

Google アシスタント会話ロボット

「自然会話ロボットを作ろう」を発売以来、多くの読者にご愛読していただき、誠にありがとうございました。しかし、残念ながら本書が採用した NTT ドコモの「自然対話 API」は、2019 年 6 月 28 日以後に利用できなくなります。この問題の解決には、会話ロボットの対話 API を「Google アシスタント」に切り替えるサポート情報を提供します。

1. ロボット本体の用意

ロボットの本体は、Arduino の制御ソフトウェア（スケッチ）を含めて、従来「自然会話ロボットを作ろう」のまま、変更する必要はありません。従来の Raspberry Pi のハードウェアは、そのまま使えますが、ソフトウェアは Google アシスタントに対応するプログラムに書き換えるので、次の節で説明します。

「自然会話ロボットを作ろう」を参照して、下表「○」の章節内容を実装して、ロボット本体(手づくりロボットまたは Rapiro)を用意してください。「×」の内容は、今度使わないので、実装しなくてもよいです。特にロボット制御のスケッチ「AiRobot」は、Arduino ボードまたは Rapiro の制御基板に書き込みしてください。

「自然会話ロボットを作ろう」の実装

章節	タートル	実装
第 1 章 01-03 節	OS をインストールする	×実装しない
第 1 章 01-04 節	Raspberry Pi を使う	×実装しない
第 2 章	ラズパイで人口知能会話を実現しよう	×実装しない
第 3 章	自作の自然会話ロボットを作ろう	○手作りロボットに実装。
第 4 章	Rapiro で自然会話ロボットを作ろう	○Rapiro に実装。

2. Google アシスタントの導入

Google アシスタントの詳細について、解説書「スマートスピーカーを作ろう!（秀和システム 2018/9）」を参照してください。「スマートスピーカーを作ろう!」には、「AIY Voice Kit」と「自作スマートスピーカー」の 2 種類のハードウェアを紹介しましたが、今度の会話ロボットには、「自作スマートスピーカー」のハードウェアに採用します。

2.1 Google アシスタントのソフトウェア

「スマートスピーカーを作ろう!」を参照して、下表「○」の章節内容を実装して、Google アシスタントのソフトウェアを用意してください。「×」の内容は、今度使わないので、実装しなくてもよいです。

「スマートスピーカーを作ろう!」 Google アシスタントの実装

章 節	タ ー ト ル	実装必要性
第 1 章 03-03 節	ハードウェア	×実装しない
第 2 章 02-03 節	イメージの書き込み	○実装
第 2 章 02-04 節	Raspberry Pi の初期設定	○実装
第 3 章 03-01 節	ソフトウェア	○実装
第 2 章 03-05 節	マイクロフォンとスピーカー	○実装
第 2 章 03-06 節	Google API の取得	○実装。
第 3 章 03-04 節	ボタンと LED	×実装しない
第 3 章 03-07-01	アシスタント gRPC の認証と登録	○実装
第 3 章 03-07-02	ボタン押下での会話	×実装しない
第 3 章 03-08 節	アシスタントライブラリの会話	×実装しない
第 4 章	家電の音声制御	×実装しない
第 5 章	スマートスピーカーを完成させよう	×実装しない

2.2 注意事項：「apt-get upgrade」実行しないこと

注意事項として、「**apt-get upgrade**」というコマンドを必ず実行しないでください。「**apt-get upgrade**」はインストール可能なパッケージの「一覧」を更新するコマンドです。いったん「**apt-get upgrade**」を実行すると、今度使っている Google の「aiy」ライブラリ 1.0 は、バージョン 1.1 にアップデートして、会話ロボットのプログラムは動作できなくなります。

2.3 音声デバイスの動作確認とトラブル対策

「スマートスピーカーを作ろう!」03-05-01 項「音声デバイスの管理」を参照して、マイクロフォンとスピーカーの動作の確認に、ターミナルに次のようなコマンドを入力して、スクリプト `test_audio.sh` を実行してください。

```
pi@raspberrypi - $ sh MY-spaker/test/test_audio.sh↓
```

録音した音声が正常に再生されれば、デフォルト音声デバイスの設定には、特に問題ないことを示します。

通常では、「スマートスピーカーを作ろう!」03-05-02 項の「**オンボード音声回路の有効化**」を実施すれば、音声デバイスが正常に動作できますが、プログラミング作業中、動作したマイクロフォンはまだ動作しないことがあります。この場合、次のように確認してください。

ターミナルに USB デバイス検索コマンド “`lsusb`” を入力して、マイクロフォンが USB デバイスとして正しく認識されているか確認してください。

```
pi@raspberrypi - $ lsusb↓
.....
Bus 001 Device 007: ID 0d8c:013c C-Media Electronics, Inc. CM108 Audio Controller
.....
```

USB 接続に問題ない場合、ターミナルに次のようなコマンドを入力して、PulseAudio ユーティリティ `pactl` で、「デフォルトソース」音声デバイス設定を確認してください。

```
pi@raspberrypi $ pactl info↓
.....
デフォルトソース: alsa_input.usb-C-Media_Electronics_Inc._USB_PnP_Sound_Device-00.
analog-mono
.....
```

正常の場合、「デフォルトソース」を USB マイクロフォンに設定されているはずですが、Raspberry Pi が勝手に別のデバイスに切り替えてしまうことがあります。この場合には、スクリプト `audio.sh` を再び実行してください、

```
pi@raspberrypi - $ sh MY-spaker/setup/audio.sh↓
```

通常では、起動後にトラブルを解決できますが、どうしてもトラブルを解消できない場合、イメージの再ロードしかできません。

3. Google アシスタント会話ロボット

3.1 サンプルコード

Google アシスタントを使い、会話ロボットの実現方法の情報とサンプルコードを提供します。秀和システムのサポートページへアクセスして、サンプルファイル「`robot.zip`」をダウンロードすることができます。下表にはサンプルファイルの一覧を示します。

