

Node-RED on UniPi Neuron

Step-by-step guide

The following manual will guide you through the process of powering up your UniPi Neuron controller. Before the start, please check the device for any damage that occurred during shipment, or any other visible defect.

Powering up

Power the device up and connect it to your computer according to the instruction leaflet included in the package.

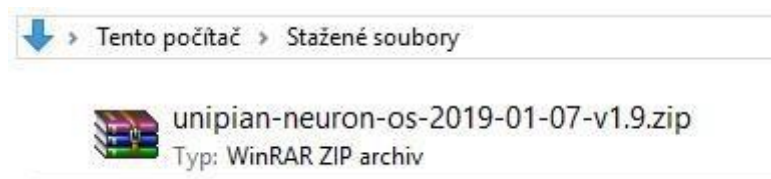
OS installation

If the controller is up and running, it is time for the next step – uploading the **UniPian Neuron OS** image to the controller's Raspberry Pi computing module, using an **SD card**.

Preparations

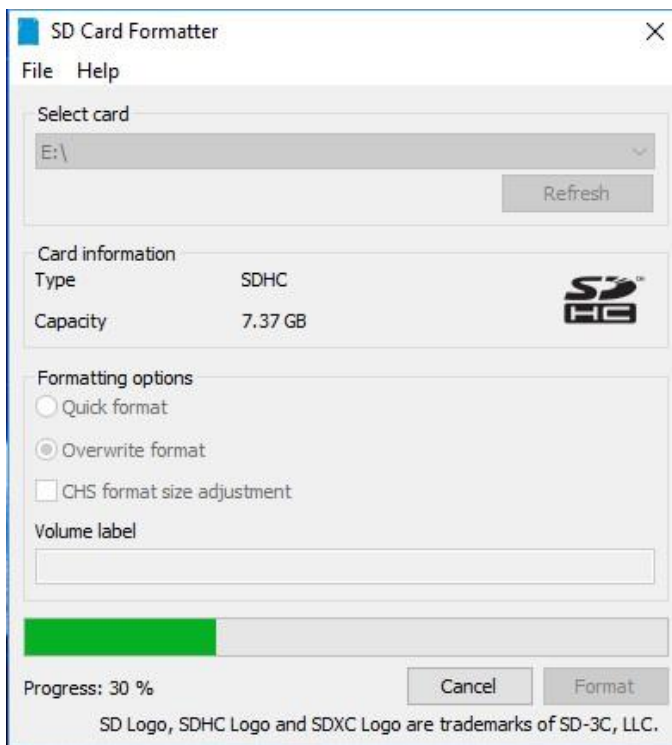
Prepare an SD card to which the OS will be burned. You can purchase the card in a single package with the controller – if the card was not purchased this way, you need to either [purchase](#) one or use your own card. The complete software package is approximately 2.5GB large; while a 4GB memory card would be sufficient theoretically, we strongly recommend using a **8 GB** or **16 GB**. The card's speed is important, we thus recommend to choose a faster card of **Class 10** or more.

After obtaining the card, download the **UniPian** OS image. In the [Files & Downloads](#) section, you can find the latest release; upon extracting the RAR archive, you will get an **.img** file ready to be burned into the SD card.



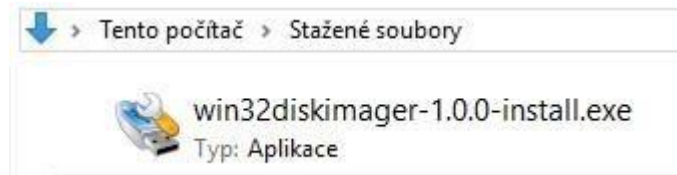
Screenshot of the downloaded UniPian RAR file

We also recommend to format the card first using the [SD Card Formatter](#).



Installing on Windows

Download a program for burning the image onto the SD card. The easiest method is to use the [Win32DiskImager](#) application. After its download and install, launch it as **Administrator**

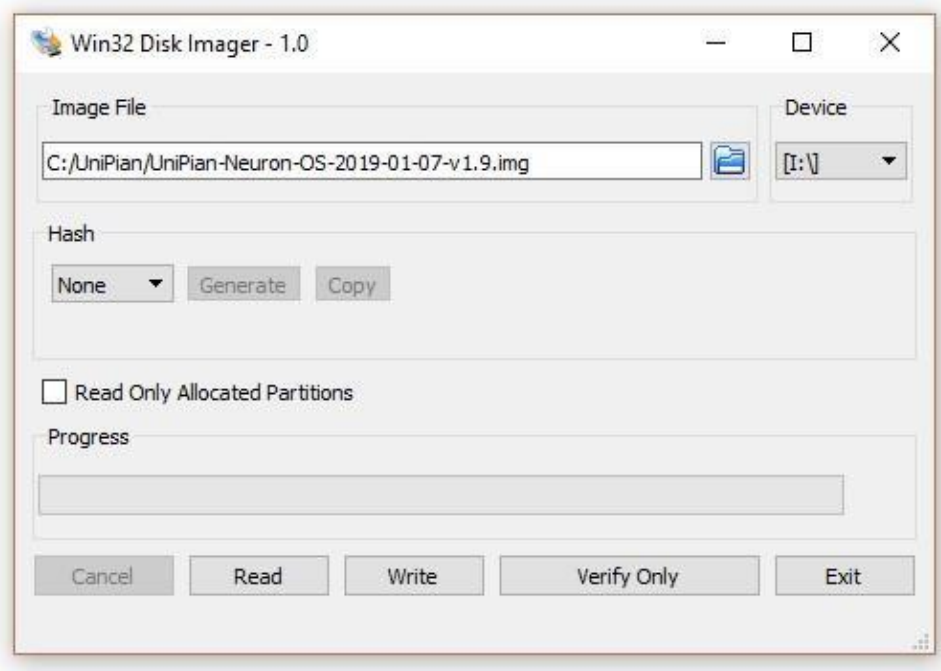


Screenshot of the downloaded Win32DiskImager installer

Launch the app's installer and follow displayed instructions.

