

# Building a CPS as an Educational Challenge

<http://dx.doi.org/10.3991/ijoe.v10i4.3765>

P. González-Nalda, I. Calvo, I. Etxeberria-Agiriano, A. García-Ruiz, S. Martínez-Lesta, D. Caballero-Martín  
University of the Basque Country (UPV/EHU), Vitoria-Gasteiz, Spain

**Abstract**—Cyber-Physical Systems (CPS) are highly multi-disciplinary systems that involve disciplines such as embedded computing, control theory and communication technology. Even though building this kind of systems is a challenging task, its introduction during the last courses of computer science and engineering university degrees has proven very valuable to teach students to deal with complexity. This paper presents an educational experience in the field of Computer Science at the University College of Engineering of Vitoria-Gasteiz (UPV/EHU). As part of an elective course, students had to design and develop an Android-based remote control system for teleoperating an educative mobile robot controlled by means of a Raspberry Pi. They had to apply and combine concepts learnt in previous disciplines (such as programming, operating systems or communication networks), reinforcing already acquired skills, as well as gaining other generic skills demanded in their future professional career, such as: (1) effectively working in groups, (2) integrating different technologies, (3) proactive problem solving, or (4) coping with system complexity.

**Index Terms**—CPS, PBL, Teleoperation, Mobile robotics, Education

## I. INTRODUCTION

The term Cyber-Physical Systems (CPS) was coined at the middle of last decade by scientists from different disciplines, mainly in the fields of real-time and embedded systems, networks communications and control systems (see Figure 1), to describe those *systems* that require connecting computers (*cyber*) with the *physical* world. Specifically, CPS allow integrating computation and physical processes [1, 2]. Somehow, CPS recover cybernetics concepts introduced in 1948 by Norbert Wiener [3] but adapted to the XXI century technologies, where the miniaturization of digital systems and current communication technologies allow its pervasiveness and ubiquity. Nowadays, these systems are gaining higher importance due to the requirements of the new complex control systems, especially in terms of accuracy and quality, which typically involve combining increasing amounts of information.

So far, such systems, traditionally known as embedded systems, have received little attention in the educational curricula, particularly in graduate studies, since they were considered very complex and specific. In fact, most often, the main concern in the design of such systems laid in optimizing the use of their resources (mainly CPU, memory and communication bandwidth) which tended to be scarce. However, this situation is changing rapidly. Current market trends, especially the continuing advances and cost reductions in the domains of information technology and computing, are making this kind of systems very common in modern applications. Nowadays, it is easier and cheaper to have flexible, powerful and low cost embedded platforms, such as Raspberry Pi or even Ardu-

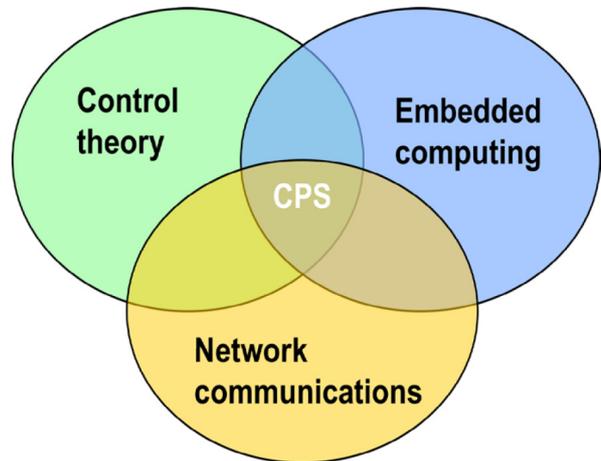


Figure 1. Main disciplines involved in CPS.

ino boards, which facilitate the construction and prototyping of such systems. Actually, CPSs are becoming a rising research topic, being financed by a growing number of projects and funding bodies [4, 5].

CPS are characterized by the strong interaction with physical processes which are dynamic by nature. In this type of systems time and concurrency are key issues that do not show up at all, or only partially, in general purpose systems. Actually, real-time requirements, i.e. those in which the correctness of a result includes the time taken to obtain it, are typical in CPS.

On the other hand, the implementation of the so-called active methodologies in education is turning classical teaching methods into learning experiences. Such methodologies are based on the application of a basic idea that comes from cognitive science: learning is the result of living repeated experiences in first person. By applying active methodologies students acquire skills (including theoretical and practical knowledge) through experiences (e.g. case studies, problems, or projects) which force them to take part actively, by seeking information and discovering appropriate mechanisms, in order to find a solution. A common active methodology in the field of engineering is the so-called Project Based Learning (PBL) [6]. Indeed, it is possible to find many examples of its application to teaching in the literature [8, 9, 10].

