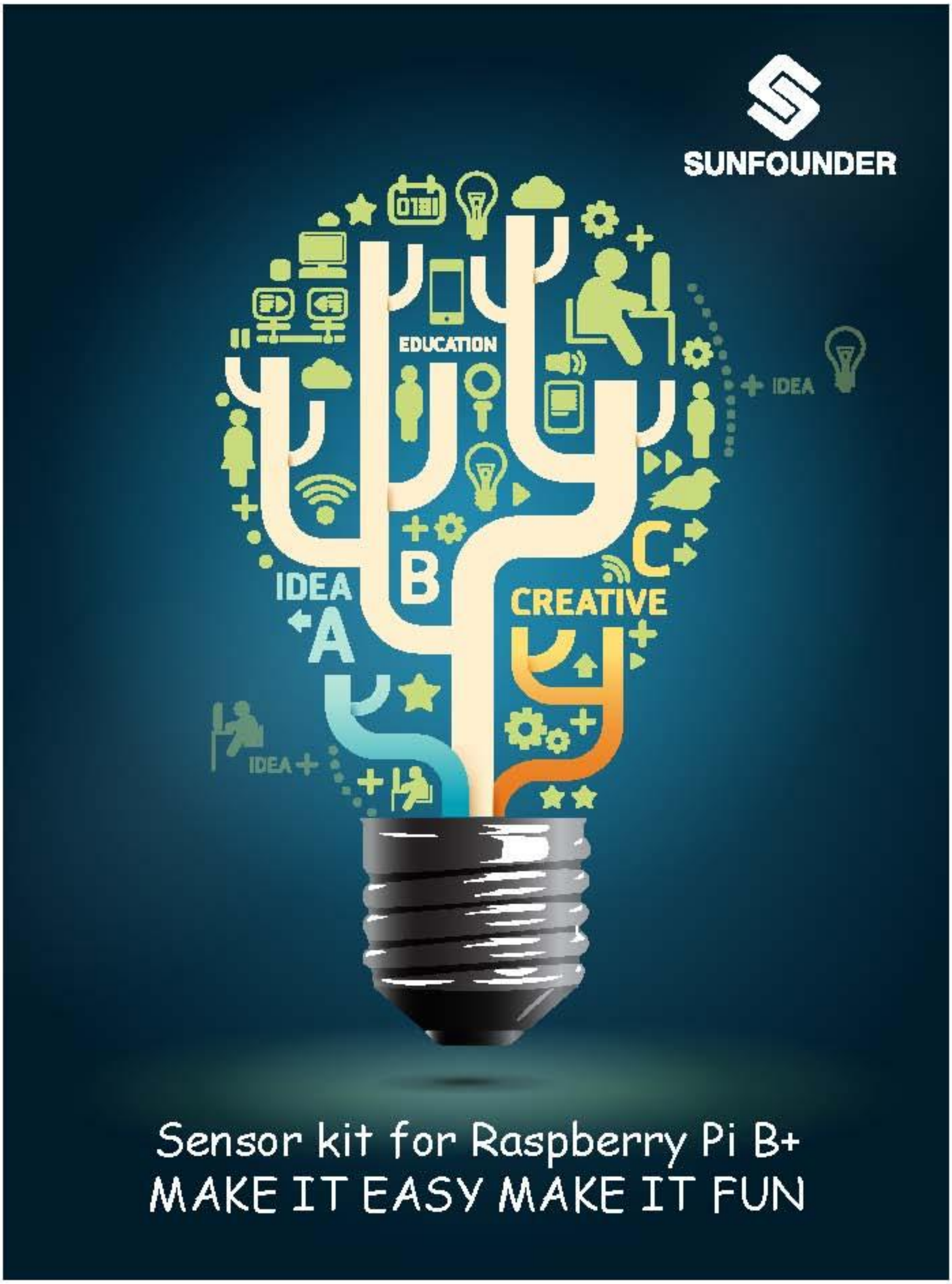

SUNFOUNDER
EDUCATION
IDEA
A
B
C
CREATIVE
IDEA
IDEA
Sensor kit for Raspberry Pi B+
MAKE IT EASY MAKE IT FUN

Component list

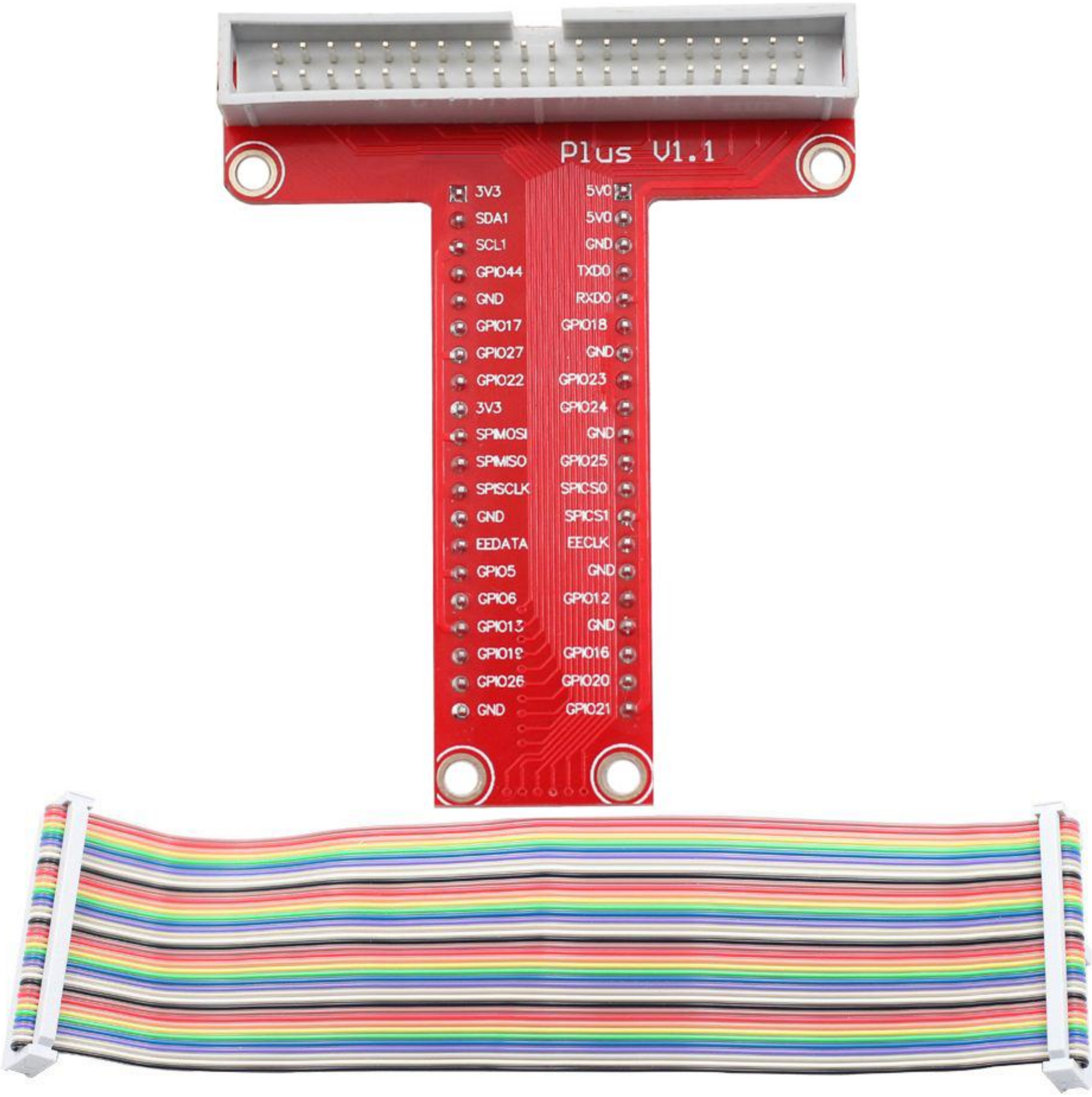
Analog Hall Sensor x2
Switch Hall Sensor x1
RGB LED Module x2
Two-color Common-Cathode LED x2
Shock switch x1
Knock sensor x1
Infrared transmitter x1
Laser Transmitter x1
Reed switch x1
Mini Reed x1
Infrared-receiver x1
Analog-temperature sensor x1
Digital-temperature sensor x1
Active Buzzer x1
Passive Buzzer x1
Button Switch x1
Photo-interrupter x1
Tilt-switch module x1
Mercury switch module x1
Magic Cup x2
DS18B20 Temperature Sensor x1
Rotary Encoder module x1
7-color Auto-flash LED module x1
Photoresistor Sensor x1
Humiture Sensor x1
Obstacle Avoidance Sensor x1
Tracking Sensor x1
Microphone Sensor x1
High-sensitive Voice Sensorx1
Metal Touch Sensor x1
Flame Sensor x1
Relay module x1

Joystick PS2 x1
MQ-2 Gas Sensor x1
MQ-2 Smoke sensor module x1
Breadboard x1
ADC0832 x1
T-Cobbler x1
Jumper wires (male to female) x40
Jumper wires (male to male) x20

Content

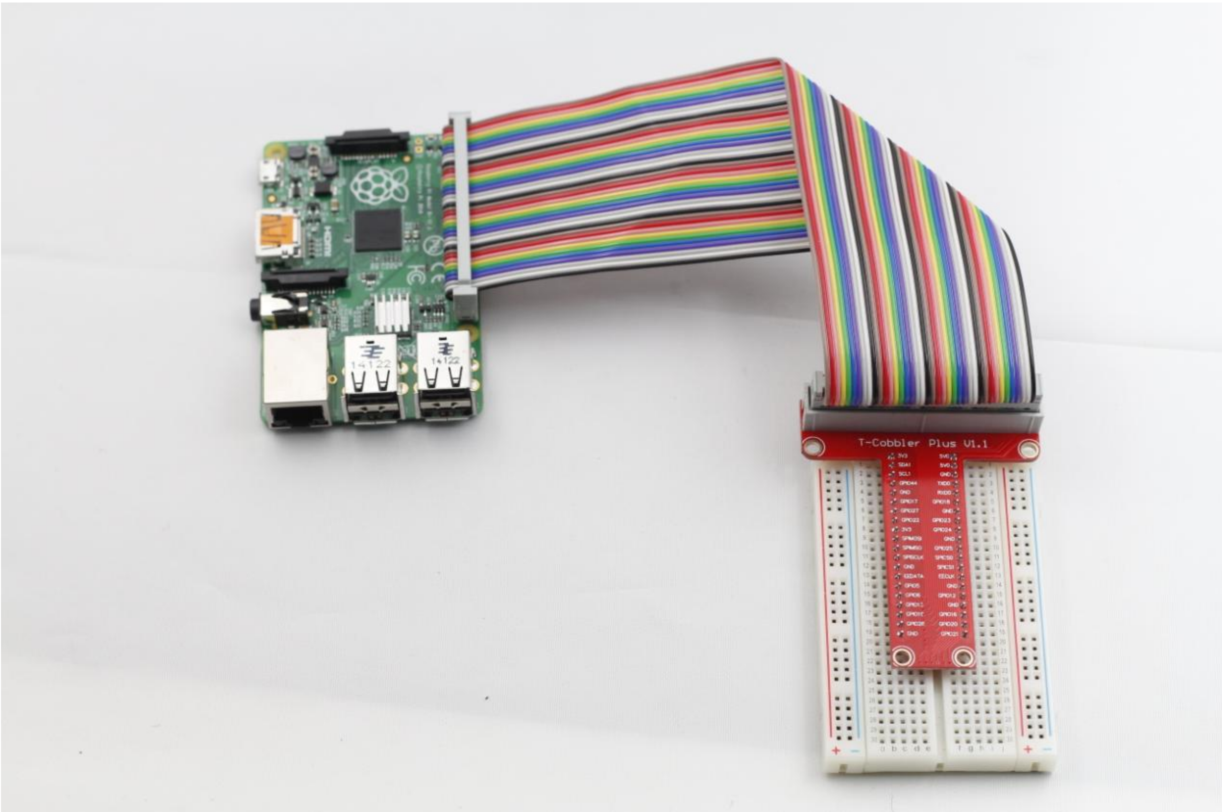
Notice

This is our GPIO Extension Board and GPIO cable.



You must connect the GPIO cable to the Raspberry Pi B+ like this:

After connection, as shown below:



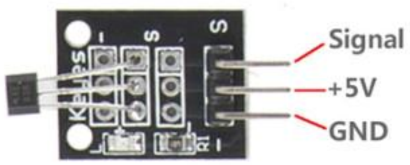
Lesson 1 Hall Sensor

Introduction

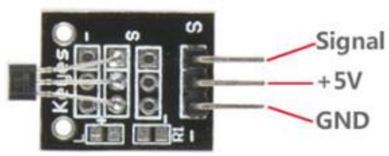
Based on hall effect, a hall sensor is a sensor that varies its output voltage in response to a magnetic field. Hall sensors are used for proximity switching, positioning, speed detection, and current sensing applications.

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

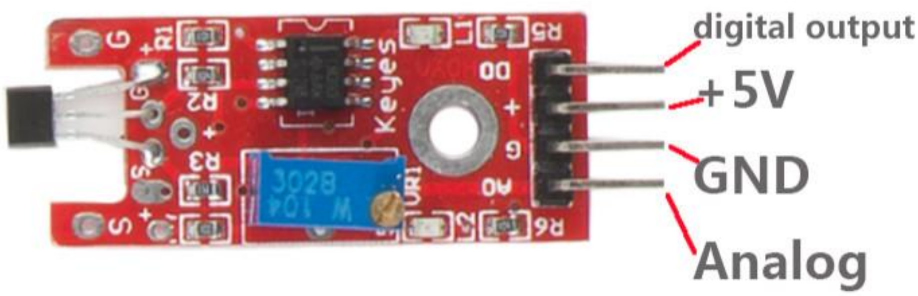
There are three different types of hall sensor modules in this kit (as shown below): linear hall sensor which outputs analog signals, switch hall sensor which outputs digital signals. If we add a comparator on the basis of linear hall sensor, it will be able to output both analog and digital signals.



Linear Hall



Switch Hall



Linear Hall

Components

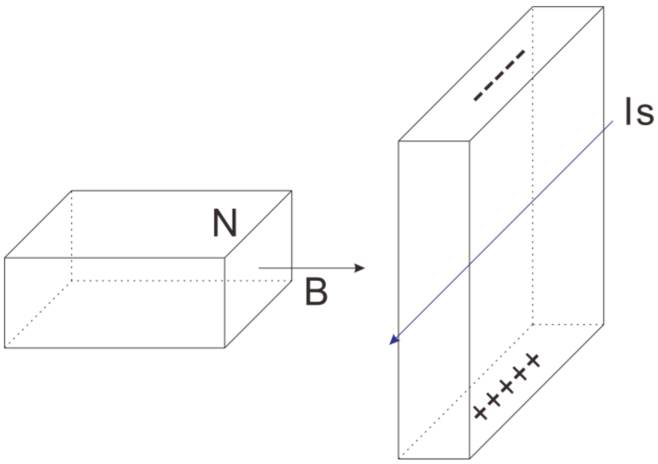
- 1*Raspberry Pi B+
- 1*Breadboard
- 1*Network cable (or USB wireless network adapter)
- 1*Linear sensor module

- 1*Two-color Common-Cathode LED module
- 1*Switch hall module
- Several jumper wires

Experimental Principles

Hall effect

Hall effects is a kind of electromagnetic effect. It was discovered by Edwin Hall in 1879 when he was researching conductive mechanism about metals. When a current is perpendicular to external magnetic field and flows through a conductor, a potential difference will be generated on the two surfaces of the conductor that is parallel to the direction of the magnetic field and the current. This phenomenon is Hall effect.



Hall sensor

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

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 **switch Hall sensor**, please perform the following steps

Step 1: Connect the circuit

Connect Raspberry Pi B+ GPIO0 to pin S on Switch Hall Module

Connect Raspberry Pi B+ GPIO1 to pin S on Double Color LED Module

Connect Raspberry Pi B+ GND to Switch Hall Module GND and double Color LED Module GND

Connect Raspberry Pi B+ pin 3.3V to pin '+' of Switch Hall Module and pin '+' of double Color LED Module

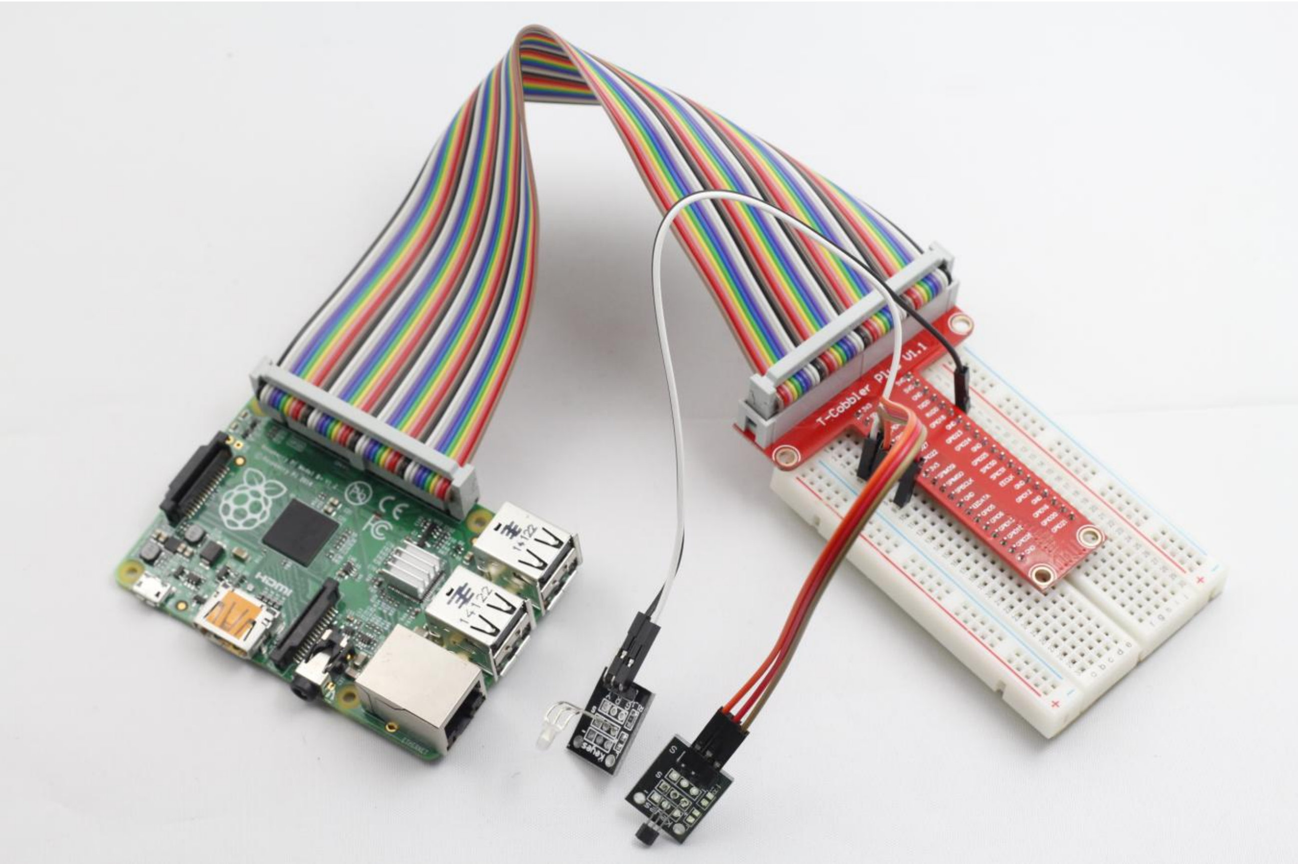
Step 2: Edit and save the code(see path/Rpi_SensorKit_code/01_ hall/switch_hall.c)
Step 3: Compile the code

    gcc switch_hall.c -lwiringPi

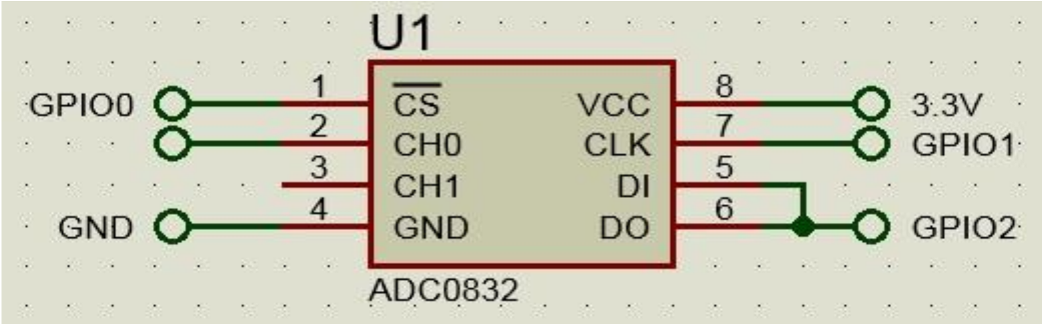
Step 4: Run the program

    ./a.out

Press Enter, if a magnet approaches the Switch Hall sensor, a string "Detected magnetic materials" is printed on the screen, and the LED will light up.



