

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

Кафедра інформаційних технологій

**РОЗРОБЛЕННЯ ПРОГРАМНОГО І АПАРАТНОГО
ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО КРОКОМІРА НА ПЛАТФОРМІ
ARDUINO І МЕМС-АКСЕЛЕРОМЕТРИ ДЛЯ КОНТРОЛЮ ФІЗИЧНОЇ
АКТИВНОСТІ ПІД ЧАС ТРЕНУВАНЬ ПОЛІЦЕЙСЬКИХ**

Кваліфікаційна робота
здобувача вищої освіти
4 курсу денної форми навчання
Назарій МАРИНЯК

Науковий керівник:
доцент, кандидат технічних наук
Андрій ГОЛОВАТИЙ

Рецензент:
доцент, кандидат технічних наук

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

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

Львів
2026

СПИСОК УМОВНИХ СКОРОЧЕНЬ

ВС (вбудована система) – спеціалізована комп'ютерна система, інтегрована в пристрій для виконання обмеженого кола задач керування.

МК (мікроконтролер) – малогабаритна мікросхема, що поєднує процесор, пам'ять та периферійні інтерфейси на одному кристалі.

МЕМС (MEMS) – мікроелектромеханічні системи, що об'єднують мікроелектронні та мікромеханічні компоненти в єдиному корпусі.

АЗ (апаратне забезпечення) – сукупність фізичних електронних компонентів та вузлів системи.

ПЗ (програмне забезпечення) – набір алгоритмів та інструкцій для керування апаратною частиною пристрою.

САПР (система автоматизованого проектування) – програмний комплекс для розробки та моделювання електронних схем.

AVR – архітектура 8-розрядних RISC-мікроконтролерів (Atmel/Microchip).

ЦП (цифровий пристрій) – технічна система, що обробляє інформацію у дискретному (цифровому) вигляді.

ЕК (електрична схема) – графічне представлення з'єднань між електронними компонентами системи.

ПЗП (ROM/Flash) – енергонезалежна пам'ять, призначена для зберігання виконуваного коду програми та констант.

ОЗП (RAM) – оперативна пам'ять для тимчасового зберігання даних під час виконання обчислень.

EEPROM – тип енергонезалежної пам'яті з можливістю багаторазового електричного перепрограмування для зберігання налаштувань.

I2C (Inter-Integrated Circuit) – послідовна шина даних, що використовує дві лінії (SDA та SCL) для зв'язку між МК та сенсорами на коротких відстанях.

SPI (Serial Peripheral Interface) – високошвидкісний синхронний інтерфейс для повнодуплексної передачі даних між МК та периферією.

РКД (LCD) — рідкокристалічний дисплей для візуального відображення текстової або графічної інформації користувачеві.

АНОТАЦІЯ

Мариняк Н., Головатий А. (керівник). Розроблення програмного і апаратного і забезпечення цифрового крокоміра на платформі Arduino і МЕМС-акселерометрі для контролю фізичної активності під час тренувань поліції. Бакалаврська кваліфікаційна робота. – Львівський державний університет внутрішніх справ, Львів, 2026.

У кваліфікаційній роботі реалізовано цифровий крокомір для контролю фізичної активності під час тренувань поліції. Пристрій побудовано на мікроконтролерній платформі Arduino Uno та 3-х осьовому акселерометрі ADXL345. Пристрій дозволяє здійснювати моніторинг фізичної активності поліцейського, обчислюючи кількість кроків, подолану дистанцію та обсяг витрачених калорій з урахуванням індивідуальних антропометричних даних.

Поліцейському доступна персоналізація профілю через введення антропометричних даних. Крім того, передбачена можливість конфігурації сповіщень: можна встановити звукові та графічні індикатори, які спрацюватимуть при виконанні плану з енерговитрат або подоланої відстані.

Пристрій підтримує подвійний вивід даних: на символічний РКД та через послідовний інтерфейс. Перелік інформації, що відображається, включає: оперативні показники: прогрес ходьби, спалені калорії та активний режим; конфігурація: параметри поліцейського (вага/зріст) та задані порогові значення для сповіщень; сервісні сигнали: попередження про досягнення встановлених обмежень. Спроектовано принципову електричну схему та створено цифрову модель крокоміра. Процес розробки включав написання програмного коду на мові C (Arduino IDE) та побудову алгоритму функціонування. Працездатність пристрою підтверджена шляхом симуляції в середовищі Proteus VSM (ISIS), а також через тестування реального прототипу (макету).

Ключові слова: цифровий педометр, вимірювач активності, апаратна платформа Arduino, рідкокристалічний індикатор (РКІ), мікроконтролери сімейства AVR, комп'ютерне моделювання в Proteus VSM, мікропрограмне забезпечення, інструментарій Arduino IDE.

ABSTRACT

Marynyak N., Holovatyy A. (supervisor). Development of software and hardware for a digital pedometer on the Arduino platform and MEMS accelerometer for monitoring physical activity during police training. Bachelor's thesis. – Lviv State University of Internal Affairs, Lviv, 2026.

In the qualification work, a digital pedometer was implemented to monitor physical activity during police training. The device is built on the Arduino Uno microcontroller platform and the ADXL345 3-axis accelerometer. The device allows monitoring the physical activity of a police officer, calculating the number of steps, distance traveled, and calories burned, taking into account individual anthropometric data.

The police officer can personalize his profile by entering anthropometric data. In addition, there is the possibility of configuring notifications: you can set sound and graphic indicators that will be triggered when the plan for energy consumption or distance traveled is fulfilled.

The device supports dual data output: to the symbolic LCD and via a serial interface. The list of displayed information includes: operational indicators: walking progress, calories burned and active mode; configuration: policeman parameters (weight/height) and set threshold values for notifications; service signals: warnings about reaching set limits.

A schematic electrical circuit was designed and a digital model of the pedometer was created. The development process included writing the program code in C language (Arduino IDE) and building the functioning algorithm. The device's performance was confirmed by simulation in the Proteus VSM (ISIS) environment, as well as by testing a real prototype (mockup).

Keywords: digital pedometer, activity tracker, Arduino hardware platform, liquid crystal display (LCD), AVR microcontrollers, computer simulation in Proteus VSM, firmware, Arduino IDE toolkit.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....	10
1.1. Огляд обладнання для вимірювання дистанції та основна типологія крокомірів.....	10
1.2. Класифікація та характеристика методів та засобів розрахунку пройденої відстані.....	17
РОЗДІЛ 2. ЗАСОБИ РОЗРОБКИ АПАРАТНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО КРОКОМІРА ДЛЯ КОНТРОЛЮ ФІЗИЧНОЇ АКТИВНОСТІ ПІД ЧАС ТРЕНУВАНЬ ПОЛІЦЕЙСЬКИХ....	25
2.1. Програмний комплекс для проектування та імітаційного моделювання цифрових пристроїв Proteus VSM.....	25
2.2. Середовище Arduino IDE для програмування систем на мікроконтролерах AVR.....	27
РОЗДІЛ 3. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО КРОКОМІРА ДЛЯ КОНТРОЛЮ ФІЗИЧНОЇ АКТИВНОСТІ ПІД ЧАС ТРЕНУВАНЬ ПОЛІЦЕЙСЬКИХ.....	35
3.1. Вибір елементної бази для проектування цифрового крокоміра.....	35
3.2. Проектування цифрового крокоміра для контролю фізичної активності.....	53
РОЗДІЛ 4. ПРОГРАМНО-АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО КРОКОМІРА ДЛЯ КОНТРОЛЮ ФІЗИЧНОЇ АКТИВНОСТІ ПІД ЧАС ТРЕНУВАНЬ ПОЛІЦЕЙСЬКИХ.....	58
4.1. Алгоритм функціонування цифрового крокоміра для контролю фізичної активності.....	58
4.2. Реалізація програмного інтерфейсу для роботи з датчиком прискорення ADXL345.....	69
4.3. Розроблення програмних модулів режимів роботи крокоміра.....	69
4.4. Розроблення головного програмного модуля функціонування цифрового крокоміра для контролю фізичної активності.....	70
4.5. Дослідження моделі цифрового крокоміра для контролю фізичної активності в Proteus ISIS.....	71
ВИСНОВКИ.....	83
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	85
ДОДАТКИ.....	88

ВСТУП

Використання сучасних малогабаритних акселерометрів у поєднанні з мікроконтролерами дозволяє проектувати доступні крокоміри з високою точністю обробки даних. Реалізація подібного функціонала на платформі Arduino забезпечує комплексний моніторинг фізичної активності: від простого підрахунку кроків до калькуляції спалених калорій та пройденого шляху.

Цифровий крокомір розробляється для моніторингу фізичних навантажень особового складу поліції під час тренувального процесу. Основні завдання включають реалізацію алгоритмів детекції кроків, обчислення енерговитрат користувача, впровадження системи сповіщень про досягнення цільових показників та створення адаптивного інтерфейсу для персоналізації параметрів роботи.

Пристрій передбачає візуалізацію даних на РК-дисплеї та передачу їх через послідовний інтерфейс для подальшого моніторингу. Спроектований пристрій є універсальним рішенням, що може бути впроваджене для контролю фізичної підготовки курсантів та діючих співробітників поліції. Крім того, прилад має високу практичну цінність для медичної галузі, зокрема в програмах реабілітації пацієнтів та терапії осіб із порушеннями метаболізму (надмірною вагою).

Сучасні MEMS-акселерометри дозволяють проектувати точні вимірювальні прилади для визначення прискорення та сили механічного впливу. Завдяки надмалим розмірам та здатності працювати з різними типами прискорень (статичними й динамічними), вони є оптимальними для компактної електроніки. Обмін інформацією та керування режимами роботи пристрою здійснюється за допомогою стандартного протоколу I2C.

Апаратне забезпечення крокоміра реалізовано на базі Arduino з використанням восьмибітних мікроконтролерів AVR. Дана платформа забезпечує розробника необхідним набором інтерфейсів, таких як UART та USB, для зручної передачі даних та програмування. Завдяки модульній структурі МК (наявності вбудованої флеш-пам'яті та портів загального призначення), стає

можливим створення компактних енергоефективних пристроїв без потреби у значній кількості зовнішніх компонентів.

Мета роботи полягає у розробці апаратно-програмного комплексу цифрового педометра, адаптованого для моніторингу фізичних навантажень особового складу поліції. Для досягнення мети необхідно вирішити наступні **завдання**:

- спроектувати архітектуру пристрою на базі контролера Arduino UNO (мікроконтролер ATmega328P) та триосьового акселерометра ADXL345 із використанням символного РК-індикатора типу 16x2;
- здійснити схемотехнічне моделювання цифрового педометра в середовищі САПР Proteus VSM;
- реалізувати програмний інтерфейс для коректної взаємодії з MEMS-сенсором ADXL345;
- сформулювати логічну структуру та алгоритм функціонування крокоміра на платформі Arduino.
- написати вбудоване програмне забезпечення мовою C/C++ у середовищі Arduino IDE відповідно до розробленого алгоритму;
- провести верифікацію та аналіз роботи віртуального прототипу в емуляторі Proteus ISIS, підтвердивши відповідність пристрою технічним вимогам.

Об'єкт дослідження – процес розробки та апаратно-програмної реалізації цифрового педометра на базі мікроконтролерної платформи Arduino та сенсора ADXL345, орієнтованого на моніторинг інтенсивності фізичних навантажень особового складу поліції.

Предмет дослідження – сукупність методів моделювання, технічних засобів та алгоритмічних рішень, необхідних для розробки цифрового педометра на базі контролера Arduino та акселерометра ADXL345, призначеного для оцінки рівня фізичного навантаження працівників поліції.

Методи дослідження. Для досягнення поставленої мети та розв'язання визначених завдань застосовано комплекс наукових методів пізнання. Зокрема, бібліографічний аналіз дозволив вивчити стан сучасних розробок у галузі

МЕМС-технологій та мікроконтролерних систем. Застосування діалектичного та системного підходів сприяло всебічному розгляду функціональних можливостей педометрів. Використання низки загальнонаукових методів, таких як аналіз, синтез, індукція та дедукція, забезпечило логічну послідовність розробки, а метод комп'ютерного моделювання став основою для перевірки функціонування спроектованої системи в середовищі Proteus.

Структура роботи. Кваліфікаційна робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел, додатків. Обсяг основного тексту роботи складає 83 сторінки, 48 рисунків, 9 таблиць, 2 додатки і 22 бібліографічних джерела. Загальний обсяг роботи – 90 сторінок.

РОЗДІЛ 1

СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1. Огляд обладнання для вимірювання дистанції та основна типологія крокомірів

Пристрої, що призначені для фіксації кількості кроків, пройшли еволюційний шлях від простих механічних конструкцій до складних цифрових систем. На сьогодні виділяють такі основні види педометрів: механічні: працюють за принципом маятника, що замикає контакти при кожному русі; електронно-механічні: комбінують механічний датчик із цифровим дисплеєм для відображення результатів; цифрові (MEMS): базуються на багатоосьових акселерометрах, що забезпечують найвищу точність вимірювань.

Сучасна цивілізація створила умови для “енергетичного голоду” організму: через комфортний побут та доступний транспорт ми витрачаємо критично мало енергії (в межах 2000–2500 ккал на добу). Це провокує справжню епідемію неінфекційних захворювань – від діабету до серцевої недостатності. Ожиріння перестало бути проблемою лише дорослих; сьогодні воно загрожує майже мільярду людей у всьому світі, включаючи дітей. Відсутність природної рухової активності (бігу, ігор, плавання), яка має займати п’яту частину часу дитини, закладає підґрунтя для серйозних проблем зі здоров’ям майбутніх поколінь.

Проблема ожиріння часто криється у “непомітному” профіциті енергії. Щоденне вживання лише двох зайвих шматочків цукру (близько 40 ккал) здатне збільшити вагу тіла на 1 кг протягом року. Протягом життєвого циклу цей показник зростає пропорційно: за 20 років людина може отримати до 20 кг зайвої маси. Такий поступовий приріст є критичним, оскільки він створює значне навантаження на організм і стає фундаментом для патологій серцево-судинної системи та опорно-рухового апарату.

Регулярне фізичне навантаження виступає ефективним інструментом контролю маси тіла навіть за умов висококалорійного раціону. Згідно з фізіологічними нормами, працівникам розумової праці або сфер із дефіцитом

рухової активності необхідно щоденно долати дистанцію у 6–8 км. Це мінімально достатній рівень активності для нівелювання негативних наслідків гіподинамії.

Регулярна локомоторна активність, зокрема ходьба, виступає потужним превентивним засобом проти низки захворювань. Досягнення встановлених добових цілей щодо кількості кроків дозволяє оптимізувати метаболічні процеси, підтримувати мінеральну щільність кісток та покращувати ментальне здоров'я. Крім того, активна ходьба активізує нейропластичність мозку, що безпосередньо впливає на працездатність та психоемоційний стан людини.

Популярність стратегії “10 000 кроків щодня” зумовлена її простотою та ефективністю у стимулюванні регулярних тренувань. Встановлені медичні нормативи передбачають щонайменше 150 хвилин активності на тиждень (розподілених по 30 хвилин протягом 5 днів). Використання цифрових педометрів для фіксації десятитисячного рубежу кроків дозволяє користувачеві об'єктивно оцінювати відповідність свого способу життя загальноприйнятим стандартам здоров'я.

Норматив у 10 000 кроків (що становить близько 8,05 км) має історичне коріння, що сягає Японії другої половини ХХ століття. Попри своє емпіричне походження, ця концепція пройшла верифікацію сучасними клінічними дослідженнями. Сьогодні встановлено пряму кореляцію між щоденним виконанням цієї норми та зниженням індексу маси тіла, а також загальним зміцненням фізіологічних показників здоров'я користувача.

Для математичного моделювання енерговитрат у вбудованому ПЗ крокоміра необхідно враховувати фізичний еквівалент калорії. Енергія, що витрачається при ходьбі, корелює з термічними одиницями: 1 ккал дорівнює 4,1868 кДж. Використання джоуля як одиниці роботи (Вт·с) дозволяє інтегрувати показники потужності руху в загальну систему аналізу активності користувача.

Визначення калорії базується на властивостях питомої теплоємності води. Згідно з цим підходом, одна калорія еквівалентна обсягу теплової енергії, що

витрачається на підвищення температури 1 г води на 1°C за умови нормального атмосферного тиску (101 325 Па).

Під калоріями розуміють енергетичну цінність раціону. Стабільна вага можлива лише за умови енергетичної рівноваги; у разі ж позитивного енергобалансу (надлишкового надходження палива) організм трансформує невичерпану енергію у жирову тканину.

Для встановлення калорійності різних продуктів використовують методику прямої калориметрії. Вона полягає в спалюванні харчового продукту всередині ізольованого приладу (калориметра) для фіксації теплової енергії, що вивільняється. Цей процес моделює енергетичний обмін у людському тілі, де калорії стають паливом для метаболізму, фізичної праці та збереження стабільної температури тіла.

Як підрахувати енергію, яку ми спалюємо під час роботи? Один із варіантів – помістити людину у масштабний калориметр для вимірювання теплопродукції тіла. Оскільки це не завжди зручно, науковці розробили альтернативні способи. Наприклад, за кількістю використаного кисню та виробленого CO₂ можна точно визначити інтенсивність спалювання “палива” в організмі. Навіть звичайний пульсометр може стати інструментом для такого розрахунку. Важливо також пам’ятати про таймінг: левову частку раціону (понад 80%) варто з’їдати до обіду, щоб енергія використовувалася максимально ефективно.

Для наочності дані про енерговитрати при різних видах діяльності представлено у Табл.1.1.

Таблиця 1.1. Енерговитрати при різних видах діяльності

Вид активності	Інтенсивність	Витрати енергії (ккал/год)
Біг по сходах	Вгору	900
Спринт	16 км/год	750
Велоспорт	20 км/год	540
Бігові лижі	Середній темп	485
Плавання	Швидке (2,4 км/год)	460

Скільки енергії ми витрачаємо на дистанцію в 10 000 кроків? Універсальної відповіді не існує, адже кожен організм працює по-різному. Кількість “спаленого палива” залежить від вашої ваги, зросту, швидкості руху та

навіть того, чи йдете ви вгору, чи по рівній дорозі. Традиційно вважалося, що головними факторами є фізичні параметри тіла, проте інтенсивність і умови навколишнього середовища мають не менше значення для фінального результату.

Хоча фізика стверджує, що більша маса потребує більше палива для руху, біологія людини працює інакше. Новітні дослідження виявили наступне: традиційний погляд: висока вага = високі енерговитрати; нові дані: при порівнянні питомих витрат енергії (з поправкою на масу), люди з більшою вагою не обов'язково спалюють більше калорій; причина: ймовірні адаптивні реакції обміну речовин на тривалу фізичну активність.

Окрім фізичних параметрів людини, на розхід енергії впливають швидкість руху та тип поверхні. Ходьба під нахилом – чудовий приклад: при швидкості 8 км/год: спалюється понад 7 ккал/хв; при швидкості 5–6 км/год: показник варіюється в межах 3,5–7 ккал/хв. Отже, навіть незначне збільшення темпу на складному рельєфі може майже вдвічі підвищити ефективність тренування.

Енергетична різниця при однаковій дистанції (10 000 кроків): швидкий темп (6 км/год): додатково спалює ~153 ккал; повільний темп (3 км/год): базовий рівень енерговитрат; висновок: збільшення швидкості вдвічі суттєво підвищує метаболічний відгук на тій самій відстані.

Хоча середня норма спалювання енергії на 10 000 кроків складає 400 ккал, цей розрахунок залежить від багатьох змінних: фізичні дані: індивідуальний зріст та вага людини; метаболізм: швидкість обміну речовин, що є унікальною для кожного організму; генетика та фізіологія: навіть ідентичні близнюки демонструють різні результати енерговитрат при однакових активностях.

Орієнтовна енергетична вартість одного кроку становить близько 0,04 калорії. Для ефективного схуднення варто дотримуватися правила енергетичного дефіциту: зменште свій добовий раціон на 500 калорій відносно рівня підтримки ваги.

Процес корекції ваги має базуватися на принципах безпеки та індивідуального розрахунку: для набору ваги: створіть профіцит у 500 ккал на

день; межа безпеки: не знижуйте калорійність раціону більш ніж на 1200 ккал від вашої базової потреби; темп змін: фізіологічно здоровою нормою вважається втрата 0,8–1 кг за тиждень; розрахунок: використовуйте спеціальні сервіси для обчислення витрат енергії, базуючись на вашому способі життя (малорухливий, активний чи дуже активний).

Оцінити рівень енергетичних витрат при ходьбі допомагають методики, засновані на обліку персональних показників. Зокрема, існують табличні дані, що пов'язують фізичну активність людини з її масою тіла та зростом.

Визначення дистанції за допомогою крокоміра базується на середніх показниках довжини кроку, що корелюють зі зростом людини: понад 182 см: крок – 0,8 м; від 167 до 180 см: крок – 0,73 м; до 165 см: крок – 0,67 м. Додатково в розрахункових таблицях (1.2–1.4) враховано фактор маси тіла, який варіюється навіть серед людей однакового зросту.

