Multiplatform educational robotics course to introduce children in robotics

Pedro Plaza
Departamento de Ingeniería Eléctrica, Electrónica y Control (DIEEC) (UNED)
Madrid, España
pplaza@plazarobotica.es

German Carro
Departamento de Ingeniería Eléctrica, Electrónica y Control (DIEEC) (UNED)
Madrid, España
germancf@ieee.org

Manuel Blazquez
Departamento de Ingeniería Eléctrica, Electrónica y Control (DIEEC) (UNED)
Madrid, España
manuel.blazquez.merino@gmail.com

Elio Sancristobal
Departamento de Ingeniería Eléctrica, Electrónica y Control (DIEEC) (UNED)
Madrid, España
elio@ieec.uned.es

Manuel Castro
Departamento de Ingeniería Eléctrica, Electrónica y Control (DIEEC) (UNED)
Madrid, España
mcastro@ieec.uned.es

Félix García-Loro
Departamento de Ingeniería Eléctrica, Electrónica y Control (DIEEC) (UNED)
Madrid, España
fgarcialoro@ieec.uned.es

Abstract*—Robotics and computational thought are ideal tools for developing science, technology, engineering and mathematics (STEM) pedagogy. Throughout this paper a modular and adaptive course is presented, the main objective of which is to make known simple and economic tools of educational robotics. This course is aimed at those who want to discover the possibilities of educational robotics in the context of the introduction to robotics. Today, robotics training tools are raised with the aim of promoting innovation and motivation of students during the learning process. Robots are becoming more and more common in our daily lives; therefore, it is important to integrate robots into all levels of our society. The course is designed to work with the Scratch, Crumble and Arduino tools as STEM enhancers. Using Scratch, interactive stories, games and animations can be programmed. Scratch helps young people to acquire and improve skills such as think creatively, think systematically, and work collaboratively. Scratch is a project of MIT Media Lab's Lifelong Kindergarten Group. It is offered free of charge. On the other hand, Crumble is an easy-to-use programmable controller. Its programming interface uses a block programming language based on Scratch that makes it easy for children from 10 years old to use it. In addition, the hardware elements associated with Crumble are very intuitive and easy to connect. Last, but not least, Arduino is an open source electronic platform based on hardware and software that is easy to use. It is a platform that incorporates a simple microcontroller and an interface development environment to create the applications to be downloaded on the board. The course offers a three-tiered journey through three levels with each of the three tools. It consists of a total of 9 modules. This course has a very practical approach. A project-based pedagogical methodology is used. Experiments are promoted, where trial and error are part of learning and self-discovery. The student learns to have more autonomy and responsibility. Knowledge is acquired in different disciplines. It develops: motor skills (scale mobility in the hands), group skills, allowing people to socialize, creative abilities, and learning in a fun way. The operational details, materials used and examples of activities for some modules are also presented with the expectation that all teachers will be able to adapt these activities in their class. In addition, results are included from several groups of students who have already completed some modules. Despite not having a large number of students, the experience provided results that may be useful for other teachers to promote a course with similar or equal content for more results. The results of this work show that it is important to combine theory and practice to include fun tasks intertwined with the challenges of applying theory to problem solving.

Keywords—education; programming; robotics; STEM

I. Introduction

Robotics and computational thinking are ideal tools for developing science, technology, engineering and mathematics (STEM) pedagogy. This is because robotics integrates with areas of knowledge such as mechanics, electricity, electronics and computer science. In addition, robotics has been one of the main drivers of the modernization and continuous improvement of most processes for several decades now [1].

But what is robotics? The creator of the term "robotics" was Isaac Asimov in its 1941 history. Although, as Asimov himself later acknowledged in Gold, he thought he was using an existing word. "The Robot Chronicles." One such story is the well-known book "I, Robot". The image of the robot that appears in his work is that of a well-designed machine with a guaranteed safety that acts in accordance with the three laws of robotics.

From 1941 to the present day, robotics has become a benchmark for science and technology. Today, robotics is a key part of modernizing and improving most processes. This is due to the ease with which robots can be integrated into industrial processes [2]. The author in [3] presents the development of an active ankle-foot orthosis (AFO) to improve walking ability. Another example of robotic implementation is the robot with light, elastic legs that uses surface tension to stay on the surface of the water described in [4].

As noted, robotics is being included as a powerful tool to encourage students to access science, technology, engineering and mathematics (STEM) subjects. STEM education has great potential and its popularity is growing day by day [5]. Primary education is confronted with the development of STEM thinking and STEM-friendly attitudes in early learners. Reference [6] describes design, implementation and evaluation as a solution to address both problems. Reference [7] provides vision and guidance for the future on many interesting aspects: innovative teaching methods and tools (including assessment), community aspects, aspects of curriculum design and aspects of instructional design that consider the costs of UMI (ubiquitous computing, mobile computing and the Internet of Things) to improve STEM education. Educational technology is a powerful tool that is increasing its presence day by day. Some examples are those provided by [8] as part of the eMadrid network.

However, the introduction of robotics as a learning process is not a trivial matter. Robotics combines areas of knowledge such as mechanics, electricity, electronics and computing. Robots are machines capable of making decisions and adapting to different situations. They are usually built with elements such as sensors, actuators and process units [2].

