https://developers.google.com/assistant/sdk/guides/service/python
** 괜히 블로그 찾지 말고 이거대로 따라하는 게 나을 것 같음.

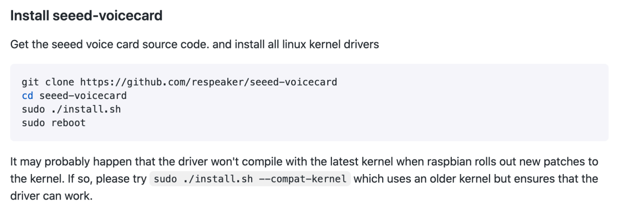
1. Set up hardware and Network Access :

그냥 라즈베리파이 쉘 접속하는 방법. 예림이 핫스팟으로 접속

2. Congifure and Test the Audio :

참고 자료 : https://github.com/respeaker/seeed-voicecard

현재 우리가 가지고 있는 모델(https://www.elecrow.com/wiki/index.php?title=Speech_Interaction_board_for_Raspberry_Pi_)은 정보가 많이 없어서, 동일하게 마이크 두 개 달려있는 respeaker github 참고하는 것이 나을 것 같다고 해서 따라함.



Install seeed-voicecard

Get the seeed voice card source code. and install all linux kernel drivers

```
git clone https://github.com/respeaker/seeed-voicecard
cd seeed-voicecard
sudo ./install.sh
sudo reboot
```

It may probably happen that the driver won't compile with the latest kernel when raspbian rolls out new patches to the kernel. If so, please try `sudo ./install.sh --compat-kernel` which uses an older kernel but ensures that the driver can work.

- 초기 셋팅법

```
git clone https://github.com/
respeaker/seeed-voicecard
cd seeed-voicecard
sudo ./install.sh
sudo reboot
```

((현재 sudo ./install.sh —compat-kernel 까지 완료한 상태임))
>> 다 설치 되었으면 아래와 같은 결과창이 뜨는 게 정상.

```
pi@raspberrypi:~ $ source env/bin/activate
(env) pi@raspberrypi:~ $ cd seeed-voicecard

(env) pi@raspberrypi:~/seeed-voicecard $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 7/7
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 IEC958/HDMI [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: ALSA [bcm2835 ALSA], device 2: bcm2835 IEC958/HDMI1 [bcm2835 IEC958/HDMI1]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: seeed2micvoicec [seeed-2mic-voicecard], device 0: bcm2835-i2s-wm8960-hifi
wm8960-hifi-0 [bcm2835-i2s-wm8960-hifi wm8960-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

3. Configure a Developer Project and Account Settings :

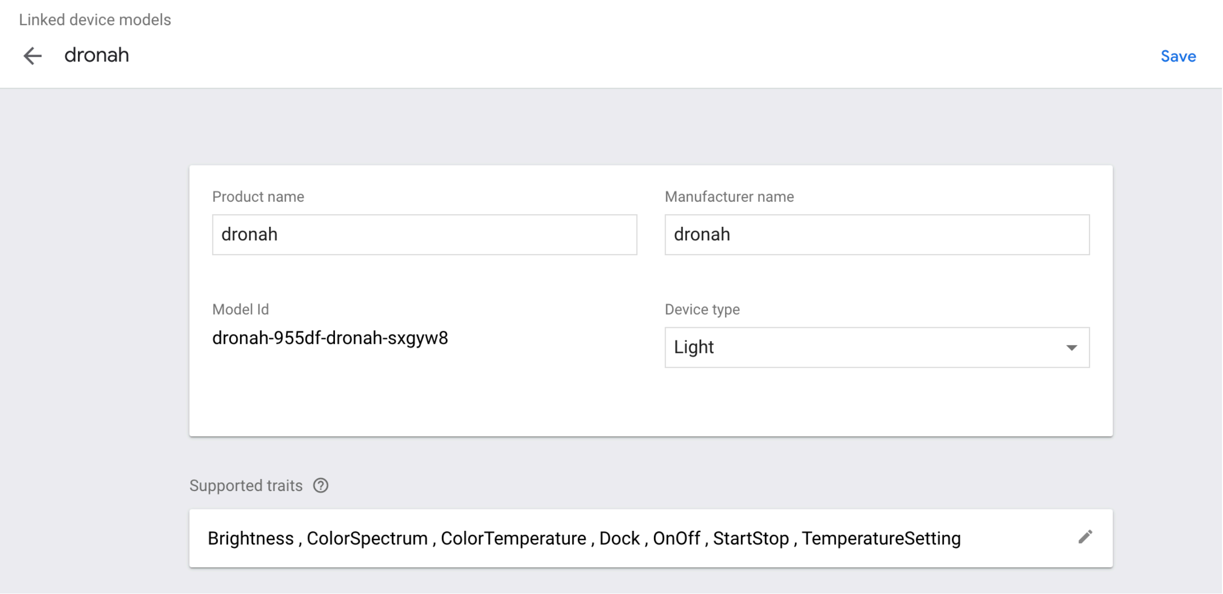
https://console.actions.google.com/u/0/project/dronah-955df/deviceregistration/ 로 가면 보임.

>>

4. Register the Device Model:

https://developers.google.com/assistant/sdk/guides/service/python/embed/register-device

일단 내용을 읽어보면 UI를 통해서 디바이스 모델을 등록하고 난 다음에, 해당 디바이스로부터 어시스턴트 접근이 가능함. 즉 모델(device model) - 개별 디바이스(device instance) 로 상하단이 나뉘어있음.



dronah

Product name: dronah

Manufacturer name: dronah

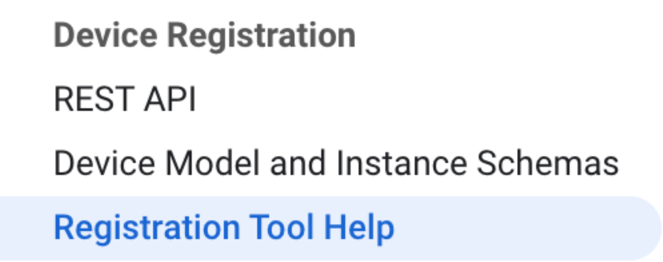
Model Id: dronah-955df-dronah-sxgyw8

Device type: Light

Supported traits: Brightness , ColorSpectrum , ColorTemperature , Dock , OnOff , StartStop , TemperatureSetting

-> 모델이 만들어진 상태이고, 해당 모델에 대한 크레덴셜도 등록되어있음!!

https://developers.google.com/assistant/sdk/reference/device-registration/device-tool



-> 인스턴스 등록 방법은 세가지인데, registration tool이용하는게 가장 간단해 보임

