User & Technical reference manual

Contents

1 Introduction

NSPanel Manager is a custom software solution for the Sonoff NSPanel (not the NSPanel pro). The software is designed to be easy to use on a day to day basis and to easily manage multiple NSPanels around your home. The interface on the NSPanel itself has been designed to be intuitive to use for people of all ages and backgrounds.

All the NSPanel that are installed with the NSPanel Manager solution communicate back to a Docker container that is used to manage the panels, NSPanel Manager specific solutions and also all communication back and forth to/from Home Assisant and/or OpenHAB.

2 Quick start guide

2.1 Container setup

2.1.1 Windows

Important: There are currently problems running NSPanel Manager container on WSL2 as WSL2 does not handle networking properly. For more information, see the following issue on GitHub.

If you wish to run the NSPanel Manager container on Windows the current solution is to run it in a virtual machine. When you have your virtual machine up and running, you can follow the "Standalone docker container" installation below.

2.1.2 Standalone Docker container

Prerequisites:

- Repository cloned/downloaded.
- Working Docker installation.

At the moment, the container image is not available on any registry. This means that the image has to be built manually to start it. If running MacOS or Linux this is easy enough to do, perform the following to build and start the container:

- Open a terminal and navigate to the downloaded repository.
- Execute `cd docker` to navigate to the "docker"-directory.
- Run the script "docker-build_and_run.sh" to build the container & start it up using the default values. This can be done by executing `bash docker-build_and_run.sh`.

Important: By using the included script, the database with all settings and stored data will be stored within the downloaded directory. If you remove this directory, all data will be destroyed.

If you wish to manually build and run the container or change options or settings, see below for advanced setup.

2.1.3 Home Assistant add-on

As the repository is set to private currently, there is no way of adding it as a repository to Home Assistant. This can be worked around by doing the following:

- In Home Assistant, install the Samba or SSH add-on.
- Access the "add-ons"-directory in Home Assistant and create a new directory called "nspanelmanager".
- Copy all the files in the "docker"-directory from the GitHub repository to the newly created "nspanelmanager"-directory on the Home Assistant machine.
- In the Home Assistant web interface, navigate to Settings -> Add-ons -> Add-on store.
- In the upper right corner, press the three dots and choose "Check for updates".In a few seconds the "NSPanel Manager" add-on should show up under "Local add-ons".

Note: If the add-on has not appeared within 30 seconds, try refreshing the page and trying again.

Note: Some users have reported that Home Assistant has to be restarted for it to show the new add-on.

- Select the "NSPanel Manager" add-on and install it.
- Check that the add-on should start automatically.
- Start the add-on.

2.2 Container settings

The following has to be done in order to get a fully working container:

- Navigate to the web interface. If the port was not changed it is available at port **8000**.
- Enter MQTT server.

> **Note:** If you are running the MQTT server as an add-on to Home Assistant, enter the IP-address of your Home Assistant server.

- Enter MQTT port if changed from default **1883**.
- If authentication is used for MQTT, enter username and password.
- If running Home Assistant, enter address with http or https and port, Ex. *http://192.168.1.5:8123*. Also enter Home Assistant access token.

> **Info:** To get an access token in Home Assistant, navigate to Home Assistant and press your username in the bottom left. Scroll down and create a "Long-Lived Access Token".

> **Info:** If you are running the container as an Home Assistant add-on, the address and access token will already be set. Do not change these.

- If running OpenHAB, enter address with http or https and port, Ex. *http://192.168.1.5:8080*. Also enter OpenHAB access token.

> **Info:** To get an access token in OpenHAB, navigate to OpenHAB and press your username in the bottom left. Scroll down and create an "API Token".

- Save the new settings and continue to NSPanel flashing.

2.3 NSPanel flashing

Prerequisites:

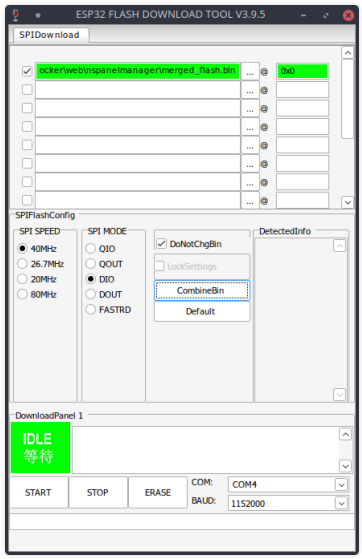
- Working TTL flasher for 3.3V.
- Working serial setup for your PC and known serial port (in Windows known as COM-port).

In order to connect to the NSPanel and be able to flash it, you must dismantle it. For a guide on how to dismantle and connect your serial flasher to the NSPAnel, refer to this guide from MarkWattTech.

