

Stepper Motor Design Report

Owen Cook, Sarah Wiederman

Daniel Nalepa, Elie Chauvin

April 19, 2025

Contents

1	Abstract	2
2	Introduction	2
3	Requirements	3
4	Results	4
5	Design and Analysis	5
5.1	Logic	7
5.2	Clock	10
5.3	Motor	12
5.4	Counter	15
5.5	Reset	18
5.6	Debounce/Switch	20
6	Limitation and Issues	22
7	Conclusion	23
8	Appendix	24

1 Abstract

Author: Dan

This project involved the design and implementation of a finite state machine to control a stepper motor in four states. These states, Idle, Free Range, Clockwise, and Counterclockwise, were controlled on the basis of three switches and are as follows: Run, Free Range, and Handwave. The finite state machine was built and simulated in Logisim using logic like D flip-flops, a counter, and combinational logic to flip between states. A 555 timer was used as a clock to generate a 4 Hz signal that would be used as the stepping rate. This rate would drive a motor controller to operate a motor to meet the requirements of the design.

2 Introduction

Author: Dan

The goal of this project was to design a digital circuit on hardware for a stepper motor using a finite state machine. The digital system was designed to drive a stepper motor in four states (FR, I, CW, CCW) and, based on the state of three switches (H, R, FR), make adjustments accordingly. In Idle state, the motor needed to stay stationary. In the Free Range state, the motor spins continuously in the clockwise direction until it is deactivated. Finally, in the Clockwise and Counterclockwise states, the circuit would alternate between the two, moving 36 steps clockwise and counterclockwise at a frequency of 4 Hz. Furthermore, the circuit was designed to properly shut down when transitioning from each of the states, ensuring that there was no unnecessary motor movement. To complete these tasks, this team first developed logic and implemented it into Logisim using D flip-flops, combinational logic, and a synchronous counter. Then, on hardware, the team worked on designing and building a synchronous up counter for handwave operations, a 555 timer circuit to produce a 4 Hz signal, and a motor driver circuit to manage the motor's operations utilizing an L297 motor driver and an L297 stepper controller. Once this was complete, the team implemented the logic on hardware and connected each component of the system together to produce the final FSM. This report will further outline the system requirements, the design and methodology, the hardware implementation, and the final results of the project.

3 Requirements

Author: Owen

1. Use only parts in your lab kit, general lab stock, or part of the attached parts list.
2. Use the available lab supplies to power your circuit.
3. The device shall have the following logical inputs, implemented using mechanical switches with two positions (logic true and logic false):
 - **Run** – When false, the circuit will not apply power to the stepper motor coils and enters Idle mode. This overrides all other inputs. If the device is in another mode and Run becomes false, it must perform the shutdown sequence and return to Idle mode.
 - **Handwave** – When true and Free Range is false, the circuit shall enter Handwave mode. The device must perform the Handwave shutdown sequence if false while in Handwave mode.
 - **Free Range** – The circuit shall enter Free Range mode when true and Handwave is false. If false while in Free Range mode, the device must perform the Free Range shutdown sequence.
4. The device shall have 3 operating modes:
 - **Handwave Mode** – Drive the motor 36 steps clockwise, pause 0.25 ± 0.05 s, then 36 steps counterclockwise, pause 0.25 ± 0.05 s. Repeat until Handwave is false, then return to the initial position and enter Idle.
 - **Free Range Mode** – Rotate the motor continuously clockwise. Repeat until Free Range is false, then stop and enter Idle.
 - **Idle Mode** – All motor rotation stops. This is the default state unless conditions are met for another mode.
5. All motor steps shall be commanded at a rate of 4 steps per second $\pm 15\%$.
6. On power-up, the device shall meet all logical requirements of operation. A delay of up to 1 second is allowed before entering a mode, but no uncommanded motor movement may occur.
7. The device shall drive the supplied stepper motor provided to each group.

4 Results

Author: Owen

Requirement	Desired	Result	Error
Clock Frequency	4Hz \pm 15%	4.15Hz	3.75%
Power Up Delay	1s (Max)	489ms	-
Clockwise Steps	36	36	-
Counter-clockwise Steps	36	38	2 Steps

Table 1: Test Results with Error Measurements

The results are shown above. The circuit met all of the requirements except for the counterclockwise steps. Resistor and capacitor values were chosen to set the clock frequency, the power-up delay was selected to be less than the maximum delay and around two clock cycles. The counter-clockwise steps were not consistent with what was expected, but the clockwise steps were. There is Further discussion on the theory and design in the following sections.

