

Overview

- [RaspberrySoftware](#)
 - [Azioni Preliminari](#)
 - [Se non abbiamo WIFI](#)
 - [Aggiornamento del sistema](#)
 - [Uno sguardo al sistema](#)
 - [Abilitazione](#)
 - [Installazione-base 2022](#)
 - [Java](#)
 - [GIT](#)
 - [cmake](#)
 - [Altre Installazioni](#)
 - [Node ed Express](#)
 - [Samba](#)
 - [aiocoap](#)
 - [Shellinabox](#)
 - [Python](#)
 - [Installazione del modulo GPIO per Python3](#)
 - [Ambiente virtuale Python](#)
 - [Installazione di virtualenv](#)
 - [Installezione di virtualenvwrapper](#)
 - [Creazione di un virtual environment](#)
 - [Multi-Media](#)
 - [Audio con alsa](#)
 - [RaspiCam](#)
 - [WebCam](#)
 - [mjpg-streamer](#)
 - [pygame](#)
 - [opencv](#)
 - [ngrok](#)
 - [wiringpi on Bullseye](#)
 - [websocket](#)
 - [Docker](#)
 - [Docker-compose](#)
 - [Senza wifi](#)
 - [wifi su bullseye](#)

RaspberrySoftware

Costa poco e permette di fare (e di imparare) molto.

Riportiamo alcuni appunti per rendere più agevole l'attivazione e l'uso del RaspberryPi. Il modello di riferimento è il [RaspberryPi 3 B+](#) ma possono essere usati anche anche modelli precedenti e ovviamente il [RaspberryPi 4 B](#).

- Un interessante articolo sui microchips: <https://ig.ft.com/microchips/>
- Come caricare il (nuovo) sistema operativo: <https://www.raspberrypi.com/news/raspberry-pi-bullseye-update-april-2022/>

Azioni Preliminari

1. Crea scheda *SecureDigital* (SD) (usando [RpiImager](#)) con [2020-02-13-raspbian-buster-lite](#) o con il più recente [Bullseye](#).

CTRL+SHIFT+X: si apre una finestra di personalizzazione del sistema operativo

In precedenza:

1. Con la scheda inserita in un PC, crea un file vuoto [boot/ssh](#).
2. Crea un file [boot/wpa_supplicant.conf](#) con il seguente contenuto

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=IT
network={
    ssid="... Your_ESSID "
    psk="... Your_wifi_password"
    key_mgmt=WPA-PSK
}
```

A questo punto inserisci la scheda SD sul Raspberry, alimentalo con **5VDC 700mA** e poi (Si veda anche [Raspberry Pi Wi-Fi & Bluetooth Setup](#)):

1. Leggi (opzionalmente collega il Rasperry con un cavo Ethernet) l'indirizzo IP del Raspberry sul Modem-HUB di casa.
2. Connetti il Raspberry via [ssh](#) usando [Putty](#).

Quando il sistema si attiva i file in boot è trasferito in

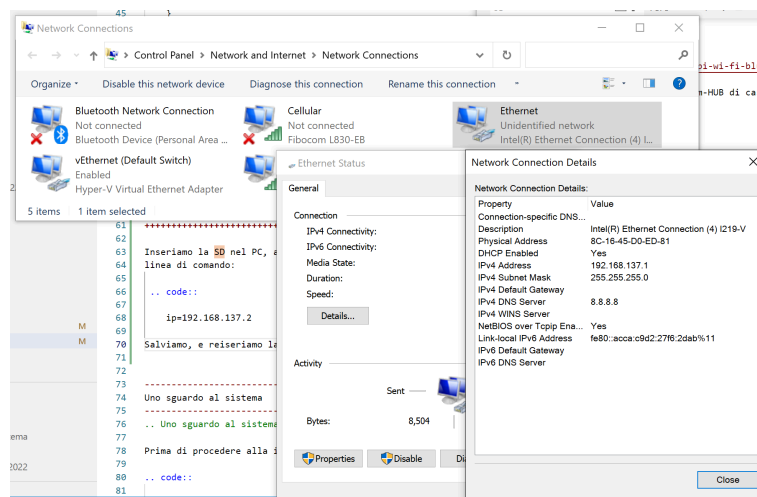
[etc/wpa_supplicant/wpa_supplicant.conf](#)

Se non abbiamo WIFI

Inseriamo la SD nel PC, apriamo con un editor il file [boot/cmdline.txt](#) e scriviamo in fondo alla linea di comando:

