

DOI: 10.31319/2519-2884.45.2024.16

УДК 629.7.05-049.7

Сотник О.А., викладач, ORCID: 0009-0005-8036-3075

Марченко С.В., к.ф.-м.н, ORCID: 0000-0002-6022-5071, smarsv1979@gmail.com

Литвиненко В.А., к.т.н., ORCID: 0000-0003-4950-5168

С'янов О.М., д.т.н, професор, ORCID: 0000-0003-4120-4926

Шульга В.В., здобувач першого (бакалаврського) рівня освіти
Дніпровський державний технічний університет, м. Кам'янське

Sotnyk Olexander, Lecturer of the of the Department of Electronics and Electronical Communications

Marchenko Serhii, Candidate of Physics and Mathematics Sciences, Head of the Department of Electronics and Electronical Communications

Lytvynenko Volodymyr, Candidate of Technical Sciences, Docent of the Department of Electronics and Electronical Communications

Sianov Oleksander, Doctor of technical sciences, Professor of the Department of Electronics and Electronical Communications

Shulha Vitalii, undergraduate student

Dniprovsky State Technical University, Kamianske

РОЗРОБКА ВІДДАЛЕНО-КЕРОВАНИХ ЛАБОРАТОРНИХ РОБІТ НА БАЗІ БАГАТОФУНКЦІАЛЬНОГО ПРИСТРОЮ REDPITAYA

У роботі представлено розробку дистанційної лабораторної роботи з аналогової електроніки з використанням програмно-апаратними засобами (ПАЗ) на основі багатофункціонального вимірювального пристрою RedPitaya [1]. Метою даної роботи є створення зразка веб-застосунку для дистанційного керування роботою напівпровідникового світлодіоду на базі RedPitaya із операційною системою (ОС) PetaLinux, яка містить в своєму складі пакет веб-серверу NGINX. Процес створення веб-застосунку для проведення даної дистанційної лабораторної роботи складався з розробки двох функціональних частин: інтерфейсу, який будується на основі технологій HTML, CSS, JavaScript, що забезпечує зручний та зрозумілий інтерфейс, та програмного контролеру, створеного на мові C++ для ефективної обробки даних на сервері. Отримані результати роботи тестового веб-застосунку, який керує режимом роботи світлодіоду на налагоджувальній платі RedPitaya, можуть бути використані для реалізації дистанційного керування над більш складними електронними схемами та пристроями.

Ключові слова: RedPitaya; PetaLinux; NGINX; Frontend; Backend; WebSocket; JSON; Zynq7000; система на кристалі; цифрова обробка сигналів.

In this paper the development of remote laboratory work with analog electronics using hardware and software (HSM) based on the multifunctional measuring device RedPitaya [1] has been presented. The purpose of this work is to create a web-based application for remote control of a semiconductor LED based on RedPitaya with the PetaLinux operating system (OS), which contains the NGINX web server package in its backend. The process of creating a web based application for carrying out this remote laboratory work consisted of the development of two functional parts: an interface, which has been based on HTML, CSS, JavaScript technologies to provide a convenient and understandable interface and a software controller created in the C++ language for efficient processing of data on the server. The obtained results of the test web based application, which controls the LED operation mode on the RedPitaya control board, can be used to implement remote control larger electronic circuits and devices.

Keywords: RedPitaya; PetaLinux; NGINX; Frontend; Backend; WebSocket; JSON; Zynq7000; system on a chip; digital signal processing.

Постановка проблеми

Удосконалення дистанційного навчального процесу в сучасних умовах, які склалися в Україні у зв'язку з військовим станом наразі є одним із актуальних питань в сучасній вищій школі [2]. Особливо важливим аспектом в дистанційному способі навчання є проведення лабораторних робіт здобувачами освіти в режимі реального часу з сучасними програмно-апаратними засобами (ПАЗ), що дозволяють на їх основі створювати віддалено-керовані лабораторні стенди.

У нинішніх умовах існує потреба проведення лабораторного практикуму у реальних лабораторіях без фізичного знаходження здобувачів вищої освіти із застосуванням інформаційно-комунікаційних технологій (ІКТ) [2]. На даний час зарекомендували віддалені варіанти виконання лабораторних експериментів, як реальних так і віртуальних. Реальна віддалена лабораторія — це лабораторія, яка фізично існує та керується віддалено через мережу Інтернет, а віртуальна віддалена — використовує спеціалізоване програмне забезпечення для симуляції поведінки експериментальних установок на комп'ютерах, в більшості випадків потребує тільки браузера для доступу в Інтернет.

Побудова віддаленої лабораторії базується на сучасних клієнт-серверних технологіях, що включає такі основні компоненти: розклад, автентифікація, інтерфейс користувача та база даних, яка зберігає всю інформацію про результати виконання віддалених лабораторних експериментів. Серед переваг віддалених лабораторій вирізняються наступні: економічність забезпечується за рахунок зменшення кількості лабораторного обладнання та лабораторних стендів, гнучкість робочого графіку обумовлено персоналізацією доступу до обладнання, заохочення та отримання навичок ІКТ, що спонукають до більш якісного навчання та мотивують до наукових досліджень.

