

音声認識システムを利用した教材開発

— IoT 開発教材作成についての検討 2 —

電子情報技術科 末永 聖平

Teaching material development using voice recognition system — Consideration about preparation of IoT development teaching materials 2 — Shohei SUENAGA

概要 前回、ESP32 を利用した Web サーバーを構築し、ハードウェア制御を行う IoT (Internet of Thing) システムの教材を作成した。この開発した教材も良かったが、さらにより良い教材を開発したいと思い、新たな教材開発を行う。学生にもより興味を持つことできる教材を開発する。開発検討した結果として、科の実技科目の中に取り入れ実施していく予定である。

1. はじめに

現在、中国職業能力開発大学校附属島根職業能力開発短期大学校(以下、島根校)の電子情報技術科では、ネットワーク、ハードウェア、ソフトウェアなどの科目が実施されている。しかし、それぞれの科目の内容があまり密接に結びつけられていない部分がある。これらの技術を結びつけた製品は IoT (Internet of Thing) と呼ばれ、様々なモノがインターネットに接続され、制御する対象になっている。

平成 27 年版の総務省発行の情報通信白書では、今後の IoT デバイス数は、さらに多くなることが予想されている。

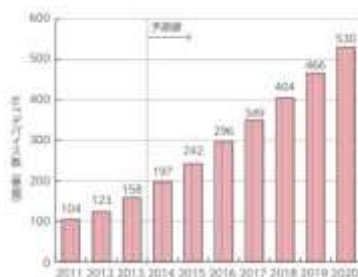


図1 IoTデバイス数の推移

図1に示すされるよう IoT 製品が益々身近になってきている。

IoT 製品を制御する方法は、スマートフォンなどの情報端末から操作することが多い。最近ではスマートスピーカーをはじめとした人の声を利用した制

御方法も利用されている。これを音声認識と言う。教材として、この音声認識を利用した制御方法について検討した。音声認識は、各社から公開されている API (Application Programming Interface) を利用することが前提である。以下のようなものがある。

- Apple Siri API
- Google Cloud Speech API
- IBM Watson Speech to Text API
- Microsoft Bing Speech API など

これらを利用するには、ユーザー登録、無料利用枠に限りがあるなど、授業展開するためには制限がある。この制限をほぼすべてなしで利用することができる音声認識技術 Web Speech API に注目した。この API を利用し、学生たちにこれらを組み合わせた IoT 技術を養うための教材開発について検討する。また、利用する教材に出来るだけ費用が発生しないよう検討を行う。

2. Web Speech API

Web Speech API は、Web ブラウザ上で実行することが可能な音声認識 API である。W3C において開発が進められており、仕様が公開されている。インターネット接続は必要となるが、音声認識自体は無料で利用することができる API である。API の呼び出

しには、JavaScript を利用する。

以下の2つの機能を Web ブラウザ上で動作させることができる。

- 音声認識（音声を文字にする）
- 音声合成（文字を音声にする）

API 仕様を確認し、併せて Google 社からはデモサイトが公開されており、ここで動作検証を実施した。その結果、教材として利用する内容として問題ないことを確認することができた。

3. 教材の構成

Web Speech API を実行するには JavaScript を利用する必要がある。そのため、JavaScript を実行できる環境を整えつつ、併せて、無線 LAN 環境を利用しハードウェアの遠隔制御を行う。

今回の教材においても、前回の教材として利用した ESP32-DevKitC（以下、ESP32）基板で、無線 LAN 環境を構築し、ハードウェア制御を行う。JavaScript、HTML ファイルの配信に、Web サーバーを構築する。そこで、図2のように構成することとした。



図2 ネットワーク構成

図2の中央のパソコンで仮想環境を用いて Web サーバーを構築する。併せて、ESP32 のプログラム開発も同時に行う。さらに、Android タブレットから、音声で ESP32 制御を行う仕組みとする。

利用した機材を以下に示す。

- 無線 LAN アクセスポイント
- Android タブレット

- パソコン（Windows 10 Enterprise）
- ハードウェア制御（ESP32）
- ブレッドボード、LED、リレーなどの制御対象

これらの利用した教材は、所属施設内にあったものの活用した。どの施設においても用意できるものと思われる。

4. 授業における展開

予定している実習は、週に1回200分、連続9週実施。受講する学生は、ネットワーク、ハードウェア、ソフトウェアに関する講義、実習は事前に受講済みである。この前提に合わせた教材開発を行う。

