Design of open source platform for automatic control systems education based on cooperative learning

Jonathan Álvarez Ariza
Uniminuto, Dept of electronic technology, Bogotá, Colombia
jalvarez@uniminuto.edu

***Abstract*—This paper exposes the design of a platform for the learning of control systems based upon collaborative efforts of the students in this area. The context of this platform arises according to some general educative and technical elements. The first element is to help to the technical processes that the students follow in the design of digital controllers in the automatic control area, e.g., modeling, simulation, evaluation and debug. The second element is focused on the development of a novel open source platform for automatic control, whose aim is to help to improve the educational process of the students in this area. The educational concept of the platform is based on cooperative learning methodology, in which the students collaborate with the platform design according to their needs and expectations. This methodology allowed to know initially the points of view of the students with the aim to establish the perspectives, motivations, skills and weaknesses of them in the automatic control area as well as in the different technical subareas needed in the design of the platform such as programming or signal conditioning. In agreement with the cooperative learning methodology, the construction of the platform was divided in small tasks with objectives clearly defined associated to the learning of the different educational contents in the components of hardware and software needed in the design of it. Our students interacted with each other, sharing common issues and difficulties associated with the technical behavior of the platform. This process increased the group motivation so as to find common aims both technical and human as likewise it encouraged the creativity in search of technical solutions in the design of the platform. The previous steps guarantee the principal concept of the cooperative learning methodology defined as *positive interdependence*, which is associated to how the student learns the different theoretical and practical educational contents and how this helps to the other group-mates to learn these contents through of the *interaction* with them.**

***Index Terms*—Cooperative learning, educational methodology, reflection, metacognitive, platform, open source, control system design.**

I. INTRODUCTION

THE Cooperative learning is a class of educational methodology in which the students working in groups in order to achieve determined purposes or goals[1]. This methodology is based upon the work of small groups with common interests according to the motivations and needs in a specific area, in this case the automatic control systems area. Considering the principal strategies of this methodology, e.g, the division of work in small tasks, the selection of heterogeneous groups with common interests and the collaboration between students for achieving the purposes defined[1], it has been designed a platform with help of the students. The development of the platform was motivated due to the lack of the technological platforms related with the control system area which provide to the students tools that serve to improve the educational processes in this area.

The stages of the project were divided as follows. A first phase of the project consisted in the inquiry of the motivations and interests both in the automatic control area and in the design of the platform, searching the needed skills that a student can have in the overall areas of hardware and software, in consonance with the principal components of the platform. A second phase involved the division of the work according to the chosen component by each student and the definition of small tasks with established objectives, e.g., the learning programming of the digital signal controller, the design of input filters or the design of the communications module for the development board among other tasks. A third phase consisted in the debugging of the platform, testing the different elements of it, e.g., the development board, the graphical user interface and the communications bridge between application and the board. Finally, a fourth phase of project consisted in the feedback given by the students involved in the design of the platform, with the aim of improving its functionality in both technical and educational aspects in future works.

In agreement with the premises mentioned before, this paper exposes the following main topics. Section II exposes the cooperative learning methodology and its implications in the project. Section III exposes the methodology of design in the components of hardware and software in the platform. Section IV exposes the results both technical and educational of the platform. Finally, section V exposes the conclusions established in the present paper.

II. THE CONCEPT OF COOPERATIVE LEARNING

The cooperative learning is understood as an educational methodology in which the students work in collaborative way searching common goals with the aim of enhancing their educative process. Johnson *et al*. [1],[2] have defined the general structure of this methodology based on the concepts of *learning goals*, *goals structure* and *positive interdependence*[1],[3]. This last term is associated to how the students *interact* each other, allowing that the knowledge to be shared and appropriated in the group. The Fig.(1) illustrates the main components of the cooperative learning (CL):

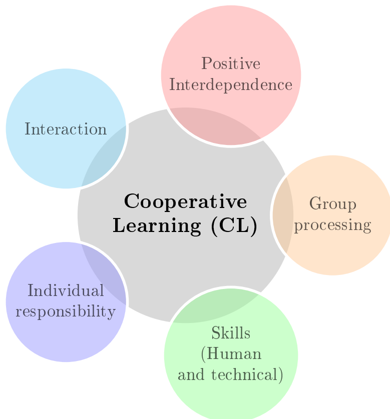

Figure 1. Main components involved in the cooperative learning methodology.

In relation to Fig.(1), the group processing is an element which defines the quality of the interaction in the students, promoting the insight learning in them. The reflection about the learning process in the student is an important issue that must be taking into account in the construction of the cooperative learning.

Some relevant researches on psychology [4] confirm that the *group interaction* and *social interdependence* are important concepts in the role that one individual has in a group as well as in the motivation that can have in relation to several activities or tasks. Furthermore, the motivation in an individual can change over time in function of several factors, e.g., the moods. The cooperative learning promotes the *interaction* in opposition to the competitive and individual learning methods. In the competitive way only the more skilled student will be called to get the knowledge regardless the method that use so as to get it. This has been a traditional method in education for a long time. In the individual method, the interaction is not important to get the knowledge and the relationship with the group is sporadic, restricted to personal interests. Both methods individual and competitive are classified into the concept of negative interdependence[2], because do not encourage the cooperativism.

