3t - Smart Power Manager

Smart power manager is the solution you need to control your raspberry on remote, here is a little list of all functionalities available :

- Manage all the devices connected to the Raspberry
- Create scenarios for your devices, for example: if you want to turn off a screen then two seconds after turning it on, you can easily create this scenario and run it in two click
- See the Ethernet Status of your device connected in real-time
- Easy installation with a simple directory to copy and paste where you want on the Raspberry

1 - Introduction

You must install "Raspberry pi OS" or "Raspberry pi OS Lite" on your Raspberry and then enable ssh on RPi.

Be sure that you provide a sufficiently powerful power source to the raspberry, otherwise, you may get untimely crashes!

Configure also the IP address of Rpi. Example :

```
sudo nano /etc/dhcpcd.conf

// uncomment and modify these lines :

# Example static IP configuration:
interface eth0
static ip_address=192.168.1.55/24
static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.1.1
static domain_name_servers=192.168.1.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

Before starting the installation, you must know that you can use the SmartPowerManager in two ways :

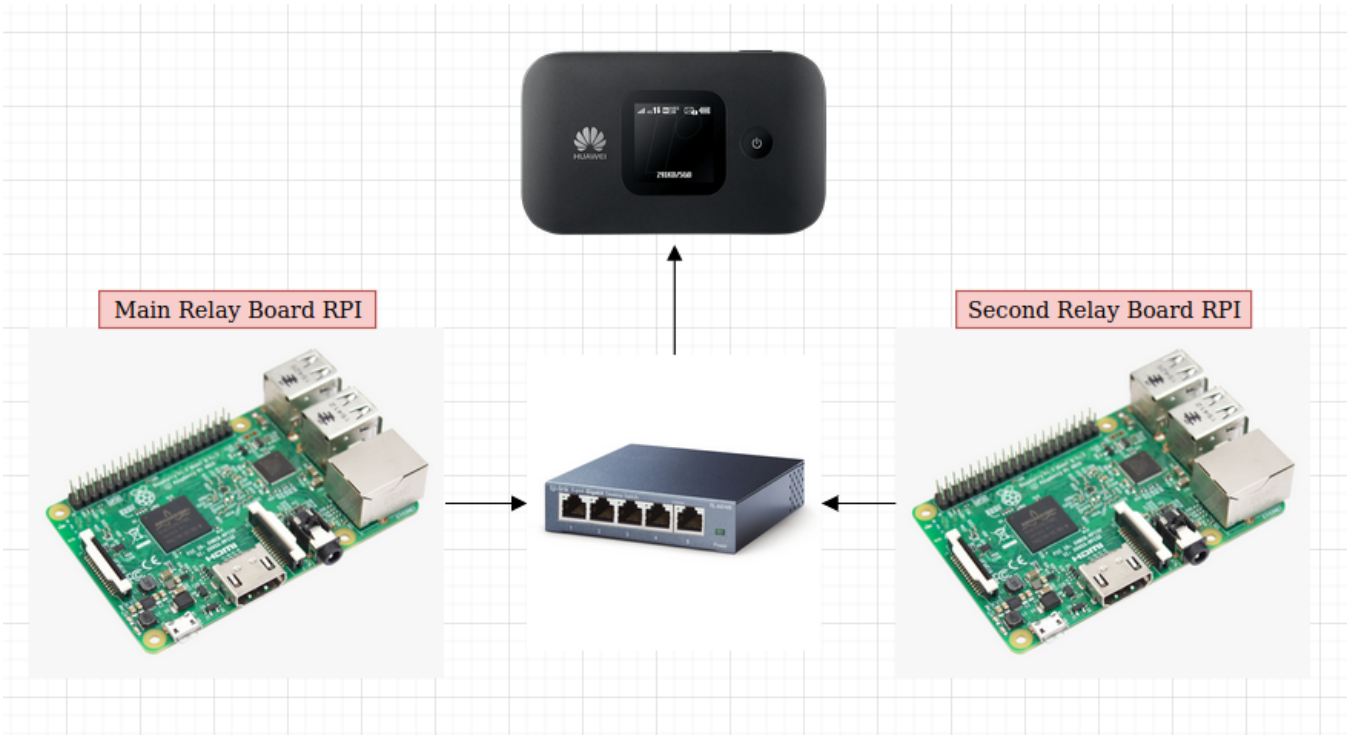
-With 1 Raspberry (28 GPIO pin controllable): you will have to run only deployOnMainRPI.sh

-With 2 Raspberry (2 * 28 GPIO pin controllable): In that case, you will have to run deployOnMainRPI.sh and deployOnSecondRPI.sh

Before starting the installation, please read all the documentation!

2 - Connection diagram before software installation

First, connect the two RPI like this (RPI must be connected to the internet for the installation, it will **not work with Faurecia internet**, please use **4g router** instead)



3 - Install SmartPowerManager on the Main RPI

1 - Installation of the Smart Power Manager for the main Relay Board

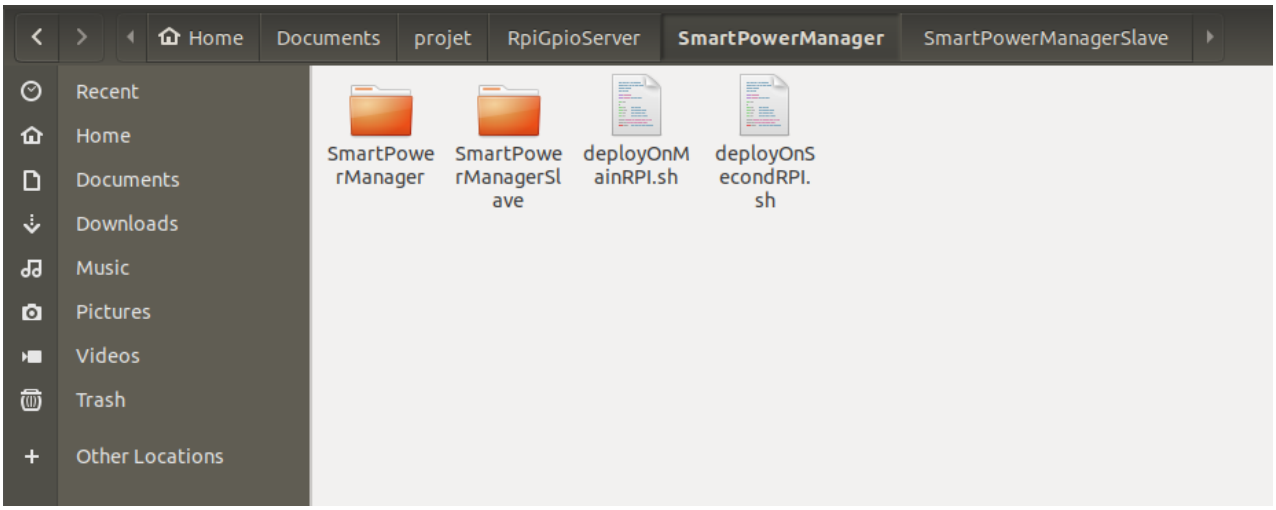
First clone the repository

