Power Electronics Final Project Report

Eric Zou

December 2025

1 Project Overview

The goal of this project was to create several circuit modules that would ultimately enable driving a stepper motor at audible frequencies to play music. To this end, there were two main modules:

1. Finite state machine and H-bridge for driving the stepper in full-step drive.
2. Boost converter with feed-forward and self-start to power the motor and control circuitry from battery-level voltage.

2 Finite State Machine (FSM) Motor Driver

2.1 Design

The goal for driving the stepper motor was to perform full-step driving, during which each of the two independent coils in the stepper motor are always energized, but with different timing. In particular, every half-period, the voltage applied across one coil must flip polarity to ensure that the stepper continuously spins in one direction. In full-step drive, the two coils are operated in this way but 90 degrees out of phase, as depicted in Fig. 1. It is apparent from this timing diagram that there are four distinct states corresponding to the polarities across each coil. Therefore, the motor driver should operate as a four-state FSM.

In practice, the FSM can be implemented by clocking a 74LS163 counter chip and using the bits Q0 and Q1 as outputs, which will allow the counter to go from 0b00 to 0b11 before resetting. To reset the counter, we can detect when Q0 and Q1 are both logic high and send that signal to the reset pin, which is easily achieved with a NAND gate.

We realize that to get the two control signals we need for the two coils, we can simply use Q1 and (Q0 XOR Q1) as the two signals. Further, it can be shown that the stepping frequency of the stepper motor will also be equal to the clock frequency, which is convenient for producing particular musical notes with the motor. A schematic of the counter circuitry is shown in Fig. 2.

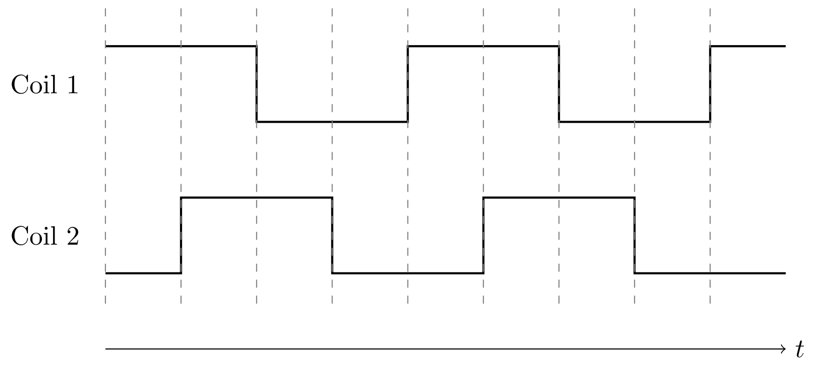
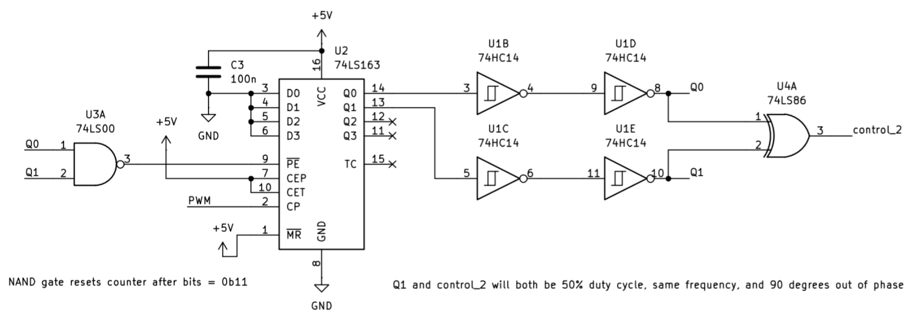


Figure 1: Illustration of the 4 states in the FSM.



NAND gate resets counter after bits = 0b11

Q1 and control_2 will both be 50% duty cycle, same frequency, and 90 degrees out of phase

Figure 2: Finite state machine counter.

The current control signals are only enough to turn on one polarity of voltage for each coil in an H-bridge. To be able to turn on voltage in the other direction, we also need the inverses of both signals. To prevent shoot-through in the H-bridges, delay circuits (Fig. 3) are constructed to provide approximately 1 µs of delay between turning off one direction and turning on the other. It should be noted that since the audible frequencies have periods much longer than 1 µs, this delay has negligible effect on the stepper motion or the sound produced.

