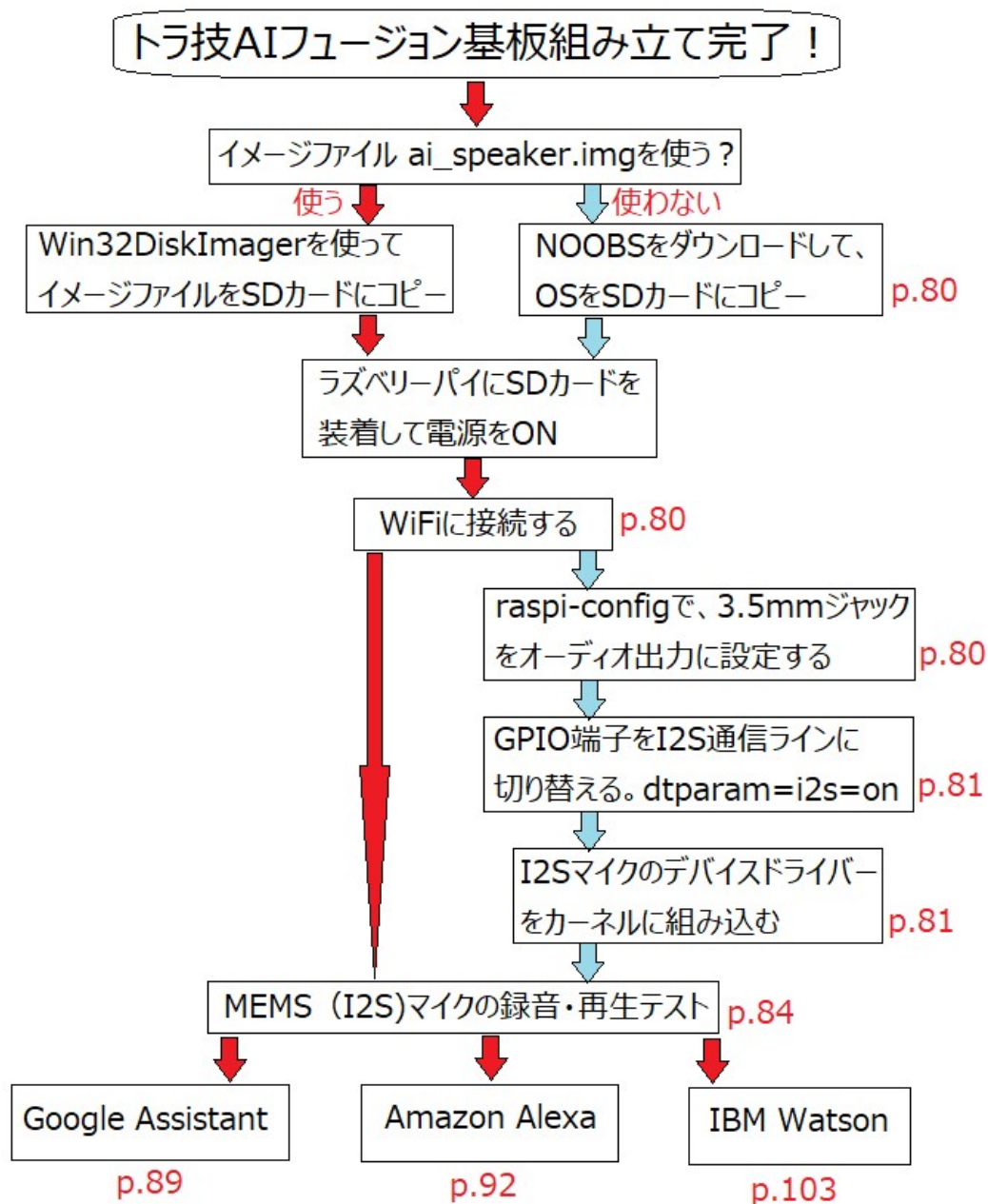


ラズベリーパイと I2S マイクで作る AI スピーカ

付録基板の組み立てが終わったら、次の手順でソフトウェアをインストールしてください。
ページ番号は、トランジスタ技術 2018 年 3 月号の該当ページです。



詳しくは、トランジスタ技術 2018 年 3 月号 p.64～p.121 の記事を参考にしてください。

いくつか修正点があります。以下の手順に進めてください。

●手順⑥ Python3 の仮想環境にインストールする

次のように入力して、ラズベリー・パイに Python3 をインストールし、仮想環境(env)を構築します。

```
sudo apt-get update . . . ①
sudo apt-get install python3-dev python3-venv . . . ②
python3 -m venv env . . . ③
env/bin/python -m pip install --upgrade pip setuptools wheel . . . ④
source env/bin/activate . . . ⑤
```

