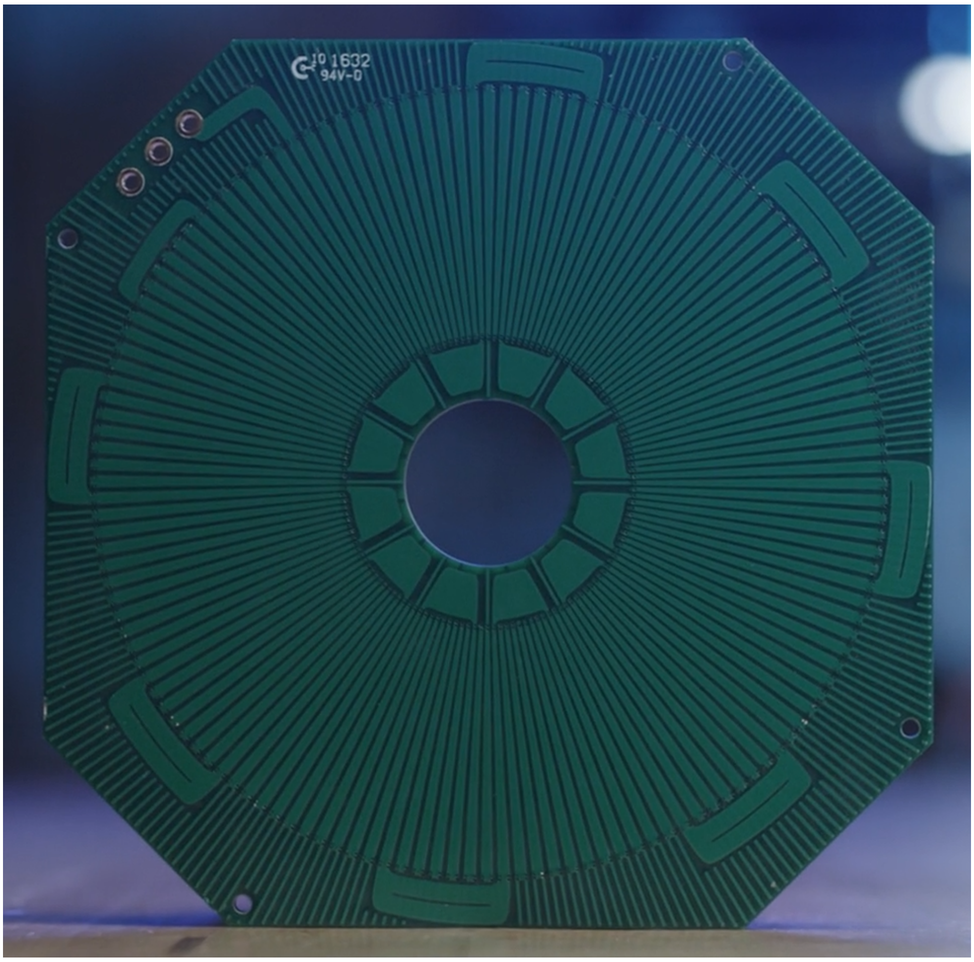


FIGURE 2.2: PCB stator from EMC Source: https://pcbstator.com/ ref: URL03

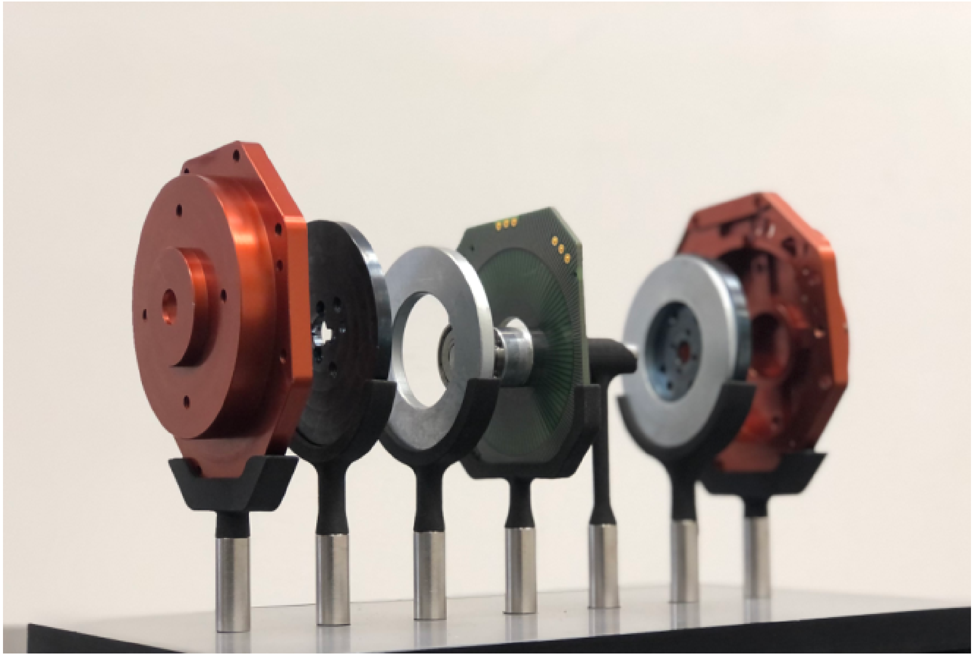


FIGURE 2.3: PCB Stator's full motor design Source: https://pcbstator.com/ ref: URL04

2.2. PCB 2D LINEAR ACTUATORS

There are some people that have used the same concept to make a linear actuator. They use linear PCB traces to move a magnet in a 2D linear fashion. There is for example a project on hackaday[5] that does just that. They actually use 3 different for each axis. They alternate those 3 coils made with PCB traces to move the magnet in the desired direction.

The concept seems to have been developped and tested by SRI[6] in 2014. They used a similar concept to move a micro robot in a 2D space.

[5] *2D stepper motor etched on PCB - micro manipulator*. en. URL: `https://hackaday.io/project/164507-2d-stepper-motor-etched-on-pcb-micro-manipulator` (visited on 08/18/2024).

[6] SRI. *Magnetically Actuated Micro-Robots for Advanced Manipulation Applications*. 2014. URL: `https://www.youtube.com/watch?v=uL6e3co4Qqc` (visited on 08/18/2024).

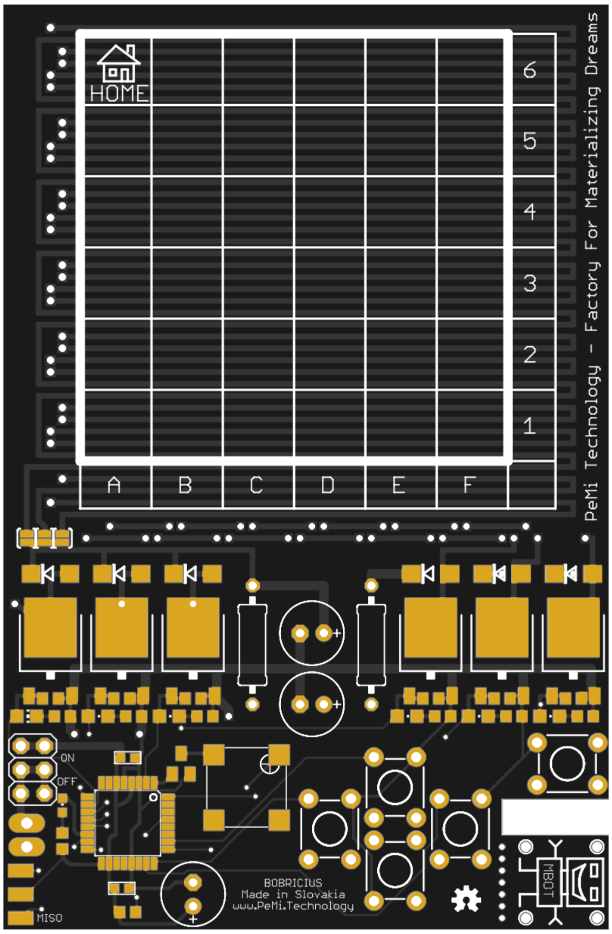


FIGURE 2.4: PCB move a magnet in X/Y axis Source: hackaday.io ref: URL06

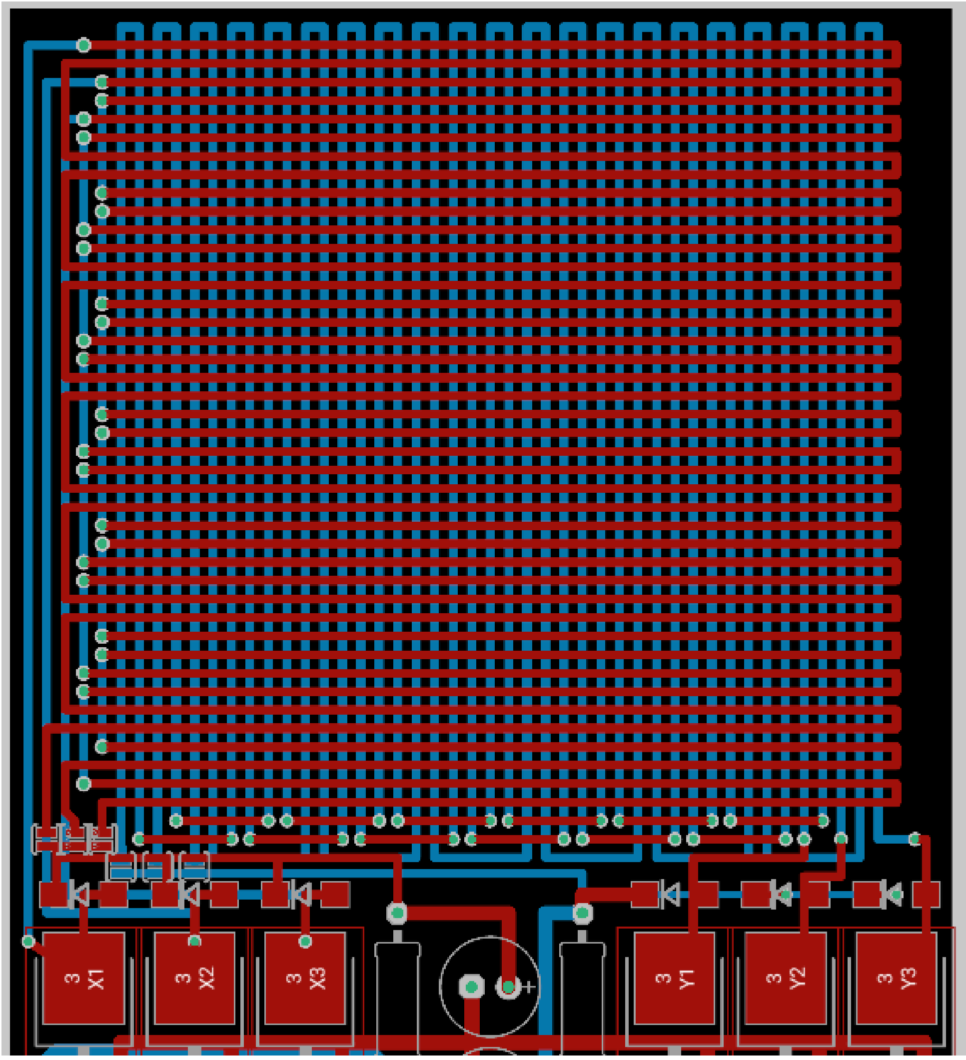


FIGURE 2.5: PCB design with linear coils Source: hackaday.io ref: URL06

We can see three different traces running on all the PCB and alternating. There is three of these traces on the TOP layer of the PCB and three on the BOTTOM layer. The magnet is moved by passing current through each coil one by one. The direction of the magnet is determined by the order of activation of the coils. So by activating the coils one then two then three, the magnet will move in one direction. By activating the coils three then two then one, the magnet will move in the opposite direction.

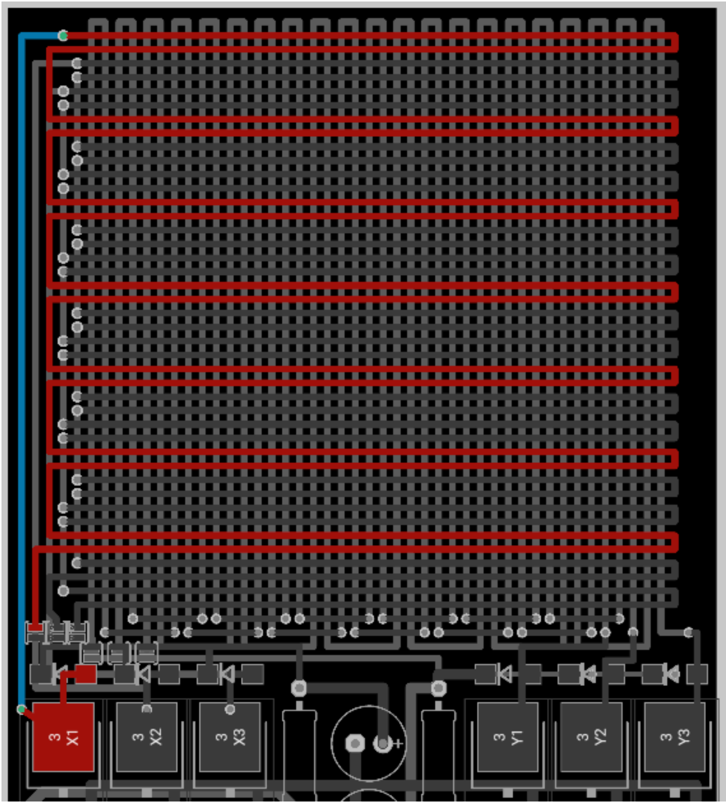


FIGURE 2.6: PCB design with horizontal coils

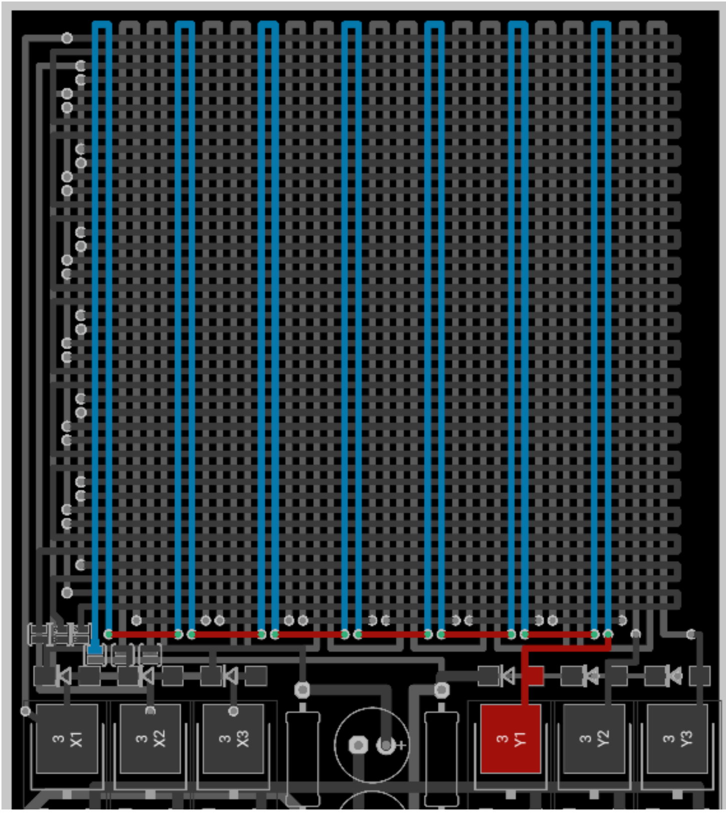


FIGURE 2.7: PCB design with vertical coils

2.3. PCB COILS

Some projects use PCB traces to draw arbitrary shaped coils like spirals or squares to generate a magnetic field. They can increase the torque by increasing the number of turns or current capacity (Bigger traces means less resistance, which means more current thus more magnetic field). They can also enhance the magnetic field by stacking multiple coils on the different PCB layers.

