

Diploma Thesis

Class-E Tesla Coil

Simon Marcher
Kassandra Peyer
Felix Hamrle

March 16, 2022

Supervised by
Dipl.-Ing. Thomas Jerman
Dipl.-Ing. Michael Lieschnegg

Powered by
HTL Arnfels

Affidavit

I declare in lieu of an oath that I have written the present thesis independently and without outside help other than the stated sources, and that I have made the passages taken from these sources recognisable as such.

Simon Marcher

Kassandra Peyer

Felix Hamrle

Abstract

During the last few decades, Tesla coils have seen a major upwind in popularity. Its impact in the scientific community reached far beyond the indented purpose of high voltage and X-ray generation. This diploma thesis guides through the design process and lays out the inner working of those devices. As opposed to most existing Tesla coils operating at 100 - 500 kHz, this coil's operating frequency is located in the lower MHz band, which poses a series of interesting challenges to overcome. Most noteworthy being the power generation, that required the use of a highly-efficient class-E amplifier. Simulation tools were used in order to verify calculated values. A proper casing and a PCB was designed to reach nominal consumer product quality. An external, AVR-based MIDI-Interrupter uses the well-known MIDI protocol to let the Tesla coil play simple music. In its final stage, it was able to produce arcs up to 10 mm with only a few watts of input power.

Zusammenfassung

In den letzten Jahrzehnten hat die Teslaspule an großer Beliebtheit gewonnen. Ihr Einfluss in der wissenschaftlichen Forschung geht weit über ihren ursprünglichen Zweck der Hochspannungs- und Röntgenstrahlenerzeugung hinaus. Diese Diplomarbeit behandelt einen Designprozess und zeigt den Aufbau und die Funktionsweise eines solchen Geräts auf. Im Gegensatz zu den meisten Teslaspulen, welche im Bereich von 100 - 500 kHz arbeiten, liegt diese Spule im niedrigen MHz Bereich, was zu einer Menge an interessanten Herausforderungen führt. Eine dieser Herausforderungen, ist die Leistungserzeugung, welche nur mit einem hocheffizienten Klasse-E Verstärker bewältigt werden kann. Ergebnisse von Berechnungen wurden durch Simulationen bestätigt. Um die Qualität eines Endverbraucherprodukts zu erreichen, wurde ein geeignetes Gehäuse mit einer Printplatte konstruiert. Um mit der Teslaspule Musik erzeugen zu können, wurde ein externer AVR-basierter MIDI-Interrupter programmiert. Am Ende konnte die Teslaspule Lichtbögen von bis zu 10 mm mit einer Eingangsleistung von nur wenigen Watt erzeugen.

Contents

Affidavit	iii
Abstract	v
Contents	vii
The Tesla Coil	1
1 Theory of Operation	3
1.1 The Tesla Resonator	3
1.2 Exciting The Resonator	4
The Spark Gap Tesla Coil	4
Solid-State Tesla Coils	4
1.3 Class-E Amplifier	7
Theory of Operation	7
Class-E for Tesla Coils	8
1.4 Musical Tesla Coils	8
2 Design and Simulation	11
2.1 The Coils	11
Designing the Secondary Coil	11
Designing the Primary Coil	12
Simulating the Coupling	12
Modeling the Transformer	13
2.2 Class-E Design	15
The Design Process	15
2.3 The Class-E Tesla Coil Driver	17
Simulation	18
2.4 Generating the Gate Signal	18
Voltage Controlled Oscillator	18
MOSFET Gate Driver	19
2.5 The Interrupter	19
Fiber Optic Transmission	19
Interrupt Synchronization	21
2.6 Synopsis	21

3 Practice and Measurements	23
3.1 Tuning the Secondary Coil	23
3.2 The Class-E Stage	23
Fine-Tuning the Setup	24
Measurements	25
3.3 Image Gallery	27
Design and Building	29
4 Casing	31
4.1 Concept design	31
4.2 Materials	32
Metals	32
Woods	32
Glass	32
Plastics	33
Conclusion	33
4.3 Structure for the Coils	34
Primary Coil	34
Secondary Coil	35
4.4 Primary Wire Bending	35
Springback	35
Bending Device	37
4.5 Top Electrode	38
It's not a Bug, it's a Feature	38
4.6 PCB Enclosure	39
PCB	39
5 Printed Circuit Board	41
5.1 Component Selection	41
ICs	41
Passive Elements	41
Peripherals	42
Miscellaneous Parts	42
5.2 Layout	42
5.3 Manufacturing	43
Exposure	43
Stripping, Etching	43
Holes and Rivets	44
Assembly	44
Potential Problems	44

MIDI Interrupter	47
6 Hardware and Setup	49
6.1 The Microcontroller	49
6.2 Programming Environment	49
6.3 The Schematic	50
7 PWM Generation	51
7.1 Timers	51
Mode of Operation	51
Prescaler	52
Output Compare Pins	53
7.2 Using the Timers	53
7.3 Signal Combination	54
8 MIDI	57
8.1 What is MIDI	57
8.2 Standard MIDI File	57
Events	57
Delta Time	58
Running Status	59
8.3 Simplifying the MIDI files	59
9 External Storage	61
9.1 Interfacing the SD Card	61
Initializing	62
Reading from the Card	62
9.2 File System	62
FAT	62
FatFs	62
10 User Interface	65
10.1 The Display	65
Communication	65
10.2 The Rotary Encoder	66
Appendix	69
Special Thanks	71
Bibliography	73
List of Terms	75
List of QR codes	77

The Tesla Coil

by Simon Marcher

1

Theory of Operation

The Tesla coil was invented by Nikola Tesla in 1891. His vision was to use this technology to wirelessly transmit power to people all over the world. While his plans did not meet the expectations by far, he still opened up a whole new field of physics, which today helps power the modern world.

Tesla coils come in a wide variety of sizes, power, and modes of operation. From simple spark gap Tesla coils, which consist of only a few passive components, to solid-state Tesla coils, whose only limitations are one's technical skills, every single one amazes anew by combining the fields of high voltage and plasma science.

In order to understand the class-E topology which this thesis is about, we have to understand the basics first.

1.1 The Tesla Resonator

A Tesla resonator, commonly called Tesla coil, is a resonant transformer consisting of two loosely coupled air-cored coils. The primary coil, hooked up to the driver circuit on one side and grounded on the other, is usually made out of a few turns of thick wire. It is placed around the bottom of the secondary coil, either shaped like a flat spiral, a concentric cylinder, or at any angle in between. The secondary coil on, the other hand, is always shaped cylindrically and usually has a few hundred to a few thousand turns.

While in most cases, parasitic effects of components are undesirable, a Tesla coil makes use of exactly that. The effect relevant to a Tesla coil's operation is the parasitic capacitance. A capacitance is just two separated voltage potentials, which can be found along every single winding of a coil. Figure 1.1 illustrates this effect and shows that every coil is in fact an LC oscillator. The lower the inductance and capacitance of the coil, the higher the resonant frequency. Typically it is easier to work with larger capacitances, which is the reason why many Tesla coils have an additional top load connected to the secondary coil. Those act as an additional capacitance towards ground and lower the resonant frequency.

If a high voltage, whose frequency is the resonant frequency of the secondary coil, is now applied to the primary coil, the LC circuit on the secondary side starts oscillating and a very high voltage builds up gradually. Depending on the size, efficiency and input power of the Tesla coil, this voltage can range from a few thousand to a few million volts. Once the voltage is high enough to ionize the air around the top¹, it quickly discharges and the cycle starts over again.

1.1 The Tesla Resonator	3
1.2 Exciting The Resonator	4
The Spark Gap Tesla Coil	4
Solid-State Tesla Coils	4
1.3 Class-E Amplifier	7
Theory of Operation	7
Class-E for Tesla Coils	8
1.4 Musical Tesla Coils	8

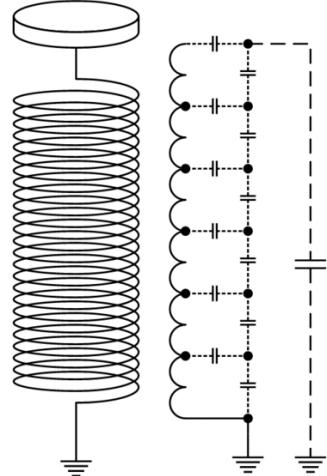


Figure 1.1: Stray capacitances of the secondary coil

1: This usually happens at a designated spark point

1.2 Exciting The Resonator

2: The size of the secondary coil primarily determines its resonant frequency.

Various circuits can be used for the excitation of the coil. Different drivers can be used depending on the desired spark length, size², sensitivity to environment changes, noise level, and efficiency. Back then, Tesla himself only had the resources for building spark gap topologies, but modern technology and solid-state devices allow for sophisticated designs with more control and flexibility.

The Spark Gap Tesla Coil

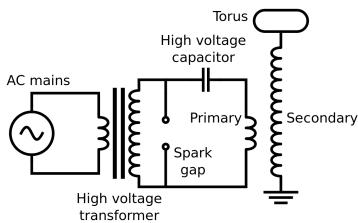


Figure 1.2: A simple spark gap Tesla coil

The simplest and oldest of all drivers is the spark gap driver. In the first stage, the 230 V mains voltage is transformed into a few kilovolts. Neon sign or microwave transformers are popular choices for this task. The capacitor gets charged until the voltage is high enough to break through the spark gap. In this state it has a very low resistance and allows the current to oscillate between the capacitor and the primary coil. This oscillation frequency is usually tens to hundreds of kilohertz and is the same frequency as the resonant frequency of the secondary coil. In every cycle, a small amount of energy is transferred to the secondary coil via the magnetic coupling of the coils. When the voltage in the secondary coil gets too high, a breakout occurs. This can happen one or more times within one period of the AC mains voltage. Once all the energy in the primary circuit has been transferred to the secondary side or dissipated as heat, the spark gap *quenches*, allowing the capacitor to charge up again.

Today, spark gap Tesla coils are primarily used for educational purposes only. Due to their many disadvantages, they are avoided when power efficiency and noise level play a role. The disadvantages include:

- ▶ A spark gap dissipates a lot of power, part of it as heat and part of it in creating ozone, which also poses a health hazard when operated in closed rooms.
- ▶ The rapid ignition and quenching of the spark gap creates a lot of noise.
- ▶ The whole operation cycle only depends on passive component values, making it hard to control the power output or other characteristics.

Solid-State Tesla Coils

3: Unlike relays or hard disk drives, their semiconductor counterparts have no moving components. Hence the name "solid-state".

Aside from vacuum tube Tesla coils, which have existed before, semiconductor technology³ opened up a whole new world of possibilities for Tesla coils. It was now possible to build sophisticated amplifier circuits to excite the primary coil. Some still rely on resonant circuits, others use feedback loops to self-excite the primary coil, and others use ICs to control the driver.

The most common solid-state topologies are:

- ▶ Slayer exciter
- ▶ Half-bridge / full-bridge (single or dual resonant)
- ▶ class-E

Slayer Exciter Circuit

The slayer exciter topology is the one of the most common under hobbyist coilers⁴ because it is the simplest one, only consisting of very few components. At its heart sits the Transistor T controlling the current through the primary coil L_1 . When first powering on the circuit, current flows through R into the base of T , allowing current to flow through L_1 , creating a magnetic field⁵. The rapid change in the field induces a high voltage in the secondary coil L_2 . The parasitic capacitance C of L_2 , however, only allows the voltage across it to rise slowly, therefore pushing the potential of the lower end of L_2 to a negative voltage. As soon as the voltage falls below -0.7 V , the diode D starts conducting and limits the negative voltage while providing current to charge up C ⁶. Additionally, since the base-emitter voltage is now negative, T stops conducting. This causes the magnetic field to start collapsing, causing the potential at the base to slowly rise until T starts conducting again. The circuit is now in its initial state, except that C is now charged to a slightly higher voltage, and the process starts over again.

Since the circuit's timing only depends on the values of L_1 , L_2 , and C , the circuit tunes itself, which makes it very stable to operate. However, because of its simplicity, it comes with a few disadvantages. Firstly, while T is conducting, it shorts L_1 , which leads to a high current and therefore power dissipated in T . This lost power can be reduced by choosing R to be relatively high. This makes the base current smaller and, therefore, also the collector current. While this helps reducing the power dissipation, it also reduces the output power of the coil. However, with a bit of extra complexity, this circuit can be made more efficient.

4: This is how people building Tesla coils often call themselves

5: Since the magnetic field flows through both coils in the same direction, the winding direction of the secondary has to be reversed so that the output voltage will not be reversed.

6: The diode could be left out, since the base-emitter junction also acts as diode, but in the most cases this damages the transistor

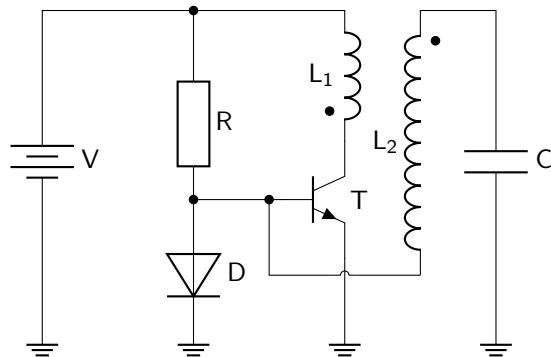


Figure 1.3: Slayer exciter circuit

Half-Bridge / Full-Bridge

This design drives the primary coil by a half- or full-bridge. While this requires large and expensive power MOSFETs or IGBTs as well as carefully designed Gate Drive Transformers (GDTs), it offers a lot of flexibility and power up to many thousand watts.

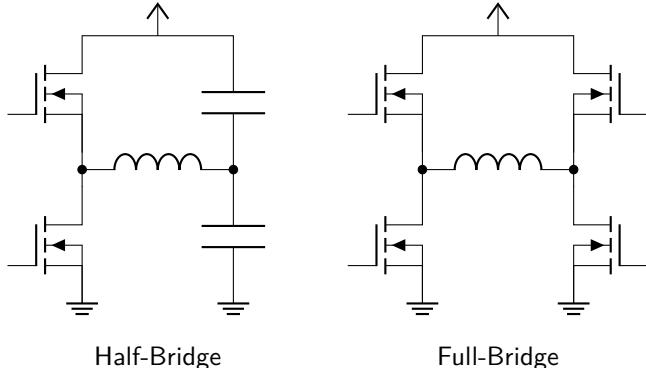


Figure 1.4: Different Bridge Designs

The decision between a half-bridge or full-bridge design is a tradeoff between the number of parts, and the output power. Due to the two capacitors in the half-bridge design, one end of the primary coil is always biased at $V_{CC}/2$, which means that the maximum voltage across the primary coil can also be only $V_{CC}/2$. Luckily, a working half-bridge design can easily be expanded to a full-bridge design, which can then utilize the whole supply voltage.

7: Mostly known as Dual Resonant SSTC or DRSSTC

Both designs can be deployed either single or dual resonant⁷. In order to turn a single resonant design into a double resonant design, a capacitor is added to the primary circuit, which turns it into a resonator with the same resonant frequency as the secondary coil. Since the reactance of an LC circuit is the lowest at its resonant frequency, it is able to draw a lot more current than the primary coil alone, resulting in more power transmitted to the secondary coil.

8: Or other switching devices, like IGBTs

This design is often used for medium-sized to large Tesla coils because of its high output power. However, due to the MOSFETs⁸ switching a high current at a high voltage, a lot of power is dissipated and the system must be adequately cooled. The stress on the MOSFETs can be reduced by interrupting⁹ the Tesla coil. Additionally, this allows the plasma channels to quench during each interrupting cycle, raising its resistance and, therefore, the breakout voltage, which in turn raises the length of the arcs. So this technique lowers the stress on the switching devices, lowers the input power and raises the arc length. The only drawback is that the arcs sound a lot harsher and louder.

9: Turning the coil off and on very quickly

A big advantage of the bridge design is that unless a dual resonant design is chosen, the switching frequency is only determined by the electronics driving the MOSFETs. This allows it to work across a very wide range of frequencies and even adjust to the slightly changing resonant frequency of the secondary coil in real time.

1.3 Class-E Amplifier

The problem with conventional class-B or class-C amplifiers is that with faster switching speed, the efficiency falls drastically. By using a technique called Zero Voltage Switching (ZVS) / Zero Current Switching (ZCS), the class-E amplifier can greatly reduce the power dissipated in the switching device and, therefore (in theory) reach efficiency of up to 100%. In reality, expected efficiencies are over 90% for frequencies of a few MHz and over 60% for frequencies of a few GHz^[1].

[1]: Sokal (2001), "Class-E RF Power Amplifiers"

Theory of Operation

The goal of the class-E amplifier is to reach very high efficiencies for frequencies upwards of a few MHz by minimizing its dissipated power. Power dissipation is the product of voltages times current, so the easiest way to minimize it is to keep either voltage or current close to zero at all times. Typically, high current and voltage occur simultaneously during the switching process, when the switching device has non-zero, finite resistance.

Therefore this kind of amplifier utilizes a reactive load network, which forces the voltage across the switching device to zero just before turn-on and the current to zero just before turn-off. This also means that the switching device is no longer used as an amplifier but as a switch¹⁰ and the load network then shapes the output voltage. Figure 1.5 shows the basic class-E amplifier and figure 1.6 depicts its nominal voltage across and current through an actual MOSFET.

