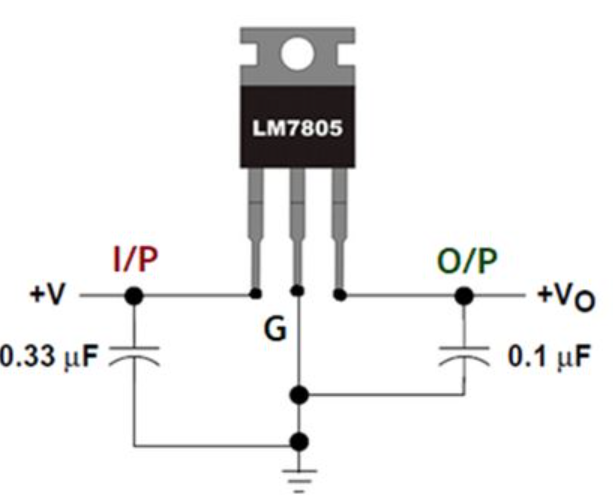
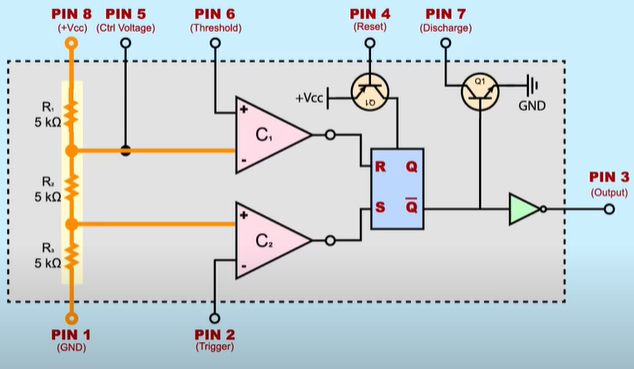
1. We can use a MOSFET (either PMOS or NMOS) to solve this issue. The main circuit can be connected between the Drain and Source terminals, and the Power Supply between the Gate and Source terminals. If the Power Supply is working properly, then the MOSFET would just act as a closed switch with very low impedance. In case of a reverse current, the Gate-Source polarity would become reversed and thus, the MOSFET would behave like an open switch, breaking off the circuit between the Drain and Source terminals.
2. A potential divider will not be the best solution, as its output might fluctuate whereas we want a steady output for the microcontroller. Also, a lot of heat would be dissipated in the resistors. We can try implementing a voltage regulator IC, such as LM7805.



The output voltage would be a steady waveform, with an approximate value of 5V.