2.3.1 Flashing with Espressif ESP32 DOWNLOADER TOOL (Windows only)

To flash the panel, perform the following:

- Download the tool from Espressif from here.
- Open the tool and choose to flash an ESP32 chip.
- Check one checkbox and select the "merged_flash.bin"-file in the "docker/web/nspanelmanager/"-directory.
- Enter "0x0" as the upload address.
- Connect your flasher to the NSPanel and press "START".



2.3.2 Flashing with ESPtool (all operating systems)

By installing esptool it is possible to upload the merged flash using the command line. Do the following:

- Open a terminal.
- Navigate to the "docker/web/nspanelmanager/"-directory.
- To determine if you have selected the right port, run `esptool.py flash_id --port <port>`. You will have to replace "<port>" with the actual port connected to the NSPanel. This will do a check and see if the tool can communicate with the NSPanel.
- Run `esptool.py --baud 921600 --port /dev/ttyUSB0 write_flash 0x0 merged_flash.bin`. You will have to replace "/dev/ttyUSB0" with the actual port connected to the NSPanel.

> **Info:** On Windows it might be just "esptool" without the ".py" at the end.

> **Info:** On Windows "/dev/ttyUSB0" will have to be replaced by something like "COM4". If using MacOS or Linux the port will be something similar to "/dev/ttyUSB0".

After the flashing is complete you can continue with NSPanel configuration.

2.4 NSPanel configuration

To configure the NSPanel, do the following:

- Power up the panel.

> **Warning:** If you have flashed multiple NSPanels, power them up one at a time as they will all have the same WiFi access point name.

> **Info:** The GUI file has not been flashed yet so there will not be any visible change on the NSPanel screen.

- Connect to the new WiFi network "NSPMPanel" when the panel has started. WiFi password is **password**.
- When connected to the new WiFi network, make sure your device does not disconnect because it detects no internet access. Then open a browser and navigate to "http://192.168.1.1".

> **Info:** There have been issues when using Android and the Chrome browser that it sometimes just shows a blank page. If this is the case, either use a different browser (Ex. Firefox) or another device with WiFi to access the web page.

- Enter a friendly name for the NSpanel.

Note: This name is only used to register the NSPanel the first time. After the panel has been registered the name can be changed from the manager web interface.

- Enter the IP-address for the manager docker container.
- Enter the port for the manager if changed during container setup.
- Enter WiFi name and password.
- Enter MQTT address and port.

Note: If you are running the MQTT server as an add-on to Home Assistant, enter the IP-address of your Home Assistant server.

- Press the "Save" button on the bottom of the page. The panel will reboot and try to connect to the WiFi network.

Info: If the panel fails to connect to the WiFi network for three minutes it will revert back and start the access point again. It will periodically scan for the configured WiFi and, if it detetacts that the configured WiFi has come back, it will reboot and connect to it.

- Connect to your WiFi again and go to the NSPanel Manager web interface. If all things are working and setup correctly the panel should show up in the list of panels on the first page.

Info: If this is a US NSPanel version then it has to be set in the panel settings. Press the name of the NSPanel in the list and check the "Is US panel"-checkbox.

- Flash the new GUI file to the panel by pressing the "Actions"-button on the right and select "Update screen".

Note: The flashing of the GUI file may be finicky and might require multiple tries before it succeeds. If it fails and reboots or you see a "System data error", just try again.

- Continue on to create new rooms and add entities to your configuration. Continue with the section on the web interface and how it work here.

3 NSPanel Manager web interface

The web interface is partitioned into 4 main sections.

1. The first page where panel status and management can be done.
2. The panel settings page.
3. The Room page.
4. The global settings.

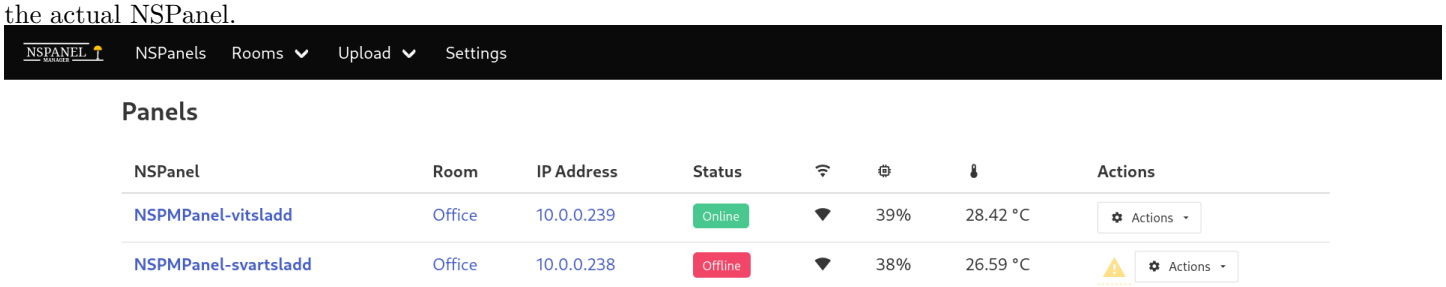
Below, each section of the web interface is described.