```
ip=192.168.137.2
```

Salviamo e reiseriamo la SD nel RaspberryPi e in Windows [Network and Sharing Center](#) di *Control Panel/Network and Internet/* vediamo:



Accendiamo il Raspberry, che dovrebbe essere accessibile all'indirizzo [192.168.137.2](#).

Aggiornamento del sistema

```
sudo apt update
sudo apt full-upgrade

sudo raspi-config

Abilita l'opzione 3 Interface Options -> P3 VNC

https://www.realvnc.com/en/connect/download/viewer/
```

Uno sguardo al sistema

Prima di procedere alla installazione di nuovo software, conviene dare un'occhiata alla versione iniziale

```
cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 10 (buster)"
NAME="Raspbian GNU/Linux"
VERSION_ID="10"
...

uname -a
Linux raspberrypi 4.19.118-v7+ #1311 SMP Mon Apr 27 14:21:24 BST 2020 armv7l GNU/Linux
```

Visualizziamo i dispositivi

```
arecord -l #visualizza i dispositivi
**** List of CAPTURE Hardware Devices ****
card 1: Camera [HD Web Camera], device 0: USB Audio [USB Audio]

arecord -L #visualizza in modo esteso
plughw:CARD=Camera,DEV=0
HD Web Camera, USB Audio
Hardware device with all software conversions
```

Abilitazione

```
sudo raspi-config
```

Installazione-base 2022

Java

```
//sudo apt install openjdk-8-jdk openjdk-8-jre
sudo apt install openjdk-11-jdk openjdk-11-jre
update-alternatives --config java
java -version
-----JAVA HOME -----
Include in /etc/environment
JRE_HOME=/usr/lib/jvm/java-8-openjdk-armhf/jre/bin/java
```

Installazione di **OpenJDK 11**, il default Java development e runtime nell'ultima versione di Raspbian OS, basata su Debian 10, Buster.

```
sudo apt update
sudo apt install default-jdk
java -version
```

GIT

```
sudo apt-get install git
git --version          #git version 2.20.1
```

cmake

cmake è un software multiplatforma gratuito e open source per l'automazione della build, il test, il confezionamento e l'installazione di software utilizzando un metodo indipendente dal compilatore. Ha dipendenze minime, richiedendo solo un compilatore C++ sul proprio sistema di compilazione.

```
sudo apt-get install cmake
```

Altre Installazioni

Node ed Express

Node.js è un runtime system open source multiplatforma orientato agli eventi per l'esecuzione di codice JavaScript, costruita sul motore JavaScript V8 di Google Chrome. Molti dei suoi moduli base sono scritti in JavaScript, e gli sviluppatori possono scrivere nuovi moduli in JavaScript.

```
sudo apt remove nodejs nodejs-legacy -y
sudo apt remove npm -y
sudo apt remove --purge node

//updates our Debian apt package repository
//to include the NodeSource packages
curl -sL https://deb.nodesource.com/setup_13.x | sudo -E bash -
sudo apt-get install -y nodejs // to install Node.js 13.x and npm

node -v          //v13.7.0
npm --version    //6.13.6

npm init          //create package.json
npm install serialport //--save is implicit

npm install express
```

Samba

Samba consente la condivisione di file e stampe tra computer che eseguono Microsoft Windows e computer che eseguono Unix.

```
sudo apt install -y samba
```

aiocoap

Il package **aiocoap** è un'implementazione di **CoAP** (*Constrained Application Protocol*). È scritto in Python 3 utilizzando i suoi metodi **asyncio** nativi per facilitare le operazioni simultanee mantenendo un'interfaccia facile da usare.