To supply the clock signal to the 74LS163 (the PWM net in Fig. 2), any TTL square wave could be used, including that from an Arduino for pre-programmed songs. For the purposes of the project, a Schmitt trigger oscillator was made, with buttons connecting a 150 nF capacitor to different resistors to produce oscillations at frequencies corresponding to different musical notes.

2.2 Results

Fig. 4 shows the control signals for the gate drivers of the two H-bridges. As we can see, the finite state machine circuit is able to create four control waveforms for turning on each "diagonal" on the two H-bridges.

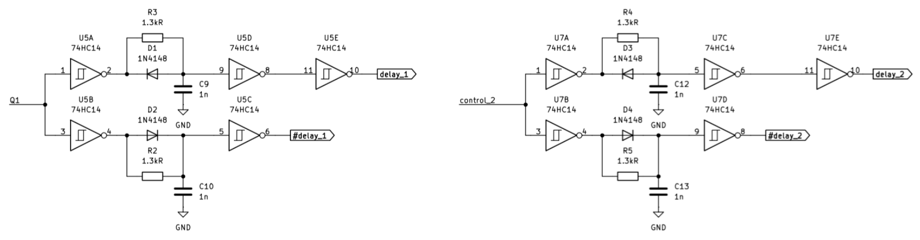


Figure 3: Delay circuit.

Fig. 5 depicts the voltage at two opposite nodes at the output of the H-bridge (yellow and green), as well as the voltage across the H-bridge (pink) when the input voltage to the H-bridge is 12 volts. Meanwhile, while operating a motor on the output, the waveforms are somewhat less clean due to motor back EMF but overall still of good quality, as shown in Fig. 6. Thus, we conclude that the FSM motor driver is a success.

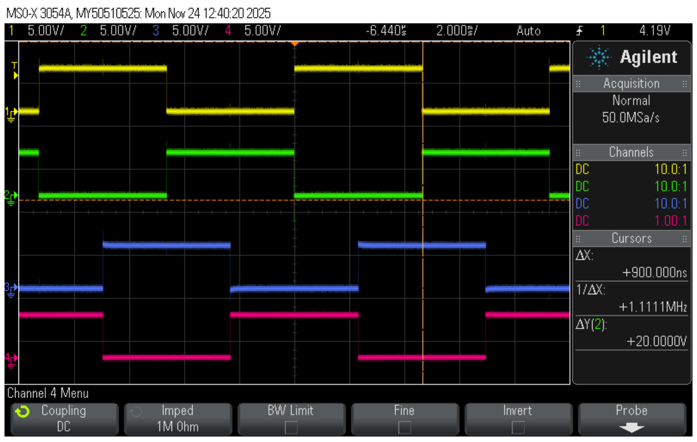
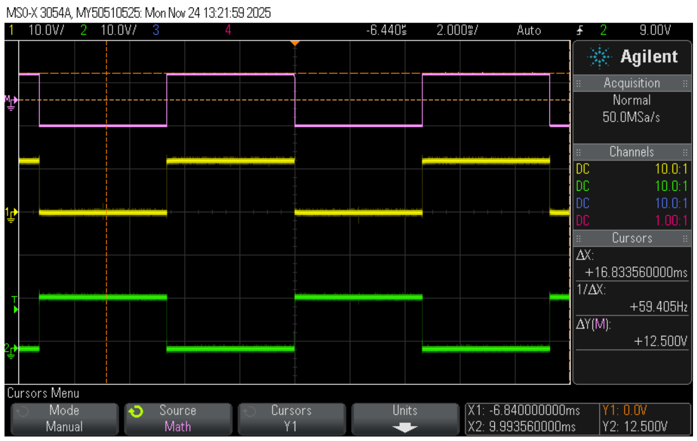


Figure 4: Control signals for H-bridges.

3 Feed-forward Boost Converter

As an additional design challenge, it was decided that the whole system should run off of battery-level voltage (6 volts). To produce the higher voltage that is required to power the stepper, a boost converter was designed to step 6 volts up to 15 volts.


