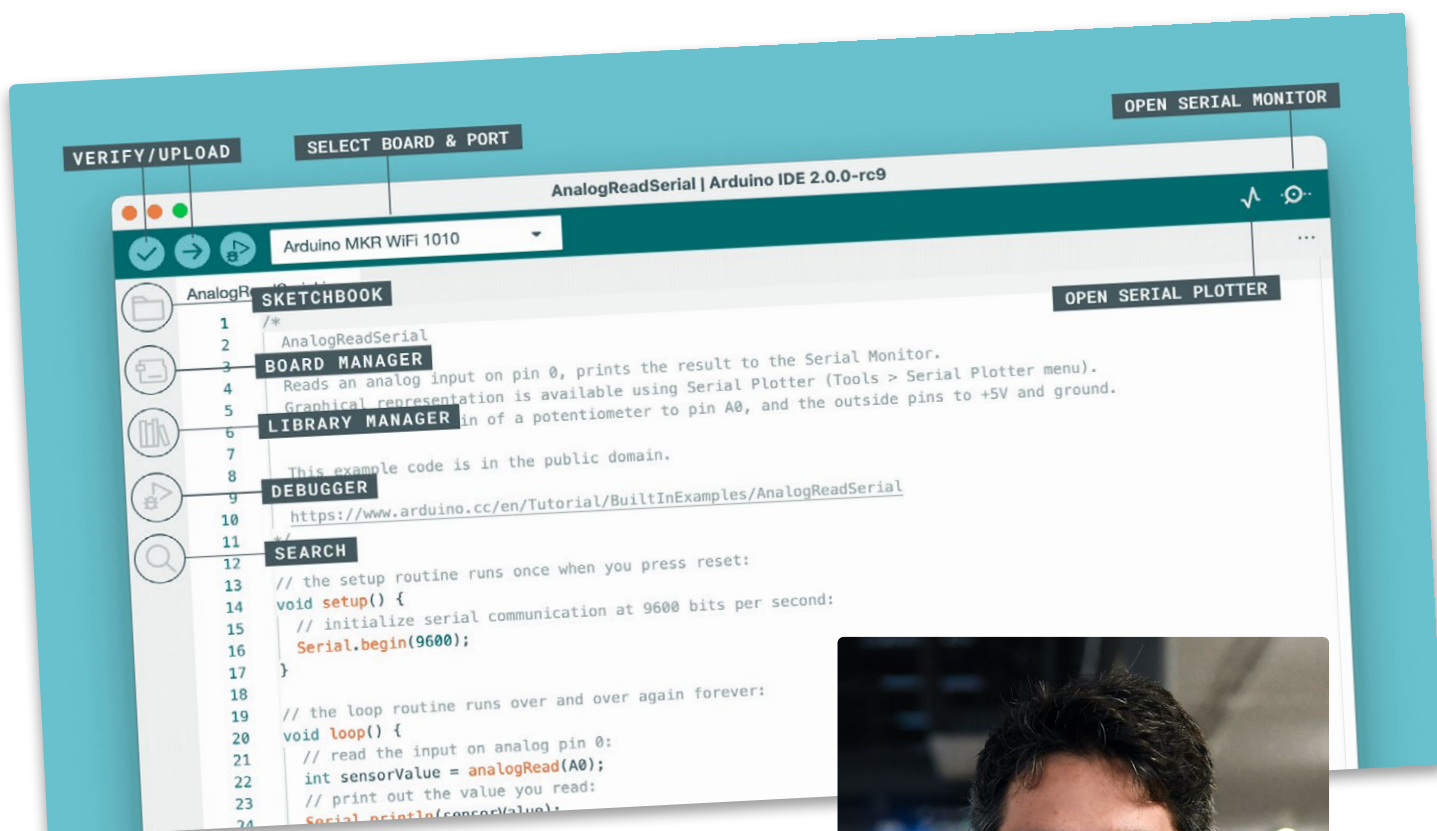




Arduino-sketches schrijven is nog beter geworden



Alessandro Ranellucci

Stuart Cording (Elektor)

Arduino IDE 2.0 breekt met het oude zonder met de Arduino-ervaring te breken. Benieuwd naar de details? Arduino's Alessandro Ranellucci biedt wat inzichten.