```
(env) pi@raspberrypi:~/seeed-voicecard $ googlesamples-assistant-devicetool
Usage: googlesamples-assistant-devicetool [OPTIONS] COMMAND [ARGS]...

Options:
  --project-id TEXT    Enter the Google Developer Project ID that you want to
                       use with the registration tool. If you don't use this
                       flag, the tool will use the project listed in the
                       client_secret_<client-id>.json file you specify with
                       the --client-secrets flag.  [required]
  --verbose            Shows detailed JSON response
  --api-endpoint TEXT  Hostname for the Google Assistant API. Do not use this
                       flag unless explicitly instructed.  [default:
                       embeddedassistant.googleapis.com]
  --credentials TEXT   File location of the generated credentials file. The
                       google-oauthlib-tool generates this file after
                       authorizing the user with the client_secret_<client-
                       id>.json file. This credentials file authorizes access
                       to the Google Assistant API. You can use this flag if
                       the credentials were generated in a location that is
                       different than the default.  [default:
                       /home/pi/.config/google-oauthlib-tool/credentials.json]
  --help               Show this message and exit.

Commands:
  delete           Delete given device model or instance.
  get              Gets all of the information (fields) for a...
  list             Lists all of the device models and/or...
  register         Registers a device model and instance.
  register-device  Registers a device instance under an existing...
  register-model   Registers a device model.
```

내용을 읽어보면 우리는 register-device로 등록해야함을 알 수 있음.

