

PS2 Joystick Experiment

Introduction

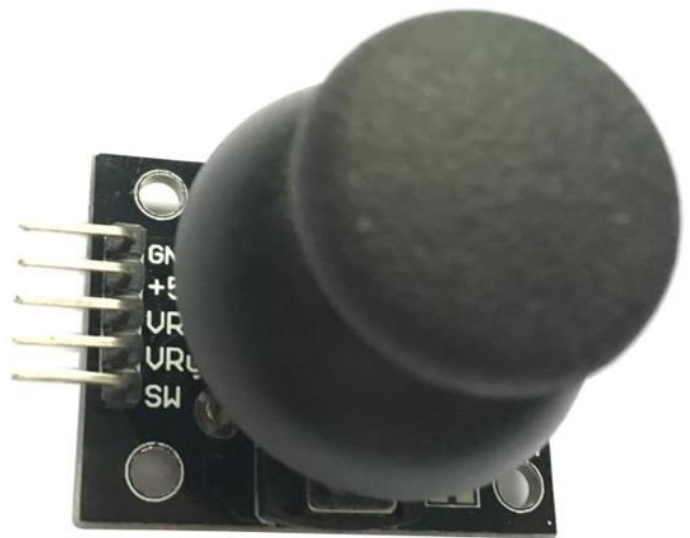
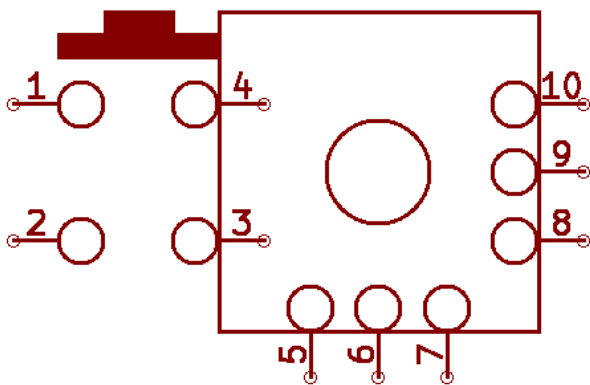
PS2 Joystick Module

Interface: standard interface and 2.54mm pin interface for Electronic building blocks

1、 The cross joystick is two two-way 10K resistors. The Ohms of the joystick varies with the direction of the joystick.

2、 This module uses 5V power supply. In the original state, X and Y read voltage as about 2.5V. Press it down in the direction of the arrow and the reading voltage increases to 5V; Press it down in the opposite direction of the arrow, the reading voltage is decreased till to a minimum of 0V.

3、 The module also has a “downward” button. While the button is pressed, the SW port output the low level voltage.



Joystick Structure

In fact, the joystick is structured very simply. It contains a touch button (z-axis) and two potentiometers (X-axis and Y-axis). The joystick controls movement according to two touch points: one of the touch point goes left and right, the other goes up and down. The joystick movement determines the position of the touch, just like the latitude and longitude of the earth, different positions correspond to different voltages, and the controller may be able to then read different voltage values via the AD sensor to identify specific remote positions.

Analog signals input VRx values from 0 to 1023 from the top of the joystick's VRx, VRy (X and Y axis) pins to indicate the position from left to right, while the values of VRy from 0 to 1023 represent positions from top to bottom, respectively. If the button is not pressed, both values are 512, which is the middle value.

SW (Z) is a digital signal input pin that is connected to a digital port and enable it to pull up resistors.

SW: 1 means not being pressed, and 0 means being pressed.

