Empowering STEAM Activities With Artificial Intelligence and Open Hardware: The BitDogLab

Article *in* IEEE Transactions on Education · June 2024

DOI: 10.1109/TE.2024.3377555

Fabiano Fruett
State University of Campinas (UNICAMP)

Empowering STEAM Activities With Artificial Intelligence and Open Hardware: The BitDogLab

Fabiano Fruett, *Member, IEEE*, Fernanda Pereira Barbosa, Samuel Cardoso Zampolli Fraga, and Pedro Ivo Aragão Guimarães

***Abstract*—Open-source hardware and software platforms have played an important role in democratizing access to technology and education in computer science and engineering. Recent advancements in tools, such as KiCad, MicroPython, and Thonny, integrated development environment potentially accelerate low-budget educational applications, providing a smooth and consistent learning curve for users. The BitDogLab, utilizing these platforms, is designed and developed for Science, Technology, Engineering, Arts, and Mathematics activities based on the Raspberry Pi Pico Board, featuring artificial intelligence (AI)-supported programming via the GPT model. BitDogLab is an open-source hardware solution with a flexible design for continuous expansion in both software and hardware areas. Its functionality, based on open-source code, can be enhanced by integrating various software resources. Additionally, BitDogLab's adaptability to sophisticated hardware like robotic systems or advanced sensors expands its educational applications. An evaluation of BitDogLab, conducted through a hands-on workshop at the Latin America Open Technology Conference—Latinoware 2023, revealed unanimous interest among 36 respondents, mainly for teaching programming (58%), and valuing its low cost, diverse components, and ease of AI-assisted programming. 75% of respondents expressed a strong desire to use BitDogLab, despite 19% preferring familiar boards like Arduino or Micro:bit. The project's open distribution encourages community contributions, promoting collaborative learning and innovation from children to preuniversity students.**

***Index Terms*—Artificial intelligence (AI) in education, open hardware, project-based learning, Science, Technology, Engineering, Arts, and Mathematics (STEAM) education.**

Manuscript received 1 July 2023; revised 30 November 2023; accepted 15 February 2024. This work was supported in part by the IEEE Electron Devices Society through the Project titled "School 4.0—Hands-on Activities for Young Students with Project-Based Learning in STEM" under Grant 2023.13; in part by the National Council for Scientific and Technological Development (CNPq), Brazil, under the National Institute of Science and Technology (INCT) Program, specifically through the National Institute of Science and Technology for Micro and Nanoelectronic Systems (Namitec). This funding and support reflect a collaborative effort to advance scientific and technological development, under the coordination of the Ministry of Science, Technology, and Innovations (MCTI), Brazil. *(Corresponding author: Fabiano Fruett.)*

Fabiano Fruett, Samuel Cardoso Zampolli Fraga, and Pedro Ivo Aragão Guimarães are with the School of Electrical and Computer Engineering, Universidade Estadual de Campinas, Campinas 13083-872, Brazil (e-mail: fabiano@unicamp.br; samuelczf@gmail.com; pedrodsk@gmail.com).

Fernanda Pereira Barbosa is with the Elementary Education I, SESI-SP-Itatiba-Unit CE013, Itatiba, Brazil (e-mail: fernandapbarbosa@gmail.com).

Digital Object Identifier 10.1109/TE.2024.3377555

I. Introduction

BRAZIL, along with other developing countries, faces a significant challenge regarding the inclusion of technology in elementary and secondary school classrooms, especially in activities or projects involving Science, Technology, Engineering, Arts, and Mathematics (STEAM) [1]. This challenge is influenced by various factors, such as teacher training, IT infrastructure, social and economic inequality, and more [2]. In Brazil, one particular factor is the reality of cascading taxes (federal, state, and municipal) that make technologically dependent solutions, subject to importation, prohibitively expensive for many schools, especially public ones. These taxes, including the profit margin from the trade around these educational tools, result in an approximately 120% increase in the final cost for consumers [3]. Furthermore, elementary and high-school teachers face difficulties in using electronic components connected to a breadboard [4]. These difficulties mainly arise from common problems of poor contact on breadboards and the fragility of connections, which can lead to errors and frustration during practical teaching activities. Ideally, electronic boards designed for educational purposes should be easy to use and operate under the principle of open-source hardware, providing their manufacturing "recipe" to encourage sharing and continuous improvement. It is also important for them to be well-documented to facilitate independent learning for students [5]. The programming environment of the board should prioritize logical thinking over the syntax of the programming language, as the latter can become an obstacle for students who are starting to develop their programming and computational thinking skills [6].

An initiative evaluated the effectiveness of Fab Lab-based learning and shows that students are more interested in science lessons after using Fab Lab-based learning [7]. This research corroborates an analysis of 225 studies comparing student performance in STEM courses using traditional learning versus active learning methods, revealing a significant positive effect of active learning on student performance [8]. Given this scenario, it is essential to seek low-cost solutions that meet the requirements while providing quality learning experiences and effective development of student's skills. Therefore, we have developed BitDogLab: an open-source, low-cost, and user-friendly hardware board designed specifically to teach STEAM concepts in an interactive and engaging manner. This article will describe the development and capabilities