Throughout this paper we present a modular and adaptive course, whose main objective is to introduce simple and economical tools of educational robotics. This course is aimed at those who want to discover the possibilities of educational robotics in the context of the introduction to robotics. Today, robotics training tools are designed to promote innovation and motivation of students during the learning process. Robots are becoming increasingly common in our daily lives; therefore, it is important to integrate robots into all levels of our society.

The tools chosen for the proposed course are Scratch, Crumble and Arduino. Scratch is presented as the simplest and most user-friendly tool for programming and robotics. Authors in [9] presents the use of Scratch - a widely used tool - as a tool to guide students in the acquisition of robotics skills. This paper shows Scratch as a first step in introducing students to robotics. The reference [10] shows the opportunities that Scratch offers for use in the training of future designers of the worlds offered by computing. Thus, it aims to turn consumers of computing resources into explorers and designers of the worlds offered by computing. In research such as that shown in [11] the authors examine predictive analysis of users' written communication through commentary in an open online social networking forum at Scratch.mit.edu. Reference [12] details a comparative study to investigate any differences in the transition of students' motivation to learn programming using Scratch and App Inventor for Android in K-12 educational environments.

Crumble is the tool of choice, as the programming interface is visual, Scratch-based, and it makes programming tasks easier for students as they delve into the hardware elements that are introduced with this tool.

Despite being an emerging tool, the use of Crumble is beginning to spread as a tool that promotes STEM knowledge learning. Due to its reduced cost and simplicity of use, it is not only limited to use in classrooms or private educational centers, but also allows for its use at home [13].

Last but not least, Arduino is used. This tool is the most complex of the three, but it is also the tool that allows more development. In the field of digital design, Arduino-based environments are used as the main platform in courses on microcontrollers and advanced digital design techniques. Thanks to the use of Arduino students showed better design skills and motivation compared to students in past courses [14]. Arduino based developments, in part due to their low cost, ease of use, flexibility and wide adoption in both consumer and industrial applications are being considered in education. The reference [15] shows as an example the transformation of automotive laboratories using Arduino shields. The authors in [16] show the new possibilities in education provided by the Bring Your Own Device (BOYD) policies adopted at different stages of learning. This is easy to implement with devices such as Arduino, Arduino shields and visual software (Visual Basic and Scratch) for educational purposes. A very important component of engineering is that the knowledge acquired in the classroom can be applied in practice. In the reference [17] the authors review concepts such as the application of practical teaching laboratories to the technological areas related to the project. They also include in their study the use of the Arduino and Android platforms. Arduino features support for tools such as LabVIEW for students participating in Virtual Measurement and Instrumentation programming courses [18]. There are also developments such as the haptic controller for the Arduino based educational environment. This platform allows students to implement haptic algorithms in multiple mechatronic devices [19].

This paper consists of four sections. Section II describes the course from the point of view of the tools used and the levels of difficulty involved. Section III details an example of educational material that can be used for the development of the course. Section IV includes the results obtained from the deployment of the first module using Scratch and a basic difficulty level. The last section contains the conclusions of the material presented.

II. COURSE DESCRIPTION

The context of this course is the introduction to robotics for boys and girls. Given the context of the course and the target audience, no prior knowledge or specific skills are required. This course is designed to be started by children of 6 years of age and depending on the development of skills and the acquisition of competencies, students can program through the different modules that make up the course.

This course is based on three tools widely used in teaching. The tools are characterized by two main parameters: the degree of simplicity of their use and the degree of complexity of the applications that can be made with them.

On the other hand, the course consists of three levels of difficulty. A first level is very simple and has as its main objective to be a contact of the students with the tool in question. This is followed by an intermediate level, and finally a level of greater complexity with which it is intended that students can develop complex applications in the field of the tool used and in the framework of educational robotics.

Figure 1 shows the set of matrix modules placed by type of tool used and level of difficulty.

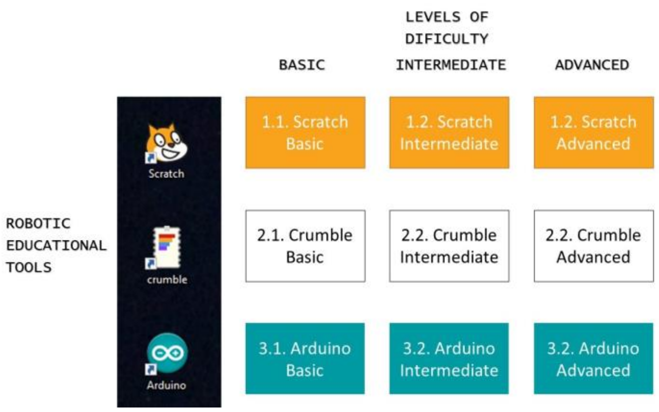

Fig. 1. Course matrix.

For all the above reasons, the course has been designed as a set of three difficulty modules with each of the three tools. From there a total of nine modules are obtained for the complete development of the course.

Each module is designed to be deployed in one-hour sessions on a weekly basis. Therefore, the deployment of each module can be balanced with the school terms. For example, if the school year begins in October and ends in December, a module can be planned to be delivered in four one-hour sessions for each of the months of October, November and December.

