

TEACHING ARDUINO PROGRAMMING AT HIGH SCHOOL LEVEL

Valeriu Manuel IONESCU¹, Cosmin STIRBU²

University of Pitesti, Romania

¹valeriu.ionescu@upit.ro, ²cosmin.stirbu@upit.ro

Keywords: education, Arduino, teaching, high-school

Abstract: Developing the programming skills for the high school pupils in the Romanian educational system is mostly a theoretical issue, with very little applications to real hardware systems. This can be in some cases an issue as having practical applications to theoretical algorithms is a major driving force for pupils that motivates them to research more and correct or improve the software-hardware interaction. This paper presents considerations regarding the teaching of Arduino programming for high school level pupils.

1. INTRODUCTION

The advent of low cost programming platforms (Arduino, Raspberry Pi) has made the accessible for a large audience and (for simple applications) the programming skills necessary are low.

This is why at university level, in Romania there are many initiatives where hardware manufacturers or companies have presented to the students access to Arduino or Arduino compatible hardware that is especially used in the first undergraduate years due to the low programming skills necessary and high practical applicability. [1]

Based on this, the University of Pitesti has initiated in 2017 a contest for high schools [2] in the Arges County where more than 160 pupils were enrolled. This allows the pupils to increase their performance, interact with the university environment and motivate them by creating a competitive context.

This paper presents considerations related to the teaching Arduino programming environment for high school pupils in the Romanian Educational System.

This paper is structured as follows: chapter 2 presents short study of the Romanian National Education curricula for schools and high schools and the moment when Arduino programming can be integrated in scholar and extra scholar activities; chapter 3 investigates the Arduino

platform (with its versions) and the programming environment while outlining the differences and targeted audiences; the paper concludes with considerations regarding the applicability of integrating Arduino applications in the high school programming classes.

2. HIGH SCHOOL CURRICULA PERQUISITES

In Romania, the presence of Programming and IT&C in high schools determined by the published orders of the Ministry of National Education [3].

These documents point out the main information that should be transmitted to the pupil, but also allows a certain degree of freedom in choosing the content. For example, the school curricula cover roughly 75% from the teaching-learning time, giving the teachers 25% time to allocate for implementing and adapting to the changing environment according to the pupils needs. [4]

Arduino programming and platform development needs not only basic programming skills but also understanding electrical circuits and digital logic. In the gymnasium, the taught IT&C skills only provide basic digital competences for using digital devices and searching relevant educational information on the internet. [5]

High school studies in Romania take 4 years to complete and two distinct levels can be identified:

- The inferior cycle (classes IX and X) where Informatics and IT&C is taught 1-2 hour/week with the possible addition of an extra 1 hour/week that can be allocated if the school decides accordingly.
- The superior cycle (classes XI and XII) when the compulsory education in the Romanian educational system ends, with the number of IT&C classes presented in Table I (according to the high school profile) [6]

Table 1. IT&C classes time allocation for superior cycle

Profile	Discipline	Hours per week
Mathematics/Informatics	Informatics	4
Nature science	IT&C	2
Social Science	IT&C	2
Philology	IT&C	1
Vocational - military	IT&C	1
Vocational – military (Mathematics/Informatics specialization)	Informatics	4
Vocational - artistic	Computerized image processing	1
Vocational educational	IT&C	1
Vocational - Sports	-	0
Technologic	IT&C assisted technologies	2

For all the cases presented in Table 1 an additional 3-4 hours/week can be allocated for the IT&C purpose (among other possible subjects) but this is a decision that is taken individually for each high school. The decision

takes into consideration the pupils request, the number of teachers and the availability of the informatics laboratory. [7]

With few exceptions (sports profile – with no IT&C classes and artistic profile with classes oriented towards image processing) all the other profiles study aspects related to IT&C with Mathematics/Informatics profile having a large number of hours allocated to Informatics. This shows that the percentage of high school of pupils that can take advantage of the Arduino platform features is large.

The manuals used in high schools that are agreed by the Education Ministry [8] (some being presented in Table 2) indicate that the languages agreed to be used for Informatics and IT&C target the Pascal and C++ languages (C# only in intensive informatics profiles) and Database fundamentals (Access). The C language is considered to be a system level programming language, with no support for classes and objects, while C++ is considered a general purpose programming language with a wider range of applications.

