Alternatives to replace LEGO EV3: using the next-generation Mindstorms and Stem:Bit kits in education

GAÁL Bence, SOLYMOS Dóra

Abstract. Anyone who has dealt with robotics education has evidently encountered the LEGO Mindstorms EV3 model, which has played a crucial role in this field since 2013. However, the production of these robots has officially ceased, and their support will end in the near future. In our article, in addition to introducing the new generation LEGO Mindstorms Robot Inventor, we also aim to present a cheaper alternative in the form of Stem:Bit, which we recommend as a replacement for the EV3. A vital aspect of comparing these devices is to examine what functions they provide compared to the EV3, and what advantages and disadvantages each has in robotics education.

Keywords: robotics, programming, Mindstorms EV3, Robot Inventor, Stem:Bit

1. Introduction

In September 2020, the new National Core Curriculum (NAT) was introduced in Hungary, which includes the basics of robotics as early as the third grade of primary school. [1] In the lower grades, children typically start learning robotics with popular floor robots. However, upon advancing to the upper grades, they can get to know more severe robots with greater knowledge and more programming possibilities. This group includes the LEGO Mindstorms EV3 developed by LEGO Education, which was phased out of the market coinciding with the introduction of the new National Core Curriculum. There are numerous articles about the educational role and usability of the device [2][3][4]; however, if someone wants to acquire it now, they will unfortunately have to look for an alternative, as it is no longer available in stores.

We aim to provide assistance by introducing two devices. The first device is the successor to the EV3, officially named the LEGO Mindstorms Robot Inventor, known in Hungarian as the "Robot Feltaláló" but seldom referred to as the EV4. The other device is the Stem:Bit, a robot controlled by a BBC Micro:bit, containing LEGO-compatible elements.

In our article, we present the changes and the modifications needed for the use of tasks. Based on our experience, both devices can be used in education, and the examined tasks can be solved with minor modifications, making the materials prepared for the previous version usable.

2. Introduction of the LEGO Mindstorms Robot Inventor (51515)

The latest LEGO Mindstorms device, the Robot Inventor (51515), was launched in 2020, replacing its predecessor, the EV3. The new device arrived in a cardboard box, as seen in Figure 1. The dimensions of the box are 47,5 cm wide, 37 cm high, and 6,7 cm thick. Upon opening, a large number of LEGO pieces, a little sticker sheet, and a booklet with the part list are visible. Opening the box has been simplified compared to the EV3 home edition (31313), as the parts can be nested together like a shoebox, though the educational version (45544) of the EV3 had a better solution with an internal organizer, which is missing from this version.

Recommended age group	from age 10
Type of programming language	block-based/Python
Number of building elements	949
Number of motors	4
Type of sensors	distance sensor, light and color sensor, compass, gyroscope
Other accessories	-
Current price	359,99 €/package[1]

Table 1: General overview of the LEGO Mindstorms Robot Inventor 51515

Figure 1: The box of the LEGO Mindstorms Robot Inventor (51515) set[2].

The box contains a total of 949 pieces, which, when compared, is nearly 1,5 times the number of pieces in the home version and twice as many as in the educational version. Among these pieces, a significant portion of the elements found in the EV3 sets are present. A notable addition, however, is the inclusion of several larger-sized elements in the box, among them a board-like element that is suitable for attaching sensors, motors, and other elements. (The best comparison for this new LEGO board would be with the BBC Micro:bit test panels.) The metal ball found in the EV3 set has been removed from this kit, which could potentially causing issues for vehicle-like builds. The appearance of the elements has been modernized and made more colorful, making it more akin to the LEGO Education SPIKE™ Prime (45678) set.

2.1. Models

The developers designed five models for the kit, whose building instructions can be found in the application used for programming. (We will write more about this later.) The models can also be seen on the left side of Figure 1. Their names, from left to right, are: M.V.P, GELO, BLAST,

[1] Source: https://www.brickeconomy.com/set/51515-1/lego-mindstorms-robot-inventor

[2] Source: https://www.lego.com/hu-hu/product/robot-inventor-51515

CHARLIE and TRICKY. Most of the models come with the option for modifications, which also includes building instructions. For example, the base model of TRICKY can be rebuilt to play basketball (with a lifting arm to move the ball), play soccer (with leg-like arms to kick a ball), go bowling (shooting a ball by spinning it), or draw (with a raisable pen stand that can be attached).

To evaluate the device, we tried out several models, but the TRICKY configuration, as suggested by the developers, seemed the most practical for educational use. (Figure 2) We slightly altered the model, choosing not to add the decorative elements to facilitate potential future modifications, which proved to be a wise decision in the long run. According to the instructions, only the distance sensor is initially incorporated into the base model. However, we wanted to test as many sensors as possible, so we attached the color sensor and then the pressure sensor to the back of the robot. (Figure 3)

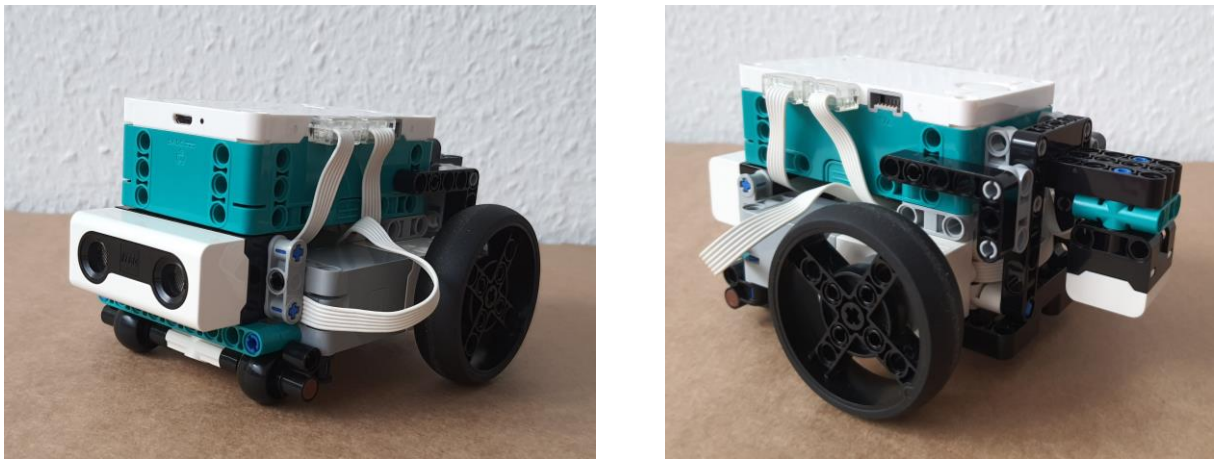

Figure 2: The model named TRICKY is shown in the image, equipped with an additional color sensor.[3]

2.2. Motors and sensors

There has been an increase in the quantity of motors; instead of the two large and one medium that were previously provided, we have four large motors of the same size. Additionally, they became much smaller and have more connecting surfaces than the previous Mindstorms motors. This appearance and functionality were previously encountered in the SPIKE set but in a different color. The motor system has also been changed; occasionally, the program alerts users to the need for an update due to a new version that has appeared.