Таблиця 1.2. Визначення витрат енергії (зріст ≥ 182 см) Дані, подані нижче, дозволяють оцінити кількість спалених калорій для користувачів, чий зріст становить 182 см або вище, залежно від їхньої ваги та активності

Кроки	ВАГА (кг)								
	45	55	64	73	82	91	100	114	125
1000	28 кал.	33	38	44	49	55	60	69	75
2000	55	66	76	87	98	109	120	137	150
3000	83	99	114	131	147	164	180	206	225
4000	110	132	152	174	196	218	240	274	300
5000	138	165	190	218	245	273	300	343	375
6000	165	198	228	261	294	327	360	411	450
7000	193	231	266	305	343	382	420	480	525
8000	220	264	304	348	392	436	480	548	600
9000	248	297	342	392	441	491	540	617	675
10000	275	330	380	435	490	545	600	685	750

Таблиця 1.3. Споживання калорій при зрості 167...180 см

Кроки	ВАГА (кг)								
	45	55	64	73	82	91	100	114	125
1000	25 кал.	30	35	40	45	50	55	62	68
2000	50	60	69	79	89	99	109	125	136
3000	75	90	104	119	134	149	164	187	205
4000	100	120	138	158	178	198	218	249	273
5000	125	150	173	198	223	248	273	311	341
6000	150	180	207	237	267	297	327	374	409
7000	175	210	242	277	312	347	382	436	477

8000	200	240	276	316	356	396	436	498	545
9000	225	270	311	356	401	446	491	560	614
10000	250	300	345	395	445	495	545	623	682

Таблиця 1.4. Споживання калорій при зрості 165 см та нижче

Кроки	ВАГА (кг)								
	45	55	64	73	82	91	100	114	125
1000	23 кал.	28	32	36	41	45	50	57	63
2000	46	55	63	73	82	91	100	114	125
3000	69	83	95	109	123	136	150	171	188
4000	92	110	127	145	163	182	200	228	250
5000	115	138	158	181	204	227	250	285	313
6000	138	165	190	218	245	273	300	343	375
7000	160	193	222	254	286	318	350	400	438
8000	183	220	253	290	327	363	400	457	500
9000	206	248	285	326	368	409	450	514	563
10000	229	275	317	363	408	454	500	571	625

Представлені дані базуються на системі MET і враховують середні витрати енергії при пересуванні пішки зі швидкістю 3–6 км/год. Щоб відповідати медичним стандартам фізичної активності, людині варто проходити мінімум 10 000 кроків за добу. Контролювати виконання цієї норми допомагають цифрові або механічні крокоміри.

Педометр (або крокомір) — це портативний пристрій для обліку фізичної активності. Етимологія назви виходить з грецької мови та буквально означає «вимірювач поверхні». Прилад дозволяє точно фіксувати кількість кроків та обчислювати пройдений простір, перебуваючи безпосередньо при користувачеві. Пристрій для моніторингу ходи, відомий як крокомір, може мати як просту механічну конструкцію, так і складну електронну начинку. Сьогодні це не просто лічильник кроків, а мультифункціональний гаджет, що може замінити калькулятор калорій, плеєр або годинник. Понад те, завдяки технологічному прогресу, функціонал крокоміра тепер вбудований безпосередньо у смартфони [2].



Рис. 1.1. Електронний педометр TFA Hitrax Walk (422003)

Цифровий крокомір TFA Nitrah Walk (422003) забезпечує точний підрахунок кроків, дистанції та спалених калорій. Оптимальним місцем для кріплення приладу є поясна зона. Основні характеристики включають: універсальність: вимірювання шляху в км або милях; автономність: функція автовимкнення через 120 секунд простою; ергономічність: надійне кріплення на ремені для правильної роботи сенсорів.



Рис. 1.2. Цифровий крокомір inSPORTline Strippy I.

Модель inSPORTline Strippy II є багатофункціональним цифровим інструментом для моніторингу фізичної активності. Пристрій фіксує не лише кількість кроків та подолану дистанцію, а й обчислює енерговитрати та загальну тривалість ходьби. Однією з ключових переваг є вбудована пам'ять, що зберігає результати за останні 30 днів. Можливість синхронізації з ПК дозволяє користувачеві систематизувати дані, проводити порівняльний аналіз та відстежувати статистику. Додаткову практичність забезпечують легкий вологозахисний корпус із силікону, вбудований будильник та режим звичайного годинника.

1.2. Класифікація та характеристика методів та засобів розрахунку пройденої відстані

Огляд охоплює існуючі інженерні рішення для побудови вимірювальних систем, від класичних крокомірів до сучасних інтегрованих модулів у смартфонах, з аналізом їхньої точності та похибок.

При огляді літературних джерел та інтернет - ресурсів отримано необхідну кількість інформації для розробки педометра (крокоміра).

Попередній огляд спеціалізованих видань та профільних інтернет-майданчиків дозволив акумулювати теоретичну базу, що є достатньою для обґрунтування архітектури та функціоналу розроблюваного педометра.

Більшість актуальних систем моніторингу ходьби побудовані за подібним принципом: вони використовують датчики, що реагують на інерційні зміни та прискорення тіла людини.

Основою вимірювання в крокомірах є фіксація ударного прискорення під час кроку. В застарілих механічних приладах це реалізовано через інерційний механізм з пружиною. Електронні ж версії трансформують кожен струс у цифровий імпульс. Найбільш точними є сучасні пристрої з трьохосьовими сенсорами: вбудовані алгоритми аналізують амплітуду рухів, що дозволяє пристрою відрізнити звичайну ходьбу від випадкових жестів чи нахилів.

Точність роботи крокомірів залежить від низки факторів, а допустимий рівень похибки для таких систем становить до 10%. Можна виділити такі основні чинники, що впливають на якість вимірювань: вартість та складність датчиків: існує пряма залежність між ціною пристрою та його точністю; умови експлуатації: вібрації при русі транспорту по вибоїнах часто сприймаються алгоритмами як фізична активність; антропометричні особливості: специфічний стиль ходьби конкретної людини може змінювати амплітуду прискорень, що фіксуються акселерометром; коректність розміщення: дотримання рекомендацій щодо кріплення (наприклад, на ремені) є критичним для правильних розрахунків. Окремим сегментом ринку є мобільні телефони, де функція педометра реалізована за допомогою вбудованих сенсорів.

Телефон Nokia 5500 Sport використовує тривимірний сенсор прискорення для вимірювання фізичних показників (кроки, відстань, час). Основним недоліком є неможливість цілодобового фонового моніторингу, оскільки робота акселерометра призводить до швидкого розряду акумулятора.

Моделі W710 та W580 від Sony Ericsson використовують двовимірні акселерометри для трекінгу активності. В W710 статистика кроків доступна на зовнішньому екрані при закритій кришці. Система підтримує автоматичне

архівування даних у спеціальний лог-файл кожного дня о 24:00, одночасно з обнуленням денного результату.

Основні недоліки простих моделей крокомірів: відсутність аналізу інтенсивності: прилад не розрізняє швидкий та повільний темп; обмежений функціонал: відсутні алгоритми обчислення енергозатрат (калорій) та перерахунку кроків у метричну відстань.

Наприклад, при однаковій кількості кроків (1000) енерговитрати при русі вгору та по рівній поверхні суттєво відрізняються, проте базовий крокомір не здатний диференціювати ці умови, видаючи однакове підсумкове значення.

Спеціалізовані версії крокомірів відрізняються багатофункціональністю, де набір доступних інструментів моніторингу залежить від цінового сегмента та специфікації обраної моделі.

Функціональні можливості та вимірювальні параметри сучасних педометрів:

- Базові метрики: реєстрація сумарної кількості кроків та обчислення подоланої дистанції за визначений інтервал часу.
- Аналітика енерговитрат: автоматичний розрахунок спалених калорій на основі інтенсивності руху.
- Часові параметри: наявність інтегрованого таймера та секундоміра для моніторингу тривалості фізичних навантажень.
- Управління даними: збереження статистики у внутрішній пам'яті (зазвичай глибиною до 7 діб) та можливість синхронізації з ПК для довгострокового зберігання.
- Навігація та геолокація: використання вбудованих GPS-модулів для трекінгу маршрутів та роботи з електронними картами.
- Додаткові опції: мультимедійні функції (FM-радіо), а також специфічні датчики для оцінки психоемоційного стану користувача.

Архітектура та класифікація крокомірів. Для коректної конфігурації приладу необхідно розібратися у механізмах фіксації фізичної активності. На сьогодні технології підрахунку кроків базуються на трьох ключових підходах:

- Механічний тип крокоміра, який зараз зустрічається вкрай рідко, працює завдяки коливанням внутрішнього вантажу. Пружинний механізм реагує на кожен крок, передаючи імпульс на шестерню. Остання повертає стрілку циферблата, де кожна поділка фіксує один пройдений крок. Ця проста конструкція забезпечує пряму залежність між механічним рухом приладу та відображенням результату.
- В основі роботи електронно-механічного крокоміра лежить чутливий датчик, що реагує на вібрації під час ходьби. Кожен крок створює механічне коливання, яке система миттєво перетворює в електричний сигнал. Цей імпульс передається на електронне табло, де один зареєстрований сигнал прирівнюється до одного зробленого кроку.
- Електронний тип пристрою є еволюційним продовженням попередніх поколінь. Його головна особливість полягає у використанні інтегральних схем і складного програмного забезпечення для аналізу рухів. Завдяки математичним обчисленням такі девайси демонструють високу точність, а флагманські моделі додатково підтримують зв'язок із супутниковими системами навігації для детальної побудови треків.

Сьогоднішні педометри – це складні електронні системи, що використовують датчики типу MEMS (мікроелектромеханічні системи). Вони аналізують прискорення за декількома осями, а спеціальні алгоритми обробляють ці дані для розпізнавання кроків. Використання мікропроцесорної обробки та багатоосьових сенсорів забезпечує високу якість вимірювань, проте зумовлює вищу вартість таких девайсів на ринку.

Щоб уникнути спотворення даних про пройдену дистанцію, особливу увагу слід приділити способу кріплення крокоміра. Невдалий вибір точки розташування на тілі може призвести до значних відхилень у показниках. Найбільш достовірні результати пристрій демонструє при фіксації на ремені в зоні передньої поверхні стегна, де амплітуда рухів є найбільш чіткою та зрозумілою для механізму.

У продажі можна знайти педометри від ключових гравців індустрії. Географія брендів охоплює як азіатських, так і європейських виробників, зокрема:

1. Японську корпорацію Omron.
2. Китайську марку Sigeta.
3. Швейцарський бренд Maniquick.
4. Італійську компанію Kopus.

Для порівняння застарілих та сучасних технологій, нижче представлені зображення класичного механічного приладу та його електронного аналога.



Рис. 1.3. Крокоміри (механічний (зліва) і електронний (справа))

Ефективність роботи крокоміра напряму залежить від його моделі та способу використання. Щоб отримати достовірні дані, прилад слід надійно закріпити у визначеному виробником місці (зазвичай на ремені). Будь-яке відхилення від інструкції може нівелювати точність підрахунку. Пам'ятайте, що пристрої часто сприймають вібрації в автомобілі чи автобусі як ходьбу, тому їх варто вимикати під час поїздок. Не забувайте і про людський фактор: специфічні особливості вашої ходи також можуть коригувати точність вимірювань.

Бренди, що випускають сучасні крокоміри, досі не дійшли згоди щодо того, що саме вважати “правильним кроком”. Як наслідок, користувачі спостерігають розбіжності в цифрах, залежно від обраного гаджета.

Систематичні піші прогулянки є доступним і дієвим інструментом підтримки здоров'я. Оптимальна щоденна активність сприяє профілактиці

депресивних станів, допомагає контролювати масу тіла, стимулює когнітивні функції та зміцнює опорно-руховий апарат. З огляду на комплексну позитивну дію ходьби на якість життя, створення технологічних рішень для її моніторингу є затребуваним і актуальним завданням.

Дана кваліфікаційна робота присвячена розробці апаратно-програмного комплексу цифрового крокоміра на базі мікроконтролера Arduino та тривісного MEMS-акселерометра ADXL345. Пристрій спроектовано для моніторингу фізичних навантажень особового складу поліції. Функціонал системи забезпечує підрахунок кроків, визначення пройденої дистанції та оцінку енерговитрат (калорій). Передбачено механізм сповіщень про досягнення заданих показників, передачу даних через послідовний інтерфейс, а також можливість адаптивного налаштування робочих параметрів педометра.

Функціонал крокоміра дозволяє оперативно змінювати режими роботи та здійснювати моніторинг фізичних навантажень за допомогою символічного РК-дисплея. Пристрій автоматично розраховує пройдену дистанцію та калорії, сповіщаючи про критичні значення звуковими та світловими сигналами. Окрему увагу приділено автономності: користувач може вимикати екран для економії заряду акумулятора.

Запропоноване рішення поєднує низьку собівартість із широкими експлуатаційними можливостями: інтегрованим LCD-екраном, багаторівневим меню та підтримкою різних профілів роботи. Використання внутрішньої пам'яті EEPROM дозволяє зберігати конфігурацію пристрою після вимкнення живлення. Програмне забезпечення передбачає передачу діагностичної інформації в Serial-порт та систему адаптивних попереджень (звукових і світлових) залежно від встановлених лімітів.

Апаратна реалізація крокоміра базується на платформі Arduino, що дозволяє використовувати відкрите програмне забезпечення та стандартизовані бібліотеки Arduino IDE. Це забезпечує надійну взаємодію з датчиками через готові програмні інтерфейси, мінімізуючи витрати часу на написання низькорівневого коду для драйверів.

Трьохосьові акселерометри на базі МЕМС-технологій застосовуються для моніторингу фізичної активності, зокрема для ідентифікації кроків шляхом аналізу динамічного прискорення.

Параметри моделі ADXL345 (13 біт при шкалі +/-16g) гарантують чутливість близько 0,002g на одиницю молодшого розряду. Такої деталізації даних вистачає, щоб ефективно детектувати динамічні навантаження, характерні для ходьби.

Через механічну природу МЕМС-сенсорів статичні чинники провокують деградацію їхньої чутливості та виникнення адитивної похибки (зміщення). Як наслідок, точність визначення прискорень знижується. Корекція цих помилок реалізується шляхом калібрування чутливості та “нуля” акселерометра.

Для підвищення точності роботи ADXL345 понад заводські стандарти необхідно виконати індивідуальне калібрування пристрою. Суть методу полягає в обчисленні корегувальних значень для кожної координатної осі. Отримані коефіцієнти застосовуються до вихідних сигналів для отримання достовірних значень прискорення вільного падіння, де базовою одиницею є g (9,8 м/с²).

Алгоритм калібрування полягає в отриманні шести екстремальних значень прискорення шляхом послідовної орієнтації датчика в просторі. Отримані дані (мінімальні та максимальні значення по кожній осі) використовуються як вхідні параметри для обчислення поправок, що мінімізують похибку вимірювань.

Для обчислення калібрувальних коефіцієнтів (зміщення та чутливості) необхідно отримати вихідні дані в шести положеннях пристрою. Прийmemo за приклад наступні виміряні значення для ADXL345:

Вісь	Макс. (+)	Мін. (-)
X	+257	-271
Y	+232	-298
Z	+219	-290

Для корекції роботи акселерометра необхідно перевести “сирі” дані у фізичні величини, використовуючи обчислені коефіцієнти.

1. Зміщення (Offset): компенсує асиметрію показників щодо нуля. Як видно з розрахунків, найбільший зсув спостерігається по осі Z (-35,5 LSB).

2. Масштабний коефіцієнт (Sensitivity): відображає відхилення реальної чутливості від еталонної (256 LSB для 1g). Отримані значення (наприклад, 1,031 для осі X) будуть використані для нормування вихідного сигналу.

Розрахунки базуються на константі $D=256$, що є характерним для режиму $\pm 2g$ при роботі з 10-бітною сіткою значень (де повна шкала охоплює 512 одиниць на кожне g).

Оскільки кожен MEMS-сенсор має унікальні апаратні похибки, процедура обчислення коефіцієнтів є обов'язковою для кожного пристрою. Показником успішного калібрування є отримання значень ± 256 при повторній перевірці орієнтації осей. Підсумкові значення прискорення з урахуванням внесених поправок обчислюються за формулою $A_r = (A - Offset)/Sensitivity$, що дає наступні результати для всіх осей:

1. Вісь X: $(257+7)/1,031=256$; $(-271+7)/1,031=-256$.
2. Вісь Y: $(232+33)/1,035=256$; $(-298+33)/1,035=-256$.
3. Вісь Z: $(219+35,5)/0,994=256$; $(-290+35,5)/0,994=-256$.

Ідеальний збіг отриманих даних із цільовим показником ± 256 свідчить про успішне усунення апаратних похибок (зміщення та неточності масштабування) датчика ADXL345.

Кінцевим етапом обробки сигналу є конвертація дискретних значень сенсора у метричні одиниці прискорення. Математична модель для кожної координатної осі виглядає так:

- Вісь X: $A_x = \frac{(X - X_{offset}) \cdot k}{X_{sens}}$, $A_x = (X - X_{offset}) \cdot k / X_{sens}$
- Вісь Y: $A_y = \frac{(Y - Y_{offset}) \cdot k}{Y_{sens}}$, $A_y = (Y - Y_{offset}) \cdot k / Y_{sens}$
- Вісь Z: $A_z = \frac{(Z - Z_{offset}) \cdot k}{Z_{sens}}$, $A_z = (Z - Z_{offset}) \cdot k / Z_{sens}$

де $k=0,0383$ – масштабувальний множник для переходу до одиниць m/s^2 .
Перевірка розрахунків показує, що для максимального значення $X=257$ система реєструє прискорення $9,8 m/s^2$, підтверджуючи високу точність налаштування.

Принцип дії крокометра на ADXL345 базується на аналізі сумарного вектора прискорення за всіма осями. Система порівнює ці дані із заданим порогом для розпізнавання кроку. Завдяки алгоритму вибору осі з найбільшою амплітудою, педометр коректно працює при будь-якій орієнтації корпусу. Обов'язковим етапом перед початком підрахунку є процедура калібрування при ввімкненні пристрою.

РОЗДІЛ 2

ЗАСОБИ РОЗРОБКИ АПАРАТНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО КРОКОМІРА ДЛЯ КОНТРОЛЮ ФІЗИЧНОЇ АКТИВНОСТІ ПІД ЧАС ТРЕНУВАНЬ ПОЛІЦЕЙСЬКИХ

2.1. Програмний комплекс для проєктування та імітаційного моделювання цифрових пристроїв Proteus VSM

Програмний комплекс Proteus VSM – це універсальна САПР для створення електричних схем. Програма дозволяє моделювати роботу пристроїв на базі платформ Arduino, а також численних сімейств мікропроцесорів та мікроконтролерів [3].

Proteus VSM від британського розробника Labcenter Electronics – це потужна платформа для імітаційного моделювання електроніки. Завдяки розгалуженій базі віртуальних моделей, програма дозволяє досліджувати роботу схем на базі мікроконтролерів, Arduino та сигнальних процесорів. Користувачі можуть не лише проєктувати топологію кола, а й здійснювати покрокову відладку ПЗ, застосовуючи широкий набір віртуального вимірювального обладнання.

Окрім засобів розробки схем, Proteus пропонує модуль тривимірного моделювання для візуального контролю компонування друкованих плат. Програмне забезпечення дозволяє проводити верифікацію та симуляцію коду для різноманітних мікропроцесорних родин, від класичних 8051 та PIC до потужних ARM7 та контролерів AVR.

Внутрішній каталог Proteus містить понад 6 тисяч моделей радіоелектронних компонентів, охоплюючи широкий спектр мікроконтролерів (зокрема ARM, AVR, PIC). Важливою перевагою є підтримка SPICE-параметрів: якщо необхідна деталь відсутня в базовій бібліотеці, її можна додати самостійно або скопіювати офіційну модель із сайту виробника, що забезпечує максимальну відповідність віртуальної схеми реальному прототипу.

Програма підтримує роботу з бінарними файлами багатьох компіляторів, що спрощує процес тестування прошивок. У Proteus VSM реалізовано функцію

паралельного моделювання декількох мікроконтролерів, що дає змогу налагоджувати складні пристрої зі змішаною архітектурою в єдиному віртуальному просторі.

Слід розуміти, що жодна симуляція не гарантує 100% відповідності реальним умовам експлуатації. Тим не менш, для етапу розробки та дебагінгу (налагодження) коду мікроконтролера потужностей Proteus цілком вистачає, щоб виявити та усунути логічні помилки в програмі.

Структурно Proteus поділяється на два взаємопов'язані модулі, що забезпечують повний цикл розробки пристрою:

- Модуль ISIS: Використовується як візуальний редактор схемотехніки. Його функціонал дозволяє не лише створювати схеми, а й готувати інформаційну базу для переходу до етапу конструювання друкованих плат у середовищі ARES.
- Модуль ARES: Призначений для розробки топології друкованих плат. Програма підтримує інтелектуальне розміщення елементів та автоматичне розведення доріжок за допомогою трасувальника ELECTRA, спираючись на велику вбудовану бібліотеку корпусів.

