Contrôleur Poêle Pellet Marina Interstove

Table des matières

pascal_bornat@hotmail.com

Introduction

Les commandes décrites dans ce document ont été extraites des communications entre le module wifi et la carte contrôleur EVO LCD 7 du poêle à pellet. La liste des commandes fournies dans ce document n'est pas exhaustive. Elle ne couvre que les fonctionnalités suivantes :

- Allumage
- Arrêt de chauffe
- Lecture de l’état (allumé, éteint, refroidissement, etc...)
- Commande de la température de consigne
- Commande de la puissance de chauffe

pascal_bornat@hotmail.com

Description de l'interface série

La carte contrôleur du poêle possède une interface de contrôle sur laquelle vient se connecter le boîtier wi-fi du constructeur. Cette interface est constituée de 4 bornes :

1. La masse
2. L'alimentation 5V
3. Le signal Rx du port serie 5v TTL
4. Le signal Tx du port serie 5v TTL

Ce port série asynchrone fonctionne à une vitesse de 115200bps. Pour information, à l'initialisation du contrôleur, un message est envoyé à une vitesse standard juste inférieure. Cette interface ne semble pas avoir de contrôle de flux type xon/xoff.

pascal_bornat@hotmail.com

Description des commandes

Format des commandes

Les commandes doivent être envoyées vers la carte contrôleur qui émet systématiquement une réponse, soit un acquittement, soit la réponse à la requête demandée. Les commandes sont constituées d'un caractère de début de trame et d'un caractère de fin de trame. Si l'un ou l'autre des caractères est manquant, la commande n'est pas prise en compte. Le caractère de début de trame est ESC et le caractère de fin de trame est &.

Exemple de trame : <ESC>RDA00067<&> ce qui donne ←RDA00067&

Lorsqu'une commande envoyée ne nécessite pas de valeur de retour, une réponse d'acquittement est retournée. Cette valeur est : 00000020

Liste des commandes

Ci-dessous la liste des commandes que l'on doit envoyer sur le port série de la carte contrôleur

Statut :

Valeur:←RD90005f&

Cette commande renvoie une trame du type 0a0000yy. Je n'utilise pas la valeur de l'octet yy. Voir l'exemple de code pour plus d'information.

Les valeurs de a sont les suivantes :

Valeur	Description
1	Allumage
2 *	Chauffage 1
4 *	Chauffage 2
8	Refroidissement

*A noter que je n'ai pas trouvé la différence entre les valeurs 2 et 4.

Température ambiante

Valeur :←RD100057&

Cette commande renvoie une trame du même format. La valeur retournée contient la température ambiante codée comme suit :

XXXX00YY

La valeur XX contient en hexadécimal la valeur de la température qu'il faut diviser par 10 pour obtenir une valeur décimale.

Exemple : **00E9**003E → x00E9=233 → 233/10=23,3°c

pascal_bornat@hotmail.com

Je ne tiens pas compte de la valeur YY.

Puissance

Valeur :←RF00X0YY&

Cette commande permet d'ajuster la puissance de chauffe de 1 à 5. Dans la commande, X représente la valeur de la puissance (1 à 5). Dans la commande, YY représente la valeur de X à laquelle on ajoute x58 ou 88 en décimal.

Exemple : ←RF001059& règle la puissance sur le niveau 1, ←RF00205A& règle la puissance sur 2, etc..

La valeur retournée par cette commande est la valeur d'acquittement : 00000020

Consigne de température

Valeur :←RF2XX0YY&

Cette commande permet d'ajuster la température de consigne. Dans la commande, XX représente la consigne de température en hexadécimal. Dans la commande, YY représente la valeur de consigne de la température à laquelle on ajoute x4B ou 75 en décimal.

Exemple : ←RF21405f& règle la température de consigne sur 20°c. A noter que la température de consigne est un nombre entier.

La valeur retournée par cette commande est la valeur d'acquittement : 00000020

Allumage

Valeur :←RF001059&

La valeur retournée par cette commande est la valeur d'acquittement : 00000020

A noter qu'il est préférable d'envoyer cette commande uniquement quand le statut du poêle est : Éteint

Extinction

Valeur :←RF000058&

La valeur retournée par cette commande est la valeur d'acquittement : 00000020

A noter qu'il est préférable d'envoyer cette commande quand le statut du poêle est : Allumé ou Allumage

Mode ECO on

Valeur :←10010022&

La valeur retournée par cette commande est la valeur d'acquittement : 00000020

A noter que ce mode n'est pas disponible sur tous les appareils

pascal_bornat@hotmail.com

Mode ECO off

Valeur :←10030024&

La valeur retournée par cette commande est la valeur d'acquittement : 00000020

A noter que ce mode n'est pas disponible sur tous les appareils

Lecture des erreurs

Valeur :←RDA00067&

Cette commande permet de lire les codes d'erreur lorsque le poêle est en défaut, par exemple lorsque le réservoir de granulé est vide. A noter que je n'ai pas trouvé comment remettre à zéro à distance le contrôleur du poêle lorsqu'un défaut est présent. Il est possible que pour des questions de sécurité, ce ne soit pas possible. C'est dommage essentiellement pour l'erreur n°8 (coupure de courant) qui n'est pas vraiment une panne et qui interdit de redémarrer le poêle à distance tant que l'erreur n'est pas acquittée manuellement.

Les valeurs retournées ont le format suivant :

00xx00yy

Dans la valeur retournée, l'octet xx représente le code d'erreur. Je n'utilise pas l'octet yy qui semble être la valeur de l'erreur à laquelle est ajouté 0x20.