講義、実習内容の検討を行い、9回の授業内容を表1のようにすることとした。

表1 検討授業内容

1 週目	開発環境の構築、ハードウェアの準備
2 週目	ハードウェアの構成、接続、ESP32 ソフトウェア開発
3 週目	ネットワーク構成、仮想環境構築、Web サーバー構築
4 週目 ～7 週目	Web ページソフトウェア開発、Web Speech API の実装
8 週目 ～9 週目	自由課題とし、今までの実習内容に基づき開発を行う

実習の目標としては、ESP32 と LED、リレーなどのハードウェアを正しく結線し、これらを制御するプログラムを記述することができる。さらに、作成したものを無線 LAN により接続し、Android タブレットから音声認識情報により制御することができることとする。

5. ESP32 構成

ESP32 には、3つの開発方法が用意されている。

- ESP-IDF
- Micro Python

- Arduino IDE

今回は、授業展開の中で利用している当施設の学生が使い慣れた Arduino IDE 開発環境を採用した。ESP32 の開発環境を整えるためにインターネット接続環境を用意することにより、環境を整えることができる。

6. ハードウェア制御対象

音声認識を用いてハードウェア制御を行う対象を以下の3つと組み合わせとした。

1. LED
2. リレー
3. サーボモーター
4. 上記1~3の組み合わせ

この3つとした理由は、所属施設内に十分な数があり、ESP32 のプログラム自体の開発も比較的容易であるところからである。

7. 無線ネットワークの構成

ESP32 の仕様上、無線機能として無線 LAN (IEEE802.11b/g/n)、Bluetooth (v4.2、BLE) が搭載されている。

実習環境は、Web Speech API を利用するために、インターネット接続が必要となる。そこで、学生 1 人ずつに無線 LAN アクセスポイントを渡し、アクセスポイント経由でインターネット接続を自分で確保するように構成した。図 3 に示す。

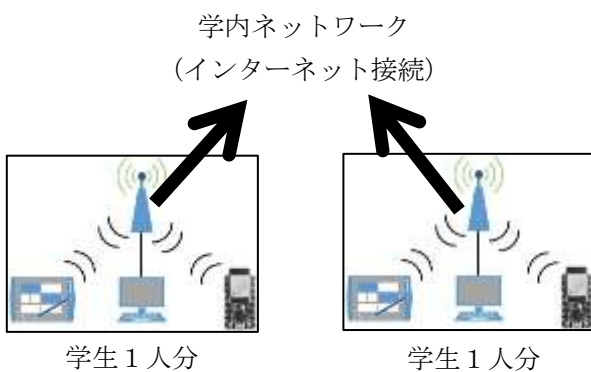


図 3 教室内でのネットワーク構成

図 3 のネットワーク構成が必要になるため各学生に、以下の 5 つの IP アドレス割り振り重複しないよう構成した。

- 無線 LAN アクセスポイント
- Android タブレット
- Windows 10 Enterprise
- 仮想環境 (Linux)
- ESP32

ちなみに、Web Speech API を動作させるためにインターネット接続が必要となるのは、Android タブレットのみである。

8. ソフトウェア開発

ソフトウェアによるハードウェア制御に関しては、デジタル制御が中心とした。ESP32 制御プログラムが Arduino マイコンのプログラム記述方法と違いはないため授業の復習として実施する。

無線 LAN アクセスポイントと ESP32 を接続する。図 4 のように制御対象となる ESP32 は、固定 IP アドレス設定とする。固定とした理由は、制御情報を渡す際に DHCP サーバーを利用すると IP アドレスが変更され、変更の度に設定等、プログラムを作り直さなければならないためである。ESP32 の無線接続プログラムは、Arduino IDE にサンプルプログラムが用意されており、これを活用する。