Unlike the pcb stator motors where the magnets are on a free rotating shaft, the coils here have to generate enough force to attract the magnet while countering the friction between the magnet and the PCB.

Carl Bugeja[7] has made a project where he uses a spiral coil to move a magnet in a linear path, he actually the design with FR4[8] basic PCB but also with flexible PCB. He made a small video[9] of the magnet moving on the PCB. He mostly uses a Ball magnet to help reduce the

[7] *Carl Bugeja*. en-US. URL: `https://www.carlbugeja.com` (visited on 08/19/2024).

[8] *FR-4*. en. Page Version ID: 1218528760. Apr. 2024. URL: `https://en.wikipedia.org/w/index.php?title=FR-4&oldid=1218528760` (visited on 08/19/2024).

[9] Carl Bugeja. *Actuating Magnets with PCBs*. 2018. URL: `https://www.youtube.com/watch?v=`

friction between the magnet and the PCB.

FIGURE 2.8: PCB design of spiraling coils that form a linear path for a magnet Source: hackaday.io ref: URL07

LudKMv7M13Q (visited on 08/19/2024).

CHAPTER 3 : SIMULATIONS

In this chapter, we will present the different simulations that were done to test the directivity and force of different coil shapes. There's is a wide variety diferrent software that can be used to simulate magnetic fields. I found Python scripting to be more versatile and easier to get the results I wanted.

The goal of these simulations is to determine if one shape is better than an other and if the magnetic field is strong enough to move a magnet. The simulations only take into account a fixed current of 1A and a number of turns in a determined area. We only really want the magnetic field in the X and Y direction as we will only move the magnet in a 2D space. But we should keep in mind that the Z direction might add some friction.

We'll do out calculations with a 6x6x6 cubic magnet with a magnetic field strength of 1.29T and a magnet weight of 1.6416 grams. We'll consider a friction coefficient of 0.2 for the magnet on the surface of the PCB.

In order to calculate the force of the magnetic field on the magnet, we'll need to compute the magnetic moment of a specific magnet, we need to know the volume of the magnet and the residual magnetism.

- Residual magnetism: $B_r = 1.29\,\mathrm{T}$
- Side length of the cubic magnet: $6\,\mathrm{mm} = 0.006\,\mathrm{m}$

The volume V of the cubic magnet is calculated as:

$$V = (\text{Side length})^3 = (0.006\,\mathrm{m})^3 = 2.16 \times 10^{-7}\,\mathrm{m}^3$$

The magnetic moment m is given by:

$$m = B_r \cdot V$$

$$m = 1.29\,\mathrm{T} \times 2.16 \times 10^{-7}\,\mathrm{m}^3$$

$$m = 2.78 \times 10^{-7}\,\mathrm{A} \cdot \mathrm{m}^2$$

3.1. LINEAR COIL

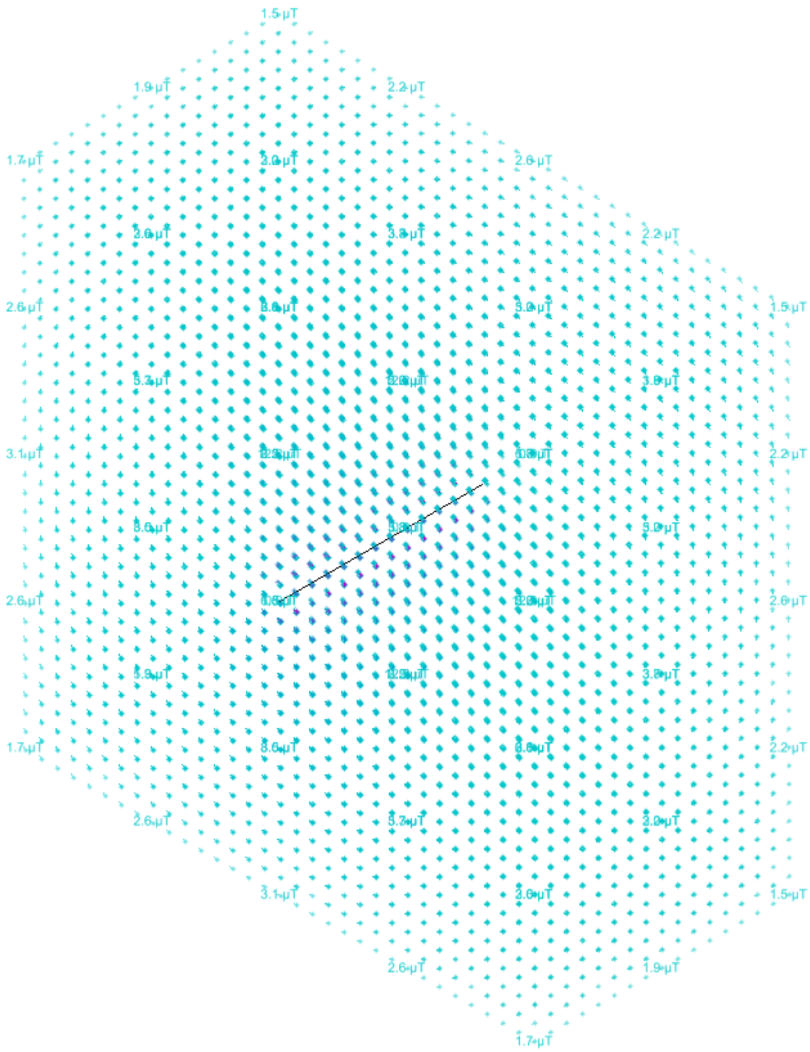


FIGURE 3.1: Simulation of a simple wire in a 10mm x 10mm x 5mm zone, with a current of 1A

As we can observe, the magnetic field is weak and gets weaker as we move away from the wire. We'll take a middle value of 8 uT at around 3mm from the wire.

Given data:

- Magnetic flux density of the external field: $B = 8\,\mu\text{T} = 8 \times 10^{-6}$ T
- Magnetic moment: $m = 2.78 \times 10^{-7}$ A $\cdot$ m^2
- Magnet weight: $m_{\text{weight}} = 1.6416$ g $= 0.0016416$ kg

- Coefficient of friction: $\mu = 0.2$
- Acceleration due to gravity: $g = 9.81$ m/s^2
- Angle between magnetic moment and external field: $\theta = 90°$, so $\sin(\theta) = 1$

Calculate the Magnetic Force

$$F_{\text{mag}} = m \cdot B \cdot \sin(\theta)$$

$$F_{\text{mag}} = 2.78 \times 10^{-7}\,\text{A} \cdot \text{m}^2 \times 8 \times 10^{-6}\,\text{T}$$

$$F_{\text{mag}} = 2.22 \times 10^{-12}\ \text{N}$$

Calculate the Friction Force

$$F_{\text{friction}} = \mu \cdot m_{\text{weight}} \cdot g$$

$$F_{\text{friction}} = 0.2 \times 0.0016416\ \text{kg} \times 9.81\ \text{m/s}^2$$

$$F_{\text{friction}} \approx 0.0032\ \text{N}$$

Calculate the Total Force

$$F_{\text{total}} = F_{\text{mag}} - F_{\text{friction}}$$

$$F_{\text{total}} = 2.22 \times 10^{-12}\ \text{N} - 0.0032\ \text{N}$$

$$F_{\text{total}} \approx -0.0032\ \text{N}$$

3.2. CIRCULAR OR SPIRAL COIL

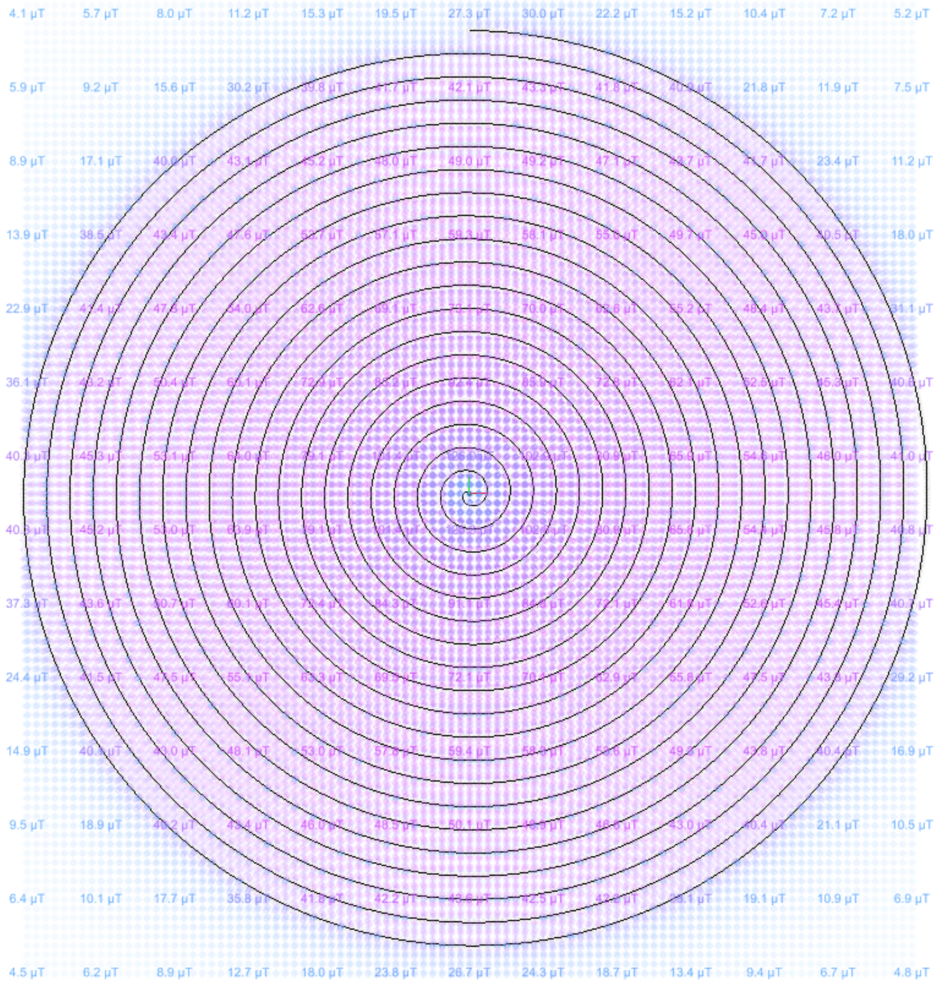


FIGURE 3.2: Simulation the magnetic field in a spiral coil with 20 turns for a total of 10mm diameter, with a current of 1A

We'll take a middle value of 50 uT at around 3mm from the wire.

Given data:

- Magnetic flux density of the external field: $B = 50\,\mu\text{T} = 50 \times 10^{-6}$ T
- Magnetic moment: $m = 2.78 \times 10^{-7}$ A $\cdot$ m^2
- Magnet weight: $m_{\text{weight}} = 1.6416$ g $= 0.0016416$ kg
- Coefficient of friction: $\mu = 0.2$

- Acceleration due to gravity: $g = 9.81$ m/s^2
- Angle between magnetic moment and external field: $\theta = 90^\circ$, so $\sin(\theta) = 1$

Calculate the Magnetic Force

$$F_{\text{mag}} = m \cdot B \cdot \sin(\theta)$$

$$F_{\text{mag}} = 2.78 \times 10^{-7}\,\text{A} \cdot \text{m}^2 \times 50 \times 10^{-6}\,\text{T}$$

$$F_{\text{mag}} = 1.39 \times 10^{-10}\ \text{N}$$

Calculate the Friction Force

$$F_{\text{friction}} = \mu \cdot m_{\text{weight}} \cdot g$$

$$F_{\text{friction}} = 0.2 \times 0.0016416\ \text{kg} \times 9.81\ \text{m/s}^2$$

$$F_{\text{friction}} \approx 0.0032\ \text{N}$$

Calculate the Total Force

$$F_{\text{total}} = F_{\text{mag}} - F_{\text{friction}}$$

$$F_{\text{total}} = 1.39 \times 10^{-10}\ \text{N} - 0.0032\ \text{N}$$

$$F_{\text{total}} \approx -0.0032\ \text{N}$$

This means that the magnet is too heavy to move from the middle of the coil. We'll have to increase the magnetic field strength to move the magnet.

3.3. RECTANGULAR COIL

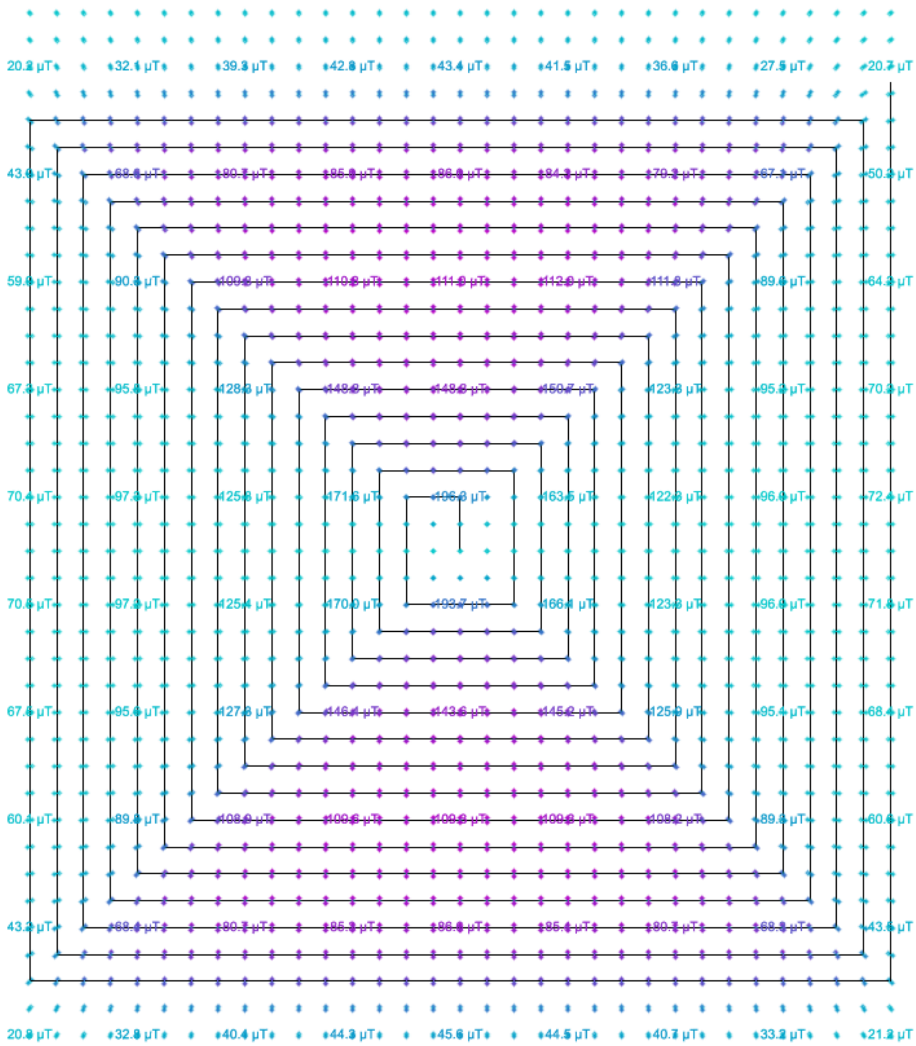


FIGURE 3.3: Simulation of the magnetic field of a square coil with 15 turns for an area of 8mm x 8mm, with a current of 1A

The forces applied on the magnet seem really weak, i might have issues with my calculations or the friction coefficient. The simulation are not showing what i can really see in real life.

CHAPTER 4 : TESTING SOME COIL DESIGNS

We will now test several of the designs discussed in the previous chapter by creating PCBs incorporating these coil designs. Additionally, we will include a driver and control system to regulate the current flowing through the coils.

