

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ УПРАВЛІННЯ, ПСИХОЛОГІЇ
ТА БЕЗПЕКИ**

Кафедра інформаційних технологій

**РОЗРОБЛЕННЯ ПРИХОВАНОЇ ОХОРОННОЇ СИСТЕМИ, ЧУТЛИВОЇ
ДО ВІБРАЦІЇ ГРУНТУ З ВИКОРИСТАННЯМ
МЕМС-АКСЕЛЕРОМЕТРА**

Кваліфікаційна робота
здобувача вищої освіти
4 курсу денної форми навчання
Дарини ДЕРКАЧ

Науковий керівник:
доцент, кандидат технічних наук
Ігор ФАРМАГА

Рецензент:

вчене звання, науковий ступінь

(Ім'я ПРИЗВИЩЕ рецензента)

Кваліфікаційна робота допущена до захисту
« ___ » _____ 2026 р., протокол № _____

Завідувач кафедри інформаційних технологій
_____ **Олег ЗАЧЕК**
(підпис)

Львів
2026

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....	9
1.1. Вимірювання вібрації ґрунту.....	9
1.2. Огляд існуючих підходів до апаратно-програмної системи моніторингу вібрації ґрунту	14
РОЗДІЛ 2. ЗАСОБИ РОЗРОБКИ АПАРАТНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИХОВАНОЇ ОХОРОННОЇ СИСТЕМИ, ЧУТЛИВОЇ ДО ВІБРАЦІЇ ҐРУНТУ З ВИКОРИСТАННЯМ МЕМС-АКСЕЛЕРОМЕТРА.....	23
2.1. Система автоматизованого проектування і моделювання програмованих пристроїв Proteus VSM.....	23
2.2. Засоби розробки ПЗ в середовищі Arduino IDE для вбудованих систем на платформі Arduino.....	25
РОЗДІЛ 3. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ПРИХОВАНОЇ ОХОРОННОЇ СИСТЕМИ, ЧУТЛИВОЇ ДО ВІБРАЦІЇ ҐРУНТУ З ВИКОРИСТАННЯМ МЕМС- АКСЕЛЕРОМЕТРА.....	33
3.1. Опис апаратної платформи Arduino UNO з МК AVR ATmega328.....	33
3.2. Опис трьохосьового МЕМС-акселерометра ADXL345.....	39
3.3. Опис та вивід значень на рідкокристалічний дисплей.....	48
3.4. Проектування апаратно-програмної охоронної системи, чутливої до вібрації ґрунту в САПР Proteus VSM.....	51
РОЗДІЛ 4. ПРОГРАМНО-АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ АПАРАТНО- ПРОГРАМНОЇ ОХОРОННОЇ СИСТЕМИ, ЧУТЛИВОЇ ДО ВІБРАЦІЇ ҐРУНТУ.....	56
4.1. Алгоритм роботи апаратно-програмної охоронної системи, чутливої до вібрації ґрунту	56
4.2. Розробка програмних модулів для роботи з цифровим трьохосьовим МЕМС- акселерометром ADXL345.....	57
4.3. Програмна реалізація головного алгоритму роботи апаратно-програмної охоронної системи, чутливої до вібрації ґрунту.....	58
4.4. Моделювання та аналіз результатів роботи апаратно-програмної охоронної системи, чутливої до вібрації ґрунту в Proteus ISIS.....	59
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	66
ДОДАТКИ.....	68

СПИСОК УМОВНИХ СКОРОЧЕНЬ

AVR – сімейство 8-розрядних мікроконтролерів архітектури RISC фірми Atmel.

AЗ – апаратна складова системи (технічне забезпечення, hardware);

АЦП (ADC) – аналого-цифровий перетворювач (пристрій, що трансформує вхідну напругу в цифрове значення);

EEPROM – напівпровідникова пам'ять для тривалого зберігання даних;

FLASH – енергонезалежна пам'ять з можливістю багаторазового циклу запису/стирання інформації;

ПЗП (ROM) – постійний запам'ятовувальний пристрій (пам'ять, для зчитування);

ОЗП (RAM) – оперативна пам'ять (для тимчасового зберігання даних);

SPI – послідовний інтерфейс обміну даними (високошвидкісний шинний протокол для зв'язку компонентів);

ПК – персональний комп'ютер (обчислювальна техніка індивідуального користування);

МК – мікроконтролер (спеціалізована мікросхема, для автоматизації управління електронними вузлами).

ЕС – електрична схема (графічне представлення з'єднань між елементами).

ПЗ – програмна складова (сукупність інструкцій та алгоритмів, software).

ВС – вбудована система (комп'ютерний вузол, інтегрований безпосередньо в об'єкт керування);

ЦП – цифровий пристрій (обладнання, що працює з дискретними сигналами);

САПР – система автоматизованого проєктування (інструментарій для розробки технічної документації та моделей);

РКД (LCD) – рідкокристалічний дисплей (пристрій для візуалізації інформації);

I2C – двопровідна шина передачі даних (протокол обміну між мікросхемами за допомогою ліній даних та синхронізації).

АНОТАЦІЯ

Деркач Д., Фармага І. (керівник). Розроблення прихованої охоронної системи, чутливої до вібрації ґрунту з використанням мемс-акселерометра. Бакалаврська кваліфікаційна робота. – Львівський державний університет внутрішніх справ, Львів, 2026.

У кваліфікаційній дипломній роботі розроблено апаратне та програмне забезпечення прихованої охоронної системи, чутливої до вібрації ґрунту з використанням MEMS-акселерометра.

Апаратно-програмна охоронна система, чутлива до вібрації ґрунту виконує вимірювання динамічних параметрів ґрунту, відображає результати на дисплеї та реагує на аномальну вібраційну активність. Чутливість системи можна налаштовувати.

Апаратно-програмна охоронна система побудована на МК сімейства AVR, до якого підключено цифровий 3-ох осьовий MEMS-акселерометр, та рідкокристалічний дисплей. Основою пристрою є мікроконтролер архітектури AVR, інтегрований із цифровим триосьовим сенсором та модулем індикації. Технічна частина роботи містить розробку схемотехнічних рішень та симуляційної моделі в середовищі Proteus VSM. Програмне забезпечення реалізовано мовою C з використанням інструментарію Arduino IDE, а працездатність алгоритмів підтверджена шляхом імітаційного моделювання в Proteus ISIS.

Ключові слова: охоронна система, ADXL345 MEMS-акселерометр, рідкокристалічний символічний дисплей, мова програмування C, вбудоване програмне забезпечення, САПР Proteus VSM, Arduino IDE.

ABSTRACT

Derkach D. Farmaha I. (supervisor). Development of a hidden ground vibration detection security system, using a MEMS accelerometer. Bachelor's thesis. – Lviv State University of Internal Affairs, Lviv, 2026.

The bachelor's thesis describes the development of hardware and software for a covert security system sensitive to ground vibration, utilizing a MEMS accelerometer.

The hardware-software security system measures the dynamic parameters of the soil, displays the results on a screen, and responds to anomalous vibrational activity. The sensitivity of the system is adjustable.

The system is built on an AVR family microcontroller, interfaced with a digital 3-axis MEMS accelerometer and a liquid crystal display (LCD). The core of the device is an AVR-architecture microcontroller integrated with a digital triple-axis sensor and a visualization module. The technical part of the work includes the design of circuit solutions and a simulation model within the Proteus VSM environment.

The software is implemented in C using the Arduino IDE toolkit, and the functionality of the algorithms is verified through simulation in Proteus ISIS.

Keywords: security system, ADXL345 MEMS accelerometer, character LCD, C programming language, embedded software, Proteus VSM CAD, Arduino IDE.

ВСТУП

Сучасний розвиток мікроконтролерної техніки та цифрових сенсорних модулів відкриває широкі можливості для проектування бюджетних вимірювальних комплексів. Одним із перспективних напрямків застосування таких технологій є створення програмно-апаратних засобів для моніторингу вібрацій ґрунту.

Метою даного дослідження є проектування апаратної структури та розробка вбудованого програмного забезпечення для охоронної системи, чутливої до вібрації ґрунту. Комплекс призначений для вимірювання вібраційних показників, їх аналізу та подальшої активації виконавчих механізмів згідно з визначеним алгоритмом. Інформація про поточний стан системи та результати вимірювань транслюється на персональний комп'ютер і символний рідкокристалічний дисплей.

Для побудови апаратно-програмної охоронної системи, чутливої до вібрації ґрунту обрано мікроконтролер архітектури AVR (Atmel), інтегрований у платформу Arduino. Розроблена система характеризується універсальністю, що дозволяє використовувати її як у стаціонарних умовах, так і в складі мобільних вимірювальних лабораторій.

На сьогодні мікроконтролери (МК) складають технологічну базу більшості інтелектуальних систем. з позиції інженерного проектування, головна перевага використання МК полягає у спрощенні та здешевленні процесу реалізації керуючих алгоритмів, що також дозволяє суттєво мінімізувати габарити кінцевого пристрою. Завдяки високому рівню інтеграції периферійних вузлів безпосередньо на кристалі, МК здатні ефективно виконувати математичні операції та забезпечувати двосторонній обмін даними з виконавчим обладнанням.

Сучасна мікропроцесорна техніка за своїми обчислювальними потужностями, обсягами пам'яті та компактністю значно випереджає громіздкі обчислювальні машини минулого століття.

Аналіз останніх досліджень і публікацій. Використання мікропроцесорної техніки у створенні цифрових пристроїв є об'єктом уваги в наукових працях та інженерних розробках. Ці напрацювання формують базу для розвитку сучасних автоматизованих систем контролю та менеджменту. Особливої уваги заслуговує динамічний розвиток методів реєстрації сейсмічних коливань ґрунту, що потребує постійної модернізації технічних рішень. З огляду на це, розробка охоронного комплексу на основі мікроконтролерів AVR, який здатен детектувати вібраційну активність, є актуальним і перспективним науково-технічним напрямком. Наведене обґрунтування дозволяє сформулювати цілі та завдання даного дослідження.

Метою роботи є проектування апаратного і програмного забезпечення прихованої охоронної системи, чутливої до вібрації ґрунту.

Для успішної реалізації поставленої мети необхідно виконати такі науково-технічні завдання:

- проектування архітектури прихованої охоронної системи для детектування сейсмічних коливань ґрунту із застосуванням такої елементної бази: мікроконтролерна платформа Arduino Uno (на базі чіпа ATmega328), цифровий триосьовий акселерометр ADXL345, символний рідкокристалічний модуль 16×2 (WH1602B-YGK-CTK);
- розробка прототипу пристрою в середовищі автоматизованого проектування Proteus;
- формування алгоритму функціонування вбудованої системи моніторингу;
- написання програмного коду для забезпечення коректного зчитування даних із сенсора ADXL345;
- створення програмних модулів для візуалізації вимірювальних показників на символному дисплеї;
- верифікація та аналіз роботи спроектованої моделі шляхом імітаційного моделювання в програмному комплексі Proteus ISIS.

Об’єкт дослідження – проектування охоронної системи, чутливої до вібрації ґрунту.

Предмет дослідження – моделі та засоби проектування охоронної системи, чутливої до вібрації ґрунту

Методи досліджень. Реалізація мети дослідження та розв’язання поставлених завдань базується на застосуванні системного підходу та комплексу наукових методів. Зокрема, за допомогою бібліометричного аналізу було опрацьовано фахову літературу, що дозволило виявити ключові тенденції у розвитку технологій моніторингу. Фундаментальне теоретичне осмислення об’єкта розробки здійснювалося через призму діалектичного, синергетичного, феноменологічного та аксіологічного підходів. Прикладний аспект роботи та розробка технічних рішень ґрунтуються на використанні моделювання, системного аналізу та синтезу, а також застосуванні методів індукції, дедукції та структурного методів.

Структура роботи. Структурно кваліфікаційна робота включає вступну частину, чотири основні розділи, загальні висновки, перелік використаної літератури та додатки. Пояснювальна записка містить 64 сторінки основного тексту, проілюстрованого 37 рисунками та 8 таблицями. Інформаційну базу дослідження складають 17 бібліографічних джерел. Повний обсяг роботи, враховуючи 1 додаток, становить 75 сторінок.

РОЗДІЛ 1

СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1. Вимірювання вібрації ґрунту

В давнину козакі прикладали вухо до землі і визначали звідки рухається ворог і як він далеко. Розвиток мережі станцій моніторингу та модернізація вимірювальної техніки дозволили науковцям фіксувати дедалі більше підземних поштовхів. Окрім суто наукових завдань, сучасні сейсмографи успішно використовуються і в системах безпеки. Вивченням природи сейсмічних хвиль, що виникають унаслідок вібрацій земної кори, займається наука сейсмологія. Залежно від потужності коливань, землетруси класифікують як мікроземлетруси, слабкі та сильні. Для кількісної оцінки енергії, що вивільняється в гіпоцентрі землетрусу, тривалий час використовують локальну шкалу магнітуд Ріхтера. Збільшення показника на одиницю означає, що кількість вивільненої енергії зростає приблизно у 32 рази. Поштовхи магнітудою 2 майже не відчуються людьми, тоді як магнітуда 7 є критичною межею, після якої починаються масштабні руйнування на великих територіях.

Інтенсивність вібрацій ґрунту визначає не загальну енергію, а конкретний вплив стихії на поверхню, споруди та живі організми. У різних країнах діють свої стандарти:

- EMS-98: Європейська макросейсмічна шкала (прийнята як основна в ЄС).
- Shindo: Шкала Японського метеорологічного агентства (7-бальна).
- MM: Модифікована шкала Меркаллі (США).
- MSK-64: 12-бальна шкала, що використовується в Україні.

Характеристика впливу за 12-бальною шкалою:

1. I бал (непомітний): Реєструється виключно приладами.
2. II бали (дуже слабкий): Можуть відчути лише люди у стані спокою на верхніх поверхах або надчутливі тварини.

3. III бали (слабкий): Відчувається всередині приміщень як легка вібрація.
4. IV бали (помірний): Дрижать вікна, двері, поштовхи відчутні для більшості людей.
5. V балів (досить сильний): Хитаються люстри, скрипить підлога, може осипатися побілка.
6. VI балів (сильний): Поява дрібних тріщин у штукатурці та печач.
7. VII балів (дуже сильний): Пошкодження стін, димарів, поява тріщин у вологому ґрунті.
8. VIII балів (руйнівний): Падіння карнизів, великі тріщини в будівлях, розколи на гірських схилах.
9. IX балів (спустошливий): Руйнування дахів і перекриттів, обвали в горах, значні зсуви ґрунту.
10. X балів (нищівний): Масове руйнування будівель, розриви в ґрунті завширшки до 1 метра.
11. XI балів (катастрофічний): Величезні обвали в горах, численні тріщини на земній поверхні, повна руйнація інфраструктури.
12. XII балів (сильна катастрофа): Глобальна зміна ландшафту, величезні зсуви, жодна споруда не витримує навантаження.

Сучасні методи оцінювання сейсмічної активності. Починаючи з другої половини XIX століття, для аналізу наслідків підземних поштовхів науковці використовують спеціалізовані сейсмічні шкали. Попри те, що в побуті часто згадують шкалу Ріхтера, важливо розуміти: вона вимірює магнітуду (енергію в епіцентрі), тоді як для оцінки руйнувань на поверхні застосовують бальні шкали інтенсивності. Світова практика передбачає використання різних систем оцінювання залежно від регіону:

- Україна та країни СНД: Традиційно використовується 12-бальна шкала МСК-64.
- Європейський Союз: Основним стандартом є Європейська макросейсмічна шкала (EMS-98). Вона була остаточно затверджена у 1998 році як вдосконалена версія прототипу 1992 року.

- Інші регіони: У США поширена 12-бальна шкала Меркалі, у країнах Латинської Америки подекуди зустрічається 10-бальна система, а Японія використовує власну 7-бальну шкалу (Shindo).

Таблиця 1.1. Модифікована шкала Меркалі інтенсивності землетрусів

Характеристика землетрусів по модифікованій шкалі Меркалі (Modified Mercalli Intensity Scale)				
Інтенсивність інструментальна	Рівень відчуття	Діапазон прискорень (g)		Руйнування
		мінімум	максимум	
II-III	Слабий (weak)	0,0017	0,014	немає
IV	Легкий (light)	0,014	0,039	немає
V	Помірний (moderate)	0,039	0,092	дуже малі
VI	Сильний (strong)	0,092	0,18	невеликі
VII	Дуже сильний (very strong)	0,18	0,34	помірні
VIII	Руйнівний (severe)	0,34	0,65	помірно-великі
IX	Катастрофічний (violent)	0,65	1,24	великі
X	Екстремальний (extreme)	1,24	>1,24	Дуже великі

Таблиця 1.2. Японська шкала інтенсивності землетрусів (JMA) шиндо

Характеристика землетрусів по Японській шкалі інтенсивності (JMA)				
Інтенсивність Шиндо	Рівень відчуття	Діапазон прискорень ґрунту (m/s^2)		Руйнування
		мінімум	максимум	
Шиндо 1	Відчувають не всі люди	0,008	0,025	немає
Шиндо 2	Відчувають більшість людей	0,025	0,08	немає
Шиндо 3	Хитаються ел. дроти	0,08	0,25	немає
Шиндо 4	Відчутна вібрація ґрунту	0,25	0,80	дуже малі
Шиндо 5-мінус	Видно коливання ел. стовпів	0,80	1,40	невеликі
Шиндо 5-плюс	Важко пересуватися	1,40	2,50	помірні
Шиндо 6-мінус	Важко встояти на ногах	2,50	3,15	помірно-великі
Шиндо 6-плюс	Пересуватися дуже важко	3,15	4,00	великі
Шиндо 7	Пересуватися неможливо	4,00	>4,00	Дуже великі

Сучасне обладнання для моніторингу сейсмічної активності базується на вимірюванні прискорення, яке виникає під час коливань земної поверхні. Історія таких вимірювань розпочалася ще у 132 році, коли китайський філософ і винахідник Чжан Хен (Zhang Heng) створив перший прототип сейсмоскопа. Хоча за сучасними мірками пристрій був доволі простим, він ефективно виконував завдання з фіксації наближення землетрусу та визначення вектора його поширення. Цікаво, що цей же механізм використовували й у військових цілях – для виявлення пересування ворожої кінноти за вібраціями ґрунту.

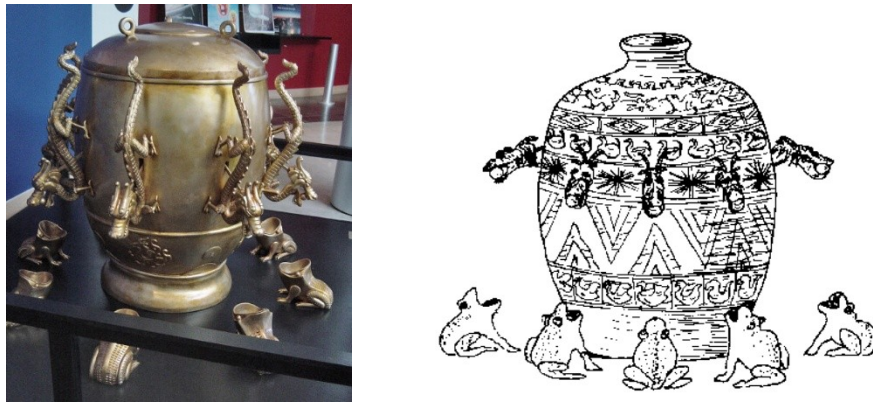


Рис. 1.1. Сейсмоскоп Чан Хенга 132 рік

Поява приладів, здатних здійснювати безперервну фіксацію вібрації ґрунту, припадає на кінець XIX століття. Перші зразки таких пристроїв були розроблені в Італії, а вже у 1895 році в Японії дослідник Джон Мілн представив маятниковий сейсмоскоп, призначений для вимірювання сейсмічних вібрацій. У 1906 році Борис Голіцин створив перший електромагнітний сейсмометр, що вивело точність та якість реєстрації даних на принципово новий рівень.

Сейсмограф (від грецьких слів *seismos* – землетрус та *grapho* – писати), який також називають сейсмометром. Його основним завданням є автоматична реєстрація та запис усіх видів сейсмічних хвиль, що виникають унаслідок підземних поштовхів або штучних вібрацій (Рис.1.2).

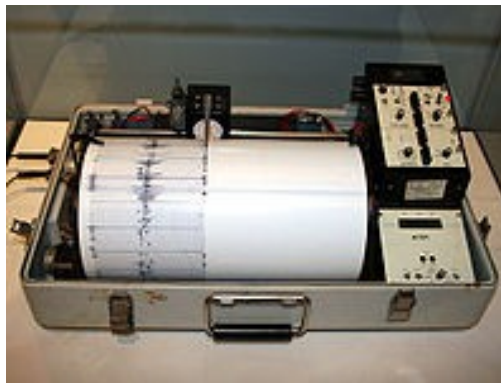


Рис. 1.2. Сейсмограф

В основі конструкції більшості сейсмографів лежить принцип інерції. Типова схема приладу включає масивний вантаж, закріплений на спеціальній пружинній підвісці. Під час підземних поштовхів цей вантаж завдяки своїй інертності намагається зберегти стан спокою, тоді як корпус приладу та його

опора рухаються разом із земною поверхнею. Це створює відносне зміщення між нерухою масою та основою пристрою. За функціональним призначенням прилади поділяють на два основні типи: горизонтальні: призначені для фіксації зміщень у площині поверхні землі, та вертикальні: реєструють коливання, спрямовані вгору та вниз. Традиційно сейсмічні хвилі фіксувалися механічним способом – за допомогою спеціального пера (стилуса), яке креслило графік коливань на рулоні паперу, що безперервно обертався. Для прикладу на Рис.1.3, 1.4 зображено схему такого сейсмографа.

На зміну аналоговим методам прийшли електронні (цифрові) сейсмографи. У таких пристроях паперові носії відсутні, а дані про вібрації перетворюються на електричні сигнали та зберігаються у внутрішній цифровій пам'яті. Це дозволяє здійснювати автоматизовану обробку інформації та передавати її на великі відстані в режимі реального часу.

