



SMART BUD

Demo (https://bud.eco-sensors.ch) | Tutoriels (https://eco-sensors.ch/tutoriels/)



NOUS SOUTENIR

Shop (https://eco-sensors.ch/shop/) & Sponsoring (https://eco-sensors.ch/product-category/dons/)



SOCIAL

Facebook (https://www.facebook.com/ecosensors) - Contact (https://eco-sensors.ch/contact/)



(/)



(/)

Utilisation d'une camera thermique

HOME (HTTPS://ECO-SENSORS.CH/)
/ SYSTÈMES EMBARQUÉS (HTTPS://ECO-SENSORS.CH/CATEGORY/SYSTEMES-EMBARQUES/)
/ UTILISATION D'UNE CAMERA THERMIQUE

BY ECOSENSORS (HTTPS://ECO-SENSORS.CH/AUTHOR/ECOSENSORS/)

/

13 DÉCEMBRE 2018 (HTTPS://ECO-SENSORS.CH/CAMERA-THERMIQUE/)

♥ 0 👁 2,123 💬 0 f 🐦

SYSTÈMES EMBARQUÉS (HTTPS://ECO-SENSORS.CH/CATEGORY/SYSTEMES-EMBARQUES/)

Dans cet article je vais brièvement vous montrer comment configurer une camera thermique ARG8833 avec un Raspberry Pi

French

4 et un display PiTFT 3.5" resistive touch (320×480)

*Cet article n'est pas terminé mais sert
comme aide-mémoire*

Préparation de votre Raspberry

Pour l'installation de votre Raspberry, vous pouvez vous aider de cet article. (<https://eco-sensors.ch/2-faire-ses-sauvegardes-sans-connexion/#installation>)

Privilégier, la dernière version de Rasbian, évidemment.

**⚠ MAIS ATTENTION, DANS LE CADRE DE CETTE EXERCICE,
NE SUPPRIMER PAS LE COMPTE PI.**

Il vous faudra encore faire les mises à jour et installer git (et vim et ntpdate, si ce n'est pas fait)

```
1 sudo apt-get update
2 sudo apt-get upgrade
3 sudo apt-get install vim ntpdate
4 sudo apt-get install git
```

Installation de l'écran

Référence: <https://learn.adafruit.com/adafruit-pitft-3-dot-5-touch-screen-for-raspberry-pi?view=all> (<https://learn.adafruit.com/adafruit-pitft-3-dot-5-touch-screen-for-raspberry-pi?view=all>)

<https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/installing-circuitpython-on-raspberry-pi> (<https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/installing-circuitpython-on-raspberry-pi>)

circuitpython-on-raspberry-pi)

```
1 cd ~
2 sudo apt-get install -y git python3-pip
3 sudo pip3 install --upgrade adafruit-python-sl
4 git clone https://github.com/adafruit/Raspber
5 cd Raspberry-Pi-Installer-Scripts
6 sudo python3 adafruit-pitft.py --display=35r .
```

Redémarrer votre Raspberry et votre écran fonctionnera

*La partie ci-dessous est obsolète. Je la garde
pour mémoire*

Select configuration:

1. PiTFT 2.4", 2.8" or 3.2" resistive (240x320)
2. PiTFT 2.2" no touch (240x320)
3. PiTFT 2.8" capacitive touch (240x320)
4. PiTFT 3.5" resistive touch (320x480)
5. Quit without installing

SELECT 1-5: 4

Select rotation:

1. 90 degrees (landscape)
2. 180 degrees (portait)
3. 270 degrees (landscape)
4. 0 degrees (portait)

SELECT 1-4: 1

[PITFT] Checking init system...

Found systemd

/boot is mounted

[PITFT] System update

Updating apt indexes...

.....

Reading package lists...

.....

[PITFT] Installing Python libraries & Software...

Installing Pre-requisite Software...This may take a few minutes

[PITFT] Updating /boot/config.txt...

[PITFT] Updating SysFS rules for Touchscreen...

[PITFT] Updating TSLib default calibration...

Would you like the console to appear on the PiTFT display?

[PITFT] Making sure console doesn't use PiTFT

Removing console fbcon map from /boot/cmdline.txt

Screen blanking time reset to 10 minutes

Would you like the HDMI display to mirror to the PiTFT display?

[PITFT] Adding FBCP support...

Installing cmake...

W: --force-yes is deprecated, use one of the options starting with --

Downloading rpi-fbcp...