4.1. EXISTING DESIGNS

To determine the optimal solution for our application, we have tested various existing designs that use PCB coils to move a magnet. For all these tests, we will utilize a driving circuit and a microcontroller mounted on a breadboard.

a. Driver circuit

In order to control the current in the coils, we will need some driving circuitry. We will use simple NMOS transistors. Leds were added to visually see which coil is activated. The schematics of the driver circuit is shown in figure 4.1 and the mounted circuit is shown in figure 4.2. Since these coils are inductive, flyback diodes were added to protect the transistors and the power supply.

When setting the GPIOs to a logic high, the transistor will conduct and the current will flow from the power supply to the coil. When setting the GPIOs to a logic low, the transistor will stop conducting.

The same circuit will be used for testing two different existing coil designs.

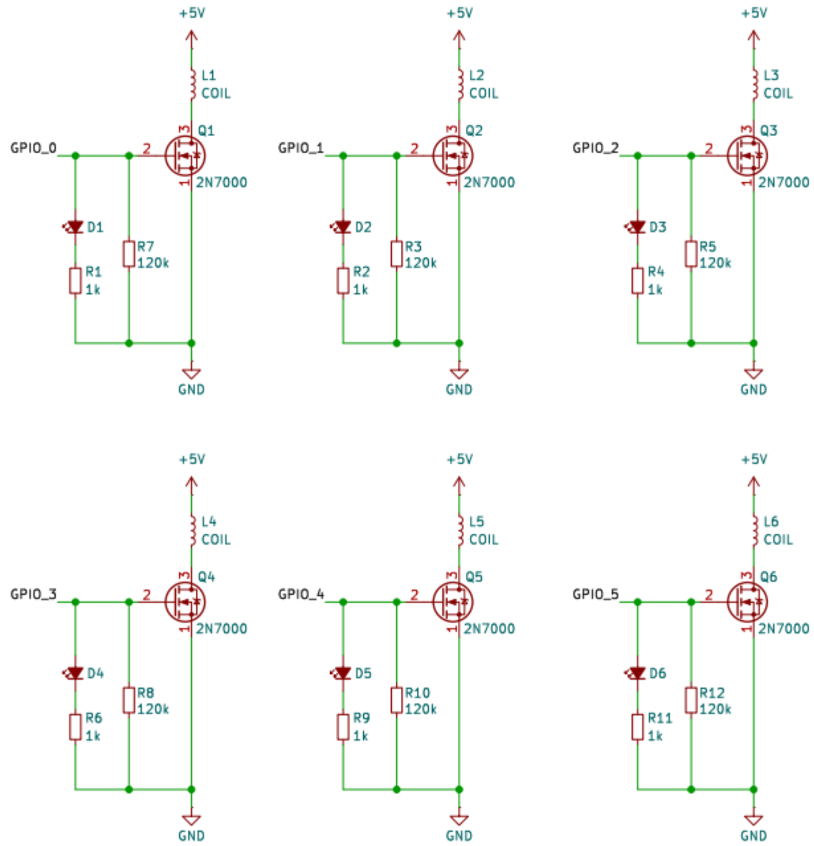


FIGURE 4.1: Schematics of the driver breadboard circuit to control the coils

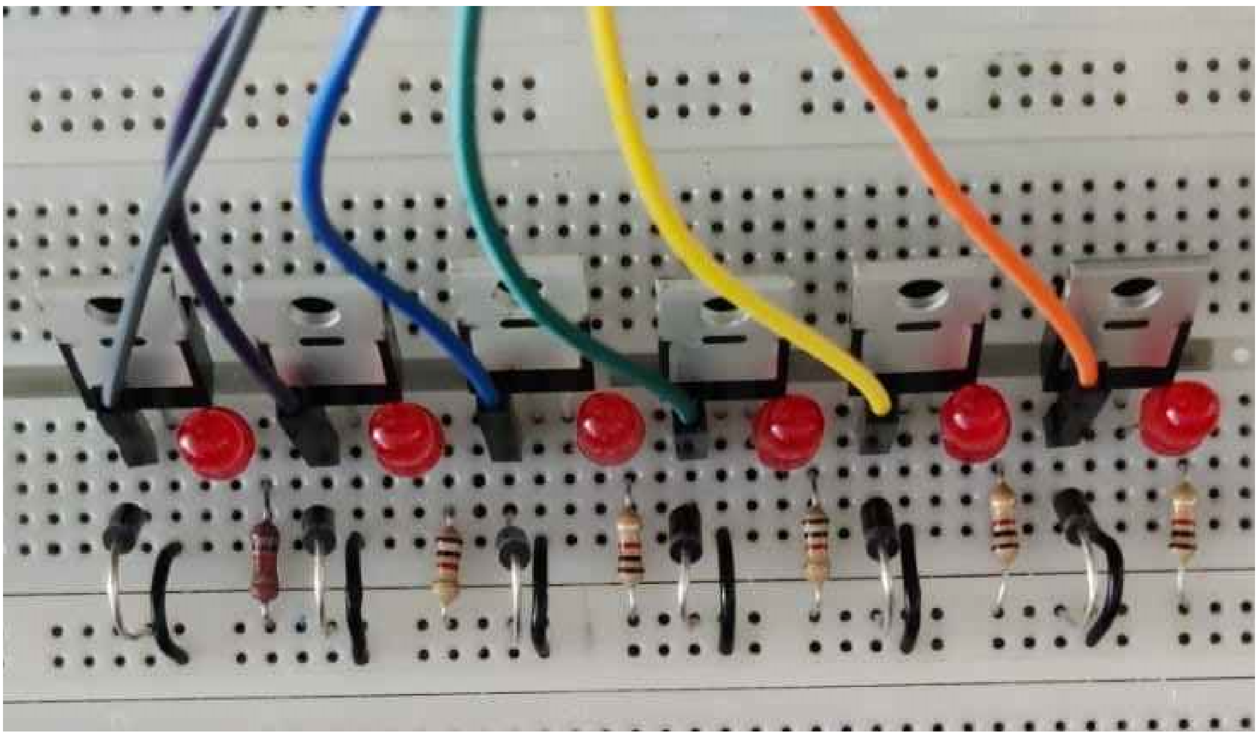


FIGURE 4.2: Mounted Breadboard circuit

b. Control firmware

For this tests, the coils are controlled via a Raspberry Pi PICO[10] programmed in MicroPython and six General Purpose Input Output (GPIO)s are connected to the transistors and will be used to control the current in the coils. The code is shown in figure 4.3.

```python
import digitalio
import board
import time
import pwmio

pins = [board.GP0, board.GP1, board.GP2, board.GP3, board.GP4, board.GP5, ]
coils = [pwmio.PWMOut(pin, frequency=50000) for pin in pins]

for coil in coils:
    coil.duty_cycle = 0

print("STARTING TEST")

while True:
    for i in range(3):
        coils[i].duty_cycle = 65000
        print(f"Coil {i} ON")
        time.sleep(1)

        coils[i].duty_cycle = 0
        print(f"Coil {i} OFF")
        time.sleep(1)
```

FIGURE 4.3: Micropython code to control the coils with a PWM

c. Pemi Technology linear coils

The first one comes from Pemi Technology[11] and uses the alternating linear coils. We'll use the driving circuit to control the current in the coils. There is two 4 pin connectors on the PCB, one connector for each (X / Y) axis. One of these pins is common to the three coils and will be

[10] Raspberry Pi Ltd. *Buy a Raspberry Pi Pico*. en-GB. URL: https://www.raspberrypi.com/products/raspberry-pi-pico/ (visited on 08/19/2024).
[11] *Home - PeMi Technology - Factory For Materializing Dreams*. URL: http://www.pemi.technology/ (visited on 08/19/2024).

connected to the power supply, the other three pins are connected to the other side of the coils and will be pulled to ground by transistors to activate the coils.

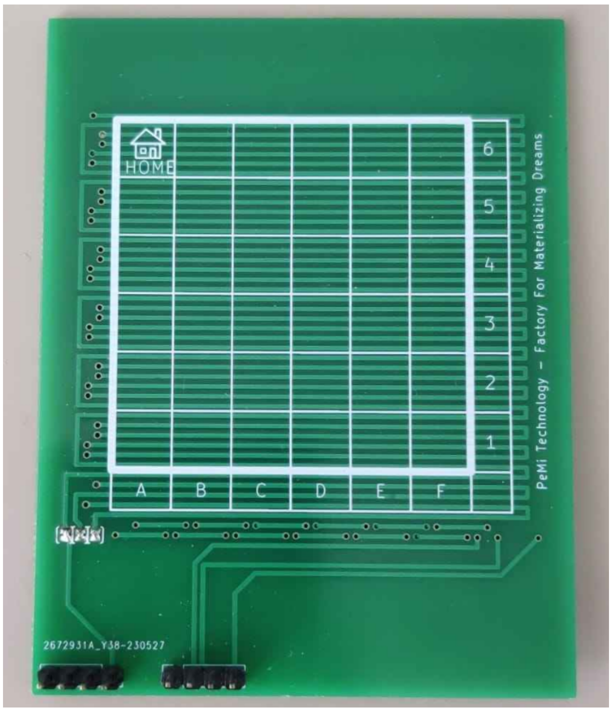


FIGURE 4.4: Pemi Technology linear coils PCB

The PCB characteristics are:

- 3 coils for each axis
- 10 mils (0.254mm) spacing between each coil
- 20 mils (0.508mm) width for the traces

The magnet used for the tests:

- 3mmx3mmx3mm cubic magnet
- 5mmx5mmx5mm cubic magnet
- 2mm diamter round magnet
- 2mmx2mmx1mm cubic magnet

The tests consist on alternating the activation of the coils and see if we could move a magnets. We used different magnet sizes and shapes to see if we could move them.

We could only move the smallest magnets (2mmx2mm) and only in one axis. The direction was managed by the order of activation of the coils, and changing this order did change the diection. There still was some moments where the magnet would be stuck or moving back and forth on the same three coils.

The axis where the magnet was moving was the one on the top layer of the PCB, this means that the coils on the bottom layer were not strong enough to move the magnet bacause they were further to the magnet than the top layer ones.

d. HEPIA linear coils

In a previous intership[12], a PCB was designed with the same linear coils concept. It's actually compose of 4 identical board, each with 9x9 cell. These cells should be individually controllable to move a magnet in each sub surface. The PCB has some issues, notably all the columns are connected together but should be individually controllable. This renders the final PCB unusable for our application.

Besides the electrical issues, the PCB is made on 6 layers with bind vias, making it very hard to develop, expensive to produce and really hard to debug.

- 9x9 cell
- 6 layer board
- Bind vias
- 3 coils for each axis
- 0.3mm spacing between each coil
- 0.5mm width for the traces

[12] *arthur.abelkalo / ChessGame · GitLab*. en. Aug. 2023. URL: `https://githepia.hesge.ch/arthur.abelkalo/chessgame` (visited on 08/19/2024).

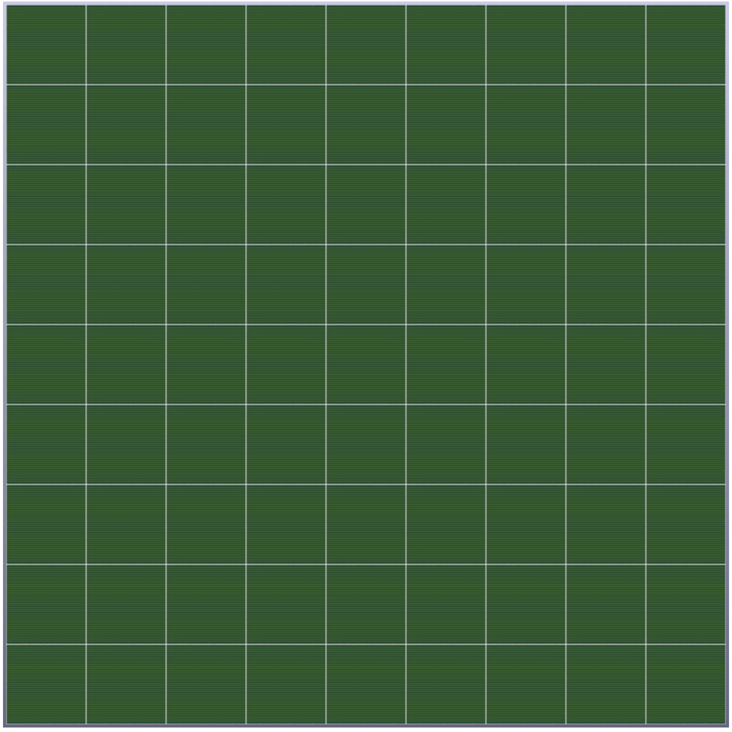


FIGURE 4.5: HEPIA Intern's PCB with linear coils

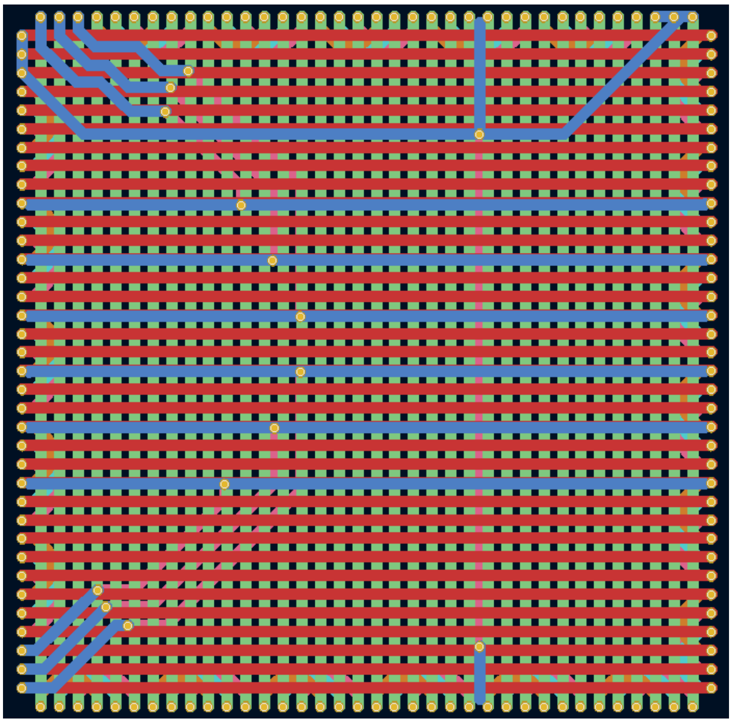


FIGURE 4.6: Single cell of the intern's PCB with all layers

4.2. CUSTOM PCB TO TEST DESIGNS

In order to test some coil design and to have a better control over the coils while also testing some ways to control the current in the coils, we designed a custom PCB with 4 different coil designs. Two of these designs are still linear coils but with different spacing and width. The other two designs are circular coils with different trace width and spacing to see if more turns or more current impacts the magnetic field while keeping the same diameter.

We also wanted to keep the PCB as 2 layers to keep the cost down and see if it was enough to move a magnet.