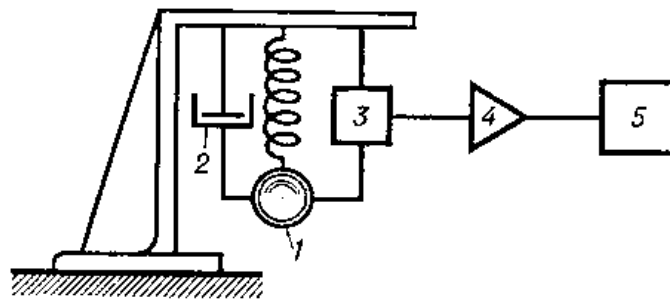


Рис. 1.3. Схема сейсмографа: 1 – вантаж сейсмометра; 2 – демпфер (пристрій для гашення власних коливань вантажу); 3 – перетворювач руху вантажу в електричний сигнал; 4 – фільтр-підсилювач; 5 – самописець

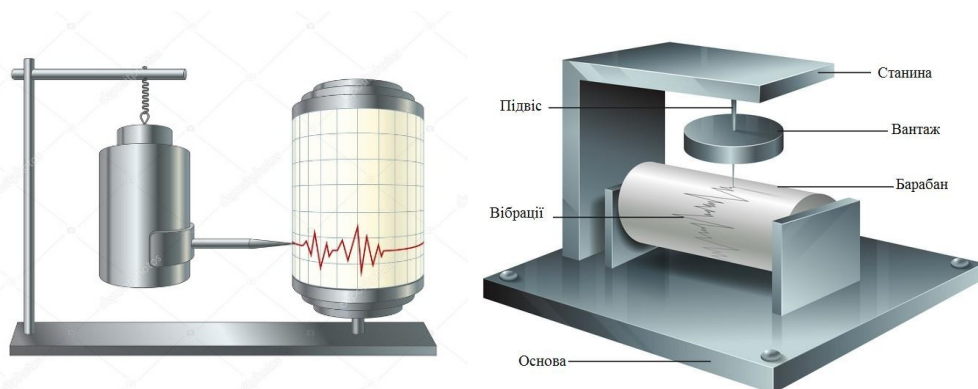


Рис. 1.4. Зовнішній вид механічних сейсмометрів

Для вимірювання параметрів вібрації ґрунту використовують сейсмометри і акселерометри. Сейсмометри – це в більшості високочутливі

інструменти, які вимірюють малі вібрації на великій віддалі. Акселерометри можна використати у випадках, коли не потрібна велика точність, а цікавий сам факт вібрації ґрунту на невеликій відстані, що можна використати для охоронних систем. Трьохосьові акселерометри дозволяють визначити не тільки інтенсивність вібрації ґрунту але й напрямок.

1.2. Огляд існуючих підходів до апаратно-програмної системи моніторингу вібрації ґрунту

На основі вивчення науково-технічної літератури та актуальних онлайн-ресурсів було зібрано вичерпний масив даних щодо сучасних розробок у галузі реєстрації вібрацій ґрунту. Сьогодні на ринку представлено широкий спектр цифрових систем моніторингу. Серед них виділяють як традиційні механічні рішення, так і сучасні комплекси, інтегровані з мікроконтролерними системами керування. До недавня механічні та електромеханічні пристрої склали основу сейсмологічного обладнання.

Проте такі прилади мають суттєві недоліки: висока собівартість: через наявність прецизійних механічних вузлів ціна інструментів є надто високою для масового використання чи аматорських досліджень. Технологічна складність: жорсткі вимоги до точності виготовлення деталей обмежують можливості їхнього серійного промислового виробництва.

Альтернативним вектором розвитку сейсмометрії став відхід від використання твердих інерційних мас. У таких системах робочим тілом виступає рідкий електrolіт. Зовнішні сейсмічні коливання провокують рух рідини всередині датчика, що згодом трансформується в електричний сигнал. Такі прилади мають низку експлуатаційних переваг: доступність: відсутність складних механічних вузлів значно знижує витрати на виробництво; довговічність: середній термін служби таких сенсорів становить близько 15 років; надійність: спрощена конструкція полегшує як процес виготовлення, так і подальше обслуговування пристрою в польових умовах.

Інтеграція обчислювальної техніки та аналого-цифрових перетворювачів (АЦП) спричинила справжню революцію у сфері сейсмометрії. Це дозволило вивести функціональні можливості обладнання на якісно новий рівень. Появилася можливість одночасного фіксування і аналізу в реальному часі сигналів з багатьох сеймосенсорів, враховувати спектри сигналів. Сучасні комплекси можуть попередити про початок руйнівних коливань за кілька хвилин до їхнього приходу. Головним бар'єром для масового розгортання таких систем залишається їхня вартість: професійний сейсмограф високого класу коштує в середньому близько 20 000 доларів, що робить розробку доступніших альтернативних пристроїв надзвичайно актуальним завданням.

На Рис.1.5 зображено промислові сейсмометри для детектування вібрацій ґрунту.

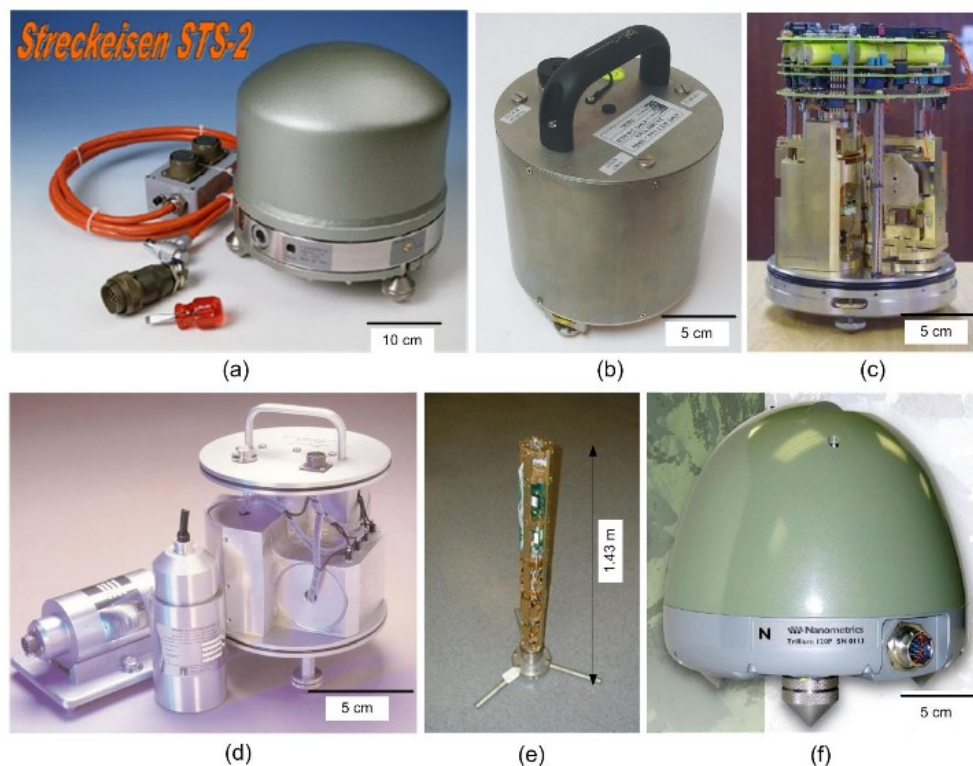


Рис. 1.5. Промислові сейсмометри (Streckheisen STS-2 (a), Guralp CMG-40T (b), Guralp CMG-3T (c), Sercel L-4 (d), Geotech KS-2000 (e), Nanometrics Trillium 120P (f))

Класичні сейсмометри характеризуються значною масою, великими габаритами та високою вартістю. Ці фактори суттєво обмежують їхнє масове

впровадження у багатьох сферах діяльності. На противагу їм, сучасні МЕМС-сейсмометри (мікроелектромеханічні системи) вирізняються мініатюрністю, малою вагою та доступною ціною, що робить їх оптимальним вибором для розгалужених мереж моніторингу. Конструкції таких компактних пристроїв наведено на Рис. 1.6. TM63C: Цифровий віброметр ручного типу (орієнтовною вартістю 60\$) призначений для промислової діагностики обладнання, але він не пристосований для моніторингу сейсмічної активності. Grillo Active System: На сьогодні це чи не єдина спеціалізована модель у бюджетному сегменті (близько 50\$), розроблена в Мехіко. Пристрій функціонує як портативна система, здатна сповістити про наближення підземних поштовхів за 120 секунд до їх початку. Завдяки низькій вартості ця система раннього попередження доступна для широких верств населення. Це свого роду портативна охоронна сигналізація, якою можуть користуватися навіть небагаті люди



Рис. 1.6. Ручний цифровий сенсор вібрацій TM63C (60\$) та Grillo Active System Offers Early Earthquake

Технічна архітектура сучасних портативних сейсмографів зазвичай базується на поєднанні акселерометра, мікроконтролера, а також модулів Wi-Fi та GPS-антени. Така конфігурація дозволяє автоматизувати процес передачі даних у глобальну мережу для оперативного сповіщення населення та подальшого аналізу сейсмічної ситуації. Після успішної апробації цієї методики в Мексиці розглядається можливість масштабування подібних мереж на світовому рівні. На сьогодні повноцінна національна цифрова

система раннього попередження функціонує лише в Японії, проте її впровадження та утримання потребують колосальних фінансових витрат. Дефіцит доступних аналогів у світі призводить до трагічних наслідків: щорічно через землетруси гине близько 13 тисяч людей, а економічні втрати сягають мільярдів доларів. Це підтверджує критичну необхідність створення недорогих та ефективних систем моніторингу. Отже розробка подібних систем є актуальною.

В дипломній роботі запропоновано автономну охоронну систему, чутливу до вібрації ґрунту яка функціонує на основі трьохосьового МЕМС-акселерометра та призначена для реєстрації вібраційних коливань ґрунту.

Запропоноване рішення поєднує в собі можливості індикаторного сейсмографа з розширеним функціоналом для безперервного моніторингу, цифрової обробки та візуалізації отриманих даних. Серед ключових переваг системи варто виділити: конструктивну досконалість-відсутність масивних механічних вузлів робить пристрій максимально компактним і надійним, економічну доступність- собівартість розробки є значно нижчою порівняно з традиційними механічними аналогами, що мають схоже цільове призначення. Цей підхід дозволяє створити ефективний та бюджетний інструмент для локального моніторингу сейсмічної активності або захисту периметра об'єктів від несанкціонованого доступу.

Для вимірювання вібрацій ґрунту можна використати 3-х осьові МЕМС акселерометри, Ними можна виміряти вібрації ґрунту і оцінити його інтенсивність та напрям. Для прикладу: MEMS акселерометр ADXL345 має 13 біт – роздільну здатність при діапазоні вимірювання 16g. Його чутливість буде $16/2^{13}=0.002\text{g/lsb}$, це цілком достатньо для детектування легкої вібрації ґрунту, не кажучи вже про екстремальну. Акселерометр встановлюють на платформу міцно зв'язаною з землею. В початковому положенні осі x і y знаходяться в площині горизонту, а вісь z ортогональна до осей x і y (Рис. 1.7). На рисунку зображено кути нахилу α – кут між горизонтом і x-віссю

акселерометра, β – кут між горизонтом та у-віссю, і γ як кут між гравітаційним вектором тяжіння і z-віссю.

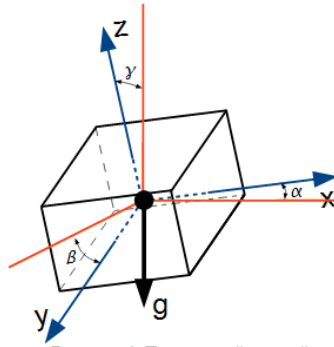


Рис.1.7. Осі трьохосьового акселерометра

Коли сила тяжіння діє тільки на вісь z, отримаємо, значення інших кутів рівними 0. Значення кутів обчислюємо за формулами:

$$\alpha = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \quad (1.1)$$

$$\beta = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \quad (1.2)$$

$$\gamma = \arctan\left(\frac{A_z}{\sqrt{A_x^2 + A_y^2}}\right) \quad (1.3)$$

де A_x, A_y, A_z – виміряні акселерометром значення прискорень.

Потрібно виставити платформу таким чином, щоб значення кутів α, β, γ були рівні 0. Тоді, проєкції прискорення $A_x=0, A_y=0$, а $A_z=g$ де ($g=9,8 \text{ м/с}^2$). Загальний вектор прискорення можна обчислити за формулою:

$$A = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (1.4)$$

де A – загальний вектор прискорення $=1g$ ($9,8 \text{ м/с}^2$). Приведене значення прискорення A_0 , яке використовують для аналізу вібрації ґрунту рівне: $A_0=A-g$, де $g=9,8 \text{ м/с}^2$ – прискорення вільного падіння. Для отримання більш точних вимірювань потрібне калібрування акселерометра.

Калібрування акселерометра. MEMS-акселерометр це механічний пристрій і на нього буде діяти статичне “навантаження”. В свою чергу це приведе до зміни чутливості і зміщенню прискорень. Акселерометр буде міряти значення прискорень з гіршою точністю. Знизити похибку допоможе

калібровка нульового значення акселерометра і його чутливості. На Рис.1.8 зображено платформу для калібрування акселерометрів, яка дозволяє точно повертати акселерометр на заданий кут.

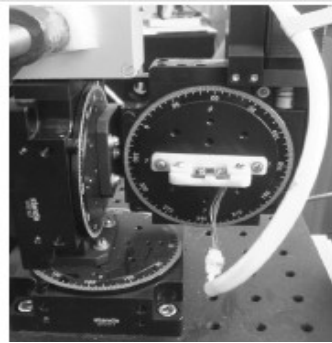


Рис. 1.8. Пристрій для калібрування акселерометра

Для калібровки акселерометра не обов'язково використовувати дороге обладнання. Достатньо взяти декілька показів акселерометра, при дії на нього тільки сили тяжіння. З врахуванням чутливості сенсора і початкового зміщення, значення від акселерометра можна представити такою формулою:

$$A_1 = A_0 + K \cdot A_d \cdot \sin(\alpha) \quad (1.5)$$

де K – коефіцієнт чутливості, A_0 – початкове зміщення, A_d – дійсне значення прискорення, що діє на сенсор, і рівне $1g$, α – кут між дійсним прискоренням і чутливою віссю сенсора. Задача калібровки зводиться до знаходження величин A_0 і K . Для знаходження вказаних величин, потрібно зняти покази з акселерометра, в положеннях, коли вісь чутливості послідовно повернута на кути 0° , 90° , 180° і 270° відносно початкового положення. Отримані значення можна представити:

$$A_1 = A_0 + K \cdot A_d \cdot \sin(\alpha) \quad (1.6)$$

$$A_2 = A_0 + K \cdot A_d \cdot \sin\left(\alpha + \frac{\pi}{2}\right) \quad (1.7)$$

$$A_3 = A_0 + K \cdot A_d \cdot \sin(\alpha + \pi) \quad (1.8)$$

$$A_4 = A_0 + K \cdot A_d \cdot \sin\left(\alpha - \frac{\pi}{2}\right) \quad (1.9)$$

Після додавання всіх виразів отримаємо:

$$A_0 = \frac{1}{4}(A_1 + A_2 + A_3 + A_4) \quad (1.10)$$

Для обчислення коефіцієнта чутливості використаємо формули:

$$\sin\left(\alpha + \frac{\pi}{2}\right) = \cos(\alpha) \text{ і } \sin^2(\alpha) + \cos^2(\alpha) = 1$$

Отримаємо: $(A_1 - A_3)^2 + (A_2 - A_4)^2 = 4 \cdot K^2 \cdot A_0^2 \cdot (\sin^2(\alpha) + \cos^2(\alpha))$

Звідки

$$K \cdot A_0 = \frac{1}{2} \cdot \sqrt{(A_1 - A_3)^2 + (A_2 - A_4)^2} \quad (1.11)$$

Розглянутий спосіб калібрування не потребує початкової орієнтації акселерометра. Наведену послідовність дій необхідно провести для кожної осі акселерометра. Калібрування ADXL345 акселерометра полягає у знаходженні значень Δx , Δy , Δz – зміщень для кожної його осі X, Y, Z та обчислення відповідних значень для запису їх у реєстри зміщення OFSX, OFSY, OFSZ. Вміст кожного реєстра автоматично додається до виміряного значення прискорення по відповідній осі. Молодшому біту реєстрів зміщення відповідає значення 15,6 mg/LSB. Нехай акселерометр працює в режимі $\pm 16g$ з чутливістю 3,9 mg/LSB. При калібруванні отримали значення для Z-осі $\Delta z = 420$. В реєстр OFSZ потрібно занести комплементарне число offsetZ до числа $(\Delta z \cdot 3,9 / 15,6)$:

$$offsetz = 255 - \left(\Delta z \cdot \frac{3,9}{15,6}\right) = 255 - \left(420 \cdot \frac{3,9}{15,6}\right) = 150 \quad (0 \times 96) \quad (1.12)$$

де 255 – роздільна здатність для чутливості 3,9 mg/LSB.

На Рис.1.9 зображено акселерометр та його система координат.



Рис. 1.9. Осі чутливості ADXL345 акселерометра

При орієнтації акселерометра в просторі він приймає додатні та від'ємні значення, Рис.1.10.

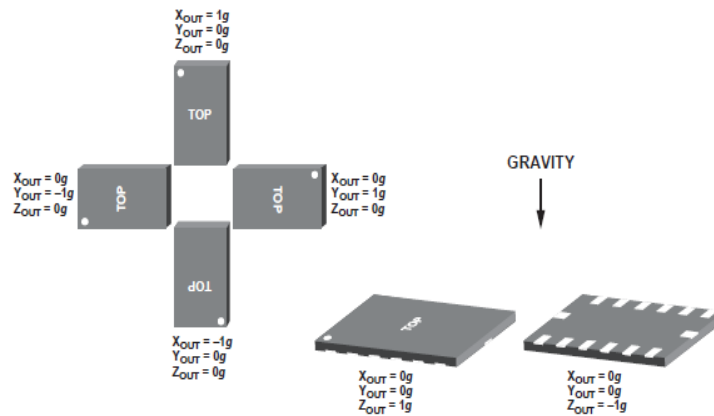


Рис. 1.10. Реакція акселерометра на зміну просторової орієнтації

В Табл.1.3 наведені значення вихідних даних ADXL345 акселерометра в залежності від вибраної роздільної здатності.

Таблиця 1.3. Значення вихідних даних ADXL345 акселерометра в залежності від вибраної роздільної здатності

Діапазон прискорення та роздільна здатність	Чутливість	Величина зміряного прискорення	Значення вихідних даних
±2g (10 біт)	3,9 mg/LSB	+1 g	256(0X100)
±2g (10 біт)	3,9 mg/LSB	-1 g	768(0X300)
±4g (10 біт)	7,8 mg/LSB	+1 g	128(0X80)
±4g (10 біт)	7,8 mg/LSB	-1 g	896(0X380)
±8g (10 біт)	15,6 mg/LSB	+1 g	64(0X40)
±8g (10 біт)	15,6 mg/LSB	-1 g	960(0X3C0)
±16g (10 біт)	31,2 mg/LSB	+1 g	32(0X20)
±16g (10 біт)	31,2 mg/LSB	-1 g	992(0X3E0)
±4g (11 біт)	3,9 mg/LSB	+1 g	256(0X100)
±4g (11 біт)	3,9 mg/LSB	-1 g	1792(X700)
±8g (12 біт)	3,9 mg/LSB	+1 g	256(0X100)
±8g (12 біт)	3,9 mg/LSB	-1 g	3840(0XF00)
±16g (13 біт)	3,9 mg/LSB	+1 g	256(0X100)
±16g (13 біт)	3,9 mg/LSB	-1 g	7936(1F00)

Для знаходження значень зміщення, встановлюють акселерометр на горизонтальну платформу таким чином, що вісь Z акселерометра співпадає з вектором сили тяжіння = 1g, а по осях X та Y сила тяжіння рівна 0. Калібрування полягає в знаходженні мінімального та максимального значень для кожної осі акселерометра (шість значень, по два для кожної осі). Маючи X_{amin} та X_{amax} обчислюємо:

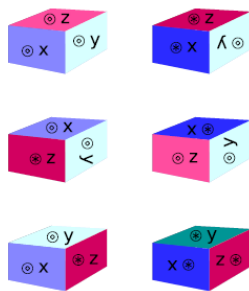
$$\Delta x = 0,5 \cdot (X_{amax} + X_{amin}) \quad (1.13)$$

Нехай $X_{amin} = -250$ LSB, $X_{amax} = 270$ LSB, $(270 + (-250)) \cdot 0,5 = 10$ LSB. Отже, $\Delta x = 10$ LSB. Аналогічно знаходимо Δy та Δz .

$$\Delta y = 0,5 \cdot (Y_{amax} + Y_{amin}) \quad (1.14)$$

$$\Delta z = 0,5 \cdot (Z_{amax} + Z_{amin}) \quad (1.15)$$

Розмістити вісь x ADXL345 вздовж вектора g . Розмістити вісь x ADXL345 протилежно вектору g . Отримати два значення X_{amin} , X_{amax} . Знайти Δx . Такі ж дії виконати для z та y осей. Знайти Δz та Δy .



План калібровочних замірів

Орієнтація ADXL345	Ax	Ay	Az
z вісь вверх	0	0	+1g
z вісь вниз	0	0	-1g
y вісь вверх	0	+1g	0
y вісь вниз	0	-1g	0
x вісь вверх	+1g	0	0
x вісь вниз	-1g	0	0

Отримавши значення X_{amin} , X_{amax} , Y_{amax} , Y_{amin} , Z_{amax} , Z_{amin} можна обчислити чутливість акселерометра вздовж кожної осі. Чутливість $Gain_x$ по осі X рівна половині суми абсолютних значень X_{amin} і X_{amax} і обчислюється за формулою:

$$Gain_x = \frac{(|X_{amin}| + |X_{amax}|)}{2} \quad (1.19)$$

Для прикладу: $X_{amin} = -250$, $X_{amax} = 270$, $Gain_x = (|-250| + |270|) / 2 = 210$. Обчислення проводимо для знаходження $Gain_y$, $Gain_z$. Реальне прискорення вздовж осі X з врахування чутливості та зміщення знаходимо по формулі:

$$A_{xreal} = \frac{(A_x - A_{xoff})}{Gain_x} \quad (1.20)$$

де A_x – виміряне ADXL345 значення по осі X, A_{xoff} – похибка зміщення по осі X, $Gain_x$ – чутливість по осі X. Для осей Z та Y:

$$A_{yreal} = \frac{(A_y - A_{yoff})}{Gain_y} \quad (1.21)$$

$$A_{zreal} = \frac{(A_z - A_{zoff})}{Gain_z} \quad (1.22)$$

Розмірність прискорень A_{xreal} , A_{yreal} , A_{zreal} в одиницях 1g. Отже з допомогою каліброваного трохосьового МЕМС акселерометра можна побудувати охоронну систему, чутливу до вібрації ґрунту.