10: This type of amplifier is often referred to as switching mode amplifier

The power dissipation in the switching device can also be thought of as a problem of impedance matching - the power transferred to the load (the switching device) tops out when the resistance of the load is equal to the resistance of the power source (the load network). However, when the resistance of the load is either zero or infinite, no power will be transferred.

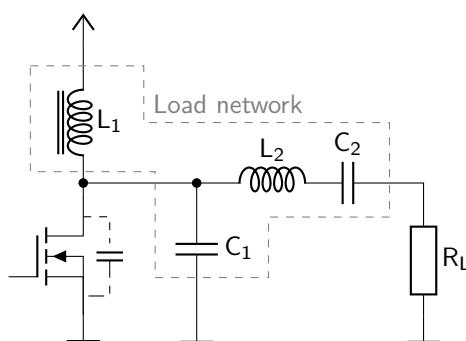


Figure 1.5: Class-E amplifier

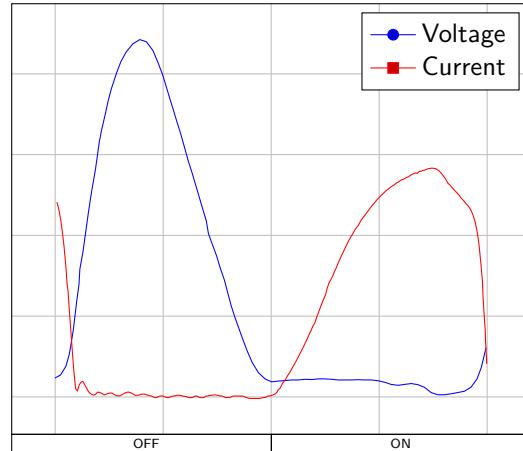


Figure 1.6: Nominal Class-E waveforms

11: When driving a mismatched load, this number only gets bigger

However, one big problem with this type of amplifier is that the voltage and current quickly exceed the safe operating range of the switching device. The voltage can get up to four times the supply voltage and the current up to three times the supply current¹¹. This is, however, dependent on the duty cycle, which should also be taken into consideration when designing such an amplifier. However, because this would add a lot of complexity to both the design process and the driver circuit, the duty cycle was defined to be 50%.

The Class-E Amplifier as Tesla Coil Driver

So far, the class-E amplifier has only been discussed for driving purely resistive loads. A Tesla resonator, however, is everything but that. When taking into account factors like mutual inductance, stray capacitances of the secondary coil, and the feedback from the secondary coil in general, it is impossible to assign a single impedance value to the load that holds true for more than a single frequency. Ideally, all harmonics should have been filtered out by the time they reached the Tesla resonator, but in reality, the few harmonics that make their way through make it impossible to form an equivalent circuit.

Luckily, there are already well-documented projects about class-E Tesla coils online, like the ones from Eirik Taylor^[2], Richie Burnett^[3] or Steve Ward^[4]. Those projects mostly used the trial-and-error method for component values to get the best result.

1.4 Musical Tesla Coils

Musical Tesla coils have been getting a lot of attraction since the mid-2000s and are probably the reason for the popularity of Tesla coils in general. To many, it is simply mind-blowing how a single arc can produce anything musical. But the physics behind it is, in fact, very similar to that of a conventional speaker using a membrane.

In a Tesla coil's, usual uninterrupted operation mode, the high voltage in the secondary coil causes a plasma channel (the arc) to form, discharging the coil. Since the plasma takes up a much higher volume, it expands

[2]: Taylor (2022), *4.096 MHz Class E SSTC*

[3]: Burnett (2022), *HF-SSTC*

[4]: Ward (2022), *Class-E SSTC*

rapidly and pushes air away from the coil. Once discharged, the plasma channel extinguishes and the secondary coil begins to charge up again. A single discharge creates a “snapping” sound well known from arcs, but in a Tesla coil, this cycle repeats many times a second, resulting in a harsh, square-wave-like sound. Mostly depending on the size of the coil, this frequency can be a few hundred hertz to tens of kilohertz; some smaller coils even exceed the human-audible range, creating a noiseless flame-like discharge.

By modulating another frequency onto this carrier frequency, a.k.a. turning the coil on and off at this rate, the frequency is also modulated onto the sound wave created by the arcs. Since the Tesla coil has to complete at least one cycle before being turned off, the modulation frequency cannot be greater than the carrier frequency.

Figure 1.7 shows the arcing of the Tesla coil for the example of a 500 Hz signal modulated onto a 10 kHz carrier wave. Figure 1.8 visualizes the movement of the air molecules around the Tesla coil¹². By performing a Fourier analysis on figure 1.8, a graph of all frequency components of the sound wave can be created, as shown in figure 1.9. It is clearly visible that not only the modulated signal but also the carrier frequency is audible, which distorts the sound. A method to counteract this is by using smaller Tesla coils, which often have carrier frequencies well above the human hearing range.

12: This graph is very likely not accurate and only intended for demonstration purposes

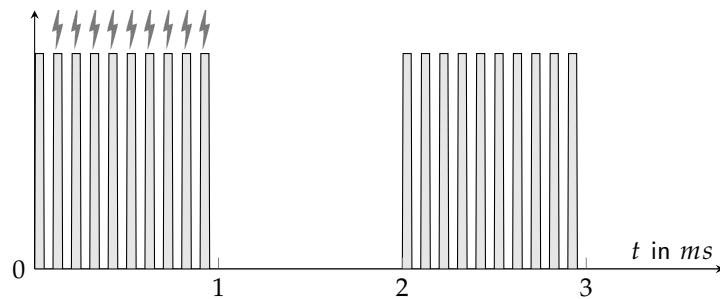


Figure 1.7: Arcing

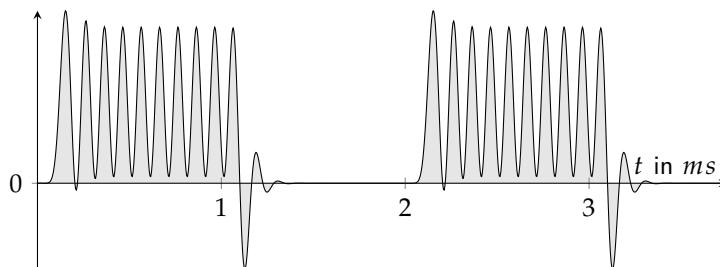


Figure 1.8: Sound Pressure

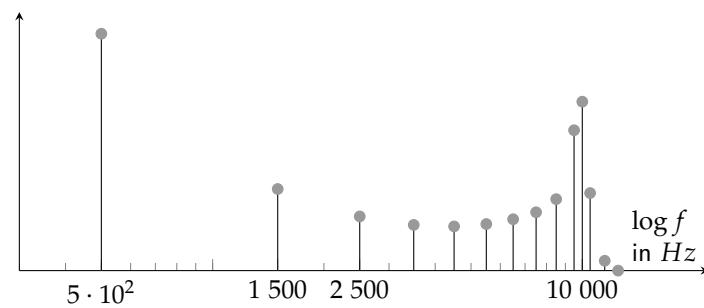


Figure 1.9: Frequency Magnitude

2

Design and Simulation

Disclaimer: This process was a lot more lengthy and tedious than depicted in the following chapter. It mainly consisted of trial and error, which started with a more or less educated guess and ended in cluelessness and despair. But those errors do not add a lot of value to understanding the topic, so this section will not go into much detail.

2.1 The Coils

The goal was to design an air coil with a resonant frequency of 4MHz. This frequency was chosen since it is high enough to work well with a class-E amplifier but low enough to not run into too many RF related problems. It was also already known to have worked with a few other class-E Tesla coils.

Designing the Secondary Coil

The inductance of a single-layered air solenoid coil can be calculated using the well-known equation 2.1.

$$L = \mu_0 \frac{N^2 A}{l} \quad (2.1)$$

The parasitic capacitance, depending on its length l and its diameter D can be calculated as^[5]

$$C_L = \frac{4\epsilon_0}{\pi} \cdot l \cdot \left(1 + 0.8249 \frac{D}{l} + 2.329 \left(\frac{D}{l} \right)^{1.5} \right). \quad (2.2)$$

Defining the length-to-diameter ratio of the coil to be 4, and d_w to be the diameter of the wire, the two previous equations can be rewritten to only depend on l and d_w .

$$L = \mu_0 \frac{l^3 \pi}{32d_w^2} \quad (2.3)$$

$$C_L = \frac{4\epsilon_0}{\pi} \cdot l \cdot 1.49735 \quad (2.4)$$

Those equations can then be set into the formula for the resonant frequency of an LC circuit.

2.1 The Coils	11
Designing the Secondary Coil	11
Designing the Primary Coil	12
Simulating the Coupling	12
Modeling the Transformer	13
2.2 Class-E Design	15
The Design Process	15
2.3 The Class-E Tesla Coil Driver	17
Simulation	18
2.4 Generating the Gate Signal	18
Voltage Controlled Oscillator	18
MOSFET Gate Driver	19
2.5 The Interrupter	19
Fiber Optic Transmission	19
Interrupt Synchronization	21
2.6 Synopsis	21

[5]: Knight (2016), "The self-resonance and self-capacitance of solenoid coils"

$$f = \frac{1}{2\pi \cdot \sqrt{0.1872 \cdot \frac{1}{d_w^2} \cdot \mu_0 \epsilon_0 \cdot l^4}} \quad (2.5)$$

Rearranging this equation to l gives

$$l = \sqrt[4]{\frac{1}{4\pi^2 f^2 \cdot 0.1872 \cdot \frac{1}{d_w^2} \cdot \mu_0 \epsilon_0}}. \quad (2.6)$$

With a frequency of 4 MHz and a wire diameter of 0.35 mm , the length of the coil turns out to be around 100 mm , and the diameter around 25 mm .

Due to the materials available, a 30 mm tube was used as a core for the secondary coil. By using equation 2.1 and 2.2, another equation can be derived, which describes the relationship of all relevant variables.

$$f = \frac{1}{2\pi \frac{D}{d_w} \sqrt{\mu_0 \epsilon_0 (l^2 + 0.8245 D l + 2.329 D^{1.5} \sqrt{l})}} \quad (2.7)$$

Using Wolfram Alpha to solve this equation for l with $D = 30\text{ mm}$ gives a length of 112 mm .

Designing the Primary Coil

The primary coil offers a lot of design freedom and flexibility, but this also means that there is no “correct” or “ideal” design, but only one, which has been observed to work well. It mostly comes down to optimizing the coupling coefficient between the two coils. If it is too low, not enough energy will be transferred to the secondary, but to raise it, the coils must be moved closer together, which leads to flashover due to too low insulation. This limitation can be bypassed to a specific degree by making the primary coil smaller at the bottom, where the voltage is lower and larger at the top, where the voltage is higher. This results in the well-known conical shape known from many Tesla coils.

Simulating the Coupling Coefficient

In order to simulate the resonant transformer, its coupling coefficient has to be known. It describes how tightly the two coils are magnetically coupled and how much of the generated magnetic field of one coil is induced in the other. The simulation tool EleFAnT2D¹ was used to simulate the magnetic flux in the system.

¹: Electromagnetic Field Analysis Tool, developed by IGTE, TU Graz

The coupling coefficient can be described as

$$k = \frac{\Phi_{ind}}{\sqrt{\Phi_P \cdot \Phi_S}}, \quad (2.8)$$

where Φ_{ind} is the flux linkage between the two coils and Φ_P and Φ_S are the flux of the primary and secondary coil, respectively. By using this formula and the simulated values, the coupling factor as a function of the vertical displacement of the primary coil has been calculated and can be seen in figure 2.1.

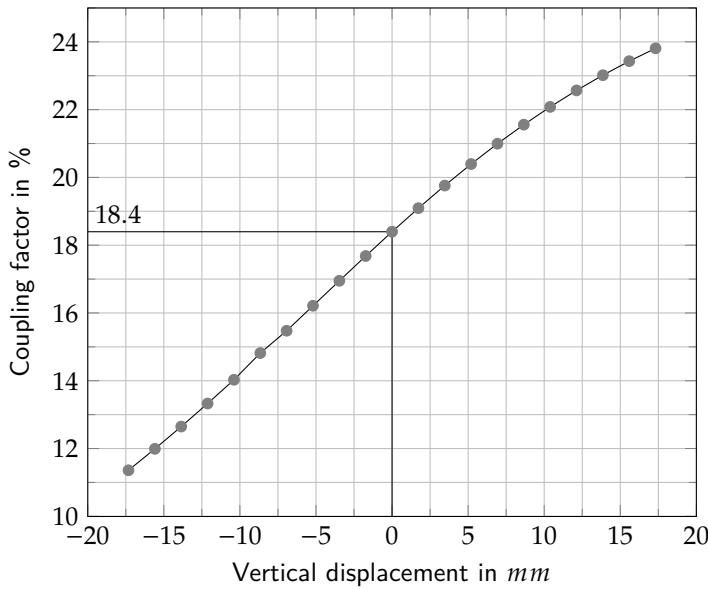


Figure 2.1: Position dependent coupling factor

As expected, the coupling factor gets smaller the further away the primary coil moves from the center of the secondary coil.

Modeling the Transformer

The analog electronic circuit simulator LTSpice, which was used for this project, unfortunately, did not provide any tools to simulate the arc breakout at the secondary coil, which is necessary for testing the continuous operation performance. Also, modeling the magnetic coupling of the transformer involved a trick, which is worth mentioning.

The Magnetic Coupling

To simulate a transformer in LTSpice, the K directive can be used. For example, `K L1 L2 0.3` couples the inductivities L1 and L2 with a coupling coefficient of 30 % [6]. LTSpice then automatically takes care of the stray inductivities and their voltage drop.

[6]: Brocard (2013), *Simulation in LTSpice IV*

The Arcing

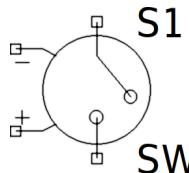


Figure 2.2: Voltage Controlled Switch

[6]: Brocard (2013), *Simulation in LTSpice IV*

When the voltage in the secondary coil V_{SC} is high enough to ionize the air around it, a spark forms. This spark persists as long as the coil can provide enough current to keep it alive. After V_{SC} drops to a very low level, the spark extinguishes and allows the secondary coil to charge up again.

A very basic way to model this behavior in LTSpice is to use a voltage controlled switch with some conditional logic. A voltage controlled switch is an ideal, lossless component that simply connects or disconnects its two terminals depending on the voltage difference of its + and - pins^[6]. The switch is connected in parallel to the secondary coil's capacitance, which gets discharged rapidly when the switch closes.

To specify the switch's characteristics, a new switch model has to be defined. For the example of a switch without a hysteresis, this can look like `.model model_name SW(Vt = 1k)`. This switch is closed above and opened below 1 kV and has an R_{ON} of 1 Ω, R_{OFF} of 1 TΩ, and no additional parasitics.

Unfortunately, it is impossible to use V_{SC} directly as the control voltage because it returns to zero two times per period, which would instantly turn off the switch again. Instead, the switch should turn off when the current falls below a certain threshold. For this, a D-type latch and an arbitrary behavioral voltage source can be used^[7]. This kind of voltage source can be configured to output an arbitrary voltage based on a logical and mathematical expression. The complete spark gap model can be seen in figure 2.3.

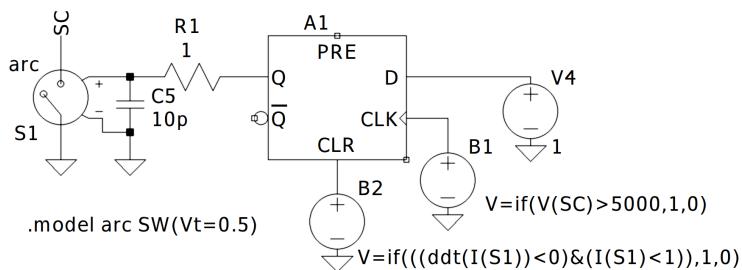


Figure 2.3: Spark Gap Model in LTSpice

V_{SC} supplies a constant HIGH signal to the latch's input. B_1 turns the CLK input HIGH as soon as V_{SC} reaches 5 kV. And lastly, B_2 resets the latch if the current falls below 1 A. The falling edge is essential here because when the switch turns on, the current is still below 1 A, which would turn off the switch again. With the additional `ddt(I(S1))<0` condition, it was made sure that the current has to be decreasing. The low pass filter consisting of R_1 and C_5 adds a tiny delay to the latch's output signal, without which LTSpice would not be able to simulate the circuit correctly. Figure 2.4 shows the voltage V_{SC} after the spark gap has been put into place.

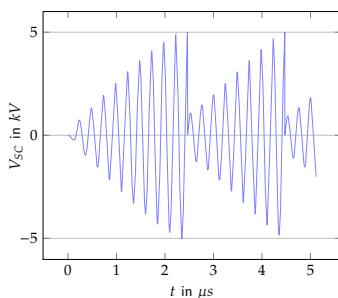


Figure 2.4: Spark Gap Model

2.2 Class-E Design

As already explained in section 11, it is everything but trivial to design a class-E amplifier to drive a Tesla resonator. However, for the sake of demonstration, the following subsection will go through the design process of a class-E amplifier with an ohmic load.

The Design Process

Nathan Sokal, who popularized the class-E amplifier along with Alan Sokal, presented a set of revised design equations in 2001^[1]. Unlike previously published equations, they include the dependence of the loaded Q factor² as well as on the output power.

