relevé des compteurs d'eau basé sur l'IA

Partie 2 : intégrez votre ancien compteur dans l'IdO !

Daniel Scaini (Italie)

Les compteurs d'eau intelligents sont déjà sur le marché depuis un certain temps, mais le remplacement de nos anciens compteurs n'est souvent pas si facile, pour des raisons techniques ou administratives. Ce projet permet de transformer n'importe quel compteur analogique en compteur numérique à l'aide d'une plateforme ESP32-CAM, et d'un système d'intelligence artificielle (IA). Dans ce dernier épisode, nous verrons comment installer le micrologiciel et configurer l'objectif de la caméra, pour une mise au point et un positionnement parfaits sur le compteur d'eau. Et bien sûr, nous abordons l'ensemble du processus, basé sur l'IA, pour la reconnaissance et la lecture correctes des éléments du compteur.

Elettronica In
WWW.ELETTRONICAIN.IT

Installation du micrologiciel

Dans un premier temps, vous devez télécharger le micrologiciel précompilé à partir du projet GitHub « AI on the Edge Device ». De nombreux développeurs contribuent à ce projet assez vaste et performant, qui est très largement et bien documenté. Plusieurs versions du firmware sont disponibles sur la page *Releases* [1], chacune étant accompagnée d'un changelog, un fichier texte indiquant les changements, les nouvelles implémentations ou les corrections de bugs effectuées. Nous allons ensuite télécharger le fichier *.zip* contenant le micrologiciel précompilé. A l'intérieur de ce fichier *.zip*, nous trouverons trois fichiers avec une extension *.bin*, qui représentent le firmware à installer.

En plus, un autre fichier *.zip* renferme le contenu qui doit être téléchargé sur la carte SD. Commençons par préparer notre carte SD, dont la capacité ne doit pas dépasser 16 Go. En effet, des erreurs de lecture peuvent se produire avec des cartes de plus grande capacité, comme les cartes de 32 ou 64 Go. Au niveau logiciel, la carte SD ne doit pas avoir de table de partitionnement et doit être formatée au format FAT32. Si notre module ESP32-CAM rencontre des problèmes ou des dysfonctionnements au démarrage, il est conseillé de remplacer immédiatement la carte SD avant de continuer plus loin.

Une fois que nous avons extrait le fichier téléchargé, nous trouverons plusieurs dossiers à l'intérieur. Le dossier *config*, qui contient les fichiers TensorFlow, et le fichier de configuration appelé *config.ini*, nous intéressent particulièrement ici. Dans le fichier *config.ini*, nous trouverons tous les paramètres initiaux, qui peuvent être difficiles à lire ou à interpréter pour ceux qui n'ont pas directement compilé le firmware. Pourtant, parmi la longue liste de paramètres présents, nous sommes particulièrement intéressés par l'entrée `SetupMode = true`. Nous allons changer la valeur de ce paramètre de *true* à *false* pour désactiver le mode de configuration initial. Ce paramètre est responsable de ce que nous verrons dans notre navigateur, lorsque nous y accéderons pour la première fois. En gardant la valeur *true*, nous afficherons un assistant de configuration qui peut être inutile, et pouvant aussi entraîner des problèmes d'écriture sur la carte SD. En changeant la valeur du paramètre à *false*, nous accéderons directement au panneau principal, avec lequel nous devrons nous familiariser.

Il existe un deuxième fichier que nous devons éditer afin d'améliorer le démarrage initial, et c'est *wlan.ini*. Dans ce fichier, en effet, nous devons entrer le SSID et le mot de passe de notre réseau Wi-Fi. Si nous ne donnons pas ces identifiants de connexion, la puce démarrera en mode AP (Access Point), montrant son réseau aux clients. À ce stade, nous pouvons procéder à l'installation du firmware proprement dit, et il y a deux manières possibles. Soit à travers l'outil *Flash Tool* d'Espressif, ou à partir de la console avec *esptool*.

Dans l'ordre, en commençant par la procédure la plus simple, nous téléchargeons le logiciel Flash Download Tools à l'adresse [2]. il est décompressé dans un dossier et nous exécutons le programme *.exe* qui s'y trouve. Nous sélectionnons *ESP32* et *Develop* (**Figure 1**) pour

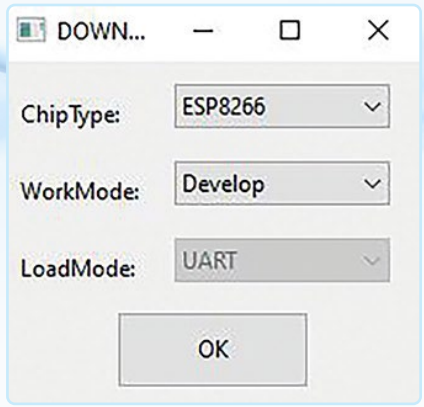

*Figure 1. Sélection de **ESP32** et de **Develop** pour charger le fichier **.bin** dans l'ESP32-CAM.*

programmer le fichier ***.bin***. Nous sélectionnons ensuite les fichiers, dans l'ordre décrit ci-dessous, et entrons leurs index respectifs à côté :

- *bootloader.bin* 0x1000
- *partitions.bin* 0x8000
- *firmware.bin* 0x10000

La vitesse du bus SPI doit être réglée sur 40 MHz et le mode (*MODE*) sur DIO (**figure 2**). Nous pouvons maintenant connecter notre programmateur, choisi d'après les configurations précédentes, et appuyer sur *START*. Après quelques instants, le micrologiciel sera correctement chargé.
Le second mode de chargement nécessite l'utilisation du terminal et d'un environnement avec Python. Par exemple, on peut utiliser Anaconda [3], mais nous recommandons cette procédure principalement pour les utilisateurs expérimentés d'Ubuntu. Dans un premier temps, il faut installer l'outil *esptool*. Pour ce faire, nous devons taper la commande suivante dans la ligne de commande :

```
pip install esptool
```

