applied sciences

MDPI

Article

Developing a Cloud and IoT-Integrated Remote Laboratory to Enhance Education 4.0: An Approach for FPGA-Based Motor Control

Héctor A. Guerrero-Osuna [1], Fabián García-Vázquez [1], Salvador Ibarra-Delgado [2], Marcela E. Mata-Romero [3,*], Jesús Antonio Nava-Pintor [1], Gerardo Ornelas-Vargas [1], Rodrigo Castañeda-Miranda [1], Víktor I. Rodríguez-Abdalá [2] and Luis Octavio Solís-Sánchez [1]

[1] Posgrado en Ingeniería y Tecnología Aplicada, Unidad Académica de Ingeniería Eléctrica, Universidad Autónoma de Zacatecas, Zacatecas 98000, Zacatecas, Mexico; hectorguerreroo@uaz.edu.mx (H.A.G.-O.); fabian.garcia@uaz.edu.mx (F.G.-V.); jesus.nava@uaz.edu.mx (J.A.N.-P); ornelas@uaz.edu.mx (G.O.-V); rcastm@uaz.edu.mx (R.C.-M.); lsolis@uaz.edu.mx (L.O.S.-S.)

[2] Posgrado en Ingeniería para la Innovación Tecnológica, Unidad Académica de Ingeniería Eléctrica, Universidad Autónoma de Zacatecas, Zacatecas 98000, Zacatecas, Mexico; sibarra@uaz.edu.mx (S.I.-D.); abdala@uaz.edu.mx (V.I.R.-A.)

[3] Subdirección de Investigación, Centro de Enseñanza Técnica Industrial, C. Nueva Escocia 1885, Guadalajara 44638, Jalisco, Mexico

* Correspondence: mmata@ceti.mx; Tel.: +52-33-36429349 (ext. 310)

Abstract: Remote laboratories are essential in addressing access and quality challenges in technical education. They enable students from various locations to engage with real equipment, overcome geographic and economic constraints, and provide solutions during crises, such as pandemics, when in-person learning is limited. As a key element of Education 4.0, remote labs promote technical skill development, enhance engineering education, and support diverse learning approaches. This study presents a remote laboratory based on Field Programmable Gate Arrays (FPGAs), developed using a waterfall methodology integrating IoT and Cloud Computing technologies to facilitate close interaction between hardware and software. The lab focuses on controlling DC, servo, and stepper motors, allowing students to apply theoretical concepts such as digital signals, pulse-width modulation (PWM), and data representation in bits in a practical setting. The testing phase involved 50 robotics and mechatronics engineering students who participated in hands-on sessions for one month, followed by a structured survey evaluating their experience, interaction, and the educational relevance of the platform. The survey shows high student satisfaction, highlighting the platform's strengths and identifying areas for improvement. The results also underscore the system's potential to significantly enhance the educational experience in remote environments, aligning with the United Nations Sustainable Development Goals (SDGs).

Keywords: remote laboratory; digital education; internet of things; cloud computing; Education 4.0

check for updates

Citation: Guerrero-Osuna, H.A.; García-Vázquez, F.; Ibarra-Delgado, S.; Mata-Romero, M.E.; Nava-Pintor, J.A.; Ornelas-Vargas, G.; Castañeda-Miranda, R.; Rodríguez-Abdalá, V.I.; Solis-Sánchez, L.O. Developing a Cloud and IoT-Integrated Remote Laboratory to Enhance Education 4.0: An Approach for FPGA-Based Motor Control. *Appl. Sci.* **2024**, *14*, 10115. https://doi.org/10.3390/app142210115

Academic Editor: Linda Daniela

Received: 10 October 2024
Revised: 1 November 2024
Accepted: 3 November 2024
Published: 5 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the traditional Science, Technology, Engineering, and Mathematics (STEM) fields, physical laboratories help students transfer skills. In these labs, students interact with real equipment, gain practical skills, and deal with the challenges of real-world systems. However, the shift from in-person to online learning has led to remote or virtual laboratories [1,2], driven by the COVID-19 pandemic declared by the World Health Organization (WHO) in March 2020, which led to a quick shift to online work and education. This change required the rapid development of tools more suitable for online work environments than previously available ones. Furthermore, transitioning from in-person classrooms has reduced interactions between teaching assistants, classmates, and teachers [3–5].

Addressing these challenges is crucial in STEM, which relies on hands-on experiences for effective learning. Universities usually use science laboratories to encourage hands-on tasks and teach technical skills. However, building and maintaining high-quality scientific laboratories can be costly for universities in underdeveloped and developing countries [6,7].

Remote labs are environments in which physical equipment is controlled over the Internet. These labs enable users to conduct hands-on experiments using real equipment in different locations. This approach supports practical and authentic learning experiences, allowing students to manipulate variables, observe real-time results, and develop practical skills despite traditional laboratories' geographical and time constraints [8–10].

Education 4.0, driven by the fourth industrial revolution, represents a disruptive educational innovation that has gained relevance, especially during and after the COVID-19 pandemic. In the context of remote labs, Education 4.0 offers a key solution by allowing students to conduct practical experiments from any location using real, remotely controlled equipment. This facilitates experiential learning in a digital environment and responds to the need for inclusive and accessible education aligned with the United Nations (UN) Sustainable Development Goals (SDGs) [11]. Therefore, Education 4.0 improves the quality of remote learning and promotes a more personalized and flexible approach to teaching technical skills, which is essential for success in an increasingly digitalized world [12,13].

Integrating emerging technologies into university teaching as part of Education 4.0 must prioritize maintaining educational quality. It should also take advantage of the opportunities and provide better education access for students far from higher education institutions or affected by regional conflicts. The successful operation of remote laboratories relies on integrating technologies such as the Internet of Things (IoT), Cloud Computing (CC), and web-based platforms. This integration enables the remote control and monitoring of physical laboratory equipment, ensuring an interactive and seamless educational experience [14].

Incorporating new technologies has led to the adoption of remote laboratories. Field Programmable Gate Arrays (FPGAs) are commonly used in traditional engineering labs, where each student needs to install specialized digital circuit design software and connect an academic/physical FPGA board. This process can be expensive and difficult to manage on a large scale. The shift to remote laboratories for embedded systems with FPGAs has made it possible to create platforms that offer remote access to these devices, eliminating the need for local software installations and making experiential learning more accessible [15–17].

The typical architecture of a remote FPGA lab includes a main server that provides circuit design tools, system access management tools, FPGA cards, webcams to monitor the behavior of the designed circuits, and remote device servers that emulate the available peripherals. This approach not only makes FPGA training more accessible but also allows for modular and extensible resource management, improving the efficiency and effectiveness of engineering education [18,19].

This paper proposes a remote FPGA lab for motor control using the Basys 3 FPGA board, a Raspberry Pi 5, a cloud management system, and a web platform. The lab enables students to upload and run digital hardware designs remotely 24/7, enhancing experiential learning and managing their practice time. Through the web platform and Raspberry Pi, students can transmit control signals to the Basys 3, which operates a Direct Current (DC) motor, a servo motor, and a stepper motor. A live streaming system is also set up using a camera and the YouTube API, allowing students to observe the lab in real time. The proposed practices concentrate on digital motor control, covering key concepts such as digital signals, Pulse-Width modulation (PWM), and data representation in bits.

The contributions of this paper can be summarized as follows: a remote lab aims to provide equal access to technical learning resources, overcoming geographical and economic obstacles so that students from anywhere in the world can use real equipment and conduct experiments in real time. The incorporation of emerging technologies, such as IoT, CC, and web platforms, enables remote control and laboratory equipment monitoring to ensure an interactive and accessible educational experience, helping teach essential

technical skills and reducing barriers that limit access to quality infrastructure in regions with limited resources, as well as the assessment of students' perceptions regarding their learning experiences with the remote FPGA laboratory, specifically evaluating how effectively it facilitates equitable access to technical learning resources. While the initial plan was to implement the laboratory in Mexico, it has been designed with a global perspective to promote inclusive and fair access to educational resources, which is why the research question was: Is it a feasible idea to implement remote labs as a complement/replacement of an in-site lab?

The paper is organized as follows: Section 2 shows the existing works related to remote labs. Section 3 details the proposed methodology, and Section 4 explains how a remote FPGA lab is designed and implemented. Section 5 presents the proposed practices for motor control. The results are presented in Section 6, while Section 7 provides a discussion, and Section 8 discusses limitations and future research. The paper closes with Section 9, which summarizes the findings and presents the conclusions.

2. Related Works

The paper in [20] proposes a remote lab based on Arduino Uno and LM35 sensors to identify and adjust proportional, integral, and derivative controllers. It integrates simulation with practical implementation to enhance real-time interaction. Similarly, the work in [21] describes an IoT-based remote laboratory for microscopes, employing tools like TeamViewer and Arduino. This setup helped overcome equipment limitations during the pandemic and improved time efficiency.

In [22], the authors discuss virtual and remote labs that utilize simulators like MATLAB/Simulink and remote lab hardware, enabling students to conduct hands-on experiments conveniently and flexibly. Research in [23] details an IoT-based remote lab for sensor experiments, employing an STM32 microcontroller and an ESP12E Wi-Fi module to enable remote access and enhance understanding of industrial applications. Authors in [24] introduce a cost-effective platform for remote labs based on ESP32 and Node-RED, allowing students to conduct hands-on experiments in photovoltaics and operational amplifiers. The paper in [25] examines the use of Tinkercad to support online lab work in microprocessor system design. The study observed a 5.4% increase in the quality of knowledge acquired by students using Tinkercad compared to traditional methods, demonstrating the effectiveness of simulation tools in distance learning.

In [26], a microservices-based system enables users to upload, compile, and load designs into FPGAs through a web interface. Docker is used for portability, while Redis and Postgres are used for data storage. The system demonstrates high effectiveness, with a 95% success rate in programming the FPGAs. The study in [27] also details the development of a hardware development kit for a remote training platform on FPGA devices. This kit includes a Raspberry Pi, a display, a camera, and LEDs, allowing students to perform engineering experiments remotely and facilitating hands-on learning in FPGA design. The work in [28] presents practical sessions for teaching and learning digital electronics using an FPGA-based development platform. This platform is accessible both in the campus lab and online through a remote lab, allowing students to design and test specific modules.

Research in [29] discusses a system for remote design and testing of electronic circuits and devices with FPGAs during the COVID-19 pandemic. The system uses the IoT concept for remote visual inspection of the board and connected devices, making it easier for students to learn during periods of confinement. Similarly, the paper in [30] details the LabEAD project, a remote laboratory in Brazil that integrates students' smartphones with IoT devices to conduct digital electronics experiments. This approach has shown high acceptance and preference among students. The authors in [16] present VirtLAB, a low-cost platform for laboratory experiments in electronics. VirtLAB combines programmable prototyping circuitry based on a microcontroller and an FPGA with typical laboratory measurement instruments, allowing analog and digital electronics experiments to be performed in an accessible and economical way.

The work in [31] discusses using FPGA technologies to create a remote laboratory for signal acquisition and design verification. It involves implementing an adaptive data compression technique to acquire signal samples and facilitate efficient debugging. Similarly, the study detailed in [17] explains the implementation of the RLAB-UOC remote laboratory for telecommunications engineering students. This allows students to conduct analog and digital electronics experiments, enhancing their practical skills.

Research in [32] presents a TCP/IP architecture that utilizes open-source hardware and software to manage sensors and actuators in real time. It emphasizes the use of protocols such as MQTT and Node-RED. By implementing tools like Arduino and Raspberry Pi, this system enables effective control in remote laboratories. Authors in [33] discuss the development of remote laboratory experiments following an open-source model. In this framework, students interacted with control systems and real-time data using MATLAB, Simulink, and remotely accessible hardware. The results indicated an improved understanding of control concepts and increased interest and motivation towards control engineering.

3. Research Methodology

The remote FPGA lab employs the Waterfall methodology, which is particularly beneficial for systems requiring tight integration between hardware and software, such as IoT-based projects. This methodology offers a sequential and structured approach, allowing for clear and detailed planning of each development phase. It minimizes risks and ensures that each stage is completed before progressing to the next one. The guiding stages of this process include system requirements analysis, software requirements analysis, system design, implementation, testing and validation, and delivery and maintenance, as shown in Figure 1 [34].

