

Moodle-Enabled Virtual Lab for Fieldbus Process Control

Vítor Viegas¹, José Miguel Costa Dias Pereira²

¹CINAV - Portuguese Naval Academy, Base Naval de Lisboa, Alfeite, 2810-001 Almada, Portugal (vviegas2@gmail.com) ORCID [0000-0002-8929-1574](https://orcid.org/0000-0002-8929-1574); ²Setúbal School of Technology, Polytechnic Institute of Setúbal, Campus do IPS - Estefanilha, 2910-761 Setúbal, Portugal (dias.pereira@estsetubal.ips.pt) ORCID [0000-0002-0183-132X](https://orcid.org/0000-0002-0183-132X)

Abstract

The paper presents a Moodle-enabled remote laboratory intended to teach the principles of process control using the Foundation Fieldbus technology. The student logs into Moodle and accesses a “Virtual PC” learning activity that opens a private remote desktop with all set to run an FF simulator. Behind the scenes, a virtualization infrastructure, hosted on the Azure cloud, handles storage, cloning and deployment of virtual machines. The paper explains how this infrastructure was set up and how it works internally. Positive results were achieved with a limited number of students, on an elective course, making us believe that this infrastructure can be extended to highly populated courses.

Author Keywords. Remote Lab, Virtual Machine, Virtual Desktop, Azure, Moodle, Process Control, Foundation Fieldbus.

Type: Research Article

 Open Access  Peer Reviewed  CC BY

1. Introduction

The paper presents a remote laboratory, fully integrated in Moodle, intended to teach the principles of process control using the Foundation Fieldbus (FF) technology (Glanzer 2003). The laboratory makes use of a software simulator that emulates the application layer of FF by means of a function library built in LabVIEW. The simulator, named Quimera (Viegas et al. 2016), has been used at the Polytechnic of Setúbal since 2016 to teach courses in the areas of process control and industrial automation. In the past few months, we have tried to make the simulator available remotely and this paper describes part of that work.

The paper focus on the integration of the FF simulator into the Moodle infrastructure to provide a fully integrated experience of blended learning. The integration was possible thanks to Moodle’s Virtual PC activity, which provides access to virtual machines powered by virtualization platforms (e.g. Microsoft Azure). Students can start a virtual machine remotely, open LabVIEW and use the simulator. All this is made under the control of Moodle, which provides context, scheduling, and assessment, among other facilities useful to the learning process.

The paper is organized as follows: section 2 presents the FF simulator to be used by students; section 3 explains how the simulator was virtualized and made remote; and section 4 presents results and extracts conclusions.

2. FF Simulator

The Quimera simulator (Viegas et al. 2016) exploits the resources provided by LabVIEW – in particular its dataflow programming paradigm – to design, execute, and debug FF control applications. FF function blocks are implemented as virtual instruments, data links are represented by virtual wires, and bus monitoring tools are replaced by application debugging

tools. The simulator can be connected to virtual (simulated) processes through shared variables, as well as to physical (real) processes through data acquisition (DAQ) channels. The goal is to provide an easy-to-use, low-cost training tool that mimics faithfully the application layer of FF.

The Quimera simulator is composed by a set of virtual instruments located in the Quimera palette (see Figure 1), divided in the categories input, output and control & calculation. Each virtual instrument implements the logic of a given FF function block, and several virtual instruments can be combined to build a FF control strategy like the one shown in Figure 2. The simulation is an ordinary LabVIEW application that works as follows:

- The AI block (AI.vi) reads the process variable from a shared variable or DAQ channel, filters it, and scales it to engineering units.
- The PID block (PID.vi) implements the proportional-integral-derivative control algorithm that makes the process variable to converge to the setpoint (SP).
- The AO block (AO.vi) writes the manipulated variable to a shared variable or DAQ channel. This block feeds back its state to the controller to track a “fault state” condition.
- The control loop runs continuously with a sample period of 500 ms.
- The values of the setpoint, process variable and manipulated variable are all plotted on the same chart.
- Each virtual instrument can be configured individually by double clicking on its icon and editing the parameters available on its front panel.

