10th IFAC Symposium Advances in Control Education
The International Federation of Automatic Control
August 28-30, 2013. Sheffield, UK

A Complete Solution for Developing Remote Labs

A. Mejías Borrero*. M. A. Márquez Sánchez.**
J. M. Andújar Márquez*. M. R. Sánchez Herrera******

** Electronic Engineering Computing Systems and Automatic Department, Huelva University,*
Huelva, Spain, (e-mail: mjias@uhu.es).
***Pintor Pedro Gómez Secondary School, Huelva, Spain (e-mail:marcoa@iesppg.net)*
**** Electronic Engineering Computing Systems and Automatic Department, Huelva University,*
Huelva, Spain, (e-mail: andujar@diesia.uhu.es).
***** Electrical Engineering Department, Huelva University,*
Huelva, Spain, (e-mail: reyes.sanchez@die.uhu.es)

Abstract: The use of remote laboratories for education is increasing because it is a method that allows a better use of resources (laboratories operate 24 hours a day) and students interact with real equipment from different locations. However, we can say that there are as many particular solutions as remote labs can be found today. In this paper, we present a design framework for remote laboratories, covering the various aspects involved and completely based on free software. Within this approach, professors have all the tools required to convert a laboratory experiment in a remote experiment. Moreover, they can also design an user interface that includes augmented reality techniques to enrich the user experience.

Keywords: Remote labs, augmented reality, Easy Java Simulations, Online education.

1. INTRODUCTION

Lab classes are undoubtedly essential in engineering studies to make that the future engineer obtains a suitable training. Although these lab classes have been traditionally carried out in the lab classroom, the current evolution of communications and computing resources in general, allows the conducting of remote practices to be faced with increasing means. Among others, in areas such as automation, robotic or electronics, different approaches of remote laboratories are emerging, which allows the access to didactic or professional material of different kind. For example, in Indrusiak et al. (2007) the reader can get a general overview of remote laboratories for digital systems design. Andújar et al. (2011) presents an example of this kind of lab, which allows programming and interacting with a Field Programmable Gate Array (FPGA) based development board. Vargas et al. (2011) and Santana et al. (2013) show remote labs devoted to Automatic Control. In addition, in the last one a comparative of the most outstanding remote labs of this kind is included. An example of remote labs applied to the engineering measurement can be seen in Restivo et al. (2009). Marín et al. (2005), Jara et al. (2008), and Mejías and Andújar (2012) present different remote labs devoted to teaching robotics.

However, in these and many other approaches found in the Literature, the designed solutions to develop remote labs can be considered specific. Each of them solves in its own way the different aspects implicated in the remote lab development.

In a general way, the remote access to lab classes can be divided in three strongly interconnected systems (Fig. 1) and present in this kind of training activities:

1. *Learning Management System (LMS)*. Associated to the didactics of the lab (teaching-learning process). The Human Machine Interface (HMI) needed to interact with the experimental system can or can not be integrated into the LMS.
2. *Experimental System*. It is defined as the set of devices, located at the University lab classroom that allows the development of a specific kind of lab class. For example, a class lab where students must program a Programmable Logic Controller (PLC) to control lab equipment connected to the PLC. The experimental system would be constituted by the PLC, the lab equipment, the camera which shows the assembly, the lighting system, supply sources, Data Acquisition System (DAQ) and the needed actuators.
3. *Communications System*. Responsible of encapsulate and unify the access to all the experimental systems through a public network (Internet) from the HMI.

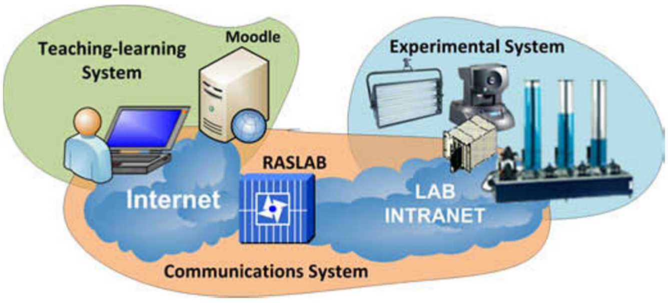


Fig. 1. General diagram of a remote lab.

