!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.c file

```
nano pwm.c
```

Step 2: Writing code

```
#include<stdio.h>
#include<wiringPi.h>            //Import wiringPi.h library
#include<softPwm.h>             //Import softPwm library
#define PWM_Pin 1               //Define PWM_Pin as pin1 of wPi, corresponding to GPIO1

int main()
{
        printf("This is an experiment of output PWM\n");
        wiringPiSetup();                        //Initialize wiringPi
        softPwmCreate(PWM_Pin,0,100);   //The current pwmRange is 100 and the frequency is 100Hz. If pwmRange is 50, the frequency is 200, and if pwmRange is 2, the frequency is 5000
        softPwmWrite(PWM_Pin,50);               //Duty cycle = value/pwmRange, current duty cycle = 50/100 = 50%
        while(1)
        {

        }
        return 0;
}
```

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: Compile this .c file.

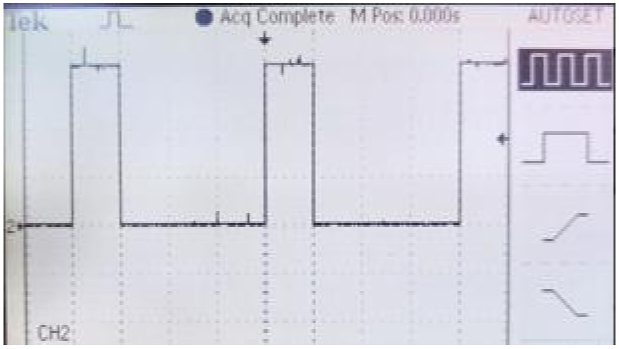
```
gcc pwm.c -o pwm -lwiringPi
```

Step 4: Run this code

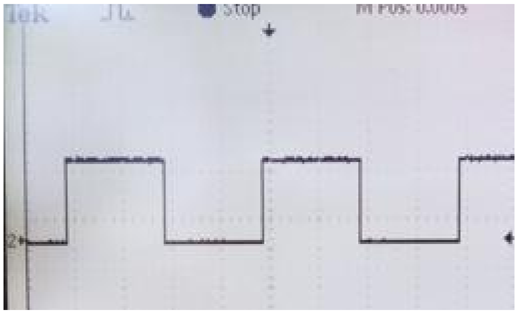
```
./pwm
```

Step 5: 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

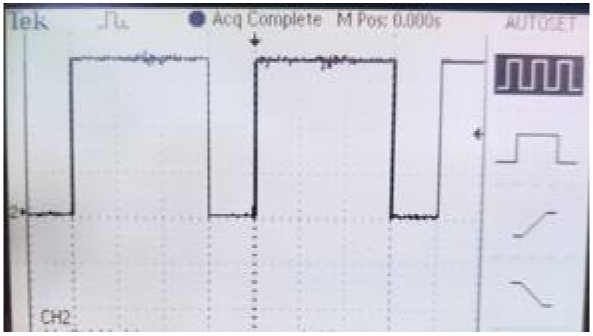
- When Value is 25, the oscilloscope detects that the PWM duty cycle of GPIO1 output is approximately 25%. As shown below.



- 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.


Tek
Acq Complete
M Pos: 0.000s
AUTOSET
2
CH2