Sound sensor

Overview

We will use the Raspberry Pi to control the sound sensor module,capture the sound sensor signal, and print it to the command line of Raspberry Pi.

Materials Needed

RaspberryPi \*1

Breadboard \*1

Small sound sensor \*1

ADC0832 \*1

Dupont Line

Preparation

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

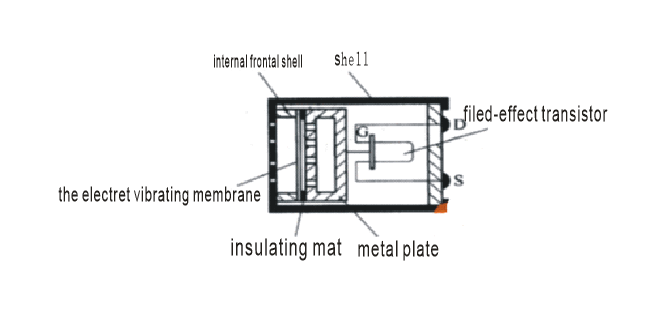
See the attached <<Installing a Python Interpreter and Corresponding Libraries in a Raspberry Pi System>> for details.

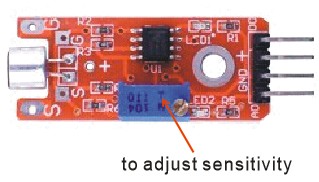
Description

Brief Introduction

The sound sensor is a kind of sensor that can detect sound, and is widely used in daily life, military, medical, industrial, territorial waters, aerospace, etc., and has become an indispensable part of the development of modern society.

The sound sensor incorporates a sound-sensitive capacitive electret microphone. Acoustic waves oscillate the electret film in the microphone, causing a change in capacitance and producing a tiny voltage that changes accordingly. This voltage is then converted to a voltage of 0-5V, accepted by the data collector after A/D conversion, and sent to the controller. In this lesson we use the Raspberry Pi to detect the sound sensor module's signal and make the appropriate action based on this signal.





Features and Spec

◆There are 3.6mm mounting screw holes

◆ Use 5v DC power supply

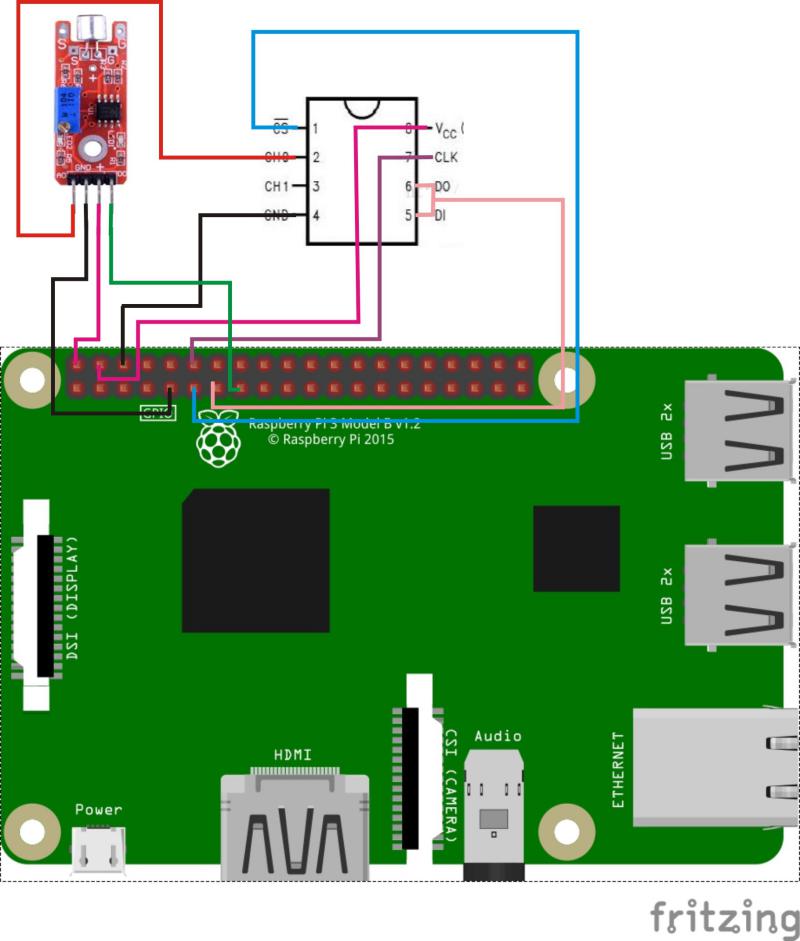
◆ AO analog output, real-time output sound voltage signal

◆ There is a D0 threshold level output, when the sound intensity reaches a certain threshold, the output high and low level signal, the threshold - sensitivity can be adjusted by the potentiometer

◆ Power indicator

◆ Comparator output has indicator light

Cable Connection



Sample Code

Python Code

#!/usr/bin/env python

import RPi.GPIO as GPIO

import ADC0832

import time

Sound\_DO\_PIN = 15

def init():

GPIO.setmode(GPIO.BOARD)

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

ADC0832.setup()

def loop():

while True:

global digitalVal

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

if(digitalVal == 0):

print 'DO is %d' % digitalVal

print "voice in..."

print 'Current analog value is %d'% ADC0832.getResult(0)

else:

pass

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 Sound\_DO\_Pin 3

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(Sound\_DO\_PIN, INPUT);

pullUpDnControl(Sound\_DO\_PIN, PUD\_UP);

printf("Please speaking...\n");

while(1)

{

if(!(digitalVal = digitalRead(Sound\_DO\_PIN)))

{

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

printf("Voice in...");

printf("Current analog value is %d.\n", get\_ADC\_Result());

}

else

{

;

}

}

return 0;

}

Experimental phenomena

The sound sensor detects the sound. When the sound intensity reaches the set threshold, the current digital value after the ADC conversion is printed in the command line interface of the Raspberry Pi system.