In this paper, the authors propose the unification of the necessary elements to design a remote lab, based on the next five fundamental objectives:

1. Obtain a general remote access method. It is, independent of the experiment nature (communications, automation, physics, robotic, digital electronics, electrical machines, etc.).
2. Create a set of design tools that allows the professor to be helped in the task of getting the remote access to the lab class of his subjects.
3. Develop design tools that allow the HMI to be built. This HMI, from LMS, will allow to the student to access and interact with the experimental system in the remote lab class. This group also includes some tools of added value, which enrich the student's didactic experience, as for example the use of augmented reality (AR).
4. Enable the use of DAQs and low cost actuators in the experimental systems used in the remote lab class.
5. The complete solution must be based on free software.

All these objectives have been formulated with the intention of making that a professor with average knowledge of information technologies can easily convert lab classes traditionally carried out at the University lab classroom in remote lab classes.

The tools presented in this paper have been developed within the *Control and Robotics* research group, within the research line dedicated to the education on engineering. These tools, together with free software like *Moodle*, cover all aspects involved in conducting a remote class lab. The paper is organized as follow: In section 2, the scheme adopted to support the development of a remote lab is presented. In section 3, the tools incorporated to EJS to allow the access to hardware of data acquisition and actuators, as well as the AR system, are exposed. In section 4 the communications management system, named Remote Access System for Laboratories (RASLAB), is presented. It also incorporates control functions of supply, security and access. In section 5, two examples are briefly shown, implemented according to this structure and using the presented tools. Finally, conclusions and future working lines are presented.

2. INTEGRAL SOLUTION TO DEVELOP A REMOTE LAB

To meet the objectives exposed above, the three systems listed in section 1 are divided into the next subsystems:

- *User Interface*. Responsible for providing to the student a way of interaction with the remote experimental system, so that it will allow the lab class proposed by the professor to be carried out. This interface incorporates video at real time that shows the experimental system. In addition, if necessary, it can also incorporates sound.
- *Reservation manager*. Through this subsystem, the user makes a time reservation to use a specific experimental system and, so, to be able to carry out the lab class proposed by the professor.
- *Access control and communications*. This subsystem allows students to access, from any computer connected to Internet, to the experimental system for carry out the lab class. Since it is desirable for the user to make the lab class even from home, it is imperative that this subsystem involves security solutions suitable to the communication links that must be generated.
- *Management of information associated with a lab class*. This subsystem is responsible for making all the information associated with a specific lab class (manuals, tutorials, user guides, practice statement, etc.) easily available to the student.
- *Power Management of the Experimental System.* This subsystem controls the power supplied to the different systems of the remote lab. It turns on or off the different systems depending on whether they are in use or not. From this point of view, the subsystem is also a power manager of the remote laboratory.
- *Access to DAQs and actuators.* To handle an experimental system, the student should be able to change its parameters, read sensor values, act on elements such as motors or valves, etc. This subsystem is responsible for enabling the HMI to manage these hardware elements.

In those lab classes whose main objective is the programming and/or configuring a specific device, such as an industrial PLC or a FPGA based development board, the use of Integrated Design Environments (IDE) is usual. The design of these tools is not necessary, because manufactures provide it to the users of their products. Although these tools are independent of the other subsystems, they use the remote lab communications services, to allow students to remotely program a specific device located at the University lab classroom.

The developed solution covers all points raised, providing to the professor the tools that allow a remote lab to be put in operation, always using free software. As far as possible, available tools are used, but the remainder have been designed specifically to fit into the general structure. Specifically, Easy Java Simulations (EJS), Esquembre (2011), is used as the main tool to design the HMI. In addition, this known simulation tool is completed with new elements which make it able to access directly to specific hardware of data acquisition and actuation. Furthermore, to enhance the capabilities and capacities of the remote lab class, an AR system has been integrated in EJS. It increases the interactivity possibilities of the student with the remote experimental system.

Regarding the communications system, RASLAB, has been fully in order to handle the large variety of experiments possible. Furthermore, in order that EJS can use RASLAB, has been integrated a client on it. RASLAB besides creating the needed communications links, assumes the control of the experimental system supply and enables the suitable access to resources existing in the remote lab, using the data provided by the reservation manager.