Si vous utilisez Python 3, vous devez utiliser la commande :

```
pip3 install esptool
```

Branchons maintenant notre programmateur et entrons la commande :

```
esptool erase_flash
esptool write_flash 0x01000 bootloader.bin 0x08000
partitions.bin 0x10000 firmware.bin
```

Bien entendu, en tant qu'utilisateur expérimenté, vous devez vérifier si COM est spécifiée automatiquement ou non. Si ce n'est pas le cas, des problèmes peuvent survenir dès la première instruction. Après avoir écrit le firmware, nous pouvons déconnecter le câble USB de notre PC et enlever tout le câblage fait précédemment, en laissant seulement 5 V et GND sur notre ESP32-CAM, comme indiqué.
Après avoir redémarré le module ou l'avoir relié à une alimentation externe, lors du premier démarrage du micrologiciel, nous devons vérifier si le fichier *wlan.ini* est disponible et s'il contient des informations d'identification valides. Si ces données sont manquantes ou invalides, le module démarrera en mode AP, sans mot de passe ni protection pour le réseau Wi-Fi. Et un nouveau réseau Wi-Fi appelé *AI-on-the-Edge* sera créé.
Il suffit de quelques étapes pour télécharger le fichier directement via la page web du module. Mais procédons dans l'ordre, car la première étape consiste à télécharger le fichier *remote-setup* toujours disponible

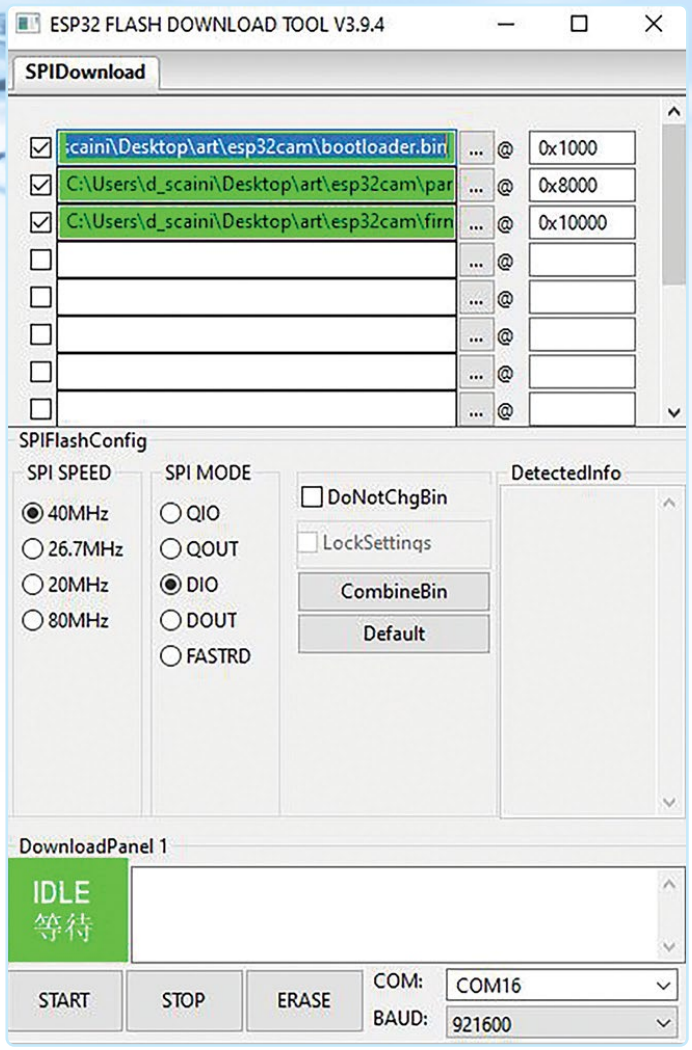

*Figure 2. Le champ **SPI SPEED** doit être réglé sur 40 MHz, tandis que le champ **SPI MODE** doit être réglé sur **DIO**.*

dans les versions du projet ici [1]. Ensuite, la connexion au réseau Wi-Fi "AI-on-the-Edge" affichera une page nous demandant de télécharger un fichier.
Nous chargeons le fichier que nous venons de télécharger et attendons, comme indiqué sur la page. Après un court téléchargement qui peut durer jusqu'à 60 secondes, l'écran de saisie des informations d'identification du réseau Wi-Fi apparaît. Nous entrons notre SSID et notre mot de passe, puis nous cliquons sur *Write wlan.ini*. Si tout s'est bien passé, le redémarrage sera demandé.
L'ESP32 est équipé d'une LED rouge, dont le clignotement a plusieurs significations :

- 5 clignotements rapides (< 1 seconde) : connexion en cours
- 3 clignotements lents (1 seconde on/off) : connexion WLAN établie

Réglage de l'optique de la caméra

Le réglage de la mise au point de l'objectif de la caméra est un point crucial, après avoir téléchargé le logiciel dans le module ESP32-CAM. Par défaut, la mise au point est réglée pour les très longues distances et non pour les distances courtes, comme cela est nécessaire dans notre cas. Lors de cette étape, il est important de procéder très délicatement, en essayant de faire tourner l'extérieur de l'objectif avec des pincettes et, si nécessaire, en enlevant la colle placée à la base de l'objectif à l'aide d'un cutter. Ces opérations de réglage doivent être effectuées avec précaution, comme le montre la **figure 3**.

