Instructions for scaling of production

This document presents instructions for how to setup new production modules and expand the production process. At the time of writing of this document there were two modules functioning with a MES python file that was hardcoded for them. For future it is important that the MES file is expanded in a way that will accommodate the scaling of production without much change to the MES file and ThingsBoard widgets.

The document will cover the setup of new RPi computers, installation procedures for necessary libraries, Node Red and transfer of python files and Node Red flows.

From now on Node Red is NR and Things Board is TB.

Table of Contents

[1. RPi5 setup 3](#_Toc216788986)

[1.1. Connection to VNC 3](#_Toc216788987)

[1.2. Install Argon software 4](#_Toc216788988)

[1.3. Install NR 4](#_Toc216788989)

[2. Transfer of files 5](#_Toc216788990)

[2.1. Transfer of NR flows 5](#_Toc216788991)

[2.1.1. Procedure of transferring NR flows: 5](#_Toc216788992)

[2.1.2. Procedure of installing node packages 8](#_Toc216788993)

[2.2. Transfer of python files 10](#_Toc216788994)

[3. Creating a TB device 11](#_Toc216788995)

[3.1.1. Creating a TB device 11](#_Toc216788996)

[3.2. Configuring NR flow 12](#_Toc216788997)

[4. Hardware configuration 15](#_Toc216788998)

[4.1. Dobot magician 15](#_Toc216788999)

[4.2. Dobot peripheral devices 16](#_Toc216789000)

[4.2.1. Dobot conveyor 16](#_Toc216789001)

[4.2.2. Dobot vacuum pump 16](#_Toc216789002)

[4.3. Conveyor hardware configuration 17](#_Toc216789003)

[4.4. Camera configuration 19](#_Toc216789004)

[4.5. RPi hardware 19](#_Toc216789005)

Table represents what has already been implemented on what RPi. Expand the table as you include new devices.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RPi name | Current IP | Argon | NR | Flows | Python files | TB |
| RPi 1 | 192.168.9.114 |  |  |  | Kamera? |  |
| RPi 2 | 192.168.9.123 |  |  |  | Kamera? |  |
| RPi 3 | 192.168.9.153 |  |  |  | Kamera? |  |
| RPi 4 | 192.168.9.150 |  |  |  |  |  |
| RPi 5 | 192.168.9.148 |  |  |  |  |  |
| RPi 6 | 192.168.9.141 |  |  |  |  |  |

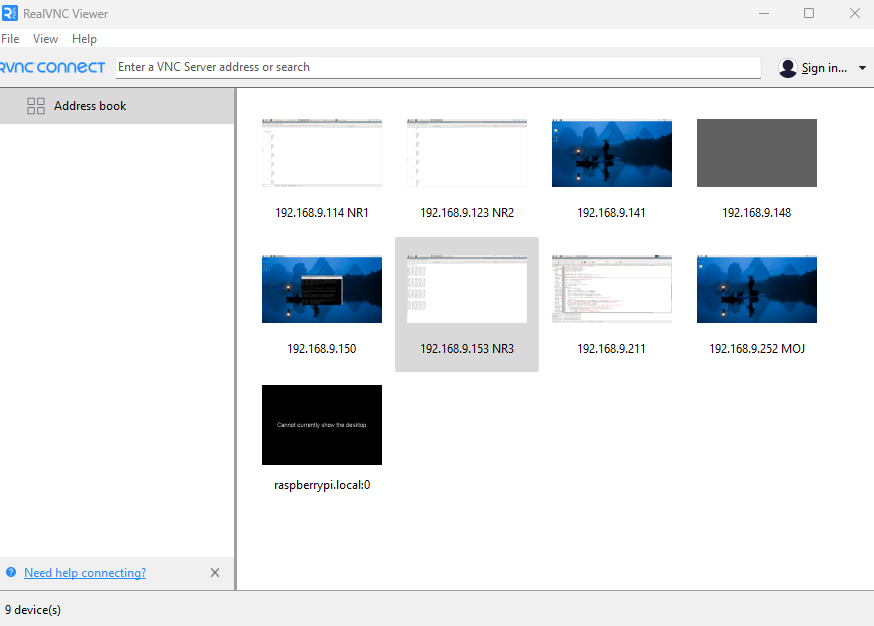
# RPi5 setup

The setup of RPi5 consists of several steps: connection to VNC, installation of Argon software (the housing of RPi5), installation of NodeRed (if not done so yet).

## Connection to VNC

At the beginning it may be possible that the RPi is not set to be accessed in “headless” mode so we will first have to set that up.

* Connect the RPi with microHDMI cable to a monitor, connect mouse and keyboard
* Connect to the industrial wifi in Demo center 2
  + SSID: DC-SF\_2
  + Password: deo7W^dA8u
* Activate VNC on RPi
  + *sudo raspi-config* -> Interface Options - > VNC -> “Yes” enable VNC server
  + take note of your IP address that is written on the RPi, it should be the same as is when you hover over the wifi sign on the top right of the screen (if it is not, use the IP from wifi sign and not the one written on the RPi)
* Install VNC on your PC (if not installed already) and open it
  + In top corner write the IP address of your RPi and connect to it



IP of your RPi

* + You will have to provide credentials of the RPi to connect to it
    - Username: pi
    - Password: rpi
  + Set “remember password” so future connections are more seamless
* After you connect to it and see the window on your PC disconnect microHDMI and you are running the RPi “headless”

## Install Argon software

If the software is installed, skip over this section. The software enables you to customize your RPi housing and most importantly fan speed. The instructions for this process are provided by the producer in a pdf in this folder (ARGON\_ONE\_V3\_product\_guide). The installation configures power button, fan control etc. (page 13)

You can configure the RPi housing with *argon-config* in terminal.

