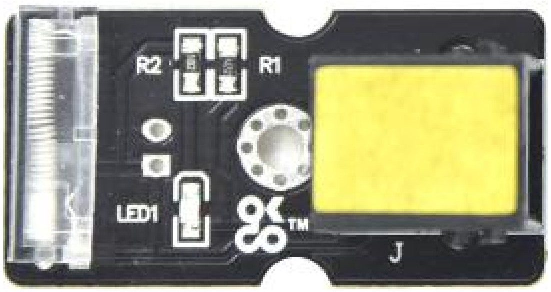
Knock Sensor
(000x0000 Article Number)
(TS2142)

DESIGN THE WORLD

Product Details



The TelePort knock sensor is mainly composed of a SW-280 vibration switch, which is not only an inductive knock switch but also an electronic switch.

When it is knocked, the sensing results will be transmitted to the circuit device and the circuit will start work.

Features and Benefits

- Compatible with RJ11 6P6C OKdo TelePort Control boards and expansion shields.
- A non-directional vibration module based on the SW-280 sensor that outputs HIGH when stable, LOW when vibration is detected.

Technical Specifications

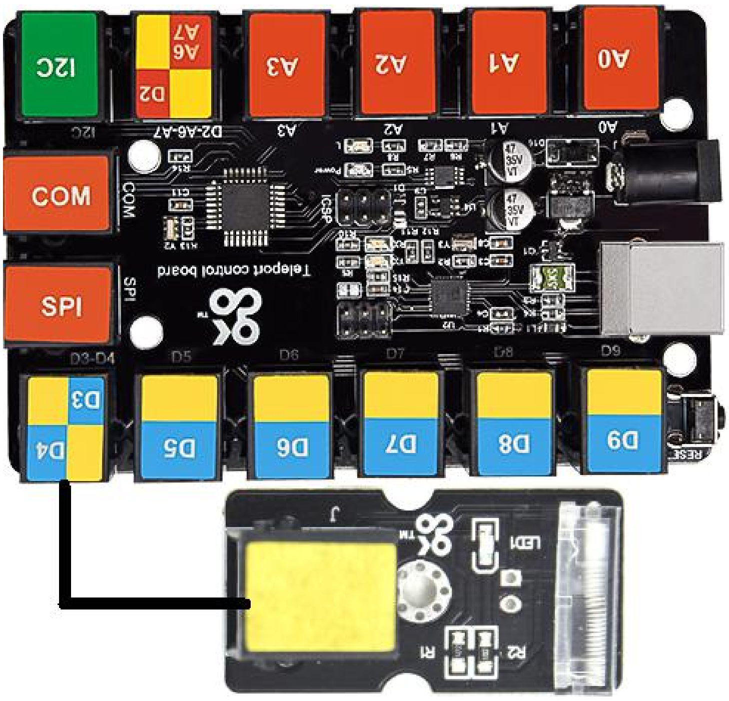
Sensor type	Digital input
Working voltage	3.3V-5V
Dimensions	44mm*20mm*18mm
Weight	5.5g

