

ラズベリーパイで
AIスピーカーを作ろう！

Raspberry Pi+ReSpeaker+Google Assistant SDK+α

2018/04/22(日)
FabLab関内

準備:必要な物の確認

【ご持参、事前準備頂くもの】

- スマートホンまたはタブレット (Android, iOS)
- ノートPC (Windows, Mac, Linux)
- Googleアカウント
- Raspberry Pi 3 本体
- Raspberry Pi用電源
 - ACアダプター or モバイルバッテリー

【貸出するもの】

- LANケーブル(貸出)
- SDカードリーダー ライター (microSD, SDカード用)

【配布するもの】

- microSDカード(設定済)
- ReSpeaker 2-Mics Pi HAT
- マイクロ・スピーカー
- プッシュ・ライト

お配りしたmicroSDについて

- OS: Raspbian Stretch (2018-03-18版)
- OS初期設定の主な内容(詳しくは別途)
 - コンソール起動、自動ログイン無し
※ GUI(デスクトップ)起動にするとGoogle Assistantが正常に動かなくなります！
 - 地域設定(タイムゾーンなど): 日本(東京)
※ デフォルトのロケールは「C.UTF-8」→コンソール・メッセージなどは英語
 - ssh、SPI・I2Cインターフェース: 有効
- 主なプレインストールアプリ(他にもいろいろ入れてます)
 - Google Assistant SDK
 - Python開発環境
 - OpenJTalk: 日本語音声合成ソフト
→ 起動時にIPアドレスを読上げるように設定してあります！
 - VLC: マルチメディア・プレーヤー

microSDのバックアップについて

PCで全ファイルを普通にコピーしてもダメです！！

ディスクイメージを全てコピーするソフトが必要です。
(バックアップ時も、復元時も)

※ 自分でRaspbian OSイメージをmicroSDに書き込むときにも必要

- Windows: 「Win32 Disk Imager」、「Etcher」など
- Mac: ターミナルのddコマンド、「ApplePi Baker」、「Etcher」など
- Linux: ddコマンド

ここでやること

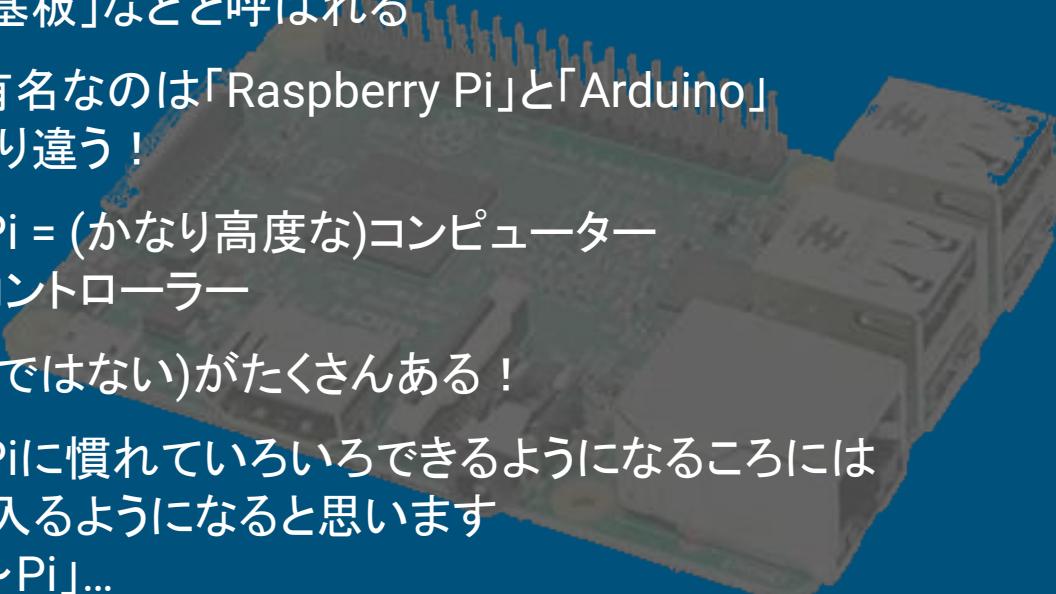
- Raspberry PiとReSpeaker基板とGoogle Assistant SDKを組み合わせて、AIスピーカーを作ります。
- Raspberry Piの利点を活かして、改造・拡張をして、
製品にない拡張や機能が実現できることを体験します。
※「スキル」や「Google Assistantアプリ」は作りません！！



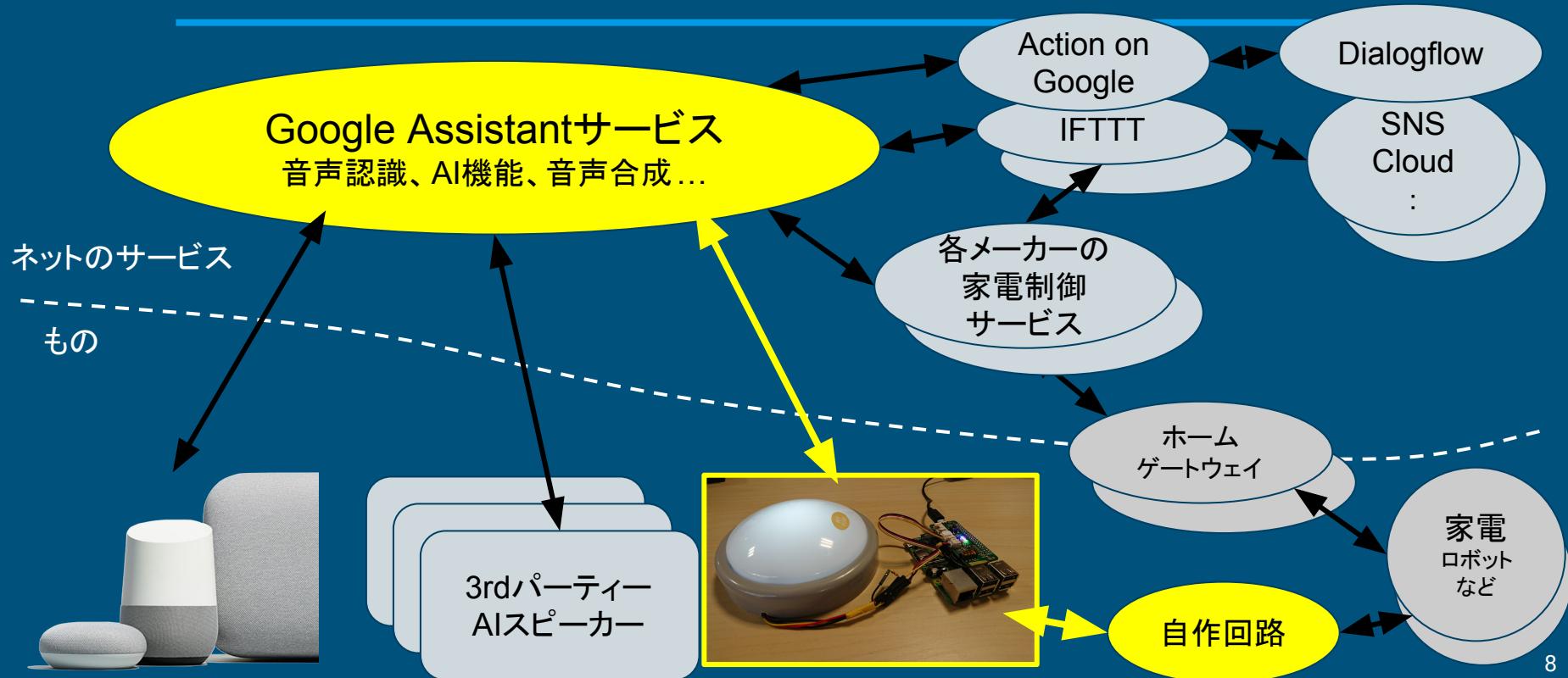
Raspberry Pi
+
ReSpeaker
+
Google Assistant SDK
+
a(アルファ)

Raspberry Piについて

- 電子工作で、いろいろな部品を制御する「頭脳」に相当
一般に「マイコン基板」などと呼ばれる
- この分野で最も有名なのは「Raspberry Pi」と「Arduino」
しかし、実はかなり違う！
 - Raspberry Pi = (かなり高度な)コンピューター
 - Arduino = コントローラー
- 類似品(偽物とかではない)がたくさんある！
 - Raspberry Piに慣れていろいろできるようになるころには
自然に耳に入るようになると思います
 - 「～ino」、「～Pi」...



