라즈베리파이를 이용한 인공지능 스피커 만들기

Part 3

김영준
목원대학교 겸임교수
前 마이크로소프트 수석연구원
헬로앱스 대표이사
Http://www.helloapps.co.kr
splduino@gmail.com

나만의 프로젝트 파일 생성

예제 다운로드

Get the source code

You are now ready to start your own project:

```
(env) $ git clone https://github.com/googlesamples/assistant-sdk-python
```

python3 -m venv env
source env/bin/activate

git clone https://github.com/googlesamples/assistant-sdk-python

예제 실행하기

cd assistant-sdk-python/google-assistant-sdk/googlesamples/assistant/library

python hotword.py --device-model-id **<my-model>**

자신의 디바이스 모델 id로 대체

배치 실행 파일 생성

한번에 실행되는 명령어 만들기

텍스트 편집기를 열어서 새로운 파일 생성
아래의 4개 라인을 편집기 파일에 복사한 후, 파일 저장
<my-model-id> 부분은 자신의 값으로 수정

```
python3 -m venv env
source env/bin/activate
cd assistant-sdk-python/google-assistant-sdk/googlesamples/assistant/library
python hotword.py --device-model-id <my-model-id>
```

쉘 파일 (배치파일)로 만들기

텍스트 편집기를 열어서 새로운 파일 생성
/home/pi 폴더 아래에 아래의 이름으로 저장

launcher.sh

```
python3 -m venv env
. env/bin/activate
cd assistant-sdk-python/google-assistant-sdk/googlesamples/assistant/library
python hotword.py --device-model-id <my-model-id>
```

쉘 파일 (배치파일)로 만들기

터미널에서 아래의 명령어 실행

chmod 755 launcher.sh

쉘 파일 (배치파일) 실행 시키기

터미널에서 아래의 명령어 실행

```
sh launcher.sh
```

자동 실행 환경 구성

자동 실행 환경 구성 (부팅후 자동 실행)

터미널에서 아래의 명령어 실행

```
crontab -e
```

자동 실행 환경 구성 (부팅후 자동 실행)

화면 편집기에서 맨 아래에 아래의 코드를 새로운 라인으로 붙여넣기

```
@reboot sh /home/pi/launcher.sh
```

Ctrl+O 키를 눌러 저장
엔터 키를 한번 더 눌러 파일 저장

Ctrl+X 키로 빠져 나감

crontab –e 를 다시 실행하여 수정되었는 지 확인

Ctrl+X 키로 빠져 나감

자동 실행 환경 구성 (부팅후 자동 실행)

재시작 함 (Reboot)

