rpi-backlight

Release 2.6.0

Dec 29, 2022

Contents

1 Contents **3**
- 1.1 Introduction . . . 3
- 1.2 Installation . . . 3
  - 1.2.1 Requirements . . . 4
  - 1.2.2 Installation . . . 4
- 1.3 Usage . . . 4
  - 1.3.1 Python API . . . 4
  - 1.3.2 Command line interface . . . 5
  - 1.3.3 Graphical user interface . . . 6
- 1.4 API reference . . . 8
- 1.5 Changes . . . 10
  - 1.5.1 2.6.0 . . . 10
  - 1.5.2 2.5.0 . . . 10
  - 1.5.3 2.4.1 . . . 10
  - 1.5.4 2.4.0 . . . 10
  - 1.5.5 2.3.0 . . . 10
  - 1.5.6 2.2.0 . . . 10
  - 1.5.7 2.1.0 . . . 10
  - 1.5.8 2.0.1 . . . 11
  - 1.5.9 2.0.0 . . . 11
  - 1.5.10 1.8.1 . . . 11
  - 1.5.11 1.8.0 . . . 11
  - 1.5.12 1.7.1 . . . 11
  - 1.5.13 1.7.0 . . . 11
  - 1.5.14 1.6.0 . . . 11
  - 1.5.15 1.5.0 . . . 12
  - 1.5.16 1.4.0 . . . 12
  - 1.5.17 1.3.1 . . . 12
  - 1.5.18 1.3.0 . . . 12
  - 1.5.19 1.2.1 . . . 12
  - 1.5.20 1.2.0 . . . 12
  - 1.5.21 1.1.0 . . . 13
  - 1.5.22 1.0.0 . . . 13
- 1.6 License . . . 13

2 Indices and tables **15**

Python Module Index **17**

Index **19**

Version 2.6.0

Author Linus Groh

Contact mail@linusgroh.de

License (code) *MIT license*

License (docs) This document was placed in the public domain.

CHAPTER 1

Contents

1.1 Introduction

When I bought the official Raspberry Pi 7" touch LCD, I was quite happy about it - with one exception: *you can't change the display brightness in a simple way out of the box*.

I did some research and hacked some Python code together. Time passed by, and the whole project turned into a Python module: `rpi-backlight`.

Currently it has the following features:

- Change the display brightness **smoothly** or **abrupt**
- Set the display power on or off
- Get the current brightness
- Get the maximum brightness
- Get the display power state (on/off)
- Command line interface
- Graphical user interface

Now you are able to easily set the brightness of your display from the command line, a GUI and even Python code!

1.2 Installation

This section covers the installation of the library on the Raspberry Pi.

1.2.1 Requirements

- A **Raspberry Pi** including a correctly assembled **7" touch display v1.1 or higher** (look on the display's circuit board to see its version) running a Linux-based OS. Alternatively you can use rpi-backlight-emulator on all operating systems and without the actual hardware.
- Python 3.6+
- Optional: `pygobject` for the GUI, already installed on a recent Raspbian

1.2.2 Installation

Note: This library will **not** work with Windows IoT, you'll need a Linux distribution running on your Raspberry Pi. This was tested with Raspbian 9 (Stretch) and 10 (Buster).

rpi-backlight is available on PyPI, so you can install it using `pip3`:

```
$ pip3 install rpi_backlight
```

Note: Create this udev rule to update permissions, otherwise you'll have to run Python code, the GUI and CLI as root when *changing* the power or brightness:

```
$ echo 'SUBSYSTEM=="backlight",RUN+="/bin/chmod 666 /sys/class/backlight/%k/
↪brightness /sys/class/backlight/%k/bl_power"' | sudo tee -a /etc/udev/rules.d/
↪backlight-permissions.rules
```

rpi-backlight is now installed. See *Usage* to get started!

1.3 Usage

1.3.1 Python API

Make sure you've *installed* the library correctly.