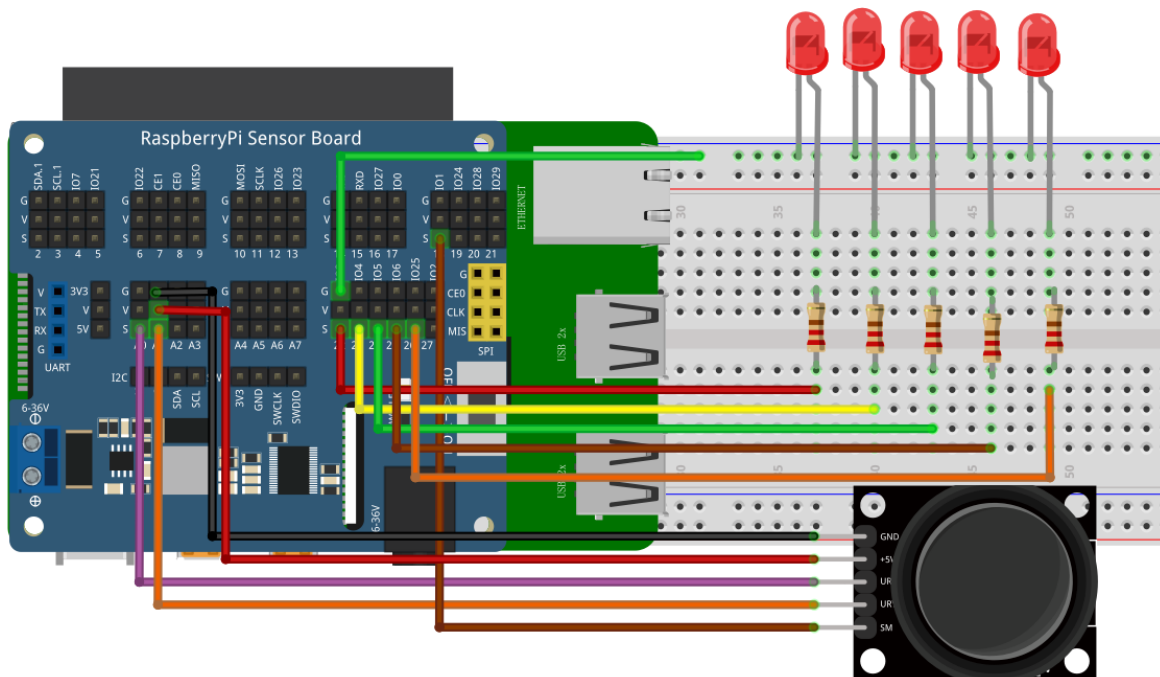
Experimental Purpose

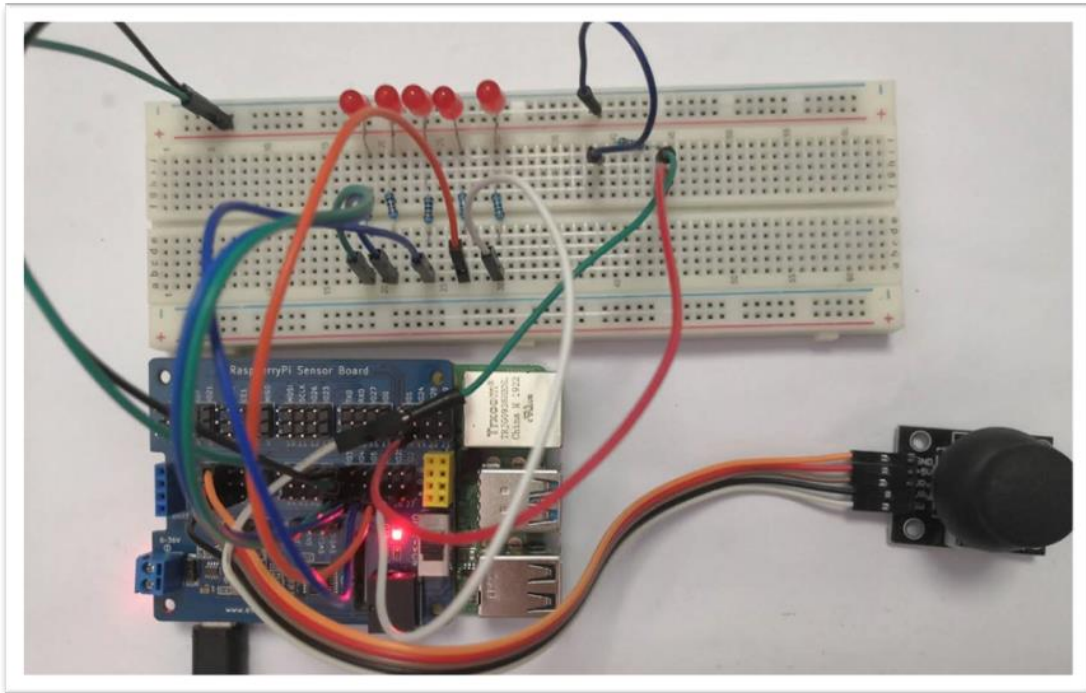
This experiment is designed to control the LED with a joystick. While the joystick is moved up, down, left, right, and it is pressed down, make sure that the corresponding 1-5 LEDs are on respectively.