The appearance of the sensors has undergone significant changes; they now resemble the elements found in the SPIKE set, just like the motors. Within the package, there's an ultrasonic distance sensor like the one found in the educational version of the EV3, but it has a reduced operational range, effectively detecting objects only up to 200 cm away. This change in the sensors marks an improvement in terms of design, aligning them more closely with the newer SPIKE technology. Additionally, an enhancement has been made by including four lamps around the sensor, which can be programmed individually, adding more versatility and interaction possibilities to the device's use. Based on our experiences, an improvement in the device's accuracy has been noted, indicating a positive advancement in sensor technology within this kit.

[3] Source: Dóra Solymos

The light and color sensor can distinguish eight colors and the neutral color. The previous list was adjusted by removing brown and adding pink, as well as enhancing the set with the set's specific shade of cyan blue. Therefore, the device can detect the following colors: black, pink, blue, cyan blue, green, lemon yellow, red, white, and neutral. The reflection of light known from the EV3 is also usable here, and both its versions (reflected light and ambient light) are available on the programming interface. Moreover, the sensor can measure the extent to which the color it sees contains red, blue, or green. Thus, we could query an RGB code of a color, which could be used for tasks involving color sensing. This way, we could print a track or color tracking stripes that the device is more likely to detect than its previous version. However, the color sensor remains a weak point of the device as it does not always recognize green, blue, and the new cyan blue from its own LEGO elements, which were also problematic colors in the EV3. At certain angles and heights, the chances of successfully detecting these colors increase, yet, for reliability, it's advisable to stick with black, white, red, and yellow colors.

The compass and gyroscope have been integrated into the controller named Hub. This integration allows for functionalities such as displaying a smiley face on the Hub with a firm touch or setting up different functions for various directional movements, similar to those found with the BBC Micro:bit.

Compared to the EV3, there might be a sense of something missing regarding sensors since the set lacks a pressure sensor. However, there is one available in the SPIKE set, which is compatible with this set, meaning if we can acquire it, we can use it easily. If we wish to substitute something for it, we can use our creativity with the existing pieces to build something that could serve as an acceptable pressure sensor. Online,[4] most people create pressure sensors using motors; we were inspired by this and created the sensor shown in Figure 3, which was placed above the color sensor on the test vehicle. This setup is operational and can substitute for a pressure sensor, but its accuracy is limited.

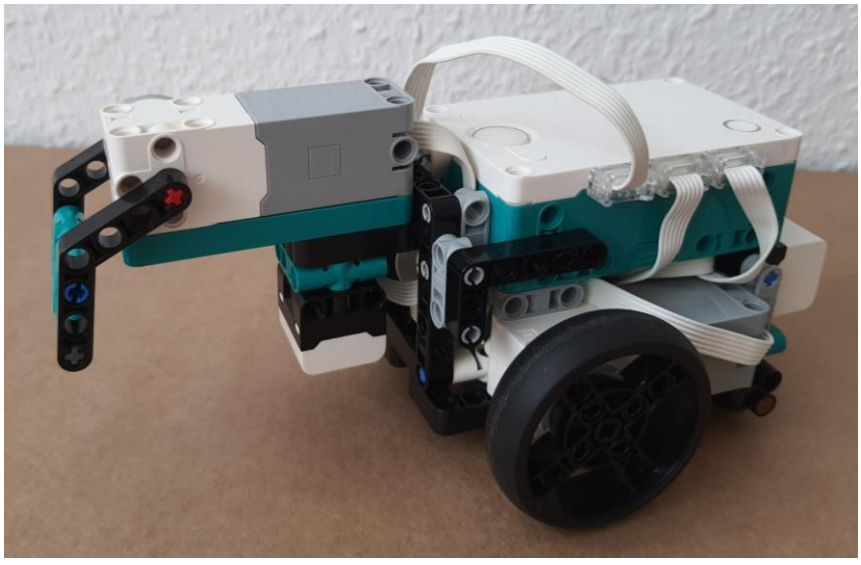

Figure 3: The test car variant equipped with a pressure sensor[5]

[4] https://www.eurobricks.com/forum/index.php?/forums/topic/183443-fix-no-force-sensortouch-sensor-in-51515-robot-inventor/

[5] Source: Dóra Solymos

2.3. The Hub

A significant change can be observed in its appearance compared to its predecessor. The disappearance of the display found on the EV3 means that many familiar functions are either inaccessible or must be accessed elsewhere. The Hub features a large power button, two smaller programmable buttons, and a Bluetooth button. The LED light surrounding the power button is programmable, similar to its predecessor, but the colors have changed, and their number has increased. The display, as a drawing surface, has been simplified in some ways and has become programmable similar to the LEDs on the Micro:bit, meaning we can control them by coordinates and adjust their brightness. Additionally, it can display animations and texts longer than the screen. It is also capable of playing sounds.

The controller features six ports that are evenly distributed along the length of the device, identified by letters A to F. There's no distinction between the ports; both motors and sensors can be connected to any of them. Additionally, there's a micro USB connection available on the Hub. This allows for connection to a computer using the included micro USB to USB cable, which is suitable not only for charging but also for data transfer.