A. Robotic educatianl tools and resources

The first tool used in the course is Scratch. Scratch was developed by the Education Division of MIT (Massachusetts Institute of Technology). The mission of the Massachusetts Institute of Technology is to advance knowledge and educate students in science, technology and other scholarship areas that will best serve the nation and the world in the 21st century. Interactive stories, games and animations can be programmed with Scratch [20].

Scratch is provided free of charge, is very easy to use and is perfectly valid for ages 6 and up.

The main advantages of Scratch are:

- Block-based visual programming language. This facilitates access for children who have never programmed before, as well as for children who are not proficient in reading comprehension.
- Allows image editing and use of predefined images.
- It has a command categorization system. This allows children to quickly and easily find the command they need to use at any given time.

The disadvantages of this tool are the following:

- Only the peripherals of the computer can be used: the screen, the speakers, the microphone, the keyboard and the mouse.
- It does not allow the interaction of programs on one computer with programs on another computer.

The second tool used in the course is Crumble. Crumble is an electronic board to which up to two motors, lights, switches and sensors can be connected to control its operation. It connects to the PC via USB and can be programmed using the free software. They are manufactured by Redfern Electronics and some accessories are available that are compatible with the Crumble [21] card.

Crumble software is provided free of charge and the controller is priced at £10. It is very easy to use but incorporates hardware and mechanical components that limit the minimum age of the users. It is perfectly valid for ages 8 and up.

The Crumble controller is a card that is capable of handling motors, analog and digital sensors, Sparkles and servos.

The main advantages of Crumble are:

- Block-based visual programming language. The programming language is based on Scratch.
- Like Scratch, it has a command categorization system,
- Allows interaction of the controller board with different types of electrical and electronic devices.
- Allows the use of static and mobile robotic platforms. This feature is of utmost importance, as it promotes motivation and helps to gain students' attention.

The disadvantages of this tool are the following:

- The number of electrical and electronic devices with which the controller board can interact is limited. In addition, no more than six elements can be connected simultaneously, limiting the applications that can be developed with Crumble.
- The interaction of the programs of one controller with the programs of another controller is very limited due to the low number of general purpose input/output ports.

The third tool used in the course is Arduino. Arduino was launched by Massimo Banzi in 2005 as a modest tool for Banzi students at the Instituto de Diseño de Interacción Ivrea (IDII). Arduino is an easy-to-use hardware and software based open source electronic platform [22]. It is a platform that incorporates a simple microcontroller and an interface development environment to create the applications to be downloaded to the board. The use of Arduino projects covers a wide range of applications, from robotics to automatic control irrigation systems.

The Arduino software is provided free of charge the controller has a price of 20 €. It is very simple to use but requires textual programming. It also incorporates hardware and mechanical components that limit the minimum age of users. It is perfectly valid for ages 12 and up.

Arduino is a card that is capable of handling both analog and digital signals. It integrates a variety of communication protocols such as serial communication, SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit) and others. Including Arduino compatible shields, the controller can be used for any type of application, from motor control to the implementation of a robotic remote laboratory.

The main advantages of Arduino are:

- Allows interaction of the controller board with any type of electrical or electronic device.
- Allows the connection of a large number of electrical and electronic devices simultaneously.
- Allows connection of multiple controllers via wired or wireless communication.
- Allows the use of static and mobile robotic platforms.
- It is possible to implement any type of robotic application.

The disadvantages of this tool are the following:

- Textual programming language. Students will experience difficulties with the language syntax. In addition, the commands used must be known by heart or they will have to resort to examples or the reference guide.
- Given the wide variety of possibilities offered by Arduino, the complexity of the applications requires a great deal of time and it is necessary to define a stricter work methodology than with the previous tools.

B. Levels of dificulty

Three levels of difficulty have been defined for this course: basic, intermediate and advanced. The basic level aims to provide students with a first contact with the tool. It is also intended that students begin to know the different elements with which the tool can interact when implementing different applications. At this level, the applications developed will be very similar for each of the three tools. Secondly, the intermediate level already starts from the premise that students have acquired the skills of a basic use of the tool. Therefore, at this level, more in-depth use of the tool will be deepened, and more complex applications will be developed. At this level, the applications developed may have a degree of similarity, but they will have connotations specific to the tool used. Finally, through the Advanced level, students will work with more complex and tool-oriented applications they are using.

C. Course metrics and indicators

To assess whether a student has acquired the necessary skills and abilities, different metrics and indicators are proposed that should be used to know if a student is ready to progress to the next module. Two types of measurement tools are proposed: firstly, the analysis of the instructor throughout the sessions, and secondly, the use of control surveys at the beginning and end of each module.

The age of the students should be considered when making the transition from one module to another. This is important, especially because of the recommended minimum age for each of the tools.

It is also important to know the concerns of students. A student may complete the intermediate Scratch module and the natural transition would be to the advanced Scratch module. But if the student is interested in starting the Crumble module, it would be advisable to take this opinion into account when choosing the next module to be taken by the student.

