

tessie User Manual

Urs Langenegger

ursslangenegger@gmail.com

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

June 19, 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	3
2	Hardware aspects	5
2.1	Raspberry Pi	5
2.2	TEC controller	5
2.3	Module probe card	5
2.4	Environmental monitoring with the Sensirion SHT85	5
2.5	CAN controlled FRAS/4 Relays	5
2.6	Lid-closure sensor	6
2.7	Cross section of module cooling infrastructure	6
3	Installation	9
4	Upgrading tessie or the TEC firmware	17
4.1	tessie upgrades	17
4.2	TEC firmware upgrades	17
5	Operational aspects	19
5.1	GUI - Raspberry Pi touchscreen interface to tessie	19
5.2	Web interface to tessie	20
5.3	MQTT - direct text-based communication with tessie	20
5.4	Safe operations limits	23
5.5	First steps	23
5.6	Direct readout of probe card	24
5.7	Traffic Lights	25
5.8	Alarm channels	25
5.9	Modes for cooling the TECs	26
5.10	Temperatures on the TEC and module	26
6	Frequently asked questions	28
6.1	Operations	28
6.1.1	Box air temperature ... too close to dew point	28
6.1.2	An alarm goes off if I plug in a module and close the lid	28
6.1.3	Alarm module temperature exceeds SAFETY_MAXTEMPM (single module) .	28
6.1.4	Alarm module temperature exceeds SAFETY_MAXTEMPM (single module) .	28
6.1.5	Alarm module temperature exceeds SAFETY_MAXTEMPM (many modules) .	28
6.1.6	The CANbus shows many errors	29
6.1.7	Persistent issue with a single TEC	29
6.1.8	Where is the printout from tessie?	29
6.1.9	The N2 flow valves are not working, but no CANbus errors appear . .	29
6.2	Installation	29
References		31

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 repository [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 accommodates the electrical connections and readout of the TEPX modules and furthermore hosts an SHT85 air and humidity sensor [5].

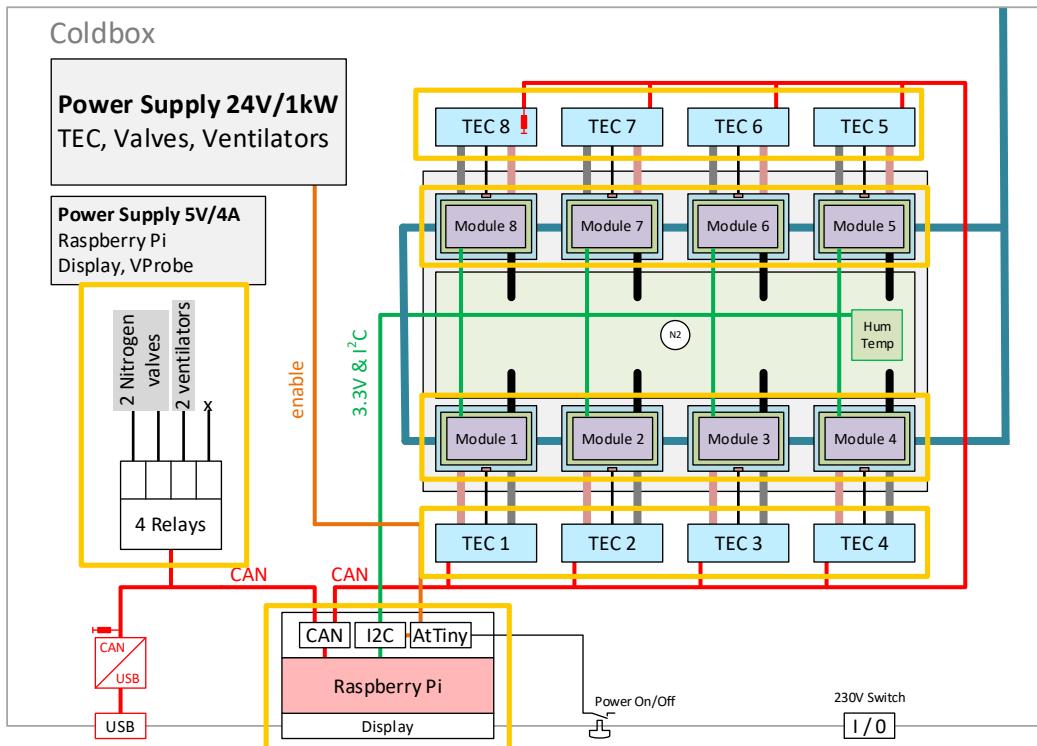


Figure 1: Sketch of the PSI coldbox showing the orientation and numbering scheme of the eight module positions, the CAN bus and I²C bus connection scheme, and the FRAS/4 Relays controlling the N₂ flow. The three white boxes above the **Raspberry Pi** are implemented using a custom “hat”. The thick blue 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

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

direct operation includes for example

- Opening and closing the lid to load or unload modules for testing
- Turning on/off the N2 flow controls

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	GPIOYELLOW	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)

A custom-made "hat" provides the interface between the `Raspberry Pi` and the CAN/I2C buses. However, because that hat has no direct consequences for `tessie`, it is not discussed further here.

