Preface

About SunFounder

SunFounder is a technology company focused on Raspberry Pi and Arduino open source community development. Committed to the promotion of open source culture, we strive to bring the fun of electronics making to people all around the world and enable everyone to be a maker. Our products include learning kits, development boards, robots, sensor modules and development tools. In addition to high quality products, SunFounder also offers video tutorials to help you build your own project. If you have interest in open source or making something cool, welcome to join us! Visit www.sunfounder.com for more!

About Sensor Kit V1.0

There are many sensors and modules in this kit, with which you can learn diverse basic knowledges by following the experiments. Also with what you've learnt during the process, you may explore more functions with the parts provided or use extra devices at hand.

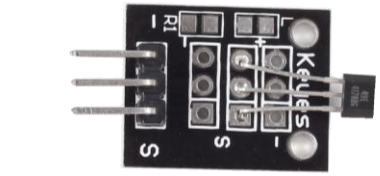
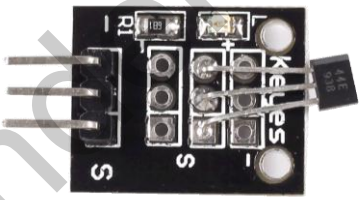
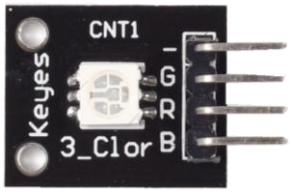
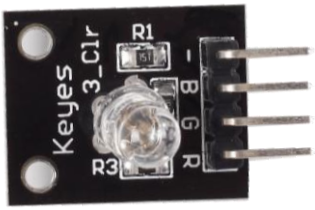
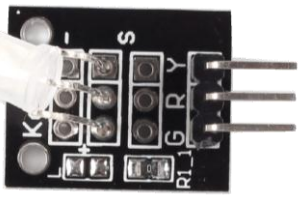
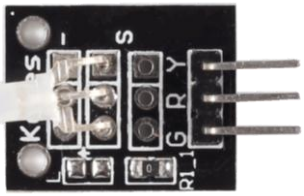
This kit is suitable for Raspberry Pi model B, model B+, Pi 2 model B and Pi 3. Though in this manual the experiments are done with model B+, you can also make them with the other two models.

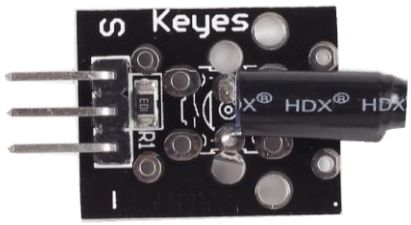
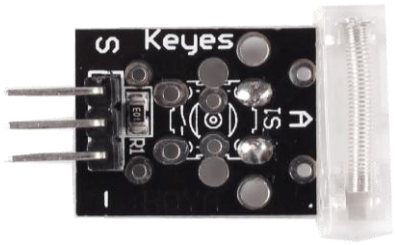
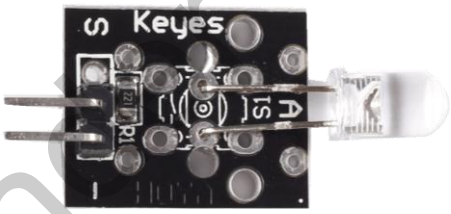
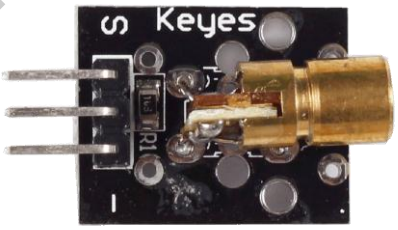
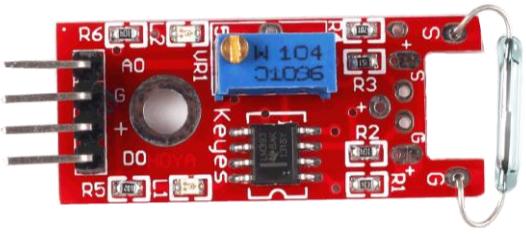
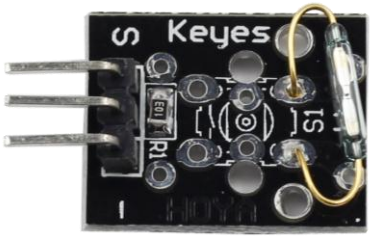
You can go to our official website www.sunfounder.com to download the related code by clicking **LEARN** -> **Get Tutorials**.

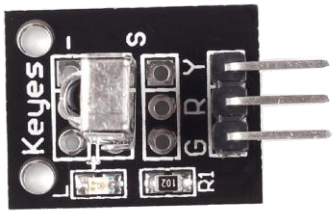
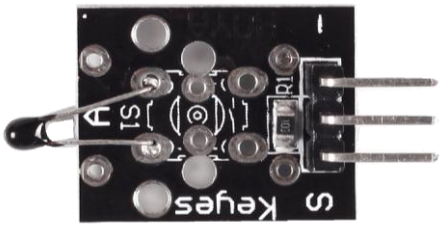
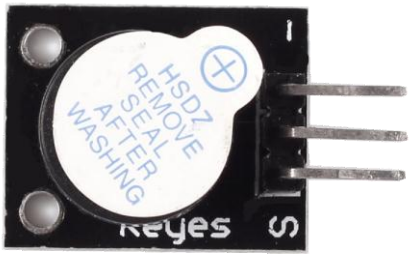
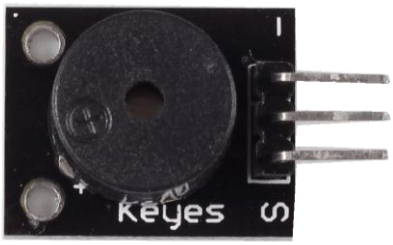
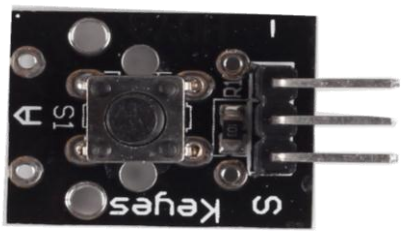
If you have any questions, please send an email to support@sunfounder.com. You can also leave a message and share your projects on our **FORUM**.

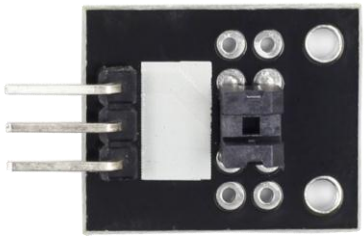
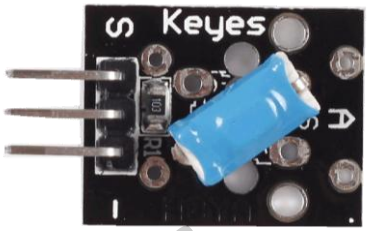
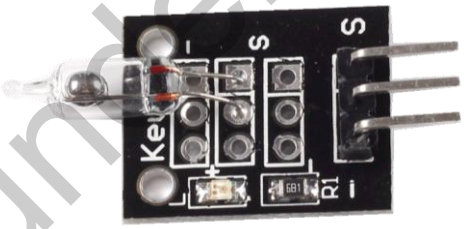
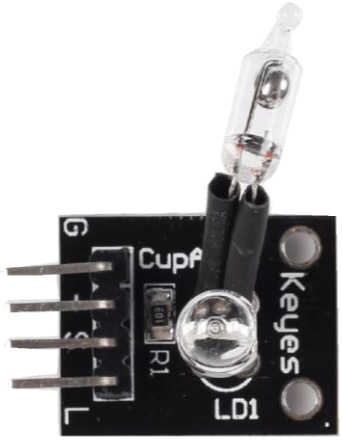
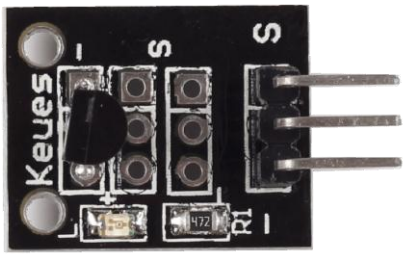
Contents

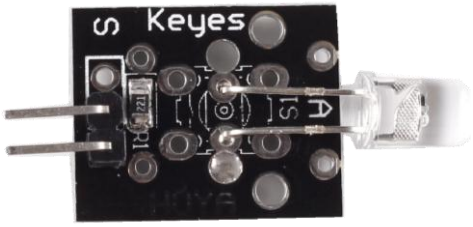
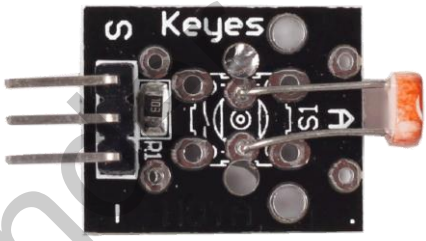
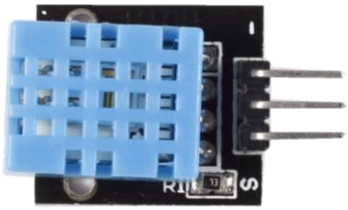
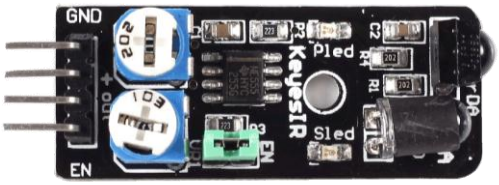
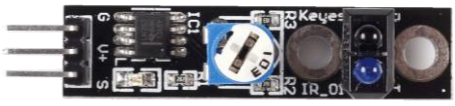
Components List

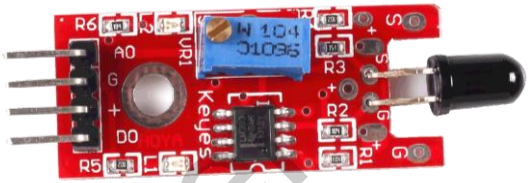
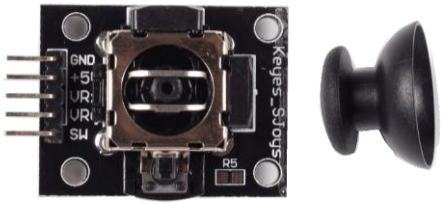
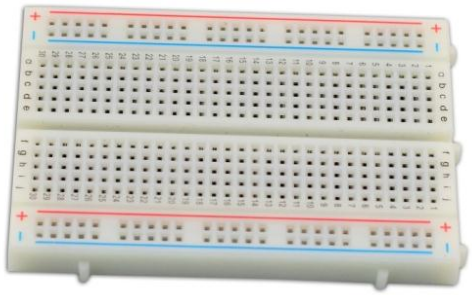
No.	Name	Qty.	Component
1	Analog Hall Sensor	2	 
2	Switch Hall Sensor	1	
3	RGB LED	2	 
4	Dual-color Common-Cathode LED	2	 

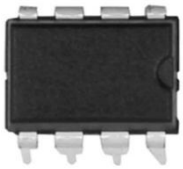
5	Shock Switch	1	
6	Knock Sensor	1	
7	Infrared Transmitter	1	
8	Laser Transmitter	1	
9	Reed Switch	1	
10	Mini Reed	1	

11	Infrared Receiver	1	
12	Analog Temperature Sensor	1	
13	Digital Temperature Sensor	1	
14	Active Buzzer	1	
15	Passive Buzzer	1	
16	Button Switch	1	

17	Photo-interrupter	1	
18	Tilt Switch	1	
19	Mercury Switch	1	
20	Magic Cup	2	
21	DS18B20 Temperature Sensor	1	

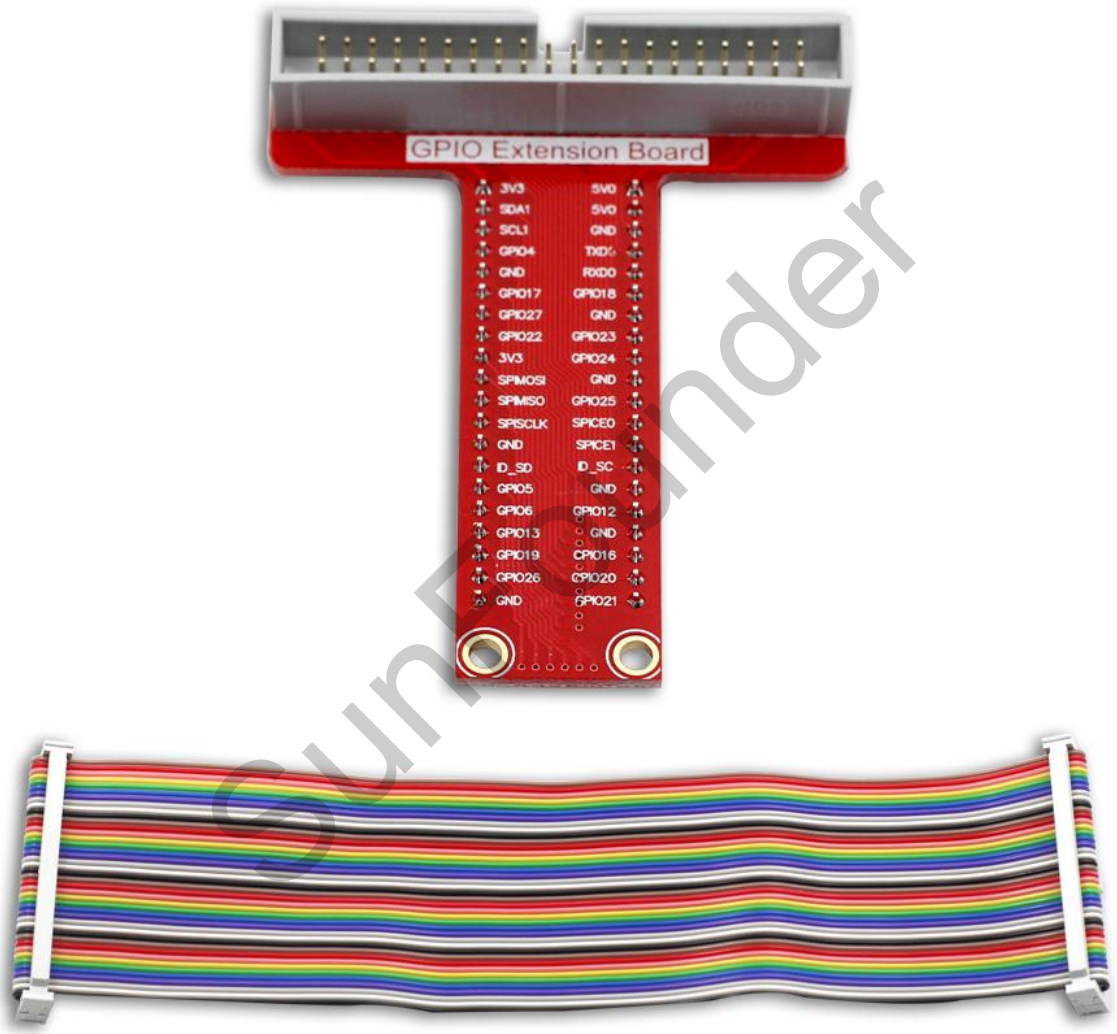
22	Rotary Encoder	1	
23	7-color Auto-flash LED	1	
24	Photoresistor Sensor	1	
25	Humiture Sensor	1	
26	Obstacle Avoidance Sensor	1	
27	Tracking Sensor	1	
28	Microphone Sensor	1	

29	High-sensitive Voice Sensor	1	
30	Metal Touch Sensor	1	
31	Flame Sensor	1	
32	Relay Module	1	
33	Joystick PS2	1	
34	MQ-2 Gas Sensor	1	
35	Breadboard	1	

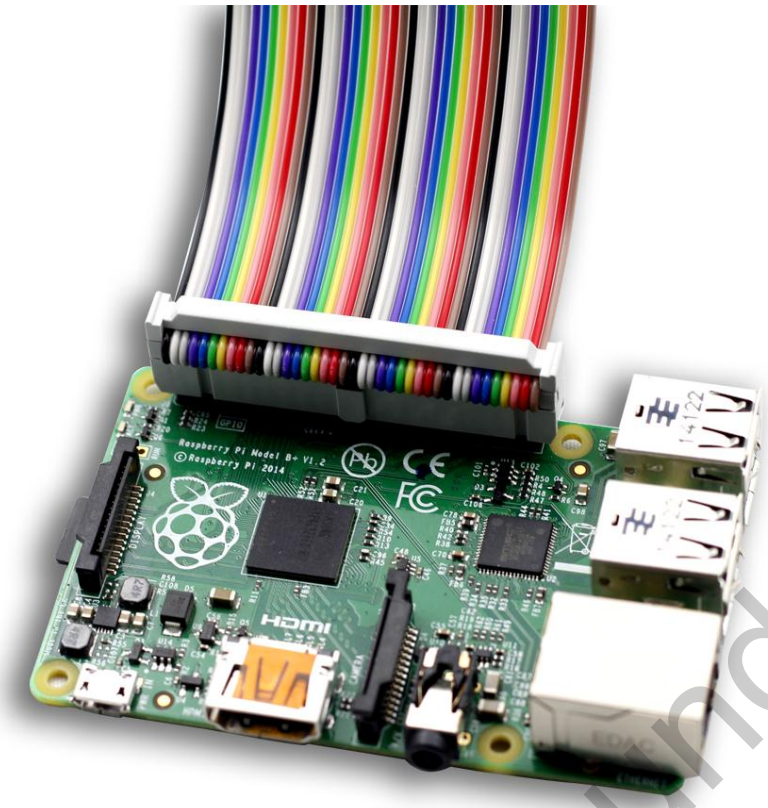
36	Jumper Wire (M to F)	40	
37	Jumper Wire (M to M)	20	
38	Jumper Wire (F to F)	20	
39	ADC0832	1	

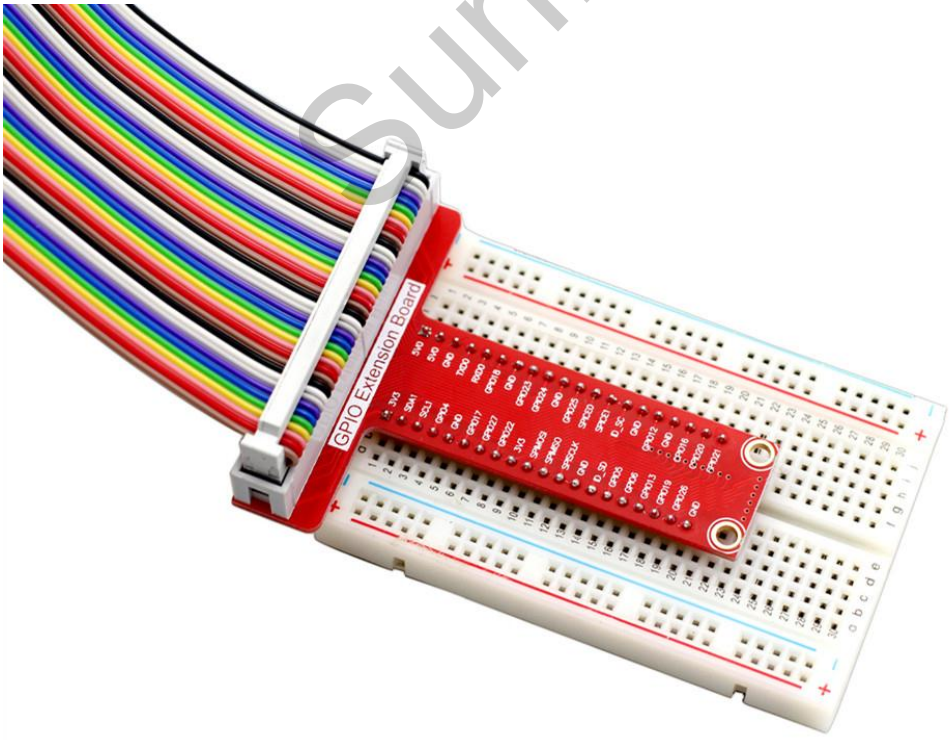
Notice

We can easily lead out pins of the Raspberry Pi to breadboard by GPIO Extension Board to avoid GPIO damage caused by frequent plugging in or out. This is our 40-pin GPIO Extension Board and GPIO cable for Raspberry Pi model B+ and Raspberry Pi 2 model B.

