

PLANTAS VIRTUALES BASADAS EN ARDUINO PARA LABORATORIO DOCENTE DE INFORMÁTICA INDUSTRIAL

Raquel Romo Gómez

Dpto. Ing. Sistemas y Automática, Univ. Valladolid, rlromo@gmail.com

Rogelio Mazaeda Echevarría

Dpto. Ing. Sistemas y Automática, Univ. Valladolid, rogelio@cta.uva.es

Rubén Martí Martínez

Dpto. Ing. Sistemas y Automática, Univ. Valladolid, ruben@autom.uva.es

Resumen

En la presente contribución se describe el diseño de un laboratorio docente de autómatas programables que utilizan como planta a controlar simulaciones realizadas en un microcontrolador de tipo Arduino. Se describe una metodología para el modelado y simulación de estas plantas virtuales. Se propone un criterio de validación de los modelos desarrollados especialmente adaptado al objetivo docente propuesto.

Palabras Clave: Educación en Control, Arquitecturas Concurrentes, Computación Paralela, Controladores Lógico-Programables, Control Secuencial

1 INTRODUCCIÓN

La especialidad de la ingeniería de control y la automática estudia, en un sentido amplio, la sustitución parcial o total de la actividad humana en el control y supervisión de los procesos industriales. El control automático es una disciplina asentada en las sólidas bases matemáticas que le proporciona la teoría de sistemas dinámicos [1]. Las herramientas que brinda tienen un muy alto grado de generalidad y puede ser aplicadas, de forma transversal, a diferentes campos aparentemente muy disímiles entre sí. Por otra parte, la tecnología específica utilizada para dar cuerpo a los controladores ha cambiado históricamente [4], pero en la actualidad la industria moderna es inconcebible sin la utilización de los computadores digitales y las redes informáticas.

La informática permea actualmente todas las esferas de la actividad industrial. Se utiliza a todos los niveles: desde la planificación de los recursos

necesarios a largo plazo (sistemas **ERP**), la administración de carácter más operativo de la producción (sistemas **MES**), hasta la supervisión y control directo de la planta industrial. En este último caso, la práctica totalidad de las soluciones de control y supervisión modernas utilizan medios de cómputo digitales.

Por ejemplo, un dispositivo como el autómata programable (**PLC** de las siglas en inglés de *Programmable Logic Controller*), utilizado de forma masiva en la industria actual, es un computador digital de propósito específico, diseñado con unas características constructivas y funcionales que están especialmente adaptadas a las tareas de control. En un principio y todavía en gran medida, el propósito era el de implementar el control lógico-secuencial en procesos de fabricación discreta, sustituyendo la anterior lógica cableada, o para implantar enclavamientos de seguridad en procesos de naturaleza diversa. Pero, cada vez en mayor medida, el **PLC** es utilizado en la implantación de controladores para procesos continuos.

En este contexto, el impacto de los ordenadores y de la informática en la docencia de la especialidad es, si cabe, todavía mayor. Para poner sólo un ejemplo, el modelado y la simulación mediante ordenador digital del proceso a controlar o supervisar, constituyen una herramienta docente inestimable, que permite traer al aula y poner a disposición del alumno, una descripción realista de los procesos industriales.

En el presente artículo se describe el diseño de un laboratorio docente de autómatas programables donde las plantas a controlar son implementadas por simulación utilizando el conocido microcontrolador **Arduino**. La estructura del artículo es la siguiente. En la sección 2 se explican los antecedentes del proyecto que se describe, en la sección 3 se detalla la arquitectura propuesta, la

sección 4 desarrolla una metodología de modelado de las plantas a controlar que incorpora un criterio de validación de los modelos especialmente adaptado al propósito que se persigue. Finalmente, en la sección 5 se brindan ejemplos de plantas modeladas para el laboratorio docente.

2 LA DIMENSIÓN PRÁCTICA EN LA DOCENCIA DEL CONTROL Y LA AUTOMATIZACIÓN

La docencia en Control y Automatización debe cuidar el balance entre el estudio teórico y la aplicación práctica. En el laboratorio, el estudiante aprende a integrar todo una serie de factores, que incluyen, además del dominio de la teoría de control, todo el conocimiento previo de otros campos de la ingeniería a la que pertenece el proceso específico que se desea controlar, junto con habilidades de programación de los dispositivos programables digitales y en la utilización de la instrumentación necesaria: sensores, actuadores, conversores A/D y D/A, entre otros.

