



PLATFORM FOR DISTANCE LEARNING OF MICROCONTROLLERS AND INTERNET OF THINGS

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

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

Received: 06-06-2022, Received after review: 14-06-2022, Accepted: 21-06-2022, Published: 01-07-2022

Abstract

Due to the increasing technological development in embedded systems and the Internet of Things (IoT), devices based on microcontrollers are 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 an Internet-connected computer system. 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, allowing their free online distribution and can be accessed from a cloud server. The open-source and multiplatform (Linux, Windows®, and Mac OSX) approach allows more significant user interaction and accessibility. The online educational products make possible the programming of ESP32 firmware remotely via OTA (over the air) and Linux embedded systems based on Raspberry Pi (Rpi), enabling virtual microcontroller laboratory applications. In addition, online educational products allow the manipulation of GPIO pins via the Internet through a graphical interface of the ESP32 and ESP8266 microcontrollers and the Rpi. In this context, the proposed online platform, running on a cloud server, was tested, and the four online educational products of distance learning and actual application of microcontrollers and the Internet of Things have been validated and worked as designed.

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

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 online para la enseñanza de microcontroladores y IoT. Estos productos educativos están basados en software de código abierto, lo que permite su distribución gratuita en línea y se puede acceder 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 de ESP32 de forma remota a través de OTA (over the air) y de sistemas embebidos Linux basados en Raspberry Pi (Rpi). Además, los productos educativos en línea permiten manipular los pines a través de la interfaz gráfica de los microcontroladores ESP32 y ESP8266, así como del Rpi. Como resultado, se prueba la plataforma en línea propuesta, ejecutada 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 se encuentran operativos tal como fueron diseñados.

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

¹ Electronics, Federal Institute of Alagoas, Alagoas, Brazil. Corresponding author ✉: 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. Introduction

The COVID-19 pandemic demonstrated that the teaching-learning process in educational institutions was hampered by the lack of and need for face-to-face classes. Several strategies are implemented for schools and universities to reduce the negative impact on students' learning and continuity of education [1,2]. Adopting distance practices has become a crucial part of teaching, especially in Engineering, Science, and Technology courses [3]. Thus, this paper describes a didactic tool that allows the implementation of online practices of microcontrollers and the Internet of things.

Due to the increasing technological development in the areas of embedded systems and the Internet of things (IoT), microcontroller-based devices are increasingly applied in various areas of knowledge such as motor control, security systems, elevator control systems [4], power generation monitoring [5] and in applied scientific research [6]. Therefore, teaching embedded systems, microcontrollers, and IoT topics becomes more and more relevant in Science and Engineering courses. Since online educational products based on microcontrollers and IoT are used over the Internet, remote teaching practices of these teaching resources are possible. In this way, online educational platforms and products are considered remote laboratories since students can access physical devices from anywhere as long as they have a computer system with Internet access [7,8].

1.1. Related Works

The single-board computer, called Raspberry Pi (Rpi), used in the distance education teaching platform proposed in this work, was developed by the Raspberry Pi Foundation to help children and adults learn in the computational area. It is a low-cost portable computer based on free software and interfaces for various peripherals. Other features are 1 GB RAM shared with a GPU (Graphics Processing Unit), VideoCore IV 3D graphics core, 40 general-purpose input/output (GPIO) pins, four ports with USB 2.0 interface for connecting keyboard and mouse used in desktop computers, HDMI (High Definition Multimedia Interface) port, Ethernet port with RJ-45 interface, 56 micro SD (Secure Digital) card interface, camera and monitor interface, as well as 5V power supply and audio output [9].

For developing online educational products based on RPi, a set of libraries and packages are used to ensure communication between all system components. Then, to access Rpi, SSH (Secure Shell) and VNC (Virtual Network Computing) protocols are tested to allow console and graphical environment access, respectively. SSH is a protocol that enables secure communication between two systems using a client/server architecture

and allows users to connect to a remote host. Unlike other remote protocols such as FTP (File Transfer Protocol) or Telnet, SSH encrypts the login session, making it virtually impossible for intruders to collect unencrypted passwords [10].