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. 3 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

The SHT85 sensor [5] is read out in the high-repeatability mode. The I2C device address of the sensor is 0x44. The data transferred consist of 2×2 bytes together with 2 bytes for a CRC checksum.

2.5 CAN controlled FRAS/4 Relays

FIXME

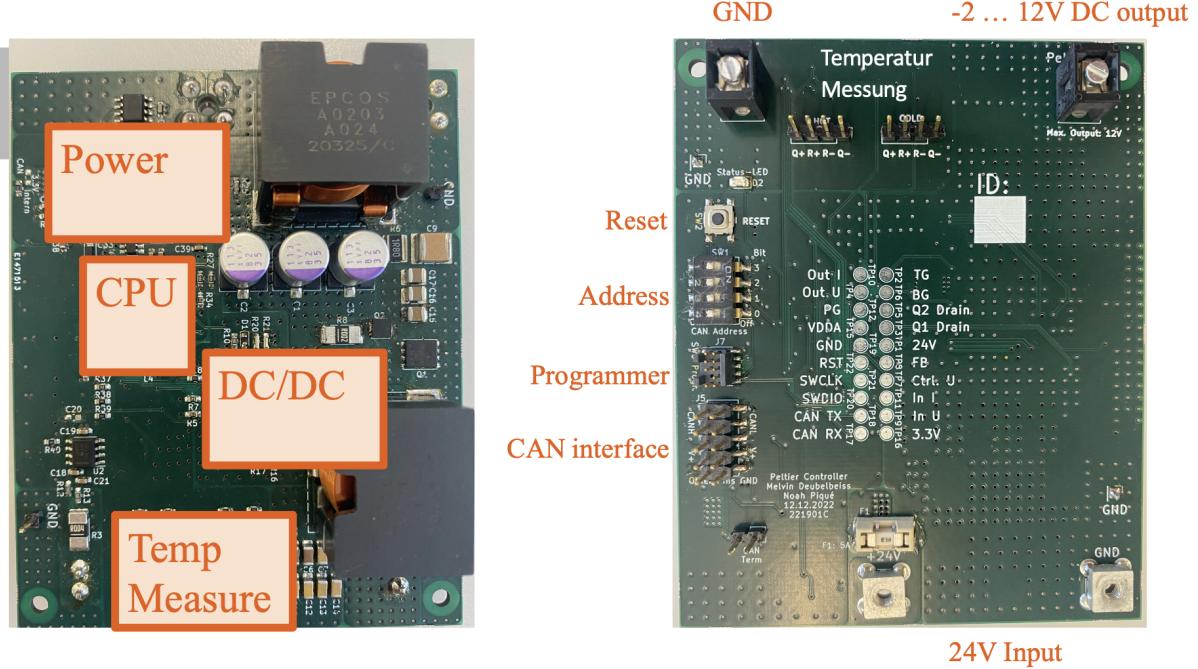


Figure 2: Photo of the (left) frontside and (right) backside of the TEC controller [6]. The fuse F1 is visible on the backside.