```
git clone "oem/NIS8100_POC/RpiGpioServer"
```

Then change of branch

```
git checkout develop
```

Then open the folder downloaded "SmartPowerManager" :

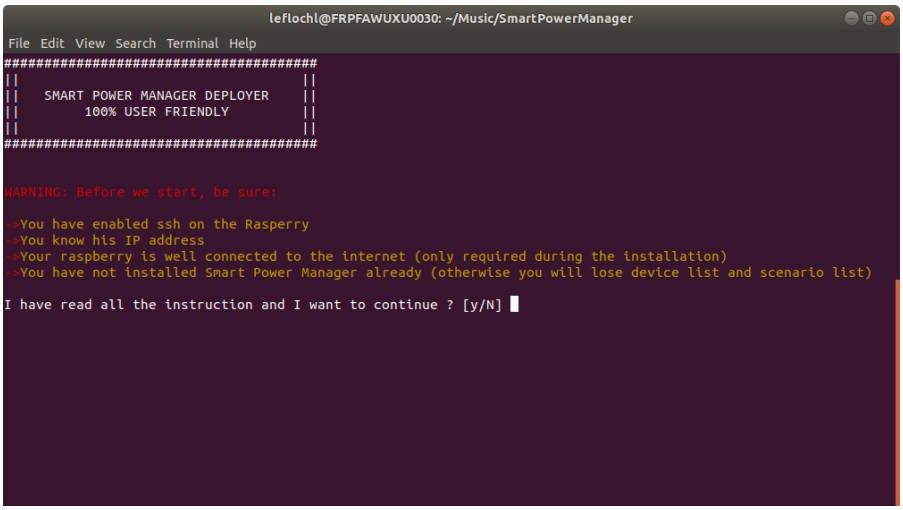


Launch deployOnMainRPI.sh by doing the following command :

Open a terminal in the directory opened, then type :

```
bash deployOnMainRPI.sh
```

You will have :



Now follow the instruction displayed then come back (note that the default password on a Raspberry pi is "raspberry")

2 - Verify if the deployOnMainRPI.sh has done the job

To do that: Connect to Raspberry in ssh typing in a cmd (replace by your raspberry Ip) :

```
ssh pi@[main rpi board controller ip adress]
```

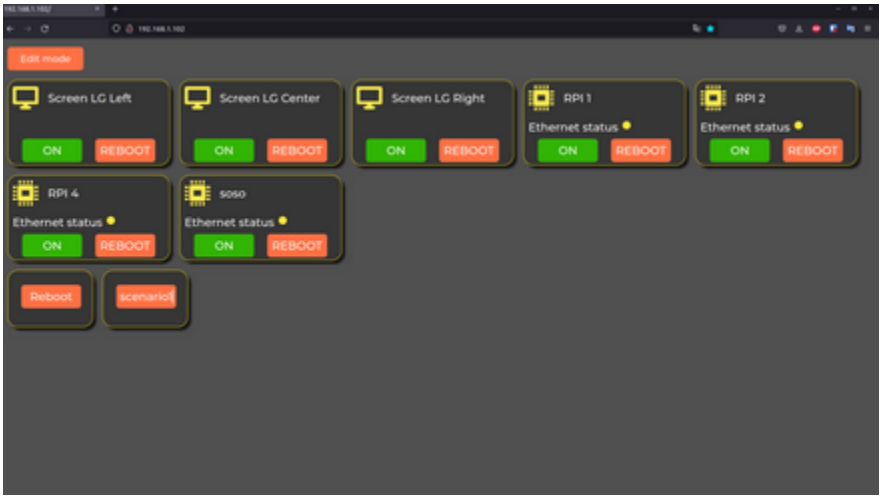
Once you are connected in ssh, go to /home/pi/ and verify if the folder named "SmartPowerManager" has been well created with **ls** command :