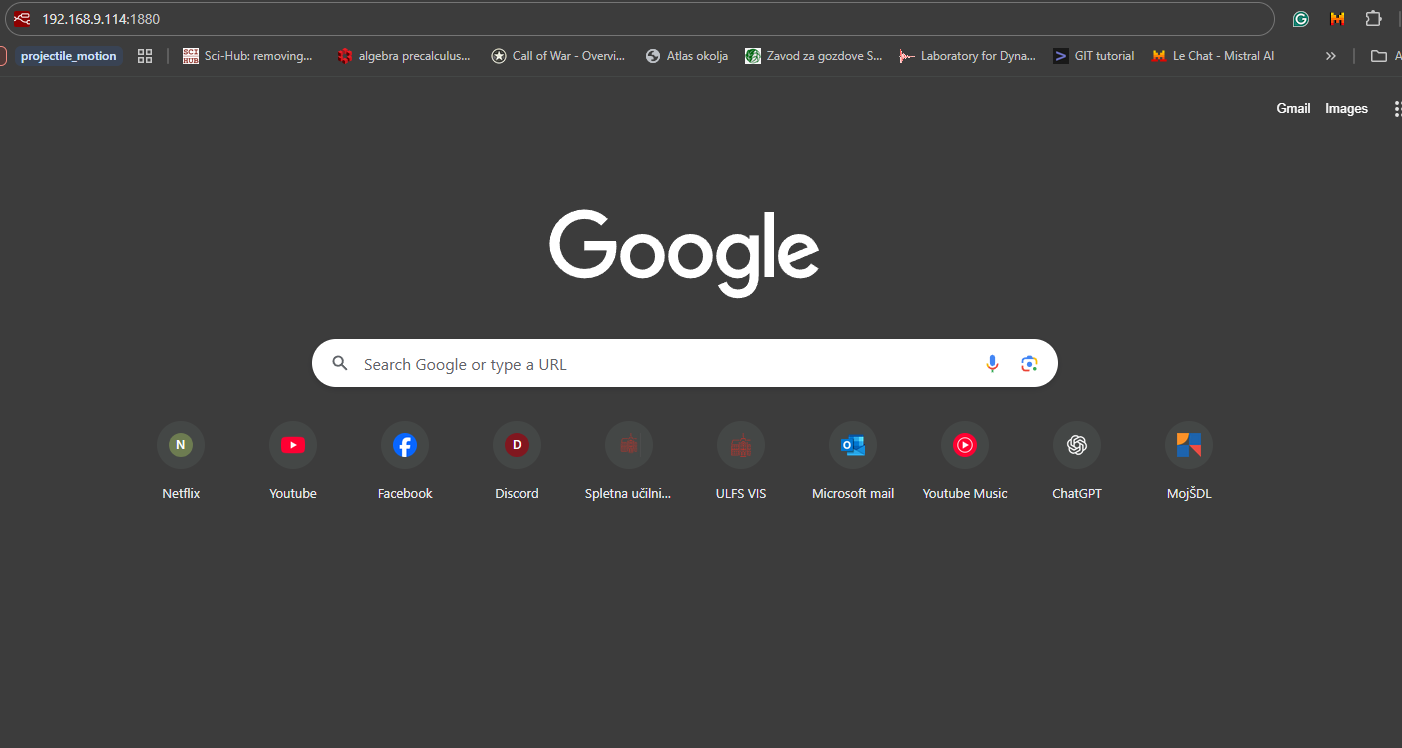
## Install NR

NR is a very important software which will be used to control the behaviour of production modules. Each RPi has its own NR installed and the flows which control the actions of peripherals that are connected to the RPi. More on that will be explained in later.

To install NR, you can follow the instructions below or just visit the website of NR.

Website: <https://nodered.org/docs/getting-started/raspberrypi>

* Copy this command in terminal: *bash <(curl -sL* [*https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered*](https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)*)*
* During the installation you will have to create a username and password.
  + Username: pi
  + Password: demo\_center\_2
  + In case there was a mistake on one of the first RPi-s you can login with password (demo\_center2) but the new password is as above
* After installation run the next command, which will enable NR to open upon reboot or startup: *sudo systemctl enable nodered.service.*
* After that you should be able to open RPi-s NR on any device within network by writing 192.168.9.xxx:1880 on URL bar.



IP of your RPi written in the URL bar

In case you have to change your password for your NR device, follow the instructions below.

* Connect to your device over RealVNC or ssh and open the terminal
* Write: *node-red admin hash-pw*
* When the terminal asks you for a password, write your new password. The terminal will then return a password hash which you have to copy to clipboard
* Then write: *nano ~/.node-red/settings.js*
* When the settings.js file opens find the authorisation structure and replace the current password hash with the one you copied to your clipboard
* Save and exit the file and reboot the device

# Transfer of files

After we have prepared our RPi for operation, we will have to transfer the flows and python files.

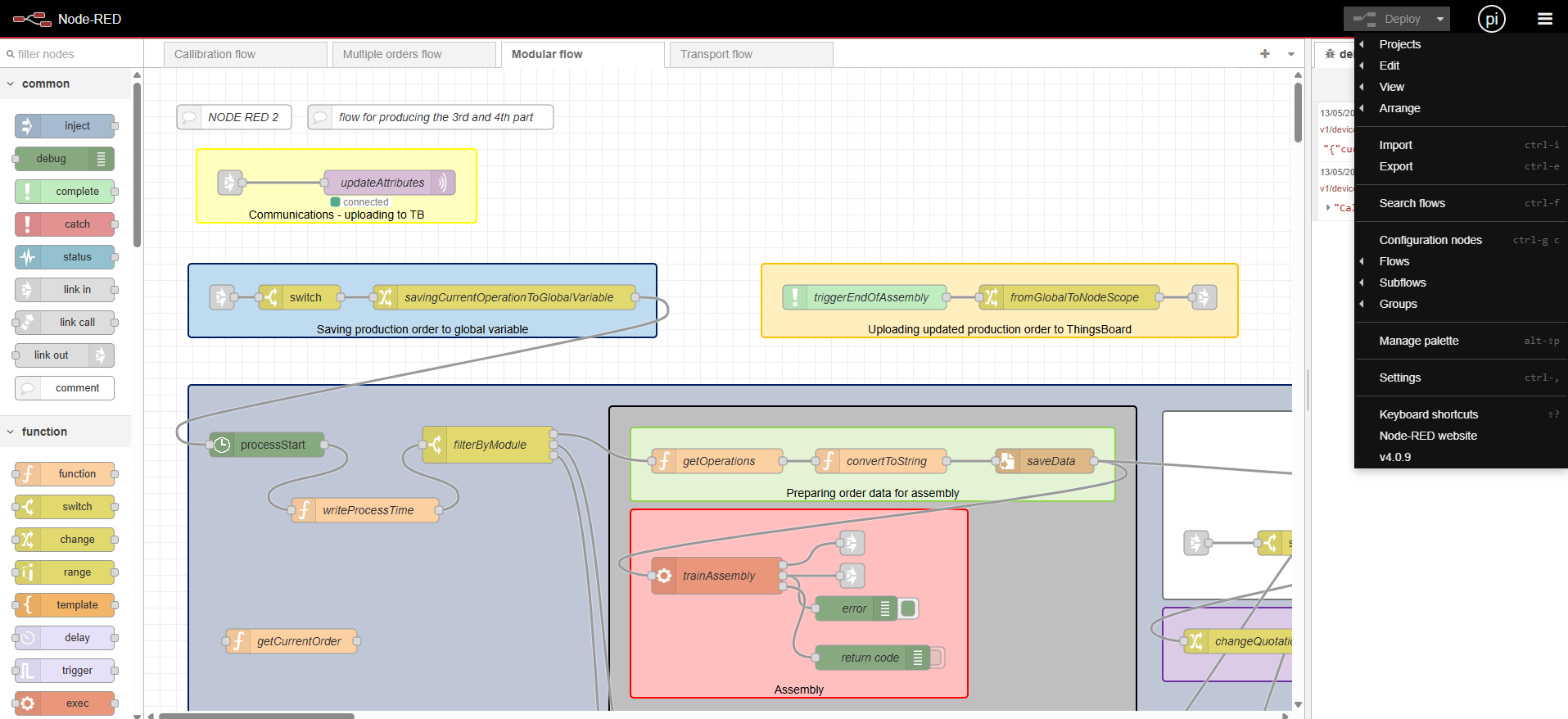
## Transfer of NR flows

NR flows are a way we control what will happen when we send data to RPi. It works without a dedicated python script and in a flow configuration which shows us exactly what is happening and where the data is going. Only after we are satisfied with the flow do we put some execute nodes to activate python files which control the peripheral devices. It is important to note that NR flow transports information one by one and that is why we have used JSON format where we packaged variety of data required by our python scripts.

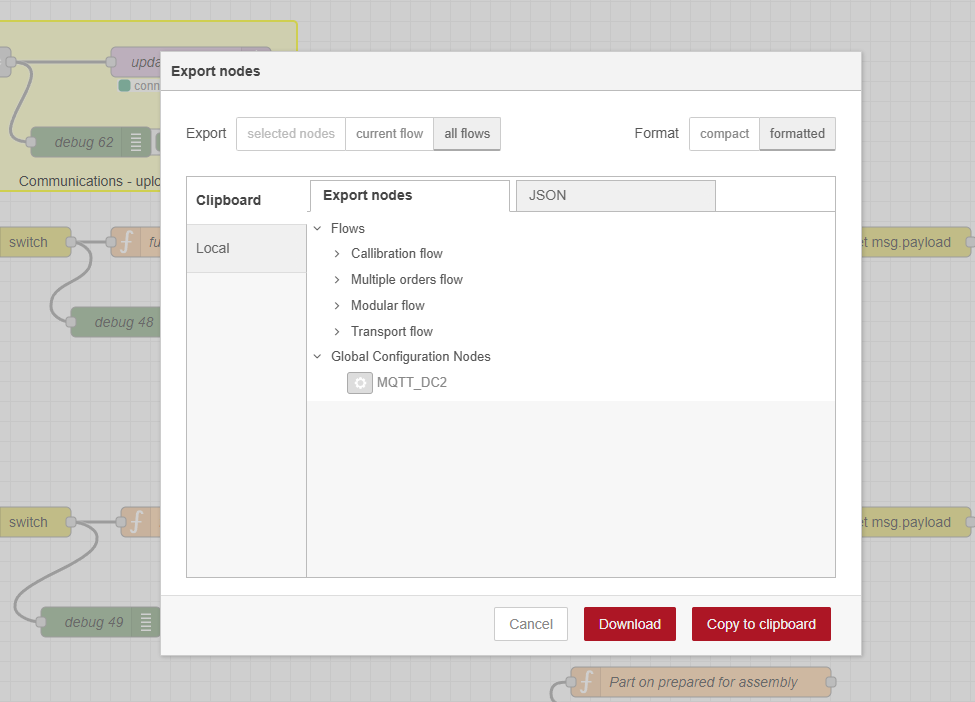
NR is accessed from any device on the network by the previously mentioned method. That means that transferring NR flows can be done on the same device where we edit them. When we export them, it is usually with JSON format. In that file is the data of all flows. The procedure is explained below.

