

The 555 Timer

Author: Michael Dimitrov

Section: 009 Mingqi Yang

Date: 1110/12/2025

Abstract

The 555 Timer is one of the most used Integrated Circuits for hobbyists in electronics. Understanding its function and use is important for being able to add timing into a circuit, and can be used for anything from traffic lights, to police sirens, to waveform generators.

The first experiment of this report will explore using the 555 Timer in Monostable vs Astable mode. A Monostable Timer is triggered by a short pulse (like a button press) and outputs “high” voltage for a programmable length of time. This is programmable by the resistor capacitor combinations of the circuit. Conversely an Astable Timer runs on a loop, outputting sections of high and low voltage, which are again programmable by the resistor and capacitor combinations of the circuit. The Monostable vs Astable Experiment will compare these two modes and analyze their oscilloscope readings, as well as investigate how to easily program the durations of high and low voltage and the period, using a potentiometer.

The second experiment of this report will explore the design process for a state machine using the 555 Timer as an output to signal state transitions. The design uses binary counter, a demultiplexer, and several 555 Timers. There is also NAND gate logic in between the different sub-circuits that help convert outputs of one subcircuit into the required input of another.

1. Objectives

Explore how to use the 555 Timer to control output voltage in various ways. Design a circuit that uses the 555 Timer in a state machine.

1.1. Monostable vs Astable Timer

Compare the operation of the 555 Timer in Monostable and Astable modes. Discuss the differences and how to set up each configuration. Explore how to make easily adjustable timings using a potentiometer. Verify the timing equations discussed in the lab manual for each configuration and explain why they may not match exactly with the measured values. Analyze oscilloscope readings of the input, output, and capacitor voltages of the 555 Timing circuit (in both monostable and astable modes) and explore the relationships between them.

1.2. State Machine Experiment

Design a state machine that counts the number of times a button is pressed, and upon transition to a new state, pulses the LED corresponding to that state using the 555 Timer circuit. Explain the design process, and the considerations that went into each component. Analyze the oscilloscope readings of the input button presses and compare it to the output voltages of the LEDs of the different states. Discuss the real-world use cases of such a circuit and analyze improvements that can be made to the design in the future.

This circuit interests me because as a CS major, we have learned about theoretical state machines, and drawn flow charts of them, however, it would be very cool to have a real-life state machine, and stores its state as a binary count, decodes the binary count into one of 4 “states” and then does a task (in this case pulsing and LED) when a new state is reached. It will also be an interesting challenge to convert pulses (from the input button) into states (which are a constant voltage), and then back into a pulse to trigger the correct 555 Timer circuit. It will require me to use NAND gates that I learned about in systems architectures class, but have never really used in any complex circuit.

2. Theory

2.1. Monostable Circuit

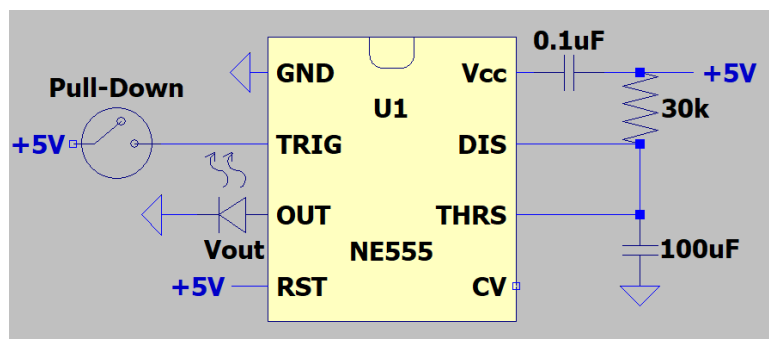


Figure 1: 555 Timer Monostable Circuit

General Flow of a 555 Timer when in Monostable mode:

1. The trigger pin is set to low.
2. The flip-flop inside the 555 integrated chip is set.

3. This causes a short circuit across the 100uF capacitor, which drives the output to high.
4. The voltage across the capacitor exponentially increases for $1.1 * R_A * C$ seconds.
 - a. (In Figure 1 $R_A = 30k \text{ ohm}$ and $C = 100\mu F$)
5. The comparator within the 555 chip resets the flip-flop.
6. The capacitor is discharged (since the short no longer goes across it), driving the output to low.
7. Thus, the pulse is completed

In steps 1, 2, 6, 7 the output voltage of the chip (V_{out}) is high, and in steps 3, 4, 5 the output voltage is low. The output voltage in the above schematic is measured across an LED (The LED is used for visualizing the output when performing the experiment).

Note: Figure 1 is a schematic for a 555 Monostable Timer that is “normally on”. This means that the input to the trigger needs to be high by default, and when you press the button, this will pull down the voltage to 0. What should look like on the output of the chip is that when the button is not pressed, the output will be low, and when it is pressed, the output will be high. In contrast, a “normally off” version of the timer will require the input to be low by default, with pressing the button pulling the input up, and it will exhibit the same output behavior.

