

Oled-ssd1306-hwmonitor

Per Raspberry pi a 40 pin dal PiZero al Pi4:

```
sudo apt-get update && sudo apt-get upgrade -y && sudo apt-get autoremove -y  
sudo apt-get install -y python3-pip git  
sudo pip3 install --upgrade setuptools  
sudo pip3 install vcgencmd  
sudo pip3 install adafruit-circuitpython-ssd1306  
sudo reboot now  
sudo pip3 install --upgrade adafruit-python-shell  
wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/raspi-blinka.py  
sudo python3 raspi-blinka.py  
sudo rm *.py
```

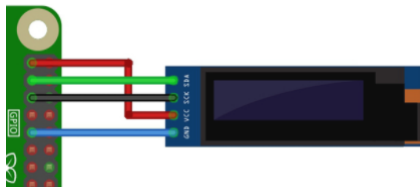
Gpio Fan - timezone

```
sudo raspi-config
```

Attivare:

```
4 Performance Options  
- P4 Fan (yes, 14, ok, 80, ok, ok) Set the behaviour of a GPIO connected 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.



Eseguire il comando seguente dal prompt del terminale per analizzare/rilevare i dispositivi I2C :

```
sudo i2cdetect -y 1
```

Dovrebbe essere visualizzato quanto segue (solo se é collegato ai Gpio):

```
pi@raspberrypi: ~  
Setting up read-edid (3.0.2-1) ...  
Setting up i2c-tools (4.1-1) ...  
Processing triggers for man-db (2.8.5-2) ...  
Processing triggers for libc-bin (2.28-10+rpil) ...  
pi@raspberrypi:~$ i2cdetect -y 1  
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
30:  --  --  --  --  --  --  --  --  --  --  --  3c  --  --  --  
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
pi@raspberrypi:~$
```

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 .

Verifica funzionamento del dispositivo OLED 128x32 I2C

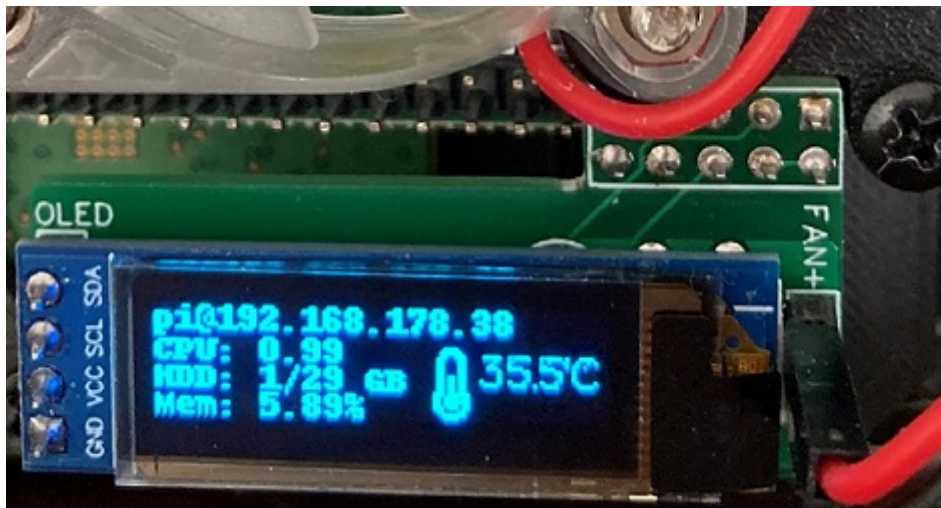
Scaricare:

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

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

```
| python3 hwmonitor.py
```

e dovresti vedere qualcosa come la seguente immagine (**ctrl+c**) per terminare:



```
| cd
```

```
| sudo nano launcher.sh
```

Scrivere:

```
#!/bin/bash
cd
cd Oled-ssd1306-hwmonitor
python3 hwmonitor.py &
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 ./launcher.sh > /dev/null 2>&1
```

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

Impostare i parametri con la voce "modifica" per inviare e ricevere avviso con email.

```
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 inviare.

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()
```

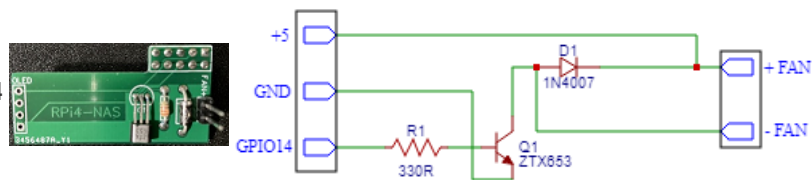
```
#per utilizzo in un altro file .py
#
#from mail import send_email
#import time
#
#while True:
#    send_email()
#    time.sleep(0.5)
#    break
#
```

Per attivazione controllo automatico della temperatura con ventola:

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

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

Interfaccia comando ventilatore RPi4



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
```

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

```
sudo reboot now
```