РОЗДІЛ 2

ЗАСОБИ РОЗРОБКИ АПАРАТНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИХОВАНОЇ ОХОРОННОЇ СИСТЕМИ, ЧУТЛИВОЇ ДО ВІБРАЦІЇ ҐРУНТУ З ВИКОРИСТАННЯМ МЕМС- АКСЕЛЕРОМЕТРА

2.1. Система автоматизованого проектування і моделювання програмованих пристроїв Proteus VSM

Proteus VSM – це багатофункціональна система автоматизованого проектування (САПР), розроблена британською компанією Labcenter Electronics. Програмний комплекс призначений для наскрізного проектування електронних пристроїв, зокрема тих, що базуються на мікроконтролерах різних архітектур та платформі Arduino. Основною особливістю Proteus є поєднання потужних засобів схемотехнічного моделювання з можливістю віртуального виконання програмного коду мікропроцесорних систем. Програма дозволяє:

- Розробляти складні електричні принципові схеми.
- Проводити інтерактивну симуляцію роботи пристрою з використанням віртуальних вимірювальних приладів (осцилографів, вольтметрів тощо).
- Виконувати відлагодження програмного забезпечення безпосередньо у віртуальному середовищі.

Програмний пакет містить велику базу даних, що налічує понад 6000 моделей аналогових і цифрових компонентів. Proteus підтримує моделювання широкого спектра мікроконтролерів, включаючи сімейства AVR, PIC, ARM7, 8051, Motorola та інші. Система сумісна з більшістю сучасних компіляторів та асемблерів, що дозволяє відлагоджувати складні багатопроцесорні системи. Важливою перевагою є підтримка SPICE-моделей. Якщо необхідний компонент відсутній у стандартній бібліотеці, користувач може імпортувати його модель безпосередньо з сайту виробника або створити власну.

Система Proteus складається з двох ключових модулів, що забезпечують повний цикл розробки:

- **ISIS (Intelligent Schematic Input System):** Графічний редактор для проектування принципів схем. Саме в цьому модулі відбувається основний процес моделювання та налагодження логіки роботи пристрою. Отримана інформація про проект може бути автоматично передана для подальшого проектування друкованої плати.
- **ARES (Advanced Routing and Editing Software):** Спеціалізований модуль для трасування та проектування друкованих плат. Він оснащений вбудованим автотрасувальником ELECTRA, функцією автоматичного розміщення компонентів та засобами тривимірної візуалізації готового виробу.

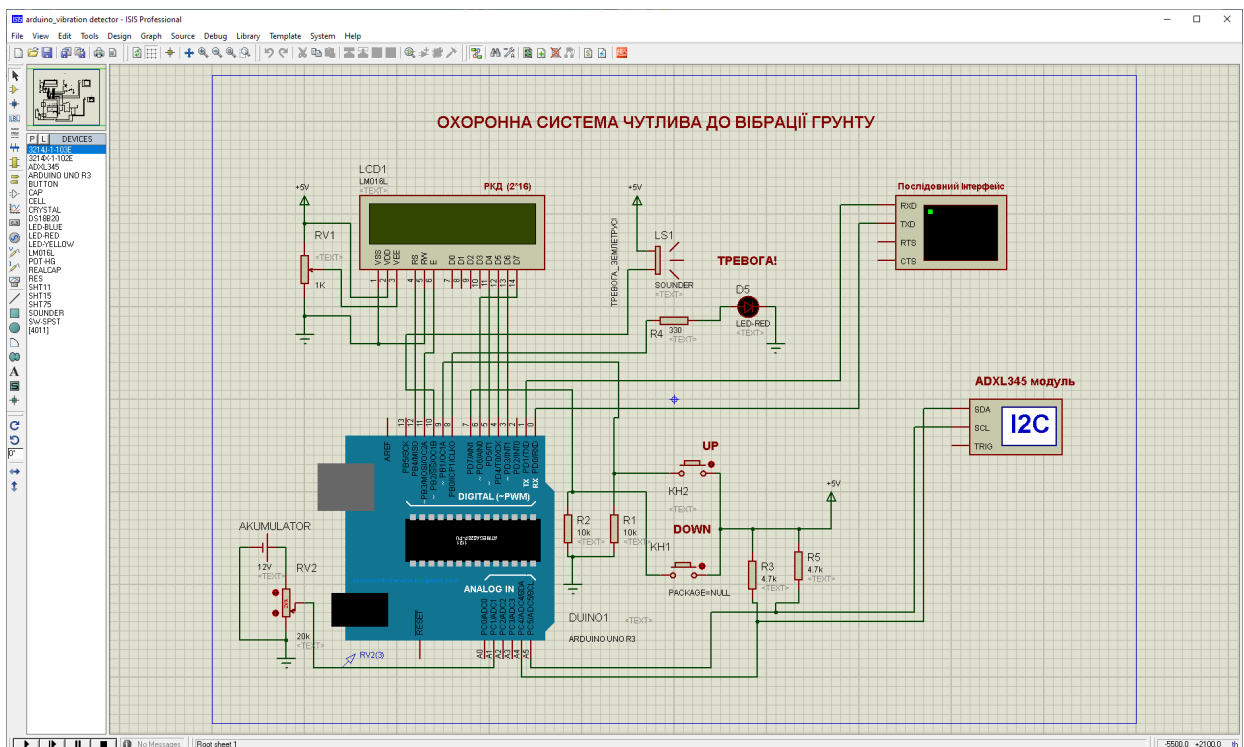


Рис. 2.1. Середовище Proteus ISIS

Програмний комплекс Proteus володіє розширеним набором інструментів для взаємодії віртуальних моделей із фізичним обладнанням, що значно розширює межі тестування:

- **COMPIB:** Спеціалізований компонент, який забезпечує зв'язок віртуальної схеми з фізичним COM-портом комп'ютера. Це дозволяє

тестувати обмін даними між симуляцією та реальним зовнішнім пристроєм.

- USBCONN: Інструментарій для підключення проекту до реального USB-інтерфейсу ПК, що необхідно для відлагодження USB-пристроїв.

Головною перевагою Proteus VSM є можливість верифікації вбудованого програмного забезпечення (Firmware) на етапі до створення прототипу. Система демонструє високу сумісність із провідними середовищами розробки, зокрема:

- Arduino IDE (за умови встановлення відповідних бібліотек компонентів);
- CodeVisionAVR та WinAVR (орієнтовані на роботу з мікроконтролерами архітектури AVR);
- ICC (підтримує широкий спектр МК, включаючи AVR, MSP430 та ARM7);
- HiTECH (використовується для програмування чипів серій 8051 та PIC).

Використання Proteus у поєднанні з цими інструментами дозволяє розробнику виявити логічні помилки в коді та схемотехнічні недоліки ще до початку монтажних робіт, що суттєво заощаджує час та ресурси.

2.2. Засоби розробки ПЗ в середовищі Arduino IDE для вбудованих систем на платформі Arduino

Arduino представляє собою універсальну екосистему для швидкої розробки електронних систем. Висока популярність платформи зумовлена її доступністю, відкритістю архітектури та інтуїтивно зрозумілим програмним середовищем. Важливою особливістю є можливість прошивки мікроконтролера безпосередньо через інтерфейс USB, що усуває потребу у використанні додаткових зовнішніх програматорів.

Пристрої, побудовані на базі Arduino, здатні ефективно взаємодіяти з навколишнім середовищем шляхом зчитування сигналів від широкого спектра

аналогових та цифрових датчиків. Керування різноманітними виконавчими механізмами та актуаторами.

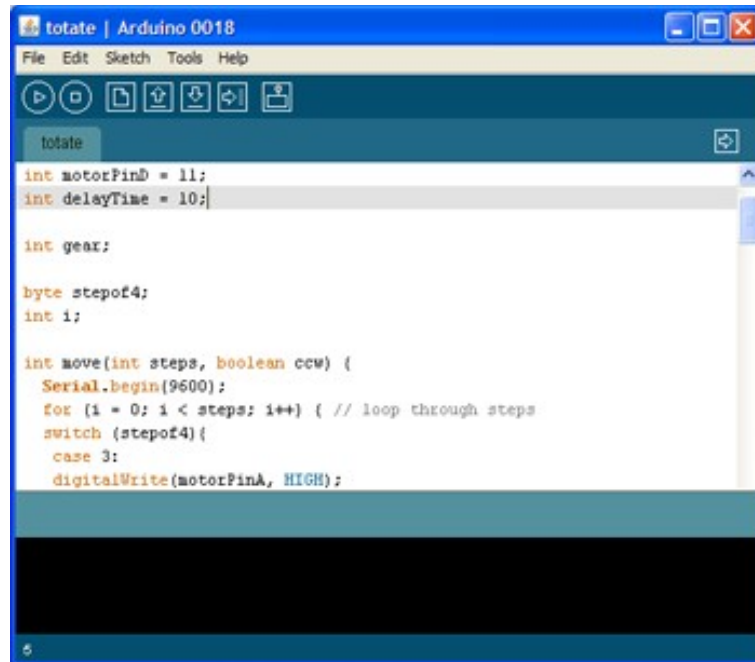


Рис. 2.2. Середовище Arduino IDE



Основним інструментом розробки є спрощена версія мов C/C++, яка відрізняється низьким порогом входження, що робить її зручною навіть для початківців. Проте, програмна частина Arduino є гнучкою і має значний потенціал для масштабування:





- Відкритий вихідний код: Професійні розробники можуть вносити власні корективи безпосередньо в середовище розробки.
- Бібліотечна підтримка: Функціонал легко розширюється шляхом підключення стандартних і призначених для користувача бібліотек C++.
- Сумісність: Платформа дозволяє інтегрувати низькорівневий код AVR-C безпосередньо в проекти Arduino, що дає змогу оптимізувати роботу пристрою на рівні регістрів мікроконтролера.

Програмне забезпечення для роботи з Arduino – це багатофункціональне середовище, що поєднує інструменти для написання коду та його подальшої прошивки в мікроконтролер. Структура інтерфейсу містить такі ключові елементи:

- Редактор вихідного коду: Текстова область, призначена для написання програм. Вона підтримує стандартні функції редагування, такі як копіювання, вставка, а також інструменти швидкого пошуку та заміни фрагментів тексту.
- Область сповіщень: Спеціальне інформаційне поле, де відображаються результати поточних операцій (наприклад, підтвердження збереження файлу або звіти про виявлені синтаксичні помилки).
- Консоль виводу (термінал): Вікно, що демонструє детальні системні звіти про хід компіляції програми та об'єм пам'яті, який вона займає. Тут також відображаються розширені повідомлення про критичні помилки.
- Панель керування та меню: Набір функціональних клавiш, що дозволяють одним натисканням виконати верифікацію (перевірку) коду, завантажити його в апаратну частину, створити новий проект або відкрити монітор послідовного порту для зчитування даних у реальному часі.

Програмний код, що створюється в середовищі Arduino, отримав специфічну назву – скетч. Процес взаємодії між софтом та “залізом” відбувається безпосередньо через USB-з'єднання, що забезпечує швидкий цикл розробки: від написання коду в редакторі до його виконання на платі. Скетч пишуть в текстовому редакторі, що має інструменти вирізки/вставки, пошук/заміна тексту. Під час збереження проекту в області повідомлень є пояснення, а також відображаються помилки програмування. Клавiші панелі інструментів дозволяють перевірити і записати програму, створити, відкрити і зберегти скетч, відкрити доступ до послідовної шини:

- *Verify/Compile* – Верифікація вихідного коду, виявлення синтаксичних помилок та подальша компіляція програми.
-  *Stop* – Завершення сеансу передачі даних через Serial Monitor, що супроводжується блокуванням або візуальним затемненням кнопок інтерфейсу.
-  *New* – Створення нового скетча.

-  *Open* – Перехід до каталогу проектів у блокноті. Меню відкривається шляхом натискання на відповідний елемент керування у поточному робочому вікні.
-  *Save* – Збереження скетча.
-  *Upload to I/O Board* – Компілює програмний код і завантажує його в Arduino-пристрій.
-  *Serial Monitor* – Відкриття моніторинга послідовної шини (Serial monitor).

Інтерфейс програми містить п'ять основних розділів: File, Edit, Sketch, Tools та Help. Перелік доступних функцій у цих меню може змінюватися залежно від поточного етапу розробки чи активного завдання.

Edit. Copy for Discourse: функція призначена для копіювання програмного коду у форматі, придатному для розміщення на форумах. *Copy as HTML:* для розміщення на веб-сторінках, копіює код скетча в буфер обміну як HTML код.

Sketch. Import Library: вставляє в робочий скетч бібліотеку, з допомогою директиви `#include`. *Show Sketch Folder:* відкриває на робочому столі папку, що містить файл скетча. *Add File...:* додає файл в скетч. Новий файл з'явиться в новій закладці у вікні скетча.

Tools. Auto Format: цей інструмент автоматично форматує текст програми, вирівнюючи відкриваючі та закриваючі фігурні дужки по вертикалі та встановлюючи належні відступи між ними для покращення структури. *Board:* вибір платформи. *Serial Port:* меню надає доступ до переліку реальних і віртуальних COM-портів системи. Дані в списку оновлюються в момент відкриття меню Tools, що забезпечує відображення лише підключених на цей час пристроїв. *Burn Bootloader:* за допомогою функцій цього розділу можна прошити завантажувач (Bootloader) у мікроконтролер Arduino. Ця процедура зазвичай не потрібна під час щоденного використання платформи, проте є

необхідною для підготовки нових чипів ATmega, що не мають попередньо встановленого ПЗ.

Блокнот (Sketchbook). Середовищем Arduino використовується принцип блокнота: місце для зберігання скетчів. Скетчі відкриваються через меню File > Sketchbook або клавішею Open на панелі інструментів. При першому запуску програми Arduino автоматично створює директорію для блокнота.

Закладки, файли і компіляція. Надають можливість працювати з декількома файлами (кожен відкривається в окремій закладці). Файли можуть бути стандартними Arduino (без розширення), файлами C (розширення *.c), файлами C++ (*.cpp) або файлами (.h).

Завантаження скетча в Arduino. Перед початком прошивки необхідно сконфігурувати параметри у вкладках Tools>Board та Tools>Serial Port. У середовищі Windows системні імена портів зазвичай мають вигляд COM1/COM2 для пристроїв із послідовним інтерфейсом або COM4 і вище для USB-платформ. Процес ініціюється натисканням кнопки завантаження або через меню File>Upload. Більшість сучасних моделей Arduino підтримують автоматичне скидання (reset) перед записом коду, тоді як застарілі модифікації вимагають ручного натискання кнопки перезавантаження. Візуальним підтвердженням передачі даних є мерехтіння індикаторів RX та TX. Запис здійснюється за допомогою Bootloader – компактної сервісної програми, що вже міститься в пам'яті контролера. Вона забезпечує передачу коду без зовнішнього програматора. Завантажувач активується на короткий час під час кожного старту плати, що супроводжується характерним блиманням світлодіода, підключеного до 13-го виводу.

Бібліотеки. Розширити можливості скетчів, зокрема для взаємодії з периферійними пристроями чи складних обчислень, можна за допомогою бібліотек. Їх підключення здійснюється через меню Sketch>Import Library, після чого на початку програмного коду автоматично додаються відповідні директиви #include. Слід враховувати, що використання бібліотек збільшує обсяг зайнятої пам'яті мікроконтролера, тому зайві модулі варто видаляти.

Базовий набір інструментів уже інтегрований у середовище Arduino IDE, проте додаткові бібліотеки доступні на офіційному сайті Arduino.cc та сторонніх ресурсах. Для ручного встановлення необхідно в кореневому каталозі скетчів створити теку `libraries` і розпакувати туди архів. Наприклад, файли бібліотеки `DateTime` повинні бути розташовані за шляхом: `/libraries/DateTime`.

Моніторинг послідовної шини (Serial Monitor). Цей інструмент слугує для моніторингу даних, які передаються на плату Arduino (через USB-порт або послідовний інтерфейс). Для надсилання інформації слід ввести текст у відповідне поле та натиснути `Enter` або кнопку `Send`. Важливо, щоб обрана у випадуючому списку швидкість передачі збігалася з параметром, указаним у функції `Serial.begin` скетчу.

Налаштування. Налаштування змінюють у вікні `Preferences`. Решта налаштувань містяться у файлі, місцезнаходження якого вказано в вікні `Preferences`.

Цифрові виводи. Контакти платформи Arduino характеризуються універсальністю: вони можуть функціонувати як у режимі введення, так і виведення сигналу. Зокрема, аналогові входи контролера ATmega можна програмно налаштувати для роботи як цифрові порти введення/виведення (I/O). За замовчуванням усі піни сконфігуровані на вхід, тому використання функції `pinMode()` для цього режиму не є обов'язковим. У стані введення порти мають високий вхідний опір (високоімпедансний стан), що мінімізує вплив на електричну схему та потребує незначного струму для перемикавання логічних рівнів. Проте за відсутності підключення на таких виводах можуть виникати випадкові значення через електромагнітні завади. Для стабілізації сигналу застосовують підтягувальні резистори до шини живлення (+5 В) або до «землі». Мікроконтролер ATmega оснащений внутрішніми резисторами опором 20 кОм, які активуються програмно:

```
pinMode(pin, INPUT); // встановлення режиму введення  
digitalWrite(pin, HIGH); // активація внутрішнього підтягуючого резистора
```

Варто враховувати, що максимальне навантаження на один вивід обмежене значенням 40 мА. Цього струму замало для безпосереднього керування силовими компонентами (реле, електродвигунами). Спроби підключення потужних споживачів або виникнення короткого замикання можуть призвести до виходу з ладу вихідних каскадів або всього чипа.

Широтно-імпульсна модуляція (ШИМ). Широтно-імпульсна модуляція (ШИМ) є методом імітації аналогового сигналу за допомогою дискретних цифрових засобів. Суть процесу полягає у швидкому перемиканні виходу між високим (5 В) та низьким (0 В) логічними рівнями. Результуюча напруга визначається «шириною» (тривалістю) активної фази імпульсу в межах одного циклу. Пропорційно змінюючи цей параметр, ми отримуємо середнє значення напруги, яке відповідає площі під графіком імпульсів. У середовищі Arduino для цього використовується функція `analogWrite()`. Параметр 255 забезпечує постійний сигнал 5 В (заповнення 100%), тоді як 127 формує сигнал 2,5 В, що відповідає робочому циклу 50%

Аналогові входи. В архітектуру мікроконтролерів ATmega, що використовуються в платах Arduino, інтегровано 6-канальний аналого-цифровий перетворювач (АЦП). Завдяки 10-бітній роздільній здатності, пристрій трансформує вхідну напругу в цифровий код у діапазоні від 0 до 1023 пунктів. Важливою особливістю є можливість експлуатації аналогових входів як звичайних цифрових портів. У такому режимі вони отримують ідентифікатори з 14 по 19. Приклад програмного налаштування:

```
pinMode(14, OUTPUT); // конфігурація як цифрового виходу
digitalWrite(14, HIGH); // встановлення високого логічного рівня
```

Структура програми на Arduino. Процес програмування Arduino базується на використанні фреймворку Wiring. Варто зазначити, що Wiring не є самостійною мовою в класичному розумінні й не має власного компілятора. Написаний користувачем код спочатку трансформується у повноцінну програму на C/C++, після чого обробляється стандартним компілятором AVR-

GCC. Таким чином, розробка для Arduino – це фактично робота з адаптованою під мікроконтролери AVR версією мов С та С++.

Апаратні платформи Arduino. Модельний ряд Arduino представлений різними модифікаціями, що відрізняються апаратною базою. Наприклад, плата Leonardo функціонує на базі чипа ATmega32u4. Популярна модель Uno, як і її попередниця Duemilanove, спроектовані на мікросхемі Atmel ATmega328. У старіших розробках, таких як Diecimila, застосовувався контролер ATmega168, а в найперших версіях платформ використовувався ATmega8. Найпотужніша модель лінійки – Arduino Mega2560 – базується на високопродуктивному мікроконтролері ATmega2560.

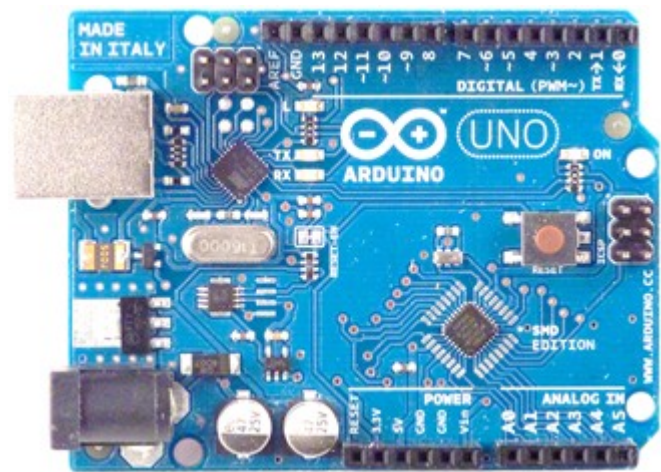


Рис. 2.3. Апаратна платформа Arduino UNO

Вибір платформи впливає на швидкість ЦП та швидкість передачі даних, і на налаштування запису загрузчика (Bootloader) мікроконтролера.

РОЗДІЛ 3

АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ПРИХОВАНОЇ ОХОРОННОЇ СИСТЕМИ, ЧУТЛИВОЇ ДО ВІБРАЦІЇ ГРУНТУ З ВИКОРИСТАННЯМ МЕМС- АКСЕЛЕРОМЕТРА

Елементною базою для проектування апаратно-програмної охоронної системи, чутливої до вібрації ґрунту було вибрано:

- апаратну платформу Arduino UNO з МК AVR (Atmel ATmega328);
- символічний LCD-дисплей 16x2. Модель: WH1602B-YGK-CTK;
- ADXL345 МЕМС акселерометр.