Voor de meeste Arduino-gebruikers was hun eerste interactie met een programmeertaal de geïntegreerde Arduino-ontwikkelomgeving (IDE). Met zijn groene randen en codebewerkingsvenster hebben nieuwelingen uren besteed aan het ontcijferen van de uitvoer terwijl ze hun sketches maakten en laadden. Sinds de lancering in 2005 is de IDE gegroeid in mogelijkheden, ondersteunt hij meer boards, biedt een betere toegang tot gedeelde bibliotheken en voegt een plotter toe om het maken van grafieken van gegevens te ondersteunen. Echter, naarmate het aantal professionele ontwikkelaars is gegroeid en de makers hun vaardigheden hebben verbeterd, is een opvallend gebrek de debug-ondersteuning. Om meer te weten te komen, sprak ik kortgeleden met Alessandro Ranellucci, hoofd Maker Business, Open Source & Community bij Arduino.

Compleet herschreven

In 2018 zette het Arduino-team een belangrijke stap die niet alleen de mogelijkheden van de IDE zou veranderen, maar de hele codebase zou vernieuwen. IDE 1 was geschreven in Java als een monolithisch blok code (figuur 1). Dit maakte het een uitdaging om steun te krijgen van enthousiaste aanhangers uit de community. Bovendien werd het steeds moeilijker om Java als desktopapplicatie op besturingssystemen en appstores te ondersteunen, waardoor veel tijd werd besteed aan het oplossen van compatibiliteitsproblemen. “We zijn begonnen met het herstructureren van de hele toolchain, waarbij we een command-line interface (CLI) met de naam Arduino CLI hebben gemaakt,” legt Alessandro uit. “Geschreven in Golang en TypeScript, bezit het alle functionaliteit van de oude IDE, evenals alle nieuwe functies.” [1]

Dankzij de modulaire aanpak is het gemakkelijker te koppelen aan de voorkeurs-IDE van de ontwikkelaar, wat ook de nieuwe IDE van Arduino mogelijk maakte. Bovendien is de CLI iets wat professionele ontwikkelaars sterk toejuichen, omdat het hen in staat stelt de Arduino-omgeving gemakkelijk samen te voegen met andere populaire componenten die bij softwareontwikkeling worden gebruikt, zoals tools voor continue integratie/continue implementatie (CI/CD).

De IDE vernieuwen

Nu er vooruitgang was geboekt met Arduino CLI, was het tijd om te onderzoeken hoe de visuele interface het best kon worden geïmplementeerd. Terwijl beginners tevreden waren met de oorspronkelijke editor, misten degenen met meer ervaring enkele van de kleine extra's die standaard in de IDE's van de grote leveranciers van halfgeleiders en ontwikkelingstools zitten. Naarmate de code groeit, kan de navigatie moeilijk worden, dus de mogelijkheid om commentaar om te schakelen stond hoog op het verlanglijstje. De meeste IDE's implementeerden ook auto-inspringen, waardoor code leesbaar blijft, en auto-haakje-sluiten, waardoor veelgemaakte fouten bij het invoeren van code worden vermeden. Maar bovenal was ondersteuning voor debuggen een duidelijke wens van de community. Voor ervaren embedded software-ontwikkelaars wordt de mogelijkheid om de inhoud van variabelen te bekijken en de uitvoering van code te volgen beschouwd als een standaard IDE-mogelijkheid. Ter vergelijking: het trage edit/build/download-proces van IDE 1 om een enkele bewerking te debuggen was voor velen frustrerend.

“We kozen Theia met Electron als softwarestack voor de IDE,” zegt Alessandro, “omdat het open-source projecten zijn die ons in staat stellen flexibele, meertalige cloud en desktop IDE-ondersteuning te bieden.” Dit zou ervoor zorgen dat de meer dan zestig talen worden ondersteund. In 2021 werd de eerste bèta van IDE 2.0 gedeeld met de community. [2]

