

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

*Освітньо-професійна програма: «Обслуговування
комп'ютерних систем і мереж»*

Група: 4КС-57

Дипломний проект

**здобувача освіти денної форми навчання
КС.57.22.000.ДП**

**ТІГІНЯНА
СЕРГІЯ ДМИТРОВИЧА**

**м. Одеса
2024 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Обслуговування комп'ютерних систем і мереж»

Група: 4КС-57

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

**Розробка системи спостереження за сейсмічною активністю
на базі платформи STM32**

Проектний матеріал складається з пояснювальної записки на 86 сторінках та графічного (презентаційного) матеріалу на 18 аркушах (слайдах)

Дипломник _____ (Тігіян С.Д.)

Керівник _____ (Кривченко Ю.В.)

Консультанти:

з економічного розділу _____ (Іванченков В.С.)

з розділу охорони праці та техніки безпеки _____ (Чорновол Н.І.)

з нормоконтролю _____ (Петрашова В.І.)

старший консультант _____ (Кривченко Ю.В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю.В.)

Завідувач відділення _____ (Скорнякова О.В.)

Захист «18» 06 2024 р. Протокол ЕК № 2

Оцінка ЕК 4/добре / 85%

Секретар ЕК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 123 «Комп'ютерна інженерія»
Освітньо-професійна програма «Обслуговування комп'ютерних систем і мереж»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І.В.

« 15 » 01 2024 р.

ЗАВДАННЯ

на дипломний проект

Здобувачеві освіти Тігіяну Сергію Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема проекту Розробка системи спостереження за сейсмічною активністю на базі платформи STM32

затверджена наказом по коледжу від «02» листопада 2023 р. № 244-А2-ОБ

2. Термін здачі закінченого проекту 10.06.24 р.

3. Вихідні данні до проекту 1. Передбачити можливість спостереження за сейсмічною активністю з урахуванням положення пристрою на поверхні;

2. У якості платформи використовувати плату STM32F407vg;

3. У якості акселерометру застосувати вбудований датчик прискорення LIS3DSH;

4. Передбачити відправлення даних спостереження на ПК через USB-порт;











5. Забезпечити можливість моніторингу сейсмічної активності у часі

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Огляд пристроїв реєстрації сейсмічних даних; Розробка структури системи спостереження за сейсмічною активністю; Реалізація взаємодії складових системи спостереження за сейсмічною активністю; Налаштування мікроконтролера у STM32CubeMX; Програмування мікроконтролера у Keil uVision; Розробка програмного забезпечення для зчитування і відображення даних з акселерометра

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Типи хвиль сейсмічної активності; Структура MEMS-акселерометра; Функціональна схема акселерометра для STM32F407VG; Типова структура зв'язків і ліній інтерфейсу SPI; Основні характеристики платформи STM32F407VG; Структурна схема апаратних засобів; Принципова електрична схема апаратних засобів; Встановлені режими портів введення-виведення; Налаштування дерева тактування Clock Configuration; Блок-схема алгоритму обробки даних акселерометра; Інтерфейс застосунку сейсмічної активності

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Кривченко Ю.В.		
Економічний розділ	Іванченков В.С.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання 15.01.24

Керівник Кривченко Ю.В.

Завдання прийняв до виконання Тігінян С.Д.


(підпис)

(підпис)

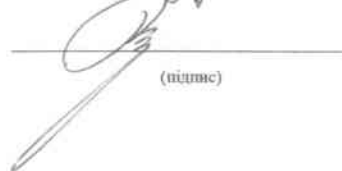
КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Вступ. Постановка мети та задач проектування	29.04.2024	Виконано
2	Обґрунтування доцільності розробки застосунку	1.05.2024	Виконано
3	Аналітичний огляд методів реєстрації сейсмічних хвиль	3.05.2024	Виконано
4	Огляд пристроїв реєстрації сейсмічних даних	6.05.2024	Виконано
5	Вибір технічних та програмних засобів розробки	8.05.2024	Виконано
6	Опис структури проекту та підключення датчиків	14.05.2024	Виконано
7	Розробка структури системи спостереження	13.05.2024	Виконано
8	Розробка алгоритмів та структури застосунку	17.05.2024	Виконано
9	Розробка програмного забезпечення проекту	19.05.2024	Виконано
10	Створення інтерфейсу програми та налагодження	21.05.2024	Виконано
11	Реалізація моделі проекту та її випробування	26.05.2024	Виконано
12	Аналіз результатів, підготовка слайдів презентації	29.05.2024	Виконано
13	Економічні розрахунки та питання з охорони праці	2.06.2024	Виконано
14	Підготовка графічної частини проекту	5.06.2024	Виконано
15	Підготовка проекту до захисту та тестування ПЗ	8.06.2024	Виконано

Дипломник


(підпис)

Керівник


(підпис)

ЗМІСТ

Вступ.....	7
1 Основна частина.....	8
1.1 Аналітичний огляд методів реєстрації сейсмічних хвиль.....	8
1.2 Огляд пристроїв реєстрації сейсмічних даних.....	10
1.3 Аналіз технічного завдання на проектування.....	15
1.4 Вибір та аналіз апаратного забезпечення проекту.....	15
1.4.1 Огляд та порівняння датчиків лінійного прискорення.....	16
1.4.2 Структура та порівняльна характеристика акселерометрів STM32.....	18
1.4.3 Вибір і обґрунтування обчислювальної платформи.....	22
1.4.4 Аналіз можливостей мікроконтролерів платформи STM32F4.....	24
1.5 Вибір програмного забезпечення для програмування мікроконтролеру STM32.....	28
1.5.1 Середовище розробки програмного забезпечення для мікроконтролерів Keil uVision.....	29
1.5.2 Середовище розробки програмного забезпечення для мікроконтролерів STM32CubeMX.....	31
1.6 Застосування мови C для налаштування зовнішніх пристроїв.....	35
1.7 Розробка структури системи спостереження за сейсмічною активністю.....	37
1.8 Реалізація взаємодії складових системи спостереження за сейсмічною активністю.....	40
1.9 Налаштування мікроконтролеру у STM32CubeMX.....	41
1.10 Програмування мікроконтролеру у Keil uVision.....	43
1.11 Розробка програмного забезпечення для зчитування і відображення даних з акселерометра.....	46
2 Економічний розділ.....	51
2.1 Резюме.....	51
2.2 Розрахунок ціни програмного продукту нормативним методом.....	51
3 Розділ охорони праці та техніки безпеки.....	57

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

3.1 Аналіз небезпечних і шкідливих чинників, що впливають на розробника	57
3.2 Розробка заходів з охорони праці.....	57
3.2.1 Виробничі приміщення.....	57
3.2.2 Освітлення.....	58
3.2.3 Гігієнічні нормування параметрів повітря робочої зони.....	58
3.2.4 Організація робочих місць з ПК.....	59
3.3 Вимоги до проведення профілактичних медичних оглядів користувачів ВДТ.....	60
3.4 Пожежна безпека.....	61
Висновки.....	62
Перелік використаних інформаційних джерел.....	63
Додаток А. Лістинг програми налаштування мікроконтролера STM32.....	64
Додаток Б. Лістинг програми для перетворення та відображення даних з акселерометра.....	69
Додаток В. Слайди мультимедійної презентації.....	78

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Сейсмостійкість є критичним фактором в проектуванні будівель, особливо у районах, схильних до землетрусів. Здатність конструкції протистояти силам, створеним сейсмічною активністю, визначається різними факторами, включаючи тип ґрунту, на якому споруджено будівлю. Котрі ґрунти, такі як ті, що знаходяться у прибережних районах, більш сприйнятливі до розрідження під період землетрусу, що спроможне призвести до серйозних структурних пошкоджень. З іншого боку, тверді ґрунти, такі як скелі, забезпечують кращий опір сейсмічній активності.

Контроль поза пересуванням людей і транспорту доволі розповсюджене завдання в сучасному світі. Один із методів контролю такого пересування є аналіз сейсмічної активності, яку в будь-якому випадку провокує все що пересуваються по землі.

Акселерометр, встановлений на об'єкті, надає визначати ускорення, з яким об'єкт рухається. При цьому поява ускорення спроможне існувати разом з цим обумовлена зовнішніми впливами на об'єкт – зміною положення, ударами, струшуванням чи виведенням його зі стану рівноваги. Чутливість таких сенсорів надає фіксувати навіть дуже незначний вплив.

Принципи обміну даними поміж датчиками і предметами задля людської безпеки швидко набирають популярність. Прикладом такого принципу є «SmarCity», у якому елементи інфраструктури обмінюються поміж собою.

В даному дипломному проекті будуть розроблені програмно-апаратні засоби системи спостереження поза сейсмічною активністю на основі платформи STM-32. Розроблюваний пристрій у перспективі можливо буде використовувати задля визначення сейсмоактивних зон і позначення їх на карті.

Розроблюваний пристрій спроможне стати частиною більш складного проекту, направлено на розробку алгоритмів опрацювання сейсмічних даних, котрі несуть у собі інформацію про рух людей, їх швидкість, скільки ваги вони несуть на собі, з якою швидкістю пересуваються і інше. Така інформація спроможне існувати корисна задля охоронних компаній і державних органів.

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

1 ОСНОВНА ЧАСТИНА

1.1 Аналітичний огляд методів реєстрації сейсмічних хвиль

Задля розроблювання системи спостерігання поза сейсмічною активністю на основі платформи STM-32 треба ознайомитись з типами і особливостями сейсмічних хвиль, котрі несуть у собі інформацію про досліджувані цілі. Сейсмічні хвилі є інформаційними носіями про рух об'єктів на поверхні ґрунту. Через це вони можуть існувати використані задля виявлення і класифікації рухомих об'єктів у військовій і цивільній сферах.

Сейсмічними хвилями називають коливання земної товщі котрі переносять енергію пружних коливань. Джерелом таких пружних коливань можуть існувати як природні явища на кшталт землетрусів, виверження вулканів, обвалення у карстових пустотах так та штучні явища такі як вибухи, удари, рух окремих об'єктів і інше.

Класифікувати сейсмічні хвилі прийнято поза напрямом коливання часток ґрунту по відношенню до розповсюдження фронту хвилі. Виділяють такі основних типи хвиль: повздовжні хвилі, поперечні хвилі, хвиля Релея і хвилі Лява.

З іншого боку сейсмічні хвилі можливо розділити на дві групи:

- хвилі котрі розповсюджуються з рівномірним, але швидким загасанням у товщі землі (приблизно $1/r$) сюди відносяться повздовжні і поперечні хвилі;
- хвилі Релея і Лява, котрі розповсюджуються поверхнею на не великих глибинах і не мають такого сильного загасання (приблизно $1/r^{0.5}$).

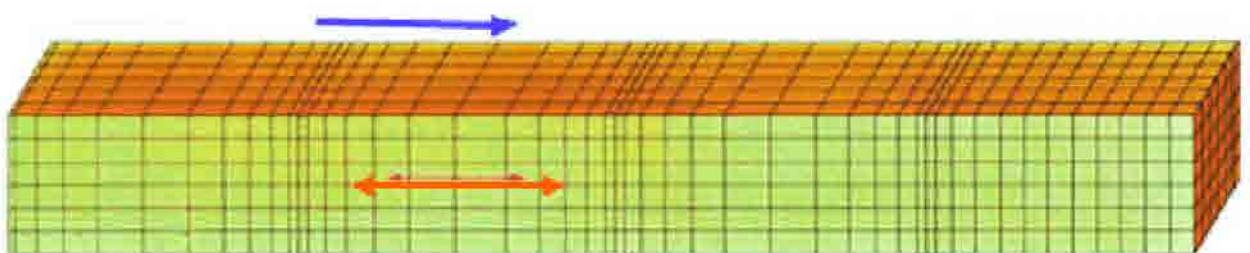


Рисунок 1.1. Повздовжня хвиля в пружному середовищі

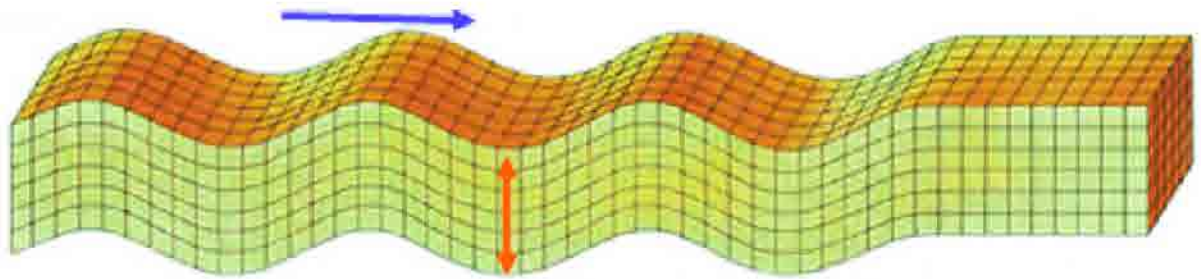


Рисунок 1.2. Поперечна хвиля в пружному середовищі

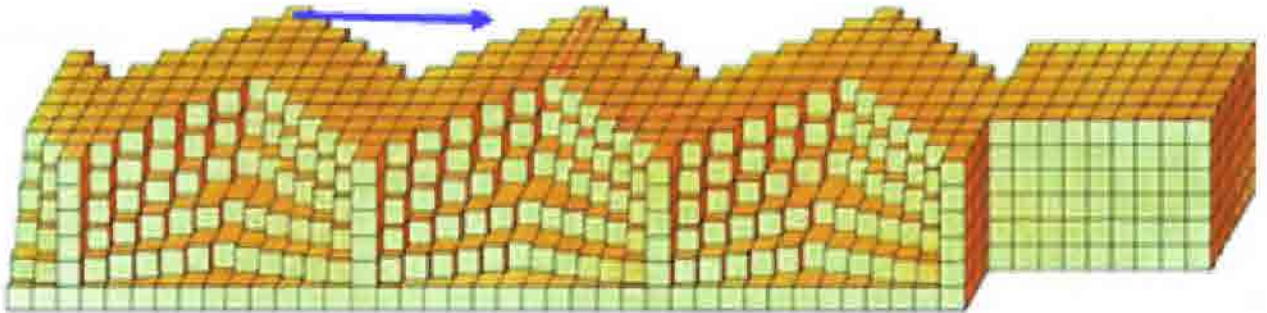


Рисунок 1.3. Хвиля Лява

Хвиля Лява (рис 1.3) утворюється частинками котрі коливаються з боку у бік в горизонтальній площині яка є перпендикулярною до напрямку розповсюдження хвилі.

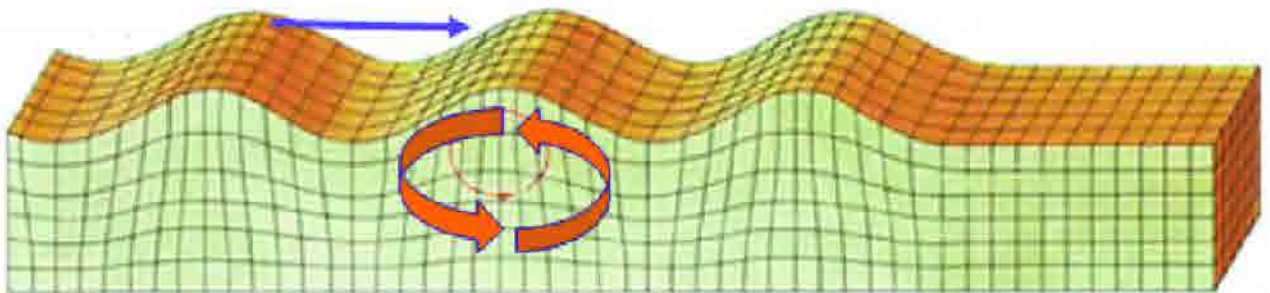


Рисунок 1.4. Хвиля Релея

Хвиля Релея (рис. 1.4) утворюється коловим рухом частинок в вертикальній площині, площина руху орієнтована в напрямку розповсюдження хвилі. У цілях збільшення радіусу визначення цілі найбільш цікаві хвилі другої групи. Разом з цим відомо, що загальна енергія впливу на земну поверхню, яку переносять поперечні і повздовжні хвилі є не більшою ніж 33%, але хвилі Релея 67%.

Найбільш вигідними з енергетичної точки зору є хвилі Релея. Особливістю

даного типу хвиль є еліптична поляризація з переважаючою вертикальною компонентою. Задля вирішення завдань, що пов'язані з виділенням корисної інформації з суміші сигнал-завада, треба мати апріорну інформацію про сигнал, котрий є предметом пошуку. Іншими словами, треба робити достатню кількість досліджень, котрі дозволять створити класифікатор досліджуваних сигналів [1].

Спеціалізовані прилади під назвою сейсмографи чи реєстратори використовують задля реєстрації сейсмічних хвиль. Вони фіксують навіть незначні коливання земної поверхні. Більшість таких приладів базуються на використанні чутливих індуктивних геофонів. Цей МК сейсмічних коливань являє собою котушку з підпружиненим осердям, яке рухається під дією зовнішніх механічних вібрацій.

1.2 Огляд приладів реєстрації сейсмічних даних

Збирання даних про сейсмічну активність спроможне існувати з виконуватись задля різних застосувань. Це спроможне існувати комплекс моніторингу сейсмічної активності землетрусів, де важливим параметром є реєстрація хвиль першої групи (повздовжніх і поперечних), георозвідка. Задля подальшої розроблювання системи спостереження поза сейсмічною активністю на треба проаналізувати напрацювання георозвідницької сфери і сфери оборонного комплексу. Задля контролю землетрусів використовуються сейсмічні станції, котрі складаються з декількох сейсмоприймачів. Прикладом такої системи є продукція Zetlab. Розглянемо сейсмічну станцію ZET 048-E [2].

Технічні характеристики станції ZET 048-E (рис.1.5):

- Кількість каналів – з 4 до 16;
- Тип входу – диференціальний;
- Вхідна смуга – 1кГц;
- Розрядність АЦП – 24 Біт;
- Максимальний діапазон вхідної напруги – з -10В до +10В;
- Динамічний діапазон – 124 дБ.

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10



Рисунок 1.5. Сейсмічна станція ZET 048-E

Станція ZET 048-E працює з кабельними геофонами такими як BC1313 Triaxial seismograph виробництва Zetlab (рис.1.6).



Рисунок 1.6. Кабельний геофон BC1313 Triaxial seismograph

Кабельний геофон BC1313 містить наступні технічні характеристики:

- Чутливість – $0.2 V/(m \cdot s^{-2})$;
- Частотний діапазон – з 0,3 до 400 Гц;
- Рівень власних шумів – $25 m/s^2$;
- Частота власного резонансу – 900 Гц;
- Напряга живлення – двополярна $\pm(10 \dots 15) В$.

Зм.	Арк.	№ докум.	Підпис	Дата

КС 57. 22 000. 00 ДП ПЗ

Арк.

11

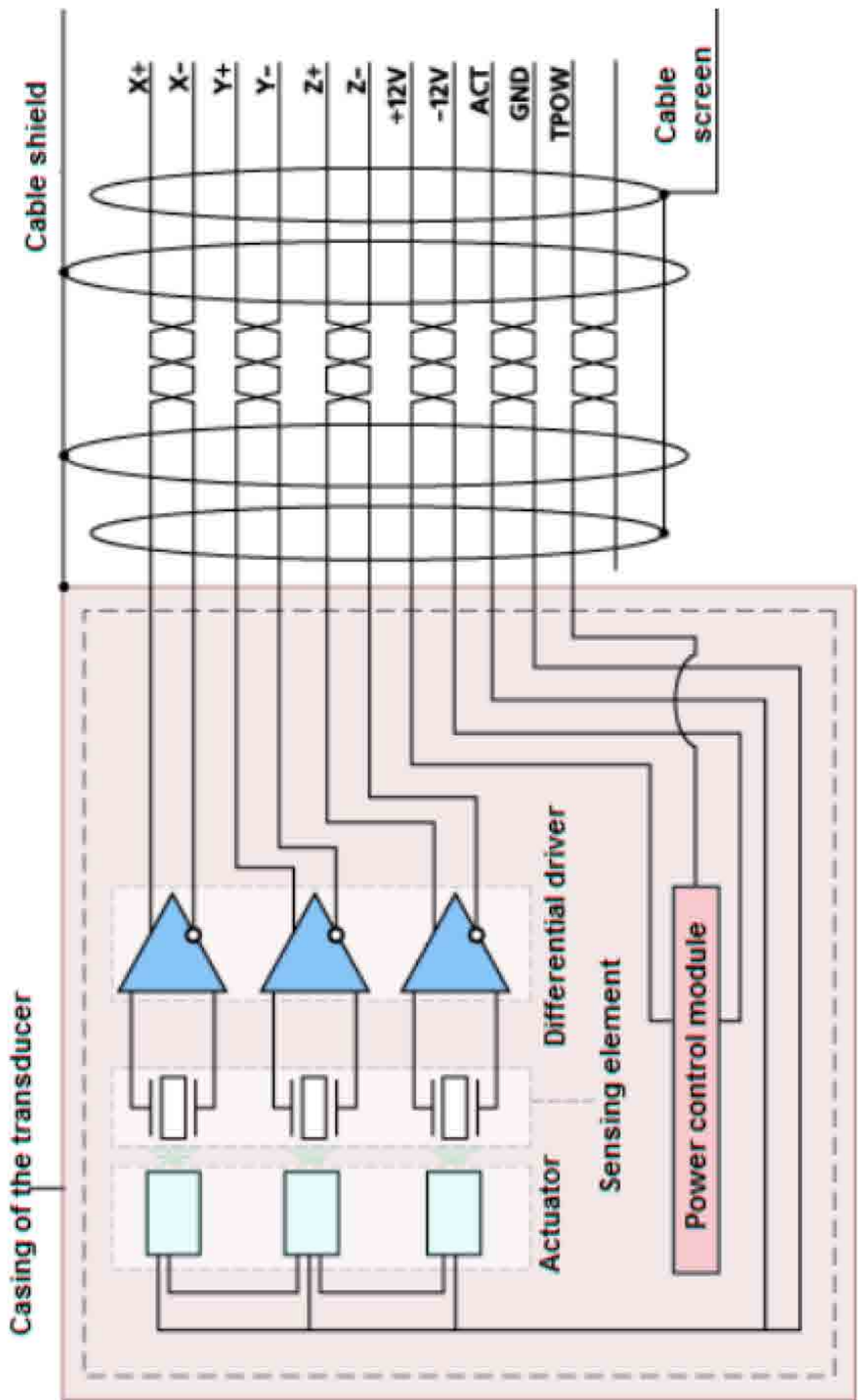


Рисунок 1.7. Структурна схема геофона Triaxial seismograph BC1313

Зм.	Арк.	№ докум.	Підпис	Дата

КС 57. 22 000. 00 ДП ПЗ

Арк.

12

Система ВС1313 (рис.1.7) використовує три сейсмічні геофони, котрі відповідають трьом осям вимірювання. Сейсмоприймачі розташовуються на віддалені з 10 до 600 метрів один з одного і з станції. Це надає у подальшому аналізувати отриманні дані з кожного датчика окремо і на основі часових зсувів робити висновки про місцезнаходження сейсмічної активності [3]. Недоліками системи можливо вважати її високу вартість, низьку мобільність і обмежений функціонал. Система здатна вести запис геоданих з прив'язкою до місцевості, але система не виконує жодної опрацювання отриманих даних.

В сфері оборони аналіз сейсмічної активності використовують в мінно-вибухових пристроях з сейсмічними детонаторами. Прикладом такого застосування є протипіхотна міна ПОМ-3 "Медальон" (рис.1.8), в складі якої є геофон і електронна система аналізу сейсмічної активності.



Рисунок 1.8. Протипіхотна міна ПОМ-3 "Медальон"

Система ПОМ-3 "Медальон" побудована на принципах аналізу сейсмічної активності у ближній зоні з датчика. Система опрацювання даних відсіює активність, не пов'язану з переміщенням живої сили ворога, що зменшує вірогідність хибного спрацювання детонатора.

Автономний бездротовий сейсмограф Pathfinder з датчиками Unattended Ground Sensors (E-UGS) виробництва компанії BSS (Broadfield security services) спроможне проводити обробку отриманих сейсмічних даних самостійно і надсилати результати класифікації до корінного концентратора.



Рисунок 1.9. Зовнішній вигляд датчиків Pathfinder E-UGS

Технічні характеристики датчиків Pathfinder E-UGS (рис. 1.9) є такими:

- Розміри – 6,6x19,6см;
- Батарея – Li-ion 3,6В;
- Період автономної роботи – 24 місяці;
- Споживання в стані Вимк. – < 2 мкА;
- Споживання в режимі сканування – < 600 мкА;
- Споживання під період передачі – < 250 мА;
- Робоча частота радіомодуля – 916 МГц.

