

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Факультет комп'ютерних наук та технологій  
Кафедра комп'ютерних систем та мереж

## Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти)

на тему FGPA СИСТЕМА ВІДДАЛЕНОГО МОНІТОРИНГУ

Виконав: студент 2 курсу, групи КНТ-612м  
спеціальності \_\_\_\_\_

123 Комп'ютерна інженерія

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Спеціалізовані комп'ютерні системи

ІЛ'ЯНЦЕВ В. В.

(ПРИЗВИЩЕ та ініціали)

Керівник ГРУШКО С.С.

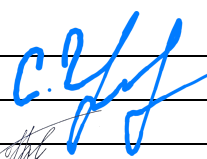
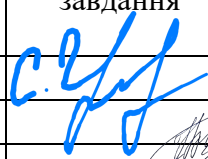


(ПРИЗВИЩЕ та ініціали)

Рецензент КАПЛІЄНКО Т.І.

(ПРИЗВИЩЕ та ініціали)




## 6. Консультанти розділів проєкту (роботи)

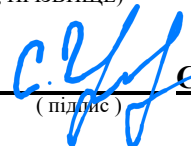
Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	ГРУШКО С. С., доцент		
нормоконтроль	ЩЕРБАК Н.В., ст. викл.		

7. Дата видачі завдання 01.10.2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз технології блокчейну, огляд апаратних платформ і алгоритмів хешування функцій	10.10.2023 р.	
2	Розробка архітектури системи	15.10.2023 р.	
3	Вибір апаратного забезпечення	20.10.2023 р.	
4	Вибір програмного забезпечення	05.10.2023 р.	
5	Розробка системи	20.11.2023 р.	
6	Оформлення отриманих результатів у ПЗ	25.11.2023 р.	
7	Оформлення графічного матеріалу	01.12.2023 р.	
8	Оформлення допоміжного матеріалу	10.12.2023 р.	

Студент  Владислав ЛІЯНЦЕВ  
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник проєкту (роботи)  Світлана ГРУШКО  
(підпис) (ім'я, ПРІЗВИЩЕ)

## РЕФЕРАТ

ПЗ: 70 сторінок, 20 рисунків, 9 таблиці, 24 джерела

### FPGA, IOT, АПАРАТНА ПЛАТФОРМА, ВІДДАЛЕНИЙ МОНІТОРИНГ

Об'єкт розробки – системи віддаленого моніторингу.

Предмет розробки - використання FPGA в системах віддаленого моніторингу.

Мета дослідження - розробка та аналіз ефективних та надійних рішень для систем віддаленого моніторингу з використанням FPGA, задля оптимізації обробки даних та забезпечення надійності та ефективності цих систем.

В роботі розкривається тема використання в системах віддаленого моніторингу. Зосереджено увагу на тому, як технології неперервно розвиваються, і як важливою частиною сучасного життя та промисловості стає віддалений моніторинг, що включає спостереження за об'єктами, процесами, середовищем, а також збір та аналіз даних на відстані. Основна увага звертається на важливість забезпечення надійності та ефективності цих систем, особливо при моніторингу в реальному часі та обробці великих обсягів даних. Документ висвітлює актуальність теми у контексті зростаючих вимог до віддаленого моніторингу в різних сферах, таких як безпека, сільське господарство, медицина, і наголошує на значенні дослідження використання FPGA для створення надійних, продуктивних систем моніторингу, які відрізняються низькою латентністю та здатністю реагувати на події в реальному часі. Ціль роботи полягає у вирішенні актуальних проблем та наданні практичних рекомендацій щодо використання FPGA в системах віддаленого моніторингу.

## ABSTRACT

EN: 70 pages, 20 figure, 9 tables, 24 sources

FPGA, IOT, HARDWARE PLATFORM, REMOTE MONITORING

Object of development - remote monitoring systems.

Subject of development - the use of FPGA in remote monitoring systems.

Research objective - to develop and analyze effective and reliable solutions for remote monitoring systems using FPGA, in order to optimize data processing and ensure the reliability and efficiency of these systems.

In this work, the topic of using remote monitoring systems is explored. Attention is focused on how technologies are continuously developing, and how remote monitoring becomes an important part of modern life and industry. This includes observation of objects, processes, and the environment, as well as data collection and analysis from a distance. Main emphasis is placed on the importance of ensuring the reliability and efficiency of these systems, especially in real-time monitoring and processing of large data volumes. The document highlights the relevance of the topic in the context of growing demands for remote monitoring in various fields, such as security, agriculture, medicine, and underscores the significance of researching the use of FPGA to create reliable, productive monitoring systems characterized by low latency and the ability to respond to events in real time. The aim of the work is to address current problems and provide practical recommendations for the use of FPGA in remote monitoring systems.

## ЗМІСТ

Вступ.....	8
1 Огляд літератури.....	10
1.1 Інтернет речей (IoT) та його роль у віддаленому моніторингу.....	10
1.1.1 Сутність інтернету речей.....	10
1.1.2 Основні компоненти та архітектура IoT систем .....	11
1.1.3 Застосування інтернету речей у віддаленому моніторингу.....	12
1.1.4 Переваги та можливі обмеження використання IoT у віддаленому моніторингу.....	14
1.1.5 Спільні технології та стандарти для розвитку IoT-рішень у сфері моніторингу.....	15
1.2 FPGA технології та їх застосування в IoT системах.....	17
1.2.1 Основні поняття та принципи роботи FPGA.....	17
1.2.2 Застосування FPGA в IoT системах .....	19
1.2.3 Приклади реалізації IoT рішень на базі FPGA.....	20
1.3 Застосування FPGA в системах віддаленого моніторингу .....	22
1.4 Висновки до розділу 1 .....	24
2 Вибір апаратного і програмного забезпечення .....	25
2.1 Вибір FPGA платформи.....	25
2.1.1 Xilinx Zynq-7000 .....	25
2.1.2 PYNQ-Z2 .....	26
2.1.3 Basys 3 Artix-7 FPGA Board .....	28
2.1.4 Arty A7.....	29

2.1.6 Zybo Z7 .....	31
2.2 Вибір модуля вимірювання освітленості.....	33
2.2.1 SparkFun Ambient Light Sensor Breakout з TEMT6000 .....	33
2.2.2 Adafruit TSL2591 High Dynamic Range Digital Light Sensor .....	34
2.2.3 Grove - Light Sensor v1.2 .....	35
2.2.4 Seeed Studio Luminance Sensor .....	35
2.2.5 Digilent Pmod ALS .....	36
2.3 Вибір програмного забезпечення .....	38
2.3.1 Xilinx ISE.....	38
2.3.2 AMD Vivado Design Suite .....	39
2.3.3 Xilinx Vitis Unified Software Platform .....	41
2.3.4 IFTTT Hue service .....	42
2.3.5 AMD PYNQ Framework .....	46
2.4 Висновки до розділу 2 .....	51
3 Розробка системи .....	52
3.1 Опис процесу розробки системи .....	52
3.2 Висновки до розділу 3 .....	65
Висновки .....	67
Перелік джерел посилання .....	68

## ВСТУП

У світі, де технології невпинно розвиваються, віддалений моніторинг став важливою складовою сучасного життя та промисловості. Віддалений моніторинг включає в себе спостереження за станом об'єктів, процесів, середовища, а також збір і аналіз даних на відстані. Однак однією з ключових проблем є забезпечення надійності та ефективності цих систем моніторингу, особливо коли потрібно здійснювати моніторинг в режимі реального часу та оброблювати великі обсяги даних. Віддалений моніторинг вимагає потужних рішень, здатних виконувати завдання навіть у важких умовах.

Актуальність теми полягає в тому, що сучасний світ стикається з рядом викликів, які передбачають віддалений моніторинг як необхідний інструмент для їх вирішення. Наприклад, сфера безпеки потребує систем віддаленого відеоспостереження та виявлення незвичайних подій. Сільське господарство потребує моніторингу урожаїв та стану ґрунту. Медицина вимагає систем телемедицини для віддаленого нагляду за станом хворих. Усі ці сфери потребують надійних та продуктивних рішень для віддаленого моніторингу. При цьому потрібні інструменти, які забезпечують якість та ефективність моніторингу, а також забезпечують безпеку та конфіденційність даних.

Дослідження у галузі використання FPGA в системах віддаленого моніторингу має велику практичну цінність. Впровадження цих технологій дозволяє створювати надійні та продуктивні системи моніторингу, які забезпечують надійність, низьку латентність та можливість реагувати в реальному часі на зміни та надзвичайні події. Використання FPGA дозволяє оптимізувати обробку даних та ефективно використовувати ресурси, що робить їх ідеальним інструментом для віддаленого моніторингу в різних галузях, від промисловості та сільського господарства до медицини та екології. Таким чином, дана дипломна

робота спрямована на вирішення актуальних проблем та надання практичних рекомендацій щодо використання FPGA в системах віддаленого моніторингу.

## 1 ОГЛЯД ЛІТЕРАТУРИ

### 1.1 Інтернет речей (IoT) та його роль у віддаленому моніторингу

#### 1.1.1 Сутність інтернету речей

Інтернет речей (IoT) - це технологічна концепція, яка визначає мережу підключених фізичних об'єктів, пристроїв та систем до Інтернету. Ідея полягає в тому, щоб надати здатність об'єктам спілкуватися, обмінюватися даними і виконувати дії без прямої людської участі. Основна мета IoT - забезпечити спрощення життя і роботи людей, зробивши навколишнє середовище більш інтелектуальним та зв'язаним.

Сутність IoT полягає в наступному:

- усі об'єкти, які входять в IoT, здатні підключатися до Інтернету. Це може включати в себе різноманітні пристрої, від домашніх приладів і транспортних засобів до датчиків та виробничого обладнання;
- об'єкти IoT зазвичай оснащені датчиками, які збирають різноманітні дані, такі як температура, вологість, рух і т. д. Ці дані передаються через мережу для подальшої обробки та аналізу;
- однією з ключових переваг IoT є можливість взаємодії об'єктів між собою та виконання автоматизованих дій на основі зібраних даних. Наприклад, "розумний" будинок може автоматично регулювати температуру в приміщенні залежно від погодних умов та присутності людей;
- IoT надає змогу збирати та отримувати дані в режимі реального часу, що важливо для віддаленого моніторингу та прийняття рішень на основі актуальної інформації;
- дані в IoT можуть бути оброблені за допомогою аналітики та машинного навчання, що дозволяє виділяти важливі зв'язки та вдосконалювати процеси на основі отриманих даних.

### 1.1.2 Основні компоненти та архітектура IoT систем

Інтернет речей (IoT) складається з складної архітектури [1], яка об'єднує різноманітні компоненти для забезпечення функціональності і ефективності системи. Основні компоненти та архітектура IoT систем включають:

- датчики та пристрої збору даних;
- мережеві пристрої;
- хмарні сервіси;
- аналітику та обробку даних;
- застосунки та інтерфейси користувача;
- мережу зв'язку.

Датчики та пристрої збору даних. Це перші компоненти системи, які відповідають за збір різних видів даних, такі як температура, вологість, рух, освітленість, тиск тощо. Датчики можуть бути вбудовані в об'єкти або підключатися до них.

Мережеві пристрої. Ці компоненти відповідають за забезпечення підключення датчиків та пристроїв до мережі. Це можуть бути мікроконтролери, модулі зв'язку, маршрутизатори, інтернет-шлюзи тощо.

Хмарні сервіси. Дані, зібрані датчиками, передаються до хмарних серверів, де вони обробляються та зберігаються. Це дає можливість доступу до даних з будь-якої точки світу та забезпечує їх безпеку.

Аналітика та обробка даних. Дані зібрані з датчиків піддаються обробці та аналізу для виявлення зв'язків, шаблонів та важливих подій. Це важливо для винесення важливої інформації та подальшого прийняття рішень.

Застосунки та інтерфейси користувача. Додатки та інтерфейси дозволяють користувачам взаємодіяти з IoT системою. Це може включати мобільні додатки, веб-інтерфейси та інші засоби для відображення даних та управління системою.

Мережа зв'язку. Ця компонента забезпечує зв'язок між різними складовими системи та передачу даних. Це може бути бездротовими мережами, такими як Wi-Fi, Bluetooth, Zigbee, або дротовими мережами.

Архітектура IoT системи базується на принципі збору даних, передачі їх до хмарних сервісів для обробки, аналізу та зберігання, а також надання доступу до даних користувачам через різні інтерфейси. Ця архітектура дозволяє створити ефективну та потужну систему моніторингу та управління на основі даних, зібраних з IoT-пристроїв і датчиків.

### 1.1.3 Застосування Інтернету речей у віддаленому моніторингу

Інтернет речей став надзвичайно важливою технологічною концепцією в сфері віддаленого моніторингу. Віддалений моніторинг передбачає можливість стеження за станом об'єктів, які знаходяться на відстані від спостерігача, і передачу даних з цих об'єктів для аналізу та прийняття рішень [2].

IoT датчики можуть вимірювати різні параметри, такі як температура, вологість, тиск, рух, вібрація, рівень рідини та багато інших. Ці дані можуть бути важливі для стеження за станом об'єктів, наприклад, у виробництві, логістиці, агропромисловості та медицині.

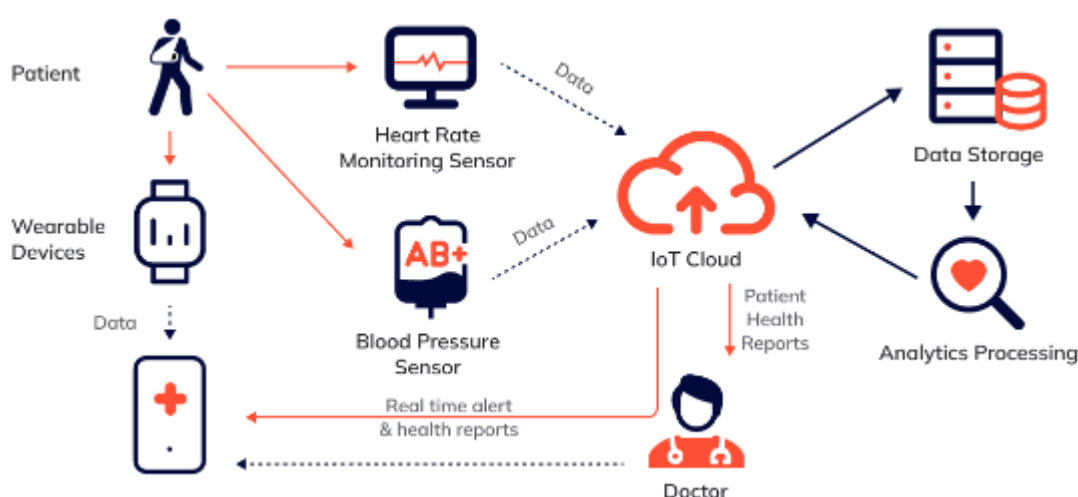


Рисунок 1.1 - IoT в медицині

IoT системи можуть бути встановлені на віддалених об'єктах, таких як сонячні електростанції, вітрові турбіни, нафтові свердловини або транспортні засоби. Це дозволяє операторам віддалено стежити за роботою обладнання, збирати дані про ефективність та проводити заздалегідь розкладану обслуговуючу діяльність.

Віддалений моніторинг за допомогою IoT важливий у сферах безпеки та спостереження. Відеокамери, датчики руху та системи контролю доступу можуть бути підключені до мережі IoT для віддаленого спостереження за територією, будівлями або об'єктами.

IoT дозволяє вдосконалити управління енергоспоживанням. Наприклад, "розумний" будинок може автоматично регулювати освітлення та опалення в залежності від наявності людей та погодних умов, що допомагає зменшити споживання електроенергії.

IoT може використовуватися для моніторингу навколишнього середовища, включаючи контроль якості повітря, води та рівнів забруднення. Це важливо для охорони довкілля та забезпечення здоров'я населення.

В сільському господарстві IoT використовується для моніторингу росту рослин, стану ґрунту та визначення потреби в поливі. Це сприяє підвищенню врожаю та оптимізації виробництва [3].

