

Project Based Learning on Industrial Informatics: Applying IoT to Urban Garden

Javier Hormigo and Andrés Rodríguez

Department of Computer Architecture, University of Málaga,

29071 Málaga, SPAIN

Emails: fjhormigo@uma.es , andres@uma.es

Abstract—The fast evolution of technologies forces teachers to trade content off for self-learning. PBL is one of the best ways to promote self-learning and simultaneously boost motivation. In this paper, we present our experience introducing project-based learning in a last year subject. New Internet of Things (IoT) topic allows us to carry out complete projects, integrating different technologies and tools. Moreover, the selection of open-source and standard free technologies makes easy and cheap the access to hardware and software platforms used. We carefully have picked communication, data management, and programming tools that we think would be attractive to our students. They can start making fast prototyping with little initial skills and, at the same time, these are serious and popular tools widely used in the industry. In this paper, we report on the design of a project-based learning for our course and the impact this has on the student satisfaction and motivation. Surveys taught us that tuning the courses towards developing real projects on the field, has a large impact on acceptance, learning objectives achievements and motivation towards the course content.

I. INTRODUCTION

This work presents the implementation of a Project Based Learning (PBL) methodology in the subject of Industrial Informatics. This is a 60-hour course corresponding to the fourth year of Electronic, Robotic and Mechatronic Engineering degree in the Industrial Engineering School at Universidad de Málaga. This year, there were about 40 students in the course. Although the number of students has been growing exponentially in the past (because the recent degree implantation), we think that it is not going to grow much more next years. The presented project belong to a more general Innovative Education Project (PIE), which aims to generalize the use of the same hardware platform, specifically the Raspberry Pi, through different courses of the same degree.

There are multiple reasons for selecting a PBL methodology. First, taking into account the fast evolution of technology in this field, preparing the students for self-learning should be a priority for any engineering curriculum. Second, being a last year subject, a more practical approach is sought. Third, it has been demonstrated that PBL greatly improves motivation of students [1]. This quality is very important since our students are typically reluctant to embrace self-learning. On the other hand, teamwork skills are also fundamental for any engineer nowadays. That is the main reason why cooperative learning has been also included in the course. Besides that, the student can accomplish a complete project by working in groups and it also facilitates to monitoring the work progress by the teachers.

Internet of Things (IoT), together with cyber-physical systems, cloud computing, and machine learning is the foundation of the new industry, which has been called "Industry 4.0". In this course, we have the opportunity to introduce most of these ideas to the students through Industrial Informatics. To be really effective, the PBL methodology requires of a credible and tangible project. We thought that an urban garden could be an ideal place to design a cyber-physical system by deploying several IoT nodes. These IoT nodes could gather data from the garden and also perform some actuation. The collected data could be also processed in the cloud and machine learning could be used to take the different decisions.

We knew that the faculty of Biology has an urban garden for teaching and research, and we thought inter-discipline collaboration may be also an interesting way for enriching the experience. Therefore, we arrange a cooperation with the Plant Biology Department to launch a project that could be useful for both, their students and ours. In a first stage, our students would design a system mainly to collect and monitor information from the urban garden. In a second step, biology students will use the system in their subjects and they will provide feedback and test information about it. In a third step, our students (the next year) will use the feedback to improve the system and also to include actuation capabilities on it. This refinement process could be repeated for several years.

Another important aspect of the project is the use of accessible open source software/open hardware platforms. The extensive community of developers who use these open systems offers a large number of online examples, tutorials, videos, etc. These are materials that our students prefer better than classic textbooks or reference manuals. In addition, the use of such accessible hardware platforms stimulates natural curiosity and desire to try and experiment on their own in our students. Many of them show interest in acquiring these materials for using it at home on small do-it-yourself projects.

II. PROJECT DESCRIPTION

In this section, we describe the project and how we have implemented it in our course. The project statement was given to the student after the first three weeks where they had been working on more general topics about industrial informatics. Then, they had about twelve weeks to develop the proposed project. First, let show you the project statement that we gave to the students.



Fig. 1. Urban garden at Science School facilities

A. Project statement

The Department of plant biology of the University of Málaga manages an urban garden in the Faculty of Sciences (Fig.1) to study about this and other topics. So far the collection of data required for these studies is performed manually by the researchers or students of related subjects. With this project, we want to improve the process of collecting and processing data through the use of "Internet of Things". For this reason, we want to develop a system based on a network of "sensor nodes" able to collect useful data for their studies, and a "base station" that receives, processes, and displays the information to these researchers in a user-friendly way. The system is completed with "actuator nodes" that can receive orders to act on the environmental conditions of the cultivation (irrigation, lighting, ventilation,...).

In a first phase, we will focus especially on the system for data collection and representation of the information. In a second phase, we will focus on the control system to act on cultivation according to the collected data. The first phase will be carried out by the students of Industrial Informatics course 2017-18, while the second phase will be performed by students in the following year. Therefore, it is fundamental to generate a well-structured system and a complete and thorough documentation, for both users and maintenance or further development.