Перевагами датчика Pathfinder (E-UGS) є спроможність навчання алгоритму ідентифікації руху об'єктів і простота в встановленні на місцевості. Датчик містить захист з вологи і пилу. Недоліком датчику можливо вважати відносно невеликий період автономної роботи і необхідність заміни батареї [4]. Система Pathfinder складається з безпосередньо сейсмометра і приймача Pathfinder receiver. Перевагами цього пристрою є високі показники гнучкості

застосування. Адже пристрій спроможне існувати задіяний у різних комбінаціях співпраці з іншим обладнанням цього виробника. Система спроможне працювати з декількома приймачами. Приймач відповідає військовому стандарту MIL-STD-810. Недоліками системи є необхідність працювати крізь комплектний приймач і її висока вартість.

1.3 Аналіз технічного завдання на проектування

У рамках дипломного проекту, згідно обраної теми, треба розглянути існуючі технології, щоб обирати оптимальне апаратне забезпечення, провести порівняння технологічних рішень, обирати необхідне програмне забезпечення і середовище розроблювання задля реалізації програмної частини проекту.

Наступним кроком треба розробити блок-схему алгоритму задля виконання поставленої задачі. У ході розроблювання блок-схеми алгоритму потрібно реалізувати методи зчитування і опрацювання даних з давачів і їх відображення, створити схему пристрою.

Завершальним етапом дипломного проектування є реалізація системи спостереження поза сейсмічною активністю на основі платформи STM-32, яка містить виконувати такі функції:

- зчитування даних з акселерометру;
- усунення похибок вимірювання і перетворення даних;
- обробка даних і їх відображення, створення графіку (сейсмограми).

Опісля розроблювання системи треба провести її тестування на основі платформи STM-32 (див. технічне завдання).

1.4 Вибір і аналіз апаратного забезпечення проекту

У цьому підрозділі треба обирати і розглянути необхідне апаратне забезпечення задля подальшої реалізації проекту системи спостереження поза сейсмічною активністю. Задля визначення сейсмічної активності на поверхні в даному проекті передбачається застосування давачів лінійного ускорення (акселерометрів).

1.4.1 Огляд і порівняння давачів лінійного ускорення

Задля визначення лінійного ускорення використовують акселерометри. У залежності з необхідної кількості напрямів лінійного ускорення, акселерометри поділяються на одно-, двох-, і трьохосові. Найчастіше використовуються трьохосові, оскільки вони більш універсальні і дозволяються реалізовувати задачі з будь-якою кількістю осей.

Акселерометри поділяються на такі класи:

- п'єзоелектричні;
- поверхнево-інтегральні;
- об'ємні інтегральні.

П'єзоелектричний МК являє собою стрижень, котрий створює тиск на п'єзокристал. Вироблення електричного струму здійснюється під період вібрації, показники ускорення визначаються вимірюваними значеннями сили струму. Такі датчики мають недолік – високий ступінь похибки. Датчики відрізняються поміж собою своїми параметрами конфігурації і залежать з таких факторів як:

- температура;
- тиск;
- вологість.

Об'ємні інтегральні датчики складаються зі сплавлених поміж собою двох пластин кремнію. Три тонкі кремнієві смужки, до яких приєднана інерційна маса, розміщуються на пластині. З одного боку до кремнієвої рамки приєднаний вагомий важок. Під період руху датчику в просторі з певним прискоренням, маса починає рух у різні сторони, деформуючи п'єзорезистори, котрі утворюють поміж собою напівміст. Схема опрацювання кристалів, яка розміщена поза межами кристалу, разом з датчиком утворюють електронний сигнал, котрий під період руху містить напругу 50-100 мВ. Зменшення розмірів кристалу дає підвищення його механічної міцності та зниження вартості. Інтегральні датчики лінійного ускорення мають декілька недоліків:

- крізь складне з'єднання об'ємних елементів з плоскими інтегральними платами вони складні в виробництві;

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

– МК містить великий розмір.

Якщо розмістити схему формування сигналу разом з датчиком, то зайнята площа збільшиться [5].

Поверхнева інтегральна схема є найбільш поширеною. Технологію застосування цієї схеми називають мікро-електромеханічними системами MEMS (рис.1.10). Робота датчиків побудована на перетворенні позначення зміщення інерційної маси у пропорційний електричний сигнал. Сутність роботи датчику полягає в наступному: МК ускорення, розміщений у центрі кристалу, складається з великої кількості конденсаторів, їх обкладки мають плоскі смужки полікремнієвої плівки. Стрижні, з'єднані на молекулярному рівні, розміщуються на 1 мікрон вище з поверхні кристала у повітрі, називаються нерухомими обкладками. Показники напруги змінюються з конфігурації конденсатора і використовуються задля обчислення показників ускорення. Конфігурацію конденсатора змінюють рухомі стрижні поміж обкладками.

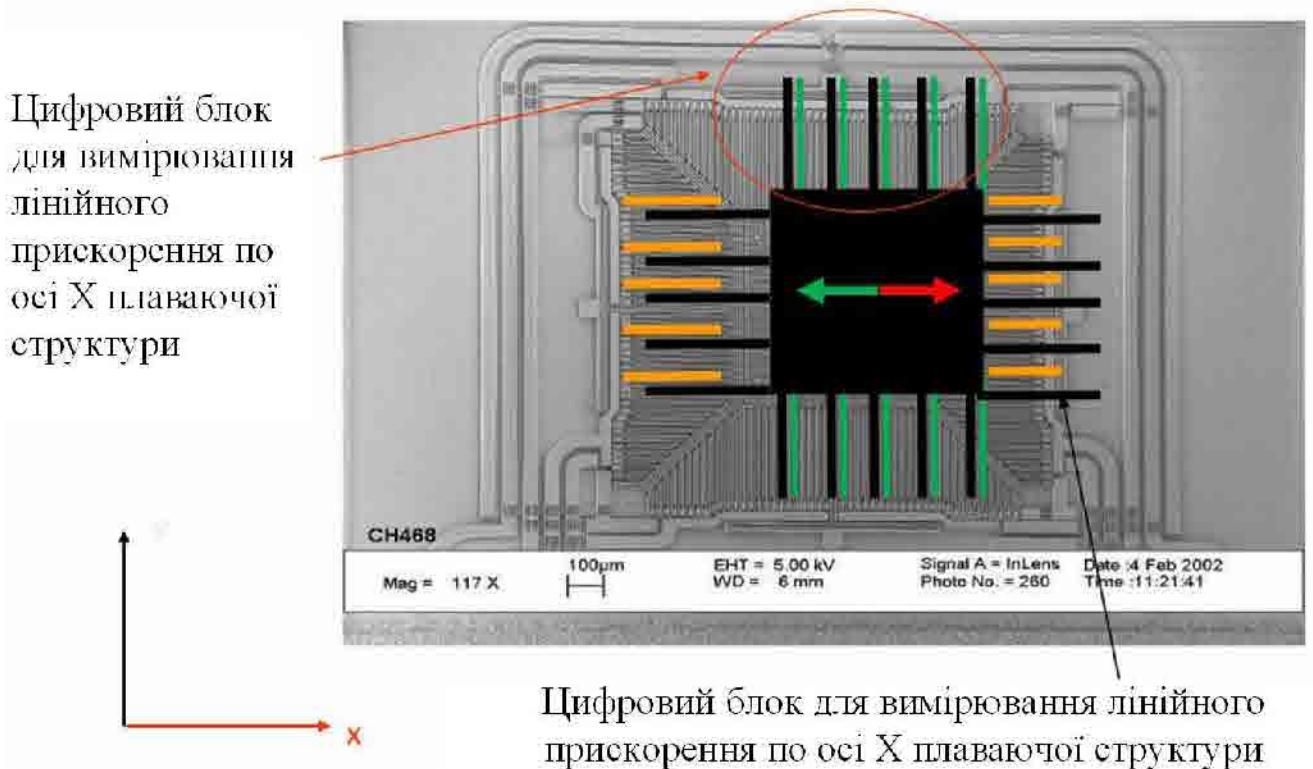


Рисунок 1.10. Структура MEMS-акселерометра

Датчики MEMS є дуже доступними задля вільного продажу і їх ціна є низькою, крізь простоту побудови та малі розміри. Ця технологія містить добрі

показники точності. Порівняння давачів лінійного ускорення представлено в табл. 1.1.

Таблиця 1.1. Порівняння давачів лінійного ускорення

Тип	Точність	Вартість	Спеціалізація	Особливості
Об'ємні інтегральні	Висока	Висока	Нахили, вібрації, інерційні сили	Малий розмір, завершена конструкція
Поверхнево-інтегральні	Середня	Середня	Нахили, вібрації, інерційні сили	Низький ступінь шуму. Великий розмір. Складно налаштовується
П'єзоелектричні	Низька	Низька	Удари, вібрації	Показники приблизні. Висока чутливість до температури і тиску

Найбільш високу точність серед розглянутих давачів містить об'ємний інтегральний МК.

1.4.2 Структура і порівняльна характеристика акселерометрів STM-32

Відповідно до технічного завдання треба порівняти існуючі поверхневі інтегральні датчики і обирати найбільш відповідну модель.

STMicroelectronics – компанія, яка є провідною на ринку мікроелектроніки. Комплектуючі, котрі були створені цією компанією, використовуються в великій кількості промислових проектів.

Дані з аналого-цифрового перетворювача потрапляють в виді 8 старших і молодших бітів. Опісля складання даних значень і проходження відновлюючого фільтру вони стають доступними задля виведення поза поміччю інтерфейсів, таких як I²C (Inter-Integrated Circuit) чи S-P-I (Serial Peripheral Interface). Всі акселерометри мають схожу функціональну структуру, яка представлена на рис. 1.11.

Датчик, котрий спроможне орієнтуватися у просторі поза поміччю магнітометра, містить назву LSM303DLHC. Його відмінністю є суміщення разом двох давачів на одному кристалі, окрім наявного акселерометра.

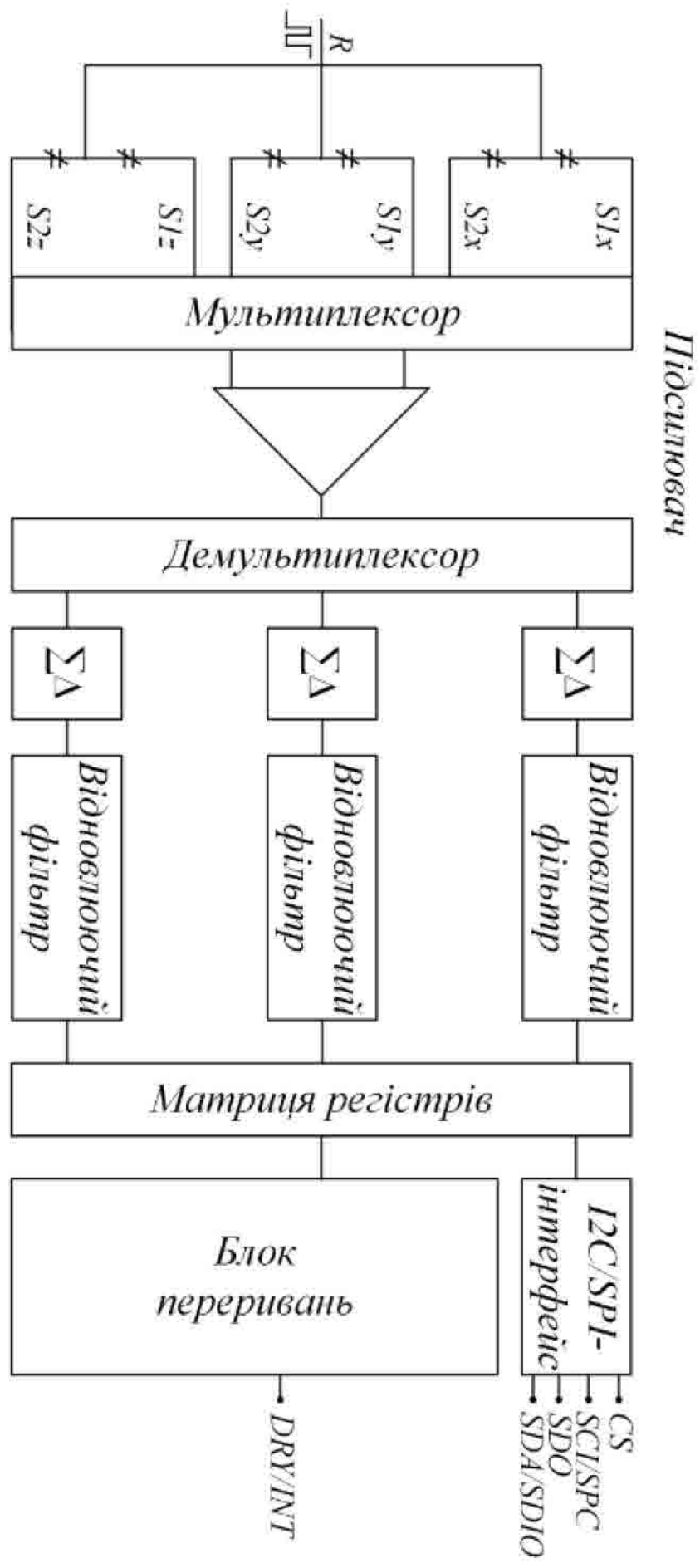


Рисунок 1.11. Функціональна схема цифрового датчику ускорення задля STM-32

Акселерометр на основі цього датчика підтримує розрахунок ускорення задля трьох осей. У його можливості входить налаштування граничних значень коливань з 2g до 16g, де g дорівнює прискоренню вільного падіння.

Передача даних з датчика здійснюється по шині поза поміччю інтерфейсу I²C. Ця шина підтримує стандартний і швидкий режим роботи та працює при напрузі 2.16 – 3.6 Вольт. Шина складається з двох двонаправлених ліній:

- послідовної лінії даних;
- послідовної лінії тактування.

Ці лінії приєднанні до живлення і керуються крізь відкритий колектор.

Акселерометр LIS3DSH є трьохосьовим датчиком лінійного ускорення, налаштовується на проміжні позначення, завдяки чому можливо економити заряд, і підтримує вимірювання у діапазонах з 2g до 16g. Цей МК надстабільний і спроможне витримувати перевантаження до 10000g. Основними його плюсами є малі розміри і висока енергоефективність. Задля роботи датчику необхідна напруга 1.71 Вольт.

Передача даних здійснюється на тактовій частоті МК, котрий використовує цей МК. При цьому застосовується синхронний зв'язок S-P-I, котрий створений задля двостороннього спілкування периферії і МК. Окрім застосування інтерфейсу S-P-I акселерометр LIS3DSH спроможне разом з цим використовувати зв'язок I²C. У інтерфейсі S-P-I завжди є один ведучий елемент, зазвичай мікросхема, та один чи декілька ведених елементів. Обмін даними здійснюється по 4 лініям зв'язку:

– MOSI (Master Out Slave In) – служить задля передачі даних з мікросхеми до ведених елементів;

– MISO (Master In Slave Out) – служить задля передачі даних з ведених елементів до мікросхеми;

– CS (Chip Select) – вибирається ведений елемент і здійснюється обмін даних з ним;

– SCLK (Serial Clock) – послідовний тактовий сигнал, котрий надсилається з МК до усіх ведених елементів.

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

Щоб обирати необхідний пристрій задля обміну даних, сигнал подається по лінії Chip Select. Частота обміну дорівнює тактовій частоті ведучого пристрою [5]. Поза поміччю зсуву останні 8 біт регістру веденого заносяться у перші 8 біт ведучого та навпаки крізь лінії зв'язку MISO і MOSI (рис. 1.12). Через це принцип роботи інтерфейсу полягає у обміні даними поміж зсувними регістрами МК і периферії. Разом з цим S-P-I надає обмінюватися даними поміж мікросхемою і одразу декількома периферійними пристроями.

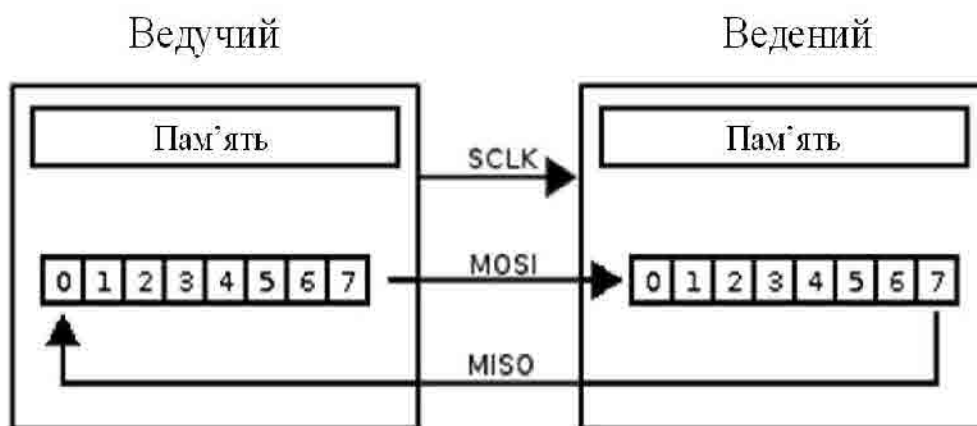


Рисунок 1.12. Типова структура зв'язків та ліній інтерфейсу S-P-I

Датчик задля STM-32, котрий разом з цим підтримує інтерфейси I²C і S-P-I називається LSM9DS1. Він на одному чіпі одразу три датчики: акселерометр, гіроскоп і магнітометр. Такий вид периферійних приладів називають інерціальними модулями.

Таблиця 1.2. Порівняльна характеристика акселерометрів компанії STM-32

Назва	Функціонал	Вольтаж	Способи передачі даних	Межі вимірювання	Чутливість
LSM303DLHC	Акселерометр, магнітометр	2.16–3.6В	I ² C	±2g/±8g/±16g	0.09g
LIS3DSH	Акселерометр	1.71–3.6В	S-P-I і I ² C	±2g/±4g/±6g / ±8g/±16g	0.06g
LSM9DS1	Акселерометр, магнітометр, гіроскоп	1.9–3.6В	S-P-I і I ² C	±2g/±4g / ±8g/±16g	0.061g

Цей МК містить найбільший функціонал, але він містить меншу енергоефективність і зменшену спроможність налаштування порогів

вимірювання, що призводить до зменшення точності вимірювання проміжних результатів. Він потребує живлення 1.9 – 3.6 Вольт. Датчик LSM9DS1 містить достойні показники задля своєї функціональності, але разом з цим містить спроможність застосування додаткових функцій, але більш спеціалізованим і енергоефективним є МК LIS3DSH (табл.1.2).

1.4.3 Вибір та обґрунтування обчислювальної платформи

Треба розглянути і обирати доречні апаратні засоби задля опрацювання інформації з давачів сейсмічної активності. Це впливає на швидкість і якість опрацювання даних, можливості і складність розроблювання системи.

Першою розглянутою платформою є Arduino. Застосування однопроцесорних цифрових плат Arduino задля отримання даних з електронних давачів є оптимальним задля початківців, котрі не мають навичок у комп'ютерній електроніці, ці платформи прості і небагато коштують. Розробники платформи Arduino створили середовище розроблювання Arduino IDE (Integrated Development Environment) і спеціалізовану C-подобну мову програмування. Завдяки надсиланню чітких команд МК, розміщеному на платі, Arduino легко програмуються. Тактування здійснюється на частоті 8 чи 16 МГц з напругою 3.3–5 Вольт. Плата містить велику кількість аналогових і цифрових входів і виходів та складається з МК Atmel AVR, котрий містить вбудований завантажувач і контакти задля приєднання додаткових елементів. Поза поміччю середовища розроблювання платформи з легкістю програмуються Arduino IDE. Програми пишуться мовою C з додатковими функціями задля реалізації вводу/виводу з контактів платформи. Зібрати проект можливо на макетній дошці, але щоб приєднати додаткові електричні деталі не потрібне паяння. Платформа Arduino містить значну кількість переваг, але саме:

- середовище розроблювання спроможне працювати з будь-якою операційною системою;
- програмне і апаратне забезпечення з відкритим кодом;
- низька вартість;

– легко програмувати.

Наступною розглянутою платформою є Raspberry Pi. Ця плата містить накопичувач в виді SD-карти. Raspberry Pi являє собою мікрокомп'ютер, у якому всі складові розміщені на одній платі. Він складається з 4-ядерного процесору ARM (Advanced RISC Machine) Cortex-A53 з частотою 700 МГц, 512 мегабайтів оперативної пам'яті і відеокарти VideoCore IV. ARM Cortex-A53 являє собою 32-бітний RISC (Reduced Instruction Set Computing) процесор, котрий використовується у портативних пристроях. Ця архітектура використовує енергоефективні технології, що є плюсом задля приладів, задля яких головним є економія енергії. На Raspberry Pi можливо встановити ОС Linux. Офіційною мовою розроблювання є мова Python. Цей апаратний засіб містить широкий спектр застосування у залежності з того, котрі периферійні пристрої приєднати до нього. Він підтримує всі необхідні периферійні пристрої задля роботи, такі як: мишки, клавіатури, монітори та т.д. Виробники цифрових приладів використовують його задля аудіосистем, метеостанцій і спеціалізованих мікрокомп'ютерів.

Наступною розглянутою платформою є STM-32 з компанії STMicroelectronics. Цифровий сигнальний процесор – це мікропроцесор, спрямований на цифрову обробку сигналів. Мікроконтролери STM-32 використовують такі процесори під назвою ARM Cortex, і вважаються найкращими задля систем опрацювання сигналів у реальному часі. Разом з цим ці мікроконтролери використовують мало енергії. У робочому стані показники струму не перевищуватиме 12 мА. Головними задачами цифрової опрацювання сигналів є:

- фільтрація і пошук сигналів;
- дискретне перетворення Фур'є.

Основними перевагами цифрових сигнальних процесорів є:

- використання генераторів адресних послідовностей;
- одночасний доступ до декількох ділянок пам'яті;
- апаратна реалізація циклічного виконання набору певних команд;

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

– виконання операцій множення з накопиченням поза один машинний цикл.

Щоб отримувати дані з акселерометра в розроблюваній системі спостереження поза сейсмічною активністю треба використовувати мікроконтролер. Опісля проведеного короткого аналітичного огляду визначено, що самим оптимальним варіантом буде платформа STM-32 з компанії STMicroelectronics адже:

- обраний акселерометр LIS3DSH належить компанії STMicroelectronics;
- в компанії STMicroelectronics наявний широкий асортимент різноманітних МК;
- програмування МК зрозуміле і виконується мовою високого ступеня;
- переналаштування на плату іншої моделі потребує незначних змін у конфігуруванні.

Більшою продуктивністю володіють мікроконтролери серії STM-32 F4, котрі побудовані на основі мікропроцесора ARM Cortex-M4. Ці мікроконтролери виготовляються з використанням унікальних задля підвищення результатів продуктивності технологій, створених на ядрі Cortex-M, котрі мають назву STM ART Accelerator. Разом з цим у мікроконтролерах цієї лінійки розміщений модуль операцій задля роботи з числами з плаваючою комою, котрий дає спроможність використовувати їх задля задач широкого профілю [7]. Мікросхеми цієї серії мають низьке енергоспоживання, що пов'язано з динамічним споживанням електроенергії. Витрати по живленню складають з 140 мкА до 238 мкА при роботі на максимальних частотах. Мікроконтролери на основі STM32F4 об'єднують продуктивність сигнальних процесорів при обробці сигналу, роботу у реальному часі і реалізацію п'яти класів сигнальних МК. Всі мікроконтролери відрізняються лише додатковим функціоналом.

1.4.4 Аналіз можливостей МК платформи STM32F4

Характеристики усіх МК STM32F4 є такими:

- підтримка цифрових сигнальних процесорів;

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

- напруга живлення 1.8...3.6 Вольт;
- сімнадцять 16- і 32- розрядних таймерів;
- порти USB (Universal Serial Bus) 2.0;
- вбудовані RC-генератори з частотою 16 МГц;
- робоча температура з -40 до 105 °С;
- підтримка інтерфейсів I²C, S-P-I;
- контролер SDIO (Secure Digital Input Output);
- 12-бітний цифро-аналоговий перетворювач;
- зовнішнє джерело переривань з 4 до 26 МГц;
- ASB (Advanced System Bus) матриця шин;
- вбудований генератор випадкових чисел;
- 32-бітний процесор ARM Cortex-M4.

Головні відмінності МК STM32F4 (рис.1.13):

- STM32F401 відрізняються тим, що містять найменше енергоспоживання і малий розмір;

- STM32F4x5 містить 1 Мбайт флеш-пам'яті з можливостями шифрування;

- STM32F4x7 містить 1 Мбайт флеш-пам'яті і зв'язок Ethernet MAC;

- STM32F4x7 містить 2 Мбайт флеш-пам'яті;

- STM32F4x9 містить 2 Мбайт флеш-пам'яті з інтерфейсом SDRAM (Synchronous Dynamic Random Access Memory), дисплей TFT LCD (Liquid Crystal Display) і спроможне досягти більшої продуктивності та меншого енергоспоживання. В мікроконтролерах STM-32 широко використовуються GPIO (General purpose input-output) – порти вводу/виводу загального призначення 16-ти розрядні. Стан виводів встановлюється програмно та вони не є входними чи вихідними сигналами якогось вузла МК. Порти поміж собою незалежні та їх стан встановлюється окремо один з одного.