2.4. Introduction of the programming interface

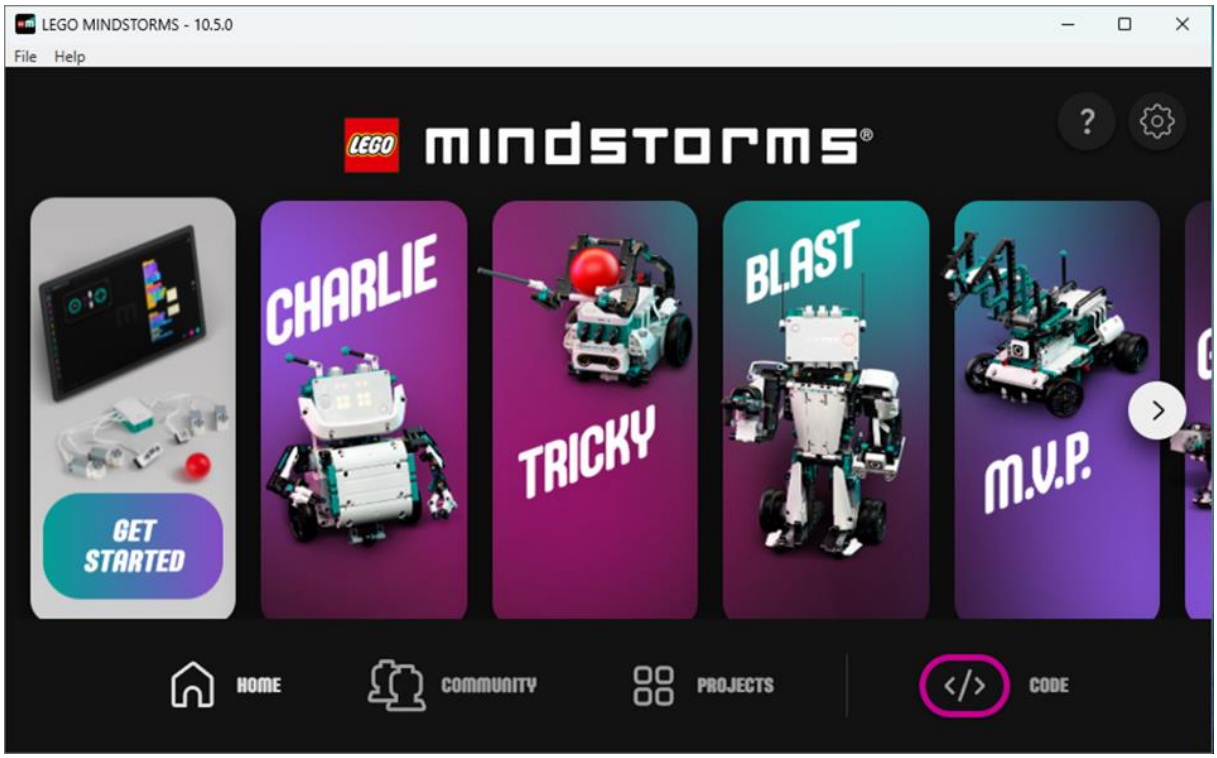

Figure 4: The Interface of the LEGO® MINDSTORMS® Robot Inventor app [6].

To program the Robot Inventor, we need to download an application available on the Microsoft Store, Play Store, and App Store. The application is called LEGO® MINDSTORMS® "Feltaláló" in Hungarian and LEGO® MINDSTORMS® Robot Inventor in English. We can choose from

[6] Source: Dóra Solymos

several languages, but the app will initially start in the default language of the installed device. Upon opening the app, we will see the interface shown in Figure 4, which is the homepage of the app. Choosing the *Get started* option provides assistance with using the device, starting from the basics. Clicking on the pictures of the models grants access to the building instructions, and after building, we receive a code to test out our constructed robot. Clicking on the *Community* tab takes us to building instructions created by users. A drawback of the interface is that we can only view any instructions if we download all of them. The third option in the main menu is *Projects*, where we can find our previous codes, including both block-based and Python-written codes. The last option on the homepage is *Code*, which takes us to the actual programming interface.

In the official software for the EV3, we could program using icon-based blocks, which had to be placed horizontally next to each other. Later, programming was also possible in MicroPython. Nevertheless, we could program using text-based blocks on the Makecode platform, where the blocks had to be inserted into each other vertically, and a simulator was also available. On the new Mindstorms block-based interface, we can use text-based blocks that need to be inserted into each other vertically. The environment itself is based on Scratch, so the appearance and some blocks may be familiar. Additionally, the device can also be programmed in Python using MicroPython. For this, a documentation has been created, which is available in English here: https://lego.github.io/MINDSTORMS-Robot-Inventor-hub-API/index.html

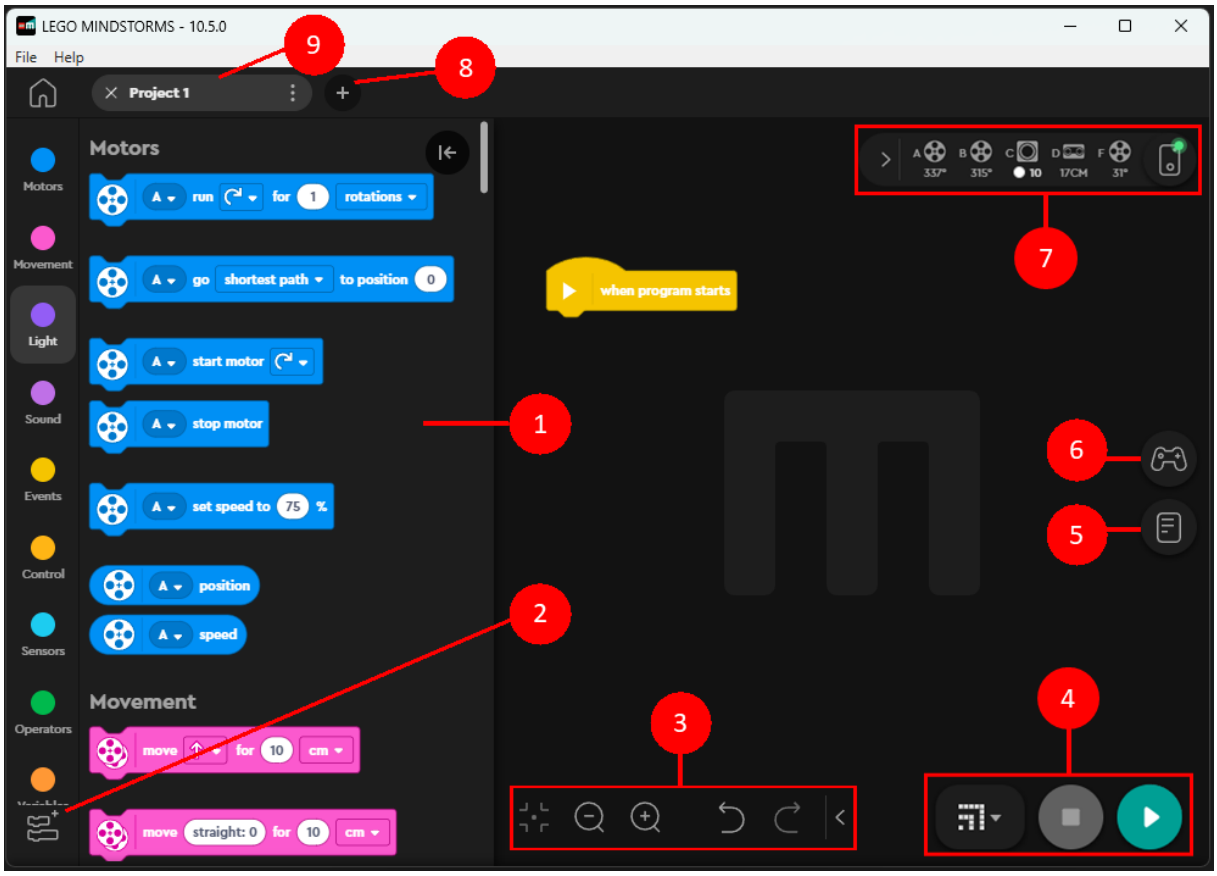

Figure 5: The homepage of the interface used for block programming.[7]