For Arduino programming the current curricula (C++) is a good match and satisfies the knowledge requirements. The Pascal language on the other hand, good for describing algorithms and easy to understand as it is close to natural language, is not very popular and it is difficult to find libraries for specific real world problems.

It is interesting to consider a recent study that showed which are the most popular programming languages (in order): JavaScript (build websites and hybrid programming), Python, Java (Android and general purpose cross platform programming), PHP (Wordpress, Facebook, Wikipedia), C++, Ruby (A general purpose, object oriented programming language), C# and Swift/Objective-C (Apple applications).

Table 2. Manuals for high school superior cycle - Informatics and IT&C classes

Title	Class	Authors	Editure
Informatică	XI	Vlad Huțanu, Tudor Sorin	L&S Soft
Informatică C++	XI	Sanda Junea, Adriana Simulescu	Corint Gimnasiu
Informatics C++		Daniela Oprescu, Liana Bejan Ienulescu	Niculescu ABC
Informatică Pascal	XI	Daniela Oprescu, Liana Bejan Ienulescu	Niculescu ABC
Informatică - tehnologii asistate de calculator, varianta Access	XII	Mariana Panțiru, Ionuț Panțiru, Irina Panțiru	L&S Info-Mat
Tehnologia informației și a comunicațiilor, TIC3 - Sisteme de gestiune a bazelor de date	XI	Mioara Gheorghe, Monica Tătărăm, Corina Achinca, Constanța Năstase	EDP

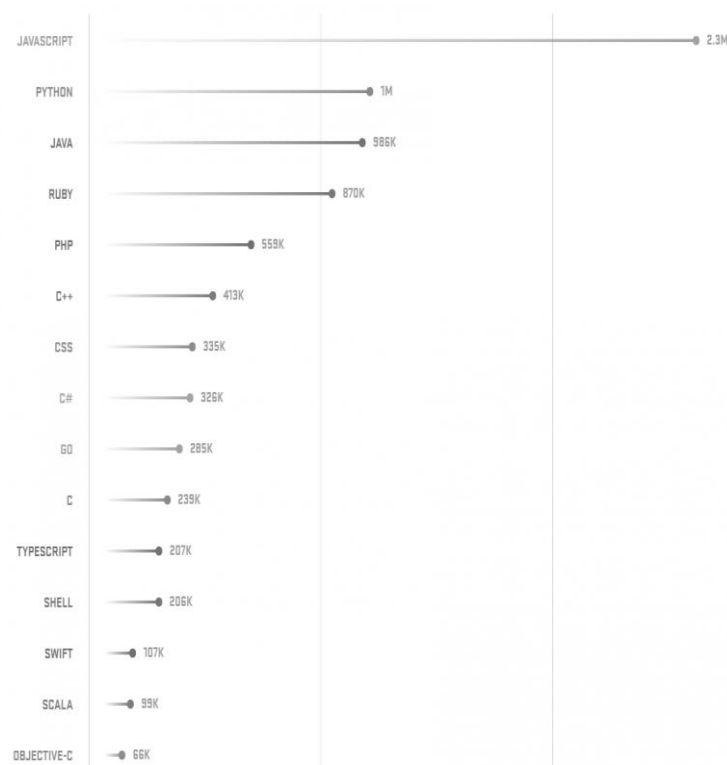


Fig. 1 The github statistics [9] regarding the most sought programming languages in 2017

This shows that the languages C++ and Pascal are no longer the main languages sought by programmers.

This means that there could be other languages that the pupils could be taught in order to be better prepared to real life applications. There are some aspects that make this idea hard to implement.

First the purpose of high school programming is to teach algorithms and the C++ and Pascal languages are well designed for that. The practical applications for the presented algorithms are not the main goal in the educational process, even in vocational high schools. Applicative purpose for algorithms is reserved for university level teaching.

The other thing to consider is that changing the languages taught in high schools is a difficult task to implement because it requires the training of professors and the creation of new manuals.

In the long run universities should prepare for this change and this can also be a major issue.

The computer performance level in the existing laboratories does not need to be

improved as even 10 year old systems are able to compile and run code at an acceptable level.

The main software used in programming classrooms: CodeBlocks, FreePascal, Visual Studio (for programming) and Oracle or Visual Fox (for databases).

The interfaces presented by these environments allow easily switching to and using the Arduino IDE.