An important feature in cooperative learning methodology is related with the division of the work. It must be divided in small tasks with defined objectives which must be achieved with the purpose of to meet an overall goal, e.g., the design of one specific project or the learning of a determined educational topic. In order to get the positive interdependence, it must promote the *motivation* in the group as likewise make aware to the students in the importance of the responsibility that each one has in the tasks assigned. Grow[5] in the *Stage Self Direct Learning* (SSDL) model indicates that in this stage the teacher changes his role of instructor to motivator. This stage is recognized by the students as *good teaching.* Nevertheless, in the educational process in the student, the stage of *good teaching* is an intermediate step between the led-learning and self-oriented learning. In this last phase, the students choose the information according to their needs as well as they resolve problems without the assistance of the teacher which becomes a guide in the learning process.

Conforming to the SSDL model, an approach to the educational concept in the platform is shown in the table as follows:

Table I
APPROACH TO THE EDUCATIONAL CONCEPT IN THE PLATFORM ACCORDING TO SSDL MODEL.

Stage	Type of student	Type of activity
1	**Dependent**	Assignment. Learning tools associated with the platform, under the direction of the teacher.
2	**Interested**	Learning by cases of use and discussion of the problems and their possible solutions in the design of the platform.
3	**Involved**	PBL (*Problem Based Learning*) in function of the needs in the platform design in the software and hardware areas.

Due to the nature of our program (*technological*), it has chosen the stages one to three of the model. The stage four requires an amount of time employed by the student in terms of the self-learning and self-management of the knowledge which is suitable for engineering or master degree programs. In the *stage one*, the student is dependent, that is to say, the students make their activities with the direction of the teacher, then the knowledge is transferred to each student in an unique way. This type of learning is not suggested to create a critical thinking in the student during his educational process. However, it is the initial approach to the interaction between learning and teaching methods in the classroom. In the *stage two*, the teacher changes his role of instructor to motivator as mentioned. It is important contextualize to the student in the project and its implications in his educational process, encouraging the motivation in the role that this represents in the project. A strategy in this stage is to know the point of view of the students associated with the perspectives, problems and possible solutions that they could have in a determined project. They will feel that their opinion is taking into account, enhancing the educational process and the relationship between the teacher and the student. Finally, in the *stage three,* the teacher accompanies the learning process providing experiences that can help to the student to resolve a problem. The teacher changes of motivator to a guide, facilitating the way in that the student learns. The students in this stage propose solutions and contribute with their experiences in the evolution of the project. A good methodology in this stage is PBL (*Problem Based Learning*)[6] which allows to generate in the student *insight*[7] as well as it encourages the creativity in the solution of a problem.

III. METHODOLOGY

In function of the framework mentioned, the beginning step in the methodology adopted was the division of the students by groups in the hardware and software areas. This division was made according to previous survey, where the students were inquired about of their motivations and skills in the mentioned

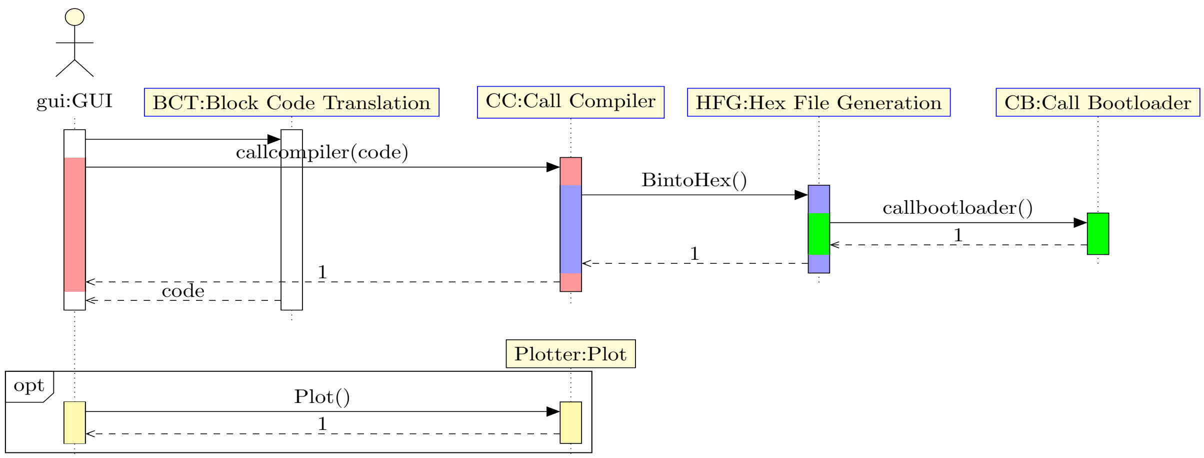

Figure 2. *UML* sequence diagram for the software user interface.

areas. The division of work was structured in levels as is shown in the following tables:

Table II
STRUCTURE OF TASKS ASSIGNED BY LEVELS TO THE HARDWARE GROUP.

