

ARDILLA: PLATAFORMA DE PRÁCTICAS DOCENTES ONLINE MEDIANTE ARDUINO

David Úbeda, Arturo Gil, J.A. Lucas, L.M. Jiménez, Óscar Reinoso
Universidad Miguel Hernández. Dpto. Ingeniería Sistemas Industriales. Avda. de la Universidad, S/N
Edificio Quorum V. Elche (Alicante)
{ubeda, arturo.gil, luis.jimenez, o.reinoso}@umh.es, jalucasmartinez@gmail.com

Resumen

El presente documento pretende servir de apoyo al profesorado de asignaturas de programación en Ingenierías, integrando en sus prácticas docentes una plataforma online en la cual se pueden realizar las mismas en entornos diseñados para implementar su funcionamiento, en diferentes módulos hardware creados a tal efecto. Para ello se parte desde Arduino, una plataforma de hardware libre con entradas/salidas analógicas y digitales, y a partir de ahí se han implementado además de las placas de circuito impreso para los módulos, toda la de programación de servidores de entrega de prácticas, compilación online y filtrado de datos de dicha compilación, y muestra de estadísticas de utilidad relevante para el profesor y el alumno.

Palabras Clave: Arduino, C/C++, prácticas docentes, ingeniería.

1 MOTIVACIÓN

Una de las principales dificultades a las que se debe enfrentar el profesor universitario en los últimos años en la asignatura de programación en C/C++, es de falta de motivación por parte del alumnado. Son alumnos de primer curso los que reciben dichas asignaturas en la titulación de Ingeniería Industrial, y se ha podido comprobar que éstos se suelen motivar enormemente mediante un enfoque tangible de las asignaturas, con prácticas donde puedan visualizar contenidos de las mismas aplicados en experimentos realizados en tiempo real. Dicho hábito, probablemente haya sido adquirido en Educación Secundaria.

Aunque las asignaturas de programación C/C++ sí que permiten un enfoque muy práctico, al mismo tiempo es una materia la cual nunca les ha sido impartida, por lo que supone un gran esfuerzo respecto al resto de asignaturas. Adicionalmente, la programación requiere por parte del programador una visión esquemática de la solución a los problemas que se deben resolver mediante código, ya sea

implementar algoritmos o cualquier otro tipo de aplicaciones.

Para solucionar uno de los problemas planteados, el de la ausencia de motivación, los profesores responsables de estas asignaturas, hemos tratado de buscar un enfoque eminentemente práctico mediante simulación en el PC de distintas situaciones reales. Sin embargo, resultaba bastante complejo obtener una reacción convincente por parte del alumnado: la asistencia a clases prácticas seguía cayendo hasta umbrales del 50%, e incluso a medida que nos aproximábamos al final de curso, ésta disminuía hasta un 20 %. Como consecuencia, el examen final era un completo fracaso debido a la poca soltura adquirida durante el curso para programar. Por tanto, se debía buscar otra solución, descartando la simulación.

Se optó por prácticas diseñadas en entornos reales, con dispositivos hardware reales, mediante los cuales los alumnos pudieran visualizar el resultado de su programación. Esta solución se tomó debido a que se ha comprobado que, aparentemente, las asignaturas de microcontroladores, impartidas en cursos superiores, suelen motivar al alumnado en mayor medida, puesto que en éstas se comienza a usar dispositivos hardware capaces de enfocarlos para generar prácticas, cuyo uso puede resultar llamativo, como es el caso del uso de sensores de temperatura, fotorresistencias o LDR[10], y LCD's.

Una vez vislumbrada la posibilidad que podíamos motivar y despertar el interés en el alumnado mediante estos dispositivos, la cuestión principal se trasladaba ahora en como englobar todo esto en una programación C/C++ sin interferir con otras asignaturas de circuitos y sin que éstos debieran aprender nada de ellos, siendo los mismos transparentes al usuario e independientes respecto a la programación.