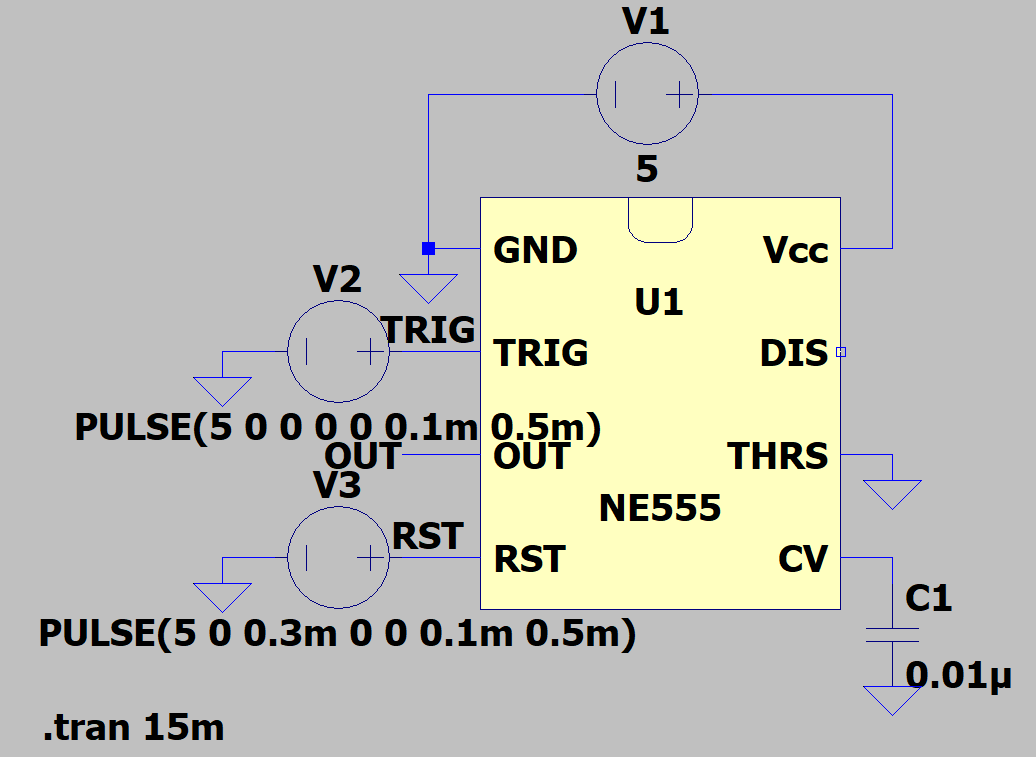
1. **Timer IC**

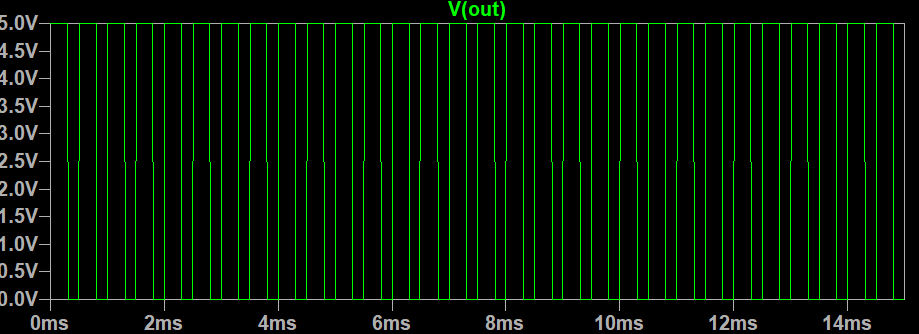


It would be best to understand each component of the IC by looking at different modes of operation of the timer.