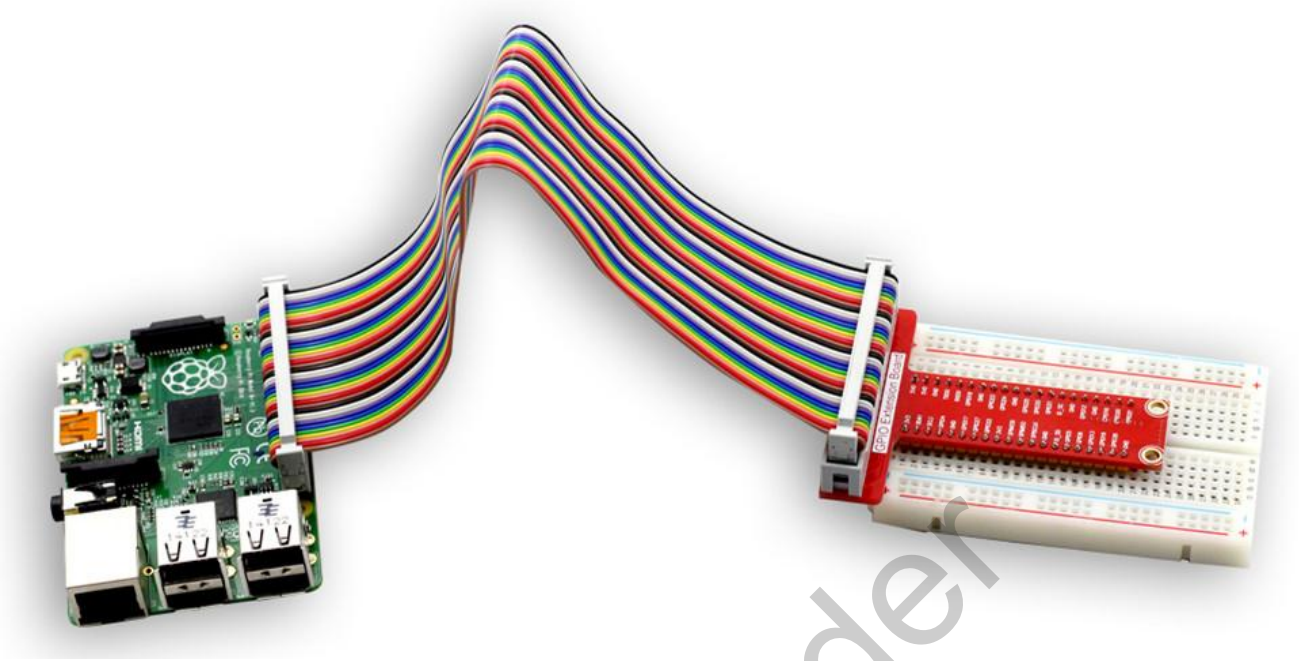


Connect the GPIO cable to the Raspberry Pi B+ like this:



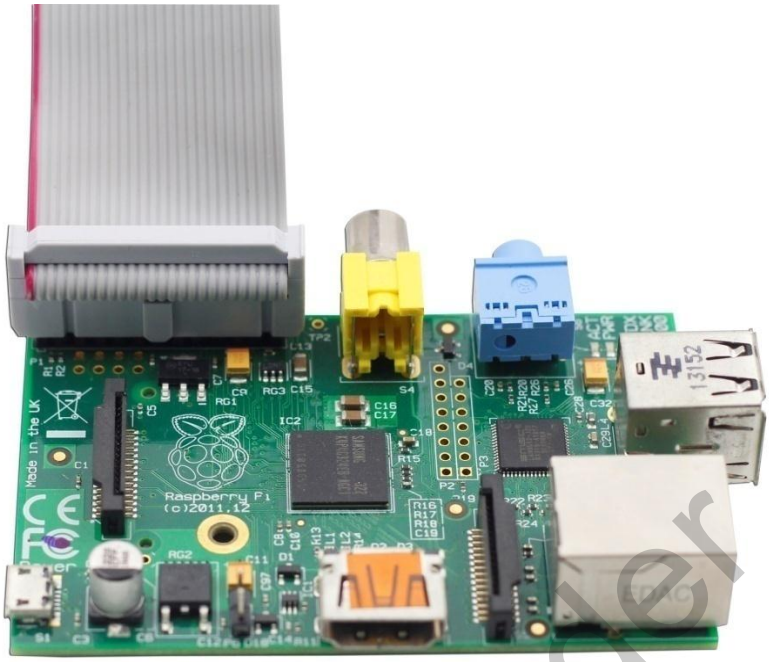


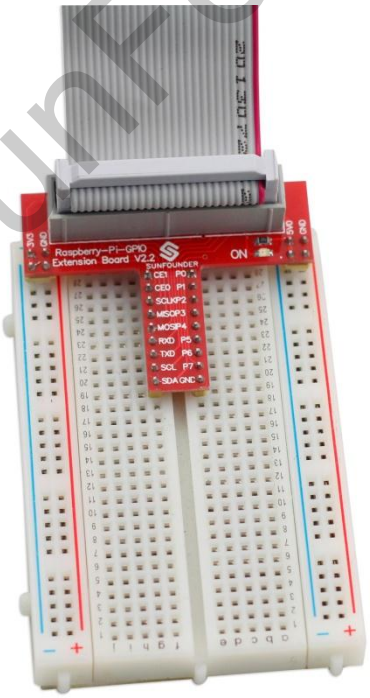
After connection, it is shown as follows:



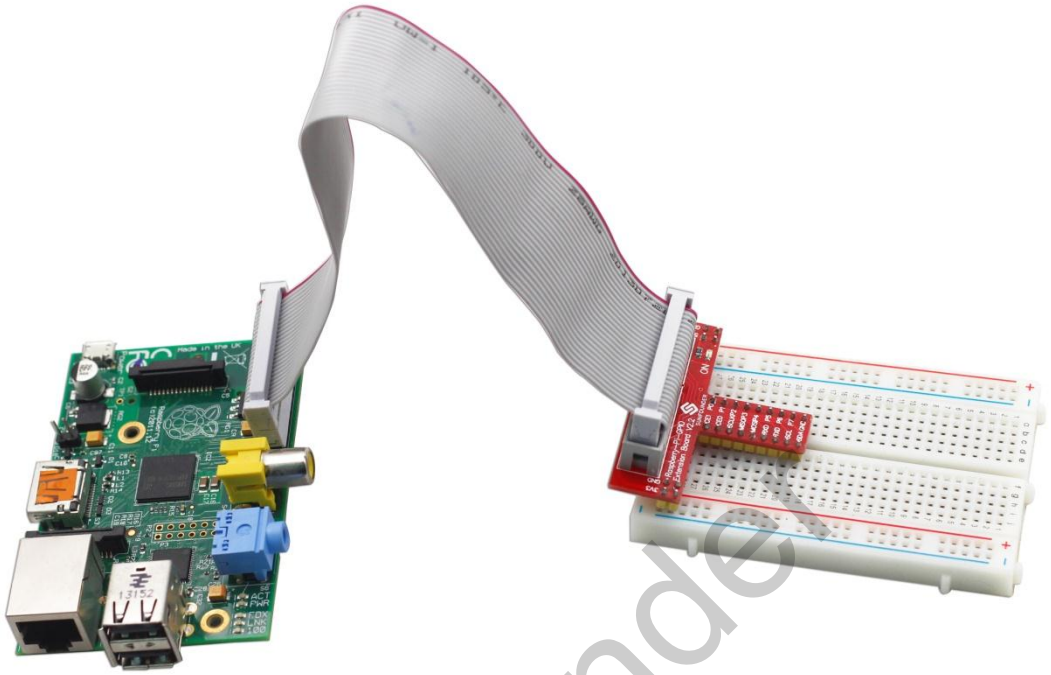
This is our 26-pin GPIO Extension Board and GPIO cable for Raspberry Pi model B.

Connect the GPIO cable to the Raspberry Pi B like this:





After connection, it is shown as follows:



Raspberry Pi Pin Number Introduction

wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin
–	–	3.3v	1 | 2	5v	–	–
8	R1:0/R2:2	SDA0	3 | 4	5v	–	–
9	R1:1/R2:3	SCL0	5 | 6	0V	–	–
7	4	GPIO7	7 | 8	TXD	14	15
–	–	0V	9 | 10	RXD	15	16
0	17	GPIO0	11 | 12	GPIO1	18	1
2	R1:21/R2:27	GPIO2	13 | 14	0V	–	–
3	22	GPIO3	15 | 16	GPIO4	23	4
–	–	3.3v	17 | 18	GPIO5	24	5
12	10	MOSI	19 | 20	0V	–	–
13	9	MISO	21 | 22	GPIO6	25	6
14	11	SCLK	23 | 24	CE0	8	10
–	–	0V	25 | 26	CE1	7	11
30	0	SDA. 0	27 | 28	SCL. 0	1	31
21	5	GPIO. 21	29 | 30	0V	–	–
22	6	GPIO. 22	31 | 32	GPIO. 26	12	26
23	13	GPIO. 23	33 | 34	0V	–	–
24	19	GPIO. 24	35 | 36	GPIO. 27	16	27
25	26	GPIO. 25	37 | 38	GPIO. 28	20	28
		0V	39 | 40	GPIO. 29	21	29
wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin

For RPi B

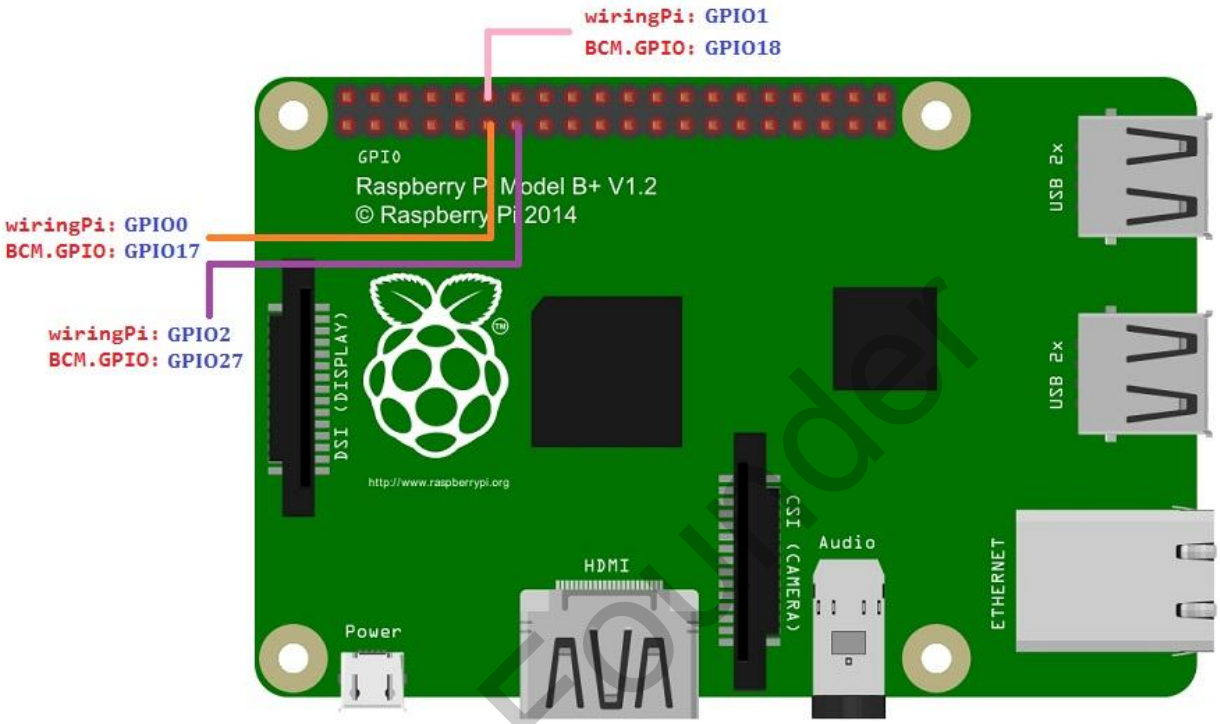
For RPi B+/ 2 model B

Currently, there are three methods of pin numbering for Raspberry Pi.

1. Numbering based on the physical location of pins
2. Numbering appointed by C language GPIO library wiringPi
3. Numbering appointed by BCM2835 SOC

If you want to operate Raspberry Pi GPIOs in C language based on the wiringPi library, choose numbering appointed by wiringPi. You can see from the above diagram that GPIO0 in wiringPi corresponds to pin 11 numbered by physical location, and GPIO30, to pin 27.

The picture below demonstrates the physical numbers of the three pins 11, 12, and 13 in detail:



If you are a C user, please install wiringPi.

WiringPi introduction:

WiringPi is a GPIO library function applied to the Raspberry Pi platform. It complies with GUN Lv3. The functions in wiringPi are similar to those in the wiring system of Arduino. They enable the users familiar with Arduino to use wiringPi more easily.

How to install

Step 1: Get the source code of wiringPi

```
git clone git://git.drogon.net/wiringPi
```

Step 2: Install wiringPi

```
cd wiringPi
git pull origin
./build
```

After you press Enter, with the script *build,* the source code of wiringPi will be compiled automatically and installed to the appropriate directory of Raspberry Pi OS.

Step 3: Test whether wiringPi is installed successfully or not
WiringPi includes lots of GPIO commands which enable you to control all kinds of interfaces on Raspberry Pi. You can test whether the wiringPi library is installed successfully or not by the following instructions.

```
gpio -v
```

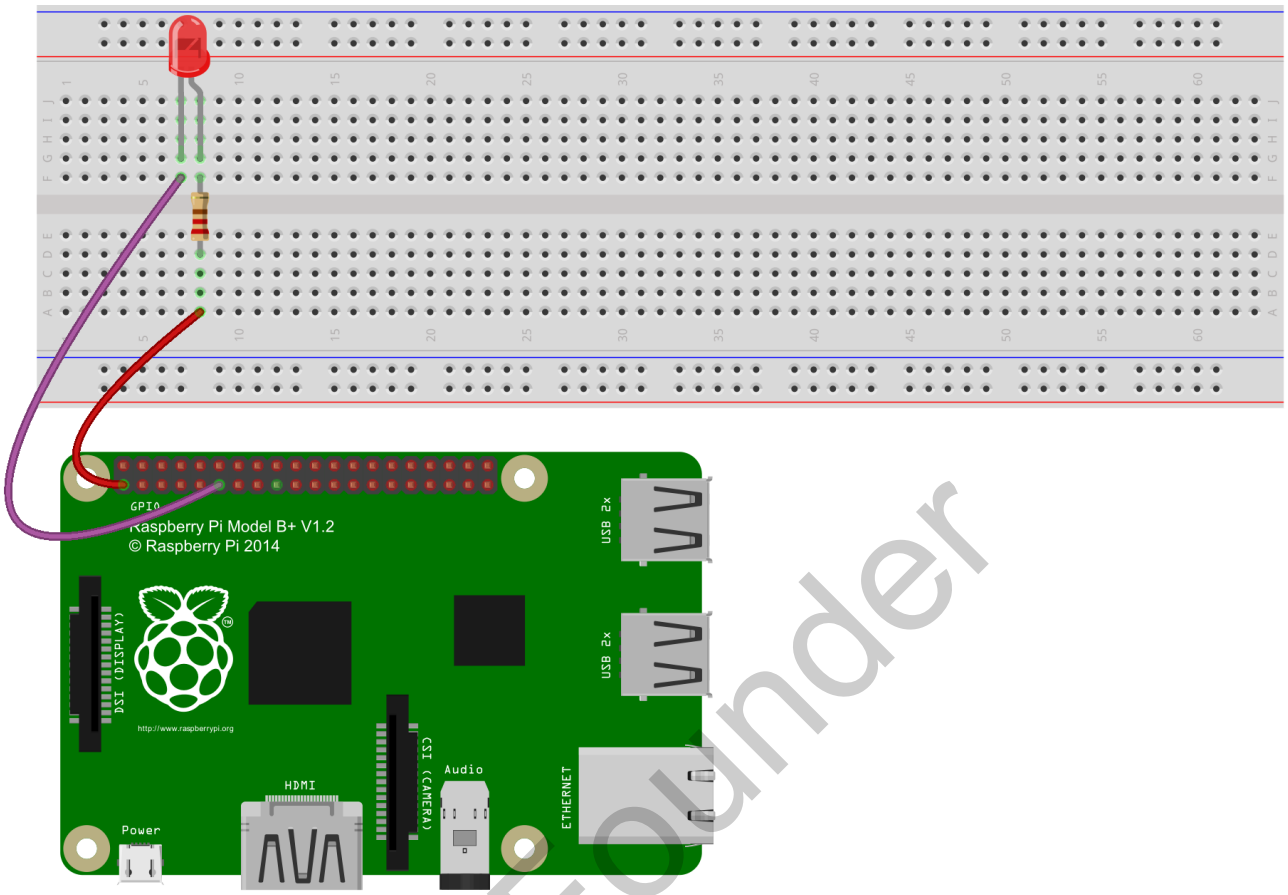
```
pi@raspberrypi ~ $ gpio -v
gpio version: 2.26
Copyright (c) 2012-2015 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Model B+, Revision: 1.2, Memory: 512MB, Maker: Sony
```

If the message above appears, the wiringPi is installed successfully.

```
gpio readall
```

```
pi@raspberrypi ~ $ gpio readall
 +-----+-----+---------+------+---+--B Plus--+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 | ALT0 | 1 |  3 || 4  |   |      | 5V      |     |     |
 |   3 |   9 |   SCL.1 | ALT0 | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 0 |  7 || 8  | 1 | ALT0 | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | ALT0 | RxD     | 16  | 15  |
 |  17 |   0 | GPIO. 0 |   IN | 0 | 11 || 12 | 0 | IN   | GPIO. 1 | 1   | 18  |
 |  27 |   2 | GPIO. 2 |   IN | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
 |  10 |  12 |    MOSI | ALT0 | 0 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO | ALT0 | 0 | 21 || 22 | 0 | IN   | GPIO. 6 | 6   | 25  |
 |  11 |  14 |    SCLK | ALT0 | 0 | 23 || 24 | 1 | OUT  | CE0     | 10  | 8   |
 |     |     |      0v |      |   | 25 || 26 | 1 | OUT  | CE1     | 11  | 7   |
 |   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | IN   | SCL.0   | 31  | 1   |
 |   5 |  21 | GPIO.21 |   IN | 1 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |   IN | 1 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
 |  13 |  23 | GPIO.23 |   IN | 0 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 |   IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
 |  26 |  25 | GPIO.25 |   IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 | 28  | 20  |
 |     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 | 29  | 21  |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+--B Plus--+---+------+---------+-----+-----+
```

Next, use C language to program. You will learn how to use the wiringPi module to control Raspberry Pi GPIOs. Take blinking LED for example. First, build the circuit:



Note: The resistor here is 220Ω. If the resistance is too high, the LED will be too dim or even won't light up. .

Step 1: Create a C file named *led.c*

```
touch led.c
```

Step 2: Use nano or other code edit tools to open *led.c*, write down the following code and save it.