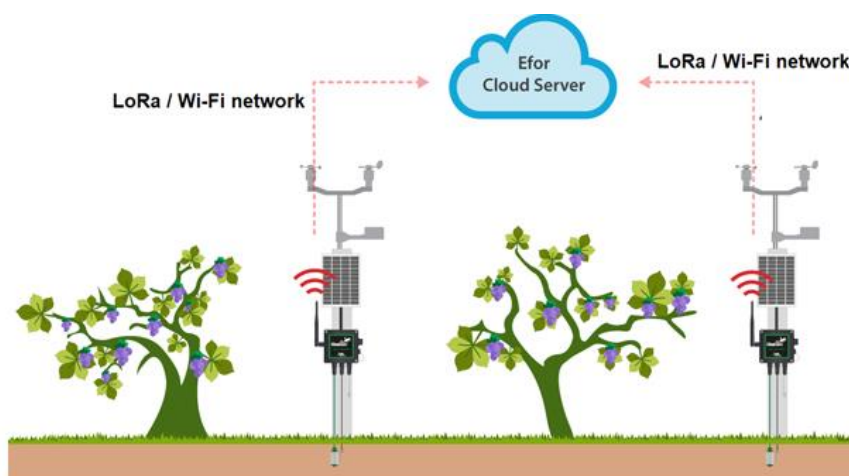


Рисунок 1.2 - IoT в сільському господарстві

Застосування Інтернету речей в віддаленому моніторингу має великий потенціал для покращення ефективності та безпеки в різних галузях. Вона дозволяє отримувати важливу інформацію в реальному часі, що сприяє прийняттю інформованих рішень та покращенню контролю за об'єктами та процесами.

#### **1.1.4 Переваги та можливі обмеження використання IoT у віддаленому моніторингу**

Використання Інтернету речей у віддаленому моніторингу включає в себе численні переваги, але також супроводжується певними можливими обмеженнями. Розглянемо їх детальніше.

Переваги використання IoT у віддаленому моніторингу:

- віддалений доступ до даних. IoT дозволяє віддаленим операторам отримувати доступ до даних у реальному часі з будь-якого місця, що робить можливим віддалений моніторинг та прийняття рішень;
- покращена ефективність та автоматизація. Інтернет речей може оптимізувати процеси та забезпечити автоматизацію дій на основі зібраних даних, що призводить до покращення продуктивності та зниження витрат;
- економія часу та ресурсів. Можливість реагувати на події в реальному часі дозволяє ефективно використовувати ресурси та зменшувати час, необхідний для вирішення проблем;
- зниження витрат на обслуговування. Віддалені моніторингові системи дозволяють передбачати несправності та проводити профілактичне обслуговування, що зменшує витрати на ремонт та заміну обладнання;
- можливість інтеграції з існуючими системами. Інтернет речей може бути легко інтегрований з існуючими системами та інфраструктурою, що спрощує впровадження та масштабування.

Можливі обмеження використання IoT у віддаленому моніторингу:

- забезпечення безпеки даних. IoT системи можуть бути схильні до кібератак та порушень конфіденційності даних, тому необхідно ретельно забезпечувати безпеку;

- залежність від доступу до Інтернету. IoT системи вимагають доступу до Інтернету, тому відсутність зв'язку може призвести до втрати здатності віддаленого моніторингу;
- вартість розгортання і підтримки. Розгортання та підтримка IoT системи може бути витратною, включаючи витрати на обладнання, програмне забезпечення та підтримку;
- стандартизація та сумісність. Різні виробники пропонують різні рішення IoT, що може призвести до проблем із сумісністю та інтеграцією;
- приватність і етика. Збір та використання даних віддаленого моніторингу може порушувати питання приватності та етики, тому необхідно дотримуватися відповідних норм та правил.

Загалом, використання Інтернету речей у віддаленому моніторингу може значно покращити управління процесами та забезпечити доступ до важливої інформації у реальному часі, але вимагає ретельного планування, безпеки та управління ризиками.

### **1.1.5 Спільні технології та стандарти для розвитку IoT-рішень у сфері моніторингу**

У сфері IoT моніторингу і віддаленого нагляду існують спільні технології та стандарти, які сприяють розвитку інтероперабельних та ефективних рішень. Ці стандарти та технології грають ключову роль у створенні сумісних систем та забезпеченні їх безпеки та надійності.

Протокол MQTT (Message Queuing Telemetry Transport) є легковаговим та ефективним протоколом для обміну повідомленнями між IoT пристроями та хмарними серверами. Він особливо корисний для віддаленого моніторингу, оскільки дозволяє передавати дані в реальному часі та мінімізує обсяг трафіку даних. MQTT відзначається мінімальним споживанням бандширини та можливістю роботи в умовах обмежених ресурсів. Він забезпечує гнучку підтримку QoS (Quality of Service), що дозволяє вибирати рівень доставки повідомлень - від "по одному разу" до "найменшої гарантованої доставки".

Протокол CoAP (Constrained Application Protocol) розроблено спеціально для обміну даними в обмежених обчислювальних ресурсах IoT пристроїв. Він дозволяє створювати легковагові та високоефективні IoT рішення. Цей протокол був розроблений для роботи в умовах обмежених ресурсів, таких як низька пропускна здатність, обмежені обчислювальні можливості і низька споживана енергія.

Технологія LoRaWAN (Long Range Wide Area Network) використовується для створення великих мереж IoT пристроїв з великим діапазоном покриття. Однією з ключових переваг технології LoRaWAN є її здатність передавати дані на дуже великі відстані. Зв'язок може бути встановлений на відстані до кількох кілометрів від точки доступу, що робить цю технологію ідеальною для використання в сільському господарстві, містах, промислових комплексах та інших великих територіях.

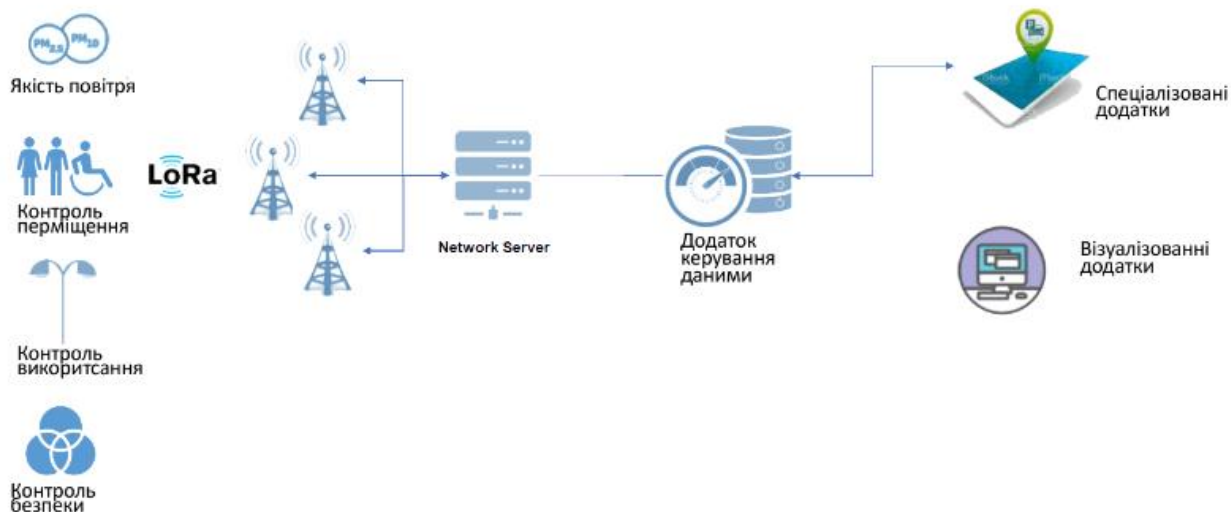


Рисунок 1.3 – Використання технології LoRaWAN

Технологія NB-IoT (Narrowband IoT) - це стандарт для підключення пристроїв IoT до мобільних мереж. Вона забезпечує низьку споживану потужність, довгий діапазон дії та надійну зв'язок, що робить її ідеальною для віддаленого моніторингу. Вона спроектована для підключення мільйонів низькопотужних

пристроїв, які вимагають довгого терміну служби батареї та працюють в обмеженому діапазоні пропускнуї здатності.

Забезпечення безпеки даних важливо в IoT моніторингу. Протоколи шифрування, такі як TLS (Transport Layer Security) та HTTPS, забезпечують захист передачі та зберігання даних.

JSON (JavaScript Object Notation) та XML (eXtensible Markup Language) є поширеними форматами для обміну даними між IoT пристроями та системами. Вони дозволяють легко інтегрувати та обробляти дані.

Спільні технології та стандарти є фундаментом для розвитку різноманітних IoT-рішень у сфері моніторингу. Вони спрощують розробку, інтеграцію та забезпечують безпеку цих систем, роблячи їх більш доступними та надійними для віддаленого нагляду та управління.

## **1.2 FPGA технології та їх застосування в IoT системах**

### **1.2.1 Основні поняття та принципи роботи FPGA**

Для розуміння та застосування FPGA (Field-Programmable Gate Array) в контексті IoT систем, розглянемо основні поняття та принципи роботи цієї технології.

Field-Programmable Gate Array, або FPGA, є спеціалізованим обчислювальним пристроєм, який відрізняється від звичайних мікроконтролерів і мікросхем своєю апаратною гнучкістю. Основними компонентами FPGA є програмована логіка, логічні елементи (гейти), блоки пам'яті та вхідно-вихідні порти. Принцип роботи полягає в можливості перепрограмування FPGA для виконання специфічних обчислювальних завдань. Програмування полягає у визначенні логіки та з'єднань між гейтами, створюючи спеціалізовані логічні схеми, які реалізують певну функціональність. FPGA використовується для

реалізації широкого спектру завдань, від обробки сигналів до симуляції та прискорення обчислень, та може бути перепрограмовано без зміни апаратної складової, що робить його ідеальним для сценаріїв, де потрібна адаптація до різних завдань та вимог.

Основний принцип роботи FPGA полягає в можливості реалізації складних логічних функцій та обчислювальних операцій шляхом програмування логічних елементів та їх з'єднань. Це дає можливість створювати спеціалізовані обчислювальні схеми, які можуть прискорювати різні завдання, включаючи обробку сигналів, криптографічні операції, машинне навчання та багато інших. Важливим аспектом FPGA є їх здатність до перепрограмування, що дозволяє використовувати одну апаратну платформу для різних завдань без необхідності зміни апаратних компонентів. Ця гнучкість і потужність робить FPGA важливою технологією для великої кількості застосувань в сучасних обчислювальних системах.

Принципи роботи FPGA базуються на використанні LUT (Look-Up Table), які зберігають комбінації вхідних сигналів і визначають вихідний сигнал відповідно до програми. Також, FPGA мають блоки пам'яті, які дозволяють зберігати дані та конфігураційну інформацію для програмування. Ці блоки пам'яті дозволяють FPGA зберігати стан і дані між операціями.

Додатково, FPGA мають велику швидкість реакції, оскільки обчислення відбуваються паралельно, що дозволяє виконувати операції в режимі реального часу. Вони також володіють великими можливостями для обробки сигналів і великими обсягами даних завдяки паралельному обчисленню та великій кількості логічних елементів.

Узагальнюючи, FPGA - це потужні обчислювальні пристрої з великим потенціалом для виконання різноманітних завдань завдяки їх гнучкості, високій швидкості та можливостям паралельної обробки даних. Вони знаходять застосування в широкому спектрі галузей, від вбудованих систем до наукових досліджень і обчислень загального призначення.

### 1.2.2 Застосування FPGA в IoT системах

Використання FPGA у сфері Інтернету речей відкриває безліч можливостей для розробки потужних та гнучких систем моніторингу та керування. Розглянемо деякі з основних застосувань FPGA в області IoT:

- FPGA відзначаються високою швидкістю обробки даних та здатністю працювати в реальному часі. Вони ідеально підходять для задач обробки сигналів, таких як обробка відеопотоків, аналіз сигналів з датчиків або обчислення великих обсягів даних;

- такі мікросхеми мають велику кількість логічних блоків, що дозволяє виконувати паралельні обчислення. Це особливо корисно для задач, які вимагають одночасного обчислення багатьох завдань, таких як обробка сигналів, машинне навчання або обчислення штучного інтелекту;

- програмована логіка може використовуватися для реалізації різних протоколів зв'язку, таких як Ethernet, Wi-Fi, Bluetooth або спеціалізовані мережеві протоколи. Це дозволяє створювати пристрої, які легко інтегруються в мережі IoT;

- FPGA може бути використаний для обробки відеопотоків та зображень в реальному часі. Це корисно для систем відеоспостереження, розпізнавання облич, аналізу якості зображень та інших застосувань;

- мікросхеми програмованої логіки можуть використовуватися для реалізації апаратних засобів шифрування та обробки даних. Це допомагає забезпечити високий рівень безпеки та конфіденційності в мережах IoT.

Однією з найбільших переваг FPGA є їх програмована природа. Вони можуть бути перепрограмовані для виконання різних завдань в залежності від потреб системи IoT.

Використання FPGA в системах Інтернету речей має численні переваги, але також супроводжується деякими обмеженнями.

Переваги використання FPGA в IoT системах:

- FPGA можуть виконувати паралельні обчислення та обробку в реальному часі, що дозволяє розробляти швидкі та потужні IoT системи;

- мікросхеми FPGA можуть бути перепрограмовані для виконання різних завдань, що дозволяє їм адаптуватися до змінних потреб систем IoT;
- низька латентність - FPGA здатні працювати з мінімальною затримкою, що особливо важливо для задач реального часу;
- FPGA можуть бути використані для реалізації апаратних засобів шифрування та захисту даних, забезпечуючи високий рівень безпеки в системах IoT;
- такі мікросхеми можуть бути легко інтегровані з іншими технологіями, такими як машинне навчання, штучний інтелект та облік даних, що розширює можливості систем IoT.

Обмеження використання FPGA в IoT системах:

- FPGA вимагають глибоких знань апаратного програмування, що може бути складним для розробників з досвідом в програмуванні на рівні високого рівня;
- розробка систем на базі FPGA може бути витратною, включаючи витрати на обладнання та розробку програмного забезпечення;
- FPGA мають обмежений обсяг ресурсів, що може стати обмеженням для деяких завдань;
- щоб досягти найкращої продуктивності, розробники повинні мати глибоке розуміння потреб системи та споживачів, що може вимагати додаткових зусиль;
- хоча FPGA дозволяють перепрограмувати, це може бути трудомістким і вимагати внесення змін в пристрій вручну.

Враховуючи ці переваги та обмеження, використання FPGA в системах IoT варто розглядати з урахуванням конкретних вимог та завдань проекту. Вони можуть стати потужним інструментом для досягнення високої продуктивності та безпеки в сфері моніторингу та керування IoT систем.

### **1.2.3 Приклади реалізації IoT рішень на базі FPGA**

Розглянемо деякі конкретні приклади успішної реалізації IoT рішень на базі FPGA.

Системи відеоспостереження. Фіксовані логічні матриці в програмованих логічних пристроях (ПЛІС) FPGA дозволяють створити потужну обчислювальну систему для обробки відеопотоків у реальному часі. FPGA надає можливість виконувати обчислення паралельно, що ідеально підходить для обробки великого обсягу відеоданих без значних затримок.

При обробці відеопотоків у реальному часі, FPGA може виконувати операції, такі як розпізнавання обличчя, відстеження руху, визначення об'єктів, фільтрацію та аналіз відеоданих, а також компресію та передачу великих обсягів відео без затримок. Ця спроможність робить FPGA ідеальним вибором для застосувань, де важлива висока швидкість та точність обробки відеоданих, такі як системи безпеки, медичинські пристрої, робототехніка та віртуальна реальність. FPGA дозволяє реалізувати власні алгоритми обробки відео та швидко адаптувати їх до конкретних завдань без зміни апаратної складової системи.

Медичні системи моніторингу. FPGA можуть бути використані для обробки біометричних даних, які збираються в реальному часі. Наприклад, вони можуть виконувати аналіз сигналів ЕКГ, контролювати рівень глюкози в крові або вимірювати пульс пацієнтів у медичних IoT пристроях.

Промислові системи моніторингу та керування. У сфері промисловості, FPGA знайшли широке застосування у системах моніторингу та керування завдяки своїй потужній обчислювальній спроможності та низькій латентності. Вони використовуються для вимірювання та аналізу різних параметрів промислових процесів, таких як температура, тиск, вологість, рівень, швидкість, та інші.

FPGA дозволяють збирати та обробляти дані в режимі реального часу, що дозволяє операторам миттєво реагувати на події та управляти промисловими процесами. Вони також використовуються для автоматизації та оптимізації промислових систем, що дозволяє зменшити витрати енергії та матеріалів, збільшити продуктивність та забезпечити безпеку на робочих місцях.

