Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій та технічного захисту інформацій

Кафедра Радіотехнологій інформаційно-комунікаційних систем

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

ГЮІК.509ХХХ.001 ПЗ

(позначення документа)

Програмно-апаратний комутатор інтерфейсів вбудованих систем

(тема)

Виконав:
студент II курсу, групи ІКТм -18-1
Шевцов І. О.
(прізвище, ініціали)

Спеціальність
122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми
освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Інформаційно-комунікаційні технології
(повна назва освітньої програми)

Керівник доцент Бітченко О.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри РТІКС _____ Цопа О.І.
(підпис) (прізвище, ініціали)

2019 р.

Не містить відомостей заборонених для відкритого публікування.

Студент І.О. Шевцов

Керівник О.М. Бітченко

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Інформаційних радіотехнологій та технічного захисту інформацій
Кафедра Радіотехнологій інформаційно-комунікаційних систем
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
Тип програми Освітньо-професійна
Освітня програма Інформаційно-комунікаційні технології

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2019 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові **ШЕВЦОВУ Івану Олександровичу**
(прізвище, ім'я, по батькові)

1. Тема роботи **ПРОГРАМНО-АПАРАТНИЙ КОМУТАТОР ІНТЕРФЕЙСІВ ВБУДОВАНИХ СИСТЕМ**

затверджена наказом по університету від **21 листопада 2019 р. № 1730Ст**

2. Термін подання студентом проекту (роботи) **12 грудня 2019 р.**

3. Вихідні дані до проекту (роботи)
3.1 Програмно-апаратний комутатор передає повідомлення від контролера верхнього рівня до периферійних пристроїв і навпаки.
3.2 Програмно-апаратний комутатор підтримує роботу з інтерфейсами периферійних пристроїв UART, SPI, I2C.

4. Перелік питань, що потрібно опрацювати в роботі
Реферат. Перелік умовних позначень, символів, одиниць, скорочень і термінів. Вступ. 4.1 Огляд та аналіз аналогічних рішень. 4.2 Розробка концепції пристрою зв'язку. . 4.3 Вибір відлагоджувальної плати. 4.4 Вибір середовища розробки пристрою. 4.5 Розробка програмного забезпечення комутатора периферійних інтерфейсів. Висновки. Перелік посилань. Додатки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

Комп'ютерна презентація

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по-батькові	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Бітченко Олександр Миколайович		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вступ	5.09-10.09	Виконано
2	Огляд та аналіз аналогічних рішень	10.09-20.09	Виконано
3	Розробка концепції пристрою зв'язку	20.09-10.10	Виконано
4	Вибір відлагоджувальної плати	10.10-14.10	Виконано
5	Вибір середовища розробки пристрою	14.10-24.10	Виконано
6	Розробка програмного забезпечення комутатора периферійних інтерфейсів	24.10-24.11	Виконано
9	Висновки	24.11-.27.11	Виконано
10	Оформлення пояснювальної записки	27.11-8.12	Виконано
11	Оформлення презентації	8.12-11.12	Виконано
12	Подання роботи на кафедру	12.12.2019	Виконано

Дата видачі завдання **4 вересня 2019 р**

Студент _____
(підпис)

Керівник роботи_____ ___доц. Бітченко О.М.____

РЕФЕРАТ

Магістерська атестаційна робота складається з пояснювальної записки, котра містить: XXX сторінок тексту, XX рисунків, XX джерело та X додатки.

IoT, АВТОМАТИЗАЦІЯ, ЗВ'ЯЗОК З ПЕРИФЕРІЙНИМИ ПРИСТРОЯМИ, КОМУТАРОР, ПРИСТРІЙ СПОЛУЧЕННЯ

Об'єкт роботи – програмне забезпеченя комутатора інтерфейсів периферійніх пристроів.

Мета роботи – вибір апаратної платформи та розробка програмного забезпечення комутатора інтерфейсів периферійніх пристроів.

Для досягнення поставленої мети проаналізовані аналогічні рішення, на їх основі сформовану концепцію розроблюваного пристрою. Вибрано апаратну платформу та середовище розробки, обґрунтовано використання апаратних і програмних засобів, розроблено програмне забезпечення комутатора інтерфейсів периферійних пристроїв.

THE ABSTRACT

The master's degree work consists of an explanatory note, which contains: XXX pages of text, XX figures, XX sources and X appendices.

IoT, AUTOMATION, COMMUNICATION WITH PERIPHERAL DEVICES, SWITCH, COMMUNICATION DEVICES

Object of work – the software of the switch of interfaces of peripheral devices.

The purpose of the work is to select the hardware platform and to develop the switch software for the peripheral device interfaces.

To achieve this goal, similar solutions are analyzed, based on them the concept of the developed device is formed. Hardware platform and development environment are selected, hardware and software usage are justified, and peripheral interface switch software has been developed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – прикладний програмний інтерфейс;

TCP/IP – мережева модель передачі цифрових даних;

АЦП – аналогово-цифровий перетворювач;

ЦАП – цифро-аналоговий перетворювач;

ПЗП – постійний запам'ятовувальний пристрій;

ОЗП – оперативно запам'ятовувальний пристрій;

ППЗП – програмуємий постійний запам'ятовувальний пристрій;

ФАПЧ – фазове автопідлаштування частоти;

ПЛК – програмований логічний контролер;

SoC – System on Chip –система на чіпі;

IoT – Internet of Things – інтернет речей;

I2C – Inter-Integrated Circuit – шина даних зв’язку інтегральних схем;

SPI – Serial Peripheral Interface – послідовний периферійний інтерфейс;

CAN – Controller Area Network – мережа контролерів;

UART – Universal Asynchronous Receiver Transmitter – універсальний асинхронний приймач-передавач;

USART – Universal Synchronous Asynchronous Receiver Transmitter – універсальний синхронно-асинхронний приймач-передавач;

OSI – Open Systems Interconnection – відкрита система взаємозв’язку;

DSP – Digital Signal Processor – цифровий сигнальний процесор;

MPSoC – Multiprocessor System-on-Chip – багатопроцесорна система на чіпі;

CRC – Cyclic redundancy check – циклічний надлишковий код;

USB – Universal Serial Bus – універсальна послідовна шина;

DRAM – Dynamic Random Access Memory – динамічна пам'ять з довільним доступом;

МК – мікроконтролер;

ОСРЧ – операційна система реального часу.

ВСТУП

На сьогоднішній день ми спостерігаємо потужний спалах інтересу до інтернету речей саме в останні кілька років, концепція технології існує з 1999 року, і вже тоді відчувалася масштабність її характеру. Агрегування даних з підключених пристроїв і датчиків допомагає оптимізувати бізнес-процеси, і отримувати більш персоналізовані і якісні послуги/інфраструктуру споживачам.

Ефективний розвиток IoT зводиться не тільки до проникнення «підключених» пристроїв в усі аспекти життя, але до створення технологічної екосистеми, об'єднаних рішень для збору, передачі, агрегації даних на платформі, що дозволяє обробити дані і використовувати їх для реалізації ефективних рішень. Наявність великої кількості недорогих датчиків і периферійних пристроїв, що мають різні інтерфейси і протоколи сполучення, для роботи з контролерами IoT. Стандартизовані інтерфейсів і протоколів зв'язку більшості контролерів автоматизації накладає обмеження на використання безлічі пристроїв. При цьому, часто немає можливість використовувати перетворювачі інтерфейсів, для того щоб підключити не підтримуваний пристрій до контролера автоматизації [1].

Метою даної кваліфікаційної роботи є програмна розробка – програмне забезпечення комутатора інтерфейсів вбудованих систем.

Для того щоб досягнути поставленої мети сформулюємо наступні задачі:

- проаналізувати аналогічні рішення які існують на даний час і виконують ті ж самі, або подібні функції;
- проаналізувати основні принципи побудування комутатора інтерфейсів вбудованих систем;
- вибрати апаратну платформу для реалізації програмного забезпечення комутатора інтерфейсів вбудованих систем;
- вибрати середовище розробки для проектування програмного забезпечення вбудованих систем;

- на основні проаналізованих принципів побудування комутатора інтерфейсів вбудованих систем розробити структурну схему програмного забезпечення, а також протоколи зв’язку
- відповідно до обраної структурної схеми розробити програмне забезпечення комутатора інтерфейсів вбудованих систем;
- перевірити роботу комутатора інтерфейсів.

1 ОГЛЯД ТА АНАЛІЗ АНАЛОГІЧНИХ РІШЕНЬ

1.1 Огляд способів вирішення завдання і критерії їх порівняння

Необхідність зв'язку з великою кількістю периферійних пристроїв (датчиками, реле, засобами обліку електричної енергії і т.д.) при розробці систем автоматизації виникає досить часто і вирішується різними способами:

- універсальні контролери автоматизації;
- ПЛК контролери;
- реалізація рішень на основі збірки декількох пристроїв, кожне з яких реалізує набір функцій, в одну мережу, зв'язок з якою здійснюється з використанням певних протоколів, наприклад. ModBus, ProfiBus, TCP/IP і інші.

Критерії порівняння обраних способів:

- функціональність – наявність більшості часто використовуваних інтерфейсів і протоколів для організації зв'язку з великою кількістю периферійних пристроїв. Також наявність додаткового функціоналу дозволяє описувати власні сценарії для обробки інформації і подій.
- модульність (розширюваність) – критерій визначає можливість приєднання додаткових модулів розширення для збільшення кількості інтерфейсів, або дискретних/аналогових портів введення/виведення, або розширення функціоналу;
- простота введення в експлуатацію – цей критерій визначає складність налаштування пристрою для першого запуску. Також визначається складність розширення системи та додавання нового функціоналу;
- простота експлуатації – критерій визначає складність використання системи в повсякденному житті.

1.2 СПК1хх сенсорні панельні контролери з Ethernet

Сенсорні панельні контролери СПК1хх з Ethernet є розвитком лінійки СПК1хх. Покращені технічні характеристики, розширений набір інтерфейсів і оновлене програмне забезпечення дозволяють використовувати їх для вирішення широкого спектра завдань автоматизації в різних галузях промисловості.

Контролер позиціонується як рішення для автоматизації будь-якої складності, а також в якості центрального контролера для «розумного будинку» і є провідним контролером компанії WirenBoard.

Схема застосування СПК1хх приведена на рисунку 1.1.

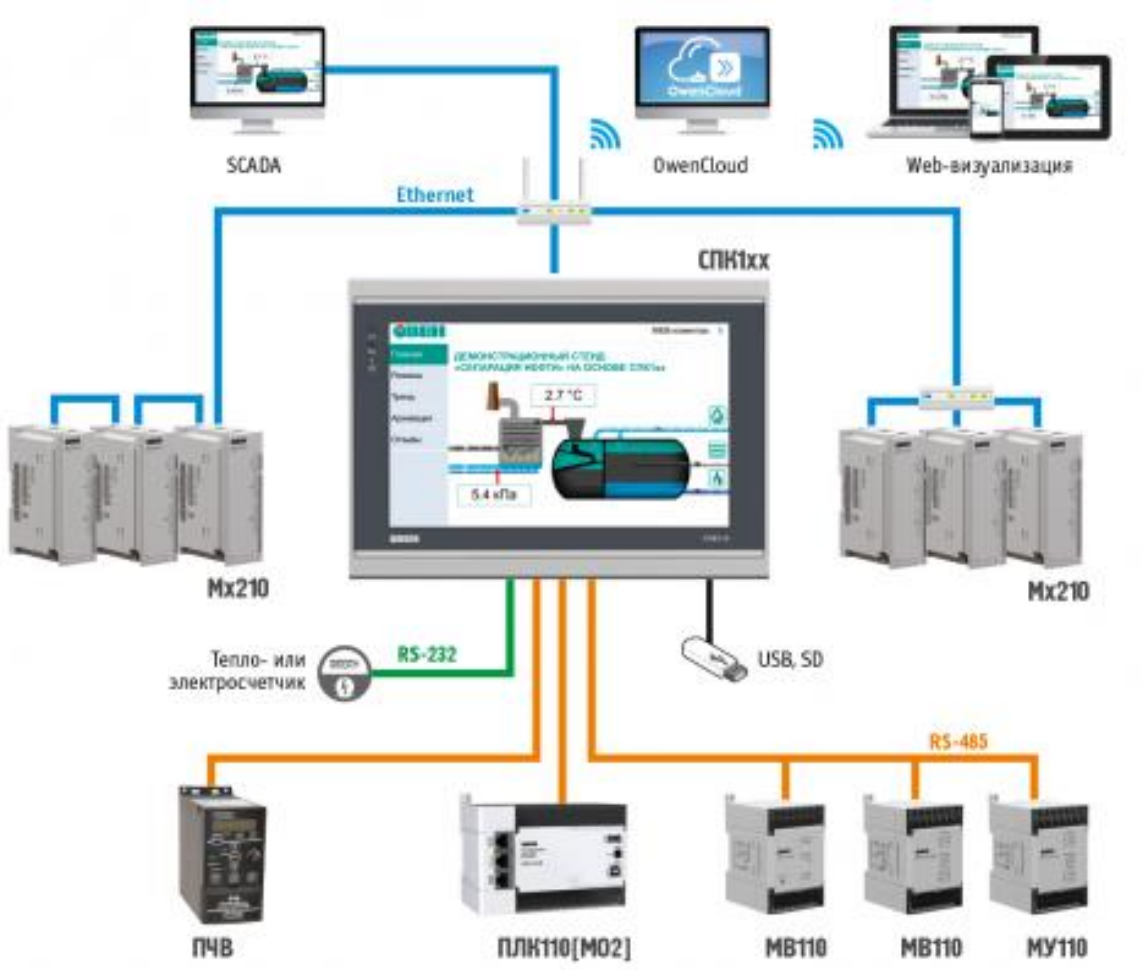


Рисунок 1.1 –Схема применения СПК1хх

Функціональна схема СПК1хх приведена на рисунку 1.2.

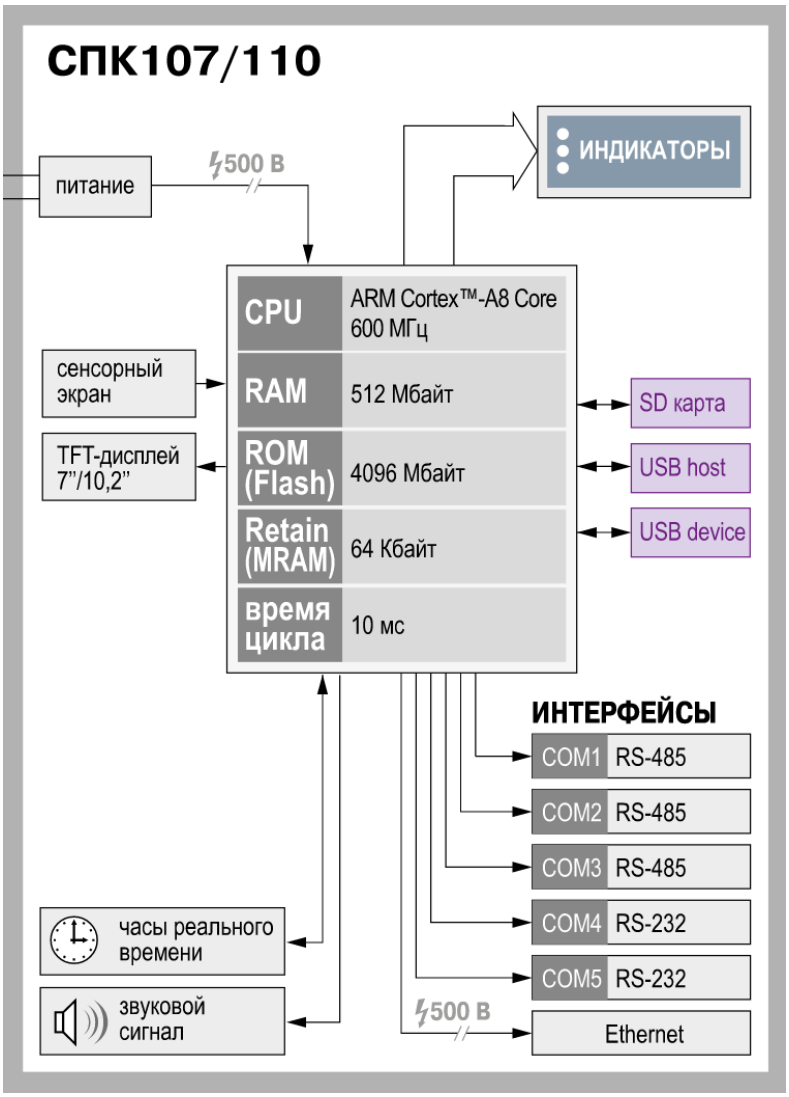


Рисунок 1.2 –Функциональная схема СПК1хх

Сенсорні панельні контролери серії СПК1хх призначені для вирішення завдань промислової автоматизації та можуть використовуватися в якості локальних АРМ. Пристрій розроблений на базі процесора TexasInstrument Sitara 600 MHz ARM Cortex ™ -A8 Core і використовує ОС Linux [2].

Переваги використання:

- розробка алгоритмів керування і візуалізації в єдиному середовищі програмування;
- економія монтажного простору в шафі автоматики;
- зниження загальної вартості системи управління.

Відмінні особливості:

- об'єднання функцій програмованого контролера і панелі оператора в одному корпусі (ПЛК + HMI);
- сенсорний резистивний дисплей 7 "або 10.2" (800 × 480);
- широкий набір комунікаційних інтерфейсів: Ethernet, 3 × RS-485, 2 × RS-232, USB Host, USB Device, слот для SD-карт;
- підтримка протоколів обміну Modbus (RTU, ASCII, TCP), ОВЕН, можливість реалізації нестандартних протоколів;
- можливість поновлення проектів і вбудованого ПО (прошивки) з USB- і SD-накопичувачів;
- підтримка web-візуалізації;
- інтеграція з хмарним сервісом OwenCloud;
- вбудована операційна система Linux;
- гнучко настроюється сторожовий таймер (WatchDog);
- підтримка протоколів NTP, FTP;
- програмування в середовищі CODESYS V3.5 SP11 Patch 5;
- розширення кількості точок введення/виведення здійснюється шляхом підключення зовнішніх модулів введення/виведення за допомогою одного з вбудованих інтерфейсів.

Програмування контролерів здійснюється в професійній, поширеною середовищі CODESYS V3.5, максимально відповідає стандарту МЕК 61131-3:

- підтримка 5 мов програмування, що підходять для фахівців будь-якої галузі;
- інтегрований редактор візуалізації і конфігуратор обміну;
- просунуті засоби налагодження і емуляції;
- безліч бібліотек, що спрощують розробку проекту;
- Відсутність явних обмежень на число використовуваних змінних, блоків і т.д.

1.3. Сенсорні панельні контролери СПК207 для автоматизації розподілених систем [3].

Загальний вигляд панелі контролера наведено на рисунку 1.3.

Рисунок 1.3 – Загальний вигляд сенсорної панелі контролера СПК207

Відмінні особливості:

- оновлена модифікація з підвищеною продуктивністю, процесор 600МГц;
- об'єднання функцій ПЛК та графічної панелі оператора в одному корпусі;
- розробка програм візуалізації і алгоритмів керування в єдиному середовищі програмування;
- сенсорний екран управління;
- додаткові кнопки управління зі світлодіодною індикацією;
- програмне перемикання режимів роботи універсальних інтерфейсів RS-232/RS-485;
- індикація стану обміну по послідовним інтерфейсів на лицьовій панелі;
- вбудований інтерфейс Ethernet;
- вбудована операційна система Linux;
- повномодемний порт RS-232;

- підтримка протоколів обміну Modbus (RTU, ASCII, TCP), ОВЕН;
- можливість роботи безпосередньо з портами контролера для підключення нестандартних пристроїв;
- контролер має вбудований годинник, для створення систем управління з урахуванням реального часу;
- наявність великого обсягу Flash пам'яті, з можливістю розширення на SD-карті для архівування даних;
- розширення кількості точок введення/виведення здійснюється шляхом підключення зовнішніх модулів введення/виведення за допомогою одного з вбудованих інтерфейсів.

Рекомендується до використання:

- в системах HVAC В сфері ЖКГ. В АСК водоканалів;
- для управління кліматичним устаткуванням;
- у сфері виробництва будівельних матеріалів;
- на транспорті.

Оптимально для побудови розподілених систем управління і диспетчеризації з використанням як провідних, так і бездротових технологій. Програмування контролерів здійснюється в професійній, поширеною середовищі CoDeSys v.3.5, максимально відповідає стандарту МЕК 61131:

- підтримка 5 мов програмування, для фахівців будь-якої галузі;
- потужний засіб розробки і налагодження комплексних проектів автоматизації на базі контролерів;
- функції документування проектів;
- кількість логічних операцій обмежується тільки кількістю вільної пам'яті контролера;
- практично необмежену кількість використовуваних в проекті лічильників, тригерів, генераторів.

1.4 Універсальні контролери автоматизації компанії WirenBoard

Компанія WirenBoard займається розробкою засобів домашньої і промислової автоматизації. Пристрої компанії використовуються для рішень автоматизації в нафтовидобувній промисловості, банківських відділень, житлових районів, супермаркетах і офісах. Рішення активно застосовуються в задачах моніторингу стану обладнання, збору даних з датчиків, засобів обліку електроенергії, в якості контролерів для «розумного будинку»

Компанія розробляє тільки контролери автоматизації та модулі розширення до них, однак, не рішення «під ключ».

Компанія WirenBoard широко проводить виставки свого обладнання. Також на сайті компанії встановлений демо-кейс з універсальним контролером автоматизації Wiren Board 6, який дозволяє попрацювати з контролером до придбання вироби [4].

Розглянемо універсальний контролер для автоматизації Wiren Board 6, функціональна схема якого приведена на рисунку 1.4.

Рисунок 1.4 – Функціональна схема універсальногоконтролера Wiren Board 6

Контролер позиціонується як рішення для автоматизації будь-якої складності, а також в якості центрального контролера для «розумного будинку» і є провідним контролером компанії WirenBoard.

У контролері встановлений процесор Freescale iMX6ULL 800 MHz Cortex A7 з ОС Linux Debian. Основний функціонал реалізується користувальницький додаток, підтримуються призначені для користувача сценарії, для розширення функціональності. Також підтримується зберігання архіву і веб-інтерфейс для налаштування пристрою. У мінімальній комплектації вже присутні наступні інтерфейси: RS-485, CAN, 1-Wire, Ethernet, USB, GSM/GPRS, а також дискретні/аналогові входи/виходи [4].

1. Функціональність. Контролер має такою базовою функціональністю [4]:

- порти RS-485, CAN, 1-Wire/дискретні входи;
- дискретні/аналогові входи/виходи;
- два порти Ethernet;
- два порти USB;
- модель стільникового зв'язку (2G/3G);
- Wi-Fi;
- Bluetooth.

Програмне забезпечення Wiren Board 6 забезпечує наступну функціональність:

- відкрите розширюване ПО. Вихідний код ПО поширюється під ліцензією MIT, і викладений на GitHub. Архітектура ПО побудована з використанням MQTT протоколу і підтримує гнучке додавання нового функціоналу;
- призначені для користувача сценарії. Можливе створення призначених для користувача сценаріїв на мові програмування JavaScript, або засобами візуального програмування Node-RED. Використовуються для обробки подій периферії;
- зберігання архіву;

– веб-інтерфейс, для настройки та управління пристроєм.

2. Модульність контролера забезпечується наявністю трьох слотів для установки модулів розширення, а також можливістю написання призначених для користувача сценаріїв і модифікацією вихідного коду ПЗ.

Внутрішні модулі розширення – це невеликі плати, що встановлюються всередину корпусу Wiren Board 6 і розширюють його функціональність: додаткові порти RS-485, RS-232, релейні виходи і т. Д.

У контролері є три слота для підключення модулів розширення двох різних типів. Для двох з цих модулів виведено по 3 зовнішніх підключення для кожного.

Модулі вводу-виводу стикуються з боковим роз'ємом на WB6 з правого боку; кожен модуль додає до контролера від 8 до 16 цифрових або аналогових портів. Послідовно можна підключати до 8 модулів: до 4 модулів введення і до 4-х модуля виводу.

Модуль резервного живлення – додаткова мезоніном плата, встановлюється другим поверхом в корпус на DIN-рейку. Містить Li-Ion (Li-Pol) акумулятором ємністю 2200 mAh, забезпечує роботу контролера до 3 годин.

3. Простота введення в експлуатацію – введення пристрою в експлуатацію вимагає знання області, в якій буде застосовуватися пристрій. Наявність великої кількості модулів розширення і роз'ємів підключення, дозволяє досить швидко формувати необхідну комплектацію.

Обробка даних можлива, як і на стороні контролера, так і передача їх на віддалений сервер. При цьому описувати весь процес роботи з даними необхідно програмісту, що виконує введення експлуатацію. Оскільки підтримується лише невеликий набір протоколів, то роботу з нестандартними протоколами необхідно реалізовувати самостійно, або перетворювати їх, або передавати на віддалений сервер для подальшої обробки.

Завдання введення в експлуатацію спрощується наявністю веб-інтерфейсу налаштування та інструментів візуального програмування призначених для користувача сценаріїв.

4. Простота експлуатації. Пристрій має досить високу складність експлуатації. Знизити її можна налаштуванням на етапі введення в експлуатацію.

Проаналізуємо основні переваги та недоліки універсального контролера автоматизації Wired Board 6.

Переваги:

- широкий функціонал і хороша розширюваність;
- наявність безлічі портів введення/виводу;
- графічна система програмування призначених для користувача сценаріїв;
- гнучка, відкрита архітектура.

При цьому можна виділити наступні недоліки:

- відсутність підтримки безлічі інтерфейсів для призначених для користувача датчиків, наприклад SPI, I2C, повноцінного CAN;
- відсутність підтримки багатьох промислових протоколів зв'язку;
- висока складність введення в експлуатацію.

1.5 Промисловий панельний контролер VP-4131-EN [5].

Загальний вигляд контролера наведено на рисунку 1.5.

Рисунок 1.5 – Зовнішній вигляд контролера VP-4131-EN

Характеристики контролера:

- тип встановленого процесора – Intel PXA270;
- частота процесора – 520 МГц;
- тип оперативної пам'яті – SDRAM;
- встановлений обсяг оперативної пам'яті – 128 МБ;
- обсяг енергозалежної пам'яті SRAM – 512 кБ;
- обсяг вбудованої Flash-пам'яті – 128 МБ;
- обсяг пам'яті EEPROM – 16 кБ;
- кількість COM-портів всього – 2: один RS-232, один RS-485;
- кількість портів USB v1.1 – 2;
- кількість портів 10/100 Mbit/s – 1;
- кількість слотів всього – 3;
- встановлена операційна система – Windows CE 5.0.

1.6 Сенсорний панельний контролер LSIT 07-400[6].

LSIT 07-400 (рисунок 1.6) – це потужний ПЛК з сенсорною панеллю, що дозволяє створювати інтерактивний інтерфейс користувача. Контролер обладнаний процесором ARM9 400 МГц і має пам'ять 100 МБ, чого з надлишком вистачає для управління технологічним процесом і його візуалізації.

СПК забезпечений набором портів (COM0, COM1, Ethernet) для підключення модулів розширення, за допомогою яких можна будувати системи управління самого різного масштабу і різного рівня складності, чому служать вбудовані в модулі ПІД-регулятори з можливістю автоналаштування.

При своїй багатофункціональності LSIT 07-400 досить компактний: його розмір трохи більше двадцяти сантиметрів (213 × 146 × 48,2 мм), а маса не перевищує 800 г – почасти через пластикового корпусу. При цьому ступінь захисту корпусу – IP65, тобто СПК повністю забезпечений від пилу і дощу. Також корпус стійкий до впливу ультрафіолетових променів.

Рисунок 1.6 – Сенсорний панельний контролер LSIT 07-400

Модулі вводу/виводу, що підключаються до контролера за протоколом зв'язку Modbus RTU, мають оптимальну комбінацію входів/виходів

тазабезпечують рішення задач по створенню різних АСУ ТП з мінімальними витратами. З модулями від компанії «Інформаційні технології» користувачі отримують наступні переваги:

- оптимальну комбінацію входів/виходів для кожної області застосування;
- можливість замовити конфігурацію входів/виходів під індивідуальні завдання;
- утиліту, що дозволяє швидко налаштувати модуль;
- інтеграцію з програмним середовищем розробки ScreenEditor і можливість роботи з пристроями сторонніх виробників.

