

Experimentación remota basada en sistemas Arduino

Sergio Martín, Atilano Fernández-Pacheco, José A. Ruipérez-Valiente, German Carro and Manuel Castro

Title—Remote experimentation through Arduino-based remote Laboratories

Abstract— This paper describes the architecture of an Arduino remote lab that supports the deployment of many Arduino-based experiments, such as a sensors remote lab consisting on eleven sensors and a LCD display connected to an Arduino MEGA, or a 3D RGB LED cube remote lab consisting on 16 RGB LEDs connected to an Arduino UNO-compatible board. The proposed on-line system allows students to write code on a website to be executed on these experiments. The execution results can be observed in real time through an IP camera. The use of this kind of on-line didactic tools is very important to provide high quality on-line education programs on technical fields.

Index Terms— Remote lab, Arduino, Sensors, Raspberry Pi

I. INTRODUCCIÓN

Uno de los mayores retos al que se enfrenta la educación *on-line* es como facilitar las actividades de experimentación en el campo técnico. Las prácticas de laboratorio son esenciales en STEM (Ciencias, Ingeniería, Tecnología y Matemáticas - *Science, Engineering, Technology and Mathematics* por sus siglas in inglés) disciplinas que ayudan a los alumnos asimilar correctamente los conceptos teóricos y facilitar que puedan adquirir las habilidades prácticas imprescindibles para su carrera profesional una vez que finalicen sus estudios universitarios.

Tradicionalmente, las sesiones de prácticas han sido impartidas en las instalaciones de la universidad, donde los estudiantes pueden físicamente experimentar con los equipamientos físicos en cierto horario. Estas sesiones de laboratorio son vitales para el desarrollo de los estudiantes y la universidad debe asegurar una cantidad, variedad y calidad de este tipo de prácticas.

Tradicionalmente, las prácticas de laboratorios se han venido realizando en las instalaciones de las universidades (clases y otros espacios especialmente equipados) con unos horarios previamente fijados que no dan al alumno mucha flexibilidad. Estas prácticas de laboratorio normalmente tienen lugar en pequeños grupos para facilitar que los profesores puedan proporcionar una atención personalizada y asegurar un uso seguro y adecuado del equipamiento [1][2].

De acuerdo con muchos autores [3] [4] [5] [7], una de las potenciales soluciones a esos problemas es el uso de laboratorios remotos y virtuales [8] [9]. El uso de estos laboratorios *on-line* en los estudios de ingeniería están aumentando en relevancia todos los días debido a la

necesidad de proveer flexibilidad, escalabilidad y acceso de alta calidad para adquirir competencias profesionales con instrumentación real en cualquier momento y desde cualquier lugar sin necesidad de estar físicamente en clase.

Este artículo es una extensión del artículo publicado por los autores [11] como parte de la conferencia EDUCON 2019. Este nuevo artículo incluye una presentación rehecha de los trabajos relacionados con Arduino en educación, y extiende considerablemente la sección de experimentación donde se describe en detalle el laboratorio remoto del Cubo LED 3D y un laboratorio de sensores bajo la arquitectura propuesta. Tanto la sección del resumen, introducción y conclusiones han sido extendidas o modificadas.

En este caso de estudio, el laboratorio remoto propuesto será usado por estudiantes de Ingeniería Electrónica de una Universidad de educación a distancia como parte de las prácticas de laboratorio obligatorias de la asignatura "Microprocesadores y Microcontroladores". Queremos señalar que este trabajo compartirá pautas y lecciones aprendidas que serán de interés para una audiencia amplia de estudiantes, docentes y profesionales interesados en el aprendizaje remoto y los laboratorios de tecnología.

Los principales objetivos de este artículo son:

- Describir la arquitectura de un laboratorio remoto de Arduino basado en Raspberry Pi.
- Describir la configuración del laboratorio remoto para soportar varios experimentos, tales como, Cubo LED 3D y el Laboratorio de Sensores.

II. TRABAJO RELACIONADO

A. *Uso de Arduino en educación*

Actualmente existen muchas universidades donde Arduino está siendo usado como recurso para el estudio de sistemas embebidos. Los autores en [12] comenzaron en 2010 a introducir la plataforma Arduino como posibilidad para que los alumnos realizasen los proyectos en clase. Acorde a los autores, existía un alto número de estudiantes que elegían Arduino como plataforma de desarrollo para implementar sus proyectos en lugar de FPGAs o microcontroladores PIC. Además, tanto la variedad de proyectos como su complejidad crecieron en comparación con los años anteriores a la inclusión de Arduino. Entre los proyectos que se llevaron a cabo en esa universidad con Arduino se encuentra un analizador de espectro o una pistola de paintball que dispara de forma autónoma, donde el Arduino UNO controla los servos [13].

Otro ejemplo de usar Arduino como parte curricular de las clases se encuentra en la Universidad Tecnológica de

Bratislava [14]. En esta universidad, el producto comercial Boe-Bot fue usado para el desarrollo de robots. Sin embargo, los estudiantes informaron que el uso de BASIC como lenguaje de desarrollo en sus proyectos era una desventaja. Por lo tanto, los coordinadores de la asignatura decidieron adaptar el curriculum para usar la placa Arduino en lugar de las placas hardware que se estaban utilizando. El resultado fue muy satisfactorio. Los coordinadores informaron que con pocas lecciones los estudiantes ya podían programar un robot que fuera capaz de seguir una línea evitando diferentes obstáculos.

