# RPi Text to Speech (Speech Synthesis)

This guide shows you three easy methods of getting your Raspberry Pi to talk, and describes the pros and cons of each.

## Contents



## Why use Text to Speech?

It's very easy add to your program - just output a string to the speech function instead of the screen. You don't need an expensive/complicated LCD or monitor for your project - just use any old mp3 player loudspeaker or PC loudspeaker which you have probably got lying around - or even an earphone works well for debugging purposes too.

You could use speech output for: (i) status messages - e.g. internet connection made or IP address on a headless RPi; (ii) user interface - e.g. speak the mode selected or station name with button presses on an RPi internet radio; (iii) main functionality - e.g. tell the time and read the weather forecast on your RPi alarm clock.

## Install supporting packages

Speech output requires a few audio software packages to be installed on your RPi. They may be already there but it does no harm to try to install these listed below anyway. The installer will let you know if the package is already present on your RPi. The instructions below are based on the Raspbian distribution (August 2012).

Firstly I recommend updating your Raspbian distribution if you have not recently already done so. Speech did not work for me until I did this. This may take 30 - 60 minutes depending on your connection speed etc. To do this:

```
sudo apt-get update
sudo apt-get upgrade
```

If you do not already have sound on your RPi then you will need the alsa sound utilities:

```
sudo apt-get install alsa-utils
```

and edit the file /etc/modules using:

```
sudo nano /etc/modules
```

to have line:

```
snd_bcm2835
```

If this line is already there then leave the file as is!

Install the mplayer audio/movie player with:

```
sudo apt-get install mplayer
```

To sort out the mplayer error message, edit file /etc/mplayer/mplayer.conf using:

```
sudo nano /etc/mplayer/mplayer.conf
```

to add line

```
nolirc=yes
```

# Cepstral Text to Speech

Cepstral is a commercial Text to Speech engine that is installed on the Pi and does not require an Internet connection. The voices are higher quality than open source solutions and pricing is dependent on the use case. More information is available is their website:

https://www.cepstral.com/raspberrypi

# Festival Text to Speech

The first speech package I tried was Festival. It worked fine and produces a voice like a rough sounding robot. This may be just what you need if you are adding speech to your RPi robot project.

Install Festival with:

```
sudo apt-get install festival
```

Try out Festival with:

```
echo “Just what do you think you're doing, Dave?” | festival --tts
```

or to speak RPi’s IP address:

```
hostname -I | festival --tts
```

# Espeak Text to Speech

Espeak is a more modern speech synthesis package than Festival. It sounds clearer but does wail a little. If you are making an alien or a RPi witch then it’s the one for you! Seriously it is a good allrounder with great customisation options.

Install Espeak with:

```
sudo apt-get install espeak
```

Test Espeak with: English female voice, emphasis on capitals (-k), speaking slowly (-s) using direct text:-

```
espeak -ven+f3 -k5 -s150 "I've just picked up a fault in the AE35 unit"
```

# Google Text to Speech

Google’s Text to Speech engine is a little different to Festival and Espeak. Your text is sent to Google’s servers to generate the speech file which is then returned to your Pi and played using mplayer. This means you will need an internet connection for it to work, but the speech quality is superb.

I used used ax206geek’s bash script to access the Google Text to Speech engine (this is an updated version of that script):

Create a file speech.sh with:

```
nano speech.sh
```

Add these lines to the file and save it (in nano editor use CTRL-O writeOut)

```
#!/bin/bash
say() { local IFS=+;/usr/bin/mplayer -ao alsa -really-quiet -noconsolecontrols "http://translate.google.com/translate_tts?ie=UTF-8&client=tw-ob&q=$*&tl=en"; }
say $*
```

Add execute permissions to your script with:

```
chmod u+x speech.sh
```

Test it using:

```
./speech.sh Look Dave, I can see you're really upset about this.
```

EXTRA: Dan Fountain improved on the above script to speak any length of text (Google limits you to 100 bytes normally). His excellent easy-to-read webpage describes this at https://web.archive.org/web/20151016182428/http://danfountain.com/2013/03/raspberry-pi-text-to-speech/

P.S. This link will direct you to an archive of Dan Fountain's tutorial due to his website currently being under construction.

UPDATE: Dan Fountain's script is outdated. An up-to-date version of the script is available here (also archived): http://archive.is/OgeSS

# Pico Text to Speech

Google Android TTS engine. Very good quality speech.

```
sudo apt-get install libttspico-utils
pico2wave -w lookdave.wav "Look Dave, I can see you're really upset about this." && aplay lookdave.wav
```

You can also pipe the output by creating a symlink with the extension `.wav` to `/dev/stdout`[1].

# Recommendations

I hope this guide has given you some ideas of how you can make use of speech output in your own project. As to which speech package to recommend, Festival works well enough, Espeak is clearer and so easier to understand. Pico (Android TTS) gives very good quality and does not require any internet connection - it's got everything going for it and is the one I use nowadays.

Take a look at the Adafruit article on RPi speech synthysis (https://learn.adafruit.com/speech-synthesis-on-the-raspberry-pi/introduction) - they have some great ideas there too!

All comments/suggestions welcome! Let me know for what you have used speech on your Pi - StevenP on the official Raspberry Pi Forum.

1. Pipe output from program which only outputs to a file (https://unix.stackexchange.com/a/325020/9583)