3. NEW ELEMENTS FOR EASY JAVA SIMULATIONS: REMOTE ACCESS TO HADWARE AND AUGMENTED REALITY SYSTEM

EJS is a known tool which allows very different simulations to be created. It also has the option to generate a view of the simulation conducted by a hierarchical structure based on the plug and play technical with great capability of interactivity, which gives it strong potential in educational settings. In this work, EJS is used to generate the HMI that the student uses to remotely access to the laboratory. This interface is completely integrated into the LMS, because EJS introduces new features for embedding its applets into a Moodle course from version 4.3.7. Reader is referred to Esquembre (2013), where this feature is detailed. Additional software elements have been developed which allow EJS directly accesses DAQ boards and actuators from the generated HMI.

EJS has been fitted with an AR system based on ARToolkit software library, Kato (2004), to enhance the possibilities and capabilities of the remote laboratory. In addition, in order to create a sense of presence in the lab classroom it provides real-time image received from the laboratory of interactive 3D objects, registered in the three-dimensional space formed by the field of view of the camera. The AR system is easy to set up, without requiring a comprehensive knowledge of AR techniques. EJS supports simultaneously multiple AR systems with different cameras.

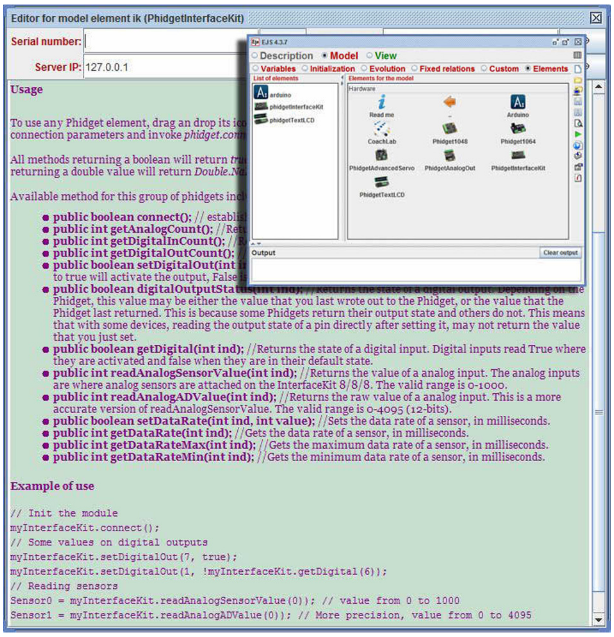
3.1 Direct access to hardware from EJS

New software elements have been developed to allow access to DAQs and actuators from EJS. It allows remote reading of sensors and the generation of control signals from the HMI. These software elements are oriented to access two device families with low cost and wide acceptance: products for USB sensing and control from Phidgets^{TM} Inc. and Arduino^{TM} boards. Readers are referred to Phidgets (2013) and Arduino (2013) web sites.

1.1 3.1.1 Accessing Phidgets Devices

Figure 2 shows in the foreground the EJS *Elements* panel (*hardware* folder), where the user can choose the specific device to access (locally or remotely). To do this, it is only needed to drag the selected device to the *List of Elements* panel.

From that moment, the user can access all available methods for using that device. Fig. 2 shows in the background the available help for a Phidgets Interface Kit device (a DAQ board with digital output capabilities), where some of the methods available can be seen as well as a simple example of how to use it in EJS.



Serial number:

Server IP: 127.0.0.1

Usage

To use any Phidget element, drag an drop its ic
connection parameters and invoke *phidget.con*

All methods returning a boolean will return *tru*
returning a double value will return *Double.Na*