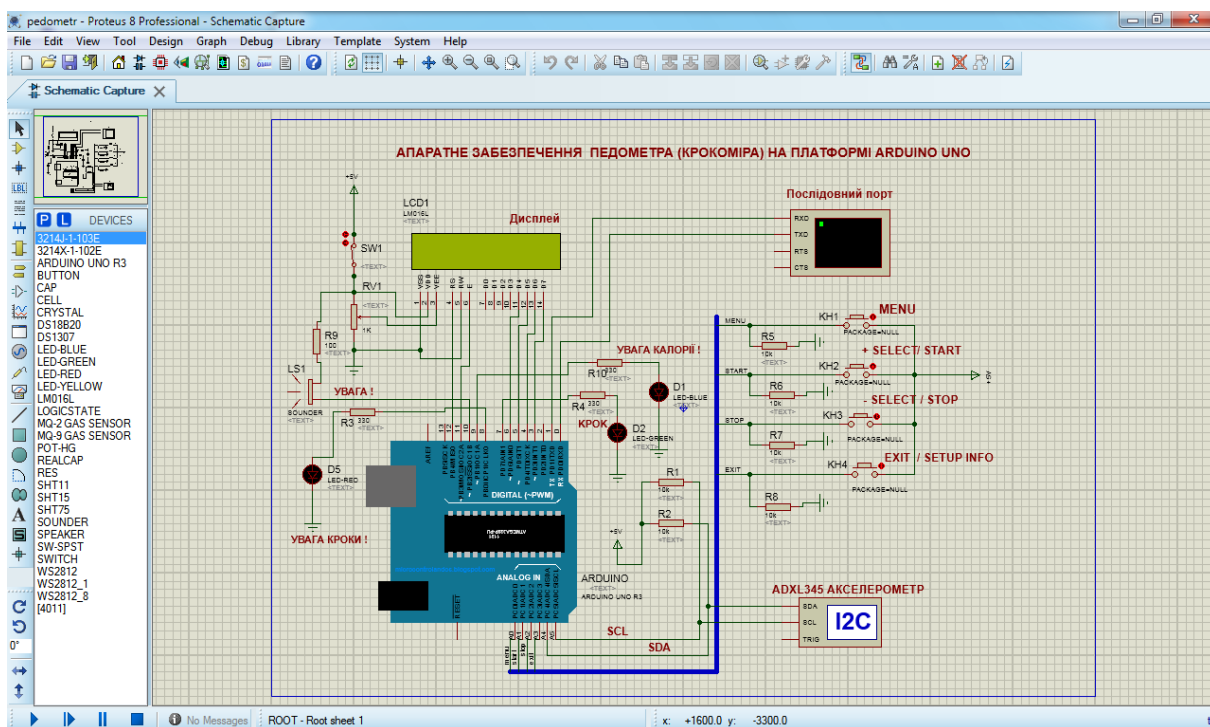


Рис. 2.1. Середовище Proteus ISIS

Серед додаткових переваг пакету Proteus слід виділити можливості апаратно-програмної взаємодії. Використання функцій COMPIМ та USBCONN дозволяє інтегрувати модельовані пристрої в реальні інформаційні мережі через фізичні порти ПК. Це, у поєднанні з потужними засобами візуальної відладки в Proteus VSM, забезпечує повний цикл розробки та верифікації мікропроцесорних систем до моменту створення дослідного зразка.

2.2. Середовище Arduino IDE для програмування систем на мікроконтролерах AVR

Для розробки ПЗ у межах екосистеми Arduino використовується однойменна IDE. Це кросплатформове середовище програмування забезпечує швидкий цикл створення, компіляції та завантаження програм у пам'ять пристроїв, пропонуючи розробнику зручний інтерфейс та набір готових інструментів для роботи з електронікою [4-10].

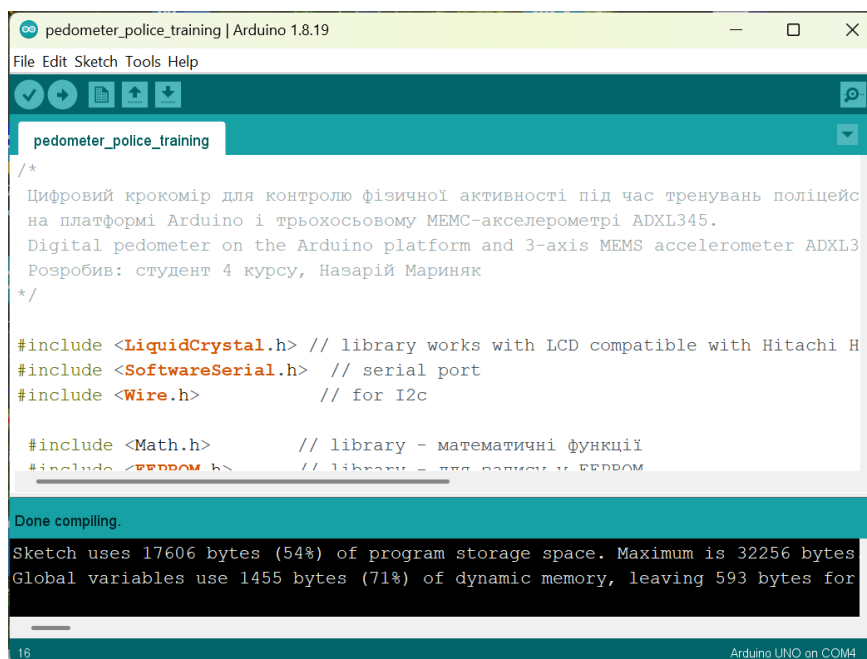


Рис. 2.2. Arduino IDE (Integrated Development Environment)

Цей інструментарій для мікроконтролерів Arduino здобув визнання завдяки відкритому коду та архітектурі. Полегшена мова програмування у поєднанні з величезним масивом доступних драйверів робить середовище ефективним як для швидкого прототипування, так і для повноцінної розробки ПЗ.

Платформи Arduino підтримують пряме програмування через USB, не потребуючи сторонніх програматорів. Програмна оболонка Arduino IDE має ергономічний дизайн, який об'єднує текстовий редактор, інформаційну консоль для моніторингу процесів, вікно логування помилок та панель швидких команд для ефективної роботи з кодом.

Програмне забезпечення для Arduino розробляється мовами C та C++ у формі так званих “скетчів”. Arduino IDE надає зручний текстовий редактор з повним набором інструментів для правки коду. Система зворотного зв'язку реалізована через спеціальну панель сповіщень, де користувач може бачити статус завантаження програми в апаратну частину або детальний опис помилок, виявлених при компіляції.

У консолі середовища користувач може переглядати повні звіти про помилки та системні сповіщення. Основні маніпуляції зі скетчами (створення, відкриття, збереження та запуск процесу компіляції) винесені на панель швидкого доступу, що значно спрощує роботу з програмним забезпеченням.

Основні операції в середовищі розробки реалізовані через систему графічних кнопок. Команда Verify/Compile відповідає за верифікацію коду, тоді як функція Upload забезпечує фінальну стадію – перетворення програми в машинний код та її завантаження в апаратну частину. Робота з файловою системою здійснюється через стандартні інструменти New, Open та Save. Для діагностики та налагодження пристрою використовуються функції Serial Monitor (відстеження даних у реальному часі) та Stop (деактивація моніторингу).

Додатковий функціонал середовища розробки згруповано у відповідних категоріях системного меню. Зокрема, розділи Sketch та Tools містять інструменти для конфігурації плати, тоді як доступ до конкретних функцій адаптується відповідно до поточного контексту виконуваних операцій.

Функціонал меню Edit дозволяє експортувати програмний код у різних форматах: наприклад, опція Copy for Forum готує текст для форумів, а Copy as HTML — для веб-ресурсів. У меню Sketch зосереджені команди управління структурою проєкту. Зокрема, функція Include Library відповідає за підключення

бібліотек через директиву `#include`, а команда Add File дозволяє розширювати проєкт новими файлами, що відображаються у вигляді окремих закладок. Для швидкого доступу до вихідних файлів на накопичувачі передбачено пункт Show Sketch Folder.

Додатковий функціонал меню Tools забезпечує конфігурацію параметрів компіляції та зв'язку з обладнанням:

- Опція Auto Format здійснює автоматичну стилізацію вихідного тексту, вирівнюючи логічні блоки та дужки.
- Через пункти Board та Serial Port розробник визначає апаратну конфігурацію системи та канал передачі даних (фізичний COM-порт або віртуальний інтерфейс).
- Команда Burn Bootloader призначена для ініціалізації мікроконтролера шляхом прошивки завантажувача. Це критично важлива операція, що потребує попередньої перевірки типу обраної платформи та вибору коректного порту при використанні зовнішніх інтерфейсів програмування.

Меню File містить розділ Sketchbook, призначений для організації та швидкого доступу до скетчів. Програма самостійно створює цю папку при першому запуску, а її розташування легко коригується у вкладці Preferences. Відкрити проєкт із блокнота можна як через системне меню, так і за допомогою гарячої клавіші Open.

Функціонал Arduino IDE передбачає роботу з проєктами, що складаються з декількох модулів, які відкриваються паралельно в окремих закладках інтерфейсу. Система забезпечує коректну обробку текстових файлів із різними розширеннями, охоплюючи як специфічні формати платформи, так і стандартні файли синтаксису C/C++ та заголовочні файли форматів .cpp, .c та .h.

Перед інсталяцією скетчу в пристрій необхідно чітко визначити модель плати та активний послідовний порт у вкладці Tools. В операційних системах сімейства Windows підключення через USB зазвичай відображається як порти з високими номерами (наприклад, COM5 чи COM8). Завершальним етапом є

натискання кнопки “Завантажити” або вибір пункту Upload у меню Sketch, що запускає процес передачі даних у мікроконтролер.

Актуальні версії платформ Arduino не потребують ручного втручання для старту прошивки завдяки системі автоматичного перезавантаження. На противагу старим модифікаціям, де використовувалася кнопка Reset, сучасна Arduino UNO автоматизує цей процес, відображаючи статус передачі даних через миготіння світлодіодів RX та TX.

Програмна архітектура Arduino передбачає наявність завантажувача (Bootloader) – резидентної мікропрограми, яка ініціалізується при старті контролера. Цей інструмент виконує роль посередника, дозволяючи завантажувати скетчі в область пам’яті програм через стандартні порти. Завдяки такому рішенню процес оновлення ПЗ стає доступним без застосування спеціалізованих технічних засобів (ISP-програматорів). Завантажувач (Bootloader) запускається під час кожного рестарту Arduino та в процесі оновлення ПЗ. Про активну фазу роботи цього модуля розробника інформує миготіння контрольного світлодіода, що є невід’ємною частиною візуальної діагностики стану пристрою.

Бібліотеки додають скетчам необхідний функціонал для керування апаратною частиною. Їх підключають через меню Sketch, що додає в код відповідні директиви `#include`. Оскільки бібліотеки займають місце в пам’яті Arduino, важливо залишати лише ті з них, що реально використовуються в проєкті. Нові бібліотеки встановлюються шляхом копіювання розпакованого архіву до папки `libraries` у вашому Sketchbook.

Для моніторингу обміну даними в Arduino IDE передбачено вікно Serial Monitor. Воно відображає вхідний потік інформації від контролера та дозволяє здійснювати вихідну передачу даних. Щоб надіслати повідомлення на платформу, достатньо ввести текст у верхній рядок і натиснути Send.

Основна частина параметрів Arduino IDE корегується у вікні Preferences. Для доступу до розширених опцій використовується окремий конфігураційний файл, місцезнаходження якого автоматично вказується в інтерфейсі

налаштувань. Це дозволяє користувачеві швидко знайти та змінити необхідні системні змінні.

Програмування МК-пристроїв на платформі Arduino. Платформи Arduino здійснюють обмін даними з периферією через систему інформаційних виводів. Залежно від програмних налаштувань, вони можуть приймати або надсилати сигнали. Важливою особливістю є те, що аналогові порти пристрою є мультифункціональними: вони здатні працювати як у своєму основному режимі, так і в ролі звичайних цифрових входів/виходів.

Функція `pinMode()` призначена для налаштування напрямку роботи портів. Оскільки входні порти у відключеному стані мають високий імпеданс, вони чутливі до завад. Для встановлення фіксованого логічного рівня за відсутності сигналу входи підключають до +5 В або “землі” через резистори. У мікроконтролерах ATmega вже передбачені вбудовані резистори на 20 кОм, які забезпечують необхідну підтяжку входу до позитивного полюса живлення.

Програмування виводів базується на використанні функцій `pinMode(pin, INPUT)` для налаштування входу та `digitalWrite(pin, HIGH)` для активації внутрішньої підтяжки до шини живлення. Важливо пам'ятати, що порти Arduino здатні витримувати струм лише до 40 мА. Спроби живити від них двигуни чи соленоїди напряму, а також випадкові короткі замикання, створюють надмірне навантаження, яке руйнує внутрішню структуру мікроконтролера.

Для керування аналоговими пристроями (наприклад, яскравістю світлодіода або швидкістю двигуна) в Arduino використовуються виходи ШІМ. Змінюючи ширину імпульсу, можна отримати будь-яку середню напругу між 0 В та 5 В. Використання функції `analogWrite()` дозволяє задавати рівень напруги в межах 256 значень: від 0 (повне вимкнення) до 255 (максимальна потужність). Наприклад, при значенні 127 ширина імпульсу становитиме половину періоду, що відповідає 2,5 В.

Платформа Arduino має шість аналогових входів, що працюють на базі 10-бітного АЦП, видаючи значення від 0 до 1023. Ці входи є універсальними: їх

можна використовувати як цифрові порти (номери 14 – 19). Програмне звернення до них нічим не відрізняється від роботи зі звичайними цифровими пінами, що дозволяє легко перемикає режим роботи виводу з аналогового зчитування на цифровий вивід через `pinMode` та `digitalWrite`.

Структура програмного забезпечення для систем Arduino спирається на програмний інтерфейс Wiring. Фактично, написаний користувачем код проходить етап препроцесингу, перетворюючись на синтаксично коректний код C/C++. Кінцева компіляція та генерація бінарного файлу здійснюється за допомогою інструментарію AVR-GCC. Середовище розробки є закритою екосистемою, орієнтованою на підтримку та прошивку лише сумісних апаратних рішень Arduino [4-10].

Апаратні платформи Arduino. Лінійка апаратних рішень Arduino представлена різноманітними модифікаціями плат. Зокрема, модель Leonardo базується на мікроконтролері ATmega32u4. Водночас такі популярні версії, як Uno та Duemilanove, функціонують на базі чипа Atmel ATmega328.

Еволюція пристроїв Arduino супроводжується постійним покращенням характеристик мікроконтролерів. На зміну простим чипам ATmega8 та ATmega168, що використовувалися в застарілих платах Diecimila, прийшли потужні рішення рівня ATmega2560, які стали основою для продуктивної платформи Mega2560. Список платформ Arduino:

- Контролери Arduino Nano або Duemilanove, оснащені чіпом ATmega328. Пристрій функціонує на тактовій частоті 16 МГц. Аналогічні параметри характерні також для моделей Arduino Pro та Pro Mini, побудованих на цьому ж мікроконтролері.
- Платформа Arduino Mega, побудована на чіпі ATmega1280, функціонує на частоті 16 МГц. Важливою особливістю цієї моделі є інтегрована можливість автоматичного перезавантаження під час прошивки або роботи.
- При використанні Arduino Mini варто враховувати, що вона ідентична Arduino NG або застарілим варіантам із чіпом ATmega168. Такі плати

функціонують на 16 МГц, проте вимагають ручного перезавантаження через відсутність автоматичного механізму.

- Для проєктів, що потребують низького енергоспоживання, використовується LilyPad з чіпом ATmega328. Його тактова частота обмежена 8 МГц при живленні 3,3 В, проте плата оснащена системою автоперезавантаження. За програмною та апаратною частиною ця модель відповідає низьковольтним версіям Arduino Pro Mini.

Апаратна платформа Arduino UNO (МК ATmega328) зображена на Рис.2.3. Пристрій працює з тактовою частотою 16 МГц (живлення 5 В) та оснащений механізмом автоматичного перезавантаження [11, 12].



Рис. 2.3. Апаратна платформа Arduino UNO

Для успішної прошивки через USB-порт вибираємо у середовищі програмування тип плати: Tools>Board>Arduino AVR Boards>Arduino UNO. Це критично для правильної ідентифікації пристрою системою.

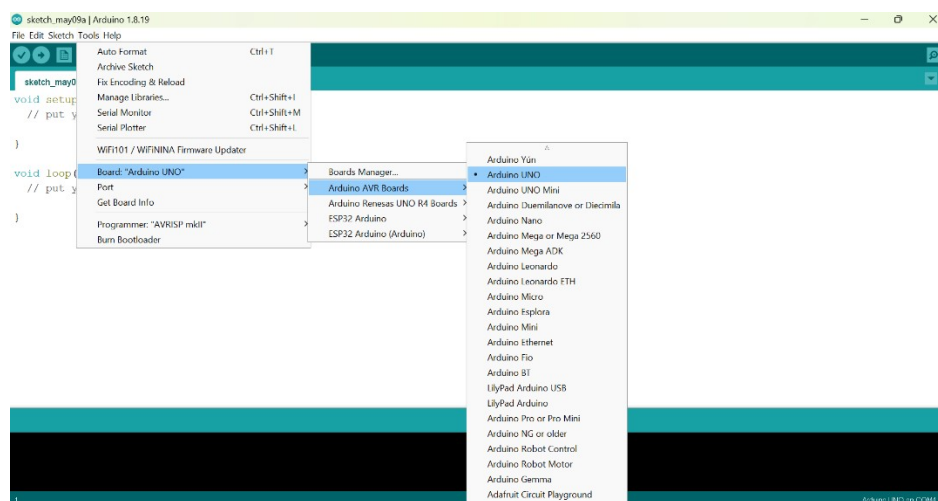


Рис. 2.4. Вибір платформи Arduino UNO

Для встановлення з'єднання з ПК обираємо у середовищі розробки порт, до якого фізично підключено пристрій (шлях: Tools > Port). Це дозволить IDE правильно розпізнати апаратну частину.

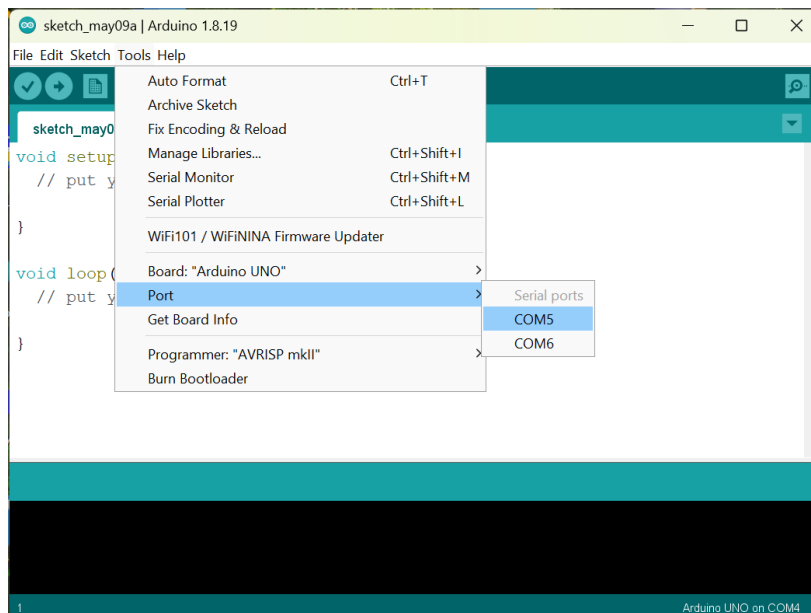


Рис. 2.5. Вибір порта до якого підключена плата Arduino UNO

Процес інсталяції програмного забезпечення в мікроконтролер плати Arduino UNO здійснюється через функцію завантаження (Sketch > Upload). Після натискання код буде автоматично перевірений та записаний у пам'ять контролера.

РОЗДІЛ 3

АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО КРОКОМІРА ДЛЯ КОНТРОЛЮ ФІЗИЧНОЇ АКТИВНОСТІ ПІД ЧАС ТРЕНУВАНЬ ПОЛІЦЕЙСЬКИХ

3.1. Вибір елементної бази для проєктування цифрового крокоміра

Апаратне забезпечення цифрового педометра базується на поєднанні платформи Arduino Uno (ATmega328) та триосьового сенсора ADXL345. Останній дозволяє точно реєструвати рух завдяки MEMS-технології. Для виведення розрахованих показників на екран у проєкті застосовано стандартний дисплейний модуль 16x2 (WH1602B).

Опис апаратної платформи ARDUINO UNO з МК ATmega328.

Апаратною основою проєктованого крокоміра виступає модуль Arduino Uno на базі чипа ATmega328P. Вибір цієї платформи зумовлений її високою продуктивністю та широким функціоналом, необхідним для обробки даних про інтенсивність навантажень поліцейських. Крім того, доступна ціна та надійність архітектури AVR роблять її вдалим інструментом для створення прототипів засобів індивідуального моніторингу.

Для обробки даних у проєкті залучено мікроконтролер ATmega328P, який має такі характеристики: архітектура: високопродуктивна 8-розрядна RISC-структура; програмне забезпечення: легка інтеграція та прошивка через стандартне середовище Arduino IDE; надійність: підтримка до 10 тисяч ітерацій перепрограмування; ефективність: оптимальне співвідношення швидкодії та енерговитрат для портативних пристроїв.

