

Toggle OV2640 settings

WiFi RSSI	-69 dBm
Resolution	XGA(1024x768)
Special Effect	Color
LED Intensity	0 <input type="range"/> 100
Get Still	Stop Stream



une **caméra** dans un **train miniature**

installation d'un module ESP32 CAM

Gilbert Ghyselbrecht (Belgique)

Découvrez comment améliorer votre train miniature grâce au module ESP32-CAM, abordable et polyvalent. Suivez étape par étape l'installation de la caméra dans votre locomotive, la création d'une carte mère sur mesure pour l'alimentation électrique et la configuration de l'indispensable logiciel.



Figure 1. Une locomotive qui convient pour le module caméra.

Lorsqu'on m'a demandé s'il était possible d'installer une caméra dans la locomotive d'un train miniature, j'ai immédiatement pensé à un module ESP32-CAM. Il est disponible partout à très bas prix et avec différentes options d'objectifs et de caméras, et des câbles plats de différentes longueurs pour la connexion de la caméra au module. Cet article décrit l'installation, une carte mère pour l'alimentation électrique et le logiciel nécessaire.

Tout d'abord, j'ai dû trouver une locomotive adaptée avec suffisamment de place pour le module ESP32-CAM. Elle doit de préférence être munie d'un capot en plastique si on veut loger l'antenne Wi-fi à l'intérieur. J'ai trouvé un modèle approprié dans la série HO de Märklin, à savoir la locomotive électrique allemande de type B103 (**figure 1**). En déplaçant le module de décodage numérique du train, j'ai libéré suffisamment d'espace pour le module caméra et une carte mère de ma conception (**figure 2**).



Figure 2. L'espace disponible dans la locomotive pour le module camera.

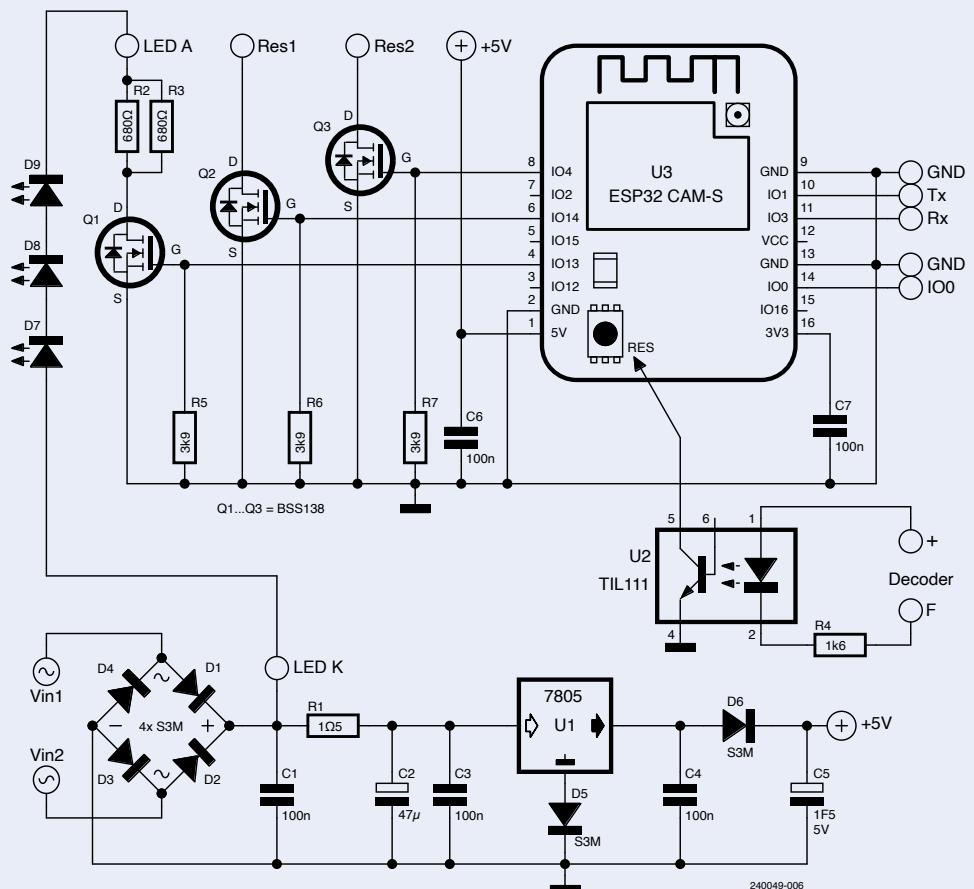


Figure 3. Schéma de la carte d'alimentation.