У лінійку даних пристроїв входять дві моделі: модуль IT 1704 і модуль IT 1705. Модуль IT 1704 (рисунок 1.7) оптимально підходить для вирішення завдань в області створення систем управління вентиляцією. Модуль IT 1705 призначений для вирішення завдань в галузі управління компресорно-конденсаторним блоком (ККБ).

Рисунок 1.7– Модуль введення/виведення IT 1704

Середовище розробки ScreenEditor дозволяє створювати графічний інтерфейс користувача, застосовуючи готові елементи візуалізації з вбудованою бібліотеки, а також індивідуальні та шаблонні програми управління різними

технологічними процесами. Загальний вигляд бібліотеки зображень середовища програмування ScreenEditor наведено на рисунку 1.8. У програмі велика кількість вбудованих функцій, графічних елементів, забезпечена можливість використовувати власні графічні елементи, включаючи анімацію і управління ними.У програмі ScreenEditor розробник може створювати, редагувати і зберігати призначені для користувача вікна, які будуть відображатися на дисплеї СПК, створювати алгоритм управління технологічним процесом, використовуючи вбудовані математичні функції середовища розробки.

Налагодити роботу створеного проекту можна за допомогою наданої компанією «Інформаційні технології» програми-емулятора мережі пристроїв, яка дозволяє побачити роботу системи в режимі налагодження на комп'ютері розробника. Готову до роботи програму завантажують в панель за допомогою USB-кабелю або SD-карти.

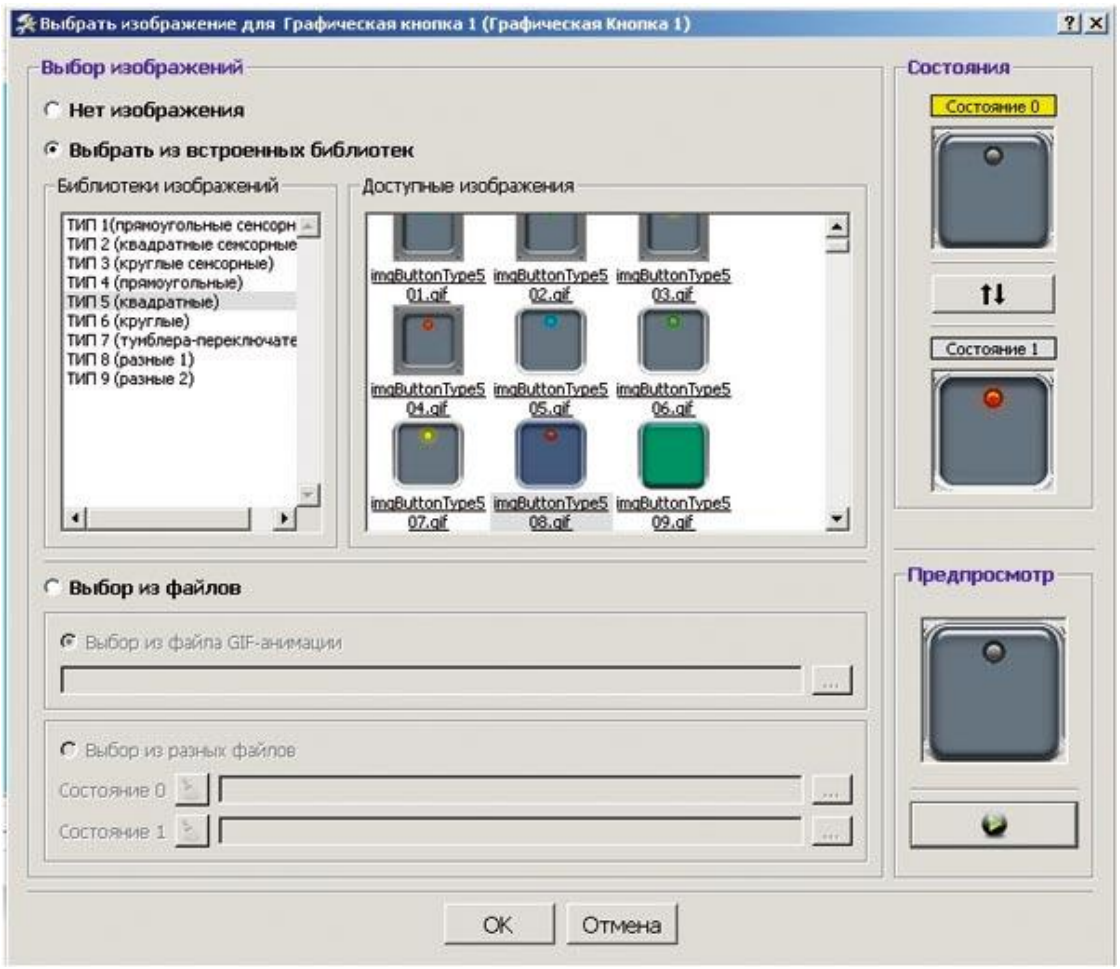


Рисунок 1.8 – Бібліотека зображень середовища програмування

1.7 Контролер PACSystems RX3i компанії General Electric [7]

Контролер PACSystems RX3i компанії General Electric (рисунок 1.9) являє собою апаратну платформу для побудови керуючих і розподілених систем введення/виводу в сучасну епоху промислового інтернету. Серед основних переваг промислових контролерів:

- висока продуктивність промислового контролера;
- гнучкість і масштабованість застосування (Standalone і Backplane);
- використання в резервуються конфігураціях (Hot і Cold Redundancy);
- сумісність з модулями введення/виведення застарілого ПЛК серії 90-30;
- застосування суперконденсаторів для батарейній підтримки пам'яті;
- підтримка сучасних відкритих комунікацій (Profibus DP, DeviceNet, CANOpen, Modbus TCP/RTU, EGD, Genius, OPC-UA, HART, DNP3, IEC60870/61850, PROFINET);
- широкий набір модулів введення/виведення з підтримкою їх гарячої заміни в контролері;
- є спеціальні версії модулів General Electric (раніше GE Fanuc, нове – Emerson) на широкий температурний діапазон від -40 до +60 оС;
- програмування промислового контролера на стандартних інженерних мовах IEC 1131-3;
- готовність для використання в інтернет-орієнтованих системах управління (outcome optimizing control), в системах віддаленого моніторингу та діагностики RMD або проактивного обслуговування (predictive maintenance).

Рисунок 1.9 –КонтроллерPACSystemsRX3i

В основі керуючої системи ПЛК контролера PACSystems RX3i компанії General Electric (раніше GE Fanuc, нове Emerson) лежить лінійка процесорних модулів на базі сучасних високопродуктивних багатоядерних чіпів Intel і AMD, забезпечених пам'яттю до 64 Мб.

Тобто, доступні процесори ПЛК для «всіх випадків життя»:

- для бюджетних систем – CPE302/305;
- якщо потрібні COM-порти і більше пам'яті для логіки – CPE310;
- гаряче резервування і вбудований PROFINET – CPE330 (з використанням модулів рефлекторної пам'яті RMX) або CPE400 (синхронізація по Ethernet);
- широкий діапазон температур, розподілений введення/виведення, вбудований інтернет-шлюз – CPE400;
- CPL400 – вільно програмований контролер під Linux, якщо потрібен шлюз не для Predix, а будь-якого хмари або локального користувача рішення.

Промисловий контролер PACSystems RX3i складаються з базової плати на 7, 12 або 16 слотів і можуть включати до 7 плат розширення на 5 або 10 слотів під модулі вводу/виводу. З метою сумісності з модулями введення/виведення застарілої платформи програмно логічного контролера

серії 90-30 (GE Fanuc), PACSystems RX3i мають, поряд з швидкими слотами паралельної шини PCI 27 МГц (для модулів серії IC695), слоти для послідовного обміну даними (для модулів серій IC694 і IC693).

Електропостачання 24В постійного і 220В змінного струму. Дискретні модулі 5/12/24/48/120/240В постійного і змінного струму до 32 каналу на модуль, реле 4А. Аналогові модулі вводу/виводу 0-20mA, 4-20mA, +/- 20mA; +/- 10V, 0-10V, +/- 5V, 0-5V, RTD. Всі модулі повністю індивідуально програмно-настроюються. Для аналогового введення застосовуються АЦП 24-біта на виході яких 32-бітові змінні з плаваючою комою. Модулі вводу/виводу мають поканального самодіагностикою, автокалібруванням, забезпечують перевірку на достовірність, генерацію тривог і підтримують режим «гарячої заміни» без зупинки виконання логіки.

Як інтерфейс для розподілених систем введення/виведення General Electric (раніше GE Fanuc, нове – Emerson) може застосовуватися найбільш сучасний, швидкісний, зручний для настройки і відмовостійкий Ethernet-протокол PROFINET. Модуль контролера PROFINET забезпечує обмін даними з 64 слейв-пристроями по оптоволокну або міді на швидкості до 1000 Mbps. При топології мережі «кільце» може використовуватися протокол MRP (Media Redundancy Protocol).

Системи високої готовності, програмовані логічні контролери PACSystems RX3i, включають в себе крім спеціалізованих процесорних модулів, модулі рефлекторної пам'яті RMX, які дозволяють синхронізувати дані в основному і резервному контролері по оптоволоконного зв'язку на швидкості 2,12 Гб два рази за цикл (рисунок 1.10).

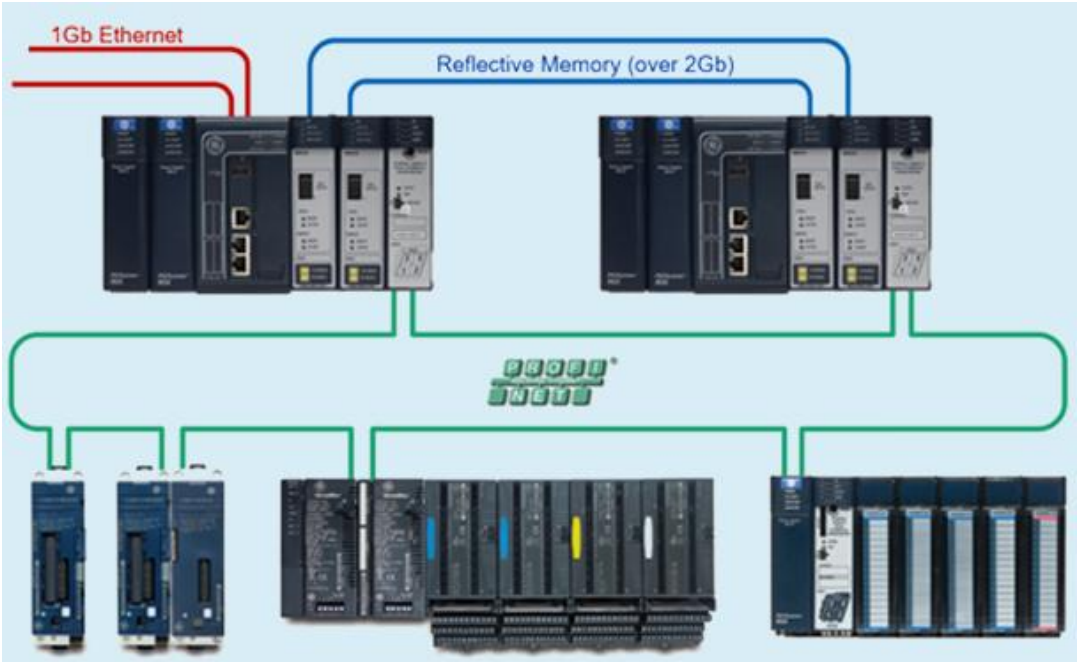


Рисунок 1.10 – Модульная конфигурация платформы

Конфігурація та налаштування платформи PACSystems RX3i компанії General Electric (раніше GE Fanuc, нове Emerson) проводиться в інтегрованої інструментальної середовищі Machine Edition. Логіка управління складається з програмних блоків, які можуть бути написані на мовах релейно-контактних схем LD, функціональних блоків FBD і структурованого тексту ST.

На базі платформи PACSystems RX3i General Electric і програмної технології EGD (Ethernet Global Data) може бути розгорнута гібридна розподілена програмно-апаратна система управління (DCS) – Proficy Process Systems.

1.8 Промисловий панельний контролер VP-2117[8]

Промисловий панельний контролер VP-2117 (рисунок 1.11) – це програмований контролер на базі MS DOS сумісної операційної системи з вбудованим ISaGRAF Версія 3 і вбудованими засобами операторського інтерфейсу (HMI), поєднуючи в собі функції управління технологічним

процесом, відображення технологічної інформації і введення команд оператора . VP-2117 має на задній стороні 3 слота розширення для модулів серій I-8K/I-87K високого профілю.

Рисунок 1.11 – Зовнішній вигляд контролера VP-2117

Основні характеристики контролера VP-2117:

- операційна система: MiniOS7;
- встановлене програмне забезпечення: ISaGRAF;
- процесор: 80186 або сумісний (16-біт, 80 МГц);
- оперативна пам'ять: 768 Кб;
- Flash-пам'ять: 512 Кб;
- кількість слотів розширення: 3;
- порт Ethernet: 1 x 10/100 Base-TX;
- LCD дисплей: STN, 128x64 Dot Matrix LCD;
- розміри: 182 мм x 158 мм x 125 мм;
- робоча температура від -20 до +70 ° C.

1.9 Перетворювачі протоколів компанії ADFweb.com

ADFweb.com –, італійська компанія, що швидко розвивається та займається виробництвом обладнання для систем зв'язку. Основними

продуктами є перетворювачі інтерфейсів і протоколів засобів автоматизації, а також мости, комутатори, сплітери, повторювачі сигналів і т.п. Виробляють обладнання для великої кількості інтерфейсів і протоколів. Компанія має повний цикл виробництва від розробки до впровадження та пропонує комплексні технічні рішення, а також рішення «під ключ» [9].

Оскільки обладнання компанії працює практично з усіма промисловими протоколами, то комбінуючи різні пристрої, можна підключати будь-які датчики до систем збору інформації. Системи збору інформації при цьому також можуть мати абсолютно різні інтерфейси і протоколи.

Розглянемо рішення компанії ADFweb.com для зв'язку з великою кількістю периферійних пристроїв, за раніше описаним критеріям.

1. Функціональність. Пристрої сполучення підтримує безліч промислових протоколів і інтерфейсів. Пристрої спеціально розробляються для цілей промислової автоматизації та оснащені захистом від перенапруги і гальванічно розв'язаними входами. Корпус дозволяє встановлювати пристрої на DIN-рейку.

2. Можливість розширення. Висока модульність рішення забезпечується великою лінійкою мостових перетворювачів і комутаторів. Можливе підключення будь-якої кількості пристроїв послідовно і обмежена кількість паралельно (в залежності від використовуваного протоколу зв'язку)

3. Простота введення в експлуатацію. Введення в експлуатацію полягає в підключенні пристрою і настройки правил перетворення сигналів. Налаштування здійснюється через веб-інтерфейс, або через сервісне ПО, в разі відсутності Ethernet порту.

4. Простота експлуатації. Одного разу налаштоване пристрій не потребує подальшої налаштування. Конфігурація зберігається в незалежній пам'яті і відновлюється після включення.

Розглянемо основні переваги та недоліки система побудованих на перетворювачах протоколів ADFweb.com:

Переваги:

- широка лінійка пристроїв підтримує практично всі промислові протоколи;
- висока надійність, забезпечується простотою роботи і мінімальною кількістю налаштувань.

Недоліки:

- відсутня можливість роботи з малопоширеними, або призначеними для користувача промисловими протоколами;
- Не підтримується робота з непромисловими інтерфейсами зв'язку – SPI, I2C, 1-Wire.

1.10 Універсальний UPnP міст для вбудованих пристроїв

Проект універсального UPnP моста [10] був запропонований фахівцями Канвонського національного університету, даний проект є побудова системи для об'єднання IP мереж і не IP мереж, де в якості керуючого пристрою виступає персональний комп'ютер з встановленим на ньому дистрибутивом операційної системи Windows XP.

Розглянемо дане рішення, за раніше описаним критеріям.

1. Функціональність. Пристрої сполучення підтримує такі інтерфейси як: RS232, RS485.

2. Можливість розширення. Висока модульність рішення забезпечується використанням підключених пристроїв до материнської плати комп'ютера.

3. Простота введення в експлуатацію. Пристрій підтримує технологію Plug and Play, в результаті чого вся настройка проводитися автоматично.

4. Простота експлуатації. Одного разу налаштоване пристрій не потребує подальшої налаштування. Конфігурація зберігається в незалежній пам'яті і відновлюється після включення.

Розглянемо основні переваги та недоліки даної системи.

Переваги:

- простота роботи для кінцевого користувача в зв'язку з підтримкою технології Plug and Play.
- висока надійність, забезпечується простотою роботи і мінімальною кількістю налаштувань.

Недоліки:

- складність системи і необхідність у використанні персонального комп'ютера;
- відсутність підтримки таких інтерфейсів передачі інформації, як: SPI, I2C і CAN;
- низька кількість периферійних пристроїв, обумовлене кількістю доступних пристроїв на обраної материнської плати.

1.11 Система для взаємодії з мережами на базі архітектури ARM

Дана система була представлена Науково-технологічним університетом Тегерана [11]. Для її проектування був обраний мікроконтролер LPC2478 основні характеристики якого: ARM7 серія процесора, 96 кбайт ОЗП, 512 кбайт, наявність USB, 4 контролера UART і CAN контролер.

На базі мікроконтролера була запущена операційна система реального часу µC/OS-II. µC/OS-II – це портативна, масштабована і багатозадачна система для мікроконтролерів, мікропроцесорів і ЦСП написана на мові С стандарту ANSI [12].

Дана ОС дозволяє виконувати до 250 завдань паралельно, також вона підтримує семафори, прапори подій, взаємовиключення семафори, черги, завдання, і інше. Варто відзначити, що час виконання для більшості сервісів, що надаються µC/OS-II, є постійним, і детермінованим; час виконання не залежить від кількості завдань, які виконуються в додатку. З основних особливостей варто відзначити:

- простота у вивчанні;
- запускається на великій кількості процесорних архітектур.

Принцип роботи даної системи полягає в організації 8-х паралельних завдань, які працюють в виділені кванти часу. Перерахуємо ці завдання: відправлення даних отриманих з Ethernet, на UART; отримання даних з UART і установка прапора UART; відправлення даних отриманих з Ethernet, на USB; отримання даних з USB і установка прапора USB; відправлення даних отриманих з Ethernet, на CAN; отримання даних з CAN і установка прапора CAN; відправлення даних по Ethernet згідно з раніше встановленим прапором; Отримання даних з комп'ютера і відправлення відповідно до встановлених прапорів.

Розглянемо дане рішення по раніше описаним критеріям.

1. Функціональність. Пристрій працює з обмеженою кількістю протоколів і інтерфейсів, а саме: UART, USB, CAN і Ethernet. Пристрій дозволяє безпосередньо передавати дані між такими інтерфейсами як: UART і Ethernet; USB і Ethernet; а так же CAN і Ethernet.

2. Можливість розширення. На жаль, розширюваність у даного пристрою відсутня, однак є можливість підключати декілька пристроїв один до одного.

3. Простота введення в експлуатацію. Введення в експлуатацію полягає в підключенні пристрою і настройки правил перетворення сигналів.

4. Простота експлуатації. Одного разу налаштоване пристрій не потребує подальшої налаштування. Конфігурація зберігається в незалежній пам'яті і відновлюється після включення.

Розглянемо основні переваги запропонованої системи:

Переваги:

- висока надійність, забезпечується простотою роботи і мінімальною кількістю налаштувань.

недоліки:

- відсутність підтримки таких інтерфейсів передачі інформації, як: SPI, I2C;
- відсутність можливості передавати дані безпосередньо між не Ethernet портами;

- мала кількість периферійних пристроїв;
- складності при масштабуванні кількості доступних пристроїв.

1.12 Висновки

За результатом аналізу можна сказати, що не кожне з проаналізованих пристроїв, однозначно використовується в бізнесі. Можна виділити ряд загальних недоліків пристроїв, які, однак, не стільки є недоліками, скільки конкурентною перевагою даних пристроїв. Необхідний функціонал можуть забезпечити як ПЛК так і спеціалізовані перетворювачі інтерфейсів, яким в свою чергу ми і будемо віддавати свою перевагу. Основною перевагою же спеціалізованих перетворювачів є орієнтованість на комунікаційні інтерфейси, а також велика кількість підтримуваних протоколів.

З недоліків виділених пристроїв, варто звернути увагу на наступні моменти:

- мала кількість комунікаційних інтерфейсів – даний момент є важливим, оскільки від нього може залежати складність кінцевих систем, спроектованих на базі запропонованого рішення. У наведених вище систем відсутня можливість підключення великої кількості периферійних пристроїв, внаслідок чого, може бути ускладнене їх використання, наприклад – в системах збору інформації, в системах телеметрії і т.п. Так як необхідно буде забезпечити додаткові комунікаційні пристрої, що в свою чергу знизить надійність кінцевої системи користувача, а з іншого підвищить як складність системи, так і вартість випуску готового пристрою;
- відсутність підтримки деяких типів вбудованих інтерфейсів – даний момент є важливим, оскільки від нього залежить кількість доступних периферійних пристроїв доступних кінцевому користувачеві. Кількість же доступних пристроїв, в свою чергу може вплинути на такі параметри: надійність, вартість, складність, час виробництва і час складання однієї

одиниці продукту, ну і відповідно – на технічні характеристики готового пристрою;

- відсутність підтримки багатьох не стандартизованих протоколів. Можлива часткова підтримка користувацьких змін стандартних протоколів. Це зауваження особливо важливо для вбудованих систем, оскільки більшість вбудованих пристроїв використовують стандартизовані протоколи зв'язку. Мова йде не тільки про пристрої розробляються кінцевим користувачем, але також і про пристрої випускаються промисловістю для використання у вбудованих системах-датчиках, зовнішні мікросхеми пам'яті, ЦАП, АЦП і т.д.

Таким чином, для того щоб усунути описані недоліки розроблювальний пристрій має володіє наступними характеристиками:

- наявність великої кількості різних комунікаційних інтерфейсів;
- використовувані інтерфейси зв'язку повинні бути по можливості універсальними – тип інтерфейсу повинен налаштовуватися програмними, або програмно-апаратними засобами;
- пристрій повинен забезпечувати зв'язок влаштування верхнього рівня з безліччю периферійних пристроїв;
- можливість підключення паралельно декількох пристроїв сполучення – якщо користувачеві недостатньо кількості інтерфейсів одного пристрою;
- універсальна робота з протоколами зв'язку – пристрій не повинно залежати від якогось певного протоколу, а має працювати з будь-яким, оскільки в розробці вбудованих систем широко застосовуються стандартизовані протоколи;
- пристрій повинен підтримувати можливість паралельної роботи з усіма інтерфейсами – влаштування верхнього рівня має мати можливість спілкуватися з будь-яким підключеним пристроєм незалежно від стану інших пристроїв;
- по можливості, мінімально задіяти ресурси пристрою верхнього рівня (як обчислювальні потужності ЦПУ, так і канал зв'язку);

– зв'язок між розробляються пристроєм і контролером верхнього рівня повинна здійснюватися через стандартизовані протоколи та інтерфейси. Протокол зв'язку повинен бути розроблений з урахуванням того, що функціонал розроблюваного устрою буде розвиватися.

2 РОЗРОБКА КОНЦЕПЦІЇ ПРИСТРОЮ ЗВ'ЯЗКУ

Метою даного розділу є аналіз вимоги до функціонала розроблювального пристрою, сформульовані в розділі 1 і вибір технологій для їх реалізації.

Виходячи з поставлених вимог, нам необхідно розробити апаратну частину пристрою, яка буде реалізовувати інтерфейси і програмну частину, яка буде взаємодіяти з інтерфейсами і забезпечувати передачу даних між ними.

Області застосування пристрою перетворення інтерфейсів:

- домашня автоматизація;
- IoT;
- робототехніка.

Пристрій повинен виконувати наступні функції:

- пересилання даних між контролером верхнього рівня і периферійними пристроями незалежно від типу протоколу зв'язку і інтерфейсу;
- підтримувати налаштування інтерфейсу зв'язку програмними, або програмно-апаратними методами;
- підтримувати настройку протоколів передачі програмними методами;
- зберігання налаштувань інтерфейсів і протоколів пристрої;
- пристрій повинен підтримувати обробку помилок (зв'язку і системних) і повідомлення про них контролер верхнього рівня. Будь-які помилки не повинні призводити до збою в роботі пристрою, втрати повідомлень, або порушення їх цілісності. Можливо також ведення логу роботи пристрою та його передача контролеру верхнього рівня у разі потреби;
- користувачеві повинні бути доступні один або кілька інтерфейсів UART, SPI, I2C.

При використанні тільки деяких типів інтерфейсів, припустимо тільки UART, їх кількість може бути збільшена, оскільки вони будуть

використовувати ті ж засоби фізичного підключення, що займали б інші інтерфейси.

Оскільки планується використовувати пристрій в IoT, робототехніці і домашньої автоматизації, областях, де немає фіксованих вимог до швидкості і періоду відправки повідомлень, то вимоги не пред'являються.

Так як пристрій повинен працювати з будь-якими протоколами, в тому числі для користувача, логічним буде відокремити канальні і фізичні рівні від верхніх рівнів моделі OSI [13]. Розроблювальний пристрій буде реалізовувати канальний і фізичні рівні певних інтерфейсів, а також деякі настройки для роботи з верхніми рівнями OSI. Дані настройки будуть залежати від реалізованого інтерфейсу.

Для простоти реалізації, виділимо наступні режими:

- режим налаштування пристрою;
- режим налаштування інтерфейсів;
- режим відправки-отримання повідомлення через інтерфейс.

Оскільки вимоги до зміни налаштувань в процесі роботи не обумовлені, то пристрій не буде їх підтримувати. Таким чином, кожен вищестоящий режим буде блокувати роботу з нижчестоящим. Отже, в режимі налаштування пристрою, неможливо буде налаштовувати інтерфейси, а в режимі настройки конкретного інтерфейсу, неможливо буде проводити обмін по ньому.

Для зберігання налаштувань необхідно передбачити незалежну пам'ять (зовнішню, або внутрішню). З огляду на те, що налаштування планується змінювати не часто, а також те, що до режиму налаштування порту не надаються спеціальні вимоги, крім можливості багаторазового налаштування, то необхідно вибрати тип пам'яті що підтримує множинні цикли стирання запису і не дуже вимогливу до швидкості роботи.

Налаштування, які можливо змінювати при кожній передачі (наприклад, швидкість інтерфейсу, або час очікування відповіді), не повинні зберігатися в незалежну пам'ять.

Оскільки зв'язок між розробляються пристроєм і контролером верхнього рівня повинна здійснюватися через стандартизовані протоколи та інтерфейси, необхідно також визначитися який інтерфейс найкраще використовувати.

З метою перевірки працездатності розробленого пристрою, в схемі необхідно застосувати емулятор сигналів передаються за обраним протоколу.

Аналіз пред'явлених вимог до пристрою визначив його структуру, яка приведена на рисунку 2.1.