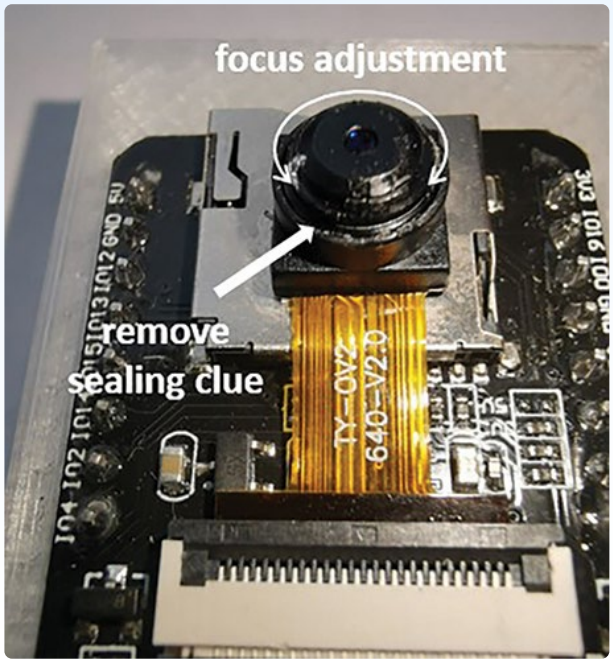

Figure 3. Réglage de l'optique pour la prise de vue rapprochée.

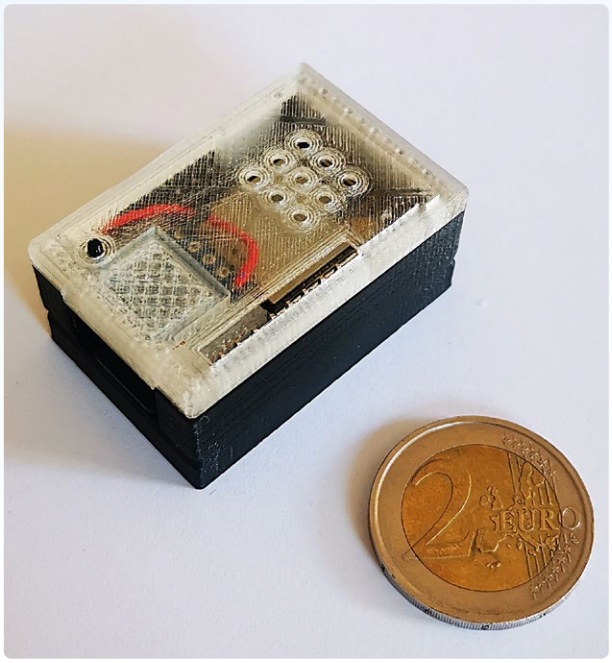

Figure 4. Le module ESP32-CAM dans sa boîte imprimée en 3D.

Positionnement de l'ensemble

Après avoir installé le micrologiciel et réglé la mise au point de la caméra, nous pouvons placer notre montage sur le compteur que nous voulons surveiller. Nous devons essayer de placer la caméra, pour rester à une distance suffisante pour lire correctement les données, et pour qu'elle puisse rester immobile pendant toutes les opérations. Pour surmonter cette difficulté, il existe plusieurs solutions. La meilleure serait sans aucun doute d'imprimer en 3D un support en plastique, qui est fermement fixé au compteur, pour abriter notre montage et le maintenir protégé et immobile à l'intérieur d'un petit boîtier. Un exemple tiré de Thingiverse peut être téléchargé à l'adresse [4]. Sur cette page, vous trouverez le fichier STL pour le boîtier du module (**figure 4**), ainsi que l'entretoise qui vous permet de fixer correctement l'électronique sur le compteur, comme illustré dans la **figure 5**.

Figure 5. Un adaptateur a été fabriqué avec l'imprimante 3D, pour installer l'ESP32-CAM sur le dessus du compteur.

Une alternative plus simple et plus rapide consiste à utiliser un cube en nylon ou en polystyrène, couramment utilisé pour l'emballage. Nous plaçons le module à l'extérieur du cube de polystyrène, face vers le bas, et nous créons un trou pour permettre à l'objectif de l'appareil photo et au flash de passer à travers. Le trou central est dimensionné exactement à la taille de l'indicateur. Si l'indicateur possède également un couvercle, comme le montre la photo, nous pouvons modeler davantage le support pour qu'il s'adapte parfaitement.

Premiers réglages et résultats

Avant d'allumer notre système, nous allons retirer la carte SD et l'insérer dans le PC. Vérifions l'intérieur de la carte SD, dans le dossier *config*, le fichier *config.ini*, pour vérifier que nous avons bien suivi toutes les étapes, et que l'installation du firmware s'est déroulée avec succès À la fin de ce fichier, nous trouverons le paramètre *SetupMode*, en rappelant qu'il doit être réglé sur *false*. Cette étape nous permettra d'accéder directement à l'écran principal et non à l'écran de l'assistant d'installation.

Ensuite, réinsérons la carte SD dans le module et allumons-le. Avec notre PC, connectons-nous à l'adresse que notre routeur aura donnée à l'ESP32, dans notre cas 192.168.43.102. Nous obtenons l'écran de la **Figure 6**. Évidemment, laissons de côté les mesures et l'identification de rectangles ou de photos pour l'instant, et concentrons-nous sur la partie menu située dans la barre noire en haut. Nous allons dans *Settings* et un menu déroulant s'ouvre, affichant quatre éléments : *Set Previous Value*, *Configuration*, *Alignment*, *Region of Interest*.

