

# Tasmota



[Tasmota doc EN](#)

Firmware open source pour les appareils ESP

1. -Contrôle local total avec configuration et mises à jour rapides.
2. -Contrôle à l'aide de MQTT, Web UI, HTTP ou série.
3. -Automatisez à l'aide de minuteries, de règles ou de scripts.
4. -Intégration avec des solutions domotiques.
5. -Incroyablement extensible et flexible.

## Compilation tasmota , pour la personnalisation du firmware

[personnalisation du firmware](#)

## Commencer

### Conditions préalables~Matériel nécessaire~

#### Dispositif ESP~

Chaque appareil basé sur le chipset Espressif ESP8266, ESP8285, ESP32, ESP32-S ou ESP32-C3 peut être flashé avec Tasmota. Le terme ESP fait référence à l'un d'entre eux.

#### Adaptateur série vers USB~

La puissance fournie à l'appareil est l'un des éléments les plus importants pour le flashage de l'appareil et pour un fonctionnement stable. Vous devez vous assurer que l'appareil reçoit une alimentation suffisante (courant ET niveau de tension approprié) pour flasher correctement le micrologiciel sur l'appareil.

1. -CONSEILLÉ CH340G est le plus fiable et le moins cher pour démarrer ( 1 , 2 ).
2. -FTDI FT232 - ces adaptateurs ont beaucoup de contrefaçons sur le marché, alors achetez

uniquement auprès de sources fiables ( exemple ). N'achetez que la variante avec un régulateur 3,3 V séparé sur PCB !

3. -CP2102 ou PL2303 - fonctionne avec certains appareils, mais l'utilisation d'une alimentation externe de 3,3 V peut être nécessaire. Déconseillé aux débutants !
4. -NodeMCU Vous pouvez également utiliser un NodeMCU (ou similaire) comme adaptateur série vers USB fiable si vous désactivez l'ESP intégré en reliant GND à la broche RST ou EN, et connectez TX et RX directement à un autre ESP82xx au lieu de les croiser.
5. -RaspberryPi - uniquement pour les utilisateurs avancés. Alimentation externe 3.3V nécessaire.

N'oubliez pas d'installer les pilotes de votre adaptateur série vers USB.

## **Danger**

Certains adaptateurs peuvent être commutés entre 3,3 V et 5 V pour les broches de données, mais fournissent toujours 5 V sur la broche d'alimentation, ce qui détruira irrémédiablement votre appareil. Vous DEVEZ vous assurer que les broches de données (RX et TX) et VCC sont réglées sur 3,3 V.

Certains appareils, tels que les produits NodeMCU , D1 mini ou M5Stack , ont un port de téléchargement USB et l'adaptateur série-USB intégré.

De nombreux appareils CH340G ne fonctionneront pas, le "golden CH340G" a un régulateur de tension pour fournir suffisamment de puissance à l'appareil ESP - beaucoup ne le font pas. L'ESP nécessite au moins 150 mA, de nombreux programmeurs série 3,3 V ne fournissent pas autant de courant car de nombreuses tâches de programmation série ne nécessitent pas une grande quantité d'énergie.

Lorsque vous utilisez une alimentation externe de 3,3 V, assurez-vous simplement que la masse (GND) des deux est connectée, cela garantit une masse commune. Une alimentation PC peut être une source d'alimentation 3,3 V CC.

## **Outils de soudure**

Pour souder, vous aurez bien sûr besoin d'un fer à souder, d'un étain à souder et d'un peu de flux. Si vous débutez dans la soudure, regardez quelques vidéos de didacticiel de soudure pendant que vous y êtes.

Si vous êtes intimidé par la soudure, des gabarits imprimés en 3D sont disponibles pour différents modules et appareils. Au pire, vous pourriez vous en tirer en tenant fermement les en-têtes avec des fils de raccordement dans les trous de broche pendant le clignotement, mais ce n'est pas un processus infaillible et le clignotement peut échouer.

## **Fils de cavalier**

Vous pouvez utiliser n'importe quel type de fil, mais les fils de raccordement (également appelés fils DuPont) sont plus pratiques que le soudage et le dessoudage.

## En-têtes de broches~

Les en-têtes de broches sont disponibles en version mâle ou femelle. Choisissez en fonction de vos connecteurs de fil de cavalier.

## Ordinateur avec Linux, Windows ou MacOS

Vous avez besoin d'un ordinateur avec un port USB pour télécharger le firmware sur votre appareil et le configurer.

