

!Note:

The maximum continuous input and output voltage of the Raspberry Pi's GPIO pin is 3.3V. Do not connect it directly with other electronic components, otherwise it will damage the Raspberry Pi.

Step 1: Create and open pwm.py file

`nano pwm.py`

Step 2: Writing code

```
import wiringpi
```

```
OUTPUT = 1
```

```
PIN_TO_PWM = 1
```

```
wiringpi.wiringPiSetup()
```

```
wiringpi.pinMode(PIN_TO_PWM, OUTPUT)
```

```
wiringpi.softPwmCreate(PIN_TO_PWM, 0, 100)
```

```
wiringpi.softPwmWrite(PIN_TO_PWM, 50)
```

After writing, press **Ctrl + X** to exit this file.

The system will prompt you whether you need to save, press **Y** to save and exit.

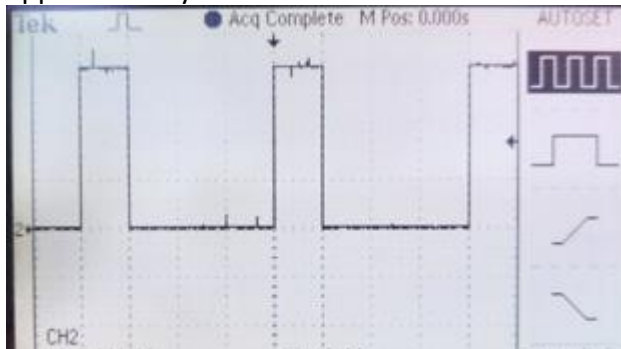
Step 3: Run this code

`python3 pwm.py`

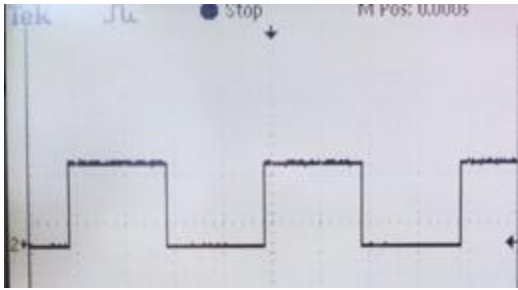
Step 4: Use an oscilloscope to measure GPIO1 of the Raspberry Pi.

Change the value of Value to 25, 50, 75, we can find that the waveform on the oscilloscope also changes accordingly

- When Value is 25, the oscilloscope detects that the PWM duty cycle of GPIO1 output is approximately 25%. As shown below.



- When Value is 50, the oscilloscope detects that the PWM duty cycle of GPIO1 output is approximately 50%. As shown below.



- When Value is 75, the oscilloscope detects that the PWM duty cycle of GPIO1 output is approximately 75%. As shown below.

