

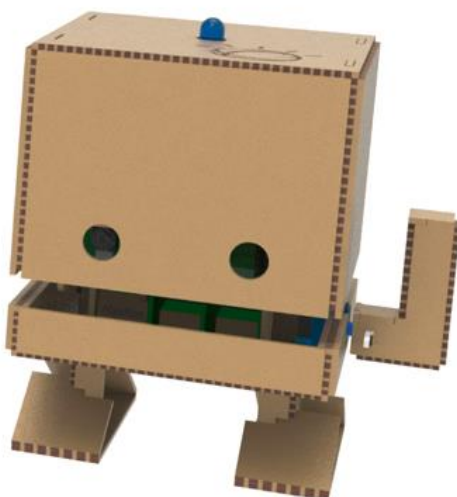


# マイク・スピーカーを使って TJBOT と 手を振って会話。

Watson 「Speech to Text」、「Text to Speech」と Raspberry Pi、USB マイク、Bluetooth スピーカー、LED、サーボモータを使って、Watson と簡単な会話をします

## <使用する Watson サービス>

### 「Speech to Text」、「Text to Speech」



こんにちは、私の名前は TJBOT です！

私は、楽しい方法で Watson サービスを理解するために、お手伝いをするオープンソースプロジェクトです。

TJBOT は、IBM 最初の会長兼最高経営責任者（CEO）であるトーマス・J・ワトソンの名前を愛称としています。

TJBOT は、IBM Research の Maryam Ashoori によって、認知対象の設計と実装におけるベストプラクティスを見つけるための実験として作成されました。

## もくじ

- 【1】Web Site 情報
- 【2】内容
- 【3】H/W セットアップ
- 【4】Watson「Speech to Text」、「Text to Speech」サービスの作成
- 【5】Rasbian の最新化と Node.js、npm、のインストール
- 【6】コードのダウンロードとインストール
- 【7】「Speech to Text」、「Text to Speech」、構成情報の設定
- 【8】USB マイクの設定
- 【9】プログラムの実行
- 【10】日本語対応に変更
- 【11】日本語対応版プログラムの実行

## 材料 (価格は変わる可能性があります。消費税、郵送料は含まれません)

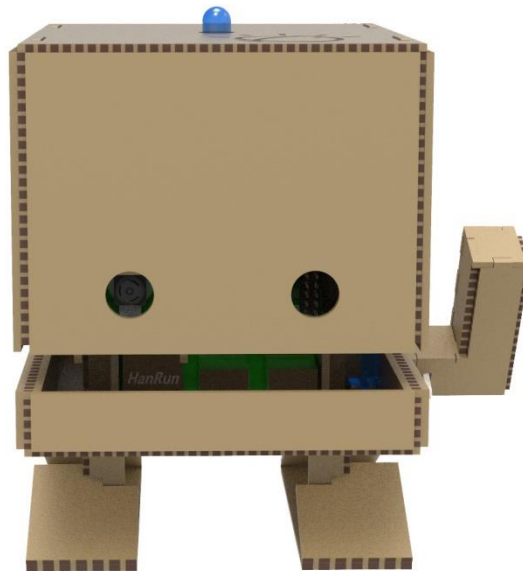
材料名	補足	価格 (消費税、送料含まず)	購入先例
厚紙	2mm厚	¥650	東急ハンズ
レーザーカット費用	1000円/10分	¥500	coromoza
Raspberry Pi3 Model B	Bluetooth、無線LAN含む。	¥5,600	秋月電子通商
電源 (Raspberry Pi3用)	スイッチングACアダプター5V2.5A AD-B50P250、USBケーブル Aオス-マイクロBオス 0.15m A-microB	¥1,210	秋月電子通商
microSD Card 8GB Class10	Transcend microSDHCカード 8GB Class10 (無期限保証)	¥1,200	アマゾン
LED	NeoPixel スルーホールLED	¥691	スイッチサイエンス
Bluetoothスピーカー	Anker SoundCore mini コンパクト Bluetoothスピーカー	¥2,399	アマゾン
USBマイク	BU-Bauty PC Mac用USBマイク 超小型 超ミニ 22mmx18mmx5mm	¥599	アマゾン
マイクロサーボ SG92R		¥500	アマゾン
	小計	¥13,349	
HDMIディスプレイ	Raspberry Pi3初期設定時のみ使用。テレビで代用		
HDMIケーブル	Raspberry Pi3初期設定時のみ使用。HDMI 1.4ケーブル	¥250	秋月電子通商
USBマウス	Raspberry Pi3初期設定時のみ使用。Logicool ロジクール 有線光学式3ボタンマウス M100r ブラック	¥475	アマゾン
USBキーボード	Raspberry Pi3初期設定時のみ使用。サンワサプライ USBキーボード(ブラック) SKB-L1UBK	¥664	アマゾン
	小計	¥1,389	
	合計	¥14,738	