2.6 Lid-closure sensor

FIXME

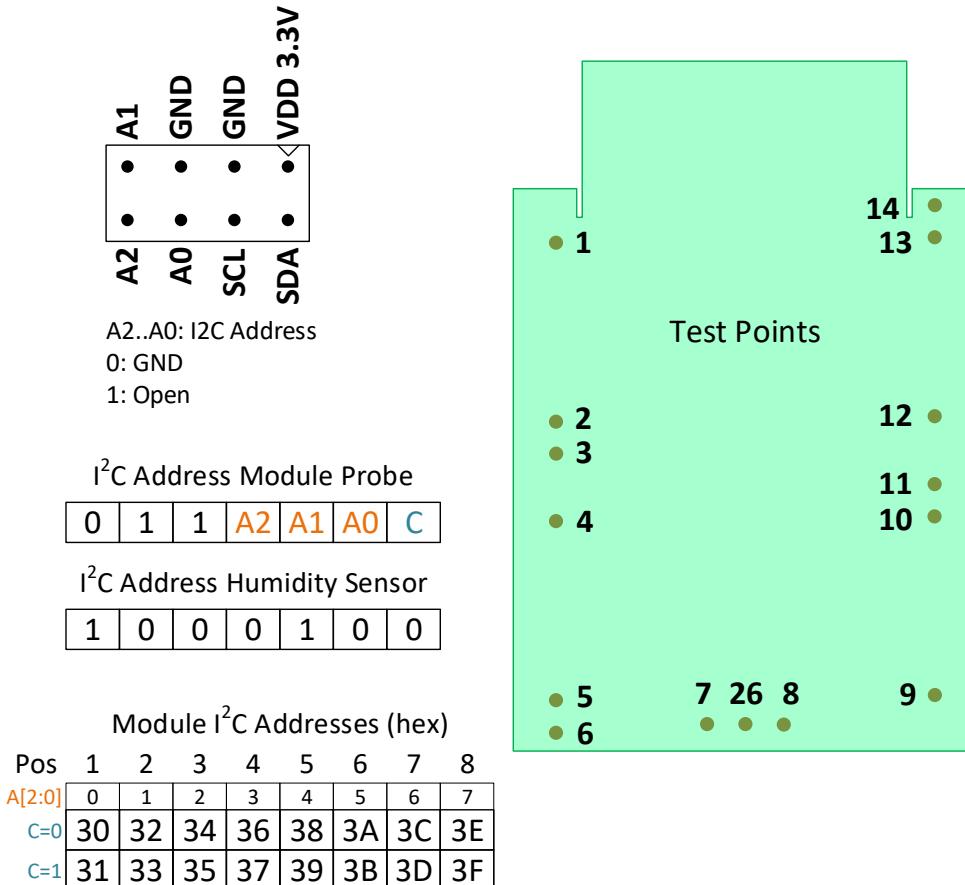
4200 when open, something else (possibly negative) when closed. To be seen.

2.7 Cross section of module cooling infrastructure

The cross section of the module cooling infrastructure is shown in Fig. 4. 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. 10 in section 5.10.

Module Probe V2



Module ADC Readout Data 18 Bytes																		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
C=0	TP 12	TP 8	TP 7	TP 11	TP 10	TP 9	TP 5	TP 6	201	1								
C=1	TP 1	TP 8	TP 26	TP 14	TP 13	TP 2	TP 3	TP 4	201	1								

$$\text{ADC value conversion: } U[V] = \frac{V_{DD}}{65536} \cdot \text{ADC}$$

Beat Meier PSI 12.6.2023

Figure 3: 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.

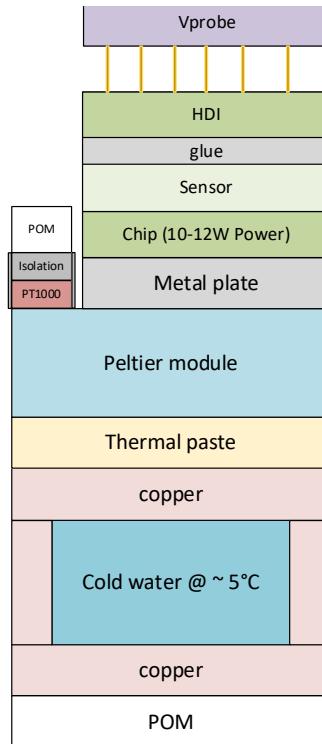


Figure 4: Cross section of the cooling infrastructure for a single module inside the coldbox. From the top: "Vprobe" is the probe card (subsection 2.3), the structures from the "HDI" down to and including the "Chip" represent the module, positioned on a metal plate in direct contact to the TEC (Peltier 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 [here](#). Make sure that you do *not* use XZ utils versions 5.6.0 or 5.6.1 since they contain malicious code (cf. [link](#)).

It is recommended to apply a few changes to the default setup as illustrated in Fig. 5, 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.

