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 keyestudio)

- 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

Installazione DeepSpeech (pre addestrato)

DeepSpeech è il motore STT di Mozilla Scaricare ed estrarre questo modello basato su DeepSpeech: https://coqui.ai/italian/jaco-assistant/v0.0.1

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
# copiare in questa cartella il modello tflite e lo scorer scaricati
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: (assicurarsi che le path siano corrette)

```
...
"deepspeech": {
    "model" :"/home/pi/deepspeech/model.tflite",
    "scorer" :"/home/pi/deepspeech/kenlm_it.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.j$
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

Addestramento manuale di una rete neurale

ATTENZIONE

Prosegui solo se non vuoi usare il modello già addestrato e vuoi addestrare manualmente una rete neurale DeepSpeech.

Requisiti: PC Desktop con scheda grafica nvidia Assicurarsi che il sistema operativo supporti CUDA 10.0.0. Non usare versioni più recenti di cuda. (Testato con Ubuntu 18.04)

Verificare se i driver nvidia sono installati:

nvidia-smi

Questo comando dovrebbe mostrare una tabella con alcune proprietà della gpu

Installare Cuda 10.0.0, potrebbe essere necessario disinstallare i driver nvidia e reinstallarli dopo cuda https://developer.nvidia.com/cuda-downloads

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 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:

Ho preso un tool di Mozilla-Italia che semplifica l'addestramento, ne ho risolto alcuni bug

```
cd $HOME
git clone https://github.com/danieltinazzi/DeepSpeech-Italian-Model
cd DeepSpeech-Italian-Model/DeepSpeech
chmod +x generate_base_dockerfile.sh
./generate_base_dockerfile.sh
docker build . -f Dockerfile.train -t deepspeech/base:0.9.3
```

Avviare l'addestramento

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

docker build . -f Dockerfile it.train -t deepspeech/it

Se il processo occupa troppa memoria della gpu potrebbe essere necessario ridurre il BATCH_SIZE, esempio:

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

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

Permanent link: https://wiki.csgalileo.org/projects/iotaiuto/mycroft

Last update: 2021/12/15 13:10