Мікроконтролери архітектури AVR представлені трьома основними групами. Серія Tiny орієнтована на мінімальні габарити (8-пінові корпуси). Серія Classic забезпечує продуктивність до 16 MIPS при невеликих обсягах пам'яті (Flash до 8 Кб). Найбільш функціональною є лінійка Mega, яка підтримує швидкість до 20 MIPS, має розширену пам'ять (Flash до 128 Кб, SRAM до 4 Кб) та вбудований багатоканальний АЦП. Головною перевагою екосистеми Atmel є

уніфікація команд: програмний код є переносним у межах усього сімейства у бік збільшення потужності.

На Рис. 3.1 зображено зовнішній вигляд модуля Arduino Uno із зазначенням функціонального призначення роз'ємів керуючого контролера ATmega328.

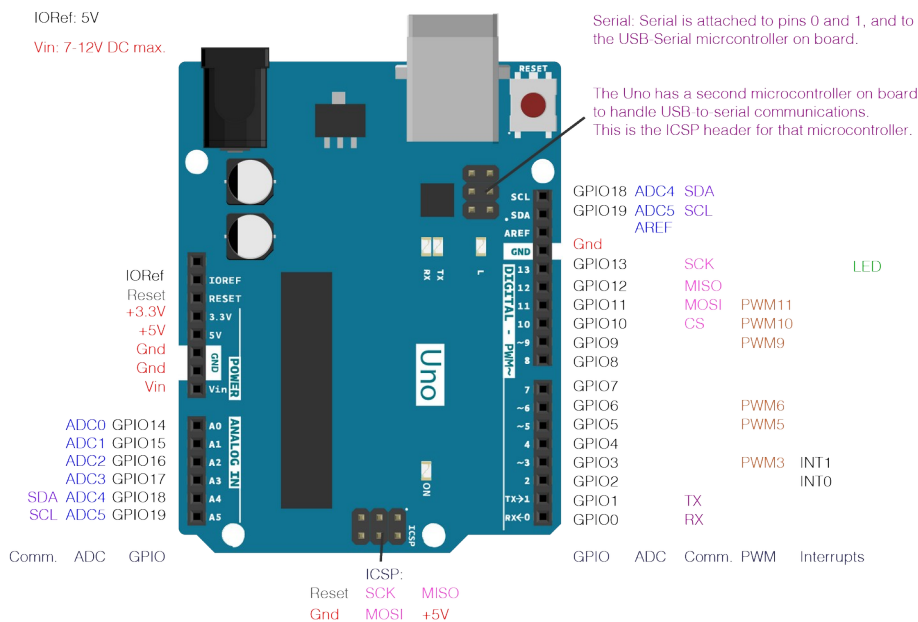


Рис. 3.1. Апаратна платформа Arduino UNO

Технічні параметри мікроконтролера ATmega328P. ATmega328P являє собою високопродуктивний 8-бітний мікроконтролер архітектури AVR, що поєднує низьке енергоспоживання з ефективною RISC-архітектурою. Даний чип є ядром популярних платформ розробки та широко застосовується в автономних пристроях.

1. Обчислювальне ядро та продуктивність

- Архітектура: Використання RISC-команд (131 інструкція), більшість із яких виконуються за один такт.
- Регістри: Наявність 32 робочих регістрів загального призначення.
- Швидкодія: Максимальна продуктивність сягає 16 MIPS при робочій тактовій частоті 16 МГц.

2. Конфігурація пам'яті

- Flash-пам'ять: 32 Кбайт для зберігання програмного коду (ресурс до 10 000 циклів перезапису).

- EEPROM: 1 Кбайт енергонезалежної пам'яті для зберігання даних (до 100 000 циклів стирання/запису).
- SRAM: Внутрішня оперативна пам'ять об'ємом 2 Кбайт.
- Програмування: Підтримка роботи з пам'яттю та конфігураційними бітами через інтерфейс JTAG.

3. Периферійні модулі та інтерфейси

Для взаємодії із зовнішніми компонентами МК оснащений потужним набором інструментів:

- Таймери: Два 8-бітних та один 16-бітний програмовані таймери-лічильники, а також окремий сторожовий таймер (Watchdog).
- Аналогові блоки: 8-канальний 10-розрядний АЦП та вбудований аналоговий компаратор.
- ШІМ: 6 виходів для формування сигналів широтно-імпульсної модуляції.
- Комунікації: Інтегровані інтерфейси USART та SPI для послідовного обміну даними.

4. Енергоспоживання та фізичні характеристики

- Енергозбереження: Підтримка 6 спеціальних режимів роботи (Idle, ADC Noise Reduction, Power-save, Power-down, Standby та Extended Standby).
- Стабільність: Система скидання при критичному зниженні напруги (Brown-out) та вбудований RC-генератор.
- Живлення: Робочий діапазон напруги становить від 2,7 В до 5,5 В.
- Інтерфейси вводу/виводу: 32 програмовані порти I/O.
- Корпуси: Випускається у варіантах PDIP (28 контактів) або TQFP (32 контакти).

На Рис.3.2 продемонстровано схему розташування контактів (pinout) МК ATmega328P у двох варіантах виконання: 32-вивідному TQFP та 28-вивідному PDIP.

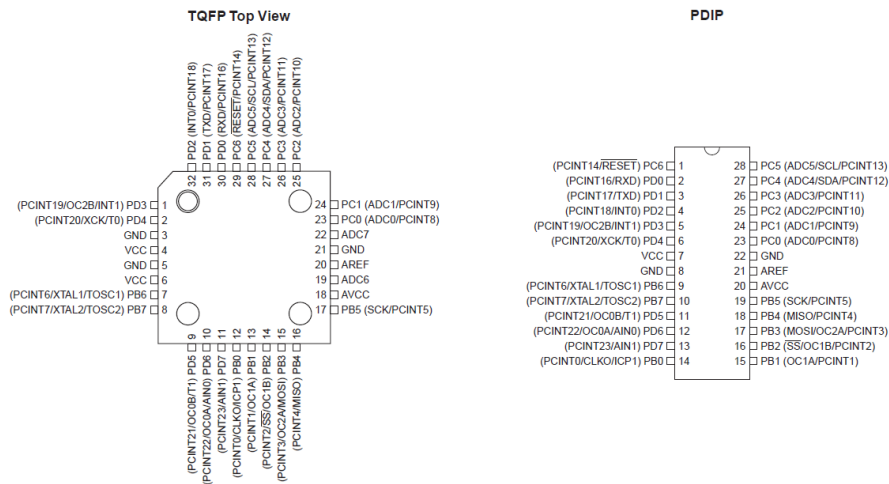


Рис. 3.2. Призначення та розташування контактів мікроконтролера ATmega328P (корпуси TQFP і PDIP)

Завдяки реалізації RISC-архітектури, модель ATmega328P забезпечує високу ефективність обробки даних. Використання 32 регістрів загального призначення та оптимізованого набору інструкцій дозволяє досягти значної переваги у швидкості – показники продуктивності цього МК у десять разів вищі порівняно з традиційними CISC-системами.

Архітектура обраного мікроконтролера характеризується наявністю таких модулів: пам'ять: 32 КБ (Flash), 2 КБ (SRAM) та 1 КБ (EEPROM); інтерфейси зв'язку: двопровідна шина (I2C-сумісна), послідовний порт SPI та інтерфейс JTAG; аналогові блоки: восьмиканальний АЦП з 10-бітною роздільною здатністю; системні модулі: три незалежні таймери-лічильники, система обробки переривань та апаратний Watchdog для захисту від збоїв.

На Рис.3.3 наведено узагальнену структурну схему МК ATmega328P, що відображає організацію його апаратної частини згідно з джерелом [12].

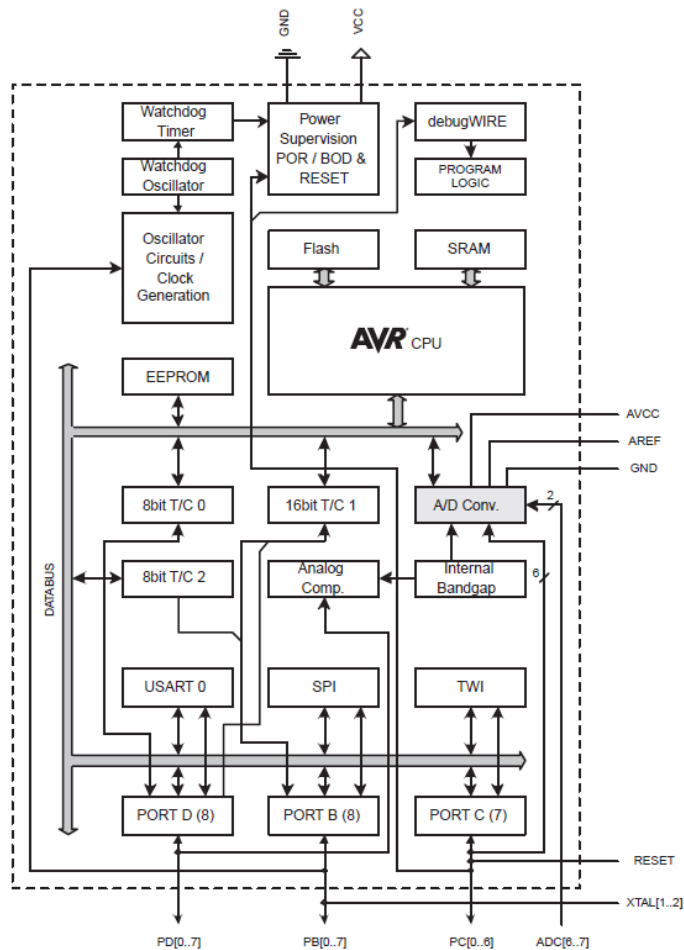


Рис. 3.3. Структурна схема мікроконтролера ATmega328P

Мікроконтролер підтримує декілька енергоефективних конфігурацій. Режим фільтрації шумів АЦП вимикає більшість модулів, окрім самого АЦП та асинхронного таймера, що важливо для точності розрахунків. У режимі відключення (Power-down) робота відновлюється лише через зовнішні сигнали або Reset. Режим Standby обмежує активність системи тільки тактовим генератором, суттєво зменшуючи енерговитрати пристрою під час паузи в обчисленнях.

Особливості експлуатації та програмування обраного контролера: пробудження: швидке повернення до активного стану через зовнішні переривання; режим Extended Standby: стан, у якому продовжують функціонувати асинхронний та головний генератори; методи програмування: доступне перезаписування Flash-пам'яті через послідовний інтерфейс SPI або за допомогою попередньо встановленого bootloader; ефективність: поєднання

RISC-архітектури та вбудованої пам'яті робить ATmega328P потужним інструментом, що став основою популярного модуля Arduino Uno.

На Рис. 3.4 представлена принципова схема апаратної платформи Arduino Uno, що демонструє особливості підключення та обв'язки МК ATmega328P.

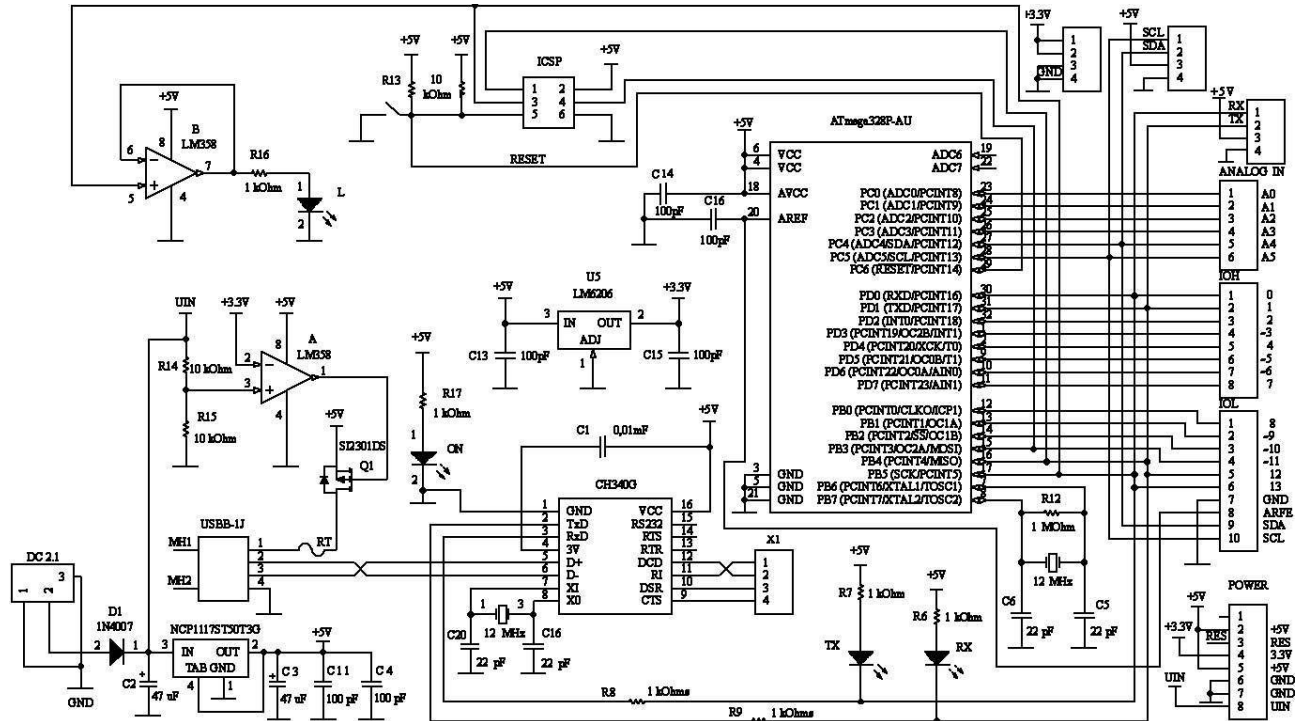


Рис. 3.4. Повна схема електричних з'єднань платформи Arduino UNO (МК ATmega328P)

Огляд характеристик алфавітно-цифрового модуля РКД. Використання дисплеїв на базі драйвера HD44780 у проекті зумовлено їхньою економічною доцільністю та низьким рівнем споживання струму. Основні переваги таких пристроїв включають: інформативність: можливість відображення складних символічних даних; автономність: низьке енергоспоживання, що критично для переносних гаджетів; універсальність: вбудоване підсвічування для роботи в різних умовах освітленості; стійкість: робота в температурних межах -20...+70 °С дозволяє експлуатувати пристрій поза приміщеннями.

Апаратна частина РК-дисплея складається з матриці рідких кристалів та вбудованого драйвера управління. Візуальний вигляд приладу показано на Рис. 3.5. Цей тип індикаторів має світлодіодне підсвічування та дозволяє відображати інформацію у дворядковому форматі (по 16 символів у рядку). Формування кожного знака відбувається в межах матриці 5×8 точок, а наявність

однопиксельних інтервалів між ними забезпечує легке зчитування даних користувачем.

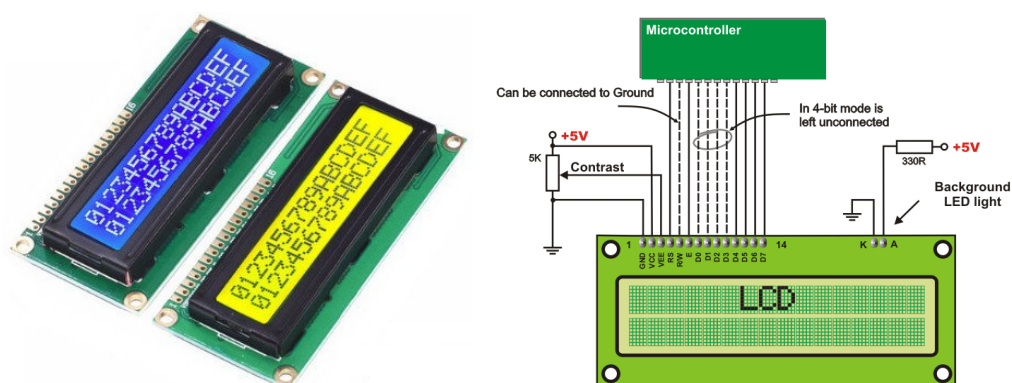


Рис. 3.5. Зовнішній вигляд алфавітно-цифрового дисплея та спосіб його з'єднання з мікроконтролером

Кожен текстовий знак, що відображається на екрані, прив'язаний до відповідного коду в оперативній пам'яті модуля. Система пам'яті РКД включає область для стандартних кодів символів та спеціальну область знакогенератора, де користувач може програмувати власні графічні об'єкти. Взаємодія між пам'яттю та фізичними сегментами панелі забезпечується внутрішньою логікою керування.

РКД-модуль на базі контролера HD44780 забезпечує широкі можливості для побудови інтерфейсу користувача. Завдяки вбудованій підтримці кирилиці та латиниці, пристрій легко адаптується під локалізовані завдання. Важливою перевагою є підтримка 4-бітного режиму, що дозволяє економити виводи мікроконтролера. Окрім стандартного набору команд, дисплей дозволяє створювати до восьми власних символів (наприклад, для індикації рівня заряду батареї чи іконок меню). Наявність функцій керування курсором та контрастністю дозволяє оптимізувати відображення інформації під конкретні умови експлуатації.

Для програмування РКД на базі HD44780 використовується інтерфейсний протокол, завдяки якому мікроконтролер надсилає байти інформації або в регістр команд (IR), або в регістр даних (DR). Роль перемикача між цими операціями виконує лінія RS. Якщо розробник встановлює на цій лінії логічний нуль, контролер сприймає дані на шині як команди. Встановлення логічної

одиниці переводить пристрій у режим передачі символів, які записуються у пам'ять для подальшої візуалізації.

Регістр даних (DR) виступає буфером для запису байтів у DDRAM або CGRAM, причому цільове місце зберігання вказується лічильником адреси (AC). У той же час регістр команд (IR) відповідає за прийом керуючих слів, які після дешифрації активують необхідні функції контролера. Такий розподіл дозволяє

Пам'ять DDRAM (80 байт) призначена для кешування кодів відображуваних символів. Вона має жорстку логічну структуру: два рядки по 40 байт. Така організація є стандартною для всіх типів РКД на базі даного контролера. Навіть якщо пристрій фізично має іншу кількість рядків чи стовпців (80×1, 20×4 тощо), програмування та адресація здійснюються виходячи з базової моделі “2 рядки по 40 символів”.

Вивід інформації на дисплей реалізовано через механізм динамічного сканування, яким керує інтегрована мікросхема. Сама РК-панель являє собою матрицю сегментів (по 200 у кожному рядку). Програмно можна встановити наступні режими відображення: однорядковий (5×7 точок): використовується 8 ліній матриці; однорядковий (5×10 точок): використовується 11 ліній матриці; дворядковий (5×7 точок): задіюється 16 ліній матриці.

Базові можливості мікросхеми HD44780 обмежені 40 виходами сегментів, чого достатньо для роботи з дисплеями на 8 символів. У випадках, коли пристрій має більшу інформаційну ємність, основний контролер працює в парі з допоміжними мікросхемами (наприклад, HD44100), які касадно розширюють кількість ліній керування. Гнучкість налаштування логіки обробки даних у такій системі забезпечується через систему внутрішніх статусних прапорців.

Для керування виводом даних на РКД використовується вбудована бібліотека `LiquidCrystal.h`, адаптована під протокол контролера HD44780. Вона дозволяє абстрагуватися від низькорівневих команд і легко працювати з дисплеями різних розмірностей (наприклад, 2x16 або 2x20). Практична реалізація підключення модуля до цифрових портів Arduino UNO проілюстрована на Рис.3.6.

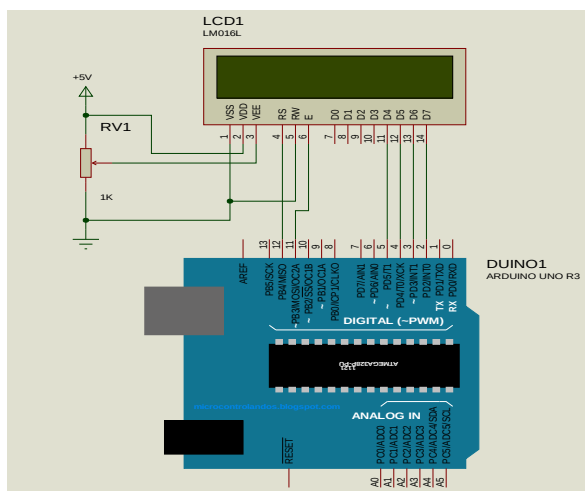


Рис.3.6. Апаратне сполучення РКД на базі HD44780 з платою Arduino UNO (4-бітний режим передачі даних)

Огляд трьохосьового сенсора ADXL345. ADXL345 – це інтелектуальний MEMS-акселерометр, що забезпечує високу точність вимірювання лінійного прискорення вздовж трьох осей [14]. Датчик здатен аналізувати два типи впливів: статичне прискорення: використовується для вимірювання сили тяжіння Землі та визначення просторової орієнтації пристрою; динамічне прискорення: виникає внаслідок раптових рухів, кроків або вібрацій. Саме поєднання цих можливостей дозволяє реалізувати на базі ADXL345 точний алгоритм підрахунку кроків.

Прискорення описує динаміку зміни швидкісних характеристик тіла. Основними одиницями вимірювання є метри на секунду в квадраті (m/s^2), проте для налаштування акселерометрів зазвичай застосовують кратні одиниці гравітаційної постійної (g). При цьому базове співвідношення приймається як $1g = 9,8 m/s^2$.