3.1 First page

The first page is used to get an overview of each registered NSPanel as well as perform actions on each panel. These actions are:

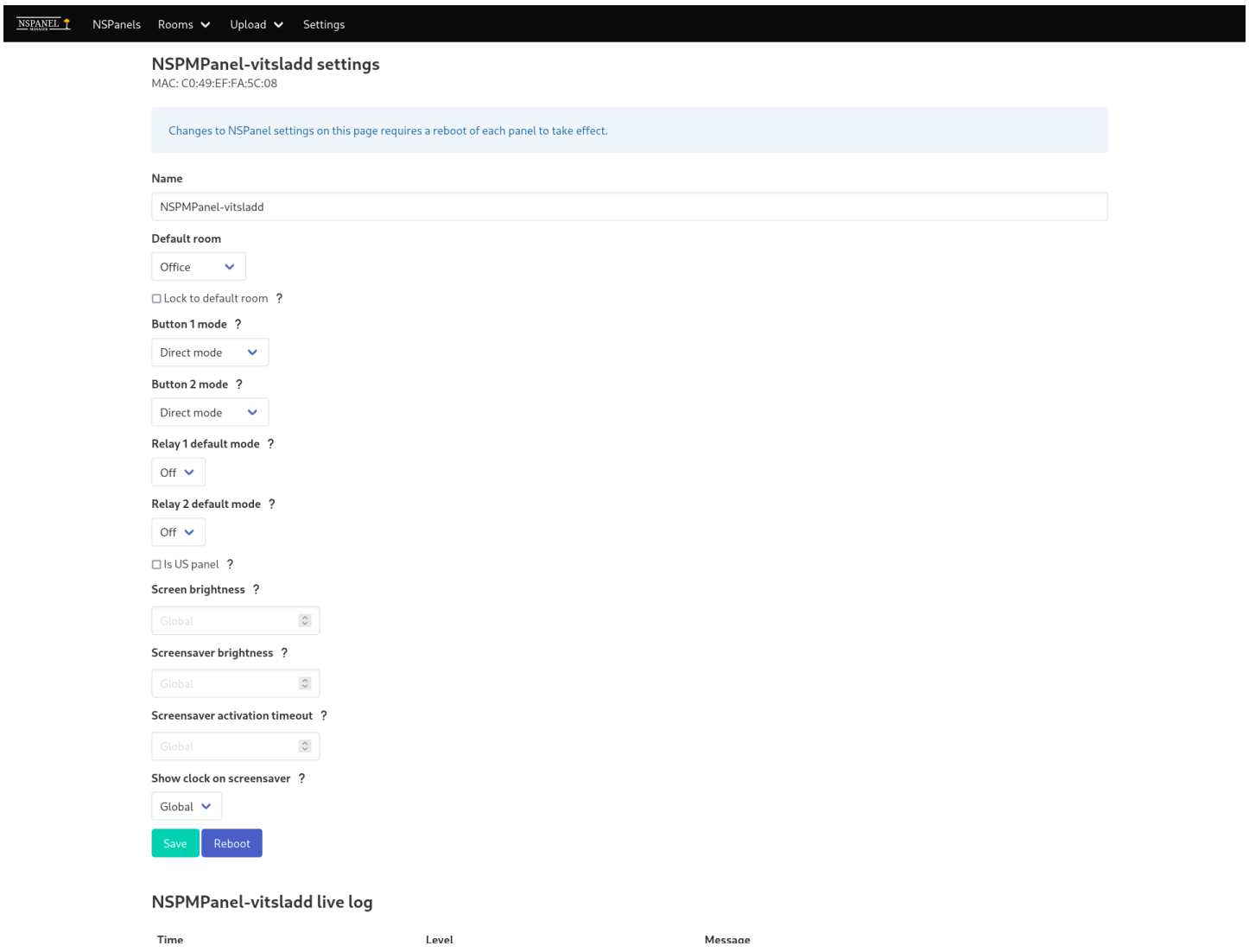
- Reboot.
- Firmware update.
- GUI update.
- Delete.

By pressing the name of the NSPanel you will get to the NSPanel settings page. By pressing the room name you will get to the relevant room settings page. By pressing the IP-address of the panel a new tab will open with the configuration page on the actual NSPanel.