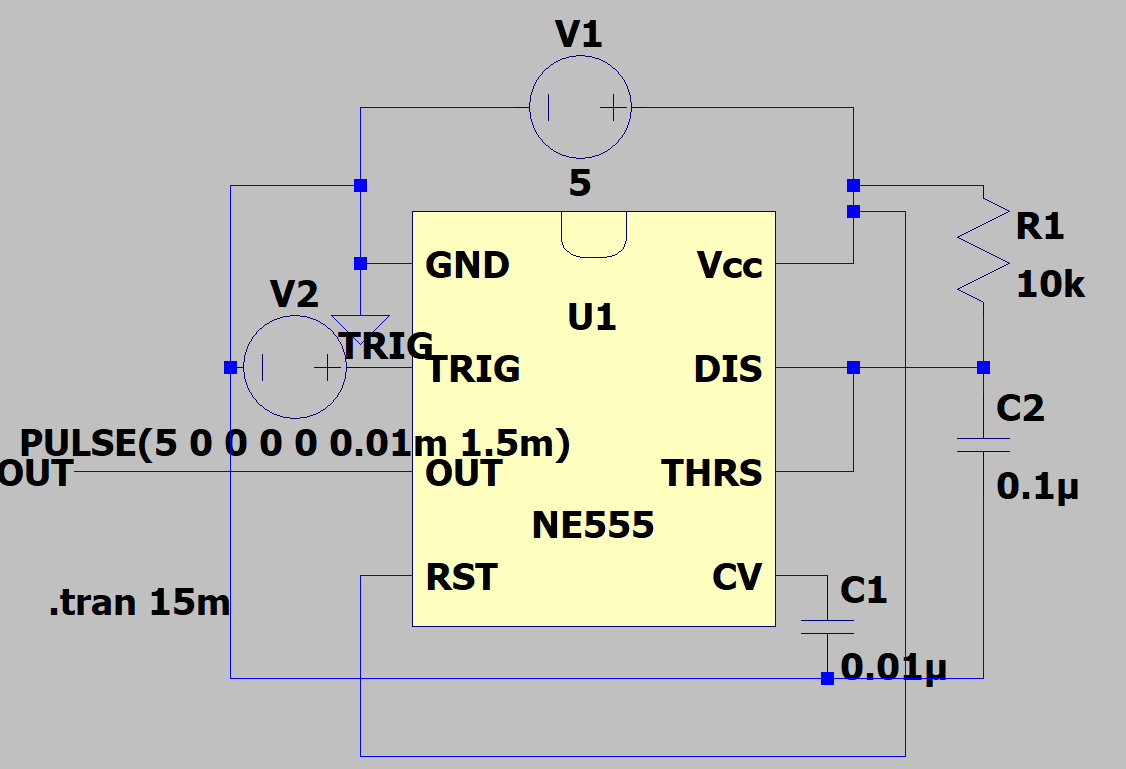
*Bistable mode*

Since THRS = 0, R = 0 always. When TRIG = RST = 1, S = 0 and the flip-flop is on hold (say OUT was LOW initially). At first, TRIG = 0 (since RST has a delay), making S = 1. This makes OUT = 1. Then, when TRIG goes back to LOW, S = 0 and the flip-flop is back on hold. Then, RST goes LOW. This makes OUT = 0. When RST becomes HIGH again, OUT stays LOW as flip-flop is on hold.

**



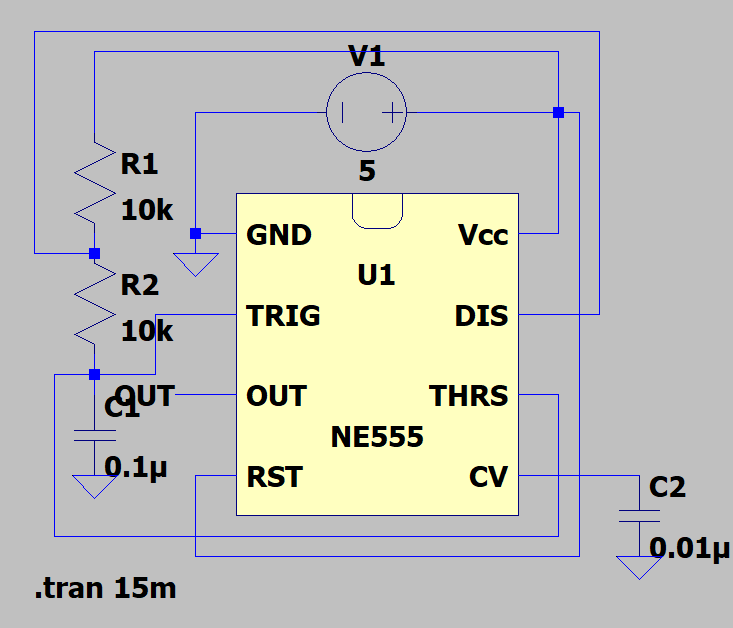
*Monostable mode*