Uncompressing rpi-fbcp...

Building rpi-fbcp...

Installing rpi-fbcp...

Remove fbcp from /etc/rc.local, if it's there...

We have systemd, so install fbcp systemd unit...

Created symlink /etc/systemd/system/multi-user.target.wants/rpi-fbcp.service

Setting raspi-config to boot to desktop w/o login...

Created symlink /etc/systemd/system/default.target → /lib/systemd/system/rpi-fbcp.service

Configuring boot/config.txt for forced HDMI

Using x1.5 resolution

[PITFT] Updating X11 default calibration...

[PITFT] Success!

Settings take effect on next boot.

REBOOT NOW? [y/N] y

Installation de la camera thermique

<https://learn.adafruit.com/adafruit-amg8833-8x8-thermal-camera-sensor/raspberry-pi-thermal-camera> (<https://learn.adafruit.com/adafruit-amg8833-8x8-thermal-camera-sensor/raspberry-pi-thermal-camera>)

Connections

Vin à la broche t3V ou 5V

GND à la broche **GND**

SDA à la broche o **SDA**

SCL à la broche **SCL**

Contrôler les connections:

```
1 | sudo i2cdetect -y 1
```

Ce qui devrait afficher

```
0 1 2 3 4 5 6 7 8 9 a b c d e f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- 69 --
69 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Installation des softs

Documentation (<https://eco-sensors.ch/wp-content/uploads/2018/12/adafruit-amg8833-8x8-thermal-camera-sensor.pdf>)

<https://learn.adafruit.com/adafruit-amg8833-8x8-thermal-camera-sensor/raspberry-pi-thermal-camera> (<https://learn.adafruit.com/adafruit-amg8833-8x8-thermal-camera-sensor/raspberry-pi-thermal-camera>)

sensor/raspberry-pi-thermal-camera)

```
1 sudo apt-get install libatlas-base-dev
2 sudo pip3 install adafruit-circuitpython-amg88xx
3 sudo apt-get install -y python3-scipy python3-pil
4 sudo pip3 install colour
```

Le script

Créer le script

```
1 sudo nano /home/pi/thermcam.py
```

Le script d'adafruit

```
1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4 """This example is for Raspberry Pi (Linux) (C) 2021 Adafruit Industries
5     It will not work on microcontrollers running MicroPython
6
7 import os
8 import math
9 import time
10
11 import numpy as np
12 import pygame
13 import busio
14 import board
15
16 from scipy.interpolate import griddata
17
18 from colour import Color
19
20 import adafruit_amg88xx
21
22 i2c_bus = busio.I2C(board.SCL, board.SDA)
23
```

```

24 # low range of the sensor (this will be blue
25 MINTEMP = 15.0
26
27 # high range of the sensor (this will be red
28 MAXTEMP = 28.0
29
30 # how many color values we can have
31 COLORDEPTH = 1024
32
33 os.putenv("SDL_FBDEV", "/dev/fb1")
34 # pylint: disable=no-member
35 pygame.init()
36 # pylint: enable=no-member
37
38 # initialize the sensor
39 sensor = adafruit_amg88xx.AMG88XX(i2c_bus)
40
41 # pylint: disable=invalid-slice-index
42 points = [(math.floor(ix / 8), (ix % 8)) for
43 grid_x, grid_y = np.mgrid[0:7:32j, 0:7:32j]
44 # pylint: enable=invalid-slice-index
45
46 # sensor is an 8x8 grid so lets do a square
47 height = 400
48 width = 400
49
50 # the list of colors we can choose from
51 blue = Color("indigo")
52 colors = list(blue.range_to(Color("red"), COI
53
54 # create the array of colors
55 colors = [(int(c.red * 255), int(c.green * 25
56
57 displayPixelWidth = width / 30
58 displayPixelHeight = height / 30
59
60 lcd = pygame.display.set_mode((width, height)
61
62 lcd.fill((255, 0, 0))

```

```

63
64 pygame.display.update()
65 pygame.mouse.set_visible(False)
66
67 lcd.fill((0, 0, 0))
68 pygame.display.update()
69
70 # some utility functions
71 def constrain(val, min_val, max_val):
72     return min(max_val, max(min_val, val))
73
74
75 def map_value(x, in_min, in_max, out_min, out_max):
76     return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min
77
78
79 # let the sensor initialize
80 time.sleep(0.1)
81
82 while True:
83
84     # read the pixels
85     pixels = []
86     for row in sensor.pixels:
87         pixels = pixels + row
88     pixels = [map_value(p, MINTEMP, MAXTEMP,
89
90     # perform interpolation
91     bicubic = griddata(points, pixels, (grid_
92
93     # draw everything
94     for ix, row in enumerate(bicubic):
95         for jx, pixel in enumerate(row):
96             pygame.draw.rect(
97                 lcd,
98                 colors[constrain(int(pixel),
99                 (
100                     displayPixelHeight * ix,
101                     displayPixelWidth * jx,

```