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

#define  LedPin    0

int main(void)
{
```

```
    if(wiringPiSetup() == -1){//when the wiringPi initialization fails,
                              print message to the screen.
        printf("setup wiringPi failed !");
        return 1;
    }

    pinMode(LedPin, OUTPUT);

    while(1){
        digitalWrite(LedPin, LOW);    //led on
        printf("led on...\n");
        delay(500);
        digitalWrite(LedPin, HIGH);  //led off
        printf("...led off\n");
        delay(500);
    }
    return 0;
}
```

Step 3: Compile

```
gcc led.c -o led -lwiringPi
```

Step 4: Run

```
sudo ./led
```

You should see the LED blinking. Press Ctrl+C to terminate the program.

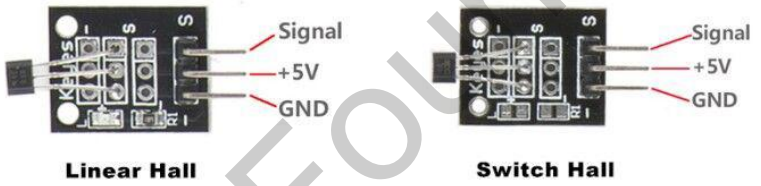
Lesson 1 Hall Sensor

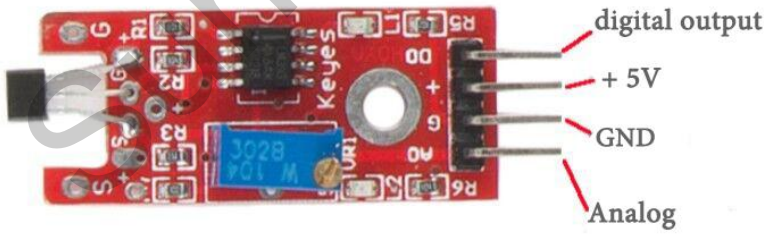
Introduction

Based on the Hall Effect, a hall sensor is one that varies its output voltage in response to a magnetic field. Hall sensors are used for proximity switching, positioning, speed detection, and current sensing applications.

Hall sensors can be categorized into linear (analog) Hall sensors and switch Hall sensors. A switch Hall sensor consists of voltage regulator, Hall element, differential amplifier, Schmitt trigger, and output terminal and it outputs digital values. A linear Hall sensor consists of a Hall element, linear amplifier, and emitter follower and it outputs analog values.

There are three types of hall sensor module in this kit (as shown below): linear Hall sensor which outputs analog signals (in two forms), and switch Hall sensor which outputs digital signals. If you add a comparator to the linear Hall sensor, it will be able to output both analog and digital signals.





Linear Hall

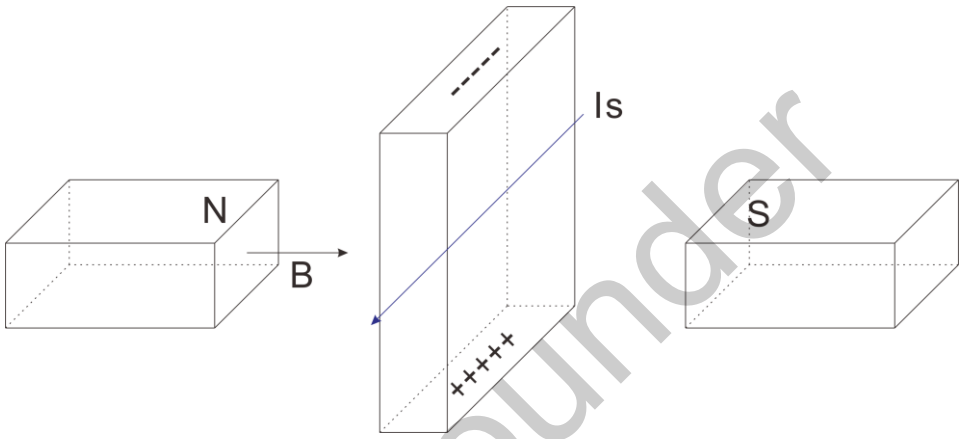
Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Linear Hall sensor module
- 1 * Dual-color Common-Cathode LED module
- 1 * Switch Hall module
- Several jumper wires

Experimental Principle

Hall Effect

Hall Effect is a kind of electromagnetic effect. It was discovered by Edwin Hall in 1879 when he was researching conductive mechanism about metals. The effect is seen when a conductor is passed through a uniform magnetic field. The natural electron drift of the charge carriers causes the magnetic field to apply a Lorentz force (the force exerted on a charged particle in an electromagnetic field) to these charge carriers. The result is what is seen as a charge separation, with a buildup of either positive or negative charges on the bottom or on the top of the plate.



Hall Sensor

A hall sensor is a kind of magnetic field sensor based on it.

Electricity carried through a conductor will produce a magnetic field that varies with current, and a Hall sensor can be used to measure the current without interrupting the circuit. Typically, the sensor is integrated with a wound core or permanent magnet that surrounds the conductor to be measured.

Experimental Procedures

For the **switch Hall sensor**, take the following steps.

Step 1: Build the circuit

Connect GPIO0 of the Raspberry Pi to pin S of the Switch Hall Module

Connect GPIO1 of the Raspberry Pi to pin S of the Dual-Color LED Module

Connect GND of the Raspberry Pi to GND of the Switch Hall Module and GND of the Dual-Color LED Module

Connect pin 3.3V of the Raspberry Pi to pin + of the Switch Hall Module and pin + of the Dual-Color LED Module

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/01_ hall/switch_hall.c)

Step 3: Compile

gcc switch_hall.c -lwiringPi

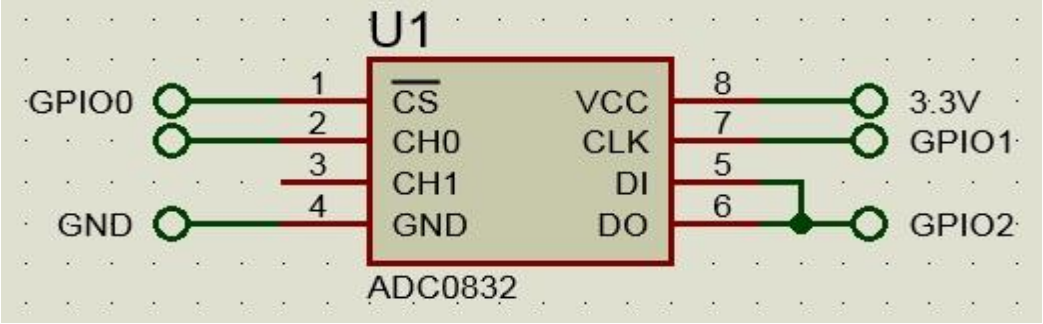
Step 4: Run

./a.out

Place a magnet close to the switch Hall sensor. Then a string "**Detected magnetic materials**" will be printed on the screen, and the LED lights up.

For **linear Hall sensor**, take the following steps.

Step 1: Build the circuit



Connect pin S of the linear Hall sensor to pin CH0 of ADC0832

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/02_LinearHall/linearHall.c)

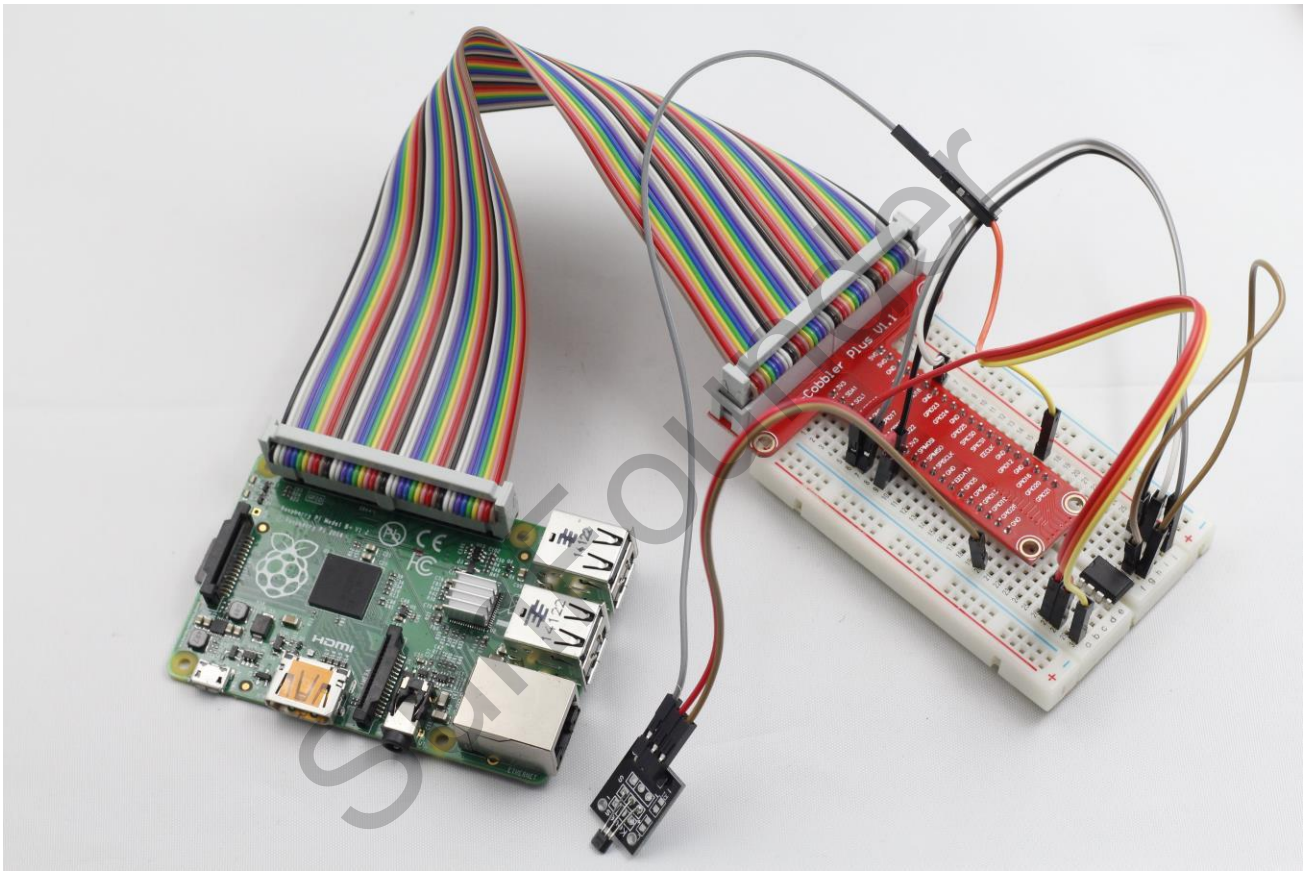
Step 3: Compile

```
gcc  linearHall.c -lwiringPi
```

Step 4: Run

```
./a.out
```

Place a magnet close to the linear Hall sensor and the value printed on the screen will increase.



For **linear Hall sensor (with a comparator)**, take the following steps.

Step 1: Build the circuit

Connect pin AO of the linear Hall sensor to pin CH0 of ADC0832

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/03_LinearHall/linearHall.c)

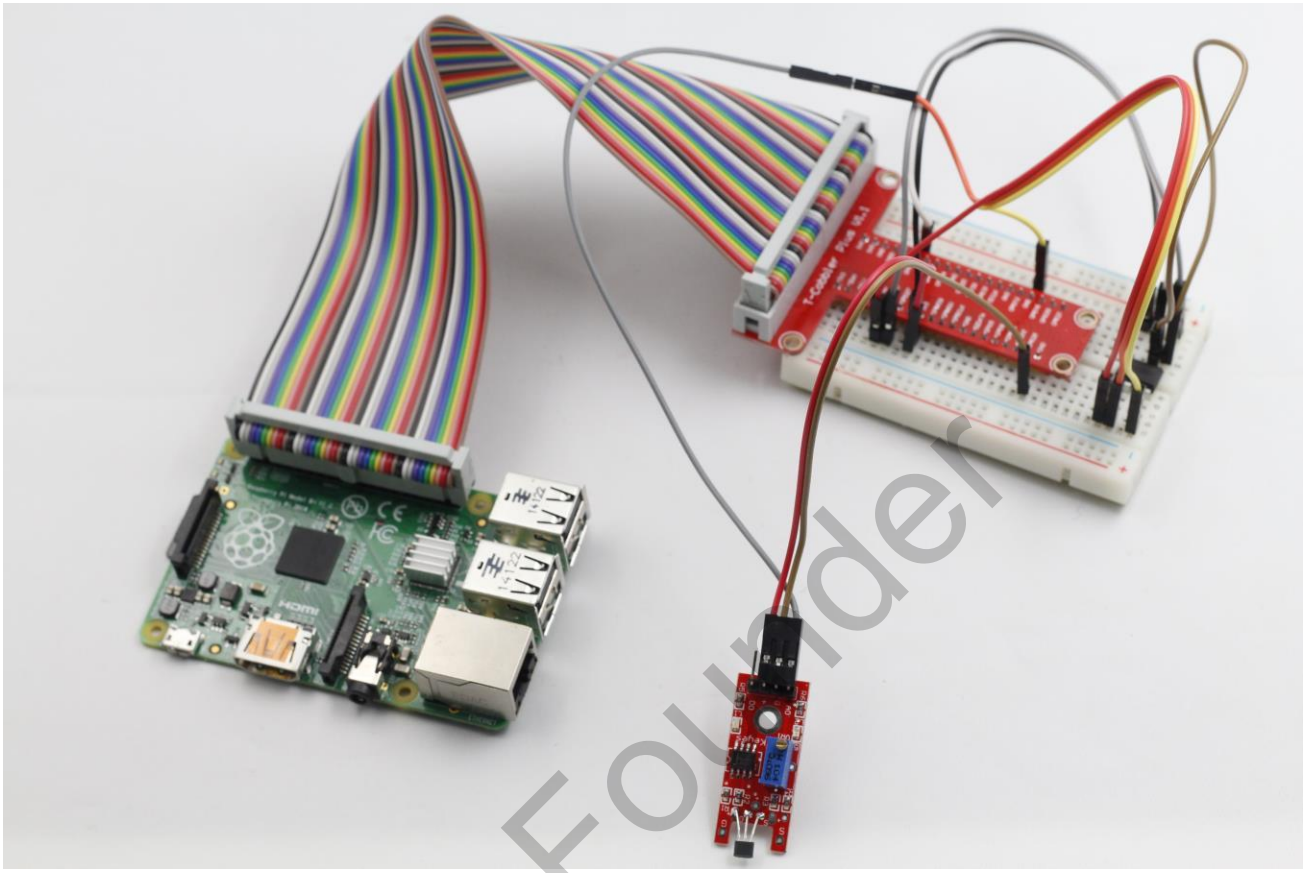
Step 3: Compile

```
gcc  linearHall.c -lwiringPi
```

Step 4: Run

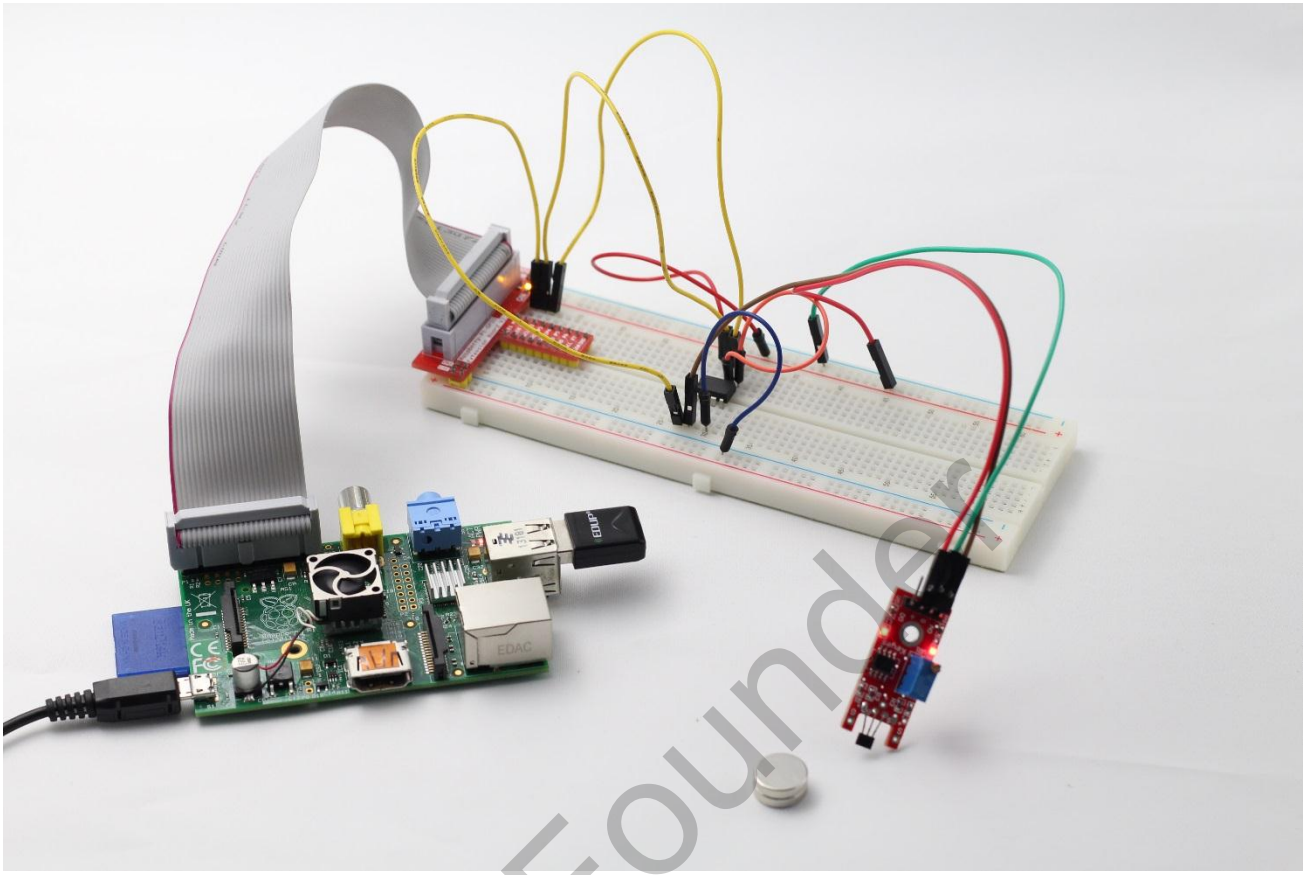
```
./a.out
```

Put a magnet near the linear Hall sensor. The indicator light on the linear Hall sensor will brighten. At the same time, the current intensity of the magnetic field will be printed on the screen.



If you have a Raspberry Pi **model B**, the wiring and command are the same with model B+.

The only difference is the 26-pin T-cobbler for model B.

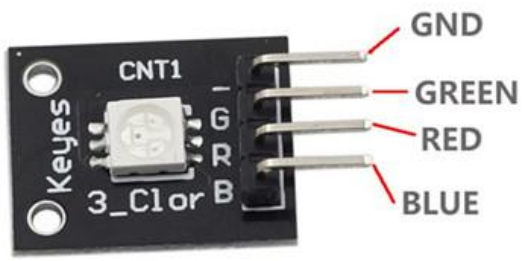


The case is similar for the following lessons in this kit.

Lesson 2 RGB LED

Introduction

There are two kinds of packages for RGB LED (as shown below) in this kit. One is Surface Mount Device (SMD) type, and the other is Dual In-line Package (DIP) type.



SMD Package

DIP Package

RGB LED modules can emit various colors of light. Three LEDs of red, green, and blue are packaged into a transparent or semitransparent plastic shell with four pins led out. The three primary colors, red, green, and blue, can be mixed and compose all kinds of colors by brightness, so you can make an RGB LED emit colorful light by controlling the circuit.

Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * RGB LED module
- Several jumper wires

Experimental Principle

The three primary colors can be mixed into various colors by brightness. The brightness of LED can be adjusted with PWM. Raspberry Pi has only one channel for hardware PWM output, but it needs three channels to control the RGB LED, which means it is difficult to control the RGB LED with the hardware PWM of Raspberry Pi. Fortunately, the *softPwm* library simulates PWM (softPwm) by programming. Thus, you only need to include the header file *softPwm.h* (for C language users), and then call the API it provided to easily achieve multi-channel PWM output to control the RGB LED so as to display all kinds of colors.

RGB LEDs can be categorized into common anode type and common cathode type. In this experiment, the latter is used.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi	RGB LED
GPIO0	R
GPIO1	G
GPIO2	B
GND	-

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/04_RGB/rgb.c)

Step 3: Compile