Режими настроювання вихідних портів:

- Input floating – вхід без підтягуючого резистора;

- Input pull up – вхід з резистором, підключеним до живлення;

- Input pull-down – вхід з підтягуючим резистором, підключеним до загального проводу (землі);
- Analog – аналоговий вхід;
- Output open-drain – вихід з відкритим стоком. Функціонально аналогічний виходу з відкритим колектором. При низькому логічному рівні замикає контакт на землю;
- Output push-pull – звичайний активний вихід. При низькому логічному рівні видає напругу, яка на контакті дорівнює нулю, але при високому – подається напруга, близька до живлення МК, зазвичай +3 Вольт;
- Alternate function push-pull – альтернативна функція виведення у звичайному (активному) режимі;
- Alternate function open-drain – альтернативна функція виведення у режимі відкритого стоку.

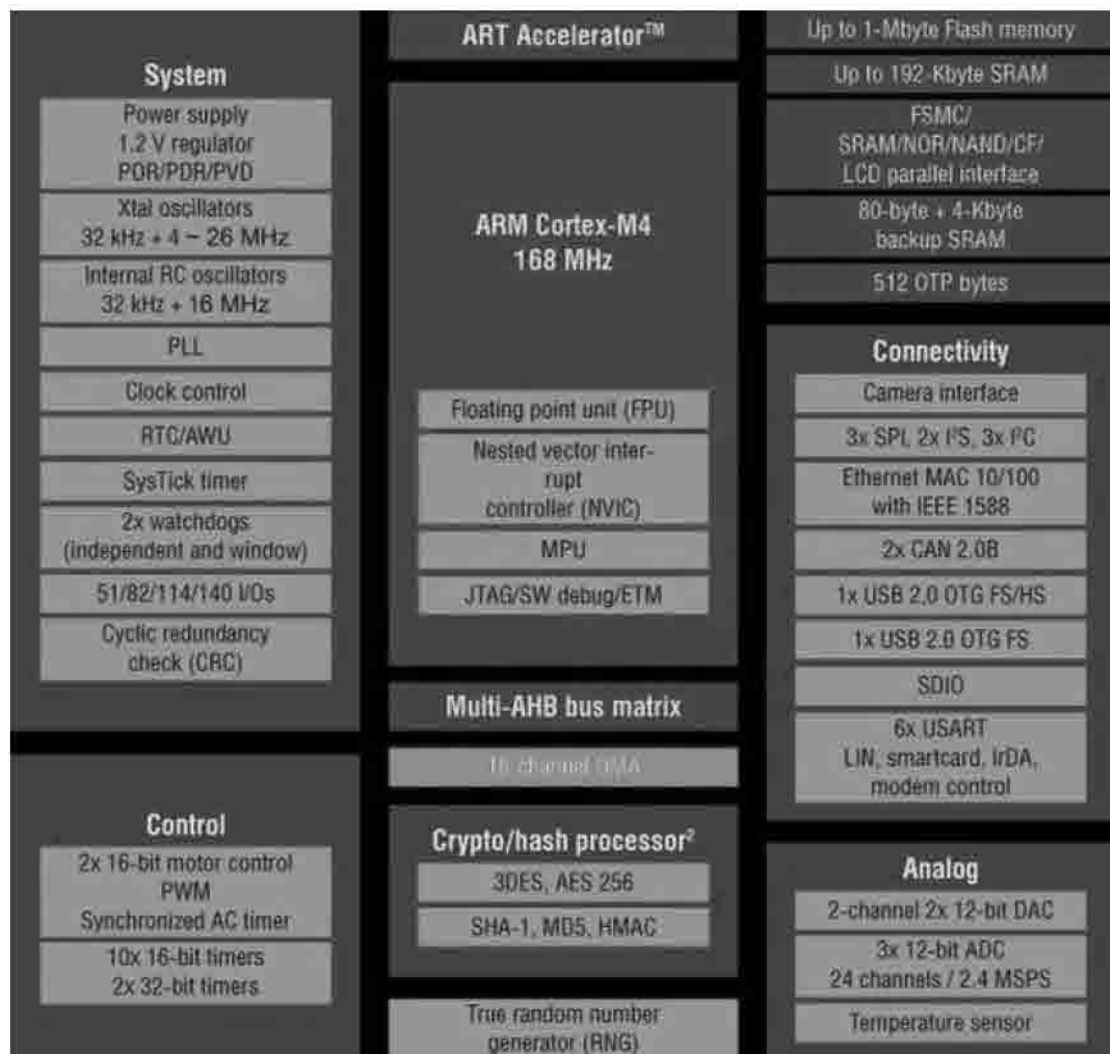


Рисунок 1.13. Основні характеристики МК STM32F4

Задля активізації системи захисту конфігураційних параметрів порту треба виконати певну послідовність дій над регістрами конфігурації, але розблокування здійснюється тільки опісля скидання цих конфігурацій.

Аналогово-цифрові перетворювачі мають високу швидкість перетворення, але загалом існує 24 аналогових канали. Всі мікроконтролери цієї лінійки мають три аналого-цифрових перетворювача і два цифро-аналогових перетворювача.

Формування стереосигналу можливе через це, що ЦАП містить два канали і володіє роздільністю у 12 біт. Перетворення здійснюється у 8/12-бітовому форматі з можливістю вирівнювання по краю.

Виходи даних чотирьохпровідного інтерфейсу JTAG (Join Test Action Group) і двопровідного SWD (Serial Wire Debug), котрі містить мікроконтролер, мультиплексовані поміж собою і служать задля підключення до налагоджувача.

Платформа STM-32-F-407-VG містить найбільш оптимальний мікроконтролер з усієї серії. STM-32-F-407-VG є налагоджувальною платою з програматором–налагоджувачем ST-Link. У платі встановлена різноманітна периферія, але саме:

- мікрофон;
- USB;
- акселерометр;
- аудіокодек з роз'ємом задля навушників чи акустики.

Цей мікроконтролер базується на 32-бітному ядрі ARM Cortex-M4, котрий працює на частоті 168 МГц. Ядро Cortex-M4 містить арифметичний елемент, котрий підтримує інструкції задля роботи з плаваючою комою. Всі характеристики МК платформи STM-32-F-407-VG представлені на рис. 1.14.

Щоб почати програмувати плату STM-32-F-407-VG, треба підключити її до комп'ютера поза поміччю спеціального кабелю miniUSB і встановити середовище розроблювання. Разом з цим передбачено зв'язок Ethernet, котрий можливо використовувати задля обміну даними. На платі розташовано вбудований акселерометр LIS3DSH, що економить кількість зайнятих входів і виходів на платі та надає спроможність їх подальшого застосування.

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Функціонал МК програмувати STM-32-F-407-VG достатньо широкий і повністю підходить задля подальшого створення системи спостереження поза сейсмічною активністю.

STM32F407

System	ART Accelerator™	Up to 1-Mbyte Flash memory
Power supply 1.2 V regulator POR/PDR/PVD	Arm® Cortex®-M4 CPU 168 MHz	Up to 192-Kbyte SRAM
Xtal oscillators 32 kHz + 4 ~26 MHz		FSMC/ SRAM/NOR/NAND/CF/ LCD parallel interface
Internal RC oscillators 32 kHz + 16 MHz		80-byte + 4-Kbyte backup SRAM
PLL		Connectivity
Clock control		Camera interface
RTC/AWU		3x SPI, 2x I ² S, 3x I ² C
SysTick timer		Ethernet MAC 10/100 with IEEE 1588
2x watchdogs (independent and window)		2x CAN 2.0B
51/82/114/140 I/Os		1x USB 2.0 OTG FS/HS
Cyclic redundancy check (CRC)		1x USB 2.0 OTG FS
Control	Memory Protection Unit (MPU)	SDIO
10x 16-bit timer	Multi-AHB bus matrix	6x USART LIN, smartcard, IrDA, modem control
2x 16-bit motor control PWM synchronized AC timer	16-channel DMA with Batch Acquisition Mode (BAM)	Analog
2x 32-bit timer	True random number generator (RNG)	2-channel 2x 12-bit DAC
		3x 12-bit ADC 24 channels/2.4 MSPS
		Temperature sensor

Рисунок 1.14. Основні характеристики МК STM-32-F-407-VG

1.5 Вибір програмного забезпечення задля програмування МК STM-32

В даному підрозділі треба розглянути і обирати необхідну мову програмування і середовище розроблювання задля програмування апаратного засобу, але разом з цим методи взаємодії МК з інтерфейсами. Самими важливими задля будь-якої розроблювання програмного забезпечення є такі елементи:

- мова програмування;
- бібліотеки і операційна система;
- середовище програмування, яке містить вбудований компілятор;
- система автоматизованого проектування.

1.5.1 Середовище розроблювання програмного забезпечення задля МК Keil uVision

Середовище програмування Keil uVision призначене задля розроблювання програмного забезпечення МК STM-32. Це комплексне програмне рішення, яке представляє з себе набір утиліт, необхідних задля забезпечення усіх потреб під період програмування МК. Keil uVision містить зрозумілий інтуїтивний зв'язок, але всі необхідні функції можливо знайти у контекстному меню.

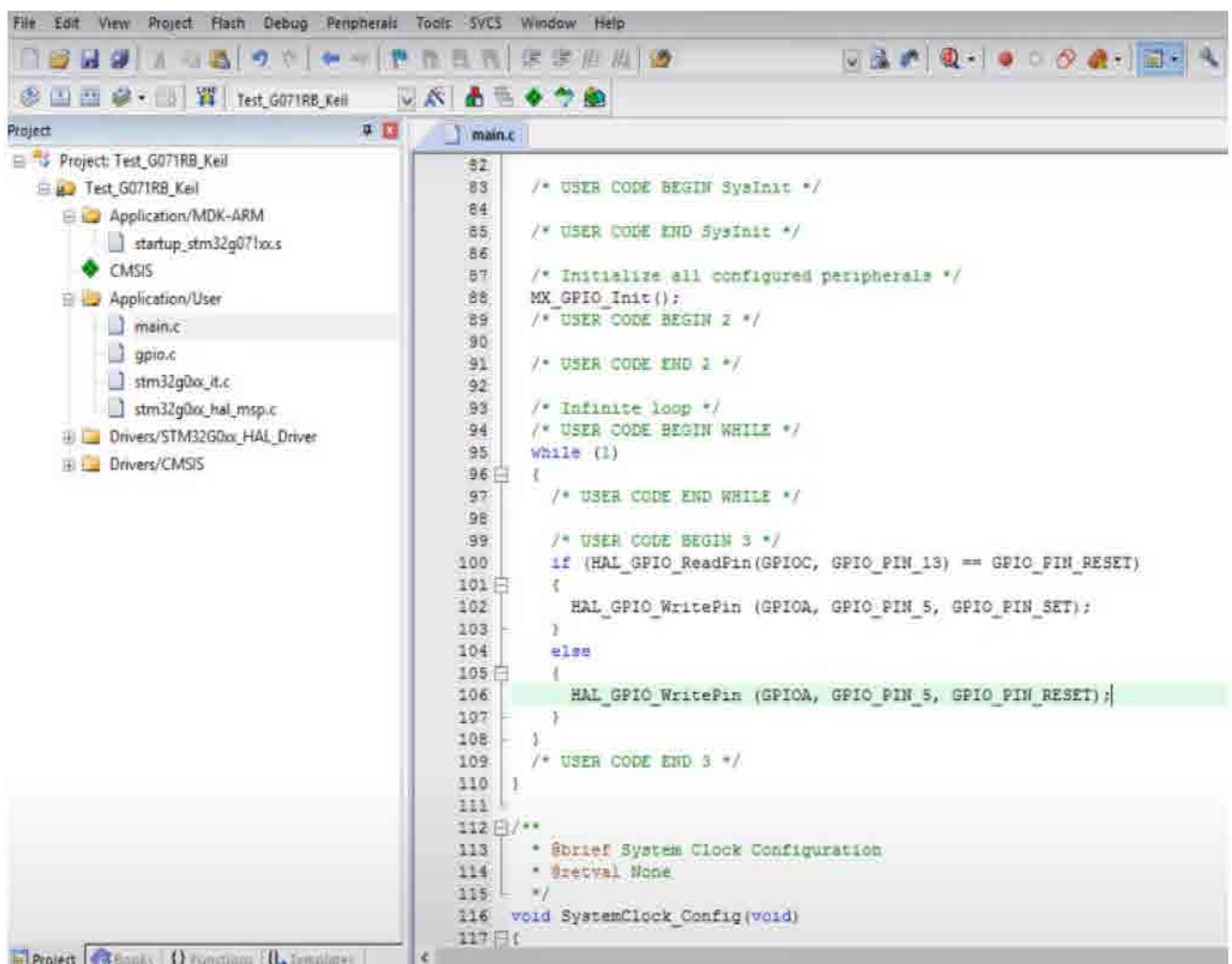


Рисунок 1.15. Середовище розроблювання Keil uVision

Серед основних складових програмних засобів середовища Keil uVision

(рис.1.15) можливо відзначити:

– базу даних МК, що містить докладну інформацію про всі підтримувані пристрої. В цієї основі зберігаються їх конфігураційні дані та посилання на джерела інформації з додатковими технічними описами. При додаванні нового пристрою у проект всі його унікальні опції встановлюються автоматично;

– менеджер проектів, котрий служить задля об'єднання окремих текстів програмних модулів та файлів у групи, що оброблюються поза єдиними правилами. Подібне угруповання надає набагато краще орієнтуватися серед безлічі файлів;

– вбудований редактор, котрий полегшує роботу з вихідним текстом поза рахунок застосування багатівіконного інтерфейсу, виділення синтаксичних елементів іншим шрифтом і кольором. Існує опція індивідуального настроювання. Редагування залишається доступним та під період налагодження програми, що надає відразу виправляти помилки чи відзначати проблемні ділянки коду;

– засоби автоматичної компіляції і компонування проекту, котрі призначенні задля створення виконавчого модуля програми. При цьому поміж файлами автоматично генеруються нові асемблерні та компіляторні зв'язки, котрі у подальшому дозволяють обробляти тільки ті файли, у яких відбулися зміни чи файли, що знаходяться у залежності з вже змінених. Функція глобальної оптимізації проекту надає досягти найкращого застосування регістрів МК шляхом неодноразової компіляції вихідного коду. Компілятори Keiul uVision працюють з текстами, написаними на мові C чи асемблері задля контролерів сімейства ARM, MSC51, C166 і ін. Крім того, можливе застосування компіляторів інших виробників;

– відлагоджувач-симулятор, що відлагоджує програму, яка була скомпільована на віртуальній моделі мікропроцесора. Моделюється робота ядра контролера та його периферійного обладнання: портів введення-виведення, таймерів і контролерів переривань. Задля полегшення комплексного налагодження розроблювального програмного забезпечення можливе

підключення програмних моделей нестандартного обладнання;

– додаткові утиліти, що полегшують виконання найбільш поширених завдань. Існують такі додаткові утиліти:

– Source Browser містить базу даних програмних символів задля швидкого пошуку;

– Find in Files призначений задля пошуку заданого коду в усіх файлах зазначеної папки чи проекту;

– Tools Menu надає використовувати утиліти сторонніх виробників;

– PC-Lint аналізує вихідний текст програми і виділяє ділянки коду, де є помилки;

– Flash tool програмує FLASH-пам'ять МК.

Застосування цього середовища розроблювання надає спрощувати процес пошуку помилок, тестування і настроювання МК задля коректної роботи, але разом з цим надає спроможність виконувати код по рядкам, переглядати вміст внутрішніх регістрів під період налагодження МК.

1.5.2 Середовище розроблювання програмного забезпечення задля МК Stm-32-cube-mx

Існує інше спеціалізоване середовище розроблювання з компанії STMicroelectronics під назвою Stm-32-cube-mx. Це графічний інструмент задля генерації коду на мові C задля ядра ARM Cortex-M і задля швидкого конфігурування мікропроцесорів і МК STM-32. Він надає налаштовувати пристрої власної розроблювання в зручному візуальному режимі. Перед початком роботи з мікроконтролером треба налаштувати і конфігурувати плату. Треба встановити тактову частоту, активувати роботу усіх необхідних інтерфейсів і визначати роботу портів введення-виведення задля подальшої розроблювання програмного забезпечення. Робота з програмою дуже проста і зрозуміла задля користувачів, котрі тільки починають ознайомлюватися з програмуванням МК STM-32.

У Stm-32-cube-mx (рис.1.16) є база даних, де зберігаються базові настроювання задля усіх існуючих МК серії STM-32. Перед початком роботи з

У даному пакеті разом з цим містяться драйвери задля роботи з зовнішніми пристроями і драйвер BSP (Board Support Package) задля роботи з елементами, розміщеними на конкретній платі, але разом з цим реалізовані інтерфейси задля роботи з периферією. Драйвери цього ступеня являють собою набір інтерфейсів API (Application Programming Interface), котрі створені задля взаємодії МК з верхніми рівнями архітектури (рис. 1.18).

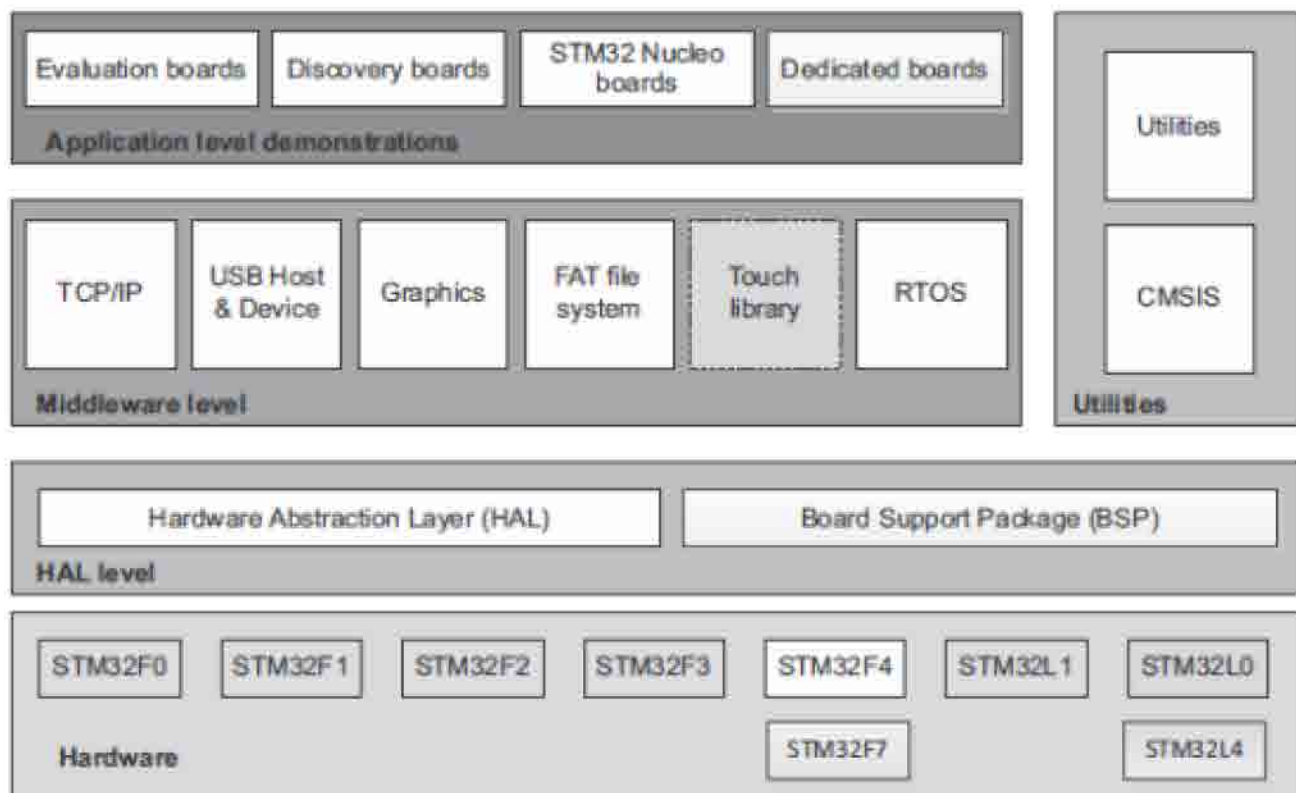


Рисунок 1.18. Три ступеня архітектури STM32CubeF4_FirmWare

При такій архітектурі верхній ступінь не прив'язаний до МК, але звертається до нього лише крізь бібліотеку драйверів. Застосування цієї структури надає спроможність у інших проектах використовувати той самий код. Драйвери нульового ступеня надають набори API, котрі спрощують реалізацію взаємодії з мікроконтролером.

Драйвери низького ступеня знаходяться на найближчому рівні до апаратного забезпечення. Він реалізує API низького ступеня (регістрів) задля кращої оптимізації.

Нульовий ступінь перевіряє отримані позначення задля усіх своїх функцій, але разом з цим у ньому реалізована динамічна перевірка на помилки під період

виконання, що значно прискорює розробку застосунків.

У бібліотеці є великий набір методів задля роботи з пристроями комунікації. У ній містяться інтерфейси ініціалізації та конфігурування пристрою, керування обміном даних, переривання і управління помилками зв'язку. Рівень HAL створений задля функцій, котрі виконуються верхніми рівнями поза рахунок застосовування проміжного ступеня. API-інтерфейси реалізують спільні функції і інтерфейси задля усіх приладів серії МК і складаються з елементів, котрі спрощують роботу необхідну задля приладів. Middleware Level – набір бібліотек на першому рівні, де взаємодія між його компонентами здійснюється завдяки виклику API. На цьому рівні розміщені компоненти проміжного ступеня. Другий ступінь відображує у реальному часі перший і низький ступінь додатків.

1.6 Застосування мови С задля настроювання зовнішніх приладів

Застосування мови програмування С є основною задля розроблювання програмного забезпечення МК. Ця мова системного програмування використовується задля низькорівневого і високорівневого програмування та є універсальною. Вона є найпоширенішою мовою програмування МК і периферійних приладів. Є підтримка принципів процедурного програмування, але треба уникати конструкцій мови, котрі несуть додаткові витрати. Основною задачею цієї мови програмування є спрощення написання великих програм зі зменшенням кількості помилок у порівнянні з асемблером.

Мова С містить високий ступінь переносимості, через це програми, написані цією мовою, гарно компілюються на пристроях з різними архітектурами, але самі компілятори створюються з підтримкою безлічі платформ. Мова програмування С містить багато переваг та особливостей, але у той же період містить та недоліки, зокрема:

- підтримує принципи процедурного програмування;
- швидко працює з однотипними операціями;
- велика і проста мовна база;

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

- прямий доступ до пам'яті;
- велика кількість бібліотек, котрі надають багато можливостей;
- невелика синтаксична база;
- спроможність передавання параметрів, використовуючи вказівники;
- наявність областей видимості.

Недоліки:

- немає засобів об'єктно-орієнтованого програмування;
- відсутнє керування пам'яттю;
- неможна повернути більше одного позначення з однієї функції;
- немає вкладених функцій.

Всі програми, написані цією мовою, представляють з себе набір файлів з кодом, котрі можуть існувати перекомпільовані у об'єктні файли. Ці файли компонуються поміж собою і з файлами зовнішніх бібліотек та утворюють бібліотеки. Зв'язок файлів поміж собою і бібліотеками вимагає явного опису використовуваних функцій, зовнішніх змінних та необхідних типів даних у кожному файлі. Поза поміччю директиви `#include` підключаються дані з окремих заголовкових файлів. Ця директива підставляє текст іншого файлу у той, у якому вона була викликана, та весь код об'єднується у одному файлі.

Мова C дуже добре оптимізована, але код, створений цією мовою, працює швидше і ефективніше у порівняння з мовами високого ступеня. По рівню швидкодії C поступається лише коду, котрий написаний на асемблері через це, що він надсилає команди напряму до процесора. Мова C надає ефективно реалізувати компілятори задля різноманітних платформ, але разом з цим відчувати, як саме буде виконуватись програма. Всі файли, котрі об'єднані поза поміччю директиви `#include`, можуть існувати:

- файлами з вихідним кодом;
- об'єктними бібліотеками з необхідними заголовковими файлами;
- динамічними бібліотеками;
- статичними бібліотеками;
- набором декількох заголовкових файлів.

Мова програмування C містить спроможність створення кросплатформних програм, котрі можуть працювати з різними операційними системами. Ця мова використовується при розробці високопродуктивних програм з ретельною перевіркою на помилки, через це вибір цієї мови програмування є найбільш оптимальним задля написання програмного забезпечення обраного МК.

1.7 Розробка структури системи спостереження поза сейсмічною активністю

Розумне місто (Smart City) є ефективною інтеграцією фізичних, цифрових та людських систем у штучному середовищі заради сталого, благополучного та всебічного майбутнього задля громадян. Розумне місто разом з цим є єдиною системою, у якій органічно взаємопов'язані міські комунікації, інформаційні технології передачі даних і пристрої IoT (інтернет речей) [5]. Метою створення «розумного міста» є покращення і спрощення управління містом, благоустрій міського середовища, забезпечення безпеки і підвищення якості життя жителів міста. У концепції Smart City всі елементи спілкуються поміж собою, багато давачів посилають інформацію один одному.