Level	Task description
Level 1 ***(Investigation stage)***	1. Investigation about of the hardware devices needed in the core, power supply and conditioning signal in the development board. 2. Selection of the hardware devices in function of several parameters such as robustness, energy efficiency or velocity of processing *in MIPS* (Million of Instructions Per Second). 3. Learning tools needed for the DSC programming.
Level 2 ***(Design stage)***	1. *Starting design*: Signal conditioning for sensors and actuators, design of power supply for the development board.
Level 3 ***(Debugging stage)***	1. *Debugging of starting design*: Fixing mistakes presented in prototype design.
Level 4 ***(Design stage)***	1. Schematic construction of the development board taking into account the considerations established in the levels two and three.
Level 5 ***(Design stage)***	1. *PCB* (Printed Circuit Board) design of the development board. 2. Debug of the development board and its interaction with the *GUI* (Graphical User Interface).
Level 6 ***(Debugging stage)***	PCB debugging.
Outcome	*PCB version (1.0)*

Each level has a determined function in the evolution of the learning in the student, e.g., the initial levels allow to the students develop the investigation, providing them of cognitive elements that will be used in the solution of problems associated with the platform design. In the next sections, it will establish the technical methodology followed in the design of the hardware and software components in the platform.

A. Software structure

Taking into account the previous concepts in the cooperative learning methodology, the groups of students were divided in tasks associated with the software components as mentioned. The Fig.(2) illustrates the *UML* (*Unified Model Language*) sequence diagram for the *GUI*; it was the first approach to

Table III
STRUCTURE OF TASKS ASSIGNED BY LEVELS TO THE SOFTWARE GROUP.

Level	Task description
Level 1 ***(Investigation stage)***	1. Investigation about the programming languages (structure, type of language (*dynamic or static*), types of variables, etc. 2. Selection of the suitable programming language in the design of the GUI.
Level 2 ***(Design stage)***	1. Design of the UML (*Unified Model Language*) sequence diagram for the *GUI*. 2. Division of the GUI by classes and methods needed in the interaction between development board and the user.
Level 3 ***(Design stage)***	Starting design of the *GUI*.
Level 4 ***(Design stage)***	Design of additional blocks for controllers and peripheral such as PWM (*Pulse Width Modulation*), ADC (*Analog to Digital Converter*), among others for the GUI.
Level 5 ***(Design stage)***	Debugging of the interaction between the compiler and the bootloader in the *GUI*.
Level 6 ***(Debugging stage)***	Debugging of the *GUI* interface.
Outcome	*Graphical user interface, version (1.0)*

software development. The purpose of this diagram consisted in to establish the principal classes and objects important in the design. One student (*actor*) can interact with the software interface such is shown in the Fig.(2), the student creates the controller through of graphical blocks, setting the parameters of it. The code generated by each block is directly converted in the equivalent programming code for the *DSC* (*Digital Signal Controller*, *dsPIC 33FJ128MC802*)[8]. This operation is generated in real time.

We have chosen this method due to the efficiency and simplicity that it represents for the student that use the platform. According to the table (III), this was the second level in relation to the tasks assigned to the students belonging to the software group. The UML diagram allowed to the students to understand the different parts that compose the interface which are divided in *classes and methods*, establishing a relation between the operations that one actor can execute and the flow code of the user interface. One important fact is that our curriculum does not contain a specific subject for the software design. Thus, we created the educational material about these topics and we shared it with the group in the different theoretical and practical classes in the academic semester.

The priority interest in the software design was the use of open source tools. Therefore, we have chosen Java and JavaFX [9] library for the user interface. JavaFX allows to integrate web contents into the application, making it suitable for the handling of the interface *Blockly*[10].

Blockly is a web engine interface designed by Google which allows to generate the translation of a graphical block in a specific programming language, e.g., C, Python, JavaScript, Dart, among others [11]. For this case, the students have developed a C language converter according to the conventions of the compiler *Microchip XC16* [12], whose function is compiling the programs for the *DSC*. This process was executed in the levels three and four in concordance with the table (III). The code generated by each block does not have functionality because it must be compiled through of the compiler (*XC16*). It checks the syntax and the structure of the code, generating an executable file so as to programming the DSC. A further consideration in our work was the design and debugging the libraries needed for the implementation of digital controllers in the platform. These were added to the core of the compiler helping to generate a structure of code efficient and organized. Each one of these libraries implements the routines for the digital controllers based on *difference equations*. Therefore, these routines need several parameters such as *set point (reference), control error, proportional gain* (K_p), *integral gain* (K_i), *derivative gain* (K_D), *sample time* T_s, among others. Our starting point in the design of the libraries of these controllers was structured in function of the steps that a student follows in the analysis and design of control systems:

1. The student converts the control plant to discrete using *Z transform* and choosing a suitable sampling time T_s, typical method for this process is ZOH (*Zero Order Holder*).

2. With the control plant converted in the *z domain*, the student designs his controller through some method, e.g., *root locus, lag or lead compensator (frequency methods), state state,* among others.

