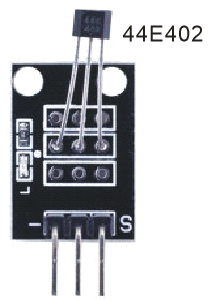
Hall Magnetic Sensor  


Overview

Hall effects are magnetic sensors, and detect changes in the magnetic field. Where linear, analog Hall sensors vary their voltage output in relation to the strength or weakness of the field, a magnetic switch sensor such as 44E402 simply signals the presence or absence of a nearby magnet, making them ideal for use with a simple bar or rod magnet.

<<TBW. Nick still needs an understanding of how this device differs from ANALOG HALL if at all. See https://github.com/Alion3064492356/Sensors-for-RaspberryPi/issues/10 >>

Materials Needed

Raspberry Pi x1

Breadboard x1

Hall magnetic sensor x1

ADC0832 x1

LED (3 pin) x1

Dupont jumper wires

Any magnet (you provide)

Experimental Procedure

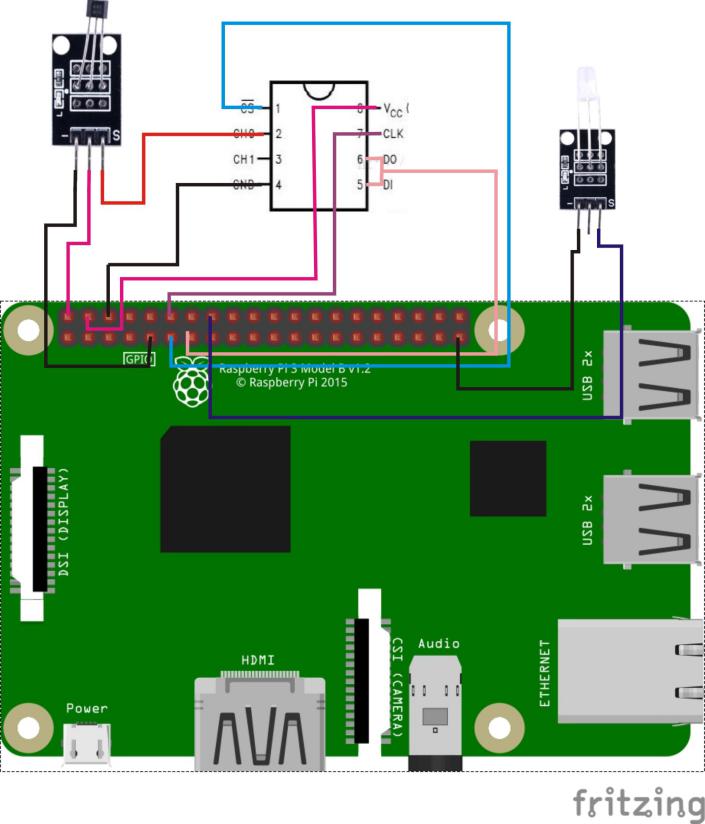
1. If you have not done so already, prepare your development system by installing the Python interpreter, RPi.GIO library, and wiringPi library as described in READ\_ME.TXT.
2. Install the ADC0832 analog/digital converter IC, Hall magnetic sensor, and three-pin LED on your breadboard, and use Dupont jumper wires to connect them to each other and your Raspberry Pi as illustrated in the Wiring Diagram below. (The three-pin LED provided in this kit includes onboard series resistors, so no additional resistors are needed.)
3. Execute the sample stored in this experiment’s subfolder.

If using C, compile and execute the C code:  
cd Code/C  
gcc hallMagnetic.c -o hallMagnetic.out –lwiringPi  
./ hallMagnetic.out

If using Python, launch the Python script:  
cd Code/Python  
python hallMagnetic.py

1. Make experimental observations.  
   When you hold your magnet vertically close to the sensor, the Hall effect generates an (analog) voltage, which the ADC converts to a (digital) signal readable by the Raspberry Pi. The sample code then turns on the LED if that voltage exceeds a certain threshold.

Wiring Diagram



AD0382 Pin position:

CS ↔ Raspberry Pi Pin 11

CLK ↔ Raspberry Pi Pin 12

DI ↔ Raspberry Pi Pin 13

D0 ↔ Raspberry Pi Pin 13

CH0 ↔ Hall Magnetic Sensor Pin "S"

VCC ↔ Raspberry Pi +5V

GND ↔ Raspberry Pi GND

Hall magnetic pin position:

"S" ↔ ADC0382 CH0

"+" ↔ Raspberry Pi +5V

"-" ↔ Raspberry Pi GND

LED pin position:

"S" ↔ Raspberry Pi Pin 16

"-" ↔ Raspberry Pi GND

Sample code

Python Code

#!/usr/bin/env python

import ADC0832

import time

import RPi.GPIO as GPIO

LedPin = 16

thresholdVal = 150

def init():

ADC0832.setup()

GPIO.setup(LedPin, GPIO.OUT)

def loop():

while True:

analogVal = ADC0832.getResult(0)

print 'analog value is %d' % analogVal

if(analogVal > thresholdVal):

GPIO.output(LedPin, GPIO.HIGH)

else:

GPIO.output(LedPin, GPIO.LOW)

time.sleep(0.2)

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

init()

try:

loop()

except KeyboardInterrupt:

ADC0832.destroy()

print 'The end !'

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 LedPin 4

#define thresholdVal 150

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 analogVal;

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return 1;

}

pinMode(ADC\_CS, OUTPUT);

pinMode(ADC\_CLK, OUTPUT);

pinMode(LedPin, OUTPUT);

while(1)

{

analogVal = get\_ADC\_Result();

printf("Current analog : %d\n", analogVal);

if(analogVal > thresholdVal)

{

digitalWrite(LedPin, HIGH);

}

else

{

digitalWrite(LedPin, LOW);

}

delay(200);

}

return 0;

}