Les codes d'erreurs sont les suivants :

Valeur	Description
0	OK
1	Allumage raté
2	Capteur d'aspiration défectueux
3	Aspiration air insuffisant
4	Température d'eau
5	Pellet terminé
6	Pressostat
7	
8	Absence de courant
9	Moteur fumées
10	Surtension carte
11	Date dépassée
12	
13	Régulation capteur aspiration
14	Surchauffe

pascal_bornat@hotmail.com

A noter que je fais l'hypothèse que les codes d'erreurs retournés par l'interface de commande sont les mêmes que ceux décrits dans le manuel. J'ai pu vérifier que ces codes sont identiques pour les erreurs 5 et 8 qui sont les plus courantes sur un poêle en état.

pascal_bornat@hotmail.com

Intégration Jeedom

L'objectif de la phase de rétro ingénierie décrite précédemment est de permettre l'intégration du contrôle du poêle dans Jeedom. En ce qui me concerne, je souhaite pouvoir contrôler à distance le poêle, mais cela était déjà possible avant au moyen du module wi-fi qui est une option du constructeur. L'intérêt de cette intégration est de pouvoir développer des scénarios dans Jeedom pour gérer automatiquement les commandes. Dans mon cas particulier, le poêle vient en complément d'un chauffage centrale à radiateur. Le contrôle du poêle d'une façon automatique me permet de mieux utiliser les deux moyens de chauffage, ensemble ou en complément , et de réduire ma facture énergétique.

Intégration matérielle

Le plus simple dans mon cas était de contrôler le poêle à distance via wi-fi, je devais donc trouver une passerelle wi-fi vers le port série du contrôleur. En cherchant une solution de ce type, je suis tombé sur une solution à base de carte ESP8266 et du logiciel associé ESPeasy. Dans cette configuration, en sélectionnant une « Device » du type « Communication - Serial Server « j'obtiens directement ce que je recherche.

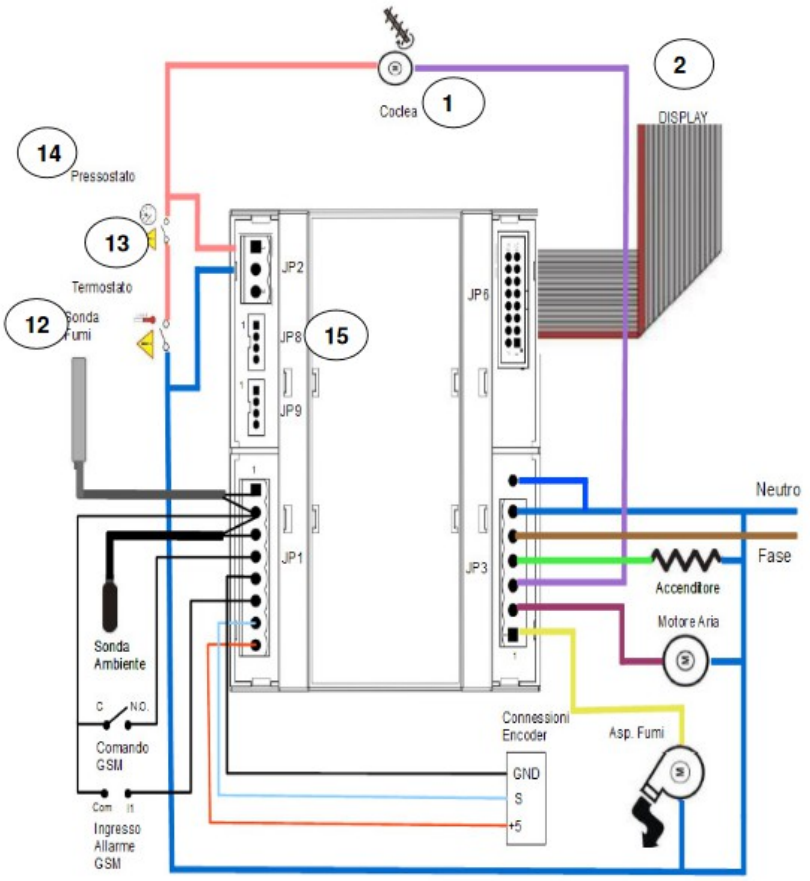
Pour information, compte tenu du faible courant nécessaire à la carte ESP8266, j'ai décidé d' alimenter celle-ci par le 5V fourni par la carte contrôleur du poêle.

pascal_bornat@hotmail.com

Port série du contrôleur

Le port série TTL du contrôleur se trouve sur le connecteur JP8, emplacement n° 15 sur le schéma ci-dessous.

Carte électronique



1 : vis sans fin ; 2 :display ; 3 : neutre ; 4 :phase ; 5 : bougie ; 6 : ventilateur air ; 7 : extracteur des fumées ; 8 : connexions encodeur, 9 : accès alarme GSM ; 10 : commande GSM ; 11 : sonde ambiante ; 12 : sonde fumées ; 13 : thermostat ; 14 : pressostat ; 15 : JP8 branchement câble dispositif wi-fi

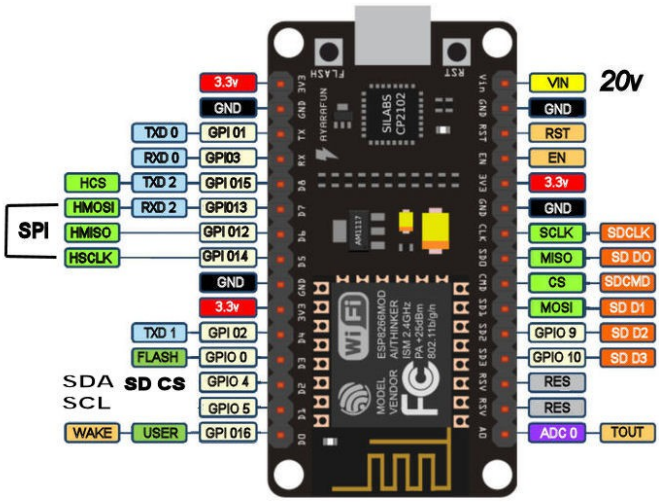
Brochage du connecteur JP8

Connecteur JP8.

