`tessie` User Manual

Urs Langenegger

urslangenegger@gmail.com

Paul Scherrer Institute
CH-5232 Villigen-PSI, Switzerland

May 3, 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. For future reference, the necessary hardware information is also included.
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].

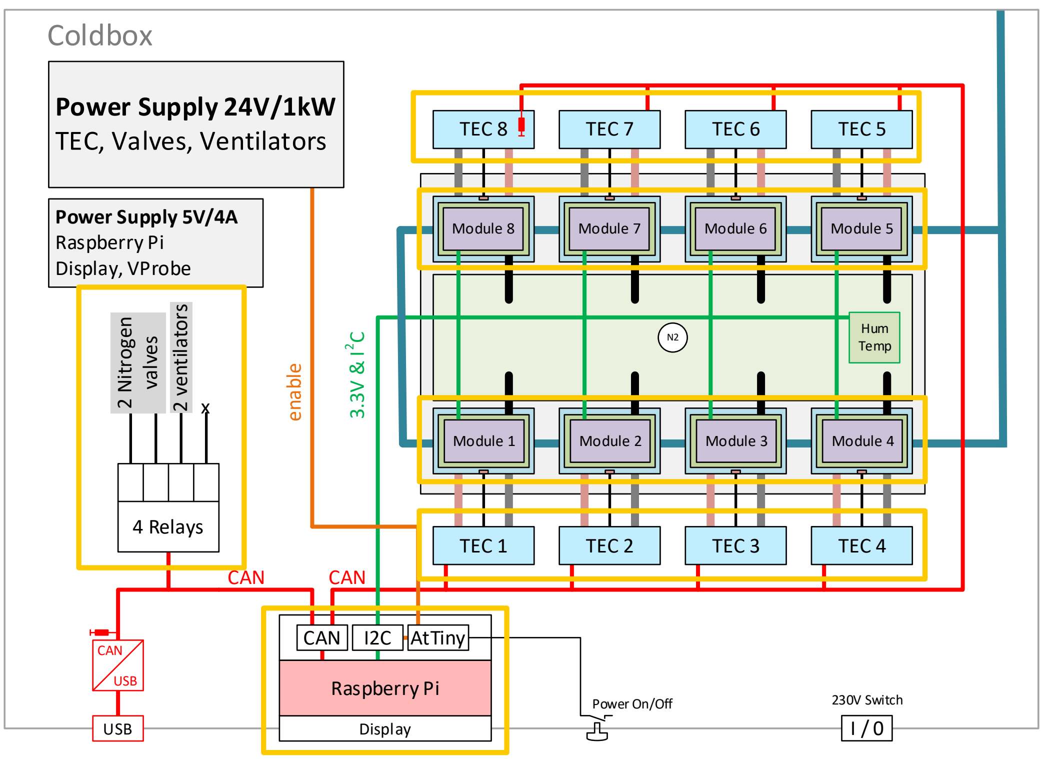

Figure 1: Sketch of the PSI coldbox showing the orientation and numbering scheme of the eight module positions, the CAN bus and I2C bus connection scheme, and the FRAS/4 Relays controlling the N2 flow. The thick blue line exiting at the top right indicates the water pipe to the chiller. Note that `TEC` numbering starts at 1, not 0 (zero). Figure from Ref. [6]

The coldbox can be operated in a manual direct manner and through scripts. Manual direct operation includes for example

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

- Opening and closing the lid to load or unload modules for testing
- Turning on/off the N2 flow controls
- Controlling the `TEC` parameters through the touchscreen GUI (not recommended) or the web GUI interface (recommended) as well as turning on/off the `TEC`.

In addition, it may be required to log into the coldbox Raspberry Pi using `ssh` to update and recompile the code and/or restart the `tessie` program and/or its webserver.

2 Hardware aspects

2.1 Raspberry Pi

The coldbox is operated with a Raspberry Pi Model 4B. Table 1 provides an overview of `tessie`'s pin usage of the `J8` connector on the `Raspberry Pi` (*cf.* the result of the `pinout` command issued on the `Raspberry Pi`).

Table 1: `Raspberry Pi` pins (physical and GPIO numbering scheme) used by `tessie`.

phys.	GPIO	Name	Remarks
11	17	GPIORED	red light of traffic light
13	27	GPIOYELLO	yellow light of traffic light
15	22	GPIOGREEN	green light of traffic light
16	23	GPIOINT	interlock for external usage
18	24	GPIOPSUEN	always `HIGH` (not used to date)

2.2 `TEC` controller

The `TEC` controllers are a central element of the coldbox and allow the direct or automated (PID) control of the `TECs`, in addition to their powering and monitoring. A coldbox houses 8 `TEC` controllers, one for each module position. The `TEC` controllers are attached to the CAN bus and have board IDs (addresses) 1. . . 8.
FIXME: Discuss CAN bus communication
FIXME: List all registers, commands, etc.

