Augmented Reality enhanced universal Educational Robotics framework

Authors: Mikhail Fominykh, Dimitris Karampatzakis, Ioannis Kazanidis, Petros Amanatidis, and Avgoustos Tsinakos

Educational Robotics at Schools Online with Augmented Reality - eROBSON project has received funding from the European Union's Erasmus Plus programme, grant agreement 2020-1-NO01-KA226-SCH-094120.

eROBSON Consortium | March 2024

White paper

Augmented Reality enhanced universal Educational Robotics framework

Authors

Mikhail Fominykh, Dimitris Karampatzakis, Ioannis Kazanidis, Petros Amanatidis, and Avgoustos Tsinakos

Acknowledgement

Educational Robotics at Schools Online with Augmented Reality - eROBSON project has received funding from the European Union's Erasmus Plus programme, grant agreement 2020-1-NO01-KA226-SCH-094120.

Disclaimer

Co-funded by the Erasmus+ Programme of the European Union

License

Cite as

Mikhail Fominykh, Dimitris Karampatzakis, Ioannis Kazanidis, Petros Amanatidis, and Avgoustos Tsinakos: Augmented Reality enhanced universal Educational Robotics framework, a white paper (2024). eROBSON Consortium. https://e-robson.eu/

Summary

The augmented reality enhanced universal educational robotics framework describes how Augmented Reality technology can be used to enhance learning with Educational Robotics. The purpose of Augmented Reality in this framework is to simulate (a) Educational Robotics equipment / Educational Robotics kit components, including its appearance, behavior, and user interaction and (b) some of the Educational Robotics learning processes and methods. Conceptually, the framework and its blocks were made platform- and technology-independent.

There are many reasons for using Augmented Reality to enhance learning with Educational Robotics. Augmented Reality can simulate Educational Robotics equipment and processes, making it possible to engage with Educational Robotics remotely and without the physical Educational Robotics kits, while keeping the educational experience hands-on and tangible. The motivation for the conceptual framework presented in this document has been to enable delivering Educational Robotics and other laboratory-dependent activities in the context of pandemic restrictions.

We explore the possibility of delivering laboratory-dependent courses and activities, specifically for Educational Robotics, with potentially low implementation cost, good adoption prospects, and of zero risk of transmitting COVID-19 or other viruses.

Basics

Augmented Reality (AR) enhances human perception with additional, computer-generated sensorial input to create a new user experience[1]. Most AR systems provide visual experience, blending digital graphics with the features of the real physical space around the user. AR experience can be delivered to the users on mobile devices, smartphones and tablets, and on specialized smart glasses. AR applications for these devices use the camera of the device and other sensors to detect and track some features of the physical space around the user and use them as anchor points for the digital content.

Educational robotics (ER) teaches the design, analysis, application and operation of robots. Robots include articulated robots, mobile robots or autonomous vehicles. Educational robotics can be taught from elementary school to graduate programs. Robotics may also be used to motivate and facilitate the instruction other, often foundational, topics such as computer programming, artificial intelligence or engineering design[2].

Conceptual ideas

Developing the augmented reality enhanced universal educational robotics framework (AR-uER framework), we considered several principal ideas that influenced our decisions and designs.

- ER is an educational approach, so the conceptual framework should be suitable for designing an educational system.
- AR can be used to simulate the work with ER platforms.
- The conceptual framework should be independent of ER platforms and AR platforms.
- The conceptual framework should operate general principles of ER that are common to multiple ER platforms.
- ER platforms contain programmable and non-programmable components.
- The conceptual framework should operate general principles of AR that are common to multiple AR apps/platforms.
- The conceptual framework should consider two major ER approaches, the first coming from education researchers (designed Lego Mindstorms, Cricket) and the other from interaction designers (Arduino), focusing on the selective exposure of the ER kit technology[3].
- AR provides a possibility for visual simulation of ER platform components with a certain level of realism and abstraction.
- AR provides a possibility for behavior simulation of ER platform components, making them realistically react to:
    - manipulation by the user, translation, rotation, connecting to and disconnecting from each other, interacting with controllers, buttons and other elements
    - environment factors, such as gravity, collision with physical objects, light, sounds, pressure
    - other digital objects, including other ER platform components, such as changing their states based on active connections to other components
    - programmed logic