1) *Description and system requirements:* The project will consist of designing and implementing a system for monitoring and control of urban gardening. Such a system shall include the following elements:

- Two types of IoT nodes based on the module Wi-Fi ESP8266. The first, the "sensor node", may lay or stand on any part of the garden and will measure, store temporarily and transmit wirelessly, fundamental parameters (temperature, humidity, light exposure...). The second, the "actuator node", will be used to control solenoid valves from irrigation and other actuators, receiving orders via the wireless communication network.
- A "base station" based on the Raspberry Pi will implement a SCADA system and a database. The "base station" will serve also as a message broker for IoT

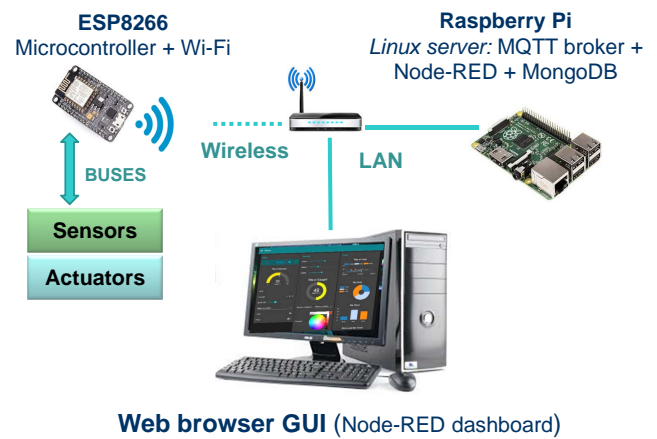


Fig. 2. Project system main architecture

communication (using MQTT Protocol) and it will receive the information from the "sensor nodes" (we must assume a network of several sensors). The "base station" is responsible for displaying the gathered data to the users, store and upload them to the cloud. This device may have a wired connection.

- A graphical web interface for interaction with the users. This will be programmed using Node-RED.
- The "base station" will also need a database to store the measured data and accept requests to retrieve that information. Then, it must allow analyzing and displaying these data. A NoSQL database system, called MongoDB, has been already installed. It allows simple and direct storage of documents in JSON format.
- The following services have already been installed in the RaspberryPi and they are already running: an MQTT broker (mosquitto), Node-RED and the NoSQL database MongoDB. In the laboratory, the Raspberry IP address will be 172.16.53.xx, where xx is the number that appears on the board ID (on the Ethernet connector). The ports where those installed services respond are the standards:

TABLE I
STANDARD TCP PORTS USED FOR THE DIFFERENT SERVICES

Service	Address in LAN/URL (IP: port)
mosquitto (MQTT)	172.16.53.xx:1883
Node-RED	http://172.16.53.xx:1880
MongoDB	172.16.53.xx:27017

The functional requirements of the system are:

- The system must collect and store at least the data of temperature and humidity, sunlight exposure and soil moisture.
- The last values and also the stored ones must be displayed through a graphic interface. The interface must allow providing queries to sort and represent graphically those values. It will analyze the extracted data (average values, standard deviation, boundaries, etc.). It must be allowed

downloading the raw data to a file for analysis with other tools.

- The user will be able to select the frequency in which data for each parameter will be collected.
- All nodes will communicate with the "base station" using a common interface in the same network (MQTT messages with content formatted in JSON).

The non-functional requirements of the system are:

- **Autonomy.** The system must be maintained with as less as the possible human intervention. To do this, the power consumption in the nodes will be a very important part. Different ways of energy saving should be used, thinking that the nodes will operate with batteries which will be charged by solar panels.
- **Compatibility.** The system must allow the coexistence of nodes of different types and from different development team (working group), as well as different base stations. To do so, it will be necessary to define a standard interface of communication for all IoT nodes (which commands are they going to accept, the format of those, which sensors values are going to be transmitted, ...).
- **Accuracy.** The system must provide reliable and synchronized data. Thus, the different sensors must be calibrated to provide consistent data on standard measurement units and the nodes must be synchronized in time and date.
- **Security.** Although the security of the system is not a key aspect in this preliminary phase of the project, a study of possible vulnerabilities and a proposal for security controls must be included.
- **Friendliness.** The end-user of the system may not have any knowledge of electronics or computer science. Hence, the user interface and the routine operations of maintenance and use should be very simple. They have to be perfectly explained in a user's manual.
- **Maintenance.** It must be easy to repair or expand the system, because, for several years, it will be used and maintained by people other than those who designed it initially. For this reason, all the elements of the system (both hardware and software) must be perfectly documented.