Connexion	Description
1	GND Sériel
2	Signal RX TTL
3	Signal TX TTL
4	+5V Sériel

pascal_bornat@hotmail.com

Brochage de l'ESP8266



Interconnexion des ports série

JP8 Contrôleur	Connecteur ESP
Broche 1 - GND	Broche - GND
Broche 2 - Rx	Broche - Tx *
Broche 3 - Tx	Broche – Rx *
Broche 4 - +5v	Broche - VIN

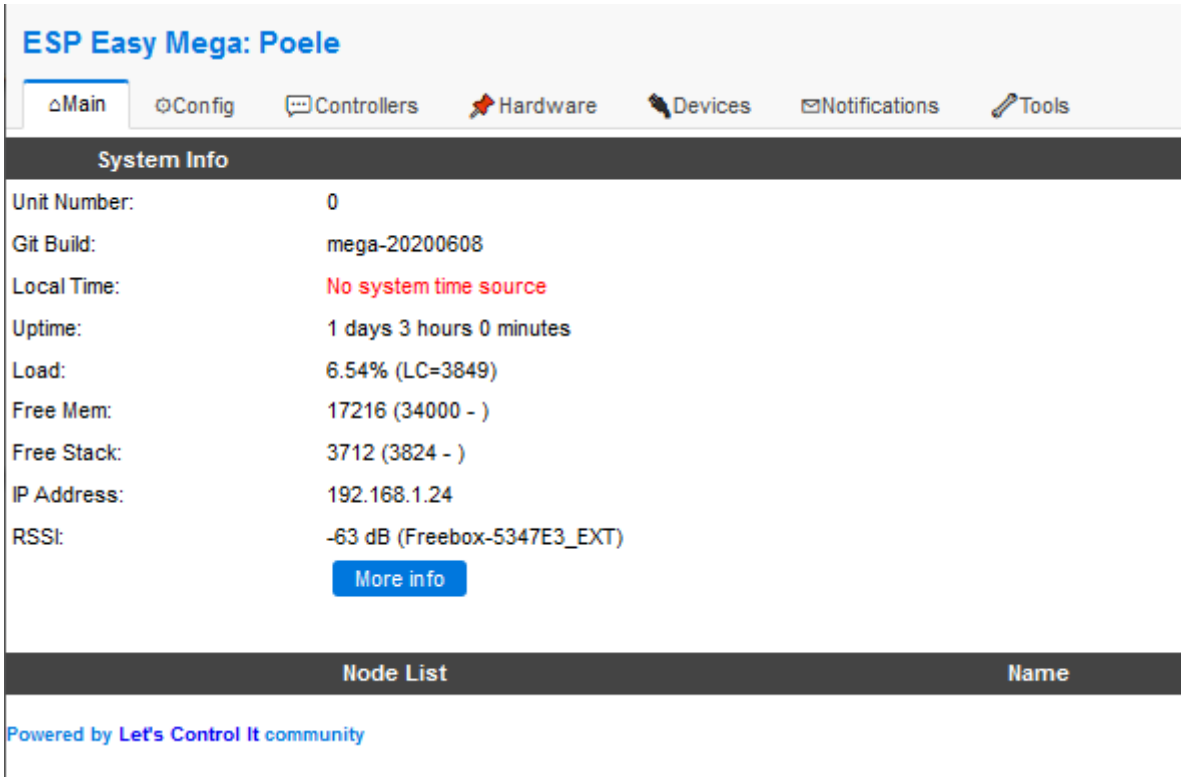
- **Note : compte tenu de l'écart de tension entre la carte contrôleur du poêle (5v) et l'ESP8266, il est impératif d'intercaler un circuit « Level shifter » entre les boches Tx et Rx. Pour ma part j'ai opté pour un circuit à 4 entrées/sorties.**
- **Le branchement du circuit « level shifter » nécessite une alimentation 5V, une alimentation +3.3V que l'on peut prendre à partir d'une sortie d'alimentation de la carte ESP8266. Il faut ensuite connecter les broches Tx et Rx du controleur coté HV et les broches Rx et Tx de l'ESP8266 coté LV. Sur ce type de circuit, les signaux sont indifféremment entrée ou sortie.**
- **Enfin n'oubliez pas de croiser les signaux Rx et Tx au niveau de l'ESP8266.**