```
gcc  rgb.c –lwiringPi -lpthread
```

Step 4: Run

```
./a.out
```

Here you should see the RGB LED flashing different colors in turn.

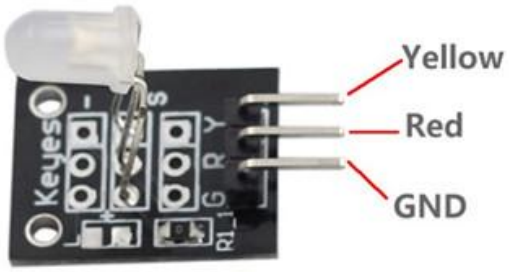
Further Exploration

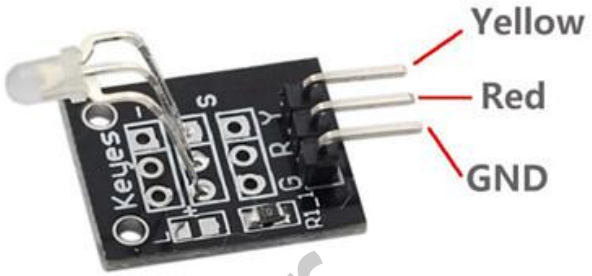
Try to modify the parameters of the function *ledColorSet( )* to change the color of the LED.

Lesson 3 Dual-color Common-Cathode LED

Introduction

There are two kinds of dual-color Common-Cathode LED (as shown below) in this kit. Their only difference is the LED package size.





Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Dual-color Common-Cathode LED module
- Several jumper wires

Experimental Principle

Connect pin Yellow and Red to GPIOs of Raspberry Pi, then program the Raspberry Pi to change the color of the LED from red to yellow, and then use PWM to mix other colors.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi	Dual-color Common-Cathode LED
GPIO0	R
GPIO1	Y
GND	G

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/05_doubleColorLed/doubleColorLed.c)

Step 3: Compile

```
gcc  doubleColorLed.c –lwiringPi -lpthread
```

Step 4: Run

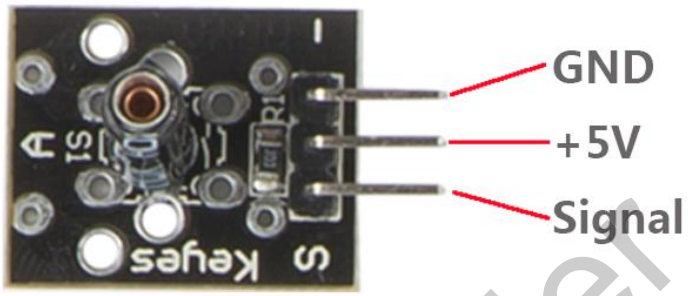
```
./a.out
```

You can see the dual-color LED flash yellow, red and then various mixed colors.

Lesson 4 Shock Switch

Introduction

A vibration switch, also called spring switch or shock sensor, is an electronic switch which induces shock force and transfers the result to a circuit device thus triggering it to work. It contains the following parts: conductive vibration spring, switch body, trigger pin, and packaging agent.



Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Shock switch module
- 1 * Dual-color Common-Cathode LED module
- Jumper wires

Experimental Principle

The shock switch works like this: the conductive vibration spring and trigger pin are precisely placed in the switch and fixed by adhesive. Normally, the spring and the trigger pin are separated. Once the sensor detects shock, the spring will vibrate and contact with the trigger pin, thus conducting and generating trigger signals.

In this experiment, a dual-color LED module is used to indicate shock signals. When the shock switch inducts shock signals, the LED will light up.

Experimental Procedures

Step 1: Build the circuit

Shock switch connection: connect pin S of the Shock switch module to GPIO0 of the Raspberry Pi; GND to GND; pin + to 3.3V

Dual-color LED module connection: connect pin R of the dual-color LED module to GPIO1 of the Raspberry Pi; GND to GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/06_shockSwitch/shockSwitch.c)

Step 3: Compile

```
gcc  shockSwitch.c  -lwiringPi
```

Step 4: Run

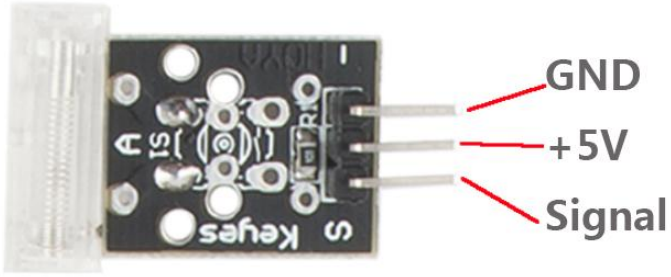
```
./a.out
```

Shake the switch and you will see the string "**Detected shaking ! count = ?**" printed on screen, and the LED will light up.

Lesson 5 Knock Sensor

Introduction

A knock sensor (as shown below) is similar to a shock switch. Being more sensitive, it can feel slighter vibrations.



Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Knock sensor module
- 1 * Dual-color Common-Cathode LED module
- Several jumper wires

Experimental Principle

It's similar to the shock switch. When you knock the sensor, the two spring leaves will get touched and the circuit will conduct. At the same time, pin S will output Low. In this experiment, we will judge the knock signal by detecting the output voltage. A dual-color LED module is used to indicate knock signals. When the knock switch generates knock signals, the LED will light up.

Experimental Procedures

Step 1: Build the circuit

Knock switch connection: connect pin S of the knock switch module to GPIO0 of the Raspberry Pi; GND to GND; pin + to 3.3V

Dual-color LED module connection: connect pin R of the dual-color LED module to GPIO1 of the Raspberry Pi; GND to GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/07_knockSensor/knockSensor.c)

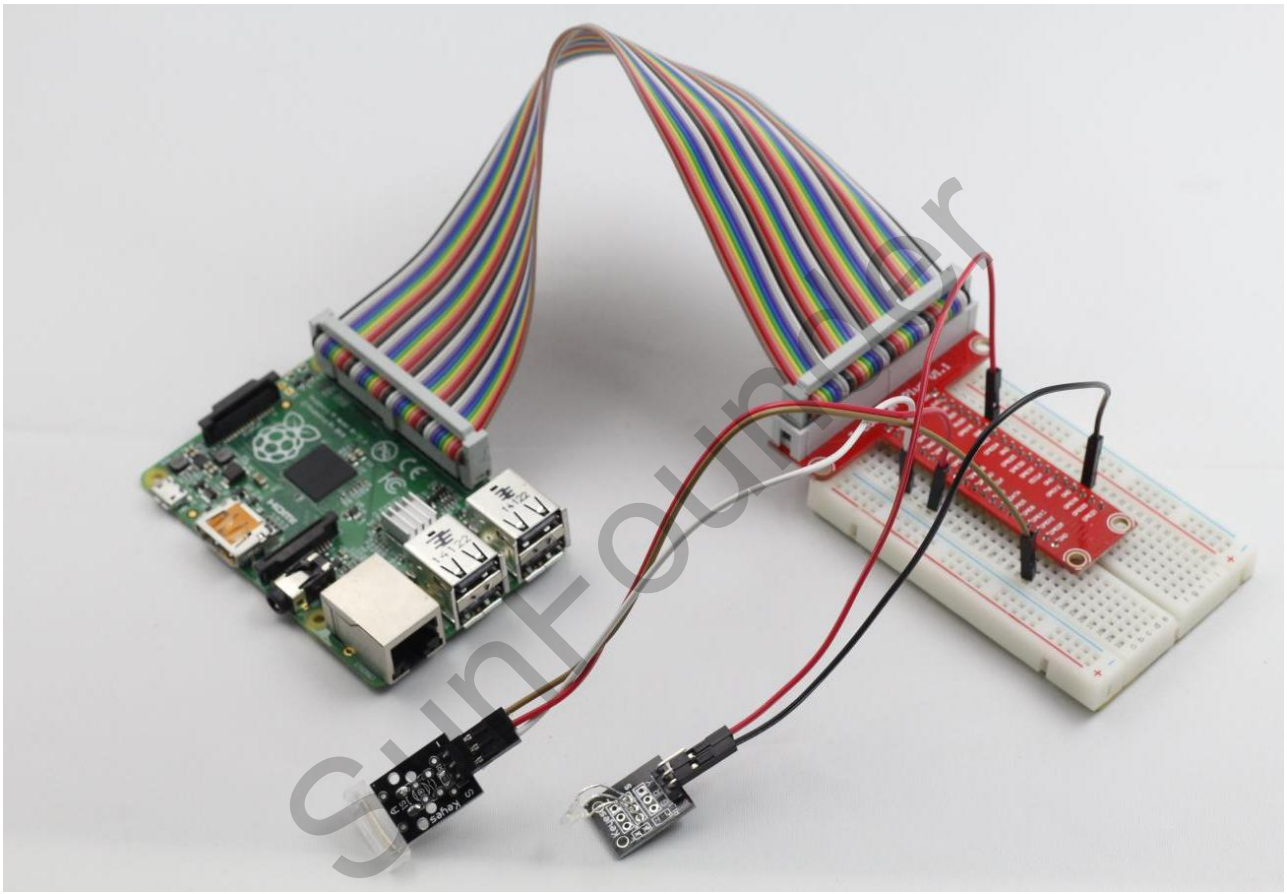
Step 3: Compile

```
gcc knockSensor.c -lwiringPi
```

Step 4: Run

```
./a.out
```

Shake the sensor. Then pin S will output low level and you will see "**Detected knocking**" displayed on the screen, and the LED will light up.



Lesson 6 Infrared Transmitter

Introduction

An infrared transmitter (as shown below) is a type of remote control devices. It can emit rays within a certain range through infrared transmitting tube so as to control signals. Infrared transmitter is widely used in consumer electronics, industry and communication, etc.

Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Infrared transmitter module
- Several jumper wires

Experimental Principle

Connect the IR transmitter module with the Raspberry Pi and make the module emit infrared rays by programming. Since infrared ray is invisible to naked eyes, you can use a camera to observe.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi		Infrared Transmitter
GPIO0	----------	S
GND	----------	GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/08_infrared/infrared.c)

Step 3: Compile

```
gcc  infrared.c  -lwiringPi
```

Step 4: Run

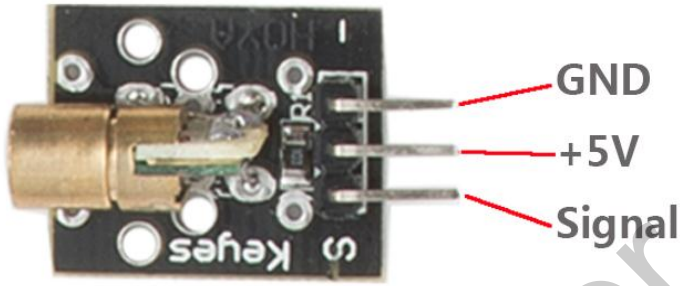
```
./a.out
```

Now you can see by a camera the infrared diode on the module emit infrared rays. (Note: Infrared rays are not visible, but can be captured by camera).

Lesson 7 Laser Transmitter

Introduction

Laser is widely used in medical treatment, military, and other fields due to its good directivity and energy concentration. The Laser Transmitter module (as shown below), as the name suggests, is a one that can emit laser.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Laser-transmitter module
- Several jumper wires

Experimental Principle

A laser is a device that emits light through a process of optical amplification based on the stimulated emission of electromagnetic radiation. Lasers differ from other sources of light because they emit light coherently.

Spatial coherence allows a laser to be focused to a tight spot, enabling applications like laser cutting and lithography, and a laser beam to stay narrow over long distances (collimation), enabling applications such as laser pointer. Lasers can also have high temporal coherence which allows them to have a very narrow spectrum, i.e., they only emit a single color of light. And its temporal coherence can be used to produce pulses of light—as short as a femtosecond.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi	Laser-transmitter
GPIO0	S
3.3V	+5v
GND	GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/09_laser/laser.c)

Step 3: Compile

```
gcc  laser.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

Here you can see the laser blinking constantly.

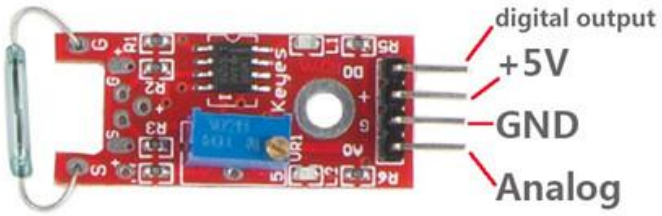
Note: DO NOT look directly at the laser head. It can cause great harm to your eyes.

Lesson 8 Reed Switch

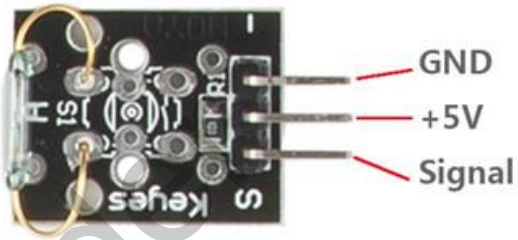
Introduction

A reed switch (as shown below) is also a sensor used to detect a magnetic field. Hall sensors are generally used to measure intelligent vehicle speed and count products on assembly lines, when reed switches are often to detect the existence of magnetic field.

There are two kinds of reed switch in this kit: **reed switch** and **mini reed**. They share the same principle.



Reed Switch



Mini Reed

Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Reed switch module
- 1 * Mini reed module
- 1 * Dual-color Common-Cathode LED module
- Several jumper wires

Experimental Principle

A reed switch is a type of line switch component that realizes control by magnetic signals. It induces by a magnet. The "switch" here means dry reed pipe, which is a kind of contact passive electronic switch component with the advantage of simple structure, small size, and convenient control. The shell of a reed switch is commonly a sealed glass pipe in which two iron elastic reed electroplates are equipped and inert gases are filled.

Normally, the two reeds made of special materials in the glass tube are separated. However, when a magnetic substance approaches the glass tube, the two reeds in the glass tube are magnetized to attract each other and get touched due to the magnetic force. As a result, the two reeds close the circuit connected with the nodes.

After the external magnetic force disappears, the two reeds will be separated with each other again because their like poles are placed near which intend to repel them apart, thus breaking the circuit. Therefore, as line switch components function by magnetic signals, they can be used to count stuff, restrict positions and so on. At the same time, it is widely used in a variety of communication devices.

Experimental Procedures

For Reed Switch

Step 1: Build the circuit

Reed switch connection: connect pin DO on Reed switch module to GPIO0 on Raspberry Pi; GND to GND; pin + to 3.3V

Raspberry Pi		Reed Switch
GPIO0	----------	DO
3.3V	----------	+
GND	----------	G

Dual-color LED module connection: connect pin S of the dual-color LED module to GPIO1 on Raspberry Pi; GND to GND; pin + to 3.3V

Raspberry Pi		Dual-color LED
GPIO1	----------	S
3.3V	----------	+
GND	----------	-

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/10_magicRing/magicRing.c)

Step 3: Compile

```
gcc  magicRing  -lwiringPi
```

Step 4: Run

```
./a.out
```

Put a magnet close to the reed switch. Then pin DO will output high level, "**Detected Magnetic Material**" will be displayed on the screen and the dual-color LED will light up.

For Mini Reed

Step 1: Build the circuit

Same connections with Reed switch except pin S of Mini Reed to GPIO0 of Raspberry Pi.

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/10_magicRing/magicRing.c)

Step 3: Compile

```
gcc  magicRing.c  -lwiringPi
```

Step 4: Run

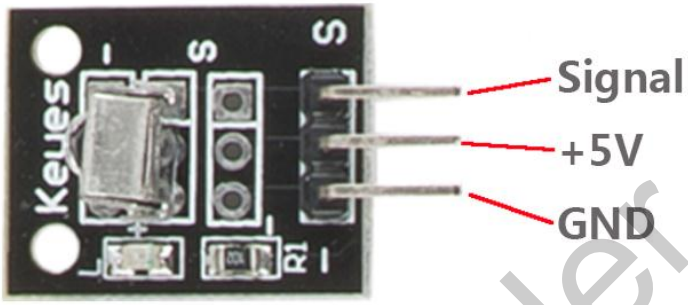
```
./a.out
```

Then you'll see the similar result to that of the reed switch.

Lesson 9 Infrared-Receiver

Introduction

An infrared-receiver is a component that receives infrared signals and can independently receive infrared ray and output signals compatible with TTL level. It's similar with a normal plastic-packaged transistor in size and it is suitable for all kinds of infrared remote control and infrared transmission.



Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Infrared-receiver module
- Several jumper wires

Experimental Principle

Infrared receiving head receives infrared signals.

Experimental Procedures

Step 1: Build the circuit

Infrared-Receiver Module	Raspberry Pi
S	GPIO0
+	3.3V
-	GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/29_irRecv/irRecv.c)

Step 3: Compile

```
gcc irRecv.c -lwiringPi
```

Step 4: Run

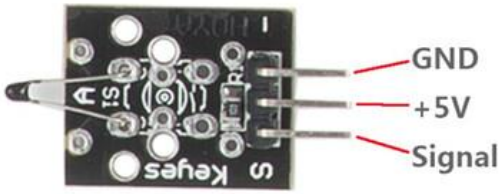
```
./a.out
```

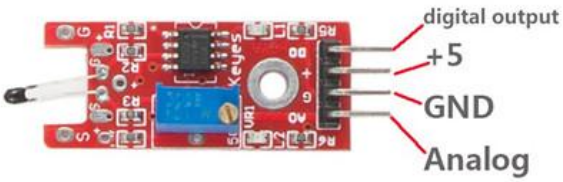
Here you can see the LED on the module light up when infrared rays are received. At the same time, infrared pulses will be printed on the screen.

Lesson 10 Analog Temperature Sensor

Introduction

Analog-temperature sensors are a type of devices to detect temperature changes. Their core component is a thermistor. There are two kinds of analog-temperature sensors in this kit (as shown below). In this lesson, we take the left one (try to use the other sensor yourself!).





Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Analog-temperature Sensor module
- 1 * ADC0832
- Several jumper wires

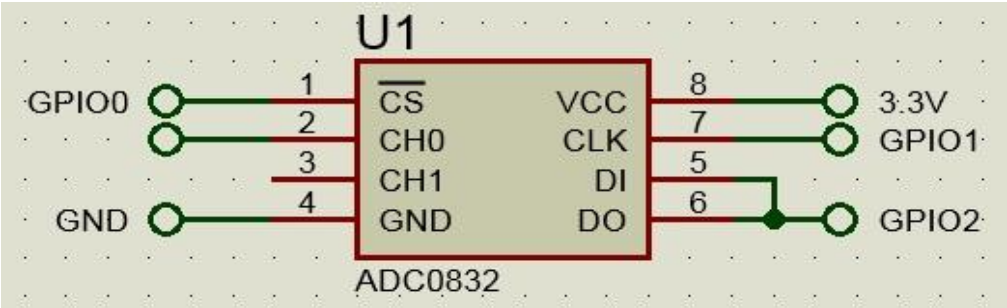
Experimental Principle

This module is developed based on the principle of thermistor, whose resistance varies significantly with ambient temperature changes. When the temperature increases, the resistance decreases; when the temperature decreases, the resistance increases. It can detect surrounding temperature changes in real time.

In this experiment, we need to use an analog-digital converter ADC0832 to convert analog signals into digital ones.