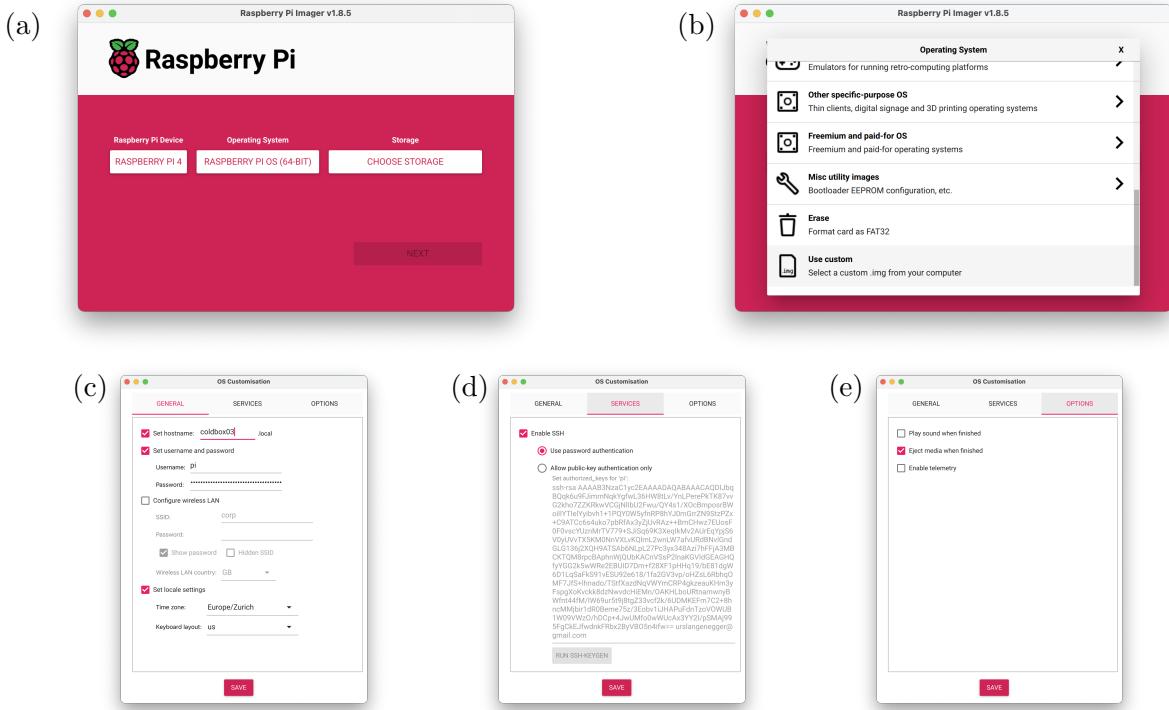


Figure 5: Screen shot of the Raspberry Pi Imager and customization examples. (a) Select the model and OS, (b) choose ‘EDIT SETTINGS’ to reach the lower panels, (c) enter the hostname and set a password for the default user ‘pi’ (use this!), (d) enable ssh connections, (e) if you want.

- *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/src
qmake -o Makefile tessie.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 tessie.pro`.

- *Mosquitto setup*

Using the `nano` editor in `sudo` mode, *i.e.*, `sudoedit` in a vanilla system, edit the file `/etc/mosquitto/mosquitto.conf` and add the following two lines to the end of the file:

```
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
User=pi
Group=staff
Environment="XAUTHORITY=/home/pi/.Xauthority"
Environment="DISPLAY=:0"
WorkingDirectory=/home/pi/tessie/src
ExecStartPre=/home/pi/tessie/resetCAN.sh
ExecStart=/home/pi/tessie/src/tessie
StandardOutput=inherit
StandardError=inherit

[Install]
WantedBy=graphical.target

```

Note that `tessie` is run under the user `pi`. 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
```

- *Configure sound*

This optional section provides information on how to enable sound output to an attached USB loudspeaker. First determine which sound card your loudspeaker is attached to with

```
cat /proc/asound/cards
```

This will result in a list similar to

```

0 [vc4hdmi0      ]: vc4-hdmi - vc4-hdmi-0
                      vc4-hdmi-0
1 [vc4hdmi1      ]: vc4-hdmi - vc4-hdmi-1
                      vc4-hdmi-1
2 [Headphones     ]: bcm2835_headpho - bcm2835 Headphones
                      bcm2835 Headphones
3 [UACDemoV10    ]: USB-Audio - UACDemoV1.0
                      Jiel Technology UACDemoV1.0 at usb-0000:01:00.0-1.3, full speed

```

Here the loudspeaker is attached to card 3. With this information sudoedit the file /etc/asound.conf with the contents

