

surveillance et débogage sans fil

Une solution pour Arduino, ESP32 et Cie

Peter Tschulik (Autriche)

Les cartes à microcontrôleur modernes, équipées d'un ESP32 ou d'un ESP8266 par exemple, sont, pour un prix modique, très performantes et d'un grand confort de programmation, notamment avec l'EDI Arduino. La fonction Over-The-Air (OTA) est particulièrement intéressante. Elle permet de télécharger facilement des mises à jour de programmes et des données via un chargeur d'amorçage qui utilise la communication par Wi-Fi, ce qui permet de se passer de la connexion physique de la carte au port USB d'un PC ou d'un ordinateur portable. Cependant, pour transmettre sans fil des données série (telles que les informations de débogage), il nous faut une autre solution.



Kit « Long Distance Wireless UART-RS232 ».

Les mises à jour logicielles d'appareils de l'Internet des Objets, donc qui fonctionnent à distance « sur le terrain », peuvent s'effectuer par voie hertzienne (OTA). Un bon tutoriel sur le sujet avec l'EDI Arduino est disponible à l'adresse [1]. Cette méthode présente toutefois un inconvénient : le moniteur série n'est pas pris en charge, de sorte que les informations de sortie et de débogage ne peuvent pas être transférées. Sur mon dernier projet, un système d'arrosage de jardin basé sur l'ESP32, j'ai eu un problème avec un capteur défectueux. Au lieu de tirer

un câble USB à travers la terrasse jusqu'à l'ordinateur dans mon salon, j'ai commencé à chercher une solution sans fil. Malgré des recherches intensives, je n'ai pas pu trouver de module standard qui réponde à mes exigences et j'ai donc entrepris de développer ma propre solution. Je me souvenais vaguement de systèmes qui permettaient de transmettre sans fil des données d'interface série ; en fait, j'ai rapidement trouvé ce que je cherchais auprès d'une entreprise d'Extrême-Orient. Le kit *Long Distance*

Wireless UART-RS232 [2] offre des taux de transfert allant jusqu'à 115 200 bauds avec des options de configuration et une portée allant jusqu'à 1 km. Pour 75 € la paire émetteur/récepteur, j'ai pensé que cela représentait un bon rapport prix/performance pour ce projet.

Adaptateur hôte-USB

Le choix d'un convertisseur USB/série s'est avéré beaucoup plus difficile. Bien qu'il soit possible d'appliquer le signal de données série du système d'arrosage distant directement au

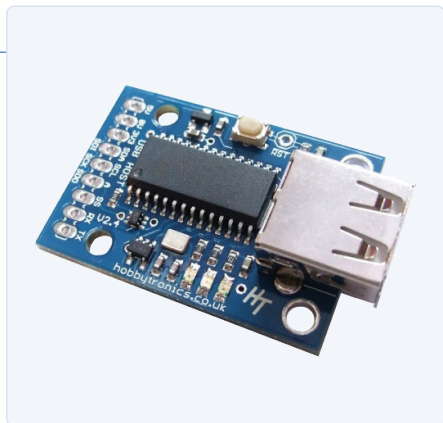


Figure 1. Petite par la taille, grande par le talent : la carte hôte-USB.

(Source : www.hobbytronics.co.uk)