With the installation completed, insert the SD card into your computer and start the Disk Imager.

In the program, choose the .img file and select the SD card (the „Device“ drop menu). Then click on "**Write**" and confirm the overwrite warning.



Note: The writing process will create two partitions on the SD card. Only one will be visible – the second, hidden partition has the ext4 Linux file format.

Installing on Linux

If you are a [Linux](#) user, the installation process uses the "**dd**" utility for low-level file copying. It is also possible to use various extensions such as the [Image Writer](#) in Ubuntu, but only „dd“ guarantees a 100% functionality. Installation itself is then performed by following these steps

- ✦ Extract the UniPian Neuron archive using `unzip` command `unzip UniPian-Neuron-2019-01-07-v1.9.zip`
- ✦ Run `df -h` command to return all connected directories. If the SD card is inserted, it will appear either as a `/dev/mmcblk0p1` or `/dev/sdd1`. A specific example can look like this `/dev/mmcblk0p1 7,3G 4,0K 7,3G 1% /media/user/SDcard`.

The last part („p1“ or „1“) represents the directory number. **When writing the command it is necessary to describe the whole path to the SD card, not only the directory.** The `dd` command parameter must contain `"dev/mmcblk0"` or `"/dev/sdd"`.

Note: The writing process will create several partitions on the SD card, as the Raspberry Pi has several directories

- ✦ Once you determine the path to the SD card, remove it from the PC; if the card contains several directories, it is necessary to disconnect them individually through the `umount` command. The command line must contain the whole identification of the directory, eg. `"/dev/mmcblk0p1"` or `"/dev/sdd1"`.
- ✦ Now onto the "`dd`" command itself. The command has three parameters
 - an input file (*input file if=*) – here, insert the downloaded image:
`if=UniPian-Neuron-2019-01-07v1.9.img`

- an output file (*output file of=*); insert the SD card: `of=/dev/mmcblk0`. Bear in mind it is again necessary to state the SD card itself, not only a directory such as `sdds1`, `sddp1` or `mmcblk0p1`.
- A block size (*block size bs*). This parameter determines the size of blocks used for writing onto the SD card. The default size is 4 MB – in case of issues, it is also possible to use 1MB size. The resulting command to write data looks like the following:

```
sudo dd bs=4M if=UniPian-Neuron-2019-01-07-v1.9.img of=/dev/mmcblk0
```

- ✦ The "`dd`" program does not indicate its operation, nor displays any information. Writing onto the SD card may take up to 5 minutes; that said, simply wait until some result will display.
- ✦ Upon completing the operation, the SD card should display two partitions – you can check using the `df -h` command.

If the process fails for any reason, it might be useful to follow the **SD card preparation manual** accessible on this link [zde](#).

Editing the SD card directory

If the image was burned successfully, the card will load as `boot` when inserted to the PC.

Before inserting the card into the controller, it is necessary to create **ssh file**. The easiest method is to create a .txt file named `ssh.txt` [Inserting the card](#)

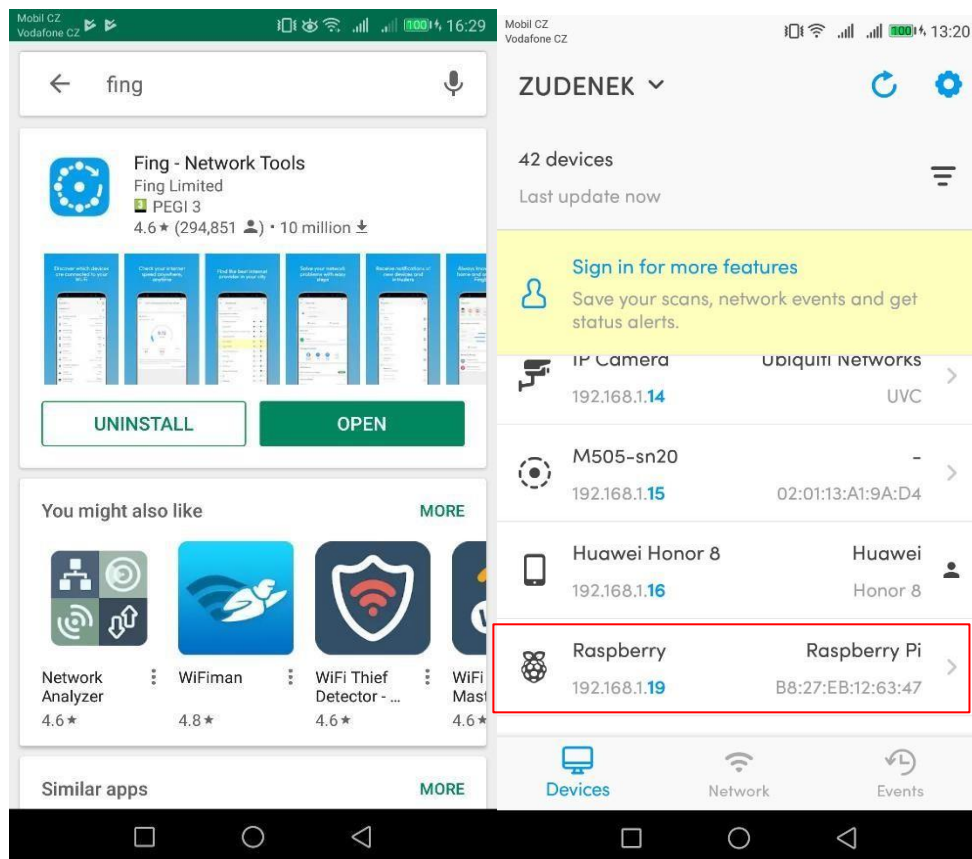
Upon burning the image and adding the `ssh` file, the card is ready to be inserted into the controller.

