

Joseph Abroquah  
ECEN 5730 – Fall 2025

## Lab 2 Report: SSB-build 555 Timer

### Objective

The objective of this lab is to gain practical experience in circuit prototyping by building a functional circuit on a solderless breadboard (SSB). This exercise includes performing oscilloscope measurements, finding components, interpreting datasheets, and making some design decisions while balancing tradeoffs between component selection and overall performance.

### Plan of record

- A SSB is powered by a 5V rail on the Arduino board
- A 555 used in an astable vibrator circuit to produce a square wave with a frequency of 500 Hz and 60% duty cycle
- Get the rise time and falling time of the waveform
- Connect four LEDs of the same color with series resistors 50, 300, 1k, and 10k Ohms to the output of the 555 and
- Measure the open circuit voltage at the output of the 555
- Measure the voltage and current through the 50, 300, 1k and 10k Ohm LED
- Then compare the performance of two timers: NE555P and TLC55IP

### Component Listing

The specifications of the components used in this lab session are listed in the table below.

Component	Specification	Quantity	Component	Specification	Quantity
Resistor	1kΩ	1	TLC555IP	-	1
	100 Ω	2	NE555P	-	1
Capacitor	0.01 uF	1	LEDs	Red	4
	1 uF	1	Arduino Uno	1	1

### Circuit Diagram

A sketch of the major components of the circuit is shown in Fig.1.(a). The components on the SBB as illustrated in Fig.1.(c) are configured according to the astable vibrator circuit shown in Fig.1.(b).

### Theoretical analysis

The duty cycle and frequency of operation are controlled by the resistors and capacitors according to equation 1 as specified in the datasheet. The capacitance has no effect on the duty cycle but affects the frequency. A higher capacitance reduces the frequency

$$DC = \frac{R_A}{R_A + 2R_B}; f = \frac{1.44}{(R_A + 2R_B)C} \quad (1)$$

To obtain 50% duty cycle and 500 Hz, we choose  $C$ ,  $R_A$  and  $R_B$  to be  $1\mu F$ ,  $50 \Omega$   $1k \Omega$  respectively based on the availability of the components. To obtain  $50 \Omega$  resistance, two  $100 \Omega$  resistors are connected in parallel. The calculated duty cycle and frequency are 48% and 704 Hz.

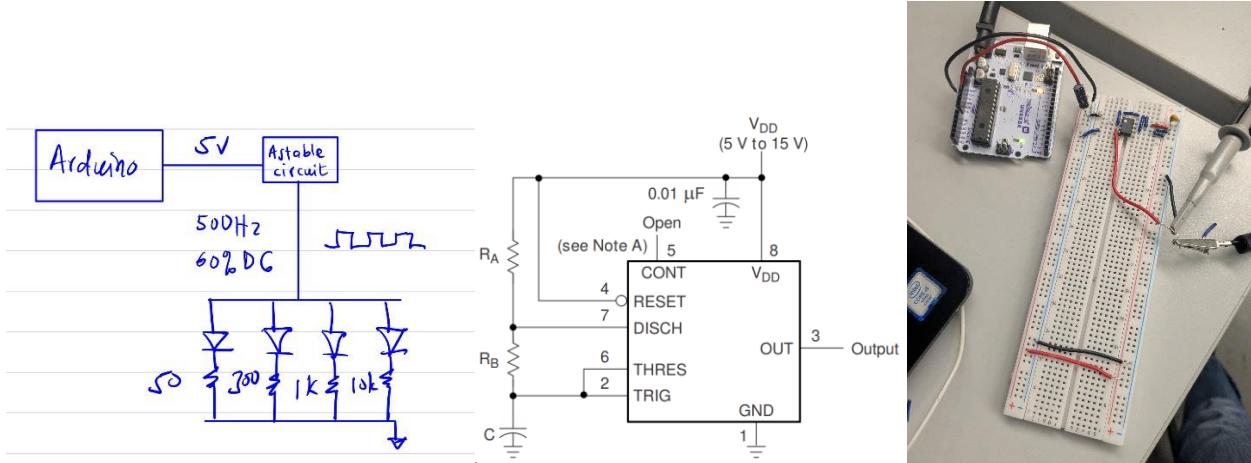


Fig. 1. Napkin sketch (a), Astable vibrator circuit (b), Photo of breadboard circuit (c)

## Measurement and Discussion

Fig. 2. shows the frequency and duty cycle of the 555 timers. The frequency is 453 Hz and 361 Hz for the TLC555 and NE555 timer respectively. The duty cycle is 64.6% and 41.81% respectively. We see that the measured values deviate from the calculated values. This could be attributed to tolerances of the components. The rise times are 10ns and 100ns respectively as illustrated in Fig. 3. This shows that the TLC555 has a faster rise time than the NE555. The peak to peak voltage across the  $50 \Omega$  branch is measured to be about 2.5 V. This means that the peak voltage is 1.25V. The root mean square voltage for the square waveform is then 1.25V.