## Téléphone intelligent

Tasmota installé à partir d'un binaire précompilé doit être configuré pour fonctionner avec votre réseau Wi-Fi avant de pouvoir accéder à l'interface utilisateur Web Tasmota. Cela se fait généralement en se connectant à un point d'accès Wi-Fi Tasmota avec votre smartphone (ou tablette ou ordinateur avec Wi-Fi).

## Logiciel nécessaire

### Binaire du micrologiciel Tasmota

Téléchargez un fichier binaire du micrologiciel Tasmota (.bin). Si vous ne savez pas quel binaire est le bon pour vous, commencez simplement par **tasmota.bin** ou consultez [le tableau des versions](#) pour voir les fonctionnalités dont vous avez besoin.

Les binaires de version officiels peuvent être téléchargés à partir [du serveur de micrologiciel ESP8266 FR](#) .[Binaires tasmota ESP32 EN](#)

Les derniers binaires de branche de développement sont [disponibles uniquement à partir de notre serveur OTA](#) . Le dernier code de développement fusionné est compilé toutes les heures.

### Outils Flashing

1. [-Tasmota Web Installer](#) - flashez Tasmota à l'aide d'un navigateur basé sur Chrome pour ESP82XX et ESP32
2. [-Tasmotizer](#) - outil de flashage et de téléchargement du firmware pour ESP82XX uniquement . (Windows, Linux ou Mac)
3. [-ESP-Flasher](#) - Flasher GUI pour Tasmota basé sur esptool.py pour ESP82XX et ESP32. (Windows, Linux ou Mac)
4. [-Esptool.py](#) - l'outil de clignotement officiel d'Espressif pour ESP82XX et ESP32. (Nécessite Python)



Outils de compilation (facultatif)



Si vous souhaitez modifier le code ou les paramètres par défaut et [compiler votre propre firmware Tasmota](#) .

## Connaissance MQTT

Tasmota est conçu pour être contrôlé et communiquer via [MQTT](#) . Pour l'utiliser au maximum de son potentiel, vous avez besoin [d'un courtier MQTT](#) .

Lisez notre [article sur MQTT](#) pour savoir pourquoi il est essentiel dans Tasmota.

[Doc FR MQTT](#)

## Config Reseau

/etc/moquitto/mosquito.conf

```
#listener 1883
port 1883
allow_anonymous true
```

et ouvrir les ports sur le raspberry

```
sudo ufw allow 1883
sudo ufw allow 1880
sudo ufw allow 22
```

## Préparation du matériel

Nous devons nous connecter à l'interface de programmation série de la puce ESP. Cela se fait en connectant nos broches TX et RX de convertisseur série-USB aux broches ESP RX et TX et en alimentant la puce avec les broches 3,3 V et GND.

Dans la plupart des cas, ces broches sont disponibles sur le circuit imprimé sous la forme de trous de broches ou de pastilles de soudure, mais les en-têtes de broches ou les fils de raccordement doivent être soudés ou appliqués d'une autre manière. Dans certains cas, vous devrez souder des fils directement sur les broches de la puce, ce qui nécessite une certaine expérience et un bon équipement de soudure.



- **NE CONNECTEZ PAS D'APPAREILS AU SECTEUR LORSQUE LE COUVERCLE EST OUVERT ET LE CIRCUIT IMPRIMÉ EXPOSÉ !!!**
- **N'ESSAYEZ JAMAIS DE FLASH LORSQUE VOTRE APPAREIL EST CONNECTÉ AU SECTEUR !!!**
- **VOUS POUVEZ ETRE ELECTROCUTE SI VOUS NE SAVEZ PAS CE QUE**



## VOUS FAITES !

Si vous ne faites pas attention, votre propre santé sera en danger. Court-circuiter votre interface série avec l'alimentation secteur fera frir votre appareil et votre adaptateur série et endommagera ou détruira également votre ordinateur. Il est important que tous les câbles d'alimentation secteur soient toujours déconnectés de l'appareil lors de la connexion en série ou même lorsque le boîtier de l'appareil est ouvert.

### Connexion série

Chaque appareil a ses broches étiquetées différemment. Si l'étiquetage n'est pas visible sur le circuit imprimé, veuillez vous reporter au guide de clignotement des appareils ou rechercher sur Internet les emplacements corrects des broches. Les instructions et restrictions spécifiques aux appareils sont documentées dans le référentiel des appareils pris en charge par Tasmota . Les brochages des modules Wi-Fi couramment utilisés se trouvent ici

Lorsque vous avez identifié les broches sur votre appareil, connectez les fils selon le tableau :

Adaptateur série	Dispositif ESP
3V3	3V3 ou VCC
TX	RX
RX	TX
Terre	Terre



Notez que TX de votre adaptateur va à RX sur l'appareil ESP et RX de l'adaptateur va à TX sur l'appareil !

