

A survey of resources for introducing coding into schools

Francisco J. García-Peñalvo
GRIAL Research Group,
Research Institute for Educational Sciences
University of Salamanca, Spain
fgarcia@usal.es

Angela Marie Rees,
Jenny Hughes
Pontydysgu, Welsh educational research institute, UK
angela.marie.rees@gmail.com, jenhughes@mac.com

Ilkka Jormanainen,
Tapani Toivonen
School of Computing, University of Eastern Finland, Finland
{ilkka.jormanainen, tapani.toivonen}@uef.fi

Jens Vermeersch
GO! Onderwijs van de Vlaamse Gemeenschap, Brussels, Belgium
jens.vermeersch@g-o.be

ABSTRACT

Within TACCLE 3 – Coding European Union Erasmus+ KA2 Programme project, a review and evaluation of a set of resources that can contribute to teaching programming to younger children has made. This paper presents a survey of this review including the most outstanding products in order to help teachers to introduce programming in pre-university studies.

CCS Concepts

- Social and professional topics → Computational thinking
- Social and professional topics → K-12 education

Keywords

Computational Thinking; Coding; Resources; TACCLE 3

1. INTRODUCTION

Many European countries are introducing computing, with a special emphasis in programming issues, as a core curriculum subject [1]. Some have already done so; many others are intending to.

Many teachers at Primary and Secondary Education levels are using many activities that are useful in developing competence in coding although they might not recognize this. Others perhaps are fully lost in the way to tackle this challenge. TACCLE 3 – Coding European Project (Ref. 2015-1-BE02-KA201-012307) [2-4; 9] focusses on supporting school teachers and developing their confidence to deliver the new computing curriculum including coding [1] and computational thinking [6; 10-12] approaches.

Specifically, TACCLE 3 project has three main objectives:

1. To equip fellow classroom teachers, whatever their level of confidence, with the knowledge and the materials they need to teach coding effectively.
2. To develop a website of easy-to-follow and innovative ideas and resources to aid teachers in teaching coding. They will also find a review of the current academic research and an overview of the resources currently available for teaching coding.

- To provide national and international in-service training courses and other staff development events to help support and develop confidence and competences in teaching coding.

This paper is related partially to aim two. It presents a survey of the reviewed resources for introducing coding into schools. Section 2 explains the review methodology. Section 3 presents an overview of the analyzed resources. Section 4 makes a deeper survey of two resource categories. Finally, Section 5 closes the paper with the conclusions.

2. REVIEW METHODOLOGY

TACCLE 3 will equip classroom teachers with the knowledge and the materials they need by developing a website of ideas and resources together with in-service training courses and other staff development events.

In the TACCLE 3 coding project, a lack of didactic material for teachers to get started teaching coding to young pupils from primary school level on, was identified. In order to compensate such deficit, a survey of resources and starter kits to support the teachers' approach to teach coding at primary school level was undertaken.

A collection of ideas, and pupil oriented tools and environments such as iconic programming software, literature, and examples of good practice in video towards coding, computational thinking and STEM was reviewed, analysed and evaluated by partners of 6 countries (Belgium, Estonia, Finland, Germany, Spain and United Kingdom).

During the April-September 2016 period resources have been analysed, evaluated and documented following the TACCLE 3 template [5]. In this template the resources were classified following the TACCLE 3 criteria in these categories: Algorithms, Using logic, Controlling things, and Creating and Debugging.

After that, in order to create a resource catalogue for introducing to programming, a resource map has been generated using other complementary classification: App for teaching coding, Robotic, Maker stuff, Programming language, Book, Info site, and Training course.

It is important to underline that the same resource might be classified in multiple TACCLE 3 and Complementary categories.

With the record-set of the reviewed resources, more detailed information is elaborated and published in the project website. Section 4 presents exemplary cases of this.

3. REVIEW RESULTS

TACCLE 3 has a resource database with 37 records, most of them have a free or creative common license (See Figure 1). Following figures summarize the results of this review. Table 1 presents the title and the license of the reviewed resources.