3. THE ARDUINO PLATFORM AND INTEGRATED DEVELOPMENT ENVIRONMENT

The platform was developed in 2003 [10] as the need to create personalized projects has increased.

Arduino targets mainly educational projects that aim to automate tasks in various domains, such as vehicles, smart homes, games, etc.

The programming level required to create applications for this platform is low, making it a good entry level board, with many already made

examples that can be used to increase the development speed.

The lack of complex applications is due to the reduced memory size, as ATmega328 Boards have only 32kB Program Space that can store for example a web server with LCD control occupies over 70% of the memory [11] The Flash and SRAM memory that a sketch file will use are shown at the compilation time. When the available memory is very limited, unexpected behavior can occur.

Techniques such as careful handling of strings (due to large memory requirements) and placing char types in header files can reduce the memory footprint and should be considered for more complex projects.

The first Arduino board was created in 2005 and allowed connecting software applications to hardware projects using the UART, PWM, digital I/O and the analog inputs.

Having a single platform to develop applications (Arduino one of the first Open Source Hardware projects) has allowed a large number of people to contribute with ideas and extend the support at global level.

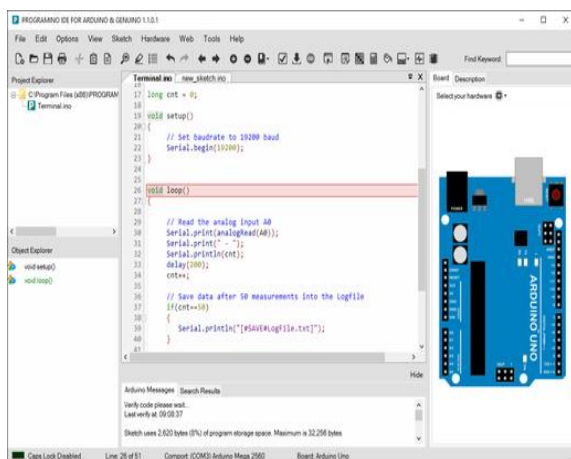
The microcontroller used is ATmega328 operating at 5V (16MHz frequency), with a 7-12V input voltage necessary for the board.

The Arduino language [12] is a set of C/C++ functions that have automatic generation of the prototype. However, the library for the AVR products has a reduced set of library functions, making it closer to C than C++. [13]

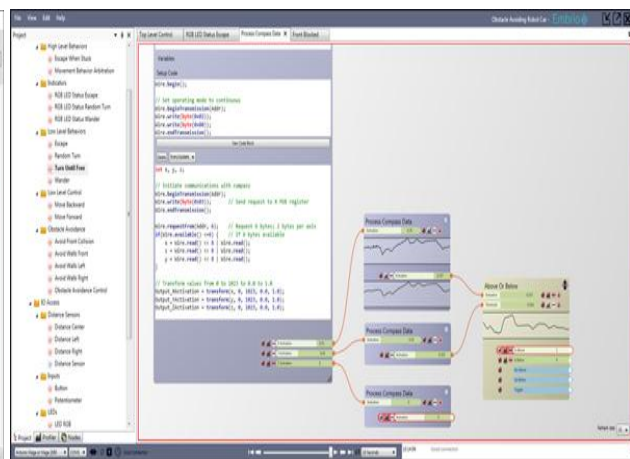
More complex aspects of programming for Arduino involve using preprocessor directives (such as `#include`, `#define`, `#ifndef`, `#endif`, etc.) for designing libraries, better managing the debugging code or handling the inclusion of multiple libraries. [14]

Table 3. Different Arduino boards [15]. CTC 101 and MATERIA 101 are kits including many additional resources for quick educational hardware development

Level	Board name
Entry level	UNO, LEONARDO, 101, ESPLORA
Enhanced features	MEGA, ZERO, DUE, MEGA ADK M0, M0 PRO
Internet of things	YÚN, ETHERNET, TIAN, INDUSTRIAL 101, LEONARDO ETH, MKR FOX 1200, MKR WAN 1300, MKR GSM 1400
Educational	CTC 101
Wearable	GEMMA, LILYPAD (multiple versions)
3D Printing	MATERIA 101



(a)



(b)

Fig. 2 PROGRAMINO (a) and Embrio (b) IDEs for Arduino

Starting from the initial board, different versions (even clones or extended kits) were

created in order to better target a specific public. Different versions are presented in Table 3.