Available method for this group of phidgets inc

- **public boolean connect();** // establi
- **public int getAnalogCount();** //Ret
- **public int getDigitalInCount();** //R
- **public int getDigitalOutCount();** //
- **public boolean setDigitalOut(int i**
  to true will activate the output, False i
- **public boolean digitalOutputStatus(int ind);** //Returns the state of a digital output. Depending on the Phidget, this value may be either the value that you last wrote out to the Phidget, or the value that the Phidget last returned. This is because some Phidgets return their output state and others do not. This means that with some devices, reading the output state of a pin directly after setting it, may not return the value that you just set.
- **public boolean getDigital(int ind);** //Returns the state of a digital input. Digital inputs read True where they are activated and false when they are in their default state.
- **public int readAnalogSensorValue(int ind);** //Returns the value of a analog input. The analog inputs are where analog sensors are attached on the InterfaceKit 8/8/8. The valid range is 0-1000.
- **public int readAnalogADValue(int ind);** //Returns the raw value of a analog input. This is a more accurate version of readAnalogSensorValue. The valid range is 0-4095 (12-bits).
- **public boolean setDataRate(int ind, int value);** //Sets the data rate of a sensor, in milliseconds.
- **public int getDataRate(int ind);** //Gets the data rate of a sensor, in milliseconds.
- **public int getDataRateMax(int ind);** //Gets the maximum data rate of a sensor, in milliseconds.
- **public int getDataRateMin(int ind);** //Gets the minimum data rate of a sensor, in milliseconds.

Example of use

```
// Init the module
myInterfaceKit.connect();
// Some values on digital outputs
myInterfaceKit.setDigitalOut(7, true);
myInterfaceKit.setDigitalOut(1, !myInterfaceKit.getDigital(6));
// Reading sensors
Sensor0 = myInterfaceKit.readAnalogSensorValue(0)); // value from 0 to 1000
Sensor1 = myInterfaceKit.readAnalogADValue(0)); // More precision, value from 0 to 4095
```

Fig. 2. Using *Elements* in EJS to access hardware devices.

There are already numerous supported devices. They comprise I/O boards (analog and digital inputs and outputs), relays, controllers for several kinds of engines (DC engines, servos...), sensors (GPS, accelerometers, gyroscope, compass, thermocouples, proximity, etc.) and single boards computers (running GNU/Linux operating system), with an integrated DAQ system, USB hub (to connect USB cameras, sensors, etc.), optional WIFI connection, all in a very compact board around 98 by 80 mm.

3.1.2 Accessing Arduino Boards

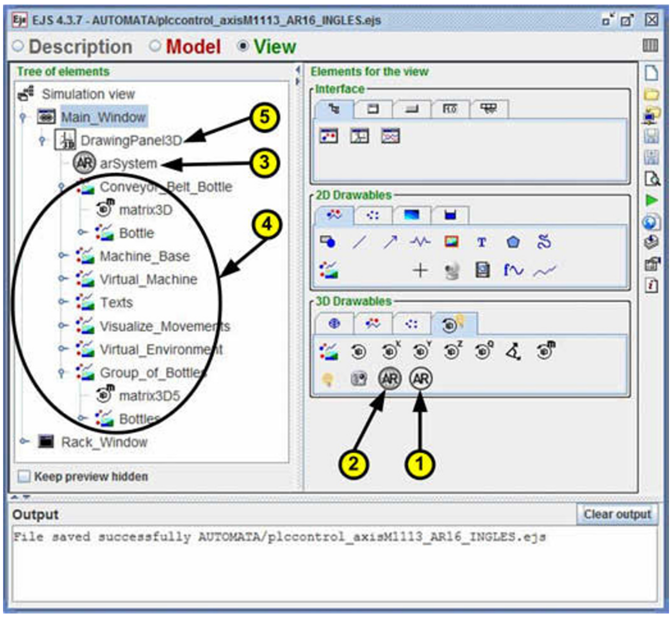
Arduino is an open-source electronics prototyping platform which has recently experienced a great success in educational settings. Even different subjects in different Universities incorporate it to their lab class. We have added an item to EJS that allows the access to the various existing Arduino boards (Uno, Mega, Ethernet, Due, etc.). Operations that can be performed with these boards include:

- Reading digital inputs.
- Writing on digital outputs.
- Reading analog inputs.
- Using PWM control.
- Using servomotors.
- Accessing to intelligent sensors with I2C interface (experimental).

