



Plugins DAP Pymodaq Branly 2026



pymodaq_plugins_raspberrypi3

Notice d'Installation

Sommaire

Introduction.....	p.3
Architecture en deux parties.....	p.3
Prérequis matériel.....	p.4
Côté Raspberry Pi	
 système et environnement.....	p.5
Installation de l'OS.....	p.5
Interfaces matérielles.....	p.5
Environnement virtuel + dépendances.....	p.6
Lancement et mise en service automatique (systemd).....	p.6
Configuration matérielle (config.py).....	p.7
Côté ordinateur	
 plugin PyMoDAQ.....	p.9
Environnement Conda.....	p.9
Configuration du plugin (config_raspberrypi3.toml).....	p.10
Création du preset.....	p.11
Vérification.....	p.13

Introduction

Plugin

- `pymodaq_plugins_raspberrypi3` (PyMoDAQ ≥ 5.0)

Communication

- ZeroMQ DEALER (PC) \leftrightarrow ROUTER (Raspberry)
- TCP port 5555
- trames JSON

Architecture en deux parties

Partie	Emplacement
Plugin PyMoDAQ (<code>pymodaq_plugins_raspberrypi3</code>)	Ordinateur de supervision
Scripts serveur (<code>src_raspberry/</code>)	Raspberry Pi 3B

Les deux sont nécessaires.

Le plugin est modulable : les connexions de base (ZMQ côté réseau, I2C côté capteurs) peuvent être remplacées en changeant les classes Python correspondantes.

Prérequis matériel

- Raspberry Pi 3B + alimentation 5 V / 2,5 A + carte micro-SD (8 Go min.)
- Ordinateur Windows ou Linux, relié au Raspberry par Ethernet (ou même sous-réseau)
- Dispositif thermique câblé, exemple : 4 \times TMP102, 1 \times AHT10, 1 \times EMC2101 (I2C) ; 1 \times MOSFET + résistance RB50 + dissipateur ; 1 \times ventilateur ; alimentation Hi-Link 15 V

Séparation logique (3,3 V, bus I2C) / puissance (15 V, hacheur PWM) pour protéger le Raspberry.

Côté Raspberry Pi système et environnement

Installation de l'OS

1. Flasher **Raspberry Pi OS** sur la carte SD avec **Raspberry Pi Imager**.

Raspberry Pi OS (64-bit)

Compatible with

3B 3B+ 3A+ 4B 400 5 500
500+ CM3 CM3+ CM4 CM4S CM5
Zero 2 W

Raspberry Pi OS

A port of Debian Trixie with the Raspberry Pi Desktop

Release date	4 Dec 2025	Download
System	64-bit	Download torrent
Kernel version	6.12	View archive
Debian version	13 (trixie)	View release notes
Download size	1 266 Mo	
Storage required	6 152 Mo	

► SHA256 file integrity hash

2. Dans la personnalisation : noter le **hostname** et l'utilisateur, **activer SSH**.
3. Insérer la carte, démarrer, connecter au réseau. Connexion à distance via SSH (PowerShell ou MobaXterm).
4. Mettre à jour : `sudo apt update && sudo apt upgrade -y`.
IP statique possible via `nmtui`.

