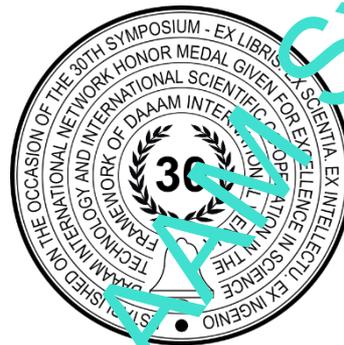


# STEM CLASSROOM: CREATING A PHYTON APPLICATION FOR AN EV3 BRICK ROBOTIC SYSTEM USED TO TRANSPORT 3D PRINTED BOXES

Boris Crnokić, Filip Pehar, Jozo Spajić & Ivan Bagarić



**This Publication has to be referred as:** Crnokić, B[oris]; Pehar, F[ilip]; Spajić, J[ozo]; & Bagarić, I[van]; (2020). STEM Classroom: Creating a Python Application for an EV3 Brick Robotic System Used to Transport 3D Printed Boxes, Proceedings of the 31st DAAAM International Symposium, pp.xxxx-xxxx, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-xx-x, ISSN 1726-9679, Vienna, Austria  
DOI: 10.2507/31st.daaam.proceedings.xxx

## Abstract

Robotics is certainly one of the key tools within the STEM based education of the future. The acquisition of practical knowledge and skills in the field of technical disciplines will certainly be more comprehensive through teaching based on the STEM curriculum. This paper presents a student project within the course "Programming of Robotic Systems" at the Faculty of Science and Education at the University of Mostar. LEGO® MINDSTORMS® EV3 robotics training set was used to create a simple robotic system for transporting of small 3D printed boxes. In order to unlock the full potential and all the possibilities of the Lego Mindstorms set and the created robotic system, EV3 MicroPython firmware was used. A 3D printer "Ultimaker 2+" was used to produce suitable small boxes that are transported within a robotic system. The project proved to be very successful. The robotic transport system had all the functionalities that were planned. Through working on this project, students have greatly improved their practical skills and abilities related to programming, design and production, creativity, working with advanced technical tools, teamwork, etc.

**Keywords:** STEM; EV3 Robotics; Python; 3D Printing; Student Project

## 1. Introduction

Robotics is an indispensable part of modern human society. Robotics is also used in education, from the youngest ages to adults, as an important tool for teaching various technical disciplines. STEM is an educational paradigm based on the principles of teaching based on the synergy of four elementary disciplines: Science, Technology, Engineering and Mathematics. [1] A central aspect of STEM education is problem solving, and the application of robots in such a model of education has shown excellent results. Robotics is becoming an important teaching tool from computer science, computing, mechanical engineering, electrical engineering and other engineering disciplines. [2] In the world of educational robotics there are a lot of manufacturers that offer a wide range of different educational sets of robotics. Perhaps the most famous educational robots are those from the world-renowned toy manufacturer Lego. Their most famous set of educational robotics is "Lego Mindstorms" which is used around the world at different levels of the educational cycle. LEGO® MINDSTORMS® Education EV3 brings project-based learning to the computer science and

STEM, enabling students to improve critical thinking, problem-solving, and collaboration skills. This kit combines LEGO Technic elements, classroom-friendly software and curriculum-aligned lessons to study subjects like engineering, coding and physics with intuitive guides and smart bricks, and places real-life STEM topics right at students fingertips [3], [4] This educational set has found application at various levels of education, and even within the framework of scientific research. Some studies have resulted in the development of a model capable of mimicking human hand using LEGO Mindstorms EV3, which can be used as an education tool for students. [5] The multipurpose use of Lego bricks allows students to design, built and test simple robots easily without requiring special assembly skills.[6] Basically Lego Mindstorms EV3 is a modular set, and its application is possible in various projects that include robot programming, development and testing of control algorithms[7], testing of mobile robot navigation systems, planning the movements of multiple robots together[8], etc. There are great opportunities to learn programming using this kit, it has been proven to be a perfect platform for introducing programming concepts to kids. Lego has produced software support in the form of a graphical programming environment.[9], [10] However, the kit can also be programmed in standard programming languages, such as Python/MicroPython. [11]–[13]

Presented student project was made as part of the activities on the course "Programming of Robotic Systems". As part of the STEM project, students had the task to create a robotic transport system from the LEGO EV3 kit, to construct and 3D print a small transport boxes, and to program the application in the Python programming language.