This amplifier aims to create 50 W of output power in a load of 22Ω at a frequency of 4MHz . The Q factor of the load network has to be chosen by the designer and involves

“a trade-off among the operating bandwidth (wider with lower Q_L), harmonic content of the output power ([...] lower with higher Q_L) and power loss in the parasitic resistances of the load-network inductor L_2 and capacitor C_2 (lower with lower Q_L).”

The chosen value for Q_L is 5 because it is a common starting value for simple class-E amplifiers. The switching device will be a MOSFET, so the saturation voltage used in the equations have already been set to zero. Also, to make the calculations clearer, all numbers, as well as the results, have been rounded to three significant digits. The necessary supply voltage for the amplifier is

$$V_{CC} = \sqrt{R \cdot P \cdot 1.73 \cdot \frac{1}{1 - \frac{0.452}{Q_L} - \frac{0.402}{Q_L^2}}} = 46.1\text{ V}.$$

The expected peak drain-source voltage on the MOSFET is 3.56 times the supply voltage plus a safety factor of around 20%, which means that the MOSFET has to have a drain-source breakdown voltage of at least 197 V.

The choke inductance L_1 , again, has to be chosen in order to calculate the remaining component values. It has to be big enough to force enough current into C_1 and keep the current ripple low. A rule of thumb is that X_{L1} should be at least 30 times greater than X_{C1} . However, since C_1 slightly depends on L_1 , this is an iterative process. A value for L_1 , which turns out to work well, is $130\mu\text{H}$.

[1]: Sokal (2001), “Class-E RF Power Amplifiers”

2: It describes how damped a resonator is, or in other words, how quickly it dissipates its oscillating energy.

C_1 , C_2 , and L_2 can then be calculated with

$$C_1 = \frac{1}{34.2 \cdot fR} \cdot \left(1 + \frac{0.914}{Q_L} - \frac{1.03}{Q_L^2} \right) + \frac{0.6}{(2\pi f)^2 \cdot L_1} = 387 \text{ pF},$$

$$C_2 = \frac{1}{2\pi f R} \cdot \frac{1}{Q_L - 0.105} \cdot \left(1 + \frac{1.01}{Q_L - 1.79} \right) - \frac{0.2}{(2\pi f)^2 \cdot L_1} = 482 \text{ pF}, \text{ and}$$

$$L_2 = \frac{Q_L \cdot R}{2\pi f} = 4.38 \mu\text{H}.$$

Since C_1 is the sum of the capacitor C_{1*} and the MOSFET's output capacitance C_{1M} , this output capacitance cannot be greater than C_1 itself.

Based on the previous calculations, the MOSFET must have

- ▶ a drain-source breakdown voltage greater than 197 V,
- ▶ an output capacitance less than 387 pF,
- ▶ switching times much smaller than 125 ns ($T/2$), and
- ▶ a maximum power dissipation of at least 5 W.

One MOSFET which meets all these criteria is the BSC12DN20NS3. It has a drain-source breakdown voltage of 200V, an output capacitance of 39pF, turn-off and turn-on times of less than 5 ns and a maximum power dissipation of 50W. The capacitor C_{1*} , therefore, is $387 \text{ pF} - 39 \text{ pF} = 348 \text{ pF}$.

The driver was then simulated using LTSpice. The Spice model for the BSC12DN20NS3 can be found in QR code 1. This ZIP file contains a file called OptiMOS3_200V_LTSpice.lib, which contains multiple MOSFET models. To include the correct one into the schematic, a standard nmos has to be added and its value has to be set to BSC12DN20NS3_L1³. The L1 means that switching losses will be included in the simulation.

3: This can be done by right-clicking the symbol.

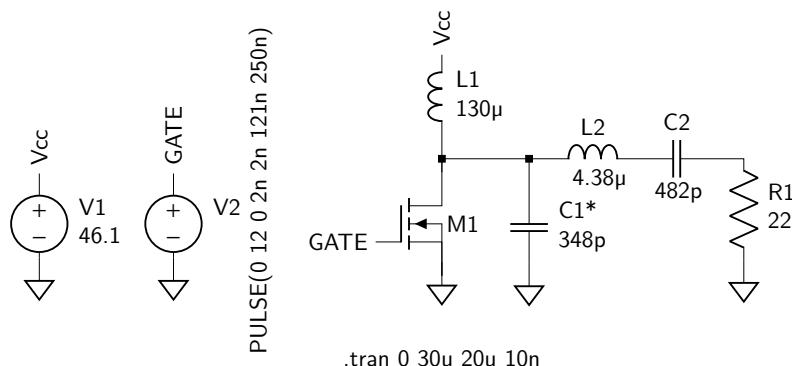


Figure 2.5: Simulation setup in LTSpice

The time-domain for the transient simulation was set to 0 to 30 μs , while only the last 10 μs were recorded to ensure that the amplifier has reached a steady state.

Figure 2.6 shows the voltage across the MOSFET on the left axis and the current through the MOSFET on the right axis.

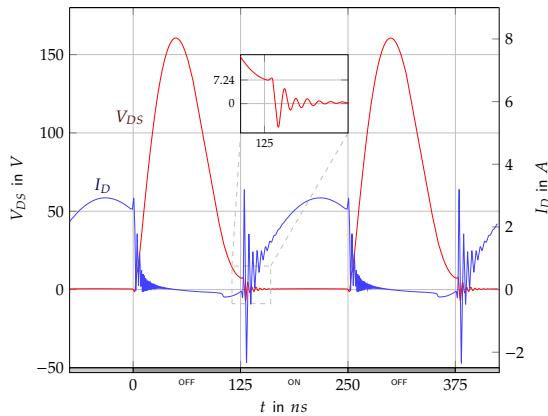


Figure 2.6: Voltage across and current through the MOSFET

While the current does not look as ideal as the one shown in figure 1.6, the overlap between current and voltage is still minimal. The power dissipation in the MOSFET spikes to around 30 W at the transitions but averages out at a mere 625 mW. The total efficiency of the amplifier then is

$$\eta = \frac{P_{in}}{P_{out}} = \frac{46.3 \text{ W}}{48.8 \text{ W}} = 94.9\%. \quad (2.9)$$

This equation uses the averaged powers of the voltage source and the load resistor after a steady state has been reached. By manually fine-tuning the component values, the efficiency can be further increased. Resource [1] is an excellent resource for this.

[1]: Sokal (2001), “Class-E RF Power Amplifiers”

2.3 The Class-E Tesla Coil Driver

Because of the limited engineering experience, some design stages of this project did not go as hoped. Deriving equations for a class-E amplifier with a Tesla resonator as load goes far beyond this education's curriculum. However, it was still attempted.

A very professional tool to design and simulate class-E amplifiers for all kinds of applications is Keysight's *PathWave Advanced Design System (ADS)* and its Class-E amplifier extension, which Keysight was so kind for giving us a free license to. However, without any previous training for this tool, it was hard to use its full potential, which is why LTSpice was preferred.

After many failed attempts to create a simpler equivalent circuit and many scraped hypotheses on how the class-E amplifier could work with a Tesla coil, the priorities shifted from *understanding how it works* to *making it work*. For this, Richie Burnett's project page^[3] has proven very useful.

[3]: Burnett (2022), *HF-SSTC*

An interesting aspect of the above and similar projects is that they describe the class-E amplifier in a completely different way than the theoretical publications do. The filter capacitor C_2 is merely thought of as a DC-blocking capacitor, which shines a different light on its purpose. The capacitor C_1 is just thought of as the value to fine-tune for the best power setting.

Simulation

The schematic presented by Richie Burnett features a class-E output stage, where the primary coil is the load and the load network inductor at the same time. The DC-blocking capacitor C_2 is given with 100nF and L_1 with $47\mu\text{H}$. Blindly inserting these values into the simulation generated surprisingly good results. The optimal value for C_1 has been found to be somewhere between 20 and 150nF depending on the other component's values. The supply voltage was initially set to 50V but later raised to 54V as this was the output voltage of the only suitable boost-converter.

From the simulation alone, it is hard to tell how well the amplifier will work in practice. Also, fine-tuning capacitor values down to picofarads does not make much sense because of the parasitic capacitances introduced by a real-life setup. The best simulation setup by only tuning C_1 can be seen in figure 2.7.

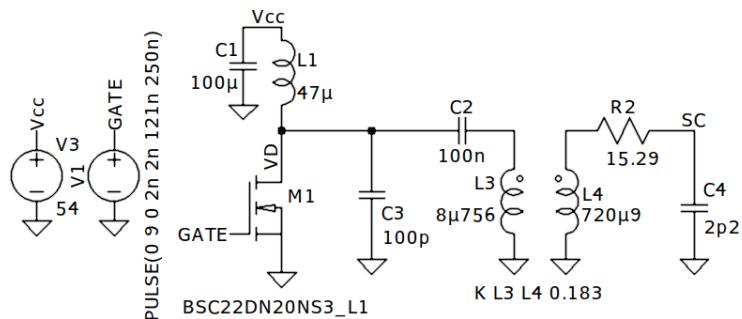


Figure 2.7: Complete Tesla Coil Driver in LTSpice

2.4 Generating the Gate Signal

To start the class-E amplifier, a 4MHz signal has to be applied to the MOSFET gate. The easiest way to generate this signal would be to use a quartz crystal or an oscillator. The difference between those two is that while a quartz crystal is a passive device and needs external circuitry to oscillate, an oscillator only needs to be supplied to output a signal. Also, a crystal generates a sine wave, which needs to be converted into a logic pulse signal by using e.g. a Schmitt trigger, and an oscillator directly generates a logic signal. However, one big disadvantage of both of them is that they have a fixed frequency.

Voltage Controlled Oscillator

Using a variable frequency can be very convenient for tuning the Tesla coil. Every small change in the coil's surroundings causes the resonant frequency to change. Adjusting the operating frequency to the resonant frequency by changing a component's value is easier than vice-versa by changing the coil's number of turns. This is exactly what a Voltage Controlled Oscillator (VCO) is able to do. By far, the most popular one is the NE555. In its astable mode, it can be used to generate a clock signal with variable frequency. Sadly, its maximum frequency lies well below the required 4MHz .

Instead, an LTC1799 was used. Its main frequency can be set by a single resistor and ranges from 100 kHz to 33 MHz . It can be further divided by 10 or 100 by connecting its DIV pin to either $+5\text{ V}$ or letting it float. When connected to GND, which means no division, the output frequency can be calculated as

$$f_{osc} = 10\text{ MHz} \cdot \frac{10^4}{R_{SET}} . \quad (2.10)$$

To set the output frequency to any arbitrary range, R_{SET} can be split up into a fixed and a variable resistor. For example, with a combination of a $23.7\text{ k}\Omega$ resistor and a $5\text{ k}\Omega$ potentiometer, the range can be narrowed down to 3.48 MHz to 4.22 MHz , which should cover the resonant frequency of the coil in any situation.

MOSFET Gate Driver

Due to its high output resistance, this VCO can, however, not supply the necessary power to directly drive the MOSFET at those frequencies. There are, again, a lot of possibilities how this amplification could be implemented, but one of the most straightforward is the use of a MOSFET driver IC. They are specifically designed for this purpose and are able to sink and source large currents to charge and discharge the MOSFET's gate capacitance. The IX4310T achieves this by using a push-pull output, allowing the gate-source voltage to be bigger than the input voltage. With its stated output peak current of 2 A , and rise and fall times of less than 6 ns , this IC can easily drive the MOSFET at 4 MHz .

2.5 The Interrupter

While, in theory, adding an interrupting feature is as simple as turning the coil on and off, a little more thought has to be paid to the actual implementation. For example, switching the power supply on and off would be very impractical and would probably cause issues with too slow rise and fall times. Connecting or disconnecting the primary coil would quickly destroy the class-E stage because of load mismatch. Modulating the interrupting signal onto the MOSFET's control signal on the other side sounds more promising.

Then there is the question about how to generate the interrupting signal. Adding another oscillator for this job would be possible, but this makes it hard to control the interrupter's behavior. Getting the signal from an external source - in this case, the MIDI Interrupter - adds both flexibility and modularity.

Fiber Optic Transmission

Fiber optic cables are a popular choice for sending signals from an interrupter to a Tesla coil. The first reason is that the Tesla coil emits a strong electromagnetic field, which could induce a current in any reasonably long cable. Depending on the output power of the coil, this can even cause

bitflips in a 5 V logic transmission line and damage components due to overvoltage. The second reason is that any failure in the power amplifier or discharges between nearby PCB traces could send a high voltage spike back to the interrupter and destroy it.

Optical fibers, on the other hand, are not susceptible to Electromagnetic Interference (EMI), and if a high voltage is applied to the receiver, it will go into its limit or directly break, but in no way can it induce this same voltage across the transmitter. The used cable was a 1 m long SH-4001 $1000\text{ }\mu\text{m}$ core jacketed optical fiber.

The Transmitter

The used transmitter LED was an IF E91A, which is compatible with any standard $1000\text{ }\mu\text{m}$ optic fiber. It has a peak wavelength of 940 nm , which lies in the near-infrared spectrum and therefore is not visible. When used with a 1 m long SH-4001 fiber, the nominal output power is $70\text{ }\mu\text{W}$ at a forward voltage I_F of 20 mA . At 10 mA , which will be used instead as to not overload the interrupter, only about 85% of this power ($60\text{ }\mu\text{W}$) can be expected.

4: The MOSFET's $R_{DS(ON)}$ of only 1.8Ω is negligible.

To drive the LED, a 2N7000 small-signal MOSFET in a low-side switch configuration was used. With a voltage drop of 1.2 V across the LED, its series resistor at a supply voltage of 5 V should be 380Ω ⁴. Additionally, adding a 330Ω resistor between the interrupter and the gate of the MOSFET limits the gate current to 15 mA while still maintaining a high enough transient frequency.

The Receiver

5: A phototransistor is merely a bipolar transistor with a photodiode connected from its collector to its base

To translate the light pulses back into an electric signal, an IF-D92 phototransistor⁵ has been used. At the wavelength of 870 nm , the responsivity is specified as $100\text{ }\mu\text{A}/\mu\text{W}$ and drops to $94\text{ }\mu\text{A}/\mu\text{W}$ for a wavelength of 940 nm . This means that the collector current I_C is expected to be around 5.6 mA .

The phototransistor alone cannot produce a signal large enough to be used for a logic input. This is why it was combined with a RUM001L02 MOSFET to form a *Darlington pair*. This means that the phototransistor was connected to the gate of the MOSFET, which amplifies the signal to 12 V . A pulldown resistor in the form of a trimmer potentiometer has been added to the MOSFET's gate to discharge it and set the amplifier's sensitivity. The source resistance has been set to $4.7\text{ k}\Omega$.

Figure 2.8 shows the transmitter and receiver circuit.

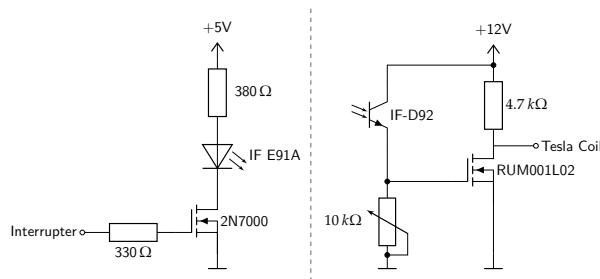


Figure 2.8: Transmitter and Receiver Circuit

Interrupt Synchronization

Directly applying the interrupting technique to the class-E circuit by just turning it on and off at any given moment would violate both the ZVS and ZCS conditions. In the worst case, switching the MOSFET on when the voltage is at its maximum and turning it off when the current is at its maximum a few thousand times a second would result in huge switching losses and device failure. Therefore a security mechanism has to be implemented.

One way to mitigate this problem is by sampling the interrupter signal and only applying it on every falling edge of the 4MHz base signal. The sampling can be done with a falling-edge triggered D-type latch, whose output gets fed into an AND gate along with the base signal.

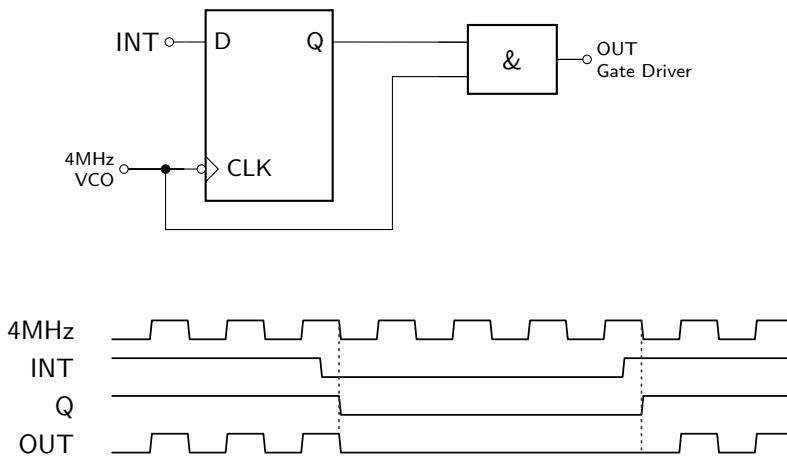


Figure 2.9: Synchronizing Circuit

2.6 Synopsis

Figure 2.10 shows the whole circuitry involved in the Tesla coil⁶. To supply the VCO and the synchronization circuit, the supply voltage of 12 V gets stepped down to 5 V by a linear voltage regulator. On the other side, the power stage needs a higher voltage, which could be achieved by using a boost converter.