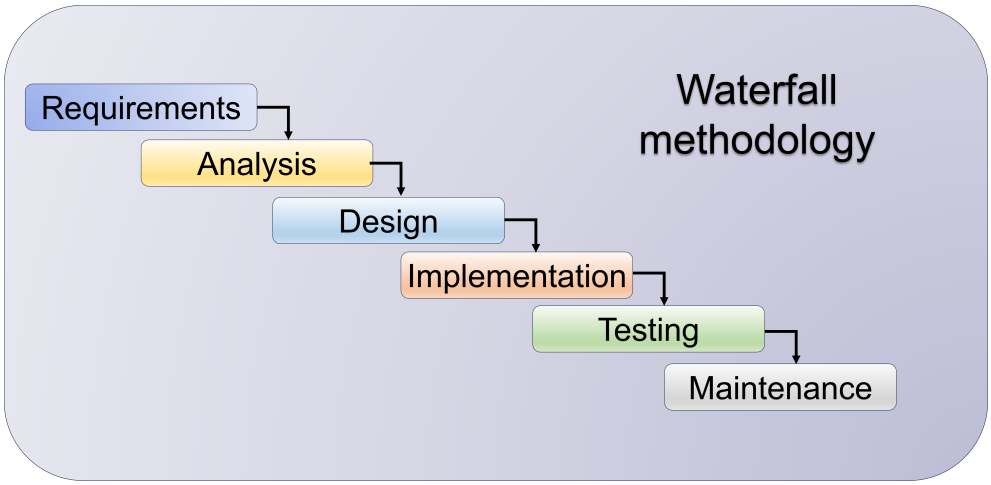


Figure 1. Waterfall methodology stages applied to the development of the FPGA remote lab.

The need to provide 24/7 remote access to educational resources was identified during the system requirements phase. This access allows students to engage in practical activities from anywhere, overcoming geographical and economic barriers and fostering a practical learning environment aligned with Education 4.0. In the analysis phase, a platform was defined to manage and monitor activities in real time, facilitating remote interaction. The system design phase organized the integration of various laboratory components, ensuring efficient communication and a smooth user experience. During implementation, necessary solutions were developed and configured to ensure that the laboratory could operate correctly and be accessible remotely. The testing and validation phase included a one-month pilot program involving 50 students from the Programmable Logic Devices course in the Bachelor's degree program for Robotics and Mechatronics Engineering. Feedback was collected through surveys to identify areas for improvement. Finally, the laboratory

was continuously operated in the delivery and maintenance phase. Its usage was monitored to facilitate future enhancements and expand available activities according to student needs.

The testing phase emphasized assessing remote lab performance and student satisfaction. The research participants, methods, and instruments used for data collection and analysis are detailed below.

3.1. Research Participants

The testing phase consisted of a one-month pilot program involving 50 students from the Bachelor of Robotics and Mechatronics Engineering program at the Autonomous University of Zacatecas, located in Zacatecas, Mexico. The participants in their third year of study, aged between 20 and 25, provided a representative sample of platform users in an academic setting. While the university is situated in Zacatecas, most students hail from various regions within the same state, and a smaller percentage come from other states in Mexico. This adds a diverse geographic perspective to the analysis. The group comprised 22% women and 78% men, with average grades ranging from 7 to 9.7. This diversity allows for a comprehensive evaluation of the remote lab experience among students with a similar educational background but differing levels of familiarity with remote learning environments. The selection of these sociodemographic characteristics was crucial for analyzing the platform's performance and accessibility in a technical academic context.

3.2. Research Method and Instrument Description

A structured survey was used to assess student perceptions. This survey consisted of 10 questions organized into several dimensions to evaluate key aspects of the lab experience: ease of use, accessibility, educational value, and quality of interaction. These dimensions allow exploring how students perceive different lab elements and their technical learning needs and expectations.

3.3. Measurement Scales Used in the Survey

Most items in the survey employed an ordinal scale with response options such as "Yes", "No", and "Partially", along with satisfaction levels ranging from "Very Satisfied" to "Dissatisfied." This approach facilitated the collection of quantitative data regarding satisfaction levels and students' perceptions of access and interaction in the remote lab. Additionally, an open-ended question was included to gather qualitative feedback, allowing students to suggest improvements and describe their overall experiences.

3.4. Data Collection Method

Data were collected using Google Forms, which allowed survey responses to be gathered anonymously and accessible. This tool facilitated an efficient collection process and ensured student opinions were recorded without identification, promoting unbiased and objective feedback. This approach helped accurately represent student perceptions regarding the remote lab experience.

3.5. Data Analysis

Quantitative data were processed and analyzed in Google Colab using Python 3, where descriptive statistics were employed to assess levels of satisfaction, ease of interaction, and perceptions of the educational value of the lab. A thematic analysis approach was used for qualitative analysis of open-ended responses, identifying common patterns and themes in the feedback. This methodology in Google Colab enabled structured and efficient data processing, allowing for a comprehensive understanding of the platform's strengths and areas for improvement to enhance the educational experience in the remote lab.

4. Remote FPGA Lab System Design

This section outlines the overall architecture of the remote FPGA lab system, created for motor control using digital signals. The system enables students to conduct motor

control experiments in a remote setting and incorporates technologies and devices such as the Basys 3 FPGA [35], the Raspberry Pi 5 [36], Firebase cloud services [37], React web developing [38], and YouTube streaming video [39]. It describes the hardware and software components and the methods used for implementing and operating the remote laboratory, as shown in Figure 2.

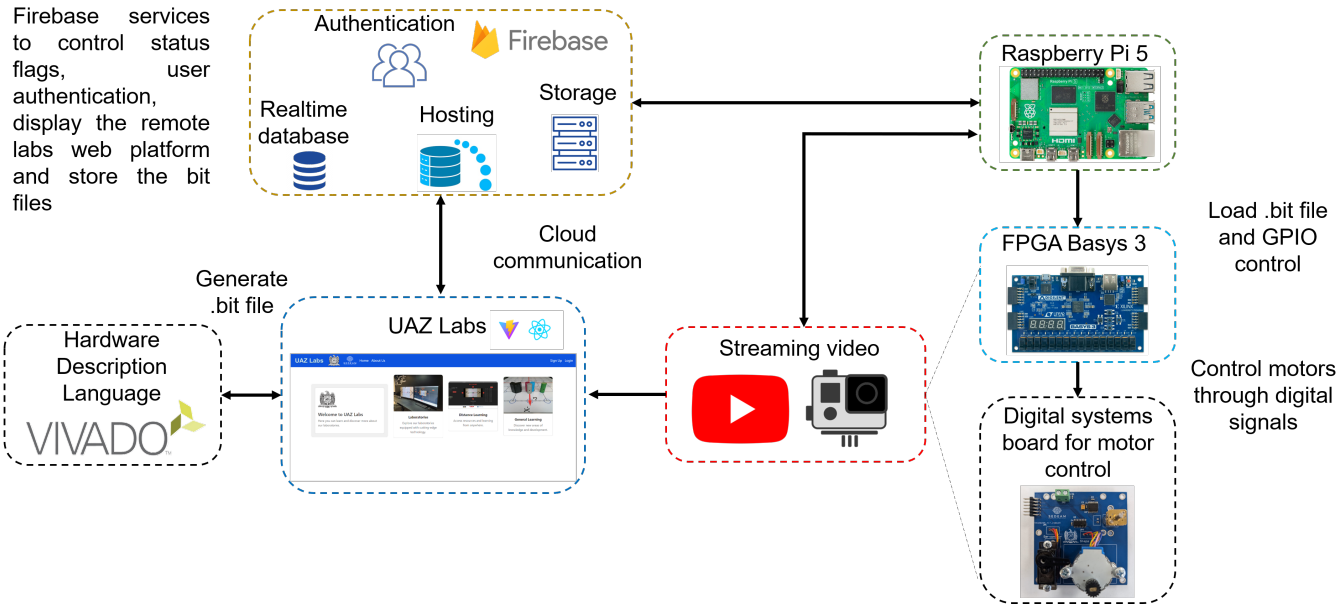


Figure 2. Remote FPGA lab architecture.

A bitstream file (.bit) must be created using the Vivado 2024.1 software specifically created for FPGA design. This tool is essential for programming in Verilog or VHDL and creating digital projects, enabling students to develop and synthesize their projects effectively.

Once the student has generated the .bit file, the hardware can be updated remotely using a web platform called the Autonomous University of Zacatecas (UAZ) Labs platform. This platform utilizes various services and technologies for its operations. Google Firebase is used for cloud storage of .bit files, enabling efficient and secure transfer from students' PCs to the FPGA. The UAZ Labs platform provides a web interface that allows students to schedule and control the labs in real time. A key feature of this platform is the ability to live-stream via YouTube, allowing students to observe the lab during their sessions.

The hardware setup includes the Basys 3 FPGA, an evaluation board for digital systems used as the primary practice device for students. A Raspberry Pi 5 is also used to communicate with the FPGA and connect to the Internet. The Raspberry Pi 5 plays a crucial role in detecting any change of the .bit files uploaded to Google Firebase and transferring them to the Basys 3 FPGA using OpenFPGALoader [40], an open-source tool essential for uploading the generated .bit files to the Basys 3.

The remote lab has an additional motor board specifically designed for the project. This board includes three motors to be controlled: a DC motor, a servo motor, and a stepper motor. Students can practice using any of these motors, adjusting speed, angles, and directions through the control signals.

The following sections detail each of the main components of the remote FPGA lab architecture, providing a deeper insight into its operation and features.

4.1. Hardware Description Language

Once the student has learned the relevant theory and analyzed the practice of the FPGA system, the next step is to generate the .bit file for uploading to the platform. To achieve this, the student needs to follow the different stages of the design process, as shown in Figure 3. This involves using hardware design tools like Xilinx's Vivado tool, which offers a robust environment for utilizing Hardware Description Languages (HDL) to synthesize, simulate, and implement complex digital circuits.

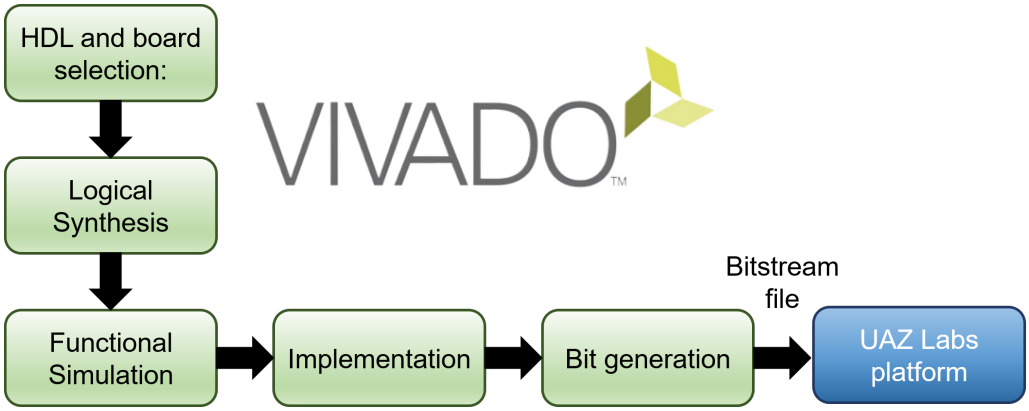


Figure 3. Bitstream file generation process in Vivado.

1. HDL and board selection: The process starts with selecting the HDL programming language, Verilog, VHDL, or SystemVerilog. It is important to choose the Basys 3 board to ensure that the design aligns with the specifications and capabilities of the remote lab.
2. Logical Synthesis: During synthesis, Vivado translates the HDL code into an equivalent logical representation. This step involves optimizing and generating a digital circuit structure from the provided HDL code.
3. Functional Simulation: The design undergoes functional simulation to verify its behavior under various input conditions. This process enables the detection and correction of logical errors before the physical implementation of the FPGA.
4. Implementation: Vivado assigns physical resources in the Basys 3 FPGA for the synthesized design, mapping the digital circuit to the programmable logic matrix and optimizing interconnection paths to meet timing and performance requirements.
5. Bit generation: The bitstream file contains the specific configuration that needs to be loaded into the FPGA. The .bit file configures the Basys 3 FPGA with the implemented design, enabling the hardware to operate according to the specifications defined in the original HDL code.

4.2. UAZ Labs Platform

The UAZ Labs platform serves as the central core of the remote lab system. It offers an intuitive and efficient interface for students to interact with hardware over the Internet. Students can schedule practice sessions, upload .bit files, and remotely verify the connection with the Basys 3 FPGAs through this platform.

