**Build Instructions**

**Stereo Sound Level Detector**

1. Bill of Materials/Budget

This project requires a Raspberry Pi version 2 or higher, and 2 sound sensor sensors from sunfounder. Additionally a long cable and usb keyboard/mouse may be needed if these are not available.

Raspberry Pi 2: 100 CAD

2 Sunfounder Sound Sensor Modules: 18 CAD

PCF8591: 12 CAD

Total: 130 CAD + Tax ~150 CAD

Additional parts (Cables, LED, USB keyboard, Mouse up to 30 CAD).

1. Time commitment

The time needed for the various activities building this device with instructions varies.

No more than 2 hours of soldering

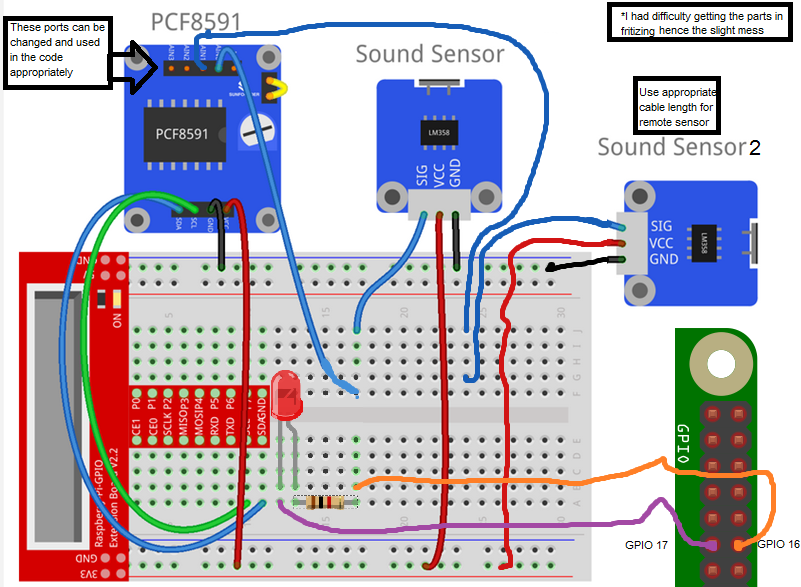
No more than 2 hours of programming/calibration

No more than 2 hours of setting up Raspberry Pi

~ 30 minutes of connecting parts

1. Mechanical Assembly/Schematic

See below for assembly of the system….



1. PCB/Soldering

The PCB used was a stripped down version of the PCB used for the hardware technology class, link attached <https://github.com/six0four/StudentSenseHat/blob/master/README.md>

The main difference is that the connection to the PCF is now direct, changing the voltage supply from 5V to 3.5V that is better suited for the project. Resistors R4-R7 are removed, and instead direct paths are provided (Through soldering cables or re-design). Q1 and Q2 components are also unnecessary.

After it’s done, the user can just mount it on top of the Raspberry Pi

1. Power UP/Running software

The Raspberry Pi needs to be initialized with a Ubuntu system, SSH and I2C protocols need to be enabled in the Raspberry Pi configuration.

The program to run the device is written below the build instructions, as well as attached as a file, run with **sudo ./NoiseDetector** or the name of the file. Compile with **gcc –Wall –o nameofprogramtorun NoiseDetector.c –lwiringpi -lm**

1. Unit Testing

Each of the sound sensors should be tested and calibrated appropriately, they should be tested and confirmed to have close to equal sound sensitivity, if not, slots or hardware may need to be replaced. Changing cables and the PCF input slots for the sound sensors can provide a different result.

The PCB also should be tested appropriately, as well as the LED on it.

1. Summary

By following these instructions you can reproduce my device fairly accurately, the key differences could be in hardware, because the units in the hardware come in different variations to a certain amount, sensitivity of the sound sensors may be different to mine, this calibration needs to be done individually in the code (There are a few keys set up in the code that can easily be changed for different results) and in the hardware.

Code:

#include <stdio.h>

#include <stdlib.h>

#include <wiringPi.h>

#include <pcf8591.h>

#include <math.h>

#define PCF       120

int main (void)

{

int value1, value2;

int data1, data2;

int counter = 0;

int step = 1; //1

int i = 0;

int offset = 85;  //85

int deviation = 10; //10

int noiseCounter = 0;

int noiseDuration = 5; //20

int noiseOffCounter = 0;

int noiseOffDuration = 1000; //1000

int noise = 0;

if (wiringPiSetup () == -1) {

printf("Error at wiringPiSetup()");

    return 1 ;

}

pinMode (0, OUTPUT) ;         // aka BCM\_GPIO pin 17

pinMode (1, OUTPUT) ; // aka BCM\_GPIO pin 18

// Setup pcf8591 on base pin 120, and address 0x48

// printf("%d\n", ++step);

if (pcf8591Setup (PCF, 0x48) == -1) {

printf("Error at pcf8591Setup()");

    return 1 ;

}

       printf("Lowest noise level \t\t\t%d\n", offset);

       printf("Data has been generalized with weight value of %d\n", step);

while(1) // loop forever

{

data1 += analogRead  (PCF + 0); //white - local

data2 += analogRead  (PCF + 1); //yellow - remote

counter++;

if(noise) {

noiseOffCounter++;

}

if(counter == step ) {

counter = 0;

value1 = round((double)data1 / step);

value2 = round((double)data2 / step);

data1 = 0;

data2 = 0;

if ((value1 < offset) || (value2 < offset)) {

if( abs (value1 - value2) < deviation)  {

if (++noiseCounter > noiseDuration) {

noiseCounter = 0;

noiseOffCounter = 0;

noise = 1;

}

}

else  {

if(noise) {

if(noiseOffCounter > noiseOffDuration) {

noiseOffCounter = 0;

noise = 0;

}

}

else if( --noiseCounter < 0)  {

noiseCounter = 0;

}

}

if(noise) {

printf("\n\tNOISE !!!!  #1:%3d #2:%3d", value1, value2);

//both microphones are sensing noise - red

digitalWrite (0, LOW);

digitalWrite (1, HIGH);

}

else {

printf("\n\tVoice In!!  #1:%3d #2:%3d", value1, value2);

// Voice in at any of microphones - green

digitalWrite (0, HIGH);

digitalWrite (1, LOW);

}

for(i = offset; (i > offset - value1) && (i > 0); i--) {

printf(" ");

}

for(i = offset; i > value1; i--) {

printf("-");

}

printf("|");

for(i = offset; i > value2; i--) {

printf("+");

}

}

else  {

if(!noise) {

digitalWrite (0, LOW);

digitalWrite (1, LOW);

}

}

}

}

return 0;

}