For **linear Hall sensor**, please perform the following steps
Step 1: Connect the circuit



Connect pin S on Linear Hall sensor to pin CH0 on ADC0832

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

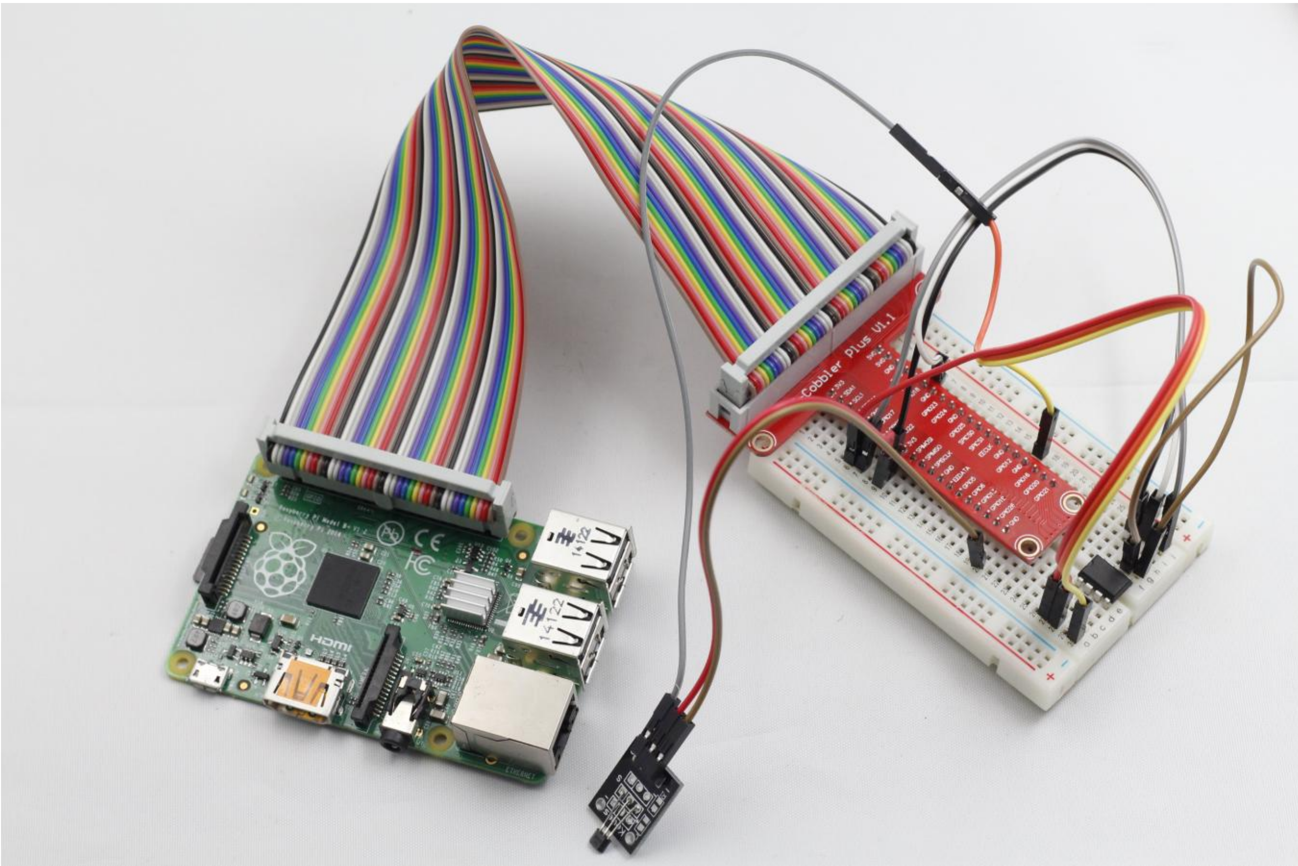
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

Press Enter, if a magnet approaches the analog Hall sensor, the value printed on the screen will increase.



For **linear Hall sensor (added a comparator)**, please perform the following steps

Step 1: Connect the circuit

Connect pin AO on Linear Hall sensor to pin CH0 on ADC0832

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

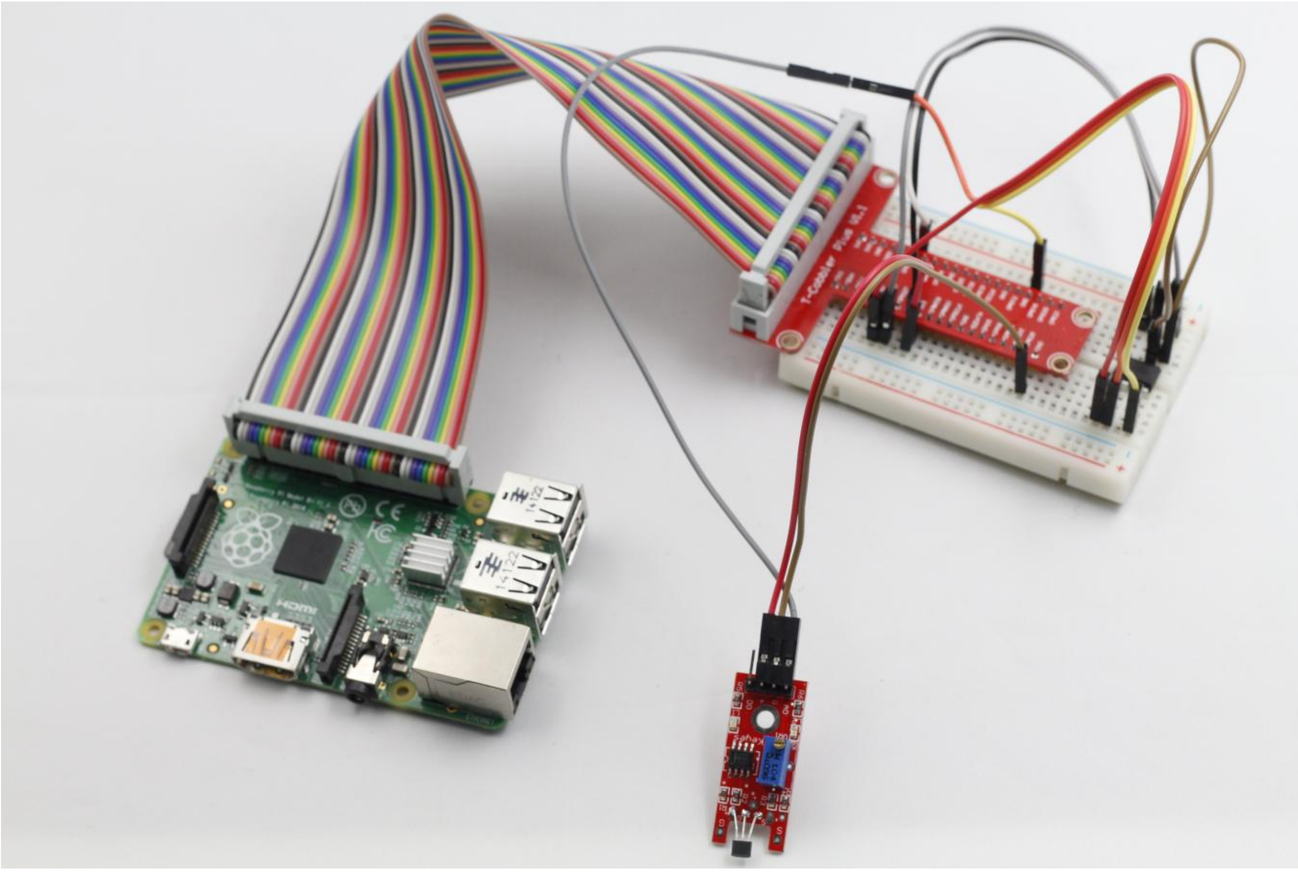
Step 3: Compile the code

gcc linearHall.c -lwiringPi

Step 4: Run the program

./a.out

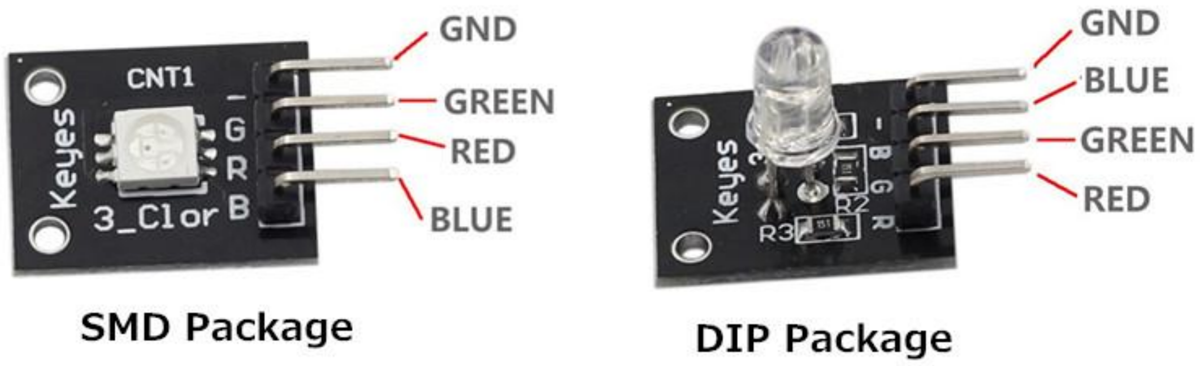
Press Enter, if a magnet approaches the linear Hall sensor, the indicator light on the linear Hall sensor will light up, and at the same time, current intensity of magnetic field will be printed on the screen.



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.



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

Components

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

Experimental Principle

The three primary color red, green and blue of a RGB LED can compose various colors by brightness. We can adjust the brightness of RGB LED with PWM technology. Raspberry Pi B+ has only one channel hardware PWM output, but we need three channels to control the RGB LED. As a result, it is difficult to realize with the hardware PWM of Raspberry Pi B+. Do not worry! Fortunately the softPwm library simulates PWM (softPwm) for us with software method. Based on this, we only need to include the header file softPwm.h, then call the API it provided to easily achieve multi-channel PWM output to control the RGB LED to display all kinds of colors.

RGB LEDs can be categorized into common anode type and common cathode type. In this experiment, we use common cathode RGB LED.