pascal_bornat@hotmail.com

Configuration ESPeasy

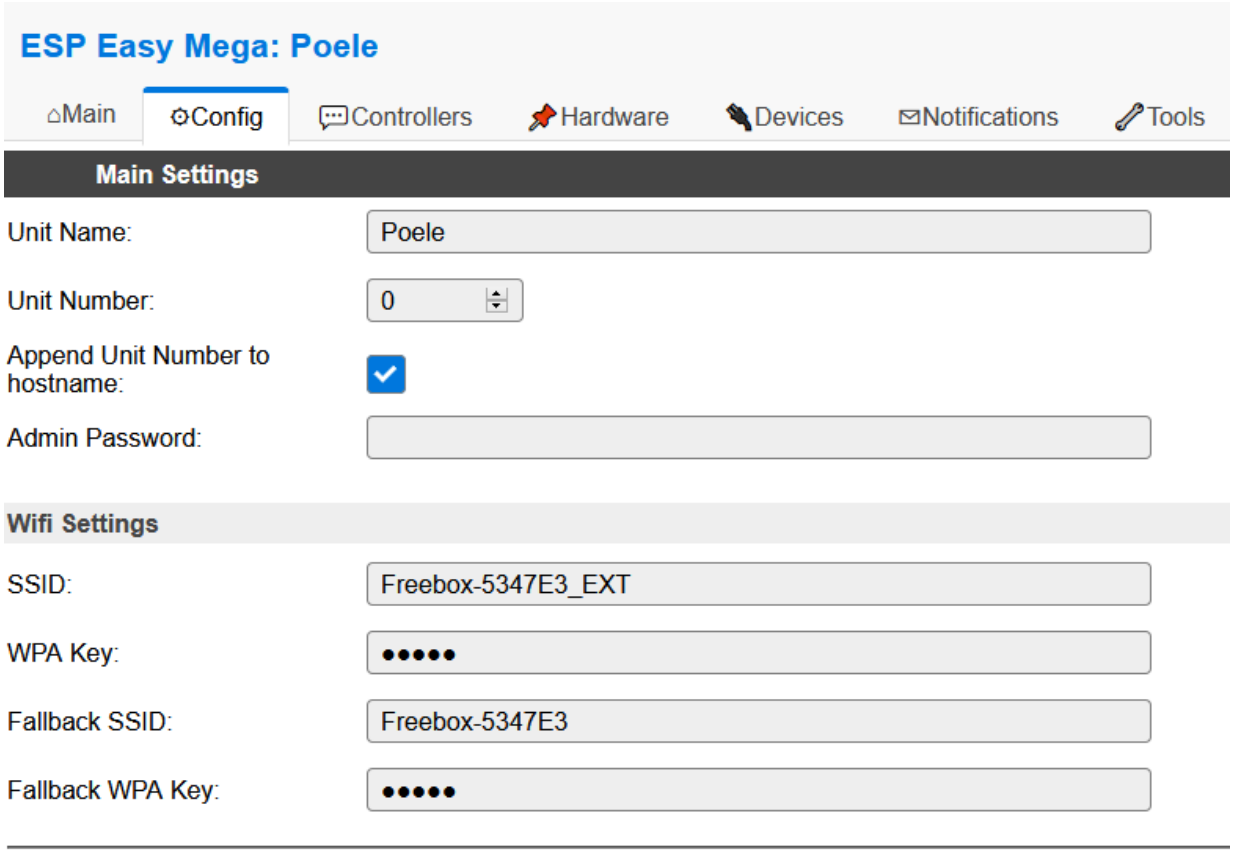
Lorsqu'on reçoit la carte ESP8266, celle-ci doit être flasher avec le code ESPeasy. Je ne détaille pas ici cette étape car on trouve sur le net plusieurs tutoriels qui présentent cette opération. Lorsque vous faites l'acquisition de la carte ESP8266, je vous conseille de prendre le modèle qui possède un port usb, ce qui facilite grandement l'opération de flashage avec un PC.

Une fois le flashage de la carte terminé, on obtient une adresse IP. Il faut ouvrir un navigateur web à cette adresse et on obtiens une serie d'onglets comme ci-dessous.



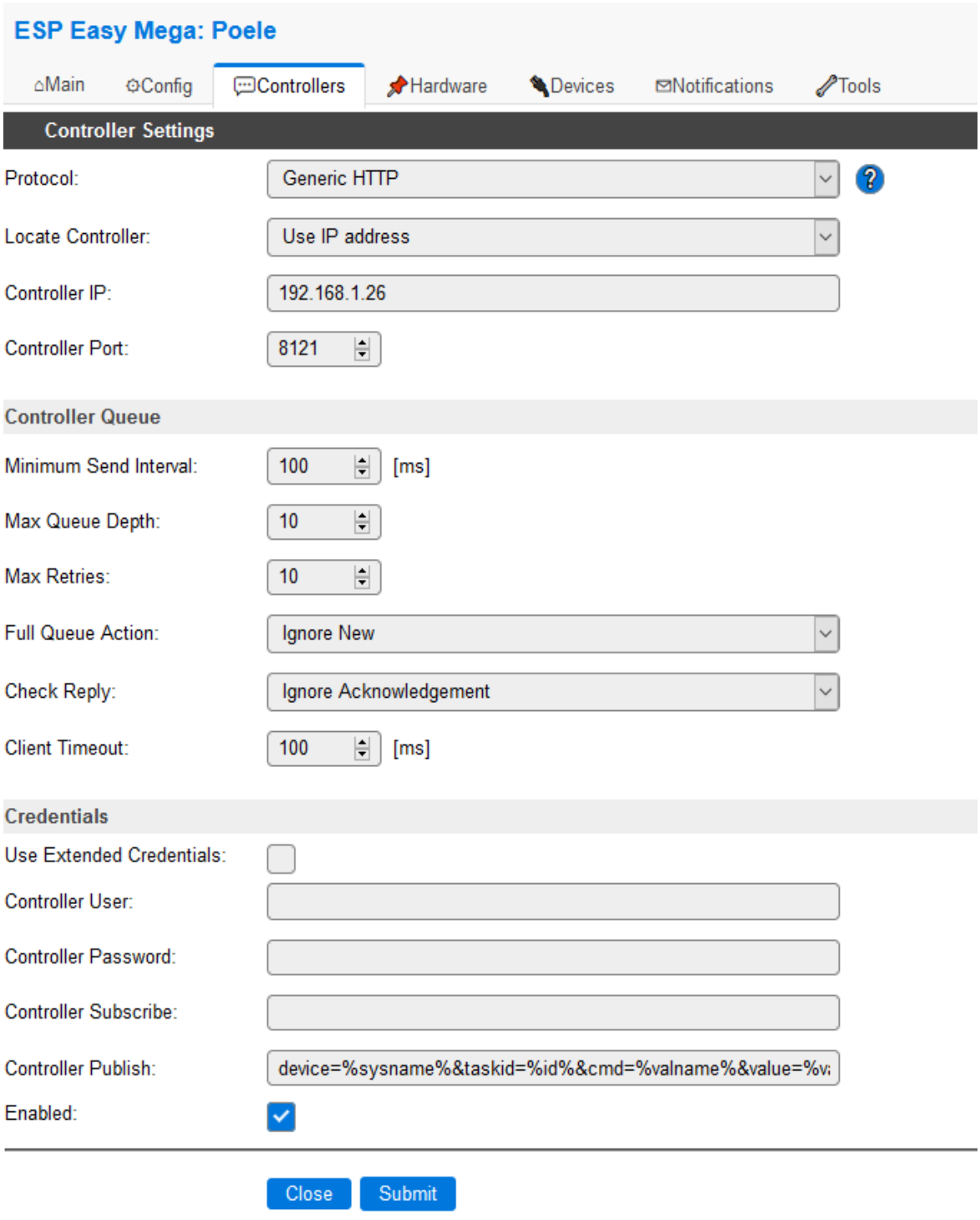
Je présente ci-dessous une série de copies d'écran qui montrent les différentes configurations des onglets.