2.3 Module probe card

The module probe card [7], illustrated in Fig. 2 and sometimes referred to as "Vprobe" below, is used to measure voltage levels on test points on all four chips of a module. Since its readout is connected to the coldbox I2C bus, `tessie` is used to obtain the readings of this probe card.

2.4 Environmental monitoring with the Sensirion SHT85

FIXME

2.5 CAN controlled FRAS/4 Relays

FIXME

2.6 Lid-closure sensor

FIXME
4200 when open, something else (possibly negative) when closed. To be seen.

2.7 Cross section of module cooling infrastructure

The cross section of the module cooling infrastructure is shown in Fig. 3. A few remarks may be useful.

- The PT1000 providing the temperature reading for the `TEC` register called Temp_M is not in thermal contact with the module but rather with the `TEC` (the Peltier module).
- The module has a thermal power of about 10-12 W (after accounting for voltage drops between the power supply and the module). This implies that the temperature on the module is substantially higher (of the order 15-20°C) on the module than the `TEC`. This is discussed and illustrated in Fig. 7 in section 5.7.

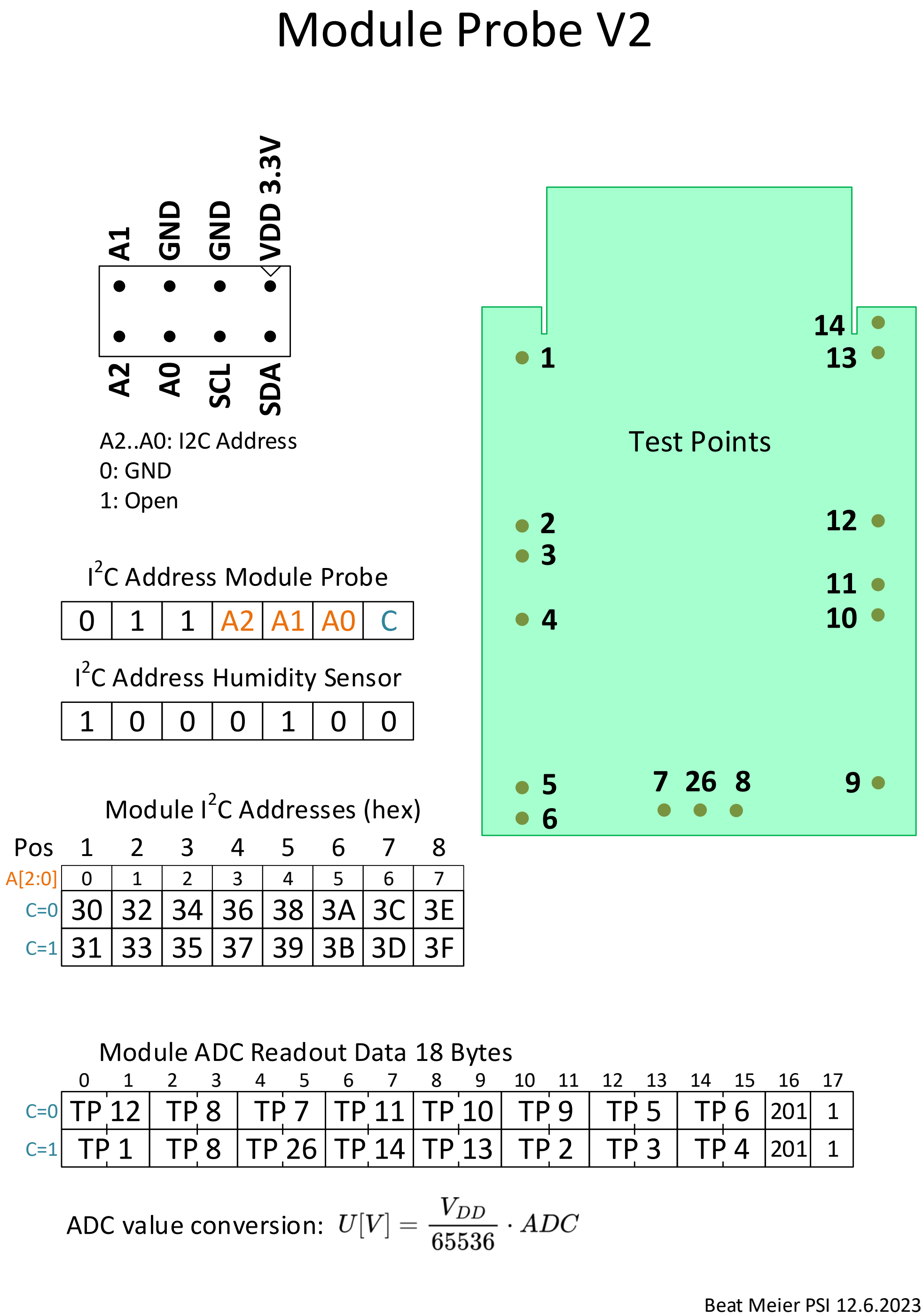

Figure 2: Module probe card documentation [7]. Note that $V_{DD} = 3.3114$, with a permille-level variation between boards. In the readout, bytes 16 have the value 201 as cross-check, and bytes 17 indicate the software version of the board.

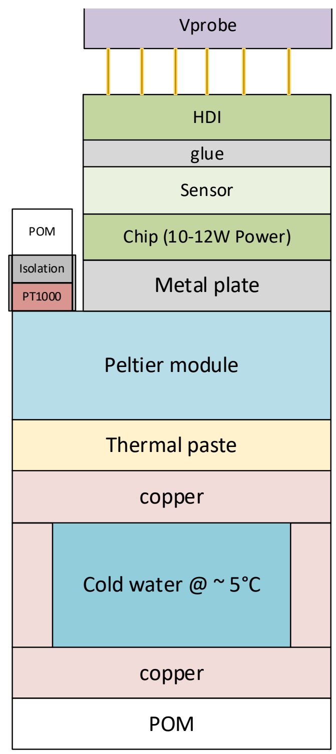

Figure 3: Cross section of the cooling infrastructure for a single module inside the coldbox. From the top: "Vprobe" is the probe card (subsection 2.3), the structures from the "HDI" down to and including the "Chip" represent the module, positioned on a metal plate in direct contact to the `TEC` (Peltier module). The cold water from the chiller is circulating in a copper enclosure. Note the placement of the PT1000 measuring the "`Temp_M`" register of the `TEC` controller: despite its name, it does *not* measure the module temperature but rather the `TEC` temperature of the module!

3 Installation

We assume that you have a coldbox where the hardware is completely configured according to the instructions [8] and connected to the internet. 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.
- There is a script-based installation procedure provided by Branislav (Bane) Ristic at his forked repository. Depending on the image you start with, his setup may work well for you. It should be rather straightforward to merge that branch with the `master` branch of the main `tessie` repository since it is only about installation.

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

- Using the "Rasberry Pi Imager" [9], 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`

  It is recommended to apply a few changes to the default setup as illustrated in Fig. 4, 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.

  On its first power-up, the Raspberry Pi will reboot various times. Be patient.

  At the end of the startup/boot sequence you should see a bluish background image showing a cormorant fisherman of Guilin, China.