Experimental Procedures

Step 1: Connect the circuit according to the following method

Raspberry Pi B+ RGB LED module

```
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 the code

gcc rgb.c –lwiringPi -lpthread

Step 4: Run the program

./a.out

Press Enter, you will see the RGB LED light up, and display different colors in turn.

Further Exploration

I hope you can modify the parameters of the function ledColorSet( ) by yourself, then compile and run the program to see the color changes of the RGB LED.

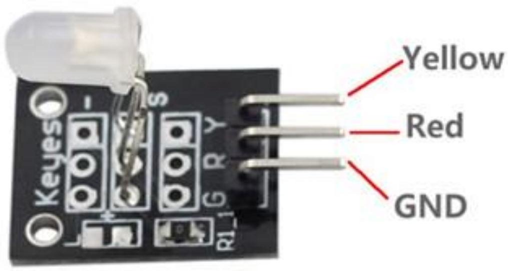
Experimental Summary

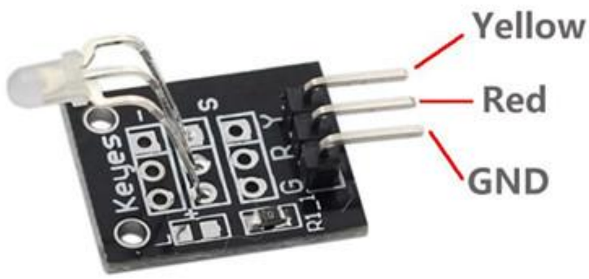
You have learnt how to control RGB LEDs with softPwm of the Raspberry Pi B+ through this lesson. I hope you can continue to explore softPwm application in DC motor speed regulation.

Lesson 3 Two-color Common-Cathode LED

Introduction

There are two kinds of two-color Common-Cathode LED in this kit. The only difference between them is their LED package size as shown below:





Components

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

Experimental Principle

Connect pin Yellow and Red to GPIOs of Raspberry Pi B+, then program the Raspberry Pi B+ to turn the LED on/off, and then use PWM to composite several kinds of colors.

Experimental Procedures

Step 1: Connect the circuit

Raspberry Pi B+	Two-color Common-Cathode LED module
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 the code

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

Step 4: Run the program

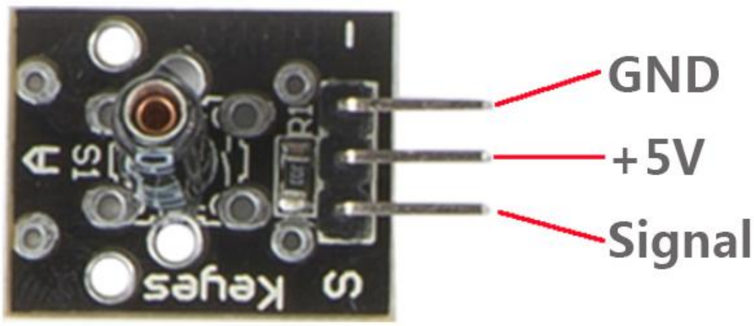
```
./a.out
```

Press Enter, you can see the two-color LED appears yellow, red and transition color.

Lesson 4 Shock Switch

Introduction

A shock switch (as shown below), also called vibration switch, spring switch or shock sensor, is an electronic switch which induces shock force and transfer induced result to circuit device to make the circuit start to work. It contains such main parts as conductive vibration spring, switch ontology, trigger pin, and so on.



Experimental Conditions

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

Experimental Principle

The main principle of shock switch is that, conductive vibration spring and trigger pin are precisely placed in switch ontology and bond to curing position through adhesive. Normally, the spring and the trigger pin do not contact. Once shook, the spring will shake and contact with trigger pin to conduct and generate trigger signals.

In this experiment, we will use a two-color LED module to indicate shock signals. When the shock switch inducts shock signals, the LED will light up.

Experimental Procedures

Step 1: Connect the circuit

Shock switch connection: connect pin 's' on Shock switch module to GPIO0 on Raspberry Pi B+; connect GND on Shock switch module to GND on Raspberry Pi B+; connect pin '+' on Shock switch module to 3.3V on Raspberry Pi B+

Two-color LED module connection: connect pin 'R' on two-color LED module to GPIO1 on Raspberry Pi B+; connect GND on two-color LED module to GND on Raspberry Pi B+

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/06_shockSwitch/

shockSwitch.c)

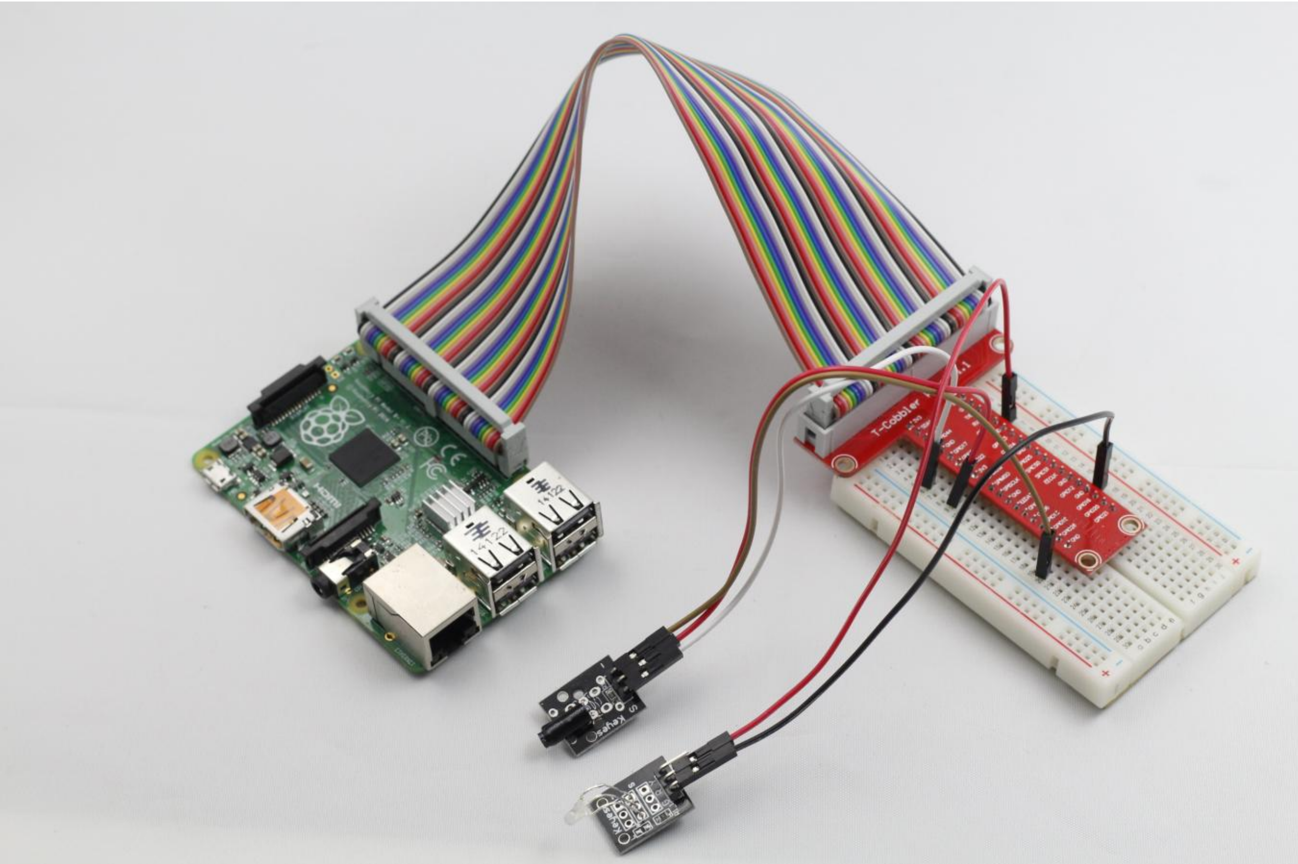
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

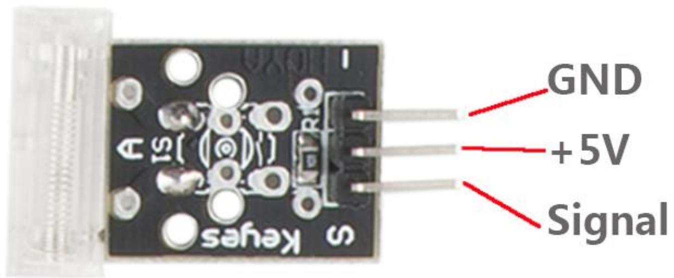
Press Enter, if you shake the switch, you will see the string "Detected shaking ! count = ?" printed on screen, and the LED will light up.



Lesson 5 Knock Sensor

Introduction

A knock sensor (as shown below) is similar to the shock switch. Just more sensitive. It can feel small amplitude virbration.



Components

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

Experimental Principle

Similar to the shock switch. When you knock the sensor, the two spring leaf will contact 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.

In this experiment, we will use a two-color LED module to indicate knock signals. When the knock switch inducts knock signals, the LED will light up.

Experimental Procedures

Step 1: Connect the circuit

Knock switch connection: connect pin 's' on knock switch module to GPIO0 on Raspberry Pi B+; connect GND on knock switch module to GND on Raspberry Pi B+; connect pin '+' on knock switch module to 3.3V on Raspberry Pi B+.

Two-color LED module connection: connect pin 'R' on two-color LED module to GPIO1 on Raspberry Pi B+; connect GND on two-color LED module to GND on Raspberry Pi B+

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/07_knockSensor/

knockSensor.c)

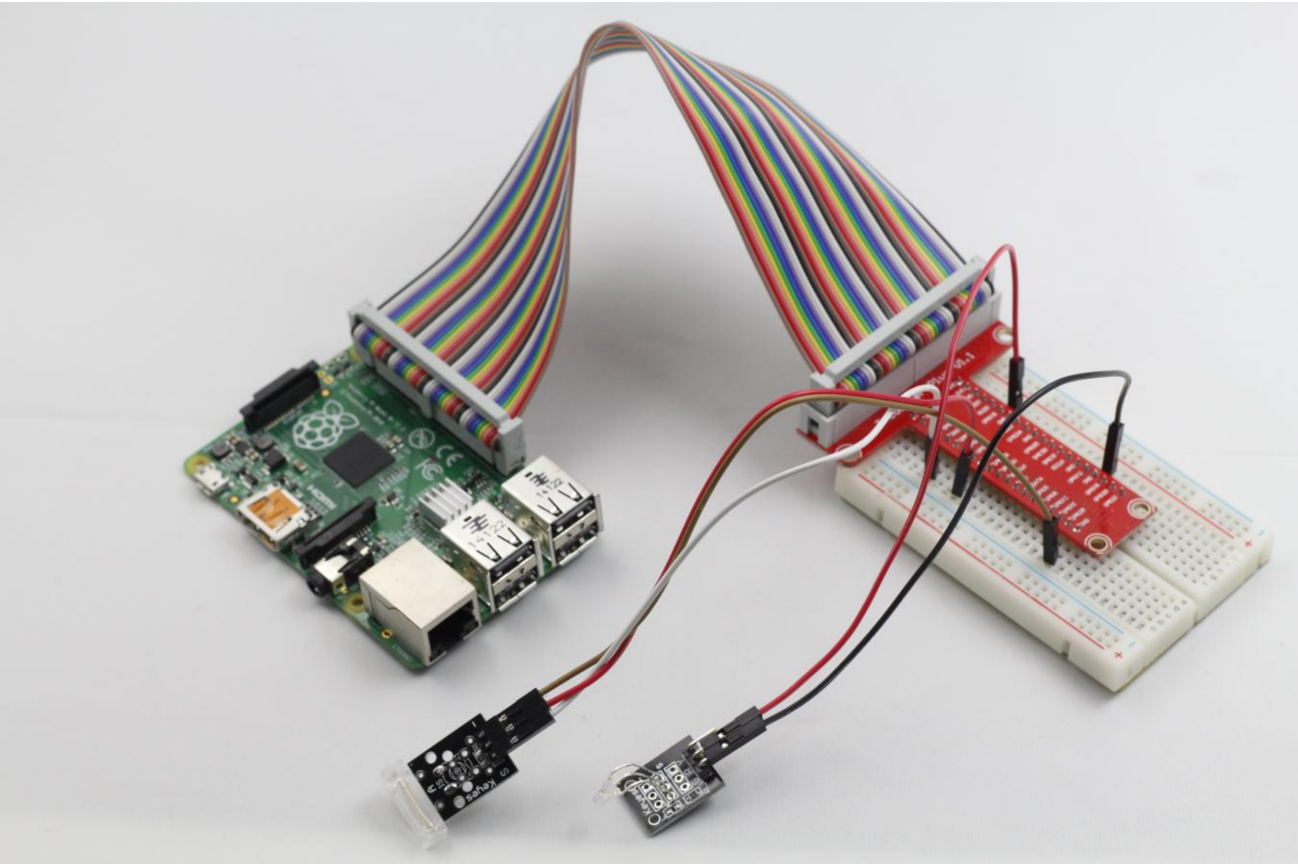
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

Press Enter, if you shake the sensor, pin 's' will output Low 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 device with remote control function. It can emit rays within a certain range through infrared transmitting tube so as to control signals. Infrared-transmitters are widely used in consumer electronics, industry and communication, etc.

Components

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

Experimental Principle

After connecting the circuit, we will let the module emit infrared rays by programming. Since infrared ray is invisible, so we cannot see it. But we can see it by using a camera.

Experimental Procedures

Step 1: Connect the circuit

Raspberry Pi B+ Infrared transmitter module

GPIO0 -------------------------------------- ‘s’

GND -------------------------------------- ‘-’

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

Step 3: Compile the code

gcc infrared.c -lwiringPi

Step 4: Run the program

./a.out

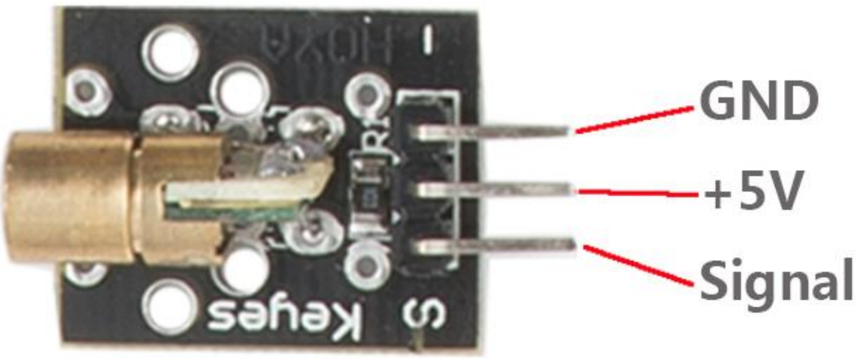
Press Enter, then you can see the infrared diode on the module emit infrared rays by a camera.
(Note: Infrared rays are not visible, but can be captured by cameras).

Lesson 7 Laser Transmitter

Introduction

