# `tessie` User Manual

Urs Langenegger

*urslangenegger@gmail.com*

*Paul Scherrer Institute*
*CH-5232 Villigen-PSI, Switzerland*

April 15, 2024

**Abstract**

This document provides an overview of the `tessie` software and is intended to provide all information required to install the `tessie` software and safely operate the coldbox for the CMS phase-2 pixel module testing.
This document is work in progress. Please send all comments, in particular bug reports and complaints, to the email address given above. Many thanks!

# Contents

# 1 Introduction

The `tessie`[1] program controls all aspects of the safe operation of the PSI coldbox developed for the testing of the CMS phase-2 pixel modules. It is hosted in a github respository [1].

`tessie` is a multi-threaded C++ program running on the Raspberry Pi with a custom hardware "hat" inside the coldbox. Originally, it started as a Qt5 GUI (graphical user interface) and it can still be operated in that way. However, in a production setup, it is mostly controlled through a web interface. The threads in `tessie` control, respectively, the graphical display, the underlying hardware (CAN [2] and I2C [3] bus), and the MQTT messaging service [4].

The coldbox, sketched in Fig. 1, comprises eight positions, where TEPX modules can be positioned in thermal contact to Peltier elements, each controlled by custom `TEC` controllers (`TEC` is the abbreviation for ThermoElectric Cooler and a synonym for Peltier element). A centrally placed PCB accomodates the electrical connections and readout of the TEPX modules and furthermore hosts an SHT85 air and humidity sensor [5].

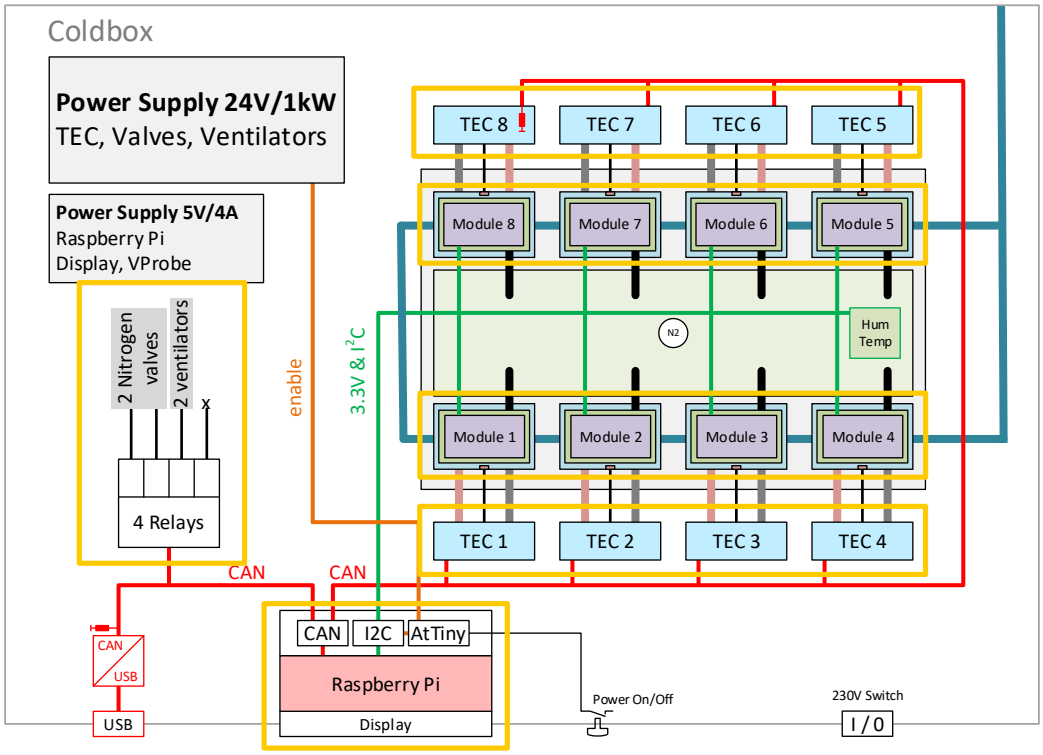


Figure 1: Sketch of the PSI coldbox showing the orientation and numbering scheme of the eight module positions, the CAN bus and I2C bus connection scheme, and the FRAS/4 Relays controlling the N2 flow. The thick blue line exiting at the top right indicates the water pipe to the chiller. Figure from Ref. [6]

[1] Etymology: tessie sounds better than TC (box), for temperature cycling (box)

## 2 Installation

We assume that you have a coldbox where the hardware is completely configured according to the instructions [7]. In this section we describe how to prepare the Raspberry Pi, starting from creating its boot device, installing all required software components, and setting up the automatic `tessie` startup at boot time.

