

l'écriture des croquis Arduino en net progrès



Alessandro Ranellucci

Stuart Cording (Elektor)

L'EDI Arduino 2.0 innove en préservant l'acquis Arduino. Alessandro Ranellucci d'Arduino vous invite à découvrir ce nouvel EDI en détail.

Pour une majorité d'utilisateurs Arduino, le 1er contact avec la programmation fut l'environnement de développement intégré (EDI) d'Arduino. Au fil de l'écriture et des « téléversements » de leurs croquis, les débutants ont passé des heures à décrypter les messages de sortie de l'écran à bordures vertes. Depuis son lancement en 2005, l'EDI n'a cessé de gagner en puissance : toujours plus de cartes prises en charge, accès amélioré aux bibliothèques partagées et même un traceur graphique de données. Cependant, la compétence des amateurs et le nombre de développeurs professionnels allant croissant, l'absence de débogage se faisait cruellement sentir. Pour en avoir le cœur net, j'ai récemment rencontré Alessandro Ranellucci, responsable « *Maker business, Open Source & Community* » chez Arduino.



L'EDI Arduino 2.0 garde la clarté de l'opus précédant et offre une foule de nouvelles fonctionnalités aux développeurs pro et aux amateurs éclairés.

Une réécriture complète

En 2018, l'équipe Arduino prit une décision cruciale qui allait non seulement changer les capacités de l'EDI, mais aussi rafraîchir l'ensemble de la base de code. Un bloc monolithique de code Java constituait L'EDI 1 (figure 1). Il était donc difficile d'obtenir le soutien d'une communauté enthousiaste. De plus, la prise en charge de Java comme application de bureau sur les OS et dans les boutiques d'applis était de plus en plus difficile : résoudre les problèmes de compatibilité prenait trop de temps.

Alessandro évoque la 1^{ère} étape : « Refonte de toute la chaîne d'outils, avec création d'une interface en ligne de commande (CLI) *Arduino CLI* ». « Écrite en *Golang* et *TypeScript*, elle présente toutes les fonctionnalités de l'ancien EDI ainsi que toutes les nouvelles ». [1] L'approche modulaire permet à la fois son association à l'EDI préféré du développeur et la création du nouvel EDI d'Arduino. En outre, les développeurs professionnels plébiscitent la CLI, qui leur permet d'intégrer facilement l'environnement Arduino à d'autres composants très utilisés dans le développement de logiciels, par ex. des outils d'intégration continue/déploiement continu (CI/CD).

Rafraîchir l'EDI

Vu l'avancement de la CLI Arduino, il devenait temps de s'attaquer à l'amélioration de l'interface visuelle. Si les débutants se contentaient de l'éditeur d'origine, les plus aguerris déploraient l'absence des extras livrés avec ceux des éditeurs d'outils de développement et des grands fondeurs. À mesure que le code s'étoffe, s'y retrouver peut devenir difficile : beaucoup souhaitaient une bascule de visualisation des commentaires.

La plupart des EDI bénéficient de l'autoindentation qui préserve la lisibilité du code, et de l'autofermeture des parenthèses qui évite bien des erreurs de saisie. Mais surtout, le débogueur faisait l'unanimité au sein de la communauté. Pour un développeur expérimenté de logiciels embarqués, pouvoir explorer le contenu des variables et suivre l'exécution du code est une capacité standard d'un EDI. Pour beaucoup, le cycle édition-compilation-téléchargement de l'EDI 1 pour déboguer chaque croquis était lent et frustrant.

« Le choix de *Theia* et *Electron* comme pile logicielle pour l'EDI est idéal : ce sont des projets open-source fournissant une prise en charge (cloud et bureau) souple et multilingue de l'EDI » poursuit Alessandro. Cela garantit la prise en charge de plus de 60 traductions linguistiques. La 1^{ère} version bêta de l'EDI 2.0 fut partagée avec la communauté en 2021. [2]

Capacité de débogage

Certaines cartes Arduino, par ex. équipées de microcontrôleurs Arm Cortex-M (MCU), ont déjà le matériel permettant la prise en charge du débogage. Souvent appelé *puce de débogage intégré* (EDBG), ce matériel communique avec l'interface de débogage du MCU cible et fournit aussi l'interface série COM virtuelle via USB. D'autres ont un connecteur pour les débogueurs tiers, tel que J-Link. [3]. Les deux sont généralement pris en charge par un autre projet de logiciel libre, *GDB* (*GNU Project Debugger*). [4]

« Le débogage était une exigence claire de la communauté des développeurs pro. », rappelle Alessandro. Notez aussi la cohérence avec le lancement d'*Arduino Pro* [5], une nouvelle gamme de matériel s'appuyant sur une connectivité sécurisée qui offre une plateforme IdO tout-en-un. Cependant, tout ne change pas. Les développeurs débutent toujours par le classique croquis *.ino*, mais à l'inverse des autres EDI, ces fichiers sont reconnus comme code exécutable.

Aperçu de l'EDI 2.0

Le nouvel EDI Arduino [6] conserve la charte couleur caractéristique et la disposition familière (figure 2). Cependant, un nouveau bouton « **Start Debugging** » apparaît en haut ainsi qu'un panneau sur le

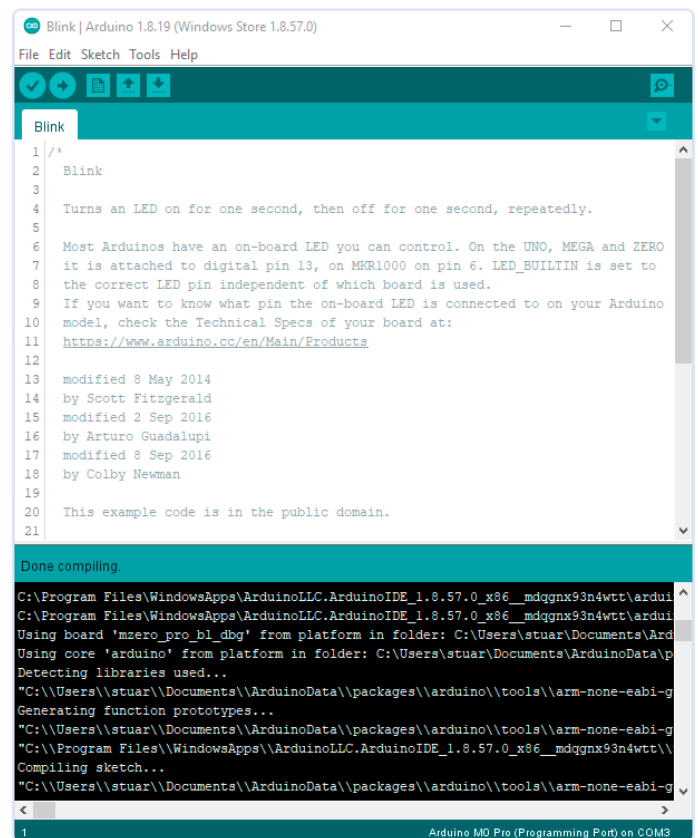


Figure 1. Le célèbre EDI Arduino 1.0, basé sur Java.

côté gauche. C'est là que le développeur accède aux registres du MCU et au déroulement de son code. Par ailleurs, presque identique, le menu comprend des ajouts dans certaines listes déroulantes. À gauche, un bandeau vertical d'icônes donne accès à divers composants de l'EDI, par ex, les gestionnaires de cartes et bibliothèques : ces utilitaires sont intégrés dans une fenêtre placée à gauche de l'éditeur de texte (**figure 3** et **figure 4**). Malgré ces changements

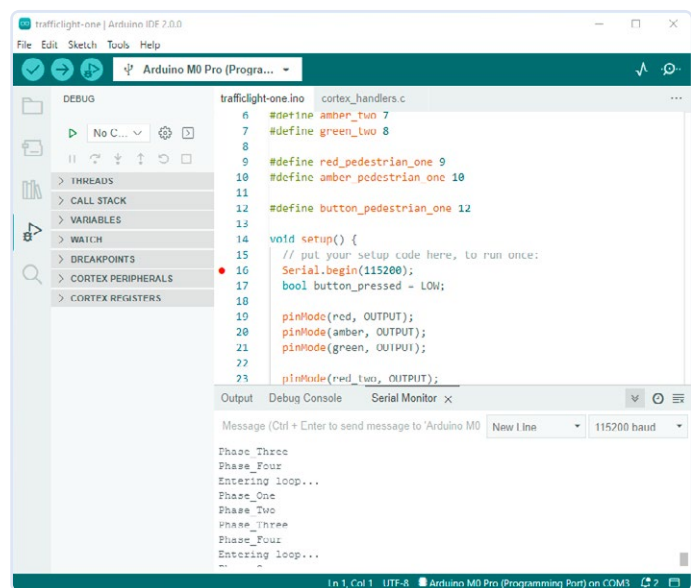


Figure 2. Familier, mais différent. L'EDI Arduino 2.0 garde la clarté de l'opus précédant et offre une foule de nouvelles fonctionnalités aux développeurs pro. et aux amateurs éclairés.

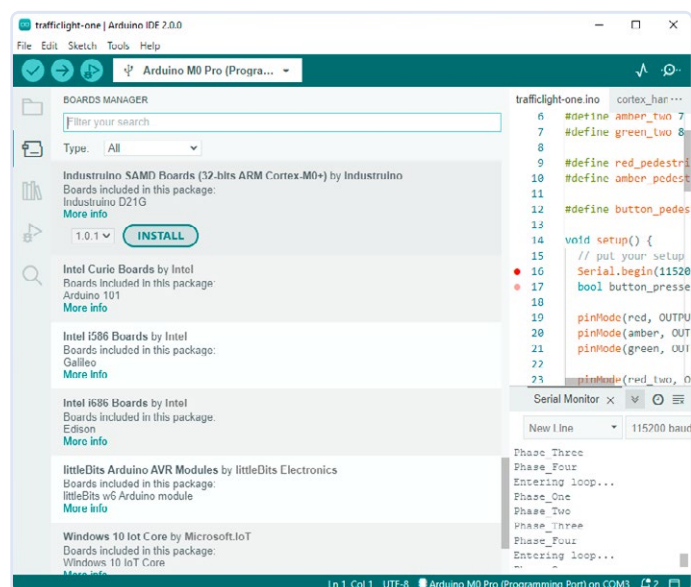


Figure 3. Le gestionnaire de cartes est familier, mais il a déménagé.

visuels, les utilisateurs seront ravis de noter que la fonctionnalité est globalement identique. Cette fenêtre montre aussi l'icône de débogage et les vues y afférentes. Le moniteur série s'ouvre désormais sous forme d'onglet dans la fenêtre principale de l'EDI, à côté des messages de compilation et téléchargement, et non dans des fenêtres séparées. Pour voir les données, la fonction *traceur série* est toujours là et ouvre sa propre fenêtre (**figure 5**).

L'éditeur bénéficie de quelques améliorations. « Parmi les principales, citons la complétion de code et l'environnement propose de possibles noms de variables en cours de frappe », confirme Alessandro. « Autre ajout : le développeur peut désormais naviguer dans le code et sauter rapidement à la ligne de définition d'une fonction ».

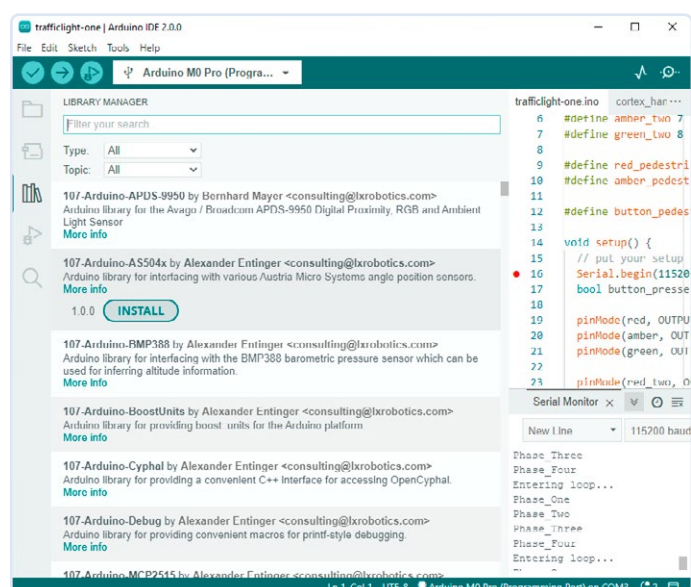


Figure 4. Le gestionnaire de bibliothèque fonctionne comme avant, mais est ancré à côté de l'éditeur.



Figure 5. Une fenêtre traceur série mise à jour dessine les graphes de données.

La fonction de m. à j. du logiciel des cartes wifi et des certificats SSL racine aide les développeurs d'applications IdO.

De nos jours, l'un des plus grands défis des outils de développement de logiciels embarqués est la gestion des versions de bibliothèques. Construire un même projet sur divers ordinateurs en est un autre : c'est acrobatique car les chemins de fichiers soit n'existent pas encore, soit changent d'un système d'exploitation à l'autre. Arduino résout ce problème grâce à un métafichier stockant la version de toutes les bibliothèques et paquets de cartes utilisés. Les développeurs peuvent changer de poste de travail, partager leur projet et même le rendre disponible sur GitHub tout en gardant la capacité de le compiler. La durée de compilation n'est pas encore optimisée : le code source est toujours recompilé en totalité, même si une seule ligne de code a été modifiée. Alessandro me dit que « Arduino CLI peut se contenter de recompiler les seuls fichiers modifiés », on peut donc espérer que l'EDI en bénéficiera aussi un jour.

Évolution probable de l'EDI Arduino

Après les versions destinées au test fonctionnel par la communauté, le 1^{er} « build » officiel d'Arduino IDE 2.0 fut annoncé en septembre 2022. Comme l'EDI 1.0, c'est un projet à durée indéfinie : il va évoluer et se développer au gré des observations des utilisateurs.

Alessandro note aussi que « Grâce à l'approche modulaire, il sera plus facile pour la communauté de contribuer à développer l'EDI ». C'est ainsi que le développement fonctionne désormais. Arduino ne compte que six développeurs internes à plein temps sur le projet. Des centaines de contributeurs les soutiennent avec compétence, des dizaines ont participé à l'élaboration de nouvelles fonctionnalités et aux tests. Le projet étant hébergé sur GitHub, tout le monde peut contribuer, signaler des problèmes ou suggérer des améliorations.

En tant que vétéran de l'open-source, j'interroge Alessandro sur le plus grand défi de la gestion d'un projet auquel tant de monde contribue. « Garder tout le monde concentré sur la mission d'Arduino », répond-il. « En tant que facilitateur technique, il faut nous rappeler que l'EDI Arduino va au-delà des cartes Arduino – toute une industrie devient possible. »

220520-04 – Yves Georges

À propos de l'auteur

Stuart Cording est ingénieur et journaliste. Il a plus de 25 ans d'expérience dans l'industrie électronique. Un grand nombre de ses articles Elektor récents sont ici : www.elektormagazine.com/cording. Pour Elektor, il anime aussi le mensuel de diffusion en direct, Elektor Engineering Insights (www.elektormagazine.com/eei), et enseigne à Elektor Academy (www.elektormagazine.com/elektor-academy).

Des questions, des commentaires ?

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

À propos d'Alessandro Ranellucci

(responsable Maker Business, Open Source & Community)

Alessandro Ranellucci, responsable Maker Business, Open Source & Community, rejoint Arduino en 2020 pour diriger les produits et la stratégie pour les créateurs, ainsi que l'écosystème open source. Auparavant, il a travaillé comme responsable de l'open source et a été membre du groupe de travail « transformation numérique » pour le gouvernement italien. Il est le conservateur de Maker Faire Rome et l'auteur principal de Slic3r, un outil d'impression 3D open-source répandu.

LIENS

[1] Le projet Go : <https://go.dev/>

[2] Arduino, « Annonce de l'édition EDI Arduino 2.0 (beta) », 1er mars 2021 : <https://blog.arduino.cc/2021/03/01/announcing-the-arduino-ide-2-0-beta/>

[3] Sondes de débogage Segger, J-Link : www.segger.com/products/debug-probes/j-link/

[4] Free Software Foundation, « GDB : Le projet débogueur GNU » : www.sourceware.org/gdb/

[5] Arduino Pro : www.arduino.cc/pro/

[6] Téléchargements Arduino : www.arduino.cc/en/software