Proposal for the Possibility of Teaching Programming in the Subject of Computer Science using Autodesk Tinkercad

Ondrej Takáč[1], Robert Kanta[2] and Andrea Takáčová[3]
[1] J. Selye University
Bratislavská cesta 3322, 945 01 Komárno, Slovakia
takac.ondrej@gmail.com

[2] Základná škola – Alapiskola Jelenec
Školská 330, 951 73 Jelenec, Slovakia
kanta.robert@gmail.com

[3] Základná škola – Alapiskola Jelenec
Školská 330, 951 73 Jelenec, Slovakia
andrea.ada.takacova@gmail.com

ABSTRACT---- *In today's world, we are witnessing a rapid boom in the implementation of various elements of robotics, intelligent, learning and other systems in our daily lives. The emphasis is on a multidisciplinary approach. These systems build on both hardware and software. Therefore, the question of programming is topical and we meet processors at every step. Our goal is to show within a few lessons the possibility of learning the principles of processor programming in a simple way, without the need for deeper knowledge of electronics or hardware. The use of the Arduino platform appears to be advantageous in this regard. It provides enough power and flexibility for educational purposes, while the Arduino IDE development environment is simple enough. Due to its diversity in terms of robustness and performance, Arduino is also widely used in practice, usually as an extension of existing systems. As an example, we can mention e.g. unmanned UAVs, camera or security systems, control and signaling systems, etc. Therefore, we see the sense of dealing with Arduino in education as well.*

Keywords— Arduino, Programing, Education, Primary school, Tinkercad.

---

1. INTRODUCTION

Arduino is an open-source platform based on easy-to-use hardware and software. Arduino was developed at Ivrea Interaction Design Institute as a simple rapid prototyping tool, mainly for students without deep knowledge of electronics and programming. The Arduino software (IDE) is easy to use for beginners, yet flexible enough to be used by advanced users. [1]

In addition to its educational goals, Arduino also provides a sufficient basis for a variety of superstructure applications and activities, even in non-typical areas such as heritage conservation or photogrammetry. [8] We can also point to the field of robotics, namely the possibility of controlling drive wheel motors. The master can then use its computational power to make further decisions and the implementation of the control of the motors remains with the slave system - the Arduino system. [6] Furthermore, in the context of motor control, we can also mention the speed control of the rotating laser platform, which serves as the source of the rotating laser beam for the fixed sensors. The detection time of the laser beam by the different sensors ultimately gives the possibility of accurately determining the position of the mobile robot. [7] Last but not least, various applications can also be found in the field of unmanned UAVs. E.g. as a base board for signal processing and data storage from air quality sensors, control board for communication-signaling devices, etc. In addition to its use in practice and educational process, Arduino will also find its use in students' interest and leisure activities. [3][9]

One form of learning programming is online programming. Nowadays, it is becoming more and more widespread, especially due to its visual and simulation nature. [11] Taking advantage of the opportunity to learn programming on real-world platforms, e.g. Arduino, contributes significantly to the development of students' competences and thinking skills. [10] These practices and resources also foster a relationship with robotics, engineering science and contribute to fostering cross-curricular relationships among pupils. [2][4]

2. METODOLOGY

Procuring an Arduino platform would be cost prohibitive for many schools. Although, one Arduino board costs about 20 euros, which may not be that much, but in addition to that, you need to purchase a contact array, sensors, control and signaling elements, wiring and power supplies, which can already increase the price to several hundred euros per set. That is why we take advantage of the option offered by Autodesk Tinkercad. Tinkercad is a free web application for 3D design, electronics but also programming and simulation. This one also supports the Arduino board, it has both components and sensors. We will start our whole activity by registering and logging in at *https://www.tinkercad.com*. We will then choose the electronic circuits option and create a new project. The window of the new project is shown in Fig. 1.