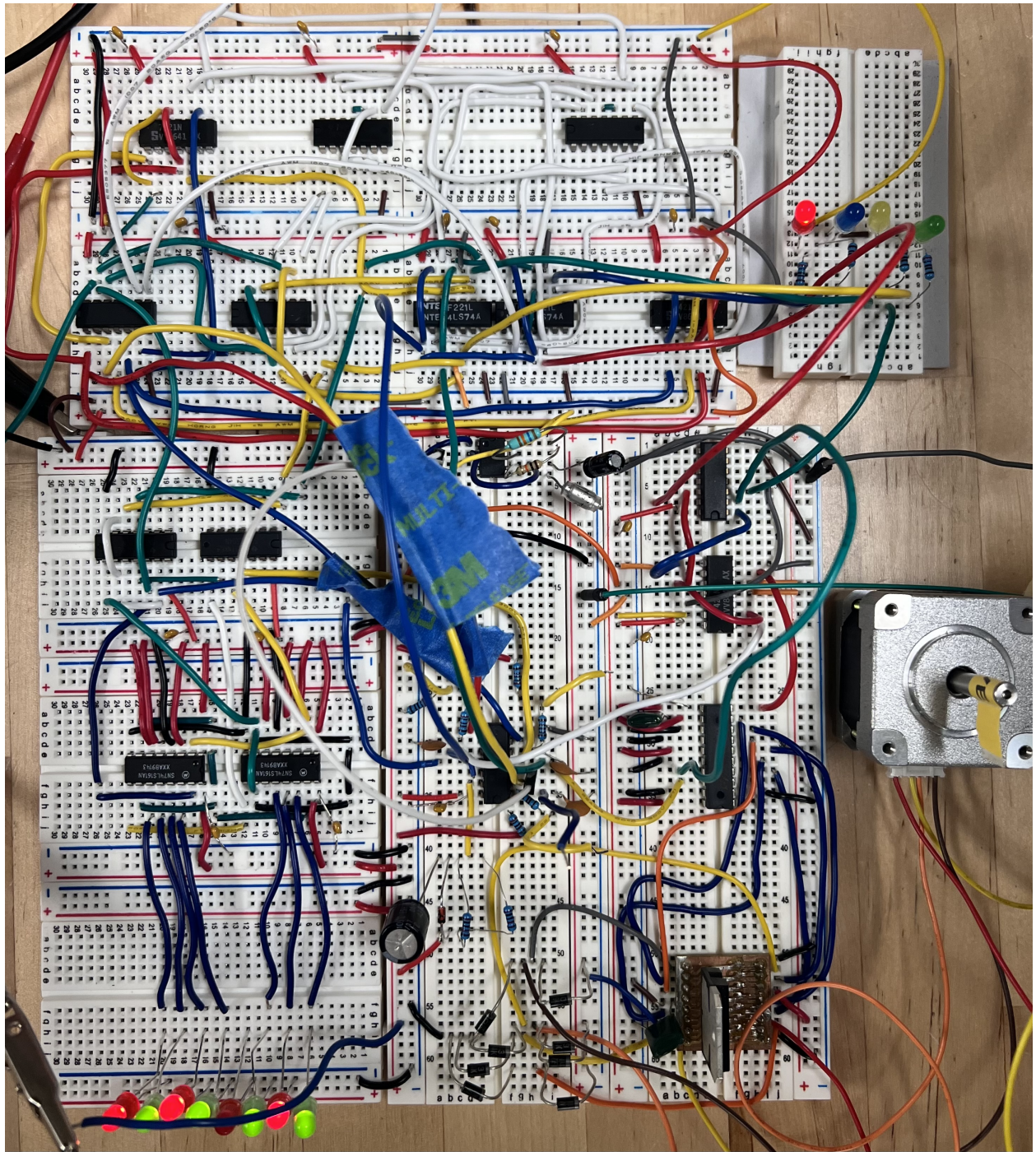


Figure 2: Breadboard Build

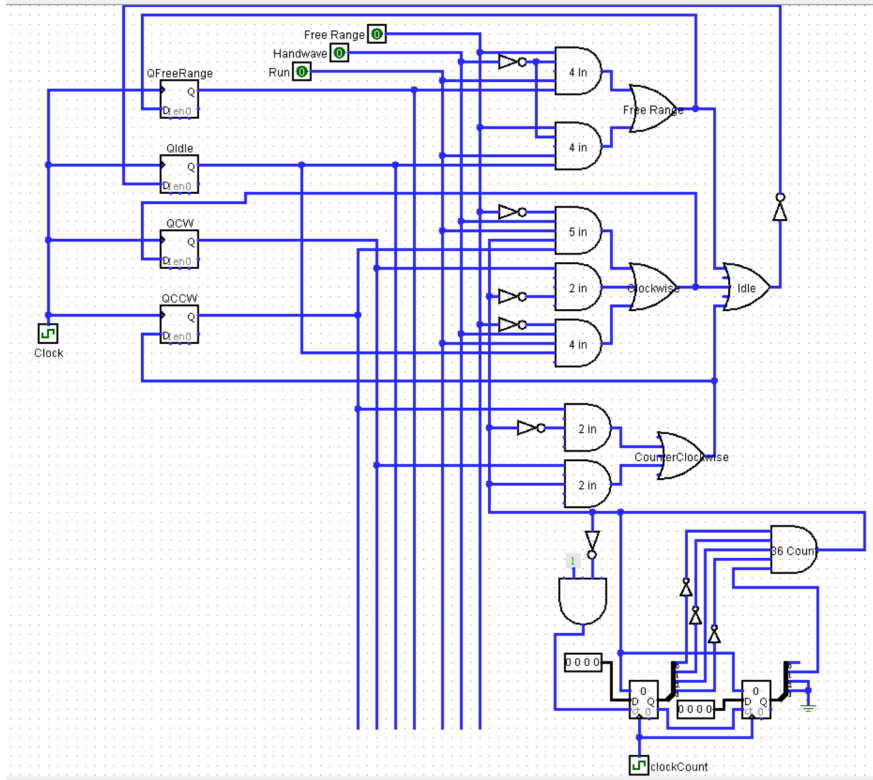


Figure 4: Logisim Implementation of the State Logic

The logic for each state is implemented through Boolean equations derived from the FSM. These equations determine the input for each D flip-flop (D_{FR} , D_{CW} , D_{CCW} , D_I) based on the current-state outputs and input signals. In the clockwise and counter-clockwise states, there is also a "count complete" input that goes high when the counter has counted to 36. This input was added later as a way to signify that 36 steps have been completed to the logic and change the direction of the motor. This clock input is denoted as C .

Transition to Free Range (FR)

The system enters Free Range when it is idle, not already in FR, and the condition $R \cdot F \cdot \overline{H}$ is satisfied:

$$D_{FR} = (Q_I \cdot \overline{Q_{FR}}) \cdot (R \cdot F \cdot \overline{H})$$

Transition to Handwave Clockwise (CW)

CW is activated if the system is in Idle or transitioning from CCW, and the handwave mode is active:

$$D_{CW} = Q_{CCW} \cdot R \cdot H \cdot \overline{F} \cdot C + Q_{CW} \cdot \overline{C} + Q_I \cdot R \cdot H \cdot \overline{F}$$

Simplified using grouping and common terms:

$$D_{CW} = RH\overline{F}(Q_{CCW}C + Q_I) + Q_{CW}\overline{C}$$

Transition to Handwave Counterclockwise (CCW)

