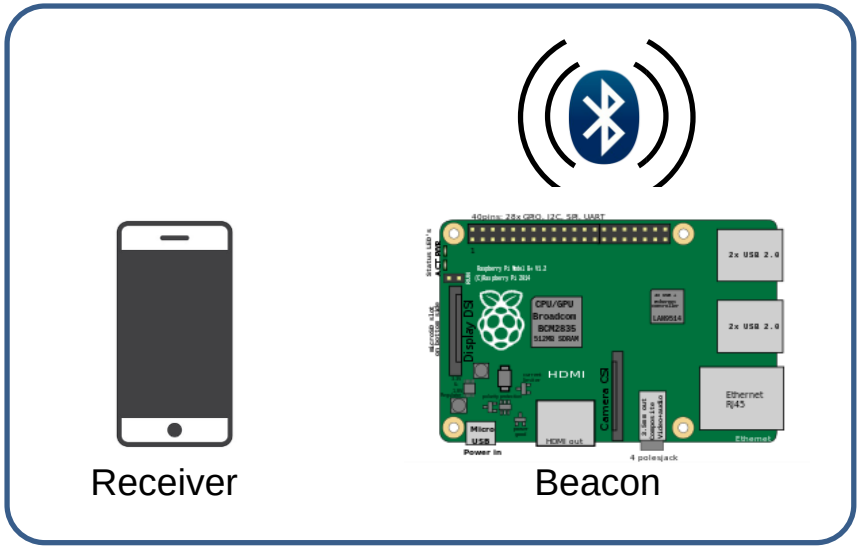
嵌入式系統設計概論與實作

曾煜棋、吳昆儒

National Yang Ming Chiao Tung University

Last week

- 嵌入式應用: BLE beacon
  - Beacon applications
  - Eddystone, iBeacon protocol



This week

- 嵌入式應用: 語音助理
  1. Mel-Frequency Cepstral Coefficients
  2. Speech to text (STT)
  3. Text to speech (TTS)

  - 語音識別 (Speech recognition)
  - 自動語音辨識 (Automatic Speech Recognition, ASR)
  - 電腦語音識別 (Computer Speech Recognition)
  - 語音轉文字識別 (Speech To Text, STT)
  - 自然語言處理 (Natural Language Processing, NLP)
    - 讓電腦擁有理解人類語言的能力

https://linux.die.net/man/1/arecord

Test and play microphone

- In terminal
  - **Check device**
    - aplay -l
    - arecord -l
  - **Record your voice**
    - arecord -D plughw:1 -f cd Filename.mp3 # use “ctrl + c” to stop recording
    - arecord -D plughw:1 -f cd -d 2 Filename.mp3 # record 2 seconds


```
pi@raspberrypi:~$ arecord -D plughw:1 -f cd Filename.mp3
arecord: main:828: audio open error: No such file or directory
pi@raspberrypi:~$ arecord  -f cd Filename.mp3
Recording WAVE 'Filename.mp3' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
^CAborted by signal Interrupt...
```

  - **Play audio**
    - omxplayer -o local -p Filename.mp3

遇到此問題時, 把"-D plughw:1"拿掉試試

Check your device

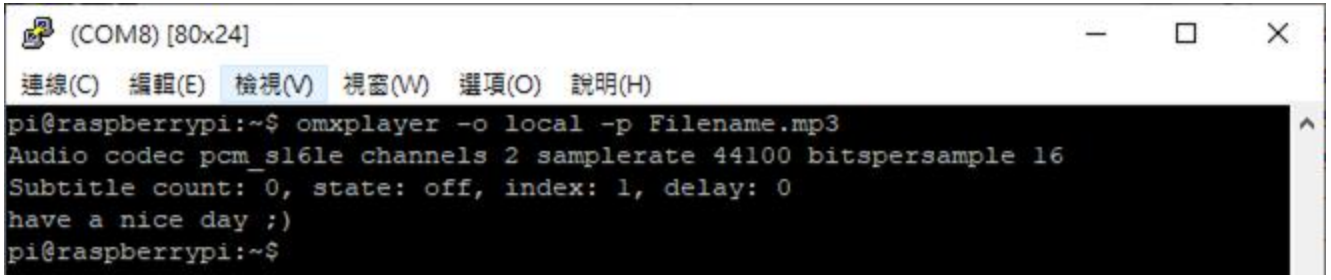
- aplay -l

- arecord -l

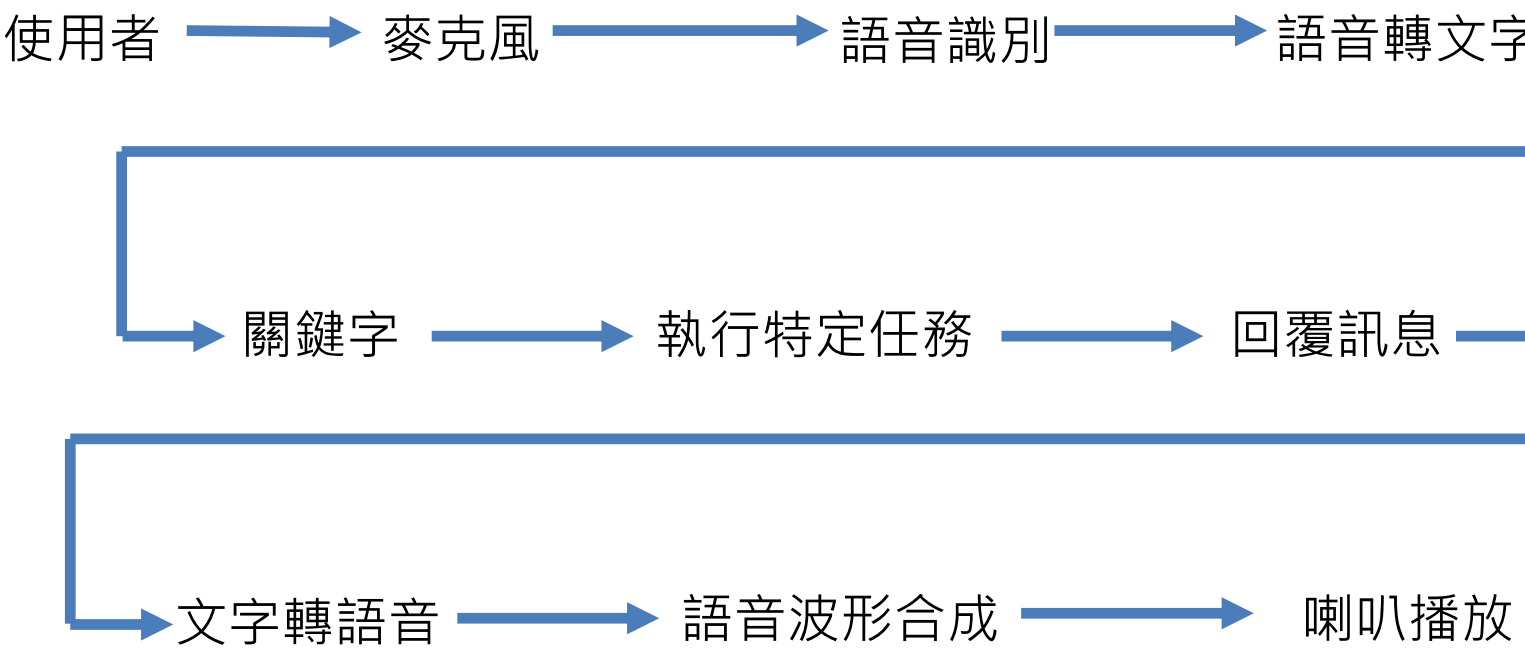
Test and play microphone

- **Record your voice**
  - arecord -D plughw:1 -f cd Filename.mp3
    - # use “ctrl + c” to stop recording or use “-d” to set duration

- **Play audio**
  - omxplayer -o local -p Filename.mp3



語音助理流程



Speech recognition
Speech to text
Text to speech

Outline

- 嵌入式應用: 語音助理
  1. Mel-Frequency Cepstral Coefficients
  2. Speech to text (STT)
  3. Text to speech (TTS)

  - 語音識別 (Speech recognition)
  - 自動語音辨識 (Automatic Speech Recognition, ASR)
  - 電腦語音識別 (Computer Speech Recognition)
  - 語音轉文字識別 (Speech To Text, STT)
  - 自然語言處理 (Natural Language Processing, NLP)
    - 讓電腦擁有理解人類語言的能力

https://zh.wikipedia.org/wiki/%E6%A2%85%E7%88%BE%E5%80%92%E9%A0%BB%E8%AD%9C
https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

Mel-Frequency Cepstral Coefficients

- MFCCs are commonly used as features in speech recognition systems, such as the systems which can automatically recognize numbers spoken into a telephone.