6: Except for the optical transmission circuitry

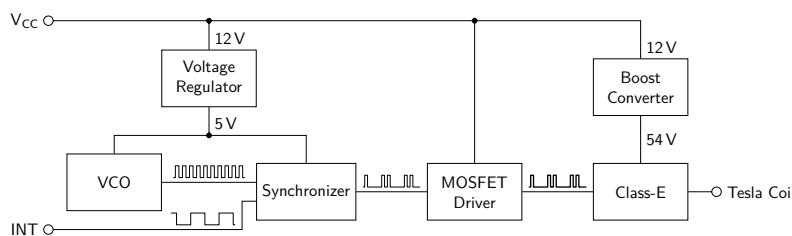


Figure 2.10: Complete Block Diagram

3

Practice and Measurements

3.1 Tuning the Secondary Coil

Tuning the secondary coil to the correct frequency is essential because all calculated values from section 2.1 are highly idealized. For example, equation 2.2 is said to have a standard deviation of σ_{CL} in $pF = 3.6 \cdot D$, which leads to a standard deviation of around 116 kHz of the coil's resonant frequency. In addition, the relative permeability and permittivity of the carrier material is not taken into consideration in the calculations.

The coil was tuned by exciting the base of the secondary coil with a sinusoidal voltage. An oscilloscope probe, formed into a current loop, was placed near the top of the coil. The closer the excitation frequency was to the coil's resonant frequency, the higher the measured voltage on the oscilloscope. The frequency at which the measured voltage was at its maximum was the resonant frequency of the coil. By adding or removing windings, the resonant frequency could be lowered or raised until it exactly matched 4 MHz .

In order to avoid adding windings, which is more tedious than removing them, the coil was initially wound 120 mm long, which is 8 mm longer than calculated. After going through the procedure of sweeping through all frequencies close to 4 MHz , finding out the exact resonant frequency, and tuning it slightly towards the correct value over and over again, the final length turned out to be 113 mm . This corresponds to a deviation of only 0.9% from the calculation.

3.2 The Class-E Stage

The class-E amplifier was by far the trickiest part to get running. It took about seven months from the official project start and four prototypes to get first results.

In the first prototype, it was attempted to design the class-E amplifier using the design equations mentioned in section 2.2. The Tesla coil was converted into a single impedance value and seen as part of the load network. The calculated value for the capacitor C_2 was already contained in this value, so it was simply left out. Guaranteed failure predicted by the simulation was simply ignored. In fact, without the DC-blocking capacitor, the whole amplifier looked like a short for direct current and the network had no chance to start oscillating.

First, it was assumed that this could be solved by reducing the noise in the PCB with a ground plane, which finally was not the solution.

3.1 Tuning the Secondary Coil	23
3.2 The Class-E Stage	23
Fine-Tuning the Setup	24
Measurements	25
3.3 Image Gallery	27

The third design finally solved this problem by adding a 100 nF capacitor, but soon, another issue was discovered. The 4 MHz signal was still coming from a signal generator connected via a coaxial cable. Even though the Tesla coil did not create any arcs yet, it still produced an electromagnetic field. This field induced a voltage in the long cable from the signal generator and, after being amplified by the MOSFET driver, turned on the MOSFET when it should have been off.

Initially, this was believed to be black magic, but the fourth design tackled this flaw by adding the previously described VCO very close to the MOSFET driver. With some tweaking, it was able to produce arcs, even without any help from a grounded object.

The next step was to involve the interrupter signal. The first design did not use a latch, but two NPN transistors, which form an AND gate. This does, of course, not take care of the synchronization between the interrupter and base signal. Another bad decision was using current-driven transistors, which usually need to be handled with care in logic circuits. Even though the calculated resistor values seemed to be correct, the VCO always broke because the base of the connected transistor apparently drew too much current.

The signal synchronization and current issues were fixed by the sixth and final design. Its schematic corresponds to the one derived in chapter 2. When operated in continuous mode by tying the interrupter signal to V_{CC} , the coil behaved just like expected. In interrupted mode, the coil was able to play a single, if not only very distorted, frequency. Unfortunately, due to time constraints, this was as far as it got and the project had to be put temporarily on hold.

Fine-Tuning the Setup

1: And this is true for almost any Tesla coil

It is very unlikely that a class-E Tesla coil¹ can achieve a stable operation on the first try. There are many different parameters that influence the behavior of the coil, but three have been found to have a major impact.

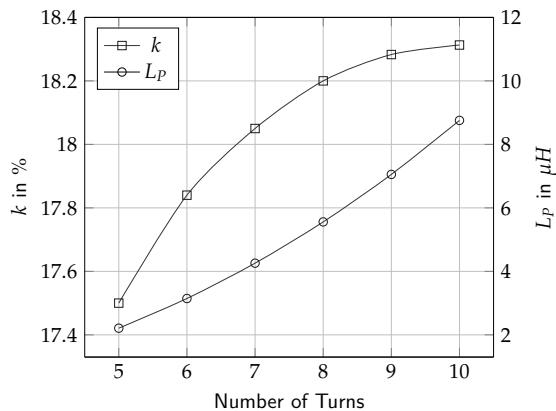
Number of Primary Windings

The turns ratio of a transformer is one of its most important properties, and a Tesla transformer is no exception. With only ten turns, adding or removing a winding greatly affects the primary coil's inductivity as well as the transformer's coupling factor. This relationship has been simulated in EleFAnT2D and is portrayed in figure 3.1.

It has been shown that using between 7 and 8.5 windings yielded the best results. With 10 windings, the amplifier does run stable but cannot produce enough output power to create an arc.

Operating Frequency

As already observed in simulations, the overall efficiency falls drastically as the operating frequency diverges from the secondary coil's resonant frequency. With the ability to change the frequency generated by the

**Figure 3.1:** Tuning the Primary

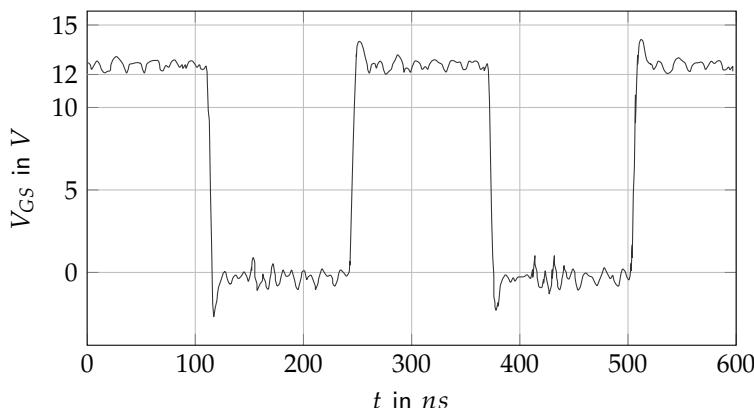
LTC1799 while the coil is running, it was relatively easy to determine the optimal frequency for each setup. In almost all cases, it was located somewhere between 3.7 MHz and 3.95 MHz .

Supply Voltage

One last thing whose impact impact on the coil's output power was bigger than expected was the supply voltage. While in theory, the output power should be proportional to the square of the voltage, raising V_{CC} from 50 V to 54 V roughly doubled the maximum stable arc length.

Measurements

In the earlier prototypes, it was always important to monitor the MOSFET's gate signal. For one, to confirm that the signal generation worked as intended and for another, to measure the current operating frequency. Figure 3.2 shows the gate signal for a frequency of 3.85 MHz .

**Figure 3.2:** MOSFET Gate Signal

Even while the Tesla coil was operating and generating arcs, it was not directly obvious whether the class-E amplifier operated in a stable manner or if the MOSFET could break any moment. Therefore it was crucial to be able to keep a look at the voltages in the circuit, especially V_{DS} . If the drain-source voltage was either clipped, did not return to zero before turn-on, or looked problematic in any other way, it was safer to quickly turn off the Tesla coil and look for potential issues. Figure 3.3 shows the V_{DS} with a supply voltage of 50 V and an operating frequency of 3.85 MHz.

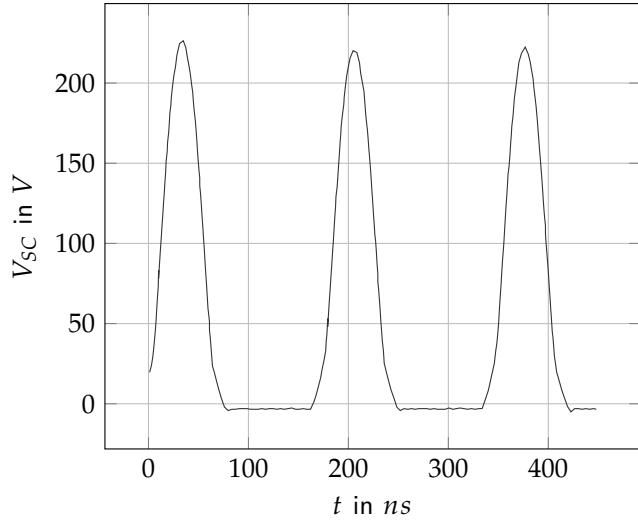


Figure 3.3: V_{DS} at 50 V and 3.85 MHz

As already mentioned, increasing the supply voltage to the indented 54 V clearly increased the output power, but it also pushed the MOSFET to its limits, as shown in figure 3.4. This can be seen by the abrupt drop of the second and third half-wave. Also, the ringing after turn-on and the different heights of the voltage peaks are a clear indicator that the amplifier is not running ideally.

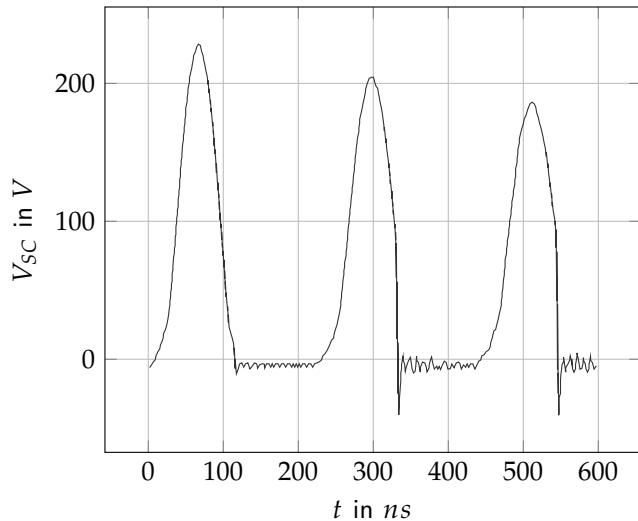


Figure 3.4: V_{DS} at 54 V and 3.85 MHz

3.3 Image Gallery

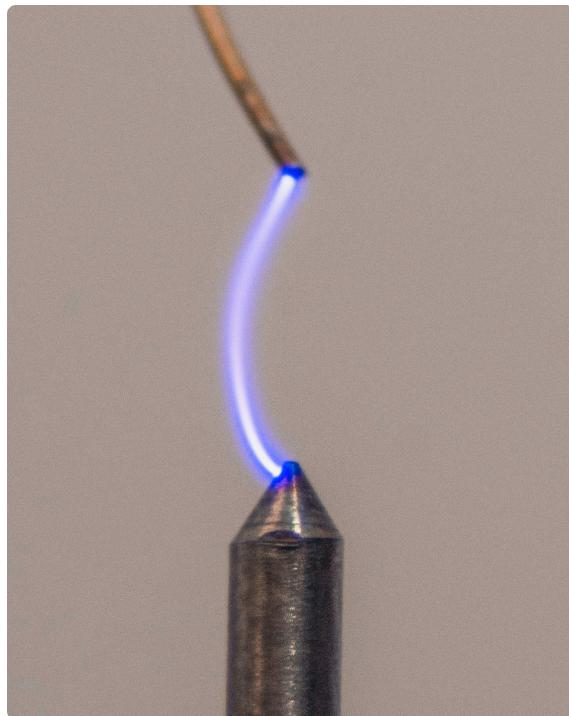


Figure 3.5: Close-Up of the Arc



Figure 3.6: Lighting a Lamp Wirelessly

Design and Building

by Kassandra Peyer

4

Casing

It might seem like the casing is not a very significant part of this project, but the design and function of all kinds of devices' exteriors are crucial. Even the simplest casings often have multiple design stages and variations with different strengths. Since the Tesla coil is intended as a consumer product, the casing plays an essential role in its appearance. However, apart from visuals, there are many safety hazards, which the casing has to account for. Most notably is the genuine fire risk, which goes along playing with open plasma.

4.1 Concept design

There are four main parts that are essential to consider while constructing the Tesla coil - the primary coil, the secondary coil, the PCB, and the top electrode. Theoretically, only the placement of the two coils are relevant for its correct operation, but a reasonable practice is to keep everything as close together as possible.

As shown in figure 4.1, the secondary coil is perpendicular, and the primary coil is angled at 30° to the horizontal plane. The primary coil design choices have already been detailed in the previous part at the end of section 2.1. The secondary coil has a simple cylindrical form as this is the most straightforward and well-proven design.

The Tesla coil has to provide a breakout point for the arc, which in theory could be any sharp conductive object, even the secondary wire itself. However, due to the high temperatures involved, it would quickly melt away and a separate top electrode is needed. It has to be out of a high temperature resistant material that is also conductive. The placement for the top electrode seems very obvious, but it is not without a reason. As the top electrode is directly connected to the secondary coil, it becomes a part of the secondary coil and changes its properties. It is desirable to keep this effect as small as possible by keeping the electrode short and close to the coil. If the breakout point of the electrode is placed further away from the secondary, the behavior of the coils will become less predictable the more distance is between them. So the most practical placement is in the middle and a few centimeters above the secondary coil.

Technically, the PCB can be placed anywhere¹ as long as it can be conveniently connected to the coils. However, as a consumer product, the whole device should be as compact and self-contained as possible, which can be achieved by placing the PCB right beneath the coils. This way, everything is kept in one place.

4.1	Concept design	31
4.2	Materials	32
	Metals	32
	Woods	32
	Glass	32
	Plastics	33
	Conclusion	33
4.3	Structure for the Coils	34
	Primary Coil	34
	Secondary Coil	35
4.4	Primary Wire Bending	35
	Springback	35
	Bending Device	37
4.5	Top Electrode	38
	It's not a Bug, it's a Feature	38
4.6	PCB Enclosure	39
	PCB	39

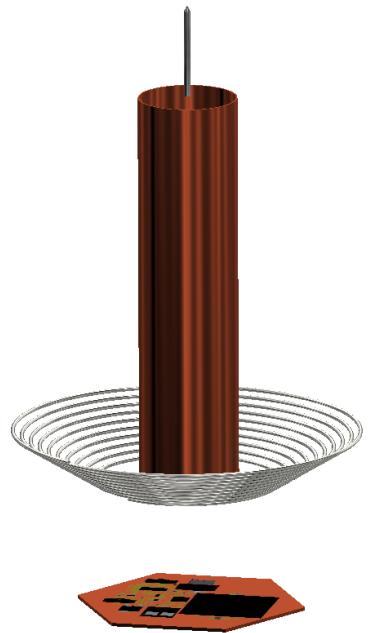


Figure 4.1: Concept design

1: However not too close to the coils to avoid EMI.

4.2 Materials

Selecting the right materials for every part of the coil plays a crucial role for its correct operation. There are a lot of details and possibilities that have to be considered when designing a Tesla coil. Many factors determine the best working material, most affecting the coils directly and therefore being unsuitable, but others were discarded because of their low availability or their high costs.

Four criteria have been established of which two, namely the temperature resistance which has to be considered for the top electrode as well as its surroundings, and the magnetic permeability, which is important around the secondary coil. Unfortunately, due to the limited budget, some materials also turn out as unsuitable because of their price. The fourth criterion, the design aspects, were given the least priorities.

Metals

Metal is a popular choice for building high-grade casings because of its sturdiness, low price and good machinability. However, metals are rather unsuitable for a Tesla coil, mainly because of their electrical conductivity, which would pose an unnecessary safety risk to the user when touched. On one side, because some internal connection error could in the worst case be lethal, but also because if not properly grounded, static electricity could build up in the casing and shock everyone touching it. Another issue is that metals have a very high permeability, weakening any magnetic field. This would not be a big problem for the PCB enclosure but would certainly cause issues when used for the supports of the primary and secondary coil.

Woods

One material that is rarely used for casings, especially for commercial products, is wood, mainly because it is more expensive and harder to work with than metal. Even here, wood is not a good option, mostly because of the anxiety involved in keeping an open plasma flame near a piece of inflammable wood. The same is true for the PCB enclosure, because the high voltages present in the circuitry's are in the worst case more than capable of creating a spark.

Glass

Glass is an electrical insulator does not interfere with magnetic fields, has very high-temperature resistance, and has a very polished look. It also comes in various opacities and surface finishes making the casing's appearance very customizable. The big disadvantage tough is its high production costs, which eliminates it as a possibility.

Plastics

The only remaining option for the casing's material would be some sort of plastic. The two types of plastic which were available in the workshop, were PVC-P and PMMA². There is also PVC-U, which is superior to PVC-P in some aspects, but less available and harder to process. PVC-P is cheaper and more widely available than PMMA, however inferior in some other critical regions.