### Procedure of transferring NR flows:

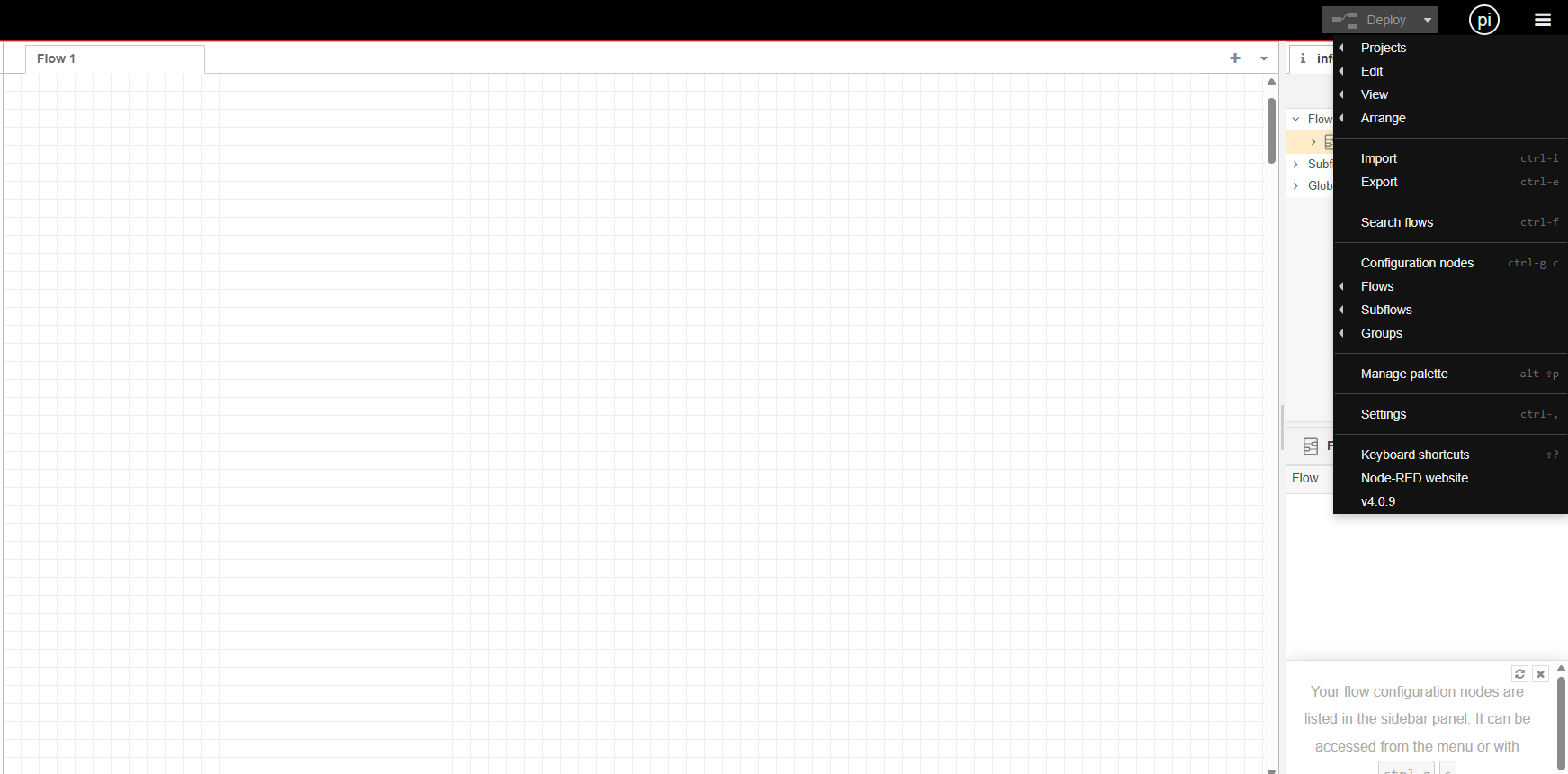
* Click on a three bars icon on top right and then click “Export”.



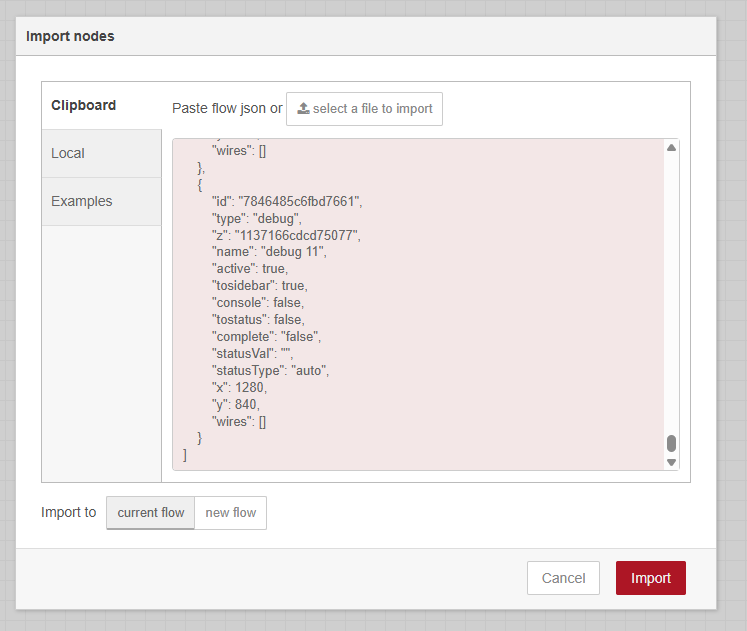
* Then a window will open where you select what you want to export. You can export just a node, several nodes, entire flow or all flows. For our goal of expanding our modular production we should select “all flows”. When you are satisfied with the selection you can see the JSON structure which contains the flows data and can manually copy it or you can just click the button “Copy to clipboard” on the bottom of the window.



* Now that we have our flows in our clipboard we go to the NR of the new production module. We again click on the top right menu and now select “Import”.

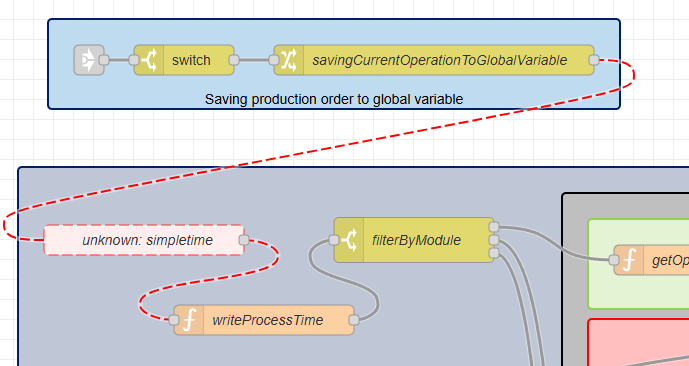


* When the import window opens, paste the entire JSON structure that we obtained before. You can then decide whether to import new flows to current or new flow but that is irrelevant for us since NR should now be empty. If we are copying only some features or flows then we must be careful where we import the data. Then click “Import” button and the flows should be imported in the new NR.