- MFCC(梅爾倒頻譜係數)
  1. Take the Fourier transform of a signal (with sliding window)
  2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
  3. Take the logs of the powers at each of the mel frequencies.
  4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
  5. The MFCCs are the amplitudes of the resulting spectrum.

- Application: music information retrieval
  - audio similarity measures

Dependency

- pip3 install matplotlib
- sudo apt install llvm
- sudo apt-get install libatlas-base-dev
- pip3 install numba==0.48.0
  - 會自動裝llvmlite-0.31
- pip3 install librosa==0.4.2
  - 會自動裝audioread-2.1.9 joblib-1.0.1 scikit-learn-0.24.2 scipy-1.6.3 threadpoolctl-2.1.0

- Record your voice
  - arecord -D plughw:1 -f cd -d 2 Example.mp3
  - use “ctrl + c” to stop recording or use “-d” to set duration
  - Ex: “-d 2” = 2 seconds

MFCC sample

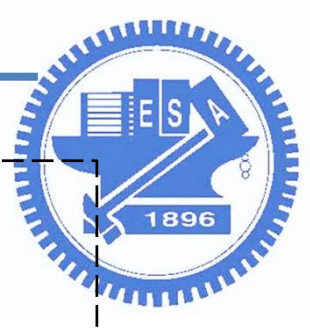
- python3 1.mfcc_fig.py
- python3 1.mfcc_normalized.py

- Error message? (for python3 1.mfcc_fig.py)

```
pi@raspberrypi:~ $ python3 1.mfcc_fig.py
/home/pi/.local/lib/python3.7/site-packages/scipy/__init__.py:140: UserWarning: A
NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected
version 1.16.2)
```

- Sol:
  - pip3 install numpy
  - pip3 install numpy --upgrade

https://librosa.github.io/librosa/generated/librosa.feature.mfcc.html
https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.figure.html



```
import librosa
import matplotlib.pyplot as plt
import numpy as np
import librosa.display

# Load an audio file as a floating point time series.
y, sr = librosa.load('Example.mp3')

# Mel-frequency cepstral coefficients (MFCCs)
mfccs = librosa.feature.mfcc(y=y, sr=sr)

print (mfccs)

plt.figure(figsize=(10, 4)) # figure with width, height in inches

# Display a spectrogram/chromagram/cqt/etc.
librosa.display.specshow(mfccs, sr=sr, x_axis='time')
plt.colorbar()
plt.title('MFCC')
plt.tight_layout()
plt.show()
```

Parameters:

y:np.ndarray [shape=(n,)] or None

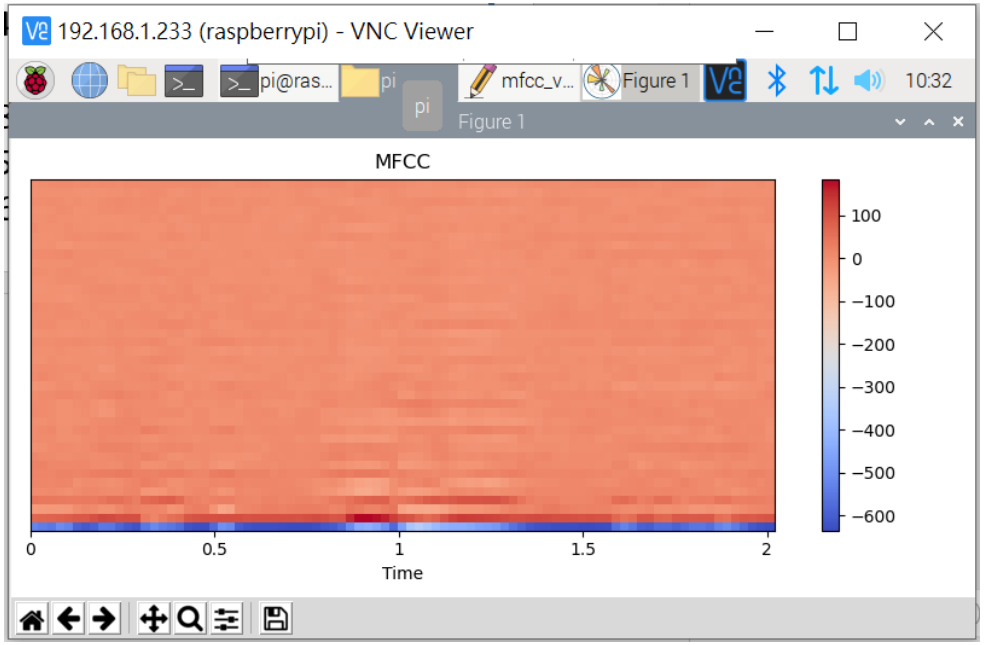
audio time series

sr:number > 0 [scalar]

