



PLATAFORMA DE ENSEÑANZA A DISTANCIA DE MICROCONTROLADORES E INTERNET DE LAS COSAS

PLATFORM FOR DISTANCE LEARNING OF MICROCONTROLLERS AND INTERNET OF THINGS

Renata Pereira^{1,*} , Cleonilson de Souza² , Darwin Patino³ , Juan Lata⁴ 

Recibido: 06-06-2022, Recibido tras revisión: 14-06-2022, Aceptado: 21-06-2022, Publicado: 01-07-2022

Resumen

Debido al creciente desarrollo tecnológico de los sistemas embebidos y el internet de las cosas (IoT), los dispositivos basados en microcontroladores se aplican cada vez más en diversas áreas del conocimiento. En este contexto, las plataformas y productos educativos online se consideran laboratorios virtuales remotos, ya que los estudiantes pueden acceder a los dispositivos físicos en cualquier lugar siempre que dispongan de un sistema informático conectado a internet. En este sentido, este trabajo describe el diseño y desarrollo de una plataforma con cuatro productos educativos en línea para la enseñanza de microcontroladores e IoT. Estos productos educativos están basados en *software* de código abierto, lo que permite su distribución gratuita en línea y el acceso a ellos desde un servidor en la nube. El enfoque de código abierto y multiplataforma (Linux, Windows® y Mac OSX) permite una mayor interacción y accesibilidad del usuario. Los productos educativos en línea hacen posible la programación del firmware del microcontrolador ESP32 de forma remota a través de OTA (*Over the Air*) y de sistemas embebidos Linux basados en Raspberry Pi (Rpi), permitiendo aplicaciones virtuales de laboratorio de microcontroladores. Además, los productos educativos en línea permiten manipular los pines vía internet a través de la interfaz gráfica de los microcontroladores ESP32 y ESP8266, así como del Rpi. Como resultado, se probó la plataforma en línea propuesta, ejecutándola en un servidor en la nube, y se validaron los cuatro productos educativos de aprendizaje a distancia y la aplicación real de microcontroladores e IoT, que se encuentran operativos, tal como fueron diseñados.

Palabras clave: plataforma, educación a distancia, microcontrolador, internet de las cosas, sistemas embebidos, laboratorio virtual

Abstract

Due to the increasing technological development in embedded systems and the Internet of Things (IoT), devices based on microcontrollers are being increasingly applied in various areas of knowledge. In this context, online educational platforms and products are considered virtual remote laboratories, because students can access the physical devices anywhere as long as they have a computer system connected to the internet. In this sense, this work describes the design and development of a platform with four online educational products for teaching microcontrollers and IoT. These educational products are based on open-source software, which enables their free online distribution and the access to them from a cloud server. The open-source and multiplatform (Linux, Windows®, and Mac OSX) approach enables a greater user interaction and accessibility. The online educational products make possible the remote programming of the ESP32 microcontroller firmware via OTA (over the air) and Linux embedded systems based on Raspberry Pi (Rpi), enabling virtual microcontroller laboratory applications. In addition, the online educational products enable the manipulation of GPIO pins via the Internet through the graphical interface of the ESP32 and ESP8266 microcontrollers and the Rpi. As a result, the proposed online platform was tested, running it on a cloud server, and the four-distance learning educational products and the actual microcontroller and IoT application have been validated, and are operating as designed.

Keywords: platform, distance learning, microcontroller, Internet of Things, embedded systems, virtual laboratory

¹ Electronics, Federal Institute of Alagoas, Alagoas, Brazil. Autor para correspondencia ✉: renata.pereira@ifal.edu.br

² Electrical Engineering, Federal University of Paraíba, Paraíba, Brazil.

³ Electrical Engineering, Universidad de Guayaquil, Guayaquil, Ecuador.

⁴ Electrical Engineering, Universidad Politécnica Salesiana, Guayaquil, Ecuador.

1. Introducción

La pandemia de la COVID-19 demostró que el proceso de enseñanza-aprendizaje en instituciones educativas fue obstaculizado por la falta y la necesidad de clases presenciales. Varias estrategias se han implementado en las escuelas y universidades para reducir el impacto negativo sobre el aprendizaje de los estudiantes y la continuidad de la educación [1, 2]. Adoptar prácticas a distancia se ha convertido en una parte crucial de la enseñanza, especialmente en cursos de Ingeniería, Ciencias y Tecnología. Así, este artículo describe una herramienta didáctica que permite la implementación de prácticas en línea de microcontroladores y el internet de las cosas [3].

Debido al creciente desarrollo tecnológico en las áreas de sistemas embebidos y el internet de las cosas, los dispositivos basados en microcontroladores se aplican cada vez más en diversas áreas del conocimiento tales como control de motores, sistemas de seguridad, sistemas de control de elevadores [4], monitoreo de generación eléctrica [5] y en investigación científica aplicada [6]. Por lo tanto, la enseñanza de tópicos como sistemas embebidos, microcontroladores e IoT se hace cada vez más relevante en cursos de Ciencias e Ingeniería.

Dado que los productos de educación en línea basados en microcontroladores e IoT se utilizan sobre el Internet, es posible realizar prácticas de enseñanza remota de estos productos. De esta manera, las plataformas y los productos educativos en línea se consideran laboratorios remotos, dado que los estudiantes pueden acceder a dispositivos físicos desde cualquier lugar siempre que tengan un sistema informático con acceso a internet [7, 8].

1.1. Trabajos relacionados

La computadora de placa única, conocida como Raspberry Pi (Rpi), empleada en la plataforma educativa de enseñanza a distancia propuesta en este trabajo, fue desarrollada por la Raspberry Pi Foundation para ayudar a niños y adultos a aprender en el área de computación. El Raspberry Pi es una computadora portátil de bajo costo basada en *software* libre e interfaces para varios periféricos. Otras características son 1 GB de RAM compartida con un GPU (unidad de procesamiento gráfico, *Graphics Processing Unit*, en inglés), núcleo gráfico VideoCore IV 3D, 40 pines de entrada/salida de propósito general (*general-purpose input/output*, GPIO, por sus siglas en inglés), cuatro puertos con interfaz USB 2.0 para conectar el teclado y el ratón en computadoras de escritorio, puerto HDMI (interfaz multimedia de alta definición, *High Definition Multimedia Interface*, en inglés), puerto Ethernet con interfaz RJ-45, interfaz de tarjeta 56 micro SD (digital segura, *Secure Digital*, en inglés), interfaz de cámara y

monitor, así como también fuente de poder de 5 V y salida de audio [9].