D. Student progress

Given the structure of the course, it is possible to make many transitions between modules. Figure 2 shows different possible transitions. First, you could take a tour from basic advanced Arduino Scratch to all levels of difficulty and all the tools. This path is represented by Figure 2 as blue arrows marked with identifier 1. It is also possible, if the student's age is equal to or greater than the recommended age for each of the tools, for a student to attend the basic levels of each tool first, then the intermediate levels and finish the advanced levels of each of the tools. advanced Arduino Scratch to all levels of difficulty and all the tools. This path is represented by Figure 2 as green arrows marked with identifier 2.

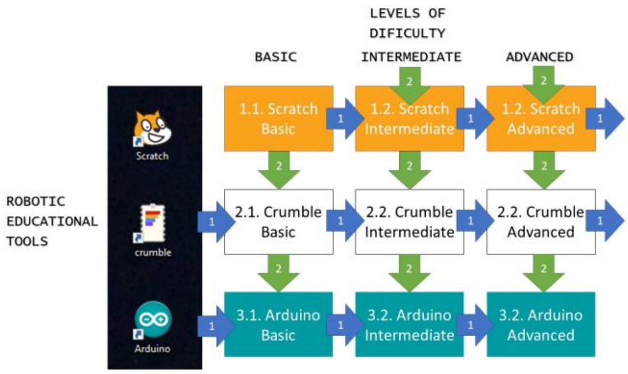

Fig. 2. Module transitions.

In addition, it is not a prerequisite to start at the basic level of Scratch. If a student has previous experience with programming and robotics or has an age and knowledge that tools like Scratch and Crumble are too trivial for him or her, this student could start with the basic Arduino module.

III. Example of Material

This section describes a cross-platform material oriented to the basic level. That is, the same content is used to be deployed with Scratch, Crumble and Arduino. The idea is to develop a material with very similar objectives, but with an implementation adapted to each of the tools. Throughout the basic level, three elements of real life will be dealt with, with which the students may be familiar. The first element is a traffic sign, the second element is a garage access barrier, and lastly, it will work a follower lines - navigation of mobile platforms using sensors. Each of the modules is divided into four parts. A first part to familiarize the students with the corresponding tool. A second part for them to work on the implementation of the traffic signal. A third part for students to develop the garage access barrier. And a last part for the line follower to be implemented by the students.

A. Scratch basic level

In the first part of this module, students are shown the structure of the development interface and its different elements. The panels that make up the development interface, command categorization, some basic commands, and the display panel are explained. At this point, we move on to activities in which the students have more participation. First, you work on programming without a computer. Students must define invented commands, define a list of them, and finally, execute that set of instructions. Then they must start programming with Scratch. The first challenge is to program the Scratch cat to move from left to right. The next exercise aims to get the Sprite to paint the background as it advances. Using the program generated so far, students must modify the program so that the Sprite draws a triangle and a square. To end this part, students must define a word, share it with others, and create a story with all the words. With that story defined, students must do a program with Scratch.

In the second part we work on implementation of a traffic signal with Scratch. The first step is to create a traffic sign with Sprites. To do this, students will use the Sprites as traffic signal lights. Then they will create a junction with two traffic signs, and then they will create a junction with four traffic lights. At this point, students are challenged to create a circuit with two traffic signs. Subsequently, students must use Sprites to create the lights for the pedestrian crossing portion. Finally, the final objective is to implement a traffic signal to control the crossing of vehicles and pedestrians.

In the third part, we work on the implementation of a garage access barrier. To do this, students must use a Sprite to implement the barrier. They must also include a color sensor in the programming so that the barrier knows when a vehicle is waiting, passing, or no vehicle is present. In addition, the garage barrier must include a traffic light signal, to inform you when it can pass, when caution is required because the barrier is opening or closing and, finally, to inform you when it cannot pass because the barrier is closed.

In the last part, students should use a Sprite containing color detectors so that a black line can be detected on a white background. They must also modify the background to create a circular-shaped circuit. Once these two elements and the Sprite program are developed, students must modify their programs to make sure that their virtual robot follows lines can move as quickly as possible without leaving the circuit. The next step is to use a rectangular shaped circuit. Finally, the Sprite will be used to circulate in more complex circuits that the students themselves must create.

B. Crumble basic level

The first part of this module shows students the Crumble hardware. The controller board is described and the hardware elements to be used along the module are described. These hardware items are such as the battery holder, the electrical connection cables, the USB cable for programming the controller card, the Sparkle, the ultrasonic sensor, the motors and the line detection sensor. The following describes the structure of the development interface and its different elements. The panels that make up the development interface, the categorization of commands and some basic commands are explained. At this point, we move on to activities in which the students have more participation. The first challenge is for them to program the controller so that the Sparkle looks a color of their choice. The following exercise is intended to make the Sparkle blink with a one-second power-on period and a one-second power-off period. To finish this part, students should program the controller so that the Sparkle displays a succession of colors of their choice.

In the second part, we work on the implementation of a traffic signal with Scratch. The first step is to create the sequence of aspects to be displayed by the Sparkle. Students will use the following sequence: two seconds red, two seconds yellow and two seconds green. They will then modify their program so that the sequence runs cyclically. In a next step, students should modify their schedule for the yellow aspect, instead of being a fixed aspect, or an intermittent aspect. At this point, students are challenged to create the lights on the pedestrian crossing side, i.e. fixed red, fixed green and flashing green. Finally, the final objective is to implement a traffic signal to control the crossing of vehicles and pedestrians.