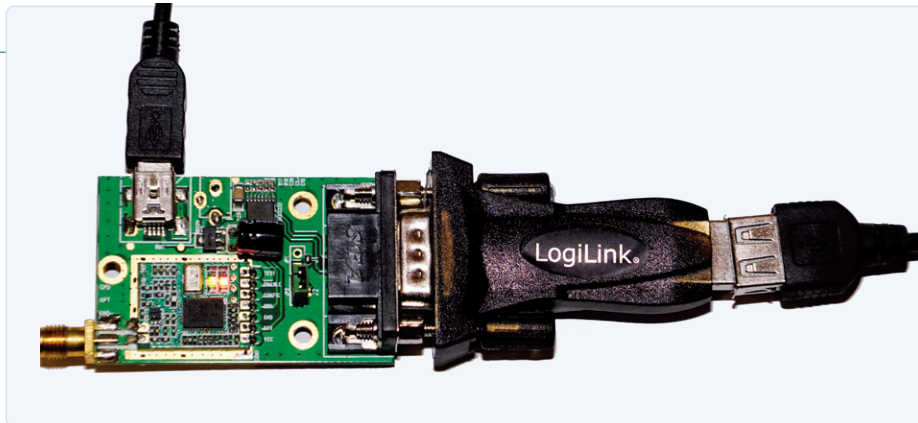


Figure 2. Il faut configurer les modules « Long Distance Wireless UART-RS232 ». Sur la photo, vous pouvez voir le cavalier en place et les LED verte et rouge.

module de transmission série, je souhaitais une solution plus universelle utilisant une connexion par port USB afin que la liaison puisse servir aussi facilement pour d'autres types de cartes de développement. Il faut pour cela un « adaptateur hôte-USB » qui se branche sur le port USB du site distant et communique avec la carte de développement comme le ferait un PC. Par chance, après quelques recherches intenses sur l'internet ([3] et [4]), j'ai repéré l'*USB-Host Shield* pour Arduino. Un fournisseur local avait l'article en stock, j'ai donc passé commande et quelques jours plus tard, j'étais occupé à le déballer sur mon bureau. Il ne m'a pas fallu longtemps pour découvrir un problème. Il y a de nombreuses plates-formes de développement différentes dans le monde

des makers et plusieurs marques différentes de puces d'interface USB. Les cartes ESP32 par exemple utilisent les puces CP210x ou FTDI, les cartes Arduino utilisent le contrôleur Atmel ou les puces FTDI. Malheureusement, les bibliothèques pour Arduino de l'*USB Host Shield* ne prennent en charge qu'un seul type de puce à la fois. Cela signifie que vous devez rendre le logiciel de l'*USB-Host Shield* configurable et trouver le type de puce associé à la plate-forme de développement que vous utilisez. Ce n'est pas vraiment une solution prête à l'emploi ! J'ai fini par trouver une carte d'interface hôte-USB en provenance du Royaume-Uni qui s'adaptait beaucoup mieux aux différentes marques de puce d'interface (fig. 1) [5].

L'intérêt majeur de cette carte, c'est qu'elle peut fonctionner avec une large gamme de périphériques tels que des clés de mémoire flash, des claviers, joysticks et souris USB, des manettes Dualshock PS3/PS4, et dispose de pilotes série pour FTDI, CP210X, PL2303, CH340/1 et CDC, des périphériques MIDI et des modems USB. Leur site web fournit la liste de tous les micrologiciels gratuits. Si vous connaissez exactement l'application pour laquelle vous voulez utiliser la carte, vous pouvez la spécifier lors de la commande et elle sera fournie avec le micrologiciel installé. Pour cette application, j'avais besoin du micrologiciel *USB Host - Serial Driver for FTDI, CP210X, PL2303, CH340/1 and CDC* [6] ; il prend en charge

toutes les puces USB utilisées par les différentes cartes de développement.

Pour programmer et configurer les modules UART RS232 sans fil ou la carte hôte-USB, il faut au minimum un convertisseur USB vers série, que l'on peut obtenir chez le même fabricant que pour les modules UART RS232 sans fil [8] ou par ex. chez Logilink [9]. Pour mon projet, j'avais besoin de deux convertisseurs de niveau RS232. J'ai choisi la bonne vieille solution du MAX232 [7].

Configuration de la liaison

Lorsque j'ai reçu le kit *Long Distance Wireless UART-RS232*, la façon dont je devais configurer les modules n'était pas claire pour moi. J'ai contacté le vendeur et reçu le lendemain un courriel contenant le logiciel, un manuel et un lien vers une vidéo de configuration [10]. Tout d'abord, j'ai dû sortir les modules de leur boîtier en plastique et souder une barrette à l'emplacement du connecteur J2 (une rangée de plots juste devant le connecteur RS232). J'ai ensuite placé des cavaliers sur les broches selon les instructions afin que le module entre en mode configuration. Ensuite, j'ai connecté le module au PC via l'adaptateur USB/série. Le module est alimenté en 5 V (par ex. une alimentation USB enfichable) via le connecteur mini-USB (fig. 2).