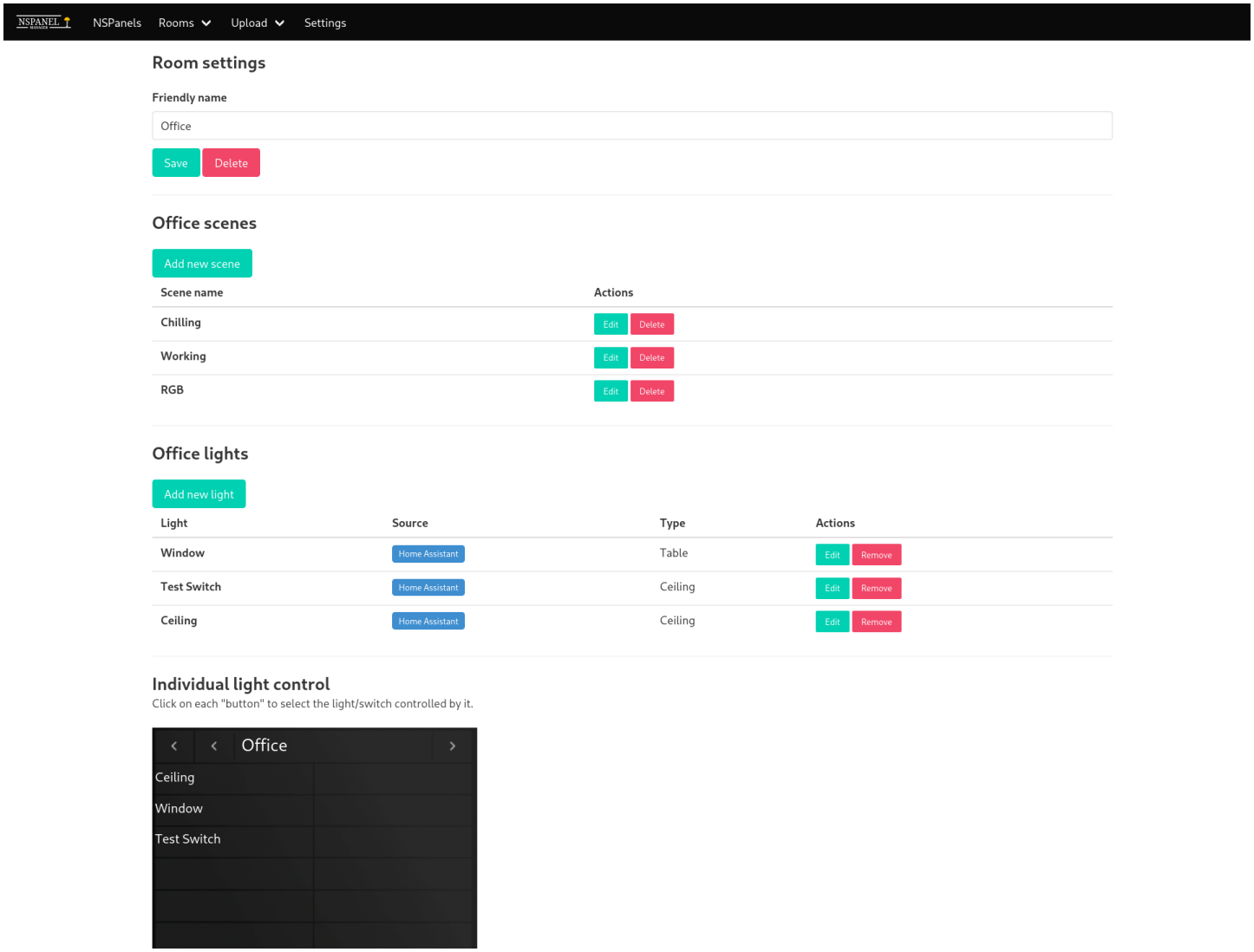
3.2 NSPanel page

On the NSPanel settings page, settings specific for one NSPanel can be made. This is things like name, default room and so on. There is also a live display of any log messages sent from the chosen NSPanel. This depends on the selected log level in the NSPanel configuration page which is available at the IP address of the NSPanel itself.



3.3 Room page

This section will describe how to manage rooms. Most of the configuration done with NSPanel Manager will be done in rooms, please read this chapter for a full understanding on how to work with rooms.



3.3.1 Scenes

At the moment, only NSPanel Manager scenes are available. They are easy enough to use. Simply create a scenes in the room page and they will be available in the "Scenes"-list for the room on all NSPanels.
To save a scene, on the NSPanel hold the save button for the scenes for 3 seconds to save all the states of lights **currently added to the room**.
To recall/activate a scene, on the NSPanel press the name of the scene and all the saved light states will be restored for that scene.

Note: If a light was added after a scene was saved, that light is not affected by that scenes until the scene is saved again.