[7] Source: Dóra Solymos

The programming interface (illustrated in Figure 5) is streamlined, featuring only the buttons necessary for programming. However, some of these can be hidden if they are not needed. The interface includes the following parts:

1. Block palette
2. Adding extensions
3. Zoom, undo, and redo
4. Play modes (download/streaming), stop and play
5. Help and monitor
6. Remote control
7. Hub Connection
8. New project
9. Current project

The interface allows hiding the block palette (1), the zoom, undo, and redo functions (3), and the Hub connection (7).

Clicking on the "Add Extensions" button pops up a window where we can choose from a variety of extensions, including model blocks, multiple motors, multiple movements, music, weather, LEGO powered up, and additional sensors. Furthermore, although in experimental status, the following extensions can be tested: machine learning, XBOX ONE controller, DUALSHOCK 4 controller, and Hub-Hub communication.

On Figure 5, marked with the number 4, we have the option to download the program or run it via Bluetooth connection. The second mode can be accessed by selecting Streaming mode, which allows us to run our program immediately without downloading. The download mode has changed from the previous version in that the number of downloadable programs has been maximized to 19, and before downloading, we can select which slot we want to download the program to. After downloading, the Hub immediately runs the program; however, if we want to start it from the device, we need to find its number on the Hub using the left-right buttons and start it with the power button.
In the EV3, we could assign names to our programs, and we had to locate the file we wanted from all the downloaded files based on that name. Managing files on the EV3 was cumbersome, as files had to be deleted individually. However, this process has been greatly simplified here. The new application makes it simple to remove unnecessary files with a single button click.
The two modes available on the Robot Inventor can be distinguished on the device by the different patterns in which the lights activate: during downloading, the sequence number of the download location is indicated, while in stream mode, a Wi-Fi icon is displayed.
The interface includes a mode selection button alongside stop and play buttons (shown in Figure 5, button group number 4). These may not seem so important when creating the first programs; however, after a few more attempts, they become indispensable. Due to the fact that there is no button on the device to stop the currently running program. (On the EV3, we had a grey button for this function that was situated in the bottom left corner of the screen.) After executing the blocks, the program does not exit as its predecessor did but continues running even if there is nothing left to do. At this point, by clicking the play button, we can restart the program, while the stop button allows us to end the execution. Another way to exit from the program is by inserting a "stop and exit program" block (from the orange Control Blocks category) at the end of the code. Then, there is no need to use the stop button to finish running the program.

With the controller-shaped button labelled number 6 in Figure 5, we can bring up a small panel (see Figure 6) where we can place various buttons related to control, sliders, and sensors. Then, we

can format, name and program these elements. During execution, we can use this interface as a remote control. This feature allows us to create a program similar to the application available for the EV3, which enabled the device to be controlled from a phone or tablet.

The help feature on the interface serves as a sort of knowledge base, providing not just descriptions of the blocks but also videos, guides, and snippets of code for using the various functions. However, the help feature is not always understandable on the Hungarian interface because the translation is sometimes meaningless or does not convey the same as its English version.

By clicking on the Hub Connection button (labelled number 7 on Figure 5), we can connect to the device. After connection, we can change the device's name, manage programs on the Hub, modify the modes of motors and sensors, and update the motors. Additionally, we can check the current values of the built-in sensors. During programming, next to the Hub icon, a small bar shows which device is connected to which port, and it even displays the current value of the sensors. (For motors, it can show the current rotation angle or speed.)

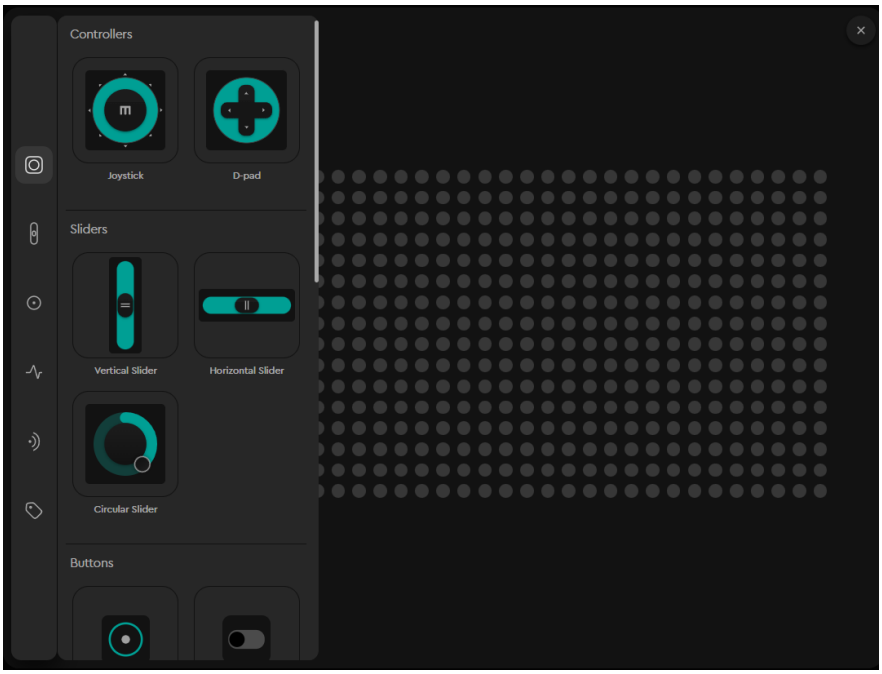

Figure 6: The interface for creating a remote control.[8]

2.4.1. Introduction of the block groups

The interface itself is user-friendly, especially if we already have experience in a block-based environment. However, compared to the EV3, some tasks must be approached differently. For instance, settings that were previously consolidated into a single block for motors, such as names, speed, and control mode (degree, rotation, seconds), are now distributed across separate blocks. In the *Motors* (blue) category, each block allows us to select the desired motor, but we must specify the speed, direction, and duration of movement in different blocks. The mode previously known as "rotation" has been translated to "forgás" in Hungarian, which may be confusing because it is

[8] Source: Dóra Solymos

situated directly above the degree (in Hungarian: "fok") mode, potentially leading to mix-ups among children.