One problem with PVC-P is its water absorption because due to water's diamagnetic properties, it heats up in an oscillating magnetic field, i.e., it dissipates energy. Its water absorption over 24 hours lies at a maximum of 1% of its total weight. PMMA, in comparison, has an absorption factor of at most 0.4%. On the other hand, PVC-U has an even more ideal water absorption levels of under 0.1%^[8]. The casing should not be made out of PVC-P or PVC-U because of their poor temperature resistance, which start to deform at around 60°C^[9], while PMMA only starts at about 105°C^[10]. PMMA is not ideal in any way, but due to its easy availability and better properties compared to the other options, it was chosen for the casing.

The most ideal plastic would be PTFE³, which has a maximum of 0.01% water absorption, a ten times better than PVC-U. The temperature resistance is also superior than PMMA because it can be used at working temperatures up to 280°C^[9]. However, PTFE was not used, because it has not been taken into consideration until very late projects stages, but will definitely be considered in future builds.

2: also commercially known as acrylic glass or plexiglass

[8]: Omnexus (2018), *Water Absorption 24 hours - Absorption Properties of Polymers*

[9]: (2014), *Fachkunde Metall*

[10]: Wilkes (2005), *PVC Handbook*

3: commercially also known as Teflon

[9]: (2014), *Fachkunde Metall*

Conclusion

Figure 4.2 summarizes all previously discussed materials by giving them a subjective score from one to ten in the four major categories. The total score shown in the circle is then determined by taking the average points of these categories, and visualizes which material is best suited for the Tesla coil.

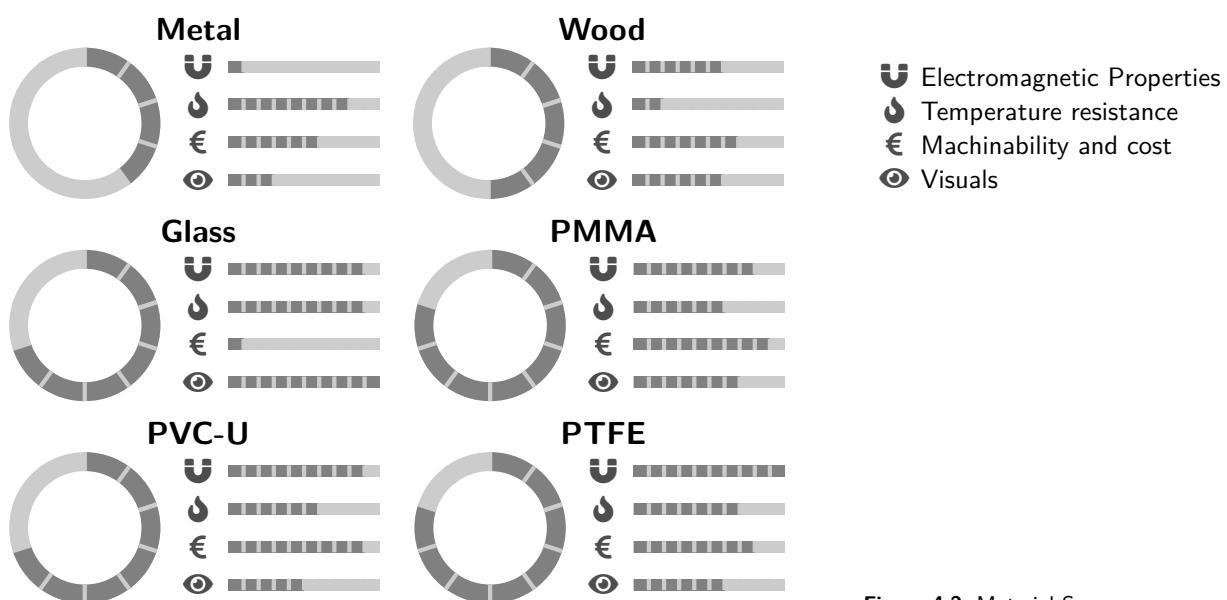


Figure 4.2: Material Score

4.3 Structure for the Coils

As mentioned in section 4.1, both coils have a fixed position and shape, but to ensure this, the casing has to be built around them. The secondary is easy to support, but the primary has a unique shape that has to be kept in form.

Primary Coil

The primary coil's holding structure consists of six supports arranged in a hexagonal form. Every support is a sloped plate with an angle of 30° , on top of which the primary coil rests. In order to keep the coil in place, small half-circles have been cut out for every winding. To keep the coil's shape, the grooves on each support have to be adjusted to the current radius of the coil.

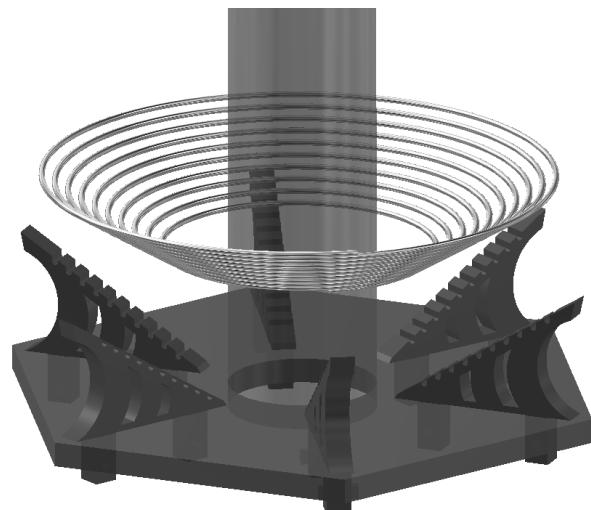


Figure 4.3: 3D view of the primary supports

To avoid gluing the supports to the base plate, a bolt was used to keep it in place, as shown in figure 4.4. This mechanism was mainly designed for testing purposes as it is easier to assemble and disassemble. In a possible future version or commercial release, it would be safer if the supports were glued on. Also, as shown, the supports have a unique design. This was mostly done to make them look elegant and not stand out too much. That makes them lighter, but given that the supports are just a tiny fraction of the Tesla coil's weight, it does not make much of a difference.

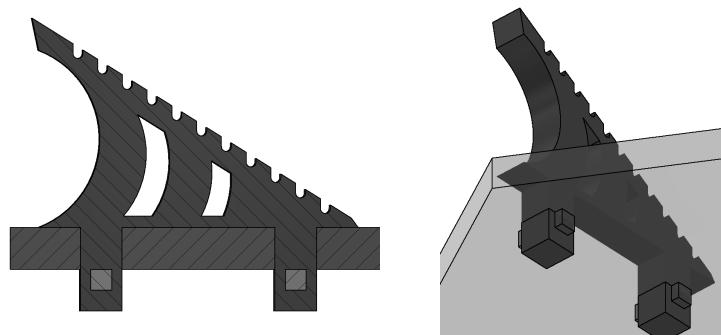


Figure 4.4: Fastening of the supports

Manufacturing

Small and complex structures like the primary supports are best manufactured with an automated process like 3D printing. This makes it possible to print both the supports and the platform they are mounted on in one part, erasing the need for bolts entirely. Because of the way 3D printing works by drawing multiple layers of fine plastic lines, the final part ends up very inaccurate. This makes it unsuitable for the small grooves on the supports, as they will most likely end up too small for the wire of the primary coil.

Another way to manufacture the supports would be to laser cut them out of a 5 mm plastic sheet. This method only allows to construct 2D components, which means that the supports and the platform had to be made separately. One big advantage is its high accuracy, which ensures that the slots for the primary and the bolts fit with only minor post-processing.

Secondary Coil

The core of the secondary coil was one of the easier parts to design because it essentially just consists of a 30 mm PMMA tube with a wall thickness of 3 mm and a length of 120 mm . A few millimeters from the top of the tube, a small hole was drilled to guide the wire to the inside and connect it to the top electrode. The 330 turns of 0.35 mm wire were wound by hand and then coated with protective and insulating varnish to prevent the wire from loosening and cross-arching to occur.

4.4 Primary Wire Bending

After the supports for the primary coil have been designed, the coil itself still has to be bent. Simply bending the wire by placing it in the grooves one after another is not an option because it takes a considerable amount of force to hold a tensioned 1.2 mm wire in place⁴. This force is due to the elastic component of the wire's deformation, which tries to bend the wire back to its original shape.

4: This force could be brought up by glue, but that would be very unprofessional

Springback

5: Especially not for silver-coated copper wires

To remove this component, the wire has to be overbend by a specific angle and to a specific radius. Those values are determined by the springback factor k_s , and the desired angle and radius, which remain due to purely plastic deformation. k_s highly depends on the material and geometry, and usually can be found in large databases.

However, since those are often not freely available⁵, this factor had to be measured by hand. This was done by bending the wire by a specific angle α_1 and measuring the angle α_2 that it sprung back to. Because k_s is geometry-dependent, this measurement had to be repeated for multiple diameters. k_s is then defined as the ratio between α_2 and α_1 . Figure 4.5 shows these calculated values as well as their linear regression fit.

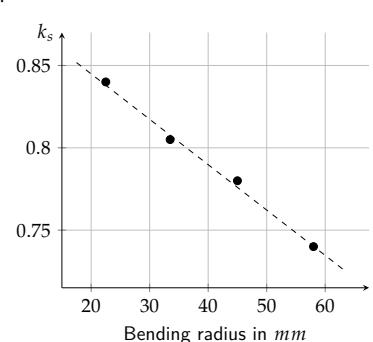


Figure 4.5: Springback factor of the copper wire

[9]: (2014), *Fachkunde Metall*

In the domain of 25 mm to 55 mm , which is the radius of the innermost and outermost winding of the primary coil, the springback factor looks very linear. The equation for the linear regression fit is $k_s(r_1) = -0.002756r_1 + 0.9$, where r_1 is the radius, the wire has to be bent to. The necessary bending radius can be calculated as^[9]

$$r_1 = k_s(r_1) \cdot (r_2 + r_w) - r_w , \quad (4.1)$$

where r_2 is the desired constant radius and r_w is the wire radius. Solving this equation for r_1 gives

$$r_1 = \frac{0.9 \cdot (r_2 + r_w) - r_w}{1 + 0.002756 \cdot (r_2 + r_w)} . \quad (4.2)$$

6: To be correct, the projection of the curve along its symmetry line

In the end, the goal is to create an expression which describes the shape the wire has to be bent into. The first important piece of information is the equation for the curve of the secondary coil¹⁶, which can be described as

$$r_2 = 25 + \frac{1.5}{\pi} \theta . \quad (4.3)$$

Inserting equation 4.3 into equation 4.2, setting r_w to 0.6 mm and simplifying the equation yields that

$$r_1 = 326.56 - \frac{248622}{\theta + 813.557} . \quad (4.4)$$

Figure 4.6 shows r_1 and r_2 side by side for the domain of $[0 : 20\pi]$.

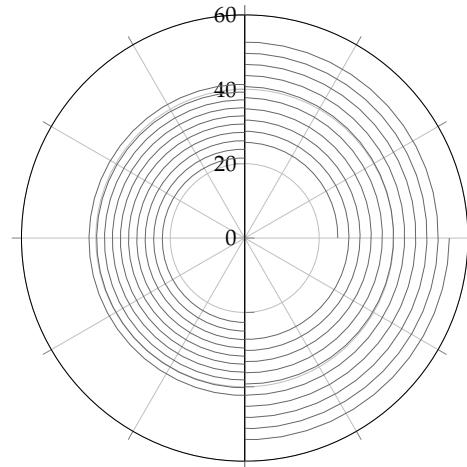


Figure 4.6: Coil Springback

Even though it looks like r_1 has a uniform inclination, the distance between each turn actually falls with an increasing radius. This is because the bigger the bending radius, the more it must be overbent. In this case, the distance between the two innermost turns is about 2.578 mm , while the distance between the two outermost turns is only 2.262 mm .

During the transformation from r_1 to r_2 , the wire not only increases its diameter but also decreases its number of turns. This is because the total length of the wire has to stay the same. The overbending angle α_1 , which

directly relates to the number of turns, is simply the remaining angle α_2 divided by k_s . However, k_s is a function of r_1 , which in turn is a function of α_1 , which is a function of α_2 . Since this does not only sound impractical, another approach was taken.

The domain of r_1 has to be chosen so that its length is the same as the length of r_1 in the domain $[0 : 20\pi]$. To calculate the length of the curves, a WolframAlpha widget⁷ has been used. In the domain $[0 : 20\pi]$, the length of the curve r_2 turns out to be 2.51 m while r_1 is with just above 2 m a little too short. By using the trial-and-error method, it has been found that r_1 needs to have 11.77 turns to achieve the desired length.

However, one remaining problem is that r_2 is only the projection of the actual primary coil. A conic spiral is longer than its flat projection and also needs to be bent, which introduces additional springback. However, the average inclination of the wire is with 1.7 mm per turn, only about 0.0002 degrees, which is negligible.

7: an URL to which can be found in QR code 2

Bending Device

Since bending a wire according to a specific curve equation just by looking at it is rather hard, a bending device was designed. It is just a cone with a spiral-shaped grooving, where the wire can be pressed into. After taking it out again, it bends on the correct form of the primary coil. The grooving was made slightly wider than the wire's diameter to ensure that the wire does not get stuck.

The trickiest part was drawing the path for the grooving in Autodesk Inventor. First, a 2D spiral was drawn by adding a polar equation curve for r_1 , which was then projected onto a conical plane as shown in figure 4.7. Designing the rest of the bending device was relatively straightforward, and figure 4.8 shows a sliced view of the final part.

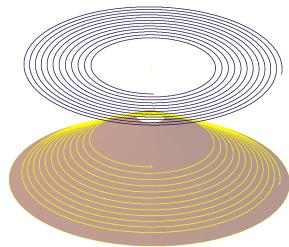


Figure 4.7: Projected Curve for the Bending Device

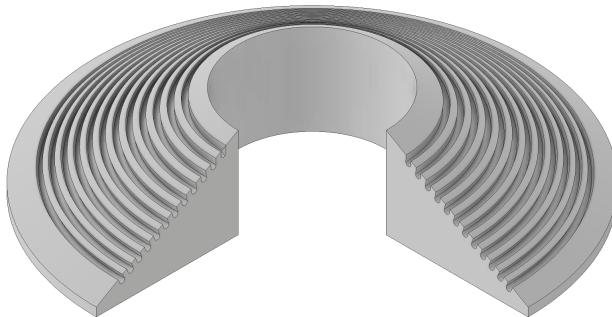


Figure 4.8: Bending Device

Given this device's peculiar shape, which has to be very accurate, the best way to manufacture would be 3D printing. Even though the grooving was made bigger, the wire was still hard to press into the bending device because of how a 3D printer works. In the end, the bent wire of the primary still had to be glued to the supports so it would not fall off.

4.5 Top Electrode

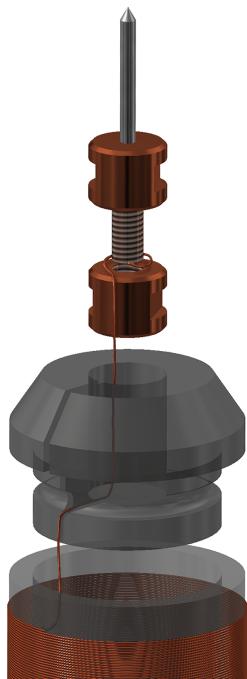


Figure 4.9: Mounting of the top electrode

The top electrode needs to be made out of a temperature resistant and conductive material. It should already be available as thin rod, because turning would not be an option for such a small diameter. Welding electrodes fit these criteria, because they are a few millimeters thick and consist of a tungsten alloy with a melting point of around 3400°C ^[11]. By using a bench grinder, one side of the electrode could easily be whetted to a pointy spike.

The holder that keeps the electrode at the top of the coil is made out of two elements. As seen in figure 4.9, the first one is two copper pieces that are kept together with a thread, and lock the wire tight in between. There are two notches on each one to be able to fasten and open them with a flat wrench. To mount the tungsten electrode, a press fit was drilled into the top surface of the copper piece. The second part is a PMMA piece in which the first part is mounted. It has a slit on one side to be able to take out the copper piece without releasing the wire first.

It's not a Bug, it's a Feature

Because the Tesla coil driver was very under-dimensioned, the plasma flame forming on the top electrode was very small and often had to be ignited and stabilized by another conductor held close to it. This was realized by placing a second, ancillary electrode over to the main electrode. This electrode is held by a canopy-like structure, which rests on three PMMA rods. It was press fitted into a flat copper cylinder, which was loosely screwed into the cap. This way, its height and the gap between the two electrodes can be changed by turning the ancillary electrode.

To further examine the effect of the cap's material and shape, three different versions have been produced. One out of PMMA and two out of Aluminium, one of which is as smooth as possible and one as pointy as possible.

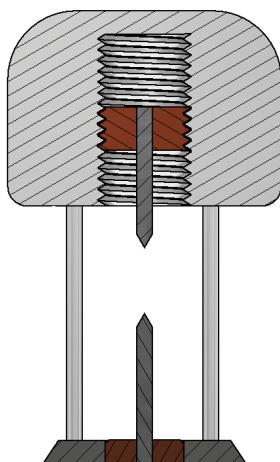


Figure 4.10: Mounting of the ancillary electrode



Figure 4.11: Smooth and Pointy Cap

[11]: Australia Pty Ltd (2019), *Ceriated Tungsten Electrodes For Welding*

While in section 4.2, it was explained that metals pose a safety hazard, this is not entirely true for the secondary side of the Tesla coil. Despite the high voltages present at the top electrode, it is relatively safe to touch. This is because the secondary coil cannot provide enough current and power to cause any physical damage, except the usual burned skin due to the flame's heat.