En la elección del tipo de planta a utilizar en los laboratorios de la especialidad hay que tener en cuenta muchos factores [2]. Idealmente una buena planta docente debe ser simultáneamente barata, segura, sencilla y fácil de comprender. Debe poder ilustrar con facilidad los aspectos teóricos de interés pero al mismo tiempo no debe alejarse demasiado de las plantas realmente existentes en la industria. La dificultad para encontrar el equilibrio adecuado entre requerimientos tan diversos resulta agravada por el hecho de la mencionada transversalidad de la disciplina de control: sería conveniente que las plantas de laboratorio representen las diferentes áreas, por ejemplo, la mecatrónica (control de velocidad y posición de motores), o plantas típicas de la industria de procesos (control de nivel, de temperatura y de caudal), o de sistemas discretos como prototipos de pequeños sistemas de fabricación.

Obviamente, el uso de plantas simuladas por ordenador, ofrece ventajas importantes a la hora de cumplir con algunos de estos requerimientos. La creación de una gran variedad de plantas, a coste mínimo, específicamente adaptadas para ilustrar algún elemento teórico de interés, no sería, en este caso, un problema. Sin embargo, la utilización de modelos y plantas simuladas en la enseñanza práctica de la automatización y el control, aunque imprescindible, no debe nunca sustituir completamente la exposición del alumno a prototipos físicos de plantas reales. Como bien se ha expresado en [5]:

“La simulación es ciertamente útil pero no puede reemplazar al equipo físico real. Es importante que los alumnos se vean expuestos desde el principio, a los ‘caprichos’ y ‘contrariedades’ del mundo real – ellos deben tener que lidiar con el ruido de medida, la fricción, las válvulas y motores que se saturan. También es muy importante para ellos el adquirir alguna experiencia en hacer que un sistema real funcione.”

La clave, por supuesto, es el combinar ambos enfoques en el plan de estudios: una exposición frecuente al estudio mediante simulación, menos costoso y más flexible, vinculado a sesiones frente a plantas reales, más caras pero también más realistas.

Aquí, proponemos una solución que se encuentra a medio camino entre el enfoque puramente basado en simulación y la utilización de plantas reales. La motivación surge a partir de la necesidad de dar una solución razonablemente económica para la utilización docente de un conjunto de autómatas programables, modelo S7-1200, de la firma Siemens [7], de muy buenas características técnicas. Estos PLCs se encontraban ociosos y en cantidad suficiente como para dotar un aula de prácticas. La alternativa preferible de acometer la creación de plantas reales, a las cuales aplicar los mencionados autómatas, no resultaba abordable por consideraciones de su coste económico y del escaso espacio físico existente para su despliegue. Téngase en cuenta, además, que un requerimiento natural desde el punto de vista de la organización del proceso docente, era el de disponer de un conjunto de plantas similares para adiestrar simultáneamente a los alumnos divididos en grupos de trabajo de tamaño razonable. Por otra parte, la transversalidad ya mencionada de la disciplina de control y de la informática industrial, que se imparten en varias especialidades ingenieriles, recomendaba el uso de plantas diversas que ilustraran los diferentes campos de aplicación.

La estrategia finalmente considerada como más adecuada, dadas las restricciones antes descritas, ha sido la de crear simulaciones de las plantas a automatizar, pero hacerlo mediante tarjetas electrónicas dotadas de un microcontrolador, que estuvieran físicamente conectadas, mediante *hardware*, a las líneas de entrada y salida de cada uno de los autómatas reales. Esta solución se colocaría a mitad de camino en el espectro que va de la simulación completa mediante ordenador, donde tanto al autómata como la planta no serían más que programas informáticos, a la solución completamente realista con el autómata real

controlando la planta real. La placa electrónica con microcontrolador elegida para llevar a cabo la emulación de las plantas ha sido una de las variantes del conocido **Arduino** [8].

Es cierto que, como es evidente, en el laboratorio virtual que se propone, no están presentes todos los ‘*caprichos*’ y ‘*contrariedades*’ inherentes al mundo real que se mencionaban en [5], pero sí que permanecen algunos. Por ejemplo, el ruido achacable a los sensores de una planta real está, por supuesto, ausente, puesto que estos últimos tampoco existen; pero en su lugar puede aparecer ruido asociado a la transmisión de las señales eléctricas entre el **Arduino** y el PLC.