## 【1】Web Site 情報

<https://www.instructables.com/id/Build-a-Waving-Robot-Using-Watson-Services/>

## 【2】内容

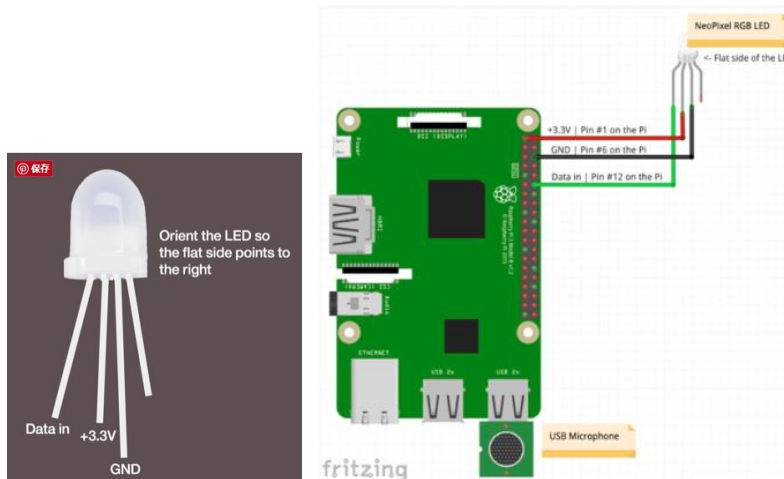
Watson 「Speech to Text」、「Text to Speech」と Raspberry Pi、USB マイク、Bluetooth スピーカー、LED、サーボモータを使って、Watson と簡単な会話をします



### 【3】H/W セットアップ

#### (1) LED

スイッチサイエンス社 NeoPixel スルーホール LED

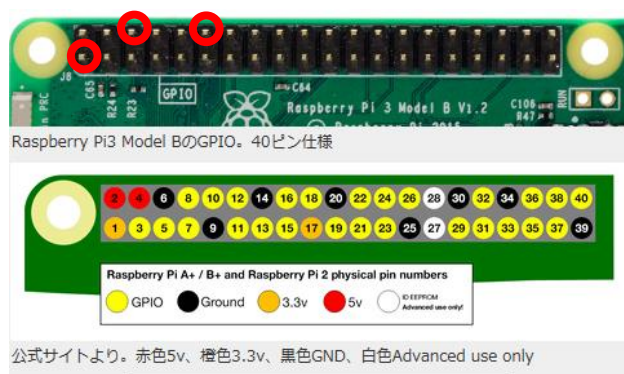


LED は平らにカットされている方から

1. --- 使用しない
2. --- GND ⇒ 6 番ピン
3. --- +3.3V ⇒ 1 番ピン
4. --- Data in ⇒ 12 番ピン

になっているが、

(注意！) RaspberryPi に接続する際は、2 ⇒ 4 ⇒ 3 の順番に接続する



## (2) サーボモータ



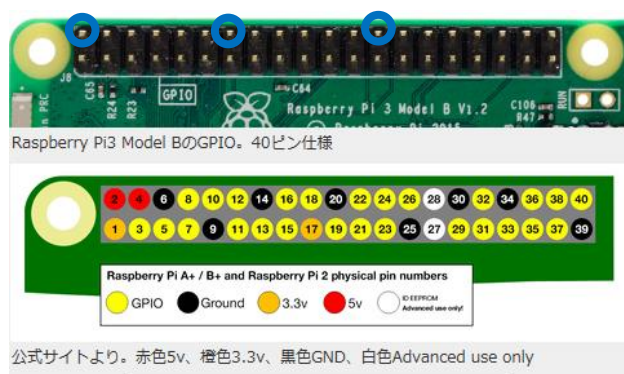
マイクロサーボ SG92R

配線は：茶 = GND、赤 = 電源 [+ ]、橙 = 制御信号

- 1. --- GND           ⇒14
- 2. --- +5V           ⇒2
- 3. --- Data in       ⇒26

になっている。

(注意！) RaspberryPi に接続する際は、1 ⇒ 3 ⇒ 2 の順番に接続する



## (3) USB マイク

BU-Bauty PC Mac 用 USB マイク 超小型 超ミニ 22mmx18mmx5mm を使用しました。



\$ lsusb

コマンドで認識されているか確認します。

```
pi@raspberrypi:~$ lsusb
Bus 001 Device 004: ID 0d8c:0016 C-Media Electronics, Inc.
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

“Bus 001 Device 004: ID 0d8c:0016 C-Media Electronics, Inc”で認識されています。

以下のコマンドでハードウェアデバイスを調べます。