The platform's web development utilized React and Vite to build an efficient and responsive user interface. React, a widely used JavaScript library, enables creating modular and scalable user interfaces. Vite, a rapid-build tool, offers an agile development environment with instant module reloading and streamlined configuration. Figure 4 depicts the main interface of the UAZ Labs platform.

The platform features a home section that showcases a list of available labs. Students need to choose the lab where they intend to conduct their practice. Once inside the lab interface, different sections are presented: theory, tutorial, schedule, and practice.

1. Theory: Introduces the theoretical concepts and foundations needed to understand the practice.
2. Tutorial: Provides detailed step-by-step guides to help effectively perform the practices.
3. Schedule: Helps students schedule and manage practice sessions to plan their learning effectively.
4. Practice: Allows users to upload .bit files and interact directly with the FPGA remotely.

At the bottom of the interface, the list of appointments the student schedules to perform the practices is displayed, providing a clear view of their upcoming sessions. Figure 5 shows the Basys 3 FPGA lab interface.

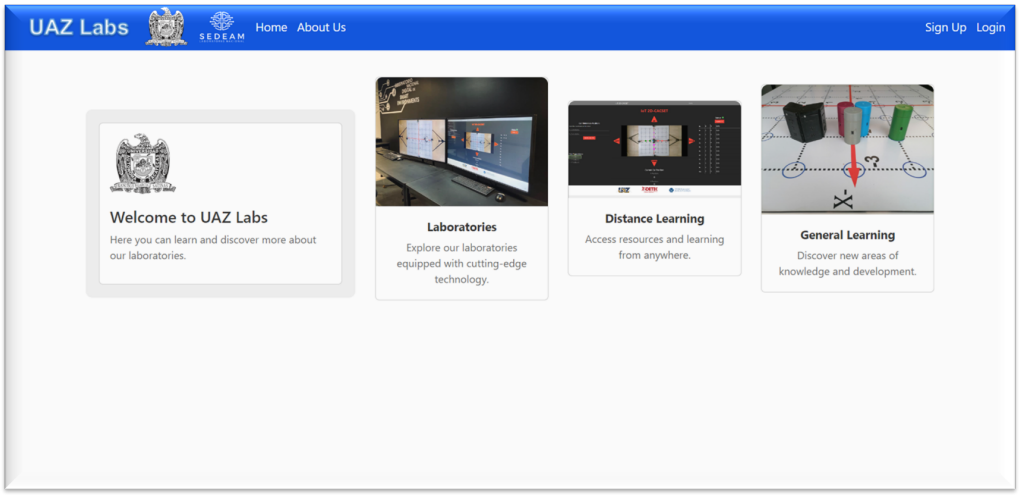


Figure 4. Main interface of the UAZ Labs platform.

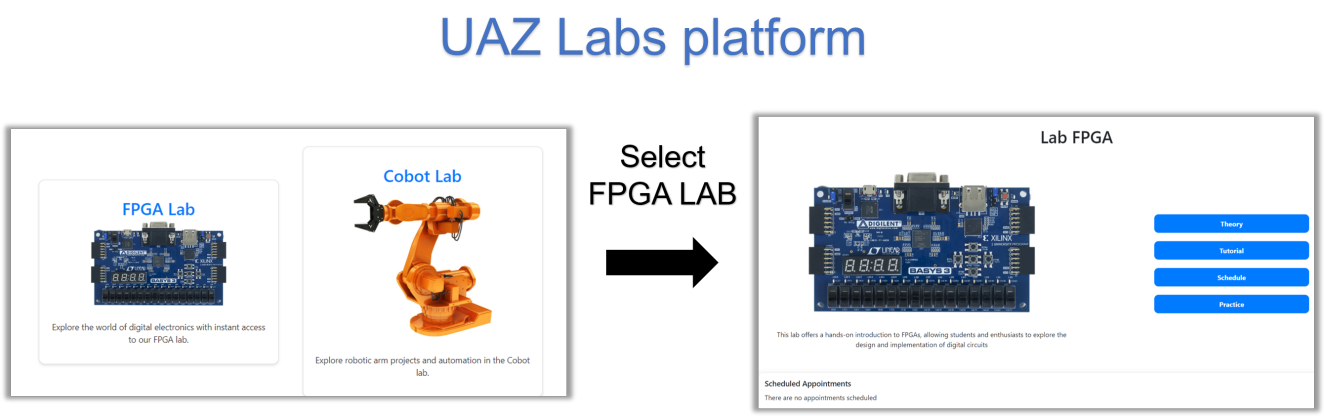


Figure 5. FPGA Laboratory in the UAZ Labs platform.

After scheduling an appointment for the lab, the user will be directed to the practice section. In this section, the student must have the .bit file ready and ensure connectivity with the lab. Once confirmed, the user uploads the .bit file to the Firebase storage service, which triggers real-time status flags in the database. These flags enable the Raspberry Pi to verify the connection, update the Basys 3 with the new file, and control the General Purpose Input/Output (GPIOs) accordingly. Figure 6 illustrates the flowchart showing the interaction between the UAZ labs platform and Firebase to conduct the FPGA lab practice, whereas Figure 7 shows the web platform interface.

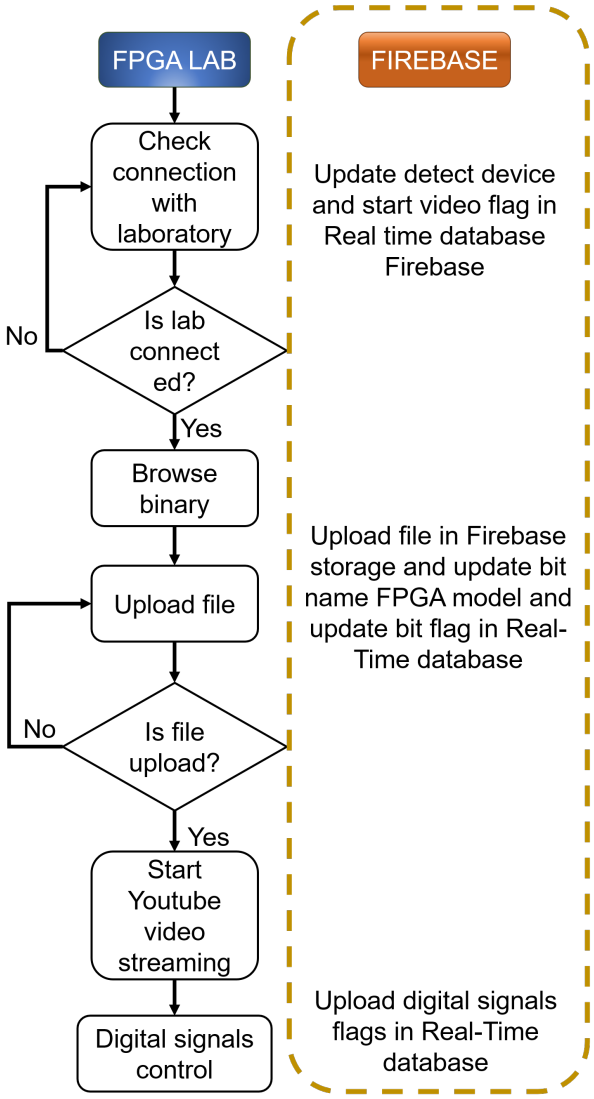


Figure 6. Flowchart illustrating the interaction between the UAZ Labs platform and Firebase for conducting FPGA lab practices.

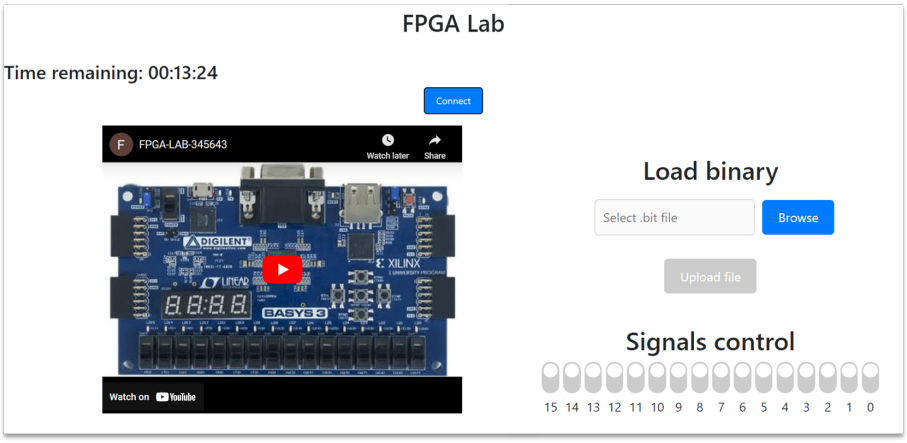


Figure 7. Interface of FPGA lab in UAZ Labs platform.

4.3. Cloud Computing

Firebase is a mobile and web application development platform provided by Google. It offers a variety of tools and services, providing a robust Backend as a Service (BaaS) that

facilitates agile development and the integration of advanced functionalities without the need to manage external infrastructures.

Firebase services implement advanced security measures to protect user information. All data transmitted through the platform are encrypted using the HTTPS protocol, ensuring the confidentiality of communications and preventing third parties from intercepting sensitive information. Additionally, customer data are logically isolated, protecting against unauthorized access and ensuring users' interactions remain private and secure. Firebase has received recognized security certifications, adhering to ISO 27001 standards [41] and passing SOC 1, SOC 2, and SOC 3 assessments. This demonstrates that Firebase services not only comply with privacy regulations but are also designed to maintain high security in data management.

The remote FPGA lab utilizes Firebase services to facilitate storage, user management, real-time communication, and web application hosting. These services are important intermediaries, enabling communication between the Raspberry Pi and the UAZ Labs platform, as illustrated in Figure 8. The Raspberry Pi leverages Firebase to detect and handle bitstream files (.bit) uploaded by students to receive control signals from the platform and update status flags. On the other hand, UAZ Labs utilizes Firebase to update real-time status, manage student sessions, and offer an interactive and accessible interface.



Figure 8. Firebase Services in the remote FPGA lab.

4.3.1. Realtime Database

The Firebase Realtime Database is a Not Only SQL (NoSQL) database based on JavaScript object notation (JSON), allowing real-time data synchronization between connected clients. The NoSQL database comprises multiple nodes that enable the modification of specific flags. This allows users to continuously update the streaming Uniform Resource Locator (URL), control the start and stop of video streams, detect FPGA devices, and update .bit files needed by the FPGA. Additionally, the database verifies the state of signals control and ensures that actions are only performed when necessary. This facilitates effective data management and synchronization between different system components.

It is important to emphasize that the data uploaded to the database is encoded to protect the information's security. Additionally, this database does not manage user data or passwords, which helps reduce the risk of exposing sensitive information. This approach strengthens data protection and ensures critical information remains secure and protected from unauthorized access.

4.3.2. Authentication

The Firebase Authentication Service provides a complete solution for managing access to the UAZ Labs platform, making it easier to handle user logins and user management. This service allows users to authenticate using their email and password, ensuring that only authorized users can access the platform. For the UAZ Labs platform, the authentication system is limited to institutional emails from the Autonomous University of Zacatecas

(uaz.edu.mx), ensuring that only university students and academic staff have access. Firebase Authentication also offers tools for managing passwords, recovering accounts, and protecting against unauthorized access, providing a secure and personalized user experience. Integration with other Firebase services allows for efficient permission and access management, working with the real-time database and other system functions to provide a robust and secure platform.

This service implements a secure and robust login system that protects access credentials and enhances password strength through complex policies designed to minimize the risk of unauthorized access. Firebase Authentication also provides tools for managing passwords, recovering accounts, and protecting against unauthorized entry, ensuring a secure and personalized user experience.

4.3.3. Storage

The Firebase Storage service allows for the secure and efficient management of files, including uploading, downloading, and overall data management. In the context of the remote FPGA lab, Firebase Storage stores the .bit files generated by students for FPGA programming. Users can select the file from the platform's file browser, and the selected file is then stored in Firebase Storage. Once the file is uploaded, the upload bit and bit name flags are updated in real time in the database. This update notifies the FPGA that a new hardware description file has been uploaded to the platform and needs to be updated on the FPGA board.