부팅후, 한 10초 정도 기다렸다가 대화 시도해 봄

대화는 진행되지만 별도의 화면 응답이 없음에 유의함

원래 되로 되돌리기

터미널에서 아래의 명령어 실행

```
crontab -e
```

원래 되로 되돌리기

화면 편집기에서 아래의 코드 맨 앞에 # 추가하여 **주석처리함**

#@reboot sh /home/pi/launcher.sh

Ctrl+O 키를 눌러 저장
엔터 키를 한번 더 눌러 파일 저장

Ctrl+X 키로 빠져 나감

crontab –e 를 다시 실행하여 수정되었는 지 확인

Ctrl+X 키로 빠져 나감

원래 되로 되돌리기

재시작함

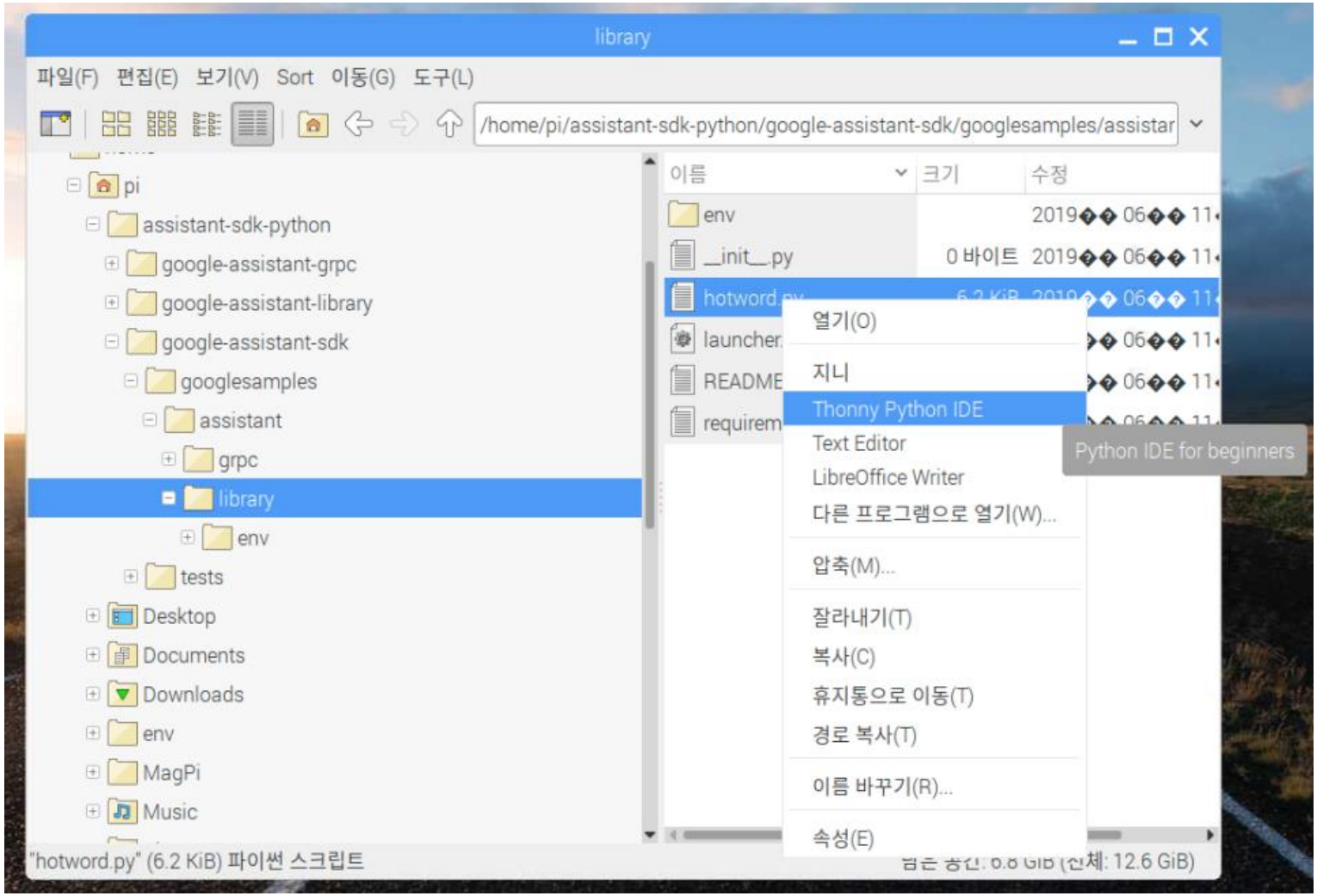
소스 둘러 보기

소스 둘러보기

/home/pi/

assistant-sdk-python/google-assistant-sdk/googlesamples/assistant/library/hotword.py

소스 둘러보기


library
파일(F) 편집(E) 보기(V) Sort 이동(G) 도구(L)
/home/pi/assistant-sdk-python/google-assistant-sdk/googlesamples/assistar
pi
assistant-sdk-python
google-assistant-grpc
google-assistant-library
google-assistant-sdk
googlesamples
assistant
grpc
library
env
tests
Desktop
Documents
Downloads
env
MagPi
Music
이름
크기
수정
env
__init__.py
0 바이트
hotword.py
launcher
README
requirem
열기(O)
지니
Thonny Python IDE
Text Editor
LibreOffice Writer
다른 프로그램으로 열기(W)...
압축(M)...
잘라내기(T)
복사(C)
휴지통으로 이동(T)
경로 복사(T)
이름 바꾸기(R)...
속성(E)
Python IDE for beginners
"hotword.py" (6.2 KiB) 파이썬 스크립트

