STEM Education LLC

Pilot Competition 2023-24

Open Category

(3rd - 6th grades of Primary School)

City of My Dreams Smart City

Rules and Scoring

A' Edition (October 2023)

Diligence

Ap. Ambariotis, An. Ladias, G. Souvatzoglou, F. Fotinakis

STEM education LLC

Mediterranean Sea Source of Life and Culture

Contents

Description of the subject

A smart city is a technologically modern urban area that uses different types of electronic methods and sensors to collect specific data. Information gained from that data is used to manage assets, resources and services efficiently; in return, that data is used to improve operations across the city. This includes data collected from citizens, devices, buildings and assets that is processed and analyzed to monitor and manage traffic and transportation systems, power plants, utilities, urban forestry,[3] water supply networks, waste, criminal investigations, information systems, schools, libraries, hospitals, and other community services. Smart cities are defined as smart both in the ways in which their governments harness technology as well as in how they monitor, analyze, plan, and govern the city. In smart cities, the sharing of data is not limited to the city itself but also includes businesses, citizens and other third parties that can benefit from various uses of that data. Sharing data from different systems and sectors creates opportunities for increased understanding and economic benefits.

The smart city concept integrates information and communication technology ('ICT'), and various physical devices connected to the Internet of things ('IoT') network to optimize the efficiency of city operations and services and connect to citizens. Smart city technology allows city officials to interact directly with both community and city infrastructure and to monitor what is happening in the city and how the city is evolving. ICT is used to enhance quality, performance and interactivity of urban services, to reduce costs and resource consumption and to increase contact between citizens and government. Smart city applications are developed to manage urban flows and allow for real-time responses. A smart city may therefore be more prepared to respond to challenges than one with a conventional "transactional" relationship with its citizens. Yet, the term itself remains unclear in its specifics and therefore, open to many interpretations. Many cities have already adopted some sort of smart city technology.

Deakin and Al Waer list four factors that contribute to the definition of a smart city:

- The application of a wide range of electronic and digital technologies to communities and cities.
- The use of ICT to transform life and working environments within the region.
- The embedding of such Information and Communications Technologies in government systems.
- The territorialisation of practices that brings ICT and people together to enhance the innovation and knowledge that they offer.

Source:
https://en.wikipedia.org/wiki/Smart_city

The identity of the Competition

STEM - Educational Robotics Pilot Competition (October – March) has the following category, for students:

- Elementary school

Elementary level includes:

- The **Open Category (STEM)** and
- the Football 2x2

The **Open Category** of the Primary School Competition is divided into two levels for:

- **the grades (1st-3rd) and**
- **the grades (3rd-6th).**

The tender to which this document refers and which will henceforth be referred to as ***Contest*** is:

- **of the Open category,**
- **of the "Last classes (3rd-6th)",**
- **of the Primary School,**
- **of the STEM Educational Robotics Pilot Competition,**

The ***Contest***:

- is an educational activity,
- has a subject related to Smart Cities
- addressed to groups of 3-6 children, who are accompanied and guided by an adult coach (teacher, parent...),
- will be held (the final) live (unless unforeseen) in ……place…,
- is announced in October and is completed at the beginning of the next Spring (end of March),
- is organized by STEM Education LLC
- offers for coaches, free intensive webinars related to the educational tools listed to the contest.
- In general it aims to act as a nursery of students experienced in automation and robotics and to create an active core of educators-coaches trained in the philosophy of STEM.

The philosophy of the Competition

The Competition was designed from the beginning as a nursery for the qualitative upgrading of students' skills in educational robotics, within the context of STEM education.

His goal is to create a numerous **pool of students with the necessary skills**, who in turn will be the pioneers for the introduction of STEM in education.

The philosophy of the Competition is the young and beginner students involved:

- to acquire knowledge through analysis and synthesis when trying to solve problems,
- to accumulate experiences working creatively in various fields such as:
    - the structural (mechanical) construction,
    - the electronic part of automation and
    - programming the simulation with (synchronous) animation,
- to freely develop their imagination in the construction of the model,
- to improve their communication skills when presenting the project,
- to familiarize themselves with teamwork,
- to enjoy the joy of creation and
- to gain confidence in feeling that they are producers of potentially innovative products.

The cognitive background that can be both a prerequisite and a result of participation in ***Competition*** is the solid knowledge of engineering systems development and computer programming (for more see the Appendix paragraph "**The dual nature of educational robotics projects**"). We believe that educational robotics rests on two pillars: **constructions of mechanical systems** and **computer programming**.

The participation project

The group of students participates in ***Competition*** with one **original work** that has been developed during the preparation period.