If we want to set both the duration and speed of motor movement within a single block, we need to activate the *More Motors* extension from the available extensions. At the very bottom of the block groups, we will find the blocks suitable for this purpose. These include blocks that let us set the motor's speed or power output at startup. Both blocks look the same, differing only in the last expression, but there aren't any differences in their operation. This phenomenon is also present in the *Multiple Movements* extension.

Among the *Movement* blocks, we find blocks similar to those previously used for operating two motors simultaneously. However, here we only see blocks similar to the *Move Steering* block known from the EV3, which works with direction determination. In order to start our two motors at various speeds, we need to turn on the *More Movement* extension. We can discover some rather complex blocks in this section since one block can change two to three variables.

In the *Light* (purple) blocks section, we find blocks that use the Hub's "screen". A new feature is the ability to play animations, for which there is a library available containing animations created for the models. However, we can also create our own animations. Additionally, we can display individual images on the screen, but there isn't a library available for this. Furthermore, texts are not constrained by the screen size and are shown on the device like an advertising tape. This function resembles the text display feature known from the BBC Micro:bit, as does the display of points by coordinates. We can achieve this with the *set pixel at 1, 1 to 100%* block. A new block for modifying the lights surrounding the distance sensor and a block for programming the LEDs surrounding the power button is also included in this group. Moreover, two blocks for setting the device's orientation are also located here.

In the *Sound* block group, we encounter blocks familiar from the EV3. However, the capability to play sounds on both the programming device and the Hub has been added as a new feature. Consequently, the playable sounds are divided into two groups. The *Sounds on the Hub* section contains sounds that can be played on the Hub, while the "Library" section includes those that can be played on the programming device. A new functionality allows us to edit sounds from the library, and making sound recordings has also been simplified. The names of the blocks for playing piano sounds have changed; they start with *play beep*. In the EV3, this was done by selecting the *Play Note* mode.

In the *Events* group, we can find trigger blocks that allow us to start our program based on the response of a sensor, movement of the Hub, or a button press. Additionally, we find blocks known from Scratch that initiate program execution.

In the *Sensors* group of blocks, you can find blocks necessary for color detection as well as for detecting reflected light. However, in this section, we only have access to blocks that work in the previously known reflected light mode. If we want to use the ambient light mode, activating the "More Sensors" group allows us to do so with the *set mode to ambient*, *ambient light < 50%* and *ambient light* blocks. Among the sensors, new functionalities such as gesture detection and orientation have been introduced, functioning similarly to the Micro:bit. In the *More Sensors* section, we find a block for the raw reflected light value for red, blue, and green colors. (The color value is a number between 0 and 255.) Additionally, this section includes blocks for the pressure sensor found in the Spike set.

2.4.2. What tasks can be solved with the Robot Inventor?

For the Robot Inventor, a few English-language publications have been released, however there are currently no Hungarian instructional resources. Therefore, we selected some tasks from the ones listed in [5] that were originally designed for the EV3 and attempted to solve them within the Robot Inventor block-based environment. During our testing, we were able to solve a significant portion of the tasks with this tool, however in certain cases, we could only achieve this by rephrasing the tasks.

While using the colored LEDs, we encountered an issue where the LEDs could not be turned off, nor did the block have a flashing function, demanding modifications to the related tasks. We resolved the issue of turning off the LEDs by setting them to black color and achieved flashing through a combination of loop and wait blocks.

Due to the absence of a touch sensor, we had to solve related tasks using a motor. This substitution required a complete rephrasing of the tasks and the omission of some. Considering that testing the pressed, released, and bumped functions became meaningless since we don't have a button but an arm that is very sensitive to minor changes, however, to rotate it, we have to turn it by hand. Our tests showed that if the robot approaches an object too quickly, its pressure-sensing motor does not turn, and the device tips over. If the robot moves slowly, it can be used as a touch sensor, but even minimal movement is enough for detection. This can be seen in the code shown in Figure 7.

Figure 7: The code for the following task is shown in the image: when the motor functioning as a touch sensor is "pressed," a smiling face appears on the display[9].

The tasks related to creating parallel programs in the teaching material ([5] page 27, tasks 1-4) can also be solved in this environment, but they will not be parallel programs, only single-threaded ones. Additionally, the task related to drawing a house ([5] page 45, task 5) cannot be solved or can only be solved with significant simplification of the task due to the 5x5 pixel size of the display severely limiting the images that can be displayed. Moreover, the line-following task ([5] page 33, task 3) can only be solved by providing a mathematical formula due to the inaccuracy of the color

[9] Source: Dóra Solymos

and light sensors. More information about the formula and how it works can be found in this video: https://youtu.be/Z8Cv60f75LI.

The names of colors are not included among the sounds. However, since creating sound recordings has become quite simple, and the device can even be programmed from a phone, children can record the names of colors in Hungarian directly in the app using their phones. These recordings can then be used to solve the task. This way, they can also learn a bit about sound editing, and test the operation of the "Hangos színek" task ([5] page 36, task 3) in their language, which could be an advantage for children learning German as their first foreign language.

For tasks not specifically mentioned, we encountered no significant problems; we were able to solve them using the blocks available to us.

3. Introduction of the Stem:Bit [6][7]

Recommended age group	from age 10
Type of programming language:	block-based/python/ Javascript/scratch
Number of building blocks	302
Number of motors	2
Type of sensors	obstacle sensor, line tracker, ultrasound connection option
Other accessories	vibrating motor, 2 x 10mm RGB LEDs, 3x small RGB LEDs, infrared receiver unit, remote control
Current price (micro:bit included)	117,95€/package

Table 2: General description of Stem:Bit[10]

Figure 8: The Stem:Bit package[10]

The Stem:Bit package is divided into two main parts, one is the building blocks and the other is the expansion board itself. To this expansion board we can connect the micro:bit, which will

[10] Source: https://shop.sb-components.co.uk/products/stem-bit-the-programmable-blocks-kit-for-micro-bit?variant=31064155455571

control our robot. Other accessories and sensors shown in the table (Table 2) are all integrated into this board, which may limit creativity during the building process. The bricks in this pack are fully compatible with LEGO products. In terms of learning materials, the balance is tilted in EV3's favor, as unfortunately there is no developed learning material available for Stem:Bit, and while the EV3 package only includes five official building options compared to the nine models of Stem:Bit, for the EV3 we could find certified user guides.