3.1. Опис апаратної платформи Arduino UNO з МК AVR ATmega328

Сучасний ринок пропонує широку номенклатуру мікроконтролерів від різних світових брендів. У ході розробки даного проекту було проведено порівняльний аналіз ключових критеріїв: вартості, обчислювальної потужності, енергоефективності та простоти інтеграції в електронні схеми. За результатами оцінки оптимальним рішенням було обрано платформу Arduino Uno, серцем якої є мікроконтролер ATmega328P від компанії Atmel. AVR представляє собою родину 8-бітних мікроконтролерів із RISC-архітектурою, розроблених компанією Atmel. Дані чипи є ефективним інструментом для побудови різноманітних вбудованих систем завдяки високій продуктивності та гнучкості налаштувань. Важливою перевагою є можливість внутрішньосхемного перепрограмування (до 10 тисяч циклів).

Виробник виділяє три основні серії архітектури AVR: Tiny AVR: малогабаритні та бюджетні рішення, що зазвичай випускаються у 8-контактних корпусах. Classic AVR: базова лінійка з обчислювальною потужністю до 16 MIPS. Об'єм FLASH-пам'яті варіюється в межах 2-8 Кб, EEPROM – до 512 байт, а SRAM складає від 128 до 512 байт. Mega AVR: потужні контролери (4-20 MIPS) для реалізації складних алгоритмів. Вони оснащені FLASH-пам'яттю до 128 Кб, оперативною пам'яттю (SRAM) до 4 Кб та інтегрованим 10-бітним АЦП на 8 каналів.

На Рис.3.1 продемонстровано зовнішній вигляд платформи Arduino UNO та конфігурацію сигналів мікроконтролера ATmega328 на її інтерфейсних роз'ємах.

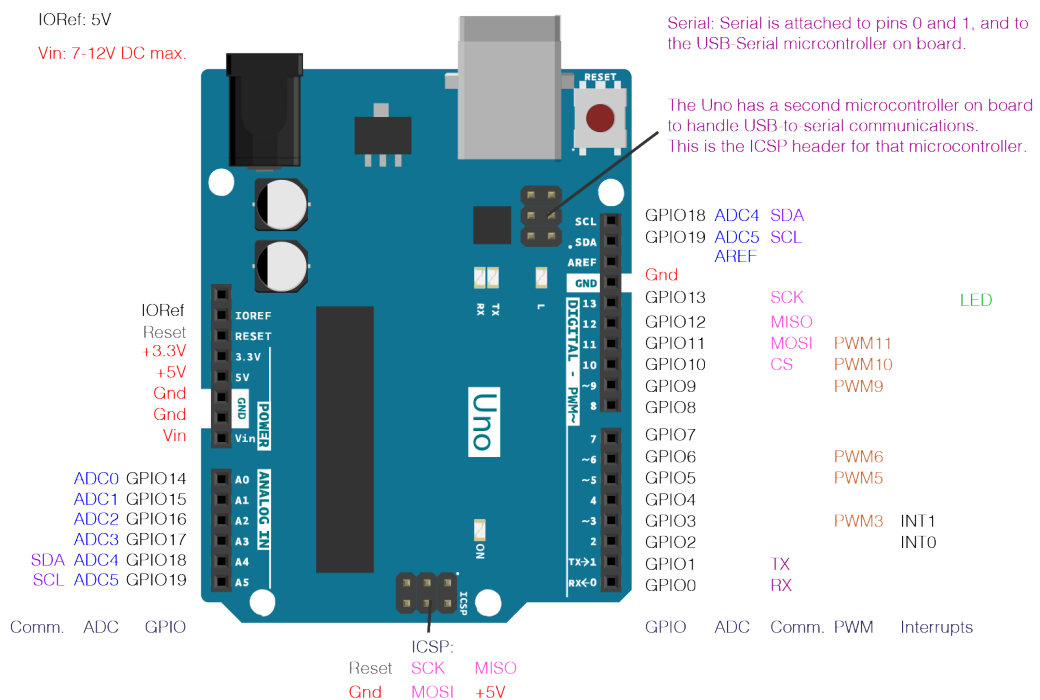


Рис. 3.1. Апаратна платформа Arduino UNO

ATmega328P – це високопродуктивний 8-бітний мікроконтролер архітектури AVR із низьким рівнем енергоспоживання та вбудованою FLASH-пам'яттю об'ємом 32 КБ. Завдяки вдосконаленій RISC-архітектурі пристрій забезпечує високу ефективність виконання операцій.

Основні обчислювальні можливості: Система команд: включає 131 інструкцію, більшість із яких реалізуються за один такт генератора. Регістрова пам'ять: містить 32 робочих реєстри загального призначення. Швидкодія: досягає 20 MIPS при робочій частоті 20 МГц. Архітектура: роздільна пам'ять для програм та даних (Гарвардська архітектура), повністю статичний режим функціонування.

Характеристики пам'яті: FLASH (програми): 32 КБ із ресурсом до 10 000 ітерацій перезапису. EEPROM (енергонезалежна): 1 КБ (до 100 000 циклів стирання/запису). SRAM (оперативна): 2 КБ внутрішньої пам'яті.

Периферійні модулі та інтерфейси: Три таймери/лічильники: два 8-бітних та один 16-бітний (із функціями захоплення та порівняння). Шість

каналів ШІМ-генерації. 10-розрядний АЦП на 8 вхідних каналів. Комунікаційні інтерфейси: послідовний USART, шина SPI та вбудований аналоговий компаратор. Спеціальні функції: сторожовий таймер (Watchdog) та годинник реального часу (RTC).

Енергоспоживання та захист: Шість режимів енергозбереження (зокрема Idle, Power-down, Standby). Системи захисту: детекція зниження напруги живлення (Brown-out Reset) та запуск після подачі живлення (Power-on Reset). Внутрішній калібрований RC-генератор.

Фізичні параметри: 32 програмовані лінії введення-виведення (I/O). Напруга живлення: у діапазоні 2,7 – 5,5 В. Тип корпусу: 28-вивідний PDIP або компактний 32-вивідний TQFP.

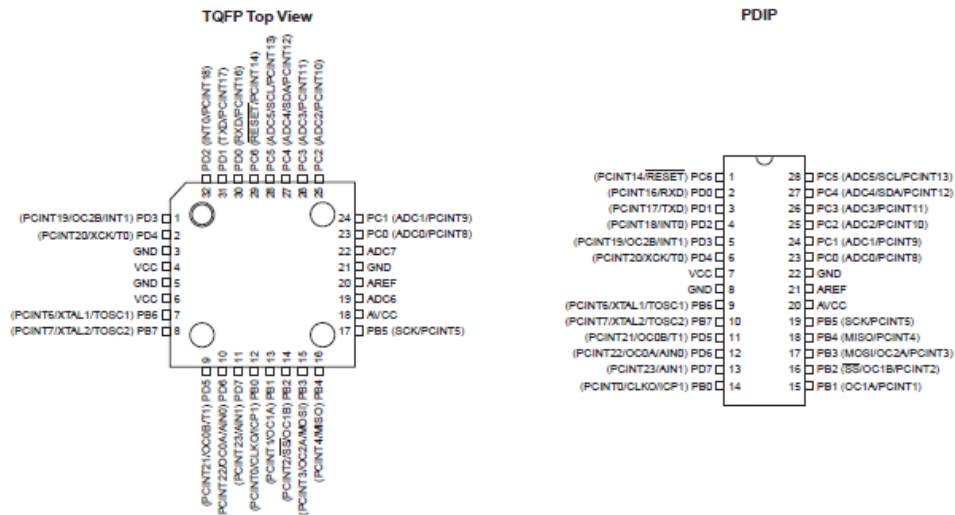


Рис. 3.2. Позначення виводів в корпусах PDIP і TQFP МК AVR ATmega328P

Основою обчислювальної потужності мікроконтролера є AVR-ядро, що використовує розширену систему команд у поєднанні з 32 робочими регістрами загального призначення. Пряме з'єднання всіх регістрів з арифметико-логічним пристроєм (АЛП) дозволяє маніпулювати двома незалежними регістрами протягом одного такту. Така архітектурна організація забезпечує на порядок вищу продуктивність порівняно з традиційними CISC-мікроконтролерами. Апаратне наповнення ATmega328P включає: Пам'ять: 32 КБ FLASH із підтримкою технології Read-While-Write, 2 КБ оперативної пам'яті (SRAM) та 1024 байти енергонезалежної пам'яті даних (EEPROM).

Контроль та налагодження: інтерфейс JTAG для сканування регістрів та вбудовані інструменти дебагінгу. Таймерні модулі: три функціональні таймери-лічильники, що підтримують режими порівняння. Комунікації: програмований USART, послідовний інтерфейс SPI та двопровідний інтерфейс (I2C/TWI). Обробка сигналів: 8-канальний 10-бітовий АЦП та система зовнішніх і внутрішніх переривань. Захист та енергозбереження: сторожовий таймер (Watchdog) з автономним осцилятором та шість режимів керування живленням. Зокрема, у режимі Idle робота центрального процесора зупиняється для зниження енергоспоживання.

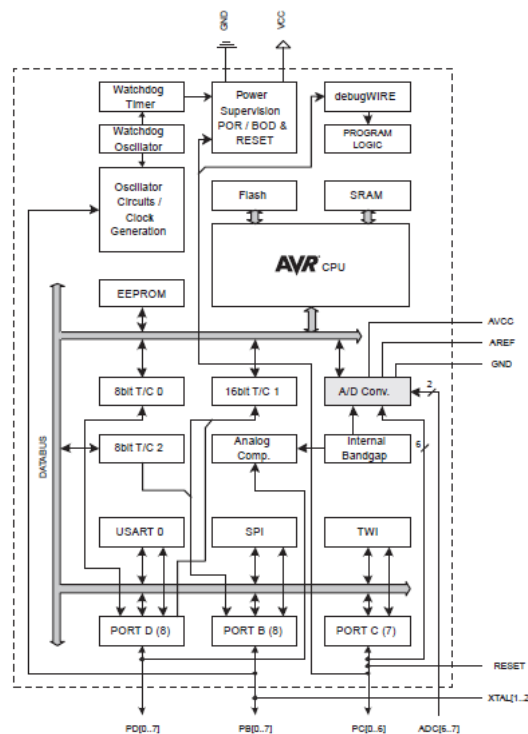


Рис. 3.3. Блок-схема МК AVR ATmega328P

Для підвищення точності вимірювань у ATmega328P передбачено спеціальний режим зниження завад АЦП. У цьому стані зупиняється більшість блоків, крім самого перетворювача та асинхронного таймера, що мінімізує вплив цифрових шумів на результат. Режими енергозбереження та програмування: Режим Power-down (вимкнення): Центральний процесор зберігає дані в регістрах, але зупиняє тактовий генератор та всі основні функції. Вихід із цього стану можливий лише за сигналом зовнішнього переривання або скидання (Reset). Режим Standby (очікування): Активним

залишається лише основний осцилятор, що забезпечує миттєве повернення мікроконтролера до робочого стану. У розширеному режимі очікування додатково функціонує асинхронний генератор. Внутрішньосистемне програмування: наявність ISP FLASH пам'яті дозволяє оновлювати програмний код через інтерфейс SPI за допомогою стандартних програматорів або вбудованого завантажувача (Bootloader).

Висока адаптивність ATmega328P зумовлена поєднанням гнучкої RISC-архітектури та можливості швидкого перемикання між режимами споживання енергії, що робить цей чип надзвичайно ефективним для вбудованих систем.

Аналого-цифровий перетворювач МК ATmega328. Архітектура мікроконтролера передбачає наявність портів введення-виведення, які зазвичай оперують виключно дискретними сигналами – логічним “нулем” або “одиницею”. Наприклад, для чипа ATmega328P при п'ятивольтовому живленні рівень логічного нуля відповідає напрузі від 0 до 1,5 В, тоді як логічна одиниця охоплює діапазон від 1,9 до 5 В. Проте для реєстрації аналогових сигналів, що можуть плавно змінюватися в межах напруги живлення, у мікроконтролерах AVR інтегровано спеціальний модуль — аналого-цифровий перетворювач (АЦП).

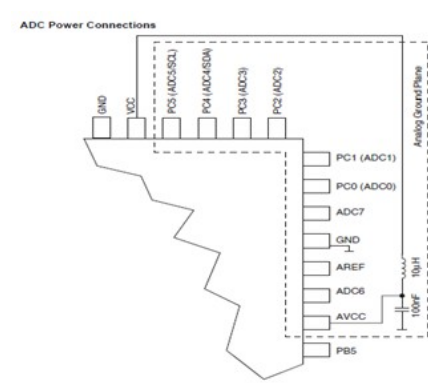


Рис. 3.4. Виводи АЦП в МК ATmega328P

У мікроконтролері ATmega328 модуль АЦП можна комутувати з будь-яким виводом порту С (PORTC). Для конфігурування та керування роботою інтегрованого перетворювача застосовуються спеціалізовані регістри:

- ADMUX – регістр управління мультиплексором;
- ADCH – старший байт результату перетворення;

- ADCL – молодший байт результату перетворення;
- SFIOR – регістр спеціальних функцій;
- ADCSRA – регістр А управління та стану.

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Таблиця 3.1. Вибір опорного джерела ADC

REFS1	REFS0	Джерело опорної напруги ADC
0	0	Зовнішнє джерело, підключено до AREF, внутрішнє VREF відключено
0	1	AVCC із зовнішнім конденсатором на виводі AREF.
1	0	Зарезервовано
1	1	Внутрішнє джерело опорної напруги 1,1В із зовнішнім конденсатором на виводі AREF.

Діапазон вхідних значень АЦП визначається рівнем опорної напруги (V_{ref}). У випадку, коли амплітуда вимірюваного сигналу перевищує опорний рівень, результат перетворення набуває максимального значення – 0x3FF. Система дозволяє обирати джерело опорної напруги серед трьох варіантів: напруга живлення аналогової частини (AVCC), внутрішній стабілізований рівень 1,1 В або зовнішній сигнал на виводі AREF. Для мінімізації впливу електромагнітних завад рекомендується шунтувати вивід AREF на “землю” за допомогою конденсатора.

Отриманий 10-бітовий код зберігається у парі 8-бітних регістрів – ADCH та ADCL. Програмно можна обрати спосіб вирівнювання результату (справа за замовчуванням або зліва через біт ADLAR). Критично важливим є порядок зчитування: звернення до ADCL блокує оновлення даних в обох регістрах до моменту зчитування ADCH, що гарантує цілісність отриманого значення.

Вибір активного входу (від ADC0 до ADC7) або диференціальної пари здійснюється через біти MUX у регістрі ADMUX. Для забезпечення заявленої 10-бітної точності частота тактування АЦП, що встановлюється вбудованим

подільником (ADPS), повинна знаходитися в межах від 50 до 200 кГц. Вищі частоти дозволяють прискорити процес за рахунок зниження розрізняювальної здатності. Стандартний цикл перетворення триває 13 тактів АЦП, окрім першого запуску після активації модуля (ADEN), який потребує 25 тактів для стабілізації аналогових кіл. Завершення процесу супроводжується встановленням прапорця ADIF. Система підтримує як поодинокі вимірювання, так і режим безперервного сканування (Free Running mode). Для простого перетворення:

$$ADC = V_{in} \cdot 1024 / V_{REF} \quad (3.1)$$

де V_{REF} – опорна напруга, V_{in} – напруга, яку вимірюємо.

3.2. Опис трьохосового MEMS-акселерометра ADXL345

Акселерометр представляє собою вимірювальний прилад, призначений для визначення прискорення. Фізична величина прискорення відображає динаміку зміни швидкості об'єкта за одиницю часу і виражається в метрах на секунду в квадраті (m/s^2) або у відносних одиницях g. На поверхні Землі стандартне значення прискорення вільного падіння становить приблизно $9,8 m/s^2$. Основними сферами застосування цих датчиків є моніторинг вібраційних процесів та визначення просторової орієнтації пристроїв.

Технологія MEMS (мікроелектромеханічні системи) дозволяє створювати акселерометри, здатні реєструвати як статичні (гравітаційні), так і динамічні (ударні навантаження, коливання) сили. На сучасному ринку напівпровідникових компонентів представлено велику кількість MEMS-сенсорів, що класифікуються за кількістю осей вимірювання (одно-, дво- та трьохосові). Найбільш універсальними є трьохосові моделі, які здатні фіксувати вектори прискорення в декартовій системі координат (XYZ).

Відповідно до технічних даних виробника Analog Devices (Табл.3.2), при виборі моделі сенсора ключову роль відіграють такі параметри: діапазон вимірюваних прискорень, напруга живлення, енергоспоживання, роздільна здатність (чутливість), тип комунікаційного інтерфейсу та ціна.

Таблиця 3.2. Технічні параметри акселерометрів та їх ціна

Part#	Accelerometer Range	I _s (typ) (A)	V _{s+} (min) (V)	Temp Range	BW (typ) (Hz)	Output Type	Sensing Axis	# of Axes	Price (1000+) (\$ US)
ADXL1005	100 g	1m	3	-40 to 125°C	23k	Analog	X	1	\$39.41 (ADXL1005BCPZ)
ADXL1004	500 g	1m	3.3	-40 to 125°C	24k	Analog	X	1	\$35.85 (ADXL1004BCPZ)
ADXL372	200 g	22μ	1.6	-40 to 105°C	3.2k	PC, SPI	X, Y, Z	3	\$6.33 (ADXL372BCCZ-RL7)
ADXL357	10 g, 20 g, 40 g	200μ	2.25	-40 to 125°C	1k	PC, SPI	X, Y, Z	3	\$31.87 (ADXL357BEZ)
ADXL356	10 g, 20 g, 40 g	150μ	2.25	-55 to 125°C	1.5k	Analog	X, Y, Z	3	\$28.68 (ADXL356BEZ)
ADXL251	120 g, 240 g, 480 g, 60 g	6m	3.135	-	400	PSI5, SPI	X, Y	2	-
ADXL1002	50 g	1m	3.3	-40 to 125°C	11k	Analog	X	1	\$29.66 (ADXL1002BCPZ)
ADXL1001	100 g	1m	3.3	-40 to 125°C	11k	Analog	X	1	\$29.66 (ADXL1001BCPZ)
ADXL355	2 g, 4 g, 8 g	200μ	2.25	-40 to 125°C	1k	PC, SPI	X, Y, Z	3	\$28.33 (ADXL355BEZ)
ADXL354	2 g, 4 g, 8 g	150μ	2.25	-40 to 125°C	1.5k	Analog	X, Y, Z	3	\$25.50 (ADXL354BEZ)
ADXL151	120 g, 240 g, 480 g	5m	4.5	-	400	PSI5	X	1	-
ADXL316	16 g	350μ	1.8	-40 to 105°C	1.6k	Analog	X, Y, Z	3	\$3.76 (ADXL316WBCSZ)
ADXL700	14.2	5m	3.135	-	-	SPI	X, Y, Z	3	-
ADXL363	2 g, 4 g, 8 g	1.8μ	1.6	-40 to 85°C	200	SPI	X, Y, Z	3	\$4.57 (ADXL363BCCZ-RL7)

З таблиці видно ADXL1004, ADXL1005, ADXL151 – це одноосьові акселерометри, ADXL295, ADXL251 ADXL288 – це двохосьові та ADXL357, ADXL355, ADXL345 – трьохосьові. Найдешевший з трьохосьових акселерометрів є ADXL345, що має задовільні технічні параметри для реалізації цього проекту. На Рис.3.5 зображено трьохосьовий MEMC акселерометр ADXL345.

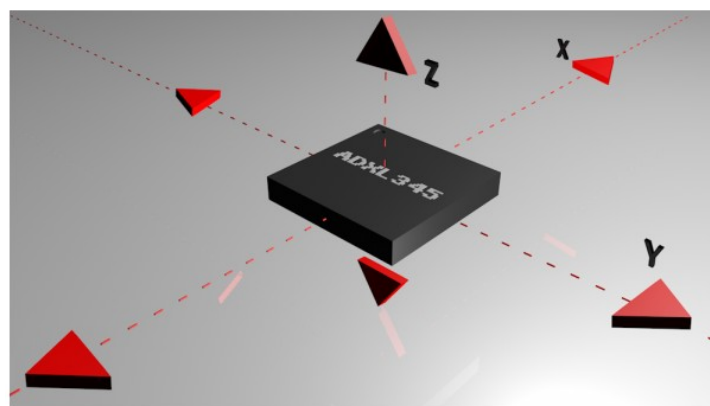


Рис. 3.5. Осі вимірювання акселерометра ADXL345

Сучасні акселерометри класифікуються за кількістю осей вимірювання (одна, дві або три). Завдяки поступовому зниженню вартості виробництва, трьохосьові моделі стали найбільш розповсюдженим стандартом у мікроелектроніці.

В основі роботи MEMC-акселерометра лежить система мікроскопічних конденсаторних пластин. Конструкція передбачає наявність нерухомих

елементів та рухомої маси, підвішеної на пружинах. Під дією сил прискорення рухома частина зміщується, що спричиняє зміну електричної ємності між пластинами. Вимірюючи ці коливання ємності, сенсор обчислює вектор та амплітуду прискорення.

Більшість цифрових сенсорів дозволяють програмно налаштувати діапазон вимірювань — від прецизійних $\pm 2g$ до значних $\pm 250g$. Варто враховувати зворотну залежність: чим вузьчий обраний діапазон, тим вищою буде роздільна здатність (чутливість) датчика. Наприклад, для детектування ледь помітних вібрацій поверхні доцільно обирати мінімальний діапазон, тоді як шкала у $250g$ використовується в аерокосмічній галузі для моніторингу прискорень ракетної техніки. Архітектурні особливості типового трьохосьового датчика розглянуто на прикладі моделі ADXL345, функціональну схему якої наведено на Рис.3.6.