Google Assistantサービスについて



Google Assistant SDKについて

「Google Assistant SDK」

<https://developers.google.com/assistant/sdk/>

Googleが無料で公開しているSDK(Software Development Kit)の一つ。
「Google Home」と同等のAIスピーカーを自分で作ることができる！

今回は、特にPythonで手軽に利用できるようになっている
「Google Assistant Library for Python」

<https://developers.google.com/assistant/sdk/guides/library/python/>
を使います。

「hotword」と「pushtotalk」サンプル

Google Assistant SDKには2つのサンプルがあり、内部動作が異なる

(1)「googlesamples-assistant-hotword」

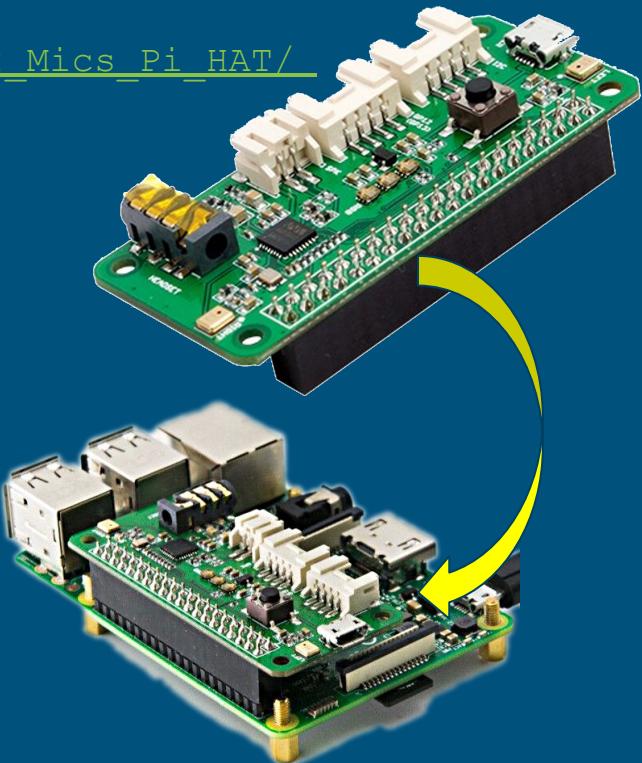
- 「Ok, Google」、「Hey, Google」などホットワードに反応する
- 「Google Assistant Library」として実装されている
(Raspberry Pi Zero Wには未対応)

(2)「googlesamples-assistant-pushtotalk」

- ターミナルの[Enter]キーに反応する(「OK, Google」はできない)
- gRPCによる「Google Assistant Service」として実装されている
(Raspberry Pi Zeroでも実行可能 ... でも、かなり遅い)

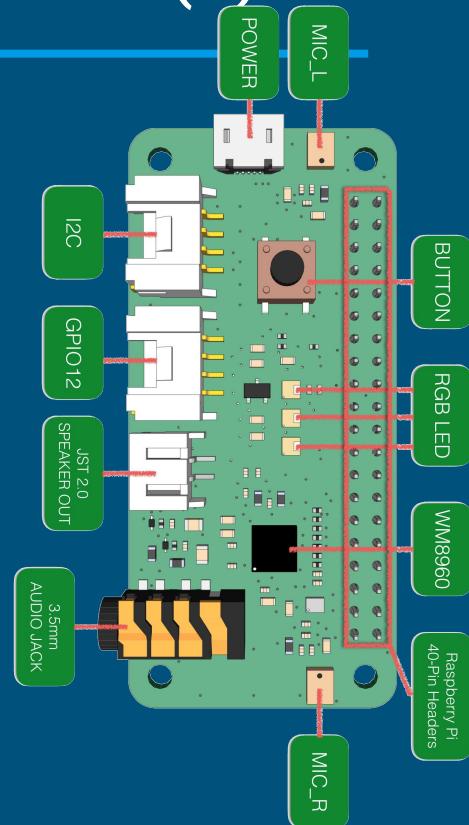
ReSpeaker 2-Mics Pi HATについて(1)

- Seeed Studio社製
http://wiki.seeedstudio.com/ReSpeaker_2_Mics_Pi_HAT/
- マイク、スピーカー端子など、
AIスピーカーに必要な回路が
一枚の基板にまとまってる
- Raspberry PiのGPIOピンにスッポリはまる、
いわゆる「HAT基板」
(Arduinoの「シールド」に相当)



ReSpeaker 2-Mics Pi HATについて(2)

- マイク x 2 (L, R)
- スピーカー端子: JST 2.0 PH規格
- ボタン: GPIO17
- フルカラーLED x 3個
- micro USB端子(電源用)
- GPIO端子: GPIO12, GPIO13, Grove対応
- I2C端子: I2C-1, Grove対応
- 3.5mmオーディオジャック: ヘッドホン用



なぜ、ReSpeaker 2-Micsなのか？

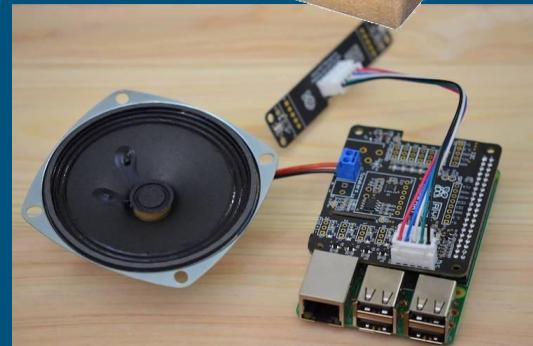
- **安い** (他のキットに比べて半額以下)
※Amazonではぼったくり価格なので注意！
※DigiKey経由で輸入がオススメですが、買い物の合計 6,000円未満だと送料が高い！
- **小さい** → Raspberry Pi Zeroと同じ
※ただし、Raspberry Pi Zeroでは動きません … 後述
- AIスピーカーに必要な**マイクとスピーカー出力を備えている**
(USBマイクとスピーカー用アンプが不要)
- **十分な拡張性も備えている**(Grove SystemのGPIO、I2C)

(参考) 他のAIスピーカーキット(1)

【Google AIY Voice Kit】

<https://aiyprojects.withgoogle.com/voice/>

- Google社AIYプロジェクト(Do-It-Yourself AI)純正のAIスピーカー自作キット
- ダンボール製の箱と押しボタン特徴
- Google純正のため、サポートや情報が充実

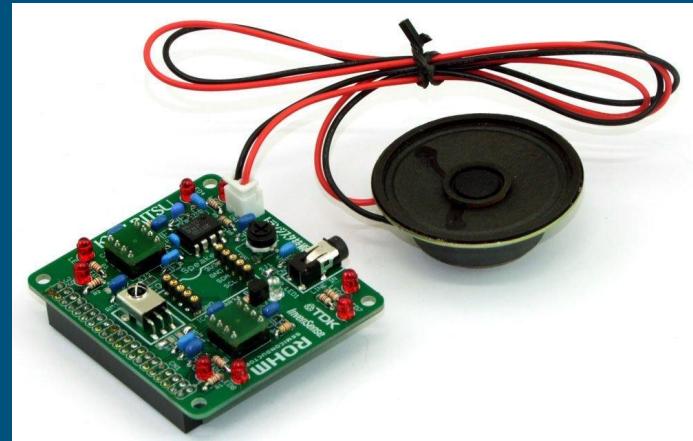


(参考) 他のAIスピーカーキット(2)

【トランジスタ技術 2018年3月号 付録】

<http://shop.cqpub.co.jp/hanbai/books/MTR/MTR201803.html>

- 雑誌の付録としては基板のみ、
部品は別売り
- AIスピーカーのための機能以外に、
多様なセンサー類や高度な回路が一枚に
まとまっている
(学習リモコン用、サーモカメラなど)



「+ a」について

デモプログラムのソースコードを修正したり、
外部の回路を接続して、
製品ではできないカスタマイズをします！

- (製品とは違う)確認音を鳴らす
- 外部機器(LEDライト)を制御
- 外部入力(ボタン)と連動
- 「連続対話機能」: 独自機能

連続対話機能:ここ独自の機能 !

