



Clemens Valens (Elektor)

Le marché mondial des microcontrôleurs est plus diversifié qu'on ne le pense. Passons en revue quelques-uns des microcontrôleurs et fabricants pas souvent apparus dans *Elektor*. Vous pourriez en trouver un ou plusieurs utiles pour un futur projet.

Pour un projet Elektor, le choix d'un microcontrôleur dépend surtout de la disponibilité à bas prix d'outils de développement du logiciel et de programmation de la puce, et de la possibilité pour les particuliers de les acquérir. Pour de nombreux amateurs d'électronique, cela restreint leur vision du marché mondial des microcontrôleurs. Le monde des microprocesseurs va bien au-delà des PIC, AVR, ARM ou ESP, si souvent au cœur de nos projets maison. Jetons un coup d'œil à quelques-uns de ces MCU qui échappent à notre perception.

Tout a commencé avec quatre bits

Introduit en 1971, l'Intel 4004 est considéré comme le premier microprocesseur produit commercialement (vous trouverez plus d'informations à son sujet dans le numéro spécial d'*Elektor Industry* consacré à 60 ans d'électronique [1]). Il s'agissait d'un composant à 4 bits. Avec ses puces périphériques, il formait la famille MCS-4. Elle a été suivie par la famille MCS-40 avec l'unité centrale 4040. Le premier microcontrôleur (pas un microprocesseur) de Texas Instru-

ments, le TMS1000 de 1974, avait également une architecture à 4 bits qui, comme le 4040, a animé de nombreuses calculatrices de poche. Dans un monde où les fabricants de microcontrôleurs semblent vouloir allonger autant que possible les mots de données (64 bits est devenu banal), vous seriez surpris du nombre de microcontrôleurs à 4 bits qui sont encore utilisés aujourd'hui. Mais pourquoi ? La réponse est probablement un mélange d'héritage, de consommation d'énergie et de coût.

Un MCU à 4 bits comprend moins de transistors que les puces avec des mots plus longs. Il consomme donc comparativement moins d'énergie, un bonus pour la durée de vie des piles. Moins de transistors signifie également moins d'espace, ce qui permet de caser un cœur de 4 bits dans un coin de la puce, là où un cœur plus grand ne tiendrait pas. La taille de la puce elle-même est plus petite, ce qui permet de réduire les coûts (dans quelle mesure peut-on se demander). Les applications en grande série telles que les calculatrices, les minuteries, les horloges et les montres, les ordinateurs de bicyclette, les jouets et les télécommandes,

etc. utilisent des MCU à 4 bits et ce depuis de nombreuses années. Les fabricants évitent généralement de modifier les produits qui marchent, ce qui explique la persistance sur le marché de ces composants.

Si vous désirez essayer un microcontrôleur à 4 bits, jetez un coup d'œil aux familles NY... de la société taiwanaise Nyquest. Les outils de développement peuvent être téléchargés gratuitement (**fig. 1**). Autres fabricants : le Suisse EM Microelectronic, le Chinois CR Micro, le Taiwanais Tenx Technology.

8051

Avant qu'ARM ne devienne le principal fournisseur de cœurs de microcontrôleurs pour presque tous les fabricants de semi-conducteurs de la planète, il y avait le 8051 à 8 bits. Créé par Intel en 1980 sous le nom de MCS-51, son cœur (**fig. 2**) a été cédé sous licence à plusieurs concurrents et a été intégré dans une foule de produits, dont beaucoup, avec de nombreuses variantes, sont encore fabriqués aujourd'hui. Quarante ans après l'introduction du 8051, de nouveaux produits sont toujours conçus à partir de ses dérivés.

De plus, les utilisateurs du 8051 ont développé une quantité de logiciels et de savoir-faire au cours de cette période, que l'arrivée d'un meilleur microcontrôleur n'a pas rendus caducs.

Enfin, le cœur du 8051 est devenu quasiment gratuit, ce qui en fait une option intéressante pour les fabricants de semi-conducteurs tentés par la production de composants à très bas coût. Ils ne le mentionnent pas toujours dans la fiche technique, mais s'il est indiqué quelque chose comme « 1T instruction cycle », il y a fort à parier qu'il s'agit d'un dérivé du 8051. Le 8051 original consommait 12 cycles d'horloge (appelés « 12T ») pour la plupart des instructions, alors que les plus modernes n'en nécessitent qu'un seul (d'où le « 1T »). En plus de demander moins de cycles d'horloge par instruction, l'exécution du programme est également (beaucoup) plus rapide car certains de ces composants modernes fonctionnent à des fréquences allant jusqu'à 450 MHz au lieu des 12 MHz de l'original. Les MCU 8051 modernes sont donc à la fois puissants et bon marché.