## 2. Hardware and software used

**LEGO® MINDSTORMS® EV3** kit was used to create a robotic transport system. This is the third generation in the Lego Mindstorms generation (EV3- evolution 3). EV3 has a ARM9 CPU running Linux, USB connector and Micro SD slot (up to 32GB). It is possible to create 17 different robots/projects. EV3 kit uses a program "Lego Mindstorms EV3 Home Edition", which is powered by LabVIEW, to write code using blocks instead of lines, and also it can be programmed on the actual robot and saved. It has support for MicroPython. One EV3 Home set consists of: 1 EV3 programmable brick, 2 Large Motors, 1 Medium Motor, 1 Touch Sensor, 1 Colour Sensor, 1 Infrared Sensor, 1 Remote Control, cables, USB cable, and 585 TECHNIC elements. The Education EV3 Core Set consists of: 1 EV3 programmable brick, 2 Large Motors, 1 Medium Motor, 2 Touch Sensors, 1 Colour Sensor, 1 Gyroscopic Sensor, 1 Ultrasonic Sensor, cables, USB cable, 1 Rechargeable battery and 547 TECHNIC elements. An expansion set for the Educational Core Set contains 853 Lego elements. [14]–[18] Figure 1. a) shows LEGO Mindstorms Education EV3 Core Set.

**Ultimaker 2+** is a desktop 3D printer, which was used for printing of small boxes. Its main features are: Build volume: up to 223 x 223 x 305 mm; Layer resolution: up to 20 microns; Print head travel speed: 300 mm/s; Open filament system – print with any 2.85 mm material. Figure 1. b) shows Ultimaker 2+.[19]



Fig. 1. a) LEGO Mindstorms Education EV3 Core Set b) Ultimaker 2+ [15], [19]

**MicroPython** is a full Python compiler and runtime that runs exclusively on microcontrollers. MicroPython includes a small subset of the Python standard library and is packed full of advanced features such as an interactive prompt, arbitrary precision integers, closures, list comprehension, generators, etc.[20] MicroPython uses services provided by the *ev3dev OS* itself, which is based on a Debian Linux distribution customized and fully compatible with the Lego

Mindstorms EV3 platform. EV3dev is a more dual-boot system that runs from a microSD card and does not interfere with the standard EV3 firmware. So, resetting to the factory firmware requires only easy shutdown and removal of the microSD card. MicroPython comes with a Visual Studio Code extension that includes documentation. Using the MicroPython library provides many exciting features that make EV3 robots more adaptable. Among other things, this includes brand new drivers and a new API that allows more precisely and easily actuator control.

**Robot Commander App** is the official command app from LEGO® MINDSTORMS®. Robot Commander connects via Bluetooth to the EV3 Brick. This application allows manual control of EV3 robots without connecting to a computer. Robot Commander has two versions: for the Android and for the iPhone. The Commander app supports: voice control/command, pinch control, virtual analog-joysticks, acceleration-sensors controls, etc.[21] The Robot Commander App interface is shown in Figure 2.

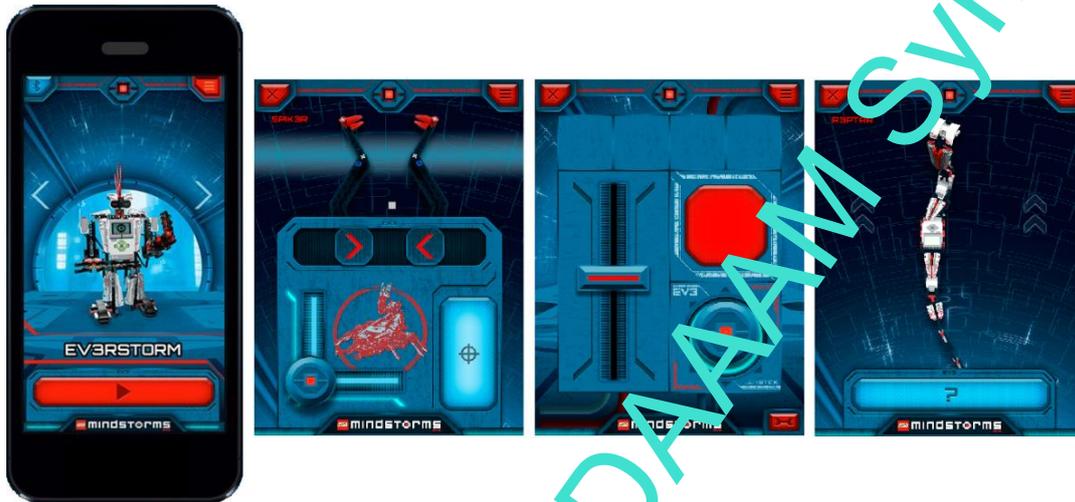


Fig. 2. Robot Commander App [22]

