

Raspberry Pi Controller for Remote Laboratory Hardware Access

Adrian Sieber, Nicolas Hentschel, Isabel Heinze, Theo Kaminsky, Michelle Marasas,
Rico Beier-Grunwald, Alexander Lampe, Marc Ritter, Christian Roschke, Matthias Vodel
Hochschule Mittweida - University of Applied Sciences

Technikumplatz 16, 09648 Mittweida, Germany

Email: {esieber, nhentsch, iheinze, tkaminsk, mmarasas, beier1, lampe, ritter, roschke, vodel}@hs-mittweida.de

Abstract—Throughout the pandemic, institutional education switched from face-to-face lectures and practical application to remote learning facilitated by learning management platforms and remote laboratory systems. While conventional lecture formats were replaceable, especially in engineering laboratory work was limited by available remote laboratory systems, the possibility to integrate them as well as their range of functions and financial attractiveness. We introduce RaspCon, an easily implementable Raspberry Pi controlled web-application based on open-source software, as an extension to the LabCon system, allowing not only for remote access to real laboratory hardware but also the reconfiguration of its interconnections.

Keywords—digital learning; e-learning; raspberry pi; remote hardware access; web application.

I. INTRODUCTION

In response to the challenges posed by the COVID pandemic, the landscape of education underwent a rapid transformation from traditional face-to-face instruction to remote learning modalities. This shift necessitated the utilization of digital platforms, such as video conferencing software like Zoom and Microsoft Teams, alongside Learning Management Systems (LMS), such as Moodle, Opal, and Adobe Captivate. However, work-based learning activities, like laboratory work, faced considerable hurdles in adapting to remote environments. Remote Lab Systems (RLS) that allow and manage access to professional laboratory equipment are offered by several companies [1][2], but these often incur substantial costs. Moreover, adapting them to support existing laboratory practicals or integrating them into standard LMS platforms, including user authorisation, has proven to be partly challenging.

In recent years, a proliferation of cost-effective solutions has emerged, leveraging open-source software alongside readily available, affordable hardware like Arduino or Raspberry Pi boards. These solutions have been tailored to cater to a diverse array of basic experimental applications, as exemplified in [3][4][5] and references therein. Additionally, remote laboratories featuring commercial-grade industrial hardware have been developed. For instance, Grodotzki et al. introduced a remote lab designed for conducting automation and control experiments with actual robots [6]. Similarly, García et al. described a tool for managing PCs utilized in student experiments [7].

Achilles et al. developed LabCon [8], a web-based RLS integrated with LMS functionality based on open-source software that enables authorised users to gain access to laboratory hardware. Employing the LabCon system for remote experimentation involving multiple Measurement Objects (MOs)

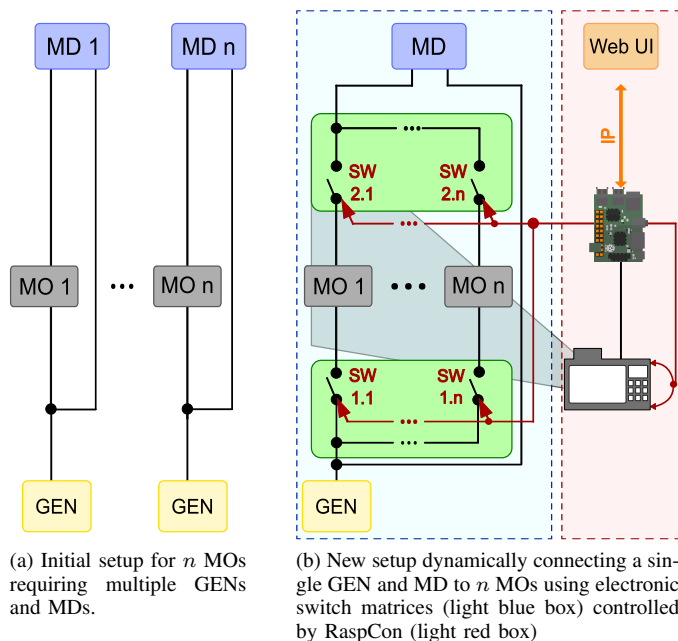


Figure 1. Comparison of initial and new LabCon hardware setup.

typically requires the hard wiring of one signal source or Generator (GEN) and one Measurement Device (MD) for each measurement object, as depicted in Figure 1a. Any adjustments to the configuration demand manual intervention, compelling the physical presence of an operator within the laboratory premises. To enhance flexibility, eliminate the necessity for manual hardware reconfiguration, and reduce the number of instruments needed, digitally controllable electronic hardware switch matrices have been developed [9]. As highlighted in light blue in Figure 1b, these matrices facilitate the routing of the generator signal to the input of any of the multiple measurement objects and connecting any measurement object's output to an input channel of the measurement device.

