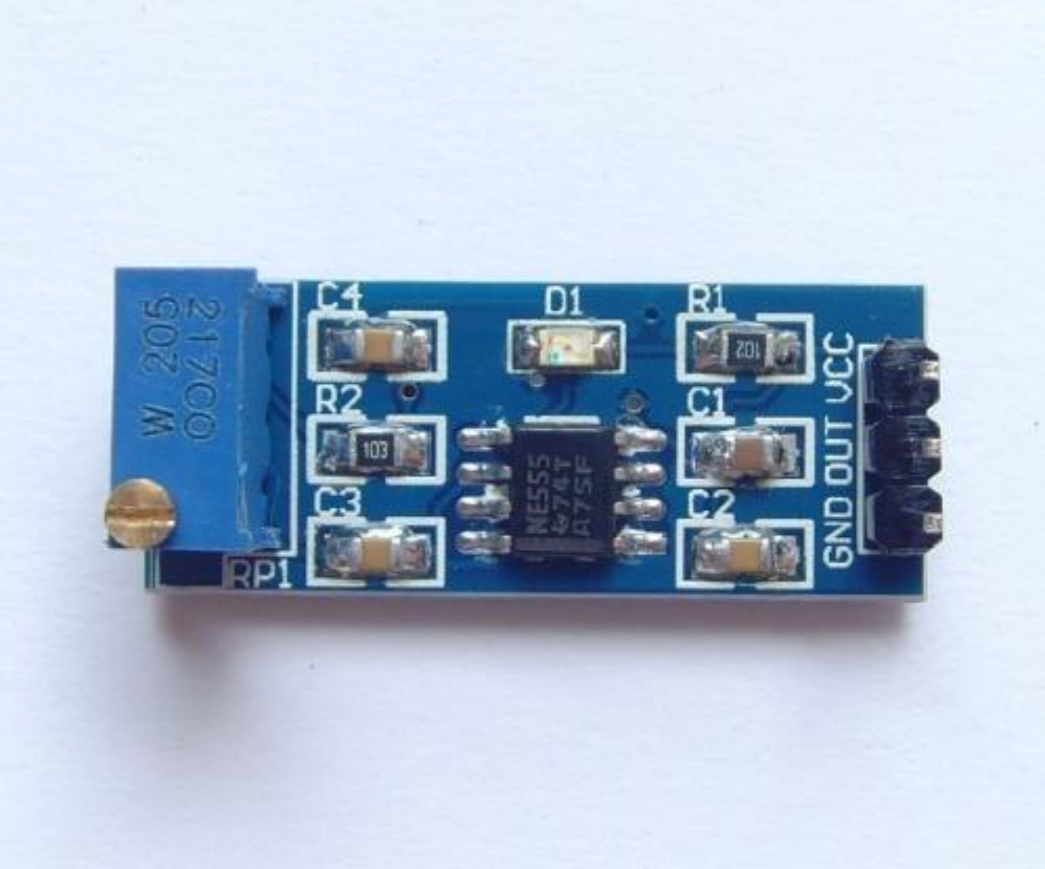
NE555 Frequency Adjustable Pulse Generator Module



General Description:

The 555 timer IC is an integrated circuit (chip) used in a variety of timer, pulse generation, and oscillator applications. The 555 can be used to provide time delays, as an oscillator, and as a flip-flop element. Derivatives provide up to four timing circuits in one package. Introduced in 1971 by Signetics, the 555 is still in widespread use due to its ease of use, low price, and stability. It is now made by many companies in the original bipolar and also in low-power CMOS types.

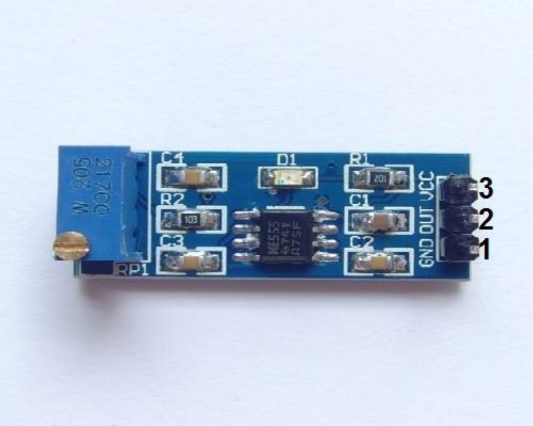
This module is use as a square-wave signal generator. It generates square wave signals used for experimental development. It also generates square wave signal driving motor drive. With adjustable pulses generated for use by the MCU and adjustable pulse control related circuits.