### 3. Robotized system for transporting of small boxes

This chapter shows the design, creation, production and assembly of all hardware and software parts of a robotized system for transporting small 3D printed boxes

#### 3.1. Creating a robotic system from the LEGO® MINDSTORMS® EV3 kit

The LEGO MINDSTORMS EV3 kit was used to realize this part of the project. A robotic arm, a conveyor belt and a remote-controlled car are integrated into one system (Fig. 3.). All three components of the robotic system are assembled from the compatible sensors, electric motors, and brick as the central processing unit. There are 8 RJ12 ports on the brick (4 for sensors and 4 for motors), so the robotic arm and the conveyor belt use a common brick.



Fig. 3. Robotic arm, conveyor belt and remote-controlled car

**Robotic arm** has two sensors and three motors, which are built into a structure made of Lego building blocks and auxiliary parts. This robotic arm is based on the Lego Mindstorms H25 robotic arm, which has been modified for this project to make it possible to attach a conveyor belt to it. The touch sensor is located below the mechanism and blocks the movement of the hand towards the area below the brick, so that the auxiliary element, which is in the same plane as the sensor and rotates together with the arm about the y axis, activates the sensor when approaching the danger zone. The second sensor, i.e. the colour recognition sensor, is located on the 'elbow' of the robotic arm. This sensor is programmed to recognize a specific colour, in this case white, in order to be able to recognize an auxiliary element that must be exclusively in the shades of the specified colour. The behaviour of the auxiliary element, in a similar way as with the touch sensor, shows when it will stop the movement of the actuator, but in this case, it is the stint that represents the upper part of the mechanism. Two motors are used, one of which is used to position the robotic arm and the other to manipulate the fist. The **conveyor belt** consists of three gears attached to the structure. A tape is attached around these gears. The motor drives one of the gears at a constant speed. Figure 4. shows a robotic arm and a conveyor belt with some details.

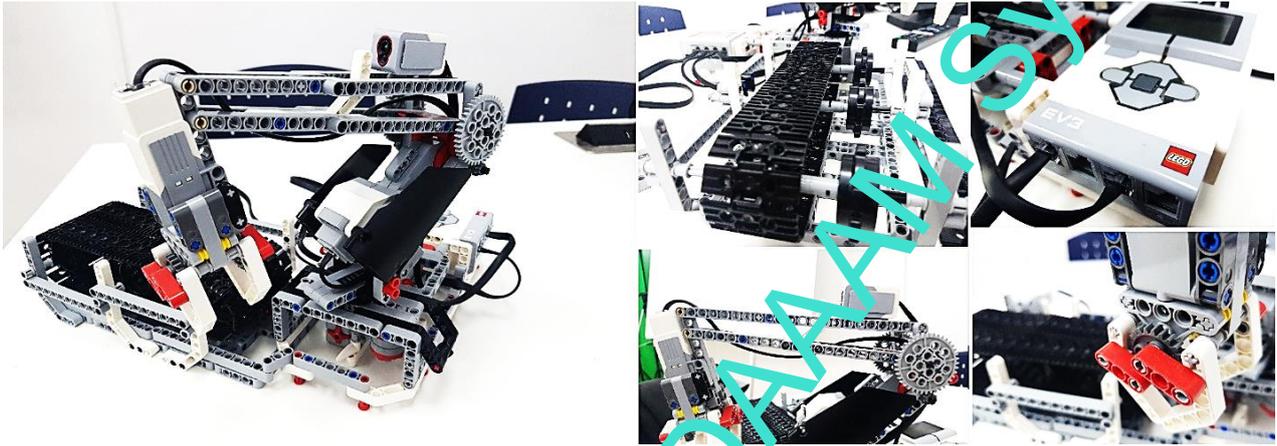


Fig. 4. Robotic arm and a conveyor belt with some details

**Remote-controlled car** has two motors and central bricks, which are built into a structure made of Lego building blocks and auxiliary parts. The front actuator is used to control the turning of the front wheels (left / right), while the second actuator drives the rear wheels and allows the car to move forward or backward. It is possible to communicate with Brick via Bluetooth and WiFi, so the easiest way to control this car is via a smartphone. The Robot Commander App was used to operate this remote-controlled car. This remote-controlled car is shown in Figure 5.