sampling rate of *y*

MFCC-1

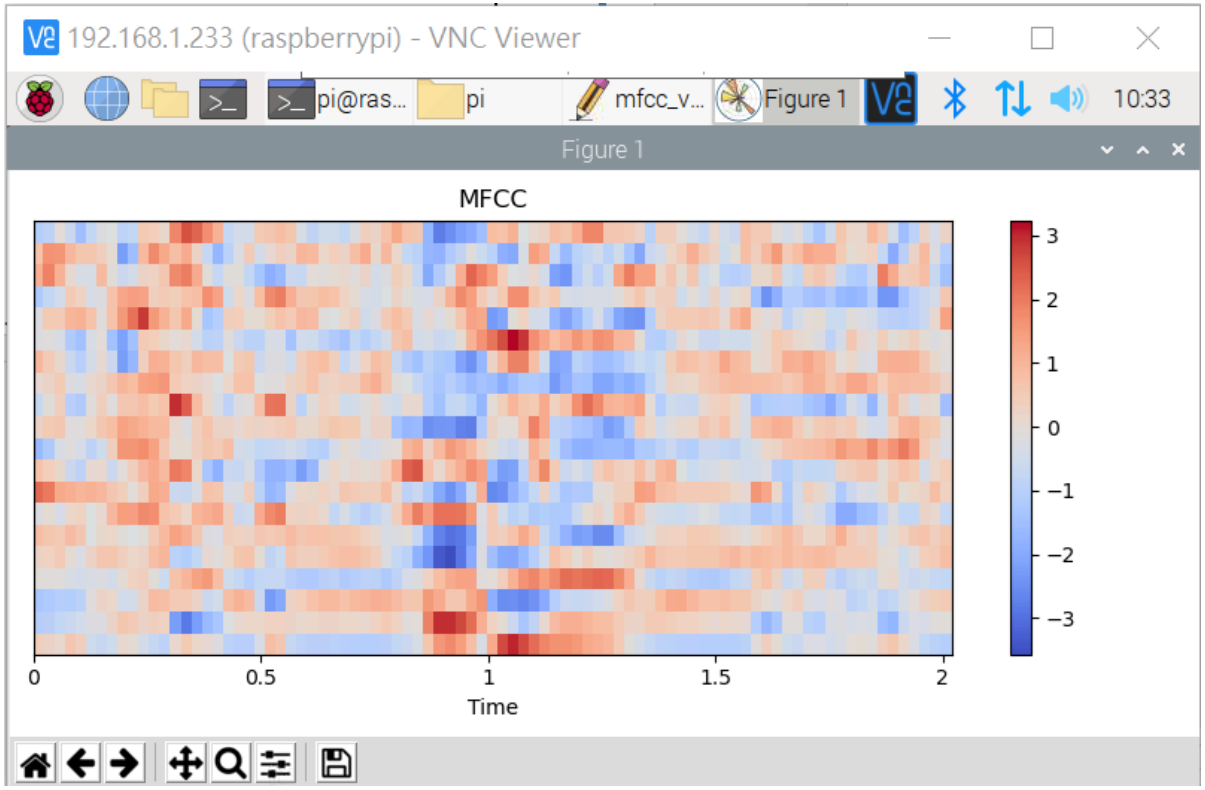
- computes MFCCs across an audio signal



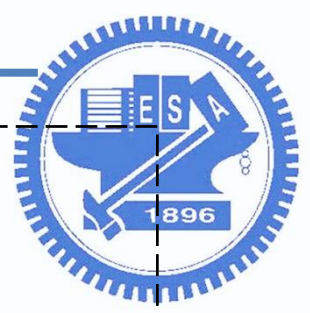

```
[[-524.0882   -539.76855  -550.3053   ... -609.8008   -621.7471   -630.12366 ]
 [  70.95046    72.8979    77.97775  ...  89.76895    96.699974   98.163864 ]
 [ -23.950603  -22.821901  -17.783375 ...  13.485247   17.420559   19.213781 ]
 ...
 [   7.9192953  11.753521   7.4945087 ...   6.2203283  -1.3843718  -5.0579705]
 [   2.2354372   6.6398587  6.363016  ...   3.404626   -1.8755012  -5.631446 ]
 [  -2.2584667  -1.39823    2.0566473 ...   3.3421202   0.806429    3.2034779]]
```

MFCC-2

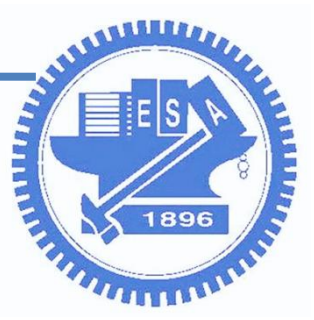
- scale the MFCCs such that each coefficient dimension has zero mean and unit variance



https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.scale.html
https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-scaler



```
import librosa
import matplotlib.pyplot as plt
import numpy as np
import librosa.display
import sklearn

y, sr = librosa.load('Example.mp3')
mfccs = librosa.feature.mfcc(y=y, sr=sr)
print (mfccs)

# Standardize a dataset along any axis
# Center to the mean and component wise scale to unit variance.
mfccs = sklearn.preprocessing.scale(mfccs, axis=1)
print (mfccs.mean(axis=1))
print (mfccs.var(axis=1))

plt.figure(figsize=(10, 4))
librosa.display.specshow(mfccs, sr=sr, x_axis='time')
plt.colorbar()
plt.title('MFCC')
plt.tight_layout()
plt.show()
```



Original MFCC:

```
[[-524.0882   -539.76855  -550.3053   ... -609.8008   -621.7471  -630.12366 ]
 [  70.95046    72.8979    77.97775  ...  89.76895    96.699974   98.163864 ]
 [ -23.950603  -22.821901  -17.783375 ...  13.485247   17.420559   19.213781 ]
 ...
 [   7.9192953  11.753521    7.4945087 ...   6.2203283  -1.3843718  -5.0579705]
 [   2.2354372   6.6398587   6.363016  ...   3.404626   -1.8755012  -5.631446 ]
 [  -2.2584667  -1.39823     2.0566473 ...   3.3421202   0.806429    3.2034779]]
```

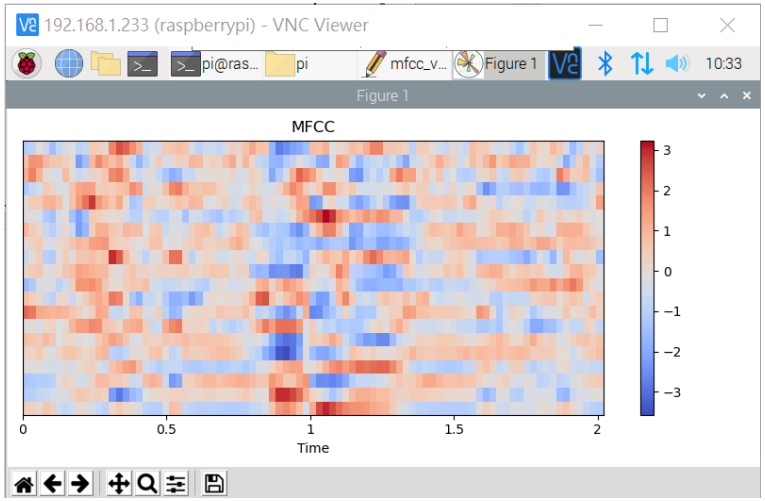
Normalized MFCC:

```
[[ 0.43912765  0.2216706   0.07554583 ... -0.7495444  -0.9152169  -1.0313843 ]
 [-1.0342532  -0.96486586 -0.78387076 ... -0.36375016 -0.11679772  -
0.06463929]
 [-1.2954026  -1.2480938  -1.0369073  ...  0.27369633  0.43864232   0.5138041 ]
 ...
 [ 1.149406    1.9089625   1.0652562  ...  0.81284225 -0.69364154  -1.4213779 ]
 [ 0.694803    1.6624304   1.6016096  ...  0.9516675  -0.20834756  -1.033508  ]
 [-1.1192613  -0.9052428  -0.04570303 ...  0.27411008 -0.35674512
0.23961723]]
```

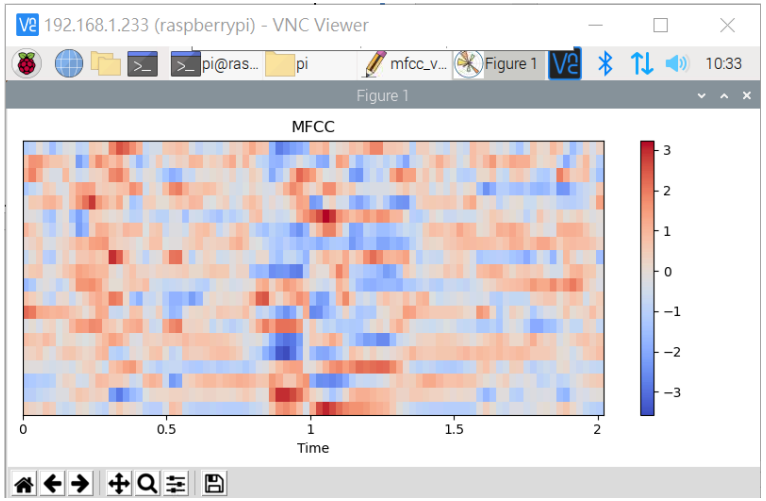
Discussion 1

- Try to say the same sentence twice, then plot the MFCC for both result.