As mentioned, the charge time for the capacitor (also the time the output will be high) T_{high} is:

$$T_{high} = 1.1R_A C \quad (1)$$

During the experiment we will verify this equation holds true, by choosing values for a Monostable Timer that will have high output for 3.3 seconds. Using $R_A = 30K \text{ Ohm}$ and $C = 100 \text{ uF}$, we can plug into equation 1 and see that this combination of resistor and capacitor in fact yield a high output of 3.3 seconds.

2.2. Astable Circuit

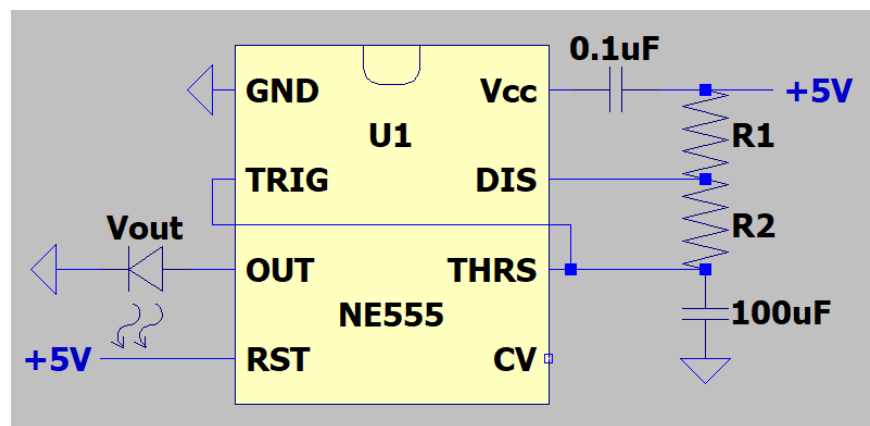


Figure 2: 555 Timer Astable Circuit

General flow of the 555 Timer when in Astable mode:

1. The trigger pin is set to low.
2. The flip-flop inside the 555 integrated chip is set.
3. This causes a short circuit across the 100uF capacitor, which drives the output to high.
4. The voltage across the capacitor exponentially increases until the capacitor reaches $2/3 V_{cc}$.

5. The comparator within the 555 chip resets the flip-flop.
6. The capacitor is discharged to $1/3 V_{cc}$ driving the output to high
7. The process repeats from step 1.
 - a. This is because the trigger/threshold pin is being compared with the capacitor and once the capacitor goes below $1/3 V_{cc}$, the trigger is below the voltage divided V_{cc} voltage (of $1/3 V_{cc}$).

In steps 1, 2, 6, 7 the output voltage of the chip (V_{out}) is high, and in steps 3, 4, 5 the output voltage is low. The output voltage in the above schematic is measured across an LED (The LED is used for visualizing the output when performing the experiment).

The following equations define the charge and discharge time of the capacitor (100uF) (high and low output voltage):

$$Duty\ Cycle = \frac{T_{high}}{T_{low}} = \frac{R_1 + R_2}{R_1 + 2R_2} \quad (2)$$

$$T_{high} = 0.693(R_1 + R_2)C \quad (3)$$

$$T_{low} = 0.693(R_2)C \quad (4)$$

During the experiment we will verify these equations by choosing resistor and capacitor values that get a 60% duty cycle with a 2 second period, and a 75% duty cycle with a 1 second period. 60% Duty cycle means 60% of the period is High output with the other 40% being Low. For a 2-second period this breaks down into 1.2 seconds High, 0.8 seconds Low. Equivalent calculations are made for 75% Duty Cycle Below are theoretical resistor and capacitor pairings calculated from the above equations:

60% Duty Cycle 2 second Period	75% Duty Cycle 1 second Period
C = 100uF	C = 100uF
$0.8 = 0.693 * C * R_2 \rightarrow R_2 = 11544 \text{ Ohm}$	$0.25 = 0.693 * C * R_2 \rightarrow R_2 = 3607 \text{ Ohm}$
$1.2 = 0.693 * C * (R_1 + R_2) \rightarrow R_1 = 5772 \text{ Ohm}$	$0.75 = 0.693 * C * (R_1 + R_2) \rightarrow R_1 = 7215 \text{ Ohm}$