2) *Work plan, results and project evaluation:* Below is the proposed work plan:

1. The technical description of the two types of IoT nodes. A document showing which sensors are going to be used, how are they going to be connected to the ESP (number of pins, communication protocol), which functions are going to be developed for communication with sensors, actuators, and base station. Furthermore, an initial budget of the nodes. (week 6)

2. The first prototype of the IoT "sensor node". A functional prototype of the node which must be able to control all sensors and perform data collection. However, the communication with the "base station" does not have to be implemented yet. (Week 7)

3. The base station nodes interface (MQTT). A document

defining the messages that will be used to communicate the nodes with the base station, the protocols, and the syntax. (Week 8)

4. The technical description of the whole system. (week 9)

5. The Testing, calibration and security plan of the whole system. Document describing which tests will be done to calibrate and check the correctness of the system. Furthermore, It must describe how to detect possible security problems and propose some solutions. (Week 10)

6. The final prototype IoT nodes (sensor and actuator) and the first prototype of the base station. (week 12)

- The IoT nodes (with whole functionality) which communicate with the base station according to the specified interface. Also the complete documentation of the node.
- the base station which accepts messages from the nodes, check and store them, etc. It is not necessary to have the full user interface.

7. The complete prototype (nodes-base station-SCADA). A demonstration-presentation, development and user documentation. (week 14)

8. The individual extension. Each component of the development team, on an individual basis, must perform an extension/modification of their own system, based on the proposal of the teacher. (week 15)

Based on the qualifications of the delivered results, the score of the project shall be calculated using the following weights distribution:

- The first prototype of the "sensor node". 10% (group score)
- Plan of testing, calibration, and security. 10% (group score)
- Final sensor and actuator nodes. 30% (group score)
- Complete project. 30% (group score)
- Individual extension. 10% (individual) + 10% (if the whole group passes it)

3) *List of available components:* In the laboratory, there are the following elements which can be used at the discretion of each group to design their system:

- Two NodeMCU DevKit v1.0 (ESP8266)
- A RaspberryPi B+
- Digital temperature and humidity sensor (DHT11 / DHT22 serial bus with its own protocol)
- (DS18B20 with One-wire bus) digital temperature sensor
- (Analog output) TMP36 temperature sensor
- (Analog output) soil moisture probe
- Light sensor (SPI bus / photoresistor)
- ADC Converter 4 channels (ADS1015 with I2C bus)
- Real time clock(ZS-042 with I2C bus module)
- Rechargeable battery (3.7V 1100mah)
- Solar panel (6V 1W)
- Battery Charge controller
- Relays (5V)
- Small material: connection wires, NPN transistors, resistors, capacitors, leds, buzzers, push-buttons,...

These elements are enough to meet the minimum requirements of the system. However, besides these, the groups could decide to buy or build other elements that they believe necessary for their system.

B. Involved Technologies

As we mention above, the project is developed using several open source software programs and open hardware platforms. Here we present them.

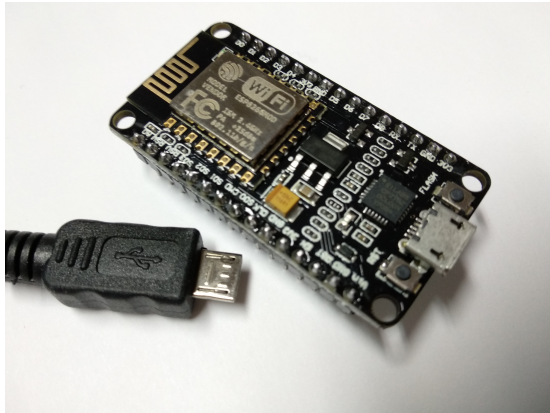


Fig. 3. NodeMCU DEVKIT v1.0 Wi-Fi developing microcontroller board

1) *ESP8266 platform* : For IoT sensor and actuator nodes implementation we use the Wi-Fi enabled NodeMCU DEVKIT V1.0 [2] (Fig.3), an open hardware development board based on ESP-12-E (4MBytes flash version). This board has an ESP8266EX core, a microcontroller that integrates a Tensilica L106 32-bit RISC processor, which achieves extra-low power consumption and reaches a maximum clock speed of 160 MHz. The Real-Time Operating System (RTOS) and Wi-Fi stack allow about 80% of the processing power to be available for user application programming and development. The board is programming using Lua scripts thanks to the open-source firmware NodeMcu[3], that provides a complete set of ready to use modules [4] and helps developers to prototype their IoT product within a few code lines.

Once the firmware is flashed using the convenient onboard USB interface, the Lua scripts are easily uploaded to the board flash using the development environment (ESPlorer IDE [5] Fig.4). the user can even run Lua sentences from an interactive console. Results and messages are printed on the console while connected through USB interface. Thus, fast prototype debugging is easy.

This ESP8266 based boards can also be programmed using C language and the same tools and IDE used for Arduino's family microcontrollers.