The use of this element is similar to the Phidgets elements discussed above. To access an Arduino board from EJS, it is only needed to record the *StandardFirmata* sketch, which is available on the Arduino software. This sketch implements the firmata protocol on Arduino hardware. Readers can see Firmata (2013) for more information about this protocol. There is a sketch called *StandardFirmataEthernet* that allows remote access to Arduino boards with Ethernet connection capability, such as *Arduino Ethernet* board or *Arduino Uno* board with an *Ethernet Shield*).

3.2 Augmented Reality System in EJS

The video image received from the laboratory is an essential element in the HMI of a remote class lab. The possibility of enhance this image with interactive elements (acting with real elements located in the lab classroom) provides to the students attractive and amazing possibilities by the interface on his computer.



```
File saved successfully AUTOMATA/plccontrol_axisM1113_AR16_INGLES.ejs
```

Fig. 3. View Panel in EJS showing a design with an AR system.

The AR system developed and integrated in EJS allows ARToolkit markers to be used for the tracking of integrated 3D objects into the received video stream. Using the interactive capabilities of the graphic elements which may be added to the *View* panel of EJS and the use of 6DOF tracking using fiducial markers, the whole system behaves as a system with full AR capabilities, fulfilling all features listed in Azuma (1997). Several AR systems can be created in the same interface operating simultaneously, each of them using multiple markers.

The location of the icons which may be dragged to include an AR system in the *View* panel can be seen in Fig. 3. In this approach, there are two kinds of AR systems in EJS, according to the camera (local or IP camera) to be used as video source: local (1 in Fig. 3) or remote (2 in Fig. 3). Their use and configuration is completely similar.

Fig. 3 also shows the appearance of the EJS View panel with an example of a HMI that uses an AR system. It can be seen how the added AR system (3) is integrated into the structure of the view. 3D objects and groups (4 in Fig. 3) that can incorporate the AR system are already available in the *Elements for the view* panel in EJS.

3.2.1 Designing an AR system in EJS

The definition of an AR system in EJS is an easy 3 steps task:

1. Each marker to be used must have two associated variables:
   - A translation matrix (1 x 3) which receives the marker position with respect to the camera.
   - A 3D matrix (4 x 4) which receives the marker 3D data (scaling, rotation, etc.).
2. Add a *DrawingPanel3D* (5 in Fig. 3) to the *Tree of Elements* in the *View* panel. It will be the container of the AR system to be created.
3. Add and configure an AR system. The configuration includes:
   a. List markers to be used as well their real size.
   b. Sets the camera. It visualizes the real time video stream from the remote lab. 3D virtual objects are superimposed on the video, using tracking data from the AR system.

4. REMOTE ACCESS SYSTEM FOR LABORATORIES (RASLAB)

RASLAB is the responsible for centralizing and unifying the access to laboratory experiences, encapsulating all the communications, allowing the collaborative access, and also managing the power supply to the components of the experimental system. RASLAB has client-server architecture, and an EJS element has been developed to allow the RASLAB client to be included into the HMI. RASLAB Server runs as a service on a computer with two network interfaces running GNU/Linux operating system. This server acts as a borderline between the Internet (public IP addresses) and the lab intranet (private IP addresses) as can be seen in Fig. 4.

As outlined in Fig. 4, the access to a remote lab class within Moodle through a HMI with a RASLAB element built into it, will cause the activation of communications in a transparent way between the user interface and the corresponding experimental system. The access to the lab class is validated from the reservation manager in Moodle.