Alimentation électrique

Un deuxième défi consistait à assurer une alimentation fiable de 5 V pour l'ESP32. Nous supposons ici que les voies sont commandées numériquement. Avec une voie analogique à l'ancienne utilisant un transformateur, il n'y a pas de tension sur les rails lorsque le train est immobile. Avec un système numérique, en revanche, il y a toujours une tension alternative d'environ 18 V sur les rails, en provenance de l'unité de commande. Peu importe le système ou la marque utilisés. Le courant est fourni au train par les roues (ou, dans le système Märklin, par les roues et un contact coulissant sur le rail central). Cependant, cette alimentation n'est pas stable à 100 % lorsque le train roule, car de brèves interruptions peuvent parfois se produire au niveau des aiguillages ou lorsque la voie est sale. Ceci peut conduire à la réinitialisation de l'ESP32 ou à la détection d'une baisse de tension : La connexion Wi-fi est interrompue et tout doit être redémarré.

J'ai rapidement rejeté l'idée d'utiliser une petite batterie comme « alimentation de secours », car cela aurait nécessité un circuit supplémentaire pour la charger en toute sécurité. Il aurait aussi fallu un circuit pour couper l'alimentation au bout d'un moment afin d'éviter la batterie de se décharger complètement. Vu l'espace limité dans la locomotive, impossible d'y intégrer tout cela. Une solution consisterait à loger l'alimentation à batterie dans un wagon séparé et à l'atteler derrière la locomotive, c'est-à-dire en d'autres termes, la renaissance du tender arrière.

Après de nombreuses tentatives, j'ai décidé d'utiliser un supercondensateur de 5 V et 1,5 F de dimensions $25 \times 17 \times 9$ mm [1]. Grâce à cela, la caméra continue à fonctionner normalement pendant environ huit secondes après une interruption de l'alimentation électrique, ce qui est suffisant pour un fonctionnement fiable dans des conditions normales.

Le circuit

La figure 3 montre le schéma complet du circuit. En bas à gauche, la tension de 18 V provenant des rails est connectée au pont redresseur (D1...D4). Le reste du circuit ne doit pas entrer en contact avec les parties métalliques de la locomotive ! La résistance R1 de 1,5 Ω limite les pointes de courant excessives et sert également de fusible. Le régulateur de tension U1, un 7805 ordinaire, est relié à la surface métallique de la locomotive par une plaque isolante qui sert à dissiper la chaleur. La diode D5 dans la ligne de masse du 7805 compense la perte dans la diode en série D6 vers le supercondensateur C5, qui est ainsi chargé à 5 V. Le supercondensateur est chargé très rapidement, en 3 s environ, et le courant de charge est limité à environ 1,5 A par la protection contre les courts-circuits du régulateur 7805 (figure 4).



Figure 4. Charge du supercondensateur mesuré sur R1.

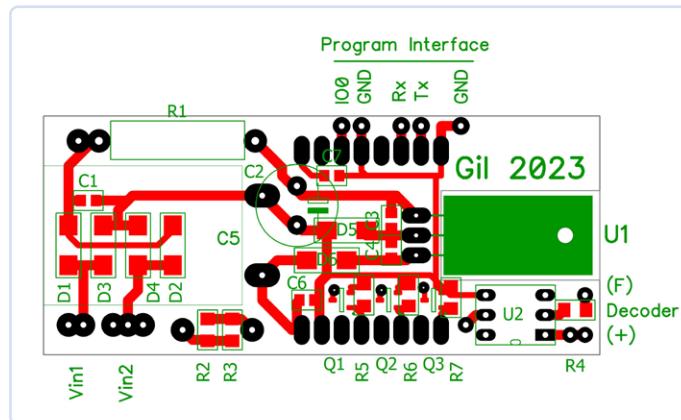


Figure 5. Conception du circuit imprimé de la carte d'alimentation.