Промислові системи моніторингу та керування, які використовують FPGA, можуть бути використані в різних сферах, включаючи автомобільну

промисловість, виробництво, нафтогазову промисловість, енергетику, інфраструктуру та багато інших. FPGA надають можливість створювати спеціалізовані рішення для конкретних завдань та оптимізувати промислові процеси з використанням розроблених на замовлення алгоритмів та обчислень. Системи машинного навчання та штучного інтелекту: FPGA можуть бути використані для реалізації апаратно-програмних платформ для задач машинного навчання. Вони забезпечують високу продуктивність та ефективність для реалізації складних алгоритмів штучного інтелекту в системах IoT.

Системи керування енергоефективністю. FPGA можуть використовуватися для реалізації систем керування енергоефективністю в будівлях та промислових об'єктах. Вони дозволяють оптимізувати використання енергії, вмикаючи та вимикаючи електронні пристрої згідно з реальними потребами.

Системи водопостачання та очищення води. FPGA можуть бути використані для моніторингу та управління системами водопостачання та очищення води. Вони дозволяють оптимізувати роботу насосів, вимірювати рівень забрудненості води та вчасно виявляти витіки.

Ці приклади ілюструють широкий спектр застосування FPGA в реалізації ефективних та потужних систем IoT. Використання FPGA дозволяє створювати високопродуктивні та надійні IoT рішення, які можуть впроваджуватися в різних сферах від індустрії та медицини до енергозберігаючих технологій та водопостачання.

### **1.3 Застосування FPGA в системах віддаленого моніторингу**

Розглянемо таблицю (табл. 1.1) характеристик сучасних мікросхем FPGA. Кожен рядок таблиці представляє різні мікросхеми FPGA від різних виробників, такі як Xilinx, Intel (Altera), Lattice, Microchip та QuickLogic.

Таблиця 1.1 – Характеристики деяких FPGA

<b>Мікросхема FPGA</b>	<b>Логічні елементи</b>	<b>Блоки пам'яті</b>	<b>Вбудована ОЗП</b>	<b>Максимальна швидкість</b>	<b>Інтерфейси</b>
Xilinx Virtex UltraScale+	До 2,586,720	До 194,400 МБіт	До 173,500 МБіт	Понад 50 Гбіт/сек	PCIe, Ethernet, USB, інші
Intel (Altera) Stratix 10	До 2,072,160	До 134,070 МБіт	До 59,952 МБіт	Понад 1 Тбіт/сек	PCIe, Ethernet, USB, інші
Lattice ECP5	До 84,640	До 9,216 МБіт	До 21,816 МБіт	Понад 3 Гбіт/сек	PCIe, Ethernet, USB, інші
Microchip PolarFire	До 504,000	До 22,400 МБіт	До 60,000 МБіт	Понад 12 Гбіт/сек	Ethernet, CAN, SPI, інші
QuickLogic EOS S3	До 16,000	До 5,600 МБіт	До 3,840 МБіт	Понад 10 Гбіт/сек	I2C, SPI, UART, інші

Логічні елементи - це кількість логічних елементів, які доступні в мікросхемі FPGA для створення логічних функцій і обчислень. Це важливо для визначення складності завдань, які можна виконати на цій FPGA.

Блоки пам'яті - це обсяг вбудованої пам'яті в мікросхемі FPGA, яка використовується для зберігання даних та конфігураційної інформації. Більше пам'яті може бути корисним для обробки та зберігання даних.

Вбудована ОЗП - це обсяг оперативної пам'яті, яка доступна для виконання обчислень. Більше ОЗП дозволяє виконувати більше завдань одночасно та швидше.

Максимальна швидкість - це максимальна швидкість передачі даних або обчислення, яку може підтримувати мікросхема FPGA. Вона важлива для застосувань, де важливий час виконання.

Інтерфейси - це типи вбудованих інтерфейсів, які підтримуються мікросхемою FPGA. Інтерфейси включають PCIe, Ethernet, USB та інші, і вони важливі для підключення FPGA до інших пристроїв та систем.

Зазначені дані наведені для ілюстрації загальних характеристик різних мікросхем FPGA і можуть різнитися в залежності від конкретної моделі та

конфігурації. Важливо відзначити, що ці характеристики можуть бути важливими при виборі FPGA для конкретного застосування, і розробники часто вибирають мікросхему, яка найкраще відповідає їх потребам у відповідності з обсягом обчислення, швидкістю обробки, обсягом пам'яті та підтримуваними інтерфейсами.

## 1.4 Висновки до розділу 1

Сучасний світ переживає епоху цифрової трансформації, і IoT є ключовим інструментом у цьому процесі. Його здатність збирати та обробляти дані в реальному часі дозволяє нам здійснювати ефективний контроль та приймати обґрунтовані рішення. Від сільського господарства та промисловості до медицини та екології, ця технологія змінює підхід до віддаленого моніторингу. Вона сприяє автоматизації процесів та оптимізації виробництва, що робить її незамінною в різних галузях.

Перевагами використання IoT є зниження витрат, підвищення продуктивності та можливість вчасно реагувати на події роблять IoT привабливою технологією.

Однак, було також наголошено на протиріччя з якими стикається IoT. Кібербезпека, приватність даних та стійкість до відмов залишаються актуальними питаннями, які потребують ретельного вирішення.

У цілому, IoT визнається як ключовий фактор у подальшому розвитку віддаленого моніторингу, а розуміння його ролі та можливостей допомагає суспільству визначати правильні напрямки для використання цієї технології у майбутньому.

Сучасні дослідження визнають FPGA як потужний інструмент для віддаленого моніторингу та пророчать їхню успішну інтеграцію в різні сфери

застосування, сприяючи вдосконаленню систем моніторингу та керування у реальному часі. Нові технології та рішення, які виникають на базі FPGA, відкривають широкі перспективи для покращення продуктивності та ефективності систем віддаленого моніторингу.

## **2 ВИБІР АПАРАТНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **2.1 Вибір FPGA платформи**

#### **2.1.1 Xilinx Zynq-7000**

Плати серії Xilinx Zynq-7000 є високопродуктивними системами на кристалі (SoC), які поєднують в собі високопродуктивний FPGA (поле програмованих вентилів) і двоядерний процесор ARM Cortex-A9. Це робить їх ідеальними для розробки вбудованих систем, де потрібно висока обчислювальна потужність та гнучкість програмування [11].

Основною особливістю серії Zynq-7000 є її архітектура, що інтегрує як програмовану логіку, так і процесорні ядра на одному чіпі, що забезпечує значну економію місця та енергії порівняно з традиційними рішеннями, де FPGA та процесори знаходяться на окремих чіпах. Це також поліпшує продуктивність системи, оскільки зменшує затримки, що виникають при обміні даними між процесором та FPGA.

З точки зору характеристик, серія Zynq-7000 пропонує різноманітність у логічних клітинах та DSP блоках, що дозволяє вибрати конкретну модель відповідно до потреб проекту. Ці плати також оснащені значним обсягом блокової пам'яті, високошвидкісними трансіверами та ефективними рішеннями для управління годинником, такими як PLLs та MMCM.

Серія Zynq-7000 широко використовується в промисловості, освіті та наукових дослідженнях, оскільки вона дозволяє розробникам ефективно використовувати переваги FPGA для специфічних задач, одночасно використовуючи звичні та потужні можливості процесорів ARM для загального управління системою.

Основні характеристики подані у табл. 2.1.

Таблиця 2.1 - Основні характеристики Zynq-7000

<b>Specification</b>	<b>Details</b>
Manufacturer	Xilinx
Chip Series	Zynq-7000
Processor	Dual ARM Cortex-A9 MPCore
Logic Cells	Up to 235K
DSP Slices	Up to 740
Block RAM	Up to 17.6 Mb
Transceivers	Up to 16 (6.25 Gb/s)
Clock Management	PLLs, MMCM
On-board Interfaces	Depends on specific board model

### 2.1.2 PYNQ-Z2

PYNQ-Z2 є розробницькою платою, створеною з метою спрощення використання програмованих логічних інтегральних схем (FPGA) завдяки використанню Python-орієнтованого фреймворку PYNQ [12]. Ця плата базується на чіпі Xilinx Zynq, що комбінує в собі можливості традиційного FPGA з можливостями процесора ARM Cortex-A9, що надає велику гнучкість та ефективність для розробки вбудованих систем.

Особливістю PYNQ-Z2 є її орієнтація на Python, що дозволяє розробникам використовувати Python для програмування логіки FPGA, роблячи цей процес доступнішим для тих, хто не має глибоких знань у галузі апаратного проектування.

Ця платформа також підтримує Jupyter Notebook, що забезпечує інтерактивне середовище для розробки та тестування програм.

PYNQ-Z2 має різноманітні інтерфейси для підключення, включаючи HDMI, USB, Ethernet та інші, що робить її придатною для широкого спектру застосувань, від освітніх проектів до промислових застосувань. Оснащена також аудіо та відео входами та виходами, плата PYNQ-Z2 може використовуватися для розробки мультимедійних систем.

Завдяки своїй гнучкості та відносно простому використанню, PYNQ-Z2 стала популярним вибором у галузі освіти та серед хобістів, які цікавляться розробкою FPGA та вбудованими системами.

Табл. 2.2 надає огляд основних технічних характеристик PYNQ-Z2, включаючи тип чіпа, характеристики процесора, обсяг логічних клітин, DSP блоків, блокової пам'яті, можливості управління годинником, наявні інтерфейси для підключення та методи програмування.

Таблиця 2.2 – Основні характеристики Xilinx Zynq-7000 (Z-7020)

<b>Specification</b>	<b>Details</b>
Manufacturer	TUL Corporation
Chip	Xilinx Zynq-7000 (Z-7020)
Processor	Dual ARM Cortex-A9
Logic Cells	85K
DSP Slices	220
Block RAM	140 36Kb BRAM blocks
Clock Management	PLLs, MMCM
On-board Interfaces	HDMI, Audio, USB, Ethernet, Pmod
Programming	Python, Jupyter Notebook
Special Features	Designed for PYNQ framework, Python programming

Особливо важливою є підтримка Python та фреймворку PYNQ, що робить цю плату зручною для розробки та освітніх цілей.

### 2.1.3 Basys 3 Artix-7 FPGA Board

Basys 3 Artix-7 FPGA Board є популярною розробницькою платою від компанії Digilent, призначеною для студентів та хобістів, які цікавляться вивченням та використанням FPGA (поле програмованих вентилів). Ця плата базується на FPGA чіпі Artix-7 від Xilinx, що надає достатньо потужності для виконання широкого спектру завдань, від простих навчальних проектів до складніших розробок [13].

Основні технічні характеристики плати наведені в табл. 2.3.

Таблиця 2.3 - Основні технічні характеристики Basys 3 Artix-7 FPGA Board

Specification	Details
Manufacturer	Digilent
Chip	Xilinx Artix-7 (XC7A35T-1CPG236C)
Logic Cells	33K
DSP Slices	90
Block RAM	1.8 Mb
Clock Management	5 Clock Management Tiles
On-board Interfaces	USB, VGA, Pmod connectors
Programming	Vivado Design Suite
Target Use	Education, Hobbyist Projects

Basys 3 оснащена численними інтерфейсами та компонентами, які дозволяють реалізувати різноманітні проекти без потреби в додатковому обладнанні. На платі є роз'єми для вводу-виводу, світлодіоди, кнопки, перемикачі та семисегментні індикатори, які можна використовувати для безпосередньої взаємодії з FPGA. Однією з ключових переваг Basys 3 є її підтримка середовища

Vivado Design Suite від Xilinx, що є одним з найпопулярніших інструментів для розробки на FPGA. Це робить плату ідеальною для освітніх установ та самонавчання, оскільки Vivado часто використовується в промисловості.

Крім того, Basys 3 має модульну структуру, що дозволяє легко розширювати її функціональність за допомогою додаткових модулів та плат розширення. Ця гнучкість робить плату відмінним вибором для проектів, які вимагають спеціалізованого апаратного забезпечення.

Загалом, Basys 3 Artix-7 є надійною, доступною та багатofункціональною платою, яка підходить для широкого спектру освітніх та хобійних проектів у галузі цифрової логіки та FPGA. Основні характеристики цієї плати наведені в таблиці

#### **2.1.4 Arty A7**

Arty A7 є розробницькою платою, створеною компанією Digilent, яка орієнтована на використання в освіті, хобі-проектах та дослідженнях. Плата базується на чіпі Artix-7 від Xilinx, який є високопродуктивним FPGA, забезпечуючи гнучкість та потужність для реалізації широкого спектру проектів.

Arty A7 [14] оснащена різноманітними можливостями для вводу-виводу, включаючи USB, Ethernet і ряд портів Pmod, які роблять її ідеальною для створення вбудованих систем та застосувань IoT. Ці інтерфейси забезпечують легке підключення до різних периферійних пристроїв та дозволяють розширити функціональність плати.

Підтримка Vivado Design Suite від Xilinx надає користувачам потужні інструменти для розробки та налагодження проектів на FPGA. Це робить Arty A7 доступною для широкого кола користувачів від студентів, які тільки починають працювати з FPGA, до досвідчених розробників, які шукають гнучку та масштабовану платформу для складніших проектів.

Загалом, Arty A7 є популярним вибором у галузі цифрової логіки та вбудованих систем, забезпечуючи баланс між потужністю, гнучкістю та доступністю.

Ключові технічні характеристики Arty A7 FPGA Board подані у табл. 2.4.

Таблиця 2.4 - Ключові технічні характеристики Arty A7 FPGA Board

<b>Specification</b>	<b>Details</b>
Manufacturer	Digilent
Chip	Xilinx Artix-7
Processor	Not Applicable
Logic Cells	Up to 35K
DSP Slices	Up to 90
Block RAM	Up to 4.9 Mb
Clock Management	PLLs, MMCM
On-board Interfaces	USB, Ethernet, Pmod connectors
Programming	Vivado Design Suite
Target Use	Education, Hobbyist Projects, Prototyping

### 2.1.5 DE10-Nano Kit

DE10-Nano Kit є компактною та потужною розробницькою платою, створеною Terasic для Intel Cyclone V SoC (System on Chip). Ця плата є відмінним інструментом для навчання, дослідження та розробки в галузі FPGA, оскільки вона поєднує в собі можливості програмованої логіки FPGA та двоядерного процесора ARM Cortex-A9 [15].

DE10-Nano Kit вирізняється своєю гнучкістю та функціональністю. Плата містить широкий спектр інтерфейсів для підключення, включаючи HDMI, USB, Ethernet, а також має порти для розширення можливостей, що робить її ідеальною для розробки різноманітних вбудованих систем та застосувань IoT. Також вона оснащена мікросхемою для управління живленням, акселерометром та іншими датчиками, що збільшує її універсальність.

Ця плата широко використовується у навчальних установах та серед хобістів, оскільки вона дозволяє легко вивчати та експериментувати з FPGA та SoC. Ця плата популярна серед розробників, які хочуть розвивати навички в області

цифрової логіки, обробки сигналів та вбудованих систем. Табл. 2.5 відображає основні технічні характеристики DE10-Nano Kit.

Таблиця 2.5 - Основні технічні характеристики DE10-Nano Kit

<b>Specification</b>	<b>Details</b>
Manufacturer	Terasic for Intel
Chip	Cyclone V SE 5CSEBA6U23I7N
Processor	Dual-core ARM Cortex-A9
Logic Elements	110K
DSP Blocks	112
Block RAM	5.5 Mb
Clock Management	PLLs
On-board Interfaces	HDMI, USB, Ethernet, GPIO, Arduino and MIPI connectors
Programming	Quartus Prime
Target Use	Education, Prototyping, IoT

Завдяки своїй потужності, гнучкості та доступності, DE10-Nano Kit є відмінним вибором для проектів, що вимагають високої продуктивності та різноманітності функціональних можливостей.