Open a Python shell and import the `Backlight` class:

```
>>> from rpi_backlight import Backlight
```

Create an instance:

```
>>> backlight = Backlight()
```

Now you can get and set the display power and brightness:

```
>>> backlight.brightness
100
>>> backlight.brightness = 50
>>> backlight.brightness
50
>>>
>>> with backlight.fade(duration=1):
...     backlight.brightness = 0
```

(continues on next page)

(continued from previous page)

```
...
>>> backlight.fade_duration = 0.5
>>> # subsequent `backlight.brightness = x` will fade 500ms
>>>
>>> backlight.power
True
>>> backlight.power = False
>>> backlight.power
False
>>>
```

To use with ASUS Tinker Board:

```
>>> from rpi_backlight import Backlight, BoardType
>>>
>>> backlight = Backlight(board_type=BoardType.TINKER_BOARD)
>>> # continue like above
```

See the *API reference* for more details.

1.3.2 Command line interface

Open a terminal and run `rpi-backlight`.

```
$ rpi-backlight -b 100
$ rpi-backlight --set-brightness 20 --duration 1.5
$ rpi-backlight --get-brightness
20
$ rpi-backlight --get-power
on
$ rpi-backlight -p off
$ rpi-backlight --get-power
off
$ rpi-backlight --set-power off :emulator:
$
```

To use with ASUS Tinker Board:

```
$ rpi-backlight --board-type tinker-board ...
```

You can set the backlight sysfs path using a positional argument, set it to *:emulator:* to use with *rpi-backlight-emulator*.

Available options:

```
usage: rpi-backlight [-h] [--get-brightness] [-b VALUE] [--get-power]
                     [-p VALUE] [-d DURATION] [-B {raspberry-pi,tinker-board}]
                     [-V]
                     [SYSFS_PATH]

Get/set power and brightness of the official Raspberry Pi 7" touch display.

positional arguments:
SYSFS_PATH            Optional path to the backlight sysfs, set to
                        :emulator: to use with rpi-backlight-emulator
```

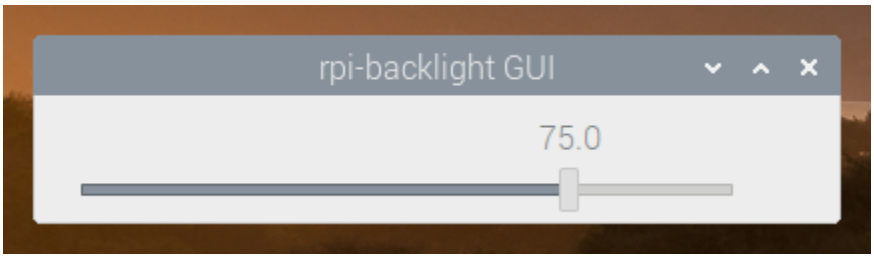
(continues on next page)

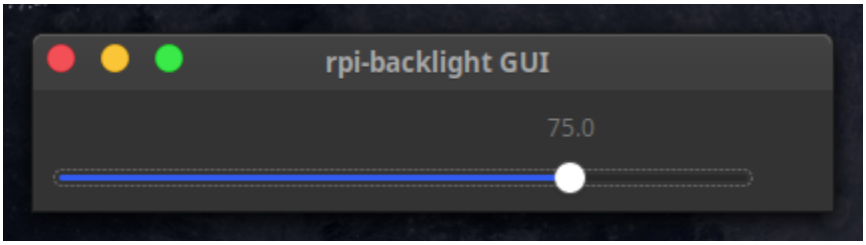
(continued from previous page)

```
optional arguments:
-h, --help            show this help message and exit
--get-brightness      get the display brightness (0-100)
-b VALUE, --set-brightness VALUE
                        set the display brightness (0-100)
--get-power           get the display power (on/off)
-p VALUE, --set-power VALUE
                        set the display power (on/off/toggle)
-d DURATION, --duration DURATION
                        fading duration in seconds
-B {raspberry-pi,tinker-board}, --board-type {raspberry-pi,tinker-board}
                        board type
-V, --version         show program's version number and exit
```

1.3.3 Graphical user interface

Open a terminal and run `rpi-backlight-gui`.