```
pip install aiocoap
```

Si veda anche [*Copper for Chrome \(Cu4Cr\) CoAP user-agent*](#)

Shellinabox

Shellinabox utilizza la tecnologia **AJAX** per fornire l'aspetto di una shell nativa tramite un browser web. Il demone **shellinabox** implementa un server web che ascolta sulla porta specificata (il default è **4200**). Il server web pubblica uno o più servizi che verranno visualizzati in un emulatore **VT100** implementato come applicazione web **AJAX**.

Shellinabox è incluso in molte distribuzioni Linux tramite repository predefiniti. Per motivi di sicurezza, è bene cambiare la porta predefinita (ad esempio in 6754) per rendere difficile a chiunque raggiungere la casella SSH.

Durante l'installazione viene creato automaticamente un nuovo certificato SSL autofirmato in [/var/lib/shellinabox](#) per utilizzare il protocollo *HTTPS*.

```
sudo apt-cache search shellinabox
sudo apt-get install openssl
sudo apt-get -y install shellinabox

SHELLINABOX_PORT=6754           #Cambio della porta
sudo service shellinabox start   #Attiva il servizio
sudo /etc/init.d/shellinabox restart

/etc/init.d/shellinabox status
sudo service --status-all       #visualizza i servizi
sudo netstat -tlnp               #visualizza i servizi and le porte TCP in ascolto
sudo netstat -nap | grep shellinabox #verifica

https://localhost:4200
```

Python

Dando i comandi:

```
python --version
python3 --version
```

su versioni del sistema operativo precedenti la **11** (*bullseye*) si possono avere risposte diverse.

Una volta installato, Python3 si trova in [/usr/bin/](#) ed è un symlink di [/usr/bin/python3](#), che a sua volta è un symlink di [/usr/bin/python3.7](#) (il vero binario).

```
which python3
/usr/bin/python3
ls -lart /usr/bin/python3
lrwxrwxrwx 1 root root 9 Mar 26 2019 /usr/bin/python3 -> python3.7
ls -lart /usr/bin/python3.7
-rwxr-xr-x 2 root root 4275580 Dec 20 2019 /usr/bin/python3.7

-----
22/07/2024

USO Rasp4 con buster10
wget -qO - https://raw.githubusercontent.com/tvdsluijs/sh-python-installer/main/python.sh | sudo bash
//ci mette parecchio tempo ma tutto ok ...

pip install virtualenv
python -m venv ml24 //in /home/pi

source ml24/bin/activate
deactivate

https://www.youtube.com/watch?v=2rJCGyHQ_zM
https://github.com/modzy/hugging-face-raspberry-pi?tab=readme-ov-file#environment-setup
pip install notebook
pip install torch transformers[torch] numpy chassism1 modzy-sdk grpcio~=1.50.0 protobuf~=4.21.9 google

jupyter notebook
http://192.168.1.248:8888/tree?token=41edd75110919fa451f1e08f0c87366cba48841470cc3dd7
http://127.0.0.1:8888/tree?token=a2ef06ab91841da02dc67d660a1e92d1a71074b34709747c
```

Installazione del modulo GPIO per Python3

Il modulo Python per interagire con i GPIO non è preinstallato per la versione Python3. Per installarlo, procedere come segue:

```
sudo apt update //aggiorna le informazioni sul repository del software
sudo apt upgrade //installa gli aggiornamenti disponibili (ci vuole tempo ... )

sudo apt install python3-rpi.gpio
```

Ambiente virtuale Python

Un ambiente virtuale è uno strumento Python per la gestione delle dipendenze e l'isolamento del progetto. Consentono ai Package del sito Python (librerie di terze parti) di essere installati localmente in una directory isolata per un particolare progetto, invece di essere installati globalmente (cioè come parte di un Python a livello di sistema).

Installazione di virtualenv

Fase 1: aggiungiamo un'opzione di configurazione al file hidden denominato `.bashrc` aggiungendo il comando (alias) `myenv`.