●手順⑦ Google Assistant Library を取得する

Google Assistant Library は、音声による問いかけを録音したり、Google Assistant からの答えを再生したり、ホットワード”Ok Google”と”Hey Google”を検出したりするライブラリです。次の pip コマンドで、ラズベリー・パイにインストールします。

```
sudo apt-get install portaudio19-dev libffi-dev libssl-dev . . . ⑥
python -m pip install --upgrade google-assistant-library . . . ⑦
python -m pip install --upgrade google-assistant-sdk[samples] . . . ⑧
```

コマンドはすべて仮想環境下で入力してください。プロンプト冒頭に(env)が表示された状態です。

●クライアント ID 認証ツールを走らせる

次のように入力して、ラズベリー・パイに認証ツールをインストールします。

```
python -m pip install --upgrade google-auth-oauthlib[tool] . . . ⑨
```

次のように入力して、認証ツールを実行します。ラズベリー・パイにつないだキーボードを使うときは、最後の--headless は不要です。

```
google-oauthlib-tool --client-secrets $(find $HOME -name client_secret_*.apps.googleusercontent.com.json) --scope https://www.googleapis.com/auth/assistant-sdk-prototype --save --headless . . . ⑩
```

認証ツールが実行されると、次のように表示されます。

Please go to this [URL:https://...](https://...)

Web ブラウザでこの URL にアクセスして認証が成功すると、コード”4/XXXX”が返ってきます。このコードを次の行の後に貼り付けます。

Enter the authorization code:

次にデバイスモデルを登録します。

```
googlesamples-assistant-devicetool register-model --manufacturer "Assistant SDK developer"
--product-name "Assistant SDK light" --type LIGHT --model cq-aispeaker . . . ⑪
登録できたかどうかは次のように確認してください。
googlesamples-assistant-devicetool get --model cq-aispeaker . . . ⑫
```

●手順⑨ Google Assistant お試し版を起動

Google Assistant Library が無事インストールされると、図 12 のような表示が出ます。(env) プロンプトにて、次のように入力すると会話ソフトウェアが動き出します。

```
google-assistant-demo --device_model_id cq-aispeaker
```

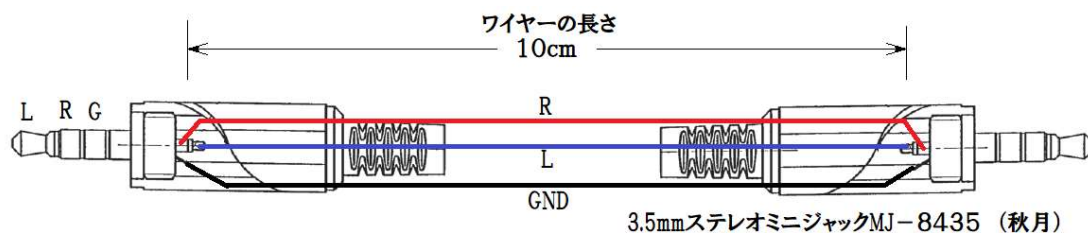
・・・⑬

“Ok Google”，または“Hey Google”に続けて、英語でしゃべりかけてみてください。図 13 に示すのは、会話中に画面に流れるメッセージです。

2 回目からは、⑤で Python3 の仮想環境(env)に入って⑬を実行するだけです。すんなり仮想環境に入れない場合は、③④⑤⑬と実行してみてください。仮想環境から抜けるコマンドは、
deactivate
です。

●オーディオケーブルの作り方

本誌 p.82 に触れていますが、ラズベリーパイの 3.5mm オーディオ出力端子のピン順は、一般の AV ケーブルとは異なります。市販のステレオ用オーディオケーブル（両端が 3.5mm プラグのもの）が使えます。ただ、ビデオ出力をショートしてしまうため、ノイズの原因になる可能性があるかも知れません。ビデオ出力をショートせず、線長も短いケーブルを自作するとスマートに接続できます。下図のように接続してください。ワイヤーはノイズの点では、2 芯シールド線が好ましいですが、ワイヤー長が短いので AWG 28 以下のビニール電線でも大丈夫です。



●電源ノイズ対策

ラズベリーパイの 3.3V 電源には、ときどき大きな雑音がかかることがあります。これが原因で、スピーカーから聞き苦しい雑音が出る場合があります。この場合、スピーカを基板から離してください。オーディオ回路は 5V で動作しているので、3.3V ラインから直接影響を受けているわけではありませんが、3.3V 電源ラインから飛ぶノイズをスピーカーケーブルやスピーカー自体が拾い、フィードバックがかかるようです。3.3V 電源回路の C2 (10 μ F) に並列に 100 μ F のコンデンサを入れても改善できます。