---

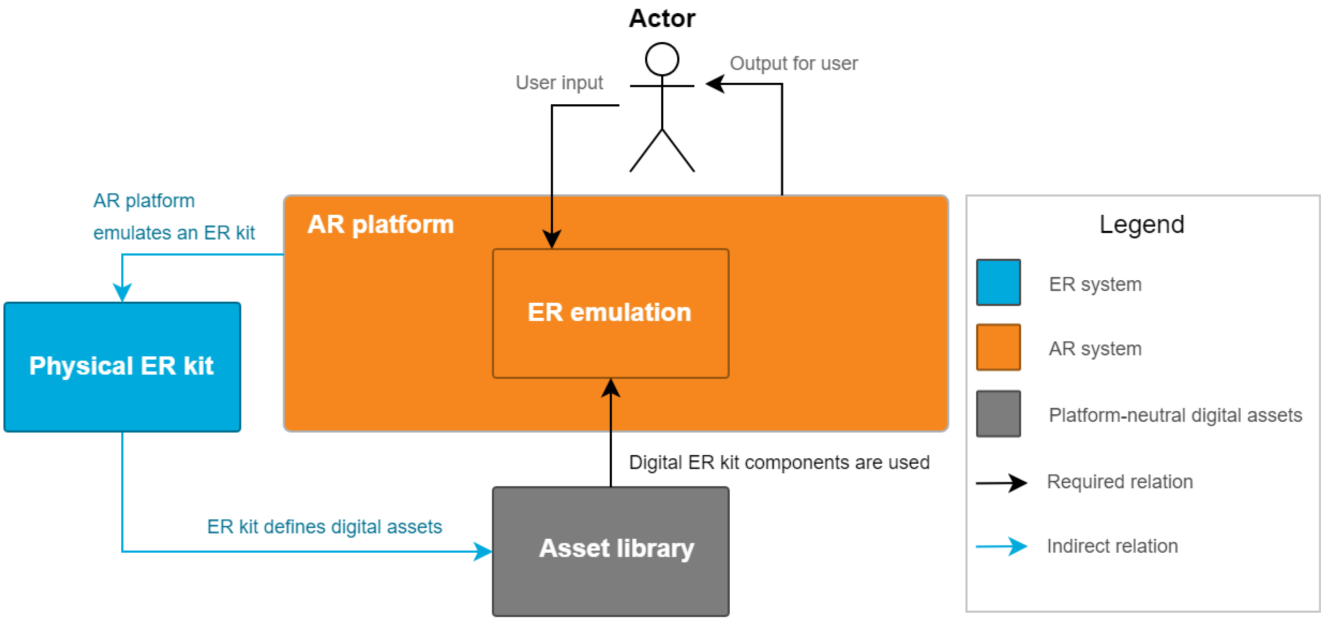
[1] Akçayır M. & Akçayır G. (2017) Advantages and challenges associated with augmented reality for education: A systematic review of the literature. Educational Research Review. https://doi.org/10.1016/j.edurev.2016.11.002

[2] P. De Cristoforis, S. Pedre, M. Nitsche, T. Fischer, F. Pessacg and C. Di Pietro (2013) A Behavior-Based Approach for Educational Robotics Activities, in IEEE Transactions on Education, https://doi.org/10.1109/TE.2012.2220359

[3] Paulo Blikstein (2015), Computationally Enhanced Toolkits for Children: Historical Review and a Framework for Future Design, https://doi.org/10.1561/1100000057

High-level AR-uER diagram

The high-level AR-uER conceptual diagram consists of three blocks and shows the user interaction with the system.