## Mode de programmation

### Emplacement GPIO0 typique

L'ESP doit être mis en mode de programmation ou en mode flash avant que le micrologiciel puisse être téléchargé. Cela se fait en connectant la broche GPIO0 à GND pendant le démarrage de la puce.

Sur de nombreux appareils, le bouton de commande installé est connecté à GPIO0 et GND, ce qui facilite l'entrée en mode de programmation. Sur d'autres, vous devrez ponter les broches sur le PCB ou directement sur la puce avec un fil de liaison. Les emplacements GPIO0 pour les modules populaires peuvent être trouvés dans [Pinouts](#) !

Les instructions spécifiques à l'appareil sont documentées dans [Tasmota Supported Devices Repository](#) .

Pour mettre l'ESP en mode programmation :

1. -Débranchez l'adaptateur série vers USB et l'alimentation

2. -Pont GPIO0 et GND (en appuyant sur le bouton intégré ou en connexion avec un fil)
3. -Connectez l'adaptateur série-USB à votre ordinateur
4. -Après quelques secondes, déconnectez GPIO0 de GND (relâchez le bouton ou retirez la connexion filaire). Sur les appareils qui ne fournissent pas le bouton connecté GPIO0, il peut être plus facile de laisser le pont filaire en place tout au long du processus de clignotement (effacement et téléchargement). Cela ne créera aucun problème. Une fois le micrologiciel téléchargé avec succès, retirez le pont. Cela permet à l'appareil de démarrer normalement.

### Test du mode de programmation esptool.py

Vous pouvez tester si votre appareil est en mode de programmation en essayant de lire les informations de la puce ESP82xx. Cela nécessite esptool.py. Les instructions d'installation et d'utilisation esptool sont fournies ci-dessous. Par exemple (COM5 sera votre port COM):



```
esptool.py -p COM5 read_mac (Il devrait lire l'adresse MAC. Il peut ensuite échouer lors du téléchargement et de l'exécution d'un "stub". C'est normal.)  
esptool.py -p COM5 flash_id
```

Si tout s'est bien passé, vous êtes maintenant en mode de programmation et prêt à continuer à clignoter. Si le processus de clignotement ne peut pas démarrer, déconnectez l'appareil et réessayez les étapes.

### Erreurs fréquentes

1. -Connexions de fils et joints de soudure - Vérifiez à nouveau toutes les connexions et vérifiez également le débordement de soudure.
2. -Utiliser un câble USB **câble de données**- Certains câbles USB servent uniquement à charger et ne connectent pas les lignes de données nécessaires pour charger le micrologiciel sur l'appareil.
3. -Alimentation insuffisante fournie via l'adaptateur série-USB. Cela conduit à des échecs de clignotement ou à un flash corrompu. Fournissez plus de puissance avec une alimentation séparée de 3,3 V ou procurez-vous un adaptateur avec une meilleure alimentation. Assurez-vous que toutes les tensions CC utilisent la même ligne GND.
4. -Revérifiez votre adaptateur série vers USB afin de vous assurer qu'il fournit une tension de 3,3 V et NON de 5 V. 5V endommagera la puce ESP !
5. -Relâcher le bouton/fil GPIO0 avant la fin du démarrage - Il est prudent de laisser GPIO0 connecté à GND pendant tout le processus de programmation (effacement et téléchargement). Assurez-vous simplement de retirer le pont GPIO0 vers GND avant de démarrer l'appareil pour un fonctionnement normal.
6. -Assurez-vous que la broche RX est connectée à la broche TX entre l'adaptateur série et votre appareil ESP, et vice versa.
7. -Effacez d'abord le flash et redémarrez avant de télécharger le binaire du micrologiciel Tasmota. Ne pas effacer peut laisser des restes du contenu flash précédent qui peuvent interférer avec le nouveau fonctionnement du micrologiciel.

## Flashing

Si vous avez suivi la préparation du matériel , votre appareil doit être en mode de programmation et prêt pour l'installation d'un binaire du micrologiciel Tasmota.

Vous voudrez peut-être sauvegarder le micrologiciel du fabricant de l'appareil sur une chance sur un million que vous n'aimez pas Tasmota.

## Choisissez une méthode d'installation

1. -Installeur Web
2. -tasmotiseur !
3. -esptool.py
4. -Conversion OTA

### Installeur Web

Flashez Tasmota directement depuis votre navigateur Web.