Si les deux LED s'allument, vous pouvez lancer le programme **HY-TRP Setting GUI.exe** et régler le bon port COM et la vitesse de transmission par défaut sur 9 600. Cliquez ensuite sur le bouton *Open COM* et lisez toutes les valeurs avec *Read All Settings*. Les valeurs sont définies comme indiqué à la figure 3. On peut trouver plus de détails dans le manuel [10]. Certains problèmes de fiabilité de la liaison ont été résolus en réduisant le débit en bauds de la liaison de transmission

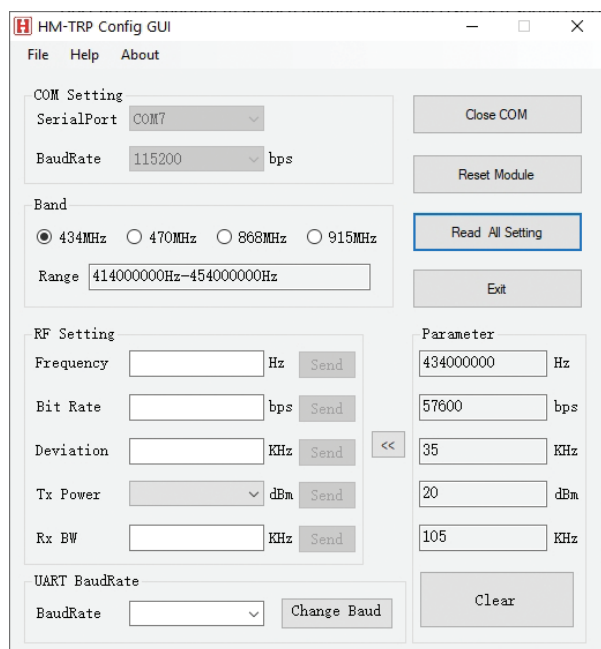


Figure 3. La configuration recommandée pour le module « Long Distance Wireless UART-RS232 ».

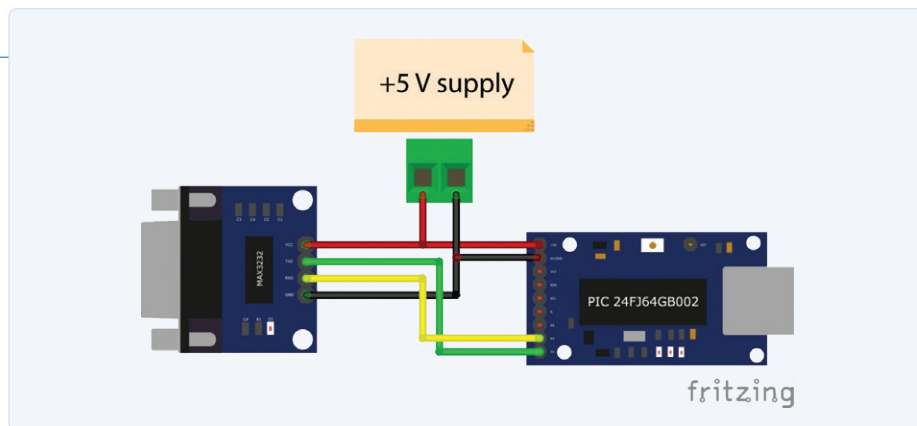


Figure 4. Câblage de la carte hôte-USB pour la configuration.

de 115 200 à 57 600. La carte hôte-USB possède un tampon qui tolère différentes vitesses de transfert. Les deux modules doivent être configurés de manière identique. Enfin, une fois la configuration terminée, n'oubliez pas de retirer les cavaliers.

