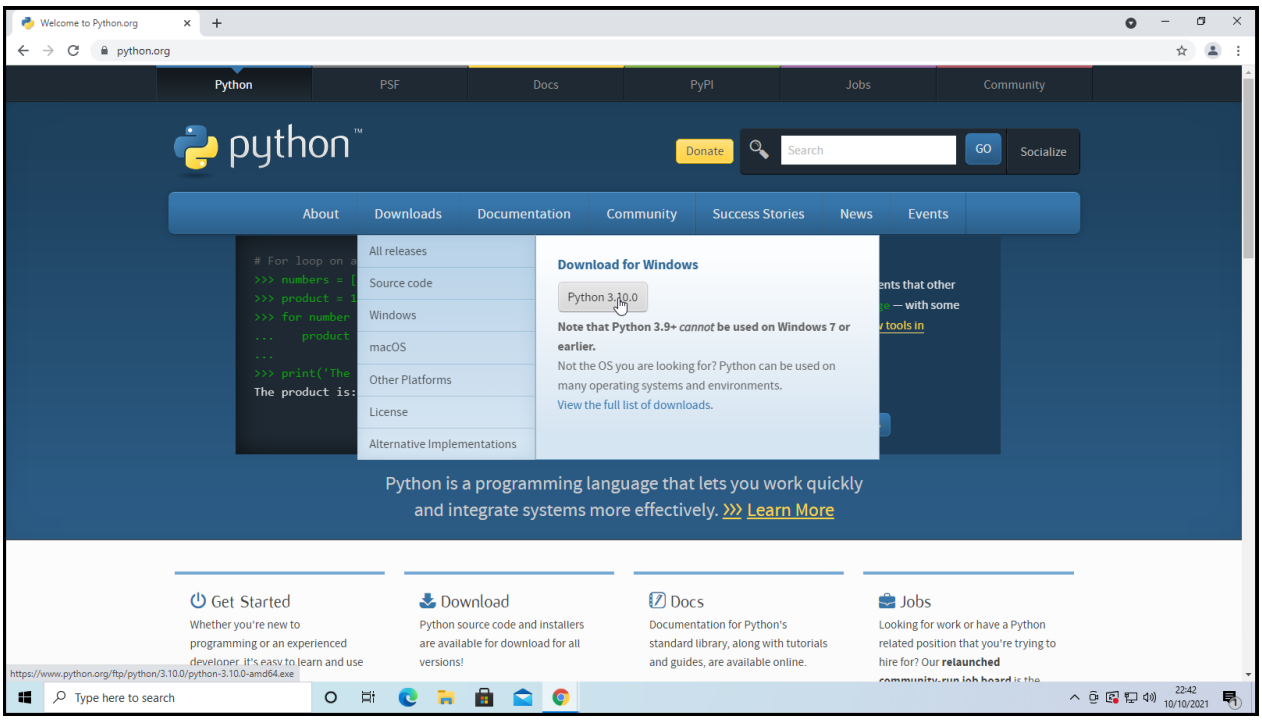
# How-to guide: Build Sowillo firmware with patched ESPhome on Windows

This document describes how to locally build the latest Sowillo firmware for the **Sowillo IoT board** on Windows.

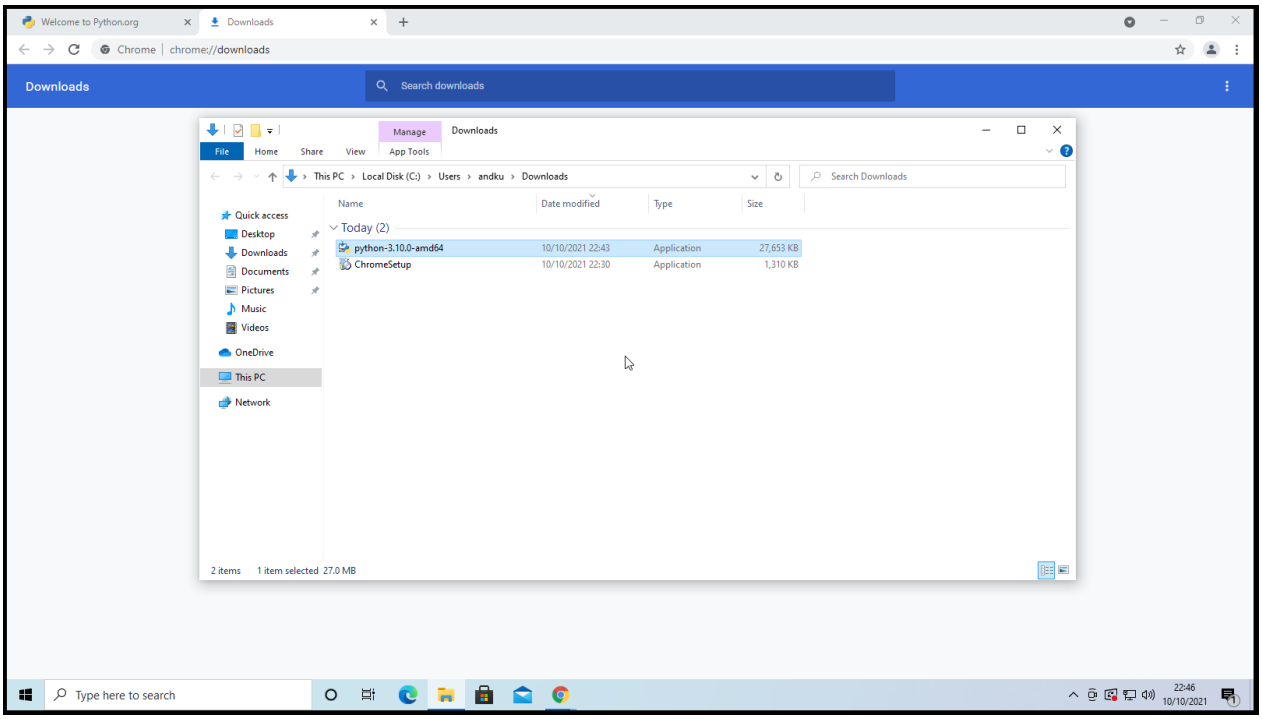
Please contact us if you have any questions

