

Oled ssd1306 in Ambiente Virtuale Python

Aggiornamento Raspberry Pi:

```
| sudo apt update && sudo apt upgrade -y && sudo apt autoremove -y
```

Installare Python3 (pip e venv) e git:

```
| sudo apt-get install python3-pip git python3-venv -y
```

Enable I2C - Gpio Fan e italiano:

```
| sudo raspi-config
```

Attivare:

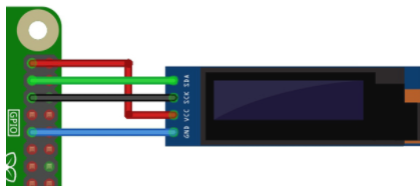
```
3 Interface Options
- I5 IC2

4 Performance Options
- P4 Fan (yes, 14, ok, 80, ok, ok) Impostare il comportamento di un GPIO uscita FAN.

5 Localisation Options
- L1 Locale (selezionare con tasto spazio it_IT.UTF-8 UTF-8, ok evidenziare e ancora ok)

Finish
```

Collegamento ssd1306 128x32 al GPIO del Raspberry Pi.



Ricollegarsi con SSH:

Accelerare la visualizzazione

Per ottenere migliori prestazioni, eseguire questa modifica di configurazione con:

```
| sudo nano /boot/config.txt
```

e aggiungere alla voce presente che trovi **,i2c_baudrate=1000000**

```
| dtparam=i2c_arm=on,i2c_baudrate=1000000
```

Usare **CTRL+O** per salvare, **Invio** e quindi **CTRL+X** per uscire .

Controllo della temperatura in automatico con ventola :

Sfruttando il **GPIO-Fan**; si spegne la ventola quando raggiunge 10° in meno del valore che abbiamo stabilito per attivarla in **config.txt**.

Per pilotare il ventilatore (FAN) dal pin GPIO14 servono:

- Q1: Transistor NPN , (ZTX653)
- R1: 330 R
- D1: diodo 1N4001...7

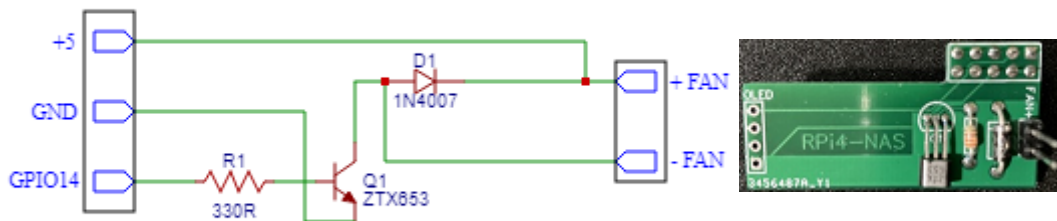
digitare:

```
sudo nano /boot/config.txt
```

modificare il valore temp=80000 in (temp=70000) oppure quello che si desidera.

```
[all]
```

```
dtoverlay=gpio-fan,gpiopin=14,temp=70000
```



Installare programma per il dispositivo OLED 128x32 I2C

Scaricare

```
git clone https://github.com/Tato9591/Oled-ssd1306-hwmonitor.git
```

```
cd Oled-ssd1306-hwmonitor
```

Creare ambiente virtuale con venv:

```
python3 -m venv .venv
```

Attivare l'ambiente virtuale:

```
source .venv/bin/activate
```

Installare con pip le dipendenze del programma:

```
pip install Pillow
```

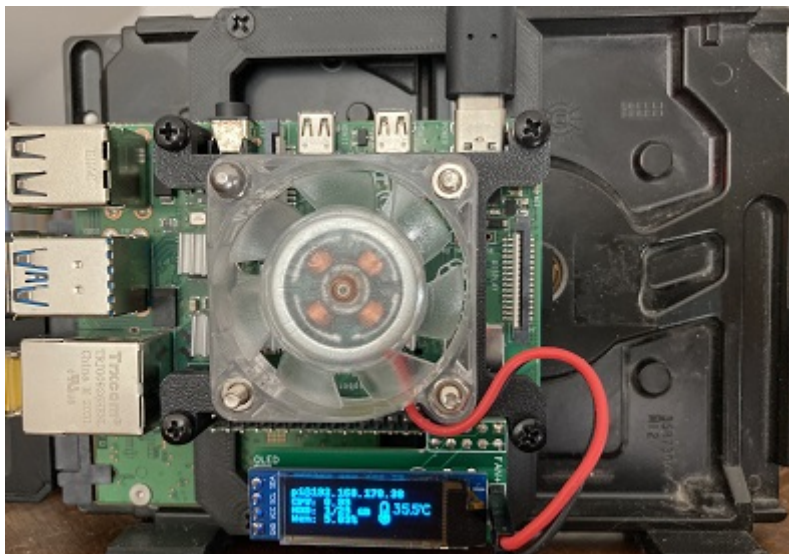
```
pip install adafruit-circuitpython-ssd1306
```

```
pip install vcgencmd
```

```
pip install gpiozero
```

```
python hwmonitor.py
```

e dovresti vedere qualcosa come la seguente immagine:



per uscire dal programma **Ctrl + c**.

per disattivare l'ambiente virtuale:

```
deactivate
```

inserire i dati per usare l'invio di posta nel file mail.py

```
sudo nano mail.py
```

```
import smtplib
import time

#-----#
#Se stiamo utilizzando il servizio Gmail di Google per inviare posta. |
#Abbiamo bisogno di alcune modifiche alle impostazioni, per consentire |
#l'accesso, è necessario impostare "Accesso meno sicuro alle app" |
#nell'account Google. Se la verifica in due passaggi è in corso, non possiamo |
#utilizzare l'accesso meno sicuro. |
# |
#Per completare questa configurazione, vai alla console di amministrazione |
#di Google e cerca la configurazione meno sicura dell'app. |
#-----#
```

```
# io uso la mail del modem per evitare problemi

FROM = "modifica@from.it"
TO = "modifica@to.com"

OGGETTO = "modifica_oggetto"
TEXT = "modifica_TEXT.\n\n"

message = f"From: {FROM}\nTo: {TO}\nSubject: {OGGETTO}\n\n{TEXT}"

def send_email():
    server = smtplib.SMTP('modifica_SMTP', modifica_PORTA)
    server.ehlo()
    server.starttls()
    server.ehlo()
    server.login("modifica_user", "modifica_password")
    server.sendmail(FROM, TO, message)
    server.quit()

#per utilizzo in un altro file nel mio caso hwmonitor.py
#
#from mail import send_mail
#import time
#
#while True:
#    send_email()
#    time.sleep(0.5)
#    break
#
```

```
cd
```

```
sudo nano launcher.sh
```

Scrivere:

```
#!/bin/bash
cd
cd /home/pi/Oled-ssd1306-hwmonitor
source .venv/bin/activate
python hwmonitor.py &
deactivate
exit 0
```

Usare **CTRL+O** per salvare, **Invio** e quindi **CTRL+X** per uscire.

```
sudo chmod +x launcher.sh
```

digitare:

```
crontab -e
```

Inserire a fine riga:

```
| @reboot /home/pi/.launcher.sh > /dev/null 2>&1
```

Usare **CTRL+O** per salvare, **Invio** e quindi **CTRL+X** per uscire.

```
| sudo reboot
```