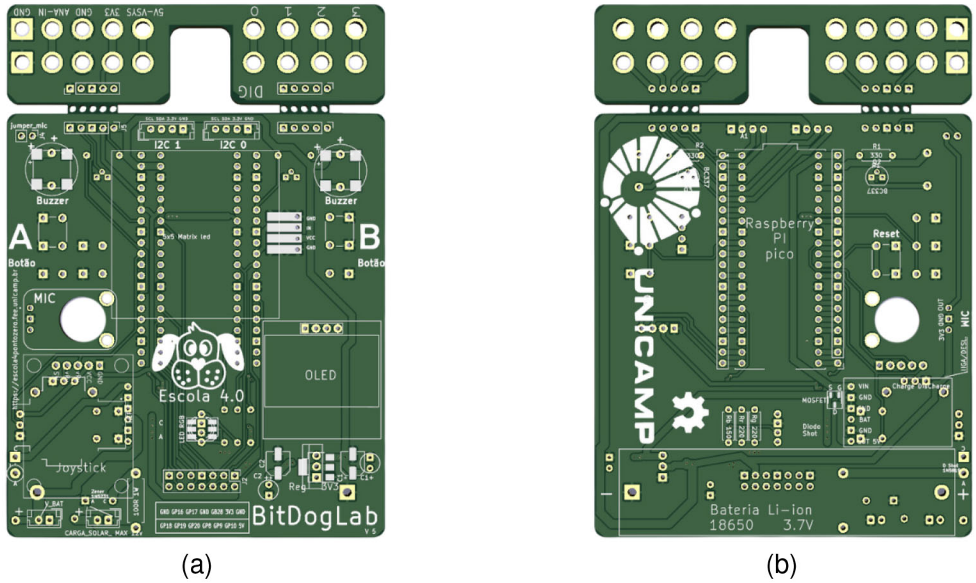

Fig. 1. Layout design of the BitDogLab PCB: (a) front and (b) back side.

of the BitDogLab, demonstrating how it can be programmed with the support of artificial intelligence (AI) to facilitate and expedite the learning process. BitDogLab accentuates active learning by enabling learners to tackle more complex programming challenges and interact with the devices in novel, nontraditional ways. The integration of AI into the learning process not only simplifies programming tasks but also opens new avenues for exploration, making it a truly distinctive contribution in the realm of STEAM education.

II. Design of the BitDogLab Printed Circuit Board

While the use of protoboards is a common educational tool, it can pose technical challenges for elementary and high-school teachers and students. Therefore, we consider it advisable to avoid the protoboard as the first form of contact with electronic components. In this way, our project is based on a specially designed printed circuit board (PCB) for educational activities. We have named the board BitDogLab, in honor of our mascot BitDog, which is a playful element of the Escola 4.0 project [9] to which this initiative belongs. This PCB uses as its central component a Raspberry Pi Pico, a module with a low-cost (U.S.$6) and high-performance ARM microcontroller, which offers programming flexibility and robust features for teaching concepts of STEAM [10]. The BitDobLab PCB was designed to have two parts that can be easily separated. Its layout is shown in Fig. 1(a) and (b). The upper part, which has several drilled holes, is a terminal bar to facilitate connection with alligator-type clamps. The main core of the board is where the components are soldered. The main core of the board measures 98 mm in length, 88 mm in width, and 1.6 mm in thickness. The terminal bar is 23 mm in length, while retaining the same width and thickness as the main core.

The BitDogLab PCB connects the microcontroller with various peripheral components that can be soldered onto the board as pedagogical needs demand or as the user's interest increases. These connections are made with two levels of metallization, one on each face of the board. The board was designed to offer flexible and accessible alternatives for electronic components, both in the through-hole assembly (PTH) mode and in the surface-mount device (SMD) mode. This approach allows the user to choose the best-assembly strategy according to their skills and learning goals. For example, PTH components may be more suitable for beginners due to the ease of soldering, while SMD components can offer a more advanced experience, closer to what is found in the industry. This feature is shown in Fig. 2.

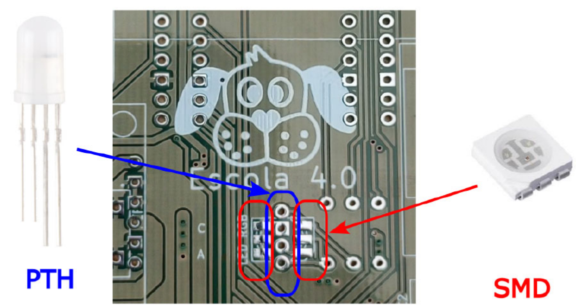

Fig. 2. Detail of the choice of the type of component and how to solder the colored LED on the BitDogLab PCB. The choice depends on the user's skills and knowledge, it can be: through-hole (PTH) or surface mount (SMD).

The repository with all Gerber files for manufacturing and also the KiCad source file for BitDogLab is available at [11]. There are several manufacturers around the world that offer PCB prototyping. The manufacturing cost of this board depends on the quantity requested. For less than a dozen units, the approximate cost is $1 per board. The components of the project are detailed in the next section.

A. Materials

In addition to the BitDogLab Board, the materials used in this project are: a five by five matrix of colored (RGB) LEDs, two buzzers, three push buttons, a microphone, a joystick, an RGB LED, a mini display, and an insulation-displacement

TABLE I
BITDOGLAB'S BILL OF MATERIALS

item	tag	description	qnty
1		20-pin female header	2
2		Raspberry pi pico (H or W)	1
3	LED	5mm RGB LED common cathode	1
4	Rb	100 ohm resistor 1/8 W 5%	1
5	Rg, Rr	200 ohm resistor 1/8 W 5%	2
6	A, B, Reset	button	3
7	A, B	passive buzzer	2
8	Q1, Q2	NPN transistor BC337	2
9	R4, R5	330 ohm resistor 1/8W 5 %	2
10	M	5 x 5 Matrix RGB LEDs 5050 WS2812B	1
11	JOY	KY023 analog joystick	1
12	reg	Voltage regulator 3V3 LM1117	1
13	MIC	PDM microphone module MP34DT01	1
14	C1-C6	100uF 16V electrolytic capacitor	6
15	DSP	0.96" I2C 128 x 64 OLED display	1
16	bat	rechargeable battery holder for PCB	1
17	CHM	charge module tp4056 mini USB	1

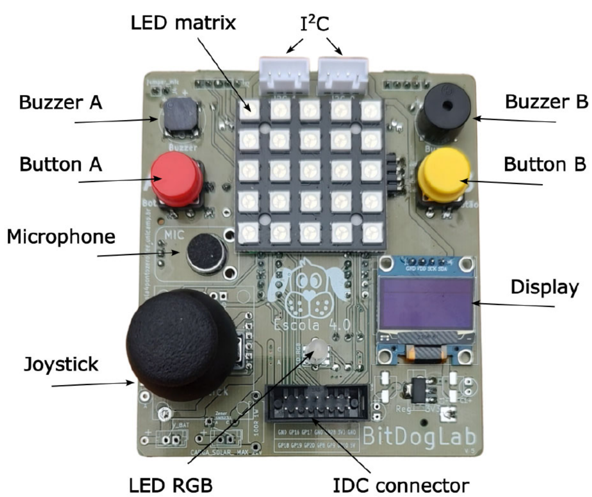

Fig. 3. Front view of the assembled BitDogLab with all its components.

connector (IDC) that allows externally connecting other components or boards, expanding its functions. In addition to components, such as resistors and capacitors. All of these components are commercially available and easy to acquire. The list of materials is shown in Table I.