Акселерометри є незамінними інструментами для вимірювання вібраційних навантажень та контролю просторового положення механічних систем. Вони дозволяють з високою точністю реєструвати миттєве прискорення рухомих тіл, що необхідно для коректного функціонування інерціальних модулів.

Широкий модельний ряд доступних акселерометрів на базі MEMS-технологій дозволяє підібрати оптимальне рішення залежно від вимог проекту.

Основними диференціюючими параметрами пристроїв є: геометричні розміри: важливі для портативних пристроїв (як-от крокоміри); метрологічні характеристики: рівень чутливості до динамічних змін; енергетичні параметри: сумісність із різними джерелами живлення; вартість: впливає на фінальну ціну готового виробу.

Виходячи з кількості вимірювальних осей, сенсори класифікують на пристрої з однією, двома або трьома осями чутливості. Найбільш функціональними є трьохосьові акселерометри, що забезпечують комплексний аналіз динаміки тіла за осями X, Y та Z. Використання саме такого типу датчика є критично важливим для систем навігації та крокомірів, де напрямок руху може змінюватися.

Для обґрунтування вибору датчика в табл. 3.1 наведено основні параметри MEMS-сенсорів фірми Analog Devices. Оцінка доцільності використання тієї чи іншої моделі базується на комплексному аналізі таких характеристик: діапазон вимірювань: максимальна величина g, яку здатний зафіксувати прилад; чутливість: мінімальна зміна прискорення, що реєструється; конструктивні особливості: кількість площин вимірювання (X, Y, Z); електричні та програмні параметри: напруга живлення та підтримувані протоколи зв'язку (I2C/SPI); економічний фактор: ціна за одиницю продукції.

Таблиця 3.1. Ключові параметри MEMS-акселерометрів від Analog Devices

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	I ² C, SPI	X, Y, Z	3	\$6.33 (ADXL372BCCZ-RL7)
ADXL357	10 g, 20 g, 40 g	200μ	2.25	-40 to 125°C	1k	I ² C, 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	I ² C, 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)

Аналіз номенклатури датчиків дозволяє розділити їх на три групи: одноосьові (ADXL151, ADXL1004/05), двохосьові (ADXL288/295/251) та трьохосьові (ADXL355/357/345). Для розробки крокоміра пріоритет було надано трьохосьовому акселерометру ADXL345. Вибір обумовлений його економічною доступністю та задовільними метрологічними параметрами, які дозволяють ефективно реєструвати динамічні прискорення тіла людини при мінімальних витратах на комплектуючі. На Рис.3.7 зображено осі вимірювання трьохосьового акселерометра ADXL345.

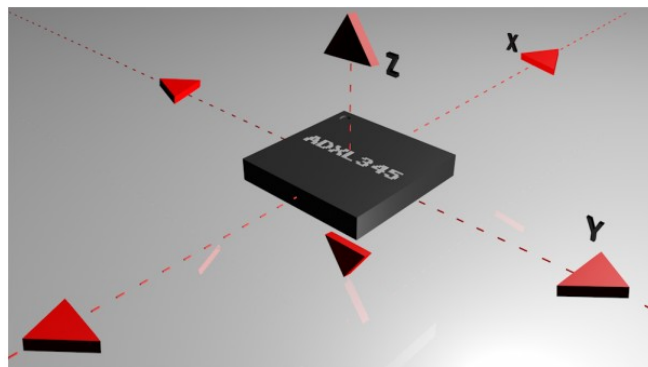


Рис. 3.7. Осі вимірювання трьохосьового акселерометра ADXL345

Принцип дії акселерометра базується на вимірюванні ємності мікроскопічних конденсаторів. Конструктивно пристрій містить інерційну масу з гребінчастими виступами, яка з'єднана з корпусом через пружні елементи. При виникненні прискорення інерційна маса зсувається, змінюючи зазор між рухомими та статичними пластинами. Оскільки ємність конденсатора обернено пропорційна відстані між його обкладками, електроніка датчика реєструє ці зміни та перетворює їх у цифрове значення прискорення.

Контролери акселерометрів дозволяють програмно обирати межі вимірювання прискорення. Для завдань, що потребують високої деталізації сигналу, обирають вузькі діапазони, оскільки вони забезпечують максимальну чутливість до незначних механічних коливань. Навпаки, великі діапазони вимірювань (порядку 250g) розраховані на роботу в умовах критичних навантажень, характерних для випробувань ракетних двигунів. У даному проекті використання чутливого режиму з малим діапазоном є необхідним для коректного виявлення амплітуди кроку.

Для розробки крокоміра на основі акселерометра ADXL345 оптимальним вважається налаштування чутливості на рівні $\pm 2g$. Такий вибір забезпечує високу дискретність сигналу (приблизно 3.9 мг/LSB), завдяки чому пристрій легко фіксує амплітудні коливання при кожному кроці. Використання ширшого діапазону, наприклад $\pm 16g$, є недоцільним, оскільки в такому разі рівень шумів може поглинути слабкі сигнали, характерні для звичайної ходьби.

Внутрішня організація та логіка роботи акселерометра ADXL345 ілюструються за допомогою функціональної схеми, зображеної на Рис.3.8.

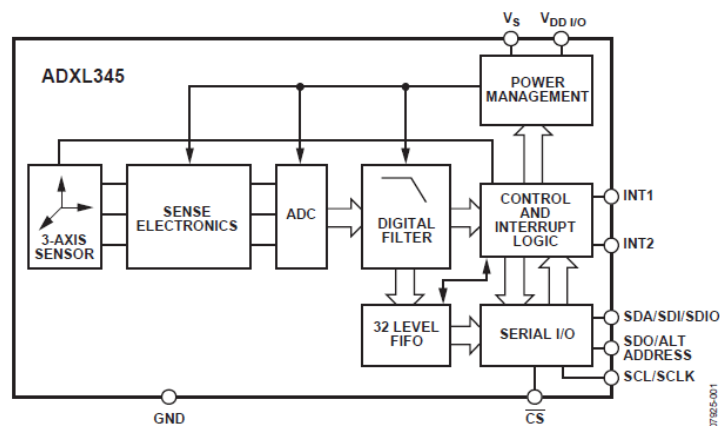


Рис. 3.8. Структурна організація логічних та аналогових вузлів сенсора ADXL345

Через технологічні особливості виробництва МЕМС-структур, вихідні сигнали по осях X, Y, Z можуть мати відхилення (“зміщення нуля”). Для мінімізації цих похибок проводиться процедура калібрування акселерометра. Отримані корегувальні коефіцієнти вносяться до вбудованих регістрів зміщення (OFSX, OFSY, OFSZ). Це забезпечує апаратну компенсацію похибки безпосередньо в самому сенсорі, звільняючи обчислювальні ресурси мікроконтролера Arduino від додаткових математичних операцій.

Модель ADXL345 вирізняється гнучкістю налаштувань: користувач може самостійно обирати межі вимірювання прискорення в діапазоні до $\pm 16g$. Висока розрядність вихідних даних (13 біт) гарантує точність детектування кроків. Датчик підтримує цифрові протоколи обміну даними (I2C/SPI), що спрощує його інтеграцію з мікроконтролером Arduino. Робочий частотний діапазон сенсора (до

1600 Гц) робить його універсальним інструментом для аналізу динамічних зміщень та просторової орієнтації пристрою.

При виборі сенсора необхідно враховувати його динамічні властивості, зокрема смугу пропускання. Вона вказує на максимальну частоту механічних коливань, які акселерометр здатен обробити без суттєвої втрати точності. Важливо пам'ятати, що частота опитування АЦП повинна бути як мінімум у два рази більшою за частоту вимірюваного сигналу ($f_s > 2f_{max}$). Для коректного вимірювання ударів слід обирати пристрій з відповідним динамічним діапазоном та достатньою швидкодією, щоб часова константа датчика була меншою за тривалість самого удару.

Критерії безпеки авіаційного обладнання базуються на тривалості впливу прискорень. Якщо перевантаження у 10g протягом кількох секунд є критичним для пілота, то короткочасний імпульс тієї ж сили (до 10 мс) не становить загрози навіть для звичайної людини.

Дослідження конструктивної міцності базується на аналізі реакції об'єкта на ударні навантаження (резонанс, демпфування), а не на простій передачі енергії удару. Для отримання коректних даних характеристики акселерометра мають чітко відповідати очікуваному діапазону частот. Варто пам'ятати, що низькочастотні прилади не здатні адекватно реєструвати високодинамічні прискорення.

Широкосмугові акселерометри ефективно працюють із низькорівневими сигналами прискорень. Спеціалізовані мікросхеми, що підтримують діапазони від $\pm 70g$ до $\pm 500g$ у спектрі до 30 кГц, складно знайти у вільному продажу через їхню високу ціну. Для простих пристроїв вимірювання кроків оптимальним вибором за критерієм «ціна-якість» залишається модель ADXL345.

Модуль ADXL345 працює від джерела напруги 2,0 – 3,6 В і дозволяє гнучко налаштувати межі вимірювань від $\pm 2g$ до $\pm 16g$. Пристрій підтримує два режими дискретизації: стандартний 10-бітний та режим повної роздільної здатності, де при максимальному діапазоні точність сягає 13 біт (стала чутливість 4 mg на одиницю молодшого розряду). Датчик оснащений

вбудованими алгоритмами для виявлення падіння та динамічних поштовхів. Обмін даними здійснюється через шини SPI або I²C, а для оптимізації роботи передбачено два виводи переривань. Висока надійність підтверджується здатністю витримувати критичні перевантаження до 10000g та стабільною роботою в широкому температурному діапазоні (-40...+85°C). Схематичне зображення корпусних виводів (пінів) акселерометра ADXL345 та їхнє маркування наведено на Рис. 3.9.

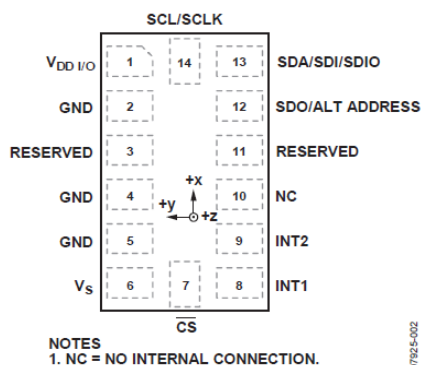


Рис. 3.9. Схема розташування контактів мікросхеми ADXL345

Функціональне призначення кожного контакту деталізовано у Табл. 3.2.

Таблиця 3.2. 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

Регістрова карта акселерометра ADXL345, що складається з 31 адресної адресної комірки, представлена в Табл. 3.3. Керування станом пристрою здійснюється через регістр POWER_CTL. За замовчуванням акселерометр перебуває в режимі *Standby*. Щоб ініціювати збір даних, слід активувати біт D3 (Measure). Повернення до стану очікування відбувається шляхом обнулення цього ж біта. У табл. 3.4 відображено пропорційний зв'язок між частотою оновлення даних внутрішнього АЦП та обсягом споживаної потужності пристрою ADXL345.

Таблиця 3.3. Перелік та опис регістрів акселерометра 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 поштовок
0X22	34	Latent	R/W	00000000	Затримка поштовок
0X23	35	Window	R/W	00000000	Тимчасове вікно для другого поштовок
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
0X2A	42	TAP_AXES	R/W	00000000	Управління осями для поштовок/подвійного
0X2B	43	ACT_TAP_STATUS	R	00000000	Джерело поштовок/ подвійного поштовок
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	DATA0	R	00000000	X - вісь DATA 0
0X33	51	DATA1	R	00000000	X - вісь DATA 1
0X34	52	DATA0	R	00000000	Y - вісь DATA 0
0X35	53	DATA1	R	00000000	Y - вісь DATA 1
0X36	54	DATA0	R	00000000	Z - вісь DATA 0
0X37	55	DATA1	R	00000000	Z - вісь DATA 1
0X38	56	FIFO_CTL	R/W	00000000	Управління FIFO
0X39	57	FIFO_STATUS	R	00000000	Статус FIFO

Таблиця 3.4. Енергоспоживання акселерометра 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

Для зниження енерговитрат в акселерометрі ADXL345 передбачена можливість автоматичного переходу в режим сну під час бездіяльності.

Конфігурація цієї функції передбачає задання порогових параметрів у регістрах THRESH_INACT та TIME_INACT з подальшою активацією бітів AUTO_SLEEP (D4) та Link (D5) у регістрі POWER_CTL. Найбільш ощадним є режим очікування (Standby Mode), у якому споживання струму мінімізується до 0,1 мкА, проте активне вимірювання припиняється. Перемикання в цей стан здійснюється шляхом скидання біта D3 у регістрі керування живленням.

Передача даних у ADXL345 реалізована за допомогою інтерфейсів SPI та I2C. Згідно з технічною документацією, акселерометр працює лише в режимі Slave [15].

Для забезпечення коректної передачі даних на частотах 3200 Гц та 1600 Гц рекомендовано використовувати протокол SPI зі швидкістю від 2 МГц. Якщо частота дискретизації становить 800 Гц, швидкість інтерфейсу має бути не меншою за 400 кГц. Для нижчих значень частоти вибірки необхідна швидкість обміну даними визначається пропорційно.

Для використання інтерфейсу I2C необхідно підтягнути пін CS до лінії живлення. Гнучкість адресації забезпечується виводом ALT ADDRESS: за наявності на ньому високого рівня адреса ADXL345 визначається як 0x1D. У процесі обміну даними ця адреса трансформується: для операцій запису (біт R/W = 0) вона зміщується до 0x3A, а для операцій зчитування (біт R/W = 1) – до 0x3B.

Якщо на виводі ALT ADDRESS встановлено рівень “лог. 0”, пристрій ідентифікується за адресою 0x53. У контексті повного байта керування (включно з бітом напрямку даних) адреса трансформується у 0xA6 для запису та 0xA7 для читання. Наочне зображення схеми підключення модуля для роботи через інтерфейс I2C наведено на Рис. 3.10.

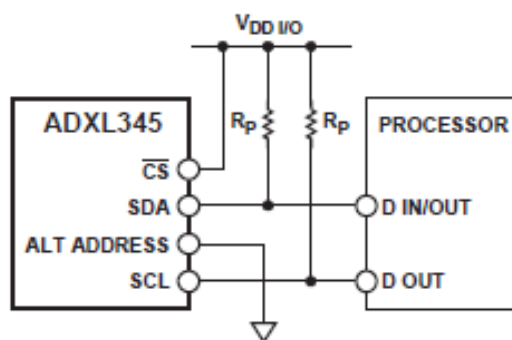


Рис. 3.10. Типова схема під'єднання ADXL345 на шину I2C

Для оперативного сповіщення зовнішнього мікроконтролера ADXL345 використовує два програмовані виводи переривань: INT1 і INT2. Датчик здатний детектувати такі критичні стани, як поява нових даних у регістрах, механічні імпульси (удари), стан невагомості (вільне падіння), а також автоматично фіксувати початок руху або тривалу бездіяльність.

Режим роботи виходів переривань ADXL345 за замовчуванням передбачає високий активний рівень. Конфігурація регістра DATA_FORMAT (зокрема біта INT_INVERT) дозволяє інвертувати полярність сигналу до низького рівня. Завдяки регістру INT_ENABLE можлива паралельна активація декількох джерел переривань, а їхня маршрутизація на фізичні виводи INT1 або INT2 здійснюється шляхом програмування відповідних розрядів у регістрі INT_MAP.

Під час читання результатів вимірювань виклик нових переривань блокується. Конфігурація регістра INT_MAP дозволяє розподілити події між виходами INT1 та INT2.

0x2F INT_MAP (read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
DATA_READY	SINGLE_TAP	DOUBLE_TAP	ACTIVITY	INACTIVITY	FREE_FALL	WATERMARK	OVERRUN

Біт DATA_READY сигналізує про готовність даних: встановлений стан (“1”) свідчить про наявність незчитаних значень прискорення, тоді як скинутий стан (“0”) вказує на відсутність нових вимірювань.

Активація біта SINGLE_TAP відбувається у випадку, коли амплітуда прискорення перевищує поріг, заданий у регістрі THRESH_TAP, причому тривалість цього імпульсу є меншою за часове вікно, визначене в регістрі DUR.

Біт DOUBLE_TAP встановлюється в “1”, якщо система фіксує два удари, що відповідають критеріям THRESH_TAP та DUR. Ключовою умовою є дотримання часових інтервалів: другий імпульс повинен виникнути не раніше, ніж закінчиться час затримки (LATENT), і не пізніше завершення вікна очікування другого удару (WINDOW).

Реєстрація стану активності (біт ACTIVITY) відбувається у момент, коли амплітуда прискорення стає більшою за поріг, визначений у регістрі THRESH_ACT. Біт FREE_FALL активується, якщо прискорення за кожною з осей не виходить за межі значення THRESH_FF упродовж певного періоду.

Прапорець WATERMARK спрацьовує, якщо обсяг даних у FIFO досягає ліміту, встановленого в FIFO_CTL. Біт деактивується одразу після зчитування вмісту черги. Прапорець переповнення (OVERRUN) встановлюється щоразу, коли незчитана інформація втрачається через надходження нових даних. Для режиму bypass це означає оновлення вмісту регістрів DATAX/Y/Z до того, як користувач встиг отримати попередні виміри. Для решти режимів стан OVERRUN сигналізує про вичерпання вільного місця у FIFO. Біт скидається системою самостійно після читання вмісту буфера. Щоб визначити джерело виникнення переривання, слід прочитати стан регістру 0x30. Доступ до результатів вимірювань здійснюється через послідовне читання шести регістрів: DATA0, DATA1, DATAY0, DATAY1, DATAZ0 та DATAZ1. Початковою точкою для операції читання є адреса 0x32 (регістр DATA0).

Вибір діапазону вимірювань реалізується через налаштування відповідних бітів у регістрі DATA_FORMAT.

0x31 DATA_FORMAT (read/write)

D7	D6	D5	D4	D3	D2	D1	D0
SELF_TEST	SPI	INT_INVERT	X	FULL_RES	JUSTIFY	RANGE	

Параметр FULL_RES визначає розрядність вихідних даних. Значення “1” відповідає режиму повної роздільної здатності, а значення “0” обмежує її до 10 біт. Параметр JUSTIFY відповідає за спосіб розміщення даних у вихідних регістрах: значення “0” активує вирівнювання по правому краю, тоді як значення “1” перемикає його на лівий край. За допомогою комбінації бітів RANGE встановлюється шкала вимірювань пристрою (Табл. 3.5).

Таблиця 3.5. Шкали вимірювань акселерометра ADXL345

D1	D0	g-Range
0	0	±2 g
0	1	±4 g
1	0	±8 g
1	1	±16 g

Для забезпечення стабільної точності рекомендується активувати біт FULL_RES (D3 регістра DATA_FORMAT). У такому разі акселерометр працює з 13-бітною роздільною здатністю, що гарантує фіксовану чутливість 3,9 mg/LSB для будь-якої шкали вимірювань.

Практична реалізація пристроїв на базі ADXL345 часто передбачає використання готових модулів (Рис. 3.11 – 3.12).

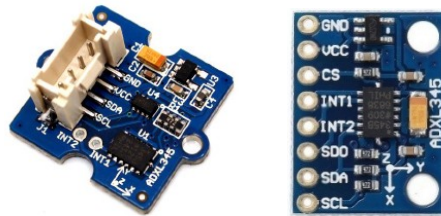


Рис. 3.11. Типові варіанти модулів з мікросхемою ADXL345

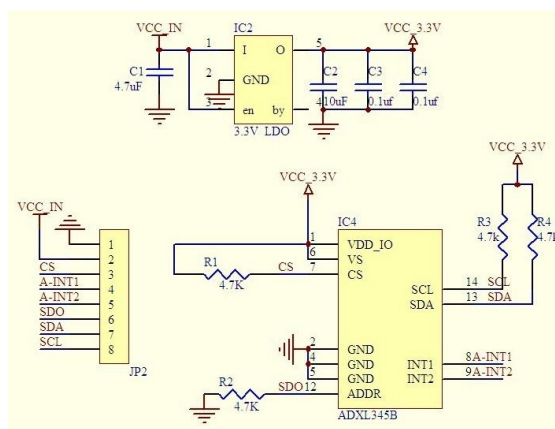


Рис. 3.12. Схематичне рішення модуля GY-291 з акселерометром ADXL345

3.2. Проектування цифрового крокоміра для контролю фізичної активності

Цифровий педометр, призначений для моніторингу фізичних навантажень під час підготовки особового складу поліції, має забезпечувати автоматичний облік зроблених кроків, подоланої дистанції та енерговитрат. Пристрій повинен здійснювати аналітичну обробку зібраної інформації з можливістю її трансляції через послідовний інтерфейс. Обов'язковою є наявність звукового та візуального сповіщення при досягненні встановлених лімітів активності. Для персоналізації

розрахунків передбачено меню введення антропометричних даних (зріст, вага), конфігурація яких має зберігатися в енергонезалежній пам'яті [16].

Конструкція крокоміра передбачає інтеграцію програмних алгоритмів та апаратної бази [17-22]. Апаратна конфігурація пристрою включає обчислювальний блок на базі Arduino UNO, датчик прискорення ADXL345, а також периферійні пристрої: РК-модуль, світлозвукову сигналізацію та клавіатуру (кнопки КН1–КН4) для взаємодії з інтерфейсом користувача.