WARNING: Insert the card ONLY with the controller's power unplugged!

Detecting the UniPi Neuron IP address

Now, it is necessary to detect the controller's IP address. You can find it by multiple methods

- ✦ Probably the easiest and fastest way is to use a smartphone with the `Fing` app installed (it is possible to use other similar applications). The app is available on Google Play for Android phones, or App Store for iOS

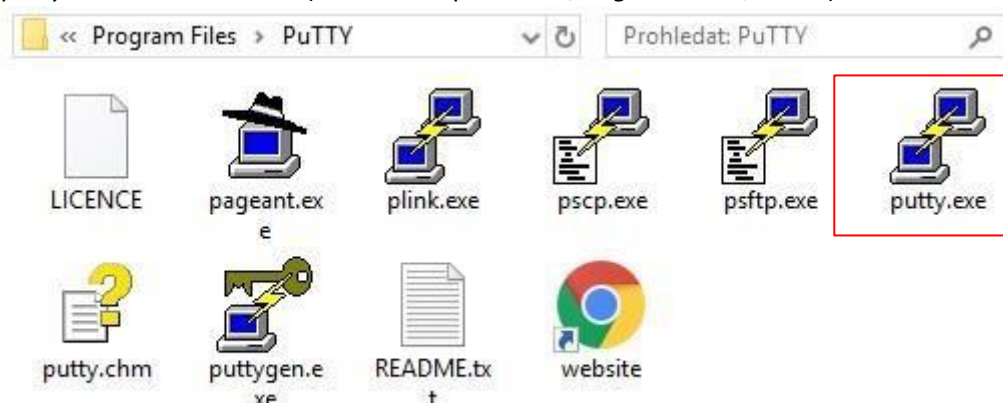


- ✦ using Windows IP address lookup tools or any other IP address lookup tool

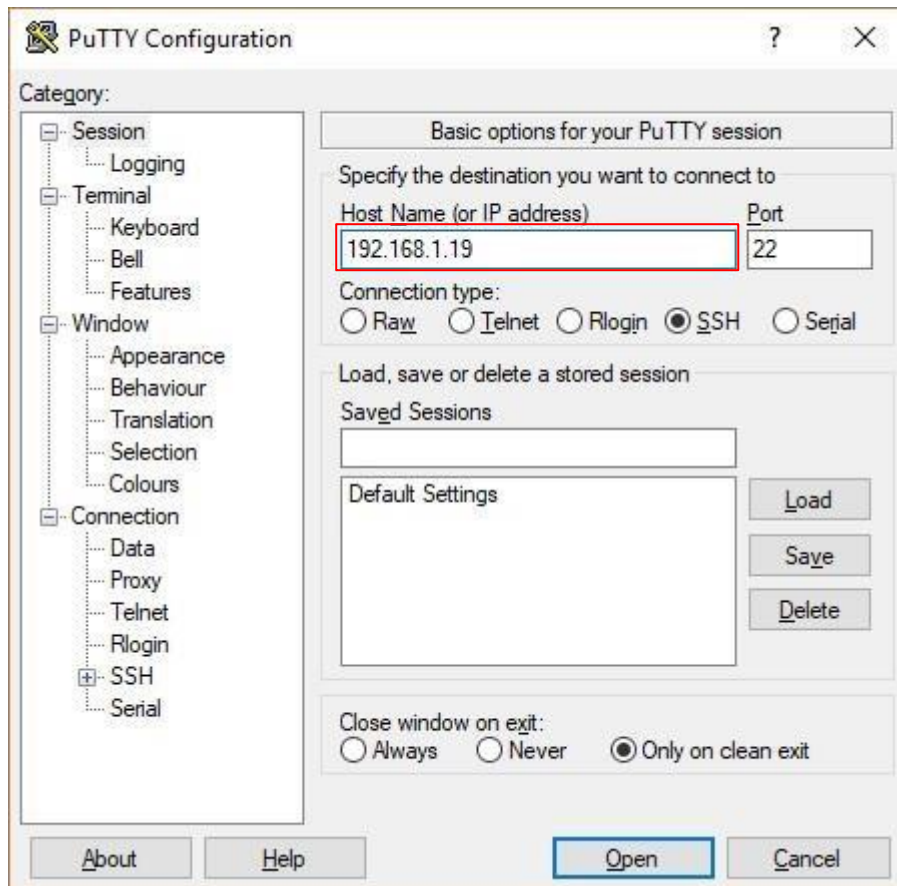
Communication through the SSH using PuTTY

Once the controller's IP address is found, you can communicate with it through an SSH connection. For this purpose the PuTTY program is needed – while other SSH programs can be used, PuTTY is the most suitable for Windows). The program can be downloaded [here](#).

After the download, install the application following the installer instructions. With the program installed, click on the *putty.exe* file to launch it (the default path is *C:\Program Files\PuTTY*.)