Component List

- ◆ Raspberry Pi main board
- ◆ Raspberry Pi expansion board
- ◆ Breadboard
- ◆ Cable
- ◆ Joystick * 1
- ◆ LED * 5
- ◆ 220Ω Resistor * 6

Wiring





C++ program

```
#include <stdio.h> //导入基础库
#include <wiringPi.h> //导入树莓派WiringPi编码IO控制库
#include <wiringPiI2C.h> //导入树莓派WiringPi编码I2C控制库

#define JOYSTICK_SW    1
#define LED_ENTER     3    //enter
#define LED_LEFT      4    //left
#define LED_UP        5    //up
#define LED_RIGHT     6    //right
#define LED_DOWN      25   //down

int value_x,value_y,value_sw;

void setup()
{
    wiringPiSetup();
    wiringPiI2CSetup(0x04);
    pinMode(JOYSTICK_SW, INPUT);
    pinMode(LED_ENTER, OUTPUT);
    pinMode(LED_LEFT, OUTPUT);
    pinMode(LED_UP, OUTPUT);
    pinMode(LED_RIGHT, OUTPUT);
    pinMode(LED_DOWN, OUTPUT);
}
```

```
digitalWrite(LED_ENTER, LOW);
digitalWrite(LED_LEFT, LOW);
digitalWrite(LED_RIGHT, LOW);
digitalWrite(LED_UP, LOW);
digitalWrite(LED_DOWN, LOW);

}

int main()
{
    setup();
    while(1)
    {
        value_sw = digitalRead(JOYSTICK_SW);
        value_x = wiringPiI2CReadReg8(0x04, 0x10);
        delay(200);
        value_y = wiringPiI2CReadReg8(0x04, 0x11);
        if(value_x==0)
        {
            digitalWrite(LED_LEFT, HIGH);
            delay(500);
            printf("left");
            digitalWrite(LED_LEFT, LOW);
        }
        if(value_x==255)
        {
            digitalWrite(LED_RIGHT, HIGH);
            delay(500);
            printf("right");
            digitalWrite(LED_RIGHT, LOW);
        }
        if(value_y==0)
        {
            digitalWrite(LED_UP, HIGH);
            delay(500);
            printf("up");
            digitalWrite(LED_UP, LOW);
        }
        if(value_y==255)
        {
            digitalWrite(LED_DOWN, HIGH);
            delay(500);
```

```
        printf("down");
        digitalWrite(LED_DOWN, LOW);
    }
    if(value_sw == 0 )
    {
        delay(200);
        if(value_sw == 0 )
        {
            digitalWrite(LED_ENTER, HIGH);
            delay(500);
            //printf("enter");
            digitalWrite(LED_ENTER, LOW);
        }
    }
}
```