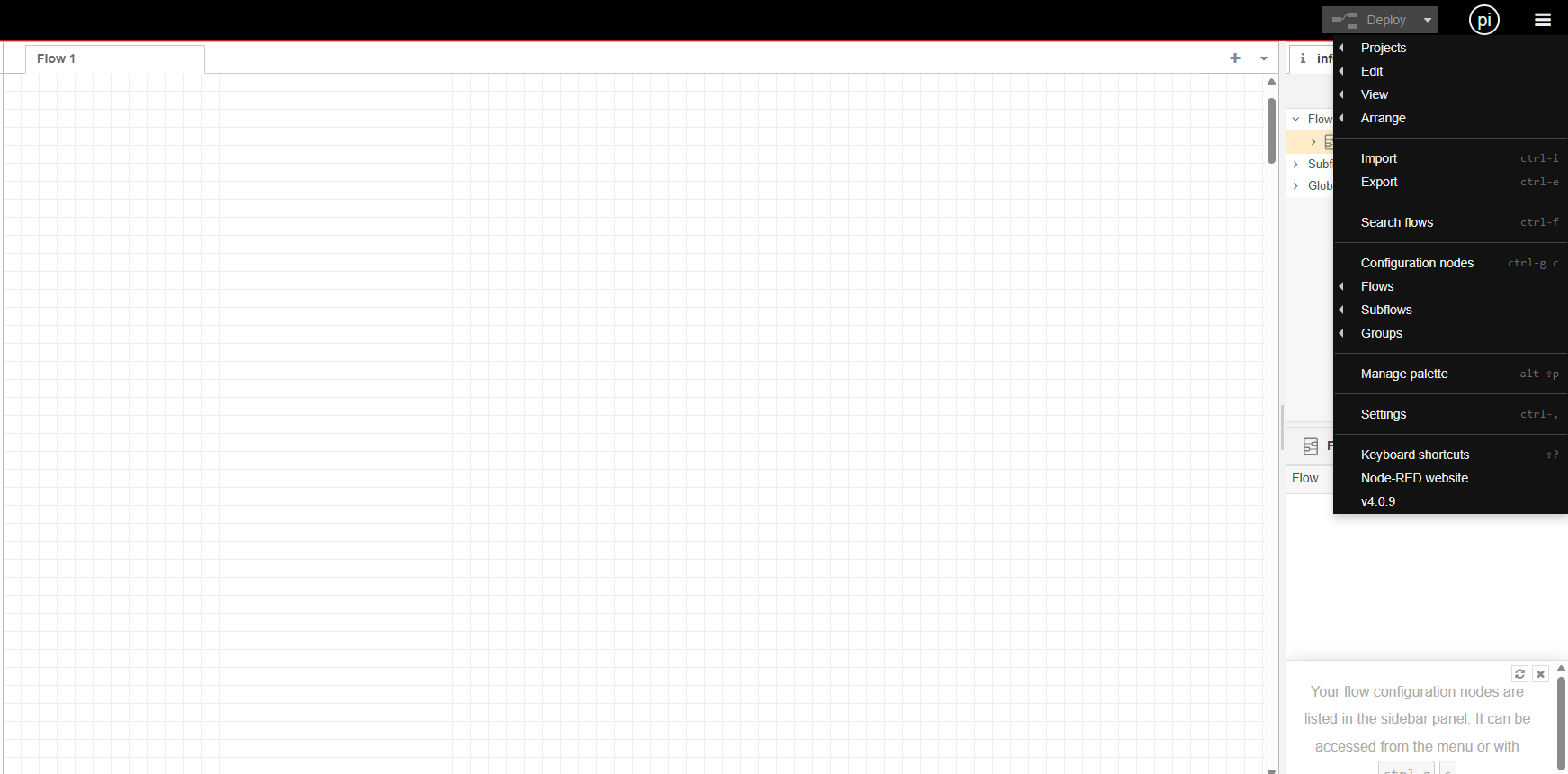
* After importing the flows, you should get an error stating that there are invalid nodes in your flows. That means you will have to install the node packages which will be explained below.

### Procedure of installing node packages

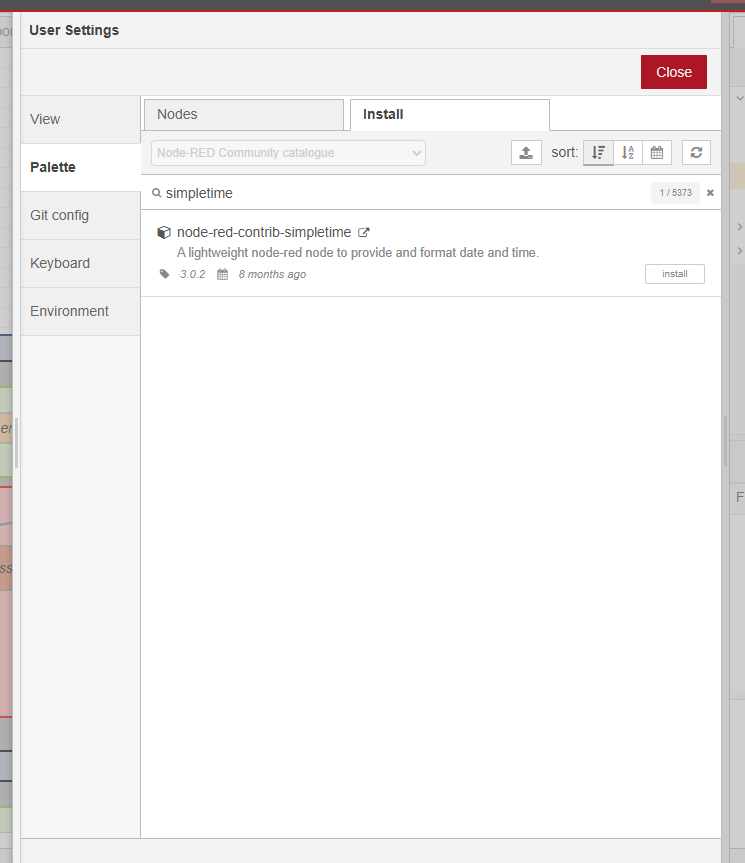
In order to have a functioning flow in our new device we also have to install new node packages. You can identify the “unknown” nodes by their special look, which is shown in the image below.

Note how most connections between nodes are grey and connections between red dashed node are also red dashed. That means that we have to install the package which enables the use of that node. The process of installing packages is explained below.