Basado en la literatura reciente que se ha revisado, y por nuestra propia experiencia, los mayores beneficios de usar Arduino dentro del plan de estudios son:

- Fácil configuración. Una vez que se ha instalado el entorno de Arduino, es muy directo poder utilizar el entorno de programación.
- Existen muchos ejemplos y muchos manuales muy accesibles que pueden servir de guía para controlar periféricos.
- Software libre y fácilmente accesible.
- Una comunidad Arduino muy activa y colaborativa que comparte código de los proyectos. Esto es una gran ventaja para los estudiantes, ya que, es fácil acceder a ejemplos para resolver problemas o dudas que surjan en sus desarrollos.
- El coste de mantenimiento. En caso de daño en algún componente, el coste de reemplazo es bajo. Hay microcontroladores que pueden ser adquiridos con *bootloader* o sin él. El segundo puede ser usado en el caso de tener un microcontrolador programado.
- Los estudiantes pueden construir prototipos de forma rápida. Esta posibilidad de obtener resultados tangibles en el corto plazo, incrementa la motivación de los estudiantes. Por lo tanto, mejora el proceso de aprendizaje.

Por lo tanto, es importante que las instituciones educativas puedan tener acceso a laboratorios Arduino de alta calidad donde los estudiantes puedan aprender habilidades relacionadas con esta tecnología con el objetivo de mejorar las competencias técnicas de los estudiantes y sus perspectivas laborales futuras.

B. Laboratorios remotos.

Estos laboratorios a menudo no están vinculados a un área específica. En cambio, suelen cubrir amplias áreas de conocimiento y aplicaciones (ej. programación, sistemas de control, etc.) y, por lo tanto, los laboratorios se pueden utilizar en una variedad de asignaturas. El objetivo general de estos laboratorios es habilitar la posibilidad de que los estudiantes pongan en práctica los conocimientos teóricos que han adquirido a través de las clases magistrales. Posteriormente, mediante el uso de una interfaz virtual permite al estudiante interactuar con el laboratorio remoto. Durante este proceso, los estudiantes son capaces de realizar prácticas de experimentación con un escenario real, facilitando la transición de la teoría a la práctica y consolidando el conocimiento teórico aprendido.

Estos laboratorios definen diferentes roles y permisos de acceso, que a menudo se organizan en dos niveles de seguridad: el primero especifica qué usuarios pueden

acceder al laboratorio y el segundo especifica las acciones que cada usuario puede realizar dentro de la aplicación. Además, todos los laboratorios deben introducir algún tipo de control de seguridad y autorregulaciones que tenga como objetivo minimizar los errores que puedan causar daños a los equipos y dispositivos. En el caso del software que se utiliza para interactuar con el laboratorio remoto, el sistema debe controlar los posibles escenarios que se puedan producir, para que no se ejecuten acciones inadecuadas.

La presencia de este tipo de laboratorios remotos ha aumentado considerablemente en la universidad española, por ejemplo, en la UPV / EHU, donde los estudiantes tienen a su disposición un buen número de estos laboratorios. Esto se ha debido a que los laboratorios remotos tienen muchas ventajas [15]:

- No hay restricciones de tiempo y ofrecen un alto grado de flexibilidad para el estudiante.
- Es posible incluir controles sencillos y retroalimentación para evitar un uso inadecuado que podría ser potencialmente dañino para los equipos.
- Puede aumentar la cantidad máxima de estudiantes que pueden usar estos laboratorios, ya que no hay restricciones de espacio físico. Este factor tiene una limitación, que es el número de usuarios concurrentes.
- Puede aumentar el número de estudiantes que pueden utilizar este equipo para reducir el coste por estudiante y así aumentar la calidad en otras áreas donde se requiere mayor inversión.

Según trabajos previos [6], los principales requisitos que debe cumplir un laboratorio remoto son los siguientes:

- Alta disponibilidad. Una característica clave es la posibilidad de ser utilizado sin restricción de tiempo y en un escenario de alta disponibilidad. Es fundamental que esta característica se tenga muy en cuenta como parte del diseño y la arquitectura del laboratorio.
- Simultaneidad. La arquitectura del laboratorio debe adaptarse al escenario de caso habitual en el que varios usuarios interactúan con el laboratorio simultáneamente; todos los usuarios deberían poder trabajar concurrentemente. De lo contrario, habrá una cola de solicitudes y el sistema tardaría en su procesamiento.
- Bajo coste operativo. Dado que se utilizarán con alta frecuencia, el mantenimiento de estos laboratorios debe tener un bajo coste operativo y financiero.
- El laboratorio remoto debe contar con un sistema de control de autorregulación para evitar un uso inadecuado. En otras palabras, el sistema debe protegerse de ser dañado por las ejecuciones de los estudiantes.

C. Laboratorios remotos de Arduino.

Carro [16] desarrolló un laboratorio remoto para que los estudiantes de Primaria controlasen el color de una luz LED conectada a un Arduino. Este laboratorio no permitía programar el Arduino en sí, sólo proporcionaba una interfaz web para seleccionar la mezcla de colores que se mostrarán en el LED. Por lo tanto, este laboratorio no fue diseñado para enseñar programación Arduino, sino para

facilitar una explicación sobre los componentes de la luz.