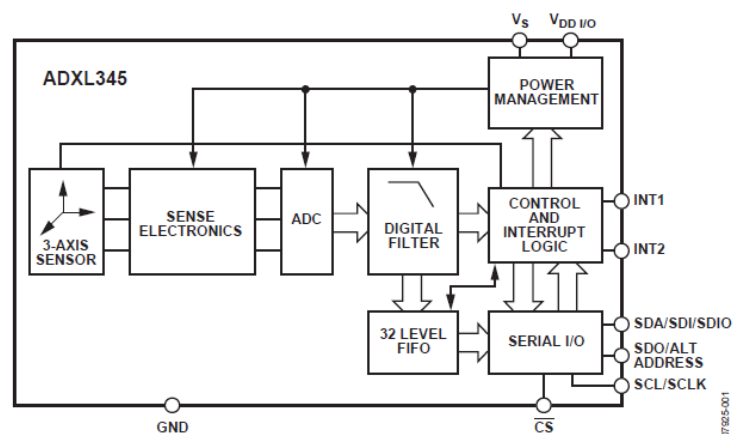


Рис. 3.6. Функціональна схема акселерометра ADXL-345

ADXL345 – це невеликий 3-х осьовий акселерометр з високою роздільною здатністю (13 біт) і з діапазоном вимірювання прискорень до $\pm 16g$. Діапазон вимірювання можна вибрати з ряду: $\pm 2g$, $\pm 4g$, $\pm 8g$ і $\pm 16g$. Результат вимірювання можна отримати по інтерфейсу SPI або I2C у вигляді 16-ти біт даних. ADXL345 відноситься до ємнісних MEMS акселерометрів, має вузьку полосу пропускання (0,05...1600 Гц), і добре зарекомендував себе для вимірювання низкочастотних вібрацій та статичного прискорення.

При виборі акселерометра треба знати параметри прискорення, наприклад, для удару -діапазон прискорень і тривалість їх дії. При

випробуваннях промислових конструкцій досліджують реакцію конструкції на удар: резонансну частоту, згасання коливань. При виборі акселерометра треба орієнтуватися на частотний діапазон прискорень і їх величину.

Акселерометри з широким діапазоном частот можуть вимірювати малі прискорення. Чіпи високочастотних MEMS-акселерометрів $\pm 70g$, $\pm 250g$ і $\pm 500g$, з діапазоном частот 0..30кГц дорогі і дефіцитні. При потребі вимірювання прискорення в 20g...500g, і невисокій частоті (до 400Гц), відповідні чіпи MEMS-акселерометров в десятки раз дешевші. Для мікроконтролерної охоронної системи, чутливої до вібрації ґрунту підійде MEMS акселерометра ADXL345.

Технічні характеристики трьохосового акселерометра ADXL345:

- Електричні характеристики: робоча напруга пристрою становить 2,0-3,6 В. Модуль вирізняється високою енергоефективністю: споживання варіюється від 25 до 130 мкА під час активної роботи та знижується до 0,1 мкА у стані спокою.
- Сенсор підтримує роботу у фіксованому 10-бітному режимі або в режимі повної роздільної здатності (до 13 біт при $\pm 16g$), що дозволяє зберігати сталий крок вимірювання (4 mg/LSB) незалежно від обраної шкали.
- Апаратна підтримка розпізнавання поодиноких і подвійних ударів, а також детектування стану вільного падіння.
- Обмін даними здійснюється через протоколи I2C або SPI (підтримуються 3-х та 4-провідні схеми підключення).
- Передбачено два незалежні цифрові виводи з можливістю гнучкого призначення подій переривання.
- Користувач може програмно задавати як діапазон чутливості, так і смугу пропускання фільтра.
- Пристрій стабільно функціонує в температурному інтервалі $-40...+85^{\circ}\text{C}$ та витримує екстремальні механічні перевантаження до 10 000g.

Зовнішній вигляд мікросхеми ADXL345 представлено на Рис.3.8.

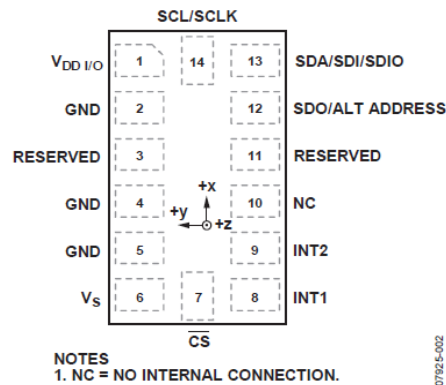


Рис. 3.8. MEMS-акселерометр ADXL345

Таблиця 3.3. Призначення виводів ADXL345

Вивід	Позначення	Опис
1	V _{DD I/O}	Напруга живлення цифрового інтерфейсу
2	GND	Земля
3	RESERVED	Зарезервовано. Під'єднати до V _s або залишити не під'єднаним
4	GND	Земля
5	GND	Земля
6	V _s	Напруга живлення
7	CS	Вибір мікросхеми (лог.0)
8	INT1	Вихід переривання 1
9	INT2	Вихід переривання 2
10	NC	Не використовується (залишити не під'єднаним)
11	RESERVED	Зарезервовано. Під'єднати до GND, або залишити не під'єднаним
12	SDO/ALT ADDRESS	Вихід послідовної передачі даних(SPI 4-wire) Вибір адреса I2C
13	SDA/SDI/SDIO	Послідовна лінія даних (I2C)/Вхід прийому даних (SPI)
14	SCL/SCLK	Послідовна лінія тактування. SCL для I2C і SCLK для SPI

Акселерометр ADXL345 має 31 регістр – призначення яких наведено в Табл. 3.4.

Таблиця 3.4. Регістри ADXL345

Адрес Hex	Адрес Dec	Назва регістра ADXL345	Тип	Значення По- замовчуванню	Призначення
0X00	0	DEVID	R	11100101	ID – пристрою
0X01 ..0X1C	1..28	Reserved	R/W	00000000	Зарезервовано
0X1D	29	THRESH_TAP	R/W	00000000	Чутливість до легкого удару
0X1E	30	OFSX	R/W	00000000	Зміщення осі X
0X1F	31	OFSY	R/W	00000000	Зміщення осі Y
0X20	32	OFSZ	R/W	00000000	Зміщення осі Z
0X21	33	DUR	R/W	00000000	Тривалість поштовху
0X22	34	Latent	R/W	00000000	Затримка поштовху
0X23	35	Window	R/W	00000000	Тимчасове вікно для другого поштовху
0X24	36	THRESH_ACT	R/W	00000000	Поріг активності
0X25	37	THRESH_INACT	R/W	00000000	Поріг неактивності
0X26	38	TIME_INACT	R/W	00000000	Час неактивності
0X27	39	ACT_INACT_CTL	R/W	00000000	Управління осями для визначення активності
0X28	40	THRESH_FF	R/W	00000000	Поріг вільного падіння
0X29	41	TIME_FF	R/W	00000000	Час вільного падіння

Адрес Hex	Адрес Dec	Назва регістра ADXL345	Тип	Значення По- замовчуванню	Призначення
0X2A	42	TAP_AXES	R/W	00000000	Управління осями для поштовху/подвійного
0X2B	43	ACT_TAP_STATUS	R	00000000	Джерело поштовху/ подвійного поштовху
0X2C	44	BW_RATE	R/W	00001010	Управління частотою виборки і режимами живлення
0X2D	45	POWER_CTL	R/W	00000000	Управління енергозбереженням
0X2E	46	INT_ENABLE	R/W	00000000	Управління дозволом переривань
0X2F	47	INT_MAP	R/W	00000000	Призначення виводів переривання
0X30	48	INT_SOURCE	R	00000010	Джерело переривань
0X31	49	DATA_FORMAT	R/W	00000000	Налаштування формату даних
0X32	50	DATA_X0	R	00000000	X - вісь DATA 0
0X33	51	DATA_X1	R	00000000	X - вісь DATA 1
0X34	52	DATA_Y0	R	00000000	Y - вісь DATA 0
0X35	53	DATA_Y1	R	00000000	Y - вісь DATA 1
0X36	54	DATA_Z0	R	00000000	Z - вісь DATA 0
0X37	55	DATA_Z1	R	00000000	Z - вісь DATA 1
0X38	56	FIFO_CTL	R/W	00000000	Управління FIFO
0X39	57	FIFO_STATUS	R	00000000	Статус FIFO

Процедура підключення давача наступна. При включенні живлення акселерометр переходить в режим очікування (standby mode, рекомендується скинути біт вимірювання (біт D3) в регістрі POWER_CTL) і очікує, команди на вимірювання (measurement mode) для цього потрібно встановити біт D3 в регістрі POWER_CTL. Скидання цього біта вимірювання повертає акселерометр в режим очікування. ADXL345 споживає енергію пропорційно частоті дискретизації внутрішнього АЦП (Табл.3.5).

Таблиця 3.5. Струм споживання ADXL345

Частота дискретизації (Гц)	Полоса пропускання (Гц)	Код	Струм Споживання (µA)
3200	1600	1111	140
1600	800	1110	140
800	400	1101	140
400	200	1100	140
200	100	1011	140
100	50	1010	140
50	25	1001	90
25	12,5	1000	60
12,5	6,25	0111	50
6,25	3,13	0110	45
3,13	1,56	0101	40
1,56	0,78	0100	34
0,78	0,39	0011	23
0,39	0,20	0010	23
0,20	0,10	0001	23
0,10	0,05	0000	23

Датчик ADXL345 підтримує два типи послідовних інтерфейсів – I2C та SPI, функціонуючи при цьому виключно як відомий пристрій (Slave). Вибір конкретного протоколу передачі даних залежить від стану виводу CS (Chip Select). Для активації I2C необхідно забезпечити на вході CS високий логічний рівень (підключити до напруги живлення “лог.1” VDD I/O). Для роботи через SPI: лінія CS має контролюватися безпосередньо ведучим пристроєм (Master).

Вибір частоти дискретизації безпосередньо корелює з пропускнуою здатністю обраного інтерфейсу. Частоти 3200 Гц і 1600 Гц рекомендовані для роботи по SPI лінії, з пропускнуою здатністю рівній або більше 2 МГц. Частота виборки 800 Гц для швидкості передачі 400 кГц, а решта значень можна визначити пропорційно. ADXL345 підтримує стандартний Standard Mode (до 100 кГц) та Fast Mode (до 400 кГц). режими обміну даних. Мережева адреса пристрою залежить від логічного стану виводу ALT ADDRESS. Алгоритм визначення адреси базується на 7-бітному ідентифікаторі, до якого додається біт напрямку передачі даних (R/W): R/W=0 – операція запису, R/W=1 – операція читання.

Якщо вивід ALT ADDRESS під’єднано до “лог. 1”, 7-разрядним адресом пристрою на шині I2C буде 0x1D. Разом з бітом R/W (читання – “1”, запису – “0”), даний адрес для операції запису буде рівний 0x3A, а для операції читання – 0x3B. Альтернативний адрес 0x53 отримуємо з’єднанням піна ALT ADDRESS на “землю”, а з врахуванням наступного за цим адресом біта R/W, адреса для операцій запису та читання будуть 0xA6 і 0xA7 відповідно. При роботі по інтерфейсу I2C виводи CS і ALT ADDRESS треба підключити на землю, або до VDD I/O!, Рис.3.9.

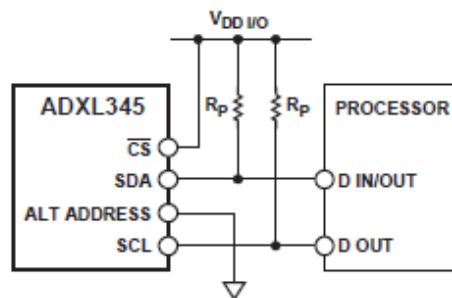


Рис. 3.9. Схема з’єднань ADXL345 для роботи по I2C

ADXL345 має два виходи переривання – INT1 і INT2, які можна програмувати. Доступними є функції переривання: DATA_READY - (наявність даних), SINGLE_TAP – (поштовх), ACTIVITY (активність), DOUBLE_TAP (подвійний поштовх), INACTIVITY (відсутня активність), WATERMARK, і OVERRUN (переповнення), FREE_FALL (вільне падіння). Виводи є двухтактними, кожен з яких працює за двотактною (push-pull) схемою. Гнучкість налаштувань дозволяє використовувати всі доступні функції переривань як окремо, так і в комбінації. Конфігурація виходів здійснюється через спеціальні регістри. Стандартно активним станом вважається високий рівень (“лог. 1”). Для інверсії сигналу (переходу на активний низький рівень) необхідно встановити прапор INT_INVERT у регістрі DATA_FORMAT. Переривання дозволяємо шляхом встановлення відповідного біта в регістрі INT_ENABLE, а розподіл подій між фізичними виводами INT1 та INT2 визначається параметрами регістра INT_MAP. На час читання регістрів даних виклик переривань призупиняється.

Призначення бітів регістра INT_MAP:

- Біт DATA_READY встановлюється якщо є нові дані і скидається якщо дані відсутні.
- Біт FREE_FALL встановлюється, якщо по всіх осях акселерометра (логічне AND) протягом деякого часу (що перевищує значення вказане в регістрі THRESH_FF).
- Біт WATERMARK (Водяний знак) встановлюється, якщо число виборок (результатів вимірів) у FIFO дорівнює вмісту бітів виборок в регістрі FIFO_CTL. Біт WATERMARK скидається автоматично, як FIFO прочитане.
- Біт OVERRUN (Переповнення) встановлюється, коли нові дані змінюють непрочитані дані. Операція функції переповнення залежить від режиму FIFO. Біт OVERRUN автоматично очищується після читання вмісту FIFO.

Для дослідження, яка саме функція визвала переривання треба читати регістр INT_SOURCE. Для оптимізації роботи керуючого мікроконтролера та зниження навантаження на системну шину, акселерометр оснащений вбудованим 32-рівневим буфером типу FIFO (First In – First Out). Цей стек пам'яті дозволяє накопичувати результати вимірювань по осях X, Y та Z перед їх зчитуванням. Користувачеві доступні чотири варіанти роботи пам'яті, які обираються за допомогою конфігурації бітів FIFO_MODE (позиції D7 та D6) у регістрі керування FIFO_CTL:

1. Bypass (Обхід): Буфер не використовується, дані передаються безпосередньо.
2. FIFO: Накопичення до повного заповнення (32 значення), після чого запис припиняється.
3. Stream (Потік): Безперервний запис, де нові дані заміщують найстаріші за принципом кільцевого буфера.
4. Trigger (Тригер): Режим, що зберігає останні дані перед виникненням певної події (переривання).

Повинна бути затримка не менше 5 мкс між подією і початком зчитування даних з FIFO, що дозволяє FIFO зберегти потрібні результати вимірів. Для отримання результатів вимірювань необхідно прочитати підряд шість регістрів з даними вимірювань вздовж осей X, Y і Z – а це, відповідно, регістри DATA_X0, DATA_X1, DATA_Y0, DATA_Y1, DATA_Z0 і DATA_Z1, починати читати з регістра DATA_X0 (адреса 0x32). Діапазон вимірювання залежить від вмісту регістра DATA_FORMAT. Оптимальним режимом роботи акселерометра є 13-бітний (встановити в регістрі DATA_FORMAT біт D3). В цьому режимі цьому точність вимірювання завжди 3,9mg/LSB. Це важливо, у випадках, коли важко передбачити які максимальні значення прискорення слід очікувати. Виробники випускаються готові модулі з мікросхемою акселерометра ADXL345 на платі та інтерфейсом для його підключення Рис.3.10, 3.11.

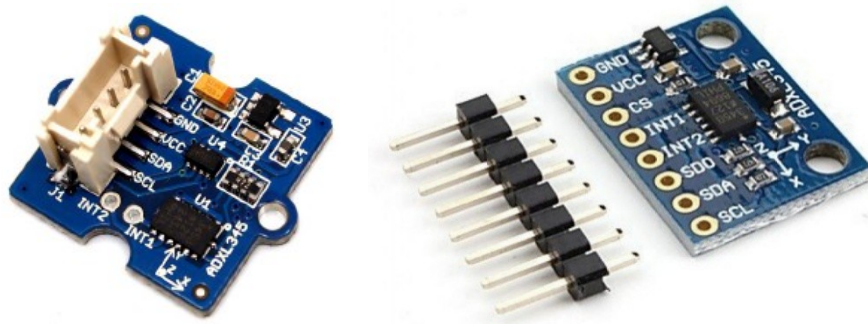


Рис. 3.10. Модулі з ADXL345 акселерометром

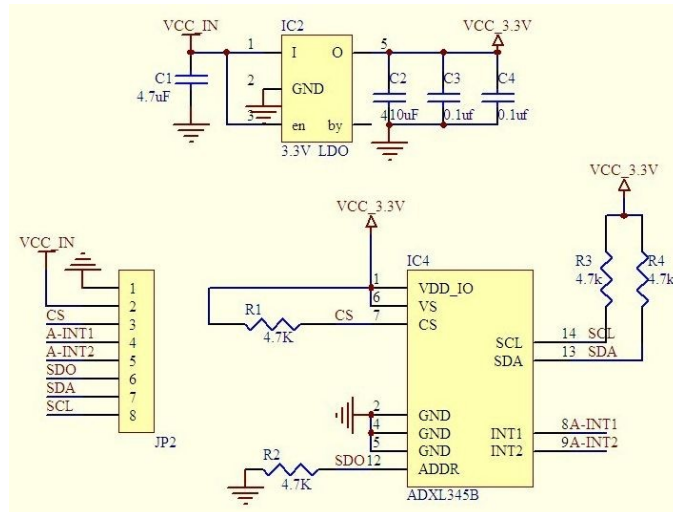


Рис. 3.11. Схема модуля з ADXL345

3.3. Опис та вивід значень на рідкокристалічний дисплей

Використання рідкокристалічних дисплеїв (РКД) у сучасних розробках є економічно вигідним та ефективним способом візуалізації даних. Такі модулі дозволяють суттєво скоротити терміни проектування, забезпечуючи при цьому вивід значного обсягу інформації за умови мінімальних енерговитрат. РК-індикатори адаптовані до роботи в різноманітних умовах завдяки наступним властивостям:

- Наявність вбудованого світлодіодного підсвічування гарантує чітке зчитування символів навіть у повній темряві.
- Розширений робочий діапазон температур (від -20°C до $+70^{\circ}\text{C}$) дозволяє інтегрувати модулі в бортову, польову та портативну апаратуру, що працює у складних режимах.

- Кожен модуль є завершеним пристроєм, що складається з рідкокристалічної панелі та спеціалізованого керуючого контролера.

Одним із найбільш розповсюджених варіантів є дворядковий модуль (Рис. 3.12.), здатний відображати по 16 знаків у кожному рядку. Його ключові характеристики:

1. Формат символу: Кожна літера або цифра формується матрицею розміром 5x8 точок.
2. Розподільча здатність: Для забезпечення кращої читабельності між окремими символами передбачені інтервали завширшки в один піксель.



Рис. 3.12. Символьний РКД 16x2. Модель: WH1602B-YGK-СТК

Дисплей WH1602B-YGK-СТК – це алфавітно-цифровий пристрій виводу, що формує темні символи на жовто-зеленому фоні. Модуль побудований на базі галузевого стандарту – контролера HD44780 (або його повного аналога), що забезпечує високу сумісність із більшістю мікроконтролерних систем. Кожному символу відповідає його код в комірці ОЗП контролера HD44780. Модуль містить два види пам'яті – кодів символів і знакогенератора користувача, а також логіку для управління РК панеллю.

Функціональні можливості контролера. Вбудована логіка керування забезпечує широкий спектр операцій:

1. Гнучкість інтерфейсу: Підтримка обміну даними як через 4-бітну, так і через 8-бітну шину (налаштовується програмно при ініціалізації).
2. Двомовність: Наявність двох сторінок вбудованого знакогенератора, що включають кириличний та латинський алфавіти.
3. Персоніфікація: Можливість створення та збереження в ОЗП до 8 унікальних символів користувача.
4. Керування курсором: Доступні режими статичного або миготливого курсора двох різних типів.

5. Двосторонній обмін: Окрім запису команд та даних, система дозволяє зчитувати вміст пам'яті та перевіряти статус готовності модуля (Busy Flag).

Внутрішня архітектура та логіка управління. Взаємодія з дисплеєм здійснюється через два ключові регістри:

- Регістр команд (IR): Для передачі інструкцій (очищення екрана, налаштування курсора тощо). Активується при низькому рівні на лінії RS ($RS=0$).
- Регістр даних (DR): Для запису або читання кодів символів. Активується при високому рівні RS ($RS=1$).

Організація пам'яті. Система керування використовує два типи пам'яті:

- CGRAM (Character Generator RAM): Служить для зберігання описів нестандартних символів.
- DDRAM (Display Data RAM): Відеопам'ять об'ємом 80 байтів. Архітектурно вона розділена на два рядки по 40 комірок у кожному.

Незалежно від фізичної конфігурації конкретної моделі дисплея (у нашому випадку 16x2), адресація пам'яті контролера завжди відповідає двом логічним рядкам по 40 символів. Це означає, що частина пам'яті може використовуватися для зсуву тексту або зберігання даних, які не відображаються на екрані одночасно.

Підключення до плати Arduino UNO модуля РКД. Для роботи з РКД використовуються функції інтерфейсу між програмами на С і РКД модулями, побудованими на мікросхемах Hitachi HD44780 або подібних. Arduino IDE має готові бібліотеки для роботи з такими РКД – модулями і підтримує формати РКД: 1x8, 2x12, 3x12, 1x16, 2x16, 2x20, 4x20, 2x24 и 2x40 символів. Схема підключення до плати Arduino UNO модуля РКД на Рис.3.13.

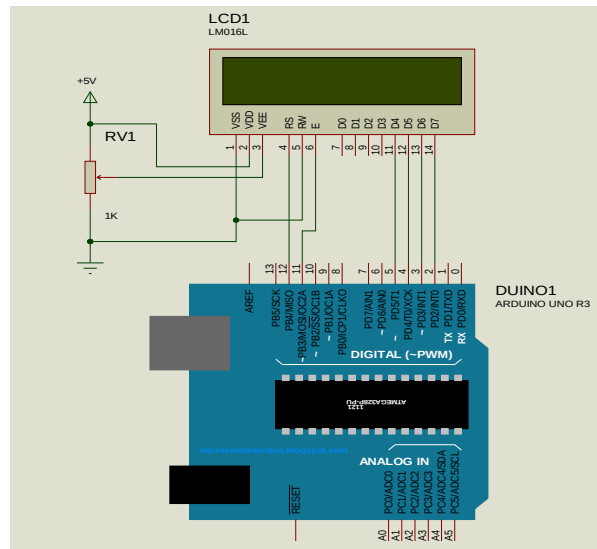


Рис. 3.13. Підключення модуля РКД до плати Arduino UNO

3.4. Проектування апаратно-програмної охоронної системи, чутливої до вібрації ґрунту в САПР Proteus VSM

Проектований комплекс апаратно-програмної охоронної системи, чутливої до вібрації ґрунту повинен вимірювати рівень вібрації ґрунту, аналізувати отримані дані, передавати їх по послідовному порту в ПК, та сигналізувати про перевищення рівнів вібраційної активності включенням відповідної звукової та візуальної сигналізації. На Рис.3.14 зображено запропоновану структуру апаратно-програмної охоронної системи, чутливої до вібрації ґрунту.