A few important remarks before describing the installation procedure.

- If the Raspberry Pi's screen turns white (at the end of the boot process) very likely the screen flatband connector is not properly inserted. This can happen easily when inserting the SD card.
- It has been observed that a few power-cycles are required to have the touchscreen work properly (instead of displaying "nothing", which can mean a white screen or a black screen). Alternatively, try to connect via ssh and do (in a terminal) `sudo shutdown -r now`. Note that "nothing" is not the same as the white screen indicating a flatband cable-connector issue.

  It seems that this issue is due to newer releases of Debian version 12 (bookworm) in image files dated 2024-03-13 and 2024-03-15. It is not present in the image file dated 2023-12-05, referred to below.
- The following instructions have been tested verbatim (line by line copy-paste) with a Raspberry Pi 4 Model B Rev 1.4 with 8 GB RAM. Please provide feedback if you run into problems with a different Raspberry Pi.

Installing `tessie` is straightforward, if the following steps are followed.

- Using the "Rasberry Pi Imager" [8], available for macOS, Windows, and Linux, burn a SD card with the 2023-12-05 image file, available from

  `https://downloads.raspberrypi.com/raspios_arm64/images/raspios_arm64-2023-12-06/2023-12-05-raspios-bookworm-arm64.img.xz`

  Note the mismatch between the directory name and the file name. It is recommended to apply a few changes to the default setup, in particular set the user name and password, the hostname, and allow ssh access for remote work, this is available from the Imager after you have specified the model, the OS, and the storage device (the SD card).
- Insert the SD card into the foreseen slot beneath the touchscreen flatband cable connector and power up the Raspberry Pi, i.e., plug in the (USB-C) power cable. If the touchscreen goes white, it is likely due to the "big" flatband connector not being plugged in correctly. Try again.

  At the end of the startup/boot sequence you should see a nice background image.
- *Installation of dependencies* Open a terminal (it should be accessible from one of the icons at the top of the display) and do the following:

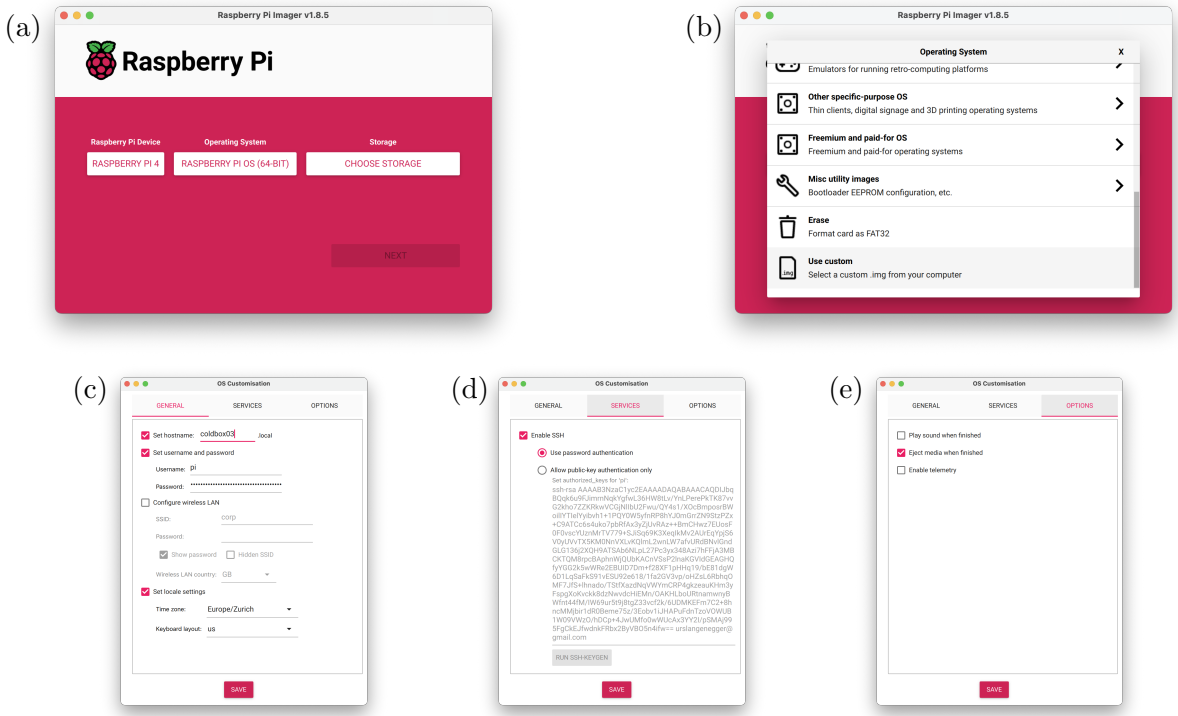


Figure 2: Screen shot of the Raspberry Pi Imager and customization examples. (a) Select the model and OS, (b) choose 'EDIT SETTINGS' to reach the lower panels, (c) enter the hostname and set a password for the default user 'pi' (use this!), (d) enable ssh connections, (e) if you want.

```
sudo apt install -y nodejs
sudo apt install -y npm

sudo apt-get install -y pigpio

sudo apt-get install -y libmosquitto-dev libmosquittopp-dev
sudo apt install -y mosquitto mosquitto-clients

sudo apt install -y libqt5charts5 libqt5charts5-dev

sudo apt-get install -y nginx

sudo date -s "Mon Apr 8 17:04:00"
```

Do enter the correct day, date, and time in the line above.

- *Installation of tessie* Get the `tessie` software and compile it:

```
cd /home/pi
```

```
git clone https://github.com/ursl/tessie.git
cd tessie/test1
qmake -o Makefile  test1.pro
make -j2
```

Note: In case you want to compile `tessie` on a non-Raspberry Pi host without I2C/CAN bus, invoke `qmake "CONFIG+=NOPI" -o Makefile test1.pro`.

- *Mosquitto setup* Using the `nano` editor in `sudo` mode, edit the file `/etc/mosquitto/mosquitto.conf` to contain the following two lines:

```
listener 1883
allow_anonymous true
```

In case the above instructions are unclear, the following is what you should type into the terminal: `sudo nano /etc/mosquitto/mosquitto.conf`, jump to the end, insert the two lines, and exit the editor (using in sequence: `CTRL-x  y  RET`).

- *Hardware (I2C and CAN) bus configuration* Using the `nano` editor in `sudo` mode, edit the file `/boot/firmware/config.txt` to contain the following two lines:

```
dtparam=spi=on
dtoverlay=mcp2515-can0,oscillator=12000000,interrupt=25
dtoverlay=spi-bcm2835-overlay
dtparam=i2c_vc=on
```

- *Hardware power buttton configuration* Download the auxiliary software package and install it:

```
cd /home/pi
git clone https://github.com/Howchoo/pi-power-button.git
./pi-power-button/script/install
```

- *Splash screen configuration* (Note: This is not compulsory. There is no real need to change the splash screen.) Using the `nano` editor in `sudo` mode, edit the file `/boot/firmware/cmdline.txt` to contain *on one line* the following two lines (they are provided here on two lines such that they can be copied in their entirety):

```
console=serial0,115200 console=tty1 root=PARTUUID=7a0cea11-02  rootfstype=ext4
fsck.repair=yes rootwait quiet splash plymouth.ignore-serial-consoles
```

Using the `nano` editor in `sudo` mode, edit the file `/boot/firmware/config.txt` to contain

```
disable_splash=1
```

Enter the following in a terminal:

```
cd /usr/share/plymouth/themes/pix/
sudo mv splash.png splash.png.bac
sudo cp /home/pi/tessie/splash.png ./
```

- *tessie webserver* Setup the tessie web server by installing all required node packages

```
cd /home/pi/tessie/node/test1
npm install --save express socket.io mqtt
```