MSO-X 3054A, MY50510525: Mon Nov 24 13:21:59 2025

1 10.0V/ 2 10.0V/ 3 4 -6.440s 2.000s/ Auto 2 9.00V

Acquisition Normal 50.0MSa/s

Channels DC 10.0:1 DC 10.0:1 DC 10.0:1 DC 1.00:1

Cursors ΔX: +16.833560000ms 1/ΔX: +59.405Hz ΔY(M): +12.500V

Cursors Menu

Mode Manual | Source Math | Cursors Y1 | Units

X1: -6.840000000ms X2: 9.993560000ms Y1: 0.0V Y2: 12.500V

Figure 5: Voltage across one H-bridge.

3.1 Design

Some design specifications for operation of the boost converter were laid out, including:

1. Switching frequency of 30 kHz.
2. Output voltage at 15 volts for any input voltage < 15 volts.
3. Inductor operating in continuous conduction mode (CCM) for output current as low as 0.4 amps.
4. Output voltage ripple no more than 10% (1.5 volts).

Ideally, a feedback loop would be designed to ensure that the output voltage is 15 volts regardless of the input voltage or loading conditions. However, it was determined that implementing current-mode boost converter control would be too complex to create in the allotted time, so instead, a feed-forward system was designed to compute the duty cycle required for the boost converter to step up any input voltage to 15 volts.

To do this, we start with the boost converter gain equation (assuming CCM), which is

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1}{1-D}.$$

Rearranging, we find that $D = 1 - \frac{v_{\text{in}}}{v_{\text{out}}} = 1 - \frac{v_{\text{in}}}{15}$. This means that we could find D as a voltage using some straightforward analog arithmetic: a voltage divider can perform the division, and an op amp subtractor can take the difference from 1 volt. However, since $0 < D < 1$, the op amp would be operating close to the supply rail, which may cause errors in the computation. So instead, we can scale everything by a factor of 5 to avoid being near the rail. This gives

$$5D = 5 - \frac{v_{\text{in}}}{3}.$$

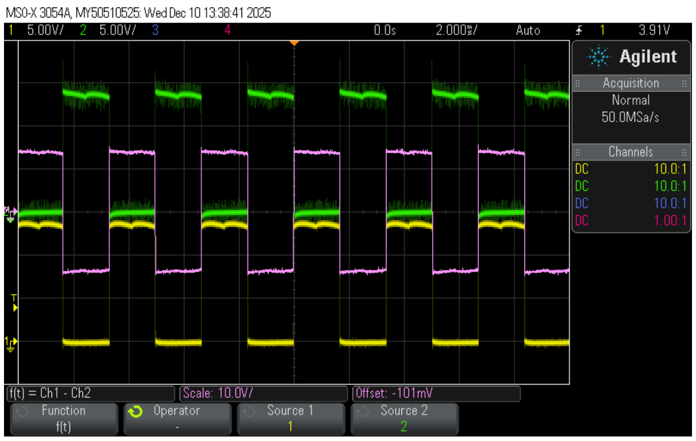


Figure 6: Voltage across H-bridge with a motor running.

The circuit that performs this computation is shown in Fig. 7.

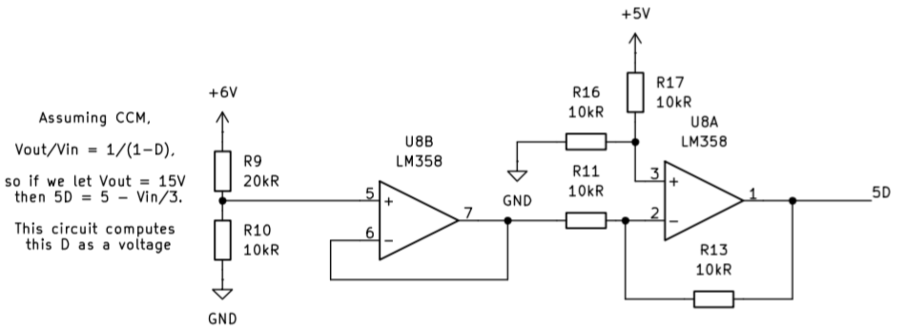


Figure 7: Duty cycle calculation.