Specifications:

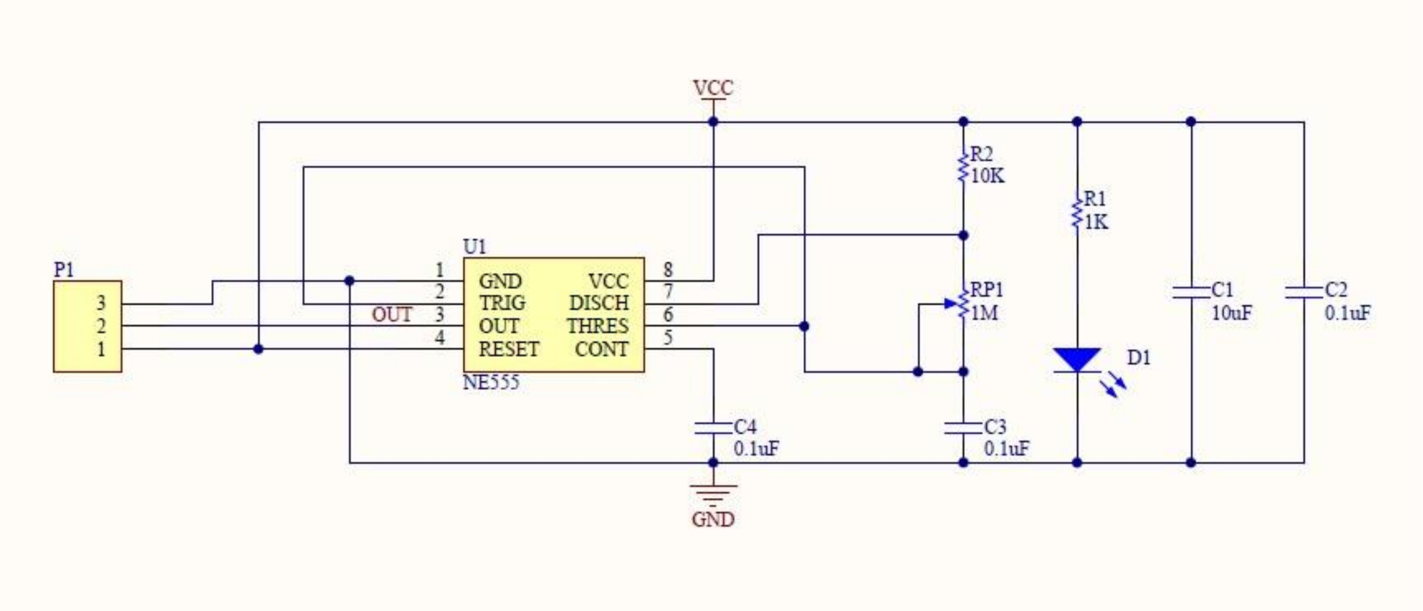
- Onboard NE555 chip.
- Single-channel signal output, the output square wave duty ratio is about fifty percent.
- Onboard adjustable resistance, adjustable resistance can control the output frequency.
- Onboard power light.
- Working voltage: 5 ~ 12 v.
- The size of the board: 29 (mm) x12 (mm).
- Minimum - 50% duty cycle, 3.7 Hz
- Maximum - 98% duty cycle, 1.3KHz

Pin Configuration:

1. Ground
2. Output
3. VCC



Schematic Diagram:



How to test:

1. Connect your Arduino microcontroller to the computer.
2. Connect the VCC pin of your module to the to the 5V pin of your Arduino.
3. Connect the GND pin of your module to the GND pin of your Arduino.
4. Connect the Output pin of your module to the A0 pin of your Arduino.
5. Enter this program to your Arduino Integrated Development Environment (IDE):