```
( echo; echo '##### added for myenv #####'; echo 'export PATH=/home/pi/.local/bin:$PATH'; echo "alias
. ~/.bashrc
```

Fase 2: attiviamo il nuovo virtualenv e entriamo in esso:

```
pip3 install --upgrade pip
python3 -m pip install virtualenv
python3 -m virtualenv myenv
```

In coppia con [virtualenv](#), è consigliabile l'installazione del modulo [virtualenvwrapper](#) che contiene una serie di utilities per facilitare la gestione degli ambienti virtuali.

Installazione di virtualenvwrapper

```
sudo pip3 install virtualenv virtualenvwrapper
pip install virtualenvwrapper-win //su Windows10
```

Verifichiamo l'avvenuta installazione e la relativa versione:

```
virtualenv --version
virtualenv 20.10.0 from /home/pi/.local/lib/python3.7/site-packages/virtualenv/__init__.py
```

Per fruire degli ambienti aggiungiamo i riferimenti e le risorse nel profilo

```
nano ~/.profile
-----
# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
source /usr/local/bin/virtualenvwrapper.sh
-----
```

Ricarichiamo le risorse del profilo:

```
source ~/.profile
```

Creazione di un virtual environment

Posizionamoci in una directory di lavoro e creiamo un ambiente per Python 3 denominato **myenv**.

```
mkvirtualenv myenv
```

Per entrare nel virtualenv appena creato basta digitare:

myenv

Per uscire dal virtualenv:

deactivate.

Per visualizzare gli ambienti virtuali creati, occorre avere installato *virtualenvwrapper*:

```
lsvirtualenv -l
```

Multi-Media

Audio con alsa

Nella directory */home/pi/nat* inseriamo file wav ed mp3 ed eseguiamo (per sentire audio mediante cell-phone TRRS Headset):

```
aplay example_WAV.wav          #(non usare per mp3)
omxplayer Oratore.mp3          #(- + regola volume)
omxplayer example_WAV.wav      #riproduce - + modifica volumer
omxplayer /home/pi/Music/Ella_Fitzgerald_Summertime.mp3

speaker-test -c2 --test=wav -w Rear_Center.wav
(cerca in /usr/share/sounds/alsa/xxx.wav)
```

Una libreria di suoni: <https://pdsounds.org/>

RaspiCam

```
raspistill -o image.jpg          #scatta immagine
raspivid -o video.h264 -t 5000   #crea video (in msecs)
```

WebCam

```
sudo apt install fswebcam
fswebcam image1.jpg              #crea immagine 640x320
fswebcam -r 1280x720 image2.jpg  #crea immagine 1280x720
fswebcam -r 320x240 image3.jpg   #crea immagine 320x240
```

Se abbiamo installato una WebCam con microfono, controlliamone il funzionamento

```
arecord -D plughw:Camera test.wav
arecord -D plughw:Camera,0 -d 5 -f cd test.wav -c 2
omxplayer test.wav              #riproduce - + modifica volumer

alsamixer #F6 seleziona la scheda
```

mjpg-streamer

```
git clone https://github.com/jacksonliam/mjpg-streamer.git
sudo apt-get install cmake libjpeg9-dev
cd mjpg-streamer/mjpg-streamer-experimental/
make
sudo make install
```

Attivazione e prova:

```
./start.sh      or  bash start.sh  
http://localhost:8080/stream.html
```

Se la **webcam si ferma** dopo qualche secondo, modificare in **/boot/config.txt** da `gpu_mem=128` a `gpu_mem=256`.

Altro codice di attivazione:

```
/usr/local/bin/mjpg_streamer -i "input_uvc.so -r 1280x720 -d /dev/video0 -f 30" -o "output_http.so  
http://localhost:8085/stream.html
```

pygame

PyGame di solito viene installato con l'ultima distribuzione Raspbian Pygame v1.9 is in raspi os at install.

