Lesson 9 Ne555

Introduction

If you ask anyone in the know to rank the most commonly and widely used IC, the famous 555 time base IC would certainly be at the top of the list. The 555 – a mixed circuit composed of analog and digital circuits – integrates analogue and logical functions into an independent IC, and hence tremendously expands the application range of analog integrated circuits. The 555 is widely used in various timers, pulse generators, and oscillators.

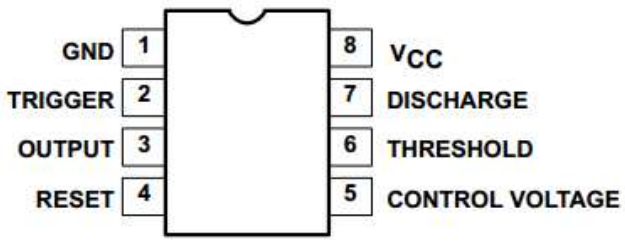
Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * NE555
- 2 * 104 ceramic capacitor
- 1 * Potentiometer (50KΩ)
- 1 * Resistor (10KΩ)
- 1 * USB cable
- Jumper wires
- 1 * T-Extension Board
- 1 * 40-Pin GPIO Cable

Principle

The 555 IC was originally used as a timer, hence the name 555 time base circuit. It is now widely used in various electronic products because of its reliability, convenience, and low price. The 555 is a complex hybrid circuit with dozens of components such as a divider, comparator, basic R-S trigger, discharge tube, and buffer.

555 chip pins are introduced as follows:

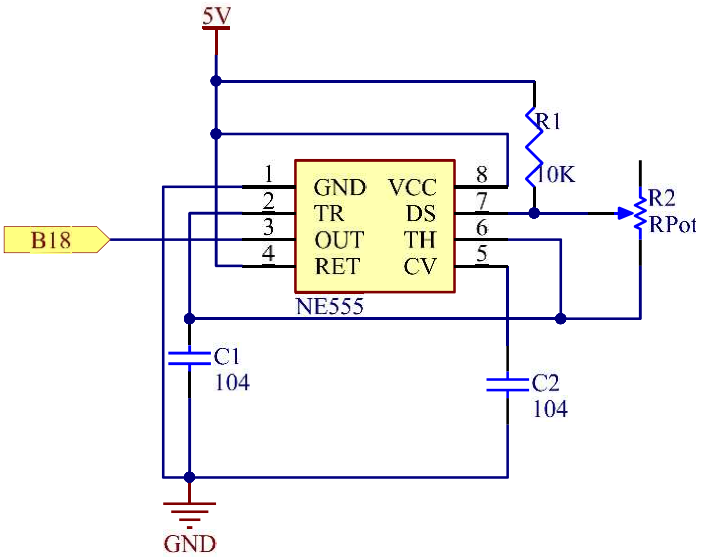


As shown in the picture, the 555 IC is dual in-line with the 8-pin package. Thus:

- Pin 1 (GND): the ground;
- Pin 2 (TRIGGER): the input of lower comparator;
- Pin 3 (OUTPUT): having two states of 0 and 1 decided by the input electrical level;

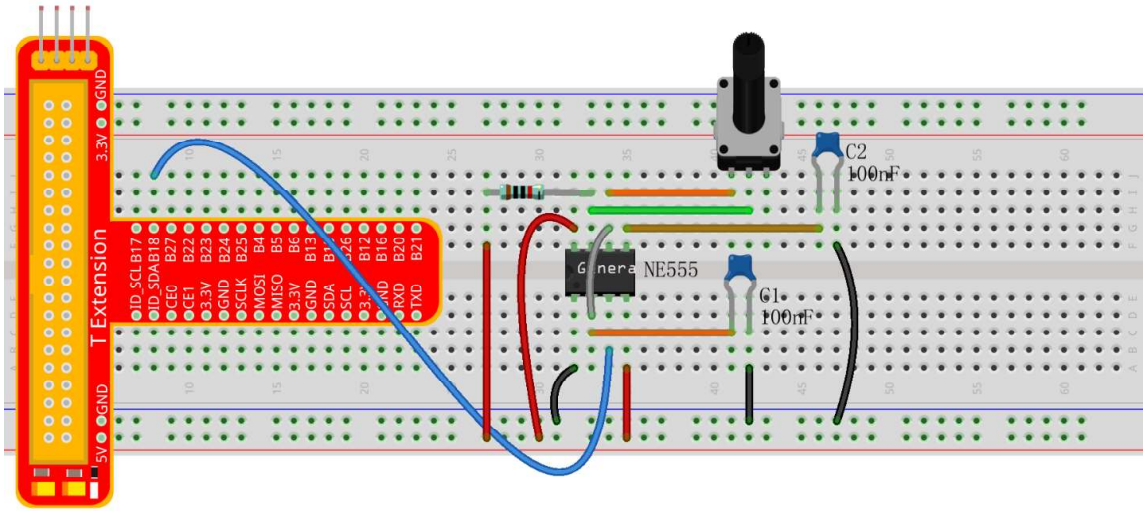
- Pin 4 (RESET): output low level when supplied a low one;
- Pin 5 (CONTROL VOLTAGE): changing the upper and lower level trigger values;
- Pin 6 (THRESHOLD): the input of upper comparator;
- Pin 7 (DISCHARGE): having two states of suspension and ground connection also decided by input, and the output of the internal discharge tube;
- Pin 8 (VCC): the power supply;