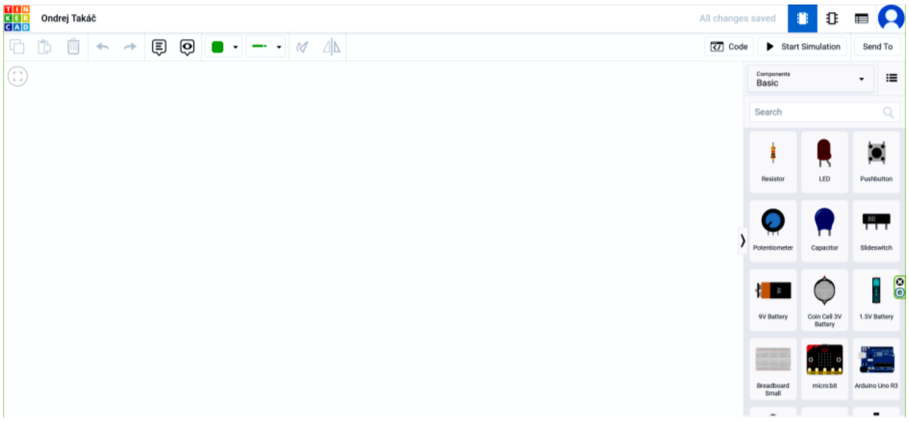

Figure 1: New project window

We are aware that teaching students to code is a task that may take a whole semester and not just a few lessons. Therefore, we will not program in a classical way, but we will use the possibility of block programming, which seems to be very advantageous especially for pupils of upper primary school or even secondary school. In this way we can fully concentrate on promoting algorithmic thinking and learning the principles of programming. An example of programming using blocks can be seen in Figure 2. Due to the user-friendly design of the Autodesk Tinkercad website, without further explanation pupils can get straight into creating projects, which we describe in more detail in the following text. As can be seen, programming in the classic text-based way is still available. If we find its window annoying, we can hide it with the "blocks only" option. We notice that under the colored targets various input, output, flow control or mathematical functions are hidden. Here we can also choose variables or labels. We start the simulation itself by clicking on the "run simulation" button.

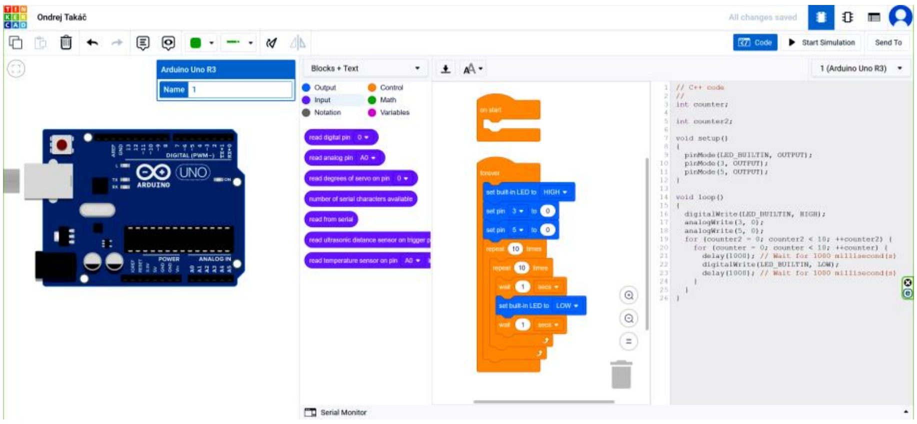

Figure 2: Demonstration of programming using blocks

Here we would like to mention that we will continue to use the popular and widely used in practice affordable Arduino UNO R3 board. The examples mentioned in this article are from the more advanced category, so they are not quite suitable for complete beginners, although, under the right guidance of a teacher, they could be mastered.

3. RESULTS

We will present the results of our work in the form of practical projects. In the introductory part, the teacher will introduce the students to the topic, introduce Arduino, show the UNO R3 physical board, introduce Tinkercad and show the principle of using blocks. Similar to programming a real Arduino, we have to divide the programming into two parts:

- The first part is labeled: *on start* - see. Fig. 2. If necessary, here we set the initialization and the basic parameters valid during the whole operation time, such as the communication speed of the serial port, analog or digital input/output, etc. In Arduino programming language: ***void** setup()*.
- The second part is labeled: *forever* - see. Fig. 2. The microprocessor must run in an infinite loop in order to perform its function. If it did not run, it would very quickly execute all the required instructions and terminate. But when it runs in an infinite cycle, it can continuously monitor a button press or communicate over a serial channel. In Arduino language: ***void** loop()*.

We start with a very simple task, namely turning on the LED for 1 second and then turning it off also for 1 second. The Arduino will perform this action continuously, so we will create a blinking signal.