Sarik y Kymissis [17] presentan un laboratorio distribuido (o laboratorio para llevar a casa) para la programación de Arduino. Consistía en proporcionar un kit Arduino a los estudiantes que pudieran llevarse a casa. La principal ventaja de estos laboratorios es que se pueden montar en casa es que los estudiantes pueden trabajar con el hardware real conectado a sus ordenadores. Estas configuraciones también tienen algunas limitaciones que detallamos a continuación:

- Este tipo de laboratorios para llevar a casa solo son viables para diseños electrónicos que no son muy complejos, ya que los estudiantes no tienen acceso a los equipos específicos necesarios para preparar los diseños. Sin embargo, una posibilidad factible es montar estos complejos laboratorios solo una vez y compartirlos con tantos estudiantes como sea necesario.
- Los laboratorios para llevar a casa tampoco son viables cuando el experimento propuesto incluye componentes con un alto coste, ya que implicaría comprar tantos componentes con un coste elevado como estudiantes se tuvieran en el laboratorio. Esto multiplica el coste del laboratorio. Sin embargo, cuando se trabaja con laboratorios remotos solo se necesita un kit, por lo que disminuye claramente su coste final.
- El mantenimiento se vuelve más complejo ya que el transporte de delicados diseños electrónicos de la escuela a la casa y viceversa, a menudo puede incurrir involuntariamente en romper o desconectar algunas piezas. Sin embargo, cuando trabajamos con laboratorios remotos, no existe ningún riesgo de transporte para el equipo.
- Los laboratorios remotos permiten reutilizar la misma práctica de laboratorio para varias clases al mismo tiempo.
- Cuando un estudiante usa un laboratorio para llevar a casa, las instrucciones programadas por el estudiante podrían dañar el laboratorio si existe un error en el código que pueda comprometer al equipo. Por ejemplo, al programar el movimiento de un brazo robótico, el estudiante puede dañar uno de los motores de articulación si el estudiante no controla el límite de torsión de los brazos. Los laboratorios remotos permiten incluir una capa de software adicional que puede controlar estas acciones peligrosas.

Otros autores, como Mejías Borrero et al. [18], desarrolló un *framework* para convertir un experimento de laboratorio en un laboratorio remoto. Este *framework* permite interactuar con pines de entrada/salida de Arduino y configurar una cámara web. Sin embargo, este *framework* no permite que el estudiante escriba el código Arduino para ejecutarlo.

Un problema similar se encuentra en el trabajo de Kalúz et al., [19] así como en Cardoso y Sousa [20]. Sus laboratorios utilizan Raspberry Pi y Arduino como base para crear laboratorios remotos. Sin embargo, los experimentos no se centran en permitir que el alumno escriba el código a ejecutar en la placa Arduino. La placa Arduino está preprogramada para gestionar algún

experimento específico (planta hidráulica o un modelo de levitación magnética) permitiendo al alumno modificar algunos de los parámetros aplicados al experimento, pero no el código en sí. Así, el objetivo de este trabajo no es permitir que los estudiantes aprendan y mejoren sus habilidades de programación Arduino, sino que experimenten con la configuración de diferentes parámetros a ejecutar en un laboratorio remoto.

III. ARQUITECTURA

En este apartado se explican los diferentes elementos arquitectónicos. El dispositivo principal es una placa Arduino que se conecta por un lado a otros dispositivos como el cubo LED, grúa mecánica, sensores, etc., y por el otro lado a una Raspberry Pi que actúa como servidor web.

La conexión entre la Raspberry Pi y la placa Arduino se realiza a través del puerto USB. La cantidad de placas Arduino que se pueden conectar al servidor Raspberry Pi está limitada debido a la cantidad de puertos USB disponibles.

El administrador debe configurar el laboratorio remoto indicando las diferentes placas Arduino conectadas. Para realizar esta tarea, el administrador debe agregar una línea por cada nuevo Arduino conectado. La línea de formato es la siguiente:

"Port, name device, microprocessor, programmer, baud, default program"

La descripción de cada uno de estos parámetros es la siguiente:

- *Port*: es el puerto del dispositivo, que será utilizado por la placa Arduino.
- *Name device*: es el nombre del dispositivo que los usuarios visualizarán en la web.
- *Microprocessor*: este parámetro indica el tipo de procesador Arduino, por ejemplo, atmega328p, atmega2560, etc.
- *Programmer*: este parámetro indica el tipo de programador que se utilizará en la conexión, por ejemplo, Arduino o *wiring -D*.
- *Baud*: es necesario definir la velocidad de conexión. Este parámetro es muy importante porque una incorrecta definición puede provocar errores cuando se carga el programa en el dispositivo.
- *Default program*: este parámetro indica el programa predeterminado de reinicio que se ejecutará cuando el tiempo del usuario haya expirado. Este es un fichero binario (.hex) que realiza todas las tareas necesarias en el dispositivo para reiniciarlo y dejarlo en un estado estable para el próximo usuario. Este programa configura el dispositivo con sus configuraciones predeterminadas.

Algunos ejemplos de líneas de configuración predeterminadas para diferentes dispositivos, como un cubo LED 3D basado en un microprocesador ATmega328p y un laboratorio Arduino MEGA basado en ATmega2560, son los siguientes:

```
/dev/ttyUSB0,Cubo Arduino,atmega328p,arduino,57600,ini_cubo.hex
/dev/ttyACM0,Sensores,atmega2560,wiring -D,115200,ini_sensores.hex
```

Aunque el sistema puede admitir muchos usuarios al

mismo tiempo, solo uno puede ejecutar instrucciones de forma concurrente. En consecuencia, es necesario introducir mecanismos de sincronización y cola para orquestar las diferentes ejecuciones. Por lo tanto, cuando se están ejecutando las instrucciones de un usuario, el experimento utilizado en concreto (es decir, laboratorio de sensores, cubo de led) se bloquea para otros usuarios durante diez minutos o hasta que el usuario actual finalice la sesión con el laboratorio. Durante ese tiempo, si otro usuario intenta ejecutar otro programa, el usuario recibirá una notificación que indica que la acción no se puede realizar y cuánto tiempo más permanecerá bloqueado el dispositivo. En este caso, el usuario ingresa a una cola esperando su turno para obtener acceso al laboratorio remoto en función de la hora de llegada, es decir, el primero en llegar primero para obtener acceso.