В даному проекті розроблюване програмно-апаратне забезпечення в майбутньому буде мати спроможність вбудовування у систему Smart City.

Багато проектів, котрі використовують концепцію Smart City, реалізуються з використанням МК компанії STM. Разом з цим ця компанія створила інтегровані середовища розроблювання, що дозволяють економити період протягом створення програмного забезпечення.

Крізь високу функціональність і енергоефективність була обрана платформа STM-32-F-407-VG з мікроконтролером сімейства STM-32 Cortex-M4. Створюване програмне забезпечення можливо буде з легкістю перенести на інший мікроконтролер даного сімейства шляхом зміни конфігурації самого МК і портів введення-виведення. Поза поміччю порту USB, котрий розміщений на платі, можливо передавати дані на комп'ютер поза поміччю послідовного COM-порту (рис. 1.19). Програма керування задля МК буде мати спроможність зчитувати і оброблювати інформацію з COM-порту, але потім виводити цю

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

інформацію на екран комп'ютера.

Зв'язок S-P-I був розроблений задля передачі даних поміж зовнішніми пристроями і мікроконтролерами. Кожна передача синхронізована з тактовим сигналом, котрий генерується ведучим пристроєм (мікроконтролером).

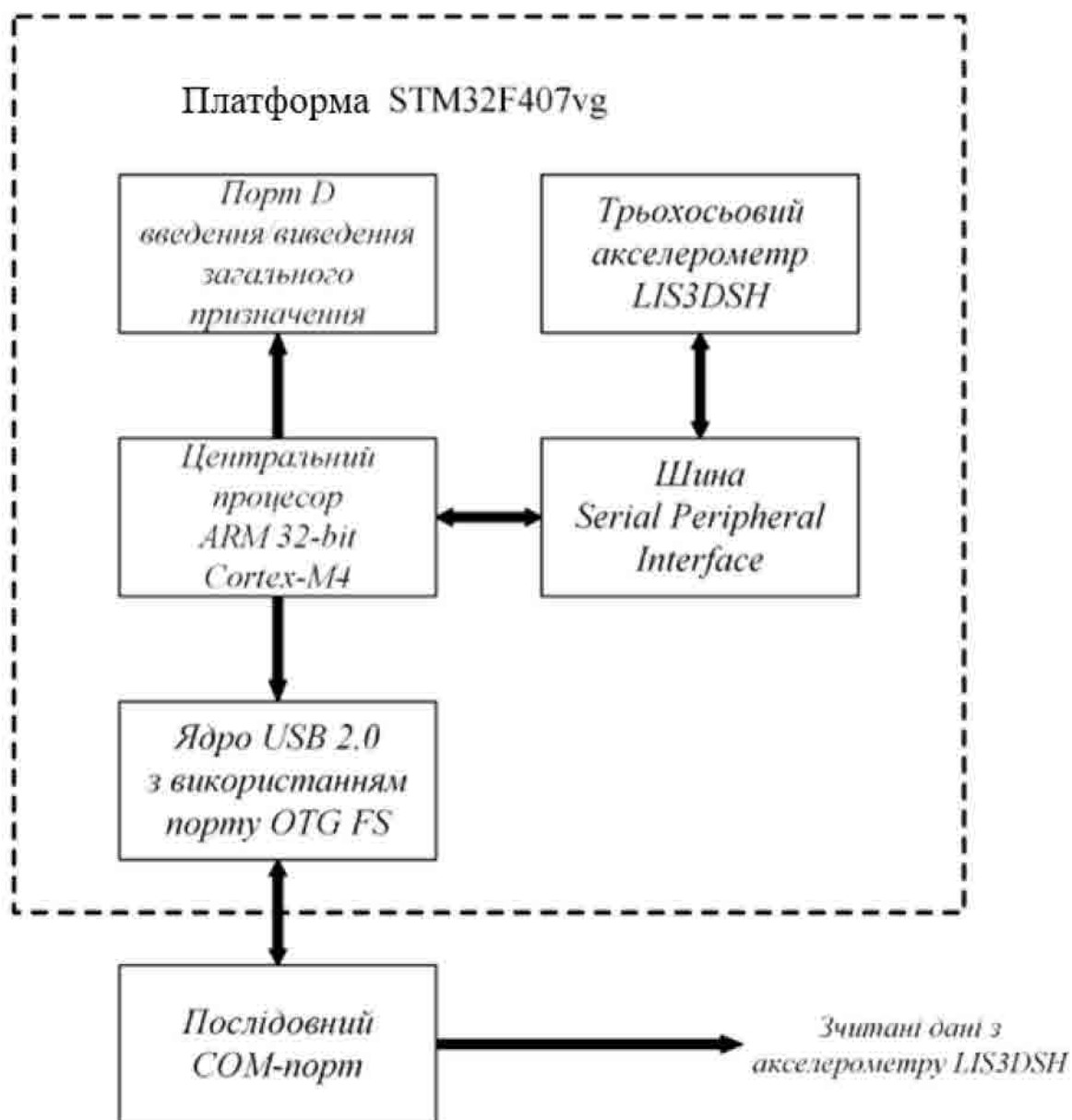


Рисунок 1.19. Структурна схема апаратних засобів системи спостереження поза сейсмічною активністю

Поза поміччю інтерфейсу S-P-I акселерометр LIS3DSH спроможне передавати інформацію до МК. Цей МК разом з обраним мікроконтролером, добре працюють поміж собою через це, що акселерометр LIS3DSH вже вбудований у плату платформи STM-32-F-407-VG.

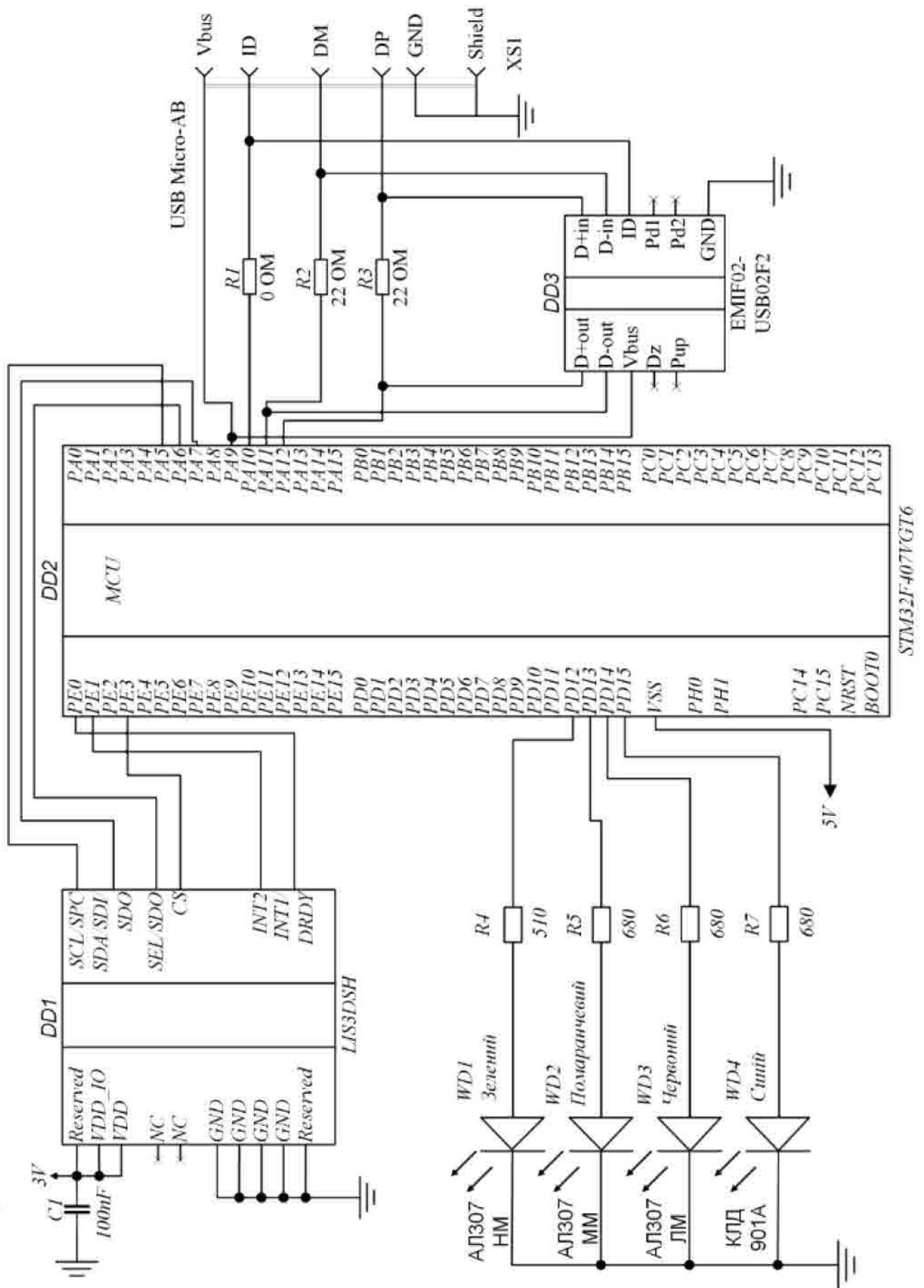


Рисунок 1.20. Принципова електрична схема апаратних засобів системи спостереження поза сейсмічною активністю

1.8 Реалізація взаємодії складових системи спостереження поза сейсмічною активністю

Виконати налаштування та конфігурування МК, портів, інтерфейсів платформи STM-32-F-407-VG системи спостереження поза сейсмічною активністю дозволить розглянуте вище середовище розроблення Stm-32-cube-mx. Перевірка працездатності платформи STM-32-F-407-VG при підключенні зовнішніх приладів буде відбуватися під період підключення МК до комп'ютеру крізь USB-з'єднання. Якщо опісля підключення системи до живлення повідомлень про помилки немає (встановлені відповідні драйвери), система починає налаштування і запуск необхідних задля роботи інтерфейсів. Якщо під період перевірки буде виявлена помилка, система зупинить роботу та на платі STM-32-F-407-VG можливо буде побачити запалений червоний світлодіод, котрий буде повідомляти про невірні налаштування системи.

З початкового положення платформи STM-32-F-407-VG залежить точність отримуваних даних. Чотири кольорові світлодіоди по центру платформи будуть вказувати на нахил платформи та зміну її положення. Відповідно поза цими індикаторами можливо буде визначати сейсмічну активність. Задля корекції положення треба нахилити плату в протилежну з світлодіода сторону.

Поза поміччю порту USB, розміщеного на платі, дані з акселерометру про сейсмічну активність крізь послідовний COM-порт будуть передаватися на комп'ютер в виді бітових значень. Зчитана інформація з COM-порту містить оброблюватися і приводитися до десяткових значень ускорення вільного падіння, сили та напрямку вібрації. Опісля цього вже оброблені дані мають відображуватися на екрані підключеного комп'ютера в графічному виді з використанням алгоритмів, котрі дозволяють визначити ступінь сейсмічної активності. Графічне поле буде мати межі допустимих значень, при яких землетрус не є суттєвим поза шкалою Ріхтера. Опісля оброблення усіх необхідних даних кінцевим результатом буде їх графічне представлення, завдяки якому можливо буде побачити сейсмічну активність поза певний період. Динамічний графік буде відображувати дані ускорення, але разом з цим служити

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

індикатором сейсмічної активності (землетрусу), котрий червоніє, якщо ускорення різко змінюється.

1.9 Настроювання МК в Stm-32-cube-mx

Перед початком реалізації програмного забезпечення системи спостереження поза сейсмічною активністю треба налаштувати мікроконтролер платформи STM-32-F-407-VG в середовищі розроблювання Stm-32-cube-mx. Разом з цим, задля того щоб отримувати дані з платформи STM-32-F-407-VG на комп'ютер, треба налаштувати USB-порт.

Перед початком створення програмного забезпечення треба налаштувати мікроконтролер STM-32 і дерево тактування (Clock Configuration). Опісля запуску середовища Stm-32-cube-mx треба створити новий проект, обравши необхідний мікроконтролер задля подальшого настроювання (рис. 1.21).

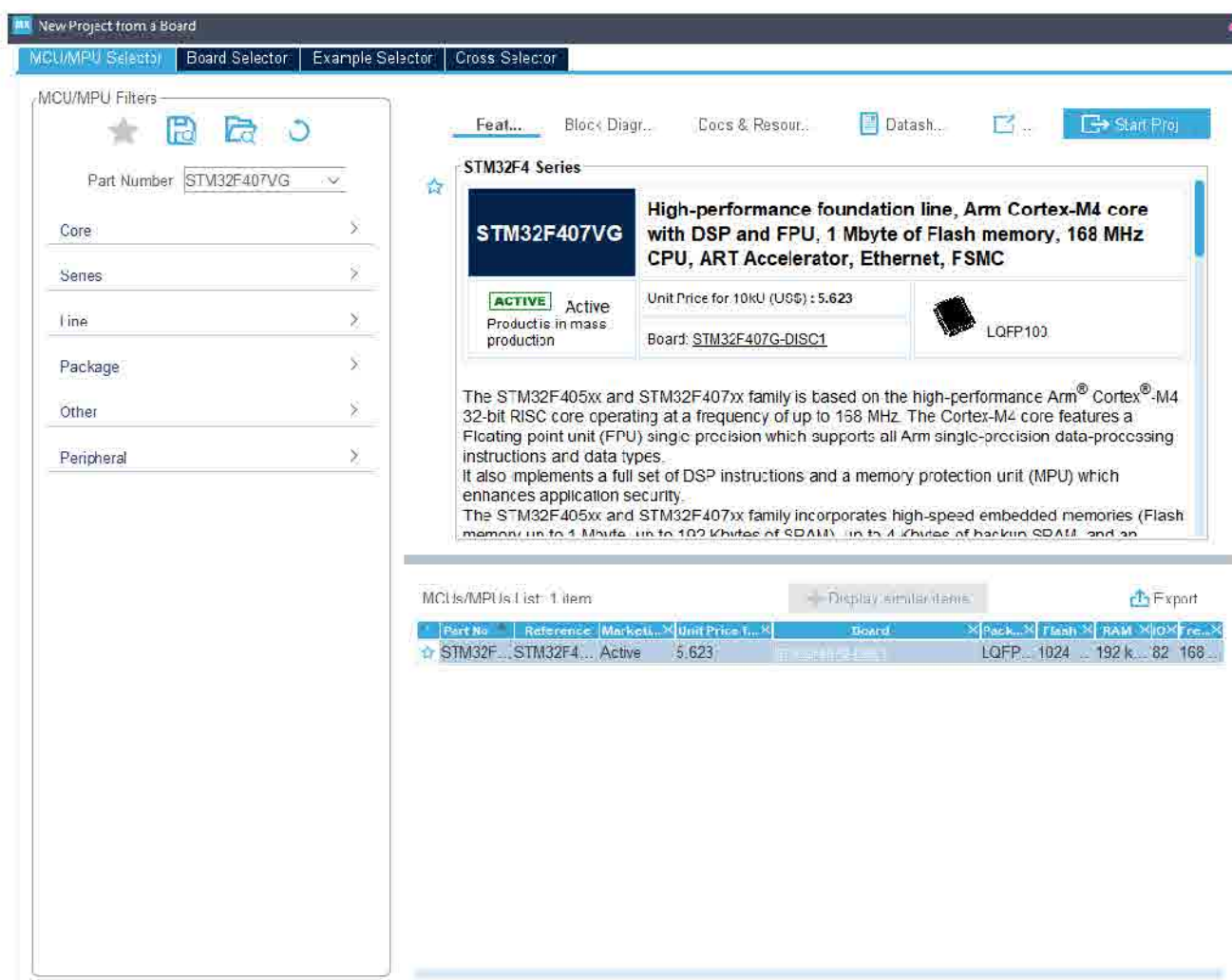


Рисунок 1.21. Меню вибору МК в Stm-32-cube-mx

Зм.	Арк.	№ докум.	Підпис	Дата

КС 57. 22 000. 00 ДП ПЗ

Арк.

41

Опісля натискання в Stm-32-cube-mx кнопки Start Project відкривається головне меню, де можливо знайти всі необхідні параметри задля подальшого настроювання платформи (рис.1.22).

Треба відключити зв'язок I²C і підключити SPI1, на якому розпаяний МК LIS3DSH. Задля цього треба обирати режим Full-Duplex Master. В розділі Clock Parameters, пункт Prescaler, треба встановити позначення на 16, що відповідає поза швидкість передачі. Порти PE0 і PE1, котрі відповідають поза виявлення переривань, треба налаштувати як входи. Порт PE3, котрий відповідає поза вибір чіпу, треба налаштувати як вихід. Треба змінити максимальну швидкість порту PE3 на Medium в пункті GPIO. Задля відстежування значень з датчику ускорення і керування різнокольоровими світлодіодами на платі треба налаштувати порти PD12, PD13, PD14, PD15 як виходи. В пункті Project Manager треба змінити позначення стеку і купи на 0x4000 і 0x2000 відповідно, щоб USB-пристрій опісля встановлення драйвера не видавав помилку. В розділі USB_DEVICE треба встановити позначення COM-порту. Встановлені режими портів введення-виведення показані на рис.1.23.

Задля налаштувати дерева тактування треба обирати Clock Configuration і встановити позначення відповідно до рис.1.24.

1.10 Програмування МК в Keil uVision

Опісля встановлення середовища розроблювання Keil uVision в Stm-32-cube-mx треба обирати IDE (Integrated Drive Electronics) mdk-arm версії 5 і згенерувати код (рис.1.25). Опісля складання проекту утворюються наступні файли:

- main.h – заголовний файл, котрий містить макропідстановки, необхідні задля запалювання світлодіодів;
- main.c – виконавчий файл, що викликається під період запуску програми;
- stm32f4xx_it.c – виконавчий файл, що відповідає поза переривання;
- usbd_cdc_if.c – виконавчий файл, котрий містить у собі зв'язок задля COM-порту.

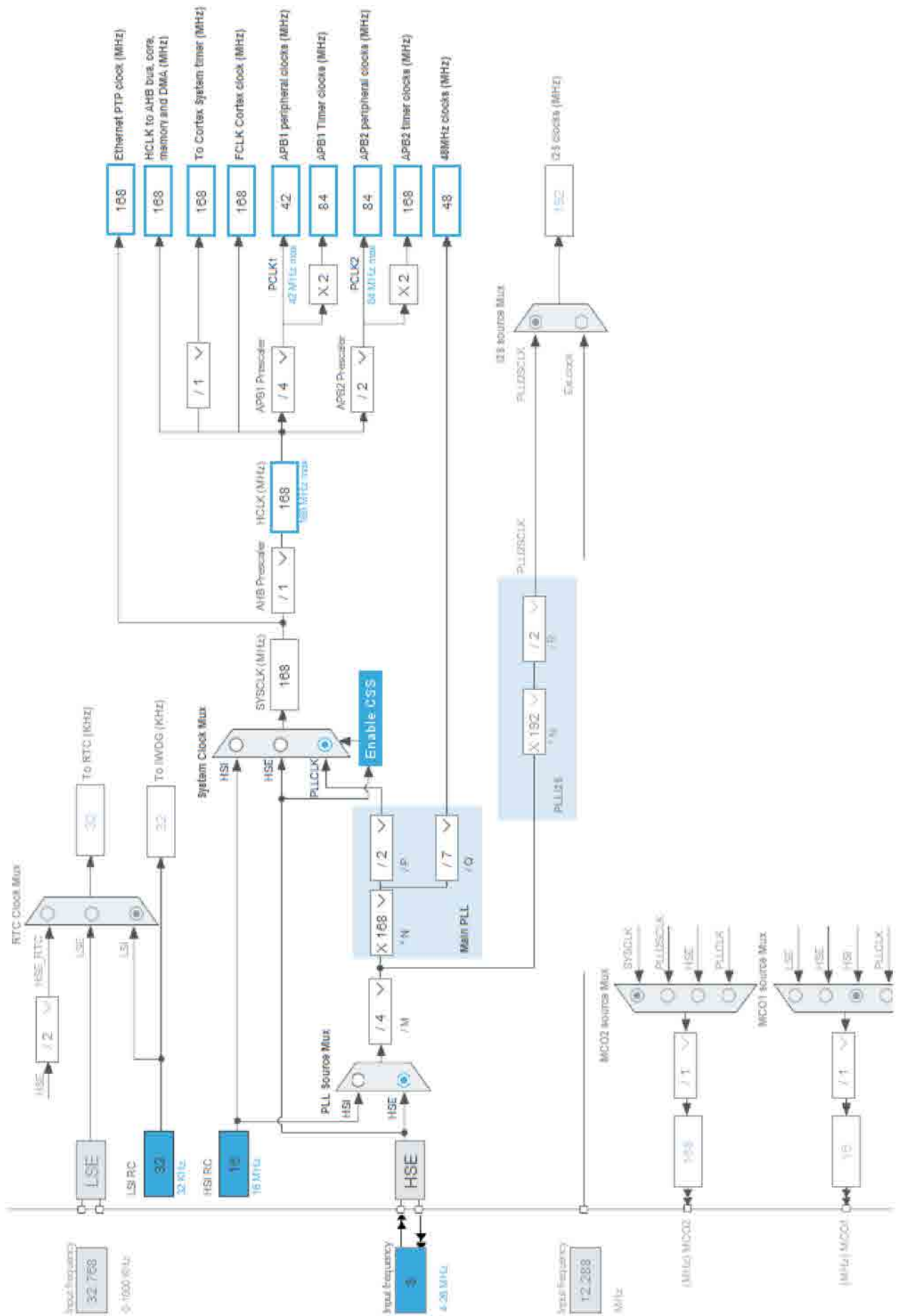


Рисунок 1.24. Настроювання дерева тактування Clock Configuration

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

КС 57. 22 000. 00 ДП ПЗ

Арк.

44

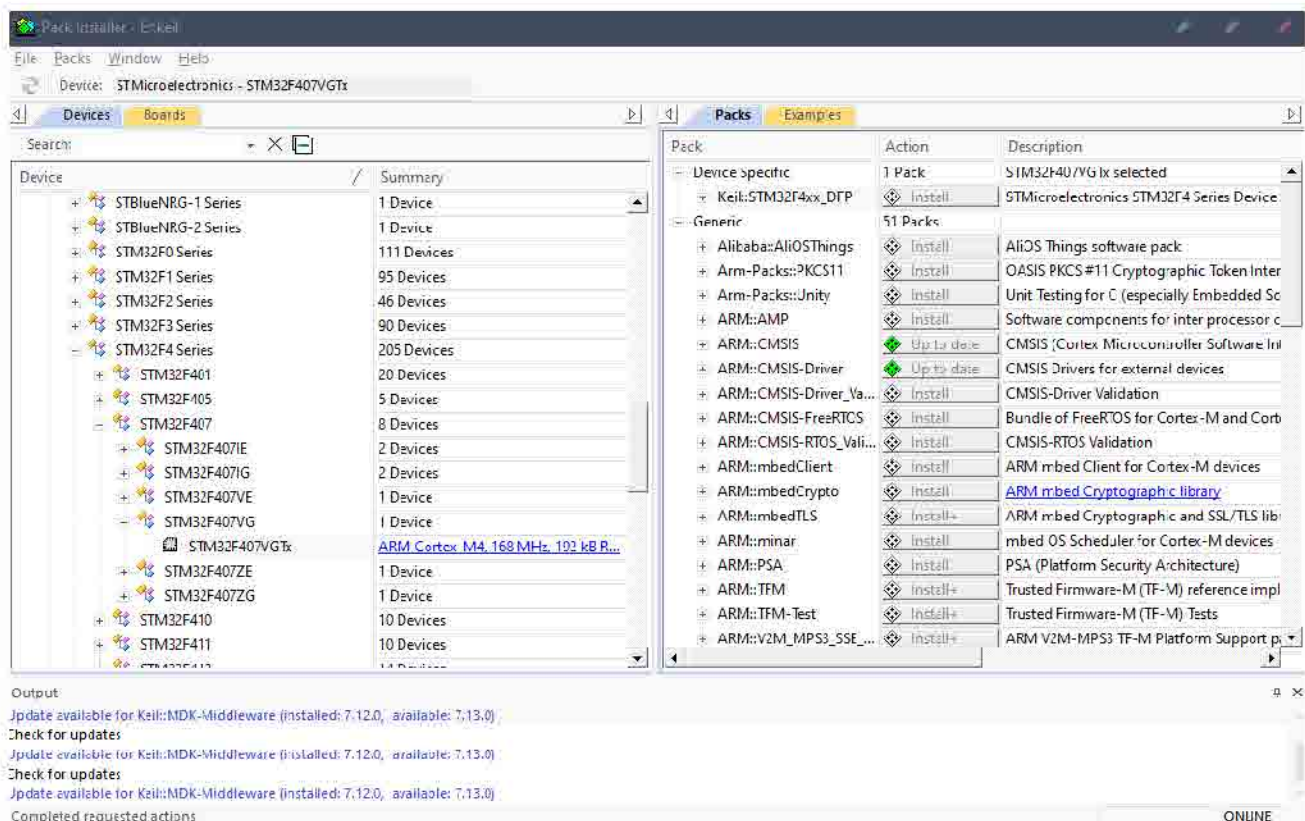


Рисунок 1.25. Меню налаштувань Keil uVision

Разом з цим треба створити виконавчий і заголовний файл, але саме:

- LIS3DSH.h – заголовний файл, котрий містить макропідстановки, необхідні задля отримання даних з акселерометру, але разом з цим його налаштування;

- LIS3DSH.c – виконавчий файл, котрий містить набори функцій задля отримання даних з акселерометру, але разом з цим задля вмикання світлодіодів.