### **2.1.6 Zybo Z7**

Zybo Z7 є багатофункціональною, готовою до використання платою для розробки вбудованого програмного забезпечення та цифрових схем, побудованою на базі сімейства Xilinx Zynq-7000. Це друге покоління популярної плати Zybo, яка була випущена у 2012 році. Сімейство Zynq базується на архітектурі "Все Програмоване Система-на-Чипі" (AP SoC) від Xilinx, яка тісно інтегрує двоядерний процесор ARM Cortex-A9 з логікою FPGA (полів програмованих вентильних матриць) серії Xilinx 7. Zybo Z7 доповнює Zynq багатим набором мультимедійних та комунікаційних периферій, створюючи потужний одноплатний комп'ютер,

навіть до того, як врахувати гнучкість та потужність, яку додає FPGA. Функціональність Zybo Z7, що підтримує відео, включаючи сумісний з MIPI CSI-2 конектор Pcam, вхід HDMI, вихід HDMI та високу пропускну здатність DDR3L, була обрана, щоб зробити її доступним рішенням для висококласних застосувань вбудованого бачення, для яких FPGA Xilinx є популярними. Підключення додаткового апаратного забезпечення спрощується за допомогою конекторів Pmod Zybo Z7, що дозволяє отримати доступ до каталогу Digilent з більш ніж 70 дошок периферійних пристроїв Pmod, включаючи контролери двигунів, сенсори, дисплеї та багато іншого [16].

Таблиця 2.6 - Основні технічні характеристики Zybo Z7

<b>Feature</b>	<b>Details</b>
Memory	1 ГБ DDR3L з 32-бітною шиною на 1066 МГц
Flash Memory	16 МБ Quad-SPI
Buttons and Switches	6 кнопок (2 з'єднані з процесором), 4 слайдери-перемикачі
LEDs	5 світлодіодів (1 з'єднаний з процесором), 1 RGB світлодіод
Pmod Ports	5 портів
Processor I/O	8 загальних
FPGA I/O	40 загальних
Analog Interfaces	4 аналогові пари, здатні до 0-1.0V диференційного входу до XADC
With SDSoc Voucher	Поставляється з ваучером SDSoc для розширення можливостей розробки

Zybo Z7 розроблено з метою надання гнучкого та ефективного рішення для вбудованих систем, зокрема у сфері вбудованого бачення, де FPGA від Xilinx

знаходять широке застосування через свою високу пропускну здатність та адаптивність.

## **2.2 Вибір модуля вимірювання освітленості**

### **2.2.1 SparkFun Ambient Light Sensor Breakout з TЕМТ6000**

SparkFun Ambient Light Sensor Breakout з TЕМТ6000 є компактним і простим у використанні модулем для вимірювання рівня освітлення в навколишньому середовищі. Основний компонент цього модуля - це фототранзистор TЕМТ6000, який є чутливим до видимого світла, схоже на людське око, і відтворює його сприйняття освітлення [17].

Фототранзистор TЕМТ6000 має високу чутливість та широкий кут огляду, роблячи його ідеальним для використання у застосуваннях, де необхідно відстежувати загальний рівень освітлення. Сенсор виводить аналоговий сигнал, який пропорційний інтенсивності освітлення, що потім може бути зчитано за допомогою аналогового входу мікроконтролера або іншої системи.

SparkFun Ambient Light Sensor Breakout має малі розміри та включає в себе стандартні отвори для монтажу, що робить його зручним для інтеграції у різноманітні проекти. Модуль може бути легко підключений до розробницьких плат, таких як Arduino, за допомогою стандартних дротів підключення. Основні технічні характеристики наведені в табл. 2.7.

Завдяки своїй простоті та ефективності, цей сенсор освітлення використовується у широкому спектрі застосувань, включаючи автоматизацію освітлення в будинку, системи безпеки, пристрої, що відповідають на зміни у рівні освітлення, та навчальні проекти.

Таблиця 2.7 – Основні технічні характеристики SparkFun Ambient Light Sensor Breakout з TEMT6000

Specification	Details
Sensor Type	Phototransistor
Peak Wavelength	570 nm (visible spectrum)
Spectral Bandwidth	Approx. 100 nm
Illuminance Range	Suitable for ambient light detection
Output	Analog voltage
Viewing Angle	Approx. 120 degrees
Supply Voltage	3.3V to 5V
Dimensions	Small breakout board size
Applications	Home automation, security systems, light-responsive gadgets

### 2.2.2 Adafruit TSL2591 High Dynamic Range Digital Light Sensor

Adafruit TSL2591 є цифровим датчиком світла з високим динамічним діапазоном, який використовується для точного вимірювання рівня освітлення. Цей сенсор від Adafruit Industries є високоефективним і здатний вимірювати від дуже низьких рівнів освітлення (як у місячну ніч) до дуже яскравого сонячного світла [18].

Основною перевагою TSL2591 є його висока чутливість та здатність розрізняти дуже тонкі зміни в освітленості. Це досягається завдяки використанню двох діодів: одного для інфрачервоного світла та іншого для видимого світла, що дозволяє датчику точно вимірювати рівні освітленості в різних умовах.

TSL2591 підключається через I2C інтерфейс, що робить його сумісним з більшістю мікроконтролерів та розробницьких плат, включаючи Arduino та Raspberry Pi. Це також полегшує інтеграцію сенсора в різноманітні проекти.

Завдяки своїм характеристикам, Adafruit TSL2591 часто використовується у проєктах, пов'язаних з автоматизацією будинку, системами контролю клімату, екологічним моніторингом та будь-яких інших застосуваннях, де необхідно точно вимірювати рівень освітлення.

### **2.2.3 Grove - Light Sensor v1.2**

Grove - Light Sensor v1.2 є модулем для вимірювання рівня освітлення, який розроблено компанією Seeed Studio. Цей сенсор використовує фоторезистор, який змінює свій електричний опір в залежності від інтенсивності освітлення. Чим більше світла падає на фоторезистор, тим менший стає його опір.

Grove - Light Sensor v1.2 легко підключається до різних платформ, таких як Arduino, за допомогою системи Grove, яка забезпечує стандартизовані роз'єми та інтерфейси для спрощення процесу розробки. Сенсор дає аналоговий вихід, що дозволяє легко інтегрувати його в електронні проєкти [19].

Використання Grove - Light Sensor v1.2 є дуже популярним у освітніх проєктах, хобі та прототипуванні, де потрібно вимірювати рівень освітлення. Це може включати автоматизацію освітлення, системи безпеки, садові контролери та інші застосування, де важливо знати рівень освітлення в навколишньому середовищі.

### **2.2.4 Seeed Studio Luminance Sensor**

Seeed Studio Luminance Sensor є високоефективним датчиком світла, розробленим для точного вимірювання рівня освітленості. Цей сенсор, який виробляється компанією Seeed Studio, відомою своїми інноваційними рішеннями в області електроніки та Інтернету речей, ідеально підходить для широкого спектру застосувань.

Використовуючи цифровий інтерфейс, зазвичай I2C, Seeed Studio Luminance Sensor забезпечує легке підключення до багатьох мікроконтролерів та розробницьких плат, таких як Arduino або Raspberry Pi. Ця особливість, разом із компактними розмірами, робить сенсор зручним для інтеграції у різноманітні проєкти.

Сенсор може бути використаний у багатьох сценаріях, від домашньої автоматизації та систем контролю освітлення до складніших додатків, таких як моніторинг довкілля та інтелектуальне управління освітленням в комерційних та промислових середовищах [20].

Завдяки своїй універсальності, точності та легкості використання, Seeed Studio Luminance Sensor є популярним вибором серед хобістів, освітніх установ та професійних розробників для втілення різноманітних інноваційних проектів у сфері освітлення та IoT.

### 2.2.5 Digilent Pmod ALS

Digilent PmodALS представляє собою простий пристрій, який трансформує світло у цифрові дані за допомогою вбудованого датчика навколишнього світла. Цей модуль, розроблений інженерами Digilent, базується на аналого-цифровому перетворювачі ADC081S021 від Texas Instruments та фототранзисторі ТЕМТ6000Х01 від Vishay Semiconductor. Основні характеристики:

- функціонує як простий датчик навколишнього світла;
- здатний перетворювати світлові сигнали на цифрові дані з роздільною здатністю 8 біт;
- має 6-контактний порт Pmod із SPI інтерфейсом.

Опис функціонування: PmodALS використовує фототранзистор для визначення рівня освітлення. Він вимірює інтенсивність світла, яка впливає на ALS, і генерує відповідний рівень напруги. Ця напруга подається на АЦП, який перетворює її у 8-бітові дані, де 0 означає низький рівень світла, а 255 – високий [21].

Інтерфейсування з Pmod: PmodALS комунікує з основною платою через SPI протокол в режимі SPI 3. АЦП у цьому модулі призначений тільки для читання, тому для SPI протоколу необхідні лише Chip Select, Master-In-Slave-Out та Serial Clock. Розміщення цих ліній на роз'ємі Pmod наведено у відповідній табл. 2.8.

Таблиця 2.8 - Таблиця опису виводів Digilent PmodALS

Пін	Функція
1	CS (Chip Select)
2	NC (Не Підключено)
3	SDO (Master-In-Slave-Out)
4	SCK (Serial Clock)
5	GND (Земля Живлення)
6	VCC (Джерело Живлення 3.3V/5V)

PmodALS передає дані на основну плату, коли аналого-цифровий перетворювач ADC081S021 знаходиться в стандартному режимі. Це досягається шляхом зниження рівня піну CS і виконує одне читання за кожні 16 циклів SCLK. Щоб PmodALS функціонував коректно, частота SCLK повинна знаходитися в діапазоні від 1 МГц до 4 МГц. Інформація передається на спадаючому фронті SCLK і є валідною на наступному зростаючому фронті SCLK, складаючись з трьох початкових нулів, восьми біт даних з MSB на початку, та чотирьох заключних нулів. Іноді може бути захоплений додатковий, четвертий, початковий нуль, якщо пін CS опускається для розпочатку нової передачі даних перед зростаючим фронтом SCLK.

Будь-яке зовнішнє живлення, що подається на PmodALS, має бути в межах від 2.7V до 5.25V, хоча рекомендується використовувати 3.3V для оптимальної роботи.

Щодо фізичних розмірів, контакти на роз'ємі розташовані з інтервалом у 100 міл, тоді як довжина друкованої плати PCB становить 0.8 дюйма по сторонах, які паралельні роз'єму, і 0.8 дюйма по сторонах, що розташовані перпендикулярно до нього.

## 2.3 Вибір програмного забезпечення

### 2.3.1 Xilinx ISE

Xilinx ISE (Integrated Synthesis Environment) є програмним забезпеченням, яке було розроблено компанією Xilinx для розробки на її FPGA та CPLD мікросхемах. ISE пропонує повний набір інструментів для проектування, від синтезу та моделювання до оптимізації та відладки. Цей пакет програмного забезпечення був основним вибором для розробників, які працювали з мікросхемами Xilinx, особливо зі старшими поколіннями, такими як серії Spartan-6 та Virtex-6 [22].

ISE забезпечує інтегроване середовище, що дозволяє розробникам створювати складні системи. Воно включає інструменти для HDL (Hardware Description Language) синтезу, які перетворюють код, написаний на Verilog або VHDL, у логічні схеми. Також ISE включає симулятори для тестування та перевірки дизайну перед фізичною реалізацією на FPGA.

Основні особливості Xilinx ISE Design Suite:

- інтегрована середа розробки, яка забезпечує ефективне управління всіма етапами проектування;
- підтримка різних методів опису розроблюваних устроїв та систем, включаючи графічні у формі принципів схем та текстові з використанням мов опису апаратури HDL;
- можливість використання проектів, підготовлених у інших системах проектування;
- схемотехнічний редактор з широким набором уніфікованих бібліотек;
- інтерактивні засоби створення HDL-описів, які автоматично формують шаблони на основі інформації, наданої користувачем;
- використання високоефективних вбудованих засобів синтезу HDL-проектів;

- інтегрований інтерфейс для засобів синтезу та моделювання інших компаній;
- розвинені засоби верифікації проекту для скорочення загального часу розробки;
- автоматичні засоби трасування проектів у кристалах програмованої логіки та розширюваних процесорних платформ;
- єдині засоби конфігурування ПЛІС та програмованих систем на кристалі різних семейств компанії Xilinx;
- вбудований комплект допомогательних програмних засобів для підвищення ефективності проектування;
- доступний та інтуїтивно зрозумілий інтерфейс користувача, а також наявність у кожному модулі пакета довідкової системи, що скорочують час освоєння САПР.

Хоча ISE все ще використовується для підтримки існуючих проектів, Xilinx переходить на Vivado Design Suite для своїх новіших мікросхем. Vivado пропонує покращення у термінах використання ресурсів, продуктивності та можливостей для нових серій FPGA.

ISE був особливо популярний в освітніх установах та серед хобістів завдяки своїй доступності та підтримці старших мікросхем Xilinx, що робило його відмінним інструментом для навчання основам цифрового проектування.

### **2.3.2 AMD Vivado Design Suite**

AMD Vivado Design Suite, розроблена Xilinx, яка тепер є частиною AMD, є передовим інструментом для розробки FPGA та SoC (системи на кристалі). Vivado Design Suite надає інтегроване середовище для синтезу, розміщення, маршрутизації та аналізу дизайнів FPGA. Основною характеристикою Vivado є його здатність підтримувати більш складні та великомасштабні дизайни, які характерні для новіших серій FPGA від Xilinx, таких як UltraScale та 7-серія [22].

Серія САПР Xilinx Vivado Design Suite є сучасним рішенням для автоматизованого проектування цифрових пристроїв та встрайованих

микропроцесорних систем. Ця серія базується на широкому використанні IP-ядер, що дозволяє значно скоротити загальний час розробки. Вона включає центральний депозитарій IP-компонентів, який містить ядра, розроблені компанією Xilinx, а також сторонніми фірмами та самими розробниками. Vivado Design Suite пропонує інструмент IP Integrator для непосредственного включення і настройки цих ядер у склад проектів. Процес проектування апаратної платформи можна повністю виконувати в інтегрованій середовищі розробки Vivado Integrated Design Environment (IDE) без необхідності використання додаткових інструментів, таких як Xilinx Platform Studio.

#### Основні особливості Vivado Design Suite:

- значне збільшення швидкості процесів розміщення та трасування проектів у кристалах програмованої логіки та розширюваних процесорних платформах Zynq-7000 AP SoC завдяки використанню ефективніших алгоритмів та глибшому аналізу топології елементів;
- підвищення щільності розміщення проектів у кристалах;
- збільшення продуктивності та зниження споживаної потужності завдяки новим технологіям;
- підтримка опису модулів та апаратної частини на HDL-мовах високого рівня VHDL та Verilog;
- підтримка средств синтезу високого рівня Vivado High-Level Synthesis для використання мов програмування C/C++ у проектуванні;
- можливість імпорту проектів, розроблених у інших системах проектування;
- наявність різних стратегій синтезу, розміщення та трасування проектів для оптимізації результатів;
- використання формату Xilinx Design Constraint (XDC) для запису часових та топологічних обмежень проекту;
- можливість углобленого аналізу синтезованого опису проекту;
- візуальне планування розміщення користувацьких виводів кристала;

- генерація спеціалізованих звітів, що надають ісчерпну інформацію про хід та результати всіх процесів;
- підтримка непроектного режиму для більшої гнучкості управління процесом розробки;
- тісна інтеграція з інструментами Xilinx Software Development Kit для розробки програмних компонентів;
- інтегрований інтерфейс для средств синтезу та моделювання від третіх фірм.

Vivado пропонує більш ефективний синтез та аналіз залежностей, використовуючи вдосконалені алгоритми, що значно зменшують час компіляції порівняно з попередником, Xilinx ISE. Іншою ключовою особливістю є High-Level Synthesis (HLS), яка дозволяє розробникам описувати апаратні блоки високого рівня за допомогою мов програмування, таких як C++, і автоматично перетворювати ці описи у FPGA оптимізовані апаратні компоненти.

Vivado також включає набір інструментів для відладки та аналізу, які полегшують виявлення та вирішення проблем у складних FPGA дизайнах. Це включає можливості для інтерактивного графічного відображення дизайнів та їхніх компонентів, що дозволяє розробникам краще розуміти та оптимізувати свої проекти.

Vivado використовується у широкому спектрі застосувань, від промислових і телекомунікаційних систем до авіації та космічних технологій, завдяки своїй здатності оптимізувати дизайни для максимальної продуктивності та ефективного використання ресурсів FPGA.

### **2.3.3 Xilinx Vitis Unified Software Platform**

Xilinx Vitis Unified Software Platform є комплексним рішенням для розробки, яке поєднує в собі можливості програмування FPGA, систем на кристалі (SoC) та адаптивних обчислювальних прискорювальних платформ (ACAP). Платформа Vitis розроблена для забезпечення більшої гнучкості та доступності у розробці

апаратного забезпечення, особливо для інженерів-програмістів, які можуть не мати глибоких знань у галузі FPGA.