##1. Install python

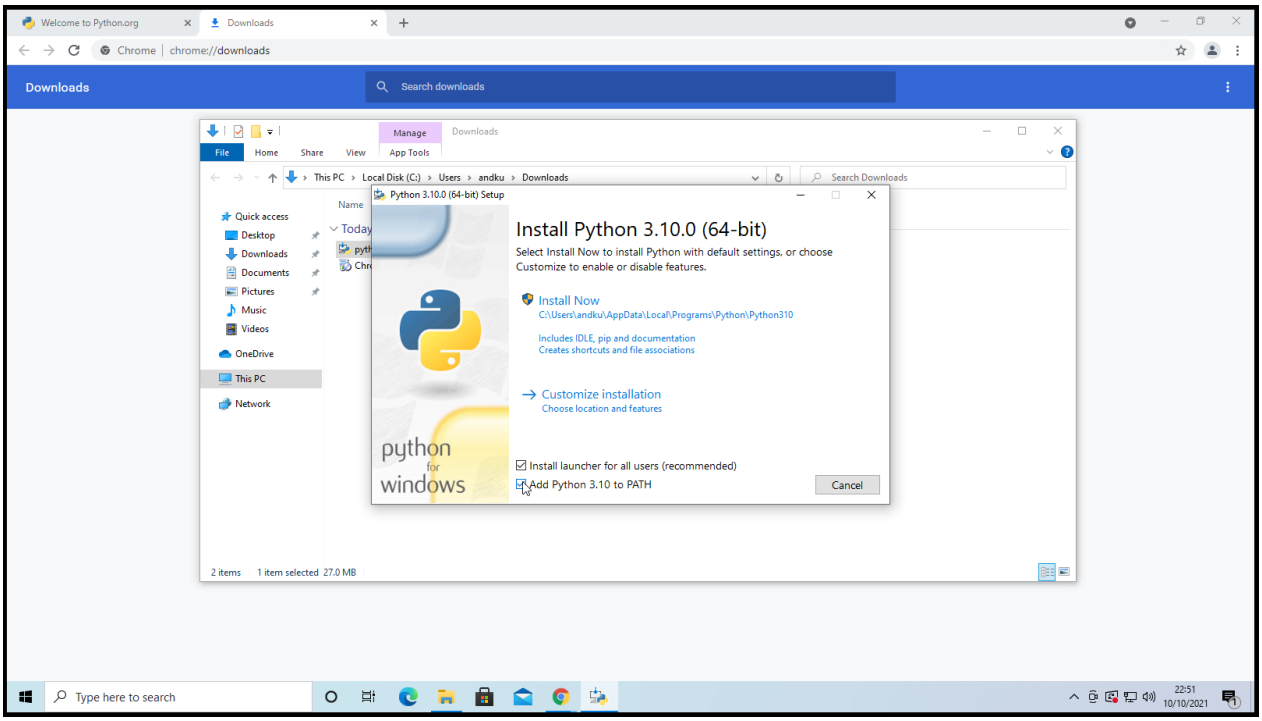
Chose the appropriate python version for your system on *python.org* and download it.




Run the installer.




On the first installation dialog select the option `Add Python to PATH` or `Add Python to environment variables`, this step is important, so far you will not be able to run `python` from windows console. If you missed it - you can add it manually later using `System Utility` in `Control Panel` on your Windows

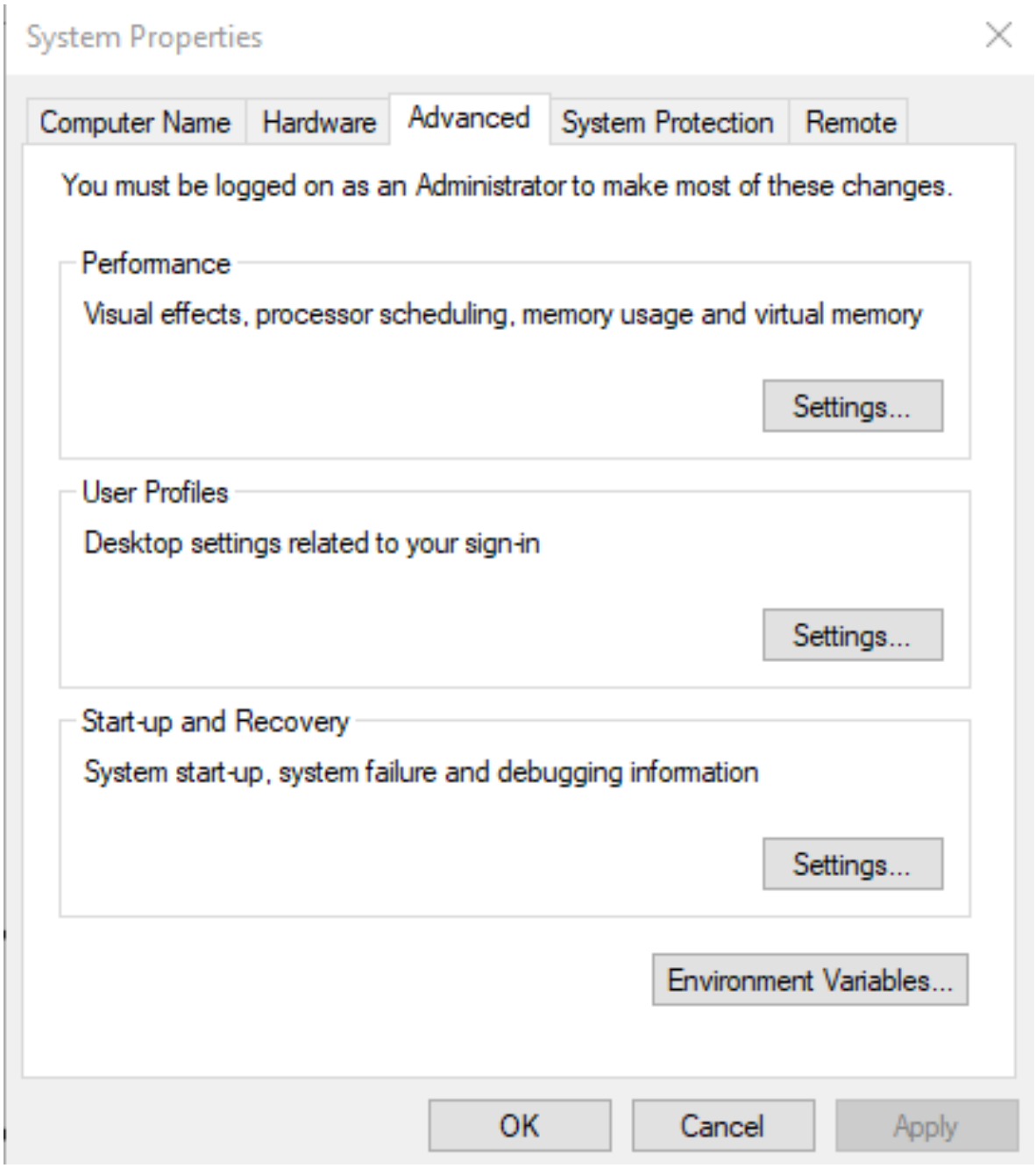



In case if you forgot to add the option `Add Python to PATH` - you won't be able to run `python` command from your `Windows Command Prompt`. Instead you will see the message

```
Python was not found; run without arguments to install from
the
Microsoft Store, or disable this shortcut from Settings
```

