

Avvio di un programma in automatico all'accessione (Raspberry) YITAO-ZHU

Visto che Raspberry ha due interfacce , perciò per configurare l'avvio del programma in automatico all'accessione abbiamo due modi diversi:

1) Interfaccia grafica con l'uso del server X:

Dopo che abbiamo avviato Server X, apriamo un terminale e creiamo la cartella "autostart" con il seguente codice :

```
mkdir ~/.config/auto start
```

All'interno della cartella auto start creiamo un file nuovo vuoto :

```
touch ~/.config/autostart/NomeProgramma.desktop
```

Al posto del NomeProgramma possiamo inserire qualsiasi nome ci aiuti a riconoscere il file, l'importante è che abbia l'estensione .desktop.

Ora dovremo modificare il file fornendogli le informazioni del programma che ci interessa avviare:

Codice:

```
sudo leafpad ~/.config/autostart/
```

In questo modo si aprirà l'editor di testo predefinito all'interno di Raspbian.

Compiliamo il contenuto del file come segue:

```
[Desktop Entry]
```

```
Name=nomeprogramma
```

```
Exec=comando
```

```
Type=application
```

"nomeprogramma" e "comando" saranno le uniche informazioni che dovrete modificare.

Dovrete a questo punto identificare quale comando permette l'avvio da terminale del programma che vi interessa, e inserirlo sotto la voce Exec=, in sostituzione al testo "comando".

Salvate, chiudete l'editor ed il gioco è fatto.

Ad ogni accensione il programma verrà avviato come richiesto.

2) Interfaccia Prompt Comandi senza X Server

Da linea di comando richiediamo
tramite l'editor "nano" la modifica
del file rc.local (se non presente
verrà creato)

Codice:

```
sudo nano /etc/rc.local
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb  2 07:25:03 2015 from 172.16.13.19
pi@raspi-garden ~ $ sudo nano /etc/rc.local
GNU nano 2.2.6 File: /etc/rc.local

# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
```

Appena prima della linea "exit" inseriamo seguente codice:

```
cd /percorso/file/
```

```
sudo mono NomeProgramma.exe &
```

& finale serve per eseguire un programma in
background.

```
sudo modprobe wl-gpio
sudo modprobe wl-therm

cd /home/pi/Acquisition-Zambelli
sudo mono GorAcquire.exe &

exit 0
```

Usare istruzione `sudo reboot now` per riavviare il raspberry

Fatto questo sarà sufficiente salvare, confermare ed uscire.

Una volta riavviato usare istruzione `ps aux NomeProgramma` per visualizzare i processi in esecuzione.

```
pi@raspi-garden ~ $ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.6  0.1   2148    720 ?        Ss   07:23   0:01 init [2]
root         2   0.0  0.0     0      0 ?        S    07:23   0:00 [kthreadd]
root         3   0.0  0.0     0      0 ?        S    07:23   0:00 [ksoftirqd/0]
root         4   0.0  0.0     0      0 ?        S    07:23   0:00 [kworker/0:0]
root         5   0.0  0.0     0      0 ?        S<   07:23   0:00 [kworker/0:0]
root         6   0.0  0.0     0      0 ?        S    07:23   0:00 [kworker/u2:0]
root         7   0.1  0.0     0      0 ?        S    07:23   0:00 [rcu_preempt]
root         8   0.0  0.0     0      0 ?        S    07:23   0:00 [rcu_bh]
root         9   0.0  0.0     0      0 ?        S    07:23   0:00 [rcu_sched]
root        10   0.0  0.0     0      0 ?        S<   07:23   0:00 [khelper]
root        11   0.0  0.0     0      0 ?        S    07:23   0:00 [kdevtmpfs]
root        12   0.0  0.0     0      0 ?        S<   07:23   0:00 [netns]
root        13   0.0  0.0     0      0 ?        S<   07:23   0:00 [writeback]
root        14   0.0  0.0     0      0 ?        S<   07:23   0:00 [klogd]
```

Per eventuali errori e problemi a causa dei driver aggiungere prima del istruzione nel `rc.local` i seguenti istruzioni:

```
sudo modprobe w1-gpio&
```

```
sudo modprobe w1-therm&
```

oppure

Se volete che i moduli vengano caricati all'avvio in automatico, date i comandi:

```
sudo nano /etc/modules
```

e aggiungere:

```
w1-gpio
```

```
w1-therm
```

Stessa cosa per device I2C
aggiungere nel `/etc/rc.local`

```
sudo modprobe i2c-  
bcm2708&
```

```
sudo modprobe i2c-dev&
```

oppure nel `sudo nano`
`/etc/modules` aggiungere:

```
i2c-bcm2708
```

```
i2c-dev
```

```
pi@garden-100 ~ $ sudo nano /etc/modules
GNU nano 2.2.6      File: /etc/modules      Mod

# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-dev
#bcm2708_wdog
w1-gpio
w1-therm
i2c-bcm2708
i2c-dev
```

Fonte: [Moddingstudio.com - Raspberry: Come avviare un programma in automatico all'accensione.](http://www.moddingstudio.com/content.php?1474-Raspberry-Come-avviare-un-programma-in-automatico-all'accensione.#ixzz3OJJvpHKp)
<http://www.moddingstudio.com/content.php?1474-Raspberry-Come-avviare-un-programma-in-automatico-all'accensione.#ixzz3OJJvpHKp>

<http://raspberrypiwebserver.com/serveradmin/run-a-script-on-start-up.html>