Como consecuencia del uso de Arduino [3] para otro proyecto en el cual nos encontrábamos trabajando en ese momento, comenzamos a plantearnos la posibilidad de utilizarlo para desarrollar el proyecto que nos ocupa. Uno de los motivos más importantes

para elegir Arduino es la comunidad virtual de usuarios y programadores que se encuentra de fondo. En ella se exponen ideas, proyectos y dudas, y se aportan soluciones a problemas comunes que pueden surgir entre los programadores de esta plataforma. Son muchos los usuarios que publican sus proyectos completos con toda la información incluida, lo cual puede dar lugar a que el alumno aprenda mientras “juega” con toda la información y proyectos previamente desarrollados. Además cuenta en su página web con multitud de manuales acerca de hardware y software que pueden servir de apoyo al estudiante para realizar las prácticas e incluso para generarle curiosidad para seguir indagando por su cuenta.

2 ARQUITECTURA DEL SISTEMA

2.1 MAQUETA

Cómo podemos observar en la Figura 1, tal y cómo hemos indicado previamente, la arquitectura de nuestro sistema está basada en Arduino, una plataforma electrónica de libre distribución que cuenta con entradas y salidas digitales y analógicas, cuyas principales características son la facilidad de programación y su flexibilidad. Arduino está capacitado para sentir el entorno y ejecutar consecuentemente acciones, como por ejemplo, activación de motores, luces o cualquier tipo de actuadores. A esto hay que sumarle su bajo precio y que se trata de un producto de libre distribución, que puede utilizarse libremente para la realización de cualquier proyecto sin ningún tipo de licencia, lo que resulta fundamental a la hora de crear un laboratorio virtual.

La placa que lo compone se encuentra gobernada por un simple micro-controlador Atmega168[4] con arquitectura RISC[17] de 8-bit y 16 KBytes de memoria Flash programable. Cuenta con 14 I/O digitales de las cuales 6 pueden generar modulación de pulso y 6 entradas analógicas con conversor analógico-digital de 10-bit que generan 1024 valores diferentes, lo que nos permite a través de sensores adquirir información del entorno de cualquier tipo y transmitirla a los dispositivos de salida que se desee.

2.2 SERVIDOR

Puesto que se pretendía reducir toda la plataforma para ocupar el mínimo espacio posible, se ha diseñado un cubo de unos 600x800x300 mm realizado con perfilera de aluminio y metacrilato para albergar en él el mayor número de módulos y plataformas, pudiendo duplicar el número de maquetas con el mismo espacio. Se ha empleado como servidor una placa base Mini-ITX con un

procesador Intel[9] Atom a una frecuencia de 533 MHz.

Todas las maquetas se comunican con el servidor mediante puerto USB, a través del protocolo serie RS232[8]. En el servidor se encuentra instalado el sistema operativo Debian [6] GNU/Linux y un servidor web Apache2[1], para el cual se ha desarrollado una aplicación para la subida y administración de código fuente por parte de los alumnos.

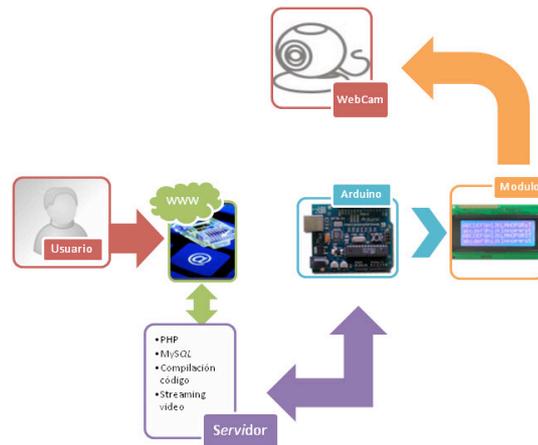


Figura 1: Esquema del funcionamiento de la plataforma

Las tareas de dicho servidor consisten en la muestra de los resultados de la compilación de la práctica correspondiente y el visionado en tiempo real del funcionamiento de la misma. Dicho visionado se puede llevar a cabo gracias a Motion[13], un software encargado de realizar streaming de video en tiempo real desde una Webcam, y al mismo tiempo, de almacenar en pequeños archivos el visionado de la ejecución de la práctica respecto de los distintos módulos de la maqueta por parte del alumno. Dicho software es capaz de detectar los píxeles de la escena y, en este caso que nos ocupa, de los módulos que han variado.