4.6 PCB Enclosure

The general placement of the PCB, being right beneath the coils, was already addressed in section 4.1. The platform where the secondary coil and the supports for the primary coil are mounted was already designed and chosen to be hexagon-shaped, mainly because it works the best with the six supports. Again, the prototype has the platform resting on top of six pillars to keep everything as modular as possible. The prototype also has only the most essential features and only consists of a framework, this also makes it easier to access the inside of the PCB enclosure. The final design would be completely closed off, and the platform would be glued on as like the primary coil's supports.

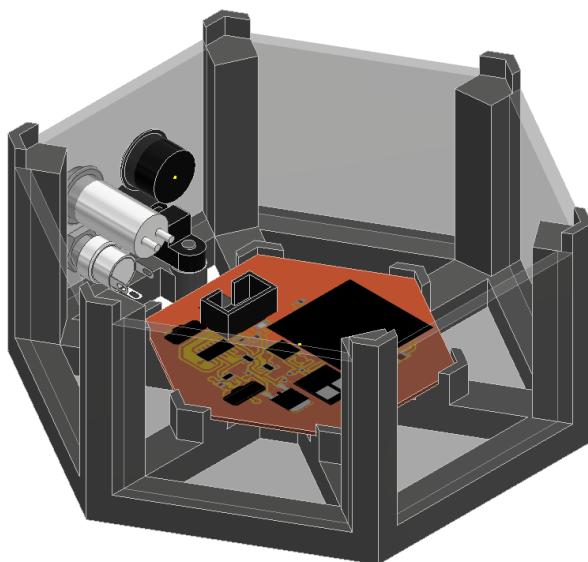


Figure 4.12: PCB Enclosure

PCB

The hexagonal PCB was placed in the center of the enclosure right under the secondary, mostly to balance the design. To ensure that the circuit can also give off heat at the bottom side, it cannot be mounted directly onto the enclosure and therefore has to be raised by a few millimeters. This is done by six steps onto which the PCB is screwed.

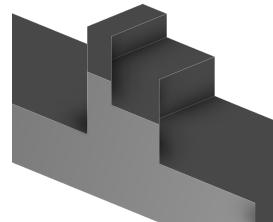


Figure 4.13: PCB Holder

5

Printed Circuit Board

PCBs play an essential role in this project. Breadboards and other prototyping techniques tend to cause issues with parasitic inductances and capacitances and do not work with Surface Mount Device (SMD) components, so PCBs were the only way to build up the circuits. Once a PCB has been produced, it works reliable and usually does not add any erratic effects. On the downside, however, PCBs take a lot of time to design, etch, and solder, and once they are manufactured, they are very inflexible. This means that a new PCB had to be made for every prototype.

5.1 Component Selection

Because almost none of the needed components were on hand in the school laboratories, they had to be ordered online. Due to the unique requirements of some parts, like the MOSFET or the VCO, they were rather hard to find and either very costly or came in an impractical package.

Since all the essential components were only available as SMDs, most other components were also shifted to Surface Mount Technology (SMT) for consistency.

ICs

Most ICs were only available in one package and did not leave much flexibility. The MOSFET driver, VCO, latch, AND gate, and MOSFET for the phototransistor came in standard SOT and SOIC packages which were easy to solder since they have the pins on their side. The class-E MOSFET and the voltage regulator were somewhat troublesome because their packages had a big metal area on the underside, only indented for one-time reflow soldering. Unfortunately, the class-E MOSFET still needed to be replaced multiple times during the testing process.

Passive Elements

All capacitors and resistors have been ordered in a 0805 package because it is not too big but just big enough to be easy to work with. The small size compared to Through Hole Devices (THDs) does, however, negatively affect the power rating of resistors and the voltage rating of capacitors, so it had to be made sure that they still operated within the safe operating range. For the capacitors, X7R devices had been chosen, which means that their operating temperature ranges from -55°C to 125°C with at most 15% capacitance change over this range^[12].

5.1 Component Selection	41
ICs	41
Passive Elements	41
Peripherals	42
Miscellaneous Parts	42
5.2 Layout	42
5.3 Manufacturing	43
Exposure	43
Stripping, Etching	43
Holes and Rivets	44
Assembly	44
Potential Problems	44

[12]: (2022), *Ceramic Capacitors Class //*

Peripherals

To provide the necessary power and lead the MIDI-Interrupter signal to the PCB as well as the ability to be turned on and off without being unplugged out, one side of the PCB enclosure has been dedicated as a control panel. For the power, a 5.5 x 2.1 mm barrel plug was used because of its widespread usage. To connect the fiber optic cable, the phototransistor has been mounted directly into the casing, so it is still accessible from the outside. Additionally, a keyswitch¹ is used to turn on the power, and an ON-OFF-ON switch either routes the output of the phototransistor for interrupted operation, a constant 12V signal for continuous operating, or a 0V signal for no operation to the interrupt input.

1: This way, only authorized people can turn on the Tesla coil and feel important

Miscellaneous Parts

Since the four peripheral components cannot be mounted directly on the PCB, because of its placement they had to be connected via wires. To cleanly organize those connections and be able to efficiently disconnect them all at once, a 6-pin connector was used. The single strands of the flat ribbon cable were then separated to connect to the switches and plugs.

A fuse was added on the Vcc rail to protect the circuit and the power supply from overcurrent. To save space and adhere to the SMD-only strategy, a 1206 package was used.

The boost converter for the 12 V to 50 V conversion was a tricky and very costly part. There were only very few parts for the desired voltage range and most of them were designed for hundreds of watts and came with a corresponding price tag. The final decision fell on the *PKE3316HPI* from *Flex Power Modules* which cost *only* 33 €.

5.2 Layout

After all components were selected and the schematic reconstructed in Autodesk Eagle, a PCB layout had to be designed. The bottom layer was almost completely designated as ground plate, as well as many parts of the top layer. Figure 5.1 shows the top and bottom layer of the final design.

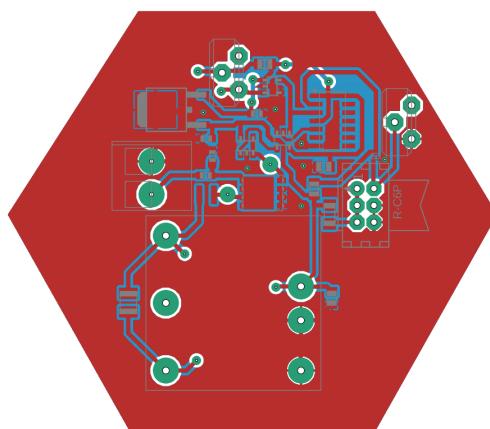


Figure 5.1: Final PCB Layout

5.3 Manufacturing

While PCBs are typically ordered from an external company, this was not possible in this case. Having them produced is very costly and takes more time, than manufacturing them in-house. The workshop's PCB department, although basic, has been a great alternative, primarily because of the many prototypes that had to be produced. Also, the flexibility of controlling every single manufacturing step is impossible to reach when ordering PCBs. On the downside, it lacked some professional equipment, like a solder resist station or a stencil cutter. This made the component assembly process significantly harder and more time-consuming.

The used manufacturing technique was acidic etching and the available PCB blanks were single or double layered, 1.5 mm FR-2 blanks with a $35\text{ }\mu\text{m}$ thick copper layer.

Exposure

The first step was to chemically apply a footprint of the traces to the PCB to the photoresist layer of the PCB. This was done with a designated UV exposure unit. The part of the positive photoresist which gets hit by UV light becomes soluble, while the part covered by a photomask stays insoluble. This photomask was just a semi-transparent sheet of paper with traces printed onto it. It had to be made sure to use as much ink as possible to make it completely lightproof.

The photomask for the top and bottom layer had to be carefully aligned, and it had to be made sure that the printed side of the paper faced the PCB. To further eliminate any possible gap between the ink and the PCB, the exposure unit comes with a vacuum pump to press the paper onto the PCB. Before removing the protective film from the PCB, the room must be dimmed to prevent light from hitting the photoresist beforehand. The exposure process itself then takes about 90 seconds.

Stripping, Etching

Stripping is the process that removes the UV exposed part of the photoresist. The PCB gets mounted in a fixture and sprayed with a sodium hydroxide solution at room temperature for one minute. The room has to be kept dim during this process and can only be lit up afterward. To clean the PCB from any remaining chemicals, it has to be cleaned with water.

The next step is etching away the naked copper areas. This is done by immersing the PCB into a stream of foamed sodium persulfate, which was heated up to 50°C . It takes about 10 minutes for the acid to fully dissolve the copper, but that slightly changes from time to time and it has to be made sure that no unwanted copper is left manually. Finally, the remaining unexposed photoresist has to be removed with propanone².

2: Formally known as acetone.

Holes and Rivets

Even though there are not many, the few Through Hole Technology (THT) devices on the board needed to be mounted in holes. Those were drilled by using a designated PCB drill station. Depending on the component, the diameter reached from 0.9 mm to 1.2 mm . Additionally, a rived had to be added for every via used in the board. Rivets are simply small copper tubes, which can be inserted into a hole and pressed tight to connect the bottom and top layer.

Assembly

The assembly had to be done without solder stop or a stencil. This complicated the whole process, but it was still manageable. To solder the components, a reflow oven has been used. This oven was capable of following a predefined heating curve, which was optimized for small SMD components. A pick-and-place station with a pneumatic pick-up tool and solder paste dispenser simplified the process.

Potential Problems

During this whole manufacturing process there, are many opportunities for things to go wrong. This is a list of the most critical mistakes gathered by personal experience:

- ▶ If the exposure time is too short, the photoresist cannot be fully stripped.
- ▶ If the exposure time is too long, the small amount of light passing through the printer ink will cause the complete photoresist to be stripped.
- ▶ If the stripping time is too short, the copper will not fully dissolve, leaving shorts.
- ▶ If the stripping time is too long, the whole copper will dissolve, leaving nothing.
- ▶ If the concentration of sodium hydroxide or sodium persulfate is too low, the process will take forever.
- ▶ If too little solder paste is applied to a pad, it can become a cold solder joint.
- ▶ If too much solder paste is applied to a pad, it can short nearby pads.

With some experience, however, these issues are easy to mitigate.

MIDI Interrupter

by Felix Hamrle

6

Hardware and Setup

6.1 The Microcontroller

The first logical step for any embedded project is to select the proper hardware. The project's budget was relatively low, and previous experience with microcontroller programming was limited to 8-bit AVR Microcontroller Units (MCUs), which narrowed down the range of options to only a few possible controllers. Additionally, MCUs which are already part of a development board like an Arduino were preferred because they do not need an external programming device.

The biggest challenge was to find an MCU with sufficient PWM capabilities. Each simultaneously played tone requires an independent hardware timer with variable frequency and duty cycle, ideally 16 bit for better accuracy. Apart from the timers for tone generation, one more timer was needed to control the length of each tone. Because of the way the MIDI protocol works, only one timer is needed for any amount of simultaneous notes. Since the human ear is much more sensitive to slightly wrong frequencies than slightly too short or too long notes, an 8 bit timer should be accurate enough. Table 6.1 shows that the ATmega2560, the heart of the Arduino Mega, is best suited for this task because it has the most 16 bit timers.

6.2 Programming Environment

In order to have the maximum amount of flexibility for configuring the hardware, the programming language C++ was chosen over the Arduino Language¹. To avoid possible headaches while writing the code, the whole project was version controlled by git.

The toolchain used for compiling and uploading was the open source tool PlatformIO. It offers a simple interface to program a wide variety of MCUs as well as a command line tool for more advanced tasks.

6.1 The Microcontroller	49
6.2 Programming Environment . . .	49
6.3 The Schematic	50

Table 6.1: MCU Timer Comparison

MCU	Timer	
	8-bit	16-bit
ATtiny85	2	0
ATmega328	2	1
ATmega32U4	1	2
ATmega2560	2	4

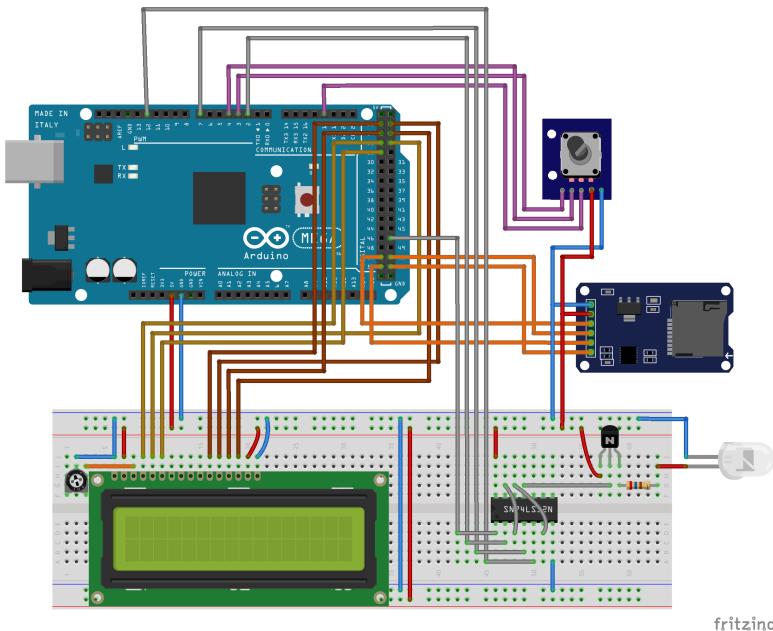
1: Especially because some Arduino libraries tend to break once the hardware it relies on has been configured manually.

Table 6.2: Connections

MCU	Arduino	Device
PB6	12	C 1
PE4	2	C 2
PH4	7	C 3
PL4	39	C 4
PB0	53	SD card CS
PB1	52	SCK
PB2	51	MOSI
PB3	50	MISO
PA4	26	RS
PA5	27	RW
PA6	28	E
PA0	22	D4
PA1	23	D5
PA2	24	D6
PA3	25	D7
PE5	3	LCD CLK
PG5	4	Rotary Encoder DT
PD3	18	SW

6.3 The Schematic

Figure 6.1 shows the complete schematic of the project, created with the open source circuit drawing tool *fritzing*. Additionally, table 6.2 shows all connections in tabular form.

**Figure 6.1:** Complete Schematic

As shown, all devices were connected using a breadboard and jumper wires. Even though special care was taken that the jumper wires did not have a loose contact, they still caused connection issues and led to long searches for alleged software bugs. To stabilize the parts and keep them in position, they were screwed onto a wooden board.

7

PWM Generation

The main task of the MIDI-Interrupter is to create four PWM signals with frequencies within the audio range and variable duty cycles to change the volume of each tone. Those separate channels had then to be merged into a single channel, while preserving the original frequencies as much as possible without distorting them.

7.1 Timers	51
Mode of Operation	51
Prescaler	52
Output Compare Pins	53
7.2 Using the Timers	53
7.3 Signal Combination	54

7.1 Timers

Mode of Operation

The 16 bit timers of the ATmega2560 have five modes of operation. Each one differs in complexity and flexibility. However, one thing they all have in common is the Output Compare Unit. It constantly compares the counter value to the value set in the `OCRnx` register. The unit can be configured to either generate an interrupt or set, clear, or toggle the corresponding `OCnx` pin when a compare match occurs.

1. The **Normal Mode** counts from zero to $2^{16}-1$, sets an overflow flag, and jumps back to zero. Because it can only count at a few fixed speeds and because using a compare match Unit in this mode is not recommended, it cannot produce the desired waveforms.
2. The **CTC Mode**, also called “clear timer on compare match”, counts up until a compare match occurs and then resets to zero. Since the value to compare to can be set by the user, this allows for variable frequencies, however the duty cycle will always be 50%.
3. The **Fast PWM Mode** counts from zero to TOP, which can be set by the user, and then resets back to zero. By setting a compare match somewhere between zero and TOP, `OCRnx` can be set on compare match and cleared at TOP. This way, both a variable frequency and a variable duty cycle can be generated.
4. The **Phase Correct PWM Mode** counts from zero to TOP and then back to zero. While upcounting, `OCRnx` will be set on compare match, and while downcounting, `OCRnx` will be cleared. Since one period in this mode is twice as long as in the previous modes, the timer acts like a 17 bit timer, which doubles its accuracy.
5. The **Phase and Frequency Correct PWM Mode** is almost the same as mode 4, except that `OCRnx` and TOP are updated on BOTTOM instead of on TOP.

Both mode 4 and 5 can either use `ICRn` or `OCRnA` as TOP. `ICRn` is not double buffered, meaning that its value will be updated immediately when written. This could lead to issues, e.g., the timer is upcounting, and TOP is set to a lower value than the counter value. The timer will then miss TOP and count to $2^{16}-1$, which has an impact on the generated PWM signal. A double buffered register like `OCRnA` can be used to mitigate

this issue. In this case, TOP will not be updated until the timer reaches either TOP in mode 4 or BOTTOM in mode 5. The disadvantage of using `OCRnA` is that it is not possible anymore to use the Output Compare Unit A. However, this is not a problem since only one PWM signal per timer is needed.

To summarize, modes 3, 4, and 5 would be able to generate the desired waveforms. However, modes 4 and 5 are twice as accurate as mode 3. Despite their names “Phase correct” and “Phase and Frequency correct”, the latter mode isn’t more frequency correct than the former. They can be equally correct or incorrect depending on whether the TOP register is double-buffered. There is practically no difference between those two, except for the time the registers are updated. The datasheet, however, states that

“It is recommended to use the phase and frequency correct mode instead of the phase correct mode when changing the TOP value while the Timer/Counter is running.”