Підключення змінних структури інтерфейсу S-P-I здійснюється поза помічною командою extern.

Виконавчий файл LIS3DSH.c містить наступні функції:

- Accel_GetXYZ – зчитує дані з акселерометру (всі три осі);
- AcclInit – первинна ініціалізація акселерометру;
- Accel_Ini – ініціалізація датчика;
- Accel_IO_Write – записи даних у регістри датчика;
- Accel_IO_Read – отримання даних з регістрів датчика;
- Accel_ReadAcc – дані з акселерометра;
- SPIx_WriteRead – прийом і передача інформації по інтерфейсу S-P-I.

Функцію, що отримує дані з акселерометра LIS3DSH, потрібно помістити в нескінченний цикл в файлі main.c. Всі файли треба зв'язати поміж собою необхідними бібліотеками поза поміччю директиви #include.

Задля прошивання програми опрацювання даних з датчика ускорення до МК платформи STM-32-F-407-VG треба приєднати її до комп'ютера поза поміччю роз'єму mini-USB і відповідного кабелю. Опісля цього можливо запускати проект в Keil uVision і завантажувати його в плату STM-32-F-407-VG. Код програми настроювання МК STM-32 задля роботи з датчиком ускорення представлений в Додатку АЛЕ.

При позиціонуванні платформи STM-32-F-407-VG з датчиком LIS3DSH в повністю горизонтальному положенні жоден з світлодіодів положення у середині платформи, що відповідають поза наявність вібрацій, не горить (рис. 1.26).

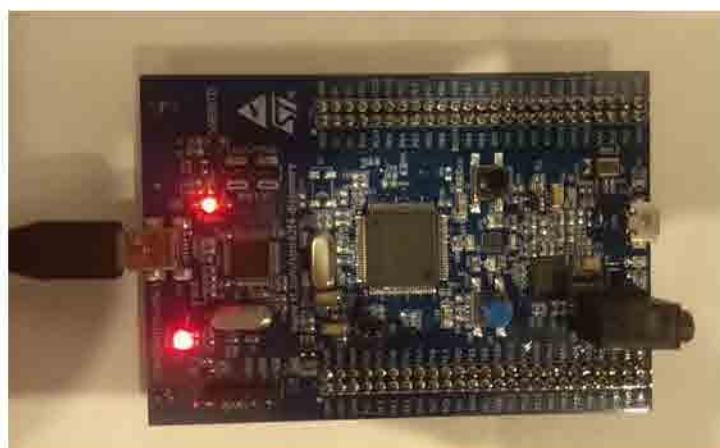


Рисунок 1.26. Позиціонування платформи в горизонтальному положенні

При позиціонуванні платформи STM-32-F-407-VG з деяким відхиленням з горизонтального положення запалюється один із світлодіодів у середині платформи, що вказує, в котрий бік нахилена плата (рис.1.27).

1.11 Розробка програмного забезпечення задля зчитування та відображення даних з акселерометра

Коли плата STM-32-F-407-VG підключена до комп'ютеру, дані постійно надходять крізь COM-порт, через це треба реалізувати алгоритм, котрий буде зчитувати дані з акселерометра і відтворювати в виді графіків на екрані

комп'ютера задля забезпечення можливості їх подальшого аналізу та збереження з метою спостерегання поза сейсмічною активністю. Задля реалізації таких дій будуть застосовуватись функції ОС Windows і зв'язок програмування WinAPI.

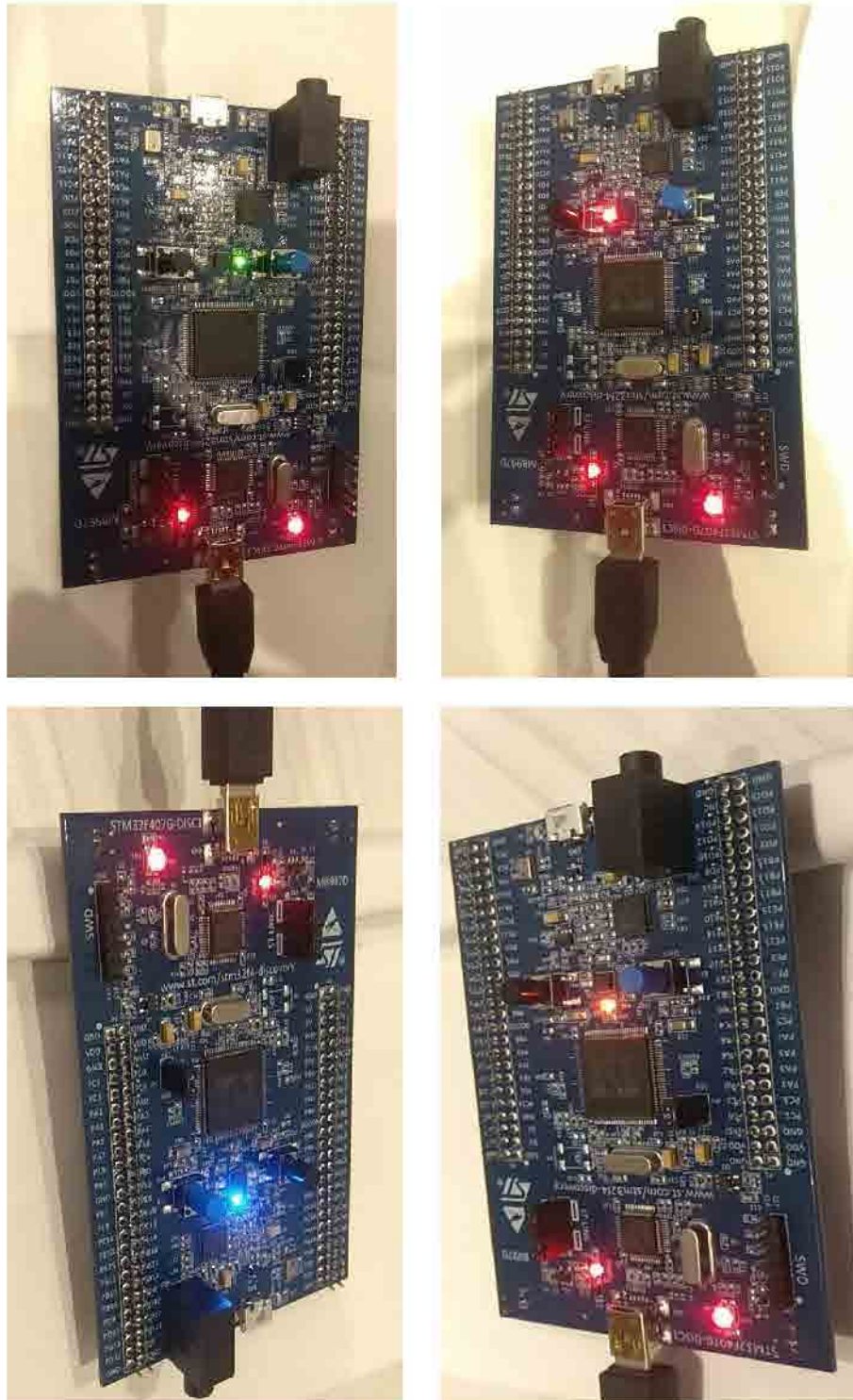


Рисунок 1.27. Позиціонування платформи з відхиленням з горизонтального положення

Алгоритм опрацювання даних з датчика, що надходять з МК платформи STM-32-F-407-VG, представлений на рис.1.28.

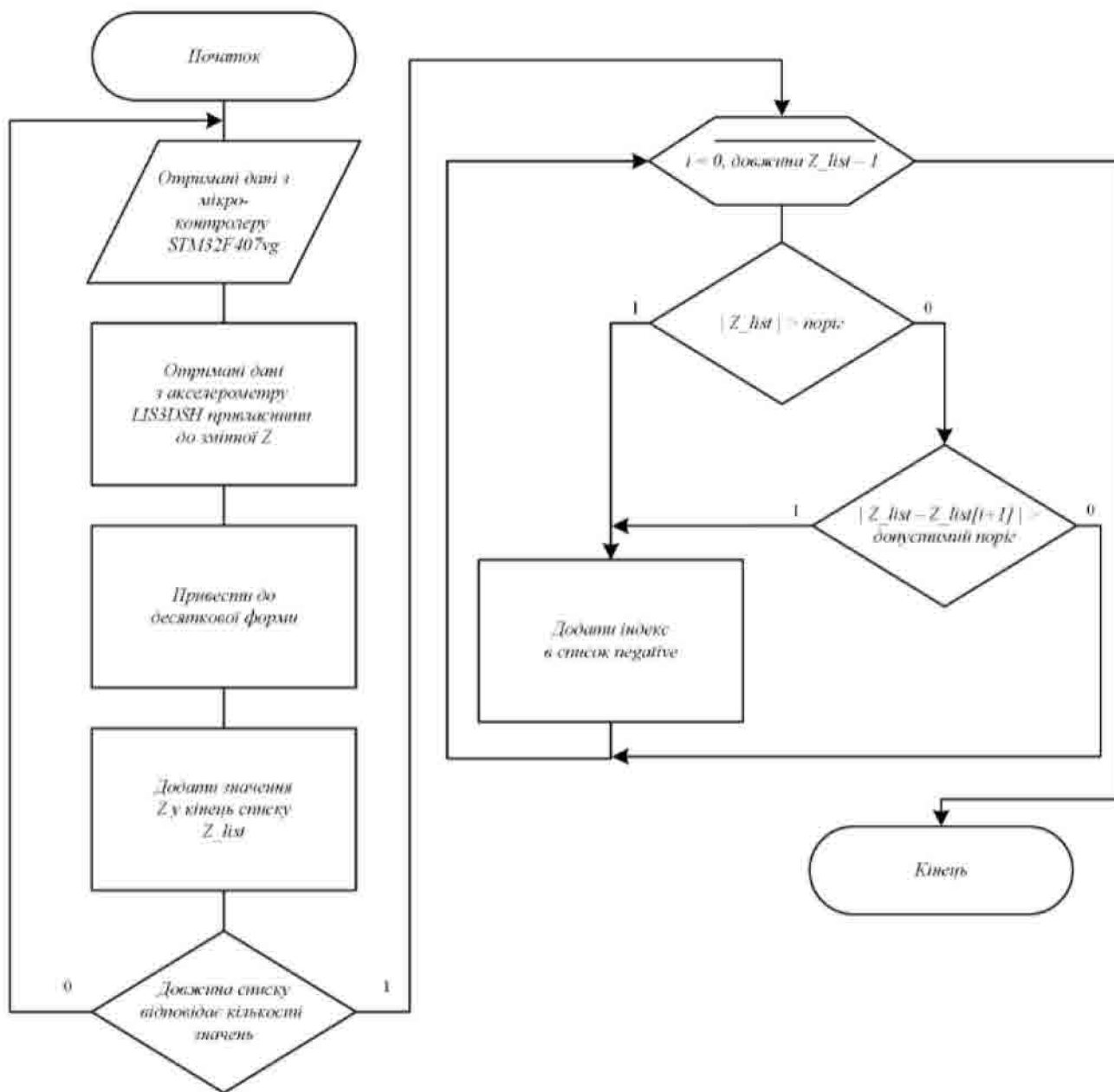


Рисунок 1.28. Блок-схема алгоритму опрацювання даних акселерометра з МК платформи STM-32-F-407-VG

Розроблюваний програмний застосунок задля зчитування та відображення даних з акселерометра буде мати візуальний зв'язок, де можливо обирати порт підключення до платформи МК і його параметри швидкості. Опісля натискання кнопки Start обраний порт відкривається і всі дані з акселерометру починають зчитуватися програмою та відображуватися на екрані в виді графіків. Якщо натиснути на кнопку Stop, зчитування даних зупиняється, але графік з вимірними до цього даними залишається на екрані. Зчитані дані мають існувати цілими числами. Акселерометр передає позначення задля усіх трьох осей (x,y,z), через це на екрані відображується 3 графіка різних кольорів, кожен з яких

відповідає своїй осі. Код програми, створеної мовою C++, представлений в Додатку Б. В програмі реалізовані такі функції:

- MyRegisterClass() – реєструє клас вікна;
- InitInstance (HINSTANCE, int) – зберігає обробку екземпляру та створює головне вікно;
- WndProc (HWND, UINT, WPARAM, LPARAM) – оброблює в головному вікні повідомлення;
- IsNumber (wchar_t *s, unsigned int n) – перевіряє рядок на ціле число;
- WM_COMMAND – оброблює головне меню;
- WM_PAINT – малює головне вікно;
- WM_DESTROY – відправляє повідомлення про вихід.

Якщо МК платформи STM-32-F-407-VG знаходиться в спокійному положенні, всі отримані позначення будуть близькі до ускорення вільного падіння та графіки на екрані будуть представляти собою майже прямі лінії (рис. 1.29), що буде позначати відсутність сейсмічної активності.



Рисунок 1.29. Зв'язок застосунку при відсутності сейсмічної активності

Якщо на акселерометр LIS3DSH платформи STM-32-F-407-VG впливають вібрації, графіки на екрані будуть відображувати зміни в положенні відносно горизонтального в різних напрямках задля осей x,y,z (рис. 1.30), що буде позначати присутність сейсмічної активності. Відхилення можливо побачити одразу ж на екрані в інтерфейсі програми, адже зчитування даних з акселерометру і будівання графіків здійснюється в реальному часі.



Рисунок 1.30. Зв'язок застосунку при наявності сейсмічної активності

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

У даному дипломному проекті розроблено систему спостереження поза сейсмічною активністю на основі платформи STM-32.

Ефективність кожного програмного продукту визначається його якістю і ефективністю процесу розроблення. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції застосування ресурсів; виконання вимог до програмного забезпечення.

Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнуою спроможністю каналів передачі даних. Оцінка якості програмного продукту включає визначення трудомісткості та вартості його створення.

Проведемо розрахунки визначення трудомісткості розроблення даного програмного продукту.

2.2 Розрахунок ціни програмного продукту нормативним методом

2.2.1 Визначення трудомісткості розроблення програмного забезпечення

Тривалість розроблення програмного продукту залежить з його обсягу, трудомісткості розроблення, кваліфікації виконавців, але разом з цим планових термінів, визначених умовами ринку.

Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, в тисячах умовних машинних команд програми аналога.

В таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, в більшому чи меншому ступені, виконує розроблений програмний продукт.

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

Таблиця 2.1

Найменування ПП	Обсяг функції ПП – V_o , усл. машинних командах.
1. ПП СУБД	2500 – 9800
2. Комплексні системи ведення БД	950 – 7430
3. ПП введення інформації	1060 – 5750
4. ПП оптимізації розрахунків	1300 – 4200
5. ПП автоматизації засобів по каталогу	680 – 7000
6. ПП автоматизованих розрахунків	1300 – 8600
7. ПП загальної математики та ПП імітаційного моделювання	7800 – 8800
8. ПП організації обчислювального процесу	13000 – 10200

Задля нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить V_o у умовних машинних командах, трудомісткості визначати на основі табл.2.2

Таблиця 2.2

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283
5.00	306
6.00	330
7.00	357
8.00	385
9.00	414
10.00	445
12.00	510
14.00	580
16.00	654
18.00	731
20.00	812

На підставі отриманого позначення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується

поправочним коефіцієнтом враховуючої умови розроблювання ПП, тобто у умовах комп'ютера, $K_k=0,7\div 0,8$): $T_{ар} = 229 \times 0,7 = 160,3$ (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розроблювання окремо на підставі трудомісткості аналога з урахуванням складності розроблювання, ступеня новизни та ступеня застосування у розробці стандартних модулів на підставі формул:

$$T_{ТЗ} = T^a p \times L_1 \times K_H \quad (2.1)$$

$$T_{ТП} = T^a p \times L_2 \times K_H \quad (2.2)$$

$$T_{РП} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Задля розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розроблювання (див. табл. 2.2);

K_H – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.3);

K_T – поправочний коефіцієнт, що враховує ступінь застосування у розробці типових програм (див. табл. 2.4).

Таблиця 2.3. Позначення питомих коефіцієнтів трудомісткості стадії у загальній трудомісткості розроблювання ПП

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Задля нашого варіанта виділено сірим кольором.

Таблиця 2.4. Позначення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Позначення K_H
А	Принципово нові ПЗ	1,75 – 1,2
Б	ПЗ – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПЗ маючий аналог	0,7

Задля нашого варіанта виділено сірим кольором. Через це що розробка системи є ПЗ, що містить аналоги програмних продуктів, то код ступеня новизни задля мого ПЗ – У, але позначення коефіцієнта $K_H=0,7$. По таблиці 2.3, знаючи код ступеня новизни, тепер можливо визначити позначення питомих коефіцієнтів трудомісткості:

$$L_1=0,12;$$

$$L_2=0,11;$$

$$L_3=0,61;$$

Таблиця 2.5. Позначення коефіцієнта ступеня застосування у розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПЗ типовими програмами, %	Позначення K_T
60 та вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Задля нашого варіанта виділено сірим кольором. В розробленому програмному продукті використовується з 40 до 60 відсотків існуючих функцій, це значить, що $K_T=0,7$.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ}=I*L_1*K_H=160,3 *0,12*0,8= 15,38 \text{ (люд/годин)}$$

Трудомісткість розроблювання технічного проекту

$$T_{ТП}=I*L_2*K_H=160,3 *0,11*0,8 = 14,11 \text{ (люд/годин)}$$

Трудомісткість розроблювання робочого проекту

$$T_{РП}=I*L_3*K_H*K_T=160,3 *0,61*0,8*0,8= 62,58 \text{ (люд/годин)}$$

Задля подальших розрахунків визначили кількість папера, витраченого на кожен етап:

- технічне завдання $N_{ТЗ}=3$ (стр),
- розробка ТП $N_{ТП}=15$ (стр),
- розробка робочого проекту $N_{РП}=20$ (стр),
- пояснювальна записка відповідно $N_{ПЗ}=30$ (стр)

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин		
	1	2	3
1.ТЗ	$T_{РТЗ}=15,38$	$T_{КК}=0,7 \cdot N_{ТЗ}= 0,7 \cdot 3=2,1$	$T_{НК}=0,15 \cdot N_{ТЗ}=0,15 \cdot 3=0,45$
2.Розробка ТП	$T_{РТП}=14,11$	$T_{КК}=0,7 \cdot N_{ТП}=0,7 \cdot 15=10,5$	$T_{НК}=0,15 \cdot N_{ТП}=0,15 \cdot 15=2,25$
3.Розробка РП	$T_{РРП}= 62,58$	$T_{КК}=0,7 \cdot N_{РП}=0,7 \cdot 20=14,0$	$T_{НК}=0,15 \cdot N_{РП}=0,15 \cdot 20=3,0$
4.Розробка ПЗ	$T_{ПЗ}=1,5 \cdot N_{ПЗ}=1,5 \cdot 30 =45$	$T_{КК}=0,7 \cdot N_{ТЗ}=0,7 \cdot 30=21,0$	$T_{НК}=0,15 \cdot N_{ПЗ}=0,15 \cdot 30 =4,5$
Усього, у т.ч.:	194,87		
- на розробку	$\Sigma T_p=137,07$		
- контроль керівника		$\Sigma T_{КК}=47,6$	
- нормоконтроль			$\Sigma T_{НК}=10,2$

3 РОЗДІЛ ОХОРОНИ ПРАЦІ І ТЕХНІКИ БЕЗПЕКИ

Роботодавець чи уповноважені ним органи зобов'язані дбати про умови праці працівників, полегшувати їх, оздоровляти навколишнє середовище, дбати про виконання правил безпеки та інструкцій по техніці безпеки.

Координує всю цю діяльність служба охорони праці, яка у залежності з чисельності працюючих спроможне функціонувати як самостійний структурний підрозділ (число працюючих 50 та більше), чи в виді групи спеціалістів чи одного спеціаліста, в через це числі поза сумісництвом (число працюючих 20 та менше). Задачі службі охорони праці і її функції викладені у Типовому положенні про службу охорони праці», яке затверджено наказом Комітетом Держнаглядохоронпраці (ДНАОП 0.00-4.21-93).

Працівники разом з цим повинні відповідально ставитись до охорони праці, знати і виконувати вимоги, визначені нормативною документацією. У сучасних умовах кожному працівнику треба постійно підтримувати високий фізичний, психологічний і фаховий ступінь, запобігати виникненню випадків травматизму і профзахворювань. Безпечні умови праці на підприємстві досягаються поза рахунок забезпечення безпеки виробничих процесів, котрі обґрунтовані та прийняті у технологічній частині дипломного проекту.

3.1 Аналіз небезпечних і шкідливих чинників, що впливають на працівника

Задля установлення можливого впливу на здоров'я користувачів ВДТ виробничих чинників містить позначення ряд якісних характеристик робочого середовища. Фізико-хімічні показники включають інформацію про вміст в повітрі іонів і різноманітних забруднювачів, але разом з цим деякі інші якісні характеристики середовища.

3.2 Розробка заходів з охорони праці

3.2.1 Виробничі приміщення

Будівлі і приміщення, де розміщені робочі місця програмістів повинні відповідати вимогам СНіП 2.09.02-85 «Производственные здания» і ДСанПіН

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

3.3.2.007 – 98 «Державні санітарні правила та норми роботи з ВДТ ЕОМ» Вони мають існувати не нижче другого ступеня вогнестійкості. Задля усіх приміщень повинно існувати визначено клас зони згідно з НПАОП 40.1-1.01-97. Відповідне позначення повинно існувати нанесено на вхідних дверях кожного приміщення.

Не дозволяється розташування приміщень з робочими місцями операторів ПК в підвалах та цокольних поверхах. Площа приміщення із розрахунку на одне робоче місце містить існувати не менше 6,0 кв.м, але об'єм – не менше 20,0 куб.м.

Задля внутрішнього оздоблення приміщень з ПК слід використовувати дифузно-відбивні матеріали з коефіцієнтом відбитті задля стелі 0,7 – 0,8, задля стін 0,5 – 0,6. Покриття підлоги повинне існувати матовим, поверхня рівною, не слизькою, з антистатичними властивостями.

Віконні прорізи приміщень задля роботи з ПК мають існувати обладнані регульованими пристроями (жалюзі, завіски, зовнішні козирки).

Забороняється задля оздоблення інтер'єру приміщень з ПК застосовувати полімерні матеріали, що виділяють в повітря шкідливі хімічні речовини. Приміщення можуть обладнуватись шафами задля зберігання документів, полицями, стелажми. В приміщеннях слід щоденно робити вологе прибирання. Вони мають існувати оснащені аптечками першої медичної допомоги. При приміщеннях з ВДТ мають існувати обладнані побутові приміщення задля відпочинку під період роботи, кімната психологічного розвантаження, де слід передбачити встановлення приладів задля приготування й роздачі тонізуючих напоїв, але разом з цим місця задля занять фізичною культурою (СНиП 2.09.04 – 87).

3.2.2 Мікроклімат робочої зони працівників, вентиляція

Висока температура повітря негативно позначається на функціональному стані людини. Хоч генерація теплоти дисплеєм досягає критичного ступеня тільки в саму теплу пору року, треба створювати комфортні теплові умови постійно. Оптимальні і допустимі мікрокліматичні параметри в приміщеннях повинні враховувати специфіку технологічного процесу при використанні

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

комп'ютерів. Згідно з діючими в нашій країні нормативними документами (ДСанПіН 3.3.2-007-98 в холодні періоди року температура повітря, швидкість його руху і відносна вологість повітря повинні відповідно складати: 22-24⁰С; 0,1 м/с; 40-60%. Температура повітря спроможне коливатись в межах з 21 до 25⁰С при збереженні інших параметрів мікроклімату.

У теплі періоди року температура повітря, його рухливість і відносна вологість повинні відповідно становити: 23-25⁰С; 0,1-0,2 м/с; 40-60 %.

Оптимальним рівнем аероіонізації в зоні дихання користувача вважається вміст легких аерофонів обох знаків з 150 до 5000 в 1 см³ повітря.

Нормалізуючий вплив на склад повітря робочої зони справляють примусова вентиляція, захисні екрани (оснащені заземленням) і застосування іонізаторів.

3.2.3 Освітлення робочого місця, шум, вібрація

Освітлення в приміщеннях з ВДТ містить існувати змішаним – природним і штучним. Природне освітлення повинно здійснюватись в виді бічного освітлення і відповідати нормам ДБН У.2.5-28-2006 «Природне та штучне освітлення».

При природному освітленні слід передбачити наявність сонцезахисних засобів, що знижують перепади яскравостей поміж природним світлом і свіченням екрана ВДТ. З цією метою можливо використовувати плівки з металізованим покриттям чи жалюзі з вертикальними ламелями, що регулюються.