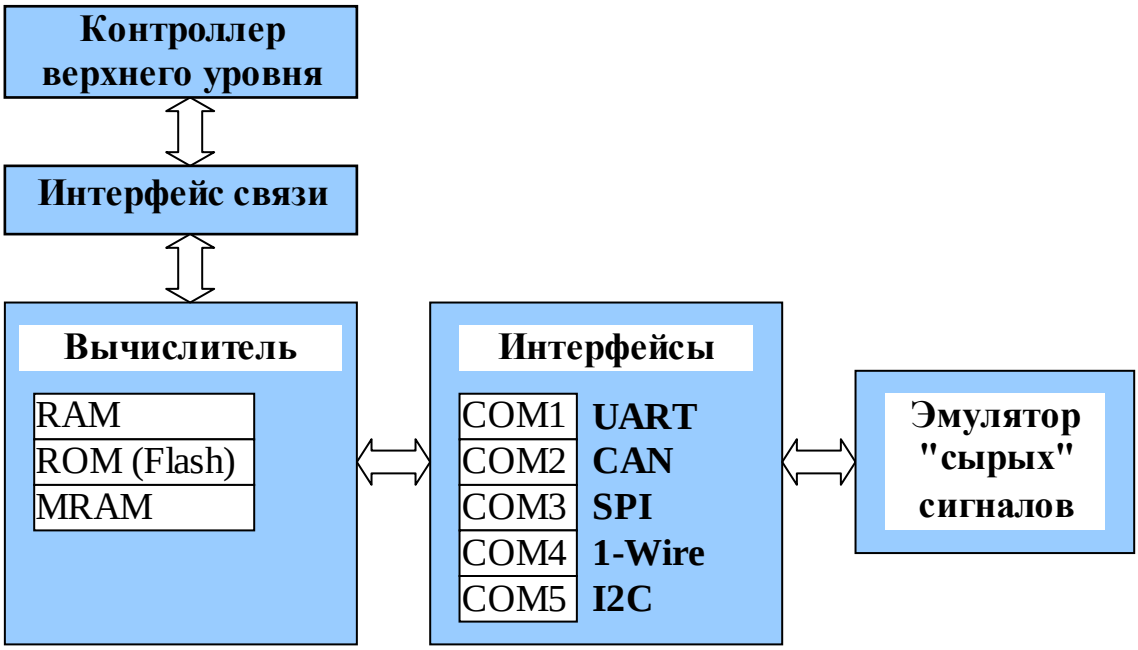


Рисунок 2.1 – Структурна схема підключення контролера з периферією

Оскільки дані вимоги, можна реалізувати на різних обчислювачах, тому для оцінки вимог до обчислювача необхідно розглянути основні їх типи, так як в залежності від обраного типу будуть актуальними ті чи інші характеристики.

2.1 Вибір і обґрунтування типу обчислювального пристрою

Розглянемо основні типи обчислювальних пристроїв з таких типів можна виділити основні три категорії, на підставі пристроїв з яких можна організувати проектування системи, а саме – система на кристалі (SoC), мікроконтролер і

програмована логічна інтегральна схема. Варто відзначити, що є й інші типи обчислювальних пристроїв, проте вони не будуть розглядатися в даній роботі внаслідок того, що функціональні можливості цих пристроїв не можуть забезпечити повноцінну реалізацію системи.

Для початку розглянемо системи на кристалах.

2.1.1 Системи на кристалі

Система на кристалі – це електронна схема, що виконує функції цілого пристрою (наприклад, комп'ютера) і розміщена на одній інтегральній схемі. Залежно від призначення вона може оперувати як цифровими сигналами, так і аналоговими, аналого-цифровими, а також частотами радіодіапазону. Як правило, застосовуються в портативних і вбудованих системах [14].

Типова SoC містить:

- один або кілька мікроконтролерів, мікропроцесорів або ядер цифрової обробки сигналів (DSP). SoC, що містить кілька процесорів, називають багатопроцесорною системою на кристалі (MPSoC);
- банк пам'яті, що складається з модулів ПЗП, ОЗП, ППЗП або флеш;
- джерела опорної частоти, наприклад, кварцові резонатори і схеми ФАПЧ (фазового автопідстроювання частоти);
- таймери, лічильники, функціонал затримки після включення;
- блоки, що реалізують стандартні інтерфейси для підключення зовнішніх пристроїв: USB, FireWire, Ethernet, USART, SPI;
- блоки цифро-аналогових та аналого-цифрових перетворювачів;
- регулятори напруги та стабілізатори живлення.

Розглянемо характеристики системи на кристалі на прикладі Amlogic S905X3, дана система не є особливо передової, проте на її базі можна ознайомитися з функціональністю розглянутого класу пристроїв. На рисунку 2.2 представлені функціональні блоки, присутні в Amlogic S905X3.

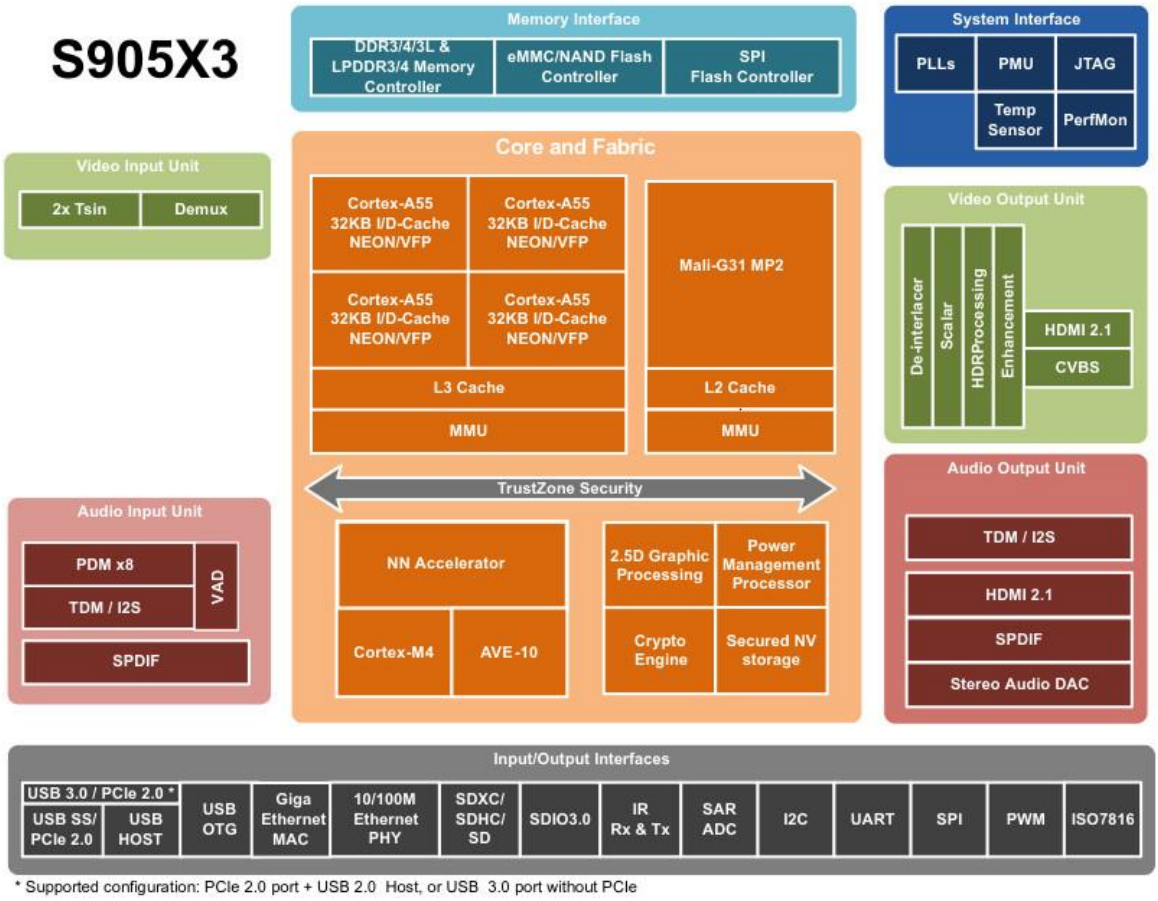


Рисунок 2.2 – Функціональні блоки SoC Amlogic S905X3

Технічні ж характеристики даної системи представлені нижче, а саме:

- Підсистема ЦП;
- Чотирьохядерний процесор Arm Cortex-A55 з Armv8-A Neon і Crypto розширенням, 8-ступінчастий конвеєр, уніфікована кеш-пам'ять L3;
- Ядро Arm Cortex-M3 для обробки системи управління;
- Блок обробки тривимірної графіки – графічний процесор Arm Mali G31MP2 з підтримкою OpenGL ES 3.2, Vulkan 1.0 і OpenCL 2.0;
- 2.5 D графічний процесор;
- Відео вхід/вихід інтерфейс;
- Аудіо декодер і вхід/вихід;
- Інтерфейс пам'яті і сховища;

- 32-розрядний інтерфейс пам'яті DRAM з двома рангами і максимальним загальним адресним простором 4 ГБ;
- Підтримка SLC/TLC NAND-флеш з 60-бітовим ECC;
- Підтримка послідовної 1, 2 або 4-бітної NOR флеш-пам'яті через SPI;
- Вбудована пам'ять 4 Кбайт для одноразового програмування для зберігання ключів;
- Вбудований 10/100/1000M Ethernet MAC з інтерфейсом RGMII;
- Інтегрований 10/100M Ethernet PHY інтерфейс;
- Підтримка Wi-Fi і Bluetooth через PCIE, SDIO, USB, UART або PCM;
- Мережевий інтерфейс оптимізований для змішаного трафіку WiFi і Bluetooth;
- Інтерфейс цифрового телебачення;
- Інтегровані контролери введення-виведення і інтерфейси;
- 1x USB XHCI OTG 2.0 порт;
- Кілька інтерфейсів PWM, UART, I2C і SPI;
- Вбудований 10-розрядний АЦП SAR з 4 вхідними каналами.

Дані технічні характеристики є достатніми для реалізації пристрою, однак як можна судити по вище наведеним характеристикам, функціональність аналогічних обчислювальних пристроїв є надмірною для даної технічної задачі. Варто зазначити що, незважаючи на наявність необхідних блоків в даних рішеннях присутня велика кількість блоків, які є зайвими, а саме таких як блоки відеоспостереження, блоки аудіо-обробки і блоки зв'язку PCIE, що обумовлено сферою використання таких рішень.

Реалізувати дане технічне завдання на базі аналогічних обчислювачів можливо, однак у цього рішення є обмеження пов'язані з вартістю таких обчислювачів. У зв'язку з вище перерахованими недоліками в даній роботі не

будуть використовуватися аналогічні рішення, тому розглянемо наступну категорію обчислювальних пристроїв.

2.1.2 Програмовані логічні інтегральні схеми

Програмована логічна інтегральна схема – електронний компонент (інтегральна мікросхема), який використовується для створення конфігурованих цифрових електронних схем. На відміну від звичайних цифрових мікросхем, логіка роботи ПЛІС не визначається при виготовленні, а задається за допомогою програмування (проектування). Для програмування використовуються програматор та IDE (середа проектування), що дозволяють задати бажану структуру цифрового пристрою у вигляді принципової електричної схеми або програми на спеціальних мовах опису апаратури: Verilog, VHDL, AHDL [15]

ПЛІС представляє матрицю з малою кількістю входів (від двох до п'яти входів) логічних елементів, тригерів, відрізків ліній зв'язку, що з'єднуються перемичками з польових транзисторів. ПЛІС програмуються зміною рівня електричного поля (field) в затворах транзисторів. На відміну, наприклад, від LPGA – Laser Programmable Gate Array. Затвори всіх «програмують» польових транзисторів підключені до виходів тригерів одного довгого зсувного регістру, який заповнюється при програмуванні ПЛІС. Деякі з ділянок цього регістра можуть також виконувати роль осередків ПЗП.

Прошивка зазвичай зберігається в ПЗП, що знаходиться поруч з ПЛІС і після включення живлення або за сигналом скидання вона автоматично переписується в програмує зсувний регістр ПЛІС. Цей процес називається конфігурацією ПЛІС. Так як основу ПЛІС складають тригери, що зберігають прошивку, то ПЛІС виготовляються за технологією мікросхем статичного ОЗП.

У порівнянні з CPLD, ПЛІС виграють

- по-перше, в необмеженій кількості перепрограмування,
- по-друге, в логічній ємності, в тому числі в питомої ємності вентилів на цент,

– по-третє, в малому енергоспоживанні.

Як правило, ПЛІС мають на два – три порядки більшу ємність в числі еквівалентних логічних вентилів, ніж CPLD і також як статичне ОЗП, майже не споживають енергії при відсутності перемикань. Крім того, у ПЛІС на порядок вище надійність (нижче інтенсивність відмов), ніж у CPLD.

До недоліків відносять необхідність зовнішнього ПЗП прошивки, генератора синхросерії. Але ПЗП з 8 виводами займає на платі значно менше місця, ніж сама ПЛІС з багатьма сотнями висновків. Те ж стосується генератора синхросерії [16].

У документації компанії Альтера зустрічається вираз Logic Array Block (LAB) – масив логіки. У компанії Xilinx в мікросхемах ПЛІС є приблизно такі ж блоки – Configurable Logic Block (CLB). Конфігурується логічний блок – це базовий елемент в ПЛІС, в ньому може бути виконана якась проста логічна функція або реалізовано зберігання результату обчислення в регістрах (тригерах).

Складність і структура логічного блоку з налаштуванням (CLB) визначається виробником.

Теоретично, конфігурується логічний блок може бути, наприклад, дуже простим, просто як окремий транзистор. Або він може бути дуже складним, як цілий процесор. Це крайні точки реалізації.

У першому випадку буде потрібно величезна кількість програмованих зв'язків, щоб потім з окремих транзисторів зібрати необхідну схему. У другому випадку зв'язків може потрібно і не так багато, але втрачається гнучкість проектування користувальницької схеми.

Саме тому конфігурується блок зазвичай представляє з себе щось середнє: він зазвичай досить складний, щоб можна було б зашити туди деяку функцію, а й досить малий, щоб розмістити безліч таких блоків всередині ПЛІС і щоб була можливість зв'язати їх в єдину схему.

Таким чином, вибір структури логічного блоку з налаштуванням виробником ПЛІС – це завжди пошук компромісу по площі кристала, по швидкодії, енергоспоживання і так далі.

Конфігурується логічний блок може складатися з одного або декількох базових логічних елементів. В англомовній літературі це Basic Logic Element (BLE) або просто Logic Element (LE). У ПЛІС зазвичай використовуються так звані LUT-based базові логічні елементи.

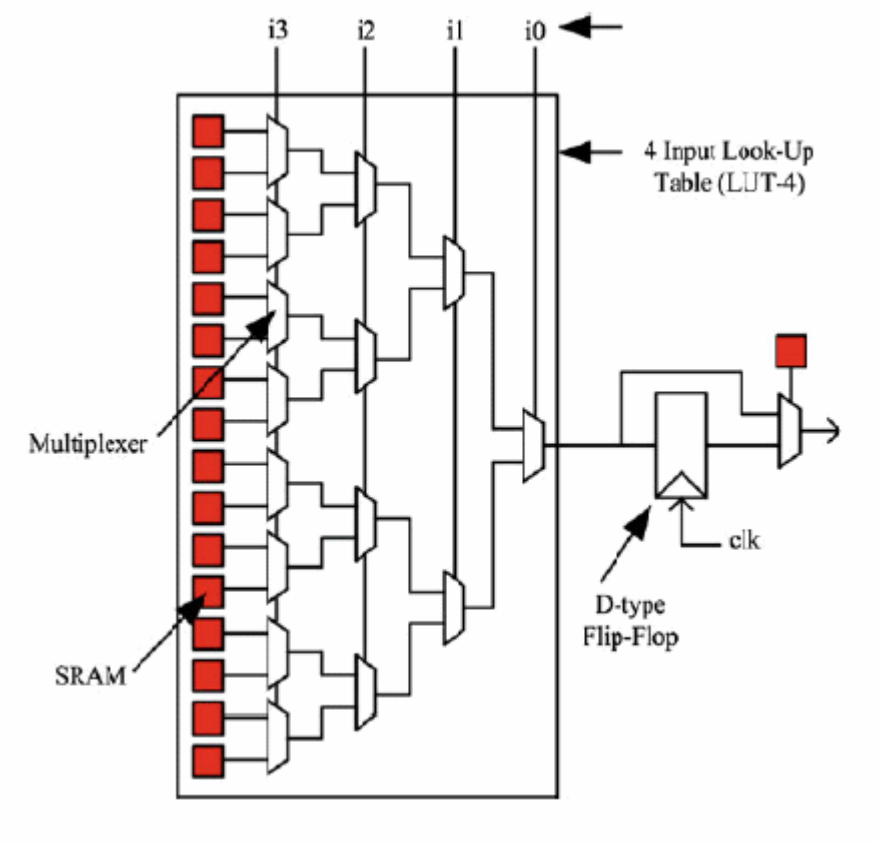


Рисунок 2.3 – Приклад традиційного базового логічного елемента

LUT – це Look-Up Table, таблиця перетворення. Наприклад, на рисунку 2.3, показаний четирехбітних LUT в складі базового логічного блоку. Тут четирехбітних числу на вході логічного функції ставиться у відповідність однобітний результат. Червоні квадратами позначено програмований елемент, регістр – це та пам'ять, де зберігається прошивка для ПЛІС. Можна помітити, що для конфігурації 4-х бітного LUT потрібно 16 конфігураційних регістрів.

Вміст цих регістрів визначають логічну функцію, реалізовану всередині базового логічного елемента.

Ще один конфігураційний регістр (одиночний червоний квадрат праворуч) визначає чи потрібно на вихід базового логічного елемента видавати прямо значення з LUT або потрібно видати зафіксоване в D-тригері значення з LUT. Фіксація і зберігання даних в цифрових схемах потрібна практично в будь-якому проекті.

В принципі базовий логічний елемент в різних ПЛІС виявляється набагато складніше, ніж наведений вище. Нижче, на рисунку 2.4 наведений базовий логічний елемент CPLD MAX II компанії Альтера.

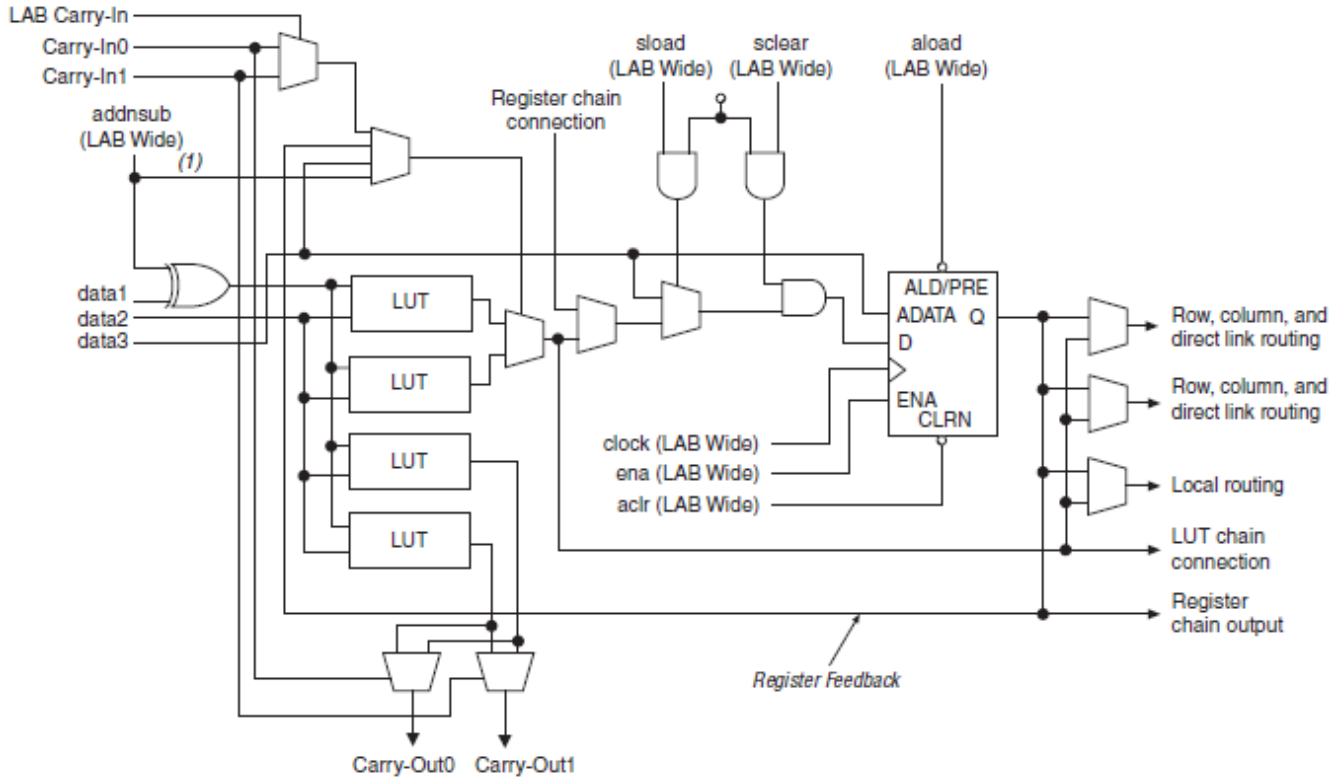


Рисунок 2.4 – Базовий логічний елемент CPLD MAX II компанії Альтера

Тут добре видно LUT і D-Тригер зберігання результату. Нижче, на рисунку 2.5 представлений базовий елемент Cyclone III.

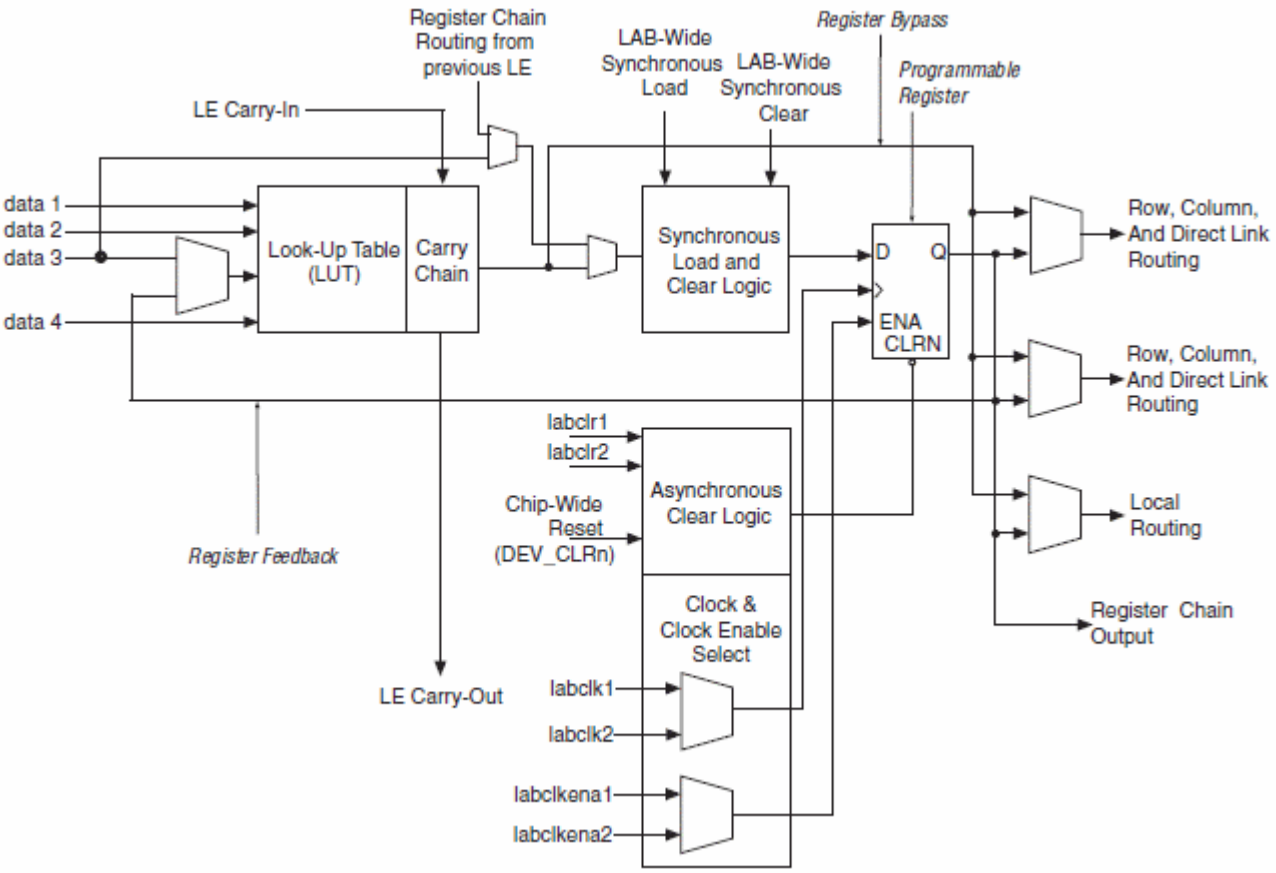


Рисунок 2.5 – Базовий логічний елемент FPGA Cyclone III компанії Альтера

У мікросхемах Альтери в одному LAB може міститися 10-16 LE.

У мікросхемах компанії Xilinx Virtex-6 базовий логічний елемент. Він представлений на рисунку 2.6.

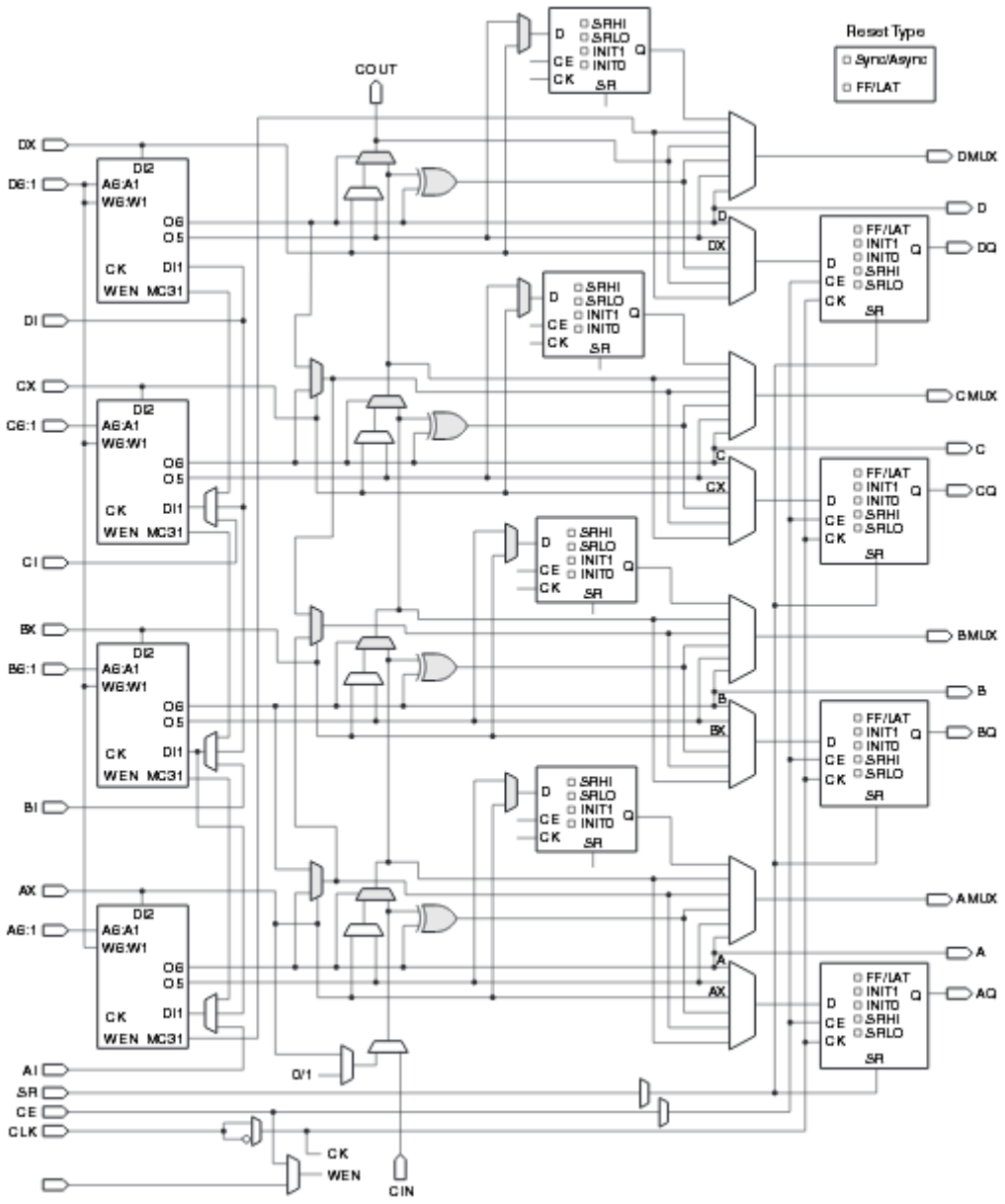


Рисунок 2.6 – Базовый элемент Xilinx Virtex-6 Slice

В одному CLB Virtex-6 є 8 LUT і 16 D-тригерів та інші елементи.

