

# Sound Sensor

来自Waveshare Wiki

跳转至: [导航](#)、[搜索](#)



## 功能简介

声音传感器

AD

I/Os

## 说明

### 产品特性

参数名称	参数
音频放大芯片	LM386(放大200倍)
工作电压	3.3V-5.5V
产品尺寸	39.0mm*21.0mm
固定孔尺寸	2.0mm

### 工作原理

LM386是一种音频集成功率放大器，具有自身功耗低、更新内链增益可调整、电源电压范围大、外接元件少和总谐波失真小等优点。主要应用于低电压消费类产品。为使外围元件最少,电压增益内置为20。在1脚和8脚之间增加一只外接电阻和电容，便可将电压增益调为200以内的任意值。

### 主要用途

检测周围环境声音的有无和判断声音强度的大小

## 接口说明

### 接口说明

引脚号	标识	描述
1	DOUT	数字量输出
2	AOUT	模拟量输出
3	GND	电源地
4	VCC	电源正(3.3V-5.3V)

## Pico使用教程

提供Python、C例程

### Pico 快速上手

- 快速学习

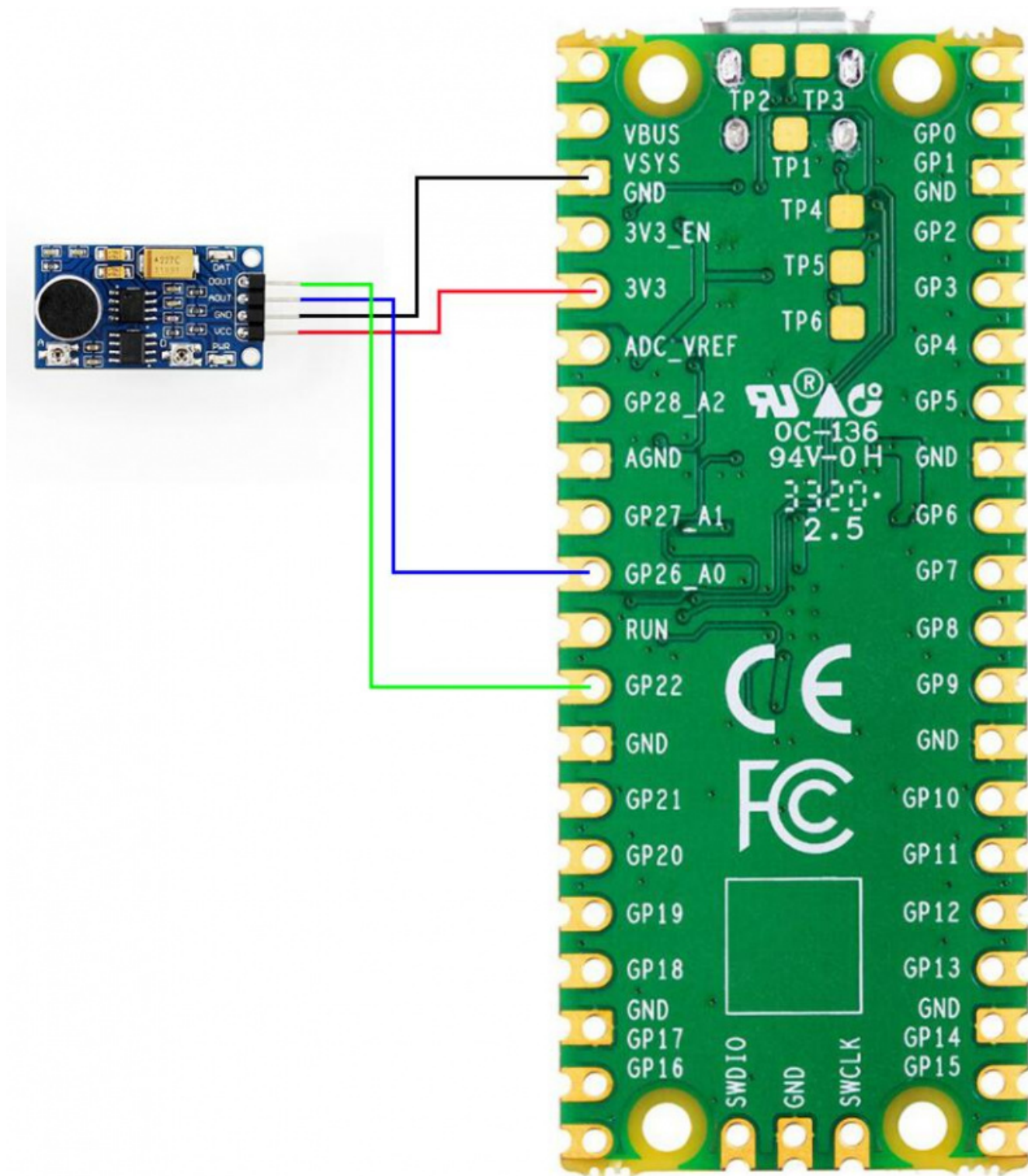
### 硬件连接

您可以对照以下表格连线。

Pico连接引脚对应关系

Sound	Pico	功能
VCC	3.3V	电源输入
GND	GND	电源地
AOUT	GP26	模拟量输出
DOUT	GP22	数字量输出

火焰传感器、霍尔传感器、红外接近传感器、液位传感器、土壤传感器、气体传感器、声音传感器、紫外线传感器这些模块的引出脚一致(根据表格进行连接, 图仅为参考):



## 基础介绍

MicroPython系列 [展开](#)

C/C++系列 [展开](#)

## 程序下载

打开树莓派终端，执行以下命令：

```
sudo apt-get install p7zip-full
cd ~
sudo wget https://www.waveshare.net/w/upload/b/bc/Sound-Sensor-code.7z
7z x Sound-Sensor-code.7z -o./Sound-Sensor-code
cd ~/Sound-Sensor-code
cd Pico/c/build/
```

## 例程使用

### C部分

- 以下教程为在树莓派上操作，但由于cmake的多平台、可移植的特点，在PC上也是能成功编译，但操作略有不同，需用户自行判断。

进行编译，请确保在c目录：

```
cd ~/Sound-Sensor-code/Pico/c/
```

在文件夹中创建并进入build目录,并添加SDK: 其中 ../../pico-sdk 是你的SDK的目录。我们示例程序中是有build的，直接进入即可

```
cd build
export PICO_SDK_PATH=../../pico-sdk
(注意：务必写对你自己的SDK所在路径)
```