Aujourd'hui, le principal inconvénient du 8051 est sans doute son incompatibilité avec les langages de programmation modernes comme le C/C++, en raison de son étrange structure de mémoire. Pour en tirer le meilleur parti, il faut une chaîne d'outils commerciale Keil ou IAR, ou équivalente. La populaire chaîne d'outils GCC a été portée sur toutes sortes de microcontrôleurs, mais pas sur le 8051. La chaîne d'outils libre SDCC fait une bonne partie du travail, mais c'est loin d'être parfait. Bien sûr, on peut toujours utiliser l'assembleur. Vous préférez le Pascal ? Voyez le Turbo51 [2].

Vous pouvez trouver des MCU basés sur le 8051 dans des produits bon marché de grande série tels que des jouets, des claviers et des souris de PC, des broches à dents, des appareils ménagers, des télécommandes, etc., surtout originaires d'Asie où le 8051 semble très populaire. Silan, SiGma Micro, SinoWealth, Silicon Laboratories, Sonix, STC et SyncMOS (pour ne citer que ceux avec un « S » initial) en fabriquent tous.

Si vous êtes tenté d'expérimenter avec une variante moderne du 8051, jetez un coup d'œil à la famille CH55x de WCH. Pas chers, mais seulement documentés en chinois, ils

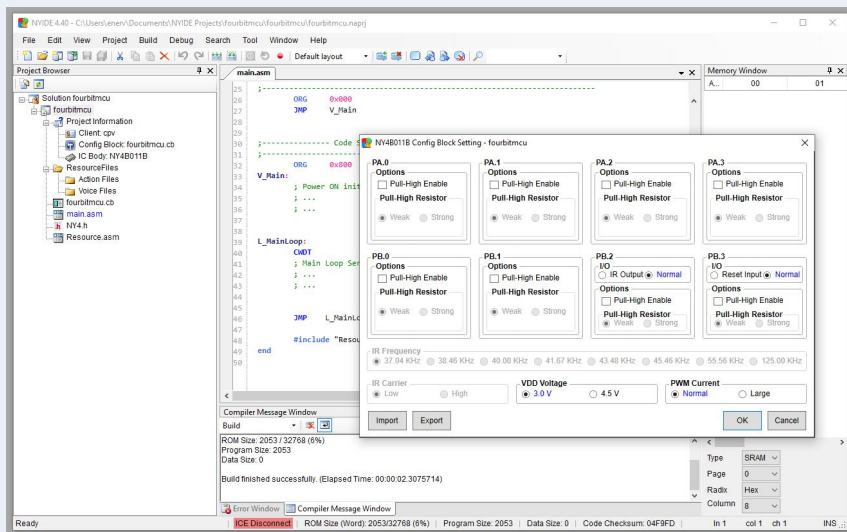


Figure 1. Le NYIDE 4.40 de Nyquest montre que les microcontrôleurs à 4 bits peuvent aussi avoir des EDI modernes.