```
cd /home/pi/
ls
```

In the list displayed, you should find the folder "SmartPowerManager", if it's the case you can continue the installation, SmartPowerManager's files have been well copied

3 - Result you should obtain

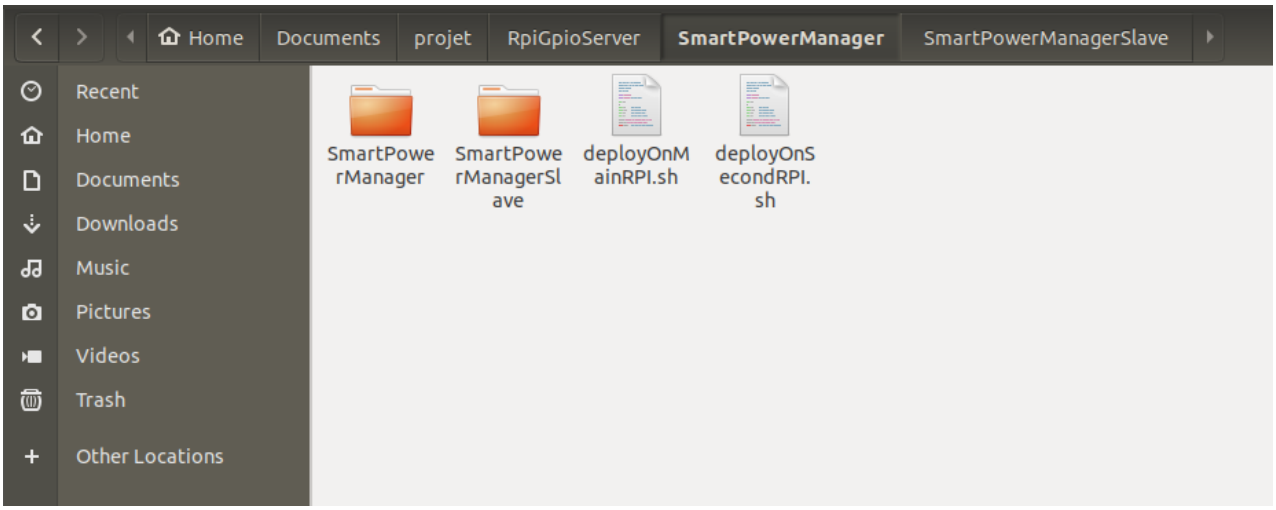
Open your internet browser in any device connected to the same network of the main Raspberry and type his IP, normally you will have something like this (if you have problems, see Troubleshooting installation part)



3 - Install SmartPowerManager on the Second RPI

1 - Installation of the Smart Power Manager for the main Second Board

Open the folder SmartPowerManager: you can see this :

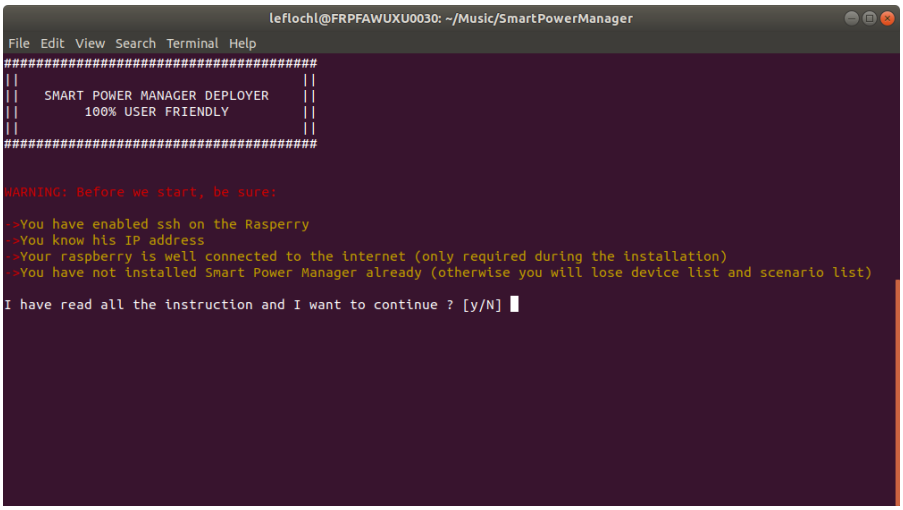


Launch deployOnSecondRPI.sh by doing the following command :

Open a terminal in the directory opened, then type :

```
bash deployOnSecondRPI.sh
```

You will have :



Now follow the instruction displayed then come back (note that the default password on a Raspberry pi is "raspberry")

2 - Verify if the deployOnSecondRPI.sh has done the job

To do that: Connect to Raspberry in ssh typing in a cmd (replace by your raspberry Ip) :