Commençons par le troisième élément, où à l'intérieur de celui-ci, nous trouvons deux autres rubriques, à savoir *Reference Image* et *Alignment Marker*. En cliquant sur *Reference Image*, cette rubrique nous permet de capturer la première image de notre compteur, et de la conserver comme référence pour toutes les vérifications ultérieures. Pour ce faire, nous cliquons sur *Create New Reference Image* et ensuite

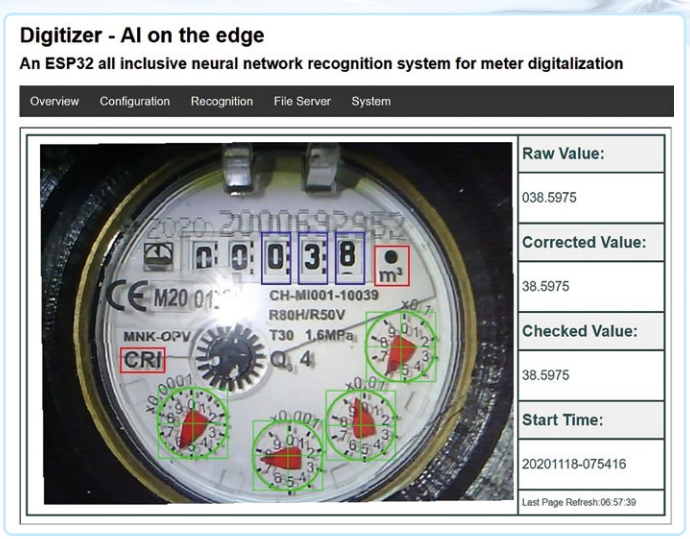

Figure 6. La page web interne à partir de laquelle le système peut être configuré.

sur *Take Reference*, ce qui enverra l'instruction de prendre la photo. En haut de ce menu, nous pouvons modifier l'intensité de la LED, l'angle de rotation de l'image, le contraste, la saturation et bien d'autres valeurs afin d'optimiser l'affichage de l'échantillon choisi. Une fois la première acquisition terminée, nous cliquons sur *Save* puis, après avoir reçu la notification de sauvegarde, sur *Restart* pour activer la référence sauvegardée. À ce stade, il est important que les numéros du compteur à l'image soient correctement alignés les uns par rapport aux autres.

Une fois que le formulaire a redémarré, nous passons à une autre section qui se trouve également dans *Settings/Alignment*, qui est *Alignment Marker* (**figure 7**). Dans cette section, nous devons faire en sorte que le système reconnaisse la position exacte de deux marques de référence que nous pouvons choisir dans le menu déroulant *Select Reference*. Le « 0 » contiendra la balise CRI, qui n'est pas toujours présente sur les compteurs (**Figure 8**). Le second, le « 1 », est l'unité de mesure, c'est-à-dire le mètre cube. Après avoir correctement déplacé la boîte pour centrer les balises, le cas échéant, cliquez sur *Update*

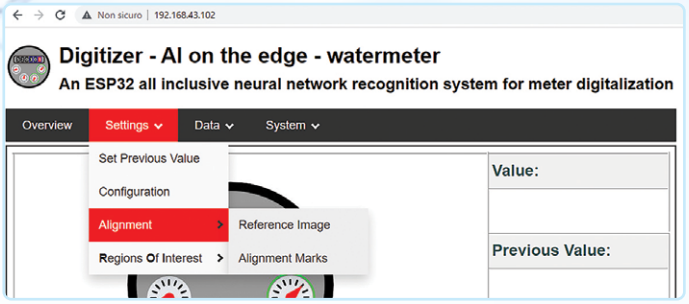

*Figure 7. Dans le menu **Alignment Marks**, nous pouvons sélectionner les marqueurs de référence.*

Reference Image, puis sur *Save* sans redémarrer l'appareil pour l'instant, car nous devons d'abord définir toutes les références.

Dans le menu déroulant, nous sélectionnons maintenant *Settings* puis *Region Of Interest*, et deux autres sous-titres apparaissent, avec *Digital ROIs* et *Analog ROIs*. Nous sélectionnons le premier, ce qui nous amène à l'écran de la **figure 9**. Ici, nous devons marquer tous les chiffres de l'indicateur à l'aide des outils qui se trouvent au bas de l'écran. Certains des numéros sont déjà identifiés dans le menu déroulant situé sous *New ROI*, mais nous devons saisir X et Y pour chaque numéro. Ne modifiez pas les cases à cocher sur le côté, si la photo a été prise correctement, car les zones qui nous intéressent seront toutes alignées et de taille égale.

Il y a deux cadres de couleur rouge pour identifier le numéro, et le cadre intérieur doit être de la même taille que le chiffre. Une fois cette étape terminée, nous cliquons sur *Save* et passons à la dernière étape. Dans la section *Analog ROIs*, nous devons répéter une opération similaire aux précédentes mais avec les indicateurs analogiques (Figure 10). Le bord rond et rouge du masque d'identification doit correspondre exactement au bord rond de l'indicateur. Là encore, les différentes cases à cocher peuvent être conservées si la photo, c'est-à-dire la référence de départ, est bonne, puisque tous les indicateurs auront la même taille. Après avoir terminé, nous cliquons à nouveau sur *Save* et enfin sur *Reboot* pour valider les changements.