소스 둘러보기

```
def process_event(event):
    """Pretty prints events.

    Prints all events that occur with two spaces between each new
    conversation and a single space between turns of a conversation.

    Args:
        event(event.Event): The current event to process.
    """
    if event.type == EventType.ON_CONVERSATION_TURN_STARTED:
        print()

    print(event)

    if (event.type == EventType.ON_CONVERSATION_TURN_FINISHED and
            event.args and not event.args['with_follow_on_turn']):
        print()
    if event.type == EventType.ON_DEVICE_ACTION:
        for command, params in event.actions:
            print('Do command', command, 'with params', str(params))
            if command == "action.devices.commands.OnOff":
```

소스 둘러보기

```
def main():
    parser = argparse.ArgumentParser(
        formatter_class=argparse.RawTextHelpFormatter)
    parser.add_argument('--device-model-id', '--device_model_id', type=str,
                        metavar='DEVICE_MODEL_ID', required=False,
                        help='the device model ID registered with Google')
    parser.add_argument('--project-id', '--project_id', type=str,
                        metavar='PROJECT_ID', required=False,
                        help='the project ID used to register this device')
    parser.add_argument('--nickname', type=str,
                        metavar='NICKNAME', required=False,
                        help='the nickname used to register this device')
    parser.add_argument('--device-config', type=str,
                        metavar='DEVICE_CONFIG_FILE',
                        default=os.path.join(
                            os.path.expanduser('~/.config'),
                            'googlesamples-assistant',
                            'device_config_library.json'
                        ),
                        help='path to store and read device configuration')
    parser.add_argument('--credentials', type=existing_file,
                        metavar='OAUTH2_CREDENTIALS_FILE',
                        default=os.path.join(
```

새로운 Trait 추가

새로운 Trait 추간

Register Traits

☆☆☆☆☆

목
Ac
Ne

The Google Assistant needs to be able to associate a query with a command to send to your device. For this to work, you need to declare what kinds of abilities your device supports. These abilities are known as *traits*. You declare these traits within your device model.

Google has already created a wide variety of common traits found on many devices. These traits are not tied to just one device type, you can use them as you choose.

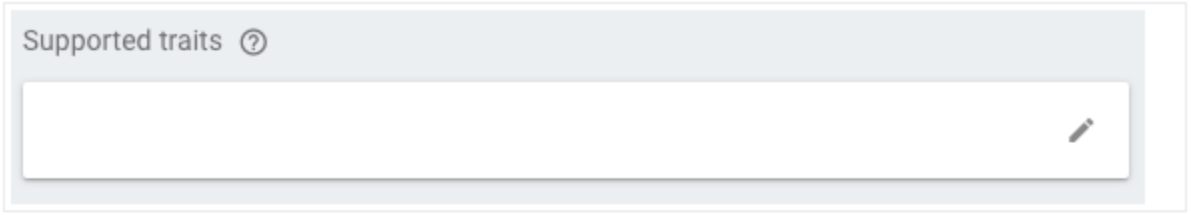
Add a trait

You previously defined a model, now update it by adding a trait. In this case, add an On/Off trait to control an LED attached to your device.

