

# Esp32 Cam et Tasmota

<https://cgomesu.com/blog/Esp32cam-tasmota-webcam-server/>

<https://makerhelp.fr/tasmota-comment-activer-le-serveur-rtsp-sur-lesp32-cam/>

[https://templates.blakadder.com/ai-thinker\\_ESP32-CAM.html](https://templates.blakadder.com/ai-thinker_ESP32-CAM.html)



Ouvrez un terminal et installez les paquets requis:

```
sudo apt update
sudo apt install wget python3 python3-pip
```

Installer esptool.py via pip3:

```
pip3 install esptool
```

Découvrez si esptool.py peut être trouvé dans votre utilisateur \$PATH.

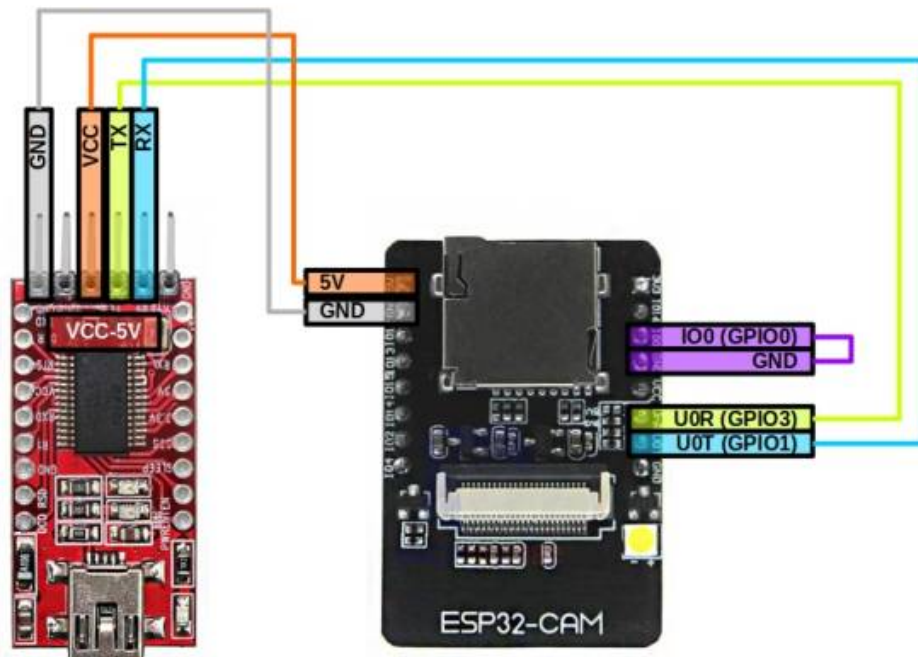
```
whereis esptool.py
```

Si esptool.py n'a pas été trouvé, cela signifie que votre utilisateur .local/bin n'est pas dans votre \$PATH. Ajoutez-le comme suit:

```
echo "export PATH=\"$HOME/.local/bin:$PATH" | tee -a \"$HOME/.bashrc" > /dev/null
```

Connectez votre caméra ESP32 à l'adaptateur USB à TTL/série en mode flash:

## ESP32-CAM wiring: Firmware flash mode (add GPIO0-GND jumper)



Attention. Assurez-vous que votre adaptateur USB vers TTL a VCC en mode 5V et dans l'ESP32, le câble VCC est connecté à la broche 5V. Vérifiez le câblage avant de passer à autre chose.

Connectez l'adaptateur à un port USB sur votre ordinateur et vérifiez le nouvel appareil /dev/:

```
ls -l /dev/ttyUSB*
```

Ajouter votre \$USER au même groupe que /dev/ttyUSB\*(c'est généralement dialout mais si différent, changer dans la commande ci-dessous) et tty:

```
sudo usermod -aG dialout,tty user
```

Déconnectez-vous et revenez. (Si vous continuez à rencontrer des problèmes d'autorisation, essayez plutôt de redémarrer. Vous pouvez vérifier les autorisations de votre utilisateur avec `id ${USER}`.)

## Flashing Tasmota32 serveur webcam

Nous sommes maintenant prêts à flasher le firmware de Tasmota. Pour référence, les informations officielles sont disponibles à l'adresse <https://tasmota.github.io/docs/ESP32>.