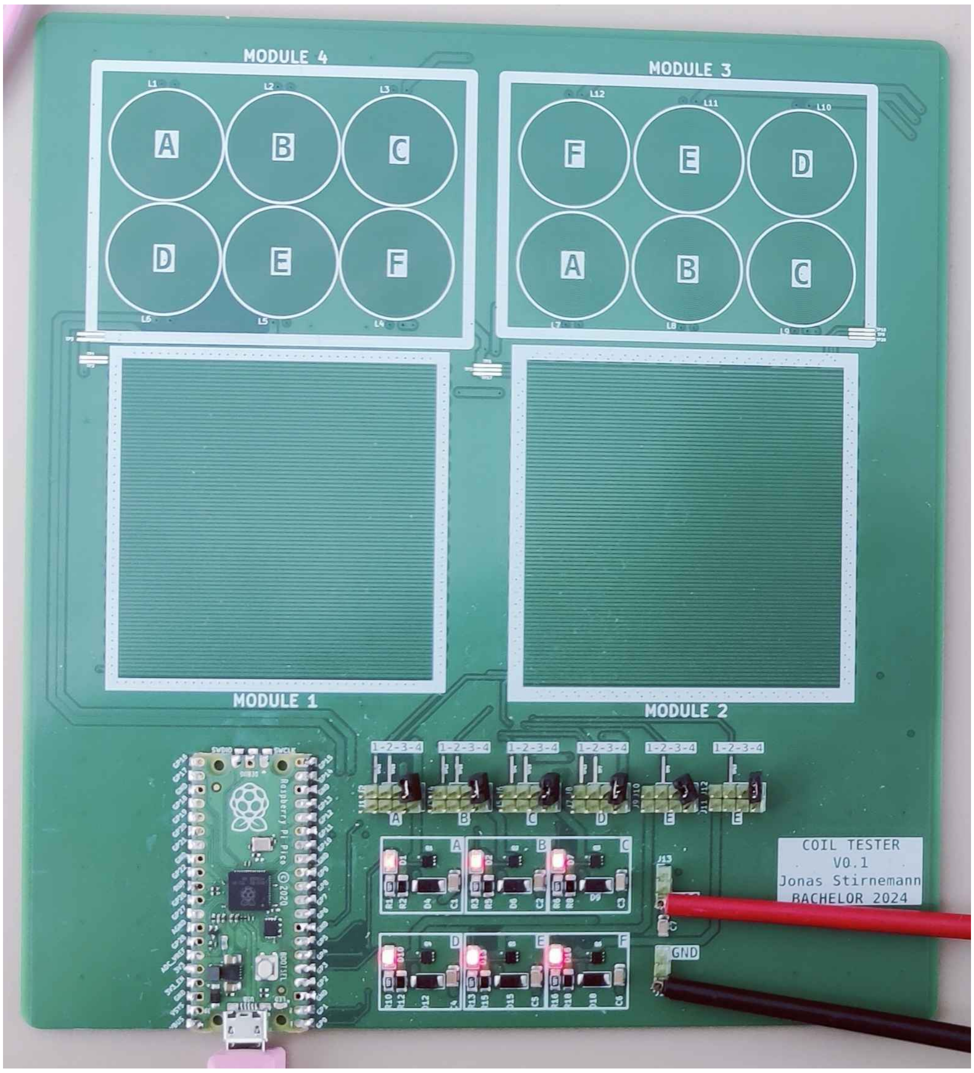


FIGURE 4.7: Mounted custom test PCB

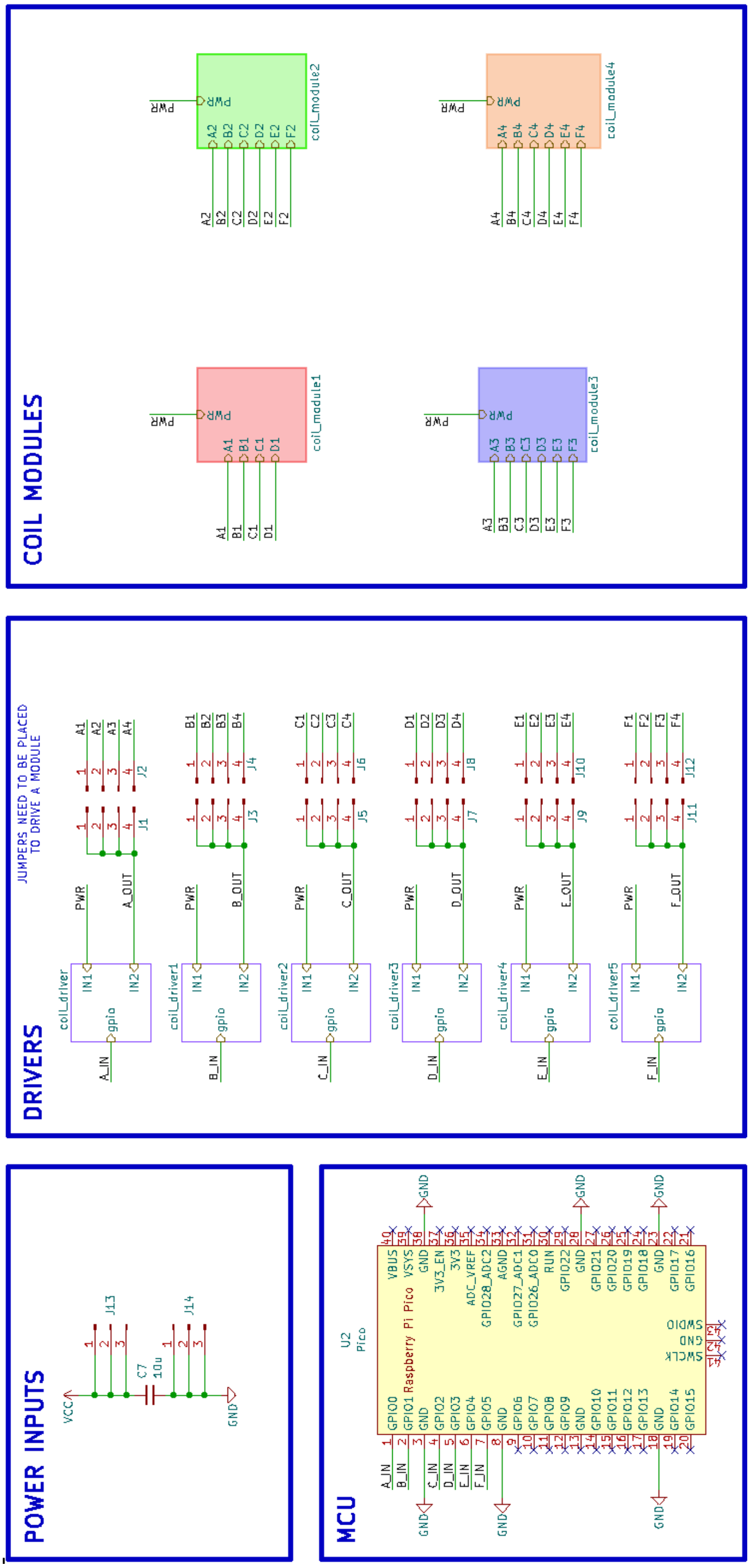


FIGURE 4.8: Custom test PCB general schematics

a. Driver circuitry

We would only test one design at the time, so we would only need one driver circuit and we would want to change which coil design is connected to it. For that we simply added pin aligned pin headers so that we can physically connect which design is connected to the driving circuit. One other advantage of this method, is that we can also easily connect any external design on the driving circuit.

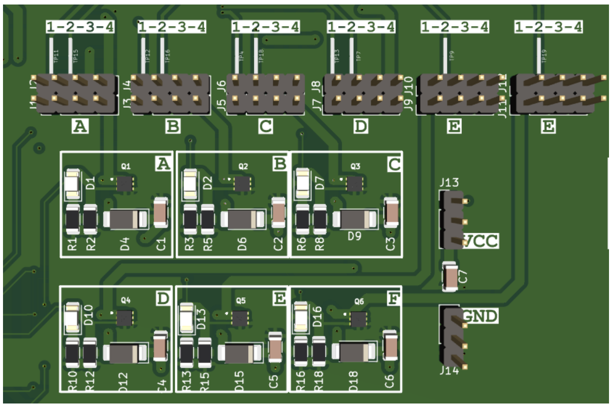


FIGURE 4.9: Pin header for chosing the driven design

The driver part is composed of 6 NMOS transistors and some LEDs to visually see which coil is activated.

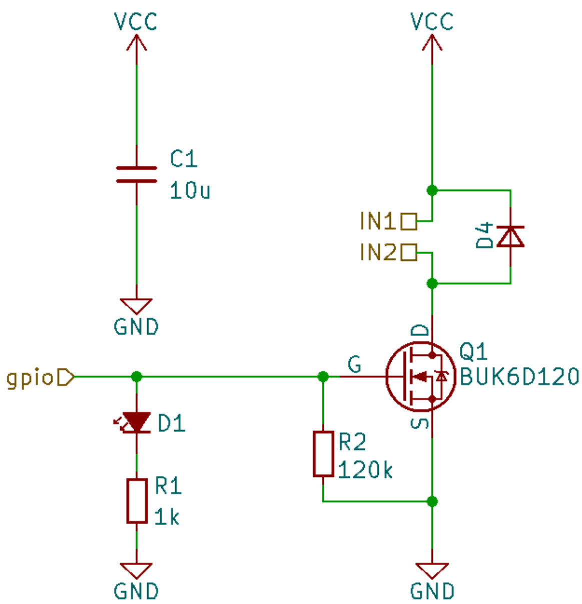


FIGURE 4.10: Schematics for transistor part of the custom PCB

We took advantage of this custom PCB to also test a specific transistor to see if it could handle the current in the coils. We used the BUK6D120 NMOS transistor.

It can handle a drain-source voltage of 40 V, and a temperature of 175°C.

We will power the coils with 5V and we can assume that they will be 1 ohm or more. This means that the absolute maximum current that can flow through the coils is 5A. But we will control it with a PWM at 20%, which means that the average current will be 1A.

We now know that we control the Gate of the NMOS with a 3V3 logic GPIO from the Raspberry Pi PICO. With a Gate Source Voltage of 3V3, we can expect a drain-source resistance of 0.28 ohm as seen in 4.11.

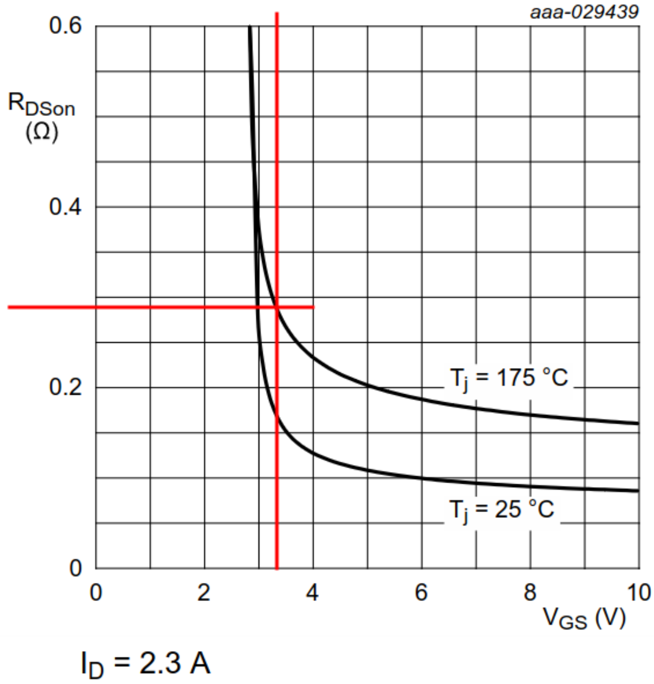


Fig. 9. Drain-source on-state resistance as a function of gate-source voltage; typical values

FIGURE 4.11: BUK6D120 NMOS RDSon graph

This means that the transistor will dissipate

$P = I^2 * R => 1^2 * 0.28 = 0.28 Watts$

The thermal resistance from junction to ambient is maximum 76°C/W, which means that the temperature rise will be $0.28 * 76 = 21.28$ °C of elevation for a constant current of 1A. This is easily manageable for the transistor.

b. Coil designs

The choice to have 4 coil designs was made not to have too big of a PCB while still testing different designs.

b.1. Design 1 : Dual Linear coils

The characteristics of the first design are:

- 2 coils for each axis
- 10 mils (0.256mm) spacing between each traces
- 20 mils (0.508mm) width for the traces

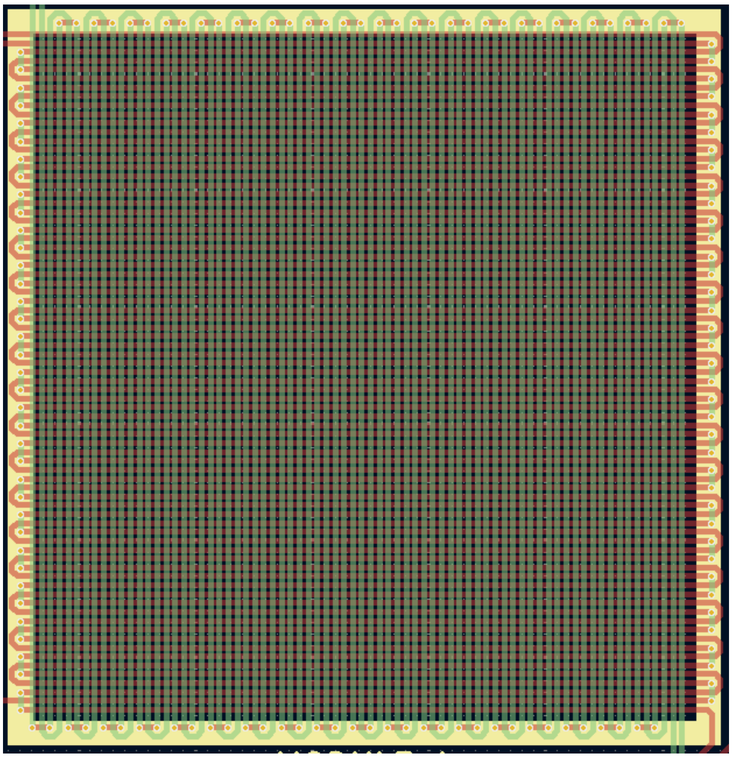


FIGURE 4.12: Dual linear coils design

This design is not working as expected. The dual part of the design makes it impossible to have a magnet move in the same direction consitently because the order of the coils does not matter anymore since there is only 2 coils.

b.2. Design 2 : Triple Linear coils

The characteristics of the second design are:

- 3 coils for each axis
- 10 mils (0.256mm) spacing between each traces
- 20 mils (0.508mm) width for the traces

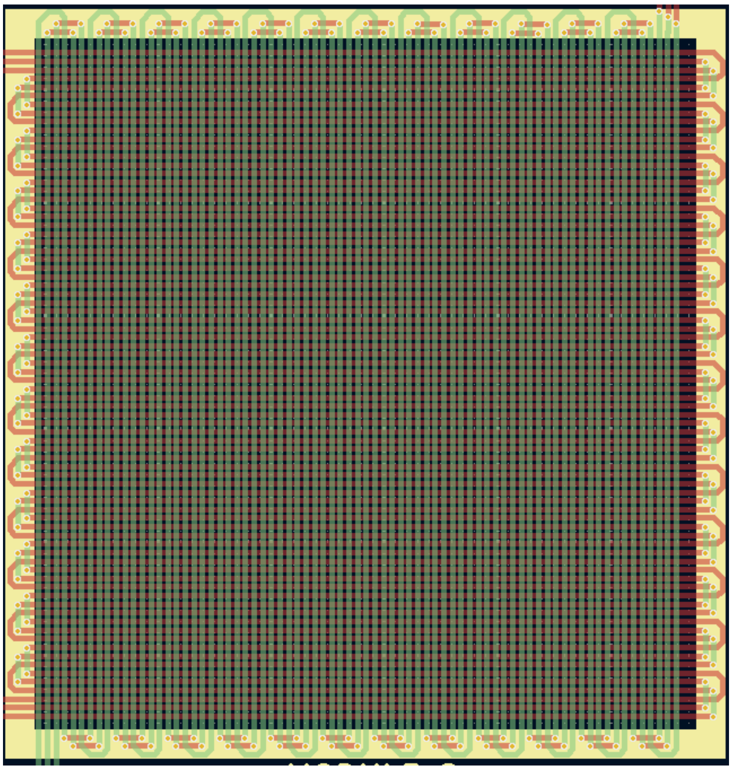


FIGURE 4.13: Triple linear coils design

The design is pretty similar to the Pemi Technology design and is working relatively similar. The main changes have actually been the PCB layers since we switched to 4 layers and have routed the traces on the first top ones. Switching to 4 layers changes the stackup so the first 2 layers are closer to the top, thus closer to the magnet. The magnet can move in both direction but is still stuck sometimes. The magnet is moving in the same axis as the top layer of the PCB.