Votre navigateur ne prend pas en charge l'installation d'éléments sur les appareils ESP. **Utilisez Google Chrome ou Microsoft Edge.**



Une fois l'installation réussie, cliquez sur SUIVANT (NEXT). Le programme d'installation recherchera les réseaux Wi-Fi et sélectionnera le plus puissant. Utilisez la liste déroulante pour sélectionner entre les réseaux.

### Configure Wi-Fi

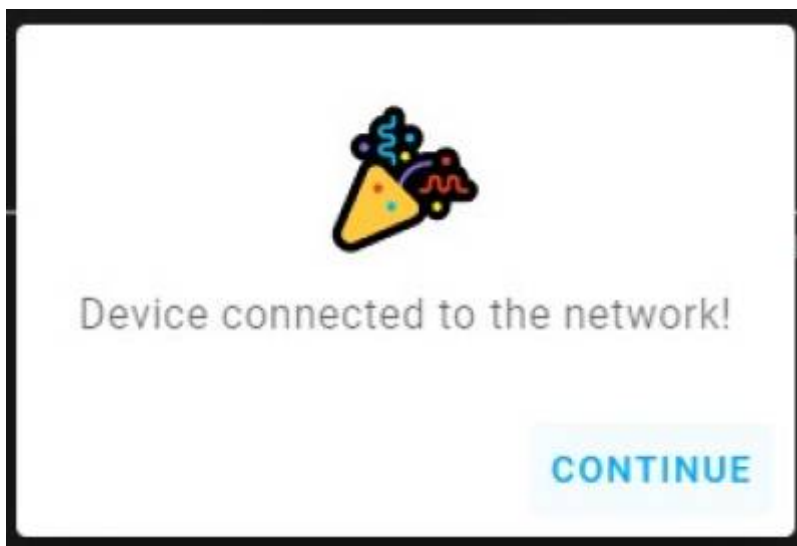
Enter the credentials of the Wi-Fi network that you want your device to connect to.

Network  
Numero

Password

SKIP CONNECT

Entrez le mot de passe Wi-Fi.



Attendez que la configuration soit terminée et cliquez sur CONTINUER



À partir de là, cliquez sur VISIT DEVICE pour accéder à l' [interface Web](#) pour une configuration plus poussée.

### Après configuration

Votre appareil exécutant Tasmota est maintenant prêt à être [contrôlé](#) .

Découvrez toutes les [fonctionnalités](#) de Tasmota et les moyens de l' [intégrer à d'autres plates-formes](#).

#### Avertissement



Si vous rencontrez des fluctuations de puissance dans votre réseau électrique, il est préférable de désactiver immédiatement la fonction de [récupération du cycle d'alimentation](#) avec la commande **SetOption65 1** immédiatement, sinon vous risquez de vous retrouver avec des paramètres par défaut du micrologiciel sur votre appareil.

## Interface Web

### WebUI

L'interface utilisateur Web de Tasmota est un moyen pratique de contrôler et de gérer votre appareil tasmotisé.



#### Avertissement



WebUI n'a pas et ne peut pas avoir toutes les fonctionnalités et commandes implémentées. Pour un contrôle précis et complet, utilisez les commandes Console ou MQTT !

Pour accéder à l'interface Web, utilisez l'adresse IP de votre appareil dans votre navigateur Web préféré.

Par défaut, WebUI démarre en mode administrateur non protégé, ce qui permet un accès complet à votre appareil à toute personne ayant accès à cette adresse IP.

Si vous souhaitez limiter les autres utilisateurs au contrôle uniquement via l'interface Web, utilisez [WebServer 1](#)

La commande [WebPassword](#) configurera un mot de passe qui doit être saisi lors de la tentative d'accès à WebUI. Il s'agit d'une très fine couche de protection puisque tout est transmis via le protocole HTTP non crypté.

## Menus



### Configuration

Le menu de configuration vous permet de tout configurer, des composants au Wi-Fi, et vous offre la possibilité de sauvegarder et de restaurer la configuration dans un emplacement sécurisé.

### Informations

Affiche une seule page chargée d'informations sur l'appareil, y compris : la version actuelle de Tasmota, les données du point d'accès Wi-Fi, les données de l'hôte MQTT, etc.

## Mise à jour du firmware

Un menu facile à utiliser pour lancer une mise à niveau du micrologiciel à partir d'un fichier téléchargé .bin ou d'un serveur OTA.