CCW occurs after 36 CW steps are completed. It depends on the current count and CW/CCW state:

$$D_{CCW} = Q_{CW} \cdot C + Q_{CCW} \cdot \overline{C}$$

Transition to Idle

The system enters Idle when none of the other state transitions are active. This is implemented as:

$$D_I = \overline{D_{CCW}} \cdot \overline{D_{CW}} \cdot \overline{D_{FR}}$$

This logic was directly implemented in Logisim using logic gates that match these expressions above. Flip-flop outputs represent the current state, and each equation is wired to drive the D input of the corresponding flip-flop. The synchronous counter feeds into CW/CCW transitions, ensuring correct handwave operation. All transitions were tested to confirm timing accuracy and correct mode switching, consistent with project requirements including the 4 steps/sec motor rate and proper shutdown sequencing when inputs are deactivated.

5.2 Clock

Author: Elie

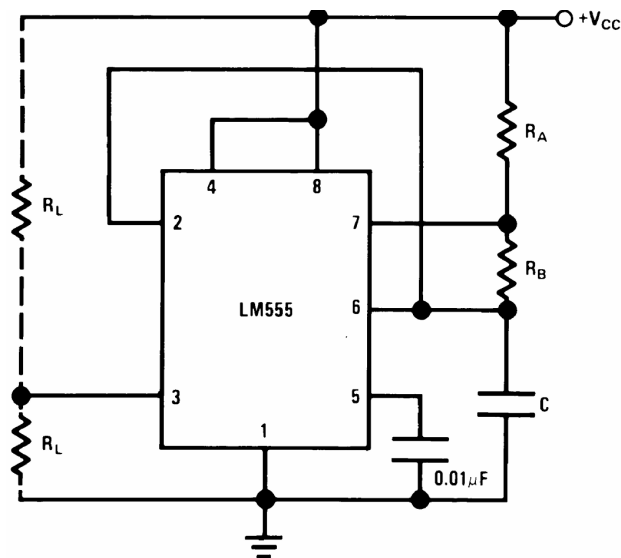


Figure 5: Clock Circuit

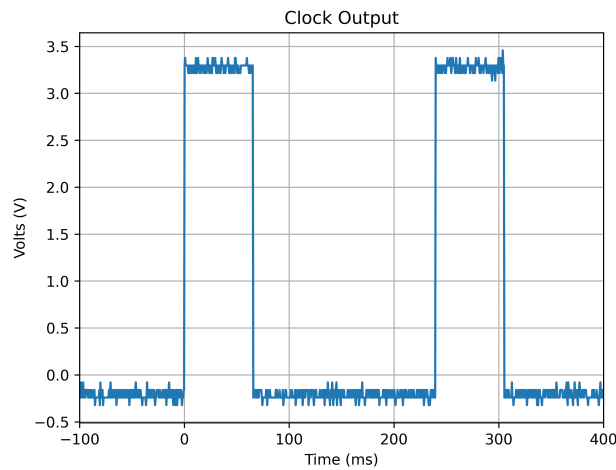


Figure 6: Clock Waveform

The figure for the clock circuit shown above was taken from the 555 timer datasheet. This configuration of the 555 timer is called astable mode because the 555 timer re-triggers itself when the capacitor voltage falls below $1/3$ of V_{cc} , resulting in a square-wave output where the frequency is controlled by R_A , R_B , and C .

