Smoke Sensor  


Introduction

We will use the Raspberry Pi to capture the smoke sensor signal to control the LED light on and off based on this signal.

Material Needed

Raspberry Pi x1

Breadboard x1

Smoke sensor x1

ADC0832 x1

LED (3-pin) x1

Resistor (330Ω) x1

Dupont jumper wires

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, smoke sensor, three-pin LED and resistor 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. Note you will connect only two of the three pins on the LED.
3. Execute the sample stored in this experiment’s subfolder.

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

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

1. Make experimental observations.

Product Description

Brief Introduction

The gas-sensitive material used in the MQ-2 gas sensor is tin dioxide (SnO2) with low conductivity in clean air. When the combustible gas is present in the environment where the sensor is located, the conductivity of the sensor increases as the concentration of combustible gas in the air increases.

A simple circuit can be used to convert the change in conductivity to an output signal corresponding to the gas concentration. The MQ-2 gas sensor is highly sensitive to liquefied gas, propane, and hydrogen and is ideal for the detection of natural gas and other combustible vapors.

This sensor can detect a variety of flammable gases, and can be used for gas leakage monitoring devices at homes and factories, is suitable for the detection of liquefied gas, butane, propane, methane, smoke, etc. It is a low-cost sensor suitable for a variety of applications.

Characteristic Parameters

◆ Product Model: MQ-2

◆ Product Type: Semiconductor Sensors

◆ Detection gas: combustible gas

◆ Detection concentration: 300ppm ~ 10000ppm

◆ Input voltage: DC5V

◆ Analog output voltage increases with higher gas concentration.

◆ Good sensitivity to liquefied gas, natural gas, city gas, smoke.

◆ There are four screw holes for easy positioning;

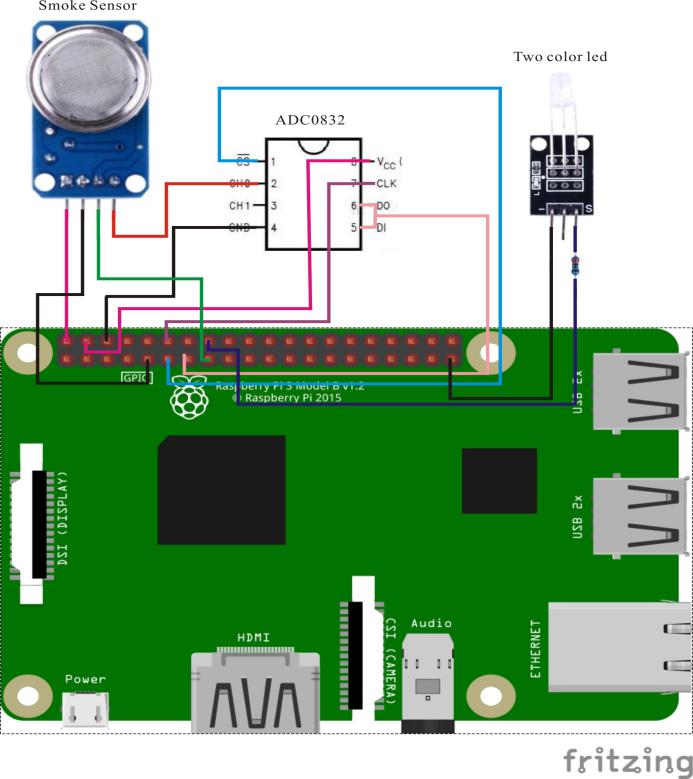
◆ Product Dimensions: 32(L)\*20(W)\*22(H)

◆ has a long service life and reliable stability

◆ Fast response recovery characteristics

◆ After the sensor is powered on, it needs to be warmed up for about 20 seconds,then the measured data is stable.It is a normal phenomenon that the sensor becomes a little hot ,but if you touch by finger and you feel very hot,it's unnormal.

Wiring Diagram



ADC0382 pin position:

CS ↔ Raspberry Pi Pin 11

CLK ↔ Raspberry Pi Pin 12

DI ↔ Raspberry Pi Pin 13

D0 ↔ Raspberry Pi Pin 13

CH0 ↔ Flame Sensor Pin A0

VCC ↔ Raspberry Pi +5V

GND ↔ Raspberry Pi GND

Flame Sensor pin position:

A0 ↔ ADC0382 Pin CH0

D0 ↔ Raspberry Pi Pin 15

GND ↔ Raspberry Pi GND

"+" ↔ Raspberry Pi +5V

LED pin position:

"S" ↔ Raspberry Pi Pin 16(through resistor)

"-" ↔ Raspberry Pi GND

Sample Code

Python code

#!/usr/bin/env python

#

# This is a program for MQ-2 Gas Sensor Module.

# It could detect danger gas and smokes.

# This program depends on ADC0832 ADC chip. Follow

# the instruction book to connect the module and

# ADC0832 to your Raspberry Pi.

#

import RPi.GPIO as GPIO

import ADC0832

import time

LedPin = 16 # Set buzzer pin

threshold = 100 # You can set the Threshold by yourself (0-255)

def setup():

GPIO.setmode(GPIO.BOARD) # Numbers GPIOs by physical location

GPIO.setup(LedPin, GPIO.OUT) # Set pins' mode is output

ADC0832.setup() # Setup ADC0832

def loop():

while True:

tmp = ADC0832.getResult(0) # Get analog value from ADC0832

print tmp # Print analog value

if tmp > threshold : # LedPin when read value greater than threshold

print ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

print ' \* ! DANGER ! \*'

print ' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

print ''

GPIO.output(LedPin, GPIO.HIGH) # (0, means detect danger gas)

time.sleep(0.25)

GPIO.output(LedPin, GPIO.LOW)

time.sleep(0.25)

else :

time.sleep(0.5) # Else delay printing.

def destroy():

GPIO.cleanup() # Release resource

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()

C Code

#include <wiringPi.h>

#include <stdio.h>

typedef unsigned char uchar;

typedef unsigned int uint;

#define ADC\_CS 0

#define ADC\_CLK 1

#define ADC\_DIO 2

#define LedPin 4

#define Threshold 100

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 tmp;

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return -1;

}

pinMode(ADC\_CS, OUTPUT);

pinMode(ADC\_CLK, OUTPUT);

pinMode(LedPin, OUTPUT);

while(1)

{

pinMode(ADC\_DIO, OUTPUT);

tmp = get\_ADC\_Result();

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

if(tmp > Threshold)

{

digitalWrite(LedPin, LOW);

delay(1000);

digitalWrite(LedPin, HIGH);

printf("\n\*\*\*\*\*\*\*\*\*\*\* \

\n Danger! \

\n\*\*\*\*\*\*\*\*\*\*\* \

\n");

delay(1000);

}

else

{

digitalWrite(LedPin, LOW);

}

delay(100);

}

return 0;

}

Experimental phenomena

When the smoke sensor detects combustible gas, and the gas concentration meets a set threshold value, the LED lamp will light up, and when the gas concentration does not meet the set threshold value, the LED lamp will turn off.