B. Method for Hardware Implementation

With the BitDogLab PCB in hand, we suggest that each component be included, by soldering it onto the board, in the order that appears in Table I. In this way, after incorporating items 1 and 2, the user has the possibility to progress and program using the approach presented in this article. As more components are included, your options for use in STEAM activities increase along with your programming options. The soldering of the components onto the board is a task that requires appropriate guidance and safety precautions. With proper supervision, young people from 12 years old are capable of performing quality soldering work. There are numerous tutorials and online videos that offer step-by-step instructions on how to solder correctly. Although this task requires some practice, once acquired, soldering skills become a lifelong useful competency. Furthermore, with the right guidance, the activity can become not only educational but also fun. The PCB with all the soldered components is shown in Fig. 3.

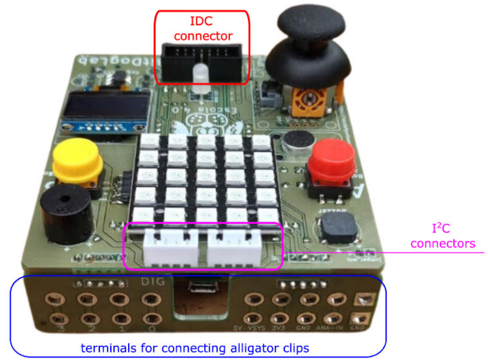

Fig. 4. Close-up view of the connection terminals of the BitDogLab that can be connected with alligator clips.

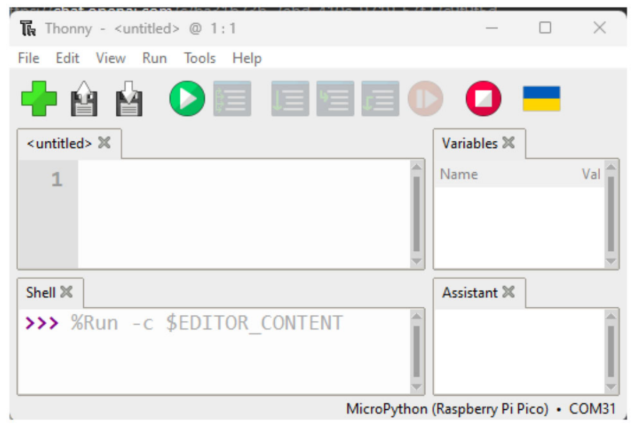

Fig. 5. Thonny's IDE.

The board has some terminals for connecting alligator clips. In this way, the board's expandability is facilitated by these terminals and also by a IDC with 14 vias, which are detailed in Fig. 4. An IDC is an electrical connector that allows connections to be made with insulated cables without the need to remove the insulation. In practice, this means that electronic components can be easily connected to cables, without the need for complex cable preparation. This simplifies the assembly and maintenance process, making it ideal for educational applications where ease of use and expansibility are fundamental.

III. PROGRAMMING ENVIRONMENT

The hardware we described earlier is the physical implementation of our tool for empowering STEAM activities. The core of this tool is a microcontroller that must be programmed to command or receive signals from components, sensors, and peripherals. This microcontroller can be programmed in a variety of ways [12]. However, the main contribution of this article is to propose a method that allows anyone, even without any prior programming experience, to use this tool in STEAM projects in the classroom.

The programming environment comprises the integrated development environment (IDE), Thonny [13], and a large language model (LLM) of the GPT type. Thonny is a Micropython IDE that is incredibly beginner-friendly, with features like step-by-step debugging and variable monitoring. The main interface of Thonny is shown in Fig. 5 [13].

Micropython is a compact and efficient implementation of the Python language that can be used to program microcontrollers [14]. Chat GPT is an AI feature [15] that can assist users in programming the BitDogLab based on its hardware configuration database (HCD).

IV. Hardware Configuration Database

The HCD, which we propose, is a structured textual list containing information about the BitDogLab Board's hardware configuration. This database includes details about the arrangement and function of the hardware components, including but not limited to: connected peripherals and the GPIO pins of the Raspberry Pi Pico microcontroller. This set of information simplifies understanding the board's architecture and allows users, and even AI models like ChatGPT, to interact efficiently with the hardware. For each peripheral, the database includes details about the connected GPIO pin, the expected function of this component, and any specific parameters associated with its use. The HCD not only serves as a reference guide but also aids in the development of STEAM applications and activities with the BitDog PicoLab Board. Moreover, it is essential for customizing the AI model's responses, enabling it to provide specific and relevant information, particularly concerning programming code, based on the board's hardware configuration. The BitDogLab HCD is the textual list shown as follows.