Апаратний склад комплексу. Структурно система базується на поєднанні обчислювальної потужності мікроконтролера та високочутливих сенсорів. До складу апаратної частини входять:

1. Прецизійний MEMС-акселерометр ADXL345, що виступає первинним перетворювачем вібрацій.
2. Плата Arduino Uno, яка відповідає за логіку роботи та взаємодію з периферією.
3. Рідкокристалічний модуль для візуалізації поточного стану та налаштувань.
4. Модулі звукової (п'єзовипромінювач) та світлової сигналізації.

5. Органи керування: Функціональні клавіші КН2 “UP” та КН1 “DOWN”, які дозволяють користувачеві вручну регулювати рівень чутливості системи до величини вібрацій ґрунту.

Система живиться від акумулятора 12 Вольт. Центральним вузлом обробки інформації в розробці є платформа Arduino Uno, побудована на базі МК ATmega328. Вона виконує роль головного керуючого пристрою, який забезпечує злагоджену роботу всіх компонентів системи.

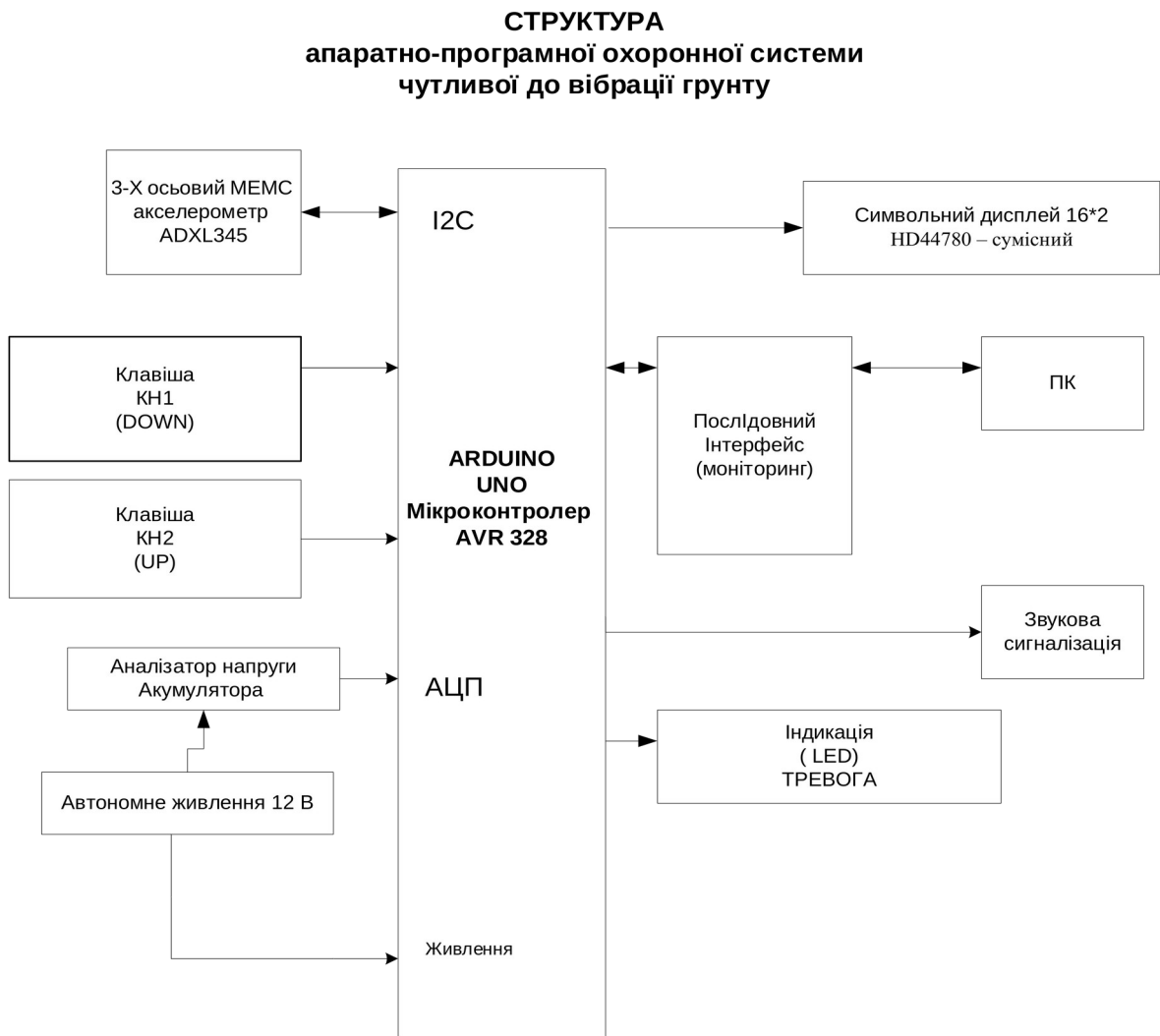


Рис. 3.14. Загальна структура апаратно-програмно охоронної системи чутливої до вібрації ґрунту на платформі Arduino

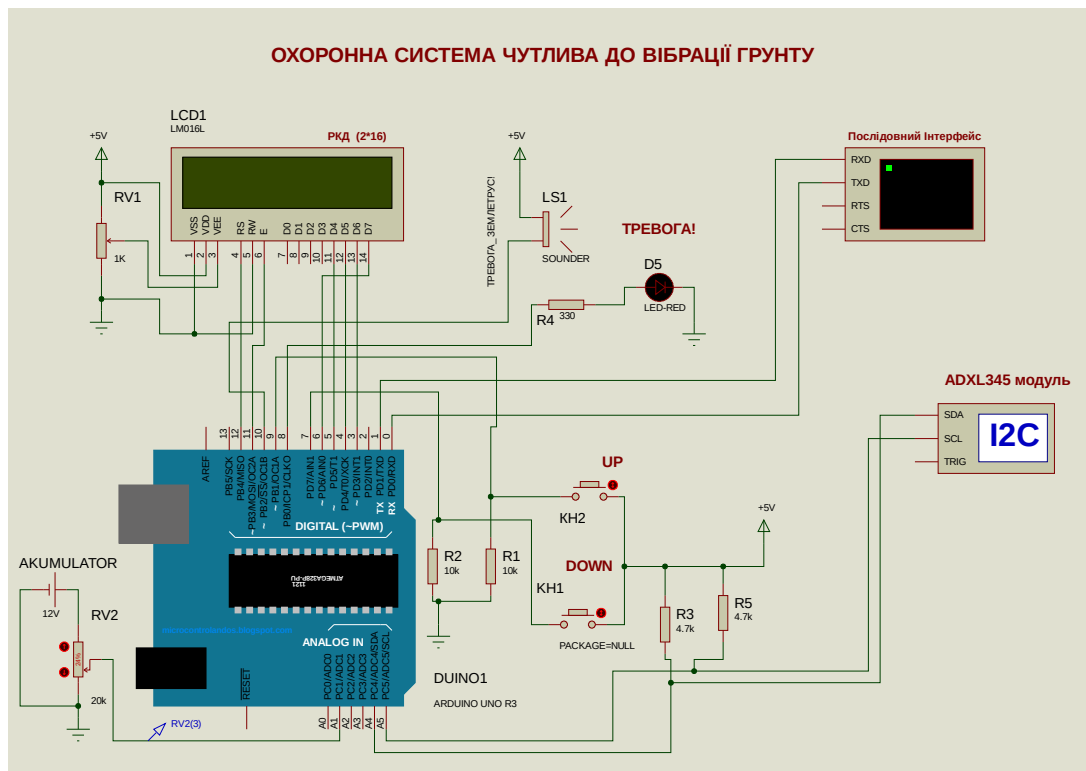


Рис. 3.15. Апаратне забезпечення апаратно-програмної охоронної системи чутливої до вібрації ґрунту на платформі Arduino UNO (в Proteus ISIS)

На Рис.3.15 представлено схему апаратно-програмної охоронної системи чутливої до вібрації ґрунту. Система постійно контролює рівень сейсмічної активності, виводить результати моніторингу на РКД та передає їх по послідовному порту на ПК. При підвищенні вібраційної активності понад допустимі встановлені клавішами КН1, КН2 рівні вмикається звукова (п'єзовипромінювач LS1), та світлова “Тривога” (блимає червоний LED) сигналізація, тривожна інформація передається на ПК. На РКД періодично виводиться інформація про рівень вібраційної активності ґрунту та проекції прискорень по осях X, Y, Z. З допомогою клавіші КН1 “DOWN” можна підвищити чутливість системи (допустимий поріг спрацювання зменшується) чутливість – збільшується. Клавішею КН2 “UP” допустимий поріг спрацювання системи моніторингу збільшують, а чутливість – зменшують. Крок регулювання встановлено 0.01 м/с^2 , що дозволяє точно налаштувати систему на відповідний рівень спрацювання. Для коректної автономної роботи системи додатково вимірюється стан заряду акумулятора, (вхід А1 – АЦП Arduino UNO). У процесі виконання основної програми (робочий режим)

система здійснює циклічне сканування стану функціональних клавіш КН1 (“UP”) та КН2 (“DOWN”). Це дозволяє оператору в реальному часі корегувати чутливість охоронного комплексу. Налаштування граничного значення сейсмічної активності (A_d) відбувається за наступною логікою:

- Збільшення порогу: При натисканні клавіші КН1 (“UP”) параметр A_d зростає з визначеним кроком. Максимально допустиме значення обмежене рівнем 0.5.
- Зменшення порогу: Натискання клавіші КН2 (“DOWN”) призводить до покрокового зниження параметра A_d до мінімальної межі, що становить 0,01.

Програмне забезпечення проводить постійне порівняння поточного показника приведенного прискорення (A) із заданим пороговим значенням (A_d): Якщо виміряна амплітуда вібрації перевищує встановлений ліміт ($A > A_d$), система ідентифікує подію як тривожну. Комплекс генерує сповіщення про небезпеку, дублюючи інформацію на рідкокристалічному дисплеї та передаючи відповідний сигнал через послідовний інтерфейс на персональний комп’ютер. Після подачі живлення мікроконтролер послідовно налаштовує роботу наступних модулів:

- Модуль візуалізації: Активація рідкокристалічного дисплея (LCD).
- Комунікаційні інтерфейси: Налаштування послідовного порту (UART) для обміну даними.
- Акустична система: Конфігурація генератора ШІМ для керування звуковим випромінювачем (пищиком).
- Органи керування: Налаштування портів введення для зчитування стану клавіш.

Окремим критичним етапом є перевірка працездатності MEMS-акселерометра:

Якщо датчик ADXL345 не відповідає або не проходить ініціалізацію, на екрані з’являється сповіщення “ADXL345 ERROR”, а подальша робота блокується. При успішному запуску: На дисплеї протягом 2 секунд

відображається привітання “VIBRATION SECURITY SYSTEM!”, що свідчить про готовність комплексу до роботи. Після завершення підготовчих процедур система переходить до виконання основних завдань у циклічному режимі:

1. Моніторинг живлення: Контроль поточної напруги на акумуляторній батареї.
2. Збір даних: Безперервне вимірювання прискорень по осях.
3. Порівняльний аналіз: Співставлення обчисленого приведенного прискорення з лімітом чутливості.
4. Інтерактивність: Можливість оперативного корегування порогу спрацювання за допомогою клавіш “UP” (KH1) та “DOWN” (KH2).

На РКД виводиться оперативна інформація. Верхній рядок: вектор прискорення A ; допустимий поріг чутливості (спрацювання сигналізації – A_d ; якщо акумулятор розряджений то символ “+ “ свідчить що напруга менша 11 вольт. У нижньому рядку відображаються реальні значення прискорень по осях A_x , A_y , A_z . Це дозволяє визначити напрям небезпеки. При вібрацій активності ґрунту, що перевищує порогове значення A_d в нижній рядок виводиться повідомлення: “ALARM !!!” (Тривога). В послідовний порт виводиться тільки тривожна інформація: AKUMULATOR- LOW LEVEL (розряджений акумулятор); ALARM !!! (Тривога), значення реального прискорення; поріг спрацювання A_d ; прискорення вздовж осей A_x , A_y , A_z . При виникненні тривоги, додатково вмикається світлодіод “Тривога” і звукова сигналізація.

Програмне забезпечення для охоронної системи чутливої до вібрації ґрунту написано в Arduino IDE з максимальним використанням бібліотек цього середовища і інструментів для подальшого завантаження компільованого коду в Arduino UNO-платформу і Proteus ISIS для симуляції.

РОЗДІЛ 4

ПРОГРАМНО-АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ АПАРАТНО-ПРОГРАМНОЇ ОХОРОННОЇ СИСТЕМИ, ЧУТЛИВОЇ ДО ВІБРАЦІЇ ГРУНТУ

4.1. Алгоритм роботи апаратно-програмної охоронної системи, чутливої до вібрації ґрунту

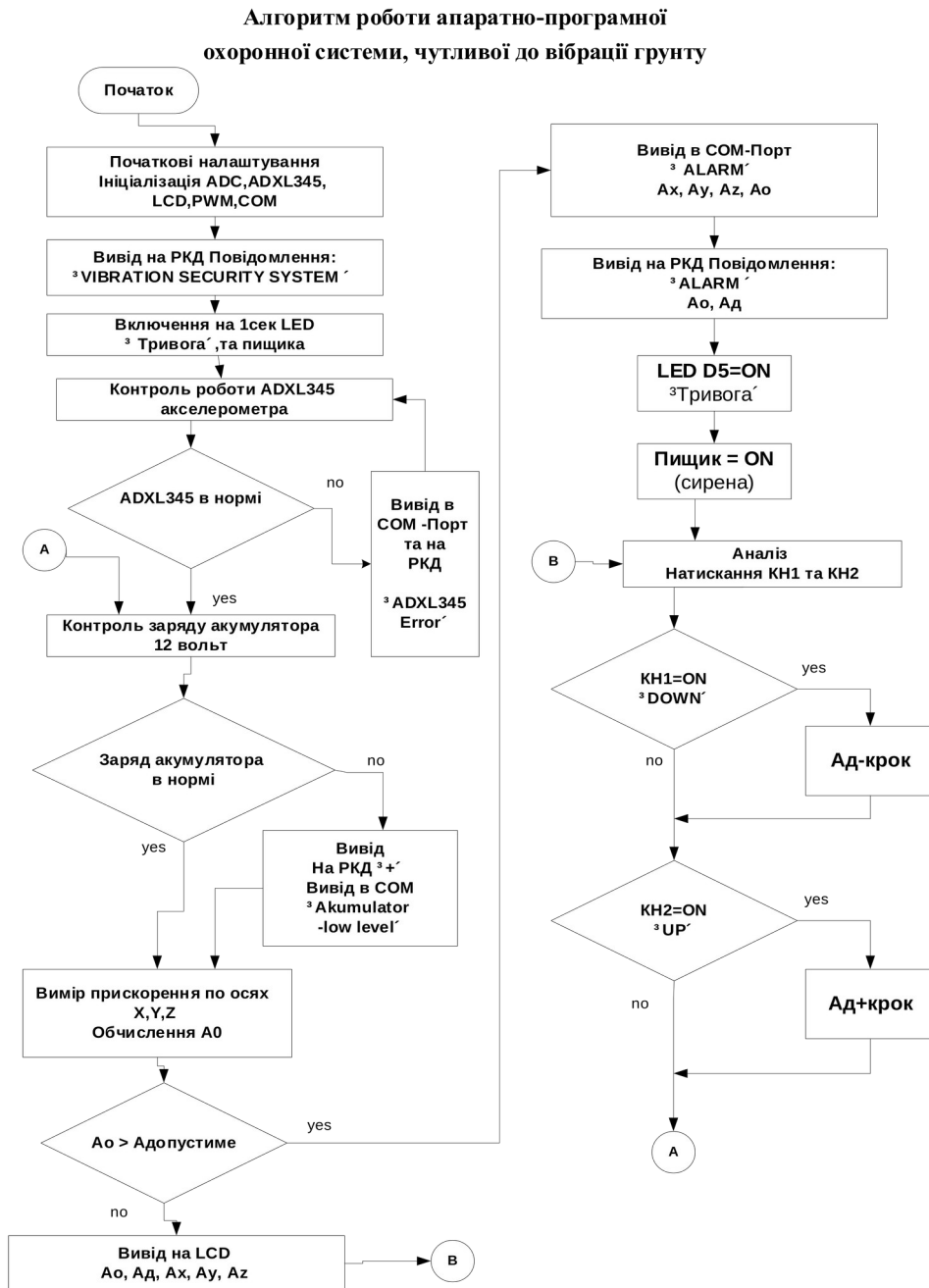


Рис. 4.1. Алгоритм роботи апаратно-програмної охоронної системи, чутливої до вібрації ґрунту

4.2. Розробка програмних модулів для роботи з цифровим трьохосьовим MEMS-акселерометром ADXL345

Для роботи з акселерометром ADXL345 розроблено такі функції:

Читати байт з регістра ADXL345:

```
char ADXLReadByte(unsigned char address)
{
    unsigned char data; // байт даних з ADXL345 регістра

    i2c_start(); // перевірка шини
    i2c_write(ADXL345_W); // write - адрес ADXL345 на i2c
    i2c_write(address); // адрес для читання
    i2c_start();
    i2c_write(ADXL345_R); // read - адрес ADXL345 на на i2c
    data=i2c_read(0); // отримані дані.
    delay_ms(20);
    i2c_stop();
    return(data);
}
```

Читати 2 байти з регістра ADXL345:

```
int ADXLReadInt(unsigned char address)
{
    unsigned char msb, lsb;
    int temp;

    i2c_start();
    i2c_write(ADXL345_W); // write - адрес ADXL345 на i2c
    i2c_write(address); // адрес регістра читання
    i2c_start();
    i2c_write(ADXL345_R); // read - адрес ADXL345 на на i2c
    msb = i2c_read(0); // read старший байт
    lsb = i2c_read(0); // read молодший байт
    delay_ms(20);
    i2c_stop();
    temp =((int)msb<<8)+lsb; // формування слова
    return ( temp );
}
```

Записати байт в регістр ADXL345:

```
void ADXLWriteByte(unsigned char address, unsigned char data)
{
    i2c_start();
    i2c_write(ADXL345_W); // write - адрес ADXL345 на i2c
    шині
    i2c_write(address);
    i2c_write(data); // дані запису
    delay_ms(20);
    i2c_stop();
}
```

Ініціалізація ADXL345:

```
/* Ініціалізація ADXL345 */
{
    i2c_start();
    i2c_write(ADXL345_W); // write - адрес ADXL345 на i2c
    i2c_write(ADXL345_REG_POWER_CTL);
    i2c_write(0); // standby mode =0
    delay_ms(20);
    i2c_stop();
    i2c_start();
    i2c_write(ADXL345_W); // write - адрес ADXL345 на i2c
    i2c_write(ADXL345_REG_POWER_CTL);
    i2c_write(1<<3); // Measure =1
    delay_ms(20);
    i2c_stop();
}
```

Вибір роздільної здатності вимірювання акселерометра 2-4-8-16g:

```
// resol - вибір роздільної здатності 2-4-8-16 g
// Full resolution, +/-16g, 4mg/LSB.
setDataFormatControl(0x0B);
void Set_Resolution (Set_Resol resol)
{
    i2c_start();
    i2c_write(ADXL345_W); // write - адрес ADXL345 на i2c
    i2c_write(ADXL345_REG_DATA_FORMAT);
    i2c_write(resol);
    i2c_stop();
}
```

Читати регістри з прискоренням:

```
void readAccel(void) // читати регістри з прискоренням
{
    Wire.beginTransmission(0x53); Wire.write(0x32); //RG X0-байт
    Wire.endTransmission(); Wire.requestFrom(0x53, 1);
    x0 = Wire.read(); // читати молодший байт ax
    Wire.beginTransmission(0x53);
    Wire.write(0x33); //RG X1- байт
    Wire.endTransmission();
    Wire.requestFrom(0x53, 1);
    x1 = Wire.read(); // читати старший байт ax
    Wire.beginTransmission(0x53);
    Wire.write(0x34); //RG Y0 -байт
    Wire.endTransmission(); Wire.requestFrom(0x53, 1);
    y0 = Wire.read(); // читати молодший байт ay
    Wire.beginTransmission(0x53);
    Wire.write(0x35); //RG Y1-байт
    Wire.endTransmission(); Wire.requestFrom(0x53, 1);
    y1 = Wire.read(); // читати старший байт ay
    Wire.beginTransmission(0x53);
    Wire.write(0x36); //RG Z0- байт
    Wire.endTransmission(); Wire.requestFrom(0x53, 1);
    z0 = Wire.read(); // читати молодший байт az
    Wire.beginTransmission(0x53);
    Wire.write(0x37); // RG Z1- байт
    Wire.endTransmission(); Wire.requestFrom(0x53, 1);
    z1 = Wire.read(); // читати старший байт z
}
```

Функція калібрування ADXL345:

```
void calibration(void)
{
    xoffset = ( xmax + xmin ) * 0.5 ; // зміщення по осі X
    yoffset = ( ymax + ymin ) * 0.5 ; // зміщення по осі Y
    zoffset = ( zmax + zmin ) * 0.5 ; // зміщення по осі Z
    xsense = (( xmax + xmin * (-1)) * 0.5) * D;
    // чутливість по осі X
    ysense = (( ymax + ymin * (-1)) * 0.5) * D;
    // чутливість по осі Y
    zsense = (( zmax + zmin * (-1)) * 0.5) * D;
    // чутливість по осі Z
}
```

4.3. Програмна реалізація головного алгоритму роботи апаратно-програмної охоронної системи, чутливої до вібрації ґрунту

Програма реалізує алгоритм роботи апаратно-програмної системи моніторингу сейсмічної активності згідно Рис. 4.1. та приведена у додатку.