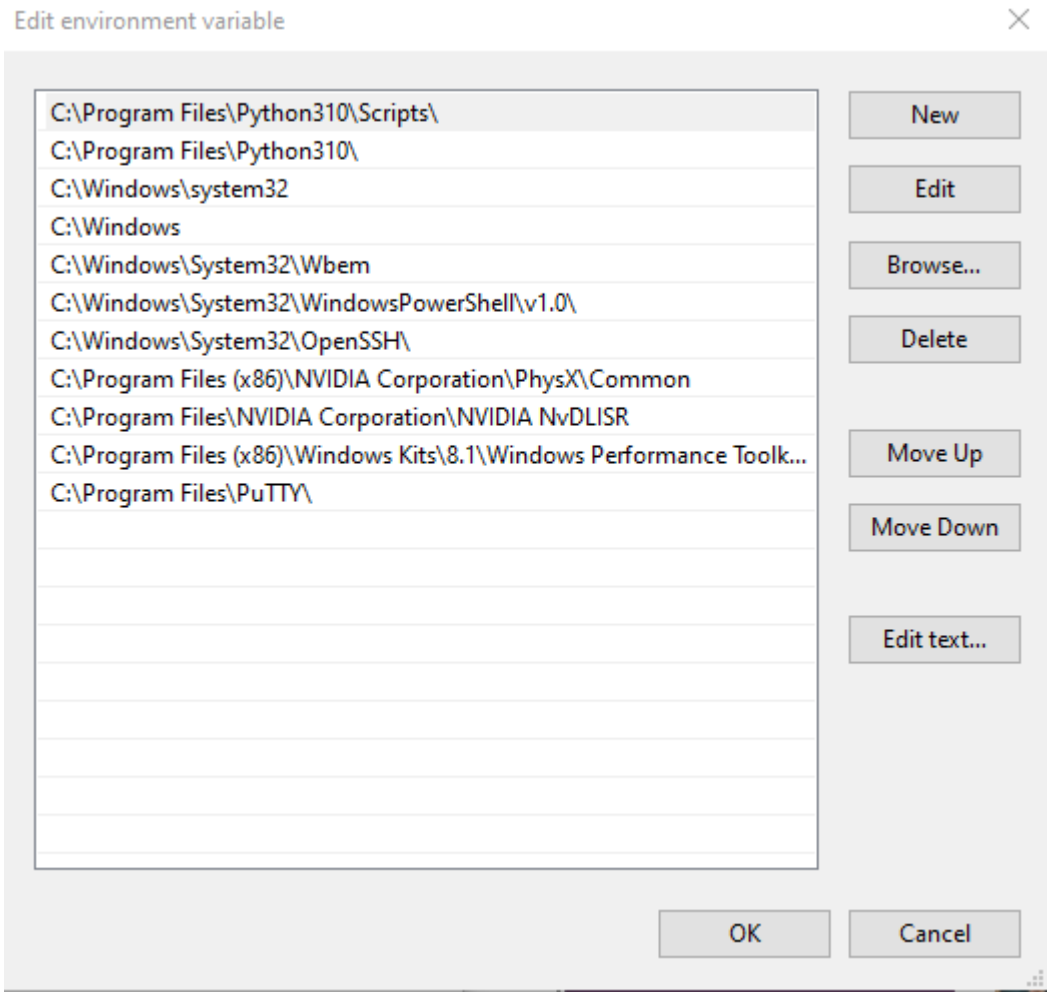
*Just in this case* do the following steps:

1. Go to the python installation directory and find there python.exe binary. You will need the path to the directory that contains the python.exe
2. Right-click on “This PC" and go to”Properties".




3. Click on the “Advanced system settings” in the menu on the left.
4. Click on the Environment Variables button on the bottom right.

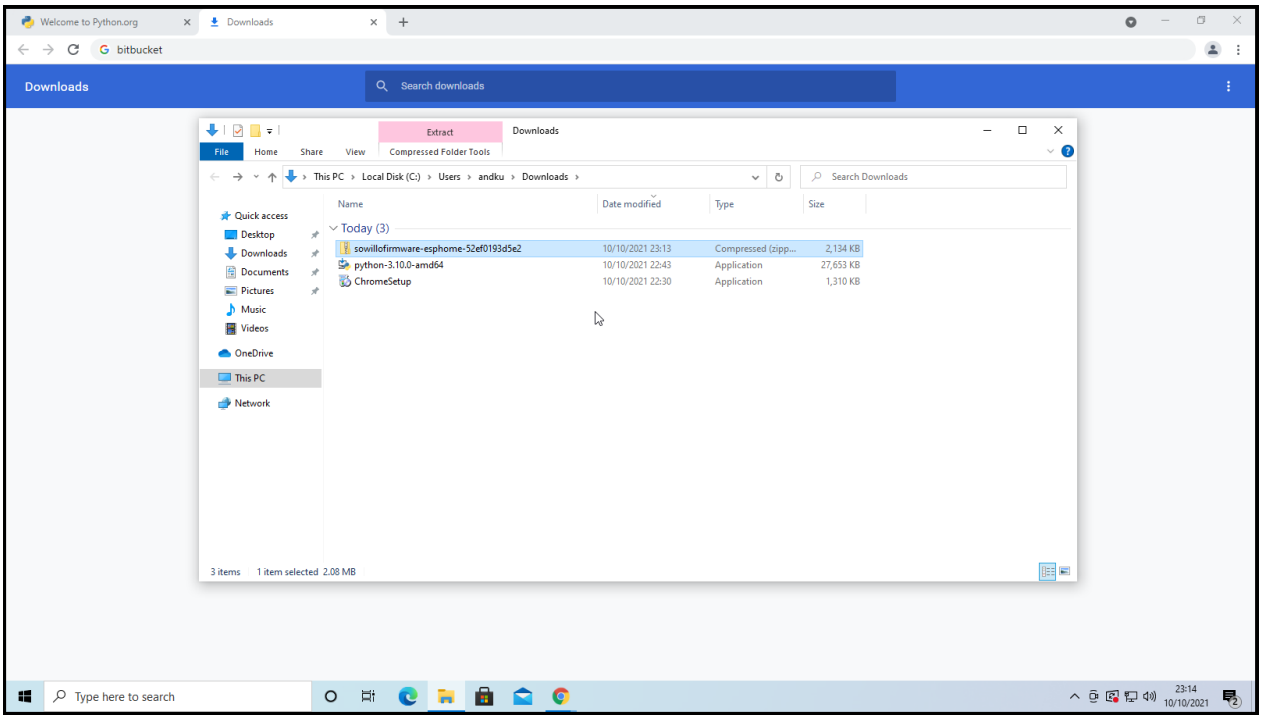
5. In the System variables section, select the “Path variable” and click on “Edit”. The next screen will show all the directories that are currently a part of the PATH variable.