Experimental Procedures

Step 1: Build the circuit



Analog-temperature Sensor	ADC0832
S	CH0
+	3.3V
-	GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/11_analogTempSensor/ analogTempSensor.c)

Step 3: Compile

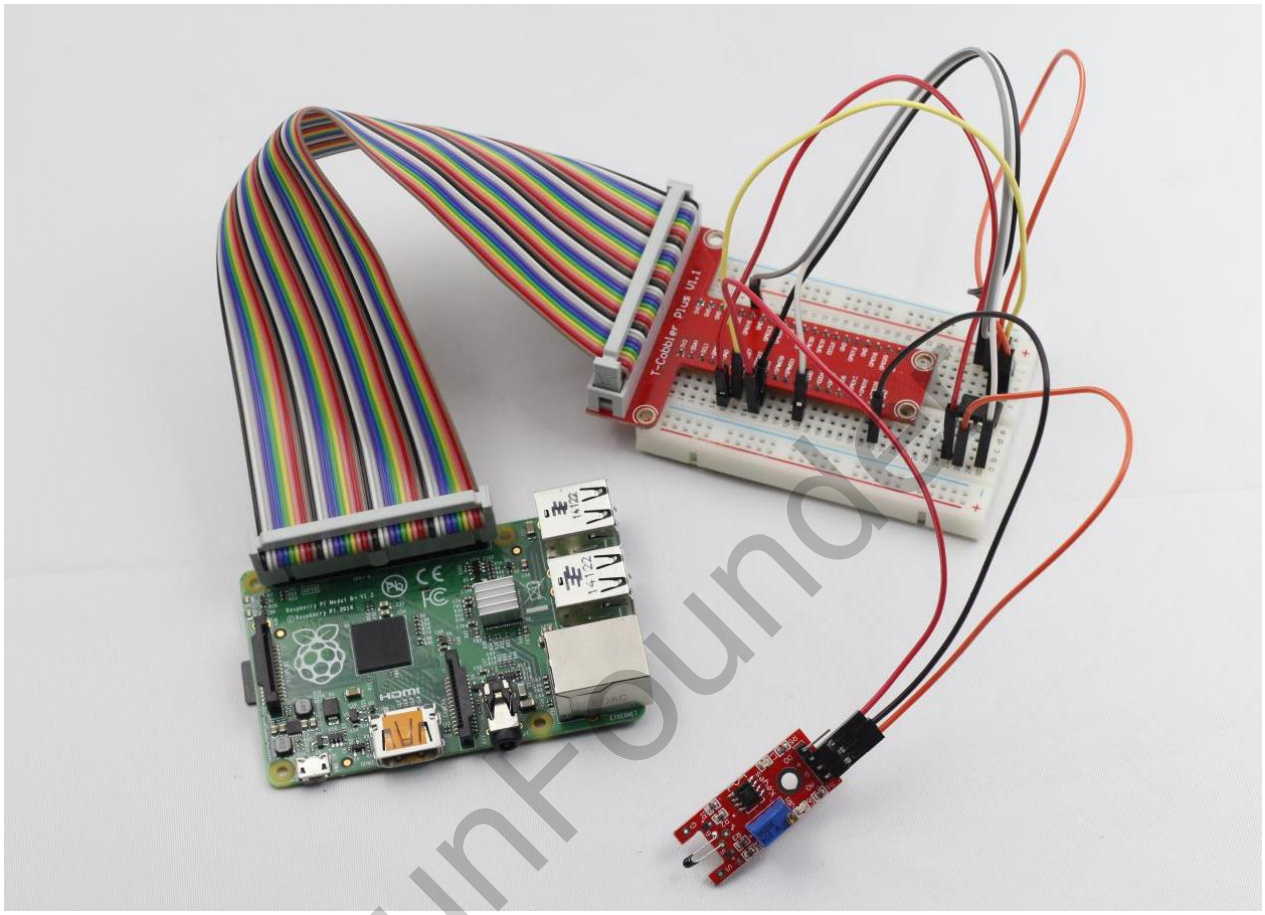
```
gcc analogTempSensor.c -lwiringPi
```

Step 4: Run

```
./a.out
```

Now, touch the thermistor on the module and you can see the current temperature value is displayed on the screen and changes accordingly.

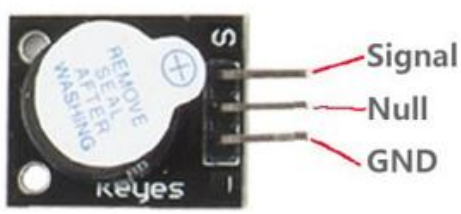
Compared with the black module on the left side in **Introduction**, the red one on the right only adds a digital output. You can adjust the threshold by the potentiometer on the module. When the output is higher than the threshold, the sensor will output HIGH; when it is lower than the threshold, the sensor will output LOW.



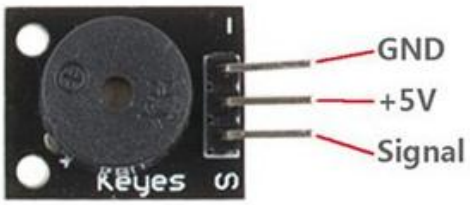
Lesson 11 Buzzer

Introduction

Buzzers can be categorized as active buzzers and passive ones (as shown below).



Active



Passive

Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Passive buzzer module
- 1 * Active buzzer module
- Several jumper wires

Experimental Principle

Place the pins of two buzzers face up and you can see the one with a green circuit board is a passive buzzer, while the other with a black tape, instead of a board, is an active buzzer, as shown below.

An active buzzer has a built-in oscillating source, so it will make sounds when electrified. But a passive buzzer does not have such source, so it will not beep if DC signals are used; instead, you need to use square waves whose frequency is between 2K and 5K to drive it. The active buzzer is often more expensive than the passive one because of multiple built-in oscillating circuits.

Experimental Procedures

Active Buzzer

Note: The active buzzer has built-in oscillating source, so it will make sounds as long as it is wired up. But it can only make sounds with a fixed frequency.

Step 1: Build the circuit

Raspberry Pi		Active buzzer
GND	----------	–
GPIO0	----------	S

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/12_buzzer/01_activeBuzzer /activeBuzzer.c)

Step 3: Compile

```
gcc activeBuzzer.c -lwiringPi
```

Step 4: Run

```
./a.out
```

Now you can hear the active buzzer beeps.

Passive Buzzer

Step 1: Build the circuit

Raspberry Pi		Passive Buzzer
3.3V	----------	–
GPIO0	----------	S

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/12_buzzer/02_passiveBuzzer/passiveBuzzer.c)

Step 3: Compile

```
gcc passiveBuzzer.c -lwiringPi
```

Step 4: Run

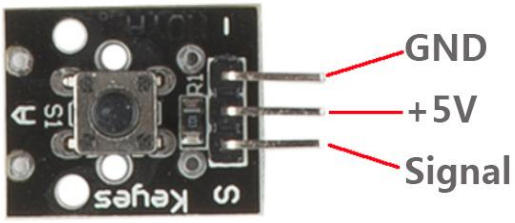
```
./a.out
```

Then you can hear the active buzzer beeping.

Lesson 12 Button

Introduction

In this lesson, we will learn how to use buttons.



Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Button module
- 1 * Dual-color Common-Cathode LED module
- Several jumper wires

Experimental Principle

Use a normally open button as an input device of Raspberry Pi. When the button is pressed, the General Purpose Input/Output (GPIO) connected to the button will turn into low level (0V). We can detect the state of the GPIO connected to the button through programming. That is, if the GPIO turns into low level, it means the button is pressed. Then you can run the corresponding code accordingly.

In this experiment, we will print a string on the screen and control an LED.

Experimental Procedures

Step 1: Build the circuit

Button Module		Raspberry Pi
S	----------	GPIO0
+	----------	3.3V
-	----------	GND

Dual-color LED module connection: connect pin R on dual-color LED module to GPIO1 of the Raspberry Pi; GND to GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/13_button/button.c)

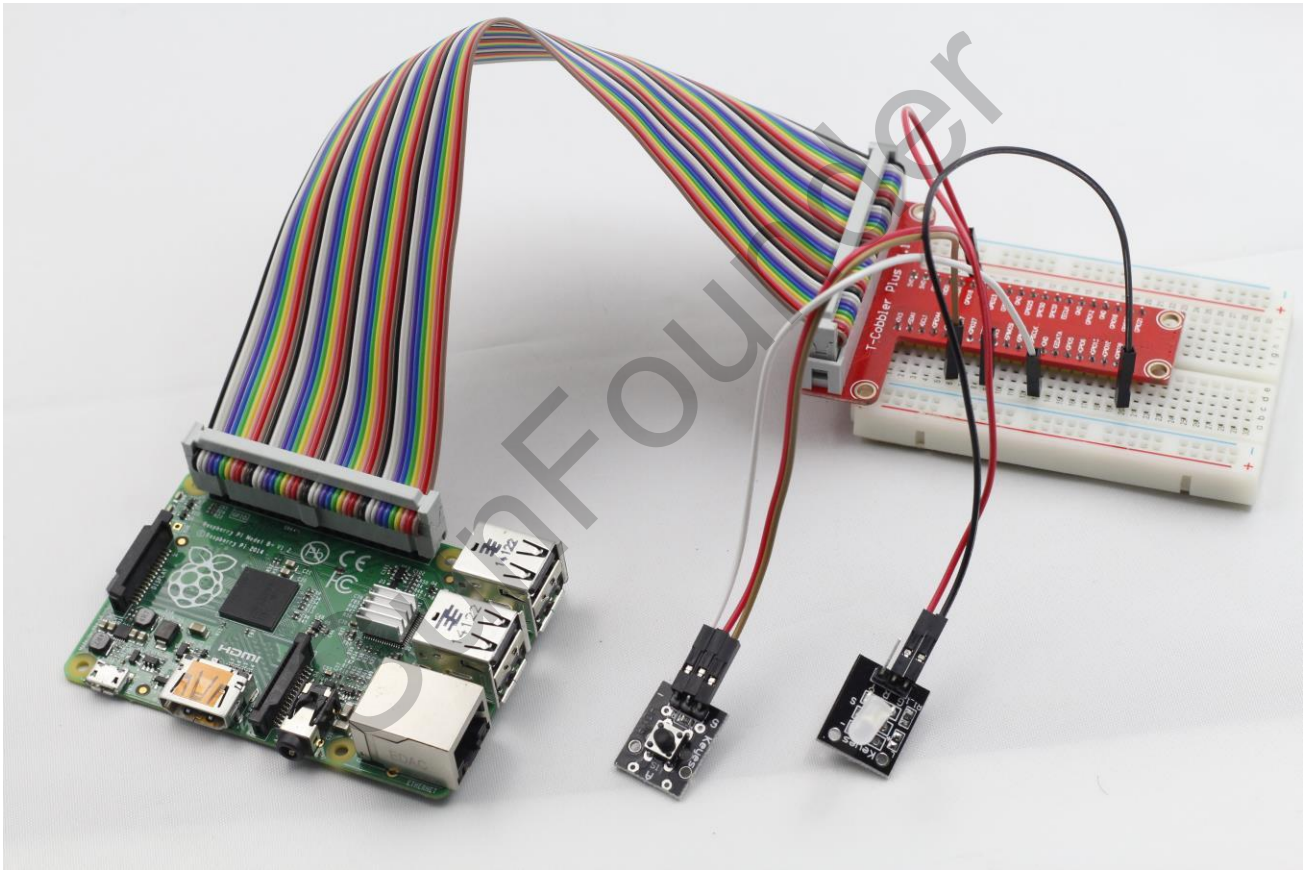
Step 3: Compile

```
gcc  button.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

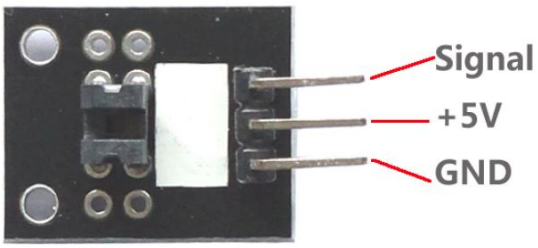
Press the button and you will see a string "**Button is pressed**" displayed on the screen, and the state of the LED will be switched ON/OFF.



Lesson 13 Photo-interrupter

Introduction

A photo-interrupter is a sensor that arranges light-emitting component and light-receiving component face-to-face and packages them together. It applies the principle that light is interrupted when an object passes through the sensor. Therefore, photo-interrupters are widely used in speed measurement.



Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Dual-color Common-Cathode LED module
- 1 * Photo-interrupter module
- Several jumper wires

Experimental Principle

Basically a photo-interrupter consists of two parts: transmitter and receiver. The transmitter (e.g., an LED or a laser) emits light and then the light goes to the receiver. If that light beam between the transmitter and receiver is interrupted by an obstacle, the receiver will detect no incoming light even for a moment and the output level will change. In this experiment, we will turn an LED on or off by using this change.

Experimental Procedures

Step 1: Build the circuit

Photo-interrupter Module		Raspberry Pi
S	----------	GPIO0
+	----------	3.3V
-	----------	GND

Dual-color LED module connection: connect pin R on dual-color LED module to GPIO1 of the Raspberry Pi; GND to GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/14_lightBreak/lightBreak.c)

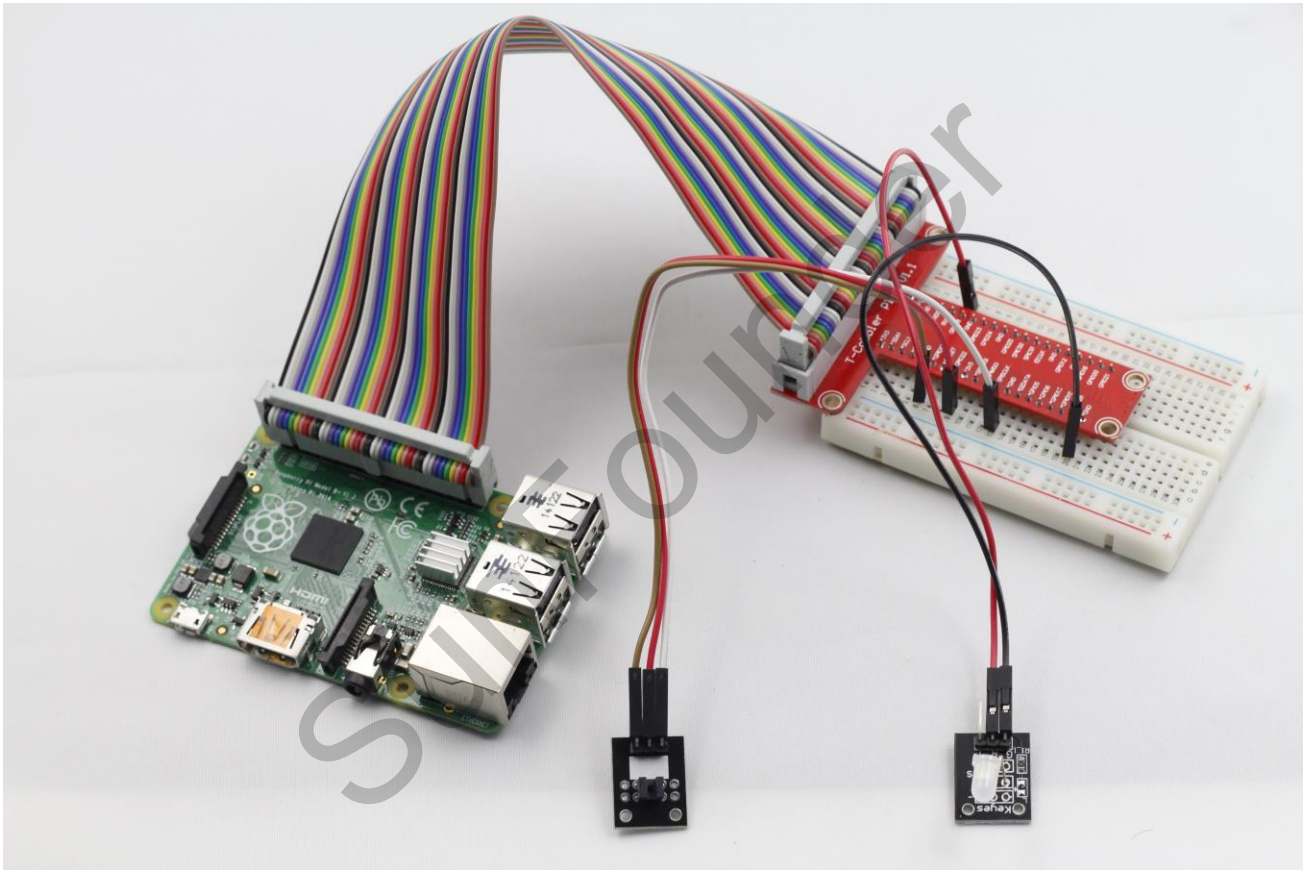
Step 3: Compile

```
gcc  lightBreak.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

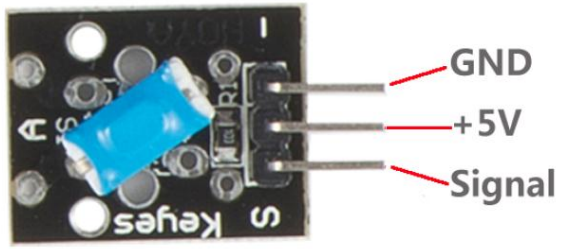
Block incident light with a piece of paper. The LED will light up and a string "**led on**" will be printed on the screen. Remove the paper and the LED will be off and a string "**led off**" will be printed on the screen.



Lesson 14 Tilt Switch

Introduction

The tilt switch sensor module is a ball tilt switch with a metal ball inside. It is used to detect inclinations of a small angle.



Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Dual-color Common-Cathode LED module
- 1 * Tilt-switch module
- Several jumper wires

Experimental Principle

The principle is very simple: the ball in the tilt switch moves with different angles of inclination to make triggering circuits. When it tilts towards either side, as long as the tilt degree and force meet the condition, the switch will be energized; thus, it will output low level signals.

Experimental Procedures

Step 1: Build the circuit

Tilt-Switch Module	Raspberry Pi
S	GPIO0
+	3.3V
-	GND

Dual-color LED module connection: connect pin R on dual-color LED module to GPIO1 on Raspberry Pi; GND to GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/15_tiltSwitch/tiltSwitch.c)

Step 3: Compile

```
gcc  tiltSwitch.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

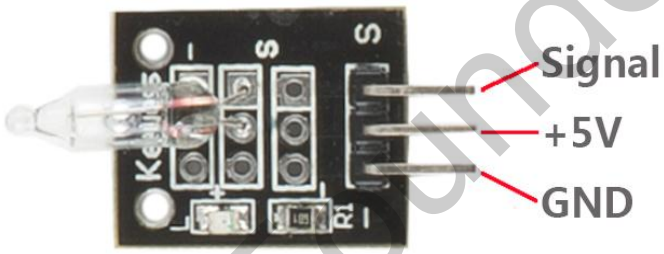
Tilt the switch and the state of LED will be switched ON/OFF.

Lesson 15 Mercury Switch

Introduction

Similar to a tilt switch, a mercury switch is used to detect slight inclinations of a large angle. A mercury switch (also known as a mercury tilt switch) is a switch which opens and closes an electrical circuit through a small amount of liquid mercury.

Mercury switches have one or more sets of electrical contacts in a sealed glass envelope which contains a bead of mercury. The envelope may also contain air, an inert gas, or a vacuum. Gravity is constantly pulling the drop of mercury to the lowest point in the envelope. When the switch is tilted in the appropriate direction, the mercury touches a set of contacts, thus completing the electrical circuit through those contacts. Tilting the switch the opposite direction causes the mercury to move away from that set of contacts, thus breaking that circuit.



Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 1 * Mercury switch module
- Several jumper wires

Experimental Principle

When you tilt the switch, the circuit will conduct and the LED will light up. When you place it horizontally, the LED will go out.

Note: Mercury is harmful to human body and environment. Thus, please BE CAREFUL when using a mercury switch in case of glass breaking. It should also be properly handled if it's no longer used.

Experimental Procedures

Step 1: Build the circuit

Mercury Switch	Raspberry Pi
S	GPIO0
+	3.3V
-	GND

Dual-color LED module connection: connect pin R on dual-color LED module to GPIO1 on Raspberry Pi; GND to GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/28_mercurySwitch/ mercury.c)

Step 3: Compile

```
gcc  mercury.c  -lwiringPi
```

Step 4: Run

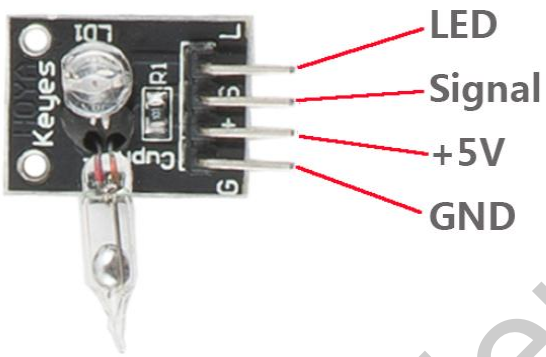
```
./a.out
```

Now, tilt the mercury switch and the state of LED will be switched ON/OFF.

Lesson 16 Magic Cup

Introduction

There are two same Magic Cup modules in this kit, and each adds a separate LED based on the mercury switch. You may learn the application of one module and then try to apply two modules together to make one dim when at the same time the other brightens.



Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Network cable (or USB wireless network adapter)
- 2 * Magic cup module
- Several jumper wires

Experimental Principle

Tilt the module and the LED onside will dim gradually. The working principle is similar to that of the Mercury Switch.

Experimental Procedures

Step 1: Build the circuit

Magic Cup	Raspberry Pi
L	GPIO1
S	GPIO0
+	3.3V
G	GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/29_magicCup/magic.c)

Step 3: Compile