Para desarrollar productos educativos en línea basados en RPi, se utilizan un conjunto de librerías y paquetes para asegurar la comunicación entre todos los componentes del sistema. Entonces, los protocolos Rpi SSH (*Secure Shell*) y VNC (*Virtual Network Computing*) que permiten el acceso a la consola y al ambiente gráfico, respectivamente. El SSH es un protocolo que permite la comunicación segura entre dos sistemas usando una arquitectura cliente/servidor, y permite a los usuarios conectarse a un host remoto. A diferencia de otros protocolos remotos tales como FTP (protocolo de transferencia de archivos, *File Transfer Protocol*, en inglés) o Telnet, el SSH encripta la sesión de acceso, haciendo virtualmente imposible para intrusos obtener claves descriptadas [10].

El paquete Samba, una implementación de código abierto del protocolo SMB (*Server Message Block*), fue instalado y configurado para asegurar la interconexión entre computadores que están ejecutando los sistemas operativos Microsoft Windows y Linux, y compartir archivos y carpetas [11].

La librería WiringPi C fue utilizada para los programas en lenguaje C del Rpi; esta librería puede usarse para programar y configurar los pines GPIO [12]. Por lo tanto, para compilar programas escritos en lenguaje C se tiene el compilador GCC (*GNU Compiler Collection*), que es una distribución accesible e integrada de compiladores para lenguajes de programación tales como C, C++, C#, y Java [13].

También se usó la librería CURL como una herramienta de línea de comando de código abierto para la construcción de sintaxis URL y la transferencia de datos a la nube [14].

El microcontrolador ESP32 IoT fue lanzado por la compañía china Espressif Systems, un fabricante de antenas wifi embebidas y módulos IoT con *software* y *hardware* libre [15]. El ESP32 es un microcontrolador (MCU) de 32 bits de baja potencia, con wifi/Bluetooth, núcleo dual y modo dual. El MCU ESP32 tiene una frecuencia de 240 MHz y un poder de procesamiento de 600 DMIPS (*Dhrystone* millones de instrucciones por segundo).

Por otra parte, comparado con el modelo más reciente, el microcontrolador de 32 bits ESP8266 NodeMCU IoT con antena wifi es una versión anterior y más simplificada, también lanzada por Espressif [16].

El microcontrolador PIC (*Peripheral Interface Controller*) es un circuito integrado que contiene toda la circuitería necesaria para instalar un sistema digital programable completo en un único dispositivo. Esta plataforma de enseñanza también utiliza el microcontrolador PIC18F2550 de 8 bits producido por Microchip Technology.

El módulo convertidor analógico a digital (ADC) del PIC18F2550 tiene diez entradas, y es capaz de

medir el voltaje analógico en cada canal y convertir el voltaje medido en un valor digital de 10 bits [17].

En este contexto, la tecnología IoT permite la comunicación entre dispositivos sin intervención humana [18, 19] y entiende que cada dispositivo es autónomo y está conectado a internet. Tales dispositivos pueden interactuar y comunicarse automáticamente, maximizando la modularidad, confort, conveniencia, seguridad y ahorro de energía [20]. Se usa típicamente una base de datos para almacenar data en la nube para poner en marcha sistemas IoT. Para este propósito, la computación en la nube proporciona acceso a una *pool* compartido de recursos computacionales configurables, tales como bases de datos no relacionales, sin requerir un conocimiento profundo de tecnologías de gestión [21, 22]. El modelo de nube simplifica la instalación, operación y mantenimiento de sistemas de información, incrementando la eficiencia y la confiabilidad del sistema y reduciendo costos.

Los sistemas nube pueden clasificarse de acuerdo con el modelo de desarrollo o de servicio, que pueden ser públicos, privados, comunitarios o híbridos. Los modelos nube de servicio son plataforma como un servicio (*Platform as a Service*, PaaS, por sus siglas en inglés), *software* como un servicio (*Software as a Service*, SaaS, por sus siglas en inglés) o infraestructura como un servicio (*Infrastructure as a Service*, IaaS, por sus siglas en inglés). En este trabajo se utiliza PaaS, que consiste en el uso de herramientas y recursos para proveer servicios para usuarios finales, los estudiantes. Dado que los usuarios finales comparten información en el mismo servidor en la nube, la privacidad y confidencialidad de los datos son las principales preocupaciones. Por lo tanto, los productos educativos de la plataforma de aprendizaje hacen posible que la data almacenada esté segura y encriptada, permitiendo ser administrada solo por el estudiante propietario.

Así, para escribir y leer data variable en la nube, la plataforma propuesta de aprendizaje a distancia usa *Firebase Realtime Database*, que es una base de datos NoSQL, es decir, que no tiene como estándar el sistema de tablas y relaciones entre datos, y donde la data se almacena como objetos JSON [23]. Adicionalmente, la plataforma didáctica en línea para aprendizaje a distancia sobre microcontroladores permite uso remoto en tiempo real por múltiples usuarios.

2. Materiales y métodos

2.1. Problema y metodología

Los dispositivos IoT requieren de manera creciente una rápida intervención para actualizar librerías y funcionalidades y mantener la seguridad de los ambientes en línea. Con este fin, actualizar rápidamente el firmware de estos dispositivos IoT es una manera efectiva de garantizar la seguridad de los datos. Así, la

plataforma instruccional de aprendizaje a distancia utiliza actualizaciones del firmware IoT vía Over-The-Air (OTA). La actualización del *firmware* OTA se realiza vía HTTP, lo cual permite descargar un archivo binario de un servidor en la nube para actualizar el firmware de los microcontroladores remotamente. La actualización del OTA a través de internet se instala típicamente de dos maneras: (i) Si está disponible una versión más nueva del firmware, el dispositivo consulta periódicamente al servidor que contiene el archivo binario compilado (*pull*) u otro servicio de activación en línea, tal como *Firebase*. (ii) El dispositivo recibe una notificación de una nueva versión del *firmware* a través de la nube (*push*), por ejemplo, vía MQTT, y realiza la actualización. El servidor almacena la versión más reciente del firmware en binario, y luego la URL variante en el tiempo del archivo binario, lo cual es más seguro y flexible que una dirección fija predefinida, se envía al microcontrolador para descarga [24]. En este tópico, se presentan los productos educacionales en línea de la plataforma didáctica de aprendizaje a distancia sobre microcontroladores e internet de las cosas.