3. Typical result of the previous step is a control structure (proportional, proportional-integral or proportional-integral-derivative). The student programming this structure in a microcontroller or a microprocessor. For example, let's consider the structure of a PI controller in *Laplace* domain as follows [13]:

$$\frac{E(s)}{C(s)} = K_P + \frac{K_I}{s} \tag{1}$$

Where $E(s)$ is the error of process, $C(s)$ is the output of controller both in *Laplace* domain, K_p is the proportional constant and K_I is the integral constant. We converted the expression (1) into *z domain* applied the left side transform method, the result of this process is:

$$\frac{C(z)}{E(z)} = K_p + \frac{K_I z}{z-1} \tag{2}$$

We rewrite the expression (2) as follows:

$$\frac{C(z)}{E(z)} = (K_p + K_I)\left[\frac{z - \frac{K_P}{K_P+K_I}}{z-1}\right] = K\left[\frac{z-a}{z-1}\right] \tag{3}$$

The expression (3) yields to a difference equation which has the following structure:

$$C(kT) = Ke(kT) - Kae(kT-1) + C(kT-1) \tag{4}$$

4. Finally, the student implements the expression (4) and debugs the controller into the control plant reconditioned whether is needed, returning to step two.

Through the platform, the student adds the parameters of the controllers graphically into the software interface, saving time in the implementation as well as reducing the possible mistakes related with the programming [11]. The program generated for the controller is directly downloaded in the program memory of the *DSC,* then the student connects the sensors and the actuators to the development board checking the functionality of the controller. The principal components of the software structure are summarized as follows:

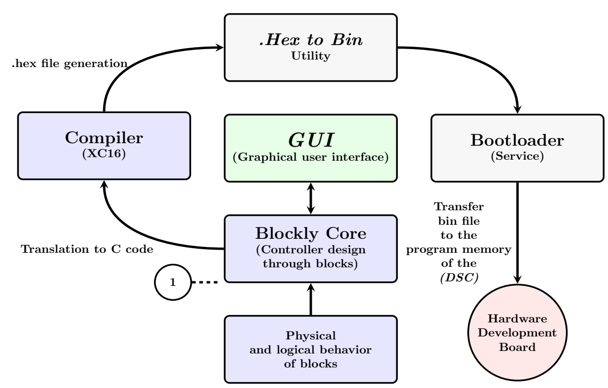

Figure 3. Overall scheme of GUI components used in the software structure.

1) .hex to bin utility and Bootloader service: The C code created by the graphical blocks, must be converted in terms of the program memory of the *DSC*. For this issue the compiler XC16 has an utility named (*.hex to bin*) conversion. The (*.hex file*) generated by this utility is downloaded to the *DSC* through the *bootloader*. This element is a small program which resides in the program memory of the *DSC*, allowing download the code into the program memory through some peripheral such as *UART (Universal Asynchronous Receiver Transmitter), CAN (Controller Area Network) or USB*[14]. In the present project, we have selected the *UART* peripheral due to ease of programming and versatility of it.

2) Physical and logical behavior of the blocks: In order to build each block employed in the interface, the students equipped to these blocks of two structures of code which define the physical and logical behavior of them. For the construction of the code structures, we established the following sequence:

- First, we created some functions which were the base of the functionality of our platform. The purpose of these functions is to allow to the designer have tools for interact with the development board and with the control design in itself. This fact guarantees that the interface will be adjusted to the needs of the design in a specific moment. Some of these functions are the following: Built in led, toggle led, PWM (*Pulse Width Modulation*), ADC (*Analog to Digital Converter*), UART (write text and write float number), proportional controller (P), proportional-integral controller (PI), proportional-integral-derivative controller (PID), among other functions.

- The previous operations are described in two files, one for the physical appearance, and the other for the logical behavior, both are written in JavaScript language, because of that *Blockly* engine has been created by means of this language. The process of compiling of the previous files generates a block with its respective code in C language, according to the specifications of the *DSC* and the compiler XC16 as it is shown below:

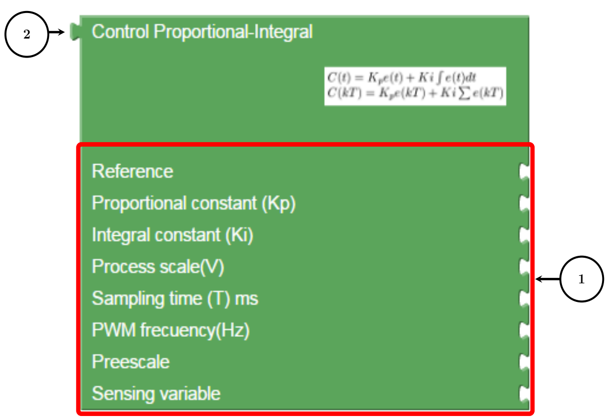

Figure 4. Physical appearance of the PI Controller. 1. Inputs for the control PI block, 2. Controller output which will be used by the students in their control processes.