Laser is widely used in medical treatment, military, and other fields due to its good directivity and energy concentration. In the world of art, laser is widely used in stage to make the effect of shadow and laser holography.

The appearance of a laser-transmit module is shown as below:



Components

- 1*Raspberry Pi B+
- 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. The term "laser" originated as an acronym for "light amplification by stimulated emission of 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. Spatial coherence also allows a laser beam to stay narrow over long distances (collimation), enabling applications such as laser pointers. 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. Temporal coherence can be used to produce pulses of light—as short as a femtosecond.

Experimental Procedures

Step 1: Connect the circuit

Raspberry Pi B+ Laser-transmitter module

GPIO0 ------------------------------------- ‘s’

3.3V --- ‘+’

GND -------------------------------------- ‘GND’

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

Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

Press Enter, you can see the laser blinking constantly.

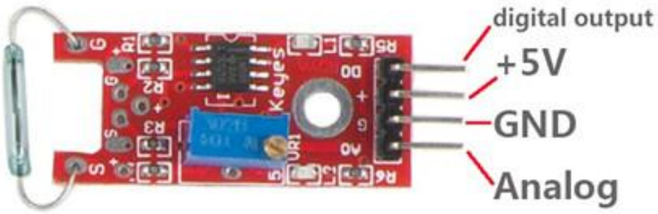
Note: Don't look directly at the laser head. It can cause great harm to your eyes.

Lesson 8 Reed Switch

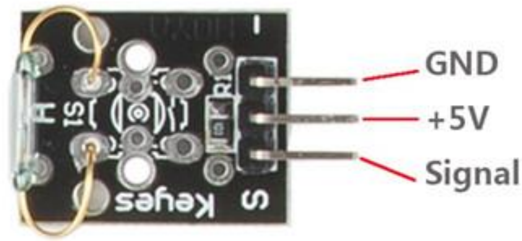
Introduction

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

There are two kinds of reed switches in this kit. One, we call it **reed switch**. The other, we call it **mini reed**. They have the same principle.



Reed Switch



Mini Reed

Components

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

Experimental Principle

A reed switch is a type of switch component that controls by magnetic signals. When magnetic material approaches the reed switch and reaches a certain intensity of magnetic field, the reed switch will close. 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 ease to 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 contact under the function of magnetic field lines.

As a result, the two reeds will pull together to connect the circuit connected with the nodes. After external magnetic force disappeared, the two reeds will be separated with each other because they have the same magnetism, so the circuit is also disconnected. Therefore, as a line switch component controlling by magnetic signals, the dry reed pipe can be used as a sensor to count, limit positions and so on. At the same time, it is widely used in a variety of communication devices.

Experimental Procedures

Step 1: Connect the circuit

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

Two-color LED module connection: connect pin 's' on two-color LED module to GPIO1 on Raspberry Pi B+; connect GND on two-color LED module to GND on Raspberry Pi B+; connect pin '+' on two-color LED module to 3.3V on Raspberry Pi B+

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

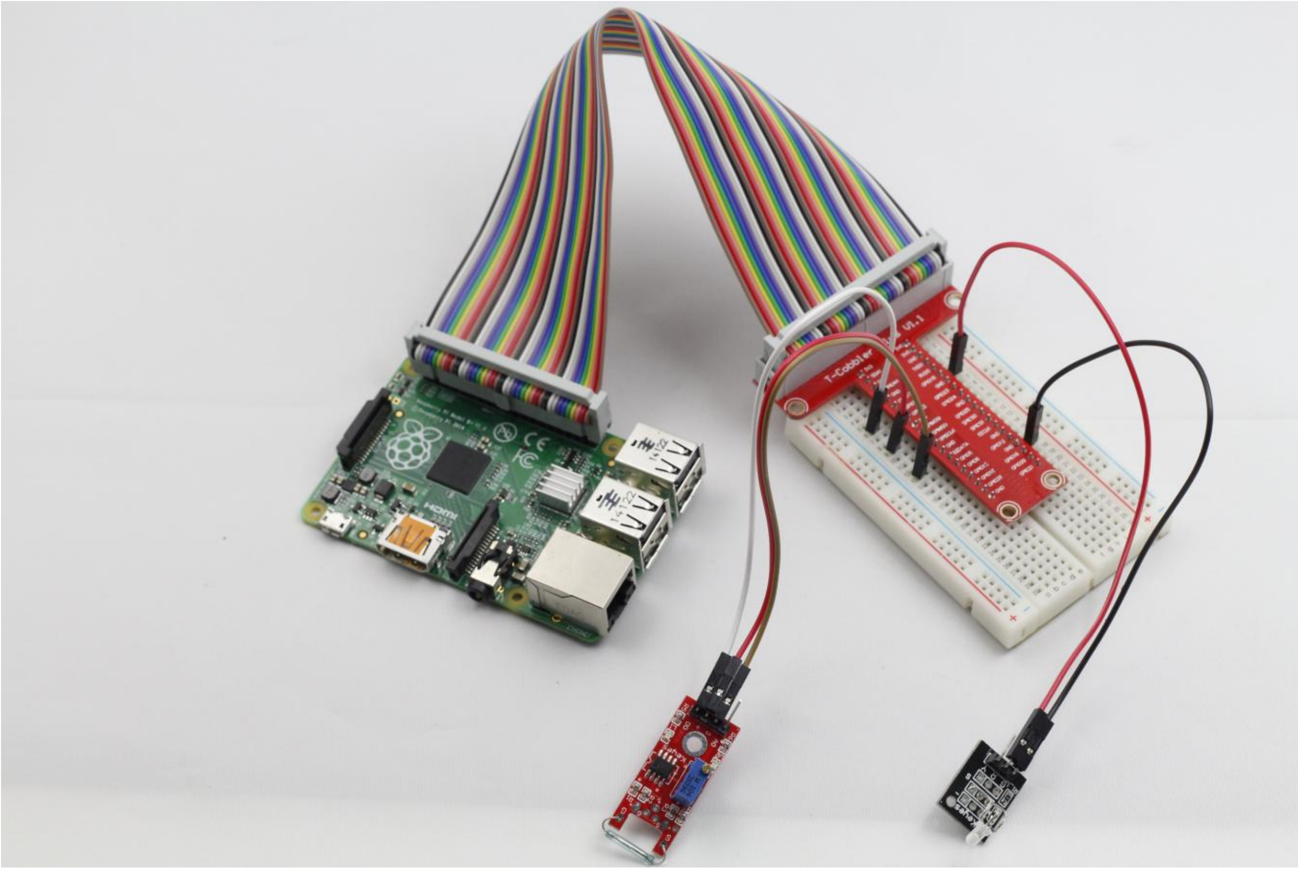
Step 3: Compile the code

```
gcc   magicRing  -lwiringPi
```

Step 4: Run the program

```
./a.out
```

Press Enter, if you use magnetic material to approach the reed switch, you will detect that pin DO output high and see "Detected Magnetic Material" displayed on the screen and the LED will light up.



Mini Reed

Experimental Procedures

Step 1: Connect the circuit

Connect pin ‘s’ on Mini Reed module to GPIO0 on Raspberry Pi B+, other connections are the same with Reed switch module

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

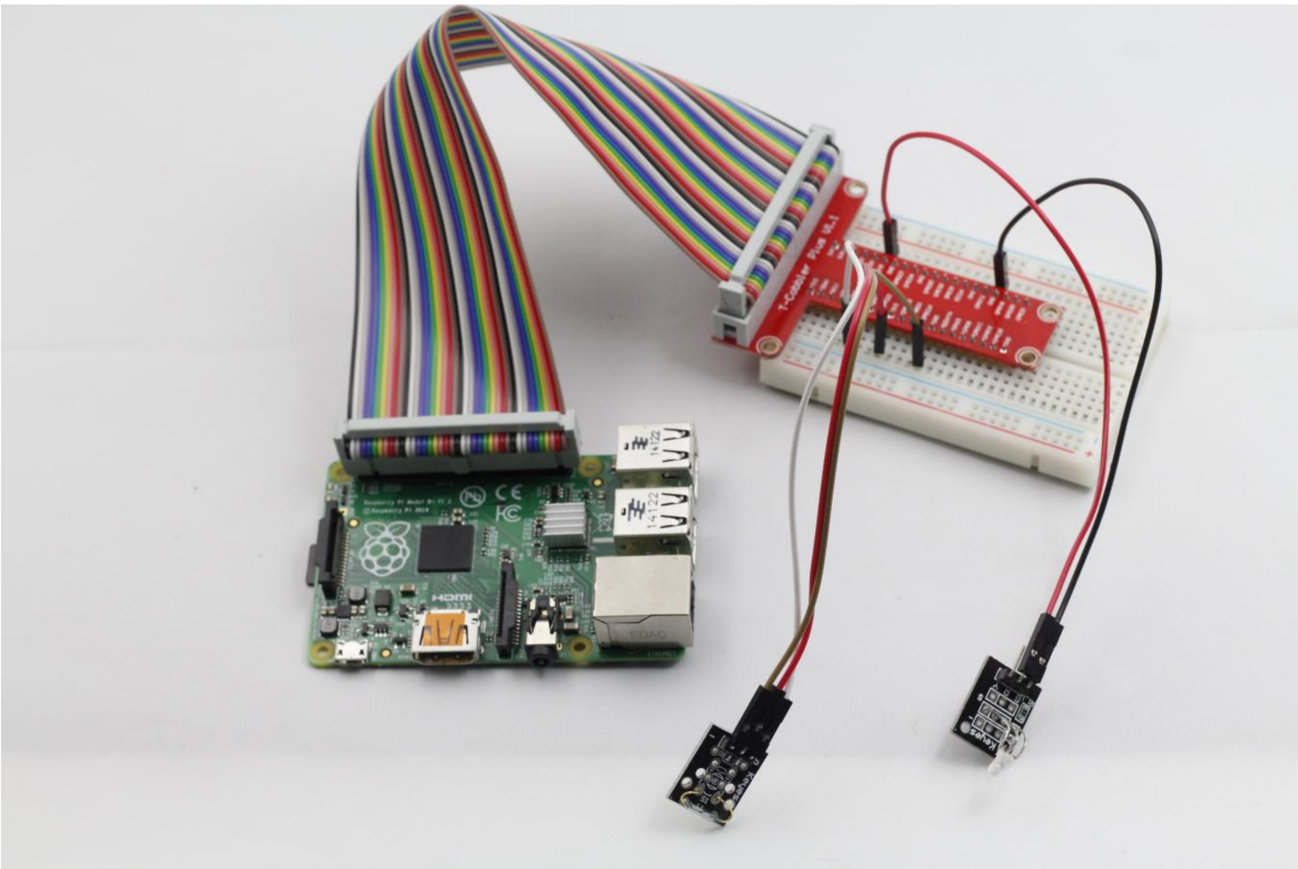
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

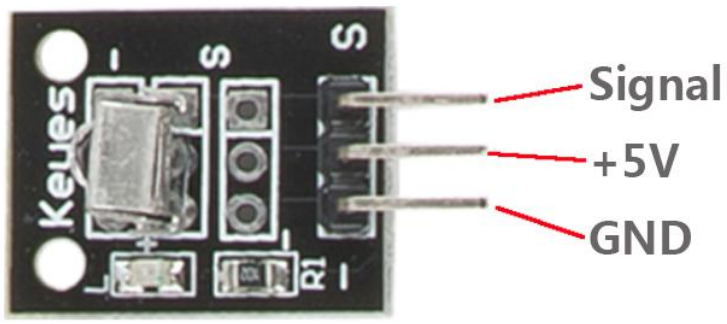
Press Enter, you will see the same result.



Lesson 9 Infrared-Receiver

Introduction

An infrared-receiver (as shown below) is a type of component which can receive infrared signals and can independently complete infrared ray reception and output compatible with TTL level signals. It has similar size with normal plastic package transistors and is appropriate for all kinds of infrared ray remote control and infrared ray data transmission.



Components

- 1*Raspberry Pi B+
- 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: Connect the circuit

Infrared-receiver module	Raspberry Pi B+
'S'	'GPIO0'
'+'	3.3V
'-'	GND

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

Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

Press Enter, you will see the LED on the module light up when receiving infrared rays. At the same time, infrared pulses will be printed on the screen.

Lesson 10 Analog-temperature Sensor

Introduction

A thermistor is the core component of an analog-temperature sensor. There are two kinds of analog-temperature sensors in this kit (as shown below). In this lesson, we take the left one for example.

Components

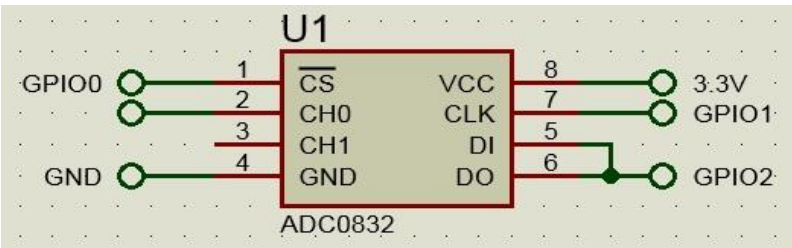
- 1*Raspberry Pi B+
- 1*Breadboard
- 1*Network cable (or USB wireless network adapter)
- 1*Analog-temperature Sensor module
- 1*ADC0832
- Several jumper wires

Experimental Principle

This module is based on thermistor principle, whose resistance varies significantly with ambient temperature. When the temperature increases, the thermistor resistance decreases; when the temperature decreases, the thermistor resistance increases. It can detect surrounding temperature changes in real time. In the experiment, we need to use an analog-digital converter ADC0832 to convert analog signal into digital signal.

Experimental Procedures

Step 1: Connect the circuit



Analog-temperature Sensor module ADC0832

‘s’ --- CH0

‘+’ --- 3.3V

‘-’ --- GND

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

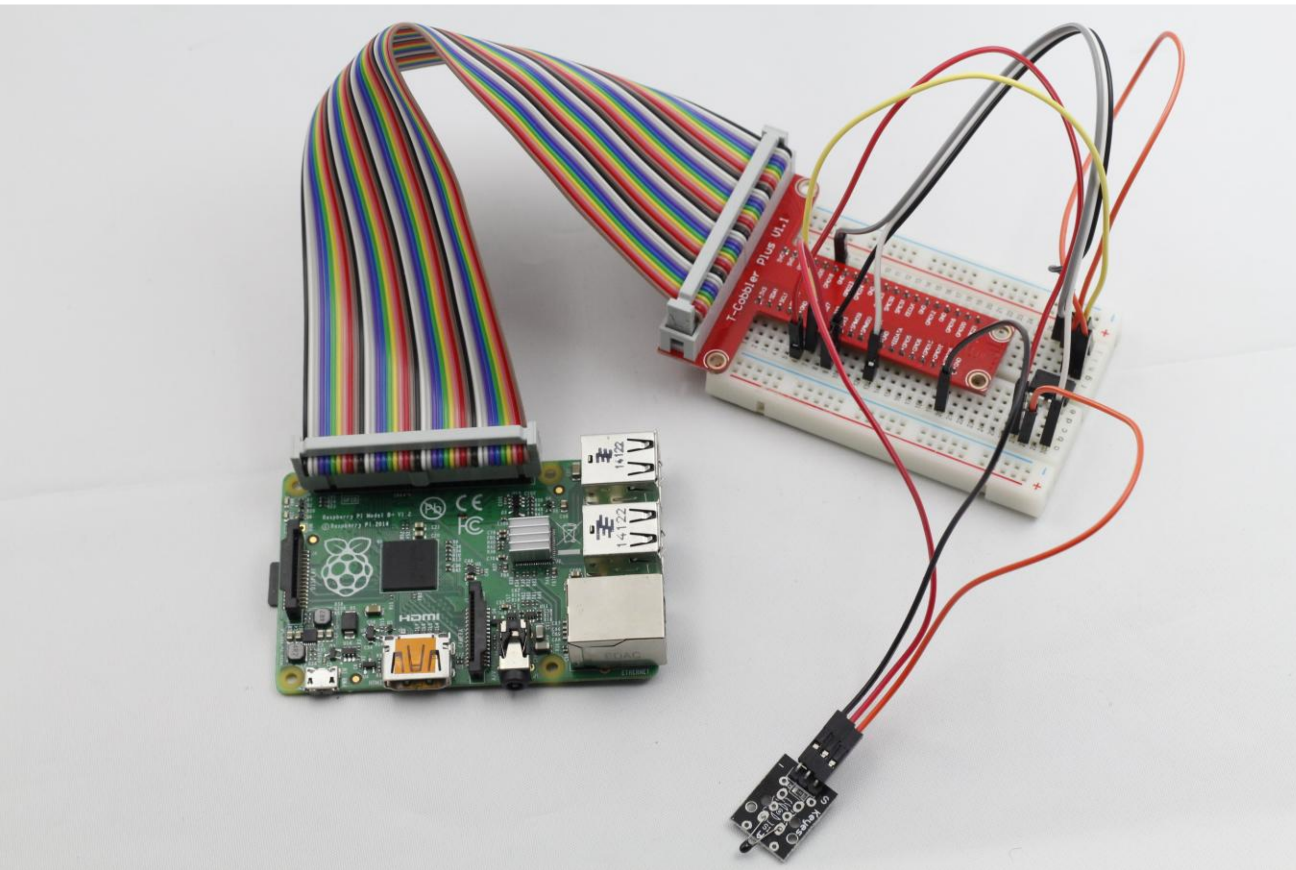
Step 3: Compile the code

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

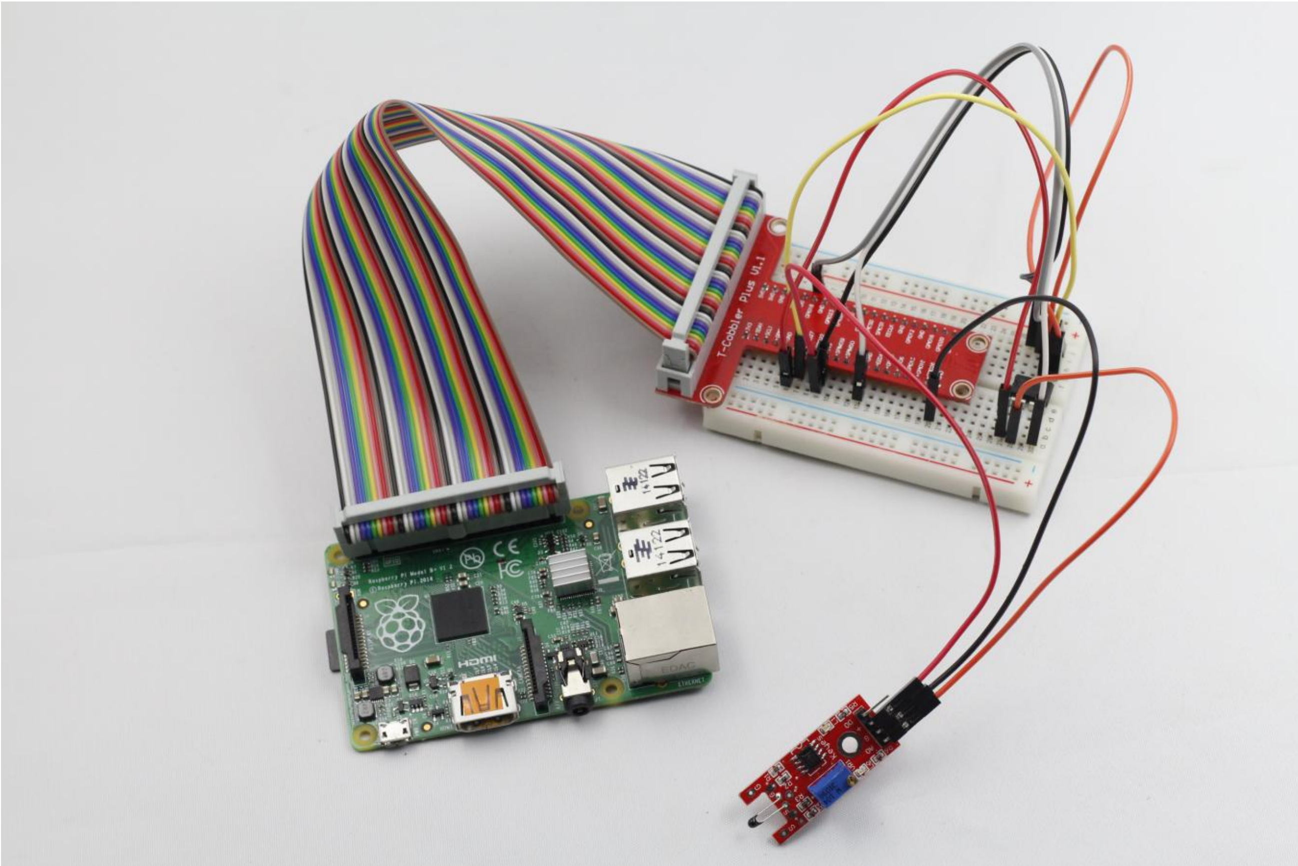
Step 4: Run the program