3. Resultados y discusión

Este artículo ilustra un producto educativo para actualizar los códigos de los microcontroladores ESP32 y ESP266 con el mismo firmware genérico se emplea <http://sanusb.org/espupdate>. Se ha instalado la versión libre de Firebase (Google's JSON object database) para generar la activación de una actualización asíncrona en la nube para archivos .bin, como se ilustra en la Figura 1.

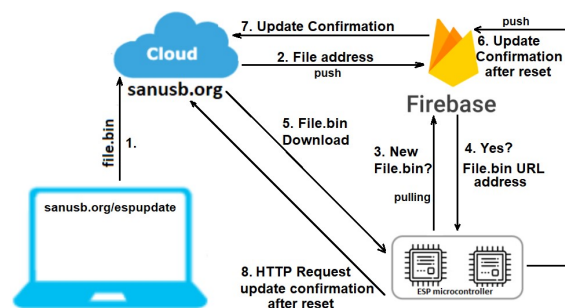


Figura 1. Ilustración de la operación del ambiente Espupdate

La actualización (OTA) transmite los archivos binarios .bin compilados a una computadora remota a través de internet, activando las aplicaciones virtuales del laboratorio de microcontroladores. Para realizar esta actualización de firmware en la nube, los usuarios deben escribir en el firmware solo el SSID, la clave, y el mismo nombre de perfil ingresado en el sitio web <http://sanusb.org/espupdate> y al subir el archivo .bin en el sitio web se actualiza remotamente

el firmware de los microcontroladores ESP a través de internet. El nombre de perfil de usuario ingresado en sanusb.org/espupdate puede ser alfanumérico.

Es posible probar este producto webcast educativo en diferentes redes accediendo al sitio web <http://sanusb.org/espupdate/> a través de la red de laboratorio o doméstica, y de los microcontroladores ESP32 o ESP8266 acoplado al teléfono inteligente conectado a la red móvil 4G y viceversa.

Vale la pena considerar que, a través de pruebas realizadas, se observó que para los archivos .bin compilados con el mismo nombre y descargas secuenciales para la actualización en la nube, puede ocurrir que el microcontrolador ESP no descarga el último archivo .bin enviado para actualización, sino un archivo .bin previamente cargado, dado que tienen el mismo nombre y URL de descarga.

Por esta razón, en este proyecto el nombre de los archivos .bin cargados al sitio tienen nombres basados en el tiempo de carga, así como también en la dirección URL de descarga; esto evita que se descargue un archivo previamente cargado u otro archivo con el mismo nombre, al generar una URL única y sin pareja. En este caso, para la versión del nombre y la dirección de descarga del archivo .bin, es año, mes, día, tiempo, y el orden cíclico de carga [25].

Una vez que se ha completado la actualización del archivo .bin y que el microcontrolador se ha reiniciado automáticamente, se envía el nuevo código de verificación a http://sanusb.org/espupdate/*Profile*/ para confirmar la actualización en el sitio web. Es importante notar que para utilizar EspUpdate no es necesario abrir puertos en el enrutador o habilitar permisos en el cortafuegos.

3.1. Producto educativo ESPGpio

El producto educativo denominado ESPGpio permite cambiar el nivel lógico de los pines de los microcontroladores ESP32 y ESP266 que pueden ser programados con el firmware de código abierto [26]; dichos pines se ilustran en la Figura 2 a través del acceso gráfico disponible en sanusb.org/esp. El nombre de perfil de usuario ingresado en sanusb.org/espgpio puede ser alfanumérico.

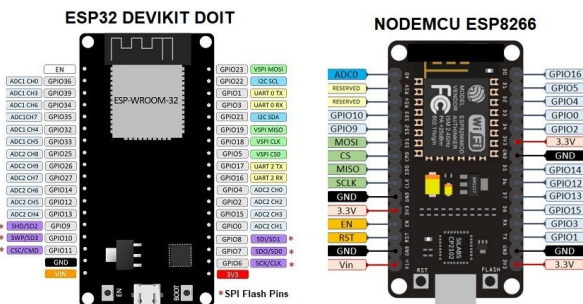


Figura 2. Ilustración de los pines GPIO del ESP32 y del ESP8266

La Figura 3 muestra una ilustración de la operación automática del entorno ESPGPIO. En este caso, se utiliza una celda de Firebase para compartir entre el controlador y el entorno gráfico de usuario. Un entero par se representa por una suma de 2; este es elevado a la potencia correspondiente al número de pines activados.

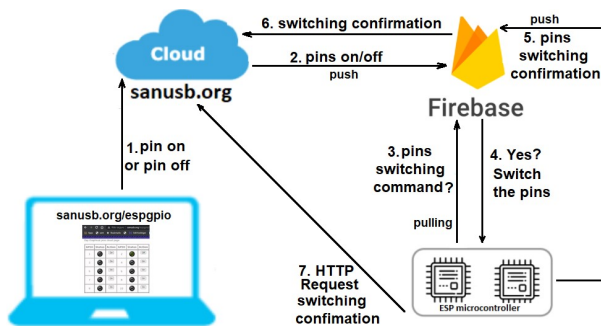


Figura 3. Ilustración los pines GPIO de un ESP32 o ESP8266

El entorno gráfico generado en sanusb.org/espgpio, ilustrado en la Figura 4, tiene treinta pines GPIO que el usuario debe seleccionar para la conmutación de carga a través de relés o dispositivos electrónicos tales como LED. Es importante notar que, si existe un fallo eléctrico o caída de voltaje, el estado de los últimos pines retorna a normal cuando se restaura la caída de voltaje, así como el estado de los pines es actualizado y consultado en Firebase, que es una base de datos persistente.

Después de completar el cambio en el estado de los pines, el nuevo estado del pin se indica con color verde en el entorno gráfico http://sanusb.org/espgpio/*Profile/, si está a un nivel lógico alto, como se ilustra en la Figura 4, donde *Profile* es el perfil dado por el usuario.