To ensure interconnection between computers running Microsoft Windows and Linux operating systems and to share files and folders, the Samba package, an open-source implementation of the SMB (Server Message Block) protocol, was installed and configured [11].

For Rpi's C language programs, the WiringPi C library was used, which can be used to program and configure the GPIO pins [12]. Thus, to compile language programs written in C, the GCC (GNU Compiler Collection) compiler is used, accessible, integrated distribution of compilers for programming languages such as C, C++, C#, and Java [13]. The CURL library was also used as an open-source command-line tool for URL syntax construction and data transfer to the cloud [14].

The ESP32 IoT microcontroller was released by the Chinese company Espressif Systems, a manufacturer of embedded wifi antennas and IoT modules with free software and hardware [15]. ESP32 is a low-power, dual-core, dual-mode 32-bit wifi/Bluetooth microcontroller (MCU). The ESP32 MCU has a frequency of 240 MHz and a processing power of 600 DMIPS (Dhrystone million instructions per second).

On the other hand, compared to the latest model, the 32-bit ESP8266 NodeMCU IoT microcontroller with wifi antenna is an earlier and more simplified version, also released by Espressif [16].

The PIC microcontroller (PIC - Peripheral Interface Controller) is an integrated circuit that contains all the circuitry necessary to realize a complete programmable digital system in a single device. This teaching platform also uses the 8-bit PIC18F2550 microcontroller produced by Microchip Technology. The analog-to-digital converter (ADC) module of the PIC18F2550 has ten inputs and is capable of measuring the analog voltage on each channel and converting the measured voltage into a 10-bit digital value [17]. In this context, IoT technology enables the communication between devices without human intervention [18,19] and understands that each device is an autonomous device connected to the Internet. Such devices can interact and communicate automatically, maximizing modularization, comfort, convenience, security, and energy savings, [20]. A database is typically used to store data in the cloud to implement IoT systems. For this purpose, Cloud Computing provides access to a shared pool of configurable computing resources, such as non-relational databases, without needing in-depth knowledge of management technologies [21,22]. The cloud model simplifies the installation, operation, and maintenance of information systems, increasing system efficiency and reliability and reducing costs.

Cloud systems can be classified according to the development or service model. Cloud development models are: public, private, community, or hybrid; Cloud service models are: Platform as a Service (PaaS), software as a Service (SaaS), or Infrastructure as a Service (IaaS). In this work, PaaS is used, which consists of the use of tools and resources to provide services to end-users who are students. Since end-users share information on the same server in the cloud, data privacy and confidentiality are the primary concerns. Thus, the educational products of the learning platform make it possible for the stored data to be secure and encrypted, allowing only the student owner to manage it.

Thus, to write and read variable data in the cloud, the proposed distance learning platform uses Firebase Realtime Database, which is a NoSQL database, i.e., it does not have as standard the system of tables and relationships between data and where data are stored as JSON objects, [23]. In addition, the online didactic platform for distance learning of microcontrollers allows real-time remote use by multiple users.

2. Materials and methods

2.1. Problem and Methodology

IoT devices increasingly require rapid intervention to update libraries and functionality and maintain the security of online environments. To this end, quickly updating the firmware of these IoT devices is an effective way to ensure data security. Thus, the proposed distance learning instructional platform uses IoT firmware updates via Over-The-Air (OTA). The OTA firmware update is performed via HTTP, which allows downloading a binary file from a cloud server to update the firmware of the microcontrollers remotely. OTA update over the Internet is typically implemented in two ways: (i) If a newer firmware version is available, the device periodically queries the server containing the compiled binary file (pull) or another online activation service, such as Firebase. ii) the device receives a notification of a new firmware version via the cloud (push), e.g., via MQTT, and performs the update. The server stores the latest firmware version in binary, and then the time-varying URL of the binary file, which is more secure and flexible than a predefined fixed address, is sent to the microcontroller for download [24]. In this topic, the online educational products of the microcontroller and Internet of things distance learning didactic platform are presented.