- *tessie startup at boot time* With `sudo nano` create the file `/lib/systemd/system/tessie.service` with the following content (i.e. do `sudo nano /lib/systemd/system/tessie.service` and copy-paste the following):

```
[Unit]
Description=tessie
After=network.target

[Service]
Type=idle
Environment="XAUTHORITY=/home/pi/.Xauthority"
Environment="DISPLAY=:0"
WorkingDirectory=/home/pi/tessie/test1
ExecStartPre=/home/pi/tessie/resetCAN.sh
ExecStart=/home/pi/tessie/test1/tessie
StandardOutput=inherit
StandardError=inherit

[Install]
WantedBy=graphical.target
```

For the tessie webserver do `sudo nano /lib/systemd/system/tessieWeb.service` with the following contents

```
[Unit]
Description=tessie
After=multi-user.target

[Service]
Type=idle
WorkingDirectory=/home/pi/tessie/node/test1
ExecStart=/usr/bin/node /home/pi/tessie/node/test1/server3.js

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

- *Configure nginx* This optional section allows connecting to `http://coldbox03` instead of `http://coldbox03:3000`. Create the `nginx` configuration file with the command `sudo nano /etc/nginx/sites-available/default` and replace the contents of the file with the following contents

```
server {
    listen 80;
    server_name coldbox03.psi.ch;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

  Do change `coldbox03.psi.ch` to your coldbox hostname and domain! Be careful when copy-pasting the inverted commas! Start the service with

```
sudo service nginx start
```

- *Startup services* Now enable the startup of the two low-level components at boot time plus tessie and its webserver

```
sudo systemctl enable pigpiod
sudo systemctl enable mosquitto.service

sudo systemctl enable tessie.service
sudo systemctl enable tessieWeb.service
```

  You can always monitor the status of these "services" with

```
systemctl status tessie
systemctl status tessieWeb
```

Now reboot the system, e.g., with `sudo shutdown -r now`. If the shutdown process gets stuck, hit the central power button. If all goes well, the touchscreen of the Raspberry Pi will show the GUI featured in Fig. 3. You can connect from any PC.
The normal manual way to interact with `tessie` is through a webbrowser. Point your favorite browser to `http://coldbox03`, cf. Fig 4.

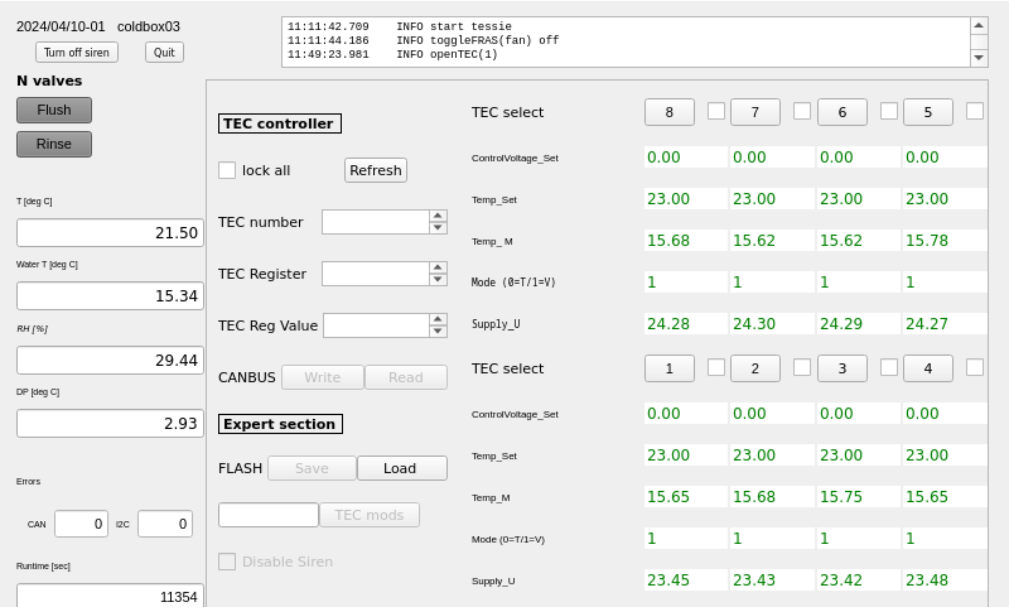


Figure 3: The GUI appearing on the Raspberry Pi touch screen.