```
gcc  magic.c  -lwiringPi
```

Step 4: Run

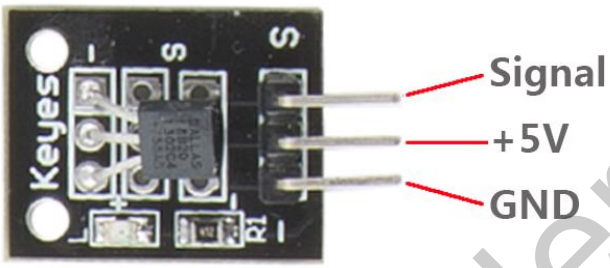
./a.out

Now, tilt the magic cup and the LED on one module will get dim.

Lesson 17 DS18B20 Temperature Sensor

Introduction

Temperature Sensor DS18B20 is a commonly used digital temperature sensor featured with small size, low-cost hardware, strong anti-interference capability and high precision. The digital temperature sensor is easy to wire and can be applied a various occasions after packaging. Different from conventional AD collection temperature sensors, it uses a 1-wire bus and can directly output temperature data.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * DS18B20 Temperature Sensor module
- Several jumper wires

Experimental Principle

With a unique single-wire interface, DS18B20 requires only one pin for a two-way communication with a microprocessor. It supports multi-point networking to measure multi-point temperatures. Eight DS18B20s can be connected at most, because too many of them will consume too much of the power supply and cause low voltage thus instability of signal transmission.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi		DS18B20 Module
GPIO7	----------	S
+5V	----------	5V
GND	----------	-

Step 2: Upgrade your kernel

sudo apt-get update

sudo apt-get upgrade

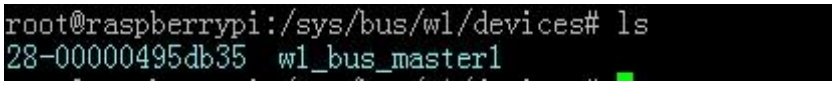
Step 3: Mount the device drivers and confirm whether the device is effective or not

sudo modprobe w1-gpio

sudo modprobe w1-therm

cd /sys/bus/w1/devices/

ls

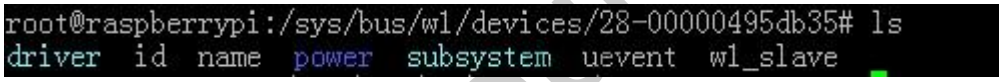


28-00000495db35 is an external temperature sensor device, but it may vary with every client. This is the serial number of your DS18b20.

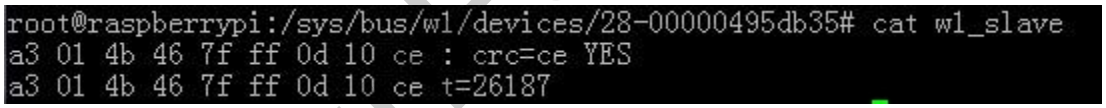
Step 4: Check the current temperature

cd 28-00000495db35

ls



cat w1_slave



The second line **t=26187** is current temperature value. If you want to convert it to degree Celsius, you can divide by 1000, that is, the current temperature is 26187/1000=26.187 ℃.

Step 5: After the device is confirmed to work normally, you can read the temperature by programming

Reference code: path/16_ds18b20/ds18b20_2.c

Step 6: Compile

gcc ds18b20_2.c

Step 7: Run

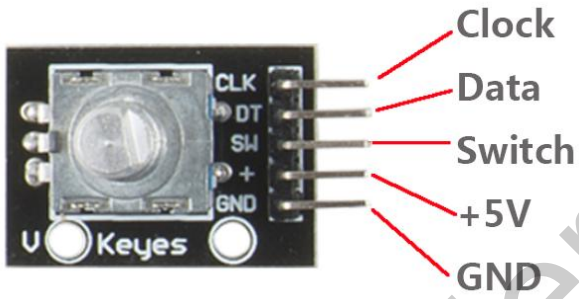
./a.out

Now, you should see the current temperature value displayed on the screen.

Lesson 18 Rotary Encoder

Introduction

A rotary encoder is an electro-mechanical device that converts the angular position or motion of a shaft or axle to an analog or digital code. Rotary encoders are usually placed at the side which is perpendicular to the shaft. Rotary encoders act as sensors for detecting angle, speed, length, position and acceleration in automation field.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Rotary Encoder module
- Jumper wires

Experimental Principle

Most rotary encoders have 5 pins with three functions of turning left, turning right and pressing down. Pin 4 and Pin 5 are switch wiring terminals used to press. They have no difference with buttons previously used, so we will no longer discuss them in this experiment. Pin 2 is generally connected to ground. Pin 1 and Pin 3 are first connected to pull-up resistor and then to microprocessor. In this experiment, they are connected to GPIO0 and GPIO1 of Raspberry Pi. When we rotate left and right, there will be pulse inputs in pin 1 and pin 3.

The figure shows, if both GPIO0 and GPIO1 are at high level, it indicates the switch rotates clockwise; if GPIO0 is at high level but GPIO1 is at low level, it indicates the switch rotates anti-clockwise. Therefore, during programming, you only need to check the state of pin 3 when pin 1 is at high level, then you can tell whether the switch is rotates clockwise or anti-clockwise.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi		Rotary Encoder
GPIO0	----------	CLK
GPIO1	----------	DT
3.3V	----------	+
GND	----------	GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/17_RotaryEncoder/ rotaryEncoder.c)

Step 3: Compile

```
gcc  rotaryEncoder.c -lwiringPi
```

Step 4: Run

```
./a.out
```

Now, gently spin the knob of the rotary encoder to change the value of the variable in the above program, and you will see the value printed on the screen. When you spin it clockwise, the value will increase; when counterclockwise, the value will decrease.

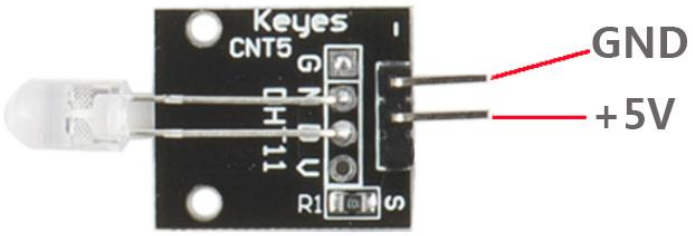
Summary

Through this lesson, you have been familiar with the principle of rotary encoder and preliminarily mastered the interruption programming way of Raspberry Pi. Interruption is a very important concept and is widely used in modern computers. Interruption greatly reduces the CPU load.

Lesson 19 7-Color Auto-flash LED

Introduction

On the 7-Color Auto-flash LED module, the LED can automatically flash built-in colors after power on. It can be used to make quite fascinating light effects.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * 7-color auto-flash LED module
- Jumper wires

Experimental Principle

When it is power on, the 7-color auto-flash LED will flash built-in colors.

Experimental Procedures

Build the circuit

7-color auto-flash LED Module	Raspberry Pi
S	5V
-	GND

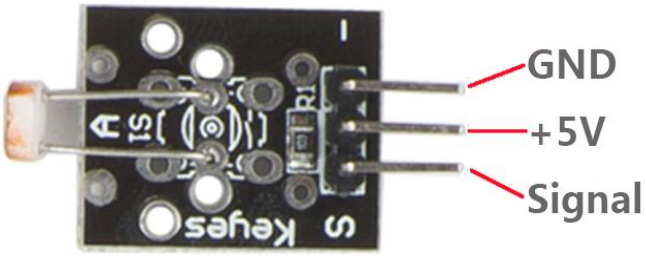
Note: Here just use the Raspberry Pi for power supply.

Now, you can see the 7-color auto-flash LED flashing seven colors.

Lesson 20 Photoresistor

Introduction

A photoresistor is a light-controlled variable resistor. The resistance of a photoresistor decreases with increasing incident light intensity.



Components

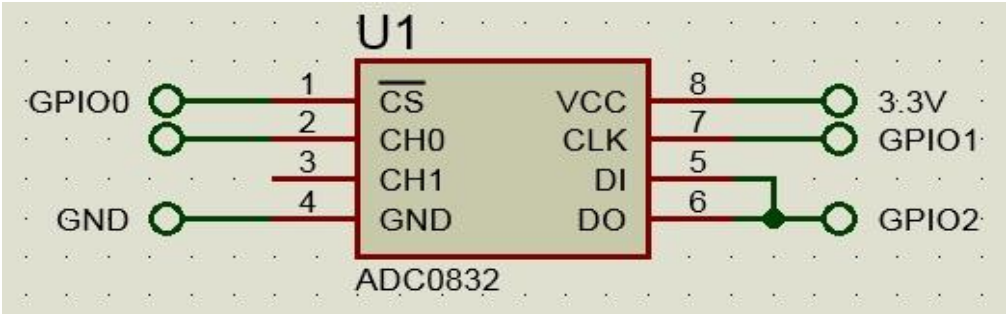
- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * ADC0832
- 1 * Photoresistor module
- Several jumper wires

Experimental Principle

With light intensity increasing, the resistance of a photoresistor will decrease. Thus the output voltage changes. Analog signals collected by the photoresistor will be converted to digital signals through ADC0832. Then these digital signals are transmitted to Raspberry Pi and printed on the screen.

Experimental Procedures

Step 1: Build the circuit



Connect pin S on the photoresistor module to pin CH0 of ADC0832.

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/19_photoResistor/photoResistor.c)

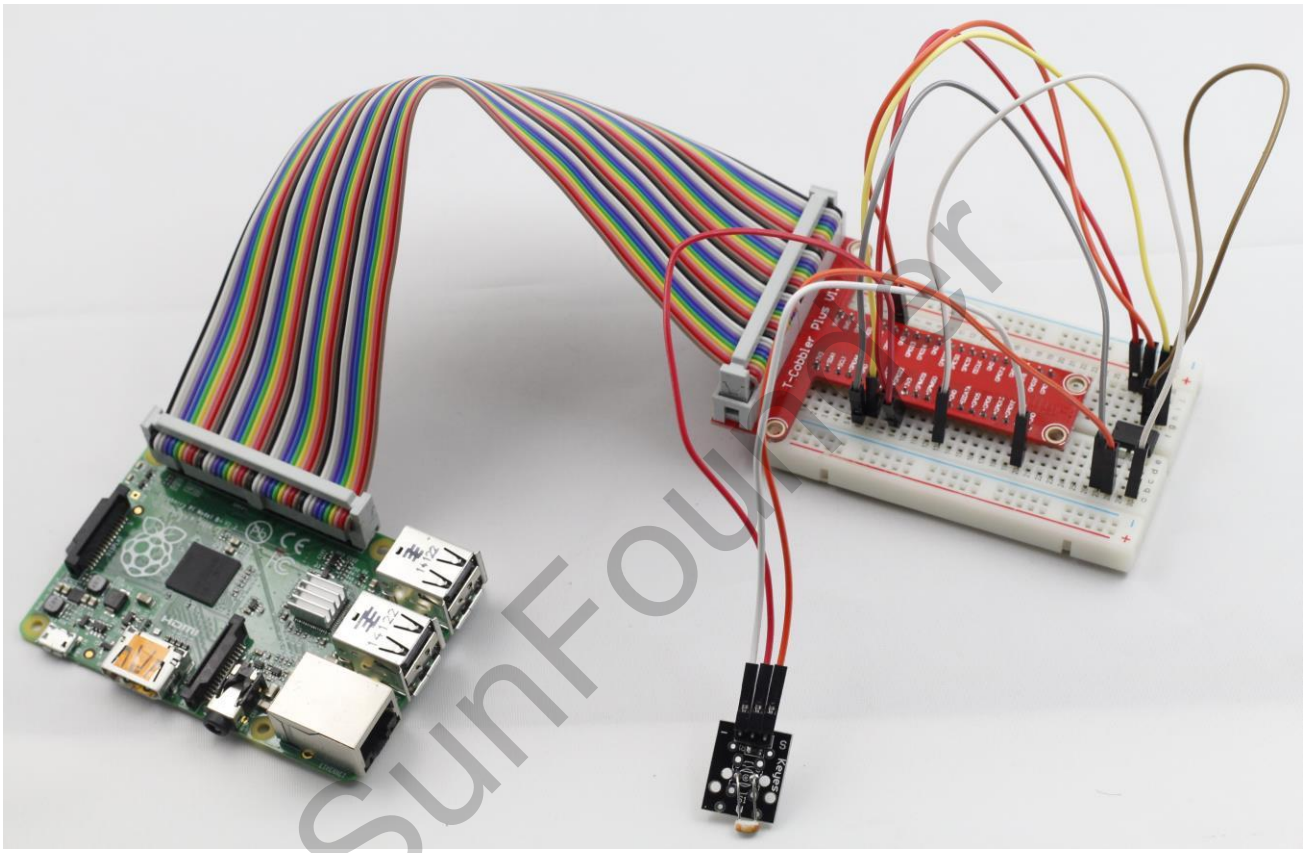
Step 3: Compile

```
gcc photoResistor.c -lwiringPi
```

Step 4: Run

```
./a.out
```

Now, cover the photoresistor with a cloth or your palm to change the light intensity. Then the value printed on the screen will change accordingly.

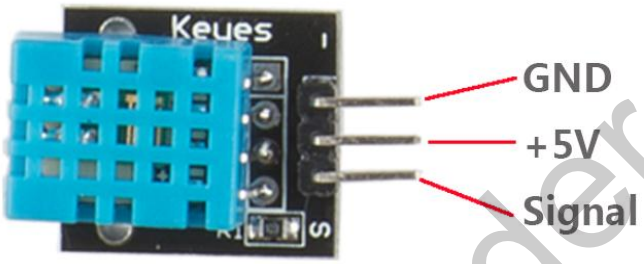


Lesson 21 DHT11 Humiture Sensor

Introduction

The digital temperature and humidity sensor DHT11 is a composite sensor that contains a calibrated digital signal output of temperature and humidity. The technology of a dedicated digital modules collection and the temperature and humidity sensing technology are applied to ensure that the product has high reliability and excellent long-term stability.

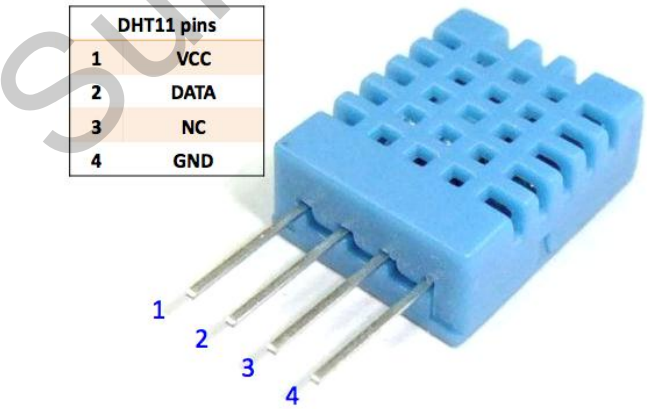
The sensor includes a resistive sense of wet component and an NTC temperature measurement device, and is connected with a high-performance 8-bit microcontroller.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * DHT11 module
- Several jumper wires

Experimental Principle



Only three pins are available for use: VCC, GND, and DATA. The communication process begins with the DATA line sending start signal to DHT11, and DHT11 receives the signal and returns an answer signal, then the host receives the answer signal and begins to receive 40-bit humiture data (8-bit humidity integer + 8-bit humidity decimal + 8-bit temperature integer + 8-bit temperature decimal + 8-bit checksum). For more information, please refer to the datasheet of DHT11.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi		DHT11 Humiture Sensor
GPIO0	----------	S
3.3V	----------	+
GND	----------	-

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/20_DHT11/DHT11.c)

Step 3: Compile

```
gcc  DHT11.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

Now, you can see humidity and temperature values printed on the screen.

```
root@raspberrypi:/home/Rpi_SensorKit/01_DHT11# ./DHT11
Use GPIO0 to read data!
Enter OS-------
Sorry! Sensor dosent ans!
Congratulations ! Sensor data read ok!
RH :34.0
TMP:54.0
Congratulations ! Sensor data read ok!
RH :34.0
TMP:27.0
Congratulations ! Sensor data read ok!
RH :35.0
TMP:27.0
```

Lesson 22 Obstacle Avoidance Sensor

Introduction

An obstacle avoidance module (as shown below) uses infrared reflection principle to detect obstacles. When there is no object ahead, infrared-receiver cannot receive signals; when there is an obstacle in front, it will block and reflect the infrared light, and then the infrared-receiver can receive related signals.

Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Obstacle Avoidance Sensor module
- Several jumper wires

Experimental Principle

An obstacle avoidance sensor mainly consists of an infrared-transmitter, an infrared-receiver and a potentiometer. According to the reflecting character of an object, if there is no obstacle, emitted infrared ray will weaken with the transmission distance and finally disappear. If there is an obstacle, when the infrared ray encounters it, the ray will be reflected back to the infrared-receiver. Then the infrared-receiver detects this signal and confirms an obstacle existing in front.

Note: The detection distance of the infrared sensor is adjustable - you may adjust it by the potentiometer on the module.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi		Obstacle Avoidance Sensor
GPIO0	----------	OUT
3.3V	----------	+
GND	----------	GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/21_obstacleAvoidence/ obstacle.c)

Step 3: Compile

```
gcc  obstacle.c  -lwiringPi
```

Step 4: Run

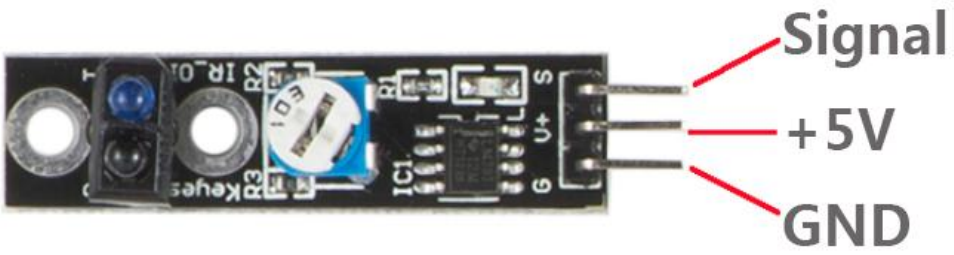
```
./a.out
```

Now, put an obstacle in front of the module, and a string "**Detected Barrier!**" will be printed on the screen.