【通常】

いちいちホットワードを言わなければならない

「Hey, Google. 今何時？」→ 応答 → (会話終了)

「Hey, Google. 今日の天気は？」→ 応答 → (会話終了)

「Hey, Google. 暖かいね」→ 応答 → (会話終了)

【独自動作】

「エンドワード(End word)」の概念を導入し、連続対話を可能に！

「Hey, Google. 今何時？」→ 応答 → (継続)

→「今日の天気は？」→ 応答 → (継続)

→「暖かいね」→ 応答 → (継続)

→「ありがとう(エンドワード)」→ 応答 → (会話終了)



Here we go !!

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



この資料の見方: ★印について

- 項目の先頭に「★」がついているものは、
実際に操作して頂くこと

※ 操作前に、一通り説明します。
- 「★」がないところは、
説明、解説、参考情報などです。
- 上部に全体的な流れを示します

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



まずは、PCで作業

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



IDや名前...いろいろあり過ぎ！

プロジェクトID	Googleの各種APIを使うアプリケーションやサービス毎
クライアントID (認証情報JSONファイル)	プロジェクトを利用するソフトや機器の ID・認証情報 JSON形式のファイルを Raspberry Pi に保存する
Googleアカウント (Gmailアドレス)	スマホ、PC、AIスピーカーなど共通のアカウント ID 個人情報アクセスの管理単位。プロジェクトの所有者。
manufacturer	メーカー企業名・作者名
product ID	製品名 ... スマホのデバイス一覧に表示される名前
(デバイス)モデルID	「～mini」、「～plus」、ライト、スイッチ、などの タイプと対応 ... プログラム実行時に指定する
デバイス(インスタンス)ID	機器ごとの固有の ID ... これを登録するとスマホのデバイス一覧に表示される

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

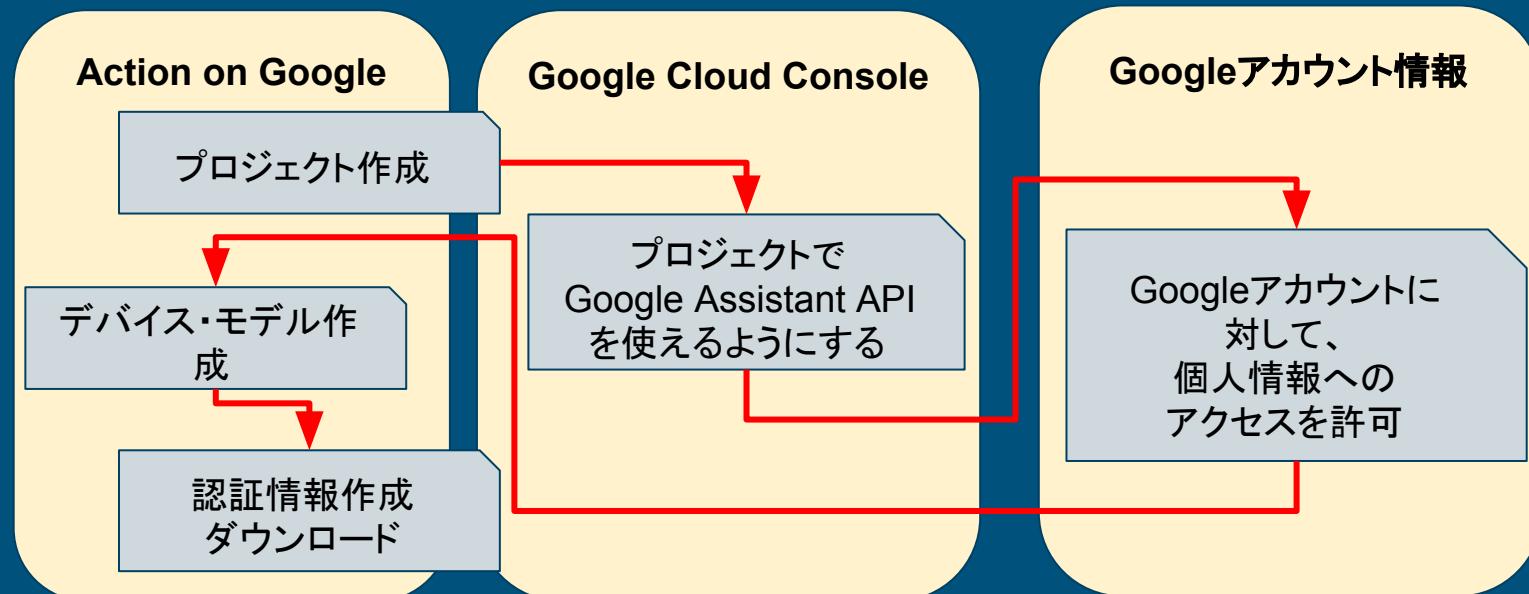
Google提供の
デモプログラム

デモプログラム
を改造！



PCでGoogle Assistantの設定(流れ)

※ 最初から、つまずきやすい作業が続きます！



ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★プロジェクトの作成

1. 「Action on Google」にアクセス＆ログイン

<https://console.actions.google.com/>

The screenshot shows the 'Actions on Google' console. At the top, there's a navigation bar with 'Actions on Google', 'Go to docs', and a user profile icon. Below it, a large heading says 'Welcome to Actions on Google'. A paragraph explains the platform's purpose: 'Actions on Google is the platform for developers to extend the Google Assistant. Join this emerging ecosystem by developing actions to engage users on Google Home, Pixel, and many other surfaces where the Google Assistant will be available.' It includes links to 'Learn more', 'Documentation', 'Sample code', 'API reference', and 'Support'. On the left, a sidebar titled 'Your projects with an Assistant app' shows a list of existing projects: 'myGoogleAssistant1' (YT1) and '+ Add/import project'. A large red arrow points from this section towards the 'CREATE PROJECT' button in the 'Add project' dialog.

The screenshot shows a modal dialog box titled 'Add project'. It has fields for 'Project name' (containing 'FabLabKannai') and 'Country/region' (set to 'Japan'). There are 'CANCEL' and 'CREATE PROJECT' buttons at the bottom. A yellow arrow points from the 'Add/import project' button on the left screen towards this dialog.

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★Google Assistant APIを有効にする

「Google cloud console」にアクセスして、
「Google Assistant API」を「有効にする」

<https://console.developers.google.com/apis/api/embeddedassistant.googleapis.com/overview?authuser=2>



ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★アクティビティ管理

「Googleアカウント情報」の「アクティビティ管理」に
アクセスして、個人情報へのアクセスを許可する

<https://myaccount.google.com/activitycontrols?authuser=2>

許可するアクティビティ(図は省略)

- [ウェブとアプリのアクティビティ]
「Chromeの閲覧履歴と...」をチェックする
- [端末情報]
- [音声アクティビティ]

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

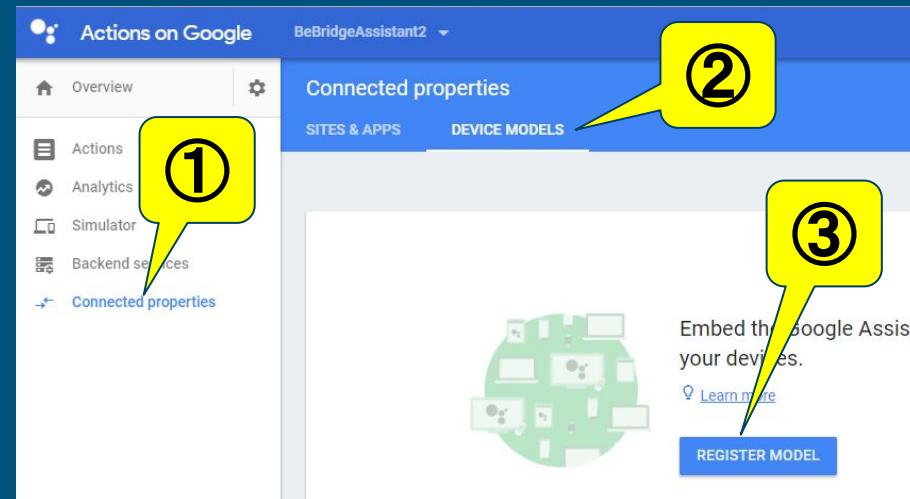
Google提供の
デモプログラム

デモプログラム
を改造！



★デバイスモデルの登録(1)