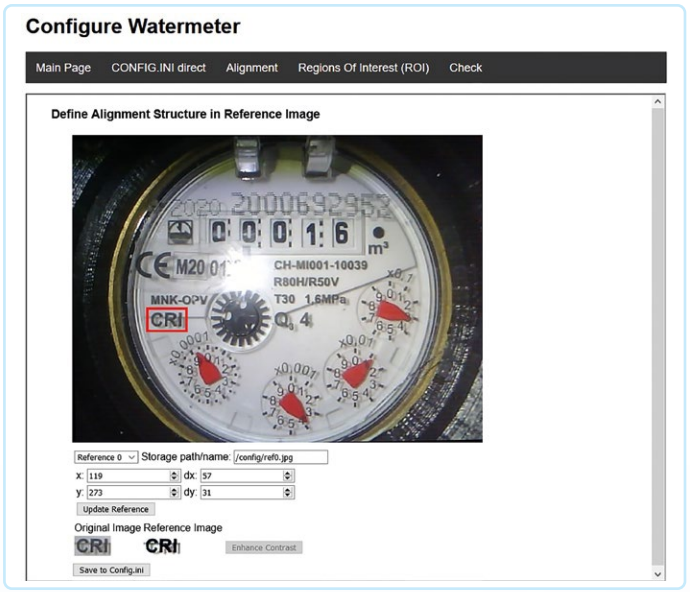

Figure 8. Le marquage CRI n'est pas toujours présent sur les compteurs d'eau.

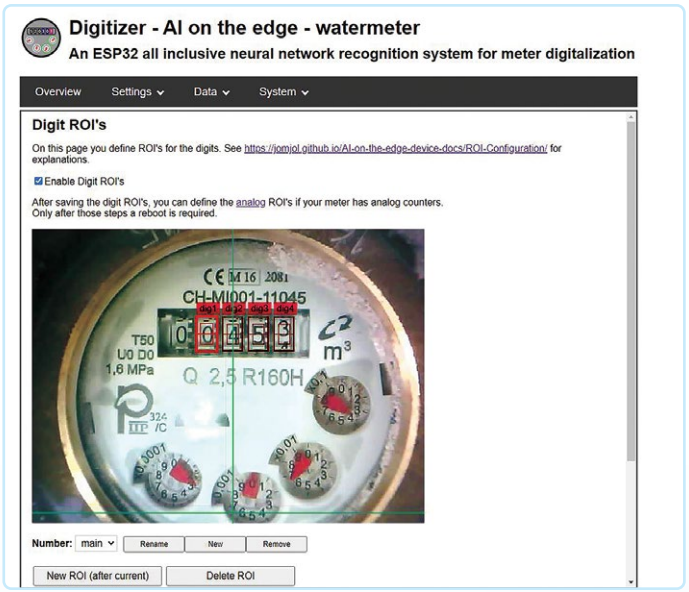

Figure 9. Marquage des chiffres du compteur.

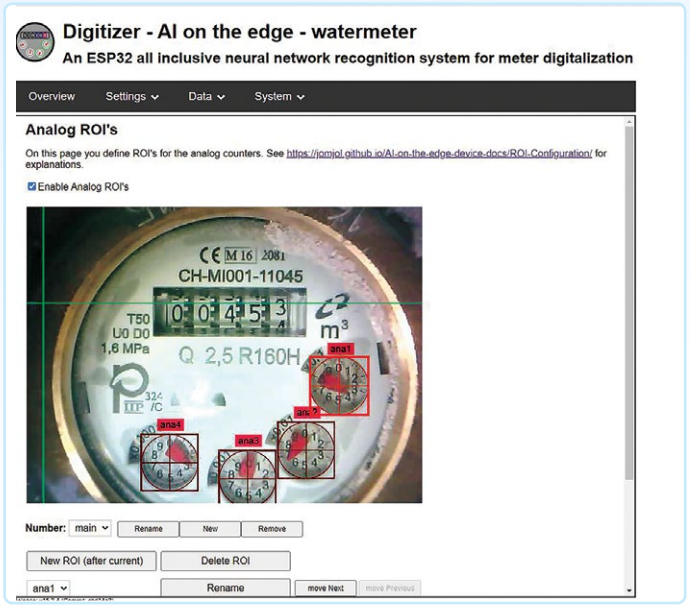

Figure 10. Marquage des aiguilles du compteur.

Configuration REST, MQTT et intégration de Home Assistant

Maintenant que notre module ESP32-CAM peut lire notre compteur, nous avons besoin d'un moyen pratique pour afficher les données. Heureusement, le code est bien écrit, et nous permet d'accéder aux données de plusieurs manières. Dans le firmware précompilé se trouve un serveur http, que nous avons déjà exploité lors de la mise au point des premiers paramètres.

Ce serveur nous fournit non seulement des pages web mais aussi des API REST, utilisables entièrement par la méthode GET. La particularité de cette méthode réside dans sa simplicité d'utilisation, en effet toutes les informations se trouvent dans l'URL et en texte clair. La partie fixe de ces appels est donnée par le point d'entrée *http://IP_DEVICE*, auquel il faut ajouter l'une des parties variables suivantes, en fonction de la finalité recherchée.

Le **tableau 1** présente les commandes qui peuvent être utilisées ainsi que leur signification. Toutes ces commandes sont très pratiques lorsque nous nous trouvons à l'intérieur du même réseau. Mais qu'en est-il si nous nous trouvons sur un réseau externe ?