In the third part, we work on the implementation of a garage access barrier. To do this, students must use a servo to implement the barrier. They must also include an ultrasonic sensor so that the barrier knows when a vehicle is waiting, passing, or no vehicle is present. In addition, the garage barrier must include a traffic light signal, to inform you when it can pass, when caution is required because the barrier is opening or closing and, finally, to inform you when it cannot pass because the barrier is closed.

In the last part, students must use a wheeled robot containing line detectors so that a black line on a white background can be detected. A white surface and black insulating tape will be used to draw the different circuits. The first basic circuit will be a circular one Once a program is developed that makes the robot follow the circular circuit, students must modify their programs to make sure that their robot follows lines can move as quickly as possible without leaving the circuit. The next step is to use a rectangular shaped circuit. Finally, it will use more complex circuitry that the students themselves must create.

C. Arduino basic level

In the first part of this module, students are shown the Arduino hardware. The controller board is described and the hardware elements to be used along the module are described. These hardware items are such as the electrical connection cables, the USB cable for programming the controller card, the LEDs (Light Emitter Diode), the ultrasonic sensor, the motors and the line detection sensor. The following describes the structure of the development interface and its different elements. It explains the structure of Arduino programs, the general syntax of the language, how the ports are configured, and how the ports on the card are written and read. Also included are explanations on how to choose the COM programming port and which card to choose before programming. Finally, it details how a program would load on the Arduino card. At this point, we move on to activities in which the students have more participation. The first challenge is to program the controller so that the LED integrated in the board looks fixed. The following exercise is intended to make the LED blink with a one second power-on period and a one second power-off period. To finish this part, students must integrate an external LED into the board and use the program created previously.

In the second part, we work on the implementation of a traffic signal with Arduino. The first step is to connect an LED for every aspect of the traffic signal. Second, students must program the sequence of aspects to be displayed by the LEDs. Students will use the following sequence: two seconds red, two seconds yellow and two seconds green. In a next step, students should modify their schedule for the yellow aspect, instead of being a fixed aspect, or an intermittent aspect. At this point, students are challenged to create the lights on the pedestrian crossing side, i.e. fixed red, fixed green and flashing green. Finally, the final objective is to implement a traffic signal to control the crossing of vehicles and pedestrians.

In the third part, we work on the implementation of a garage access barrier. To do this, students must use a servo to implement the barrier. They must also include an ultrasonic sensor so that the barrier knows when a vehicle is waiting, passing, or no vehicle is present. In addition, the garage barrier must include a traffic light signal, to inform you when it can pass, when caution is required because the barrier is opening or closing and, finally, to inform you when it cannot pass because the barrier is closed.

In the last part, students must use a wheeled robot containing line detectors so that a black line on a white background can be detected. A white surface and black insulating tape will be used to draw the different circuits. The first basic circuit will be a circular one Once a program is developed that makes the robot follow the circular circuit, students must modify their programs to make sure that their robot follows lines can move as quickly as possible without leaving the circuit. The next step is to use a rectangular shaped circuit. Finally, it will use more complex circuitry that the students themselves must create.

IV. RESUTLS FROM EXPERIENCE

This section contains all the details of the experience carried out with the deployment of the basic Scratch module to a group of children without any previous programming or robotic experience.

A. Course location and resources used

The course took place in the center of La Estera, in the town of Camarma de Esteruelas. Camarma de Esteruelas is located in the eastern part of the Community of Madrid. La Estera is an independent socio-cultural center, self-funded by neighbors and a non-profit organization.

The classroom is made up of tables that form an island. On one side of the island the visual learning material is projected. On the other hand, the instructor manages the computer that contains the audiovisual material, presents the session and interacts with the children.

Students have personal computers (PCs), each of which is used in pairs. These PCs run Lubuntu as an operating system. Lubuntu is a lightweight version of the Ubuntu distribution that is characterized by optimizing the resources of computers that do not have a lot of RAM (Random-Access Memory) or the speed of its microprocessor is not very high. This operating system was chosen because of its compatibility with Scratch, Crumble and Arduino programming environments. In addition, Lubuntu can be used free of charge and free software can be installed for educational purposes.

B. Student profiles

The student group consisted of eight students aged 6 to 11. Four students were under the age of 8. The rest of the students were between 9 and 11 years old.

One of the students is a girl (10 years old) and the other thirteen are boys. Table I summarizes the grades students have. In addition, Table I includes the correspondence between the Spanish educational level and the ISCED (International Standard Classification of Education) level.

TABLE I. STUDENTS' GRADES

Spanish educational grade	**ISCED level**	**Number of students**
Primary Education	1	14

Before starting the module, a pre-test was conducted to obtain the concerns about their attendance at the module. They were also asked about their previous knowledge about programming, robotics and the use of the Scratch tool.

Table II compiles the responses to the other questions related to previous knowledge about programming, robotics and the use of the Scratch tool. Most of the students had not programming, robotics and Scratch experience.