Fig. 2. Frequency and duty cycle of the TLC555 (left) and NE555 (right) 555 timer

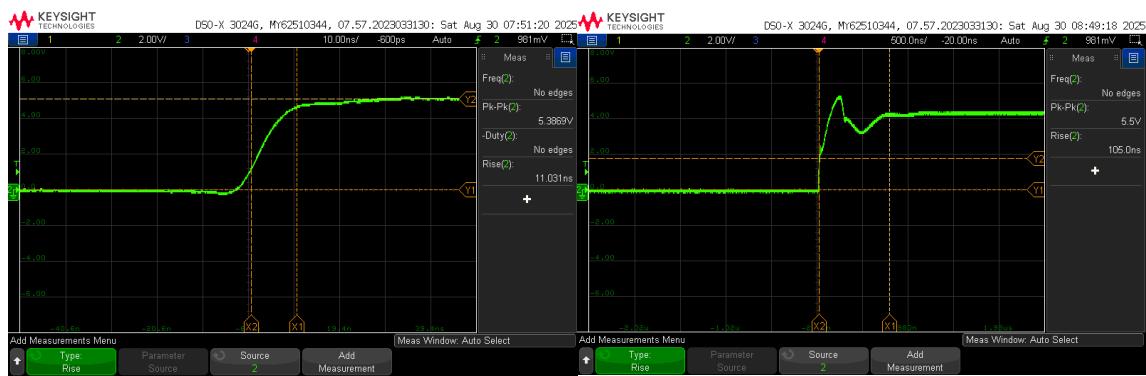


Fig. 3. Rise times of the TLC555 (left) and NE555 (right) 555 timer



Fig. 4. Voltage across the  $50\Omega$  for the TLC555 (left) and NE555 (right) 555 timer

The current through the limiting resistors are given in the table below. From Ohm's law, the higher the resistance, the lower the current through the branch. This is seen in Fig.5. through the brightness level of the LEDs.

Current Limiting Resistor [ $\Omega$ ]	Current Through Branch [mA]
50	25
300	4.2
1k	1.25
10k	0.125

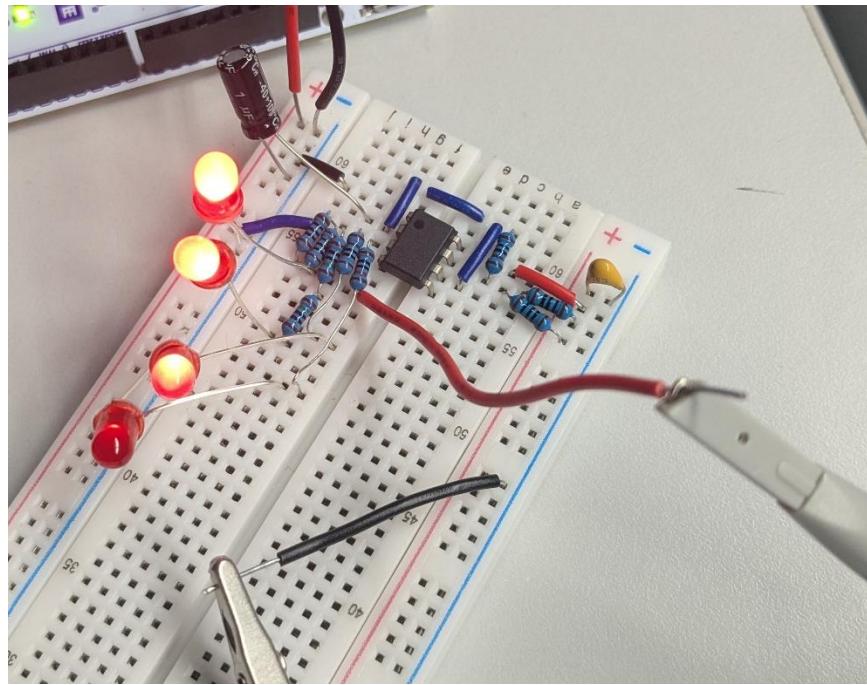


Fig. 5. Brightness of the four LEDs

### Conclusion

In this lab, I learned:

- How to use an oscilloscope and configure an SBB prototype
- How to find component information in datasheets
- How to understand the internal circuitry of a 555 timer and its various applications
- That the NE555 timer is suitable for circuits with long rise and fall times (on the order of 100ns) while the TLC555 is suitable for circuits with short rise and fall times (on the order of 10ns)