

18.Analog_Joystick_Module

Introduction

In this lesson, we will read the output data of Joystick and print it to the screen.

Hardware Required

- ✓ 1 * Raspberry Pi
- ✓ 1 * T-Extension Board
- ✓ 1 * Joystick
- ✓ 1 * 40-pin Cable
- ✓ 1 * PCF8591
- ✓ Several Jumper Wires
- ✓ 1 * Breadboard

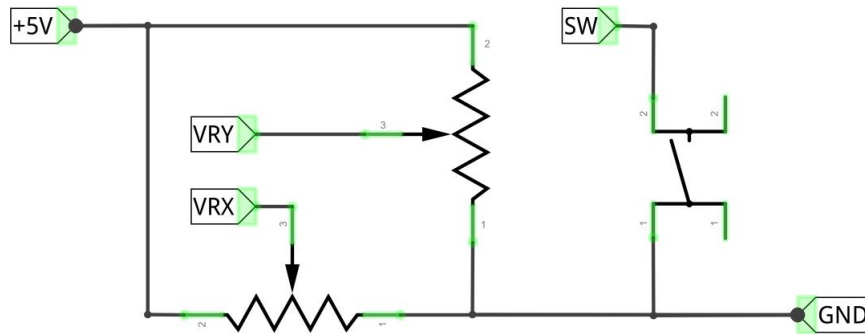
Principle

Analog Joystick Module

Joystick is a kind of sensor used with your fingers, which is widely used in gamepad and remote controller. It can shift in direction Y or direction X at the same time. And it can also be pressed in direction Z.

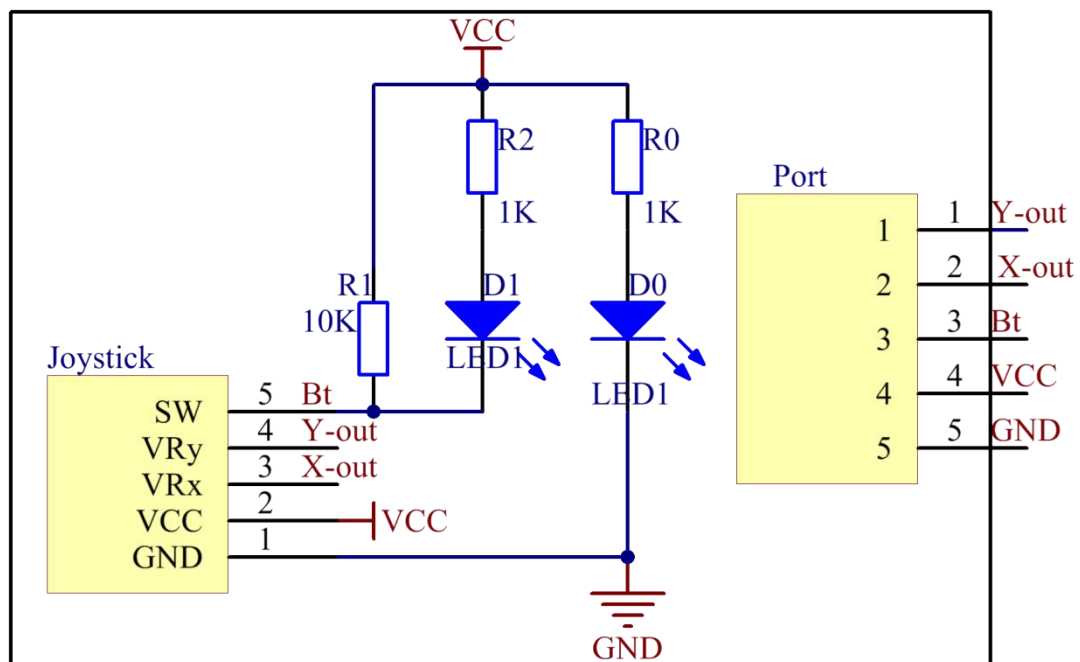


Two rotary potentiometers inside the joystick are set to detect the shift direction of finger, and a push button in vertical direction is set to detect the action of pressing.



