

Vibration Sensor Module (000x0000 Article Number) (TS2145)



Product Details

The TelePort vibration sensor is highly sensitive and omnidirectional, with a SW-420 component. When the sensor is stable, the circuit will be in OFF state and low levels will be output; when it vibrates, the circuit will be in ON state temporarily and high levels will be output.



You can turn the potentiometer knob on the sensor to adjust the sensitivity.

Features and Benefits

- Compatible with RJ11 6P6C OKdo TelePort Control boards and expansion shields.
- Built with the 801S vibration sensor at its core and a potentiometer for adjustable sensitivity.
- Detect vibration of mechanical equipment.

Technical Specifications

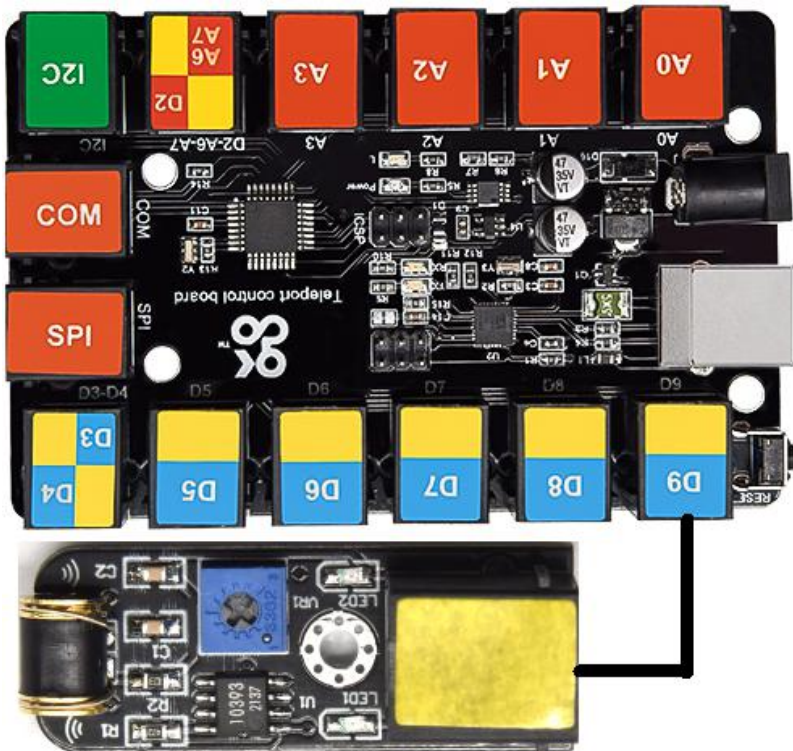
Sensor type	Digital input
Working voltage	3.3V-5V
Dimensions	46mm*16.7mm*17.6mm
Weight	6.2g

Applications

- Car, bicycle, motorcycle burglar alarm
- Game control
- Vibration detection

This module is compatible with the TS2180-Raspberry Pi shield, the TS2179-Micro:bit shield and the TS2178-TelePort main board.