1) A common-cathode RGB LED has the red electrode connected to GPIO 12 through a 200-ohm resistor, the green pin connected to GPIO 13 through a 200-ohm resistor, and the blue pin to GPIO 11 through a 100-ohm resistor.
2) A button, identified as Button A, is connected to the Raspberry Pi Pico's GPIO5. The other terminal of the button is connected to the board's GND.
3) Another button, identified as Button B, is connected to the Raspberry Pi Pico's GPIO6. The other terminal of the button is also connected to the board's GND.
4) A buzzer, identified as Buzzer A, is connected to the Raspberry Pi Pico's GPIO21.
5) Another buzzer, identified as Buzzer B, is connected to the Raspberry Pi Pico's GPIO10.
6) The input pin of a 5-line by 5-column 5050 RGB LED matrix type WS2812B (Neopixel) is connected to GPIO7.
7) A KY023 type analog joystick has its VRx output connected to GPIO26 and its VRy output to GPIO27. Its SW button is connected to GPIO22.
8) A 0.96-inch 128-column by 64-line OLED display with I2C communication has its SDA pin connected to GPIO14 and the SCL pin to GPIO15.
9) An IDC box of 14 pins is used for hardware expansion and is connected to the Raspberry Pi Pico as follows: pin 1 with GND, pin 2 with 5 V, pin 3 with 3V3, pin 4 with GPIO8, pin 5 with GPIO28, pin 6 with GPIO9, and pin 7 with analog AGND. Pin 8 with GPIO4, pin 9 with GPIO17, pin 10 with GPIO20, pin 11 with GPIO16, pin 12 with GPIO19, pin 13 with GND, pin 14 with GPIO18.
10) A terminal bar that allows the connection of alligator clip-type terminals is connected to the Raspberry Pi Pico as follows: the DIG 0, 1, 2, and 3 electrodes are connected to GPIOs 3, 2, 1, and 0, respectively. Additionally, this terminal bar has five more electrodes connected to: analog GND, GPIO 28, GND, 3V3, and 5 V of the Raspberry Pi Pico.

This HCD is the key to making the BitDogLab Board an accessible and versatile tool for STEAM education, providing all the necessary information for users and AI models to interact efficiently with the hardware.

V. AI-Assisted Programming

In the following, we will describe how an AI algorithm can be used as a Programming Support Tool for BitDogLab. When we say "programming using AI," this may be interpreted as if the AI is executing the programming autonomously. However, current AI models, including the GPT model, do not yet have this capability. They cannot write complex computer programs independently, and more specifically, they do not have the ability to program hardware boards. What AI models like GPT can do is assist humans in the task of programming. They can provide code suggestions, explain programming concepts, assist in debugging errors, generate small snippets of code in response to specific requests, and so forth. In the case of the BitDogLab project, from the HCD, the AI can help transcribe and explain the code to control different components on the board, such as the RGB LEDs, the buzzers, or the display from commands of buttons, joysticks, or sensors. Therefore, when we say that BitDogLab Board is "programmed using AI," this means that AI is being used as a tool to assist humans in programming the board. In this case, AI is used as an intelligent assistant that can suggest, guide, and explain, but the final task of writing, testing, debugging, and implementing the code is guided and developed by humans.

A. Method for Hardware Programming Using AI

We propose a programming methodology, with AI support, that can be utilized in the development of a variety of interdisciplinary learning projects and activities. This methodology is characterized by an iterative approach, focused on the use of a HCD of open-source hardware, which is incorporated into the conversational platform of the AI model. Before starting to use the methodology, it is assumed that the user is already engaged in a STEAM project that may involve electronics, sensors, actuators, and programming in its implementation. The methodology is summarized in the flow diagram shown in Fig. 6. The steps of the methodology are detailed as follows.

1) *Loading the HCD into Chat GPT:* This involves inserting the detailed information, in the form of a textual list, of the BitDogLab Board into the GPT conversational platform. This includes information about the layout of the hardware components, the functions of each GPIO pin, etc.

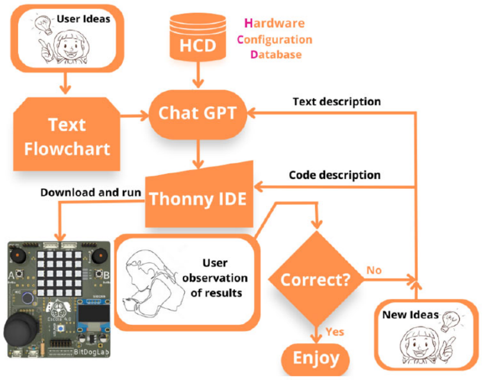

Fig. 6. Flow diagram of the programming methodology with AI assistance.

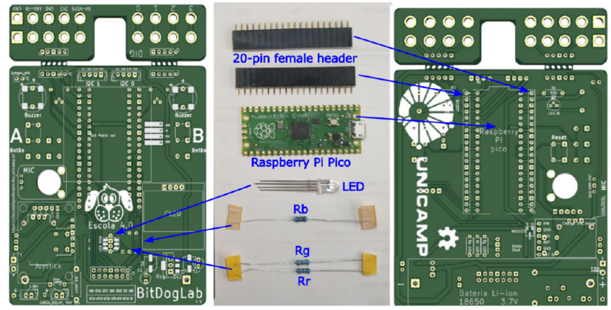

Fig. 7. Front and back view of the PCB showing the basic components to be soldered in the positions indicated on the board to carry out one of the STEAM activities.

2) *Use of a Textual Flowchart:* Editing or writing a Textual Flowchart: the user's ideas for employing the BitDogLab in the STEAM project should be transcribed in the form of a textual flowchart, using logical thinking. This flowchart serves as a guide for the AI model, directing it on how to utilize the HCD to assist in the creation of MicroPython code.
3) *Copying and Loading the Code into Thonny:* After the code is created in collaboration with the AI, it is copied and loaded into the Thonny development environment, which is specific for MicroPython programming and supports the Raspberry Pi Pico.
4) *Download the Code to the Raspberry Pi Pico:* The code is then downloaded to the Raspberry Pi Pico Board through the Thonny IDE.
5) *Execution of the Program:* From a command given by the user in Thonny, the Raspberry Pi Pico Board executes the loaded code, performing the programmed actions.
6) *Observation of Results:* The results of the execution are observed by the user who analyzes and contrasts them with their expectations and needs. At this point, opportunities to increase both the hardware, by soldering or connecting a new component, as well as modifying the code, can also be identified.
7) *Corrections With the AI:* Any deviation from the expected behavior or need for change is discussed with the AI model, which uses the HCD to suggest possible corrections to the code.

This approach allows the student to go through all the stages of a programming cycle and obtain results quickly, driving the process. Even without knowing details of the programming language syntax, the user can develop skills related to logic and computational thinking. As their interest increases, the board can be expanded with new on-board components as well as off-board ones.