# 3 Upgrades

## 3.1 `tessie` Upgrades

Upgrading **`tessie`** is straightforward in a terminal:

```
cd /home/pi/tessie
git pull
git checkout 2024/04/10-01
cd test1

qmake -o Makefile  test1.pro
make -j2
```

The command git checkout "tags/2024/04/10-01" is not compulsory, and you can work with the HEAD of the master branch (which is what you get when simplying cloning the repository). However, for production systems, it is better to work with a specific tag (in this example '2024/04/10-01'). Note the git message about being in 'detached HEAD' state. Unless you intend to do code development, you can safely ignore it. If you want to do code development and commit your changes, read the rest of the warning message and do as told.
You can check which tags are available with

```
cd /home/pi/tessie
git --no-pager tag
```

If you want to go to the HEAD of the master branch (where all development takes place), do

```
cd /home/pi/tessie
git checkout master
```

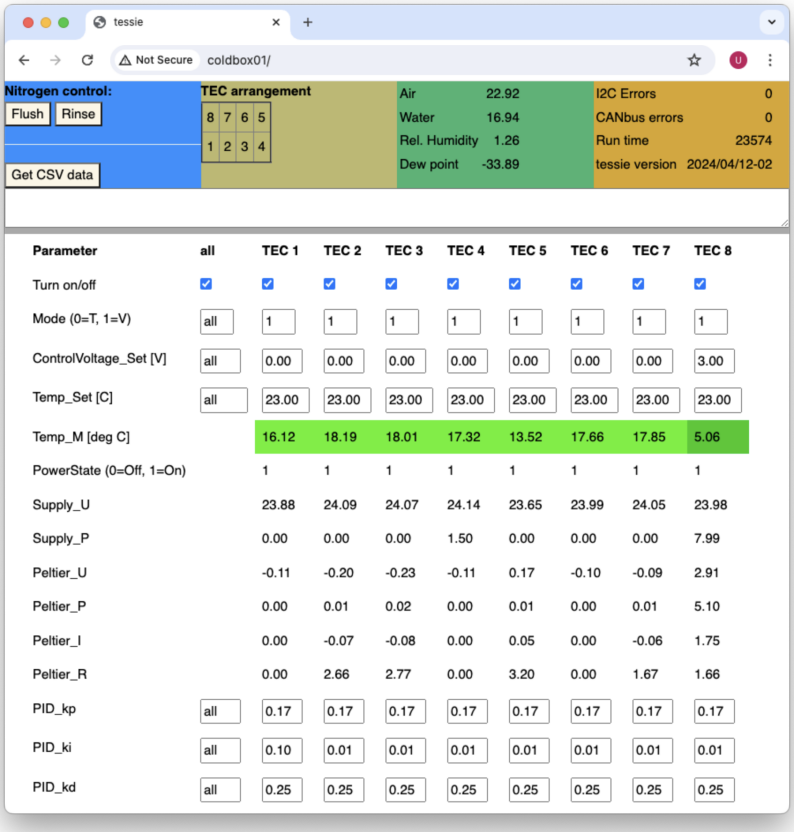


Figure 4: Web graphical interface to `tessie`. The top right brown-yellow box will turn red in case the connection to `tessieWeb` is interrupted.

To make the changes take effect, it is sufficient to restart the `tessie` service:

```
sudo systemctl restart tessie
```

There is no need to restart the web service `tessieWeb`.
Alternatively, you can reboot the coldbox, either by turning it off/on (pressing the central power button) or in a terminal:

```
sudo shutdown -r now
```

In both cases, you can verify the update by comparing the version string of the GUI displayed on the Raspberry Pi's touch screen and the web GUI (a reload of the page may be required).

### 3.2 TEC firmware upgrades

FIXME

# 4 Operational aspects

## 4.1 Direct readout of probe card

In principle, the probe card [9] readout should be handled by higher-level software. However, it is also possible to do a direct readout of the probe card with MQTT. In one terminal subscribe to the `ctrlTessie` thread with

```
mosquitto_sub  -h coldbox01 -t "ctrlTessie"
```

In a second terminal, assuming that you have a probe card at slot 8, issue the read command

```
mosquitto_pub -h coldbox01 -t "ctrlTessie" -m "get vprobe8"
```

You will receive, in the first terminal (the one where you have subscribed to the `ctrlTessie` thread), two lines with the following format:

```
get vprobe8
2024/04/12 11:34:40 -0.00040422 0.077611 -0.10232 0.10368 0.10343 5.0528e-05 0.10353 5.0528e-05 -0.10348 0.00025264
```

The first line repeats the command given in the second terminal and then the result of that readback is provided. The interpretation of the numbers is as follows

```
date time  vin  voffs  vdda0  vddd0   vdda1  vddd1   vdda2 vddd2   vdda3  vddd3
```

corresponding to the input voltage, offset, and the read digital and analog voltages of the 4 chips on the module.
FIXME: if you have better information that could be added here, please inform me.