4.4. Моделювання та аналіз результатів роботи апаратно-програмної охоронної системи, чутливої до вібрації ґрунту в Proteus ISIS

Для реалізації алгоритму роботи платформи Arduino Uno на базі мікроконтролера ATmega328 використовується середовище Arduino IDE, результатом компіляції в якому є файл формату .hex. Саме цей файл виступає прошивкою для МК. Етап тестування цифрового виробу проходить у програмному комплексі Proteus VSM, який розроблений британською компанією Labcenter Electronics. Дана САПР є універсальним інструментом для імітаційного моделювання апаратної частини, включаючи платформи Arduino та різні типи процесорів. Архітектура Proteus включає підпрограму ISIS (проектування та віртуальна перевірка схем) та компонент ARES (проектування конструкторської документації для друкованих плат). Спільне використання цих засобів дозволяє наочно продемонструвати працездатність пристрою, що підтверджується результатами симуляції.

На рисунках нижче зображено процеси моделювання апаратно-програмної охоронної системи, чутливої до вібрації ґрунту в Proteus ISIS.

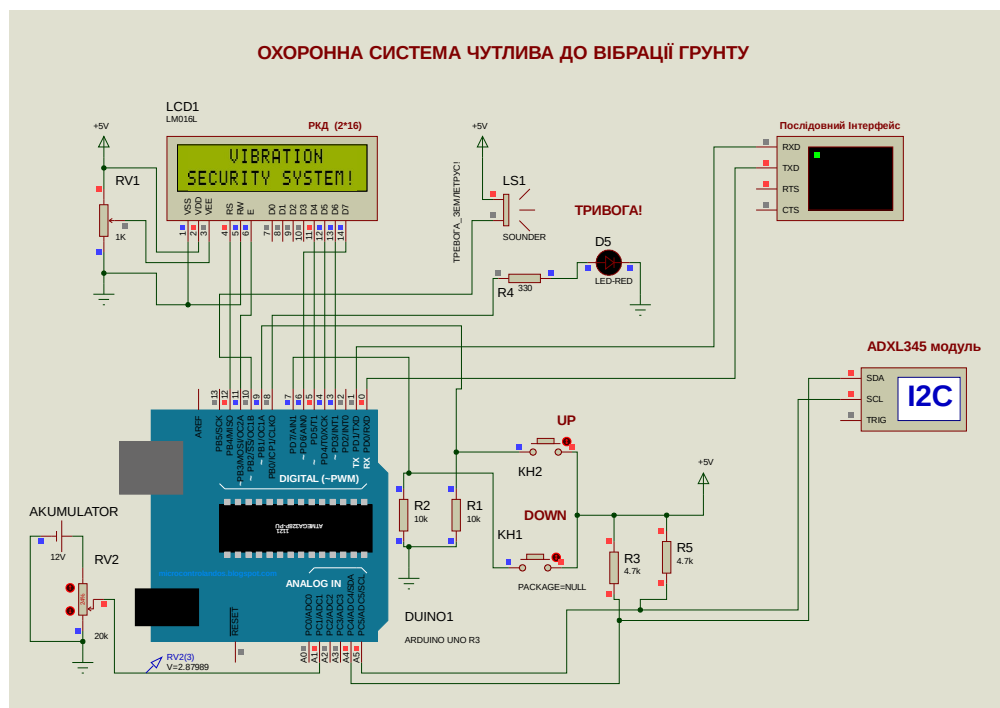


Рис. 4.2. Моделювання роботи апаратно-програмної охоронної системи, чутливої до вібрації ґрунту в Proteus ISIS – ініціалізація

При ініціалізації Рис.4.2 на РКД виводиться повідомлення “VIBRATION SECURITY SYSTEM!”. Світлодіод (D5) “ТРИВОГА” та пищик вмикаються на 1 сек. Тестується акселерометр ADXL345, якщо при його тестуванні виникла помилка то відповідні повідомлення, Рис.4.3.

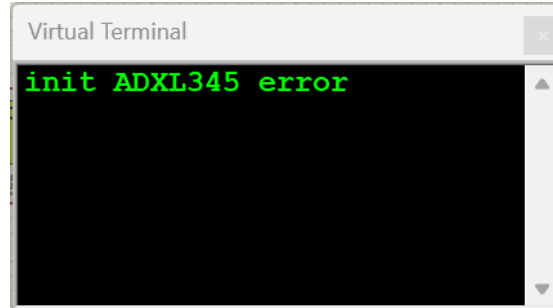


Рис. 4.3. Вивід повідомлення в ПК про помилку ADXL345 акселерометра

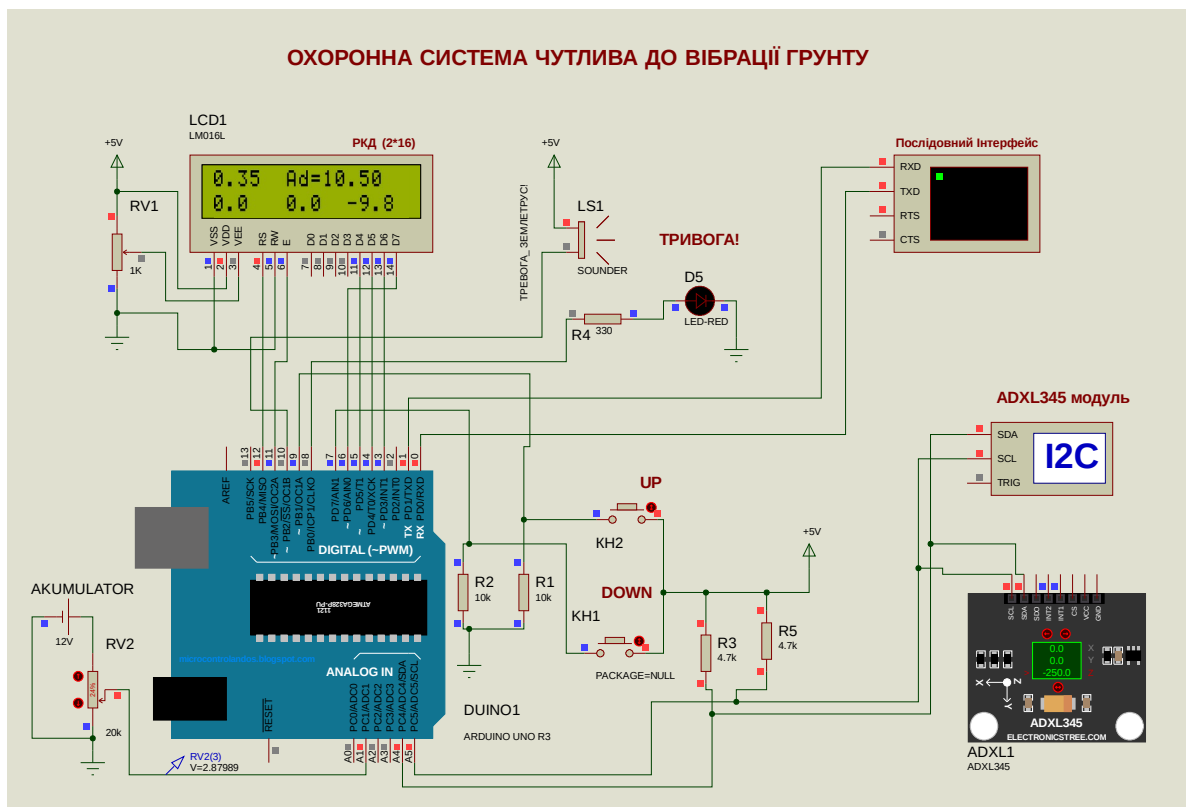


Рис. 4.4. Моделювання роботи апаратно-програмної охоронної системи, чутливої до вібрації ґрунту в Proteus ISIS – робочий режим

Ad – це допустиме прискорення (чутливість), яке можна налаштувати за допомогою KH1, KH2.

З одночасним виводом на РКД інформація виводиться в послідовний порт на ПК. Кожної секунди відбувається поновлення інформації (Рис.4.5).

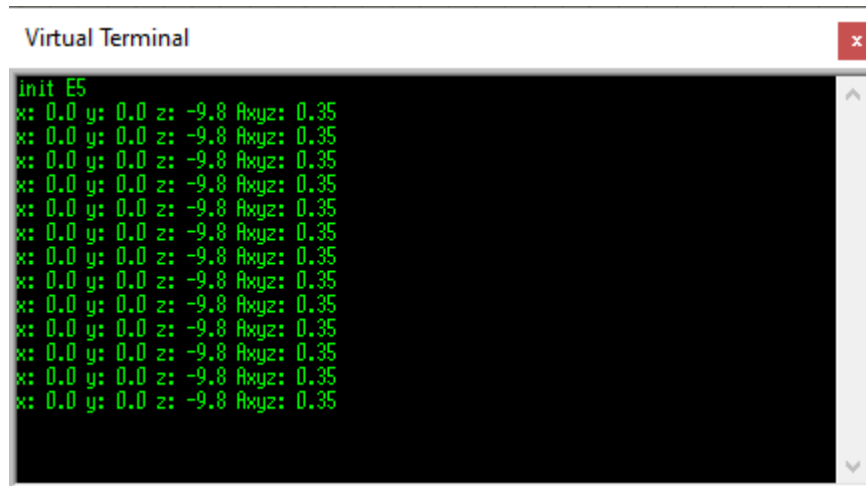


Рис. 4.5. Вивід інформації в ПК

Вивід прискорень A_x , A_y , A_z , A_d , A в послідовний порт та на РКД.

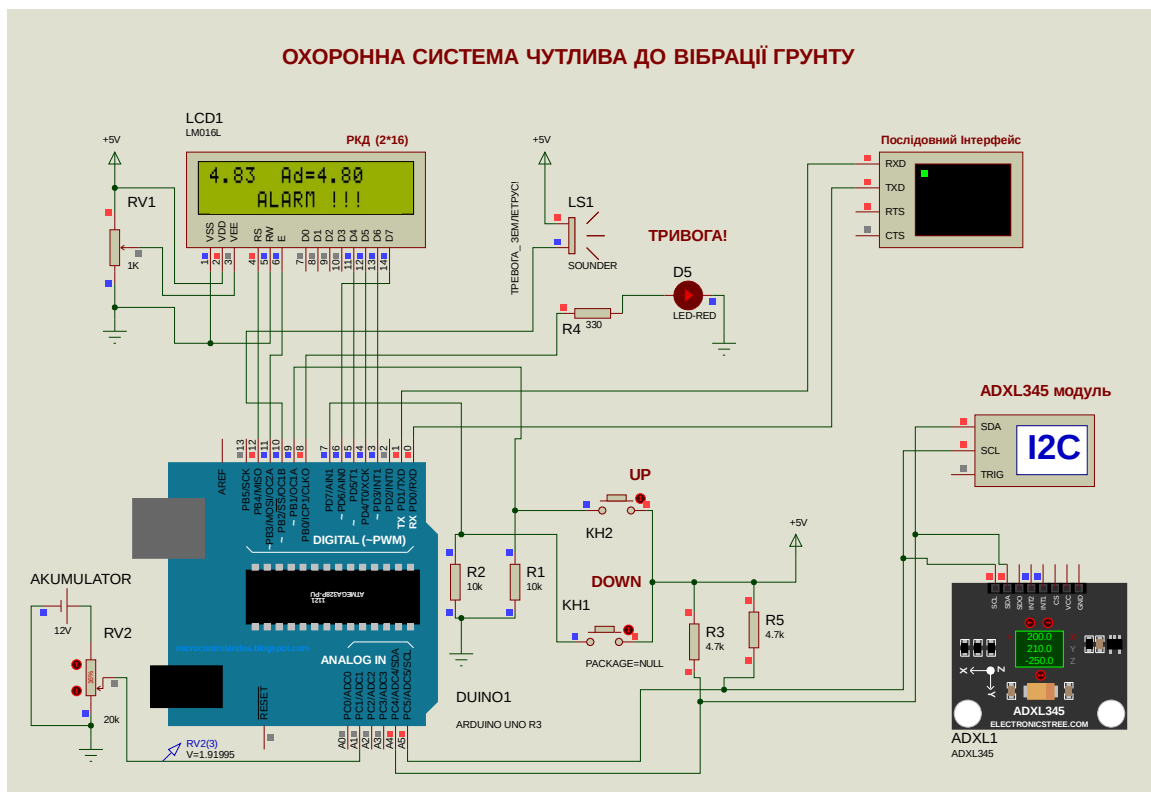


Рис. 4.6. Моделювання перевищення допустимого рівня сейсмічної активності ґрунту в Proteus ISIS

На Рис. 4.6 змодельовано перевищення допустимого рівня сейсмічної активності. Допустимий рівень встановлений клавішами КН1 та КН2 рівний кроку $0,1 \text{ м/с}^2$.

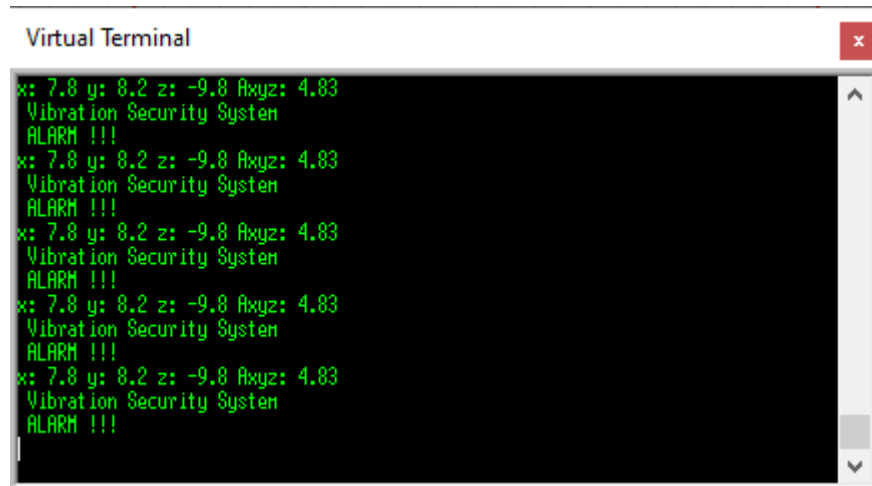


Рис. 4.7. Вивід на ПК повідомлення “Тривога”

При перевищенні цього рівня вмикається червоний світлодіод “Тривога” та звуковий сигнал тривоги (пищик LS1). На РКД виводиться повідомлення “ALARM !!!”. В порт (Рис. 4.7) виводиться повідомлення “ALARM !!!” та величина прискорення сейсмічної активності ґрунту по напрямках. Якщо вібрації ґрунту в межах встановлених допусків сигнали “Тривоги” – вимкнені. На Рис. 4.8. змодельовано пониження напруги живлення. Симуляція проводилася з допомогою потенціометра RV2. При розрядженому акумуляторі на РКД у верхній рядок РКД виводиться символ “+”.

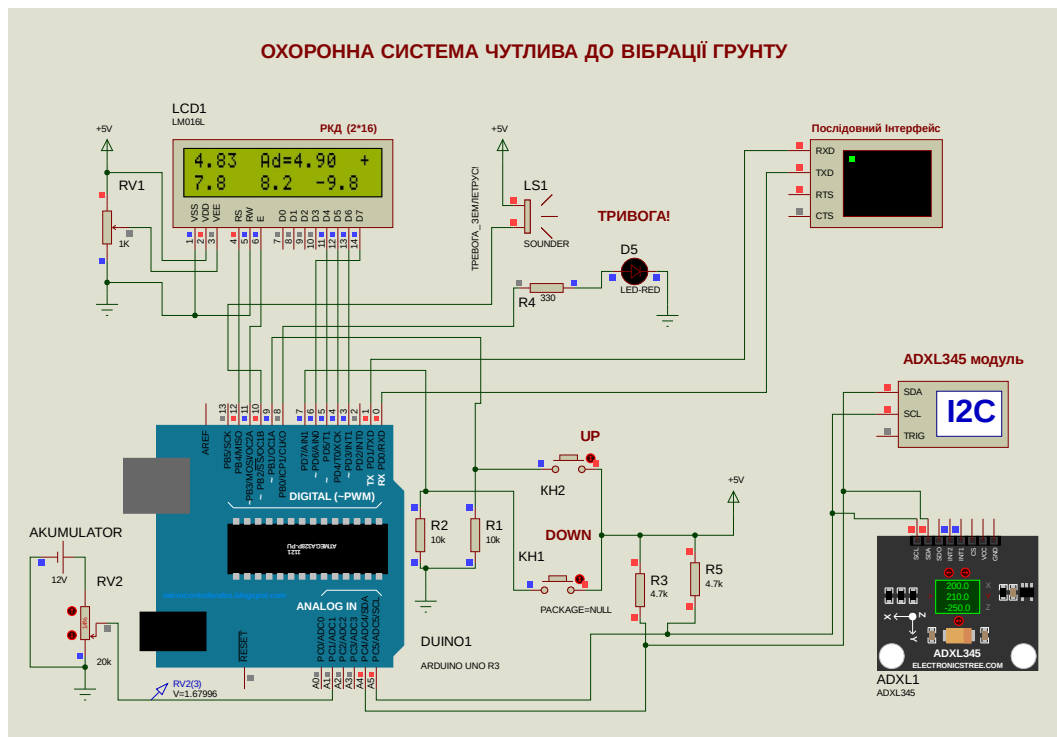
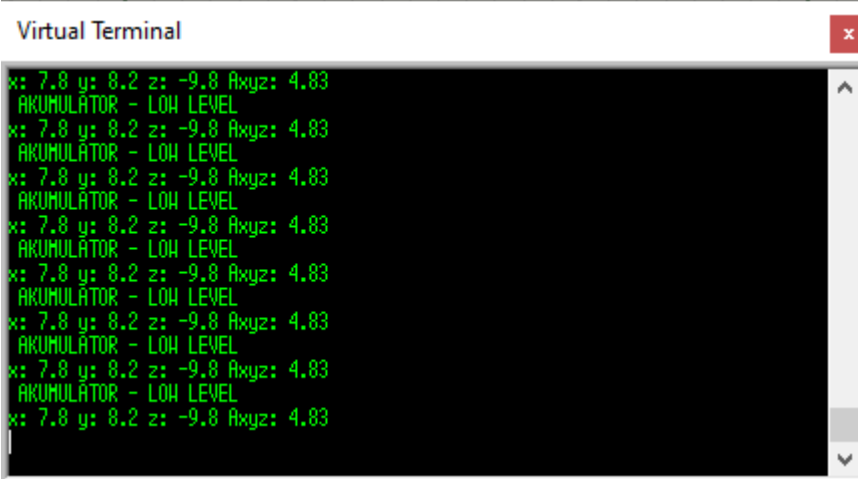


Рис. 4.8. Моделювання пониження напруги живлення



```
Virtual Terminal
x: 7.8 y: 8.2 z: -9.8 Akuz: 4.83
AKUMULĀTOR - LOW LEVEL
x: 7.8 y: 8.2 z: -9.8 Akuz: 4.83
AKUMULĀTOR - LOW LEVEL
x: 7.8 y: 8.2 z: -9.8 Akuz: 4.83
AKUMULĀTOR - LOW LEVEL
x: 7.8 y: 8.2 z: -9.8 Akuz: 4.83
AKUMULĀTOR - LOW LEVEL
x: 7.8 y: 8.2 z: -9.8 Akuz: 4.83
AKUMULĀTOR - LOW LEVEL
x: 7.8 y: 8.2 z: -9.8 Akuz: 4.83
AKUMULĀTOR - LOW LEVEL
x: 7.8 y: 8.2 z: -9.8 Akuz: 4.83
AKUMULĀTOR - LOW LEVEL
```

Рис. 4.9. Вивід на ПК повідомлення про низьку напругу живлення
Шляхом симуляції в Proteus ISIS доведено коректність логіки роботи
головного алгоритму та програмних модулів у режимі реального часу.

ВИСНОВКИ

У представлений кваліфікаційній роботі проведено комплексне дослідження сучасних методик та інструментарію для реєстрації вібрацій ґрунту. На основі аналітичного огляду було обґрунтовано вибір МЕМС-технологій як найбільш перспективних для створення первинних перетворювачів вібраційної активності.

На основі широко поширених апаратних платформах Arduino, розроблена апаратно-програмна охоронна система чутлива до вібрацій ґрунту. Спроектовано програмно-апаратний комплекс на базі популярної платформи Arduino Uno (МК Atmel ATmega328). Система забезпечує прецизійне вимірювання прискорень, адаптивне налаштування порогів спрацювання та передачу даних на ПК через UART-інтерфейс.

Окрім моніторингу вібрацій, реалізовано візуалізацію даних на рідкокристалічному індикаторі, систему звукового та світлового оповіщення, а також інтелектуальний контроль стану автономного живлення. Пристрій характеризується низькою собівартістю, компактністю та можливістю роботи у двох режимах – стаціонарному (мережа 220 В через адаптер DC +7..12 вольт) та автономному (від акумуляторних батарей).

Висока надійність охоронної системи чутливої до вібрацій ґрунту забезпечується оперативним її реагуванням на аварійні ситуації : перевищення допустимого рівня сейсмічної активності ґрунту, контроль заряду акумулятора. Своєчасні повідомлення дозволять вберегти особовий склад, матеріальні цінності від зовнішніх впливів, виявити їх джерела і своєчасно зменшити збитки.

При розробці апаратно-програмної системи охоронної системи чутливої до вібрацій ґрунту використано сучасну елементну базу. Основними її елементами є МЕМС-акселерометр ADXL345, апаратна платформа Arduino UNO з мікроконтролером Atmel ATmega328, символічний рідкокристалічний дисплей.

Для підтвердження працездатності розроблених рішень було виконано наступне: У середовищі Proteus VSM розроблено повну електричну принципову схему комплексу. Мовою С в Arduino IDE створено системне ПЗ, що включає модулі для взаємодії з цифровим акселерометром ADXL345 та алгоритми обробки сейсмічних сигналів, контролю зарядки акумулятора. Проведено моделювання роботи системи в емуляторі Proteus ISIS. Шляхом симуляції в Proteus ISIS доведено коректність логіки роботи головного алгоритму та програмних модулів у режимі реального часу. Впровадження даної системи дозволяє організувати оперативне виявлення загроз, що сприяє мінімізації матеріальних збитків та підвищенню безпеки людей у охоронних зонах.