VI. STEAM Activity Example With BitDogLab

In this section, we show an example of a STEAM activity using the BitDogLab, which can be programmed using the proposed methodology. As an initial application example, we prepared the PCB by soldering a few components, such as two 20-pin rows (to connect the Raspberry Pi Pico), a single RGB-colored LED, and three resistors on the BitDogLab Board. These components are identified in Table I by their tags: LED, Rr, Rg, and Rb. They are also identified on the PCB in the position they should be soldered, as shown in Fig. 7.

A pedagogical activity example, using only these components, involves programming so that the LED displays the seven colors of the rainbow. This activity, although simple, serves as a programming exercise, and also as a multidisciplinary teaching tool. In this programming exercise, one can explore mathematical concepts by involving the calculation of the percentages that determine the intensity of each color displayed by the LED. The same activity can also be used to introduce science and art concepts related to color theory, including primary colors and their blends. Thus, the activity incorporates STEAM disciplines concepts in an integrated way. After soldering these few components, we follow the methodology previously presented. After opening the free GPT Chat interface, we load the HCD into the dialog window. In the same GPT, using the next dialog window, we load the textual flowchart of operations to be performed. As an example, we can write: Considering our HCD, Could you make a code to vary the colors of the RGB LED with the sequence of the seven colors of the rainbow, starting with red and going to violet? The transition of the seven colors should take 21 s and restart again.

After a few seconds, the response followed by the commented code appears in the Chat GPT response window, as shown in Fig. 8.

The code can be copied simply by clicking on the 'Copy code' icon and pasting it into the Thonny's window [11] as shown in Fig. 9. The step-by-step guide on how to configure the Thonny IDE along with the microcontroller (Raspberry Pi Pico) is out of the scope of this article's innovation and will not be detailed. For users less familiar with the initial setup of the Raspberry Pi Pico using the Thonny IDE, including the installation of Micropython, we recommend the "Getting started with Raspberry Pi Pico" document from the Raspberry Pi Foundation [16] After loading the code into the BitDogLab, the program begins to execute the requested tasks. The result is shown in Fig. 10.

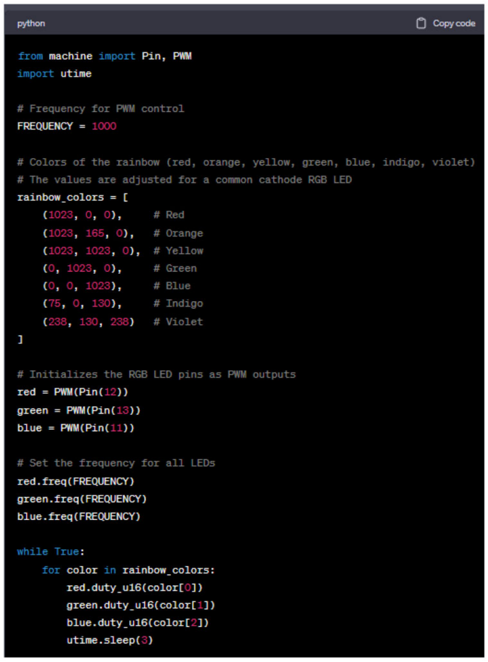

Fig. 8. Chat GPT model's response to the structured question suggested by the method of this article.

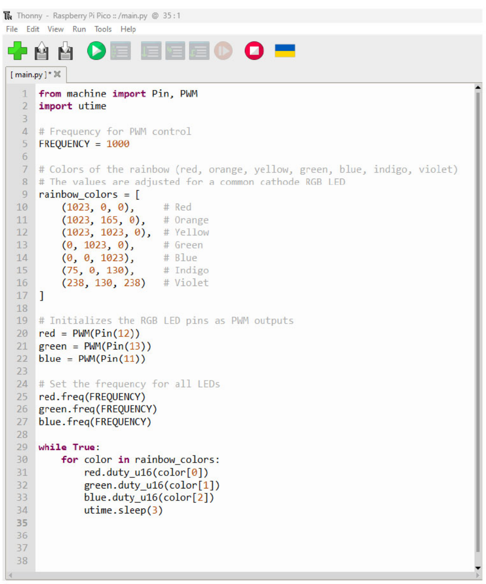

Fig. 9. Thonny IDE with code generated by Chat GPT.

Should there be any errors before the desired outcome is achieved, the error code will appear in the Thonny Shell window. This error information can be copied into a GPT Chat window that will rewrite the program, with the user simply observing and repeating the process. Observing and comparing the achieved result with the desired one, based on the textual flowchart, is a very important stage of the process for the advancement of students' STEAM skills.

For example, note that the difference between orange and yellow, as shown in Fig. 10, is very subtle and may cause difficulty in distinguishing one from the other. This particular result can become a challenge for students to improve the code. Thus, they may be motivated to adjust the intensity of each color in the code to deliver a result where yellow is more distinguishable than orange. For this, they should exercise concepts of proportion, rule of three, color blending, and logical thinking. From this example, the user can expand to include new possibilities, such as playing the Buzzer at a frequency proportional to the wavelength of each color's electromagnetic wave. This would be a suitable example to explain the electromagnetic spectrum and relate its frequencies to a practical outcome. Thus, in a more intuitive way, students could understand the relationship between frequency and wavelength in the electromagnetic spectrum.

Alternatively, they could include the display to show the wavelength (in nm) of each color being displayed by the colored LED. The commands to change the color could also come from the joystick or from claps that would be picked up by the microphone.