Adding a shortcut to the LXDE panel

First, create a `.desktop` file for rpi-backlight (e.g. `/home/pi/.local/share/applications/rpi-backlight.desktop`) with the following content:

```
[Desktop Entry]
Version=1.0
Type=Application
Terminal=false
Name=rpi-backlight GUI
Exec=/home/pi/.local/bin/rpi-backlight-gui
Icon=/usr/share/icons/HighContrast/256x256/status/display-brightness.png
Categories=Utility;
```

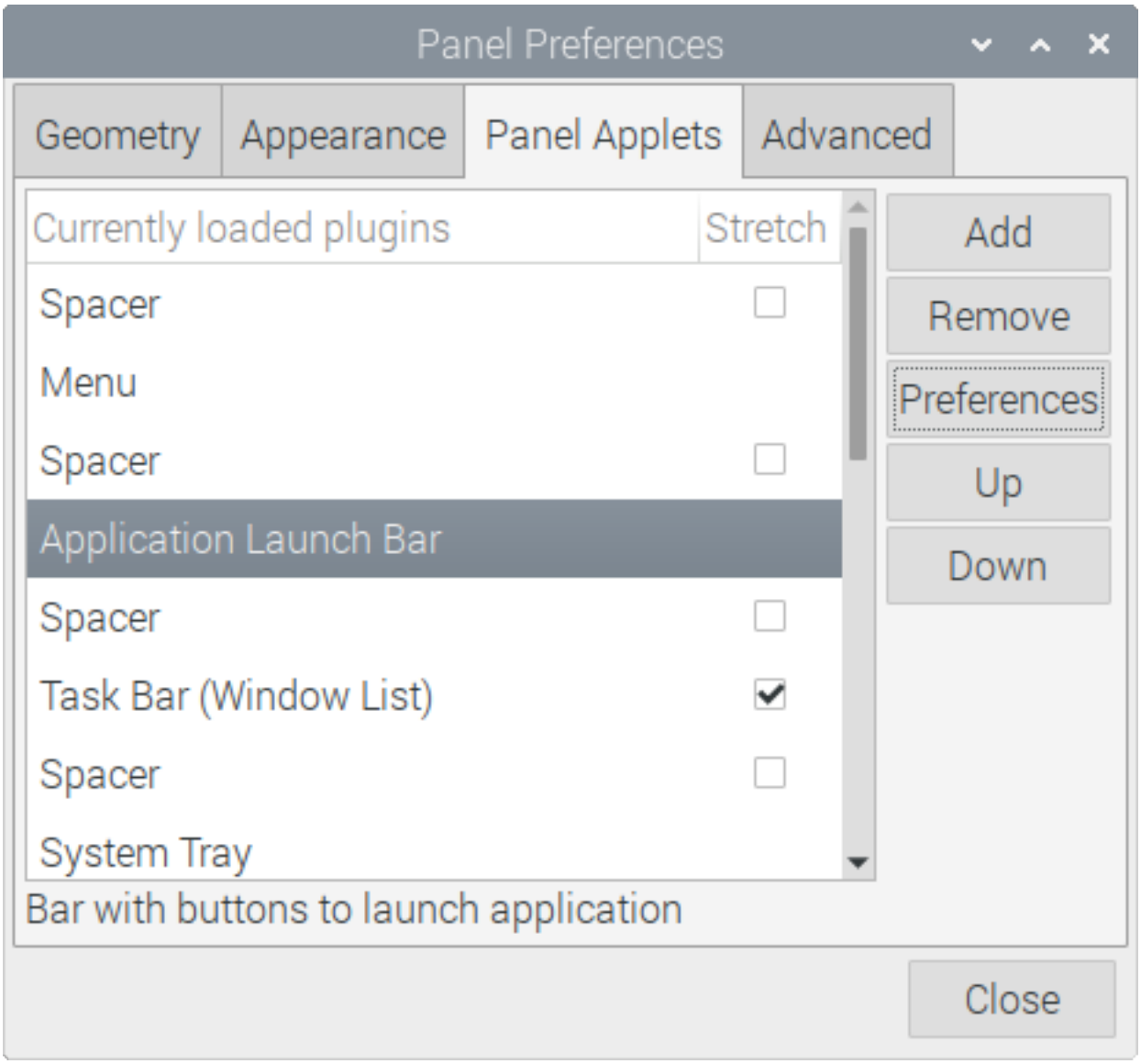
The absolute path to `rpi-backlight-gui` *might differ if you did not follow the installation instructions exactly, e.g. installed as root.*