b.3. Design 3 : Circular coils

This design is closer to the spiral coil design from Carl Bugeja. The characteristics of this third design are:

- 2 layers
- 10 mils (0.256mm) trace width
- 5 mils (0.128mm) spacing traces
- 20mm diameter
- 25 turns

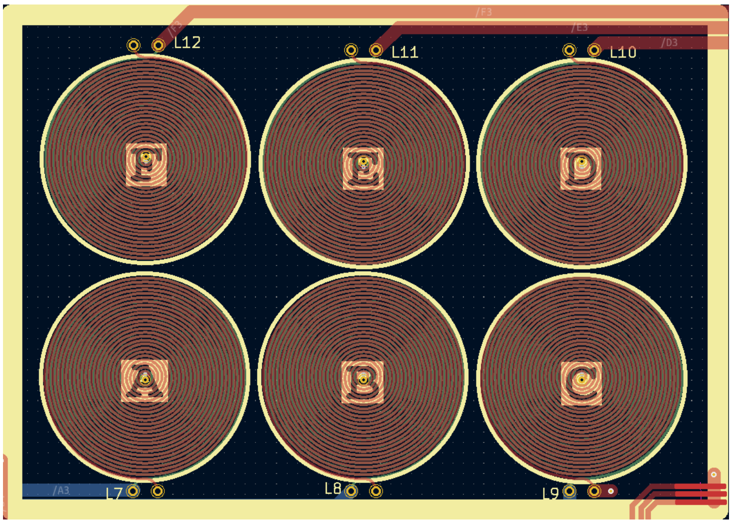


FIGURE 4.14: Circular coils design

This was the most promising design since the magnet was beig attracted to the center of the coil from the border of the spiral. But the magnetic field was not strong enough to move a magnet from a coil to the other.

b.4. Design 4 : Circular coils with more turns

This design is quite similar to the third one but with thinner traces and thinner spacing thus making more turns. The characteristics of this fourth design are:

- 2 layers
- 3.5 mils (0.089mm) trace width
- 3.5 mils (0.089mm) spacing traces
- 20mm diameter
- 50 turns

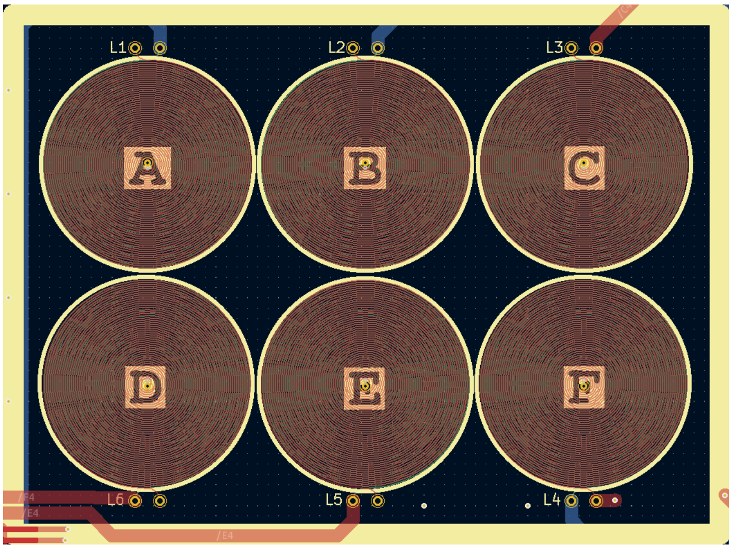


FIGURE 4.15: Circular coils with more turns

This final design should have given us a stronger magnetic for the same current, but with these small traces, the resistance was too high and the current was not enough to move a magnet.

4.3. Stand alone coil PCBs

These designs were not working perfectly so we went back to the drawing board and got some new designs on smaller PCB controlled by the custom board drivers. We only needed to have the designs and some pins to connect to the driver board.

We went for 3 different designs:

- 4 layer circular coils to double de strength of the magnetic field of the third previous design
- 2x2 layer Circular interlaced coils to have a sub-coil closer between each main coil
- 2x2 layer rectangular interlaced coils to have a sub-coil closer between each main coil

a. 4 Layer circular coils

This design has the same characteristics as the third previous design but with 4 layers of coil instead of 2. This should almost double the magnetic field strength.

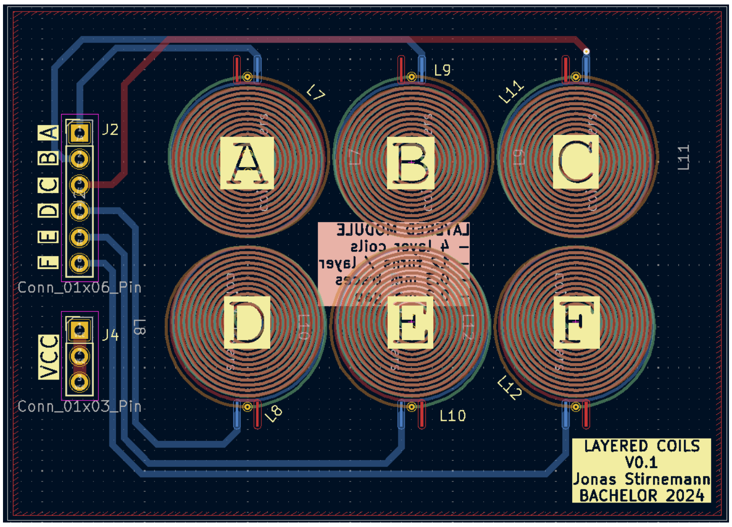


FIGURE 4.16: 4 layer circular coils design

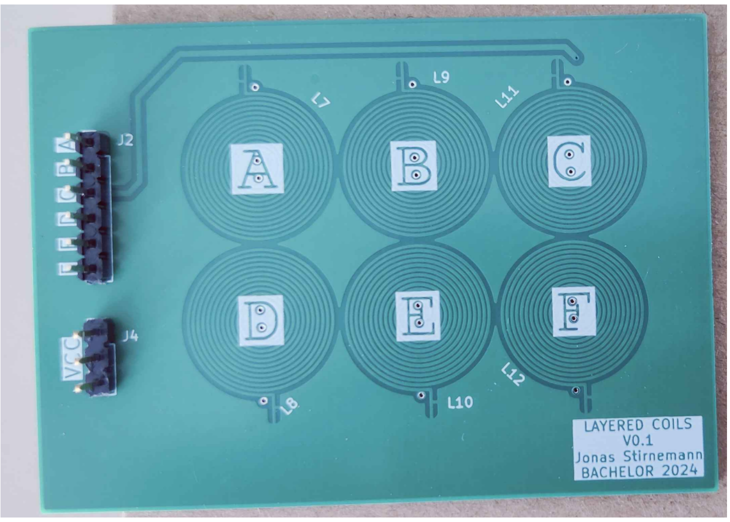


FIGURE 4.17: 4 layer circular PCB

This was not working as expected, the magnet could still not jump from one coil to the other, the doubling of the magnetic field was not enough to move the magnet.

b. 2x2 layers - Circular interlaced coils

This design has a small trick to get multiple coils to overlap each other. On the 4 layer stackup we have a coil on the top layer and the inner second layer and the second coil on the first inner layer and the bottom layer. This way, we can have the coils overlapping and it's easier to move the magnet from one coil to the other.

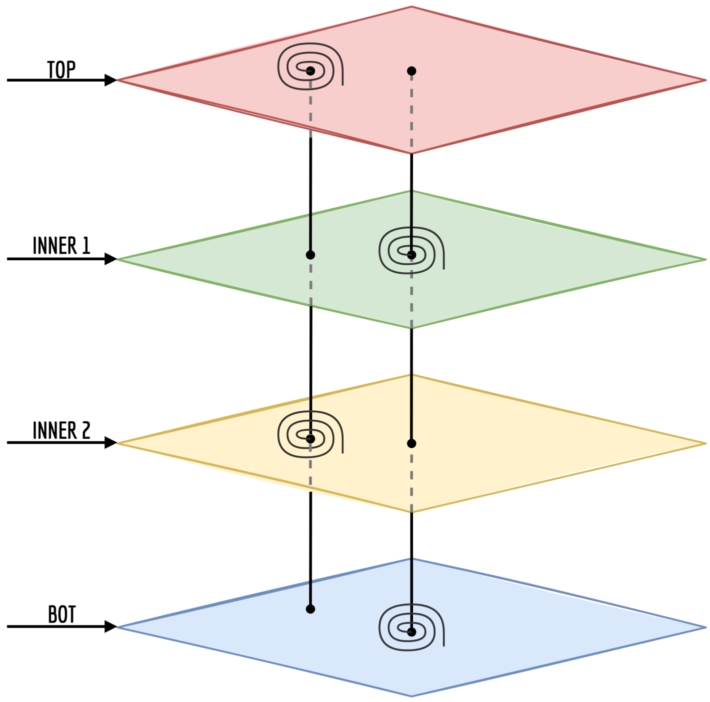


FIGURE 4.18: PCB Stackup 4 layer overlap

One issue with this shape is that it's hard to have them overlap on the 4 sides when being the same diameter, so we had to make smaller coils for the sub-coils. The center of the coils is where the force is the strongest, so we need the center of the coils to be as close as possible from each other.

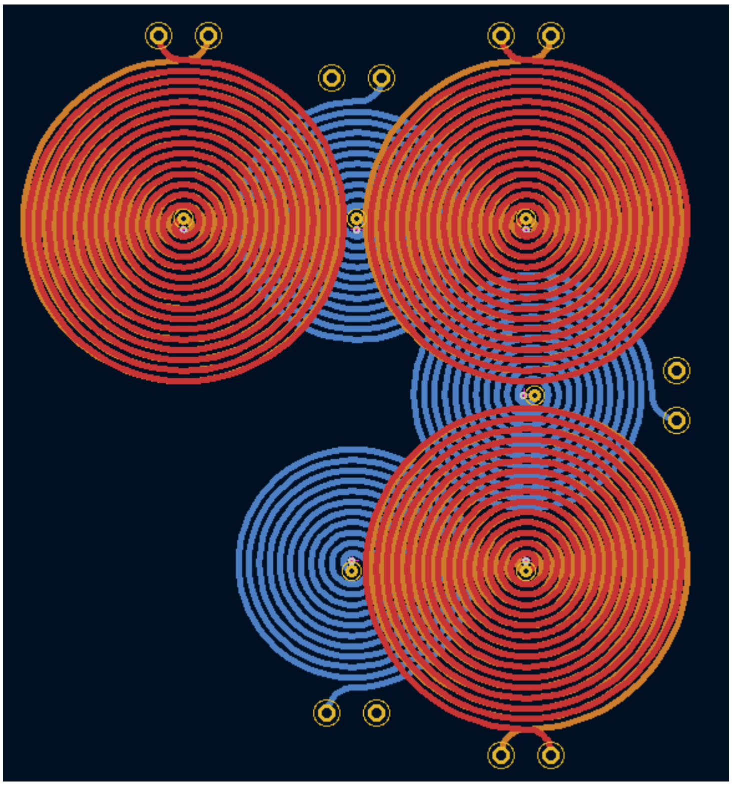


FIGURE 4.19: Circular sub coils are smaller

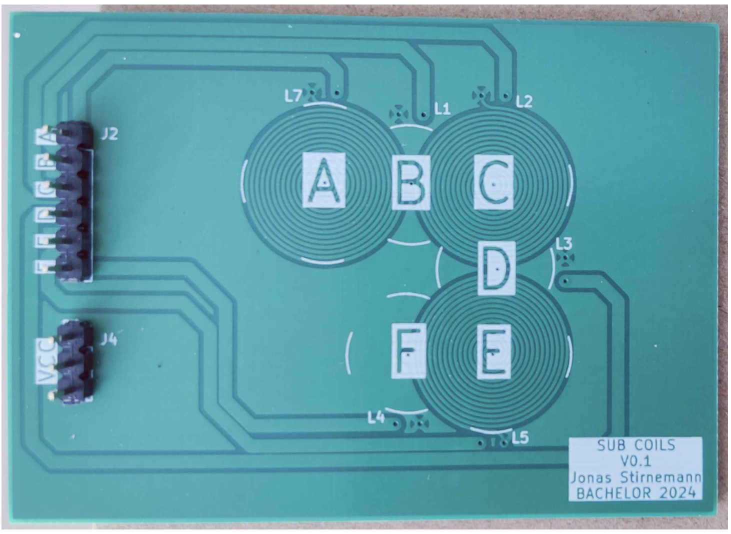


FIGURE 4.20: Circular sub-coils PCB

While promising, this design turned out to have it’s sub coils too small to be powerfull enough to move a magnets between coils. The issue comes from the circular shapes that are difficult to have them overlap on the 4 sides of the main circle.

c. 2x2 layers - Rectangular interlaced coils

This design is similar to the previous one but with rectangular coils. This way we can have the coils overlap on the 4 sides of the main coil while still being the same size as the main.

The stackup is the same, the top layer and the inner second layer have the first coil and the first inner layer and the bottom layer have the second coil.

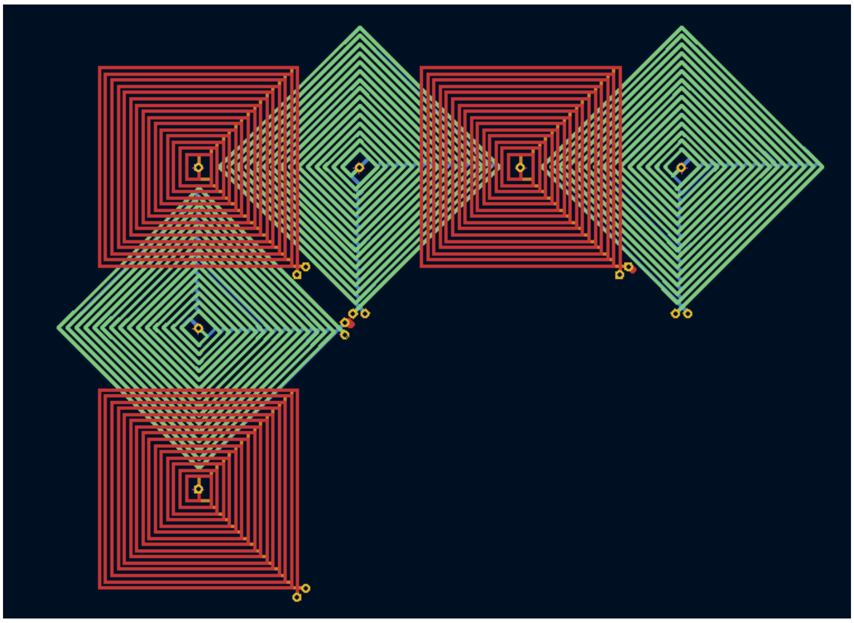


FIGURE 4.21: squares Stackup 4 layer overlap

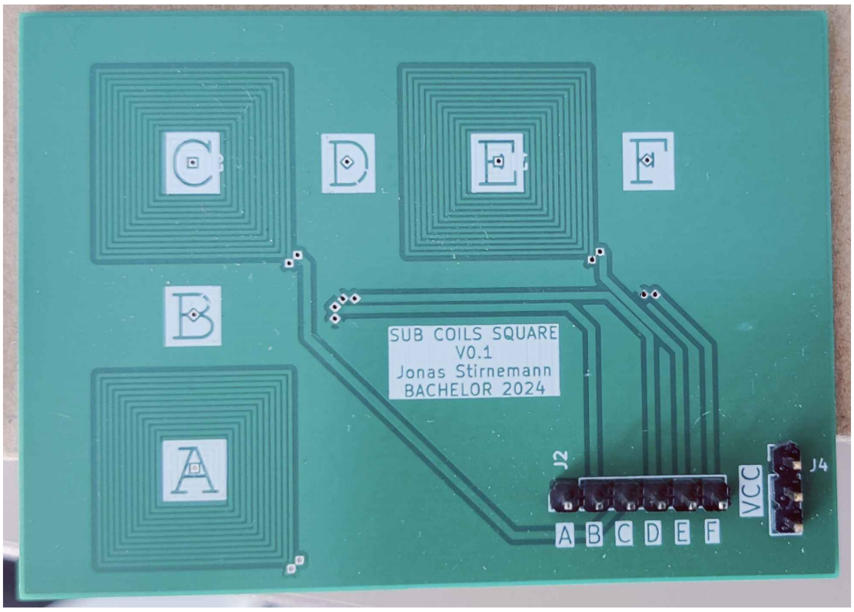


FIGURE 4.22: Square sub-coils PCB

This design was the most promising, I made a 10mm magnet move back and fourth between the coils, it was heating up because the coils were being activated almost all the time, but from what i gathered with the thermal camera, the maximum heating we had was 70°C which is clearly still manageable for the PCB and the magnet.