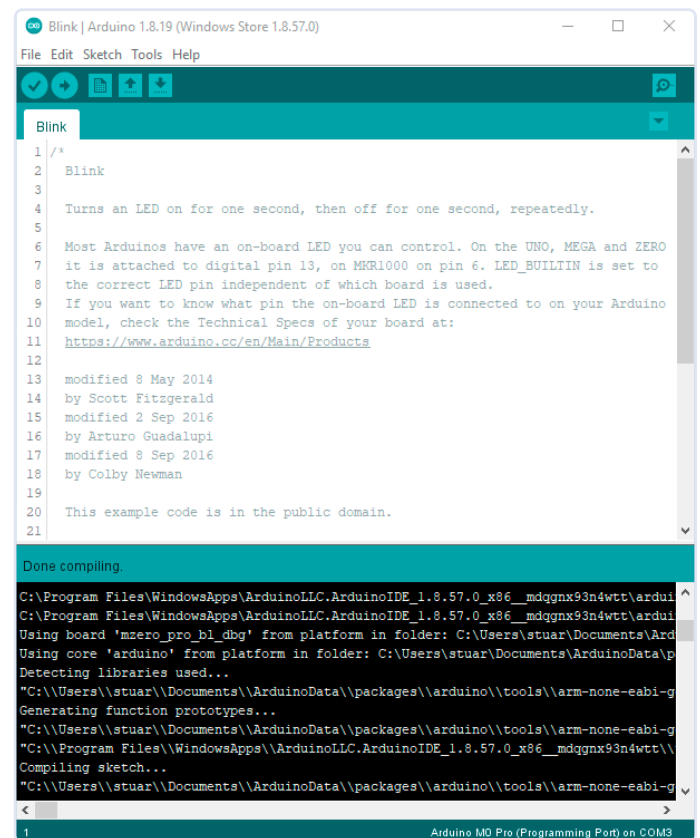
De mogelijkheid om te debuggen

Sommige Arduino-boards, zoals die met Arm Cortex-M-gebaseerde microcontrollers (MCU), hebben al de hardware die nodig is om het debuggen te ondersteunen. Vaak aangeduid als een Embedded Debug-chip (EDBG) communiceert deze met de debug-interface

Arduino IDE 2.0 behoudt de duidelijkheid van zijn voorganger, maar biedt een groot aantal nieuwe functies die geschikt zijn voor professionele ontwikkelaars en ervaren makers.

van de target-MCU, maar levert ook de virtuele seriële COM-interface via USB. Andere hebben een header voor debuggers van derden, zoals J-Link [3]. Beide worden doorgaans ondersteund door een ander open-source softwareproject, GDB (GNU Project Debugger) [4].

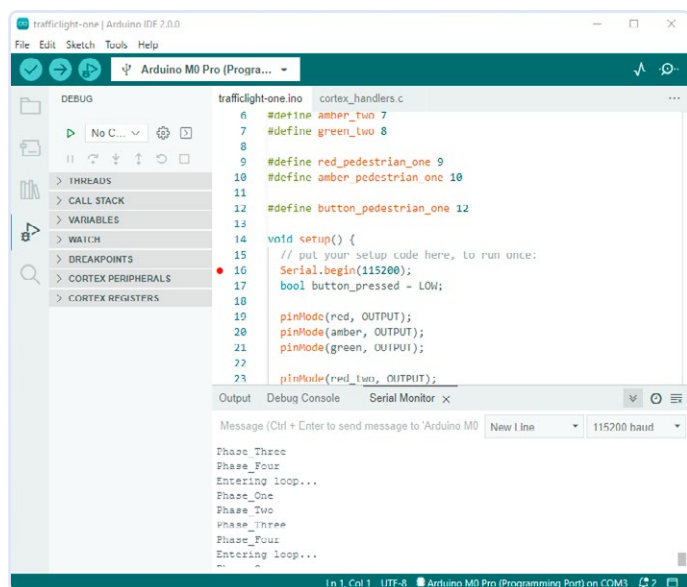
“Debuggen was een duidelijke vereiste, vooral voor de professionele ontwikkelaarsgemeenschap,” herinnert Alessandro zich. Dit sluit ook aan bij de lancering van de Arduino Pro [5], een nieuwe reeks hardware die wordt ondersteund door beveiligde connectiviteit om een alles-in-één IoT-platform te bieden. Veel blijft echter onveranderd. Ontwikkelaars beginnen nog steeds met de traditionele ino-sketch, hoewel dergelijke bestanden, in tegenstelling tot andere IDE's, worden herkend als uitvoerbare code.