➤ Arduino Application



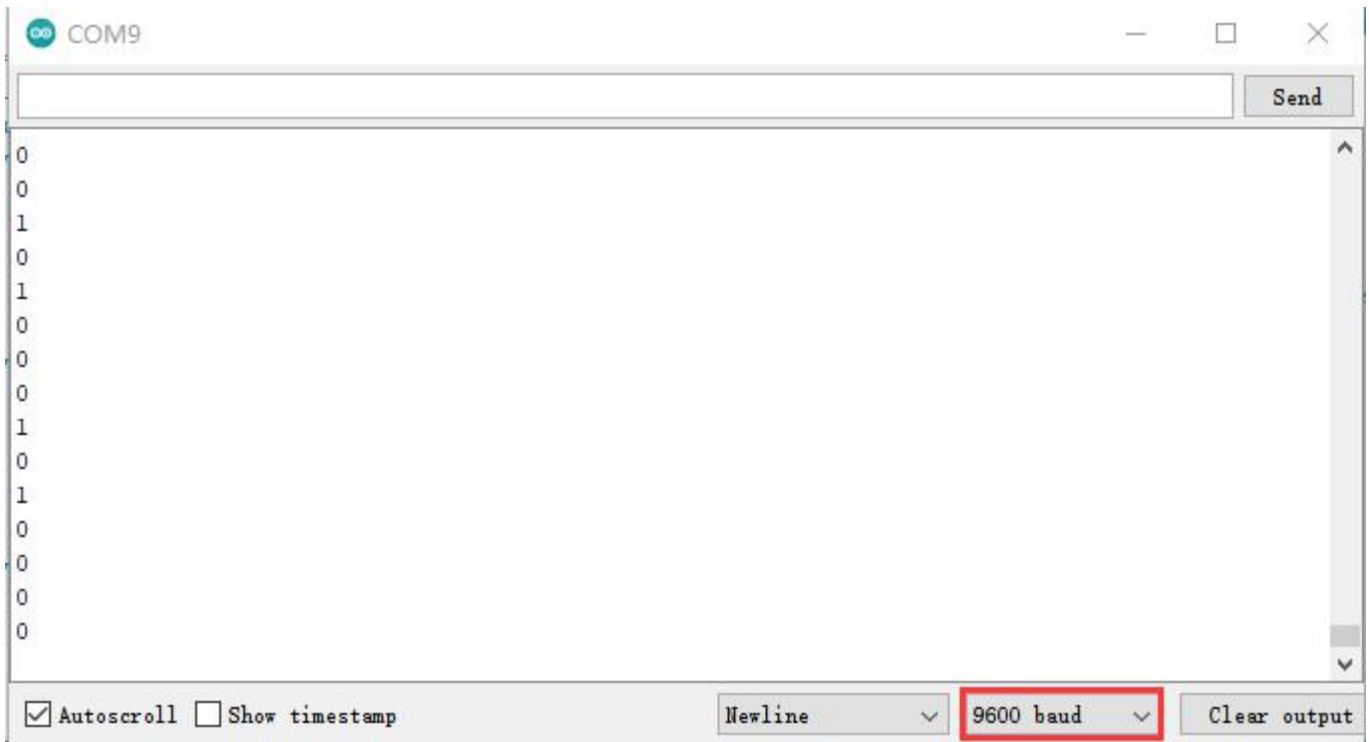
This module is compatible with the TS2178 TelePort control board.

Test Code

```
#define SensorPin 9 //Connect the sensor to digital Pin 7 which is Interrupt 1.
int val = 0; //define the initial value of variable 'val' as 0
unsigned char state = 0;
void setup()
{
  Serial.begin(9600); //set baud rate at 9600
  pinMode(SensorPin, INPUT);
  attachInterrupt(1, blink, FALLING); // Trigger the blink function when the falling edge is detected
}
void loop()
{
  val = digitalRead(SensorPin); //read and assign analog value to variable 'val'
  Serial.println(val); //print variable 'val' by Serial.print
  delay(10);
}
void blink()//Interrupts function
{ state++;
}
```