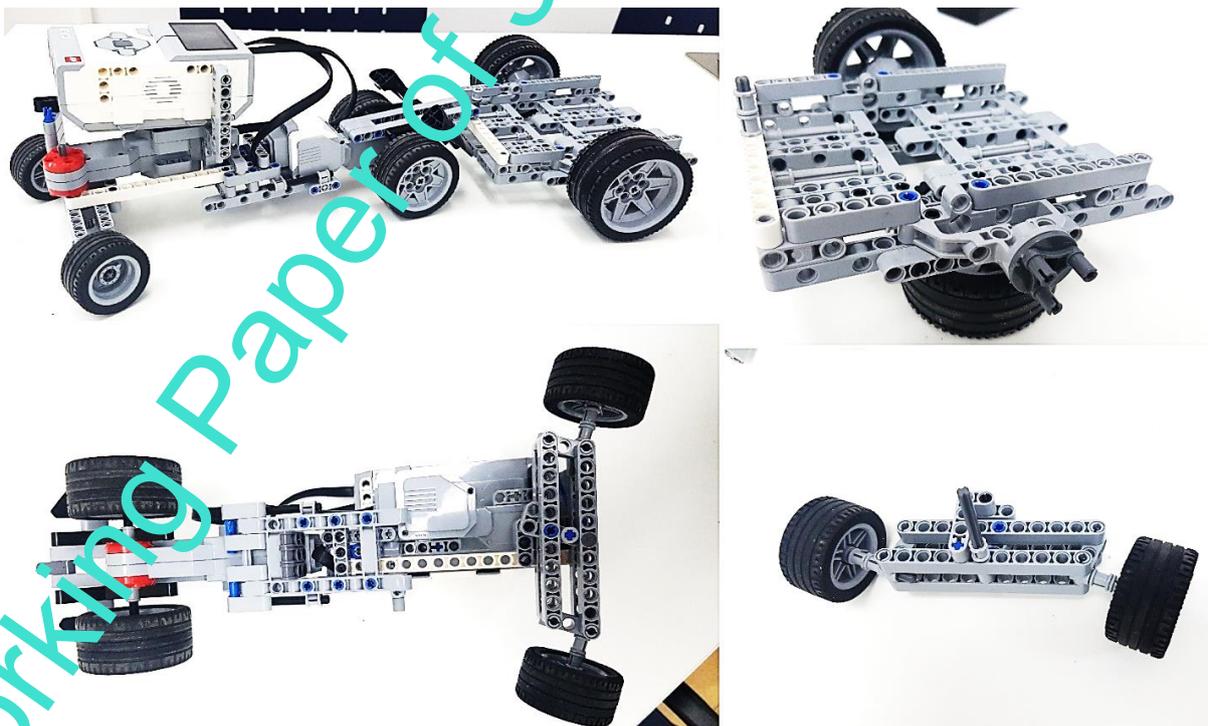


Fig. 5. Remote-controlled car

### 3.2. 3D printing of small boxes

In order to further develop their STEM skills, students had to design and print 3D objects, which will be transported by using the presented robotic system. For this project, students selected small 3D printed boxes shown in Figure 6.



Fig. 6. Small 3D printed boxes

### 3.3. Programming in MicroPython

The program related to the robotic arm and conveyor belt is procedurally programmed and consists of 3 parts:

- motor and sensor configuration,
- robot arm initialization and
- function.