Table 1: Theoretical component values for circuit in Figure 2

2.3. State Machine Circuit

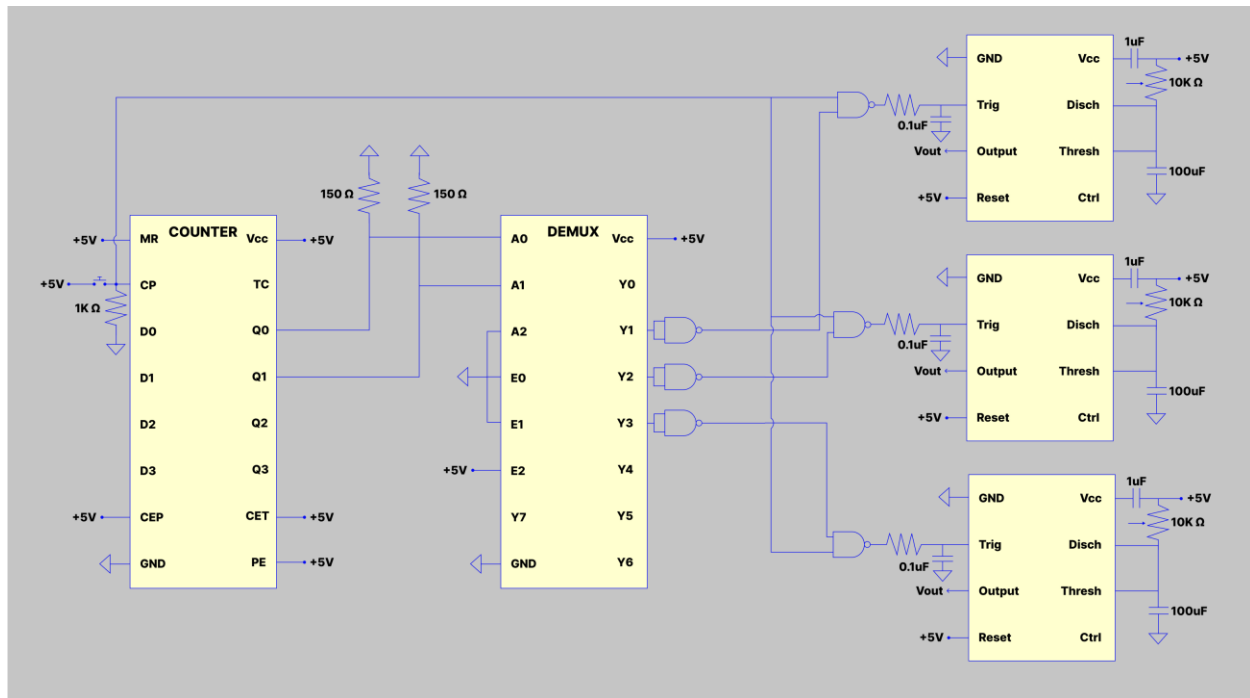


Figure 3: State Machine Circuit

There were many considerations that went into the design of this circuit. The overall goal was to create a circuit that stores its state, and when a state transition happens, the LED corresponding to the new state is pulsed (for a programmable length of time).

The first challenge was to design a way to store the state. I opted to store it as a binary number representing the number of times the input button is tapped. I wanted to have 4 states – I numbered them 0, 1, 2, 3 – with LEDs “none”, Red, green, yellow respectively. Thus, the default state is none, then when the button is tapped once, the red-light flashes. The second press triggers the yellow light. And the third press triggers the green light.

A binary counter (74hc160n) is used to increment the state (count) each time the button is pressed. This particular chip can count in binary numbers 0-9. Thus, since I only need numbers 0-3, I will simply use the first 2 pins (Q0 and Q1 in the schematic above) as my binary outputs. The MR (Master Reset) pin is always high because I want to be able to transition states. Having it powered low will prevent that. Note: In my actual circuit that I built I attached a pulldown button to the MR pin to be able to easily reset the state back to 0 for testing (this may be helpful for anyone rebuilding the circuit). The CP pin is the input of my push down button. When voltage is supplied, it increases the count. I added a 1K pull down resistor to help debounce the button input, as sometimes states would be skipped due to “bouncing” in the button’s output. The CEP, PE, and CET pins are simply parameters that define the operation of the chip. For the purposes of this circuit, I high powered all of them.

The outputs of the binary counter that are relevant to this circuit (Q0 and Q1) are inputted into the Demultiplexer (DEMUX - 74HC138). I again added pull down resistors (150 Ohm) to help with any bouncing coming from the binary counter. The way the DEMUX works is the select pins (A0-A2) control which state is being selected. They take in a binary number where A2 is the most significant bit and A0 is

the least significant. Since I don't need numbers (states) larger than 3, I will keep A2 low powered (0) and A1 and A0 will get their input from Q0 and Q1 – thus converting a binary number into a state from 0-3. The “Control” pins (E0-E2) tell the circuit which output pins (Y0-Y7) are being used. Since I want all of them, I power the E pins in the configuration in the circuit schematic (taken from the data sheet). Since state 0 is my null state – where I don't want LEDs to be on – I do not take output from pin Y0. It is important to note that the output pin corresponding to the selected state will be LOW powered, while the unselected states are HIGH powered.

There are 3 555 Timers that can be seen in the left of Figure 3. These each have an LED on them at the output. The goal is when a state is switched, the corresponding Timer is triggered with a pulse (since this monostable configuration of the 555 Timers require a low pulse, to trigger). I am using a 10K potentiometer to be able to easily adjust the amount of time each LED stays lit. This configuration is very similar to the configuration of the monostable circuit from section 2.1. The only difference is the input. I attached a small debounce circuit to the input of each 555 Timer using a resistor and capacitor, that helps to smooth the voltages coming from the DEMUX and makes it so that the wrong LED is not accidentally triggered along with the correct one.