Example:



This is a book.



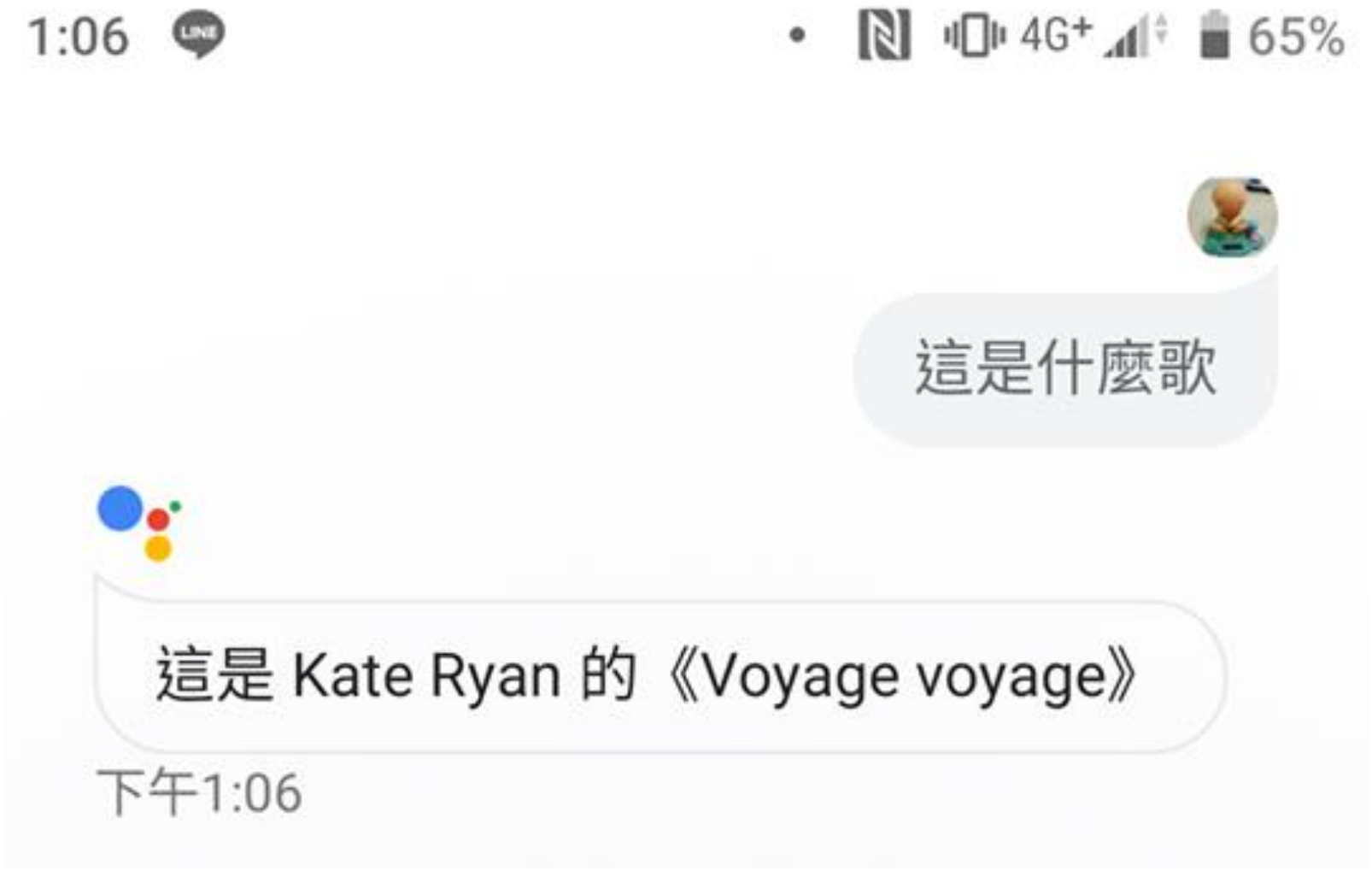
This is a book.

Outline

- 嵌入式應用: 語音助理
  1. Mel-Frequency Cepstral Coefficients
  2. Speech to text (STT)
  3. Text to speech (TTS)

  - 語音識別 (Speech recognition)
  - 自動語音辨識 (Automatic Speech Recognition, ASR)
  - 電腦語音識別 (Computer Speech Recognition)
  - 語音轉文字識別 (Speech To Text, STT)
  - 自然語言處理 (Natural Language Processing, NLP)
    - 讓電腦擁有理解人類語言的能力