1. 「Action on Google」にアクセス
<https://console.actions.google.com/?authuser=2>
2. 先ほど作成したプロジェクトを選択
3. 図の順番にクリックして、
「Register Model」
の画面を出す。



ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★デバイスモデルの登録(2)

以下の情報を入力

- プロダクト名
- メーカー名
- デバイスタイプ
- デバイスモデルID

デバイス モデルID
(変更可能、だいぶ後で使う)
タイプしやすい名前に変更して
メモして下さい！

The screenshot shows the 'Register model' dialog box. It has two tabs at the top: 'Create model' (selected) and 'Download credentials file'. Below the tabs are four input fields: 'Product name' (My Awesome Product), 'Manufacturer name' (My Awesome Company), 'Device type' (a dropdown menu showing 'Please select the device type'), and 'Device Model id' (a text input field containing 'my-awesome-product-id'). At the bottom right are 'CANCEL' and 'REGISTER MODEL' buttons.

プロダクトID
(スマホアプリに表示される)

メーカー名
(自分の名前)

デバイスタイプ
「Light」を選択

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★デバイスモデルの登録(3)

「credentials.json」を
一旦デスクトップに保存

Register model

1. Create model 2. Download credentials file 3. Specify traits

★ Keep this file secure, do not upload it to public repositories like GitHub. It's possession grants access to controlling your Assistant SDK project configuration and quotas.

1. [Download credentials.json](#)

2. Place the credentials.json file in folder where you're running the Assistant SDK.
To copy over SSH to a remote device, run the following command from your current computer:

```
scp ~/Downloads/credentials.json <username>@<device-ip-address>:</path/to/assistant-sdk/project>
```

password: password-for-device

NEXT

次へ

ネットで
Googleの設定

microSD
の準備

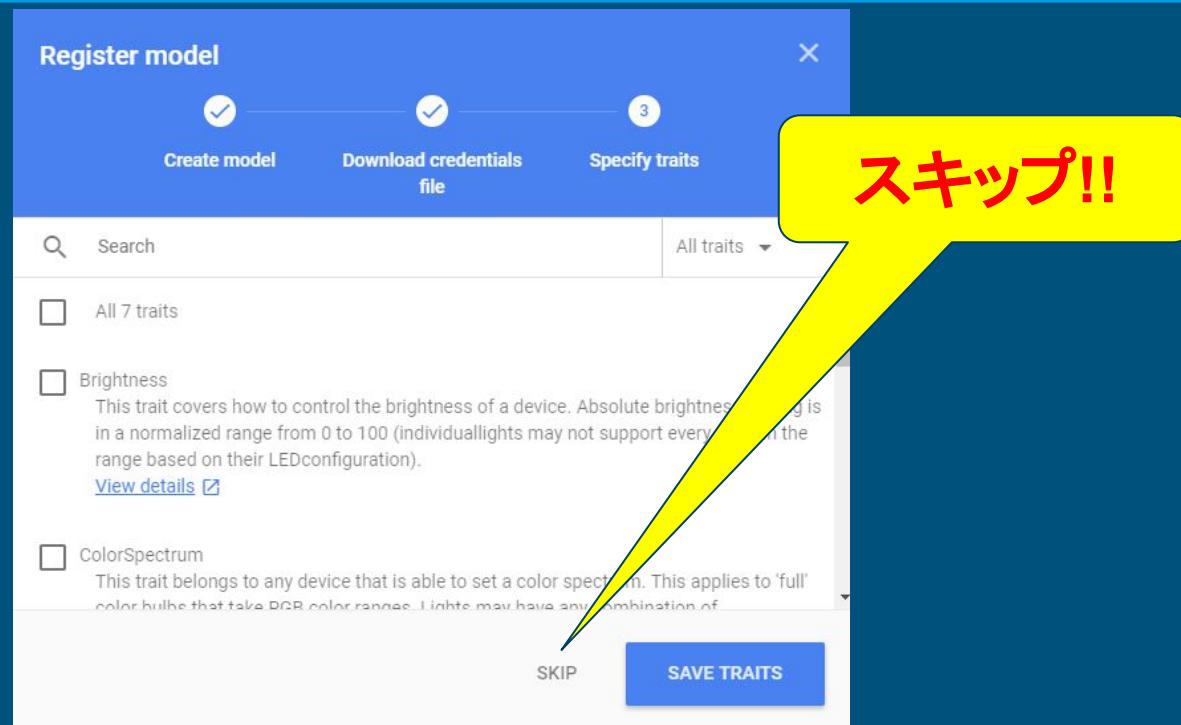
Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★デバイスモデルの登録(4)



ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



PCで microSDの準備

(Raspberry Pi の初期設定: microSD準備)

PCで...

1. Raspbian OS(ディスクイメージ)のダウンロード
<https://www.raspberrypi.org/downloads/raspbian/> 「... WITH DESKTOP」の方
2. Raspbian OSのイメージをmicro SDに書き込む

※ イメージの書き込みは、通常のコピーではダメ！

※「Win32DiskImager」(Windows)、「ApplePi Baker」(Mac)などを使う
書き込み直後、以下の「細工」をする

- a. sshを有効にするために...
「boot」ドライブに「ssh」という名前の空ファイルを作成
- b. Wi-Fi接続するために...
アクセスポイントの設定を書き込んだ「wpa_supplicant.conf」を
「boot」ドライブにコピー

→ micro SDをRaspberry Pi 本体に挿して、起動

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★認証情報をmicroSDにコピー

1. デスクトップに保存されている「credentials.json」を
microSD「boot」にコピー

↓

後で使います

※ ファイル名は、「credentials.json」のまま！

※ ファイル名を変更しない！！

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★Wi-Fiアクセスポイントの設定

1. 以下のサイトにアクセスして、
「wpa_supplicant.conf」をデスクトップに作成
<https://mascii.github.io/wpa-suplicant-conf-tool/>
2. デスクトップに作成した「wpa_supplicant.conf」を
microSD「boot」にコピー

【参考】

<https://qiita.com/mascii/items/a43d71572e1919e56398>

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



この後、
Raspberry Piで作業！

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★組み立て(電源を入れる前に!)

1. ReSpeaker 2-mics Pi HATの取付け

※ 堅いですが、「ゆっくり」と「慎重に」!

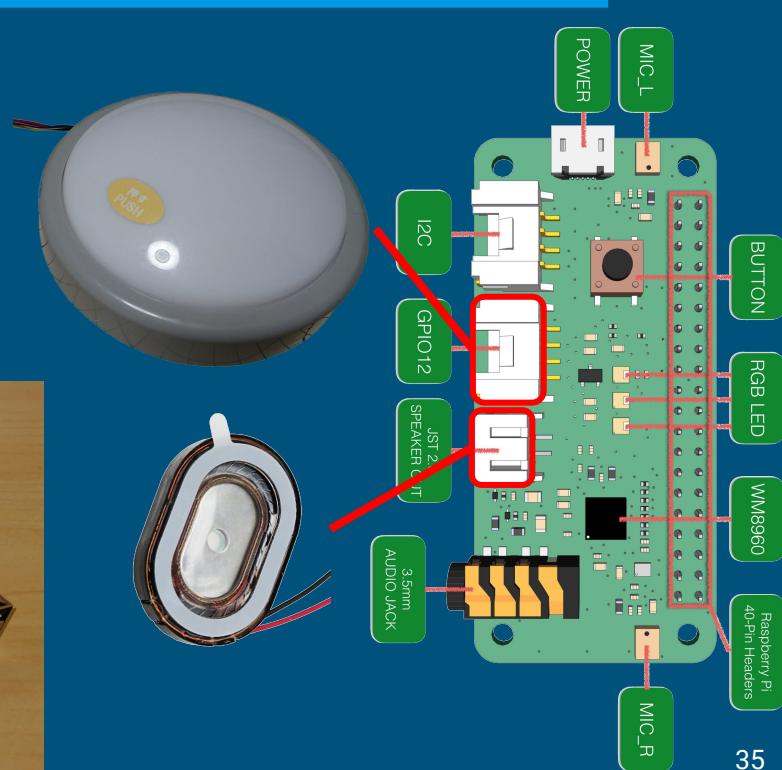
2. スピーカーの取り付け

※ スピーカーに両面テープがついてますが、

※ まだ剥がさない

3. ボタンスイッチの取付け

※ 端子を間違えないように!



ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★Raspberry Piの電源ON！

microSDを挿入・電源ケーブルを接続して、Raspberry Piを起動！

1. いきなり喋りはじめます！！
2. IPアドレスをメモして下さい！

192.168.1.____

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



※ ターミナルソフトについて

Raspberry Piには、ディスプレイもキーボードもマウスもないでの...



- Windowsのターミナルソフト
 - Putty(ranavis版): <http://www.ranvis.com/putty>
 - TeraTerm: <https://ja.osdn.net/projects/ttssh2/>
- Macのターミナルソフト
 - 標準ユーティリティの「ターミナル」の「ssh」コマンド

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



※ コマンドラインは正確に！

コマンドラインにはAI機能はありません！

1文字も間違えずに入力して下さい！

[Enter]キーを押す前に...

- 繰りは正しいですか？
- 適切にスペースを入れてますか？
- コンマ(,)とピリオド(.)を間違えてませんか？
- アンダーバー(_)とマイナス(-)は...

※ 間違えを減らすテクニックはいろいろあります！

※ コマンドライン操作 Tips

- コマンドラインは文字単位で編集できる
[←]、[→]、[Delete]、[Backspace]
- 過去のコマンドラインを呼び出すことができる
[↑]、[↓]、[Ctrl]-[R]
 - [Ctrl]-[R]で以前実行したコマンドラインを検索可能
- キー入力を省略 ... まじめに全部打つ必要はない！！
[Tab]
 - コマンド名やファイル名を途中まで入力し、[Tab]を打つと補完してくれる
- プログラムの中斷
[Ctrl]-[C] ... ターミナル中ではコピペじゃない！

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(Raspberry Pi 初回立ち上げ時の設定)

Raspberry Pi の初回起動後...

1. ssh でPCからRaspberry Piにログイン

```
$ sudo raspi-config
```

2. raspi-configで初期設定(後述)

```
$ sudo shutdown -r now
```

3. 再起動

4. パッケージを最新版にアップグレード(upgrade)

※注意:「full-upgrade」「dist-upgrade」「rpi-update」はしない！

```
$ sudo apt update && sudo apt -y upgrade
```

5. 再起動

```
$ sudo shutdown -r now
```

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(raspi-configでの設定内容)

- [Change User Password] → 「fablab」→ ★後で変更してください
- [Network Options]
 - [Hostname] → 「fablab」→ ★他人とかぶらない名前に後で変更して下さい！
- [Boot Options]
 - [Desktop / CLI] → [Console]
 - [Wait for Network at Boot] → [Yes]
 - [Splash Screen] → [No]
- [Localisation Options]
 - [Change Locale] → 追加:[en_US.UTF-8], [ja_JP.UTF-8] → Default:[C.UTF-8]
 - [Change Timezone] → [Asia]-[Tokyo]
 - [Change Wi-Fi Country] → [JP]
- [Interfacing Options]
 - [SPI] → [Yes]、[I2C] → [Yes]
- [Update]

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(ReSpeaker 2 Mics Pi HATの設定)

参考情報: <https://github.com/respeaker/seeed-voicecard>

1. 関連ファイルをダウンロードし、インストール

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

2. 一旦、シャットダウン

```
$ sudo shutdown -h now
```

3. ReSpeaker基板をRaspberry Piに取付けて、再起動

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★Raspberry Piにログイン

ターミナルソフトで、自分のRaspberry Piにsshでログイン

1. IPアドレス: メモしたもの
2. ポート番号: 22 (通常は指定不要)
3. ユーザー名: pi
4. パスワード: fablab

Macのターミナルの場合

```
$ ssh pi@192.168.1.____
```

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★スピーカー・マイク・LEDのテスト

1. スピーカーのテスト

```
$ speaker-test -t wav  
※ [Ctrl]-[C] で終了
```

2. マイクのテスト a. 録音して再生

```
$ arecord -f cd a.wav  
※ 音声を録音たら [Ctrl]-[C] で終了  
  
$ aplay a.wav
```

3. LEDの点灯テスト

```
$ cd ~/mic_hat  
$ python pixels.py  
※ [Ctrl]-[C] で終了
```

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



デモプログラム実行
の準備！

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(Python3関連パッケージのインストール)

<https://developers.google.com/assistant/sdk/guides/library/python/embed/install-sample>

1. Python3開発環境のインストール

```
$ sudo apt-get update
$ sudo apt-get -y upgrade
$ sudo apt-get -y install python3-dev python3-venv
$ sudo apt-get -y install portaudio19-dev libffi-dev libssl-dev
```

2. 仮想環境の作成(virtualenv): Pythonでいろいろやるときに便利！

```
$ cd
$ python3 -m venv env
$ ~/env/bin/python -m pip install --upgrade pip setuptools wheel
$ source ~/env/bin/activate
```

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(Google Assistant SDK for Python)

<https://developers.google.com/assistant/sdk/guides/library/python/embed/install-sample>

3. Google Assistant SDKのPythonパッケージのインストール

```
(env) $ python -m pip install --upgrade google-assistant-library
(env) $ python -m pip install --upgrade google-assistant-sdk[samples]
(env) $ python -m pip install --upgrade google-auth-oauthlib[tool]
```

(認証情報の設定)

4. 認証情報の設定【慎重に！】

```
(env) $ google-oauthlib-tool
        --scope https://www.googleapis.com/auth/assistant-sdk-prototype
        --save --headless
        --client-secrets /boot/ファイル名.json
```

改行入れず
全て1行で！

「ファイル名」の部分に、PCでmicroSDに保存した認証情報(JSON形式)の
ファイル名を入れて下さい。

5. 以下のように表示され、入力待ち【わかりにくい！...慎重に！】

「Please visit this URL ... https://.../.../... (長い！)」

- URLをPCのブラウザにコピー&ペーストする。
- アカウントを選択 → アクセスを承認
- 表示される認証コード(authorization code)をRaspberry Piの方に入力
(コピー&ペースト)する。

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(デモプログラム実行...でも、まだ...)

6. デモプログラムを一回実行！

- 先ほど登録した「モデルID」を指定
- 実行すると、最初に表示される「デバイスID」を確認！
- 英語での会話！「Hey, Google. What time is it?」など

※ 音量が小さいですが、今は気にしない！

```
(env) $ google-assistant-demo --device_model_id モデルID
device_model_id: モデルID
device_id: デバイスID
:
[Ctrl]-[C]で終了
```

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(デバイスIDの登録)

7. デバイスIDと、クライアントタイプを登録

```
(env) $ googlesamples-assistant-devicetool  
      --project-id プロジェクトID register-device  
      --client-type LIBRARY  
      --model モデルID  
      --device デバイスID
```

改行入れず
全て1行で！

8. 登録内容確認

```
(env) $ googlesamples-assistant-devicetool  
      --project-id プロジェクトID list --device
```

}

9. これで、やっとスマホアプリで言語などの設定ができ、 AIスピーカーとしてまともに使えるようになる！！

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★認証情報設定

1. 認証情報の設定

```
$ setup2-oauth.sh
```

2. 以下のように表示され、**入力待ち【まだ何も入力しない！】**

Please visit this URL ... https://.../...
...(5行ぐらい！)

