

# Smart Devices

# Agenda

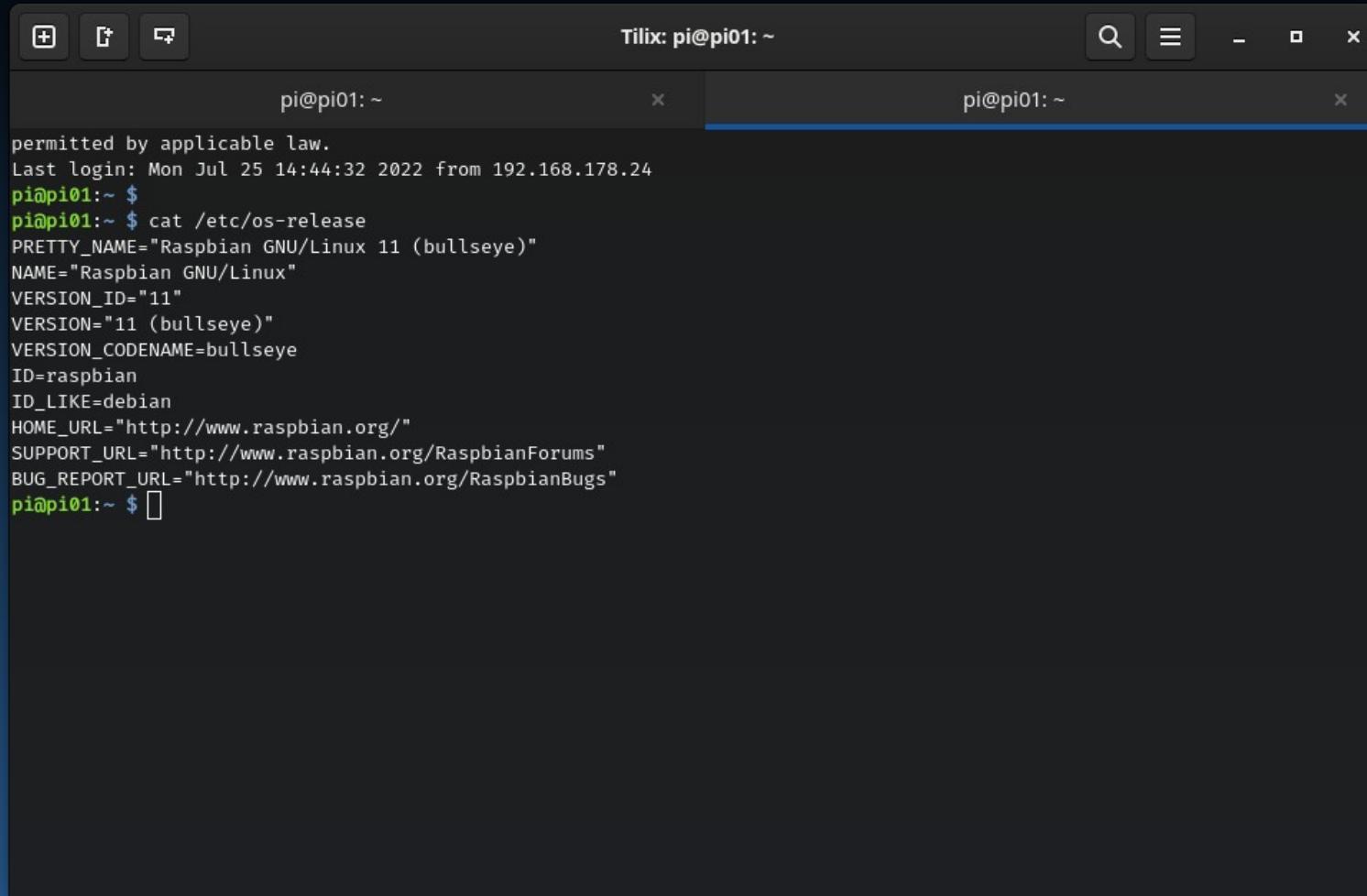
- Requirements
- Preparation
- Cabling
- First Boot
- Programming
  - LEDs
  - Buttons
  - Web Server
  - (Camera)

# Requirements

# Requirements

- Software
  - Raspberry Imager
  - Raspberry Pi OS
  - Fritzing
  - Putty or other SSH client

# Requirements



Tilix: pi@pi01: ~

pi@pi01: ~

permitted by applicable law.