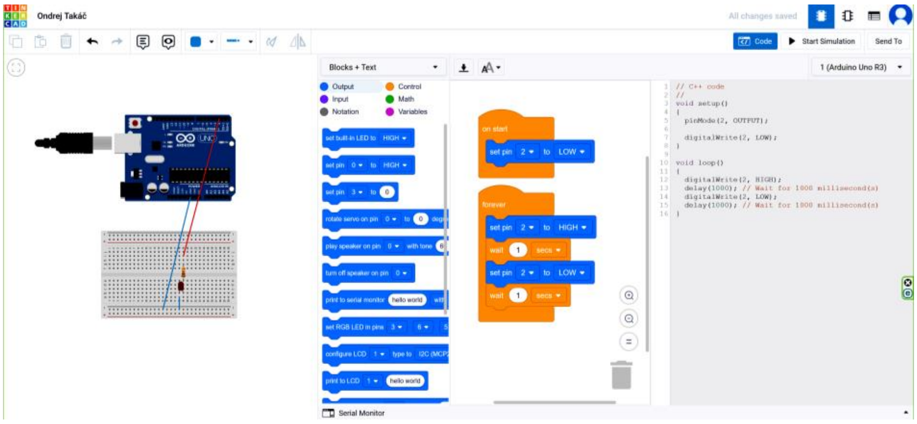

Figure 3: Flashing example solution with LED

We practically wrote our first program in about 20 minutes. But we have to warn the students that LEDs have no polarity, so it is not possible to confuse + and -. To add, we should mention that a resistor is needed to reduce the voltage, because the LED lights at about 2 V but the supply voltage is 5 V. With an assumed current of about 10 mA according to Ohm's law: about 470 Ω.

- A separate task may be to extend the example to a road or railway traffic light, for example.

We will continue the example and use the button. When we press it, the LED will change its state. So if the LED was on, it will go off, and if it was off, it will go on.

- In a separate task, we will task students with completing the example by adding a second button that, when pressed, regardless of the state of the LED, always lights it up and does not change its state.

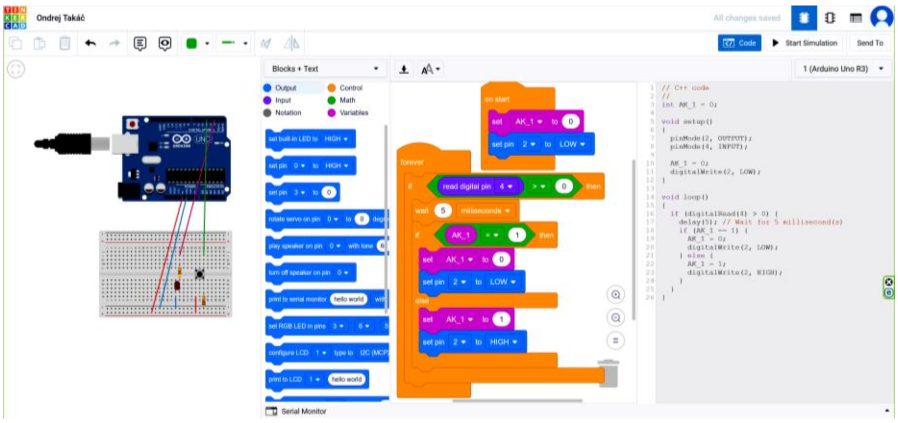

Figure 4: Solution to the example with the button

For the second example, we will focus our attention on the control of the motors and the serial port. We can use the serial port to control the operation or communicate with the Arduino board. It communicates with the computer using a USB cable. When controlling the servomotor, we must point out that for this type we specify the angle of rotation (usually 0° to 180°). Its task is therefore not to rotate, but to maintain the set angle. We start our task by turning the servomotor rotor 45°, wait two seconds and continue turning again until 180°. This is a more complex example, but students will also use a FOR cycle in this example, or two cycles in the second problem.