Algorithm 1 Code returned by the block of Fig.(4).

```
ControlPi(Reference ,Kp ,Ki ,PWM period ,Sensvar ,Sample
        Time); //Proportional-integral control function;
```

The *ControlPi* function executes the PI controller with the sample period (T_s) specified by the user, in function of the constants (K_p, K_I) and the reference of the control process. The function returns as parameter the value $C(kT)$ specified in the expression(4).

B. Hardware structure

In accordance with the considerations mentioned in the table (II), the students of the hardware area followed a similar methodology to the software group. The first step in the hardware design was the selection of the hardware elements in function of some features such as robustness, current consumption, instruction velocity, number of remappable pins or number of peripherals. In accordance with the previous specifications, we have chosen as core of the hardware to the device DSC *(dsPIC 33FJ128MC802)* from Microchip Technology inc[8]. The steps established by the students in the hardware design were the following:

1) Identify the features of the *DSC* (number of ports, current consumption, instruction set, etc).
2) Design the low pass filters needed for the conditioning of the signals coming from of the sensors of a control process.
3) Establish of power consumption of the hardware devices selected in the design.
4) Do the respective tests needed in order to handle the DSC.
5) Do the tests of communications with the software interface using the *UART* peripheral.
6) Check the *bootloader* service downloading some programs into the DSC program memory.
7) Check the hardware performance through start-up of a control process.

These steps allowed to the students learn the basic concepts associated with the handling of the DSC. During the course of the project, were made the initial educative resources about the topics related with the DSC with the aim of helping to future students in the next stages of the project. With the purpose of having a global perspective of the hardware interface was built a general scheme which is shown in the Fig.(5). We have included several elements with asterisk (*) symbol; These elements are essential in the functionality of hardware structure. We divided by colors the different units that make up the development board. In color black, it is presented the power supply unit; in color green, the principal components of the *DSC* functionality; in color blue, the elements that interacting with the graphical user interface and the control process.

An important feature of the hardware that we have used is the concerning to the capacity of the reconfiguration of functions over certain ports in the *DSC*. This means that in a specific moment one peripheral can be assigned to a *pin A*. If the design requires that this same peripheral will be assigned to a *pin B*, this issue can be solved through the use of the remappable pins in the *DSC*. To reduce the current consumption in the power supply and obtain a good efficiency in the design, we have chosen a DC-DC buck converter *(LM2576)*, the voltage tolerance of this device is $\pm 4\%$, and its efficiency (η) in critical conditions $(V_{in} = 12V, I_{out} = 3A)$ is 75%[15]. The Fig.(7) illustrates the design of the PCB (*Printed Circuit Board*) in base to the elements mentioned before.

For the design the input filters, we have selected the rail to rail operational amplifier *MCP6004* [16]. The cutoff frequency

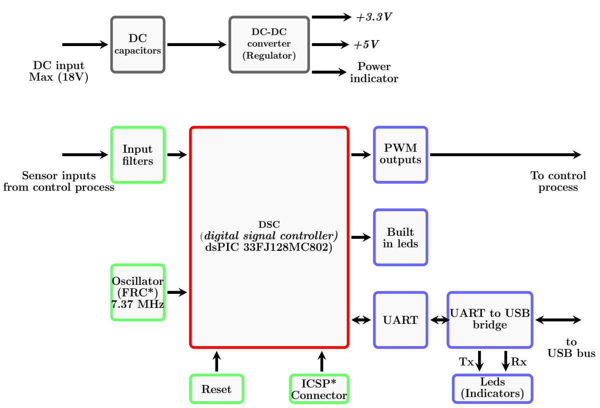

Figure 5. General hardware scheme of the development board.

selected was $f_c = 10KHz$ [17], due to the need to reduce the noise coming from sensor signals of a control process.

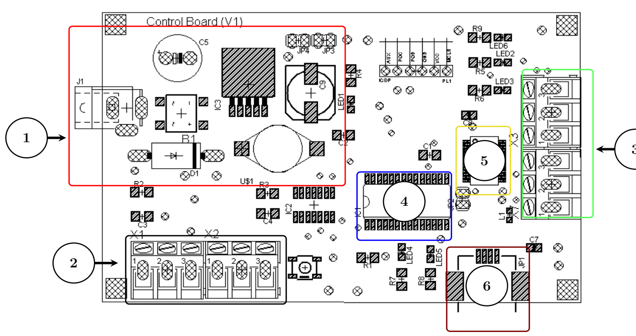

Figure 7. General view of the development board. 1. Power supply unit, 2. input connectors, 3. output connectors, 4. *DSC*, 5. *USB* bridge, 6. *USB* connector.

IV. Results

In consonance with the framework associated with the cooperative methodology employed in the concept of the platform shown in section II and the technical aspects described in the section III, we present the results associated to these topics which will be divided in technical and educational results.