Python program

```
import time
import smbus as smbus
import RPi.GPIO as GPIO

JOYSTICK_SW = 18
LED_ENTER = 22#enter
LED_LEFT = 23#left
LED_UP = 24#up
LED_RIGHT = 25#right
LED_DOWN = 26#down

ADC=smbus.SMBus(1)#Declare to use I2C 1
GPIO.setmode(GPIO.BCM)
GPIO.setup(JOYSTICK_SW, GPIO.IN)
GPIO.setup(LED_ENTER, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(LED_LEFT, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(LED_UP, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(LED_RIGHT, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(LED_DOWN, GPIO.OUT, initial=GPIO.LOW)

while True:
    ADC.write_byte(0x04, 0x10)#Write a byte to the slave
    value_x = ADC.read_byte_data(0x04, 0x10)
    value_y = ADC.read_byte_data(0x04, 0x11)
```

```
value_sw = GPIO.input(JOYSTICK_SW)

time.sleep
if value_x==0:
    GPIO.output(LED_LEFT, True)
    time.sleep(0.5)
    print("left");
GPIO.output(LED_LEFT, False)
if value_x==255:
    GPIO.output(LED_RIGHT, True)
    time.sleep(0.5)
    print("right")
    GPIO.output(LED_RIGHT, False)
if value_y==0:
    GPIO.output(LED_UP, True)
    time.sleep(0.5)
    print("up")
    GPIO.output(LED_UP, False)
if value_y==255:
    GPIO.output(LED_DOWN, True)
    time.sleep(0.5)
    print("down")
    GPIO.output(LED_DOWN, False)
if value_sw == 0:
    GPIO.output(LED_ENTER, True)
    time.sleep(0.5)
    print("enter")
    GPIO.output(LED_ENTER, False)
```