```
/usr/local/lib #directory delle librerie  
  
sudo apt-get install libsdl-ttf2.0-0  
python3 -m pip install -U pygame --user #install in the home directory  
pygame.mixer.music.load("/home/pi/Music/Oratore.mp3")  
pygame.mixer.music.play(0)  
  
python
```

opencv

L'ultima versione non sembra facilmente caricabile.

```
sudo apt install libatlas3-base  
pip install opencv-python==4.4.0.42  
  
python face_detection.py      #da: https://www.html.it/articoli/face-detection-python-10-linee-di-codice/  
Necessita scaricare          #da: https://github.com/opencv/opencv/tree/master/data/haarcascades
```

ngrok

1. Download di ngrok (tar -xvzf ngrok-stable-linux-arm.tgz)
2. Acquire account (ad es. con Google)
3. Acquire authtoken (xxx)
4. ngrok authtoken xxx (salvato in /home/pi/.ngrok2/ngrok.yml)
5. ngrok http 8081
6. usare il forwarding proposto (<http://1ea-95-249-218-184.ngrok.io>)

wiringpi on Bullseye

Su Bullseye wiringPi è deprecated. Possiamo però ricaricarlo come segue:

```
cd /tmp  
wget https://project-downloads.drogon.net/wiringpi-latest.deb  
sudo dpkg -i wiringpi-latest.deb
```

websocket

<https://spring.io/guides/gs/messaging-stomp-websocket/>

Docker

Installazione

```
sudo curl -fsSL https://get.docker.com -o /tmp/get-docker.sh
sudo chmod +x /tmp/get-docker.sh
sudo sh /tmp/get-docker.sh
```

Aggiunta permessi per eseguire comandi

```
sudo usermod -aG docker pi
reboot !!!
```

```
sudo dockerd &      #daemon runs with default configuration
docker version      #20.10.6
docker info
```

Test

```
docker run hello-world
```

Docker-compose

Installazione dipendenze

```
sudo apt-get install -y libffi-dev libssl-dev
sudo apt-get install -y python3 python3-pip
sudo apt-get remove python-configparser
```

Installazione docker-compose

```
sudo pip3 install docker-compose
```

Senza wifi

Richiesta: *come collegare un raspberry a un pc senza usare wifi*

Per collegare un Raspberry Pi a un PC senza utilizzare una connessione Wi-Fi, puoi utilizzare una connessione Ethernet diretta tra i due dispositivi. Questa configurazione ti consentirà di stabilire una connessione cablata tra il Raspberry Pi e il PC per condividere dati o stabilire una connessione SSH.

Ecco i passaggi da seguire per creare una connessione Ethernet diretta tra un Raspberry Pi e un PC:

- **Collega il cavo Ethernet:** Assicurati di disporre di un cavo Ethernet (anche noto come cavo LAN) e collega un'estremità del cavo all'interfaccia Ethernet (eth0) del Raspberry Pi e l'altra estremità al PC.
- **Configura l'interfaccia Ethernet sul Raspberry Pi:**
 - Se stai utilizzando Raspbian o Raspberry Pi OS, l'interfaccia Ethernet dovrebbe essere configurata in modo dinamico per ottenere un indirizzo IP tramite DHCP per impostazione predefinita. Verifica che DHCP sia abilitato.
 - Puoi anche configurare l'interfaccia Ethernet manualmente sul Raspberry Pi se preferisci un indirizzo IP statico. Per fare ciò, modifica il file `/etc/dhcpd.conf` sul Raspberry Pi. Ecco un esempio di come configurare un indirizzo IP statico:

```
plaintext
Copy code
interface eth0
static ip_address=192.168.1.2/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Assicurati che l'indirizzo IP sia nello stesso subnet della tua rete locale ma diverso dagli altri dispositivi. Il gateway (routers) dovrebbe essere l'indirizzo IP del tuo PC o del tuo router.