サンプルファイル

展開先	コピー先	ファイル	用途
robot/setup	-	samples.sh	サンプルファイル配置
robot/src	AIY-projects-python/src	robot.py	会話デモプログラム
robot/test	AIY-projects-python/checkpoints	test_serial.py	シリアル通信テスト
robot/service	/lib/systemd/system	my_robot.service	自動実行サービス

ダウンロードされた ZIP ファイルをパソコンで解凍します。解凍された「robot」フォルダを USB メモリにコピーしてください。USB メモリを Raspberry Pi に差し込み、USB メモリの「robot」フォルダを転送先「`/home/pi/`」にドラッグアンドドロップすれば、サンプルファイルが Raspberry Pi の「`/home/pi/robot/`」フォルダにコピーされます。

ターミナルに次のようなコマンドを入力して、`samples.sh` を実行してください。このスクリプトでは、サンプルファイルをコピーする上で、ファイルの実行権限も設定します。

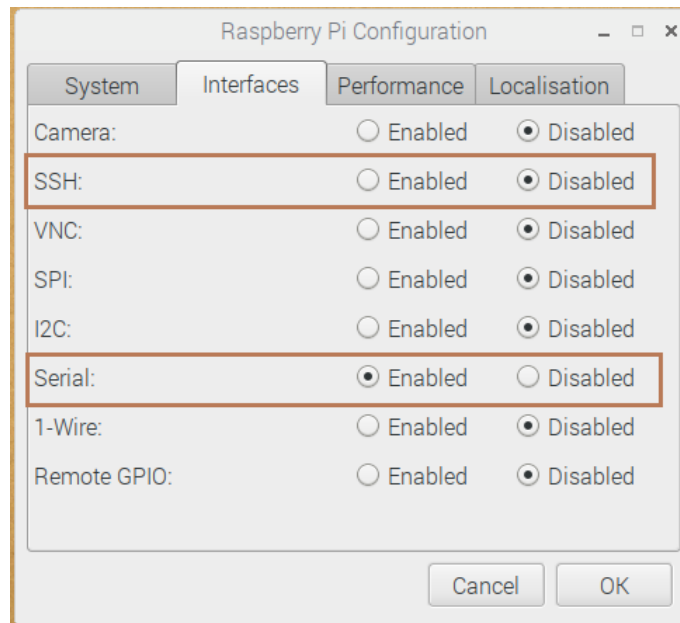
```
pi@raspberrypi - $ sh robot/setup/samples.sh↓
```

3.2 シリアル通信ポート

Raspberry Pi のロボット制御には、Raspberry Pi のシリアル通信ポートを利用しますが、リモート接続 SSH (Secure SHell) も同じシリアル通信ポートを使います。トラブルを避けるため、SSH を無効にすることを推奨します。

1. タスクパーメニュー⇒「Preferences」⇒「Raspberry Pi Configuration」メニューを選択すると、「Raspberry Pi の Configuration」ウィンドウが表示される。

2. 「Interfaces」タブの「SSH」の「Disabled」ラジオボタンを押せば、SSHを無効化することができる。さらに「Serial」の「Enabled」ラジオボタンを押せば、ロボットを操作するシリアル通信を有効化することができる。（下図）。



シリアル通信の確認には、オプションとして次の通信テストを行ってもよいです。

1. 通信ケーブル（Rapiro の場合、6 芯フラットケーブル）で、Raspberry Pi をロボットにつながる。
2. ロボットの電源を入れる。
3. デスクトップの「Start dev terminal」アイコンをダブルクリックして、「Start dev terminal」ターミナルウィンドウが表示される。
4. 「Start dev terminal」ターミナルに“checkpoints/test_serial.py”を入力して、シリアル通信のテストプログラムを起動する。

```
pi@raspberrypi :~/AIY-projects-python $ checkpoints/serial.py ↓
```

5. ロボットが左手と右手を順次に上げれば、シリアル通信に特に問題ないことを示す。

3.3 会話ロボットの実行

ターミナルに次のようなコマンドラインを入力して、会話ロボットプログラムを起動します。

```
pi@raspberrypi :~/AIY-projects-python $ src/robot.py↓
```

1. 最初には、“こんにちは、どういふご用件ですか。”と話し、ロボットは両手を振り、話しかける。
2. ロボットがマイクロフォンを握る右手を前に出し、話しかけを待つ。
3. なにか音声言葉が認識されると、ロボットは手を振る。同時に認識結果をターミナルに出力する。
4. 音声認識結果には、音声制御キーワードが含まれるかどうか判断する。ロボットが認識する制御キーワードには“前に”と“後ろに”と“右手”と“左手”

がある。たとえば、“左手を上げて”と指示すると、ロボットは目の色が赤になり、左手を上を振りながら、“はい、左手をあげます。”としゃべる。

5. 音声認識結果には、音声制御キーワードが含まれない場合、Google アシスタントの返事をしゃべる。たとえば、“いま何時ですか？”と現在時刻を聞くと、“時刻は x 時 x 分です。”のような返事が聞こえる。
6. 連続 5 回なにも話しが聞こえない場合、1.に戻り、音声で“こんにちは、どういいうご用件ですか。”と話しかける。

3.4 起動時自動実行サービスの有効化

起動時の自動起動サービス「my_robot」は、スクリプト samples.sh の実行により、すでにシステムに登録されましたが、まだ有効化にされていません。systemctl コマンドを使い、このサービスを有効化にします。次回電源を入れると、会話ロボットのプログラム「robot.py」が自動的に実行されます。

```
pi@raspberrypi - $ sudo systemctl enable my_robot
```