Lesson 23 Tracking Sensor

Introduction

A tracking sensor (as shown below) has the same principle with an obstacle avoidance sensor except its smaller transmitting power.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Tracking sensor module
- Several jumper wires

Experimental Principle

When the infrared transmitter on the sensor emits rays to a piece of paper, if the rays shine on a white surface, they will be reflected and received by the receiver, and pin S will output low level; If the rays encounter black lines, they will be absorbed, thus the receiver gets nothing, and pin S will output high level.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi		Tracking Sensor
GPIO0	----------	S
3.3V	----------	V+
GND	----------	G

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/22_trackSensor/ trackSensor.c)

Step 3: Compile

```
gcc  trackSensor.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

Draw a black line on a white or light-color surface. When the tracking sensor encounters the black line, a string "**Black Line is detected**" will be printed on the screen.

Lesson 24 Microphone Sensor

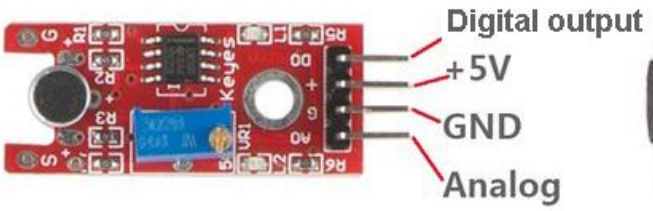
Introduction

There are two kinds of microphone sensor in this kit: microphone sensor and high-sensitive voice sensor (as shown below). The only difference between them is sensitivity. In this experiment, we will take the microphone sensor for example. You may try to apply the other sensor based on what you've got during the process.

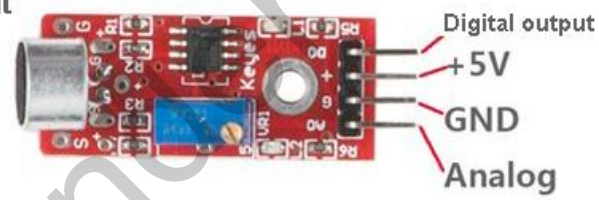
Both sensors have two outputs:

AO: analog output, to output voltage signals from microphone in a real-time manner

DO: When the sound intensity reaches a certain threshold, the sensor outputs high or low level (you can adjust the threshold by the potentiometer)



Microphone Sensor



High-sensitive Voice Sensor

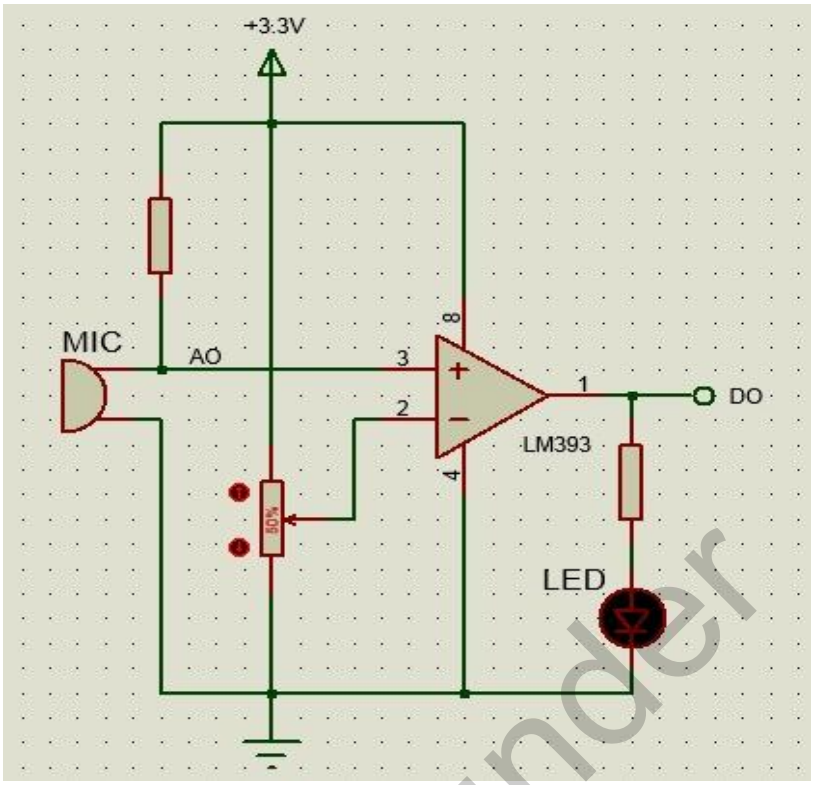
Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * ADC0832
- 1 * Network cable (or USB wireless network adapter)
- 1 * Microphone sensor module
- 1 * High-Sensitive voice sensor module
- Several jumper wires

Experimental Principle

Microphone can convert audio signal into electrical signal (analog quantity), then convert analog quantity into digital value by ADC and transfer it to MCU to process.

The schematic diagram:

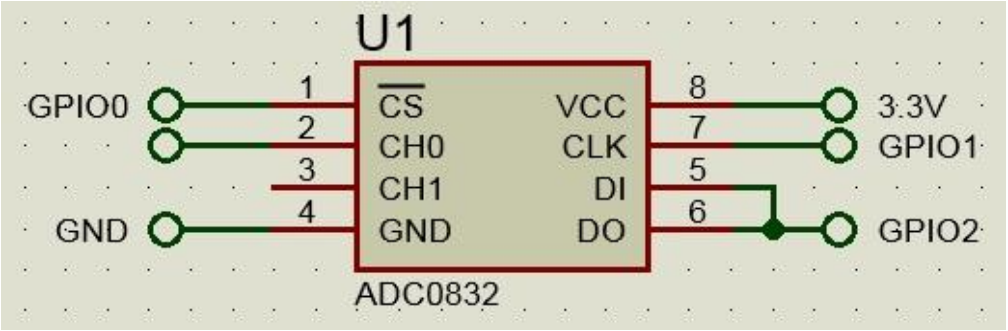


LM393 is a voltage comparator. When the voltage of the in-phase terminal (pin 3) is higher than that of the inverting terminal (pin 2), the output terminal (pin 1) will output high. Otherwise, it outputs low. First, adjust the potentiometer to make the voltage for pin 2 of LM393 less than 3.3V. When there is no voice input, the resistance of the microphone is very large. The voltage for pin 3 of LM393 is close to power supply voltage (3.3V), pin 1 outputs high and the LED is on; when there is voice input, the resistance of the microphone decreases, pin 1 outputs low and the LED is off. And connect pin 1 to IO of the SunFounder Uno board to detect sounds by programming.

ADC0832 is an 8-bit two-channel A/D conversion chip. Connect the output terminal (AO) to CH0 of ADC0832 so as to make real-time detection of the strength of voice signals.

Experimental Procedures

Step 1: Build the circuit



Connect pin DO on the microphone sensor module to pin GPIO3 on Raspberry Pi; pin AO of the module to CH0 of the ADC0832.

Step 2: Edit code and save (see path/Rpi_SensorKit_code/23_microPhoneSensor/ microPhone.c)

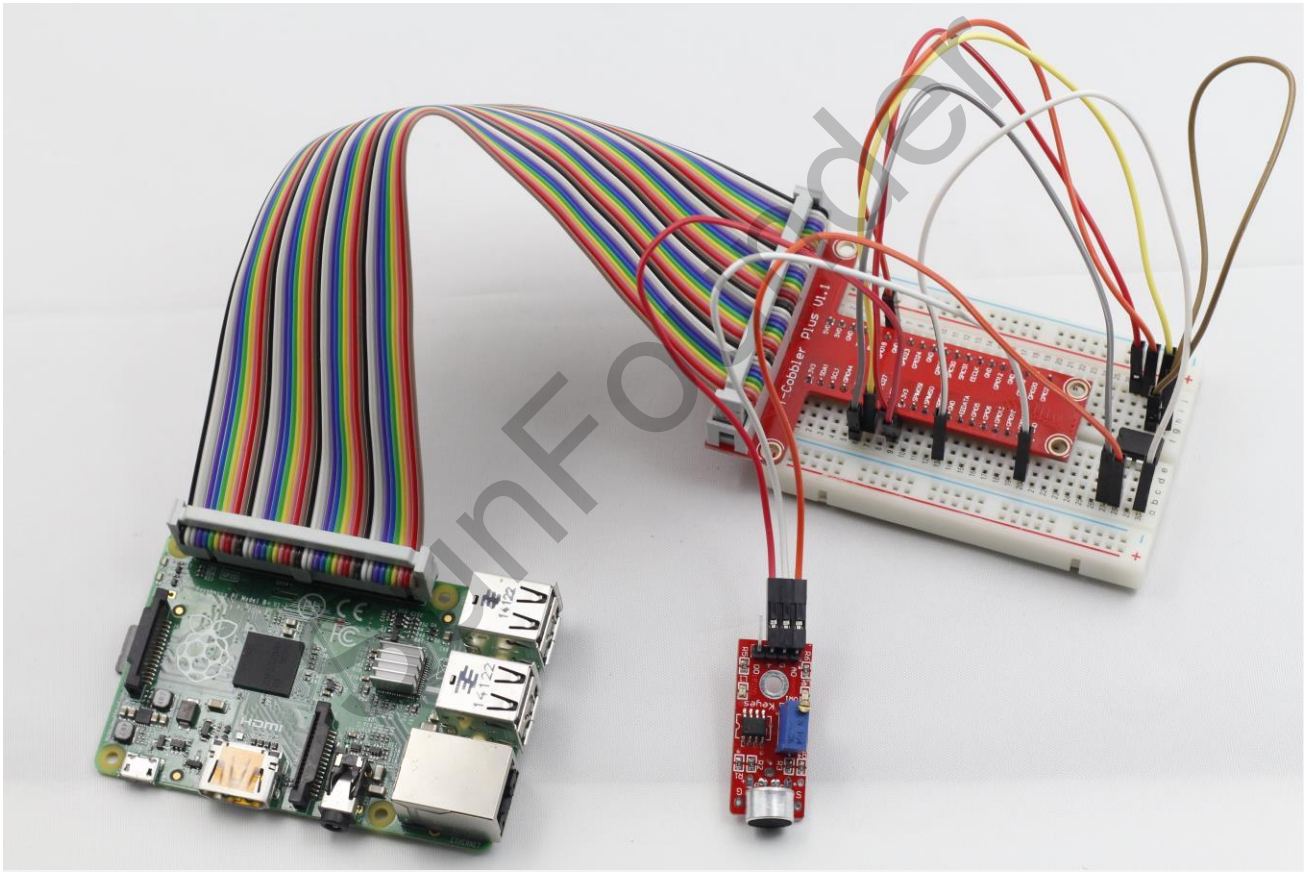
Step 3: Compile

```
gcc  microPhone.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

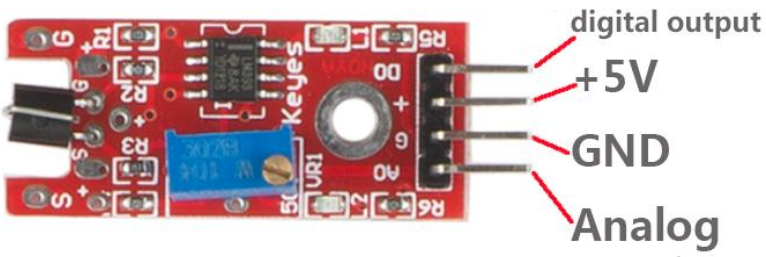
Now, speak or blow to the microphone, and you will see "**voice in**" and "**Current analog value: ***" printed on the screen.



Lesson 25 Metal Touch Sensor

Introduction

A metal touch sensor is a type of switch that only operates when it's touched by a charged body. It has a high-frequency transistor which can conduct electricity when receiving electromagnetic signals.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Metal touch sensor module
- Several jumper wires

Experimental Principle

In this experiment, touch the base electrode of a transistor with fingers to make it conduct electricity, for human body itself is a kind of conductor and an antenna that can receive electromagnetic waves in the air. These electromagnetic wave signals collected by human body are amplified by the transistor and processed by the comparator on the module to output steady signals.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi	Metal Touch Sensor
GPIO0	DO
3.3V	+
GND	G

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/24_metalTouchSensor/metalTouchSensor.c)

Step 3: Compile

```
gcc  metalTouchSensor.c  -lwiringPi
```

Step 4: Run

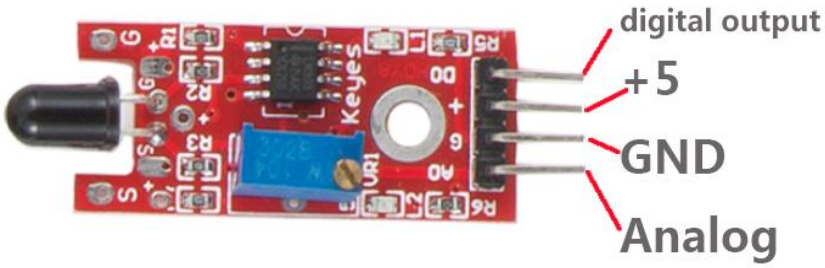
./a.out

Touch the base electrode of the transistor, and a string "**touched**" will be displayed on the screen.

Lesson 26 Flame Sensor

Introduction

A flame sensor module performs detection by capturing infrared wavelengths from flame. It can be used to detect and warn of flames.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Flame sensor module
- Several jumper wires

Experimental Principle

There are several types of flame sensors. In this experiment, we will use a far-infrared flame sensor. It can detect infrared light with a wavelength ranging from 700nm to 1000nm. A far-infrared flame probe converts the strength changes of the external infrared light into current changes. And then it converts analog quantities into digital ones.

In this experiment, connect pin DO of flame sensor module to a GPIO of Raspberry Pi to detect flame existence by programming.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi		Flame Sensor
GPIO0	----------	DO
3.3V	----------	+
GND	----------	G

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/25_flameSensor/ flameSensor.c)

Step 3: Compile

```
gcc  flameSensor.c  -lwiringPi
```

Step 4: Run

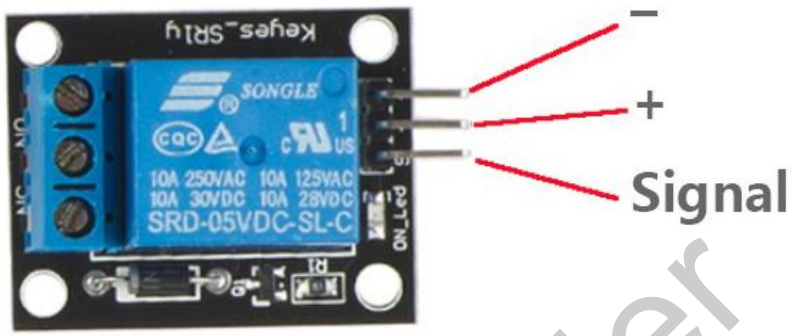
./a.out

Now, ignite a lighter in the range of 80 cm near the module and a string "**Detected Flame!**" will be displayed on the screen.

Lesson 27 Relay Module

Introduction

Relays are suitable for driving high power electric equipment, such as lights, electric fans and air conditioning. A relay can be used to control high voltages with a low voltage by connecting it to a Raspberry Pi.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * Relay module
- Several jumper wires

Experimental Principle

Connect the base electrode of the transistor to GPIO0. When we make GPIO0 output low level (0V) by programming, the transistor will conduct electricity because of current saturation. The normally open contact of the relay will be closed, while the normally closed contact of the relay will be opened; when we make it output high level (3.3V), the transistor will be cut off, and the relay will restore to the initial state.

Experimental Procedures

Step 1: Build the circuit

Raspberry Pi		Relay Module
GPIO0	----------	S
5V	----------	+
GND	----------	-

Step 2: Edit and save the code with vim (see path/Rpi_SensorKit_Code/26_relay/relay.c)

Step 3: Compile

```
gcc relay.c -lwiringPi
```

Step 4: Run

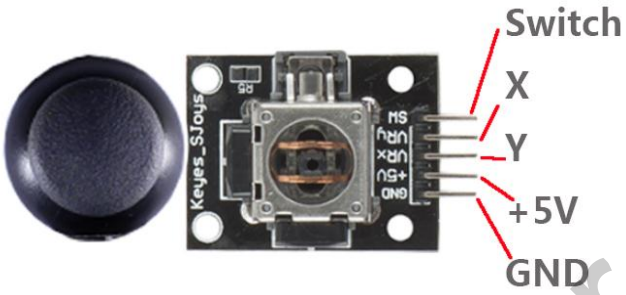
./a.out

Now, you can hear the ticktock. That's the normally closed contact opened and the normally open contact closed. You can attach a high voltage device to the module. For example, connect a bulb of 220V voltage to the output port of the relay module, and then the relay will act as an automatic switch.

Lesson 28 Joystick PS2

Introduction

A joystick is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling. Joysticks are often used to control video games and robots. A Joystick PS2 is used here.



Components

- 1 * Raspberry Pi
- 1 * Network cable (or USB wireless network adapter)
- 1 * ADC0832
- 1 * Joystick PS2 module
- Several jumper wires

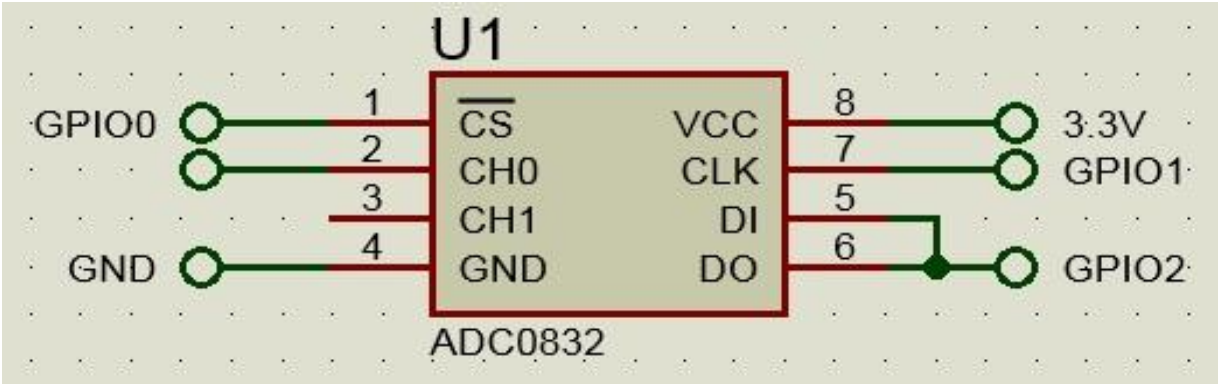
Experimental Principle

This module has two analog outputs (corresponding to X and Y biaxial offsets) and one digital output representing whether it is pressed on Z axis. The module integrates a power indicator and can display the working condition.

In this experiment, we connect pin X and Y to analog input ports of A/D convertor to convert analog quantity into digital quantity. Then program Raspberry Pi to detect the moving direction of the Joystick.

Experimental Procedures

Step 1: Build the circuit



Connect pin VRx on Joystick PS2 module to pin CH0 on ADC0832;

Connect pin VRy on Joystick PS2 module to pin CH1 on ADC0832;

Connect pin SW on Joystick PS2 module to GPIO3 on Raspberry Pi.

