ENEL200 Term 4 Project Report

Variable Vacuum Fluorescent Display Driver Board (VVFD)

Samuel Ramsay - 74900262

Luke Quigley - 54337604

Summary

Vacuum fluorescent displays (VFDs) are old technology, and their driver chips are no longer being manufactured. This means there is a need for a variable sized VFD display driver board. This report outlines the design process of a driver board that can be cascaded to fit any size VFD. The prototype was able to successfully play real time video on a FUTABA VF4001 VFD. The design required modification to meet specifications, and certain parts of the design like current sensing were non-functional. Future iterations of the VFD driver board design are needed to meet all specifications and requirements.

AI Declaration

Grammarly was used for spell checking the report. No generative AI was used for the creation of the report.

Table Of Contents

1. Introduction

With an ever-increasing amount of technology being produced, and older systems going out of production (OOP), specific systems will no longer have replacement parts available. One such technology is vacuum fluorescent displays (VFDs). These displays (and their driver chips) are application specific, thus not intended to work in a generic context. With a recent resurgence in using VFDs for personal hobby projects, a variable driver that can run any VFD is needed.

This report details the design process of a variable VFD display driver that can find the VFD pinout, and also run any input from the user using an Arduino system. A complete schematic, code libraries and matrix finder are available in Appendix A and B.

1.1. Background

Vacuum fluorescent displays (VFDs) were originally developed in 1959 [1], and were the main high-resolution display technology before LCDs were cost effective. Used for their small size, high contrast and variable application, they were often tailor-made to fit a specific device. The most common application was for audio amplifiers and car radios, and as they were often sized to fit into a single DIN slot.

VFDs became popular in the 80s with the rise of analogue electronics, with the primary manufacturer being FUTABA. With FUTABA and other manufacturers no longer making VFDs [2], there are no longer any driver chips to run the displays natively. Engineers are making custom circuits to run their VFDs in a professional environment [3].

VFDs are different from liquid crystal displays (LCDs) in principal operation. LCDs work based on changing a crystal structure to display which pixels are on [4]. Comparatively, VFDs use electric fields to bombard a fluorescent material with electrons to generate light. VFDs need both AC to emit electrons across the filament, and a high voltage (> 15 V) electric field across the pins [5, Sect. 1]. Due to this, LCD driver chips cannot be used to run a VFD without serious modification.

2. Goals

The primary goal is to design a variable vacuum fluorescent display driver (VVFDD), thereby eliminating the need for the original display driver chip. This will be a replacement for the custom display driver chip and should have all the necessary parts to run it. This includes AC, high voltage DC and a fast VFD refresh rate.

The secondary goal is to determine the pinout matrix of a VFD without user input. This will involve finding all valid pin connections and displaying this matrix to the end user. It will require an ammeter to find pin combinations and is meant to speed up the development of any display driver system.

3. Stakeholders

Engineer: The person or team responsible for creating a VFD electronics product using the variable driver board solution. Engineers have the necessary electronics knowledge to be able to modify the design and test it in an ESD-controlled environment. Engineers also possess sufficient C/C++ programming knowledge to modify libraries to support their specific VFD.

VFD product user: The person for who the engineer is designing their product. This person has no electronics or programming knowledge, thus cannot be expected to tune or maintain the driver board.

4. Context

The driver board will be predominantly used in the personal space of the VFD product user as part of a larger system designed by the engineer. In this space, the driver board will be run for long periods of time and may be subject to electrostatic discharge events. Figure 1 shows the dominant interactions between the driver board, product, and stakeholders.

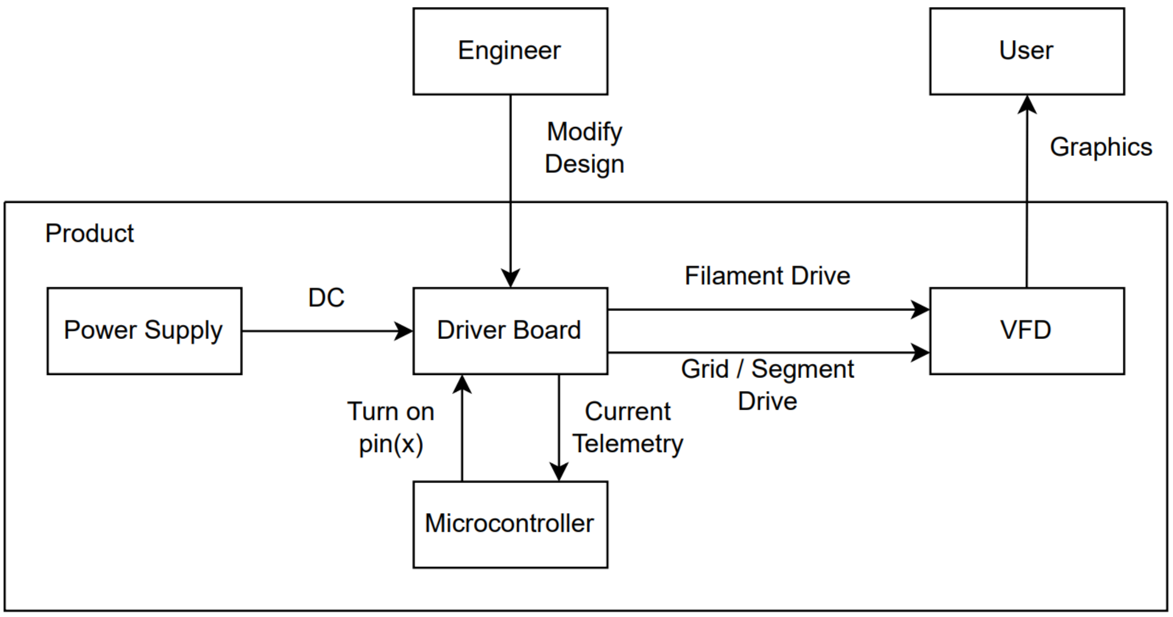


Fig. 1. VFD driver board context diagram.

5. Requirements

This section outlines design requirements to match the needs of the primary stakeholders.

1. The display driver must use onboard power electronics to produce appropriate voltages and waveforms for VFD driving
2. The grid / segment power supply must be adjustable to be able to support different VFD models
3. The display driver must feature a mechanism to expand the number of grid / segment driver pins to support any VFD
4. The design must be able to automatically distinguish grid, segment, and no connect pins
5. The design must be able to find segment pins that are associated with every grid pin
6. The design must use readily available parts
7. The display driver cannot use external logic voltage for power electronics supply

8. The design must feature protection for ESD events on user facing I/O
9. The design must be open-source and well documented so other engineers can modify the codebase to match their needs
10. Software must use Arduino programming language and libraries for microcontroller portability
11. The design must be powered via a USB-C cable for ease of use

6. Specifications

1. The driver board must have an onboard solution to deliver 12 - 30 V DC for VFD grids / segments
2. The grid / segment power supply must be able to source at least 50 mA
3. The driver board must have an onboard solution to deliver a 50 Hz (± 10 %) AC waveform between 6 and 10 V pk – pk for the VFD filament pins
4. The AC power supply must be able to source at least 500 mA
5. The driver board must feature an external connector to deliver power to the onboard power electronics
6. The driver board must use 2.54 mm pitch for external connectors
7. Logic circuits must work with 3.3 V and 5 V to support a variety of microcontrollers
8. The driver board must be capable of refreshing an attached VFD at ≥ 60 Hz

7. Constraints

1. The driver PCB must not exceed 95 x 95 mm
2. The design bill of materials must be less than $70 (NZD)
3. The design must source all parts from Digikey or the ECE parts store

8. Design Description

8.1. Software

There are two custom libraries made to use the VVFDD. The libraries are separated based on use case, and are both required for VVFDD operation.