The blue *Physical ER kit* block represents an ER system both as a blueprint for designing digital ER assets and as a target for emulation by an AR system. The *Physical ER kit* is a collection of ER kit components, each with its own behavior defined in the kit. The user is not interacting with the Physical ER kit, as the AR platform aims to replace it.

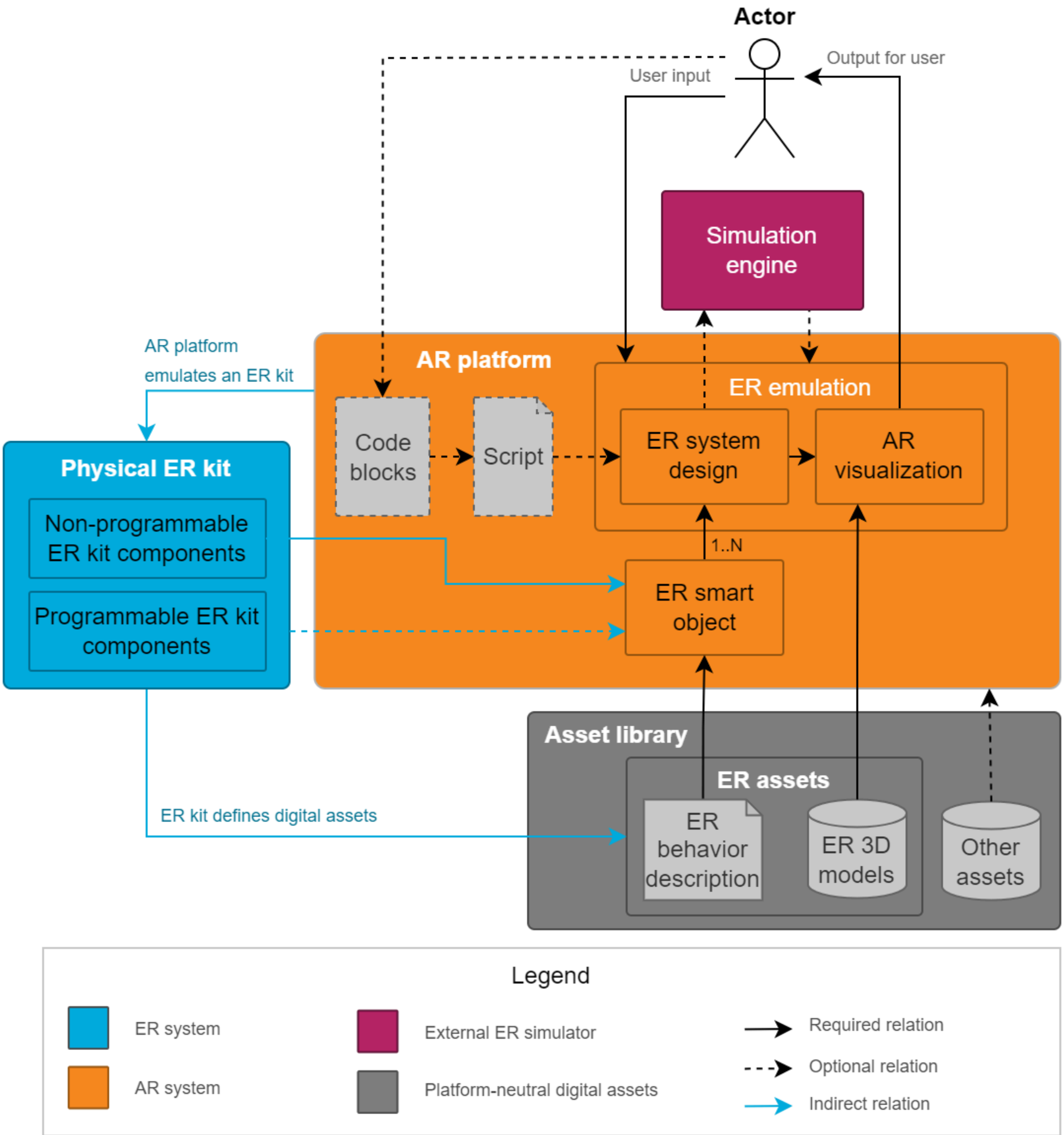
The gray *Asset library* block represents a collection of digital assets that are required by any AR system to simulate an ER kit, both visually with 3D models and logically with behavior logic descriptions. Each ER kit that is simulated in AR requires its own assets.