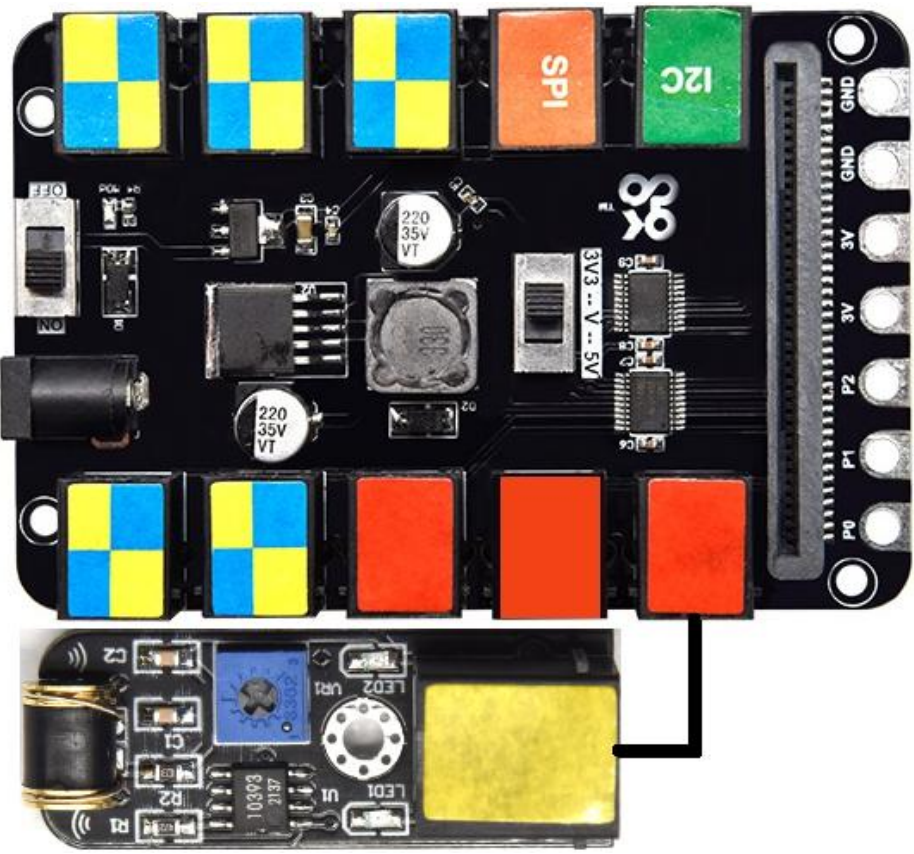
Test Result

Wire up, upload code, power it up, open serial monitor and set baud rate to 9600. When the sensor doesn't vibrate, the serial monitor will show 0; if it vibrates, the monitor will display 1.



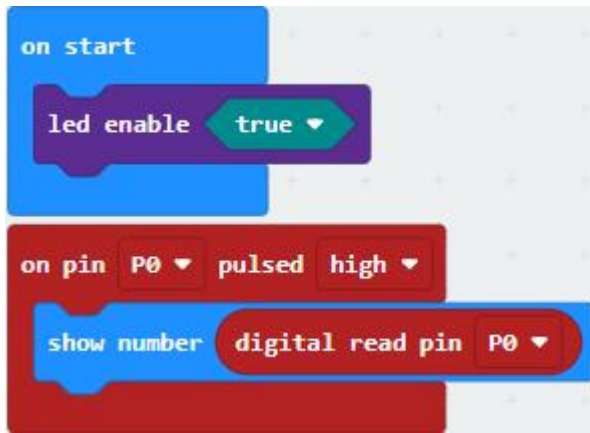
If you want to know more details about Arduino and the TelePort control board, you can refer to TS2178.

➤ **Micro:bit Application**



It is compatible with the Micro:bit board and the TS2179 Micro:bit expansion board.

Test Code



-①Run the “on start” block to boot the program
-②Open the LED matrix of the Micro:bit
-③The program is run circularly under the command of “forever” block
-④pull up the voltage at port P0
-⑤The Micro:bit will show the digital signals detected by the vibration sensor