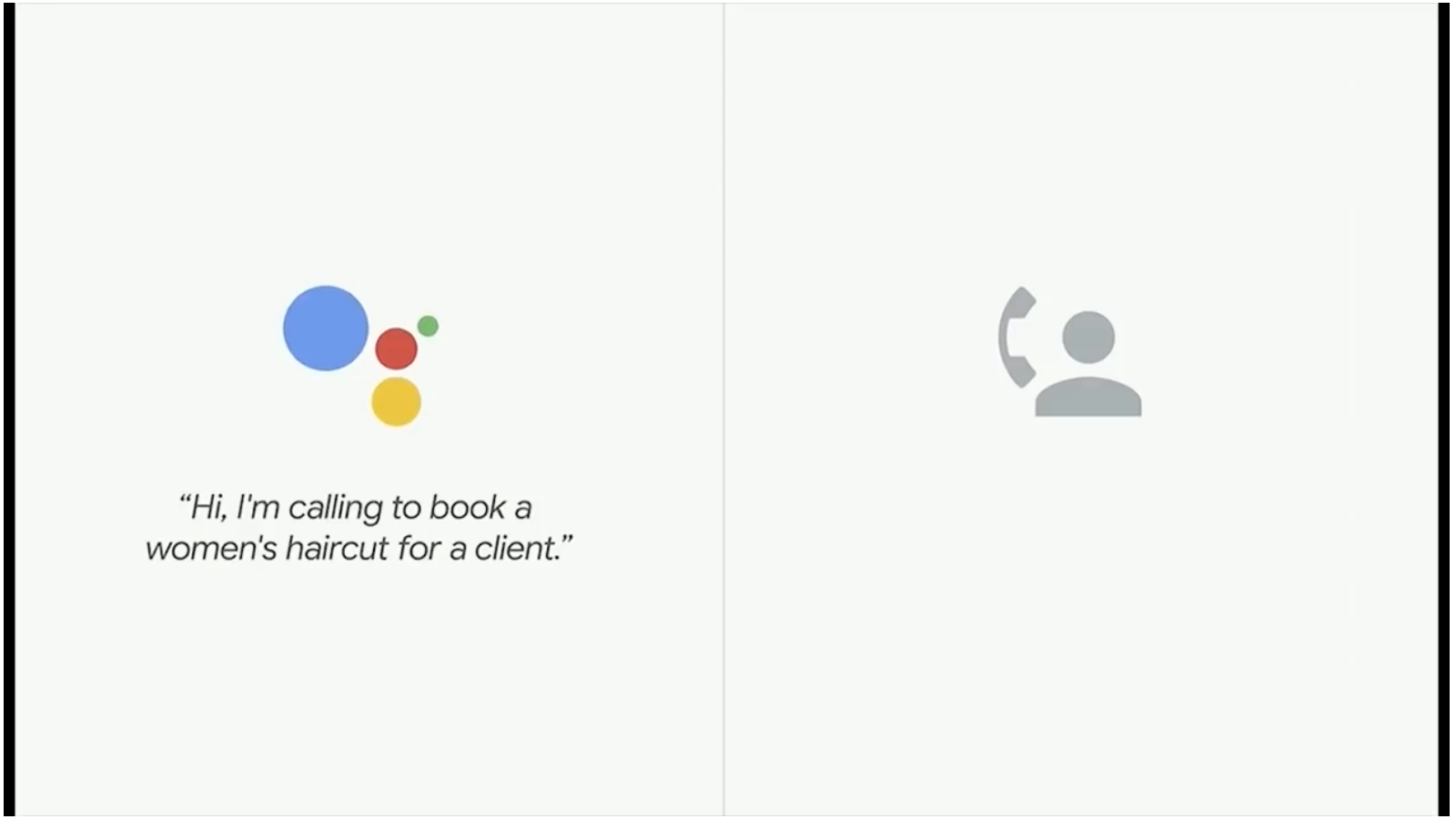
Google assistant



https://www.youtube.com/watch?v=7gh6_U7Nfjs

Google assistant

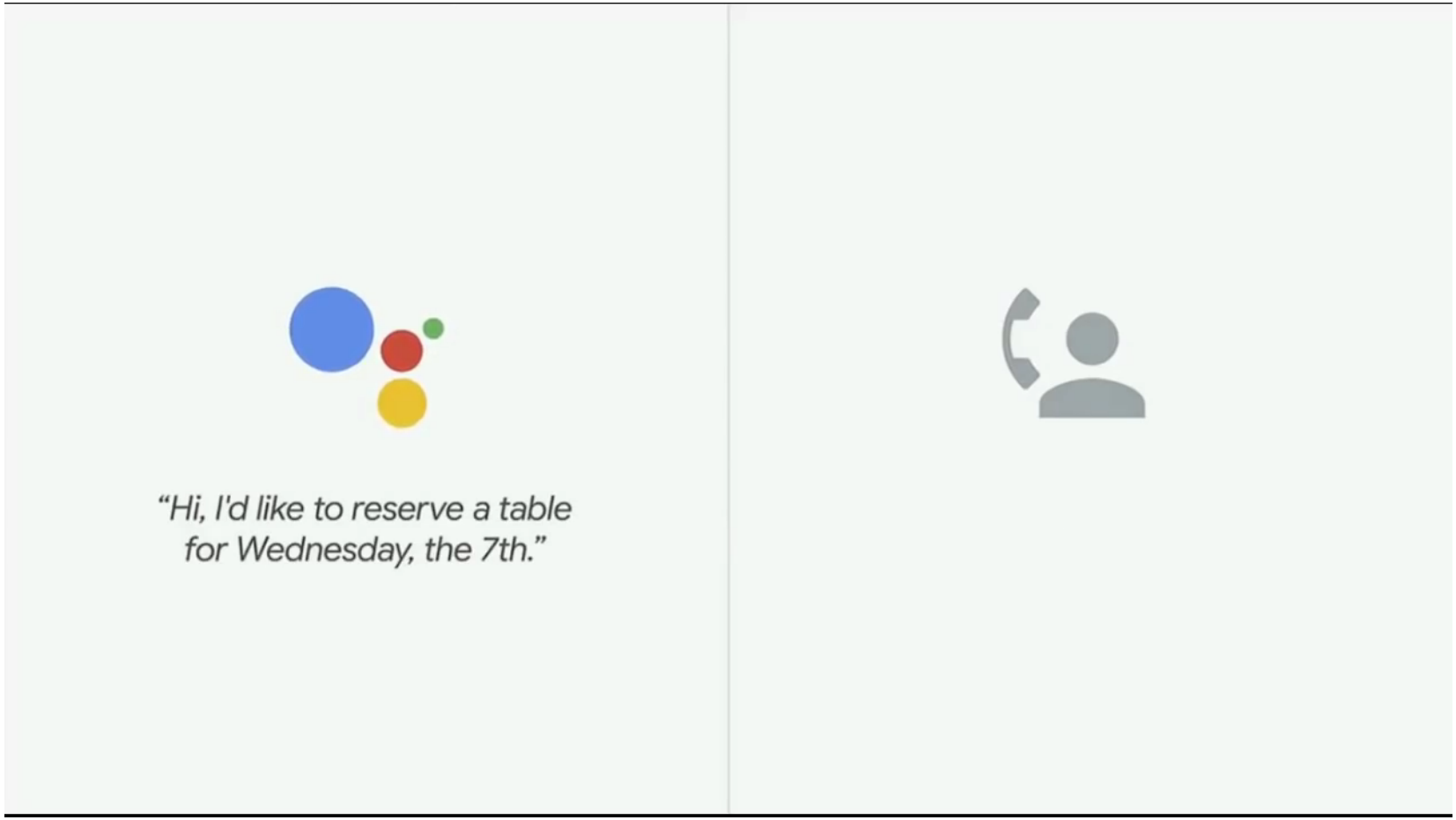
- Google Assistant will soon be able to call restaurants and make a reservation for you (2018/5/9)