CHAPTER 5 : FINAL TIC-TAC-TOE HARDWARE DESIGN

Now that we found the best coil design, we can make a full matrix of coils to move the magnet in a 2D space. We will use the square coils with sub-coils. we will also need a driver circuit power the matrix of coils individually.

5.1. CHOICE OF LAYOUT

We want to make a matrix big enough to play Tic-Tac-Toe. This means that we need at least 3x3 usable cells plus some side coils to hold the non placed pieces. So in the end we need a 5x4 usable matrix of coils. But as we describe in the next section, we also need intermediate buses to avoid attracting the wrong magnets. This means that for each usable cell, we need a triple bus for rows and columns.

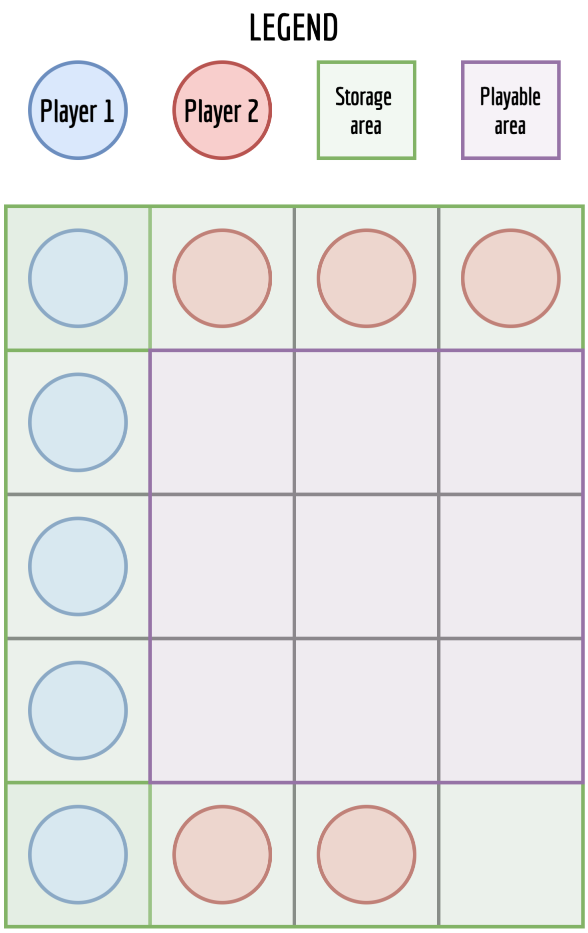


FIGURE 5.1: Description of zones for tic-tac-toe

a. Magnet buses

Since the design can attract magnets from any directions, we need intermediate buses or routes to avoid attracting the wrong magnets.

The issue is a single bus would not be enough since activation a coil on the bus with magnets on both sides would attract both magnets.

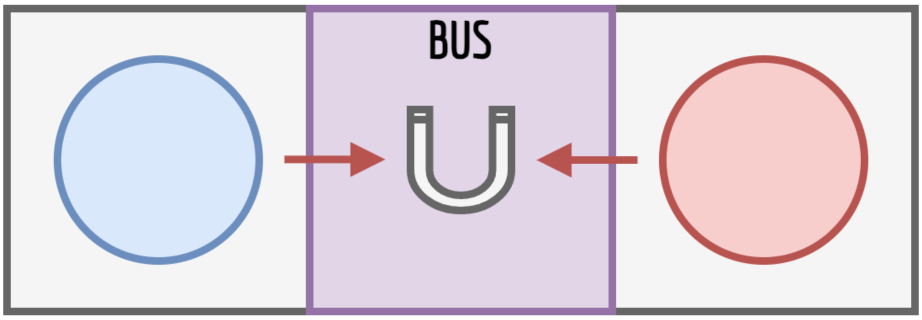


FIGURE 5.2: Magnet single bus

A dual bus would still not be enough since when moving from one cell to another, we could come across situations where the path requires to activate a coil on the bus with magnets on both sides.

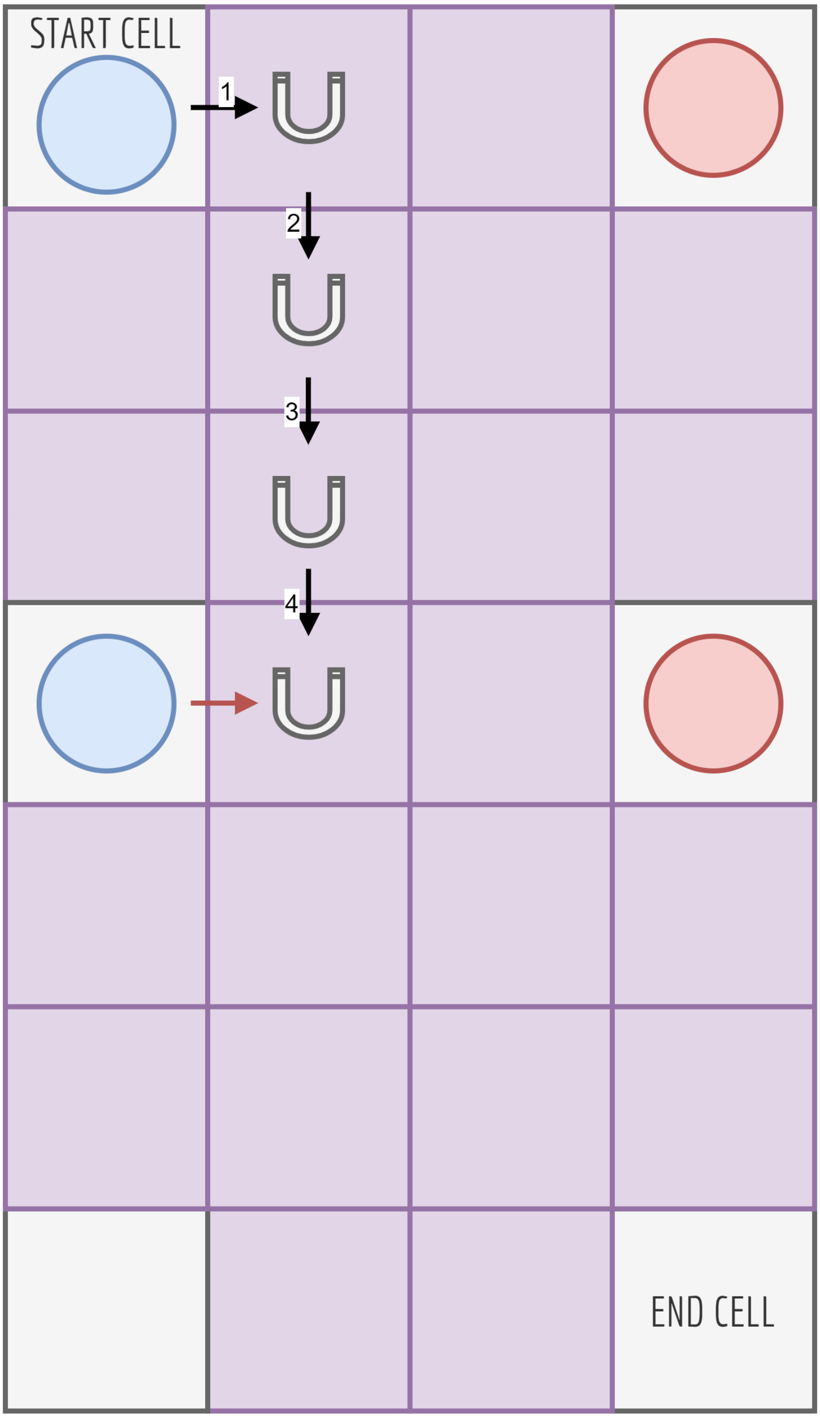


FIGURE 5.3: Magnet dual bus

The solution would be to have a triple bus system. This way we go onthe middle of the bus to move anywhere without attracting the wrong magnets.

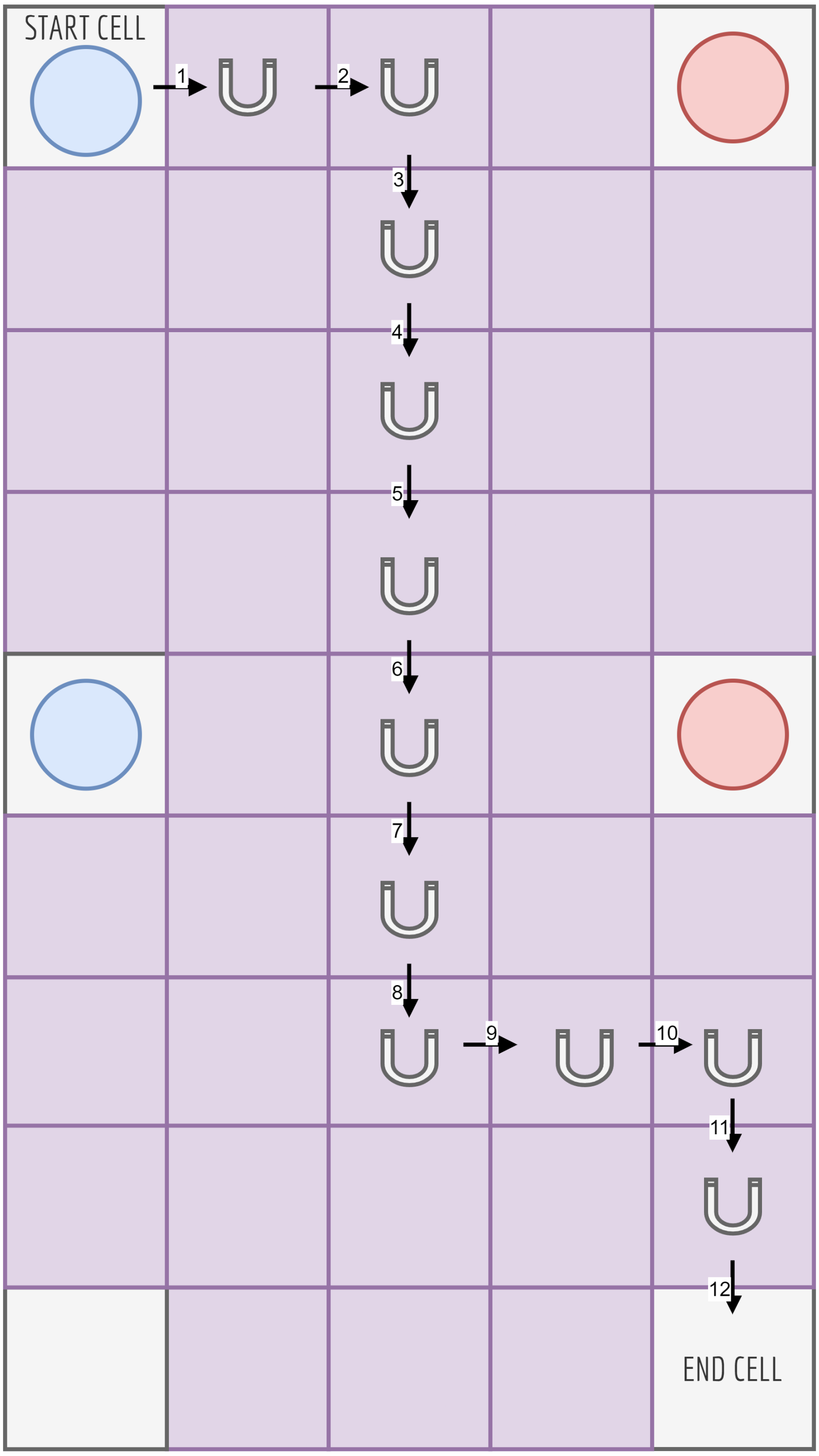


FIGURE 5.4: Magnet triple bus

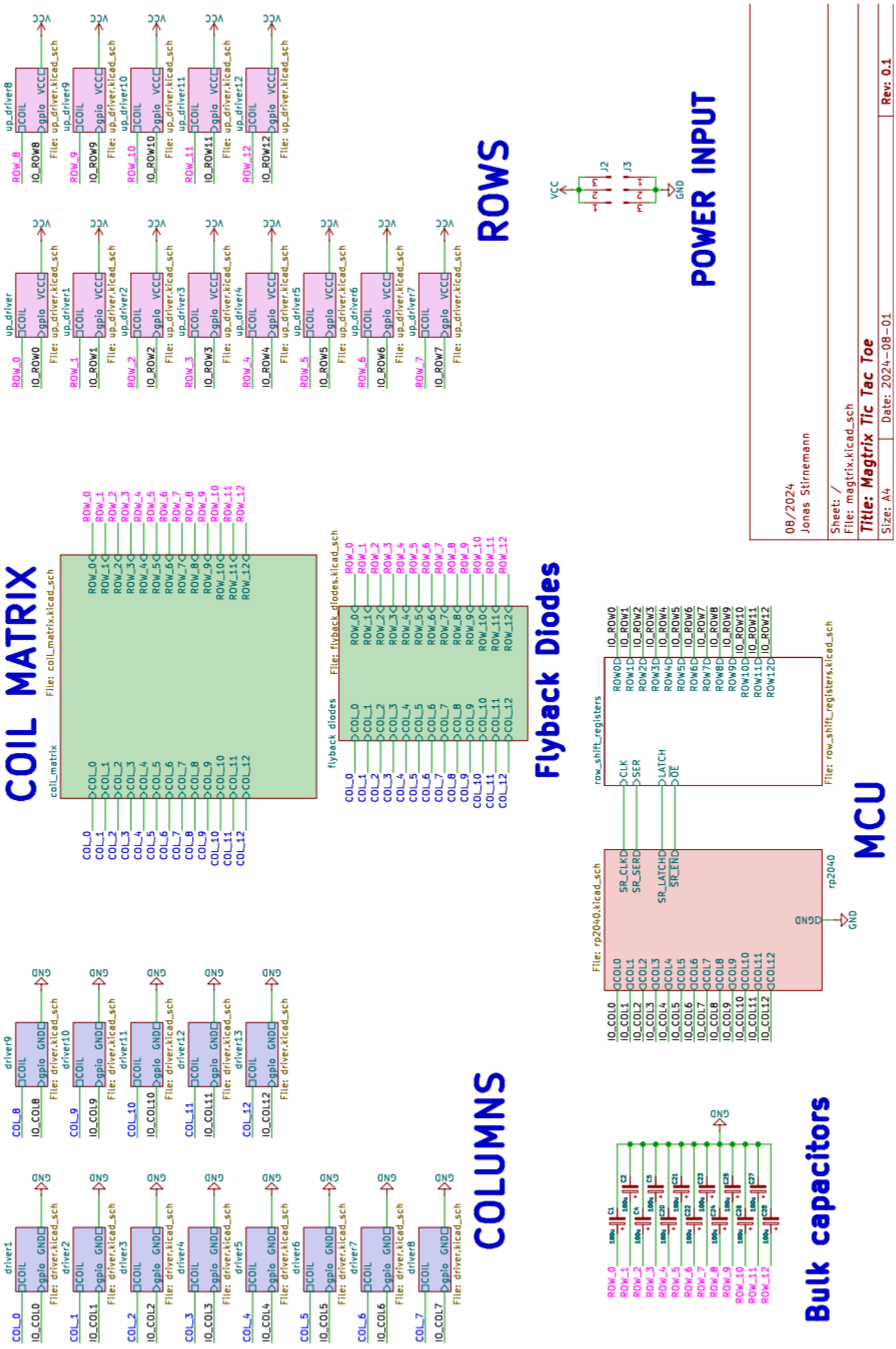


FIGURE 5.5: General schematics of final design

5.2. Driver circuit

Now that we have a matrix of coils, it would take space, time and money to have a driver circuit for each coil. We will use a row and column driving method. This means that each coil will be connected to a row and a column. We will then activate the row and column to activate the coil. This way we can activate any coil in the matrix by activating the corresponding row and column. Each row is driven by a PMOS transistor, pulling one side of the coil to VCC and each column is driven by a NMOS transistor, pulling the other side of the coil to GND.