执行cmake自动生成Makefile文件

```
cmake ..
```

执行make生成可执行文件，第一次编译时间比较久

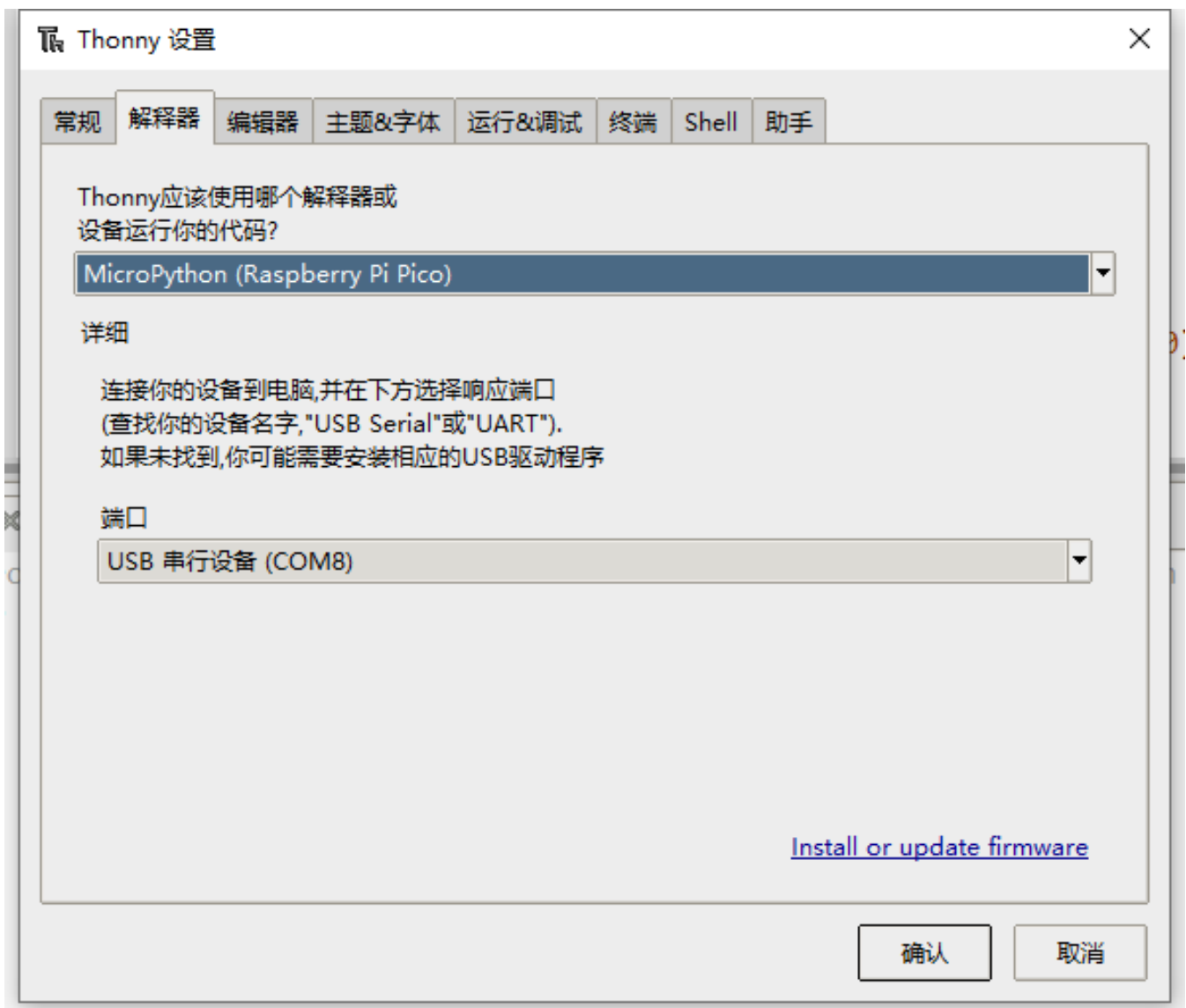
```
make -j9
```

编译完成，会生成uf2文件。按住Pico板上的按键，将pico通过Micro USB线接到树莓派的USB接口，然后松开按键。接入之后，树莓派会自动识别到一个可移动盘（RPI-RP2），将build文件夹下的main.uf2 文件复制到识别的可移动盘（RPI-RP2）中即可。

```
cp main.uf2 /media/pi/RPI-RP2/
```

### windows环境下的使用

- 1. 按住Pico板上的BOOTSET按键，将pico通过Micro USB线接到电脑的USB接口，待电脑识别出一个可移动硬盘（RPI-RP2）后，松开按键。
- 2. 将python目录中rp2-pico-20210418-v1.15.uf2 文件复制到识别的可移动盘（RPI-RP2）中
- 3. 打开Thonny IDE（注意：要使用最新版本的Thonny，否则是没有Pico的支持包的，当前Windows下的最新版本为v3.3.3）
- 4. 点击工具->设置->解释器，如图所示选择Pico及对应的端口