Onglet Config



Cet onglet contient déjà l'information SSID et Key de la connexion wifi. Cependant, vous pouvez ajouter une connexion supplémentaire.

Onglet Controllers

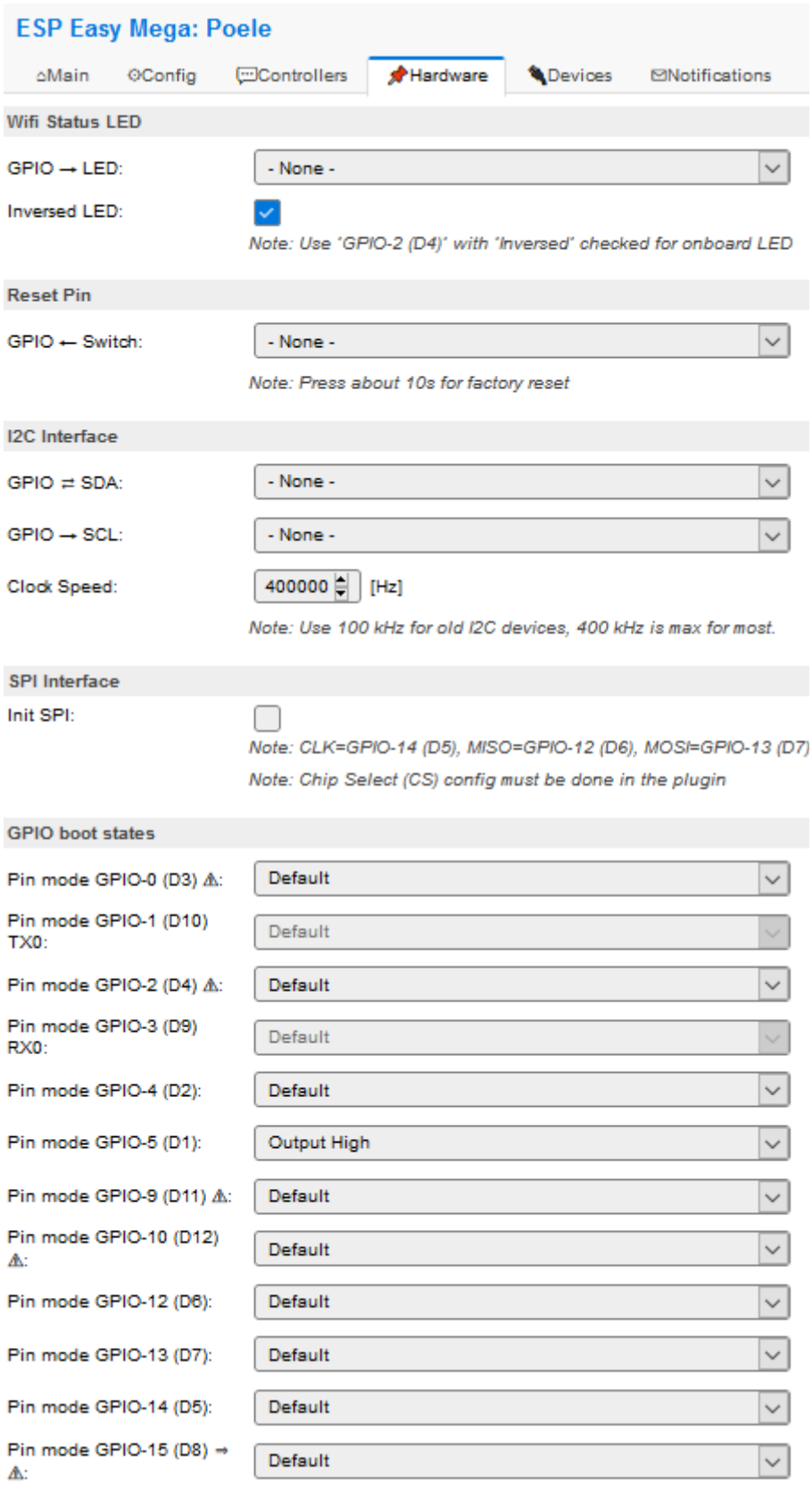


A ce stade, il faut installer le plugin ESPeasy qui contient les informations pour la suite des opérations, et activer le mode d'inclusion du plugin.

Sur cet onglet, vous devrez saisir, l'adresse IP du controlleur Jeedom, le port ainsi que le champ Controller Publish. Ces données se trouvent sur la page Configuration du plugin EspEasy.