У роботі запропонована та досліджена одна з потенційних методик побудови веб-застосунку на апаратній платформі RedPitaya, яка дозволяє організувати дистанційну взаємодію користувача з електронним обладнанням.

Аналіз досліджень та публікацій

За останні роки збільшилося використання віддалених лабораторій в багатьох областях науки та техніки. На міжнародному рівні існують важливі проекти, такі як Глобальний консорціум онлайн-лабораторій (GOLC) [3], мережа віддалених лабораторій (RexNet) [4], зокрема платформа iLab [5], розроблена Массачусетським технологічним інститутом (MIT). Зростаючий інтерес до цієї галузі проявляється у фінансуванні європейських дослідницьких проектів MARVEL [6], PEARL [7]. Особливу увагу в складних економічних умовах Української економіки треба приділяти недорогим системам для веб-серверів лабораторій та експериментів. Зокрема, в [8] автори описують ArPi Lab, повністю працездатну віддалену лабораторію вартістю близько 1000 євро, більша частина яких була вкладена у веб-камери, використовувані для запису експериментів. Дана система складається із Raspberry Pi в якості лабораторного серверу і Arduino YUN та UNO з Ethernet-екранами в якості експериментальних серверів.

Серед українських університетів можна відзначити Київський університет імені Бориса Грінченка, який отримав навчальну віддалену лабораторію GOLDi для вивчення основ електроніки, електротехніки та проектування систем керування [9]. Дана платформа дозволяє студентам з'єднуватися онлайн та відтворювати експерименти і отримувати реальний досвід в цифровому форматі. Наприклад, студенти можуть вивчати роботу електронних пристроїв на базі мікроконтролера та програмованої логічної інтегральної схеми.

Таким чином, створення програмних застосунків на базі програмно-апаратних засобів, що забезпечують дистанційне керування пристроями та приладами лабораторного практикуму є актуальною науково-прикладною задачею.

Формулювання мети дослідження

Метою роботи є розробка та впровадження веб-застосунку для віддаленого керування світлодіодом на основі ПАЗ RedPitaya. Таким чином в даній роботі буде представляємо принцип організації проведення віддалених лабораторних робіт на базі мультифункціонального модуля «RedPitaya» з метою висвітлення особливостей застосування даного ПАЗ в учбовому процесі.

Виклад основного матеріалу

Екосистема мультифункціонального пристрою «RedPitaya». RedPitaya [10] представляє собою компактну налагоджувальну плату з геометричними розмірами менше за розміри сучасного смартфона (рис. 1), призначена для виконання вимірювань, цифрової обробки сигналів та контролю параметрів різноманітного обладнання. Доступ до неї можна отримати використовуючи інтерфейс Ethernet та веб-браузер.

Апаратна частина. Зазначена налагоджувальна плата на базі SoC Zynq7010 від Xilinx складається з двох частин [11]: системи обробки даних (PS) і програмованої логіки (PL). Система обробки даних має двоядерний процесор ARM Cortex A9 із частотою 667 МГц, на якому запускаються операційна система Linux та виконуються додатки. SoC Zynq7010 має вбудовану двопортову оперативну пам'ять, доступну за допомогою FPGA, а основна пам'ять об'ємом 512 МБ (4 Гб) знаходиться на зовнішньому чіпі, підключеному через спеціальні інтерфейси пам'яті. Системи обробки даних PS в кристалі Zynq7010 містить в своєму складі: периферійні пристрої вводу-виводу та інтерфейси для стандартних шин, зокрема: Gigabit Ethernet (MAC, два порти USB 2.0 OTG, два контролера SD/SDIO 2.0, два SPI порти, два UART (до 1 Мбит/с), два ведучих та відомих інтерфейси I2C, а також декілька банків GPIO.