The orange *AR platform* block represents the design of an AR system that can emulate an ER kit. The *ER platform* contains the *ER emulation* block. The ER emulator receives input from the user in the form of connecting, disconnecting, providing input values, and changing the state of the simulated ER components.

Principal AR-uER diagram

The AR-uER framework diagram consists of four main blocks: Physical ER kit, Asset library, AR platform, and Simulation engine. The diagram also displays the user and his/her interaction with the system.

The AR-uER framework diagram has been first presented and described in a publication by Karampatzakis et al[4].



[4] Dimitris Karampatzakis, Mikhail Fominykh, Nardie Fanchamps, Olga Firssova, Petros Amanatidis, Giel Van Lankveld, Thomas Lagkas, Avgoustos Tsinakos, and Roland Klemke: "Educational Robotics at Schools Online with Augmented Reality" in *the 15th IEEE Global Engineering Education Conference 2024 (EDUCON2024)*, EDUCION2024 Proceedings.

Physical Educational Robotics kit

The blue *Physical ER kit* block represents an ER system both as a blueprint for designing digital ER assets and as a target for emulation by an AR system. The *Physical ER kit* is a collection of ER kit components, each with its own behavior defined in the kit. We categorize them to *Non-programmable ER kit components* and *Programmable ER kit components*. Components in both categories define the corresponding ER smart objects created inside the AR platform. The user is not interacting with the Physical ER kit, as the AR platform aims to replace it in the context of remote learning.

Asset library

The gray *Asset library* block represents a platform-independent collection of digital assets that are required by any AR system to simulate an ER kit. Two essential elements of the Asset library are grouped as *ER assets*. These assets are designed after the Physical ER kit components, applying the same level of selective exposure. The *ER behavior descriptions* defines the logic for the behavior of each ER component simulated in the AR system, related to selective exposure for usability (or the embedded error correction). In different ER kits, depending on the Maker's Box approach (Black or White), this may include different features of the ER kit's input/output ports and standardization, tangible user interface buttons/switches, color-coded or text labeling visual information, displays, and power systems. The *ER 3D models* define the visual appearance of the ER component simulated in the AR system. The 3D models of ER components should have some physical properties that can be utilized in AR interaction, such as movable parts (e.g., buttons, motors, flexible wires), output indicators (e.g., light and sound sources), and input reactors (e.g., collision detection). Finally, the Asset library may contain *Other assets* that might be required to create an educational AR experience, such as poster images, video instructions, 3D models of objects other than ER components, and similar.

Augmented Reality platform

The orange *AR platform* block represents the design of an AR system that can emulate an ER kit. The AR platform should contain *ER smart objects* that implement the behavior of the corresponding components of the Physical ER kit (e.g., standardized ports, interaction mechanics). When using the AR system, the user should be able to assemble multiple smart objects in real time, limited by the selective exposure for usability implemented in the ER smart objects. Any of such configurations is logically represented by the *ER system design*. It includes the list of utilized ER components and their structure with circuitry, mechanics, and an optional programmable logic. The *AR visualization* block displays the ER components to the user which are arranged according to the current ER system design. The block visualizes the ER components using the ER 3D models from the *Asset library* and places them in an AR space, seamlessly integrating the digital content into the physical space around the user. The *ER system design* and the *AR visualization* constitute the *ER emulation* block. The ER emulator receives input from the user in the form of connecting, disconnecting, providing input values, and changing the state of the ER components.