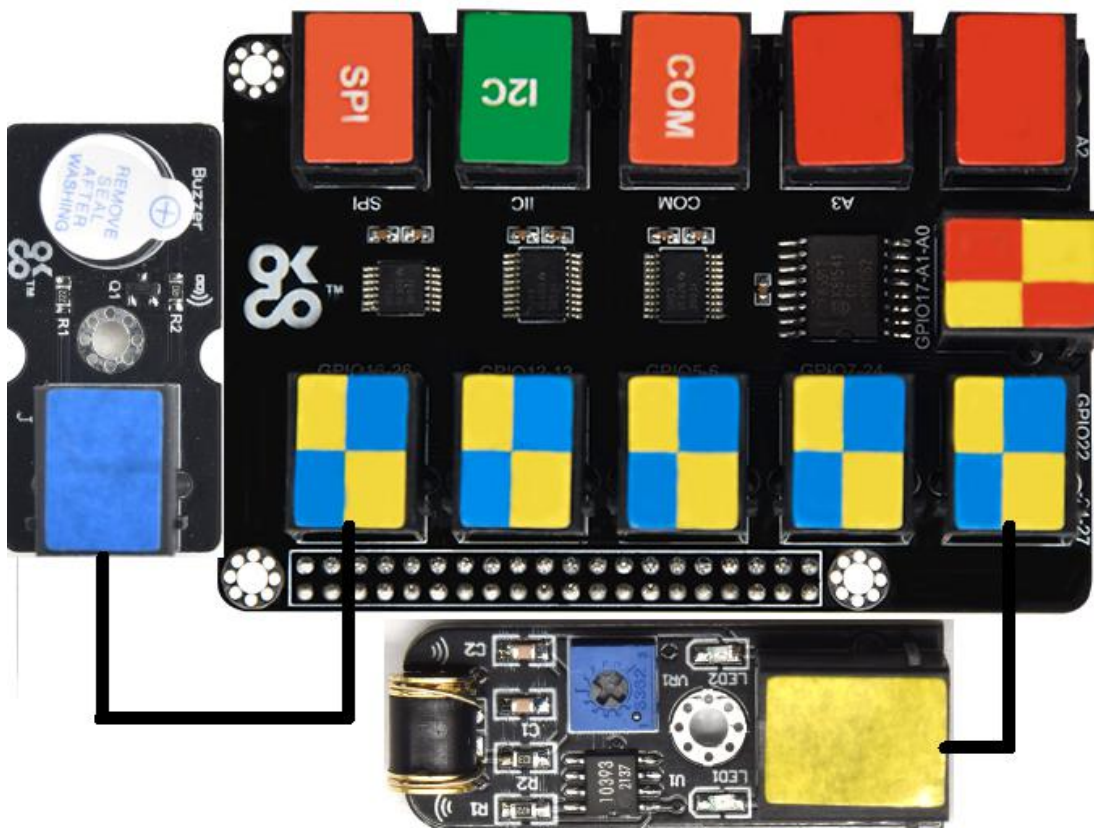
Test Result

Wire up, insert the Micro:bit V2.0 into the shield, turn the DIP switch to 3V3, upload test code and power it up.

If the sensor doesn't vibrate, the serial monitor will show 0; if it vibrates, the monitor will show 1.

If you want to know more details about the Micro:bit board and Micro:bit shield, you can refer to TS2179.