Step 2: Edit and save the code (see path/Rpi_SensorKit_Code/27_joyStickPS2/ joyStickPS2.c)

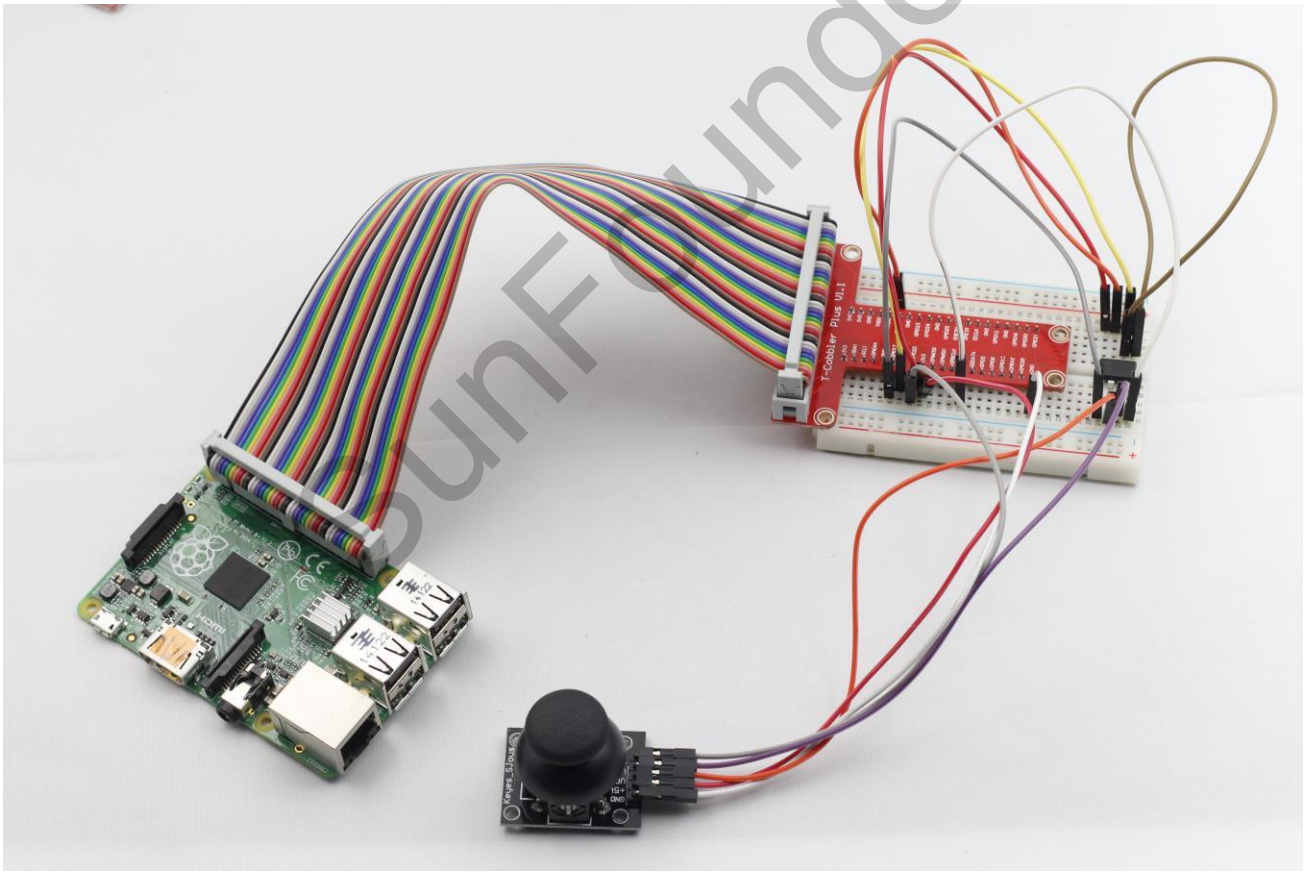
Step 3: Compile

```
gcc  joyStickPS2.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

Push the rocker up and a string "**Up**" will be displayed on the screen; pull it down, "**Down**" will be displayed; push it left, "**Left**" will be displayed; push it right, "**Right**" will be displayed; Press the button and "**Pressed**" will be displayed on the screen.



Lesson 29 MQ-2 Gas Sensor

Introduction

Gas Sensor MQ-2 is for flammable gas and smoke detection by measuring concentrations of combustible gases in the air. They are often used for detecting smoke and flammable gasses in households, industry or automobiles.

Components

- 1 * Raspberry Pi
- 1 * Active Buzzer module
- 1 * ADC0832
- 1 * MQ-2 Gas Sensor module
- Several jumper wires

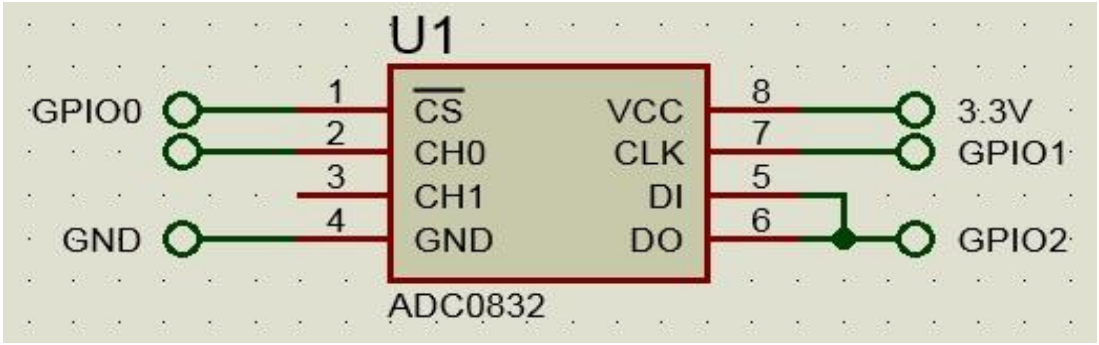
Experimental Principle

MQ-2 gas sensor applies SnO2 (an oxygen-deficient n-type semiconductor) which has a lower conductivity in the clear air as a gas-sensing material. In an atmosphere where there may be inflammable gases, the conductivity of the gas sensor raises along with the inflammable gas concentration increases. The higher the concentration is, the greater the conductivity will be, thus changing the output signal.

In this experiment, release a small amount of smoke or flammable gas around the sensor. Once the concentration of the gas exceeds a limit, the buzzer will beep.

Experimental Procedures

Step 1: Build the circuit. Connection between the Raspberry Pi and the ADC0832:



Connect pin OUT on MQ-2 gas sensor module to pin CH0 of the ADC0832;
Connect pin S on Buzzer module to GPIO3 of the Raspberry Pi.

Step 2: Edit and save the code (see path/Rpi_SensorKit_Code/28_ MQ-2/mq-2.c)
Step 3: Compile

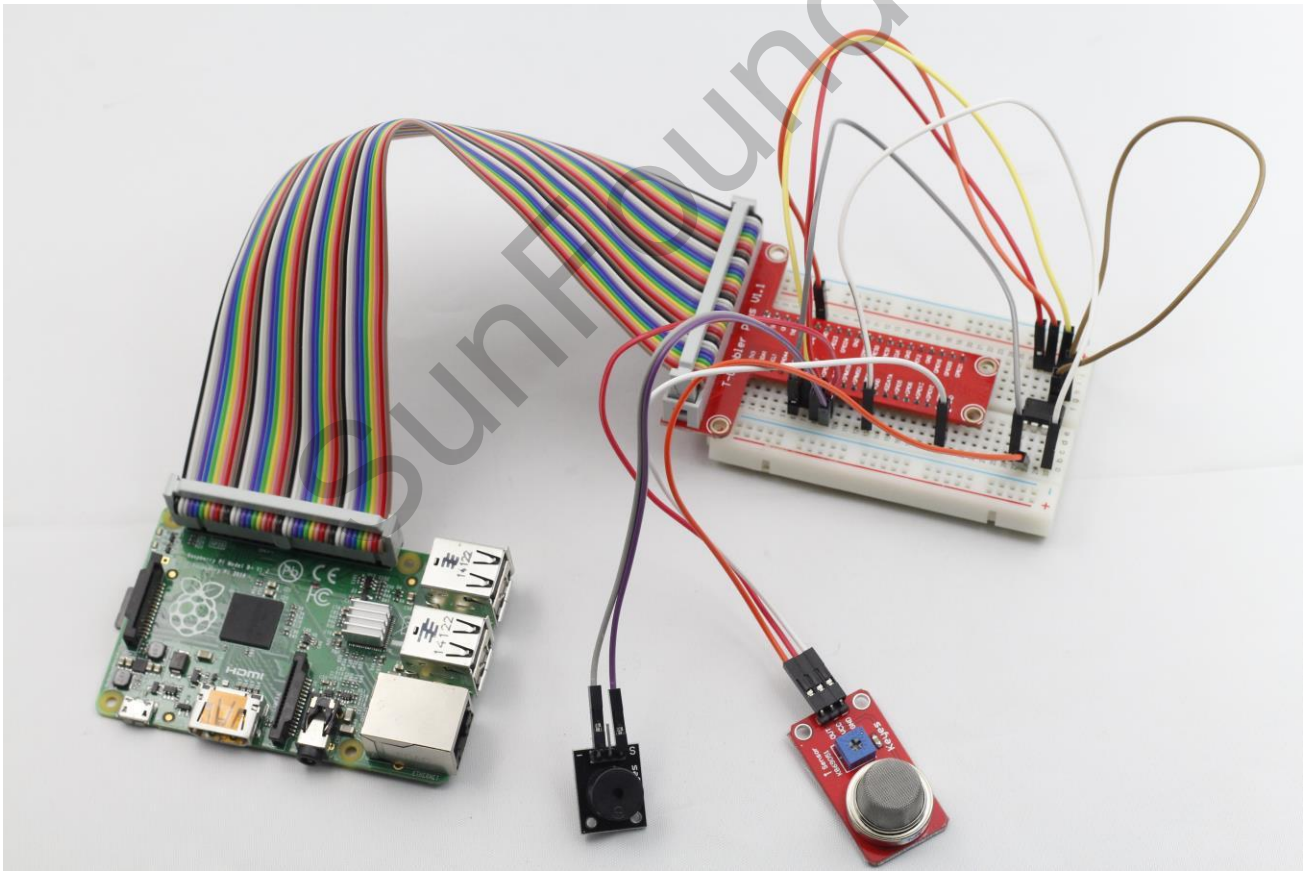
```
gcc  mq-2.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

Now, ignite a lighter or light a candle and release a small amount of smoke or gases into a flask (or other glass containers). Then place the sensor over the container, or invert the container and place the sensor at the mouth. When the concentration reaches the threshold, the buzzer will beep.

Note: It is normal that the gas sensor generates heat. Actually, the higher the temperature is, the sensor is more sensitive.



Lesson 30 Temperature Measurement System

Introduction

After having learnt so many modules, let's use several together to make a comprehensive experiment – use an RGB LED, a buzzer and a DS18B20 to build an interesting and useful temperature measurement system.

Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Active Buzzer
- 1 * RGB LED
- 1 * DS18B20
- Several jumper wires

Experimental Principle

When the ambient temperature is lower than the upper limit value, the buzzer will beep at a low frequency and the LED will flash blue; when the temperature is higher than lower limit value, the buzzer will beep at a relatively high frequency and the LED will flash red; when the temperature is between two values, the buzzer will keep silent and the LED will be green.

Note: The lower and upper limit values here can be defined and achieved by passing parameters to the main function.

Experimental Procedures

Step 1: Build the circuit

DS18b20 module connection:

DS18B20	Raspberry Pi
S	GPIO7

RGB LED connection:

RGB LED	Raspberry Pi
R	GPIO0
G	GPIO1
B	GPIO2

Buzzer module connection:

Buzzer Module	Raspberry Pi
-	GND
S	GPIO3

Step 2: Edit and save the code (see path/Rpi_SensorKit_Code/29_expand01/tempMonitor.c)

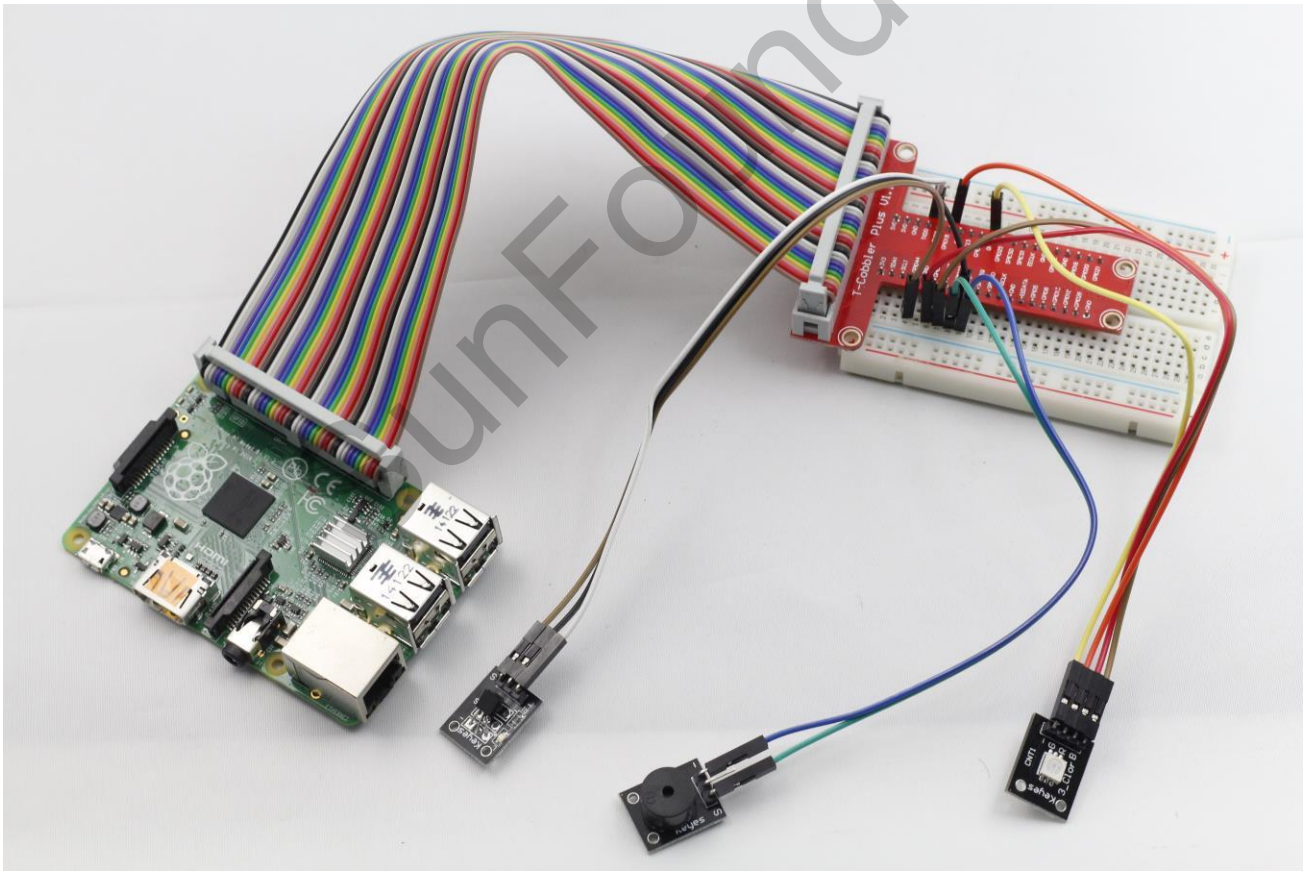
Step 3: Compile

```
gcc  tempMonitor.c  -lwiringPi
```

Step 4: Run

```
./a.out  25  30
```

Now, when the ambient temperature is lower than the lower limit value (25℃), the buzzer beeps at a lower frequency and the LED flashes blue; when it is higher than the upper limit value (30℃), the buzzer beeps at a relatively higher frequency and the LED flashes red. When it is between the two values, the buzzer is silent and the LED keeps green.



Lesson 31 Intelligent Temperature Measurement System

Introduction

Different from the last experiment, an ADC0832 and a Joystick PS2 are added here to make the measurement system more intelligent.

Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Buzzer
- 1 * RGB LED
- 1 * DS18B20
- 1 * ADC0832
- 1 * Joystick PS2
- Several jumper wires

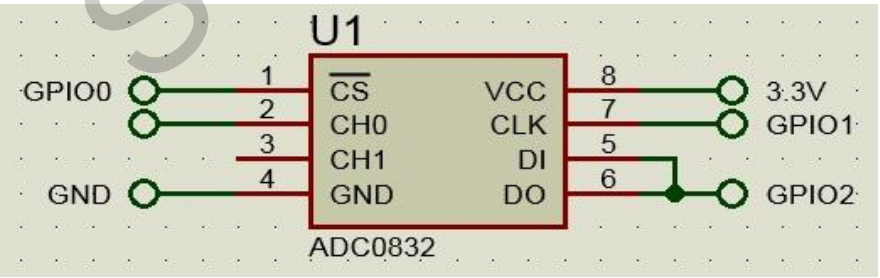
Experimental Principle

The principle is similar with that in the last experiment – only that you can adjust the lower and the upper limit value by the Joystick PS2 when programming.

Joystick PS2 has five operation directions: up, down, left, right and press-down. In this experiment, we will use left/right direction to control the upper limit value and up/down to control the lower limit value. If you press down the joystick, the system will log out.

Experimental Procedures

Step 1: Build the circuit



Raspberry Pi GPIO0 ---------- ADC0832 pin CS

Raspberry Pi GPIO1 ----------ADC0832 pin CLK

Raspberry Pi GPIO2 ---------- ADC0832 pin DIO

Joystick PS2 pin x ---------- ADC0832 pin CH0

Joystick PS2 pin y ----------ADC0832 pin CH1

Joystick PS2 pin z ---------- Raspberry Pi GPIO6

Raspberry Pi GPIO3 ---------- RGB LED pin R
Raspberry Pi GPIO4 ---------- RGB LED pin G
Raspberry Pi GPIO5 ---------- RGB LED pin B
Raspberry Pi GPIO8 ---------- Buzzer module pin S
Raspberry Pi GND---------- Buzzer module pin -

Step 2: Edit and save the code (see path/Rpi_SensorKit_Code/30_expand02/ tempMonitor.c)

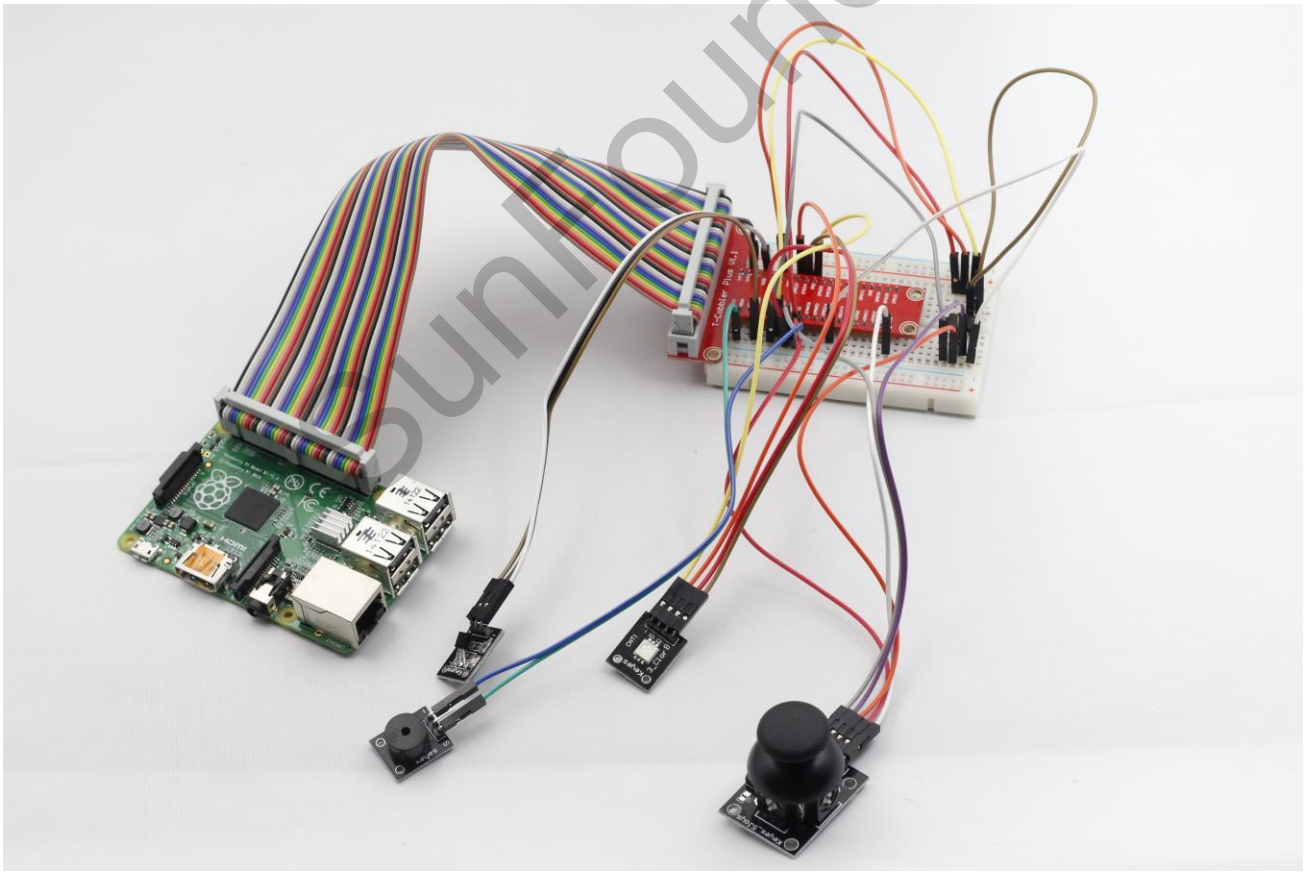
Step 3: Compile

```
gcc  tempMonitor.c  -lwiringPi
```

Step 4: Run

```
./a.out
```

Now, push the knob left and right to set the upper limit value, and up and down to set the lower limit value. If the ambient temperature reaches the upper or lower limit value, the buzzer will beep at different frequencies.



For Safe Use

All parts and devices in this kit should be powered appropriately in compliance with relevant regulations and standards applicable in the country of intended use.

The connection of unapproved external devices to the modules/boards in this kit may affect compliance or result in damage to the unit, for which we will not be responsible.

To avoid malfunction or damage to your circuit boards, please observe the following:

DO NOT expose it to water/moisture or place it on a conductive surface whilst in operation.

DO NOT expose it to heat from any source; the product is designed for reliable operation at normal ambient room temperatures.

Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.

PLEASE perform the connection or wiring based on the instructions in the manual or our website if you are not clear of the results.

Copyright Notice