Module **`main`**

main.py | Robin Forestier | 2021 / 2022

Goal : Create a user interface to controle the light of the ELO's workshop.

Explanation : I use Flask for create a Web server. A shield is plug on the Raspberry Pi (server) with optocoupleur to ABB module with the GPIO.

┌──┐ │ Raspberry Pi 3B+ │ ┌────────────────────────────────┐ │ ┌──┐ │ │ Shield Octo │ │ │ │ │ │ │ │ GPIO 2 | │ │ │ │ X1 / DRV3 | │ │ GPIO 3 | │ │ │ │ X1 / DRV4 | │ │ GPIO 4 | │ │ │ │ X2 / DRV1 | │ │ | │ │ │ │ | │ │ GPIO 10 | │ │ │ │ X3 / DRV3 | │ │ GPIO 9 | │ │ │ │ X3 / DRV2 | │ │ GPIO 11 | GPIO 8 │ │ │ │ X3 / DRV4 | X3 / DRV1 │ │ | GPIO 7 │ │ │ │ | X2 / DRV4 │ │ GPIO 0 | GPIO 1 │ │ │ │ X1 / DRV1 | X1 / DRV2 │ │ GPIO 5 | │ │ │ │ X2 / DRV2 | │ │ GPIO 6 | │ │ │ │ X2 / DRV3 | │ │ │ │ │ └────────────────────────────────┘ │ │ │ │ │ └──┘ │ │ ┌─────────┐ │ │ ┌──────┐ ┌──────┐ │ │ │ │ │ │ │ │ │ │ │ └───┴──────┴─────┴──────┴──────┴──────────┴─────┘

Liens

GitLab : http://172.16.32.230/Forestier/controle-des-lumieres-knx (http://172.16.32.230/Forestier/controle-des-lumieres-knx)

XWiki : https://xwiki.serverelo.org/xwiki/bin/view/Centre%20de%20Formation%20ELO/Projets/Controle%20des%20lumières%20KNX/ (https://xwiki.serverelo.org/xwiki/bin/view/Centre%20de%20Formation%20ELO/Projets/Controle%20des%20lumières%20KNX/)

Global variables

`var authorize_ip`

Ip of the server (for the local user) use to skip authentication.

`var buttonSts_p1`

value of all the page1 buttons

`var buttonSts_p2`

value of all the page2 buttons

`var color`

color of the 8 neon

`var rpi`

rpi is used to know if the programm is running on a Raspberry Pi or not.

rpi = True –> run on Raspberry Pi.

esle : False

`var users`

user : elo - basic user / can only modifie light admin - all acces (settings acces) local - only used for the raspberry why the tuchscreen (server) / can't acces to settings / don't have to login.

`var warning`

warning from the app to the server

Functions

`def activate_job()`

activate_job

Fonction to run job on backround of the web server (with thread)

Parameters

…

Returns

…

`def getTime()`

Get time on format : HH:MM

Parameters

…

Returns

`current_time` : `str`

The current time in format HH:MM

`def gpio_modif()`

modif all the GPIO by looking into buttonSts_p1[]

Parameters

…

`def handle_exception(e)`

handle_exception : all http error are show on the 404.html page

`def login()`

login page

Receive from POST : username / password We first check if username existe and after password. If it's a bad username or pass we flash the error.

Parameters

…

Returns

`login.html : str`
: Page login in html and (css).

`time` : `str`
: The current time.

`warning` : `str`
: The différent warning.

`def page1()`

page1

The page1 show on the left a map of the lights and on the right 9 buttons.

Parameters

…

Returns

`page1.html : str`
: Page page1 in html and (css).

`button` : `list`

List of the state of the buttons.

color : `list`
List of neon colors on the map.

time : `str`
The current time.

warning : `str`
The différent warning.

`def page2()`

page2

The page2 show on the left a map of the lights and on the right 9 buttons.

Parameters

...

Returns

`page2.html : str`
Page page1 in html and (css).

button : `list`
List of the state of the buttons.

color : `list`
List of neon colors on the map.

time : `str`
The current time.

warning : `str`
The différent warning.

`def settings()`

settings

The settings page can only be acces via admin login.

Parameters

...

Returns

`settings.html : str`

Page page1 in html and (css).

`time` : `str`

The current time.

`warning` : `str`

The différent warning.

Classes

`class User (id, username, password)`

User : creat different user for the login.

Parameters

`id` : `int`

A basic id for eache login.

`username` : `str`

The username for the login.

`password` : `str`

The password of the login.

Index

Global variables

Functions

Classes

Generated by *pdoc* 0.10.0.