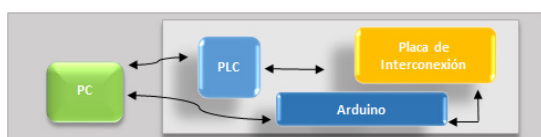


Figura 1: Solución de planta virtual

Por otra parte, el hecho de que el programa de control del autómatas y la simulación de la planta se ejecuten en procesadores diferentes, brinda el tipo de comportamiento concurrente (más bien paralelo en este caso) y asíncrono entre los eventos y variables continuas a controlar (en la simulación) y las acciones de control (en el PLC real), que es de naturaleza muy similar a la que existiría entre este último y la planta real.

Una desventaja del método propuesto es la falta de realimentación visual y auditiva desde la planta. Esta deficiencia es una realidad que inevitablemente se debe aceptar pero que puede mejorarse en la planta virtual propuesta, hasta cierto punto, utilizando las propias herramientas que brinda el ambiente de desarrollo del autómatas utilizado para la creación de interfaces gráficas con el usuario (**HMI- Human Machine Interface**). El uso de estos recursos para este propósito, brinda a su vez, de manera colateral una nueva oportunidad de entrenamiento para el alumno. Nótese, que en muchas situaciones reales, las sugerencias visuales y auditivas también pueden estar ausentes

La idea planteada es asimilable al concepto de *Hardware in the Loop* (HIL) utilizado para la puesta a punto de sistemas empujados de control en plantas complejas.

3 SOLUCIÓN PROPUESTA DE PLANTAS VIRTUALES

La solución diseñada (fig. 1) consiste en conectar mediante *hardware* las entradas y salidas del PLC

real con las correspondientes del **Arduino**. Este último ejecutará la simulación dinámica representando la planta a controlar, mientras que el programa de control residirá en el PLC. La conexión eléctrica entre el **Arduino** y el PLC se realiza mediante una placa diseñada al efecto. Esta placa debe resolver los problemas de compatibilidad existentes.

Cada puesto de trabajo incluye un ordenador convencional que se comunica con el PLC a través de un cable de **Profinet** y con **Arduino** mediante un cable **USB**. Este ordenador deberá estar dotado de los programas necesarios para interactuar tanto con el PLC como con el **Arduino**.

El controlador **SIMATIC S7-1200** es el modelo modular y compacto que **Siemens** propone para pequeños sistemas. Su diseño compacto, su bajo coste y sus potentes funciones, son elementos que hacen que salga a cuenta el utilizar estos autómatas incluso para tareas sencillas [7]. El autómatas consta, en su módulo central, de la **CPU 1214C AC/DC/Relé**. Al sistema base, se le ha añadido un módulo adicional (**SB1232**) para dotarlo de una salida analógica. De manera, que en resumen, se tienen en el PLC: 14 entradas digitales de tipo sumidero/fuente con tensión nominal de 24 V DC a 2 mA, 10 salidas digitales de tipo relé de 5-30 V DC con corriente de hasta 2 A, 2 entradas analógicas de tensión unipolar de 0-10V y, finalmente, una salida analógica de tensión (0-10V) o intensidad (0-20 mA).

Adicionalmente, se dispone del entorno de desarrollo integrado **TIA Portal V11**, que permite desarrollar, editar y observar la lógica del programa. También incluye herramientas para gestionar y configurar todos los dispositivos del proyecto, tales como controladores y dispositivos **HMI**.

Por otra parte, el **Arduino** constituye una plataforma electrónica abierta (*open hardware*) para la creación de prototipos basada en *software* y *hardware* libre. Las características técnicas de esta placa microcontroladora, junto a su precio altamente competitivo, al modelo abierto que representa y a la relativa facilidad de su programación a través de un entorno integrado que también se suministra, ha hecho que la misma experimente un éxito sin precedentes en ese mercado. La versión que utilizamos para este proyecto es la más sencilla, el **Arduino UNO**, que posee 14 puertos que pueden ser programados como entradas o salidas, con voltaje de 5 V y que pueden suministrar o consumir, en dependencia del caso, hasta 40 mA. Posee además 6 entradas analógicas de rango 0 - 5V convertibles a una

palabra de 10 bits. También posee 6 salidas codificadas en modulación de ancho del pulso (**PWM**) que pueden usarse como salidas analógicas (0- 5V con una resolución de 8 bits).