But changing the methodology is not enough for achieving a successful learning. It is also necessary, maybe essential, to propose activities that encourage and motivate students beyond the obligation to do some work in order to pass or obtain a good qualification mark. Therefore, it is fundamental to raise motivational problems that attract the attention of students. This is where the construction of CPSs can provide interesting and motivating challenges due to its novelty and multidisciplinary nature.

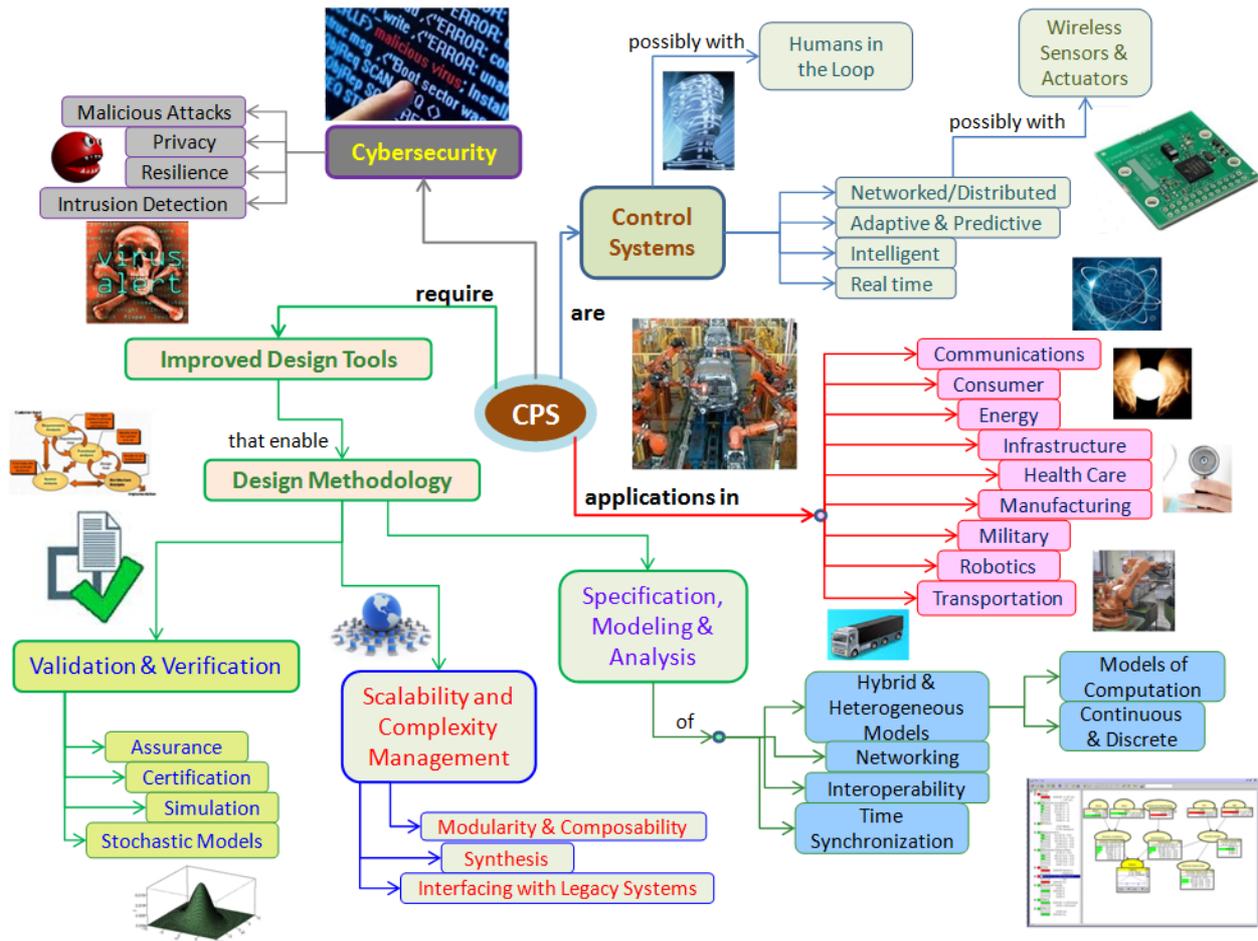


Figure 2. CPS conceptual map, adapted from [11].