Esp Graphical pins cloud page

GPIO	Status	Action	GPIO	Status	Action
1		On	2		Off
3		On	4		Off
5		Off	6		On
7		On	8		On
9		On	10		On

Figura 4. Ilustración del entorno gráfico de los pines GPIO del ESP 32 y ESP8266

Es importante notar que se recomienda evitar el pin GPIO 1 en el ESP8266 Nodemcu, que es el pin Tx, ya que este previene la comunicación de depurado del programa por el monitor serial, y los pines GPIO 6 a 11 del ESP32 Devkit, usados para grabación flash.

3.2. Producto educativo RPI GPIO

El producto educativo RPI GPIO es similar al ESP-gpio, pero utilizado para cambiar el estado de los pines lógicos en la tarjeta Raspberry Pi. La Figura 5 muestra la ilustración de los pines de una Raspberry Pi.

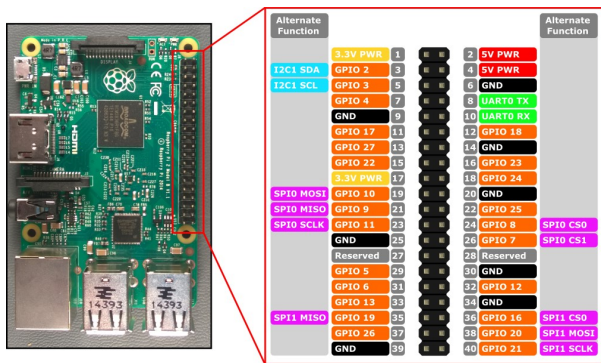


Figura 5. Ilustración de los pines GPIO de una Raspberry Pi

Después de crear el perfil en sanusb.org/gpio, se genera una página con enlaces a páginas para acceder a los pines del Rpi y los botones del firmware para la aplicación de extracción y la aplicación de inserción en el lenguaje bash shell, como se ilustra en la Figura 6.

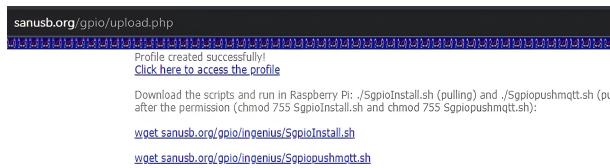
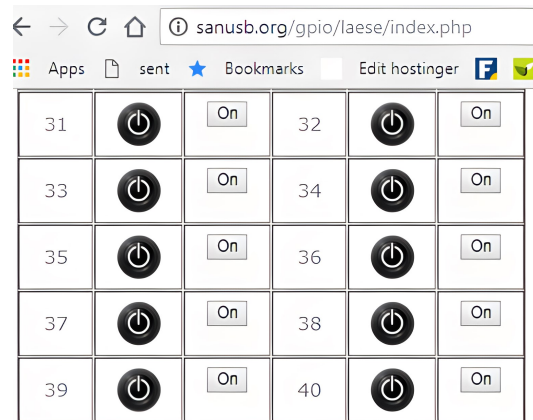


Figura 6. Enlace a páginas con botones de pines y programas para la Rpi

Tanto para la aplicación completa como para la aplicación push, es necesario, después de descargar los programas, otorgar permiso en el terminal como superusuario (sudo su) y ejecutar el permiso con los comandos `chmod 755 SgpioInstall.sh` para la aplicación pull y `chmod 755 SgpioPushmqtt.sh`, para la aplicación push [27]. Al hacer clic en el enlace del perfil, se genera una página con botones que representan cada uno de los pines I/O de una Rpi, como se ilustra en la Figura 7.

En el caso del pull, al hacer clic en el botón de un pin físico dado se envía un comando a un archivo contenido en el perfil en línea, consultado periódicamente por el script `Sgpiointall.sh` que está operando

en el Rpi. En el caso del push MQTT, al hacer clic en el botón se envía un comando publish al bróker `mqtt.eclipse.org` sobre el tópicos con el mismo nombre del perfil de usuario, laese en el caso del ejemplo, y descarga este comando al Rpi que está operando el script `Sgiopushmqtt.sh` con suscripción (`subscribe`) al mismo tópicos.



Data acquisition

wget.sanusb.org/gpio/laese/SgpioInstall.sh

wget.sanusb.org/gpio/laese/Sgpiopushmqtt.sh

Figura 7. Ilustración del entorno gráfico de los pines GPIO del Raspberry Pi

Con el entorno RPI GPIO es posible crear una aplicación IoT utilizando una computadora portátil o un teléfono inteligente para, por ejemplo, controlar la activación de electrodomésticos vía internet, tal como en ventilador vía un relé, como se ilustra en la Figura 8. En este caso se tiene un Raspberry Pi Zero, también accesible vía internet. Como puede observarse, para usar el entorno `sanusb.org/gpio` es necesario crear inicialmente un perfil. Después de completar el perfil, se genera un enlace para acceder a una página con botones, y cada botón corresponde a un pin del dispositivo Raspberry Pi IoT [27].

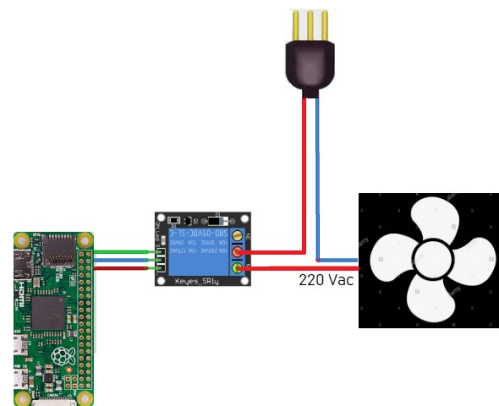


Figura 8. Ilustración de un circuito para operación IoT de un ventilador con Raspberry Pi

3.3. Producto educativo IOTUS

El producto educativo IOTUS (IoT update system, sistema de actualización IoT en español) es una PaaS, es decir, una plataforma como servicio en la cual el usuario puede crear un perfil y actualizar scripts en C, Python o lenguaje shell en el sistema embebido Linux basado en Rpi, y también actualizar el firmware en hexadecimal de la familia de microcontroladores PIC a través del puerto USB del Rpi [28], como se ilustra en la Figura 9.

Por tanto, IOTUS consta de tres partes principales, a saber: 1) Sistema embebido Linux basado en Raspberry Pi (ELS), 2) Sistema embebido de conversión Analógico a digital (ADCES), el cual utiliza un microcontrolador SanUSB basado en una herramienta de *software* y *hardware* libre con la familia PIC18Fxx5X, incluyendo la interfaz USB nativa; y 3) Página web para la actualización automática en tiempo-real (carga) de scripts.