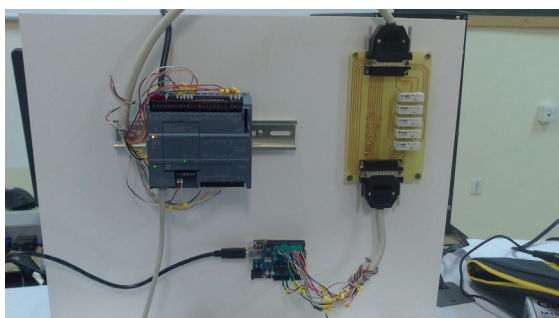


Figura 2. Interconexión entre PLC y **Arduino** con placa adaptadora

En general, dadas las características eléctricas antes expuestas, no se puede realizar la conexión directa entre el autómatas y el **Arduino**. Se requiere una placa adaptadora (fig. 2) que se encarga de garantizar la compatibilidad de las señales. Especialmente, por ejemplo, entre las salidas digitales del **Arduino** y las correspondientes entradas del PLC se necesitan sendos relés (fig. 3).

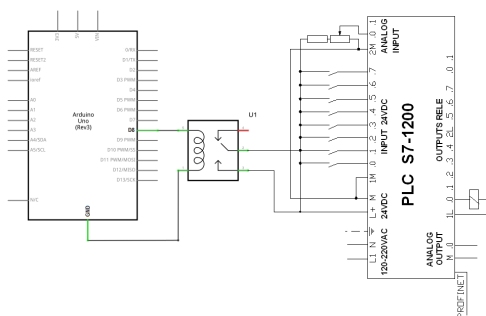


Figura 3. Detalle de la interconexión

En el caso de la conexión analógica de la salida del **Arduino** a la entrada del PLC, se ha preferido realizar la adaptación vía *software*, definiendo el rango de tensión de entrada del autómatas de los 0 – 30 V originales, a los 0 – 5 V que el **Arduino** es capaz de entregar.

4 MODELADO Y SIMULACIÓN PARA LA CREACIÓN DE LAS PLANTAS VIRTUALES.

El modelado y la simulación es una actividad básica que debe ser enfocada de forma sistemática (ver, por ejemplo [3]). El proceso debe comenzar con una formulación clara del problema y de los objetivos que se persiguen con el modelo. Cumplido este importante primer paso, se puede pasar a plantear el modelo conceptual, postulando las relaciones matemáticas y lógicas, que reflejan

la estructura y funcionamiento del sistema real, de una manera lo suficientemente exacta, como para satisfacer las expectativas representadas por los objetivos del modelo y las suposiciones de modelado que de estos objetivos se derivan. Luego se pasa a la implementación del modelo utilizando los recursos de la herramienta de simulación de que se disponga. De manera paralela se va realizando el proceso de verificación, que consiste en comprobar que la mencionada traslación desde el modelo al programa de ordenador que permite su posterior simulación dinámica, se ha realizado sin introducir errores. Si la verificación concluye exitosamente, sigue el importante paso de validación, que es un proceso de mayor calado que consiste en comprobar que el modelo realizado no puede ser probado como falso a la luz de los datos obtenidos desde la planta real, siempre teniendo en cuenta los objetivos declarados. El proceso de modelado-validación es de naturaleza iterativa. Si los experimentos simulados fallan al explicar los datos reales, tanto el modelo conceptual como las suposiciones de las que éste parte, deberán ser replanteados. El ciclo de modelado concluye con la documentación del modelo desarrollado y su posterior puesta en producción.

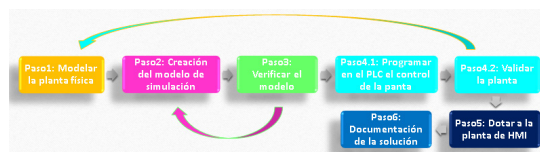


Figura 4. Ciclo de modelado y simulación

En el caso que nos ocupa, el ciclo genérico previamente descrito, ha sido adaptado de la manera que se muestra la figura 4. El propósito que se persigue en nuestro caso es que los modelos desarrollados sean lo suficientemente fieles al referente real, como para que sean de utilidad en la docencia de la programación de los autómatas programables que se utilizarán para su control. Un esfuerzo adicional en crear modelos más exactos que lo estrictamente necesario, podría considerarse innecesario.

