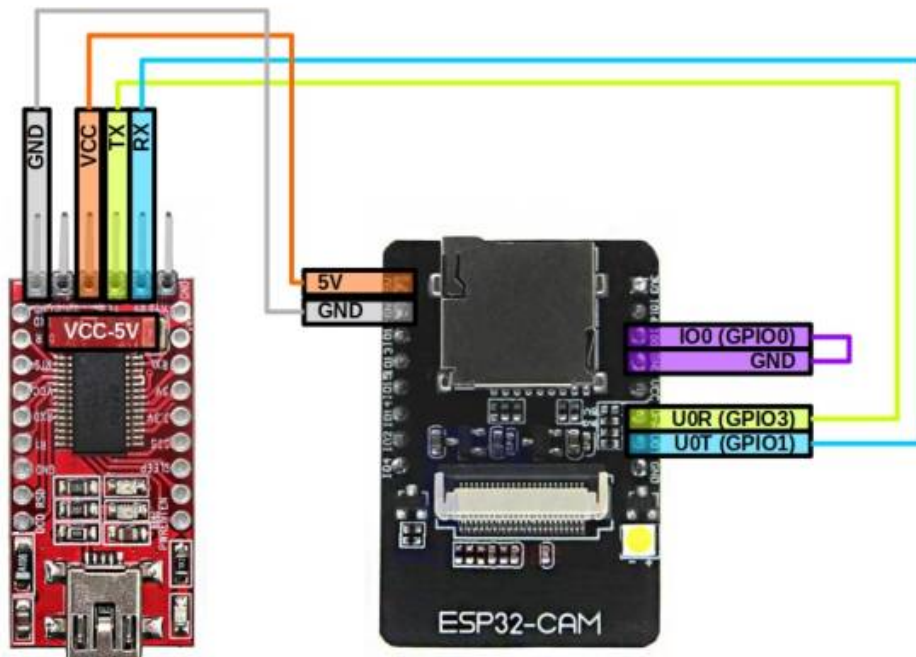


ESP32-CAM wiring: Firmware flash mode (add GPIO0-GND jumper)