На рисунку 2.7 показаний базовий логічний елемент мікросхеми FPGA компанії Microsemi, серії 40MX.

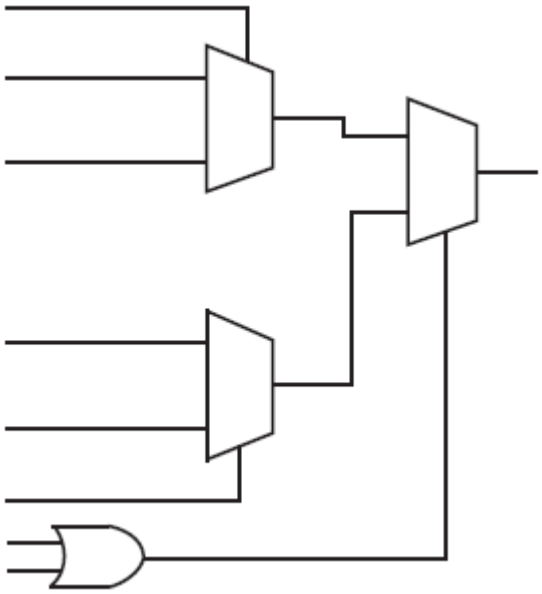


Рисунок 2.7 – Логічний модуль Microsemi-40MX серії

Вісім входів і один вихід. Тут немає ні Look-Up Table, ні навіть D-Трігера. Тригера, як і решта логіка, формуються де потрібно з елементарних блоків – Logic Module.

Щоб в ПЛІС заробила необхідна цифрова схема, необхідно; як настроїти наявні логічні блоки особливим чином, так і створити, запрограмувати зв'язку між логічними блоками.

Для цього в ПЛІС є спеціальні конфігуруються комутатори.

В англомовній документації зустрічаються такі терміни: FPGA Routing Architecture і Programmable Routing Interconnect. Це все про це, про програмованих зв'язках між логічними блоками.

Відомо дві основні методики побудови ПЛІС за типом архітектури зв'язків: острівна і ієрархічна.

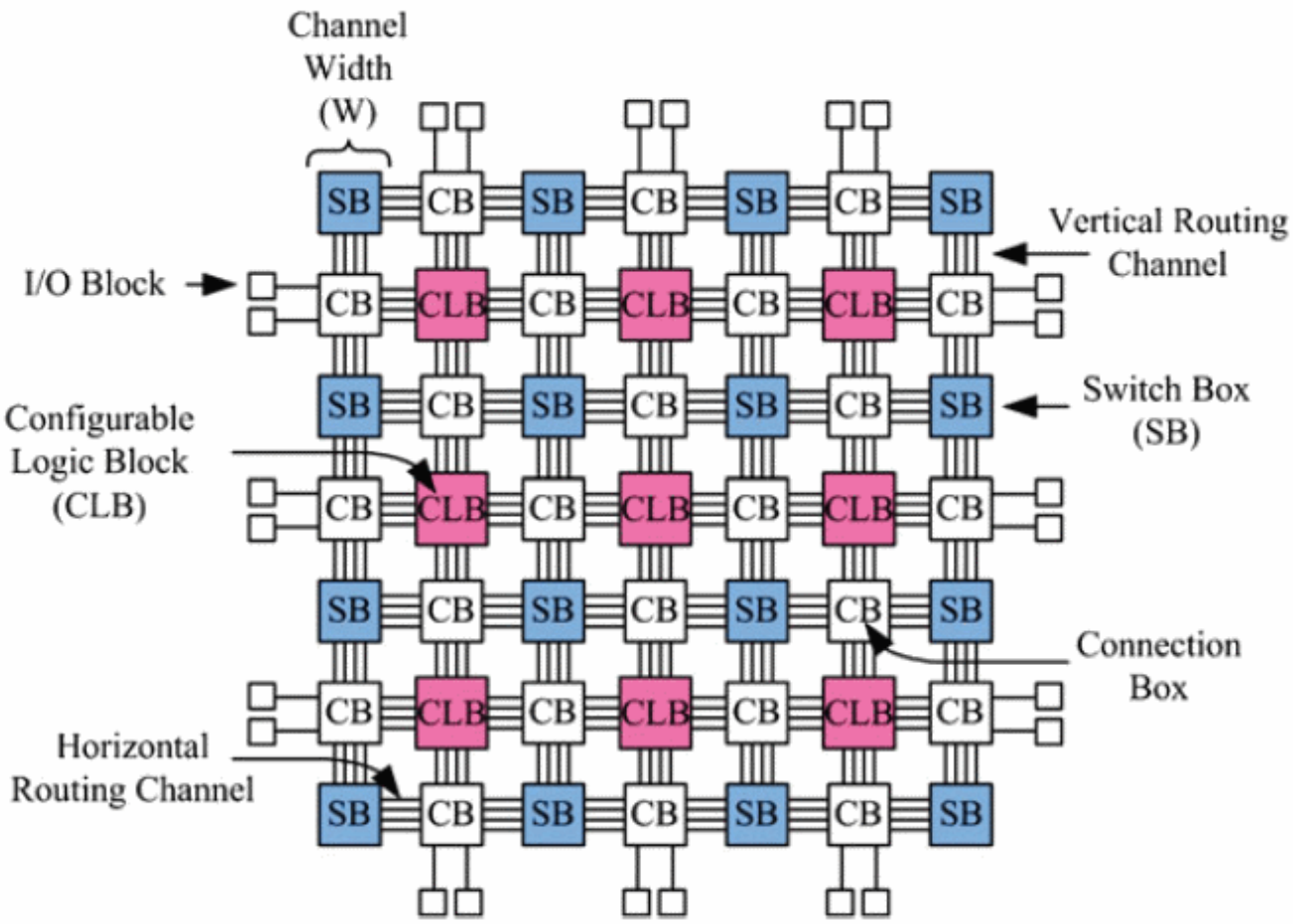


Рисунок 2.8 – Острівна ПЛІС.

Острівна ПЛІС називається так тому, що конфігуруються блоки усі рівні між собою і знаходяться, як острова в океані, між вузлами комутації та лініями зв'язку.

Основні архітектурні блоки острівної ПЛІС це: CB – Connection Box і SB – Switch Box. По суті це програмовані мультиплексори, підключають той чи інший CLB до іншої CLB через ланцюжки проводів в ПЛІС.

Це island-style FPGA або mesh-based FPGA. Типовий приклад таких мікросхем – це серії Altera Cyclone і Stratix (рисуноу 2.8).

Другий відомий тип ПЛІС – це ієрархічні ПЛІС (рисунок 2.9). Тут йде розрахунок на те, що в схемі завжди є місця, які взаємодіють один з одним більш тісно, ніж з віддаленими модулями проекту.

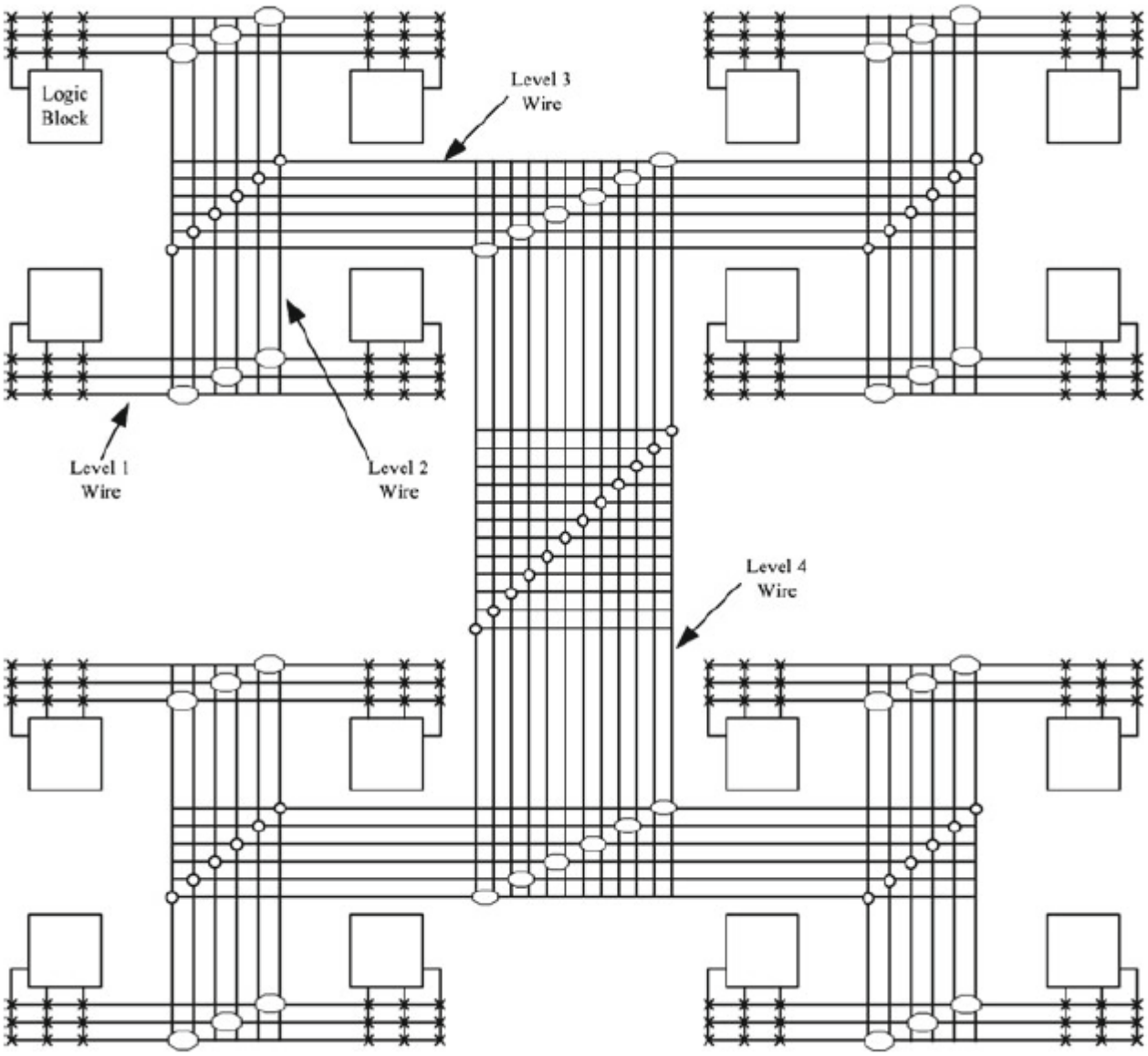


Рисунок 2.9 – Иерархическая ПЛИС.

Тут прилеглі CLB з'єднати досить просто, потрібно не багато комутаторів і виходять зв'язку працюють швидко.

Типові представники ієрархічних ПЛІС – це мікросхеми компанії Altera серії Flex10K, APEX.

Розглянемо характеристики типової плис на основі серії MAX V виробництва Altera, таблиця 2.1.

Таблиця 2.1 – Характеристики мікросхем ПЛІС серії MAX V

		MAX V (напряжение питания ядра 1.8 В)		
		5M570Z	5M1270Z	5M2210Z
Ресурси	Кількість логічних елементів	570	1270	2210
	Кількість еквівалентних макроячеек	440	980	1700
	Максимальна затримка між вхідними та вихідними висновками (нс)	9	43502	7
	Обсяг Flash-пам'яті користувача (кбіт)	8		
	внутрішнє ОЗП	Так	Так	Так
Архітектурні особливості	вбудований осцилятор	Так		
	Блок цифровий PLL	Так		
	Ланцюги граничного сканування	Так		
	Швидкісні регістри в елементах введення-виведення	Так		
	Програмований стан регістрів по включенню живлення	Так		
	Режим програмування конфігураційного Flash-ПЗП без зміни поточної конфігурації	Так		
Підсистема введення-виведення	Підтримувані рівні напруги введення-виведення (В)	1.2, 1.5, 1.8, 2.5, 3.3, 5.0		
	Підтримка виходів стандарту LVDS	Так	Так	Так
	Підтримка стандарту введення-виведення PCI 32-біт, 66-МГц	Ні	Так	Так
	Тригер Шмітта в елементах введення-виведення	Так		
	Програмована швидкість наростання вихідного сигналу	Так		

Таблиця 2.1 (продовження) – Характеристики мікросхем ПЛІС серії MAX V

Підсистема введення-виведення	Програмовані вбудовані підтягаючі резистори	Так
	Виводи з режимом "програмованої землі"	Так
	Режим виходів з відкритим стоком	Так
	Режим утримання стану шини	Так

Аналогічні характеристики дозволяють реалізувати на базі такого обчислювача високопродуктивне пристрій, який дозволить забезпечити більш високу пропускну здатність. Однак у рішень на базі ПЛІС є пару особливостей, а саме високий поріг входження і високе енергоспоживання, тому ми не будемо використовувати цей обчислювач в даній роботі. Розглянемо третій клас обчислювачів.

2.1.3 Мікроконтролери

Мікроконтролер (Micro Controller Unit, MCU) – мікросхема, призначена для управління електронними пристроями.

Типовий мікроконтролер поєднує на одному кристалі функції процесора і периферійних пристроїв, містить ОЗП і (або) ПЗП. По суті, це однокристальний комп'ютер, здатний виконувати досить прості завдання.

Відрізняється від мікропроцесора інтегрованими в мікросхему пристроями введення-виведення, таймерами і іншими периферійними пристроями [17].

Розглянемо характеристики мікроконтролерів STM32 на прикладі мікроконтролера STM32F103C8T6.

Даний мікроконтролер має наступні характеристики таблиця 2.2 [18].

Таблиця 2.2 – Технічні характеристики мікроконтролера STM32F103C8T6.

Розрядність	32 біта
Максимальна частота	72 МГц
Обсяг пам'яті програм (FLASH)	64/128 кбайт
Обсяг пам'яті даних (RAM)	20 кбайт
висновки	37
системний таймер	1
сторожові таймери	2
UART	3
SPI	2
I2C	2
CAN	1
USB	1
Контролери прямого доступу до пам'яті (DMA)	7
АЦП	2 АЦП, 10 каналів, час перетворення 1 мкс
Годинник реального часу	Так
Апаратний модуль розрахунку CRC	Так
Напруга живлення мікроконтролера	2 ... 3,6 В
Струм споживання	до 50 мА

Безумовно, характеристики даного контролера є скромними, особливо в порівнянні з характеристиками системи на кристалі, однак існують і більш потужні мікроконтролери, які також використовуються в промисловій та

домашній автоматизації та мають характеристики не гірших ніж системи на кристалі.

За сферою застосування даний тип обчислювача використовується в управлінні різними пристроями і їх окремими блоками:

- в обчислювальній техніці: материнські плати, контролери дисководів жорстких і гнучких дисків, CD і DVD, калькуляторах;
- електроніці та різноманітних пристроях побутової техніки, в якій використовується електронні системи управління – пральних машинах, мікрохвильових печах, посудомийних машинах, телефонах і сучасних приладах, різних роботах, системах «розумний будинок», і ін.

У промисловості:

- пристрої промислової автоматики – від програмованого реле і вбудованих систем до ПЛК;
- систем управління верстатами.

У той час як 8-розрядні мікропроцесори загального призначення повністю витіснені більш продуктивними моделями, 8-розрядні мікроконтролери продовжують широко використовуватися. Це пояснюється тим, що існує велика кількість застосувань, в яких не потрібна висока продуктивність, але важлива низька вартість. У той же час, є мікроконтролери, що володіють великими обчислювальними можливостями, наприклад, цифрові сигнальні процесори, що застосовуються для обробки великого потоку даних в реальному часі (наприклад, аудіо-, відео потоків) [17].

Як можна помітити мікроконтролер має характеристики схожими на системи на кристалі, проте вони не мають блоків відео- та аудіо- обробки, що дозволяє їх використовувати в роботі, так само варто відзначити, що в порівнянні з ПЛІС, дані обчислювачі мають меншу енергоспоживання. А в порівнянні з системами на кристалі, мають меншу вартість. З огляду на вищенаведені особливості, в даній роботі має сенс використовувати цей обчислювач, а саме – мікроконтролер.

2.2 Висновки

В даний момент на світовому ринку існує велика кількість різних сімейств, проте реально доступних з них, рядовому інженерові не так і багато основними доступними на українському ринку є STM32 і AVR фірми Atmel, нині придбаної компанією Microchip, розглянемо коротко плюси і мінуси кожного сімейства:

AVR – сімейство восьмибітних мікроконтролерів, що раніше випускалися фірмою Atmel, потім Microchip [19]. Можна виділити наступні переваги:

- Щодо низький поріг входження;
- Наявність dip корпусу.

Відповідно можна виділити наступні недоліки:

- Відносно висока вартість по порівняння з конкурентами;
- 8-ми бітове ядро.

В цілому сімейство AVR є, на даний момент, морально застарілим.

STM32 – сімейство 32-бітних мікроконтролерів виробництва STMicroelectronics. Чіпи STM32 групуються в серії, в рамках кожної з яких використовується один і той же 32-бітове ядро ARM, наприклад, Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0 + або Cortex-M0 [20]. З переваг, даного сімейства, можна виділити наступне:

- Співвідношення ціна/якість. Дані мікроконтролери працюють як на більш високих частотах в порівнянні з AVR, так і мають на борту більшу кількість периферії.
- Широке наявність на ринку різних моделей.
- 32-х бітове ядро.

Недоліки також варто відзначити – це більш високий поріг входження.

В цілому сімейство STM32 не тільки має переваги над сімейством AVR, а й є повноцінним технологічним стрибком, тому має сенс в даній роботі використовувати мікроконтролер сімейства STM32. Для скорочення часу розробки програмного забезпечення будемо використовувати відлагоджувальну

плату STM32. Ці плати мають всі необхідні характеристики для розробки прототипу готового виробу і налагодженні програмного забезпечення. Вибір відлагоджувальної плати буде проведено в наступному розділі.

3 ВИБІР ВІДЛАГОДЖУВАЛЬНОЇ ПЛАТИ

3.1 Загальні положення

Враховуючи те, що в якості мікроконтролера для реалізації поставленого завдання був обраний мікроконтролер серії STM, зробимо огляд і аналіз відлагоджувальних плат на основі сімейства компанії STMicroelectronics.

Засоби відлагодження і розробки можна розділити на дві основні групи:

– апаратні;

– програмні.

Обидві ці групи широко представлені компанією STMicroelectronics. До апаратних засобів, перш за все, відносяться налагоджувальні плати, які так само, як і самі мікроконтролери, на яких вони базуються (в даному випадку STM32), розділені на серії (рисунок 3.1) [21].

Рисунок 3.1 – Серії відлагоджувальних плат на базі мікроконтролера STM32

На даний момент, для користувачів є наявними більше сотні різних відлагоджувальних комплектів трьох основних груп: Nucleo, Discovery і Evaluation Boards, а також різні модулі розширення для плат серії Nucleo і комплекти розробки від сторонніх виробників.

3.2 Відлагоджувальні плати STM32 Nucleo

Відлагоджувальні плати виробництва компанії STMicroelectronics надають користувачям можливість швидко, ефективно і з мінімальними матеріальними витратами створювати прототипи, гнучкі до зміни і додаванню нового функціоналу. Також, ці плати мають відносно невисоку вартість. Зараз на ринку наявно близько тридцяти комплектів відлагоджувальних плат Nucleo [21]. В серію Nucleo входить три лінійки плат: Nucleo-32, Nucleo-64 і Nucleo-144, зовнішній вигляд і основні відмінності в інтерфейсах яких наведені на рисунку 3.2. Основна відмінність між платами полягає у кількості наявних виводів мікроконтролера.

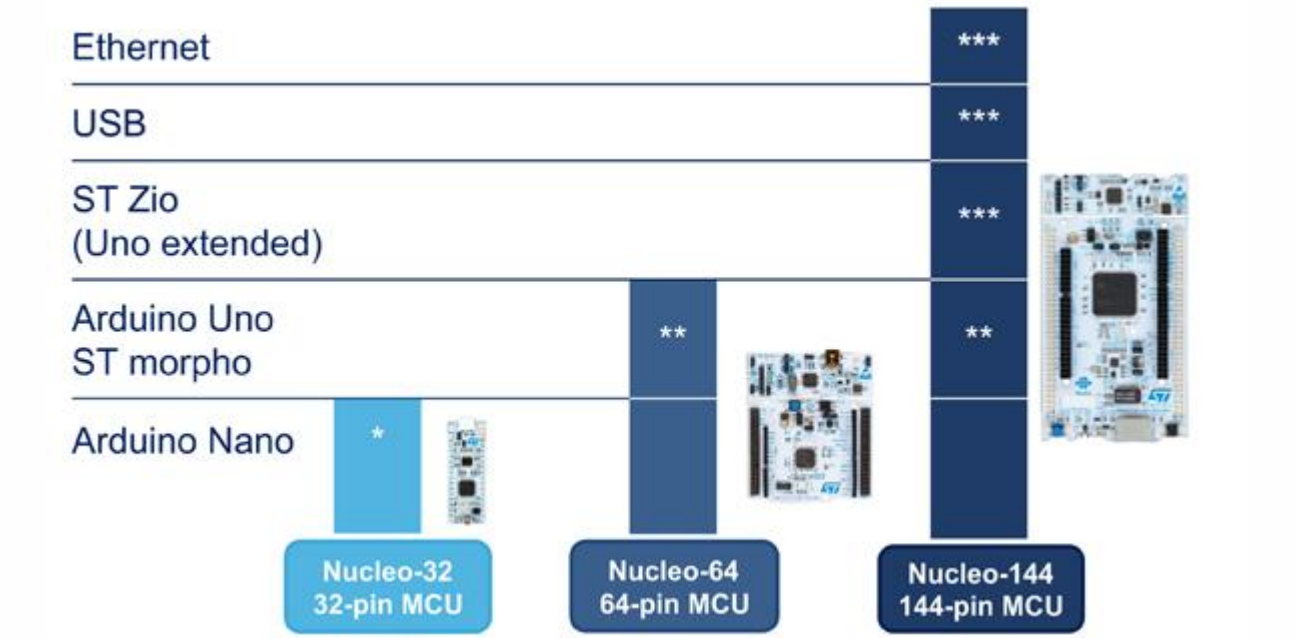


Рисунок 3.2 – Лінійки відлагоджувальних плат Nucleo

Ці плати мають досить обмежений функціонал при використанні їх в автономному режимі, вони призначені насамперед для спільного використання з модулями розширення. При використанні модулів, плата Nucleo є обчислювальним ядром системи і основою побудови всього рішення. Плати Nucleo можуть працювати з модулями Arduino: Nucleo-32 сумісні з Arduino nano, Nucleo-64 і Nucleo-144 здатні взаємодіяти з Arduino UnoV3 [21].

Можливості плат Nucleo можуть разюче відрізнятися в залежності від групи. Наприклад, Nucleo-32 і Nucleo-64 мають тільки набір з базових елементів (світлодіоди індикації стану і призначені для користувача, призначена для користувача кнопка, а також кнопка Reset) і вбудованим в плату програматором ST-LINK, а Nucleo-144 має більш розширений функціонал, що включає різні комунікаційні інтерфейси (Ethernet, USB і т.д.).

Незважаючи на загальну технологію побудови, плати серії Nucleo мають відмінності, починаючи від наведеного вище форм-фактора і вбудованих інтерфейсів і закінчуючи мікроконтролером, що використовується, (тобто і основними характеристиками плати), а також відрізняються вартістю, що важливо для розробника. На рисунку 3.3 наведена структура серії відлагоджувальних плат STM32 Nucleo.

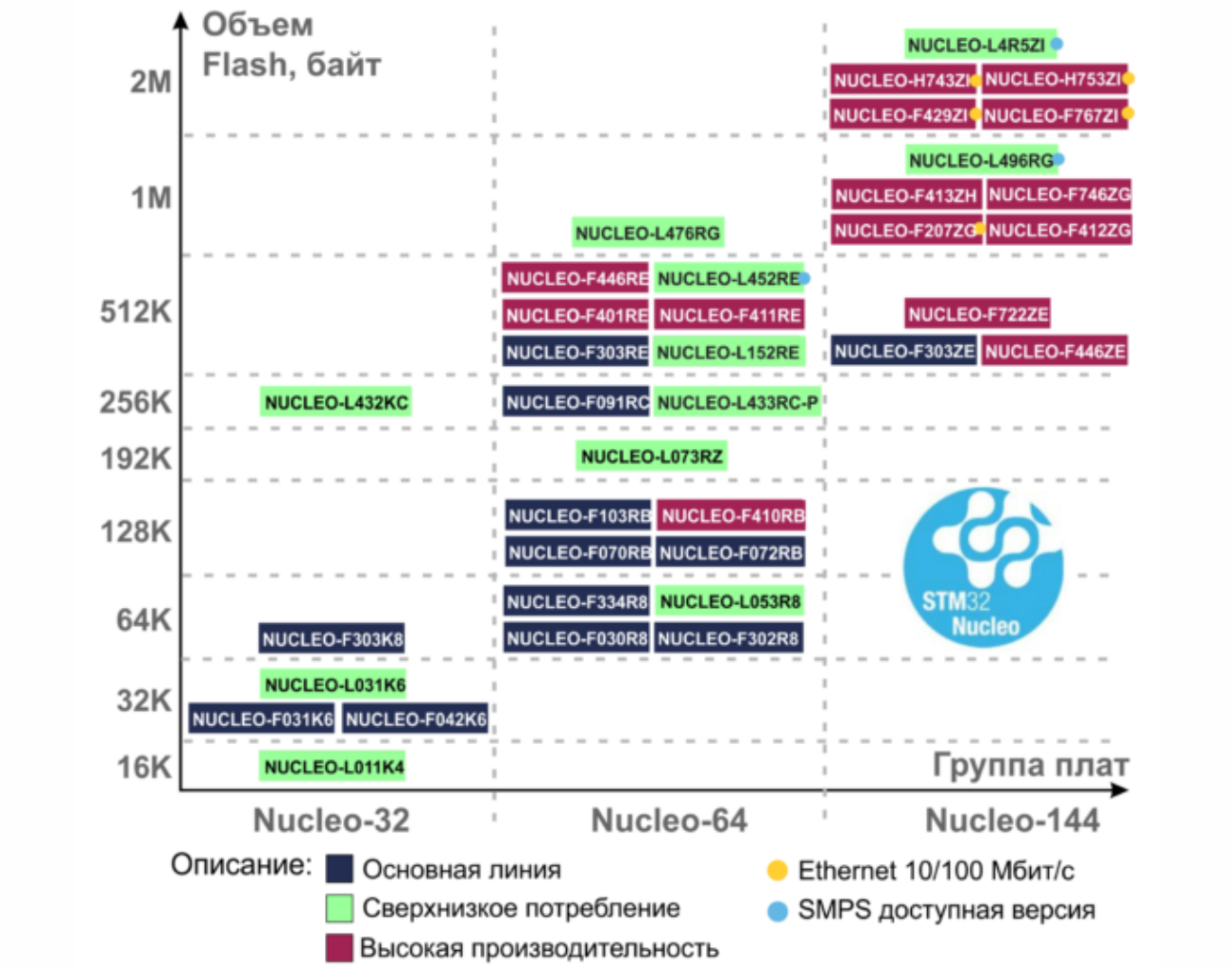


Рисунок 3.3 – Структура відлагоджувальних плат серії Nucleo

Функціонал відлагоджувальних плат Nucleo в основному визначається модулями розширень. Компанія STMicroelectronics пропонує широкий перелік таких плат (таблиця 3.1) [21]. Також можливе включення плат з підтримкою Arduino від сторонніх виробників.