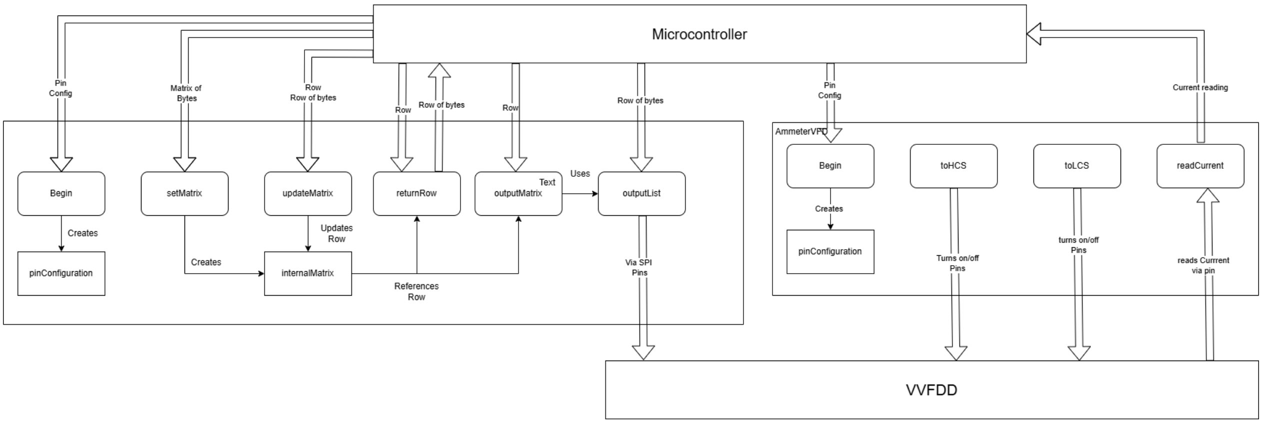


Fig. 2. Software block diagram with relations to components.

8.1.1. ShiftRegVFD

TABLE I
ShiftRegVFD API

Name	Inputs	Returns	Desciption
begin	Pin configuration	None	Initialises the shift register system.
setMatrix	2D byte array	None	Sets the internal matrix of shiftRegVFD
outputList	A single byte row	None	Outputs a row to the shift registers using SPI
outputMatrix	Column number	None	Outputs a row from the internal matrix to the shift registers using SPI
updateMatrx	column number, byte row	None	Updates the given row in the internal matrix
returnRow	Column number	Byte row	Returns the given row

ShiftRegVFD is the primary display driver library for VVFDD operation. The use case is shown in Fig. 2, and the API above. It directly interfaces with the VVFDD by providing data via SPI. It can be used for both standard operation, and finding the matrix pinout locations. Due to ghosting caused by the physical limitation of VFD displays, there is a minimum 175-microsecond uptime (not accounting for computational time) before changing the display's output. Ghosting occurs when the refresh rate is set too high with no dead time, causing unwanted pixels to light up [5, Sect. 4.2].

To output the correct data, it can either output a binary list provided by the user, or output a binary list from an internal matrix, as shown in Appendix A. The internal matrix can be updated with new information if the user needs to change a row, without changing the whole matrix.

8.1.2. AmmeterVFD

AmmeterVFD initialises which current sensing resistor will be used on the VVFDD. Due to the design of the current sensing circuit as shown in Appendix B and Fig. 2, one current sensing resistor must be enabled for VVFDD operation.

The ammeter requires two analogue and two digital pins for operation. The digital pins are used for enabling the ammeters, and the analogue pins for reading the current values. The library will only activate one current-sensing resistor at a time, and the current will be read from the pin that corresponds to the active current sense circuit.

Table II

AmmeterVFD API

Name	Input	Returns	Description
begin	Pin Configuration	None	Initialises the current sensor
toLCS	None	None	Switches on the low current and turns off the high current
toHCS	None	None	Switches on the high current sense and turns off the low current
readCurrent	None	Current reading	Reads the current reading from the ammeter.

8.1.3. Main Software Demonstration

The main.cpp file shows an example of how to run a video on a VFD, using an input–output frame system. Due to the memory limitations of microcontrollers, real-time video is too large to store on a microcontroller [6, Sect. 7]. The software takes in frames externally from the serial port and processes them. It then bit shifts the data into the right position for displaying on the VFD before updating the internal matrix with new information. It then outputs the internal matrix to the VFD, and will loop through which grids are displayed to keep consistent brightness.

As shown in Appendix A, the demonstration uses both libraries and provides an example to users of how the software works for Requirements 9 and 10.

8.2. Hardware

8.2.1. Overview

The block diagram Fig. 3 shows how each functional section of the VFD driver hardware connects. The full schematic for each subsection can be found in Appendix B.

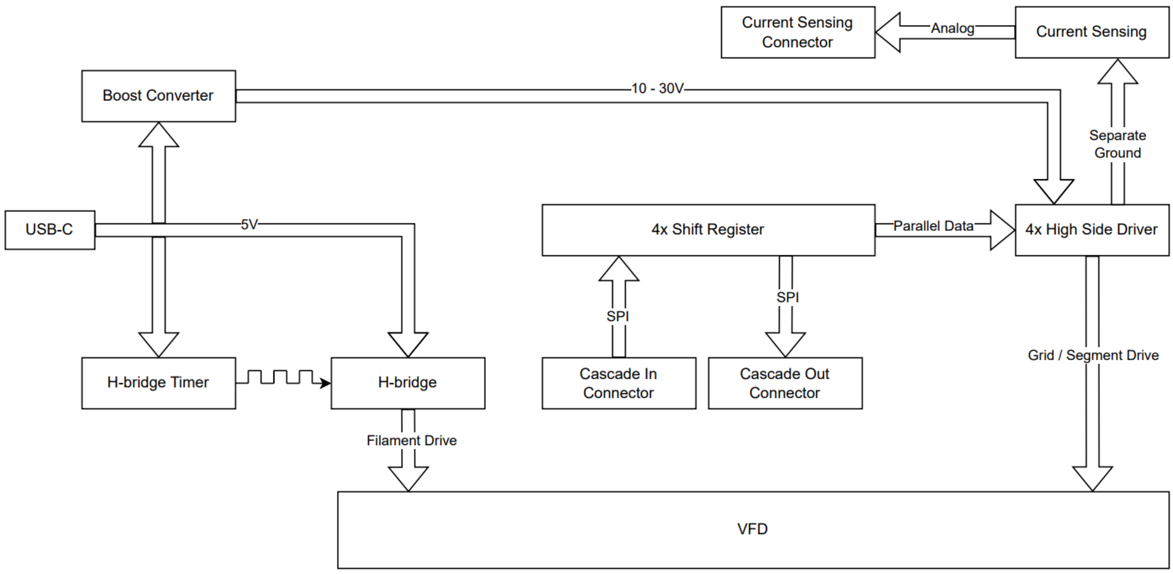


Fig. 3. Hardware block diagram.