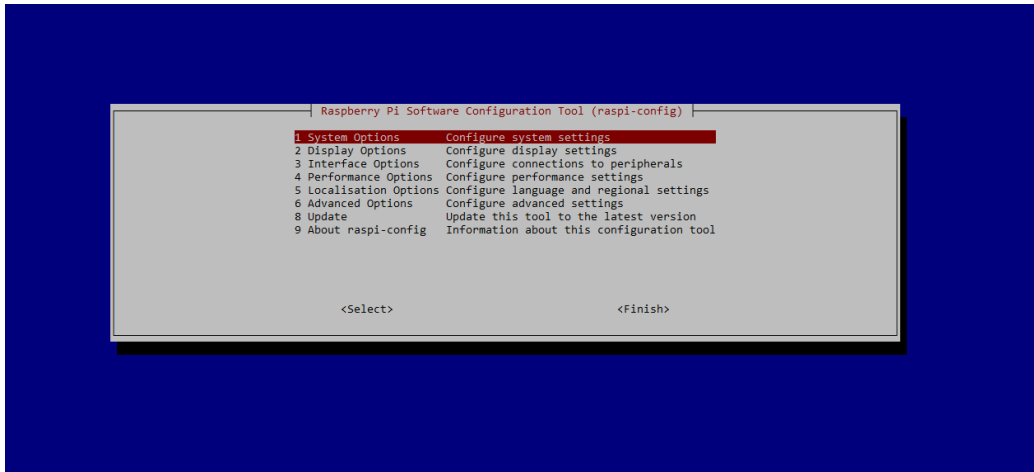
Interfaces matérielles

PWM matériel (pigpio) :

```
sudo apt install pigpio python3-pigpio
sudo systemctl enable pigpiod && sudo systemctl start pigpiod
```

Bus I2C :

```
sudo raspi-config          # Interface Options > I2C > Enable
```



Environnement virtuel + dépendances

```
cd /home/admin/PYMODAQ_PLUGIN_RASPPi3
python3 -m venv PY_ENV
source PY_ENV/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
```

requirements.txt > fige les versions exactes (extrait) :

```
pyzmq==27.1.0
smbus2==0.6.0
pigpio==1.78
RPi.GPIO==0.7.1
Adafruit-Blinka==8.69.0
adafruit-circuitpython-emc2101==1.2.12
...
```

Mode simulation automatique : si le bus I2C ou pigpio sont inaccessibles (ex. exécution sur un PC), le serveur bascule seul en simulation (drivers MagicMock/SIMULE). Pratique pour tester le réseau sans Raspberry.

Lancement et mise en service automatique (systemd)

Test manuel : `python main.py`.

En production, créer `/etc/systemd/system/pilotage.service` :

```
[Unit]
Description=Serveur ZeroMQ de pilotage thermique
After=network.target pigpiod.service

[Service]
ExecStart=/home/admin/PYMODAQ_PLUGIN_RASPPI3/PY_ENV/bin/python
/home/admin/PYMODAQ_PLUGIN_RASPPI3/main.py
WorkingDirectory=/home/admin/PYMODAQ_PLUGIN_RASPPI3
StandardOutput=inherit
StandardError=inherit
Restart=always
User=admin

[Install]
WantedBy=multi-user.target
```

⚠ Veiller à mettre **un espace** entre le binaire python et le chemin de `main.py` dans `ExecStart` (deux chemins séparés par un espace).

Puis :

```
sudo systemctl daemon-reload
sudo systemctl enable pilotage.service
sudo systemctl start pilotage.service
```

Configuration matérielle (config.py)

Toute la topologie est centralisée dans `src_raspberry/config.py` :

```
I2C_BUS_ID      = 1
VENTILATEUR_PIN = 18      # PWM 25000 Hz
RESISTANCE_PIN  = 23      # PWM 100 Hz (via MOSFET)

CAPTEUR_AHT10   = 0x38    # rh_sortie
CAPTEUR_TMP102 = 0x48    # t_resistance
# 0x49 t_dissipateur, 0x4A t_entree, 0x4B t_sortie (TMP102)
CAPTEUR_EMC2101 = 0x4C    # T_emc
```

Pour changer une broche ou une adresse, modifier `ACTUATORS_CONFIG` (actionneurs) ou `SENSORS_CONFIG` (capteurs) dans ce seul fichier.