```
ssh pi@[second rpi board controller ip adress]
```

Once you are connected in ssh, go to /home/pi/ and verify if the folder named "SmartPowerManager" has been well created with **ls** command :

```
cd /home/pi/
ls
```

In the list displayed, you should find the folder "SmartPowerManagerSlave", if it's the case you can continue the installation, SmartPowerManager's files have been well copied

3 - Result you should obtain

Open your internet browser in any device connected to the same network of the main Raspberry and type his IP, normally you will have something like this (if you have problems, see Troubleshooting installation part)

Image endommagée

3 -Troubleshooting

1 - During the installation

If nothing work after the installation :

Sometimes during this installation, the Raspberry close the ssh connection :

```
WebServeur/start.sh
             40 100%   39.06kB/s    0:00:00 (xfr#3, to-chk=5/9)
WebServeur/__pycache__/app.cpython-39.pyc
          1,959 100%    1.87MB/s    0:00:00 (xfr#4, to-chk=1/9)
WebServeur/templates/index.html
            328 100%  320.31kB/s    0:00:00 (xfr#5, to-chk=0/9)


Files were copied sucessfuly
pi@192.168.1.149's password:

Installing pip, cron, flask and all python packages needed
Get:1 http://archive.raspberrypi.org/debian bullseye InRelease [23.5 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian bullseye InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org/debian bullseye/main armhf Packages [246 kB]
Get:4 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf Packages [13.2 MB]
Connection to 192.168.1.149 closed by remote host.
```

if it is the case you just have to restart the .sh script: it's normal that at the end the raspberry closes the remote host because it'll reboot automatically to apply all change, but this should happen within a minute or two of the start of the installation.

Normally at the end of the installation, you may have something like this :