The simulation shown in Figure 2 benefits from all the resources provided LabVIEW, including an easy-to-use visual programming editor, integrated help resources, powerful debugging tools, and rich palettes of functions and controls. This greatly facilitates the simulation process, helping the user to focus on the FF aspects of the application.

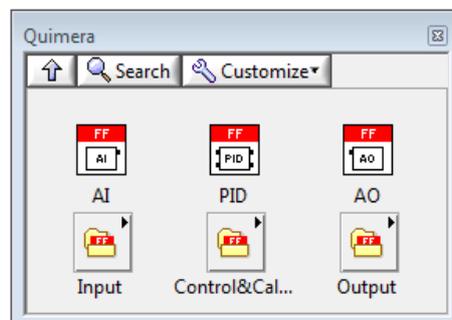


Figure 1: Quimera palette

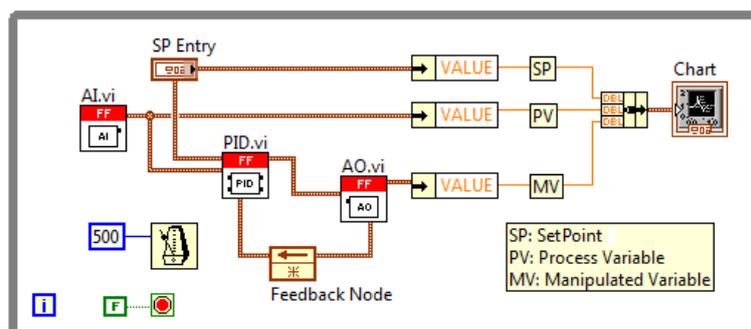


Figure 2: Simulation of a simple control loop

3. Making Things Remote

For a student to use the Quimera simulator, all (s)he needs is a computer with LabVIEW. Therefore, to make things available remotely, we need to provide a private remote desktop with all set to run the Quimera simulator. For that purpose, we built the virtualization infrastructure depicted in [Figure 3](#), which is hosted on the Microsoft's Azure cloud and includes five main actors:

1. Remote client: Is the machine from where the student connects to the Moodle server to make use of learning activities. The remote access is made from a common web browser using the HTTP or HTTPS protocols.
2. Moodle server: Is the web application that students see as the "Moodle". We are using Moodle 3.9 supported by an Apache web server running on a Windows 10 virtual machine. The Moodle server receives HTTP(S) requests from remote students, processes them, and sends back web pages as response. If a remote student asks for a remote desktop to run the Quimera simulator, the Moodle server passes the request to the virtualization broker, which, in turn, instructs the Azure service provider to clone and deploy a prestored virtual machine. The Moodle server must be enriched with the Virtual PC plugin ([Moodle, n.d.](#)) to be able to connect and communicate with the virtualization broker.
3. Virtualization broker: Is a dedicated virtual machine that acts as a mediator between third-party applications and the Azure service provider. It provides a programmable application interface through which applications can automate calls to the Azure service provider. In other words, many of the operations that we do manually on the Azure portal can be made programmatically by the virtualization broker. In the present case, we used UDS Enterprise v2.2 ([Virtual Cable, n.d.](#)) from Virtual Cable. (The same company develops the Virtual PC plugin for Moodle thus assuring full compatibility between both).
4. Template virtual machine: Is a prestored virtual machine ready to be cloned and deployed as a remote desktop to the calling student. The template is a computer image that contains all the software needed for the student to perform a specific assignment. In the present case, we created a template containing Windows 10 Professional, LabVIEW 2013, and the Quimera library, which is all the student needs to simulate a FF control loop. The Azure service provider takes the image, clones it, and launches a fresh virtual machine for each remote session.
5. Cloned virtual machine: The newly created virtual machine establishes a connection to the Moodle server, passing through the virtualization broker, using the remote desktop protocol (RDP), thus becoming available as a remote desktop for the calling student.