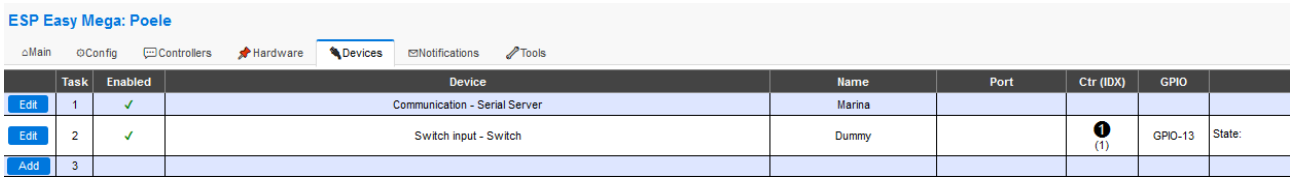
pascal_bornat@hotmail.com

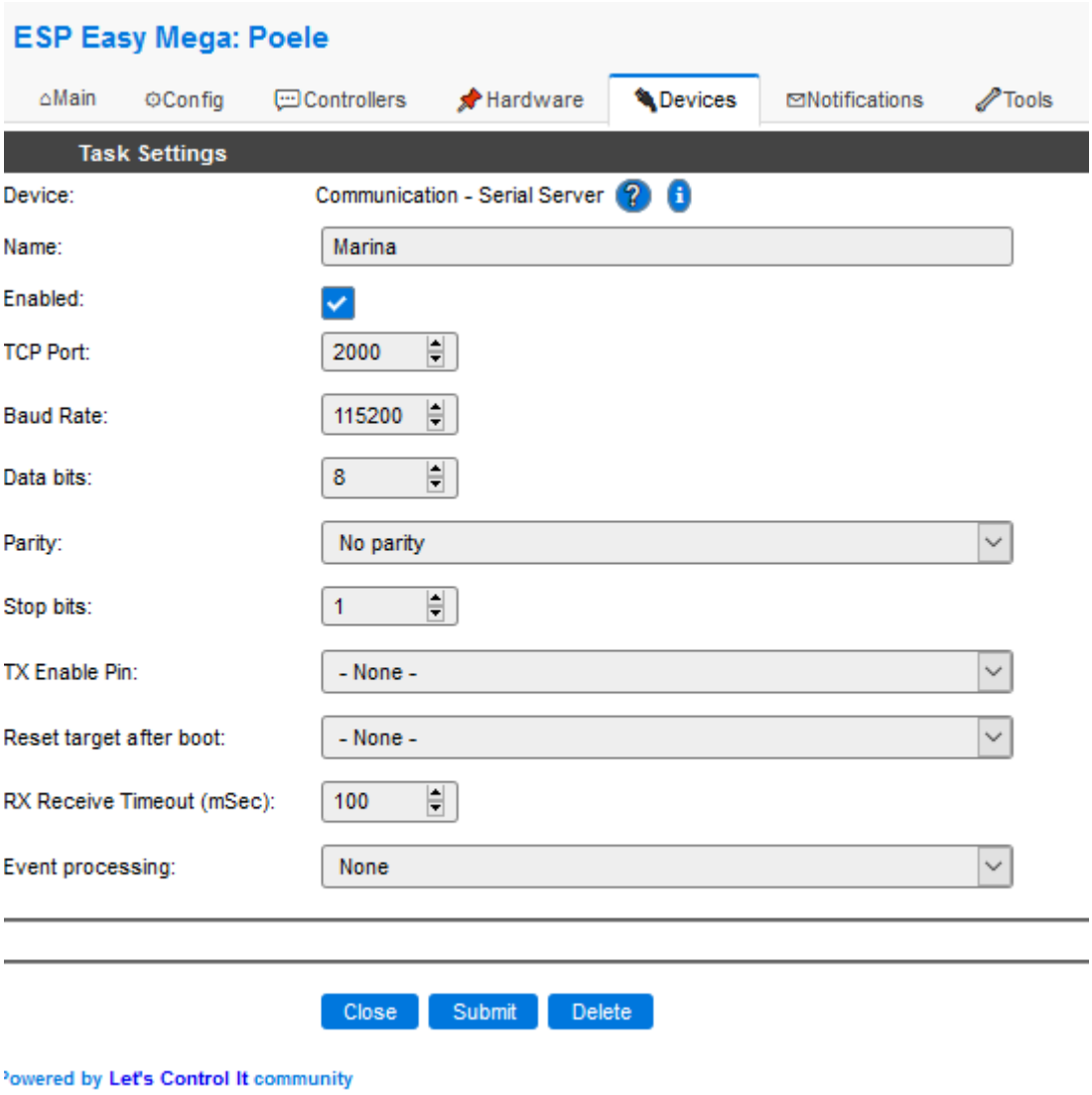
Onglet Hardware



pascal_bornat@hotmail.com

Onglet Devices





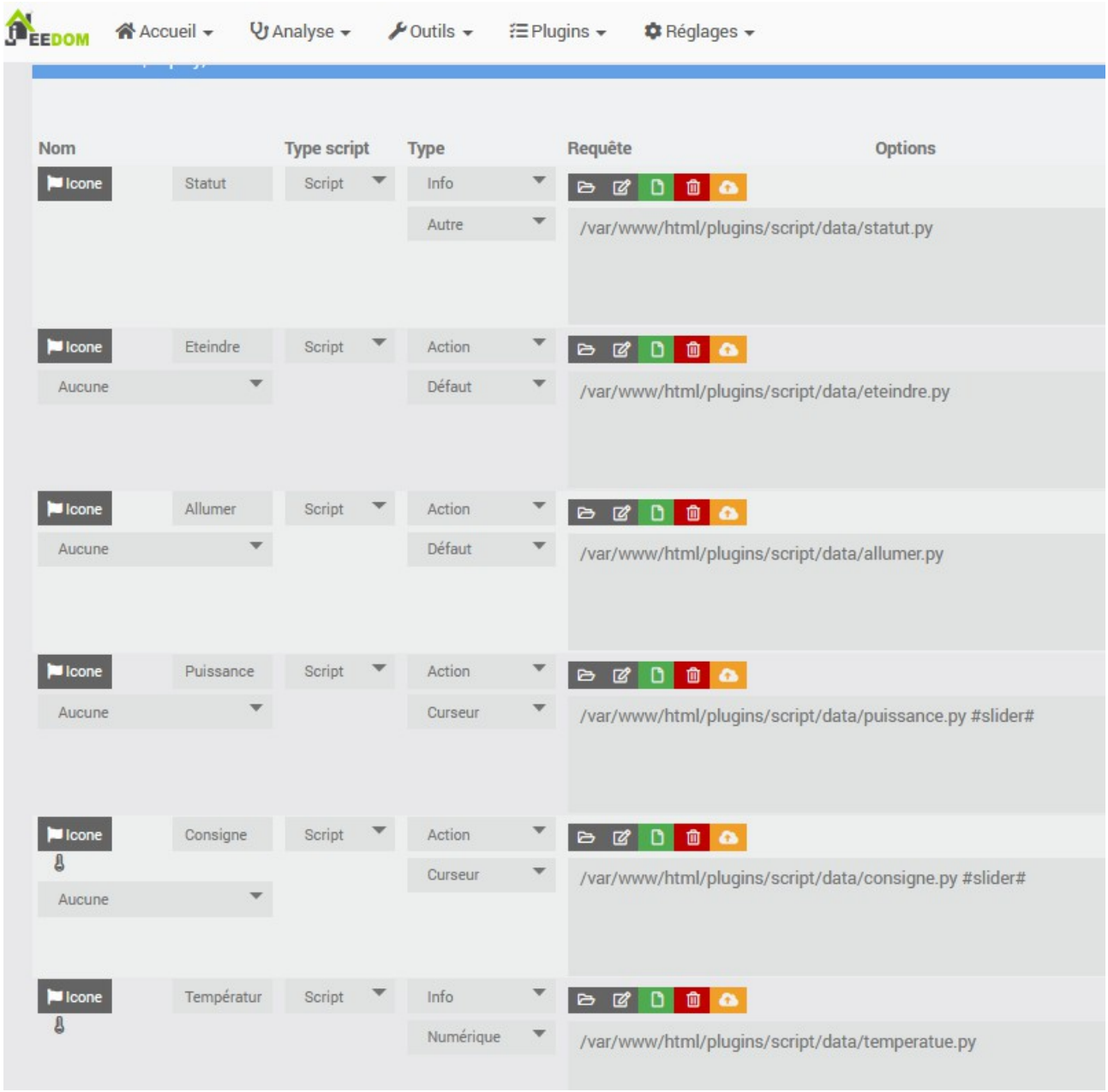
pascal_bornat@hotmail.com

Développement des commandes

Afin de permettre le développement des commandes de contrôle du poêle, j'utilise le plugin « **Script** » qui permet d'écrire du code dans différents langages. Pour ma part j'ai choisi d'écrire les commandes en Python.

A noter que l'équipement correspondant au plugin ESPeasy n'est plus utilisé dans le cas de cette communication série. Je pense qu'il doit être possible de récupérer des données de ce plugin , comme l'adresse IP par exemple. Je pense qu'il y a moyen d'impliquer davantage cet équipement, mais pour le moment, je ne sais pas comment faire.

Ci-dessous une copie d'écran de l'ensemble des commandes du plugin Script:



Chaque commande fait appel à un script en python dont le nom est mentionné dans le champ Requète. Le champ #slider # permet de passer la valeur du curseur au script associé.

Socket TCP/IP

Le « device » « communication – Serial Server » sélectionné pendant la configuration ESPeasy permet de transmettre directement au port série des données sur un socket TCP/IP. Donc une fois le socket TCP/IP ouvert, toutes les données envoyées se retrouvent sur le port série TX de l'ESP8266. Dans l'autre sens, les données lues sur le port série Rx (la réponse d'une commande) de l'ESP8266 se retrouvent en lecture sur le socket TCP/IP.