Configuration de la carte hôte-USB

Pour programmer la carte hôte-USB avec le bon micrologiciel, référez-vous à la

description donnée en [5]. Beaucoup de programmes disponibles en téléchargement chez Hobbytronics peuvent être chargés sur la carte. Le micrologiciel adéquat peut être chargé sur la carte en utilisant les instructions données en [6], et vous y trouverez également une description des paramètres de l'interface de commande. Pour la configuration, la carte hôte-USB doit être connectée à la carte convertisseur de niveau MAX3232 comme indiqué dans la **figure 4**.

Le convertisseur RS232-TTL se connecte à l'adaptateur RS232-USB, qui à son tour se connecte au PC via un câble USB. Le circuit n'a plus besoin que d'être alimenté par une alimentation stabilisée de 5 V.

Vous pouvez maintenant lancer un programme de terminal (ici Hterm), sélectionner le bon port USB et régler la vitesse de transmission sur la valeur par défaut de 9 600 bauds. En cliquant sur le bouton *Connect* en haut à gauche, vous vous connecterez à la carte hôte-USB. En haut à droite, les options *Newline at* et sous la fenêtre principale *Send on Enter* sont changées en *CR + LF*. Si vous entrez maintenant *HELP* dans le champ de saisie de la zone *Input Control* sous la fenêtre principale, les paramètres à jour devraient être listés comme indiqué dans la **figure 5**.

Nous modifions maintenant les paramètres de notre application à l'aide du champ de saisie *Input Control*. Tout d'abord, nous pouvons entrer la commande *BAUD 57600* pour modifier le débit en bauds qui doit