VII. Educational Assessment

The main objective of this research was to evaluate the effectiveness and impact of using BitDogLab as a tool to introduce physical programming to nonexperts. Our primary challenge was to avoid overwhelming users with code syntax, opting instead for a textual flowchart aided by a LLM to translate the natural language flowchart into code for loading onto BitDogLab. To achieve this, we conducted a hands-on workshop and subsequently gathered feedback from 36 participants through questionnaires. The participants' profile was diverse, primarily comprising students, teachers, and professionals in fields, such as metallurgy, IT, and sales. Out of the 36 respondents, 8 were students, 8 were teachers, 4 were professionals, and the remainder chose not to disclose their professional activities. Regarding age range, we observed an approximate variation between 18 and 60 years, with a ratio of 10 women, 14 men, and 12 who preferred not to respond. From the collected perceptions, we highlight that 15 of the participants had never had the opportunity to program and found the approach of using the board with AI assistance very interesting. Additionally, 21 participants already had programming experience, mainly with languages, such as C or Ardublocks for Arduino boards, but also with other platforms, such as ESP-32, micro:Bit, Raspberry Pi, Lego, and Kit Atto. Below are some comments from the participants.

1) "A fantastic idea that should have more publicity and reach public schools, encouraging more young people to enter the programming field." - L.H. (technical salesperson)
2) "I found multiple functionalities in a single board that were lacking in others, in addition to the gradual assembly process." - R.C. (robotics teacher)
3) "I liked the idea of programming with AI assistance. It greatly helps young people like me who are not very proficient in programming." - P.G. (student)
4) "I loved the idea and ease of use." - M.L. (teacher)
5) "Very interesting, could be implemented in schools." - F. (IT technician)

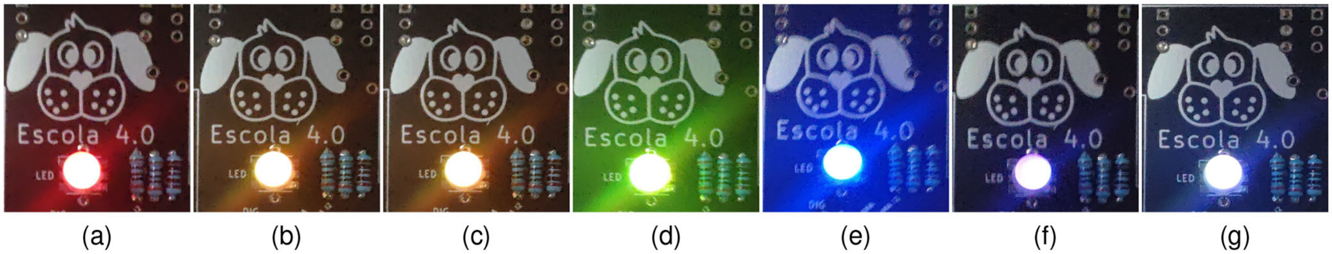

Fig. 10. Result of the RGB LED displaying the seven colors of the rainbow in sequence. (a) Red. (b) Orange. (c) Yellow. (d) Green. (e) Blue. (f) Indigo. (g) Violet.

6) "Interesting for educational robotics and introducing children to programming dynamically." - Anonymous
7) "Focus more on logic and less on AI to develop computational thinking and reasoning." - Anonymous

Regarding the validity of the research, we acknowledge some limitations. The sample size, while adequate for preliminary insights, limits the generalization of our findings. Future research should involve a larger and more diverse group of participants, including different methods of teaching programming, to nonexperts using solutions, such as Micro:bit and Arduino. Additionally, the duration of the study did not allow for a long-term evaluation of learning outcomes, which we suggest for future investigations. This report represents an initial step in evaluating BitDogLab as an educational tool, and we are committed to continuing the studies, including developing a new assessment form for future workshops and activities.

An evaluation of the BitDogLab was implemented, based on a hands-on workshop conducted during Latinoware, held from October 18 to 20, 2023, in Foz do Iguaçu, Paraná State, Brazil. Latinoware is an annual conference attracting a wide range of participants interested in open-source technologies, including hardware and software professionals, educators, and students. This diversity of opinions can enrich the assessment of teaching tools with technology in an environment involving teachers, students, and enthusiasts of open-source technologies in education. Participants of Latinoware typically have some degree of familiarity with emerging technologies, including open hardware and AI, making their opinions and feedback both critical and relevant. The conference often addresses topics related to STEAM education, making the audience more sensitive and pertinent for evaluating an educational tool. This workshop featuring the BitDogLab was prominently advertised in the event schedule. In addition to providing an appropriate space for conducting a workshop, the congress organization also offered a 12 square meter booth where two trained exhibitors (author and coauthors of this work) were available throughout the conference to showcase the board, interact with the public, and promote the workshop schedule.

During the Latinoware conference, a practical workshop was conducted using 20 BitDogLab Boards in a room equipped with 20 computers on which the Fedora operating system was preinstalled. Participants also had access to Chat GPT, preloaded with information from the BitDogLab's Hardware Information Bank. A brief presentation of the BitDogLab

TABLE II
SURVEY RESULTS

1) Did you find the BitDogLab interesting?		
Options	**Ans.**	**%**
YES	36	100%
NO	0	0%
2) What was your purpose for being interested in the BitDogLab?		
Options	**Ans.**	**%**
To teach others to program	21	58%
To program in an uncomplicated way	11	31%
For personal objectives	4	11%
3) What is the most important feature of the BitDogLab?		
Options	**Ans.**	**%**
Low cost of the board (around R$25)	24	67%
Diversity of board components	23	64%
Ease of programming with AI, e.g., using Chat GPT	23	64%
Being an open-source project with free distribution	22	61%
Ability to assemble components as needed	18	50%
4) Do you have any prior experience with programming electronic boards?		
Options	**Ans.**	**%**
YES	22	61%
NO	14	39%
5) What is your general opinion about the BitDogLab?		
Options	**Ans.**	**%**
Very interesting	27	75%
Interesting, but prefer to continue using others I know	7	19%
Unsure	2	6%
Not interesting	0	0%

and its programming capabilities with AI was given to the participants. To facilitate the process, libraries for the board's components were preloaded onto its microcontroller. Participants were initially engaged with an example as follows:

"Ask Chat GPT to generate a Micropython code so that when button A is pressed, it plays a brief melody from the Star Wars movie on Buzzer A, and when button B is pressed, the RGB LED blinks red."

From these examples, participants began to create applications on the BitDogLab that had never been realized before, such as a Pong's game, using the OLED display and joystick, as well as Conway's Game of Life [17].