Côté ordinateur plugin PyMoDAQ

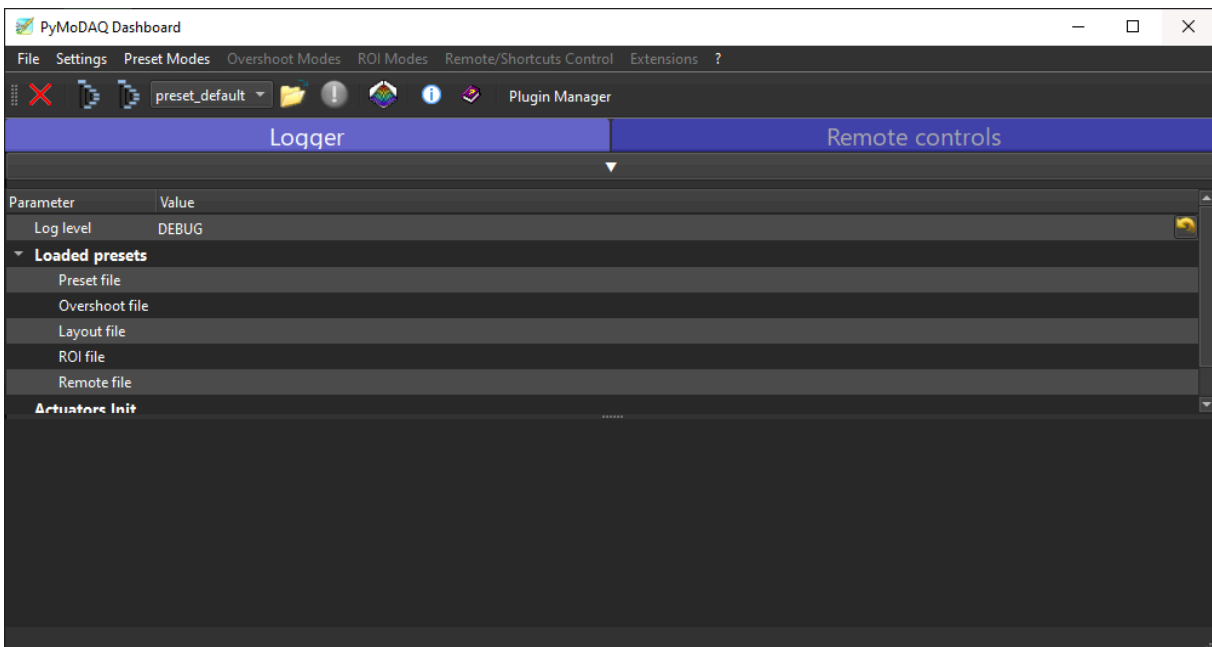
Environnement Conda

Dans Anaconda Prompt :

```
conda create -n PYMODAQ_DAP_2026 python=3.12
conda activate PYMODAQ_DAP_2026
pip install PyQt6
pip install pymodaq
pip install pymodaq_plugins_raspberrypi3 #doit être sur pipy
```

(En développeur : `pip install -e .` à la racine du dépôt cloné.)

Pour lancer l'interface, taper `dashboard` dans l'Anaconda Prompt :



Configuration du plugin (`config_raspberrypi3.toml`)

Le fichier de configuration utilisateur s'appelle `config_raspberrypi3.toml` et se trouve dans :

- Windows : `C:\Users\<UTILISATEUR>\.pymodaq\`
- Linux/macOS : `~/ .pymodaq/`

Le dossier `.pymodaq` n'existe qu'après avoir ouvert le Dashboard **au moins une fois**. Un modèle est fourni dans les fichiers du plugin dans : `resources/config_template.toml`.


