



# procédé sans fil de débogage de micrologiciels

## Tampon circulaire et serveur web sur l'ESP32

**Laurent Labbe (Hong Kong)**

Il n'est pas toujours facile d'enregistrer des données de débogage d'un microcontrôleur sur un PC. Un PC peut ne pas être disponible ou la carte de développement peut se trouver dans un endroit d'accès difficile. Ce projet d'Elektor Labs résout ce problème avec une mémoire tampon de forte taille pour les messages de débogage et une liaison wifi. Il est logé dans un boîtier imprimable en 3D et est hautement extensible.

Développer du logiciel pour les microcontrôleurs est souvent un défi. À quelques exceptions près, les microcontrôleurs offrent une procédure de débogage permettant de démarrer et d'arrêter le processeur puis d'examiner le contenu des variables et des registres, ce qui permet de diagnostiquer la cause d'un problème. Cependant, de nombreuses cartes maker, telles que l'Arduino Uno ou le BBC micro:bit, ne disposent pas d'un tel support natif, ce qui oblige le développeur à modifier et à télécharger le code de façon répétitive jusqu'à la découverte de la source du problème.

Mais cette procédure de débogage n'est pas toujours la panacée qu'on s' imagine. En cours de développement d'une application en temps réel utilisant l'Ethernet, le Wi-Fi ou l'USB, ou de commande de moteurs, arrêter le microcontrôleur au milieu du processus est

souvent impossible. Cela entraîne une interruption de la communication et peut même endommager les circuits de puissance si un MOSFET est figé dans son état passant.

La plupart des développeurs mettent à profit la présence d'un port de sortie série sur les cartes maker pour envoyer des messages de débogage vers une fenêtre terminal sur un PC via l'USB. En analysant la sortie après un test, le développeur peut suivre l'exécution du code à la trace, d'où le nom de « trace logicielle » donné à cette méthode de débogage. C'est aussi un moyen de déboguer des applications qui ne peuvent pas être arrêtées en cours de fonctionnement.

### Le Projet

Laurent Labbe, collaborateur régulier d'Elektor Labs, utilise souvent le port série pour suivre l'exécution du code embarqué, mais comme il ne dispose pas toujours d'un PC pour enregistrer les messages, il s'est mis à la recherche d'une autre solution. Armé d'un ESP32, d'un écran OLED et d'une imprimante 3D, il a mis au point le projet « Débogage par traçage sans fil » [1].

Cette solution utilise un ESP32 comme mémoire tampon pour les messages de débogage, connecté au contrôleur cible par une interface série ; la mémoire tampon est rendue accessible par wifi et un serveur Web local (**figure 1**).

La grande mémoire SRAM de l'ESP32 se prête à la réalisation d'un tampon circulaire géant. Les projets de Laurent utilisent généralement une faible vitesse de transmission de 9600 bauds (modifiable). Chaque fois qu'un caractère est reçu, il est inséré en tête du tampon. Avec sa taille de 65 530 octets, le tampon offre un large espace de stockage des messages de traçage. Serial1 est initia-

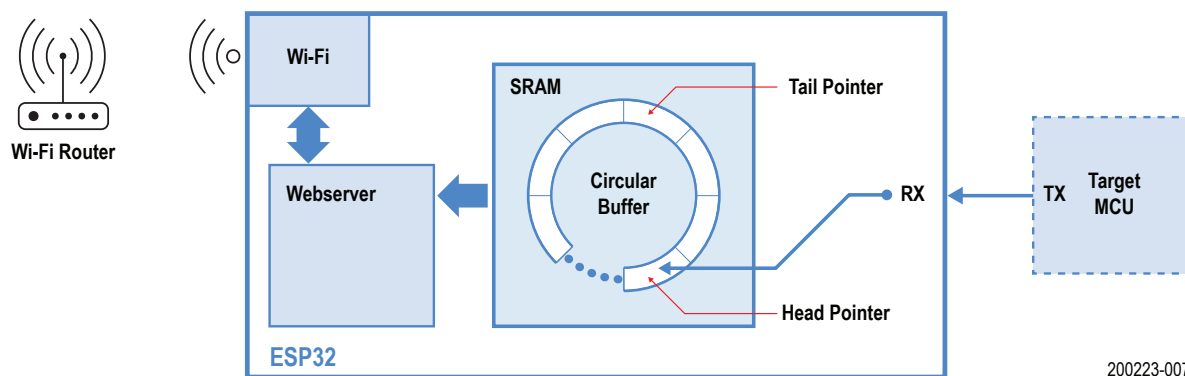


Figure 1. Ce schéma fonctionnel simplifié montre le mécanisme de stockage des données série entrantes dans le tampon circulaire. Les données sont disponibles sur demande par Wi-Fi sous forme d'une page Web.

