DOI: 10.2195/lj_edrev_sirinterlikci_en_202611_01

A Historical Overview of Arduino Development and Its Impact – Makings of an Italian Masterpiece

Arif Sirinterlikci [1] ***& Jérôme Rutinowski*** [1, 2]

[1] *Robert Morris University, USA*
[2] *TU Dortmund University, Germany*

This paper presents a historical overview of the Arduino platform, tracing its origins in the Wiring project at the Interaction Design Institute Ivrea in the early 2000s and its development into a widely adopted open-source electronics ecosystem being used in a variety of disciplines such as logistics, robotics, and teaching. It examines the evolution of Arduino hardware and software, highlighting key technological milestones, the growth of its global community, and its role in democratizing access to electronics and programming, for tinkerers and students alike. Particular attention is given to its applications in education, robotics, rapid prototyping, and logistics, including automated guided vehicles and warehouse systems. Finally, the paper discusses current developments and future perspectives, focusing on industrial applications, Internet of Things (IoT), and the integration of artificial intelligence and edge computing.

[Keywords: Arduino, Maker Movement, Robotics, Logistics, Automation]

1 Introduction

The Arduino platform represents a significant achievement in the landscape of open-source electronics and microcontrollers by breaking barriers and providing an accessible ecosystem for individuals to engage with data acquisition and control technologies interacting with the physical world [1]. Its emergence has facilitated widening access to electronics and programming, generating a powerful global community of creators including educators, students, and professionals [2]. In logistics research and practice, Arduino serves as a control and prototyping platform for various logistics systems, such as automated guided vehicles and inventory systems [3], [4]. Arduino is frequently chosen due to its low cost, ease of programming and relatively low power consumption [5], [6], which is especially attractive in teaching and low-margin industries. This article aims to provide a comprehensive historical overview of the platform's development and use as well as its origins and the evolution of its hardware and software, exploring the growth of its community and its impact. This article also presents its diverse range of applications, including logistics and teaching, while analyzing its influence on the Maker Movement, summarizing its current state and possible future directions.

2 Origins

In the early 2000s, interactive design and educational robotics were shaped by a set of hardware platforms that were powerful but often difficult to access for non-engineers. Systems such as the MIT Handy Board, a self-contained microcontroller-based robotics platform developed for teaching mobile robotics, were widely used in university robotics education, including courses at Massachusetts Institute of Technology, and distributed commercially by Gleason Research [7]. Similarly, the Parallax BASIC Stamp was a common alternative but remained relatively expensive (often around $100 for entry-level setups), limiting accessibility for students, artists, and designers [8], [9]. As a result, existing tools were frequently perceived as technically restrictive and poorly aligned with creative practice [8].

Within this context, Hernando Barragán, a Colombian student at the Interaction Design Institute Ivrea (IDII), developed the foundational work for a new approach to microcontroller programming. In 2003, as part of his master's thesis titled Wiring: Prototyping Physical Interaction Design, he introduced Wiring [10]. Wiring consisted of a simplified programming environment based on the Processing IDE, a microcontroller board using an ATmega128, and a supporting library layer designed to reduce complexity in hardware programming [10]. The project demonstrated the feasibility of a unified, accessible platform for physical computing and established a conceptual foundation for later systems aimed at non-engineers.

Article is protected by German copyright law

Building on these ideas, Massimo Banzi and colleagues at IDII recognized the potential of extending Wiring into a more broadly usable platform. Collaborating with David Cuartielles, David Mellis, and others, they began developing what would become Arduino [8], [11]. The system was named Arduino in 2005, referencing a bar in Ivrea (*Bar di Re Arduino*) where members of the team met. The core team also included Tom Igoe and Gianluca Martino, who contributed to hardware development, production, and early distribution efforts [8].

Arduino's design directly addressed the barriers identified in earlier systems by prioritizing simplicity, affordability, and openness. The platform used low-cost 8-bit AVR microcontrollers from Atmel, implemented a self-contained board design with accessible input/output interfaces, and introduced a bootloader enabling straightforward USB-based program uploads. On the software side, the integrated development environment adopted the abstraction of "sketches," further lowering the barrier to entry for users without formal programming backgrounds [8], [9]. The target price point of approximately $30 reflected the overarching goal of enabling widespread access to physical computing tools [8].

3 Public Release

The Arduino project quickly moved from concept to prototype and then became reality [12]. Following initial development and refinement, lighter and less expensive versions of Arduino boards were distributed within the open-source community. One of the first commercially available boards was the Arduino Serial Board (2005–2006), which used an ATmega8 microcontroller and relied on a serial port for communication prior to the adoption of USB connectivity [8], [12]. This design made the platform accessible beyond academic environments.

A pivotal moment occurred in 2007 with the release of the Arduino Diecimila [8]. This board is often cited as the first widely distributed Arduino version and marked a significant step in standardization and scalability. The name "Diecimila" (Italian for "ten thousand") referenced the milestone of approximately 10,000 Arduino boards produced and circulated within the community [8], [13].

4 Hardware Evolution

The journey of Arduino hardware has been shaped by continuous iteration and a commitment to addressing growing user needs [12]. The Arduino Diecimila (2007) introduced key improvements such as automatic reset via serial communication and a low-dropout voltage regulator, improving usability and power efficiency [8]. The board was based on the ATmega168 and offered expanded memory compared to earlier designs [8]. The Arduino Duemilanove (2008) further simplified the platform by enabling automatic power source selection, removing the need for a manual jumper. It initially used the ATmega168 and later the ATmega328P [8], [14]. A major milestone followed with the Arduino Uno (2010), which standardized the modern Arduino form factor. It combined the ATmega328P microcontroller with an ATmega16U2 for USB communication, replacing earlier FTDI-based designs [14]. Beyond these foundational boards, the Arduino family diversified to address a wider spectrum of requirements [12]. The LilyPad (2007) was developed for wearable electronics in collaboration with SparkFun [15]. The Arduino Mega (2008) targeted applications requiring more I/O pins and expanded memory for complex projects [15]. The Arduino Nano and Mini (2008) provided the same core functionality in smaller form factors, making them suitable for compact embedded systems [14].

In 2012, the Arduino Due introduced a 32-bit ARM Cortex-M3 processor, significantly increasing computational performance and marking a shift toward higher-end embedded applications. The Arduino Leonardo (2012) integrated native USB functionality, enabling direct USB device emulation [14]. From the late 2010s onward, the MKR family expanded Arduino into IoT-oriented compact boards with improved connectivity options [12]. More recently, Arduino has integrated ESP32-based boards such as the Nano ESP32, introducing built-in Wi-Fi and Bluetooth capabilities and leveraging Espressif's ESP32 architecture. The Arduino Pro line, introduced in the 2020s, targets industrial applications with more robust ARM-based designs [12].

The Arduino Uno R4 (2023) transitioned to Renesas RA4M1 ARM Cortex-M4 microcontrollers, significantly improving performance compared to earlier generations [13]. Overall, the diversification of Arduino hardware reflects its adaptation to a broad range of user needs, from education to industrial applications. New boards and communication technologies have supported Arduino's sustained growth, with integrated wireless standards such as Wi-Fi and Bluetooth enabling IoT applications alongside shields for motor control, GPS, and Ethernet connectivity [12].

5 Software Development

The software of the platform has evolved in parallel with its hardware, playing a key role in its accessibility and widespread adoption. The Arduino IDE was initially developed by David Mellis and the Arduino core team, building directly on the Wiring IDE, which itself was inspired by Processing [10], [12]. The Processing environment, created by Casey Reas and Ben Fry, was designed as a cross-platform programming environment for interactive graphics and multimedia applications and strongly influenced the design philosophy of simplified creative coding tools [8]. This

Article is protected by German copyright law

lineage reflects a broader effort to make programming accessible to non-technical users in design and artistic contexts.

The first version of the Arduino IDE was released in the mid-2000s by the core team including Massimo Banzi, David Cuartielles, David Mellis, Gianluca Martino, and Tom Igoe [12]. The Arduino IDE is an open-source application that allows users to write, compile, and upload sketches to Arduino boards [2], [13]. It supports multiple operating systems, including Linux, Windows, macOS, and Raspberry Pi environments [2]. The interface includes a text editor, message console, serial monitor, toolbar, and menus for development tasks [13]. Initially, sketches used the ".pde" extension, later replaced by ".ino" in Arduino 1.0 [13].

Most Arduino boards include a bootloader that simplifies uploading code via serial communication, typically over USB [14]. The IDE also provides a serial monitor for real-time communication and debugging between the computer and the board [14]. The programming language used in the Arduino IDE is a simplified framework based on C and C++, often referred to as Embedded C++ in practice [13], [14]. It is designed to lower the barrier to entry for users without extensive programming experience. The IDE handles underlying complexity, including automatic generation of function prototypes and inclusion of core headers such as Arduino.h [13]. A sketch is structured around setup(), which runs once, and loop(), which executes repeatedly, replacing the traditional main() function in standard C/C++ programs [13].



Figure 1: An Arduino and AI-enabled Robert Morris as a robotic greeter.

As the ecosystem matured, the software stack expanded beyond the traditional IDE. The Arduino CLI, introduced in the late 2010s, enables command-line interaction and automation workflows and serves as the foundation for modern Arduino tooling [13]. It underpins official development tools, including the desktop IDE and Web Editor. The Arduino Web Editor, introduced in 2019, allows browser-based development and remote code uploads from any internet-connected device [2], [13]. In 2022, Arduino released IDE 2.0 with a modern interface, code autocompletion, improved navigation, and an integrated debugger, significantly enhancing usability for advanced development workflows [2], [13]. Recent trends in Arduino technology development include a growing focus on edge computing, integration of TensorFlow Lite for Machine Learning, and increasing emphasis on energy-efficient and sustainable designs (see Fig. 1).

6 Community, Adoption, and Ecosystem Growth

Starting from its early development, the Arduino project fostered a strong sense of community, which contributed significantly to its growth. Following the initial distribution of lighter and cheaper Arduino boards, a global ecosystem of users began to emerge [12]. By the early 2010s, cumulative production of Arduino boards had reached several hundred thousand units, reflecting rapid adoption across educational and maker communities [12], [13]. The open-source nature of Arduino has been a key driver in cultivating a distributed network of developers, educators, artists, and hobbyists contributing to its ongoing evolution [13].

The Arduino Forum has served as a central online hub for discussion, support, and project sharing [13]. Beyond the official forum, platforms such as the Arduino Project Hub (formerly create.arduino.cc) and Hackster.io provide spaces for showcasing projects and tutorials [13]. The Arduino Project Hub was redesigned in 2022 to improve usability and maintainability [13]. A notable indicator of community activity is the growth of third-party libraries in the Arduino Library Manager, which reached several thousand entries by the mid-2020s, reflecting continuous ecosystem expansion [13].

The Arduino community is highly diverse, including students, hobbyists, artists, programmers, and professionals working in industrial applications [2]. This user base contributes a large body of accessible knowledge through forums, tutorials, and shared projects. Collaboration is a defining feature of the ecosystem, with users actively supporting one another and sharing implementations and improvements [13]. The Arduino platform has also been widely adopted in educational contexts, serving as a tool for teaching programming, electronics, logistics and STEM principles from primary education through university levels [2]. Dedicated education kits and structured learning materials

Article is protected by German copyright law

further support classroom integration [13]. Research in educational technology has shown that Arduino-based learning environments can improve student engagement and confidence in programming and electronics [1].



Figure 2: An Arduino based smart planter controlled via a cloud-accessible application.

Arduino is additionally used for rapid prototyping by students, educators, hobbyists, and professionals, enabling fast iteration during early-stage development [13]. Its accessibility has made it a common tool for experimentation in both research and product design contexts. The platform's roots in interactive art and design remain visible, with artists using Arduino for lighting, sound, and kinetic installations [15]. While originally focused on non-industrial use, Arduino has increasingly been adopted in professional and industrial contexts. The Arduino Pro line targets these applications with more robust hardware designs and extended compatibility with industrial requirements [12], [13]. Beyond these domains, the Arduino ecosystem supports a wide range of applications, including home automation, robotics, wearable technology, and environmental monitoring systems (see Fig. 2). This diversity reflects both the flexibility of the platform and the creativity of its global community.

In modern warehousing, Arduino-based systems have been explored for both autonomous material transport and automated storage and retrieval. Nait Him et al. describe an AGV architecture in which an Arduino Mega 2560 handles autonomous navigation, while an Arduino UNO provides Bluetooth-based auxiliary control [3]. The navigation system combines infrared line following, ultrasonic obstacle detection, RFID-based location identification, and GPS positioning. Furthermore, PID control is applied to improve path-following accuracy and reduce deviations from predefined routes. The ultrasonic sensors support distance measurement and collision avoidance, while RFID tags enable the identification of warehouse locations and destinations [3]. A complementary application is presented by Fadzli and Nawawi, who developed an Arduino Mega-based automated storage and retrieval system (ASRS) [4]. Their prototype uses infrared sensors and LED indicators to monitor nine storage compartments, while a three-axis stepper-motor mechanism automates pick-and-place operations. Compartment status is transmitted via Ethernet and Modbus TCP to CODESYS and an AnyViz-based SCADA/HMI interface, allowing warehouse inventory occupancy to be visualized remotely. Although RFID is discussed as a potential extension for recording item-specific information, it is not implemented in the prototype [4]. At the mobile robotics level, Bi and Kong present an Arduino UNO R3-based forklift-type logistics AGV integrating infrared line tracking, ultrasonic and infrared obstacle detection, camera-based QR-code positioning, and Bluetooth communication with an Android client for remote control [16]. Their experimental evaluation demonstrates the functionality of the tracking and obstacle-avoidance modules, with ultrasonic distance-measurement errors reported below 1%.

One of Arduino's most significant impacts has been its role in democratizing electronics and programming. By providing an open and user-friendly platform, Arduino lowered the barrier to entry for individuals interested in embedded systems and physical computing [8], [12]. The combination of a simplified programming model, standardized hardware design, and an accessible integrated development environment enabled users without formal engineering backgrounds to develop electronic projects more easily [13], [14], [15].

The Arduino platform is closely associated with the rise of the Maker Movement and has become a central tool for DIY electronics projects across a wide range of domains [12], [13]. Its accessibility and low-cost hardware design contributed to its adoption in both educational and informal learning environments, reinforcing its role as an entry point into electronics and programming. The availability of online resources, including tutorials, project guides, and community-driven documentation, has further supported this development by enabling users to learn and build independently [13]. These resources, combined with active community participation, have been central to Arduino's sustained growth and adoption.

7 The Present and Future of Arduino

The Arduino platform continues to be actively developed, with ongoing efforts focused on both hardware and software enhancements. New hardware releases, such as the Arduino Uno R4 and the Nano ESP32, demonstrate continued integration of modern microcontroller architectures and expanded connectivity capabilities [13]. The Arduino IDE (2.x series) is under continuous development, with recent versions improving usability, debugging, and developer workflow integration [13]. Similarly, the Arduino CLI is

Article is protected by German copyright law

maintained as a core tool supporting command-line-based workflows and automation [13]. A key development trend is expanded support for multiple microcontroller architectures, including ARM-based MCUs and ESP32-based systems, enabling more capable and heterogeneous application development [12]. There is also growing integration of cloud-based tooling and edge-computing workflows, including early support for TinyML-style applications on constrained devices [2], [13].

Recent platform evolution indicates a stronger orientation toward professional and industrial applications, as reflected in the Arduino Pro line [12]. Modern Arduino boards increasingly incorporate wireless connectivity options such as Wi-Fi and Bluetooth, reflecting the importance of IoT-oriented development workflows [13]. Energy efficiency and hardware optimization are also recurring design considerations in newer generations of boards [13]. Within this broader context, Arduino continues to extend its applicability into emerging domains such as machine learning at the edge and connected embedded systems. As part of this trajectory, Arduino Nano 33 Sense Rev2 boards are planned for use in an upcoming Machine Learning course by the author for practical exercises.

The Arduino community remains a large and active ecosystem with millions of users worldwide [2]. As of 2026, 30 million Arduino IDE downloads were reached [13]. Community contributions, including libraries and tutorials, continue to grow and support platform development [13]. This active community remains a key factor in Arduino's ongoing evolution, supporting innovation and ensuring long-term relevance.

8 Conclusions

The history of Arduino is a testament to the power of open-source collaboration and the drive to make technology accessible to masses. From its humble beginnings at the IDII in 2005 in Ivrea, Italy, Arduino has grown into a leading global platform for electronics prototyping and education. As of 2024, about 1100 companies in various industry sectors were also known to use the Arduino IDE. Its evolution across hardware and software, coupled with the unwavering support of a vast active community, has enabled its adoption in a remarkably diverse range of applications, from educational tools to artistic installations and even industrial prototypes. Within logistics, Arduino-based solutions have been explored for applications such as sensor-based inventory monitoring, object tracking, automated material handling, and the prototyping of low-cost IoT systems. Arduino's impact on the Maker Movement has been profound, playing a key role in democratizing electronics and programming and empowering a new generation of creators and innovators.

Qualcomm announced on October 7, 2025, it is acquiring Arduino to integrate its edge AI technologies with Arduino's vast developer ecosystem. While Arduino will operate as an independent brand, the partnership aims to accelerate AI and high-performance computing in IoT by launching new hardware like the Snapdragon-powered "Uno Q" board. As it continues to evolve, embracing new technologies and expanding its reach into professional and industrial domains, Arduino remains a powerful force in the world of technology, with a future that promises continued innovation and impact. Especially within STEM teaching environments, Arduino holds tremendous amounts of potential and could become a key integrator for AI and robotics use cases.

REFERENCES

[1] T. Boas. "Lecture series: How open source software is revolutionizing global markets... and communities." Rapporteur's report on a lecture by Steven Weber, Carnegie Endowment for International Peace, Accessed: Nov. 2, 2026. [Online]. Available: https://carnegieendowment.org/events/2001/05/lecture-series-how-open-source-software-is-revolutionizing-global-marketsand-communities

[2] Arduino. "Getting started: Foundation ¿ introduction," Arduino, Accessed: Nov. 2, 2026. [Online]. Available: https://web.archive.org/web/20170829015201/https://www.arduino.cc/en/guide/introduction

[3] I. Nait Him, R. Lghoul, and O. Haidar, "Digital transformation of intralogistics: A framework for AGV integration in industry 4.0 warehouses," in *Energy Futures*, ser. Materials Research Proceedings, vol. 64, Millersville, PA, USA: Materials Research Forum LLC, 2026, pp. 314–321. DOI: 10.21741/9781644904091-39

[4] M. A. F. Mohd Fadzli and S. W. B. Nawawi, "Automated storage and retrieval system for warehouse," *ELEKTRIKA: Journal of Electrical Engineering*, vol. 23, no. 1, pp. 88–95, 2024. DOI: 10.11113/elektrika.v23n1.471

[5] E. Dervishi, C. Halkiopoulos, H. Antonopoulou, and L. Theodorakopoulos, "Design and implementation of an autonomous control system based on microcontroller Arduino for use in logistics," in *Proceedings of the 4th International Conference on Contemporary Marketing Issues (ICCMI 2016)*, Heraklion, Greece, Jun. 22–24, 2016, p. 411. DOI: 10.5281/zenodo.5199477

[6] S. Sengupta et al., "A smart transportation and logistics system-based robot," Indian Army, Technical Report 622-0029-5, Aug. 2022. Accessed: Nov. 2, 2026. [Online]. Available: https://www.researchgate.net/publication/365805846_Smart_Transportation_and_Logistics_System_based_Robotic_Platform

Article is protected by German copyright law

[7] F. G. Martin, "The handy board technical reference," MIT Media Laboratory, Cambridge, MA, USA, Technical Reference, Jan. 12, 1998. Accessed: Nov. 2, 2026. [Online]. Available: https://www.cs.uml.edu/~fredm/handyboard.com/oldhb/techdocs/hbmanual.pdf

[8] M. Banzi and M. Shiloh, *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*, 4th ed. San Francisco, CA, USA: Maker Media, Inc., 2022, 288 pp., ISBN: 9781680456899. Accessed: Nov. 2, 2026. [Online]. Available: https://books.google.com/books?id=qChfEAAAQBAJ

[9] S. Monk, *Programming Arduino: Getting Started with Sketches*, 2nd ed. McGraw-Hill Education, 2016, 192 pp., ISBN: 9781259641633. Accessed: Nov. 2, 2026. [Online]. Available: https://books.google.de/books/about/Programming_Arduino_Getting_Started_with.html?hl=id&id=CanFLt8i9PYC&redir_esc=y

[10] H. Barragán, "Wiring: Prototyping physical interaction design," M.S. thesis, Interaction Design Institute Ivrea, Ivrea, Italy, Jun. 2004. Accessed: Nov. 2, 2026. [Online]. Available: https://people.interactionivrea.org/h.barragan/thesis/thesis_low_res.pdf

[11] D. A. Mellis, M. Banzi, D. Cuartielles, and T. Igoe, "Arduino: An open electronics prototyping platform," in *alt.chi Session at the CHI 2007 Conference on Human Factors in Computing Systems*, Alternative CHI presentation, San Jose, CA, USA, Apr. 28–May 3, 2007, pp. 1–11. Accessed: Nov. 2, 2026. [Online]. Available: https://www.semanticscholar.org/paper/Arduino:-An-Open-Electronics-Prototyping-Platform-Mellis-Banzi/cb7cd3288ff84d76060f8e599a8e172295e6dca5

[12] D. Kushner. "The making of arduino, How five friends engineered a small circuit board that's taking the diy world by storm," IEEE Spectrum, Accessed: Nov. 2, 2026. [Online]. Available: https://spectrum.ieee.org/the-making-of-arduino

[13] Arduino Team. "One board to rule them all: History of the arduino uno," Arduino, Accessed: Nov. 2, 2026. [Online]. Available: https://blog.arduino.cc/2021/12/09/one-board-to-rule-them-all-history-of-the-arduino-uno/

[14] M. Margolis, B. Jepson, and N. R. Weldin, *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2020, ISBN: 9781491903513. Accessed: Nov. 2, 2026. [Online]. Available: https://www.oreilly.com/library/view/arduino-cookbook-3rd/9781491903513/

[15] J. Blum, *Exploring Arduino, Tools and Techniques for Engineering Wizardry*. Indianapolis, IN, USA: John Wiley & Sons, Inc., 2013, 384 pp., ISBN: 9781118549360. Accessed: Nov. 2, 2026. [Online]. Available: https://www.wiley.com/en-us/shop/general-introductory-computer-science/exploring-arduino-tools-and-techniques-for-engineering-wizardry-2nd-edition-p-9781119405375

[16] H. Bi and M. Kong, "Design of a logistics automated guided vehicle," *Instrumentation*, vol. 9, no. 1, Mar. 15, 2022. DOI: 10.15878/j.cnki.instrumentation.2022.01.001

Arif Sirinterlikci, Ph.D., CMfgE, CPEM is a University Professor of Industrial and Manufacturing Engineering in the School of Engineering and Science at Robert Morris University. He holds a Ph.D. in Industrial and Systems Engineering from the Ohio State University and is a co-author of the recently published book, *A Comprehensive Approach to Digital Manufacturing* (Cham, Switzerland, Springer).

Jérôme Rutinowski, Ph.D., is the Head of Research and Operations at TU Dortmund University (Dortmund, Germany) and the 2026 Rooney International Visiting Scholar at Robert Morris University. He holds a Ph.D. in Mechanical Engineering from TU Dortmund University.

Article is protected by German copyright law