Este objetivo debe permear todo el ciclo de trabajo representado en fig. 4. En particular decidirá en gran medida la elección de las suposiciones de modelado y el planteamiento matemático del modelo y la validación del mismo como se explicará más adelante.

La implementación del modelo (paso 2) presenta en este caso algunas dificultades importantes. En principio, el modelo de simulación tiene que ser programado en el microcontrolador. Esta programación deberá incluir la codificación de los algoritmos de simulación dinámica necesarios y

esta tarea podría haber resultado muy ardua de utilizar exclusivamente los recursos de programación originales del **Arduino**.

Afortunadamente existe una *toolbox* de **Simulink/Matlab** denominada *Simulink Support Package for Arduino Hardware* [6] que facilita enormemente esta tarea de creación del modelo y simultáneamente también la tarea de verificación. La herramienta permite crear el modelo desarrollado utilizando los recursos de **Simulink**, conectando los bloques necesarios, siguiendo el paradigma de procesamiento de señales que esta conocida herramienta utiliza. Al mismo tiempo, se pueden realizar todos los experimentos simulados que se crean convenientes, a modo de verificación, hasta que el modelador se persuade de que el modelo implantado en **Simulink** cumple todas sus expectativas previas. En este momento, se utilizan los bloques especiales que la *toolbox* provee y que representan las entradas y salidas analógicas y digitales del **Arduino**, especificadas por el número de su patilla y que se conectan a las correspondientes entradas y salidas del modelo de **Simulink**. Posteriormente, el ejecutable es generado y descargado en el microcontrolador. El ejecutable incluye, obviamente las ecuaciones del modelo, pero también el código del algoritmo de integración de paso fijo utilizado.

En la metodología que se propone, la importante etapa de validación del modelo se enfoca de forma novedosa. Como el objetivo que se persigue es el de obtener una plataforma simulada que sirva de banco de pruebas para la docencia de la programación de autómatas, consideraremos que un modelo desarrollado y finalmente implementado en la plataforma **Arduino** será válido (no se declarará falseado por los datos reales) cuando esencialmente el mismo programa de control del autómata sea capaz de controlar tanto la planta virtual simulada como la planta real que le sirvió de referente. Nótese que los programas de **PLC** aludidos pueden cubrir toda la gama que estos modernos dispositivos soportan. En general, serán programas capaces de controlar sistemas híbridos: aquellos que pueden ser descritos a partir de la interacción de variables que evolucionan de forma continua en el tiempo, con la existencia de eventos discretos.

De manera que, en la metodología propuesta, la validación abarca los pasos 4 y 5 en la fig. 4. Nótese que se exige que el programa de **PLC** que controla tanto la planta real como la virtual sea esencialmente el mismo aunque no necesariamente idéntico. Esto es, el programa debe tener la misma estructura pero sería permisible que algunos parámetros relacionados, póngase por caso, con la

dinámica de la planta a controlar, como podrían ser, por ejemplo, los parámetros de sintonía de algún controlador PID, pudieran modificarse sin afectar por ello nuestra confianza en el modelo realizado porque tampoco afectaría su utilidad docente. Esta salvedad se hace además porque no se tienen garantías suficientes de que la implementación del modelo en el **Arduino**, en definitiva una plataforma de bajo coste y dirigida a un mercado muy amplio, pueda garantizar exigencias demasiado estrictas de tiempo real en la reproducción de las dinámicas de la planta que le sirve de referencia.

Una aclaración final, la planta virtual estará sometida a un proceso de validación constante. Nótese que el instructor o el desarrollador del modelo pueden plantearse el utilizar un programa inicial para validar el trabajo, pero el modelo una vez puesto en producción, en el laboratorio docente, estará sometido a un proceso permanente de revalidación en presencia de los programas propuestos por los estudiantes, siempre diferentes, algunos correctamente diseñados y otros con errores de diferente naturaleza, y en todos los casos el comportamiento deberá ser el esperable de la planta real.