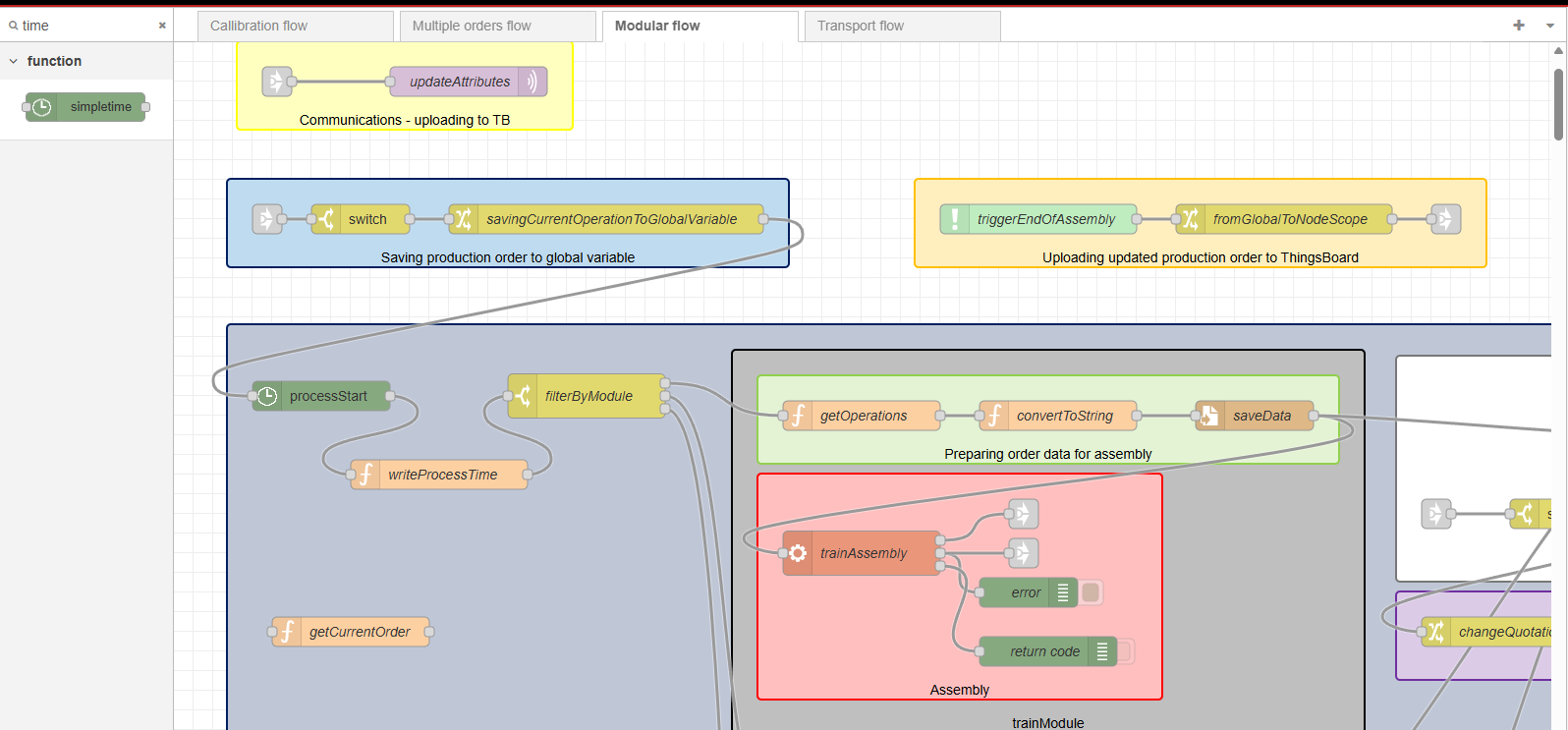
* Again click on top right three dashed menu button and then click on “Manage palette”.



* A window will open, which shows you already installed node packages under “Nodes” tab. On the next “Install” tab you can write in names of packages and then install them. Write in the name of your missing package, which should be “simpletime” and click install on it.



* You can check whether the install was successful and find it in future by searching for it in the search field on top left. All of the documentation for the nodes can be found on the next URL link and I advise that you read it before using the nodes: <https://flows.nodered.org/> (just time the name of the node “simpletime” in the search bar).



* List of nodes that should be installed manually as of 14.5.2025:
  + node-red-contrib-simpletime
  + node-red-contrib-loop

## Transfer of python files

We use python on several occasions in this project, mainly for MES and for controlling the peripherals on the production module. Following instructions are intended for transferring the python files on RPi devices for controlling the peripherals.

* You will have to find a USB drive and put it into an existing RPi on a working production module.
* The files you will be looking for are within a folder named FW\_DOBOT.



* Copy the files to the USB drive and eject it from the device.
* Mount the USB in the new device and copy the folder to the Desktop. All of the NR code uses that path so you will not have to configure anything when you connect the device to the system.
* Take note that these python files do not work with a venv. All of the libraries should be installed on global system with the usual procedure (pip install).

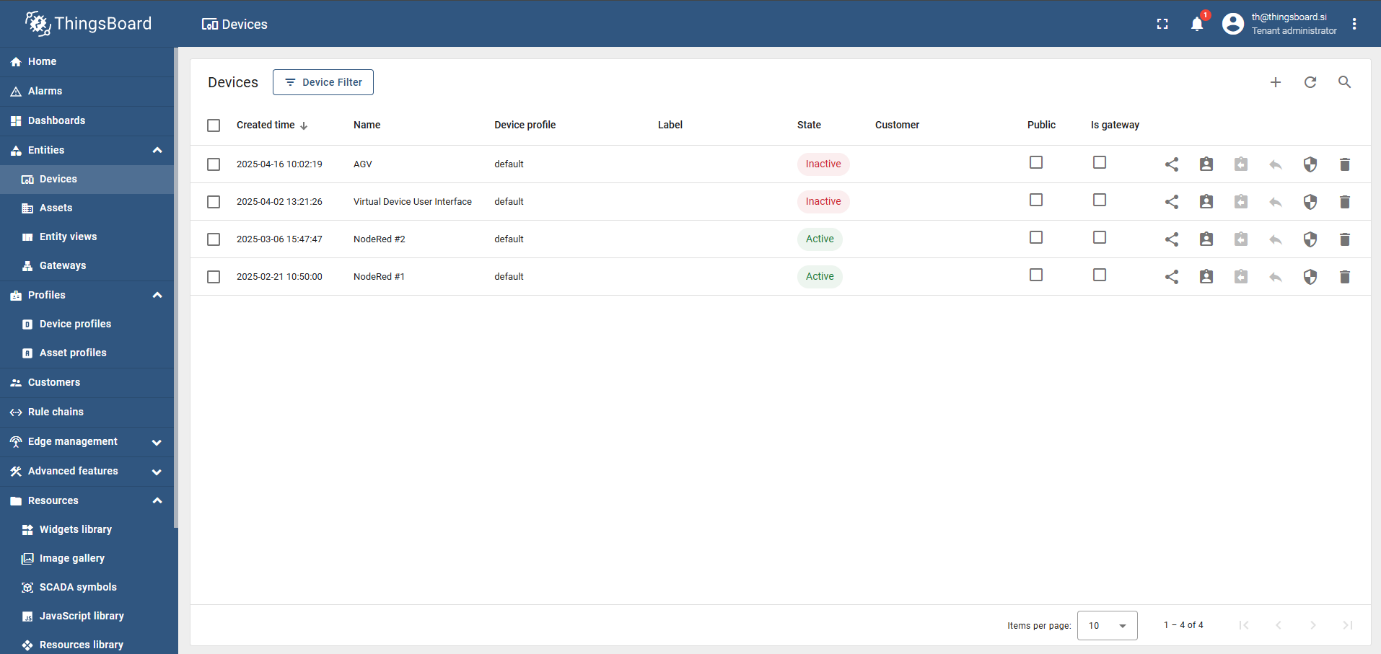
# Creating a TB device

Next step we need to do is to create a TB device, which will host the attributes which are needed for communication between MES and production module. TB device attributes are also used for visualization of module status on Virtual device. After we create the device, we will need to configure it for communication and update the credentials on new NR flow chart.