The concept map shown in Figure 2, adapted from [11], provides an alternative definition to the CPS concept based on the fact that they are Intelligent, Real Time, Networked and Distributed Control Systems involving aspects of Security, Integrity and Resilience. As seen in the figure, due to its transversal nature the design and implementation of CPS is well suited from the educational point of view for students in the final year of Computer Science and Electronics related degrees. Actually, CPS require students to apply multiple skills acquired in previous disciplines related to computer architecture, operating systems, computer networks, computer security, programming (structured and object-oriented), project management, physics and electronic, among others. Unfortunately, too often students see these subjects as isolated domains without any interaction with each other, which does not correspond to the reality of their future professional career.

As a consequence of facing such a variety of techniques and systems, CPS designers must address several challenges [12]:

1. Efficient integration of heterogeneous network technologies
2. Tolerance of the complete system against faults on individual devices
3. Prioritization of critical actions in system degraded situations
4. Efficient energy management of the distributed system

## 5. Communications security

In order to satisfy simultaneously all the above requirements an additional abstraction layer (middleware) is typically utilized [13]. This layer, located above the application layer (see Figure 3), is responsible for enabling an efficient management of the interactions in CPS. Indeed, there is evidence that using middleware in complex distributed applications results in a reduction of up to 50% on the cost of software development [14].

This paper presents an educational experience developed by final year students of the Computer Engineering in Management and Information Systems Degree of the EUI Vitoria-Gasteiz, within an elective course on Systems Administration. More specifically, following the project-based learning methodology, students were suggested to build a CPS that allowed them to strengthen specific skills acquired in previous courses and other particular transversal competences of the degree. The result of the experience was the design and development of an Android-based remote control system for teleoperating an educational mobile robot.

The remainder of this paper is organized as follows. In Section II some educational experiences that have been conducted with CPS are analyzed. In section III the project and its justification are posed. Phases and problems encountered during the development of the work are described in Section IV. Finally, Section V draws some conclusions.

## II. EDUCATIONAL EXPERIENCES WITH CPS

In addition to introducing a technology that is already happening, designing and implementing Cyber-Physical Systems (CPS) can be a suitable method in higher education to acquire knowledge and reinforce skills. In this scenario, some universities, such as UC Berkeley, are delivering introductory courses to CPS design and development. In this particular case the text of the course is available online [7] and may provide an overview about the subject. As it can be seen in the papers presented at the first workshop on education in CPS [15], there is a very active community interested in the teaching of this kind of systems which focuses on their theoretical aspects as well as in the problems that arise during their implementation in the classroom.

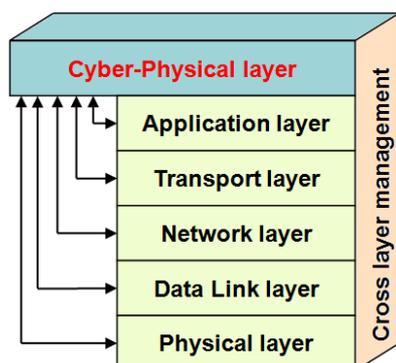


Figure 3. Cyber-Physical level and Middleware for CPS.

It is important to note that the acquisition of skills is done gradually during the development of the studies. In fact, although skills are acquired through different subjects taken along the grade, they should logically not pursue the same learning objectives. Thus, during the first degree courses, instructors try to design subjects that motivate students in order to acquire basic competence levels. For example, in [16] LEGO Mindstorms NXT robots are used to show students the importance of using flowcharts in the process of program design, and the consequences of not making them correctly. Conversely, subjects studied during the last years of the degree should include learning experiences involving other skills more related to the future professional careers of students such as: (1) working effectively in groups; (2) integrating different technologies; (3) solving problems proactively; or (4) coping with the complexity of the systems. In this scenario, projects based on the CPS concept can be very motivating and interesting for their freshness during the last years of the University degrees.

## III. PROJECT DESCRIPTION

This paper describes an experience carried out in *System Administration*, an elective final-year degree subject for Computer Science students. The instructors proposed a CPS project involving a distributed control application for teleoperating a mobile robot by means of an Android-based smartphone. Students had to evaluate different hardware platforms and technologies for remotely controlling a given educational robot, consisting of a three-wheeler, driven by two-wheels with an additional free wheel (see Fig. 6). More specifically, students analyzed in detail the possibilities offered by different hardware platforms, including Arduino and Raspberry Pi, and tested

their capacity to send appropriate control signals to the servo motors that drive the three-wheeler.