$$f = \frac{1.44}{(R_A + 2R_B) * C}$$

In the initial design, the discharge pin(pin 7) was directly connected to V_{CC} . However, in this design, the discharge pin could not properly discharge, so R_A was added. Looking at the formula for frequency above, it proved to be impossible to get a 50 percent duty cycle for the clock, and it was realized that R_A was needed. Because there was more flexibility with the values of resistors than capacitors, the capacitor value was selected to be 10uF. Before R_A was included, R_B was selected to be 18k Ω . However, these resistors could not fit into the breadboard easily. After realizing that R_A needed to be included, R_A was selected to be 10k Ω , and R_B was solved to be 16k Ω . The load resistors were not included in this design, as the output at pin three is the clock output, which drives the circuit. The measured clock frequency was 4.15 Hz, and the target frequency was 4 Hz.

The clock output was also buffered by putting it through a Schmidt Trigger. The rise and fall time of the Schmidt Trigger is faster than that of the clock. Of note is that the clock going to the counter is inverted. The idea behind this is that if the counter output happened at the same time that the motor is trying to turn, the motor enable or CW/CCW logic may change at the same time the motor is trying to turn leading to unexpected behavior.

5.3 Motor

Author: Owen

The first step in designing the motor was choosing what parts to use. Referring to the available parts, it was decided that the L297 motor controller and the L298 motor driver would be used as well as a 42BYG 12V stepper motor. The final schematic can be seen in Figure 1 in the motor section. With extensive examination of these parts's datasheets, the correct pinout was found for each device. The pinout is shown below:

Pin #	Pin Name	Connected To
1	SYNC	Open
2	GND	GND
3	HOME	Open
4	A	IN1 (L298)
5	INH1	INH1 (L298)
6	B	IN2 (L298)
7	C	IN3 (L298)
8	INH2	INH2 (L298)
9	D	IN4 (L298)
10	ENABLE	Enable Logic
11	CTRL	Vcc (5V)
12	Vs	Vcc (5V)
13	SNS1 A	GND
14	SNS2 B	GND
15	Vref	Vcc (5V)
16	OSC	GND
17	CW/ \overline{CCW}	Direction Logic
18	CLOCK	Clock
19	HALF/ \overline{Full}	GND
20	Reset	Vcc (5V)

Table 2: L297 Controller Pin Connections

Pin #	Pin Name	Connected To
1	SNS1	GND
2	OUT1	Motor A
3	OUT2	Motor C
4	V _s	V _{cc} (12V)
5	IN1	A (L297)
6	INH1 (Enable 1)	INH1 (L297)
7	IN2	B (L297)
8	GND	GND
9	V _{ss}	V _{cc} (5V)
10	IN3	C (L297)
11	INH2 (Enable 2)	INH2 (L297)
12	IN4	D (L297)
13	OUT3	Coil B
14	OUT4	Coil D
15	SNS2	GND

Table 3: L298 Driver Pin Connections

Pin #	Pin Name	Connected To
A	Coil A1	OUT1 (L298)
B	Coil B1	OUT2 (L298)
C	Coil A2	OUT3 (L298)
D	Coil B2	OUT4 (L298)

Table 4: Stepper Motor Pin Connections

Most of the pinouts were pulled directly from the datasheets and the application note, but there were a few ambiguities and choices to make about selecting certain modes. One choice that was made was whether to use chopper mode. Chopper mode enables the current to get into the motor coils more quickly as the motors act like an inductor following the below equation:

$$V = L \frac{dI}{dt}$$

With more voltage and constant inductance, the current flows more quickly through the inductor. The motor cannot operate at this high voltage, so the chopper mode sends pulses

of current to either drive the voltage up or down. This mode is most helpful when charging or discharging the coils. Chopper mode is not necessary for this project as there are no requirements in place to discharge the coils in a certain amount of time. The pins used to enable chopper mode were properly configured to ensure chopper mode was disabled. Instructions on how to disable these were found in the application note and datasheet, with few ambiguities. Without chopper mode enabled, the motor coils will discharge through the H-bridge of the L298.

The application note gave a lot of instruction on how to connect the pins to disable chopper mode, but the OSC pin on the L297 was ambiguous. The description of the pin states that the resistor and capacitor connected to this pin controlled the chopping rate. It was assumed that since the chopper is not being used that the pin could be grounded effectively setting the rate to 0. This ended up working with our design and it used less parts, so it stayed.