```
./a.out
```

Press Enter, if you touch the thermistor, you can see current temperature value displayed on the screen change accordingly.



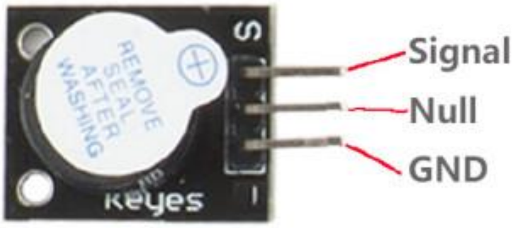
Compared with the left one, the right one only adds a digital output. You can adjust the threshold through a potentiometer. When an output is higher than the threshold, the sensor will output high; when an output is lower than the threshold, the sensor will output low.



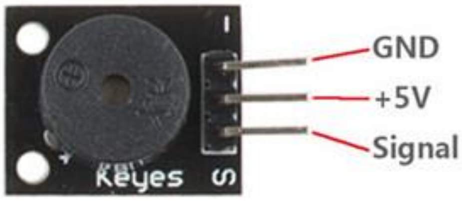
Lesson 11 Buzzer

Introduction

Buzzers can be categorized as active and passive buzzers (See the following picture).



Active



Passive

Components

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

Experimental Principle

When we place the pins of two buzzers upwards, we can see the one with green circuit board is a passive buzzer, while the one without circuit board instead of enclosing with black wax is an active buzzer.

The difference between an active buzzer and a passive buzzer is:

The active buzzer has built-in oscillating source, so it will make sounds as long as it is wired up. While the passive buzzer does not have oscillating source, so it will not tweet if you use DC signals, instead you must use square waves whose frequencies are between 2K and 5K to drive it. The active buzzer is often more expensive than the passive because multiple built-in oscillating circuits exist.

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 fixed frequency.

Step 1: Connect the circuit

Raspberry Pi B+ Active buzzer module

GND ------------------------------------- '-'

GPIO0 ------------------------------------- 's'

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

Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

Press Enter, you can hear the active buzzer make ticking sounds.

Passive Buzzer

Step 1: Connect the circuit

Raspberry Pi B+ Passive buzzer module

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 the code

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

Step 4: Run the program

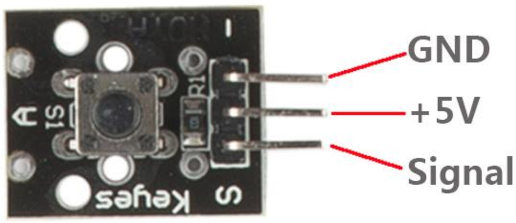
```
./a.out
```

Press Enter, you can hear the active buzzer play music.

Lesson 12 Button

Introduction

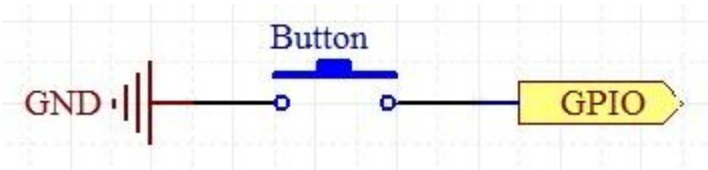
In this lesson, we will learn how to use buttons.



Components

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

Experimental Principle



Use a normally open button as an input device of Raspberry Pi B+, when the button is pressed, the GPIO (General Purpose Input/Output) 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, you can run the corresponding code according to this condition. In this experiment, we will print a string on the screen, and control a LED.

Experimental Procedures

Step 1: Connect the circuit

Button module Raspberry Pi B+

‘s’ ----------------------- GPIO0

‘+’ ---------------------- -3.3V

‘-’ ---------------------- --GND

Two-color LED module connection: connect pin ‘R’ on two-color LED module to GPIO1 on Raspberry Pi B+; connect GND on two-color LED module to GND on Raspberry Pi B+

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

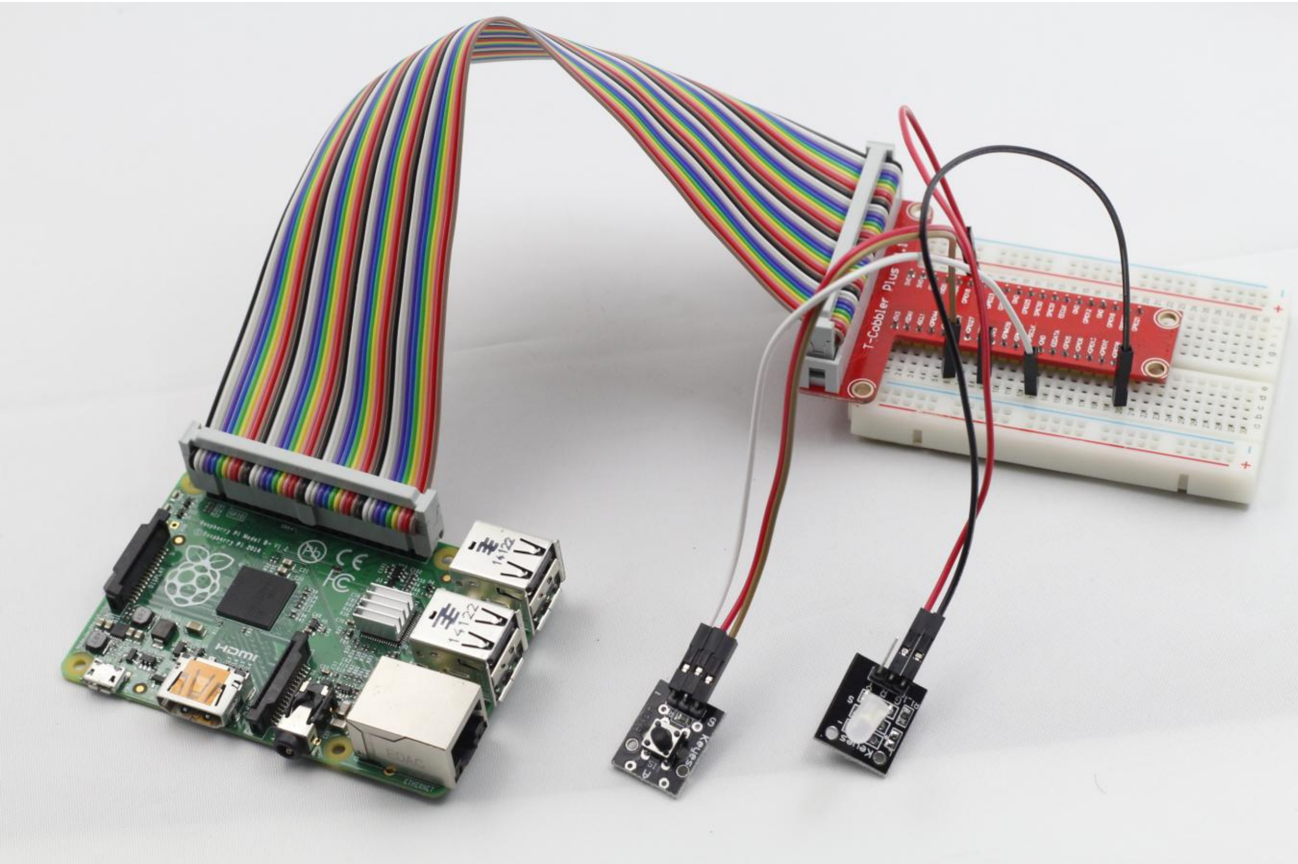
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

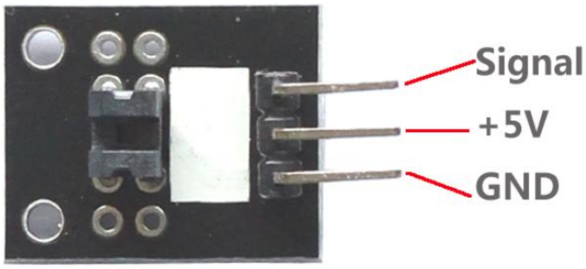
Press Enter, if you press the button, you will see a string “Button is pressed” displayed on the screen, and the state of LED will be switched (ON/OFF).



Lesson 13 Photo-interrupter

Introduction

A photo-interrupter (as shown below) is a sensor that arranges light-emitting component and light-receiving component face-to-face and packages them together. It uses the principle that light will be interrupted when there is an object passing through the sensor to realize detection function. Photo-interrupters are widely used in speed measurement.



Components

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

Experimental Principle

Well, basically a photo-interrupter consists of two parts: transmitter and receiver. Transmitter emits light (it could be, for example, an LED or a laser) and that light goes to receiver. If that light beam between transmitter and receiver is interrupted by some obstacles, and receiver detects no incoming light even for a brief moment – the output level will change. In this experiment, we will turn an LED on or off by this change.

Experimental Procedures

Step 1: Connect the circuit

Photo-interrupter module   Raspberry Pi B+

'S' ----------------------------------- GPIO0

‘+’ ---------------------------------- -3.3V

‘-’ ----------------------------------- --GND

Two-color LED module connection: connect pin ‘R’ on two-color LED module to GPIO1 on Raspberry Pi B+; connect GND on two-color LED module to GND on Raspberry Pi B+

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

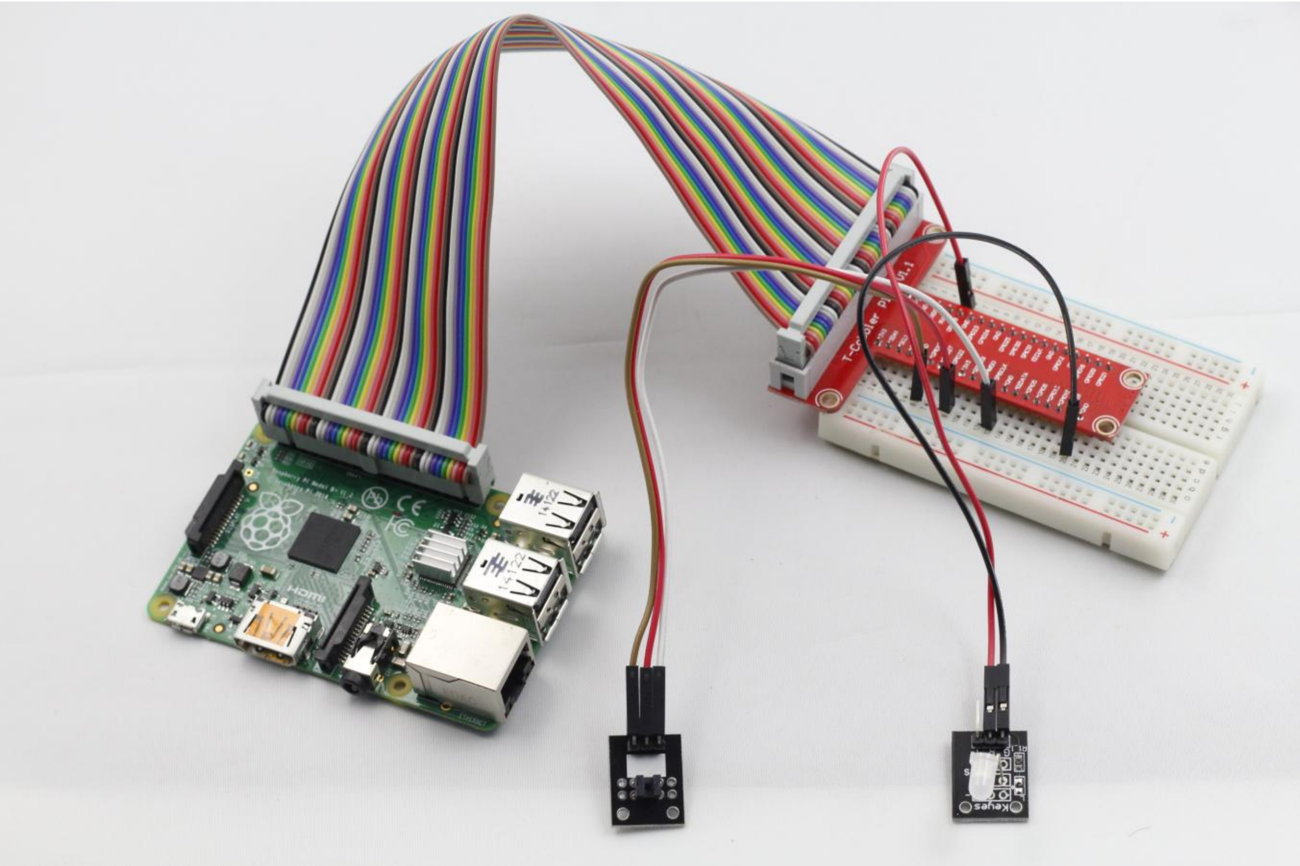
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

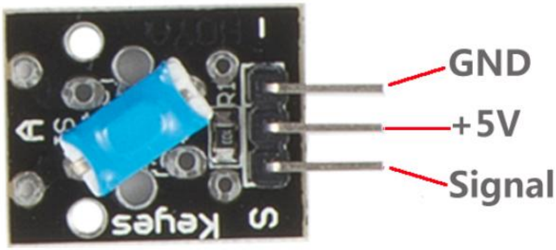
Press Enter, if you break incoming light with a piece of paper, you will see the LED turned on and a string “led on” will be printed on the screen; if you remove the paper, you will see the LED turned off and a string “led off” will be printed on the screen.



Lesson 14 Tilt-Switch

Introduction

The tilt-switch (as shown below) we use is a ball tilt-switch with a metal ball inside. It is used to detect small angle of inclination.



Components

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

Experimental Principle

The principle is very simple. It mainly uses the ball in the switch changing with different angle of inclination to achieve the purpose of triggering circuits. When the ball in tilt switch runs from one end to the other end because of external force shaking, the tilt switch will conduct, or it will break.

Experimental Procedures

Step 1: Connect the circuit

Tilt-Switch module   Raspberry Pi B+

'S' ------------------------------- GPIO0

'+' ---------------------------- -3.3V

'-' ---------------------------- --GND

Two-color LED module connection: connect pin 'R' on two-color LED module to GPIO1 on Raspberry Pi B+; connect GND on two-color LED module to GND on Raspberry Pi B+

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

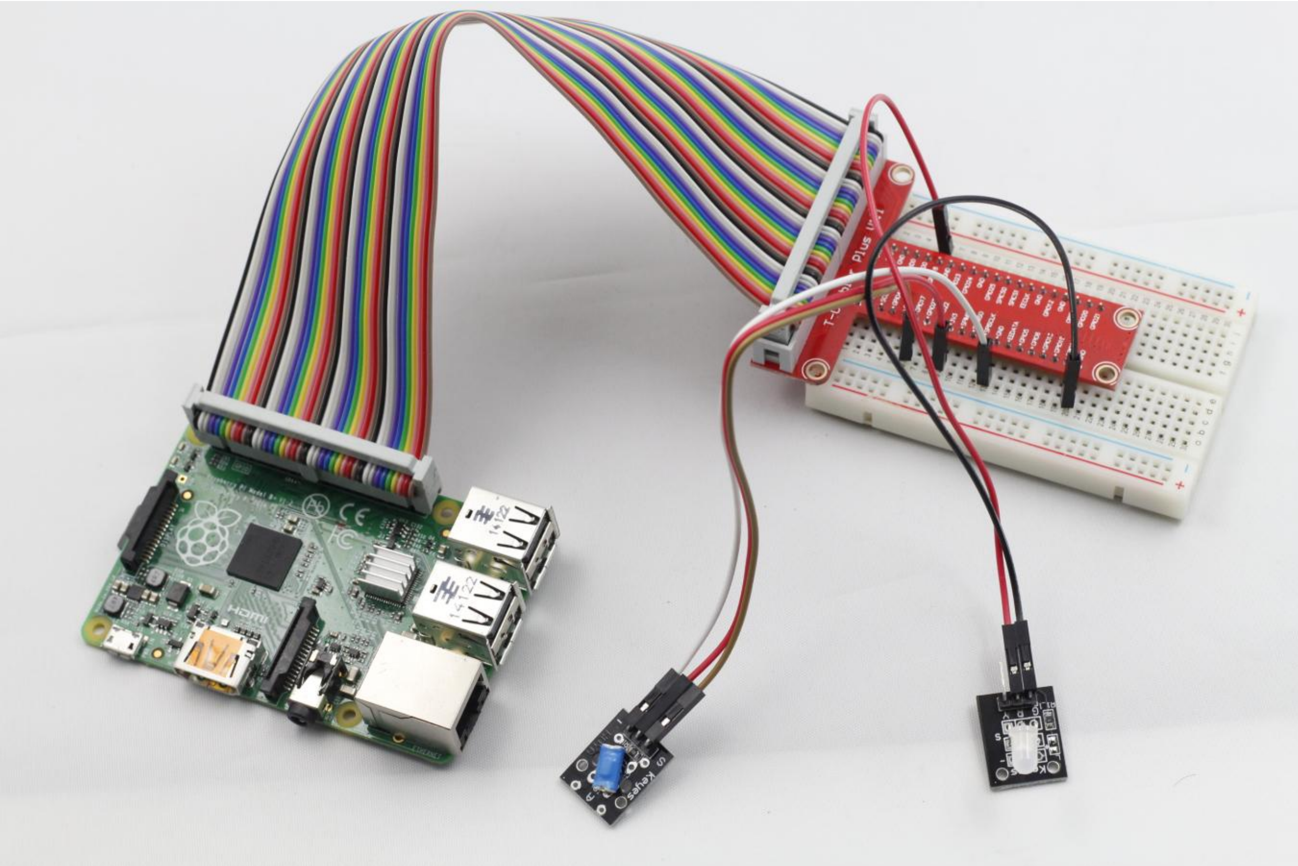
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

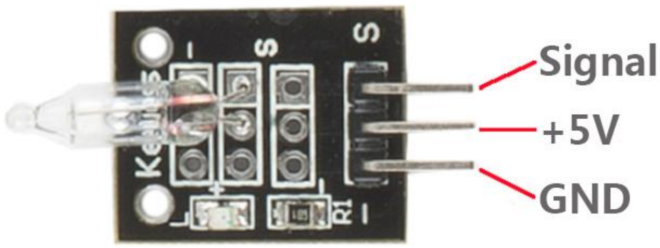
Press Enter, if you tilt the switch, the state of LED will be switched (ON/OFF).



Lesson 15 Mercury Switch

Introduction

Similar to tilt-switch, a mercury switch (as shown below) is used to detect slight large angle of inclinations.



Components

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

Experimental Principle

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.[1] The switch may contain multiple sets of contacts, closing different sets at different angles, allowing, for example, single-pole, double-throw (*SPDT*) operation.

The difference between a mercury switch and a ball switch:

First, All materials of a ball switch can meet environmental protection requirements, while a mercury switch cannot meet this requirement because of the material itself.

Second, a ball switch conducts by the metal ball connecting with the trigger pin to generate signals. Since the contact area of the ball and the trigger pin is small and the ball can move, flash breaking phenomenon will occur when conduction. However, a mercury switch conducts by mercury contacting with the trigger end. Generally speaking, a mercury switch has more stable conduction effect because of its liquid state and large contact area.

Third, a ball switch has metal shell with good structural strength.

Note: Mercury is harmful to human body and environment. Thus please be careful when using a mercury switch in case breaking out. It should also be properly handled when no longer being used.

Experimental Procedures

Step 1: Connect the circuit

Mercury switch module	Raspberry Pi B+
's'	GPIO0
'+'	3.3V
'-'	GND

Two-color LED module connection: connect pin 'R' on two-color LED module to GPIO1 on Raspberry Pi B+; connect GND on two-color LED module to GND on Raspberry Pi B+

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

Step 3: Compile the code

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

Step 4: Run the program

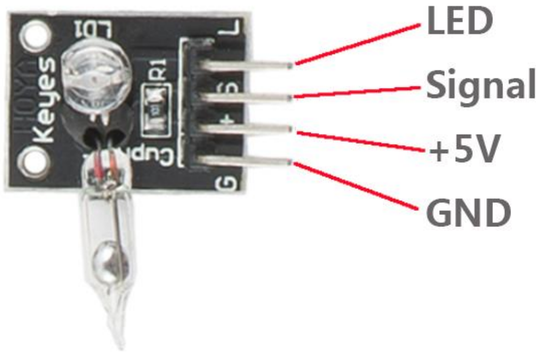
```
./a.out
```

Now, when you shake the mercury switch, the state of LED will be switched (ON/OFF).

Lesson 16 Magic Cup

Introduction

It consists of two same modules, and each module adds a separate LED on the basis of Mercury switch.



Components

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

Experimental Principle

When one mercury switch tilts, its attached LED will become darker and darker, while the other LED will become brighter and brighter. Just like telepathy.

Experimental Procedures

Step 1: Connect the circuit

Magic cup module	Raspberry Pi B+
'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 the code

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

Step 4: Run the program

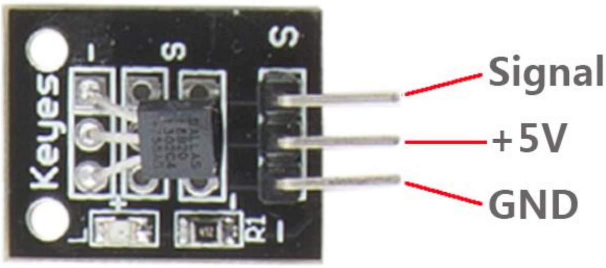
```
./a.out
```

Now, when you tilt Magic cup, you will see the LED on one module go out, while the other LED light up.

Lesson 17 DS18B20 Temperature Sensor

Introduction

DS18B20 Temperature Sensor is different from traditional AD collection temperature sensor. It uses 1-wire bus and can directly output temperature data.



Components

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

Experimental Principle

DS18B20 is a commonly used digital temperature sensor with the characters of small size, low hardware overhead, strong anti-interference ability and high precision. DS18B20 digital temperature sensor is easy to wire and can be applied to a variety of occasions after packaging, such as pipeline type, screw type, magnet adsorption type, and stainless steel package type. Packaged DS18B20 can be used to measure the temperature of cable trench, blast furnace water cycle, boiler, machine room, agriculture greenhouse, clean room, ammunition depot and other non extreme temperature occasions.

With unique 1-wire interface, DS18B20 requires only one port pin for two-way communication between microprocessor and DS18B20. It supports multi-point networking function, and up to eight DS18B20s can be connected to an only three-wire in parallel to measure multi-point temperatures. Too many DS18B20s will make the power supply voltage too low and cause the instability of signal transmissions.

Experimental Procedures

Step 1 : connect the circuit according to the following method

Raspberry Pi B+ ds18B20 module

GPIO7 ------------------ ‘s’

+5V ------------------ middle pin

GND ------------------- ‘-’

Step 2 : Upgrade your kernel

sudo apt-get update

sudo apt-get upgrade

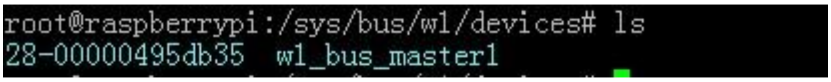
Step 3 : Mount the device drivers and confirm whether the device is effective or not

sudo modprobe w1-gpio

sudo modprobe w1-therm

cd /sys/bus/w1/devices/

ls

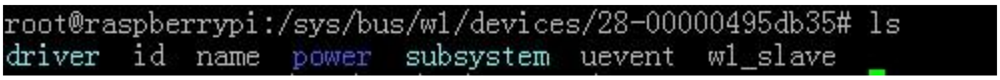


28-00000495db35 is an external temperature sensor device, but it may vary with every client. This is the serial number of your ds18b20.

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 you confirm the device is normal, we can read the temperature by programming

Reference code: path/16_ds18b20/ds18b20_2.c

Step 6 : Compile the code