The project should:

- be based on a "**innovative idea**" (novel at least for kids),
- be supported by an interesting **"script narration"**,
- to satisfy a need, to propose a **resolution** in a **problem,**
- to choose the **best feasible solution,**
- to describe the **compromises** achieved between the initial design and the final product,
- to present the solution by incorporating the project automations in a **diorama** and
- to document the **programming code implementations**.

The theme of the Competition

One of the **long-term goals** of ***Competition*** is to leave its imprint on the young participants, to teach problem solving via creativity so that they will be receptive when later in their adult lives they are called upon to face real problems. This goal is expressed through the choice of the themes of the ***Competition***. So there exists the combination of wanting to leave an imprint on the young participants with **authentic and interesting Competition themes**.

Into this context also fits the choice of the subject of the ***Competition* 2023-24**. The projects that will participate in the competition should have a topic related to the title:

" City of My Dreams - Smart City"

The aim is to propose improvements of the quality of life of citizens in the context of sustainable development.

Works not related to the competition theme will not be evaluated.

The hardware required for the project

For construction can be used:

● The **Introduction to automation** robotic kit

AND

● **Web Camera**

AND

● **up to two computers.**

AND optional

● **the micro:bit microcontroller, with additional optional external sensors**

The use of other materials (other than those mentioned above) is allowed only for the model and the scenery of the project.

The software required for the project

As software can be used:

● **Scratch-3,** (https://scratch.mit.edu/)

Or also

● **any MIT Scratch-based environment derivative** such as the **Mind+** environment (http://mindplus.cc/en.html)

Automation of projects

The project requires at least 3 automations* of which:

● **The first automation,** uses as hardware the robotic kit "**introduction to automation**". The automation **is necessarily related to its simulation in the Scratch (-like) environment**. Data from sensors are used to drive at least one physical actuator and the related simulation (animation) in Scratch Scene (top right).

AND

● **The second automation uses a USB camera for image recognition in Scratch environment**. In addition to the web camera, it can also use any other (of the permitted) actuators.

AND

● **The third automation (optional)** uses at least one **Micro:bit** and its sensors (internal or external) to **take measurements and plot** one or more basic physical quantities.

Analytically

First automation

First automation is necessarily related to programming **and animated simulation** in a Scratch environment.

● uses as hardware the robotic kit "**Introduction to automation** "

● uses at least one of the **sensors** of the systems mentioned.

● uses the allowed **actuators**. These actuators are: the motors and leds from the **introduction to automation Engineer** kit.

● Is programmed with **MIT Scratch or scratch like environment** (scratch 2 and s2bot software is recommended)

● in this software, the simulation of the automation is created on the computer screen using animation.

Second Automation

It uses **as sensor a USB camera** connected to the computer

● Is programmed with **MIT Scratch or scratch like environment** (scratch 2 and s2bot software is recommended)

● Triggered when the software camera performs some image recognition (described below).

● uses at **least one of the allowed actuators.**

These actuators are the motors and leds from the robotic kit "**introduction to automation**"

The camera monitors its environment continuously and when it perceives the requested image it activates the automation.

Movement or change of the objects in the camera's field of view (which activates the third automation) should occur **without human intervention.**

**** 2 of the 3 automations should have a motor as an actuator.**

Third automation (optional)

Third automation is necessarily related to **measuring one of the following physical quantities**: distance, temperature, weight, magnetic field intensity, brightness, sound intensity, conductivity, voltage, velocity, angular velocity, acceleration, pressure.

● uses **micro:bit microcontroller**

• uses at least one of the sensors necessary to measure the permissible physical quantities.

The allowed sensors are the sensors **built onto the micro:bit or external sensors** connected to the micro:bit (through Shields) which are suitable for measuring the physical quantities mentioned above,

● uses at least one of the **allowed actuators**.

These actuators are: the **motors** and **LEDs** from the WeDo kit, the micro:bit built-in 5x5 LED matrix and **buzzer** (loudspeaker), motors, servos, external LEDs and relay circuits that the micro:bit can somehow drive (through Shields) (LCD and OLED character & graphical displays can be used in the project (diorama) but are not considered as actuators in automations).

The software can be **MIT Scratch and/or scratch like environment** like Mind+ **and/or MakeCode (makecode.microbit.org).**

In third automation the system will take measurements of one of the previous physical quantities**. The value of the measurement will be compared to a threshold** value that the team will have determined to demonstrate the **activation** of the automation.

During an automation demonstration, **data collected from the micro:bit, should be displayed graphically, in real time**, on the computer screen (**data graph**).

Recommended materials for the implementation of the automations

Allowed systems for the implementation of the automations are systems that exist in schools and are compatible with the requested automations.