## Console

Accès terminal à Tasmota. Émettez des commandes ici ou suivez le flux d'informations. Utile pour le débogage lors de l'utilisation de `Weblog 4` .

## Personnalisation

### Bouton Web

Modifiez les noms des boutons bascule à l'aide de [WebButton](#)

### Thèmes

WebUI peut être thématisé à l'aide de [WebColor](#) commandes.

Pour appliquer un thème ci-dessous, copiez l'intégralité du bloc de code et envoyez-le à l'aide de la console ou via MQTT.

### Dark (default theme)



### 001.json

```
WebColor
{"WebColor": "#eaeaea", "#252525", "#4f4f4f", "#000000", "#dddddd", "#65c115",
"#1f1f1f", "#ff5661", "#008000", "#ffffff", "#1fa3ec", "#0e70a4", "#d43535",
"#931f1f", "#47c266", "#5aaf6f", "#ffffff", "#999999", "#eaeaea" ]}
```

### Light (default until 6.7.1.)



### 002.json

```
WebColor
{"WebColor":["#000000","#ffffff","#f2f2f2","#000000","#ffffff","#000000",
"#ffffff","#ff0000","#008000","#ffffff","#1fa3ec","#0e70a4","#d43535",
"#931f1f","#47c266","#5aaf6f","#ffffff","#999999","#000000"]}
```

### Halloween



### 003.json

```
WebColor
{"WebColor":["#cccccc","#2f3133","#3d3f41","#dddddd","#293134","#ffb000",
"#293134","#ff5661","#008000","#ffffff","#ec7600","#bf5f00","#d43535",
"#931f1f","#47c266","#5aaf6f","#ffffff","#999999","#bc4d90"]}
```

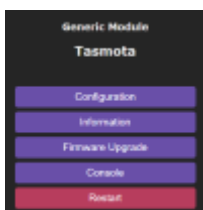
### Navy



### 004.json

```
WebColor
{"WebColor":["#e0e0c0","#000033","#4f4f4f","#000000","#dddddd","#a7f432",
"#1e1e1e","#ff0000","#008000","#ffffff","#1fa3ec","#0e70a4","#d43535",
"#931f1f","#47c266","#5aaf6f","#ffffff","#999999","#eedd77"]}
```

## Purple Rain



### 005.json

```
WebColor
{"WebColor":["#eaeaea", "#252525", "#282531", "#eaeaea", "#282531", "#d7ccff", "#1d1b26", "#ff5661", "#008000", "#faffff", "#694fa8", "#4d3e7f", "#b73d5d", "#822c43", "#1f917c", "#156353", "#faffff", "#716b7f", "#eaeaea"]}
```

## Solarized Dark



### 006.json

```
WebColor
{"WebColor":["#839496", "#002b36", "#073642", "#839496", "#002b36", "#839496", "#073642", "#b58900", "#859900", "#eee8d5", "#268bd2", "#185886", "#dc322f", "#90211f", "#859900", "#647300", "#839496", "#073642", "#839496"]}
```

### Tip



Lorsque vous utilisez un client MQTT tel que mosquitto\_pub, placez la charge utile du message entre guillemets simples ( ' )

### 007.json

```
mosquitto_pub -h 192.168.1.20 -t "cmd/myTopic/WebColor" -m '{"WebColor":["#eaeaea", "#252525", "#4f4f4f", "#000000", "#dddddd", "#65c115", "#1f1f1f", "#ff5661", "#008000", "#faffff", "#1fa3ec", "#0e70a4", "#d43535", "#931f1f", "#47c266", "#5aaf6f", "#faffff", "#999999", "#eaeaea"]}'
```

# Commandes

## Commandes

Tasmota fournit une puissante interface de contrôle à l'aide de commandes

Les commandes peuvent être émises à l'aide de MQTT , de requêtes Web , de la console WebUI et de la série Comment utiliser les commandes~

Chaque commande utilisée sans paramètre (charge utile) renvoie le paramètre actuel .

1. **-Power** renvoie l'état de la première sortie de puissance définie (généralement Relay1)

Au lieu de 0vous pouvez utiliser offou falseet au lieu de 1vous pouvez utiliser onou true.

1. **-Power ONactive** la première sortie d'alimentation définie (généralement Relay1)
1. **-Power 1 active** également la première sortie d'alimentation définie (généralement Relay1)
1. **-Power1 Trueactive** également la première sortie d'alimentation définie (généralement Relay1)

Remplacez <x>dans une commande par le numéro d'index approprié. Laissez-le vide pour utiliser le premier disponible.