Una vez validado el modelo, el ciclo propuesto incluye un paso de creación de la interfaz **HMI** (paso 5 en la fig. 4) basado en el ordenador de sobremesa del puesto de trabajo, utilizando los recursos que en ese sentido ofrece el **TIA Portal** de **Siemens**. La creación del **HMI** busca mitigar la falta de realimentación visual desde la planta virtual. Por otra parte, su eventual modificación por el estudiante, presenta oportunidades adicionales de aprendizaje. Finalmente, y previamente a su puesta en producción en el aula, el modelo deberá ser debidamente descrito (paso 6 fig. 4) en documento adjunto al guión de prácticas.

5 EJEMPLOS DE PLANTAS VIRTUALES

Para ilustrar la viabilidad del proyecto e ilustrar la metodología propuesta se han creado dos plantas virtuales que a continuación se describen.

5.1 PLANTA DE DEPÓSITO DE AGUA PARA CONTROL SECUENCIAL DE NIVEL Y TEMPERATURA

En este primer ejemplo, se quiere modelar un sistema de consistente en un depósito de agua cuya representación esquemática podemos ver en la figura 5, una fotografía de la instalación real se tiene en la figura 6. Este sistema físico consta de

dos sensores, utilizados para determinar la situación de depósito lleno (sensor₂) y el otro que se usará para detectar la ausencia de agua (sensor₁). Se tienen dos bombas: una bomba de suministro de agua (bomba₂) y otra para drenar el líquido hacia el exterior (bomba₁). La instalación posee además una resistencia eléctrica de calefacción y un sensor de temperatura (sensor₃).

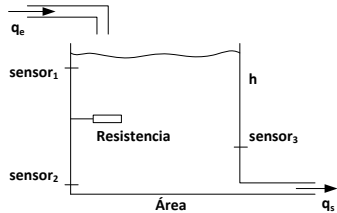


Figura 5. Diagrama de la planta



Figura 6. Fotografía de la instalación física

Las ecuaciones que describen el modelo de la planta se muestran en (1)-(2). El nivel (h) del depósito evoluciona de acuerdo a (1) en dependencia de la diferencia entre los caudales impuestos por las bombas de entrada (q_e) y de salida (q_s). Se ha considerado el área del depósito (A) como constante, al igual que la densidad del agua. Se han despreciado otras corrientes de pérdida, como por ejemplo, las debidas a la evaporación. La segunda ecuación (2), que modela el aumento de la temperatura del agua en función del tiempo, depende de la potencia disipada en la resistencia eléctrica (Q), de la capacidad calorífica del agua (cp_{H_2O}) y de la masa total de agua existente en cada momento (m).

$$A \frac{dh}{dt} = q_e - q_s \quad (1)$$

$$\frac{dT}{dt} = \frac{Q}{m \cdot cp_{H_2O}} \quad (2)$$

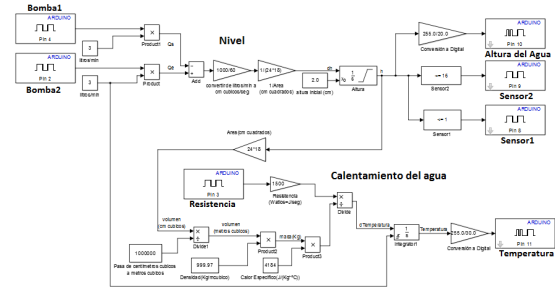


Figura 7. Implementación en Simulink

El modelo de Simulink que se cargará en Arduino es el que vemos en la Figura 7, en el podemos observar las entradas digitales de Arduino, las bombas (bomba₁ y bomba₂), y la resistencia, así como las salidas digitales (sensor₁ y sensor₂) y las salidas analógicas que se corresponden con la temperatura y con la altura del agua, estas dos últimas representan las variables a controlar.

El control secuencial de la planta se realizará de acuerdo al grafset de la figura 8.