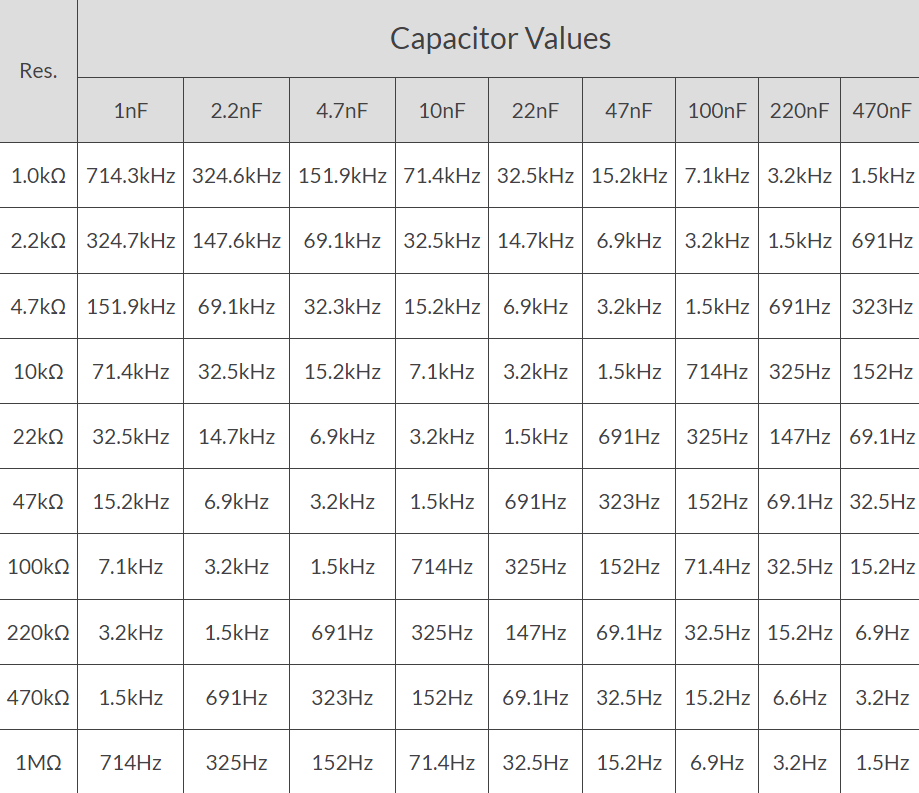
At first, C2 has no charge. So, THRS = VCC and so, R = 1. TRIG is initially HIGH and so, S = 0. This makes OUT = 0 and the BJT is ON, connecting THRS to GND. Thus, R = 0. Flip-flop is on hold. Then, TRIG is made LOW for a split second. This makes S = 1 and so, OUT = 1. The BJT is OFF and thus, C2 charges up. Until THRS reaches (2/3)VCC (at t = 1.1R1C2), R = 0, making the flip-flop at hold (since S is back to 0). Then R goes HIGH and the cycle continues.