Figure 7. Trois LED blanches à l'avant de la locomotive.

Pour l'alimentation électrique, un circuit imprimé simple face a été conçu pour s'adapter à l'espace disponible dans la locomotive. Les pièces principales sont des composants traversants, mais ils sont montés en surface. Le circuit imprimé est monté sur le châssis avec le côté cuivre vers le haut et fixé en place avec le régulateur 7805 coudé (voir **figures 5 et 6**).

Les broches de deux barrettes à huit broches sont pliées et soudées à la carte pour brancher le module ESP32-CAM. J'ai commandé deux types de caméra OV2640 avec un câble de connexion de 75 mm de long, l'une avec un objectif d'un angle de vue de 66° et l'autre de 120°. Selon l'espace disponible, la caméra appropriée est installée à l'avant de la locomotive, derrière le pare-brise. Toutefois, la fenêtre en plastique d'origine du train devant la caméra doit être retirée, car elle n'est pas suffisamment transparente.

Sorties et autres

Trois LED blanches lumineuses (D7...D9) sont fixées à l'avant de la locomotive pour bien éclairer la voie dans les sections sombres et les tunnels (**Figure 7**). Les trois LED sont connectées en série et sont commandées par un signal MLI (broche IO13 du module ESP) via le tampon MOSFET Q1 (un BSS138) et une résistance en série de 340 Ω (deux de 680 Ω en parallèle). La luminosité des LED peut être réglée via l'ESP32.



Figure 6. La carte d'alimentation dans la locomotive, fixée par le 7805.

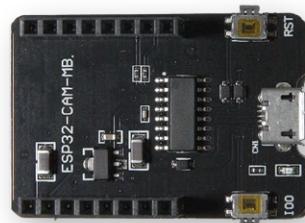


Figure 8. Module de programmation pour l'ESP32-CAM.

Deux autres sorties avec des tampons BSS138 sont disponibles pour d'autres applications, commandées via IO14 et IO4. La consommation de courant du module est de 250 mA lorsque la connexion Wi-fi est active et tombe à 15 mA au repos.

Une barrette dédiée à 3 broches sert à la programmation via un adaptateur USB-FTDI. Connectez IO0 à GND et appuyez sur le bouton de réinitialisation du module pour le mettre en mode de programmation. Il est cependant plus facile de retirer le module de la locomotive et de le brancher sur une carte de programmation ESP32-CAM (ESP32-CAM-MB) pour le programmer (**figure 8**). Ce module spécial est connecté à l'EDF Arduino via USB et comprend les boutons BOOT et RST nécessaires à l'ESP32-CAM.

Dès que le module ESP32-CAM est installé dans le train, le bouton de réinitialisation situé sur le module lui-même n'est plus accessible. Un fil supplémentaire est donc soudé au bouton de réinitialisation et connecté à la broche 5 de l'optocoupleur U2 sur la carte mère (**figure 9**). Une autre option consiste à installer un bouton de réinitialisation supplémentaire accessible de l'extérieur. La sortie AUX du décodeur du