- Introduction to automation
- micro:bit v2 starter:kit
- Primary school measurement and automation package

Electronic Automation

Automation is the process by which a real-world stimulus is perceived by an electronic **sensor**, then a **program in the processor** processes the data and gives a command that changes the state **of an electronic actuator**.

Examples of automations:

- Suppose we have an escalator in the subway that works depending on whether passengers are passing and we assume that the detection of passengers is done by a weight sensor.
  - If the weight sensor is simply triggered by the weight of the human (or a dog or a suitcase) THEN that's automation
  - If the weight sensor is a button that the passenger has to press with their hand (manipulation) THEN this is NOT automation (doesn't work with a dog or a suitcase).
- Suppose we have a drone (e.g. a quadcopter) and a tilt sensor on it.
  - If the sensor takes and transmits inclination values to a central processing unit and this unit increases or decreases the power on the corresponding motor to balance the plane, THEN this is automation.
  - Also If we demonstrate the former without flying the drone but holding the drone by hand and tilting it to simulate the flight, THEN that is automation.
  - But if we use the sensor as a joystick to remotely control the drone THEN that does **not** constitute an automation.

Image recognition

Educational material for creating programs for **image recognition:**
https://drive.google.com/drive/folders/1FRh4-Oi0pPS_b4MkmHKAbjgxNMG9YJ56?usp=sharing
The purpose of this automation is to recognize image patterns from camera shots and then give the command for an action on an electronic actuator. In Scratch-3 this is achieved by using an (internal or external USB) camera (still or moving image), the image of which is projected as the background of the Scene. With the "video sensing" feature (found in the "sensing" menu), Scratch can **detect**: movement, speed, color recognition or interaction with roles (sprites) already present in the Scene.

Similar and enhanced features are provided in the Scratch-like software application "Mind+".

The simulation of automation

In the first automation is also required a **simulation of the physical automation on the computer screen using animation graphics.** The physical automation and the simulation have to be synchronized. Specifically, during the operation of the automation, the input data from the sensor should also be used by the simulation program so that alongside with the automation activity being activated at **the same time** the simulated - virtual actuators and their action to be displayed in the form of animation in the Scratch or Mind+ environment, using graphics.

The Code-O-rama of the program

At least for the **two automations (First) & (Second),** needs to be specially documented, **to represent the code** by creating one or more code-O-ramas.

The **Code-O-rama** is a code representation tool whose main goal is to facilitate the teaching of visual programming by allowing the observer to have both an overview of the entire code and access to its details, helping him to not get disoriented while immersing himself in the details of the code. In code-O-rama, the set of scripts of all the objects of a program is displayed in a two-dimensional table (**code anatomy**) and the modes of communication between them are also represented (**code functionality**).

More information about code-O-rama can be found at the link:
https://drive.google.com/file/d/1Pwe_c_amxWdAJbUnmvjM6w6nRkzKQtNI/view?usp=share_link

The deliverable: Portfolio folder

Each group should submit a suitable electronic documentation folder i.e. a portfolio (using DROPBOX or GOOGLE DRIVE etc.), containing:

A. The documents with parental **consent** for the use of their photos or videos featuring their faces (special printable forms to be posted on the stem-educ.com website).
B. A short project **description** (MS word-type document) that will highlight the problem it solves.
C. The file/s with the **program in** Scratch and/or Mind+ or Makecode.
D. A zipped folder (*.zip) with the files with photos where the stages of the construction can be seen and in particular, the constructions of the mechanisms & automations.
E. A file (or its link) from at least one video, where the students will present their project by describing and showing the operation of all the constructions, with an emphasis on the automations (zoom-in to see the automation) in operation and its size not to exceed 100MB. (Extremely necessary if the competition is conducted online). Attention: Projects whose video size will exceed 100 MB will not be included in the portfolio evaluation process.
F. The file with the code-O-rama (in *.xls or *.pdf or *.png or *.jpg).

For the above, they will create separate **folders named A, B, C, D, E** and **F** inside **group dropbox** into which the aforementioned required files will be loaded (not compressed all together in one zip).

With Coach's **responsibility the** portfolio folder must be submitted **electronically and on time** with a specific date communicated by STEM Education (usually, at least 10 days before their participation in the regional competition of their country).
Projects submitted late, it is at the discretion of the Jury to decide whether to participate in the ***Competition*** and will be evaluated.

Steps for submitting the portfolio:

- The teams will create (e.g.) a **dropbox** account that will belong to them and upload the portfolio there. **Attention**: Make sure that access permission is granted to third parties (competition judges).
- The link to the group's **dropbox** portfolio will be shared with STEM Education LLC . Specifically, the teams will later edit their initial registration form and fill in the "Required deliverables" field with this particular link to their dropbox portfolio folder.
- Teams can find and edit the registration form at any time by checking their inbox, specifically the email with the subject "Participation Confirmation" and from **eventora**.