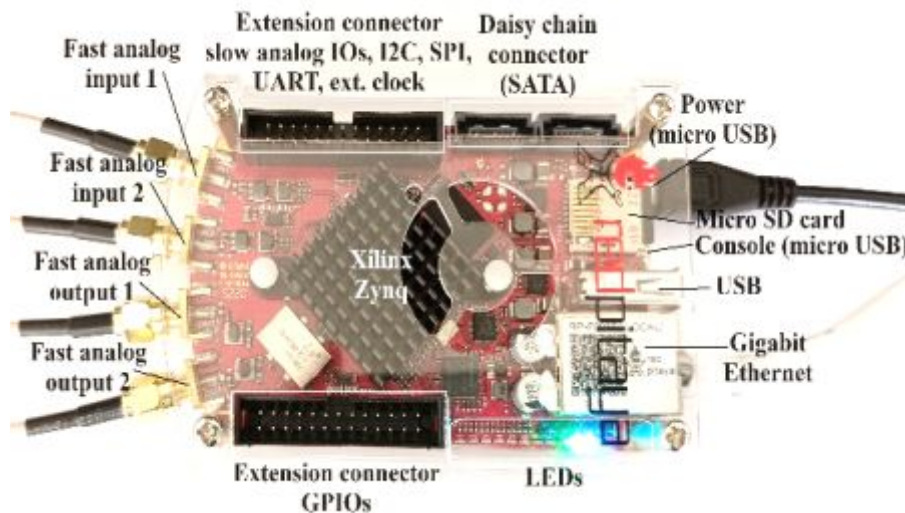


Рис. 1. Основні компоненти Red Pitaya

В якості програмованої логіки виступає FPGA з 28000 логічними комірками (17600 LUT), блоками оперативної пам'яті до 240 кБ та 80 блоками DSP. В даній FPGA реалізовано підключення високошвидкісних АЦП і ЦАП із частотою дискретизації до 125 MSPS/s із роздільною здатністю 14 біт і дозволяє реалізувати високошвидкісну обробку в реальному часі. Аналогові входи мають вхідний імпеданс 1МОм/10пФ, смугу пропускання 50 МГц, шкала вхідної напруги ± 1 або ± 20 В. ЦАП має аналогічні характеристики тільки з максимальним діапазоном напруги ± 1 В і вихідним опором 50 Ом. Крім того є чотири низько швидкісні аналогові входи та виходи з частотою дискретизації до 100 ksps та роздільною здатністю 12 біт. Таким чином RedPitaya є універсальною автономною платформою для організації експериментів із галузі телекомунікацій та електроніки. *Програмна частина* представлена по суті операційною системою (ОС) Linux [11] на базі ядра PetaLinux [12] підтримуваного компанією Xilinx для систем на кристалі (SoC) Zynq7000. Взаємодія з обладнанням здійснюється на основі даної ОС на основі клієнт-серверної взаємодії із застосування веб-серверу для взаємодії з додатками та SSH з'єднання для збірки власних додатків безпосередньо на налагоджувальній платі RedPitaya. ОС для взаємодії з обладнанням та можливості ЦОС в FPGA дозволяє змінювати файли конфігурації FPGA.

Розробка додатків за допомогою Red Pitaya. Практично всі застосунки на Red Pitaya складаються з двох частин [1, 13]: Frontend та Backend (рис. 2). Інтерфейс (Frontend) — це те, що ви можете бачити в браузері. В даному випадку для створення інтерфейсу використовується комбінація HTML5, CSS3 та JavaScript. HTML5 — для макета, CSS3 — для стилізації елементів, JavaScript для створення швидких і надійних вебзастосунків та взаємодії з серверною частиною (зокрема, дані передаються у форматі JSON). Серверна частина (Backend) представляє собою налагоджувальну плату Red Pitaya, проте в контексті користувацької програми з графічним інтерфейсом, серверна програма виступає контролером користувацької програми.

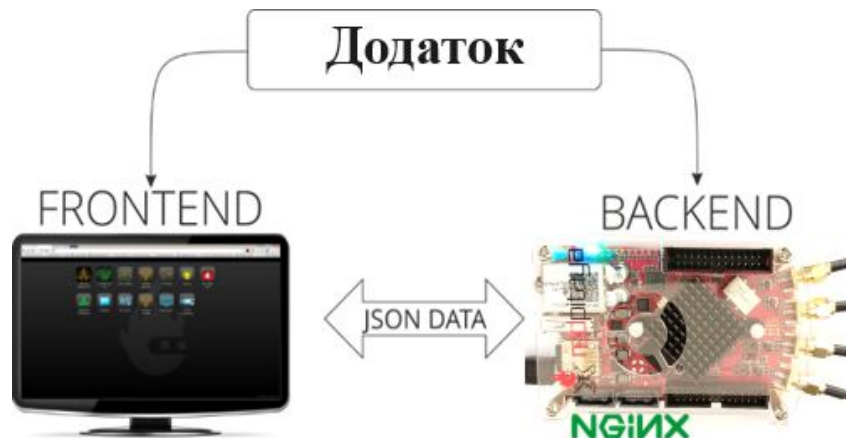


Рис. 2. Структурна схема вебзастосунків Red Pitaya

Враховуючи те, що RedPitaya призначена для застосування в студентських проектах та роботах у якості серверу було обрано сервер із відкритим кодом nginx, який дозволяє організувати взаємодію між обладнанням (Backend) та інтерфейсом користувача (Frontend) через браузер. Backend має можливість керувати світлодіодами на платі, завантажувати образи конфігурації FPGA [14], працювати з швидкими входами та виходами та багато іншого.

З метою створення гнучких та ефективних застосунків на базі Red Pitaya у якості способів взаємодії програм Frontend і Backend використовується програмний інтерфейс API (Application Programming Interface), що базується на розширеному веб-сокеті. Документація API Red Pitaya містить описи доступних функцій, їх сигнатури, приклади використання та іншу інформацію, яка допомагає розробникам ефективно використовувати можливості пристрою та інтегрувати Red Pitaya [15] у власні проекти, створювати нові додатки та розширювати функціонал пристрою.