The virtual machines represented in [Figure 3](#) (Moodle server, virtualization broker, template virtual machine and cloned virtual machines) are all hosted on the Azure cloud. The Azure subscription must accommodate all these resources, including storage space, network communications and software licenses.

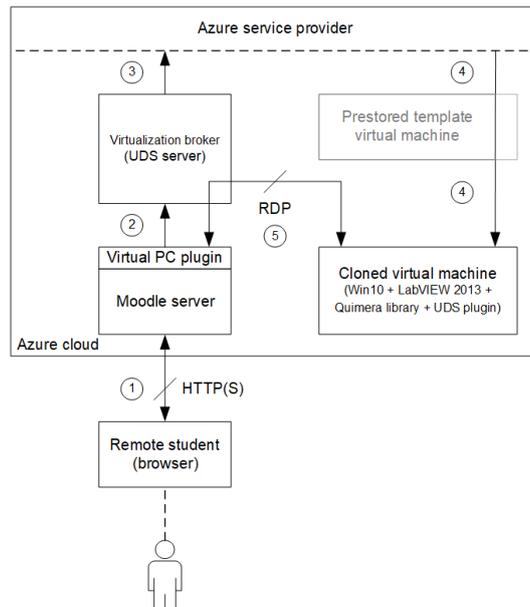


Figure 3: Virtualization infrastructure

The implementation of the virtualization infrastructure is quite demanding because it includes many steps that take some time to complete. We divided the implementation process in five main tasks described as follows:

1. Creation of base virtual machines: We started by creating and deploying virtual machines for the Moodle server (Windows 10 Professional), the virtualization broker (Linux Debian), and the template virtual machine (Windows 10 Professional), as shown in Figure 4. For Windows machines we used images provided by the Azure marketplace and followed the instructions given in "Azure VM" (2019). For the Linux machine we used an image provided by Virtual Cable, with a demo license, and followed the instructions described in pages 1-48 of UDS Enterprise (n.d.).

Home >

Virtual machines ✕

Instituto Politécnico de Setúbal

+ Add | ⌚ Reservations | ≡ Edit columns | 🔄 Refresh | 🏷️ Assign tags | ▶ Start | ⏪ Restart | 🛑 Stop | 🗑️ Delete | 📄 Services

Subscriptions: Azure subscription 1

Filter by name... | All resource groups | All types | All locations | All tags | No grouping

1 of 4 items selected

<input type="checkbox"/>	Name ↑↓	Type ↑↓	Status	Resource group ↑↓	Location ↑↓	Source	Maintenance status	Subscription ↑↓
<input type="checkbox"/>	UDS-Server	Virtual machine	Running	STE2020	West Europe	Disk	-	Azure subscription 1
<input type="checkbox"/>	Win10-Moodle	Virtual machine	Running	STE2020	West Europe	Marketplace	-	Azure subscription 1
<input type="checkbox"/>	Win10-Template	Virtual machine	Running	STE2020	West Europe	Marketplace	-	Azure subscription 1

Figure 4: Base virtual machines on the Azure portal. The machine "UDS-Server" refers to the virtualization broker, the machine "Win10-Moodle" refers to the Moodle server, and the machine "Win10-Template" refers to the pre-stored template virtual machine

2. Preparation of Windows base virtual machines: We had to install manually all the software needed for the Windows virtual machines to work properly: Moodle 3.9 in the Moodle server, and LabVIEW 2013 + Quimera library in the template virtual machine. Also in this machine, we had to install the UDS plugin (Virtual Cable 2016) and enable the remote desktop access (Start > Settings > System > Remote Desktop > Enable), so that all its clones can be accessed remotely passing through the virtualization broker.
3. Configuration of the virtualization broker: This is the most difficult task of the process as it requires many small steps:

- We start by connecting the virtualization broker to the Azure service provider. Following the instructions given in pages 49-57 of [VDI \(2020\)](#) we got [Figure 5\(a\)](#).
- Then we subscribe to the Azure clone service. Following the instructions given in pages 57-60 of [VDI \(2020\)](#) we got [Figure 5\(b\)](#).
- We register in advance all the users allowed to access the Azure clone service, including a “Moodle user” that represents the Moodle server that connects through the Virtual PC plugin. Following the instructions given in section 4.2.4 of [UDS Enterprise \(2019\)](#) we got [Figure 5\(c\)](#).
- We create an “OS manager” to indicate the operating system and the policy of persistence of the cloned desktops. Following the instructions given in section 4.4.3 of [UDS Enterprise \(2019\)](#) we got [Figure 5\(d\)](#).
- We create a “transport” to connect to the cloned desktops. Following the instructions given in section 4.6.6 of [UDS Enterprise \(2019\)](#) we got [Figure 5\(e\)](#).
- Finally, we create a “service pool” where we tie all the previous pieces (Azure clone service + authenticated users + OS manager + transport), and set the behaviour of the service (number of initial available clones, maximum number of clones, and number of clones to keep in cache). Short after the service pool is created, the Azure service provider starts cloning the template virtual machine as shown in [Figure 5\(f\)](#).



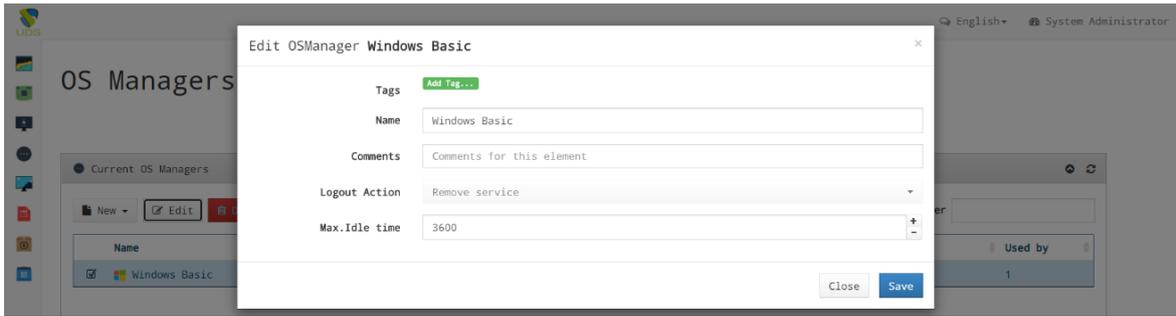
(a)



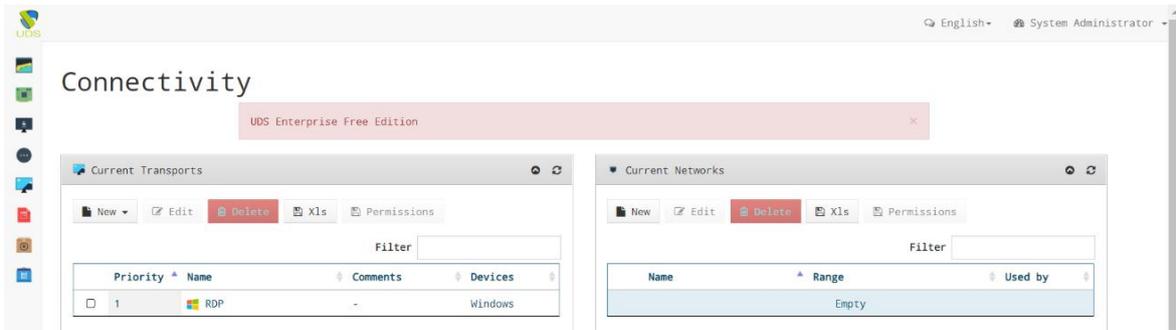
(b)