The mockup of the project

The project should be supported by a "**script narration**" which will take place in some place. This space will be represented in the project with a mockup which will constitute the scenery in which the automations will be included.
On the day of the competition, each team will be allocated a "**kiosk**" with a space of approximately 150 cm x 150 cm and with a vertical heigth of approximately 2m. The printed material can be stuck on the back, e.g. the code-O-rama slide (if it is large) or to view (with a group projector) the presentation.
There will be at the booth/kiosk, a **Workbench** table size approx 100cm **x 60cm**. In this space, the model should be installed together with the automations. Alternatively, the team can also place their work on the floor, as long as it does not exceed the limits of the booth/kiosk.
There will be available at the stand a mains power **supply** with power strip.
There will be no wired internet or wifi.
The robotics systems with the possibility of developing free mechanical precision constructions which **exist** in schools for the age group we are referring to is either the Lego type or the GIGO type. Given this, all parts of the construction that contain mechanical automation or mechanical parts moving with motors should be made with plastic LEGO or GIGO building blocks. The remaining parts of the structure can be made of any other material (such as foam, paper, etc.)

The presentation of the project by the students

On the day of their performance ***Competition*** (Regional or Final) teams must:

- install in the "stand"/kiosk/booth that will be available to them, the model, the automations and the props, which are transported prefabricated and pre-assembled,
- ensure that the facility complies with the regulations,
- be ready at all times, to demonstrate and present their work to the public (if requested).

For the judgment of the projects, limited **time** is available for each group (arising from a compromise between the number of groups and the total assessment time available). This time may be seven **minutes (**an indication), of which most part (e.g. five minutes) will be for the presentation and the remaining time for questions from the judges.
During the evaluation of their work by the judges, the teams should present their narrative work, as their "innovative idea" and their invented script in a lively, "**theatrical way**". The presentation can be supported either with a short **form** or with **a Power Point presentation,** in which the main characteristics of the project should be mentioned.
In an atmosphere of teamwork, each member of the team according to the role he/she played during the development of the project takes the floor and:

- states how their work relates to the competition theme,
- narrates the "script" on which the work is based and guides the judges through the mock-up,
- explains how automation solves the requested problem
- performs a demonstration of operation of automations
- demonstrates simultaneous simulation during automation operation
- refers to code-O-rama documentation to explain the automation, simulation, and measurement sequence capture code
- answers potential questions from the judges, related to their project and its development process.

During the evaluation no help of any kind is allowed from the coach or anyone else apart his team members.

Indicative project evaluation criteria

In the ***Competition*** the evaluation of the projects is done by panels **of judges** which as a rule consist of experienced **educators specialized in STEM education and educational robotics**. Each committee consists of 2 to 5 judges who rank the projects of the groups assigned to them. In competitions in which several teams participate in the final stages of the competition, all judged teams are evaluated by the same committee. In such competitions, often in addition to the judging committee that evaluates the medals, there is also another committee that evaluates the individual thematic awards.
In the evaluation for the medals the judges can consult **the** following rubric.

Rubric with indicatively evaluation criterias

CATEGORIES CRITERIA GRADES			
Conceived Idea / Project	**TOTAL MARKS: 40**		
	1	Creativity, Research and Idea Development	20
	2	Quality of solving the challenge	20
Educational Robotics/ Automation	**TOTAL MARKS: 90**		
	1	Mechanical Construction, Calesthesia	20
	2	First Automation	20
	3	Second Automation	20
	4	Third Automation (micro:bit)	10
Program in Scratch	**TOTAL MARKS: 40**		
	1	**Codemap - Visual representation of code**	40
Virtual world	**TOTAL MARKS: 40**		
	1	Correctness of Logic, Complexity of Software and automations	20
	2	Automation simulation with animation, Interface, Aesthetics	20
Presentation	**TOTAL MARKS: 30**		
	1	Assessment of Presentation, Communication Skills and Collaboration	20
	2	Decoration, Videos, Posters	10
MAX SCORE: 220			

Complaints and objections procedure

The "waterfall process" followed in ***Competition does*** not allow time delays and because of this **It is not feasible** to effectively implement an objection process during the competition.
Objections, complaints, objections and suggestions - in written form - are accepted, evaluated and utilized by the scientific and organizational committee of the competition, for the continuous improvement of the following competitions.
In our experience, the few objections to the evaluation arose from a lack of understanding - usually on the part of those who expressed them, perhaps with justification - of the rules of the competition.
Due to the open nature of the competition, the evaluation includes factors that do not allow a weighted and "objective" score for non-quantifiable (but recognizable) qualities of the contestants' works such as originality, aesthetics, presentation, etc. Thus, from the first competition - although a rubric with a quota for the score is indicatively used - it was adopted the **"cup" model,** which highlights the cup winner with successive **exclusions** instead of the "championship" model where the champion emerges with the points he collects. This "cup" model is adopted in the evaluation phases, in which groups (the groups of teams that each committee will evaluate) are formed. The rubric can act as an advisory for panel judges.
The spatial proximity of teams belonging to different groups and the comparison between them gives rise to complaints of unfair treatment. The random placement of the teams in the various groups is something that objectively cannot be avoided and it is the duty of the team coaches to understand this way of operation and to explain it to the children of their teams and to the parents.