Штучне освітлення в приміщеннях з ВДТ треба здійснювати в виді комбінованої системи освітлення з використанням люмінесцентних джерел світла в світильниках загального освітлення. На робочих місцях містить існувати забезпечена рівномірна освітленість поза поміччю переважно відбитого чи розсіяного світлорозподілу. Світлових відблисків з клавіатури, екрана і з інших частин ВДТ в напрямку очей користувача не повинно існувати. Норма освітленості на робочих місцях складає 300-500лк.

Деякі ВДТ є потенційними джерелами цілого ряду звуків, що містять як

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

коливання, котрі можливо почути, так та коливання ультразвукового діапазону. Цей шум справляє негативний вплив на стан користувача, особливо при тривалому впливі.. В користувача, діяльність якого пов'язана з переробкою інформації це виражається в знижені розумової працездатності, зростає кількість помилок, розвиток зорового втомлення, зміні відчуття кольорів, появи головного болю, послаблення уваги. Нормованим параметром шуму на робочих місцях є ступінь 50 дБ. Основними заходами боротьби з шумом є усунення чи ослаблення причин шуму у самому його джерелі в процесі проектування, застосування засобів звукопоглинання, раціональне планування виробничих приміщень.

3.2.4 Електробезпека

- Причинами ураження працівника електрострумом можуть існувати:
- Випадковий дотик до струмоведучих частин, в результаті ведення робіт поблизу чи на цих частинах;
- Випадковий дотик до струмоведучих частин, в результаті ведення робіт поблизу чи на цих частинах;
- Несправність захисних засобів, якими потерпілий доторкався до струмоведучих частин;

Помилкове прийняття устаткування, що перебуває під Електробезпека.

Позначення сили струму, що проходить крізь організм людини, залежить з напруги, під якою перебуває людина й з опору ділянки тіла, до якого прикладена ця напруга. Джерелом живлячої напруги є мережа змінного струму з напругою 229В, на яку поширюється ГОСТ 25861-83.

Основними причинами електротравматизму є:

- напругою, як відключеного;
- Несподіване виникнення напруги крізь ушкодження ізоляції там, де у нормальних умовах його існувати не повинно;
- Контакт струмопровідного устаткування із проводом, що перебуває під напругою.

Задля попередження поразок електричним струмом треба чітко й в

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

повному обсязі виконувати правила провадження робіт та правил технічної експлуатації. Треба виключити спроможність доступу оператора до частин устаткування, що працює під небезпечною напругою, до неізольованим частинам, призначеним задля роботи при малій напрузі й не підключеним до захисного заземлення, але разом з цим підводити електроживлення до ПЕОМ з розетки поза поміччю спеціальної вилки із заземлюючим контактом.

3.2.5 Організація робочого місця користувача ПК

Обладнання та організація робочого місця з ВДТ мають забезпечувати відповідність конструкцій усіх елементів робочого місця і їх взаємного розташування, ергономічним вимогам, з урахуванням характеру та особливостей трудової діяльності (ДСанПіН 3.3.2.-007-98).

Конструкція робочого місця й взаємне розташування усіх його елементів (сидіння, органи керування, засобу відображення інформації) відповідають антропометричним, фізіологічним та психологічним вимогам, але разом з цим характеру роботи. Конструкція робочих меблів дає спроможність забезпечувати спроможність індивідуального регулювання їх відповідно до потреб працівника задля підтримки зручної пози. Робочий стіл повинен існувати пофарбований матовою фарбою. Дисплей розташований так, що його верхній край перебуває на рівні очей, на відстані близько 70 см, що укладається у припустимі рамки з 60 до 90 см. Частота мерехтіння екрана дорівнює 100 Гц, що відповідає умові більше 70 Гц.

Задля зниження нервово-емоційного напруження, стомлювання, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільно впроваджувати виконання комплексу вправ, котрі наведені в Державних санітарних правилах та нормах роботи з візуальними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

ВИСНОВКИ

Кінцевим результатом дипломної роботи є розроблена модель системи спостереження поза сейсмічною активністю і її програмно-апаратне забезпечення на основі платформи STM-32.

Визначено, що застосування акселерометра задля реалізації спостереження поза сейсмічною активністю є ефективним рішенням, адже ймовірність його виходу з ладу дуже низка, його легко програмувати і можливо отримати наочний результат.

Платформа STM-32-F-407-VG компанії STMicroelectronics містить вбудований акселерометр LIS3DSH і потребує мінімум електроенергії при високій функціональності.

В якості середовища розроблювання програмного забезпечення задля спостереження поза сейсмічною активністю використовувалось Keil uVision з мовою програмування C, що дозволило ефективно програмувати мікроконтролер сімейства STM-32.

Розроблена структура апаратних і програмних засобів на основі платформи STM-32-F-407-VG повністю задовольняє поставлений в технічному завданні меті.

Розроблене програмне забезпечення надає отримувати дані з акселерометра в реальному часі і відображувати їх на екрані в виді різнокольорових графіків, кожен з яких відповідає своїй осі (x,y,z), що відповідає сейсмічній активності.

Розроблена система чітко і швидко реагує на зміни сейсмічної активності, але разом з цим надає спроможність отримувати точні дані завдяки характеристикам акселерометра і відстеження даних по трьом осям.

Розроблене програмно-апаратне забезпечення у подальшому спроможне існувати вдосконалене і розширене з метою опрацювання сейсмічних даних, котрі несуть у собі інформацію про рух людей, їх швидкість, вагу. Така інформація спроможне існувати корисна задля охоронних компаній і державних органів. Окрім того, розроблене програмно-апаратне забезпечення буде мати спроможність вбудовування у систему Smart City.

					КС 57. 22 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Квашнін В.О. Методологія програмування мікроконтролерів Stm32F4Discovery і практичного їх застосування для вирішення наукових та інженерних задач – Краматорськ: ДДМА, 2016. – 209 с.
2. Оникієнко Ю.О. Основи проектування систем Інтернету речей. Периферія мікроконтролерів STM32. Навч. посіб. – Київ: КПІ ім. І. Сікорського, 2022
3. Квашнін, В.О. Програмування та застосування мікроконтролерів STM32F4Discovery / Краматорськ: ЦТPI «Друкарський дім», 2017. – 143 с.
4. Войтенко В.П. Мікроконтролери STM32F4. Метод. вказ. до виконання лабор. робіт – Чернігів: ЧНТУ, 2017. – 81 с.
5. GNU Tools for ARM Embedded Processors [Електронний ресурс]: <https://launchpad.net/gcc-arm-embedded/+download>
6. STM32F4DISCOVERY STM32F4 high-performance discovery board [Електронний ресурс]: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00039084.pdf.
7. Keil uVision середовище розробки. [Електронний ресурс]: <https://www2.keil.com/mdk5/uvision/>
8. STM32Cube initialization code generator [Електронний ресурс]: <http://www.st.com/en/development-tools/stm32cubemx.html>
9. Clock configuration tool for STM32F40x/41x [Електронний ресурс]: <http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1533/PF257927>
10. ST Visual Programmer for programming STM32 [Електронний ресурс] <http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1533/PF210568>
11. Послідовна шина I2C [Електронний ресурс]: <https://uk.wikipedia.org/wiki/I%C2%B2C>
12. MEMS-датчики. [Електронний ресурс]: <https://ptelectronics.ru/stati/mems-tehnologii-stmicroelectronics/>
13. LIS3DSH акселерометр [Електронний ресурс]: <https://www.st.com/en/mems-and-sensors/lis3dsh.html>.

Лістинг програми налаштування мікроконтролера STM32

Код файлу *main.h*:

```

#ifndef MAIN_H_
#define MAIN_H_
#include "stm32f4xx_hal.h"
#include "lis3dsh.h"
//_____
#define LD_PORT GPIOD
#define LD3 GPIO_PIN_13 //ORANGE
#define LD4 GPIO_PIN_12 //GREEN
#define LD5 GPIO_PIN_14 //RED
#define LD6 GPIO_PIN_15 //BLUE
#define LD3_ON HAL_GPIO_WritePin(LD_PORT, LD3, GPIO_PIN_SET) //ORANGE
#define LD4_ON HAL_GPIO_WritePin(LD_PORT, LD4, GPIO_PIN_SET) //GREEN
#define LD5_ON HAL_GPIO_WritePin(LD_PORT, LD5, GPIO_PIN_SET) //RED
#define LD6_ON HAL_GPIO_WritePin(LD_PORT, LD6, GPIO_PIN_SET) //BLUE
#define LD3_OFF HAL_GPIO_WritePin(LD_PORT, LD3, GPIO_PIN_RESET) //ORANGE
#define LD4_OFF HAL_GPIO_WritePin(LD_PORT, LD4, GPIO_PIN_RESET) //GREEN
#define LD5_OFF HAL_GPIO_WritePin(LD_PORT, LD5, GPIO_PIN_RESET) //RED
#define LD6_OFF HAL_GPIO_WritePin(LD_PORT, LD6, GPIO_PIN_RESET) //BLUE
#endif /* MAIN_H_ */

```

Код файлу *LIS3DSH.h*:

```

#ifndef LIS3DSH_H_
#define LIS3DSH_H_
#include "stm32f4xx_hal.h"
#include <string.h>
//_____
#define ABS(x) (x < 0) ? (-x) : x
//_____
#define LD_PORT GPIOD
#define LD3 GPIO_PIN_13 //ORANGE
#define LD4 GPIO_PIN_12 //GREEN
#define LD5 GPIO_PIN_14 //RED
#define LD6 GPIO_PIN_15 //BLUE
#define LD3_ON HAL_GPIO_WritePin(LD_PORT, LD3, GPIO_PIN_SET) //ORANGE
#define LD4_ON HAL_GPIO_WritePin(LD_PORT, LD4, GPIO_PIN_SET) //GREEN
#define LD5_ON HAL_GPIO_WritePin(LD_PORT, LD5, GPIO_PIN_SET) //RED
#define LD6_ON HAL_GPIO_WritePin(LD_PORT, LD6, GPIO_PIN_SET) //BLUE
#define LD3_OFF HAL_GPIO_WritePin(LD_PORT, LD3, GPIO_PIN_RESET) //ORANGE
#define LD4_OFF HAL_GPIO_WritePin(LD_PORT, LD4, GPIO_PIN_RESET) //GREEN
#define LD5_OFF HAL_GPIO_WritePin(LD_PORT, LD5, GPIO_PIN_RESET) //RED
#define LD6_OFF HAL_GPIO_WritePin(LD_PORT, LD6, GPIO_PIN_RESET) //BLUE
//_____
#define CS_GPIO_PORT GPIOE
#define CS_PIN GPIO_PIN_3
#define CS_ON HAL_GPIO_WritePin(CS_GPIO_PORT, CS_PIN, GPIO_PIN_RESET)
#define CS_OFF HAL_GPIO_WritePin(CS_GPIO_PORT, CS_PIN, GPIO_PIN_SET)
#define DUMMY_BYTE ((uint8_t)0x00)
//_____
#define LIS3DSH_WHO_AM_I_ADDR 0x0F
//_____
#define READWRITE_CMD ((uint8_t)0x80)
#define MULTIPLEBYTE_CMD ((uint8_t)0x40)
//_____
#define LIS3DSH_DATARATE_POWERDOWN ((uint8_t)0x00) //power down mode
#define LIS3DSH_DATARATE_3_125 ((uint8_t)0x10) //3.125 hz normal mode
#define LIS3DSH_DATARATE_6_25 ((uint8_t)0x20) //6.25 hz normal mode
#define LIS3DSH_DATARATE_12_5 ((uint8_t)0x30) //12.5 hz normal mode
#define LIS3DSH_DATARATE_25 ((uint8_t)0x40) //25 hz normal mode
#define LIS3DSH_DATARATE_50 ((uint8_t)0x50) //50 hz normal mode

```

```

#define LIS3DSH_DATARATE_100 ((uint8_t)0x60) //100 hz normal mode
#define LIS3DSH_DATARATE_400 ((uint8_t)0x70) //400 hz normal mode
#define LIS3DSH_DATARATE_800 ((uint8_t)0x80) //800 hz normal mode
#define LIS3DSH_DATARATE_1600 ((uint8_t)0x90) //1600 hz normal mode
#define LIS3DSH_X_ENABLE ((uint8_t)0x01)
#define LIS3DSH_Y_ENABLE ((uint8_t)0x02)
#define LIS3DSH_Z_ENABLE ((uint8_t)0x04)
#define LIS3DSH_XYZ_ENABLE ((uint8_t)0x07)
#define LIS3DSH_SERIALINTERFACE_4WIRE ((uint8_t)0x00)
#define LIS3DSH_SERIALINTERFACE_3WIRE ((uint8_t)0x01)
#define LIS3DSH_SELFTEST_NORMAL ((uint8_t)0x00)
#define LIS3DSH_SELFTEST_P ((uint8_t)0x02)
#define LIS3DSH_SELFTEST_M ((uint8_t)0x04)
#define LIS3DSH_FULLSCALE_2 ((uint8_t)0x00) //2g
#define LIS3DSH_FULLSCALE_4 ((uint8_t)0x08) //4g
#define LIS3DSH_FULLSCALE_6 ((uint8_t)0x10) //6g
#define LIS3DSH_FULLSCALE_8 ((uint8_t)0x18) //8g
#define LIS3DSH_FULLSCALE_16 ((uint8_t)0x20) //16g
#define LIS3DSH_FULLSCALE_SELECTION ((uint8_t)0x38)
#define LIS3DSH_FILTER_BW_800 ((uint8_t)0x00) //800 Hz
#define LIS3DSH_FILTER_BW_40 ((uint8_t)0x08) //40 Hz
#define LIS3DSH_FILTER_BW_200 ((uint8_t)0x10) //200 Hz
#define LIS3DSH_FILTER_BW_50 ((uint8_t)0x18) //50 Hz
#define LIS3DSH_SENSITIVITY_0_06G 0.06 //0.06 mg/ digit
#define LIS3DSH_SENSITIVITY_0_12G 0.12 //0.12 mg/ digit
#define LIS3DSH_SENSITIVITY_0_18G 0.18 //0.18 mg/ digit
#define LIS3DSH_SENSITIVITY_0_24G 0.24 //0.24 mg/ digit
#define LIS3DSH_SENSITIVITY_0_73G 0.73 //0.73 mg/ digit
#define LIS3DSH_OUT_X_L_ADDR 0x28
#define LIS3DSH_OUT_X_H_ADDR 0x29
#define LIS3DSH_OUT_Y_L_ADDR 0x2A
#define LIS3DSH_OUT_Y_H_ADDR 0x2B
#define LIS3DSH_OUT_Z_L_ADDR 0x2C
#define LIS3DSH_OUT_Z_H_ADDR 0x2D
#define LIS3DSH_CTRL_REG4_ADDR 0x20
#define LIS3DSH_CTRL_REG5_ADDR 0x24
void Accel_Ini(void);
void Accel_ReadAcc(void);
//_____
#endif /* LIS3DSH_H_ */

```

Код файла main.c:

```

/* USER CODE BEGIN 2 */
Accel_Ini();
/* USER CODE END 2 */
/* USER CODE BEGIN WHILE */
while (1)
{
/* USER CODE END WHILE */
/* USER CODE BEGIN 3 */
Accel_ReadAcc();
}
/* USER CODE END 3 */

```

Код файла LIS3DSH.c:

```

#include "lis3dsh.h"
#include "usbd_cdc_if.h"
//_____
extern SPI_HandleTypeDef hspi1;
uint8_t buf2[8]={0};
char str1[30]={0};
static void Error (void)
{
LD5_ON;
}

```

```

//-----
static uint8_t SPIx_WriteRead(uint8_t Byte) // читання і запис даних через SPI
{
    uint8_t receivedbyte = 0;
    if(HAL_SPI_TransmitReceive(&hspi1, (uint8_t*) &Byte, (uint8_t*) &receivedbyte, 1, 0x1000) != HAL_OK)
    {
        Error();
    }
    return receivedbyte;
}
//-----
void Accel_IO_Read(uint8_t *pBuffer, uint8_t ReadAddr, uint16_t NumByteToRead)
{
    if(NumByteToRead > 0x01)
    {
        ReadAddr |= (uint8_t)(READWRITE_CMD | MULTIPLEBYTE_CMD);
    }
    else
    {
        ReadAddr |= (uint8_t)READWRITE_CMD;
    }
    CS_ON;
    SPIx_WriteRead(ReadAddr);
    while(NumByteToRead > 0x00)
    {
        /* генерування тактів імпульсів SPI на акселерометр (ведений пристрій) */
        *pBuffer = SPIx_WriteRead(DUMMY_BYTE);
        NumByteToRead--;
        pBuffer++;
    }
    CS_OFF;
}
//-----
void Accel_IO_Write(uint8_t *pBuffer, uint8_t WriteAddr, uint16_t NumByteToWrite)
//запис даних до регістру
{
    CS_OFF;
    if(NumByteToWrite > 0x01)
    {
        WriteAddr |= (uint8_t)MULTIPLEBYTE_CMD;
    }
    CS_ON;
    SPIx_WriteRead(WriteAddr);
    while(NumByteToWrite >= 0x01)
    {
        SPIx_WriteRead(*pBuffer);
        NumByteToWrite--;
        pBuffer++;
    }
    CS_OFF;
}
//-----
uint8_t Accel_ReadID(void)
{
    uint8_t ctrl = 0;
    Accel_IO_Read(&ctrl, LIS3DSH_WHO_AM_I_ADDR, 1);
    return ctrl;
}
//-----
void AccInit(uint16_t InitStruct) // первинна ініціалізація акселерометру
{
    uint8_t ctrl = 0x00;
    ctrl = (uint8_t) (InitStruct);
    Accel_IO_Write(&ctrl, LIS3DSH_CTRL_REG4_ADDR, 1); // наприкінці передача 1 byte
    ctrl = (uint8_t) (InitStruct >> 8); // зсув на 8 і передача в 5-й регістр
    Accel_IO_Write(&ctrl, LIS3DSH_CTRL_REG5_ADDR, 1);
}

```

```

//-----
void Accel_GetXYZ(int16_t* pData) // зчитування даних з акселерометру
{
    int8_t buffer [6];
    uint8_t ctrl,i = 0x00;
    float sensetivity = LIS3DSH_SENSITIVITY_0_06G; //0.06 mg
    float valueinfloat = 0;
    Accel_IO_Read(&ctrl, LIS3DSH_CTRL_REG5_ADDR,1); //зчитування даних з 5-го реєстру
    //зчитування старших та молодших байтів по кожній осі
    Accel_IO_Read((uint8_t*)&buffer[0], LIS3DSH_OUT_X_L_ADDR, 1);
    Accel_IO_Read((uint8_t*)&buffer[1], LIS3DSH_OUT_X_H_ADDR, 1);
    Accel_IO_Read((uint8_t*)&buffer[2], LIS3DSH_OUT_Y_L_ADDR, 1);
    Accel_IO_Read((uint8_t*)&buffer[3], LIS3DSH_OUT_Y_H_ADDR, 1);
    Accel_IO_Read((uint8_t*)&buffer[4], LIS3DSH_OUT_Z_L_ADDR, 1);
    Accel_IO_Read((uint8_t*)&buffer[5], LIS3DSH_OUT_Z_H_ADDR, 1);
    switch(ctrl&LIS3DSH_FULLSCALE_SELECTION) //перевірка зчитаних бітів з 5-го реєстра
    //потрібно для реєстріє 1,2,5, а для інших робиться скидання маскою
    {
        case LIS3DSH_FULLSCALE_2:
            sensetivity = LIS3DSH_SENSITIVITY_0_06G;
            break;
        case LIS3DSH_FULLSCALE_4:
            sensetivity = LIS3DSH_SENSITIVITY_0_12G;
            break;
        case LIS3DSH_FULLSCALE_6:
            sensetivity = LIS3DSH_SENSITIVITY_0_18G;
            break;
        case LIS3DSH_FULLSCALE_8:
            sensetivity = LIS3DSH_SENSITIVITY_0_24G;
            break;
        case LIS3DSH_FULLSCALE_16:
            sensetivity = LIS3DSH_SENSITIVITY_0_73G;
            break;
        default:
            break;
    }
    for(i=0;i<3;i++)
    {
        valueinfloat = ((buffer[2*i+1] << 8) + buffer [2*i]);
        /* sensetivity;
        pData[i] = (int16_t)valueinfloat;
    }
}
//-----
void Accel_ReadAcc(void)
{
    int16_t buffer[3]={0};
    int16_t xval, yval, zval = 0x000;
    Accel_GetXYZ(buffer);
    xval = buffer[0];
    yval = buffer[1];
    zval = buffer[2];
    buf2[0]=0x11;
    buf2[1]=0x55;
    buf2[2] = (uint8_t) (xval >> 8);
    buf2[3] = (uint8_t) xval;
    buf2[4] = (uint8_t) (yval >> 8);
    buf2[5] = (uint8_t) yval;
    buf2[6] = (uint8_t) (zval >> 8);
    buf2[7] = (uint8_t) zval;
    CDC_Transmit_FS(buf2,8);
    if((ABS(xval))>(ABS(yval)))
    {
        if(xval > 2000)
        {
            LD5_ON;
        }
    }
}

```

```

    }
    else if(xval < -2000)
    {
        LD4_ON;
    }
}
else
{
    if(yval > 2000)
    {
        LD3_ON;
    }
    else if(yval < -2000)
    {
        LD6_ON;
    }
}

HAL_Delay(20); // затримка 20 мс
LD3_OFF;
LD4_OFF;
LD5_OFF;
LD6_OFF;
}
//_____
void Accel_Ini(void)
{
    uint16_t ctrl = 0x0000;
    HAL_Delay(1000);
    if(Accel_ReadID() == 0x3F) LD4_ON;
    else Error();
    ctrl = (uint16_t)(LIS3DSH_DATARATE_100 | LIS3DSH_XYZ_ENABLE);
    //100 разів на секунду зчитування за всіма осями
    ctrl |= (uint16_t)(LIS3DSH_SERIALINTERFACE_4WIRE |
    //ввімкнути шину SPI. 0 позначає чотири лінії, а 1 позначає три лінії
    LIS3DSH_SELFTEST_NORMAL | //самотестування
    LIS3DSH_FULLSCALE_2| //ширина пропускання
    LIS3DSH_FILTER_BW_800);
    Acclnit(ctrl);
    LD6_ON;
}
//_____

```

Лістинг програми для перетворення та відображення даних з акселерометра

```

#include "stdafx.h"
#include "NSPortMonF.h"
#include <stdio.h>
#include <tchar.h>
#define MAX_LOADSTRING 100
// Глобальні змінні:
HINSTANCE hInst; // поточний екземпляр
WCHAR szTitle[MAX_LOADSTRING]; // Текст рядку заголовка
WCHAR szWindowClass[MAX_LOADSTRING]; // ім'я класу головного вікна
HANDLE handle;
LPCTSTR PortName;
DCB dcb;
COMMTIMEOUTS timeouts; //структура для встановлення таймаутів
BOOL succ;
HWND g_ChildWnd = NULL;
HWND hCombo01, hCombo02, hCombo03;
HWND hWnd;
static int t;
int index;
short y1, y2, y3;
unsigned long y1_ul, y2_ul;
float y1_f, y2_f;
short y1_buf, y2_buf, y3_buf;
short yy;
static int x_old = 0, y1_old, y2_old, y3_old = 0;
static float ydiv1; // дільник для діапазону по осі Y
static int yoffset1; // зсув даних по осі Y
static unsigned long cnt = 0;
static unsigned char buf[280] = { 0 };
static unsigned char bufw[280] = { 0 };
HWND hText01, hText02, hText03, hText04, hButton1, hButton2, hEdit01;
HBITMAP hBitmap = NULL;
wchar_t szBuf2[20];
char *szBuf1;
DWORD byteRate;
BOOL fCloseHandle; // прапорець закриття порту
TCHAR iniText[MAX_LOADSTRING]; //Змінна для ini-файлу
// Відправити оголошення функцій, включених в цей модуль коду:
ATOM MyRegisterClass(HINSTANCE hInstance);
BOOL InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK ChildWndProc(HWND, UINT, WPARAM, LPARAM);
INT_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);
void GetDiv(void); // Функція зчитування дільника з переліку
BOOL IsNumber(wchar_t *s, unsigned int n); // Функція перевірки рядка на ціле число
int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
    _In_opt_ HINSTANCE hPrevInstance,
    _In_ LPWSTR lpCmdLine,
    _In_int_ nCmdShow)
{
    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);
    // Ініціалізація дільника
    ydiv1 = 1.0f;
    // Ініціалізація глобальних рядків
    LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadStringW(hInstance, IDC_MY001, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);
    // Виконати ініціалізацію застосунку
    if (!InitInstance (hInstance, nCmdShow))
    {