## 4.2 Traffic Lights

Three lights are used for a visual display of the operations status of the coldbox, cf. Table 1.

Table 1: "Traffic light" display of the `tessie` status.

| Color | State | Meaning |
|---|---|---|
| Green | on | Safe to open the box, all environmental parameters in safe range |
| Green | off | Not safe to open the box |
| Yellow | on | At least one `TEC` turned on (e.g., during a test) |
| Yellow | blinking | No `TEC` turned on, but not safe to open the box. |
| Yellow | off | No `TEC` turned on |
| Red | on | Alarm active (see section 4.3) |
| Red | off | No alarm active |

## 4.3 Alarm channels

`tessie` raises an alarm in case operational issues require human intervention. The alarm is raised as soon as `tessie` observes a measurement violatiung the safe operation region. The alarms are propagateed via various means

- the alarm condition is broadcast to the `ctrlTessie` and `monTessie` MQTT channels

- an alarming sound is played through an attached loudspeaker inside the coldbox. All connected web GUIs will also play the sound in case the user has given the browser (tab) permission to play audio.
- the "traffic" light display of the coldbox displays a constant red light

In addition to the alarms raised, `tessie` also issues warnings for I2C and CAN bus errors. These warnings are accumulated as counters in the GUIs (both the web GUI and the GUI running on the coldbox touch screen). In addition, an warning sound is played.

### 4.4 Safe operations limits

To ensure the safety of the coldbox equipment, the `TEC` controllers and the FRAS relais require a "heartbeat" command from `tessie` at periodic intervals (30 seconds and 3 seconds, respectively; `tessie` sends the command every second). If that expected "heartbeat" signal is not registered, these components stop operating (*e.g.*, they no longer cool the `TEC`).

In addition, `tessie` continuously monitors environmental parameters to ensure a safe operation of the coldbox in case of operator error. Table 2 provides a summary of the safe operation parameters. An alarm is raised if `tessie` registers a violation, *cf.* Section 4.3.

Table 2: Safe operations parameters monitored by `tessie`. In this table "module temperature" indicates the PT1000 temperature reading mounted on the Peltier module.

| **Parameter** | **min** [deg] | **max** [deg] |
|---|---|---|
| box air temperature | n/a | 40 |
| water temperature | n/a | 30 |
| module temperature | n/a | 30 |
| difference between box air temperature and dew point | 2 | n/a |
| difference between module temperature and dew point | 2 | n/a |

## References


[1] Urs Langenegger, "tessie". https://github.com/ursl/tessie.

[2] Steve Corrigan, "Introduction to the Controller Area Network (CAN)". https://www.ti.com/lit/an/sloa101b/sloa101b.pdf.

[3] Jonathan Valdez, Jared Becker, "Understanding the I2C Bus". https://www.ti.com/lit/an/slva704/slva704.pdf.

[4] MQTT.org, "MQTT: The Standard for IoT Messaging". https://mqtt.org/.

[5] Sensirion - The sensor company, "Datasheet SHT85 - Humidity and Temperature Sensor". https://sensirion.com/media/documents/4B40CEF3/61642381/Sensirion_Humidity_Sensors_SHT85_Datasheet.pdf.

[6] Noah Piqué, "Coldbox Development at PSI". CMS Inner Tracker Modules Meeting, November 30, 2023 (https://indico.cern.ch/event/1349094/).

[7] Noah Piqué, "Coldbox assembly manual". https://psi-lab.docs.cern.ch/coldbox/assembly/.

[8] Raspberry Pi OS, "Raspberry Pi Imager". https://www.raspberrypi.com/software/.

[9] Beat Meier, et al., "Voltage probecard". https://psi-lab.docs.cern.ch/coldbox/vprobe/ https://gitlab.cern.ch/psi/moduleprobe/.