```
102                 displayPixelHeight,  
103                 displayPixelWidth,  
104             ),  
105         )  
106  
107     pygame.display.update()  
108
```

Rendre le script exécutable et redémarrer votre Pi

```
1 | sudo chmod +x /home/pi/thermcam.py
```

Lancement du script au démarrage

Editez le fichier

```
1 | sudo nano /etc/xdg/lxsession/LXDE-pi/autostar
```

et ajoutez

```
1 @lxpanel --profile LXDE-pi  
2 @pcmanfm --desktop --profile LXDE-pi  
3 @xscreensaver -no-splash  
4 @python3 /home/pi/thermcam.py &
```

Screenshot

Voici une solution simple pour faire des screenshots de l'image rendu par la caméra thermique

Installation de scrot

```
1 | sudo apt install scrot
```

Création du script

```
1 sudo nano /home/pi/scrot.sh
```

et ajoutez les lignes

```
1 #!/bin/sh
2 LOCATION="$(date +/path/to/home/Pictures/shot:
3 mkdir -p $LOCATION
4 cd $LOCATION
5 DISPLAY=:0 scrot '%Y-%m-%d-%H%M.jpg' -q 20
```

modification des permissions

```
1 chmod u+x /home/pi/scrot.sh
```

essayez

```
1 cd /home/pi/
2 ./scrot.sh
```

Prendre un screenshot toutes les 15mn

```
1 crontab -e
```

ajoutez la ligne

```
1 */15 * * * 1-7 /home/pi/scrot.sh
```

Le fichier sera exécuté toutes les 15mn, du lundi au dimanche

Discussion

<https://forums.adafruit.com/viewtopic.php?f=50&t=143684&p=709531&hilit=raspberry+numpy#p709531>
(<https://forums.adafruit.com/viewtopic.php?f=50&t=143684&p=709531&hilit=raspberry+numpy#p709531>)

📌 ARG8833 ([HTTPS://ECO-SENSORS.CH/TAG/ARG8833/](https://eco-sensors.ch/tag/arg8833/))
AUTOSTART ([HTTPS://ECO-SENSORS.CH/TAG/AUTOSTART/](https://eco-sensors.ch/tag/autostart/))
CAMERA ([HTTPS://ECO-SENSORS.CH/TAG/CAMERA/](https://eco-sensors.ch/tag/camera/)) CRONTAB
([HTTPS://ECO-SENSORS.CH/TAG/CRONTAB/](https://eco-sensors.ch/tag/crontab/)) PITFT ([HTTPS://ECO-SENSORS.CH/TAG/PITFT/](https://eco-sensors.ch/tag/pitft/))
RASPBERRY ([HTTPS://ECO-SENSORS.CH/TAG/RASPBERRY/](https://eco-sensors.ch/tag/raspberry/)) TFT ([HTTPS://ECO-SENSORS.CH/TAG/TFT/](https://eco-sensors.ch/tag/tft/))
THERMIQUE ([HTTPS://ECO-SENSORS.CH/TAG/THERMIQUE/](https://eco-sensors.ch/tag/thermique/))

PREV POST

(<https://eco-sensors.ch/luma-oled-et-python-sur-un-raspberry/>)

NEXT POST

(<https://eco-sensors.ch/installation-de-jeedom-sur-un-raspberry-pi4/>)

Leave **Comment:**

Logged in as EcoSensors (<https://eco-sensors.ch/wp-admin/profile.php>). Log out? (https://eco-sensors.ch/wp-login.php?action=logout&redirect_to=https%3A%2F%2Feco-sensors.ch%2Fcamera-thermique%2F&_wpnonce=76dd262bbf)

 COMMENT

SUBMIT NOW

WP Armour (Only visible to site administrators. Not visible to other users.)

This form has a honeypot trap enabled. If you want to act as spam bot for testing purposes, please click the button below.

Act as Spam Bot

© Copyright 2020 EcoSensors. - Tous droits réservés.