Однією з ключових характеристик Vitis є її здатність інтегрувати програмування на мовах високого рівня, таких як C, C++ та Python, з традиційними методами розробки FPGA. Це дозволяє розробникам використовувати знайомі інструменти та методології при проектуванні комплексних систем [22].

Vitis включає в себе комплексний набір бібліотек та оптимізованих функцій, які спрощують використання FPGA для різних завдань обчислювальної обробки. Це включає, але не обмежується, операціями з обробки даних, машинним навчанням, обробкою сигналів тощо.

Ще однією важливою особливістю Vitis є її інтегроване середовище розробки, яке об'єднує всі інструменти та функції, необхідні для проектування, відладки та розгортання програмного забезпечення та апаратного обладнання. Це інтегроване середовище дозволяє розробникам ефективно взаємодіяти між програмними та апаратними компонентами системи.

Крім того, Vitis підтримує розробку для широкого спектру апаратних платформ Xilinx, включаючи FPGA, SoC та ACAP. Це робить платформу ідеальною для різноманітних застосувань, від вбудованих систем до високопродуктивних обчислювальних рішень.

У цілому, Xilinx Vitis Unified Software Platform становить революційний підхід у галузі розробки FPGA та систем на кристалі, забезпечуючи розробникам потужні, але в той же час доступні інструменти для створення інноваційних обчислювальних рішень.

#### **2.3.4 IFTTT Hue service**

IFTTT (If This Then That) Hue service - це сервіс, який дозволяє автоматизувати взаємодію та управління освітленням від Philips Hue через платформу IFTTT. Philips Hue є популярною системою розумного освітлення, яка дозволяє користувачам контролювати відтінки, інтенсивність та час роботи світлових джерел через мобільний додаток або інтегровані рішення для дому [23].

IFTTT Hue service дозволяє створювати так звані "аплети" (applets), які використовують просту логіку "якщо це, то те" для автоматизації дій з освітленням. Наприклад, користувач може налаштувати аplet таким чином, що коли вони приходять додому (визначено за допомогою геолокації на смартфоні), світло в їхньому будинку автоматично вмикається.

Користувачі можуть налаштувати різноманітні аплети, використовуючи різні тригери та дії. Тригери можуть включати зміни погоди, отримання електронних листів, повідомлень у соціальних мережах, розташування GPS, тощо. Дії можуть включати вмикання/вимикання світла, зміну кольору або яскравості, а також активацію різних сценаріїв освітлення.

Цей сервіс особливо корисний для створення індивідуалізованих та автоматизованих рішень для дому, забезпечуючи більш комфортне та ефективне використання освітлювальних систем. IFTTT Hue service інтегрується з широким спектром інших сервісів та пристроїв, що дозволяє користувачам створювати комплексні системи автоматизації дому, які можуть включати не тільки освітлення, але й інші елементи розумного будинку.

Щоб користувач міг використовувати IFTTT Hue Service для автоматизації своєї системи освітлення Philips Hue, він має виконати кілька кроків:

- створення або вхід у обліковий запис IFTTT: Спочатку користувач повинен відвідати веб-сайт IFTTT або завантажити мобільний додаток IFTTT і створити новий обліковий запис або увійти в існуючий;
- підключення системи Philips Hue до IFTTT: У розділі "My Services" на платформі IFTTT користувач повинен знайти і підключити Philips Hue сервіс, увійшовши в свій обліковий запис Philips Hue та давши IFTTT необхідні дозволи для доступу;
- активація готових аплетів: На платформі IFTTT вже доступний широкий спектр готових аплетів для Philips Hue, які користувач може переглянути та активувати, наприклад, для автоматичного вмикання світла при приході додому;

- створення власних аплетів: Для створення персоналізованого аплету користувач повинен вибрати "Create", потім "This" для встановлення тригера (наприклад, зміна погоди) і "That" для вибору дії системи освітлення Philips Hue (наприклад, вмикання світла);
- настройка деталей аплету: Користувач конфігурує налаштування аплету, вказуючи специфічні параметри для тригера та дії, такі як час доби або певні умови освітлення;
- збереження та активація аплету: Після завершення налаштувань користувач зберігає аплет і активує його, після чого аплет буде автоматично реагувати на задані умови.

Використовуючи IFTTT Hue Service, користувач може значно розширити можливості своєї системи розумного освітлення Philips Hue, інтегрувавши її з багатьма іншими сервісами та пристроями, створюючи інтелектуальну та автоматизовану систему освітлення вдома.

Перед тим, як почати створювати, користувачеві слід спершу ознайомитися з тим, як працює IFTTT, щоб зрозуміти досвід користування платформою.

Послуги є основними будівельними блоками IFTTT. Прикладами таких послуг є Amazon Alexa, Twitter, Dropbox, Fitbit і esobee. Кожен продукт користувача буде представлений на IFTTT як послуга. Кожна послуга матиме власну цільову сторінку на IFTTT, яку називають "сторінкою сервісу" (рис. 2.1).

Служби об'єднуються в аплеті. Аплет — це автоматизований робочий процес, що з'єднує дві або більше послуги разом, дозволяючи користувачам робити те, чого ці послуги не можуть зробити окремо (рис. 2.2). Наприклад, користувач може використовувати аплет, щоб вмикати освітлення Philips Hue, коли камера безпеки Arlo виявляє рух, або призупиняти пилосос iRobot перед зустрічами Zoom.

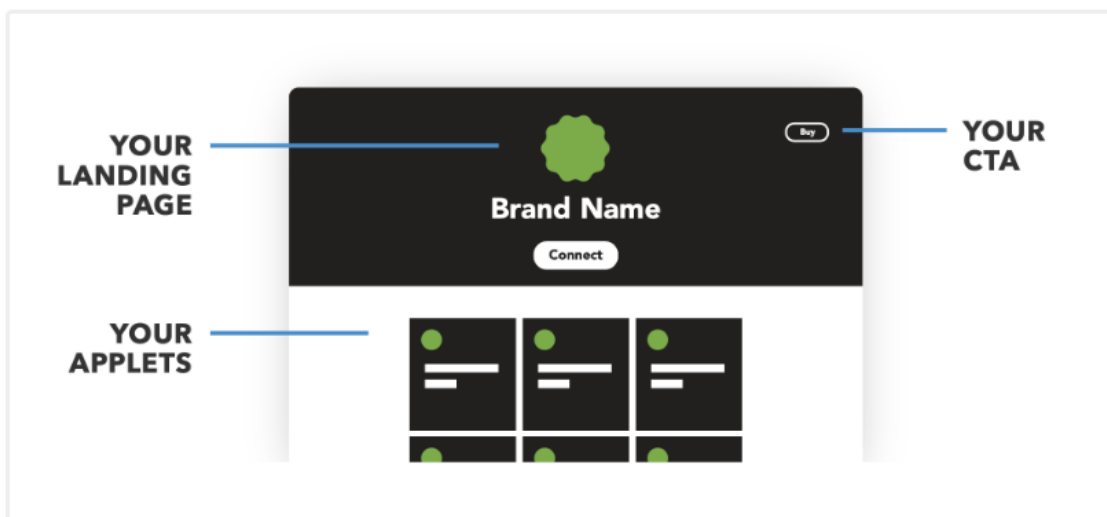


Рисунок 2.1 – Сторінка сервіса

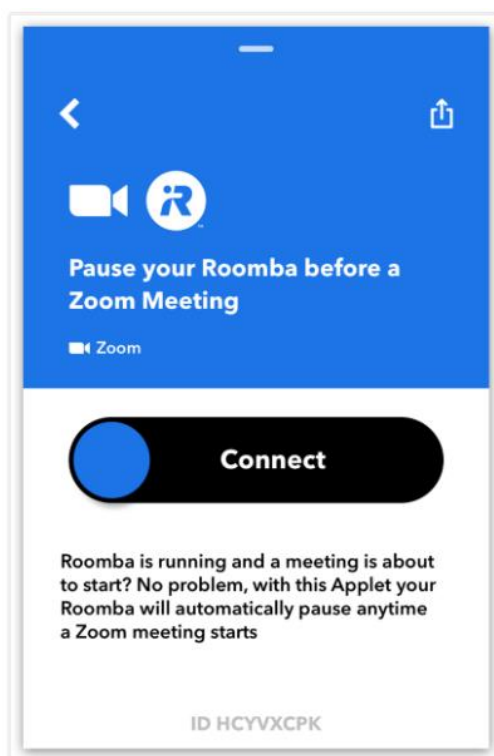


Рисунок 2.2 – Вікно аплету

Аплети складаються з тригерів, запитів і дій. Тригери активують аплет, запити забезпечують додаткові умови, а дії є результатом запуску аплету. Аплети можуть публікувати як адміністратор служби, так і самі користувачі.

Тригери, дії та запити. API продукту користувача складається з різних кінцевих точок, які використовуються для отримання, запису або відправки даних. Кожна кінцева точка API відповідає одному тригеру, дії чи запиту, які користувач хоче додати до своєї служби. Тригери відправляють дані до IFTTT, коли з продуктом відбувається подія (наприклад, вмикання термостата). Запити дозволяють IFTTT отримувати додаткові дані про продукт (наприклад, поточні налаштування температури). Дії збирають інструкції від IFTTT для виконання дій із продуктом (наприклад, вмикання пристрою).

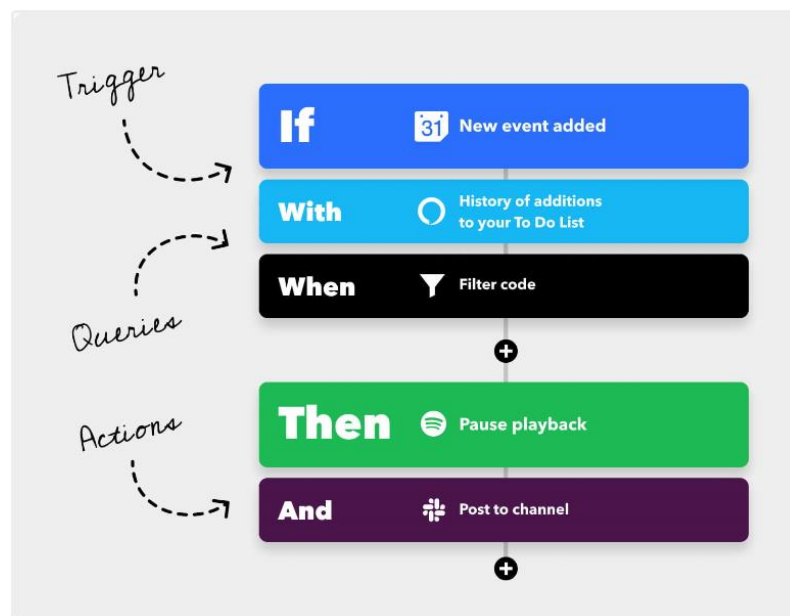


Рисунок 2.3 - Тригери, дії та запити

### 2.3.5 AMD PYNQ Framework

AMD PYNQ Framework, який раніше розвивався Xilinx, а тепер належить до AMD після їхнього злиття, є відкритим проектом, спрямованим на полегшення програмування пристроїв Zynq, які є сімейством інтегральних схем FPGA (Field Programmable Gate Array). Основна ідея PYNQ полягає у використанні Python для розробки та реалізації додатків на базі Zynq, значно спрощуючи процес програмування FPGA, завдяки доступності та читабельності Python. PYNQ

забезпечує високорівневу абстракцію, яка дозволяє розробникам використовувати потужності FPGA без глибоких знань мов опису апаратури, таких як VHDL чи Verilog. Часто для роботи з PYNQ використовуються Jupyter Notebooks, що є інтерактивним середовищем розробки, дозволяючи поєднувати текст, код і візуалізації в одному документі, що робить його потужним інструментом для навчання, прототипування та документування. PYNQ включає бібліотеку "overlays" - готові високорівневі проекти та інтерфейси для загальних функціональностей апаратури, які можна використовувати як є або модифікувати під конкретні завдання, зменшуючи час та зусилля, необхідні для розробки. З PYNQ розробники можуть використовувати паралельні обчислювальні можливості FPGA для прискорення роботи з великими обсягами даних, що є корисним у таких додатках, як машинне навчання, обробка зображень та аналіз даних в реальному часі. Як відкритий проект, PYNQ має розвиваючу спільноту та екосистему, що сприяє розвитку інструментів, overlays та матеріалів підтримки. Хоча спочатку PYNQ був розроблений для SoCs (систем на кристалі) Zynq-7000 від Xilinx, його підтримка розширена до інших продуктів Xilinx і потенційно інших платформ FPGA у майбутньому. Узагальнюючи, AMD PYNQ Framework орієнтований на зроблення програмування FPGA більш доступним та ефективним, особливо для розробників програмного забезпечення та дослідників, які можуть не мати великого досвіду у традиційних практиках розробки FPGA.

Типові вбудовані системи підтримують фіксовану комбінацію периферійних пристроїв (наприклад, SPI, IIC, UART, відео, USB). Також можуть бути доступні деякі GPIO (загального призначення вхід/вихід). Кількість доступних GPIO в системах на базі CPU зазвичай обмежена, і GPIO також контролюються основним CPU. Оскільки основний CPU керує рештою системи, продуктивність GPIO зазвичай обмежена [24].

Платформи Zynq зазвичай мають набагато більше вхідно-вихідних контактів, ніж типова вбудована система. Додаткові апаратні контролери та додаткові програмовані процесори можуть бути реалізовані в програмованій логіці

(PL) і підключені до зовнішніх інтерфейсів. Це означає, що продуктивність на цих інтерфейсах може бути набагато вищою, ніж у інших вбудованих системах.

PYNQ працює на Linux, який за замовчуванням використовує наступні периферійні пристрої Zynq PS: SD-карту для завантаження системи та розміщення файлової системи Linux, Ethernet для підключення до Jupyter notebook, UART для доступу до Linux терміналу, і USB.

USB-порт та інші стандартні інтерфейси можуть бути використані для підключення стандартних периферійних пристроїв USB та інших до Zynq PS, де їх можна керувати з Python/Linux. Образ PYNQ включає драйвери для найбільш поширених веб-камер USB, WiFi-периферій та інших стандартних пристроїв USB.

Інші периферійні пристрої можуть бути підключені до Zynq PL та доступні з нього. Наприклад, HDMI, аудіо, кнопки, перемикачі, світлодіоди та загальнопризначені інтерфейси, включаючи Pmods та Arduino. Оскільки PL програмований, для використання цих периферійних пристроїв або інтерфейсів має бути завантажений контролер.

У Vivado включена бібліотека апаратних IP, яку можна використовувати для підключення до широкого діапазону стандартів інтерфейсів та протоколів. PYNQ забезпечує Python API для низки загальноприйнятих периферійних пристроїв, включаючи відео (HDMI вхід та вихід), GPIO-пристрої (кнопки, перемикачі, світлодіоди) та сенсори та приводи. API PYNQ також може бути розширено для підтримки додаткових IP.

Платформи Zynq зазвичай мають один або кілька роз'ємів або інтерфейсів, які дозволяють підключати зовнішні периферійні пристрої або безпосередньо підключати до контактів Zynq PL. Різноманітні периферійні пристрої можуть бути підключені до інтерфейсів Pmod та Arduino. Інші периферійні пристрої можуть бути підключені до цих портів через адаптери або з використанням макетної плати. Зауважте, що хоча периферійний пристрій може бути фізично підключений до контактів Zynq PL, контролер має бути побудований в оверлеї, а програмний драйвер наданий, перш ніж периферійний пристрій може бути використаний.

Бібліотеки PYNQ забезпечують підтримку підсистеми PynqMicroBlaze, що дозволяє завантажувати готові додатки та створювати та компілювати нові додатки з Jupyter.

PYNQ також забезпечує підтримку для низькорівневого керування оверлеєм, включаючи читання/запис вводу-виводу з пам'яттю, розподіл пам'яті (наприклад, для використання майстром PL), керування та управління оверлеєм (завантаження оверлею, читання IP в оверлеї) та низькорівневе керування PL (завантаження бітового потоку).

Підсистема MicroBlaze у PYNQ. Підсистема PYNQ MicroBlaze може керуватися класом PynqMicroblaze. Це дозволяє завантажувати програми з Python, керувати виконанням, викликаючи сигнал скидання процесора, читати та записувати у спільну пам'ять даних, а також керувати перериваннями, отриманими від підсистеми.

Кожна підсистема PYNQ MicroBlaze міститься в процесорі вводу-виводу (IOP). IOP визначає набір комунікаційних та поведінкових контролерів, якими керує Python. Наразі з PYNQ поставляються три IOP: Arduino, PMOD і Logictools.