Entry level products are used for beginners to understand both software and hardware in comparison with products Enhanced features that are used for complex projects.

The IoT range is used to connect different systems and services over the Internet (e.g. home electronics).

Arduino Education is used for educational purposes to increase the experience of teachers in innovative learning.

Arduino Wearable is used for wearable projects (e.g. personal healthcare assistant) that target low power, reduced size and weight characteristics. Arduino 3D printing is used to create the control side of 3D printers, technology that constitutes the basis of many custom made projects.

This gives a wide diversity of choice both in features and prices.

The diversity is manifested also in the software department.

There are multiple IDE's that can be used to program for the Arduino platform.

The most used are those that are based on Eclipse and the Arduino software IDE.

Other alternative IDE's include for example Embrio, Zerynth and PROGRAMINO IDE. Two of these IDEs can be seen in Fig. 3 along with the fact that a great accent is placed on the visual experience.

The Arduino Integrated Development Environment (IDE) is a great tool to start programming, having multiple tools for debugging and monitoring the platform (library management, board upload, etc), however for larger projects there are other IDE's that can be used to better manage a larger amount of files, such as Eclipse IDE.

Finally, even if there is no physical Arduino board, there are a number of Arduino emulators that can be used [16], with one available for the CodeBlocks environment [17]

Recently, the Arduino Company proposes the online editing platform called Arduino Create.

This tool uses a web interface in order to write and test the code.

Programs written using Arduino Software (IDE) are called sketches.

They do not have a complex structure, with only two compulsory functions: *setup* – needed to run a piece of code once (usually in the beginning) and a section that is run continuously called *loop*.

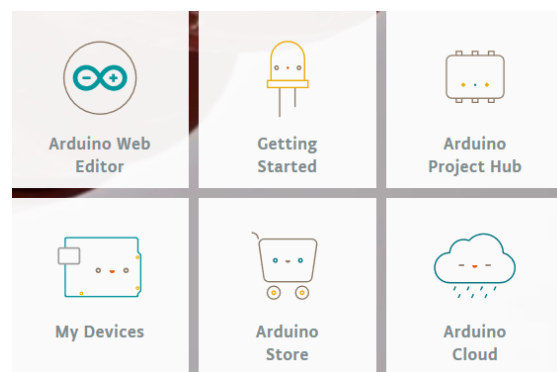


Fig. 3. Arduino Create online platform

One of the first sketches that can be taught is Led Blink sketch (included in the Arduino examples list) with the following program structure (hardware presented in Fig. 4a).

```
void setup()
{
    pinMode(13, OUTPUT);
}

void loop()
{
    digitalWrite(13, HIGH);
    delay(1000); //wait 1000 ms
    digitalWrite(13, LOW);
    delay(1000); //wait 1000 ms
}
```

Understanding the hardware for the Blink program needs basic Physics notions concerning Ohm's law and Kirchoff theorems that is taught in gymnasium classes [19] (if external LED is used instead of the platform's integrated one – highlighted for Arduino 1 R3, pin 13, in Fig. 4b).

