Text to Speech Method 1 and Method2 to Rpi3 Agentless

1. Download and install raspberry OS using Noobs. Unzip Noobs on Desktop
2. Copy and paste Noobs Files into micro SD card for Rpi3 on Desktop
3. Connect usb microphone and 3.5 mm jack
4. Connect to wifi
5. Add ssh
6. Sudo raspi-config
   1. Desktop login automatically logged in as Pi user
   2. Set locales
   3. Enable audio force 3.5 mm
7. Sudo alsamixer turn PCM to 90
8. normal tasks of setting up our Raspberry Pi. E.g expand filesystem, enable SSH, overclocking, etc...(the question is, does Noobs provide a framebuffer conf file? As the SD Image contains)
9. pi@raspberrypi ~ $ sudo raspi-config  
   pi@raspberrypi ~ $ sudo rpi-update  
   pi@raspberrypi ~ $ sudo reboot
10. **Update**
11. We now want to update our software and OS;
12. pi@raspberrypi ~ $ sudo apt-get update  
    pi@raspberrypi ~ $ sudo apt-get upgrade  
    pi@raspberrypi ~ $ sudo reboot
13. If you want X to automatically load onto PiScreen. Open up the framebuffer conf file;
14. pi@raspberrypi ~ $ sudo nano /usr/share/X11/xorg.conf.d/99-fbturbo.conf
15. Change the below line;  
    Option "fbdev" "/dev/fb0"  
    to  
    Option "fbdev" "/dev/fb1"
16. Or, if you would like to disable X from starting automatically, just run sudo raspi-config and look for the option to disable it.
17. Open /boot/config.txt
18. pi@raspberrypi ~ $ sudo nano /boot/config.txt
19. For PiScreen 2;  
    Add this line to the bottom
20. dtoverlay=piscreen2r
21. Now reboot

SETTING UP HARDWARE notes manually cont vs bootstrap.sh

**Use this method only if you have difficulties with the Bootstrap scriptor or if running an older version of Raspian (pre-Jessie). Skip this use for troubleshooting**

1. *Optional*. Disable ipv6. In some networking environments, disabling ipv6 may help your Pi get on the network.

echo " ipv6.disable=1" | sudo tee -a /boot/cmdline.txt

It is safe to skip this step. We only recommend doing this step if necessary.

1. *Optional*. Enable Quad9 DNS. In some networking environments, using Quad9's nameservers may speed up DNS queries and provide extra security and privacy..

echo "nameserver 9.9.9.9" | sudo tee -a /etc/resolv.conf

echo "nameserver 149.112.112.112" | sudo tee -a /etc/resolv.conf

It is safe to skip this step. We only recommend doing this step if necessary.

sudo apt-get update

sudo apt-get -y dist-upgrade

You’ll need to do apt-get update first because that updates the repository cache. Otherwise, apt-get dist-upgrade won't do anything because it doesn't know there is a distribution upgrade.

During the upgrade, if you are asked about replacing the lightdm.conf file with the package maintainers version, say "Y".

1. Install Node.js.

Install Node.js 6 for Raspian (Jessie)

curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash -

sudo apt-get install -y nodejs

Install Node.js 9 for Raspian (Stretch)

curl -sL https://deb.nodesource.com/setup\_9.x | sudo -E bash -

sudo apt-get install -y nodejs

Note: you will encounter problems with versions of Node.js older than 6.x.

1. Install additional software packages (Jessie only).

sudo apt-get install -y alsa-base alsa-utils libasound2-dev git pigpio

1. *Optional*. Remove outdated software packages.

sudo apt-get -y autoremove

This step removes old, outdated software from your Raspberry Pi and will free up some storage space. It is safe to skip this step.

1. *Optional*. Enable the Raspberry Pi camera.

let rp=('require-promise');

function main(params){

    if(params.action==='joke'){

        const options={

            uri:'http://api.icndb.com/jokes/random',

            json:true

        }

        return rp(options)

        .then(res=>{

            return{response:res}

        })

    }

}

Bluemix Watson Assistant UI Side console.bluemix.com

If Assistant Recognizes:

Joke

Then callout Webhooks

Parameters

Key Value

action “joke”

Return Variable:

webhook\_result\_1

Then respond:

If Assistant Recognizes Respond With

1.$webhook\_result\_1 $webhook.result\_1.value.joke

2.anything else not feeling funny?