One of the requirements of the project was to provide a low-cost solution, since the available budget was limited. After the analysis, students finally chose a Raspberry Pi board for directly controlling the three-wheeler, and an Android phone as user interface. These devices were programmed in personal laptops under the Ubuntu GNU/Linux operating system.

The project involved controlling the mobile robot by means of the gyroscope available at the mobile phone. An application was developed which translated the left and right tilt movements of the smartphone into motion commands to be followed by the mobile robot. The application executed on the phone provided a small graphical user interface for controlling the speed of the mobile robot. The communication between these two platforms was established by means of TCP sockets over Wi-Fi technology.

### A. Platform Justification

Instructors considered that the proposed project was particularly suitable for the course as it required using varied hardware, which demonstrated the skills of operating systems management and development acquired in the course. At the Raspberry system the *Raspbian* platform was chosen, which is a Debian-based *GNU/Linux* distribution aimed at Raspberry Pi hardware. Raspbian provides great flexibility and allowed students to demonstrate their ability in applying the knowledge acquired in the theoretical part of the course. Android was used to provide the user interface. Note that both decisions were also responding to the student's desire of experimenting with these platforms, based on their novelty.

### B. Mobile Robotics

It has always been very appealing to work with mobile robots. This can be motivated for being the ultimate tool for humans, our own reflection or our Frankenstein, as explained by Gubern [17]. Undoubtedly, human beings have pursued the autonomous robot idea for a long time, robots that could facilitate our lives. Nowadays, such devices have already begun to enter our houses becoming increasingly popular, e.g. in the form of autonomous vacuum cleaners. However, these systems are fairly complex to build.

Although mobile robots have a long tradition in engineering, both the physical and control parts are frequently resolved separately. In the majority of cases top-down approaches are used, based on classical artificial intelligence techniques. In other cases, as it can be found in disciplines such as Artificial Life, a global bottom-up incremental development is sought, which can be easily applied to CPS. For example, in the field of *Evolutionary Robotics* (ER) this approach is radical [18]. Precisely in the ER there was a revolution with the consideration of the concepts *corporeity* and *location*, necessary for obtaining intelligence [19, 20, 21]. The system should have a place in the environment and must be able to interact with the environment through physical form.

## IV. DEVELOPMENT OF THE EXPERIENCE

The experience addressed in this work can be divided into the following four phases: (1) project conception; (2) system behavior; (3) problems encountered; and (4) results, which will be described through this section.

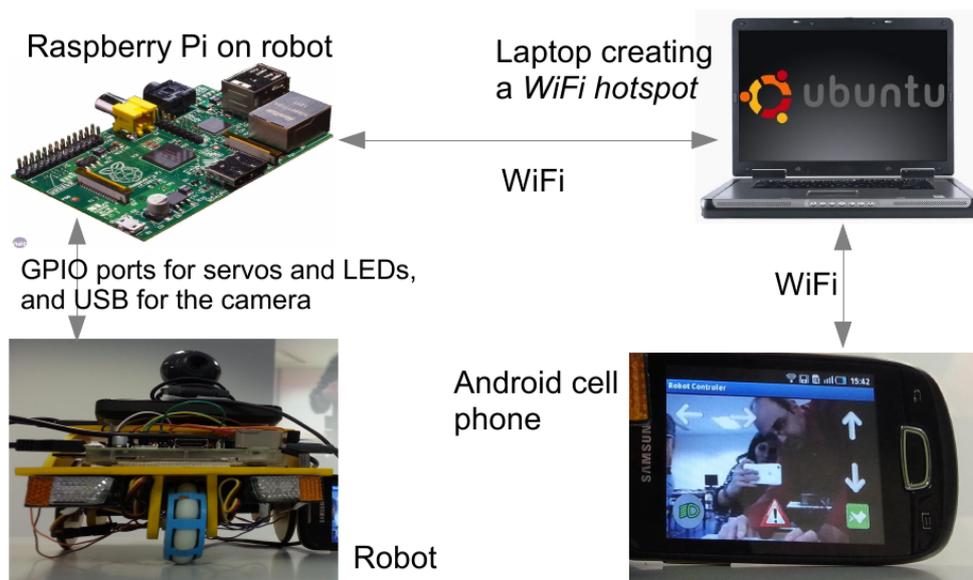


Figure 4. Initial scheme of device interaction. In the final system the Raspberry Pi creates the hotspot.

