5.Output PWM

1. Create and open pwm.py file

```
nano pwm.py
```

2. Write program

```
import wiringpi

OUTPUT = 1
PIN_TO_PWM = 1      #Define PIN_TO_PWM as pin 1
wiringpi.wiringPiSetup()       #Set the GPIO number to wPi mode
wiringpi.pinMode(PIN_TO_PWM,OUTPUT)   #Set PIN_TO_PWM to OUTPUT output mode
wiringpi.softPwmCreate(PIN_TO_PWM,0,100)     #Set the PWM output pin to
PIN_TO_PWM, the PWM range is 0-100
wiringpi.softPwmWrite(PIN_TO_PWM,50)        #Output PWM with a duty cycle of 50
from PIN_TO_PWM
```

After writing is completed, press the shortcut key to exit

"Ctrl+X"

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

"Y"

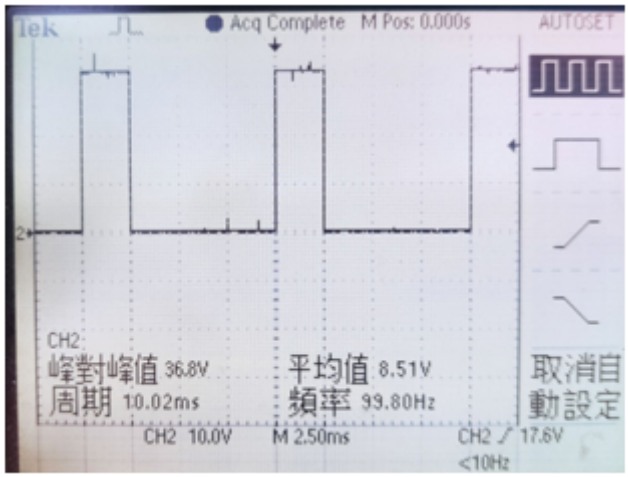
3. Run program

```
python3 pwm.py
```

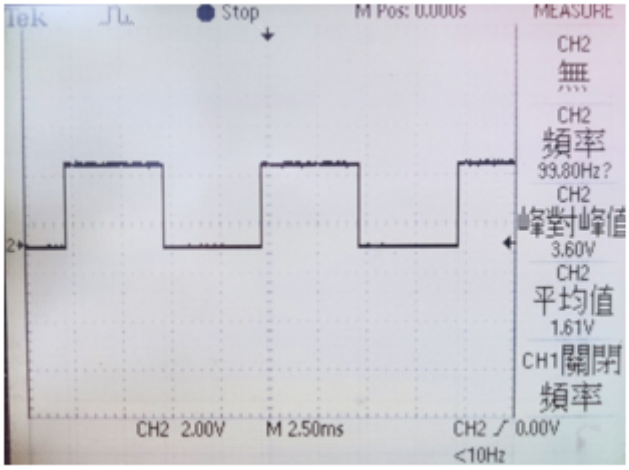
4. Use an oscilloscope to measure the GPIO1 of the Raspberry Pi

Change the Value values to 25, 50, and 75 respectively, and you can find that the waveform on the oscilloscope also changes accordingly.

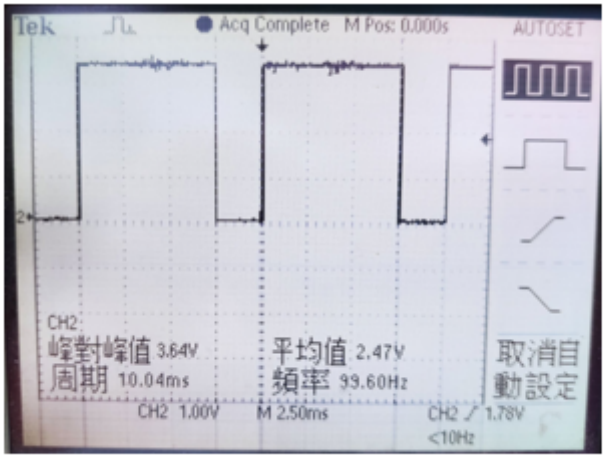
- When Value is 25, the oscilloscope detects that the PWM duty cycle of GPIO1 output is approximately 25%.



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



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



The PWM duty cycle of the pin output changes as Value changes