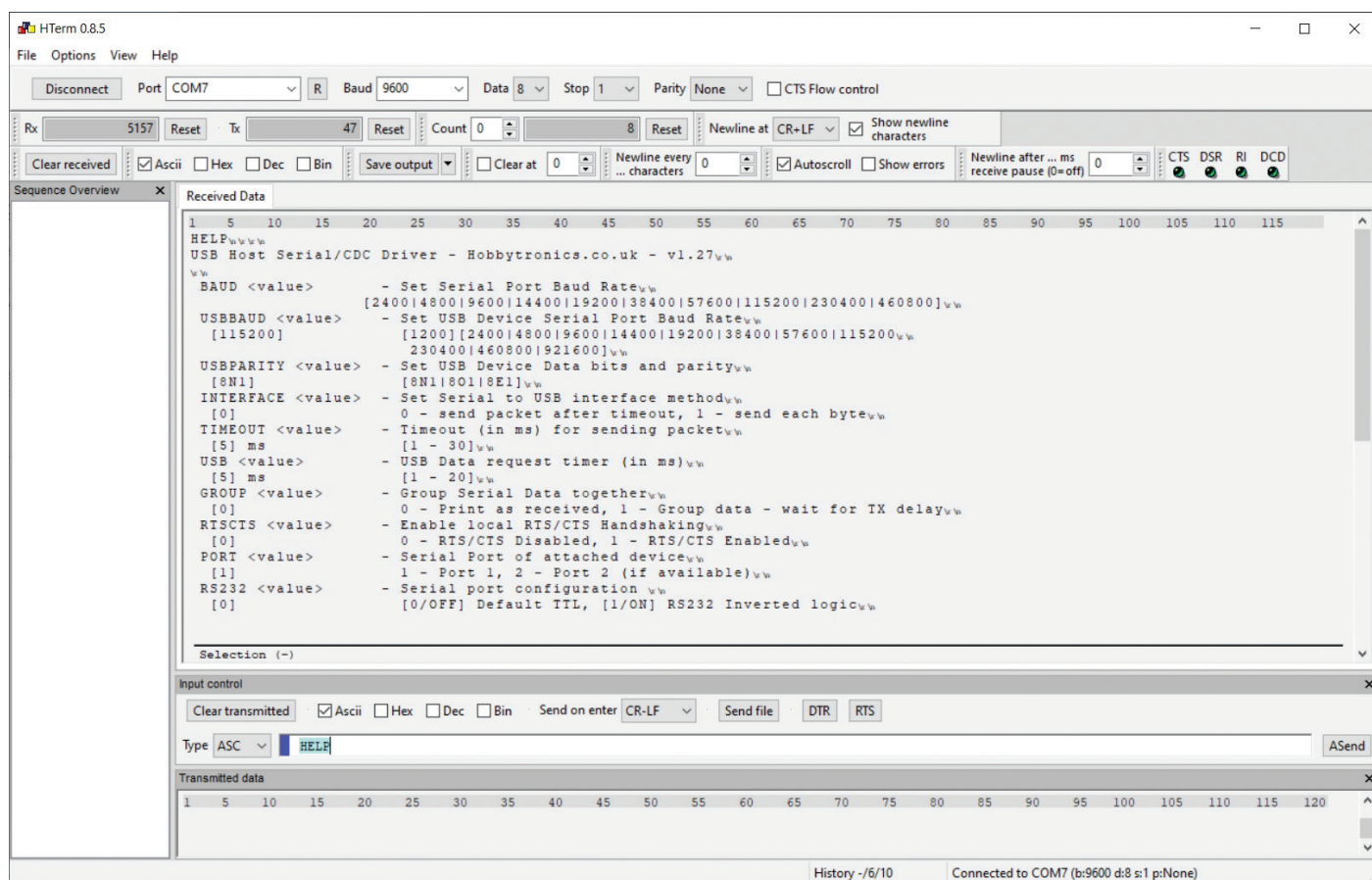


Figure 5. Configuration de la carte hôte-USB à l'aide de Hterm.

Tableau 1. Paramétrage dans HiTerm

BAUD	57600
USBBAUD	115200
USBPARITY	8N1
INTERFACE	0
TIMEOUT	5ms
USB	5ms
GROUP	0
RTSCTS	0
PORT	1
RS232	0

correspondre au débit en bauds défini pour les systèmes *Wireless UART*. Utilisez *Disconnect* puis *Connect* pour déconnecter et reconnecter le terminal au système, puis la commande *HELP* qui devrait vous permettre de vérifier la liaison avec la carte hôte-USB.

La commande *USBBAUD 115200* règle le débit en bauds de l'USB sur 115 200. Cela signifie que l'interface série de la carte Arduino/ESP32/ESP8266 connectée dans l'EDI Arduino doit toujours commencer par la commande *Serial.begin (115200)*.

On utilise les paramètres par défaut pour *USBPARITY*, *INTERFACE*, *TIMEOUT*, *USB*, *RTSCTS*, *PORT* et *RS232*. La valeur par défaut de l'option de regroupement des données série est OFF. La commande *GROUP 0* définit également ces propriétés à leur valeur par défaut. Le **tableau 1** donne un résumé des paramètres corrects. La configuration est ainsi terminée.

Câblage des modules

Avant de pouvoir câbler l'émetteur (du côté de la carte IdO sur le terrain) et le récepteur (du côté du PC), il est nécessaire de retirer du circuit imprimé le connecteur rond d'alimentation (à côté de la prise USB) sur les modules *Long Distance Wireless UART-RS232* afin de dégager les plots.

Du côté de l'émetteur, la prise mini-USB de la carte hôte-USB alimentera l'unité, voir le câblage à la **figure 6**. Pour connecter le module *Long Distance Wireless UART-RS232* à l'adaptateur RS232/TTL, vous devez utiliser un câble court terminé par des connecteurs SUBD-9