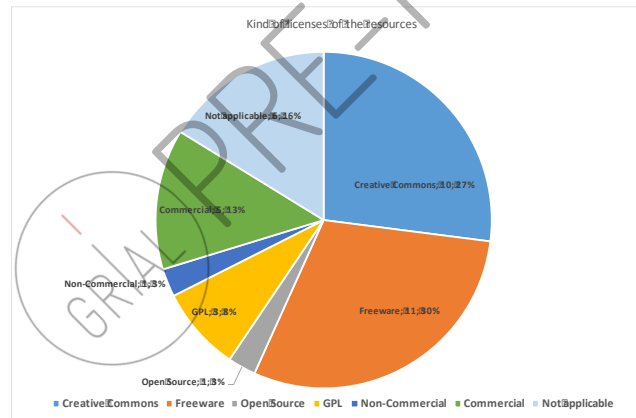


Figure 1. Licenses of the reviewed resources

Table 1. Reviewed resources list

Title	License
BAXTERROBOT	Commercial
GeomagicTouchHaptic	Commercial
Minecraft	Commercial
SCRATCH	CreativeCommons
PedagogicalConversationalAgentDrRoland	CreativeCommons
SphereKids	Freeware
AMICIProgrammingEnvironmentwithAnIconicInterfaceforArduinoLilyPadandSmartTextile	GPL
EduWearStarterKitWearableIntelligenceforClothesSportsandGames	GPL
ZauberschuleInformatikEinErsterEinblickinDieWeltDerInformatik	CreativeCommons
KoduSameLab	PersonalandNon-Commercial
3piRoboticsPlatform-course	CreativeCommons
MSWLogo	Freeware
Studio.code.orgCourse2	CreativeCommons
Ozobot	Commercial
Minetest	LGPL
SoyMinero	-
TACCLECodingWebSite	CreativeCommons
123DDesign	Freeware
TynkerCodingForCode	-
TynkerHourOfCode	-
BlocklyforDashDotRobots	Freeware
PathforDashRobot	Freeware
Code.org	-
CodeStudio	-
Code.orgHourOfCode	-
MITAppInventor	CreativeCommons
MaKeyMaKey	Commercial
Arduino	OpenSource
CSUnpluggedComputerScienceWithoutaComputer	CreativeCommons
MakingAktivitätenmitKindernundJugendlichenHandbuchzumKreativenDigitalenGestalten	CreativeCommons
MagicalClothing	Freeware
Koodiaapinen	CreativeCommons
Koodikirja	Freeware
HelloRuby	Freeware
Koodikoulu	Freeware
TheFoss	Freeware
ProjeTiger	Freeware

Figure 2 presents the reviewed resources classified by the TACCLE 3 criteria.

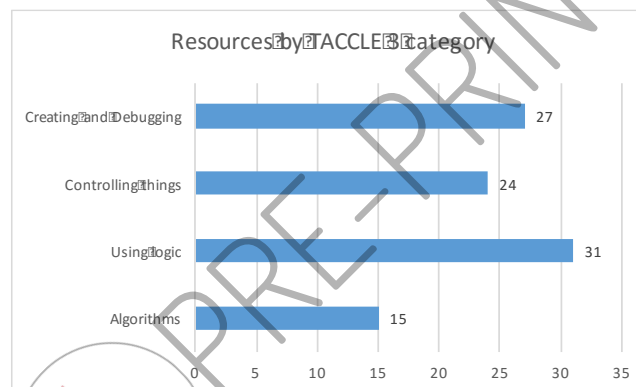


Figure 2. Reviewed resources classified by category

Figure 3 shows the reviewed resources classified by the complementary category, which present a significant presence of information websites and apps for teaching coding.

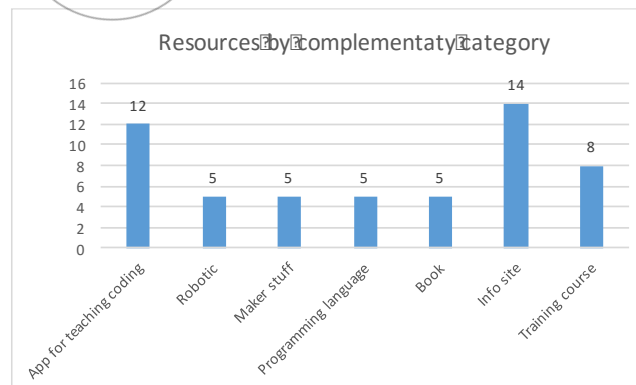


Figure 3. Reviewed resources classified by the complementary classification

Figure 4 shows the languages in which the resources are available. English is the most usual language.