Таблиця 3.1 – Плати розширення для відлагоджувальних плат сімейства Nucleo

Найменування	Короткий опис
X-NUCLEO-IKS01A2	Плата розширення, що базується на основі МЕМС-датчиків руху, а також датчиків стану навколишнього середовища.
X-NUCLEO-6180XA1	Плата розширення з датчиками контролю освітлення.
X-NUCLEO-53L0A1	Плата розширення на базі модуля лазерного дальноміра і датчика жестів VL52L0X, що застосовує технології Time-of-Flight і FlightSense
X-NUCLEO-CCA02M1	Плата розширення з мікрофоном на базі MP34DT01-M
X-NUCLEO-IDB05A1	Плата розширення з Bluetooth-модулем SPBTLE-RF
X-NUCLEO-NFC04A1	Плата розширення з NFC / RFID-міткою на основі ST25DV04K
X-NUCLEO-PLM01A1	Плата розширення на основі ST7580
X-NUCLEO-GNSS1A1	Плата розширення з модулем навігації GNSS
X-NUCLEO-IHM01A1	Плата розширення для управління кроковим двигуном. На основі мікросхеми L6474
X-NUCLEO-IHM02A1	Плата розширення на основі драйвера L6470 двухфазного біполярного крокового двигуна з підтримкою мікрошаговий режиму.
X-NUCLEO-IHM03A1	Плата драйвера потужного крокового двигуна на основі системи в корпусі (SiP) powerSTEP01
X-NUCLEO-IHM04A1	Плата розширення для управління кроковим двигуном. Базується на основі мікросхеми L6206
X-NUCLEO-IHM07M1	Трифазний драйвер для BLDC / PMSM-двигунів
X-NUCLEO-IHM06A1	Плата розширення на основі мікросхеми низьковольтного драйвера крокового двигуна STSPIN220
X-NUCLEO-IHM05A1	Плата розширення для управління кроковим двигуном. Базується на основі мікросхеми L6208
X-NUCLEO-IHM12A1	Плата розширення на основі мікросхеми низьковольтного драйвера крокового двигуна STSPIN240

Таблица 3.1 (продовження) – Плати розширення для відлагоджувальних плат сімейства Nucleo

X-NUCLEO-IKA01A1	Мультифункціональна плата розширення, заснована на операційних підсилювачах
X-NUCLEO-CCA01M1	Плата розширення на основі мікросхеми високоефективної цифрової аудіосистеми STA350BW.
X-NUCLEO-LED61A1	Плата розширення, що була розроблена в якості прикладу використання чіпа LED6001 в компактному драйвері світлодіодів.

Відлагоджувальні плати Nucleo є найбільш ергономічним рішенням для побудови прототипу пристрою або вивчення роботи мікроконтролера, однак необхідність покупки додаткових модулів, а також обмежені можливості самих плат часто змушують користувачів звернутися до інших рішень, що включають в себе більш широкий набір вбудованої периферії, наприклад плати серії Discovery .

3.3 Відлагоджувальні плати Discovery

Відлагоджувальні плати сімейства DISCOVERY (рисунок 3.4) для 32-бітних мікроконтролерів мають необхідну інфраструктуру, що дозволяє демонструвати специфічні характеристики пристроїв, а HAL бібліотека і комплекс програмних прикладів дозволяють скористатися всіма перевагами виробів. Роз'єми розширення, встановлені на платах, відкривають доступ до більшості виводів I/O мікроконтролера і роблять можливим функціональне розширення за рахунок підключення додаткових плат [22].

Рисунок 3.4 – Відлагоджувальні плати сімейства DISCOVERY

В таблиці 3.2 наведені найменування відлагоджувальних плат і типи мікроконтролерів на основі яких випускаються відповідні налагоджувальні плати сімейства STM32 Discovery [22].

Таблица 3.2 – Отладочные платы семейства STM32 Discovery

Найменування відлагоджувальної плати	Мікроконтролер	ARM ядро	Flash, кБ	SRAM, кБ	Fmax, МГц
STM32F0308-DISCO	STM32F030R8T6	Cortex-M0	64	8	48
STM32F072B-DISCO	STM32F072RBT6	Cortex-M0	128	16	48
STM32F0DISCOVERY	STM32F051R8T6	Cortex-M0	64	8	32
STM32L0538-DISCO	STM32L053C8T6	Cortex-M0+	64	8	32
STM32L100C-DISCO	STM32L100RCT6	Cortex-M3	256	16	32
STM32L152C-DISCO	STM32L152RBT6	Cortex-M3	128	16	32
STM32VLDISCOVERY	STM32F100RBT6B	Cortex-M3	128	8	24
STM32F3348-DISCO	STM32F334C8T6	Cortex-M4	64	16	72
STM32F411E-DISCO	STM32F411VET6	Cortex-M4	512	128	100
STM32F429I-DISCO	STM32F429ZIT6	Cortex-M4	2048	256	180
STM32F469I-DISCO	STM32F469NIH6	Cortex-M4	2048	384	180
STM32L476G-DISCO	STM32L476VGT6	Cortex-M4	1024	128	80
STM32F3DISCOVERY	STM32F303VCT6	Cortex-M4	256	48	72
STM32F4DISCOVERY	STM32F407VGT6	Cortex-M4	1024	192	168
STM32F746G-DISCO	STM32F746NGH6	Cortex-M7	1024	340	216

Серія Discovery, як і Nucleo, має в своєму складі програматор ST-LINK, інтегрований в саму відлагоджувальну плату, який може бути використаний для програмування зовнішніх пристроїв на базі контролерів STM.

Сімейство плат підтримується багатьма широковідомими інтегрованими

середовищами розробки – IAR Embedded Workbench, MDK-ARM (Keil), TrueStudio (Atollic) та іншими.

3.4 Відлагоджувальні плати Evaluation Boards

Відлагоджувальні плати серії Evaluation Boards (рисунок 3.5) перевершують попередню серію в розмаїтті периферії. Якщо плати Discovery мають в своєму складі лише стартовий набір, необхідний для початкового ознайомлення і відлагодження роботи мікроконтролера, то плати Evaluation Boards значно куди більшим переліком додаткових компонентів в число яких входять:

- програматор ST-LINK;
- дисплеї, побудовані за технологіями TFT, E-Ink і PKI;
- інтерфейси CAN, USB, Ethernet і т.д .;
- різні роз'єми для підключення додаткової периферії, світлодіоди, датчики та інше.

Рисунок 3.5 – Отладочные платы серии Evaluation Boards

На даний момент розробникам доступно 27 варіантів відлагоджувальних плат Evaluation Boards, побудованих на основі мікроконтролерів STM32 всіх існуючих сімейств, починаючи від економною серії STM32L і закінчуючи

флагманом STM32H7. Серед останніх новинок [21] слід виділити:

– STM32L4R9I-EVAL – відлагоджувальна плата на базі контролера STM32L4R9AI. Мікроконтролер з лінійки STM32L4+ має наднизьке споживання енергії, здатен працювати з частотою до 120 МГц, а також має 2 Мбайт Flash і 640 кбайт RAM-пам'яті. На платі розташований 1,2 дюймовий LCD-дисплей круглої форми дозволом 390 × 390 пікселів, що працює по MIPI DSI-інтерфейсу, а також 4,3 дюймовий TFT LCD-дисплей роздільною здатністю 480 × 272 пікселів з режимом RGB. Також на платі розташовані 2x ST-MEMS цифрові мікрофони, слот під microSD, підтримка інтерфейсу I2C, порт RS-232, USB OTG FS Micro-AB порт, CAN 2.0A/B-сумісний порт і ін.

– STM32F769I-EVAL – відлагоджувальна плата на базі контролера STM32F769NI. Мікроконтролер знаходиться в лінійці STM32F7 і має високу продуктивність, реалізований в корпусі BGA216 і здатний працювати з частотою до 216 МГц, а також має 2 Мбайт Flash і 512 + 16 + 4 кбайт RAM-пам'яті. На платі розташований 4-дюймовий сенсорний LCD дисплей, що працює за MIPI DSI-інтерфейсу. Плата також включає в себе наступні інтерфейси: 4 I2C, 6 SPI, SDIO, 2 SAI, інтерфейс 8-14-бітного модуля цифрової камери, Ethernet MAC, FMC і Quid-SPI. Вона також має 4 USART і 4 UART, 2 CAN шини, 3 12-бітних АЦП, 2 12-бітних каналу ЦАП, RS-232 і т.д.

– STM32H743I-EVAL – відлагоджувальна плата на базі контролера STM32H743XI. Мікроконтролер є на сьогоднішній день флагманом серед всієї лінійки STM32, здатний працювати з частотою до 400 МГц, а також має 2 Мбайт Flash і 1 Мбайт RAM-пам'яті. На платі розташований 5,7 дюймовий TFT-дисплей із сенсорною панеллю і роздільною здатністю 640 × 480 пікселів; Плата має 3 USB з Micro-AB роз'ємами, RS-232, Ethernet RJ45, FD-CAN роз'ємом, роз'єм для підключення динаміків і мікрофона, слот для microSD карти, JTAG / SWD і ETM і т.д.

Аналогічно до попередньої серії Discovery, відлагоджувальні плати Evaluation Boards можуть використовуватися автономно, без підключення додаткових модулів.

Застосовуючи відлагоджувальні плати STMicroelectronics для своїх рішень, користувачі отримують доступ не тільки до апаратних засобів розробки, а й до широкого спектру супровідної інформації, до якого входять [21]:

- принципову схему відлагоджувальної плати Gerber-файли та специфікацію;
- керівництво користувача та приклади використання плати;
- безкоштовні та вільно розповсюджувані програмні засоби розробки від STMicroelectronics;
- перелік інформації створений самими користувачами, що знаходиться у відкритому доступі.

Однак для повноцінного використання відлагоджувальних плат насамперед потрібно створити програмний код, який би включав в себе алгоритм роботи контролера і його взаємодії з периферією плати або, як мінімум, завантажити такий код з представлених прикладів для первинного ознайомлення. І в тому і в іншому випадку необхідна наявність спеціалізованого програмного забезпечення, і тут великим плюсом є можливість застосування безкоштовних засобів розробки програмного коду від компанії STMicroelectronics (STM32CubeMX, STM Studio і т.д.). Звичайно, існують і інші середовища розробки програмного забезпечення, наприклад, Keil, IAR і т.д., однак ліцензія на роботу з ними коштує кілька тисяч доларів. У цьому сенсі компанія STMicroelectronics разом з партнерами суттєво спрощують життя незалежним компаніям зі малим бюджетом.

3.5 Висновки

Компанія STMicroelectronics завоювала ринок мікроконтролерів, в основному, завдяки всебічній підтримці своєї продукції і користувачів, що використовують їх продукцію в своїх рішеннях. Мікроконтролери серії STM32 добре підходять як для створення прототипів, так і для використання їх в серії,

а різноманіття їх сімейств дозволяє підібрати для рішення найбільш оптимальний контролер, враховуючи параметри продуктивності, енергоспоживання, вартості і т.д.

Вибір мікроконтролера, а також ознайомлення з принципами його роботи багато в чому полегшуються завдяки відлагоджувальним платам компанії: плати Nucleo мають низьку вартість і дають користувачеві можливість створити найбільш оптимальне рішення за допомогою додаткових модулів, плати Discovery добре підходять для ознайомлення і відлагодження початкових реалізацій і мають в своєму складі всю необхідну для цього периферію; якщо ж периферії плат Discovery недостатньо, на допомогу приходять Evaluation Boards, здатні задовольнити вимоги самих вибагливих розробників.

Безкоштовне програмне забезпечення також значно спрощує процес створення рішень на базі контролерів STM32, а широкий спектр документації, зменшує інформаційні труднощі освоєння контролерів на первинних етапах до мінімуму.

Аналіз розглянутих видів відлагоджувальних плат показав доцільність використання в програмно-апаратному комутаторі, що проектується, інтерфейсів вбудованих систем відлагоджувальної плати з сімейства Nucleo-144 з мікроконтролером STM32F7.

Рисунок 3.6 – Відлагоджувальна плата Nucleo F767-ZI

Такий вибір пояснюється низькою вартістю відлагоджувальної плати, продуктивним мікроконтролером, а також тим, що розробнику доступно безліч невикористаних виводів мікроконтролера, чого немає в платах Discovery цієї серії. Таким чином, для проектування ПО будемо використовувати відлагоджувальну плату Nucleo F767-ZI (рисунок 3.6).

4 ВИБІР СЕРЕДОВИЩА РОЗРОБКИ ПРИСТРОЮ

4.1 Огляд середовищ розробки

До найбільш відомих середовищ розробки можна віднести наступні [23]:

- Eclipse;
- TASKING ;
- Keil;
- IAR Embedded Workbench;
- Atollic TrueSTUDIO;
- Arduino;
- CooCox IDE.

Розглянемо їх основні характеристики і сфери застосування.

4.1.1 Середовище розробки Eclipse

Eclipse – вільне інтегроване середовище розробки модульних кроссплатформенних додатків.

Eclips є в першу чергу платформою для розробки розширеннь, чим він і завоював популярність. Будь-який розробник може розширити Eclipse своїми модулями. Безліч розширень доповнює середу Eclipse диспетчерами для роботи з базами даних, серверами додатків і ін. [23].

Eclipse написана на Java, тому є платформо-незалежним продуктом, за винятком бібліотеки SWT (Standard Widget Toolkit), яка розробляється для всіх поширених платформ. Бібліотека SWT використовується замість стандартної для Java бібліотеки Swing. Вона повністю базується на платформі більш низького рівня (операційну систему), що забезпечує швидкість і натуральний зовнішній вигляд призначеного для користувача інтерфейсу, але іноді викликає на різних платформах проблеми сумісності і стабільності додатків [23].

Основою Eclipse є платформа розширеного клієнта RCP (Rich Client Platform). В неї входять наступні компоненти [23]:

- ядро платформи (завантаження Eclipse, запуск модулів);
- OSGi (Open Services Gateway Initiative) (стандартне середовищі поставки комплектів;
- SWT (портуємий інструментарій віджетів);
- JFace (файлові буфери, робота з текстом, текстові редактори);
- робоче середовище Eclipse (панелі, редактори, проекції, майстри).

Графічний інтерфейс GUI (graphical user interface) в Eclipse написаний з використанням інструментарію SWT. Останній, на відміну від Swing, використовує графічні компоненти даної операційної системи. Інтерфейс Eclipse також залежить від проміжного шару GUI, названого JFace, який спрощує побудову призначеного для користувача інтерфейсу, що базується на SWT.

Гнучкість Eclipse забезпечується за рахунок модулів, завдяки чому можлива розробка не тільки на Java, але і на інших мовах, таких, як C/C++ та інших.

4.1.2 Середовище розробки TASKING

Програма TASKING австралійської компанії Altium Limited -середовище розробки та програмування сигнальних процесорів, 8-, 16-, 32-бітних мікропроцесорів і мікроконтролерів з підтримкою повноцінної арифметики з плаваючою точкою, видачею оптимального коду з найменшими часовими витратами [23].

Середовище розробки має такі переваги:

- в GUI інтерфейсі вікна відлагоджувач CrossView Pro з'являються в міру необхідності;
- добре розвинена і продумана структура формування проектів. У вікні Project Space міститься п'ять папок: Source files, Header files, Project files, Resources і Other files, в які автоматично розподіляються створені вами фай-ли;
- GUI інтерфейс є абсолютно налаштовуваним для створення

необхідного меню, для додавання в нього необхідних опцій і панелей інструментів. Все середовище розробки може бути налаштовано відповідно до вимог.

4.1.3 Середовище розробки Keil

Середовище розробки Keil, є набором утиліт для виконання повного комплексу заходів з написання програмного забезпечення для мікроконтролерів.

Keil дозволяє працювати з проектами будь-якого ступеня складності, починаючи з введення і редагування коду і закінчуючи внутрішньосхемним налагодженням коду і програмуванням ПЗП мікроконтролера. Від розробника прихована велика частина другорядних функцій, що сильно розвантажує інтерфейс і робить управління інтуїтивно зрозумілим. Однак при зростанні складності реалізованих завдань, завжди можна задіяти весь потенціал модулів, що функціонують під управлінням єдиної оболонки [23].

Програма працює на персональних комп'ютерах під управлінням тільки операційної системи Windows версій 2000, XP, Vista і 7. Програма Keil є платною і коштує дуже великих грошей.

4.1.4 Середовище розробки IAR Embedded Workbench

Середовище розробки IAR Embedded Workbench це багатофункціональне середовище розробки додатків на мовах C, C++ і асемблері для цілого ряду мікроконтролерів від різних виробників.

Основні переваги пакету – дружній інтерфейс користувача і неперевершена оптимізація генерованого коду. Крім цього реалізована підтримка різних операційних систем реального часу і JTAG-адаптерів сторонніх компаній.

На даний момент IAR Embedded Workbench підтримує роботу з 8-, 16-, 32-розрядними мікроконтролерами від Atmel, ARM, NEC, Infineon, Analog Devices, Cypress, Microchip Technologies, Micronas, Dallas Semiconductor/Maxim, Ember, Luminary, NXP, OKI, Samsung, National Semiconductor, Texas Instruments,

STMicroelectronics, Freescale, TI/Chipcon, Silicon Labs и Renesas. Для кожної платформи існує своє середовище розробки, зокрема ARM мікроконтролерів відповідає версія пакету IAR Embedded Workbench for ARM [23].

Інтегрована система допомоги полегшує написання програм в даного середовищі. Передбачено взаємодія з утилітою AVR Studio. Крім іншого в IAR Embedded Workbench існує можливість самостійного управління оптимізацією окремих модулів проекту, що спрощує процес налагодження, а також дозволяє прискорити роботу програми або заощадити на пам'яті [23].

Працює середовище під управлінням тільки операційної системи Microsoft Windows версій 95, 98, NT, 2000, XP, Vista, 7 (не має значення 32-х або 64-бітних).

4.1.5 Середовище розробки Atollic TrueSTUDIO

Інтегроване середовище розробки програм для ARM-процесорів, що включає в себе GNU компілятор і відлагоджувач.

Середовище розробки Atollic TrueSTUDIO була створена на базі популярної платформи з відкритим вихідним кодом – Eclipse.

Atollic TrueSTUDIO підтримує безліч ARM-ядер (сімейств ARM9, ARM7, Cortex-A, Cortex-R, Cortex-M і т.д.) від таких відомих компаній, як Atmel, Infineon, Freescale, NXP, Silicon Labs, Renesas, Spansion, STMicroelectronics, Toshiba, Texas Instruments і деяких інших. Також є підтримка 2-ядерних і багатопроцесорних пристроїв [23].

Середовище розробки взаємодіє з відлагоджувачами ST-LINK і ST-LINK/V2 від STMicroelectronics, SAM-ICE від Atmel, Segger J-Link, OSJTAG і P&E Multilink.

Програмне забезпечення було розроблено для операційних систем сімейства Microsoft Windows – XP, Vista и 7.

4.1.6 Середовище розробки Arduino

Arduino – торгова марка апаратно-програмних засобів для побудови простих систем автоматики і робототехніки, орієнтована на непрофесійних користувачів. Програмна частина складається з безкоштовної програмної оболонки (IDE) для написання програм, їх компіляції та програмування апаратури. Апаратна частина являє собою набір змонтованих друкованих плат, що продаються як офіційним виробником, так і сторонніми виробниками. Повністю відкрита архітектура системи дозволяє вільно копіювати або доповнювати лінійку продукції Arduino [23].

Мікроконтролери для Arduino відрізняються наявністю попередньо прошитого в них завантажувача (bootloader). За допомогою цього завантажувача користувач завантажує свою програму в мікроконтролер без використання традиційних окремих апаратних программаторов. Завантажувач з'єднується з комп'ютером через інтерфейс USB (якщо він є на платі) або за допомогою окремого перехідника UART-USB. Підтримка завантажувача вбудована в Arduino IDE і виконується в один клік [23].

У лінійці пристроїв Arduino в основному застосовуються мікроконтролери Atmel AVR ATmega328, ATmega168, ATmega2560, ATmega32U4, ATTiny85 с частотою тактування 16 або 8 МГц. У старих виробах застосовувалися ATmega8, ATmega1280 і інші.

Мовою програмування Arduino є стандартний C++ (використовується компілятор AVR-GCC) з деякими особливостями, що полегшують для новачків написання першої працюючої програми.

4.1.7 Середовище розробки CooCox CoIDE

Високоінтегроване програмне середовище CooCox CoIDE, призначене для розробки коду мікроконтролерів архітектури ARM.

CooCox CoIDE є одним з найпростіших і швидких в плані установки, освоєння і налаштування рішень, що дозволяє навіть починаючим користувачам домагатися в ньому істотних результатів. Успішний старт

перших проектів забезпечує майстер, що допомагає пройти через всі основні етапи розробки шляхом відповідей на прості запитання. Якісно зроблене середовище CooCox CoIDE дозволяє завантажувати код програми, редагувати його, проводити компіляцію (сторонніми засобами), прошивати контролер і проводити відлагодження [23].

Програма заснована на базі Eclipse і має всі її переваги. Редактор коду включає в себе підсвічування синтаксису і вспливаючі підказки. Присутні функції глобальної заміни змінної та пропозиції варіантів закінчення коду. Середовище підтримує мікроконтролери серії ST, а також ряд інших сімейств: Atmel, Holtek, Freescale, Nuvoton, NXP, Energy Micro, Texas Instruments і деякі інші. Список чіпів постійно збільшується з кожною версією програми. Вбудований відлагоджувач ST-Link підтримує всі основні режими відлагодження [23].

Робоча платформа розглянутої середовища розробки – операційні системи Windows XP (необхідний SP3), Vista (SP2), 7. Крім цього, можлива робота з програмою в середовищі Linux за допомогою Wine (вільне програмне забезпечення, що дозволяє користувачам UNIX-подібних систем виконувати 16-, 32- і 64- бітові додатки Microsoft Windows.

Дане середовище розробки абсолютно безкоштовна і має відкритий код.

Одним з недоліків CooCox CoIDE варто відзначити відсутність компілятора GCC (GNU Compiler Collection), який потрібно завантажити і встановити окремо.

4.2 Висновки

Огляд і аналіз інтегрованих середовищ розробки показав, що для проектування електронних пристроїв на базі мікроконтролерів ARM- рекомендовано застосування середовища розробки IAR Embedded Workbench.

Вибір наведеної середовища розробки заснований на наступних характеристиках [23]:

- якісна оптимізація коду компілятором;
- підтримується налагодження ОСРЧ;
- дозволяють використовувати налагодження;
- мова програмування є стандартним C++;
- багато готових прикладів, які можна завантажувати в середу розробки.

5 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМУТАТОРА ПЕРИФЕРІЙНИХ ІНТЕРФЕЙСІВ

Метою даного розділу є розробити ПО універсального пристрою зв'язку з периферійними пристроями. Для цього необхідно:

- Вибрати інтерфейс сполучення з контролером верхнього рівня і розробити протокол сполучення;
- Розробити структуру ПО, а саме, яким чином планується реалізувати функціонал зв'язку з великою кількістю периферійних пристроїв на обраної відлагоджувальній платі;
- Розробити емулятори ПО для тестування і налагодження цільового ПО;
- Описати процес розробки ПО.

5.1 Вибір інтерфейсу зв'язку з контролером верхнього рівня.

Перед початком будь-якої розробки необхідно визначитися з деякими базовими моментами, від яких ми будемо відштовхуватися при проектуванні пристрою. Для початку виберемо інтерфейсу зв'язку з контролером верхнього рівня. Існує велика кількість різних інтерфейсів зв'язку. Основними промисловими інтерфейсами є: CAN, RS-485 і Ethernet. Коротко розглянемо кожен з них.

CAN – стандарт промислового мережі, орієнтований, перш за все, на об'єднання в єдину мережу різних виконавчих пристроїв і датчиків. Режим передачі – послідовний, широкомовний, пакетний. Стандарт розроблений компанією RobertBoschGmbH в середині 1980-х і в даний час широко поширений в промисловій автоматизації, технології «розумного будинку», автомобільної промисловості та багатьох інших областях. Стандарт для автомобільної автоматики [24].

Як було зазначено вище, даний стандарт є промисловим, проте він не володіє широкою підтримкою на споживчому ринку домашньої електроніки, що, безумовно, є серйозним мінусом.

RS-485 – стандарт фізичного рівня для асинхронного інтерфейсу. Регламентує електричні параметри повнодуплексної диференціальної лінії зв'язку типу «загальна шина» що підтримує множинні підключення. Стандарт набув великої популярності і став основою для створення цілого сімейства промислових мереж, широко використовуваних в промисловій автоматизації [25].

На жаль, даний стандарт не володіє широкою підтримкою на споживчому ринку домашньої електроніки, що, безумовно, є серйозним мінусом. Інтерфейс RS-485 найчастіше має на увазі короткі посилки і стандартні протоколи зв'язку. Використання даного інтерфейсу не має на увазі паралельну роботу з безліччю пристроїв на одній шині, оскільки канальний рівень практично всіх протоколів не підтримує запобігання колізій зв'язку. Ця умова обмежує використання RS-485 при розробці нашого пристрою, оскільки планується паралельна робота з безліччю пристроїв.

Ethernet – сімейство технологій пакетної передачі даних між пристроями для комп'ютерних і промислових мереж. Стандарти Ethernet визначають дротяні з'єднання і електричні сигнали на фізичному рівні, формат кадру та протоколи управління доступом до середовища – на канальному рівні моделі OSI. Ethernet в основному описується стандартами IEEE групи 802.3. Ethernet став найпоширенішою технологією LAN в середині 1990-х років, і залишається найпопулярнішою і поширенішою і нині.

У даній роботі буде застосовуватися саме вона, через поширеності даної технології, оскільки таке рішення дозволити використовувати пристрій простим користувачам без використання додаткових перетворювачів або ж комутаторів. Оскільки Ethernet протоколів CSMA/CD, то можлива робота з декількома різними пристроями, підключеними по одному каналу Ethernet.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection – множинний доступ з прослуховуванням несучої і виявленням колізій) – технологія множинного доступу до загального передавального середовища в локальній комп'ютерній мережі з контролем колізій. CSMA/CD відноситься до децентралізованих випадковим методам. Він використовується як в звичайних мережах типу Ethernet, так і в високошвидкісних мережах (Fast Ethernet, Gigabit Ethernet).

Так само називають мережевий протокол, в якому використовується схема CSMA/CD. Протокол CSMA/CD працює на канальному рівні в моделі OSI [26].

Наведені вище аргументи не означає, що даній пристрій взагалі не буде підтримувати роботу з контролерами верхнього рівня по інтерфейсах RS-485 і CAN. Даний функціонал планується реалізовувати, пізніше, в процесі розвитку даного програмного продукту.

5.2 Розробка протоколу зв'язку з контролером верхнього рівня.

Безумовно, мікроконтролер повинен якимось чином зв'язуватися і взаємодіяти іншому, для цього ж необхідно використовувати якийсь уніфікований протокол зв'язку. На жаль, під час виконання даної роботи не було знайдено готових проток, тому будемо розробляти його самостійно. Для початку розробки протоколу, необхідно визначитися в якому форматі буде здійснювати зв'язок, і на підставі цього можна буде виробляти подальшу розробку.

В основному ми можемо організувати зв'язок в цифровому, символьному чи ж в текстовому форматі. Оскільки в даній роботі немає особливих вимог щодо швидкості передачі інформації і пропускної здатності каналу, має сенс для більшої наочності використовувати текстовий формат обміну даними. З основних же текстових форматів обміну даними можна виділити два JSON та XML. Розглянемо кожен з них.

JSON (JavaScript Object Notation) – текстовий формат обміну даними, заснований на JavaScript. Як і багато інших текстові формати, JSON легко читається людьми. .JSON-текст являє собою (в закодованому вигляді) одну з двох структур:

- Набір пар ключ: значення. У різних мовах це реалізовано як запис, структура, словник, хеш-таблиця, список з ключем або асоціативний масив. Ключем може бути тільки рядок (чутливі до регістру, – імена з буквами в різних регістрах вважаються різними), значенням – будь-яка форма;
- Впорядкований набір значень. У багатьох мовах програмування це реалізовано як масив, вектор, список або послідовність.

Структури даних, що використовуються JSON, підтримуються будь-яким сучасною мовою програмування, що і дозволяє застосовувати JSON для обміну даними між різними мовами програмування і програмними системами [27].