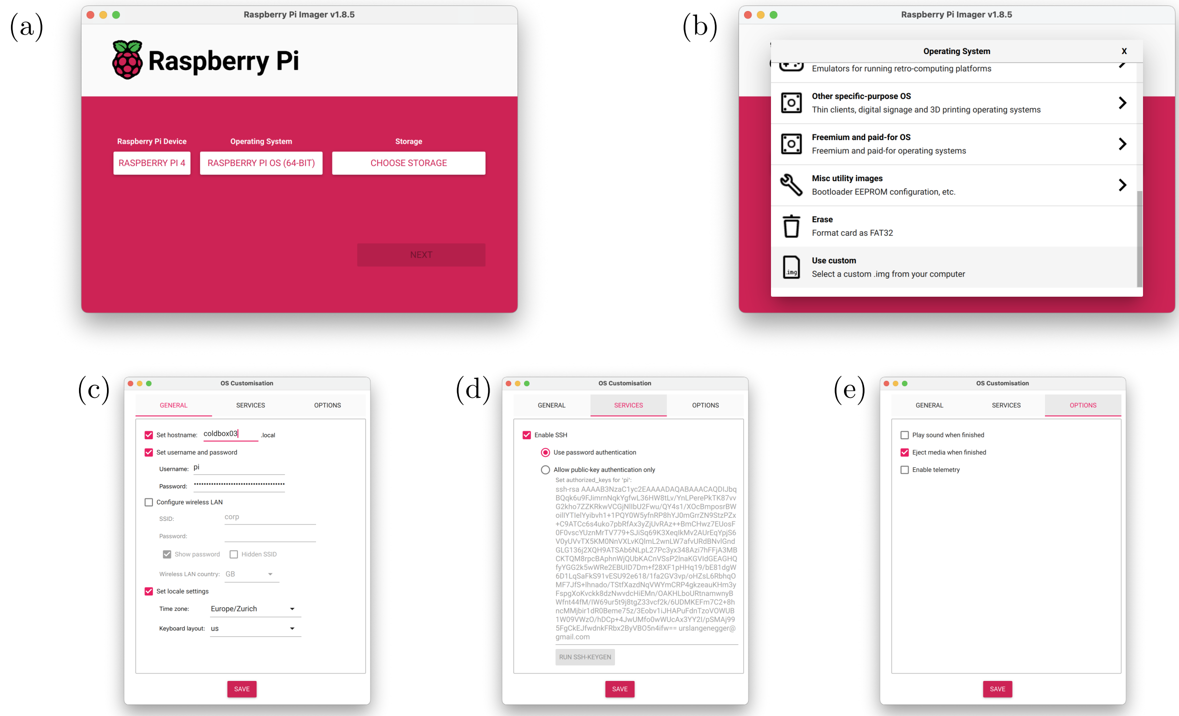

Figure 4: Screen shot of the Raspberry Pi Imager and customization examples. (a) Select the model and OS, (b) choose 'EDIT SETTINGS' to reach the lower panels, (c) enter the hostname and set a password for the default user 'pi' (use this!), (d) enable ssh connections, (e) if you want.

- *Installation of dependencies*
  Open a terminal (it should be accessible from one of the icons at the top of the display) or, better, login from another computer using `ssh` and do the following:

```
sudo date -s "Wed Apr 17 2024 10:00:00"

sudo apt-get update

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

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
```

Do enter the correct day, date, and time in the first line above. Else you will get “server certificate verification” failures further down and other issues will arise.

- *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 1: It is not recommended to use more than 2 cores for the compilation (“-j2”) because of potential memory shortages.

  Note 2: 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, *i.e.*, `sudoedit` in a vanilla system, 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: `sudoedit /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 `sudoedit`, 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
```

- *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 `sudoedit` create the file `/lib/systemd/system/tessie.service` with the following content (i.e. do `sudoedit /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 `sudoedit /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 `sudoedit /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
  ```

- *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
  ```

  Warning: If you do this step on a Raspberry Pi that is *not* in a coldbox with a central power button, it will likely shutdown down and not properly power up!

- *Splash screen configuration*
  (Note: This is not compulsory. There is no real need to change the splash screen.)
  Using `sudoedit`, 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 `sudoedit`, 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 ./
```

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. 5. You can connect from any PC.

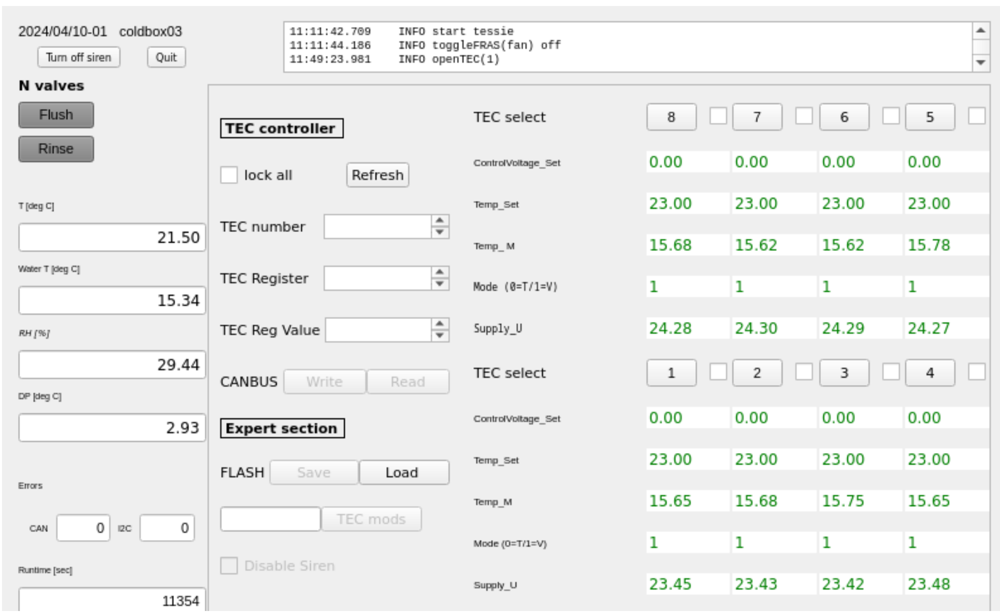

Figure 5: The GUI appearing on the Raspberry Pi touch screen.

The normal manual way to interact with `tessie` is through a webbrowser. Point your favorite browser to `http://coldbox03`, cf. Fig 6.