Por otro lado, la Raspberry Pi es el hardware elegido como servidor de aplicaciones. El fundamento de esta decisión es muy similar al que se indicó anteriormente para la selección de Arduino. Es posible implementar un servidor completo con características de alto rendimiento a un costo reducido. Además, la comunidad de desarrolladores de Raspberry Pi está bien establecida y extendida. Raspberry se basa en el sistema Linux y, por lo tanto, es muy fácil de implementar como servidor web.

Los pasos principales realizados por el servidor web para ejecutar el código fuente de los estudiantes en la placa Arduino son muy similares a los que realiza el IDE de Arduino al compilar y ejecutar un código fuente en una placa conectada. La principal diferencia es que un script makefile hace todos los pasos en lugar de ser ejecutado por el propio IDE de Arduino. Estos son los principales pasos involucrados:

1. Preparar los parámetros: Un script makefile prepara todos los parámetros de compilación teniendo en cuenta microcontrolador, puerto, baudios, etc.
2. Compilación: El script makefile invoca al compilador Arduino para convertir el código fuente ".c" en un objeto ".o".
3. Vinculación: El módulo *linker* busca las bibliotecas utilizadas en nuestro código fuente y las vincula a nuestro código compilado para generar un archivo binario que podemos cargar en la placa Arduino.
4. Carga y ejecución: Una vez creado el archivo binario, lo cargamos a la placa Arduino para que sea ejecutado.

Otra característica interesante es la posibilidad de migrar con unos sencillos pasos a un servidor Linux tradicional. Por lo tanto, Raspberry Pi es una gran opción para implementar un sensor web [10]. Además, la arquitectura (Figura 1) agrega una Webcam para que el alumno pueda observar visualmente las acciones físicas que sus instrucciones están provocando en el equipo.

Este último punto es muy importante porque permite al alumno conectar las instrucciones virtuales con el equipo físico mediante la visualización de las pruebas, lo que también ayuda a comprender y resolver problemas con mayor facilidad. Además, ayuda al alumno a sentirse más cercano al laboratorio y a evitar la sensación de utilizar solo un entorno virtual en lugar de instrumentación real.

En cuanto a la experiencia del usuario, el tiempo

necesario para compilar el código fuente escrito por el alumno depende de la longitud del código y de las bibliotecas utilizadas, pero la mayoría de los programas que tienen que compilar nuestros alumnos tardan unos tres segundos en compilarse. Este tiempo de compilación es el mismo que puede tomar cuando se usa una placa Arduino física conectada a nuestro propio ordenador en lugar de usar un laboratorio remoto.

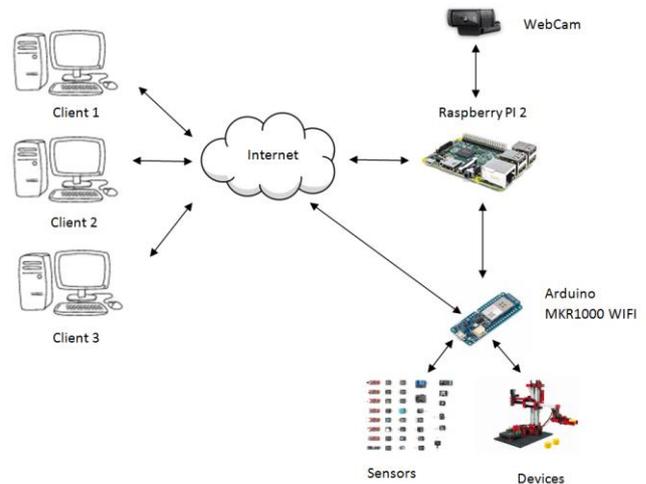


Fig. 1 Arquitectura del laboratorio remoto [3].

Una vez compilado el código, debe cargarse en la placa y que el dispositivo inicie su ejecución. Este proceso tarda aproximadamente otros tres segundos adicionales. De manera análoga, si tenemos una placa Arduino real conectada a nuestro ordenador, el tiempo de ejecución requerido al cargar el código binario en la placa es aproximadamente el mismo. Así, el alumno no tiene ninguna desventaja desde el punto de vista del tiempo de ejecución.

Desde la perspectiva de la latencia, una vez que se ejecuta el código, hay un retardo de 500 ms en la transmisión de la transmisión de la cámara web. Sin embargo, esto no es muy relevante y no afecta la ejecución de la práctica, ya que las prácticas propuestas a los estudiantes no hacen uso de ningún mecanismo de entrada, solo hacen uso de salidas visuales, como pantallas LED o LCD. Por tanto, no hay un retardo real desde el punto de vista del estudiante: la ejecución de la misma, se visualiza sin problemas.

Finalmente, con respecto al soporte de usuarios concurrentes, dentro de la arquitectura actual solo un estudiante puede ejecutar código simultáneamente en cada experimento, el resto de estudiantes tiene que esperar en una cola para obtener acceso.

IV. DESCRIPCIÓN DEL EXPERIMENTO

La cantidad de experimentos que los alumnos pueden realizar con este laboratorio remoto es muy diversa y variada. Todos ellos son experimentos remotos basados en Arduino a los que se puede acceder desde cualquier navegador web (Firefox, Chrome, Opera, etc.) sin necesidad de instalar extensiones ni ningún otro software. Aún más, la arquitectura es multiplataforma y los estudiantes pueden acceder desde los sistemas operativos Windows, Linux o Mac. Por lo tanto, no hay barreras de uso por la plataforma que se use, su hardware o software y no es necesario utilizar un ordenador de alto rendimiento para tener una buena experiencia de usuario. En

esta sección, los experimentos se explican con un mayor detalle.