Applications

- Electronic drum
- Knock 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

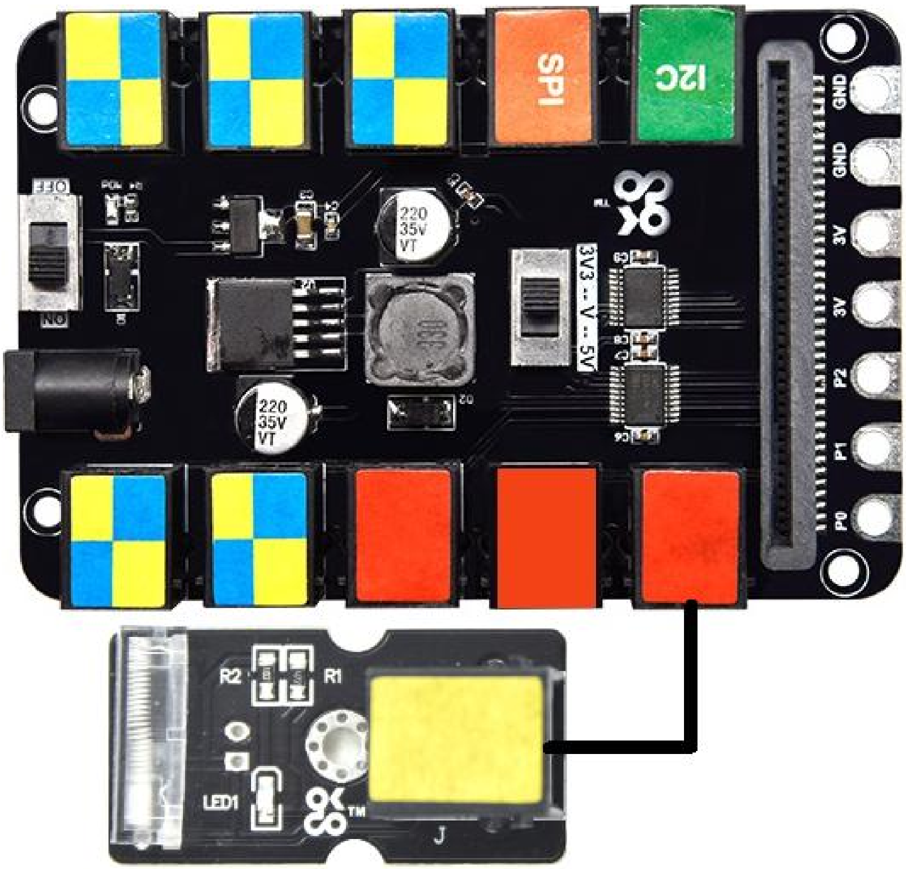
```
int ledPin = 13;// choose the pin for the LED
int knockPin = 3;// Connect sensor to input pin 3
void setup() {
pinMode(13, OUTPUT);// declare LED as output
pinMode(3, INPUT_PULLUP);// declare knock as INPUT_PULLUP
}
void attachInterrupt_fun_FALLING_3() {
  digitalWrite(13,HIGH);
}
void loop(){
  attachInterrupt(digitalPinToInterrupt(3),attachInterrupt_fun_FALLING_3,FALLING);
  delay(500);
  digitalWrite(13,LOW);
}
```

Test Result

Wire up, upload code and power it up. When you knock the white object on the module, the D13 on the control board will be on; if not, the D13 will be off.

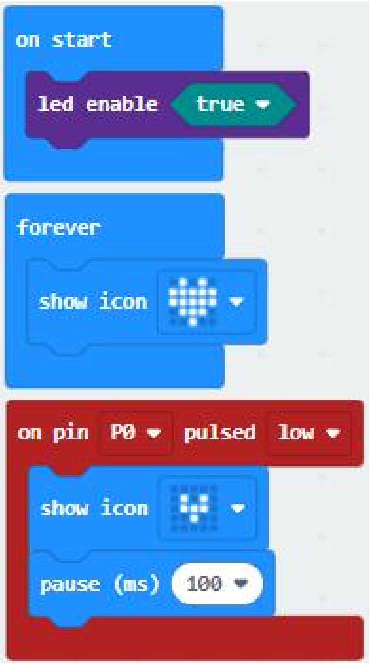
If you want to know more details about Arduino and the TelePort control board, you can refer to TS2178.

➢ Micro:bit Application



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

Test Code



```
on start
    led enable  true ▾
forever
    show icon  ♥ ▾
on pin  P0 ▾  pulsed  low ▾
    show icon  ▦ ▾
    pause (ms)  100 ▾
```

..............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

..............④the Micro:bit will display “♥ ”

..............⑤pull down the voltage at port P0

..............⑥the Micro:bit will show “▦”