6. Click on New and enter the path to the Python’s install directory.

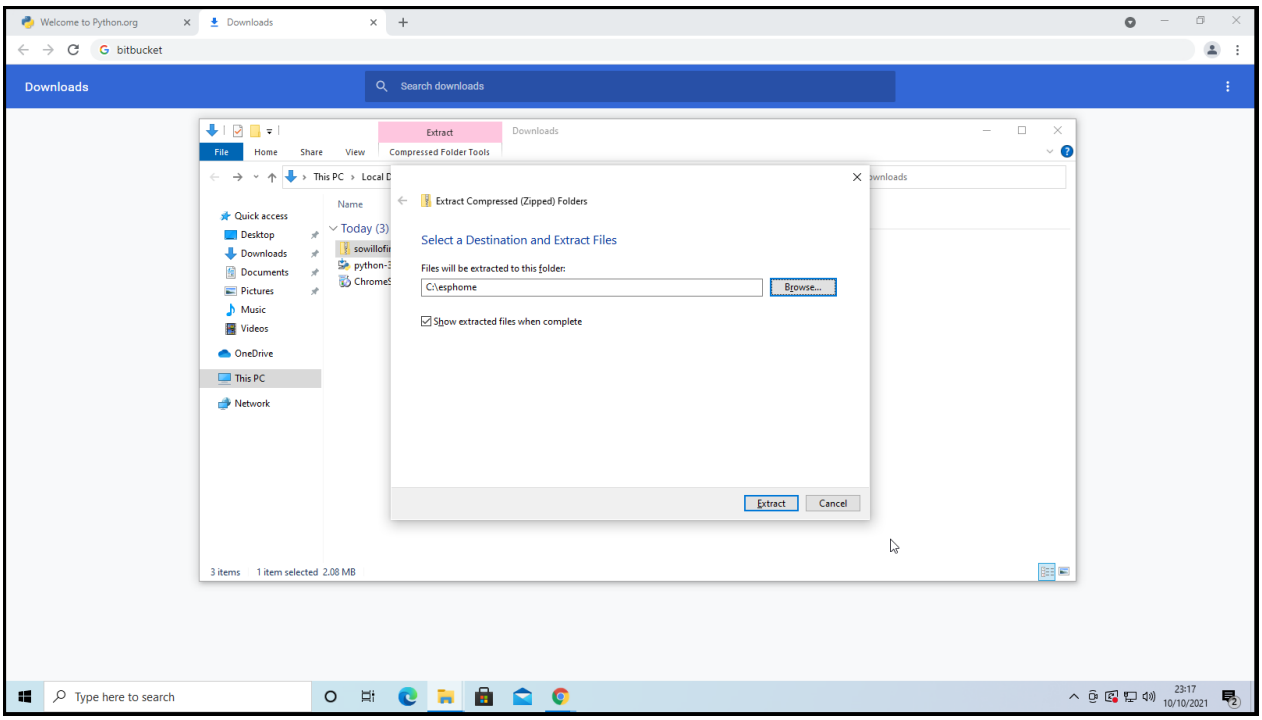
# 2. Download ESPHome:

Download the Sowillo's patched ESPHome from this link. **ESPHome** - is an open-source python package that allows you to build and flash firmware for ESP8266/ESP32 boards, in our case - ESP32. While we worked on our boards we found some issues in ESPHome, but fixes are still not included to upstream. For now we providing to the user the patched version of ESPHome and looking forward for our changes will be included to the ESPHome upstream.



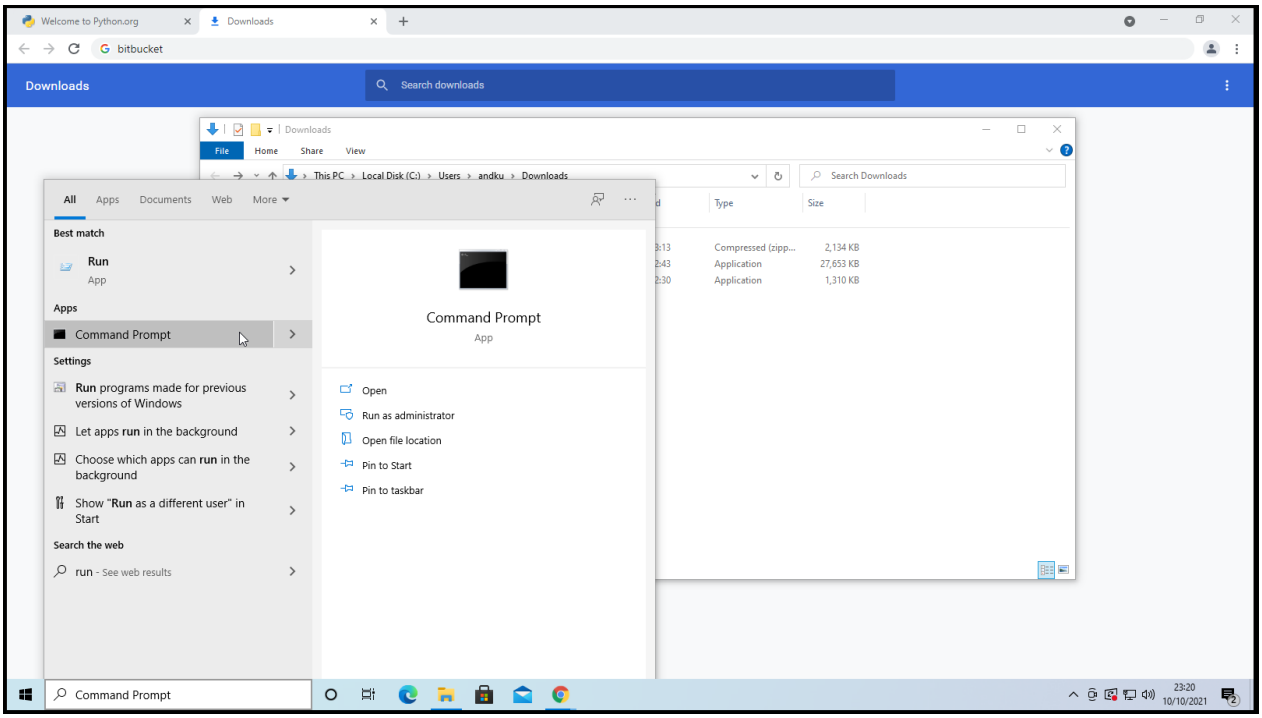

# 3. Install ESPHome

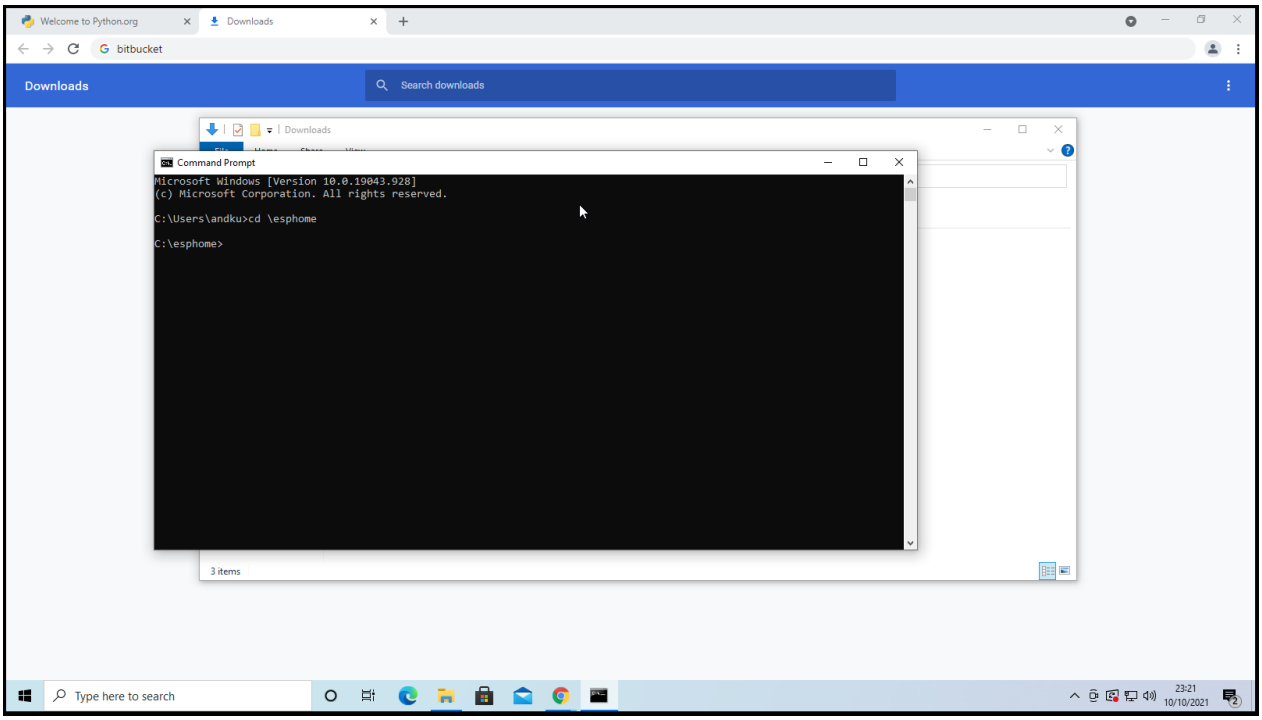
Unpack the ESPHome package to some directory. In this tutorial we'll use the`C:\esphome` path.




