4 System Setup Manual

4.1 Operating System

The Raspberry Pi OS is imaged onto the microSD card using the official Raspberry Pi Imager utility. First, the Imager software is downloaded from the Raspberry Pi website and installed on the host computer (source). Upon launch, Raspberry Pi Zero 2 W is selected as the target device [fig 5.30], followed by the recommended Raspberry Pi OS (64-bit) Debian Bookworm with desktop image [fig 5.31]. An SD-card reader is used to insert the microSD, which appears under Storage in the Imager interface [fig 5.32]. Before writing, the image may be customized: default username and password, Wi-Fi SSID and passphrase, device hostname, locale, keyboard layout, and SSH or VNC enablement. Once configuration is complete, the OS image is written to the card, producing a fully integrated, ready-to-use software environment.

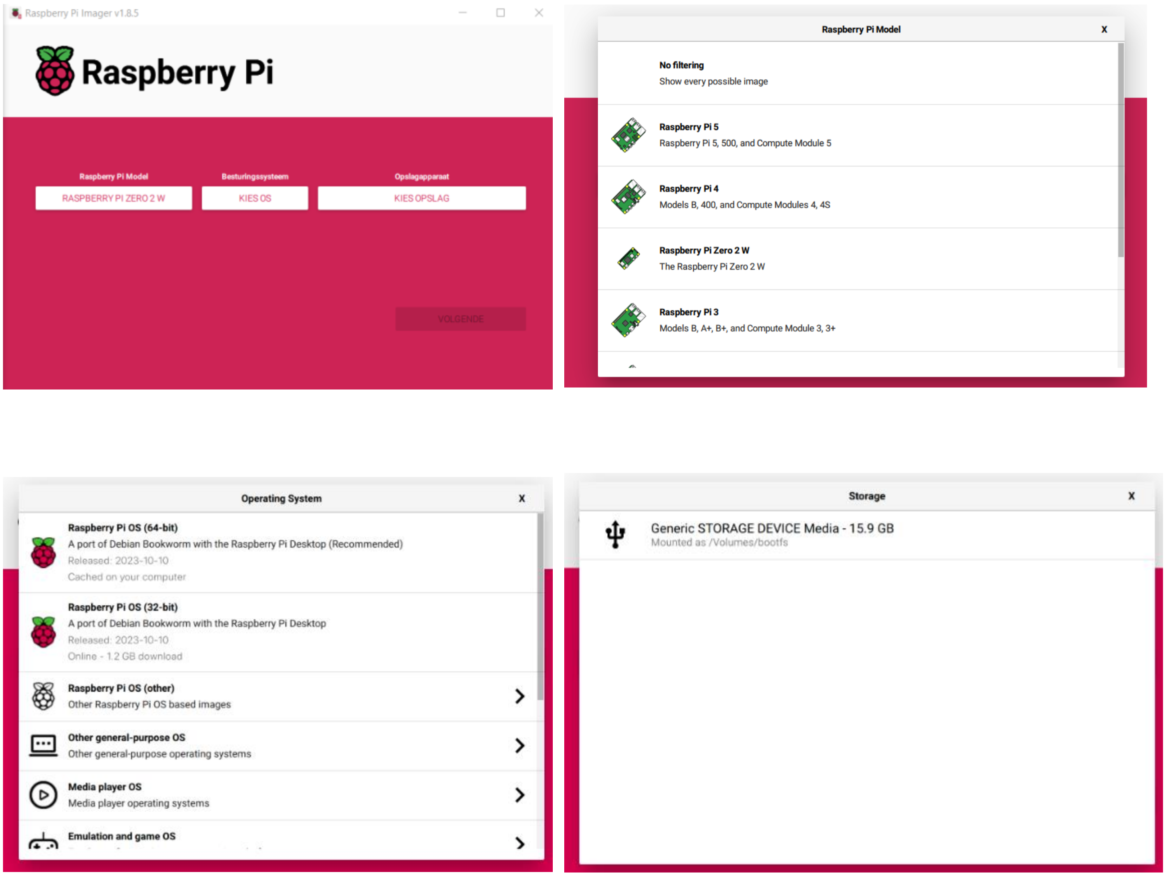

Figure 5.29 – 5.30 – 5.31 – 5.32: : DESCRIPTION

4.2 First-boot configuration: swapfile and updates

To allow initial GUI-based configuration, the Pi Zero 2 W can be connected to an external monitor (HDMI → mini-HDMI) and controlled via USB hub (micro-USB → USB-A) with keyboard and mouse. After booting into the desktop, the default 100 MB swapfile is increased to 1024 MB to improve stability under image-processing load; care should be taken to avoid excessive SD-card wear (by increasing the swapfile size too much) (source). This can be achieved using the terminal as follows:

1. Disable swap temporarily, allowing to adjust its size.

   Sudo dphys-swapfile swapoff

2. Edit the swap configuration file

   Sudo nano /etc/dphys-swapfile

Figure 5.33: : DESCRIPTION

3. Increase swap size
   - Make sure the CONF_SWAPSIZE is increased from 100 to 1024.
   - Save changes by CTRL + O, CTRL + X, confirm changes Y + ENTER.
4. Apply new swap size

   Sudo dphys-swapfile setup

5. Re-enable swap

   Sudo dphys-swapfile swapon

6. Reboot system

   Reboot

After reboot, the system is stable for all GUI and processing tasks. Regular package updates are then performed:

Sudo apt update

Sudo apt upgrade

4.3 Package Installation

The water-level detection software depends on both system-level camera drivers and a set of Python libraries. Before operation, ensure the following packages are installed on Raspberry Pi OS:

System & camera features

- Libcamera-apps
- Libcamera-dev
- V4l-utils

Core python libraries

- python3-picamera2 (Picamera2 API & libcamera bindings)
- python3-pillow (PIL image handling)
- python3-numpy (array operations)
- python3-matplotlib (plotting backend)
- python3-scipy (signal processing & statistics)
- python3-requests (HTTP for ORC API)
- python3-psutil (system metrics: CPU, memory)
- python3-rpi.gpio (GPIO control)

This can be achieved by running the following line in the terminal command prompt:

sudo apt update

sudo apt install -y libcamera-apps libcamera-dev v4l-utils python3-picamera2 python3-pillow python3-numpy python3-matplotlib python3-scipy python3-requests python3-psutil python3-rpi.gpio

4.4 AUTO-START SERVICE

To enable the water-level detection algorithm to launch automatically after any reboot, create a systemd service unit. This makes the system self-recovering. If it loses power or crashes, it will come back online and resume measurements without manual intervention. In conjunction with a power-management board, you can further minimize energy use by only powering the Pi when a measurement cycle is due (8.3 limitations: power management system).

Create and edit the service file with:

Sudo nano /etc/system/system/<filename>.service

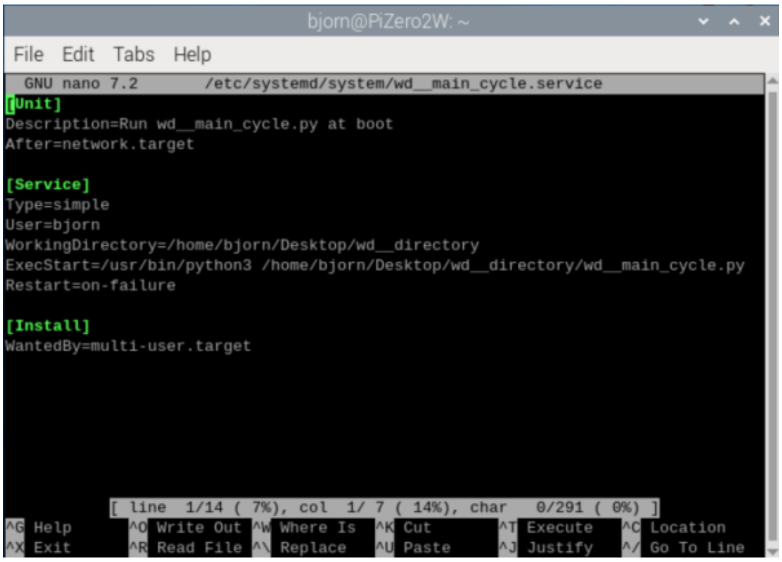

Figure 5.42: DESCRIPTION

This file will be available on <Github link>

The service file begins with a [Unit] section that names the task and tells systemd to wait until the network is online before starting. In the [Service] section, it specifies that systemd should run the Python interpreter on your wd__main_cycle.py script as the pi user, from the project's working directory, and automatically restart it if it ever crashes. Finally, the [Install] section hooks the service into the normal multi-user boot sequence so that enabling it causes the script to launch on every reboot without any further intervention.

After placing the service unit file into /etc/systemd/system/, the systemd manager must reload its configuration so that the new unit becomes available. The service can then be configured to start automatically on boot or prevented from doing so, and it supports manual start and stop operations as well as status inspection. This ensures that the water-level detection algorithm recovers automatically after power interruptions and can be managed interactively for troubleshooting or maintenance.

The relevant commands are:

sudo systemctl daemon-reload

sudo systemctl enable <filename>.service

sudo systemctl disable <filename>.service

sudo systemctl start <filename>.service

sudo systemctl stop <filename>.service

sudo systemctl status <filename>.service