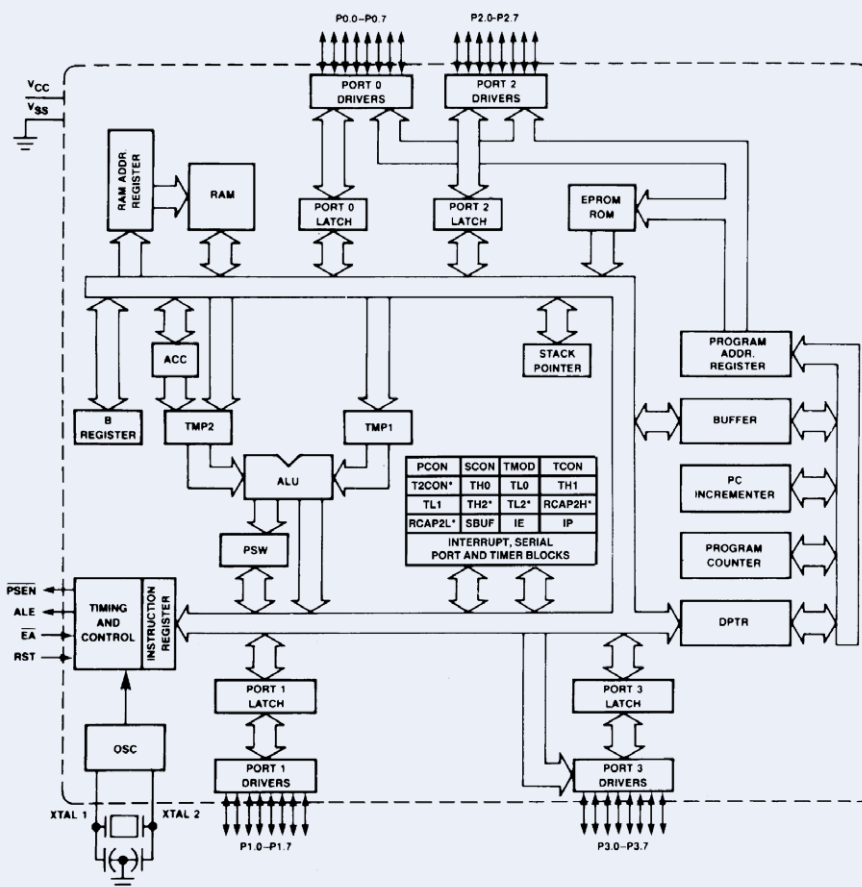


Figure 2. Voici l'architecture de base de ce qui pourrait bien être le cœur de microcontrôleur le plus utilisé au monde, le 8051. (Source : Intel)



Figure 3. Le Commodore 64, l'un des célèbres ordinateurs de salon des années 1980, contient un 6502. (Source : ralfsfotoseite @ Pixabay)

disposent d'une interface USB qui facilite leur programmation et sont utilisés dans des projets ouverts. En faire un Arduino ? Voir par exemple [3].

Vieux de la vieille et survivants

À côté du 8051 de 1980, il existe encore quelques autres cœurs de processeurs de cette période, notamment le 6502 et le Z80, mais aussi le 68000 de Motorola (maintenant NXP), un peu plus récent. Le Z80 et la version CMOS du 6502, le W65C02, sont toujours fabriqués et activement supportés par leurs créateurs Zilog et WDC (WDC a créé le W65C02, pas le 6502). La survie du 68000 semble surtout due aux besoins de maintenance de vieilles applications (mais qui sait combien de sociétés en ont pris une licence ?).

6502

Le 6502 a été utilisé dans plusieurs des premiers ordinateurs célèbres comme le Commodore 64, l'Apple II et le BBC Micro, et a connu un grand succès (fig. 3). Sa version CMOS améliorée et à faible consommation est toujours d'actualité, bien qu'assez chère. La raison en est que la plupart des utilisateurs ne prennent une licence que pour l'utiliser dans des FPGA, des ASIC et similaires. Comme tout est confidentiel, il est difficile de savoir de quels composants il s'agit.

Cependant, WDC produit des puces embarquées que vous pouvez essayer. Un exemple est le microcontrôleur W65C265S avec un CPU W65C816S à 16 bits qui est entièrement compatible avec le W65C02S à 8 bits et qui fonctionne à partir de 1,8 V seulement. Il existe aussi des modules à contrôleur, et même une carte sœur avec des connec-

teurs Seeed Studio Grove, Sparkfun QWIIC et MikroE Click.

Z80

Le Z80 est un autre processeur à succès de la fin des années 70 et du début des années 80. Zilog en a développé le cœur en 1975 et il est toujours en production. Sa licence a été accordée à de nombreux fabricants qui l'ont copié et cloné dans le monde entier, ce qui a donné naissance à une énorme base d'utilisateurs. Plusieurs familles ont vu le jour, comme l'eZ80 qui fonctionne avec des horloges allant jusqu'à 50 MHz, ou les Z8 et eZ8 Encore!, qu'on retrouve notamment dans la ligne de produits ZMOTION, une famille de MCU optimisés pour la détection de mouvement PIR.

Si vous voulez mettre les mains dans

le cambouis, essayez le Z8FS040BSB avec ses 4 Ko de mémoire flash et ses cinq broches GPIO dans un boîtier SOIC à 8 pattes. Notez que le compilateur gratuit SDCC supporte plusieurs MCU basés sur le Z80, comme ceux de Rabbit (maintenant Digi) et de la Nintendo Gameboy.

Composants à très faible coût

De nombreux produits électroniques contenant des microcontrôleurs sont fabriqués en très grande série. Pensez aux appareils électroménagers, aux horloges, aux brosses à dents électriques, aux e-cigarettes, aux testeurs de coronavirus, aux cartes à puce, aux détecteurs de fumée, aux jouets, etc. (fig. 4). Pour avoir une idée des chiffres : les consoles de jeu comme la Gameboy, la Wii et la Switch de Nintendo ont toutes dépassé les 100 millions d'unités vendues. Imaginez ce que sont les chiffres pour les cartes à puce, par exemple. Économiser un centime sur le coût d'un tel produit est énorme et il existe donc un grand marché pour les microcontrôleurs ultras bon marché. Parmi leurs fabricants, certains ont attiré l'attention des amateurs d'électronique, dont Padatauk, MDT et Holtek.