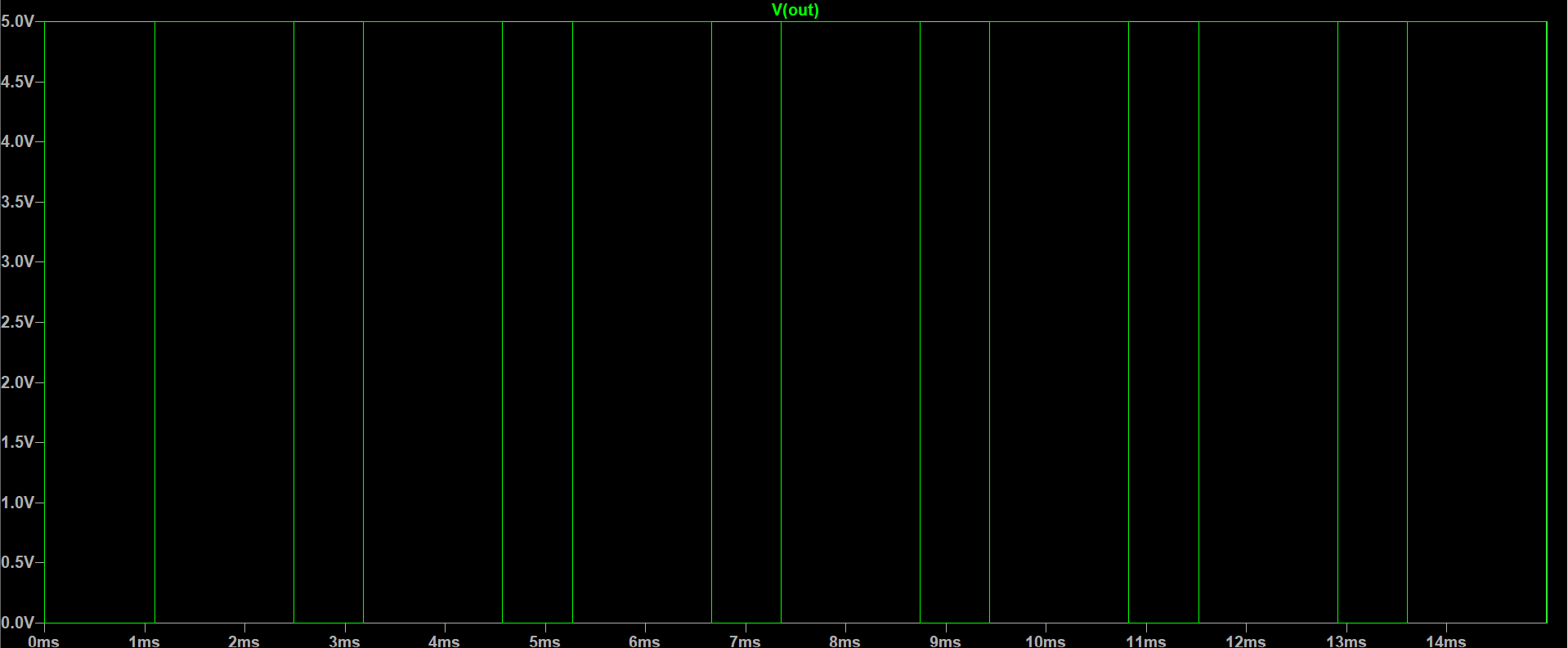
*Astable mode*

**

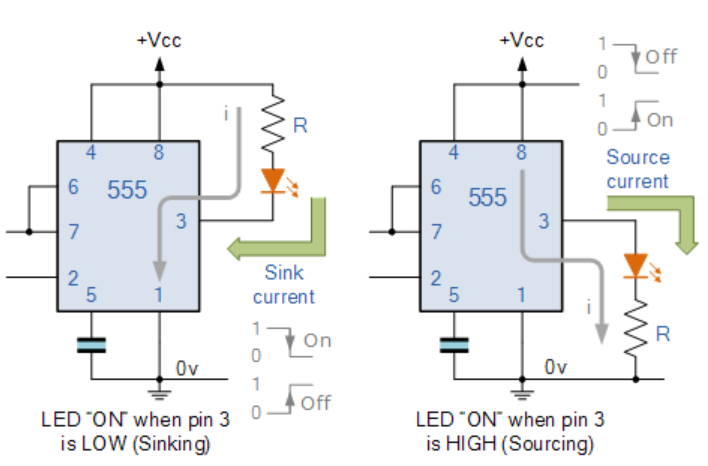
At first, C1 has no charge. So, THRS = TRIG = 0. So, R = 0 and S = 1. So, OUT = 1 and BJT is OFF. Thus, C1 charges up. When C1 charges to VCC/3, S = 0 and so, flip-flop is on hold. Then, C1 charges to (2/3)VCC and so, R goes HIGH. This makes OUT = 0 and BJT is ON. Hence, C1 discharges. When C1 discharges below (2/3)VCC, R = 0 and flip-flop is on hold. Then, as C1 discharges below VCC/3, S goes HIGH and the cycle continues.

The output waveform’s frequency is inversely proportional to the resistances and capacitance. Thus, lower their values are, higher the frequency would be. Here’s an idea about the values.