3.3.2 Lights

To add a new light, simply press the "Add new light"-button. When doing so, a list of all lights and switches gathered from Home Assistant and OpenHAB will be shown. Simply search or scroll to find the desired light and press it.

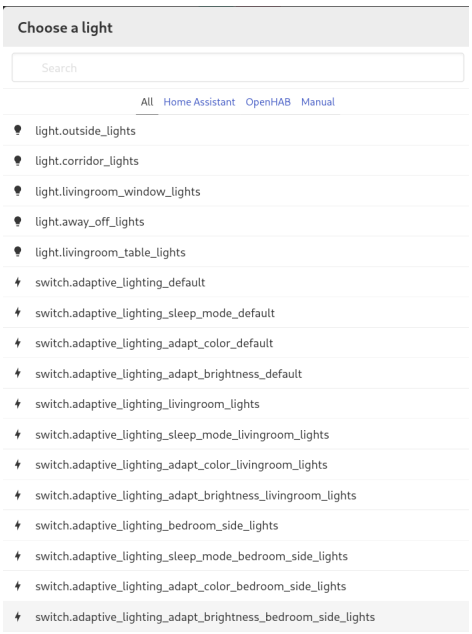
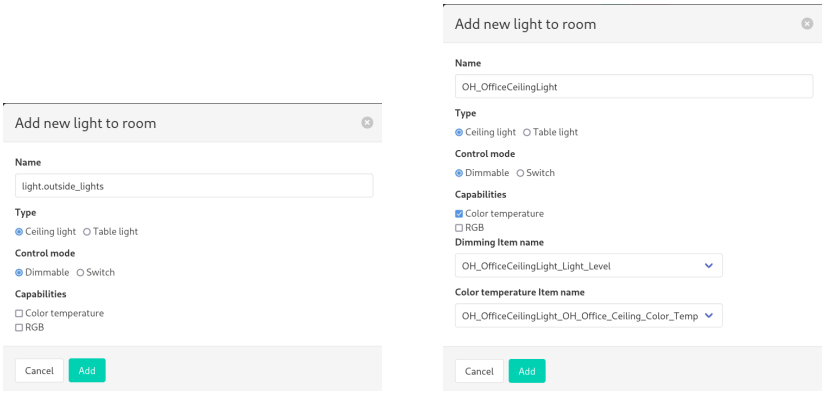


Figure 1: Adding a new light to a room

When done, a new screen will show up and depending on if the selected entity was from Home Assistant or OpenHAB.



(a) Home Assistant (b) OpenHAB

Figure 2: Add/Edit light

When adding a Home Assistant entity, simply set a friendly name for it, select the type (Ceiling or Table light), select if it's a switch or dimmable light and what other capabilities it has.
If you are adding an OpenHAB light or switch, things aren't as simple unfortunately. There is really no way around this but the user has to chose all the same settings as for Home Assistant but also has to select the appropriate OpenHAB items that corresponds to each capability of the light.

3.3.3 Individual light control

There is place for up to 12 lights (per room) to be controlled individually from the NSPanel. The image on the bottom shows a preview on how this might look. When a new light is added to the room it will automatically be assigned to the next free slot on the page. By pressing a slot with an assigned entity you can chose to assign a new entity (if any entity is unasigned) or "clear" the slot which will remove the light from the page but it will still be attached to the room.

Info: Each entity may only be assigned to one slot. If the list of entities is empty then all entities has been assigned a slot.

3.4 Global settings

These settings will apply to all NSPanels (if they do not have specific configurations), and the NSPanel Manager container. There are two things that need to be set in order to get up and running. Those are:

1. Connection details to the same MQTT server that were set in the NSPanel configuration.
2. Connection details to Home Assistant and/or OpenHAB.

Important: Failing to meet both requirements listed above will result in a non-working setup!

Info: If running the NSPanel Manager container as a Home Assistant add-on then the Home Assistant connection details will already be configured.

There are also other settings that might be worth taking a look at while here, such as global scenes that apply to all entities, showing a clock on the screensaver, how bright the NSPanels should be, the Min & Max of color temperature and so on. Go ahead and explore by yourself.