```
defaults.pcm.card 3
defaults.ctl.card 3
```

- *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 button 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. 6. 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 7.

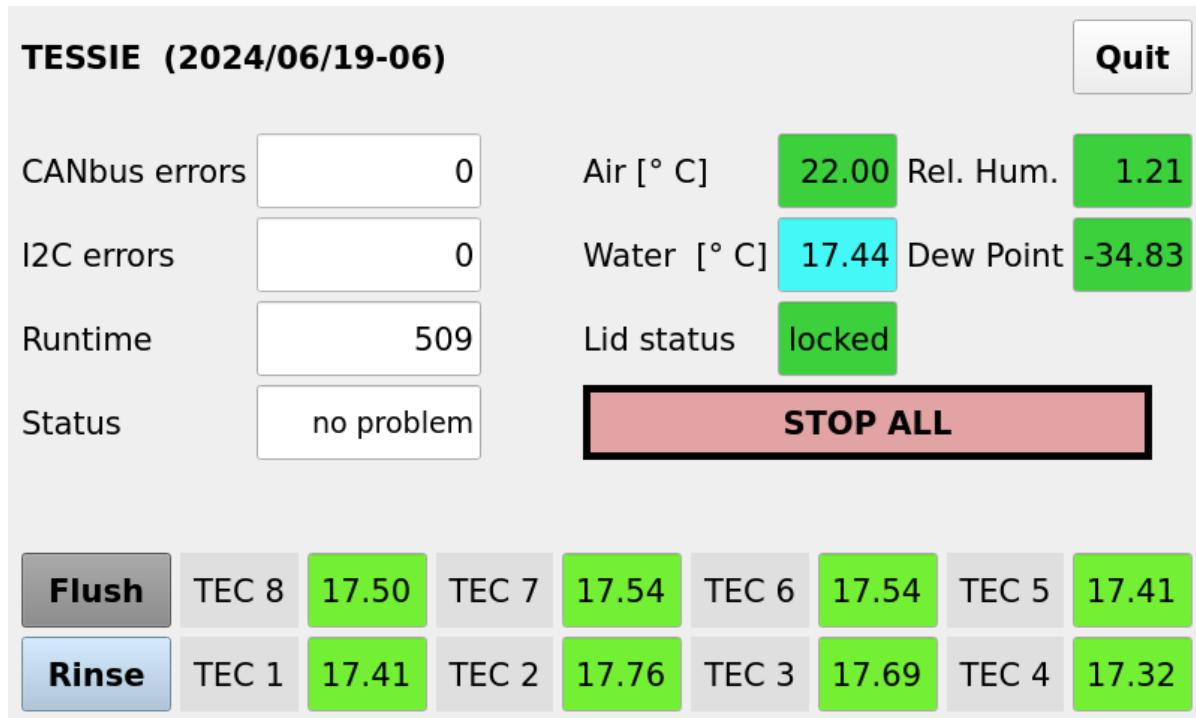


Figure 6: The GUI appearing on the Raspberry Pi touch screen. The interaction possibilities are limited to (1) controlling the N2 flow with "Rinse" and "Flush", (2) stop operations with "STOP ALL" which will turn off all TECs and turn on maximal N2 flow by opening both the rinse and flush valves, and (3) "Quit"-ing the program. The bluish background color shade on the "Rinse" button indicates that this valve is open and N2 is flowing (and therefore the relative humidity is low). For full control of `tessie` you should use the web interface shown in Fig. 7.

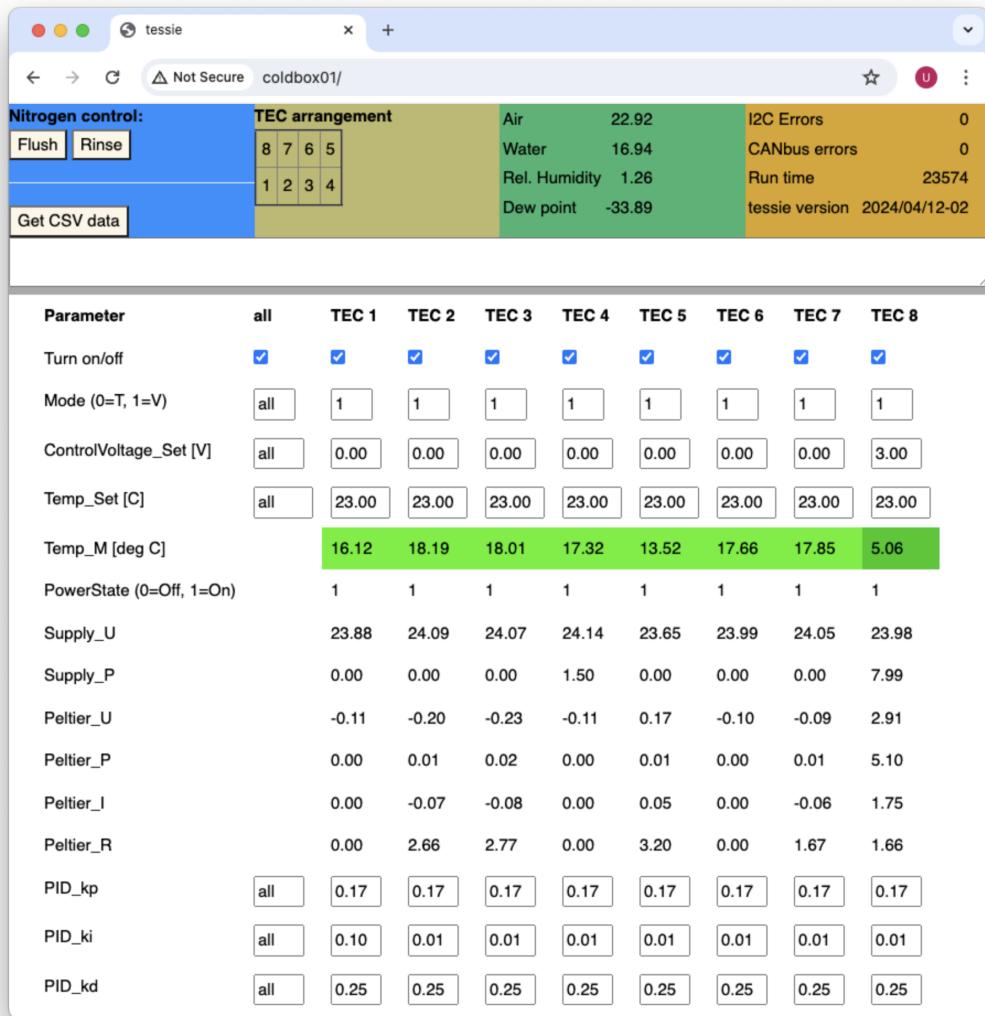


Figure 7: Web graphical interface to **tessie**. The top right brown-yellow box will turn red in case the connection to **tessieWeb** is interrupted.

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

There are three methods to control the behavior of `tessie`:

- Graphical User Interface on the Raspberry Pi touchscreen
- Web Interface accessible with a web browser
- Text based interface based on MQTT

This section starts with a description of these three methods and afterwards discusses various aspects relevant to operations.

5.1 GUI - Raspberry Pi touchscreen interface to `tessie`

The GUI Raspberry Pi touchscreen interface is shown in Fig. 8 (where an error condition, rapidly accumulating CANbus errors, is indicated through the red background in the corresponding field).

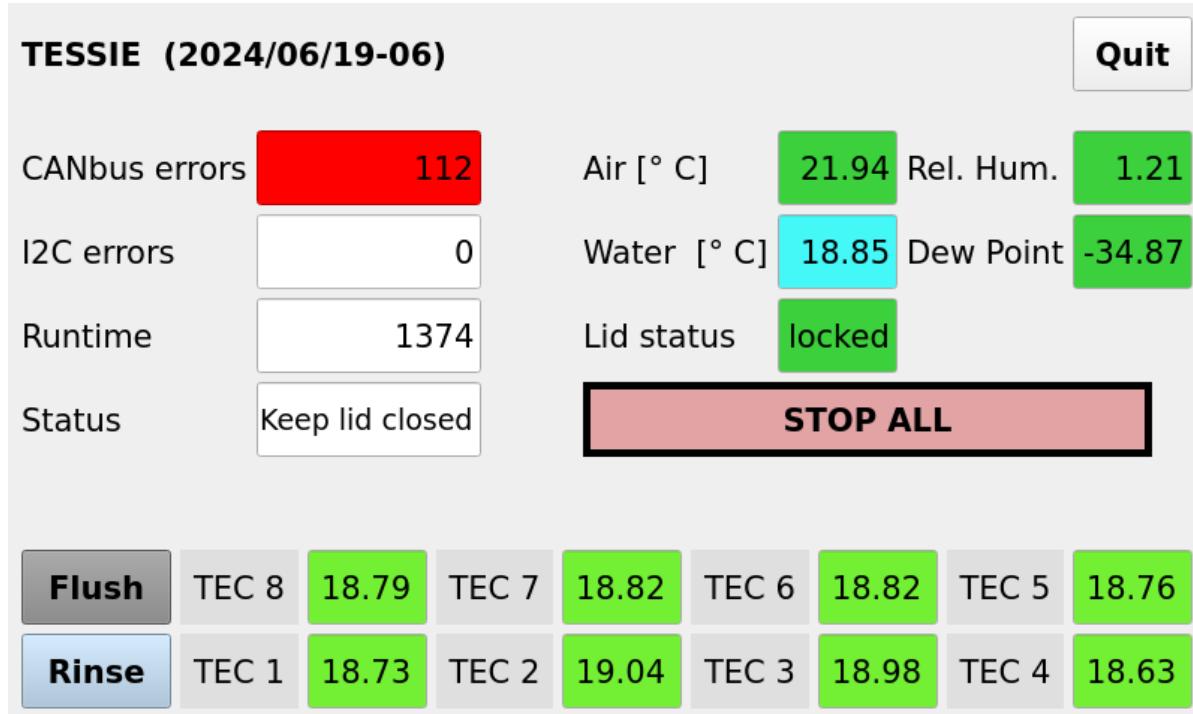


Figure 8: The GUI Raspberry Pi touchscreen interface to `tessie`