To convert the voltage $5D$ into some kind of duty cycle D, we can use it as a fixed reference and compare it to some triangle wave. If the reference is a fraction D of the way up the triangle, then using the fixed reference and the triangle wave as the noninverting and inverting inputs of a comparator would produce a square wave with duty cycle D. To create an approximate triangle wave, we can use a 555 timer as shown in Fig. 8. The 555 timer oscillates between voltages of $V_{CC}/3$ and $2V_{CC}/3$. If we supply $V_{CC} = 5$ volts, then we need our reference to be a fraction D of the way between 1.67 volts and 3.33 volts. This would be $1.67 + 5D/3$ volts. To compute this value, the buffers and summer depicted in Fig. 9 are used. Finally, this computed reference and the trinagle wave from the 555 timer are compared using a comparator to create the control signal for the boost converter, as shown in Fig. 10.

Assuming that the input voltage will always be around 6 volts, then the

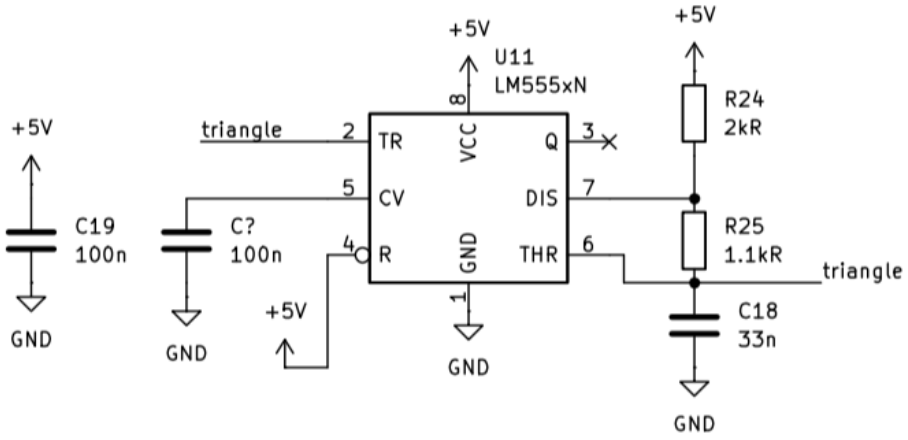


Figure 8: 555 timer as triangle wave generator.

condition for maintaining CCM in the inductor is

$$L > \frac{V_{\text{in}}(1-D)DT}{2I_{\text{load}}}$$

which is around 60 µH. For additional insurance of CCM, I chose $L = 90$ µH to account for varying duty cycle that may occur due to some variance in the input voltage. This requires 30 turns of wire around a T106-26 micrometals iron powder inductor core.

For the voltage ripple condition, we require

$$C \geq \frac{i_{\text{out}}DT}{\Delta v}.$$

I used a 220 µF electrolytic capacitor in parallel with two 1 µF ceramic capacitors to limit the voltage ripple.

The gate driver for the switching MOSFET in the boost converter is an IR2125, which requires a supply voltage of at least 12 volts to function consistently. This means that we could power the IR2125 with the boost converter output, but only if we first get the boost converter to produce that output without the MOSFET. To initiate this, a self-start circuit was also designed. This circuit pulses current into the inductor when the output voltage is less than 12 volts, forcibly charging the output capacitor until the output voltage is 12 volts. The circuit is shown in Fig. 11.

3.2 Results

Figs. 12-14 depict the control signal duty cycle and the output waveform for the boost converter when the input voltage is 6, 7.5, and 10 volts. We can see