Open the Windows Command Prompt and navigate to the esphome directory. Note: this Command Prompt window must be opened *after* you installed python*, not before.*

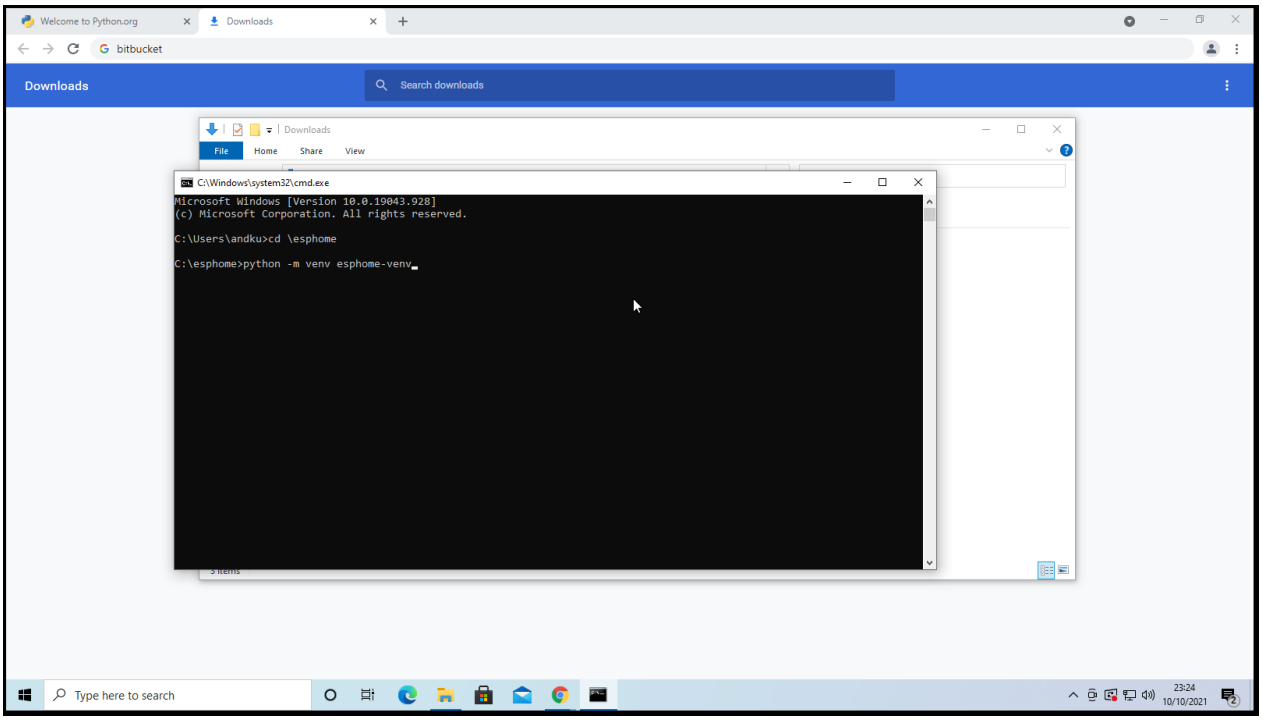
```
cd \esphome
```



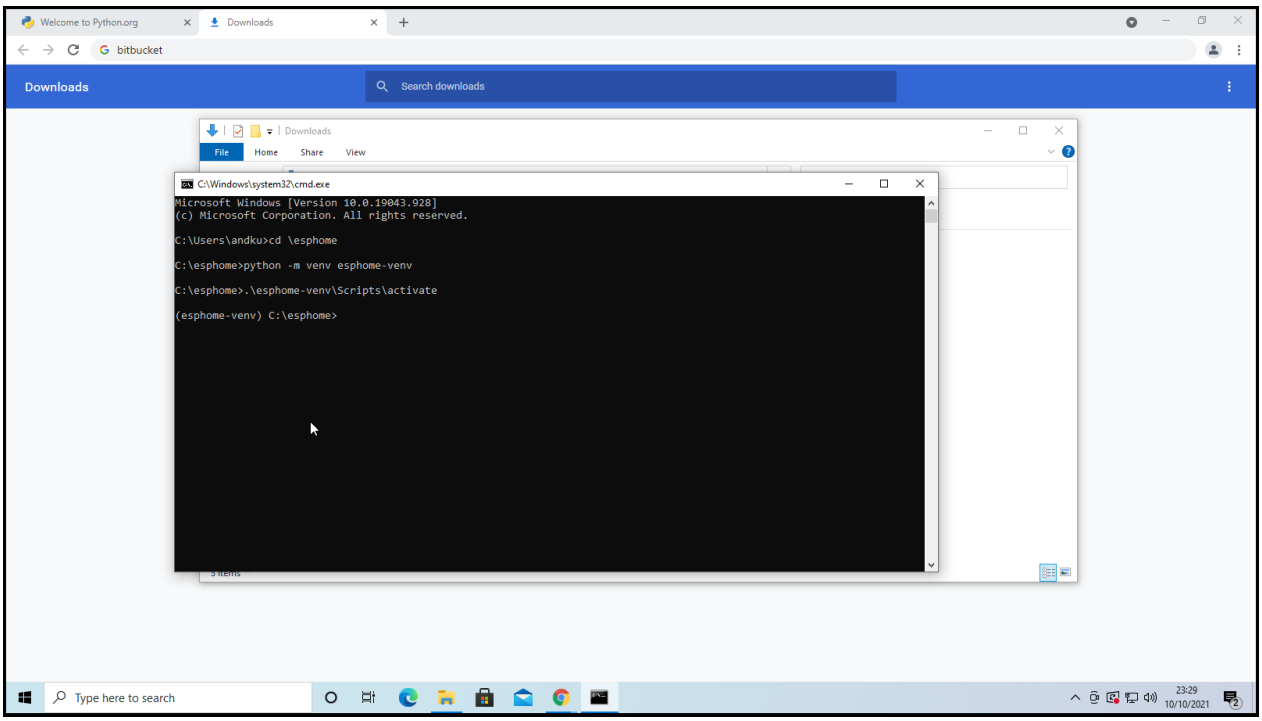

Now we'll create the python virtualenv for ESPHome. Don't worry, all further steps will not affect your system.

```
python -m venv esphome-venv
```