3. Results and discussion

The paper demonstrates an educational product to update ESP32 and ESP266 microcontroller codes with the same generic firmware using

<http://sanusb.org/espupdate>. The free version of Firebase (Google's JSON object database) has been implemented to generate an asynchronous update trigger in the cloud for .bin files, as illustrated in Figure 1.

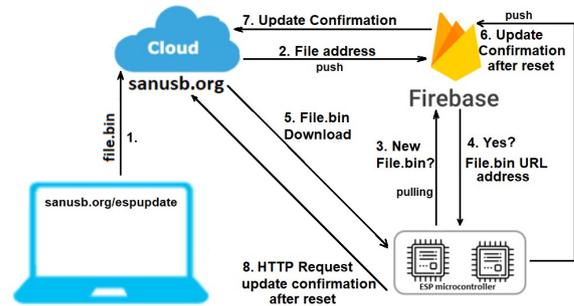


Figure 1. Illustration of the operation of the Espupdate environment

The update (OTA) transmits the compiled binary .bin files to a remote computer over the Internet, enabling microcontroller virtual lab applications. To perform this firmware update in the cloud, users must type in the firmware only the SSID, password, and the same name as the profile entered on the website <http://sanusb.org/espupdate> and by uploading the .bin to the website, the firmware of the ESP microcontrollers is updated remotely over the Internet. The user profile name entered at sanusb.org/espupdate can be alphanumeric.

It is possible to test this webcast educational product on different networks by accessing the website <http://sanusb.org/espupdate/> through the lab or home network and the ESP32 or ESP8266 microcontroller attached to the smartphone connected to the 4G mobile network, or vice versa.

It is worth considering that, through tests performed, .bin files compiled with the same name and sequential downloads for the update in the cloud, it may happen that the last .bin file sent for the update is not downloaded by the ESP microcontroller but a previously uploaded .bin File, since they have the same name and download URL. For this reason, in this project, the name of the .bin files uploaded to the site have names based on the time of upload, consequently, the download URL as well; this prevents a previously uploaded file or another file with the same name from being downloaded, generating a unique and unmatched URL. In this case, for the version of the .bin file name and download URL, the following are used: year, month, day, time, and the cyclic order of upload [25].

Once the .bin file update is completed and the microcontroller automatically restarts, the new verification code is sent to http://sanusb.org/espupdate/*Profile/ to confirm the update on the website. It is important to note that

to use EspUpdate, it is not necessary to open ports on the router or enable firewall permissions.

3.1. ESPGpio Educational Product

The educational product, called ESPGpio, allows changing the logic level of the ESP32 and ESP266 microcontroller pins that can be programmed with open source firmware [26], illustrated in Figure 2 through the graphical access available at sanusb.org/esp. The user profile name entered at sanusb.org/espgpio can be alphanumeric.

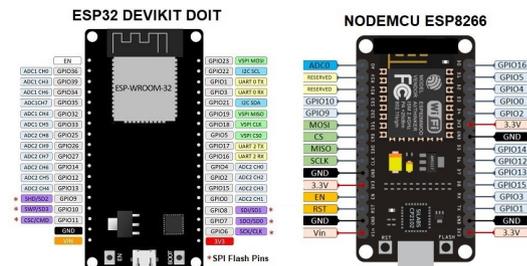


Figure 2. Illustration of ESP32 and ESP8266 GPIO pins

Figure 3 shows an illustration of the automatic operation of the ESPGPIO environment. In this case, a firebase cell is used to share between the microcontroller and the user’s graphical environment. An even integer is represented by the sum of 2. It is raised to the power of the number of activated pins.

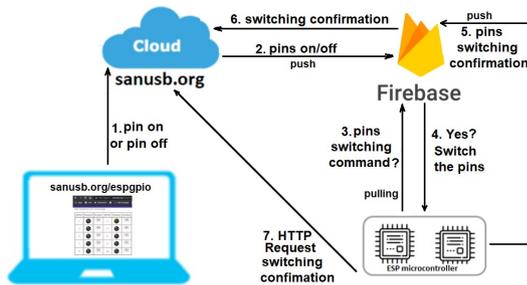


Figure 3. Illustration of the GPIO pins of an ESP32 or ESP8266

The graphical environment generated at sanusb.org/espgpio, illustrated in Figure 4, has 30 GPIO pins that the user must select to load switching through relays or electronic devices such as LEDs. It is important to note that if there is a power failure or voltage drop, the state of the last pins when the voltage drop is restored returns to normal, as the state of the pins is updated and queried in Firebase, which is a persistent database.