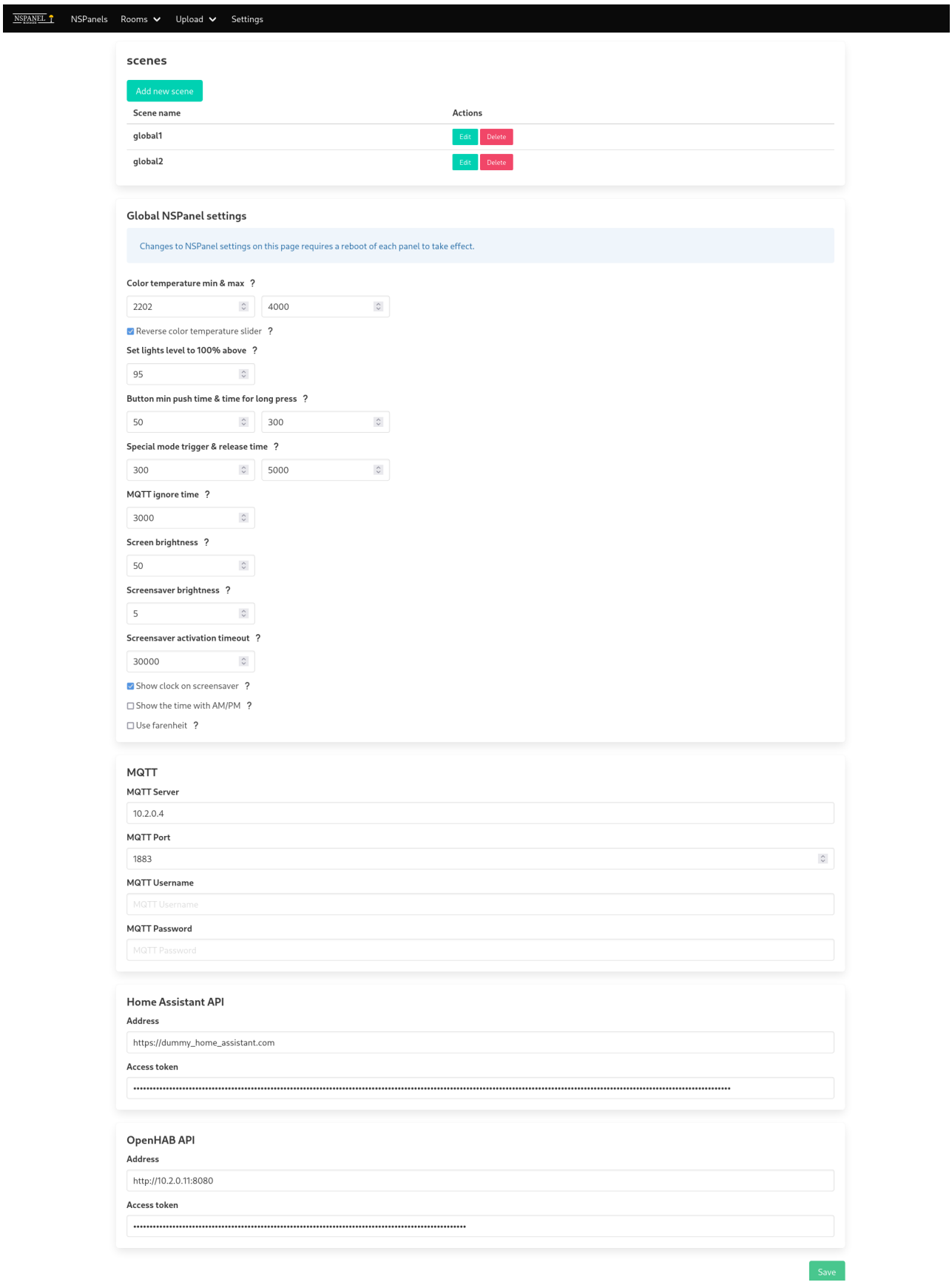


Figure 3: Global settings

4 Panel functions

4.1 Main page

4.1.1 Lights control logic

The NSPanel main page might be a bit have some behavior that seems odd at first but the logic of it will be described here. The first page will affect entities in the selected room (if in room-mode) or all the configured lights (if in "All lights"-mode). The two left button for ceiling and table lights will always behave the same. Pressing a button that is "off" will turn on all the lights of that type. Pressing a button that is "on" will turn off all lights of that type. The sliders will always display an average value of all entities that will be affected of changes. There is really a few different scenarios:

One or more lights on When changing the sliders, the changes will only be sent out to the lights currently on. If turning on a group of lights or individual lights they will be turned on to the current brightness of the slider. I.e. average dimming level in the room.

Info: If you wish for the light to always turn on with color temperature even though you turned it off from RGB, there is a setting in the global settings.

No lights on When changing the sliders, the changes will be sent out to all lights selected (depending on room or "all lights"-mode).

Lock mode You can lock which light to affect by pressing and holding either the ceiling or table-lights button. This will enter a special mode where changes to the sliders will only affect the selected type of lights. By pressing the same button again you can exit the "special mode". The "special mode" will also time out after a few seconds.