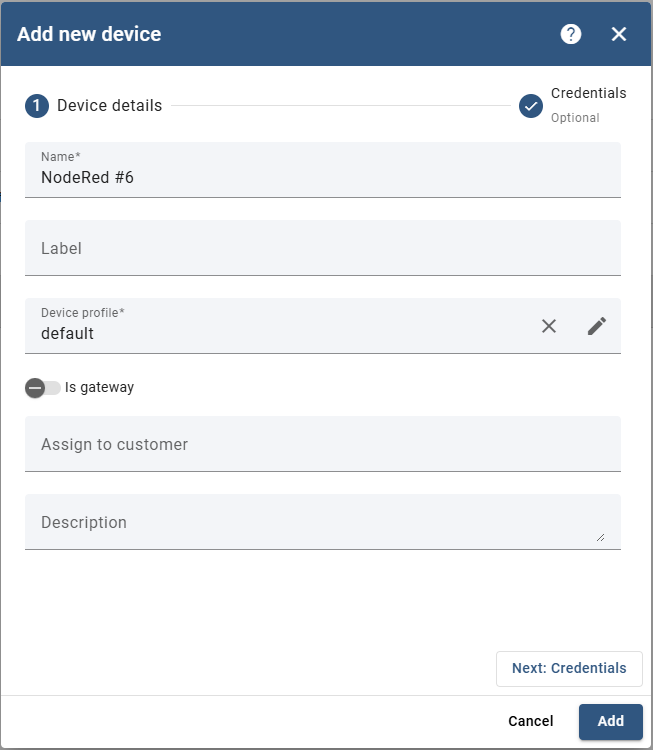
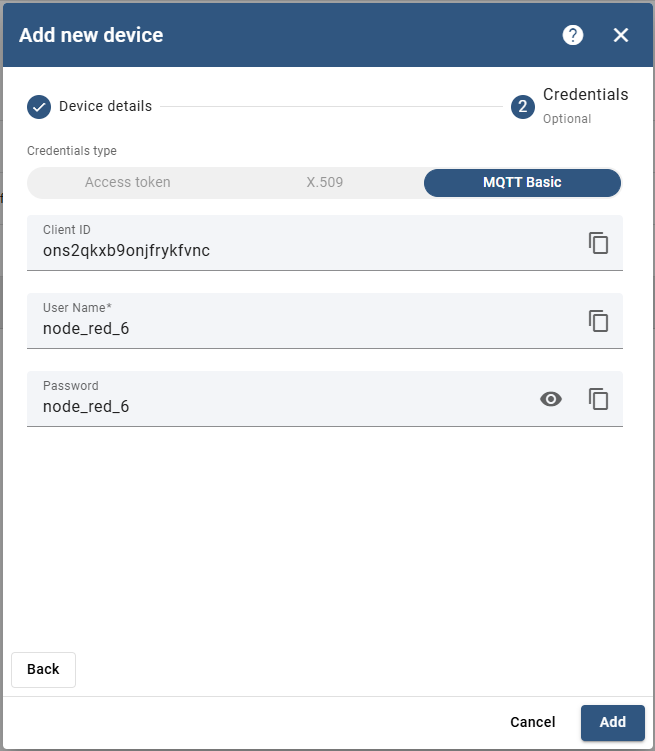
## Creating a TB device

In order to create a TB device, we need to have admin privileges and be logged in TB. In order to access TB, you will have to be connected to the local network and type the next IP in the URL bar in your search engine: 192.168.9.108:8080.

* When in TB navigate left side of the window and click “Entities” -> “Devices”. When the Devices window opens you will se all of the current devices. You can see if the device is connected to a production module in the “State” column, except for the Virtual Device which will always be inactive since it is being used as a user interface of our production line.



* Next step is creating the device by clicking on the “+” sign on top right and then clicking “Add new device”.
* New window will open and here you will have to write in Name and Credentials for the new device. Instead of the number 6 you will write the number of RPi that you are configuring.
  + “Device details” tab:
    - Name: NodeRed #6
    - Everything else should be unchanged.
  + “Credentials” tab:
    - Client ID: if the field is empty there is circular arrows sign that lets you generate a client ID. Use that and generate an ID.
    - User Name: node\_red\_6
    - Password: node\_red\_6



* After you have filled out the required boxes you should click “Add” and close the next window.

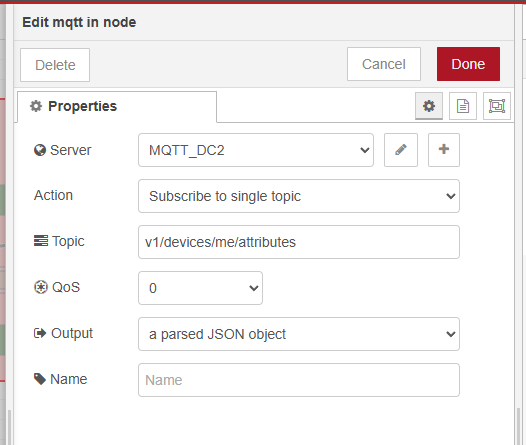
## Configuring NR flow

After configuring the device, you will have to reconfigure the communication node on your new NR flow.

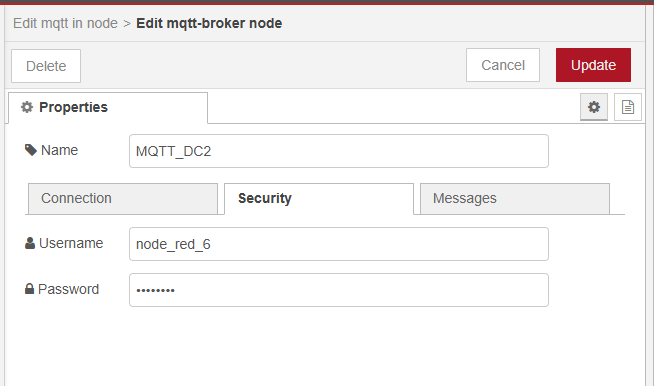
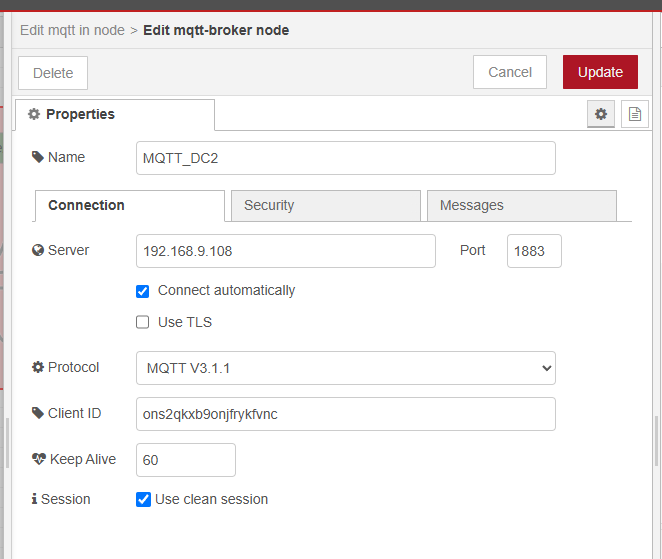
* When you navigate to the NR window go to the calibration flow. There will be a mqtt node that is responsible for listening to the TB attributes topic.



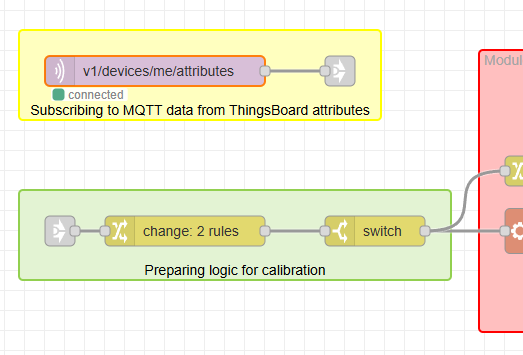
* Double click on the node to edit it. Next you will click on the pen icon to edit the current mqtt broker. The broker is already set up so you will just have to configure it for your exact TB device.



* When you open the broker editing window you will have to paste the “Client ID” that was generated earlier. After that click on the “Security” tab and type in previously set username and password for the device.



* After that click “Update” and the MQTT node should have a green light and be in a “connected” state. Now the device is set up and connected and any changes of attributes on the device will be picked up by NR and transmitted through all the flows.



# Hardware configuration

In this chapter all of the physical components will be presented and the way they integrate together.