There is also logic in place that deals with which direction the motor spins. This logic is applied to the CW/\overline{CCW} pin of the L297 motor controller. The logic follows the below equation:

$$CW/\overline{CCW} = \overline{\overline{Q_{Idle}} * \overline{Q_{CCW}}}$$

This ensures that the pin is at a logic high (CW) when in Free Range or CW, and low (CCW) when in Idle or CCW. This logic can likely be simplified, but it worked, so it was kept for the final design.

5.4 Counter

Author: Sarah

To control the number of steps taken by the stepper motor during handwave mode, two 74LS161 4-bit synchronous binary counters were cascaded to function as an 8-bit counter. This configuration allows the system to count from a preset value up to a maximum of 255.

At the beginning of each cycle, the counters are preloaded with the value 219. From there, they count upwards on each clock pulse until they reach 255. This range—from 219 to 255—corresponds to exactly 36 counts, which causes the motor to take 36 steps.

When the counter reaches 255, it triggers a signal that:

- Sends a control signal to the motor
- Activates the **Load** input of the counter through an OR gate, reloading the value 219 to restart the count.

This loop continues until handwave mode is exited and completed, with the counter cycling from 219 to 255, ensuring consistent 36-step increments, until the circuit exits *handwave mode*.

This circuit also has activation logic because counting is not necessary for the free range and idle state. When exiting handwave mode, the counter is again loaded to 219. The counter remains loaded to this value as it is not active in the idle and free range states, if the counter were enabled in these states, it would not be set to the correct value upon re-entrance to handwave mode. The enable logic follows the below equation.

$$\text{ENABLE} = Q_{CW} + Q_{CCW} + \text{Reset}$$

The reason that the reset is included is because the device needs to be loaded on power up and the counter needs to be enabled to be loaded.

The load pin is also tied to logic to ensure that the device is loaded on power up and when the terminal count is reached following the below equation.

$$\text{LOAD} = \text{RCO} + \text{Reset}$$

RCO is the ripple carry out pin which goes high when every bit is 1 and the terminal count is reached.

Pin #	Pin Name	Connected To
1	Clear (MR)	+5V
2	Clock (CP)	Inverted clock output, Clock (2nd counter)
3	A (D0)	+5V
4	B (D1)	+5V
5	C (D2)	GND
6	D (D3)	+5V
7	Enable P (CEP)	Enable P, Or Gate Output - QCW, QCCW, Reset
8	GND	GND
9	Load (PE)	Load (2nd counter), Or Gate Output - RCO (2nd counter), Reset
10	Enable T (CET)	Enable T, Or Gate - QCW, QCCW, Reset
11	Q_D (Q3)	LED
12	Q_C (Q2)	LED
13	Q_B (Q1)	LED
14	Q_A (Q0)	LED
15	Ripple Carry Out (TC)	Enable P, Enable T (2nd counter)
16	V_{CC}	+5V

Table 5: 1st Counter

Pin #	Pin Name	Connected To
1	Clear (MR)	+5V
2	Clock (CP)	Inverted clock output, Clock (1st counter)
3	A (D0)	+5V
4	B (D1)	+5V
5	C (D2)	GND
6	D (D3)	+5V
7	Enable P (CEP)	RCO (1st counter)
8	GND	GND
9	Load (PE)	Load (2nd counter)
10	Enable T (CET)	RCO (1st counter)
11	Q _D (Q3)	LED
12	Q _C (Q2)	LED
13	Q _B (Q1)	LED
14	Q _A (Q0)	LED
15	Ripple Carry Out (TC)	OR Gate Input to Load (1st counter)
16	V _{CC}	+5V