本例程提供了一个程序：

- 5. 文件->打开->Sound\_Sensor.py,点击运行即可，如下图所示：

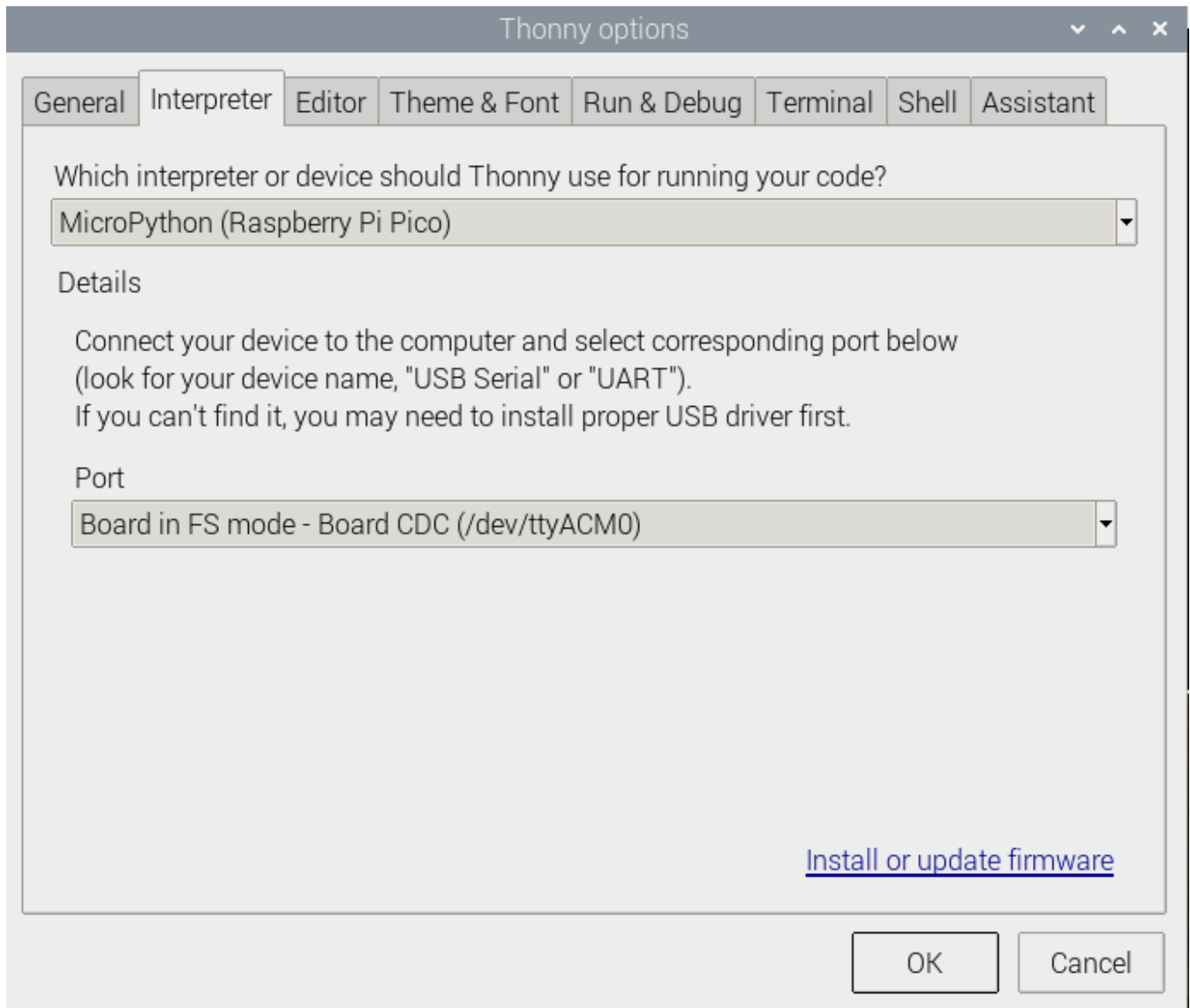


## 树莓派环境下的使用

- 1.刷固件的过程与Windows上一样，你可以选择在PC或者树莓派上将rp2-pico-20210418-v1.15.uf2 文件拷入pico中。
- 2.在树莓派上打开Thonny IDE（点击树莓logo -> Programming -> Thonny Python IDE），你可以在Help->About Thonny查看版本信息

以确保你的版本是有Pico支持包的，同样你可以点击Tools -> Options... -> Interpreter选择MicroPython(Raspberry Pi Pico 和ttyACM0端口

如图所示：



如果你当前的Thonny版本没有pico支持包，输入以下指令来更新Thonny IDE

```
sudo apt upgrade thonny
```

3.点击File->Open...->python/Sound\_Sensor.py，运行脚本即可

## 现象

- 1、当模块的咪头靠近发声源时，模块上的信号指示灯点亮，当模块的咪头远离发声源时，模块上的信号指示灯熄灭。
- 2、随着传感器与发声源距离的变化，串口输出数据有相应的变化。