Last login: Mon Jul 25 14:44:32 2022 from 192.168.178.24

pi@pi01:~ \$

pi@pi01:~ \$ cat /etc/os-release

PRETTY\_NAME="Raspbian GNU/Linux 11 (bullseye)"

NAME="Raspbian GNU/Linux"

VERSION\_ID="11"

VERSION="11 (bullseye)"

VERSION\_CODENAME=bullseye

ID=raspbian

ID\_LIKE=debian

HOME\_URL="http://www.raspbian.org/"

SUPPORT\_URL="http://www.raspbian.org/RaspbianForums"

BUG\_REPORT\_URL="http://www.raspbian.org/RaspbianBugs"

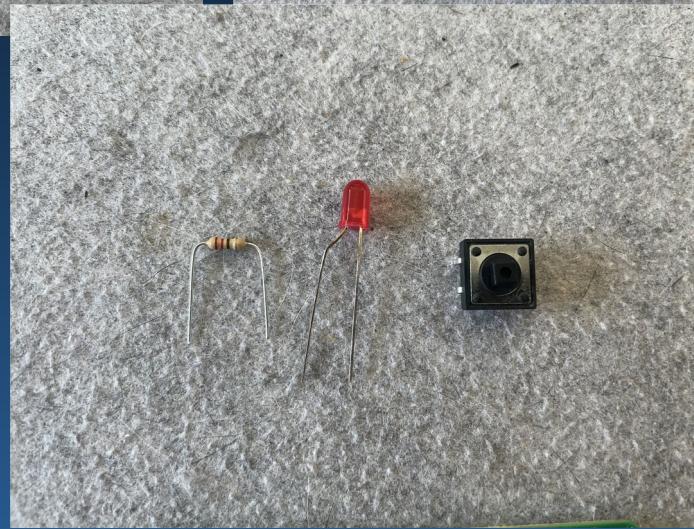
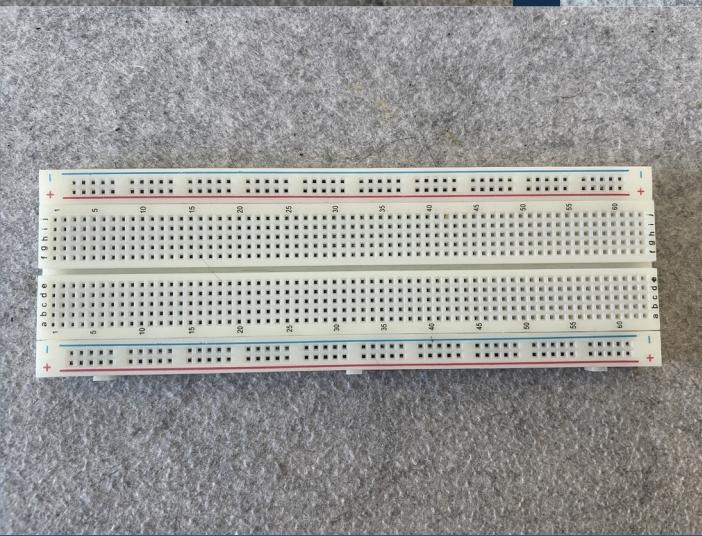
pi@pi01:~ \$