When launched, the following window is displayed. Enter the controller's IP address to the *Host Name* tab, then click on *Open*.



A *PuTTY Security Alert* window will appear. Select „Yes“ option and continue.



After clicking Yes, an SSH terminal command prompt will open.

Note: The login has a set timeout, after which the login will not be possible! If the time runs out, restart PuTTY and repeat the login.

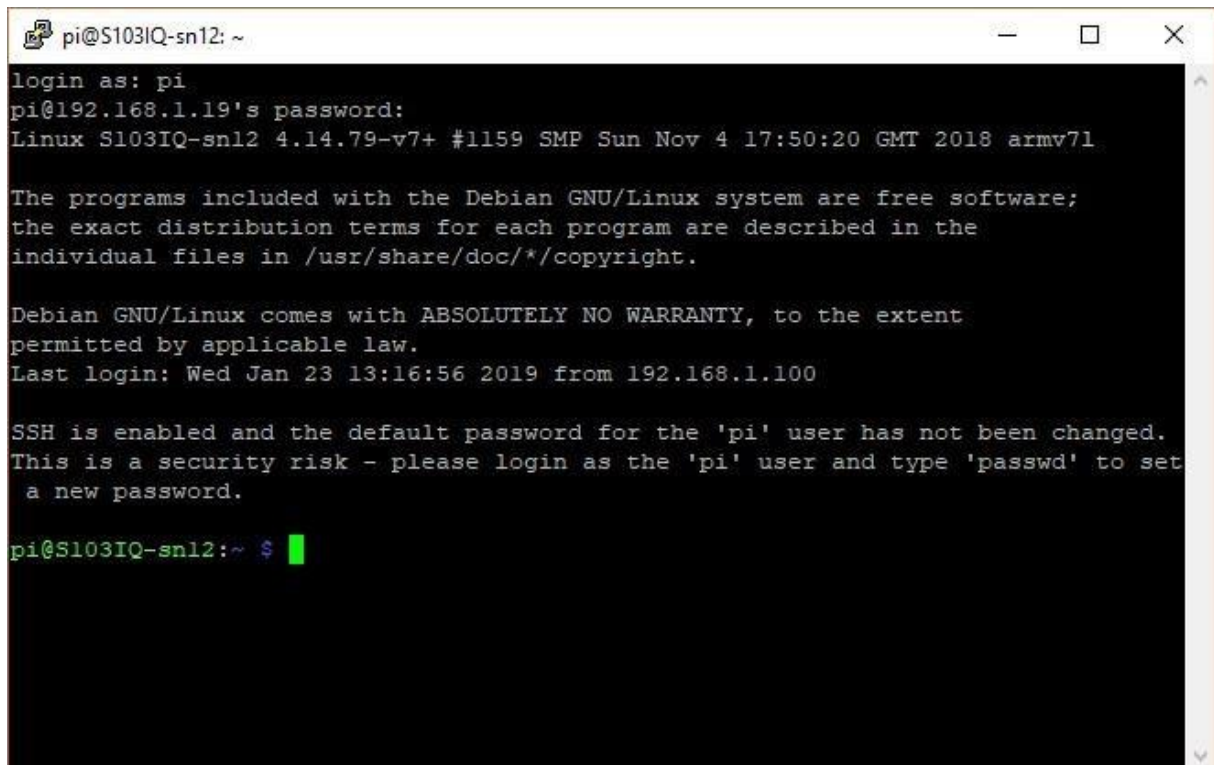
Enter the following login:

Login: **pi**

Password: **raspberrypi**

Note: The password text is written as invisible – if the cursor is not moving during the password insertion, it is not an error!

Upon successful login, this screen will appear:



```
pi@S103IQ-snl2: ~  
login as: pi  
pi@192.168.1.19's password:  
Linux S103IQ-snl2 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed Jan 23 13:16:56 2019 from 192.168.1.100  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
pi@S103IQ-snl2:~ $
```

With the SSH running, the controller is ready for the upload of remaining parts of the Node-RED setup.

Installing EVOK

The EVOK install is described in detail on [this link](#). The link contains a brief description of EVOK and the install guide.