```
  Downloading https://www.piwheels.org/simple/ping3/ping3-3.0.2-py3-none-any.whl (12 kB)
Collecting python-socketio==5.5.0
  Downloading https://www.piwheels.org/simple/python-socketio/python_socketio-5.5.0-py3-none-any.whl (55 kB)
Requirement already satisfied: requests==2.25.1 in /usr/lib/python3/dist-packages (from -r /home/pi/SmartPowerManagerSlave/WebServeur/requirements.txt (line 12)) (2.25.1)
Collecting requests-oauthlib==1.0.0
  Downloading https://www.piwheels.org/simple/requests-oauthlib/requests_oauthlib-1.0.0-py2.py3-none-any.whl (21 kB)
Collecting responses==0.12.1
  Downloading https://www.piwheels.org/simple/responses/responses-0.12.1-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: RPi.GPIO==0.7.0 in /usr/lib/python3/dist-packages (from -r /home/pi/SmartPowerManagerSlave/WebServeur/requirements.txt (line 15)) (0.7.0)
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.9/dist-packages (from Flask==2.0.2->-r /home/pi/SmartPowerManagerSlave/WebServeur/requirements.txt (line 1)) (2.0.1)
Requirement already satisfied: Werkzeug>=2.0 in /usr/local/lib/python3.9/dist-packages (from Flask==2.0.2->-r /home/pi/SmartPowerManagerSlave/WebServeur/requirements.txt (line 1)) (2.0.2)
Requirement already satisfied: click>=7.1.2 in /usr/local/lib/python3.9/dist-packages (from Flask==2.0.2->-r /home/pi/SmartPowerManagerSlave/WebServeur/requirements.txt (line 1)) (8.0.3)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages (from Flask==2.0.2->-r /home/pi/SmartPowerManagerSlave/WebServeur/requirements.txt (line 1)) (3.0.3)
Collecting bidict>=0.21.0
  Downloading https://www.piwheels.org/simple/bidict/bidict-0.21.4-py3-none-any.whl (36 kB)
Collecting python-engineio>=4.3.0
  Downloading https://www.piwheels.org/simple/python-engineio/python_engineio-4.3.1-py3-none-any.whl (52 kB)
Collecting oauthlib>=0.6.2
  Downloading https://www.piwheels.org/simple/oauthlib/oauthlib-3.1.1-py2.py3-none-any.whl (146 kB)
Requirement already satisfied: urllib3>=1.25.10 in /usr/lib/python3/dist-packages (from responses==0.12.1->-r /home/pi/SmartPowerManagerSlave/WebServeur/requirements.txt (line 14)) (1.26.5)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from responses==0.12.1->-r /home/pi/SmartPowerManagerSlave/WebServeur/requirements.txt (line 14)) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from Jinja2>=3.0->Flask==2.0.2->-r /home/pi/SmartPowerManagerSlave/WebServeur/requirements.txt (line 1)) (2.0.1)
Installing collected packages: python-engineio, bidict, python-socketio, oauthlib, mysqlclient, responses, requests-oauthlib, ping3, numpy, mysql-connector, mysql, iptools, Flask-SocketIO, Flask-Ext
Successfully installed Flask-Ext-0.1 Flask-SocketIO-5.1.1 bidict-0.21.4 iptools-0.7.0 mysql-0.0.3 mysql-connector-2.2.9 mysqlclient-2.1.0 numpy-1.19.5 oauthlib-3.1.1 ping3-3.0.2 python-engineio-4.3.1 pyth