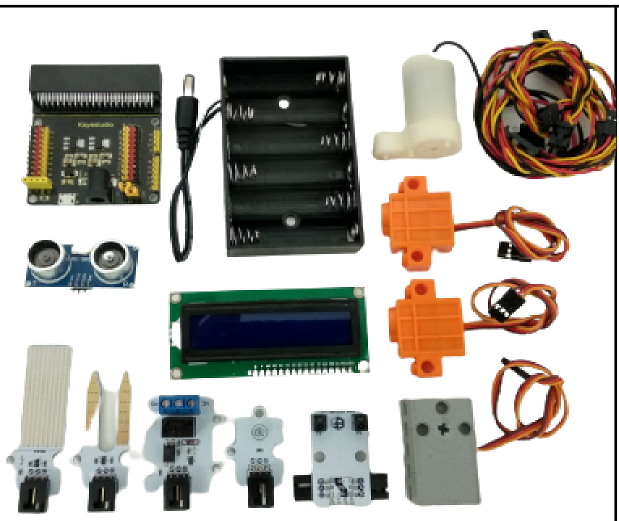
It is important as a personal assessment that each child compares himself with how he/she was before getting involved in the competition process and what he/she has conquered on his/her behalf and how he/she has evolved through his/her participation in the competition.

ANNEX

Allowable materials, sensors and motors from Introduction to automation:

	[Smart Hub 2 – Interface With PC or Tablet]()
	[Medium Motor]()
	[Tilt Sensor]()
	[Motion Sensor]()
	[Introduction to Automation]()

Recommended micro:bit compatible materials

	[Elementary Measurements and Automation Package](#) This material package has been tested and works perfectly with the proposed materials and software of the contest, and is supported by relevant examples. • Shield for Micro:Bit • Geekservo 2kg 360 Degrees compatible with lego • 2x360 servo motors compatible with lego • Base for 6xAA Batteries with Jack 5.5×2.1 • LCD 1602 I2C Module • Distance sensor 2-400cm HR-SR04 (digital) • Octopus 1 Channel 3v Relay Module • Octopus light sensor (analog) • Octopus Soil Moisture Sensor (Analog) • Octopus LM35 Analog Temperature • Octopus Water level sensor (analog) • Octopus 2 Channel Tracking Module (digital) • Water pump • Jumper Wires F/F • Jumper Wires M/F

Technical clarifications related to “Introduction to Automation” system

- The **connection of the sensors to the computer is done** (for Introduction to Automation) via bluetooth and for Microbit via USB connection cables only.
- In case a group does **not have two hubs**, the first sensor can be placed on the hub to present the first automation, then the first sensor can be removed, the second sensor can be placed to present the second automation. A similar procedure can be followed in the case of using two **motors** and a sensor through a hub.

Technical clarifications related to the micro:bit

Instructions for simultaneous interface micro:bit, WeDo 2.0 and USB camera on Mind+

In the proposed procedure Mind+ software app (largely compatible with Scratch-3) is used in combination with its own software version of **ScratchLink** to achieve the desired result.
Basic requirements for the bluetooth connection of WeDo 2.0 to the computer, the computer during the WeDo search process must have Internet. After the bluetooth connection is restored, the Internet is not necessary. As a wireless network is not officially provided in the area, then it is suggested to give the team's PC computer, a wireless network through a mobile phone.

In the following procedure the Micro:bit is connected to the computer with a USB cable and the camera is the internal computer or external USB.