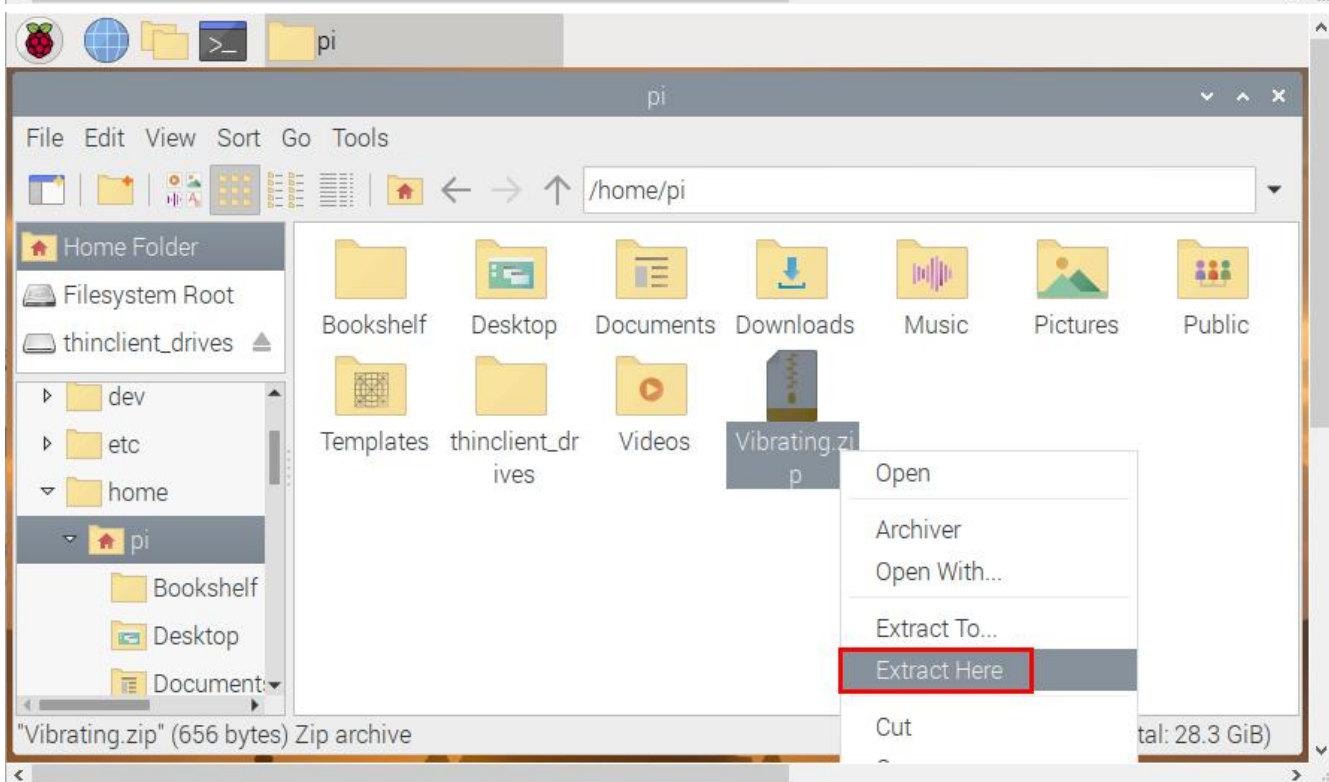
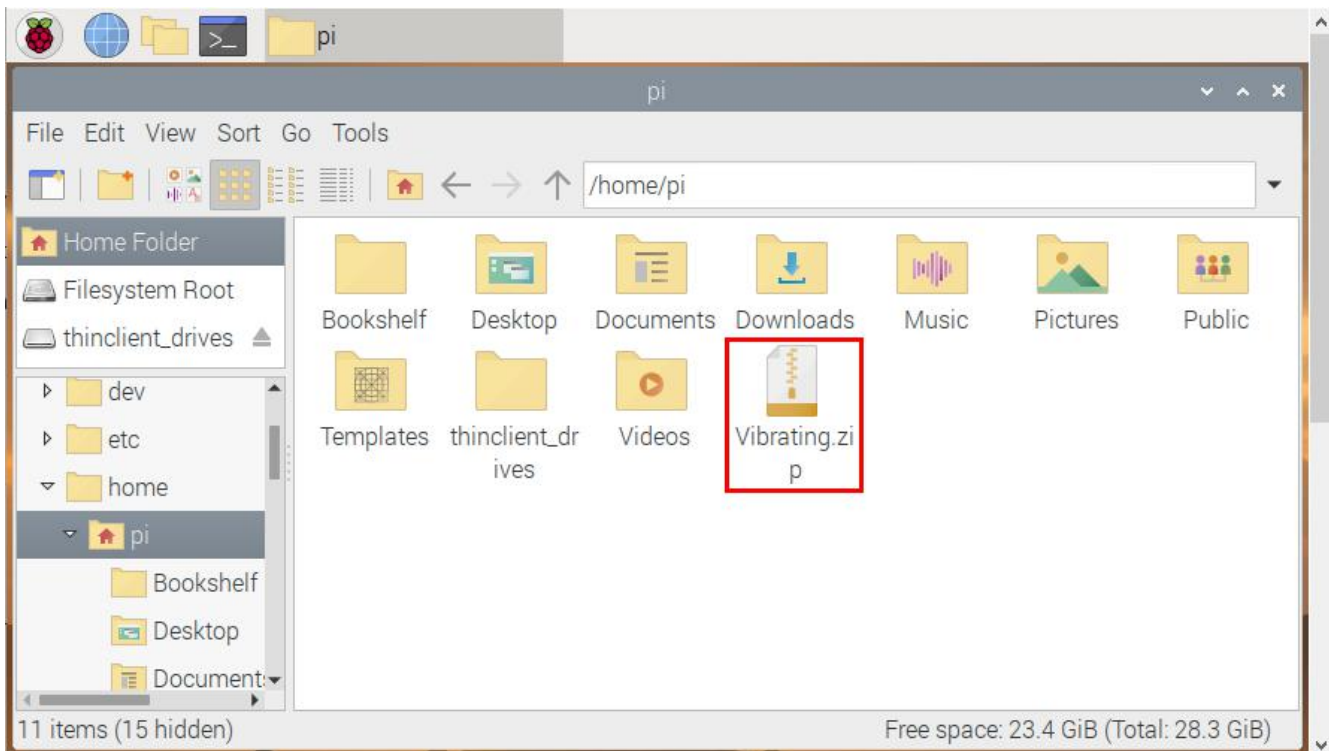
➤ Raspberry Pi Application