Optionally, the AR platform can work with programmable ER kit components. They require *Scripts* to perform their functions. The Scripts can be written directly by the user or engineered with *Code blocks*, like when working with the programmable components of the Physical ER kits. The AR platform can either support scripting or import scripts created elsewhere as text files.

Simulation engine

The magenta *Simulation engine* block inputs the logic of the ER system design with every change made by the user, simulates the work of the ER system, and returns its state to the ER emulation inside the AR platform. In this context, the integration of a simulator is crucial for the AR emulation, and conceptually, we model it as an external system interacting bidirectionally with the AR platform. This simulator could be, for example, an open Arduino simulator or an open Python-based simulation tool.

Actor

The *Actor* block represents the user of the system who perceives the AR visualization and provides input for the ER emulation by interaction. The *Actor* can also input *Code blocks* and *Scripts* if *Programmable ER kit components* are used.

Reference implementation

The AR-uER formwork was applied to two AR platforms: ARTutor[5] and MirageXR[6]. The functionality and services provided by these platforms, coupled with the AR-uER building blocks, equip teachers with useful tools to prepare learning content for blended and online teaching. These platforms employ different workflows to create AR-ER educational material.

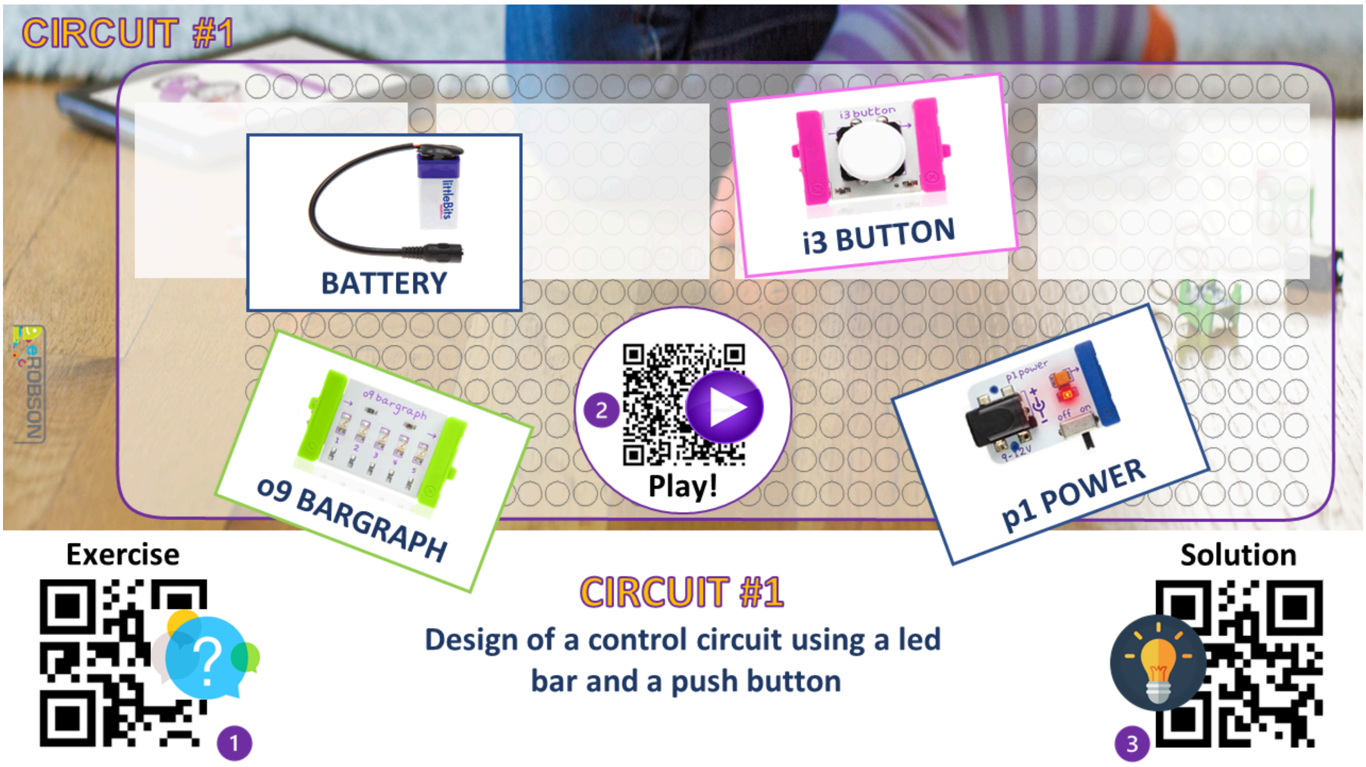
ARTutor

ARTutor is an AR platform designed for widespread use by educators and students. The ARTutor web editor offers a content authoring environment that allows teachers with limited computer skills to create AR-enhanced digital materials quickly. The ARTutor mobile application plays the AR content in an easy way for the students.

ARTutor is using the Book metaphor. Each digital material is treated as a book, and the user can enhance it using different types of AR content such as images, sounds, videos, 3D models, and web links. The web editor generates a QR code for each AR-enhanced book that serves as its unique identifier. The students can access all associated materials by simply scanning the QR code using the mobile app.

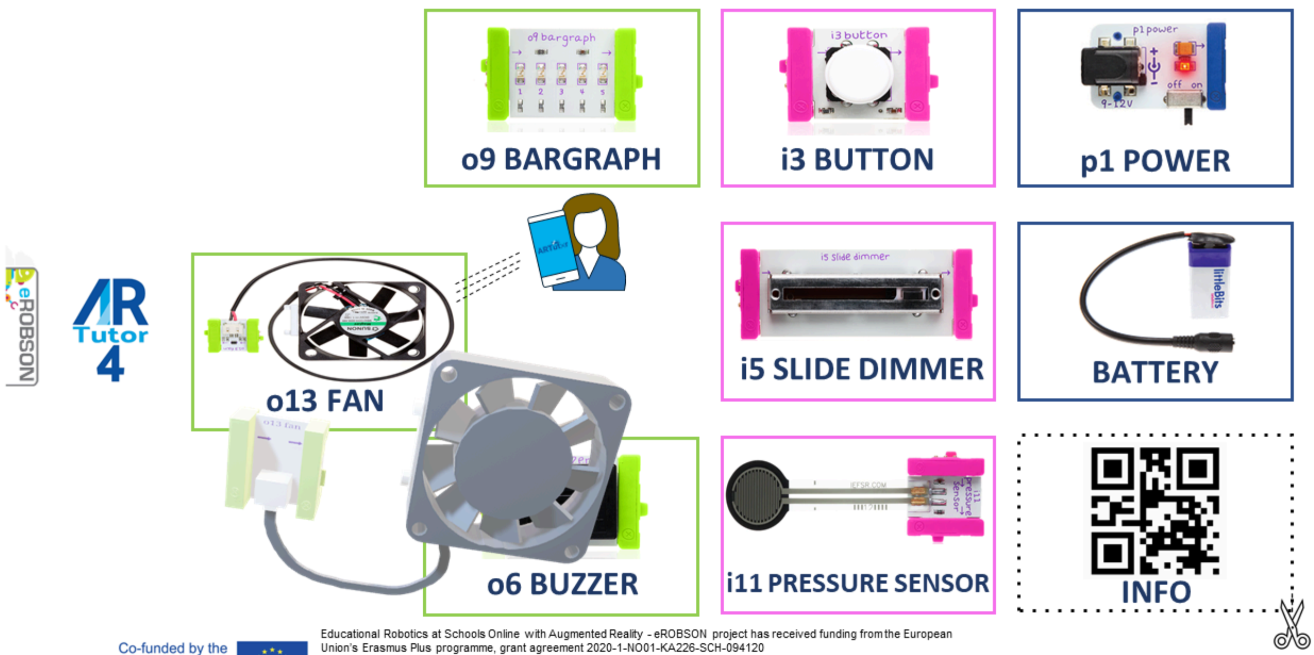
In an ARTutor book, marker-based AR detects predefined image markers and tracks them in real time to provide such digital content anchors. In the figure below, we illustrate paper cards with image markers representing ER components.