Структура веб-додатків, які створюються на базі RedPitaya складається з трьох основних елементів, які показані на рис. 3 [16]:

- веб-інтерфейс складається з файлу HTML і JavaScript з використанням JQuery, які працюють на базі веб-серверу NGINX (для обміну даними використовується формат JSON, запити GET і POST — для взаємодії з обладнанням).

- Контролер: набір функцій скомпільованих у вигляді динамічної бібліотеки, яка завантажується модулем веб-сервера, коли потрібний будь-який зв'язок з FPGA.

- = FPGA (кожен додаток потребує своєї конфігурації): управління пристроями АЦП і ЦАП, ЦОС за допомогою мови опису апаратного забезпечення.

Крім того слід зазначити, що при вирішенні завдань пов'язаних з додатковим функціоналом стандартних додатків таких як осцилограф чи генератор сигналів, найбільш швидким та логічним рішенням буде взяти з основу вже готовий додаток розроблений RedPitaya, що включає образ конфігурації для FPGA. В свою чергу готовий образ конфігурації FPGA дозволить не застосовувати мови опису апаратури та не встановлювати додаткове спеціалізоване програмне забезпечення (ПЗ).

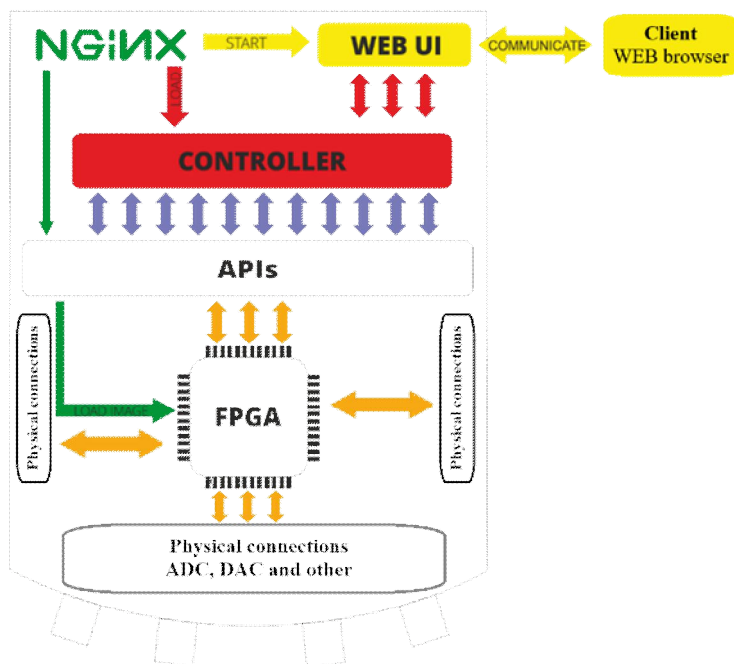


Рис. 3. Структура додатків, розроблених для Red Pitaya

Методика створення веб-застосунку для проведення дистанційної лабораторної роботи на прикладі керування режимом роботи напівпровідникового світлодіоду. Процес створення веб-застосунку для проведення віддаленої лабораторної роботи буде складатися з наступних етапів:

1. Налаштування середовища розробки
2. Збірка та інсталяція програми. Структура файлової системи та веб-застосунку.
3. Розробка веб-інтерфейсу
4. Створення програмного контролера
5. Образу конфігурації для FPGA.

Нижче буде представлена у якості прикладу методика побудови веб-застосунку для віддаленого керування режимом роботи світлодіоду.

На першому етапі налаштування середовища розробки виконується підключення до налагоджувальної плати RedPitaya через з'єднання типу SSH (рис. 4) [17], що дозволяє отримати доступ до консолі управління ОС.

Для збереження внесених змін в ОС та роботи з файловою системою в режимі читання/запису виконується команда `rw` [18]. Щоб контролювати версії файлів власних додатків та синхронізувати їх з хмарним сховищем встановлюється GitHub. З метою отримання доступу до користувацьких додатків за наступним шляхом `«/root/RedPitaya/Examples/web-tutorial/»` виконуємо клонування репозиторію Red Pitaya в папку `root` (рис. 5) [18]:

```
$ cd /root/ $ git clone https://github.com/RedPitaya/RedPitaya.git
```

Другий етап (збірки та інсталяція програми) здійснюється за допомогою наступних команд [18]:

```
$ cd /opt/redpitaya/www/apps/myFirstApp/
$ make INSTALL_DIR=/opt/redpitaya
```