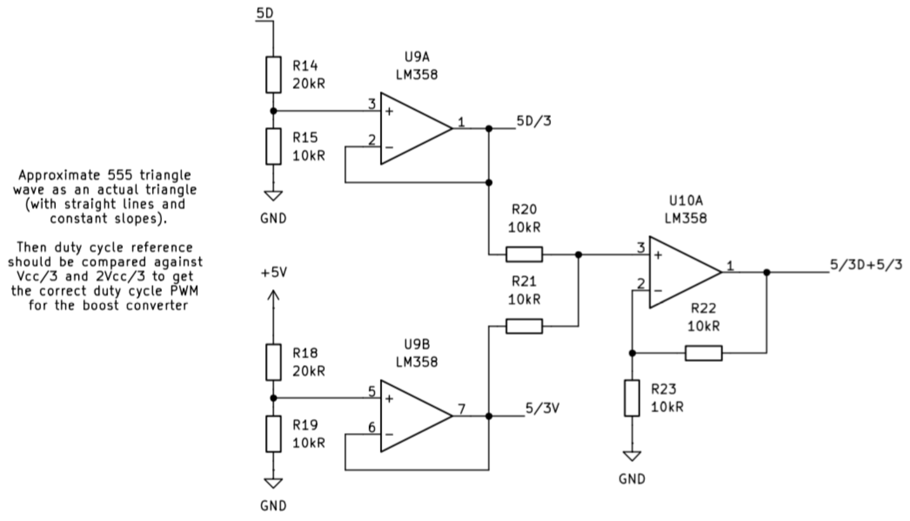

5D
R14
20kR
R15
10kR
GND
U9A
LM358
5D/3
R20
10kR
R21
10kR
U10A
LM358
5/3D+5/3
R22
10kR
R23
10kR
GND
+5V
R18
20kR
R19
10kR
GND
U9B
LM358
5/3V
Approximate 555 triangle wave as an actual triangle (with straight lines and constant slopes).
Then duty cycle reference should be compared against Vcc/3 and 2Vcc/3 to get the correct duty cycle PWM for the boost converter

Figure 9: Buffers and summer.

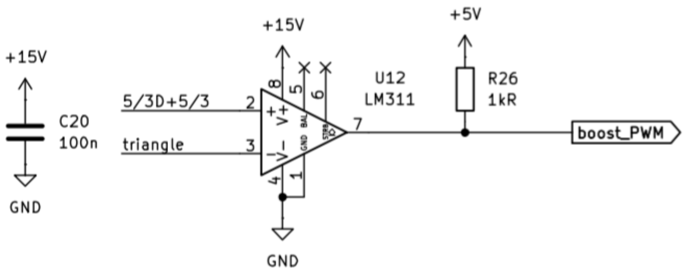


Figure 10: Comparator.

that the duty cycle computation is reasonably close to the correct values that would produce 15 volts on the output, but it is not perfect. Additionally, we can see that the voltage ripple is consistently around 1.2 volts.

4 Conclusion

A photo of the whole setup put together is shown in Fig. 15. When using the boost converter to also power the motor driver and motor, the output voltage drops to around 12 volts. Because of this the self-start circuit switches on and off repeatedly, which leads to poor efficiency. When the motor was not running, the DC power supply that was used to supply 6 volts showed that 0.6 amps of current were being drawn. When the motor was running, the DC power supply showed that 1.3 amps were being drawn. This is an efficiency of a measly 48%. Some solutions for fixing this could be to implement a chopper on the output

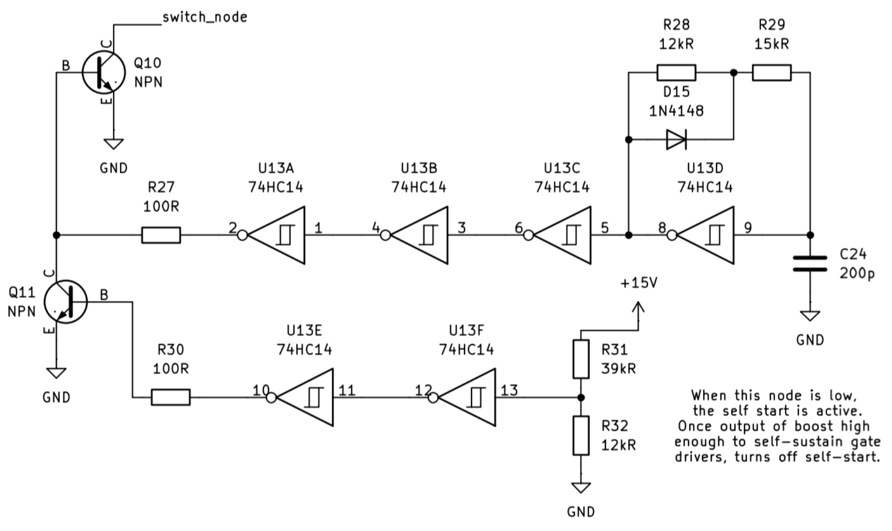
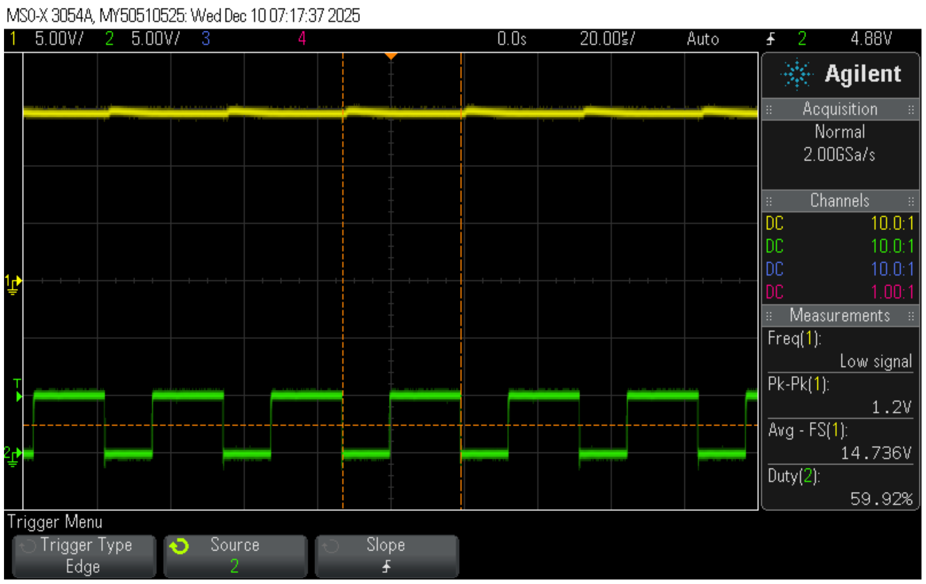