1. **-Power1 et Power** les deux contrôlent la première sortie de puissance définie (généralement Relay1)

Dans les commandes avec x..y des paramètres de valeur, utilisez un nombre allant x de à y.

Lorsqu'une commande mentionne la réinitialisation sur "firmware default" , cela signifie que le paramètre reviendra à celui du fichier binaire flashé. Si vous avez utilisé user\_config\_override.hau moment de la compilation, il reviendra à ceux-ci.

## Noter

Toutes les commandes sont standard sous la forme : COMMAND< INDEX> DATAIl n'y a qu'une seule SPACEentre INDEXet DATA. (Il n'y a pas de commande dans Tasmota qui autorise le =signe)

## Exemple

Définissez les adresses IP réseau (XXX.XXX.XXX.XXX) à l'aide de la commandeIPAddress<x>:IPAddress1 192.168.1.123

Le =signe (Il peut être compris comme le signe :) est utilisé uniquement pour signifier la référence d'explication :IPAddress1 = set device IP address

## Noter

Outre les résultats initiés par une commande (synchrone), vous pouvez obtenir des résultats asynchrones initiés par un déclencheur de règle, un événement de télémétrie, des commandes provenant d'une autre source ou des valeurs d'appareil modifiées. En termes simples, d'autres messages peuvent précéder les messages publiés à la suite de vos commandes.

## Exemple

Un `tele/%topic%/STATUSmessage` (envoyé toutes les 300 secondes par défaut) peut apparaître exactement après l'émission de la `Power off` commande et avant la réception du `stat/%topic%/RESULT = {"POWER":"OFF"}` message. avec MQTT~

Pour envoyer des commandes et afficher les réponses, vous aurez besoin d'un client MQTT .

Les commandes via MQTT sont émises en utilisant le sujet `cmnd/%topic%/<command>` et la charge utile `<parameter>` où `%topic%` est le sujet de l'appareil auquel vous envoyez la commande. S'il n'y en a pas `<parameter>` (un message/charge utile MQTT vide), une requête est envoyée pour connaître l'état actuel du fichier `<command>`.

Voir l'article [MQTT](#) pour en savoir plus.

## Avec les requêtes Web

Les commandes peuvent être exécutées via des requêtes Web (HTTP), par exemple :

`http:<ip>/cm?cmnd=Power%20TOGGLE` `http:<ip>/cm?cmnd=Power%20On`

`http:<ip>/cm?cmnd=Power%20off`

`http:<ip>/cm?user=admin&password=joker&cmnd=Power%20Toggle`

Tous les espaces ou caractères spéciaux doivent être remplacés par leurs codes hexadécimaux ASCII

Vous devez faire précéder chaque code hexadécimal de `%`. Les codes les plus utilisés sont : `space=%20` et  `; = %3B`.

## Pointe

Utilisez [URLencoder.org](http://URLencoder.org) pour convertir facilement vos commandes.

Si vous avez défini un mot de passe pour l'accès à l'interface utilisateur Web, celui-ci doit être inclus (en clair) dans l'URL de la requête HTTP, comme suit :

[008.html](#)

```
http://<ip>/cm?user=<username>&password=<password>&cmnd=Power%20On
```

## dans la console dans l'interface utilisateur Web

Le menu de la console dans l'interface utilisateur Web est un endroit pratique pour envoyer des commandes et il se comporte comme une connexion de terminal via un pont série.

### Avertissement

Les contrôles de l'interface graphique n'ont pas et ne peuvent pas avoir toutes les fonctionnalités et commandes implémentées. Pour un contrôle précis et complet, utilisez les commandes de la console ! sur pont série~

Si vous avez flashé l'appareil via la méthode série, vous pouvez vous y connecter avec une application de terminal (par exemple Termite ou Arduino IDE Serial Monitor) pour émettre des commandes et suivre les réponses. Il s'agit d'un moyen pratique de Backlogconfigurer votre nouvel appareil.