TABLE II. PRE-TEST RESPONSES

Question	**Student's level (1 – None; 5 – High)**				
	1	**2**	**3**	**4**	**5**
Programming experience	14	0	0	0	0
Programming knowledge	14	0	0	0	0
Robotics knowledge	14	0	0	0	0
Scratch experience	12	1	1	0	0
Scratch knowledge	12	1	1	0	0

C. Course results

In the first part, most of the students had their first contact with Scratch. This part is eminently theoretical but combined with the participation promoted by the instructor. As the instructor explained the theoretical concepts, the participants could experiment to a greater or lesser extent. All participants completed the first activity. The rest of the activities required additional clarification and assistance from the instructor. Table III summarizes information regarding the activities that were completed by the students.

TABLE III. FIRST PART ACTIVITIES AND STUDENTS WHICH COMPLETED

Activity title	Completed without help	Completed with some help	Completed with help
Scratch envorionment	14 (100 %)	0 (0 %)	0 (0 %)
Programming without computer	0 (0 %)	2 (14 %)	12 (86 %)
Basic programming	14 (100 %)	0 (0 %)	0 (0 %)
Programming a simple story	4 (29 %)	10 (71 %)	0 (0 %)

Table IV summarizes information regarding the activities that were completed by the students along the second part. During the first activities, the students needed help from the instructor to achieve the objectives they had set for themselves. However, the students completed the last two activities without having to be assisted by the instructor.

TABLE IV. SECOND PART ACTIVITIES AND STUDENTS WHICH COMPLETED

Activity title	Completed without help	Completed with some help	Completed with help
Working on Sprites	0 (0 %)	2 (14 %)	12 (86 %)
Basic traffic signal lights (I)	0 (0 %)	4 (29 %)	10 (71 %)
Basic traffic signal lights (II)	14 (100 %)	0 (0 %)	0 (0 %)
Complete traffic signal lights	14 (100 %)	0 (0 %)	0 (0 %)

In the third part, again needed support from the instructor during the first activities. As with the second part of the module, in this third part the students completed the last activities independently. Table V summarizes information regarding the activities that were completed by the students.

TABLE V. THIRD PART ACTIVITIES AND STUDENTS WHICH COMPLETED

Activity title	Completed without help	Completed with some help	Completed with help
Basic garage barrier	2 (14 %)	12 (86 %)	0 (0 %)
Garage barrier with detection	4 (29 %)	10 (71 %)	0 (0 %)
Garage barrier with lights	14 (100 %)	0 (0 %)	0 (0 %)
Complete garage barrier	14 (100 %)	0 (0 %)	0 (0 %)

Table VI summarizes information regarding the activities that were completed by the students during the fourth part. As can be seen from the results obtained, all the students showed greater independence in solving the challenges they had been faced with.

TABLE VI. FOURTH PART ACTIVITIES AND STUDENTS WHICH COMPLETED

Activity title	Completed without help	Completed with some help	Completed with help
Color detector	10 (71 %)	2 (14 %)	2 (14 %)
Circuit	12 (86 %)	2 (14 %)	0 (0 %)
Increasing speed	12 (86 %)	2 (14 %)	0 (0 %)
Complex circuits	14 (100 %)	0 (0 %)	0 (0 %)

Additionally, at the end of module, a post-test was conducted to obtain the opinion about the student's outcomes from the module. Firstly, a battery of question was asked with aim of getting the students' opinion about the module. Table VII compiles the students' opinion about the module.

TABLE VII. STUDENTS' OPINION ABOUT THE MODULE

Module topic	Number of students			
	Most liked	Less liked	Easiest	Hardest
Programming	3	2	10	3
Creating histories	0	12	3	6
Making real life aplications	11	0	1	5
Theory	0	0	0	0
Nothing	0	0	0	0

All questions related to verify that the students had acquired the knowledge that was intended to be transmitted throughout the course were answered correctly by all students.

On the other hand, Table VIII compiles the responses to the other questions related to acquired knowledge about programming, robotics and the use of the Scratch tool. Most of students increased their perception about programming, robotics and Scratch knowledge. Furthermore, most of students showed an increase in their curiosity about robotics.

TABLE VIII. POST-TEST RESPONSES

Question	Student's level (1 – Nothing; 5 – Many)				
	1	2	3	4	5
Programming knowledge	0	0	2	7	5
Robotics knowledge	0	0	3	3	8
Robotics interest	0	1	3	2	8
Scratch knowledge	0	0	2	7	5

At the end of the module we could see how all the students increased their motivation to program and create simple projects. At the beginning of the module, all the students had knowledge of new technologies such as computers, tablets and smartphones. Despite this, the use they made of it was a basic user use this knowledge was limited to Internet queries, games or basic functionalities. During the course, they scaled up their skills such as systems thinking, programming mentality, active learning, math, science, judgment and decision making, good communication, technological design, complex problem solving and persistence.

D. Discusion

As it can be seen from the results shown in the previous sections, a pedagogical strategy that combines theory and practice contributes to a level of attention that helps students acquire knowledge and become involved in the learning process.

As it can be seen from the results shown, Scratch is an ideal tool for introducing students to robotics. Scratch is distributed free of charge, can be used in its online or offline versions for Windows, MAC OS and Linux operating systems, and is a tool that does not require large resources from the device where it runs.