У процесі роботи над проектом було поглиблено професійні навички у сферах проектування вбудованих систем (Embedded Systems), алгоритмізації процесів керування та практичного застосування МЕМС-сенсорів у сучасних засобах автоматизації.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Sarful Hassan, Mehadi Hasan Maruf. Arduino Programming for Absolute Beginners: Master Arduino from Scratch with Easy-to-Follow Projects and Examples. – Published by Kindle Edition, 2024. – p. 478.
2. Norman Dunbar. Arduino Software Internals: A Complete Guide to How Your Arduino Language and Hardware Work Together (Maker Innovations Series). – Published by Apress (Kindle Edition), 2nd ed., 2024. – p. 717.
3. Sarful Hassan. Arduino Programming Essentials: A Practical Handbook for Makers, Engineers, and Students. – Independently published, 2024. – p. 308.
4. Jonathan Osher and Hugh Blemings. Practical Arduino: Cool Projects for Open Source Hardware. – Springer-Verlag New York, 2009 – p. 445.
5. Massimo Banzi. Getting Started with Arduino. – 2nd Edition. O'Really Media Inc., 2011 – p. 130.
6. Emily Gertz & Patrick Di Justo. Environmental Monitoring with Arduino. 32. Building Simple Devices to Collect Data About the World Around Us. – Published by O'Really Media Inc., 2012 – p. 96.
7. Середовище розробки Arduino IDE. Електронний ресурс: <https://www.arduino.cc/en/Main/Software>.
8. Kimmo and Tero Karvinen. Make: Arduino Bots and Gadgets. – Published by O'Really Media Inc., 2011 – p. 297.
9. Brian Evans. Beginning Arduino Programming. Writing Code for the Most Popular Microcontroller Board in the World. – Springer Science + Business Media, 2011 – p. 271.
10. Michael Margolis. Arduino Cookbook, Second Edition. – Published by O'Reilly Media Inc., 2012. – p. 724.
11. Enrique Ramos Melgar and Ciriaco Castro Diez with Przemek Jaworski. Arduino and Kinect Projects. Design, Build, Blow Their Minds. – Springer Science+Business Media New York, 2012. – p. 411.

12. Brian Huang and Derek Runberg. The Arduino Inventor's Guide. – SparkFun Electronics, 2017. – p. 472.
13. Farine, M., N.Thorburn, and D. Mougnot. “General application of MEMS sensors for land seismic acquisition - is it time?”, The Leading Edge, 23, pp. 246-250, 2004.
14. Takao Aizawa, Toshinori Kimura, Toshifumi Matsuoka, Tetsuya Takeda, Youichi Asano; “Application of MEMS accelerometer to geophysics”; International Journal of the JCRM; Japanese committee for Rock Mechanics; Volume 4, number 2 , pp 1-4, 2008.
15. Yih-Min Wu; Kanamori, H. “Development of an Earthquake Early Warning System Using Real – Time Strong Motion Signals” Sensors 8, pp. 1-9, 2008.
16. Sanjib Kalita, J.N. Borole and Dr. K.P. Rane,”Wireless Earthquake Alarm System using ATmega328p, ADXL335 and XBee S2”, June 2014.
17. Analog Devices Datasheet for ADXL345 – 3 Axis Digital Accelerometer (Rev. B). Norwood: Analog Devices, 2009. http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf.

ДОДАТКИ

Додаток 1

Головний програмний модуль прихованої охоронної системи, чутливої до вібрації ґрунту з використанням МЕМС-акселерометра

```

/*****
Project : Апаратно-програмна охоронна система, чутлива до вібрації ґрунту
Comments: this program is designed on the base of AVR ATmega328 microcontroller
Chip type      : ATmega328
Clock frequency : 16,000000 MHz
*****/

#include <SoftwareSerial.h> // бібліотека COM
#include <LiquidCrystal.h> // бібліотека LCD
/*
 * Для підключення LCD screen to your board, connect the following pins:
 * LCD RS      pin to digital pin 12
 * LCD Enable  pin to digital pin 11
 * LCD D4      pin to digital pin 5
 * LCD D5      pin to digital pin 4
 * LCD D6      pin to digital pin 3
 * LCD D7      pin to digital pin 6 !!!
 */
// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 6;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

#include <Math.h> // бібліотека математичних функцій
#include <Wire.h> // бібліотека інтерфейсу I2C
#define DEVICE (0x53) // Адрес ADXL345 на шині I2C
#define ADXL345_REG_DEVID (0x00)
// Device ID має значення 0xE5 (only read)
#define ADXL345_REG_THRESH_TAP (0x1D)
#define ADXL345_REG_OFSX (0x1E) // X-axis offset коефіцієнт 15,6 mg/LSB) #define
ADXL345_REG_OFSY (0x1F) // Y-axis offset
#define ADXL345_REG_OFSZ (0x20) // Z-axis offset
// Tap duration - макс. Тривал. поштовху - 625 мкс/LSB
#define ADXL345_REG_DUR (0x21)
// Tap latency -часова затримка після першого оштовху
#define ADXL345_REG_LATENT (0x22)
// Tap window часовий інтервал - 1,25 мс/LSB
#define ADXL345_REG_WINDOW (0x23)
// -виявлення факта зміни прискорення
#define ADXL345_REG_THRESH_ACT (0x24)
// Inactivity threshold -подія бездіяльність
#define ADXL345_REG_THRESH_INACT (0x25)
// Inactivity time - для часу 1 сек/LSB
#define ADXL345_REG_TIME_INACT (0x26)
#define ADXL345_REG_ACT_INACT_CTL (0x27) // Axis enable control
#define ADXL345_REG_THRESH_FF (0x28) // Free-fall threshold - 62,5 mg/LSB
// Free-fall time- 5 мс/LSB, значення 100 до 350 мс.
#define ADXL345_REG_TIME_FF (0x29)
#define ADXL345_REG_TAP_AXES (0x2A) // задати осі X, Y, Z, поштовхів

```

```

// Source for single/double tap
#define ADXL345_REG_ACT_TAP_STATUS (0x2B)
//частота вихідних даних (1100)= 400 Гц
#define ADXL345_REG_BW_RATE (0x2C)
#define ADXL345_REG_POWER_CTL (0x2D)
// DATAx, DATAy, DATAz-обновляються з част. Wakeup
#define ADXL345_REG_INT_ENABLE (0x2E) // Interrupt enable control
#define ADXL345_REG_INT_MAP (0x2F) // Interrupt mapping control
#define ADXL345_REG_INT_SOURCE (0x30) // Source of interrupts
#define ADXL345_REG_DATA_FORMAT (0x31) // Data format control FULL_RES
#define ADXL345_REG_DATAx0 (0x32) // X-axis data 0
#define ADXL345_REG_DATAx1 (0x33) // X-axis data 1
#define ADXL345_REG_DATAy0 (0x34) // Y-axis data 0
#define ADXL345_REG_DATAy1 (0x35) // Y-axis data 1
#define ADXL345_REG_DATAz0 (0x36) // Z-axis data 0
#define ADXL345_REG_DATAz1 (0x37) // Z-axis data 1
#define ADXL345_REG_FIFO_CTL (0x38) // FIFO control
#define ADXL345_REG_FIFO_STATUS (0x39) // FIFO status
// output data rate - швидкість передачі даних
// Used with register 0x2C (ADXL345_REG_BW_RATE) to set bandwidth
#define ADXL345_BW_1600 0xF // 1111 - 3200 output data rate
#define ADXL345_BW_800 0xE // 1110 - 1600 - швидкість передачі даних
#define ADXL345_BW_400 0xD // 1101 - 800
#define ADXL345_BW_200 0xC // 1100 - 400
#define ADXL345_BW_100 0xB // 1011 - 200
#define ADXL345_BW_50 0xA // 1010 - 100
#define ADXL345_BW_25 0x9 // 1001 - 50
#define ADXL345_BW_12 0x8 // 1000 - 25
#define ADXL345_BW_6 0x7 // 0111 - 12,5
#define ADXL345_BW_3 0x6 // 0110 - 6,25
#define ADXL345_BW_1 0x5 // 0101 - 3,125
#define ADXL345_BW_07 0x4 // 0100 - 1,56
#define ADXL345_BW_03 0x3 // 0011 - 0,78
#define ADXL345_BW_01 0x2 // 0010 - 0,39
#define ADXL345_BW_00 0x1 // 0001 - 0,195
#define ADXL345_BW_0 0x0 // 0000 - 0,098
#define offsetX 0 // OFFSET values
#define offsetY 0 // 15.6 mg per increment
#define offsetZ 0 // 15.6 mg/LSB
#define gainX 1 // GAIN factors
#define gainY 1
#define gainZ 1
// offsetAxis = 0.5 * (Acel+1g + Accel-1g)
// gainAxis = 0.5 * ((Acel+1g - Accel-1g)/1g)
char buffer[33]; // for LCD conversion
byte _buff[6];
byte id;
//змінні - для зберігання значень прискорень по осях
// (додатні і від'ємні значення)
volatile signed int axisX,axisY,axisZ;
volatile signed int X,Y,Z;
float k_offset=15.6;
// для регістра зміщення коефіцієнт рівний 15,6 mg/LSB.

```

```

float k_fr=3.9; // коефіцієнт вихідних даних 3,9 mg/LSB,
float g = 9.81; // прискорення вільного падіння м/с2
volatile float xa=0.0,ya=0.0,za=0.0;
float daxyz=10.1; // зміщення для прискорення
float Ad = 10.5; // допустиме нульове прискорення
float krok = 0.1; // крок налаштування допустимого прискорення
int i=0; // duty для пищика
const char SPEAKER = 10; // пищик під'єднано на pin 10
const int RED = 8; // LED "тревога" під'єднано на pin 8
const int BUTTON1 = 7; // на pin.7 під'єднано клавішу DOWN
const int BUTTON2 = 9; // на pin.9 під'єднано клавішу UP
int val = 0, val2=0; // val will be used to store the state of the input pin
int old_val = 0; // this variable stores the previous value of "val"
int old_val2 = 0;
const int AKUMIN = 37; // 3.7 вольт*10 = 11 вольт
float akumulator=0; // напруга на акумуляторі
long int n=0; // напруга на акумуляторі*10
//Вибір роздільної здатності ADXL345_REG_DATA_FORMAT
typedef enum
{
RESOLUTION_FULL = 8, // або 0xB = 11
RESOLUTION_2 = 0,
RESOLUTION_4 = 1,
RESOLUTION_8 = 2,
RESOLUTION_16 = 3
} Set_Resol;
//Вид функції переривання - Регістр INT_ENABLE (0x2E)
typedef enum
{
OVERRUN = 1,
WATERMARK = 2,
FREE_FALL = 4,
INACTIVITY = 8,
ACTIVITY = 16,
DOUBLE_TAP = 32,
SINGLE_TAP = 64,
DATA_READY = 128
} INT_Func;
//призначити вивід переривання
typedef enum
{
INT_1 = 1,
INT_2 = 2
} Set_Map;
//дозвіл/заборона роботи функції
typedef enum {
ENABLE = 1,
DISABLE = 2
} Set_Func;
void setup() {
lcd.begin(16, 2); // set up the LCD's number of columns and rows
pinMode(BUTTON1, INPUT); // and BUTTON1 is an input
pinMode(BUTTON2, INPUT); // and BUTTON is an input

```

```

lcd.setCursor(0, 0);
lcd.print("VIBRATION SECURITY SYSTEM"); // Print a message to the LCD.
lcd.setCursor(1, 1);
lcd.print(" MONITORING");
delay(500);
delay(500);
lcd.clear();
Wire.begin();           // ініціалізувати шину i2c
Serial.begin(9600);     // ініціалізувати serial порт for output.
Serial.print("init ");
// ADXL345_Init ();
writeTo(ADXL345_REG_POWER_CTL, 0x00); // standby mode =0
// clear any existing offsets
writeTo(ADXL345_REG_OFSX, 0); // обнулення OFSX
writeTo(ADXL345_REG_OFSY, 0); // обнулення OFSY
writeTo(ADXL345_REG_OFSZ, 0); // обнулення OFSZ
writeTo(ADXL345_REG_FIFO_CTL, 0); // Bypass - буфер відключено
writeTo(ADXL345_REG_BW_RATE, ADXL345_BW_200); // Buffer Speed- 400Hz
writeTo(ADXL345_REG_DATA_FORMAT, 0x0B);
// Full resolution, +/-16g, 3,9 mg/LSB.
Set_Activity_Threshold (75); // поріг активності
Set_Inactivity_Threshold (75); // поріг неактивності
Set_Time_Inactivity (10);
// How many seconds of no activity is inactive?
Set_Tap_Threshold (50); // 62.5 mg per increment
Set_Tap_Duration (15); // 625 us per increment
Set_Tap_Latency (80); // 1.25 ms per increment
Set_Tap_Window (200); // 1.25 ms per increment
Get_Source_Interrupt (FREE_FALL); // аналіз переривань - (FREE_FALL)...
writeTo(ADXL345_REG_POWER_CTL, 0x08); // Measure =1
readFrom(ADXL345_REG_DEVID, 1, _buff);
// читати id - один байт в буфер
id=_buff[0];
if(id!=0xE5) // Check connection
{
  Serial.println("ADXL345 error ");
// вивід в послідовний порт інф. про помилку
  lcd.setCursor(2, 0); lcd.print("ADXL345 error ");
  звук(); // включити пищик
  pinMode(RED, OUTPUT); // sets the digital pin as output
  digitalWrite(RED, HIGH); // turns the RED LED - on ТРЕВОГА
  delay(500);
  pinMode(SPEAKER, INPUT);
// sets the digital pin as input - звук off
  digitalWrite(RED, LOW); // turns the RED LED - off ТРЕВОГА
}
else {
Serial.println( id,HEX ); lcd.setCursor(2, 0); lcd.print(id,HEX); // вивести ID акселерометра на
дисплей
  delay(1000); }; }

void loop() {
  аку_test(); // тестування заряду акумулятора

```

```

Get_Data (3); // читати прскорення по осях x/y/z
  xa= (axisX*gainX)*(k_fr/100);
// обчислення прскорення по осі x - *9.81 м/с2
  ya= (axisY*gainY)*(k_fr/100);
// обчислення прскорення по осі y - *9.81 м/с2
  za= (axisZ*gainZ)*(k_fr/100);
// обчислення прскорення по осі z - *9.81 м/с2
  axyz = abs(sqrt(xa*xa+ya*ya+za*za)-daxyz);
// беремо абсолютне значення
  lcd.setCursor(0,1);  lcd.print("      ");
  lcd.setCursor(0,1);  lcd.print(xa,1);
// вивід прскорення по осі X
  lcd.setCursor(6,1);  lcd.print(ya,1);
// вивід прскорення по осі Y
  lcd.setCursor(11,1); lcd.print(za,1);
// вивід прскорення по осі Z
  lcd.setCursor(0,0);  lcd.print("      ");
  lcd.setCursor(0,0);
  lcd.print(axyz,2);
// вивід результуючого вектора прскорення за мінусом
//прскорення вільного падіння
  lcd.setCursor(6,0);
  lcd.print("Ad=");
  lcd.print(Ad,2); // вивід допустимого прскорення Ad
  if (axyz>=Ad)
  {
    Serial.println( " ALARM !!!" );
    //lcd.clear();
    lcd.setCursor(0,1); lcd.print("      ");
    lcd.setCursor(0,1); lcd.print("ALARM !!!");
    zvyk(); // включити пищик
    digitalWrite(RED, HIGH); // turns the RED LED - on ТРИВОГА
    delay(500);
    pinMode(SPEAKER, INPUT);
// sets the digital pin as input - zvyk off
    digitalWrite(RED, LOW); // turns the RED LED - off ТРИВОГА
    delay(500);
  };
  Serial.print("x: "); Serial.print(xa,1);
  Serial.print(" y: "); Serial.print(ya,1);
  Serial.print(" z: "); Serial.print(za,1);
  Serial.print(" Аxyz: ");
  Serial.println( axyz,2 ); // вивід значення прскорення в термінал
  val = digitalRead(BUTTON1); // читати значення клавiшi
  delay(50);
  if ((val == HIGH) && (old_val == LOW)) // аналіз натискань клавiшi
  {
    Ad = Ad - krok; // зменшити на крок
    if (Ad<=0){ Ad=0; }; // мінімальне допустиме значення рівне нулю
  };
  old_val = val; // val is now old, let's store it
  val2 = digitalRead(BUTTON2); // читати значення клавiшi
  delay(50);

```

```

if ((val2 == HIGH) && (old_val2 == LOW)) // аналіз натискань клавiші
{
  Ad = Ad + krok;    // збільшити на крок
  if (Ad>=4){ Ad=4; }; // максимально допустиме значення рівне 4
};
old_val2 = val2; // val is now old, let's store it
delay(50);      // затримка
}
void readAccel()
// функція читає шість значень з регістрів даних ADXL в масив _buff
//та формує значення int прискорень
{
  uint8_t howManyBytesToRead = 6;
  readFrom( ADXL345_REG_DATAX0, howManyBytesToRead, _buff);
//read the acceleration data from the ADXL345
// thus we are converting both bytes in to one int
  X = (((int)_buff[1]) << 8) | _buff[0];
  Y = (((int)_buff[3]) << 8) | _buff[2];
  Z = (((int)_buff[5]) << 8) | _buff[4];
}
void writeTo(byte address, byte val)
{
  Wire.beginTransmission(DEVICE); // start transmission to device
  Wire.write(address);           // send register address
  Wire.write(val);               // send value to write
  Wire.endTransmission();        // end transmission
}
/* Ініціалізація ADXL345 arduino */
void ADXL345_Init (void) {
  writeTo(ADXL345_REG_POWER_CTL, 0x00); // standby mode =0
  delay(20);
//Put the ADXL345 into Measurement Mode by writing 0x08 to the POWER_CTL
  writeTo(ADXL345_REG_POWER_CTL, 0x08); // Measure =1
  delay(20);
}
// Reads num bytes starting from address register on device
// in to _buff array
void readFrom(byte address, int num, byte _buff[]) {
  Wire.beginTransmission(DEVICE); // start transmission to device
  Wire.write(address);           // sends address to read from
  Wire.endTransmission();        // end transmission
  Wire.beginTransmission(DEVICE); // start transmission to device
  Wire.requestFrom(DEVICE, num); // request 6 bytes from device
  int i = 0;
  while(Wire.available()) // device may send less than requested
  { _buff[i] = Wire.read(); // receive a byte
    i++; }
  Wire.endTransmission();        // end transmission
}
//      Отримання даних вимірювання
//      (середнє арифметичне з num_sample значень)
//=====
void Get_Data (char num_sample) {

```

```

char x; axisX=axisY=axisZ=0;
for(x=0; x<num_sample; x++) {
    readAccel(); delay(10); axisX += X; axisY += Y; axisZ += Z;
}
axisX = axisX /num_sample; axisY = axisY /num_sample; axisZ = axisZ /num_sample;
}
// Встановлення порогу Неактивності set_Inactivity_Threshold(75);
//=====================================================
void Set_Inactivity_Threshold (unsigned char data) {
    writeTo(ADXL345_REG_THRESH_INACT, data); // 0x25
    delay(10);
}
// Встановлення часу Неактивності set_Time_Inactivity(10);
//=====================================================
void Set_Time_Inactivity (unsigned char data) // Скільки секунд неактивності?
{
    writeTo(ADXL345_REG_TIME_INACT, data);
    delay(10); }
// Встановлення порогу Активності setActivityThreshold(data);
//=====================================================
void Set_Activity_Threshold (unsigned char data)
// поріг активності 62.5mg per increment
{ writeTo(ADXL345_REG_THRESH_ACT, data); // (0x24) data =(0..255)= 75
  delay(10); }
// Встановлення тимчасового вікна для другого поштовху (Window)
//=====================================================
void Set_Tap_Window (unsigned char data) {
    writeTo(ADXL345_REG_WINDOW, data); // setDoubleTapWindow(200);
    delay(10);
}
// Встановлення затримки для поштовху (Latent)
//=====================================================
void Set_Tap_Latency (unsigned char data) // 1.25 ms per increment
{ writeTo(ADXL345_REG_LATENT, data); // setDoubleTapLatency(80);
  delay(10);}
// Встановлення ліміту часу для поштовху (DUR) setTapDuration(15);
//=====================================================
void Set_Tap_Duration (unsigned char data) // 625 us per increment
{ writeTo(ADXL345_REG_DUR, data); delay(10); }
// Встановлення порогу поштовху setTapThreshold(50);
//=====================================================
void Set_Tap_Threshold (unsigned char data) // 62.5 mg per increment
{ writeTo(ADXL345_REG_THRESH_TAP, data); delay(10); }
// функ - вид функції переривання Функція повертає "1", якщо
// переривання відбулося від опитуваного джерела, інакше - "0"
//=====================================================
char Get_Source_Interrupt (INT_Func func) {
    char data=0;
    //читання регістра INT_SOURCE
    readFrom(ADXL345_REG_INT_SOURCE, 1, _buff);
// читати - один байт в буфер
    data = _buff[0];
    if(data & (1<<func)) return 1;
}

```

```
    return 0; }  
// Звук ШІМ  
void zvyk(void) // функція генерування звуку  
{ pinMode(SPEAKER, OUTPUT); // sets the digital pin as output  
  i=127; // duty= 50%  
  analogWrite(SPEAKER, i); // speaker on, speaker- pin.  
}  
// тест заряду акумулятора  
void aku_test(void) {  
  akumulator = analogRead(1); // analog Read-pin A1- напруга акумулятора  
  n=(akumulator*100)/1024; // напруга на акумуляторі*10  
  if (n<=AKUMIN) {  
    lcd.setCursor(15, 0); // встановити курсор в позицію X=15, Y=1  
    lcd.print("+");  
    Serial.println( " AKUMULATOR - LOW LEVEL" );  
  } else {  
    lcd.setCursor(15, 0); // встановити курсор в позицію X=15, Y=1  
    lcd.print(" ");  
  }  
}
```