Мікроконтролер ATmega328 у складі плати Arduino Uno реалізує основний алгоритм крокоміра: зчитування даних з ADXL345, керування РК-дисплеєм, обмін даними по UART та моніторинг порогових значень для спрацювання сигналізації. Всі налаштування пристрою депонуються в EEPROM МК. Структурну організацію цифрового крокоміра на основі ADXL345 та Arduino UNO наведено на Рис. 3.13.

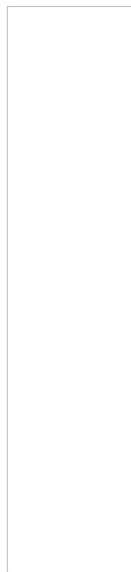


Рис. 3.13. Архітектура цифрового крокоміра на основі Arduino UNO та MEMS-акселерометра ADXL345

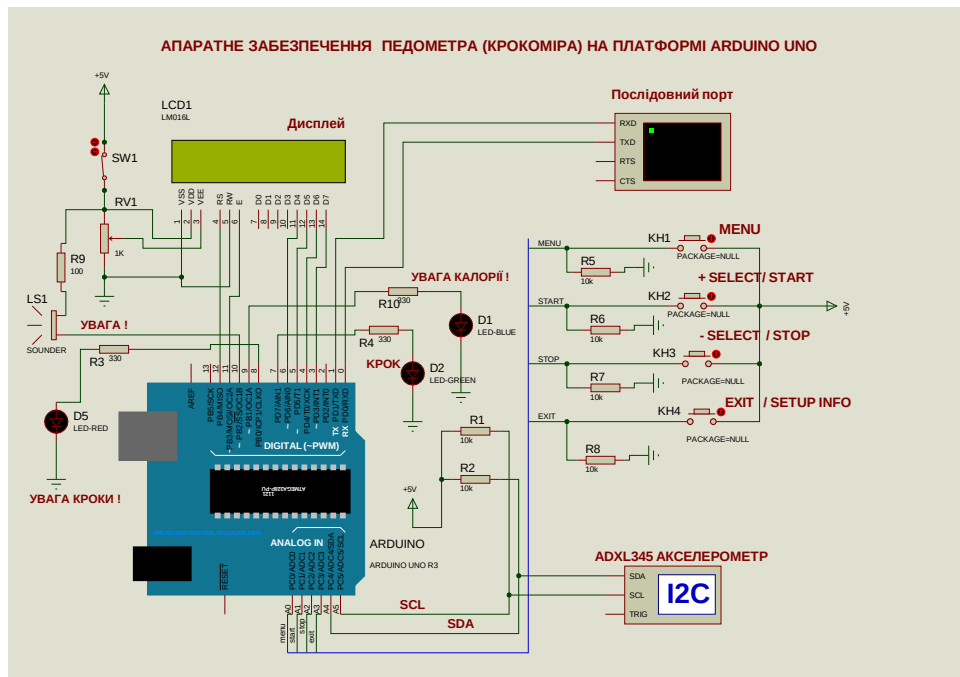


Рис. 3.14. Апаратне забезпечення цифрового крокоміра в Proteus ISIS

Принцип роботи апаратної частини крокоміра, що показана на Рис. 3.14, полягає у відстеженні змін вектора прискорення за допомогою датчика ADXL345 та плати Arduino Uno. Кожен зафіксований крок ініціює короткочасне ввімкнення індикатора D2. Окрім прямого підрахунку кроків, девайс здійснює обчислення метаболічних показників та подоланого шляху, адаптуючи розрахунки під зріст і вагу людини. Виведення інформації реалізовано через монітор послідовного порту та підключений ПК-модуль.

Через меню налаштувань користувач може встановити цільові показники за кроками або калоріями. При досягненні ліміту калорій система активує синій світлодіод D1 та зумер LS1, а при виконанні норми кроків – червоний світлодіод D5 та аналогічний звуковий сигнал. Усі тривожні повідомлення паралельно транслюються в послідовний порт для подальшого аналізу. Інтерфейс крокоміра передбачає виведення службових підказок на дисплей: режим контролю кроків маркується як “AS”, а режим спалених калорій – як “AC”. Повний контроль над роботою приладу забезпечується клавіатурою з чотирьох кнопок: КН1 відповідає за меню, КН2 та КН3 виконують функції вибору (плюс/мінус) та керування стартом/стопом, а КН4 слугує для виходу або отримання інформації про налаштування.

З допомогою клавіші КН1-“MENU” здійснюється перехід до налаштувань профілю: “WEIGHT”: редагування даних про вагу. “HEIGHT”: внесення показників зросту. Параметри зросту та ваги безпосередньо впливають на визначення довжини кроку та обсяг спалених калорій за один цикл. Для автоматизації розрахунків ці відомості, взяті з розділу 1, були попередньо занесені до пам’яті керуючого пристрою. “ATTENTION STEPS”: встановлення цільової норми кроків. “ALARM CALORICITY”: налаштування ліміту спалених калорій для сповіщення. “MODE AS/AC”: перемикання між режимами відстеження (кроки/калорії) або вимкнення функції контролю (режим OFF).

Кнопка КН1 слугує для вибору потрібного пункту меню. Коригування параметрів у межах обраної опції здійснюється клавішами КН1, КН2, КН3 та КН4. Система підтримує збереження конфігурації в енергонезалежній пам’яті пристрою.

Клавіша КН2 – “+SELECT/START” має два режими роботи: детектування: запуск крокоміра для підрахунку активності; навігація: вибір необхідних опцій у меню налаштувань. Кнопка КН3 (“-SELECT/STOP”) призначена для виконання двох операцій: зупинка: перехід крокоміра в пасивний стан (детектування кроків вимикається); вибір: використання клавіші як селектора під час навігації в налаштуваннях. За допомогою кнопки КН4 – “EXIT/SETUP INFO” можна ознайомитися з поточною конфігурацією пристрою або вийти з налаштувань у попереднє меню. Система передбачає два канали виведення даних: на LCD-дисплей: постійне відображення статистики кроків, дистанції та калорійності; у послідовний порт: передача сирих даних з датчика ADXL345 (прискорення по трьох осях та загальний вектор) для діагностичних цілей.

Алгоритм функціонування системи передбачає два основні стани:

- Робочий режим: включає безперервне опитування сенсорів та кнопок керування, аналіз даних акселерометра для виявлення кроків, а також візуалізацію результатів.
- Режим очікування: підрахунок кроків призупиняється, але пристрій продовжує виводити актуальну інформацію на РКД та в послідовний порт.

Для підвищення енергоефективності приладу реалізована можливість вимкнення підсвічування або самого РКД через вимикач SW1. Це дозволяє економити заряд батареї, не впливаючи на фонову роботу мікроконтролера.

Після подачі живлення мікроконтролер виконує наступні кроки:

1. Конфігурація модулів: РКД, UART-інтерфейсу, клавіатури та звукового випромінювача (ШІМ).
2. Тестування сенсора: запит до акселерометра ADXL345.
3. Обробка результатів: При виявленні несправності: вивід тексту “ADXL345 ERROR”. При штатному режимі: відображення статусу “PEDOMETR OK” протягом двох секунд.

У стані очікування мікроконтролер продовжує вимірювати прискорення та відстежувати активність користувача через клавіатуру. Педометр залишається в цьому режимі до моменту ініціації команди “START” через клавішу KН2, після чого система переходить до безпосереднього алгоритму підрахунку кроків.

Інтерфейс РКД розділений на зони для зручності моніторингу:

1. Рядок 1: Статистика активності (кроки та енерговитрати).
2. Рядок 2: Дистанція (метри) та код поточного режиму (“W” – детектування активоване, “S” – підрахунок зупинено).

Окрім основних даних, система виводить позначку режиму контролю: “AS” для кроків або “AC” для калорій. Якщо користувач долає встановлену межу, педометр реагує комплексно: надсилає текст попередження в послідовний порт, вмикає звукову сигналізацію та запускає блимання світлодіода, що сигналізує про досягнення ліміту за кроками або калоріями.

Програмна частина проекту реалізована за допомогою інструментарію Arduino IDE. Використання бібліотек середовища дозволило оптимізувати код, а інтеграція з Proteus ISIS забезпечила можливість попереднього тестування пристрою. Заключним етапом стала апаратна реалізація – завантаження відкомпільованого бінарного файлу в плату Arduino UNO.

РОЗДІЛ 4
ПРОГРАМНО-АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО
КРОКОМІРА ДЛЯ КОНТРОЛЮ ФІЗИЧНОЇ АКТИВНОСТІ ПІД ЧАС
ТРЕНУВАНЬ ПОЛІЦЕЙСЬКИХ

4.1. Алгоритм функціонування цифрового крокоміра для контролю фізичної активності



Рис. 4.1. Алгоритм функціонування цифрового крокоміра (початок)

Рис. 4.2. Алгоритм функціонування цифрового крокометра (продовження 1)

Рис. 4.3. Алгоритм функціонування цифрового крокоміра (продовження 2)

Рис. 4.4. Алгоритм функціонування цифрового крокоміра (продовження 3)

Рис. 4.5. Алгоритм функціонування цифрового крокометра (продовження 4)

Рис. 4.6. Алгоритм функціонування цифрового крокоміра (продовження 5)

Рис. 4.7. Алгоритм функціонування цифрового крокоміра (продовження б)

Рис. 4.8. Алгоритм функціонування цифрового крокоміра (продовження 7)

Рис. 4.9. Алгоритм функціонування цифрового крокометра (продовження 8)

Рис. 4.10. Алгоритм функціонування цифрового крокометра (продовження 9)

Рис. 4.11. Алгоритм функціонування цифрового крокометра (продовження 10)

4.2. Реалізація програмного інтерфейсу для роботи з MEMS-датчиком прискорення ADXL345

Програмний модуль для роботи з ADXL345 включає набір низькорівневих та високорівневих підпрограм. Низькорівневий рівень представлений функціями ADXL345_ReadByte, ADXL345_ReadInt та ADXL345_WriteByte, які відповідають за безпосередній обмін даними по шині. До високорівневих методів належать процедура ініціалізації (ADXL345_Init), налаштування формату даних (ADXL345_SetResolution), а також алгоритми отримання результатів вимірювань (readAccel) та калібрування (ADXL345_CalculateCalibration).

Лістинг функцій бібліотеки для роботи з акселерометром ADXL345 винесено у додаток 1.

4.3. Розроблення програмних модулів режимів роботи крокоміра

Програмний комплекс педометра побудований за модульним принципом. Нижче наведено перелік ключових функцій бібліотеки, які відповідають за обробку даних, керування режимами та інтерфейс користувача:

```
// записування даних long int в пам'ять EEPROM:
void writeLongIntoEEPROM(int address, long number) {
    // Використовуємо update, щоб не переписувати ті самі дані марно
    EEPROM.update(address, (uint8_t)(number >> 24));
    EEPROM.update(address + 1, (uint8_t)(number >> 16));
    EEPROM.update(address + 2, (uint8_t)(number >> 8));
    EEPROM.update(address + 3, (uint8_t)(number & 0xFF));
}
// читання даних long int з пам'яті EEPROM:
long readLongFromEEPROM(int address) {
    long number = 0;
    EEPROM.get(address, number);
    return number;
}
// генерування звукового сповіщення:
void zvyk(void) {
    pinMode(SPEAKER, OUTPUT);
    // Вмикаємо звук (ШИМ з заповненням 50%)
    analogWrite(SPEAKER, 127);
    // Затримка 2 секунди
    delay(2000);
    // Вимикаємо звук, подаючи логічний 0
    analogWrite(SPEAKER, 0);
}
```

```

// ідентифікатори пунктів меню
enum MenuItem {
    ITEM_HEIGHT = 0,
    ITEM_WEIGHT,
    ITEM_STEPS_ALARM,
    ITEM_CALORIE_ALARM,
    ITEM_MODE,
    MENU_TOTAL // Загальна кількість пунктів
};
// Масив назв пунктів (вирівняний для дисплея 16x2)
const char* menu_items[MENU_TOTAL] = {
    "  HEIGHT  ",
    "  WEIGHT  ",
    "ATTENTION STEPS",
    "ALARM CALORICITY",
    "  MODE AS/AC  "
};
int currentItem = 0; // Поточний обраний пункт
/* Головна функція відображення та керування меню */
void mainMenu(void) {
    // 1. Код для виводу menu_items[currentItem] на LCD дисплей
    // display_print(menu_items[currentItem]);
    // 2. Логіка обробки кнопок (спрощено)
    /*
    if (button_down_pressed) {
        currentItem = (currentItem + 1) % MENU_TOTAL;
    }
    if (button_up_pressed) {
        currentItem = (currentItem - 1 + MENU_TOTAL) % MENU_TOTAL;
    }
    if (button_select_pressed) {
        executeMenuItem(currentItem); // Викликаємо відповідне підменю
    }
    */
}
/* Виклик функцій підменю залежно від вибору */
void executeMenuItem(int item) {
    switch (item) {
        case ITEM_HEIGHT:    selectzrist();    break;
        case ITEM_WEIGHT:    selectvaga();     break;
        case ITEM_STEPS_ALARM: selectalarmkrok(); break;
        case ITEM_CALORIE_ALARM: selectalarmcalor(); break;
        case ITEM_MODE:      controlsc();     break;
    }
}
// --- Функції підменю ---
void selectzrist(void) { // Логіка встановлення зросту (напр. 150-220 см) }
void selectvaga(void) { // Логіка встановлення ваги (напр. 40-150 кг) }
void selectalarmkrok(void) { // Логіка: встановити ліміт кроків }
void selectalarmcalor(void) { // Контроль спалених калорій (max = 2750) }
void controlsc(void) { // Вибір режиму: контроль за кроками (AS) чи калоріями (AC) }

```

4.4. Розроблення головного програмного модуля функціонування цифрового крокоміра для контролю фізичної активності

Взаємодія мікроконтролера Arduino з сенсором ADXL345 для ідентифікації кроків реалізована програмно (додаток 2). Послідовність виконання операцій та алгоритмічні рішення ілюструються графічними матеріалами, поданими на Рис.4.1–4.11.

4.5. Дослідження моделі цифрового крокоміра для контролю фізичної активності в Proteus ISIS

Для прошивки Arduino Uno використовується скомпільований об'єктний файл (.hex), отриманий за допомогою штатного інструментарію Arduino Software. Перевірка пристрою здійснюється у два етапи: спочатку шляхом фізичного тестування контролера з алгоритмом крокоміра, а також за допомогою віртуальної симуляції в Proteus, що дозволяє виявити помилки в схемотехніці або коді до збирання заліза.

Графічні матеріали, подані нижче, відображають результати імітаційного моделювання пристрою. На рисунках показано роботу програмно-апаратної моделі педометра, побудованої на платформі Arduino з використанням сенсора ADXL345.

Для автоматизації розробки електроніки використовується програмний комплекс Proteus VSM. Він надає розробнику повний цикл інструментарію: відтворення та аналіз роботи схем у підпрограмі ISIS, а також подальшу підготовку конструкторської документації для виготовлення плат у модулі ARES. На Рис.4.12 продемонстровано результати моделювання моменту ввімкнення крокоміра. Візуалізовано процедуру самодіагностики та ініціалізації периферійних пристроїв, що відбувається під час старту програми. Процес ініціалізації пристрою передбачає налагодження роботи COM-порту, РК-індикатора та датчика прискорення. Програма зчитує збережені дані з пам'яті EEPROM, після чого виводить на дисплей службове повідомлення про готовність до роботи: “* Pedometr (Krokomir) *”. На етапі самодіагностики

здійснюється контрольне ввімкнення індикаторів D1, D2, D5 та пищика тривалістю 1 секунда. Окремо перевіряється стан акселерометра ADXL345. Результат тестування відображається на РКД та дублюється у COM-порт: повідомлення “ADXL345 error!” свідчить про збій, тоді як напис “ok” підтверджує готовність датчика до роботи.

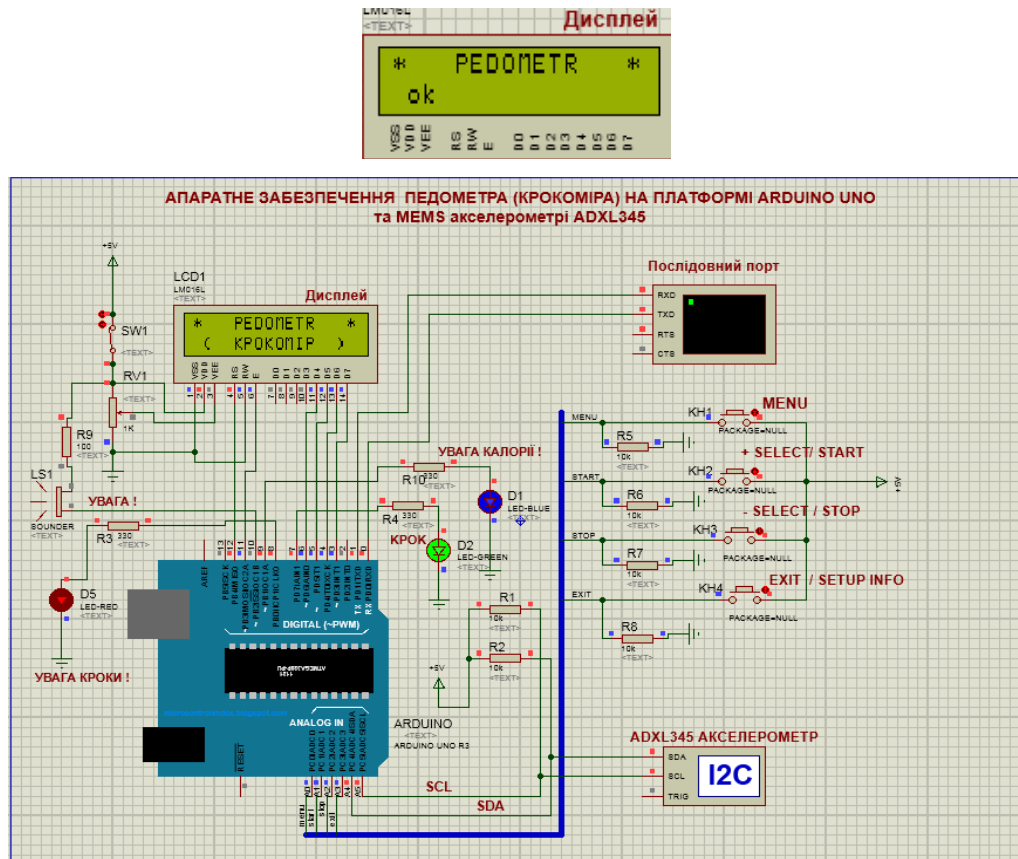


Рис. 4.12. Демонстрація етапу запуску та самодіагностики крокоміра при ввімкненні (симуляція в Proteus ISIS)

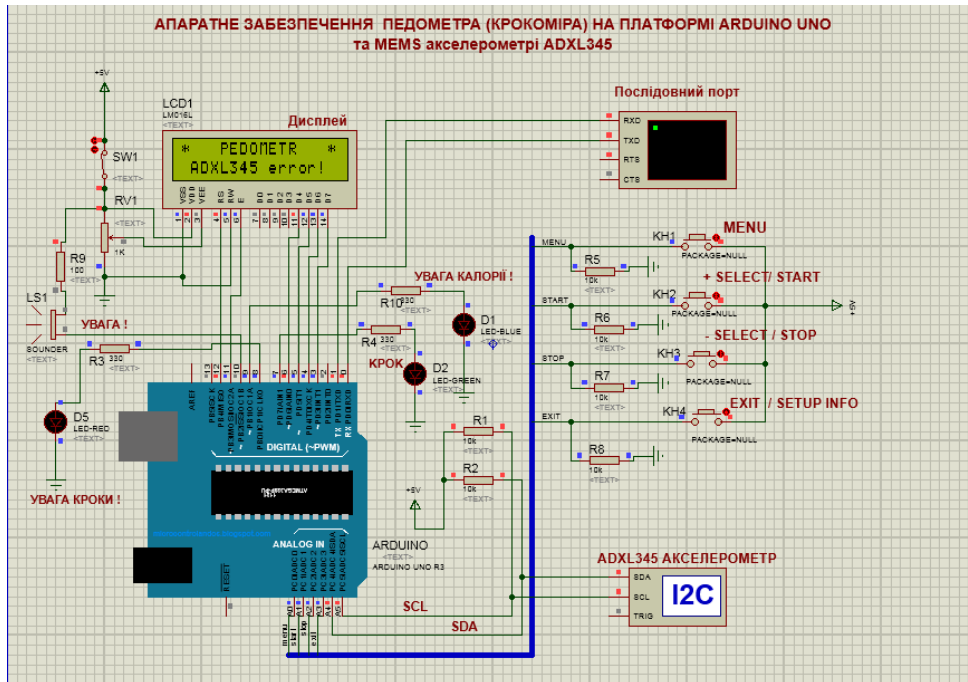


Рис. 4.13. Результат моделювання відмови датчика ADXL345 під час самодіагностики крокоміра

Після успішного проходження тесту крокомір здійснює зчитування значень проєкцій прискорення (a_x , a_y , a_z). Програмне забезпечення розраховує загальний вектор руху та виводить відповідну статистику через послідовний інтерфейс, як зображено на Рис. 4.15.