Table 6: 2nd Counter

5.5 Reset

Author: Owen

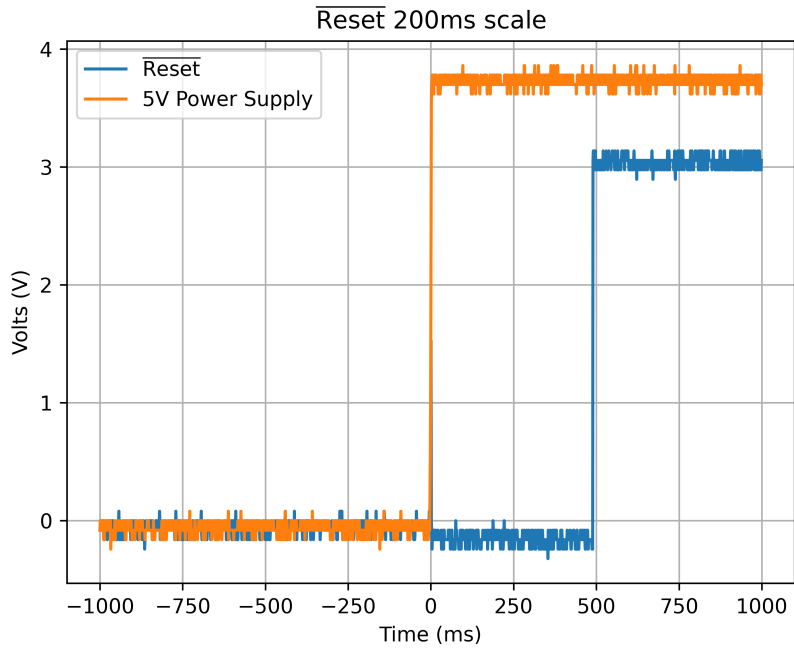


Figure 7: Active Low Reset

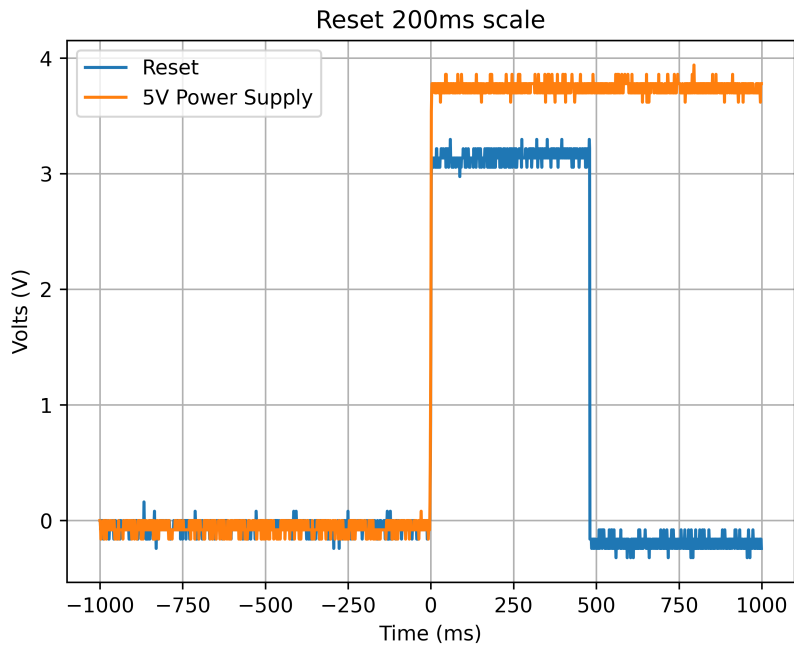


Figure 8: Active High Reset

The reset circuit is key for the circuit to work. On power up, the logic gates and other chips could see a voltage that is not at the correct value that it will be. The reset circuit and bypass capacitors ensure the circuit comes up in a known state on power up and the voltage near the chip stays at a constant level.

The circuit can be seen in the top right of Figure 1. The circuit consists of a resistor and a capacitor such that the time constant is less than one second as to satisfy the one second power up delay requirement. A time of 0.5s of to have the reset circuit active was chosen and the following equation was used to find the resistor and capacitor.

$$V(t) = V_{CC}(1 - e^{-\frac{t}{RC}})$$

A resistor value of $10k\Omega$ was chosen to get a capacitor in the hundreds of milliFarad range as the lab had these in stock. The capacitor value came out to be around $220\mu F$. Two $20k\Omega$ resistors in parallel were used as well as a $220\mu F$ capacitor. A diode was put in parallel with the resistor with the cathode at the 5V power supply. This diode allows for faster discharge when the circuit is powered off. The current can discharge through the diode which is closer to a short rather than the resistor, so there will be little to no time constant.

The connection between the resistor and the capacitor is then connected to two Schmidt Triggers, which is an inverter with hysteresis. This device has fast rise and fall time. On power up, the first schmidt trigger will stay low until the voltage across the capacitor reaches the threshold of the trigger, then will transition low and stay low while the circuit is powered.

This reset logic is used in almost every part of the circuit. The D flip-flops are set to come up in idle state as each mode can be entered from idle mode, the stepper motor is set to be disabled to prevent unexpected stepping, and the counter is loaded with its initial value to prepare to enter handwave mode. This ensures that the circuit comes up in a known state and is ready to transition to any other state without violating any of the other requirements.