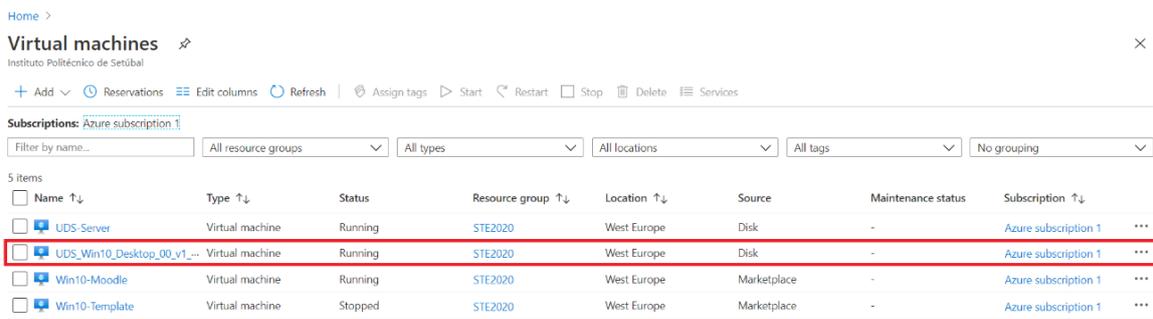
(c)



(d)



(e)



(f)

Figure 5: Configuration of the virtualization broker: (a) connection to the Azure service provider; (b) subscription to the Azure clone service; (c) authorized users; (d) OS manager; (e) connectivity using the RDP protocol; (f) inside the squared box we can see a new virtual machine being cloned from the template machine

- Installation and configuration of the Virtual PC plugin: This plugin is need for the Moodle server to connect to the virtualization broker. It makes part of the official list of Moodle plugins and its installation follows the same steps as any other official plugin (Moodle 2020). After the installation, we followed the instructions given in UDS Enterprise (2020) to connect it to the virtualization broker.
- Creation of the learning activity: Finally, on Moodle, we were able to create a learning activity that makes use of the Virtual PC plugin, as shown in Figure 6. By clicking on the button “UDS Access to Virtual PC”, the student has access to a remote desktop with all the components needed to run the Quimera simulator.

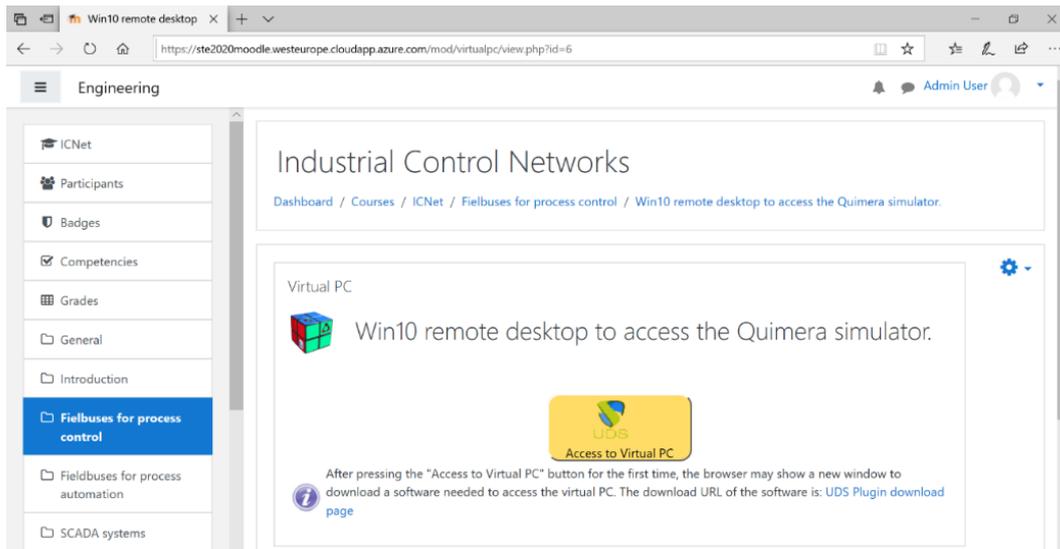


Figure 6: Virtual PC activity giving remote access to the Quimera simulator

4. Results and Conclusions

To deliver the Quimera simulator remotely we just have to create a Virtual PC activity on a Moodle’s course, as shown in Figure 6. A click on the button “UDS Access to Virtual PC” triggers the following chain of events: first, the virtualization broker asks for the Azure service provider to make a new clone of the template machine; then, the virtualization broker establishes a remote desktop connection with the newly created machine; and finally, the virtualization broker channels the connection to the Moodle server. All this is transparent to the student, who just sees is a clean, fresh remote desktop ready to run the Quimera simulator, as shown in Figure 7.