The DEMUX outputs a constant voltage for each state (where unselected is HIGH and selected is LOW). This must be converted into a pulse for the 555 Timer. The way I chose to do this was to compare each of the output states of the DEMUX with the button using NAND gate logic. The idea is when both the button and the DEMUX output meet a condition, then the output of the NAND logic will go from high to low, triggering the corresponding 555 Timer. Since the output of the DEMUX is low for selected states, and high for unselected ones, I invert this by first connecting each Y bin of the DEMUX into both inputs of a NAND gate (this acts as an inverter). Now, selected states will be high, and unselected will be low. I then use a second NAND gate for each state to compare the button with the inverted state output from the DEMUX. When both are high, the NAND will output low (but only while the button is pressed thus acting as a pulse). In any other case, the NANDs will output high. Thus, I have created a pulse by comparing the state output to the button press, in such a way that the signal goes from high to low when the button is pressed, and only goes to the correct 555 Timer for the selected state.

Thus, the LED for the newly selected state will be triggered, as per the specifications described in section 1.

3. Procedure

3.1. Monostable vs Astable Timer| Monostable Circuit

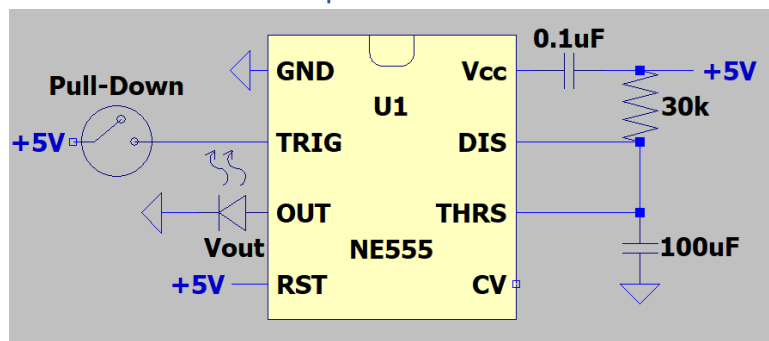


Figure 4: 555 Timer Monostable Circuit

The circuit in Figure 4 was built according to the components shown in Figure 3. For the 30K Ohm resistor, 3 10K Ohm resistors were used as the lab kit did not include a 30K Ohm resistor. The voltage was measured with an oscilloscope in Roll Mode at the output of the Pull-Down switch (equivalent to the input to the 555 timer), at Vout (across the LED), and across the 100uF capacitor.

Once the circuit is constructed and required voltage is provided, press the pull-down switch to observe the LED turn on for 3.3 seconds, then turn off. The equivalent should be shown in the oscilloscope readings. I will explain my results in relation to the expected results more in depth in the Results, and Conclusion sections

3.2. Monostable vs Astable Timer| Astable Circuit

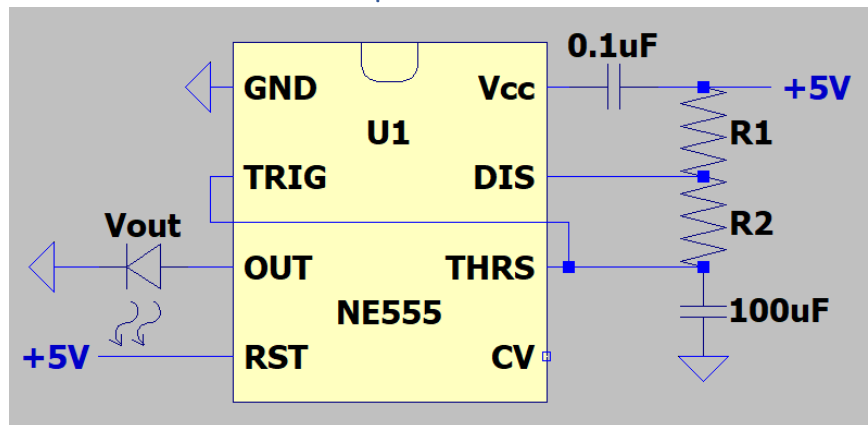


Figure 5: 555 Timer Astable Circuit

Duty Cycle Period	Components	Values
60% Duty Cycle 2s Period	R1	5.6K Ohm
	R2	11K Ohm = 10K + 1K Ohm
75% Duty Cycle 1s Period	R1	7.1K Ohm = 5.6K + 1.5K Ohm
	R2	3.6K Ohm = 3.3K + 330K Ohm

Table 6: 555 Astable Circuit Values

The circuit in Figure 5 was built according to the components listed in Table 2, for each stage. First the 60% Duty Cycle with 2 second period was constructed. The output was measured across Vout and across the 100uF capacitor using an oscilloscope in roll mode.

This was then repeated the same for the 75% Duty Cycle with 1 second period.

Lastly, R1 and R2 were replaced with a 10K potentiometer to dynamically adjust the timing of the circuit.