[5] ARTutor https://artutor.ihu.gr/home/
[6] MirageXR https://github.com/WEKIT-ECS/MIRAGE-XR



We developed educational material for the littleBits ER platform using ARTutor. The materials comprise an introductory course covering the basics of the Internet of Things. The course consists of 10 lessons/circuits in paper-based material and an ARTutor AR Book. The printable material is the main cardboard and the library of available modules (Bits), as presented in figure above. The student scans with the ARTutor mobile app the generated QR code. The material can be sent to students in PDF format. They can then begin by printing and cutting out the library cards, placing them on cardboard. Students can download the Book using the QR code and interact with the cards to access AR content.

LIBRARY OF AVAILABLE LITTLEBITS



Co-funded by the Erasmus+ Programme of the European Union

Educational Robotics at Schools Online with Augmented Reality - eROBSON project has received funding from the European Union's Erasmus Plus programme, grant agreement 2020-1-NO01-KA226-SCH-094120
https://erasmus-plus.ec.europa.eu/projects/search/details/2020-1-NO01-KA226-SCH-094120
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

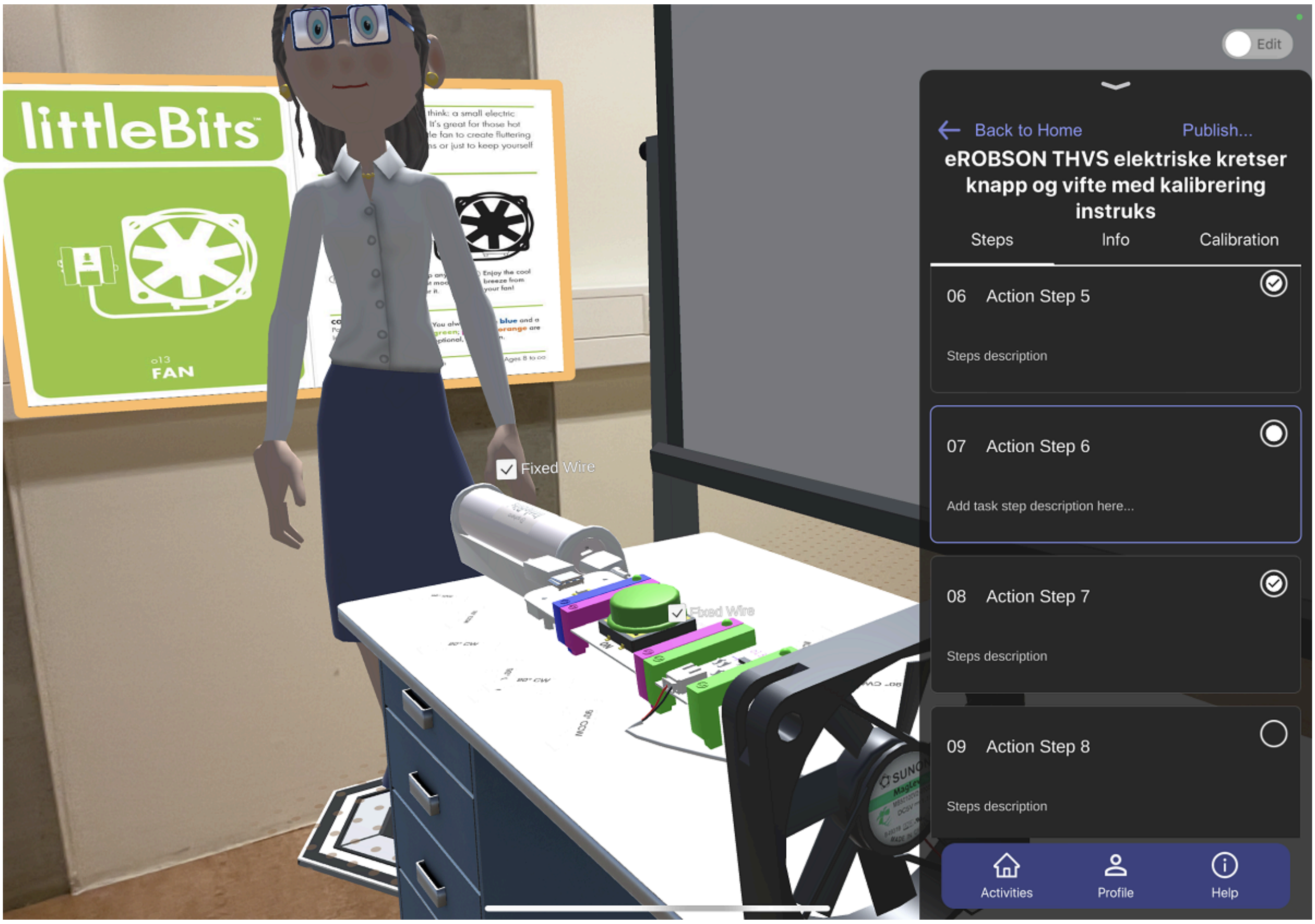
Students need to follow three steps to complete each lesson. The initial step is the Exercise marker, which provides instructional guidance through a video. The next step is the Play marker, utilizing LearningApps quizzes to examine Bit module placement on the cardboard, and the last step is the Solution marker featuring a video of the actual circuit in operation.