Make it executable:

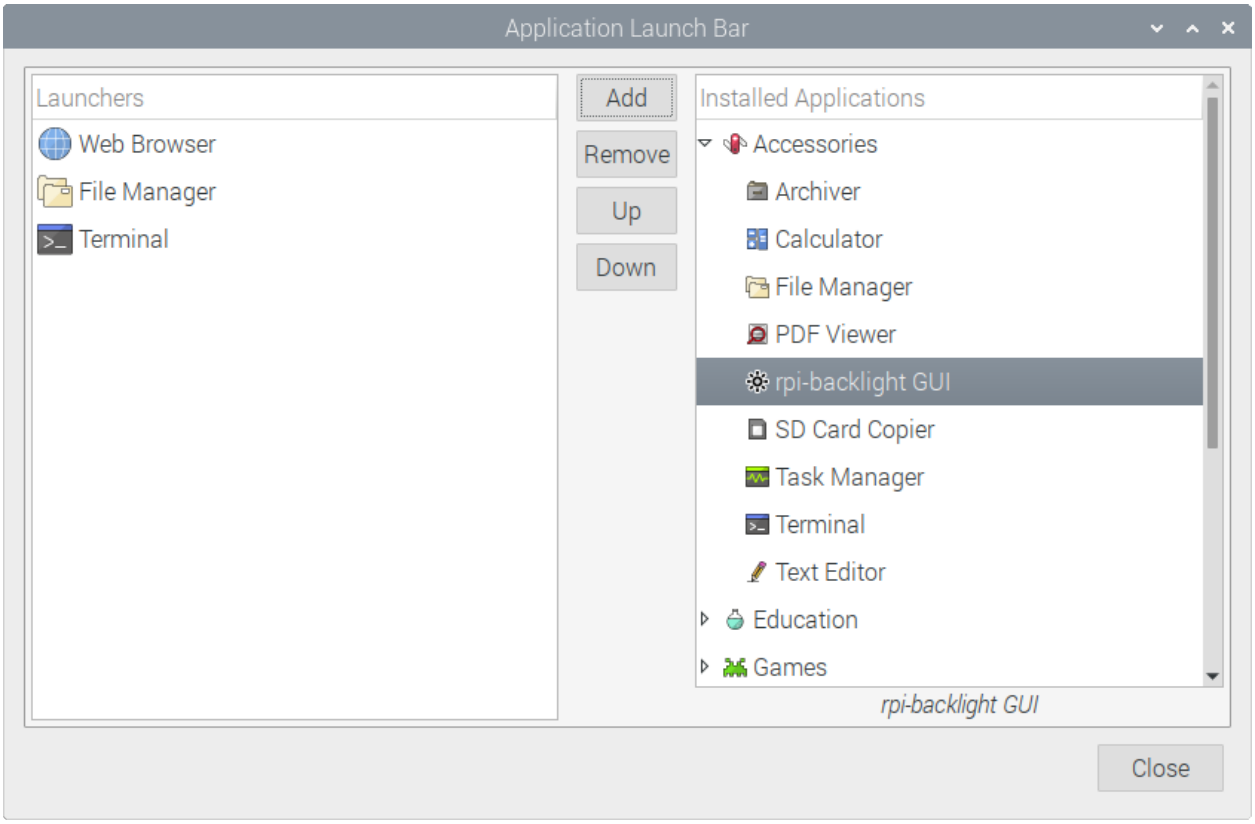
```
$ chmod +x /home/pi/.local/share/applications/rpi-backlight.desktop
```

You should now be able to start the rpi-backlight GUI from the menu: `(Raspberry Pi Logo)` → `Accessoires` → `rpi-backlight GUI`.