```

```

    return FALSE;
}
HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC_MY001));
MSG msg;
// Цикл основного повідомлення
while (GetMessage(&msg, nullptr, 0, 0))
{
    if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}
return (int) msg.wParam;
}
// Функція: MyRegisterClass()
//
// Призначення: реєструє клас вікна
//
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEXW wcex;
    wcex.cbSize = sizeof(WNDCLASSEX);
    wcex.style = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc = WndProc;
    wcex.cbClsExtra = 0;
    wcex.cbWndExtra = 0;
    wcex.hInstance = hInstance;
    wcex.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_MY001));
    wcex.hCursor = LoadCursor(nullptr, IDC_ARROW);
    wcex.hbrBackground = (HBRUSH)(COLOR_INACTIVECAPTION);
    wcex.lpszMenuName = MAKEINTRESOURCEW(IDC_MY001);
    wcex.lpszClassName = szWindowClass;
    wcex.hIconSm = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI_SMALL));
    return RegisterClassExW(&wcex);
}
ATOM RegisterWndClass(LPCWSTR lpszClassName, WNDPROC wndProc)
{
    WNDCLASS wc = { 0 };
    wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1);
    wc.hInstance = hInst;
    wc.lpfnWndProc = wndProc;
    wc.hCursor = ::LoadCursorW(NULL, IDC_ARROW);
    wc.lpszClassName = lpszClassName;
    wc.style = CS_DBLCLKS | CS_HREDRAW | CS_VREDRAW;
    return ::RegisterClassW(&wc);
}
// Функція: InitInstance(HINSTANCE, int)
// Призначення: зберігає обробку екземпляру і створює головне вікно
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    hInst = hInstance; // Зберіає дескриптор екземпляру в глобальній змінній
    hWnd = CreateWindowW(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, 1150, 670, nullptr, nullptr, hInstance, nullptr);
    if (!hWnd)
    {
        return FALSE;
    }
    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);
    return TRUE;
}
// Функція: WndProc(HWND, UINT, WPARAM, LPARAM)
// Призначення: опрацьовує повідомлення в головному вікні
// WM_COMMAND – опрацьовувати меню застосунку
// WM_PAINT – намалювати головне вікно
// WM_DESTROY – відправити повідомлення про вихід і повернутися
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

```

```

{
    static LPCWSTR lpszChildClassName = L"ChildWindow";
    static HINSTANCE hInstance;
    static LPCWSTR lpszChildWndText = L"Дочірнє вікно";
    static TCHAR lpszPort[10] = {0};
    static TCHAR* pszCurrentDevice;
    TCHAR szDevices[MAXWORD];
    DWORD dwChars;
    char str[20] = { 0 };
    wchar_t str2[20] = { 0 };
    int i;
    int nn;
    DWORD dwBytesRead;
    RECT rect_user;
    switch (message)
    {
    case WM_CREATE:
        hInstance = ((LPCREATESTRUCT)lParam)->hInstance;
        RegisterWndClass(lpszChildClassName, ChildWndProc);
        g_ChildWnd = CreateWindowW(lpszChildClassName, lpszChildWndText, WS_CHILD |
            WS_BORDER, 60, 32, 1003, 550, hWnd, NULL, hInst, NULL);
        ShowWindow(g_ChildWnd, SW_SHOW);
        hText01 = CreateWindowW(L"static", L"Com Port", WS_CHILD | WS_VISIBLE | SS_LEFT,
            10, 8, 70, 15, hWnd, NULL, NULL, NULL);
        hText02 = CreateWindowW(L"static", L"Speed", WS_CHILD | WS_VISIBLE | SS_LEFT,
            165, 8, 50, 15, hWnd, NULL, NULL, NULL);
        hText03 = CreateWindowW(L"static", L"FScale", WS_CHILD | WS_VISIBLE | SS_LEFT,
            315, 8, 50, 15, hWnd, NULL, NULL, NULL);
        hText04 = CreateWindowW(L"static", L"Offset", WS_CHILD | WS_VISIBLE | SS_LEFT,
            470, 8, 50, 15, hWnd, NULL, NULL, NULL);
        hEdit01 = CreateWindowW(L"edit", L"", WS_CHILD | WS_VISIBLE | WS_GROUP | WS_TABSTOP
            | SS_RIGHT,
            525, 8, 80, 15, hWnd, NULL, NULL, NULL);
        hButton1 = CreateWindowW(L"button", L"Start", WS_VISIBLE | WS_CHILD,
            623, 5, 80, 20,
            hWnd, (HMENU) 1, hInst, NULL);
        hButton2 = CreateWindowW(L"button", L"Stop", WS_VISIBLE | WS_CHILD,
            730, 5, 80, 20,
            hWnd, (HMENU)2, hInst, NULL);
        EnableWindow(hButton2, FALSE);
        // Додати перелік доступних портів
        hCombo01 = CreateWindowW(L"combobox", L"ListCom01",
            WS_CHILD | WS_VISIBLE | CBS_SORT | CBS_DROPDOWNLIST,
            83, 3, 75, 150,
            hWnd, 0, hInst, NULL);
        SendMessage(hCombo01, CB_RESETCONTENT, 0, 0);
        dwChars = QueryDosDevice(NULL, szDevices, MAXWORD);
        if (dwChars)
        {
            int i = 0;
            for (;;)
            {
                pszCurrentDevice = &szDevices[i];
                if (3 < _tcslen(pszCurrentDevice) && _tcslen(pszCurrentDevice) < 6 &&
                    !_tcsnicmp(pszCurrentDevice, TEXT("COM"), 3))
                    //&& _istdigit(pszCurrentDevice[4]))
                {
                    SendMessage(hCombo01, CB_ADDSTRING, 0, (LPARAM)pszCurrentDevice);
                }
                while (szDevices[i] != _T('\0')) i++;
                if (szDevices[++i] == _T('\0')) break;
            }
        }
        if (!SendMessage(hCombo01, CB_GETCOUNT, 0, 0)) break;
        SendMessage(hCombo01, CB_SETCURSEL, 0, 0);
        // Додати перелік швидкостей портів
        hCombo02 = CreateWindowW(L"combobox", L"ListCom02",
            WS_CHILD | WS_VISIBLE | WS_VSCROLL | CBS_DROPDOWNLIST,

```

```

218, 3, 90, 150,
hWnd, 0, hInst, NULL);
SendMessage(hCombo02, CB_RESETCONTENT, 0, 0);
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"1200");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"2400");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"4800");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"9600");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"14400");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"19200");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"28800");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"38400");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"56000");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"57600");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"115200");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"128000");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"230400");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"256000");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"460800");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"512000");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"921600");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"1572864");
SendMessageA(hCombo02, CB_ADDSTRING, 0, (LPARAM)"12582912");
if (!SendMessage(hCombo02, CB_GETCOUNT, 0, 0)) break;
SendMessage(hCombo02, CB_SETCURSEL, 0, 0);
// Додати перелік діапазонів вимірювань
hCombo03 = CreateWindowW(L"combobox", L"ListCom02",
WS_CHILD | WS_VISIBLE | WS_VSCROLL | CBS_DROPDOWNLIST, 370, 3, 90, 150, hWnd, 0, hInst, NULL);
SendMessage(hCombo03, CB_RESETCONTENT, 0, 0);
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 1");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 2");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 5");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 10");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 20");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 50");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 100");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 200");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 500");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 1000");
SendMessageA(hCombo03, CB_ADDSTRING, 0, (LPARAM)"+ 2000");
if (!SendMessage(hCombo03, CB_GETCOUNT, 0, 0)) break;
SendMessage(hCombo03, CB_SETCURSEL, 0, 0);
// Взяти налаштування з файлу ini
GetPrivateProfileString(_T("General"), _T("portname"), _T("COM1"), iniText, MAX_LOADSTRING,
_T("./config.ini"));
//Знайти рядок у переліку hCombo01
index = SendMessage(hCombo01, CB_FINDSTRING, 0, (LPARAM)iniText);
SendMessage(hCombo01, CB_SETCURSEL, index, 0);
//Те саме для hCombo02
GetPrivateProfileString(_T("General"), _T("portspeed"), _T("9600"), iniText, MAX_LOADSTRING,
_T("./config.ini"));
index = SendMessage(hCombo02, CB_FINDSTRING, 0, (LPARAM)iniText);
SendMessage(hCombo02, CB_SETCURSEL, index, 0);
//Те саме для hCombo03
GetPrivateProfileString(_T("General"), _T("fscale"), _T("+ 1"), iniText, MAX_LOADSTRING,
_T("./config.ini"));
index = SendMessage(hCombo03, CB_FINDSTRING, 0, (LPARAM)iniText);
SendMessage(hCombo03, CB_SETCURSEL, index, 0);
// Те саме для hEdit01
GetPrivateProfileString(_T("General"), _T("offset"), _T("1"), iniText, MAX_LOADSTRING,
_T("./config.ini"));
SendMessage(hEdit01, WM_SETTEXT, 0, (LPARAM)iniText);
// Оновити показники шкали
GetDiv();
break;
case WM_TIMER:
if(t==0) t++;
if ((t >= 1)&&(t<=1024))
{

```

```

WriteFile(handle, &bufw, 8, &dwBytesRead, NULL);
Sleep(1);
ReadFile(handle, &buf, 10, &dwBytesRead, NULL);
Sleep(1);
if (dwBytesRead != 0) {t++;}
if (dwBytesRead == 0) {t++; break;}
//y1
//Взяти 1 показник з плаваючою крапкою
y1_f = *(float*)(buf + 2);
//y1_f = -1.0f;
//застосувати дільник/множник (якщо 1, то 250 крапок) і зсунути на yoffset
y1 = (y1_f-yoffset1)*250/ydiv1;
// вивести 0 на середину
y1 = 250 - y1;
rect_user.left = t-2;
if (y1_old < y1)
{
    rect_user.top = y1_old;
    rect_user.bottom = y1;
}
else
{
    rect_user.top = 0;
    rect_user.bottom = y1_old;
}
rect_user.right = t;
InvalidateRect(g_ChildWnd, &rect_user, TRUE);

//y2
// Взяти 2 показник з плаваючою крапкою
y2_f = *(float*)(buf + 6);
// застосувати дільник/множник (якщо 1, то 250) і зсунути на yoffset
y2 = (y2_f - yoffset1) * 250 / ydiv1;
//вивести нуль на середину
y2 = 250 - y2;
if (y2_old < y2)
{
    rect_user.top = y2_old;
    rect_user.bottom = y2;
}
else
{
    rect_user.top = 0;
    rect_user.bottom = y2_old;
}
InvalidateRect(g_ChildWnd, &rect_user, TRUE);

//перемалювати область для виведення тексту з лічильником
rect_user.left = 10;
rect_user.top = 511;
rect_user.right = 400;
rect_user.bottom = 530;
InvalidateRect(g_ChildWnd, &rect_user, TRUE);
if (t == 1000)
{
    t = 1;
    InvalidateRect(g_ChildWnd, NULL, TRUE);
}
}
cnt++;
break;
case WM_COMMAND:
{
    int wmlId = LOWORD(wParam);
    // Визначити вибір в меню:
    switch (wmlId)
    {
        case IDM_ABOUT:

```

```

DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);
break;
    case IDM_HELP:
        DialogBox(hInst, MAKEINTRESOURCE(IDD_DIALOG1), hWnd, About);
        break;
    case 1: //Кнопка Start
        GetWindowTextW(hEdit01, szBuf2, 10);
        //перевірити рядок на ціле число
        if (!IsNumber(szBuf2, wcslen(szBuf2)))
        {
MessageBox(hWnd, TEXT("Зсув по осі Y має бути цілим числом!"), TEXT("Увага!!!"), MB_OK);
            SendMessage(hEdit01, EM_SETSEL, 0, -1);
            SetFocus(hEdit01);
            break;
        }
        yoffset1 = (float) _wtoi(szBuf2);
        //Скинути прапорець закриття порту
        fCloseHandle = FALSE;
        EnableWindow(hButton1, FALSE);
        EnableWindow(hButton2, TRUE);
        szBuf1 = new char[10];
        GetWindowTextA(hCombo01, szBuf1, 7);
        handle = ::CreateFileA(szBuf1, GENERIC_READ | GENERIC_WRITE, 0, 0,
OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);
        GetWindowTextA(hCombo02, szBuf1, 9);
        byteRate = atoi(szBuf1);
        //Оновити показники шкали
        GetDiv();
        if (handle == INVALID_HANDLE_VALUE)
        {
MessageBox(hWnd, TEXT("Не вдалося!!!"), TEXT("Увага!!!"), MB_OK);
            return FALSE;
        }
        dcb.DCBlength = sizeof(dcb);
        succ = GetCommState(handle, &dcb);
        memset(&dcb, 0, sizeof(dcb));
        dcb.DCBlength = sizeof(DCB);
        GetCommState(handle, &dcb);
        dcb.BaudRate = byteRate;
        dcb.ByteSize = 8;
        dcb.Parity = NOPARITY;
        dcb.StopBits = ONESTOPBIT;
        dcb.fAbortOnError = FALSE;
        dcb.fDtrControl = DTR_CONTROL_DISABLE;
        dcb.fRtsControl = RTS_CONTROL_DISABLE;
        dcb.fBinary = TRUE;
        dcb.fParity = FALSE;
        dcb.fInX = dcb.fOutX = FALSE;
        dcb.XonChar = 0;
        dcb.XoffChar = 0;
        dcb.fErrorChar = FALSE;
        dcb.fNull = FALSE;
        dcb.fOutxCtsFlow = FALSE;
        dcb.fOutxDsrFlow = FALSE;
        //dcb.XonLim = 128;
        //dcb.XoffLim = 128;
        SetCommState(handle, &dcb);
        timeouts.ReadIntervalTimeout = 0; //таймаут між двома символами
        timeouts.ReadTotalTimeoutMultiplier = 100; //загальний таймаут операції зчитування
        timeouts.ReadTotalTimeoutConstant = 100; //константа для таймауту операції зчитування
        timeouts.WriteTotalTimeoutMultiplier = 100; // загальний таймаут операції запису
        timeouts.WriteTotalTimeoutConstant = 100; // константа для таймауту операції запису
        //записати структуру таймаутів у порт
        //якщо не вдалося - закрити порт і вивести повідомлення про помилку у рядку стану
        if (!SetCommTimeouts(handle, &timeouts))
        {
            CloseHandle(handle);
MessageBox(hWnd, TEXT("Не вдалося встановити тайм-ауту"), TEXT("Увага!!!"), MB_OK);
        }
    }
}

```

```

        return FALSE;
    }
    // встановити розміри черги прийому та передачі
    SetupComm(handle, 2000, 2000);
    PurgeComm(handle, PURGE_RXCLEAR); //очистити приймальний буфер порту
    // Стерти графік, адже після Stop може залишитись відображення
    InvalidateRect(g_ChildWnd, NULL, TRUE);
    t = 0;
    //ініціалізуємо буфер для запису
    bufw[0] = 1;bufw[1] = 2;bufw[2] = 3;bufw[3] = 4;bufw[4] = 5;bufw[5] = 6;bufw[6] = 7;bufw[7] = 8;
    SetTimer(hWnd, 1, 200, NULL);
    break;
    case 2: //Кнопка Stop
        KillTimer(hWnd, 1);
        if(fCloseHandle==FALSE) CloseHandle(handle);
        //Встановлюємо прапорець закриття файлу
        fCloseHandle = TRUE;
        EnableWindow(hButton1, TRUE);
        EnableWindow(hButton2, FALSE);
        break;
    case IDM_EXIT:
        DestroyWindow(hWnd);
        break;
    default:
        return DefWindowProc(hWnd, message, wParam, lParam);
    }
}
break;
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    SetTextColor(hdc, RGB(255, 255, 0));
    SetBkColor(hdc, RGB(0, 0, 0));
    for (i = 0; i < 21; i++)
    {
        sprintf(str2, TEXT("%02.1f"), ydiv1 - ((float)i*ydiv1/10) + yoffset1);
        TextOut(hdc, 5, i*25+25, (LPCWSTR)str2, wcslen(str2));
    }
    EndPaint(hWnd, &ps);
}
break;
case WM_DESTROY:
    //Занесення налаштувань у ini-файл
    GetWindowTextW(hCombo01, iniText, 7);
    WritePrivateProfileString(_T("General"), _T("portname"), iniText, _T("./config.ini"));
    GetWindowTextW(hCombo02, iniText, 9);
    WritePrivateProfileString(_T("General"), _T("portspeed"), iniText, _T("./config.ini"));
    GetWindowTextW(hCombo03, iniText, 9);
    WritePrivateProfileString(_T("General"), _T("fscale"), iniText, _T("./config.ini"));
    GetWindowTextW(hEdit01, iniText, 9);
    WritePrivateProfileString(_T("General"), _T("offset"), iniText, _T("./config.ini"));
    // Перевірка прапорця закриття файлу, щоб не намагалися його закрити повторно
    if(fCloseHandle == FALSE) CloseHandle(handle);
    PostQuitMessage(0);
    break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}
LRESULT CALLBACK ChildWndProc(HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam)
{
    BITMAP bm;
    HDC hdc, hmdc;
    int i = 0;
    wchar_t str1[50] = { 0 };
    char szImageName[] = "IDB_BITMAP2";

```

```

switch (msg) {
case WM_CREATE:
    hBitmap = (HBITMAP)LoadImageW(hInst, MAKEINTRESOURCE(IDB_BITMAP2),
IMAGE_BITMAP, 0, 0, LR_COPYFROMRESOURCE);
    break;
case WM_ERASEBKGDND:
    RECT rect;
    ::GetClientRect(hWnd, &rect);
    ::FillRect((HDC)wParam, &rect, ::GetSysColorBrush(COLOR_APPWORKSPACE));
    break;
case WM_PAINT:
    {
        PAINTSTRUCT ps;
        HPEN hPen1, hPen2, hPen3;
        hPen1 = CreatePen(PS_SOLID, 2, RGB(255, 255, 0));
        hPen2 = CreatePen(PS_SOLID, 2, RGB(0, 70, 255));
        hPen3 = CreatePen(PS_SOLID, 2, RGB(255, 0, 70));
        hdc = BeginPaint(hWnd, &ps);

        hmdc = CreateCompatibleDC(hdc);
        SelectObject(hmdc, hBitmap);
        GetObject(hBitmap, sizeof(bm), (LPSTR)&bm);
        for (i = 0; i < 40; i++) BitBlt(hdc, i * 25, 0, bm.bmWidth, bm.bmHeight, hmdc, 0, 0, SRCCOPY);
        if (t == 1)
        {
            x_old = 0;
            y1_old = y1;
            y2_old = y2;
            y3_old = y3;
        }
        else if ((t > 1) && (t <= 1000))
        {
            SelectObject(hdc, hPen1);
            MoveToEx(hdc, x_old, y1_old, NULL);
            LineTo(hdc, t - 1, y1);
            SelectObject(hdc, hPen2);
            MoveToEx(hdc, x_old, y2_old, NULL);
            LineTo(hdc, t - 1, y2);
            x_old = t - 1; y1_old = y1; y2_old = y2;
        }
        SetTextColor(hdc, RGB(150, 150, 150));
        SetBkColor(hdc, RGB(0, 0, 0));
        swprintf(str1, TEXT("Count:%07ld"), cnt);
        TextOut(hdc, 10, 512, (LPCWSTR)str1, wcslen(str1));
        SetTextColor(hdc, RGB(255, 255, 0));
        swprintf(str1, TEXT("y1: %06f"), y1_f);
        TextOut(hdc, 120, 512, (LPCWSTR)str1, wcslen(str1));
        SetTextColor(hdc, RGB(0, 70, 255));
        swprintf(str1, TEXT("y2: %08f"), y2_f);
        TextOut(hdc, 230, 512, (LPCWSTR)str1, wcslen(str1));
        SetTextColor(hdc, RGB(255, 0, 70));
        swprintf(str1, TEXT("y3: %06d"), y3_buf);
        TextOut(hdc, 370, 512, (LPCWSTR)str1, wcslen(str1));
        EndPaint(hWnd, &ps);
        DeleteObject(hPen1);
        DeleteObject(hPen2);
        DeleteObject(hPen3);
        DeleteDC(hmdc);
    }
    break;
case WM_DESTROY:
    g_ChildWnd = NULL;
    DeleteObject(hBitmap);
default:
    return ::DefWindowProcW(hWnd, msg, wParam, lParam);
}
return NULL;
}
}

```

```

// Обробник повідомлень для вікна "Про програму"
INT_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
    UNREFERENCED_PARAMETER(lParam);
    switch (message)
    {
    case WM_INITDIALOG:
        return (INT_PTR)TRUE;
    case WM_COMMAND:
        if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
        {
            EndDialog(hDlg, LOWORD(wParam));
            return (INT_PTR)TRUE;
        }
        break;
    }
    return (INT_PTR)FALSE;
}
// Функція зчитування дільника із списку
void GetDiv(void)
{
    switch (SendMessage(hCombo03, CB_GETCURSEL, 0, 0))
    {
    case 0: {ydiv1 = 1.0f;}
            break;
    case 1: {ydiv1 = 2.0f;}
            break;
    case 2: {ydiv1 = 5.0f;}
            break;
    case 3: {ydiv1 = 10.0f;}
            break;
    case 4: {ydiv1 = 20.0f;}
            break;
    case 5: {ydiv1 = 50.0f;}
            break;
    case 6: {ydiv1 = 100.0f;}
            break;
    case 7: {ydiv1 = 200.0f;}
            break;
    case 8: {ydiv1 = 500.0f;}
            break;
    case 9: {ydiv1 = 1000.0f;}
            break;
    case 10: {ydiv1 = 2000.0f;}
            break;
    default:
        break;
    }
    //перемалювання головного вікна для застосування дільника до шкали
    InvalidateRect(hWnd, NULL, TRUE);
}
// Функція перевірки рядку на ціле число
BOOL IsNumber(wchar_t *s, unsigned int n)
{
    for (int i = 0; i < n; i++)
    {
        if (s[i] < '0' || s[i] > '9')
        {
            if (s[i] != '-')
            {
                return FALSE;
            }
        }
    }
    return TRUE;
}

```

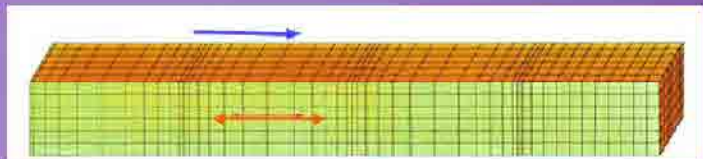
ДОДАТОК В. Слайди мультимедійної презентації

Розробка системи спостереження за сейсмічною активністю на базі платформи STM32

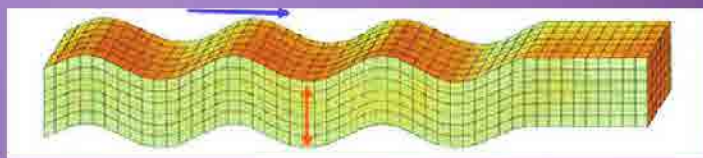
Тігінян Сергій, гр. 4КС-57

ВСП «ОТФК ОНУ»