### A. Project Conception

The main problem was to design a system based on a client-server architecture connected by TCP sockets over Wi-Fi technology. This scheme allows distributing the work in the system in a natural way. Project work planning was divided into the following tasks (see Fig. 5):

- **Server Programming:** An embedded server listening to the movement commands issued by the client application encoded as strings. The server was implemented as a Python programmed service over the Raspbian OS at the Raspberry board.
- **Hardware Configuration:** Removing and changing the mobile robot proprietary control and electronics in order to connect directly the GPIO ports of the Raspberry Pi to the servos. At this stage the use of an Arduino board was considered for the interface between the Raspberry Pi and the robot, but this solution was finally discarded.
- **Client Programming:** A Java client application on top of Android capable of generating the appropriate interaction with the server via TCP sockets and string messages.
- **Communications:** Defining the messages sent among the client and the server (including images from the robot's camera, a standard USB webcam compatible with the V4L of the Linux kernel) and testing of a hotspot for WiFi communications.
- **Documentation:** Documentation of the whole project.

The development of the project followed an iterative and incremental methodology, that is, at first, basic functions were implemented and new functionalities were added as investigated. During this process requirements were modified along with the system design, which is a common approach in projects where agile methodologies and free software are used.

### B. System behavior

The client connects to the server via a socket and responds to the events generated by the Android GUI appli-

cation by means of sending text strings and receiving "OK" messages when commands are executed correctly. In case of lack of events, only background video images captured by the camera are sent and shown by means of the `android.widget.VideoView` class.

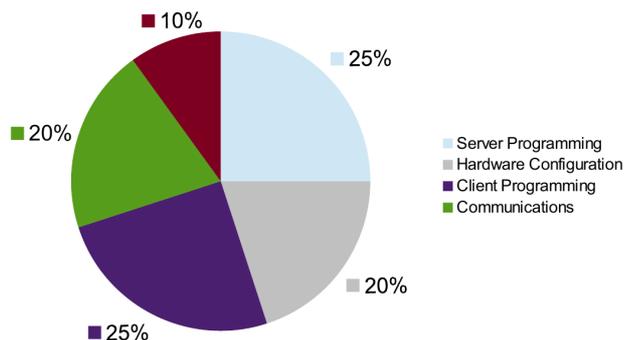


Figure 5. Relative weight of the different project tasks.

An application protocol between the client (executed on top of Android) and the server (executed at the Raspberry Pi) was defined, which included the messages shown in Table I. The first two messages were related to the management of the connection. Five additional control commands, with their corresponding parameters, were used to manage the vehicle. The server receives the string commands which are processed individually. It also reports "OK" in all successful operations as a means of basic error handling. The information flow is detailed in Fig. 6.

TABLE I.  
 COMMUNICATION MESSAGES OF THE IMPLEMENTED PROTOCOL

Connection	Control	Errors
"STARTCONNECTION"	"MOVE"	"OK"
"ENDCONNECTION"	"BRAKE"	
	"LIGHTS"	
	"DIRECTIONLIGHT"	
	"EMERGENCYLIGHTS"	