2) *Raspberry Pi platform:* For monitoring, data acquisition and control programming we use the well-known Raspberry Pi platform [6]. The Raspberry Pi boards are running the latest Raspbian operating system (a Linux distribution) and for programming our IoT applications we use Node-RED [7], a flow-based programming tool for the Internet of Things. Node-RED programming tool and deployed programs are running on

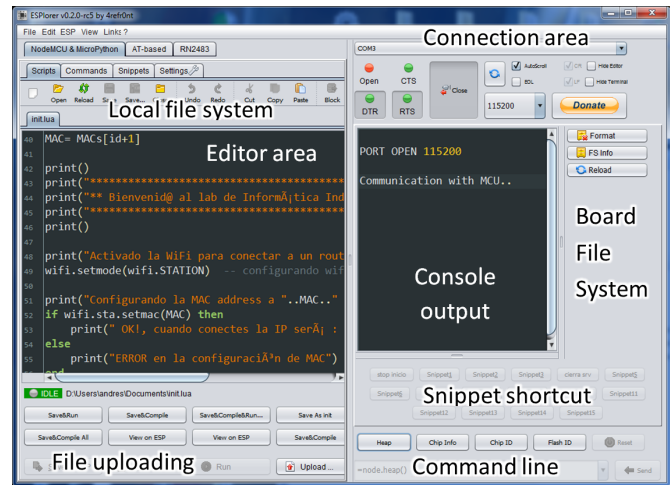


Fig. 4. ESPlorer IDE

the Raspberry Pi, but we also use the board to run a database server (MongoDB) for managing all data gathered from IoT nodes and an IoT communication service (an MQTT broker) for connecting all IoT nodes. All nodes are connected using a standard TCP/IP connection over Wi-Fi and a wired Ethernet connection in the case of the Raspberry Pi board.

3) *MQTT protocol*: It's very easy to install your own IoT message broker on the Raspberry Pi using the open-source MQTT broker Eclipse Mosquitto [8]. MQTT [9] stands for Message Queue Telemetry Transport and is a machine-to-machine (M2M) Internet of Things connectivity protocol. The MQTT protocol provides a lightweight method of carrying out messaging using a publish/subscribe model. This makes it suitable for the Internet of Things messaging such as with low power sensors or mobile devices such as phones, embedded computers or microcontrollers. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.

MQTT says nothing about the format in which the data should be sent, we have chosen using JSON (JavaScript Object Notation) [10] to format our messages because is a very compact and easy to manage format and is native for Node-RED, our programming tool.

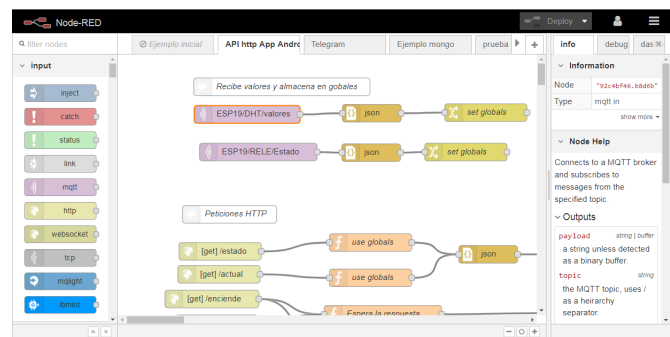


Fig. 5. Node-RED programming environment

4) *Node-RED programming tool*: Node-RED is an open-source flow-based programming tool for wiring together hardware devices, APIs, and online services in new and interesting ways [7]. It provides a browser-based editor (Fig.5) that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single click. Most of the programming can be achieved without writing a single line of code, but users also have the flexibility of programming the behavior of a general "function" node using JavaScript code. It has direct support for MQTT communication with nodes for publishing (sending data) and subscribing (receiving data) on MQTT topics. It also has nice support for MongoDB database, where data represented in JSON can be directly stored in the database and can be queried later on. Data structures are managed by Node-RED as JavaScript objects so their translation to JSON is flawless.

Node-RED also offers a set of nodes to build a dashboard, a very easy way to program an interactive web user interface [11]. You can show data from sensors graphically (gauges, charts, etc.) and also receive commands from users (buttons, selectors, sliders, etc.). It makes it ideal for programming the monitoring and data exploration application for our IoT project (Fig.6).

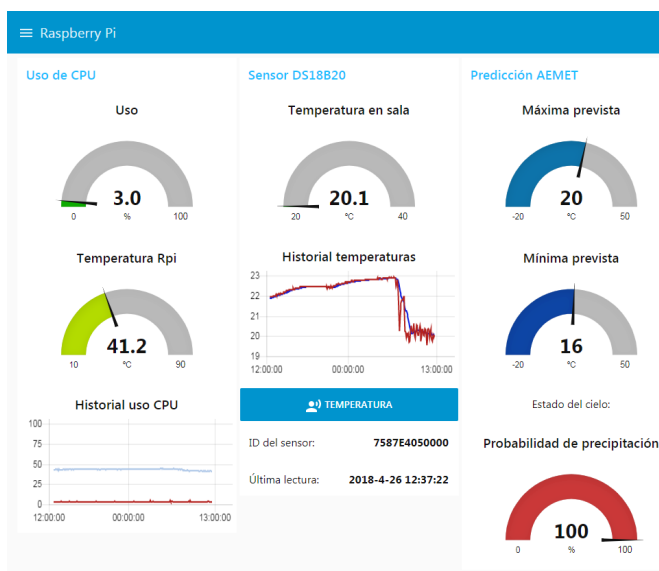


Fig. 6. Node-RED dashboard UI example

5) *MongoDB database*: We have chosen this free, open-source, No-SQL database to manage all data gathered from our IoT nodes. MongoDB [12] stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time. It can store directly the data managed by Node-RED represented in JSON, making the use of the database very easy for students who are not familiar with the classic SQL database design process.