```
$ arecord -l
```

```
pi@raspberrypi:~$ arecord -l
**** ハードウェアデバイス CAPTURE のリスト ****
カード 1: Micophone [USB Micophone], デバイス 0: USB Audio [USB Audio]
  サブデバイス: 1/1
  サブデバイス #0: subdevice #0
```

カード番号は1です。

マイクの感度を調整します

```
$ amixer sset Mic 50 -c 1
```

(-cに続く番号は card 番号です)

```
pi@raspberrypi:~$ amixer sset Mic 50 -c 1
Simple mixer control 'Mic',0
  Capabilities: cvolume cvolume-joined cswitch cswitch-joined
  Capture channels: Mono
  Limits: Capture 0 - 62
  Mono: Capture 50 [81%] [16.59dB] [on]
```

81%の値がマイク感度です。

録音してみます。

```
$ arecord -D plughw:1,0 -d 10 -f cd test.wav
```

(plughw:1,0 は arecord -l のカード番号とデバイス番号です。生成する音声ファイルは test.wav)  
PCなどに転送して音を確認してください。

#### (4) Bluetooth スピーカー

Anker SoundCore mini コンパクト Bluetooth スピーカーを使用します。

##### ・Bluetooth パッケージ PulseAudio パッケージをインストール

```
$ sudo apt-get update
```

```
$ sudo apt-get dist-upgrade
```

```
$ sudo apt-get install bluetooth
```









```
pi@raspberrypi:~$ sudo systemctl start pulseaudio.service
pi@raspberrypi:~$ sudo systemctl enable pulseaudio.service
Created symlink from /etc/systemd/system/multi-user.target.wants/pulseaudio.service to /etc/systemd/system/pulseaudio.service.
pi@raspberrypi:~$ sudo gpasswd -a pi pulse-access
ユーザ pi をグループ pulse-access に追加
```

ここまでで PulseAudio 関連の設定が完了しましたので、次に bluetoothctl で Bluetooth デバイスを接続します。

スピーカーの電源をオン。

```
$ sudo bluetoothctl
```

```
[bluetooth]# power on
```

デバイスをスキャンし、スピーカーを探します。

```
[bluetooth]# scan on
```

Bluetooth スピーカーが見つかったら、スキャンを停止します。

```
[bluetooth]# scan off
```

```
pi@raspberrypi:~$ sudo bluetoothctl
[NEW] Controller B8:27:EB:C1:8F:AF raspberrypi [default]
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:27:EB:C1:8F:AF Discovering: yes
[NEW] Device 00:E0:4C:58:6C:EC SoundCore mini
[NEW] Device CF:E9:B1:E0:D6:98 ID107 HR
[CHG] Device 00:E0:4C:58:6C:EC Class: 0x240404
[CHG] Device 00:E0:4C:58:6C:EC Icon: audio-card
[bluetooth]# scan off
[CHG] Device CF:E9:B1:E0:D6:98 RSSI is nil
[CHG] Device 00:E0:4C:58:6C:EC RSSI is nil
[CHG] Controller B8:27:EB:C1:8F:AF Discovering: no
Discovery stopped
[bluetooth]#
```

[NEW] Device 00:E0:4C:58:6C:EC SoundCore mini  
で Bluetooth スピーカーのデバイス情報が入手できました。

```
[bluetooth]# trust 00:E0:4C:58:6C:EC
```

```
[bluetooth]# pair 00:E0:4C:58:6C:EC
```

```
[bluetooth]# connect 00:E0:4C:58:6C:EC
```

「ピロリン」という音が鳴ると、接続完了です。

```
[bluetooth]# exit
```

で bluetoothctl から抜けます。

#### ・なかなか接続できない場合

<https://raspberrypi.stackexchange.com/questions/44497/having-an-issue-with-bluetooth-manager-pairing-on-the-pi>

Syslog を確認します。

```
$ cat /var/log/syslog
```

“bluetoothd[794]: a2dp-sink profile connect failed for 00:E0:4C:58:6C:EC: Protocol not available”というエラーの場合下記手順を試してください。

(1) PulseAudio Bluetooth ライブラリーがインストールされているか確認