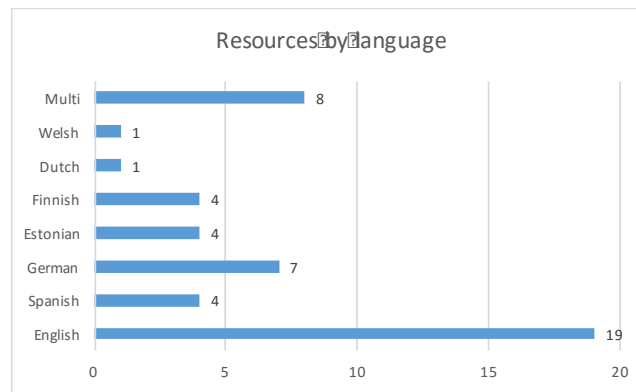


Figure 4. Languages of the reviewed resources

4. RESOURCE SURVEYS

Once the resources database is made and TACCLE 3 researches have the resources catalogue, it is possible to make more detailed surveys oriented to pre-university in-service teachers or future teachers.

Following sections reflects two different developed surveys, one about the best apps for teaching programming, and other about the best robotics kits.

4.1 Apps for teaching programming survey

In [7] comments about these apps may be also found.

4.1.1 ScratchJr

In this app, kids, ages 5 to 7 can learn to program their own games and interactive stories. Kids snap together programming blocks to make their characters move, jump, dance, and sing. ScratchJr was inspired by the popular Scratch programming language [8], developed by MIT. It is a great way to introduce young kids to programming. Figure 5 shows the app logo.



Figure 5. ScratchJr logo

4.1.2 Tynker

Kids with no knowledge of programming can learn logic with this iPad app. The app contains story puzzles that kids can solve through dragging and dropping to make visual code blocks. Kids learn to recognize patterns and break down a problem into small steps. Tynker includes one free game and additional games are available with in-app purchasing. Recommended for ages 9 to 11. Figure 6 shows the app logo.



Figure 6. Tynker logo

4.1.3 ScriptKit

ScriptKit is best for older kids, ages 12 and up. ScriptKit is a drag and drop programming app for the iPad. With it, kids can build simple mobile prototypes using native iOS UI components and social media APIs. This means they can design and write code on the iPad. Figure 7 shows the app logo.



Figure 7. ScriptKit logo

4.1.4 Hopscotch: Coding for Kids

Kids, ages 8 to 12, can use simple, intuitive building blocks to create games, animations, and apps. They can drag and drop blocks of code to create anything they can dream up. Figure 8 shows the app logo.



Figure 8. Hopscotch logo

4.1.5 Lightbot

In this game, your kids, ages 9 to 11, can guide a robot to light up tiles and solve the different levels using commands. Lightbot can help introduce kids to programming concepts and coding. There exists an easier version of Lightbot designed for younger kids, ages 6 to 8, it is so called Lightbot Jr. Figure 9 shows the app logo.



Figure 9. Lightbot logo

4.1.6 Kodable

With this free app, your kids can help the fuzzFamily explore the planet Smeeborg's Technomazes. While playing, they'll get a kid-friendly introduction to programming concepts. Gameplay is easy, you just drag and drop instructions for your fuzzes to follow and help them get through their mazes. Perfect for kids ages 5 and up. Figure 10 shows the app logo.



Figure 10. Kodable logo

4.1.7 Robozzle

Kids can use simple commands to help a robot solve puzzles, follow a linked list, or even count in binary. With a growing database of over 1,000 puzzles, the kids will have plenty of opportunities for problem solving and fun. Figure 11 shows the app logo.



Figure 11. Robozzle logo

4.1.8 Cargo-Bot

Cargo-Bot is a puzzle game where children teach a robot to move crates. The game features 36 puzzles. It was created entirely on the iPad using a program called Codea. Figure 12 shows the app logo.



Figure 12. Cargo-Bot logo

4.1.9 SpaceChem Mobile

In this game, kids will take on the role of a Reactor Engineer working for SpaceChem. They can create factories to transform raw goods into valuable products, and streamline designs to meet production quotas and beat other engineers. Figure 13 shows the app logo.



