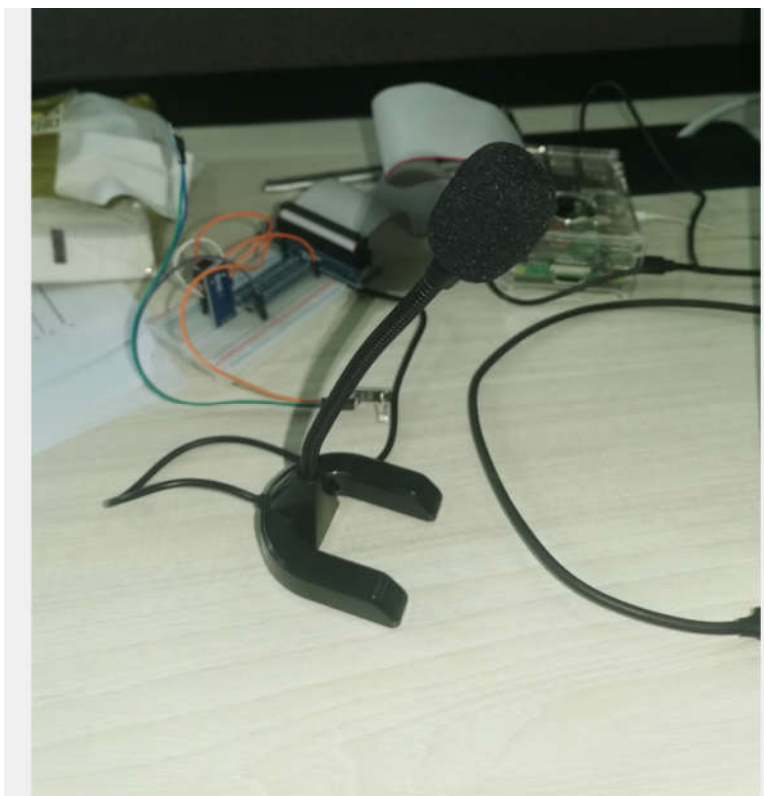


智能系统与控制

树莓派: SnowBoy 命令字检测



于泓
鲁东大学
信息与电气工程学院
2022.7.13

Usb麦克风的安装

执行 `cat /proc/asound/cards` 检测USB麦克风是否安装成功

```
pi@raspberrypi:~/snowboy $ cat /proc/asound/cards
0 [Headphones      ]: bcm2835_headphonbcm2835 Headphones - bcm2835 Headphones
bcm2835 Headphones
1 [UACDemoV10      ]: USB-Audio - UACDemoV1.0
Jieli Technology UACDemoV1.0 at usb-0000:01:00.0-1.3, full speed
```

`sudo apt-get install omxplayer` 安装播放器

```
pi@raspberrypi:~/snowboy $ sudo arecord -D "plughw:1,0" -d 5 temp.wav
MoTTY X11 proxy: Unsupported authorisation protocol
xcb_connection_has_error() returned true
Recording WAVE 'temp.wav' : Unsigned 8 bit, Rate 8000 Hz, Mono
```

录音测试

安装一些依赖包

```
sudo apt-get install pulseaudio
```

```
sudo apt-get install sox
```

```
sudo apt-get install python3-pyaudio
```

```
sudo apt-get install swig
```

```
sudo apt-get install libatlas-base-dev
```

Sox测试 `sox -d -d`

```
pi@raspberrypi:~/snowboy/m_snowboy $ sox -d -d
sox WARN alsa: can't encode 0-bit Unknown or not applicable
sox WARN alsa: can't encode 0-bit Unknown or not applicable

default: (alsa)

File Size: 42949668.24E
Encoding: Signed PCM
Channels: 2 @ 16-bit
Samplerate: 48000Hz
Replaygain: off
Duration: unknown

In:0.00% 00:00:08.36 [00:00:00.00] Out:401k [=====|=====] Hd:0.0 Clip:0
```

2022/8/24

修改默认麦克设备 `sudo nano ~/.asoundrc`

修改前

```
cm.!default {
    type asym
    playback.pcm {
        type plug
        slave.pcm "output"
    }
    capture.pcm {
        type plug
        slave.pcm "input"
    }
}

pcm.output {
    type hw
    card 0
}

ctl.!default {
    type hw
    card 0
}
```