3.3. State Machine Experiment

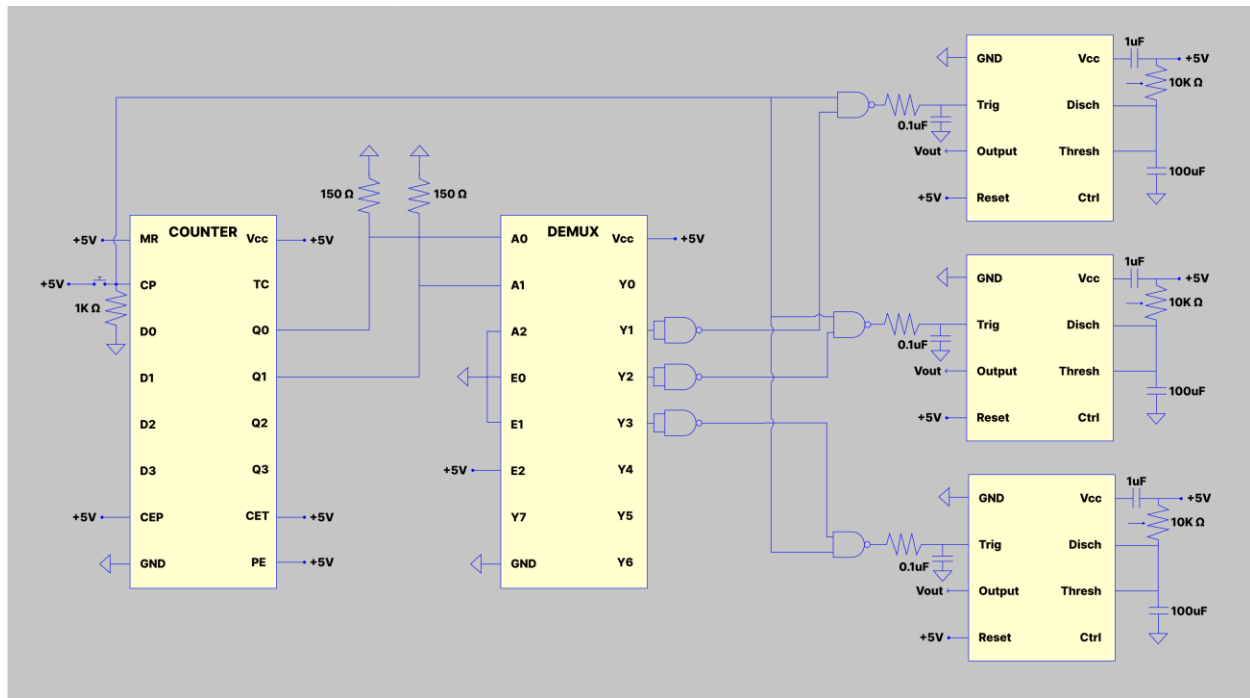


Figure 7: State Machine Circuit

The state machine shown in Figure 6 was built according to the components in the schematic. Additionally, LEDs were added at the outputs of each 555 Timer circuit (at Vout). LEDs were also added at the Y0, Y1, Y2, and Y3 of the DEMUX to easily track the states. More LEDs were added at Q0-Q3 to see what the binary count is. All of these LEDs were mainly for debugging the circuit but were extremely helpful for decoding between the different stages of the circuit. A pull-down button was also added at the MR (master reset) pin of the Binary Counter, in order to reset the state. This is because the counter counts up to 9, but the circuit only deals with 4 states. Ideally, up to 10 states could be added, or a binary counter that only counts to 4, however, using a reset button worked as well. Each time state 4 is reached, the master reset must be pressed. It must also be pressed when connecting power, to ensure the binary count is 0.

For my NAND gates I opted to use the 74hc00n integrated circuit - two input quad NAND gates. I needed 2 of them (since each one has 4 NAND gates built-in and I needed 6 total). I also used 1 binary counter (74hc160n), 1 DEMUX (74HC138), and 3 LMC555 Timers. (The schematic shows NE555 because LTSpice only had that one).

The voltage was measured using an oscilloscope across the output of each 555 Timer, and at the input of the button (the one attached to the binary counter CP pin).