На рідкокристалічний екран виводяться динамічні дані про роботу приладу (Рис. 4.14). Користувачеві доступна інформація про кількість зроблених кроків ($N=1000$), спалені калорії ($K=23.0$) та пройдену відстань у метрах ($L=670$). Символ “S” на панелі інформує про активний режим очікування.

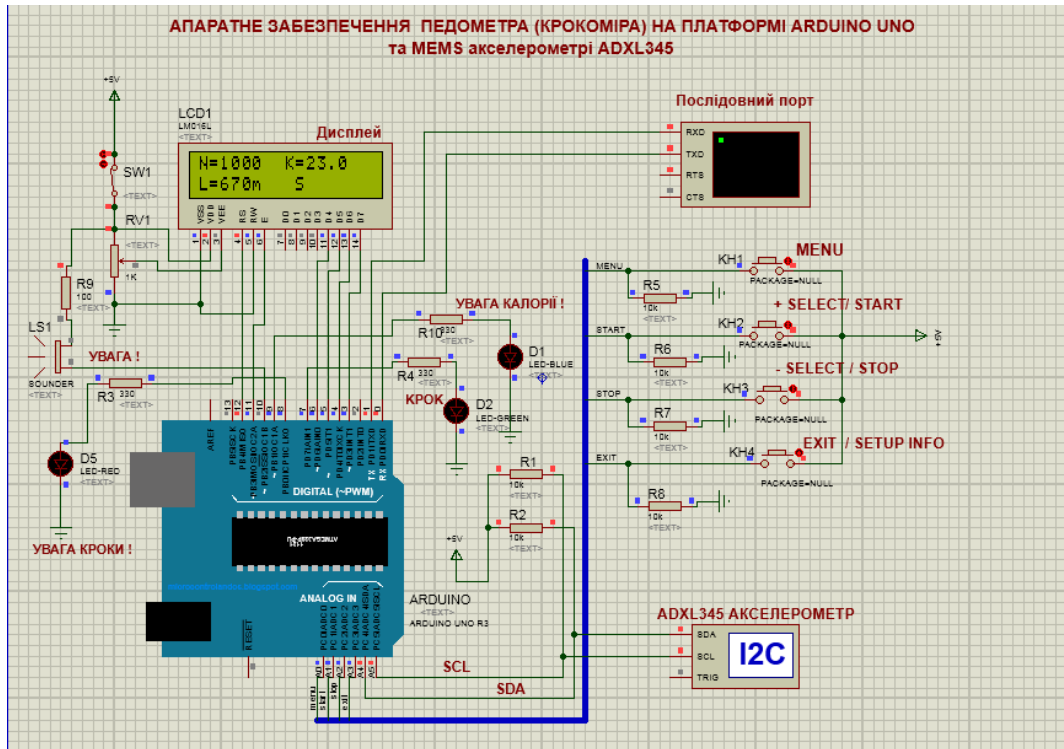


Рис. 4.14. Відображення параметрів педометра в режимі Standby (модельовання у середовищі Proteus ISIS)

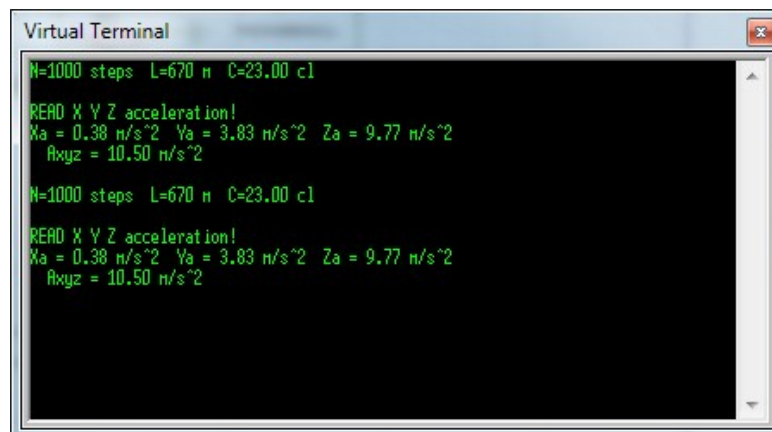


Рис. 4.15. Виведення службової інформації та поточних показників через послідовний інтерфейс (режим Standby)

Згідно з даними на Рис. 4.15, система фіксує проєкції прискорення вздовж координатних осей (X_a , Y_a , Z_a) та обчислює загальну величину вектора A_{xyz} . Щоб перевести педометр зі стану очікування до активної роботи, необхідно скористатися клавішею KH2 (“+SELECT/START”) (Рис. 4.16).

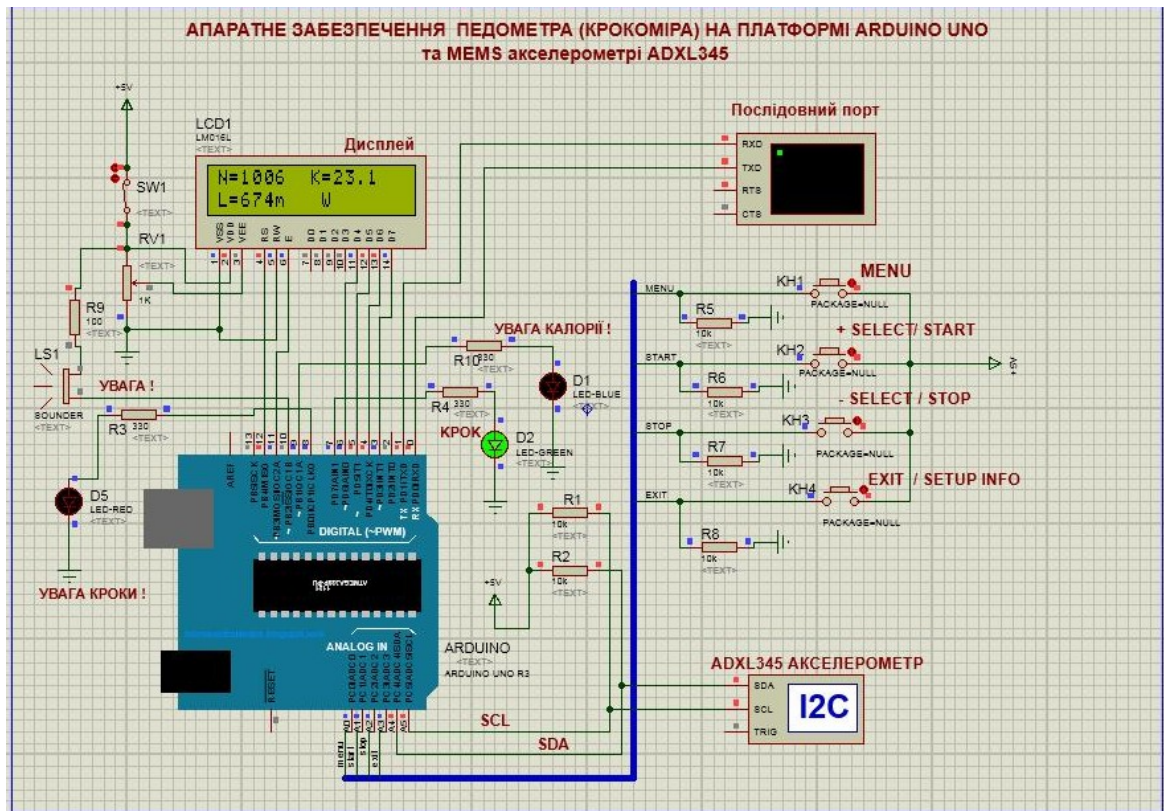


Рис. 4.16. Моделювання процесу реєстрації активності пристрою в середовищі Proteus ISIS

Після переходу в робочий режим на екрані з'являється індикатор "W". Система починає безперервний аналіз даних з акселерометра для детектування кроків, обчислення подоланого шляху та спалених калорій. Візуальне підтвердження кожного кроку реалізовано через мигтіння індикатора D2. Разом із цим програма веде постійний контроль встановлених користувачем порогів (прапорців) та реагує на маніпуляції з клавішами КН1 – КН4.

Перехід до режиму очікування (Standby) з активного стану здійснюється натисканням клавіші КН3. Для оперативного контролю конфігурації пристрою передбачена клавіша КН4, яка ініціює виведення короткочасного інформаційного повідомлення з довідковими даними на екран (Рис. 4.17). Ця функція доступна незалежно від поточного статусу роботи педометра.

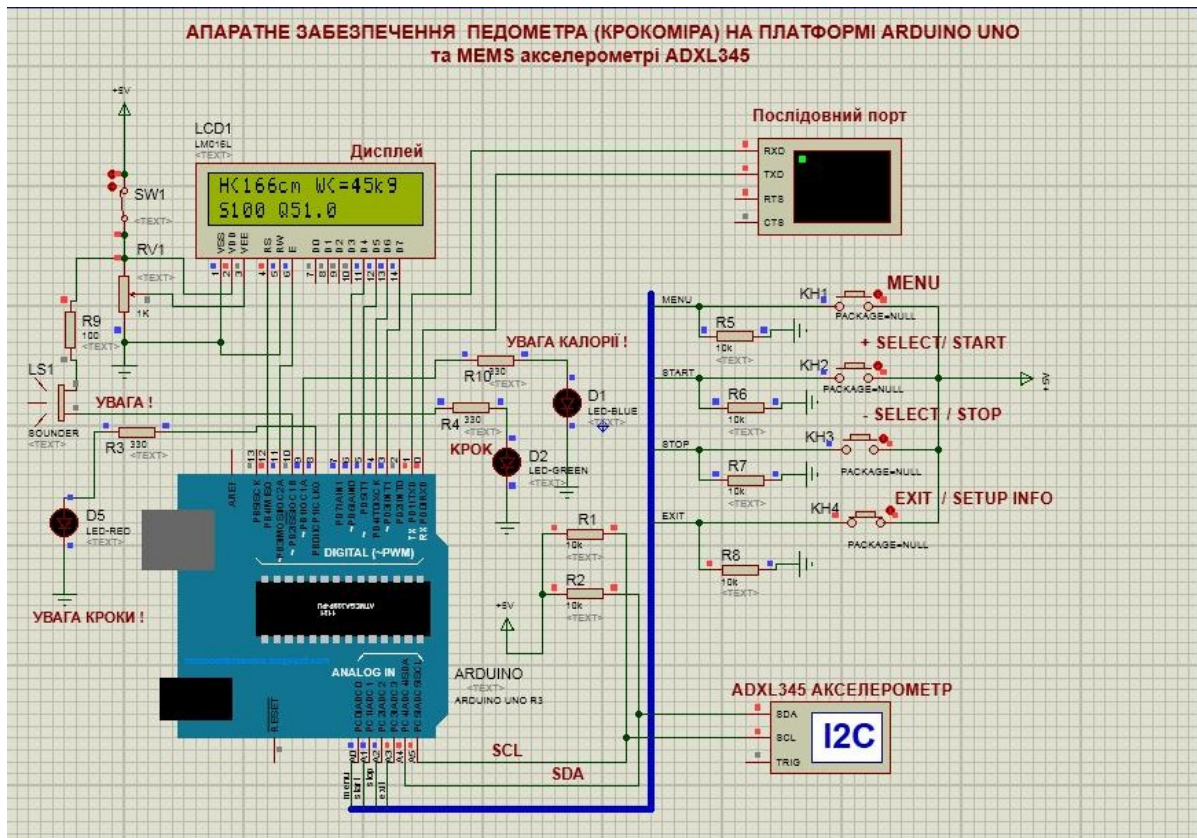


Рис. 4.17. Моделювання перегляду довідкових даних крокоміра у Proteus ISIS

Довідкова інформація крокоміра містить такі налаштування: параметр H вказує на зріст нижче 166 см, а W – на вагу менше 45 kg. Окрім персональних даних, відображаються прапорці контролю: ліміт за кількістю кроків (S100) та порогове значення спалених калорій (Q51.0).

Через послідовний інтерфейс виводиться схожа інформація про налаштування пристрою (Рис.4.18). До цих налаштувань належать фізичні характеристики (зріст 166 см, вага ≤ 45 кг) та контрольні ліміти (100 кроків і 51 калорія). Позначка OFF для режимів AS (Steps) та AC (Caloricity) підтверджує, що функції контролю обмежень вимкнені в поточній конфігурації.

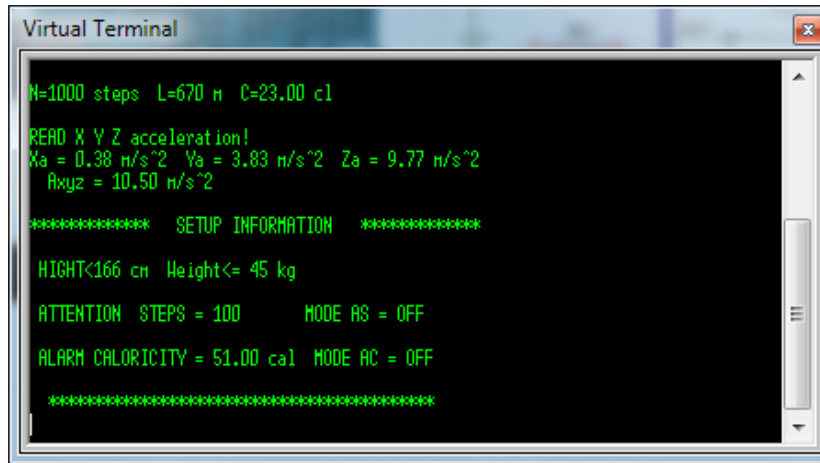


Рис. 4.18. Виведення налаштувань крокометра через інтерфейс UART

Завдяки довідковій інформації можна пересвідчитися, що пристрій налаштовано належним чином. Якщо поточні установки потребують коригування, користувач може скористатися відповідним розділом меню. Перехід до режиму редагування параметрів виконується натисканням кнопки КН1 (“MENU”) (Рис.4.19).

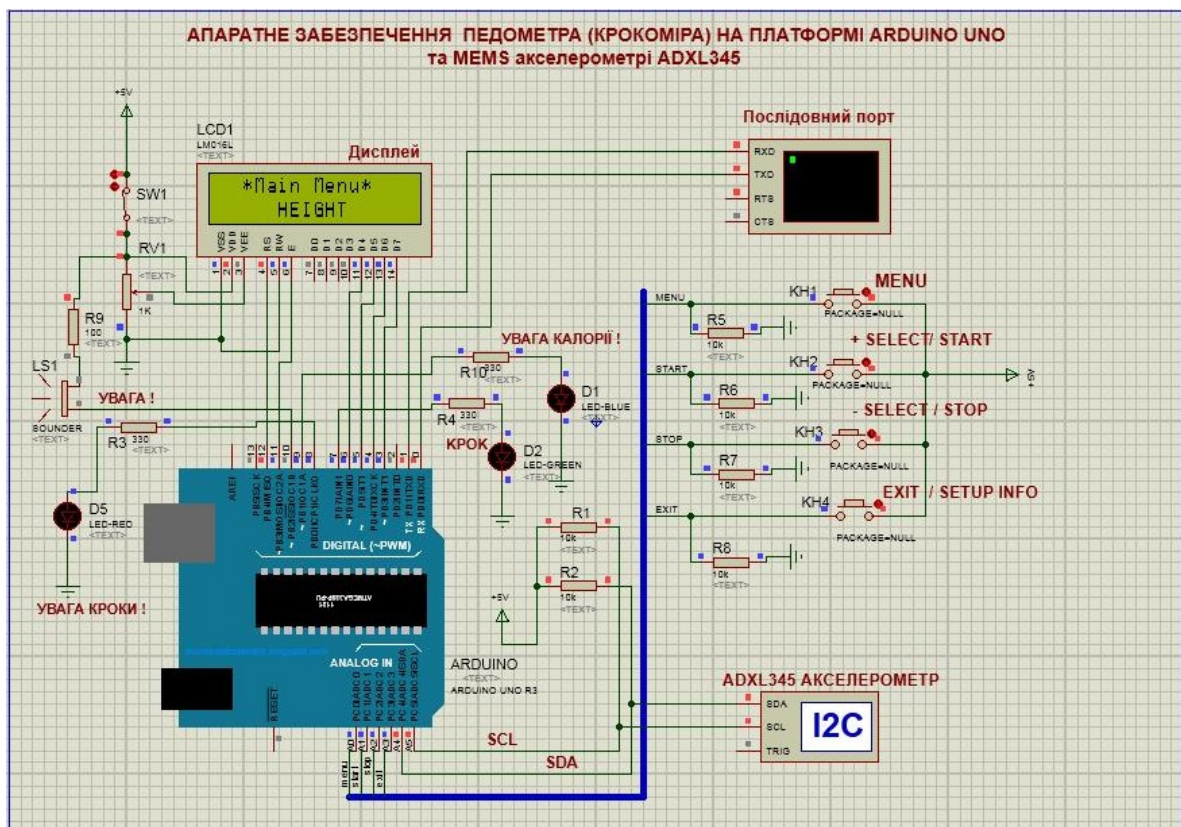


Рис. 4.19. Меню налаштувань крокометра (моделювання в Proteus)

Меню налаштувань пристрою (Рис. 4.19) містить п'ять основних пунктів. Перші два (“HEIGHT” та “WEIGHT”) призначені для введення персональних фізичних параметрів. Пункти “ATTENTION STEPS” та “ALARM CALORICITY”

дозволяють встановити контрольні межі активності. Завершальний пункт “MODE AS/AC” відповідає за вибір логіки роботи системи оповіщення та ввімкнення відповідних програмних прапорців.

Навігація за пунктами меню виконується за допомогою клавіш “+SELECT/START” та “-SELECT/STOP”, які відповідають за перехід між налаштуваннями. Активація вибраного пункту здійснюється натисканням “MENU”, а для повернення до попереднього режиму або завершення роботи з налаштуваннями передбачена кнопка “EXIT/SETUP INFO”. Для забезпечення точності вимірювань необхідно попередньо ввести персональні дані (зріст та вагу). Ці показники є базовими для алгоритму, оскільки вони визначають довжину кроку та інтенсивність спалювання калорій. Програмна логіка базується на статистичних даних із Табл. 1.1–1.3, що інтегровані в пам’ять мікроконтролера. Приклад встановлення зросту в підменю “HEIGHT” наведено на Рис. 4.20.

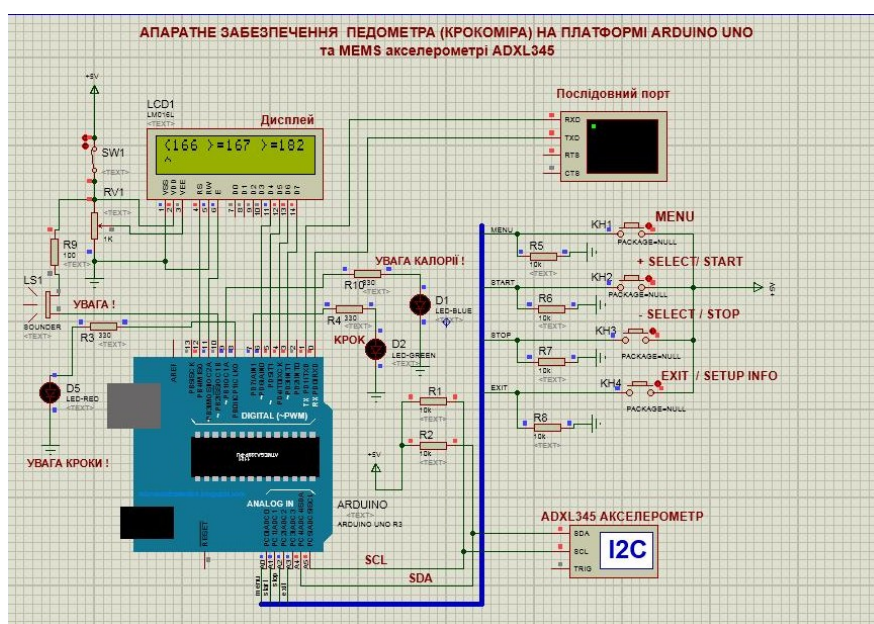
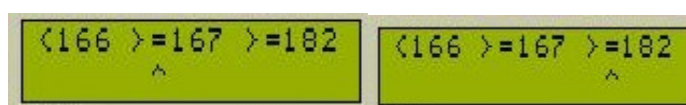


Рис. 4.20. Відображення інтерфейсу налаштування антропометричних даних. Позиціювання курсора реалізовано через клавіші вибору (“+SELECT/START”/”-SELECT/STOP”)



Для збереження встановленого показника зросту користувач має натиснути клавішу “MENU”. Скасування операції або вихід із поточного підменю реалізовано за допомогою клавіші “EXIT/SETUP INFO”. Наступним етапом конфігурування є встановлення ваги у розділі меню “WEIGHT” (Рис.4.21).