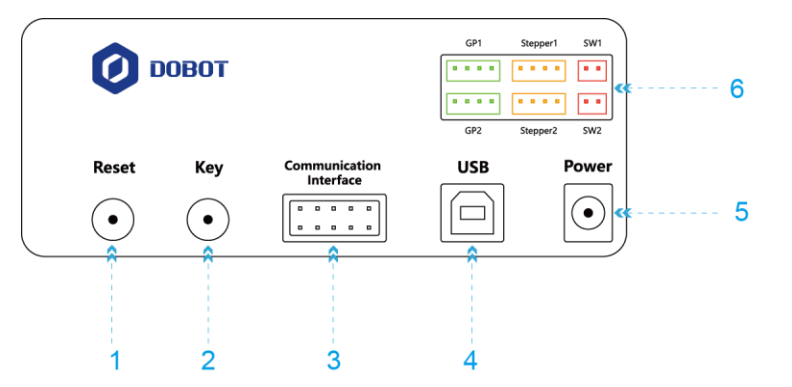
* Dobot Magician
* Dobot peripheral devices:
  + Dobot conveyor
  + Dobot vacuum pump
  + Dobot gripper/vacuum end effector
* For conveyor:
  + L-profile x2
  + Left and right pallet positioners
  + Left and right pallet receivers
  + IR sensor switch
* Camera
* RPi 5

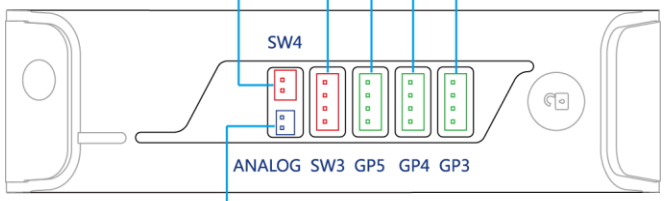
## Dobot magician

Dobot Magician (DM) is a 4-axis robot used in this project. The library used is pydobotplus which can be downloaded from github with the link below:

* <https://github.com/sammydick22/pydobotplus>

This library uses lists of coordinates, rotations and binary (0 and 1) for delay and end effector, to perform linear movements. So far only linear movements have been implemented. For robot jobs there are arrays of movement points which are saved in previously mentioned lists. For each robot job the python file will cycle through the array of lists (they contain the points) and perform them one by one.

DM also enables the use of peripheral devices which will be presented in the following sections. The mentioned library is supposed to control all of the dobot peripherals but we have had problems with IR sensor. End effectors and conveyor work without issues with the mentioned library. The peripheral devices are controlled with multiple interfaces on both the robot arm and robot base as are presented on images below.

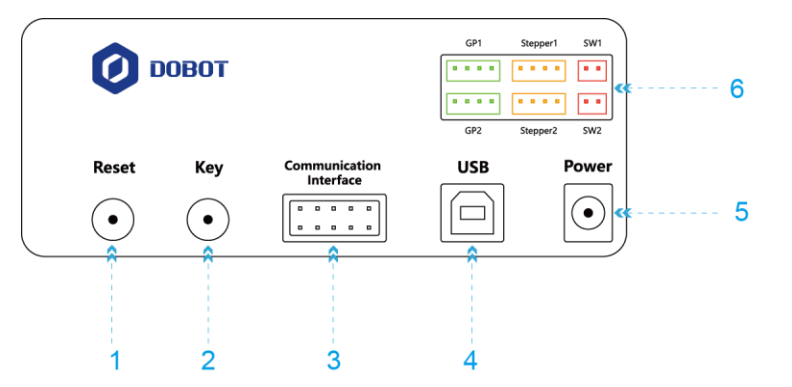


## Dobot peripheral devices

In this section dobot peripheral devices will be presented.

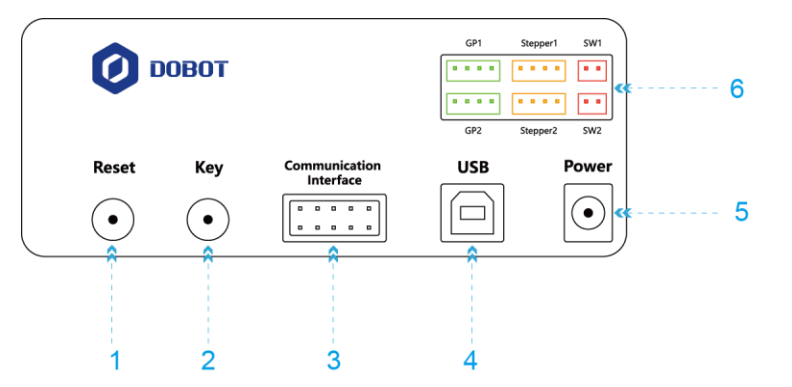
### Dobot conveyor

Dobot conveyor is connected to DM over base peripheral interface Stepper1 as shown below with red ellipse. DM controller controls the stepper motor of the conveyor and the library functions work well within this system. The functions used are called *conveyor\_belt()* and *\_set\_stepper\_motor()*.



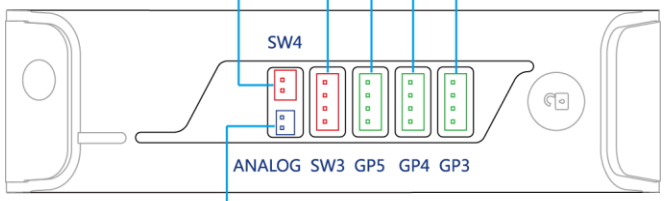
### Dobot vacuum pump

Dobot vacuum pump is used for vacuum end effector and is controlled with peripheral interfaces with green ellipses on image below. These are incorporated within the *suck()* function.



* + 1. Dobot gripper/vacuum end effector

In this project the vacuum end effector is used for picking up products. Peripheral interface used is shown below with red ellipse.