Micro:bit control allows Stem:Bit to be a device that can be used across a wide age spectrum. For this reason, the tool also has all the advantages of micro:bit, such as cost-effectiveness, widespread application and success, and a sufficiently high volume of extensibility. [8] There are many descriptions and articles about the micro:bit available, so we have not specifically addressed it as a control unit in this article.

3.1. Models

Nine official models from the package can be built, and the manual provides instructions for these. We did not encounter any difficulties in building these models. It is important to note that there are models that are not capable of movement. The model most capable of solving EV3 tasks is the IR Car (Figure 9[10]). When it comes to line tracking and object avoidance, the two robots are similar, but it's important to consider that we can't build a robot with one Stem:Bit that moves and can move objects at the same time, so this can be a big disadvantage. Both robots perform similarly to line tracking when it comes to detecting and avoiding obstacles. Due to the rules of the WRO competition, this package is not suitable for racing, as it can only be participated with LEGO robots.

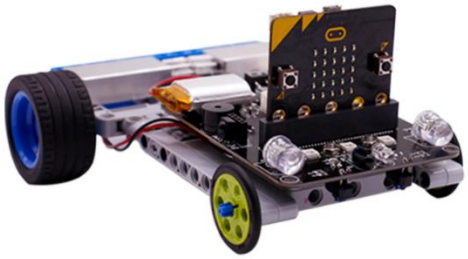

Figure 9: Model of the IR Car

However, we also have the opportunity to make two tracked vehicles, which can be a good basis for making field measuring vehicles, which we can also operate with remote control. The complex sensor package provided by the micro:bit provides an opportunity to take the tool into account more seriously in science education in addition to digital culture classes.

In addition to the tools mentioned above, the package includes a crane, a slingshot, a tuk-tuk cart, a six-legged robot that can climb off-road, a so-called robot car (not suitable for line tracking), and a device for grabbing objects.

3.2. Motors and sensors

In the Stem:Bit package, we get a total of two motors. Not only is the quality of the package inferior to the EV3 in terms of the number of units, but also in terms of the type of motors, as here we do not get servo motors. For this reason, it can be difficult to implement turtle graphics on the floor, as we can only specify the time and speed of operation of the motors, the latter of which will be greatly influenced by the battery charge. However, in terms of extendability, we get three places for servo motors, so we can connect a maximum of 5 motors to the extension board.

When it comes to sensors, we can talk about an extremely diverse device. One of the reasons for this is that we can also use the set of sensors provided by the micro:bit in the course of our work, so we get: accelerometer, compass, light sensor, thermometer, microphone and touch sensor (the latter two only for v2 micro:bit). In addition, the board is equipped with an infrared obstacle sensor and a line tracking sensor, and it is also possible to connect an ultrasound sensor. The obstacle avoider and line tracker made relatively few mistakes during the tests, it can be said that the sensors work properly.

Due to the lack of a color sensor, we will also not be able to perform all tasks performed by the EV3, such as the ability to distinguish colors. In addition, there is an infrared receiver on the device, which allows us to control the robot using the remote control that comes with the package. In terms of extensibility, we get a much more complex device if we look at sensors. The package is inferior to LEGO products in terms of the number of bricks, but due to compatibility, expansion is possible.

What else is worth mentioning and can be useful for children in such a session is the output peripherals. Here we get a vibrating motor, as well as 3 small RGB LEDs that we can program, as well as two more 10mm LEDs located on the front of the board, as if it were the headlight of a car. If we are using micro:bit v2, a speaker will be added to the output devices.

2.7. Introduction of the programming interface

Block-based language can be perfect for using the tool at any age to introduce programming or motivate students more. Block programming appears in Hungary already in the third grade [1], but there is also international research that proves that block programming has many advantages even in high school [9], as it is an easy-to-read, simple programming solution, so it provides a great opportunity to make people love programming and increase motivation.

The tool and micro:bit can be a good option for introducing python in addition to the block-based programming approach. An important consideration when choosing a programming language may be the simplicity of the language and support for object-oriented programming. Based on these aspects, the Python language can be an ideal choice, especially for novice programmers. [10] One of the main advantages of language is the obligation to code readably, beautifully, which allows students to structure their code properly and create code that is easy to maintain. In addition, the Python language is also very popular in the industry and is used by many large companies such as Google, Instagram, and Spotify. Due to the widespread use of Python, learning Python can be a good choice when starting a programming career, especially nowadays.

The current role of Python in the world is well reflected in the fact that, according to the latest TIOBE Index developer survey, Python is the most popular programming language in the world. [11] According to Stack Overflow's 2023 report, for those learning to code, Python rank third among languages chosen for this purpose, after HTML/CSS and JavaScript. Participants who

learned its programming were more likely to indicate that they were using Python compared to the developers' responses (Python: 56% vs. 45%, C++: 31% vs. 20% and C#: 20% vs. 29%). In addition to the increase in popularity, there is also evidence that students are more willing and easier to learn Python than programming languages they have already learned, and based on student feedback, they also found Python more fun. [12] Python is therefore not only perfect as a starting language, but also as the basis for many new, emerging workspaces for students. Such areas include machine learning, data analysis and manipulation, forecasting and the development of artificial intelligence, which are priority and extremely developing areas of today's digital world. For comparison, however, in this article we will only dwell on the block interface in more depth.

Due to the spread of the micro:bit, instead of presenting the official interface in detail, we would like to focus on the new blocks related to the given device and their integration on the Makecode interface.

The blocks responsible for managing the board are not only available for use, but we can also edit them, since the code and resources for the Stem:Bit are available in a GitHub repository[11]. The names of the blocks are clear and with minimal knowledge of English it is easy to understand what we will be able to use each block for.

New blocks can be accessed by clicking on extensions. Here it is necessary to insert the link to the repository, after which we can already select the tool. (Figure 10)