Some participants also transformed the BitDogLab into a badge, where their name scrolled, letter by letter, across the LED matrix.

After 45 min from the start of the workshop, printed questionnaires were distributed to the participants. This questionnaire contained five questions with multiple-choice answer options. In the end, 36 perception forms were returned. The results were as follows in Table II.

According to the results shown in the table, all 36 respondents expressed interest in the BitDogLab, with the majority

(58%) aiming to teach programming to others. As for valued features, the low cost of the board was the most cited (67%), closely followed by the diversity of components and the ease of programming with AI, both at 64%. The majority (61%) had previously programmed using another electronic board. In terms of overall opinion, 75% consider the BitDogLab very interesting and want to use it, while 19% prefer to continue using other boards they are familiar with. Nobody found the BitDogLab uninteresting. In addition to the questionnaire, some individuals who visited the stand after the workshop were interviewed. These individuals were encouraged to express their opinions, suggestions, or criticisms about the BitDogLab in a free format. The result of this interview can be summarized in the following 6 points.

1) *General Approval and Efficacy:* The majority, 90%, of participants expressed a positive view of the development and, particularly, the educational methodology that the project facilitates. Of these, approximately one-third also expressed appreciation for the utility and functionality of the BitDogLab for learning through trial and error, starting from the Hardware Information Bank, supported by the IABot.
2) *Ease of Use:* Several comments highlighted the ease of programming with the BitDogLab, especially for those, 39%, who had no prior programming experience.
3) *Functionality and Flexibility:* The possibility of gradual assembly and the diversity of components were pointed out by 50% of the interviewees as positive differentials compared to other known solutions.
4) *Promotion and Dissemination:* About 30% of the respondents expressed a desire for the BitDogLab to be more widely publicized, both in schools and in the media, to reach a broader and more diverse audience.
5) *Issues Surrounding AI:* While some see the use of AI as an interesting and helpful way to learn programming, others, about 15%, expressed that it might be more beneficial to focus on developing skills related to the syntax of computational language.
6) *Opportunities for Improvements:* Feedback from 2 out of 36 participants suggests focusing less on AI and more on logic and reasoning, indicating a potential area for adjustment or improvement.

VIII. Comparative Analysis of BitDogLab, Micro:bit, and Arduino

During the activities with BitDogLab at Latinoware, some participants mentioned having prior experience with two well-known electronic platforms: 1) Arduino and 2) Micro:bit. This information makes it interesting to conduct a comparison between these boards and BitDogLab as shown in Table III. However, it is important to emphasize that BitDogLab is not a commercial board, but rather an open solution, where each user builds their own version from an openly available design.

The comparative analysis of the BitDogLab, Micro:bit, and Arduino highlights distinct features of each platform in an educational context. The Micro:bit excels with built-in sensor capabilities, suitable for physics-related projects, while the Arduino is renowned for its flexibility and popularity in maker communities, particularly for analog input projects. BitDogLab stands out with its advanced Dual-core ARM Cortex M0+ processor and substantial memory, supporting complex projects. Its compatibility with MicroPython and C/C++, along with AI-assisted programming, enhances accessibility for beginners. The modular component design and secure connectivity foster hands-on learning and experimentation. As a fully open-source platform, BitDogLab encourages transparency and innovation in teaching hardware and software concepts. Its low cost and sustainable power options (rechargeable battery or solar panel) make it a cost-effective educational tool. BitDogLab's design aligns with active learning and problem-solving in STEAM education, offering a unique blend of affordability, functionality, and adaptability, making it a valuable tool for modern technology and engineering education.

TABLE III
Comparative Analysis of BitDogLab, Micro:bit, and Arduino Features for Educational Use

Feature	Micro:bit	Arduino	BitDogLab
Master Chip	nRF51822	ATmega328P	RP2040
Processor	32-bit ARM Cortex M0	8-bit AVR	Dual-core ARM Cortex M0+
Flash ROM	256KB	32KB	2MB
RAM	16KB	2KB	264KB
Oscillator Speed	16M	16M	Up to 133M
Display	5x5 LED - only red- dot matrix	1 LED	OLED 128x64 and a 5 x 5 RGB LED matrix
Button	3 buttons (A, B, Reset)	None	3 buttons (A, B, Reset)
Bluetooth	Yes	None	Optional (Pico W version)
wifi	None	None	Optional (Pico W version)
Accelerometer on board	Yes	None	None
Digital Compass on board	Yes	None	None
Powered By	USB/Battery Case	USB/AC power	USB, recharble baterry or solar panel
Programming Language	MakeCode, Python	C/C++	MicroPython, C/C++
Additional Features	Edge Connector with support for alligator clamps	Analog Inputs	2 Independent Buzzers, Joystick, Neopixel, Analog Microphone, 2 I2C connectors, 1 14pin IDC connector, secure connectors for alligator clamp
Open Source	Partial	Yes	Yes

IX. Discussion and Conclusion

This work highlights the importance of an integrated and suitable approach to the development of STEAM projects, promoting a holistic and practical understanding, as well as including hands-on experience with electronics and programming. The experimental nature of the BitDogLab enables students to apply theoretical knowledge, by creating and programming real devices, thus solidifying meaningful learning. The BitDogLab also offers interactive features that enhance the educational experience. This is exemplified by the correlation of buzzer frequencies with electromagnetic wavelengths emitted by LEDs, providing a practical and multimodal approach to teaching complex concepts, such as the electromagnetic spectrum. The implementation of various user interfaces, involving lights, sounds, and movement, increases the interactivity of the learning experience provided by BitDogLab and demonstrates its flexibility. Future research can assess the effectiveness of this methodology in stimulating interdisciplinary learning and the development of programming skills. The application of this methodology in other contexts and with different types of hardware is also a promising area for future investigation. Therefore, the central contribution of this work to the field of science and technology education is the proposal of an iterative programming process that integrates AI into the development process, using the HCD. This approach, which harmoniously unites AI and open hardware in the form of BitDogLab, has been specifically designed to simplify the programming task for both students and educators. In doing so, it not only demystifies a complex subject, such as dealing with complex coding syntax, but also actively encourages the development of critical thinking and problem-solving skills. In this way, BitDogLab can be an empowering tool for STEAM activities, providing an environment where curiosity is rewarded with knowledge and challenges serve as stepping stones to understanding. This confluence of elements propels BitDogLab beyond a mere educational tool, shaping it as a catalyst to inspire lifelong learners in the ever-evolving landscape of STEAM education. Future research will focus on implementing BitDogLab in diverse educational settings, especially in courses for nonengineering majors, to understand how it can be integrated across various curricula, thereby expanding the reach and appeal of STEAM education to a broader student base.