This GUI starts up on the Raspberry Pi touch screen after booting and provides a minimalist control surface to `tessie`. It allows for

- turning off and on the "Flush" and "Rinse" valves controlling the N2 flow,
- stopping all "dangerous" activities with the "STOP ALL" button that will turn off all TECs, turn on both "Flush" and "Rinse" valves, and disable the interlock. It will make

sure that the relative humidity in the box is minimal and that the power dissipated in the box is also minimal. This is the button you should hit if things go berserk.

- quitting the `tessie` program.

All other controlling interactions with `tessie` should be initiated from the web interface, described in section 5.2.

The GUI displays the status of the system in three different blocks with multiple text displays with changing background colors.

System status The "CANbus errors" field accumulates the total number of CANbus errors over `tessie`'s running time. It will turn (shortly) red in case of a new CANbus error. If there is a persisting CANbus problem, it will stay red. The "I2C errors" field provides the same functionality for the I2C bus. "Runtime" show the `tessie` running time in seconds. The "Status" box tries to indicate the status of the box in normal terms.

Environmental status The "Air" field indicates the air temperature as measured with the SHT85, located at one end of the PCB inside the coldbox. Its background changes with the air temperature between blue, green, yellow, and red. The "Water" field provides the same information for the cooling water temperature inside the coldbox, with the same color coding as for the "Air" temperature field. The "Rel. Hum." field shows the air relative humidity in percent. It will turn green for values < 5%. The "Dew Point" shows the calculated dew point and is green if both the air temperature is at least 2°C warmer than the dew point. The "Lid status" indicates whether the lid is locked or not and changes color accordingly.