After completing the change of the pin states, the new pin state is indicated with green color in the graphical environment http://sanusb.org/espgpio/*Profile*/, if it is at a high logic level, as illustrated in Figure 4, where *Profile* is the profile given by the user.



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

Figure 4. Illustration of the graphical environment of the ESP32 and ESP8266 GPIO pins

It is important to note that it is recommended to avoid using Gpio pin one on the ESP8266 Nodemcu, which is the Tx pin, as it prevents serial communication of program debugging by the serial monitor, and Gpio pins 06 to 11 on the ESP32 Devkit, as they are used for flash recording.

3.2. Educational product RPI GPIO

The educational product RPI GPIO is similar to ESPGpio but is used to change the state of the logic pins on the Raspberry Pi board. Figure 5 shows the illustration of the staples of a RaspberryPi.

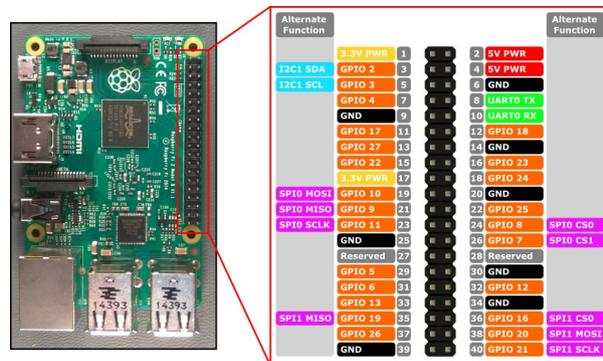


Figure 5. Illustration of the GPIO pins of a Raspberry Pi

After creating the profile at sanusb.org/gpio, a page is generated with links to pages to access the Rpi pins and firmware buttons for the extraction application and the insertion application in bash Shell language, as illustrated in Figure 6.



Figure 6. Link to pages with pin buttons and programs for Rpi

For both the complete application and the push application, it is necessary, after downloading the programs, to grant permission in the terminal as superuser (sudo su) and execute permission with the commands `chmod 755 SgpioInstall.sh`, for the pull application and `chmod 755 Sgpiopushmqtt.sh`, for push application [27]. Clicking on the profile link generates a page with buttons representing each of the I/O pins of a Rpi, as illustrated in Figure 7.

In the pull case, clicking on the button for a given physical pin sends a command to a file contained in the online profile periodically queried by the `Sgpiointall.sh` script that is operating on the Rpi. In the case of push MQTT, clicking the button sends a publish command to the broker `mqtt.eclipse.org` on the topic with the same name as the user's profile; in the case of the example, lease and downloads this command to the Rpi that is operating the `Sgiopushmqtt.sh` script with subscription (subscribe) to the same topic.

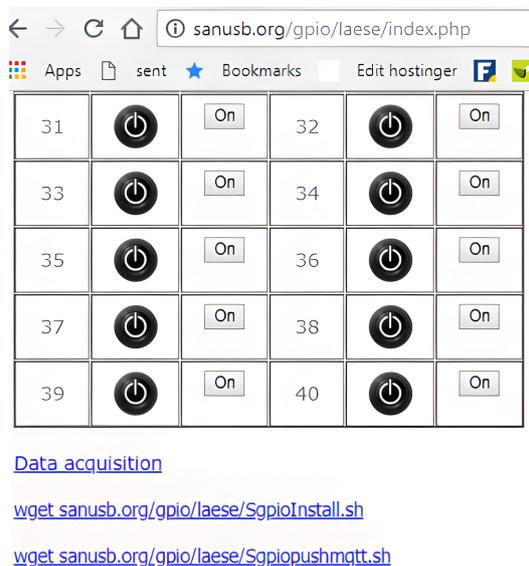


Figure 7. Illustration of the graphical environment of Raspberry Pi GPIO pins

With the RPI GPIO environment, it is possible to create an IoT application using a laptop or smartphone to, for example, control the activation of household appliances via the Internet, such as a fan via a relay, as illustrated in Figure 8. In this case, a Raspberry Pi Zero, also accessible via the Internet, is used. As can be seen, to use the `sanusb.org/gpio` environment, it is