L'intégration du service MQTT répond à cette question. *Message Queue Telemetry Transport* (MQTT) est un protocole de messagerie très léger, de type publication-abonnement, utilisé lorsque la bande passante est limitée ou que l'on souhaite créer un impact minime au niveau du réseau. Si l'on se réfère à la littérature, ce protocole, créé par Andy Stanford-Clark et Arlen Nipper en 1999, a d'abord été adapté à la surveillance des oléoducs.

L'objectif était de créer un protocole vraiment léger, efficace en termes de bande passante et consommant peu d'énergie électrique, un sujet épineux à l'époque comme aujourd'hui. Cependant, ce protocole a connu une nouvelle vie en 2013 lorsque la version 3.1 a été publiée par IBM et a commencé à s'imposer, en partie grâce à une forte poussée du marché vers l'Internet des objets. L'un des exemples les plus marquants de l'utilisation de MQTT est certainement Facebook Messenger, car en effet, Facebook a déclaré avoir adopté ce protocole pour améliorer la messagerie et limiter l'utilisation du réseau et de la consommation électrique.

Le client est un système quelconque, allant d'un simple microcontrôleur à un serveur complet, qui exécute une bibliothèque MQTT et est connecté à un courtier (broker). Cet acteur envoie des informations au courtier, appelées *payloads*, classées en fonction de leur sujet. L'information, en fait, est organisée en fonction du sujet, appelé *topic*. Lorsqu'un client envoie un nouveau *payload* à un courtier, celui-ci relaie l'information à tous les clients qui lui sont connectés sur un

Tableau 1. Commandes pouvant être envoyées à partir du point d'entrée http://IP_DEVICE.

Chemin	Fonctionnalité	Report/Réponse

PATH_FISSO/GPIO	Fournit une interface pour activer ou connaître l'état d'une broche de notre ESP	Pour définir un statut
		`?GPIO=&Status=high or low`
		
		Pour lire un statut
		`?GPIO=`

PATH_FISSO/reboot	Pour redémarrer le système	
PATH_FISSO/json	Pour obtenir les résultats au format JSON de la dernière lecture effectuée	Exemple de réponse : `{ "main": { "value": "521.17108", "raw": "521.17108", "pre": "521.17108", "error": "no error", "rate": "0.023780", "timestamp": "2023-01-13T16:00:42+0100" } }`

Chemin	Fonctionnalité	Report/Réponse

PATH_FISSO/value	Pour obtenir les résultats sous la forme d'une valeur unique et non au format JSON	Pour extraire une valeur autre que la valeur par défaut nous pouvons utiliser les paramètres GET suivants :
		`?all=true&type=value`
		`?all=true&type=raw`
		`?all=true&type=error`

PATH_FISSO/img_tmp/raw.jpg	Pour obtenir une image brute du compteur	
PATH_FISSO/img_tmp/alg_roi.jpg	Pour obtenir une image brute avec les différentes zones d'intérêt (ROI) en superposition	
PATH_FISSO/statusflow	Il indique exactement quel est le processus actif sur l'appareil à ce moment-là, y compris un horodatage	Un exemple de réponse pourrait être : `Take Image (15:56:34)`
PATH_FISSO/rssi	Indique la puissance du signal WiFi reçu (Unité : dBm)	Voici un exemple de réponse : `-51`
PATH_FISSO/cpu_temperature	Affiche la température de la CPU (Unités : °C)	
PATH_FISSO/sysinfo	Affiche toutes les informations du au format JSON	Un exemple de réponse peut être : `[{"firmware": "", "buildtime": "2023-01-25 12:41", "gitbranch": "HEAD", "gittag": "", "gitrevision": "af13c68+", "html": "development-branch: HEAD (Commit: af13c68+)", "cputemp": "64", "hostname": "WaterMeterTest", "IPv4": "192.168.43.102","freeHeapMem": "2818330"}]`
PATH_FISSO/startime	Indique la « date de mise en service »	Exemple de réponse : `20230113-154634`
PATH_FISSO/uptime	Indique depuis combien de temps il est actif	Voici un exemple de réponse : `0d 00h 15m 50s`
PATH_FISSO/lighton	Modifie l'état du flash de l'appareil photo en le réglant sur ON	
PATH_FISSO/lightoff	Modifie l'état du flash de l'appareil photo en le réglant sur OFF	
PATH_FISSO/capture	Capture une nouvelle image (sans flash)	
PATH_FISSO/capture_with_flashlight	Capture une nouvelle image (avec flash)	
PATH_FISSO/save	Enregistre une nouvelle image sur la carte SD	Vous avez en plus les paramètres suivants : `?filename=test.jpg&delay=1000`
PATH_FISSO/log	N'affiche que la dernière partie des données de la journée (seulement les derniers 80 kB)	
PATH_FISSO/logfileact	Affiche tous les relevés de la journée	

sujet donné. Le client n'a aucune idée du nombre ou de la position des abonnés (*subscribers)* connectés à ce sujet (*topic*).
Dans notre cas, le client est la puce elle-même, tandis que le courtier peut être une installation Hassio/Home Assistant. Pour activer ce service et son intégration, nous allons dans *Settings*, *Configuration*. En bas de la page, nous trouvons la section sur la configuration de MQTT, que nous pouvons éditer comme dans la **Figure 11**. Dans les URI (identifiant de ressource de réseau), nous devons entrer *mqtt://_BROKER ADDRESS:PORT* (habituellement 1883) et, au cas où nous en aurions besoin, le nom d'utilisateur et le mot de passe.
Après avoir saisi ces deux dernières valeurs, allons dans le tableau de bord de Home Assistant et là, sous *Settings → People*, ajoutons un nouvel utilisateur qui peut se connecter avec les mêmes informations d'identification. Dans notre tableau de bord Home Assistant, allons dans *File Editor*, un plugin hautement recommandé qui permet d'éditer des fichiers de configuration directement depuis l'interface web. Nous trouvons ici le fichier */config/configuration.yaml*, et nous ajoutons les lignes suivante :