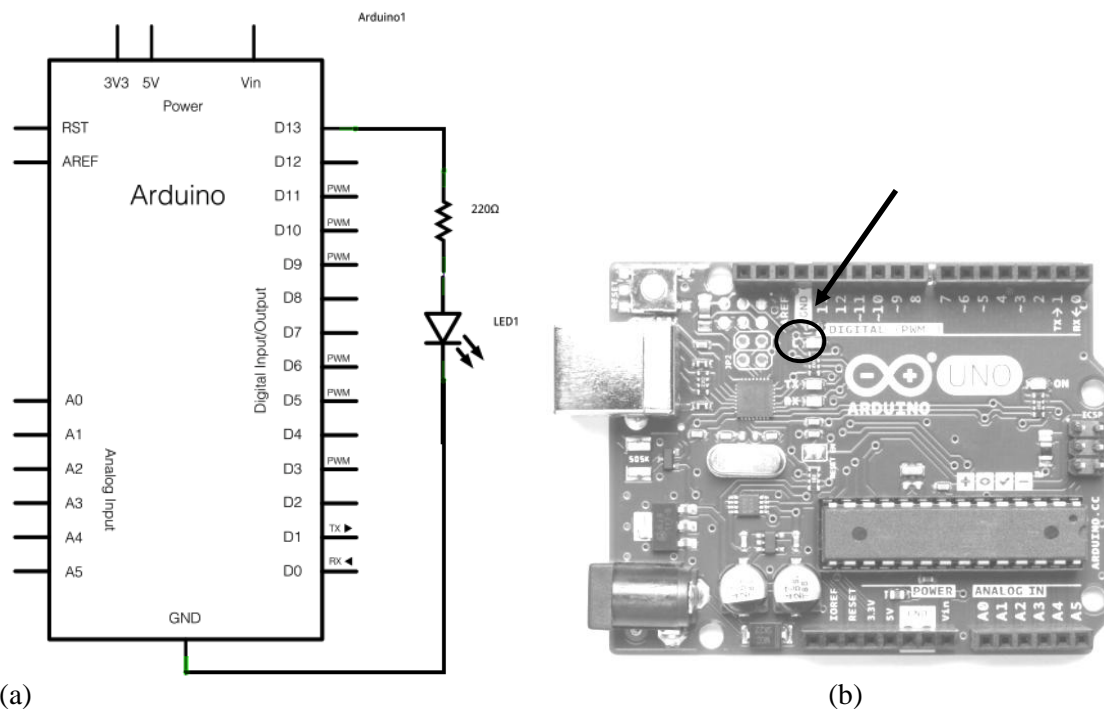


Fig. 4. (a) First Sketch Led Blink hardware connection [18] and (b) Arduino Uno R3 with an LED integrated in the board connected to pin 13.

In conclusion the accessibility of the software and the programming tools is to a level that can be easily followed by high school students.

The extensive online help and the community with forums and wiki groups make the process of developing an initial application simpler. If English language can be considered a problem to understand (as most tutorials on the internet are in the English language), there are also numerous guides in Romanian language provided usually by the hardware resellers that are interested in targeting a wider market share with their products [20].

Finally it is possible to identify different degrees of difficulty for Arduino applications and this can be used to determine which applications will be presented to the high school pupils:

A. Simple applications

- Turn on or off a LED
- Turn on a LED until a button is pressed
- Cyclically turn on/off a led at each button pressing
- Blink a led with 1Hz frequency
- Blink a LED with 10Hz only when the button is pressed

B. Medium difficulty

- Using the GPIO interrupts turn on the led at button pressing
- Using timer interrupts blink a led with 10Hz frequency
- Using GPIO and timer interrupts blink the led with 10Hz when the button is pressed and 20Hz when the button is released

C. High difficulty

- Creating libraries that can be easily ported to different Arduino boards
- Developing Arduino Cores that allow using different chips for Arduino

For high school students, depending on the number of hours dedicated to IT&C classes, it is easy to create the simple or even some of the medium difficulty applications presented above.

The projects that target the high difficulty applications require both a greater understanding of the programming environment and testing on multiple Arduino devices that can prove costly to acquire.

4. CONCLUSIONS

This paper overviews the context of presenting Arduino programming to the high school students.

As most high school profiles include programming classes in both inferior and superior cycles and the Physics notions necessary to create basic circuits are presented at the gymnasium level, it is easy to include Arduino related classes at any level in high schools, especially in the superior cycle (classes XI and XII).

The University of Pitesti has initiated in 2017 a contest for high schools in the Arges County where pupils need to program an Arduino board controlling a simple follow the line robot.

The contest that had an even than larger then intended audience, proposing as a target about 20 teams of 3-4 pupils and finalizing with more than 160 enrolled pupils from more than 10 high schools.

The teachers directing the pupil's groups were also invited to participate at the course in order to improve their knowledge on the subject and interact with the university staff.

The interaction between high schools and universities is beneficial for both high schools and university professors as it is possible to assess the interest and the knowledge that the high school students have.

Programming for Arduino is simple, and needs C programming skills that are taught in high schools. The computers performance level necessary to run the software is adequate even for those that are a decade old.

The hardware necessary for practical applications is also cheap to acquire or build if necessary. It is also possible to run the code in an emulator if the costs are considered prohibitive.

This shows that the only limits in creating applied programming to the Arduino (and similar) platforms at high school level are the professors and students interest to improve and be better prepared for the labor market.