4.3.4. Hosting

Firebase Hosting is utilized to deploy and serve the UAZ Labs platform web application. This service ensures the platform website can be hosted, allowing users to access the user interface from any browser. Firebase Hosting offers a dependable and high-performing infrastructure with support for hyperText transfer protocol secure. It guarantees that the UAZ Labs platform is available online in the remote FPGA lab, enabling students to program and control the FPGA, access their labs, and manage their lab sessions remotely. The platform website was developed using React to build modular user interfaces and Vite to build and reload modules quickly. Firebase Hosting provides the infrastructure to deploy and serve the website.

4.4. Raspberry Pi

The Raspberry Pi is a series of single-board microcomputers developed by the Raspberry Pi Foundation. These microcomputers are designed to provide an affordable platform for various computing projects. The Raspberry Pi 5 model, released in 2024, is the latest version of this line. It offers significant improvements in performance and capacity compared to its predecessors. The Raspberry Pi 5 has the following notable features:

- Processor: It has a quad-core 2.4 GHz ARM Cortex-A76 chip.
- Memory: It supports up to 8 GB of Low Power Double Data Rate 4 Random Access Memory (LPDDR4) RAM.
- Connectivity: The Raspberry Pi 5 features USB 3.0, USB 2.0, and Gigabit Ethernet ports and supports Wi-Fi 6 and Bluetooth 5.0.
- Storage: It supports external storage through a USB 3.0 port and has an M.2 interface for Solid State Drive (SSDs).

The board has Wi-Fi capability, but its Ethernet port is essential for deploying it in the remote FPGA lab. This port provides a more stable and lower-latency internet connection than Wi-Fi, which is crucial for smooth video streaming and reduced latency in the lab environment.

The recommended operating system for the Raspberry Pi is the 64-bit Raspbian image, which has been integrated for use in the remote FPGA lab. This image offers a stable environment compatible with various applications and devices.

The Raspberry Pi 5 is designed for rugged environments, so a heatsink has been added to prevent overheating issues. This is important for applications that require high continuous performance. A 512 GB NVMe SSD connected via a HAT Drive has also been included. This SSD offers up to 10 times faster file read speeds than a Secure Digital (SD) card, eliminating bottleneck issues and significantly improving system performance compared to using SD card storage alone.

Figure 9 shows the Raspberry Pi 5 setup with all connected peripherals, providing a comprehensive view of the system's robust configuration.

- Power: The Raspberry Pi 5 requires a minimum 5 V 3A USB-C power supply. This power supply is essential for proper operation of all connected peripherals.
- HDMI connector: Used to connect a display, allowing display of the system interface and real-time operations monitoring.
- NVMe SSD connector: The Raspberry Pi 5 features a HAT Drive expansion connector for connecting a 512 GB NVMe SSD drive, which provides high-speed storage with superior read and write times compared to an SD card.
- GPIOs for control signals: The 16 GPIOs pins transmit control signals between the Raspberry Pi 5 and the FPGA. These pins are crucial for remote lab operations.
- Fan connector: An additional connector is available for a cooling fan. The fan works with the heatsink to maintain proper operating temperature during intensive operations and prevent overheating.
- USB connector for camera: The USB port connects a camera to capture real-time video, enabling streaming to platforms like YouTube and monitoring remote lab operations.
- USB connector for the Basys 3: Another USB port is connected to the Basys 3 FPGA, enabling bitstream uploading and control communication with the Raspberry Pi.
- Ethernet communication connector: The Ethernet port provides a stable, high-speed, low-latency network connection. This is essential for real-time video streaming and efficient communication with servers and the FPGA.
- USB connector for mouse and keyboard: USB port available to connect input devices such as a mouse and keyboard for initial setup or maintenance purposes.

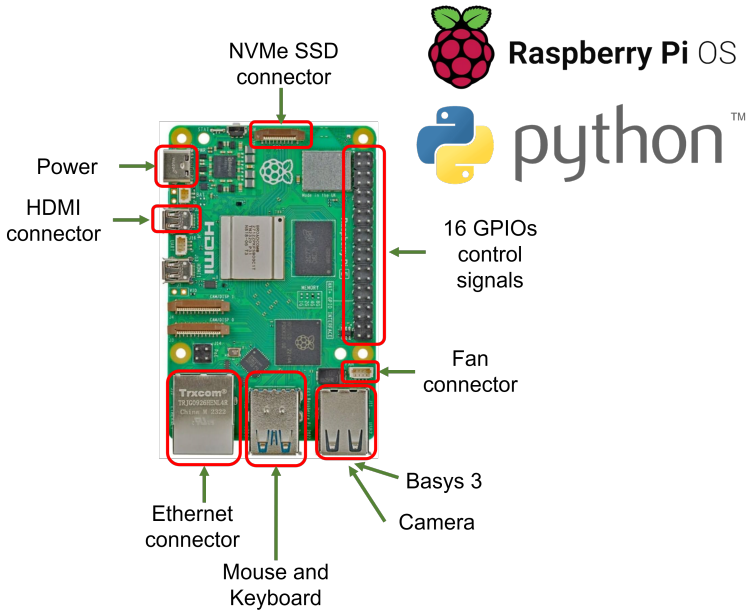


Figure 9. Peripheral connection diagram for Raspberry Pi 5 in remote FPGA lab.

The 16 GPIO signals are sent from the UAZ Labs platform to Firebase, and they are then transmitted to the Raspberry Pi and directed to the GPIOs on the FPGA. Table 1 and Figure 10 illustrates the connection of the control signals between the Raspberry Pi and the Basys 3 board.

Table 1. Assignment of control signals to GPIO pins on Raspberry Pi.

Function	Raspberry pi 5 GPIO Port	Basys GPIO Port
Signal control 1	8	G3
Signal control 2	7	G2
Signal control 3	10	H2
Signal control 4	11	J2
Signal control 5	12	K2
Signal control 6	13	L2
Signal control 7	16	H1
Signal control 8	15	J1
Signal control 9	22	N1
Signal control 10	21	N2
Signal control 11	24	M1
Signal control 12	23	M2
Signal control 13	26	M3
Signal control 14	27	L3
Signal control 15	28	K3
Signal control 16	29	J3
Ground	GND	GND

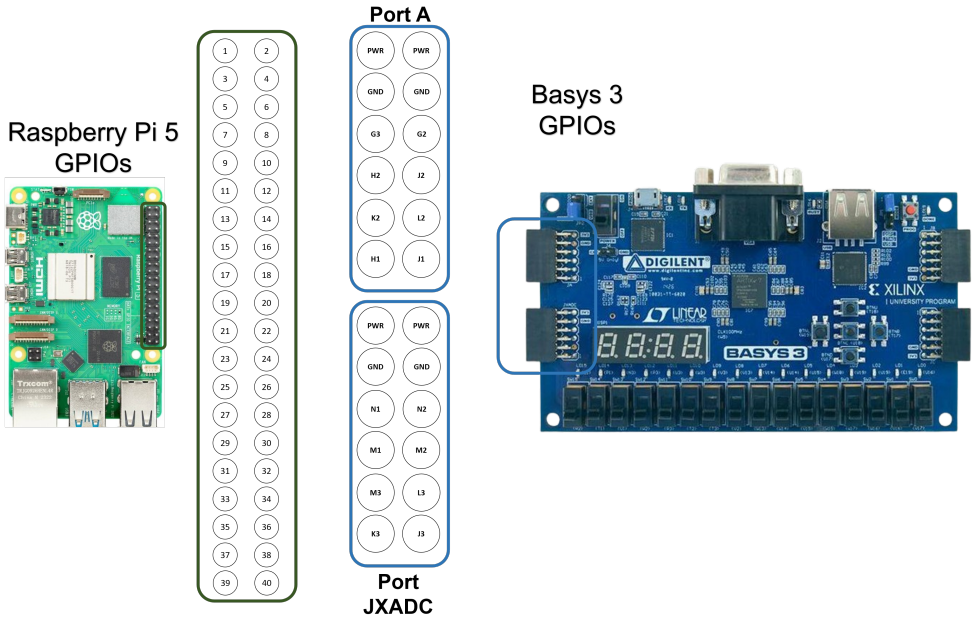


Figure 10. Raspberry Pi 5 pin assignment for control signals.

Python 3 plays a crucial role in developing the remote lab system. It coordinates the interactions between the UAZ Labs platform, Firebase, and the FPGA using the Raspberry Pi. The Python code implements several key functions:

1. Connecting to Firebase and handling data: The code checks the Internet connection and performs read and write operations on the Firebase Realtime Database. It dynamically updates the streaming URL and manages the state of the video stream based on the data received from Firebase.
2. Handling control signals: Python ensures the system's responsiveness by updating the state of the signals control. This is achieved by continuously reading data from Firebase and updating the states in real time.
3. Interacting with the FPGA: The code downloads necessary update files to the FPGA and verifies its state to perform the bitstream update when necessary. File download functionality and file integrity validation ensure that the FPGA stays up to date. The bitstream is uploaded to the FPGA with openFPGALoader, a versatile and open-

source tool that facilitates the programming and configuration of various FPGAs, ensuring a reliable and efficient upload process.

4. Video streaming to YouTube: Google Cloud services and the YouTube Application Programming Interface (API) manage the live stream. The camera captures the video in real time using OpenCV, while Google services allow for authenticating access and obtaining the necessary stream ID. Video capture is performed on a separate thread to ensure a smooth stream, and stream start and stop events are handled to manage the video stream efficiently.
5. Error handling: Robust error handling is implemented to ensure system stability. This includes handling connection issues to Firebase and detecting failures during code execution.

4.5. FPGA Basys 3

The Basys 3 is a development board based on the Xilinx Artix-7 FPGA. It provides an ideal environment for learning and experimenting with digital systems. The Basys 3 is essential for students to upload and run their digital hardware designs remotely. This is done using a streamlined workflow that includes the Raspberry Pi 5 and a cloud management system.

The Basys 3 is powered and connected to the Raspberry Pi through a USB data transfer cable. This cable provides the necessary power for the Basys 3 to operate and enables communication between the two devices. The USB connection lets the Raspberry Pi directly interact with the Basys 3 and automatically load bitstream files.

The tool used to load the bitstream is OpenFPGALoader. Once uploaded to Firebase, it allows the .bit file to be automatically transferred and programmed into the Basys 3 FPGA without manual intervention. This automation ensures that students can update and verify their FPGA designs without doing anything else beyond uploading the file to the platform.

In the remote laboratory, the Raspberry Pi 5 sends 16 control signals to the JA1 and JXADC ports of the Basys 3. These signals control various motor parameters such as frequency, direction, and speed for direct current motors, servomotors, and stepper motors. There are 2^{16} possible combinations of these signals, providing students various configurations and adjustments to apply to the motors during digital control practices.

Port C of the Basys 3 is linked to the motor control board through signals that enable precise control of different types of motors. These signals include the following:

- DC Motor: Two control signals for an H-bridge are used to control a direct current motor, and one PWM signal is dedicated to adjusting the speed of the direct current motor through the H-bridge.
- Servo motor: A PWM signal is used to control the servo motor's position.
- Stepper motor: Four control signals are required to operate the stepper motor.

The signals assigned to the GPIO pins of the Basys 3 FPGA are shown in Table 2 and Figure 11.

Table 2. Assignment of motor control signals to GPIO pins on Basys 3.