```
(env) pi@raspberrypi:~ $ googlesamples-assistant-devicetool --project-id "dronah-955df" --
verbose register-device --device No1 --model "dronah-955df-dronah-sxgyw8" --client-type
"SERVICE"
Converted retries value: 3 -> Retry(total=3, connect=None, read=None, redirect=None,
status=None)
{"id": "No1", "model_id": "dronah-955df-dronah-sxgyw8", "client_type": "SDK_SERVICE"}
Starting new HTTPS connection (1): embeddedassistant.googleapis.com:443
https://embeddedassistant.googleapis.com:443 "GET /v1alpha2/projects/dronah-955df/
devices/No1 HTTP/1.1" 404 None
Creating new device
https://embeddedassistant.googleapis.com:443 "POST /v1alpha2/projects/dronah-955df/
devices HTTP/1.1" 200 None
Device instance No1 successfully registered
{
  "id": "No1",
  "modelId": "dronah-955df-dronah-sxgyw8",
  "clientType": "SDK_SERVICE"
}
```

이렇게 인스턴스 등록자체는 가능하나,

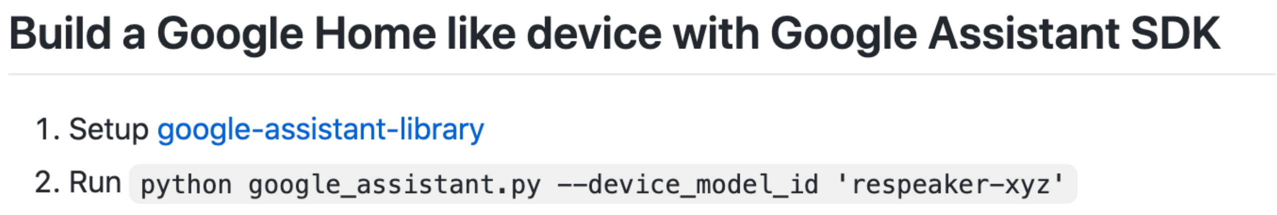
```
(env) pi@raspberrypi:~ $ googlesamples-assistant-devicetool list --model
Usage: googlesamples-assistant-devicetool [OPTIONS] COMMAND [ARGS]...

Error: Missing option "--project-id".
(env) pi@raspberrypi:~ $ googlesamples-assistant-devicetool --project-id "dronah-955df" list --
model
Error: Failed to list resources: 403
The caller does not have permission
```

리스트 열람 조회는 되지 않는 상태고,

```
(env) pi@raspberrypi:~ $ google-assistant-demo --device_model_id "dronah-955df-dronah-
sxgyw8"
/home/pi/env/lib/python3.7/site-packages/google/assistant/library/assistant.py:90:
DeprecationWarning: Google Assistant Library for Python is deprecated
  warnings.warn('Google Assistant Library for Python is deprecated', DeprecationWarning)
device_model_id: dronah-955df-dronah-sxgyw8
device_id: 10EC0556C64E69ADFFD809B4CBC818AE


    This device is not registered. This means you will not be able to use
    Device Actions or see your device in Assistant Settings. In order to
    register this device follow instructions at:

    https://developers.google.com/assistant/sdk/guides/library/python/embed/register-device

ON_MUTED_CHANGED:
  {"is_muted": false}
ON_START_FINISHED
ON_MEDIA_STATE_IDLE
^CSegmentation fault
The caller does not have permission
```

구글 어시스턴스 실행 시, 이 인스턴스가 무슨 인스턴스인지, 디바이스 등록안됐다고 뜨면서 인식하지 못하고 있는 상황임 !!! 하지만 접속자 제어를 못하는거지 구글어시스턴스 실행자체는 계쏙 됨.

https://github.com/respeaker/mic_hat/blob/master/google_assistant.py
>> 그리고 아직 배송오지 않고 있는 mic_hat은 이러한 일련의 과정을 자동화 시켜주는 ,,, 소스코드가 있음
이것도 근데 크레덴셜이 필요해서, 일단 모델 등록까지는 알아서 완료해야하는 거고



Build a Google Home like device with Google Assistant SDK

1. Setup google-assistant-library
2. Run `python google_assistant.py --device_model_id 'respeaker-xyz'`

어쨌든 device_model_id를 본인 임의로 하는 것 같음. 그래서 이해가 필요하긴 했음.
Google assistant.py 부분 수정을 통해서 라즈베리파이 쉘 내부에서 정돈된 로그 띄우는 게 가능하도록 해야할 것 같음.

———————————-——————————-————————————-———————

참고한 한국어
https://m.blog.naver.com/chandong83/221081942490
https://ukayzm.github.io/installing-google-assistant/

일단 모든 사용자에 대한 접근은 허용한 상태기 때문에, google assistant와 action on google을 연동시키는 것을 하면 DB변동은 가능할 것 같음!! 이부분에 대해서 예림이가 해야함!!