```
$ sudo apt-get install pulseaudio-module-bluetooth
```

(2) Pulse Audio server が稼働しているか確認

まず停止します。

```
$ sudo killall pulseaudio
```

起動します。

```
$ pulseaudio --start
```

(3) bluetoothctl で connect します。

```
[bluetooth]# connect 00:E0:4C:58:6C:EC
```

## 【4】Watson「Speech to Text」、「Text to Speech」サービスの作成

Bluemix にログイン

<https://console.bluemix.net/>



### Log into IBM Bluemix

IBMID またはメール・アドレスを入力してください IBM IDをお忘れですか?

Email or IBMID

次へ

New to Bluemix? [Sign Up](#)



### Log into IBM Bluemix

IBM ID:

パスワード パスワードをお忘れの場合

| Password

Log in

別の IBM アカウントまたは E メールを使用してください

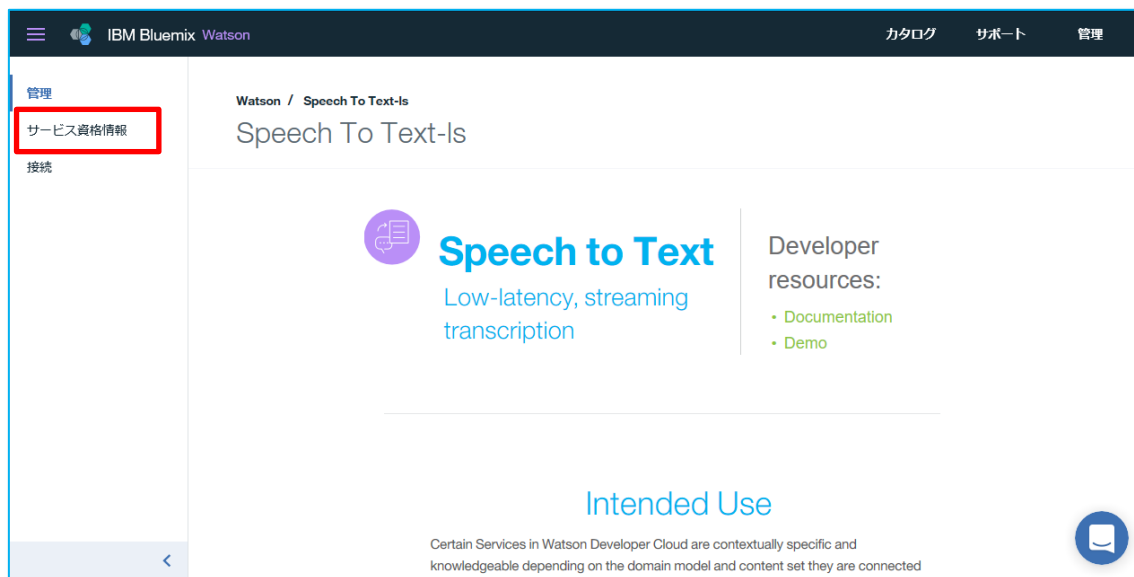
## (1) カタログより「Speech to Text」を選択

The screenshot shows the IBM Bluemix Catalog search results for 'speech to text'. The search bar contains 'speech to text' and a 'フィルター' (Filter) button. The results are categorized under 'サービス' (Services) and 'Watson'. A red box highlights the 'Speech To Text' service card, which includes the text '低遅延、ストリーミング・トランスクリプション' (Low latency, streaming transcription) and the IBM logo. The left sidebar shows navigation options like 'インフラストラクチャー' (Infrastructure), 'アプリ' (Apps), and 'サービス (1)' (Services (1)).

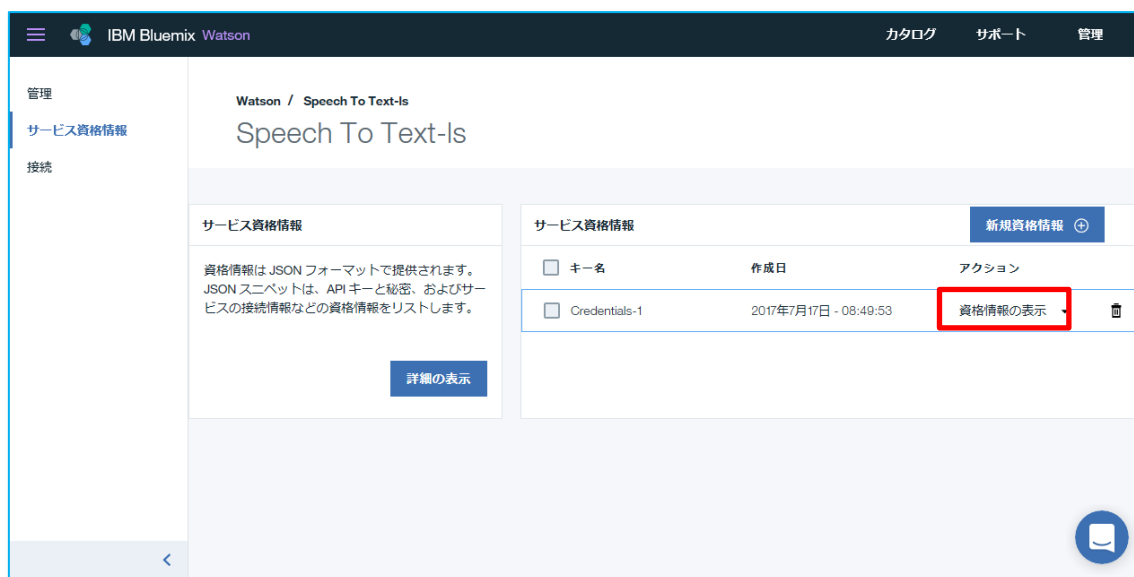
## 「作成」をクリック

The screenshot shows the details page for the 'Speech To Text' service. The page includes a description of the service, its name ('Speech To Text-Is'), and its credentials ('Credentials-1'). Under the 'フィーチャー' (Features) section, it lists '使用可能な言語' (Supported languages) and 'メタデータ' (Metadata). A red box highlights the '作成' (Create) button at the bottom right of the page. The page also includes a '接続' (Connect) dropdown menu set to 'アンバインドのまま' (As is) and a 'ヘルプが必要ですか?' (Need help?) link.

「サービス資格情報」をクリック



「資格情報の表示」をクリック



「username」「password」を後程使用しますので、メモ帳などにコピー保存してください。

The screenshot shows the IBM Bluemix Watson console interface. The main heading is "Speech To Text-Is". On the left, there is a navigation menu with "サービス資格情報" (Service Credentials) selected. The main content area displays a table of service credentials. One credential is listed with the name "Credentials-1", created on "2017年7月17日 - 08:49:53", and an action "資格情報の表示". Below the table, a JSON snippet is shown, containing the URL, username, and password for the service, with some fields redacted with yellow boxes. A "詳細の表示" (Show details) button is visible at the bottom of the JSON snippet.