8.2.2. Power

To drive a VFD, a high voltage (≥ 12 V) is needed for the grids and segments. Additionally, a low AC voltage (3 - 5 V pk-pk) is required for the VFD filament wire. The filament may consume greater than 200 mA, while a single grid/segment combination only consumes ~1 mA.

8.2.2.1. Boost Converter

To power grids and segments, a boost converter was constructed to convert 5 V from a USB-C receptacle to an adjustable range of 10 - 33 V. This range of output voltages was selected to ensure the design would meet the specifications with some tolerance for imperfect component values. The boost converter was designed to safely work up to 100 mA, allowing for many grids and segments to be active at a time.

The main component of the boost converter is a MC34063 chip which controls the switch timing on the circuit to maintain a feedback voltage of 1.25 V. All component values for the boost converter were selected using equations from the MC34063 datasheet [7, Sect. 9.2.2.2].

8.2.2.2. H-bridge

An H-bridge was used to create an AC voltage for the VFD filament wires. The H-bridge allows the high current filament wire to be controlled by a logic level timer. An AC waveform is needed to eliminate the luminance slant which is present when using a DC filament supply [5, Sect 5.2].

8.2.2.3. H-bridge Timer

To switch the H-bridge, two out-of-phase square waves are needed. A dual 555 timer circuit was used to achieve this; the first phase generates a 50 Hz square wave, and the second phase creates a 10 ms pulse on the falling edge of the first wave. Component values were selected using equations provided in the 555 timer datasheet [8]. Figure 4 shows the output waveform of the timer circuit.

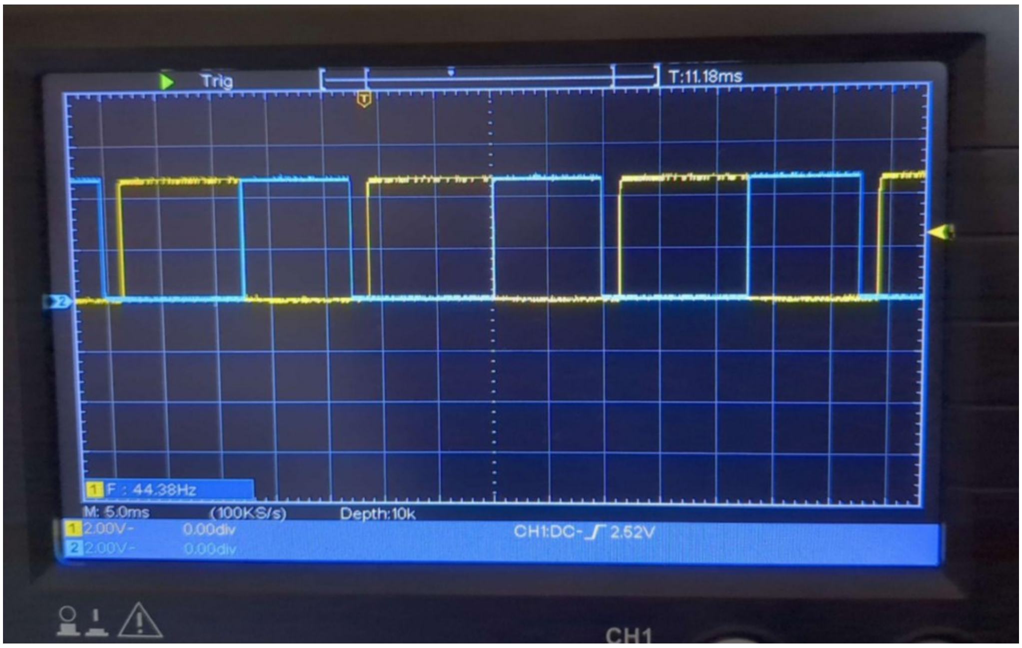


Fig. 4. H-bridge timer producing two out of phase square waves.

8.2.3. Grid and Segment Pin Switching

One driver board has 32 grid/segment output pins that a microcontroller can switch. Most microcontrollers do not have 32 GPIO, so 74HC595 shift registers were used to combine 32 output pins into 3 SPI connections to the microcontroller. However, the 74HC595 shift registers are only intended for 3.3 - 5 V operation [9, Sect. 6.3]. To switch ~20 V VFD grid/segments, TBD62783 high-side driver arrays were utilised.

8.2.4. Current Sensing

The design needs to be able to identify grid, segment, and no-connect pins. This can be achieved by monitoring the current draw from the boost converter, as grids and segments draw different currents. To sense small current changes with a microcontroller, opamps were used to amplify a voltage drop across a resistor. The voltage drop is generated from current flowing from the high-side driver ground through a resistor to the main ground. Two different current sense resistors can be activated with n-channel MOSFETs, which allows for a high-precision and low-precision current sensing circuit. Multiple stages of opamps were used to create separate outputs for 3.3 V and 5 V logic level MCUs with full dynamic range.

8.2.5. Additional Features

The design features basic ESD protection in the form of a TVS diode on the USB-C power input. The TVS diode clamps to ~9 V [10] to protect downstream components in the event of an ESD-induced transient.

The shift registers share a common reset line; this line is connected to an RC circuit such that the reset line is at GND at startup, then goes to VCC after a short time has elapsed. This resets all shift registers when the power is connected, ensuring all grid/segment pins are initialised to an off state.

The PCB was designed to allow the VFD driver section to operate on power input from another board. This allows one master board populated with power electronics to supply power to all “slave” or “cascade” connected boards. In this arrangement, all shift registers would be connected to the same SPI bus; thus, the whole system operates as one unit. Each cascaded board would add 32 grid/segment driver pins to the system.

9. Design Rationale

9.1. Software

9.1.1. ShiftRegVFD

As per requirement 9, the Code must be easily accessible and understandable to a non-technical audience. To achieve this, shiftRegVFD is a single library with all functions handled internally like a standard Arduino library.

The reason for storing the matrix in a binary list format is due to memory limitations on microcontrollers. Initially it was a 2D array matrix; however, every 0 and 1 is one byte. Compared to a binary 1D array, the total memory usage in O(n) is:

2D Array

$$Pincount\ \times 8\ \times Gridcount\ =\ 64\ bits\ per\ grid$$

Binary array

$$\frac{Pincount}{8}\ \times Gridcount\ =\ 64\ bits\ per\ grid$$

Conducting tests using the VF4001, there was a decrease in memory usage of 89% per grid. This will allow the design to be more efficient and store larger VFDs on the same microcontroller.

SPI was chosen as the communication protocol with the shift registers due to baud rate compatibility. Since the Arduino library has serial as blocking, the baud rate can be changed to allow syncing without data loss. It also allows for a faster output compared to the ShiftOut command [11], which can work on any pin. SPI can operate at the maximum frequency of the microcontroller, ensuring minimal downtime for the microcontroller to receive information.