Next, right-click on the panel and choose `Add / Remove panel items`. Select `Application Launch Bar` and click `Preferences`:



Select `rpi-backlight GUI` on the right and click `Add`:



You're done!

1.4 API reference

class rpi_backlight.**Backlight**(*backlight_sysfs_path: Union[str, PathLike[str], None] = None, board_type: rpi_backlight.BoardType = <BoardType.RASPBERRY_PI: 1>*)

Main class to access and control the display backlight power and brightness.

Set `backlight_sysfs_path` to `":emulator:"` to use with rpi-backlight-emulator.

brightness

The display brightness in range 0-100.

```
>>> backlight = Backlight()
>>> backlight.brightness  # Display is at 50% brightness
50
>>> backlight.brightness = 100  # Set to full brightness
```

Getter Return the display brightness.

Setter Set the display brightness.

Type float

fade(*duration: float*) → Generator[T_co, T_contra, V_co]

Context manager for temporarily changing the fade duration.

```
>>> backlight = Backlight()
>>> with backlight.fade(duration=0.5):
...     backlight.brightness = 1  # Fade to 100% brightness for 0.5s
...
>>> with backlight.fade(duration=0):
...     backlight.brightness = 0  # Set to 0% brightness without fading, use␣
↪if you have set `backlight.fade_duration` > 0
```

fade_duration

The brightness fade duration in seconds, defaults to 0. Also see *fade()*.

```
>>> backlight = Backlight()
>>> backlight.fade_duration  # Fading is disabled by default
0
>>> backlight.fade_duration = 0.5  # Set to 500ms
```

Getter Return the fade duration.

Setter Set the fade duration.

Type float

power

Turn the display on and off.

```
>>> backlight = Backlight()
>>> backlight.power  # Display is on
True
>>> backlight.power = False  # Turn display off
```

Getter Return whether the display is powered on or off.

Setter Set the display power on or off.

Type bool

class rpi_backlight.**BoardType**

Enum to specify a board type in the *Backlight* constructor.

MICROSOFT_SURFACE_RT = 4

Microsoft Surface RT

RASPBERRY_PI = 1

Raspberry Pi

TINKER_BOARD = 2

Tinker Board

TINKER_BOARD_2 = 3

Tinker Board 2

rpi_backlight.cli.**main**()

Start the command line interface.

rpi_backlight.gui.**main**()

Start the graphical user interface.

rpi_backlight.utils.**detect_board_type**() → Optional[BoardType]

Try to detect the board type based on the model string in `/proc/device-tree/model`.

class rpi_backlight.utils.**FakeBacklightSysfs**

Context manager to create a temporary "fake sysfs" containing all relevant files. Used for tests and emulation.

```
>>> with FakeBacklightSysfs() as backlight_sysfs:
...     backlight = Backlight(backlight_sysfs_path=backlight_sysfs.path)
...     # use `backlight` as usual
```

1.5 Changes

1.5.1 2.6.0

- Add Python 3.11 as supported version, drop 3.6
- Add support for Microsoft Surface RT (#54, @apandada1)

1.5.2 2.5.0

- Add Python 3.10 as supported version
- Support alternate backlight sysfs path (#50, @j-coopz)

1.5.3 2.4.1

- Fix board type detection

1.5.4 2.4.0

- Drop support for Python 3.5, which reached end-of-life in late 2020. Supported versions as of this release are 3.6 - 3.9
- Implement automatic board type detection (#32, @p1r473), passing a `board_type` to the constructor is no longer necessary in most cases, even when using an ASUS Tinker Board
- Fix setting brightness to max value and brightness fading loop condition (#35, @Martin-HiPi)