Al mismo tiempo, hemos programado una serie de scripts para almacenar y asociar cada video generado por dicho software al alumno que se encuentra en ese momento autenticado en el sistema

2.3 ARQUITECTURA DE RED

Cómo explicábamos anteriormente, la comunicación del servidor con las maquetas se realiza mediante protocolo serie RS-232 a través de cableado USB. El servidor se encuentra conectado a Internet mediante red Fast Ethernet.

2.4 DISEÑO HARDWARE DE LOS MÓDULOS

Partiendo del concepto de hardware de libre distribución, Arduino nos permite la adición de cualquier tipo de hardware. En nuestro proyecto hemos desarrollado diversos módulos interactivos que permitan la ejecución del software enviado por los alumnos de tal manera que les ayude a comprender su ejecución de una manera visual. Estos módulos son intercambiables por otros, fácilmente modificables y adaptables a los requerimientos de los conocimientos a aplicar.

ARDUINO	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MODULE 1	LED0 (4)	RS (6)	EN (6)	LCD(14)	LCD(13)	LCD(12)	LCD(11)	LED1	LED2	LED3	LED4	CHNN	TX	RX	
MODULE 1	Green led	LIQUID CRISTAL DISPLAY CONNECTIONS							Red led	Yellow led	Green led	Channel selection bit	Read/Write pins		
MODULE 0	DP(7)	A (2)	B(15)	C (13)	D (11)	E (5)	F (5)	G (14)	M(-)	M(-)	LED5	CHNN	TX	RX	
MODULE 0	Green led	SEVEN SEGMENT DISPLAY CONNECTIONS							DC MOTOR CONNECTIONS	Red led	Channel selection bit	Read/Write pins			

Figura 2: Esquema de entradas/salidas del módulo 0

Los módulos desarrollados consisten en elementos hardware de uso habitual que proporcionan la información al alumno a través de elementos visuales. Esta información es capturada mediante cámaras web y transmitida por la red mediante streaming de vídeo a una página web diseñada para tal propósito.

El proyecto inicial se divide en dos módulos de trabajo:

- Módulo 1: En la Figura 3 podemos ver que dicho módulo contiene una pantalla alfanumérica de cristal líquido (LCD) de 20x4 y 6 diodos LED de diferentes colores.
- Módulo 2: Contiene un display 7 segmentos, diodos LED, un motor DC con reductora y contador de vueltas, que podemos ver en la Figura 4, y un sensor de iluminación (LDR).

Estos dos módulos se conectan a Arduino a través de una placa de circuito impreso que se ha diseñado e implementado para el proyecto que nos ocupa, la cual se encarga de demultiplexar las I/O digitales de Arduino. A este circuito lo nombraremos de aquí en adelante Modulo 0, y podemos ver un esquema de sus entradas y salidas utilizadas en la Figura 2. Como explicábamos, Arduino incorpora 14 I/O digitales, suficientes para la mayoría de aplicaciones, pero gracias a este circuito conseguimos duplicar este número, consiguiendo un total de 28 I/O digitales.

De esta forma, cada uno de los módulos 1 y 2 estaría conectado en paralelo a Arduino, y mediante una I/O digital (1-bit) disponible seleccionaríamos a cuál de los dos módulos (1 o 2) dirigimos los datos. Este bit de selección de modulo es programado por software, y en cualquier momento del programa se puede dar la instrucción de cambio de modulo (1 o 2) según con

que parte del hardware queramos trabajar. Por ejemplo, podríamos activar el motor que se encuentra en el módulo 2, cambiar al módulo 1 y escribir en la LCD que el motor ha sido activado. Para cambiar de módulo se ha implementado una función en una librería que cambia de nivel lógico la I/O digital que conecta a la patilla de selección de canal del integrado del modulo 0.



Figura 3: Ejemplo de funcionamiento del módulo 1

Las conexiones entre el modulo 0 y los módulos 1 y 2 se realizan mediante zócalos de cable plano de 20 pines. Adicional y previamente a la conexión de los módulos 1 y 2 se encuentra un poste de conexiones hembra donde se reflejan duplicadas todas las salidas de Arduino. Este poste contiene las I/O digitales de Arduino, junto con la alimentación 5V y GND mas 3 entradas analógicas. Esto ofrece un acceso instantáneo a los valores digitales de estas, siendo esto de gran ayuda para operaciones de depuración de errores de hardware y posible ampliación de módulos.