9.1.2. Choosing Communication Baud Rate

The main limitation of the VFD refresh rate is the microcontroller that is used. The 74HC595 has a maximum clock frequency of 25MHz [9, Sect. 6.6], and minimum of 4.2MHz when undervolted to 2V. Due to this, the main limitation is the microcontroller's clock speed, which restricts output information to the microcontroller's maximum frequency. To improve the efficiency of the VVFD, a maximum SPI frequency of 14MHz was chosen. Testing in section 10.1.2 shows why. As a standard Arduino Uno runs normally at 8MHz, but can be overclocked to 16MHz [6, Sect. 29], and to give headroom in case of manufacturing defects. Since the Arduino SPI library always selects the microcontroller's frequency if the stated speed exceeds it, this ensures that the output only blocks the input from receiving new data for the minimum amount of time [14].

9.1.3. Serial In

Serial In was chosen as the communication for sending frames to the microcontroller. This was chosen because an external device can send data to the microcontroller via a USB Cable, allowing for fast testing to verify if the VFD display was properly synced to an external source, such as audio.

9.2. Hardware

9.2.1. Boost Converter

To produce the 10 - 30 V required for the VFD grid and segment pins, a boost converter was used. A flyback converter was also considered but rejected due to increased complexity and cost. Alternatively, an external power supply could have been used, but the boost converter was favoured so the driver solution could be fully self-contained.

The MC34063 chip was used as the central component of the boost converter. This chip was selected because it does not require an external MOSFET for switching. Additionally, the MC34063 is produced by multiple manufacturers, making it well-documented and cheap.

9.2.2. H-bridge and Timer

Initially, an MCU PWM timer was used over a dedicated hardware timer to drive the filament h-bridge. This solution had the advantage of reducing part count but was rejected due to being MCU-dependent. A software timer solution could still be implemented, but would need to be run by the CPU and could not be offloaded to timer peripherals. This arrangement is more computationally demanding, making it less ideal for quick display updates. Ultimately, 555 timer chips were selected to offload h-bridge timing from the MCU, and because of their low cost.

An H-bridge was chosen to handle the high current draw of the VFD filament wire. Driving the filament wire directly from a timer was considered, but was rejected due to the filament wires high current draw.

An H-bridge is typically constructed with four n-channel MOSFETs; these designs rely on switching a high voltage load to keep VGS high. However, a VFD filament has a similar voltage amplitude to the MCU logic level; this would make VGS = 0 for high-side n-channel MOSFETs. To fix this issue, high side p-channel MOSFETs were selected over high side n-channel MOSFETs.

9.2.3. Grid and Segment Pin Switching

A P-channel MOSFET array was chosen to switch the high-voltage grid/segment pins of a VFD. Low-side n-channel MOSFET switching arrays and individual p-channel MOSFETS were also considered. Low side switching was rejected because it requires pull-up resistors, resulting in higher current draw and reduced power efficiency. Individual p-channel MOSFETs were rejected because they required more board space. Ultimately, a high-side driver IC was selected for high power efficiency and compact size.

To expand the MCU I/O shift registers were used; alternatives such as demultiplexing and pin matrices were also considered. Shift registers were selected because more registers can be connected without increasing the pin count, whereas demultiplexing or pin matrices would use more pins. This simplified trace routing and made the master/slave expandable board arrangement possible.

Powering VFD grid and segment pins directly with the shift registers was not possible because the 74HC595 only operates between 2 V and 6 V [9, Sect. 6.5]. Some shift registers feature high-voltage outputs suitable for driving a VFD, but these were not considered due to increased cost.

9.2.4. Current Sensing

For the current sensing circuit, the TLV274IN chip was used. This chip was used for its rail-to-rail output (important for maximising ADC dynamic range) and four onboard opamps. Although cheaper options with similar specs are available on digikey, the TLV274IN was selected as the best value option from the UC ECE component shop.

10. Evaluation

10.1. Testing

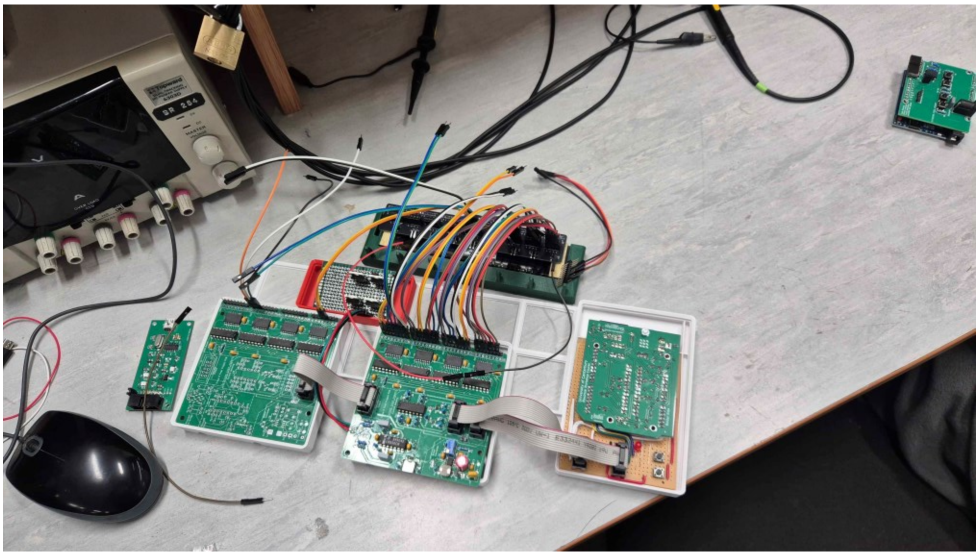


Fig. 5. The testing setup used.

10.1.1. Software

Software was tested using a series of milestones in the order below, with the hardware setup shown in Fig. 5:

1. Output a static image
2. Output a blinking image
3. Output two separate multiplexed grids
4. Output a non-synced video
5. Output synced video

These milestones were chosen to slowly build up to sending real-time video over the serial port, and to check that all outputs were accurate to the input. All data can be found in Appendix E.

The first milestone was achieved by directly outputting the matrix to the shift registers. It was achieved in roughly one hour of work. The second milestone was achieved after 2-3 hours of work, and showed multiple issues with the code. The main problem was how the outputList function outputted the matrix, and it was reworked because of it.

The third milestone was achieved in approximately an hour due to the reworked system; however, updating the internal matrix correctly required a significant amount of time. The code was revised again to ensure the next milestone could be achieved successfully.

Milestones four and five took 30 hours to complete. The extra time was due to having to debug both the matrix updater system and the baud rate blocking issue. Once both bugs were resolved, it took another five hours to fully sync the video as there were multiple bugs on the serial-in side.

10.1.2. VFD Refresh Rate

Two main concerns were found when refreshing the VFD. The first one is known as ghosting. The second is that due to the Arduino library, both serial and SPI are blocking [11][14]. If they are in phase, the display will not update correctly.

To test the refresh rate of the VFD and interference from blocking input/output commands, various delays were tested, as the delay time needs to be out of phase with the input so that no blocking can occur. To find the optimal delay time to minimise flickering, the computation time also had to be accounted for.

The delay time was determined through trial and error, as slight variations in the microcontroller's clock speed can impact the flickering. Starting with a 2ms delay time, and working down in milliseconds until the ghosting was noticeable. Once a low number in microseconds was found with noticeable ghosting, the time was then offset to prevent it from fitting perfectly into a second of video, thereby decreasing the amount of ghosting. The time found with the optimal delay and minimal ghosting was 175 microseconds.