修改后

```
cm.!default {
    type asym
    playback.pcm {
        type plug
        slave.pcm "output"
    }
    capture.pcm {
        type plug
        slave.pcm "hw:1,0"
    }
}

pcm.output {
    type hw
    card 0
}

ctl.!default {
    type hw
    card 1
}
```

Snow编译

下载: `git clone https://github.com/Kitt-AI/snowboy.git`

进行编译 `cd snowboy/swig/Python3 && make`

新建一个文件夹
复制文件

/home/pi/snowboy/m_snowboy/

Name	Size (KB)
..	
__pycache__	
resources	
_snowboydetect.so	977
demo.py	1
demo2.py	1
demo3.py	1
demo4.py	2
snowboy.umdl	792
snowboydecoder.py	10
snowboydetect.py	6

/home/pi/snowboy/snowboy/examples/Python3/

Name	Size (KB)
..	
__pycache__	
snowboydetect.py	1
snowboydecoder.py	10
snowboy.umdl	792
resources	1
requirements.txt	1
demo4.py	2
demo3.py	1
demo2.py	1
demo.py	1
_snowboydetect.so	1

/home/pi/snowboy/snowboy/resources/models/

Name	Size (KB)
..	
computer.umdl	774
hey_extreme.umdl	774
jarvis.umdl	3 449
neoya.umdl	2 202
smart_mirror.umdl	812
snowboy.umdl	792
subex.umdl	774
view_glass.umdl	774

/home/pi/snowboy/snowboy/swig/Python3/

Name	Size (KB)
..	
snowboydetect.py	6
snowboy-detect-swig.o	404
snowboy-detect-swig.i	1
snowboy-detect-swig.cc	182
Makefile	2
_snowboydetect.so	977

编译成功
后的文件

/home/pi/snowboy/snowboy/

Name	Size (KB)
..	
swig	
scripts	
resources	
lib	

修改 snowboydetector.py 第5行

```
1  #!/usr/bin/env python
2
3  import collections
4  import pyaudio
5  import snowboydetect
```

执行测试: python3 demo.py snowboy.umdl

```
pi@raspberrypi:~/snowboy/m_snowboy $ python3 demo.py snowboy.umdl
```

对话筒清晰的说出 snowboy

检测输出

```
pi@raspberrypi:~/snowboy/m_snowboy $ python3 demo.py snowboy.umdl
Listening... Press Ctrl+C to exit
Expression 'alsa_snd_pcm_hw_params_set_period_size_near( pcm, hwParams, &alsaPeriodFr
pi/alsa/pa_linux_alsa.c', line: 924
Expression 'alsa_snd_pcm_hw_params_set_period_size_near( pcm, hwParams, &alsaPeriodFr
pi/alsa/pa_linux_alsa.c', line: 924
Expression 'alsa_snd_pcm_hw_params_set_period_size_near( pcm, hwParams, &alsaPeriodFr
pi/alsa/pa_linux_alsa.c', line: 934
Expression 'alsa_snd_pcm_hw_params_set_period_size_near( pcm, hwParams, &alsaPeriodFr
pi/alsa/pa_linux_alsa.c', line: 934
Expression 'alsa_snd_pcm_hw_params_set_period_size_near( pcm, hwParams, &alsaPeriodFr
pi/alsa/pa_linux_alsa.c', line: 934
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
Expression 'alsa_snd_pcm_hw_params_set_period_size_near( pcm, hwParams, &alsaPeriodFr
pi/alsa/pa_linux_alsa.c', line: 924
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
INFO:snowboy:Keyword 1 detected at time: 2022-07-13 10:07:05
INFO:snowboy:Keyword 1 detected at time: 2022-07-13 10:07:13
INFO:snowboy:Keyword 1 detected at time: 2022-07-13 10:07:16
```

Snow的应用

















(1) 自定义模型训练

<https://snowboy.hahack.com/>

Model Name:

music_off

Examples (3 required)

1.  0:02 / 0:02   
2.  0:02 / 0:02   
3.  0:02 / 0:02   
4.  0:02 / 0:02   

joan YouTube 多瑞影院 Google 学术搜索 虎扑体育论坛 Google 翻译 This man is trying t...