Після процесу компіляції [19] контролера програми на мові C++ та її інсталяції, додаток з'являється в головному меню Red Pitaya в браузері. Необхідно зазначити, що компіляція потрібна, якщо не скомпілювали її або змінили код проєкту. Додавати функціонал до контролера необхідно до його компіляції та встановлення разом з програмою, тобто якщо змінюється лише WEB-файли перекомпілювати контролер не обов'язково.



```
rp-f098e9.local - root@rp-f098e9: ~ - VT
File Edit Setup Control Window Help
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.9.0-xilinx armv7l)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
#####
# Red Pitaya GNU/Linux Ecosystem
# Version: 1.04
# Build: 15
# Branch:
# Commit: 27498667c96692b6f15a7878462480c66c6c0abf
# U-Boot: "redpitaya-v2016.4"
# Linux Kernel: "branch-redpitaya-v2017.2"
# Pro Applications: ""
#####
Last login: Wed Nov 1 13:03:06 2023 from 192.168.0.100
root@rp-f098e9:~# rw
root@rp-f098e9:~# █
```

Рис. 4. Веб-інтерфейс програми аналізатора спектра, наданого RedPitaya

- ▼ **css - файли каскадної таблиці стилів**
 - style.css - файл відповідає за оформлення інтерфейсу користувача
- ▼ **info - файли, що містять інформацію про програму**
 - icon.png - іконка програми
 - info.json - опис програми
- ▼ **js - файли JavaScript**
 - app.js - відповідає за взаємодію інтерфейсу клієнта з сервером
 - jquery-2.1.3.min.js - бібліотека jquery
- ▼ **src - файли контролера на сервері**
 - Makefile - файл опису збірки для утиліти make
 - main.cpp - програм контролера написана на мові C++
 - main.h - файл заголовків програми контролера
 - Makefile - файл опису збірки для утиліти make
 - fpga.conf - файл конфігурації FPGA
 - fpga.sh - скрипт завантаження
 - index.html - головна сторінка додатку на стороні користувача

Рис. 5. Структура папки проекту myFirstApp

Згідно з документацією [16, 18], для маніпулювання світлодіодами за допомогою API Red Pitaya необхідно спочатку завантажити fpga_0.94.bit (відповідну конфігурацію програмованої логіки) використовуючи команду терміналу для ОС 1.04: `cat /opt/redpitaya/fpga/fpga_0.94.bit > /dev/xdevcfg`. Таким чином усі доступні конфігурації FPGA знаходяться в директорії: `/opt/redpitaya/fpga`. В свою чергу бібліотекам до яких звертаються користувацькі програми введено: `/opt/redpitaya/lib`, а самим програмам: `/opt/redpitaya/www/apps/`. У відповідності з документації [13, 16] типовий додаток RedPitaya складається з файлів HTML (структура макету) [20], CSS (каскадна таблиця стилів) [21], JavaScript (визначає логіку взаємодії клієнтської програми з сервером) [22] та C++(файли контролера на сервері) [23], опису програми у форматі JSON [24]. Крім того варто зазначити, що назва папки програми визначає унікальний ID програми, тому при вказанні ID програми даний момент необхідно обов'язково враховувати.

На третьому етапі розробляється веб-інтерфейс [16] для керування периферійними пристроями Red Pitaya. Описується його структура (файл `index.html`) та логіка взаємодії інтер-

фейсу користувача з сервером (файл `app.js`), графічне оформлення інтерфейсу (`style.css`). Алгоритм роботи програми наступний (рис. 6 і рис. 7): коли користувач натискає кнопку мишею повинен змінюватися локальний `led_state` кожного разу, коли натискається кнопка, і надсилати поточне значення `led_state` на серверну частину, щоб Red Pitaya міг оновлювати реальний стан світлодіоду.

Точкою входу у додаток JavaScript [22] є `APP.startApp()`. Надсилається запит на статус завантаження програми. Якщо статус не «ОК», запит буде надіслано повторно. Якщо програму було завантажено, програма JS намагається підключитися до Red Pitaya через WebSocket, викликаючи `APP.connectWebSocket()`.

Спочатку програма файл `app.js` перевіряє підтримку WebSocket у браузері. Потім створюється нове підключення WebSocket. Існує чотири зворотні виклики WebSocket, а саме: `APP.ws.onopen()` та `APP.ws.onclose()` викликається, коли з'єднання з WebSocket було успішно відкрито або закрито відповідно; `APP.ws.onerror()` — коли виникає помилка під час встановлення з'єднання через WebSocket; `APP.ws.onmessage()` — викликається при отриманні повідомлення.