3. 上記なが〜いURLをPCのブラウザにコピー&ペースト ... 慎重に！
4. 「アカウント」を選択
5. アクセスを「承認」
6. 表示された「認証コード(authorization code)」を
Raspberry Piに入力(コピー&ペースト)する。

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★デバイスIDの登録

7. デモプログラムをとりあえず実行...デバイスIDの確認

- 先ほど登録した「デバイスマodel ID」を指定
- 実行したら、最初に表示される「デバイス ID」を確認！

```
$ setup3-run-demo.sh デバイスマodel ID
device_model_id: モデルID
device_id: デバイスID
:
[Ctrl]-[C]で終了
```

8. デバイスIDを登録

```
$ setup4-register-device.sh デバイスマodel ID デバイスID
```

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★スマホで、確認・言語設定など

9. スマホに「Google Assistant」アプリをインストール
(Androidは不要:ホームボタン長押し)

10. スマホのGoogle Assistantアプリで、デバイスを確認
 - a. 言語を確認(設定)

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



デモプログラム
起動！

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★AIスピーカー起動！

9. デモプログラムの起動！

```
$ setup3-run-demo.sh デバイスマルID  
  
ON_MUTED_CHANGED:  
{'is-muted': False}  
ON_START_FINISHED
```

10. いろいろ話して、試してみましょう！

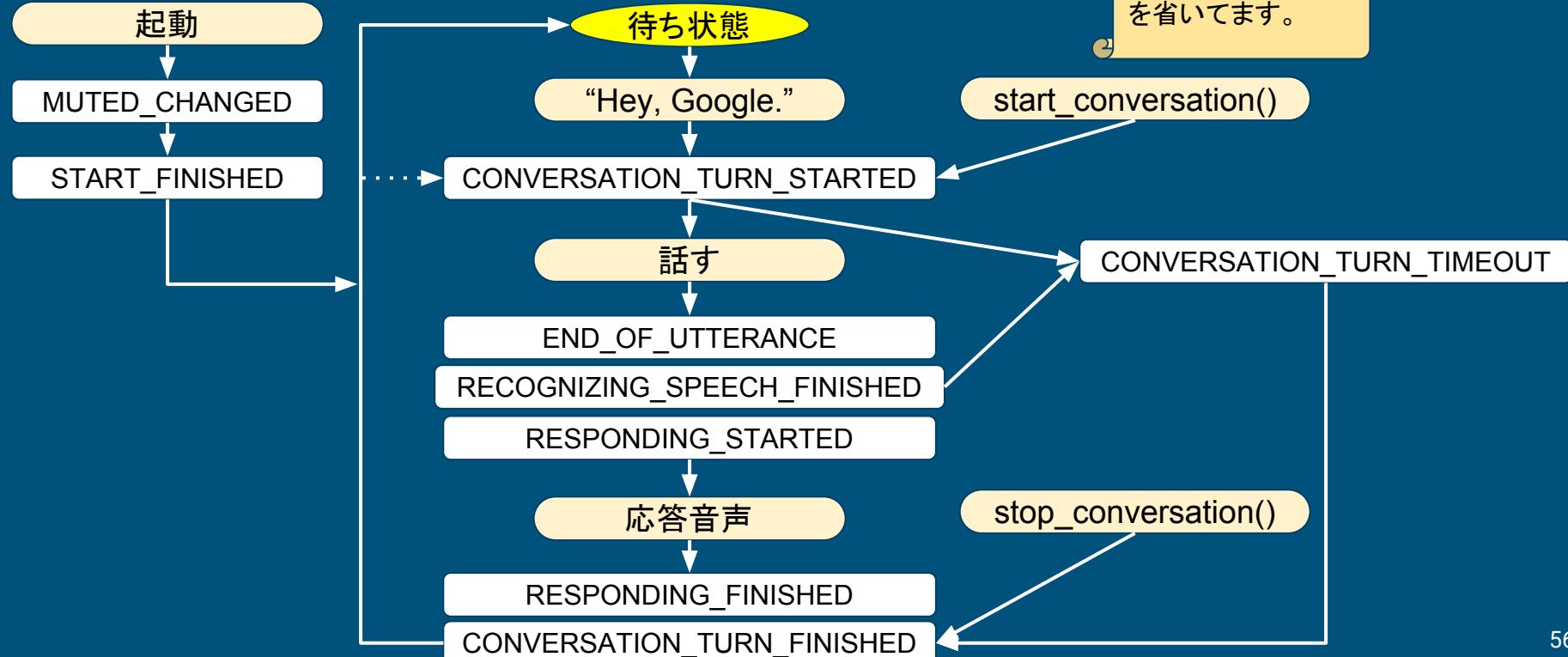
「Hey, Google、音量を95%にして」

「Hey, Google、今何時？」

11. 画面を注意深く観察 → イベント遷移を確認

12. [Ctrl]-[C]で終了

イベント(状態)遷移(概要)



ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



いよいよ改造！

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



ここで使うプッシュ・ライトは...

ダイソーの「プッシュライト」(108円)！！



ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

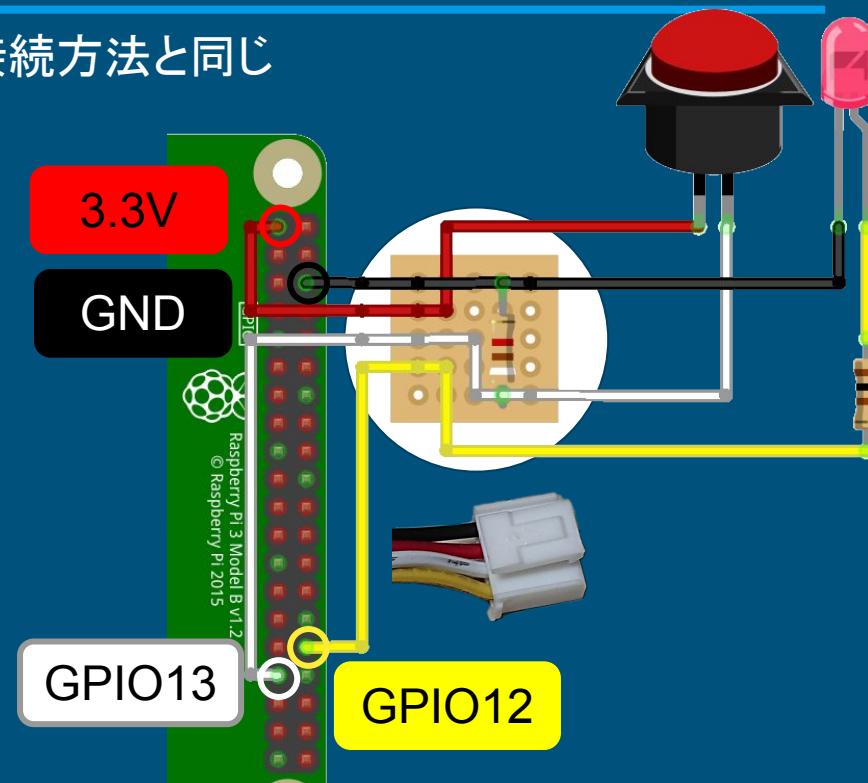
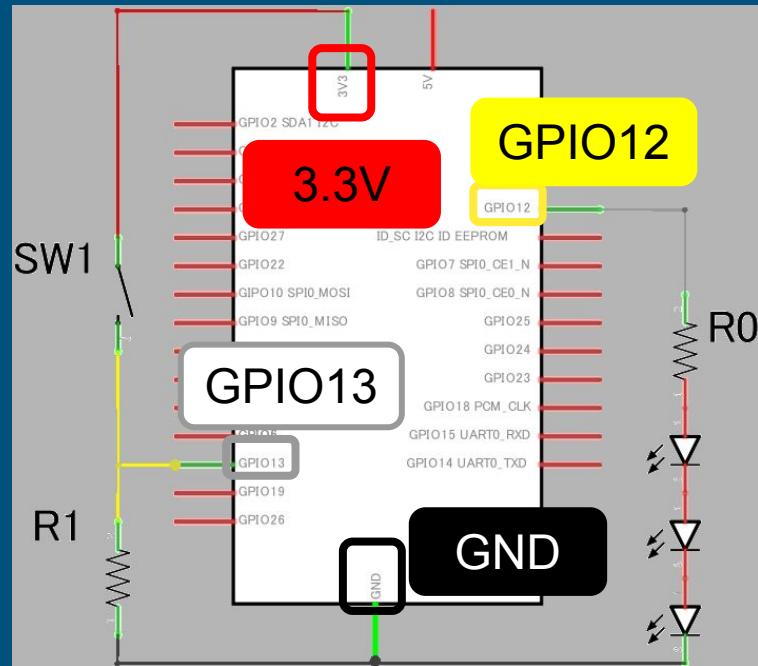
Google提供の
デモプログラム