4.1.2 Scenes button (top left corner)

The little cute functional house in the top left corner is for entering the Scenes page. In NSPanel Manager there are both Room Scenes and Global Scenes. If you're in Room Lights mode (button in lower right corner) you will enter the Room Scenes page. If you're in All Lights mode you will enter the Global Scenes page. You'll also see that the house icon changes when toggling between Room Lights and All Lights mode. One yellow window and you will enter Room Scenes page when pressing it. All windows yellow and you will enter the Global Scenes page.

4.1.3 Swipe down

You can swipe downwards when on the first page to enter the Smart Home Control page. This page is a work in progress. Design and funcionality is not finished or decided yet.

4.2 Smart Home Control page

Accessed by swiping down on Main page. Work in progress. Design and funcionality is not finished or decided yet.

4.3 Scenes page

To enter Scenes page, press the little house in top left corner on Main page. Depending on if you're in Room Lights mode or All Lights mode you will enter Room Scenes page or Global Scenes page.
Scene names that show up here are the ones you have configured in the NSPanel Manager web interface. To store a scene simply hold the save button for three seconds and the current values of the lights in the room you are in or all lights if in All Lights mode will be saved. To activate a scene and send out those saved values you just press the scene name.

4.4 Room page

Enter Room page by pressing the room name on Main page. All devices configured for that room will show up here. To control a device individually press the device name.

4.5 Individual Lights page

All the capabilities of the chosen light will be shown on this page. If the light is RGB capable there will be an icon in the top right corner to toggle between Color Temperature mode and Color mode.´

5 Logs

While logs are normally sent over MQTT, any logs that are created before WiFi-connection are sent out on Serial. If you wish to see the logs going over MQTT, you can look at the topic `nspanel/<panel name>/log`. If you wish to look at the logs going over serial, you can use programs like Putty. Connect to the NSPanel with the serial programmer as usual but **dont't** connect IO0 to GND. In Putty enter your serial port in the "Serial line" box and choose baud 115200. You should then be able to connect by pressing the "Open"-button. Example:

Info: On Windows "/dev/ttyUSB0" will have to be replaced by something like "COM4". If using MacOS or Linux the port will be something similar to "/dev/ttyUSB0".

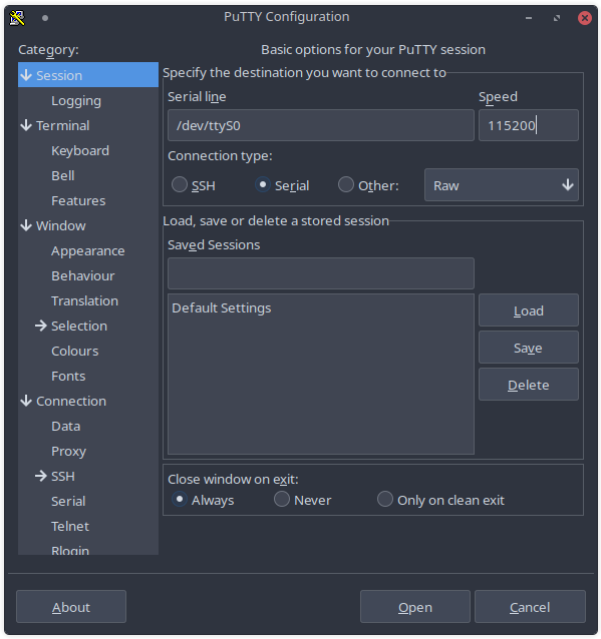


Figure 4: Connecting to Serial with Putty

6 Advanced setup

6.1 Manual Docker container setup

If you wish to manually build and setup the Docker container. To build the container, use the command `docker build -t nspanelmanager`. This will always be the same. To then start the container, the following command can be used:

```
docker run --name nspanelmanager -v /etc/timezone:/etc/timezone:ro -v \
"$(pwd)/web/nspanelmanager/db.sqlite3":"/usr/src/app/nspanelmanager/db.sqlite3" \
-d -p 8000:8000 -p 8001:8001 nspanelmanager
```

If you wish to change the timezone, there are two options. Either, do as the command above, pass in the local machine /etc/timezone. This might not always work though as you'r server might be set to Etc/UTC then you can set the environment variable like this, for example: `-E TZ=Europe/Stockholm` and remove the volume mapping for /etc/timezone.
If you wish to change where the database is stored, replace "$(pwd)/web/nspanelmanager/db.sqlite3" with where you wish your database to be stored on the local machine.

7 Functional information

7.1 Software components

There are really three software components written for the NSPanel Manager project. These are described as below:

- **Web interface:** The web interface that you interact with is built on top of the Django framework. This software gives the user an interface to interact and configure the project with. This software also manages the database with settings.
- **MQTT Manager:** There is a second software running in the background on the NSPanel Manager container that hosts the web interface. This component is named "MQTTManager". The MQTTManager handles all things with MQTT. It loads the config from the web interface via the API and then processes all commands from panels, state updates from Home Assistant and OpenHAB and so on. It's basically the glue that makes the panel's actions affect Home Assistant and OpenHAB. The MQTTManager is also the software that send state updates from Home Assistant and OpenHAB to the panels when changes occur.
- **The NSPanel firmware:** The firmware written for the NSPanel has been specifically designed to be as response and easy to use as possible. The firmware handles all communication with the TFT (Nextion) display and with MQTTManager via MQTT.

