Reed Switch

 Overview

We will use the Raspberry Pi to catch the signal of the reed switch and use this signal to control the on and off of the LED lamp.

Material Needed

RaspberryPi \*1

Breadboard \*1

ReedSwitch \*1

ADC0832 \*1

Led \*1

Dupont Line

 Preparatory work

1. Install python interpreter in your Raspberry Pi system

2. Install the RPi.GPIO library in your Raspberry Pi system

3. Install the wiringPi library in your Raspberry Pi system

See the attached “Installing a Python Interpreter and Corresponding Libraries in a Raspberry Pi System” for details.

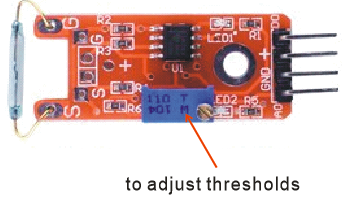
Product Description

Brief Introduction

Reed switches are electrical switches that operate through an applied magnetic field. The basic type is to seal two pieces of magnetic reed in the glass tube. The two pieces overlap, but there is a small gap in the middle. When there is a foreign magnetic field, the two magnetic reeds will be made to contact with each other, and then conduct. Once the magnet is away from the switches, they will return to its original position.

Such switches have small sizes, light weight, and have good corrosion resistance and long life. They are widely used in homes and industries.

In this experiment, a suitable threshold is set by adjusting the resistor on the module. When a magnet approaches the reed switch, the analog signal value of the module AO pin will change. When the analog signal value meets the threshold condition, The level of the DO pin will flip, and the analog value of the AO pin will be converted to digital value through ADC and the digital value will be passed to the Raspberry Pi . When we use the Raspberry Pi to detect that the level of the DO pin has flipped, it will reading the converted digital value, when the value meets certain conditions, the Raspberry Pi will control the LED light to turn on and off.



Characteristic Parameters

◆Using normally opened reed switch

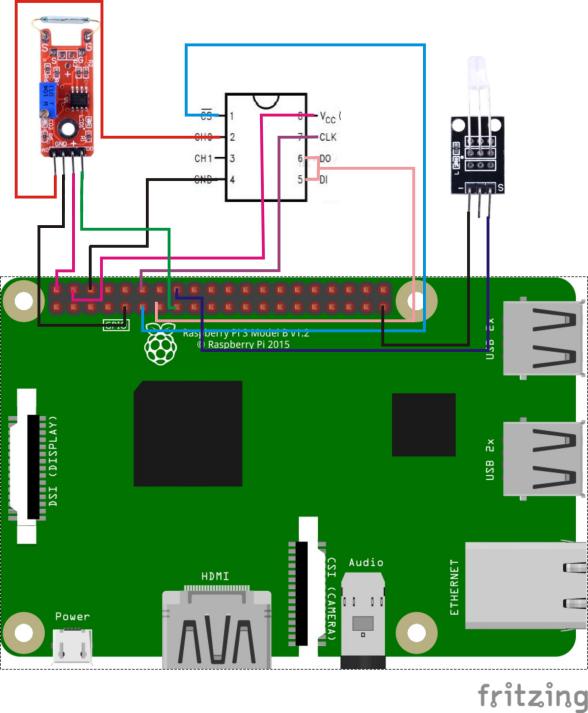
◆Comparator output, clean signal, good waveform, strong driving ability, over 15mA.

◆Working voltage 3.3V-5V

◆Output form : Digital switch output (0 and 1)

◆with a fixed bolt hole, easy to install  
◆Use Wide Voltage LM393 Comparator

Wiring diagram



Sample Code

1. Python Code

#!/usr/bin/env python

import RPi.GPIO as GPIO

import ADC0832

import time

Reed\_DO\_PIN = 15

LedPin = 16

thresholdVal = 100

def init():

GPIO.setmode(GPIO.BOARD)

GPIO.setup(Reed\_DO\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

GPIO.setup(LedPin, GPIO.OUT)

ADC0832.setup()

def loop():

while True:

global digitalVal, analogVal

digitalVal = GPIO.input(Reed\_DO\_PIN)

if(digitalVal == 1):

print 'DO is %d' % digitalVal

analogVal = ADC0832.getResult(0)

print 'Current analog value is %d'% analogVal

if(analogVal < thresholdVal):

GPIO.output(LedPin, GPIO.HIGH)

time.sleep(0.2)

else:

GPIO.output(LedPin, GPIO.LOW)

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

init()

try:

loop()

except KeyboardInterrupt:

ADC0832.destroy()

print 'The end !'

2. C Code

#include <wiringPi.h>

#include <stdio.h>

#include <string.h>

#include <errno.h>

#include <stdlib.h>

#define ADC\_CS 0

#define ADC\_CLK 1

#define ADC\_DIO 2

#define Reed\_DO\_Pin 3

#define LedPin 4

#define thresholdVal 100

typedef unsigned char uchar;

typedef unsigned int uint;

uchar get\_ADC\_Result(void)

{

uchar i;

uchar dat1=0, dat2=0;

digitalWrite(ADC\_CS, 0);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,0); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

for(i=0;i<8;i++)

{

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0); delayMicroseconds(2);

pinMode(ADC\_DIO, INPUT);

dat1=dat1<<1 | digitalRead(ADC\_DIO);

}

for(i=0;i<8;i++)

{

dat2 = dat2 | ((uchar)(digitalRead(ADC\_DIO))<<i);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0); delayMicroseconds(2);

}

digitalWrite(ADC\_CS,1);

pinMode(ADC\_DIO, OUTPUT);

return(dat1==dat2) ? dat1 : 0;

}

int main(void)

{

uchar digitalVal = 1;

uchar analogVal = 0;

if(wiringPiSetup() == -1){ //when initialize wiring failed,print messageto screen

printf("setup wiringPi failed !");

return 1;

}

pinMode(ADC\_CS, OUTPUT);

pinMode(ADC\_CLK, OUTPUT);

pinMode(Reed\_DO\_Pin, INPUT);

pullUpDnControl(Reed\_DO\_Pin, PUD\_UP);

pinMode(LedPin, OUTPUT);

while(1){

if((digitalVal = digitalRead(Reed\_DO\_Pin)))

{

printf("Do is %d.\n", digitalVal);

analogVal = get\_ADC\_Result();

printf("Current analog value is %d.\n", analogVal);

if(analogVal < thresholdVal)

{

digitalWrite(LedPin, HIGH);

}

delay(200);

}

else

{

digitalWrite(LedPin, LOW);

}

}

return 0;

}

Experimental phenomena

When a magnet approaches the reed switch, the LED lights up, and when the magnet is away from the reed switch, the LED goes out. If you find that the LED does not respond when the magnet come close, please try turning the resistor on the module to adjust the threshold value.