necessary to create a profile initially. After completing the profile, a link is generated to access a page with buttons, and each button corresponds to a pin of the Raspberry Pi IoT device [27].

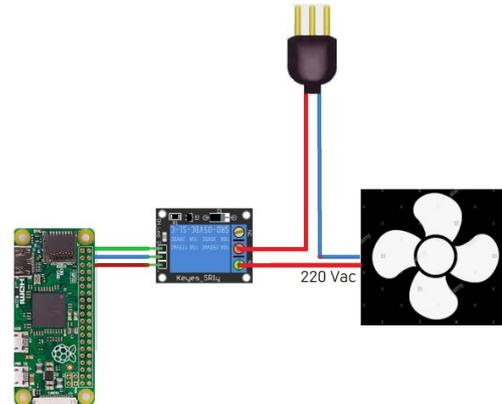


Figure 8. Illustration of a circuit for IoT operation of a fan with Raspberry Pi

3.3. IOTUS educational product

The educational product IOTUS (IoT update system) is a PaaS, i.e., a platform as a service in which the user can create a profile and update scripts in C, python, or shell language on the Rpi-based Linux embedded system and also update the firmware in hexadecimal of the PIC family microcontroller through the USB port of the RPI [28], as illustrated in Figure 9.

Thus, IOTUS consists of three main parts, namely: 1) Raspberry Pi-based Linux embedded system (ELS), 2) Analog-to-digital converter embedded system (ADCES), which uses a SanUSB microcontroller, based on a free software and hardware tool with the PIC18Fxx5X family, including the native USB interface; and 3) WEB page for automatic real-time update (upload) of scripts.

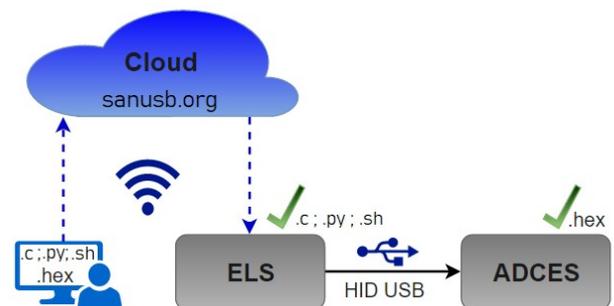


Figure 9. Remote firmware and script upgrade

ADCES consists of the open-source development tool SanUSB illustrated in Figure 9. SanUSB is composed of open-source software and hardware of the PIC18Fxx5x family with a native USB interface. This

free tool is efficient in project development because removing the microcontroller to update the firmware is unnecessary, unlike other development systems that typically use specific recording hardware and require a wired connection. In addition, the SanUSB tool is cross-platform, i.e., it can be used on Windows®, Mac OSX, and Linux, as well as being plug and play, i.e., it is automatically recognized by the operating systems without the need to install any driver [29].

Thus, this tool allows the compilation, recording, and emulation of a program to be performed quickly and efficiently by connecting the microcontroller to a computer system via USB. Its application is an RPi. The ELS RPi-based computer system can directly implement electronic designs, mainly due to its general-purpose input/output (I/O) pins (GPIO) with support for digital sensors, actuators, and I²C and SPI protocols to perform communications with peripherals [30]. The communication variety allows the RPi to communicate with a broader range of peripherals, and the Wiring Pi pin access library, written in C, facilitates programming and configuring the GPIO pins through a command-line utility "gpio." This work has two ways of communication between ADCES and RPi: serial communication and USB interface. To realize serial communication between a PIC microcontroller and the RPi pins of the ELS, the ground (GND), GPIO 14 (TX), and GPIO 15 (RX) pins are used. In this work, the computer system used to register the PIC microcontroller (ADCES) is the RPi (ELS), whose connection pins for USB and serial communication are shown in Figure 10.