Figure 11: Self-start circuit.

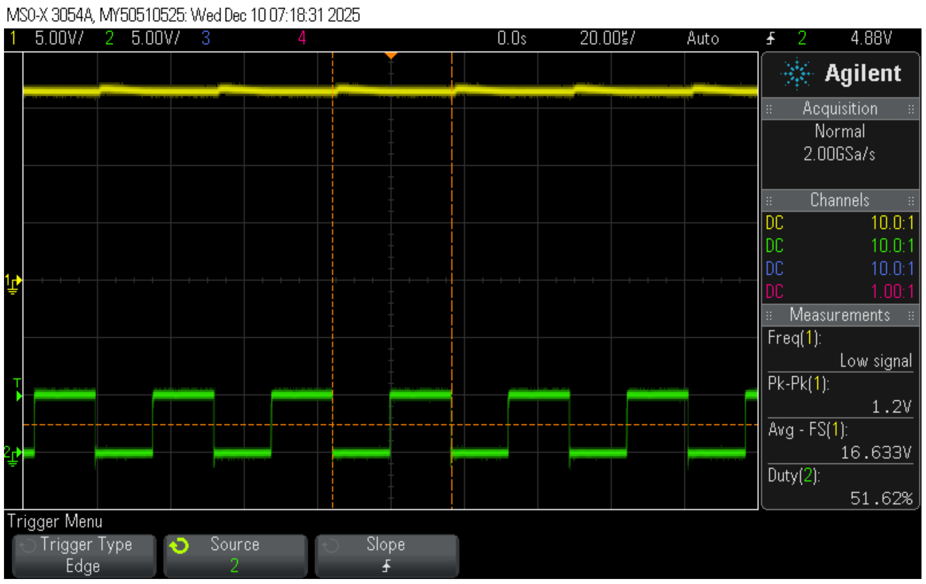
to limit the amount of current being drawn as well as ensure that the output voltage does not drop so low that the self-start circuit has to turn back on.

Despite the poor efficiency, the motor driver did work as intended, and I was able to play "Mary Had a Little Lamb" on the stepper motor. Additional improvements to pursue in the future would be to lay out the whole system on a PCB to reduce wiring problems, to find ways to improve the efficiency of the circuit, to add over-current, over-voltage, and reverse voltage polarity protection, and to integrate multiple motors together to play the several different parts of a song.



```
MSO-X 3054A, MY50510525: Wed Dec 10 07:17:37 2025
1  5.00V/   2  5.00V/   3   4        0.0s   20.00s/   Auto   2   4.88V
Acquisition
Normal
2.00GSa/s
Channels
DC   10.0:1
DC   10.0:1
DC   10.0:1
DC   1.00:1
Measurements
Freq(1):
        Low signal
Pk-Pk(1):
        1.2V
Avg - FS(1):
        14.736V
Duty(2):
        59.92%
Trigger Menu
Trigger Type: Edge   Source: 2   Slope
```