Figure 13. SpaceChem logo

4.1.10 Code Combat

Code Combat is an on-line game where children use JavaScript to advance through levels, defeat ogres and wizards. With single-player and multi-player levels, it is perfect for any aged kid. Plus, it is free to play and a fun way to learn coding. Figure 14 shows the app logo.



Figure 14. Code Combat logo

4.1.11 Puzzlets

With Puzzlets (formerly called Ludos), kids can use plastic pieces on a base connected to their tablet to direct the character in the game and solve puzzles. The pieces have simple commands directing the character in the game to go up, right, down, and even jump. Figure 15 shows the app logo.



Figure 15. Puzzlets logo

4.1.12 Bee-Bot

Bee-Bot is another app that is perfect for the younger kids. Help kids, ages 4 and up, learn directional language and programming while directing their bee robot through various scenarios. Figure 16 shows the app logo.



Figure 16. Bee-Bot logo

4.1.13 KidsRuby

Kids can learn the programming language Ruby through this downloadable computer program. While not a game like the other apps, KidsRuby lets you write code and see the output instantaneously. The help files will take you step-by-step through the process. Figure 17 shows the app logo.



Figure 17. KidsRuby

4.2 Robotics kits survey

This section explores the most outstanding robot kits for introducing kids to the robotic world.

4.2.1 Lego Mindstorms EV3

Lego Mindstorms series is an educational robotics platform by Lego. There have been quite several versions of Mindstorms, EV3 series being the most recent one. While Mindstorms has been around for a while, there is a large community of users playing around with it so if you stuck to any challenges, getting help won't be an issue. EV3 robots are programmed with a specific visual programming environment with a drag-and-drop interface. Although Legos are not prone to damage, the programmable EV3 intelligent brick has a display that breaks easily if it is dropped. For users with an experience on programming languages, EV3 can be programmed with some popular programming languages such as Java and Python.

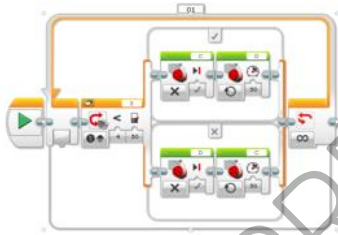


Figure 18. Lego Mindstorms EV3

While the EV3 set (which costs about 250€) provides sensors, motors, EV3 intelligent brick and some Legos, you can design and build your robot with regular Legos too, so EV3 series is very flexible and in the end, only your imagination is the limit. The visual programming environment is easy to use and learn. This does not however mean that the programming environment would not be powerful: kids can program their robot to follow the line or avoid obstacles but also it is possible to program a soccer playing team of robots to compete in robot soccer (See Figure 18).

EV3 robots can be used in various settings; it can be used at elementary schools but it has been also used at the university level. The visual programming environment supports basic programming structures such as loops and conditional statements. EV3 robots can also be controlled remotely with a smart phone with a specific app that can be downloaded from the app stores (Google play and App store).

The Mindstorms series is mature robotics kit so the hardware or the software don't suffer from bugs or constant changes. The set can be bought from a larger stores and it can be ordered online. While EV3 is flexible, powerful and widely used, it is relatively expensive and the programmable intelligent brick breaks easily. The battery life of EV3 is relatively long lasting and can last even days.

Pros:

- Flexible (can be extended with regular Legos and many sensors).
- Easy to program.
- Very powerful.
- Popular.
- Wide scope of use scenarios from elementary schools to university level.

Cons:

- Expensive.
- EV3 intelligent bricks breaks easily.

More information available at <http://www.lego.com/en-us/mindstorms/about-ev3>.

4.2.2 Bee-Bot

Bee-Bot (see Figure 19) is an affordable robotics kit that includes only the Bee-Bot robot. Bee-Bot is a colorful and easy-to-operate robot designed especially to younger children. Bee-Bot is programmed with buttons that are placed in the top of the Bee-Bot robot. Bee-Bot remembers up to 40 command and executes the commands sequences by moving around. Bee-Bot does not break easily and because it is very easy to use and does not require computer to program it suits well for young students.



Figure 19. Bee-bot

Even if a robotics kit uses visual programming environment it can take a long time for students to learn to program the robot. This is the case especially with younger children. Bee-Bot is designed to be easy to use. The commands that Bee-Bot can take are: forward, backward, left and right. Bee-Bot executes the first given command for some time (something like 1 second) and then moves to execute the second command. Bee-Bot is a good way to teach children logical thinking and problem solving.