```
gcc  ds18b20_2.c
```

Step 7 : Run the program

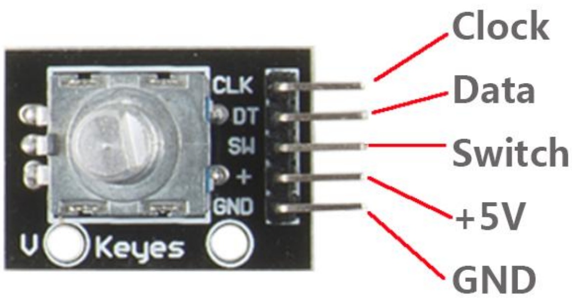
```
./a.out
```

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.



Experimental Conditions

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

Experimental Principle

Rotary Encoder is a switch electronic component with a set of regular and strict timing sequence pulses. When using with IC, it can achieve increment, decrement, page turning and other operations. For example, mouse scroll, menu selection, acoustic sound regulation, frequency regulation, toaster temperature regulation, and so on.

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 B+. When we rotate left and right, there will be pulse inputs in pin 1 and pin 3.

The figure shows, if GPIO0 is at high level and GPIO1 is also 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 judge the state of pin 3 when pin 1 is at high level, then you can judge whether the switch is rotates anti-clockwise or anti-clockwise.

Experimental Procedures

Step 1: Connect the circuit

Raspberry Pi B+ Rotary Encoder module

GPIO0 ---------------------------------- CLK

GPIO1 ---------------------------------- DT

3.3V ---------------------------------- ‘+’

GND ---------------------------------- GND

Step 2: Edit and save the code (see path/Rpi_SensorKit_code/17_RotaryEncoder/

rotaryEncoder.c)

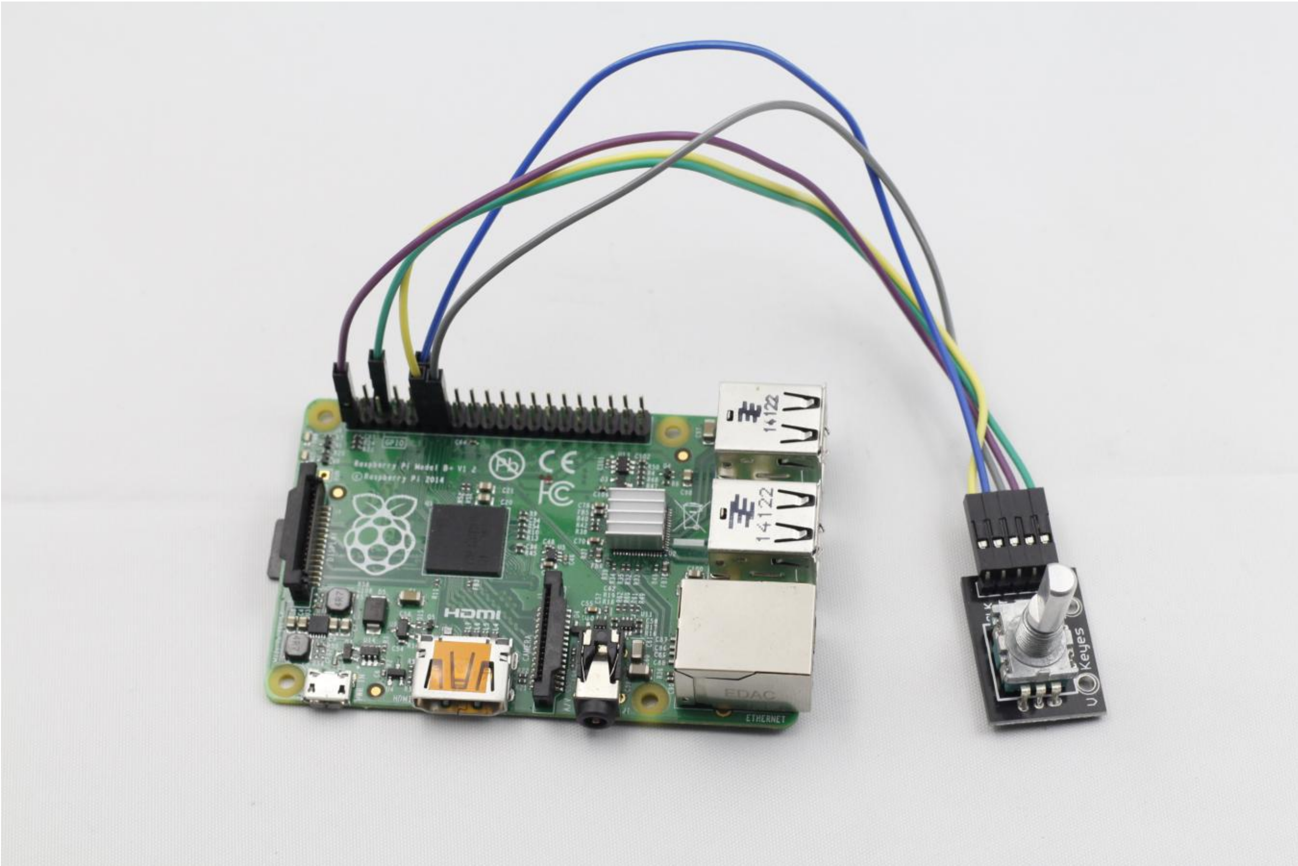
Step 3: Compile the code

gcc rotaryEncoder.c -lwiringPi

Step 4: Run the program

./a.out

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



Further Exploration

In this experiment, we do not use the button function of rotary encoder switch. I hope you can realize this function by yourselves. That is, when you press the switch, previously counted pulses will be reset.

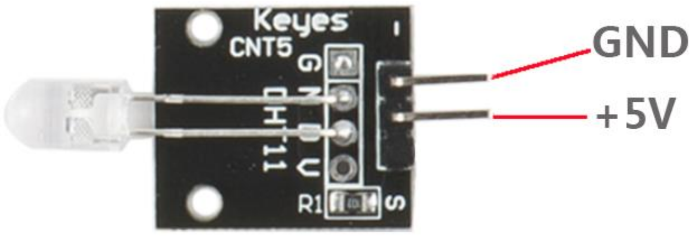
Summary

Through this lesson, you have been familiar with the principle of operation and programming realization for rotary encoder switch and preliminarily mastered the interrupt programming method of Raspberry Pi B+. Interrupt is a very important concept in computers and is widely used in modern computers. Interrupt greatly reduces the CPU load. I believe you will experience the charm of interrupt mechanism more deeply in later study.

Lesson 19 7-Color Auto-flash LED

Introduction

It can automatically flash built-in colors after being power on.



Experimental Conditions

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

Experimental Principle

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

Experimental Procedures

Connect the circuit

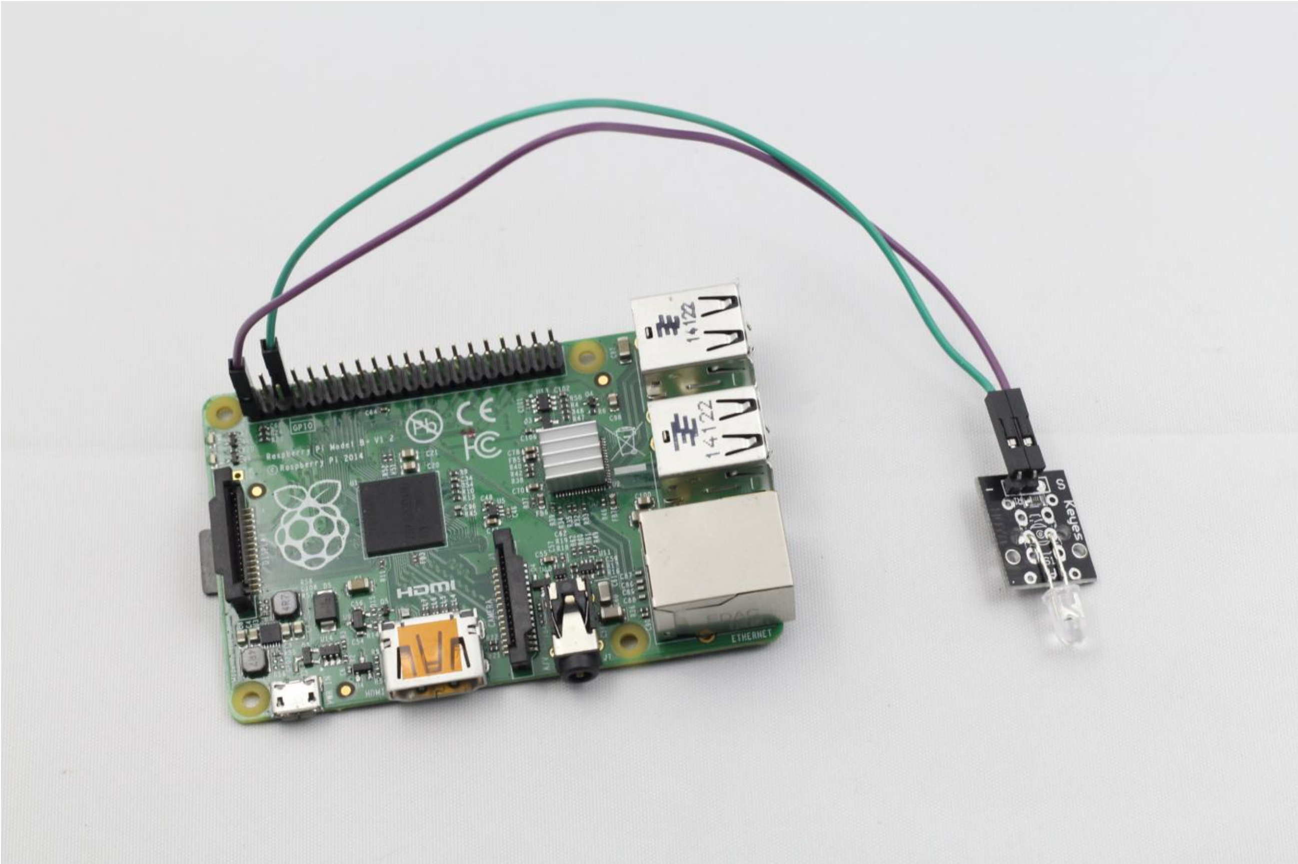
7-color auto-flash LED module **Raspberry Pi B+**

‘s’--------------------------------------5V

‘-“----------------------------------- GND

Note: we just take electricity from the Raspberry Pi B+.

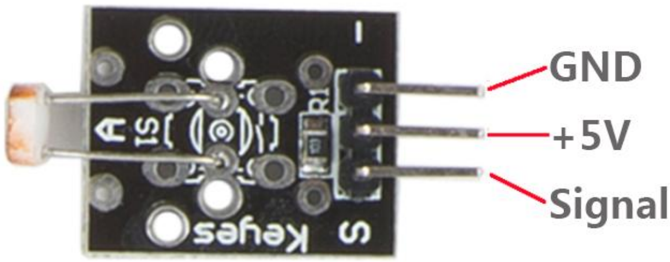
Now, you will see 7-color auto-flash LED flashing seven colors.


14122
14122
HDMI

Lesson 20 Photoresistor

Introduction

A photoresistor or light-dependent resistor (LDR) or photocell is a light-controlled variable resistor. The resistance of a photoresistor decreases with increasing incident light intensity; in other words, it exhibits photoconductivity. A photoresistor can be applied in light-sensitive detector circuits, and light- and dark-activated switching circuits.



Experimental Conditions

- 1*Raspberry Pi B+
- 1*Network cable (or USB wireless network adapter)
- 1*ADC0832
- 1*Photoresistor module
- Jumper wires

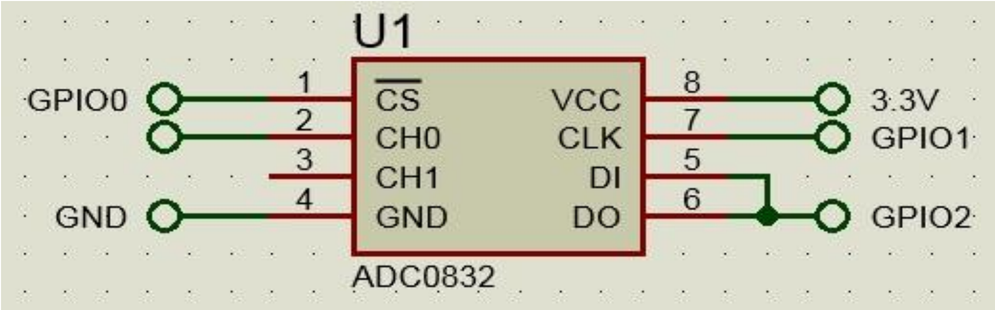
Experimental Principle

With light intensity increasing, the resistance of a photoresistor will decrease. Thus the output voltage will change.

Analog signals collected by the photoresistor are converted to digital signals through ADC0832. Then we send these digital signals to Raspberry Pi B+ and display them on the screen.

Experimental Procedures

Step 1: Connect the circuit



Connect pin S on Photoresistor module to pin CH0 on ADC0832

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

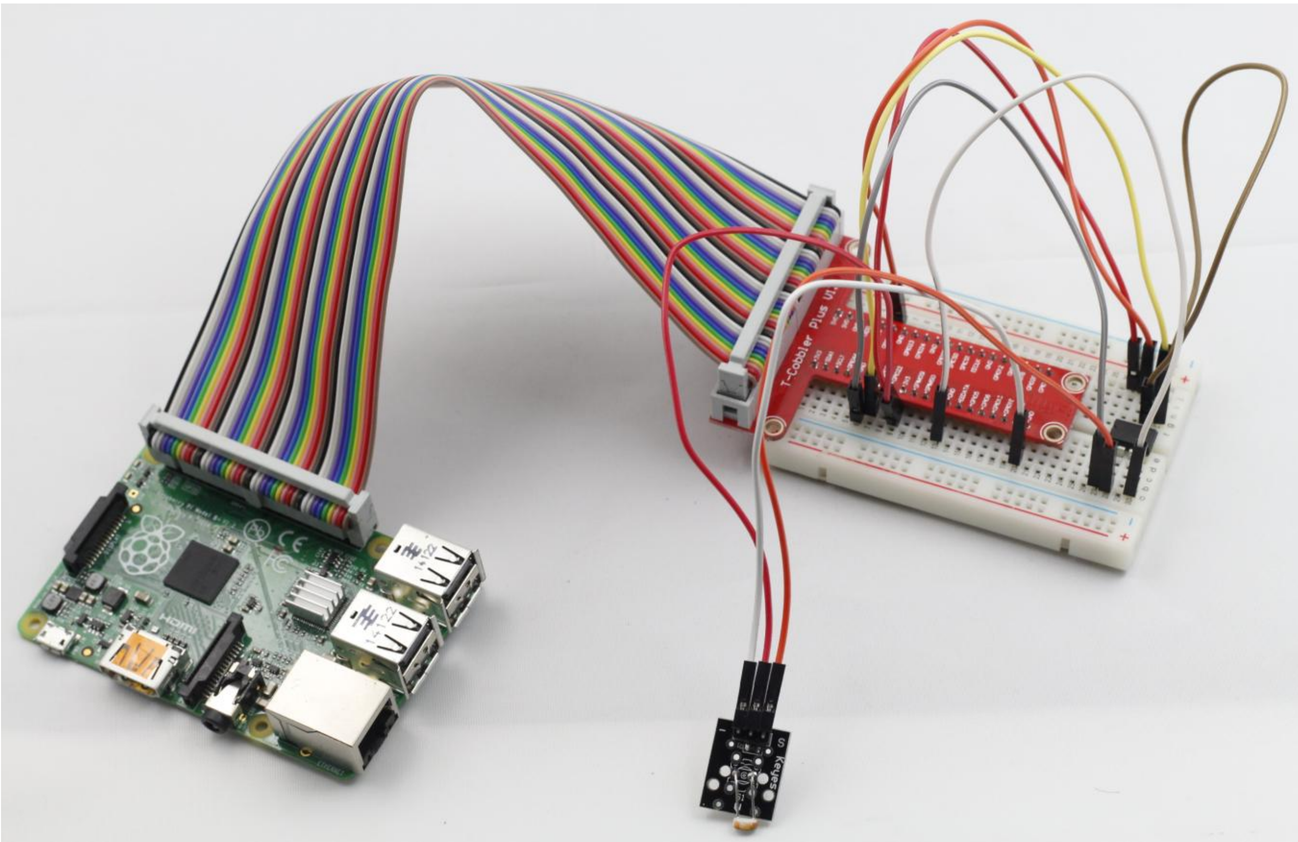
Step 3: Compile the code

gcc photoResistor.c -lwiringPi

Step 4: Run the program

./a.out

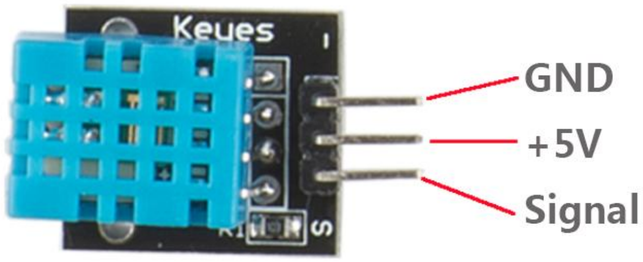
Now, if you change light intensity, you will find the value printed on the screen change accordingly.



Lesson 21 DHT11 Humiture Sensor

Introduction

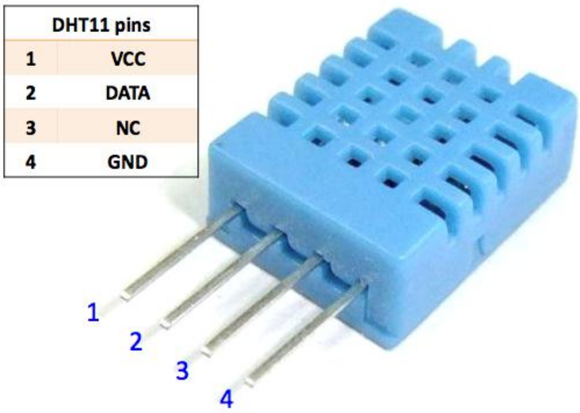
DHT11 digital temperature and humidity sensor (as shown below) is a composite Sensor contains a calibrated digital signal output of the temperature and humidity. Apply a dedicated digital modules collection technology and the temperature and humidity sensing technology to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive sense of wet component and a NTC temperature measurement device, and connected with a high-performance 8-bit microcontroller.



Experimental Conditions

- 1*Raspberry Pi B+
- 1*Network cable (or USB wireless network adapter)
- 1*DHT11 module
- Jumper wires

Experimental Principle



Only three pins are available for use, that is, VCC, GND, and DATA. Communication process begin with DATA line sending starting signal to DHT11, and DHT11 receives the signal and returns a answer signal, then the host receive the answer signal and begin 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 DHT11 datasheet.

Experimental Procedures

Step 1: Connect the circuit

Raspberry Pi B+ DHT11 module

GPIO0 -------------------------------------- 's'

3.3V --------------------------------------- '+'

GND --------------------------------------- '-'

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

Step 3: Compile the code

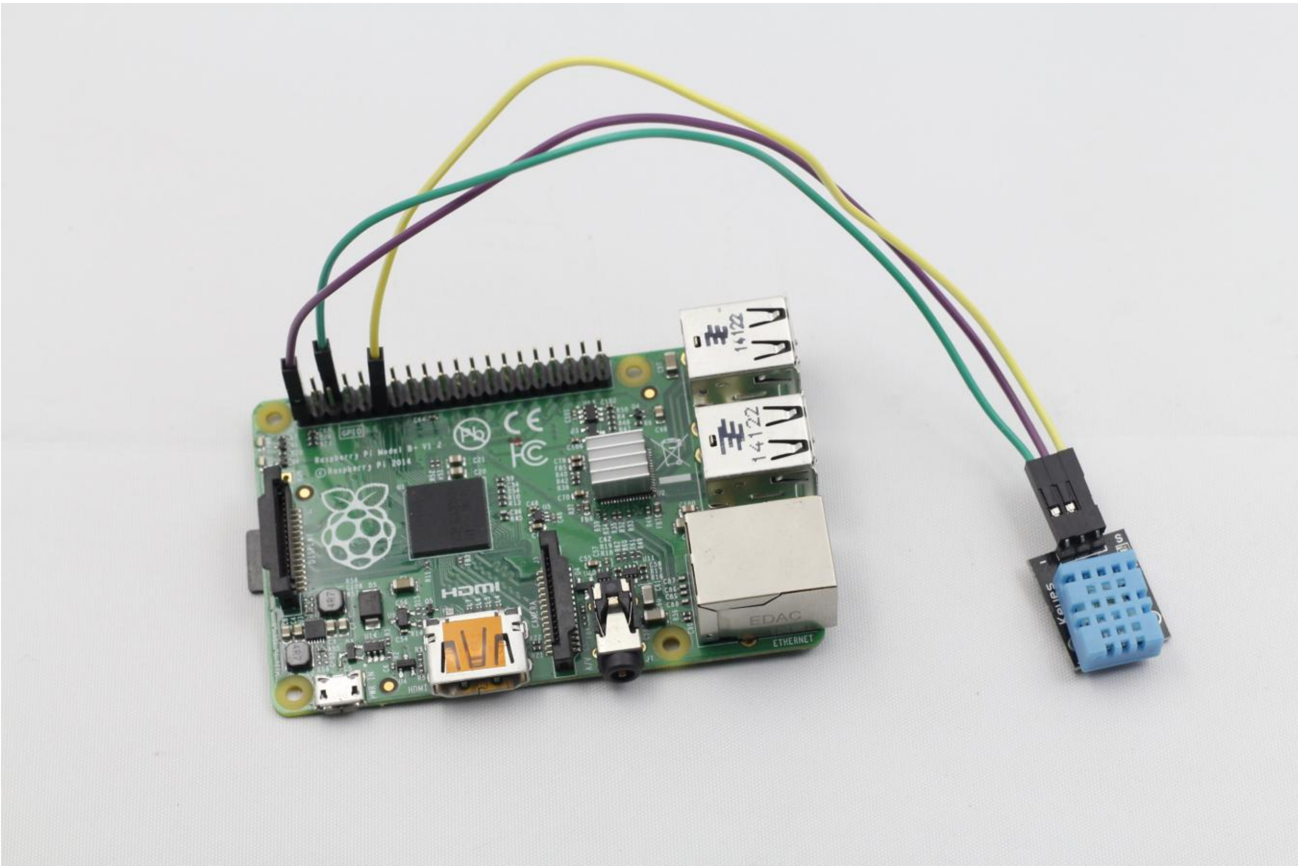
gcc DHT11.c -lwiringPi

Step 4: Run the program

./a.out

Now, press Enter, you will see the related value of humidity and temperature printed on the screen:

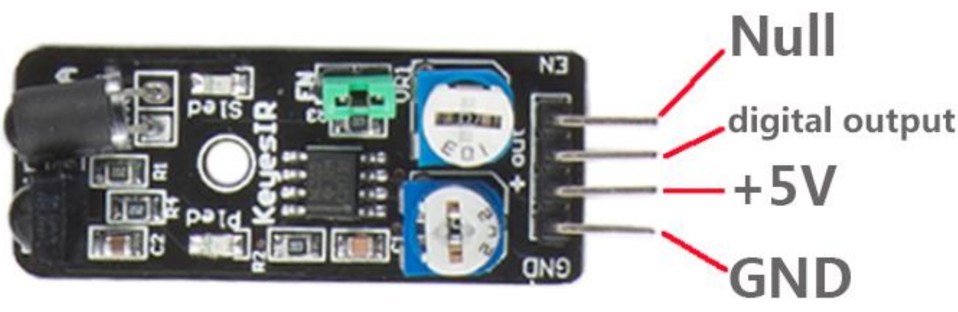
```
root@raspberrypi:/home/Rpi_SensorKit/01_DHT11# ./DHT11
Use GPIO0 to read data!
Enter OS-------
Sorry! Sensor dosent ans!
Congratulations ! Sensor data read ok!
RH :34.0
TMP:54.0
Congratulations ! Sensor data read ok!
RH :34.0
TMP:27.0
Congratulations ! Sensor data read ok!
RH :35.0
TMP:27.0
```



Lesson 22 Obstacle Avoidance Sensor

Introduction

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



Components

- 1*Raspberry Pi B+
- 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 propagation distance and finally disappear. If there is an obstacle, when infrared ray encounters an obstacle, it will be reflected back to the infrared-receiver. Then the infrared-receiver detects this signal and confirms an obstacle exists in front.

Note: the obstacle avoidance distance of infrared sensor is adjustable, you can adjust the obstacle avoidance distance by adjust the potentiometer.

Experimental Procedures

Step 1: Connect the circuit

Raspberry Pi B+ Obstacle Avoidance Sensor module