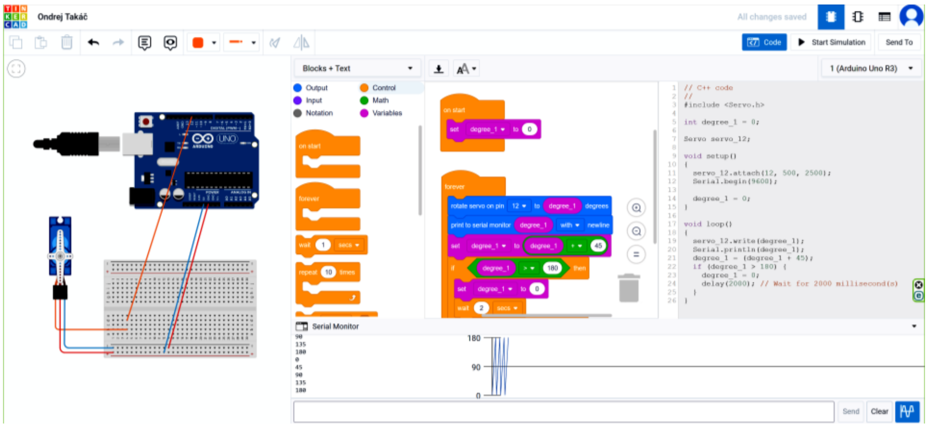

Figure 5: First task with motors - setting the servomotor rotation and using the serial port

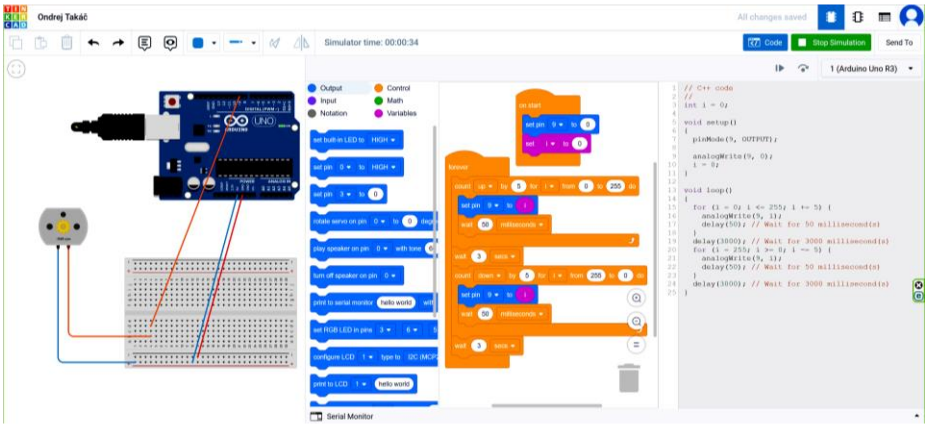

Figure 6: The second task with motors - the principle of DC motor control

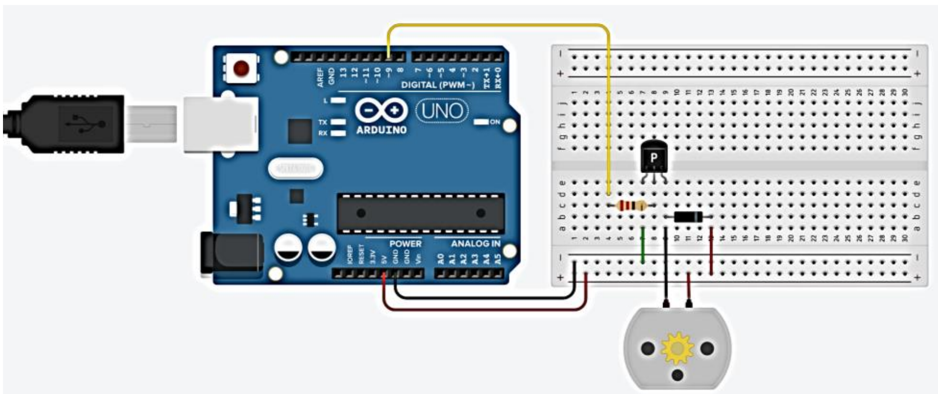

Figure 7: Wiring for higher power engines (pkareeyah, 2020)

In the second problem, we could notice that we control the DC motor by the magnitude of the voltage, which we change in two FOR cycles. The first cycle slowly starts the motor up to max speed, the motor stays in this state for 3 seconds. Then gradually the speed decreases to zero, the motor also stays in this state for 3 seconds. The start-up and deceleration speed is set by the wait in the cycle body, in our case 50 ms. We must note that in case of using a real Arduino board and higher power motors, we must not start the motor directly from the Arduino because it does not have enough current output. We have to use a transistor according to the following picture.