La interfaz de usuario y las operaciones disponibles dentro de cada experimento son muy intuitivas, por lo que cualquier usuario podría utilizar este programa sin mucha dificultad y con una baja curva de aprendizaje. La página web del laboratorio se divide en dos partes. En el lado izquierdo, hay una sección de código donde los estudiantes pueden escribir su código Arduino (Figura 2).

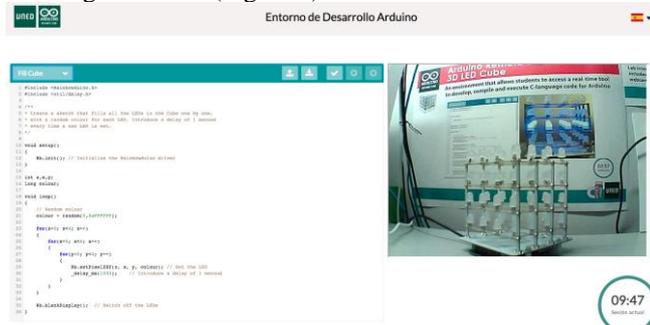


Fig. 2 Detalle del lado izquierdo de la interfaz de usuario del laboratorio remoto.

El estudiante tiene un botón que le permite compilar el código y ejecutarlo en la placa Arduino remota. Esta área de la GUI también incluye un menú desplegable con algunos ejemplos de uso predefinidos para cada experimento, como se puede ver en las Figuras 3 y 4.



Fig. 3 Detalle del menú desplegable con código de ejemplo del laboratorio de sensores totalmente funcional.



Fig.4 Detalle del menú desplegable con código ejemplo del laboratorio del Cubo LED totalmente funcionales.

En el lado derecho, los estudiantes encuentran el vídeo de la cámara web, que les permite ver el resultado de la ejecución del código en los instrumentos remotos como parte del experimento (Figura 5).

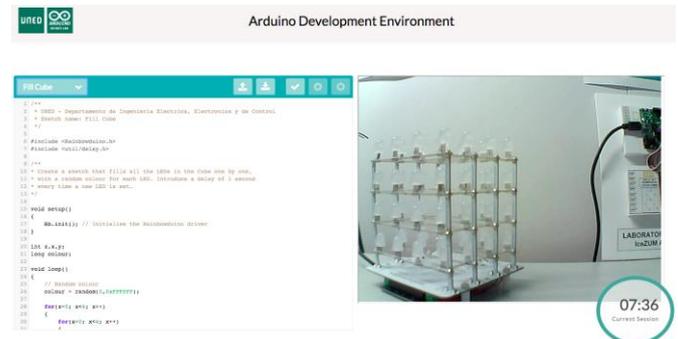


Fig. 5 GUI del laboratorio remoto de Arduino del Cubo Led 3D.

El flujo de trabajo de ejecución tiene únicamente tres pasos: carga de código, compilación y ejecución en la placa. Por lo tanto, el proceso es sencillo y es un enfoque apropiado para presentar a aquellos estudiantes sin experiencia previa a la programación de Arduino sin la necesidad de comprar ningún equipo.

A. Cubo Led 3D

El primer experimento que describimos es un cubo LED 3D. Este experimento consiste en un conjunto de 64 LEDs RGB dispuestos en una estructura de cubo de 4x4 (Figura 6).

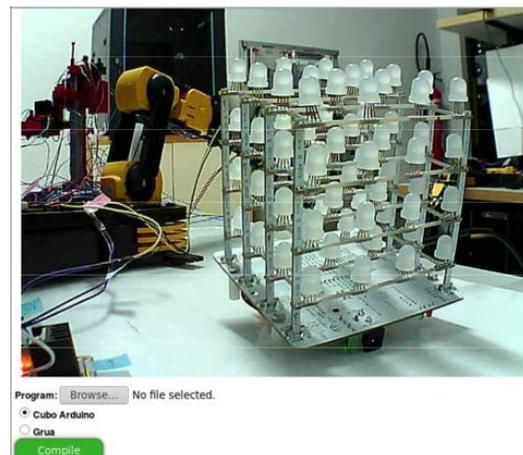


Fig. 6 Cubo LED 3D.

Los estudiantes pueden escribir código Arduino para encender o apagar cada uno de estos LED. También pueden programar el color y la intensidad de cada LED en cualquier momento. Esto les permite crear animaciones coloridas basadas en bucles definidos.

Las competencias que adquieren los estudiantes gracias a este experimento remoto son las siguientes:

- Los estudiantes aprenden a interactuar con los pines Arduino GPIO analógicos. Además, los estudiantes aprenden a definir la intensidad y el color de cada uno de los LED conectados a los pines analógicos GPIO.
- Los estudiantes mejoran las competencias relacionadas con las declaraciones condicionales y de bucle. Tienen que programar 64 LED RGB que están organizados en un plano tridimensional. Esto requiere a los estudiantes trabajar con bucles, ya que no es práctico interactuar con cada LED de forma independiente. Además, también tienen que generar

animaciones de varios segundos con diferentes colores e intensidades, lo cuál hace que el alumno comprenda mejor la forma de gestionar los LEDs en grupos.

- Los estudiantes mejoran su pensamiento espacial. El hecho de que tengan que trabajar en una disposición tridimensional hace que los estudiantes piensen cómo manejar los LED en los planos X, Y, y Z para crear animaciones apropiadas.