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 `echo`, 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
```



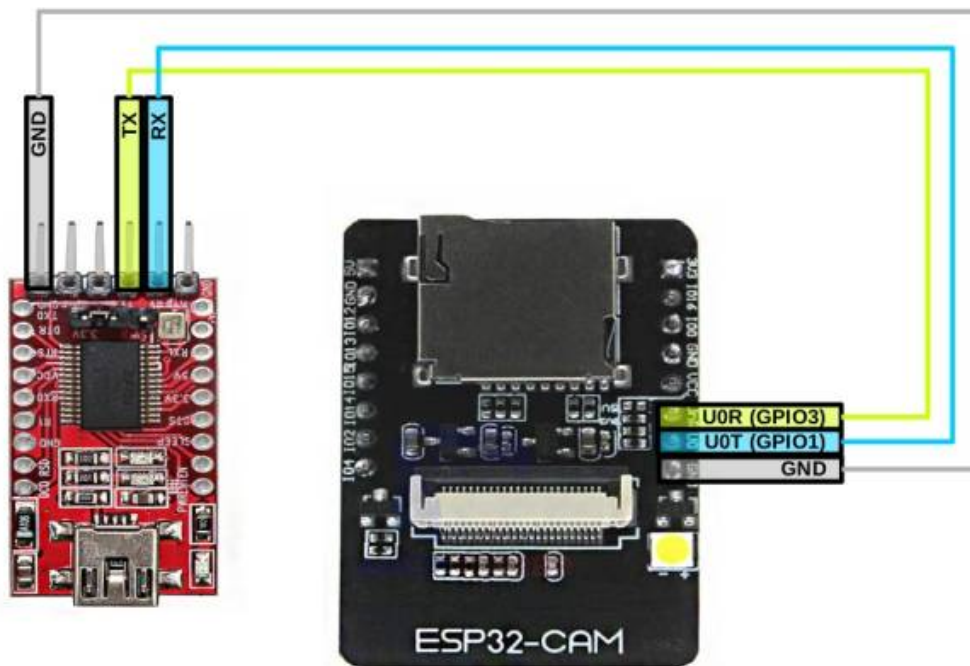
Attendre jusqu'à esptool.py est complètement fait avant de passer à autre chose. Clignoter un firmware peut prendre quelques minutes à compléter. Si vous rencontrez des problèmes tout en clignotant, essayez un taux de baud différent (-b) que le défaut 115200, comme -b 921600. La FAQ Tasmota peut vous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



Si esptool.py Pends à Connecting..., puis appuyez sur le bouton Redémarrer (RST) sur votre module ESP-cam.

Attendre jusqu'à esptool.py est fait. Ensuite, déconnectez votre adaptateur USB en TTL de votre ordinateur et retirez le sautoir 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. Dans les sections suivantes, nous verrons comment surveiller l'appareil via l'adaptateur TTL, qui est facultatif, puis comment alimenter l'appareil et le configurer, qui sont tous deux nécessaires.

Console série

Cette section est optionnelle mais fortement recommandée pour faciliter les problèmes de puissance de dépannage, la spécification incorrecte des composants, les redémarrages aléatoires, et ainsi de suite. N'hésitez pas à passer au câblage autonome si vous n'avez pas envie de configurer une connexion série pour surveiller votre carte pendant que vous la configurez.

La configuration d'une connexion série à votre appareil Tasmota ESP32-cam vous permet de surveiller son état via une connexion filaire à votre ordinateur. Ceci est utile pour résoudre les problèmes qui se produisent avant que vous puissiez accéder à l'interface Web de l'appareil (par exemple, incapable de se connecter à son point d'accès, boucles de démarrage) et pendant les étapes de configuration initiales, car il n'y aura pas de marqueurs physiques de l'état de l'appareil sur lesquels se fier.

Pour créer une console série pour votre appareil, vous aurez besoin d'un (a) adaptateur USB vers TTL et d'un (b) émulateur de terminal. Si vous avez suivi ce guide, vous devriez déjà avoir un adaptateur USB à TTL, qui devrait maintenant être connecté à la carte ESP32-cam comme suit:

 Pour établir un moniteur série, nous utiliserons une application GNU appelée screen. Tout d'abord, ouvrez un terminal et vérifiez s'il est installé sur votre ordinateur ou non:

Tasmota a été créé et il est toujours entretenu par Theo Arends. Il a commencé comme alternative à la société commerciale Sonoff et est passé à un projet indépendant, gratuit et open source qui fournit

plusieurs micrologiciels pour les appareils basés sur ESP8266. Les micrologiciels sont livrés avec une interface Web simple qui vous permet de contrôler et de configurer les modules principaux de la carte ainsi que l'intégration avec un serveur MQTT et plus encore. Même si le soutien de Tasmota à l'ESP32 est toujours en développement bêta, mon expérience avec elle a été très positive.

L'un des principaux micrologiciels de webcam pour la caméra ESP32 est celui fourni par Espressif lui-même, le croquis CameraWebServer Arduino. Celui-ci a des caractéristiques que le micrologiciel de la webcam Tasmota32 n'offre pas, telles que la reconnaissance faciale et la détection de mouvement. Cependant, mon expérience avec le streaming vidéo a été négative. Plus précisément, le streaming fonctionne sans problème lorsque la résolution vidéo est faible (640x480) mais il se débat un peu en courant à une résolution moyenne à élevée, c'est-à-dire que le nombre d'images par seconde diminue sensiblement. J'ai également remarqué que la carte fonctionne très chaud lors de l'exécution du croquis CameraWebServer Arduino, même lorsque les tâches les plus intensives en CPU (détection de mouvement et reconnaissance de visage) sont désactivées.

D'autre part, le serveur de webcam Tasmota32 semble être beaucoup mieux performant dans les zones avec lesquelles le croquis CameraWebServer Arduino se débat. Plus précisément, le streaming est plus fluide et la planche ne semble pas aussi chaude. Je n'ai pas eu l'occasion d'enquêter sur les raisons pour lesquelles cela se produit et de mesurer la différence réelle dans les images par seconde et la température, alors ne prenez pas mon opinion trop au sérieux. En outre, je ne peux pas dire si cela se produit pour toutes les cartes ESP32-cam parce que je n'ai testé qu'avec le module AI-Thinker. Dans l'ensemble, cependant, mon expérience avec le micrologiciel Tasmota32 a été meilleure qu'avec le micrologiciel Espressif dans la zone qui, je pense, est la plus pertinente pour un module de caméra, à savoir les performances de streaming vidéo. En plus de cela, le micrologiciel Tasmota offre une multitude de méthodes pour interagir avec l'ESP32-cam à distance, tandis que le croquis Espressif est très limité à cet égard.

Si vous n'avez jamais entendu parler de Tasmota auparavant, consultez la vidéo d'introduction de Robbert (The Hook Up):

Avant de pouvoir flasher le serveur de webcam Tasmota32 sur la caméra ESP32, nous devons installer quelques paquets et configurer les autorisations de notre utilisateur Linux.

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
```