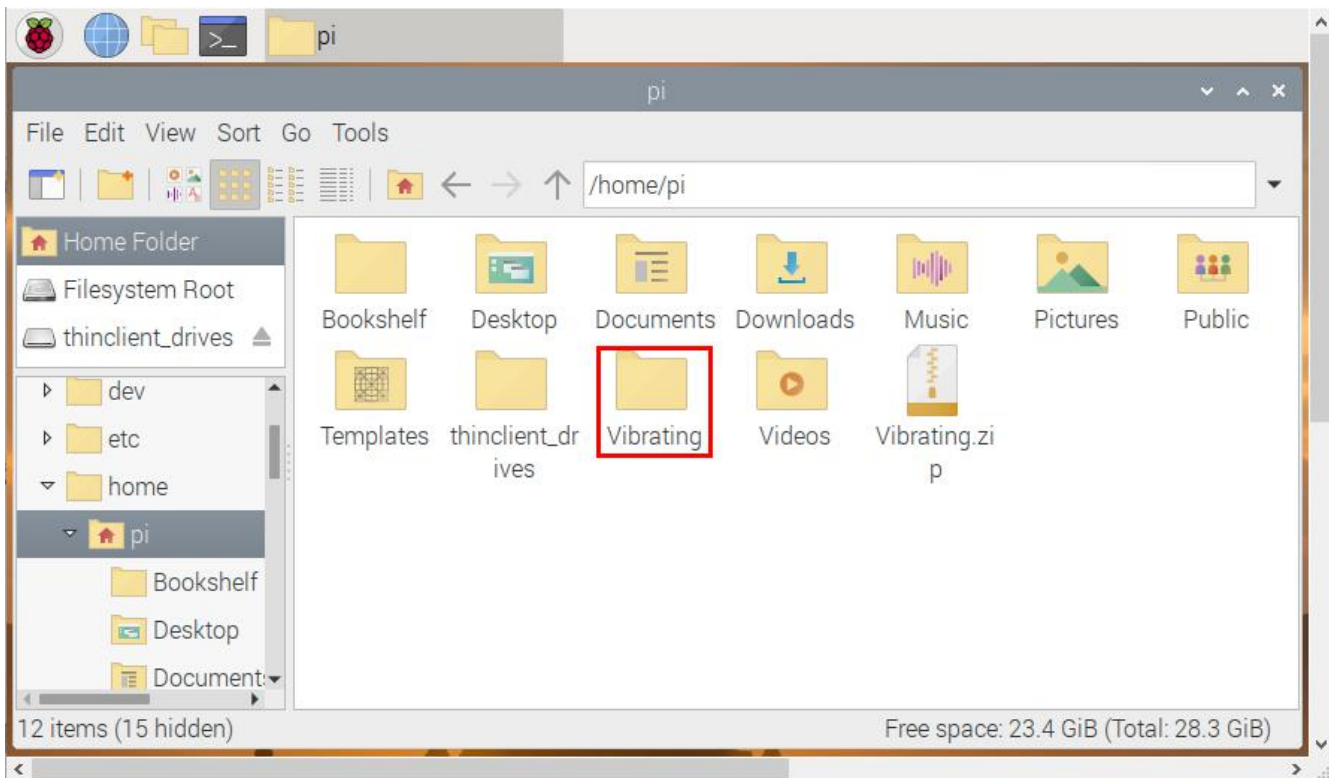


This module is compatible with the Raspberry Pi board and the TS2180 Raspberry Pi shield.

Copy the test code to Raspberry Pi system to run it

(1) Save the test code in the **pi** folder of Raspberry Pi system. Then place the **Vibrating.zip** file we provide in the **pi** folder, right-click and click **Extract Here**. As shown below:





(2) Compile and run test code:

Input the following code and press "Enter"

```
cd /home/pi/Vibrating
gcc Vibrating.c -o Vibrating-lwiringPi
sudo ./Vibrating
```

(3) Test Result:

Insert the shield into the Raspberry Pi board. After programming finishes, when you knock the vibration sensor, the terminal will show "buzzer ring..., ...buzzer off" and the buzzer will chime. If you don't knock it, the buzzer won't emit sounds and the terminal will show "...buzzer off".

Note: press Ctrl + C to exit code running

Test Code

File name: **Vibrating.c**

```
#include <wiringPi.h>
#include <stdio.h>

#define vibPin 3 //vibration pin BCM GPIO 22
#define buzPin 25 //buzzer pin BCM GPIO 26
int buz_status = 0;

void swbuz(void)
{
    buz_status = ~buz_status;
    digitalWrite(buzPin, buz_status);
}
```

```
if(buz_status == 1)
{
    printf("buzzer ring ...");
}
else
{
    printf("...buzzer off");
}
}

int main()
{
    wiringPiSetup();
    pinMode(buzPin, OUTPUT);
    pinMode(vibPin, INPUT);
    pullUpDnControl(vibPin, PUD_UP);
    wiringPiISR(vibPin, INT_EDGE_FALLING, swbuz); //interrupt

    while(1)
    {
        //val=digitalRead(vibPin); //Receive
        //printf("value = %d\n", val);
    }
}
```

If you want to know how to utilize Raspberry Pi and the Raspberry Pi shield, you can refer to TS2180.

END