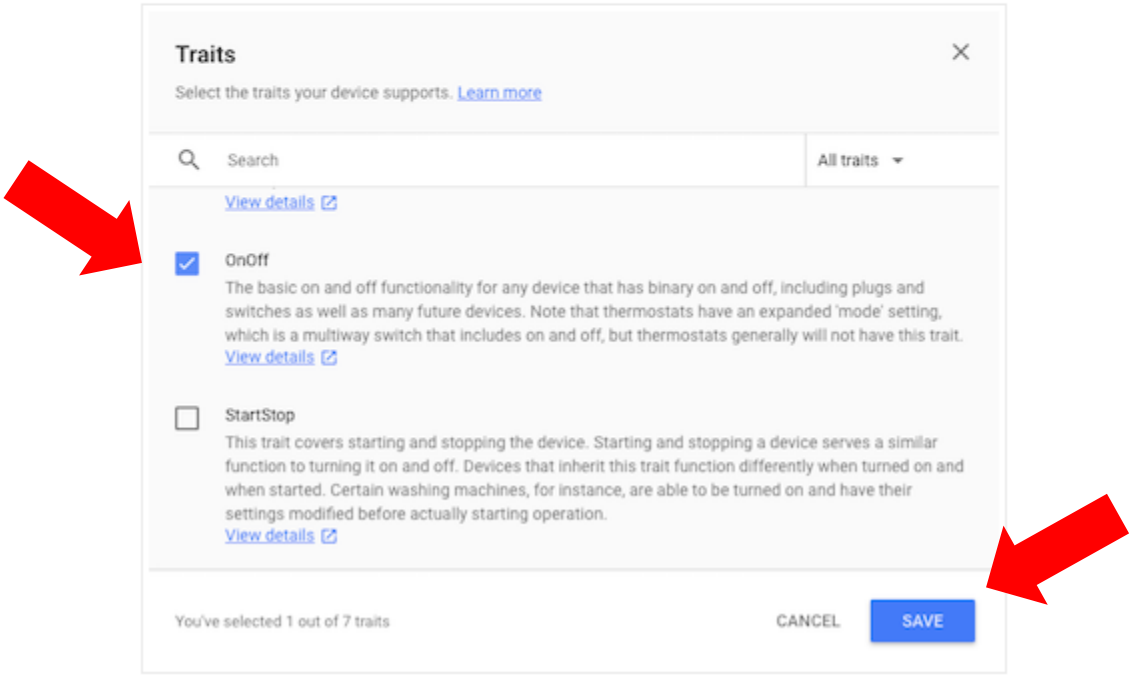
1. Open the project in the Actions Console.
2. Select the **Device registration** tab from the left navbar.
3. Click a model from the list to edit it.