10.1.3. Boost Converter

The boost converter was tested to ensure it meets the specification of having an output range of 12 - 30 V. By adjusting the feedback potentiometer, the boost converter on the prototype was found to have a minimum voltage of 11.59 V and a maximum of 32.94 V. This range lies within the specified values.

10.1.4. H-bridge and Timer

The h-bridge and timer circuit were prototyped on a breadboard prior to ordering a prototype PCB. This breadboard circuit worked and was implemented on the final design. However, when testing the circuit on the PCB, the H-bridge MOSFETs became hot. This occurred because the breadboard model utilised larger power MOSFETs, which could handle a momentary short circuit due to insufficient timer dead-time. The small SMD MOSFETs could not sustain the short circuit and failed, so large external MOSFETs had to be attached to the PCB to temporarily fix the problem. For future iterations, a new timer design has been devised with three stages. The new stage adds a short delay (dead-time) in between the out-of-phase square waves to protect the MOSFETs.

After implementing the temporary h-bridge fix, the output waveform was measured (Fig. 6).
This waveform lies within the range defined in the project specifications, so it is suitable for use with a VFD.

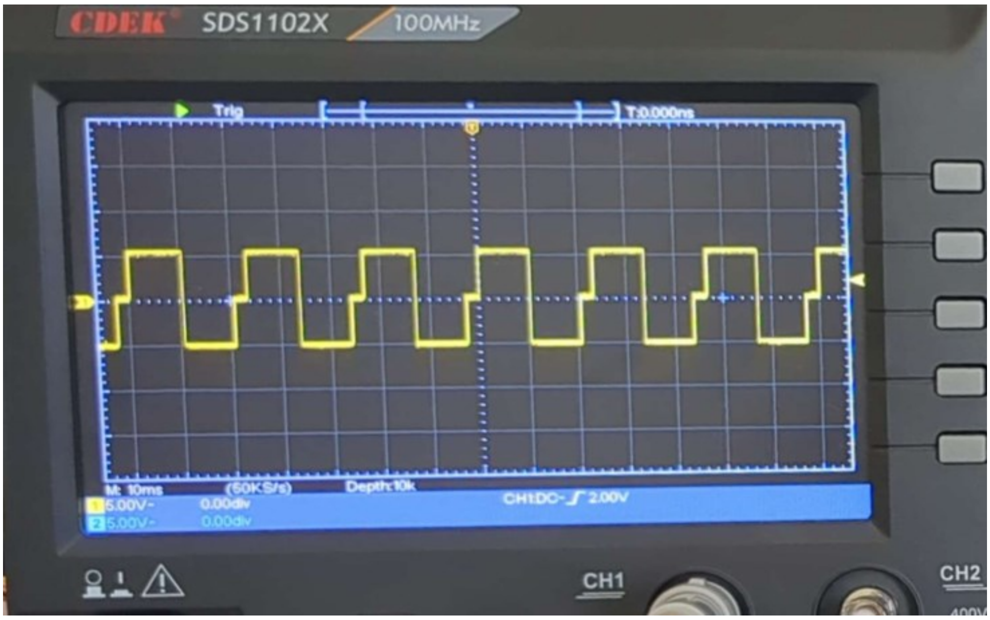


Fig. 6. H-bridge AC waveform.

Additionally, the h-bridge needs to be able to source at least 500 mA. The maximum current can be inferred from the MOSFET datasheets [12] [13]. Both MOSFETs have a continuous drain current higher than 3 A, so can easily source 500 mA continuously.

10.1.5. Grid and Segment Pin Switching

To test if the grid and segment pins could be switched on or off, a simple test program was written. The test program sends bytes over SPI to be latched into the shift registers. To simplify testing, a simple function and macro can be used: *switch_pins(PIN(1)).* Switching high-side driver pins with this function worked as expected, as a high-side driver only outputted the boost converter voltage when it was purposely switched on.

10.1.6. Current Sensing

Current sensing was tested by turning on different combinations of VFD pins connected to the device and observing the ADC value read from the current sense opamps. Any combination of active pins immediately maxed out the range on the high-precision current-sensing circuit. High-side driver leakage current is much higher than individual grid/segment pin current draw; this was overlooked during the development of the current sensing circuit.

This issue could not be resolved for this design iteration but could be resolved in future iterations by using individual p-channel MOSFETs. Individual p-channel MOSFETs could have pull-up or pull-down resistors connected in such a way that they wouldn't draw current visible to the current sensing circuit.

10.2. Conclusion

Table III, Table IV, and Table V summarise how the design meets requirements, specifications and constraints respectively. All requirements, constraints and specifications have been met except for current sensing and pin finding requirements. The design can drive a VFD from any manufacturer using a simple, expandable hardware design and modular codebase. This was the main goal of the project; therefore, the project can be deemed successful.

TABLE III

REQUIREMENTS

Requirement No.	Requirement	Requirement met?	How was the requirement met?
1	Onboard power electronics	✓	Onboard boost converter and h-bridge
2	Adjustable grid / segment voltage	✓	Boost converter has adjustable output voltage

3	Expandable grid / segment pins	✓	Multiple boards can be chained together
4	Design can find VFD pin types	✕	N/A
5	Design can find associated grid and segment pins	✕	N/A
6	Design uses readily available parts	✓	Use common components like NE555, 74HC595, MC34063
7	Design doesn't use low current capability logic voltage for power electronics	✓	Power electronics are supplied by USB-C, which can source 3 A.
8	ESD protection	✓	USB-C 5 V protected by TVS diode
9	Open Source	✓	Repository is public on GitHub, schematics and code feature appropriate comments
10	Software uses Arduino language and libraries	✓	Software Arduino compatible
11	Design must use USB-C for power	✓	Used USB-C connector

TABLE IV

SPECIFICATIONS

Specification No.	Specification	Specification met?	How was the specification met?
1	Design needs 12 - 30 V DC power supply for grids / segments	✓	Use boost converter with 11.59 - 32.94 V output
2	Grid / segment power supply must source > 50 mA	✓	Boost converter can source ~100 mA
3	Design needs 50 Hz (± 10 %), 6 - 10 V pk – pk power supply	✓	H-bridge does 45 Hz, 10V pk-pk
4	AC power supply can source 500 mA	✓	H-bridge mosfets have low R_{DS}(on) and continuous drain current >> 500 mA
5	The driver board must use an external power connector	✓	Use USB-C for power electronics supply

6	The driver board needs 2.54 mm pitch connectors	✓	Use 2.54 mm pitch IDC and pin headers
7	Design can use 5 V or 3.3 V logic	✓	Shift registers tolerate both input voltages
8	The driver board must be able to refresh a VFD at ≥ 60 Hz	✓	Shift registers tested with 8 MHz clock for fast refreshing

TABLE V
CONSTRAINTS

Constraint No.	Constraint	Constraint met?	How was the constraint met?
1	PCB must not exceed 95 x 95 mm	✓	PCB is exactly 95 x 95 mm
2	Bill of materials must be less than $70 (NZD)	✓	One driver board cost $48 (NZD)
3	Components must be sourced from Digikey or the ECE parts store	✓	Used Digikey and ECE parts store exclusively