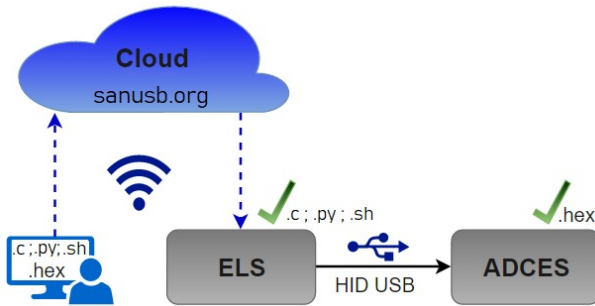


Figura 9. Actualización remota de firmware y script

El ADCES corresponde a la herramienta de desarrollo SanUSB de código abierto, ilustrada en la Figura 9. El SanUSB está constituido por *software* y *hardware* de código abierto de la familia PIC18Fxx5x con una interfaz USB nativa. Esta herramienta libre es eficiente en el desarrollo de proyectos porque no es necesario remover el microcontrolador para actualizar el firmware, a diferencia de otros sistemas de desarrollo que típicamente utilizan hardware específico de grabación y requieren una conexión cableada. Además, la herramienta SanUSB es una multiplataforma, es decir, puede ser empleada en Windows®, Mac OSX, y Linux, así como también es plug and play, es decir, es reconocida automáticamente por los sistemas operativos sin la necesidad de instalar ningún driver [29].

Por tanto, esta herramienta permite que la compilación, grabación y emulación de un programa sean realizadas de manera rápida y eficiente, conectando el microcontrolador a un sistema informático vía USB. Su aplicación es un RPi. El sistema informático ELS basado en RPi puede implantar diseños electrónicos directamente, debido principalmente a sus pines de entrada/salida de propósito general (GPIO) con soporte para sensores digitales y actuadores, y protocolos

I2C y SPI para establecer comunicaciones con los periféricos [30]. La variedad de comunicación permite al RPi comunicarse con un rango más amplio de periféricos, y la librería de acceso a los pines Wiring Pi, escrita en C, facilita la programación y configuración de los pines GPIO a través de la utilidad de línea de comando «gpio». Este trabajo tiene dos maneras de comunicación entre ADCES y RPi: comunicación serial e interfaz USB. Para poner en funcionamiento la comunicación serial entre un microcontrolador PIC y los pines RPi del ELS, se consideran los pines de tierra (GND), GPIO 14 (TX) y GPIO 15 (RX). En este trabajo, el sistema informático para registrar el microcontrolador PIC (ADCES) es el RPi (ELS), cuyos pines de conexión para la comunicación USB y serial se muestran en la Figura 10.

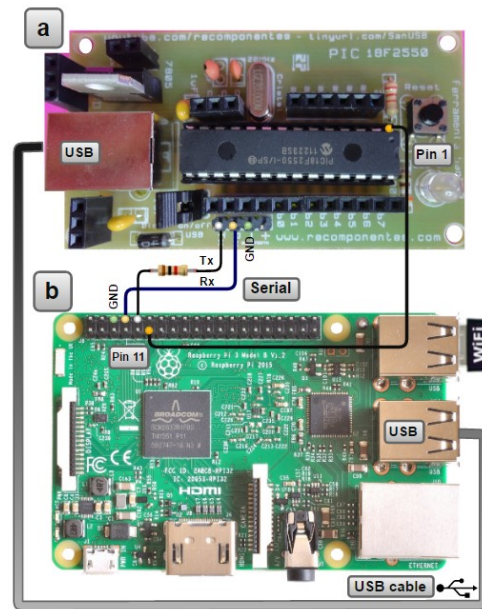


Figura 10. Conexión serial entre ADCES (tarjeta SanUSB) (a) y RPi (b)

Por lo tanto, se desarrolló una interfaz gráfica y un *software* de grabación para transferir directamente el firmware de la interfaz del sistema Linux embebido al PIC ADCES, utilizando el protocolo de comunicación del dispositivo de interfaz humana (HID, *Human Interface Device*, en inglés) a través del puerto USB. El ADCES propuesto es un sistema RISC embebido que permite actualizaciones en línea del firmware desde la nube. Los archivos pueden ser cargados remotamente para actualizar el firmware del ADCES (archivos .hex) y actualizar los scripts RPi (archivos .c, .py o .sh). Se requiere una conexión física entre el pin 11 del RPi y el pin 1 del ADCES para actualizar el ADCES. Los perfiles PaaS (PaaS A, B o N) ilustrados en la Figura 11 son libres y son generados automáticamente por el servidor en línea luego de registrado el perfil del usuario, donde es posible depurar y/o probar

una aplicación IoT. Así, cualquier usuario (A, B o N) puede crear y configurar un perfil IoT en la nube en tiempo real, y actualizar firmware y scripts a través de internet.

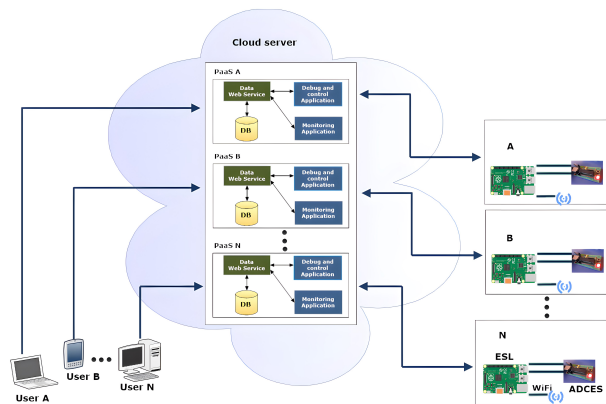


Figura 11. Perfil propuesto PaaS de IOTUS

El acceso a la página web <http://sanusb.org/iotus/sanusb.php>, ilustrado en la Figura 12, permite enviar y actualizar scripts en el ELS con extensiones de lenguaje en C (.c), Python (.py), scripts shell (.sh) y hexadecimal (.hex) en el ADCES a través del servidor. Debido a la latencia de la red, las direcciones de los archivos cargados al ELS y ADCES son variables y son renombradas con el perfil de usuario junto con la versión actualizada, que comienza con «0» y es incrementada en cada carga. Considerando la forma «ingenious», si el primer archivo es Python, será automáticamente renombrado como «ingenius0.py» e «ingenius1.py» será el nombre del script y, en consecuencia, de la nueva URL que se utiliza para descargar por el ELS.