The main idea for such a programming concept arose from the application of acquired knowledge of programming during studies. The code for this program is shown below:

```
#!/usr/bin/env pybricks-micropython

from pybricks import ev3brick as brick
from pybricks.ev3devices import Motor, TouchSensor, ColorSensor
from pybricks.parameters import Port, Stop, Direction
from pybricks.tools import wait

#####
# CONFIGURING MOTORS AND SENSORS #
#####

# Configure the gripper motor on Port A.
# Configure the track motor on Port D.
gripper_motor = Motor(Port.A)
trackMotor = Motor(Port.D)

# Configure the elbow motor. It has an 8-teeth and a 40-teeth gear
# connected to it. We would like positive speed values to make the
# arm go upwards. This corresponds to counterclockwise rotation
# of the motor
elbow_motor = Motor(Port.B, Direction.COUNTERCLOCKWISE, [8, 40])

# Same as elbow just different values for gears.
base_motor = Motor(Port.C, Direction.COUNTERCLOCKWISE, [12, 36])

# Limit the elbow and base accelerations. This results in
# very smooth motion.
elbow_motor.set_run_settings(50, 120)
base_motor.set_run_settings(50, 120)

# Set up the Touch Sensor. It defines the starting point of the base.
base_switch = TouchSensor(Port.S1)
```

```
# Set up the Color Sensor. This sensor detects when the elbow
# is in the starting position. This is when the sensor sees the
# white beam up close.
elbow_sensor = ColorSensor(Port.S3)

#####
# INITIALIZATION #
#####

# Initialize the elbow. First make it go down for 500 ms.
# Then make it go upwards slowly (15 degrees per second) until
# the Color Sensor detects the white beam. Then reset the motor
# angle to make this the zero point. Finally, hold the motor
# in place so it does not move.
elbow_motor.run_time(-30, 500)
elbow_motor.run(15)
while elbow_sensor.reflection() < 32:
    wait(10)
elbow_motor.reset_angle(0)
elbow_motor.stop(Stop.HOLD)

# Initialize the base. First rotate it until the Touch Sensor
# in the base is pressed. Reset the motor angle to make this
# the zero point. Then hold the motor in place so it does not move.
base_motor.run(-60)
while not base_switch.pressed():
    wait(10)
base_motor.reset_angle(0)
base_motor.stop(Stop.HOLD)

# Initialize the gripper and track. First rotate the motor until it stalls.
# Stalling means that it cannot move any further. This position
# corresponds to the closed position. Then rotate the motor
# by 90 degrees such that the gripper is open.
gripper_motor.run_until_stalled(200, Stop.COAST, 50)
gripper_motor.reset_angle(0)
gripper_motor.run_target(200, -90)

# Run track (we want track to run continuously)
trackMotor.run(140)

#####
# DEFINING FUNCTIONS AND DESTINATIONS #
#####

def robot_pick(position):
    # This function makes the robot base rotate to the indicated
    # position. There it lowers the elbow, closes the gripper, and
    # raises the elbow to pick up the object.
    # Rotate to the pick-up position.
    base_motor.run_target(60, position, Stop.HOLD)
    # Lower the arm.
    elbow_motor.run_target(60, -40)
    # Close the gripper to grab the wheel stack.
    gripper_motor.run_until_stalled(200, Stop.HOLD, 50)
    # Raise the arm to lift the wheel stack.
    elbow_motor.run_target(80, 0, Stop.HOLD)
```

```
def robot_release(position):
    # This function makes the robot base rotate to the indicated
    # position. There opens the gripper to
    # release the object. Then it raises its arm again.
    # Rotate to the drop-off position.
    base_motor.run_target(60, position, Stop.HOLD)
    #This one sparks joy!
    elbow_motor.run_target(60, -5, Stop.HOLD)
    # Open the gripper to release the wheel stack.
    gripper_motor.run_target(200, -90)

# Play three beeps to indicate that the initialization is complete
brick.sound.beeps(3)

# Define destinations for picking up and moving the wheel stacks
MIDDLE = 100
RIGHT = 4
LEFT = 200

# This is the main part of the program. It is a loop that repeats endlessly.
#
# First, the robot moves the object on the left towards to the track on right.
while True:
    robot_pick(LEFT)
    robot_release(RIGHT)
```

#### 4. Review and analysis of the system functionality

Transfer of small boxes is performed in the following steps:

- I. Small boxes are located on the surface next to the robotic arm (Fig. 7. a))
- II. The system finds the boxes and a robotic arm lifts them one by one (Fig. 7. b))
- III. The robotic arm lowers the boxes one by one slowly onto the conveyor belt (Fig. 7. c))
- IV. The boxes move on the conveyor belt one after the other and fall into the trailer of the remote-controlled car (Fig. 7. d))
- V. The Robot Commander App is used to control the car, and the boxes are driven away from the conveyor belt.
- VI. Finally, the boxes can be removed manually from the trailer to restart the process.