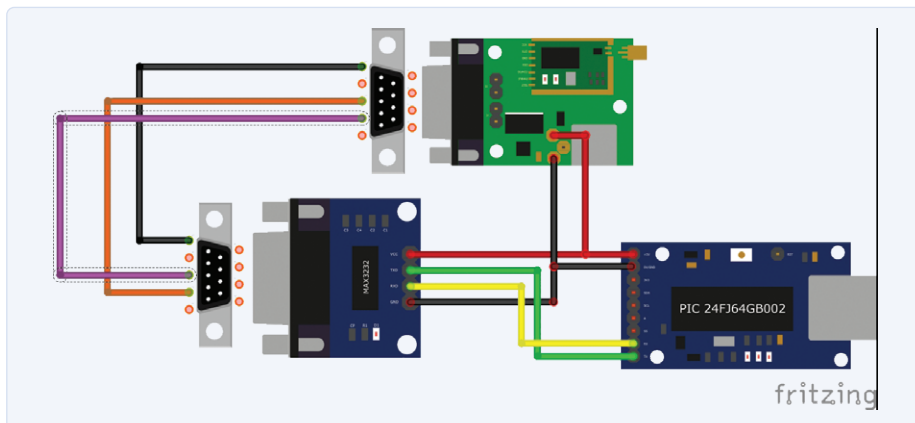


Figure 6. Câblage de l'émetteur à distance (emplacement de la carte IdO). La carte verte est le module sans fil. La carte IdO fournissant les données se connecte au port USB en bas à droite.

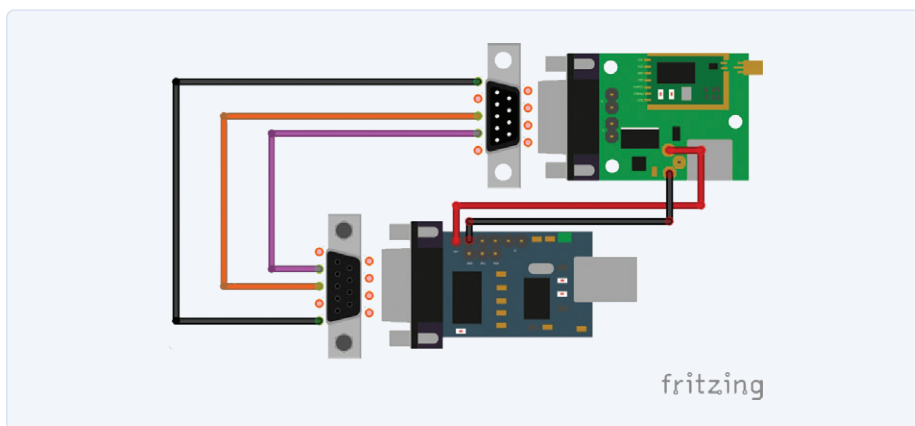
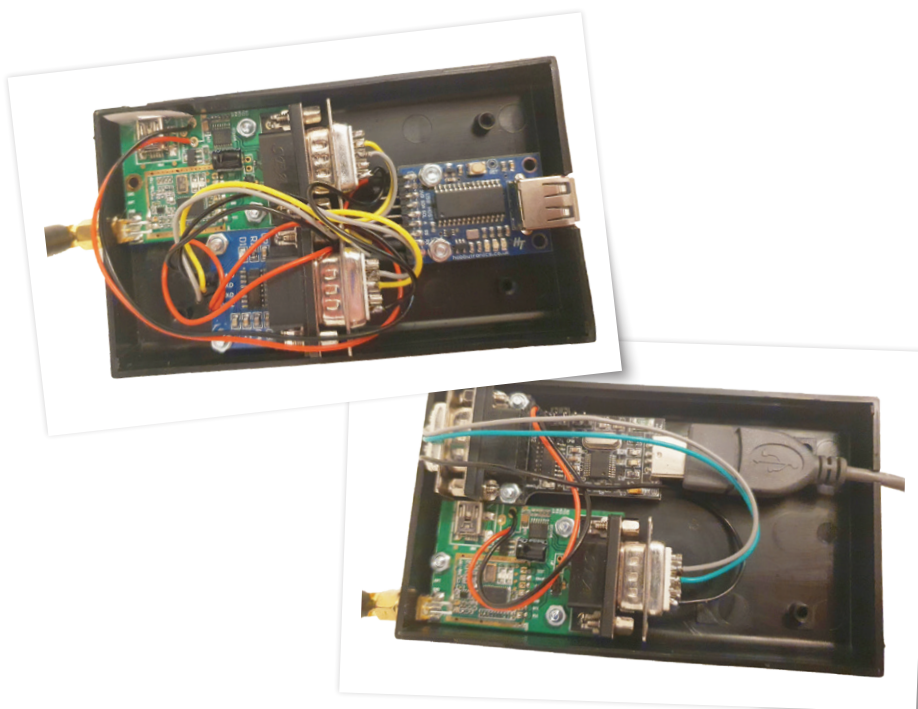


Figure 7. Câblage de la station réceptrice. Le PC est connecté en bas à droite par un câble d'extension USB.

mâles. Les fils des broches 2 et 3 sont croisés dans le câble comme le montre l'image.