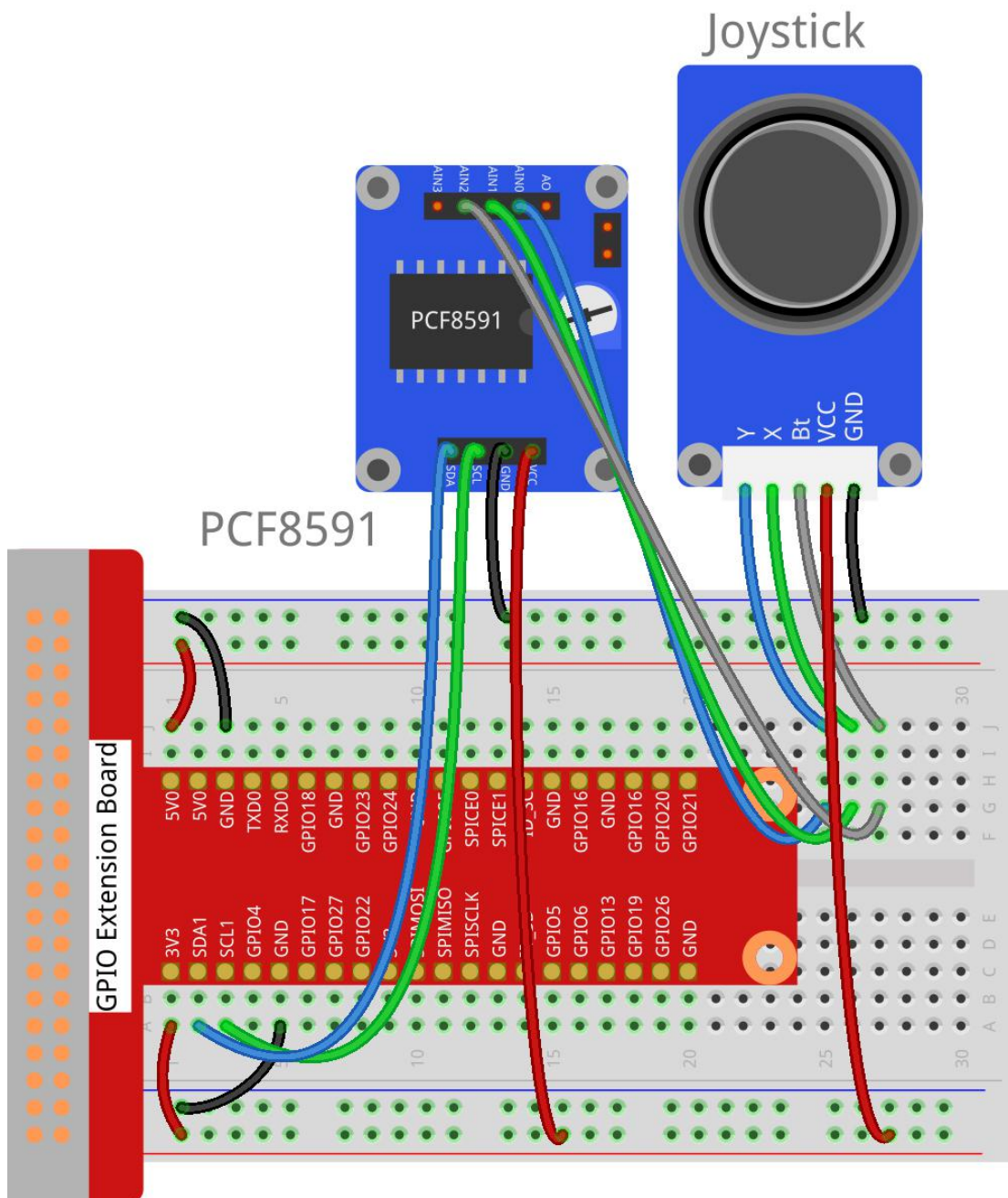
When read the data of joystick, there are some different between axis: data of X and Y axis is analog, which need to use ADC. Data of Z axis is digital, so you can directly use the GPIO to read, or you can also use ADC to read.

Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.



For C Language Users

Step 2: Open the code file.

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/C/18.Analog_Joystick
```

Step 3: Compile the code.