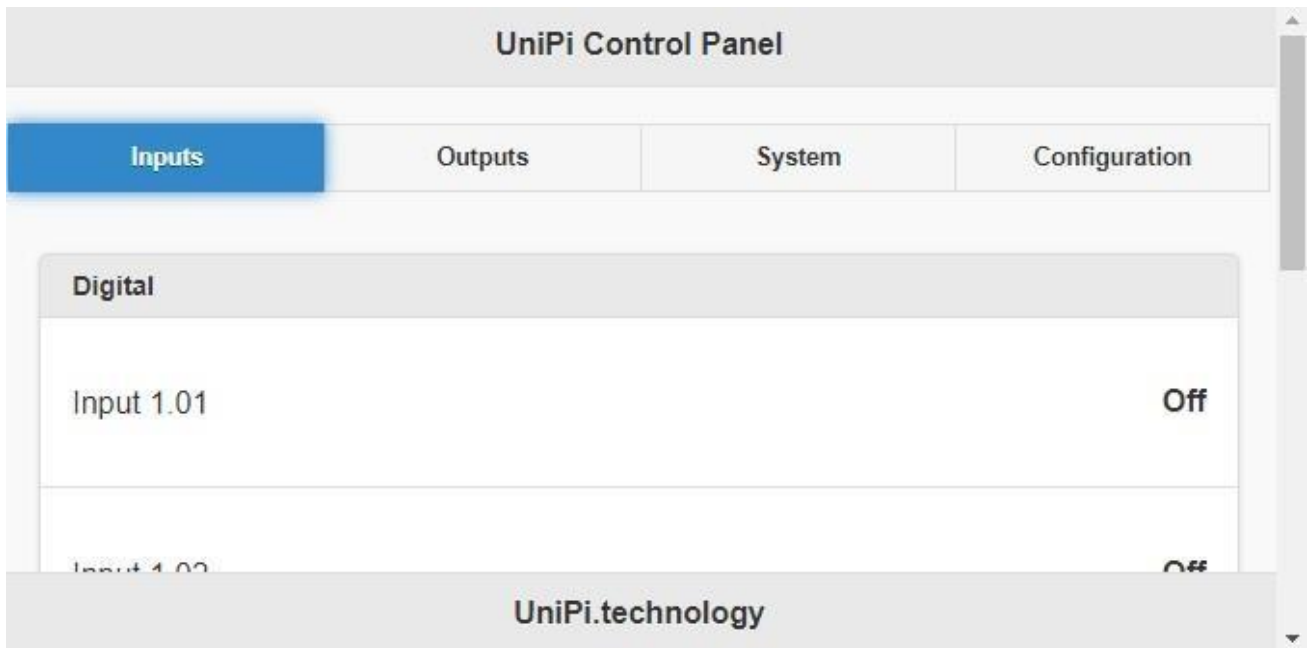
For the purpose of this guide, only several commands will be needed to be inserted into the SSH terminal. Bear in mind that it is recommended to paste commands by lines. It is also possible to run several blocks at once, but we strongly recommend to perform the procedure one step at a time.

It is worth noting that if the UniPian image was used, the EVOK is already contained within it and it is not needed to install it again. It is, however, necessary to check if the latest version is used. All you need to do is to use following commands:

- ✦ **sudo su**
- ✦ **apt-get install evok**
- ✦ **apt-get update**
- ✦ **reboot**

Note: The Raspbian image can be also used instead of UniPian. This option however requires the EVOK to be installed according to the installation manual on GitHub)

If everything was done correctly, you can now open a web browser and insert the IP address of your controller. The EVOK user interface will open.



Installation of Node-RED

Now, it is time to install the Node-RED itself. To use Node-RED, **the latest EVOK release is required (2.0.7g or higher)**.

The installation is again performed through the SSH terminal. Insert the first command

✦ `sudo apt-get install build-essential`

The installation should run without any further input. Upon its completion, insert the following command:

✦ `bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)`

To finish the install, three questions will appear.

1. Are you really sure you want to install as root? [y/N]?
2. Are you really sure you want to do this? [y/N]?
3. Would you like to install the Pi-specific nodes? [y/N]?

Answer them in the following order – **Y, Y, N** (check the following screenshot for reference)


```
Node-RED update
root@S103IQ-snl2:/home/pi# bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)

Root user detected. Typically install as a normal user. No need for sudo.

Are you really sure you want to install as root ? (y/N) ? y

This script will remove versions of Node.js prior to version 7.x, and Node-RED and
if necessary replace them with Node.js 10.x LTS (dubnium) and the latest Node-RED
from Npm.

It also moves any Node-RED nodes that are globally installed into your user
~/.node-red/node_modules directory, and adds them to your package.json, so that
you can manage them with the palette manager.

It also tries to run 'npm rebuild' to refresh any extra nodes you have installed
that may have a native binary component. While this normally works ok, you need
to check that it succeeds for your combination of installed nodes.

To do all this it runs commands as root - please satisfy yourself that this will
not damage your Pi, or otherwise compromise your configuration.
If in doubt please backup your SD card first.

Are you really sure you want to do this ? [y/N] ? y

Would you like to install the Pi-specific nodes ? [y/N] ? N
Running Node-RED update for user root at /root

This can take 20-30 minutes on the slower Pi versions - please wait.

  Stop Node-RED                ✓
  Remove old version of Node-RED ✓
  Remove old version of Node.js ✓
  Install Node.js LTS
  Clean npm cache
  Install Node-RED core
  Move global nodes to local
  Install extra Pi nodes
  Npm rebuild existing nodes
  Add shortcut commands
  Update systemd script
```

Node-RED installation

All steps must be marked with the green tick in order to complete the install. When all steps are marked as completed, the Node-RED is successfully installed.

By default, **node-red** command must be used each time to run Node-RED. In order to make things easier, enter a one additional command to the SSH terminal:

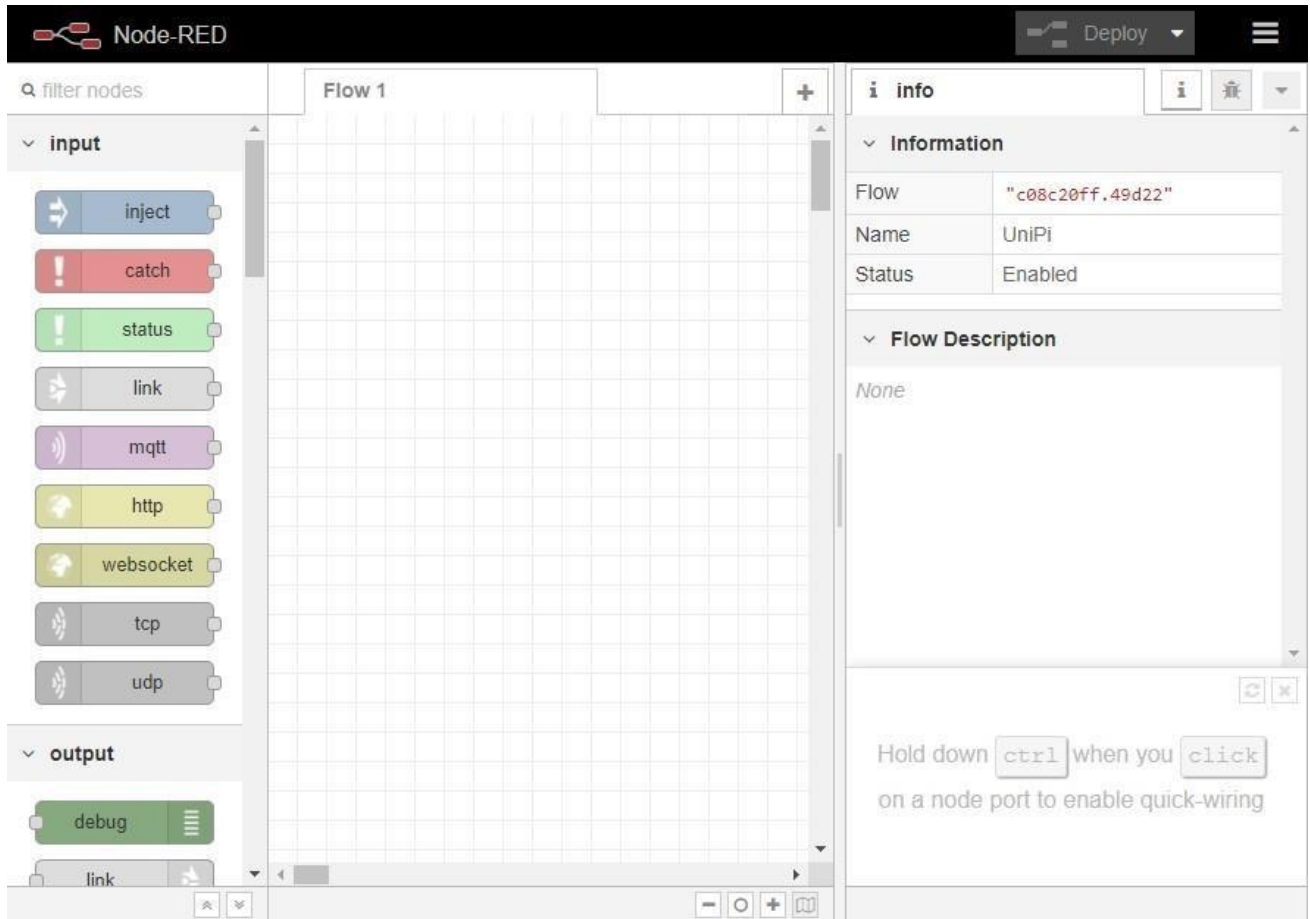
✦ **sudo systemctl enable nodered.service**

This command will set Node-RED to launch on device startup.

You can now enter the Node-RED user interface. All you need to do is to open a web browser, enter the controller's IP address and place a **:1880** suffix.



