

# **CyPilot Pi 4 Installation**

---

### SUMMARY

1	Operating System .....	3
2	Install Cypilot.....	5
2.1	Install package .....	5
2.2	Complete installation.....	5
2.3	Customize configuration to match your boat specifications .....	6
3	Running Autopilot.....	6
3.1	Using Raspberry PI GUI.....	6
3.2	Using Web UI .....	6
3.3	Using Command line .....	6
3.3.1	Running Autopilot servers.....	6
3.3.2	Running Autopilot clients .....	7
3.3.3	Test and configuration scripts.....	7
4	How to Prepare system for development.....	7
4.1	Clone project from github .....	7
4.2	Install package .....	7
4.3	Complete installation from development environment.....	7
4.4	How to build a package .....	8
4.4.1	Install build if not already done .....	8
4.4.2	Build .....	8
5	Autopilot and Signal K server.....	8

## 1 Operating System

Current CysBOX software has been tested with Raspberry Pi 4 running Raspberry Pi OS bullseye (Debian version 11)

The latest Bullseye release are available from the following URL:

- Raspberry Pi OS with desktop

[https://downloads.raspberrypi.org/raspios\\_arm64/images/raspios\\_arm64-2022-09-26/2022-09-22-raspios-bullseye-arm64.img.xz](https://downloads.raspberrypi.org/raspios_arm64/images/raspios_arm64-2022-09-26/2022-09-22-raspios-bullseye-arm64.img.xz)