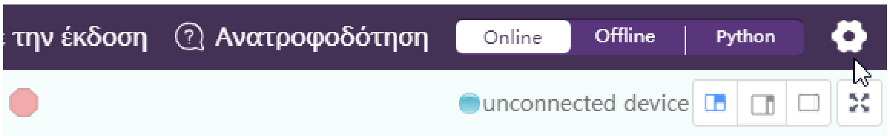
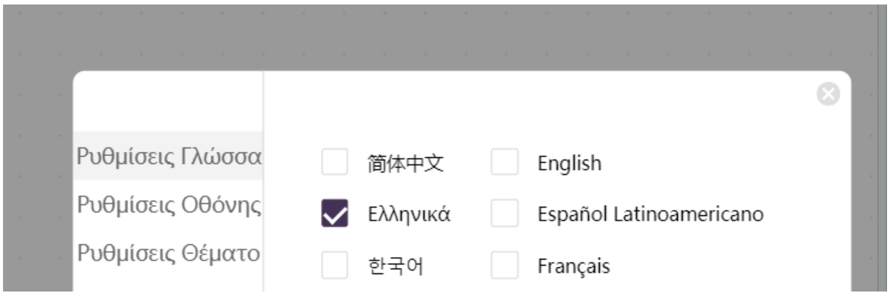
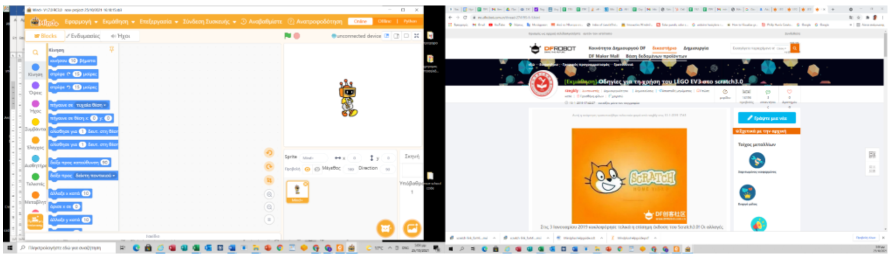
Download and install mind+ software

- Download the software from: https://drive.google.com/drive/folders/1W55GOq-xhqUWy3BtZFbEv0uoDxTv2rLi
- Run the executable
- Follow the installation process

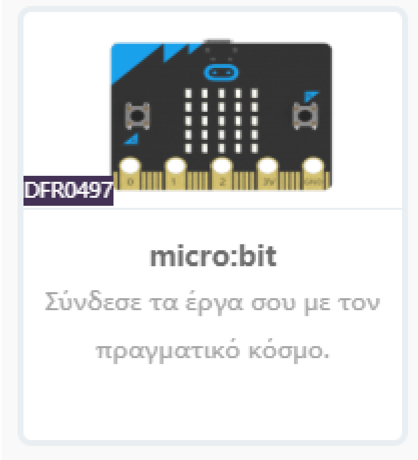
Download and install a specific version of the Scratch-Link software

- Download the program from the link: Scratch-Link http://download3.dfrobot.com.cn/scratch-link_forMindPlus-v1.1.msi
- Follow the installation process.

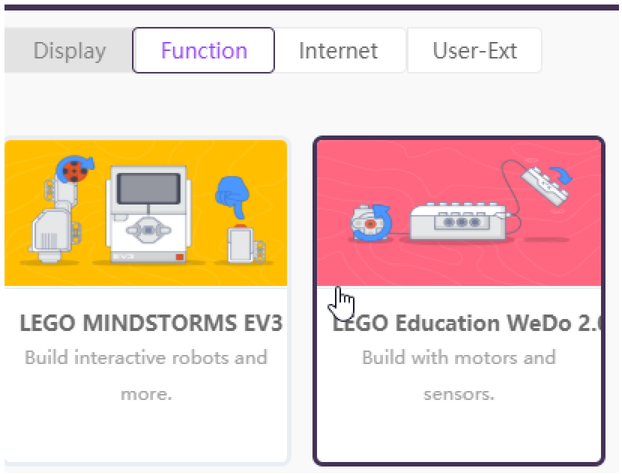
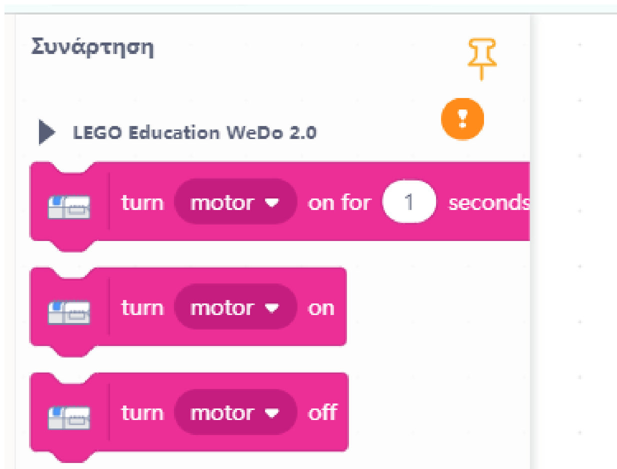
First launch of mind+ software

At the first launch the program will be in another language, whenever to change it click on the top right gear.	
Select Language (Greek).	
Select Online mode	