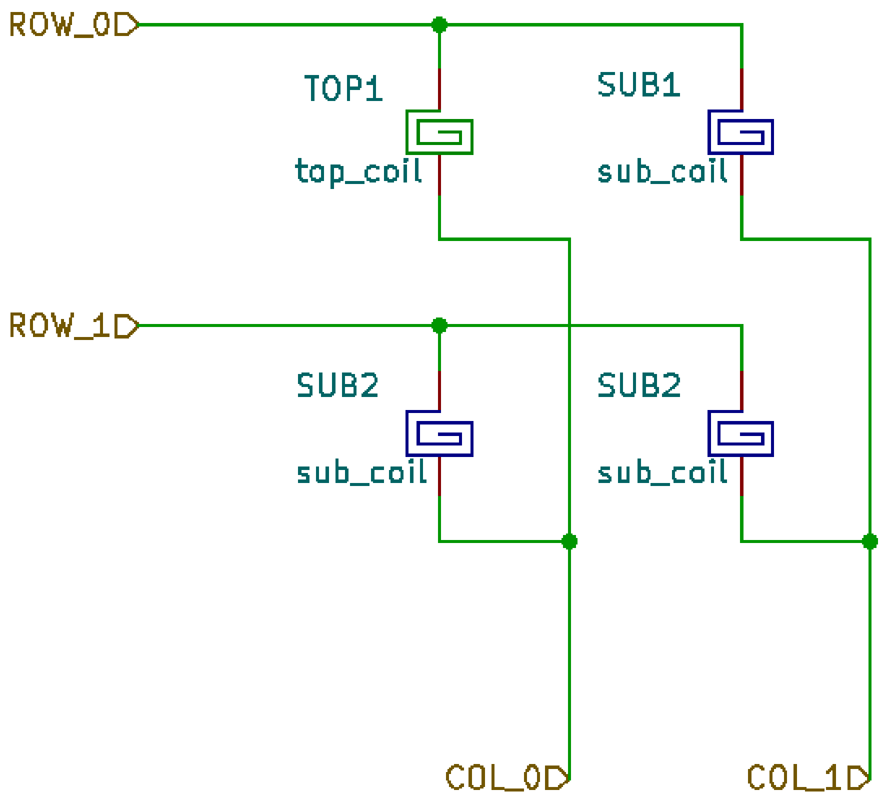


FIGURE 5.6: Matrix driving concept

This means that each coil is driven by the equivalent of this schematics.

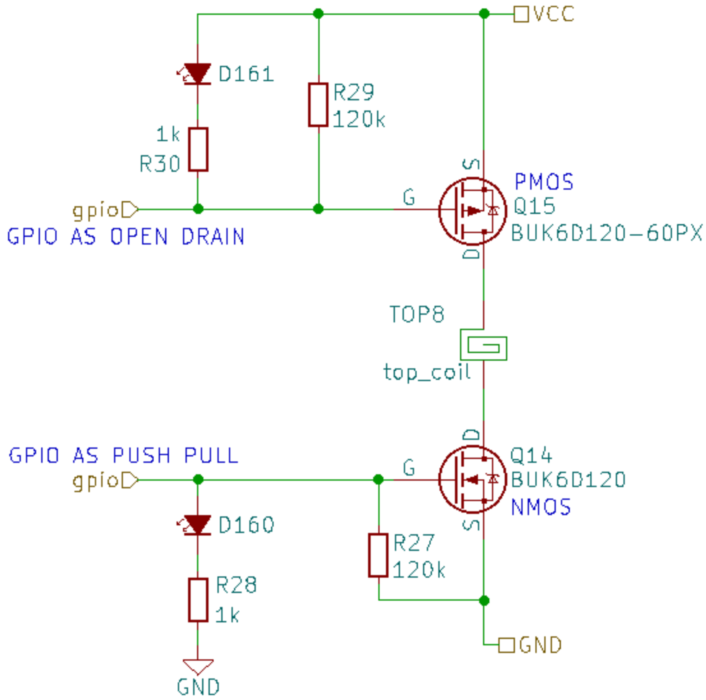


FIGURE 5.7: Equivalent Dual mos driver circuit

5.3. KICAD SCRIPTING

I've tried to leverage the power of scripting in Kicad and create three simple scripts. They need to run with the Kicad's python instance since they use the Kicad API.

a. Footprint generation

One of them to create the footprint of the coil itself by specifying the parameters of the coils:

- The traces width,
- The spacing between traces
- The side of the square

```python
if __name__ == "__main__":
    coiler = Coiler(side_width=8, line_width=0.3, spacing=0.2)

    coiler.create_and_save(
        name="coil",
        description="Test Coil",
        filename="top_coil",
        output_dir="outputs",
        layers=["F.Cu", "In2.Cu"],
    )

    coiler.create_and_save(
        name="coil",
        description="Test Coil",
        filename="sub_coil",
        output_dir="outputs",
        layers=["In1.Cu", "B.Cu"],
    )
```

FIGURE 5.8: Code to generate Kicad square footprint

b. Schematic generation

Now that we have the footprint, we want to generate a matrix of coils in the schematic. For that we need to create a blank schematics and add and configure 2 coils. Then the script will duplicate the coils and connect them to the rows and columns.

```python
if __name__ == "__main__":
    in_file = out_file = "../test_project/test_project.kicad_sch"

    cmm = ComponentMatrixMaker(
        schematics_filename=in_file,
        component_prefix1="TOP",
        component_prefix2="SUB",
        size=(16, 16),
    )

    cmm.save_to_schematics(save_to=out_file)
```

FIGURE 5.9: Code to generate Kicad schematic

c. PCB generation

Now that we have the schematic, we want to place the coils in a matrix on the PCBnew software. For that, we need to already have exported the components into PCBnew. We also need text files with the list of names of the coils to position.

```
pcb_path = "magtrix/magtrix.kicad_pcb"
ref_layout_path = "sub_coils.csv"
pcb = pcbnew.LoadBoard(pcb_path)
components = pcb.GetFootprints()

# Only keep the SUB referenced components
components = [component for component in components if "SUB" in component.GetReference()]
number_of_coils = len(components)
layout = read_layout_csv(ref_layout_path)
final_arr = []
for i in range(layout.shape[0]):
    row = []
    for j in range(layout.shape[1]):
        for component in components:
            #print(f"{i} {j} {component.GetReference()} {layout[i][j]}")
            if component.GetReference() == layout[i][j]:
                row.append(component)
                break
    final_arr.append(row)
for i in range(layout.shape[0]):
    for j in range(layout.shape[1]):
        print(f"{final_arr[i][j].GetReference():^10} ", end="")
    print("")
sub_delta_x = 26
sub_delta_y = 13
sub_delta_half = 13
origin = 0, 0
for y_axis in range(layout.shape[0]):
    for x_axis in range(layout.shape[1]):
        component = final_arr[y_axis][x_axis]

        if "SUB" in component.GetReference():
            if y_axis % 2 == 0:
                pos = pcbnew.wxPointMM(
                    float(origin[0] + x_axis * sub_delta_x),
                    float(origin[1] + y_axis * sub_delta_y),
                )
            else:
                pos = pcbnew.wxPointMM(
                    float(origin[0] + x_axis * sub_delta_x - sub_delta_half),
                    float(origin[1] + y_axis * sub_delta_y),
                )
            component.SetOrientationDegrees(45)

        print(f"Moving {component.GetReference()} to {pos}")
        print(f"{component}")
        component.SetX(pos.x)
        component.SetY(pos.y)
# Save the updated PCB file
pcb.Save(pcb_path)

print("PCB file updated")
```

FIGURE 5.10: Code to position coils on the PCB

5.4. CONTROL

We will use a microcontroller to control the matrix. We integrated the RP2040 microcontroller from Raspberry Pi on the PCB. It has enough GPIOs to control each row and column of the matrix and it can produce at least 16 PWM signals from a dedicated peripheral to control the columns NMOS.

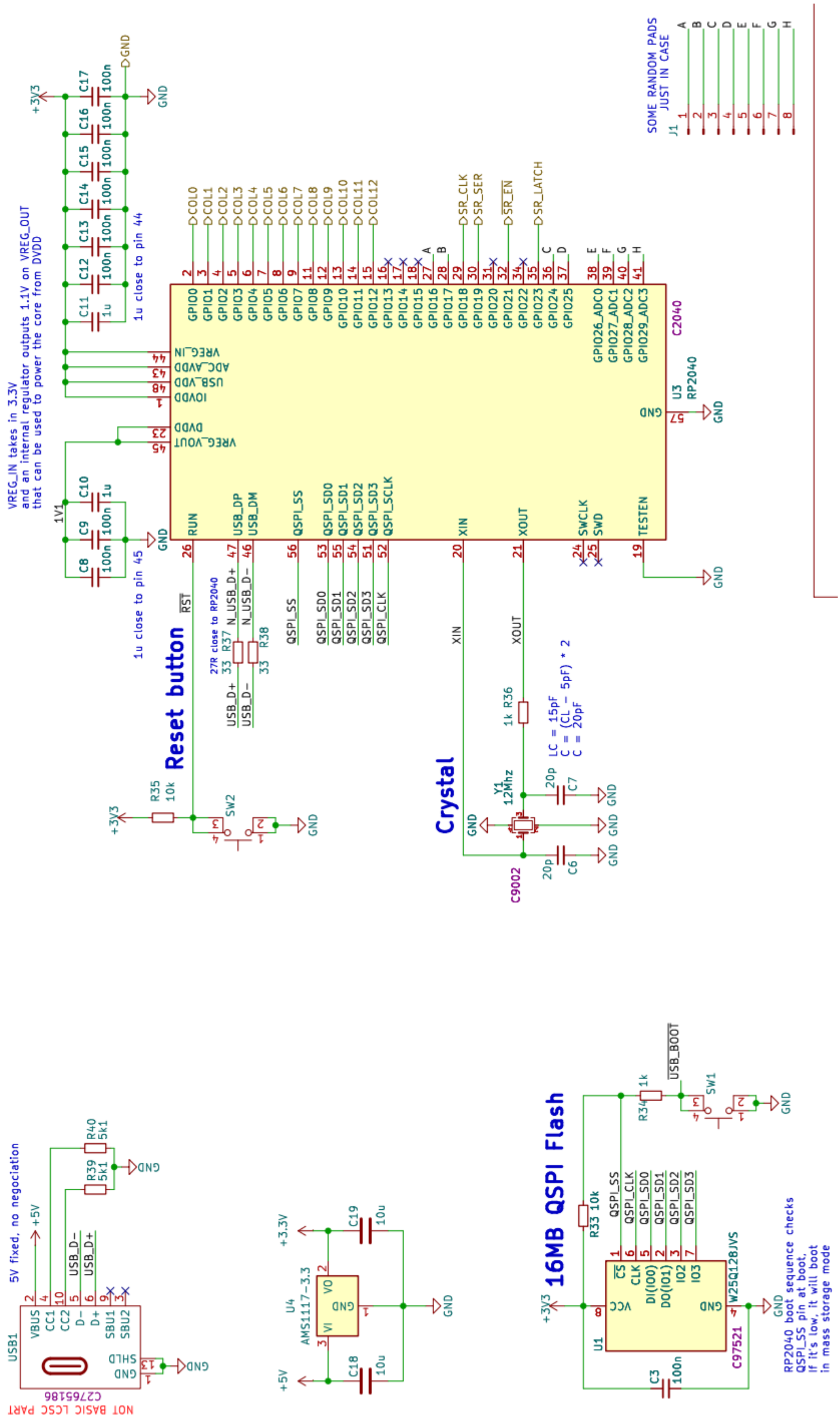


FIGURE 5.11: RP2040 microcontroller schematics

In order to control bigger matrix, we might have to use a shift reigster to control the rows PMOS, for that purpose, we added a shift register on the PCB to future proof the design. Unfortunately, we needed a Serial In Parallel Out shift register with open drain outputs to control the PMOS transistors but I picked the wrong one, so i had to hardwire gpios to the PMOS...