5.6 Debounce/Switch

Author: Sarah

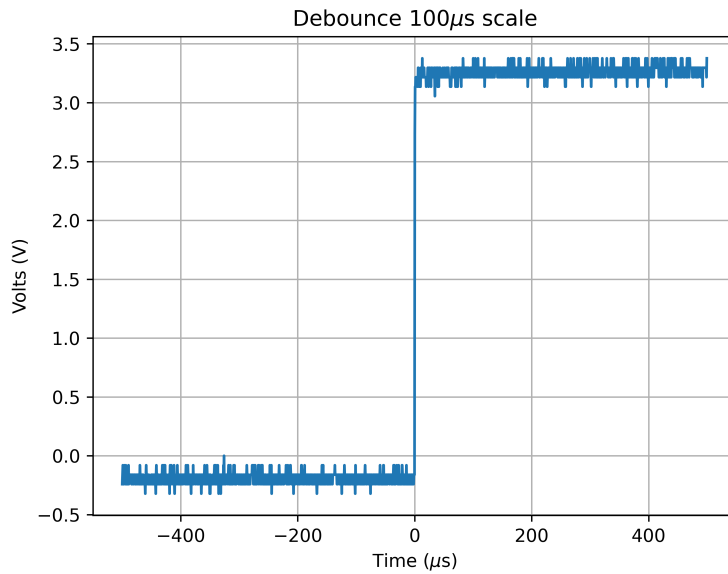


Figure 9: Debounce 100µs Scale

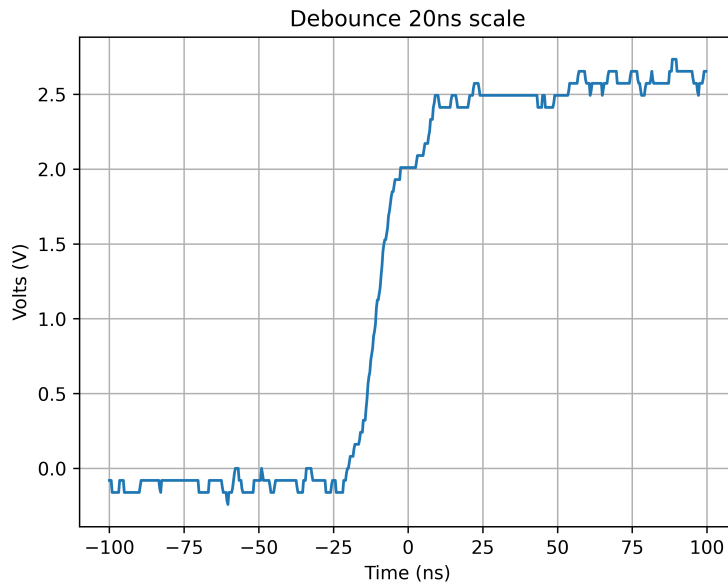


Figure 10: Debounce 20ns Scale

Instead of an off-the-shelf switch, the circuit used wires for each of the three inputs which can be left floating when not activated, and grounded to activate. This was preferred over the switches because of the design's simplicity, and ease of use and troubleshooting. We

found that the commercial switches available were very small, which made it more difficult than desired to switch on and off.

In the design, a debounce circuit was implemented to manage the noisy output from a mechanical switch, which can generate multiple signals due to contact bounce. This was done to prevent interference with the rest of the circuit.

A simple circuit using a capacitor was employed to filter out the unwanted bounces, ensuring that only a single pulse is registered per switch actuation. However, this method relies on the switch's initial bounce to fully discharge the capacitor, which may not always occur. To deal with this issue, a second resistor was added, further stabilizing the signal by filtering out both rising and falling edges. However, this means that the edges of the signal may become slow and non-monotonic, which could potentially cause double triggering at certain voltage thresholds. To address this, a Schmitt trigger inverter was used as well. The Schmitt trigger adds hysteresis to the input signal, ensuring a clear distinction between the rising and falling edges, preventing unintended triggers from slow or non-monotonic signal behavior. This solution not only stabilizes the signal but also minimizes power consumption by avoiding excessive current from the slow input edges.

In the original design, somewhat arbitrary values for the resistors were chosen to be 6.8kOhms. However, when testing the circuit we found that these resistors were too high as they were limiting the current necessary to power the rest of the circuit. To fix this, we swapped the resistors with 2kOhm resistors.

6 Limitation and Issues

Author: Owen

Chopper mode was not necessary in this circuit as there were no requirements to include it, but including this would allow for faster charge and discharge of the coils. The datasheets and application note for the stepper motor parts contain clear instruction on how to use chopper mode if one wanted to enable it.

One issue that arose at the end of the project is that when in handwave mode, the motor takes 36 steps clockwise followed by 38 steps counter-clockwise. The true reason behind this is unknown to the group, but some theories arose including the inversion of the clock causing the issue. The clock is not exactly 50% duty cycle, so when the clock is inverted, the motor may turn an extra turn, while the counter output is high. This could also be due to all the inverting of the clock, the inversion causes a slight buffer, which could delay the motor receiving the signal to change directions.

Another issue with the circuit is that when the power is activated and deactivated in quick succession, the reset circuit does not properly set the circuit for correct use. The counter would not get loaded in this case, causing a lot more than 36 steps. The reason for this being that the capacitor is likely charged and not fully discharging before the power is reapplied, so the active high and active low reset never return high or low respectively, and the circuit is not reset. Removing the diode from the reset circuit would likely solve this issue, but another issue could arise from the slow discharge. A more involved fix for this includes a comparator tied to the 5V power supply to check whether to discharge the capacitor quickly or slowly.