TEC status The labels with the TEC numbers change the color depending whether the TEC is powered or not, in a similar fashion as the valve buttons color scheme. The fields provide the temperature $_{\text{rec}}$ [°C] of the PT-1000 on the TEC (*cf.* Fig. 4) with a changing background color scheme.

5.2 Web interface to `tessie`

FIXME

5.3 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.*)

The screenshot shows a web-based interface for the **tessie** system. At the top, there's a navigation bar with icons for back, forward, refresh, and a note that it's 'Not Secure'. Below the bar, the URL is `coldbox01/`. The interface is divided into several sections:

- Nitrogen control:** Buttons for **Flush** and **Rinse**, and a **Get CSV data** button.
- TEC arrangement:** A grid showing TEC numbers 1 through 8 arranged in two rows of four.
- Sensor Data:** Air temperature (22.92), Water temperature (16.94), Rel. Humidity (1.26), Dew point (-33.89), I2C Errors (0), CANbus errors (0), Run time (23574), and tessie version (2024/04/12-02).
- Parameter Control:** A table where rows represent parameters and columns represent TEC units (all, TEC 1, TEC 2, TEC 3, TEC 4, TEC 5, TEC 6, TEC 7, TEC 8). The table includes fields for Turn on/off (checkboxes), Mode (dropdowns with values 1 or all), ControlVoltage_Set [V] (text inputs like 0.00), Temp_Set [C] (text inputs like 23.00), and various temperature and power readings (e.g., 16.12, 23.88, 0.00).

Figure 9: Web graphical interface to **tessie**.