Le module compact de [8] est utilisé comme convertisseur RS232/USB pour le récepteur et est câblé selon la **figure 7**. J'ai utilisé un câble d'extension USB entre le PC et l'unité de réception pour donner une certaine flexibilité dans le positionnement de l'unité à l'intérieur. Enfin, j'ai installé les unités d'émission et de réception dans deux boîtiers assortis. Les tests de mon système OTA ont montré la fiabilité de la transmission à travers deux étages avec des plafonds en béton armé. Dans l'ensemble, le système procure une installation de « câble USB virtuel » pratique, fiable et à longue portée, utile pour le développement et le débogage à distance d'applications IoT. ◀

200549-01

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (peter.tschulik@chello.at) ou contactez Elektor (redaction@elektor.fr).

Contributeurs

Idée, conception et texte : Peter Tschulik
Rédaction : Rolf Gerstendorf
Mise en page : Giel Dols
Traduction : Denis Lafourcade

Mode d'emploi

- › Réglez le débit de l'interface de communication série de la carte utilisée sur 115 200 bauds en utilisant `Serial.begin(115200);` dans l'EDI Arduino.
- › Une fois le micrologiciel transféré sur la carte distante, alimentez l'émetteur en 5 V via la prise mini-USB du module *Long Distance Wireless UART-RS232*.
- › Puis connectez-le à la carte distante à l'aide d'un câble USB court.
- › Branchez l'unité de réception sur un port USB libre du PC. Le récepteur est alimenté par le PC.
- › Démarrez l'EDI Arduino, choisissez n'importe quelle carte, sélectionnez l'interface USB de la carte du récepteur et démarrez le moniteur série.
- › Toutes les données envoyées par la carte devraient maintenant s'afficher.



PRODUITS

- › **Offre groupée : livre « The Complete ESP32 Projects Guide » + carte ESP32-DevKitC-32D**
www.elektor.fr/19897



- › **Compilation d'articles sur l'ESP32 et l'ESP8266 (PDF)**
www.elektor.fr/18516

LIENS

- [1] Les bases de la programmation Over The Air (OTA) de l'ESP32 avec l'EDI Arduino : <https://lastminuteengineers.com/esp32-ota-updates-arduino-ide/>
- [2] Module « Long Distance Wireless UART-RS232 » : <http://www.aliexpress.com/i/32242647632.html>
- [3] USB Host Shield de SparkFun : <http://www.sparkfun.com/products/9947>
- [4] USB-Host-Shield pour Arduino : <https://bit.ly/3fJQrTO>
- [5] Carte hôte-USB v2.4 : <https://www.hobbytronics.co.uk/usb-host-board-v24>
- [6] Hôte-USB – Pilote série pour FTDI, CP210X, PL2303, CH340/1 et CDC : <https://www.hobbytronics.co.uk/usb-host-serial>
- [7] Adaptateur RS232-TTL : <https://bit.ly/3peJH3r>
- [8] Câble adaptateur 2 fonctions USB COM Port DB9 TTL232 RS232 TTL 232 CH340T USB2.0 : <https://bit.ly/3fHUGQ3>
- [9] Adaptateur USB-2.0 vers série LogiLink : <https://bit.ly/3ma21Kg>
- [10] Instructions de configuration pour le module « Long Distance Wireless UART-RS232 », fichier HY035_HM-TRP-RS232.rar : <https://1drv.ms/u/s!AjrAGEbCBLsugxiqmd1FRRCJjic>