Algunas de las prácticas propuestas a los estudiantes en base a este cubo LED 3D son las siguientes:

- Aleatorio: cree un programa que coloree todos los LED del cubo uno por uno, con un color aleatorio para cada LED. Introduzca un retardo de un segundo cada vez que se configure un nuevo LED.
- Países: muestra de forma secuencial las banderas de varios países, cambiándolas cada cuatro segundos.
- Planos: defina una función para cambiar todos los LED en un plano Z dado con un color deseado. Luego defina una función similar para encender todos los LED en un plano Y determinado. Usando ambas funciones, los estudiantes deben desarrollar un programa donde un plano Z azul atraviese todo el cubo secuencialmente, es decir, primero para $z = 0$, luego $z = 1$, $z = 2$ y finalmente $z = 3$. Sólo se puede configurar un plano en cada momento dado. Posteriormente, siguiendo las mismas reglas, los estudiantes deben definir un plano Y rojo que vaya de $y = 0$ a $y = 3$. Finalmente, los estudiantes deben introducir un retraso de dos segundos cada vez que cambia de plano.
- Arco iris: defina una matriz con los siete colores del arco iris. Inicialmente, el núcleo del cubo (todos los LED internos) debe establecerse en rojo y el exterior (todos los LED externos) en violeta. Cada dos segundos, el color del núcleo se transfiere al exterior y el siguiente color en la secuencia de la matriz llena el núcleo.
- Espiral: una espiral es una curva, que emana de un punto central, alejándose progresivamente a medida que gira alrededor del punto inicial. En esta práctica, el alumno debe crear una espiral azul invertida tridimensional que emana de un punto lejano creando una curva para llegar al punto central. La espiral debe crearse a partir del punto de origen (0,0), que representa los ejes Z e Y respectivamente y llenando todo el eje X. Es decir, cuando nace la espiral, se crea una línea en $z = 0$, e $y = 0$, encendiendo todos los LED del eje X. Luego la espiral se extiende hasta $z = 1$, $y = 0$, volviendo a configurar todos los LED del eje X en ese punto y así sucesivamente, hasta llegar al punto central y configurar todo el cubo. Posteriormente, aparece una nueva espiral desde el mismo punto inicial (0,0) cerrando el eje X en ese punto y repitiendo el mismo proceso hasta que se apaga todo el cubo.

B. Laboratorio de Sensores

El segundo experimento que describimos es un laboratorio de sensores Arduino (Figura 7). Este laboratorio remoto permite a los estudiantes aprender a programar los sensores más populares que se utilizan normalmente para Arduino, como:

- Sensor de temperatura. El sensor utilizado incluye el amplificador operacional LM358 y una resistencia de $10K\Omega$. El termistor TTC03 se utiliza como sensor de temperatura, que tiene un rango de medición entre $-40^\circ C$ y $+125^\circ C$ con una precisión de $\pm 1,5^\circ C$.
- Sensor profesional de temperatura y humedad. El sensor de temperatura y humedad pro utiliza el sensor DHT22. Este es un sensor digital de bajo costo que se usa para medir la humedad y la temperatura. Utiliza un sensor capacitivo para medir la humedad y un termistor para medir la temperatura. Incorpora internamente un chip que realiza una conversión de analógico a digital, por lo que el sensor proporciona únicamente una salida digital. Esta es una ventaja frente a otros sensores que tienen salida analógica, ya que de esta forma se evitan las fluctuaciones de voltaje. La desventaja de este sensor es que el tiempo mínimo de muestreo es de dos segundos.
- Módulo de infrarrojos pasivos (PIR). Es un sensor de infrarrojos que es capaz de detectar tanto personas como objetos. Es muy utilizado en el campo de los sistemas de seguridad electrónica para detectar por ejemplo el paso de una persona a cualquier zona previamente programada.
- Sensor de luz. Es un módulo que está listo para trabajar directamente con él, ya que está compuesto por un divisor de voltaje que utiliza el sensor de luz *Light Dependent Resistor* (LDR) GL5528. La resistencia de este sensor disminuye cuando aumenta la luz ambiental; la resistencia del LDR se encuentra entre 10 y $20 K\Omega$ cuando hay 10 lux y $20 M\Omega$ cuando está en un ambiente oscuro.
- Sensor de fuego. Está compuesto por el sensor DFR0076 que detecta longitudes de onda entre 760 y 1100 nm de luz. Tiene un rango de medición de 60° y una sensibilidad bastante alta.
- Módulo de micrófono con amplificador. El módulo consta de un micrófono, el amplificador LM358 y un potenciómetro. Se utiliza para ajustar la ganancia del amplificador, que oscila entre 20 y 200 . El módulo tiene una salida analógica que Arduino mostrará con un valor entre 0 y 1023 , y se alimenta con un voltaje de entrada entre 4 y $12V$.
- Sensor de gases y humos MQ2. Este módulo para la detección de gases y humos está compuesto por el sensor MQ2, que se utiliza para detectar concentraciones de gas natural comprimido; sin embargo, también puede detectar propano, gases licuados de petróleo, humo, isobutano y alcohol. Es un sensor de alta sensibilidad y alta velocidad de respuesta. El funcionamiento interno del sensor se basa en el hecho de que cuando está en una atmósfera con aire limpio, la conductividad interna es muy baja; mientras que si es en una atmósfera donde hay una concentración del gas a medir, la conductividad del sensor aumenta a medida que aumenta la concentración del gas.
- Sensor ultrasónico PING. El sensor ultrasónico que se utilizará en el proyecto es el sensor TS601P01. Este telémetro eléctrico tiene un rango de medición de

entre 3 cm y 3 m con una precisión de ± 2 cm. Con la implementación de tecnología de procesamiento inteligente a través de software se puede obtener una mejora en las medidas del sensor, así como mejorar la capacidad de evitar posibles interferencias. El funcionamiento genérico de un sensor ultrasónico es similar al de un radar. El sensor emite un pulso de sonido de alta frecuencia, imperceptible para el oído humano, y mide el tiempo desde que el pulso se emite y se recupera desde la superficie hasta el origen.