on-socketio-5.5.0 requests-oauthlib-1.0.0 responses-0.12.1
Connection to 192.168.1.149 closed by remote host.
leflochl@FRPFAWUXU0030:~/Documents/projet/RpiGpioServer/SmartPowerManager$
```

If this did not solve the problem, be sure that your Raspberry is running on Raspberry Pi OS (lite or not) AND that you run the installation connected at a normal network (no Faurecia network, because it will block the installation !)

2 - During the use

If some buttons on the mainboard become white (the button ON/OFF of the devices wired to the second board) :

Be sure that you have defined the second board's IP, if it is the case restart the second board to solve the problem

I cant' access the website

Be sure you are typing the good IP in your browser: the IP to type is the IP of the mainboard, you must type it like this : (example) 192.168.1.110 and NOT like this https:/192.168.1.110

4 - Manual

The list of devices connected to the power manager is located in the main Raspberry in the device.txt file located in the/home/pi/SmartPowerManager folder. (JSON format)

The list of power manager scenarios is located in the main Raspberry in the scenario.txt file located in the/home/pi/SmartPowerManager folder. (not a JSON format, one line per scenario)

To start the power manager, just turn on Main and Second Raspberry and wait about 40 seconds.

To access the website, type the IP address of the main Raspberry in your browser

On the web page, give the IP address of the second RPI to allow the first RPI to connect to it

5 - Next Task

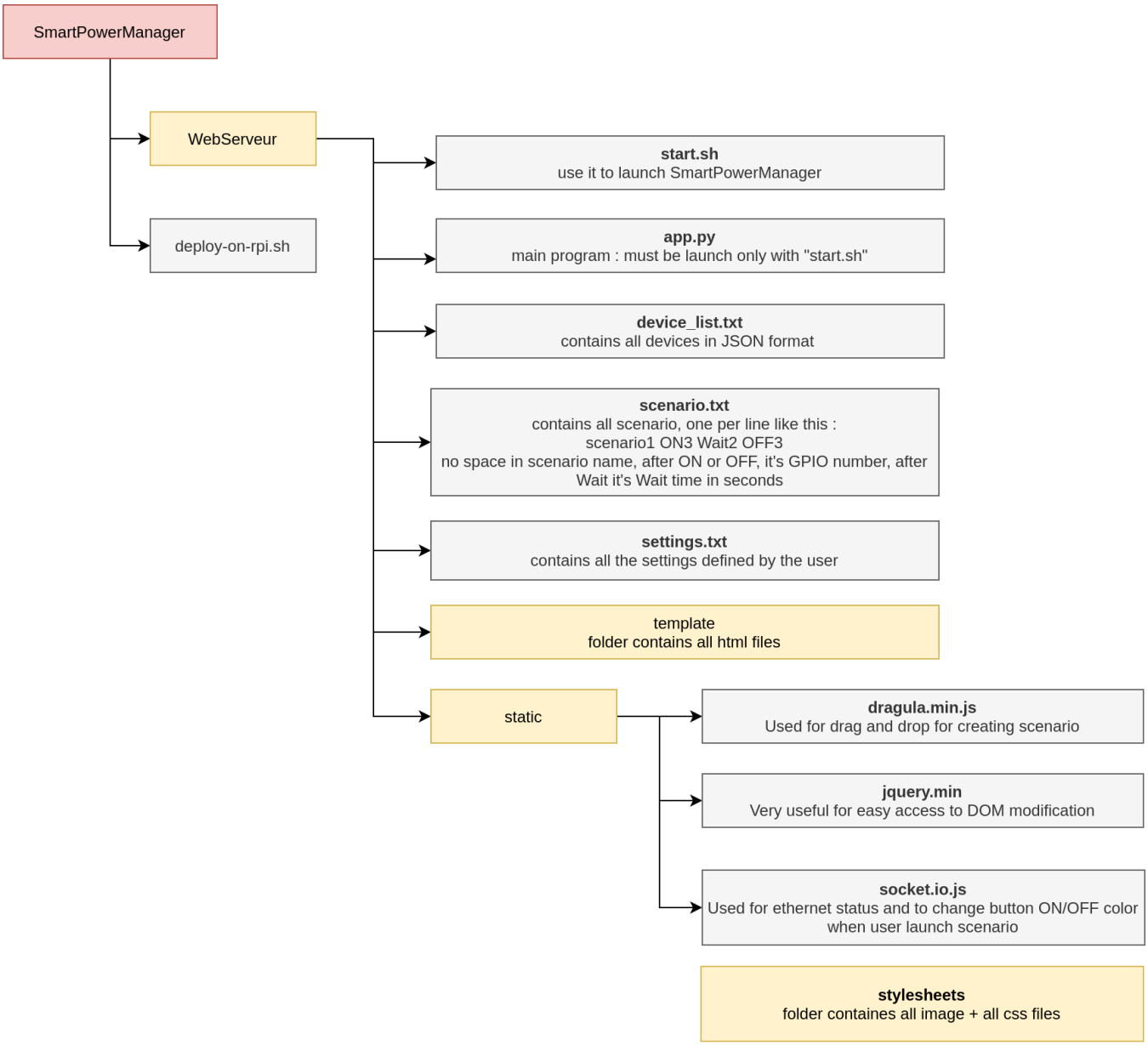
Ethernet status is deactivated because of incoming crash: to be solved

Remove limitation on save zip

6 - How does work SmartPowerManager (technical details)

SmartPowerManager works with :

- Flask: *Flask* is a web framework, it's a Python module that lets you develop web applications easily
- Dragula: javascript library for easy drag and drop
- jQuery: it is a fast, small, and feature-rich JavaScript library.
- Socket.io: JavaScript library for bidirectional and low-latency communication for every platform


SmartPowerManager
WebServeur
deploy-on-rpi.sh
start.sh
use it to launch SmartPowerManager
app.py
main program : must be launch only with "start.sh"
device_list.txt
contains all devices in JSON format
scenario.txt
contains all scenario, one per line like this :
scenario1 ON3 Wait2 OFF3
no space in scenario name, after ON or OFF, it's GPIO number, after Wait it's Wait time in seconds
settings.txt
contains all the settings defined by the user
template
folder contains all html files
static
dragula.min.js
Used for drag and drop for creating scenario
jquery.min
Very useful for easy access to DOM modification
socket.io.js
Used for ethernet status and to change button ON/OFF color when user launch scenario
stylesheets
folder containes all image + all css files