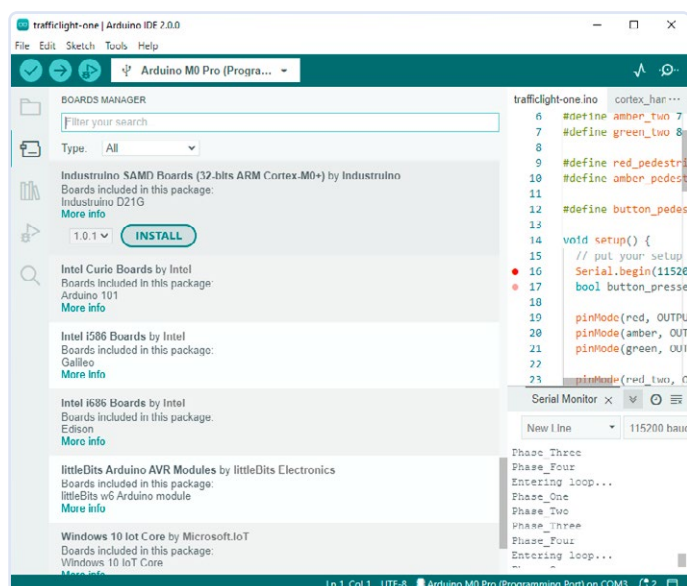
Figuur 1. De bekende Arduino IDE 1.0, gebaseerd op Java.

Een rondleiding door IDE 2.0

De nieuwe Arduino IDE [6] behoudt het direct herkenbare kleurenschema en de vertrouwde lay-out (**figuur 2**). Bij nadere beschouwing is er echter een nieuwe knop “Start Debugging” bovenaan en een paneel aan de linkerkant verschenen. Hier krijgen ontwikkelaars toegang tot het inwendige van de MCU en een overzicht van wat er in hun code gebeurt. Verder is het menu min of meer hetzelfde, maar met een paar extra's in sommige dropdown menu's.

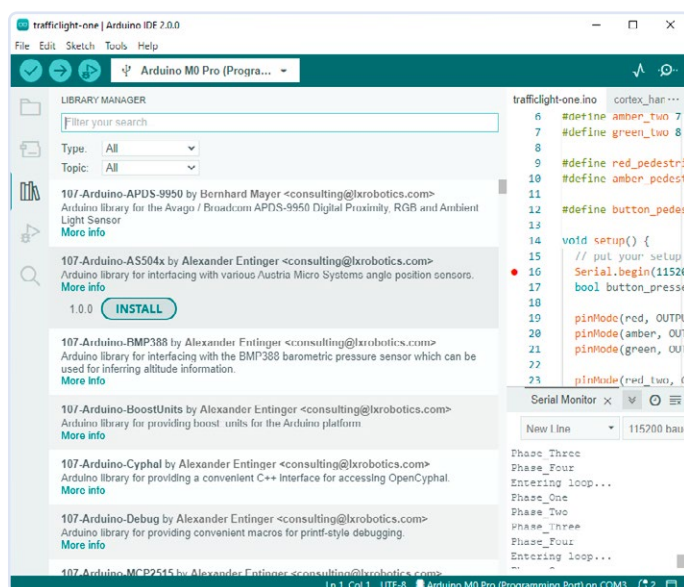


Figuur 2. Vertrouwd, maar anders. Arduino IDE 2.0 behoudt de duidelijkheid van zijn voorganger, maar biedt een groot aantal nieuwe mogelijkheden die geschikt zijn voor professionele ontwikkelaars en ervaren makers.



Figuur 3. De boardmanager is bekend, maar zijn locatie is verplaatst.

Een smal paneel biedt toegang tot enkele IDE-componenten met behulp van pictogrammen aan de linkerkant. De board- en de bibliotheekmanager zijn twee voorbeelden, waarbij deze hulpprogramma's zijn ingebed in een venster links van de teksteditor (**figuur 3** en **figuur 4**). Ondanks de visuele veranderingen zullen gebruikers blij zijn te weten dat de algemene functionaliteit in wezen dezelfde is. In dit venster kan ook de debugfunctie worden geselecteerd en bekeken. De seriële monitor opent nu binnen het hoofdvenster van de IDE als een uitvoer met tabbladen, naast waar de compilatie- en downloadberichten verschijnen, in plaats van in afzonderlijke vensters. Voor de visualisatie van gegevens is de seriële plotterfunctie nog steeds beschikbaar in een eigen venster (**figuur 5**). Ook bij de editor is er het een en ander gewijzigd. “Enkele belang-



Figuur 4. De bibliotheekmanager behoudt dezelfde functionaliteit, maar wordt nu naast de editor gedockt.