III. COURSE PLANNING AND DEVELOPED ACTIVITIES

This course is organized in 15 weeks with 4.5 hours of in-person class. We concentrated the sessions such as there is only one session per week. The students were divided into groups of their own election. Each group had three or four students and they had to develop a complete system. First, they worked on more general topics about industrial informatics for three weeks. Then, the project was developed in the last 12 weeks. We have prepared some introductory exercises for presenting and getting an initial insight into each technology and tool, but not further teaching material preparation was needed. There are plenty of examples, tutorials, videos, etc. available online that the students can use for learning about each topic. Moreover, the students seem to prefer this kind of online community supported materials over the classic and formal textbooks and manuals.

The evaluation of the course was defined following [13] in this way: 15% for handing in all request on time (failing more than 20% of requests means not passing the course); 25% from a test of basic knowledge which is compulsory to pass with an A, but there are several sittings (this part is grade according to the number of sitting utilized); 50% the grade of the project; and 10% participation in class and within their own groups.

In the first four weeks of the project, the students had to focus on developing the "sensor node". We provided technical documentation and links for the ESP platform and all sensors. We also organized a jigsaw to allow the student to acquire the fundamental concepts required to use the ESP platform with nodeMCU and Lua. Furthermore, the documentation we asked for in this period, was intended to provoke they stop working to plan and think about the system they had to develop. At the end of this period, they showed us the first prototype of the sensor node. It was able to read the information from a few sensors.

In the following two weeks they were focused on the communication between the sensor node and the base station. Along with the references for learning about MQTT, JSON, node-RED, and MongoDB, a new jigsaw was organized with this content. Groups were asked to propose a syntax and protocol for communicating IoT nodes and base station. Their proposal should have been used to define a standard protocol for all groups. However, during the process of discussing this standard, we realized that the activity was very time consuming and students did not participate as expected. Hence, in the end, we had to define the standard protocol based on the ideas proposed in the class. We also used a guided exercise to interconnect IoT nodes with the base station through a wifi network. Nevertheless, although the process to program the ESP-12E module for connecting it to the wifi network was thoroughly explained in the exercise, they had serious problems to achieve this goal. Those were mainly caused due to the event-driven nature of the Lua programming language.

During the following three weeks, they developed the communication between the sensor node and the base station using the messages and very simple protocols defined in our

standard. They used Lua to program the IoT nodes to send messages to the base station and receiving other messages from it. At the same time, they used node-RED at the base station to process the received messages from the IoT nodes and send others messages back. At the end of this period they had to hand in a prototype of the system which was peer-reviewed into different ways: using the sensor node and base station of the same group, but also using the sensor node of one group with the base station of the group that was evaluating. The latter test reveals that many prototypes did not fulfill the standard, mostly because many groups did not respect the key names for the key-value pairs within the messages. In this period of time, another important aspect they started to investigate was how to use the deep-sleep mode of the ESP8266 to save energy.

After Christmas holidays, they had the first sitting of the test of basic knowledge. This test consisted in a few very simple theoretic questions and basic programming exercises using Lua and node-RED. On the other hand, they had two weeks to finish the project. At this point, every group was dealing with their own problems. In the second last week of the course, they presented their project and showed the working prototype, in front of the teachers and their classmates. They also had to answer some general questions about their design. In the last session, we asked for an extension of their own project on an individual basis. We proposed the same extension for all student and they had to suggest the required changes in both the IoT node and the base station. This exercise was performed in class as an exam. Although this exercise seems completely individual, it should actually encourage to help the group-mates because there was a component of the mark which depended on the fact that all the members of a group pass the exercise. In practice, if this happened, they double the mark. However, only two groups achieved this goal.

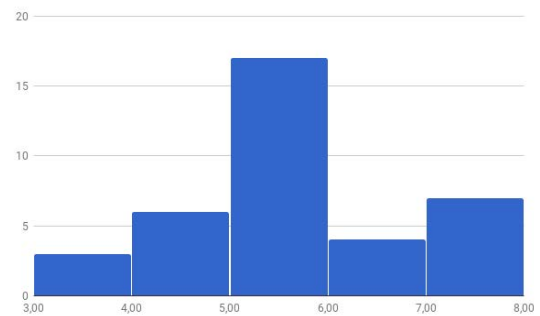
IV. RESULTS AND DISCUSSION

The results have been bittersweet. Following we present several facts about the obtained results.