1. Enable microphone (required for Chrome)
2. Click Record and wait until ready
3. Speak your wake word and **wait for end**
4. Repeat until you have 3 examples
5. Enter a model name, submit audio, and click "Save Model" button


Microphone ready!

 Record

点击Record 录音

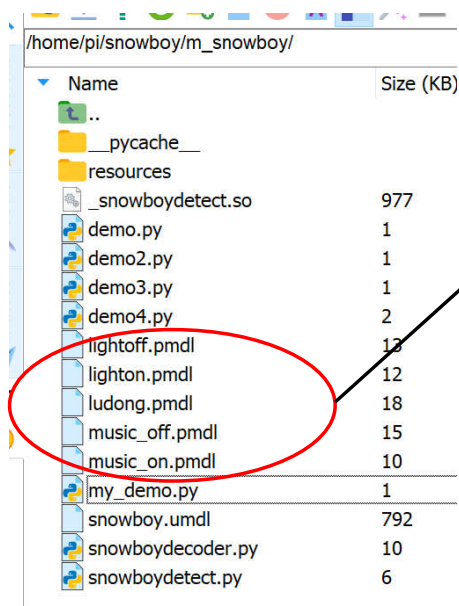
 Submit

上传

 Save Model

模型下载

2022/8/24



Name	Size (KB)
..	
__pycache__	
resources	
_snowboydetect.so	977
demo.py	1
demo2.py	1
demo3.py	1
demo4.py	2
lightoff.pmdl	13
lighton.pmdl	12
ludong.pmdl	18
music_off.pmdl	15
music_on.pmdl	10
my_demo.py	1
snowboy.umdl	792
snowboydecoder.py	10
snowboydetect.py	6

开灯
关灯
奏乐
停止演奏

生命模型

检测后
的响应
函数

定义检测器

```
if __name__ == "__main__":  
  
    # 定义模型以及相应的响应函数  
    models = ['lighton.pmdl', 'lightoff.pmdl', 'music_on.pmdl', 'music_off.pmdl']  
    callbacks = [lambda: callbakck_light_on(),  
                 lambda: callbakck_light_off(),  
                 lambda: callbakck_music_on(),  
                 lambda: callbakck_music_off()]  
  
    # 定义检测器  
    signal.signal(signal.SIGINT, signal_handler)  
    sensitivity = [0.5]*len(models)  
    detector = snowboydecoder.HotwordDetector(models, sensitivity=sensitivity)  
    print('开始检测')  
  
    # 主循环, 每隔一段时间检测一次  
    detector.start(detected_callback=callbacks,  
                  interrupt_check=interrupt_callback,  
                  sleep_time=0.03)  
  
    detector.terminate()
```

```
import snowboydecoder
import sys
import signal
from pin_dic import pin_dic
from light import light
import threading
from buzzer import Runing_Song
import time

# 定义小灯对象
global m_light
m_light = light(pin_dic["G17"])

# 蜂鸣器
global m_runing_song
m_runing_song = Runing_Song(pin_dic['G18'])

def callbakck_light_on():
    global m_light
    m_light.on()
    print("light_on")

def callbakck_light_off():
    global m_light
    m_light.off()
    print("light_off")
```

```
def callbakck_music_on():
    global m_runing_song
    if m_runing_song.isAlive() == False:
        # 没有线程 创建线程并启动
        m_runing_song = Runing_Song(pin_dic['G18'])
        m_runing_song.file_load('music.txt')

        m_runing_song.setDaemon(True)
        m_runing_song.start()

    else:
        # 如果正在演奏, 先停止
        m_runing_song.dostop()
        time.sleep(0.1)
        m_runing_song.join()

        # 重新加载
        m_runing_song = Runing_Song(pin_dic['G18'])
        flag = m_runing_song.file_load('music.txt')
        m_runing_song.setDaemon(True)
        m_runing_song.start()

    print("music_on")

def callbakck_music_off():
    global m_runing_song
    if m_runing_song.isAlive() == True:
        m_runing_song.dostop()
        time.sleep(0.1)
        m_runing_song.join()
    print("music_off")
```



```
interrupted = False

def signal_handler(signal, frame):
    global interrupted
    interrupted = True

def interrupt_callback():
    global interrupted
    return interrupted
```