Bee-Bot is powered by battery that is charged from USB-port. It is not clear how long the battery survives, but it is likely that it is powerful enough to cater the need in most typical use scenarios. The Bee-Bots are relatively cheap: a unit costs about 25 euros. Because Bee-Bot is designed for younger children, older learners might find it boring after some time because of low level of challenges it provides is. Bee-Bots cannot be extended with additional building blocks or such.

Pros:

- Very easy to use.
- Cheap.
- No computer needed for programming.
- Durability.

Cons:

- Not very challenging.
- Cannot be extended and is not very flexible.

More information available at <https://www.bee-bot.us/>.

4.2.3 Robbo

Robbo offers two different robotics kits. The first kit, so called “Lab kit” is a board with different sensors and buttons attached to it. The Lab kit is connected to a computer with USB cable or with Bluetooth and it can be used together with Scratch programming environment. Because Lab kit is built with Arduino microcontrollers, it can also be programmed with Arduino programming language. This is a good option for more advanced learners. The Lab kit sends commands to the computer and it can be used for example to control characters in Scratch. The price of the Lab kit is around 100 euros.

While Lab kit itself cannot be extended with different Arduino equipments, the programming environment is very flexible because all of the Scratch-based commands are available. It is a good way to extend the Scratch to the physical world. The price of the Lab kit is reasonable (taking into account that it features an Arduino board). The kit is reasonably durable, although it is not recommended to drop it many times as the plastic housing may get damaged. Because Lab kit can be programmed with Scratch, it suits well for younger students but also because it can be used together with Arduino programming language, older students can use it and find challenges on it. Lab kit has a microphone, light sensor, speaker, leds and for example buttons that can be used together with Scratch.

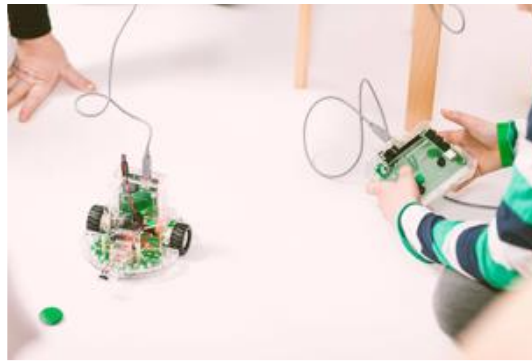


Figure 20. Robbo moving around

Robbo also provides an Arduino-based robot, “Robot kit”. Robot kit is an assembled Arduino robot that has flash light, light sensor, touch sensor, proximity sensor, two line sensors and motors to move the robot around (see Figure 20). The sensors can be mounted to a robot with a magnet so they are easy to add and remove. Robot kit, like the Lab kit, can be programmed with both Scratch and Arduino so the children and youth from various age groups can use with and find it challenging.

Besides plug and play -type of sensors, Robot kit cannot be extended with regular Arduino parts so it is not super flexible. The price is relative expensive with the price tag of 250 euros.

Robot kit seems to be durable although it should be handled as carefully as any high-tech device. Both Lab kit and Robot kit are good alternatives for Mindstorms. Although Mindstorms' focus is in both programming and building the robot, Robbo puts its focus in just programming the robot. Although Robbo a newcomer in the market, the community behind Arduino and Scratch is wide, and getting help will not be a problem. The company producing Robbo kits is driven with open source principles, and they provide complete set of blueprints and schematics for both products to be used in do-it-yourself projects.

Pros of Lab kit:

- Relatively cheap (150€).
- Durable.
- Can be programmed with both Scratch and Arduino.

Cons of Lab kit:

- Limited flexibility.

Pros of Robot kit:

- Can be programmed with both Scratch and Arduino.
- Different sensors.
- Durable.

Cons of Robot kit:

- Expensive (250€).
- Limited flexibility.

More information available at <http://robbo.world/>.

4.2.4 Cubelets

Cubelets are block-like robots that can be paired with other Cubelets. There are three different types of Cubelets: Sense blocks that sense the world with different type of sensors, Think blocks that can be used for logical thinking such as calculating mathematical equations and Act blocks that have motors. The way that Cubelets act is based on their order how they are connected to each other and students learn this by trial and error. Although Cubelets can also be programmed with C programming language or controlled remotely with your tablet or smart phone. Cubelets are meant to engage students to think logically.