# Requirements

## Hardware

- Raspberry Pi
- Breadboard
- LED
- Button
- Resistor
- Cables

# Hardware

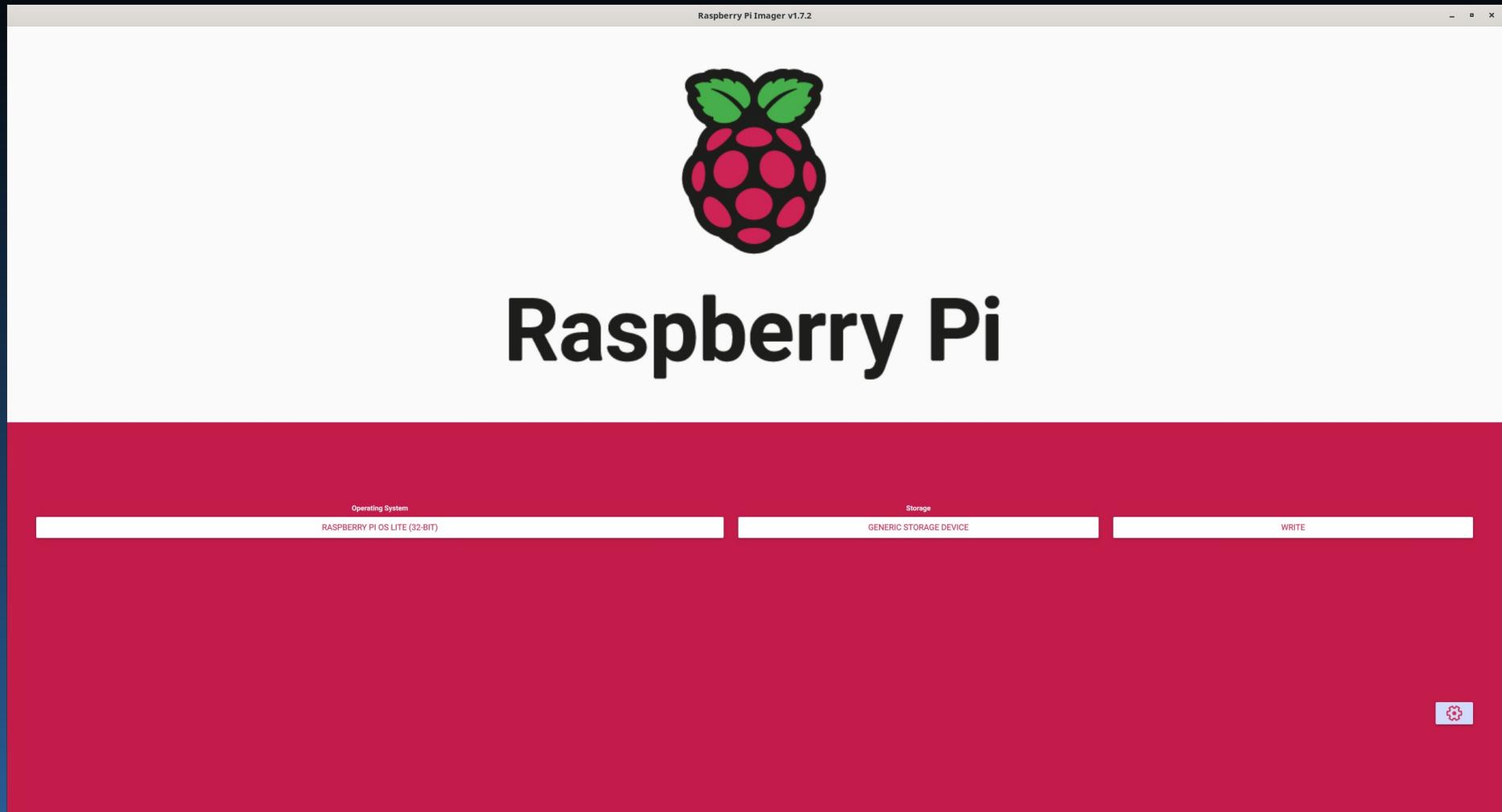


# Preparation

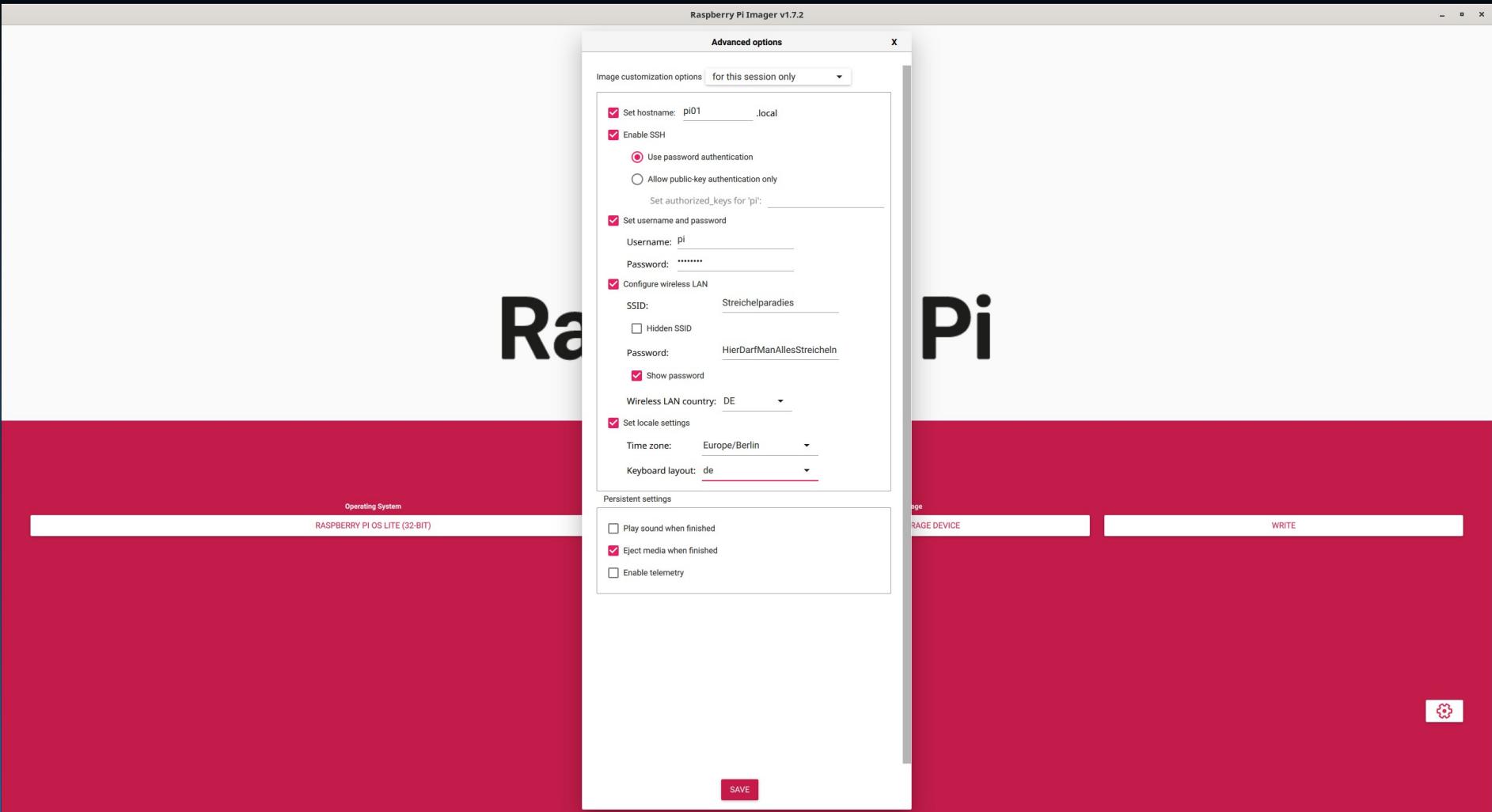
# Preparation

- Manual Install
  - Download lite image  
\$ wget https://downloads.raspberrypi.org/rpios\_lite\_armhf\_latest
  - Download regular image  
\$ wget https://downloads.raspberrypi.org/rpios\_armhf\_latest
  - Write SD card  
\$ unzip -p <IMAGE> | sudo dd of=/dev/mmcblk0 oflag=sync status=progress bs=4M
- Guided Install
  - Download Raspberry Imager