## STM32使用教程

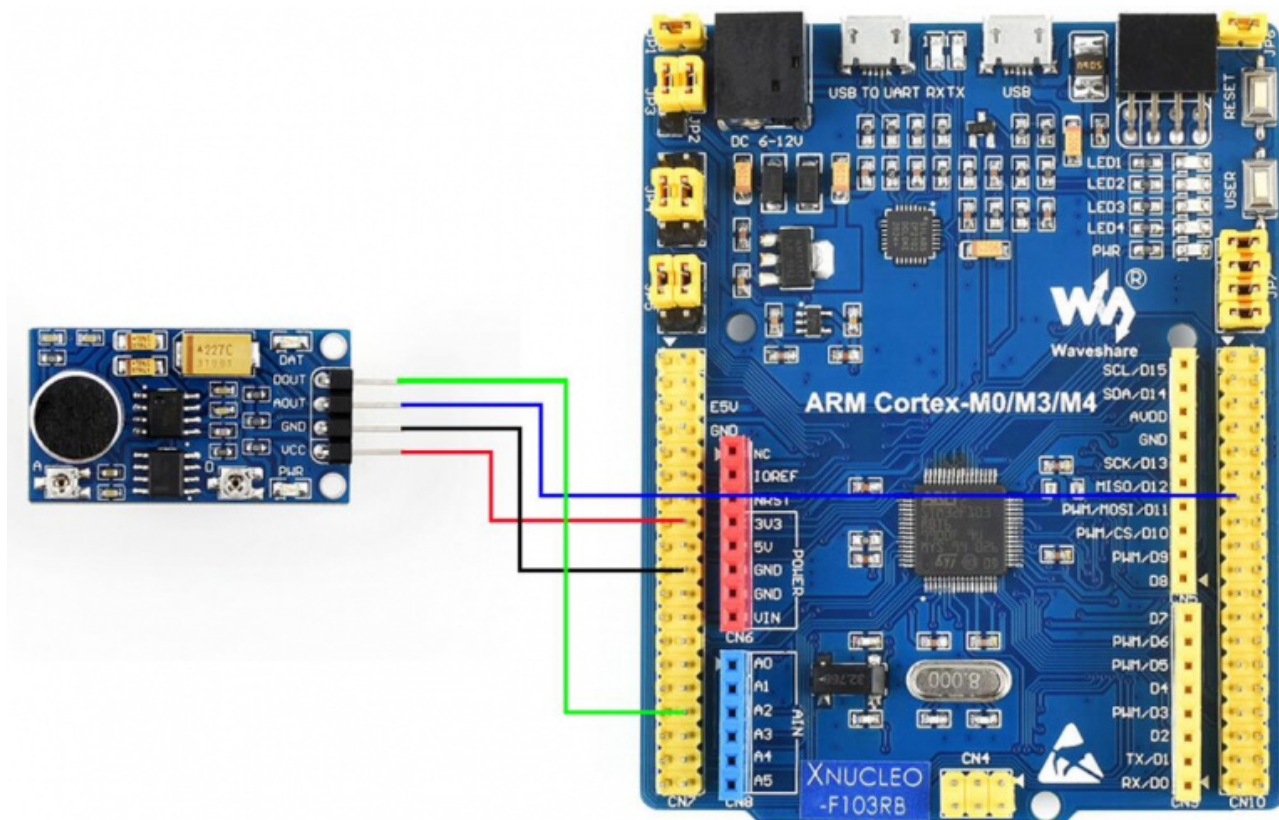
我们提供的例程是基于STM32F103RBT6跟STM32H743的，提供的连接方式是对应的STM32F103RBT6的引脚为例，如果有需要使用其他STM32，请按实际引脚连接

### 硬件连接

STM32F103RB连接引脚对应关系

Sound	STM32	功能
VCC	3.3V	电源输入
GND	GND	电源地
AOUT	PA6	模拟量输出
DOUT	PA4	数字量输出

火焰传感器、霍尔传感器、红外接近传感器、液位传感器、土壤传感器、MQ5气体传感器、声音传感器、紫外线传感器这些模块的引出脚一致(根据表格进行连接，图仅为参考):