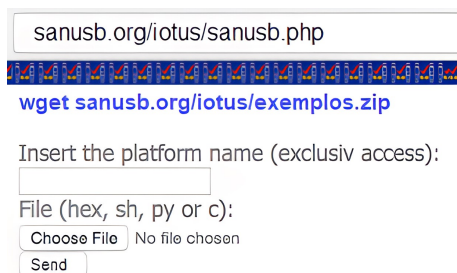


Figura 12. Actualización IoT de la página web

El IOTUS ha sido programado para cargar los archivos al servidor a través de la página <http://sanusb.org/iot/sanusb.php>, siguiendo estos pasos: 1. Ingreso del perfil del estudiante y por el ELS para descargar el archivo. 2. Buscar el firmware (.c, .py, .sh o .hex) que será enviado para la actualización automática. Pueden descargarse scripts de muestra desde sanusb.org/iotus/examples.zip. 3. El botón 'Submit' ejecuta la transferencia del archivo al servidor, el cual

es luego renombrado y transferido automáticamente al ELS. IOTUS está programado para operar de forma descentralizada y distribuida, permitiendo a múltiples usuarios y a múltiples comunicaciones simultáneas utilizar el mismo servidor en la nube. Luego de ingresar un nombre de perfil y enviar cualquier firmware, como se ilustra en la Figura 13, el `SloderInstall.sh` es generado automáticamente y queda disponible para ser descargado.

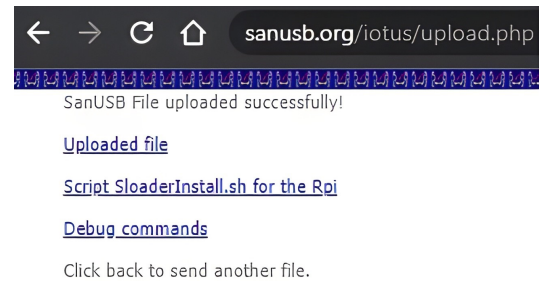


Figura 13. El sitio web con un enlace para descargar el script `SloderInstall.sh`

Por lo tanto, para actualizar el firmware (.c, .py, .sh, or .hex) a través de la nube, es necesario ejecutar previamente el script `SloderInstall.sh` en el ELS para revisar si hay nuevas actualizaciones de firmware en intervalos de tiempo definidos por el usuario, descargar el firmware actualizado y realizar una tarea específica de acuerdo con la extensión del archivo. En el arranque, el script `SloderInstall.sh` verifica si las librerías requeridas por este producto educativo están instaladas, y si no, las instala en el Rpi. Si se envía un archivo «hexadecimal», el ELS escribe el archivo en el ADCES a través de la conexión cableada entre el pin 11 del Rpi y el pin 1 del ADCES. Si se envía un archivo «c», el ELS compila el script y luego ejecuta el archivo compilado. En el caso de un script shell o de un archivo creado en Python, el ELS ejecuta directamente el archivo en segundo plano. La Figura 14 ilustra el diagrama de flujo de la operación de verificación del `SloderInstall.sh`.

Otra característica de la PaaS IOTUS propuesta es que para este producto educativo no es necesario desbloquear los puertos del enrutador y/o del cortafuegos. Como una aplicación para el IOTUS desarrollado, se utiliza un módulo fotovoltaico (PV, Photovoltaic, en inglés) que alimenta una carga de 50 W, con las siguientes especificaciones: Modelo YL95P-17b 2/3, Máxima Potencia 95 WP, Eficiencia 14,3 %, voltaje de circuito abierto (Voc, open circuit voltage) 22,5 V y corriente de cortocircuito (Isc, short circuit current) 5,59 A.

Las horas de sol para Fortaleza, en Brasil, donde fue instalado el sistema, son de 6 a. m. a 4 p. m. Las condiciones meteorológicas locales permiten una irradiación solar estable desde las 8 a. m. hasta las 2 p. m. (entre 800 y 1250 W/m²). Sin embargo, en este período

se midieron altas temperaturas del módulo PV, lo cual reduce la eficiencia de dicho módulo. Utilizando como ejemplo datos de enero, la Figura 15 muestra que la temperatura del módulo PV puede alcanzar valores desde 49 °C a las 8 a. m. hasta 62 °C al mediodía [31].

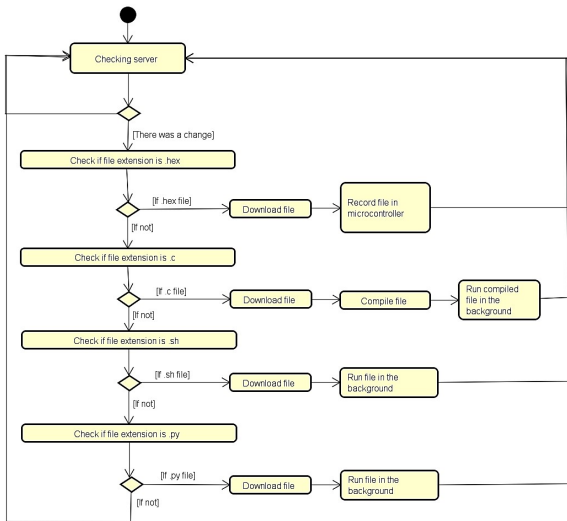


Figura 14. Diagrama de flujo de la operación del SloaderInstall.sh

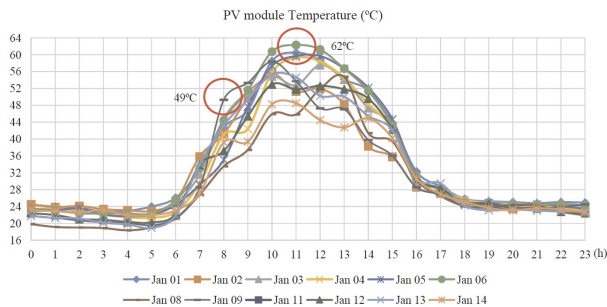


Figura 15. Temperatura del módulo PV en enero en Fortaleza-BR

Como una segunda aplicación, el sistema IOTUS ESP32 se valida usando un controlador lógico programable (PLC, *Programmable Logic Controller*, en inglés), con un canal AD de 12 bits de resolución [32]. La Figura 16 muestra la curva promedio mensual de la temperatura medida del módulo PV. Para valores de temperatura por encima de 52 °C, el error entre el monitoreo IoT propuesto usando ESP32 y el PLC aumenta aproximadamente 2 °C; para menores valores de temperatura, la medición sigue el valor de referencia. La Figura 17 muestra la correlación entre la temperatura del módulo PV medida por el monitoreo IoT y el valor de referencia del PLC para las mediciones promedio. Por medio de la raíz cuadrada del error cuadrático medio (RMSE, *Root Mean Square Error*, en inglés), la correlación es 0,9989.