GPIO0 -------------------------------------- OUT

3.3V -- '+'

GND -- GND

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

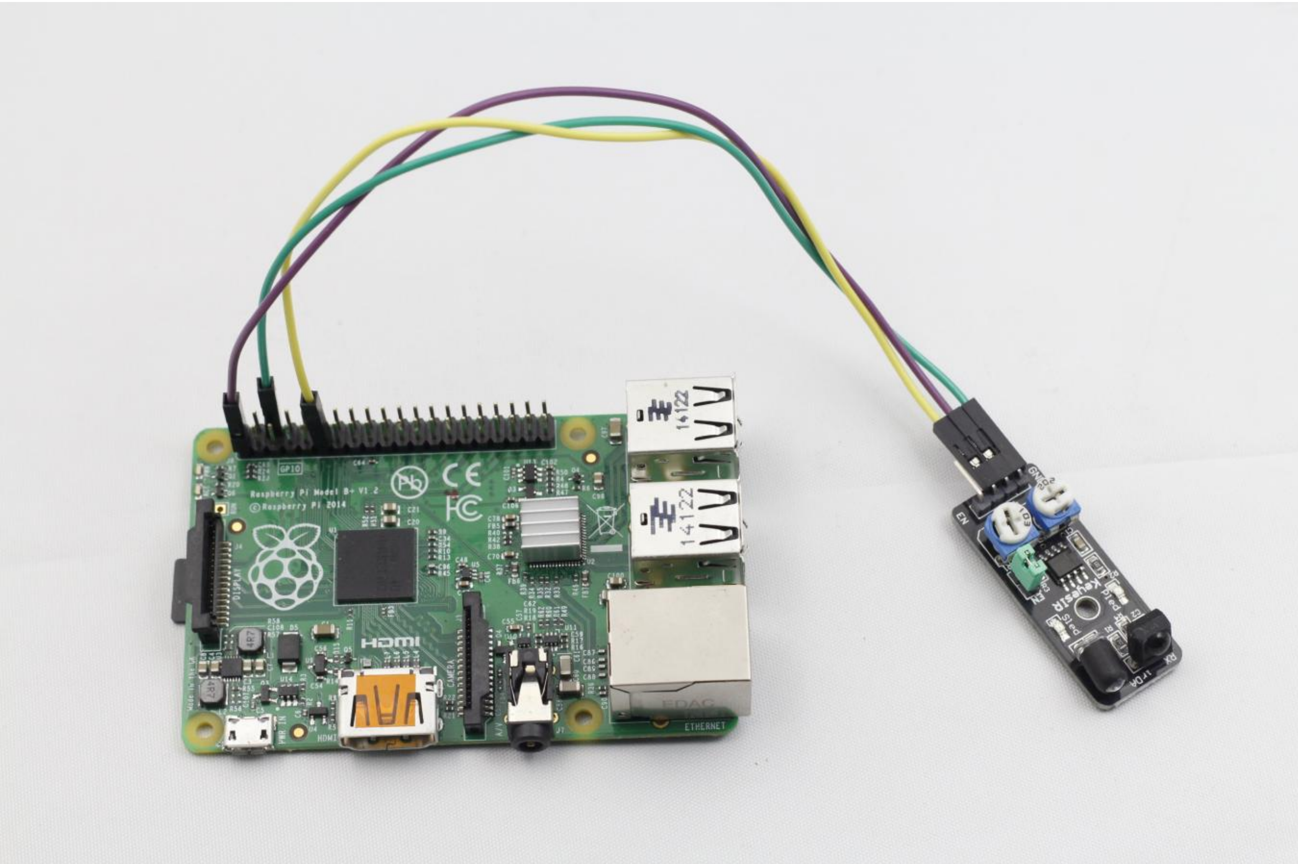
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

Press Enter, if there is an obstacle in front, a string "Detected Barrier!" will display on the screen.



Lesson 23 Tracking Sensor

Introduction

A tracking sensor (as shown below) has the same principle with an Obstacle Avoidance Sensor but has small transmitting power.

Components

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

Experimental Principle

When infrared transmitter emits rays to a piece of paper, if the rays encounter white, it will be reflected and received by the receiver. The pin 'S' will output low; if the rays encounter black lines, it will be absorbed and will not received by the receiver. The pin 'S' will output high.

Experimental Procedures

Step 1: Connect the circuit

Raspberry Pi B+		Tracking sensor module
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 the code

gcc trackSensor.c -lwiringPi

Step 4: Run the program

./a.out

Press Enter, when the tracking sensor encounters black lines, a string "Black Line is Detected" will display on the screen.

Lesson 24 Microphone Sensor

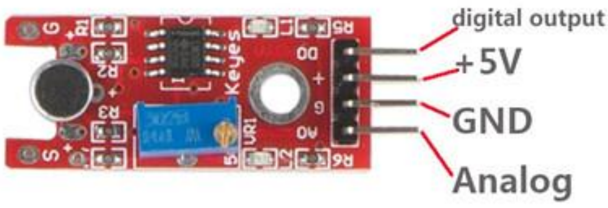
Introduction

There are two kinds of microphone sensors in this kit (as shown below). One, we call it microphone sensor. The other, we call it high-sensitive voice sensor. They are based on the same principle. The only difference between them is their sensitivity. The latter has higher sensitivity. In this experiment, we will take the microphone sensor as an example.

They both have two outputs:

A0: analog output, used to output voltage signals from microphone in real-time

D0: When sound intensity reaches a certain threshold, the sensor outputs high or low level (threshold can be adjusted by potentiometer)



Microphone sensor

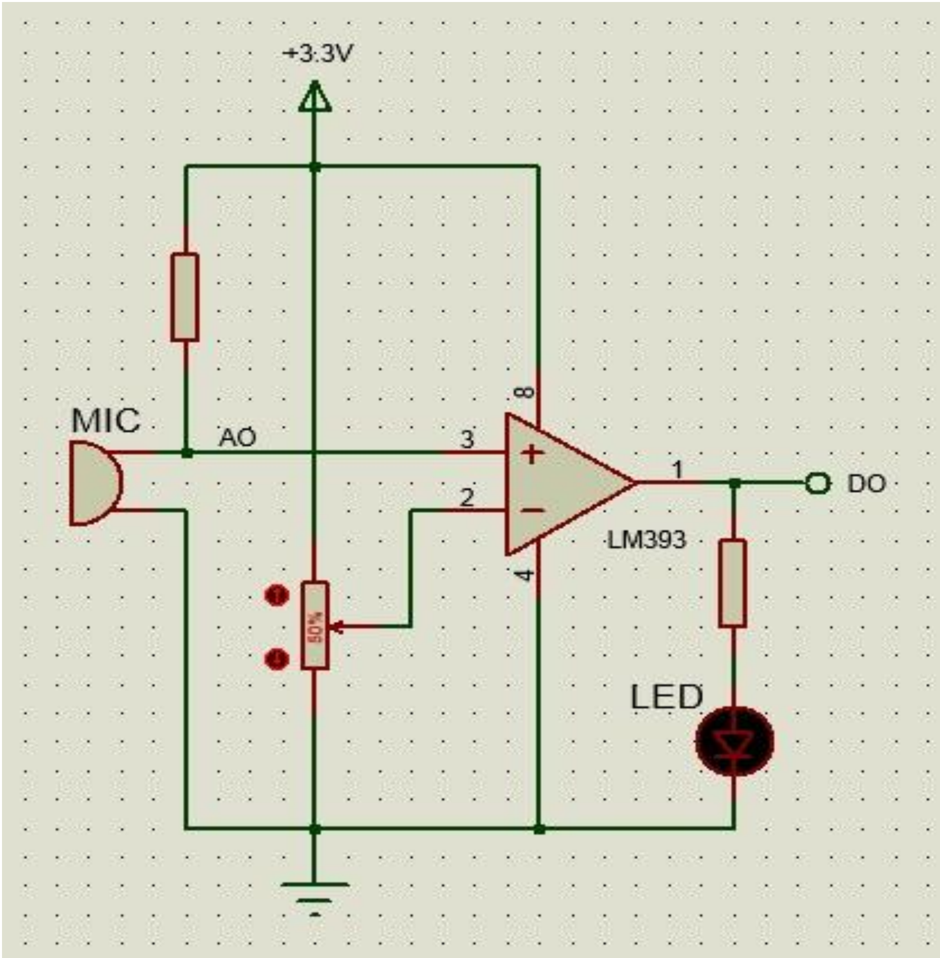
High-sensitive voice sensor

Components

- 1*Raspberry Pi B+
- 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 quantity by ADC and transfer it to MCU to process. The schematic diagram of microphone sensor module is shown as follow:

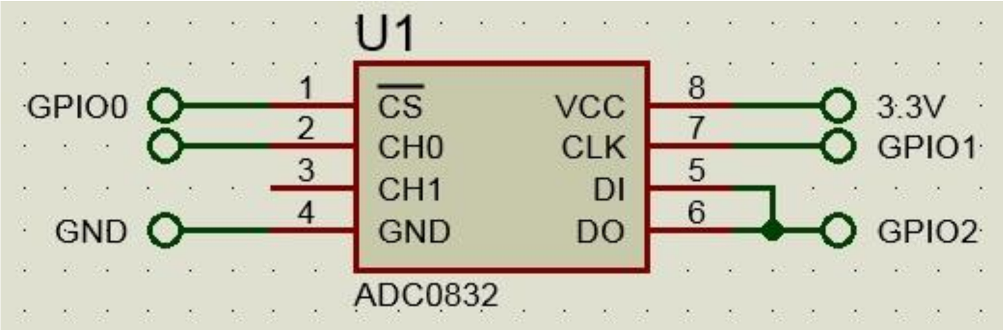


LM393 is a voltage comparator. When the voltage of in-phase terminal (pin 3) is higher than that the inverting terminal (pin 2), output terminal (pin 1) will output high. Otherwise, it outputs low. First, adjust potentiometer to make the voltage for pin 2 of LM393 less than 3.3V. When there is no voice 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 turned on; when there is voice input, the resistance of the microphone decrease, pin 1 outputs low and the LED is turned off. We connect pin 1 to GPIO3 of the Raspberry Pi B+ to detect whether a sound is made by programming.

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

Experimental Procedures

Step 1: Connect the circuit according to the following method



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

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

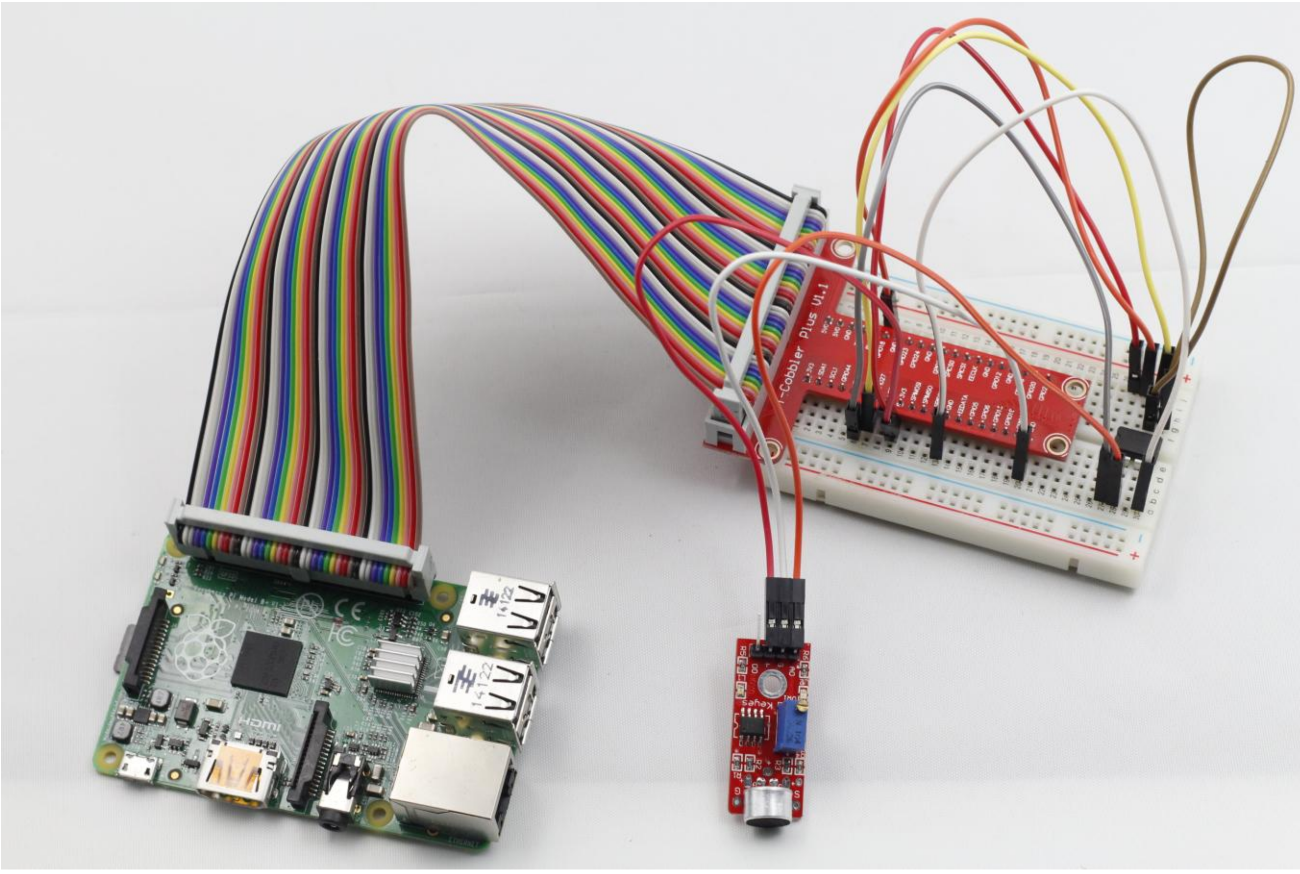
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

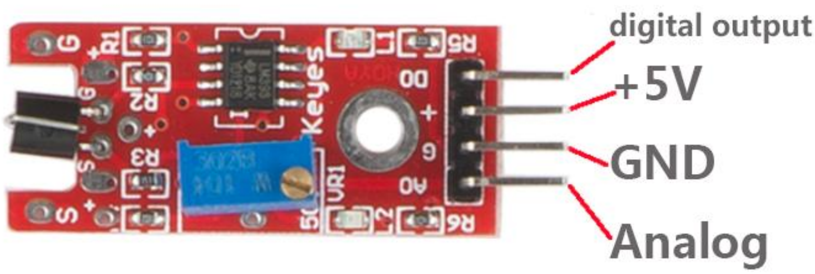
Now, when you speak or blow to the microphone, you will see ”voice in” and ”Current analog value ：***” printed on the screen.



Lesson 25 Metal Touch Sensor

Introduction

A **metal touch sensor** uses the conductivity of human body to operate. When you touch the metal on the base electrode of the transistor, the pin DO will output LOW.



Components

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

Experimental Principle

In this experiment, we will use our fingers to touch the base electrode of the transistor to make it conduct as 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: Connect the circuit

Raspberry Pi B+ Metal touch sensor module

GPIO0 ---------------------------------- 'DO'

3.3V --------------------------------- '+'

GND ---------------------------------- 'G'

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

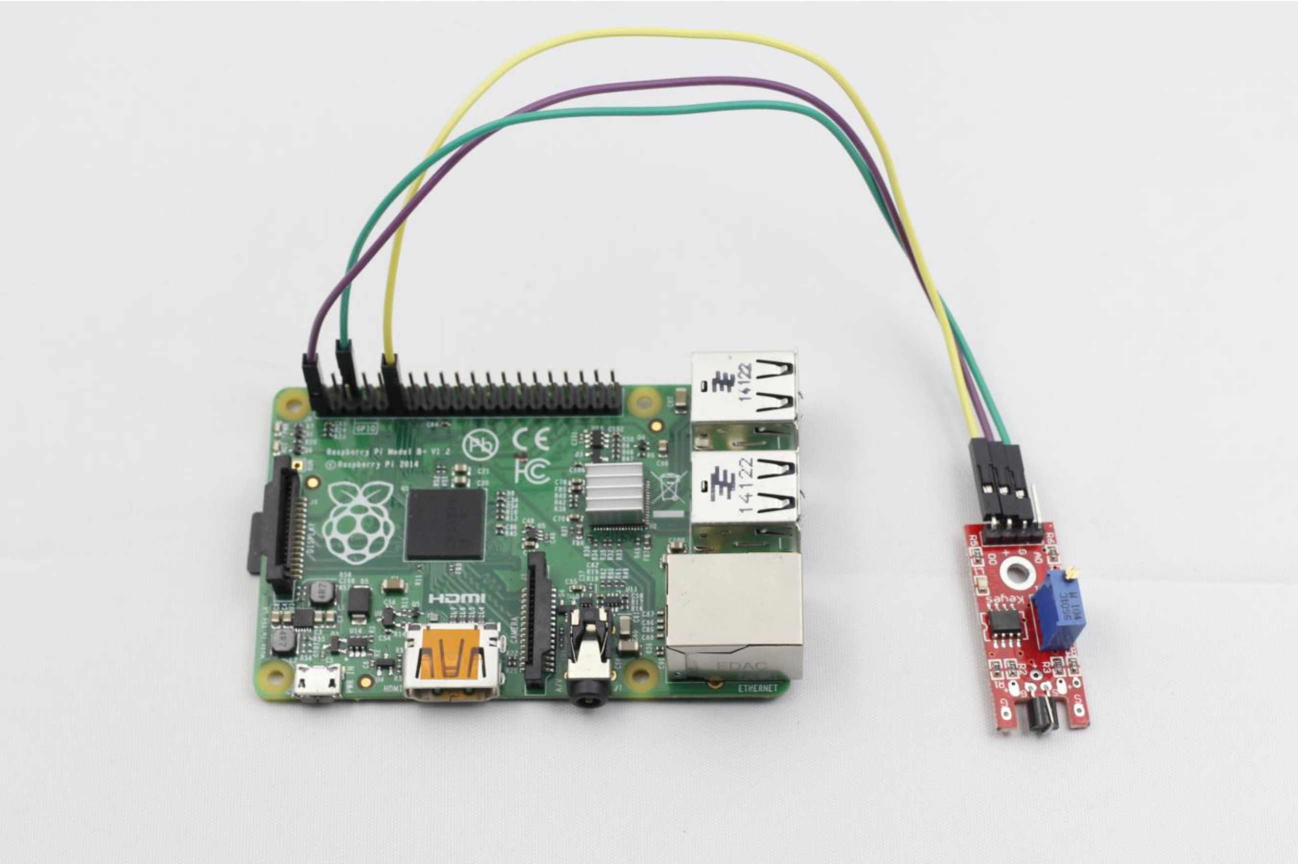
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

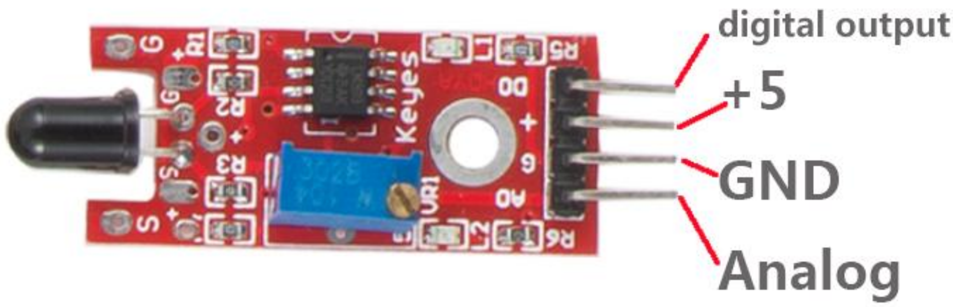
Press Enter, if you touch the base electrode of the transistor, a string “touched” will display on the screen.



Lesson 26 Flame Sensor

Introduction

A flame sensor (as shown below) performs detection by capturing infrared rays with specific wavelengths.



Components

- 1*Raspberry Pi B+
- 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 far-infrared flame sensor. It can detect infrared light with wavelength ranging from 700nm to 1000nm. Far-infrared flame probe converts the strength changes of external infrared light into current changes. And then convert analog quantity into digital quantity. We connect pin DO of flame sensor module to a GPIO of Raspberry Pi B+ to detect flame existence by programming.

Experimental Procedures

Step 1: Connect the circuit

Raspberry Pi B+ Flame sensor module

GPIO0 -------------------------------- 'DO'

3.3V -------------------------------- '+'

GND --------------------------------- ‘G’

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

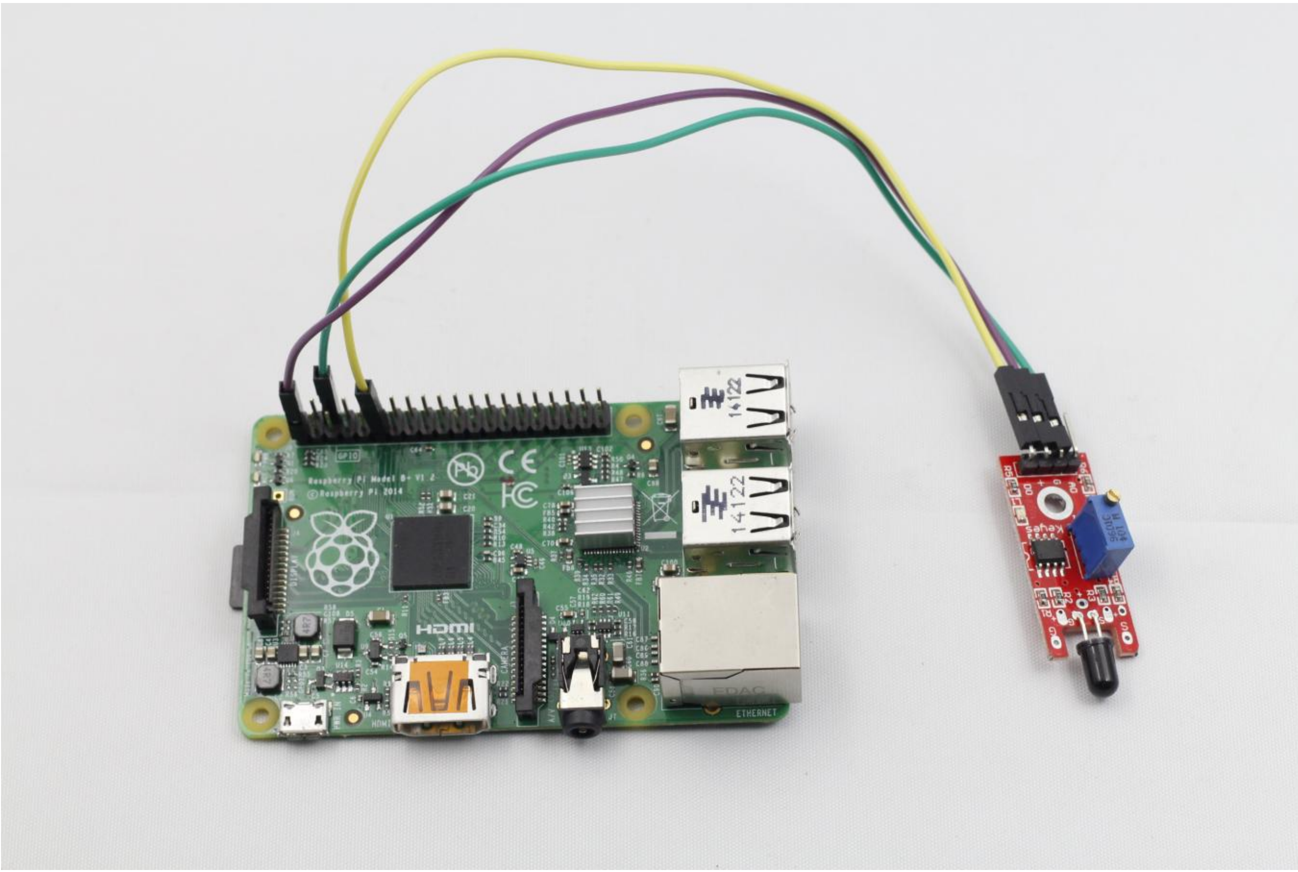
Step 3: Compile the code

gcc flameSensor.c -lwiringPi

Step 4: Run the program

./a.out

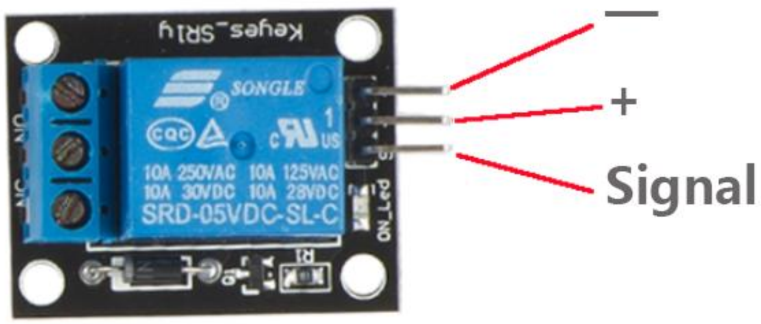
Press Enter. Now, if you ignite a lighter in the range of 80 cm, a string “Detected Flame !” will be displayed on the screen.



Lesson 27 Relay Module

Introduction

Relays are suitable for driving high power electric equipment, such as lights, electric fans and air conditioning. We can use a relay to realize low voltage to control high voltage by connecting it to Raspberry Pi B+.



Components

- 1*Raspberry Pi B+
- 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 because of current saturation. The normally open contact of the relay will be closed, while the normally closed contact of the relay will be broken; when we make it output high level (3.3V), the transistor will be cut off, and the relay will recover to initial state.

Experimental Procedures

Step 1: Connect the circuit

Raspberry Pi B+	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 the code

gcc relay.c -lwiringPi

Step 4: Run the program

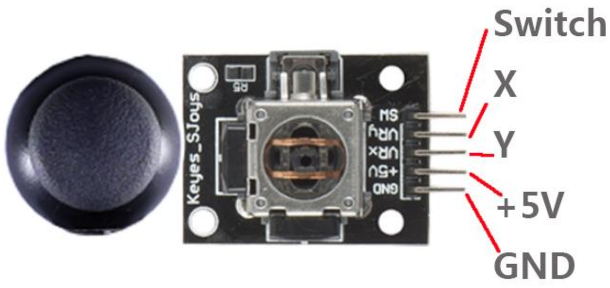
./a.out

Press Enter. Now, you should be able to hear ticking sound. This sound is made by breaking normally closed contact and closing normally open contact. You can attach a high voltage device you want to control, for example, a bulb with 220V voltage, to the output port of relay, so the relay plays a role of automatic switch.

Lesson 28 Joystick PS2

Introduction

The Joystick PS2 has five operation directions: up, down, left, right and press-down.



Components

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