Підсистема PYNQ MicroBlaze складається з процесора MicroBlaze, міжсистемного з'єднання AXI, контролера переривань, запитувача переривань, зовнішнього системного інтерфейсу, а також блочної пам'яті RAM та контролерів пам'яті.

Міжсистемне з'єднання AXI з'єднує MicroBlaze з контролером переривань, запитувачем переривань та зовнішнім інтерфейсом.

Контролер переривань є інтерфейсом для інших комунікаційних або поведінкових контролерів, підключених до процесора MicroBlaze.

Запитувач переривань надсилає запити на переривання до системи обробки Zynq.

Зовнішній інтерфейс дозволяє підсистемі MicroBlaze спілкуватися з іншими комунікаційними, поведінковими контролерами або з пам'яттю DDR.

Блокова пам'ять RAM зберігає інструкції та дані MicroBlaze.

Блокова пам'ять RAM має два порти: один порт підключений до інструкційних та даних портів MicroBlaze; інший порт підключений до процесора ARM® Cortex®-A9 для зв'язку.

Якщо зовнішній інтерфейс підключений до пам'яті DDR, DDR може використовуватися для передачі великих сегментів даних між PS (Python) та підсистемою.

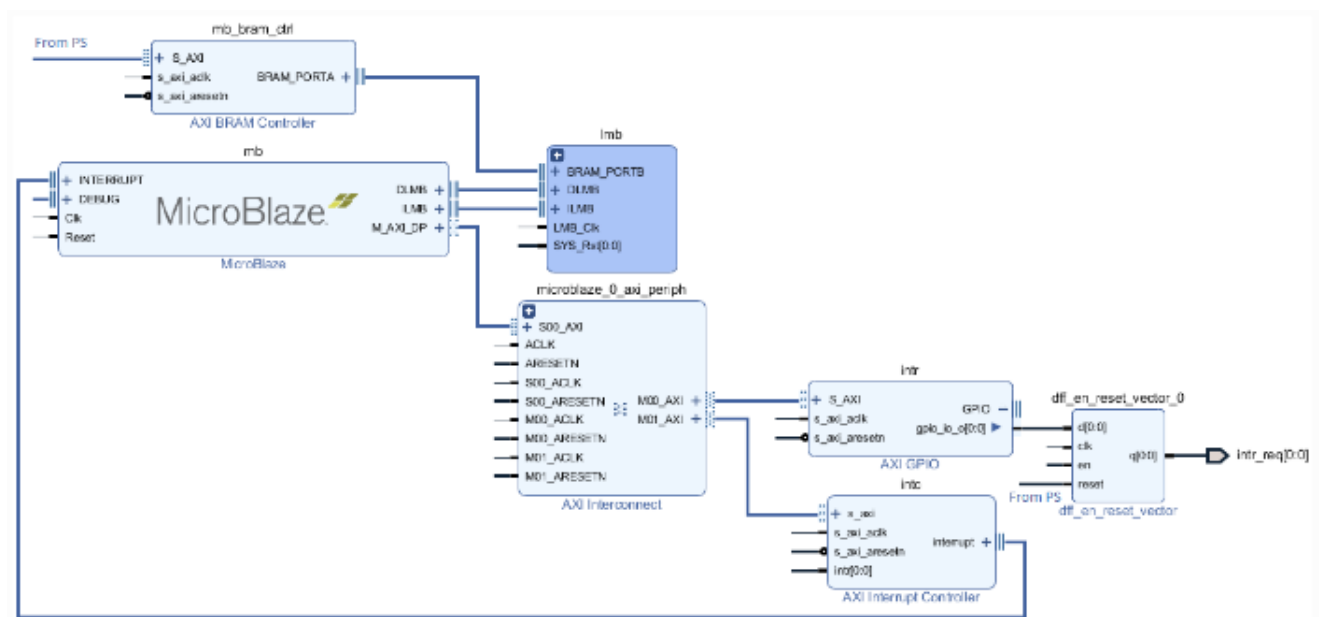


Рисунок 2.4 – Блок схема підсистеми MicroBlaze

Створення нової підсистеми PYNQ MicroBlaze вимагає ряду кроків та специфікацій, щоб забезпечити її сумісність та ефективну інтеграцію.

Кожна ієрархія, яка містить MicroBlaze і відповідає вимогам, щодо наявності AXI-доступної пам'яті коду, PS лінії переривання та лінії скидання, може використовуватися в PYNQ.

Щоб використовувати MicroBlaze з IPython magic, необхідно надати пакет підтримки плати (BSP). BSP генеруються з Xilinx SDK та містять всі драйвери та конфігураційні дані для периферійних пристроїв, приєднаних до MicroBlaze.

Створення BSP. PYNQ забезпечує TCL-скрипт для генерації BSP з файлу опису апаратного забезпечення, який можна знайти в директорії boards/sw\_repo репозиторію, разом з драйверами, необхідними для комунікації Python/C, та

бібліотекою апаратної абстракції `runqmb`. Створення та використання BSP включає наступні кроки:

- експорт апаратного забезпечення з Vivado для генерації HDF-файлу. У директорії `boards/sw_repo` запусить команду `make HDF=$HDF_FILE`. Якщо HDF не надано, то будуть згенеровані BSP для Base Overlay;
- копіювання згенерованого BSP на плату і переконання, що він названий `bsp_{$hierarchy}`, де `{$hierarchy}` - назва ієрархії MicroBlaze у вашому дизайні. Якщо у вас кілька MicroBlazes, це може бути префікс, спільний для всіх ієрархій;
- виклик `add_bsp(BSP_DIR)` в Python. Це можна зробити як частину ініціалізації вашого оверлею.

Ці кроки інтегрують BSP в середовище виконання PYNQ та дозволяють використовувати IPython `microblaze magic` з новою підсистемою MicroBlaze.

## 2.4 Висновки до розділу 2

Цей розділ охоплює детальний вибір апаратного та програмного забезпечення для реалізації проекту. В ньому розглядаються різні плати FPGA, такі як Xilinx Zynq-7000, PYNQ-Z2, Basys 3 Artix-7, Arty A7, DE10-Nano Kit, та Zybo Z7. Кожна з цих плат відрізняється своїми унікальними характеристиками, що робить їх придатними для різних застосувань, від освітніх проектів до промислових застосувань.

Також описано вибір модулів вимірювання освітленості, включаючи SparkFun Ambient Light Sensor Breakout з TCMТ6000, Adafruit TSL2591, Grove - Light Sensor v1.2, Seeed Studio Luminance Sensor та Digilent Pmod ALS. Ці сенсори відрізняються своїми технічними характеристиками та призначеннями, пропонуючи гнучкість у виборі відповідно до специфічних потреб проекту.

У розділі розглядаються різні програмні інструменти для розробки на FPGA, такі як Xilinx ISE, AMD Vivado Design Suite, Xilinx Vitis Unified Software Platform, а також IFTTT Hue service для автоматизації взаємодії з системою розумного освітлення Philips Hue. Ці інструменти та сервіси забезпечують широкий спектр

можливостей для проектування, розробки, тестування та автоматизації систем, що дозволяє розробникам ефективно використовувати FPGA для різноманітних задач.

Загалом, цей розділ надає вичерпний огляд доступного апаратного та програмного забезпечення, необхідного для розробки високоефективних та адаптивних систем на базі FPGA

## **3 РОЗРОБКА СИСТЕМИ**

### **3.1 Опис процесу розробки системи**

Процес створення образу PYNQ для розробницької плати Zybo включає кілька ключових етапів, які дозволяють інтегрувати плату з різними сенсорами та хмарними сервісами IoT. На цьому етапі проекту основна увага приділяється підготовці необхідного програмного забезпечення та апаратного середовища для подальшої ефективної роботи системи.

За основу взято плату ZYBO Z7, обрану через її гнучкість та наявність шести інтерфейсів Pmod. Ці інтерфейси є ключовими для підключення різноманітних сенсорів, таких як датчики температури, освітленості, акселерометри тощо. Використання плати ZYBO Z7 дозволяє розширити можливості системи і забезпечити необхідну адаптивність для різних типів сенсорів.

Першим кроком у процесі створення образу PYNQ є розробка базового дизайну апаратного забезпечення у Vivado. Цей етап включає реалізацію необхідної логіки та налаштувань для забезпечення функціональності системи.

Першим кроком у процесі створення образу PYNQ є розробка базового дизайну апаратного забезпечення у Vivado. Цей етап включає реалізацію необхідної логіки та налаштувань для забезпечення функціональності системи. Використання програмованих процесорів вводу-виводу (IOP) у програмованій логіці, таких як процесори Microblaze, дозволяє підтримувати різноманітні

стандарти ІО (ІС, SPI, UART, GPIO), що є критично важливим для взаємодії з різними сенсорами та пристроями.

Інтеграція з хмарними сервісами ІоТ.

Для інтеграції розробленої системи з хмарними сервісами ІоТ, такими як adafruit.io та IFTTT, проводиться серія налаштувань та конфігурацій. Це включає створення облікових записів у цих сервісах, отримання необхідних ключів для інтеграції, та розробку спеціалізованих аплетів для автоматизації процесів, таких як контроль освітлення на основі даних з сенсорів.

Ключовою перевагою цього підходу є можливість швидкого впровадження та легкої адаптації системи під різні потреби в контексті ІоТ, забезпечуючи високий рівень гнучкості та функціональності для широкого спектра застосувань.

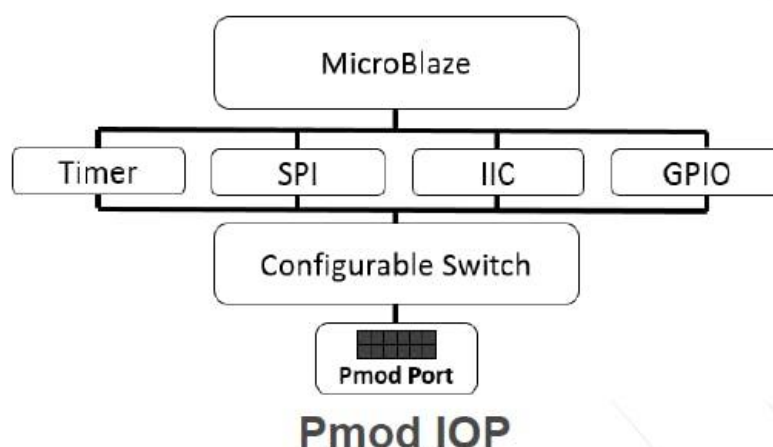


Рисунок 3.1- Концептуальна діаграма

Всі ІР, необхідні для реалізації ІОП, надаються в бібліотеці ІР Xilinx або PYNQ.

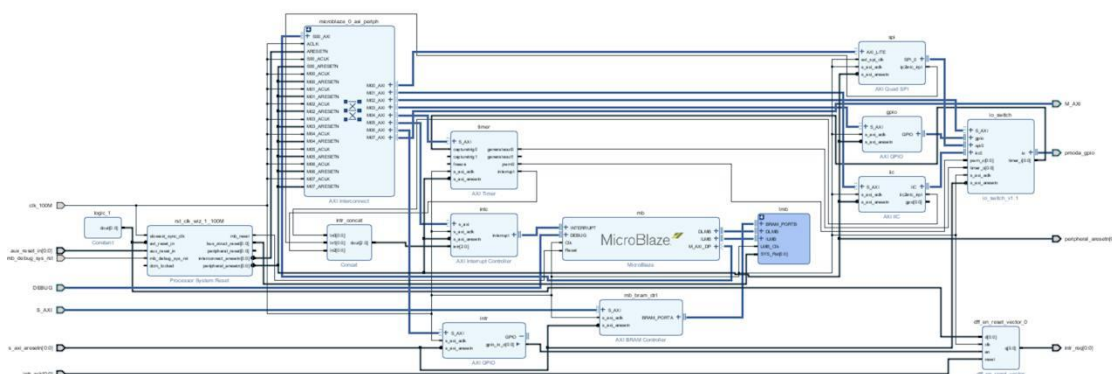


Рисунок 3.2 – IOP імплементація у Vivado

Щоб розпочати роботу з образом PYNQ, перше, що потрібно було зробити, – це клонувати директорію PYNQ, щоб можна було створити дизайн Vivado, а потім пізніше зібрати образ PYNQ.

Фреймворк PYNQ отримано з `git clone https://github.com/Xilinx/PYNQ.git`.

Фреймворк PYNQ є відкритим програмним рішенням, розробленим компанією Xilinx. Цей фреймворк спеціально призначений для легкого використання програмованої логіки FPGA у рамках платформи Zynq.

Після завантаження можна створити базовий додаток, використовуючи IP, що доступні в директорії /PYNQ/Boards/IP.

У цьому випадку готовий дизайн для плати Zybo Z7 можна побачити нижче, і він включає не тільки IOP, але й GPIO для кнопок, перемикачів, світлодіодів і трьохколірних світлодіодів.

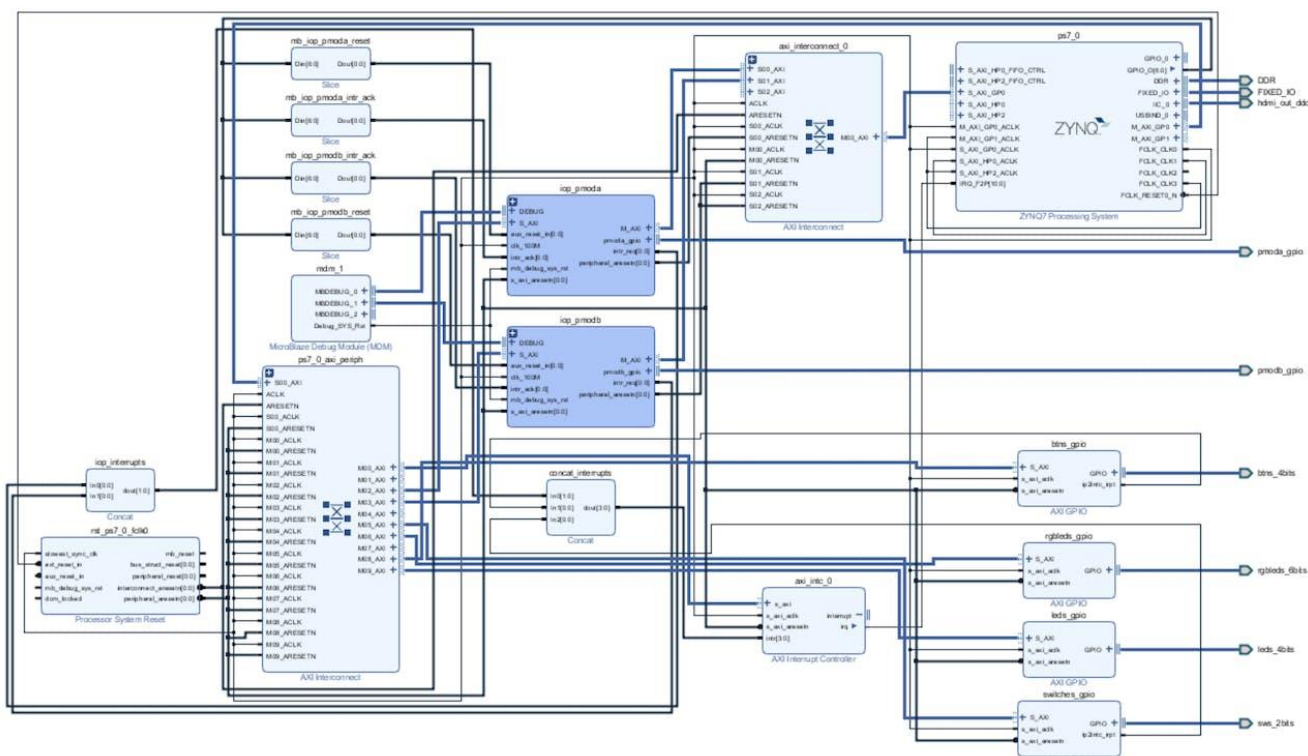


Рисунок 3.3 - Дизайн для плати Zybo Z7

Дуже важливим аспектом цього рішення є використання контролера переривань AXI для каскадування переривань до універсального контролера переривань Zynq.

Після завершення дизайну потрібно побудувати його у Vivado та експортувати, щоб можна було створити образ PetaLinux.