4. Results

4.1. Monostable vs Astable Timer| Monostable Circuit

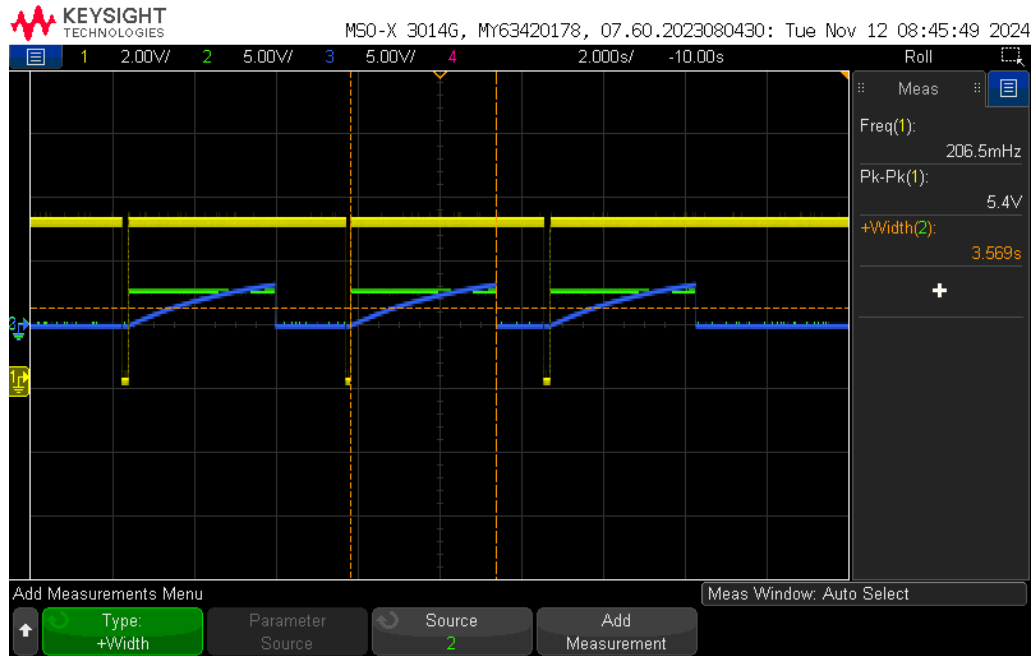


Figure 8: Monostable - Switch output (yellow), Timer Output (Green), Capacitor Voltage (Blue) - measured on oscilloscope.

Figure 7 shows the relevant voltage readings for the monostable circuit. When the Switch output voltage (yellow line) drops is when the button is pressed. This triggers the capacitor (blue line) to start charging. While the capacitor is charging up, the output of the chip (green) is high. Refer to Equation 1 for the capacitor charging time. As shown in the oscilloscope readings, this is equivalent to the time the output voltage is high.

The “Width” measurement in Figure 7 of Channel 2 (the output of the 555 timer), shows that the output is high for 3.569 seconds. I recall that the desired time (as calculated in the theory section of the report) was 3.3 seconds. The measured T_{high} is close to the expected with 8.15% error.

4.2. Monostable vs Astable Timer| Astable Circuit

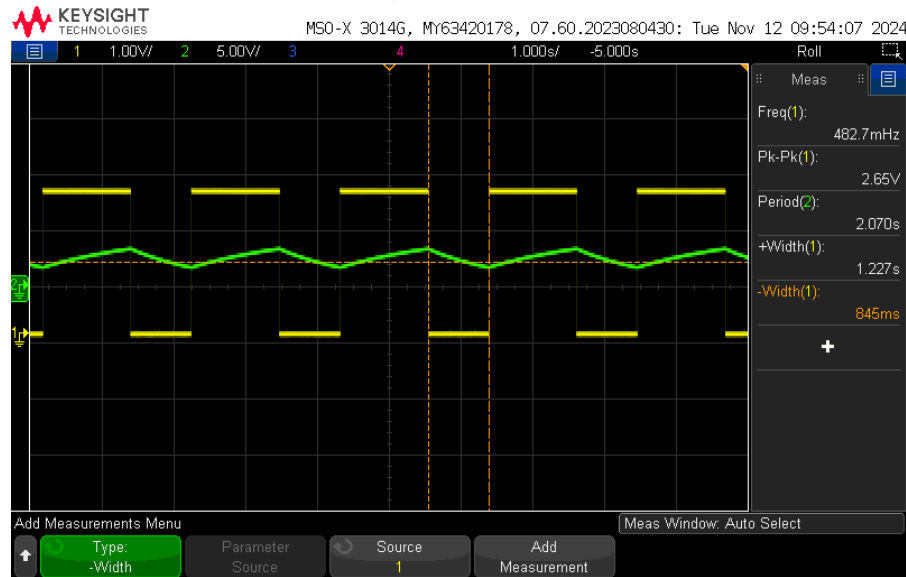


Figure 9: Astable Circuit with 60% Duty Cycle – Timer output (yellow), Capacitor Voltage (Green) - measured on oscilloscope.

Figure 8 shows the relationship between the Timer output and the capacitor voltage for the Astable Circuit with 60% Duty Cycle. As the capacitor (green line) charges, the output is high, and as it discharges, the output is low.

Recall that in the Theory section we calculated that for a 60% Duty Cycle with 2 second period, 1.2 seconds of output is supposed to be high, and 0.8 low. In Figure 8, the “+Width” measurement shows that 1.227 seconds of output is high, and the “-Width” shows that 0.845 seconds is low. This leads to 2.25% and 5.62% error, respectively. This leads to 2.25% and 5.62% error, respectively.