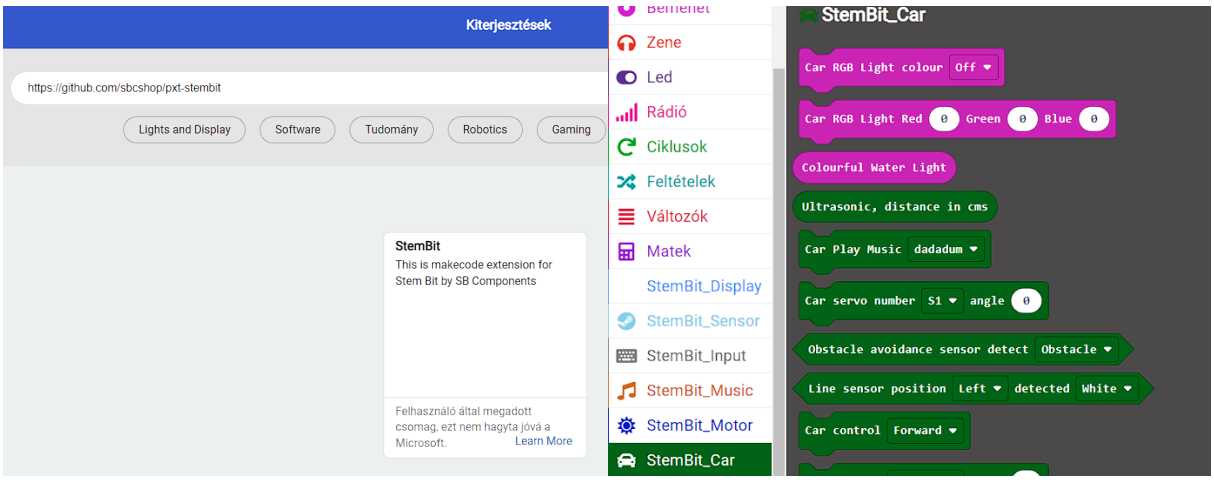

Figure 10: Attaching the extension and the new blocks[12]

After adding the extension, we get the codes that are in English, so for young children this can be difficult and may require a teacher's explanation. The structure of the blocks does not differ from the usual micro:bit environment, so it can be positive that children who have already programmed micro:bits or used block languages will find themselves facing a familiar environment.

A brief description of the interface can be found in the manual that comes with the package, in addition to the nine building instructions mentioned earlier, as well as 1-1 picture of the necessary line of code to make the device work, and at the beginning of the instructions a list of blocks that will be used.

[11] https://github.com/sbcshop/pxt-stembit

[12] Source: Bence Gaál's own picture

We also need the extension for the previously mentioned remote control and, as with Inventor, we have to program the various functions ourselves. This will be discussed in more detail in the next chapter.

3.3.1. Introduction to block groups[13]

In this subchapter we will focus mainly on those blocks that do not require any other accessories and we can control the devices in the package with them. Such block groups for remote control and "car" control blocks.

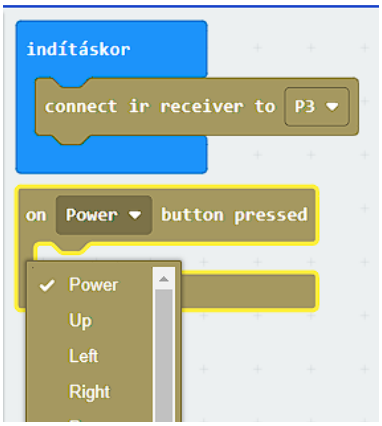

Figure 11: The remote controller related blocks

The first block group we want to introduce is related to the remote control. There are only two new blocks, one of which effectively activates the receiver. We need to know that the receiver unit is connected to P3 pin, so we have to configure it during programming. And in the other block we have the option to specify what happens when pressing which button. The principle of operation is similar to that of buttons on the micro:bit, so when a button is pressed, the code inside the block will run (Figure 11).

The StemBit_Car block group is already made up of many more elements, which can be divided into three subgroups. The first is intended to control the headlights of our imaginary car, which are marked in crimson. With these, we can display preset or created colors of our choice (by rgb code) on the two large leds of the device. The second group includes sensors, as here we can query whether there is an object in front of the obstacle detection sensor, whether the line tracker detects a white/black line or not, and the value of the ultrasonic rangefinder in centimetres. With the third unit, we can control the rotation of the motors, even separately, so we have the opportunity to turning. There is also a block playing pre-built music, and it is also possible to set how many degrees the servo motors should turn, which, like the ultrasonic sensor, must be connected externally to the device (Figure 12).

In *the StemBit_Music* block we can turn on the vibrating motor, while with the *help of the StemBit_Motor* blocks we can adjust the speed of externally connected servo motors and their degree of rotation. The *input* blocks are suitable for the management of externally connected devices, while the *sensor* block group is suitable for the treatment of microphone and ultrasound sensor in addition to the infrared located on the device. With *the help of neopixel* blocks we can make the 3 LEDs on the board work. This and the control of larger LEDs can also be done using display blocks. We do not necessarily recommend using these blocks in the classroom when introducing programming. One of the reasons for this is that these blocks are not necessarily recommended for beginners due to their complexity, and instead of turning the device on and off, we can control the Pins to make different settings.

[13] Image sources: Bence Gaál's own pictures

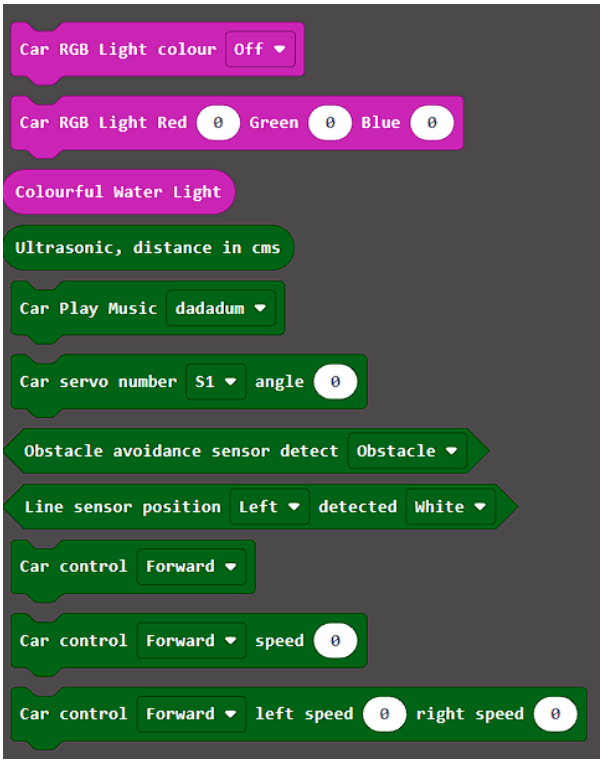

Figure 12: Engine control and sensor blocks

3.3.2. What tasks can we solve with Stem:Bit?

Line tracking tasks can basically be implemented with the tool, even problems related to avoiding objects can be solved with it. Parallel programs can be solved in this environment similarly to Inventor. In the implementation of tasks, the types of sensors and types of motors limit the device, as explained in previous chapters. The size of the display is also mentioned here, so drawing tasks are used to simplify it, since the micro:bit only has a 5x5 led panel.

When used in a competitive environment, it cannot be compared to either the previously discussed Inventor or the EV3 robot, as LEGO products can only be used by participants in WRO Robo Mission and Robo Sports competitions [14].