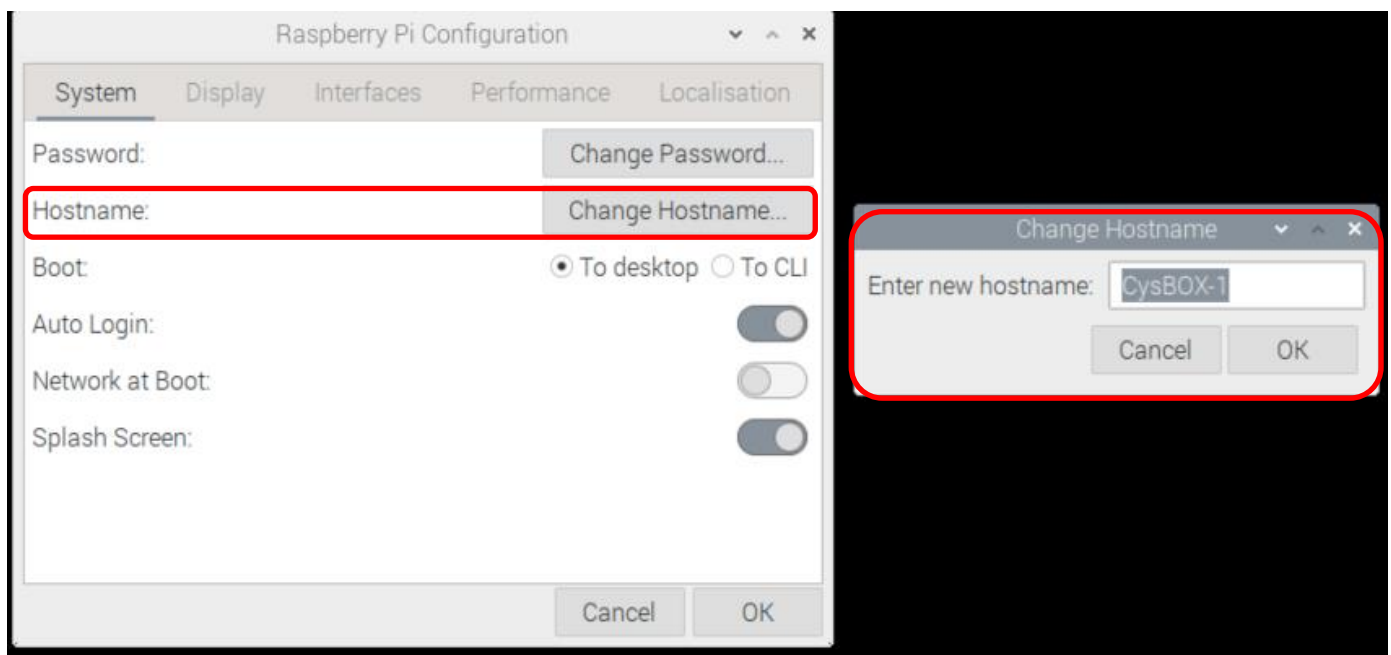
- Raspberry Pi OS Lite


[https://downloads.raspberrypi.org/raspios\\_lite\\_arm64/images/raspios\\_lite\\_arm64-2022-09-26/2022-09-22-raspios-bullseye-arm64-lite.img.xz](https://downloads.raspberrypi.org/raspios_lite_arm64/images/raspios_lite_arm64-2022-09-26/2022-09-22-raspios-bullseye-arm64-lite.img.xz)