Throughout the activities, most of the students completed the proposed activities without the help of the instructor. Some of the students completed the activities with the support of the instructor. In some cases, other students were providing support to others. The need for help was not related to the age of the students or to gender.

The students enjoyed their activities and learned a lot from the experience. In addition, they improve their STEM-related skills.

V. CONCLUSIONS

As shown throughout this paper, the combination of STEM pedagogy with educational robotics presents promising opportunities for the development of skills and competencies needed by future professionals. Despite this, the introduction to educational robotics is not an easy task. But thanks to the great variety of tools such as the proposals, it is possible to contribute with cost-effective solutions that provide great flexibility to deploy varied knowledge in a way that is attractive to students.

First, it is demonstrated how a course can be planned that takes advantage of common contents to be developed with different types of platforms. Each of the above tools has many features that fit different age ranges of students. In this way, Scratch is a pure software tool that makes it easy for students to access areas of knowledge related to computer science. It is also very useful for the development of skills such as creative thinking, systematic thinking and collaborative work. Crumble, on the other hand, is a tool that enhances everything you've acquired with Scratch and makes it easier for students to access basic electrical and electronic systems. Last but not least, Arduino is a great choice when it comes to implementing virtually any type of application.

This work also includes an example of educational material for the basic levels proposed for each of the tools. For the cases in which a student must repeat a module, an educational material could be developed based on similar lines to those proposed in the example (lights, sensors, moving elements) but with other different applications. This would continue to attract attention and motivate students to help them acquire the skills and abilities they have not been able to develop adequately.

The experience discussed at the end of the paper demonstrates the potential of the course and the Scratch tool as the start of the proposed course. We still need to continue working with the proposed materials, develop more educational materials and analyze the results of more students in order to complete the proposal presented.

Finally, the results will be integrated in an open hardware platform which promotes innovation and motivation for students during the learning process [23]. The platform which is being developed presents wirelessly connections such as Bluetooth and WiFi as enhancements [24]. This research continues the development described in [25]. The doctoral thesis is being carried out in the Engineering Industrial School of UNED (Spanish University for Distance Education) and the Electrical and Computer Engineering Department (DIEEC).

ACKNOWLEDGMENT

The authors acknowledge support from the Engineering Industrial School of UNED, the Doctorate School of UNED, and the “Techno-Museum: Dis-covering the ICTs for Humanity” (IEEE Foundation Grant #2011-118LMF) and partial support from the eMadrid project (Investigación y Desarrollo de Tecnologías Educativas en la Comunidad de Madrid) - S2013/ICE-2715, the Internet of Things for European Small and Medium Enterprises (IoT4SMEs) project, the Erasmus+ Strategic Partnership (no. 2016-1-IT01-KA202-005561), mEquity (Improving Higher Education Quality in Jordan using Mobile Technologies for Better Integration of Dis-advantaged Groups to Socio-economic Diversity), Erasmus+ Capacity Building in Higher Education 2015 (grant no. 561727-EPP-1-2015-1-BG-EPPKA2-CBHE-JP), the Platform Integration of Laboratories based on the Architecture of visiR (PILAR) project, and the Erasmus+ Strategic Partnership (grant no. 2016-1-ES01-KA203-025327). And to the Education Innovation Project (PIE) of UNED, “Desarrollos Avanzados Multi-Objetivo de Laboratorios Remotos para Actividades Educativas – DAMO-LRAE”, from the Academic and Quality Vicerectorate and the IUED (Instituto Universitario de Educación a Distancia) of the UNED as well as to the project 2018-IEQ18 from the Industrial Engineering Technical School of UNED.

The authors are also thankful to El Circulo, Incomaz, Plaza Robótica, Alejandro Muñoz Montero and César Muñoz Montero due to their collaboration.

REFERENCES

[1] G. Carro, S. Martin, E. Sancristobal, F. Mur and M. Castro, "Robotics, the New Industrial Revolution," in IEEE Technology and Society Magazine, vol. 31, no. 2, pp. 51-58, Summer 2012. DOI: 10.1109/MTS.2012.2196595

[2] Plaza P, Sancristobal E, Carro G, Castro M, Blázquez M “From 4 Wheeled Remote Robot to Serious Collaborative Remote Laboratory”. Int Rob Auto J 3(4): 00065. DOI: 10.15406/iratj.2017.03.00065

[3] Mazumder, O., Kundu, A., Lenka, P., et al.: ‘Robotic AFO to enhance walking capacity: initial development’, Electronics Letters, vol. 52, no. 22, pp. 1840-1841, 10 27 2016. DOI: 10.1049/el.2016.3056

[4] Zhou, S., Zhang, W., Zou, Y., et al.: ‘Piezoelectric driven insect-inspired robot with flapping wings capable of skating on the water’, Electronics

Letters, vol. 53, no. 9, pp. 579-580, 4 27 2017. DOI: 10.1049/el.2017.0186

[5] Pickering, T. A., Yuen, T. T., and Wang, T.: ‘STEM conversations in social media: Implications on STEM education’, 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Bangkok, 2016, pp. 296-302. DOI: 10.1109/TALE.2016.7851810