The schematic diagram



Experimental Procedures

Step 1: Build the circuit



For C language users:

Step 2: Go to the folder of the code.

```
cd /home/pi/SunFounder_Super_Kit_V3.0_for_Raspberry_Pi/C
```

Step 3: Compile

```
make 09_ne555
```

Step 4: Run the executable file above.

```
sudo ./09_ne555
```

Code Explanation

```
static volatile int globalCounter = 0 ; // a static integer variable to store the pulse count
void exInt0_ISR(void) {  //GPIO0 interrupt service routine
    ++globalCounter;
}
wiringPiISR(Pin0, INT_EDGE_FALLING, &exInt0_ISR); // set an interrupt here and the signal is falling edge for Pin 0. When the interrupt happens, execute the function exInt0_ISR(), and the pulse count will add 1.

while(1){  // if no interrupt happens, the pulse count will stay and just print it.
    printf("Current pulse number is : %d\n", globalCounter);
}
```

For Python users:

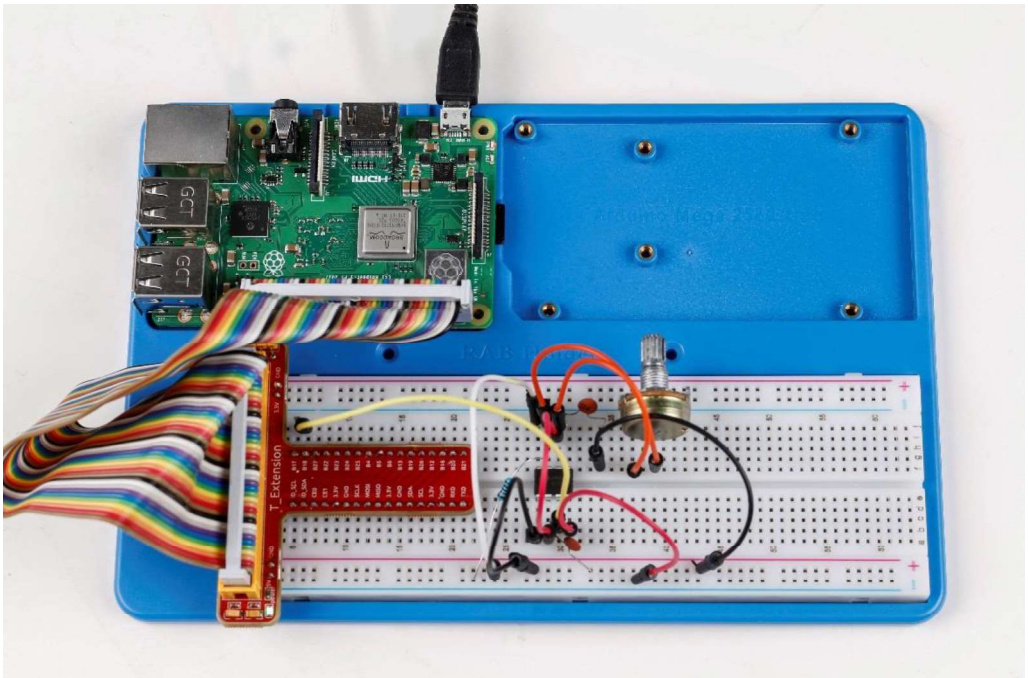
Step 2: Get into the folder of the code.

```
cd /home/pi/SunFounder_Super_Kit_V3.0_for_Raspberry_Pi/Python
```

Step 3: Run

```
sudo python 09_ne555.py
```

Now you can see the number of square waves printed. Spin the potentiometer and the value will decrease or increase.



Code Explanation

```
g_count = 0    #  a global variable used to store the pulse count
def count(ev=None):    # Define a function to be run when an interrupt happens
  global g_count  # this function will change the value of the global variable g_count, thus here we add global before it.
  g_count += 1

GPIO.add_event_detect(SigPin, GPIO.RISING, callback=count) # set an interrupt here and the interrupt signal is a rising edge for Pin Sig. It will run the function count() accordingly
while True:         # wait for the interrupt
  print 'g_count = %d' % g_count  # print the information
  time.sleep(0.001)
```