XML (eXtensible Markup Language) – розширювана мова розмітки. Специфікація XML описує XML-документи і частково описує поведінку XML-процесорів (програм, які читають XML-документи і забезпечують доступ до їх вмісту). XML розроблявся як мова з простим формальним синтаксисом, зручний для створення і обробки документів програмами і одночасно зручний для читання і створення документів людиною, з підкресленням націленості на використання в Інтернеті. Мова називається розширюваним, оскільки ним не фіксується розмітку, яка використовується в документах: розробник вільний створити розмітку відповідно до потреб конкретної області, будучи обмеженим лише синтаксичними правилами мови. Розширення XML – це конкретна граматика, створена на базі XML і представлена словником тегів і їх атрибутів, а також набором правил, що визначають, які атрибути і елементи можуть входити до складу інших елементів. Поєднання простого формального синтаксису, зручності для людини, розширюваності, а також базування на кодуваннях Юнікод для подання змісту документів привело до широкого

використання як власне XML, так і безлічі похідних спеціалізованих мов на базі XML в найрізноманітніших програмних засобах [28].

Однак при всіх вищенаведених достоїнства реалізація повноцінної інтерпретації XML є надмірною для реалізації даної роботи, тому має сенс не використовувати XML, необхідно використовувати JSON.

Протокол обміну побудований таким чином. Клієнт (контролер верхнього рівня) надсилає запит на сервер (розроблювальний пристрій), після обробки запиту сервер повертає відповідь клієнту. Сервер повинен підтримувати команди конфігурації інтерфейсу, команди відправки даних по інтерфейсу і скидання інтерфейсу. Наприклад, команда на настройку інтерфейсу SPI-3 представляє собою наступне повідомлення:

```
{
   «interface» : «spi3»,
   «manage» : «config»,
   «speed» : «10500000»,
   «polar» : «0»,
   «bits» : «8»,
   «phase» : «first2
   «bit_order» :«msb»,
   «use_cs» :«off»
}
```

У рядку «interface», через двокрапку, вибирається, з яким інтерфейсом зараз будемо працювати. Команда «manage», визначає які саме дії, необхідно виконати інтерфейсу; в даному випадку настройка («config»). Далі йдуть специфічні для даного інтерфейсу команди налаштування. Якщо якась команда налаштвання не вказана в повідомленні, то її значення залишиться без змін (тобто, по-замовчуванню). Якщо команда написана неправильно то контролер проігнорує її значення. Якщо зазначені значення не вірні, то контролер застосує значення за замовчуванням для цієї команди і поверне повідомлення, що

інтерфейс налаштований частково. Повторним вичитуванням налаштувань інтерфейсу можливо зрозуміти, які саме команди не вдалося виконати.

Дізнатися налаштування можна відправивши пусте повідомлення налаштвання:

```
{
   «interface» : «spi3»,
   «manage» : «config»,
}
```

Якщо все налаштування введені правильно, то сервер поверне відповідь:

```
{
   «interface» : «spi3»,
   «manage» : «config»,
   «status» : «ok»,
}
```

Якщо неправильно вказана команда «interface», або обраний інтерфейс не існує, то контролер поверне відповідь:

```
{
   «status» : «wrong»,
}
```

Команда «manage» не є обов'язково командою, якщо вона буде опущена, то пристрій буде використовувати її значення за замовчуванням, – тобто буде намагатися відправити повідомлення («manage» : «none»).

Відправлення повідомлення відбувається наступним чином. Передається повідомлення з обов'язковим полем «interface» і настройками для даної сесії передачі. Ключ «data» означає дані, які необхідно передати. Дані інтерпретуються як рядки і передаються так само. наприклад:

```
{
   «interface» : «uart2»,
   «wait_data_us» : «10000»,
   «timeout_ms» : «500»,
```

«data» : «hello world»,

}

В даному випадку буде відправлено повідомлення «hello world», без лапок на інтерфейс UART2. налаштування:

- wait_data_us – максимальна затримка між байтами при очікуванні відповіді;
- timeout_ms – час очікування відповіді;
- data – дані для відправки.

У разі, отримання відповіді від периферійного пристрою буде повернута відповідь по ключу «response» зі статусом «ok». Для прикладу розглянемо відповідь «hello»:

{

«interface» : «uart2»,

«status» : «ok»,

«response» : «hello»

}

Якщо час очікування відповіді вийшов, а відповідь від периферійного пристрою не прийшла, то сервер поверне відповідь зі статусом «timeout»:

{

«interface» : «uart2»,

«status» : «timeout»,

}

Вимкнення інтерфейсу проводиться командою «reset», переданої по ключу «manage»:

{

«interface» : «spi3»,

«manage» : «reset»

}

Після вимкнення інтерфейсу сервер поверне відповідь:

{

```
    «interface» : «spi3»,
    «manage» : «reset»,
    «status» : «ok»
}
```

5.3 Проектування ПО

5.3.1 Мультиплексування периферійних інтерфейсів

Обрана нами відлагоджувальна плата підтримує безліч інтерфейсів UART, SPI, I2C, CAN і т.д. Однак не всі інтерфейси можна одночасно використовувати. Обмеження, в даному випадку, накладає те, що сигнальні ланцюга різних інтерфейсів можуть бути виведені на один і той же висновок мікросхеми. При цьому програмно необхідно вибрати, який це буде інтерфейс. Також обмеження на використання всіх інтерфейсів накладає те, що частина висновків мікроконтролера на відлагоджувальній платі спочатку задіяна на певні функції, наприклад USB, і оцінний UART.

Ще одне обмеження на використання всіх інтерфейсів – обмежена кількість DMA в контролері. Дане обмеження є апаратним, яке, однак, частково можна подолати програмними засобами, якщо для інтерфейсів з повільними швидкостями обміну не використовувати DMA. Тим не менше, використання процесора (без DMA) для передачі байт повільних інтерфейсів не бажано, оскільки збільшує загальну завантаження процесора і підвищує час реакції на більш пріоритетні завдання.

У даній роботі, ми обмежимося можливістю одночасного підключення до 8 периферійних пристроїв, – оскільки в використовуваному мікроконтролері 16 потоків DMA. Кожен інтерфейс використовує 2 потоки (stream) DMA – один для отримання даних (RX), інший для відправлення даних (TX). Цим пояснюється, що 16 потоків DMA, можуть обслуговувати лише 8 пристроїв.

Після аналізу документації відлагоджувальної плати [29] та мікроконтролера STM32F767ZI [30] прийшли до висновку, що основне

обмеження на одночасне використання інтерфейсів, накладає неможливість використовувати будь-який потік DMA, для будь-якого периферійного інтерфейсу. Те, що сигнальні ланцюга різних інтерфейсів можуть бути виведені на один і той же висновок, як згадувалося раніше, якраз не накладає обмежень по причині мультиплексування однієї периферії на безліч висновків МК. Таблиці сумісності DMA представлені на рисунках 5.1 і 5.2. Потоки DMA периферійних інтерфейсів виділені кольором.

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	SPI3_RX	SPDIFRX_DT	SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX	SPDIFRX_CS	SPI3_TX
Channel 1	I2C1_RX	I2C3_RX	TIM7_UP	-	TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
Channel 2	TIM4_CH1	-	I2C4 _RX	TIM4_CH2	-	I2C4 _RX	TIM4_UP	TIM4_CH3
Channel 3	-	TIM2_UP TIM2_CH3	I2C3_RX	-	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
Channel 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
Channel 5	UART8_TX	UART7_TX	TIM3_CH4 TIM3_UP	UART7_RX	TIM3_CH1 TIM3_TRIG	TIM3_CH2	UART8_RX	TIM3_CH3
Channel 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2	-	TIM5_UP	-
Channel 7	-	TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX
Channel 8	I2C3_TX	I2C4_RX	-	-	I2C2_TX	-	I2C4_TX	-
Channel 9	-	SPI2_RX	-	-	-	-	SPI2_TX	-

Риснок 5.1 – Таблиця підключення периферії до контролера DMA1

В даних таблицях, у верхньому рядку, написані назви потоків DMA контролера. Всього їх 8 для одного DMA, а оскільки в МК два контролера DMA, то на двох 16 потоків. Крайній лівий стовпець показує номери каналів, за якими периферія МК підключається до відповідного потоку DMA. Один потік може обслуговувати тільки одне периферійний пристрій, при цьому, як видно з рисників 5.1, 5.2, більшість периферії зв'язку МК може підключатися на 3-4 різних потоку, але деякі, тільки на 2. В зв'язку, якщо для периферії зв'язку

зайняті всі її потоки DMA, то не вийде її використовувати, навіть якщо інші потоки будуть вільні.

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	ADC1	SAI1_A	TIM8_CH1 TIM8_CH2 TIM8_CH3	SAI1_A	ADC1	SAI1_B	TIM1_CH1 TIM1_CH2 TIM1_CH3	SAI2_B
Channel 1	-	DCMI	ADC2	ADC2	SAI1_B	SPI6_TX	SPI6_RX	DCMI
Channel 2	ADC3	ADC3	-	SPI5_RX	SPI5_TX	CRYP_OUT	CRYP_IN	HASH_IN
Channel 3	SPI1_RX	-	SPI1_RX	SPI1_TX	SAI2_A	SPI1_TX	SAI2_B	QUADSPI
Channel 4	SPI4_RX	SPI4_TX	USART1_RX	SDMMC1	-	USART1_RX	SDMMC1	USART1_TX
Channel 5	-	USART6_RX	USART6_RX	SPI4_RX	SPI4_TX	-	USART6_TX	USART6_TX
Channel 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
Channel 7	-	TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX	SPI5_TX	TIM8_CH4 TIM8_TRIG TIM8_COM
Channel 8	DFSDM1_ FLT0	DFSDM1_ FLT1	DFSDM1_ FLT2	DFSDM1_ FLT3	DFSDM1_ FLT0	DFSDM1_ FLT1	DFSDM1_ FLT2	DFSDM1_ FLT3
Channel 9	JPEG_IN	JPEG_OUT	SPI4_TX	JPEG_IN	JPEG_OUT	SPI5_RX	-	-
Channel 10	SAI1_B	SAI2_B	SAI2_A	-	-	-	SAI1_A	-
Channel 11	SDMMC2	-	QUADSPI	-	-	SDMMC2	-	-

Рисунок 5.2 – Таблиця підключення периферії до контролера DMA2

Всі підтримувані інтерфейси і способи їх підключення на отладочной платі приведені в таблицях 5.1 – 5.3.

Таблиця 5.1 – Виводи відлагоджувальної плати Nucleo F767-ZI для підключення зовнішніх пристроїв по інтерфейсу UART.

Інтерфейс	RX	TX	RTS	CTS
UART1	PB15	PB6	використ.	використ
UART2	PA3	PD5	PD4	PD3
UART3	PD9	PD8	PD12	PD11
UART4	PD0	PA0	PA15	PB0
UART5	PD2	PC12	PC8	PC9
UART6	PC7	PC6	PG12	PG15
UART7	PE7	PE8	PE9	PE10
UART8	PE0	PE1	PD14	PD15

Таблиця 5.2 – Виводи відлагоджувальної плати Nucleo F767-ZI для підключення зовнішніх пристроїв по інтерфейсу SPI.

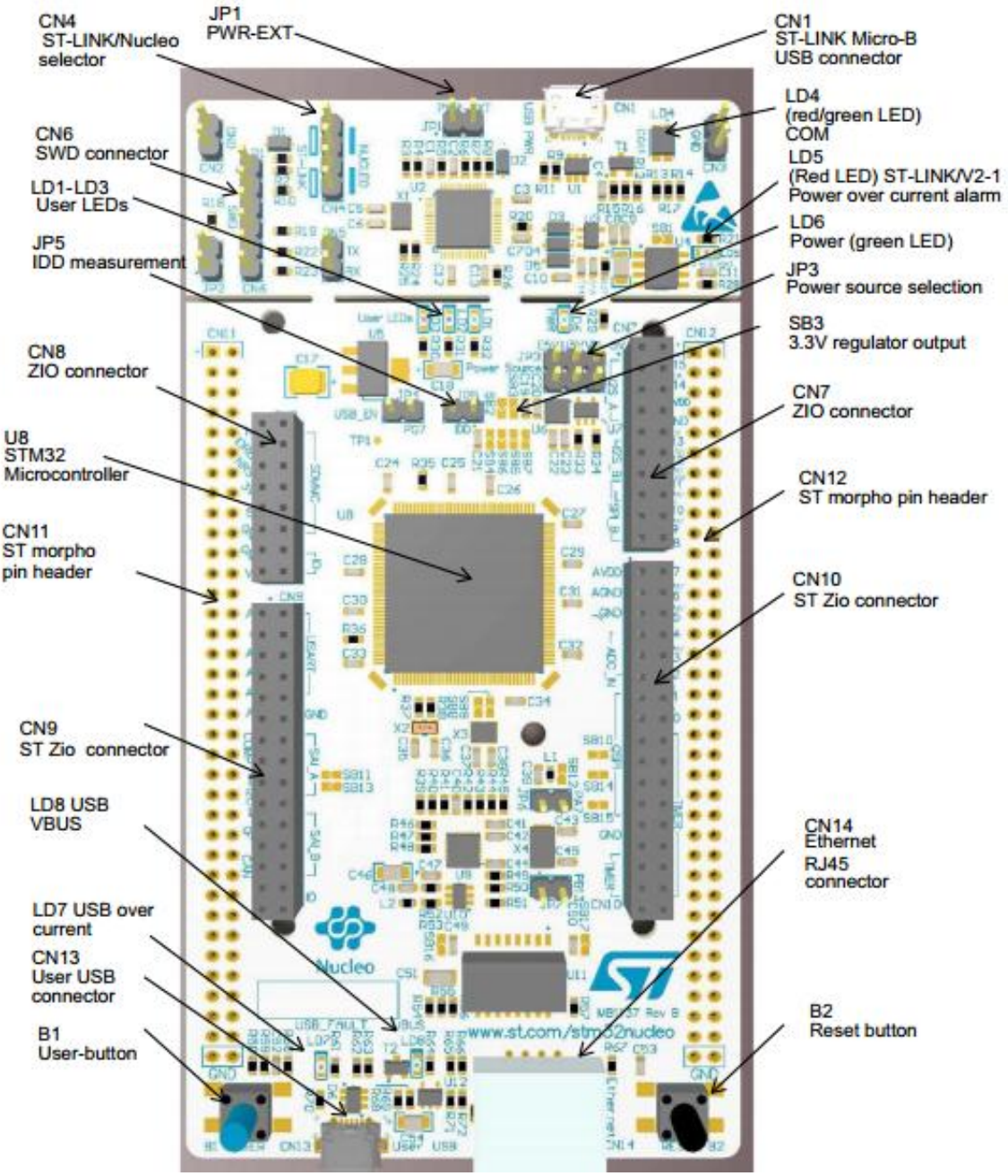
Інтерфейс	MOSI	MISO	SCK	CS
SPI1	PB5	PA6	PA5	PG10
SPI2	PC3	PC2	PB10	PB12
SPI3	PB2	PC11	PC10	PA4
SPI4	PE6	PE5	PE4	PE2
SPI5	PF9	PF8	PF7	PF6
SPI6	PG14	PA6	PA5	PG8

Таблиця 5.3 – Виводи відлагоджувальної плати Nucleo F767-ZI для підключення зовнішніх пристроїв по інтерфейсу I2C.

Інтерфейс	SDL	SCL
I2C1	PB9	PB8
I2C2	PF0	PF1
I2C3	використовується	використовується
I2C4	PF15	PF14

Таким чином одночасно можна підключити пристрої практично до всіх інтерфейсами, аби не обмеження DMA. Одночасно неможливо тільки підключити пристрої до інерфейсів SPI1 і SPI6, оскільки вони використовують загальне виводи МК – PA6 і PA5. Інтерфейс I2C3, а також сигнали RTS, CTS інтерфейсу UART1, використовуються на відлагоджувальній платі Nucleo F767-ZI для USB, тому немає можливості їх задіяти.

Фізичне розташування вищеописаних виводів на відлагоджувальній платі показано на рисунках 5.3, 5.4. Як видно з рисунка 5.4, все виводи мікроконтролера підключені до конекторів CN7 – CN10.



Риснок 5.4 – СхематичнезображенняNucleoF767-ZI

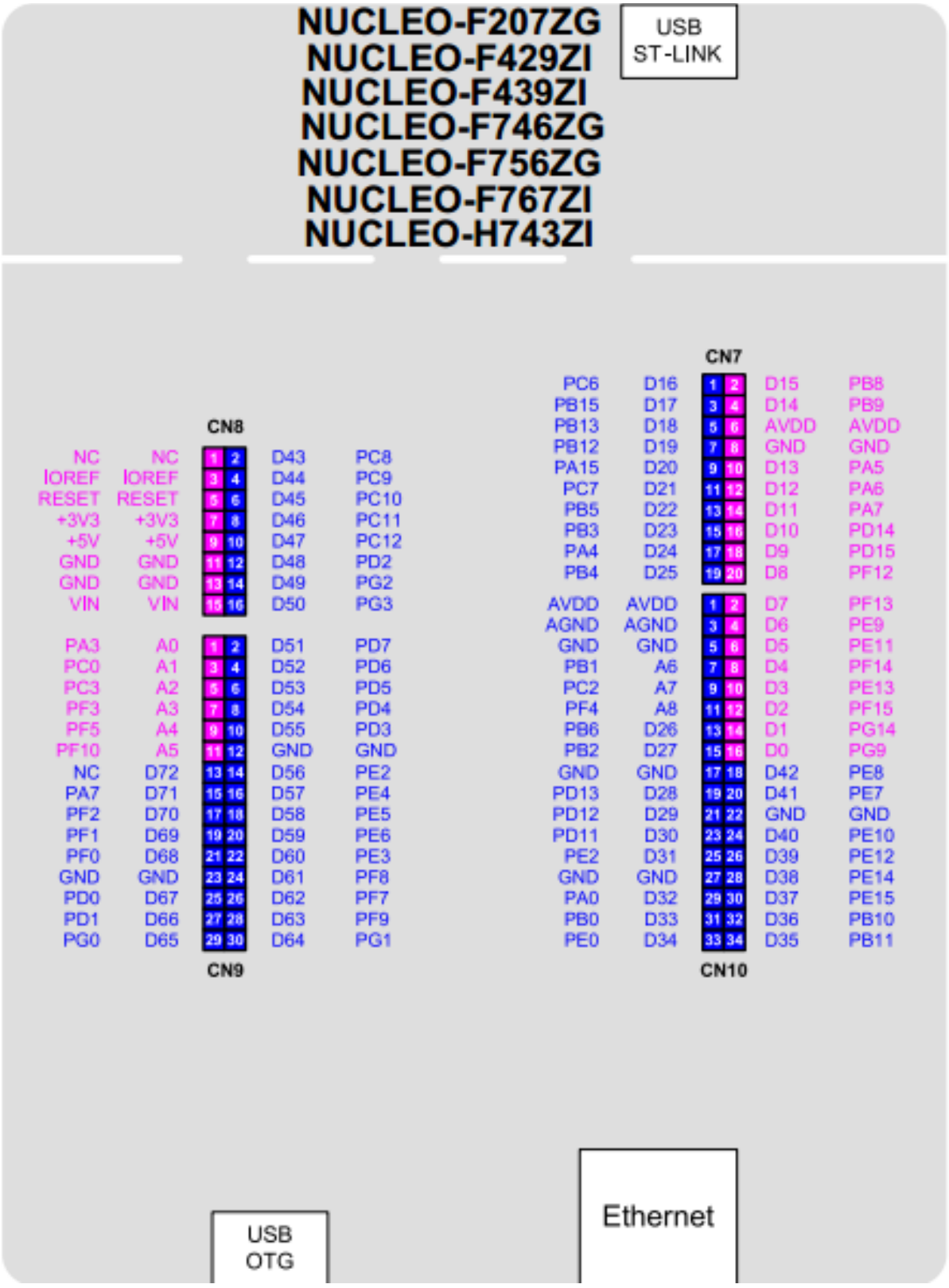


Рисунок 5.4 – Розташування інтерфейсів підключення мікроконтролера STM32F767 на відлагоджувальній платі NucleoF767-ZI

5.3.2 Розробка структури ПО

При побудові програмного забезпечення вбудованих систем користуються структурною схемою наведенню на рисунку 5.5.

Між ПЗ користувача і апаратною частиною мікроконтролера знаходяться драйвера пристроїв, потім API рівня апаратної абстракції (HAL), яке щільно може взаємодіяти зі стандартними бібліотеками мови програмування С.

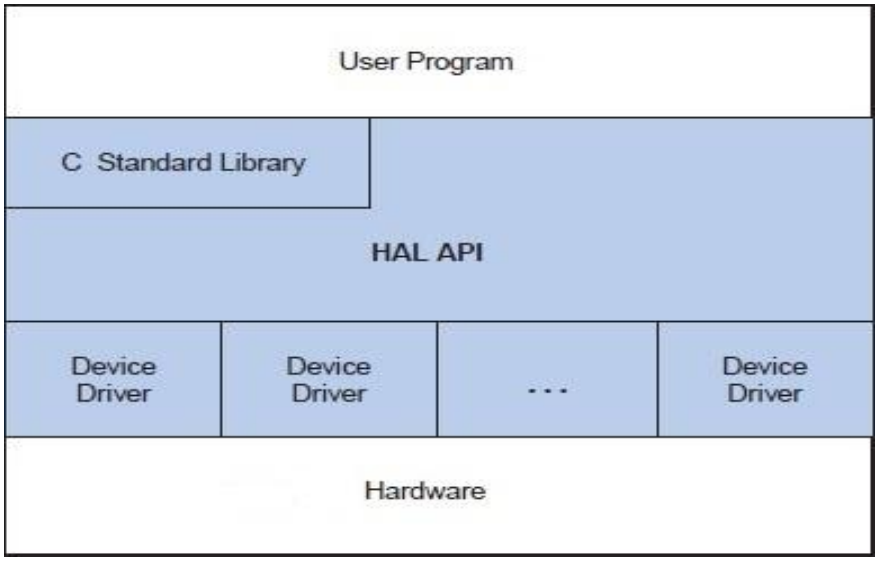


Рисунок 5.5 – Загальна структура вбудованого ПЗ

Для зниження часу розробки ми будемо використовувати драйвера пристроїв і HAL API надаються компанією STM. Для реалізації зв'язку через Ethernet будемо використовувати TCP/IP стек, розроблений Адамом Дункельсом – LwIP (lightweight IP) [31]. Даний стек спеціально оптимізований для роботи на обчислювачах вбудованих пристроїв.

Існують наступні способи побудови вбудованого ПО (рисунок 5.6):

- Моноцикл (Simple Loop, або простий цикл) – реалізується на рівні програмної завдання, без використання переривань.
- Обробка подій – програма обробляє асинхронні події та призупиняє свою роботу до виникнення наступної події.

– Комбінований спосіб – використовуються програмні завдання для основних обчислень і обробники переривань для обробки асинхронних подій. Можливо з використанням ОСРЧ, або без неї.

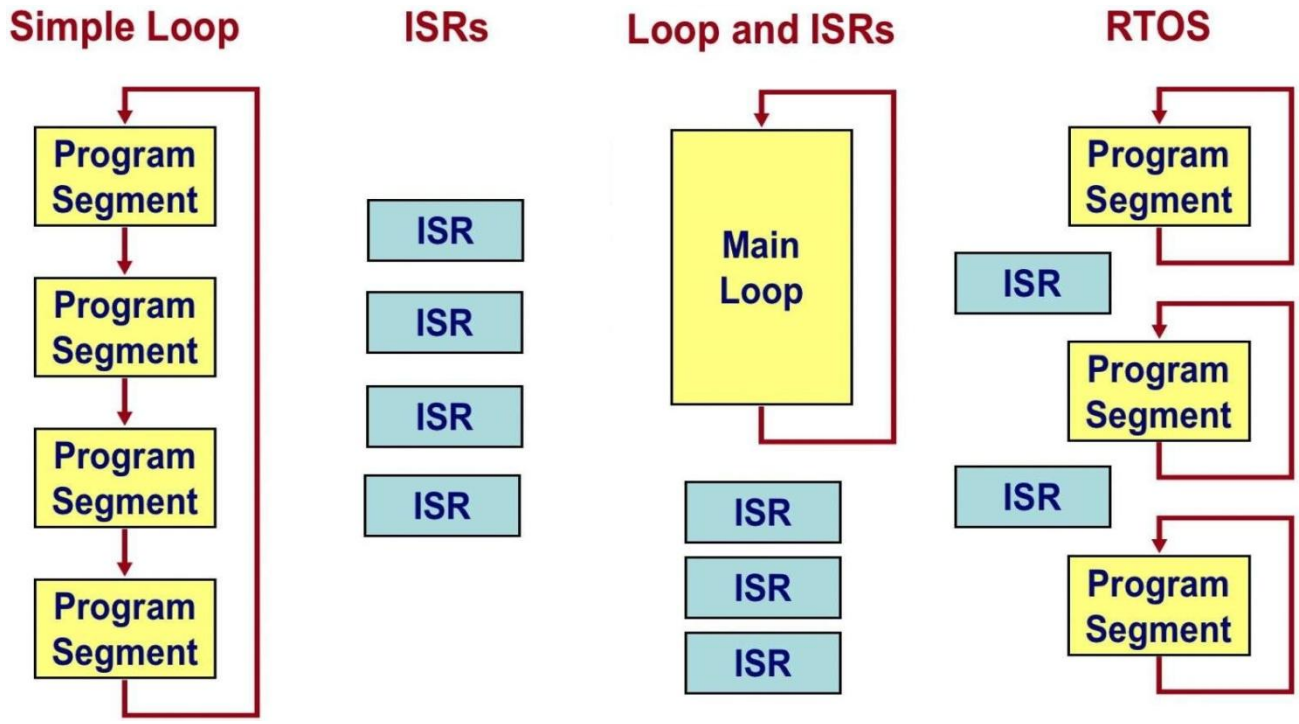


Рисунок 5.6 – Способи побудови вбудованого програмного забезпечення

Побудова програмного забезпечення комбінованим способом з використанням ОСРВ дає найбільші переваги, оскільки:

– прискорює розробку і налагодження ПЗ;
– збільшує модульність;
– знижує складність доопрацювання ПЗ;
– збільшує швидкість і стабільність роботи ПО за рахунок поділу праці та економії ресурсів процесора при роботі з периферією.

Однак є й недоліки використання ОСРВ:

– необхідні знання принципів роботи і інструментів ОСРВ, вміння проектувати і налагоджувати такі системи;
– додаткове використання ресурсів процесора для перемикання контексту між завданнями.

Для розробки ПО комутатора програмних інтерфейсів оптимально використовувати ОСРВ, оскільки це дозволяє побудувати систему обміну даними з мінімальним використанням процесорних ресурсів, а також максимальною швидкістю обробки асинхронних подій, якими є сеанси зв'язку. В якості ОСРЧ будемо використовувати FreeRTOS – багатозадачную операційну систему реального часу (ОСРЧ) для вбудованих систем. FreeRTOS портована на 35 мікропроцесорних архітектур. Поширюється під ліцензією MIT з 2017 року [32].

Обробка запити від клієнта (контролера верхнього рівня) складається з таких частин:

- прийом даних від клієнта;
- розпакування JSON даних;
- формування повідомлення;
- відправка периферійних пристроїв;
- очікування відповіді (при необхідності);
- формування JSON відповіді;
- відправка відповіді клієнту.