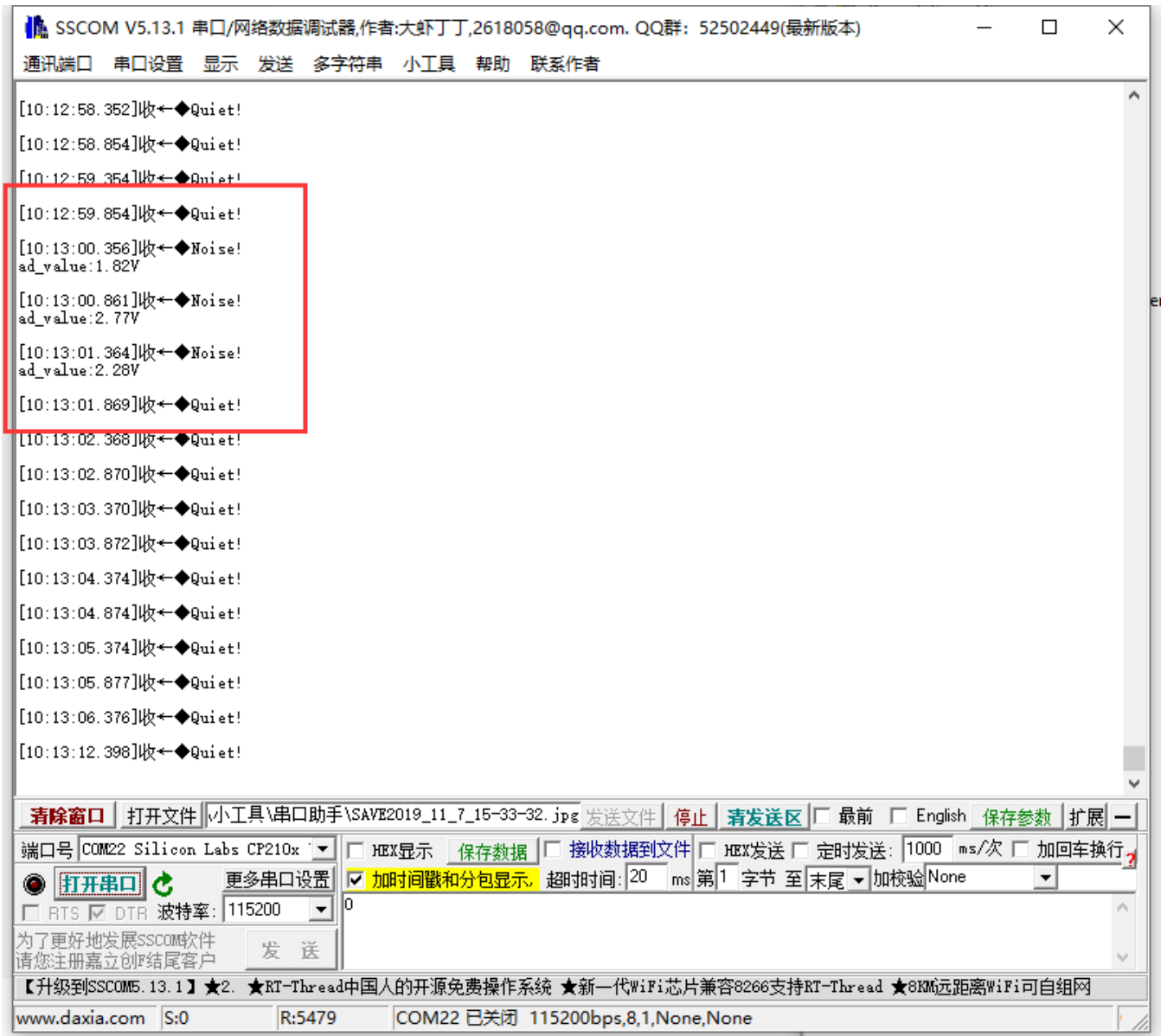


### 软件说明





下载成功后，运行SSCOM连接MQ5模块，就可以实时查看传感器的状态啦。



## 现象

- 1、当模块的咪头靠近发声源时，模块上的信号指示灯点亮，当模块的咪头远离发声源时，模块上的信号指示灯熄灭。
- 2、随着传感器与发声源距离的变化，串口输出数据有相应的变化。