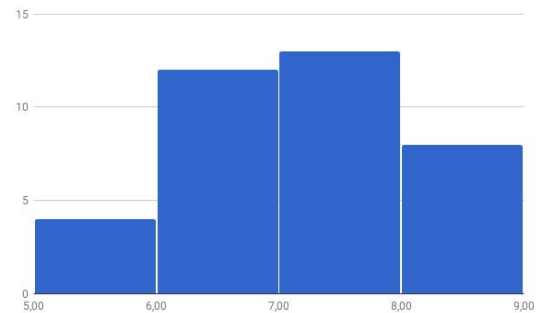
Fig.7a shows the distribution of the marks obtained in the project for each student. Most of the students handed in just acceptable projects and even one group a borderline one. Furthermore, the marks of the projects went down because only a few of the students accomplished the individual extension.

After including the rest of components most students improved their marks, Fig.7b shows the distribution of the final marks obtained in the course for each student. Only about 50% of student passed the basic test in the first sitting, although most of the rest of students only failed in one part. About 40% of them passed the test in the second sitting and only 3 students (8%) passed the test in the third sitting. In the end, none failed the basic test.

Only 2 students withdrew from the course. One of them failed ill and had to quit. In both cases, that happened at the beginning of the course. There is also one student that never attend a class.



(a) Project



(b) Course

Fig. 7. Distribution of the final marks obtained for the project and the course

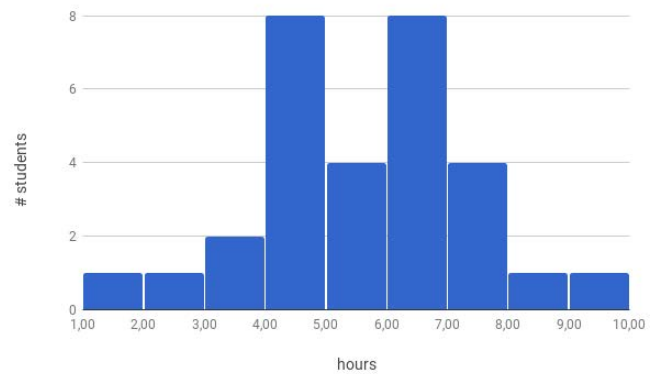


Fig. 8. Distribution of the number of hours out of class per student

Another interesting information is the number of hours out of class that each student used in this course. Near to the end of the course, we asked them to estimate anonymously the mean of the hour per week they had spent in this course at home. The result of the survey is shown in Fig.8. According to the number of ECTS, they should spend about 6 hours per week and most of them said to use between 4 and 8 hours. Even so, the mean, 5.2 hours, is a little low and extreme cases are too disperse for being satisfactory. Hence, some measures should be added to correct this in next courses.

A. Student perception

We have conducted three quick surveys along the course. The student had to write anonymously the best and the worst aspect of the course so far in two minutes. The first survey was conducted after providing them with the project statement. They had time to read the document and discuss the project with their teammates. The results of the survey are summarized in Fig.9. In Fig.9a, it is easily observed that they were excited about the project, especially with the fact that it sounds (and it was) a real project. They appreciate that the project could be useful to someone. The topic and technologies required also appeals to them and less the methodology and freedom. However, they are also worried about the amount of work required to develop the project and the necessity of learning new technologies. It is alarming that 25% of student thought that "learn new content" is the worst thing of the subject, when they supposedly enrolled in this course to learn this content. On the other hand, almost 20% of the students do not see any negative aspect.

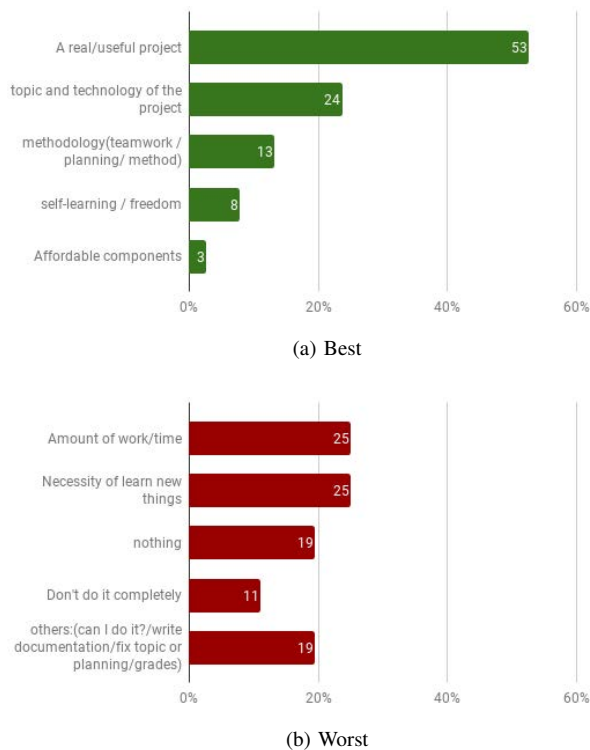


Fig. 9. Student perception after giving them the project statement

The second survey was conducted after working on the project for several weeks. The results are shown in Fig.10. The students were working on the sensor node using Lua as the programming language. Most of them were having important difficulties to debugging their codes. The project is still in the first position as the best of the course but self-learning is close behaving being claimed by 31% of the students (see Fig. 10a). In contrast, another 33% of students consider self-learning and the lack of enough help as the worst of the course (see Fig. 10b). That could be explained by the fact that the first group