```
{  
  "url": "https://stream.watsonplatform.net/speech-to-text/api",  
  "username": "e[REDACTED]3",  
  "password": "z[REDACTED]"  
}
```

## (2) カタログより「Text to Speech」を選択

The screenshot shows the IBM Bluemix Catalog interface. The search bar contains "Text to Speech". The search results show the "Text to Speech" service under the "Watson" category. The service description is "テキストから自然な音声を合成します。" (Synthesize natural sound from text). The IBM logo is visible below the description. The service is highlighted with a red box. The left sidebar shows the navigation menu with "サービス (1)" (Services (1)) selected. The bottom of the page has a footer with the text "さらに詳しい情報" (For more information) and "Bluemix Experimental Services を確認して、試験ランタイムと試験サービスを試してみてください。" (Check out Bluemix Experimental Services, try the trial runtime and trial service).

## 「作成」をクリック

IBM Bluemix カタログ

← すべて表示

## Text to Speech

Text to Speech サービスは、テキストと自然言語を処理し、適切な抑揚をつけて合成した音声出力を生成します。利用可能な音声は次のとおりです。

サービス名: Text to Speech-ja

資格情報名: Credentials-1

接続: アンバインドのまま

資料の表示

作成者: IBM  
公開日: 2017/07/18

### フィーチャー

- 英語 (米国)  
2種類の女性の声、1種類の男性の声 (Jeopardy の Watson の音声)
- 英語 (英国)  
1種類の女性の声
- フランス語  
1種類の女性の声
- ドイツ語  
1種類の女性の声、1種類の男性の声

ヘルプが必要ですか?  
[Bluemix 営業担当へのお問い合わせ](#)

月額費用の計算  
[費用計算](#)

**作成**

## 「サービス資格情報」をクリック

IBM Bluemix Watson

管理

サービス資格情報

接続

Watson / Text to Speech-fe

## Text to Speech-fe

Developer resources:

- Documentation
- Demo

### Intended Use

Certain Services in Watson Developer Cloud are contextually specific and knowledgeable depending on the domain model and content set they are connected



## 「資格情報の表示」をクリック



「username」「password」を後程使用しますので、メモ帳などにコピー保存してください。



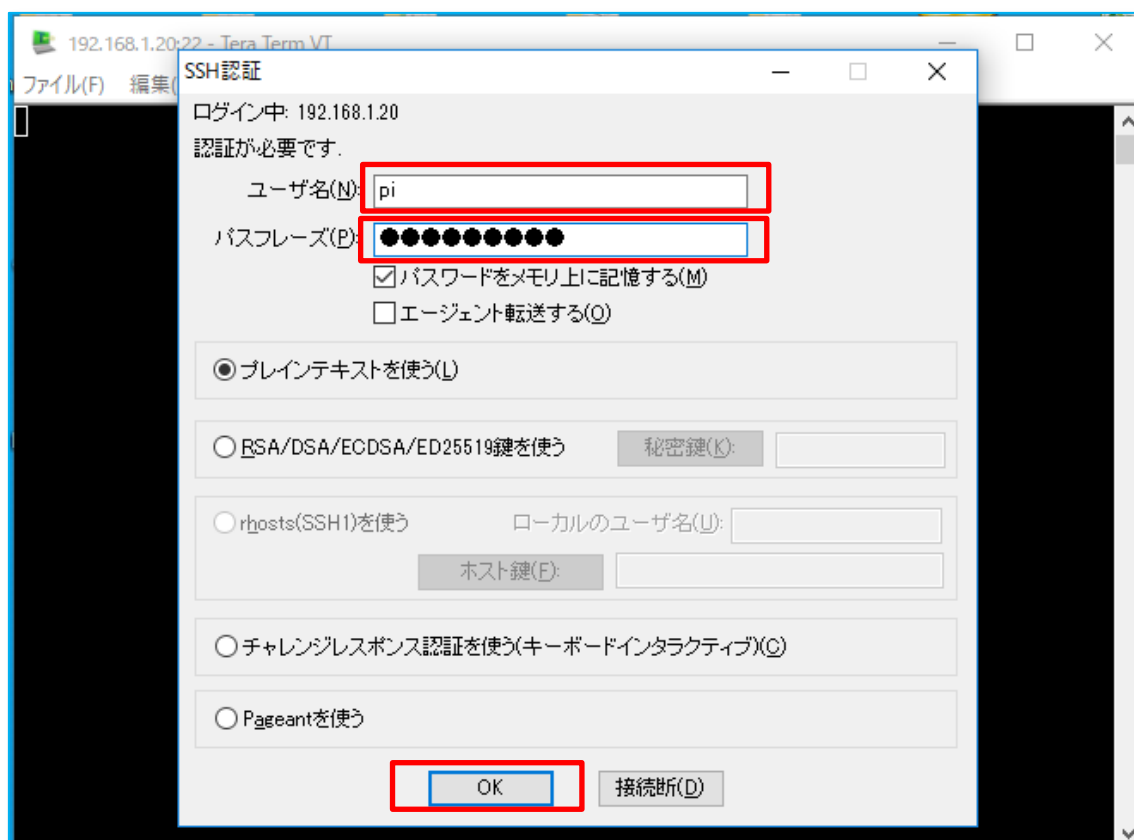
```
{
  "url": "https://stream.watsonplatform.net/text-to-speech/api",
  "username": "e[REDACTED]30",
  "password": "S[REDACTED]p5"
}
```

## 【5】Rasbian の最新化と Node.js、npm、のインストール

Teraterm などで SSH 接続します。ID、パスワードの初期値は、

ID:pi

Password: raspberry です。



以下のコマンドを実行してください。