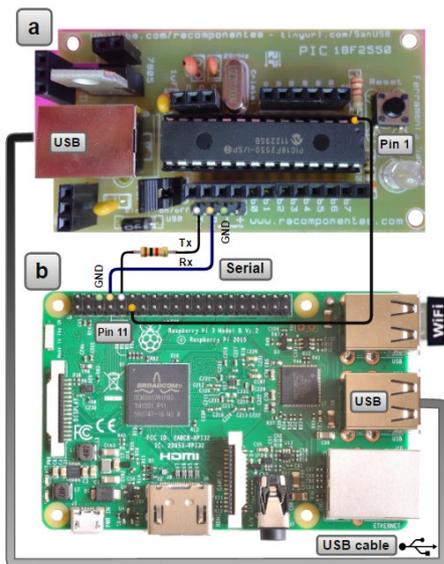


Figure 10. Serial connection between ADCES (SanUSB board) (a) and RPi (b)

Therefore, a graphical interface and recording software was developed to directly transfer the embedded Linux system interface firmware to the ADCES PIC

using the Human Interface Device (HID) communication protocol through the USB port. The proposed ADCES is an embedded RISC system and allows on-line firmware updates from the cloud. Files can be uploaded remotely to update ADCES firmware (.hex files) and update RPi scripts (.c, .py or .sh files). A physical connection between pin 11 of the ADCES is required to update ADCES. The PaaS profiles (PaaS A, B, or N) illustrated in Figure 11 are free and automatically generated by the online server after registering the user profile, where it is possible to debug and/or test an IoT application. Thus, any user (A, B, or N) can create and configure an IoT profile in the cloud in real-time and update firmware and scripts over the Internet.

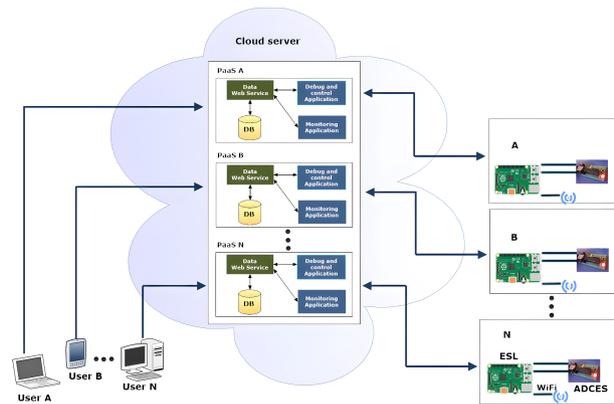


Figure 11. Proposed PaaS profile of IOTUS

Accessing the web page at <http://sanusb.org/iotus/sanusb.php>, illustrated in Figure 12, allows sending and updating scripts in ELS with language extensions in C (.c), Python (.py), Shell scripts (.sh), and hexadecimal (.hex) in ADCES through the server. Due to network latency, the address of the files uploaded to ELS and ADCES are variable and are renamed with the user's profile along with the updated version, which starts with "0" and at each upload is incremented. Considering the shape "ingenious," if the first file is Python, it will be automatically renamed to "ingenious0.py" and "ingenious1.py" will be the name of the script and, consequently, of the new URL that is used to download by ELS.

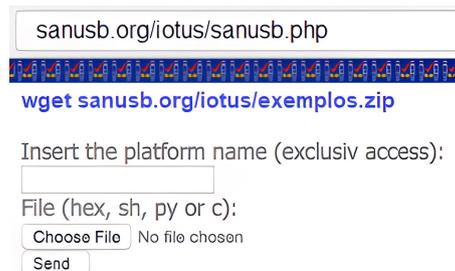


Figure 12. IoT scripts update the WEB page

IOTUS has been programmed to upload files to the server through the <http://sanusb.org/iot/sanusb.php> page by following these steps: 1. Insertion of the profile used by the student and by ELS to download the file; 2. Browse for the firmware (.c, .py, .sh or .hex) that will be sent for automatic update. Sample scripts are downloaded from sanusb.org/iotus/examples.zip. 3. The 'Submit' button executes the file transfer to the server, which is then renamed and automatically transferred to the ELS. IOTUS is programmed to operate decentralized and distributed, allowing multiple users and multiple simultaneous communications using the same cloud server. After entering a profile name and submitting any firmware, as illustrated in Figure 13, SloderInstall.sh is automatically generated and available for download.