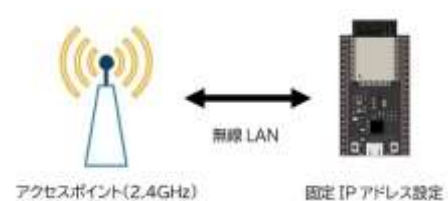


図 4 アクセスポイントと ESP32 の接続

9. 仮想環境構築

Web Speech API を動作させるためには、Web サーバーが必要となる。しかし、Web サーバーとブラウザが同じ端末上で動作することができないことになっている。図 5 のような動作環境である。そのため、

利用している Windows 10 上に仮想環境を構築することとした。



図5 Web Speech API 動作環境

仮想環境の構築に、以下の3つを検討した。

- Oracle VirtualBox
- VMWare
- Microsoft Hyper-V

今回は、Microsoft Hyper-V を採用した。Windows 10 Professional もしくは Enterprise Edition であれば無料で利用できるためである。

Hyper-V を利用し、Linux OS のインストールを行う。採用する Linux ディストリビューションは、CentOS とした。これは、学内の授業において利用されており、環境構築など学生が操作に慣れているためである。そこに、Web サーバー、SSL サーバーを構築する。具体的にインストールしたソフトウェアは、以下のとおりである。

- Web サーバーに、Apache。
- SSL サーバーに、OpenSSL。

その他 CentOS を動作させるための環境設定を行う。Web Speech API を動作させるための最低限の環境構築を行う。

SSL サーバーを利用しているのは、Web Speech API を実行するためには、暗号化通信上で実行することとされているためである。そのため、Web サーバーと連携させ https 通信を行う必要がある。公開鍵などは、自己証明書で完結させることとした。



図6 暗号化通信による動作テスト

動作検証時に、ESP32 との連携を行うために、不十分なところが判明したため、構築した仮想環境に、さらに、PHP (Hypertext Preprocessor) の動作環境を構築した。Web サーバーである Apache と連携させる。

10. Web Speech API 動作検証

以下の図7のような動作環境を用意し、Web Speech API の動作テストを行う。動作において重要な点は、Web サーバーと操作端末の通信である。併せて、音声認識を実行するためには、操作端末がインターネットに接続する必要がある。

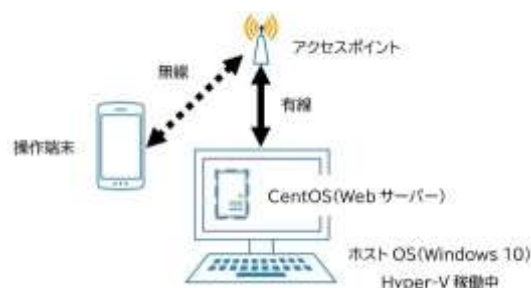


図7 Web Speech API 動作テスト

通信環境が整えることができれば、Web サーバー経由で HTML を利用した Web ページを表示させる。Web ページの表示が確認できれば JavaScript を利用する。図8に動作テストページを示す。



図8 JavaScript 動作テスト

さらに、JavaScript 内で Web Speech API の呼び出し、音声認識の検証を行う。この際の音声認識は、人から操作端末へ (音声認識)、操作端末から人 (音声合成) へどちらも声で操作できることを確認する。



図9 Web Speech API 動作テスト

ところが所属施設の授業では、Web ページ作成の授業を実施していない。そのため、学生は、HTML、JavaScript の記述は初めてである。しかし、Web Speech API 呼び出しプログラムは、数行の記述で検証が可能であるため動作に必要な個所に限定して、記述方法を説明することとした。

また、ESP32 の制御情報を格納するテキストファイルが必要となった。そのため PHP の記述についても理解を深める。図 10 に、PHP サンプルを示す。



図 10 PHP 動作テスト

PHP の動作内容としては、テキストファイルの書き換え動作のみに抑える。図 11 に示すよう、PHP ファイルを実行し、ハードウェア制御情報が書かれたテキストファイルの書き換えを行う。

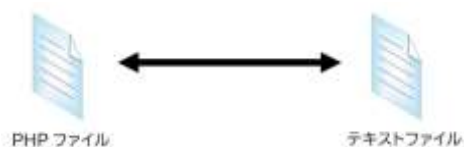


図 11 PHP によるテキストファイル操作

11. 全体構成

全体構成としては、図 12 のような構成とし、全動作の検証を行う。操作端末から Web サーバーを経由し、ESP32 を制御する。



図 12 全体動作構成

ESP32 の制御は、仮想環境上の Web サーバーとの連携が重要となる。そこで、連携について検討した内容は、以下の 3 点である

- MQTT (Message Queue Telemetry Transport) の利用
- クラウドシステム (IFTTT (if this then that)、AWS など) の利用
- HTTP を利用してテキストファイルを監視