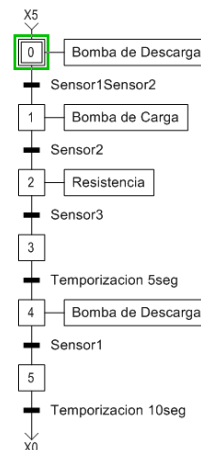


Figura 8. Grafset del control secuencial

El funcionamiento general es el siguiente: al comenzar la simulación, primero se vacía el depósito, evento cuya ocurrencia es indicada por el sensor₁. Una vez vacío, se vuelve a llenar hasta que el sensor₂ reporte que se ha alcanzado la altura necesaria, inmediatamente después se enciende la resistencia y se calienta el agua hasta que se alcance la temperatura de referencia que se haya indicado, luego se esperan cinco segundos y finalmente se procede, de nuevo, al vaciado del depósito. Diez segundos después se reinicia el ciclo.

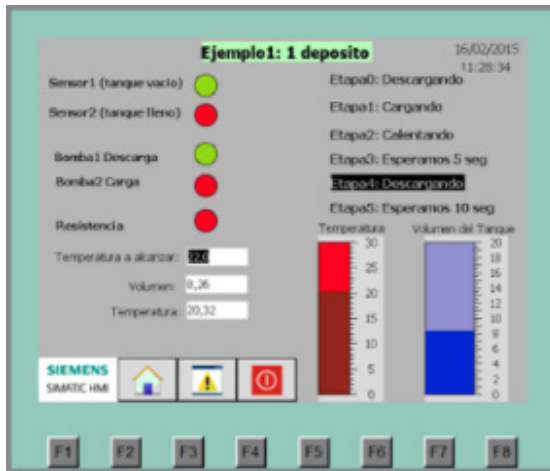


Figura 9. Pantalla HMI

El último paso será dotar a la instalación de una pantalla **HMI**. En la figura 9 podemos ver la pantalla creada para el ejemplo, que muestra las entradas y salidas activas, la etapa del control secuencial actualmente vigente y gráficas de barras para el seguimiento de la evolución de la temperatura y el nivel.

5.2 PLANTA DE DOS DEPÓSITOS PARA CONTROL CONTINUO DE NIVEL

La segunda planta virtual exige del **PLC** un control de tipo exclusivamente continuo. Se trata en este caso de dos depósitos abiertos conectados en serie, y situados a la misma altura. El caudal de entrada al primer depósito puede ser fijado a voluntad mediante, por ejemplo, una bomba, y es la variable manipulada del problema de control. La variable de proceso a controlar es la altura del agua en el segundo depósito.

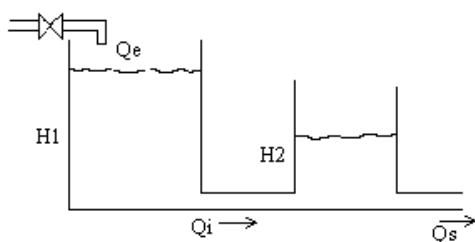


Figura 10. Diagrama de la planta de dos depósitos

El modelo dinámico de la planta es mostrado en (3)-(4). Las suposiciones de modelado son similares a las del caso previo, pero ahora la salida de agua desde el primer y del segundo depósito se realiza por gravedad y depende de la raíz cuadrada de las presiones, y por tanto de las alturas de la columna de agua, en los extremos de cada válvula. La segunda válvula descarga hacia la atmósfera. Las constantes k_1 y k_2 están relacionados con las

capacidades de la válvula de interconexión y la de salida respectivamente.

$$A_1 \frac{dh_1}{dt} = q_e - k_1 \sqrt{h_1 - h_2} \quad (3)$$

$$A_2 \frac{dh_2}{dt} = k_1 \sqrt{h_1 - h_2} - k_2 \sqrt{h_2} \quad (4)$$

La implementación del modelo en **Simulink** se muestra en la figura 11. Se puede advertir la presencia de los bloques del *toolbox* para **Arduino** que representan las entradas y salidas analógicas.