# Preparation



# Preparation



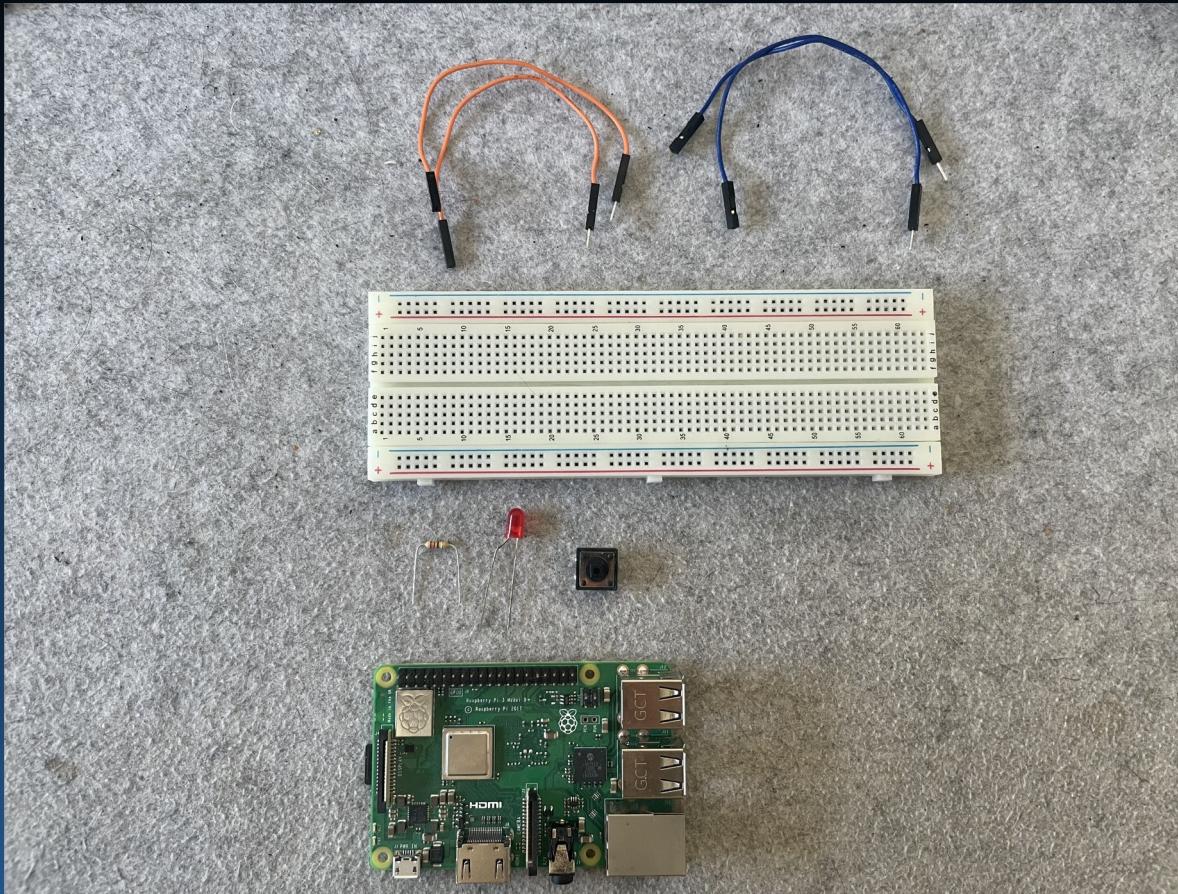
# First Boot

# First Boot

- Connect to the Raspberry
  - Use SSH or Putty to connect  
\$ ssh -l pi <IPADDRESS>
- First commands
  - Show GPIO Pins  
\$ pinout
  - Change directory  
\$ cd
  - List files  
\$ ls
  - Editor  
\$ nano <FILENAME>