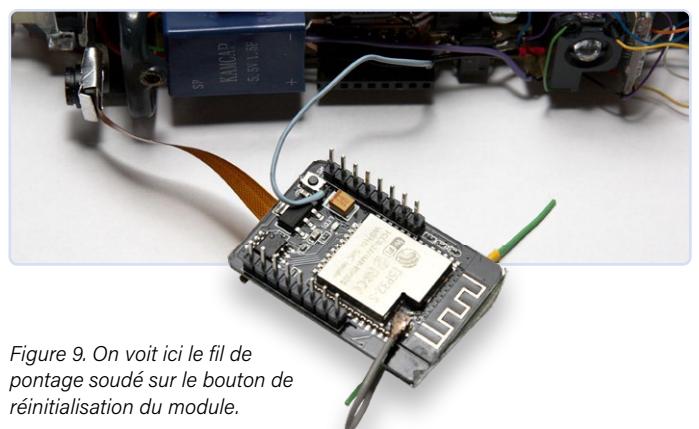


Figure 9. On voit ici le fil de pontage soudé sur le bouton de réinitialisation du module.



train sert à commander U2, qui met l'entrée de réinitialisation du module ESP à l'état bas, ce qui met la caméra hors tension ou la redémarre. J'ai remplacé l'antenne Wi-fi du module ESP par une petite antenne dipôle fabriquée à partir d'un morceau de fil et reliée par un court câble coaxial doté d'un minuscule connecteur IPEX approprié. Ce câble est parfois fourni avec le module ESP, mais vous pouvez aussi facilement l'acheter (avec un connecteur SMA à l'autre extrémité) dans un magasin spécialisé. Une petite modification doit être apportée à la carte du module : Il faut couper un pont de soudure à côté du connecteur d'antenne et en placer un autre [2]. La longueur d'onde à 2,4 GHz est de 12,5 cm. Le dipôle se compose de deux morceaux de fil de 3 cm de long, dont l'un est relié au conducteur intérieur et l'autre au blindage du câble coaxial. Une gaine thermo rétractable maintient le tout en place. L'antenne peut être montée à l'intérieur ou à l'extérieur de la locomotive, mais de préférence pas trop près du moteur ou d'autres pièces métalliques (**figure 10**).

Logiciel

Le module caméra est programmé avec l'EDI Arduino 1.8 ou 2.0. On trouve des instructions d'installation à [3]. Ouvrez *Fichier / Exemples* dans l'EDI Arduino et sélectionnez **ESP32 Camera CameraWebServer**. Après avoir adapté les champs SSID et PASSWORD, le code peut être compilé et chargé dans le module caméra.

Placez ensuite la locomotive sur les rails et allumez votre unité de commande du train. Utilisez l'unité de commande pour activer et désactiver la sortie AUX afin de réinitialiser et de démarrer l'ESP-CAM. Le module se connecte à votre réseau local sans fil (WLAN). Dans le

navigateur de votre ordinateur, entrez l'adresse IP que le routeur a attribuée au module (normalement 192.168.1.xxx ou 198.168.0.xxx) afin de vous connecter à la caméra. L'adresse correcte est affichée dans le moniteur série de l'EDI Arduino, réglé sur 115200 bauds. Vous pouvez jouer avec tous les paramètres du module caméra dans l'exemple de logiciel de test, mais le programme est complexe et parfois difficile à comprendre. J'ai donc développé un programme simple [6] qui ne contient que les paramètres les plus importants et dans lequel les pages de l'interface web sont stockées dans des fichiers d'en-tête séparés (lisibles et accessibles).

Si la connexion au WLAN échoue, le module passe en mode point d'accès (AP) en utilisant l'adresse 192.168.4.1 avec le SSID **ESP32-CAM** (sans mot de passe) et permet ainsi la connexion avec un navigateur sans routeur ni connexion Internet.

La page web est créée à l'aide des trois fichiers compilés avec les fichiers d'en-tête :

1. `index.html` in `#include "indexFile.h"`
2. `espcam.css` in `#include "cssFile.h"`
3. `espcam.js` in `#include "jsFile.h"`

Dès que le module est démarré et que la connexion Wi-fi est établie, les LED blanches à l'avant de la locomotive s'allument très faiblement. L'interface web, qui figure en tête de l'article, offre la possibilité de prendre une photo, de la sauvegarder ou de lancer un flux vidéo. La résolution de l'image peut être réglée, une résolution plus faible entraînant une fréquence d'images plus élevée.