Since this is the case for the MIDI-Interrupter, the **Phase and Frequency Correct PWM Mode** was chosen. Its timing diagram can be seen in figure 7.1.

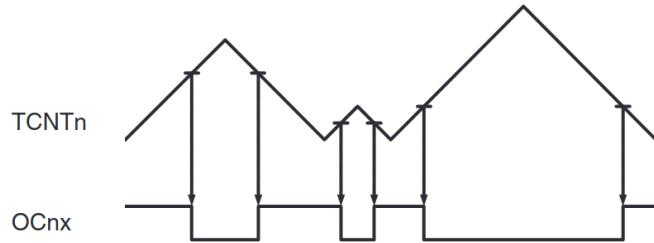


Figure 7.1: Timing diagram of the Phase and Frequency correct PWM mode

Prescaler

The MIDI protocol can encode frequencies from about 8 Hz to 13 kHz, which the timers should cover. To reach this range, a prescaler can be used, which sets the counting speed of the timers to a fraction of the clock speed. Table 7.1 shows all possible prescalers and their resulting frequency ranges.

Note that the maximum frequency is not actually the highest frequency possible for each prescaler. This is because the lower the number of steps, until TOP is reached, the lower the number of possible duty cycles. To ensure that the duty cycle resolution is at least 2% for every frequency, the minimum number of steps for each period was set to 100, which effectively divides the theoretically highest possible frequency by twenty-five.

Table 7.1: Prescalers

Prescaler	Counting Frequency	Resolution	f_{\min}	f_{\max}
1	16 MHz	62.5 ns	122 Hz	160 kHz
8	2 MHz	500 ns	15.3 Hz	20 kHz
64	250 kHz	4 µs	1.91 Hz	2.5 kHz
256	62.5 kHz	16 µs	0.48 Hz	625 Hz
1024	15.625 kHz	64 µs	0.12 Hz	156 Hz

The only prescaler in the desired frequency range was the **prescaler 8**.

Output Compare Pins

Each timer has three output compare units - A, B, and C - each with their own pin, OCnA, OCnB, and OCnC, respectively. Unit B was chosen for all timers, whose pins can be seen in table 7.2.

7.2 Using the Timers

To use the timers in a convenient way, a timer handler has been implemented, as shown in listing 7.1.

```

1 // Constructor
2 PWMTimer::PWMTimer(uint16_t _ocnb_port_addr,
3                     uint16_t _ocnb_pin_addr,
4                     uint16_t _top_addr,
5                     uint16_t _com_addr,
6                     uint16_t _TCCRnA_addr,
7                     uint16_t _TCCRnB_addr,
8                     uint16_t _TCNTn_addr) {
9     // Save register addresses for later
10    top_addr = _top_addr;
11    com_addr = _com_addr;
12    TCCRnB_addr = _TCCRnB_addr;
13    TCNTn_addr = _TCNTn_addr;
14    // Set OCnx as output
15    _SFR_MEM8(_ocnb_port_addr) |= (1 << _ocnb_pin_addr);
16    // Set Phase and Frequency correct PWM Mode with OCRnA as
17    // TOP
18    _SFR_MEM16(_TCCRnA_addr) |= (1 << WGMn0);
19    _SFR_MEM16(_TCCRnB_addr) |= (1 << WGMn3);
20    // Clear OCnB on Compare Match when upcounting, set when
21    // downcounting
22    _SFR_MEM16(_TCCRnA_addr) |= (1 << COMnB1);
23 }
24
25 void PWMTimer::start(uint16_t freq, uint8_t duty_cycle) {
26     isFree = false;
27     uint16_t top = 1000000 / freq;
28     _SFR_MEM16(top_addr) = top;
29     _SFR_MEM16(com_addr) = top * duty_cycle / 100;
30     // Set prescaler to 8 to start timer
31     _SFR_MEM8(TCCRnB_addr) |= (1 << CSn1);
32 }
33
34 void PWMTimer::stop() {
35     // Wait until OCnB is low, before turning off the timer
36     while(_SFR_MEM16(TCNTn_addr) < (top * duty_cycle / 100));
37     // Set prescaler to 0 to stop the timer
38     _SFR_MEM8(TCCRnB_addr) &= ~(1 << CSn1);
39     isFree = true;
40 }
```

Each timer is its own instance of the `PWMTimer` class. To still access all registers, the register addresses were passed to the constructor. Instead of writing, e.g., `DDRB`, which is just defined as `_SFR_MEM8(0x24)`, the macros `_SFR_MEM8`, and `_SFR_MEM16` are used directly with the register address.

To ensure that `OCnx` is low when the timer turns off, which will be important in section 7.3, the turn-off will be delayed until the counter value is bigger than `OCRnx`, which means that `OCnx` has been cleared.

Table 7.2: Output compare pins

MCU pin	Arduino pin
OC1B	12
OC3B	2
OC4B	7
OC5B	39

Listing 7.1: PWMTimer class

Starting and Stopping every timer manually is still a bit tedious, so an additional layer of abstraction was created. This layer was implemented as a namespace, which can be seen as a singleton because it does not have to be instantiated.

Listing 7.2: PWM namespace

```

1  namespace PWM {
2      namespace {
3          PWMTimer timers[4];
4      }
5      void init();
6      void start(uint16_t freq, uint8_t dc = 20);
7      void stop(uint16_t freq);
8  }
9
10 void PWM::init() {
11     timers[0] = PWMTimer(/* List of register addresses */);
12     timers[1] = PWMTimer(/* List of register addresses */);
13     timers[2] = PWMTimer(/* List of register addresses */);
14     timers[3] = PWMTimer(/* List of register addresses */);
15 }
16
17 void PWM::start(uint16_t freq, uint8_t dc) {
18     // Start the first free timer
19     for(PWMTimer & timer : timers) {
20         if(timer.isFree) {
21             timer.start(freq, dc);
22             break;
23         }
24     }
25 }
26
27 void PWM::stop(uint16_t freq) {
28     // Stop the timer playing this frequency
29     for(PWMTimer & timer : timers) {
30         if(timer.freq == freq) {
31             timer.stop();
32             break;
33         }
34     }
35 }
```

Unfortunately, most C++ standard libraries are not available for the AVR platform, so instead of storing the timers in a `std::vector<PWMTimer>`, they had to be stored in a fixed-size C array.

This namespace allows easy control over the timers as shown in listing 7.3.

Listing 7.3: PWM namespace usage

```

1 PWM::start(500); // Starts Timer 1
2 PWM::start(600); // Starts Timer 2
3 PWM::stop(500); // Stops Timer 1
4 PWM::start(700); // Starts Timer 1
5 PWM::start(800); // Starts Timer 3
```

7.3 Signal Combination

At the moment, the MIDI-Interrupter can create four PWM signals on four separate pins. However, the tesla coil needs to be fed a single PWM signal. Unlike regular music consisting of a series of sine waves, logic signals cannot be added as easily. One approach would be to convert every PWM signal into a sine wave where the duty cycle corresponds to the amplitude. Those sine waves are then added together, and for each

point, if the result is above a specific threshold, the output signal is high, else it is low.

However, this would be very computation-intensive task and would likely require an additional MCU. Instead, a simpler yet effective method was chosen - using logic gates. Just like how a specific PWM frequency modulated onto a higher PWM carrier frequency creates a sound that contains both frequencies¹, all four PWM channels can be merged using an OR gate. If only one channel is playing, then it is just passing through without any change. If two or more channels are playing, they get overlapped and therefore are all contained in the output signal.

1: This was explained in section 1.4

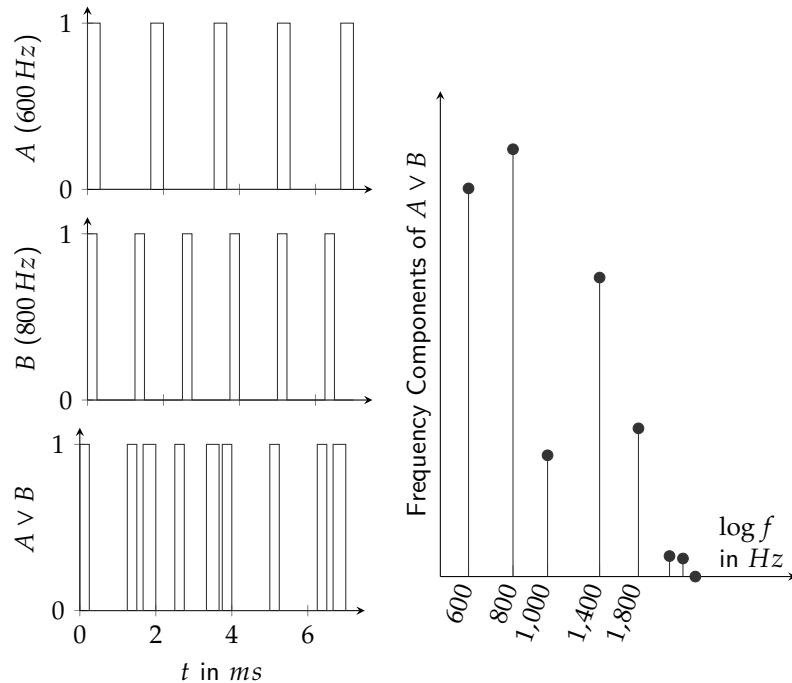


Figure 7.2: Signal Combination

This merging can also be thought of as adding the PWM signals together like analog signals and then just cutting away everything above the previous logic 1 level. However, this also leads to a problem because information is being destroyed every time something is cut away from the waveform. This result is either added unwanted frequency components, removed wanted frequency components, or both. In the worst case, if one frequency is an integer multiple of the other and they are in phase, they overlap so that the lower frequency will get removed entirely. In the same way, if they are exactly out of phase, it will seem like another, higher frequency had been added.

Despite the seemingly bad quality of the output signal, all input frequencies are clearly audible when played on a buzzer, and the distortions hardly could be noticed.

8.1 What is MIDI

Musical Instrument Digital Interface (MIDI) is a protocol for exchanging musical events between devices like synthesizers, keyboards or other electrical instruments. It first appeared as MIDI 1.0 in 1982 and is slowly being replaced by MIDI 2.0, which was introduced in 2020.

Instead of describing the sound of an audio, like, e.g., MP3 does, MIDI describes every tone's pitch, volume, and duration. This enables high flexibility for music production while still being easy to use. The Interrupter will act as a very simple synthesizer that controls the tesla coil to create the described tones for this project.

8.2 Standard MIDI File

All the information in the following section was taken from [13].

Unlike MIDI streams, which connects two or more devices and lets them communicate in real time, the MIDI-Interrupter will use so-called Standard MIDI Files (SMFs), which are binary files holding MIDI data.

SMFs are build up of a header chunk and one or more track chunks. The header chunk stores the file format, number of track chunks, and the division¹. The track chunks then hold the actual MIDI data. How these track chunks have to be read depends on the file format.

0	Single Track File Format
1	Multiple Track File Format All tracks are played simultaneously
2	Multiple Song File Format All tracks are played after one another

The most common format is the Multiple Track File Format. For songs containing multiple instruments, each instrument usually has its own track. In the same way, piano pieces typically consist of two tracks - one for the left hand and one for the right hand.

The track itself is a series of delta times with a following MIDI, Meta, or Sysex event.

Events

Events are instructions for reconstructing the sound of a song. They always consist of an identifier and some amount of data. This amount depends on the event, but some events also have a variable amount of data, where the length is given as the first argument. The three types of events are MIDI Events, Meta Events, and Sysex Events

8.1	What is MIDI	57
8.2	Standard MIDI File	57
	Events	57
	Delta Time	58
	Running Status	59
8.3	Simplifying the MIDI files	59

[13]: (2014), *The Complete MIDI 1.0 Detailed Specification*

1: The time resolution of the file

MIDI Events

MIDI-events contain the actual music information, like `NOTE_ON` or `NOTE_OFF`. They consist of a 4-bit identifier, a 4-bit channel, and (in most cases) 2 bytes of data. For the example of `NOTE_ON` and `NOTE_OFF`, the first byte of the data contains the note number, and the second byte contains the velocity, which can be interpreted as its loudness.

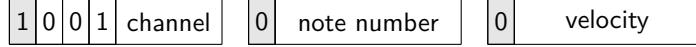


Figure 8.1: Structure of a `NOTE_ON` event

The 7-bit note number can range from 0 to 127, corresponding to the musical notes C-1 to G9. For reference, a typical 88-key piano ranges from A0 to C8 (note numbers 21 to 108). To convert this note number into a usable frequency, equation 8.1 can be used.

$$f = 440 \cdot 2^{\frac{n-69}{12}} \quad (8.1)$$

Meta Events

Non-musical information like song names, lyric texts, tempo settings, or end of track are contained in Meta events. The most common Meta events are `SET_TEMPO` and `END_OF_TRACK`, which have to be every track's last event. Every Meta event consists of a `0xFF`, an identifier, and a variable amount of data.

Delta Time

Delta times specify the time offset to the previous event. They are encoded as Variable Length Quantities, which means they can take up one to four bytes, depending on their size. The MSB of each byte shows if another byte follows, and the remaining 7 bits are the delta time. An example implementation for decoding a Variable Length Quantity in C++ can be seen in the listing 8.1.

Listing 8.1: Decoding the Delta Time

```

1 uint32_t deltaTime = 0;
2 for (uint8_t i = 0; i < 4; i++) {
3     uint8_t byte = nextByte();
4     deltaTime = (deltaTime << 7) | (byte & 0x7F);
5     if(~byte & 0x80) {
6         break;
7     }
8 }
```

The delta time is the number of ticks, which by itself is not very useful. To convert the delta time into seconds, both the division and the tempo are needed. The division is part of the header chunk, and the tempo can be set with a Meta event. Equation 8.2 shows how those values can be used to calculate the actual time.

$$\text{time } [\mu\text{s}] = \frac{\text{tempo} \left[\frac{\mu\text{s}}{\text{quarter note}} \right] \cdot \text{delta time } \left[\frac{\text{ticks}}{1} \right]}{\text{division} \left[\frac{\text{ticks}}{\text{quarter note}} \right]} \quad (8.2)$$

Running Status

One thing all event types have in common is that their first identifier byte has a Most Significant Bit (MSB) of 1, and their first argument byte has an MSB of 0. This way, in case the identifier is omitted, the argument is still clearly recognizable as such and cannot be mistaken for the identifier. The running status scheme now allows it to omit the identifier of an event if it is of the same type as the previous event. For example, after a NOTE_ON event, consisting of a delta time, an identifier, and its argument, a following NOTE_ON event could only consist of a delta time followed by the argument.

Only by combining this with another trick, which is to turn all NOTE_OFF events into NOTE_ON events with a velocity of zero, does this become useful. Now, for a simple MIDI file, almost the whole file would consist of NOTE_ON events stringed together, and only the first one would need an identifier.

ΔTime	N_ON	Note	Velocity
ΔTime	N_OFF	Note	Velocity
ΔTime	N_ON	Note	Velocity
ΔTime	N_OFF	Note	Velocity
:			

⇒

ΔTime	N_ON	Note	Velocity
ΔTime		Note	0
ΔTime		Note	Velocity
ΔTime		Note	0
:			

Figure 8.2: Running Status

8.3 Simplifying the MIDI files

To not have to decode every single MIDI message, most of which are not valuable to the MIDI-Interrupter, a tool called *tinymid*² was used. It removes all unnecessary events, moves them to channel 0, converts them to running status if possible, and most importantly, it merges all tracks of a multiple track file into a single track to make it a single track file. For this, the whole file has to be decoded at once and all events sorted by their absolute delta time, which would have been much harder to do with the limited libraries and debugging capabilities available for AVR C++.

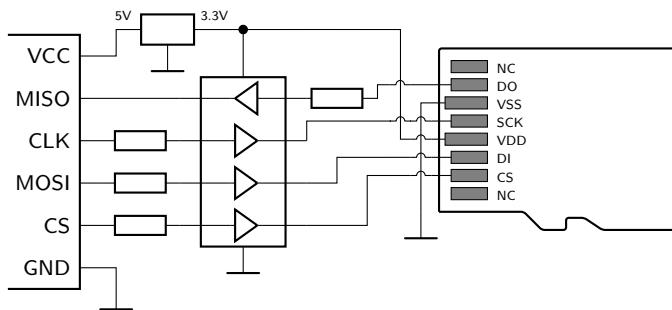
2: Its source code can be found in QR code 3.

External Storage

9.1 Interfacing the SD Card

Most of the information in this section was taken from QR code 4

A Secure Digital Memory Card (SDC) or Multi Media Card (MMC) is an easy way to store data on embedded systems. More powerful devices use the card in SD mode, which allows faster clock speeds, but microcontrollers mostly use the SPI mode, because of the already existing SPI hardware module. Since all SD devices only operate at voltages between 2.7 and 3.3 volts, a voltage translation has to be added in between the Arduino and the device. For this, a SD card adapter from AZ-Delivery was used. By reverse engineering it, the schematic shown in figure 9.1 was reconstructed.



9.1 Interfacing the SD Card	61
Initializing	62
Reading from the Card	62
9.2 File System	62
FAT	62
FatFs	62

Figure 9.1: SD Card Adapter

This adapter has two problems however. First, the voltage translation only goes in one way, so the **MISO** line still operates at 3.3V. Second, the bus buffer gate outputs are configured to be always in low impedance mode, which leads to problems, when more than two devices are communicating on the **MOSI** line.