## Créer un tasmota32Dir dans /opt:

```
cd /opt
```

```
sudo mkdir tasmota32
```

### Changer la propriété du nouveau répertoire à l'utilisateur actuel plutôt que root:

```
sudo chown ${USER}:${USER} tasmota32/
```

Télécharger le `tasmota32-webcam.factoryb.bi.factoryn` binaire et les binaires ESP32 Tasmota nécessaires du dépôt officiel de Github via `wget`. (Ce qui suit a été mis à jour le 12 août 2021.) Les binaires sont maintenant disponibles dans un dépôt différent de celui d'avant, à savoir `arendst/Tasmota-firmware` et `arendst/Tasmota-firmware`, et actuellement, le nouveau dépôt a une seule branche (`main`). Il existe deux versions de la `tasmota32-webcam.factory.bin`, un de la `release` et un autre de la `development` parties du projet Tasmota32.

Mon conseil est d'essayer la version stable d'abord, puis le développement si vous avez des problèmes, à moins qu'il n'y ait une note dans Changelog qui dit le contraire

Pour télécharger le binaire de version stable, utilisez la commande suivante:

```
wget -P /opt/tasmota32/  
https://ota.tasmota.com/tasmota32/release/tasmota32-webcam.factory.bin
```

Alternativement, pour télécharger le binaire de développement, utilisez la commande suivante:

```
wget -P /opt/tasmota32/  
https://ota.tasmota.com/tasmota32/tasmota32-webcam.factory.bin
```

Assurez-vous que votre caméra ESP32 est connectée à votre ordinateur en mode flash (sauter GPIO0-GND). Maintenant, trouvez le port USB que votre appareil utilise `/dev/` et le mettre à la variable environnementale `ESP_PORT`, comme suit:



Attention. Bien que pratique, la commande suivante suppose qu'il existe un seul adaptateur USB à série connecté à votre ordinateur. Si ce n'est pas le cas, régler manuellement `ESP_PORT` à l'emplacement que votre adaptateur USB utilise actuellement. Vous pouvez trouver le port via `ls /dev/ttyUSB*` et tester un par un jusqu'à ce que vous trouviez celui utilisé par l'adaptateur. Alternativement, il suffit de déconnecter tous les autres adaptateurs USB à série pour cette procédure et continuer.

```
ESP_PORT=$(ls /dev/ttyUSB*)
```

Veuillez noter que cela ne fonctionne que si vous continuez à utiliser la même coquille dans laquelle `ESP_PORT` a été définie. Si vous vous déconnectez ou même fermez le terminal actuel, vous devrez redéfinir `ESP_PORT` pour continuer à l'utiliser.

**Vous pouvez vérifier cela `ESP_PORT` a été correctement définie par echol'ing, comme suit:**

```
echo $ESP_PORT
```

qui devrait produire quelque chose comme ceci:

```
/dev/ttyUSB0
```

Effacez le firmware actuel (ou toutes les données) de votre ESP32-cam.



Attention. La procédure suivante effacera toutes les données sur la caméra ESP32.

```
esptool.py --port $ESP_PORT erase_flash
```



Attendre jusqu'à esptool.py est fait. Ensuite, appuyez sur le bouton de réinitialisation de la caméra ESP32. Maintenant, vérifiez ça \$ESP\_PORT est de nouveau disponible.

**Flash le tasmota32-webcam.factory.bin serveur de webcam binaire et les binaires Tasmota requis à l'ESP32-cam.**

```
esptool.py --chip esp32 \  
--port $ESP_PORT \  
--before default_reset \  
--after hard_reset \  
write_flash -z \  
--flash_mode dout \  
--flash_size detect \  
0x0 /opt/tasmota32/tasmota32-webcam.factory.bin
```



Attendre jusqu'à ce qu' esptool.py est soit complètement chargé avant de passer à autre chose. Télécharger un firmware peut prendre quelques minutes. Si vous rencontrez des problèmes tout en Téléchargeant, essayez un taux de baud différent (-b) que le défaut 115200, comme -b 921600. La FAQ Tasmota peut vous aider à résoudre ce problème et d'autres.