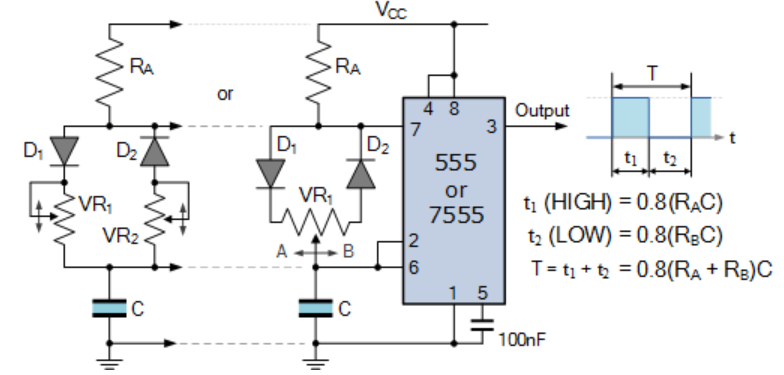


*Sourcing/Sinking current capabilities*

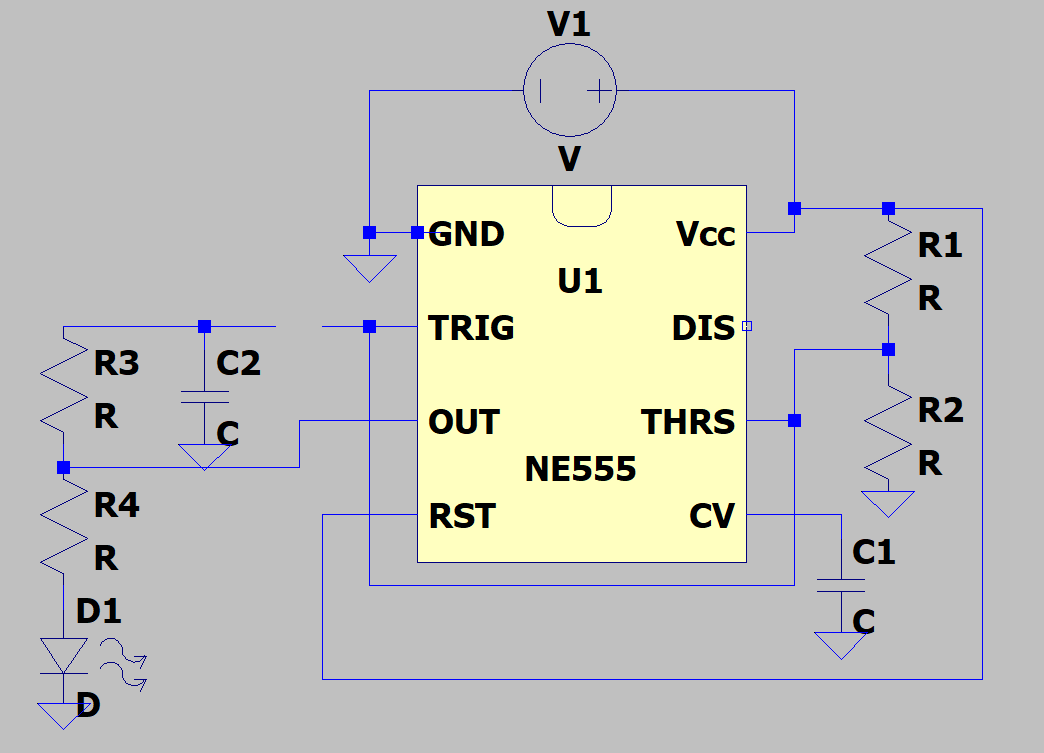


Depending on the logic of OUT, we can use it as a current source, or sink. For example, in the first circuit, the LED lights up when OUT is LOW and in the second one, it lights up when OUT is HIGH. The current passing through is enough to power other digital circuits. For more heavy analog loads like motors, we can amplify this current via transistors.

*Independent 555 oscillator*



*555 timer latching circuit*



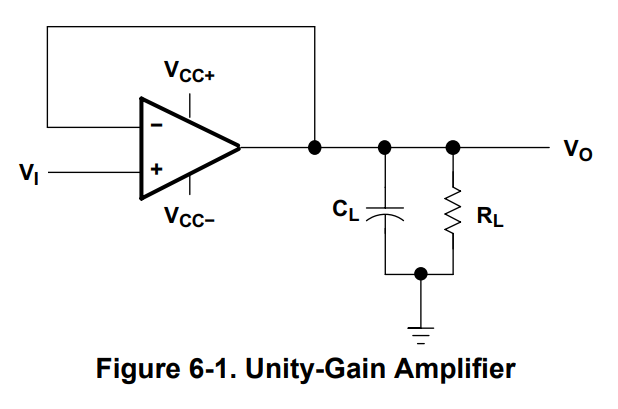
The open circuit at trigger pin is controlled by a push button. When it is open at first, then THRS = TRIG = VCC/2. So, R = 1 and S = 0, and OUT = 0. Thus, the LED is OFF. When we press the button, TRIG = THRS are both pulled to GND, while C2 charges. Thus S = 1 and R = 0. This makes the LED ON as OUT = 0. Even if the voltage across C2 crosses VCC/3, S becomes LOW and flip-flop goes on hold. Thus, the LED keeps on blinking. When C2 is fully charged and we press the button again (breaking connections), TRIG becomes HIGH and so, LED turns OFF as S goes LOW.

*TLC555 vs NE555*

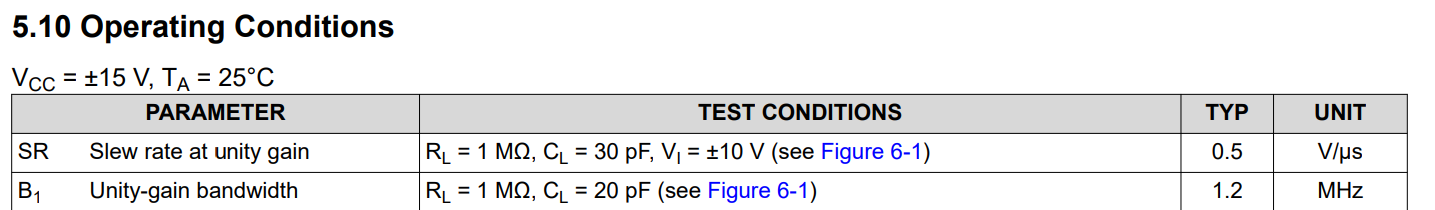
Apart from temperature and supply voltage range usages, a TLC555 can support much higher frequency than a NE555. The former can handle smaller timing capacitors than the NE555 because of its high input impedance. As a result, time delays and oscillations may be more precisely controlled.

1. We can connect the error signals onto the TRIG pin (via an AND operation, since we want to take both the errors into account) of the 555 timer IC. Since they’re active LOW, any error would make OUT go HIGH. Then, we can connect the OUT pin to the relay and by taking into account our friend’s advice, we can amplify the current by using transistors.

2) The operating circuit of the Op-Amp should’ve been something like this-



The parameters RL and CL can take the values as shown.



Since the concept of infinite resistance in an Op-Amp is ideal, there is a high, but finite resistance between the input terminals. Hence, there’s always some current flowing. The capacitor solves for an external lag compensation, which helps stabilise the buffer.