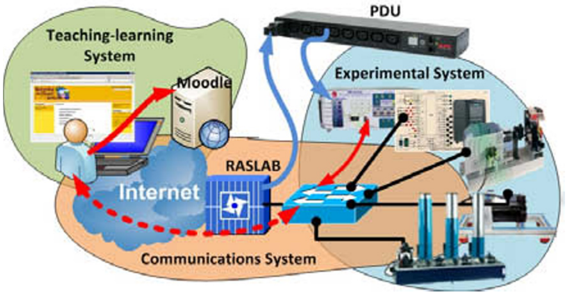
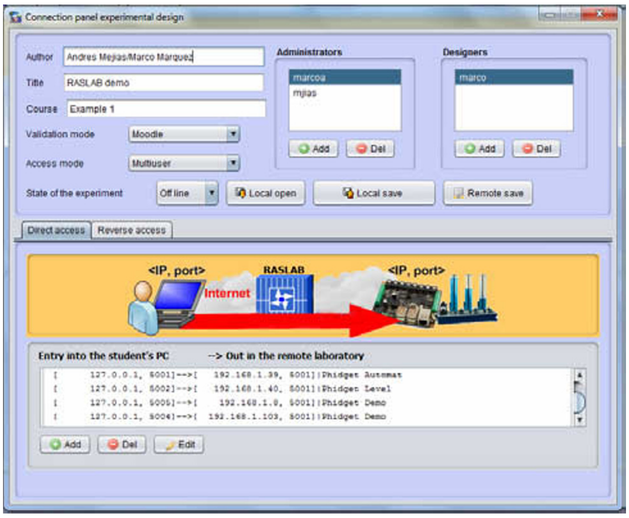


Fig. 4. The RASLAB system in a remote lab structure.

RASLAB also allows the association of lab class with the supply of the corresponding experimental system involved, by controlling one or several Power Distribution Units (PDU). The necessary settings of RASLAB are stored in XML files. To simplify an experience design, RASLAB offers a graphical tool that assists professors in designing remote lab classes (Fig. 5).



Author: Andres Mejias/Marco Marquez
Title: RASLAB demo
Course: Example 1
Validation mode: Moodle
Access mode: Multiuser
State of the experiment: Off line
Administrators: marcoa, mjias
Designers: marco

Entry into the student's PC --> Out in the remote laboratory

```
[    127.0.0.1, 6001]-->[  192.168.1.39, 6001]|Phidget Automat
[    127.0.0.1, 6002]-->[  192.168.1.40, 6001]|Phidget Level
[    127.0.0.1, 6003]-->[   192.168.1.8, 6001]|Phidget Demo
[    127.0.0.1, 6004]-->[ 192.168.1.103, 6001]|Phidget Demo
```

Fig. 5. RASLAB configuration graphic tool.

In this figure, one of the configuration panels is shown: the responsible of defining the necessary links between the user interface and the experimental system. Other panels manage the supply and the administration of user profiles. RASLAB can makes the access validation from a local database or it can delegates this function to an alternative way of validation through the network.

5. TWO EXAMPLES OF REMOTE LABS BY THE DEVELOPED SOLUTION

The developed solution for remote labs showed in this paper is being used in the Higher Technical School of Engineering at the University of Huelva (SW Spain) to develop remote labs in different subjects into the Computing Engineering, Energetic Engineering and Industrial Electronics Engineering degrees. As an example, Fig. 6 shows the HMI developed with the tools presented in this work: a classical control system, where the student must adjust a PID controller to fix the position of a servomotor. The HMI allows several configurations of the controller to be tested, displaying the behavior of the position for the different input conditions. It also provides the possibility of exporting data for their further processing by means of tools such as Matlab/Simulink, SciLab, etc. An AR system has been added in this HMI, stating to the student the position that the servomotor must achieve (a yellow circle) and the value of this objective (angle).

Fig. 7 shows the interface of an industrial automation experiment. The student programs a PLC to automate a real laboratory plant, which is "augmented" by 3D virtual objects that interact with the real ones. This figure may be compared with Fig. 8, which shows the original video stream received from the camera located in the University lab classroom.

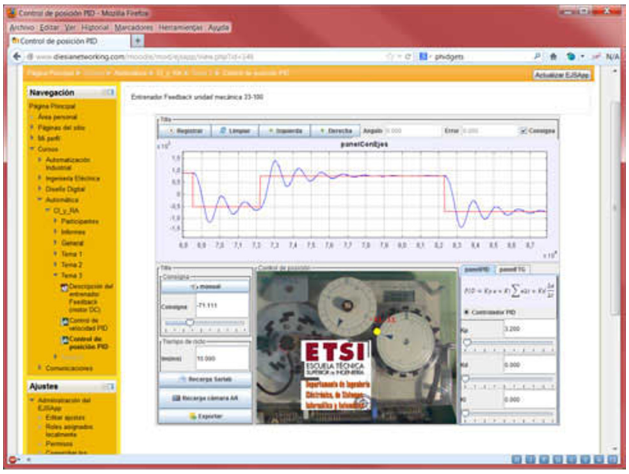


Fig. 6. A remote experiment using a servo fundamentals trainer from Feedback™ Instruments, Feedback (2013).

