

Assistente vocale Mycroft

Installazione

- Scaricare <https://mycroft.ai/to/picroft-image>
- Estrarre lo zip
- Flashare il file .img sulla sd del raspberry (si può usare balena etcher)
- Aprire terminale nella partizione rootfs della scheda sd:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Aggiungere in fondo al file le credenziali della rete wifi:

```
network={
    ssid="nome rete"
    psk="password"
}
```

- Salvare e chiudere
- Mettere la scheda sd nel raspberry e avviarlo
- Connettersi via ssh al raspberry:

```
ssh pi@192.168.0.x
```

La password di default è "mycroft"

- Saltare la configurazione guidata
- Fare il pairing del device su <https://account.mycroft.ai/devices> o su un server locale
- Al termine dell'installazione delle skill, chiudere con ctrl+c

Aggiornamento della configurazione

Dal terminale del raspberry:

```
cd ~
rm .mycroft/mycroft.conf
rm .mycroft_cli.conf
sudo apt update -y
sudo apt upgrade -y
sudo apt autoremove -y
pip install --upgrade pip
cd mycroft-core
pip install --upgrade .
```

Impostare lingua italiana

```
nano .config/mycroft/mycroft.conf
```

Aggiungere:

```
{
  ...
  "lang": "it-it",
  "system_unit": "metric",
  "time_format": "full",
  "date_format": "DMY"
  // TODO aggiungere "lang": "it-it" al componente tts
}
```

Installazione della scheda audio

Scheda audio modello ReSpeaker 2 Mics Pi HAT (seeedstudio o keystudio)

- Inserire la scheda respeaker 2-mic nella socket del raspberry
- Collegare un altoparlante all'uscita speaker della scheda
- Dal terminale del raspberry:

```
deactivate
git clone https://github.com/respeaker/seeed-voicecard
cd seeed-voicecard
sudo ./install.sh
```

Disattivare la scheda audio integrata:

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

Scrivere in fondo al file:

```
blacklist snd_bcm2835
```

Chiudere e salvare

Impostare il sample-rate utilizzato da mycroft

```
sudo nano /etc/pulse/daemon.conf
```

Scrivere in fondo al file:

```
default-sample-rate = 48000
alternate-sample-rate = 44100
```

- Chiudere e salvare

```
sudo reboot
```

Test scheda audio

Dal terminale del raspberry

```
mycroft-stop
systemctl --user stop pulseaudio

arecord -f cd -Dhw:0 test.wav

// Registrare audio, poi premere ctrl+c

aplay -Dhw:0 test.wav

// Verificare che venga riprodotto l'audio registrato
```

Regolazione del mixer

Backup della configurazione iniziale:

```
cd /etc/voicecard
sudo cp wm8960_asound.state wm8960_asound.state.save
alsamixer
```

- Premere f6 e selezionare seed-2mic-voicecard
- Premere f5 per mostrare anche il livello del microfono
- Sistemare i volumi (principalmente playback e capture) usando le frecce
- Rendere le modifiche permanenti con:

```
sudo alsactl -f /etc/voicecard/wm8960_asound.state store
sudo reboot
```

Aprire alsamixer e verificare che le modifiche non siano state cancellate

Nota: il volume di master viene regolato direttamente da mycroft

Skill homeassistant

Dal terminale del raspberry:

```
pip install quantulum3
mycroft-msm install homeassistant
cd ~/mycroft-core/skills/homeassistant.mycroftai
git pull
```

- Creare un long lived access token da home assistant
- Andare su <https://home.mycroft.ai/skills> (o server locale) e inserire ip di home assistant e token

Addestramento di una rete neurale

ATTENZIONE

Usa questa parte di guida solo se vuoi addestrare manualmente una rete neurale DeepSpeech. Se vuoi usare la rete neurale DeepSpeech italiana già addestrata, salta questa parte. A breve caricherò una versione addestrata della rete neurale aggiornata con il nuovo dataset

CommonVoice di mozilla.