7.2 Data flow

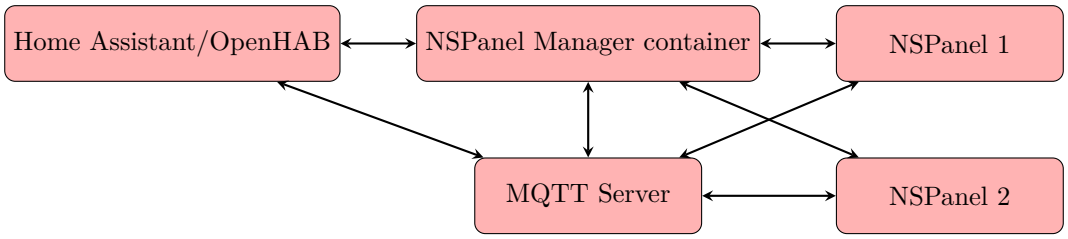


Figure 5: NSPanel Manager data flow

The data flow within NSPanel Manager might look intimidating but it's not that bad. Below is an explination of all the arrows above.

7.2.1 Home Assistant and/or OpenHAB to/from NSPanel Manager container

There is two types of trafic flowing between these nodes:

- **Websocket:** A websocket connection is setup in order for the NSPanel Manager container to receive entity updates from Home Assistant and/or OpenHAB but also to sent entity commands (ex. turn light X on typ 20%). A websocket is used to speed up the communication and also to not have to poll the home automation software for information.
- **HTTP GET API:** The usual HTTP GET API is also used. This is used when adding entities to a room, as an example. When pressing the "Add new light" button, the NSPanel Manager container will make an HTTP GET request to gather all available entities and then send them back to the client (browser) so that the user may choose what entitiy to add to the room.

7.2.2 NSPanel Manager container to/from MQTT

MQTT is used to send updated entity states received from the home automation software out to all NSPanels and also receive states and commands from NSPanels.

7.2.3 Home Assistant and/or OpenHAB to/from MQTT

Home Assistant and OpenHAB can leverage the MQTT integration through "Home Assistant MQTT Auto-discovery" (which OpenHAB can also use) to auto-discover NSPanels and automatically register entities for panel temperature reading, panel relays, screen state and so on.

7.2.4 NSPanel Manager container to/from NSPanels

The configuration of lights, scenes and so on does not reside on each panel. The panel only has localy the bare minimum configuration for setup. When the panel starts and has connected to WiFi it will do a HTTP GET request to the NSPanel Manager container in order to receive all configuration of entities, screen brightness and really, all settings available in the NSPanel Manager web interface.

7.2.5 MQTT to/from NSPanels

Each NSPanel send states (ex. temperature) and commands (ex. turning on a light) over MQTT for the NSPanel Manager container to pickup. The panel also received commands, ex. turn on relay 1, turn on screen and os on.

7.3 MQTT Topics

Below table is a description of all MQTT topics that might be of use by a user. Replace <panel_name> with the friendly name of your NSPanel:

Topic	**Payload**	**Description**
nspanel/<panel_name>/screen_cmd	1 or 0	Send a 1 or 0 to turn on/off the display.
nspanel/<panel_name>/screen_state	1 or 0	Current state of the screen.
nspanel/<panel_name>/r1_cmd	1 or 0	Send a 1 or 0 to turn on/off relay 1.
nspanel/<panel_name>/r1_state	1 or 0	The current state of relay 1.
nspanel/<panel_name>/r2_cmd	1 or 0	Send a 1 or 0 to turn on/off relay 2.
nspanel/<panel_name>/r2_state	1 or 0	The current state of relay 2.
nspanel/<panel_name>/temperature_state	Current temperature	The current temperature reading.
nspanel/<panel_name>/log	Log message	The panel will send live logs on this topic.

There are more topics that are used internally, these are:

Topic	**Payload**	**Description**
nspanel/entities/<type>/<id>/state_<attribute>	The value of the attribute	An update of entity state value sent out by MQTTManager. Example: nspanel/entities/light/42/state_kelvin
nspanel/status/time	Time	Current time sent by MQTTManager.
nspanel/<panel_name>/status_report	JSON	JSON payload with current state of the panel.
nspanel/mqttmanager/command	JSON	JSON payload from panel with a command for MQTTManager to perform.