Алгоритм передачі повідомлення представлений на рисунках5.7, 5.8:

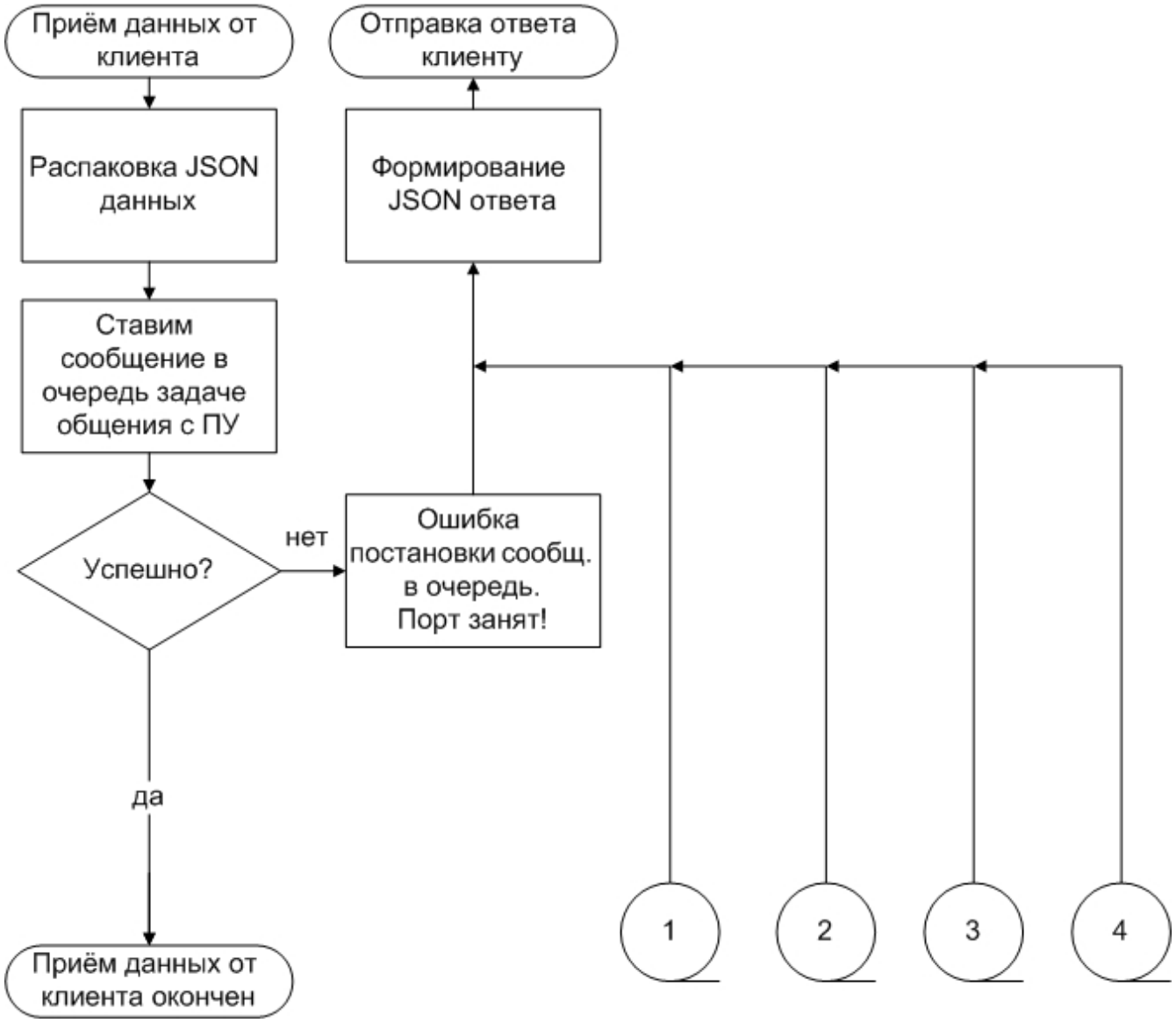


Рисунок 5.7 – Алгоритм передачі повідомлень між клієнтом і периферійним пристрій (початок)

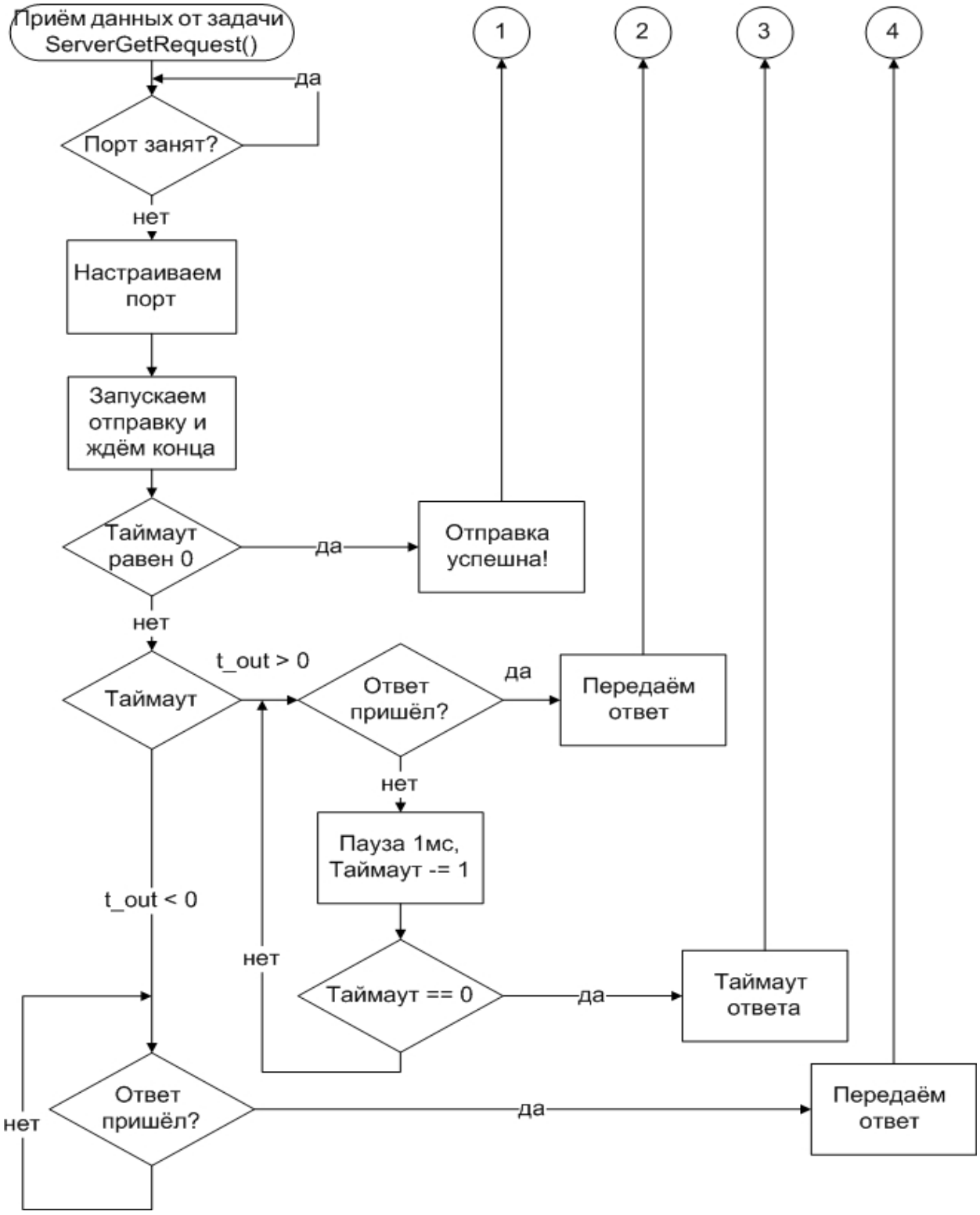


Рисунок 5.8 – Алгоритм передачі повідомлень між клієнтом і периферійним пристрій (закінчення)

Алгоритм передачі повідомлення також можна відобразити зі сторони структури ПО, тобто, які механізми ОСРЧ і апаратної частини мікроконтролера

необхідно використовувати, щоб дані передавались між клієнтом і периферійним пристроєм.

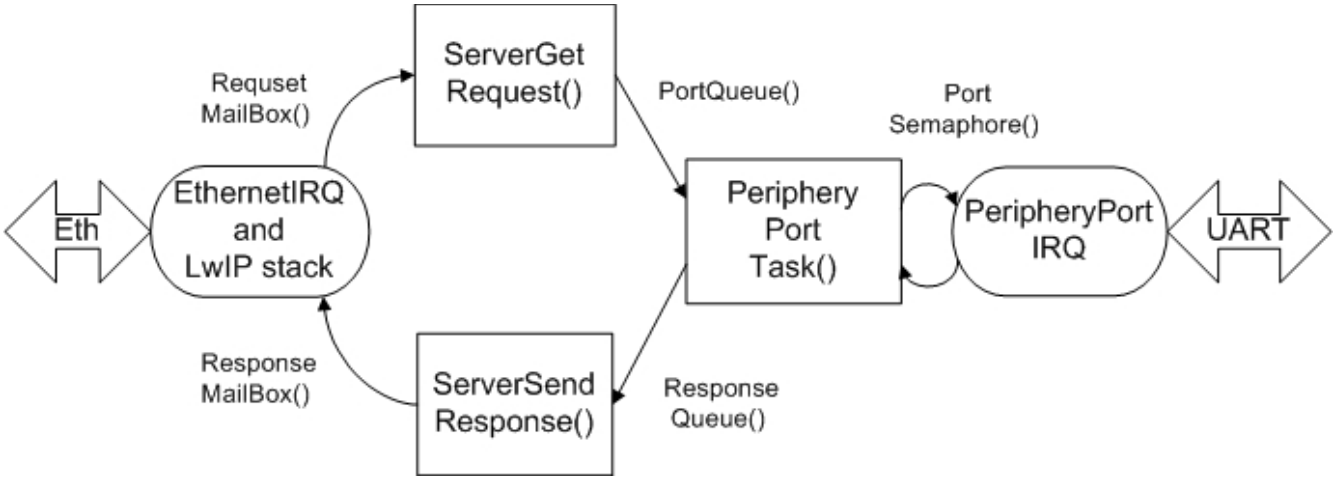


Рисунок 5.9 – Структура функціоналу обміну повідомленнями між клієнтом і периферійним пристроєм

На рисунку 5.9 представлена структура функціоналу передачі повідомлень. Наведемо компоненти цієї структури:

- Eth – вхідне повідомлення по каналу Ethernet від клієнта.
- EthrnetIRQ and LwIP stack – переривання по Ethernet і LwIP стек. Переривання Ethernet, виникає, коли прийнято, або відправлено повідомлення, LwIP стек реалізує транспортний, міжмережевий рівні і рівень мережевого доступу стека протоколів TCP/IP.
- RequestMailBox() і ResponseMailBox() – поштову скриньку [33, 34] для передачі одного повідомлення. Через них передаються повідомлення прикладного рівня від/до стека TCP/IP.
- ServerGetRequest() – програмне завдання [35], яка отримує повідомлення від клієнта і передає його відповідної задачі роботи з портом. Також перепаковуємо повідомлення з формату JSON в формат зручний для завдання порту.

- ServerSendResponse() – програмне завдання, яка приймає безліч повідомлень від завдань периферійних інтерфейсів і передає LwIP стеку для відправки клієнтові.
- PortQueue() – черга повідомлень [34] до програмної завданню порту.
- PeripheryPortTask() – програмне завдання роботи з портом (периферійним інтерфейсом) – виконує відправку повідомлення отриманого від завдання ServerGetRequest() периферійному інтерфейсу, з яким вона взаємодіє через PortSemaphore().
- ResponseQueue() – черга повідомлень від програмної завдання порту до програмної завданню формування відповіді клієнтові.
- PortSemaphore() – семафор [https://ru.wikipedia.org/wiki/Семафор_(программирование), https://www.freertos.org/Embedded-RTOS-Binary-Semaphores.html], використовується для синхронізації переривання PeripheryPortIRQ і програмної завдання PeripheryPortTask(). Використовується також для захисту від відправки повідомлення, якщо порт вже в роботі.
- PeripheryPortIRQ – програмний код переривання периферійного порту. Переривання може викликатися в наступних випадках – закінчення прийому, передачі, помилки порту.
- UART – повідомлення передаються по периферійному інтерфейсу.

Послідовно опишемо, як відбувається повний цикл передачі повідомлення. Каналом Ethernet приходить повідомлення від клієнта (Eth), яке приймає мікроконтролер і обробляє стек LwIP. Прикладна частина запиту передається через RequestMailBox() в програмне завдання ServerGetRequest().

Програмна завдання ServerGetRequest() отримує запит в форматі JSON, витягує з нього номер і назва порту для передачі повідомлення в відповідну задачу порту PeripheryPortTask().

Передача повідомлення відбувається через програмну чергу PortQueue(). Якщо чергу зайнята (попереднє повідомлення ще не відправлено), то запит не

буде поставлений в чергу, а клієнту буде надіслано повідомлення, що інтерфейс зайнятий. Якщо чергу вільна, то відбувається постановка повідомлення в чергу.

Завдання порту PeripheryPortTask() отримує повідомлення з черги PortQueue(), перевіряє, чи вільний семафор порту PortSemaphore() (а значить і сам порт). Якщо семафор вільний, то завдання захоплює його і робить налаштування інтерфейсу для передачі і запуск передачі. Якщо семафор зайнятий, то завдання очікує, поки він звільниться.

Після того як повідомлення буде передано, в залежності від значення часу очікування відповіді (таймаута відповіді), програмне завдання переналаштує порт на прийом повідомлення (якщо прийом можливий) і чекатиме відповідь. Якщо таймаут дорівнює нулю, то відразу після відправки клієнтові повернеться підтвердження відправки, відповідь в цьому випадку не очікується. Якщо значення таймаута менше нуля – означає що периферійний порт, буде очікувати відповідь, до тих пір, поки він не прийде. Якщо значення таймаута більше нуля, то порт буде очікувати задану кількість мілісекунд, після чого поверне клієнту повідомлення про таймауті, або повідомлення з відповіддю, дивлячись що, настане раніше.

Обробкою подій виникають в процесі передачі повідомлення «UART» через периферійний інтерфейс займається переривання PeripheryPortIRQ. Дане переривання обробляє події закінчення передачі повідомлення, закінчення прийому відповіді, помилок по порту та інші інтерфейсозалежні переривання. Після закінчення відправки, переривання звільняє семафор порту PortSemaphore(), тим самим передаючи управління програмної завданню порту.

Програмна завдання порту, аналізує результат відправки і повертає відповідь клієнту через чергу ResponseQueue().

Програмна завдання ServerSendResponse() отримує повідомлення від завдання порту, запаковує його в формат JSON і відправляє його через ResponseMailBox() стеку LwIP, для подальшої відправки клієнтові через Ethernet.

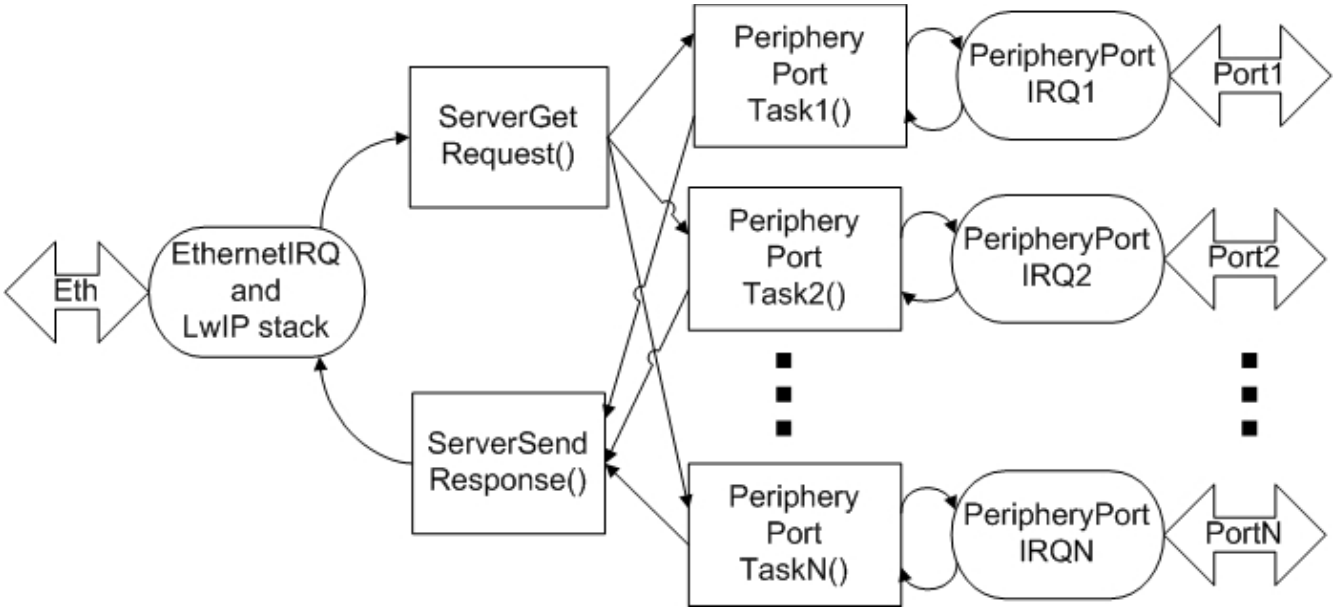


Рисунок 5.10 – Структура ПЗ коммутатора периферійних інтерфейсів

На рисунку 5.10 наведена повна структура ПО, що розробляється. Для кожного периферійного порту в момент налаштування буде створюватися окрема задача *PeripheryPortTask*(), принцип роботи якої описаний вище. Для роботи з портом Ethernet буде використовуватися два завдання, оскільки більше їх створювати не має сенсу. Обробка JSON може бути перенесена в завдання порта, в разі сильної завантаженості завдань сервера – *ServerGetRequest*() і *ServerSendResponse*().

5.4 Розробка програмного забезпечення для налагодження і тестування.

Для налагодження і тестування програмного забезпечення комутатора вбудованих інтерфейсів, скористаємося утилітою PuTTY, що дозволяє посилати дані через Ethernet. При цьому її необхідно налаштувати наступним чином (рисунок 5.11).

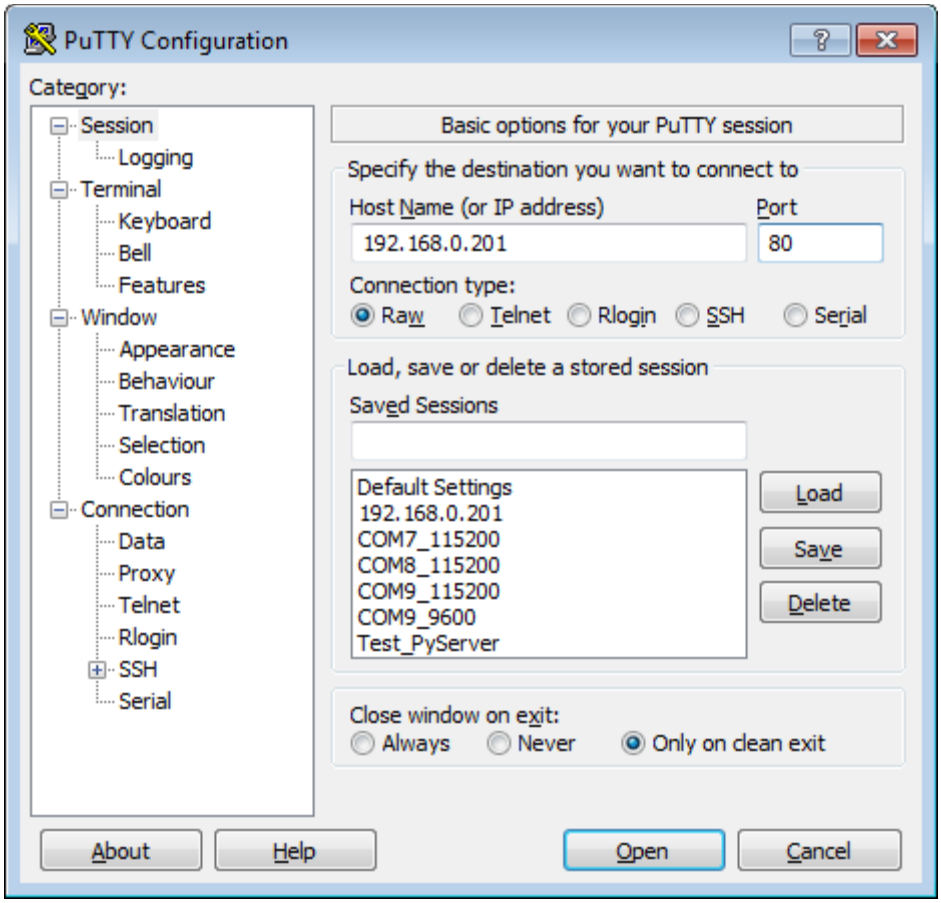


Рисунок 5.11 – Налаштування PuTTY для емуляції контролера верхнього рівня

В даному вікні необхідно в поле «Host Name» ввести IP-адресу пристрою – 192.168.0.201 і в поле «Port», номер порту 80. Перемикач «Connection type» встановити в положення «Raw». Після цього натиснути клавішу «Open» – це призведе до відкриття TCP/IP з'єднання з нашим пристроєм. Після цього можна відправляти повідомлення в JSON форматі.

Також розробимо периферійний пристрій що буде виконувати прості, візуально видимі команди. Для цього скористаємося відлагоджувальною платою STM32VLDISCOVERY (рисунок 5.12).

Рисунок 5.12 – Відлагоджувальна плата STM32VLDISCOVERY

На цій платі встановлені два світлодіоди, які і використовуємо для відлагодження. Реалізуємо в емуляторі підтримку наступних команд:

- blink blue – засвічувати синій світлодіод з періодом 500 мс, при цьому отладочная плата повинна повертати відповідь «okey blue»;
- blink green – засвічувати зелений світлодіод з періодом 500 мс, при цьому отладочная плата повинна повертати відповідь «okey green»;
- off blue – вимикає синій світлодіод і повертає відповідь «off»;
- off green – вимикає зелений світлодіод і повертає відповідь «off».

В емуляторі необхідно реалізувати можливість роботи по інтерфейсу UART з характеристиками:

Швидкість – 115200 бод;

Контроль парності – немає;

Кількість стопбіт – один;

Кількість інформаційних біт – вісім.

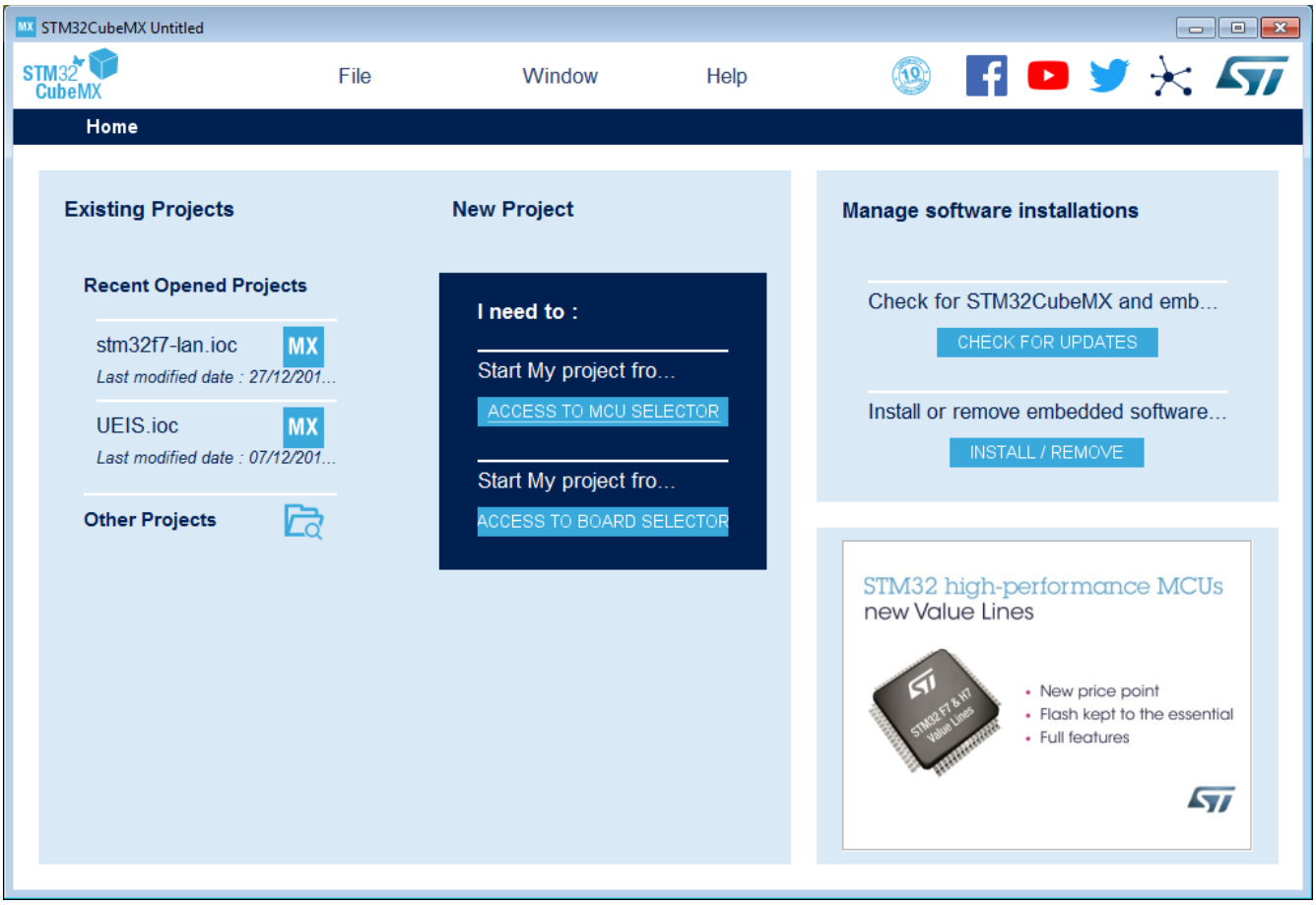
5.5 Розробка програмного забезпечення

5.5.1 Налаштування генерації проекту в середовищі STM32CubeMX

Для прискорення розробки програмного функціоналу скористаємося STM32CubeMX.

STM32CubeMX є візуальним графічним редактором для конфігурації мікроконтролерів сімейства STM32, що дозволяє генерувати код на основі мови Сі, використовуючи для цього графічний помічник [36].

Створений компанією STMicroelectronics програмний пакет STM32CubeMX є справжним автоматизованим робочим місцем для розробників систем на базі 32-бітових мікроконтролерів STM32, виконаних на основі ядер ARM Cortex. Це зручне середовище для повноцінного налаштування конфігурації МК з видачею пакета файлів ініціалізації, придатних для подальшого використання в ряді систем розробки і налагодження керуючого коду МК. STM32CubeMX значно спрощує створення вбудованого ПО, прискорює цей процес, не вимагає від початківців фахівців досконального знання документації на МК, дозволяє обійтися початковими відомостями про апаратну і програмну архітектуру контролера, про можливості бібліотек ПО [36].



Рисунокм 5.13 – Вікно роботи з проектами в редакторі STM32CubeMX

На рисунку 5.13, представлено головне вікно STM32CubeMX, що бачить розробник при відкритті програми. У ньому можна створити новий проект, або

відкрити попередні. При створенні нового проекту можна вибрати окремий мікроконтролер, або мікроконтролер в складі відлагоджувальної плати. Нас цікавить створення нового проекту з можливістю вибору відлагоджувальної плати. Для цього виберемо пункт меню «Start My project from Access To Board Selector».

