セミナー: IoT のための人工知能&コンピュータ活用術 revised: 2018.4.13

## 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。

## 第13話 [クラウド AI と会話する①] Google Assistant とおしゃべり (p.89)

いくつか修正点があります。以下の手順で進めてください。

●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	$\cdot$ $\cdot$ $\cdot$ 1
sudo apt-get install python3-dev python3-venv	· · · 2
python3 -m venv env	· · · ③
env/bin/python -m pip installupgrade 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 (3/28 修正)
python -m pip installupgrade google-assistant-library	$\cdot$ $\cdot$ $\cdot$ $(7)$
python -m pip installupgrade 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]  $\cdot \cdot \cdot 9$ 

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、--headless は不要です。

google-oauthlib-tool --scope <u>https://www.googleapis.com/auth/assistant-sdk-prototype</u> --save -headless --client-secrets \$(find \$HOME -name client\_secret\_\*.apps.googleusercontent.com.json) ····⑩ (3/28 修正)

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://...

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  $\cdot \cdot \cdot \textcircled{D}$ 

●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 · · · ①

"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 AWG28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乗ることがあります。これが原因で、 スピーカーから聞き苦しい雑音が出ることがあります。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 μF) に並列に 100 μF 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  $\bigoplus$ 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)  $\rightarrow$  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。

●I2S ドライバのコンパイル(make)ができないときの対処(4/19 タイトルと文を変更) 本誌 p.82 のコンパイル (make) ができないときの対処です。

- ① SD カードを 16GB にする。
- p.81「②Raspbian OS のカーネルを最新版にアップデート」からやりなおす。
  これでも解決しない場合は、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/の下の、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が、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 5 V 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(82k $\Omega$ )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(82k $\Omega$ )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

●Node-RED での録音音量調整(3/31 追加)

第5話:Watson リモコン・スピーカの製作②会話機能を搭載において、このままでは録音音 量が小さいので、録音ボリュームを追加してください。p.117 左マイナス 10 行目及び図5の Append 欄のコマンド引数を次のように入力します。

-d 5 -D dmic\_sv -c2 -r 16000 -f S16\_LE audio\_in.wav

このとき、下記を確認してください。

- ① p.85 リスト1を、.asoundrc ファイルに追加して、dmic\_sv を有効にする。
- ② .asoundrc を Node-RED の実行ディレクトリに置く。Node-RED の実行ディレクトリは、通常はホームディレクトリですが、p.105 左マイナス 4 行目のように、管理者権限 sudo を付けて実行した場合は、/root となります。従って、次のようにコピーしてください。 sudo cp ~/.asoundrc /root/

録音ボリュームの設定は、p.85を参照してください。

●その他の修正

p.85 右マイナス 8 行目: --format=S16LE  $\rightarrow$ S16\_LE p.85 右マイナス 5 行目: --format=S16LE  $\rightarrow$ S16\_LE p.89 右 6 行目: -file-type=raw  $\rightarrow$  --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)  $\rightarrow$  第 11 話 p.96 右 2 行目: gpio\_readall  $\rightarrow$  gpio readall p.102 左 6 行目: /Adafruit[ret]  $\rightarrow$ [ret]をトル p.94 図 7 の ID①を下図のように変更してください。



p.83 図 5 プロンプト pi@raspberrypi:\$ →pi@raspberrypi:~/rpi-i2s-audio\$

p.83 文末の reboot まですべて rpi-i2s-audio ディレクトリで作業してください。(4/13 追加)

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 チャネル)の8ピンは、下図のように GND 接続が正しいです。基板のパ ターンは下図の通りになっています。