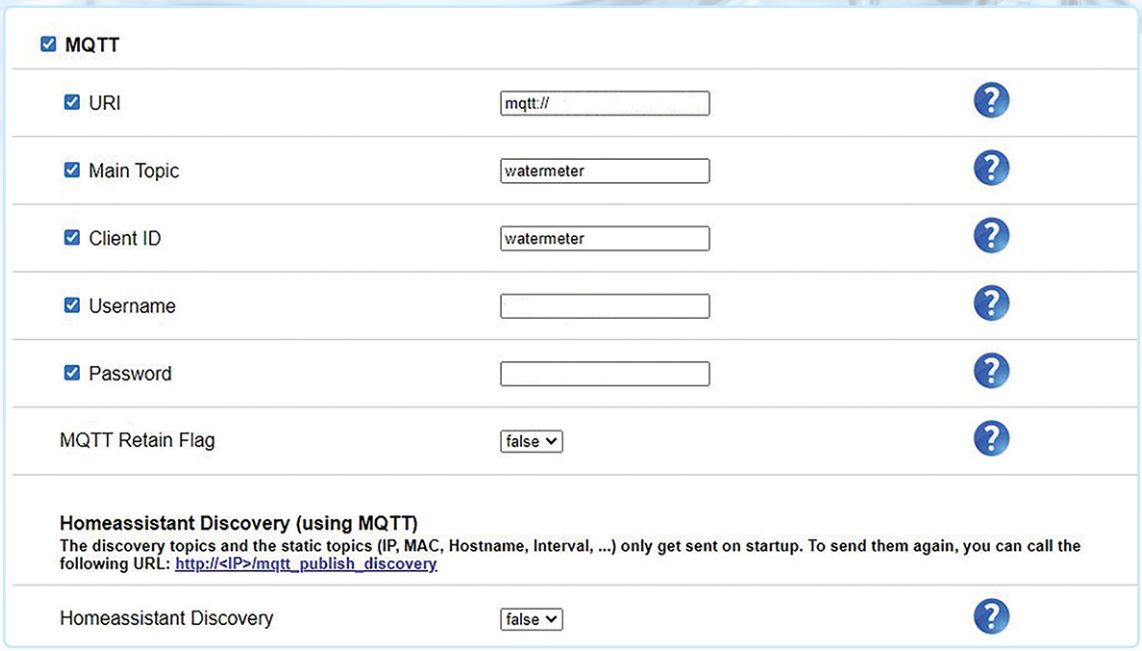

Figure 11. Menu de configuration MQTT.

```
sensor:
  - platform: mqtt
    name: "waterMeterMQTT"
    state_topic: "watermeter/main/raw"
    unique_id: watermeter_value
    unit_of_measurement: 'm³'
    state_class: total_increasing
    device_class: water # Needs HA 2022.11!
    icon: 'mdi:water-pump'
    availability_topic: watermeter/connection
    payload_available: connected
    payload_not_available: connection lost
```

Comme indiqué, si vous utilisez une ancienne version de Home Assistant, il se peut que vous n'ayez pas l'entité « water » et vous pouvez facilement utiliser « power » à la place. De même, au lieu d'utiliser « raw », vous pouvez utiliser « value » qui fournit l'information déjà traitée, au lieu de la forme brute.
Vous devez maintenant redémarrer le serveur. Allez dans *Settings*, *Server Controls* et sélectionnez *restart* pour afficher notre nouvelle entité. Une fois la connexion avec le serveur rétablie, cliquez sur *Edit dashboard* et ajoutez notre capteur comme le montre la **Figure 12**. Pour plus de détails, nous listons toutes les rubriques à utiliser dans le **Tableau 2**. Bien entendu, prenons comme exemple le fait que vous avez défini le sujet principal (main topic) comme étant le compteur d'eau (*watermeter*) pendant la phase de configuration.

Développement et modifications

Vous pouvez également modifier les sources afin de changer, par exemple, le comportement du serveur web. Dans un premier temps, vous devrez installer un client GIT. Pour ceux qui ne savent pas de quoi nous parlons, GIT est un système de gestion de versions extrêmement répandu. Pour cloner le projet à partir de la source distante :

```
git clone https://github.com/jomjol/AI-on-the-edge-device.git
cd AI-on-the-edge-device
git checkout rolling
git submodule update --init
```

