Linear Hall sensor

Description

This course uses the Raspberry Pi to capture the signal of the linear Hall sensor, and control Led on and off based on captured signal.

Experimental Materials

RaspberryPi \*1

Breadboard \*1

Linear hall sensor \*1

ADC0832 \*1

Led \*1

Dupont Line

Ready to 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

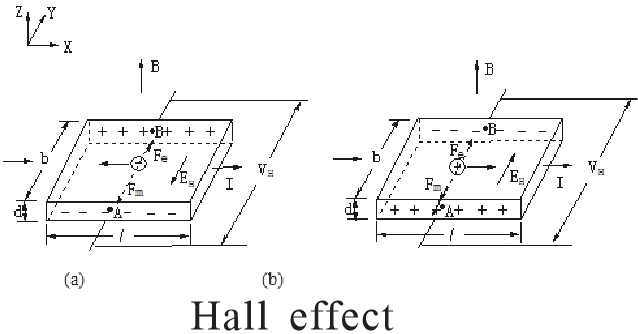
See the attached "Installing a Python Interpreter and Corresponding Libraries in a Raspberry Pi System" for details.

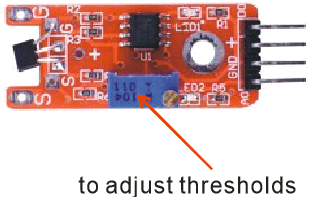
Product description

I. Introduction:

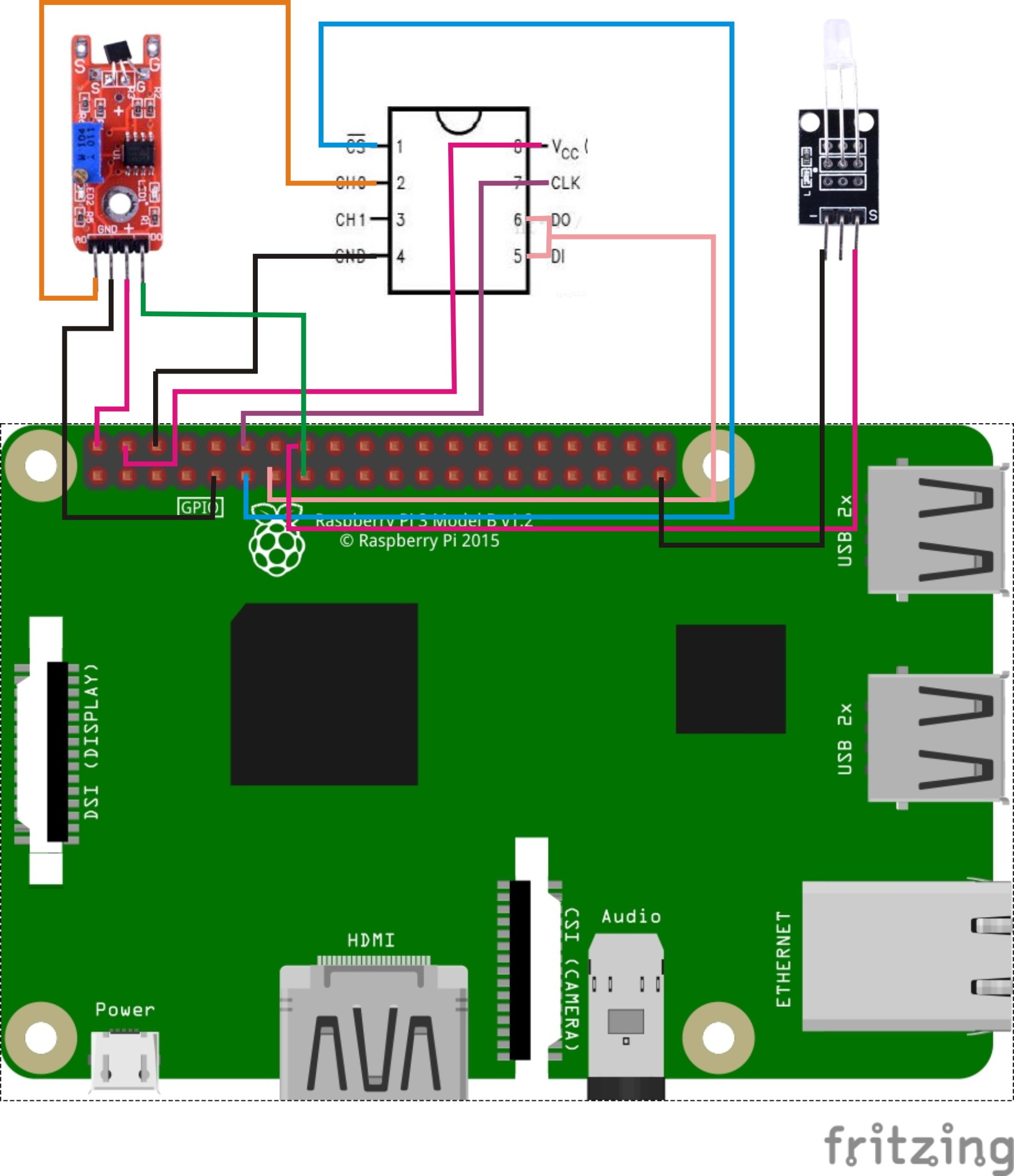
Hall Effect: When there is current at both ends of the semiconductor sheet and a uniform magnetic field with a magnetic induction strength B is applied in the vertical direction of the sheet, a Hall with a potential difference of UH will be generated in the direction perpendicular to the current and the magnetic field. Voltage.According to the Hall effect, a component made of a semiconductor material is called a Hall element. It has the advantages of being sensitive to

magnetic field, simple structure, small volume, wide frequency response, large output voltage change, long service life, etc. Therefore, it has been widely used in measurement, automation, computer and information technology.A voltage difference is generated when the Hall element and the magnet meet in the forward direction, and there is no voltage difference when the Hall element and the magnet meet in the forward direction, so that you can obtain the voltage change by the Raspberry Pi,and then judge the proximity of the magnet, and you can also control the LED light on and off according to the signal.





Wiring diagram



Sample code

1. Python code

#!/usr/bin/env python

import RPi.GPIO as GPIO

import ADC0832

import time

Hall\_DO\_PIN = 15

LedPin = 16

thresholdVal = 100

def init():

GPIO.setmode(GPIO.BOARD)

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

GPIO.setup(LedPin, GPIO.OUT)

ADC0832.setup()

def loop():

while True:

global digitalVal

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

if(digitalVal == 0):

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 !'

1. C code
2. #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 Hall\_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)

{

printf("setup wiringPi failed !\n");

return -1;

}

pinMode(ADC\_CS, OUTPUT);

pinMode(ADC\_CLK, OUTPUT);

pinMode(Hall\_DO\_Pin, INPUT);

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

pinMode(LedPin, OUTPUT);

while(1)

{

if((digitalVal = digitalRead(Hall\_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 the magnet is vertically close to the linear Hall sensor,there will be a voltage difference generated in the linear Hall sensor. When the analog value of the linear Hall sensor AO pin satisfies the level-reversal threshold condition, the level of the DO pin is inverted. The Raspberry Pi will read the ADC converted value of the analog Hall sensor and print it to the command line interface of the Raspberry Pi system,and when this value meet the set condition ,Led will on.