Acknowledgment

The authors would like to express our profound gratitude to the IEEE Electron Device Society (EDS) for the generous grant that enabled us to develop STEM activities. Their appreciation also extends to INCT Namitec (National Institute of Science and Technology on Nano and Microelectronics). This initiative, conducted by CNPq (National Council for Scientific and Technological Development) in Brazil, provided valuable resources and expertise in the development of semiconductor devices that significantly contributed to the success of our project. Additionally, they are immensely grateful to Prof. Jacobus W. Swart for his inestimable support and invaluable contributions to our work.

References

[1] S. Marginson, R. Tytler, B. Freeman, and K. Roberts. "STEM: Country comparisons: International comparisons of science, technology, engineering and mathematics (STEM) education. final report." Jan. 2013. [Online]. Available: https://dro.deakin.edu.au/articles/book/STEM_country_comparisons_international _comparisons_of_science_technology_engineering_and_mathematics_STEM _education_Final _report_/20951737

[2] S. Mejias et al., "The trouble with STEAM and why we use it anyway," *Sci. Educ.*, vol. 105, no. 2, pp. 209–231, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sce.21605

[3] "Brazilian tax aspects on import of services." pnst. Accessed: Jun. 6, 2023. [Online]. Available: https://www.pnst.com.br/en/articles/tax-consulting-and-litigation/brazilian-tax-aspects-import-services/

[4] K. Peppler, M. Sedas, and N. Thompson, "Paper circuits vs. breadboards: Materializing learners' powerful ideas around circuitry and layout design," *J. Sci. Educ. Technol.*, vol. 32, pp. 1–24, May 2023.

[5] R. Heradio et al., "Open-source hardware in education: A systematic mapping study," *IEEE Access*, vol. 6, pp. 72094–72103, 2018.

[6] P. Trivedi, P. Kajgaonkar, A. Kulkarni, N. Kolte, and B. Kanawade, "System model for syntax free coding," in *Proc. Glob. Conf. Adv. Technol. (GCAT)*, 2019, pp. 1–5.

[7] M. A. Togou, C. Lorenzo, G. Cornetta, and G.-M. Muntean, "Assessing the effectiveness of using Fab lab-based learning in schools on K–12 students' attitude toward STEAM," *IEEE Trans. Educ.*, vol. 63, no. 1, pp. 56–62, Feb. 2020.

[8] S. Freeman et al., "Active learning increases student performance in science, engineering, and mathematics," *Proc. Nat. Acad. Sci.*, vol. 111, no. 23, pp. 8410–8415, 2014. [Online]. Available: https://www.pnas.org/content/111/23/8410

[9] "Inspire seus alunos com STEAM." escola4pontozero. Accessed: Jun. 3, 2023. [Online]. Available: https://escola4pontozero.fee.unicamp.br/

[10] "Buy a raspberry pi Pico." Raspberry Pi. Accessed: Jun. 3, 2023. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-pico/

[11] "BitDogLab repository." Github. Accessed: Jun. 3, 2023. [Online]. Available: https://github.com/Fruett/BitDogLab

[12] "Raspberry pi OS." Raspberry Pi. Accessed: Jun. 3, 2023. [Online]. Available: https://www.raspberrypi.com/software/

[13] A. Annamaa, "Thonny: A python IDE for learning programming," in *Proc. 2015 ACM Conf. Innov. Technol. Comput. Sci. Educ.*, 2015, p. 343. [Online]. Available: https://doi.org/10.1145/2729094.2754849

[14] "MicroPython." Accessed: Jun. 3, 2023. [Online]. Available: https://micropython.org/

[15] OpenAI et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.

[16] "Getting started with raspberry pi Pico." Raspberry Pi. Accessed: Jun. 3, 2023. [Online]. Available: https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico/2

[17] "Conway's game of life." Wikipedia. Accessed: Mar. 27, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Conway

Fabiano Fruett (Member, IEEE) received the degree in electrical engineering from the Universidade Estadual Paulista—UNESP-Ilha Solteira, Ilha Solteira, Brazil, in 1994, the master's degree in microelectronics from the Universidade Estadual de Campinas—UNICAMP, Campinas, Brazil, in 1997, and the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, in September 2001, where the primary findings of his doctoral research were published in over a dozen scientific articles and a book authored by him.

In 1997, he was a Researcher of Microelectronic Sensors with the Delft University of Technology, Delft, The Netherlands. Since 2002, he has been a Faculty Member with the School of Electrical and Computer Engineering (FEEC), UNICAMP. He is currently an Associate Professor III with the renamed Department of Electronics and Biomedical Engineering, FEEC. His research interests are in the areas of electronic instrumentation, Internet of Things, and open-source STEAM learning tools.

Fernanda Pereira Barbosa received the M.Sc. degree in chemistry from the Universidade Federal de São Carlos, São Carlos, Brazil, in 2005.

She is the Supporter of the Escola 4.0 Project. She is also the Curator of Social Networks for the Project Escola 4.0.

Samuel Cardoso Zampolli Fraga received the B.Sc. degree in electrical engineering from the State University of Campinas—Unicamp, Campinas, Brazil, in 2024.

Pedro Ivo Aragão Guimarães received the B.Sc. degree in electronic engineering from the Federal University of Campina Grande—UFCG, Campina Grande, Brazil, in 2021. He is currently pursuing the M.Sc. degree in electronic engineering with the Universidade Estadual de Campinas—UNICAMP, Campinas, Brazil.