References

[1] W. Ribbe, "DM 160, Tube DM160; Röhre DM 160 ID19445, INDICATOR, in gene | Radiomuseum.org."
https://www.radiomuseum.org/tubes/tube_dm160.html (Accessed 14th October 2025)

[2] Futaba Corporation "Notice of Withdrawal from Vacuum Fluorescent Display and Vacuum Fluorescent Display Module Business", August 5th 2020
https://www.futaba.com/wp-content/uploads/2022/01/notice_20200805.pdf (Accessed 14th October 2025)

[3] Voltage Ventures, "Vacuum fluorescent display," *Instructables*, Jan. 17, 2024.
https://www.instructables.com/Vacuum-Fluorescent-Display-Arduino/ (Accessed 14th October 2025)

[4] Jeff Tyson "How LCDs Work", 17 July 2000. HowStuffWorks.com.
https://electronics.howstuffworks.com/lcd.htm (Accessed 14th October 2025)

[5] Noritake U.S.A "A Guide to Fundamental VFD Operation", 28th July 2001
https://www.noritake-elec.com/technology/general-technical-information/vfd-operation (Accessed 14th October 2025)

[6] Atmel, "ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET," 2015.
https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (Accessed 14th October 2025)

[7] Texas Instruements, "MC3x063A 1.5-A Peak Boost/Buck/Inverting Switching Regulators", Feb 2015, ti.com, sec. 9.2.2.2.
https://www.ti.com/lit/gpn/mc34063a (Accessed 9th October 2025).

[8] Texas Instruments, “xx555 Precision Timers”, Feb 2025, ti.com
https://www.ti.com/lit/ds/symlink/ne555.pdf (Accessed 12th October 2025).

[9] Texas Instruments, “SNx4HC595 8-Bit Shift Registers With 3-State Output Registers”, Feb 2025, ti.com
https://www.ti.com/lit/gpn/SN54HC595-SP (Accessed 12th October 2025).

[10] Diotec Semiconductor, “SMF5.0A”, August 2025, diotec.com
https://diotec.com/request/datasheet/smf50a.pdf (Accessed 12th October 2025).

[11] Karl Söderby “Arduino API”, February 2024, docs.arduino.cc
https://docs.arduino.cc/learn/programming/reference/ (Accessed 14th October 2025)

[12] Diodes Incorporated, “DMG2305UX P-CHANNEL ENHANCEMENT MODE MOSFET”, July 2021, diode.com
https://www.diodes.com/assets/Datasheets/DMG2305UX.pdf (Accessed 12th October 2025).

[13] Slkor, “SL3402 N-Channel Power MOSFET”, 2024, slkoric.com
https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/7181/SL3402%20SOT-23.PDF (Accessed 12th October 2025).

[14] Arduino “SPI”, may 2024, docs.arduino.cc
https://docs.arduino.cc/language-reference/en/functions/communication/SPI/ (Accessed 14th October 2025)

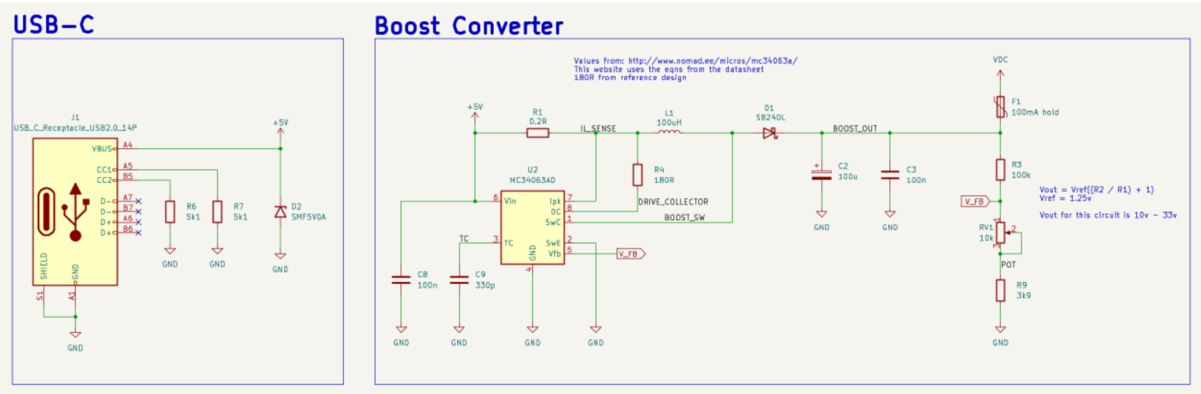
Appendix

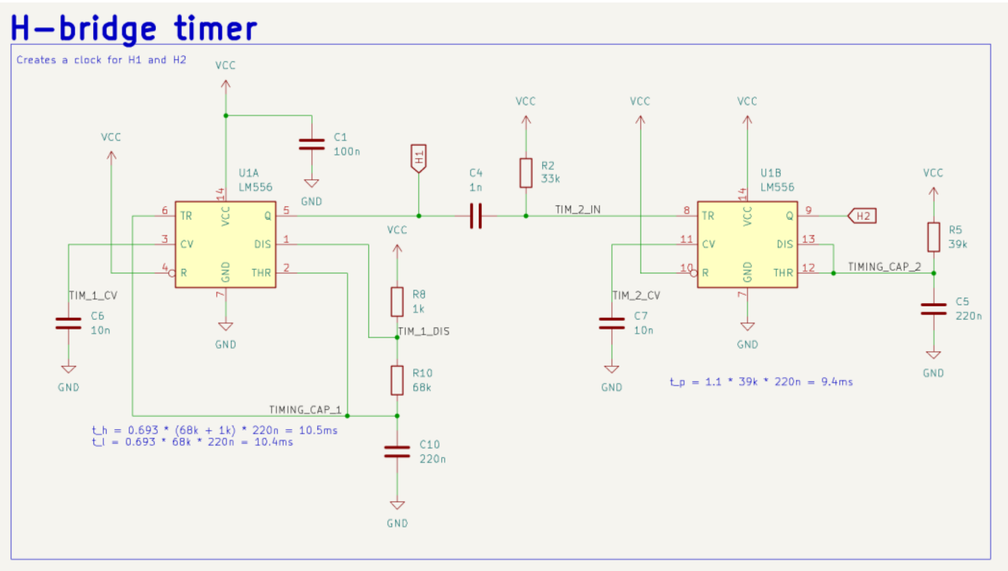
A: GitHub

The GitHub repository contains updated designs that solve issues encountered during the design process.