Function	Signal Type	Port C Basys 3	Control Motor Port
PWM Signal for DC Motor	PWM Output	L17	2
DC Motor Control Signal 1	Digital Output	M19	4
DC Motor Control Signal 2	Digital Output	P17	6
PWM Signal for Servo	PWM Output	R18	8
Stepper Control Signal 1	Digital Output	K17	1
Stepper Control Signal 2	Digital Output	M18	3
Stepper Control Signal 3	Digital Output	N17	5
Stepper Control Signal 4	Digital Output	P18	7
Ground	GND	GND	9–10

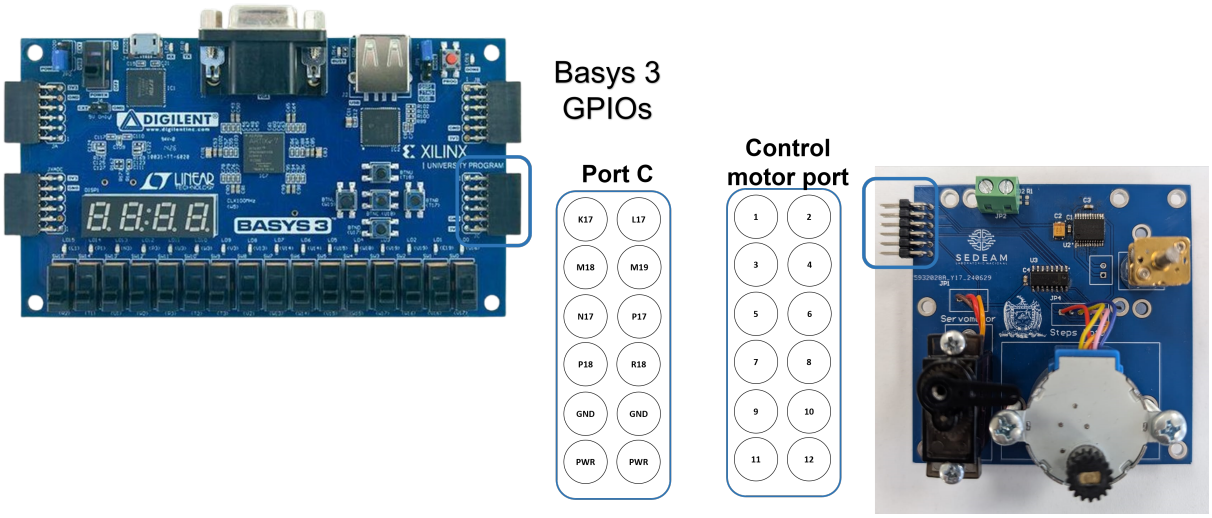


Figure 11. Basys 3 GPIO pin assignment for motor control.

4.6. Motor Control Board

The motor control board was created specifically for the FPGA Lab, measuring 7.5 cm × 7.5 cm. This Printed Circuit Board (PCB) is designed to control three types of motors: a DC motor, a servo motor, and a stepper motor. Figure 12 illustrates the layout and connections of the designed PCB and motor control board.

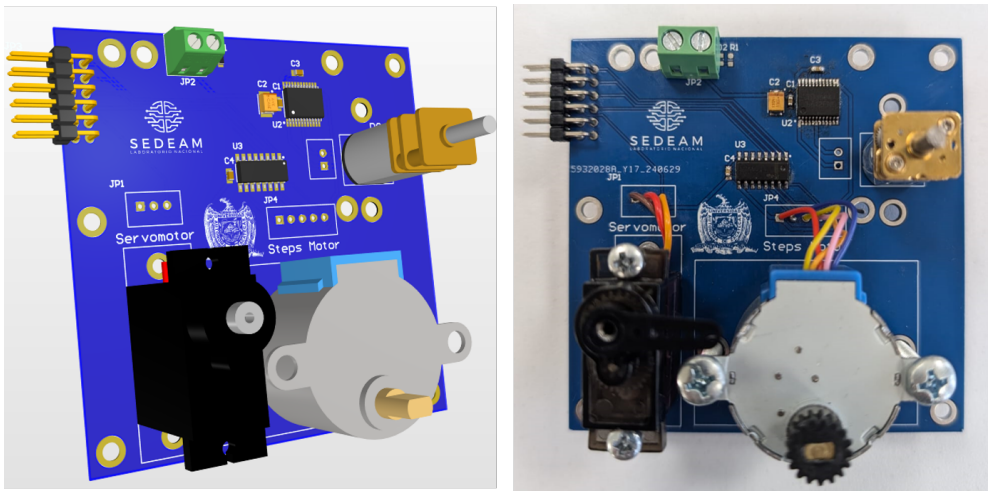


Figure 12. Motor control board design and its PCB layout.

The motor control board connects to a 5 V power supply to power the motors. However, the control signals from the Basys 3 operate at 3.3 V. This setup allows the board to efficiently drive the motors while seamlessly integrating with the system control logic.

- DC Motor: The DC motor is the N20 Gearmotor model, which runs at 400 revolutions per minute. It is connected to a TB6612FNG H-bridge, which enables control of the motor's direction and speed using PWM signals and digital logic. The N20 motor operates at 5 V, producing an output torque of approximately 0.5 kg-cm.
- Servo Motor: This remote lab utilizes the MG90S micro servo. This servo motor does not need an additional driver since the PWM signal is sent directly from the FPGA. The MG90S operates with a voltage range from 4.8 to 6 V and has a turning angle of 180 degrees and a torque of approximately 1.8 kg-cm at 5 V. The PWM signal has a modulation frequency of 50 Hz and a period of 20 ms, enabling precise control of the servo angle.

- Stepper Motor: The stepper motor is a model 28BYJ-48 with a ULN2003 driver. It operates at 5 V and has 2048 steps per revolution, allowing precise angle control. The ULN2003 can handle the control signals required for the stepper motor, with a supply voltage of 5 V and a maximum current per channel of 500 mA.

4.7. Streaming Video

The streaming system in the FPGA lab utilizes the FIT0729 USB camera to capture real-time video. This camera connects directly to the Raspberry Pi 5 via a USB port, enabling high flexibility and performance in video capture. The Raspberry Pi 5 has a 512GB NVMe SSD and a heat sink to prevent overheating, optimizing it for continuous video processing and streaming.

The system uses the YouTube Studio API for video streaming. This is achieved through a Python script that interacts with the YouTube API using the Google API client library. The API is responsible for setting up and managing the live stream, enabling students to observe the lab environment and their practice results in real time. Once the video is set up, the camera's captured data is processed in real time using OpenCV, a popular image and video processing library.

The video stream can be captured at different resolutions, such as 320 × 240, 640 × 480, and 1920 × 1080 for Full HD streams. This flexibility allows us to balance video quality with streaming latency according to the lab's requirements. Higher resolutions like 1280 × 720 or 1920 × 1080 for this project can enhance streaming quality, although this may result in increased latency. This uses FFmpeg, a media processing and streaming tool that allows real-time video encoding and delivery to YouTube using the Real-Time Messaging Protocol (RTMP) for stream management. YouTube Studio also supports HTTP Live Streaming (HLS), which provides adaptive streaming to automatically adjust video quality based on the viewer's bandwidth. The FFmpeg presets, such as ultrafast or medium, allow the balance of compression speed and streaming quality, impacting latency. The ultrafast preset is used for minimal latency, while the medium setting can improve compression quality when latency is less concerned.

The video stream to YouTube is initiated with the lab session name, the unique FPGA ID, and the broadcast key required for authentication. The stream URL is updated in the Firebase database when the student accesses their scheduled appointment through the UAZ Labs platform. The UAZ Labs platform retrieves this updated URL and displays the live video directly on the platform. Upon completion of the lab session, the stream is automatically stopped. The system detects the session closure and sends a signal to terminate the stream on YouTube, thus ending the live stream. This process ensures that the stream is active only during the student's lab session, optimizing platform resources and preventing unnecessary broadcasts. The architecture for the video stream is illustrated in Figure 13.

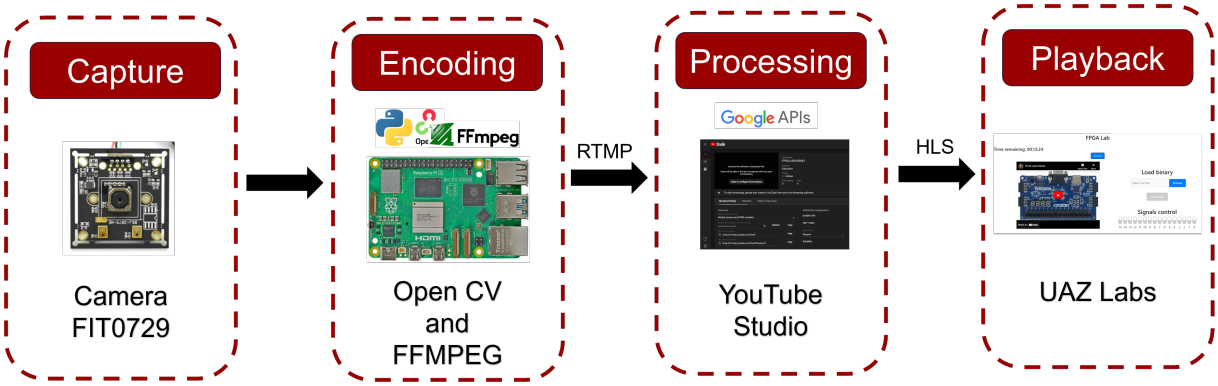


Figure 13. Video streaming architecture using RTMP and HLS in live laboratory sessions.

5. Proposed Practices for Motor Control

This section presents three practices designed to help students master mechatronics concepts about motor control in the FPGA lab. Each practice is dedicated to a specific type of motor: the DC motor, servo motor, or stepper motor. Through these labs, we aim to help students gain skills, abilities, and fundamental concepts such as digital signals, PWM, and bit representation of data, which Industry 4.0 is now demanding from students.

In the first practice, students will learn about DC motor control and discover how to adjust the motor's speed and direction using digital and PWM signals. The second practice will cover servo motor control, focusing on using PWM signals to precisely adjust the motor's position. Finally, the third practice will address stepper motor control, allowing students to experiment with sequencing digital signals to control motion in discrete steps.

The remote lab FPGA used the Mechatronics Educational Conceptual Framework (EMCF) methodology [42]. This method teaches engineering concepts through a progressive approach at three levels: concrete, graphic, and abstract, as shown in Figure 14. It is particularly useful for educational practices in digital systems and programmable logic devices, as it allows students to develop critical and mechatronic thinking. Students start by manipulating objects directly (concrete level), then move on to symbolic representation of concepts (graphic level), and finally gain theoretical and abstract understanding (abstract level).

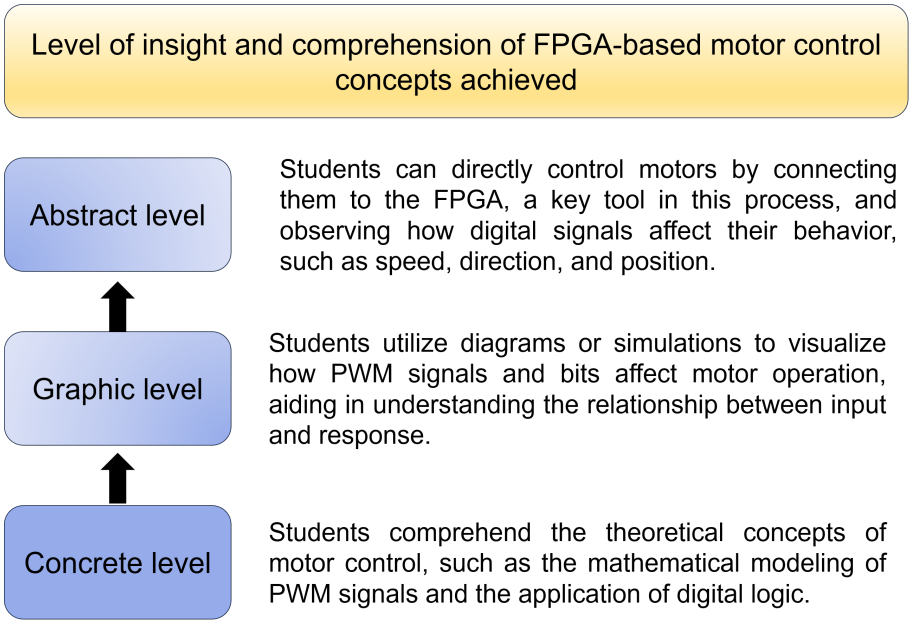


Figure 14. Levels of comprehension and development of motor control concepts using an FPGA based on the EMCF.

5.1. DC Motor Practice

In this practice, students will work with DC motors, exploring fundamental control concepts using digital signals. They will learn how to analyze and manage the PWM signal to regulate the motor's speed and direction of rotation through bit manipulation.

To control the DC motor, students must use 10 input signals. The signals are detailed below and in Figure 15.