L'interface série est réglée sur 115200 bps sauf pour les appareils qui nécessitent un débit en bauds différent le pouvoir du backlog~

BacklogLa commande permet d'exécuter jusqu'à 30 commandes consécutives avec une seule ligne de commande. Chaque commande est séparée par un point-virgule (;). Backlogest une fonctionnalité utile pour éviter de nombreux redémarrages lors de la configuration d'un nouvel appareil. Vous pouvez l'utiliser pour :

### Configurer les deux points d'accès Wi-Fi

```
Backlog SSID1 <myssid>; Password1 <mypassword>; SSID2 <myssid2>; Password2 <mypassword2>
```

Configurez l'adresse du courtier MQTT, les informations d'identification MQTT, le sujet de l'appareil et activez quelques options personnalisées

```
Backlog MqttHost <yourhost>; MqttUser <user>; MqttPassword <password>; Topic <customtopic>; SetOption53 1; PowerRetain on
```

Pour un contrôle de puissance spécifique, en utilisant le backlog comme un script

```
Backlog Status 1; Power2 on; Delay 20; Power2 off; Status 4
```

Lors de l'utilisation de requêtes Web (n'oubliez pas d'encoder "espace" en '%20' et ";" en '%3B')

### 009.html

```
http://<ip>/cm?user=admin&password=joker&cmd=Backlog%20Power%20Toggle%3BPower1%20off
```

Une Backlog commande sans argument efface une éventuelle file d' Backlogattente existante.

## Exemple

en cas de commande Backlog Power1 OFF; Delay 600; Power1 ON, l'utilisation d'une Backlogcommande supplémentaire sans aucun argument dans le délai de 1 minute supprimera toute la file d'attente Power1 OFF; Delay 600; Power1 ON. Par conséquent Power1 ON, la commande ne sera pas exécutée et l'alimentation restera coupée. Liste des commandes~

## Avertissement

Si vous utilisez des versions de Tasmota antérieures à la version actuelle, certaines commandes peuvent ne pas fonctionner. La disponibilité de certaines fonctionnalités et de leurs commandes associées dépend de la version du micrologiciel. Veuillez consulter le tableau des versions pour connaître les fonctionnalités disponibles pour chaque variante de micrologiciel.

## Noter

Presque tous les paramètres utilisant des paramètres de chaîne (sauf Ruleet MqttFingerprint) partagent une zone commune avec un maximum de 698 caractères, c'est-à-dire que la longueur totale de tous ces paramètres est limitée à cette taille (vous serez noté si cette limite est dépassée).

# Listes des Commandes Tasmota

[Liste des commandes](#)

[Liste des commandes Tasmota FR](#)

## Modeles



Le modèle (Template) est une définition d'un périphérique et de la manière dont ses GPIO sont attribués.

Les modèles vous permettent de créer, modifier et partager facilement des configurations pour divers appareils dont les fonctionnalités sont prises en charge par Tasmota.



Tasmota 9.1 a entièrement repensé la disposition des modèles pour permettre une expansion future. En savoir plus sur la conversion GPIO .

Pour faciliter le traitement par Tasmota, un modèle est écrit sous la forme d'une chaîne JSON et ressemble à ceci :

## 010.json

```
{"NAME": "Example  
Template", "GPIO": [416, 0, 418, 0, 417, 2720, 0, 0, 2624, 32, 2656, 224, 0, 0], "FLAG":  
:0, "BASE": 45}
```



Le référentiel des [appareils pris en charge par Tasmota](#) contient une [liste complète des appareils pris en charge](#). Nous encourageons tous ceux qui créent un modèle pour un nouvel appareil inconnu à le soumettre à l'aide du formulaire [Ajouter un nouveau modèle](#) .

### Comment utiliser

Allez dans Configuration - **Configurer le modèle ...**



... et vous finirez par regarder cet écran

## Sonoff Basic Module

# Tasmota

### Template parameters

Name	<input type="text" value="Generic"/>
Based on	<input type="text" value="Generic (18)"/> ▼

---

GPI00	<input type="text" value="User"/> ▼
GPI01	<input type="text" value="User"/> ▼
GPI02	<input type="text" value="User"/> ▼
GPI03	<input type="text" value="User"/> ▼
GPI04	<input type="text" value="User"/> ▼
GPI05	<input type="text" value="User"/> ▼
GPI09	<input type="text" value="User"/> ▼
GPI10	<input type="text" value="User"/> ▼
GPI12	<input type="text" value="User"/> ▼
GPI13	<input type="text" value="User"/> ▼
GPI14	<input type="text" value="User"/> ▼
GPI15	<input type="text" value="User"/> ▼
GPI16	<input type="text" value="User"/> ▼
GPI17	<input type="text" value="User"/> ▼

Il est temps de créer votre modèle. Création de votre modèle~

- 1 -Modifiez le nom du modèle (définit également le nom du module).