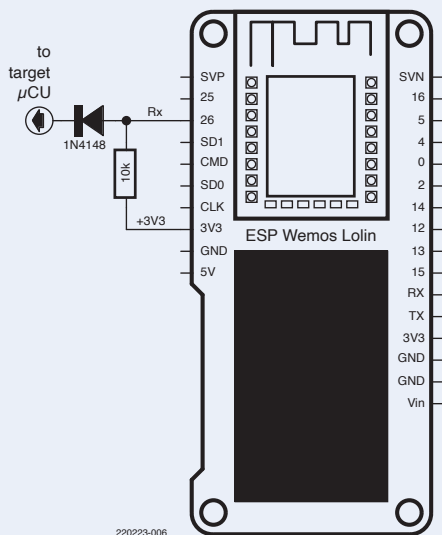


Figure 2. L'ESP32 fonctionnant sous 3,3 V, ce circuit permet de connecter un microcontrôleur 5 V.

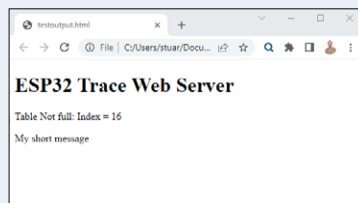


Figure 3. Un exemple de page web.

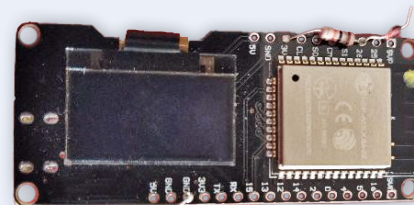


Figure 4. L'ESP32 complet avec le circuit adaptateur 5 V.



Figure 5. Une unité de débogage série sans fil complète logée dans son boîtier imprimé en 3D.

lisé pour la collecte de données en utilisant les broches 25 et 26 (TX et RX, respectivement), bien que seule la broche RX soit nécessaire. Comme l'ESP32 fonctionne sous 3,3 V, Laurent a inclus un circuit résistance/diode d'adaptation aux microcontrôleurs 5,0 V (**figure 2**).

C'est l'utilisation de la fonction Wi-Fi de l'ESP32 qui donne à ce projet son caractère hors du commun. Après s'être enregistré sur le réseau comme défini dans le code, l'ESP32 sert une page Web simple de présentation du contenu courant du tampon circulaire (**figure 3**), accessible par tout PC ou portable.

## Deuxième réseau : l'interface série et l'affichage

Quelques autres fonctions intéressantes sont également intégrées au code. Par exemple, la gestion d'un routeur wifi alternatif, (avec son SSID et son mot de passe). Si la première liaison échoue pour une raison quelconque, l'ESP32 tente automatiquement de s'y connecter. Cela améliore les chances de récupérer les messages de trace qui ont pu s'accumuler dans le tampon pendant des heures, voire des jours. Le serveur Web utilise le port 80 par défaut, mais un autre port peut être défini lors de l'initialisation.

Le tampon est circulaire : quand il est plein, les nouveaux messages entrants écrasent les plus anciens. La page Web alerte sur cette possibilité avant d'afficher le contenu du tampon circulaire du plus ancien au plus récent.

L'ESP32 émet également des messages via l'interface série (à travers l'USB), ce qui permet de déboguer le code de ce projet. Les messages de débogage reçus via `Serial1` sont émis vers `Serial` pour une visibilité immédiate. Toute donnée reçue sur l'interface `Serial` est également insérée dans le tampon circulaire.

Les informations sur l'adresse IP et d'autres détails pertinents sont transmises à un écran OLED géré par I<sup>2</sup>C et connecté aux broches 4 et 5. L'écran utilisé est le SSD1306 d'Adafruit avec leur pilote et leurs bibliothèques graphiques GFX.

Laurent a utilisé une carte WeMos Lolin32 OLED (**figure 4**, la carte est basée sur le module ESP32-WROOM-32 avec des circuits ajoutés)

pour réaliser l'interface USB-vers-UART et la fonction de connexion et de recharge d'une batterie LiIon/LiPo. Logé dans son boîtier imprimé en 3D, pour lequel un fichier CAO est disponible, cet outil de débogage par traçage sans fil peut être utilisé presque partout pour collecter les données de votre cible de manière autonome (**figure 5**).

## Options

La beauté de ce projet réside dans la combinaison de sa simplicité et de son extensibilité. La fonctionnalité du code est claire, ce qui permet aux développeurs expérimentés d'étendre et de modifier facilement ce projet. Par exemple, en étendant la SRAM disponible avec des composants externes ou en la remplaçant par une carte SD. La vitesse de l'interface série d'acquisition des données peut être modifiée ; elle pourrait aussi céder la place à une interface I<sup>2</sup>C ou SPI. Enfin, pour agrémenter vos messages d'un peu de formatage ou de couleur, il est facile d'insérer le HTML ou le CSS adéquat lors de la génération de la page Web. ◀

200223-04 VF : Helmut Müller

## Des questions, des commentaires ?

Envoyez un courriel à Elektor ([redaction@elektor.fr](mailto:redaction@elektor.fr)).



## PRODUITS

► **Module d'affichage OLED WeMos Lolin ESP32 pour Arduino (SKU 18575)**  
[www.elektor.fr/18575](http://www.elektor.fr/18575)

## LIENS

[1] Page du projet Elektor Labs: <https://www.elektormagazine.fr/labs/wireless-trace-for-debug>