Requisiti: PC linux con scheda grafica nvidia

Verificare se i driver nvidia sono installati:

```
nvidia-smi
```

Questo comando dovrebbe mostrare una tabella con alcune proprietà della gpu

Installare docker e nvidia-docker: seguire una guida specifica per il proprio sistema operativo

<https://docs.docker.com/engine/install/>

<https://github.com/NVIDIA/nvidia-docker>

Prima di proseguire è consigliato fare un test per assicurarsi che docker riesca a vedere la gpu

Scaricare il dataset più recente di CommonVoice

<https://commonvoice.mozilla.org/it/datasets>

Dovrebbe essere un file chiamato it.tar.gz

Estraetelo, rinominate la cartella "it" in "cv-it", e ricomprimetelo in "it.tar.gz".

Posizionate l'archivio nella home e lanciate questi comandi:

```
cd $HOME
mkdir -p data/sources
chmod a+rwx -R data
mv it.tar.gz data/sources
chmod a+r data/sources/it.tar.gz
```

Creare l'immagine Docker:

```
cd $HOME
git clone MozillaItalia/DeepSpeech-Italian-Model.git
cd DeepSpeech-Italian-Model/DeepSpeech
```

Nella versione di DeepSpeech che stiamo usando sono presenti alcuni bug. Possiamo correggerli cambiando il file "generate_base_dockerfile.sh" con una versione modificata che trovate qui:

https://github.com/danieltinazzi/DeepSpeech-Italian-Model/blob/master/DeepSpeech/generate_base_dockerfile.sh

```
chmod +x generate_base_dockerfile.sh
./generate_base_dockerfile.sh

docker build . -f Dockerfile.train -t deepspeech/base:0.9.3
docker build . -f Dockerfile_it.train -t deepspeech/it
```

Avviare l'addestramento

```
docker run -e "TRANSFER_LEARNING=1" --rm --gpus all --mount
```

```
type=bind,src=$HOME/data,dst=/mnt -it deepspeech/it
```

Se non avete un supercomputer potrebbe essere necessario modificare il BATCH_SIZE

Funzionante sul mio laptop con gpu 4gb:

```
docker run -e "BATCH_SIZE=32" -e "TRANSFER_LEARNING=1" --rm --gpus all --mount type=bind,src=$HOME/data,dst=/mnt -it deepspeech/it
```

Work in progress.

Installazione DeepSpeech (pre addestrato)

DeepSpeech è il motore STT di Mozilla

Dal terminale del raspberry:

```
sudo apt update
sudo apt install gfortran libatlas-base-dev libpcap-dev libpq-dev

pip install deepspeech==0.9.3 scipy av
mkdir deepspeech
cd deepspeech
curl -LO
https://github.com/MozillaItalia/DeepSpeech-Italian-Model/releases/download/
2020.08.07/transfer_model_tflite_it.tar.xz
tar xvf transfer_model_tflite_it.tar.xz
rm transfer_model_tflite_it.tar.xz

git clone https://github.com/MainRo/deepspeech-server
cd deepspeech-server
pip install .
cp config.sample.json config.json
nano config.json
```

Aggiungere a config.json:

```
...
"deepspeech": {
  "model" :"/home/pi/deepspeech/output_graph.tflite",
  "scorer" :"/home/pi/deepspeech/scorer"
},
...
```

```
sudo nano .config/mycroft/mycroft.conf
```

Aggiungere a mycroft.conf:

```
...
"stt": {
```

```
"deepspeech_server": {
  "uri": "http://localhost:8080/stt"
},
"module": "deepspeech_server"
},
...
```

Creare il servizio deepspeech-server

```
sudo nano /etc/systemd/system/deepspeech-server.service
```

Scrivere nel file:

```
[Unit]
Description=Deepspeech server

[Service]
User=pi
Type=simple
ExecStart=/home/pi/mycroft-core/.venv/bin/deepspeech-server --config
/home/pi/deepspeech/deepspeech-server/config.js
Restart=always
RestartSec=3

[Install]
WantedBy=multi-user.target
```

Attivare il servizio:

```
sudo systemctl enable deepspeech-server
sudo systemctl start deepspeech-server
```

Work in progress

From:
<https://wiki.csgalileo.org/> - **Galileo Labs**

Permanent link:
<https://wiki.csgalileo.org/projects/iotaiuto/mycroft?rev=1637694607>

Last update: **2021/11/23 20:10**