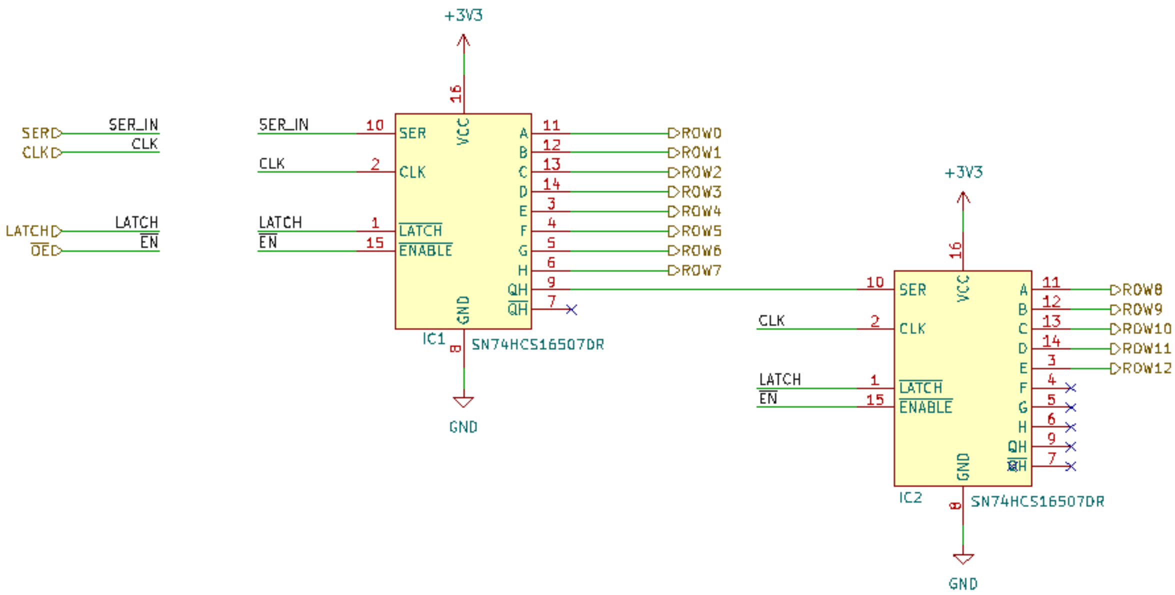


FIGURE 5.12: Shift register schematics

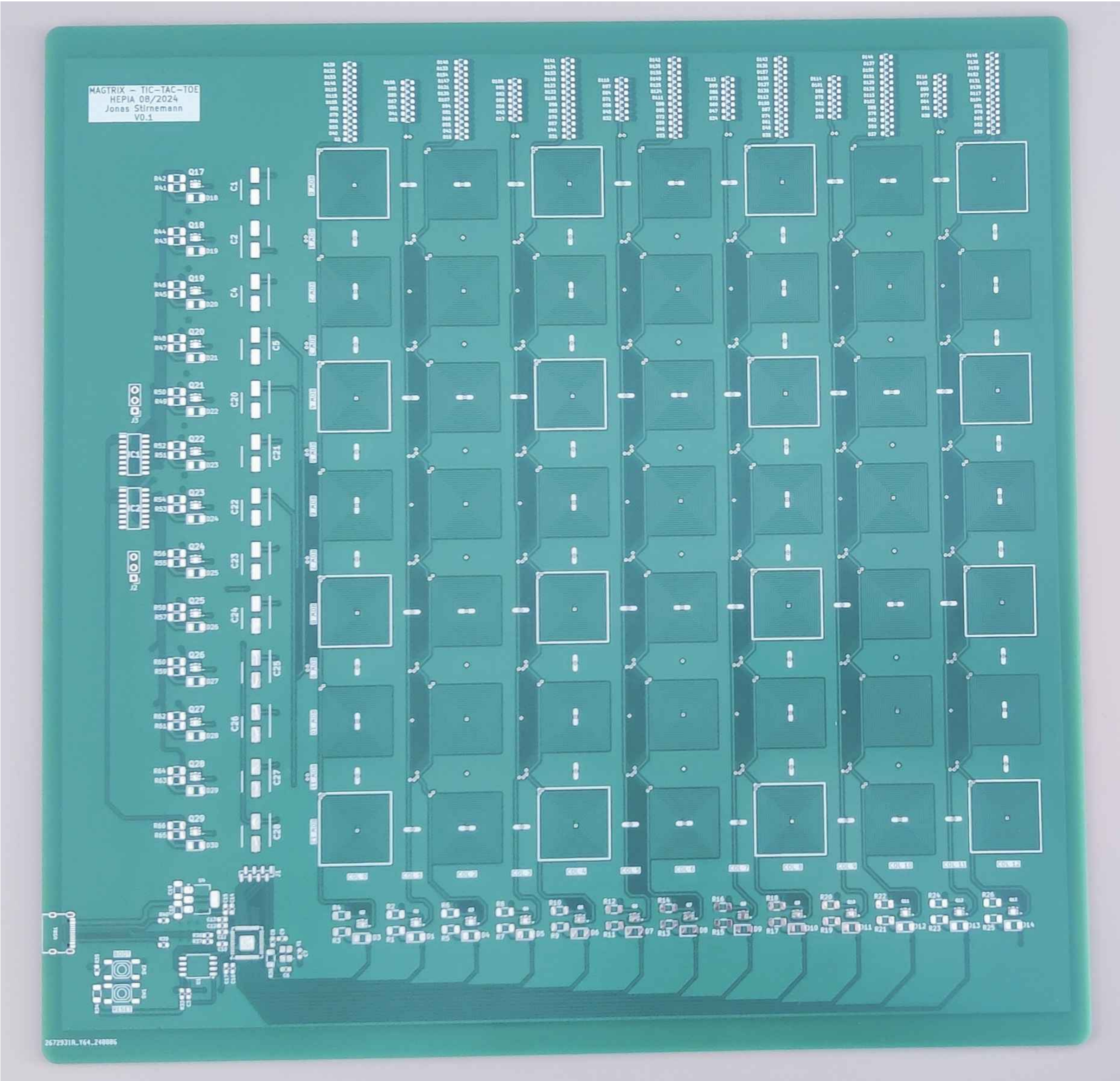


FIGURE 5.13: Final Unmounted PCB

The shift registers had to be removed and some gpios were hardwired to the PMOS transistors. We outputed some pads with the remaining free gpios to have some slack in case of issues (Like a wrong Shift register...).

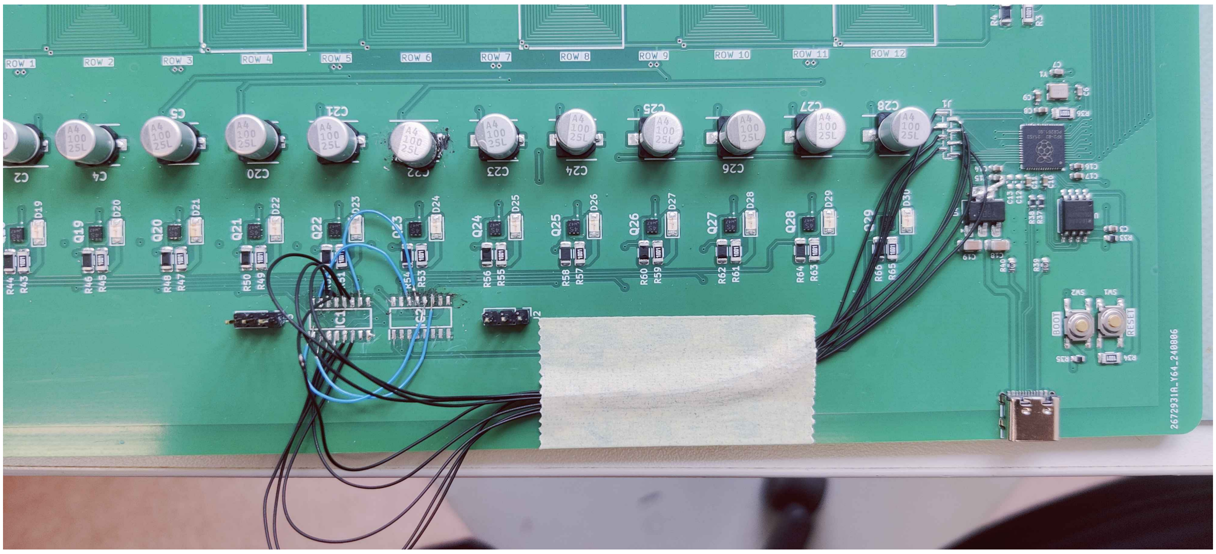


FIGURE 5.14: Hardware Shift registers fixes

Chapter 6 : Final Tic-Tac-Toe Software

For the software component of the Tic-Tac-Toe game, a web application will be used to make the moves. This web app will communicate with a backend system, which will then interface with the microcontroller to control the coils that move the game pieces.

6.1. Web App - Frontend

The web application is straightforward, featuring a 3x3 grid of buttons. When a button is pressed, it sends a request to the backend to execute the move. While the board is making a move, the buttons are temporarily disabled. The web app waits for confirmation from the backend before re-enabling the buttons for the next move.

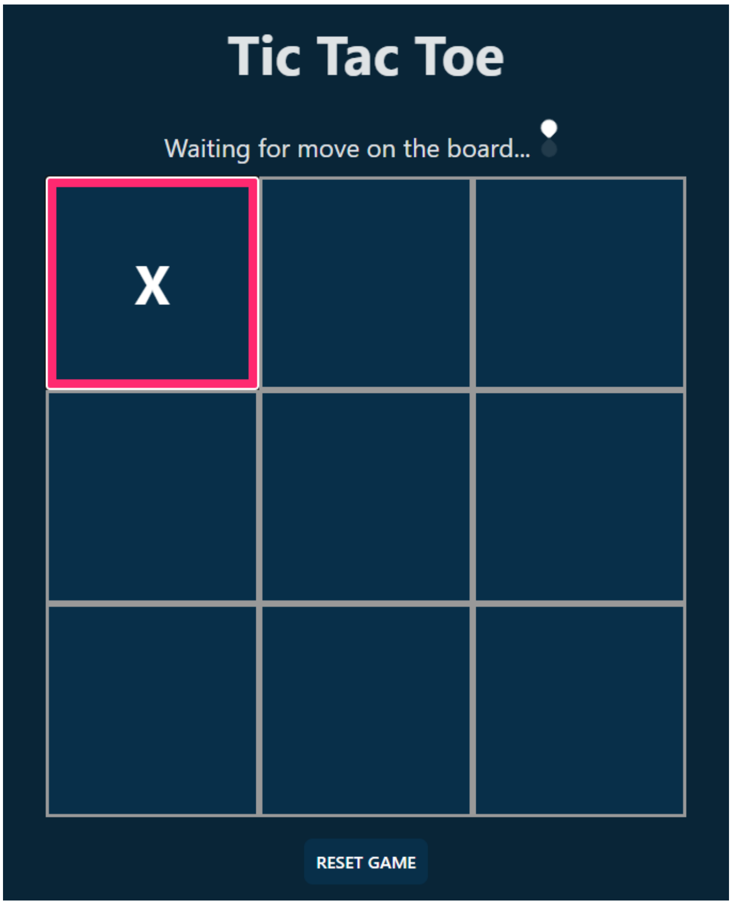


FIGURE 6.1: Web app waiting for an acknowledgement after a player move

The web app is made with React.js[13], a JavaScript library for building user interfaces. It is

[13] *React*. en. URL: `https://react.dev/` (visited on 08/20/2024).

a single page application that uses the WebSocket protocol to communicate with the backend.

6.2. Web App - Backend

The backend is a straightforward Python script that establishes a WebSocket connection with the web app. It listens for messages from the web app and forwards them to the microcontroller via serial communication. After receiving an acknowledgment from the microcontroller, the backend sends the result back to the web app.

The serial communication is also simple: the backend sends the move coordinates to the microcontroller, framed by a start byte and a stop byte. Since the frontend handles the game rules and permissions, the backend only receives valid moves, ensuring that no illegal moves are processed.

Packet from the backend to the microcontroller:

- Start byte : 0x02
- X coordinate : 0 to 2 in ASCII
- Y coordinate : 0 to 2 in ASCII
- Stop byte : 0x03

Packet from the microcontroller to the backend:

- Start byte : 0x02
- Result : OK or KO in ASCII (KO when issue happened)
- Stop byte : 0x03

6.3. MICROCONTROLLER FIRMWARE

The RP2040 is programmed in C with the Pico SDK[14] and I decided to use a Real Time Operating System for easier multi task management and timings, I chose FreeRTOS[15].

The firmware will wait for a packet from the backend, then it will find the best path to move the magnet to the desired position. It will fill a queue with the coils to activate in the correct order. It will then activate the coils one by one and wait some arbitrary time before activating the next coil. When the sequence is done, it will send an acknowledgement to the backend.

The magnets are moved by activating the coils in the correct order. For that, we have to generate a PWM signal on the columns and activate the rows.

The RP2040 has an integrated PWM peripheral that can be programmed via the Pico SDK API to generate the PWM signal.

A simpler API was made to abstract some computing to get the best timer values, so we only have to chose a Frequency and and we can then set a duty cycle from 0.0 to 1.0 when desired.

```c
mag_pwm_t led_pwm;
mag_pwm_init_from_gpio(led_pwm, LED_GPIO, 25000, 0.0);
mag_pwm_enable(led_pwm);
mag_pwm_set_duty(led_pwm, 0.5);
```

FIGURE 6.2: Custom PWM API

[14] *Hardware APIs - Raspberry Pi Documentation*. en. URL: https://www.raspberrypi.com/documentation/pico-sdk/hardware.html (visited on 08/20/2024).

[15] *FreeRTOS™ - FreeRTOS™*. URL: https://freertos.org (visited on 08/20/2024).

CONCLUSION

This thesis presents a proof of concept for a scalable, automated board game system that integrates PCB-based electromagnetics, offering a novel approach to bringing digital game tracking into the physical world. Initially focused on a Tic-Tac-Toe game, the project successfully demonstrated the ability to move game pieces automatically via coils etched directly onto the PCB, controlled by a web application.

The iterative process of coil design and control system development highlighted the potential of this technology to be adapted for more complex games like chess, with minimal adjustments. The final prototype, a 3x3 Tic-Tac-Toe board, effectively showcases the concept's viability, despite some hardware imperfections.

While the current implementation is limited to Tic-Tac-Toe, the core technology is designed with scalability in mind, offering possibilities for expansion into larger and more intricate board games. The project's been succesful in automating game piece movement through electromagnetism, but further development is needed to address issues with multiple bare magnets interacting with each other. The game is not fully playable as multiple magnets are needed.

The simulation of coil designs proved challenging, as I was unable to achieve results that matched real-world outcomes. This discrepancy primarily came from limited expertise in electromagnetism and the inherent complexity of the simulations. As a result, the design process relied heavily on a trial-and-error approach. A more in-depth theoretical understanding of electromagnetism would likely have led to more accurate simulations and a more efficient identification of the optimal coil design. Moving forward, strengthening the theoretical foundation in this area could greatly enhance the design process, leading to more precise and reliable outcomes.

APPENDIXS

APPENDIX 1

GITHUB Repository Link

https://github.com/ThePurpleOne/Magtrix

BIBLIOGRAPHY

2D stepper motor etched on PCB - micro manipulator. en. URL: https://hackaday.io/project/164507-2d-stepper-motor-etched-on-pcb-micro-manipulator (visited on 08/18/2024).

arthur.abelkalo / ChessGame · GitLab. en. Aug. 2023. URL: https://githepia.hesge.ch/arthur.abelkalo/chessgame (visited on 08/19/2024).

atomic14/kicad-coil-plugins: Some simple plugins for generating stator coils in kicad. URL: https://github.com/atomic14/kicad-coil-plugins/tree/main (visited on 08/17/2024).

Buy strong magnets online - supermagnete.ch. URL: https://www.supermagnete.ch/eng/ (visited on 08/20/2024).

Carl Bugeja. *Actuating Magnets with PCBs*. 2018. URL: https://www.youtube.com/watch?v=LudKMv7M13Q (visited on 08/19/2024).

Carl Bugeja. en-US. URL: https://www.carlbugeja.com (visited on 08/19/2024).

Factors influencing adhesive force of magnets. URL: https://www.supermagnete.ch/eng/faq/What-factors-affect-the-adhesive-force-and-strength-of-a-magnet (visited on 08/20/2024).

Faria, Andreia et al. “A Fast and Precise Tool for Multi-Layer Planar Coil Self-Inductance Calculation”. en. In: *Sensors* 21.14 (Jan. 2021). Number: 14 Publisher: Multidisciplinary Digital Publishing Institute, p. 4864. ISSN: 1424-8220. DOI: 10.3390/s21144864. URL: https://www.mdpi.com/1424-8220/21/14/4864 (visited on 08/17/2024).

FR-4. en. Page Version ID: 1218528760. Apr. 2024. URL: https://en.wikipedia.org/w/index.php?title=FR-4&oldid=1218528760 (visited on 08/19/2024).

FreeRTOS™ - FreeRTOS™. URL: https://freertos.org (visited on 08/20/2024).

Hardware APIs - Raspberry Pi Documentation. en. URL: https://www.raspberrypi.com/documentation/pico-sdk/hardware.html (visited on 08/20/2024).

Home - PeMi Technology - Factory For Materializing Dreams. URL: http://www.pemi.technology/ (visited on 08/19/2024).

Ltd, Raspberry Pi. *Buy a Raspberry Pi Pico*. en-GB. URL: https://www.raspberrypi.com/products/raspberry-pi-pico/ (visited on 08/19/2024).

nasa_magnetism. URL: https://pwg.gsfc.nasa.gov/Education/Imagnet.html (visited on 08/14/2024).

PCB Stator. en-US. URL: https://pcbstator.com/ (visited on 08/18/2024).

React. en. URL: https://react.dev/ (visited on 08/20/2024).

SRI. *Magnetically Actuated Micro-Robots for Advanced Manipulation Applications*. 2014. URL: https://www.youtube.com/watch?v=uL6e3co4Qqc (visited on 08/18/2024).