A. Technical results

The Fig.(6) shows the GUI designed by the students of the software group, in accordance with the levels shown in the table (III). This is the final level after the process of debugging. In the user's toolbar the students have all blocks needed in the construction of their controller. In this pallet the student will find the following blocks divided by sections:

$$Controllers \begin{cases} Proportional & (P) \\ Proportional - integral & (PI) \\ Proportional - integral - derivative & (PID) \end{cases}$$

$$Plant\ identification \begin{cases} PBRS \\ (Pseudo\ Binary\ Random\ Signal) \end{cases}$$

$$Input - Output \begin{cases} Led & (High - low) \\ Led & (Toggle) \\ ADC & (Analog\ to\ Digital\ Converter) \\ PWM & (Pulse\ Width\ Modulation) \end{cases}$$

$$Communications \begin{cases} UART & (Send\ integer) \\ UART & (Send\ text) \\ UART & (Send\ float) \end{cases}$$

$$Logic \begin{cases} If & \\ Comparation & (=, >, <) \\ Boolean & (and, or) \\ Boolean & (true, false) \end{cases}$$

$$Loops \begin{cases} for \\ while \end{cases}$$

$$Variables \begin{cases} Declaration \\ Assign\ value \\ Change\ name \end{cases}$$

$$Math\ operations \begin{cases} Sum \\ Substraction \\ Multiplication \\ Division \\ Exponentiation \end{cases}$$

Each one of the previous sections were designed taking into account the considerations established in the methodology of development of the platform.

If the student wants to make other type of controller, this has two options: to recompile the core of the platform, including the function of this controller or build the controller based on the blocks designed in the interface.

Fig. (8) shows the different components of the development board according to the description of the Fig.(7). This board was tested with the different blocks described before, checking their operation in each case.

B. Educational results

As mentioned in the beginning of this paper, our starting point in the conception of the platform was the lack of open source tools oriented on automatic control area in our department of electronic technology. This fact was demonstrated through a survey made to academic community of our program (7 teachers and 9 students of the automatic control subject, $n = 16$). The respectively outcome is shown in the figure (9). Based on this information, it began with the development of the platform, first selecting to the students who would participate in the development of the platform as follows: Five students were selected for work in the development of the platform, three in the software area and two in the hardware area. These students were selected in function of several elements associated with the skills in the mentioned areas as likewise searching a heterogeneous group, that is to

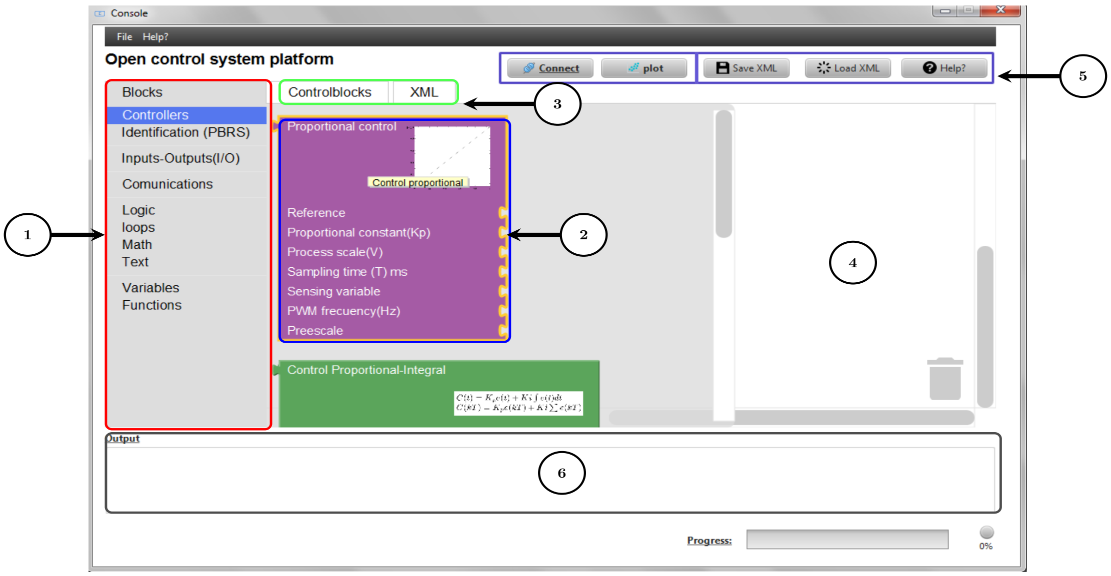

Figure 6. General view of the user's interface. 1. User's toolbar, 2. Digital controller block, 3. Translation code tab, 4. Student workspace, 5. Menu toolbar and buttons, 6. Command prompt.

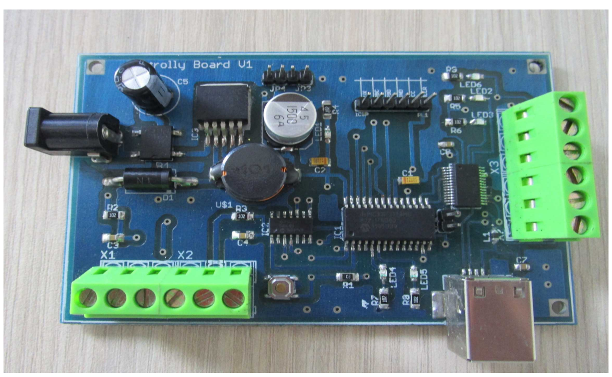

Figure 8. Development board designed by the hardware group.