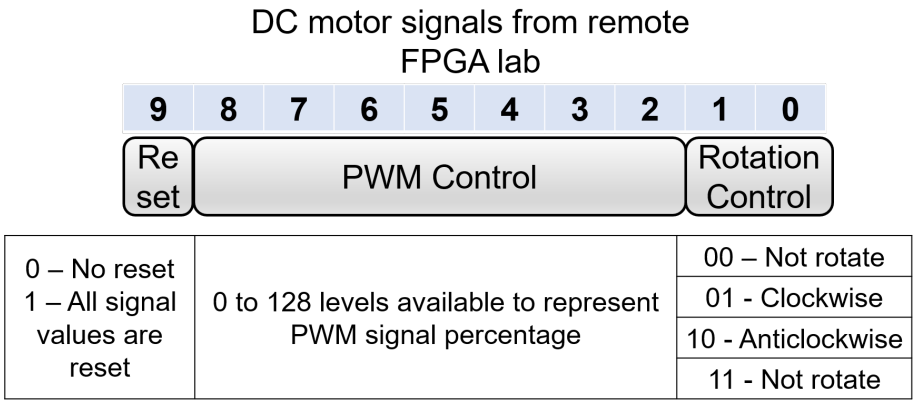


Figure 15. Methodological scheme for DC motor control using 10 input signals.

Direction of rotation control (2 bits): The two Least Significant Bits (LSB) of the input signal will determine the motor's direction of rotation.

- If both bits are 00, the motor will not rotate.
- If the first bit is one and the second is 0, the motor will rotate in one direction.
- If the first bit is 0 and the second is 1, the motor will rotate in the opposite direction.

PWM control (7 bits): Bits in Positions 2 through 8 will be used to adjust the percentage of the PWM signal. These bits will allow you to set PWM values ranging from 0 to 100. It is important to remember that, although 128 levels are available (because 7 bits are used), the PWM signal must be limited to a maximum of 100.

Reset (1 bit): Bit 9 will reset the system. When this bit is set (1), all control signal values and states will be reset.

PWM signal frequency (10 KHz): The TB6612FNG H-bridge manufacturer recommends configuring the PWM signal to 10 kHz. It is important to ensure that the duty cycle does not exceed 100.

PWM percentage display: The first three displays will show the percentage value of the PWM signal on a 7-segment display. This value will change as the PWM signal is adjusted.

5.2. Servomotor Practice

In servo motor practice, students can analyze how changing the duty cycle affects the signal and motor performance. To control the PWM signal, students must use eight input signals. The signals are described below and in Figure 16.

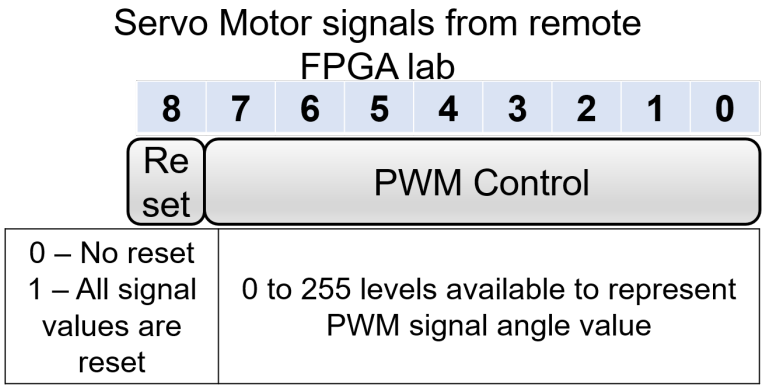


Figure 16. Methodological scheme for servo motor control using nine input signals.

PWM control (8 bits): The input signal's first 7 bits will adjust the servo motor's position. These bits enable a PWM value corresponding to an angle between 0° and 180°.

Even though 8 bits can represent 255 possible values, it is important to ensure that the maximum angle does not exceed 180 degrees. This way, the 8-bit value is translated to a range of 0 to 180, enabling precise control of the servo position.

Reset (1 bit): Bit 8 will reset the system. When this bit is set (1), all control signal values and states will be reset.

PWM signal frequency (50 Hz): The MG90S servo motor needs to be driven with a PWM signal set to 50 Hz. It is crucial to ensure that the PWM signal's duty cycle stays below 100% as it directly impacts the servo's performance and position accuracy.

PWM percentage display: The first three displays will indicate the angle value of the PWM signal on a 7-segment display.

5.3. Step Motor Practice

In this practice, students will learn about controlling a stepper motor, focusing on the precise motion generated by the motor's stepping rotation. The stepper motor moves in discrete steps, making it suitable for applications that require accurate positioning. Students will utilize digital signals to control the sequence of steps and the direction of the motor.

To control the stepper motor, students will use specific input signals detailed below, along with Figure 17.

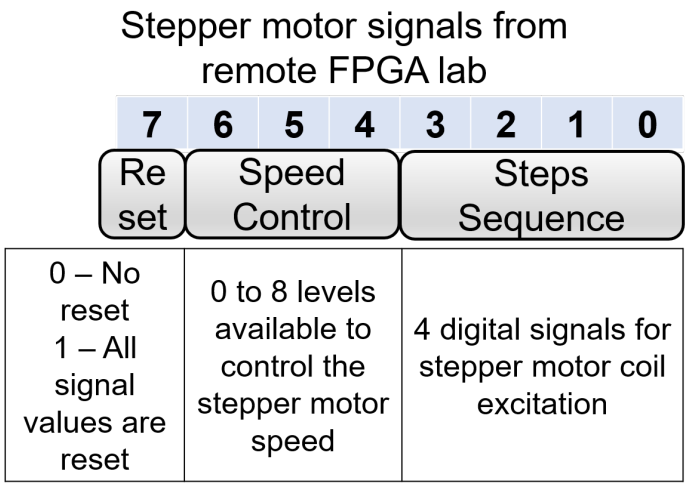


Figure 17. Methodological scheme for stepper motor control using 7 input signals.

Step sequence (4 bits): The first four input signals control the excitation sequence of the stepper motor coils. Each bit represents the state of one of the coils (on or off), and the correct activation sequence will allow the motor to advance in the desired direction. The combination of these 4 bits follows a specific pattern that determines the movement of the stepper motor.

Speed control (3 bits): These signals allow you to control the time interval between motor steps, allowing you to adjust the motor's speed. The longer the interval, the slower the rotation speed, and vice versa. This control over the motor's speed is a key aspect of its operation.

Reset (1 bit): The last input bit serves as a safety net. When activated, it triggers the reset function, ensuring that the control signals are reset and the stepper motor returns to its initial position. This reset function provides a sense of reassurance in case of any unexpected movements.

Step display: The first two 7-segment displays will show the number of steps the motor has taken in the sequence (a maximum of 16 due to the 4 bits). A third display will indicate the motor's speed. This will help students visualize the motor's progress and speed, allowing effective control adjustments.

6. Results

The development of the remote lab involved integrating and optimizing several hardware subsystems, such as the Raspberry Pi 5, the FPGA, the motor control board, the power supply, and the camera. These components were combined in a compact design based on an aluminum profile structure. This structure provides structural support and efficiently organizes all the devices. It incorporates 3D-designed joints, as shown in Figure 18. The lab was set up in the dedicated spaces for IoT and remote laboratories at the Academic Unit of Electrical Engineering of the Autonomous University of Zacatecas.

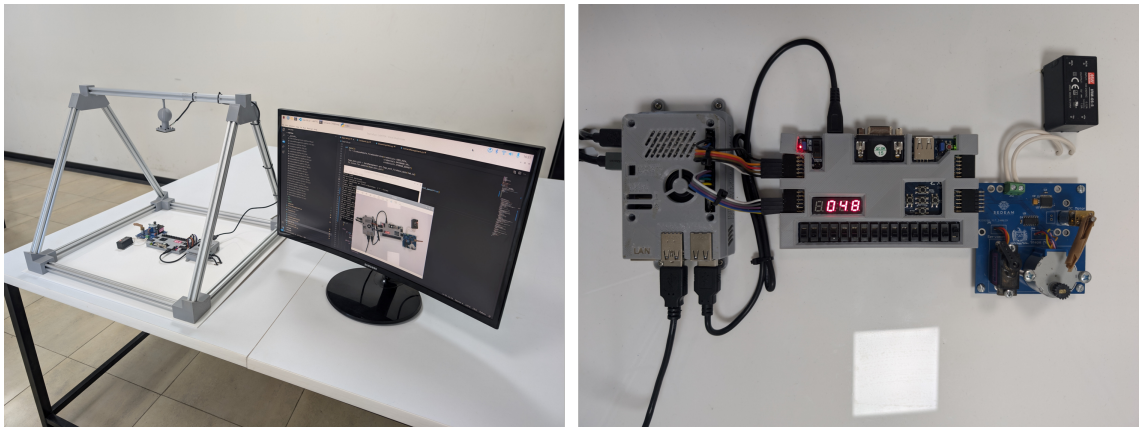


Figure 18. Design of the structure for the remote FPGA lab.

The system's various parts were placed on the structure to ensure easy access and operation. The camera was mounted to point towards the system components to achieve the visualization objective, allowing students to monitor the motor activity through the UAZ Labs platform, as shown in Figure 19. This design optimizes space and makes identifying and diagnosing possible component failures during practice easier. Additionally, using aluminum profiles provides rigidity and stability, ensuring that the structure can withstand the vibrations and movements generated by the motors during operation.

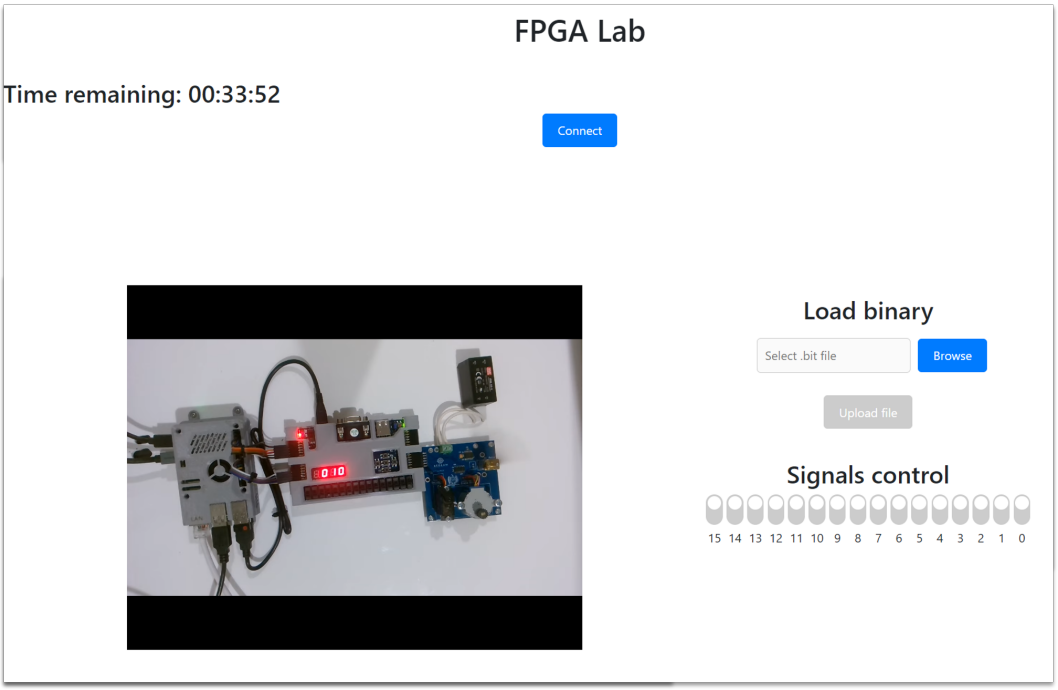
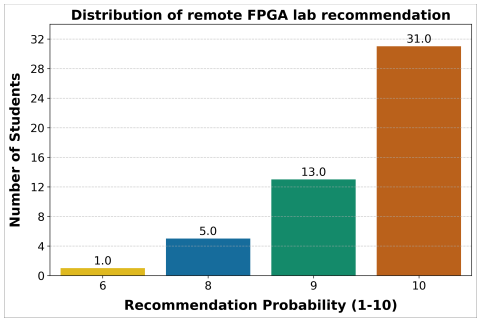


Figure 19. Camera visualization in the remote FPGA lab.

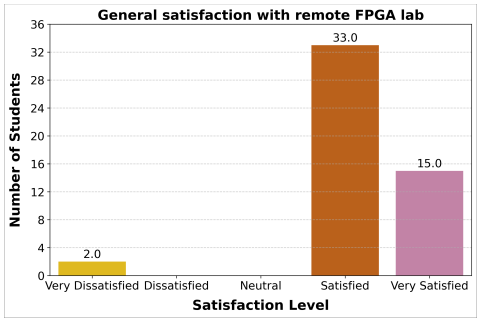
The laboratory underwent a month-long testing period by students enrolled in the same university's Bachelor of Robotics and Mechatronics Engineering program. The tests focused on Programmable Logic Devices (PLDs) and aim to equip students with essential skills for designing, programming, and implementing complex systems using programmable logic devices like FPGAs. This remote laboratory allowed students to apply their knowledge in the subject, enabling them to gain experience with FPGA-based systems and comprehend the interaction between hardware and software.