The virtualization infrastructure was tested with a limited number of students on an elective course with initial available clones = 2, maximum number of clones = 10, and number of clones to keep in cache = 2. Everything went well, without serious anomalies, making us believe that this infrastructure is capable of being extended to other courses with more students.

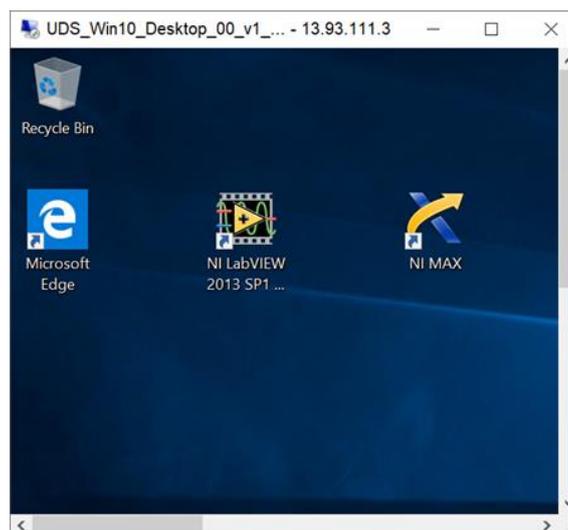


Figure 7: Remote desktop ready to run the Quimera simulator

References

- "Azure VM". 2019. <https://docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal>.
- Glanzer, D. 2003. "Foundation Fieldbus technical overview". http://www.fieldbus.org/images/stories/technology/developmentresources/development_resources/documents/techoverview.pdf.
- Moodle. n.d. "Virtual PC". Accessed November 2, 2020. https://moodle.org/plugins/mod_virtualpc.
- Moodle. 2020. "Installing plugins". https://docs.moodle.org/39/en/Installing_plugins.
- UDS Enterprise. n.d. "VDI with UDS Enterprise & Microsoft Azure". Accessed on November 2, 2020. https://www.udsenderprise.com/media/filer_public/24/5f/245fb732-529e-4f17-982a-93616d47cb0d/vdi_with_uds_enterprise_and_microsoft_azure.pdf.
- UDS Enterprise. 2019. "Installation manual, administration and user guide of UDS Enterprise 2.2.1". https://www.udsenderprise.com/media/filer_public/5e/9f/5e9f0866-63b2-45fe-8b8f-e7c17c71eb4a/uds-enterprise-221-user-administration-guide_rev1.pdf.
- UDS Enterprise. 2020. "UDS Enterprise Virtual PC plugin for Moodle installation". https://www.udsenderprise.com/media/filer_public/d2/74/d2748914-b979-45d4-af95-2fd9578e30f5/uds_enterprise_virtual_pc_plugin_for_moodle_installation.pdf.
- Viegas, V., O. A. Postolache, P. M. B. S. Girão, and J. M. D. Pereira. 2016. "Quimera: The easy way to simulate Foundation Fieldbus applications". *Computer Applications in Engineering Education* 24, no. 6: 914-25. <https://doi.org/10.1002/cae.21761>.
- Virtual Cable. n.d. "UDS Enterprise". Accessed November 2, 2020. <http://www.virtualcable.net/en/uds-enterprise/>.
- Virtual Cable. 2016. "UDS Enterprise components: UDS Plugin". <http://www.virtualcable.net/en/uds-enterprise-components-uds-plugin/>.