# Programming

# Programming - Preparation



# Programming - Preparation

- Create a working directory

```
$ mkdir project
```

- Navigate there

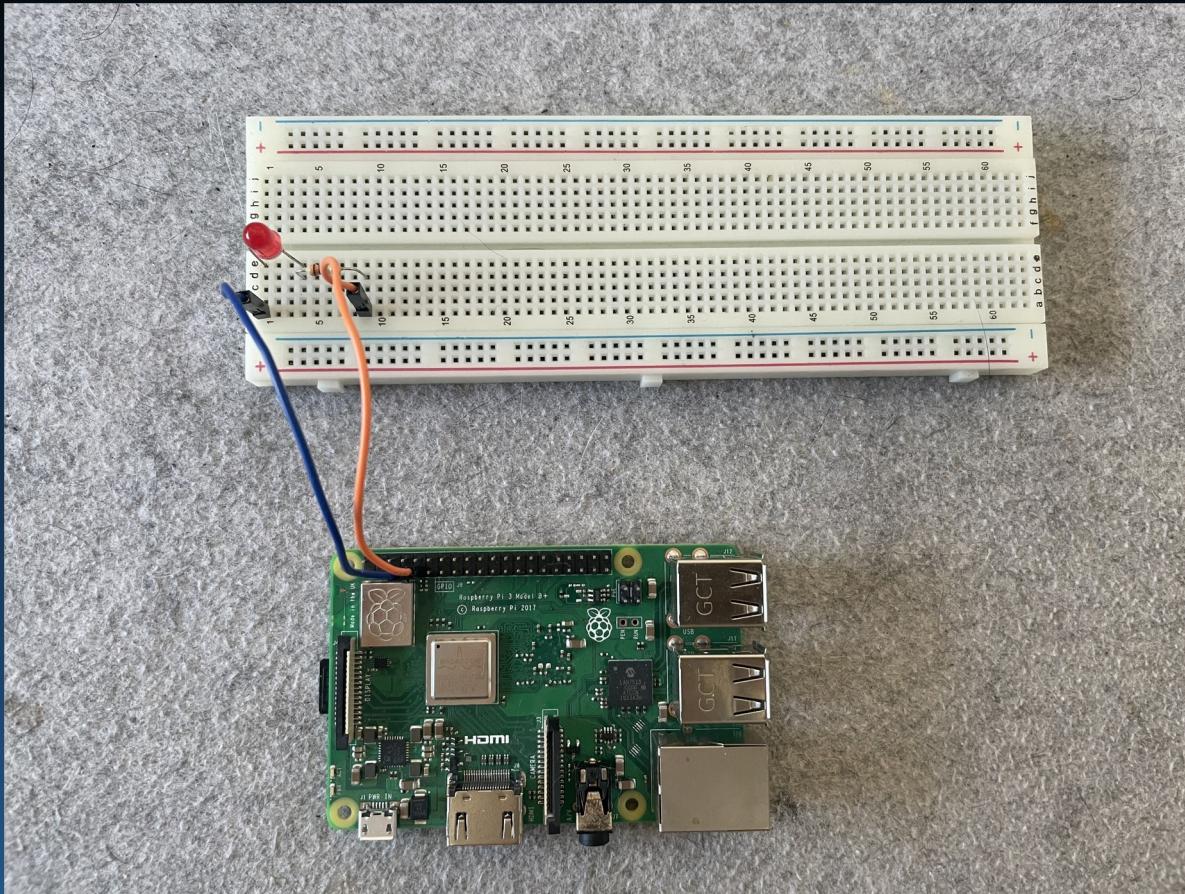
```
$ cd project/
```

- Start a new file

```
$ nano led_1.py
```

- Continue with the exercises

# Programming - LED



# Programming – LED 1

```
from gpiozero import LED  
  
from time import sleep  
  
  
led = LED(4)  
  
  
  
  
led.on()  
sleep(1)  
led.off()
```

# Programming – LED 2

```
from gpiozero import LED  
  
from time import sleep  
  
  
led = LED(4)  
  
  
  
  
  
  
while True:  
    led.on()  
    sleep(1)  
    led.off()  
    sleep(1)
```

# Programming – LED 3

```
from gpiozero import LED  
  
from signal import pause  
  
  
led = LED(4)  
  
  
led.blink()  
  
  
pause()
```