Exemple de code en python :

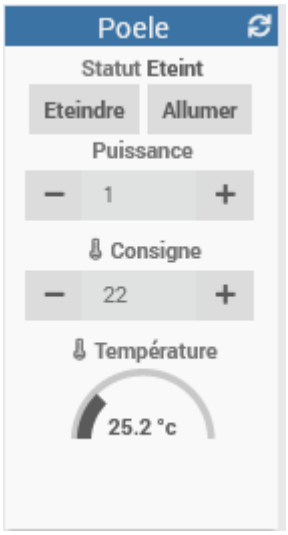
```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
sock.connect(("192.168.1.18", 2000)) #adresse IP  et numéro de port Ser2Net
data = "\x1bRD90005f&" #exemple de donnée, x1b est la valeur en hexa du caractère « ESC ».
sock.send(data.encode()) # envoi des données

#Exemple de lecture de la réponse :

dataFromServer = sock.recv(10); #on attend 10 bytes en réponse à la commande.
sock.close()# fermeture du socket
```

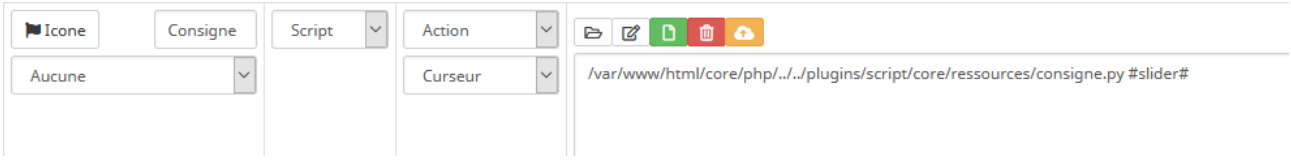
Passage d'argument en python

Le passage d'un argument entre le plugin et le script est très simple quand on sait comment faire. Par exemple pour contrôler la température, j'utilise un Slider dont la valeur est passée au script (voir image ci-dessous)



Du coté de la commande du plugin, on ajoute au nom du fichier le mot clé #slider#.

Exemple : /var/www/html/core/php/../../plugins/script/core/ressources/puissance.py #slider#



Du coté du code du script python, pour récupérer la valeur du slider , on a juste :

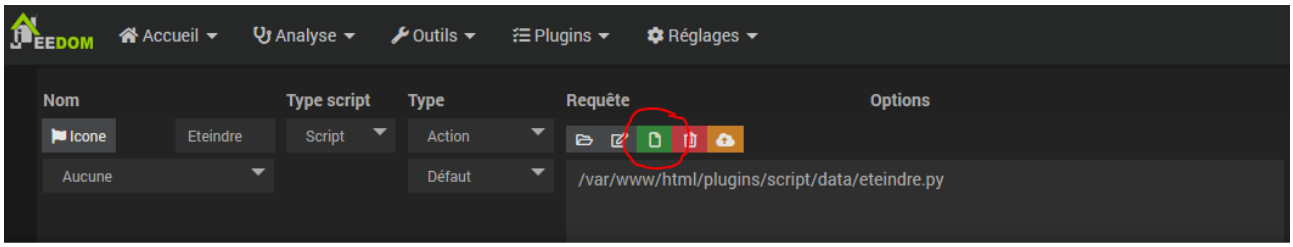
import sys

consigne=sys.argv[1]# la variable « consigne » contient la valeur sélectionnée par le slider.