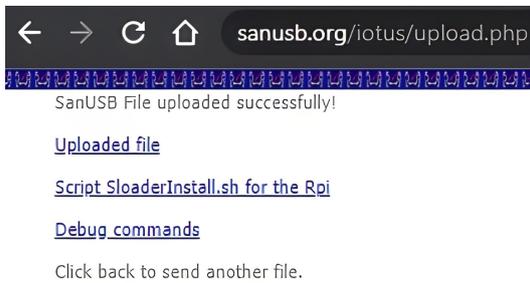


Figure 13. The website with a link to download the SloderInstall.sh script

Therefore, to update the firmware (.c, .py, .sh, or .hex) via the cloud, it is necessary to previously run the SloderInstall.sh script in ELS to check for new firmware updates at user-defined time intervals, download the updated firmware and perform a specific task according to the file extension. The SloderInstall.sh script checks at startup whether the libraries required for this educational product are installed and, if not, install them in Rpi. If a "hexadecimal" file is sent, ELS writes the file to ADCES over the cable connection between RPi pin 11 and ADCES pin 1. If a ".c" file is sent, ELS compiles the script and then executes the compiled file. In the case of a shell script or a file created in Python, ELS directly runs the file in the background. Figure 14 illustrates the flowchart of the SloderInstall.sh verification operation.

Another feature of the proposed IOTUS PaaS is that when using this online educational product, there is no need to unblock router and/or firewall ports.

As an application for the developed IOTUS, a Photovoltaic (PV) module supplying a 50 W load is used, with the following specifications: Model YL95P-17b 2/3, Maximum Power 95 WP, Efficiency 14.3%, open circuit voltage (Voc) 22.5 V and short circuit current (Isc) 5.59 A.

The sunshine hours for Fortaleza, in Brazil, where the system was implemented, are from 6 am to 4 pm.

Local meteorological conditions allow stable solar irradiance from 8 am to 2 pm (between 800 and 1250 W/m²). However, in this period high PV module temperatures are measured, which reduces the PV module efficiency. Using data from January as an example, Figure 15 shows that PV module temperature can reach values from 49 °C at 8 am to 62 °C at noon [31].