Structure du TOML :

```
[RaspPi3]
address_Rasp = '192.168.235.2'    # Adresse IP du Raspberry
port = '5555'

[RaspPi3.ACTUATOR.COMPONENT1]
title = "Ventilateur"    # UNIQUE
name = "ventilateur"    # UNIQUE
units = "%"
min = "0"
max = "100"
address = "None"
pin = "00"

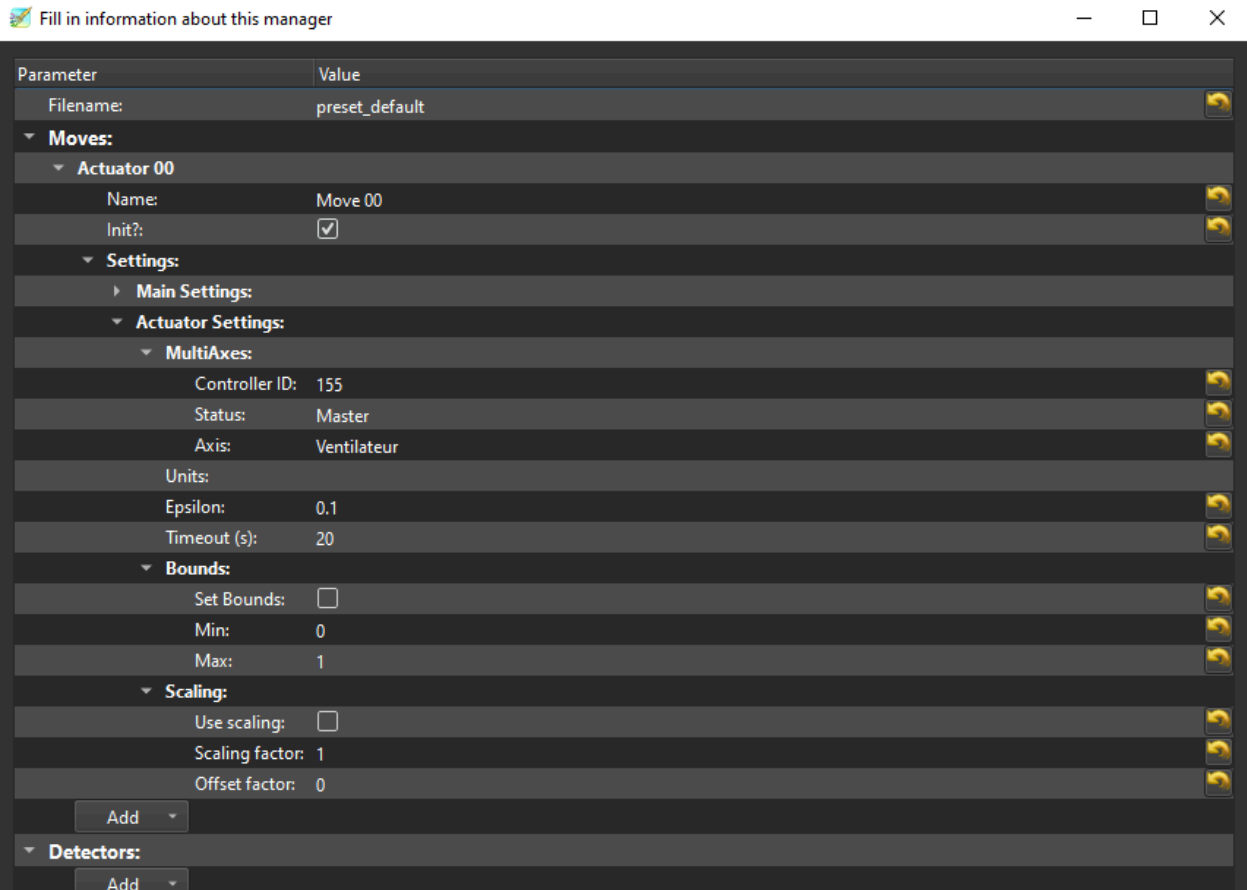
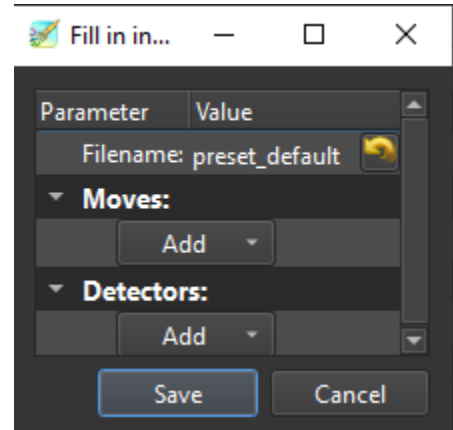
[RaspPi3.DETECTOR.COMPONENT1]
title = "t_resistance"    # UNIQUE
name = "t_resistance"    # UNIQUE
units = "°C"
address = "0x48"
pin = "None"
```

Règles : renseigner l'IP et le port (identique au Raspberry, 5555 par défaut) ; `title` et `name` **uniques** par composant.