Для створення образу PYNQ спочатку потрібно створити BSP PetaLinux, що означає необхідність створення нового проекту PetaLinux та його налаштування для апаратного забезпечення у Vivado.

Якщо дотепер дизайн Vivado створювався у Windows, то для створення образів PetaLinux та PYNQ необхідна машина на Linux. Для цього використовується віртуальна машина на базі Ubuntu 16.04, на яку встановлено PetaLinux та завантажено git-репозиторій PYNQ.

Маючи експортований HDF з Vivado (включно з бітовим потоком) у машині для збірки на Linux, потрібно виконати наступні команди:

```
petalinux-create—type project—template zynq—name zybo  
petalinux-config—get-hw-description  
petalinux-build
```

Зазначимо, що після імпортування визначення апаратури з'явиться діалогове вікно конфігурації проекту PetaLinux. Оскільки зміни робити не потрібно, вона має вийти з діалогу, не зберігаючи нічого. Процес збірки займе кілька хвилин.

Після того, як образ PetaLinux буде побудований, наступним кроком перед збіркою образу PYNQ є перевірка роботи Linux-образу на апаратурі. Це можна зробити за допомогою наведених нижче команд.

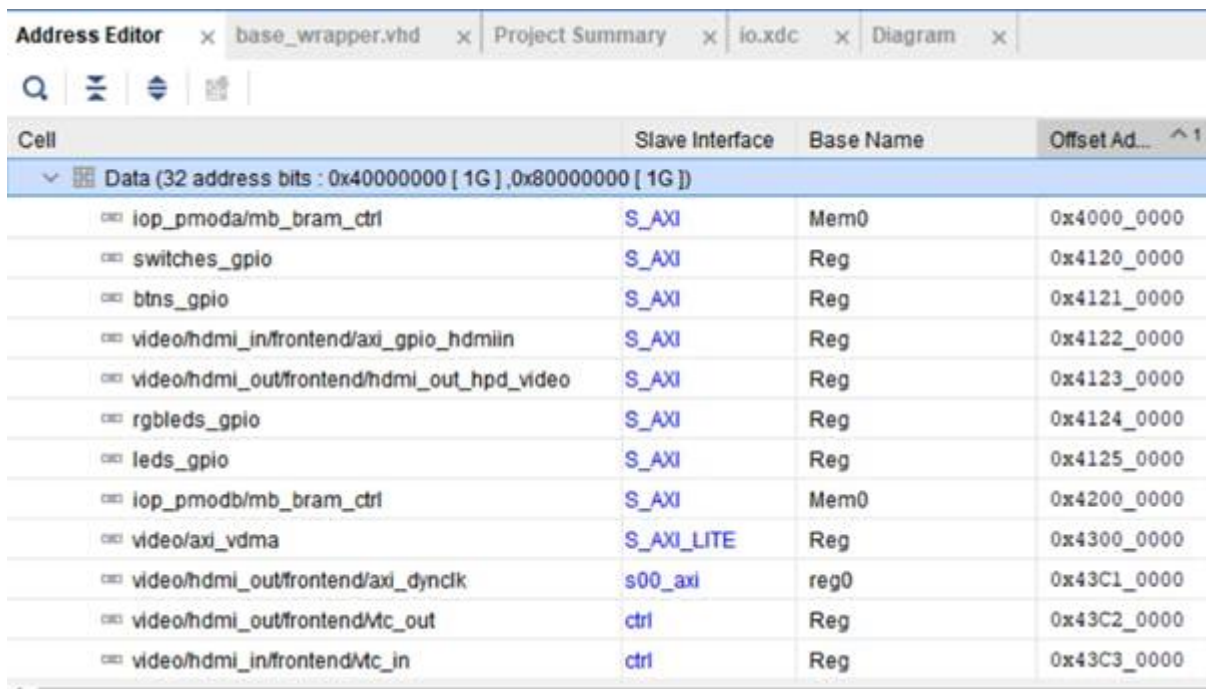
```
petalinux-boot --jtag --fpga  
petalinux-boot --jtag --kernel
```

Це завантажить образ на плату Zybo за допомогою з'єднання JTAG та запустить образ.

Після завершення завантаження за допомогою серійного терміналу потрібно перевірити, чи видно AXI GPIO у дереві пристроїв LINUX. З командного рядка можна виконати команду `dmesg`.

Це виведе повідомлення завантаження, і можна буде прокрутити повідомлення, поки не знайдеться реєстрація GPIO.

Можна порівняти це з картою пам'яті у Vivado, і адреси мають бути однаковими.



Cell	Slave Interface	Base Name	Offset Ad... ^1
Data (32 address bits : 0x40000000 [ 1G ], 0x80000000 [ 1G ])			
iop_pmoda/mb_bram_ctrl	S_AXI	Mem0	0x4000_0000
switches_gpio	S_AXI	Reg	0x4120_0000
btns_gpio	S_AXI	Reg	0x4121_0000
video/hdmi_in/frontend/axi_gpio_hdmiin	S_AXI	Reg	0x4122_0000
video/hdmi_out/frontend/hdmi_out_hpd_video	S_AXI	Reg	0x4123_0000
rgbleds_gpio	S_AXI	Reg	0x4124_0000
leds_gpio	S_AXI	Reg	0x4125_0000
iop_pmodb/mb_bram_ctrl	S_AXI	Mem0	0x4200_0000
video/axi_vdma	S_AXI_LITE	Reg	0x4300_0000
video/hdmi_out/frontend/axi_dynclk	s00_axi	reg0	0x43C1_0000
video/hdmi_out/frontend/Mc_out	ctrl	Reg	0x43C2_0000
video/hdmi_in/frontend/Mc_in	ctrl	Reg	0x43C3_0000

Рисунок 3.4 – Редактор адрес

Також можна підключитися до образу Petalinux через мережу, використовуючи SSH, щоб переконатися, що інтерфейс працює у дизайні. Після цього наступним кроком є підготовка BSP для процесу створення образу PYNQ.

Для цього потрібно спочатку упаковати образ, а потім BSP, використовуючи команди `petalinux-package → boot → fsbl zynq_fsbl.elf → u-boot u-booot.elf → fpga system.bit`.

Це створить файл `boot.bin`, необхідний для завантаження з SD-карти.

Після того, як файли для завантаження стануть доступними, останнім кроком у PetaLinux є створення BSP, для чого використовується команда

```
petalinux-package → bsp -p zybo -o zybo_pynq.bsp
```

Для пакування BSP треба перебувати поза проектом PetaLinux, коли вводимо команду.

З наявністю BSP тепер можна створювати образ PYNQ.

Створення образу PYNQ. На машині для збірки Linux з BSP для Zybo тепер можна побудувати образ PYNQ. Якщо репозиторій PYNQ ще не було клоновано на машину для збірки, то треба це зробити.

Для створення нового образу PYNQ спершу потрібно створити нове визначення плати у папці `pynq/boards`.



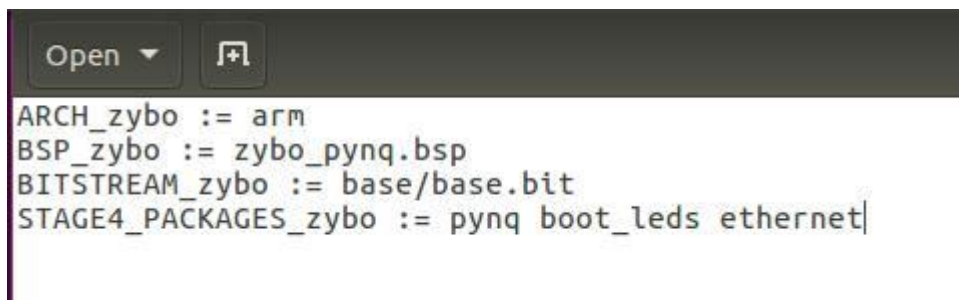
Рисунок 3.5 - Нове визначення плати у папці `pynq/boards`

У папці ZYBO потрібно включити BSP, бітовий файл та визначення плати.



Рисунок 3.6 – Включення бітового файлу

У файлі специфікацій Zybo визначається розташування бітового файлу, BSP та архітектури.



```

Open ▾ [+]
ARCH_zybo := arm
BSP_zybo := zybo_pynq.bsp
BITSTREAM_zybo := base/base.bit
STAGE4_PACKAGES_zybo := pynq boot_leds ethernet

```

Рисунок 3.7 – Файл специфікацій Zybo

Якщо це перший раз, коли запускається машина для збірки PYNQ, потрібно переконатися, що вона налаштована правильно.

Для цього запускається скрипт налаштування хоста з директорії `pynq/sdbuild/scripts` `source setup_host.sh`.

Для побудови образу з директорії `pynq/sdbuild` використовується команда `make BOARDS=zybo`.

Це займе кілька хвилин, але після завершення можна побачити готовий образ у `pynq/sdbuild/output`.

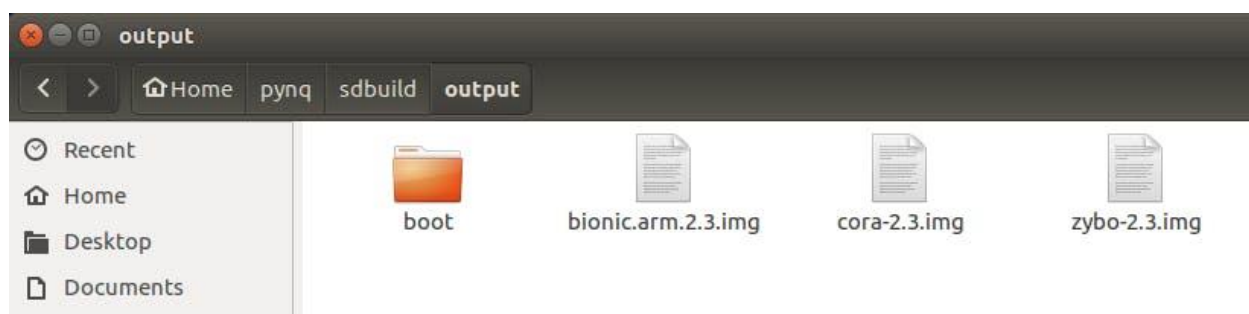


Рисунок 3.8 – Готовий образ

Якщо раніше не використовувалася `adafruit.io`, потрібно створити обліковий запис.

Після створення облікового запису, необхідні для підключення до облікового запису секретні ключі АІО можна знайти у лівій частині під назвою

"View AIO Key". Важливо захищати ці значення від інших, але вони мають бути включені до Python-додатку.

Наступним кроком є створення каналу, до якого Zybo буде відправляти дані. Треба пам'ятати, що назва цього каналу, яку використовуватимуть у додатку Zybo, чутлива до регістру.

У середовищі PYNQ потрібно відкрити термінал і встановити пакети `adafruit.io`, використовуючи команду `pip3 install adafruit.io`.

Після завантаження цих пакетів можна використовувати їх для відправки даних у хмару.

Створений у записнику PYNQ додаток можна побачити в ліст. 3.1. Хоча він дуже простий, він дозволяє реалізувати рішення на основі IoT, використовуючи хмару Adafruit та IFTTT.

Хмара Adafruit є частиною сервісів, які надає Adafruit Industries, і вона спеціалізується на підтримці проектів з електронікою та Інтернетом речей (IoT). Ця платформа надає користувачам інструменти для збору, зберігання та візуалізації даних з IoT-пристроїв, таких як датчики та контролери. Хмара Adafruit забезпечує легкий доступ до управління пристроями через Інтернет, дозволяючи користувачам моніторити та контролювати свої проекти на відстані. Вона ідеально підходить для хобістів, інженерів-електронщиків, студентів та ентузіастів DIY, оскільки забезпечує простий та доступний спосіб інтеграції електронних проектів з хмарними технологіями. Окрім того, хмара Adafruit часто використовується для освітніх та експериментальних цілей, оскільки вона сприяє легкому вивченню та розумінню основ IoT і хмарних технологій.

Хмара IFTTT (If This Then That) є платформою, яка дозволяє автоматизувати взаємодії між різними інтернет-сервісами, програмами та розумними пристроями. Вона працює на принципі "якщо це, то те", де користувачі можуть створювати та використовувати так звані "аплети", які сполучають різні сервіси та пристрої разом, створюючи автоматизовані дії на основі певних умов або подій.

Наприклад, можна налаштувати IFTTT так, щоб автоматично зберігати відповіді з електронної пошти у вигляді заміток у Google Drive, або щоб відправляти сповіщення на смартфон, коли певний аккаунт у Twitter публікує новий твіт. Інші популярні використання IFTTT включають інтеграцію з розумними домашніми пристроями, такими як термостати, світлодіодні лампи, камери безпеки, що дозволяє автоматизувати будинок згідно з ваших потреб та переваг.

IFTTT особливо корисний для створення персоналізованих рішень, що забезпечують ефективну взаємодію між різними цифровими сервісами та пристроями, роблячи повсякденне життя більш зручним і автоматизованим.

Лстинг 3.1 - Цей код читає значення освітленості за допомогою датчика і відправляє ці дані на хмарний сервіс Adafruit IO

```
# Імпорт необхідних бібліотек
from pynq.overlays.base import BaseOverlay
from pynq.lib import Pmod_ALS
from Adafruit_IO import Client
import time

# Завантаження базового оверлею для PYNQ
overlay = BaseOverlay("base.bit")

# Перевірка, чи оверлей успішно завантажений
overlay.is_loaded()

# Створення клієнта Adafruit IO з іменем та ідентифікатором
aio = Client('NAME', 'ID')

try:
    while True:
        # Читання значення освітленості
        illum = my_als.read()
        # Відправлення даних про освітленість до Adafruit IO
        aio.send('illumination', illum)
        # Затримка перед наступним читанням
        time.sleep(5)
except KeyboardInterrupt:
    # Виведення повідомлення при перериванні вручну
    print("CTRL-C!! Exiting...")
```

Код буде зчитувати дані з сенсора ALS кожні 5 секунд і відправляти зібрані дані в хмару Adafruit, як це можна побачити на графіках нижче.

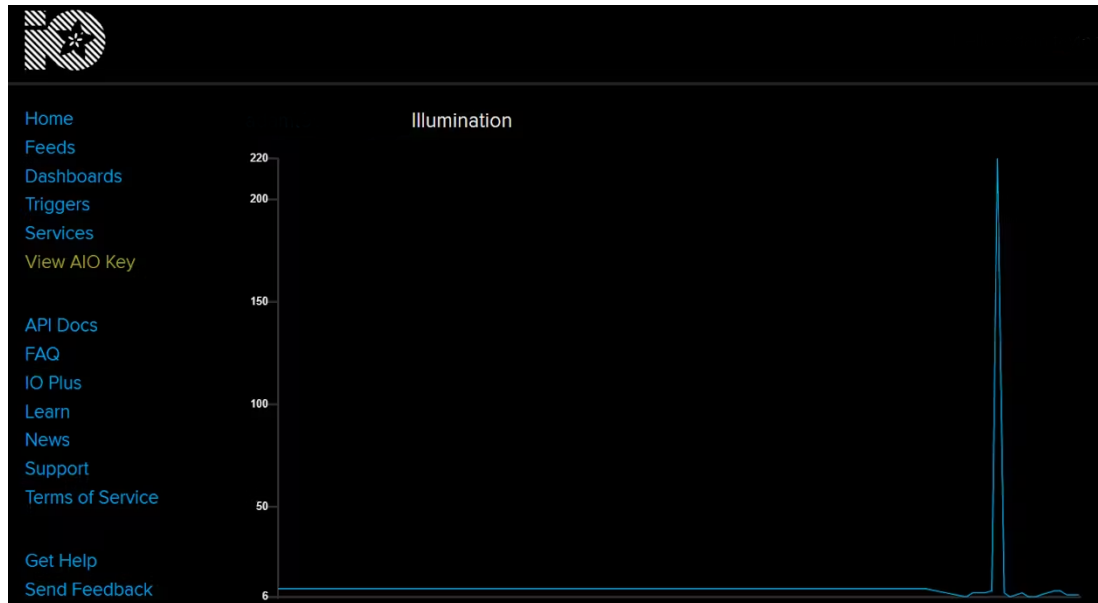


Рисунок 3.9 - Дані в хмарі Adafruit

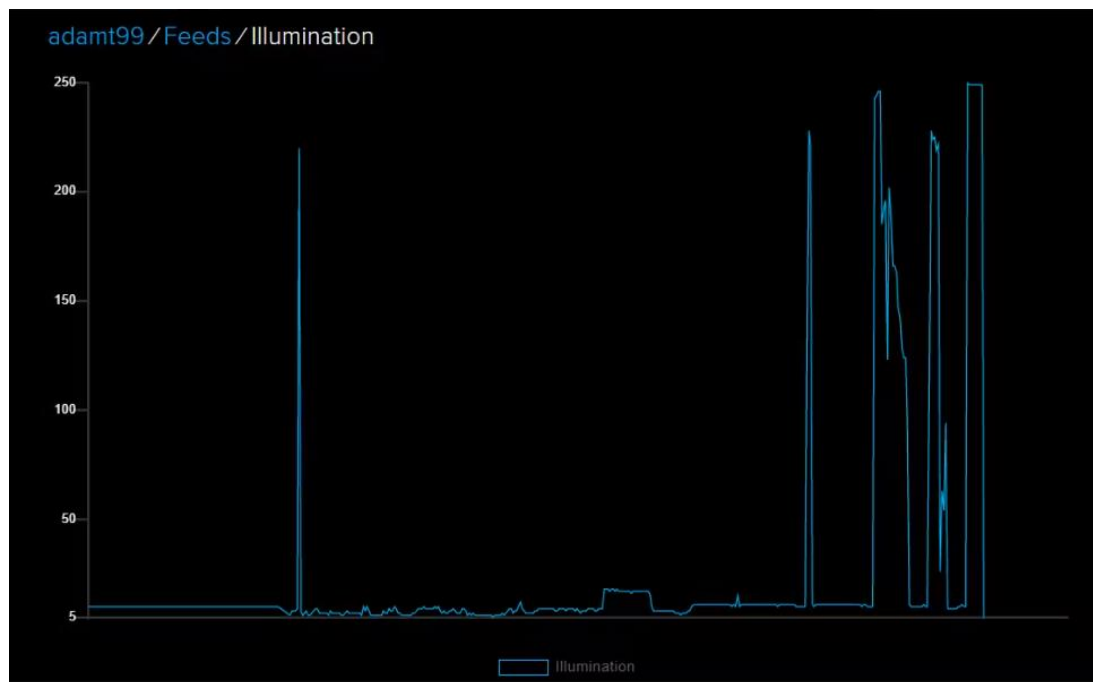


Рисунок 3.10 - Дані в хмарі Adafruit. Освітлення

Щоб спостережуване значення мало кінцевий ефект, використовується IFTTT, який моніторить канал у хмарі [adafruit.io](https://adafruit.io).