To facilitate remote operation of this enhanced hardware setup, an open-source software based application named RaspCon was developed, as shown within the light red box in Figure 1b. RaspCon enables intuitive web-based remote control of the electronic hardware switches and integrates with a rotatable camera to provide a live stream from the laboratory for visual inspection. In this paper, we will elucidate the motivation behind and intricacies of RaspCon, detailing its integration with the initial LabCon system to form the extended LabCon framework. Section II introduces the given requirements and

outlines the derived architecture of the RaspCon system. Section III describes the hardware components of RaspCon and their specifications. The controller software and Web UI are presented in Sections IV and V, respectively. Results from a user evaluation are presented in Section VI. Finally, Section VII provides a summary and outlook.

II. REQUIREMENTS AND SYSTEM ARCHITECTURE

As previously outlined, RaspCon is designed to allow for intuitive remote control of electronic switch matrices via a Web-based User Interface (Web UI), provide live camera streams from the laboratory, and seamlessly integrate into the existing LabCon framework. Consequently, the major components of RaspCon must support the following key features:

- Controller Hardware:
 - Host an IP-accessible web application.
 - Provide a sufficient number of programmable output pins to control external hardware.
 - Include a high-bandwidth camera connector.
 - Be scalable and cost-efficient.
- Controller Software:
 - Generate and apply digital control signals for hardware switches.
 - Manage preset digital experimental setups.
 - Support the streaming of live video signals. Offer a robust web server.
 - Be seamlessly integrable into the existing LabCon system to leverage its user access functions (see [8]).
 - Use open-source software components.
- Web UI:
 - Support intuitive usage for ease of operation.
 - Be resilient to misuse to ensure system integrity.
 - Provide a positive user experience.

The architectural overview in Figure 2 illustrates the integration of the current LabCon system with the new laboratory hardware setup and the RaspCon controller.

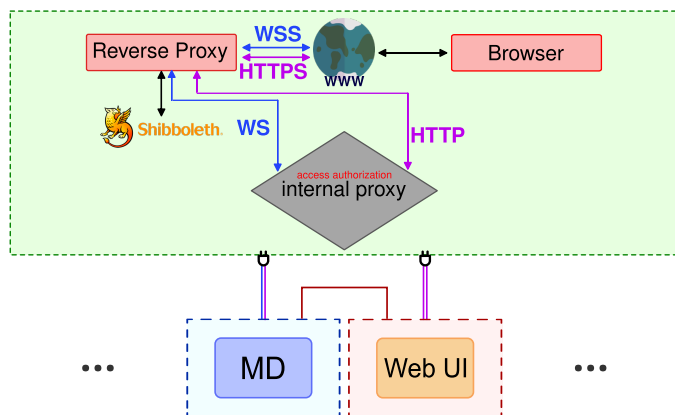


Figure 2. LabCon System Architecture: current LabCon system (light green box), LabCon laboratory hardware setup with electronic switch matrices (light blue box) and RaspCon controller (light red box).

III. CONTROLLER HARDWARE AND CAMERA MODULE

A careful selection process was undertaken to ensure that the hardware controller would meet all requirements, especially in communicating with the electronic switch matrix, hosting the web application, and supporting video streaming.

Several single-board computers were evaluated for their suitability as controller hardware, including the Arduino, Raspberry Pi, and Banana Pi. Ultimately, the Raspberry Pi Model 4B was chosen for its superior performance and versatility. This decision was driven primarily by two key factors: Firstly, its robust processing capabilities that enable the hosting of a full Linux operating system like Raspberry Pi OS, thus providing flexibility beyond embedded programming languages when choosing the software stack. Secondly, its rich range of hardware interfaces and General-Purpose Input-Output (GPIO) pins necessary for controlling both current and potential laboratory hardware, servo motors and camera. Furthermore, the Raspberry Pi’s widespread adoption ensures an extensive array of hardware extensions and libraries, reducing the need for custom hardware extension development [5]. Additionally, the Raspberry Pi’s accessibility and long-term support were pivotal considerations.

In addition to the Raspberry Pi single board computer, a Raspberry Pi 2 camera module was selected to remotely monitor the laboratory experiments. Should there be a need to adjust the camera specifications in the future, the system can seamlessly accommodate any device compatible with Video4Linux2 (V4L2). The camera module itself is mounted on a pan-tilt HAT, allowing for dynamic adjustment of the viewing area through rotation and tilting. This functionality is achieved using servo motors controlled via pulse-width modulated signals.