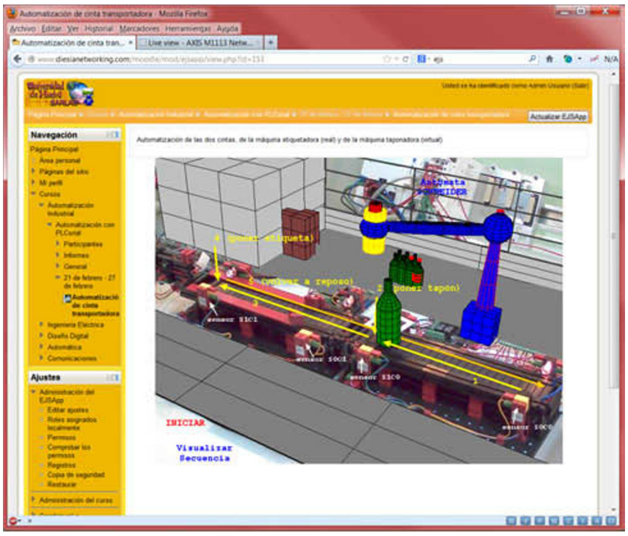


Fig. 7. The HMI (inside Moodle platform) of a remote experiment with a PLC and an industry training model from Fishertechnik™, Fishertechnik (2013).

In this figure, the ARToolkit markers used to properly superimpose the 3D virtual elements can be seen. Via his computer interface, the student can operate the remote PLC, so that he can observe the evolution of the real plant according to the schedule that he has sent to the PLC. The AR system converts the conveyor belt in an installation which must place a plug and a label into the bottle which is moved by the belt. In this case, real and virtual elements interact with each other, sharing the same three-dimensional space. The PLC programming is performed remotely using the design tool (Figure 9) provided by the PLC manufacturer. The PLC used is a Modicon M340 from Schneider Electric™, Schneider Electric (2013). This software uses the RASLAB facilities to access the remote PLC.

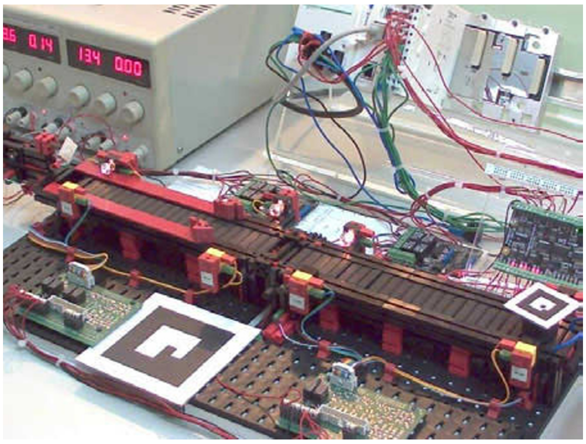


Fig.8. Appearance in the laboratory classroom of the remote experimental system corresponding to Fig. 7.

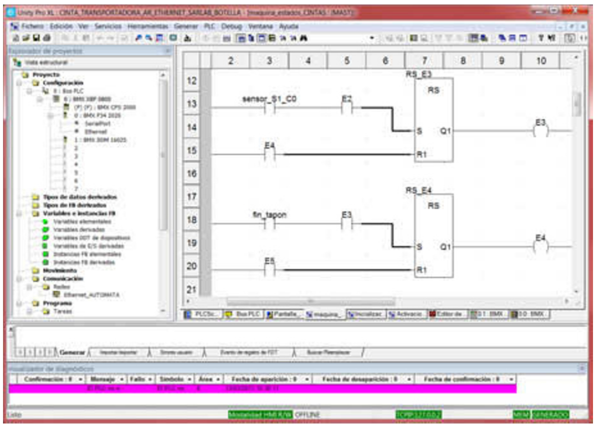


Fig. 9. Design tool (Unity Pro XL) used by the student to program the PLC.