The students were required to sign up on the platform using their institutional email, and as a result, we achieved 50 registrations. This is a crucial step in conducting tests in a remote lab, and it shows the students' keen interest and active participation in the practical activities.

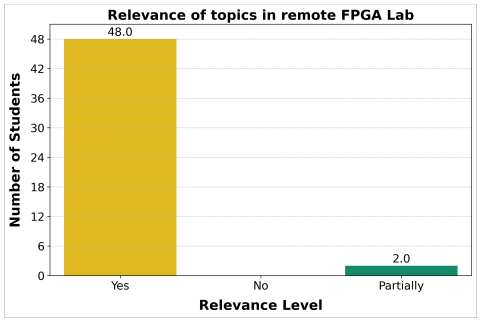
After studying the theory related to three motor control practices, students could apply their knowledge by conducting tests using the remote FPGA lab. Following this, they were given a survey containing important questions to evaluate their experience, their interaction with the platform, and the relevance of the content to their academic training, as shown in Figures 20 and 21. The data collected provided valuable insights into the students' perception, highlighting the strengths of the platform and areas for improvement for future implementations and optimizations of the laboratory.



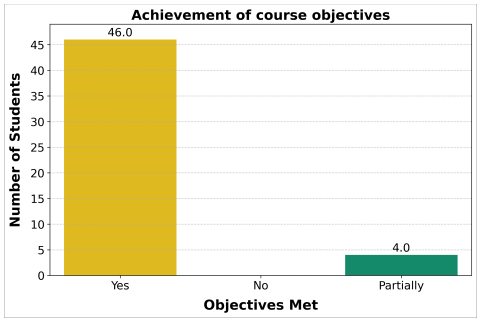
(**a**)



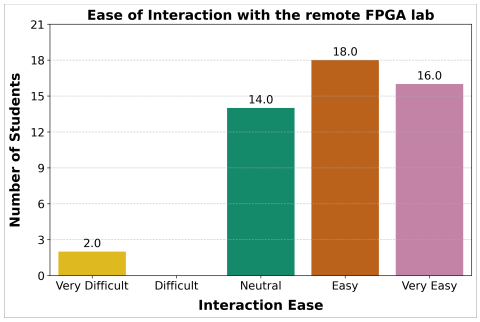
(**b**)



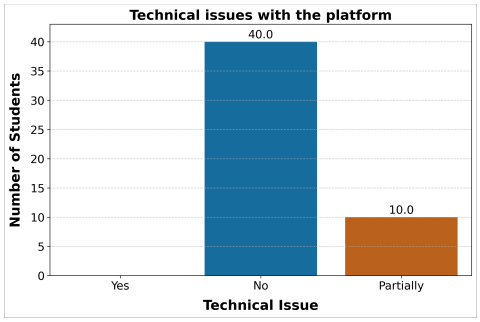
(**c**)



(**d**)

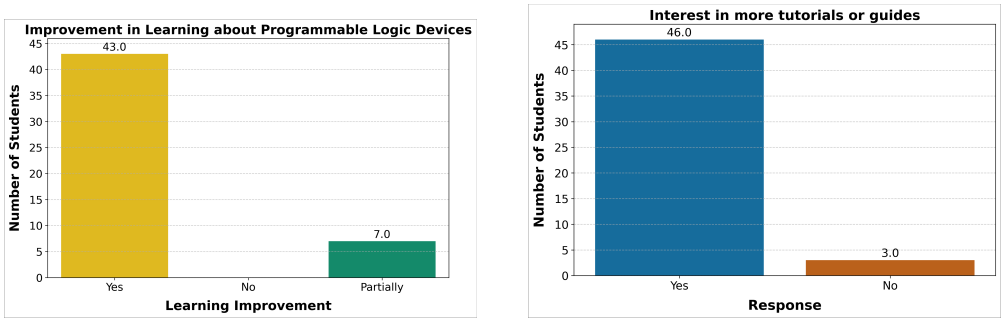


(**e**)



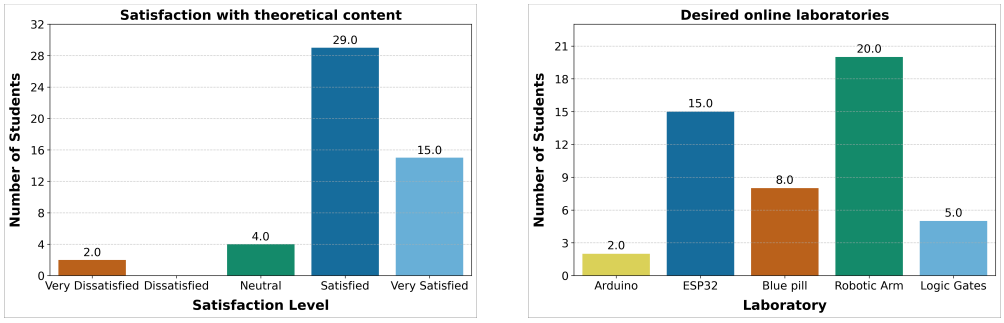
(**f**)

Figure 20. *Cont.*



(**g**) (**h**)

Figure 20. Graphs of the survey conducted with 50 students. Likelihood of recommending the platform (**a**), overall satisfaction (**b**), relevance of the topics covered (**c**), fulfillment of course objectives (**d**), ease of interaction with the lab (**e**), technical difficulties encountered (**f**), improvement of learning about programmable logic devices (**g**), and interest in receiving more tutorials or guides (**h**).



(**a**) (**b**)

Figure 21. Graphs of the survey conducted with 50 students. Satisfaction with the theoretical content provided (**a**) and other desired online laboratories (**b**).

7. Discussion

Remote labs are a key innovation in Education 4.0. They offer new access and quality solutions in the teaching and learning process. As remote labs allow students to practice with real equipment from outside the school lab, they surpass geographic and economic barriers. They were very useful during pandemic times, and they continue to be.

The use of IoT and CC as emerging technologies in remote labs enhances the teaching and learning process, promotes the development of new abilities and skills needed in Industry 4.0 and many others demanded in a digital era, and is adaptable to students' needs and learning styles, allowing them to master complex concepts at their own pace.

The remote FPGA lab aligns with the SDGs, which emphasize the importance of ensuring inclusive, equitable, and quality education and promoting lifelong learning opportunities, as highlighted in the research presented by Uchima-Marin [43]. This study links to SDG 4, which promotes quality education for all, ensuring every student can access effective educational tools. Furthermore, it relates to SDG 9, which highlights the importance of innovation, industry, and infrastructure by developing an accessible platform that facilitates the integration of new technologies in rural settings. Lastly, the study contributes to SDG 17 by fostering partnerships between the education and technology sectors, thus facilitating the achievement of the SDGs.

As proved in the results of the survey students took, remote labs have a positive impact and are highly regarded as trustworthy and satisfying by them; therefore, the integration of remote labs in Education 4.0 is crucial at this time to prepare students for their future.

The survey results for the remote FPGA lab show that students have a positive perception of the platform, which has significantly impacted their educational experience. Eighty-eight percent of the students are highly likely to recommend the platform, with 62% giving it a top rating of 10 and 26% giving it a 9. This indicates their overall satisfaction with the tool. Additionally, 66% of respondents reported being satisfied, while 30% were very satisfied with their experience. However, areas for improvement have been identified, highlighting the need for ongoing attention to enhance the user experience.

The topics covered were highly relevant, with 96% of students confirming that they aligned with their educational goals. This highlights the lab's effective approach to training in PLDs. The majority (86%) expressed that the platform significantly improved their understanding and skills, emphasizing its alignment with educational goals. However, the survey also identified areas for improvement in usability, as 4% of students found the interaction very difficult.

Insights from the survey reveal that 94% of students expressed interest in receiving additional tutorials or guides on how to use the platform, while 6% were not interested. Regarding the theoretical content, 58% of respondents reported being satisfied, and 30% indicated they were very satisfied. This suggests that, while most students view the content positively, there is room for improvement.

An analysis of the personal feedback in response to one of the survey questions was conducted, focusing on students who reported difficulties. It was revealed that these students encountered problems while using the platform. This indicates the need to enhance the user experience design to address these specific challenges, ensuring optimal accessibility and usability for all users despite the majority having a positive experience.

The remote FPGA lab is a valuable educational tool for teaching PLD. However, it is essential to continue working on the identified areas to maximize their effectiveness and align them with the Sustainable Development Goals, especially regarding inclusive and quality education.

8. Limitations and Future Research

Remote labs are submerged in a digital world where latency is important. Latency is a technological term, and it is the time any action or operation needs to be executed. Low latency means higher speed and low execution time. A low latency would be beneficial when putting remote labs in operation. It is also crucial to have low latency, considering there will be a few hundred users, and the data could be vast. That is why it is considered that the system's latency be studied to improve performance and user interaction before the remote lab is released to a larger community, whether it is the same faculty, institution, or schools elsewhere.

In addition, the survey revealed that students found the proposal to introduce remote laboratories into their semester classes both interesting and viable. They recommended incorporating more labs focused on microcontrollers, Programmable Logic Controllers (PLCs), sensors, actuators, and peripherals such as buttons, switches, and LEDs. To meet this demand, plans are underway to establish new laboratories that will cover the practical aspects of various subjects, starting with a lab designed for remote interaction with a Collaborative Robot (COBOT).

On the other hand, important observations regarding the UAZ Labs platform interface need immediate attention. Efforts will be made to enhance the user experience by making the interface more intuitive, ensuring smoother navigation, providing instant feedback on user actions, improving image quality, and offering access to technical support. Furthermore, functionality will be improved to allow students to easily add, delete, or modify their data and manage appointments more efficiently.

This remote lab is in an early phase of the project. The purpose was to evaluate the feasibility of remote labs and recommend user improvements before deploying them for students outside the Autonomous University of Zacatecas. The next phases include more

improvements to the overall system (functionality, interface, accessibility, democratization, among others).

9. Conclusions

This work presents a remote laboratory for motor control using the Basys 3 FPGA, a Raspberry Pi 5, a cloud management system, and a web platform. It enables students to upload and execute digital hardware designs from a distance. Through the platform, students can manage a DC motor, a servo motor, and a stepper motor and monitor their performance in real time via live streaming using the YouTube API. While the current emphasis is on motor control, the lab system can be expanded to accommodate other FPGA applications and tailored to different kinds of uses, thereby providing greater flexibility and remote learning opportunities.

The research's structure and findings present an integral focus that combines technical development and empirical evaluation. The research hypotheses are implicitly approached by comparing the students' experiences and the remote lab's operative success, as seen in the survey analysis and result discussions.

The practical analysis is aligned with the research's aims and objectives, which enable us to determine whether remote laboratories have a value with a significance that is equal to or complementary to in-site labs, promoting accessibility and ability acquisition, especially in the context of Education 4.0.

The lab was initially created for use at the Autonomous University of Zacatecas in Mexico. However, its vision is to have a global impact by providing equitable access to technical learning resources and overcoming geographic and economic barriers. It incorporates emerging technologies like IoT and Cloud Computing to ensure an interactive and accessible educational experience. To validate the concept, a one-month pilot test was conducted with students with a bachelor's degree in robotics and mechatronics at the same university, and it focused on PLDs. Although it was designed for a bachelor's degree, the lab can be adapted to other subjects and programs, and it can even be used in high school-level institutions, not exclusively universities, enabling the platform for more institutions.

A survey was conducted to evaluate student satisfaction with the platform, allowing us to gauge their likelihood of recommending it and gather their opinions. The feedback we received indicated a high overall satisfaction level and identified improvement areas. Participants generally agreed that the user experience on the platform needs optimization, particularly in terms of navigation, interface, and accessibility. This feedback is essential for guiding future updates and improvements to ensure that the platform not only meets current expectations but also adapts to the evolving needs of students. Additionally, we plan to develop more laboratories operating on the same platform that align with student recommendations. This expansion will provide a wider range of learning options and deliver a more comprehensive and accessible educational experience. Also, the efforts in this development were focused on digital signal applications, but remote labs have the potential to transform most engineering in-site labs into remote labs. In addition, adopting remote labs for teaching practical classes allows students to manage their time since they can practice 24/7 from anywhere they demand. Moreover, institutions have the opportunity to transform in-person teaching carriers into online teaching carriers without the lack of practical lab learning.