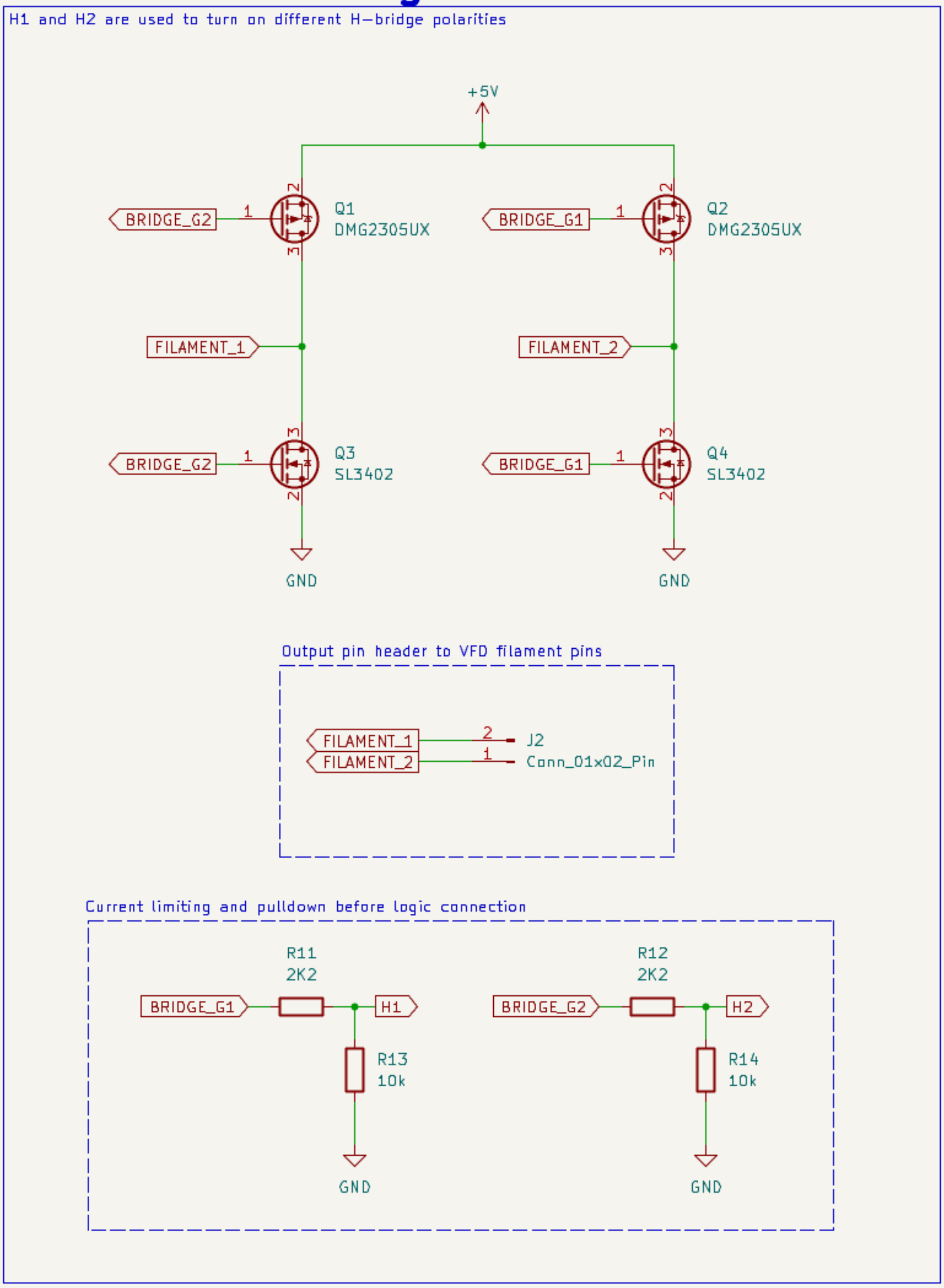
github.com/For-the-republic/VVFD (Accessed 15th October 2025)