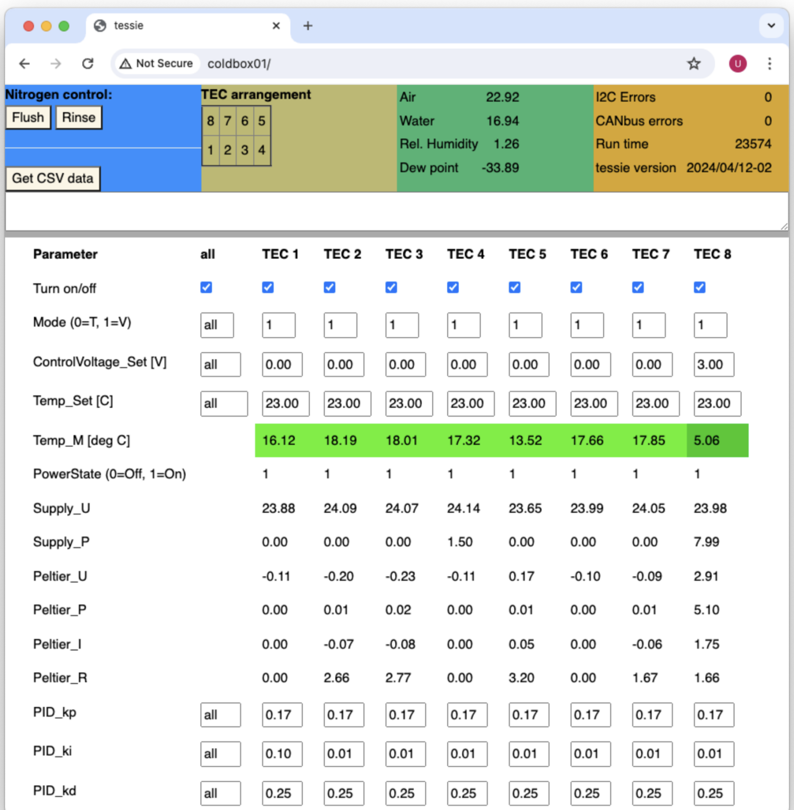

Figure 6: Web graphical interface to `tessie`. The top right brown-yellow box will turn red in case the connection to `tessieWeb` is interrupted.

4 Upgrading tessie or the TEC firmware

4.1 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 simply 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
```

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).

4.2 TEC firmware upgrades

Upgrading the **TEC** firmware is done with the OpenBLT Bootloader [10], a very convenient tool for downloading micro-controller firmware over the CAN bus (plus other options, which are not relevant for this context). No additional cabling is required. A minimal version of this software is distributed with the **tessie** repository.

Start by compiling the bootloader's library and executable with the following commands:

```
cd /home/pi/tessie/tecFirmware/Source/LibOpenBLT
mkdir _build && cd _build
cmake ..
make

cd ../../BootCommander
mkdir _build && cd _build
cmake ..
make
```

Stop `tessie` before uploading the TEC firmware! So:

```
sudo systemctl stop tessie
```

After this you can upload, via the CAN, bus the firmware, contained in file `srec/tecware.srec`, of a single `TEC` with the following command:

```
cd /home/pi/tessie/tecFirmware
./BootCommander -s=xcp -t=xcp_can -d=can0 -b=125000 -tid=0x301 -rid=0x341 srec/tecware.srec
```

Note: This is the example command to upload `TEC` 1. For `TEC i`, both the `-tid=0x30i` and `-rid=0x30i` command line arguments should be modified accordingly.

To upload all `TECs` with the same firmware file, a (very) trivial shell script is provided:

```
cd /home/pi/tessie/tecFirmware
./flashAllTECs srec/tecware.srec
```

After this procedure, you can restart `tessie`:

```
sudo systemctl start tessie
```

5 Operational aspects

5.1 First steps

To start operations on (*e.g.*) coldbox03, the following instructions might be helpful (in case you need them).

Turn on Load the module(s) into slot(s). Close the lid and ensure that it is properly locked.

- Turn on the chiller and make sure, it is not in standby mode.
- On any computer connected to the same network as your coldbox03 point a browser to `http://coldbox03`. All manual interactions described below are done on this web GUI.
- Reduce the relative humidity inside the coldbox by opening the N2 valves, initially both `Flush` and `Rinse`. Once the dew point is sufficiently low, you may turn off the `Flush` flow. Note: it is a matter of your hardware (screw) settings whether `Rinse` is sufficient to keep the relative humidity low.
- Set the voltage on the `TECs` to be operated (a good starting value is 3 V) by entering the value into the boxed field.
- Turn on the `TEC` by clicking the corresponding checkbox.
- Observe how the module temperature drops.

It is now safe to power (LV/HV) the module and run electrical tests.

Turn off Before turning off the coldbox, turn off the power (LV/HV) on the modules in the coldbox and then proceed according to the following list

- Turn off the `TECs` by clicking the checkbox.
- Turn on the N2 flow `Flush` to faster warm up the coldbox and keep the relative humidity low.
- Wait until the temperature inside the box is close to the room temperature.
- Turn off the N2 flow, both `Flush` and `Rinse`.

It is now safe to open the lid and remove the module(s).

5.2 MQTT - direct text-based communication with `tessie`

Direct communication with `tessie` proceeds via two channels, either through the touchscreen GUI or through the MQTT protocol. The web GUI is not a *direct* communication channel, since it is only an interface to MQTT messaging.

The interface with MQTT proceeds through two threads, `ctrlTessie` and `monTessie`. The former is used to communicate commands while the latter display regular monitoring information broadcast by `tessie`.
To see the traffic on a thread, you "subscribe" to it with (*e.g.*)