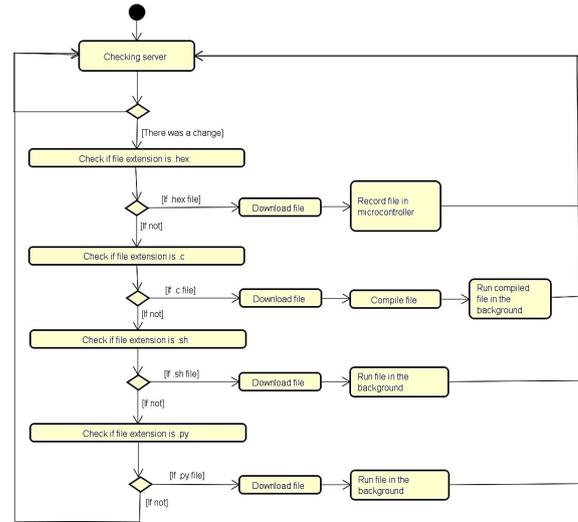


Figure 14. SloderInstall.sh operation flowchart

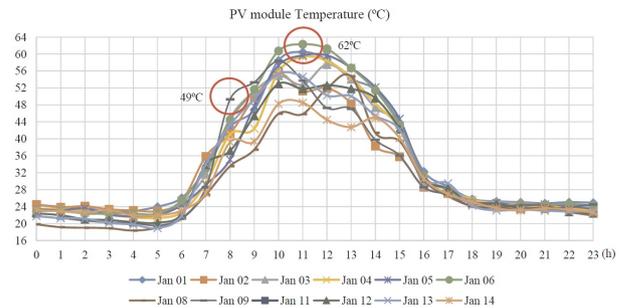


Figure 15. PV Temperature on January in Fortaleza-BR

As a second application, IOTUS ESP32 system is validated using a Programmable Logic Controller (PLC) with 12-bit resolution AD channel [32]. Figure 16 shows the monthly average curve of the PV module temperature measured. For temperature values above 52 °C, the error between the proposed IoT monitoring using ESP32 and the PLC increases, about 2 °C; for lower temperature values, the measurement follows the reference value. Figure 17 shows the correlation between the PV module temperature measured by the proposed IoT monitoring and PLC reference value for the average measurements. Using the Root Mean Square Error (RMSE), the correlation is about 0.9989.

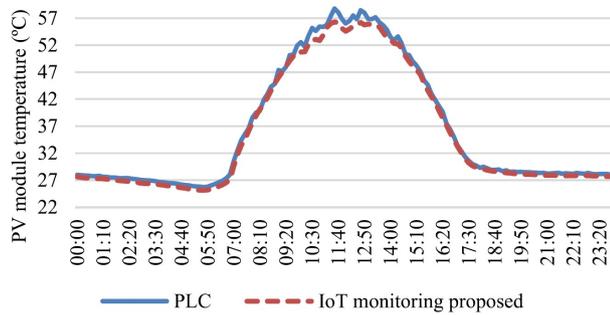


Figure 16. PV module temperature measured by the proposed IoT monitoring and PLC

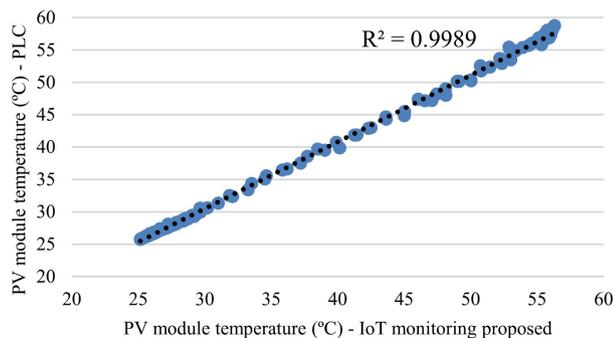


Figure 17. Correlation between PV module temperature measured by the proposed IoT monitoring and PLC reference value

4. Conclusions

This work described the design and development of four educational products in an online platform for distance learning of microcontrollers and IoT, which were tested, validated, and worked as designed. Specifically, the proposed ESP UPDATE educational product allows the update of the ESP32 and ESP266 microcontroller codes in the cloud, and it is possible to use it through the Internet anywhere and on any device (computer, smartphone), and this allows virtual microcontroller lab applications. The educational product ESPGPIO for ESP32 and ESP8266 and the RPI GPIO for Rpi allow IoT applications from a laptop or smartphone to control the activation of home appliances through the Internet intuitively and interactively.

IOTUS enables online recording of ADCES developed with a PIC microcontroller and firmware update on a Raspberry Pi-based embedded Linux system (ELS). Using open-source and cross-platform software (Linux, Windows®, and Mac OSX) for online teaching allows for more significant user interaction and accessibility due to the possibility of free distribution. Another advantage of the four proposed educational products is that there is no need to unblock ports and/or firewalls when using the cloud service.

Finally, this paper also presented the main features and advantages of the microcontrollers used in the online platform: low cost and the use of accessible hardware and software. The use of Rpi simplified the processing and provided a portable size circuit using Linux as the operating system and C as a programming language. The development of programs for the various functionalities demonstrated the possibility of building an effective and low-cost solution where the user can add functionalities and configurations according to his needs. In addition, the proposed educational products eliminate the need to install software on the computer that will access the distance education platform server.

References

- [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/3Oi0UFb>
- [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>