Alternativement, lorsque nécessaire pour courir esptool.py, au lieu de esptool.py OPTIONS, courir comme python3 -m esptool OPTIONS. Si vous choisissez de le faire, sautez l'étape suivante.

```
Si esptool.pyn'a pas été trouvé, cela signifie que votre utilisateur
.local/binn'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:

Mode flash ESP32cam

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 Permalink

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

developmentparties 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 (par exemple, le Feb 11th, 2022note).

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_PORTà 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_PORTa été définie. Si vous vous déconnectez ou même fermez le terminal actuel, vous devrez redéfinir ESP_PORTpour continuer à l'utiliser.

Vous pouvez vérifier cela ESP_PORTa 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 que \$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 esptool.py est complètement fait avant de passer à autre chose. Clignoter un firmware peut prendre quelques minutes à compléter. Si vous rencontrez des problèmes tout en clignotant, 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 Pends à Connecting..., puis appuyez sur le bouton Redémarrer (RST) sur votre module ESP-cam.

Attendre jusqu'à esptool.py est fait. Ensuite, déconnectez votre adaptateur USB en TTL de votre ordinateur et retirez le sautoir en mode flash (GPI00-GND) de la caméra ESP32.

ESP32cam mode non flash

Vous êtes maintenant prêt à alimenter votre nouveau dispositif Tasmota ESP32-cam à l'aide d'une alimentation indépendante. Dans les sections suivantes, nous verrons comment surveiller l'appareil via l'adaptateur TTL, qui est facultatif, puis comment alimenter l'appareil et le configurer, qui sont tous deux nécessaires.

Top Console série [Permalink](#)

Cette section est optionnelle mais fortement recommandée pour faciliter les problèmes de puissance de dépannage, la spécification incorrecte des composants, les redémarrages aléatoires, et ainsi de suite. N'hésitez pas à passer au câblage autonome si vous n'avez pas envie de configurer une connexion série pour surveiller votre carte pendant que vous la configurez.

La configuration d'une connexion série à votre appareil Tasmota ESP32-cam vous permet de surveiller son état via une connexion filaire à votre ordinateur. Ceci est utile pour résoudre les problèmes qui se produisent avant que vous puissiez accéder à l'interface Web de l'appareil (par exemple, incapable de

se connecter à son point d'accès, boucles de démarrage) et pendant les étapes de configuration initiales, car il n'y aura pas de marqueurs physiques de l'état de l'appareil sur lesquels se fier.

Pour créer une console série pour votre appareil, vous aurez besoin d'un (a) adaptateur USB vers TTL et d'un (b) émulateur de terminal. Si vous avez suivi ce guide, vous devriez déjà avoir un adaptateur USB à TTL, qui devrait maintenant être connecté à la carte ESP32-cam comme suit:

Moniteur série ESP32cam

Pour établir un moniteur série, nous utiliserons une application GNU appelée screen. Tout d'abord, ouvrez un terminal et vérifiez s'il est installé sur votre ordinateur ou non:

```
which screen
```

et si which ne peut pas trouver l'application, puis l'installer via votre gestionnaire de paquets système. Pour aptDistros basées:

```
sudo apt update && sudo apt install screen
```

Maintenant, connectez votre adaptateur USB à TTL à votre ordinateur et, comme auparavant, trouvez le port USB que votre adaptateur utilise /dev/:

```
ls /dev/ttyUSB*
```

et si différent que echo \$ESP_PORT, le mettre à la variable environnementale ESP_PORT; sinon, continuez.

screen a de nombreuses options, mais dans ce cas, nous avons juste besoin d'entrer ce qui suit pour établir une connexion et enregistrer la sortie (-L) dans un fichier appelé screenlog.0:

```
screen -L $ESP_PORT 115200
```

Pour arrêter screen, pressez Ctrl + a et puis \, ce qui suscitera screen pour demander si vous voulez arrêter (y).

A noter, la commande C-mentionné dans le manuel signifie Ctrl plus une autre lettre. Vous pouvez voir une liste de commandes par défaut via C-a ? lors de l'exécution de l'écran.

Voilà ! Vous êtes maintenant prêt à commencer à configurer votre appareil ou à résoudre tout problème avec celui-ci. Une fois que vous avez alimenté votre périphérique Tasmota ESP32-cam, il devrait commencer à diffuser des messages sur votre ordinateur via screen.

Câblage autonome

Si vous avez acheté un adaptateur USB à DIP, vous pouvez maintenant alimenter votre ESP32-cam indépendamment, comme suit:

Mode autonome ESP32cam

Utilisez une alimentation 5V capable de fournir au moins 1A, comme un ancien chargeur de téléphone portable. Cependant, si vous commencez à rencontrer des problèmes liés à l'énergie (p. ex.,

Brownout detector was triggered), remplacez votre alimentation pour une puissance capable de livrer au moins 2A à la place et vérifiez que votre câble est évalué pour ce courant.

Pendant le démarrage et lors de la recherche de points d'accès sans fil, la carte nécessite beaucoup plus de puissance que lors du ralenti. Parce que quelques alimentations électriques n'ont pas été conçues pour gérer de tels changements soudains de consommation d'énergie, des chutes de tension et un courant insuffisant à la carte pourraient se produire, ce qui provoque l'instabilité de l'appareil et déclenche un redémarrage.

Si brownoutLes problèmes persistent après avoir changé votre alimentation, ma recommandation est d'utiliser un convertisseur de dollar entre votre alimentation et la carte ESP32-cam. L'ajout d'un convertisseur de dollar vous donne plus de flexibilité dans le choix d'une alimentation, la possibilité d'ajuster la tension de sortie pour mieux s'adapter à votre projet, le courant de sortie supplémentaire, la protection contre les courts-circuits et le lissage/filtrage de tension intégré. Pour référence, j'ai utilisé divers convertisseurs basés sur le LM2596 bon marché dans les projets ESP32-cam avec beaucoup de succès. Vous pouvez facilement acheter l'un de ces convertisseurs pour moins de 2 \$US et si vous n'avez pas de multimètre, certains viennent même avec un écran LED pour afficher la tension d'entrée et de sortie. Cependant, en fonction de vos besoins, il existe d'autres convertisseurs de buck (un peu plus chers mais légèrement meilleurs, aussi) que vous pouvez utiliser aussi. Dans tous les cas, assurez-vous de régler vos convertisseurs avant de vous fixer à votre carte; sinon, cela pourrait endommager définitivement votre carte et vos composants.

Convertisseur de buck

Si vous ne voulez pas acheter ou n'avez pas d'espace pour ajouter un convertisseur de dollar à votre projet, une autre solution possible brownoutLes problèmes comprennent l'ajout d'un condensateur électrolytique à travers les broches 5V et GND, qui doit être placé à proximité des broches ESP32-cam. Remarquez que c'est exactement ce qui est fait dans les convertisseurs de buck dans le but de gérer l'instabilité de la tension, et assez souvent, un 220uFLe condensateur électrolytique est utilisé à une telle fin, de sorte que vous voudrez peut-être en utiliser un aussi (tout bouchon évalué 10Vou plus haut devrait être bien). Cependant, l'incapacité de régler facilement la tension de sortie peut s'avérer être un problème encore.

220uF 35v cap

Enfin, si l'instabilité de l'alimentation est inhérente à votre projet (par exemple, la batterie ou l'énergie solaire basée), consultez la section Configurations SetOption. Par défaut, le micrologiciel Tasmota implémente plusieurs configurations liées à l'alimentation qui peuvent inverser une ou toutes les modifications que vous avez apportées aux règles, modèles, composants, etc. (voir SO36), ou même réinitialiser complètement l'appareil (voir SO65).

Pour plus d'informations sur l'alimentation de ce projet et d'autres projets d'électronique, vous voudrez peut-être jeter un oeil aux vidéos suivantes faites par DroneBot Workshop et Andreas Spiess:

Configuration



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

Si vous ne trouvez pas le point d'accès sans fil Tasmota, il est possible que l'adaptateur USB ne soit pas en mesure de fournir suffisamment d'énergie pour faire fonctionner les fonctionnalités WiFi de manière fiable. Dans ce cas, vérifiez la section de câblage autonome.

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 192.168.4.0/24pool, qui signifie que l'ESP32-cam webUI est à 192.168.4.1:80; Sinon, l'interface Web sera au premier adr dans la piscine 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. 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.)

```
which screen
```

et si which ne peut pas trouver l'application, puis l'installer via votre gestionnaire de paquets système. Pour aptDistros basées:

```
sudo apt update && sudo apt install screen
```

Maintenant, connectez votre adaptateur USB à TTL à votre ordinateur et, comme auparavant, trouvez le port USB que votre adaptateur utilise /dev/:

```
ls /dev/ttyUSB*
```

et si différent que echo \$ESP_PORT, le mettre à la variable environnementale ESP_PORT; sinon, continuez.

screena de nombreuses options, mais dans ce cas, nous avons juste besoin d'entrer ce qui suit pour établir une connexion et enregistrer la sortie (-L) dans un fichier appelé screenlog.0:

```
screen -L $ESP_PORT 115200
```

Pour arrêter screen, presse Ctrl + a et puis \, ce qui suscitera screen pour demander si vous voulez arrêter (y).

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=1772639381>

Last update: 2026/03/04 16:49