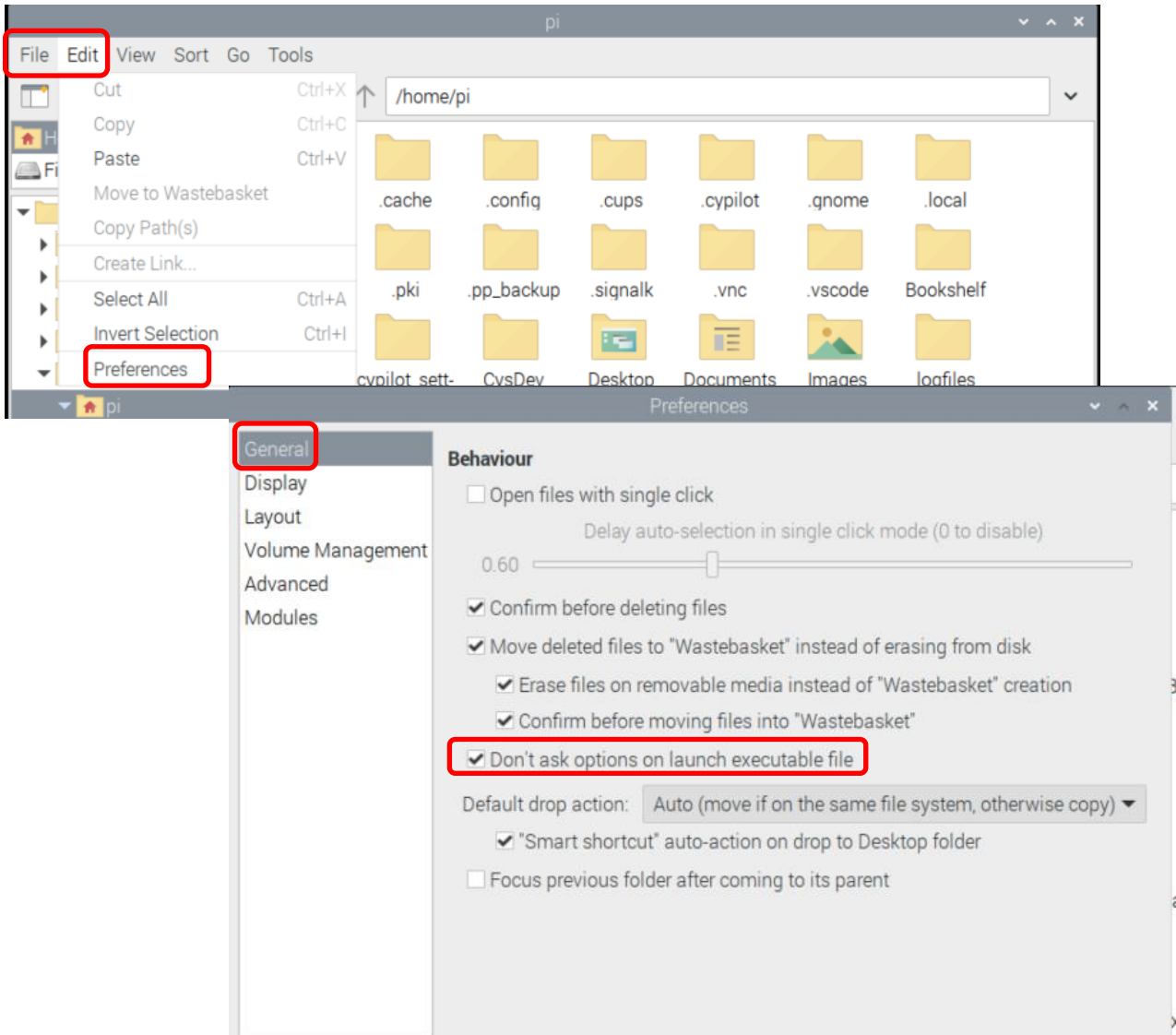
Download and install Etcher tool

<https://www.balena.io/etcher/>

- ❖ Launch Etcher software tool
- ❖ Click Flash from URL
- ❖ Copy the selected image URL to the “Use Image URL” field
- ❖ Select the SD card to flash and Flash!
  
- ❖ Insert SD card in the CysBOX
- ❖ Connect HDMI cable
- ❖ Connect mouse and keyboard cable
- ❖ Switch on and follow the install instructions (set user name **pi** pwd **cybele** and choose the Wi-Fi network and its password)  
*be patient! this operation is a bit long.*
- ❖ Restart
- ❖ When install is completed:
  - Menu → Preferences → Raspberry Pi configuration → Change Hostname (**CysBOX-1**)→OK



❖ Open  → Edition → Preferences → General



**Click on " Don't ask options on launch executable file".**

## 2 Install Cypilot

This is the basic installation for Autopilot standard user.

### 2.1 Install package

This is a complete suggested installation procedure for basic users.  
The minimal system configuration is as follow:

- RaspBerry PI4 (recommended memory : 8GB)
- OS version : Raspberry Pi OS 64-bit (bullseye)
- Kernel : 6.1.29 (GPIO I2C problems occured when using default 6.1.21 kernel)

While pip alone is sufficient to install from pre-built binary archives, up to date copies of the setuptools and wheel projects are useful to ensure you can also install from source archives:

```
sudo python3 -m pip install --upgrade pip setuptools wheel
```

Download **wheel file** from <https://cybele-sailing.com/software-package>

Install **cypilot** from wheel file:

```
sudo pip3 install cypilot-1.0.0-cp39-cp39-linux_aarch64.whl
```

note: to uninstall = `sudo pip3 uninstall cypilot==1.0.0`

### 2.2 Complete installation

After package has been installed, installation must be completed using the **cypilot\_init** command:  
**cypilot\_init**

At least, select **-1- Complete Autopilot Installation** from main menu, then from the opened submenu:

- step **-1 Install required packages**
- step **-2 Initialize configuration**
- step **-3 Create menu entries**
- then **-4 Return to main menu**

Reboot by selecting **-4-** in the main menu.

To install **signal-k** server select **step -2-** from main menu, install **signal-k dependencies** then when installing **signal-k server**, choose all default settings: **default location**, **update** or start from scratch, any **vessel name**, **MMSI** (if you have one), do not use port 80 (use **default port 3000**), **do not enable SSL**

---

### **2.3 Customize configuration to match your boat specifications**

Update the configuration files which are located in \$HOME/.cypilot directory:

- NMEA, COM, ... ports assignment
- sensor priorities
- ...

Calibrate IMU (mounting position, gyro, heading, ...) using **cypilot\_calibration** utility.  
See detailed documentation in the doc directory.

## **3 Running Autopilot**

### **3.1 Using Raspberry PI GUI**

Run **Autopilot UI** from Raspberry pi Menu: **AutoPilot > ApControl** or Double-click **ApControl** UI icon on PI desktop

The following menu shortcuts are installed ("Cypilot menu", desktop, and/or "Cypilot Tools" folder):

- Shortcuts to QT dialogs:
  - **ApCalibration** : QT calibration menu (command line : **cypilot\_calibration**)
  - **ApConfiguration** : QT generic configuration client menu (command line : **cypilot\_config**)
  - **ApControl** : QT main autopilot dialog menu (command line : **autopilot**)
  - **ApRun** : run autopilot server (command line : **cypilot**)
  - **AutotuneWizard** : Autotune configuration dialog (same as longpress on Autotune button in **ApControl** dialog)
  - **DashBoard** : Dashboard utility
  - **Learning\_GUI** : learning manager GUI (in development)
- Shortcuts to terminal windows:
  - **Learning\_Terminal** : learning manager terminal (in development)
  - **RC** : Remote-control test and configuration terminal (command line : **cypilot\_rc**)
- Web browser:
  - **SignalK** : SignalK
  - **WebUI** : Autopilot control