```
mosquitto_sub  -h coldbox03 -t "ctrlTessie"
```

To send commands over the `ctrlTessie` thread, you "publish" with (*e.g.*)

```
mosquitto_pub -h coldbox03 -t "ctrlTessie" -m "set valve0 on"
```

The `tessie` commands can be obtained with

```
mosquitto_pub -h coldbox01 -t "ctrlTessie" -m "help"
```

This will result in

```
> ===================
> hostname: coldbox01
> thread:  ctrlTessie
> ===================
>
> Note: [tec {0|x}] can be before or after {get|set|cmd XXX}, e.g.
>       cmd Power_On tec 7
>       tec 7 cmd Power_Off
>
> Note: tec numbering is from 1 .. 8. tec 0 refers to all TECs.
>
> cmd messages:
> -------------
> cmd valve0
> cmd valve1
> [tec {0|x}] cmd Power_On
> [tec {0|x}] cmd Power_Off
> [tec {0|x}] cmd ClearError
> [tec {0|x}] cmd GetSWVersion
> [tec {0|x}] cmd SaveVariables
> [tec {0|x}] cmd LoadVariables
> [tec {0|x}] cmd Reboot
>
> messages to write information:
> ------------------------------
> [tec {0|x}] set Mode {0,1}
> [tec {0|x}] set ControlVoltage_Set 1.1
> [tec {0|x}] set PID_kp 1.1
> [tec {0|x}] set PID_ki 1.1
> [tec {0|x}] set PID_kd 1.1
> [tec {0|x}] set Temp_Set 1.1
> [tec {0|x}] set PID_Max 1.1
> [tec {0|x}] set PID_Min 1.1
> set valve0 {on|off}
> set valve1 {on|off}
>
```

```
> messages to obtain information:
> -------------------------------
> get Temp
> get RH
> get DP
> get valve0
> get valve1
> get vprobe[1-8]
>
> [tec {0|x}] get Mode
> [tec {0|x}] get ControlVoltage_Set
> [tec {0|x}] get PID_kp
> [tec {0|x}] get PID_ki
> [tec {0|x}] get PID_kd
> [tec {0|x}] get Temp_Set
> [tec {0|x}] get PID_Max
> [tec {0|x}] get PID_Min
> [tec {0|x}] get Temp_W
> [tec {0|x}] get Temp_M
> [tec {0|x}] get Temp_Diff
> [tec {0|x}] get Peltier_U
> [tec {0|x}] get Peltier_I
> [tec {0|x}] get Peltier_R
> [tec {0|x}] get Peltier_P
> [tec {0|x}] get Supply_U
> [tec {0|x}] get Supply_I
> [tec {0|x}] get Supply_P
> [tec {0|x}] get PowerState
> [tec {0|x}] get Error
> [tec {0|x}] get Ref_U
> Tutorial for getting started:
> mosquitto_pub -h coldbox01 -t "ctrlTessie" -m " set valve0 on"
> mosquitto_pub -h coldbox01 -t "ctrlTessie" -m "set valve1 on"
> mosquitto_pub -h coldbox01 -t "ctrlTessie" -m "set ControlVoltage_Set 4.5"
> mosquitto_pub -h coldbox01 -t "ctrlTessie" -m "cmd Power_On"
> mosquitto_pub -h coldbox01 -t "ctrlTessie" -m "cmd Power_Off"
> mosquitto_pub -h coldbox01 -t "ctrlTessie" -m "set ControlVoltage_Set 0.0"
> mosquitto_pub -h coldbox01 -t "ctrlTessie" -m "set valve0 off"
> mosquitto_pub -h coldbox01 -t "ctrlTessie" -m "set valve1 off"
```

You should carefully read the two Notes. Issuing a command like, for instance, `cmd tec 7 Power_off` will not work, has not been promised to work, and there are no intentions to change this behavior.

5.3 Direct readout of probe card

In principle, the probe card [7] 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 -0.0004 0.07 -0.10 0.10 0.10 5.05e-05 0.10 5.06e-05 -0.10 0.001
```

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.