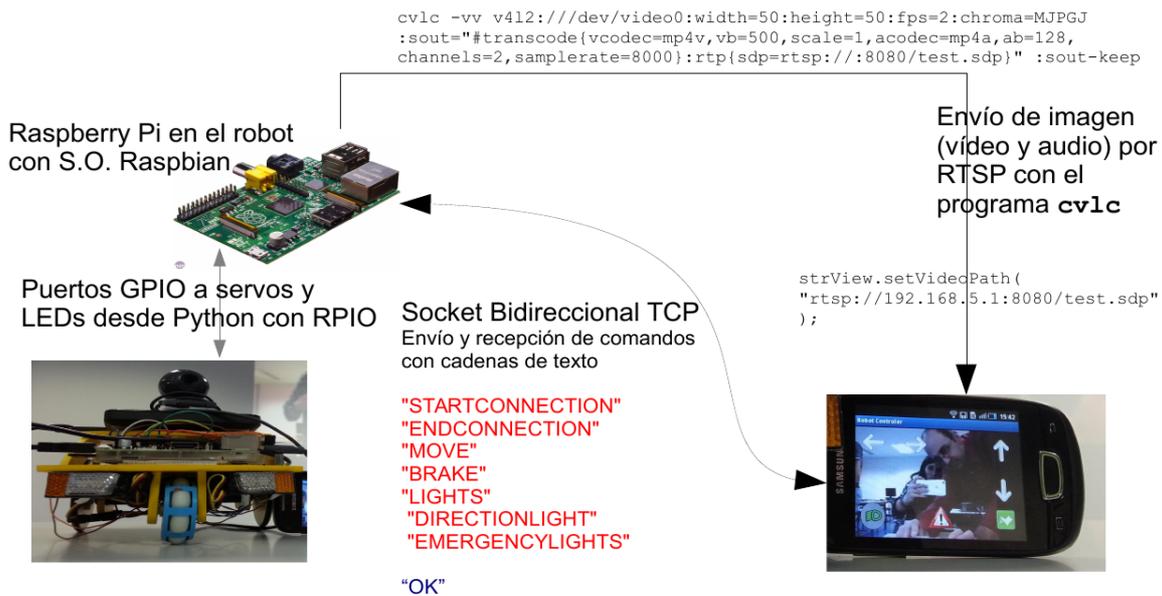


Figure 6. Illustration of the communication flow in among devices.

### C. Problems and difficulties

During the development of the project, students had to face up to the typical difficulties when unknown techniques are learnt and implemented on the fly, such as learning new programming languages or using novel platforms. They also had to review basic knowledge acquired in previous disciplines, e.g. how to use electronics to control the lights of the robot. In most cases, these difficulties were solved by the students just by searching solutions to the problems at the Internet.

One problem that took a significant amount of time to solve was finding a way to control the servos directly from the Raspberry. The design could be simplified using an Arduino board, which provides the capacity of using hardware pulse width modulation (PWM) signals. Nevertheless, for the sake of simplicity students decided to discard this option in order to reduce the number of different hardware platforms used. Instead, they used a Linux library for accessing Inputs and Outputs, including PWM signals, directly from the Raspberry Pi, the so-called RPIO. This approach required compiling, installing, and, of course, learning how to use this library. Also, students had to face the lack of information about the robot servos, since no appropriate documentation was available. Students had to use trial and error techniques to obtain the corresponding values: Between 0 and 150 pulses the movement speed is proportional backwards, and from 150 pulses on it is proportional forwards.

The decision of which device should “create” the Wi-Fi settings and its configuration also required some analysis, since the IP address of the server was defined in the client program and no service discovering mechanism was implemented. The use of the institutional *eduroam* Wi-Fi was discarded for the sake of simplicity, mainly because due to security issues traffic is limited to certain ports. At first, the Wi-Fi hotspot creator was intended to be the laptop, but later it was decided that the Raspberry itself could create it. This approach required compiling the device driver and working in the Raspbian system.

The video traffic generated at the mobile robot caused several problems. Firstly, it was necessary to compile and

install the driver of the USB camera as a Linux kernel module, so that the Raspbian could handle the image. In addition, the video format sent from the Raspberry to the Android application was very specific and involved sending dummy sound, because it was so required by the Java class responsible for displaying the Android background video: `android.widget.VideoView`. The video signal was generated by means of a real time streaming protocol (RTSP) executed with the VLC media player at command line Raspbian.

It also became necessary to take into account and observe the frequency at which the server had to read the sockets to avoid over-consuming battery and overheating the Raspberry. As the client is event-driven and the server blocked for readings, processor consumption was not significant.

Energy was mainly consumed by the servo motors, the video signal conversion and the Wi-Fi hotspot. The use of more suitable hardware could have decreased energy consumption, for example, replacing Wi-Fi communications for a more suitable ZigBee network could have saved energy at the robot. However, budget constraints rejected this option.

During the project implementation students incurred in some typical errors, which are frequent in the learning process of the technologies involved. For example, they tried to feed the Raspberry Pi from the USB port of a laptop while being programmed. However, this connector only provides 500 mA which are insufficient for both tasks. For this reason, the Raspberry Pi inexplicably used to crash or “hung”.

### D. Results

Overall, instructors consider that the learning experience has been very positive from the point of view of student learning: minimum objectives were achieved and new fresh features were included. The degree of satisfaction of students was also very high, since they were very motivated to learn new technologies. There is a demonstration video available at [22].