- **Acelerómetro ADXL335.** El sensor ADXL335 se instala en un módulo que incorpora los pines necesarios para la conexión con Arduino. Este módulo tiene seis pines, aunque solo se utilizan cinco de ellos; que corresponden al voltaje de entrada de 3.3V (Vcc), tierra (GND), señal de salida del eje X (X), señal de salida del eje Y (Y) y señal de salida del eje Z (Z).
- **Reloj en tiempo real.** Para proyectos que trabajan con sensores es muy importante implementar un RTC, ya que si quieres almacenar los datos de los sensores tienes que tener una huella temporal para poder realizar un estudio de los mismos posteriormente. Por ello se ha decidido describir cómo se podría programar esta huella temporal para futuras ampliaciones de este trabajo.
- **Sensor barométrico.** Es un módulo diseñado para ser utilizado directamente con Arduino, ya que incorpora el circuito de acondicionamiento de señal y las salidas y entradas necesarias para trabajar con Arduino. El módulo utiliza el sensor BMP005, fabricado por Bosch, que permite medir la presión barométrica y la temperatura. Actualmente es uno de los sensores de menor coste y mayor precisión comparado con otros de los que se encuentran en el mercado. El circuito de acondicionamiento permite que la alimentación del módulo se haga con los 3.3V o 5V de Arduino, y estas entradas se utilizan para obtener el voltaje de 1.8V o 3.6V necesario para alimentar el chip BMP005. Tiene un rango de medida de presión barométrica entre 300 y 100hPa (90000m y -500m sobre el nivel del mar), con una resolución de 0.03hPa (0.25 metros). En cuanto a la temperatura, puede realizar medidas entre -40°C y $+85^{\circ}\text{C}$, con una resolución de $\pm 2^{\circ}\text{C}$. El chip BMP005 ha sido fabricado utilizando tecnología piezorresistiva para tener una mayor protección frente a campos electromagnéticos, logrando alta precisión, linealidad y estabilidad.



Fig. 7 Interfaz del Laboratorio de Sensores con Arduino.

Además de todos estos sensores, el experimento incluye una pantalla LCD donde el alumno puede escribir los datos del sensor después de la lectura. Así, el resultado de la ejecución se puede observar a través de la webcam y los estudiantes entienden, nuevamente, que están usando un equipo real, no una simulación.

En este caso, debido a que los estudiantes necesitan aprender a usar cada sensor, la curva de aprendizaje es mayor que la del laboratorio de cubos LED, tanto a nivel de su funcionalidad, como sus bibliotecas y cómo leer la información y procesarla.

Las prácticas propuestas para este experimento remoto son incrementales. Por lo tanto, en primer lugar, se pide a los estudiantes que creen una función simple que escriba una cadena en la pantalla LCD. Una vez que los estudiantes tengan una idea clara de cómo escribir en la pantalla LCD, pueden comenzar a trabajar con los sensores. Luego, comienzan a trabajar con cada uno de los sensores de forma independiente. El modo de funcionamiento con un solo sensor es el siguiente:

1. Configurar las bibliotecas del sensor.
2. Escribir el código para leer desde el puerto GPIO específico donde está conectado el sensor.
3. Leer datos sin procesar del sensor.
4. Utilice la biblioteca de sensores para traducir los datos sin procesar a información comprensible para humanos.
5. Escriba la información producida en la pantalla LCD.
6. Vaya al paso 3 del bucle de nuevo.

Una vez que se ha entendido el sensor actual, el estudiante puede pasar a otro sensor. Como resultado, los estudiantes inicialmente irían mejorando sus competencias con cada sensor por separado, para luego comenzar a mezclar funcionalidades de varios sensores al mismo tiempo.

Las competencias adquiridas por los alumnos tras trabajar en este laboratorio remoto son las siguientes:

- Los estudiantes aprenden a interactuar con los pines analógicos Arduino GPIO leyendo los sensores conectados a estos pines.
- Los estudiantes adquieren comprensión del modo de operación de los sensores más populares utilizados para la programación de Arduino.
- Los estudiantes aprenden a escribir en una pantalla LCD.
- Los estudiantes mejoran sus habilidades de programación con respecto a bucles y declaraciones condicionales.

V. CONCLUSIONES

El uso de laboratorios remotos en la enseñanza de la ingeniería es cada día más importante debido a la necesidad de brindar un acceso adecuado para adquirir competencias prácticas en el uso de instrumentos reales en cualquier momento y lugar, no solo en el ámbito del aula.

Este artículo ha analizado el estado del arte de los laboratorios remotos basados en arquitectura Arduino y ha descrito el propuesto por los autores. Este laboratorio remoto incluye varios experimentos, como un cubo LED

3D RGB y un laboratorio de sensores. Estos experimentos se están utilizando en el Grado de Ingeniería Electrónica en una Universidad de Educación a Distancia como parte de las prácticas obligatorias de una asignatura.

Respecto a las limitaciones de este sistema remoto, podemos remarcar que la cantidad de puertos USB disponibles limita la cantidad de placas Arduino conectadas al servidor. Sin embargo, dado que el servidor propuesto es Raspberry Pi, con un sistema Linux, es muy atractiva en coste y rentable al disponer de 4 puertos USB (en la Raspberry Pi 3 modelo B). Por tanto, es posible disponer de varios servidores, cada uno de los cuales ofrecerá un experimento a un coste bajo. Además, dado que la literatura muestra que el uso de laboratorio remoto aumenta la motivación de los estudiantes con respecto al uso de simulaciones, la baja inversión en este equipo está absolutamente justificada.