https://www.youtube.com/watch?v=-RHG5DFAjp8

Google assistant

- Google Assistant calling a restaurant for a reservation

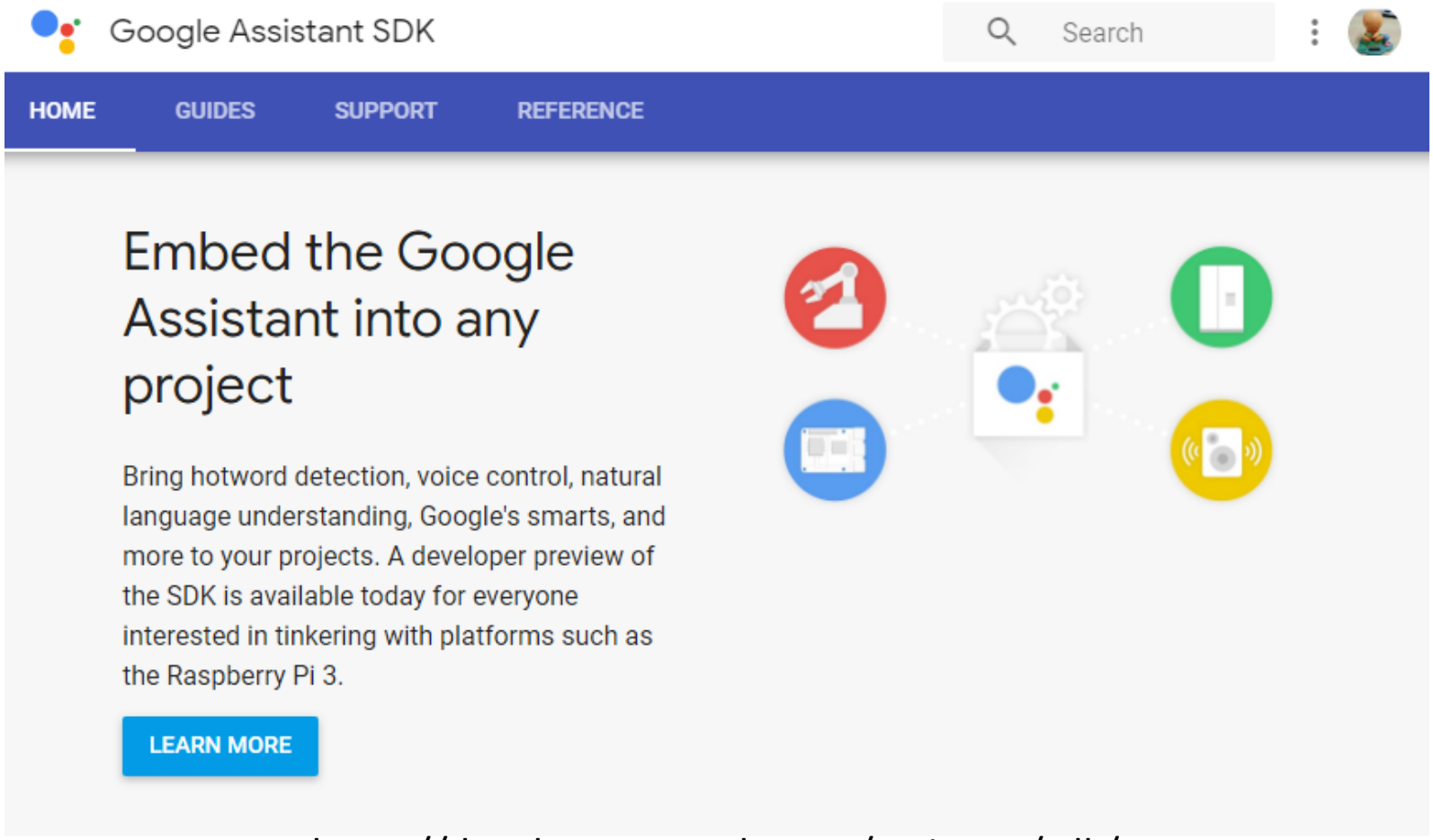


https://www.youtube.com/watch?v=3kODsHcrs2c

Google assistant

- Google IO 2019 Next Gen Google Assistant (2019/5/7)

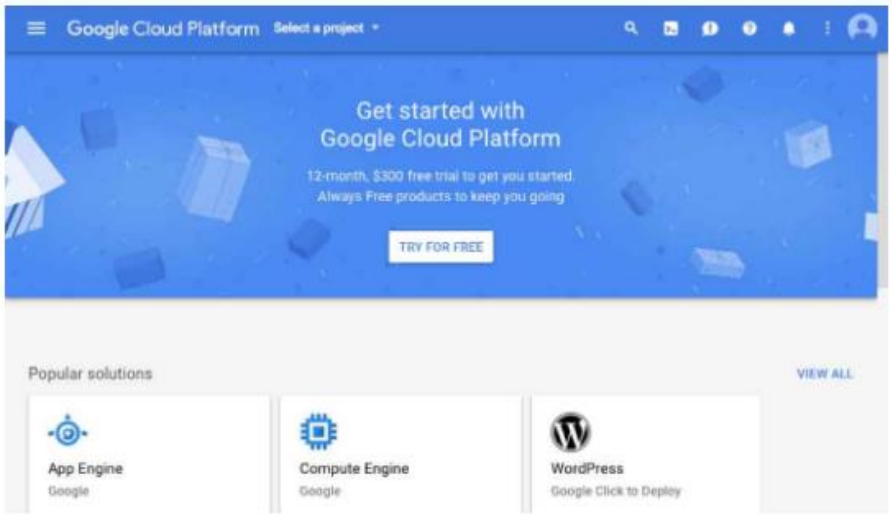
Google assistant SDK

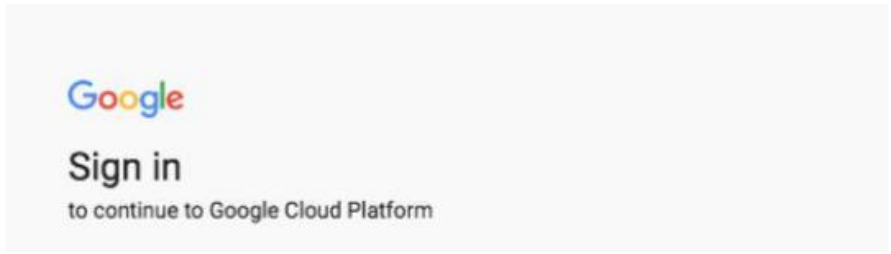


https://developers.google.com/assistant/sdk/

Voice Kit

- Do-it-yourself intelligent speaker. Experiment with voice recognition and the Google Assistant.





1. Head to the Google Cloud Platform

In order to make use of the Google Assistant and Cloud Speech APIs, you need to get credentials from Google's developer console.

On your computer (not the Raspberry Pi), go to https://console.cloud.google.com/.

2. Login using your Google account

Log in using your Google account.

Don't have a Google account? Sign up for one here.

https://aiyprojects.withgoogle.com/voice/

https://azure.microsoft.com/zh-tw/services/cognitive-services/speech-services/

Azure

語音轉換文字 - 將語音轉換成文字以取得直覺式互動

輕鬆將即時語音轉換文字的功能新增到您的應用程式之中，以應用在語音命令、即時轉譯、自動會議記錄，或是話務中心的記錄分析等等。

調整您的語音辨識模型，以適應使用者的說話方式、措辭或獨特詞彙，並根據您的情況配合特定背景雜音、口音和聲紋。

深入了解 >

文字轉換語音 - 為您的應用程式提供自然語音

建置智慧型應用程式和服務，使用文字轉換語音服務自然地與使用者交談。近乎即時地將文字轉換成音訊，並根據說話速度、音調、音量等變化進行調整。

使用自訂語音模型，為您的應用程式提供獨特且可辨識的品牌語音。只要錄製並上傳定型資料，服務就會建立專為您的錄音調整的獨特語音效果。

深入了解 >

語音翻譯

為您的應用程式提供任何支援語言的即時語音翻譯功能，並接收文字或語音翻譯。語音翻譯模型是以尖端語音辨識和神經機器翻譯系統 (NMT) 技術為基礎。這些模型已經過最佳化，能夠理解人們在真實生活中的說話方式，並產生絕佳品質的翻譯。

深入了解 >

https://pypi.org/project/SpeechRecognition/

SpeechRecognition

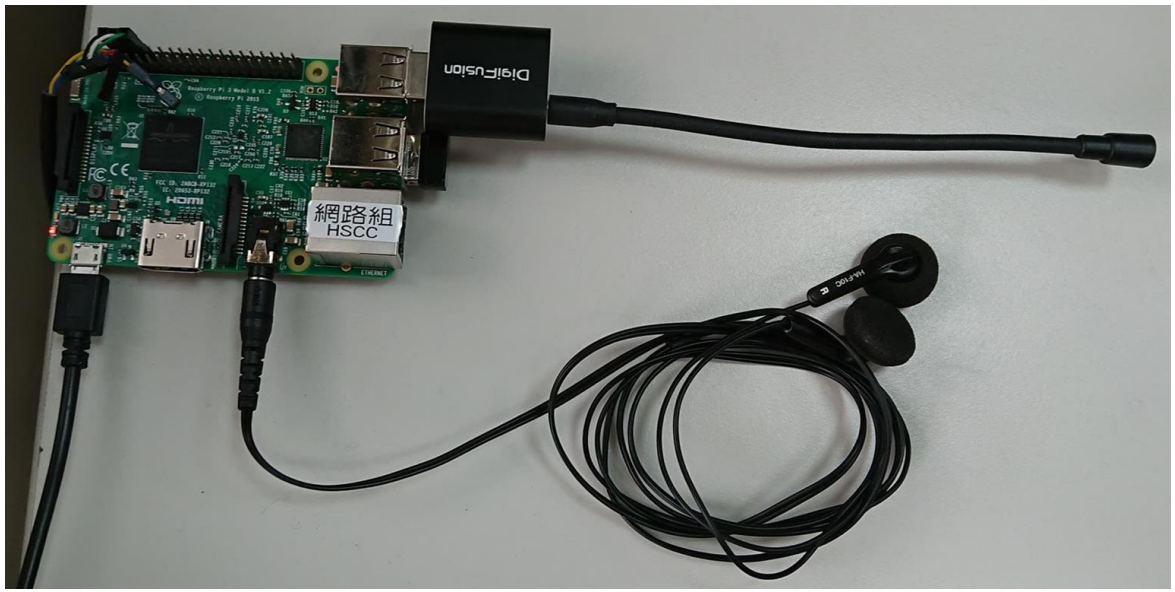
- Library for performing speech recognition, with support for several engines and APIs, online and offline.