This will create a `esphome-venv` directory inside `esphome` with python virtual environment. Let's activate it:

```
.\esphome-venv\Scripts\activate
```



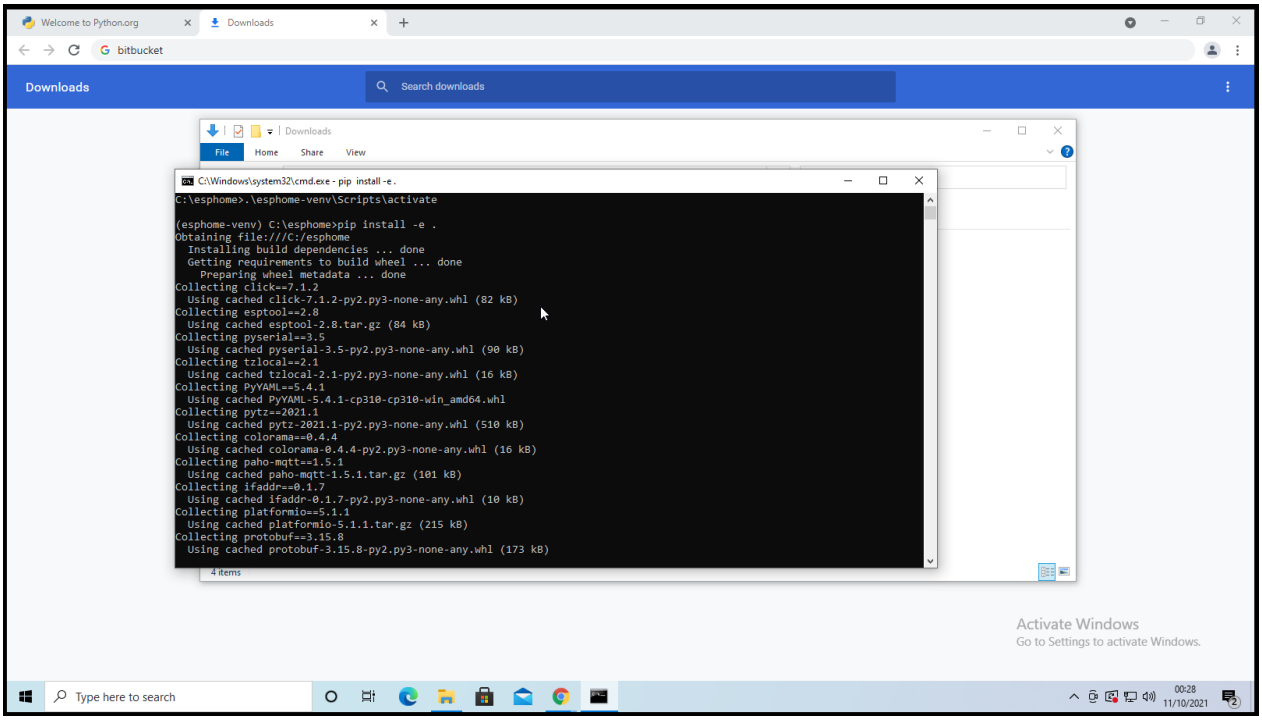

You may want to see the environment name at the beginning of the command prompt.

The next step is to install patched ESPHome package into this virtual environment:

(This command should be done from the esphome dir, don’t forget the dot at the end)

```
pip install -e .
```

Flashing dev board under windows requires the cp2102 win driver with enumerator, which could be downloaded by *link*.



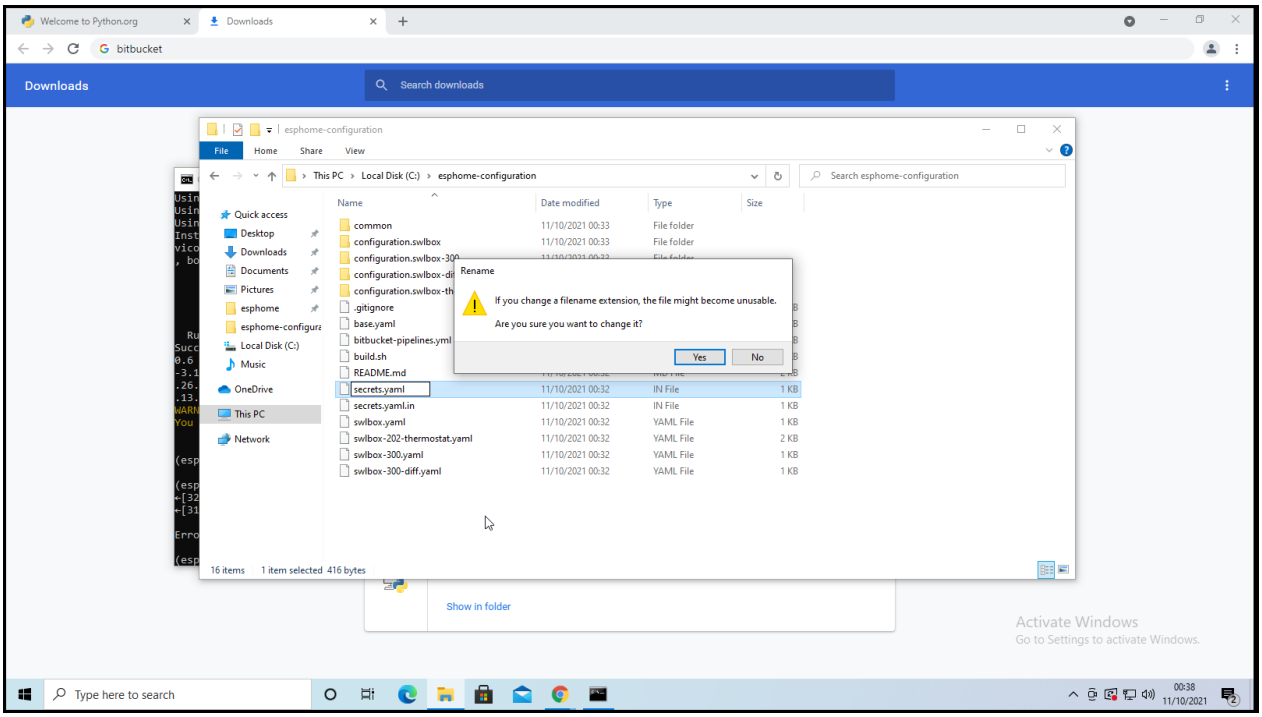

# 4. Build firmware

Download the Sowillo’s configuration for the ESPHome.