1.5.5 2.3.0

- Add support for ASUS Tinker Board 2 (#29, @p1r473)

1.5.6 2.2.0

- Add toggle functionality to CLI (#21, @p1r473)
- Replace Travis CI with GitHub actions (#20, @linusg)
- Improve tests

1.5.7 2.1.0

- Add support for ASUS Tinker Board (#19, @p1r473)

1.5.8 2.0.1

- Add mypy type checking
- Add Python 3.8 to Travis CI config
- Fix documentation readthedocs build
- Fix typo in docs
- Improve README.md
- Mark project as stable on PyPI

1.5.9 2.0.0

- New, more pythonic API
- Update CLI and GUI
- Support emulator
- Add tests

1.5.10 1.8.1

- Fix float division issue with Python 2

1.5.11 1.8.0

- Fix permission error inconsistency across Python versions
- Update link to PyPI

1.5.12 1.7.1

- Fixed typo in `CHANGES.rst`
- Fixed rendering of parameters and return types in the documentation

1.5.13 1.7.0

- Fixed bug in `get_power`, which would eventually always return False
- Added parameters and return types in docstrings

1.5.14 1.6.0

- Added `duration` parameter to `set_brightness`
- `smooth` now defaults to `False`
- Huge improvements on CLI
- Fixed renamed function in examples

- Minor code and readme improvements

1.5.15 1.5.0

- PR #3 by Scouttp: Fixed permission errors
- Added documentation
- Code improvements
- Fixed typos

1.5.16 1.4.0

- Check for `pygobject` being installed
- Code cleanup
- README improvements
  - Added external links
  - Added badges
  - Fixed typos
- Moved to Travis CI and Landscape.io for builds and code health testing
- Prepared docs hosting at readthedocs.org

1.5.17 1.3.1

- Fixed type conversion

1.5.18 1.3.0

- Added experimental GUI (start with `rpi-backlight-gui`)

1.5.19 1.2.1

- Fixed CLI and typo

1.5.20 1.2.0

- Added command line interface (`rpi-backlight` and `rpi-backlight-gui`)
- Code improvements - thanks to deets

1.5.21 1.1.0

- Fixed `set_power(on)` function
- Added function to get the current power state of the LCD
- Added docstrings
- Code cleanup and improvements

1.5.22 1.0.0

Initial release. Added necessary files and basic features:

- Change the display brightness smoothly or abrupt
- Set the display power on or off
- Get the current brightness
- Get the maximum brightness

1.6 License

The rpi-backlight source code is distributed under the terms of the MIT license, see below:


```
MIT License

Copyright (c) 2016-2022 Linus Groh

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

r

rpi_backlight, 8
rpi_backlight.cli, 9
rpi_backlight.gui, 9
rpi_backlight.utils, 9

Index

B

`Backlight` (*class in rpi_backlight*), 8
`BoardType` (*class in rpi_backlight*), 9
`brightness` (*rpi_backlight.Backlight attribute*), 8

D

`detect_board_type()` (*in module rpi_backlight.utils*), 9

F

`fade()` (*rpi_backlight.Backlight method*), 8
`fade_duration` (*rpi_backlight.Backlight attribute*), 9
`FakeBacklightSysfs` (*class in rpi_backlight.utils*), 9

M

`main()` (*in module rpi_backlight.cli*), 9
`main()` (*in module rpi_backlight.gui*), 9
`MICROSOFT_SURFACE_RT` (*rpi_backlight.BoardType attribute*), 9

P

`power` (*rpi_backlight.Backlight attribute*), 9

R

`RASPBERRY_PI` (*rpi_backlight.BoardType attribute*), 9
`rpi_backlight` (*module*), 8
`rpi_backlight.cli` (*module*), 9
`rpi_backlight.gui` (*module*), 9
`rpi_backlight.utils` (*module*), 9

T

`TINKER_BOARD` (*rpi_backlight.BoardType attribute*), 9
`TINKER_BOARD_2` (*rpi_backlight.BoardType attribute*), 9