새로운 Trait 추간

4. Click the pencil in the **Supported traits** box to add the trait.

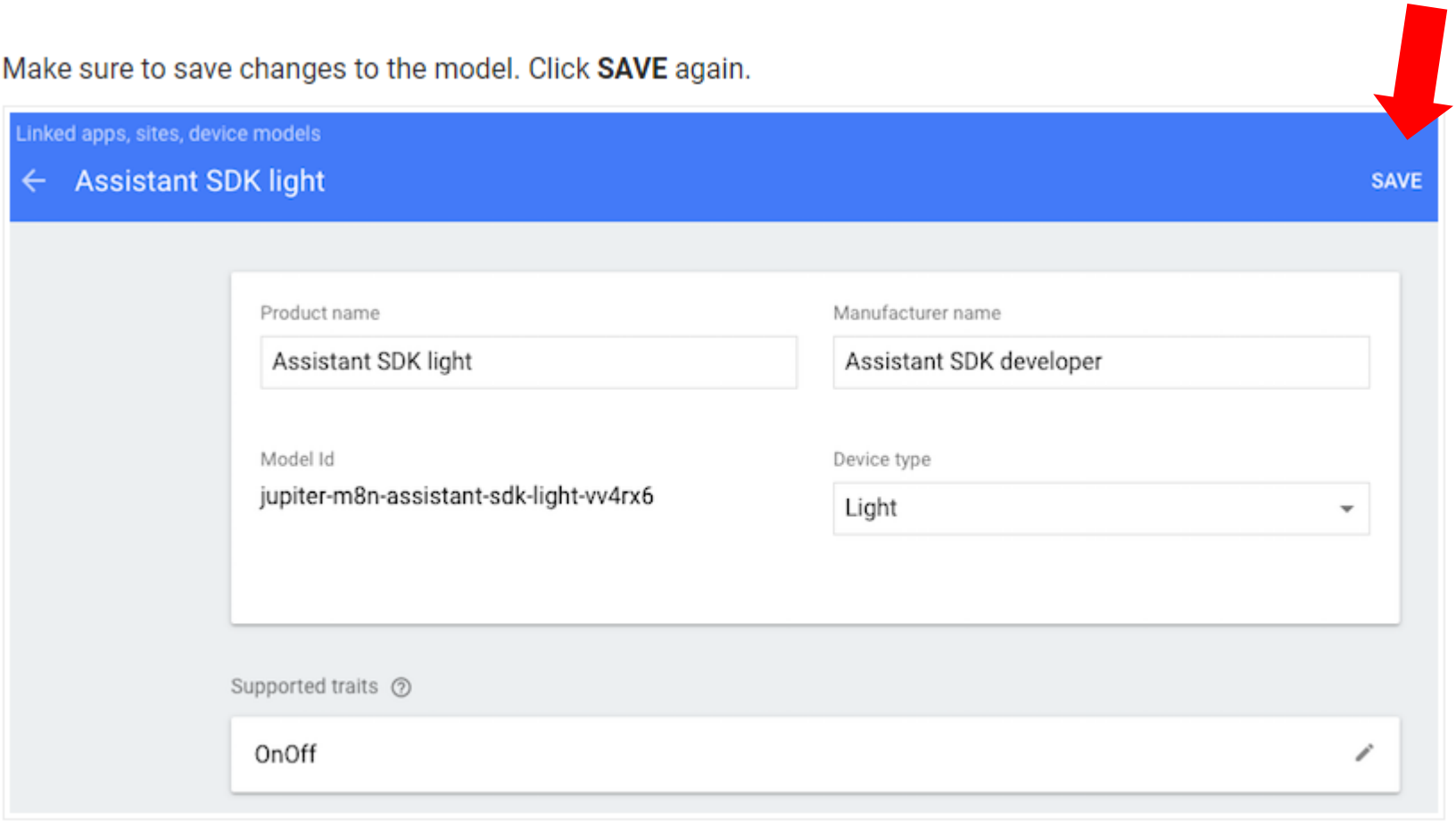


5. Select the **OnOff** checkbox. Click **SAVE**.



한번 더 SAVE 버튼 클릭

6. Make sure to save changes to the model. Click **SAVE** again.



실행 테스트

실행 후, 아래의 음성으로 테스트

OK Google Turn On

OK Google Turn Off

실행 결과

```
ON_RECOGNIZING_SPEECH_FINISHED:
  {'text': 'turn on'}
ON_DEVICE_ACTION:
  {'inputs': [{'payload': {'commands': [{'execution': [{'command': 'action.devices.commands.OnOff',
  'params': {'on': True}}], 'devices': [{'id': 'E56D39D894C2704108758EA748C71255'}]}]},
  'intent': 'action.devices.EXECUTE'}], 'requestId': '4785538375947649081'}
Do command action.devices.commands.OnOff with params {'on': True}
```

파이썬 샘플 코드 추가

파이썬 코드 추가

```
def process_event(event):
    """Pretty prints events.

    Prints all events that occur with two spaces between each new
    conversation and a single space between turns of a conversation.

    Args:
        event(event.Event): The current event to process.
    """
    if event.type == EventType.ON_CONVERSATION_TURN_STARTED:
        print()

    print(event)

    if (event.type == EventType.ON_CONVERSATION_TURN_FINISHED and
            event.args and not event.args['with_follow_on_turn']):
        print()
    if event.type == EventType.ON_DEVICE_ACTION:
        for command, params in event.actions:
            print('Do command', command, 'with params', str(params))
            if command == "action.devices.commands.OnOff":
                if params['on']:
                    print('Turning the LED on.')
                else:
                    print('Turning the LED off.')
```

파이썬 코드 추가

```
if event.type == EventType.ON_DEVICE_ACTION:
        for command, params in event.actions:
            print('Do command', command, 'with params', str(params))
            if command == "action.devices.commands.OnOff":
                if params['on']:
                    print('Turning the LED on.')
                else:
                    print('Turning the LED off.')
```

실행 테스트

실행 후, 결과 확인

OK Google Turn On

OK Google Turn Off