Si esptool.py attend à se connecter..., appuyez sur le bouton Redémarrer (RST) sur votre module ESP-cam.

Attendre jusqu'à esptool.py est soit terminé. Ensuite, déconnectez votre adaptateur USB en TTL de votre ordinateur et retirez le strap en mode flash (GPIO0-GND) de la caméra ESP32.

### ESP32-CAM wiring: Serial monitor



Vous êtes maintenant prêt à alimenter votre nouveau dispositif Tasmota ESP32-cam à l'aide d'une alimentation indépendante.

## Console série

 Utiliser l'IDE Arduino en 115000

## Configuration

### Installation firmware tasmota

Par défaut, une nouvelle installation du firmware Tasmota créera un point d'accès sans fil pour votre ESP32-cam.

Utilisez un appareil compatible wifi (par exemple, ordinateur portable) et connectez-vous à celui-ci.

La caméra ESP32 donnera à votre appareil une adresse IP, que vous pouvez vérifier via

```
ip a.
```

Habituellement, l'adresse IP de l'appareil se trouve dans le **192.168.4.0/24**, qui signifie que l'ESP32-cam webUI est à **192.168.4.1:80**;

Sinon, l'interface Web sera au premier @ip dans la plage à laquelle votre appareil est connecté après avoir rejoint le point d'accès sans fil créé par le micrologiciel Tasmota.

Ouvrez un navigateur Web de votre choix et accédez à l'interface Web ESP32-cam.

Vous devriez être invité à modifier les paramètres wifi pour permettre à votre ESP32-cam de se connecter à votre réseau wifi local.

Modifiez les paramètres, enregistrez-le et attendez que la caméra ESP32 redémarre.

Accédez au serveur DHCP de votre réseau local et trouvez l'adresse IP attribuée à votre caméra ESP32.( [nmap](#) )

À ce stade, c'est une bonne idée d'y attribuer une adresse statique. (Si vous définissez une adresse statique, redémarrez la caméra ESP32 avant de passer à autre chose.)

Accédez à l'interface Web ESP32-cam sur votre réseau local et démarrez le processus de configuration (voir ci-dessous).

## Mise à jour du template

Les modèles Tasmota sont des définitions spécifiques aux périphériques de la façon dont leurs broches GPIO sont attribuées.

Il existe plusieurs tableaux ESP32-cam avec des définitions différentes.

Dans mon cas, j'utilise le module de came AI-Thinker et par conséquent, je devrais configurer le serveur webcam Tasmota32 pour utiliser le modèle AITHINKER CAM au lieu du modèle par défaut.

(Si votre ESP32-cam est différent, alors cochez <https://tasmota.github.io/docs/ESP32/> pour le modèle approprié et utilisez celui-ci au lieu de la CAM AITHINKER.)

**Copiez le modèle AITHINKER CAM :**

Copiez le modèle AITHINKER CAM : dans "**Configuration**" -> "**Other**" -> la ligne "**Esp32 Template**" cocher "**Activate**" et "**Save**"

[templateESP32cam.json](#)

```
{ "NAME" : "AITHINKER CAM", "GPIO" : [4992, 1, 672, 1, 416, 5088, 1, 1, 1, 6720, 736, 704, 1, 1, 5089, 5090, 0, 5091, 5184, 5152, 0, 5120, 5024, 5056, 0, 0, 0, 0, 4928, 576, 5094, 5095, 5092, 0, 0, 5093], "FLAG" : 0, "BASE" : 2 }
```

esp32cam001.pdf

## Template pour ESP32 Cam S3

## Template ESP32 Cam S3

```
{“NAME”：“ESP32S3-  
EYE”,“GPIO”:[1,1,7872,1,5024,5056,5120,5152,5090,5089,5091,5088,5092,5184,1,4992,5095,5094,5  
093,0,0,896,0,0,0,0,8800,8832,8864,7808,7840,960,928,1,1,864,992],“FLAG”：0,“BASE”：1}
```

From:

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

Permanent link:

<https://chanterie37.fr/fablab37110/doku.php?id=start:arduino:esp32b:espcam2:tasmota&rev=1779187035>

Last update: **2026/05/19 12:37**