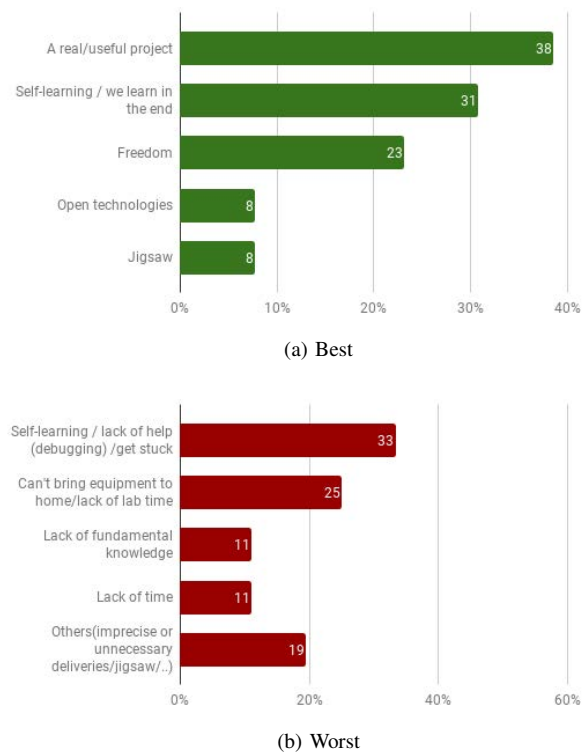


Fig. 10. Student perception after working on the project several weeks

of students had overcome the steps described in [14] but the second group had not. They consider that the second important problem is the fact that they could not bring the equipment to home and the lack of time in the laboratory. This topic was a continuous source of complaints.

The results of the third survey are shown in Fig.11. It was conducted after finishing the project and before defending it in front of the teachers. As in the previous surveys, working in a real and useful project is the most appreciated characteristic of the course, claimed for about 40% of the students (see Fig. 11a). Furthermore, the topic and technologies involved in the project were claimed by 16% of students. Self-learning is the second one with 22% of votes, but there are more students (28%) who think that self-learning was really the worst of the course (see Fig. 11b). This was only slightly outnumbered by the idea of not being able to bring the equipment to home or not enough lab time. Another important group of students (22%) think that the planning was not good or some activities made them waste time, like surveys. Surprisingly, only a few students (9%) thought that it was too much work to cope with.

B. Teacher perception

From our point of view, students were very reluctant to read the documentation we provided with. In many cases, when they were solving an exercise, they simply put together several examples without adjusting them to their own problem. They abuse the use of "trial and error", and they don't want to use any methodology to debug their codes.

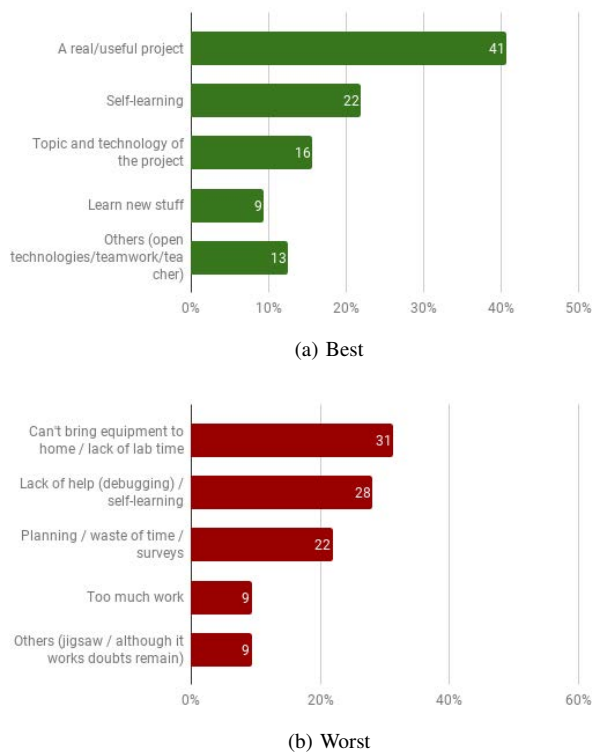


Fig. 11. Student perception after finishing the project

Furthermore, students had serious difficulties to understand the way Lua works, based on events. They keep trying to write functions sequentially, thinking wrongly that until one function is not completely finished, the next it is not going to be executed.

It is also very difficult to make them follow written instructions properly. We provided exhaustive and precise instruction for each document or code they had to hand in, including the format and the rubric for evaluation. However, they fail following simple rules, even when they know they are going to be evaluated based on these rules.

By the way, we have found some practical difficulties in evaluating Node-RED flows designed by the students. When you have 40 individual solutions proposed by students and presented as exported flows in JSON format, you have a tough work ahead to test them. You have to manually import every flow in your Node-RED programming tool, evaluate the design and then the functional aspects. We have found a very practical solution for next year: designing a Node-RED flow to automatically perform the functional testing of the student's designs. We can test the answer of the student design and store on a collection in the database the results for every student assignment, making easier the evaluation process.