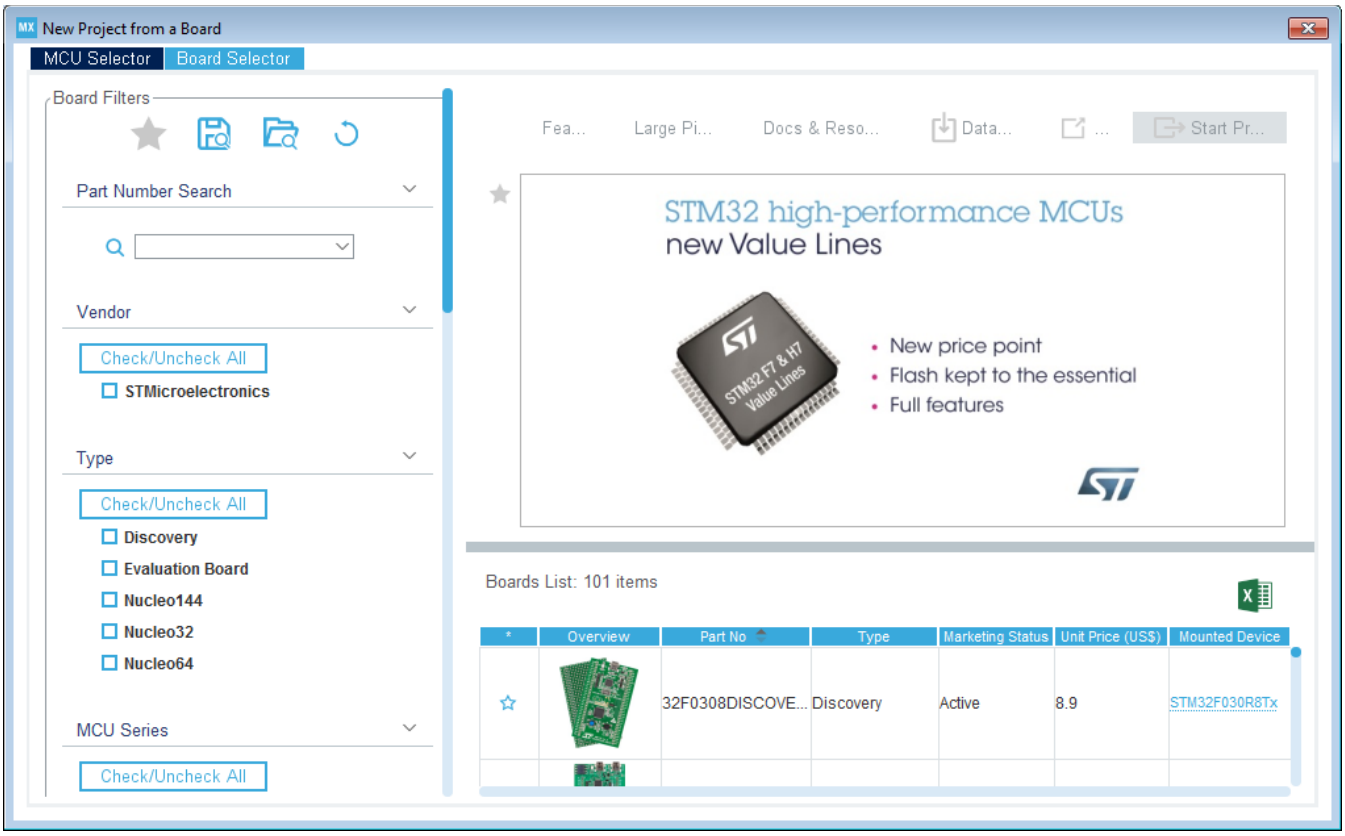


Рисунок 5.14 – Вибір отладочной плати в середовищі розробки STM32CubeMX

На рисунку 5.14 представлено вікно вибору відлагоджувальної плати. Використана нами відлагоджувальна плата має тип Nucleo144 і містить в собі мікроконтролер серії STM32F7. Виберемо дані критерії пошуку для фільтрації всіх плат. Внизу, у вікні вибору плати, виберемо NUCLEO-F767ZI і натиснемо кнопку Start Project у верхній правій частині вікна. З'явиться вікно настройки периферії мікроконтролера (рисунок 5.15). Частина периферії буде налаштована по-замовчуванню, – Ethernet, USB, і оцінний UART3. В даному вікні видно вкладки:

Pinout & Configuration – основна відкрита вкладка, призначена для налаштування режимів роботи периферії і висновків мікроконтролера

Clock Configuration – вкладка призначена для налаштування системи тактування мікроконтролера STM32.

Project Manager – вкладка налаштування проекту в середовищі розробки. Дозволяє налаштувати правила, за якими буде згенеровано програмний код для мікроконтролера.

Tools – інструменти для налагодження режимів електроживлення пристрою. Дозволяють оцінити споживану потужність при заданій налаштування МК, температуру та інше.

Рисунок 5.15 – Вікно налаштування периферії мікроконтролера в середовищі STM32CubeMX.

Результатом роботи STM32CubeMX буде проект в середовищі розробки IAR Embedded Workbench. Проект включатиме в себе файли бібліотек HAL, і

файл main.c в якому буде код ініціалізації і запуску периферії налаштованої в редакторі *STM32CubeMX*.

На даному етапі нам необхідно налаштувати використання в проекті ОС *FreeRTOS* і стека *LwIP*. Також необхідно підключити бібліотеки HAL для периферійних інтерфейсів.

Однією з переваг налаштування проекту з використанням STM32CubeMX є просте налаштування системи тактування (рисунок 5.16) і режимів роботи виводів мікроконтролера.

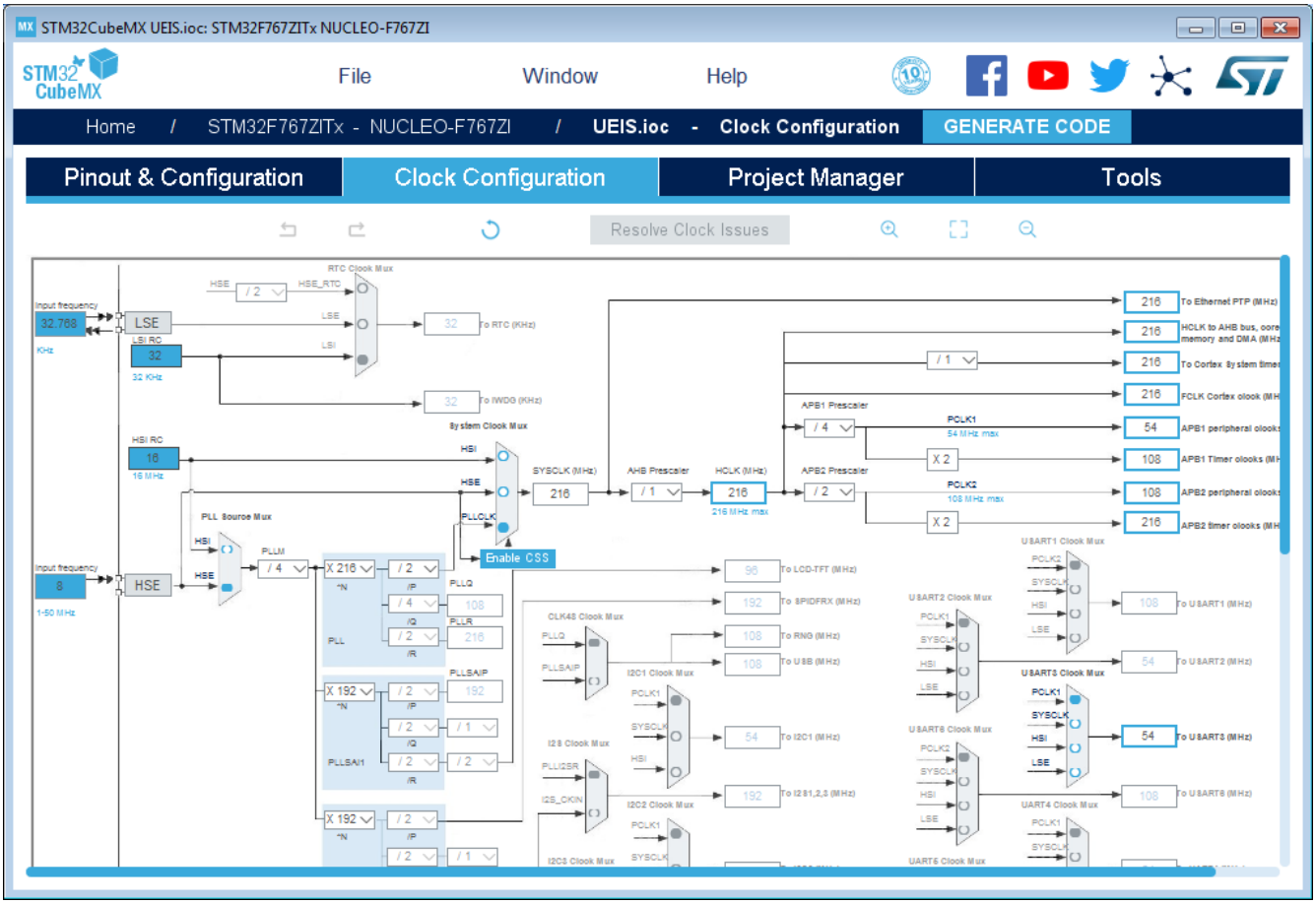


Рисунок 5.16 – Налаштування системи тактування мікроконтролера в середовищі*STM32CubeMX*

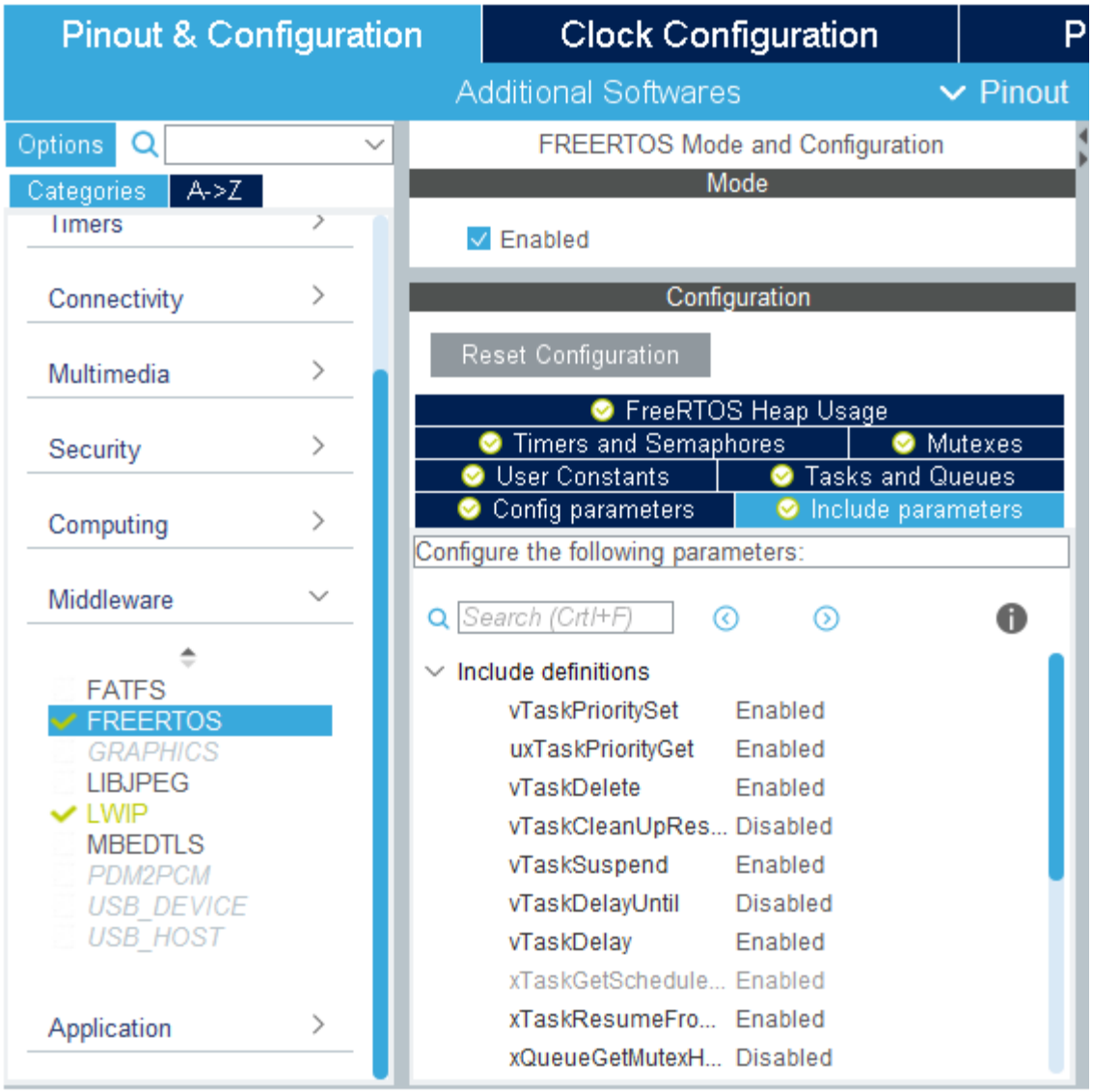


Рисунок 5.17 – Налаштування ОСРЧ FreeRTOS в середовищі *STM32CubeMX*

ОСРЧ FreeRTOS має бути налаштована наступним чином (рисунок 5.17):

– включена підтримка багатозадачності з витисненням;

– можливість створення і видалення завдань в процесі роботи;

– включена підтримка семафорів і черг.

Необхідно також створити два програмних завдання – ServerGetRequest() і ServerSendResponse() і одну чергу в задачі ServerSendResponse(). Решта завдань, а також семафори і черги будуть створюватися динамічно, в процесі роботи програми.

5.5.2 Розробка та відлагодження проекту в середовищі IAR *Embedded Workbench*

Середовище розробки *IAR Embedded Workbench* дозволяє розробляти якісне ПЗ під МК *STM32* (рисунок 5.18). Результатом роботи компілятора *IAR* є добре оптимізований код. Одним з переваг роботи з використанням *IAR* є наявність вбудованого емулятора мікроконтролера *C-SPY*. Іншою перевагою є наявність доповнень для налагодження ОСРЧ *FreeRTOS*, як в мікроконтролері, так і на емуляторі [37].

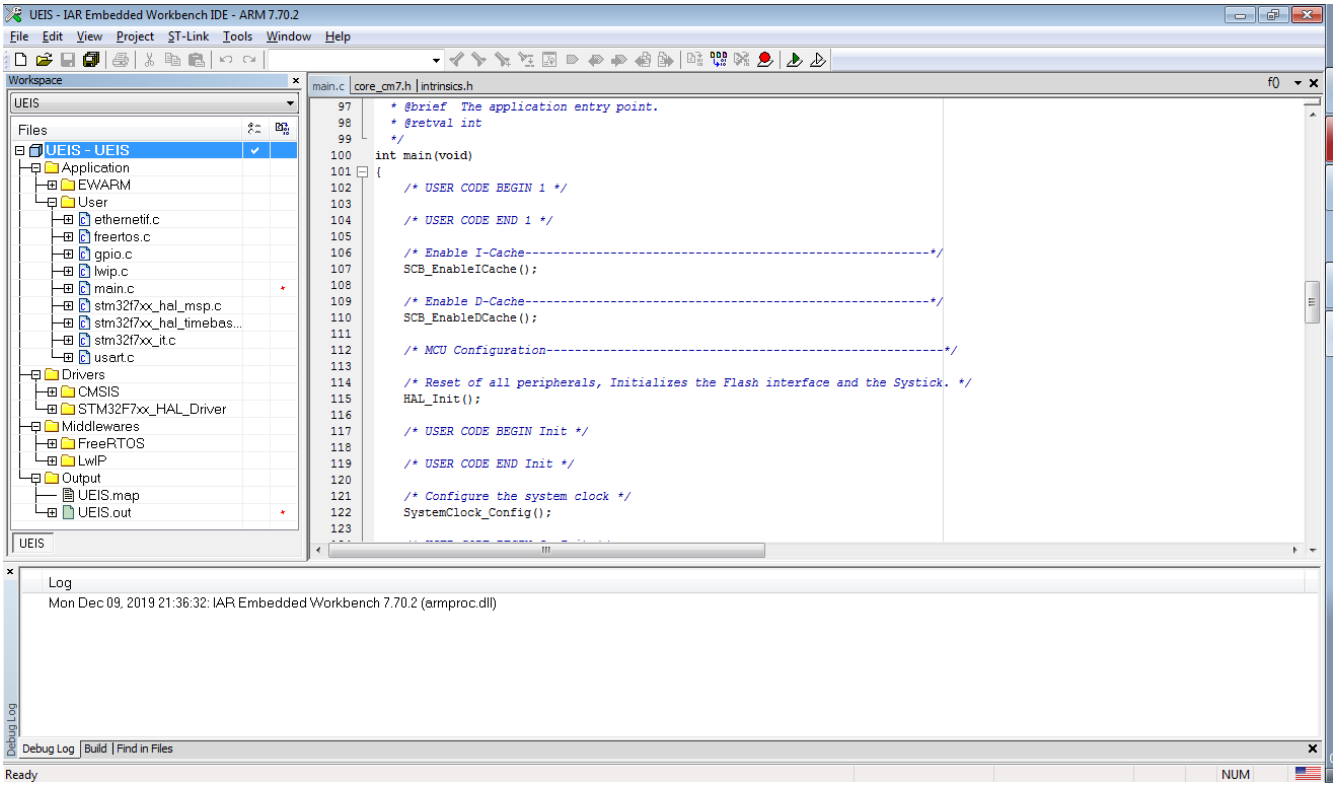


Рисунок 5.18 – Вікно редактору коду середовища *IAR Embedded Workbench*

Оскільки проект, що розробляється, нами для *IAR Embedded Workbench* був отриманий за допомогою середовища *STM32CubeMX*, то основні налаштування вже застосовані. Єдине що ще необхідно налаштувати, це відключити повністю оптимізацію – вона включена за замовчуванням, – це ускладнює налагодження коду. Відключити можна в меню «C/C++ Compilers», у вкладці «Optimization» (рисунок 5.19). Оптимізацію можна залишити для файлів ОСРВ і стека *LwIP*, оскільки вони вже написані і відлагоджені. Також

необхідно включити плагін для налагодження *FreeRTOS* в розділі «Debugger», «Plugins».

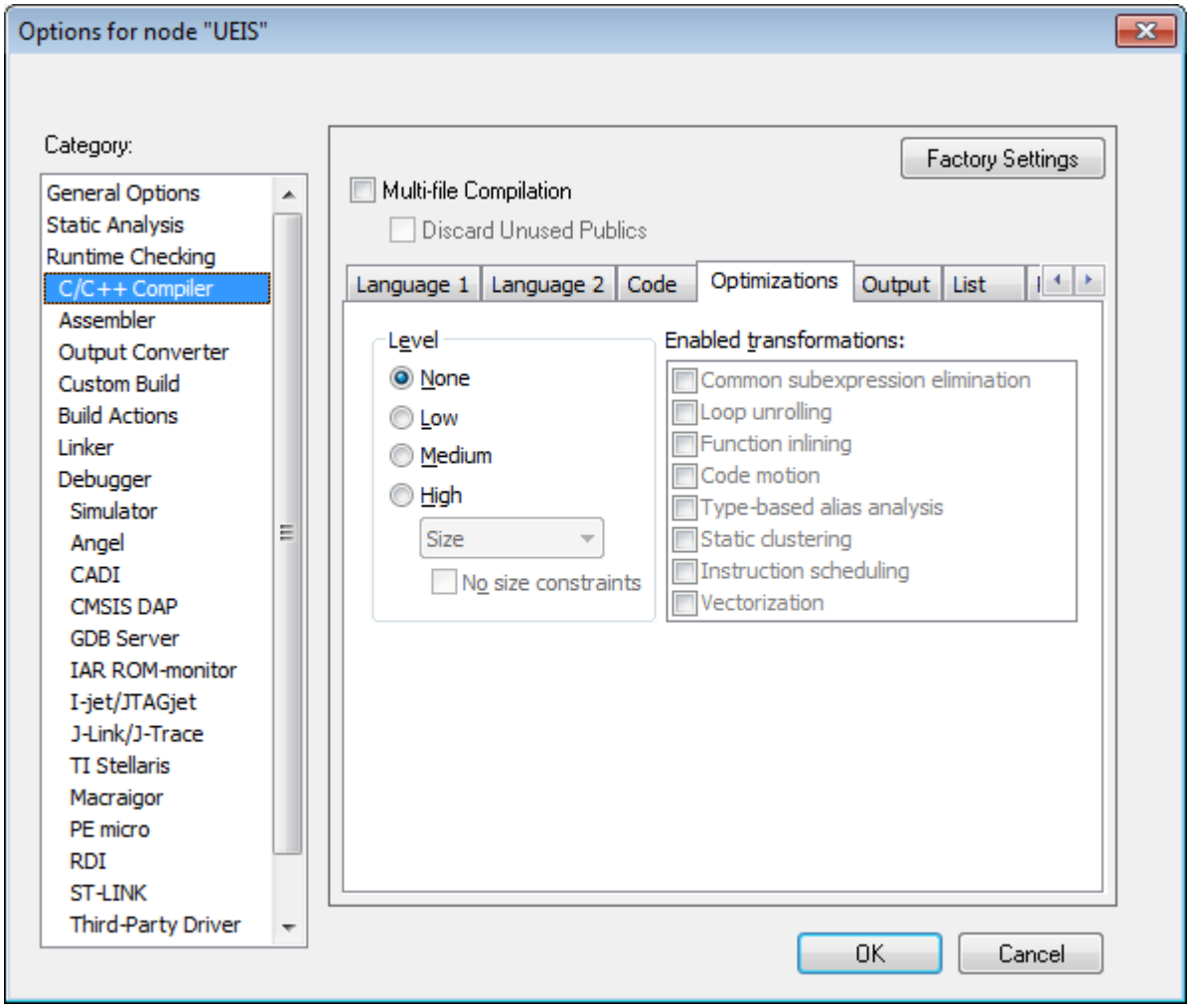


Рисунок 5.19 – Відключення оптимізації для всього проекту в середовищі *IAR Embedded Workbench*

5.6 Висновки

Розроблений пристрій дозволяє передавати повідомлення, одержані від контролера верхнього рівня, на периферійні пристрої, і повертати від них відповідь назад. При цьому пристрій підтримує роботу з периферійними пристроями через три інтерфейсу UART, SPI, I2C. Надалі планується додати підтримку CAN і 1-Wire, підтримка яких вже є в мікроконтролері STM32F767.

Лістинг коду сервера обміну повідомленнями наведено в додатку до пояснювальної записки.

ВИСНОВКИ

У даній роботі було розроблено програмно-апаратний комутатор інтерфейсів вбудованих систем Для цього був виконаний аналіз аналогічних рішень та виявлені їх переваги та недоліки. На підставі їх недоліків а також поставленої задачі була розроблена структурна програмного забезпечення. Також був проведений вибір відлагоджувальної плати та середовища розробки. Для тестування функцій пристрою був розроблений емулятор.

Програмна частина була розроблена з використанням мови програмування С з використанням ОСРЧ FreeRTOS. Частина функціоналу зв’язку з контролером верхнього рівня була перевірена за допомого утиліти PuTTY, а частина роботи с інтерфейсами – за допомогу розробленого нами емулятора

Таким чином, усі поставлені в роботі задачі були виконані.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Розвиток технології IoT: як «розумнішає» бізнес і життя навколо нас [Електронний ресурс].- Режим доступу: https://rb.ru/opinion/razvitie-tehnologii-iot/

2. СПК1xx сенсорні панельні контролери з Ethernet [Електронний ресурс].- Режим доступу: https://owen.ru/product/spk1xx (Дата звернення 25.10.2019)

3. Панельний програмуємий логічний контролер ОВЕН СПК207 [Електронний ресурс].- Режим доступу: http://automation.kg/tovary/%D1%81%D1%82%D0%B0%D1%82%D1%8C%D1%8F/23-sens-pnel-control/112-panelnyj-programmiruemyj-logicheskij-kontroller-oven-spk207 (Дата звернення 25.10.2019)

4. Wiren Board 6 [Електронний ресурс].- Режим доступу: https://wirenboard.com/ru/product/wiren-board-6/ (Дата звернення 25.10.2019)

5. Промисловий панельний контролер VP-4131-EN [Електронний ресурс].- Режим доступу: https://ipc2u.ua/catalog/vp-4131-en (Дата звернення 25.10.2019)

6. Сенсорний панельний контролер LSIT 07-400 [Електронний ресурс].- Режим доступу: https://isup.ru/articles/4/12740/ (Дата звернення 25.10.2019)

7. Контролер PACSystems RX3i компанії General Electric [Електронний ресурс].- Режим доступу: https://indusoft.com.ua/ge-automation-controls/control-systems/programmable-automation-controllers/pacsystems-rx3i/?gclid=EAIaIQobChMI57P5tdmg5gIVEc-yCh04QAxhEAMYASAAEgI66fD_BwE (Дата звернення 25.10.2019)

8. Промисловий панельний контролер VP-2117 [Електронний ресурс].- Режим доступу: https://ipc2u.ua/catalog/vp-2117 (Дата звернення 25.10.2019)

9. ADFweb [Електронний ресурс].- Режим доступу: https://www.adfweb.com/home/default.asp (Дата звернення 25.10.2019)

10. Universal UPnP Bridge for Embedded Non-IP Device with Heterogeneous Network Interfaces [Електронний ресурс].- Режим доступу: https://ieeexplore.ieee.org/document/6704002/ (Дата звернення 25.10.2019)

11. Designing an embedded system for interfacing with networks based on ARM [Електронний ресурс].- Режим доступу: https://ieeexplore.ieee.org/document/6116598 (Дата звернення 25.10.2019)

12. Real-Time Kernels: µC/OS-II and µC/OS-III [Електронний ресурс].- Режим доступу: https://www.micrium.com/rtos/kernels/ (Дата звернення 25.10.2019)

13. Мережева модель OSI [Електронний ресурс].- Режим доступу: https://en.wikipedia.org/wiki/OSI_model (Дата звернення 28.10.2019)

14. Система на кристали [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/Система_на_кристалле (Дата звернення 28.10.2019)

15. ПЛІС [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/ПЛИС (Дата звернення 28.10.2019)

16. Що таке FPGA [Електронний ресурс].- Режим доступу: http://kanyevsky.kpi.ua/ru/статьи/что-такое-fpga/ (Дата звернення 28.10.2019)

17. Мікроконтролер [Електронний ресурс].- Режим доступу: https://en.wikipedia.org/wiki/Microcontroller (Дата звернення 28.10.2019)

18. Плата STM32F103C8T6. Завантаження програми у FLASH-пам'ять мікроконтролера через системний бутлоадер. [Електронний ресурс].- Режим доступу: http://mypractic.ru/urok-2-plata-stm32f103c8t6-zagruzka-programmy-vo-flash-pamyat-mikrokontrollera-cherez-sistemnyj-butloader.html (Дата звернення 28.10.2019)

19. AVR [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/AVR (Дата звернення 28.10.2019)

20. STM32 [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/STM32 (Дата звернення 28.10.2019)

21. Відлагоджувальні плати на базі мікроконтролерів STM32 [Електронний ресурс].- Режим доступу: https://www.compel.ru/lib/94885 (Дата звернення 2.11.2019)

22. Відлагоджувальні плати сімейства DISCOVERY компанії STMicroelectronics. Огляд. [Електронний ресурс].- Режим доступу: https://www.terraelectronica.ru/news/4102 (Дата звернення 15.11.2019)

23. Бітченко О.М. Електроніка і мікросхемотехніка. Проектування і програмуваннямікропроцесорнихпристроїв: підручникдлятехн. таінж.- пед. вищихнавч. закладів / О. М. Бітченко, О. І. Цопа, Д. Г. Ганшин. – Х.: Фінарт, 2016. - 354 с.

24. CAN [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/Controller_Area_Network (Дата звернення 15.11.2019)

25. RS-485 [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/RS-485 (Дата звернення 15.11.2019)

26. CSMA/CD [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/CSMA/CD (Дата звернення 15.11.2019)

27. JSON - JavaScript Object Notation [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/JSON (Дата звернення 15.11.2019)

28. XML - eXtensible Markup Language [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/XML (Дата звернення 15.11.2019)

29. STM32F767ZI [Електронний ресурс].- Режим доступу: https://www.st.com/en/microcontrollers-microprocessors/stm32f767zi.html#resource (Дата звернення 15.11.2019)

30. NUCLEO-F767ZI [Електронний ресурс].- Режим доступу: https://www.st.com/en/evaluation-tools/nucleo-f767zi.html#resource (Дата звернення 15.11.2019)

31. LwIP [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/LwIP (Дата звернення 16.11.2019)

32. FreeRTOS [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/FreeRTOS (Дата звернення 16.11.2019)

33. Черга повідомлень [Електронний ресурс].- Режим доступу: https://ru.wikipedia.org/wiki/Очередь_сообщений (Дата звернення 25.11.2019)

34. FreeRTOS Queues [Електронний ресурс].- Режим доступу: https://www.freertos.org/Embedded-RTOS-Queues.html (Дата звернення 25.11.2019)

35. Tasks [Електронний ресурс].- Режим доступу: https://www.freertos.org/taskandcr.html (Дата звернення 25.11.2019)

36. CubeMX і Workbench: створення проекту на базі STM32 за допомогою безкоштовного програмного забезпечення [Електронний ресурс].- Режим доступу: . https://www.compel.ru/lib/75317 (Дата звернення 28.11.2019)

37. IAR Embedded Workbench for ARM [Електронний ресурс].- Режим доступу: https://www.iar.com/iar-embedded-workbench/#!?architecture=Arm (Дата звернення 28.11.2019)