

Contrôleur Poêle Pellet Marina Interstove

Table des matières

Introduction.....	2
Description de l'interface série.....	3
Description des commandes.....	3
Format des commandes.....	3
Liste des commandes.....	3
Statut :.....	3
Température ambiante.....	4
Puissance.....	4
Consigne de température.....	4
Allumage.....	4
Extinction.....	5
Intégration Jeedom.....	6
Intégration matérielle.....	6
Port série du contrôleur.....	7
Brochage du connecteur JP8.....	7
Brochage de l'ESP8266.....	8
Interconnexion des ports série.....	8
Configuration ESPEasy.....	9
Onglet Config.....	10
Onglet Controllers.....	11
Onglet Hardware.....	12
Onglet Devices.....	13
Développement des commandes.....	14
Socket TCP/IP.....	15
Passage d'argument en python.....	15
Divers.....	16

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



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.

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 &. Pour plus de clarté, les commandes décrites dans ce document ne contiennent pas ces deux caractères.

Exemple de trame : <ESC>RDA00067<&>

Lorsqu'une commande qui est envoyée ne nécessite pas de valeur de retour, une réponse d'acquiescement 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:RDA909005f

Cette commande renvoie une trame du même format que les commandes. Les valeurs de retour sont les suivantes :

Valeur	État
• 0802002a	Refroidissement – Lorsqu'on éteint le poêle, il y a une phase de refroidissement pendant laquelle on ne peut rien faire.
• 08010029	Refroidissement - idem commande ci-dessus
• 00f9003f	Éteint – Le poêle est éteint, près à être allumé
• 01010022	Allumage – Phase d'allumage du foyer
• 00000000	Allumé – Le poêle est en mode de chauffage.

Température ambiante

Valeur : RD100057

Cette commande renvoie une trame du même format que les commandes. 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

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

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.

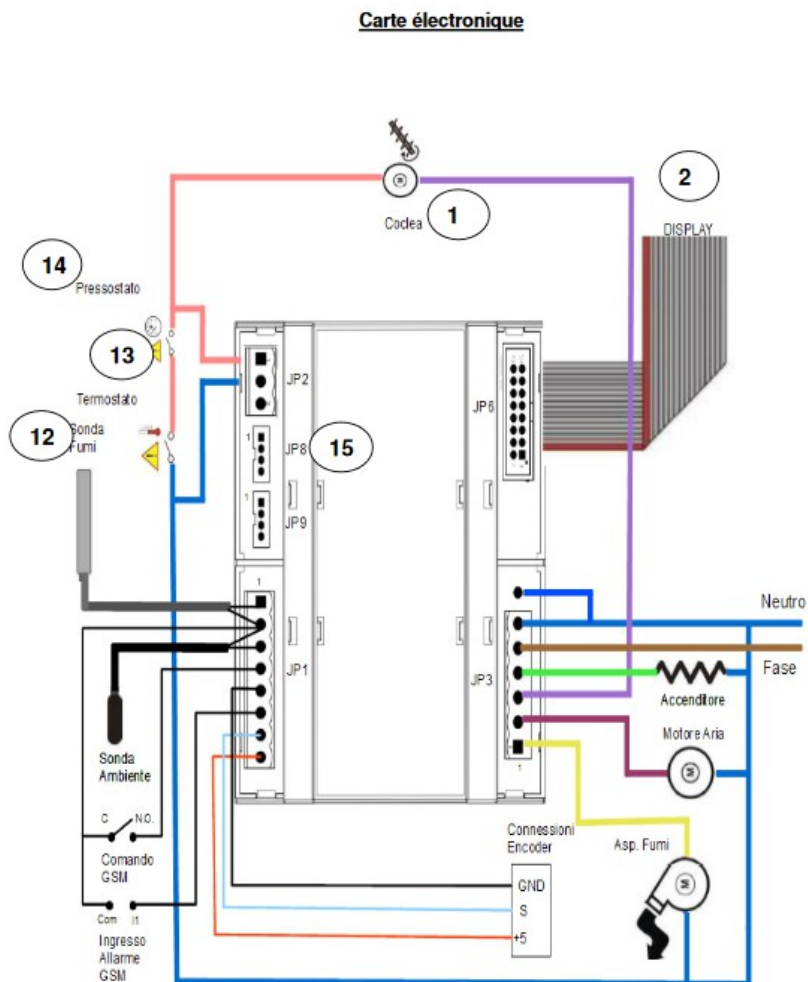
Intégration matérielle

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.