For its main use, we recommend classroom lessons, where the tool can really stand its ground in making programming more interesting and expanding the micro:bit. Thanks to the block environment, it can be used even in elementary school age, just as we utilized this feature during testing, in fourth grade we used it as a tool to attract attention and strengthen motivation, while in fifth grade it played a role in science subjects in addition to computer science classes as a modeling tool.

It can be used to realize many interesting projects that are not limited to digital culture in class. For example, because of the tracks-based models, it can be effective for modeling a research robot that takes measurements and then transmits data to another device. With the help of micro:bits we can even build vehicles that can be controlled by another micro:bit via the radio connection built-in the micro:bits.

The complexity of the expandability of the device can be beneficial for older age groups, since sensor connections and their control with pins can also strengthen skills related to electronics and

electricity. The disadvantage of this may be that expanding process is more time-consuming than in the case of a modular robot.

4. Summary

Due to the phasing out of the EV3, it is necessary to look for alternatives. Obviously, the structure and price of each robot kit will greatly influence our choice, but we may want to consider other aspects of which one we want to use. Looking at the specifications of the two robot packages discussed above, it could be a good option to replace the EV3, but neither can completely replace it, so consider other factors as well (Table 3).

	Robot Inventor (51515)	**Stem:Bit**	**EV3 Home (31313)**	**EV3 Education (45544)**
Number of motors	4	2	2+1	2+1
Number of motor connection options (ports)	6	2+3 servo	4	4
Number of sensors	2 (built-in) + 2 (modular)	6(built-in)	3 (modular)	5 (modular)
Sensor connectivity	6	beépített+4	4	4
Control unit	Hub	micro:bit	Brick	Brick
Power supply	Li-ion battery	Li-ion battery	6pcs AA battery	Li-ion battery
Total number of items	949	307	601	541
Price	Last known price: ~470 €	101,95 €	Last known price: ~570 €	Last known price: ~745 €

Table 3 Comparison of specifications

We wanted to offer a lesser-known, cheaper alternative to the well-known and established LEGO brand. The use of both devices in education largely depends on the purpose for which they are brought into the classroom. Although LEGO Robot Inventor provides a more expensive option, it is definitely a better supported device, and we get a high-quality device according to the brand. The downside is that the implementations do not necessarily reflect that this device is a successor to the EV3. There is a strong sense that the Spike product line had a big influence on the device during the design process, which will take over fully in the LEGO educational robot assortment in the near future and the manufacturer will focus only on this product.

Despite the Stem:Bit has half as many building blocks, it offers much more freedom in terms of sensory extensibility. Fewer motors and fewer bricks can be a barrier to building more complex robots and, as mentioned earlier, a barrier to creative creation. However, the price of the robot is less than half of the Inventor's price, and this is also a crucial factor in education.

Nevertheless, we believe that we get a particularly good tool for classroom tasks and for introducing and loving programming. They can be used to create interesting, spectacular measuring devices, remote-controlled cars, and can be a good solution for the cost-effective implementation of various interdisciplinary STEM projects.

Overall, both robots are particularly good for educational purposes, but they are targeted in different areas. Inventor is a more widely known tool with greater support and can be used in international competitions, while the cheaper alternative Stem:Bit leverages the versatile micro:bit

to provide a complete experience, while both tools provide the opportunity to cover the robotics education required by the Hungarian National Curriculum.

Bibliography

1. National Framework Curriculum Digital Culture For grades 3-4 https://www.oktatas.hu/pub_bin/dload/kozoktatas/kerettanterv/Digitalis_kultura_A.docx (last visited: 2024.04.05.)

2. Çam, E. & Kıyıcı, M. (2022). The impact of robotics assisted programming education on academic success, problem solving skills and motivation . Journal of Educational Technology and Online Learning , 5 (1) , 47-65 . DOI: 10.31681/jetol.1028825

3. Bradley S. Barker & John Ansorge (2007) Robotics as Means to Increase Achievement Scores in an Informal Learning Environment, Journal of Research on Technology in Education, 39:3, 229-243, DOI: 10.1080/15391523.2007.10782481

4. Solymos Dóra, „LEGO robotok felhasználási lehetőségei az oktatásban," *InfoDidact 2019,* pp. 265-275, 2019.

5. Barbalics Dóra Krisztina, Solymos Dóra, Lego Mindstorms EV3 robotok programozása, ELTE Informatikai Kar, Budapest, 2018. Elérhető: http://tet.inf.elte.hu/tetkucko/wp-content/uploads/2018/12/legomindstorms_szakkorianyag.pdf (last visited: 2024.04.05.)

6. The Stembit official page https://shop.sb-components.co.uk/products/stem-bit-the-programmable-blocks-kit-for-micro-bit?variant=31064155455571 (last visited: 2024.04.05.)

7. Stem:Bit kickstarter page https://www.kickstarter.com/projects/sb-gajendra/stem-bit/description (last visited: 2024.04.05.)

8. Gaál, B. (2022). Robotics-Enhanced Natural Science in Primary Schools. In: Bollin, A., Futschek, G. (eds) Informatics in Schools. A Step Beyond Digital Education. ISSEP 2022. Lecture Notes in Computer Science, vol 13488. Springer, Cham. https://doi.org/10.1007/978-3-031-15851-3_8

9. Weintrop, David & Wilensky, Uri. (2015). To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming. https://doi.org/10.1145/2771839.2771860

10. John M. Zelle (1999): Python as a First Language

11. TIOBE Index for March 2024 - https://www.tiobe.com/tiobe-index/ (last visited: 2024.04.02.)

12. StackOverflow Developer Survey 2023 - https://survey.stackoverflow.co/2023/#most-popular-technologies-language-prof (last visited: 2024.04.02.)

13. Linda Grandell, Mia Peltomäki, Ralph-Johan Back and Tapio Salakoski [2006]: Why Complicate Things? Introducing Programming in High School Using Python

14. General Rules WRO-2024 https://wro-association.org/competition/2024-season/ (last visited: 2024.04.05.)

Authors

GAÁL Bence
Eötvös Loránd University, Faculty of Informatics, Department of Media and Educational Informatics, Hungary,
e-mail: gaalbence@inf.elte.hu

SOLYMOS Dóra
Eötvös Loránd University, Faculty of Informatics, Department of Media and Educational Informatics, Hungary,
e-mail: solymos.dora@inf.elte.hu

About this document

Published in:

CENTRAL-EUROPEAN JOURNAL OF NEW TECHNOLOGIES IN RESEARCH, EDUCATION AND PRACTICE

Volume 6, Number 1. 2024.

ISSN: 2676-9425 (online)

DOI:

10.36427/CEJNTREP.6.1.8168

License