この中で採用したものは、図 13 で示すような学生にとって理解しやすいであろう HTTP によるテキストファイル監視である。採用した理由として、ESP32 の HTTP 接続のサンプルファイルを活用することで、プログラムの記述も比較的楽に行えるためである。

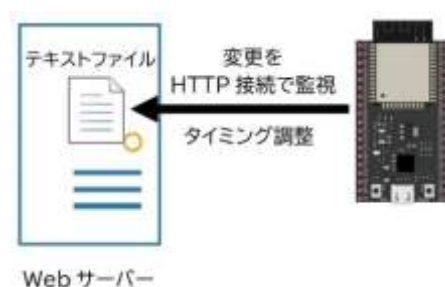


図 13 ESP32 制御方法

結果的に詳細な動作構成は、図 14 とした。

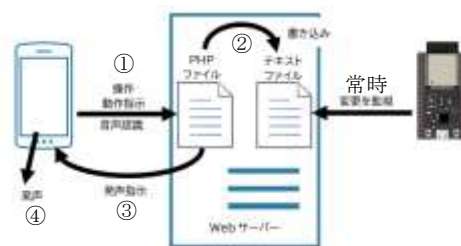


図 14 詳細全体構成

具体的には、音声認証を利用し、ロボットアーム (サーボモーター) を制御する。配線後の構成を図 15 とした。



図 15 実際の ESP32 とハードウェア配線

ロボットアームの特徴を踏まえ、動作を決定する音声ルールを以下の表 2 のように決定した。

表 2 音声動作ルール

認識音声	動作
電気を点けて	LED オン
電気を消して	LED オフ
右向いて	下サーボを 180 度へ
左向いて	下サーボを 80 度へ
はさみを開いて	上サーボを 90 度へ
はさみを閉じて	上サーボを 135 度へ
お辞儀して	中央サーボを 30 度へ
元に戻って	すべての位置情報を元の状態に戻す

今回はロボットアームを利用したが、このようなアームがなければサーボモーター単体で利用し、適当な音声ルールを作成するとよい。

12. テキスト開発と検証

以上のことを踏まえテキスト作成を行った。作成したテキストを利用し、当校 2 年生の授業において実施した。実際に、全内容を通して行うことはできなかったが、仮想環境構築、ネットワーク接続、プログラムに関する実習を一通り実施した。

学生は、以前作成した教材においても、遠隔操作を行う仕組みを理解することができていたが、今回は遠隔操作入力を自分の声で行うことができることで、スマートスピーカーの仕組みに興味を持つことを確認することができた。また、音声認識自体がどのように動作するかについても興味を持つ学生が出

てきた。このような興味を持ってくれたことから、今後の授業において音声認識の基となる人工知能などの話に展開もしやすくなると判断している。

13. まとめ

学生には IoT についてより興味を持つことができたのではないかと考える。市販されている音声認識 (Google Home、Amazon Alexa など) による制御方法について理解を深めることができたはずである。この後は、クラウド環境を利用した音声認識にも取り組んでみてもらい、クラウドシステムへより理解を深めていく方向に進めていきたい。

またこの技術を利用した卒業制作などにも展開できることを期待している。

講師側としては、テキスト内容に沿って実施するだけではなく、もっとこうしたい、もっとこうすればを導いていけるようにしなければならない。IoT は、どのようなハードウェアが繋がれば、人のためになるのかを理解し、創造する力が必要であるためである。

学生に対しては、今後必要となる技術を理解でき、これを今後のモノづくりの現場において生かされていくことを希望する。

参考文献

- 1) 総務省：レイヤー別にみる市場動向
<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r01/html/nd112130.html>
- 2) Web Speech API 動作デモ
<https://www.google.com/intl/ja/chrome/demos/speech.html>
- 3) Web Speech API
https://developer.mozilla.org/ja/docs/Web/API/Web_Speech_API

著者 E-mail Suenaga.Shohei@jeed.go.jp