### **3.2 Using Web UI**

Run **Autopilot Web UI** from Raspberry pi Menu: **AutoPilot > Web UI**

When the UI or the Web UI is started, if no active server is detected, the Autopilot server is automatically launched, and it is closed on UI termination. When the Web UI is started, a local Chromium Web Browser is launched in kiosk mode, and the Autopilot can be controled from any Wifi browser at address <http://CysBOX-1:8000> The Remote Control device can be used to control the Autopilot as soon as the Autopilot server is running.

### **3.3 Using Command line**

#### **3.3.1 Running Autopilot servers**

These server scripts can be run as tests, **only one executes at a time**:

**cypilot** -- run the complete autopilot server: main pilot processes, sensors, servo communication, remote control receiver, ... The debug trace is displayed on the console windows. \* useful for testing the complete Autopilot server

Instead of running the complete autopilot these scripts provide a server with specific functionality:

**cypilot\_boatimu** -- run imu specific to boat motions \* useful for testing the imu (gyros) or even just reading gyros  
**cypilot\_sensors** -- convert and multiplex nmea0183 data reads nmea0183 from serial ports or from tcp connections, and multiplexes the output to tcp port 20220 by default \* convert and multiplex nmea0183 data  
**cypilot\_servo** -- use to test or verify a working motor controller is detected, \* can be used to control the servo

### 3.3.2 Running Autopilot clients

Run as many of these autopilot clients to connect to an Autopilot server:

**autopilot** -- Raspberry PI GUI to command autopilot (if not active, Autopilot server is launched)  
**autopilot\_web** -- Autopilot Web UI (Chromium started in kiosk mode, and if not active, Autopilot server is launched)  
**cypilot\_calibration** -- run Autopilot calibration dialog (IMU alignment, rudder calibration, settings)  
**cypilot\_config** -- allow simple access to Autopilot current data

### 3.3.3 Test and configuration scripts

**cypilot\_version** -- display software version  
**cypilot\_bno085** -- test and configure BNO085 IMU  
**cypilot\_ble** -- test and configure Calypso Anemometer

## 4 How to Prepare system for development

This is a complete suggested installation procedure for developers.

### 4.1 Clone project from github

We advise to use our common directory structure, so to clone git to `/home/pi/CysDev/cypilot`

### 4.2 Install package

While pip alone is sufficient to install from pre-built binary archives, up to date copies of the setuptools and wheel projects are useful to ensure you can also install from source archives:

```
sudo python3 -m pip install --upgrade pip setuptools wheel
```

Use wheel distribution from the dist directory `/home/pi/CysDev/cypilot/dist`

```
cd /home/pi/CysDev/cypilot/dist  
sudo pip3 install *.whl
```

### 4.3 Complete installation from development environment

To install with menu shortcuts pointing to development files, just run `python3 cypilot_init.py` from the development environment. At least, select `-1-` from main menu, then step `-1,2,3-` from the opened submenu, and `-4-` to reboot.

```
cd /home/pi/CysDev/cypilot/src/cypilot_init  
python3 cypilot_init.py
```

To build C extensions (linebuffer, servo, ...) in the development source tree:

```
cd /home/pi/CysDev/cypilot  
python3 setup.py build_ext
```

To install **signal-k** server select **step -2-** from main menu, install signal-k dependencies then when installing signal-k server, choose all default settings: default location, update or start from scratch, any vessel name, MMSI (if you have one), do not use port 80 (use default port 3000), do not enable SSL

### 4.4 How to build a package

The development environment must have been installed (see "Prepare system for development" just before)

#### 4.4.1 **Install build if not already done**

```
sudo pip3 install --upgrade build
```

#### 4.4.2 **Build**

```
cd /home/pi/CysDev/cypilot
```

Optionally clean previous build:

```
rm -r build rm -r dist
```

Build package:

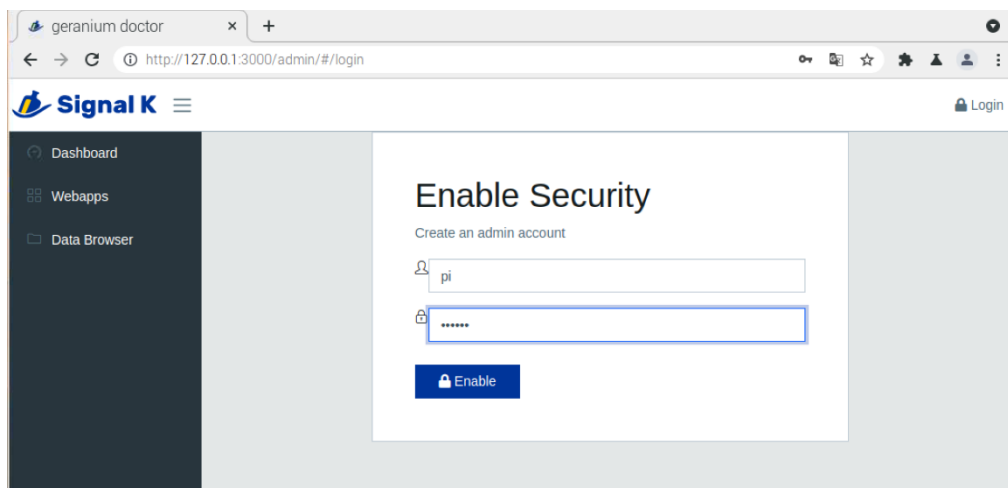
```
python3 -m build . --wheel
```

The built package is in dist repertory: **cypilot-1.0.0-cp39-cp39-linux\_aarch64.whl**

## 5 **Autopilot and Signal K server**

**NOTE:** When you first run the program

- ❖ Launch **Signal K** from desktop shortcut  
Create an admin **account pi and password password** and Enable

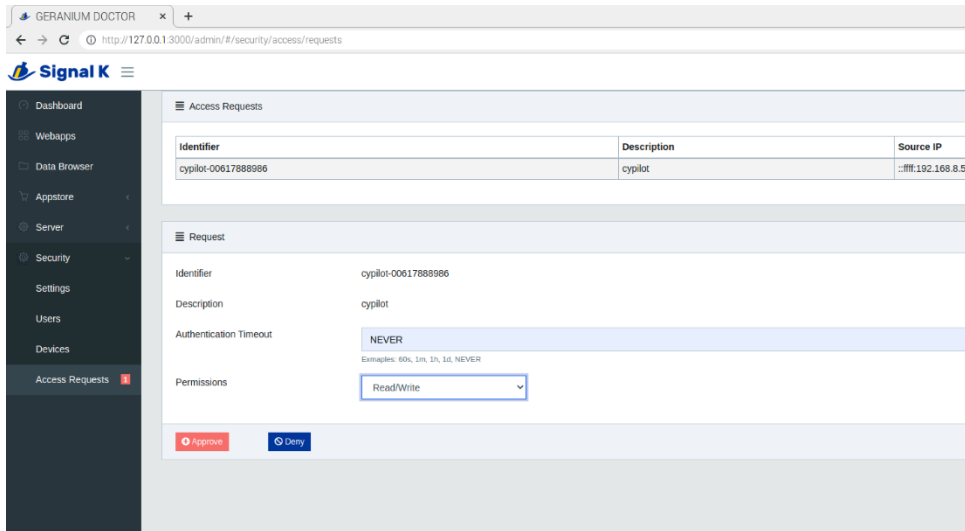


Create a user **account pi and password password** and Login

- ❖ Run Autopilot from "**ApControl**" desktop shortcut



- ❖ In the **Signal K** windows
  - To allow connexion to Signal K
    - **Security** → access requests → click on Identifier
      - Authentication Timeout: **NEVER**
      - Permissions **Read/Write**
      - Approve



- **Devices** → cypilot → click on Identifier
  - Permissions **Read/Write**
  - Apply