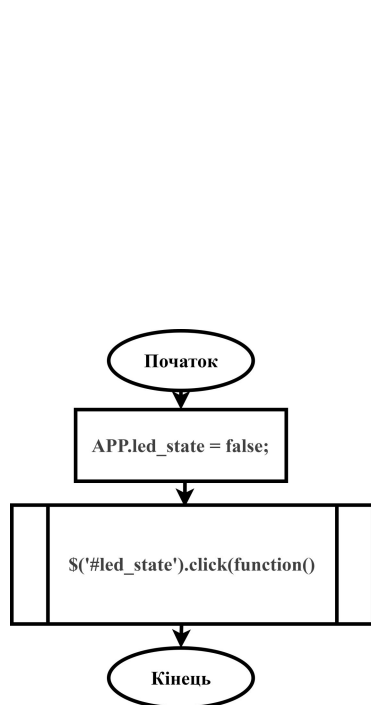


Рис. 6. Зміна стану світлодіоду при натисканні кнопки

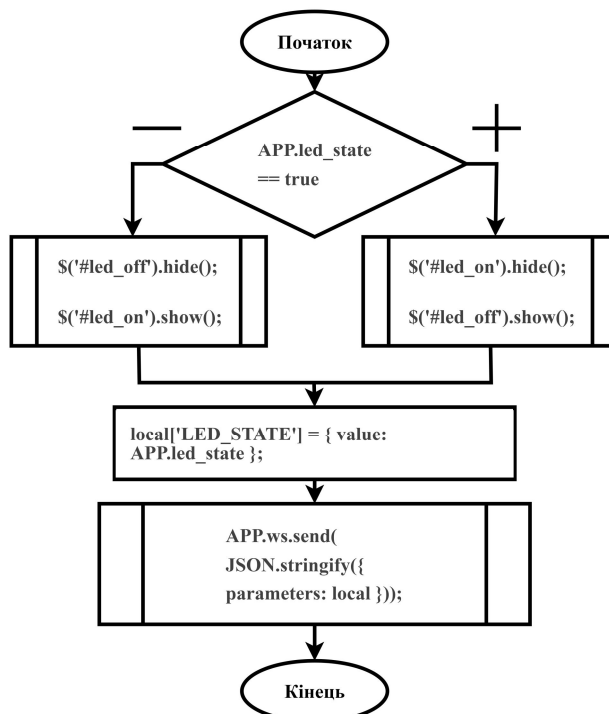


Рис. 7. Передача стану світлодіоду на сервер

Наступним етапом є створення контролера програми, який відповідає за реакцію на параметри, які були надіслані на сервер інтерфейсом користувача. Алгоритм (рис. 8) роботи контролера описується у файлі `main.cpp`. Керування режимом роботи світлодіоду виконується контролером із застосуванням глобального параметру (змінна, пов'язана з NGINX), який зчитується з сервера [16, 23]: `CBooleanParameter ledState("LED_STATE", CBaseParameter::RW, false, 0)`. Ініціалізація параметру має 4 аргументи — ім'я параметра, режим доступу, початкове значення та прапор оновлення FPGA. Ім'я параметра `LED_STATE` має бути таким же, як і в `app.js` [18], а також мати відповідний тип (`bool` — `CBooleanParameter`, `int` — `CIntParameter`, etc). Даний параметр оновлюється у функції `OnNewParams()`, яка викликається при надходженні нових параметрів.

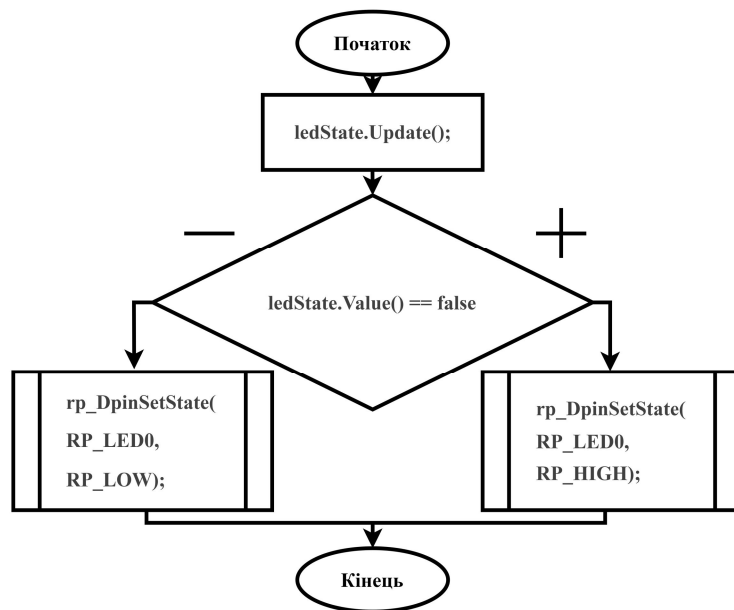


Рис. 8. Взаємодія контролера на мові C++ з обладнання на основі API RedPitaya

Функція [18, 23] `ledState.Update()` виконує оновлення значення параметру "LED_STATE", який, в свою чергу, приймає значення з серверу NGINX за назвою параметра, тому назви параметрів у контролері та `app.js` мають бути однаковими. Функція `rp_DpinSetState` є функцією із бібліотеки API Red Pitaya, яка задає стан деякого виводу інтегральної мікросхеми (ІМС). В якості аргументів даної функції виступають `rp_dpin_t pin` і `rp_pinState_t *state`.