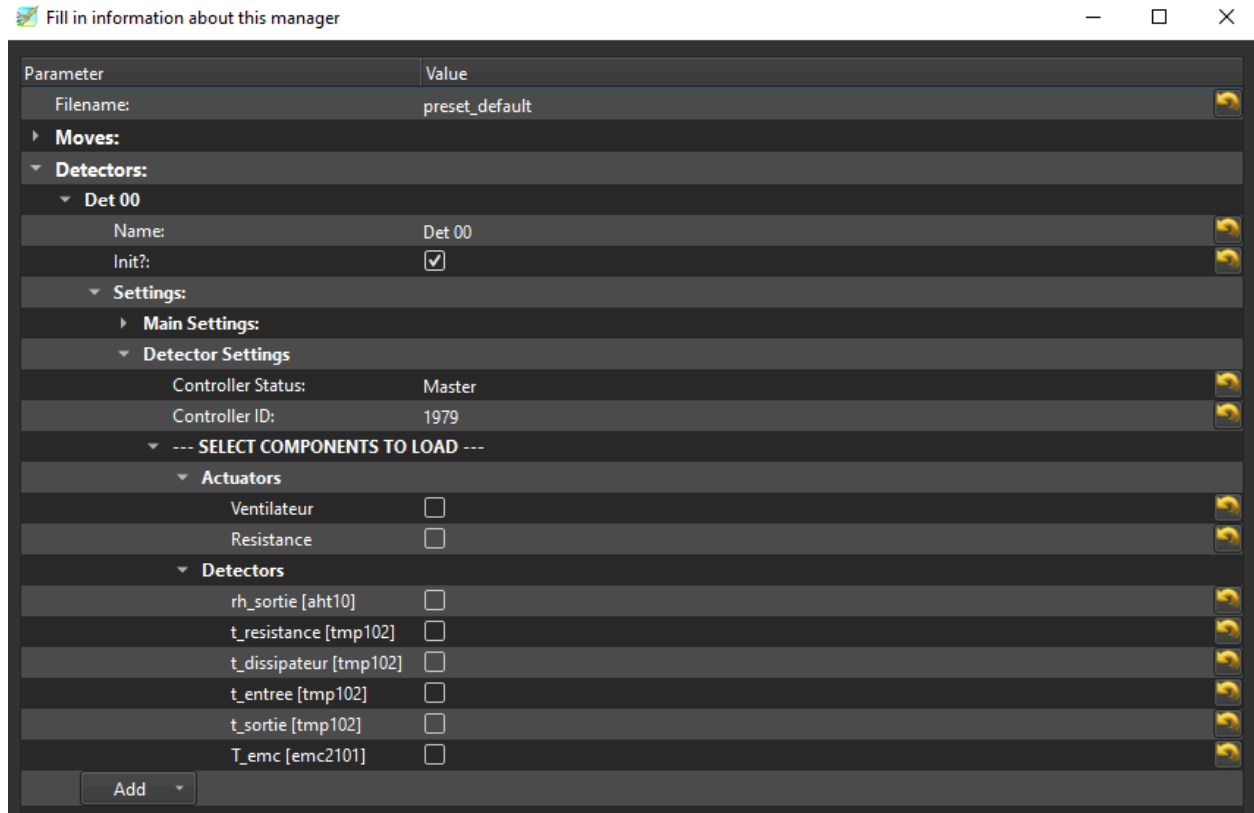
 PyMoDAQ ne signale pas les erreurs de syntaxe TOML : une faute empêche le chargement du plugin (sans message clair).

Création du preset

1. Dans le Dashboard : **Preset Modes** ► **New Preset**.
2. Renommer le preset.
3. **Add** dans *Moves* :
→ choisir **MoveRaspPi3** (actionneurs) ;
Add dans *Detectors* :
→ **ViewRaspPi3** (capteurs).
4. Côté MoveRaspPi3 : choisir l'*axis* (Ventilateur ou Resistance).



5. Côté ViewRaspPi3 : cocher les composants dans *SELECT COMPONENTS TO LOAD* (liste issue du TOML).



6. **Save**, puis sélectionnez le preset dans le menu déroulant et le charger (bouton fichier).

⚠ Avec la classe ZMQ d'origine, **ne pas** utiliser le système master/slave (Status + Controller ID) : des données peuvent passer d'un module à l'autre.

Vérification

- Raspberry : `sudo systemctl status pilotage.service` → **active (running)**.
- I2C : `sudo i2cdetect -y 1` → **adresses 38 48 49 4a 4b 4c**.
- PyMoDAQ : charger le preset, lancer une acquisition → la courbe s'affiche.

En cas d'erreur, voir la **notice de maintenance** (dictionnaire des messages d'erreur).