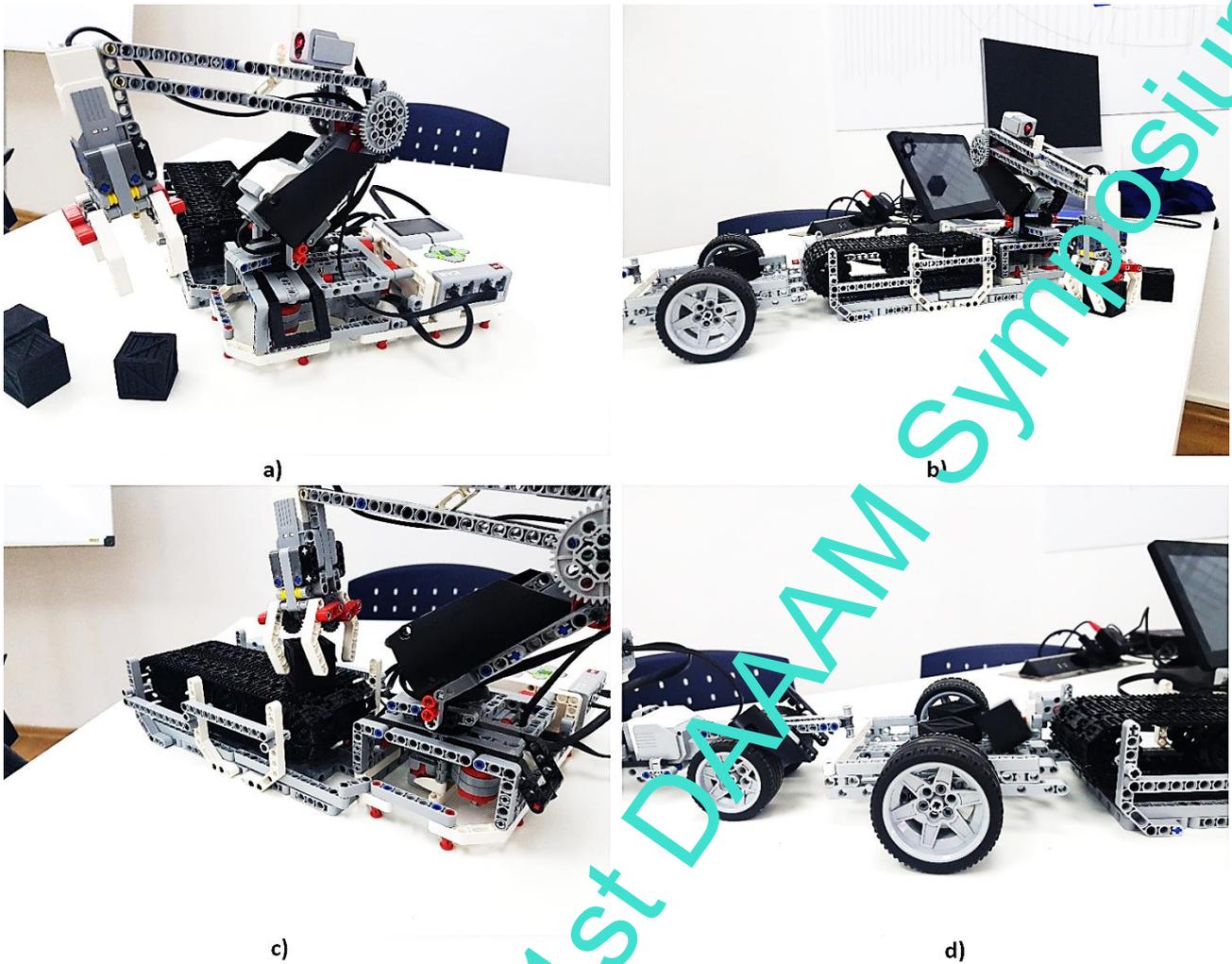


Fig. 7. Steps in the transport of small boxes

It is obvious that the transport of small boxes is performed in the way as it is defined through the main idea of this student project. Through a large number of attempts, all parts of the system performed their tasks without any errors or difficulties. The application programmed in MicroPython is optimized through several iterations, and the final version, presented in this paper, showed very good performance allowing the system to perform all functionalities at the highest level.

Figure 8. shows students working on this project within a STEM classroom.