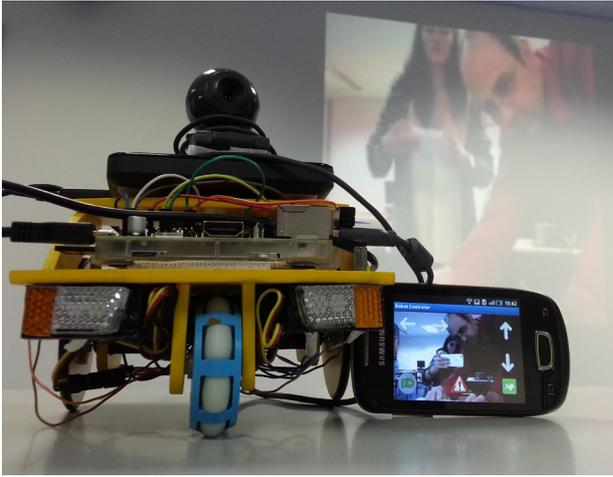


Figure 7. Final robot as presented. The Raspberry Pi, battery, webcam, Mobile Android with webcam image projector and a PC with the same video signal with a delay is observed.

## V. CONCLUSIONS

This paper describes a Project Based Learning experience involving the development of a Cyber-Physical System. The proposed project consisted of developing and Android-based remote control system for teleoperating an educative mobile robot. More specifically, the project involved: (1) Creating an Android-based client that provided the user interface to remotely control an educative mobile three-wheeler robot; and (2) Programming a Raspberry Pi-based server which controlled locally the operations of the mobile robot itself. A demonstration video is provided at [22].

The authors consider that proposing projects based on the CPS concept is very suitable for students at their final year at Computer Science and Engineering Degrees for several reasons:

1. this kind of projects combine the application of diverse skills and knowledge acquired in previous courses;
2. they allow developing transversal skills demanded in the future professional career of the students (such as, working effectively in groups, integrating different technologies, proactive problem solving, or coping with system complexity);
3. due to their novelty, students get more motivated, proactive, and inventive.

In addition, it is important to note that this type of projects allows instructors to have a context to refer the explanation of concepts related to the key skills of the subjects.

In order to further increase the motivation of the students, in the future it is envisaged to change the hardware platform to an remote controlled vehicle, already used in many situations, as in [23, 24].

## REFERENCES

[1] W. Wolf, "Cyber-physical Systems," *Computer*, vol. 42, no. 3, pp. 88–89, March 2009. <http://dx.doi.org/10.1109/MC.2009.81>

[2] R. Rajkumar, I. Lee, L. Sha and J. Stankovic, "Cyber-physical systems: The next computing revolution," *Design Automation Conference (DAC), 47th ACM/IEEE*, pp.731-736, June 2010.

[3] N. Wiener, *Cybernetics or Control and Communication in the Animal and the Machine*, 1948, Wiley and Sons, New York.

[4] W. G. Gilroy, "NSF funds Cyber-Physical Systems Project," <http://newsinfo.nd.edu/news/17248-nsf-funds-cyber-phsys/>

[5] National Science Foundation Program Solicitation, <http://www.nsf.gov/pubs/2008/nsf08611/nsf08611.pdf>.

[6] B. Barron, "Doing with understanding: Lessons from research on problem- and project-based learning," *Journal of the Learning Sciences*, 7(3&4), pp. 271-311, 1998.

[7] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems. A Cyber-Physical Systems Approach*, <http://LeeSeshia.org>, ISBN 978-0-557-70857-4, 2011

[8] I. Calvo, J.M. López-Guede and E. Zulueta, "Aplicando la metodología Project Based Learning en la docencia de Ingeniería Técnica en Informática de Gestión," *Revista de Formación e Innovación Educativa Universitaria*, Vol. 3 (4), pp. 166-181, 2010. (in Spanish)

[9] H.A. Hadim and S.K. Esche, "Enhancing the engineering curriculum through project-based learning," *Frontiers in Education, FIE 2002. 32nd Annual*, vol. 2, pp. 1-6, 2002.

[10] S. Boss and J. Krauss, "Reinventing project-based learning: Your field guide to real-world projects in the digital age," *International Society for Technology in Education*, 2007.

[11] Cyber-Physical Systems, <http://cyberphysicalsystems.org/> Last access April 2014

[12] I. Etxeberria-Agiriano, I. Calvo, A. Noguero and E. Zulueta, "Towards middleware-based cooperation topologies for the next generation of CPS," *International Journal of Online Engineering*, 8 (Special Issue 2), pp. 20-27, 2012.