# Arduino使用教程

本例程已经在Arduino uno上测试通过，直接按下表连接Arduino uno即可

## 硬件连接

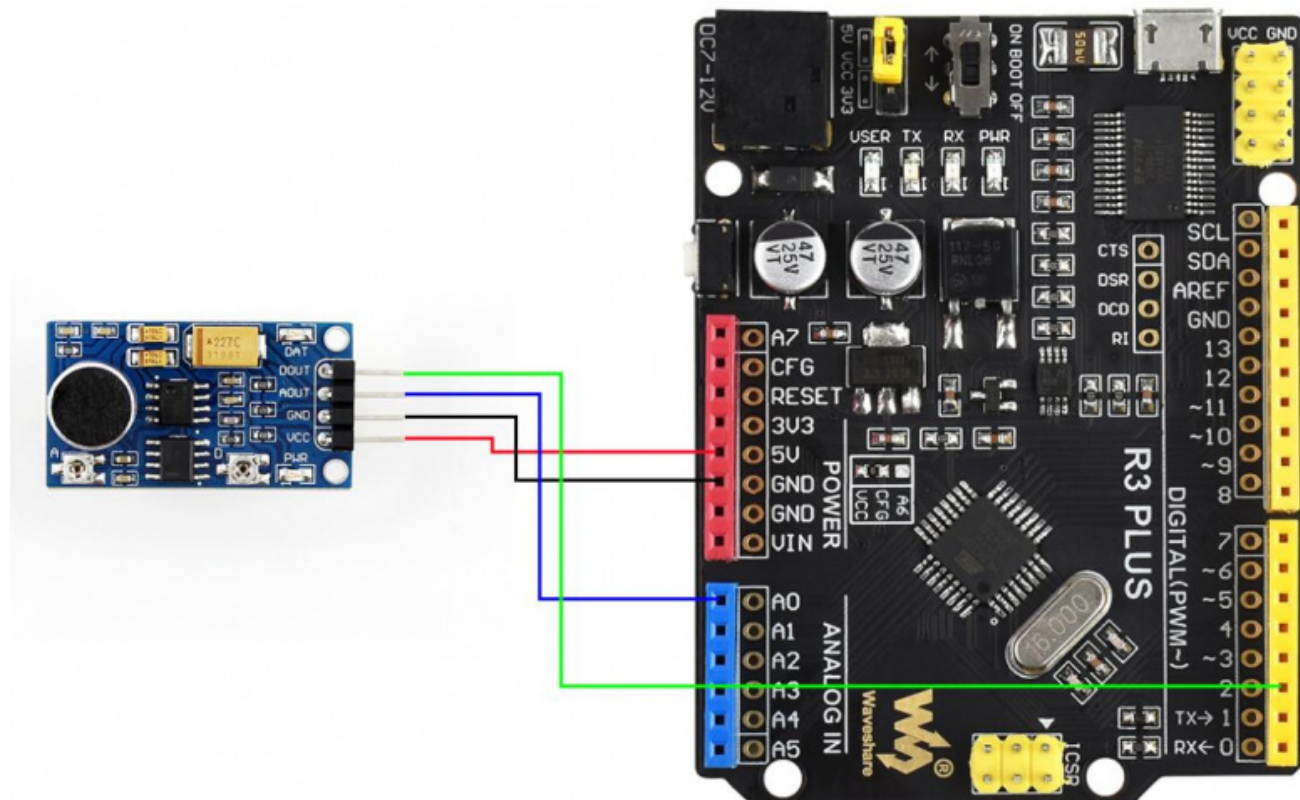
您可以对照以下表格连线。

Arduino连接引脚对应关系

Sound	Arduino	功能
-------	---------	----

VCC	5V	电源输入
GND	GND	电源地
AOUT	A0	模拟量输出
DOUT	D2	数字量输出

火焰传感器、霍尔传感器、红外接近传感器、液位传感器、土壤传感器、MQ5气体传感器、声音传感器、紫外线传感器这些模块的引出脚一致(根据表格进行连接, 图仅为参考):