This archive contains the prepared configurations for the ESPHome.

Unpack the configuration for the ESPHome archive to some directory. In this tutorial we’ll use the `C:\esphome-configuration` path, you can change it according to your own needs.

*The mandatory step* - is to create the `secrets.yaml` file in the root of the directory. You can copy it from the `secrets.yaml.in` template and fill it in `secrets.yaml` file (file extension must be `.yaml`). You may be interested in wifi `wifi_ssid` and `wifi_password` fields that allow the board to connect to your wifi. Also you may want to configure `mqtt_broker` and `mqtt_port` on your mqtt broker.




Use the virtual environment that you created in the step above (esphome-venv). If you already closed the Windows Command Prompt - just open the new, navigate to the esphome directory and activate it:

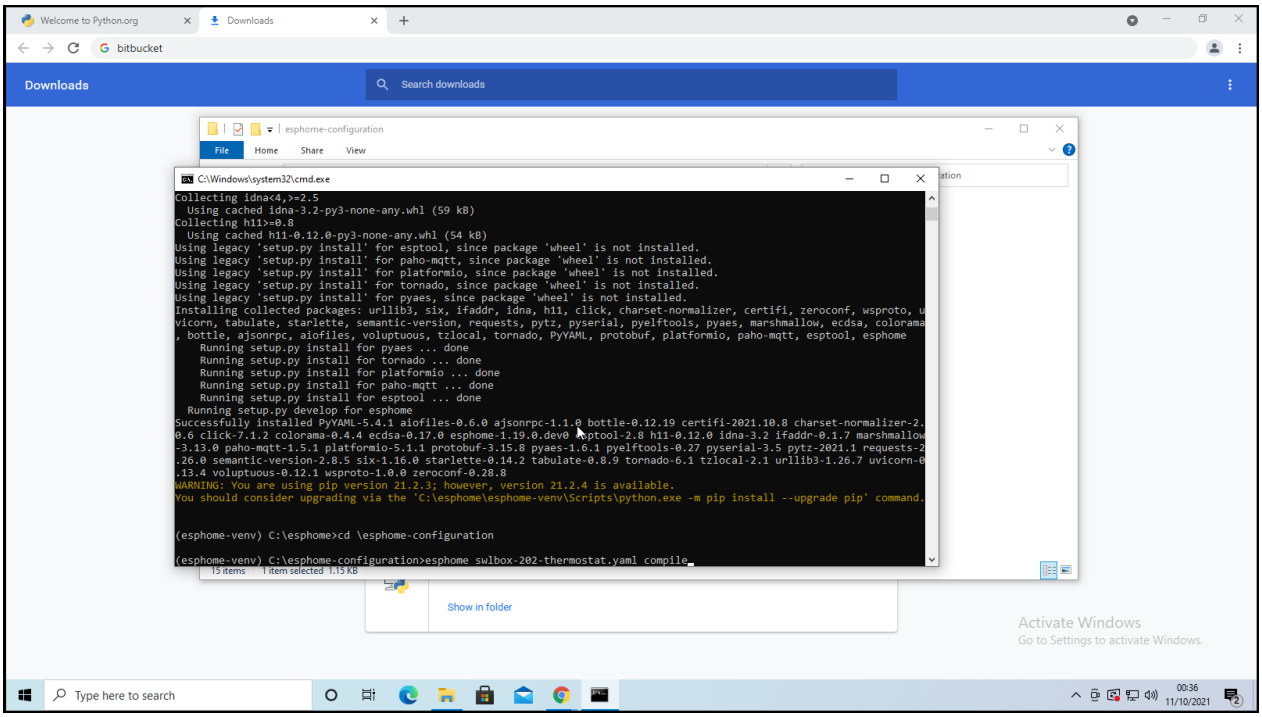
```
cd \esphome
.\esphome-venv\Scripts\activate
```

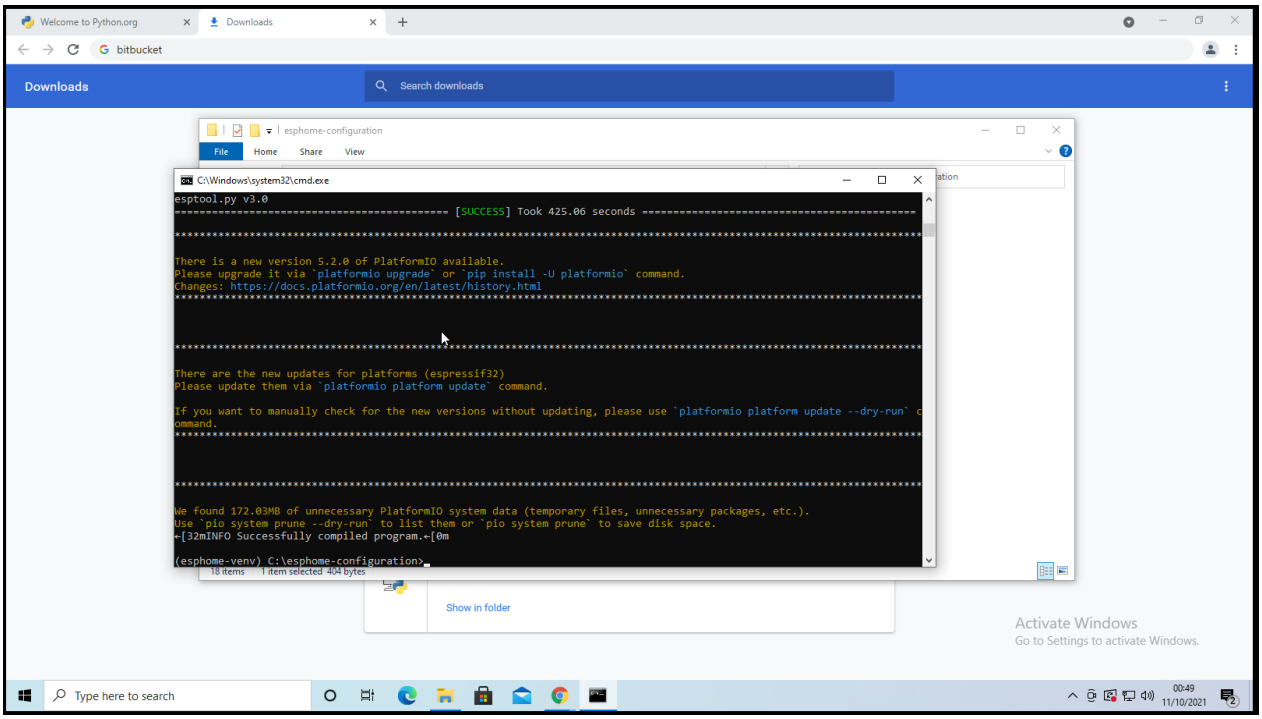
Now go to the configuration directory:

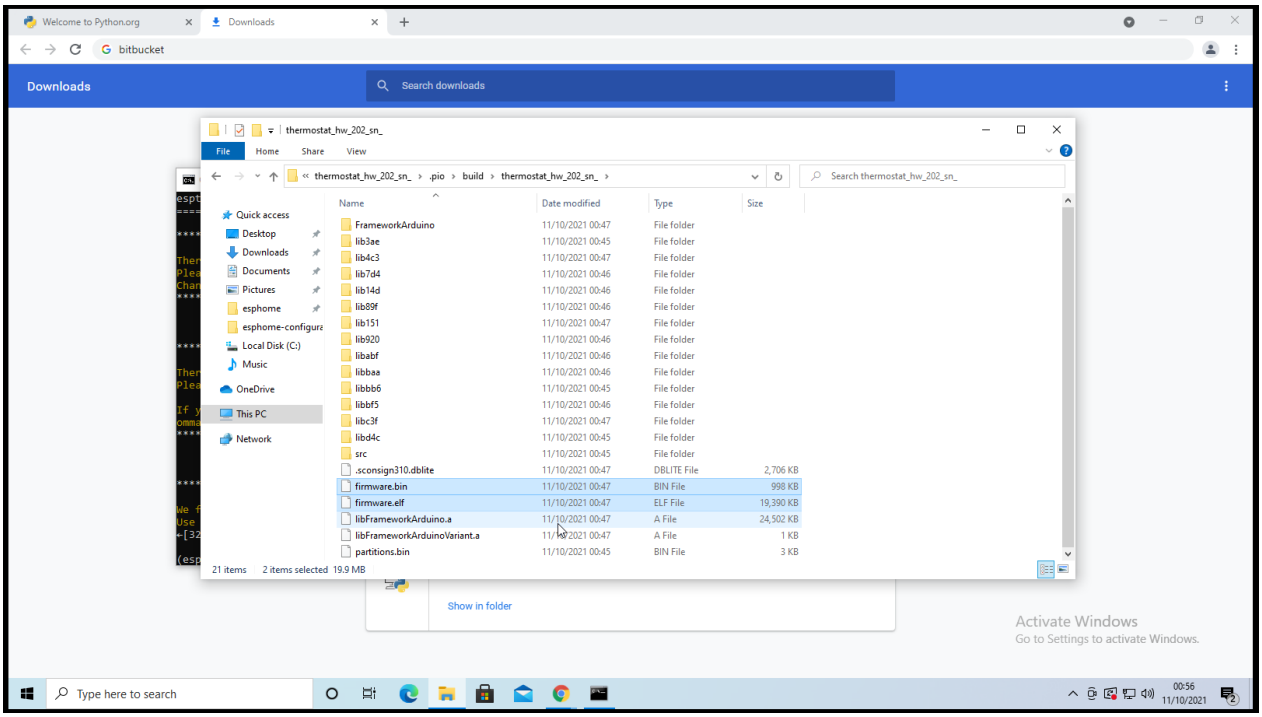
```
cd \esphome-configuration
```

Finally you can build the configuration that you want:

```
esphome swlbox-202-thermostat.yaml compile
```


Welcome to Python.org
Downloads
bitbucket
Downloads
Search downloads
esphome-configuration
File
Home
Share
View
C:\Windows\system32\cmd.exe
Collecting idna<4,>=2.5
Using cached idna-3.2-py3-none-any.whl (59 kB)
Collecting h11>=0.8
Using cached h11-0.12.0-py3-none-any.whl (54 kB)
Using legacy 'setup.py install' for esptool, since package 'wheel' is not installed.
Using legacy 'setup.py install' for paho-mqtt, since package 'wheel' is not installed.
Using legacy 'setup.py install' for platformio, since package 'wheel' is not installed.
Using legacy 'setup.py install' for tornado, since package 'wheel' is not installed.
Using legacy 'setup.py install' for pyaes, since package 'wheel' is not installed.
Installing collected packages: urllib3, six, ifaddr, idna, h11, click, charset-normalizer, certifi, zeroconf, wsproto, u
vicorn, tabulate, starlette, semantic-version, requests, pytz, pyserial, pyelftools, pyaes, marshmallow, ecdsa, colorama
, bottle, ajsonrpc, aiofiles, voluptuous, tzlocal, tornado, PyYAML, protobuf, platformio, paho-mqtt, esptool, esphome
Running setup.py install for pyaes ... done
Running setup.py install for tornado ... done
Running setup.py install for platformio ... done
Running setup.py install for paho-mqtt ... done
Running setup.py install for esptool ... done
Running setup.py develop for esphome
Successfully installed PyYAML-5.4.1 aiofiles-0.6.0 ajsonrpc-1.1.0 bottle-0.12.19 certifi-2021.10.8 charset-normalizer-2.
0.6 click-7.1.2 colorama-0.4.4 ecdsa-0.17.0 esphome-1.19.0.dev0 esptool-2.8 h11-0.12.0 idna-3.2 ifaddr-0.1.7 marshmallow
-3.13.0 paho-mqtt-1.5.1 platformio-5.1.1 protobuf-3.15.8 pyaes-1.6.1 pyelftools-0.27 pyserial-3.5 pytz-2021.1 requests-2
.26.0 semantic-version-2.8.5 six-1.16.0 starlette-0.14.2 tabulate-0.8.9 tornado-6.1 tzlocal-2.1 urllib3-1.26.7 uvicorn-0
.13.4 voluptuous-0.12.1 wsproto-1.0.0 zeroconf-0.28.8
WARNING: You are using pip version 21.2.3; however, version 21.2.4 is available.
You should consider upgrading via the 'C:\esphome\esphome-venv\Scripts\python.exe -m pip install --upgrade pip' command.
(esphome-venv) C:\esphome>cd \esphome-configuration
(esphome-venv) C:\esphome-configuration>esphome swlbox-202-thermostat.yaml compile
15 items
1 item selected 1.15 KB
Show in folder
Activate Windows
Go to Settings to activate Windows.
Type here to search
00:36
11/10/2021


Welcome to Python.org
Downloads
bitbucket
Downloads
Search downloads
esphome-configuration
File
Home
Share
View
C:\Windows\system32\cmd.exe
esptool.py v3.0
[SUCCESS] Took 425.06 seconds
There is a new version 5.2.0 of PlatformIO available.
Please upgrade it via `platformio upgrade` or `pip install -U platformio` command.
Changes: https://docs.platformio.org/en/latest/history.html
There are the new updates for platforms (espressif32)
Please update them via `platformio platform update` command.
If you want to manually check for the new versions without updating, please use `platformio platform update --dry-run` c
ommand.
We found 172.03MB of unnecessary PlatformIO system data (temporary files, unnecessary packages, etc.).
Use `pio system prune --dry-run` to list them or `pio system prune` to save disk space.
←[32mINFO Successfully compiled program.←[0m
(esphome-venv) C:\esphome-configuration>
18 items
1 item selected 404 bytes
Show in folder
Activate Windows
Go to Settings to activate Windows.
Type here to search
00:49
11/10/2021




# 5. Flash firmware

Initiate uploading firmware to ESP32 board with console command:

```
esphome swlbox-202-thermostat.yaml compile
```

after checking firmware readiness enter proper COM port connected to ESP32 board, or choose On-Air updating - if the board was flashed, but Firmware Over The Air update is necessary: