Using Shields as Educational Tools for Firmware Development on Engineering Courses

1st Otávio Barbosa
Postgraduate Program in Mechatronic Engineering
Federal University of Rio Grande do Norte (UFRN)
Natal/RN, Brazil
otavio.barbosa.017@ufrn.edu.br

2nd Samaherni Dias
Electrical Engineering Department
Federal University of Rio Grande do Norte (UFRN)
Natal/RN, Brazil
sama@laci.ufrn.br

***Abstract*—Engineering education demands the integration of theory and practice, yet most commercial kits are costly and limited to specific experiments. This work presents an educational shield compatible with the UNO R3 pin standard to support embedded systems learning. The shield features LCD and 7-segment displays, push-buttons, a buzzer, an RGB LED, a potentiometer, an R-2R ladder network, and interfaces for I^2C, SPI, and UART protocols, broadening experimentation options. The methodology comprised requirement definition, component selection, schematic design, and PCB development using EasyEDA. The shield enables practical experiments such as digital safes, digital filters (IIR, FIR), Software-Defined Radio (SDR), control systems (PID), timers with Human-Machine Interface (HMI), I^2C-based schedulers, and spectrum analyzers.**

***Index Terms*—Educational Kit; Embedded Systems; Arduino Shield; UNO R3.**

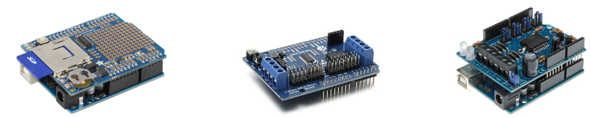
I. Introduction

Engineering programs seek to combine theory with practical laboratory activities, fostering a deeper understanding of real engineering challenges [1]. However, many commercial educational kits provide predefined experiments or lack proper documentation, making their use difficult. Moreover, such systems are often costly and may require technical support during operation [2].

Since its creation in 2005 at the Interaction Design Institute Ivrea in Italy, Arduino has transformed access to open-source hardware by enabling affordable and reusable solutions. It comprises development boards based on microcontrollers like the Atmega328p, simplifying prototyping and electronic applications. The Arduino IDE, built on C/C++, offers an accessible programming environment supported by extensive documentation and an active community [3].

Arduino has spread across multiple fields, especially education, replacing costly and hard-to-access equipment in universities and schools. In [4], it was used to teach oscillator concepts, while [5] demonstrates its application in laboratory experiments within the Master's program in Robotics and Automation at the University of Alicante.

The Arduino UNO R3 uses a standard pin configuration compatible with various shields—add-on boards that expand its functionality—as shown in Figures 1(a) and (b). This standard has been adopted by other development boards, such as the STM32F411RET6 from ST Microelectronics, simplifying shield integration. Intuitive and easy to install, as illustrated in Fig. 1(c), Arduino shields draw power and control signals directly from the host board [1].



(a) (b) (c)

Fig. 1. Commercially available shield examples.

This work proposes the development of a shield compliant with the UNO R3 standard, compatible with multiple platforms, cost-effective, and adaptable to diverse educational contexts. The shield integrates essential components into a user-friendly tool that supports a wide range of experiments and projects. By removing the need for breadboard assembly, it allows students to focus on firmware development and problem-solving, enhancing the learning process.

The shield's programming is not restricted to predefined APIs or libraries, aiming instead to extend peripheral control through the microcontroller. Consequently, it supports multiple programming languages, such as C and Assembly, fostering the development of optimized and application-specific solutions.

The shield's functionalities include data display, user interaction, and analog-to-digital and digital-to-analog conversion. For visualization, it features an LCD and three 7-segment displays. The board also includes a buzzer, push-buttons, and an RGB LED, while signal processing is supported by an R-2R resistive ladder for digital-to-analog conversion and a potentiometer for analog input simulation.

II. Literature Review

According to [6], educational material companies often offer costly products with limited compatibility and expansion, restricting access in less developed regions. Additionally, these products are typically confined to institutional use, limiting students' autonomy and learning beyond the laboratory.

The BYOD (Bring Your Own Device) concept encourages the use of personal devices in educational or corporate settings [7]. Combined with the open hardware philosophy, it makes the proposed shield more accessible and flexible, allowing use beyond the laboratory and fostering student autonomy and continued learning.

The shield's simple peripheral integration removes the need for circuit assembly, allowing students to focus on firmware development. As noted in [5], this benefits various engineering profiles: students from electrical and electronic areas can improve programming skills, while those from computer science can work on sensor and actuator optimization and control.

Another goal of the shield is to offer diverse practical interactions with the board's hardware components. It includes four push-buttons for digital input control and a built-in potentiometer for analog input, allowing the simulation of external signals.

The shield includes a buzzer and an RGB LED for interaction through sound and color variation. Both use digital pins configurable via PWM (Pulse Width Modulation), allowing exploration of concepts such as duty cycle. An integrated R-2R network enables 8-bit digital-to-analog conversion, offering a practical context for DAC studies.

The shield includes connectors that extend pins for various communication protocols, enabling integration with external devices such as additional microcontrollers or sensors. This allows system expansion through interfaces like I^2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver/Transmitter), and SPI (Serial Peripheral Interface).

Visual data output is provided by an LCD and a 7-segment display. The LCD requires implementing its specific protocol, encouraging students to consult the datasheet for practical understanding, while the 7-segment display supports learning about the PoV (Persistence of Vision) effect.

III. Methodology

The methodology used in designing the shield proposed in this work begins by defining requirements. Next, the process involves specifying the electronic schematic, components used on the shield, their placement, the PCB (Printed Circuit Board) routing, shield assembly and testing, and finally, the development of support libraries.

The shield will support a range of problems that explore digital input/output control with and without the use of interrupts, digital signal processing of analog and digital inputs, human-machine interface (HMI) development, visual and sound signaling, actuation on peripherals using both analog and digital signals, and the development of various types of digital communication protocols.

Examples include firmware development for a digital vault, digital filters (e.g., IIR, FIR), Software-Defined Radio (SDR), control techniques (e.g., PID), timers with HMI, timer scheduling with I^2C communication, spectrum analyzer, among others.

The shield is electrically and physically compatible with the Arduino UNO R3 pinout standard, considering several I/O features, such as: analog and digital inputs, push-buttons, adjustable analog input, outputs for LCD and 7-segment displays (with PoV), visual and sound signaling, DAC-based analog output, and digital communication via I^2C, SPI, and UART.

The shield has five analog pins, thirteen digital pins, and power lines (GND, Vin, 5V, 3.3V). Six digital pins support PWM, while the UART and SPI signals use other digital pins. I^2C communications uses two analog pins, allowing full functionality with boards that keep the same pin configuration.

The shield integrates a 3-digit 7-segment display driven by seven segment lines and three digit-select lines. Segment decoding is handled by an SN74LS48N. Three BC547 NPN transistors switch the digit commons to multiplex the display, and multi-digit rendering relies on persistence-of-vision (PoV).

For user interaction, the shield employs a 16×2 character LCD (LCD1602C). To save pins, it runs in 4-bit mode, with the R/W line tied low for write-only control. Display contrast is adjusted via the onboard potentiometer.

For input, we use 4-pin tact switches; each switch has two internally shorted pairs. One pair is tied to GND, and the other connects to a digital pin through a pull-up resistor. The pull-up holds the line high; pressing the button pulls it low (active-low).

A buzzer and an RGB LED provide audio and visual feedback. Both are PWM-driven: for the buzzer, the PWM frequency sets the tone while the duty cycle controls loudness; for the RGB LED, the duty cycle on each channel sets the red, green, and blue intensities.

A multi-turn potentiometer adjusts the analog input pin of the microcontroller, which is, internally, connected to the ADC converter, resulting in a proportional digital value stored in a register.

An 8-bit R–2R ladder serves as a DAC: eight digital pins act as the bit inputs. The R/2R topology produces an output voltage proportional to the binary code applied at its inputs.

Because R3 boards don't provide enough digital I/O to drive all peripherals at once, the shield uses 3-pin headers with jumper blocks to select among peripherals that share a pin. Unselected peripherals leave the pin available. Additional 2-pin headers act as feature enables.

IV. Results

The shield is a PCB that integrates the peripherals listed in Table I, commonly used in instructional labs. Development proceeded from component selection and schematic wiring to PCB layout. A 3D board view was also generated to preview component placement and mechanical fit.

PCB development was carried out in EasyEDA, a web-based EDA suite that integrates schematic capture, community-supported component libraries, and PCB layout/routing; its libraries simplified part search and selection and covered all required functionality.

Beyond schematic and layout, EasyEDA generates 3D board previews and applies JLCPCB-aligned fabrication rules, which simplifies DRC, quoting, and the transition from design to production.

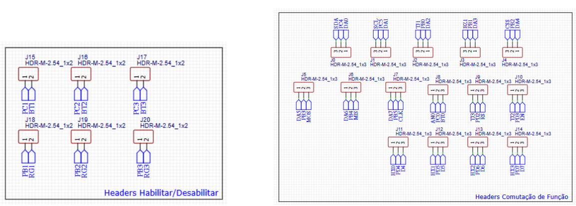
Id	Component	Quantity	Price (USD)
1	BUZZER-12X9	1	0,480
2	HEADER-M-2.54_1x4	3	0,043
3	HEADER-M-2.54_1x3	15	0,187
4	HEADER-M-2.54_1x2	8	0,023
5	BC547 NPN transistors	3	0,017
6	10k multi-turn potentiometer	2	0,080
7	22K - resistor	9	0,020
8	10K - resistor	7	0,020
9	0,47k - resistor	5	0,020
10	1k - resistor	13	0,020
11	0,1K - resistor	1	0,020
12	RGB LED	1	0,020
13	Push-Button	4	0,044
14	SN74LS48N	1	0,387
15	LCD1602 2X16	1	4,860
16	3-digit 7-Segments	1	1,450
17	Printed Circuit Board	1	1,000

TABLE I
LIST OF ELECTRONIC COMPONENTS OF THE SHIELD.

A. Schematic

The NetLabel tool simplifies the schematic by connecting identically named nets without explicit wires, improving readability. The UNO R3 connections component uses NetLabels for all analog and digital pins, as well as the power rails (GND and 5 V).

The connection headers allow specific pins to be enabled or their functions reassigned (via jumper blocks), as shown in Figs. 2(a) and 2(b).



(a) (b)

Fig. 2. Schematic of the header connections.

The push buttons use a 1 kΩ pull-up. Each button input pin is tied to VCC through the pull-up and reads high when idle; pressing the button shorts the line to GND, driving it low (active-low). The communication protocols include headers for external connection with sensors or other devices and the sensors require an extension for the power supply. The multi-turn potentiometer is wired between GND and 5 V, with its wiper feeding the ADC input. The 8-bit R–2R ladder is implemented with 10 kΩ and 22 kΩ resistors and connects to eight digital output pins. The passive buzzer is driven through a resistor of 100 Ω. The RGB LED uses 470 Ω resistors on each channel, with a common-cathode connection to GND; brightness is set via per-channel PWM.cThe LCD is hardware-configured to operate in 4-bit mode, as shown in Fig. 3. The contrast is set by a multi-turn potentiometer wired between 5 V and GND (feeding the V_O pin). The backlight remains continuously enabled, with a 470 Ω series to limit current.

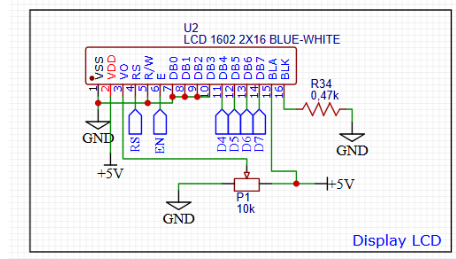


Fig. 3. Schematic of the LCD display connections.

The 7-segment display is controlled by the SN74LS48N IC and transistors, as shown in Fig. 4. The IC pins labeled with the prefix “HX” and the transisitors pins are used by the microcontroller to send a 4-bit binary number and to activate each digit individually. This allow to IC convert binary to BCD and permits to use 2 or 3 digits using PoV effect.

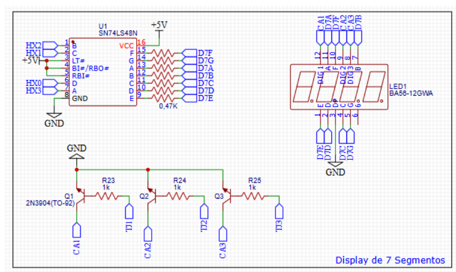


Fig. 4. Schematic of the 7-segment display connections and its peripherals.

B. Printed Circuit Board

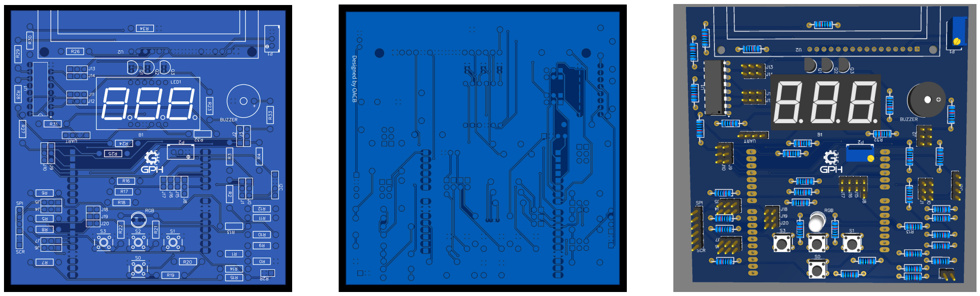
The shield follows a standard model with specific physical dimensions: rectangular boards measuring 100×100 mm with two layers (Top and Bottom). This configuration meets the requirements for PCB prototyping used in the industry. The minimum design rules include a trace width $\geq$ 0.254 mm, spacing $\geq$ 0.16 mm, via diameter $\geq$ 0.610 mm, and hole diameter $\geq$ 0.305 mm.

In the Top Layer, the traces are horizontal, as shown in Fig. 5(a), while in the Bottom Layer they are vertical, as shown in Fig. 5(b). Both layers include a GND plane connecting all ground pins across the remaining copper surface, with other pins isolated from this layer.

It is worth noting that fixed routing directions in each layer help standardize and simplify the process of creating connections between the shield’s components.

C. 3D Model

The 3D model in Fig. 5(c) shows the arrangement of the shield’s components on the top layer. The LCD extends beyond the shield’s physical boundaries to free space in the shield. All components are of the PTH (Pin Through Hole) type, chosen for their availability in standard electronics stores and ease of manual assembly.



(a) (b) (c)

Fig. 5. Top layer (a), bottom layer (b) of the PCB and 3D model (c) of the shield.

D. Production

The result of the first version of the shield is in Fig. 6. The board is fully equipped with peripherals installed through electronic soldering of the components. The power and programming inputs of the microcontroller remain unobstructed, allowing access for code upload and power supply.

Fig. 6. Shield installed on the Arduino UNO R3.

Fig. 7 presents an example of the firmware architecture. For introductory courses, the complete library set may be used. Meanwhile for low-level embedded systems courses, no libraries are provided and the solutions can be divided in HAL (Hardware Abstraction Layer), the DRV (Drivers) layer manages the shield peripherals, the MDW (Middleware) layer integrates an adapted version of FreeRTOS, and the App (Application) layer contains the user's specific code.

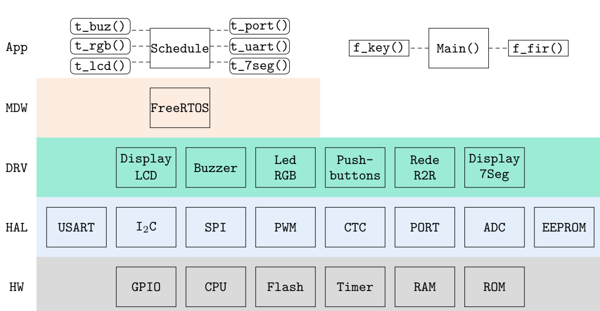


Fig. 7. Example of a firmware architecture.

V. Conclusion

This work presented the development of an educational shield that is financially accessible (less than 13 USD) and compatible with microcontroller development boards that follow the UNO R3 layout. The main objective was to consolidate electronic components in a way that facilitates their use during activities involving firmware programming.

Integrating all peripherals taxed the available MCU GPIO, so the design adopts pin sharing via headers and jumper blocks. This modular scheme lets instructors reconfigure the shield for specific exercises, enabling one or multiple peripherals to be used concurrently.

The shield ships without dedicated libraries, encouraging students to implement their own firmware solutions. Although, The shield uses readily available components and is easy to assemble, enabling students to build their own boards and deepen their understanding of circuit wiring and soldering practice. Additionally, its recommended to students and tutors to colaborate with the shield community at Github [8].

References

[1] G. Takács, E. Mikuláš, M. Gulan, A. Vargová, and J. Boldocký, "AutomationShield: An open-source hardware and software initiative for control engineering education," *IFAC-PapersOnLine*, vol. 56, pp. 9594–9599, 2023.

[2] A. Gongora, J. Fernández-Madrigal, A. Cruz-Martín, V. Arévalo-Espejo, C. Galindo-Andrades, C. Sánchez-Garrido, J. Monroy, and J. Fernández-Cañete, "Optimizing subject design, timing, and focus in a diversity of engineering courses through the use of a low-cost Arduino shield," in *Proc. ICERI Conf.*, pp. 1–8, 2020.

[3] H. Kondaveeti, N. Kumaravelu, S. Vanambathina, S. Mathe, and S. Vappangi, "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations," *Comput. Sci. Rev.*, vol. 40, pp. 1–28, 2020.

[4] P. Pereira and M. Silva, "Construção de um kit experimental com Arduino para ensino de oscilações em tempo real," *Rev. Bras. Ensino Fís.*, vol. 43, pp. 1–6, 2021.

[5] F. A. Candelas, G. J. García, S. Puente, J. Pomares, C. Jara, J. Pérez, D. Mira, and F. Torres, "Experiences on using Arduino for laboratory experiments of automatic control and robotics," *IFAC-PapersOnLine*, vol. 48, pp. 105–110, 2015.

[6] A. Takács, G. Eigner, L. Kovács, I. Rudas, and T. Haidegger, "Teacher's kit," *IEEE Robot. Autom. Mag.*, vol. 23, pp. 30–39, 2016.

[7] A. Garrigós, J. Blanes, R. Gutiérrez, I. Blanquer, and M. Cantó, "Designing Arduino electronic shields: Experiences from secondary and university courses," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, pp. 934–937, 2017.

[8] Barbosa, O. "uC-MicroLab". Github. Available: https://github.com/otavioacb/uC-MicroLab. 2025.