### Template parameters

Name	<input type="text" value="RGB Smart Plug"/> <span style="color: green; font-weight: bold; font-size: 2em;">1</span>
Based on	<input type="text" value="BlitzWolf SHP (45)"/> ▼ <span style="color: green; font-weight: bold; font-size: 2em;">2</span>

- 2 -Sélectionnez un module sur lequel BASER votre modèle . Si vous n'êtes pas sûr, **Module 18** c'est le meilleur choix. Dans cet exemple, l'appareil est basé sur le module **Blitzwolf SHP (45)**.
- 3 -Configurez les composants affectés aux GPIO pour qu'ils correspondent à votre appareil. Si

vous ne savez pas quelles broches votre appareil utilise, lisez la nouvelle procédure de configuration de l'appareil pour déterminer les affectations correctes des broches.



- Tout GPIO inutilisé qui ne peut pas avoir de périphérique connecté doit être défini sur None(0). Dans notre exemple, l'appareil n'a pas de GPIO exposés, donc ceux qui ne sont pas utilisés sont définis par Orapport au module BlitzWolf d'origine.
- Les GPIO auxquels des périphériques peuvent être connectés (GPIO exposés) doivent être définis sur User(1). Cela permet une configuration future via la boîte de dialogue Configurer le module sans qu'il soit nécessaire de créer un nouveau modèle.

### Exemple



Sonoff TH a une prise connectée au GPIO4 qui permet à un utilisateur de brancher un capteur. L'attribution de GPIO4 comme User permet à un modèle d'avoir des GPIO corrects pour cet appareil même si rien n'est branché. Mais, lorsqu'un utilisateur décide de connecter un capteur à l'aide de la prise, GPIO4 peut être défini sur le type de capteur via la page Configurer le module.

Cliquez sur Enregistrer et vous verrez ce message



Enfin, l'appareil redémarrera avec le nouveau nom de modèle

**RGB Smart Plug Module**

## Tasmota

Voltage	0 V
Current	0.000 A
Power	0 W
Apparent Power	0 VA
Reactive Power	0 VAr
Power Factor	0.00
Energy Today	0.000 kWh
Energy Yesterday	0.000 kWh
Energy Total	0.000 kWh

**OFF OFF**

Dark Bright

Toggle 1 Toggle 2

Configuration

Information

Firmware Upgrade

Console

Restart

## Périphériques



Les périphériques sont des capteurs, des écrans, des contrôleurs, des LED et d'autres appareils câblés aux broches GPIO disponibles de votre appareil



Un périphérique doit avoir une alimentation correctement câblée, GND et des broches de données sur l'appareil avant le démarrage pour que Tasmota le détecte et



l'initialise correctement.

## Périphériques pris en charge

### Paramètres Tasmota



Tasmota permet une sélection facile des périphériques et une affectation aux broches GPIO.

La configuration est possible dans la page WebUI **Configuration - Configurer le module**

ou en utilisant les commandes :[Module](#) et [GPIO](#), ou [Template](#).

### Module

Sélectionnez d'abord le module souhaité pour l'appareil (attendez le redémarrage). Selon le [type de module](#), seules certaines broches GPIO sont configurables par l'utilisateur. Le module Generic (18) a tous les GPIO configurables.

[Modules](#) affiche les modules pris en charge

### GPIO

Attribuez [un composant](#) à un GPIO.

1. **-GPIO14 2** configure le capteur AM2301 sur GPIO14\_
2. **-Backlog GPIO14 5; GPIO4 6** définit I2C SCL sur GPIO14 et I2C SDA sur GPIO4 Tasmota détectera automatiquement tous les périphériques I 2 C connectés et pris en charge. Si vous avez des adresses I 2 C en conflit, voir [I2CDEVICES](#)

[GPIOs All](#) affiche la liste de tous les composants disponibles par nom et index

Pour qu'un périphérique s'affiche, vous devrez peut-être redémarrer votre appareil au lieu d'un redémarrage progressif. [Template~](#)

Au lieu d'utiliser **Module** et **GPIO** vous pouvez tout définir en utilisant **Template**. [Lire la suite...](#)

# Tasmota esp32 cam

[Esp32 cam et Tasmota](#)

# Tasmota esp32 et servomoteur

[Ouvrir Vanne avec servo moteur](#)

# Piloter une porte de garage avec un esp32 et Tasmota

[Porte de garage avec tasmota et ESP32](#)

# Sommeil profond esp32 et Tasmota

[DeepSleep : sommeil profond de l'ESP32](#)

# Tasmota et LCD

[Tasmota LCD](#)

From:

<https://chanterie37.fr/fablab37110/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link:

<https://chanterie37.fr/fablab37110/doku.php?id=start:esp32:tasmota&rev=1744160737>

Last update: **2025/04/09 03:05**