Figuur 5. Data-visualisatie wordt ondersteund door een geactualiseerd Serial Plotter-venster.

rijke verbeteringen zijn code-aanvulling, waarbij de omgeving tijdens het typen mogelijke namen van variabelen aanbiedt”, legt Alessandro uit. “We hebben ook codenavigatie toegevoegd, waardoor ontwikkelaars snel naar de coderegel kunnen springen waar een functie is gedefinieerd.” Om degenen te helpen die IoT-toepassingen ontwikkelen, is er ook een firmware-updater voor WiFi-boards en SSL root-certificaten.

Een van de grootste uitdagingen van de hedendaagse embedded software-ontwikkeltools is het op de hoogte blijven van de gebruikte bibliotheekversies. De andere is het kunnen bouwen van hetzelfde project op een andere computer, iets wat typisch mislukt omdat bestandspaden ofwel niet bestaan of worden verminkt bij het omschakelen tussen besturingssystemen. Arduino heeft dit opgelost door een metafile toe te voegen die de versie van alle gebruikte bibliotheken en board packages opslaat. Ontwikkelaars kunnen hun werkstation verplaatsen, hun project delen en zelfs projecten beschikbaar stellen op GitHub en het nog steeds compileren. Wat nog niet verbeterd is, is de bouwtijd. Alle broncode wordt nog steeds gecompileerd, ook al is er misschien maar één regel code veranderd. Alessandro vertelt me dat “de Arduino CLI het hercompileren van alleen de gewijzigde bestanden kan ondersteunen,” dus hopelijk zal de IDE daar op een gegeven moment ook van profiteren.

Hoe verder met de Arduino IDE?

De eerste officiële release van Arduino IDE 2.0 werd aangekondigd in september 2022 na verschillende release-kandidaten waarmee de community de functies kon testen. Echter, net als IDE 1.0 zal het project nooit eindigen – het blijft gewoon veranderen en verder evolueren door feedback van gebruikers.

“Dankzij de modulaire aanpak wordt het nu gemakkelijker voor de community om bij te dragen aan de ontwikkeling van de IDE,” laat Alessandro weten.

In feite is dit hoe de ontwikkeling vandaag in zijn werk gaat. Arduino heeft maar zes fulltime interne ontwikkelaars die aan het project werken. Honderden medewerkers hebben hen vakkundig ondersteund, waarvan tientallen actief zijn geweest bij het vormgeven van nieuwe functies en testen. Omdat het project wordt gehost op GitHub, kan iedereen bijdragen aan de codeontwikkeling, het melden van problemen of het voorstellen van verbeteringen.

Als open-source veteraan vroeg ik Alessandro wat de grootste uitdaging is bij het runnen van een project waar iedereen aan kan bijdragen. “Iedereen gefocust houden op de missie van Arduino,” antwoordt hij. “Als technologie-enabler moeten we onthouden dat Arduino IDE verder gaat dan de Arduino-boards – we maken een hele industrie mogelijk.” ◀

220520 (vertaling: Willem den Hollander)

Over de auteur

Stuart Cording is ingenieur en journalist met meer dan 25 jaar ervaring in de elektronica-industrie. Je kunt veel van zijn recente Elektor-artikelen vinden op www.elektormagazine.com/cording. Naast het schrijven voor Elektor verzorgt hij de maandelijkse livestream Elektor Engineering Insights (www.elektormagazine.com/eei) en geeft hij Elektor Academy-cursussen (www.elektor-magazine.com/elektor-academy).

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via stuart.cording@elektor.com of naar de redactie van Elektor via redactie@elektor.com.

Alessandro Ranellucci

(hoofd Maker Business, Open Source & Community)

Alessandro Ranellucci, hoofd van de afdeling Maker Business, Open Source & Community, kwam in 2020 bij Arduino om de producten en strategie voor makers en het open source ecosysteem te leiden. Hij werkte eerder als hoofd open source en was lid van de taskforce digitale transformatie voor de Italiaanse overheid. Hij is de curator van Maker Faire Rome en hoofdauteur van Slic3r, een populaire open-source tool voor 3D-printers.

WEBLINKS

- [1] The Go Project: <https://go.dev/>
- [2] Arduino, “Announcing the Arduino IDE 2.0 (beta),” 1 maart 2021: <https://blog.arduino.cc/2021/03/01/announcing-the-arduino-ide-2-0-beta/>
- [3] Segger, J-Link Debug Probes: www.segger.com/products/debug-probes/j-link/
- [4] Free Software Foundation, “GDB: The GNU Project Debugger”: www.sourceware.org/gdb/
- [5] Arduino Pro: www.arduino.cc/pro/
- [6] Arduino-downloads: www.arduino.cc/en/software