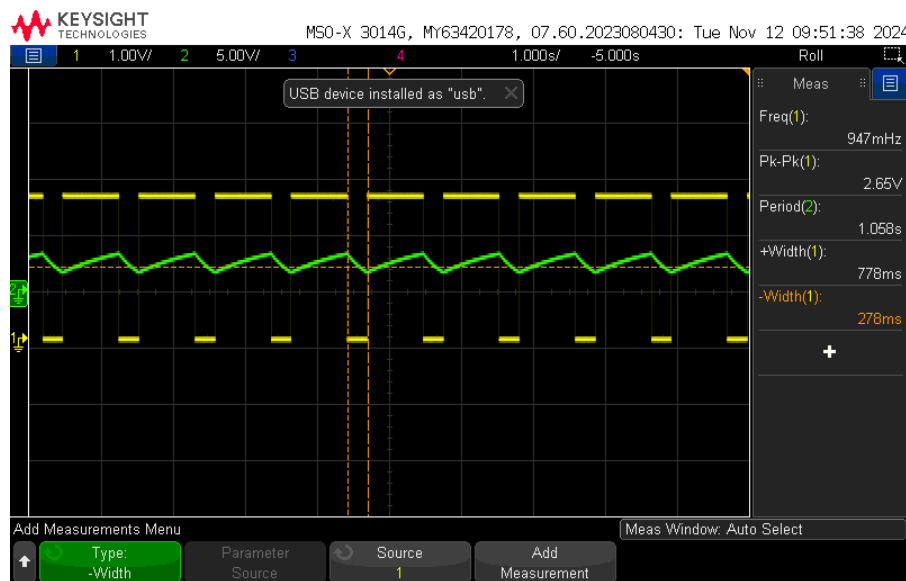


Figure 10: Astable Circuit with 60% Duty Cycle – Timer output (yellow), Capacitor Voltage (Green) - measured on oscilloscope.

Figure 9 shows the relationship between the 555 Timer output and the capacitor voltage for the Astable Circuit with 75% Duty Cycle. Recall that in the Theory section we calculated that for a 75% Duty Cycle with 1 second period, 0.75 seconds of output is supposed to be high, and 0.25 low. In Figure 9, the “+Width” measurement shows that 0.778 seconds of output is high, and the “-Width” shows that 0.278 seconds is low. This leads to 3.73% and 1.12% error, respectively.

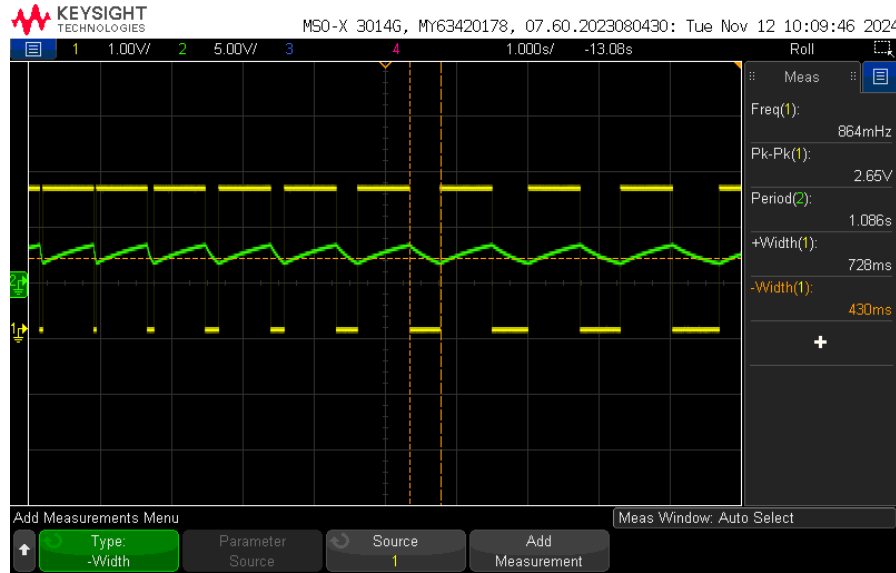


Figure 10: Astable Circuit with Potentiometer – Timer output (yellow), Capacitor Voltage (Green) - measured on oscilloscope.

Figure 10 shows the relevant voltages for the Astable circuit with a potentiometer to dynamically adjust the R1 and R2 values. Notice how the charge time of the capacitor (green line) and timer output (yellow line) do not change (are constant), while the discharge time changes by turning the potentiometer. This is because the T_{high} (Equation 3) is dependent on $R1 + R2$, which in the case of the potentiometer is always 10K (constant). However, T_{low} (Equation 4), is only dependent on R2, which does change as the potentiometer is swiped. Thus the potentiometer affects the discharge time of the capacitor and the period of the 555 Timer, but not the charge time.

The data for the Monostable vs Astable Timer Experiment is summarized fully in the following table:

Table 2: 555 Timer Configurations and their T_{high} vs T_{low} values

	T_{high} [s]	% Error (T_{high})	T_{low} [s]	% Error (T_{low})
Monostable	3.569	8.15%	Not Applicable	Not Applicable
Astable 60% Duty Cycle 2s Period	1.227	2.25%	0.845	5.63%
Astable 75% Duty Cycle 1s Period	0.778	3.73%	0.278	1.12%
Astable Potentiometer	0.728	Not Applicable	Not Applicable	Not Applicable

4.3. State Machine Experiment

The following link is a video showing the operation of the physical circuit.

https://drive.google.com/file/d/1fceu2V0tc_vlxhbfba4ugh5Kpy99OsPI/view?usp=sharing

In the video you can see as the input button is pressed the circuit cycles through the states and flashes the corresponding LED for a certain amount of time determined by the potentiometers.