Como trabajo futuro, se desarrollarán nuevos experimentos basados en este laboratorio remoto. Las nuevas placas Arduino se conectarán al servidor Raspberry Pi con nuevos periféricos educativos, como una grúa robótica, un laboratorio de IoT y laboratorios de reconocimiento de imágenes, entre otros.

AGRADECIMIENTOS

Este trabajo ha sido cofinanciado por el Gobierno de la Comunidad de Madrid a través del proyecto e-Madrid-CM (S2018/TCS-4307). El proyecto e-Madrid-CM es también co-financiado por los Fondos Estructurales (FSE y FEDER)." Los autores también quieren reconocer la ayuda dada por el proyecto IoE-EQ (2017-1-IT01-KA202-006251) y la Escuela de Ingeniería Industrial de la UNED con la ayuda 2020-IEQ-13. J. A. R-V quiere agradecer al Ministerio Español de Economía y competitividad por el soporte mediante el programa de Juan de la Cierva Formación (FJCI-2017-34926).

REFERENCIAS

- [1] Viegas, C., Pavani, A., Lima, N., Marques, A., Pozzo, I., Dobboletta, E., Atencia, V., Barreto, D., Calliari, F., Fidalgo, A., Lima, D., Temporão, G., Alves, G., "Impact of a remote lab on teaching practicum and student learning", *Computers & Education*, vol. 126, pp. 201-216, Nov 2018.
- [2] Liu, D., Valdiviezo-Díaz, P., Riofrio, G., Sun, Y.M., Barba, R., "Integration of Virtual Labs into Science E-learning", *Procedia Computer Science* 75 (2015) 95 – 102, Nov 2015.
- [3] Tawfik, M., Sancristobal, E., Martín, S., Gil, R., Díaz, G., Colmenar, A., Peire, J., Castro, M., Nilsson, K., Zackrisson, J., Hakansson, L., Gustavsson, I., Virtual instrument systems in reality (VISIR) for remote wiring and measurement of electronic circuits on breadboard. *IEEE Transactions on learning technologies* 6 (1), 60-72, 2012.
- [4] Tawfik, M., Sancristobal, E., Martín, S., Díaz, G., Peire, J. and Castro, M., Expanding the boundaries of the classroom: Implementation of remote laboratories for industrial electronics disciplines. *IEEE Industrial Electronics Magazine* 7 (1), 41-49, 2013.
- [5] Sancristobal, E., Pesquera, A., Orduña, P., Ruiz, E. Gil, R., Martín, S., Díaz, G., Albert, M., Colmenar, A., Meier, R., and Castro, M., Virtual and remote industrial laboratory: Integration in learning management systems. *IEEE Industrial Electronics Magazine* 8 (4), 45-58, 2014.
- [6] Joya-Guirado, L., Martín, S., and Castro, M., Architecture of an Open Remote Lab for Digital Electronics. *Dyna* 91 (6), 599-600, 2016.
- [7] Bonilla, M.F.Y., Gutierrez, S.M., Orueta, G.D., Gil, M.A.C., Risk analysis and recommendations for electronic design with Arduino. *DYNA* 92 (6), 607-608, 2017.
- [8] Cubillo, J., Martín, S., and Castro, M., Design of a 3D Printing Remote Lab. *DYNA* 92 (1), 15-15, 2017.
- [9] Gutierrez, S.M., Guirado, L.J., and Gil, M.A.C., *DYNA Ingeniería e Industria* 91 (6), 599-600, 2016.
- [10] Fernández-Pacheco, A., Martín, S., and Castro, M., Implementation of an Arduino Remote Laboratory with Raspberry Pi. 2019 IEEE Global Engineering Education Conference (EDUCON), 1415-1418, 2019.
- [11] Jamieson, P., Arduino for Teaching Embedded Systems. Are Computer Scientists and Engineering Educator Missing the Boat?. Universidad de Miami.
- [12] Balamuralithara, B., Woods, P. C., "Virtual Laboratories in Engineering Education: The Simulation Lab and Remote Lab". *Computer Applications on Engineering Education*, August 2018.
- [13] Balogh, R., Educational Robotic Platform based on Arduino. Proceedings of the 1st international conference on Robotics in Education, RiE2010.
- [14] Grodotzki, J., Ortelt, T.R., Tekkaya, A. E., "Remote and Virtual Labs for Engineering Education 4.0: Achievements of the ELLI project at the TU Dortmund University" *Procedia Manufacturing*, Volume 26, 2018, Pages 1349-1360.
- [15] Čolaković, A, Hadžialić, M., "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues", *Computer Networks* 144, July 2017.
- [16] G. Carro et al., "The color of the light: A remote laboratory that uses a smart device that connects teachers and students," 2014 IEEE Global Engineering Education Conference (EDUCON), Istanbul, 2014, pp. 854-860.
- [17] J. Sarik and I. Kymissis, "Lab kits using the Arduino prototyping platform," 2010 IEEE Frontiers in Education Conference (FIE), Washington, DC, 2010, pp. T3C-1-T3C-5.
- [18] A. Mejías Borrero, M.A. Márquez Sánchez, J. M. Andújar Márquez, M. R. Sánchez Herrera, A Complete Solution for Developing Remote Labs, *IFAC Proceedings Volumes*, Volume 46, Issue 17, 2013, Pages 96-101.
- [19] Martín Kalúz, Luboš Čírka, Richard Valo, Miroslav Fikar, ArPi Lab: A Low-cost Remote Laboratory for Control Education. *IFAC Proceedings Volumes*, Volume 47, Issue 3, 2014, Pages 9057-9062.
- [20] A. Cardoso, V. Sousa, P. Gil. Demonstration of a remote control laboratory to support teaching in control engineering subjects. *IFAC-PapersOnLine* Volume 49, Issue 62016, Pages 226-229.