[13] I. Calvo, I. Etxeberria-Agiriano and A. Noguero, "Distribution Middleware Technologies for Cyber Physical Systems," *Proc. of the Remote Engineering & Virtual Instrumentation (REV-2012)*, pp. 298-301, July 2012.

[14] P. Marwedel, "Embedded and cyber-physical systems in a nutshell," *DAC.COM Knowledge Center Article*, 2010.

[15] First Workshop on CPS Education, 2013, Philadelphia, PA. <http://cps-vo.org/group/edu/workshop/proceedings2013>

[16] A. Alvarez and M. Larrañaga, "Using LEGO Mindstorms to Engage Students on Algorithm Design", 2013 IEEE Frontiers in Education Conference, Oklahoma City, 2013, pp. 1346–1351.

[17] R. Gubern, *El simio informatizado*, Fundesco, Madrid, 1987 (in Spanish)

[18] P. González-Nalda, "Navegación mediante evolución de redes neuronales recurrentes y dinámicas," *Tesis Doctoral*, Universidad del País Vasco UPV/EHU, 2008. (in Spanish)

[19] R. Brooks, 1990. "Elephants don't play chess," *Robotics and Autonomous Systems*, 6, pp. 3–15. [http://dx.doi.org/10.1016/S0921-8890\(05\)80025-9](http://dx.doi.org/10.1016/S0921-8890(05)80025-9)

[20] R. Brooks, 1991. "Intelligence without reason," In J. Myopoulos and R. Reiter, eds., *Proc. 12th IJCAI*, 569–595. Morgan Kaufmann publishers Inc.: San Mateo, CA.

[21] R. Brooks, 1991. "Intelligence without representation," *Artificial Intelligence*, 47, 139–159

[22] "CPS-Challenge video demonstration," Available at April 2014, <http://ehutb.ehu.es/es/video/index/uuid/531dcbcb6aaa8.html> [http://dx.doi.org/10.1016/0004-3702\(91\)90053-M](http://dx.doi.org/10.1016/0004-3702(91)90053-M)

[23] N. Oros and J.L. Krichmar, "Smartphone Based Robotics: Powerful, Flexible and Inexpensive Robots for Hobbyists, Educators, Students and Researchers," *CECS Technical Report 13-16*. November 26, 2013. Center for Embedded Computer Systems, University of California, Irvine, 2013.

[24] N. Oros and J.L. Krichmar, "Android™ Based Robotics: Powerful, Flexible and Inexpensive Robots for Hobbyists, Educators, Students and Researchers," *Cognitive Anteatr Robotics Laboratory*. UCA, Irvine, 2013. Web. November 2013. <http://www.socsci.uci.edu/~jkrichma/ABR/index.html>

AUTHORS

**Pablo González-Nalda** is with the University College of Engineering of Vitoria-Gasteiz, Department of Computer Languages and Systems, University of the Basque Country (UPV/EHU), as Senior Lecturer, (e-mail: pablo.gonzalez@ehu.es).

**Isidro Calvo** is with the University College of Engineering of Vitoria-Gasteiz, Department of Systems Engineering and Automatic Control, University of the Basque Country (UPV/EHU), Spain, as Senior Lecturer (email: isidro.calvo@ehu.es).

**Ismael Etxeberria-Agiriano** is with the University College of Engineering of Vitoria-Gasteiz, Department of Computer Languages and Systems, University of the Basque Country (UPV/EHU), as Senior Lecturer, (e-mail: ismael.etxeberria@ehu.es).

**Alejandro García-Ruíz** is a final year student of “*Grado en Ingeniería Informática de Gestión y Sistemas*

de Información” at the University College of Engineering of Vitoria-Gasteiz, University of the Basque Country (UPV/EHU).

**Sergio Martínez-Lesta** is a final year student of “*Grado en Ingeniería Informática de Gestión y Sistemas de Información*” at the University College of Engineering of Vitoria-Gasteiz, University of the Basque Country (UPV/EHU).

**Daniel Caballero-Martín** is a final year student of “*Grado en Ingeniería Informática de Gestión y Sistemas de Información*” at the University College of Engineering of Vitoria-Gasteiz, University of the Basque Country (UPV/EHU).

This work was supported in part by the University of the Basque Country (UPV/EHU) through grant EHU13/42. Published as submitted by the author(s). Submitted, April, 14, 2014. Published as resubmitted by the authors on May, 16, 2014.