- A separate task can be, for example, to control the rotation of the servomotor using a button and to complement the second task with the possibility of starting and stopping using another button.

For the third example, we will focus on the use of an LCD display. In this task, we increment the value of two variables and print their sum on the LCD display. In the following figure, we can notice that we print Hello World first. This text is also an unwritten world symbol representing a welcome to the study of programming. It is usually the first thing students learn in programming.

- The teacher gives the pupils an independent task at his/her discretion based on what has been learnt. E.g. listing even numbers up to 10, listing on LCD read from serial port, etc.

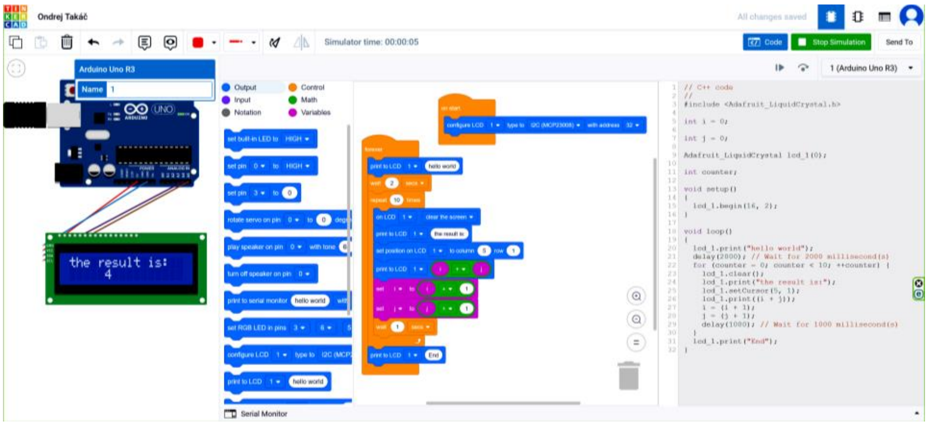

Figure 8: LCD display example solution

In the last fourth example, we use an ultrasonic distance sensor where the distance to the object - obstacle is output on the LCD display. Again, we can use the simulation option to our advantage, where by clicking on the sensor body we can also set the distance to the obstacle. Another advantage is the fact that by using the library we do not have to manually recalculate the distance, but it is directly available for listing.

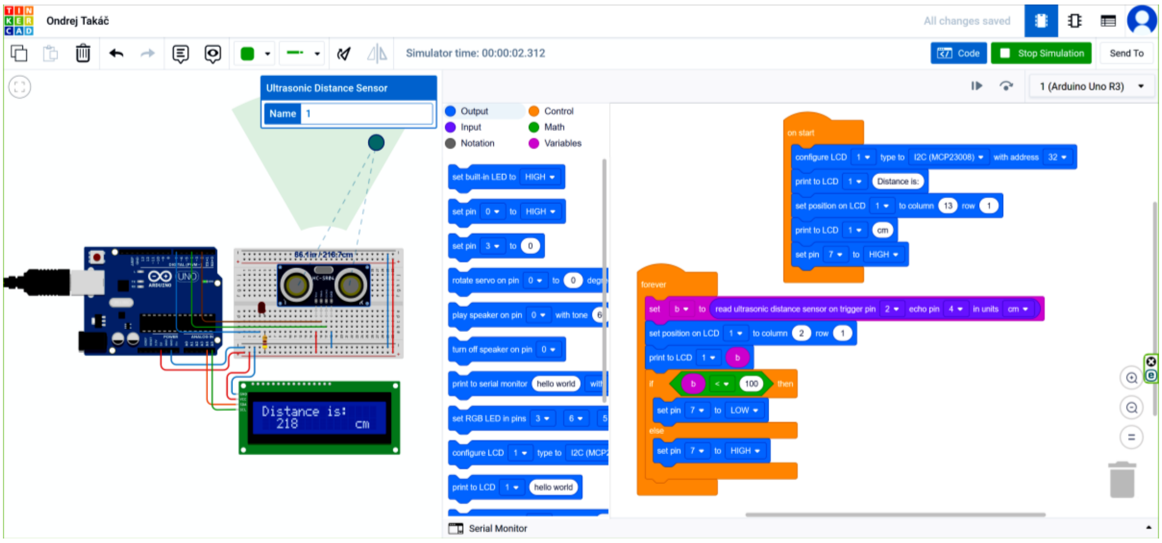

Figure 9: Solution of the example with ultrasonic proximity sensor. Top: Tinkercad website window, bottom: solution detail

- As a separate task, the example can be extended to include a signalling LED if the object or obstacle in front of the sensor is at a distance of less than 50 cm.