IV. CONTROLLER SOFTWARE

An overview of the RaspCon software stack is depicted in Figure 3. In our pursuit of an efficient and developer-friendly solution, we opted for the fullstack framework SvelteKits the foundation of our application. SvelteKit, built upon the innovative Svelte framework and running on top of the Javascript runtime environment Node.js, offers a multitude of advantages that align closely with our project requirements. One of the key

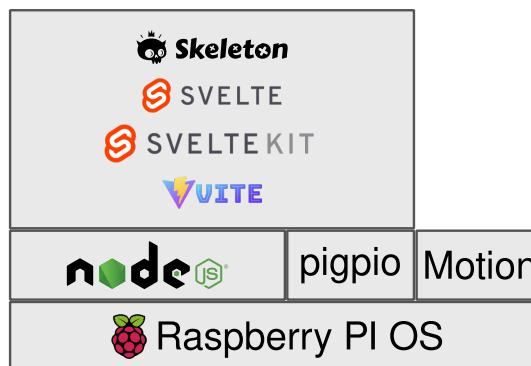


Figure 3. RaspCon software stack.

benefits is its developer-friendly nature resulting in minimal boilerplate code. Leveraging a component-based architecture that seamlessly and type-safely integrates server and client functionality, it facilitates modular development and an overall cohesive development experience. This not only streamlines the development process but also enhances scalability and maintainability. Furthermore, SvelteKit's filesystem routing simplifies the navigation structure of our application, enhancing code clarity and reducing complexity. The framework's built-in support for automatic Server-Side Rendering (SSR) ensures optimal performance and user experience.

The hardware configuration data exhibits a document-like structure and is of relatively low size and complexity. Consequently, the use of a traditional relational or even document-based database is unnecessary. Instead, we employ JSON structures as a lightweight yet effective solution. An advantageous aspect of this approach is its facilitation of exporting configuration subsets, such as presets, as JSON files. This enables users to locally store and transfer their presets to other instances with similar hardware setups.

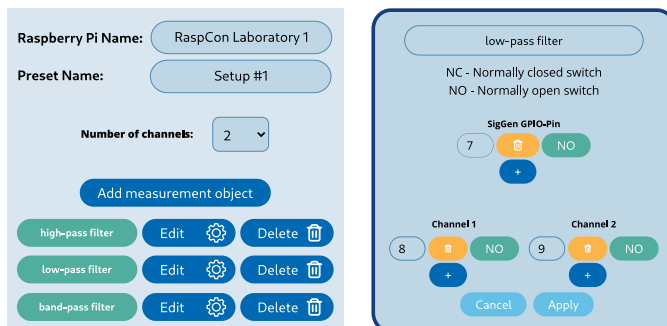
Although the system was designed as a web application, it required low-level access to the Raspberry Pi's GPIO pins to control both the electronic switch matrices and camera servo motors. This functionality was achieved by utilizing the C library pigpio, accessed through a JavaScript wrapper library. We constructed higher-level abstractions for all required hardware functions. While accessing the camera interface, transcoding video material and serving a live stream within JavaScript can be complex, we have found a straightforward solution with the software Motion. Utilizing this standard application enables us to effortlessly serve the video stream as MJPEG over HTTP, streamlining our implementation process.

V. WEB USER INTERFACE

In the context of remote laboratory access, particularly for users situated beyond the confines of the university, the RaspCon application introduces a Web UI acting as a client interface through the integration with the LabCon RLS. This Web UI is designed to provide a seamless user experience across three key pages:

- **Homepage:** Introduces the RaspCon System, succinctly outlining its purpose and capabilities for remote experimentation, thereby orienting users to the system's functionality.
- **Configuration Page:** Reserved for administrators, this page offers a suite of tools for setting up and managing experimental configurations.
- **Measurement Object Page:** Enables students to engage in experiments by connecting measurement objects with measurement devices, enhanced by a live lab camera feed.

In the RaspCon application, the configuration page serves as a key interface for administrators, facilitating the setup of experimental environments at the start of each semester. This feature allows for the creation of a digital twin of the actual laboratory setup, incorporating the configuration of measurement objects and measurement devices to accurately



(a) Definition of a preset for connected measurement objects.

(b) Assignment of GPIO pins that control the switches for one measurement object.