Padatauk

Padatauk fabrique des MCU à trois centimes, à programmation unique (OTP) et à base de flash, dont la particularité est l'architecture



Figure 4. Quelques exemples d'applications rendues possibles par les microcontrôleurs à très bas coût. (Source : Holtek.com)

basée FPP (Field-Programmable Processing units). Il s'agit de bancs de registres avec un compteur de programme, un pointeur de pile, un accumulateur et un registre de drapeaux, qui permettent un changement de contexte rapide, bien utile, par exemple pour le traitement des interruptions et le multitâche. Cela rappelle un peu le banc de quatre registres du 8051. Cependant, comme la plupart de leurs produits n'ont qu'un seul FPP (certains en ont deux, le PFC460, quatre et le MCS11, huit), ce ne sont que des MCU de base.

Le PMS150 est un bon point de départ. On trouvera une excellente présentation de ces composants sous [4]. SDCC supporte les PDK14 et PDK15. Pour le PDK13, c'est en cours.

MDT

Comme déjà mentionné, MDT, alias Micon Design Technology, fabrique des clones ou des dérivés des PIC de Microchip. La popularité des PIC chez les faiseurs a attiré une certaine attention sur les produits MDT. Cependant, au cours de mes recherches pour cet article, leur site web a soudain disparu. Une recherche sur l'internet concernant les composants MDT produit plusieurs résultats pour le craquage de MCU et les services de rétro-ingénierie des produits MDT, entre autres, ce qui suggère qu'ils sont largement utilisés.

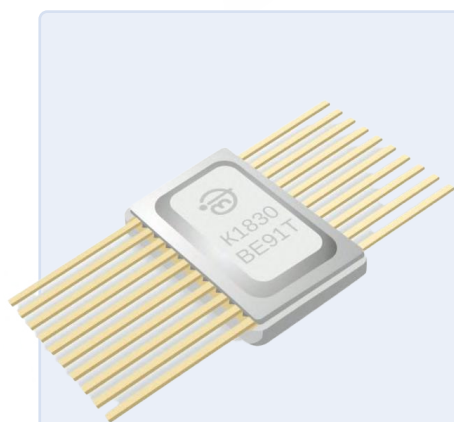


Figure 6. Le K1830BE91T de NIIET possède un cœur 8051 et est fonctionnellement équivalent au AT89C2051 de Microchip. (Source : <https://niiet.ru>)



Figure 5. La carte NuMaker Uno de Nuvoton possède un module de programmation/débogage NuLink détachable. (Source : <https://danchouzhou.blogspot.com>)

Holtek

Quelques produits Holtek ont été utilisés par le passé dans des projets Elektor. Il ne s'agissait pas de microcontrôleurs, mais de décodeurs de clavier et de canaux RC. Cependant, Holtek fabrique également des MCU, dont beaucoup sont basés sur le 8051 ou les ARM Cortex-M0 et M3 ainsi que sur leur propre cœur. Comme chez beaucoup de fournisseurs de MCU à bas prix, la gamme des produits Holtek est divisée en MCU d'usage général (« type I/O ») et spécifiques. La documentation est bonne et l'EDI HT-IDE3000 (assembleur et C) est gratuit (bien que difficile à trouver), mais un programmeur Holtek est nécessaire. Un composant que j'ai trouvé intéressant est le MCU analogique HT66F4550 qui intègre deux amplis ops et possède une sortie audio.

Et les géants d'Asie ?

Elektor a publié des centaines de projets basés le plus souvent sur des microcontrôleurs PIC ou AVR de Microchip (anciennement Atmel), ou ESP d'Espressif, ou un MCU avec un cœur ARM de chez NXP ou ST. Le MSP430 de Texas Instruments a également fait quelques apparitions. À l'exception d'Espressif, tous ces fabricants sont européens ou américains. Cela témoigne d'un biais, car il y a en Asie beaucoup de grands fabricants de MCU.

Renesas

L'un des plus grands, sinon le plus grand fabricant asiatique de semi-conducteurs est Renesas, formé à partir de divisions de NEC, Hitachi et Mitsubishi. Selon certains, c'est même le premier fournis-

seur mondial de MCU. Les lecteurs de longue date d'Elektor se souviennent peut-être des séries d'articles sur les R8C et R32C/111 d'il y a une quinzaine d'années [5]. La récente famille de microcontrôleurs à 32 bits RX671 est spécialisée dans le pilotage en temps réel et l'interface homme-machine (HMI) sans contact par détecteurs de proximité et reconnaissance vocale, l'idéal pour les interfaces HMI hygiéniques modernes. Renesas fournit également de nombreuses cartes de développement et d'évaluation que je vous encourage à essayer. N'hésitez pas à nous faire part de vos trouvailles.

Nuvoton

Issu de Winbond en 2008, Nuvoton a acquis la division des puces moribonde de Panasonic en 2020. L'entreprise propose une large gamme de MCU à cœur 8051 et ARM, ainsi que quelques composants à cœur propriétaire. Contrairement à de nombreux concurrents, Nuvoton ne dispose pas de sa propre chaîne d'outils. Pour les dispositifs 8051, ils proposent Keil et IAR tandis que pour ARM, ils utilisent Eclipse. Sur GitHub, vous pouvez trouver de l'aide pour utiliser SDCC avec certains composants Nuvoton. La carte NuMaker Uno (fig. 5) est un bon début pour utiliser Nuvoton. Avec son contrôleur NuMicro NUC131 ARM Cortex-M0, elle est compatible Arduino. Elle comprend également un module débogueur / programmeur Nu-Link détachable, utilisable avec d'autres modules. Le support logiciel est disponible sur GitHub (OpenNuvoton). Consultez le dépôt NuMaker pour le support de mbed Arduino, MicroPython et autres.



Des MCU russes ?

Pour compléter cet article, j'ai voulu ajouter quelques informations sur les microcontrôleurs russes. Malheureusement, je ne lis pas le russe, et la plupart des sites web sont en russe, ce qui complique la recherche d'informations utiles. Je suis tombé sur Milandr, Mikron et le fabless Syntacore qui font tous des contrôleurs basés RISC-V. Selon [6], Milandr a pris une licence ARM, mais n'en mentionne pas d'applications sur son site.

NIJET fabrique le K1921VK01T destiné aux applications de commande de moteurs et de mesure intelligente. Il est construit autour d'un noyau ARM Cortex-M4F. OpenOCD propose un support pour ce MCU. En octobre 2021, NIIET a annoncé un contrôleur basé sur RISC-V pour remplacer les STM32- et MSP430 qui sont actuellement utilisés dans les « équipements civils » (comme ils les appellent) en Russie. Ils ont également des MCU RISC à 8 et 16 bits et quelques MCS-51 (Intel 8051) et MCS-96 (Intel 80196) (fig. 6).

Un monde d'options

Dans cet article, nous avons passé en revue quelques microcontrôleurs et fabricants rarement utilisés dans les projets Elektor, mais qui constituent pourtant une part importante du marché mondial des MCU. Bien sûr, cet article est loin d'être exhaustif et certains composants ou fabricants intéressants ont pu être oubliés. Pendant ma recherche pour cet article, j'ai compilé une liste de plus de 50 fabricants de microcontrôleurs en activité, et je suis sûr qu'il y en a beaucoup d'autres. Si vous en connaissez d'autres, ignorés, mais intéressants, que vous aimeriez partager avec les autres lecteurs, faites-le-moi savoir. 

210630-04

Contributeurs

Idée et texte : Clemens Valens
Rédaction : Jens Nickel, C. J. Abate
Mise en page : Harmen Heida
Traduction : Helmut Müller



Le 4004 et le 4040 ont tous deux été conçus par Federico Faggin. Quels autres processeurs célèbres a-t-il également conçus ?

Intel 8080, Z80, 8088

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (clemens.valens@elektor.com) ou contactez Elektor (redaction@elektor.fr).



PRODUITS

- > Livre « Maîtrisez les microcontrôleurs à l'aide d'Arduino (3^e édition) », Clemens Valens, Elektor
www.elektor.fr/18064
- > Microcontrôleur Raspberry Pi RP2040
www.elektor.fr/19742
- > Livre en anglais, « ARM Microcontroller Projects », Elektor
www.elektor.fr/17620

LIENS

- [1] Stuart Cording, « The Birth of the Microprocessor », Elektor Industry, 03/2021 : www.elektormagazine.com/magazine/elektor-241/60042
- [2] Pascal pour 8051 : <https://turbo51.com/>
- [3] CH55xduino : <https://github.com/DeqingSun/ch55xduino>
- [4] Padouk PMS150 : <https://jaycarlson.net/2019/09/06/whats-up-with-these-3-cent-microcontrollers/>
- [5] Gunther Ewald, « R8C et compagnie », Elektor, 01/2006 : www.elektormagazine.fr/magazine/elektor-200601/10319
- [6] Microcontrôleurs russes : <https://geek-info.imtqy.com/articles/M4836/index.html>

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

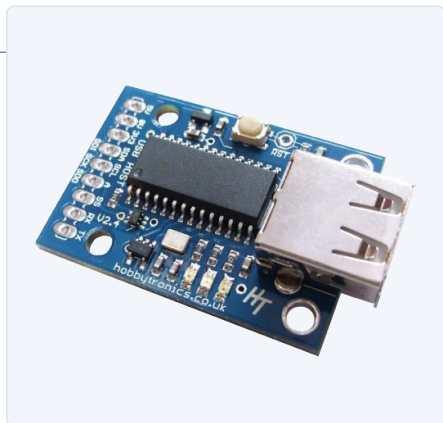


Figure 1. Petite par la taille, grande par le talent : la carte hôte-USB.

(Source : www.hobbytronics.co.uk)

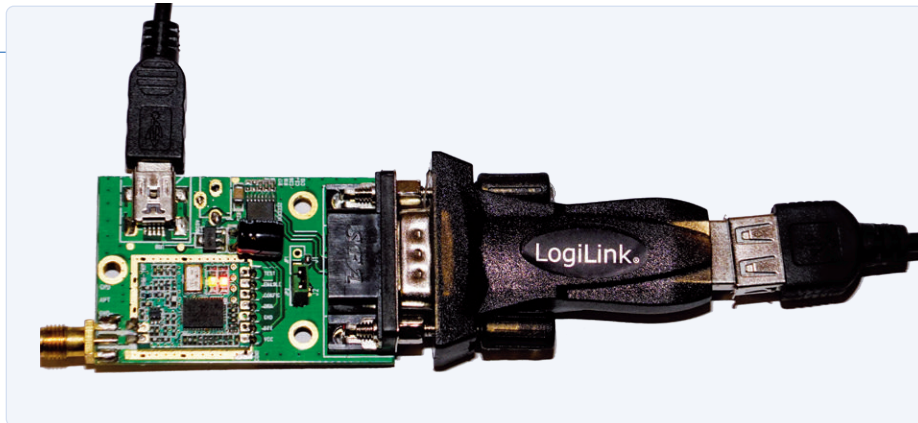


Figure 2. Il faut configurer les modules « Long Distance Wireless UART-RS232 ». Sur la photo, vous pouvez voir le cavalier en place et les LED verte et rouge.

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

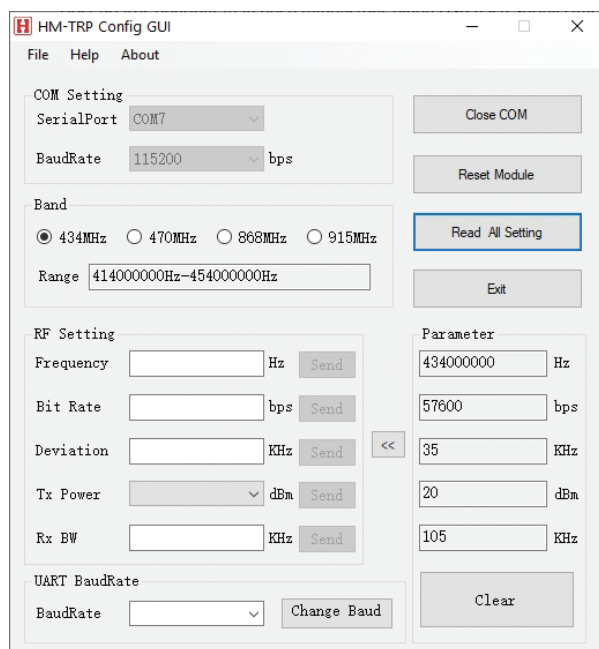


Figure 3. La configuration recommandée pour le module « Long Distance Wireless UART-RS232 ».

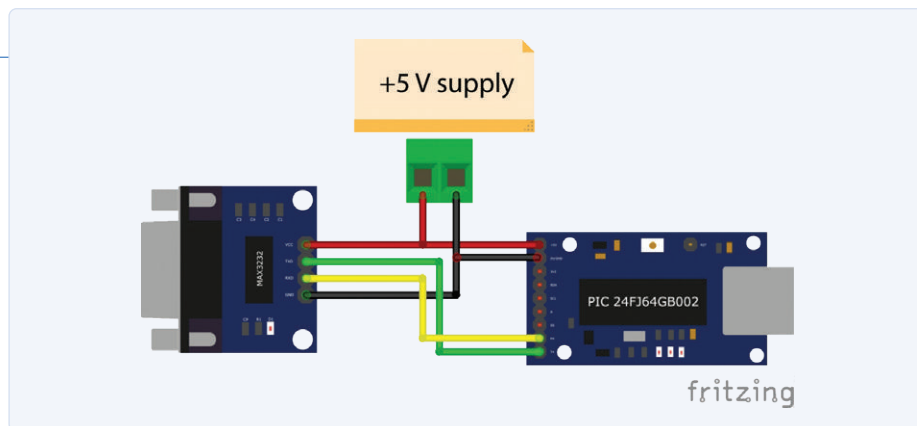


Figure 4. Câblage de la carte hôte-USB pour la configuration.

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

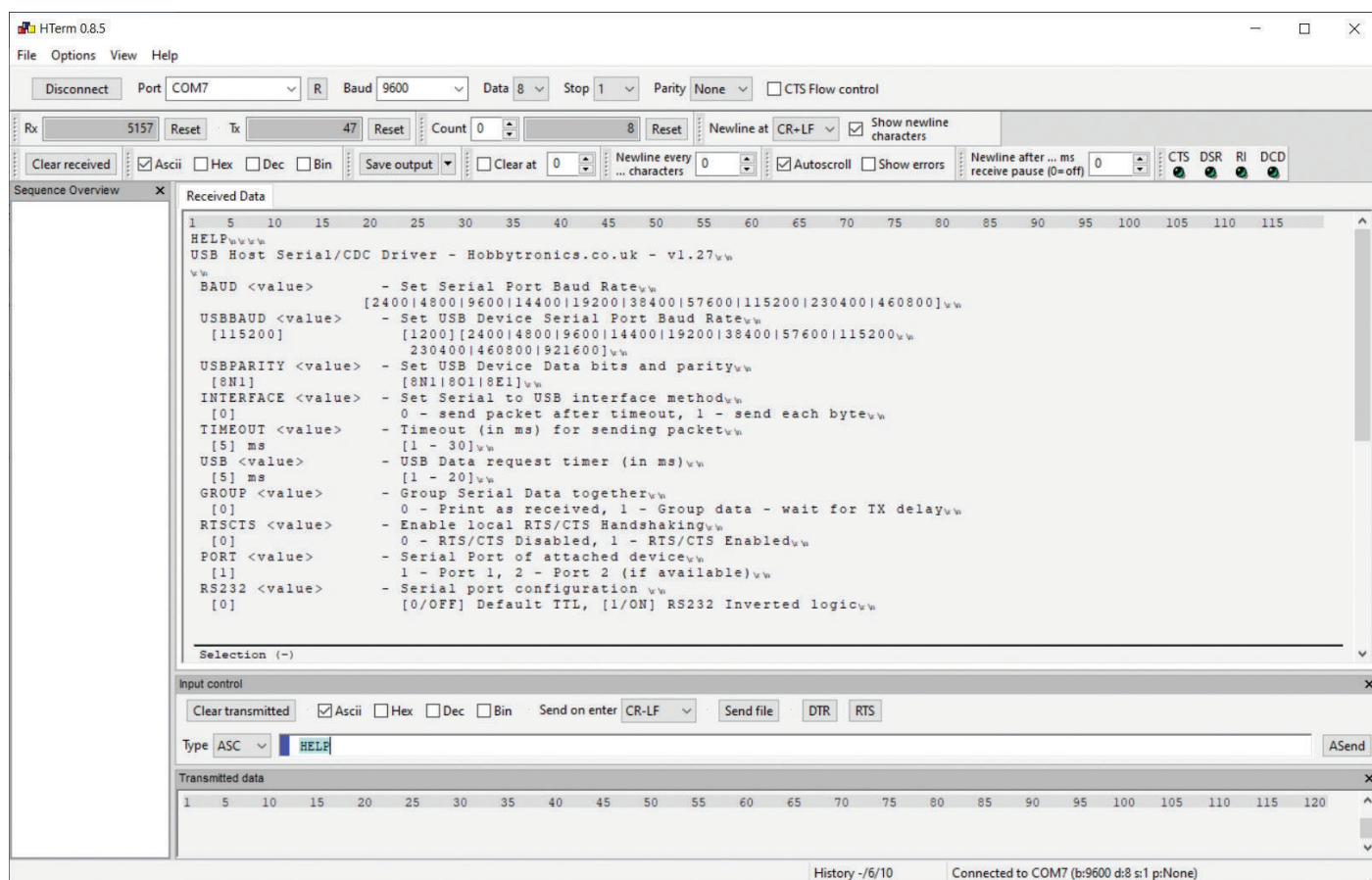


Figure 5. Configuration de la carte hôte-USB à l'aide de Hterm.