デモプログラム
を改造！



プッシュ・ライトの改造と接続

基本的なLED点灯とスイッチの接続方法と同じ



ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★手動でGPIOを操作

「gpio」コマンドで、直接GPIOを操作することができます

※ ここではピン番号を「BCM」というナンバリング方法で指定します

- GPIOピンの全ての状態確認
- 入出力モードの設定(-g:BCM)
- 値の書き込み:ライトON/OFF
- 値の読み込み:スイッチの確認

```
$ gpio readall
```

```
$ gpio -g mode 12 output  
$ gpio -g mode 13 input
```

```
$ gpio -g write 12 1  
$ gpio -g write 12 0
```

```
$ gpio -g read 13
```

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(改造の準備と動作確認)

1. SDKのデモプログラムを編集するためにコピー

```
(env) $ cd ~/env/lib/python3.5/site-packages/google/assistant/library  
(env) $ cp __main__.py ~/env/bin/FabLabKannai-GoogleAssistant.py  
(env) $ cd ~/env/bin  
(env) $ chmod +x FabLabKannai-GoogleAssistant.py
```

2. 独自のファイルをダウンロードして編集用にコピー

```
(env) $ cd ~  
(env) $ git clone https://github.com/ytani01/FabLabKannai-GoogleAssistant.git  
(env) $ cd FabLabKannai-GoogleAssistant  
(env) $ cp *.patch ~/env/bin  
(env) $ cp -r sound ~
```

3. コピーしたデモプログラム(改造前)を起動して動作確認

```
(env) $ FabLabKannai-GoogleAssistant.py --device_model_id モデルID
```

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(デモプログラムの改造)

ソースプログラムを改造して、以下の機能を追加

1. 確認音(3種類)、ボード上のイルミネーション
2. 照明ON/OFF ←「照明をつけて」、「照明を消して」
3. ボタン → ホットワードなしで会話をはじめる(push to talk)
4. 連続対話機能: エンドワードまで会話を続ける

```
$ cd ~/env/bin
$ cp FabLabKannai-GoogleAssistant.py FabLabKannai-GoogleAssistant-0.py
$ patch -b FabLabKannai-GoogleAssistant.py < all.patch
$ source ~/env/bin/activate
(env)$ python -m pip install --upgrade rpi.gpio
(env)$ FabLabKannai-GoogleAssistant.py --device_model_id モデルID
```

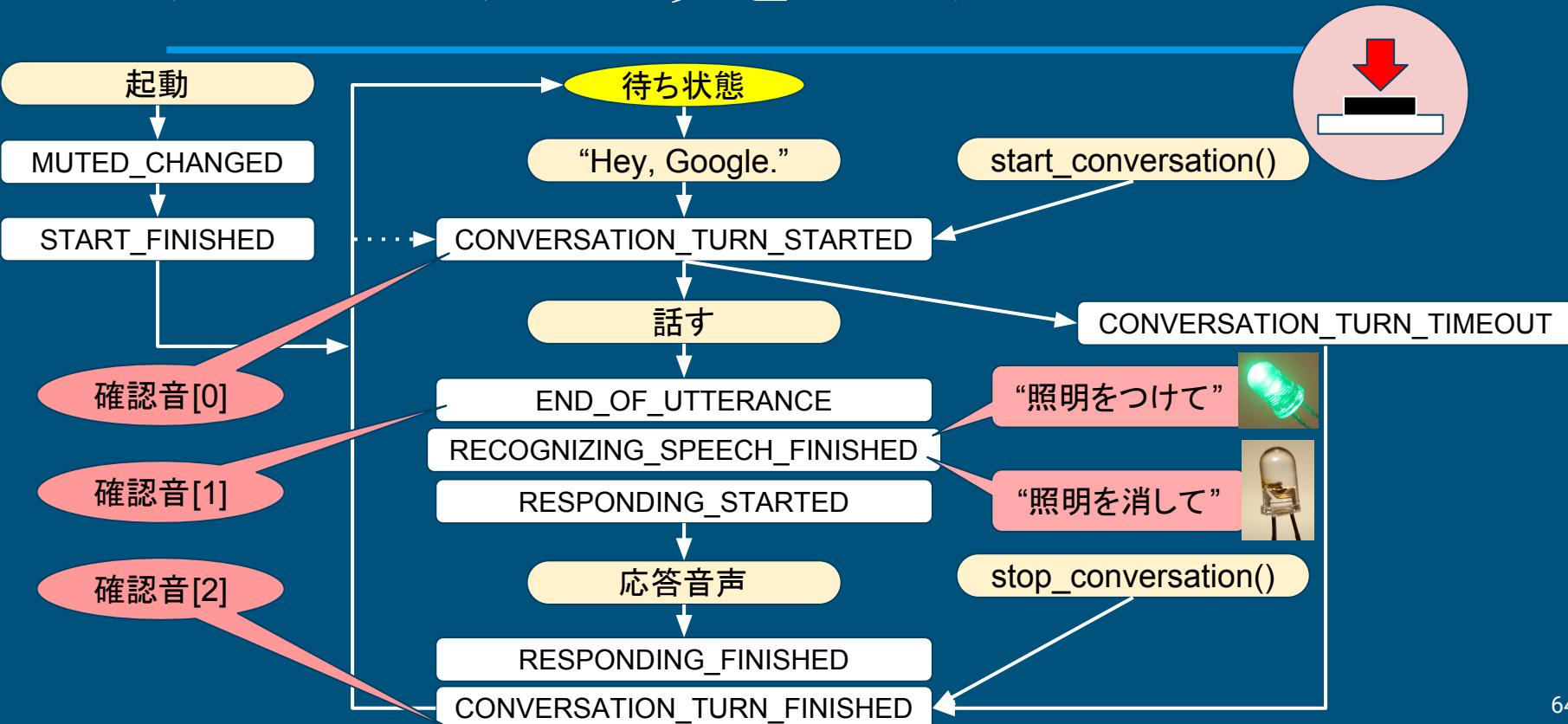
※ ソースファイル

- 全ソースファイルはネット上(github)にも
 - <https://github.com/ytani01/FabLabKannai-GoogleAssistant>
- Raspberry Piの中に既にコピーしてあります。
 - /home/pi/env/bin/...
- 改造部分を見るだけでも、勉強のきっかけや参考になります。

改造後	「FabLabKannai-GoogleAssistant.py」
改造前(Google製)	「FabLabKannai-GoogleAssistant-0.py」
変更部分	「all.patch」

 - Pythonプログラミング入門
 - AIスピーカーの基本動作
 - Raspberry PiでGPIOを使って、回路制御する方法

デモプログラムの改造のポイント



ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★起動の簡略化(シェルスクリプト)

1. 起動スクリプト:「FabLabKannaiAssistant.sh」

仮想環境をactivateしたり、モデルIDを毎回指定する必要がなくなる

```
$ cd ~/bin  
$ cp FabLabKannai-GoogleAssistant.sh.sample FabLabKannai-GoogleAssistant.sh  
$ nano FabLabKannai-GoogleAssistant.sh
```

```
#!/bin/sh  
:  
DEVICE_MODEL_ID=モデルID  
:
```

ここだけ書き換える

```
$ cat FabLabKannai-GoogleAssistant # 内容確認(間違ってたら、編集する)
```

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



★改造したプログラムの起動

1. 改造したプログラムの起動

```
$ FabLabKannai-GoogleAssistant.sh
```

2. 改造したプログラムで遊ぶ!