## 安装编译软件 (Windows教程)

arduino IDE 安装教程

## 运行程序

在我们提供的资料里面下载程序, 并解压, 再进入Sound-Sensor-code/Arduino/Sound\_Sensor目录下

双击打开Sound\_Sensor.ino文件。

选择你的开发板, 跟对应的端口。

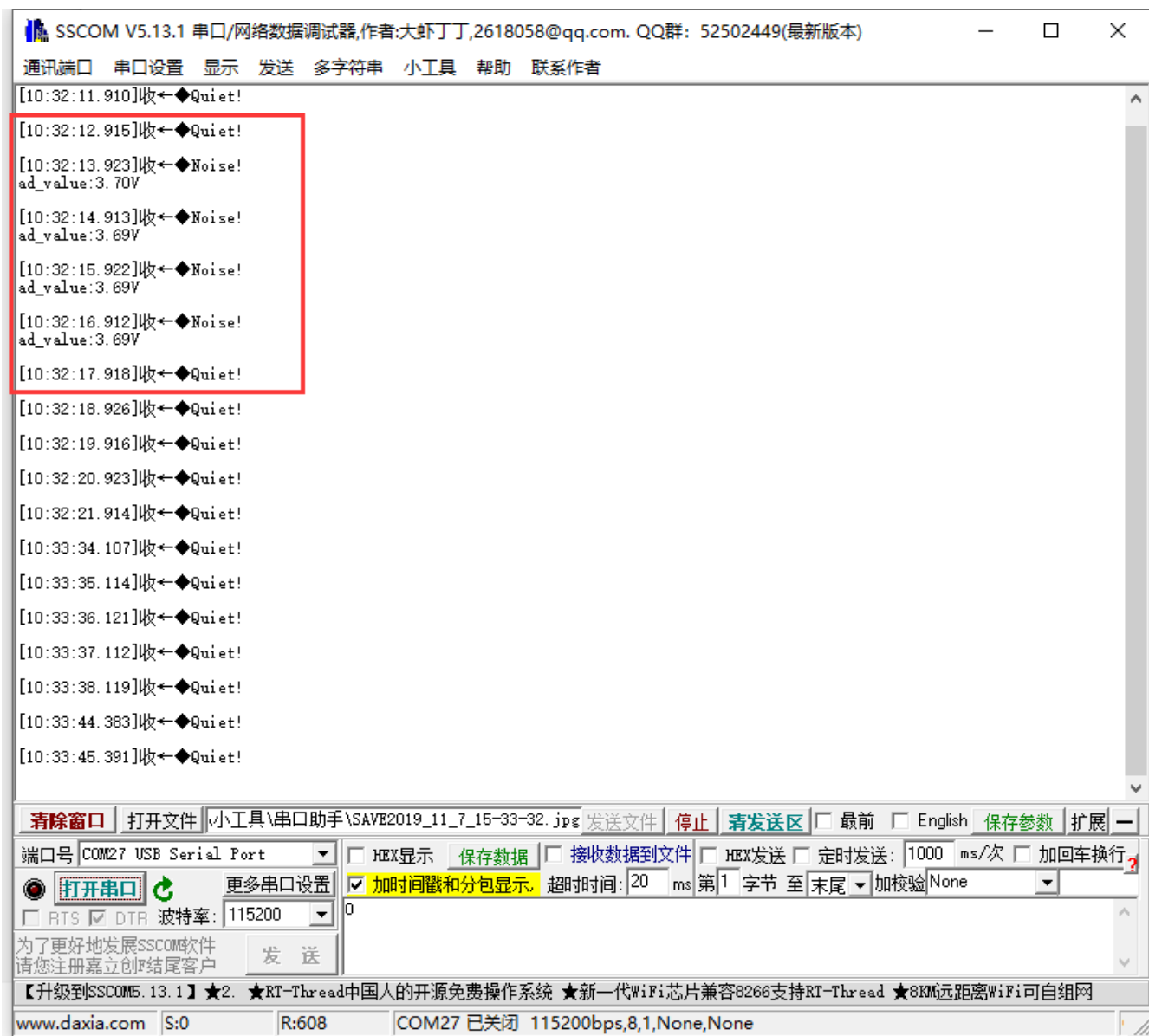


进行编译下载，如下图：

文件 编辑 项目 工具 帮助



下载成功后，运行SSCOM连接MQ5模块，就可以实时查看传感器的状态啦。



## 现象

- 1、当模块的咪头靠近发声源时，模块上的信号指示灯点亮，当模块的咪头远离发声源时，模块上的信号指示灯熄灭。
- 2、随着传感器与发声源距离的变化，串口输出数据有相应的变化。

## 资料

### 文档

- 用户手册
- 原理图

## 数据手册

---

- Im393

## 程序

---

- 示例程序