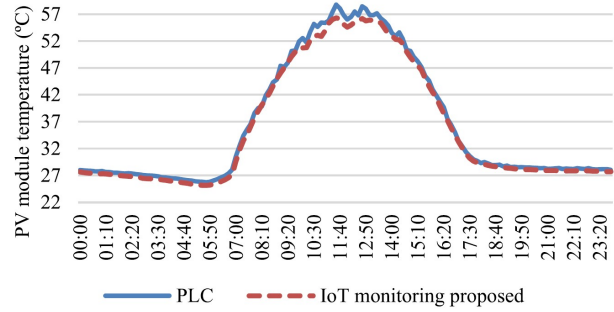


Figura 16. Temperaturas del módulo PV medidas por el monitoreo IoT propuesto y por el PLC

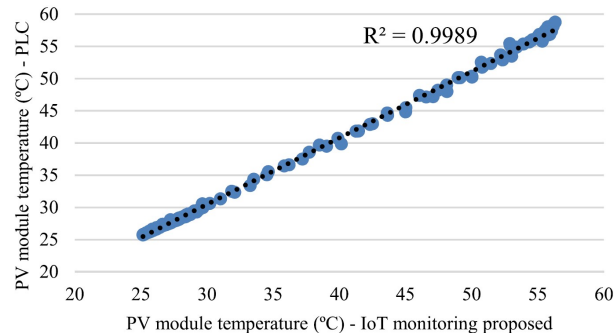


Figura 17. Correlación entre la temperatura del módulo PV medida por el monitoreo IoT propuesto y el valor de referencia del PLC

4. Conclusiones

Este trabajo describió el diseño y desarrollo de cuatro productos educativos en una plataforma en línea para el aprendizaje a distancia sobre microcontroladores e IoT; dichos productos fueron probados, validados y funcionaron de acuerdo con el diseño. Específicamente, el producto educativo propuesto ESP UPDATE permite la actualización de códigos de los microcontroladores ESP32 y ESP8266 en la nube, y es posible usarlo a través de internet en cualquier lugar y sobre cualquier dispositivo (computadora, teléfono inteligente), y esto permite aplicaciones virtuales de laboratorio de microcontroladores. El producto educativo ESPGPIO para ESP32 y ESP8266 y el RPI GPIO para Rpi permiten aplicaciones desde una computadora portátil o teléfono inteligente para controlar la activación de electrodomésticos a través de internet intuitiva e interactivamente.

IOTUS permite la grabación en línea de ADCES desarrollados con un microcontrolador PIC y la actualización de firmware en un sistema embebido Linux basado en un Raspberry Pi (ELS). El uso de *software* de código abierto y multiplataforma (Linux, Windows®, y Mac OSX) para enseñanza en línea permite mayor interacción y accesibilidad del usuario debido a

la posibilidad de libre distribución. Otra ventaja es que no es necesario desbloquear los puertos del enrutador y/o del cortafuegos cuando se utiliza el servicio de nube.

Finalmente, este artículo también presentó las principales características y ventajas de los microcontroladores en la plataforma en línea: bajo costo y el uso de *hardware* y *software* accesible. Rpi simplificó el procesamiento y proporcionó un circuito portátil teniendo Linux como sistema operativo y C como lenguaje de programación. El desarrollo de programas para las diferentes funcionalidades demostró la posibilidad de construir una solución efectiva y de bajo costo, donde el usuario puede agregar funcionalidades y configuraciones de acuerdo con sus necesidades. Adicionalmente, los productos educativos propuestos eliminan la necesidad de instalar en la computadora *software* para acceder al servidor de la plataforma educativa a distancia.