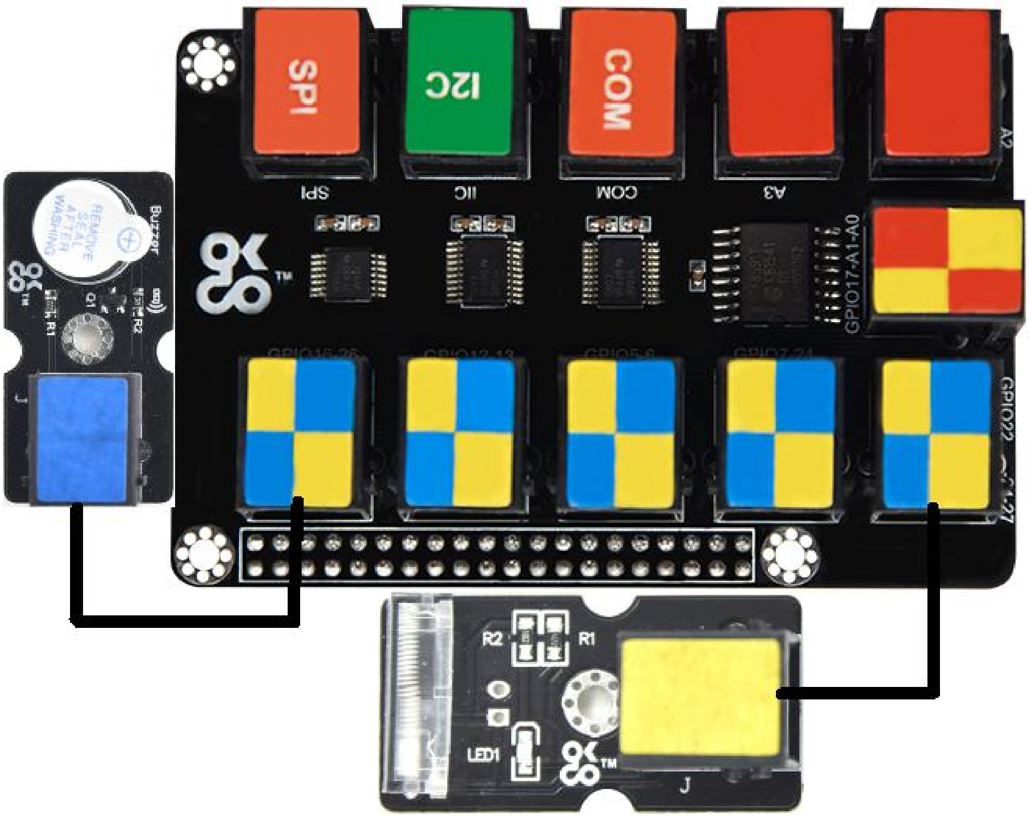
..............⑦delay in 100ms

Test Result

Wire up, insert the Micro:bit V2.0 into the shield, turn DIP switch to 3V3, upload test code and power it up. When you knock the white object on the module, the Micro:bit will show“▦”; on the contrary, the Micro:bit will display“♥ ”.