## 软件

---

- 串口调试助手

## FAQ

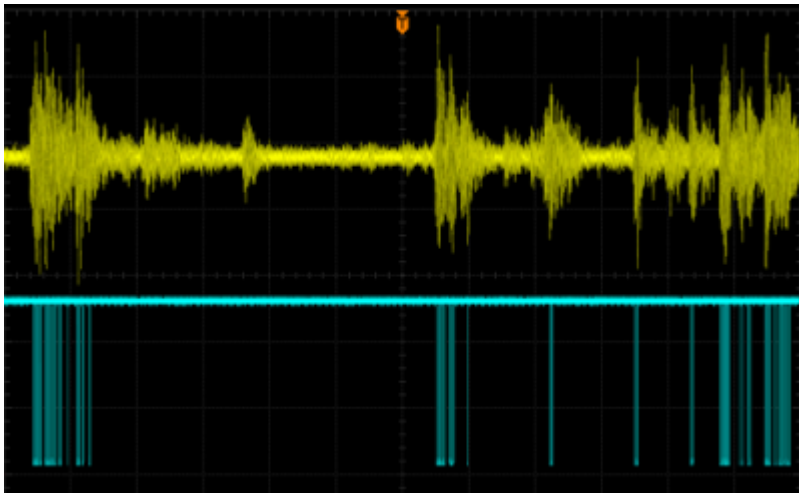
### 问题：声音传感器的输出量是模拟量还是数字量？

- 既有模拟量又有数字量，模拟量可通过示波器去观察，也可直接通过AD转换器读取。

### 问题：AOUT DOUT输出的是什么信号？

AOUT输出的实际上输出的就是声音信号，接上喇叭可以听到声音。声音信号变化很快，如果要采集声音信号需要很高采样率。例如电话采样率为8KHZ。波形的振幅表示声音的强度大小，声音越大振幅越大。故通过单片机采集到的AD值不能表示声音的大小。此模块也不能测声音具体多少分贝。

DOUT输出的是高低电平，当声音超过设置的阈值，输出低电平。



**问题：A跟D标志的两个可调电阻有什么用？**

A是调节放大增量，D是设置阈值。Aout大过阈值，Dout输出低电平，反之输出高电平

**问题：传感器可以识别的频段是多少？**

50HZ ~ 20KH