Java program

```
import com.pi4j.wiringpi.I2C;
import com.pi4j.wiringpi.Gpio;
import com.pi4j.wiringpi.GpioInterrupt;
import com.pi4j.wiringpi.GpioInterruptListener;
import com.pi4j.wiringpi.GpioInterruptEvent;
import com.pi4j.wiringpi.GpioUtil;

public class Remote_lever {
    static int JOYSTICK_SW = 1, LED_ENTER = 3, LED_LEFT = 4, LED_UP = 5, LED_RIGHT = 6, LED_DOWN
= 25;
    static int value_x,value_y,value_sw;
    static {
```

```
// setup wiring pi
if (Gpio.wiringPiSetup() == -1) {
    System.out.println(" ==>> GPIO SETUP FAILED");
}
//Gpio.pinMode(JOYSTICK_SW, Gpio.INPUT);
Gpio.pullUpDnControl(JOYSTICK_SW, Gpio.PUD_UP);
Gpio.pinMode(LED_ENTER, Gpio.OUTPUT);
Gpio.pinMode(LED_LEFT, Gpio.OUTPUT);
Gpio.pinMode(LED_UP, Gpio.OUTPUT);
Gpio.pinMode(LED_RIGHT, Gpio.OUTPUT);
Gpio.pinMode(LED_DOWN, Gpio.OUTPUT);
Gpio.digitalWrite(LED_ENTER, Gpio.LOW);
Gpio.digitalWrite(LED_LEFT, Gpio.LOW);
Gpio.digitalWrite(LED_UP, Gpio.LOW);
Gpio.digitalWrite(LED_RIGHT, Gpio.LOW);
Gpio.digitalWrite(LED_DOWN, Gpio.LOW);
}

public static void main(String args[]) throws InterruptedException{

    int fd = I2C.wiringPiI2CSetup(0x04);
    for (;;) {
        value_sw = Gpio.digitalRead(JOYSTICK_SW);
        value_x = I2C.wiringPiI2CReadReg16(fd, 0x10);
        value_y = I2C.wiringPiI2CReadReg16(fd, 0x11);

        if(value_x >= 0 && value_x <= 1500) {
            System.out.println(" left");
            Gpio.digitalWrite(LED_LEFT, Gpio.HIGH);
            Gpio.delay(500);
            Gpio.digitalWrite(LED_LEFT, Gpio.LOW);
        }

        if(value_x >= 3500 && value_x <= 4095) {
            System.out.println(" right"); //printf("right");
            Gpio.digitalWrite(LED_RIGHT, Gpio.HIGH);
            Gpio.delay(500);
            Gpio.digitalWrite(LED_RIGHT, Gpio.LOW);
        }

        if(value_y >= 0 && value_y <= 1500) {
            System.out.println(" up");//printf("up");
            Gpio.digitalWrite(LED_UP, Gpio.HIGH);
        }
    }
}
```

```
Gpio.delay(500);
Gpio.digitalWrite(LED_UP, Gpio.LOW);
}

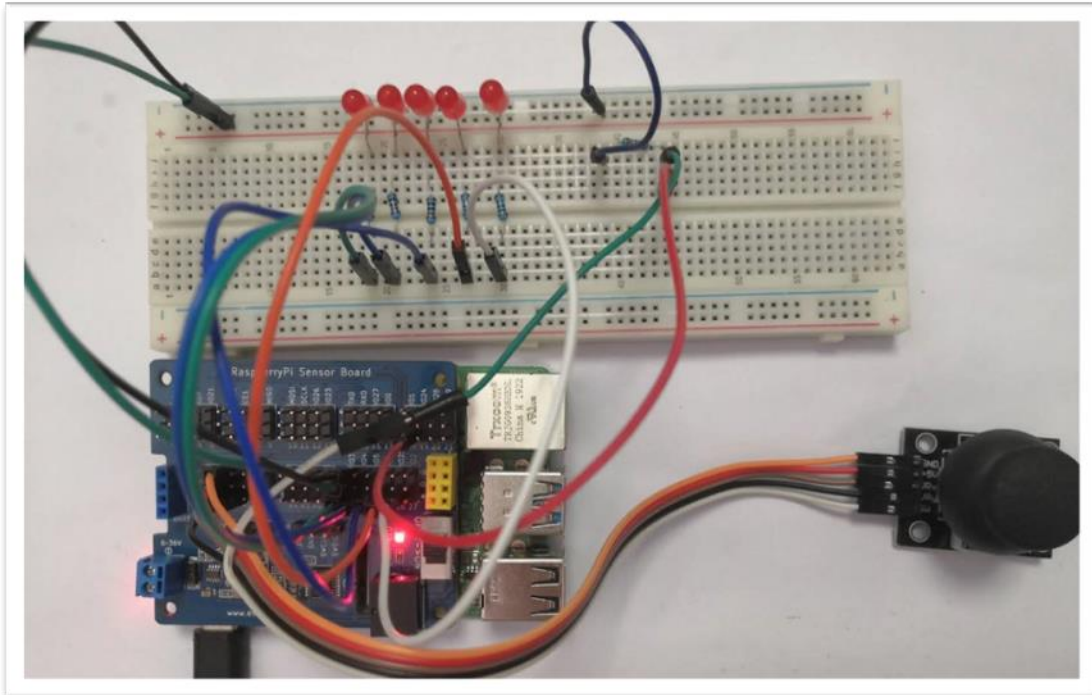
if(value_y >= 3500 && value_y <= 4095) {
    System.out.println(" down");//printf("down");
    Gpio.digitalWrite(LED_DOWN, Gpio.HIGH);
    Gpio.delay(500);
    Gpio.digitalWrite(LED_DOWN, Gpio.LOW);
}

if(value_sw == 0 ) {
    Gpio.delay(200);
    if(value_sw == 0 ) {
        System.out.println("enter");//printf("enter");
        Gpio.digitalWrite(LED_ENTER, Gpio.HIGH);
        Gpio.delay(500);
        Gpio.digitalWrite(LED_ENTER, Gpio.LOW);
    }
}

}

}
```


Experimental Effect



The Joystick moves up, down, left, right, and the “downward” button control 5 LEDs.