- a. 確認音(3種類)
- b. 基板上のイルミネーション
- c. 照明ON/OFF:「照明をつけて」、「照明を消して」
- d. ボタン → ホットワードなしで会話をはじめる(push to talk)
- e. 連続対話機能:エンドワードまで会話を続ける

3. [Ctrl]-[C]で終了

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



自動起動

再起動すると自動的に起動するように設定してあります！

1. 「crontab」というLinuxの機能を利用している

【確認方法】

```
$ crontab -l
```

2. 実際に自動起動するプログラムは、「boot.sh」

【確認方法】

```
$ cat ~/bin/boot.sh
```

3. 「boot.sh」の中で、「FabLabKannai-GoogleAssistant.sh」を起動

- a. 起動時にIPアドレスをしゃべる設定もここ！
- b. 他にもおまけ機能が…！？

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



完成！！

まとめ

1. Raspberry Pi 3と
ReSpeaker 2-Mics Pi HAT基板を組みあわせ、
Google Assistant SDKをインストールして、
AIスピーカーを作りました。
2. ダイソーのプッシュライトを改造して接続し、
照明の機能とプッシュボタンの機能を追加しました。
 - a. ホームゲートウェイやメーカーなどのサービスに登録する必要はありません。
 - b. 基本的な「Lチカ」とスイッチの回路を GPIOで操作しているため、
さらに改造するきっかけになります。
3. ソフトウェアを改造して、独自機能「連続対話」などを実現しました。
 - a. 全てPythonプログラミングの入門レベルの改造なので、
勉強のきっかけとして良い題材だと思います。



Appendix

(付録・補足)

【演習1】「ライトをつけて」

ソースプログラムを改造して、違う言葉で照明をON/OFFできるようにしてみて下さい。

例1:「ライトをつけて」「ライトを消して」

例2:「照明オン」「照明オフ」

※ 引っかけというか、罠が ! ?

ネットで
Googleの設定

microSD
の準備

Raspberry Pi
の初期設定

Google提供の
デモプログラム

デモプログラム
を改造！



(IPアドレスをしやべる設定)

1. 独自ファイルのダウンロード

```
$ cd  
$ git clone https://github.com/ytani01/speak.git
```

2. Open JTALKをインストール

```
$ sudo apt update && sudo apt -y upgrade  
$ sudo apt -y install open-jtalk open-jtalk-mecab-naist-jdic \  
> hts-voice-nitech-jp-atr503-m001
```

3. スクリプトのコピー

```
$ mkdir ~/bin  
$ cd ~/speak  
$ cp *.py *.sh ~/bin
```

4. crontabを設定し、自動起動するようにする(詳細省略)

```
$ crontab -l # 内容の確認
```

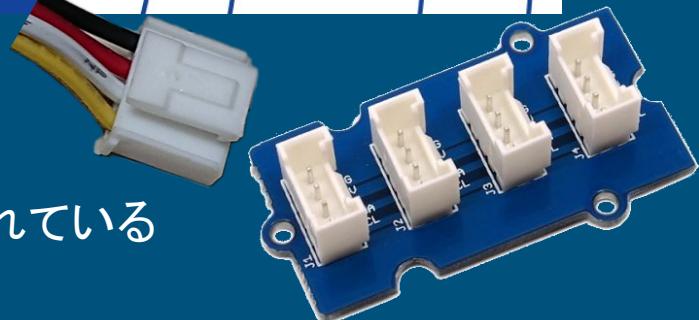
(参考) Groveシステムについて

- Seeed Studio社が規定し、開発している

http://wiki.seeedstudio.com/Grove_System/



- コネクタと配線が統一されている
 - 半田付け不要
 - 配線の間違えがない
- 様々なセンサー類・パーツ類が販売されている
 - ただし、パーツをバラで買うよりお高め



Wi-Fiの設定 (Raspberry Piでの方法)

1. 設定ファイルとコマンド

設定ファイル: /etc/wpa_supplicant/wpa_supplicant.conf

パスワード設定コマンド: wpa_passphrase

```
$ cd /etc/wpa_supplicant
$ wpa_passphrase 'SSID' 'password' | sudo tee -a wpa_supplicant.conf
$ cat wpa_supplicant.conf ←(確認)
$ shutdown -r now           ←(再起動)
```

2. 設定ファイルを修正する場合 ... 上記操作で間違えたときなど

```
$ cd /etc/wpa_supplicant
$ sudo nano wpa_supplicant.conf
```

外部サービスとの連携について(ザックリと...)

外部のサービスと連携することにより、**対応されている範囲**で、独自の機能拡張や外部機器連携を行うことができます。

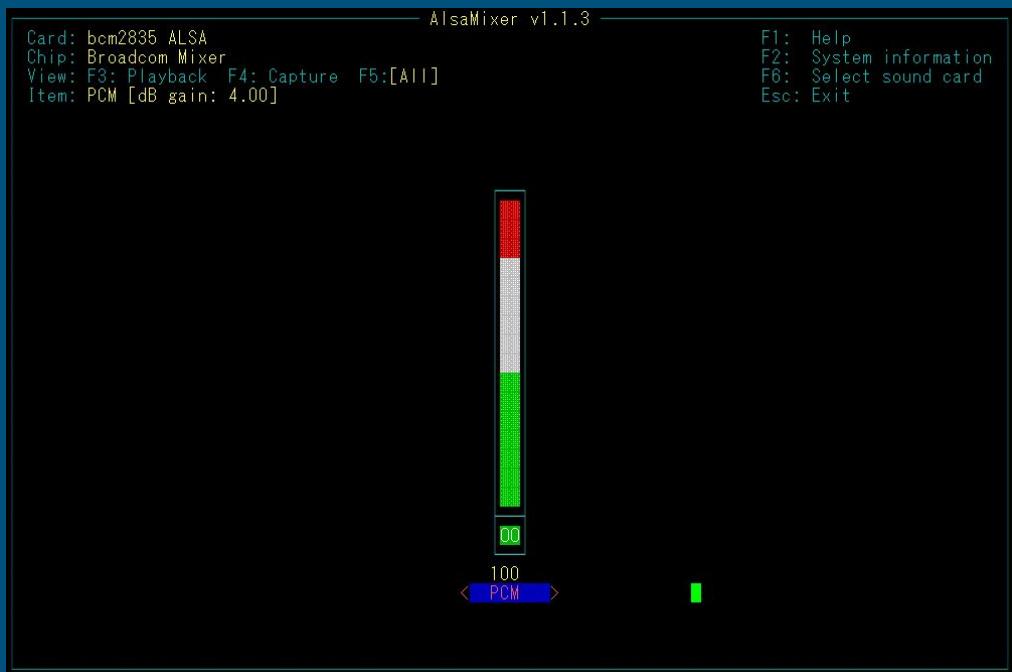
- IFTTT: サービス連携
 - SNS、家電制御、ネットサービス等多くのものと手軽に連携可能
 - 「If This Then That」という単純な考え方で簡単に設定できる
 - サービス毎に出来ることが限られている。課金される場合もある。
- Action on Google: 機能拡張
 - Google Assistantに独自機能および対話機能を追加できる。
 - Amazon Echo (Alexa)の「スキル」に似ている(?)
- Dialogflow: 自動対話
 - Google Assistantで機能を追加する際の自動対話機能を作れる。

マイク・スピーカーの設定

1. スピーカーボリューム、マイク感度の設定

```
$ alsamixer
```

- [F6] カード選択
- [F3] スピーカー
- [F4] マイク
- [Tab] View切換
- [↑][↓] Up, Down
- [Esc] 終了



パッチファイルについて

オープンソース・ソフトウェアの開発などで、ソースコードを書き換えたときは、
「パッチファイル」で他の開発者に変更点を通知することができます。

- ソースコードの **変更点が分かり易い** (テキスト形式)
- patchコマンドでソースコードを **自動修正** (エディタ不要) ←「パッチを当てる」といいます。

修正前

```
abc  
def  
ghi
```

修正後

```
abc  
12345  
67890  
ghi
```

パッチファイル

```
-- old ...
+++ new ...
@@ -1,3 +1,4 @@
 abc
-def
+12345
+67890
 ghi
```