5.4 Traffic Lights

Three lights are used for a visual display of the operations status of the coldbox, cf. Table 2.

Table 2: "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 5.5)
Red	off	No alarm active

5.5 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.

5.6 Modes for cooling the `TECs`

There are two modes for cooling the `TECs`, *cf.* Figs. 5 and 6. The setting `Mode = 0` enables the `TEC` controller's PID control algorithm to set the `TEC` temperature to the value specified in the `Temp_Set` value, while the setting `Mode = 1` applies a fixed voltage `ControlVoltage_Set` to the `TECs` (in the range 0 < `ControlVoltage_Set` < 11 V). Table 3 provides an approximate translation between the setting of `ControlVoltage_Set` and the temperature measured on the PT1000 attached to the `TEC`, *cf.* Fig. 3. In addition, the temperature measured with an NTC on the chip of a digital module is provided. The accuracy of this NTC is of the order a few degrees, as estimated from the difference between the temperatures measured in chips 14 and 15 as shown in Fig. 7.

Table 3: Approximate translation between `ControlVoltage_Set` and PT1000 temperature `Temperature_M`, with the temperature measured on a powered HDI.

`ControlVoltage_Set [V]`	`Temperature_M` °C	**HDI temperature °C**
1	20	35–38
2	15	30–35
3	7	20–25
4	0	15–20
5	-4	12–17
FIXME	add	more

5.7 Temperatures on the `TEC` and module

FIXME

5.8 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 4 provides a summary of the safe operation parameters. An alarm is raised if `tessie` registers a violation, *cf.* Section 5.5.

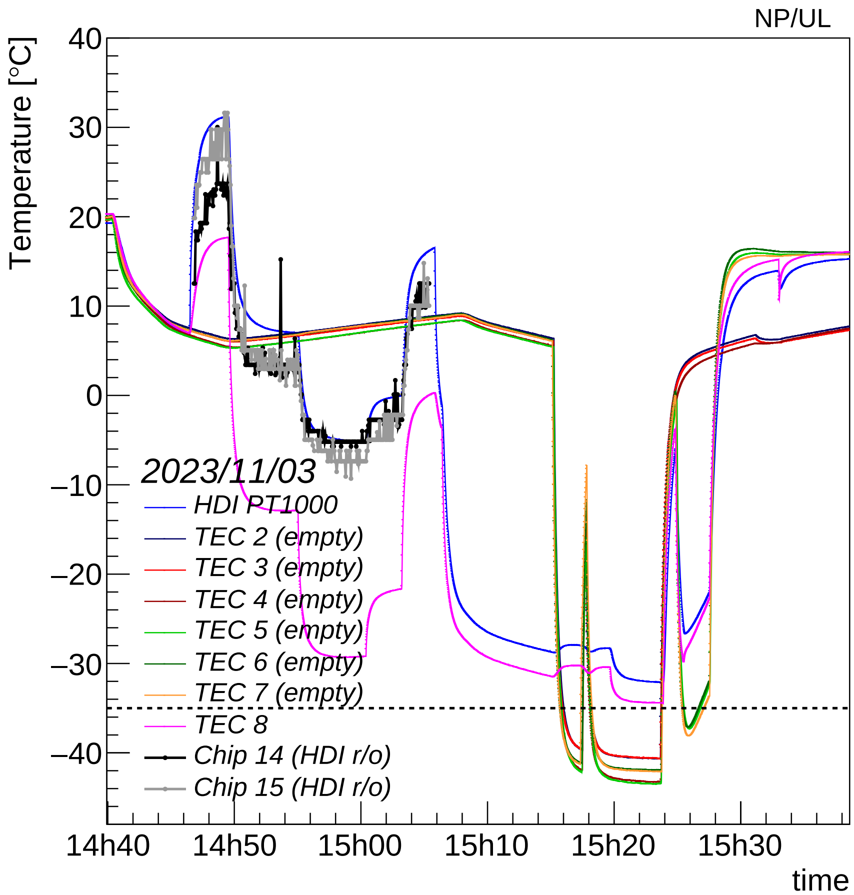

Figure 7: Temperature measurements with a digital module.

Table 4: 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

6 Frequently asked questions

This sections aims to provide help for errors and issues that may be encountered with `tessie`.