Fig. 8. Students working on the project within the STEM classroom

This approach to acquiring STEM competencies has proven to be very effective. However, certain limitations have been noted, which relate to:

- the number of EV3 sets was not enough to include all students, but only one group of three students,
- there are not enough components in EV3 kits (primarily sensors, actuators and control units) to make the system fully robotic (the car would not be remotely controlled),
- manual control of the car requires a lot of skill because the actuators have uneven initial accelerations,
- it is very difficult to automate more complex applications because it is necessary to own a large number of EV3 educational kits,
- since EV3 sets are located in the faculty classroom, it was very difficult for students to do some of the activities that involved homework,
- motivation among students was very high, however, due to previous restrictions, students showed a certain amount of dissatisfaction and reduced productivity.

Taking into account all the above limitations, the robotic system provided students with the opportunity to create a synergy of different knowledge and skills acquired through previous education. As was the case in similar projects listed in the introductory part, through this project students have acquired new competencies that are useful for further study of STEM disciplines: engineering, coding and testing of control algorithms, physics, design, construction and production, critical thinking, problem-solving, collaborating, etc.

## 5. Conclusion

The project in this paper is certainly an excellent example of teaching students within STEM disciplines. Through various parts of the project, students applied the knowledge and skills acquired so far, greatly improved them but also gained a new knowledge and skills. These knowledge and skills can be divided into technical and non-technical. Technical knowledge and skills are primarily related to:

- control of electrically driven mechatronic systems,
- basics of control, construction, measurement quantities and parameterization of control,
- basics of electric drive technology,
- commissioning of mobile robotic systems,
- sensor-guided motion control of the mobile robot,
- assemble and integrate various sensors,
- assemble and integrate different actuators,
- using a modular set of educational robotics,
- programming in well-known programming languages and adapting them to new systems and environments,
- basics of modelling in 3D virtual environments,
- basics of 3D printing technology,
- use of mobile applications for remote control of systems.

On the other hand, non-technical knowledge and skills are primarily related to:

- combining different technologies and adapting to their application,
- developing of creativity in the design and production of various constructive ideas,
- finding new solutions to improve already known functionalities,
- encouraging students to research and reflect scientifically,
- developing critical thinking and a competitive spirit,
- improving teamwork, communication and knowledge transfer,
- building a cheerful work environment and emotional connection.

Classroom robotics has proven to be a very good practice through positive application results with students such as encouraging students to focus more on jobs and careers in the STEM field, and to develop the necessary 21st century skills that will enable them to succeed in the future. The new generation of students is strongly connected with technology, so it has become necessary to further involve them in this field by creating new platforms through which they can get closer to new technologies such as robotics. For the technological challenges of the future, it is necessary to find adequate solutions and answers, i.e. new interesting and exciting ways to teach pupils and students those courses that will provide them with the necessary skills and abilities to succeed in the near future. Engaging students in STEM education is necessary to find a large number of new jobs in the future. With the commitment of academia, business and government institutions, it is possible to design and implement comprehensive and effective education that includes robotics and STEM as essential disciplines.

The EV3 kit can be applied as part of teaching at higher education institutions. However, it is necessary to have a larger number of sets with a larger number of sensors and actuators so that students have more opportunities to realize their ideas and to create more complex robotic systems. Also, by having a larger number of EV3 kits, it is possible to involve a larger number of students to work together at the same time. In future work, the idea is to use a vision sensor (VGA camera) to allow full recognition of small boxes, and to fully automate car control. This would improve the

accuracy and operating speed of the system itself. Also, it is necessary to enable students to use EV3 kits as part of homework as this will greatly help students when working on similar future projects. In the end, all the experiences with this way of teaching, focused on practical work through projects, showed excellent results. Students applied the acquired knowledge much more effectively, and it was much easier to accept new knowledge while working in a pleasant environment.