Figure 4. Configuration page of an experimental setup.

reflect the physical lab arrangement (Figure 4a). The process includes simulating electronic switch matrices by assigning the GPIO pins controlling the individual switches and defining their initial states. This emulation helps to replicate the operation of individual switches, effectively merging the virtual and physical laboratory experiences (Figure 4b), thereby enhancing the digital twin's realism. Additionally, the configuration page offers comprehensive tools for managing experimental setups. Administrators can export and import configurations locally, and save and load them from the RaspCon server. This flexibility supports easy sharing and replication of setups across semesters, allowing for efficient organisation and retrieval of diverse experimental environments.

In the RaspCon system, the measurement object page is specifically designed for students to interact directly with laboratory hardware when conducting their assigned practicals during the semester, underpinned by configurations set by administrators before. This interface allows students to connect selected measurements objects with necessary measurement devices for a specific experiment, using a visual representation of the switch matrix displayed in table form (Figure 5). Through this interface, connections between measurement objects and measurement devices can be easily established or removed by clicking a button corresponding to the desired interaction, ensuring a straightforward and intuitive user experience. To maintain the integrity of experimental setups and prevent misuse, the system automatically removes an existing connection between a measurement object and a measurement device channel when a new measurement object is assigned to that channel. The measurement object page also integrates a live stream feature of the laboratory, enhancing the hands-on experiment with visual feedback. Students can manipulate the camera angle using arrow buttons, offering personalized views of the laboratory setup matching their experimental focus, or reset the camera to a standard position with a dedicated button (Figure 6). This live feed allows students to monitor the actual laboratory environment in real-time fostering interactive

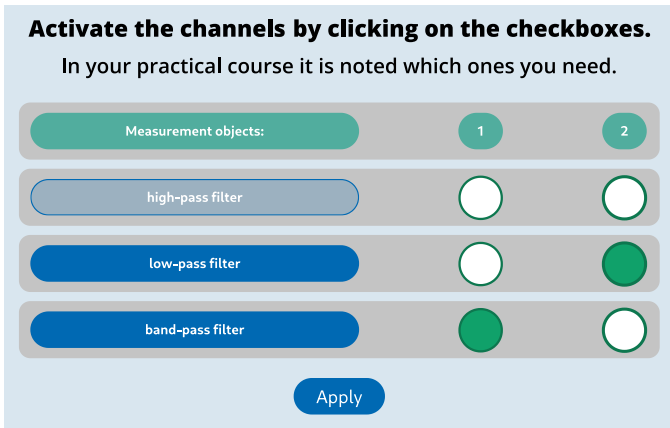


Figure 5. Example switch matrix table used to select active connections (green circles) between measurement objects and measurement device input channels.



Figure 6. Integrated camera live stream.

learning and simulating physical presence in the laboratory, despite remote access.

The development of the Web UI for the RaspCon application was achieved through the utilization of Skeleton (Figure 3), a development toolkit tailored for integration with the Svelte library and enhanced by the capabilities of Tailwind CSS. Skeleton’s provision of pre-configured components significantly streamlined the process of constructing consistent and contemporary user interfaces. A notable feature includes the implementation of a theme switcher, which greatly simplified the introduction of a dark mode. Moreover, the toolkit’s predefined drawer and modal components were instrumental in organizing the pages’ layout, ensuring a user-friendly and intuitive interface. To safeguard against unintentional modifications, e.g., of the experimental configurations currently in use, the Web UI incorporates confirmation dialog windows that prompt users for verification before proceeding with the deletion of settings.

VI. QUALITY CONTROL AND USER FEEDBACK

To maintain a high standard of quality for RaspCon, the web application underwent evaluation by its potential end-

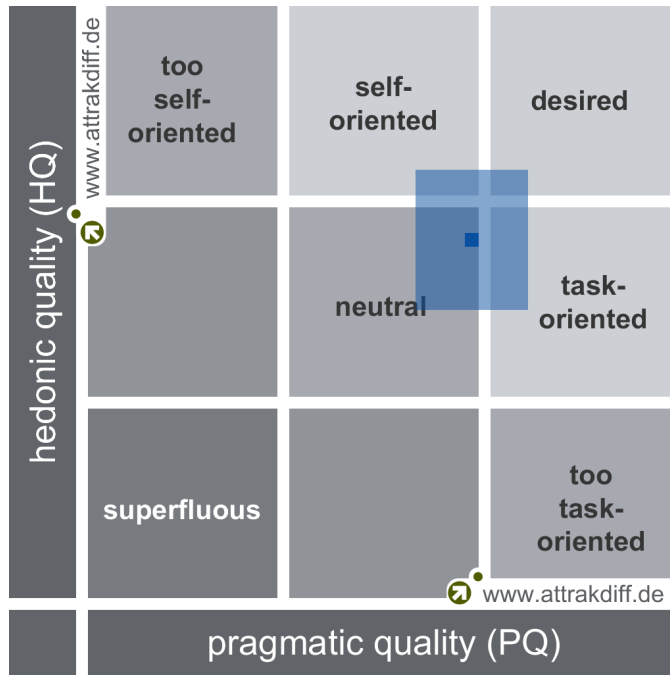


Figure 7. Portfolio format of RaspCon evaluation assessing hedonic and pragmatic quality.

users, including students and administrators, on two occasions. The objective of these evaluations was to ensure that RaspCon offered an easy and intuitive user experience and to refine the application based on the feedback received.

