**Rotary Encoder**

Overview

The purpose of this course will use the Raspberry Pi to control the rotary encoder and obtain the value of the rotary encoder.

The experimental materials

Raspberry Pi \*1

Breadboard \*1

Rotary encoder \*1

Dupont Line

The Preparatory Work

Install the python interpreter in your Raspberry Pi system.

2. Install the rpi.gpio library in your Raspberry Pi system.

3. Install wiring Pi library in your Raspberry Pi system.

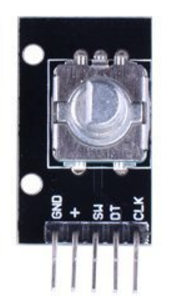
For more details, please refer to the attachment for installing the python interpreter and corresponding libraries in the Raspberry Pi system.

Product Description

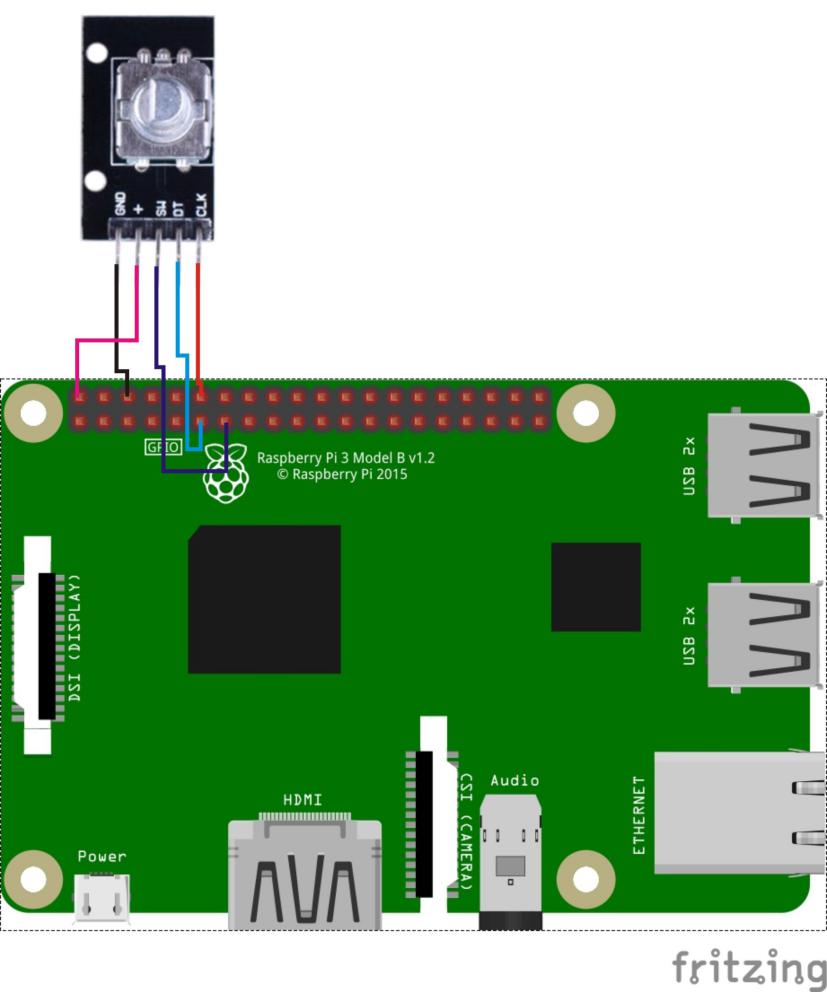
Brief Introduction

The rotary encoder can count the number of output pulses during rotation in the forward and reverse directions by rotation. The rotation count is not like a potentiometer, this rotation count is unlimited. With the key on the rotary encoder, you can reset to the initial state and counting from 0.

Working principle: The incremental encoder is a rotating sensor that converts the rotational displacement into a series of digital pulse signals. These pulses are used to control angular displacement.



Wiring diagram



**The Sample Code**

1. **Python Code**

#!/usr/bin/env python

import RPi.GPIO as GPIO

import time

RoAPin = 11

RoBPin = 12

BtnPin = 13

globalCounter = 0

flag = 0

Last\_RoB\_Status = 0

Current\_RoB\_Status = 0

def setup():

GPIO.setmode(GPIO.BOARD)

GPIO.setup(RoAPin, GPIO.IN)

GPIO.setup(RoBPin, GPIO.IN)

GPIO.setup(BtnPin, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

GPIO.add\_event\_detect(BtnPin, GPIO.FALLING, callback=btnISR)

def rotaryDeal():

global flag

global Last\_RoB\_Status

global Current\_RoB\_Status

global globalCounter

Last\_RoB\_Status = GPIO.input(RoBPin)

while(not GPIO.input(RoAPin)):

Current\_RoB\_Status = GPIO.input(RoBPin)

flag = 1

if flag == 1:

flag = 0

if (Last\_RoB\_Status == 0) and (Current\_RoB\_Status == 1):

globalCounter = globalCounter - 1

if (Last\_RoB\_Status == 1) and (Current\_RoB\_Status == 0):

globalCounter = globalCounter + 1

def btnISR(channel):

global globalCounter

globalCounter = 0

def loop():

global globalCounter

tmp = 0 # Rotary Temperary

while True:

rotaryDeal()

if tmp != globalCounter:

print 'globalCounter = %d' % globalCounter

tmp = globalCounter

def destroy():

GPIO.cleanup()

if \_\_name\_\_ == '\_\_main\_\_':

setup()

try:

loop()

except KeyboardInterrupt:

destroy()

1. **C Code**

#include <stdio.h>

#include <string.h>

#include <errno.h>

#include <stdlib.h>

#include <wiringPi.h>

#define SWPin 2

#define RoAPin 0

#define RoBPin 1

static volatile int globalCounter = 0 ;

unsigned char flag;

unsigned char Last\_RoB\_Status;

unsigned char Current\_RoB\_Status;

void btnISR(void)

{

globalCounter = 0;

}

void rotaryDeal(void)

{

Last\_RoB\_Status = digitalRead(RoBPin);

while(!digitalRead(RoAPin))

{

Current\_RoB\_Status = digitalRead(RoBPin);

flag = 1;

}

if(flag == 1){

flag = 0;

if((Last\_RoB\_Status == 0)&&(Current\_RoB\_Status == 1)){

globalCounter ++;

}

if((Last\_RoB\_Status == 1)&&(Current\_RoB\_Status == 0)){

globalCounter --;

}

}

}

int main(void)

{

int temp =0;

if(wiringPiSetup() < 0)

{

fprintf(stderr, "Unable to setup wiringPi:%s\n",strerror(errno));

return 1;

}

pinMode(SWPin, INPUT);

pinMode(RoAPin, INPUT);

pinMode(RoBPin, INPUT);

pullUpDnControl(SWPin, PUD\_UP);

if(wiringPiISR(SWPin, INT\_EDGE\_FALLING, &btnISR) < 0)

{

fprintf(stderr, "Unable to init ISR\n",strerror(errno));

return 1;

}

while(1)

{

rotaryDeal();

if(temp != globalCounter)

{

printf("%d\n", globalCounter);

temp = globalCounter;

}

}

return 0;

}

**The Experimental Phenomena:**

After the program is run, when you turn the rotary encoder, the corresponding encoding value is displayed in the command line interface of the Raspberry Pi. If you pressed, the encoding value will become zero.