5. REFERENCES

- [1]. Revista COMUNIC@ȚII Mobile, "Universitatea Politehnică din București, inclusă pe lista primelor 5 universități europene care vor utiliza plăcile Intel Galileo", web, <http://www.comunic.ro/article/universitatea-politehnic%C4%83-din-bucure%C8%99ti-inclus%C4%83-pe-lista-primelor-5-universit%C4%83%C8%9Bi-europene-0>, 18.10.2013, Accessed: 10.12.2017
- [2]. University of Pitesti, "Concursul de Robotică RoboSmart al Universității din Pitești", web, <https://www.upit.ro/ro/academia-reorganizata/facultatea-de-electronica-comunicatii-si-calculatoare-2/concurs-robosmart>, Accessed: 10.12.2017
- [3]. Ministerul Educatiei Nationale, ORDIN privind aprobarea Regulamentului-cadru de organizare și funcționare a unităților de învățământ preuniversitar, web, Accessed: 10.12.2017, https://www.edu.ro/sites/default/files/Regulament%20cadru%202016_0_2.pdf
- [4]. Ministerul Educatiei Nationale, "Ordinul ministrului Educației Naționale nr. 3.393/28.02.2017"
- [5]. Ministerul Educatiei Nationale, "Ordin privind aprobarea planurilor cadru de invatamant pentru invatamantul gimnazial", web, http://programe.ise.ro/Portals/1/Curriculum/Pl_cadru-actuale/Gimnaziu/OMENCS%203590_5%20apr%202016_Plan-cadru%20de%20C3%AEnvatamant%20pentru%20gimnaziu.pdf, Accessed: 10.12.2017
- [6]. Ministerul Educatiei Nationale, OMECI nr 3410 plan cadru 9_12._17_03_2009, web, www.edu.ro
- [7]. Cristian-Dumitru Masalagiu, Ioan Asiminoaei, Mirela-Anca Țibu, "Didactica predării informaticii", Ed. Polirom, ISBN 978-973-46-3388-3
- [8]. Liceul teoretic Vasile Alecsandri Iasi, "Catalog manual", web, Accessed: 10.12.2017, http://www.lvais.ro/images/Catalog_manuale__X_I_XII_aprobate_de_MEC.pdf
- [9]. Github, "The State of the Octoverse 2017", web, Accessed: 10.12.2017, <https://octoverse.github.com/>
- [10]. Arduino, "About Us", web, Accessed: 10.12.2017, <https://www.arduino.cc/en/Main/AboutUs>
- [11]. Starting Electronics, "A Comparison of Arduino Sketch Sizes in Memory", June 12, 2015, web, <https://blog.startingelectronics.com/a-comparison-of-arduino-sketch-sizes-in-memory/>
- [12]. Arduino, Frequently Asked Questions, 2017, <https://www.arduino.cc/en/Main/FAQ#toc13>
- [13]. Free Software, "avr-libc", web, Accessed: 10.12.2017, http://www.nongnu.org/avr-libc/user-manual/FAQ.html#faq_cplusplus
- [14]. Jan Gromes, Arduino Preprocessor Directives Tutorial, June 6, 2017, web, Accessed: 10.12.2017, <http://www.deviceplus.com/how-tos/arduino-guide/arduino-preprocessor-directives-tutorial/>

- [15]. Arduino, "Arduino Products", web, Accessed: 10.12.2017, <https://www.arduino.cc/en/Main/Products>
- [16]. Dan Mihai, "Arduino Simulators Lineup — Start Developing Without a Real Board", web, Accessed: 10.12.2017, <https://www.smashingrobotics.com/arduino-simulators-lineup-start-developing-without-real-board/>
- [17]. Stanley Huang, "CodeBlocks Arduino IDE", web, Accessed: 10.12.2017, <http://arduinodev.com/codeblocks/>
- [18]. Arduino, "Blink", web, Accessed: 10.12.2017, <https://www.arduino.cc/en/Tutorial/Blink>
- [19]. MINISTERUL EDUCAȚIEI NAȚIONALE, "Anexa nr. 2 la ordinul ministrului educației naționale nr. 3393 / 28.02.2017 «Programa școlară pentru disciplina FIZICĂ Clasele a VI-a / a VIII-a»"
- [20]. Robofun, "Arduino pentru Toti" web, Accessed: 10.12.2017, <https://www.robofun.ro/docs/Arduino%20pentru%20Toti%20pagini%201-30.pdf>