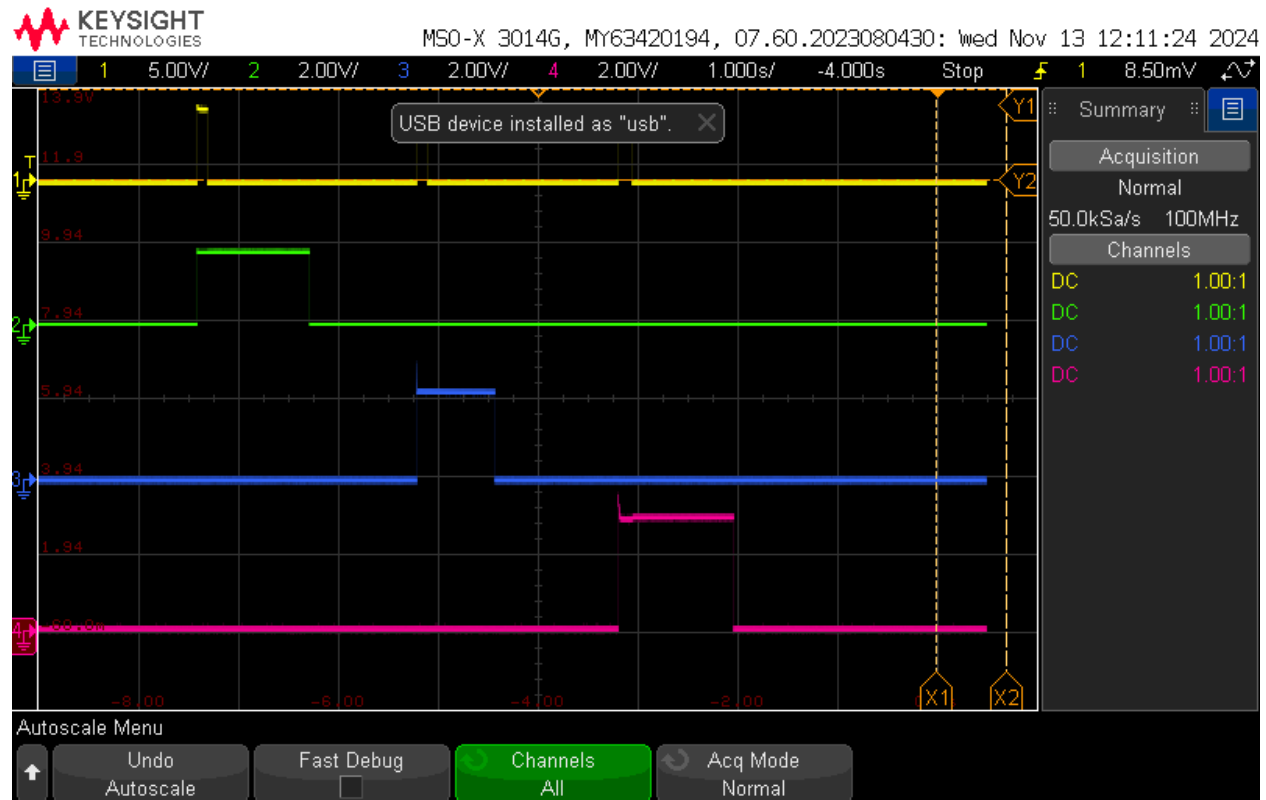


Figure 11: State Machine Circuit – Input button (yellow), State 1 Voltage (Green), State 2 Voltage (Blue), State 3 Voltage (Pink) - measured on oscilloscope.

Figure 11 shows the voltage readings for the input button, and each of the 555 Timer outputs. When the input button (yellow line) is pressed for the first time (as seen by the small moment when the voltage goes high), it signals the first Timer (green line) to output a pulse. Internally, the state is then updated to state 1 (from state 0). When the button is clicked again, it triggers only state 2 (blue line). When the button is clicked for the last time, it then triggers state 3 (pink line).

Notice how all the widths of the high pulses of the Timers are different. This is because the length of the pulse is programmable by the potentiometer in the Timer subcircuit.

5. Conclusion

5.1. Monostable vs Astable Experiment

The goal of the experiment was to compare the operation of the 555 Timer in Monostable and Astable modes and verify the timing equations discussed in the lab manual for each configuration.

For the Monostable circuit, we expected the output to be high for 3.3 seconds. Due to the percent error of 8.15% being less than 10%, we can conclude that Equation 1, which defines the output high time for a monostable circuit, is correct.

Similarly, for both the Astable circuits with 60% Duty Cycle and 75% Duty Cycle, the percent error was all less than 10%, leading me to conclude that equations 3 and 4 are also correct.

Lastly, the potentiometer configuration can be used to easily control the timing of the Astable circuit, specifically the time the output is low, and the total period. This can be used anywhere from timing traffic lights to car blinkers.

5.2. State Machine Experiment

The goal of this experiment was to create a state machine that outputs a pulse of light from a 555 Timer when the state transitions. While the design I came up with satisfied all the requirements, there are ways to improve it. Firstly, I would like the LED output of the 555 Timers to have larger period range. Currently the output is programmable by a potentiometer for each 555 Timer, however since it is a 10k potentiometer with a 100uF capacitor, the longest amount of time the LED can be on is only 1 second. If this circuit was to be used in a traffic light (which was the original goal), that would clearly not be enough time. In addition, I would like when one LED finishes its period, it triggers the state to change to the next one (just like a traffic light). Thus the user would press the start button once, the circuit will cycle through Red, Yellow, Green pulses, and then stop. Thus, it would be a Monostable traffic light.