MirageXR

MirageXR is an educational AR platform that enables experts and learners to share experiences. It uses the marker-less approach building a so-called spatial map of the physical space around the user and uses its features as anchors for the digital content.

MirageXR supports in-situ authoring and experience capture, allowing to place media content in an AR space and to create content in real time while performing tasks. The content designed in MirageXR can be shared with students via a learning management system. MirageXR is a cross-platform tool, supporting HoloLens 2 AR glasses and smart phones with iOS and Android. The same application is required for both authoring and playing the AR content.

The learning content created in MirageXR is organized as AR activities with multiple steps, supporting procedural training and operational support for complex tasks. In a figure below, we illustrate how AR-simulated ER components appear to the students in MirageXR.



In MirageXR, we developed a new content type to support the simulation of the littleBits ER platform. When authoring content, a teacher can add a littleBits component to a step in the learning activity and place it in the AR space. We implemented 13 most common littleBits components. Each of them is represented by a 3D model, designed after the physical littleBits kit. We simulated components that consist of multiple parts connected by flexible wires. We also implemented interactive input and output components. For example, the button component can be pressed ON and OFF, the slide dimmer component can be slid left and right, while the lamp component can light up and the fan component can spin. Each component can also be rotated along one axis so that the users could attempt to connect littleBits components by the wrong side, using the wrong ports.

Each component has ports and can be connected to other components, following the logic of the littleBits kit. When two or more components are connected together, MirageXR will check the circuitry and change the power state of each component. The behavior of the output components takes into consideration the input values. For example, depending on the value the user sets on the slide dimmer, the light component in the same circuit will shine brighter or darker. Finally, we designed an indicator for each component that lights up when the component has power as a form of feedback to the user, but that does not exist in the physical littleBits kit.

MirageXR supports several ER learning scenarios. For example, AR activities can be built to introduce electrical circuits, exemplified by the littleBits components. In such a scenario, the learners can observe how they work and interact with the interactable components. The activities can also contain problems. In such scenarios, the teacher must formulate the problem using non-ER content types either voice, text or a human character. The teacher must also add and connect the necessary ER components in the same step of the AR activity. In the play mode of the application, the littleBits components will appear disconnected when the activity step is started. The app gives an indicator when a student connects the littleBits components correctly, as designed by the teacher.