Los módulos que hemos desarrollado permiten desarrollar multitud de funciones que ayudan al alumno al aprendizaje del lenguaje en diversas circunstancias. El eje central de los módulos es la LCD, pues será el principal interfaz de comunicación. En ella podremos depurar errores, mostrar datos y variables del programa, el manejo de estructuras y trabajar con matrices. Además, podremos mostrar en tiempo real los datos que le proporcionan a Arduino los demás componentes. Por ejemplo, podemos mostrar el nivel de iluminación recogido en el sensor LDR, el número de vueltas que ha dado el motor y en qué sentido gira, el estado de los diodos LED, etc.

Se proporcionan las funciones para inicializar la LCD, escribir en ella y borrarla, situar el cursor en

determinado lugar y desplazar caracteres horizontalmente.

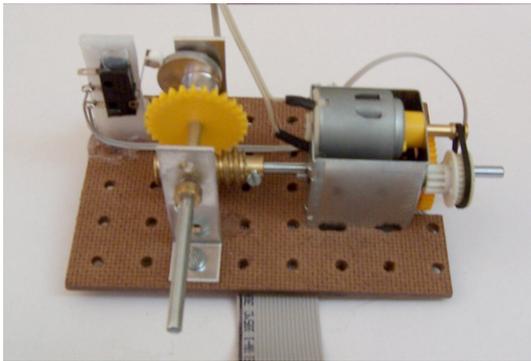


Figura 4: Imagen de la implementación del módulo 2

Por otro lado, el motor DC nos puede servir para programar tareas en paralelo y trabajar con temporizadores. Podemos pedir que se active el motor durante un determinado número de vueltas o un determinado tiempo y a la vez se muestre mediante los diodos Led el sentido de giro, se active un semáforo o se muestre el número de vueltas que restan por dar en el display siete segmentos. Por otra parte, será posible girar el motor en ambos sentidos. Los diodos Led son útiles a la hora de depurar errores en los programas e incluso pueden servir al alumno como indicadores de cómo ha avanzado su programa en tiempo de ejecución. Por ejemplo, podríamos encender el LED1 cuando salgamos de un bucle o estemos en un case (switch ... case).

La LDR es útil para comprender como leer un valor. Una vez leído, este puede ser pasado a una función o realizar con él alguna operación matemática. El display de siete segmentos funciona como si de un Led se tratara, pero para su correcto funcionamiento, será necesario activar determinados pines a la vez. Resulta también útil para comprobar por qué punto del programa nos encontramos o para mostrar numéricamente cualquier valor del programa, por ejemplo, en un bucle for, la variable contador de dicho bucle.

2.5 APLICACIÓN WEB

La aplicación web se encuentra realizada en PHP[15] apoyándose en una base de datos bajo MySQL[14], con el cometido de almacenar logs que nos permitan llevar un registro de cuantas veces un alumno ha subido y compilado una práctica, cuantas veces se ha compilado y ejecutado la misma satisfactoriamente, o alguna otra estadística individualizada que nos resulte de utilidad a la hora de evaluar al alumno.

2.5.1 Compilador online

Gracias a la capacidad de compilación a través de línea de comandos que presentan los microcontroladores Atmel en GNU/Linux, mediante su compilador `avr-gcc`[5], se ha desarrollado un módulo web en PHP para la muestra por pantalla en tiempo real, de los resultados de la compilación de aplicaciones desarrolladas para Arduino.

En este módulo, se presentan los errores que puedan devengar de una programación incorrecta. Si estos existen, se nos presentará la opción en el sistema de volver a subir el código fuente, mientras que la opción de cargar el fichero hexadecimal en el micro se encontrará desactivada, evidentemente, porque no se habrá generado ningún archivo hexadecimal como resultado de dicha compilación. Si por el contrario no se encuentran errores en la compilación, se activará, además de la opción de cargar el fichero hexadecimal en el micro, de nuevo la de volver a subir el fuente por si el programa no realiza lo solicitado en el guión de la práctica aunque éste no presentara errores de compilación.