6.1 Operations

Many of the issues discussed here are indicated by error, warning, and/or alarm messages being broadcast to the MQTT channels `ctrlTessie` and `monTessie`, the red light of the traffic light turning on, sound messages being played, and other highlighting background colors in the `tessie` GUI on the coldbox touchscreen (cf. Fig. 8).

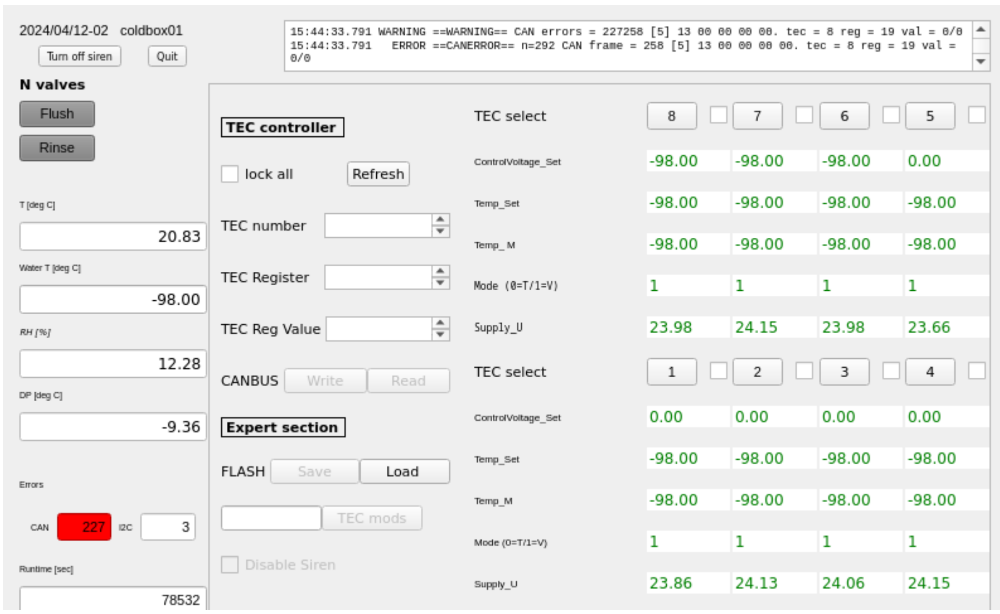

Figure 8: CAN bus errors highlighted in the GUI appearing on the coldbox touchscreen. Note (1) the red background in the CAN bus error counter field and (2) the error/warning messages in the text field (which are also broadcast to the MQTT channels).

- *Box air temperature ... too close to dew point* The reason for this condition is that the air temperature, given its relative humidity, is too close to the dew point. You likely forgot to enable the N2 flow.
  Solution: Turn of N2 flow (either flush or rinse or both).

- *Alarm module temperature exceeds (for a single module)* The reason for this condition is likely that you powered a module without turning on the corresponding `TEC`.
  Solution: Turn on the corresponding `TEC` cooling or power-off the module.

- *Alarm module temperature exceeds (for many modules)* The reason for this condition is likely that you turned on (some or all) `TEC` without turning on the chiller. It takes a few minutes to get into this state.
  Solution: Turn on the chiller.

- *The CANbus shows many errors.* The reason for this problem is likely some electrical condition affecting the CANbus due to operator manipulations.
  Solution: Restart `tessie` either by power-cycling the coldbox with the central power button or by logging in from a remote computer and doing
  `ssh coldbox`
  `sudo systemctl restart tessie`

6.2 Installation

- *The touchscreen is white* This is an indication that the flatband cable connector is not properly inserted into its socket. The fact that initially the touchscreen showed a colorful pattern is no contra-indication, that simply indicates that it has power.
  Solution: Unplug the flatband cable and insert once again.

- *The touchscreen gradually turns white and stays white.* This can happen during the first boot after inserting a fresh SD card and normally changes after repeated reboots (done automatically by the system). If it persists, then most likely you used an image that is not the recommended version.
  Solution: Burn a new iso image using the following
  https://downloads.raspberrypi.com/raspios_arm64/images/raspios_arm64-2023-12-06/2023-12-05-raspios-bookworm-arm64.img.xz
  This issue was posted to stackexchange
  https://raspberrypi.stackexchange.com/questions/147651/white-screen-after-booting-raspian-but-the-os-is-running-how-can-i-fix-that
  but is unresolved.

- *Server certificate verification* error messages when trying to run `apt-get`. This can happen if you have not set the correct time and date.
  Solution: Set the date and time using the following command (replacing the example values with the correct value)
  `sudo date -s "Wed Apr 17 2024 10:00:00"`

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

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

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

[10] FEASER, "OpenBLT Bootloader". https://www.feaser.com/openblt/doku.php?id=homepage.