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

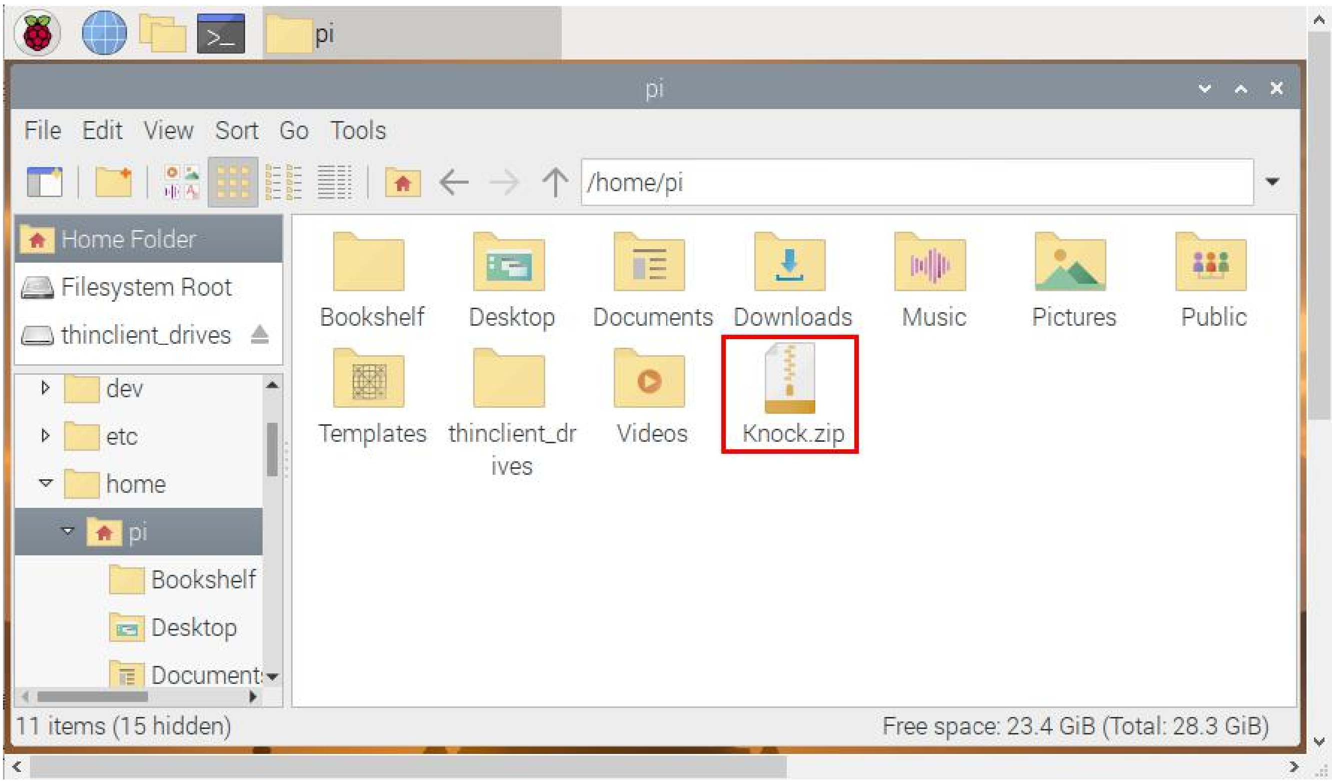
- **Raspberry Pi Application**

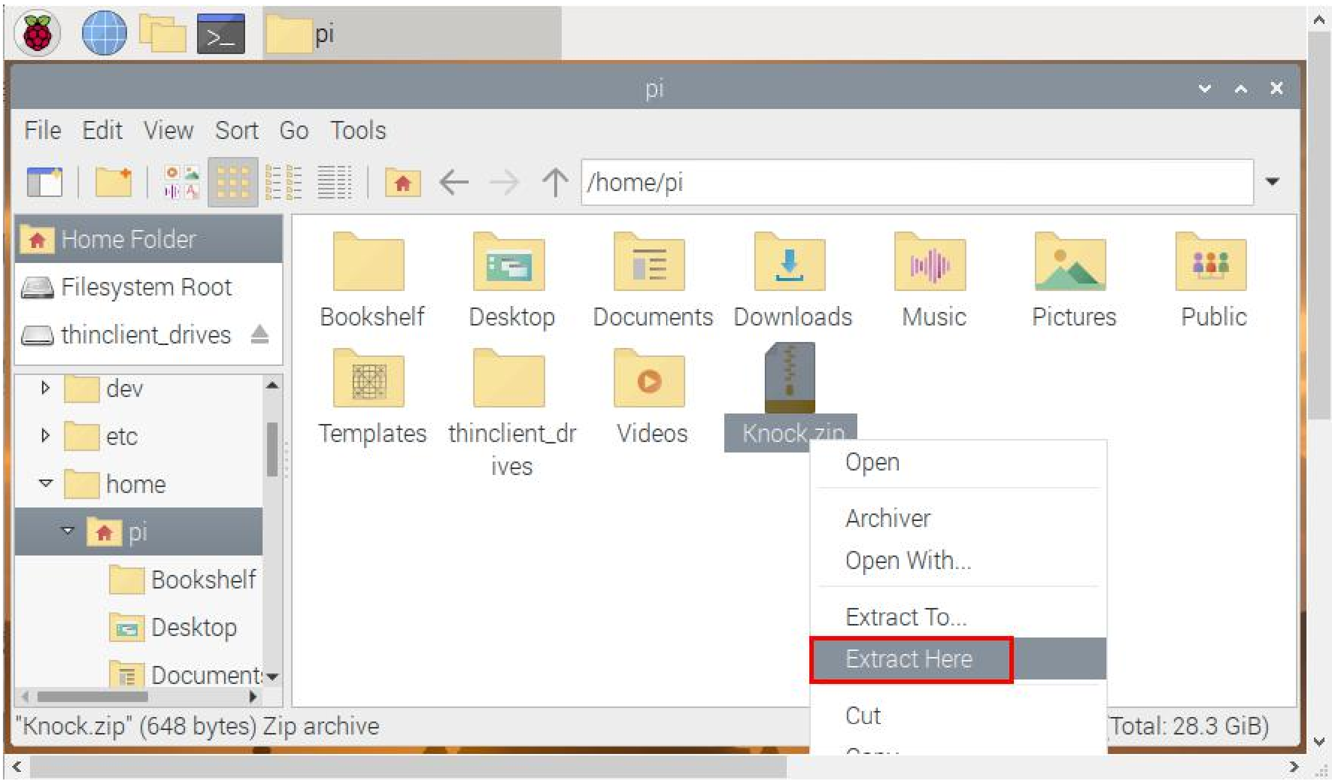


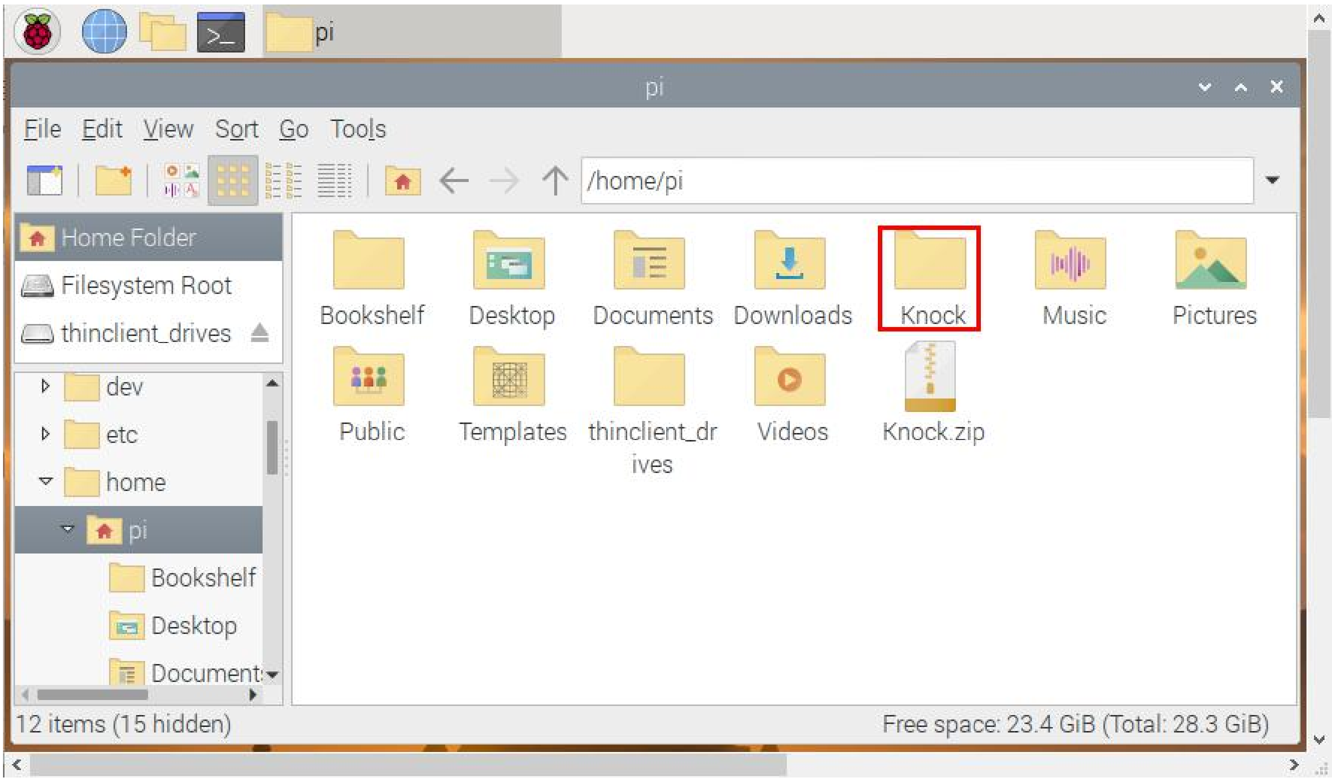
This module is compatible with the Raspberry Pi board and the TS2180 Raspberry Pi shield.

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 Knock.zip file we provide in the **pi** folder, right-click and click **Extract Here.** As shown below:


pi
File Edit View Sort Go Tools
/home/pi
Home Folder
Filesystem Root
thinclient_drives
dev
etc
home
pi
Bookshelf
Desktop
Document
Bookshelf
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
thinclient_drives
Videos
Knock.zip
11 items (15 hidden)
Free space: 23.4 GiB (Total: 28.3 GiB)


pi
File Edit View Sort Go Tools
/home/pi
Home Folder
Filesystem Root
thinclient_drives
dev
etc
home
pi
Bookshelf
Desktop
Document
Bookshelf
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
thinclient_drives
Videos
Knock.zip
Open
Archiver
Open With...
Extract To...
Extract Here
Cut
"Knock.zip" (648 bytes) Zip archive
Total: 28.3 GiB)



(2) Compile and run test code：

Input the following code and press“Enter”

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

(3) Test Result：

Insert the shield into the Raspberry Pi board. After programming finishes, when you knock the white object on the module consistently, the terminal will show“buzzer ring..., ...buzzer off”and the buzzer will chime. Nevertheless, if you don’t knock it, the terminal will show“...buzzer off”and the buzzer won’t make a sound.

Note: press Ctrl + C to exit code running

Test Code

File name: Knock.c

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

#define KnockPin 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;
  delay(100);
  digitalWrite(buzPin, buz_status);
  if(buz_status == 1)
  {
    printf("buzzer ring ...");
  }
  else
  {
    printf("...buzzer off");
  }
}

int main()
{
  wiringPiSetup();
  pinMode(buzPin, OUTPUT);
  pinMode(KnockPin,INPUT);
  pullUpDnControl(KnockPin, PUD_UP);
  wiringPiISR(KnockPin,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.