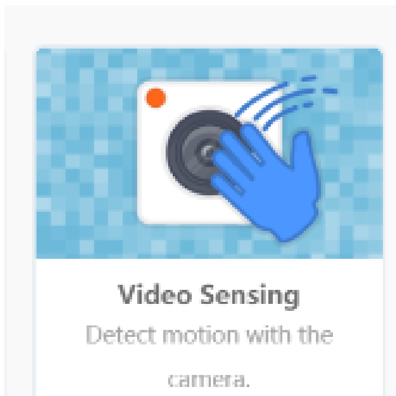
micro:bit connection

• Make sure the Scratch-Link program you installed earlier is running. • Open the mind+ program. • Click on the "Add Extension" button at the bottom left of the window. • Click on the box that says micro:bit. • Go back and while you have connected the micro:bit board with the USB cable press connect device. • Select the micro:bit board.	

“Introduction to Automation” Connection

Click on the “Add Extension” button at the bottom left of the window. Open the “Function” tab and select “LEGO WeDo 2.0”.	
Press the exclamation mark and then connect to the WeDo hub.	

Camera Connection

Click on the “Add Extension” button at the bottom left of the window. Open the “Function” tab and select “Video Sensing”.	

Indicative procedure of conducting it *Competition*

Keywords

Club, **Code** (Team), **Kiosk**, **Awards** (final ranking or Special/Thematic), **Commission** (ranking or special thematic awards), **Group** (groups), (chronological)**Zone**, **Phase** (evaluation).

Indicative schedule of its implementation *Competition*

08:00-09:30	Registration
08:00-10:30	Setting up and completing the preparation of the booths
10:30-13:00	First evaluation phase
13:00-13:30	Intermission for the teams that did not qualify to leave
13:30-15:00	Second evaluation phase
15:00-15:10	Announcement of results
15:10-16:00	Withdrawal of teams

A) Registration

On the day of the competition, the teams come to the reception area (registration) to complete their registration.

After registration, a folder with the contestants' cards and the **group code is given,** which they will have throughout the competition. This code is the team's identifier for the entire duration of the competition and cannot be changed. The students must write this code on their cards in case they are asked.

B) Stand installation

The group code corresponds to booth/kiosk **numbers** in which the team will exhibit their work. Any change of position (e.g. for reasons of accessibility for the disabled) must be done with the agreement of the President of the Organizing Committee, who will be contacted by the team coach. The team code will remain the same even if there is a change in the team stand.

Teams have the ability to install their project and do the necessary settings **and tests** until the evaluation process begins.

C) Evaluation process

The evaluation process is carried out according to the indicative criteria described in the relevant announcement of the competition.

The times of the assessment

A limited amount of time will be allocated to each group for the judging of the projects - seven minutes as an indication, of which a part (e.g. five minutes) will be for the presentation by the group and the remaining time for questions from the judges.

The duration:

- for the first assessment phase it is approximately 150 minutes,
- for the second assessment phase it is approximately 90 minutes,

After the completion of the first evaluation phase, the teams that qualify for the next phase are announced and the rest of the teams can leave.

The students of one team, after being evaluated by the judging panels, can visit the stands of other teams and see the works presented by their fellow students. But there should always be a representative at their booth who can demonstrate their work to visitors.

Students must be in their booth when the next assessment phase begins.

First Evaluation Phase

The teams are numbered (the team code) from 1 to 70. The judges are divided into 7 committees and each committee will see 10 teams. Specifically, committee No. 1 will see all the projects of the groups from 1, adding for each subsequent one 7, i.e. groups 1, 8, 15, 22, etc., committee No. 2 will see all the projects of the groups that have a number group 2, 9, 16, 23 and so on, so we continue for the rest of the committees. In order to avoid overcrowding of the committees, committee No. 1 starts the evaluation from group 1, committee No. 2 starts the evaluation from group 9, committee No. 3 starts the evaluation from group 17, and so on.

A team, if it is not ready when the panel comes to assess it, may ask the panel to pass later, provided the panel deems it possible (e.g. in terms of time).

Each committee prequalifies four of the ten teams it examines, resulting in a total of 28 teams for the next evaluation phase. There is flexibility if some committees consider that they should exceptionally qualify three or five teams, provided that the number of total teams does not change significantly.

Second Evaluation Phase

In the second phase of evaluation, there are four committees and their members result from mixing the members of the committees from the previous phase. So for example the new committee No1 will have members from the previous committees No1, No3 and No5, the new committee No2 will have members from the previous committees No2, No4 and No6 and so on. This is so that almost none of the members of a committee have reviewed the same teams, but also that each member of the committee has a more holistic view of the contestants by seeing as many teams as possible. Each committee will see six or seven teams from which they will advance three. Thus, 11-15 teams qualify for the final of the Pilot Competition.