5. CONCLUSIONS

In this paper, a complete solution for developing remote labs has been presented. The main features of the developed solution are: (1) provides a general remote access method which is independent of the experiment nature, (2) provides a set of tools that facilitates to the professor the remote lab class design, (3) Develops and integrates the HMI in the LMS, (4) enable the use of DAQs and low cost actuators in the experimental systems used in the remote lab class, and (5) the complete solution is implemented with free software. The paper develops the proposed solution and presents two practical examples. The developed system is currently in use in the Higher Technical School of Engineering at the University of Huelva (SW Spain).

REFERENCES

Andújar, J.M., Mejías, A., and Márquez, M.A. (2011). Augmented Reality for the Improvement of Remote Laboratories: An Augmented Remote Laboratory. *IEEE Transaction on Education*, 54(3), 492-500.

Arduino (2013). Open Source Electronics Prototyping Platform. [Online] Available at: <http://www.arduino.cc/> [Accessed 3 february 2013].

Azuma, R. (1997). A survey of augmented reality. *PRESENCE: Teleoperators Virtual Environments* 6(4), 355–385.

Esquembre, F. (2011). Easy Java Simulations. [Online] Available at: <http://fem.um.es/Ejs/> [Accessed 11 february 2013].

Esquembre, F. (2013). Using EJS with Moodle. [Online] Available at: < http://www.um.es/fem/EjsWiki/Main/UsingEJSWithMoodle> [Accessed 26 february 2013].

Feedback (2013). Educational Equipment. [Online] Available at: <http://www.feedback-instruments.com/products/education/control_instrumentation/servo_fundamentals_trainer> [Accessed 2 february 2013].

Firmata (2013). Generic protocol for communicating with microcontrollers. [Online] Available at: < http://www.firmata.org/wiki/Main_Page> [Accessed 6 february 2013].

Fishertechnik (2013). Industry Training Models. [Online] Available at: < http://www.fischertechnik.de/en/Home/products/industry.aspx> [Accessed 30 january 2013].

Indrusiak, L.S., Glesner, M., and Reis, R. (2007). On the Evolution of Remote Laboratories for Prototyping Digital Electronic Systems. *IEEE Transaction on Industrial Electronics*, 54(6), 3069-3077.

Jara C.A., Candelas A., and Torres F. (2008). An advanced interactive interface for Robotics e-learning. *International Journal of Online Engineering* 4(4):17–25.

Kato, H. (2004). ARToolkit. [Online] Available at: < http://www.hitl.washington.edu/artoolkit/> [Accessed 20 february 2013].

Marín R., Sanz P.J., and Del Pobil A.P. (2005). The UJI online robot: an educational and training experience. *Autonomous Robots* 15(3):283–297.

Mejias, A., and Andújar, J.M. (2012). A Pilot Study of the Effectiveness of Augmented Reality to Enhance the Use of Remote Labs in Electrical Engineering Education. *Journal of Science Education and Technology*, 21(5), 540-557.

Phidgets. (2013). Products for USB Sensing and Control. [Online] Available at: < http://www.phidgets.com> [Accessed 2 january 2013].

Restivo, M.T., Mendes, J., Lopes, A.M., Silva, C.M., and Chouzal, F. (2009). A Remote Laboratory in Engineering Measurement. *IEEE Transactions on Industrial Electronics*, 56(12), 4836-4843.

Santana, I., Ferre, M., Izaguirre, E., Aracil, R., and Hernández, L. (2013). Remote Laboratories for Education and Research Purposes in Automatic Control Systems. *IEEE Transactions on Industrial Informatics*, 9(1), 547 – 556.

Shneider Electric (2013). Automation Controllers. [Online] Available at: < http://www.schneider-electric.co.uk/sites/uk/en/products-services/automation-control/products-offer/automation-controllers/automation-controllers.page> [Accessed 24 february 2013].

Vargas, H., Sánchez, J., Jara, C.A., Candelas, F.A., Torres, F., and Dormido, S. (2011). A Network of Automatic Control Web-Based Laboratories. *IEEE Transactions on Learning Technologies*, 4(3), 197-208.