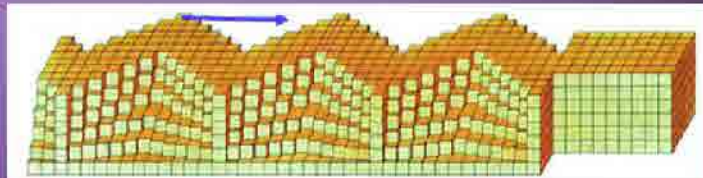
Повздовжня хвиля у
пружному середовищі



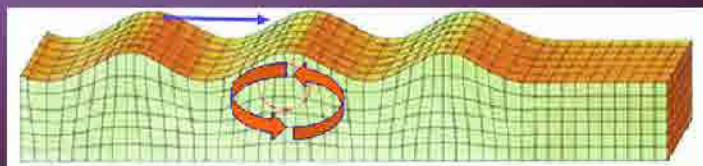
Поперечна хвиля у
пружному середовищі



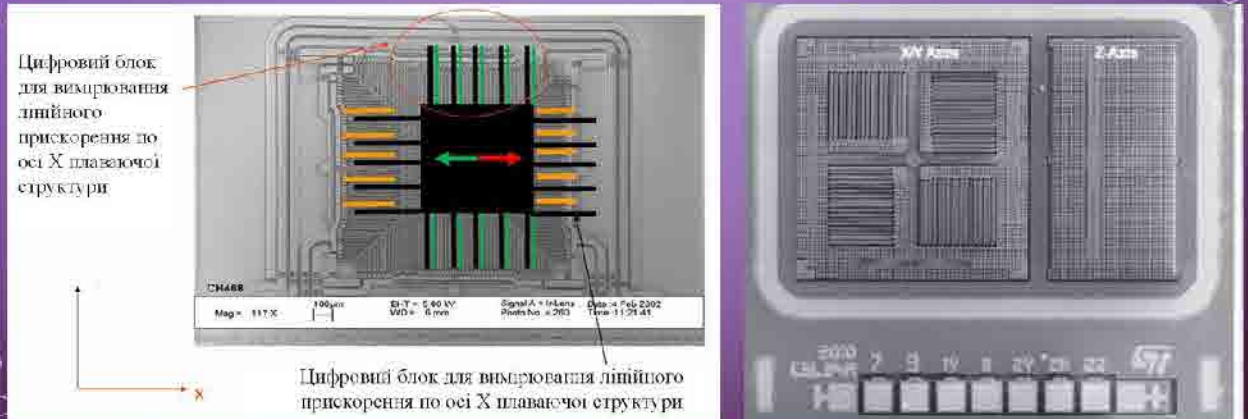
Хвиля Лява



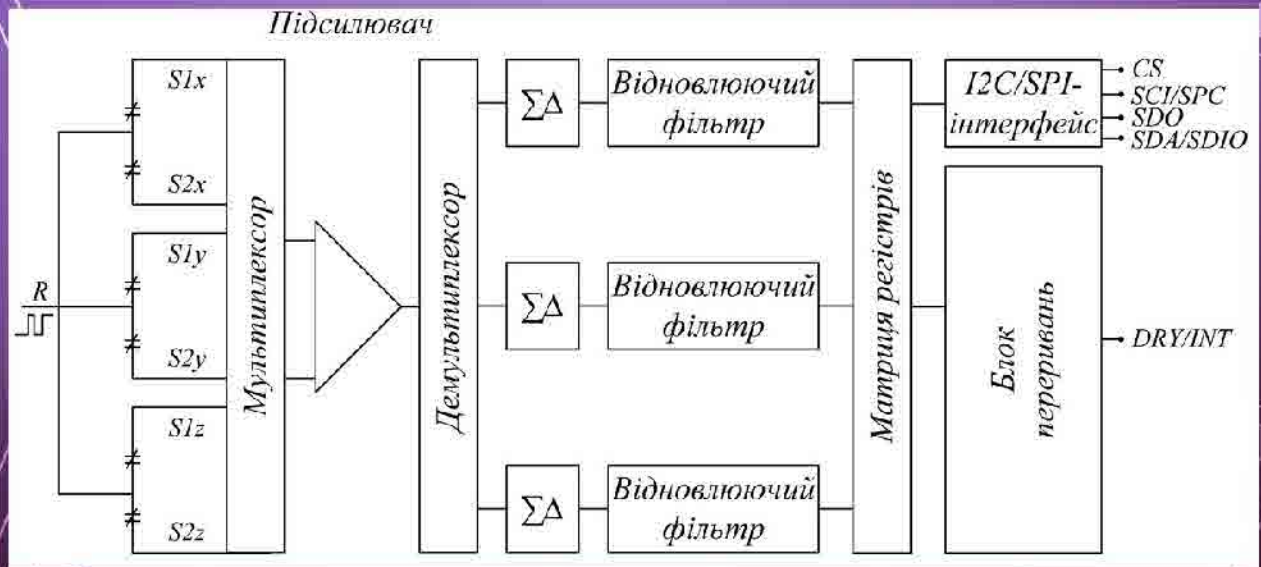
Хвиля Релея



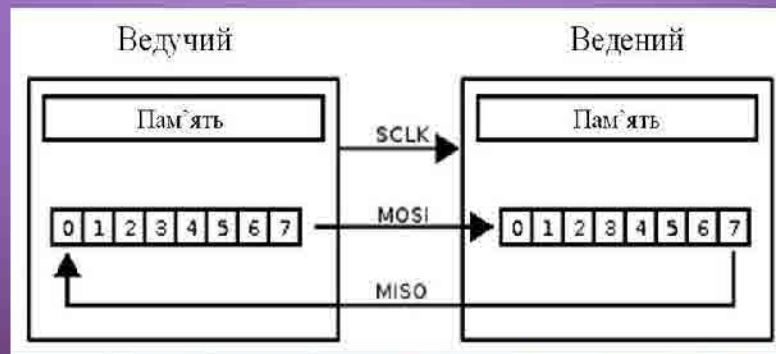
Структура трьохосьового MEMS-акселерометра



Функціональна схема акселерометра для платформи STM32F407VG



Типова структура зв'язків і ліній інтерфейсу SPI



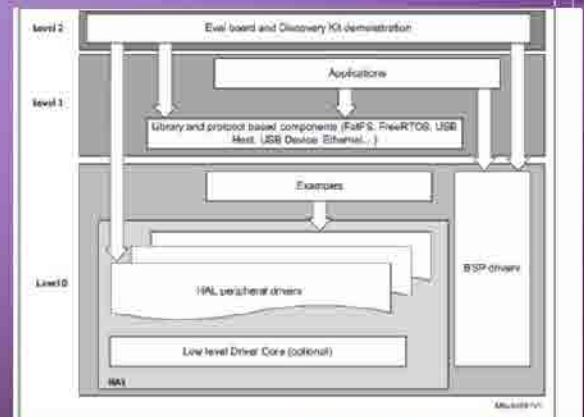
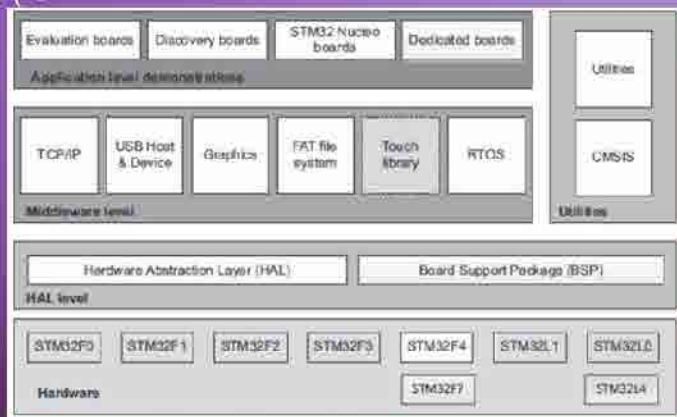
Основні характеристики мікроконтролерів STM32F4



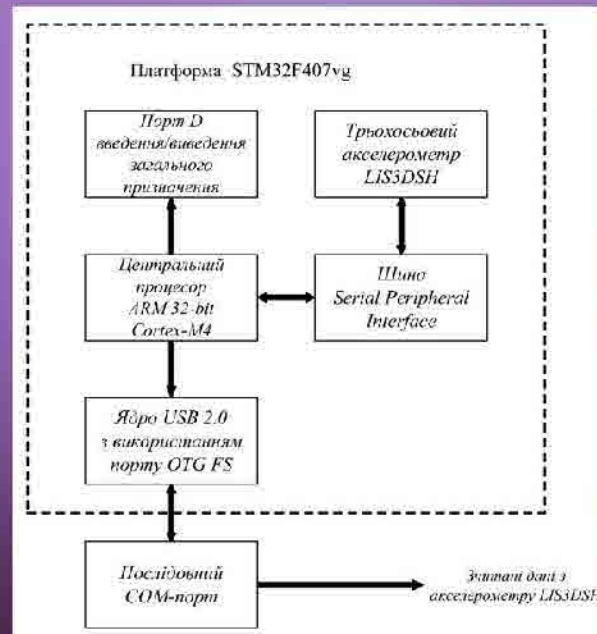
Основні характеристики платформи STM32F407VG

STM32F407		
System	ART Accelerator™	Up to 1-Mbyte Flash memory
Power supply 1.2 V regulator PCB/PDB/TVS	Arm® Cortex®-M4 CPU 168 MHz	Up to 192-Kbyte SRAM
Xtal oscillators 32 kHz + 4 ~26 MHz		FSMC/ SRAM/NOR/NAND/CF/ LCD parallel interface
Internal RC oscillators 32 kHz + 16 MHz		80-byte + 4-Kbyte backup SRAM
PLL		Connectivity
Clock control		Camera Interface
RTC/AWU		3x SPI, 2x I ² S, 3x I ² C
SysTick timer		Ethernet MAC 10/100 with IEEE 1588
2x watchdogs (independent and window)		2x CAN 2.0B
51/82/114/140 I/Os		1x USB 2.0 OTG FS/HS
Cyclic redundancy check (CRC)		1x USB 2.0 OTG FS
Control	Memory Protection Unit (MPU)	SDIO
10x 16-bit timer	Multi-AMB bus matrix	2x USART
2x 16-bit motor control PWM synchronized AC timer	16-channel DMA with Batch Acquisition Mode (BAM)	LIN, IrDA, IrDA, modem control
2x 32-bit timer	True random number generator (TRNG)	Analog
		2-channel 2x 12-bit DAC
		3x 12-bit ADC
		24 channels/2.4 MSPS
		Temperature sensor

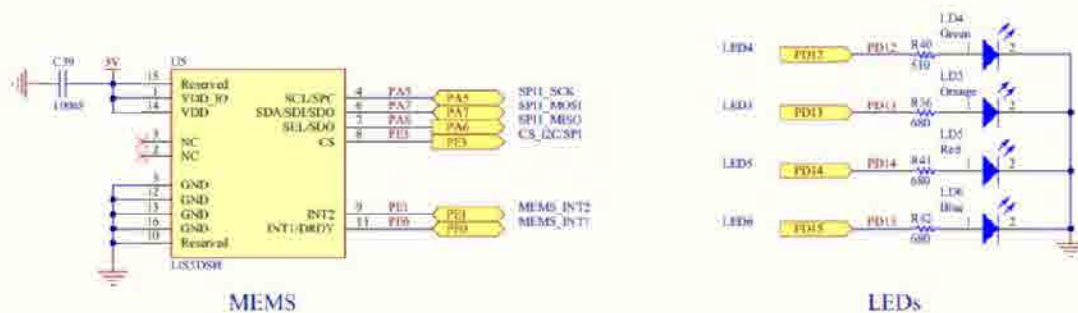
Три рівня архітектури STM32CUBE4_FirmWare



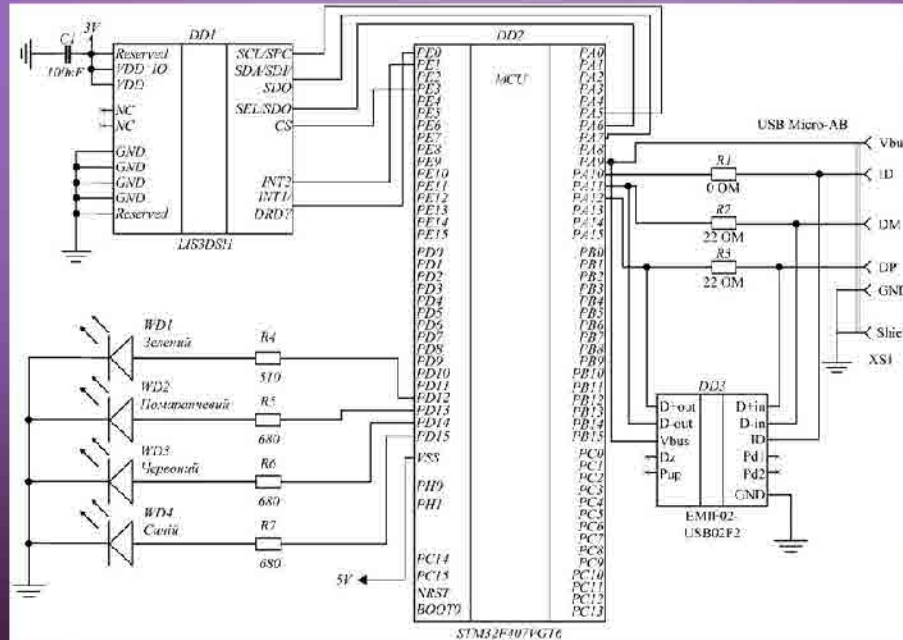
Структурна схема апаратних засобів системи спостереження за сейсмічною активністю



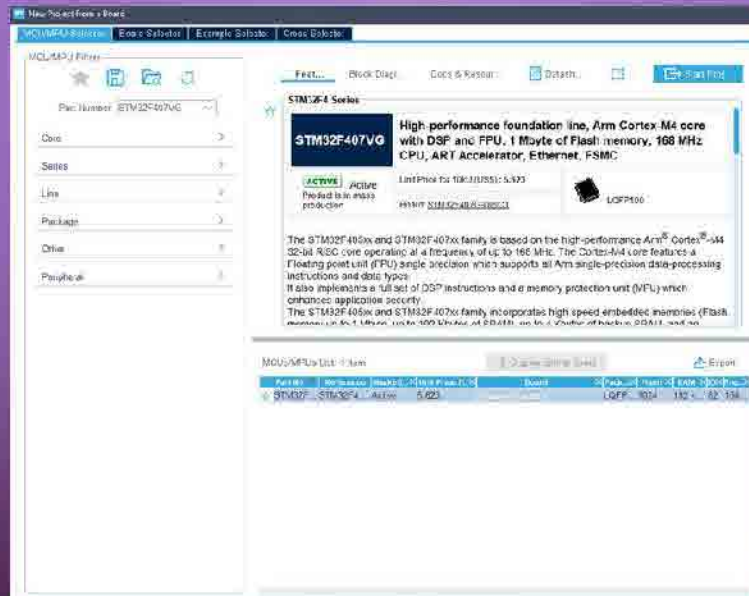
Підключення портів вводу/виводу для датчика LIS3DSH та LED



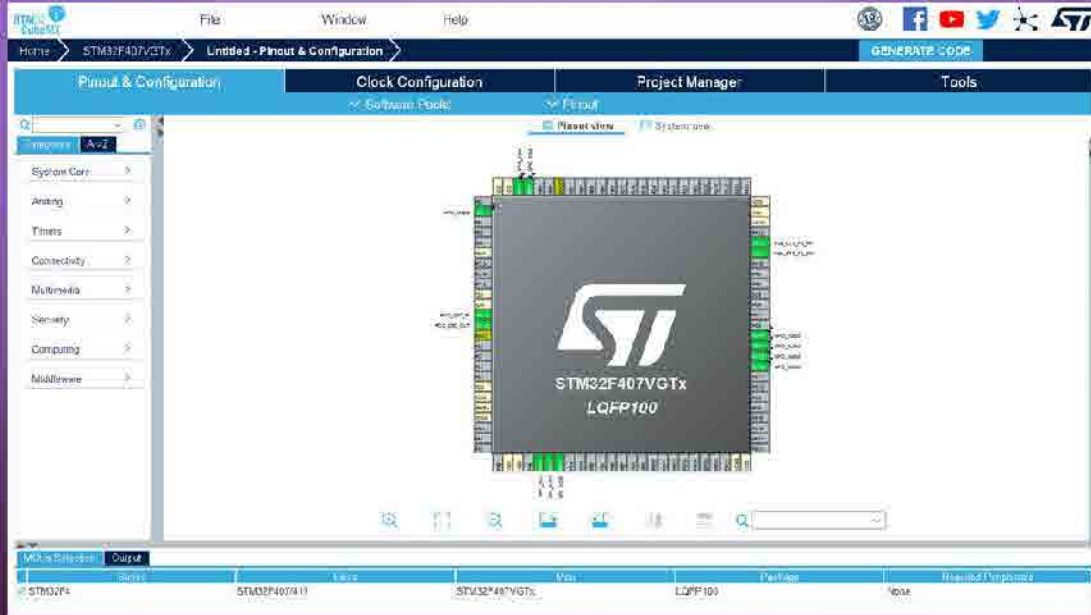
Принципова електрична схема апаратних засобів системи спостереження за сейсмічною активністю



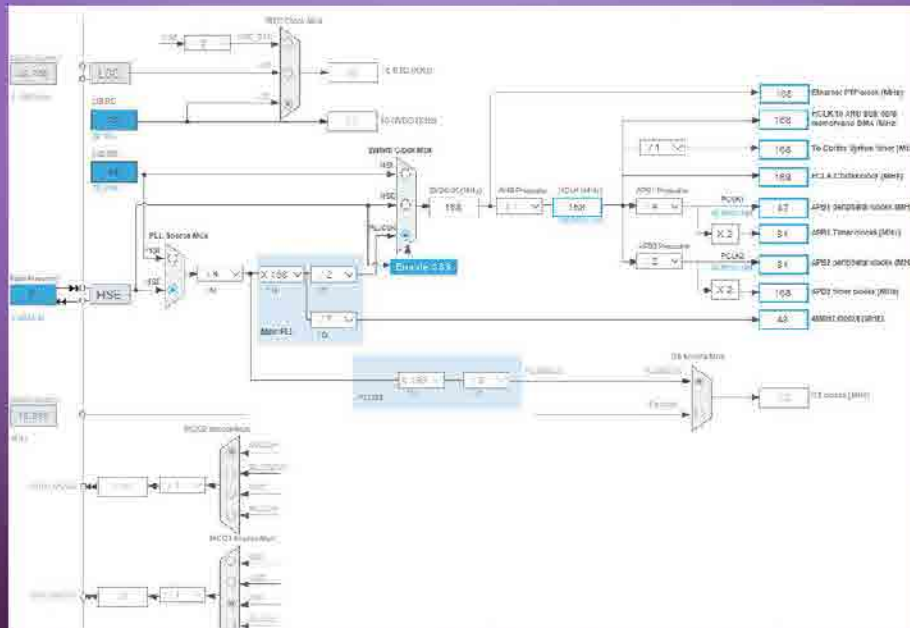
Меню вибору мікроконтролера у STM32CubeMX



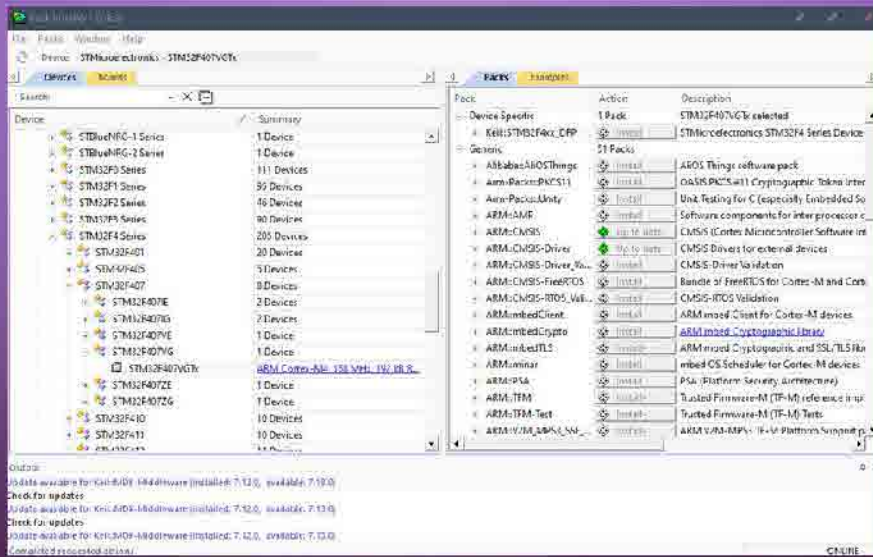
Встановлені режими портів введення-виведення мікроконтролера



Налаштування дерева тактування Clock Configuration



Меню налаштувань середовища розробки Keil uVision

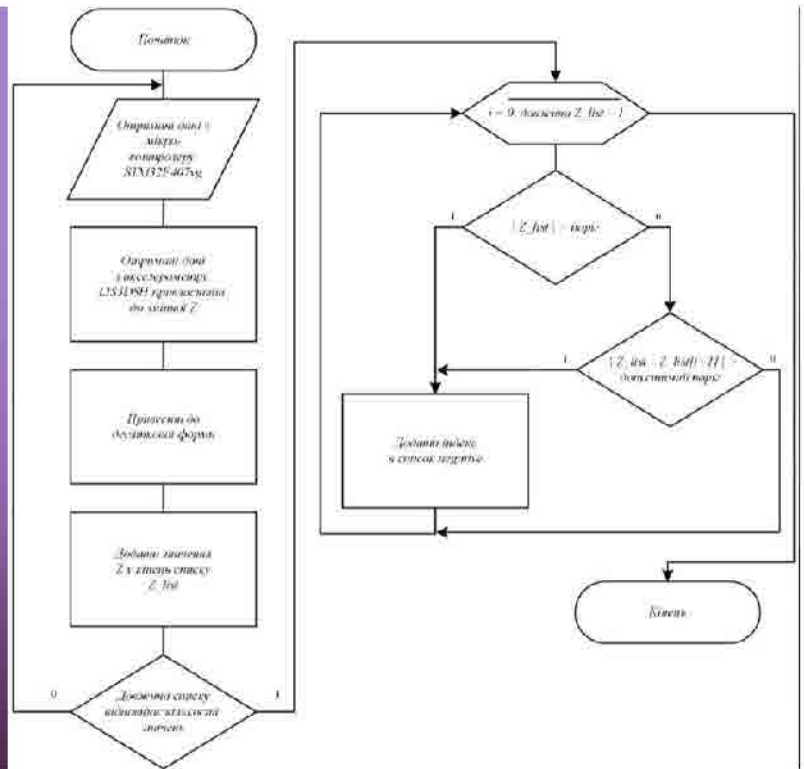


Позиціонування плати у горизонтальному положенні



Позиціонування плати з відхиленням від горизонтального положення

Блок-схема алгоритму обробки даних акселерометра з мікроконтролера плати STM32F407VG



Інтерфейс застосунку при відсутності сейсмічної активності



Інтерфейс застосунку при наявності сейсмічної активності

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Тігіяна Сергія Дмитровича

(прізвище, ім'я та по батькові)

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Обслуговування комп'ютерних систем і мереж»

Тема дипломного проекту: Розробка системи спостереження за сейсмічною активністю на базі платформи STM32

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 86 сторінок. У пояснювальній записці описано реалізацію системи спостереження за сейсмічною активністю на базі платформи STM32. Графічна частина складається з 18 слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.

б) самостійність роботи над проектом: Протягом виконання дипломного проекту здобувач освіти Тігіян Сергій поступово та послідовно виконував всі етапи, проявив ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувач освіти Тігіян Сергій під час роботи над дипломним проектом вивчив достатньо багато літературних та інтернет-джерел за даною тематикою.

Вважаю, що теоретична підготовка дипломника достатня і він готовий до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувач освіти Тігіян Сергій показав вміння організовано працювати над поставленим завданням, застосовувати знання у галузі програмування, електроніки, математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як STM32CubeMX, Keil uVision та ін.

Оцінка розрахункової частини Відмінно

Оцінка графічної частини Відмінно

Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту _____

Кривченко Юрій Вікторович

Місце роботи і посада керівника дипломного проекту ВСП «Одеський технічний фаховий коледж ОНТУ», голова циклової комісії комп'ютерних технологій та програмної інженерії

Підпис _____

« 5 » вересня 2024 р.

РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Тігіяна Сергія Дмитровича

(прізвище, ім'я та по батькові)

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Обслуговування комп'ютерних систем і мереж»

Керівник дипломного проекту (роботи) Кривченко Юрій Вікторович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка системи спостереження за сейсмічною активністю на базі платформи STM32

Обсяг розрахунково-пояснювальної записки 86 сторінок

Обсяг графічної (презентаційної) частини 18 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

Представлений дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений використанню платформи STM32 для реалізації системи спостереження за сейсмічною активністю і складається з пояснювальної записки та мультимедійної презентації, фото реалізованого пристрою та його схеми.

б) характеристика виконання кожного розділу дипломного проекту

Пояснювальна записка складається з основного розділу (аналізу предметної області, реалізації пристрою, розробки його програмного забезпечення), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки розробника. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

Графічна частина складається з 18 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять структурні, принципів та функціональні схеми, фото роботи реалізованого пристрою, блок-схему алгоритму передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання пояснювальної записки відмінна, розробку виконано у повному обсязі.

г) перелік позитивних якостей дипломного проекту *Робота представляє практичний інтерес. Використання вбудованого до платформи STM32F407 акселерометру дозволяє зменшити витрати на реалізацію проекту. Програмне забезпечення дозволяє вести історію сейсмічної активності*

д) основні недоліки дипломного проекту *1. Бажано було б передбачити корпус та окреме живлення для пристрою; 2. У роботі недостатньо обґрунтовано використання графіків показань акселерометра для прогнозування подальшої сейсмічної активності*

Оцінка розрахункової частини Відмінно

Оцінка графічної частини Відмінно

Загальна оцінка Відмінно

Прізвище, ім'я, по батькові рецензента к.т.н. Селіванова Алла Віталіївна

Місце роботи і посада рецензента Одеський національний технологічний університет, декан факультету комп'ютерної інженерії, програмування та кіберзахисту



Підпис: _____

« 17 » червня 2024 р.

Назва документа
Катерина Григорівна Краснокутська

Дата створення документа
12.05.2024 22:18:45 EEST

Дата завантаження документа
12.05.2024 22:41:54 EEST

Ідентифікатор документа
1016245844

Назва бібліотеки
Doc vs Internet + Library

Ідентифікатор бібліотеки
100011688

Назва документа
4KC-S7_Serhiy_Tigian

Дата створення документа 55 Дата завантаження документа 9882 Ідентифікатор документа 71618 Ідентифікатор бібліотеки 3.04 MB Ідентифікатор документа 1016030707



Відсоток подібності
4.73%

1000000000

Не знайдено джерел з Бібліотеки

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

Немає вилучених джерел

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Тігінян Сергій Дмитрович,
здобувач освіти гр. 4КС-57, та

Кривченко Юрій Вікторович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка системи спостереження за сейсмічною активністю на базі платформи STM32» (автор роботи – Тігінян С.Д., керівник роботи – Кривченко Ю.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Тігінян С.Д. /

Керівник



/ Кривченко Ю.В. /

«10» червня 2024 р.