Author Contributions: Conceptualization, H.A.G.-O. and M.E.M.-R.; methodology, S.I.-D. and M.E.M.-R.; software, F.G.-V., V.I.R.-A., and S.I.-D.; validation, S.I.-D., J.A.N.-P., and V.I.R.-A.; formal analysis, H.A.G.-O., R.C.-M., and M.E.M.-R.; investigation, S.I.-D., F.G.-V., and G.O.-V.; resources, G.O.-V. and R.C.-M.; data curation, J.A.N.-P. and V.I.R.-A.; writing—original draft preparation, H.A.G.-O. and F.G.-V.; writing—review and editing, H.A.G.-O., F.G.-V., and L.O.S.-S.; visualization, J.A.N.-P. and L.O.S.-S.; supervision, M.E.M.-R. and L.O.S.-S.; project administration, H.A.G.-O. and M.E.M.-R.; funding acquisition, G.O.-V. and R.C.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding .

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author .

Acknowledgments: The authors want to thank the Mexican National Council of Science and Technology CONAHCYT for its support to the National Laboratory of Embedded Systems, Advanced Electronics Design, and Micro Systems (LN-SEDEAM by its initials in Spanish), Project Numbers 282357, 293384, 299061, 314841, 315947, and 321128 and Scholarship Numbers 1012274 and 805876.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application Programming Interface
CC	Cloud Computing
COBOT	Collaborative Robot
DC	Direct Current
FPGAs	Field Programmable Gate Arrays
GPIOs	General Purpose Input/Output
HDL	Hardware Description Languages
HLS	HTTP Live Streaming
IoT	Internet of Things
PLCs	Programmable Logic Controllers
PLDs	Programmable Logic Devices
PCB	Printed Circuit Board
PWM	Pulse-Width Modulation
RTMP	Real-Time Messaging Protocol
SD	Secure Digital
SDGs	Sustainable Development Goals
SSDs	Solid State Drive
STEM	Science, Technology, Engineering, and Mathematics
UAZ	Autonomous University of Zacatecas
UN	United Nations
URL	Uniform Resource Locator
WHO	World Health Organization

References

1. Bhute, V.J.; Inguva, P.; Shah, U.; Brechtelsbauer, C. Transforming traditional teaching laboratories for effective remote delivery—A review. *Educ. Chem. Eng.* **2021**, *35*, 96–104. [CrossRef]
2. Van den Beemt, A.; Groothuijsen, S.; Ozkan, L.; Hendrix, W. Remote labs in higher engineering education: Engaging students with active learning pedagogy. *J. Comput. High. Educ.* **2023**, *35*, 320–340. [CrossRef] [PubMed]
3. Magyari, A.; Chen Y. FPGA Remote Laboratory Using IoT Approaches. *Electronics* **2021**, *10*, 2229. [CrossRef]
4. Villanueva, O.; Zimmermann, K. Transitioning an upper-level, integrated laboratory course to remote and online instruction during the COVID-19 pandemic. *J. Chem. Educ.* **2020**, *97*, 3114–3120. [CrossRef]
5. Mohammed, A.K.; El Zoghby, H.M.; Elmesalawy, M.M. Remote controlled laboratory experiments for engineering education in the post-COVID-19 era: Concept and example. In Proceedings of the 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 24–26 October 2020; pp. 629–634.
6. Mystakidis, S.; Christopoulos, A.; Pellas, N. A systematic mapping review of augmented reality applications to support STEM learning in higher education. *Educ. Inf. Technol.* **2022**, *27*, 1883–1927. [CrossRef]
7. Gunasekara, M.; Maddumapatabandi, T.; Gamage, K. Remote lab activities in a digital age: Insights into current practices and future potentials. *J. Educ. Innov. Commun.* **2021**, *3*, 59–78. [CrossRef]
8. May, D.; Morkos, B.; Jackson, A.; Hunsu, N.J.; Ingalls, A.; Beyette, F. Rapid transition of traditionally hands-on labs to online instruction in engineering courses. *Eur. J. Eng. Educ.* **2023**, *48*, 842–860. [CrossRef]

9. Krūmiņš, D.; Schumann, S.; Vunder, V.; Põlluäär, R.; Laht, K.; Raudmäe, R.; Aabloo, A.; Kruusamäe, K. Open remote web lab for learning robotics and ROS with physical and simulated robots in an authentic developer environment. *IEEE Trans. Learn. Technol.* **2024**, *17*, 1325–1338. [CrossRef]
10. Rejón, C.; Martin, S.; Robles-Gómez, A. Easy Development of Industry 4.0 Remote Labs. *Electronics* **2024**, *13*, 1508. [CrossRef]
11. Chaka, C. Is Education 4.0 a sufficient innovative, and disruptive educational trend to promote sustainable open education for higher education institutions? A review of literature trends. *Front. Educ.* **2022**, *7*, 824976. [CrossRef]
12. Mukul, E.; Büyüközkan, G. Digital transformation in education: A systematic review of education 4.0. *Technol. Forecast. Soc. Chang.* **2023**, *194*, 122664. [CrossRef]
13. Miranda, J.; Navarrete, C.; Noguez, J.; Molina-Espinosa, J.M.; Ramírez-Montoya, M.S.; Navarro-Tuch, S.A.; Molina, A. The core components of education 4.0 in higher education: Three case studies in engineering education. *Comput. Electr. Eng.* **2021**, *93*, 107278. [CrossRef]
14. Werner, E.; Matias, J.C.; Berejuck, M.D.; Saliah-Hassane, H. Evaluation of blockchain techniques to ensure secure access on remote FPGA laboratories. In Proceedings of the 2021 9th International Symposium on Digital Forensics and Security (ISDFS), Elazig, Turkey, 28–29 July 2021; pp. 1–6.
15. Aitor, V.M.; García-Zubía, J.; Angulo, I.; Rodríguez-Gil, L. Toward widespread remote laboratories: Evaluating the effectiveness of a replication-based architecture for real-world multiinstitutional usage. *IEEE Access* **2022**, *10*, 86298–86317. [CrossRef]
16. Ruo Roch, M.; Martina, M. VirtLAB: A low-cost platform for electronics lab experiments. *Sensors* **2022**, *22*, 4840. [CrossRef]
17. Monzo, C.; Cobo, G.; Morán, J.A.; Santamaría, E.; García-Solórzano, D. Remote Laboratory for Online Engineering Education: The RLAB-UOC-FPGA Case Study. *Electronics* **2021**, *10*, 1072. [CrossRef]
18. Mayoz, C.A.; da Silva Beraldo, A.L.; Villar-Martinez, A.; Rodriguez-Gil, L.; de Souza Seron, W.F.M.; Orduña, P. FPGA remote laboratory: Experience of a shared laboratory between UPNA and UNIFESP. In Proceedings of the 2020 XIV Technologies Applied to Electronics Teaching Conference (TAEE), Porto, Portugal, 8–10 July 2020; pp. 1–8.
19. Domski, W. Remote Laboratory Offered as Hardware-as-a-Service Infrastructure. *Electronics* **2022**, *11*, 1568. [CrossRef]
20. Cornelio, O.M.; González, J.G.; Ching, I.S.; Fonseca, B.B. Remote laboratory system for automatic engineering. *J. Cybersecur. Inf. Manag.* **2020**, *2*, 55–63.
21. Kustija, J.; Jayanto, N.D. IoT Implementation for Development of Remote Laboratory (Case Study on Microscope Practice). *REKA ELKOMIKA J. Pengabdi. Kpd. Masy.* **2022**, *3*, 20–29. [CrossRef]
22. Vitliemov, P.; Bratanov, D.; Marinov, M. An approach to use virtual and remote labs in mechatronics education based on cloud services. In Proceedings of the 2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE), Ruse, Bulgaria, 12–14 November 2020; pp. 1–4.
23. Ramya, M.V.; Purushothama, G.K.; Prakash, K.R. Design and implementation of IoT based remote laboratory for sensor experiments. *Int. Assoc. Online Eng.* **2020**, *14*, 227–238.
24. Abekiri, N.; Rachdy, A.; Ajaamoum, M.; Nassiri, B.; Elmahni, L.; Oubail, Y. Platform for hands-on remote labs based on the ESP32 and NOD-red. *Sci. Afr.* **2023**, *19*, e01502. [CrossRef]
25. Golubev, L.P.; Tkach, M.M.; Makatora, D.A. Using tinkercad to support online the laboratory work on the design of microprocessor systems at technical university. *Inf. Technol. Learn. Tools* **2023**, *93*, 80.
26. Al Qassem, L.M.; Stouraitis, T.; Damiani, E.; Elfadel, I.A.M. A remote FPGA laboratory as a cloud microservice. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020; pp. 1–5.
27. Alhammami, M. FPGA hardware kit for remote training platforms. *Discov. Educ.* **2024**, *3*, 102. [CrossRef]
28. Navas-González, R.; Oballe-Peinado, Ó.; Castellanos-Ramos, J.; Rosas-Cervantes, D.; Sánchez-Durán, J.A. Practice Projects for an FPGA-Based Remote Laboratory to Teach and Learn Digital Electronics. *Information* **2023**, *14*, 558. [CrossRef]
29. Melosik, M.; Naumowicz, M.; Kropidłowski, M.; Marszalek, W. Remote prototyping of FPGA-based devices in the IoT concept during the COVID-19 pandemic. *Electronics* **2022**, *11*, 1497. [CrossRef]
30. Valencia de Almeida, F.; Hayashi, V.T.; Arakaki, R.; Midorikawa, E.; Canovas, S.D.M.; Cugnasca, P.S.; Corrêa, P.L.P. Teaching digital electronics during the COVID-19 pandemic via a remote lab. *Sensors* **2022**, *22*, 6944. [CrossRef]
31. Sum, R.; Suwansantisuk, W.; Kumhom, P. Remote field-programmable gate array laboratory for signal acquisition and design verification. *Int. J. Electr. Comput. Eng.* **2024**, *14*, 2344–2360. [CrossRef]
32. Mejías, A.; Herrera, R.S.; Márquez, M.A.; Calderón, A.J.; González, I.; Andújar, J.M. Easy handling of sensors and actuators over TCP/IP networks by open source hardware/software. *Sensors* **2017**, *17*, 94. [CrossRef]
33. Reid, D.P.; Burridge, J.; Lowe, D.B.; Drysdale, T.D. Open-source remote laboratory experiments for controls engineering education. *Int. J. Mech. Eng. Educ.* **2022**, *50*, 828–848. [CrossRef]
34. Guerrero-Ulloa, G.; Rodríguez-Domínguez, C.; Hornos, M.J. Agile Methodologies Applied to the Development of Internet of Things (IoT)-Based Systems: A Review. *Sensors* **2023**, *23*, 790. [CrossRef]
35. Basys 3. Available online: https://digilent.com/ (accessed on 10 October 2024).
36. Raspberry Pi. Available online: https://www.raspberrypi.com/products/raspberry-pi-5/ (accessed on 10 October 2024).
37. Firebase. Available online: https://firebase.google.com/?hl=es-419 (accessed on 10 October 2024).
38. React. Available online: https://es.react.dev/ (accessed on 10 October 2024).
39. Youtube Studio. Available online: https://studio.youtube.com/ (accessed on 10 October 2024).
40. OpenFPGALoader. Available online: https://github.com/trabucayre/openFPGALoader/ (accessed on 10 October 2024).

41. *ISO/IEC 27001:2022*; Information Security Management. ISO: Geneva, Switzerland, 2022.
42. Castañeda-Miranda, V.H.; Luque-Vega, L.F.; Lopez-Neri, E.; Nava-Pintor, J.A.; Guerrero-Osuna, H.A.; Ornelas-Vargas, G. Two-Dimensional Cartesian Coordinate System Educational Toolkit: 2D-CACSET. *Sensors* **2021**, *21*, 6304. [CrossRef]
43. Uchima-Marin, C.; Murillo, J.; Salvador-Acosta, L.; Acosta-Vargas, P. Integration of Technological Tools in Teaching Statistics. *Sustainability* **2024**, *16*, 8344. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.