Existe la posibilidad de que los alumnos utilicen el entorno IDE disponible para Arduino en su casa, aunque el motivo de la realización de este módulo se debe a que es posible mostrar algo más de información de lo que hace dicho IDE, incluso pudiendo resaltarla dependiendo de los tipos de errores que presente. Esta información la almacenaremos en nuestro servidor y podremos consultarla para comprobar la evolución del alumno durante el curso y ver en que aspectos ha mejorado. Más adelante, se desarrollará un nuevo módulo capaz de traducir la información acerca de los errores que devuelve `avr-gcc` con el fin de realizar una traducción que sea más sencilla para nuestros alumnos que se encuentran comenzando a programar.

2.5.2 Autenticación

Para las conexiones por parte de los alumnos, se había valorado la posibilidad de realizar una cola en C, aunque finalmente se realizó un control de acceso de usuarios mediante sesiones en PHP, de tal forma que se realizan las validaciones de los alumnos mediante un usuario y una contraseña asignados previamente por el profesor. A continuación se genera una sesión límite de dos horas para poder subir la práctica al servidor, compilarla mediante el compilador online y en caso de no obtener ningún error, grabar el micro ATMEGA para poder visualizar el resultado de la práctica. El alumno podrá usar el compilador online tantas veces como desee, aunque existirá un intervalo de 24 horas entre comprobación y comprobación del funcionamiento de la programación realizada en tiempo real en la maqueta, dejando paso al resto de alumnos para el acceso a la

plataforma. La página de acceso a la aplicación podemos visualizarla en la Figura 5.

2.6 COMPILADOR OFFLINE Y ENTORNO IDE

Los creadores de Arduino también proporcionan un IDE gratuito de libre licencia. Este entorno de desarrollo y lenguaje de programación está basado en los proyectos Processing[16]/Wiring[18].

El IDE creado para Arduino está disponible para múltiples plataformas (Microsoft Windows[12], GNU/Linux, MacOS[2]). En él creamos los programas con un estilo de programación prácticamente idéntico a C/C++. Todas las sentencias que podemos compilar son las soportadas por el compilador avr-gcc. Además, Arduino también puede ser conectado a la computadora y es capaz de interactuar con diferente software como Matlab[11].

2.7 CREACIÓN DE LIBRERÍAS

Para que nuestro proyecto fuera transparente, electrónicamente hablando, para el alumno, hemos desarrollado un grupo de funciones para una aplicación y las hemos insertado en el proyecto como librerías. Estas librerías se le facilitarán al alumno para el manejo de módulos hardware que conectemos a Arduino. Por ejemplo, para realizar funciones de inicialización, borrado o escritura de una LCD alfanumérica.

Fundamentalmente, hemos implementado esa serie de librerías debido a que nuestro sistema se compone de distintos dispositivos hardware cuyo manejo consta de una serie de operaciones a bajo nivel que podrían interferir en el propósito que se desea para el alumno, que no es otro que aprenda a programar en C/C++. Siempre hemos procurado evitar las complicaciones de la comprensión del funcionamiento interno de los dispositivos, objetivo de otras asignaturas de la titulación.

2.8 RESULTADOS ALCANZADOS

Después de la puesta en marcha de este proyecto, se ha podido comprobar que el sistema aporta una serie de ventajas para el alumnado y para el profesorado, y aunque en todo momento hemos procurado cuidar la transparencia o modularidad del proyecto, también aporta una serie de desventajas.

2.8.1 Ventajas

La primera ventaja que encontramos es la transparencia del sistema electrónico de cara al alumno, es decir, el alumno sabrá cómo direccionar datos hacia una LCD pero no será necesario el conocimiento de cómo se encuentra conectada ni el

funcionamiento de las librerías para la comunicación mediante la misma.

Por otro lado, simplicidad a la hora de reparar posibles fallos de hardware de los módulos. Debido a la simplicidad que presentan los módulos electrónicamente hablando, y teniendo en cuenta que los módulos son independientes entre sí, aunque un módulo no se encuentre conectado, la maqueta podrá funcionar normalmente y por tanto será posible solucionar las averías simplemente mediante la sustitución de un módulo por otro nuevo.

Otra gran ventaja, es que al ser un sistema modularizable, éste presenta una gran flexibilidad a la hora de añadir nuevas funcionalidades a las maquetas mediante la inserción de nuevos módulos, pudiendo incluso aumentar el número de prácticas ofertadas al alumnado de una forma rápida y sin modificación del software.