```
$ sudo apt-get update
```

```
$ sudo apt-get dist-upgrade
```

```
$ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
```

```
$ sudo apt-get install -y nodejs
```

## 【6】コードのダウンロードとインストール

```
$ git clone https://github.com/victordibia/tjwave
$ cd ~/tjwave
$ npm install
$ sudo apt-get install alsa-base alsa-utils
$ sudo apt-get install libasound2-dev
$ sudo apt-get install pigpio
```

```
pi@raspberrypi:~/tjwave$ ls -al
合計 248
drwxr-xr-x  7 pi pi   4096  7月 24 14:59 .
drwxr-xr-x 22 pi pi   4096  7月 24 14:57 ..
drwxr-xr-x  8 pi pi   4096  7月 24 14:58 .git
-rw-r--r--  1 pi pi   159   7月 24 14:58 .gitignore
-rw-r--r--  1 pi pi  10014  7月 24 14:58 README.md
-rw-r--r--  1 pi pi   765   7月 24 14:58 config.default.js
-rw-r--r--  1 pi pi   4163  7月 24 14:58 dance.js
drwxr-xr-x  2 pi pi   4096  7月 24 14:58 images
drwxr-xr-x 117 pi pi   4096  7月 24 15:00 node_modules
-rw-r--r--  1 pi pi 154876  7月 24 14:58 output.wav
-rw-r--r--  1 pi pi   477   7月 24 14:58 package.json
drwxr-xr-x  2 pi pi   4096  7月 24 14:58 photos
-rw-r--r--  1 pi pi   3994  7月 24 14:58 servicerestart.js
drwxr-xr-x  2 pi pi   4096  7月 24 14:58 sounds
-rw-r--r--  1 pi pi   9063  7月 24 14:58 wave.js
-rw-r--r--  1 pi pi  15751  7月 24 14:58 wave_conversation.js
-rw-r--r--  1 pi pi   2119  7月 24 14:58 wavetest.js
-rw-r--r--  1 pi pi  13758  7月 24 14:58 workspace.json
```

## 【7】「Speech to Text」、「Text to Speech」、構成情報の設定

ディレクトリを `cd ~/tjwave` に移動します。

```
$ cd ~/tjwave
```

インストールされたデフォルト構成ファイル「`config.default.js`」を「`config.js`」という名でコピーし、「`config.js`」を編集します。

```
$ cp config.default.js config.js
```

```
$ sudo nano config.js
```

■ “【4】Watson「Speech to Text」、「Text to Speech」サービスの作成”で取得した、「`username`」「`password`」を登録します。

ここでは nano エディターを使用しました。

```
// Please replace the username and password with your Bluemix credentials

exports.STTPassword = ''; // Speech to text password
exports.STTUsername = ''; // Speech to text username
exports.STTCustomizationid = '' // Speech to text Customization id (if you are $

// Credentials for Watson Text to Speech service

exports.TTSPassword = ''; // Text to speech password
exports.TTSUsername = '';

// Credentials for Watson Conversation service
exports.ConPassword = ''; // conversation service password
exports.ConUsername = ''; // conversation service username
exports.ConWorkspace = ''; // Conversation workspace id

// Credentials for Vision Api
exports.VisionKey = '';

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page  ^U UnCut Text  ^T To Spell
```

各項目を登録し、`Ctrl+x ⇒ y ⇒ ↵`で保存します。

## 【8】USBマイクの設定

~/tjwave/wave.js

の wave.js で、USBマイクの設定値を追加します。

まず、USBマイクはのカード番号、デバイス番号を調べます。

```
$ arecord -l
```

```
pi@raspberrypi: ~/tjwave $ arecord -l
**** ハードウェアデバイス CAPTURE のリスト ****
カード 1: Microphone [USB Microphone], デバイス 0: USB Audio [USB Audio]
  サブデバイス: 1/1
  サブデバイス #0: subdevice #0
```

この例では、カード番号 1、デバイス番号 0 ですので、`plughw:1,0` を追加します。

### wave.js の 56 行あたり

```
var micInstance = mic({ 'rate': '44100', 'channels': '2', 'debug':
false,, 'exitOnSilence': 6 });
```

⇒

```
var micInstance = mic({ 'rate': '44100', 'channels': '2', 'debug': false, 'device':
'plughw:1,0', 'exitOnSilence': 6 });
```

```
context = new AudioContext
var _ = require('underscore');

/*****
* Step #2: Configuring the Microphone
*****/
In this step, we configure your microphone to collect the audio samples as you talk.
See https://www.npmjs.com/package/mic for more information on
microphone input events e.g on error, startcomplete, pause, stopcomplete etc.
*/

// Initiate Microphone Instance to Get audio samples
var mic = require('mic');
var micInstance = mic({ 'rate': '44100', 'channels': '2', 'debug': false, 'device': 'plughw:1,0', 'exitOnSilence': 6 });
var micInputStream = micInstance.getAudioStream();

micInputStream.on('data', function(data) {
  //console.log("Received Input Stream: " + data.length);
});

micInputStream.on('error', function(err) {
  console.log("Error in Input Stream: " + err);
});

micInputStream.on('silence', function() {
  // detect silence.
});
micInstance.start();
console.log("TJ is listening, you may speak now.");

/*****
* Step #3: Converting your Speech Commands to Text
*****/
line 56/283 (19%), col 1/120 (0%), char 1831/3086 (20%)
Get Help      WriteOut     Read File    Prev Page   Cut Text     Cur Pos
Exit          Justify     Where Is     Next Page   UnCut Text   To Spell
```

## 【9】プログラムの実行