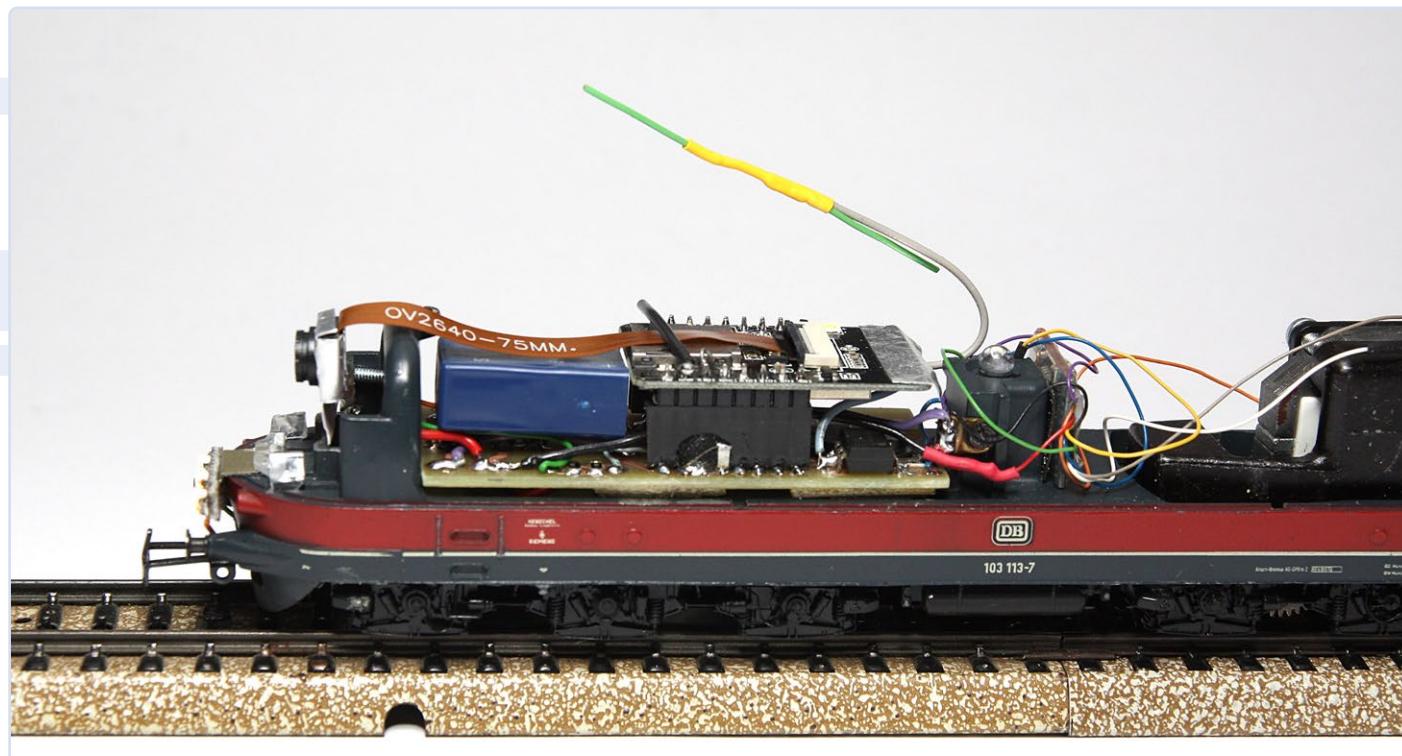


Figure 10. Le projet terminé avec l'antenne dipôle.



Flux vidéo

Le flux vidéo est un flux MJPEG [4] qui dépend du client (le navigateur web). Le navigateur maintient la connexion ouverte et effectue des requêtes image par image via JavaScript. La vitesse du navigateur et la qualité de la connexion Wi-fi déterminent le nombre d'images traitées par seconde. Si vous le souhaitez, ce nombre peut être contrôlé à l'aide d'un oscilloscope sur la broche IO12.