Port série du contrôleur

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



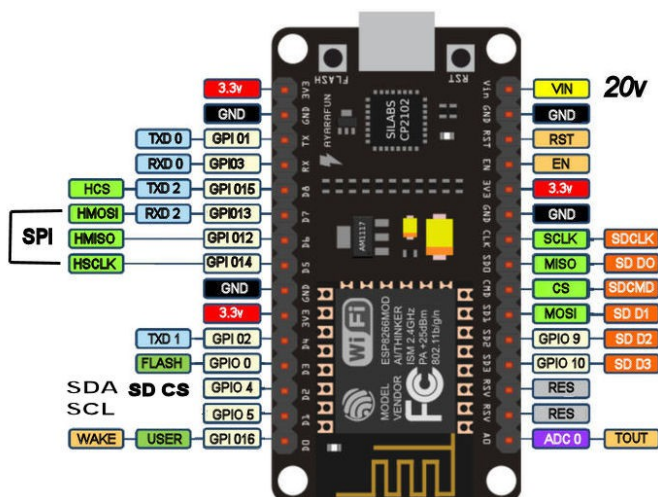
1 : vis sans fin ; 2 : display ; 3 : neutre ; 4 : phase ; 5 : bougie ; 6 : ventilateur air ; 7 : extracteur des fumées ; 8 : connexions encodeur, 9 : accès alarme GSM ; 10 : commande GSM ; 11 : sonde ambiante ; 12 : sonde fumées ; 13 : thermostat ; 14 : pressostat ; 15 : JP8 branchement câble dispositif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

Brochage du connecteur JP8

Connecteur JP8.

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

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 pôle (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 contrôleur 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.**

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.

The screenshot shows the ESPEasy web interface for a device named 'ESP Easy Mega: Poele'. At the top, there are navigation tabs: 'Main' (selected), 'Config', 'Controllers', 'Hardware', 'Devices', 'Notifications', and 'Tools'. Below the tabs is a 'System Info' section with the following data:

Unit Number:	0
Git Build:	mega-20200608
Local Time:	No system time source
Uptime:	1 days 3 hours 0 minutes
Load:	6.54% (LC=3849)
Free Mem:	17216 (34000 -)
Free Stack:	3712 (3824 -)
IP Address:	192.168.1.24
RSSI:	-63 dB (Freebox-5347E3_EXT)

Below the system info is a 'More info' button. At the bottom of the screenshot, there is a 'Node List' section with a header containing 'Node List' and 'Name'. The footer of the page reads 'Powered by Let's Control It community'.

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

Onglet Config

ESP Easy Mega: Poele

[Main](#) **[Config](#)** [Controllers](#) [Hardware](#) [Devices](#) [Notifications](#) [Tools](#)

Main Settings

Unit Name:

Unit Number:

Append Unit Number to hostname:

Admin Password:

Wifi Settings

SSID:

WPA Key:

Fallback SSID:

Fallback WPA Key:

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

Onglet Controllers

ESP Easy Mega: Poeele

△Main ⊙Config **Controllers** 📌Hardware 🖱️Devices 📧Notifications 🛠️Tools

Controller Settings

Protocol: ?

Locate Controller:

Controller IP:

Controller Port:

Controller Queue

Minimum Send Interval: [ms]

Max Queue Depth:

Max Retries:

Full Queue Action:

Check Reply:

Client Timeout: [ms]

Credentials

Use Extended Credentials:

Controller User:

Controller Password:

Controller Subscribe:

Controller Publish:

Enabled:

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 contrôleur Jeedom, le port ainsi que le champ Controller Publish. Ces données se trouvent sur la page Configuration du plugin EspEasy.

Onglet Hardware

ESP Easy Mega: Poele

▾Main ▾Config ▾Controllers **Hardware** ▾Devices ▾Notifications

Wifi Status LED

GPIO → LED:

Inversed LED:

Note: Use 'GPIO-2 (D4)' with 'Inversed' checked for onboard LED

Reset Pin

GPIO ← Switch:

Note: Press about 10s for factory reset

I2C Interface

GPIO ⇌ SDA:

GPIO → SCL:

Clock Speed: [Hz]

Note: Use 100 kHz for old I2C devices, 400 kHz is max for most.