## 6. References

- [1] T. B. Hinton, "An exploratory study of a robotics educational platform on STEM career interests in middle school students," *Diss. Abstr. Int. Sect. A Humanit. Soc. Sci.*, vol. 78, no. 11-A(E), p. 146, 2018, [Online]. Available: <http://ovidsp.ovid.com/ovidweb.cgi?T=JS&PAGE=reference&D=psyc13&NEWS=N&AN=2017-50562-131>.
- [2] B. Crnokic, M. Grubisic, and T. Volarić, "Different Applications of Mobile Robots in Education," *Int. J. Integr. Technol. Educ.*, vol. 6, no. 3, pp. 15–28, Sep. 2017, Accessed: Oct. 16, 2017, [Online]. Available: <http://arxiv.org/abs/1710.03064>.
- [3] teaching.com.au, "LEGO® Education Classroom Solutions Provide hands-on experiences to explore core STEM concepts," 2020. <https://d4iqe7beda780.cloudfront.net/resources/site/mtaau/landingpages/catalogues/pais/2020/lego-education.pdf>.
- [4] I. M. L. Souza, W. L. Andrade, L. M. R. Sampaio, and A. L. S. O. Araujo, "A systematic review on the use of LEGO® robotics in education," *Proc. - Front. Educ. Conf. FIE*, vol. 2018-1, no. October, 2019, doi: 10.1109/FIE.2018.8658751.
- [5] J. Dolinay, P. Dostálek, and V. Vašek, "Educational tool for controlling lego mindstorms robots with control web," *Ann. DAAAM Proc. Int. DAAAM Symp.*, pp. 443–450, [Online]. Available: <https://www.thefreelibrary.com/Educational+tool+for+controlling+Lego+mindstorms+robots+with+Control...-a0224712416>.
- [6] E. Afari and M. S. Khine, "Robotics as an Educational Tool: Impact of Lego Mindstorms," *Int. J. Inf. Educ. Technol.*, vol. 7, no. 6, pp. 437–442, 2017, doi: 10.18178/ijiet.2017.7.6.908.
- [7] O. Shmakov, D. Korolev, D. Popov, N. Kitaev, and A. Korotkov, "Modular mobile robotic kit for prototyping and debugging of control algorithms," *Ann. DAAAM Proc. Int. DAAAM Symp.*, pp. 950–956, 2017, doi: 10.2507/28th.daaam.proceedings.132.
- [8] O. Keskin, L. Cetin, O. Baser, and E. Uyar, "Multi robot path planning framework," *Ann. DAAAM Proc. Int. DAAAM Symp.*, vol. 22, no. 1, pp. 1015–1016, 2011.
- [9] L. Peters, "An Educational Programming Environment for Teaching the Principles of Machine Learning using LEGO MINDSTORMS," no. December, pp. 1–84, 2019, doi: 10.13140/RG.2.2.25925.60647.
- [10] J. Olayvar and E. Lindberg, "LEGO Mindstorms EV3 Programming Basics," 2016. [Online]. Available: [https://www.sos.wa.gov/\\_assets/library/libraries/projects/youthservices/legomindstormsev3programmingbasics.pdf](https://www.sos.wa.gov/_assets/library/libraries/projects/youthservices/legomindstormsev3programmingbasics.pdf).
- [11] B. Blais, "Using Python to Program LEGO MINDSTORMS Robots: The PyNXC Project," *Python Pap.*, no. January 2010, 2010.
- [12] Lego, "Getting started with EV3 MicroPython," 2019. [Online]. Available: <https://le-www-live-s.legoocdn.com/sc/media/files/ev3-micropython/ev3micropythonv100-71d3f28c59a1e766e92a59ff8500818e.pdf>.
- [13] L. Cornelissen, "Simulating Lego Mindstorms EV3 robots using Unity and Python," *Radboud University Nijmegen*, 2019.
- [14] Lego Group, "Mindstorms EV3 User Guide," 2013.
- [15] LEGO, "LEGO MINDSTORMS Education EV3 Core Set." <https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-core-set-5003400> (accessed Apr. 25, 2018).
- [16] LEGO, "LEGO education." <http://education.lego.com/?domainredir=www.legoeducation.com> (accessed Jun. 03, 2016).
- [17] "Robot – Wikipedia." <https://hr.wikipedia.org/wiki/Robot> (accessed Mar. 20, 2019).
- [18] "Lego Mindstorms EV3 - Wikipedia." [https://en.wikipedia.org/wiki/Lego\\_Mindstorms\\_EV3](https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3) (accessed Sep. 20, 2020).
- [19] "Ultimaker 2+: Robust single extrusion." <https://ultimaker.com/3d-printers/ultimaker-2-plus> (accessed Sep. 20, 2020).
- [20] "MicroPython - Python for microcontrollers." <https://micropython.org/> (accessed Sep. 20, 2020).
- [21] "LEGO® MINDSTORMS® Robot Commander - Games - LEGO.com for kids - US." <https://www.lego.com/en-us/kids/games/mindstorms/mindstorms-robot-commander-db0f6dcee8224b60b9f7cddb11c44f3b> (accessed Sep. 22, 2020).
- [22] "LEGO® MINDSTORMS® Commander, Aplikacije na Google Playu." <https://play.google.com/store/apps/details?id=com.lego.mindstorms.robotcommander> (accessed Sep. 22, 2020).