The dual nature of educational robotics projects

According to Komis (Komis et al. 2017), central concepts of educational robotics are the "construction" and "programming" of robots. These concepts are part of the pedagogical stream of Constructivism, which claims that learning consists in the organization of the internal representations and experiences of the individual (Piaget, 1974). The ideas "Learning by making" and "Learning about making" gave the impetus to the foundation of Papert's "Constructionism" approach (Papert, 2000), which aims to form a framework for utilizing ICT in the educational process , capable of causing substantial changes in the way teachers teach and students learn (Ackermann, 2001), as the student from a passive receiver of information becomes an active subject, who investigates and processes what he perceives with his senses and creates knowledge (Frangou, 2005).

The artificial structure created by the students leverages a set of building materials, having sensors to capture events, an actuator that sets the whole structure or a part of it in motion. But the operation of the structure is based on the software that determines the behaviors of the structure such as detecting a stimulus and reacting by activating a motor. So in the process of developing the physical structure and its control program, there is a two-way relationship. The design of a physical structure (hardware) as well as its mode of operation (software) presupposes the interconnection of these processes. It is considered particularly effective that on the one hand the students with the role of programmers have understood the development model of the construction and on the other hand the students with the role of builders are involved to have built the mental model of the operation of the construction. This dialectical relationship between the development of the construction and the programming of its operation will lead to constructions (at least in the initial stage) that will be possible to program effectively considering that "the programming of robotic constructions has a peculiarity with respect to the programming in other conditions or situations. It is identified with the performance of behavior in an artificial construction" (Tsovolas & Komis, 2008).

The pedagogical dimension of Scratch

The **Scratch** (http://scratch.mit.edu), was created by the Lifelong Kindergarten Group, as a multimedia-rich system for novice developers. It is a visual **programming language** which allows users to program via graphic tiles instead of text and is integrated into a dedicated programming environment. This allows the user-programmer to see the results of executing a program immediately, while at the same time overcoming the limitation of syntax difficulties that exist in traditional programming languages.

A common didactic **metaphor** used to understand the developing of a project in Scratch, is that of producing a theatrical **play.** Because they both use:

- a play-**scene** which is the space where the play takes place and in Scratch there is a 2D space corresponding to the scene at background of which various static scenery is shown,
- **actors** (objects) which are dressed in different clothing,
- **character roles** (sprites) which the actors play when they appear on stage,
- **one scenario** followed by the actor roles. In Scratch the behavior of roles is defined by pieces of code called scenarios. THE **director** and **screenwriter** coexist in developer of Scratch and
- **audience** which in Scratch correspond to the users of the program.

In Scratch event-driven **programming** and implemented combining **command-tiles** (a metaphor from the Lego bricks) that fit together, to define the behavior of the roles (sprites) of two-dimensional objects that "live" in a specific space (scene-stage). A command-tile in Scratch follows a color **coding** which refers to / corresponds to the class of commands to which it belongs. Developing programming in Scratch allows students to create and develop programs related to animation**, narratives and games**, which can broaden the understanding of computational **concepts** as well as of computing **practices**.

Scratch, as a programming language, is the medium communication between intelligent entities, of Man **with the computing Machine**. Communication takes place at the level of the lower mental entity, the task of the higher entity is to understand the "culture" of the other entity, analyze its thinking and simplify it. According to Papert ``to get the computer to do something, you have to describe the relevant process with enough precision that it can be executed by the machine…**by teaching the computer how to think, children embark on an exploration of their own way of thinking** and thinking about thinking makes the child an epistemologist.'

For teaching programming, they can be used in microcosms which are environments suitable for learning programming. As microcosms are considered small but relatively complete subsets of representation of the "real" environment, which represent a part of the theoretical "world" and can be understood in an observational, experiential and exploratory way. Although microcosms are directly aimed at skill development, problem **solving** and algorithmic way**(convergent) thinking**, under conditions can be used as educational tools for development creative **(divergent) thinking**. In such environments it may no longer seem contradictory to ask children to work creatively by addressing them in non-creative ways. A programming environment is one nursery **of potential innovations**, in which the student can accumulate experience cultivating his algorithmic thinking. Under scaffolding conditions the teacher can guide the student to think creatively outside the box and produce something "innovative" for him.

Scratch supports a variety programming **styles** such as serial programming (with elements structured **/ departmental and hierarchical programming**), The Parallel **programming**, event-driven programming (**event driven programming**) and object-oriented programming (**object based programming**), while using clones features object-oriented elements (**object oriented programming**) programming. In Scratch, the possibility of visual programming with tiles combined with the variety of programming styles it can support and in light of the pedagogical approach of “**emerging literacy**”, making it an ideal tool for novice developers with limited programming background, offering them an accessible starting point for learning.