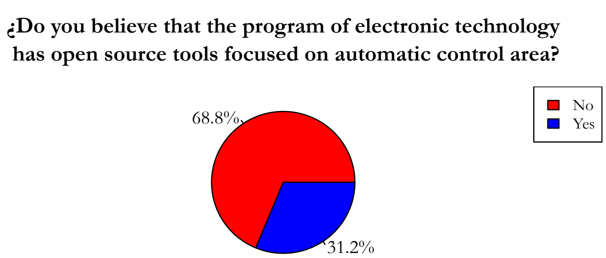

Figure 9. Perception of the open source tools focused on automatic control area in the program.

students, the results of previous premise are addressed in the Fig. (10) and (11) respectively.

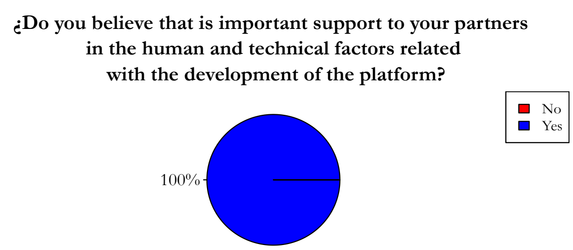

Figure 10. Perception of the *group processing* in the students who participated in the platform development.

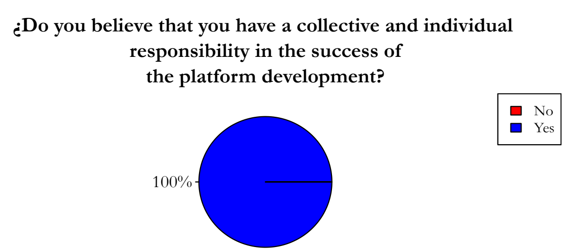

Figure 11. Perception of the *individual and collective responsibilities* in the students who participated in the platform development.

say, a group with similar features cognitive and emotional. The work of the students during the development of the platform was focused in order to achieve the main components of the cooperative learning methodology: the *positive interdependence*, the *responsibility* (collective and individual) and the *group processing*[1]. Once the work was concluded, some questions were established so as to measure if effectively the methodology had relevance in the educational process of the

The results shown in figures (10) and (11) were obtained making that the students reflected on their role, both technical and human, in the design of the platform. The structure of tasks by levels allowed that the students share experiences and knowledges during the development of hardware and software components improving the cognitive skills needed in the solution of a determined problem in the engineering area. Nevertheless, both groups did not advance in the same way, e.g., software group advanced more slowly in the design than hardware group. This issue is due to the nature of our program, there are not subjects related with the software development. To solve the previous problem, it created some educational resources focused on the software aspects, specially associated to the *Java* language programming. These contents made emphasis in the handling of classes, methods and objects as likewise in the concepts concerning to these topics such as hierarchy or polymorphism. At the same time, it worked with the hardware group in the programming of the *DSC* and the handling of it. These tasks were made simultaneously allowing a leveling in both areas.

Another element that we wanted to prove in the methodology is the related with the skills acquired by the student during the development of the platform, the students were inquired about these skills (*human and technical*). Their responses are summarized in the following table:

Table IV
SUMMARY OF THE SKILLS ACQUIRED BY THE STUDENTS DURING THE DESIGN OF THE PLATFORM.

Component	Skill
Human and cognitive	1. Fellowship.
	2. Support both tasks and work.
	3. Responsibility in the assigned tasks.
	4. Divergent thinking.
Technical	1. Programming, algorithmic thinking.
	2. Control systems analysis.
	3. Digital control design.
	4. Hardware design.
	5. Hardware prototyping.

With the intention of enhancing the educational process structured in the platform, it asked to the students who developed the platform about the methodology followed during its design. Some of their opinions were summarized in the following comments:

Student 1: *I like the software development. The methodology helped me to understand the concepts associated with this area. The methodology allows to progress with the aim of accomplishing with the objectives established.*

Student 2: *The platform is a didactic tool that will allow to future students the design of digital controllers by different suitable techniques improving their performance in this area.*

With the aim to receive the feedback of the academic community of our program (students of the automatic control subject and teachers), we presented to them the results obtained, some of their comments are synthesized below:

Students reviews

Student 1: *The research incubator helps to that theoretical components of the career will be used in the practice applying them in a project that benefits to future students.*

Student 2: *The project is a good work that encourages the motivation of the students of several semesters. The software is very comfortable and didactic in the learning of control systems.*

Student 3: *The platform helps to the interaction of new knowledges in the control area which allow to win experience in the educational and labor fields.*

Student 4: *The platform is a technological advance that allows to the students share their knowledges and ideas in team, this is something of value and innovative.*

Teachers reviews

Teacher 1: *These types of platforms allow the strengthening, implementation and appropriation of the knowledge without the limitation of academic spaces and fixed schedules.*

Teacher 2: *The open source tools such this allows to the student to improve his educational process, the self learning, and the modification of technical features in function of his needs.*

Teacher 3: *The platform allows to practice and reinforce the knowledges as likewise solve doubts of the theoretical aspects.*

The students recognize that their partners and friends made an excellent job in the design of the platform which will help them to the understanding of different concepts related with automatic control area, this process allows to improve the motivation and interest of the students in the learning associated to the control automatic area as well as by participate in the future stages of the project.