- **Configura l'interfaccia Ethernet sul PC:**

- Su un PC con sistema operativo Windows, apri "Connessioni di rete" e individua l'interfaccia Ethernet. Fai clic con il tasto destro del mouse su di essa e seleziona "Proprietà". Configura l'indirizzo IP e la maschera di sottorete manualmente in modo che siano compatibili con quelli configurati sul Raspberry Pi (esempio: IP: 192.168.1.1, Maschera di sottorete: 255.255.255.0).
- Su un PC con sistema operativo Linux, puoi utilizzare il comando `ifconfig` o `ip` per configurare manualmente l'interfaccia Ethernet. Ad esempio:

```
bash
Copy code
sudo ifconfig eth0 192.168.1.1 netmask 255.255.255.0
```

- **Testa la connessione:** Una volta configurate entrambe le interfacce Ethernet, dovresti essere in grado di accedere al Raspberry Pi dal PC utilizzando l'indirizzo IP assegnato al Raspberry Pi. Puoi utilizzare SSH per accedere al Raspberry Pi o condividere file tramite Samba (se configurato).

Assicurati di configurare correttamente l'indirizzo IP e la maschera di sottorete su entrambi i dispositivi, e dovresti essere in grado di stabilire una connessione Ethernet diretta tra il tuo Raspberry Pi e il PC senza utilizzare una connessione Wi-Fi.

wifi su bullseye

Per configurare una connessione Wi-Fi su un Raspberry Pi con il sistema operativo Raspberry Pi OS basato su Debian 11 "Bullseye", puoi seguire questi passaggi:

Nota: Assicurati che il tuo Raspberry Pi sia collegato a una tastiera, un mouse e un monitor o che abbia l'accesso SSH abilitato per eseguire questi passaggi.

- **Verifica del Wi-Fi integrato:** Se stai utilizzando un Raspberry Pi con Wi-Fi integrato (ad esempio, Raspberry Pi 3 Model B o successivo), assicurati che il modulo Wi-Fi sia riconosciuto dal sistema. Puoi farlo eseguendo il seguente comando:

```
iwconfig
```

Dovresti vedere un'interfaccia Wi-Fi elencata (ad esempio, wlan0).

- **Apri il file di configurazione delle reti:** Utilizza un editor di testo per aprire il file di configurazione delle reti:

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

- **Aggiungi la configurazione Wi-Fi:** Aggiungi le seguenti linee alla fine del file per specificare le tue informazioni di connessione Wi-Fi:

```
network={
    ssid="IlNomeDellaTuaReteWiFi"
    psk="LaTuaPasswordWiFi"
}
```

Sostituisci "IlNomeDellaTuaReteWiFi" con il nome (SSID) della tua rete Wi-Fi. Sostituisci "LaTuaPasswordWiFi" con la password della tua rete Wi-Fi. Se la tua rete Wi-Fi utilizza WPA2 o WPA3,

questo metodo funzionerà bene.

- **Salva e chiudi il file:** Premi Ctrl + O per salvare le modifiche e Ctrl + X per uscire dall'editor.
- **Riavvia il servizio Wi-Fi:** Esegui il seguente comando per riavviare il servizio Wi-Fi in modo che prenda in considerazione la nuova configurazione:

```
sudo systemctl restart networking
```

- **Verifica la connessione:** Puoi verificare la connessione Wi-Fi eseguendo il seguente comando:

```
ifconfig wlan0
```

Dovresti vedere un indirizzo IP assegnato all'interfaccia wlan0.

- **Opzionale: Abilita SSH (se necessario):** Se intendi accedere al Raspberry Pi via SSH, assicurati di aver abilitato SSH nelle impostazioni di Raspberry Pi OS. Puoi farlo tramite il comando `raspi-config`.
- **Riavvia il Raspberry Pi:** A questo punto, puoi riavviare il Raspberry Pi per assicurarti che la connessione Wi-Fi funzioni correttamente.

Una volta completati questi passaggi, il tuo Raspberry Pi dovrebbe essere connesso alla rete Wi-Fi specificata. Puoi quindi accedervi tramite SSH o utilizzare la connessione Wi-Fi per qualsiasi altra necessità di rete.

```
sudo systemctl enable NetworkManager  
sudo systemctl start NetworkManager  
sudo reboot
```