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Clone the TJBot project from Git

.

cd ~/Desktop

git clone https://github.com/ibmtjbot/tjbot.git

1. *Optional*. Disable the audio kernel modules.

In order for the LED to work, we need to disable certain kernel modules to avoid a conflict with the built-in audio jack. If you have plugged in a speaker via HDMI, USB, or Bluetooth, this is a safe operation and you will be able to play sound and use the LED at the same time. If you plan to use the built-in audio jack, we recommend **NOT** disabling the sound kernel modules.

If you are interested in playing audio over USB, we recommend purchasing a [USB sound card](https://www.amazon.com/Virtual-Channel-Audio-Adapter-Notebook/dp/B00M3UWE3Q/)).

sudo cp ~/Desktop/tjbot/bootstrap/tjbot-blacklist-snd.conf /etc/modprobe.d/

This command assumes you have cloned the tjbot git repository to your Desktop. If you have cloned it to a different directory, be sure to update the path in the above command.

If you would like to re-enable the kernel modules for built-in audio, you can do so with the following command.

sudo rm /etc/modprobe.d/tjbot-blacklist-snd.conf

Note: you will need to reboot for these changes to take effect.

1. Reboot your Raspberry Pi.

Method 2 Next. App.js and Config.js and package.json for Raspberry Pi

Package.json

{

    "name": "vicfeb2020",

    "version": "1.0.0",

    "description": "Speech to Text",

    "main": "app.js",

    "scripts": {

      "test": "node app.js",

      "start": "node app.js"

    },

    "repository": {

      "type": "git",

      "url": ""

    },

    "keywords": [

      "Watson",

      "IBM",

      "Speech",

      "To",

      "Text",

      "STT",

      "Raspberry Pi"

    ],

    "author": "Set Pech",

    "dependencies": {

      "@ibm-watson/speech-to-text-nodejs": "^2.9.0",

      "mic": "^2.1.1",

      "watson-developer-cloud": "^2.42.0"

    }

  }

appExample2.js

const watson = require('watson-developer-cloud');

const config = require('./config');  // gets our username and password

const speech\_to\_text = watson.speech\_to\_text({

    apikey: config.apikey,

    url: config.url,

    version: 'v1'

});

const mic = require('mic');

const micParams={

  rate:44100,

  channels:2,

  debug:false,

  exitOnSilence:6

};

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

var micInputStream = micInstance.getAudioStream();

micInputStream.on('data', function(data) {

  console.log("Recieved Input Stream: " + data.length);

});

micInputStream.on('silence', function() {

  // detect silence.

});

micInstance.start();

console.log("Listening, you may speak now.");

var recognizeparams = {

  content\_type: 'audio/l16; rate=44100; channels=2',

  model: 'en-US\_BroadbandModel'  // Specify your language model here

};

var textStream = micInputStream.pipe(

  speech\_to\_text.createRecognizeStream(recognizeparams)

);

textStream.setEncoding('utf8');

textStream.on('data', function(str) {

    console.log(' ===== Speech to Text ===== : ' + str); // print each text we $

    parseText(str);

});

textStream.on('error', function(err) {

  console.log(' === Watson Speech to Text : An Error has occurred =====') ; // $

  console.log(err) ;

  console.log("Press <ctrl>+C to exit.") ;

});

function parseText(str){

  /\* You can check str here \*/

}

configExample2.js

exports.workspaceId='abc123',

exports.hasCamera=true,

exports.credentials={},

exports.credentials.assistant={

    apikey:'abc123',

    url:'https://gateway.watsonplatform.net/assistant/api/v1/workspaces/bdf8e6b/message',

};

exports.credentials.speech\_to\_text={

    apikey:'abc123',

    url:'https://stream.watsonplatform.net/speech-to-text/api',

};

exports.credentials.text\_to\_speech={

    apikey:'abc123',

    url:'https://stream.watsonplatform.net/text-to-speech/api'

};

exports.credentials.visual\_recognition={

    apikey:'abc123',

    url:'https://gateway.watsonplatform.net/visual-recognition/api'

};

//option

exports.attentionWord="guru";

exports.voice="en-US\_MichaelVoice";