Результати роботи віддаленого управління світлодіодом через браузер. На основі вище представленої методики створення веб-застосунку для проведення дистанційної лабораторної роботи була виконана тестова лабораторна робота керуванням режимом роботи світлодіоду на налагоджувальній платі Red Pitaya. План проведення дистанційної лабораторної роботи складався з наступних етапів: компіляція контролеру; інсталяція додатку; завантаження необхідного образу конфігурації FPGA (розглянуто вище); запуск додатку для управління світлодіодом у вікні браузера (рис. 9, а); тестування.



Рис. 9. Інтерфейс RedPitaya в браузері

Під час тестування за допомогою веб-інтерфейс розробленого додатку (LED control, рис. 9) виконувалося ввімкнення (рис. 9) та вимкнення (рис. 10) одного з світлодіодів які розташовуються на налагоджувальній платі. Вибір світлодіоду яким потрібно керувати вказується за допомогою наступних символічних позначень RP_LED0, RP_LED1 ... RP_LED7, а стан світлодіоду вказується наступним чином: RP_HIGH (увімкнено) і RP_LOW (вимкнено).



Рис. 10. Результат роботи додатку LED control : а) вимкнення світлодіоду RP_LED0; б) вимкнений світлодіод RP_LED0; в) ввімкнення світлодіоду RP_LED0; г) ввімкнення світлодіоду RP_LED0

Висновки

У роботі розглянуто основні функціональні властивості платформи RedPitaya та особливості її роботи, що дозволяє в повному обсязі оцінити можливість використання останньої у якості основи лабораторного стенду для проведення віддалених лабораторних робіт.

Переваги платформи Redpitaya включають її компактність, стабільність, економічність та можливість налаштування. Вона також має добру продуктивність для лабораторних експериментів і дозволяє створювати адаптоване середовище для наукових досліджень. Крім того, можливість керування за допомогою різних програм робить її більш гнучкою. Недоліки включають той факт, що RedPitaya не є типовим лабораторним обладнанням, що може вплинути на його використання в дослідницьких експериментах.

Таким чином, отримані експериментальні результати керування світлодіодом довели роботу спроможність запропонованої методики розробленого веб-застосунку, а сама методика може бути в подальшому застосована для створення веб-застосунків для керування більш складними електронними приладами та пристроями.

Список використаної літератури

1. Red Pitaya documentation. URL: <https://redpitaya.readthedocs.io/en/latest/>
2. Education in conditions of war. URL: <https://pgasa.dp.ua/wp-content/uploads/2022/09/Zbirnyk-materialiv-II-Forumu-akademichnoyi-spilnoty-OSVITA-V-UMOVAH-VIJNY.pdf>
3. GOLC. URL: https://online-engineering.org/GOLC_about.php
4. Institutional Factors Governing the Deployment of Remote Experiments. URL: https://www.cyberphysicals.de/artlabWWW/public_html/publications/artec-07-mueller-REV-66.pdf
5. ILAB. URL: <https://icampus.mit.edu/projects/ilabs/>
6. About Marvel Program. URL: <https://www.marvelprogram.com/#about>
7. Pasteur education and research training laboratory (PEARL). URL: <https://pearllabsgnt.com/pearl-research-projects.php#>
8. ArPi Lab: <https://folk.ntnu.no/skoge/prost/proceedings/ifac2014/media/files/0963.pdf>
9. Віддалена лабораторія GOLDi: https://wiki.kubg.edu.ua/Віддалена_лабораторія_GOLDi
10. Red Pitaya. URL: <https://redpitaya.com/documentation/>
11. Red Pitaya User Manual. URL: <https://docs.rs-online.com/130e/0900766b8144a695.pdf>
12. PetaLinux. URL: [https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18842250/Peta Linux](https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18842250/Peta+Linux)
13. 13.System overview Web Application <https://redpitaya.readthedocs.io/en/latest/developerGuide/software/build/webapp/sysOver.html>
14. Red Pitaya FPGA programming. URL: https://halverscience.net/Electronics/FPGA/Vivado_Red_Pitaya/Vivado_Red_Pitaya.html
15. Open Source RedPitaya. URL: <https://github.com/RedPitaya>
16. Open Source Web Tutorial. URL: <https://github.com/RedPitaya/RedPitaya/tree/master/Examples/web-tutorial>
17. Red Pitaya Tutorial - SSH, Command Line Tools, Web Apps. URL: <https://www.youtube.com/watch?v=wwxDkW9CRZ0>
18. Creating first URL: <https://redpitaya.readthedocs.io/en/latest/developerGuide/software/build/webapp/firstApp.html>
19. Arm-Linux-Gnueabihf https://releases.linaro.org/14.11/components/toolchain/binaries/arm-linux-gnueabi/gcc-linaro-4.9-2014.11-x86_64_arm-linux-gnueabi.tar.gz
20. Casario M. HTML5 solutions: Essential techniques for HTML5 developers. [New York, NY] : Friends of ED, 2011. 347 p.
21. Nixon R. CSS & CSS3: 20 Lessons to Successful Web Development. McGraw-Hill Education, 2015. 304 p.
22. Crockford D. JavaScript: The Good Parts: Working with the Shallow Grain of JavaScript. O'Reilly Media, Inc., 2008. 250 p.
23. Meyers S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. O'Reilly Media, 2014. 334 p.
24. Smith B. Beginning JSON. Apress, 2015. 350 p.