Además, este proyecto plantea una reducción de la necesidad de infraestructura de laboratorios, pudiéndose realizar todas las prácticas online, adaptándose así al inminente cambio que plantean los nuevos créditos ECTS [7], dándole al alumno la posibilidad de trabajar desde casa en sus prácticas de laboratorio.



Figura 5: Servidor web de almacenamiento de prácticas y autenticación de alumnos

Este proyecto también plantea una pequeña introducción a los microcontroladores y a su programación mediante lenguajes de alto nivel, provocando que el alumnado no choque de frente una vez más con asignaturas electrónicas donde se empleen componentes de este tipo, como ya pasaba cuando el alumno en primer curso tenía que enfrentarse a asignaturas de programación sin ningún tipo de fundamentos.

En cuanto a ventajas de cara al profesorado, los métodos de corrección de prácticas se convierten en algo bastante visual a través de los vídeos que se almacenan en el servidor con los resultados de las prácticas realizadas y compiladas por alumno. El profesor será capaz de comprobar si la práctica funciona mediante la visualización del funcionamiento de los dispositivos, sin necesidad de realizar una comprobación del código, aunque evidentemente es altamente recomendable comparar los resultados con el mismo.

Otra ventaja es la abstracción de la parte más compleja del hardware como son las conexiones entre las diferentes partes. Esto resulta importante para usuarios que no hayan desarrollado su propia aplicación y no tengan esta información o que no estén familiarizados con la electrónica. En nuestro caso esto beneficia al alumno, que no deberá adquirir este tipo de conocimiento, centrándose más en la programación.

Por último, la protección de la parte electrónica del hardware respecto a usuarios malintencionados o de errores de programación que permitirían manejar las conexiones directamente, pudiendo provocar errores o incluso daño para los componentes.

2.8.2 Desventajas

La única desventaja importante que se nos plantea con este sistema es la necesidad de introducir al alumnado una clase práctica para mostrarle el funcionamiento del sistema: la autenticación en el mismo, el compilador online, la muestra de resultados y la captura de vídeo, pero al inicio de curso se les realiza una práctica de ejemplo, no puntuable, en lo que resulta ser un simulacro perfecto de entrega de prácticas, donde su procedimiento queda asimilado notablemente.

2.9 TRABAJOS FUTUROS

Como trabajo futuro nos hemos planteado conseguir introducir este proyecto dentro de la plataforma Moodle, en la cual nos estamos apoyando para impartir las asignaturas indicadas en la introducción del presente documento. Para ello, se pretende desarrollar un módulo que interactúe con las últimas versiones de la plataforma y ponerlo a disposición de las universidades que deseen hacer uso de él.

Referencias

- [1] Apache Software Foundation, <http://www.apache.org>
- [2] Apple Inc., <http://www.apple.com>
- [3] Arduino home page, <http://www.arduino.cc>
- [4] Atmel Microcontrollers, <http://www.atmel.com>

- [5] C runtime library for the AVR family of microcontrollers, <http://savannah.nongnu.org/projects/avr-libc/>
- [6] Debian GNU/Linux Distribution, Home Page, <http://www.debian.org>
- [7] ECTS, European Credits Transfer System, RD1125/2003 5th September “Establishment of European credits and qualifications system at Spanish university degrees”
- [8] Electronics Industries Association, <http://www.eia.org>
- [9] Intel Corporation, <http://www.intel.com>
- [10] LDR, Light Dependent Resistor, http://www.doctrionics.co.uk/ldr_sensors.htm
- [11] Matlab Software, The Mathworks, <http://www.mathworks.com>
- [12] Microsoft Corporation, <http://www.microsoft.com>
- [13] Motion detection software, <http://www.lavrsen.dk/twiki/bin/view/Motion/W ebHome>
- [14] MySQL open source database, <http://www.mysql.com>
- [15] PHP web scripting software, <http://www.php.net>
- [16] Processing open source programming language, <http://processing.org>
- [17] Reduced Instruction Set Computer Architecture, http://en.wikipedia.org/wiki/Reduced_instruction_set_computer
- [18] Wiring open source programming environment, <http://www.wiring.org.co>