

Node Red Dashboard

Στο rasbian stretch είναι εγκατεστημένο το Node-Red V 0.19.4 και για να βάλουμε το dashboard κάνουμε τις εξής ενέργειες:

```
sudo apt-get update
sudo apt-get install npm
cd ~/.node-red
sudo npm i -g npm@2.x
npm install node-red-dashboard
```

Σταματάμε το node-red με **node-red-stop** και το ξεκινάμε με **node-red-start**.

Mosquitto

```
sudo apt install -y mosquitto mosquitto-clients
```

Ενεργοποιούμε το service:

```
sudo systemctl enable mosquitto.service
```

Αν θέλουμε να απενεργοποιήσουμε το service γράφουμε:

```
sudo systemctl stop mosquitto.service
sudo systemctl disable mosquitto.service
```

Έλεγχος έκδοσης

```
mosquitto -v
```

Επιστρέφει 1.4.10

Αν δεν βάλουμε το service και θέλουμε να τρέχει ως δαίμονας στο background γράφουμε:

```
mosquitto -d
```

Συνδρομή σε Topic

```
mosquitto_sub -d -t testTopic
```

Δημοσίευση στο Topic

Ανοίγουμε ένα ακόμη τερματικό και γράφουμε:

```
mosquitto_pub -d -t testTopic -m "Hello world!"
```

Τότε στο πρώτο τερματικό θα δούμε:

```
Client mosqsub/21899-smart_ras sending PINGREQ
Client mosqsub/21899-smart_ras received PINGRESP
Client mosqsub/21899-smart_ras received PUBLISH (d0, q0, r0, m0,
'testTopic', ... (12 bytes))
Hello world!
```

Μπορούμε να ανοίξουμε και τρίτο τερματικό κάνοντας συνδρομή στο ίδιο topic. Αν από το 2ο τερματικό κάνουμε δημοσίευση, τότε αυτή θα εμφανιστεί στα τερματικά 1 και 3.

Voice recognition Google

Για Raspberry PI:

```
sudo apt-get install libportaudio0 libportaudio2 libportaudiocpp0 portaudio19-dev
```

Για Debian i386:

```
sudo apt-get install libportaudio2 libportaudiocpp0 portaudio19-dev
```

```
sudo apt-get install flac
```

Για Python 2.x

```
pip install pyaudio
```

Για Python 3.x

```
pip3 install pyaudio
```

```
pip3 install SpeechRecognition
```

Αρχικά φτιάχνω ένα πρόγραμμα με όνομα test.py το οποίο εμφανίζει τις συσκευές ηχογράφησης:

```
import speech_recognition as sr
for index, name in enumerate(sr.Microphone.list_microphone_names()):
    print("Microphone with name \"{1}\" found for
`Microphone(device_index={0})`".format(index, name))
```

Με το alsamixer ρυθμίζω επίπεδα για capture. Από το audacity ελέγχω την ένταση και το ποσοστό του θορύβου.

Text To Speech (T.T.S)

```
import os, subprocess, urllib

def getGoogleSpeechURL(phrase):
    googleTranslateURL = "http://translate.google.com/translate_tts?ie=UTF-8&client=tw-ob&tl=e1&"
    parameters = {'q': phrase}
    data = urllib.parse.urlencode(parameters)
    googleTranslateURL = "%s%s" % (googleTranslateURL, data)
    return googleTranslateURL

def speakSpeechFromText(phrase):
    googleSpeechURL = getGoogleSpeechURL(phrase)
    print(googleSpeechURL)
    #Raspberry PI
    #subprocess.call(["mplayer", googleSpeechURL], shell=False, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    #Debian i386
    subprocess.Popen(["mplayer", "-really-quiet", googleSpeechURL], shell=False, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)

speakSpeechFromText("Καλημέρα παιδιά. Τι κάνετε;")
```