DEVELOPMENT OF REMOTELY CONTROLLED LABORATORY WORK BASED ON REDPITAYA MULTIFUNCTIONAL DEVICE

This paper examines the main functional capabilities of the RedPitaya platform and the characteristics of its work, which allows us to fully assess the feasibility of using it as a basis for a laboratory stand for conducting remote laboratory work. The purpose of this work is to create a web-based application for remote control of a semiconductor LED based on RedPitaya with the PetaLinux operating system (OS), which contains the NGINX web server package in its backend.

The advantages of the RedPitaya platform include its compactness, stability, cost-effectiveness and customizability. It also has good productivity for laboratory experiments and allows the creation of an adapted environment for scientific research. In addition, the ability to control it with the help of different programs makes it more flexible. Disadvantages include the fact that RedPitaya does not

have standard laboratory equipment, which may affect its use in research experiments.

The process of creating a web based application for carrying out this remote laboratory work consisted of the development of two functional parts: an interface, which has been based on HTML, CSS, JavaScript technologies to provide a convenient and understandable interface and a software controller created in the C++ language for efficient processing of data on the server. The obtained results of the test web based application, which controls the LED operation mode on the RedPitaya control board, can be used to implement remote control larger electronic circuits and devices.

Thus, the obtained experimental results of LED control confirmed the feasibility of the proposed method of developing a web-device, and the method itself can be further used to create web-devices for control of more portable electronic devices and gadgets.

References

- [1] Red Pitaya documentation. URL: <https://redpitaya.readthedocs.io/en/latest/>
- [2] Education in conditions of war. URL: <https://pgasa.dp.ua/wp-content/uploads/2022/09/Zbirnyk-materialiv-II-Forumu-akademichnoyi-spilnoty-OSVITA-V-UMOVAH-VIJNY.pdf>
- [3] GOLC. URL: https://online-engineering.org/GOLC_about.php
- [4] Institutional Factors Governing the Deployment of Remote Experiments. URL: https://www.cyberphysicals.de/artlabWWW/public_html/publications/artec-07-mueller-REV-66.pdf
- [5] ILAB. URL: <https://icampus.mit.edu/projects/ilabs/>
- [6] About Marvel Program. URL: <https://www.marvelprogram.com/#about>
- [7] Pasteur education and research training laboratory (PEARL). URL: <https://pearllabsgnt.com/pearl-research-projects.php#>
- [8] ArPi Lab: <https://folk.ntnu.no/skoge/prost/proceedings/ifac2014/media/files/0963.pdf>
- [9] Віддалена лабораторія GOLDi: https://wiki.kubg.edu.ua/Віддалена_лабораторія_GOLDi
- [10] Red Pitaya. URL: <https://redpitaya.com/documentation/>
- [11] Red Pitaya User Manual. URL: <https://docs.rs-online.com/130e/0900766b8144a695.pdf>
- [12] PetaLinux. URL: <https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18842250/PetaLinux>
- [13] System overview Web Application <https://redpitaya.readthedocs.io/en/latest/developerGuide/software/build/webapp/sysOver.html>
- [14] Red Pitaya FPGA programming. URL: https://halverscience.net/Electronics/FPGA/Vivado_Red_Pitaya/Vivado_Red_Pitaya.html
- [15] Open Source RedPitaya. URL: <https://github.com/RedPitaya>
- [16] Open Source Web Tutorial. URL: <https://github.com/RedPitaya/RedPitaya/tree/master/Examples/web-tutorial>
- [17] Red Pitaya Tutorial - SSH, Command Line Tools, Web Apps. URL: <https://www.youtube.com/watch?v=wwxDkW9CRZ0>
- [18] Creating first URL: <https://redpitaya.readthedocs.io/en/latest/developerGuide/software/build/webapp/firstApp.html>
- [19] Arm-Linux-Gnueabihf https://releases.linaro.org/14.11/components/toolchain/binaries/arm-linux-gnueabi/gcc-linaro-4.9-2014.11-x86_64_arm-linux-gnueabi.tar.xz
- [20] Casario M. HTML5 solutions: Essential techniques for HTML5 developers. [New York, NY] : Friends of ED, 2011. 347 p.
- [21] Nixon R. CSS & CSS3: 20 Lessons to Successful Web Development. McGraw-Hill Education, 2015. 304 p.
- [22] Crockford D. JavaScript: The Good Parts: Working with the Shallow Grain of JavaScript. O'Reilly Media, Inc., 2008. 250 p.
- [23] Meyers S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. O'Reilly Media, 2014. 334 p.
- [24] Smith B. Beginning JSON. Apress, 2015. 350 p.

Надійшла до редколегії 14.06.2024