Exemple de Code

Le script Python ci-dessous donne un exemple de code pour chacune des commandes de l'équipement. Les valeurs **adresseIP** et **port#** dans le code ci-dessous sont à remplacer par les valeurs correspondant à votre installation.

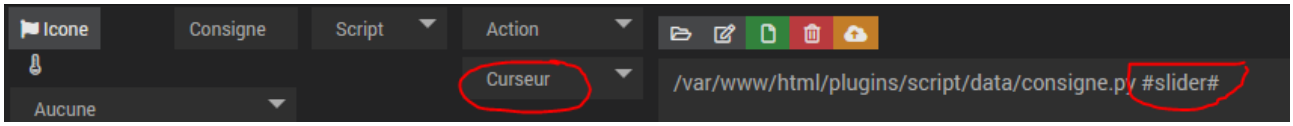
Pour associer le code à une commande, vous devez, après avoir installer le plugin Script créer une commande. Puis dans la colonne « Requête », cliquer sur « Nouveau »



Puis il faut nommer le script exemple :« eteindre » puis copier le code correspondant ci-dessous.

Sur les commandes « Consigne » et « puissance » il faut créer un curseur pour sélectionner une valeur. Le mot clé #slider# est à placer à la suite du nom du fichier. Il sera récupéré dans le code, voir chapitre « Passage d'argument en python ».

Exemple :



statut.py

```
import socket

str_eteint="00000020"
refroid=0x08000000
allumage=0x01000000
allume1=0x02000000
allume2=0x04000000

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
sock.connect(("adresseIP", port#))
cmdGetStatus = "\x1bRD90005f&"
sock.send(cmdGetStatus.encode())
dataFromServer = sock.recv(10);
statusInt=int(dataFromServer[1:9],16)
```

```
if refroid & statusInt :
    print("Refroidissement")
elif str_eteint in dataFromServer:
    print("Eteint")
elif allumage & statusInt:
    print("Allumage")
elif allume1 & statusInt:
    print("Allume1")
elif allume2 & statusInt:
    print("Allume2")
else:
    print(dataFromServer)

sock.close()
```

allumer.py

```
import socket
import sys
str_eteint="00000020"
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
sock.connect(("adresseIP", port#))
data = "\x1bRD90005f&"
sock.send(data.encode())
dataFromServer = sock.recv(10);

if str_eteint in dataFromServer:
    data = "\x1bRF001059&"
    sock.send(data.encode())
    dataFromServer = sock.recv(10);

sock.close()
```

pascal_bornat@hotmail.com

Eteindre.py

```
import socket
import sys

str_allumage="01010022"
str_allume="02010023"
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
sock.connect(("adresseIP", port#))
data = "\x1bRD90005f&"
sock.send(data.encode())
dataFromServer = sock.recv(10);

if str_allumage in dataFromServer:
    data = "\x1bRF000058&"
    sock.send(data.encode())
    dataFromServer = sock.recv(10);
elif str_allume in dataFromServer:
    data = "\x1bRF000058&"
    sock.send(data.encode())
    dataFromServer = sock.recv(10);
sock.close()
```

pascal_bornat@hotmail.com

puissance .py

```
import socket
import sys

puissance=sys.argv[1]
str_allume="02010023"
puissanceInt=int(puissance)

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
sock.connect(("adresseIP", port#))

data = "\x1bRF00x0yy&"
constInt=88
codeInt=constInt+puissanceInt
codeHexStr=hex(codeInt)
data1=data.replace("yy",codeHexStr[2:4])

puissanceHexStr=hex(puissanceInt)
cmdPuisStr=data1.replace("x",puissanceHexStr[2:3])

dataStat = "\x1bRD90005f&"
sock.send(dataStat.encode())
answStat = sock.recv(10);

if str_allume in answStat:
   sock.send(cmdPuisStr.encode())
   dataFromServer = sock.recv(10);
sock.close()
```

consigne.py

```
import socket
import sys

consigne=sys.argv[1]
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
sock.connect(("adresseIP", port#))
data = "\x1bRF2xx0yy&"
consigneInt=int(consigne)
constInt=75
codeInt=consigneInt+constInt
codeHexStr=hex(codeInt)
consigneHexStr=hex(consigneInt)
data2=data.replace("yy",codeHexStr[2:4])
data3=data2.replace("xx",consigneHexStr[2:4])
sock.send(data3.encode())
dataFromServer = sock.recv(10);
sock.close()
```

pascal_bornat@hotmail.com

temperature.py

```
import socket
import time
import sys

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
sock.connect(("adresseIP", port#))
data = "\x1bRD100057&"
sock.send(data.encode())
dataFromServer = sock.recv(10);

if len(dataFromServer) != 0:
  tempStrHex=dataFromServer[1:5]
  tempIntDec=int(tempStrHex,16)
  tempFloatDec=tempIntDec/10.0
  print(tempFloatDec)
else:
  print(dataFromServer)
sock.close()
```

Erreur.py

```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
sock.connect(("adresseIP", port#))
cmdGetStatus = "\x1bRDA00067&"
sock.send(cmdGetStatus.encode())
dataFromServer = sock.recv(10);
statusInt=int(dataFromServer[4:6],16)

if statusInt == 0 :
        print("OK")
elif statusInt == 1 :
   print("Allumage rate")
elif statusInt == 2 :
   print("Capteur aspiration defectueux")
elif statusInt == 3 :
   print("Aspiration air insuffisant")
elif statusInt == 4 :
   print("Temperature de l'eau")
elif statusInt == 5 :
   print("Pellet termine")
elif statusInt == 6 :
   print("Pressostat")
elif statusInt == 8 :
   print("absence de courant")
elif statusInt == 9 :
   print("Moteur fumee")
elif statusInt == 10 :
   print("Surtension carte")
elif statusInt == 11 :
   print("Date depassee")
elif statusInt == 13 :
   print("Regulation capteur aspiration")
elif statusInt == 14 :
```

```
    print("Surchauffe")
else:
    print(statusInt)

sock.close()
```