```
gcc 18.Analog_Joystick.c -o Analog_Joystick.out -lwiringPi
```

Step 4: Run the executable file above.

```
sudo ./Analog_Joystick.out
```

After Program is executed, the terminal window will print out the data of 3 axes X, Y, Z.

And shifting the Joystick or pressing it will make those data change.

Code

```
#include <stdio.h>

#include <wiringPi.h>

#include <pcf8591.h>

#define PCF      120

#define uchar    unsigned char

int AIN0 = PCF + 0;

int AIN1 = PCF + 1;

int AIN2 = PCF + 2;

//char *state[7] = {"home", "up", "down", "left", "right", "pressed"};

int direction(){

    int x, y, b;

    int tmp;

    x = analogRead(AIN1);

    y = analogRead(AIN0);

    b = analogRead(AIN2);

    if (y == 0)

        tmp = 1;    // up

    if (y == 255)

        tmp = 2;    // down

    if (x == 255)

        tmp = 3;    // left

    if (x == 0)

        tmp = 4;    // right

    if (y>=125 && b == 0)

        tmp = 5;    // button preesd

    if (x-125<15 && x-125>-15 && y-125<15 && y-125>-15 && b == 255)
```

```

        tmp = 0;      // home position

    return tmp;
}

int main (void)
{
    int tmp;
    int status = 0;
    wiringPiSetup ();

    // Setup pcf8591 on base pin 120, and address 0x48
    pcf8591Setup (PCF, 0x48);

    while(1) // loop forever
    {
        tmp = direction();
        if (tmp != status)
        {
            switch(tmp)
            {
                case 0: printf("home\n");break;
                case 1: printf("up\n");break;
                case 2: printf("down\n");break;
                case 3: printf("left\n");break;
                case 4: printf("right\n");break;
                case 5: printf("pressed\n");break;
            }
            status = tmp;
        }
    }

    return 0 ;
}

```

Code Explanation

```
int direction(){
    int x, y, b;
    int tmp;
    x = analogRead(AIN1);
    y = analogRead(AIN0);
    b = analogRead(AIN2);
    if (y == 0)
        tmp = 1;    // up
    if (y == 255)
        tmp = 2;    // down
    if (x == 255)
        tmp = 3;    // left
    if (x == 0)
        tmp = 4;    // right
    if (y>=125 && b == 0)
        tmp = 5;    // button preesd
    if (x-125<15 && x-125>-15 && y-125<15 && y-125>-15 && b == 255)
        tmp = 0;    // home position
}
```

Configure Z_Pin to pull-up input mode. In while cycle of main function, use analogRead () to read the value of axis X and Y and use digitalWrite () to read the value of axis Z, then print them out.

For Python Language Users

Step 2: Open the code file.

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/Python
```

Step 3: Run.

```
sudo python3 18.Analog_Joystick.py
```

After Program is executed, the terminal window will print out the data of 3 axes X, Y, Z.

And shifting the Joystick or pressing it will make those data change.

Code

```
import PCF8591 as ADC

import time

def setup():
    ADC.setup(0x48)                # Setup PCF8591
    global state

def direction():    #get joystick result
    state = ['home', 'up', 'down', 'left', 'right', 'pressed']
    i = 0

if ADC.read(0) <= 5:
    i = 1    #up
    if ADC.read(0) >= 250:
        i = 2    #down
    if ADC.read(1) >= 250:
        i = 3    #left
    if ADC.read(1) <= 5:
        i = 4    #right
    if ADC.read(1) >= 250 and ADC.read(2) == 0:
        i = 5    # Button pressed
    if ADC.read(0) - 125 < 15 and ADC.read(0) - 125 > -15 and ADC.read(1) - 125 < 15
and ADC.read(1) - 125 > -15 and ADC.read(2) == 255:
        i = 0

    return state[i]

def loop():
    status = ""
    while True:
        tmp = direction()
        if tmp != None and tmp != status:
```

```
        print tmp
        status = tmp

def destroy():
    pass

if __name__ == '__main__':      # Program start from here
    setup()
    try:
        loop()
    except KeyboardInterrupt:    # When 'Ctrl+C' is pressed, the child program
destroy() will be executed.#
    destroy()
```

Phenomenon Picture