Referencias

- [1] C. Y. Yeh, Y. M. Cheng, and S. J. Lou, "An internet of things (IoT) maker curriculum for primary school students: Develop and evaluate," *International Journal of Information and Education Technology*, vol. 10, no. 12, pp. 889–896, 2020. [Online]. Available: <https://doi.org/10.18178/ijiet.2020.10.12.1475>
- [2] P. Jacko, M. Bereš, I. Kováčová, J. Molnár, T. Vince, J. Dziak, B. Fecko, Š. Gans, and D. Kováč, "Remote IoT Education Laboratory for Microcontrollers Based on the STM32 Chips," *Sensors*, vol. 22, no. 4, 2022. [Online]. Available: <https://doi.org/10.3390/s22041440>
- [3] D. Sarpong, G. Ofori, D. Botchie, and F. Clear, "Do-it-yourself (DiY) science: The proliferation, relevance and concerns," *Technological Forecasting and Social Change*, vol. 158, no. May, 2020. [Online]. Available: <https://doi.org/10.1016/j.techfore.2020.120127>
- [4] D. De Roeck, K. Slegers, J. Criel, M. Godon, L. Claeys, K. Kilpi, and A. Jacobs, "I would DiYSE for it! A manifesto for do-it-yourself internet-of-things creation," *NordiCHI 2012: Making Sense Through Design - Proceedings of the 7th Nordic Conference on Human-Computer Interaction*, pp. 170–179, 2012. [Online]. Available: <https://doi.org/10.1145/2399016.2399044>
- [5] J. L. Victor, S. C. Jucá, R. I. Pereira, P. C. Carvalho, and L. M. Fernández-Ramírez, "IoT monitoring systems applied to photovoltaic generation: The relevance for increasing decentralized plants," *Renewable Energy and Power Quality Journal*, vol. 17, no. September, pp. 536–545, 2019. [Online]. Available: <https://doi.org/10.24084/repqj17.368>
- [6] R. I. Pereira, M. M. Camboim, A. W. Villarim, C. P. Souza, S. C. Jucá, and P. C. Carvalho, "On harvesting residual thermal energy from photovoltaic module back surface," *AEU - International Journal of Electronics and Communications*, vol. 111, p. 152878, 2019. [Online]. Available: <https://doi.org/10.1016/j.aeue.2019.152878>
- [7] A. Morris and N. Lessio, "Deriving Privacy and Security Considerations for CORE," *Internet of Things and Cloud-based Services Conference*, vol. 1, no. 1, pp. 2–11, 2018. [Online]. Available: <https://doi.org/10.1145/3267357.3267363>
- [8] D. Mijailovic, A. Dordevic, M. Stefanovic, D. Vidojevic, A. Gazizulina, and D. Projovic, "A Cloud-Based with Microcontroller Platforms System Designed to Educate Students within Digitalization and the Industry 4.0 Paradigm," *Sustainability*, vol. 13, no. 22, 2021. [Online]. Available: <https://doi.org/10.3390/su132212396>
- [9] RaspberryPi Foundation, "Raspberrypi." [Online]. Available: <https://bit.ly/3tXNSUY>
- [10] Red Hat, "Red Hat Enterprise Linux: Manual de referencia." [Online]. Available: <https://bit.ly/3xSVdpV>
- [11] Red Hat-, "Red Hat SSH." [Online]. Available: <https://bit.ly/3NpD2OI>
- [12] Wiringpi, "Wiring Pi." [Online]. Available: <https://bit.ly/39QkERe>
- [13] GCC, "GCC online documentation." [Online]. Available: <https://bit.ly/3OioUFb>
- [14] CURL, "Command line tool and library for transferring data with URLs." [Online]. Available: <https://curl.se/>
- [15] ESP, "ESP32 Series Datasheet," *Espressif Systems*, pp. 1–65, 2021. [Online]. Available: <https://bit.ly/2E6qqIt>
- [16] ESP8266EX, "ESP8266EX Datasheet," *Espressif Systems*, p. 31, 2020. [Online]. Available: <https://bit.ly/2E6qqIt>
- [17] M. Technology, "USB Microcontrollers with nanoWatt Technology," *Technology*, 2006. [Online]. Available: <https://bit.ly/3OEOKpk>
- [18] A. Kevin, "That ' Internet of Things ' Thing," *RFiD Journal*, p. 4986, 2010. [Online]. Available: <https://bit.ly/3OdZmvI>

- [19] E. Inga, J. Inga, and A. Ortega, “Novel approach sizing and routing of wireless sensor networks for applications in smart cities,” *Sensors*, vol. 21, no. 14, pp. 1–17, 2021. [Online]. Available: <https://doi.org/10.3390/s21144692>
- [20] A. Peralta, E. Inga, and R. Hincapié, “Optimal Scalability of FiWi Networks Based on Multistage Stochastic Programming and Policies,” *Journal of Optical Communications and Networking*, vol. 9, no. 12, p. 1172, 2017. [Online]. Available: <https://bit.ly/3zZJicE>
- [21] H. Tabrizchi and M. Kuchaki Rafsanjani, *A survey on security challenges in cloud computing: issues, threats, and solutions*. Springer US, 2020, vol. 76, no. 12. [Online]. Available: <https://doi.org/10.1007/s11227-020-03213-1>
- [22] E. Inga, M. Campaña, R. Hincapié, and O. Moscoso-zea, “Optimal Deployment of FiWi Networks Using Heuristic Method for Integration Microgrids with Smart Metering,” *Sensors*, vol. 18, no. 8, pp. 1–21, 2018. [Online]. Available: <https://doi.org/10.3390/s18082724>
- [23] Firebase, “Firebase.” [Online]. Available: <https://bit.ly/3blnhLa>
- [24] S. E. Ooi, R. Beuran, and Y. Tan, “Secure iot development: A maker’s perspective,” *2021 IEEE International Conference on Omni-Layer Intelligent Systems, COINS 2021*, 2021. [Online]. Available: <https://doi.org/10.1109/COINS51742.2021.9524205>
- [25] Github, “SanUSB / EspCloudUpdate.” [Online]. Available: <https://bit.ly/3Nj8azh>
- [26] J. I. Vega Luna, F. J. Sánchez-Rangel, G. Salgado-Guzmán, J. F. Cosme-Aceves, V. N. Tapia-Vargas, and M. A. Lagos-Acosta, “Red de monitorización para automatizar el sistema de enfriamiento de un centro de datos,” *Ingenius, Revista de Ciencia y Tecnología*, no. 24, pp. 87–96, 2020. [Online]. Available: <https://doi.org/10.17163/ings.n24.2020.09>
- [27] S. Jucá and R. Pereira, “Aplicações práticas de sistemas embarcados Linux utilizando Raspberry Pi,” *PoD Editoria*, 2018. [Online]. Available: <https://bit.ly/3negu8Q>
- [28] Github, “SanUSB / IOTUS.” [Online]. Available: <https://bit.ly/3zWHxgx>
- [29] S. C. Jucá, P. C. Carvalho, and F. T. Brito, “A low cost concept for data acquisition systems applied to decentralized renewable energy plants,” *Sensors*, vol. 11, no. 1, pp. 743–756, 2011. [Online]. Available: <https://doi.org/10.3390/s110100743>
- [30] I. M. Dupont, P. C. Carvalho, S. C. Jucá, and J. S. Neto, “Novel methodology for detecting non-ideal operating conditions for grid-connected photovoltaic plants using Internet of Things architecture,” *Energy Conversion and Management*, vol. 200, no. September, p. 112078, 2019. [Online]. Available: <https://doi.org/10.1016/j.enconman.2019.112078>
- [31] R. Pereira, I. Dupont, P. Carvalho, and S. Jucá, “IoT embedded linux system based on Raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant,” *Measurement Journal*, vol. 114, pp. 286–297, 2018. [Online]. Available: <https://doi.org/10.1016/j.measurement.2017.09.033>
- [32] R. Pereira, S. Jucá, and P. Carvalho, “IoT embedded systems network and sensors signal conditioning applied to decentralized photovoltaic plants,” *Measurement*, vol. 142, pp. 195–212, 2019. [Online]. Available: <https://doi.org/10.1016/j.measurement.2019.04.085>