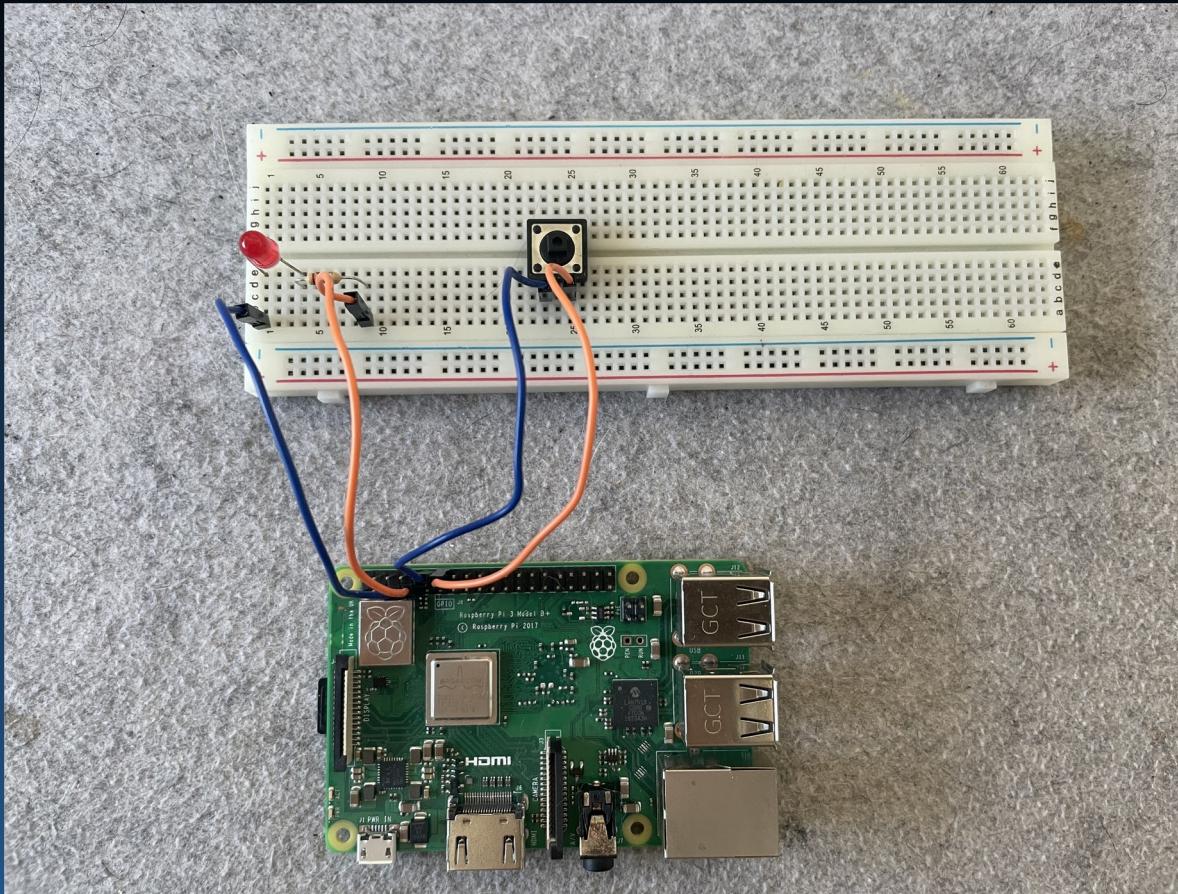
# Programming – LED 4

```
from gpiozero import PWMLED  
  
from time import sleep  
  
  
led = PWMLED(4)  
  
  
while True:  
    led.value = 0  # off  
    sleep(1)  
    led.value = 0.5 # half brightness  
    sleep(1)  
    led.value = 1  # full brightness  
    sleep(1)
```

# Programming – LED 5

```
from gpiozero import PWMLED  
  
from signal import pause  
  
led = PWMLED(4)  
  
led.pulse()  
  
pause()
```

# Programming - Button



# Programming – Button 1

```
from gpiozero import Button

button = Button(18)

while True:

    if button.is_pressed:

        print("Button is pressed")

    else:

        print("Button is not pressed")
```

# Programming – Button 2

```
from gpiozero import LED, Button  
  
from signal import pause  
  
  
led = LED(4)  
  
button = Button(18)  
  
  
button.when_pressed = led.on  
  
button.when_released = led.off  
  
  
pause()
```

# Programming – Button 3

```
from gpiozero import Button
```

```
button = Button(18)
```

```
button.wait_for_press()
```

```
print("Button was pressed")
```

# Programming – Button 4

```
from gpiozero import Button  
  
from signal import pause  
  
  
def say_hello():  
    print("Hello!")  
  
  
button = Button(18)  
  
  
button.when_pressed = say_hello  
  
  
pause()
```

# Programming - Webserver



## Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.load
|       '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`.

# Programming - Webserver

```
# Install Apache httpd  
  
$ sudo apt-get update && sudo apt-get install apache2  
  
# Enable the server  
  
$ sudo systemctl enable apache2  
  
$ sudo systemctl start apache2
```

## Docs & Links

- <https://www.raspberrypi.org/>
- <https://www.raspberrypi.com/documentation/computers/os.html#gpio-and-the-40-pin-header>
- <https://gpiozero.readthedocs.io/en/stable/>
-