SPI Interface

Init SPI:

Note: CLK=GPIO-14 (D5), MISO=GPIO-12 (D6), MOSI=GPIO-13 (D7)
Note: Chip Select (CS) config must be done in the plugin

GPIO boot states

Pin mode GPIO-0 (D3) ⚠:	<input type="text" value="Default"/>
Pin mode GPIO-1 (D10) TX0:	<input type="text" value="Default"/>
Pin mode GPIO-2 (D4) ⚠:	<input type="text" value="Default"/>
Pin mode GPIO-3 (D9) RX0:	<input type="text" value="Default"/>
Pin mode GPIO-4 (D2):	<input type="text" value="Default"/>
Pin mode GPIO-5 (D1):	<input type="text" value="Output High"/>
Pin mode GPIO-9 (D11) ⚠:	<input type="text" value="Default"/>
Pin mode GPIO-10 (D12) ⚠:	<input type="text" value="Default"/>
Pin mode GPIO-12 (D6):	<input type="text" value="Default"/>
Pin mode GPIO-13 (D7):	<input type="text" value="Default"/>
Pin mode GPIO-14 (D5):	<input type="text" value="Default"/>
Pin mode GPIO-15 (D8) ⚠:	<input type="text" value="Default"/>

Onglet Devices

ESP Easy Mega: Poele

◁Main ◯Config ☰Controllers ★Hardware 🖱️Devices 📧Notifications 🛠️Tools

Task	Enabled	Device	Name	Port	Ctrl (IDX)	GPIO	State
Edit	1	✓	Communication - Serial Server	Marina			
Edit	2	✓	Switch input - Switch	Dummy	1 (1)	GPIO-13	State:
Add	3						

ESP Easy Mega: Poele

◁Main ◯Config ☰Controllers ★Hardware 🖱️Devices 📧Notifications 🛠️Tools

Task Settings

Device: Communication - Serial Server ? i

Name:

Enabled:

TCP Port:

Baud Rate:

Data bits:

Parity:

Stop bits:

TX Enable Pin:

Reset target after boot:

RX Receive Timeout (mSec):

Event processing:

[Close](#) [Submit](#) [Delete](#)





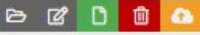

Powered by [Let's Control It](#) community

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:

Nom	Type script	Type	Requête	Options
Icone Statut Aucune	Script	Info Autre	 /var/www/html/plugins/script/data/statut.py	
Icone Eteindre Aucune	Script	Action Défaut	 /var/www/html/plugins/script/data/eteindre.py	
Icone Allumer Aucune	Script	Action Défaut	 /var/www/html/plugins/script/data/allumer.py	
Icone Puissance Aucune	Script	Action Curseur	 /var/www/html/plugins/script/data/puissance.py #slider#	
Icone Consigne Aucune	Script	Action Curseur	 /var/www/html/plugins/script/data/consigne.py #slider#	
Icone Température Aucune	Script	Info Numérique	 /var/www/html/plugins/script/data/temperatue.py	

Chaque commande fait appel à un script en python dont le nom est mentionné dans le champ Requête. 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 :

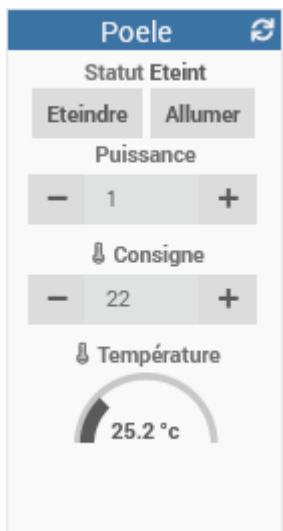
```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
sock.connect(("192.168.1.18", 2000)) #adresse IP et numéro de port Ser2Net
data = "\x1bRD90005f&" #exemple de donnée, x1b est la valeur en hexa du caractère « ESC ».
sock.send(data.encode()) # envoi des données
```

#Exemple de lecture de la réponse :

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

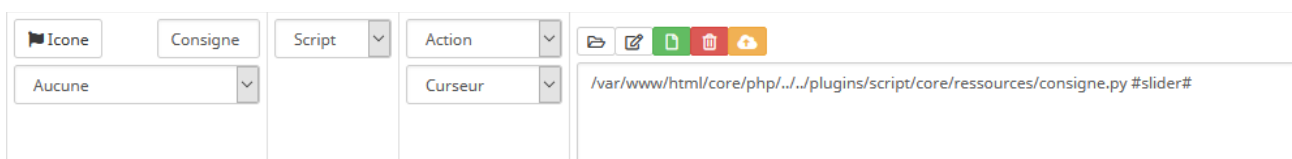
Passage d'argument en python

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.
```

Divers

J'ai remarqué pendant mes tests Jeedom que les commandes sont exécutées à la sauvegarde s'il s'agit d'Info. A l'inverse, dans le cas de commande du type Action, le code n'est pas exécuté à la sauvegarde.

Il est intéressant de typer toutes les commandes en Info, car on peut ainsi retourner des données de debug. L'inconvénient, c'est qu'à la sauvegarde, toutes les commandes sont exécutées, donc attention aux allumages répétés du poêle pendant le debug.