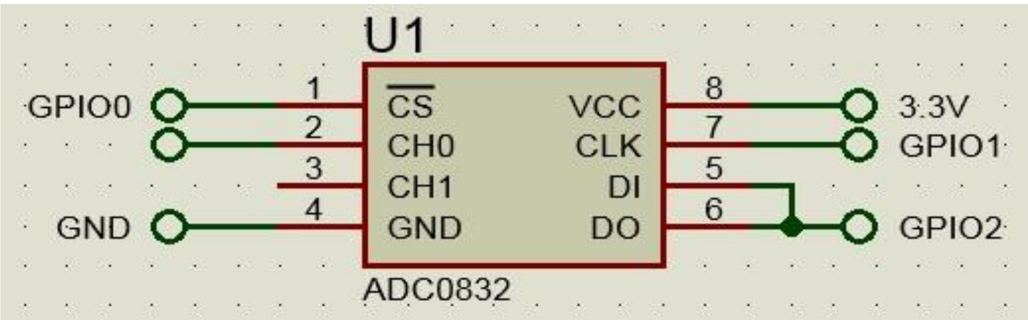
Experimental Principle

This module has two analog outputs (corresponding to X , Y biaxial offsets) and one digital output representing whether it is pressed on Z axis. The module integrates power light and can display 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 B+ to detect the moving direction of the Joystick.

Experimental Procedures

Step 1: Connect 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 B+

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

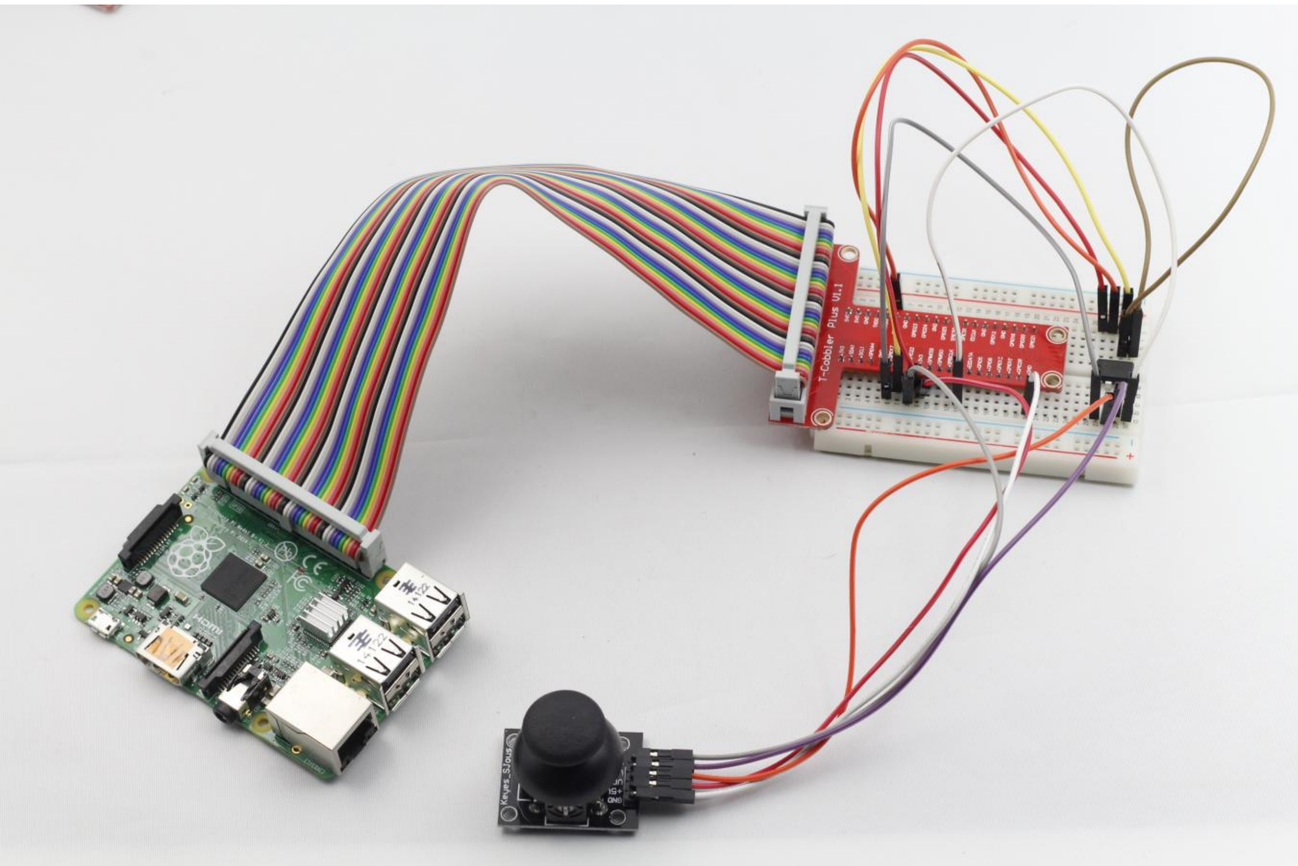
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

Press Enter, if you shake the rocker up, a string “Up” will be displayed on the screen; If you shake down, a string “Down” will be displayed on the screen; If you shake left, a string “Left” will be displayed on the screen; If you shake right, a string “Right” will be displayed on the screen; If you press the button, a value “Pressed” will be displayed on the screen.



Lesson 29 MQ-2 Gas Sensor

Introduction

MQ-2 is a flammable gas and smoke sensor detects the concentrations of combustible gas in the air and outputs its reading as an analog voltage. They are used in gas detecting equipment for smoke and flammable gasses in family and industry or car.

Components

- 1*Raspberry Pi B+
- 1*Active Buzzer module
- 1*ADC0832
- 1*MQ-2 gas sensor module
- Several jumper wires

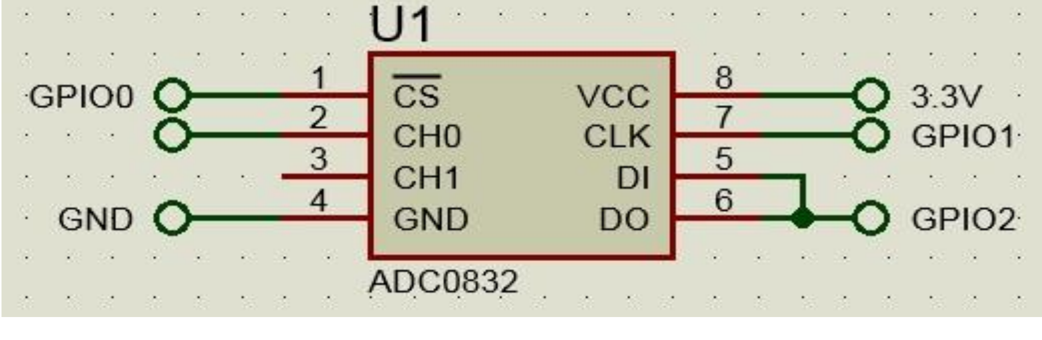
Experimental Principle

MQ-2 type smoke sensor belongs to surface ion type N type semiconductor, which uses tin oxide semiconductor gas sensitive material. When ambient temperature is in 200 ~ 300℃, tin oxide will adsorb oxygen in the air and form oxygen anion adsorption to decrease electron density in semiconductor so as to increase its resistance. When in contact with the smoke, if grain boundary barrier is modulated by the smoke and changed, it could cause surface conductivity change. So you can gain the information of the smoke existence, the higher the smoke concentration is, the greater the conductivity, thus the lower the output resistance is.

In this experiment, if harmful gases reach a certain concentration, the buzzer will make sounds.

Experimental Procedures

Step 1: Connect the circuit



Connect pin ‘OUT’ on MQ-2 gas sensor module to pin CH0 on ADC0832

Connect pin ‘S’ on Buzzer module to GPIO3 on Raspberry Pi B+

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

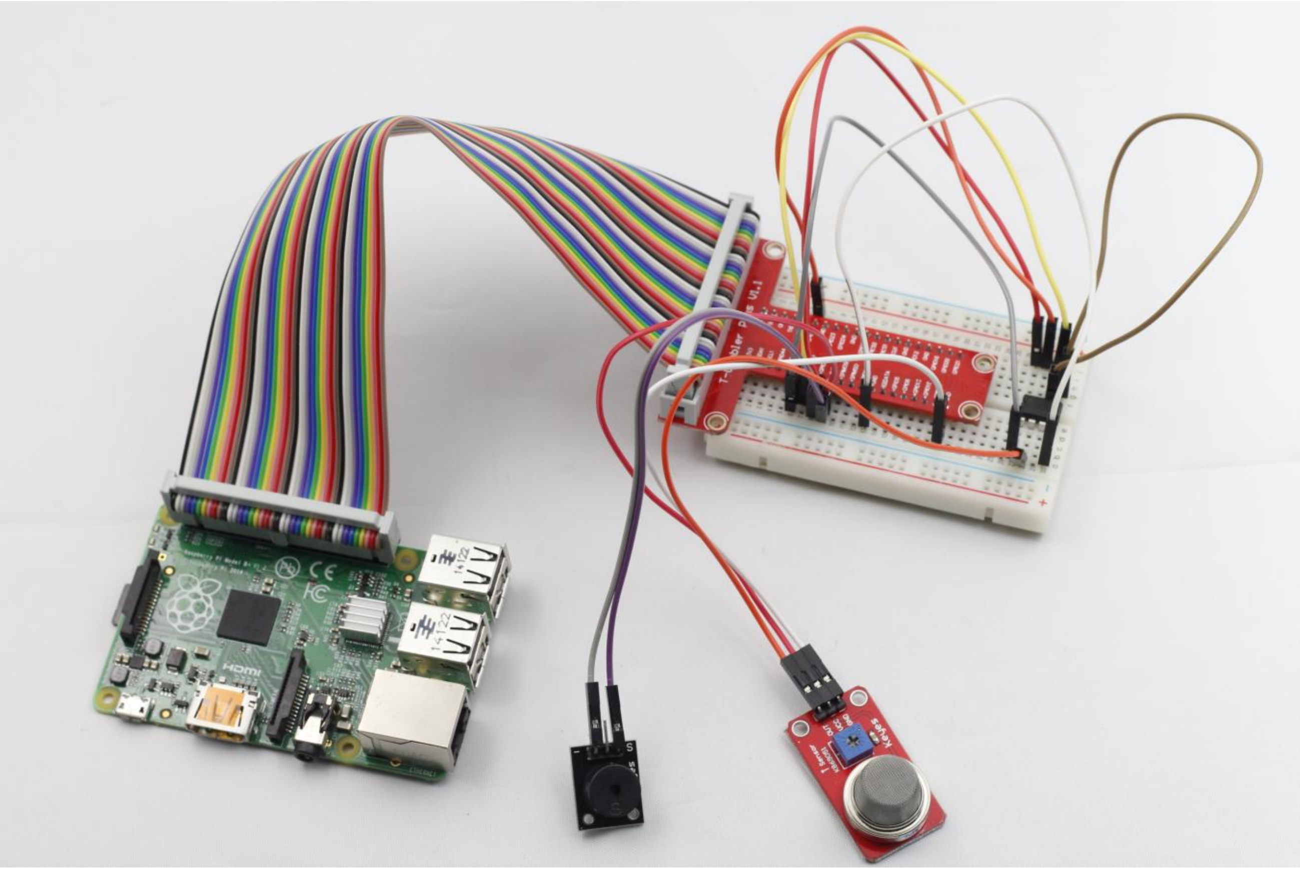
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

Now, when you strike lighters, if harmful gases reach a certain concentration, the buzzer will make sounds.



Lesson 30 Temperature Measurement System

Introduction

After having learnt so many independent modules, let's combine these modules together to make some comprehensive experiment. In this lesson, we will use a RGB, a buzzer, and a DS18B20 to assemble a temperature measurement system.

Components

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

Experimental Principle

When ambient temperature is lower than the lower limit value, the buzzer will make sounds at a relatively low frequency and the LED will emit blue light; when ambient temperature is higher than the upper limit value, the buzzer will make sounds at a relatively high frequency and the LED will emit red light; when ambient temperature is between the lower limit value and the upper limit value, the buzzer will not make sounds and the LED will emit green light.

Note: The lower limit value and the upper limit value here is user-defined and achieved by passing parameters to main function.

Experimental Procedures

Step 1: Connect the circuit

DS18b20 module connection:

DS18b20	**Raspberry Pi B+**
pin 's' ---------------------------------	GPIO7

RGB LED connection:

RGB LED	**Raspberry Pi B+**
'R' --------------------------------------	GPIO0
'G' --------------------------------------	GPIO1
'B' --------------------------------------	GPIO2

Buzzer module connection:

Buzzer module	**Raspberry Pi B+**
pin '-' --	GND
pin 's' --	GPIO3

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

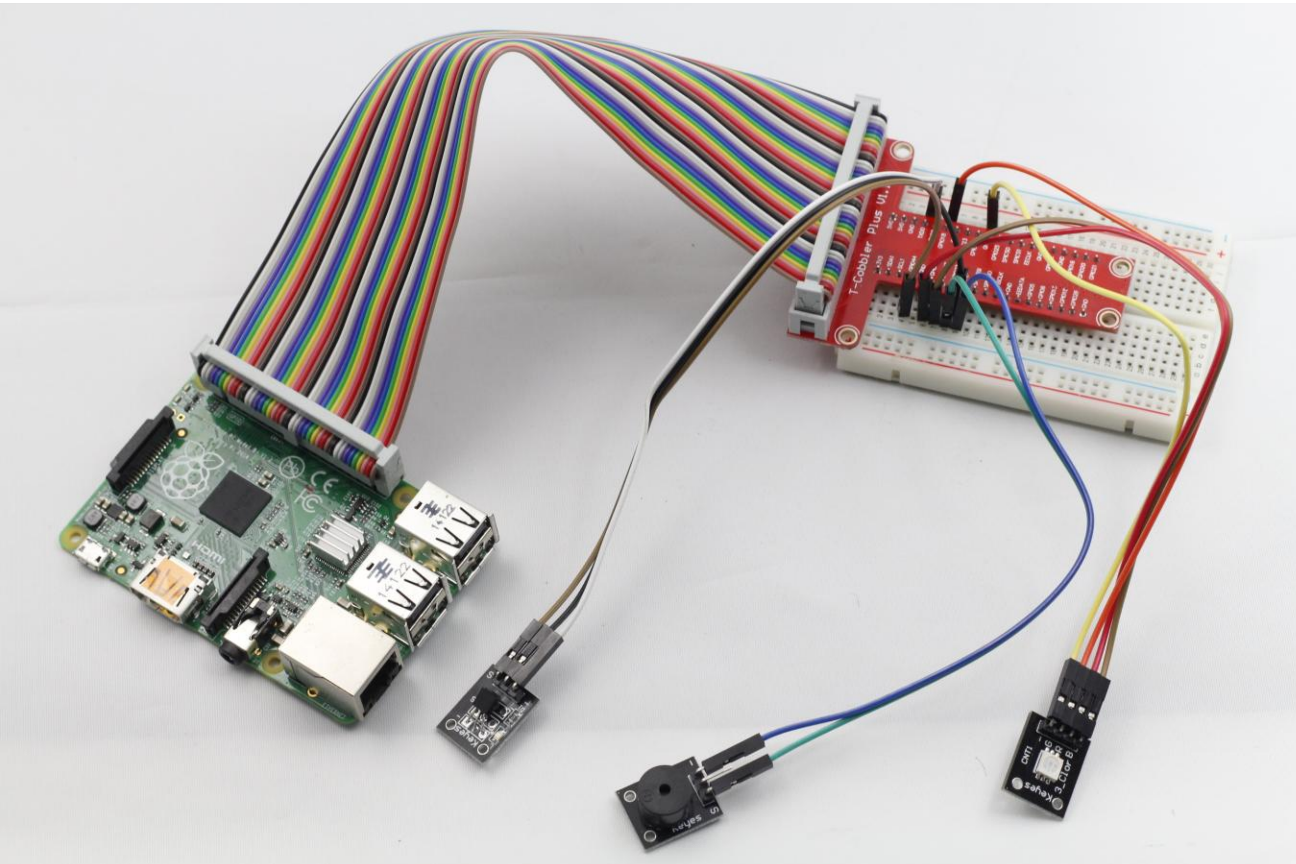
Step 3: Compile the code

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

Step 4: Run the program

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

Now, When ambient temperature is lower than the lower limit value(25℃), you will hear the buzzer make sounds at a relatively low frequency and see the LED emit blue light; when ambient temperature is higher than the upper limit value(30℃), you will hear the buzzer make sounds at a relatively high frequency and see the LED emit red light; when ambient temperature is between the lower limit value and the upper limit value, you will not hear the buzzer make sounds, just see the LED emit green light.



Lesson 31 Intelligent Temperature Measurement System

Introduction

Compared with last experiment, we only add an ADC0832 and a Joystick PS2.

Components

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

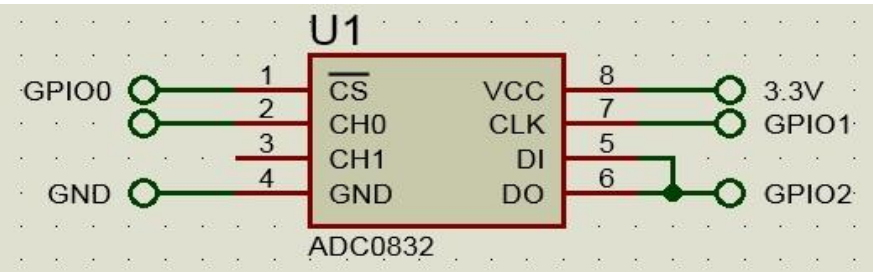
Experimental Principle

It is similar with last experiment. The only difference is that we can adjust the lower limit value and the upper limit value by the Joystick PS2 in the process of program.

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

Experimental Procedures

Step 1: Connect the circuit



Raspberry GPIO0 --------------------- ADC0832 pin ‘CS’

Raspberry GPIO1 --------------------- ADC0832 pin ‘CLK’

Raspberry GPIO2 --------------------- ADC0832 pin ‘DIO’
Joystick PS2 pin ‘x’ -------------------- ADC0832 pin ‘CH0’
Joystick PS2 pin ‘y’-------------------- ADC0832 pin ‘CH1’
Joystick PS2 pin ‘z’-------------------- Raspberry GPIO6
Raspberry GPIO3 --------------------- RGB LED pin ‘R’
Raspberry GPIO4 --------------------- RGB LED pin ‘G’
Raspberry GPIO5 --------------------- RGB LED pin ‘B’
Raspberry GPIO8 --------------------- Buzzer module pin ‘s’
Raspberry GND------------------------ Buzzer module pin ‘-’

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

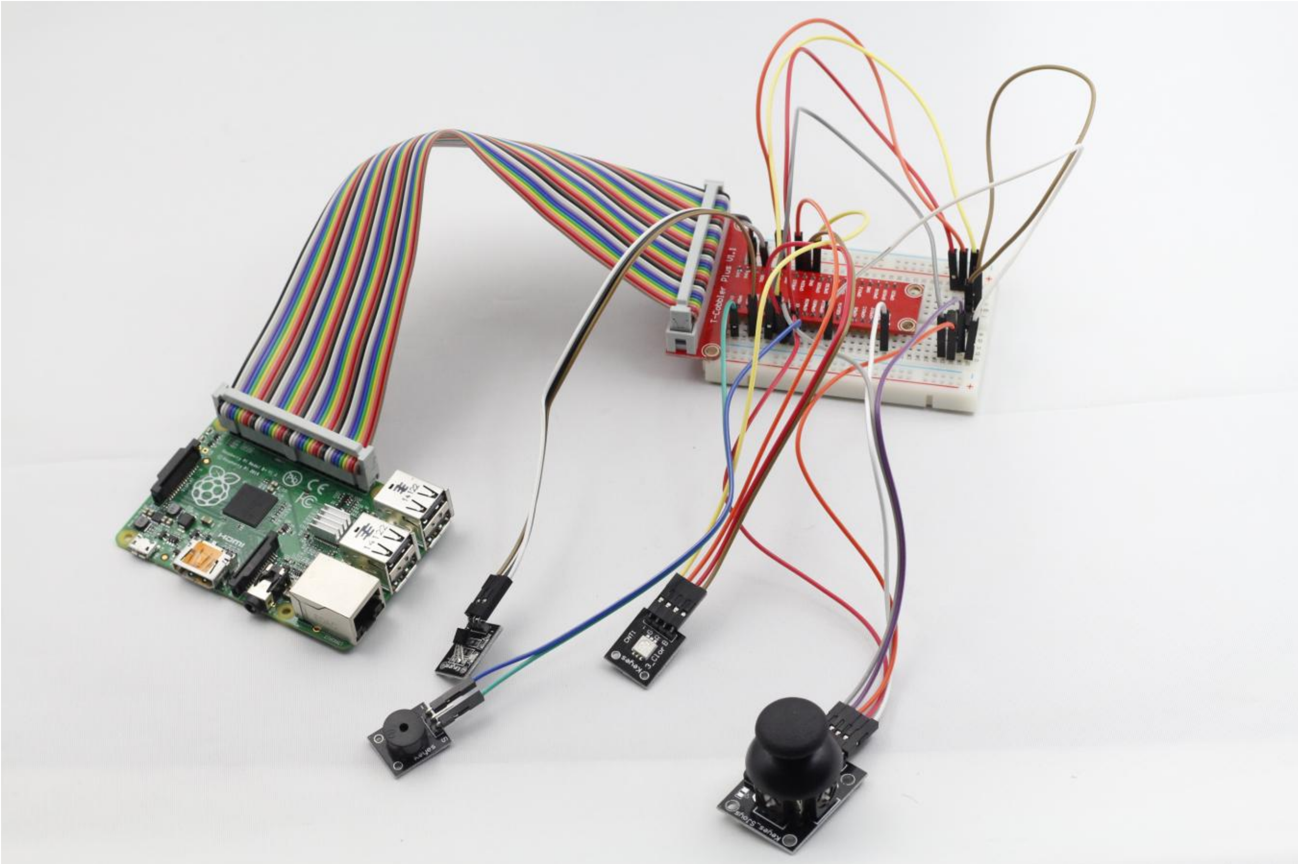
Step 3: Compile the code

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

Step 4: Run the program

```
./a.out
```

Now, you can shake left and right to set the upper limit value, and shake up and down to set the lower limit value. If the ambient temperature reaches the upper limit value or lower limit value, the buzzer will make sounds at a different frequency.



If you have any questions, please send email to support@sunfounder.com. We will reply you asap!

For more tutorials, please visit our website www.sunfounder.com.

If you have great works, welcome to our website to publish them to share with others.

Thanks!

Sunfounder technical support team

SUNFOUNDER

OPEN SOURCE
ELECTRONICS
THE BEST DIY
COMPONENTS
FOR ARDUINO

WWW.SUNFOUNDER.COM