```
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.

5.4 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 5.8.

5.5 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.

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

- 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).

5.6 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.

5.7 Traffic Lights

Three lights are used for a visual display of the operations status of the coldbox, cf. Table 3.

Table 3: “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 5.8)
Red	off	No alarm active

5.8 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.

5.9 Modes for cooling the TECs

There are two modes for cooling the TECs, *cf.* Figs. 6 and 7. 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 $0 < \text{ControlVoltage_Set} < 11$ V). Table 4 provides an approximate translation between the setting of `ControlVoltage_Set` and the temperature measured on the PT1000 attached to the TEC, *cf.* Fig. 4. 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. 10.

Table 4: Approximate translation between `ControlVoltage_Set` and PT1000 temperature `Temperature_M`, with the temperature measured on a powered HDI.

<code>ControlVoltage_Set</code> [V]	<code>Temperature_M</code> [°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

5.10 Temperatures on the TEC and module

FIXME

NTC readout with Ph2-ACF software (version v4-15) using the command `CMSITminiDAQ -f CMSIT_RD53B.xml -c ntc`.

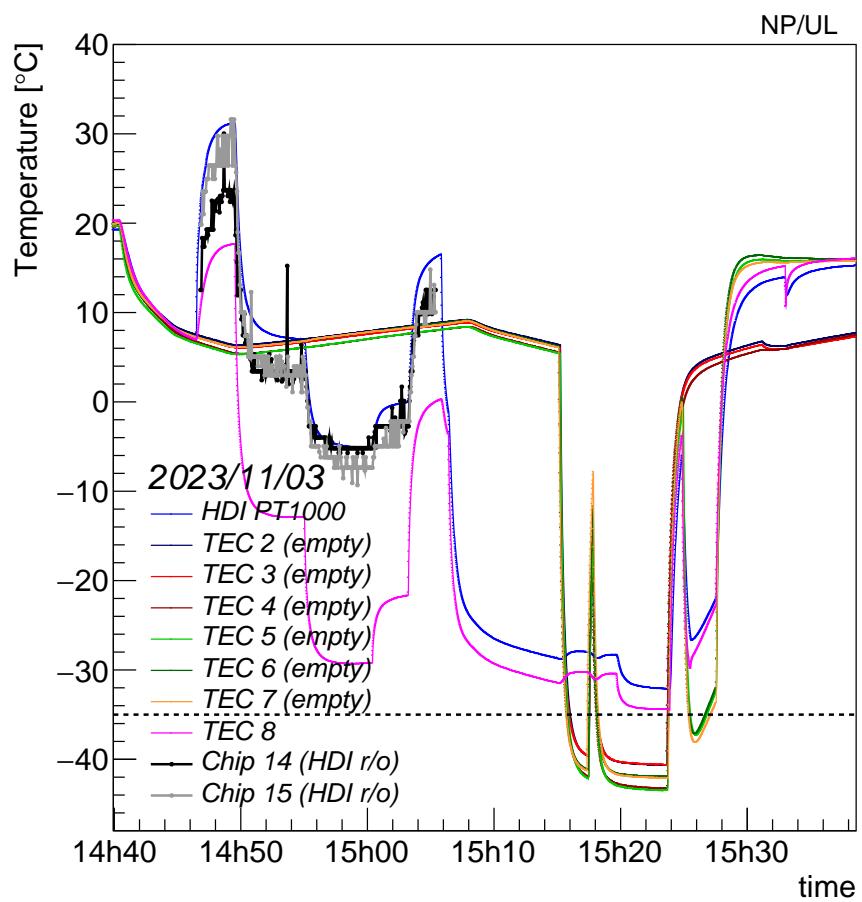


Figure 10: Temperature measurements with a digital module.

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).

6.1.1 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).

6.1.2 An alarm goes off if I plug in a module and close the lid

A possible reason for this condition could be that you turned on the chiller, opened the lid to connect a module, closed the lid and then did *not* turn on the N2 flow. The chiller water will cool the Peltier element to whatever temperature you set the water temperature to and this temperature may well be below the dew point (especially in Summer, when the ambient air humidity is high). This will trigger the alarm `module ... too close to dew point ...`

Solution: Turn of N2 flow (either flush or rinse or both).

6.1.3 Alarm module temperature exceeds SAFETY_MAXTEMPM (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.

6.1.4 Alarm module temperature exceeds SAFETY_MAXTEMPM (single module)

The reason for this condition is likely that you turned on one TEC, and then off again, without turning on the chiller (or there is no flow in the water). This can warm the entire side (1–4 or 5–8) of the coldbox and, because of variations between the different TEC temperature sensing, a single module can go above the limit before the others do so. It can take a while to get into this state.

Solution: Turn on the chiller.

6.1.5 Alarm module temperature exceeds SAFETY_MAXTEMPM (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.

6.1.6 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
```

6.1.7 Persistent issue with a single TEC

The “issue” could be CAN bus errors (showing up as, e.g. temperature readings of -98°C) or other examples. This could be due to a blown fuse on the TEC.

Solution: Replace the fuse on the TEC, cf. Fig. 2.

6.1.8 Where is the printout from `tessie`?

For a better understanding of possible issues with `tessie` sometimes you need access to the printout, especially if you have looked into the code. *Solution:* You can see the printout with `journalctl`.

```
journalctl -u tessie -b
journalctl --since "24 hour ago" -u tessie
```

The option `-b` yields all messages since the last boot, the second example restricts this to the specified time period.

6.1.9 The N2 flow valves are not working, but no CANbus errors appear

One possibility is that a connector on the FRAS has a shaky contact. This can be indicated by no LED on the FRAS being lighted.

Solution: Open the box (take off the cover for that purpose) and ensure a proper contact of the connector.

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 `img`. This issue was posted to stackexchange ([here](#)), but it remains 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>.