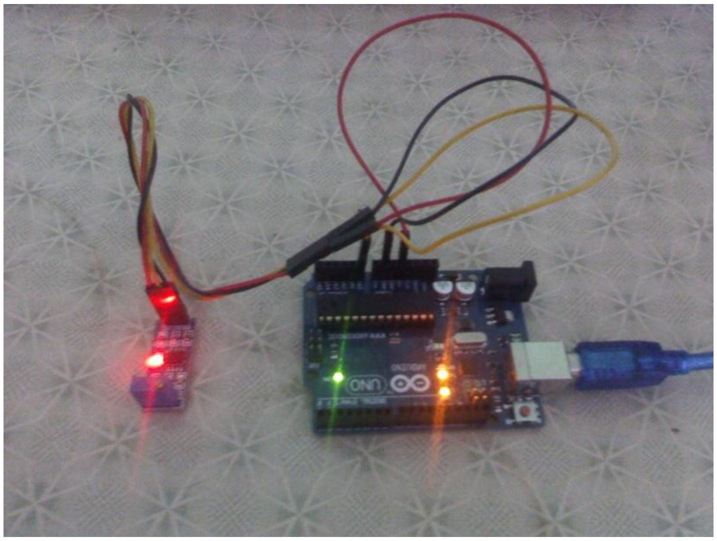
```
void setup()
{
        Serial.begin(9600);
}

void loop()
{
        analogRead(A0); Serial.println(analogRead(A0));
        delay(100);
}
```

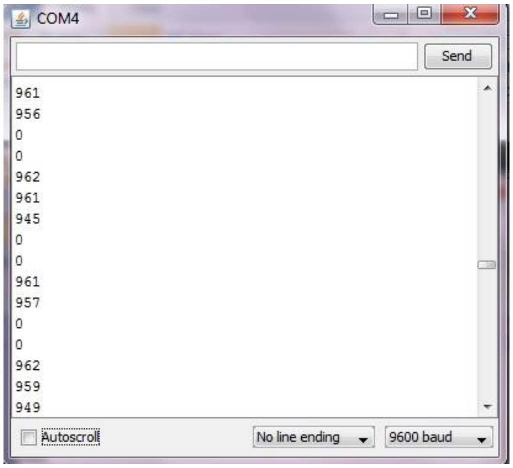
6. Click the Upload Button
7. Lastly, click the Serial Monitor button.

Testing Results:

The set-up of your module and Arduino look like this:

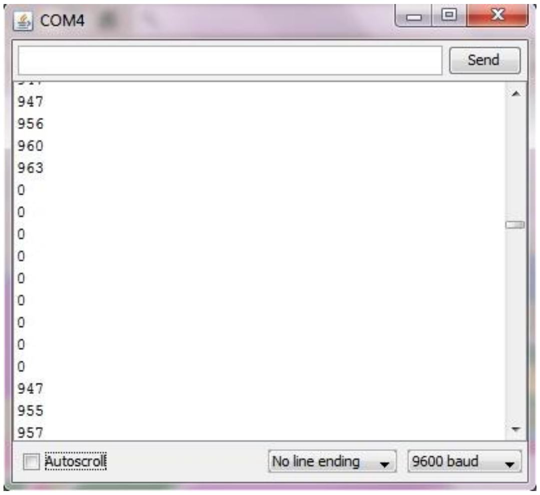


When the delay is 100ms the output frequency goes like this:



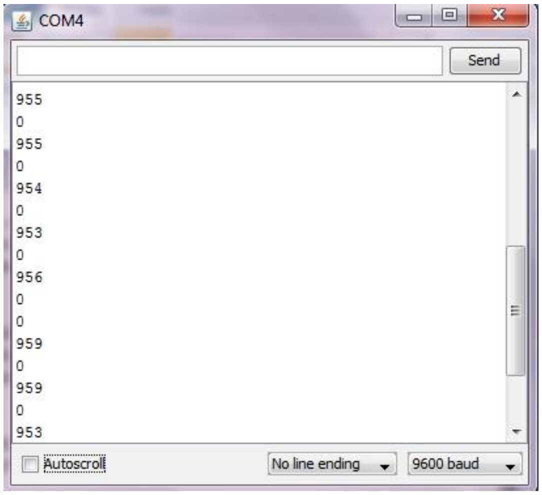
```
961
956
0
0
962
961
945
0
0
961
957
0
0
962
959
949
```

When the delay is 1ms the output frequency goes like this:



```
947
956
960
963
0
0
0
0
0
0
0
0
0
947
955
957
```

When the delay is 1s the output frequency goes like this:



```
955
0
955
0
954
0
953
0
956
0
0
959
0
959
0
953
```