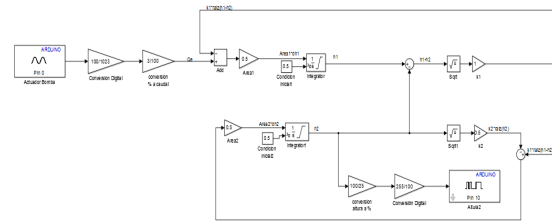


Figura 11. Diagrama en **Simulink** del modelo de los dos depósitos

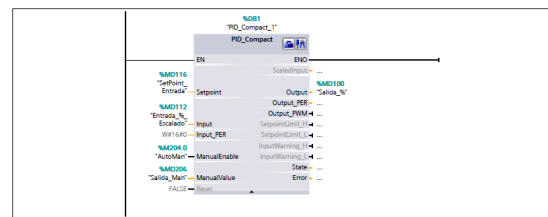


Figura 12. Bloque controlador PID de STEP 7

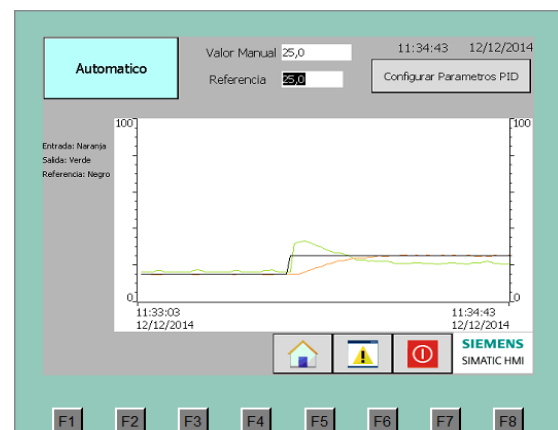


Figura 13. Pantalla HMI para ejemplo continuo

El programa del **PLC** utilizado para la validación hace uso de un bloque de organización de interrupción cíclica de **STEP 7** para alojar un bloque funcional estándar del lenguaje de tipo **PID**, tal como se muestra en la fig. 12.

Finalmente, el **HMI** diseñado (fig. 13) muestra la evolución dinámica continua del control implementado en el **PLC** actuando sobre la planta virtual.

CONCLUSIONES

Se ha descrito el diseño de un laboratorio virtual para la enseñanza de la programación de autómatas programables utilizando un enfoque de tipo *hardware in the loop*. La planta a controlar se simula en una placa microcontroladora **Arduino**. La conexión entre el autómata real y la planta simulada se realiza conectando eléctricamente las entradas y salidas analógicas y digitales correspondientes. La instalación resultante permitirá dotar un aula de prácticas de autómatas a un coste mínimo, en un espacio reducido, brindado a cada grupo de estudiantes el mismo conjunto de plantas a controlar. Se considera que la estrategia propuesta constituye un buen compromiso, a medio camino entre la solución ideal de contar con un prototipo real y un enfoque exclusivamente basado en simulación por ordenador.

Se ha explicado además una metodología para el diseño de nuevas plantas virtuales, en la que se ha hecho especial énfasis en un enfoque novedoso para la validación de las plantas virtuales diseñadas, que resulta compatible con el declarado objetivo docente de los modelos.

Agradecimientos

La contribución aquí descrita ha sido desarrollada en el marco del Proyecto de Innovación Docente (PID 14-15 N° 39) de la Universidad de Valladolid titulado “Armonización de las asignaturas relacionadas con la Informática Industrial en el Grado de Electrónica Industrial y Automática”

Los autores también desean agradecer la ayuda brindada por el MINECO a través de su programa DPI2012-31859.

Referencias

- [1] Albertos, P., Alonso, A., Ayza, J., Dornido, S., Etxebarria, V., Fuente, M. J., Galán, R., Gordillo, F., Martínez, M., Prada, C., Quevedo, J., Sagues, C., Santos, M., 2009. Libro Blanco del Control automático. Comité Español de Automática.
- [2] Balchen, A., Handlykken, M., Tyss, A., 1981. The need for better laboratory experiments in control engineering education. In:

Proceedings 8th IFAC Triennial World Congress, Kyoto, Japan.

- [3] Banks, J. (1998), Handbook of Simulation. Principles, Methodology, Advances, and Practice. Wiley Interscience. Estados Unidos.
- [4] Bennet, S., 1996. A brief history of automatic control. IEEE Control Systems, 17–15.
- [5] Kheir, N., Astrom, K., Auslander, D., Cheok, K., Franklin, G., Masten, M., & Rabins, M. (1996.). Control Systems Engineering Education. Automatica, 147-166. Vol. 32.
- [6] Mathworks Simulink Team (2013). Simulink Support Package for Arduino Hardware.
- [7] SIEMENS. (s.f.). Controlador programable S7-1200, Manual de Sistema.
- [8] Tojeiro Calaza, G. (s.f.). Taller de Arduino: Un enfoque práctico para principiantes. Marcombo.