[6] Kintsakis, D., and Rangoussi, M.: ‘An early introduction to STEM education: Teaching computer programming principles to 5th graders through an e-learning platform: A game-based approach’, IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 17-23. DOI: 10.1109/EDUCON.2017.7942816

[7] Mavroudi, A., Economides, A. A., Fragkou, O., et al.: ‘Motivating students with Mobiles, Ubiquitous applications and the Internet of Things for STEM (MUMI4STEM)’, IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 37-38. DOI: 10.1109/EDUCON.2017.7942820

[8] Delgado Kloos, C., Rodríguez, P., Velázquez-Iturbide, A., et al.: ‘Digital education in the classroom’, IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 31-32. DOI: 10.1109/EDUCON.2017.7942818

[9] P. Plaza, E. Sancristobal, G. Carro, M. Castro, M. Blazquez, J. Muñoz and M. Alvarez. “Scratch as Educational Tool to Introduce Robotics”. In: Auer M., Guralnick D., Simonics I. (eds) Teaching and Learning in a Digital World. ICL 2017. Advances in Intelligent Systems and Computing, vol 715. Springer, Cham. DOI: 10.1007/978-3-319-73210-7_1

[10] K. Brennan, "Learning Computing through Creating and Connecting," in Computer, vol. 46, no. 9, pp. 52-59, September 2013. DOI: 10.1109/MC.2013.229

[11] N. F. Velasquez et al., "Novice Programmers Talking about Projects: What Automated Text Analysis Reveals about Online Scratch Users' Comments," 2014 47th Hawaii International Conference on System Sciences, Waikoloa, HI, 2014, pp. 1635-1644. DOI: 10.1109/HICSS.2014.209

[12] S. A. Nikou and A. A. Economides, "Transition in student motivation during a scratch and an app inventor course," 2014 IEEE Global Engineering Education Conference (EDUCON), Istanbul, 2014, pp. 1042-1045. DOI: 10.1109/EDUCON.2014.6826234

[13] P. Plaza, E. Sancristobal, G. Carro and M. Castro, "Home-made robotic education, a new way to explore," 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 132-136. DOI: 10.1109/EDUCON.2017.7942837

[14] J. C. Martínez-Santos, O. Acevedo-Patino and S. H. Contreras-Ortiz, "Influence of Arduino on the Development of Advanced Microcontrollers Courses," in IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 12, no. 4, pp. 208-217, Nov. 2017. DOI: 10.1109/RITA.2017.2776444

[15] Y. Lin, J. Conners and S. Livengood, "Assessing the use of open source microcontroller board for teaching engine sensing and communication in automotive laboratory," 2017 IEEE Frontiers in Education Conference (FIE), Indianapolis, IN, 2017, pp. 1-5. DOI: 10.1109/FIE.2017.8190738

[16] A. Garrigós, D. Marroquí, J. M. Blanes, R. Gutiérrez, I. Blanquer and M. Cantó, "Designing Arduino electronic shields: Experiences from secondary and university courses," 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 934-937. DOI: 10.1109/EDUCON.2017.7942960

[17] L. Boaroli, A. D. Spacek, C. L. Izidoro, J. Mota Neto, E. Maestrelli and O. H. Ando Junior, "Data Monitoring and Hardware Control for App Android by Bluetooth Communication for Laboratory Teaching in Electrical Engineering Courses," in IEEE Latin America Transactions, vol. 15, no. 1, pp. 31-39, Jan. 2017. DOI: 10.1109/TLA.2017.7827885

[18] D. Călinoiu, R. Ionel, M. Lascu and A. Cioablă, "Arduino and LabVIEW in educational remote monitoring applications," 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Madrid, 2014, pp. 1-5. DOI: 10.1109/FIE.2014.7044027

[19] J. J. Gil, I. Diaz, X. Justo and P. Ciäurriz, "Educational haptic controller based on Arduino platform," 2014 XI Tecnologias Aplicadas a la Ensenanza de la Electronica (Technologies Applied to Electronics Teaching) (TAEE), Bilbao, 2014, pp. 1-7. doi: 10.1109/TAEE.2014.6900140

[20] Scratch [Online]. Disponible en: https://scratch.mit.edu/, visited on 6 April 2018.

[21] Redfern electronics [Online]. Disponible en: http://redfernelectronics.co.uk/, visited on 6 April 2018.

[22] Arduino [Online]. Disponible en: https://www.arduino.cc/, visited on 6 April 2018.

[23] P. Plaza, E. Sancristobal, G. Carro, M. Castro and C. Pérez, "Collaborative robotic educational tool based on programmable logic and Arduino," 2016 Technologies Applied to Electronics Teaching (TAEE), Seville, 2016, pp. 1-8. DOI: 10.1109/TAEE.2016.7528380

[24] P. Plaza, E. Sancristobal, G. Carro and M. Castro, "A Wireless robotic educational platform approach," 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV), Madrid, Spain, 2016, pp. 145-152. DOI: 10.1109/REV.2016.7444455

[25] P. Plaza, E. Sancristobal, G. Carro and M. Castro, "Robotic Educational Tool to engage students on Engineering," 2016 IEEE Frontiers in Education Conference (FIE), Eire, PA, 2016, pp. 1-4. DOI: 10.1109/FIE.2016.7757417