Un menu permet de sélectionner des images en couleur, en noir et blanc ou en sépia. Enfin, la luminosité des LED blanches situées à l'avant de la locomotive peut être réglée à l'aide d'un curseur. Notez que les LED sont très lumineuses lorsqu'elles sont réglées au maximum. Enfin, j'aimerais mentionner le programme supplémentaire, encore plus puissant, esp32-cam-webserver sur GitHub [5], qui fournit toutes les fonctions de la caméra et permet de charger le programme *over the air* (OTA) directement dans la caméra de la locomotive sans avoir à ouvrir la locomotive ou à la retirer des rails.

Ce projet

Il ne s'agit pas d'un produit clé en main, car beaucoup de choses dépendent du modèle de locomotive spécifique dans lequel vous souhaitez intégrer la caméra. Toutefois, le concept d'une alimentation sans interruption (ASI) avec supercondensateur pourrait également être intéressant pour d'autres applications. 

VF : Denis Lafourcade — 240049-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (gilbert.ghyselbrecht@skynet.be), ou contactez Elektor (redaction@elektor.fr).

À propos de l'auteur

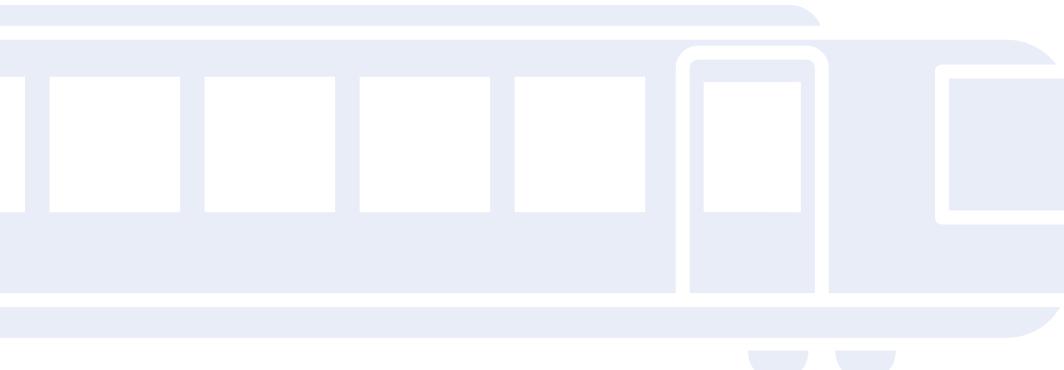
Gilbert Ghyselbrecht a longtemps travaillé comme technicien à la station radio-maritime Ostende Radio. À l'époque, il était responsable de la maintenance des équipements de diffusion. Il s'est intéressé en outre au développement et à la réalisation de systèmes de télécommande qui permettaient de commander ces équipements à distance. Son expertise a débuté avec le microprocesseur 8085 et le logiciel CP/M puis sur les commandes vocales avancées (VoIP).



Produit

› **Carte de développement ESP32-Cam-CH340**
www.elektor.fr/19333

Cette version du module ESP32-CAM ne nécessite pas d'adaptateur de programmation additionnel, mais elle est légèrement plus grande. Veuillez vérifier avant !



LIENS

- [1] Kamcap: 5.5V Fully Sealed (Waterproof) Supercapacitor :
[https://www.kamcappower.com/products/5.5v-finally-sealed-waterproof-super-capacitor/](https://www.kamcappower.com/products/5.5v-fully-sealed-waterproof-super-capacitor/)
- [2] Random Nerds Tutorials: ESP32-CAM Connect External Antenna:
<https://randomnerdtutorials.com/esp32-cam-connect-external-antenna/>
- [3] Random Nerds Tutorial: ESP32-CAM Video Streaming ... (details on the ESP32-CAM module) :
<https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/>
- [4] Wikipedia: Motion JPEG : https://en.wikipedia.org/wiki/Motion_JPEG
- [5] GitHub: ESP32 CAM Webserver : <https://github.com/easytarget/esp32-cam-webserver>
- [6] Circuit imprimé et logiciel : <https://www.elektormagazine.fr/240049-04>