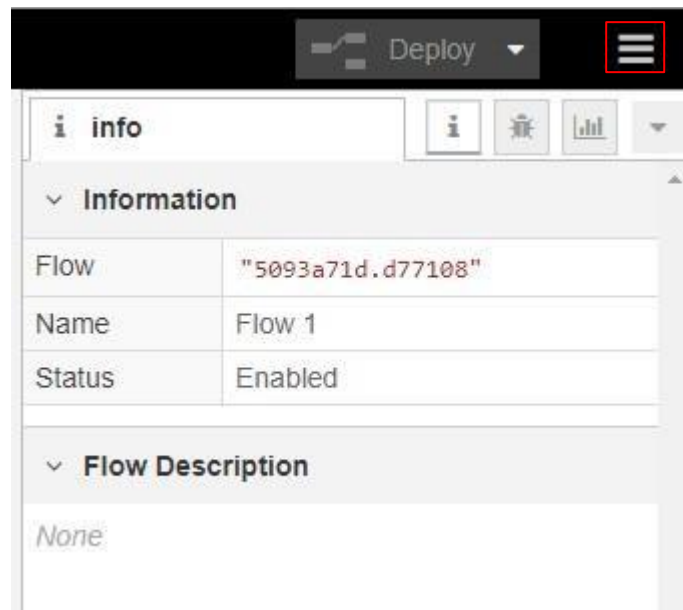
A following user interface will load



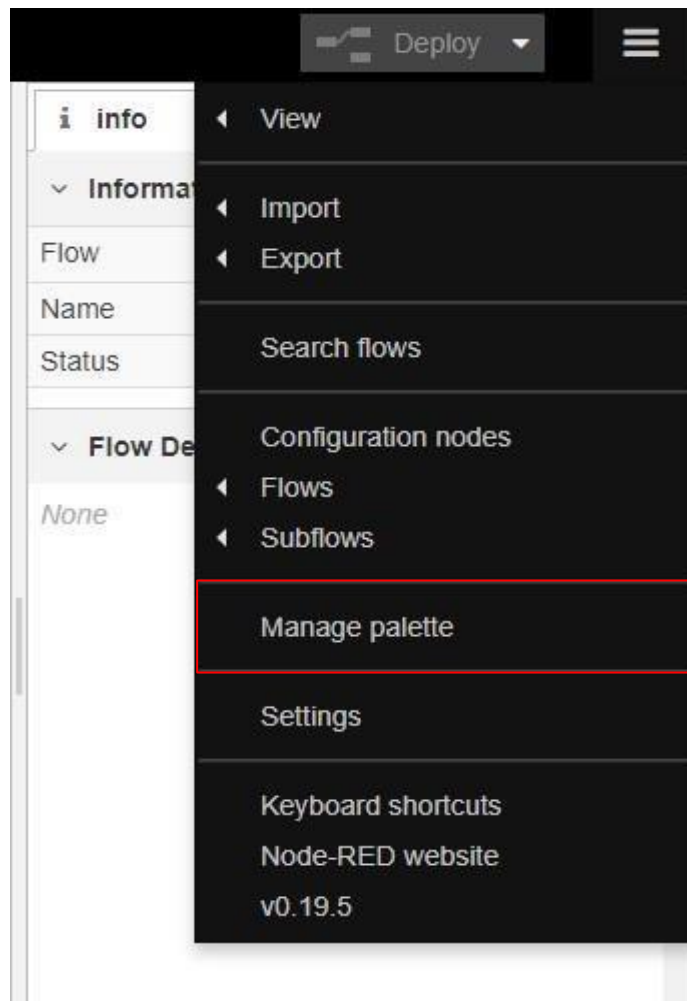
Node-RED interface

Preparing the Node-RED

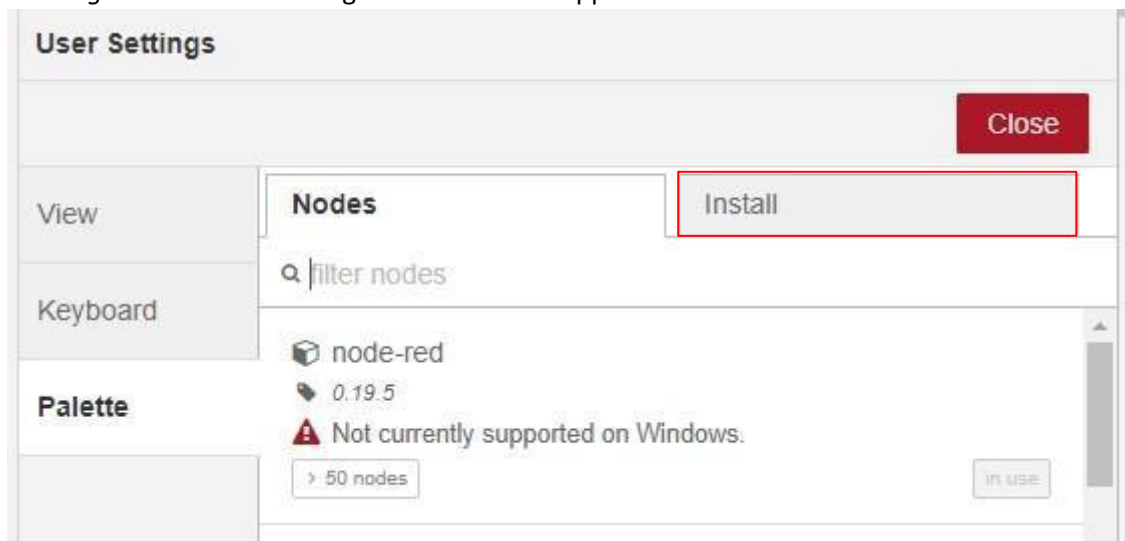
With the Node-RED functional, you will need to install nodes required for communication with the controller (nodes are displayed in the left tab). First, click on MENU on the right upper corner of your screen



A following menu will appear

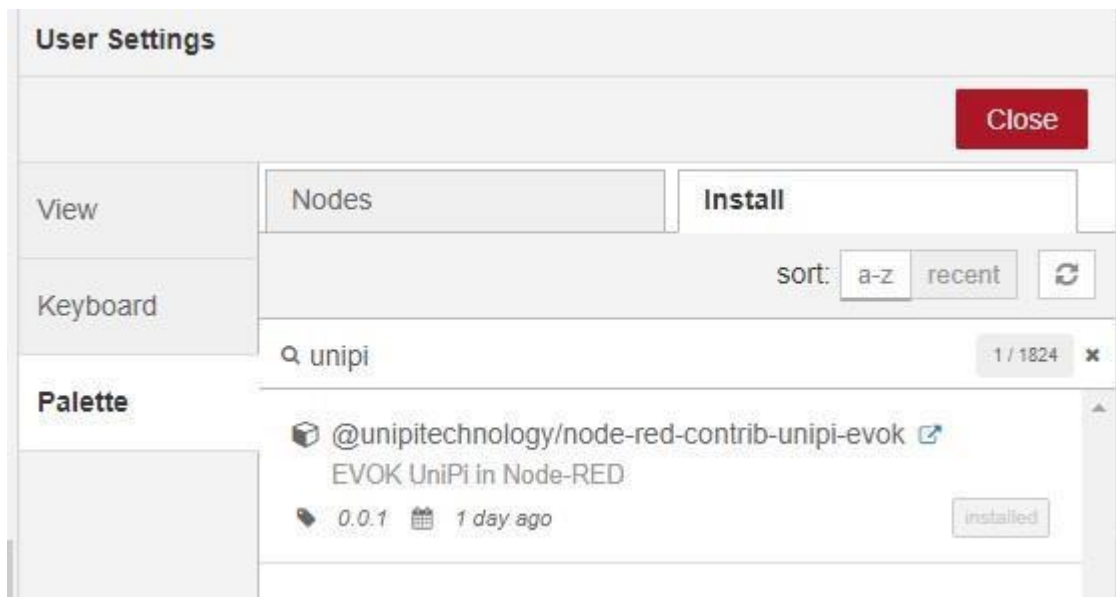


Click on *Manage Palette*. A following list of nodes will appear. Click on *Install*.