4. CONCLUSION

In conclusion, we would like to say that the number of throws did not significantly affect the structure of the course, but we contributed new knowledge and insight. We have used the I2C communication standard when working with the LCD display on. On the previous examples we used another communication standard - serial port, that is UART. We tried to incorporate branching - that is, If, conditions, cycles - quite spontaneously into the examples - especially For, but we also pointed out the importance of the infinite loop and worked with variables. These are all pillars of processor programming. Using the free Tinkercad simulation environment and a well-designed graphics page, learning these pillars is much easier for a wide range of students than with traditional programming. Of course, the generated codes can be downloaded and directly used to program the physical Arduino board. This can be done e.g. using the Arduino IDE environment via USB.

5. REFERENCES

[1] Arduino. (15. 10 2022). *What is Arduino?* Dostupné na Internete: Arduino.cc: https://www.arduino.cc/en/Guide/Introduction

[2] CZAKÓOVÁ, K. 2021. Game-based programming in primary school informatics. In: INTED2021 Proceedings. IATED, 2021. p. 5627-5632.

[3] Mocsi Gergő, M. (2020). *Design of a drone for meteorological data collection.* Komárno: J. Selye University. Bakalárska práca. školiteľ: Ing. Ondrej Takáč, PhD.

[4] Stoffová, V., Takáč, O. (2013). ROBOTICKÉ STAVEBNICE V PRÍPRAVE UČITEĽOV INFORMAČNEJ VÝCHOVY. Trendy ve vzdělávání, 6(1), 315-322.

[5] pkareeyah. (2020). Arduino DC Motor. Tinkercad: https://www.tinkercad.com/things/0RfmRZS79EE-arduino-dc-motor

[6] Takáč, O., Hrubý, D., & Cviklovič, V. (2011). POSSIBILITIES OF NAVIGATION OF MOBILE AGRICULTURAL ROBOTS ON THE PRINCIPLE OF THE GEOMETRICAL OBJETS DETECTIONS. *MANAGEMENT ECONOMIC ENGINEERING IN AGRICULTURE AND RURAL DEVELOPMENT, 11*(1), 206-208p. Volume: 11 Issue: 1. ISSN: 2284-7995. WoS :000422170400044

[7] Takáč, O., Tomášik, L., Hrubý, D., & Palková, Z. (2012). EXACT NAVIGATION OF SMALL AGRICULTURAL MOBILE ROBOTS WITH THE UTILIZATION OF LASER RAY. *MANAGEMENT ECONOMIC ENGINEERING IN AGRICULTURE AND RURAL DEVELOPMENT, 12*(1), 239-241p. Volume: 12 Issue: 1. ISSN: 2284-7995. WoS :000422176100048

[8] Takáč, O., Végh, L. (2021) USAGE OF UAVS IN THE PROTECTION OF CULTURAL HERITAGE IN THE TEACHING OF COMPUTER SCIENCE. In: INTED2021 Proceedings. IATED, 2021. p. 9987-9992.

[9] Tusor, B., Takáč, O., Molnár, A., Gubo, Š., & Várkonyi Kóczi, A. (2020). Shape Recognition in Drone Images Using Simplified Fuzzy Indexing Tables. *2020 IEEE 18TH WORLD SYMPOSIUM ON APPLIED MACHINE INTELLIGENCE AND INFORMATICS (SAMI 2020)* (s. 129-134). Herľany: SAMI 2020. ISBN: 978-1-7281-3149-8. WoS :000589772600025

[10] Udvaros, J., Takáč, O. (2020) Developing Computational Thinking By Microcontrollers. In: ICERI2020 Proceedings. IATED, 2020. p. 6877-6882.

[11] Végh, L., Takáč, O. (2021) Online Games to Introducing Computer Programming to Children. In: INTED2021 Proceedings. IATED, 2021. p. 10007-10015.