- Speech recognition engine/API support:
  - CMU Sphinx (works offline) (卡內基大學)
  - **Google Speech Recognition**
  - **Google Cloud Speech API**
  - Wit.ai (Facebook, Messenger ChatBot)
  - Microsoft Bing Voice Recognition
  - Houndify API (SoundHound,音樂識別平台)
  - IBM Speech to Text
  - Snowboy Hotword Detection (works offline)

More examples: https://github.com/Uberi/speech_recognition/tree/master/examples

Dependency

- pip3 install SpeechRecognition
- pip3 install gTTS
- sudo apt-get install libasound2-dev
- sudo apt-get install python3-pyaudio
- sudo apt-get install flac



Outline

- 嵌入式應用: 語音助理
  1. Mel-Frequency Cepstral Coefficients
  2. Speech to text (STT)
  3. Text to speech (TTS)

  - 語音識別 (Speech recognition)
  - 自動語音辨識 (Automatic Speech Recognition, ASR)
  - 電腦語音識別 (Computer Speech Recognition)
  - 語音轉文字識別 (Speech To Text, STT)
  - 自然語言處理 (Natural Language Processing, NLP)
    - 讓電腦擁有理解人類語言的能力

sudo python3 2.stt_microphone.py

Speech to text (microphone)

```
import speech_recognition as sr

#obtain audio from the microphone
r=sr.Recognizer()

with sr.Microphone() as source:
    print("Please wait. Calibrating microphone...")
    #listen for 1 seconds and create the ambient noise energy level
    r.adjust_for_ambient_noise(source, duration=1)
    print("Say something!")
    audio=r.listen(source)

# recognize speech using Google Speech Recognition
try:
    print("Google Speech Recognition thinks you said:")
    print(r.recognize_google(audio))
except sr.UnknownValueError:
    print("Google Speech Recognition could not understand audio")
except sr.RequestError as e:
    print("No response from Google Speech Recognition service: {0}".format(e))
```

sudo python3 2.stt_file.py

Speech to text (audio file)

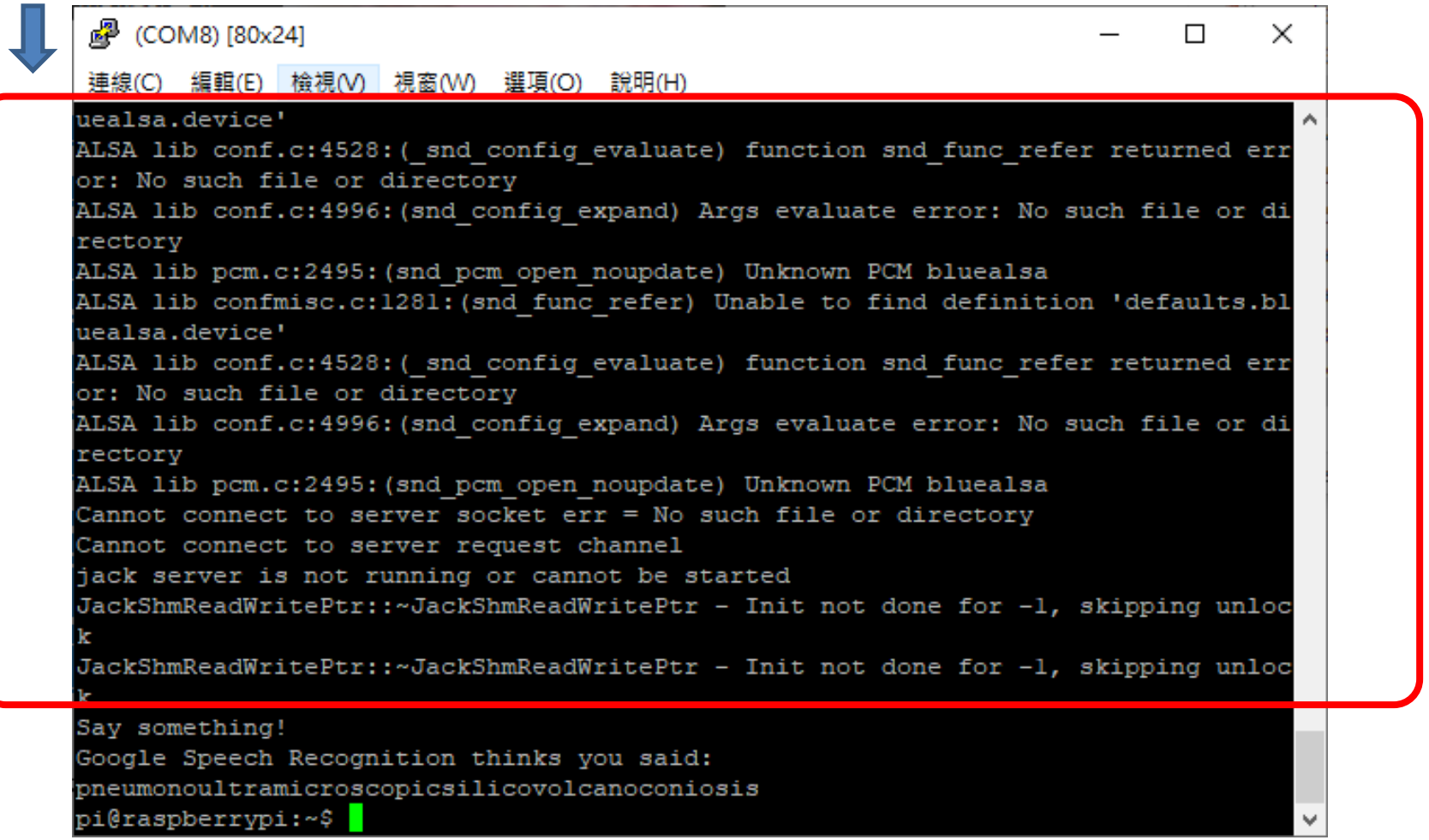
```
import speech_recognition as sr

#obtain audio from the microphone
r=sr.Recognizer()

myvoice = sr.AudioFile('hello.flac')     # not mp3, not mp3, not mp3!!
with myvoice as source:
   print("Use audio file as input!")
   audio = r.record(source)

# recognize speech using Google Speech Recognition
try:
   print("Google Speech Recognition thinks you said:")
   print(r.recognize_google(audio))
except sr.UnknownValueError:
   print("Google Speech Recognition could not understand audio")
except sr.RequestError as e:
   print("No response from Google Speech Recognition service: {0}".format(e))
```

Input format: PCM WAV, AIFF/AIFF-C, or Native FLAC

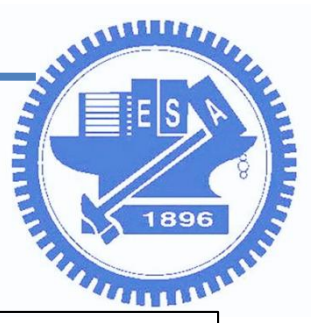
You might need: ffmpeg -i input.mp3 output.flac

Speech to text (result)

You can ignore the ALSA warning messages



https://pypi.org/project/SpeechRecognition/
https://github.com/Uberi/speech_recognition/blob/master/examples/audio_transcribe.py



SpeechRecognition

Speech recognition engine/API support:

- CMU Sphinx (works offline)
- **Google Speech Recognition**
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text

- r.recognize_sphinx(audio)
- **r.recognize_google(audio)**
- r.recognize_google_cloud(audio, credentials_json=GOOGLE_CLOUD_SPEECH_CREDENTIALS)
- r.recognize_wit(audio, key=WIT_AI_KEY)
- r.recognize_azure(audio, key=AZURE_SPEECH_KEY)
- r.recognize_bing(audio, key=BING_KEY)
- r.recognize_houndify(audio, client_id=HOUNDIFY_CLIENT_ID, client_key=HOUNDIFY_CLIENT_KEY)
- r.recognize_ibm(audio, username=IBM_USERNAME, password=IBM_PASSWORD)

Speech Recognition Library Reference
https://github.com/Uberi/speech_recognition/blob/master/reference/library-reference.rst

Outline

- 嵌入式應用: 語音助理
  1. Mel-Frequency Cepstral Coefficients
  2. Speech to text (STT)
  3. Text to speech (TTS)

  - 語音識別 (Speech recognition)
  - 自動語音辨識 (Automatic Speech Recognition, ASR)
  - 電腦語音識別 (Computer Speech Recognition)
  - 語音轉文字識別 (Speech To Text, STT)
  - 自然語言處理 (Natural Language Processing, NLP)
    - 讓電腦擁有理解人類語言的能力

sudo python3 3.tts.py

Text to speech

```
from gtts import gTTS
import os

tts = gTTS(text='hello', lang='en')
tts.save('hello.mp3')

os.system('omxplayer -o local -p hello.mp3 > /dev/null 2>&1')
```

The output format is mp3!

Parameters:

- **text** (*string*) – The text to be read.
- **lang** (*string, optional*) – The language (IETF language tag) to read the text in. Defaults to 'en'.
- **slow** (*bool, optional*) – Reads text more slowly. Defaults to `False`.
- **lang_check** (*bool, optional*) – Strictly enforce an existing `lang`, to catch a language error early. If set to `True`, a `ValueError` is raised if `lang` doesn't exist. Default is `True`.

https://pypi.org/project/gTTS/
https://gtts.readthedocs.io/en/latest/

gTTS (Google Text-to-Speech)

- An interface to Google Translator's Text-to-Speech API.

gTTS (`gtts.gTTS`)

class **gtts.tts.gTTS**(*text, lang='en', slow=False, lang_check=True, pre_processor_funcs=[<function tone_marks>, <function end_of_line>, <function abbreviations>, <function word_sub>], tokenizer_func=<bound method Tokenizer.run of re.compile('(?<=\?).|(?<=\!).|(?<=\ ؟).|(?<=\ ！).|(?<!\.[a-z])\. |(?<!\.[a-z])\, |(?<!\d)\:|\;|\—|\ ، |\...|\ ：|\n|\¿|\(|\ ، |\[|\)|\¡|\،|\..|\]|\ ۔', re.IGNORECASE) from: [<function tone_marks>, <function period_comma>, <function colon>, <function other_punctuation>]>*) [source]

Parameters:

- **text** (*string*) – The text to be read.
- **lang** (*string, optional*) – The language (IETF language tag) to read the text in. Defaults to 'en'.
- **slow** (*bool, optional*) – Reads text more slowly. Defaults to `False`.
- **lang_check** (*bool, optional*) – Strictly enforce an existing `lang`, to catch a language error early. If set to `True`, a `ValueError` is raised if `lang` doesn't exist. Default is `True`.

Discussion 2

- gTTS: An interface to Google Translator's Text-to-Speech API.
- how to make gTTS speak other language?

gTTS (`gtts.gTTS`)

class **gtts.tts.gTTS**(*text, lang='en', slow=False, lang_check=True, pre_processor_funcs=[<function tone_marks>, <function end_of_line>, <function abbreviations>, <function word_sub>], tokenizer_func=<bound method Tokenizer.run of re.compile('(?<=\?).|(?<=\!).|(?<=\ ?).|(?<=\ !).|(?<!\.[a-z])\. |(?<!\.[a-z])\, |(?<!\d)\:|\;|\—|\ ، |\...|\ ، |\n|\¿|\(|\ ، |\[|\)|\¡|\؟|\..|\]|\ ۔', re.IGNORECASE) from: [<function tone_marks>, <function period_comma>, <function colon>, <function other_punctuation>]>*) [source]

Quiz 1

- Say a specific command to Raspberry PI, it will start to measure the temperature and humidity.
  - Input could be microphone or audio file
    - gTTS can be used to generate audio file

```
Say something!
Google Speech Recognition thinks you said:
pneumonoultramicroscopicsilicovolcanoconiosis
pi@raspberrypi:~$
```

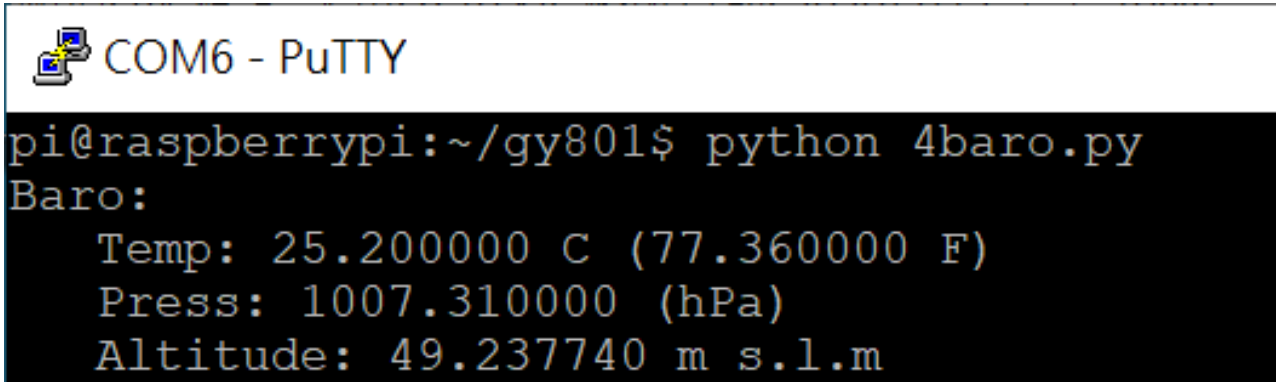
```
pi@raspberrypi ~ $ cd Adafruit_Python_DHT/examples/
pi@raspberrypi ~/Adafruit_Python_DHT/examples $ sudo ./AdafruitDHT.py 11 4
Temp=26.0*  Humidity=37.0%
```

Quiz 1

- Input
  - **Microphone**: talk to microphone directly
  - **Audio file**: recode your voice on PC, then send it to Raspberry PI. The file should be PCM WAV, AIFF/AIFF-C, or Native FLAC.
  - **gTTS**: generate the audio file from text
    - The default output is mp3. Use the following command to convert
      - ffmpeg -i input.mp3 output.flac

Quiz 2

- After measuring temperature, use gTTS (Google Text-to-Speech) to **speak out the result**.
  - Ex: the temperate is 26 degree



IMU (BMP085) can provide temperature!

Summary

- Practice Lab (MFCC, STT and TTS)
- Write down the answer for discussion
  - **Discussion (**Deadline: Before 5/21, 12:00)
    - **1. plot MFCC**
    - **2. how to make gTTS speak other language?**
- Demonstrate **Quiz 1** and **Quiz 2** to TAs
  - **Quiz1: Say command to execute task**
  - **Quiz2: Speak out the task result**
    - **You can combine quiz1 and quiz2 together.**
    - Deadline: Before 5/14, 15:10
    - Late Demo: Before 5/21, 15:10