Cubelets are very flexible and can be extended for example with Legos (see Figure 21). The number of Cubelets connected to each other is unlimited. The downside of Cubelets is that they are very expensive: 12 Cubelets starter kit costs around 320 euros. However, a set of 12 Cubelets is enough for even bigger group of students.

Cubelets offer challenges to different age groups since they can be programmed by connecting them to each other but also they can be plugged to also a computer and programmed with C programming language. There is also an app for smart devices that allows you to remotely control your Cubelets so finding new applications for Cubelets is not an issue.



Figure 21. Cubelets

Pros:

- Very flexible.
- Various ways to program.
- Durability.
- For different age groups.
- Enhance logical thinking.

Cons:

- Very expensive (320 euros for 12 Cubelets).

More information available at <http://www.modrobotics.com/cubelets/>.

4.2.5 Ozobots

Ozobot is a small and cheap robot with a novel approach to learn programming. Ozobot uses colors to control its movements and actions so the students can release their imagination and program Ozobot with drawings (see Figure 22). Ozobot has a color sensor attached at the bottom, and it has pre-programmed behavior so it will act differently when encountering different colors. Ozobot can also be programmed with smart phones or tablets to widen the use cases of Ozobot.



Figure 22. Ozobots

The programming environment of Ozobot is visual and similar to Scratch. It is easy to use and learn. However, it is clearly designed for the younger children and older students might not find Ozobot so challenging or fun. For a cheap and novel way to engage the younger students in programming with multidisciplinary approach, Ozobot might be a good choice.

Besides the robot itself, Ozobot offers printable mazes, apps for programming and playing and other accessories to modify the appearance of Ozobot so there will be a lot of playing with.

Pros:

- Cheap (55€).
- Novel way to program.
- Easy to learn, designed for children.
- A lot of accessories.

Cons:

- Robot itself cannot be extended.
- Perhaps not the right choice for older students.
- Battery life unknown.

More information available at <http://ozobot.com/>.

4.2.6 Edison

Edison robots are cheaper alternatives for Mindstorms series since their price is just about 50 euros. Edison robots include motors, microphone, distance sensor, light sensor and remote control opportunity (see Figure 23). Edison robots can be extended with regular Legos. Sensors and motors of Edison robots are fixed so extending the robot with different sensors or motors is not possible. Edison robots are also able to read bar codes, which is a remarkable feature especially for younger children that Mindstorms series does not provide.

The programming of Edison robots is made with a visual programming environment similar to Mindstorms' environment (see Figure 24). The programming environment works with both computers and smart devices, such as iPads or Android phones. There is also a Python based robot programming environment available. Although Edison robots cannot be extended with external sensors or motors, the basic set offers all important sensors and two motors and students can extend Edison robots with regular Legos.

Although much cheaper, Edison robots do not give such flexibility that EV3 robots give due the fact that no different motors or sensors can be used with Edison. Yet, Edison is a good alternative to EV3 robots with fewer opportunities and cheaper costs.



Figure 23. Edison

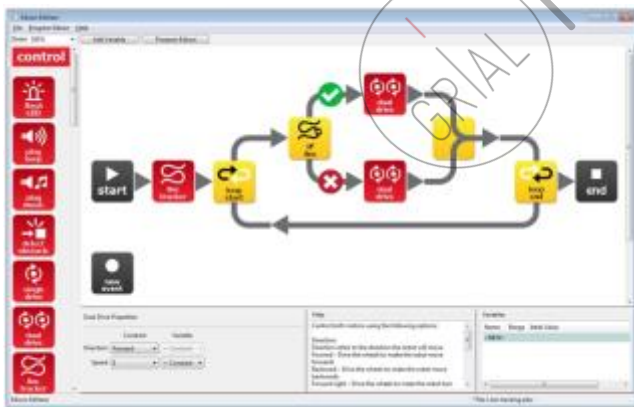


Figure 24. EdWare Robot Programming Software

Pros:

- Cheap (50€).
- Can be extended with Legos.
- Visual and Python programming environments.
- Bar code reading.

Cons:

- Sensors and motors are fixed.

More information available at <https://meetedison.com/>.