Figure 12: Boost converter output with 6V input.



```
MSO-X 3054A, MY50510525: Wed Dec 10 07:18:31 2025
1  5.00V/   2  5.00V/   3   4        0.0s   20.00s/   Auto   2   4.88V
Acquisition
Normal
2.00GSa/s
Channels
DC   10.0:1
DC   10.0:1
DC   10.0:1
DC   1.00:1
Measurements
Freq(1):
        Low signal
Pk-Pk(1):
        1.2V
Avg - FS(1):
        16.633V
Duty(2):
        51.62%
Trigger Menu
Trigger Type: Edge   Source: 2   Slope
```

Figure 13: Boost converter output with 7.5V input.

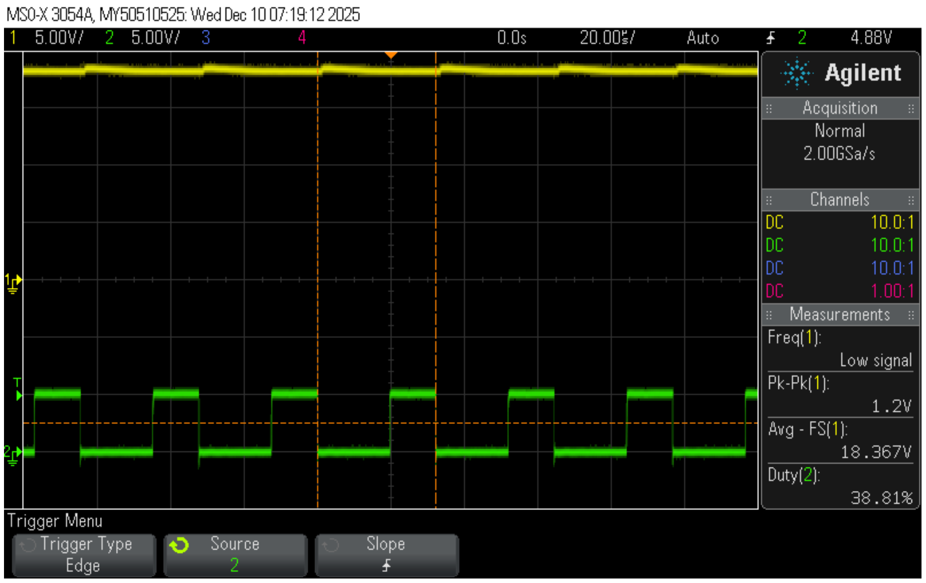


Figure 14: Boost converter output with 10V input.

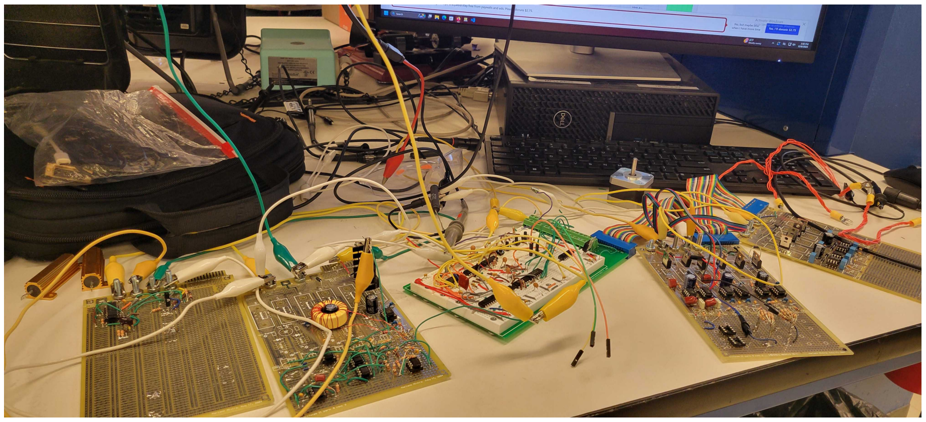


Figure 15: Everything put together.