## Conveyor hardware configuration

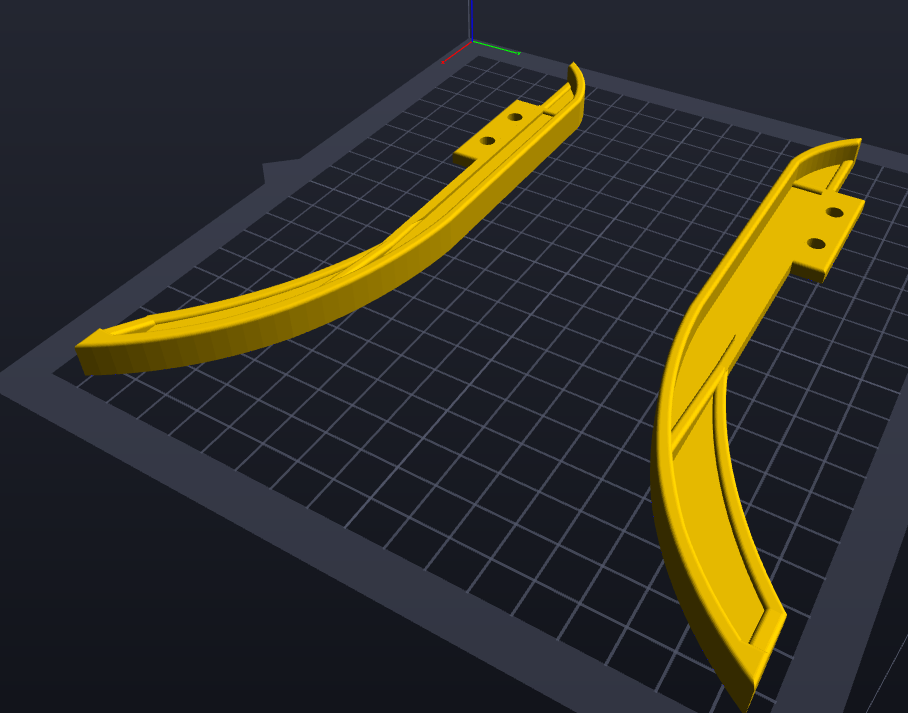
* + 1. L-profile x2

Dodaj sliko

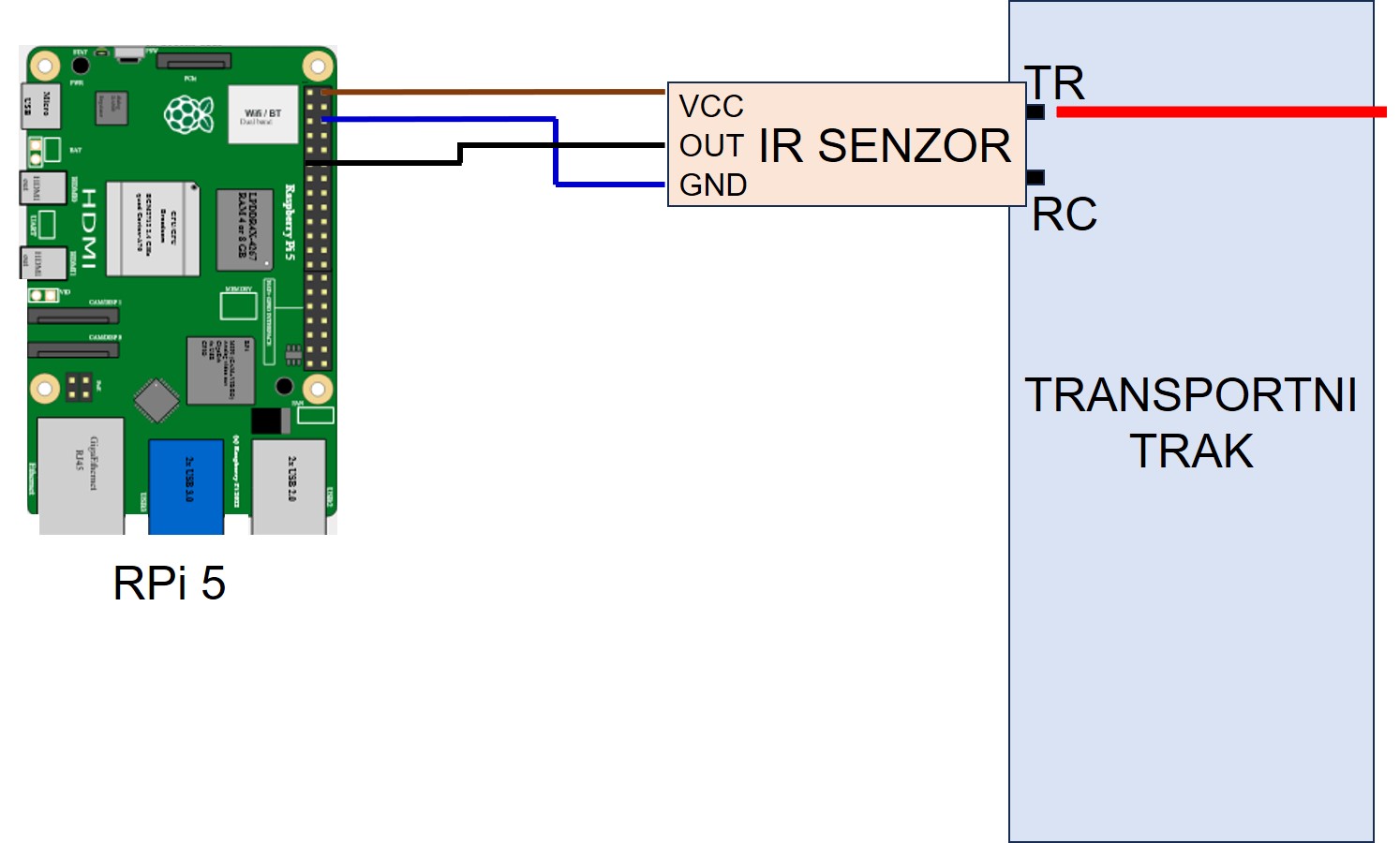
* + 1. Left and right pallet positioners

Dodaj sliko

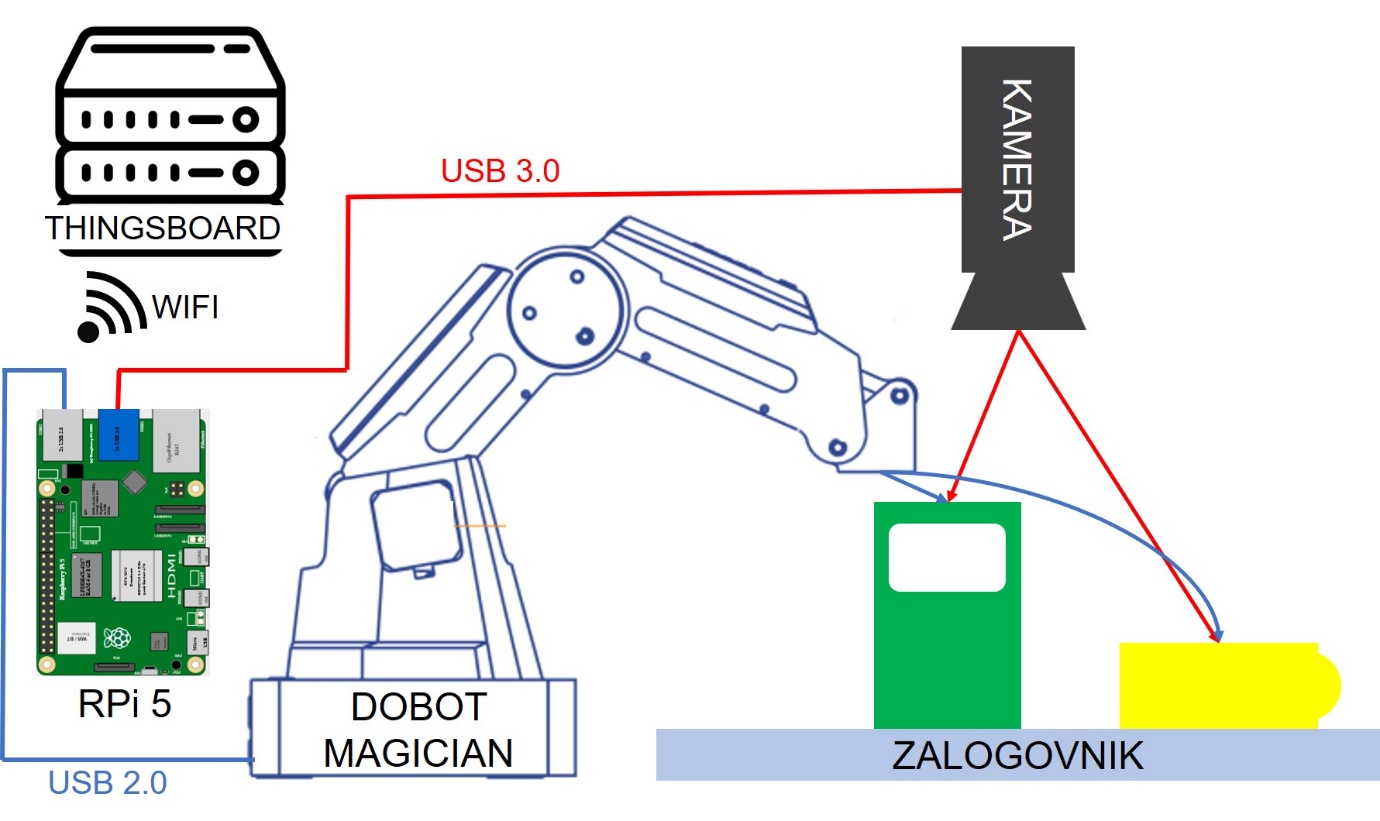
* + 1. Left and right pallet receivers



* + 1. IR sensor switch

Dodaj sliko

## Camera configuration



## RPi hardware



Tukaj bom pokazal kako je povezan IR senzor.

\*rabim še slike portov za robota in kablov od RPi-ja, vakumske črpalke\*

Napiši da je za NR in strojni vid pomembno, da se run\_save\_img.sh naredi executable z chmod a+x <file\_name>.sh