V. CONCLUSIONS AND FUTURE WORK

In the present work, we have presented the conceptual and practical design of a platform focused on automatic control area through the use of the cooperative learning methodology. This methodology allowed to contextualize to the students in different aspects of their educational process (human and technical) through the definition of tasks with common objectives and goals in the group. The methodology was joined to the technical aspects involved in the construction of the platform, allowing that during the development of it, the students interacted with each other, sharing experiences and solving difficulties, reinforcing the *group processing*.

We concluded that *positive interdependence* is achieved as long as the group understands the role that each participant has in the success of a certain aim or goal. All students helped in the design of the platform in spite of limitations in time, knowledges or budget that the project had in a specific moment. The process followed in the concept of the platform, provided to the students elements both technical and human for their personal and labor life. A limitation of this process is the referred to the size of the group that reduces the scope of the methodology. However, this problem can be solved if the group is divided in further technical or educational areas needed in a specific project.

In a future work, we will want to establish two aims: first, to validate the platform between different groups in the automatic control subject with the aim to improve the performance associated to it. The validation would be addressed in the overall aspects both educational and technical. The second aim of technical sort, it will consist in migrate the software interface to mobile platforms such as *Android*. This fact will allow to extend the educational coverage of the platform as likewise to improve the portability of it.

Acknowledgment

The author wishes to acknowledge to Uniminuto University for the financial support of this project. The author also wishes to extend their acknowledge to the academic community of the Electronic Technology program specially to the students of the automatic control research incubator and the automatic control subject. For all them, thanks for their help. Finally, the author acknowledges specially to the Professor Susana Gómez for her advices on the structure of the present paper.

References

[1] R. J. David W. Johnson, *El aprendizaje cooperativo en el aula*, P. SAICF, Ed., 1999.

[2] R. T. J. David W. Johnson and K. A. Smith, “Cooperative learning: Improving university instruction by basing practice on validated theory,” *Journal on Excellence in University Teaching*, 2013.

[3] C. C. R. Juan de Dios Arias Silva, *Aprendizaje cooperativo*, U. P. Nacional, Ed., 2005.

[4] K. Koffka, “Principles of gestalt psychology,” 1935.

[5] G. O.Grow, *Teaching learners to be self-direct*. Florida A&M University, 1996.

[6] S. University, “Problem-based learning,” *Speaking of teaching*, vol. 11, 2001.

[7] C. A. Picard, “Learning about learning: The value of insight,” *Conflict resolution quarterly*, vol. 20, 2003.

[8] M. T. Inc. (2011) dspic33fj32mc302/304, dspic33fj64mcx02/x04 and dspic33fj128mcx02/x04 data sheet. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/70291G.pdf

[9] Oracle. (2014) Javafx. [Online]. Available: http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784

[10] Google. (2015) Blockly developer. [Online]. Available: https://developers.google.com/blockly/

[11] J. Alvarez, “Controlly: Open source platform for learning and teaching control systems,” in *Automatic Control (CCAC), 2015 IEEE 2nd Colombian Conference on*, Oct 2015, pp. 1–6.

[12] (2014) Mplab xc16 c compiler users guide. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/50002071E.pdf

[13] N. S. Nise, *Control Systems Engineering, 6th Edition*, Wiley, Ed., 2011.

[14] (2013) ds30 loader main manual. [Online]. Available: http://www.ds30loader.com/images/files/ds30-Loader-main-manual.pdf

[15] (2013) Lm2576/lm2576hv series simple switcher 3a step-down voltage regulator. [Online]. Available: http://www.ti.com/lit/ds/symlink/lm2576.pdf

[16] (2009) Mcp6001/1r/1u/2/4 (1 mhz, low-power op amp). [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/21733j.pdf

[17] (1999) Anti-aliasing, analog filters for data acquisition systems. [Online]. Available: http://ww1.microchip.com/downloads/en/AppNotes/00699b.pdf

[18] P. Dalgaard, *Introductory statistics with R*. Springer, 2008.

[19] P. Gopal, “Using cognitive apprenticeship framework in technology education by pre-service teachers,” *IEEE international conference on technology for education*, 2011.

[20] J. G. Greno, *A Handbook of Educational Psychology*. MacMillan, 1996.

[21] K.Ogata, *Sistemas de control en tiempo discreto*. Prentice Hall, 1996.

[22] M.S.Fadali, *Digital Control Engineering, analysis and design*. Elsevier, 2009.

[23] (2016) Integrated development and learning environment (idle). [Online]. Available: https://docs.python.org/2/library/idle.html

[24] (2016) Python language specifications. [Online]. Available: https://www.python.org/

[25] (2008) Livegraph plotter. [Online]. Available: http://www.live-graph.org/