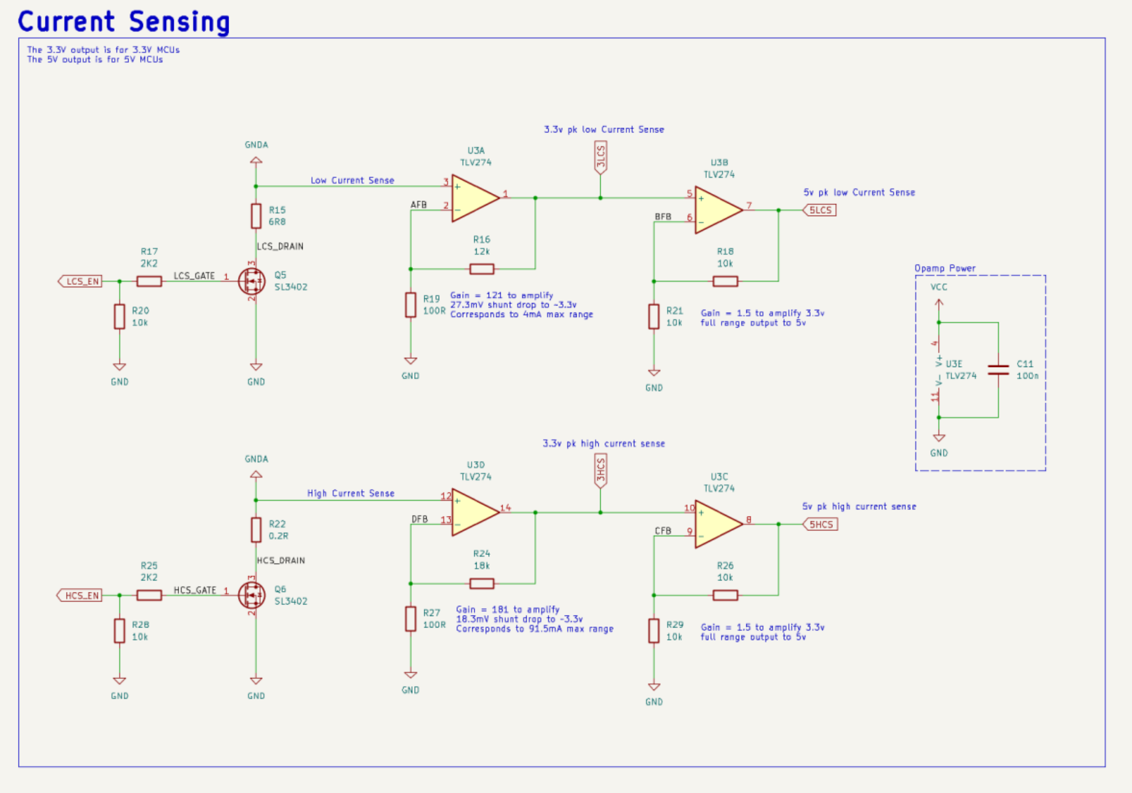
B: Schematics

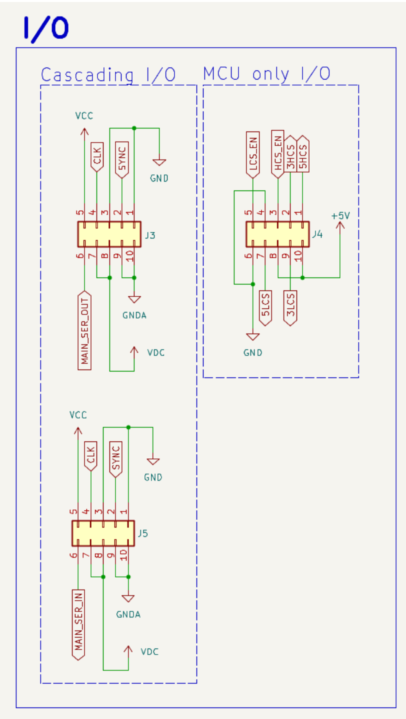




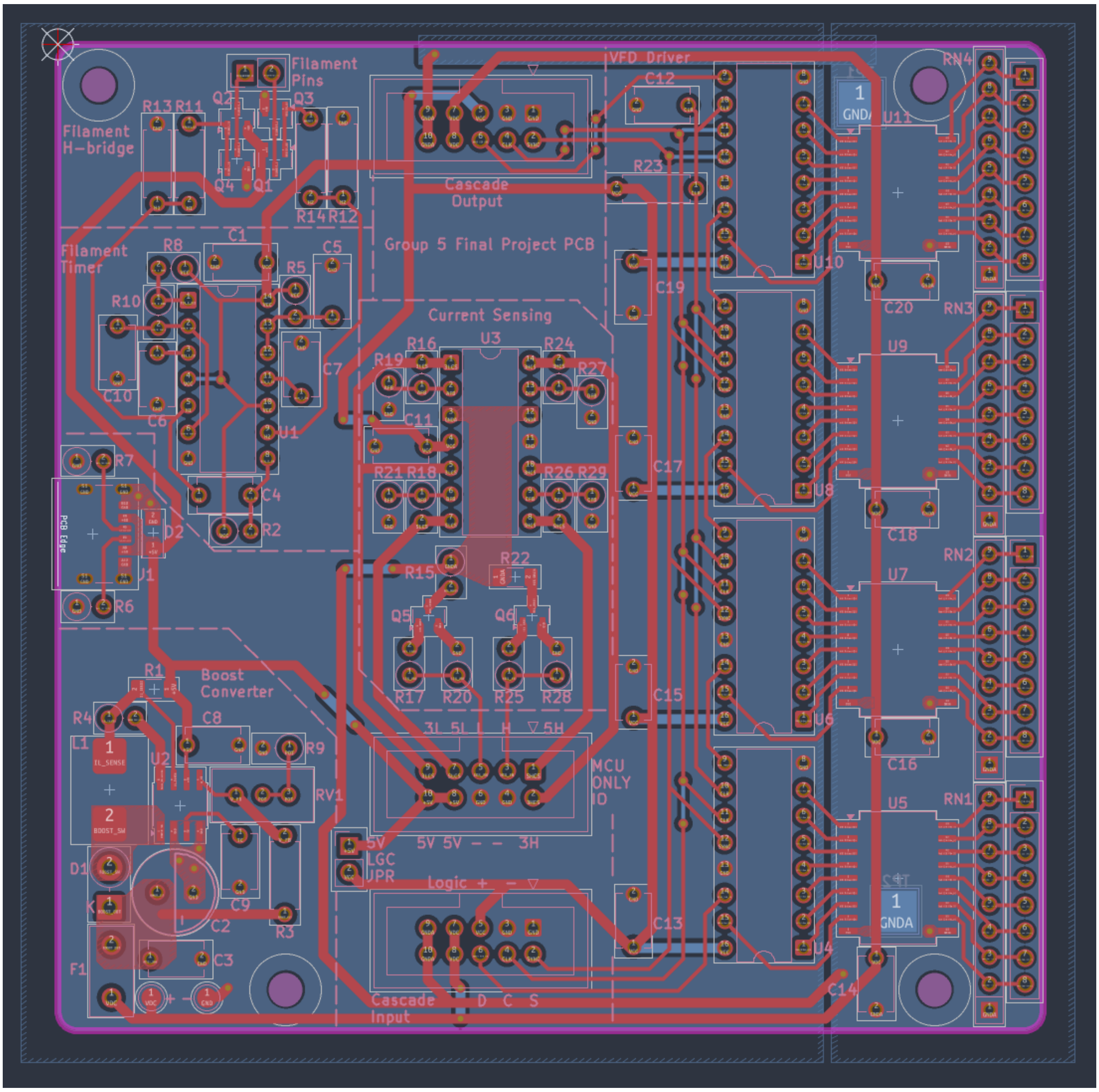
Filament H-bridge




Current Sensing
The 3.3V output is for 3.3V MCUs
The 5V output is for 5V MCUs
3.3v pk low Current Sense
Low Current Sense
5v pk low Current Sense
Gain = 121 to amplify
27.3mV shunt drop to ~3.3v
Corresponds to 4mA max range
Gain = 1.5 to amplify 3.3v
full range output to 5v
Opamp Power
3.3v pk high current sense
High Current Sense
5v pk high current sense
Gain = 181 to amplify
18.3mV shunt drop to ~3.3v
Corresponds to 91.5mA max range
Gain = 1.5 to amplify 3.3v
full range output to 5v


I/O
Cascading I/O
MCU only I/O

C: PCB Design as Manufactured



D: PCB Bill of Materials

Reference	Qty	Value	Digikey Part No	Manufacturer Part No	Unit Price ($)
C1,C3,C8,C11,C12,C13,C14,C15,C16,C17,C18,C19,C20	13	100n	UC		0.2
C2	1	100u	732-9051-1-ND	860080674009	0.7245
C4	1	1n	UC		0.2
C5,C10	2	220n	UC		0.4
C6,C7	2	10n	UC		0.2
C9	1	330p	UC		0.2
D1	1	SB240L	UC	SB240L	0.8
D2	1	SMF5V0A	4878-SMF5.0ACT-ND	SMF5.0A	0.5175
F1	1	100mA hold	UC	TRF250-120	1.2
J1	1	USB_C_Receptacl	2223-UJC-H-G-SMT-P6-TRCT-ND	UJC-H-G-SMT-P6-TR	0.851
J2	1	Conn_01x02_Pin	UC		0
J3,J4,J5	3	Conn_02x05_Coun	UC		0.45
J6,J7,J8,J9	4	Conn_01x08_Pin	UC		0.15
JP1	1	Jumper_2_Open	UC		0
L1	1	100uH	445-180962-1-ND	SPM7054VT-101M-D	2.6105
Q1,Q2	2	DMG2305UX	DMG2305UX-7DICT-ND	DMG2305UX-7	0.3565
Q3,Q4,Q5,Q6	4	SL3402	5399-SL3402CT-ND	SL3402	0.3335
R1,R22	2	0.2R	283-MFHA1206R2000FCCT-ND	MFHA1206R2000FC	0.506
R2	1	33k	UC		0
R3	1	100k	UC		0
R4	1	180R	UC		0
R5	1	39k	UC		0
R6,R7	2	5k1	UC		0
R8	1	1k	UC		0
R9	1	3k9	UC		0
R10	1	68k	UC		0
R11,R12	2	2K2	UC		0
R13,R14,R23	3	10k	UC		0
R15	1	6R8	UC		0
R16	1	12k	UC		0
R17,R25	2	2K2	UC		0
R18,R20,R21,R26,R28,R29	6	10k	UC		0
R19,R27	2	100R	UC		0
R24	1	18k	UC		0
RN1,RN2,RN3,RN4	4	1M	4609X-101-474LF-ND	4609X-101-474LF	0.9085
RV1	1	10k	UC	3296W	3
U1	1	LM556	UC	LM557	1
U2	1	MC34063AD	5399-MC34063SCT-ND	MC34063S	0.5405
U3	1	TLV274	UC	TLV274	3.15
U4,U6,U8,U10	4	74HC595	UC	74HC595	1.2
U5,U7,U9,U11	4	TBD62783A	TBD62783AFGELCT-ND	TBD62783AFG,EL	4.876

E: Software testing photos