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

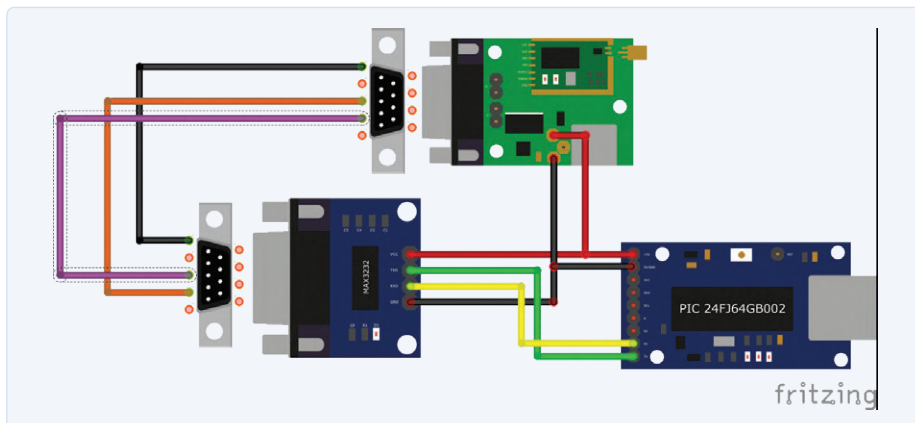


Figure 6. Câblage de l'émetteur à distance (emplacement de la carte IdO). La carte verte est le module sans fil. La carte IdO fournissant les données se connecte au port USB en bas à droite.

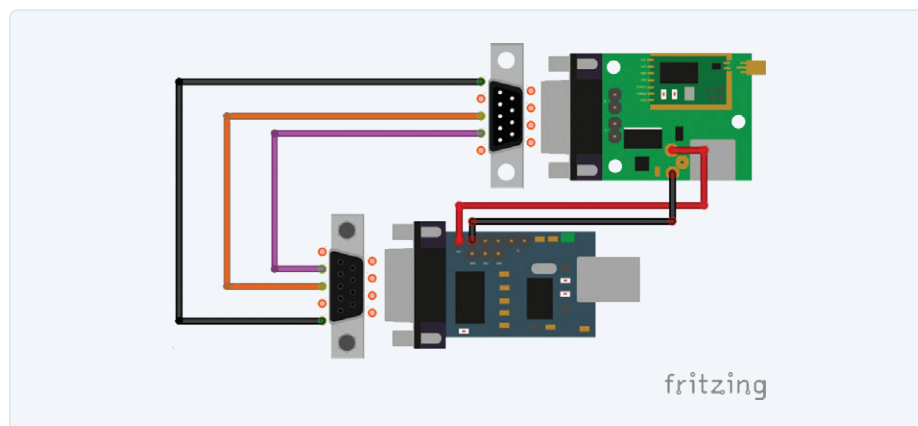
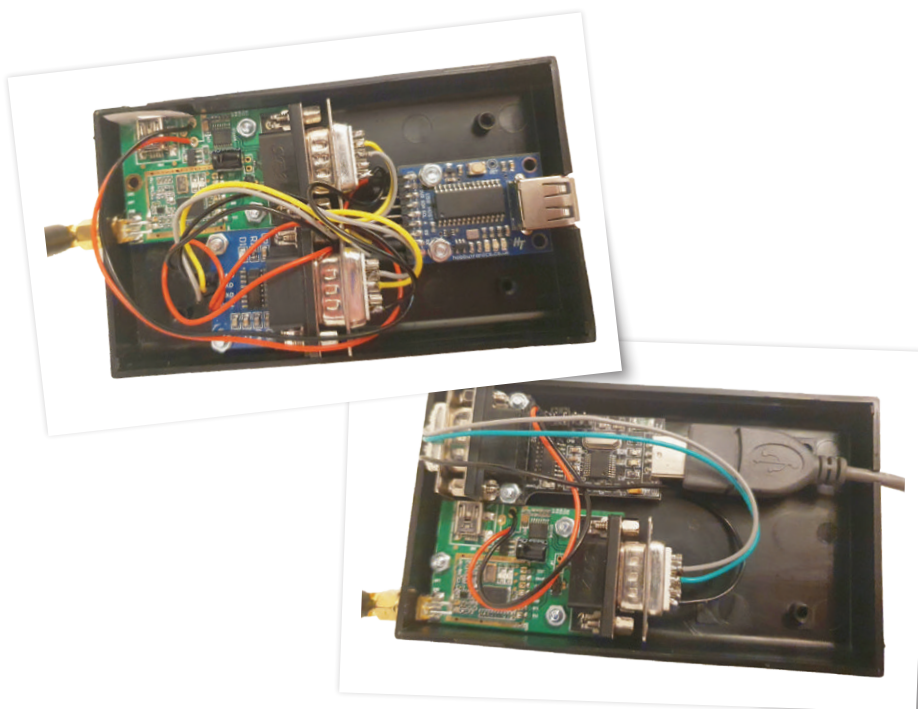


Figure 7. Câblage de la station réceptrice. Le PC est connecté en bas à droite par un câble d'extension USB.

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>