For the efficient collection and analysis of feedback across all test groups, the web based tool AttrakDiff [10] was utilized. AttrakDiff is a questionnaire-based tool used to measure the perceived usability and aesthetic appeal of products and interfaces by assessing their pragmatic and hedonic qualities. It enabled the systematic gathering of responses through a structured questionnaire completed by each participant. The results of this evaluation are depicted in a portfolio format, as shown in Figure 7.

The analysis reveals that RaspCon is perceived positively, demonstrating a tendency towards a favorable user experience characterized by both hedonic qualities (such as perceived stimulation and identification with the user) and pragmatic qualities (emphasizing usability and achievement of objectives).

VII. CONCLUSION AND FUTURE WORK

The article introduces RaspCon, a scalable module designed to enhance remote laboratory experimentation within the Lab-Con RLS. It provides a web-based platform for administrators to configure and manage laboratory setups, allowing flexibility and interchangeability for educational needs. This system enables students to perform remote experiments, facilitating dynamic adjustments to experimental configurations as required. To further enrich the educational journey, the interaction capabilities will be extended and an AI agent is to

be integrated providing dedicated support for the laboratory experiments.

REFERENCES

- [1] P. Orduña, L. Rodríguez-Gil, J. García-Zubia, I. Angulo, U. Hernández, and E. Azcuenaga, "LabsLand: A sharing economy platform to promote educational remote laboratories maintainability, sustainability and adoption," in *2016 IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–6.
- [2] Keysight Technologies, *PathWave Lab Operations for Remote Learning, PW9112EDU*, 2021, retrieved: April, 2024. [Online]. Available: <https://www.keysight.com/de/de/assets/3121-1016/data-sheets/PathWave-Lab-Operations-for-Remote-Learning-PW9112EDU.pdf>.
- [3] J. A. Ariza and C. N. Galvis, "RaspyControl Lab: A fully open-source and real-time remote laboratory for education in automatic control systems using Raspberry Pi and Python," *HardwareX*, vol. 13, p. e00396, 2023.
- [4] M. C. Niqin, N. C. Puelles, M. C. Roman, J. V. Araoz, V. Pimentel, and C. L. Pineda, "Low-Cost Remote Laboratory for Robotics Undergraduate Learning during Covid Context," in *2022 IEEE 2nd International Conference on Educational Technology (ICET)*, 2022, pp. 11–15.
- [5] A. Fernández-Pacheco, S. Martín, and M. Castro, "Implementation of an Arduino Remote Laboratory with Raspberry Pi," in *2019 IEEE Global Engineering Education Conference (EDUCON)*, 2019, pp. 1415–1418.
- [6] J. Grodotzki, T. R. Ortelt, and A. E. Tekkaya, "Remote and Virtual Labs for Engineering Education 4.0: Achievements of the ELLI project at the TU Dortmund University," *Procedia Manufacturing*, vol. 26, pp. 1349–1360, 2018, 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA.
- [7] S. García, A. Gallardo, D. F. Larios, E. Personal, J. M. Mora-Merchán, and A. Parejo, "Remote Lab Access: A Powerful Tool Beyond the Pandemic," in *2022 Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica (XV Technologies Applied to Electronics Teaching Conference)*, 2022, pp. 1–5.
- [8] M. Achilles *et al.*, "Enhancing Digital Learning: A User Management and Access System for Remote Laboratories," in *Proceedings of the Fifteenth International Conference on Mobile, Hybrid, and On-line Learning (eLmL 2023)*. Venice, Italy: IARIA, April 2023, pp. 44–45.
- [9] J. Zhao, "Design, simulation, prototypical implementation and test of a modul for internet-based switching between test objects in laboratory practicals," Bachelor's thesis, University of Applied Sciences Mittweida, Mittweida, Germany, 10 2023.
- [10] M. Hassenzahl, M. Burmester, and F. Koller, *AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität (AttrakDiff: A Questionnaire for Measuring Perceived Hedonic and Pragmatic Quality)*. Wiesbaden: Vieweg+Teubner Verlag, 2003, pp. 187–196.