4.2.7 mBot

mBot is an Arduino-based robotics kit that enables students to assemble the robot with motors and sensors without requiring students to know anything about electronics (see Figure 25). Sensors and motors to mBot are easy to plug with cables and the programming of mBot is made with Scratch environment. Being Arduino-based platform, it is possible to use Arduino programming language too. The aim of mBot is to provide a platform that can be mainly used to learn programming, so the assembling the mBot has been made easy.

mBot can be remotely controlled and it is relative cheap (about 90 euros). The Arduino board is modified so that sensors and motors can be plugged with cables in the same way as in Lego's EV3 robot. Lego bricks and similar can be used to extend mBot. Even if mBot is easy to program, extending mBot with Arduinos requires knowledge in electronics. mBot might not be the most durable option in the hands of younger students.



Figure 25. mBot

Pros:

- Cheap (90€).
- Flexible and can be extended.
- Visual programming and Arduino programming.
- Large user community.

Cons:

- Sensors and motors are fixed.

More information available at <http://www.makeblock.com/mbot-stem-educational-robot-kit-for-kids/>.

5. CONCLUSIONS

Introducing coding or programming in the pre-university studies is a big challenge for all. Timing and decision making to act formally at the curricula level is not an easy way. Too many teachers are introducing computing far away the digital literacy competences but usually they make it isolated in their subjects.

TACCLE 3 project is trying to create a significant teacher community, which shares the objective of introducing programming and/or computational thinking in their classes, and also looking for breaking this isolation effect and making an attraction effort for new teachers that want to but do not dare to give a step beyond.

The first step to create the community is having a website with attractive resources. In this paper we have presented the first approach to build up a resource catalog and some surveys to help them to find suitable teaching paths and make decision to introduce activities that help students to discover or go further into the programming and computational thinking.

6. ACKNOWLEDGMENTS

With the support of the EU Erasmus+ Programme. KA2 project "TACCLE 3 – Coding" (2015-1-BE02-KA201-012307).

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

7. REFERENCES

- [1] Balanskat, A. and Engelhardt, K., 2015. *Computing our future. Computer programming and coding Priorities, school curricula and initiatives across Europe*. European Schoolnet.
- [2] García-Peñalvo, F.J., 2016. A brief introduction to TACCLE 3 – Coding European Project. In *2016 International Symposium on Computers in Education (SIIE)*, F.J. García-Peñalvo and J.A. Mendes Eds. IEEE, USA.
- [3] García-Peñalvo, F.J., 2016. Presentation of the TACCLE3 Coding European Project. <http://repositorio.grial.eu/handle/grial/654>.
- [4] García-Peñalvo, F.J., 2016. Proyecto TACCLE3 – Coding. In *XVIII Simposio Internacional de Informática Educativa, SIIE 2016*, F.J. García-Peñalvo and J.A. Mendes Eds. Ediciones Universidad de Salamanca, Salamanca, España, 187-189.
- [5] García-Peñalvo, F.J., 2016. Template for TACCLE 3 resources reviewing. DOI=<http://dx.doi.org/https://dx.doi.org/10.6084/m9.figshare.3545033.v1>.
- [6] García-Peñalvo, F.J., 2016. What Computational Thinking Is. *Journal of Information Technology Research* 9, 3, v-viii.

- [7] Hughes, J., 2016. Best apps for teaching programming. TACCLE 3, <http://www.taccle3.eu/english/2016/05/10/best-apps-for-teaching-programming/>.
- [8] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y., 2009. Scratch: Programming for all. *Communication of the ACM* 52, 11, 60-67. DOI=<http://dx.doi.org/10.1145/1592761.1592779>.
- [9] Taccle 3 Consortium, 2016. TACCLE 3: Coding Erasmus + Project website. <http://www.taccle3.eu/>.
- [10] Wing, J.M., 2006. Computational Thinking. *Communications of the ACM* 49, 3, 33-35. DOI=<http://dx.doi.org/10.1145/1118178.1118215>.
- [11] Wing, J.M., 2008. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society a-Mathematical Physical and Engineering Sciences* 366, 1881 (Oct), 3717-3725. DOI= <http://dx.doi.org/10.1098/rsta.2008.0118>.
- [12] Wing, J.M., 2011. Computational Thinking. In *2011 Ieee Symposium on Visual Languages and Human-Centric Computing*, G. Costagliola, A. Ko, A. Cypher, J. Nichols, C. Scaffidi, C. Kelleher and B. Myers Eds., 3-3.