The commands used to communicate with the SD device are build up of a 6-bit command index, an 4-byte argument and a 7-bit Cyclic Redundancy Check (CRC) as shown in figure 9.2. After a command was send, the SD device sends back a response within the Command response time (NCR), which is at most 8 bytes. During this time **CLK** has to be active and **CS** has to stay low. Then, the card sends a one or four byte response and depending on the command also a data packet up to 2051 bytes long.

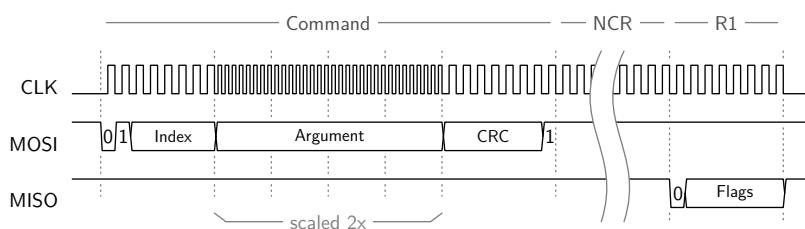
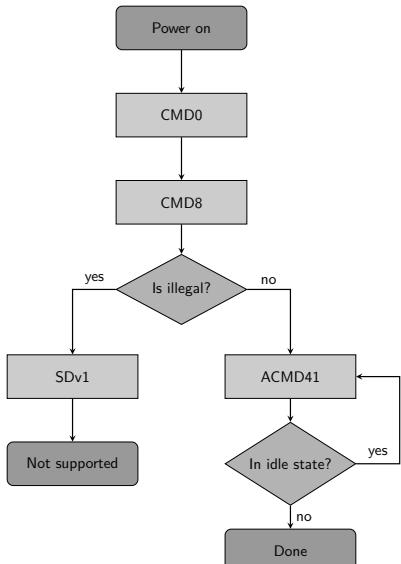


Figure 9.2: Command Structure

Initializing

Before any commands can be send to the SD card, it has to be put into SPI mode by setting both `MOSI` and `CS` high and toggling `CLK` for at least 74 clock cycles. To start the initialization process, a `CMD0` command has to be send, followed by a `CMD8` to detect the kind of device. If it responds with an "Illegal command" flag, the device is a `SDv1`, otherwise it is a `SDv2`. The next step is to send an `ACMD41`, which is just a `CMD55` followed by a `CMD41`, until the card leaves idle mode. This usually has to be done at least twice.



Reading from the Card

The SD card manages its data in so called block, which are usually chunks of 512 bytes. Those blocks can be read one at a time or as a continuous data stream, but the filesystem library Petit FatFs requires the read function to perform single block reads, so multiple block reading won't be discussed further.

In order to read a single block, `CMD17` has to be send to the card along with the sector number as argument. The card should respond with `0x00` and shortly after sends the requested data packet consisting of a `0xFE`, 512 bytes of data and a 2-byte CRC.

9.2 File System

FAT

The File Allocation Table (FAT) file system is one of the most basic file systems. There are FAT 12, 16 and 32, while FAT 12 was only ever used on old devices like floppy disks. Because of its simplicity it is very often used for small portable storage devices like USB drives or SD cards.

The disk starts with the File Allocation Table, which contains a map of all clusters in the system. One cell of the table represents a cluster and shows its allocation status. Following the File Allocation Table are two types of clusters: data and directory cluster. Directory clusters store the structure of the file tree and give information about the files, while data clusters hold the actual file content.

FatFs

In order to interface this file system a very popular library called FatFs was used. Because of the limited resources available on the low-powered microcontrollers, a smaller version called Petit FatFs was used instead. All information about this library as well as its documentation can be found in QR code 5.

As FatFs was written to be as portable as possible it does not provide any communication with the storage device. Therefore the user has to implement the communication layer by themselves. The communication layer has to provide the following functions:

<code>disk_initialize</code>	Initializes the SD card as described in section 9.1
<code>disk_readp</code>	Reads a partial sector at a specific address
<code>disk_writep</code>	Writes a partial sector at a specific address

Using FatFs

Before any operations can be done on the SD card, it needs to be mounted with the function `pf_mount(FATFS* fs)`, where `fs` is the working area of FatFs. Since the file to be played should be chosen interactively, all file information has to be read out first. In order to read a directory it has to be opened with `pf_opendir(DIR* dp, const char* path)`.

To read the directory's content, the function `pf_readdir (DIR* dp, FILINFO* fno)` is used, which reads the next directory entry on each call.

In order to reread all directory entries and save their file names and file sizes, the internal directory index has to be rewound by calling `pf_readdir` and passing a null pointer instead of `*fno`. All important file information can then be saved by again repeatedly calling `pf_readdir`.

After choosing a file with the interactive menu as described in chapter 10, the file has to be read in memory. First, the selected file has to be opened with `pf_open(const char* path)`. After the needed memory has been allocated, the file can be read with `pf_read(void* buff, UINT btr, UINT *br)`.

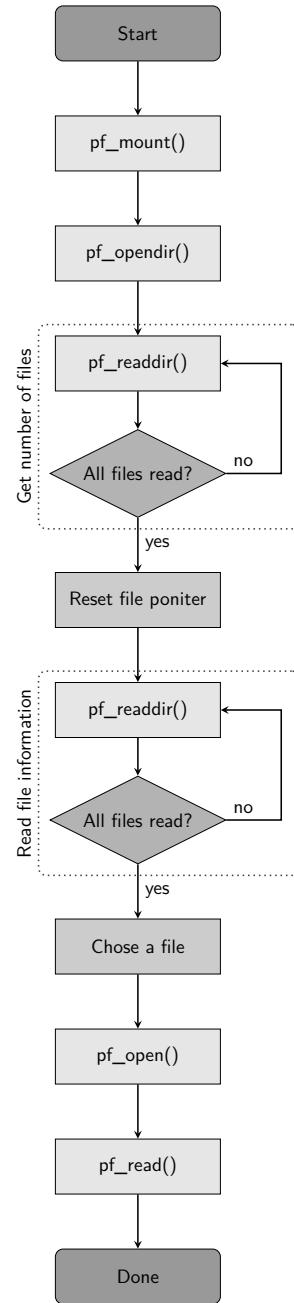


Figure 9.4: Using FatFs

User Interface

After the main features of the MIDI-Interrupter have been implemented, it needs a way to interact with the user. For debugging purposes, this can be done via the Universal Asynchronous Receiver-Transmitter (UART) protocol, but this is not very user-friendly. While there are a lot of possibilities, like a smartphone app which connect via Bluetooth, a Touchscreens display or an high resolution OLED display, the focus was mainly on simplicity. Therefore an LCD in combination with a rotary encoder has been chosen.

10.1 The Display

A Liquid Crystal Display (LCD) makes it very easy to show text or other ASCII symbols due to its builtin HD44780U controller. This controller provides a layer of abstraction to the individual pixels, which makes it possible to send high-level commands to write to the display. Such commands include `clear display`, `return home`, or `function set`.

To connect the display to the MCU, three control pins and either four or eight data pins are required. The pins `D0` to `D7` are the data pins. To save four pins, the controller offers a 4-bit mode in which only `D4` to `D7` are used by serializing the communication and sending the low nibble after the high nibble. The `RS` pin selects if the register to operate on is the data or instruction register, the `RW` pin chooses if this register is read from or written to, and lastly, the `E` pin starts any read or write operation.

Additionally, the `VSS` and `VDD` pins supplies the controller chip with 5V and the `V0` pin sets the contrast depending on the applied voltage. Finally, the `A` and `K` pin are the anode and cathode of the backlight LED and can be directly connected to +5V and ground, since the module already has a builtin series resistor.

Communication

After power on and automatic initialization of the controller, it has to be set to 4-bit mode with the `function set` command. After setting other options, like the size of the display or the cursor shape, text can be printed to the display by writing data to the data register. A complete list of all commands and more details on the initialization process can be found in the datasheet, whose URL can be found in QR code 6.

Instead of implementing this communication layer from scratch, a very popular library written by Peter Fleury has been used¹. In `lcd.h` the used mode and the port and pin number of each pin has to be specified. After that, it is very easy to use the LCD as shown in listing 10.1

10.1 The Display	65
Communication	65
10.2 The Rotary Encoder	66

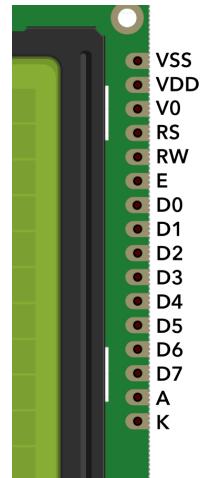


Figure 10.1: Pins of an LCD

¹: A URL to the library documentation can be found in QR code 7

Listing 10.1: LCD Library

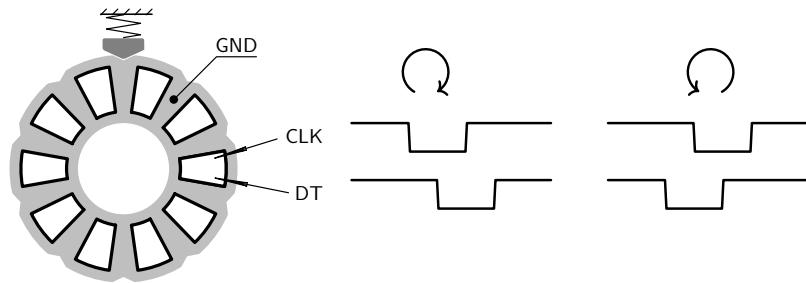
```

1 #include "lcd.h"
2
3 int main() {
4     lcd_init(LCD_DISP_ON_CURSOR); // Initializes LCD
5     lcd_clrscr();             // Clears the screen
6     lcd_home();               // Moves the cursor to position (0,0)
7     lcd_puts("Hello there"); // Prints text to the screen
8     return 0;
9 }
```

10.2 The Rotary Encoder

After the list of files can be shown on the LCD, the user has to be able to cycle through the list and select a file. A relative rotary encoder is the perfect tool for this, because it can rotate endlessly, has clearly noticeable discrete steps, a digital output signal and an integrated button to confirm the selection.

In total, the rotary encoder has five pins - two for the supply voltage, one for the button and two for the rotary encoding, `CLK` and `DT`. While the naming clearly implies that the `CLK` signal should be thought of as the main signal, in practice they are completely interchangeable. Figure 10.2 shows the internal working of the device as well as the resulting `CLK` and `DT` signals when the encoder is turned.

**Figure 10.2:** Internal working of a rotary encoder

2: Setting it on the rising edge would work equally well, however then the following logic would have to be inverted.

The easiest way to read the signal from the decoder is to set an interrupt on the falling edge² of the `CLK` signal and read the `DT` signal at this point. If it's high, the encoder has been turned clockwise, and if it's low it has been turned counter-clockwise.

To confirm the selection, the integrated button, which can be activated by pressing down on the rotary shaft has been used. It was connected to the external interrupt pin `PD3` on the ATmega2560, which was configured to trigger on a rising edge. The de-bouncing was implemented by adding a 50 ms delay in the Interrupt Service Routine (ISR) and clear any pending interrupt on this pin by setting the corresponding bit in the `EIFR` register.

Appendix

Special Thanks ...

... to our supervisors, consultants, and sponsors Thomas Jerman und Michael Lieschnegg.

... to our workshop teachers, especially Hubert Tuscher, Albin Waiker and Konrad Wilhelm.

... to Federico Marotta for providing “kaobook”, the \LaTeX document class this thesis is built upon. It is licensed under the LPPL-1.3c and can be found on GitHub (QR code 8).

... to various open source projects, like PlatformIO, fritzing, WebPlotDigitizer, MIDlopsy, and many more.

... to every kind soul on this world.

Bibliography

Here are the references in citation order.

- [1] Nathan O. Sokal. "Class-E RF Power Amplifiers". In: (2001). QR code 9 (cited on pages 7, 15, 17).
- [2] Eirik Taylor. *4.096 MHz Class E SSTC*. QR code 10. Feb. 24, 2022 (cited on page 8).
- [3] Richie Burnett. *HF-SSTC*. QR code 11. Feb. 24, 2022 (cited on pages 8, 17).
- [4] Steve Ward. *Class-E SSTC*. QR code 12. Feb. 24, 2022 (cited on page 8).
- [5] David W Knight. "The self-resonance and self-capacitance of solenoid coils". In: (2016). DOI: 10.13140/RG.2.1.1472.0887 (cited on page 11).
- [6] Gilles Brocard. *Simulation in LTSpice IV*. German. 2013 (cited on pages 13, 14).
- [7] Hawley Hobbies. *How to model a spark gap in LTSpice*. QR code 13. June 10, 2020 (cited on page 14).
- [8] Omnexus. *Water Absorption 24 hours - Absorption Properties of Polymers*. QR code 14. 2018 (cited on page 33).
- [9] *Fachkunde Metall*. German. 57th ed. Europa Lehrmittel, 2014 (cited on pages 33, 36).
- [10] Charles E. Wilkes. *PVC Handbook*. 2005 (cited on page 33).
- [11] Welding Guns of Australia Pty Ltd. *Ceriated Tungsten Electrodes For Welding*. QR code 15. Nov. 19, 2019 (cited on page 38).
- [12] *Ceramic Capacitors Class II*. QR code 16. Jan. 21, 2022 (cited on page 41).
- [13] *The Complete MIDI 1.0 Detailed Specification*. Version 96.1. QR code 17. 2014 (cited on page 57).

Special Terms

CRC Cyclic Redundancy Check. 61, 62

EMI Electromagnetic Interference. 20, 31

FAT File Allocation Table. 62

GDT Gate Drive Transformers. 6

IC Integrated Circuit. 4, 19

ISR Interrupt Service Routine. 66

LCD Liquid Crystal Display. 65, 66

MCU Microcontroller Unit. 49, 55, 65

MIDI Musical Instrument Digital Interface. 49, 51, 52, 57–59

MMC Multi Media Card. 61

MSB Most Significant Bit. 59

NCR Command response time. 61

nibble is a set of four bits. 65

PCB Printed Circuit Board. 20, 31, 32, 39, 41–44

PMMA Polymethylmethacrylate. 33, 35, 38

PTFE Polytetrafluoroethylene. 33

PVC-P plasticized Polyvinyl chloride. 33

PVC-U unplasticized Polyvinyl chloride. 33

SDC Secure Digital Memory Card. 61

SMD Surface Mount Device. 41, 42, 44

SMF Standard MIDI File. 57

SMT Surface Mount Technology. 41

THD Through Hole Device. 41

THT Through Hole Technology. 44

UART Universal Asynchronous Receiver-Transmitter. 65

VCO Voltage Controlled Oscillator. 18, 19, 21, 24, 41

ZCS Zero Current Switching. 7, 21

ZVS Zero Voltage Switching. 7, 21

List of QR codes

As of March 16, 2022, all URLs listed here were still accessible.



BSC12DN20NS3 LTSpice model

https://www.infineon.com/dgdl/Infineon-PowerMOSFET_OptiMOS_PSpice_200V_N-Channel-SimulationModels-v02_00-EN.zip?fileId=5546d4624f72be57014f73c9c774060e



Arc Length in Polar Coordinates

<https://www.wolframalpha.com/widgets/view.jsp?id=c26cbb9457ff75f58f479364ddb79cd1>



tinymid

<https://github.com/marchersimon/tinymid>



How to Use MMC/SDC

http://www.elm-chan.org/docs/mmc/mmc_e.html



Petit FatFs

http://elm-chan.org/fsw/ff/00index_p.html



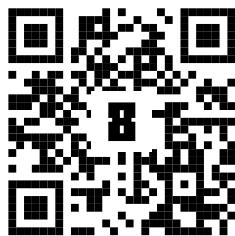
HD44780 LCD driver datasheet

<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>



LCD library

http://www.peterfleury.epizy.com/doxygen/avr-gcc-libraries/group__pfleury__lcd.html



Kaobook Template

<https://github.com/fmarotta/kaobook>



Class-E RF Power Amplifiers

<http://www.classeradio.com/sokal2corrected.pdf>



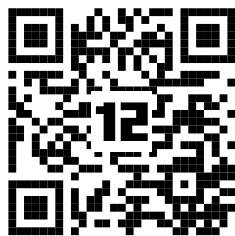
4.096@ifvmode MHz Class E SSTC

<http://uzzors2k.com/index.php?page=4MHzclassE1>



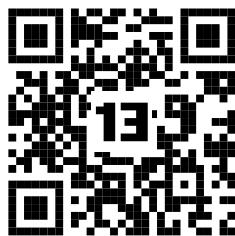
HF-SSTC

<https://www.richieburnett.co.uk/hfsstc.html>



Class-E SSTC

<https://stevehv.4hv.org/classEsstc.htm>



How to model a spark gap in LTSpice

<https://youtu.be/yiGsnCSDGuA>



Water Absorption Properties of Polymers

<https://omnexus.specialchem.com/polymer-properties/properties/water-absorption-24-hours>



Ceriated Tungsten Electrodes for Welding

<https://unimig.com.au/wp-content/uploads/2019/12/Ceriated-Tungsten-MSDS.pdf>



Ceramic Capacitors

<https://epci.eu/ceramic-capacitors-class2>



MIDI Specification

<http://www.freqsound.com/SIRA/MIDISpecification.pdf>