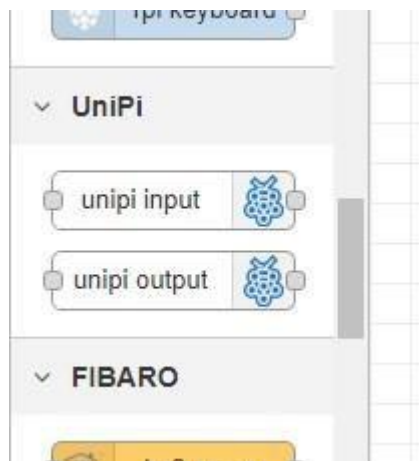


You can now install new nodes into Node-RED. You will specifically require two packages. The first one is [@unipitechnology/node-red-contrib-unipi-evok](#). Enter **unipi** search word to find the package. Then click *Install*





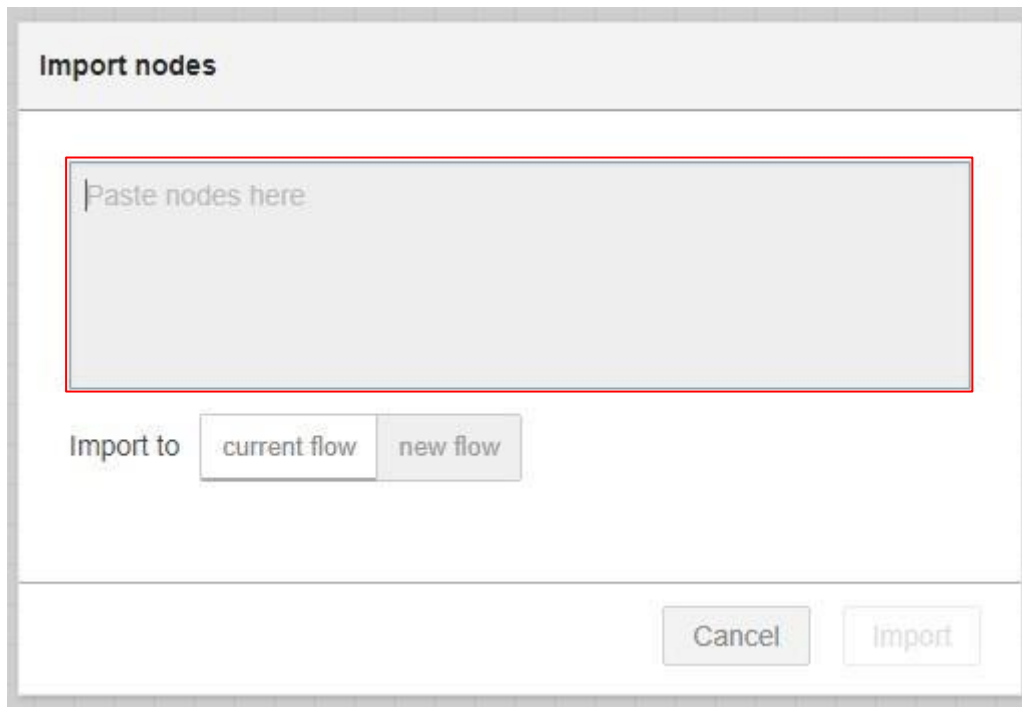
Two new nodes will appear in the list of available nodes:



Now, install the [node-red-dashboard](#) by following the exact same instructions.

Node-RED demonstration and its import

To understand the connection between UniPi and Node-RED, a demonstration was created. Its import is very easy – click on *Menu*, hover the cursor over *Import* and click on *Clipboard*.

A dialog box titled "Import nodes" with a large text area for pasting nodes, an "Import to" section with "current flow" and "new flow" buttons, and "Cancel" and "Import" buttons at the bottom.

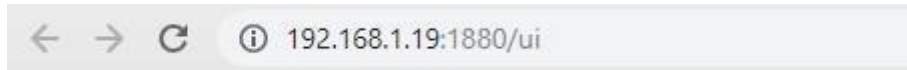
Import nodes

Paste nodes here

Import to

Now just copy and paste the content of the [flow pro navod verze 1.0.json](#) file added to this manual and click on *Import*.

With the import finished, open the user interface webpage by inserting the IP address and adding [/ui](#) as a suffix



When loaded, the page should look like the following screenshot:

Boiler controller

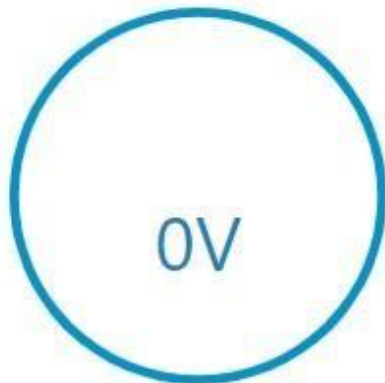
This thermostat is designed for controlling a boiler according to the set and the measured temperature. Measuring the temperature is done by a OneWire sensor and the actual temperature is compared with the set temperature by user. According to the result, the state of the relay 5 is set or reset.

Set temp  22°C

Act temp 20.15°C

Boiler controller **heat ON**

Analog input



Direct / Software switch

The input is a button. The first press of the button switches relay to the ON state, the second press of the button switches relay to the OFF state.

switch 1 

switch 2 

switch 3 

switch 4 

Analog control

Output 2.01 

Output 2.02 

Output 2.03 

Output 2.04 