In some of the figures shown throughout the report, the 5V power supply is not truly at 5V. This is likely because the voltage is being applied at one point and the busses are connected via wires which both have resistance.

7 Conclusion

Overall, this project successfully demonstrated that a finite state machine could be designed and built to control a stepper motor over four modes of operation, using combinational and sequential digital logic, a 555 timer clock circuit, and a motor driver circuit. Although there were small discrepancies, such as a 38-step observation in the counter-clockwise direction, the system functioned reliably overall. Designing the logic and integrating the hardware smoothly were both extremely important because factors such as signal timing, nonidealities, and reset behavior had to be factored into the build of the FSM. This project provided valuable insight into debugging large systems, implementing logic on hardware, and reading datasheets to effectively understand each component of a circuit.

8 Appendix

References

- [1] Texas Instruments, *LM317 3-Terminal Adjustable Regulator*, Online datasheet, Accessed April 2025. [Online]. Available: https://drive.google.com/open?id=1FIAP8yjbBL8y_sByZREtxvi4zAw_HYIh
- [2] STMicroelectronics, *L78M05 Linear Voltage Regulator*, Online datasheet, Accessed April 2025. [Online]. Available: <https://drive.google.com/open?id=1Vqb-z14wk7jjFnv91I22TKBDW5FggvDV>
- [3] Murata, *GRM Series Multilayer Ceramic Capacitors*, Online datasheet, Accessed April 2025. [Online]. Available: https://drive.google.com/open?id=151P1dE5VR2wkFJX7BhC4zQKF0t_ZYjxB
- [4] Vishay, *1N4148 Switching Diode*, Online datasheet, Accessed April 2025. [Online]. Available: <https://drive.google.com/open?id=1ACnhtwPPioqX7B17npvf5YfZWlugHjaD>
- [5] Fairchild Semiconductor, *2N3904 NPN Transistor*, Online datasheet, Accessed April 2025. [Online]. Available: <https://drive.google.com/open?id=1Uw1q6c24iNVDG0EmqZLYti9UIada9bHx>
- [6] Analog Devices, *AD620 Instrumentation Amplifier*, Online datasheet, Accessed April 2025. [Online]. Available: <https://drive.google.com/open?id=1zR01KsGXkQK5LW6Ww3L3pCjcLQTFqLpy>
- [7] Texas Instruments, *LS74161 4-bit Counter*, Online datasheet, Accessed April 2025. [Online]. Available: https://www.ti.com/lit/ds/symlink/sn54ls161a.pdf?ts=1744926988303&ref_url=https%253A%252F%252Fwww.google.com%252F
- [8] Texas instruments, Online datasheet, Accessed April 2025 [Online]. Available: *555 Timer* <https://www.ti.com/lit/ds/symlink/lm555.pdf>

Test Plan

Stepper Motor Test Plan

Test Plan

1010. (Verifies Parts Used) Verify that parts used in the circuit were all from Votey 334.
1020. (Verifies Power Supply) Verify that the circuit operates with the power supplies in Votey 334.
1030. (Verifies Coil Discharge) Connect an ammeter to the coils of the stepper motor in Idle mode and verify that this current is less than 1mA.
1040. (Verifies Handwave Step Count) Count the steps during handwave mode to verify that 36 steps are taken clockwise and then counterclockwise.
1050. (Verifies Pause in Handwave) Connect an oscilloscope to the clock pin of the LM297 to ensure that the motor pauses for $0.25s \pm 0.050s$ before changing directions in handwave mode.
1060. (Verifies Return to Start Position in Handwave) Verify that when exiting the handwave mode the motor returns to its initial position.
1070. (Verifies Free Range Mode) Verify that the motor spins freely in hand wave mode.
1080. (Verifies 4Hz Clock Input) Connect an oscilloscope to the clock input of the LM297, and verify that the input is receiving a square wave signal of $4Hz \pm 15\%$.
1090. (Verifies Function on Power Up) Verify the circuit successfully transitions between all states after power is applied.

Test Results

Student Signature _____ Instructor/TA Signature _____

Test Num	Description	Pass (Y/N)	Measurements/Notes
1020	Verify that the circuit operates with the power supplies in Votey 334.	✓	
1030	Connect an ammeter to the coils of the stepper motor in Idle mode and verify that this current is less than 1mA.	✓	
1040	Count the steps during handwave mode to verify that 36 steps are taken clockwise and then counterclockwise.	X	36 CW 38 CCW
1050	Connect an oscilloscope to the clock pin of the LM297 to ensure that the motor pauses for $0.25s \pm 0.050s$ before changing directions in handwave mode.	✓	
1060	Verify that when exiting the handwave mode the motor returns to its initial position.	X	25 steps further
1070	Verify that the motor spins freely in hand wave mode.	✓	
1080	Connect an oscilloscope to the clock input of the LM297, and verify that the input is receiving a square wave signal of 4Hz $\pm 15\%$.	✓	4.15 Hz
1090	Verify the circuit successfully transitions between all states after power is applied.	✓	