●マイクロ SD カードについて

本誌 p.80 ●OS をインストールして初期設定する (1)項を次のように訂正します。

(1)・・・マイクロ SD カード (8G バイト以上) → 16G バイト以上

8 GB の場合、p.82 のコンパイル失敗、LIRC 動作不良などが発生することがあります。

●カーネルのバージョンをダウングレードする方法

本誌 p.82 のコンパイル (make) ができないときは、カーネルのバージョンをダウングレードしてください。

まず、下記コマンドでカーネル・ヘッダを取得してください。

```
sudo apt-get install raspberrypi-kernel-headers
```

ファイルマネージャーで、lib/modules/の下、4.9.77-v7+などのカーネルのバージョンに対応したディレクトリを探してください。ここに、build というディレクトリが無い場合は、このバージョンでは本文の my_loader.c のコンパイル(make)に失敗します。いくつかのカーネルのバージョンのディレクトリを見て、build ディレクトリがあるものを探します。なければさらにダウングレードが必要ですが、ここでは必要なバージョンが見つかったとします。以下の手順でカーネルをこのバージョンにダウングレードします。

```
uname -r
```

で現在のバージョンを控えておきます。

古いバージョンを取得するには、ブラウザで次の URL にアクセスします。

<https://github.com/Hexxeh/rpi-firmware/commits/master>

該当するバージョンのハッシュ値を、Copy the full SHA ボタンを押してコピーします。

次に、

```
sudo rpi-update xxxxx
```

の xxxxx の部分に、上記ハッシュ値を貼り付けて実行してください。

ダウンロードが途切れて失敗することもあります。あきらめずに再試行してみてください。最後に、上記ハッシュ値が表示されたら成功です。リブートしてバージョンをチェックしてみてください。

コンパイルが出来たら、以降、カーネルのバージョンアップは可能です。

●リモコン受光素子出力信号のレベル不足対策

リモコン受光回路の電源電圧は 5V です。一方、ラズベリーパイの GPIO は 3.3V です。このため、OUT 端子と GND 間に R12(82kΩ)を入れてポートの電圧が 3.3V を越えることを防止しています。この抵抗値(82kΩ)は、コーデンシ PIC-A18143TC5 に対応した値です。他の品番の受光素子を使用する場合は、GPIO 端子が 3.3V となる抵抗値に変更してください。

リモコン受光素子の出力が 3.3V を大きく下回り、リモコン受光信号が得られない場合は、以下の対策をしてください。この対策は、コーデンシ PIC-A18143TC5 を含むすべてのリモコン受光素子についても、信号レベルを保証する上で有効です。

本誌第 15 話 p.96 右マイナス 1 行目～p.97 左 2 行目

```
dtoverlay=lirc-rpi
```

```
dtparam=gpio_in_pin=4
```

```
dtparam=gpio_in_pull=up
```

・・・この行を追加する。(図 4 も同様です。)

```
dtparam=gpio_out_pin=17
```

●その他の修正

p.85 右マイナス 8 行目: `--format=S16LE` → `S16_LE`

p.85 右マイナス 5 行目: `--format=S16LE` → `S16_LE`

p.89 右 6 行目: `-file-type=raw` → `--file-type`

p.89 右 8 行目: `aplay` 以下を次のように修正してください。

`aplay --format=S16_LE rate=16000 out.raw`

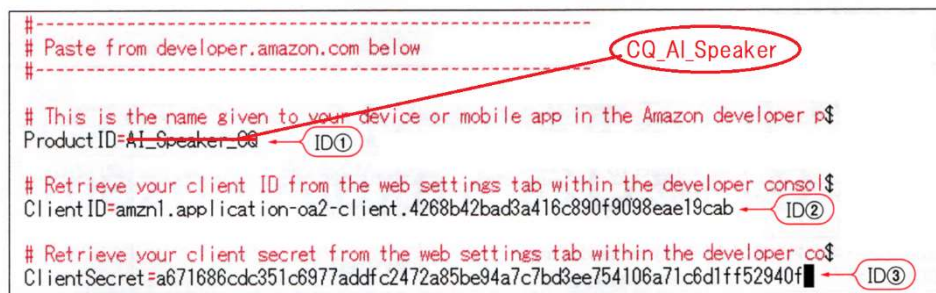
p.91 左 1 2 行目: (第 12 話参照) → 第 11 話

p.96 右 2 行目: `gpio_readall` → `gpio readall`

p.102 左 6 行目: `/Adafruit[ret]` → `[ret]`をトル

p.94 図 7 の ID①を下图のように変更してください。

図7 automated_install.sh をエディタで開いて、図4でメモした①プロダクトID、②クライアントID、③クライアント・シークレットIDをペーストする



p.67 図2 本誌の付録基板「トラ技 AI センサ・フュージョン」の回路図

音声センサ部 IC1 (L チャンネル) の 8 ピンは、下图のように GND 接続が正しいです。基板のパターンは下图の通りになっています。