```
$ cd ~/tjwave
```

```
$ sudo node wave.js
```

```
pi@raspberrypi:~/tjwave $ sudo node wave.js  
TJ is listening, you may speak now.  
█
```

「raise arm」

「weave arm」

「wave arm」

「leave arm」

「intorduce yourself」

「what your name」

「can you dance」

と話しかけると、手を振ったり、自己紹介したり、ダンスをしてくれます。

発音が悪く、なかなか認識してくれませんが、Google 翻訳の音声を使うとよく反応しました。

## 【 1 0 】日本語対応に変更

### ・wave.js 71 行あたり

```
console.log("TJ is listening, you may speak now.");
```

⇒

```
console.log("TJ は聞いてます。何か話しかけてください");
```

### ・wave.js 81 行あたり

```
content_type: 'audio/l16; rate=44100; channels=2'
```

⇒

```
model: 'ja-JP_BroadbandModel', content_type: 'audio/l16; rate=44100;  
channels=2'
```

### ・wave.js 106~109 行あたり

```
var containsWaveArm = (str.indexOf("raise") >= 0 || str.indexOf("weave") >= 0 ||  
str.indexOf("wave") >= 0 || str.indexOf("leave") >= 0 ) && ( str.indexOf($
```

```
var introduceYourself = str.indexOf("introduce") >= 0 && str.indexOf("yourself")  
>= 0 ;
```

```
var whatisYourname = str.indexOf("what") >= 0 && str.indexOf("your") >= 0 &&  
str.indexOf("name") >= 0 ;
```

```
var canYouDance = str.indexOf("can") >= 0 && str.indexOf("you") >= 0 &&  
str.indexOf("dance") >= 0 ;
```

⇒

```
var containsWaveArm = (str.indexOf("あげる") >= 0 || str.indexOf("振る") >= 0 ||  
str.indexOf("振る") >= 0 || str.indexOf("下ろす") >= 0 ) && ( str.indexOf("手") >=
```

```
0) ;
```

```
var introduceYourself = str.indexOf("自己") >= 0 && str.indexOf("紹介") >= 0 ;  
var whatisYourname = str.indexOf("あなたの") >= 0 && str.indexOf("名前") >= 0 &&  
str.indexOf("は") >= 0 ;
```

```
var canYouDance = str.indexOf("ダンス") >= 0 && str.indexOf("できます") >= 0 ;
```

### ・wave.js 113 行あたり

```
speak("Ok, I will wave my arm. Just for you.");
```

⇒

```
speak("はい。手を振ります。");
```

**•wave.js 116 行あたり**

```
 speak(" Hi, my name is TJ. I'm an open source project designed to help you access  
 Watson Services in a fun way. You can 3D print me or laser cut me, then use one of  
 my recipes to bring me to life. I can't wait to see what we do together. ");
```

⇒

```
 speak("こんにちは、私の名前は TJ です。あなたが楽しい方法で Watson Services を使うのを手助  
 けするように設計されたオープンソースプロジェクトです。あなたは 3D プリントをしたり、レーザーカットしたり  
 して、レシピの 1 つを使って私を作ります。一緒に何かやりましょう！ ");
```

**•wave.js 118 行あたり**

```
 speak(" My name is TJ. You can call me TJ Bot");
```

⇒

```
 speak("私の名前は TJ です。 TJ Bot と呼んでください。");
```

**•wave.js 123 行あたり**

```
 speak("sorry, I haven't been taught to understand that.")
```

⇒

```
 speak("すみません。よくわかりません。")
```

**•wave.js 195 行あたり**

```
 voice: config.voice,
```

⇒

```
 voice: 'ja-JP_EmiVoice',
```

**•wave.js 238 行あたり**

```
 speak("Sure. I am decoding a sound file that I will dance to. This may take a couple  
 of seconds.");
```

⇒

```
 speak("確かに。私は踊るサウンドファイルをデコードしています。これには数秒かかります。");
```



## 【1 1】日本語対応版プログラムの実行

```
$ cd ~/tjwave  
$ sudo node wave.js
```

```
pi@raspberrypi:~/tjwave $ sudo node wave.js  
TJは聞いてます。何か話しかけてください
```

「手をあげる」

「手を振る」

「手を下す」

「自己紹介」

「あなたの名前は？」

「ダンスできます？」

と話しかけると、手を振ったり、自己紹介したり、ダンスをしてくれます。