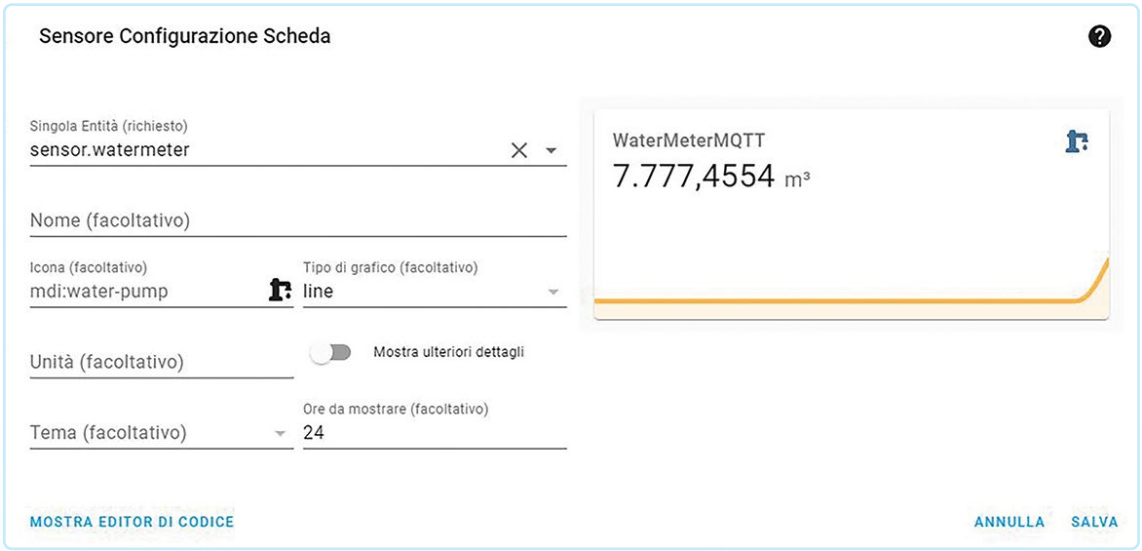

Figure 12. Fenêtre de configuration du capteur.

Tableau 2. Thèmes à utiliser avec MQTT.

Topic	Description
watermeter/MAC	Pour obtenir l'adresse MAC de l'ESP
watermeter/IP	Pour obtenir l'adresse IP de l'ESP
watermeter/Hostname	Pour obtenir le nom d'hôte
watermeter/Interval	Pour déterminer l'intervalle de temps entre deux prises de vues
watermeter/Connection	Pour connaître l'état de la connexion
watermeter/Uptime	Pour savoir combien de temps notre CPU a été active depuis le dernier redémarrage
watermeter/FreeMem	Pour connaître l'espace libre restant (en Ko)
watermeter/wifiRSSI	Pour connaître la qualité du signal
watermeter/CPUTemp	Pour connaître la température de la CPU
watermeter/main/error	Informe de l'état des erreurs de flux. En cas d'absence, il répondra « Pas d'erreur » (la réponse principale dépend de ce que nous avons défini).
watermeter/main/raw	Donne la valeur de la lecture avant le processus de post-acquisition
watermeter/main/rate	Donne la valeur de la consommation par minute
watermeter/main/rate_per_time_unit	Donne la valeur de la consommation par heure
watermeter/main/changeabsolut	Différence entre la valeur lue précédente et la valeur actuelle
watermeter/main/timestamp	Pour connaître l'horodatage de l'ESP
watermeter/main/Status	Pour connaître l'état de notre programme
watermeter/main/json	Connaître un ensemble de valeurs au format json comprenant les valeurs, valeur brute, l'erreur, le débit, l'horodatage, etc.
watermeter/GPIO/GPIO	Pour attester et connaître l'état d'une broche GPIO
watermeter/main/value	Fournit la valeur de la lecture après le processus de post-acquisition

Une fois que vous avez installé le client pour extraire le référentiel, nous vous recommandons également d'installer un environnement de développement intégré (IDE) pour obtenir une vue complète du projet. Comme IDE, nous recommandons d'utiliser Visual Studio Code, qui offre également PlatformIO, un plugin essentiel. Nous commençons par installer Visual Studio Code, en le téléchargeant sur le site web dédié [5]. Ensuite, nous ouvrons les extensions (*Extensions*) dans Visual Studio Code et recherchons *platformio*. Nous installons l'extension via le bouton approprié.

Dans notre environnement de développement, nous ouvrons le code source qui se trouve dans le dossier *AI-on-the-edge-device/code* folder. Nous ouvrons ensuite un terminal PlatformIO en cliquant sur l'icône d'extension, qui représente un extraterrestre, et en sélectionnant *New terminal*. Dans le terminal PlatformIO, pour compiler le code, nous tapons `platformio run --environment esp32cam`. Dans le dossier *AI-on-the-edge-device/code/.pio/build/esp32cam/* se trouvent les trois fichiers bin compilés : *bootloader.bin*, *partitions.bin* et *firmware.bin*. Tout le code est écrit en langage C++.

Évolution future

Le code est bien documenté et structuré, et fait l'objet d'une maintenance constante de la part des développeurs. Un développement futur possible serait d'envoyer la valeur lue à son téléphone portable à la fin de chaque mois, par exemple par le moyen d'un courrier électronique ou des réseaux sociaux. Une autre idée pour améliorer le projet pourrait être l'intégration du BLE (*Bluetooth Low Energy*). Cela permettrait de créer une plateforme où la lecture peut se faire via le smartphone, sans avoir à utiliser des périphériques externes supplémentaires.

VF : Laurent Rauber — 240213-B-04

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).

Produits

- **Carte de développement ESP32-Cam-CH340**
  www.elektor.fr/products/esp32-cam-ch340-development-board
- **Câble USB série TTL RS232 FTDI**
  www.elektor.fr/products/ftdi-serial-ttl-rs232-usb-cable

LIENS

[1] Projet GitHub, "AI on the Edge Device", page des publications : https://tinyurl.com/2n8da7jj
[2] Espressif Flash tools : https://espressif.com/en/support/download/other-tools
[3] Anaconda page de téléchargement : https://anaconda.com/download
[4] Boîtiers imprimables en 3D sur Thingiverse : https://thingiverse.com/thing:4573481
[5] Site de téléchargement de Visual Studio Code : https://code.visualstudio.com/Download
[6] Première partie de cet article : www.elektormagazine.fr/articles/releve-des-compteurs-d-eau-base-sur-l-ia-partie-1-integrez-votre-ancien-compteur-dans-l-ido