

!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 read-gpio.c file

`nano read_gpio.c`

Step 2: Writing code

```
#include<stdio.h>
#include<wiringPi.h>           //Import wiringPi.h library

#define Output_Pin 0 //Define Output_Pin as pin 0 of wPi, corresponding to GP
IO0
#define Input_Pin 1 //Define Input_Pin as pin1 of wPi, corresponding to G
PIO1
int main()
{
    printf("This is an example of a raspberry PI pin reading\n");
    wiringPiSetup();           //Initialize wiringPi
    pinMode(Output_Pin,OUTPUT); //Set Output_Pin to output mode
    pinMode(Input_Pin,INPUT);  //Set Output_Pin to input mode

    printf("Set Output_Pin : H\n");
    digitalWrite(Output_Pin,HIGH); //set Output_Pin output high level
    delay(200);                     //delay 200ms
    printf("Input_Pin : %d\n",digitalRead(Input_Pin)); //Print Input_Pin value
    delay(2000);                     //delay 2000ms

    printf("Set Output_Pin: L\n");
    digitalWrite(Output_Pin,LOW); //set Output_Pin output low level
    delay(200);
    printf("Input_Pin : %d\n",digitalRead(Input_Pin)); //Print Input_Pin value
    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 read_gpio.c -o read_gpio -lwiringPi
```

Step 4: Connect GPIO0 and GPIO1 by Dupont line.



Step 5: Run this code

```
./read_gpio
```

```
pi@raspberrypi:~/work/example/C $ gcc read_gpio.c -o read_gpio -lwiringPi
pi@raspberrypi:~/work/example/C $ ./read_gpio
This is an example of a raspberry PI pin reading
Set Output_Pin : H
Input_Pin : 1
Set Output_Pin: L
Input Pin : 0
```

We can see that the InputPin pin changes with the output of OutputPin.