Залежно від значення у каналі, світло Phillips Hue може бути ввімкнене або вимкнене.

Philips Hue є системою розумного освітлення, розробленою компанією Philips. Ця система включає в себе широкий спектр світлодіодних ламп, світильників, смуг освітлення та аксесуарів, які можуть бути керовані дистанційно через додаток на смартфоні, планшеті або за допомогою голосових команд через сумісних голосових помічників, таких як Amazon Alexa, Google Assistant та Apple HomeKit.

Для того, щоб згенерувати IFTTT потрібен обліковий запис IFTTT, а після створення можна створити новий аплет.

Щоб використовувати IFTTT з каналами у обліковому записі Adafruit IO, потрібно включити інтеграцію IFTTT, щоб IFTTT міг отримувати доступ до каналу. Це можна активувати в профілі IO Adafruit, можливо, спочатку потрібно створити обліковий запис IFTTT, щоб підключитися до нього.

Це шестиступеневий процес, перше, що потрібно перевірити, - це сервіс, у цьому випадку Adafruit.

Після вибору сервісу Adafruit потрібно вибрати канал, за яким буде проводитись моніторинг, та визначити умову, наприклад, рівне, більше за, менше за тощо.

Після визначення умови потрібно вибрати сервіс для дії, у цьому випадку це Phillips Hue. Щоб використовувати Phillips Hue з нашим додатком, необхідно авторизувати використання IFTTT в додатку управління Phillips Hue.

Тепер можна вибрати бажану дію, наприклад, ввімкнення або вимкнення обраного світла. Це дозволяє нам визначити два аплети: один для ввімкнення світла, якщо рівень ALS нижче заданого значення, інший - для вимкнення світла, якщо він вище заданого значення.

**Monitor a feed on Adafuit IO**

This Trigger fires anytime it validates the data that you send to your feed. Example: If Feed Temperature > 80, fire Trigger.

**Feed**

test

The name of the feed to check.

**Relationship**

less than

Relationship between two values.

**Value**

The value to compare against.

Create trigger

Рисунок 3.11 – Інтерфейс Adafuit IO

Після визначення умови потрібно вибрати сервіс для дії, у цьому випадку це Phillips Hue. Щоб використовувати Phillips Hue з нашим додатком, необхідно авторизувати використання IFTTT в додатку управління Phillips Hue.

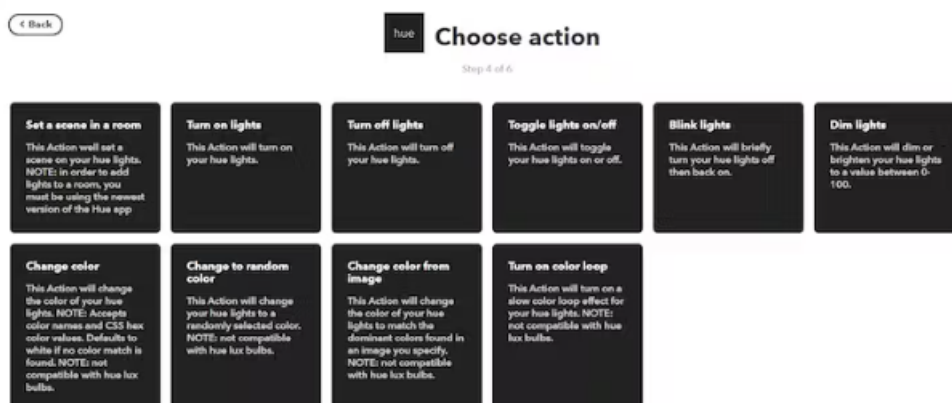


Рисунок 3.12 – Вибір бажаної дії

Тепер можна вибрати бажану дію, наприклад, ввімкнення або вимкнення обраного світла. Це дозволяє нам визначити два аплети: один для ввімкнення

світла, якщо рівень ALS нижче заданого значення, інший - для вимкнення світла, якщо він вище заданого значення.

Після визначення, чи потрібно ввімкнути або вимкнути світло, можна вибрати конкретне джерело, групу джерел або всі джерела.

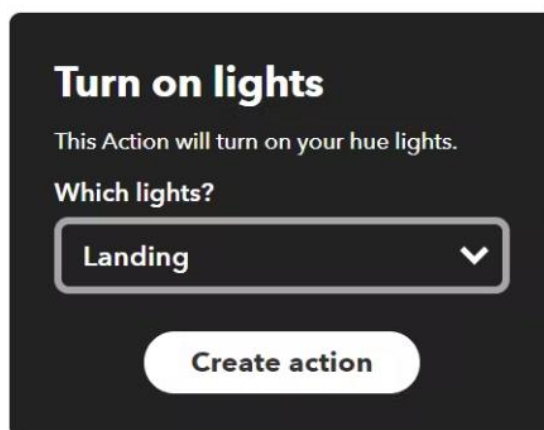


Рисунок 3.13 – Налаштування Adafruit IO

Після цього вибору завершується створення аплету.

Запускаючи цей аплет, можна побачити, як світло вмикається та вимикається залежно від освітленості, якій піддається датчик ALS. Для зміни значення ALS використовувалася функція ліхтарика, вбудована в мобільний телефон.

### 3.2 Висновки до розділу 3

Цей розділ описує розробку системи, яка використовує образ PYNQ для розробницької плати Zybo з метою легкого підключення до сенсорів та хмарних сервісів IoT. У проєкті використовується плата ZYBO Z7, обрана за наявності шести інтерфейсів Pmod для підключення різноманітних сенсорів. Перший крок

полягає у створенні образу PYNQ, починаючи з роботи у Vivado для реалізації базового дизайну апаратного забезпечення. Фреймворк PYNQ використовує процесори вводу-виводу (IOP) у програмованій логіці для підтримки інтерфейсів Pmod, переносячи завдання нижчого рівня, що вимагають реального часу, на програмовану логіку.

У розділі детально описано процес створення базового додатку за допомогою Vivado та PetaLinux, включаючи клонування директорії PYNQ, створення нового проекту PetaLinux, налаштування для апаратного забезпечення у Vivado, а також необхідність використання Linux для створення образів PetaLinux та PYNQ. Описано процес збірки, експорту, тестування та відправлення даних на хмарний сервіс Adafruit IO.

Також розглядається інтеграція з хмарою Adafruit, яка дозволяє легко моніторити та контролювати IoT-проекти на відстані, та IFTTT для автоматизації взаємодії між різними сервісами. Зокрема, розроблено застосування для керування світлом Phillips Hue через IFTTT, залежно від даних, отриманих від датчика ALS.

## ВИСНОВКИ

У роботі було проведено дослідження та розробку систем віддаленого моніторингу з використанням FPGA (Field-Programmable Gate Array). Головною метою було розробити та проаналізувати ефективні та надійні рішення для оптимізації обробки даних і забезпечення надійності та ефективності цих систем.

В першому розділі проведено опис сучасних рішень в області віддаленого моніторингу. Виявлено, що сучасні дослідження визнають FPGA як потужний інструмент для віддаленого моніторингу та пророчать їхню успішну інтеграцію в різні сфери застосування, сприяючи вдосконаленню систем моніторингу та керування у реальному часі. Нові технології та рішення, які виникають на базі FPGA, відкривають широкі перспективи для покращення продуктивності та ефективності систем віддаленого моніторингу.

В другому розділі проведено дослідження апаратних платформ, які можуть використовуватися для реалізації системи віддаленого моніторингу на FPGA. Для побудови системи обрано плату Zybo Z7, яка є багатофункціональною, готовою до використання платою для розробки вбудованого програмного забезпечення та цифрових схем, побудованою на базі сімейства Xilinx Zynq-7000. Zybo Z7 володіє багатим набором мультимедійних та комунікаційних периферій, створюючи потужний одноплатний комп'ютер.

В якості периферійного датчика освітлення обрано Digilent PmodALS, який комунікує з основною платою через SPI.

В третьому розділі описано розробку системи, яка використовує образ PYNQ для розробницької плати Zybo з метою легкого підключення до сенсорів та хмарних сервісів IoT. У розділі детально описано процес створення базового додатку за допомогою Vivado. Також розглядається інтеграція з хмарою Adafruit, яка дозволяє легко моніторити та контролювати IoT-проекти на відстані, та IFTTT для автоматизації взаємодії між різними сервісами. Зокрема, розроблено застосування для керування світлом Phillips Hue через IFTTT, залежно від даних, отриманих від датчика ALS.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Слюсар В. І. Концепція архітектури транспортних засобів як мережі мереж.//Збірник матеріалів XII науково-практичної конференції «Пріоритетні напрямки розвитку телекомунікаційних систем та мереж спеціального призначення. Застосування підрозділів, комплексів, засобів зв'язку та автоматизації в операції Об'єднаних сил» (14 — 15 листопада 2019 р.). — Київ. — С. 218—219
2. Інтернет речей та де він застосовується <https://asapdemo.com/bez-rubriki/internet-rechej-ta-rishennya-dlya-jogo-realizatsiyi/> (дата звернення: 05.12.2023)
3. Інтернет речей у сільському господарстві: 8 варіантів використання технологій для розумного землеробства <https://globalx-ua.com/internet-veschey-v-selskom-hozyaustve> (дата звернення: 05.12.2023)
4. Abramuić B. et al. Home video monitoring system for neurodegenerative diseases based on commercial HD cameras // Consumer electronics—Berlin (ICCE-Berlin), 2015 IEEE 5th international conference. – 2015. – С. 489–492.
5. Bobojonov I., Aw-Hassan A. Impacts of climate change on farm income security in Central Asia: An integrated modeling approach [Abstract] // J. Agriculture, Ecosystems & Environment. – 2014. – Vol. 188. – P. 245-255. DOI: 10.1016/j.agee.2014.02.033.
6. Rad J.C., Hancu J.O., Jakacs I., Olteanu J. Smart Monitoring of Potato Crop: A Cyber-Physical System Architecture Model in the Field of Precision Agriculture // J. Agriculture and Agricultural Science Procedia. – 2015. – Vol. 6. – P. 73-79. DOI: 10.1016/j.aspro.2015.08.041.
7. Shobri N.I., Sakip S.R., Omar S.J. Malaysian Standards Crop Commodities In Agricultural for Sustainable Living // Procedia Social and Behavioral Sciences. – 2016. – Vol. 222. – P. 485-492. DOI: 10.1016/j.sbspro.2016.05.139.
8. Adams Z.W., McClure E.A., Gray K.M., Danielson C.K., Treiber F.A., Ruggiero K.J. Mobile devices for the remote acquisition of physiological and behavioral

biomarkers in psychiatric clinical research // J Psychiatr Res. – 2017. – Vol. 85. – P. 1–14. – DOI: 10.1016/j.jpsychires.2016.10.019.

9. Li L., Chu X. M., Wu Y., Wu Q. The Development of Road Lighting Intelligent Control System Based on Wireless Network Control // Proceedings of the International Conference on Electronic Computer Technology. – Macau, 20-22 February 2009. – P. 353-357.

10. Aldaz G. et al. Hands-free image capture, data tagging and transfer using google glass: a pilot study for improved wound care management // PLoS One. – 2015. – Vol. 10, Issue 4. – P. 1–21. – DOI: 10.1371/journal.pone.0121179.

11. Xilinx. Zynq 7000 SoC Product Advantages [Электронный ресурс]. – URL: <https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html> (дата звернення: 05.12.2023).

12. Xilinx. PYNQ- Python Productivity for Zynq [Электронный ресурс]. – URL: <https://www.xilinx.com/support/university/xup-boards/XUPPYNQ-Z2.html> (дата звернення: 05.12.2023).

13. Digilent. Basys 3 Artix-7 FPGA Trainer Board: Recommended for Introductory Users [Электронный ресурс]. – URL: <https://digilent.com/shop/basys-3-artix-7-fpga-trainer-board-recommended-for-introductory-users/> (дата звернення: 05.12.2023).

14. Digilent. Arty A7 [Электронный ресурс]. – URL: <https://digilent.com/reference/programmable-logic/arty-a7/start> (дата звернення: 05.12.2023).

15. Terasic. DE10-Nano Kit [Электронный ресурс]. – URL: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=1046> (дата звернення: 05.12.2023).

16. Digilent. Zybo Z7 [Электронный ресурс]. – URL: <https://digilent.com/reference/programmable-logic/zybo-z7/start> (дата звернення: 05.12.2023).

17. SparkFun. Ambient Light Sensor Breakout [Электронный ресурс]. – URL: <https://www.sparkfun.com/products/8688> (дата звернення: 05.12.2023).
18. Raspberry. Adafruit TSL2591 High Dynamic Range Digital Light Sensor [Электронный ресурс]. – URL: <https://raspberry.com.ua/p/adafruit-tsl2591-high-dynamic-range-digital-light-sensor/> (дата звернення: 05.12.2023).
19. Seeed Studio. Grove - Light Sensor v1.2 - LS06-S Phototransistor Compatible with Arduino [Электронный ресурс]. – URL: <https://www.seeedstudio.com/Grove-Light-Sensor-v1-2-LS06-S-phototransistor.html> (дата звернення: 05.12.2023).
20. Seeed Studio. Grove - Luminance Sensor [Электронный ресурс]. – URL: [https://wiki.seeedstudio.com/Grove-Luminance\\_Sensor/](https://wiki.seeedstudio.com/Grove-Luminance_Sensor/) (дата звернення: 05.12.2023).
21. Digilent. PmodALS Reference Manual [Электронный ресурс]. – URL: <https://digilent.com/reference/pmod/pmodals/reference-manual> (дата звернення: 05.12.2023).
22. Xilinx. ISE Design Suite [Электронный ресурс]. – URL: <https://www.xilinx.com/products/design-tools/ise-design-suite.html> (дата звернення: 05.12.2023).
23. Philips Hue. Philips Hue and IFTTT [Электронный ресурс]. – URL: <https://www.philips-hue.com/en-us/explore-hue/works-with/ifttt> (дата звернення: 05.12.2023).
24. PYNQ. What is PYNQ? [Электронный ресурс]. – URL: <http://www.pynq.io/> (дата звернення: 05.12.2023).