C. Required Improvements

It is clear that the students really like the project and this, along with teamwork, encourages them to follow the course. However, some adjustment is required to improve their experience and the quality of the learning (the project). First, they

need specific material to learn how to debug systematically a program (or system). More progressive exercises for each technology have to be introduced previous to work directly in the project, especially for Lua. We should control better their individual progress in a week term basis. To do that we are planning to use an individual diary, where they will have to write down the tasks they work on and the hours used.

Another important issue we should address is the problem of bringing to home the prototype components. This year we did not allow them to do it mainly because, in our opinion, they had another work to do at home. They had to read the documentation we provided, to plan the hardware/software architecture, to design the functions, to generate the asked documentation, etc. In other words, they had to think before doing it. However, in their way of working (trial and error), those matters were a waste of time. There were also other reasons, such as preventing the situation of not being able to work in the lab because they forgot to bring the equipment back. Furthermore, if they really thought they need it at home, why did not buy it if it cost about 5 euros? We think that this complaint was more a self-excuse than a real problem. The easiest solution would be to let them bring the material to home but we do not think that it going really to help them. We better have to find a way to reinforce the task of planning and thinking ahead.

We think that utilizing open-hardware and open-software was a good decision and students appreciated it. However, taking into account that our students have serious problems with Lua, when using it was intended to help them in programming, we are considering to shift to a more traditional way of programming using C through Arduino environment. They are used to these tools and it may be easier for them.

Despite all difficulties, we think that PBL is the better way of confronting the challenge of promoting self-learning and boosting motivation. We will need several cycles to adjust the planning and small details of the methodology, but we think we are in the good way.

V. CONCLUSION

In the continuous search for improving didactic methodologies, we observed that the interest shown by Engineering students in working on real and practical projects is a great opportunity to engage them in our courses. In fact, we started some time ago introducing more and more fancy and practical laboratory exercises every year. Finally, we cross this point of not return into the PBL approach.

Setting up the new course design has been a challenging task. We have confronted the students' reluctance to face their learning process autonomously. This is a skill they need to acquire, but initially, they can feel uncomfortable with the new situation. Above all, we are satisfied with the initial results obtained.

We have observed a great positive impact on the student satisfaction and motivation. Tuning the courses towards developing real projects on the field, has a large impact on acceptance, learning objectives achievements and motivation

towards the course content. The satisfactory results and the positive response obtained from students leads us to keep on improving the PBL course design in the years to come.

We hope sharing this experience encourages other teachers to put this teaching style into practice and notice its advantages. We truly believe that some of the tools and platforms we have selected and presented in this paper can make a good service as resources for courses close to our subject area.

ACKNOWLEDGMENTS

The authors would like to thank Professors Miguel Angel Quesada Felice and Antonio Javier Matas Arroyo for their helpful advising on the Urban Garden topic. This work has been supported by project TIN2016-80920-R, from the Spanish Government and by "I Plan Propio Integral de Docencia" and Educational Innovation Project PIE17/085, from Málaga University.

REFERENCES

- [1] J. Larmer, J. R. Mergendoller, and S. Boss, *Setting the standard for project based learning: a proven approach to rigorous classroom instruction*, 2015.
- [2] (2018, Jan.) NodeMCU DEVKIT v1.0 development board. [Online]. Available: <https://github.com/nodemcu/nodemcu-devkit-v1.0>
- [3] (2018, Jan.) NodeMcu connect things easy. [Online]. Available: http://www.nodemcu.com/index_en.html
- [4] (2018, Jan.) NodeMCU documentation. [Online]. Available: <https://nodemcu.readthedocs.io/en/master/en>
- [5] (2018, Jan.) Integrated Development Environment (IDE) for ESP8266. [Online]. Available: <https://esp8266.ru/esplorer/>
- [6] (2018, Jan.) Raspberry Pi. [Online]. Available: <https://www.raspberrypi.org>
- [7] (2018, Jan.) Node-RED. [Online]. Available: <https://nodered.org/>
- [8] (2018, Jan.) Eclipse mosquitto. [Online]. Available: <https://mosquitto.org>
- [9] (2018, Jan.) MQTT protocol. [Online]. Available: <https://mqtt.org>
- [10] (2018, Jan.) JSON lightweight data-interchange format. [Online]. Available: <https://www.json.org/>
- [11] (2018, Jan.) Node-RED dashboard module. [Online]. Available: <https://github.com/node-red/node-red-dashboard>
- [12] (2018, Jan.) Welcome to the MongoDB Docs. [Online]. Available: <https://docs.mongodb.com/>
- [13] M. Valero and J. Domingo, *Técnicas de Aprendizaje Cooperativo y Aprendizaje Basado en Proyectos*, 2017.
- [14] R. M. Felder, "We never said it would be easy," *Chem. Engr. Education*, vol. 29, no. 1, pp. 32–33, 1995.