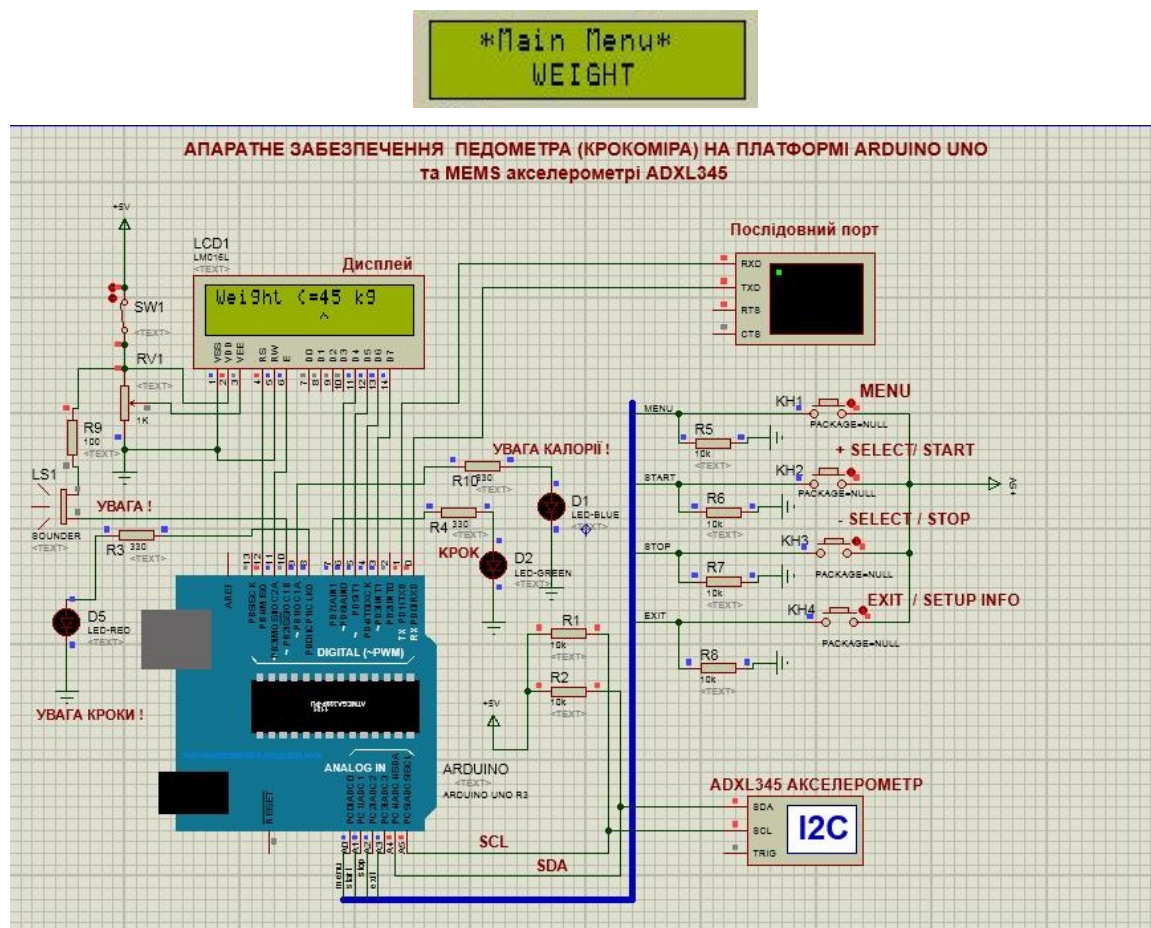


Рис. 4.21. Моделювання режиму введення вагових параметрів для коректного розрахунку енерговитрат користувача (поліцейського)

Вибір ваги реалізовано шляхом перебору встановленого ряду значень (у кг): {≤45; 55; 64; 73; 82; 91; 100; 114; 125}. Перемикання між цими величинами виконується функціональними клавішами “+SELECT/START” та “-SELECT/STOP”.



Збереження встановленої ваги виконується клавішею підтвердження “MENU”, а вихід із поточного вікна реалізовано через кнопку “EXIT/SETUP INFO”. Подальше налаштування приладу передбачає перехід до підменю

“ATTENTION STEPS”, де користувач може задати контрольну межу пройденої дистанції у кроках (Рис. 4.22).



Як тільки поточне значення кроків стає більшим за заданий поріг, пристрій переходить у режим сигналізації. Це супроводжується візуальною індикацією (миготіння червоного світлодіода) та акустичним супроводом за допомогою звукового сигналізатора.

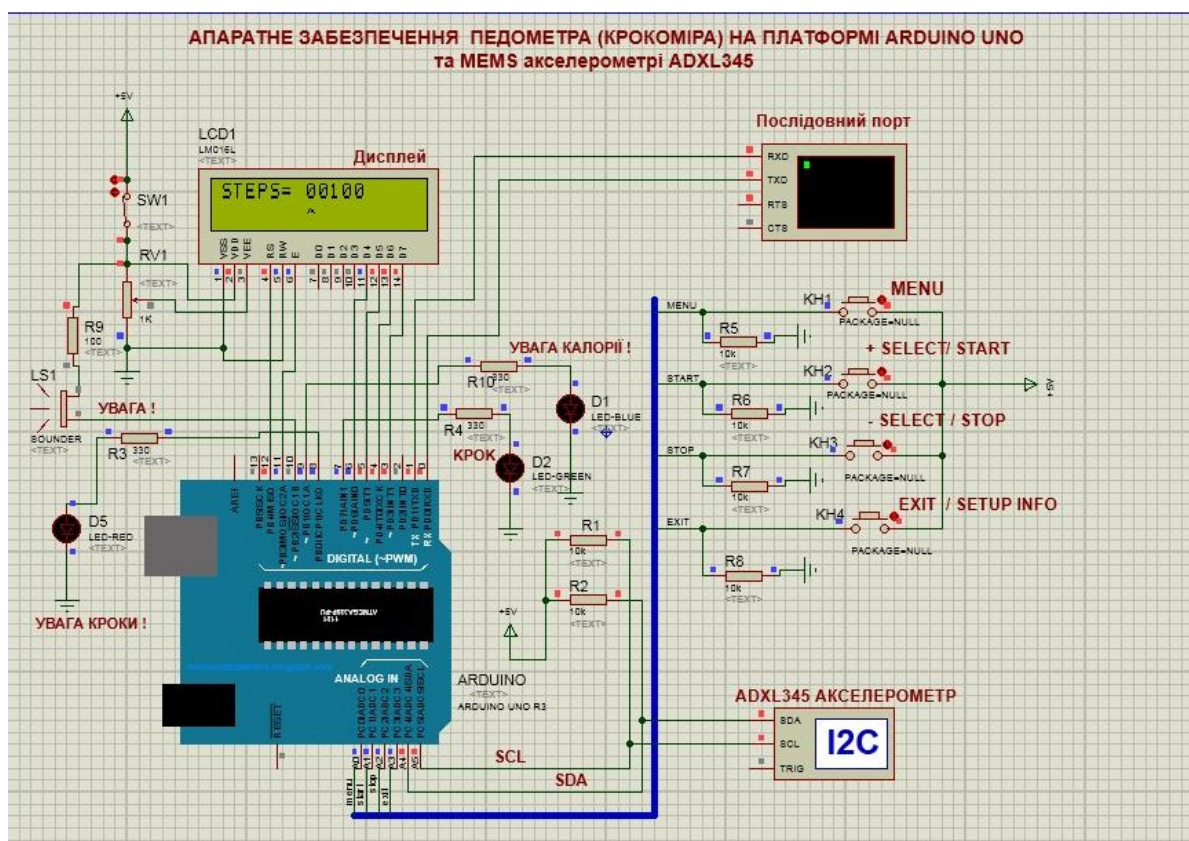
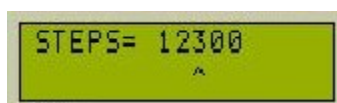



Рис. 4.22. Моделювання інтерфейсу встановлення контрольного ліміту кроків для активації системи оповіщення

Навігація між розрядами числа реалізована за допомогою клавіші “MENU”, яка забезпечує циклічне переміщення курсора. Коригування цифрових значень в активній позиції здійснюється кнопками інкременту/декременту — “+SELECT/START” та “-SELECT/STOP”.



Для збереження параметрів у пам'ять пристрою та повернення до ієрархії меню використовується кнопка "EXIT/SETUP INFO". Ідентичний порядок дій

передбачений і для пункту "ALARM CALORICITY" , де користувач має змогу визначити цільовий показник енерговитрат (Рис. 4.23).

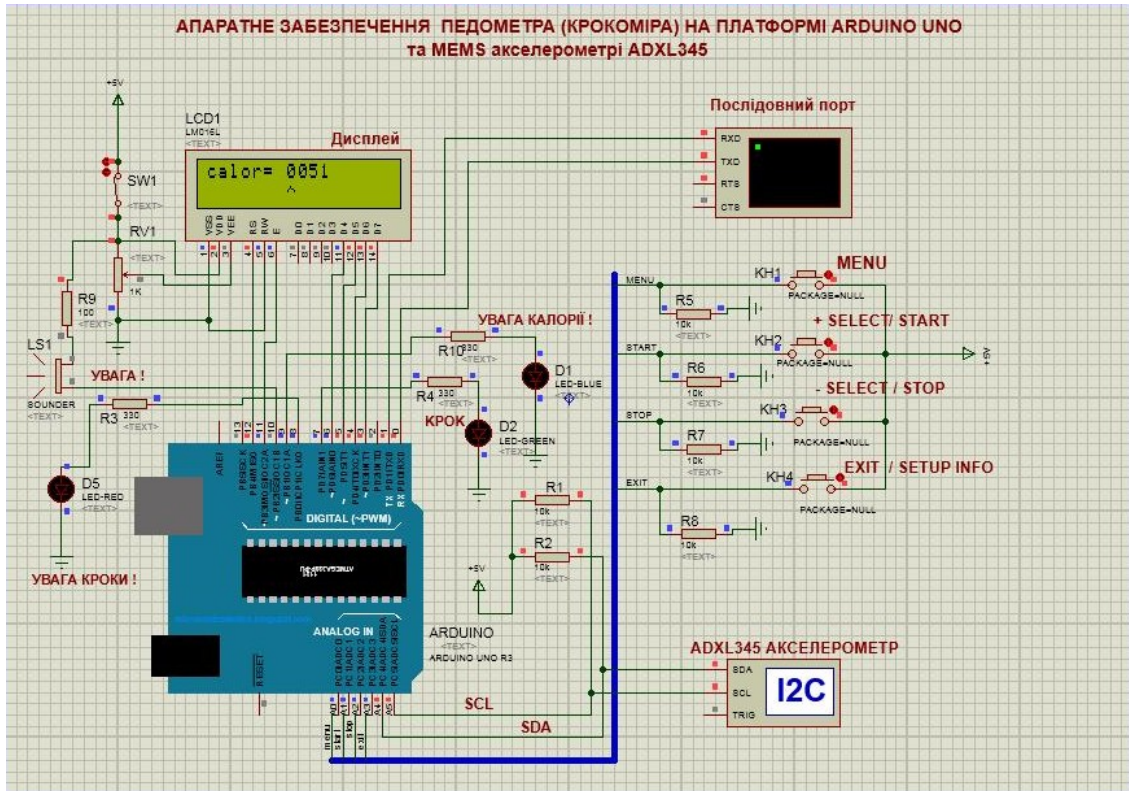



Рис. 4.23. Відображення інтерфейсу налаштування цільового показника спалених калорій "ALARM CALORICITY"

Як тільки розрахована кількість спалених калорій стає більшою за задану межу, педометр ініціює попереджувальний сигнал. Це супроводжується візуальною індикацією за допомогою синього світлодіода та акустичним супроводом (звуковий сигналізатор).

Пункт меню "MODE AS/AC"  призначений для конфігурування системи сповіщень. Користувач може активувати контроль кроків, контроль калорій або вимкнути будь-які попередження, вибравши відповідний режим роботи пристрою (Рис. 4.24).

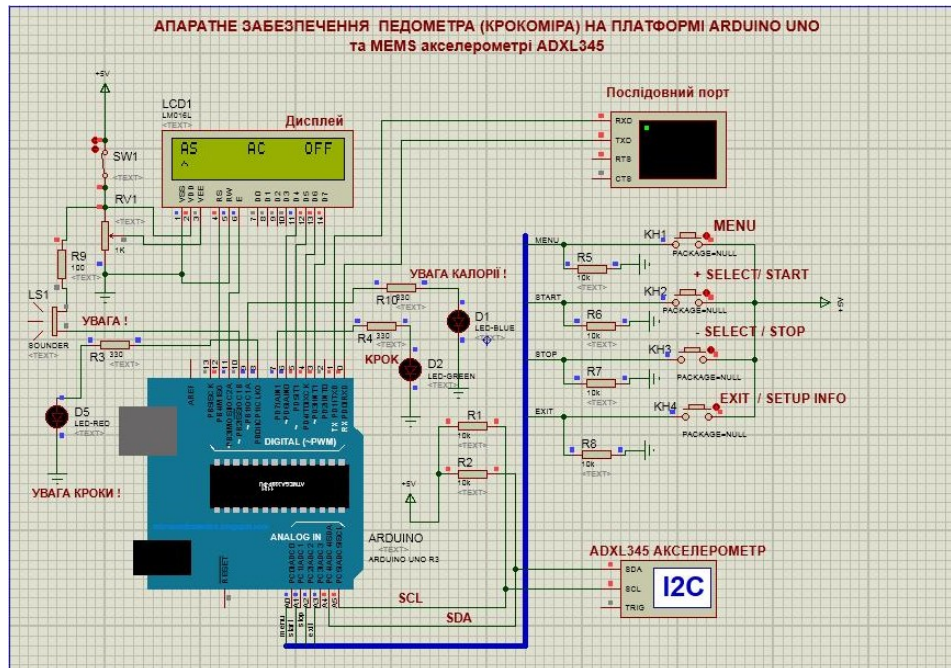


Рис. 4.24. Налаштування режиму контролю в “MODE AS/AC”

Навігація між режимами контролю здійснюється за допомогою клавіш інкременту та декременту (“+SELECT/START”/“-SELECT/STOP”). Система дозволяє встановити контроль кроків (AS), калорій (AC) або повністю вимкнути цю функцію (OFF). Підтвердження обраного параметра виконується кнопкою “MENU”, тоді як вихід до попереднього рівня ієрархії меню реалізовано через клавішу “EXIT/SETUP INFO”. Поточний статус контролю виводиться на РК-екран у вигляді відповідних позначень (Рис. 4.25 та 4.26).

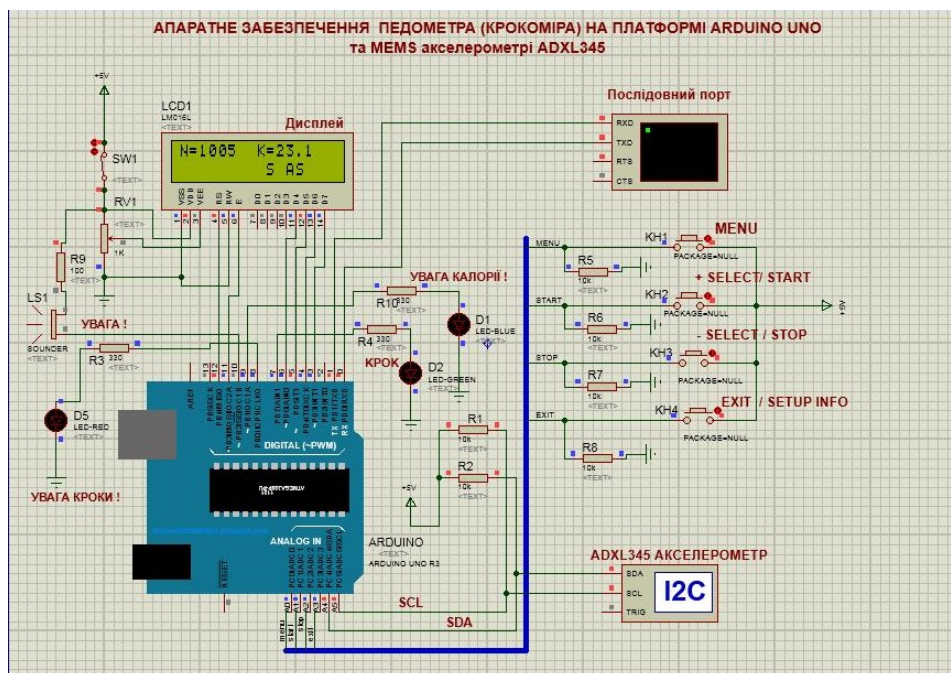


Рис. 4.25. Режим підрахунку кроків (AS) у вікні симуляції крокоміра

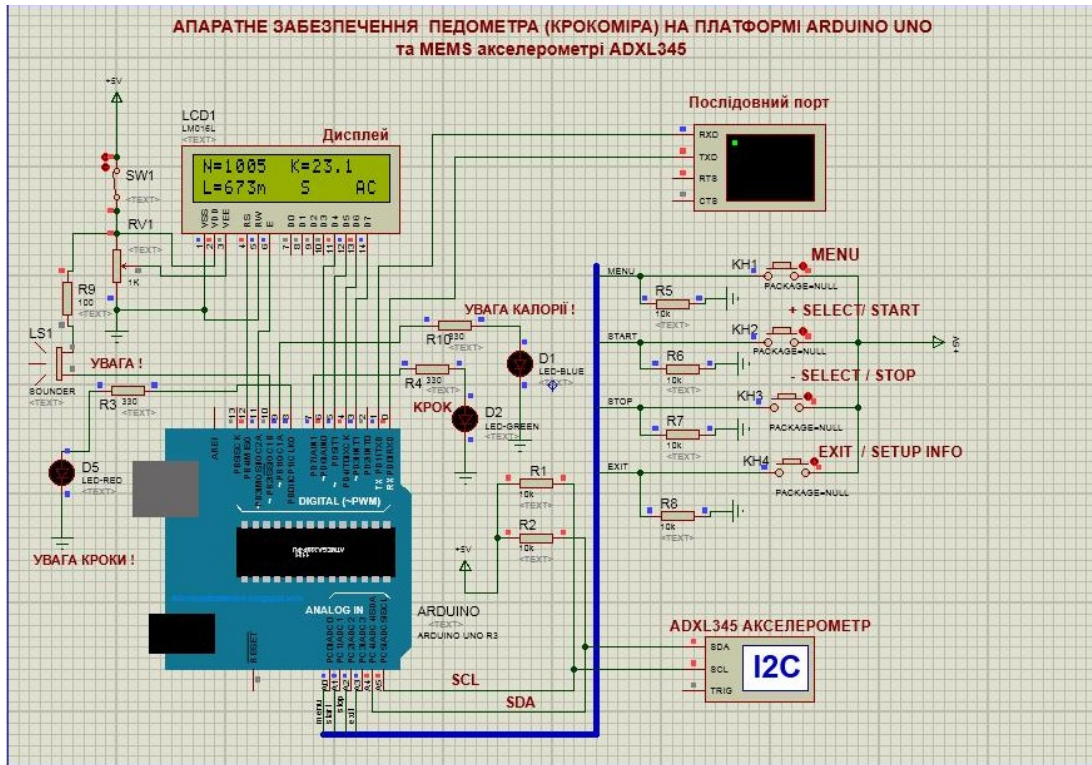


Рис. 4.26. Відображення параметрів спалених калорій у вікні моделювання Proteus ISIS при виборі опції контролю “AC”

Для оптимізації живлення в режимі детектування кроків використовується перемикач SW1. З його допомогою можна відключити індикацію на РКД, не перериваючи роботу основного алгоритму.

ВИСНОВКИ

Результатом роботи є створення спеціалізованого пристрою для підрахунку кроків, адаптованого до умов професійної підготовки поліцейських. Апаратна частина інтегрує сенсор ADXL345 (триосьовий акселерометр) та обчислювальний модуль Arduino, що забезпечує точне відстеження динамічних параметрів руху.

Апаратна частина педометра реєструє прискорення по осях X, Y, Z, а програмний алгоритм обробляє отриманий вектор для виявлення моментів наступання (кроків). Окрім кількісного обліку, система реалізує функції фітнес-трекера: розрахунок дистанції та енергообміну. Коректність цих обчислень гарантується можливістю внесення індивідуальних параметрів тіла через меню налаштувань.

Інтерфейс педометра базується на символічному РК-модулі, що забезпечує користувача повною статистикою тренування (кількість кроків, шлях, енерговитрати). Система оповіщення працює у комбінованому режимі — через візуальні індикатори та звукові сигнали. Важливою особливістю є режим енергозбереження, який дозволяє відключати вивід інформації на дисплей для продовження автономної роботи пристрою.

Завдяки використанню доступної елементної бази, собівартість педометра є значно нижчою за ринкові пропозиції при вищому рівні функціональності. Програмно-апаратна архітектура підтримує багаторівневе меню керування, декілька режимів експлуатації та автоматичне збереження користувацьких пресетів у внутрішній пам'яті мікроконтролера. Наявність діагностичного виводу в COM-порт та розвиненої системи індикації робить пристрій універсальним інструментом контролю активності.

Конструкція крокоміра реалізована на базі сучасної мікроконтролерної техніки. Процес збору та обробки інформації забезпечується мікросхемою ATmega328, тоді як за реєстрацію фізичної активності відповідає цифровий акселерометр ADXL345. Візуалізація результатів вимірювання та взаємодія з користувачем здійснюється за допомогою стандартного дисплея LCD 1602.

Процес проектування включав створення віртуальної моделі педометра та його схематичного рішення в пакеті Proteus. Логіку функціонування пристрою описано за допомогою алгоритму, що ліг у основу програми для МК ATmega (платформа AVR). Написання та компіляція коду здійснювалися в інструментарії Arduino IDE з використанням мови програмування C.

Програмне забезпечення пристрою включає підсистему зчитування та обробки даних з датчика прискорення та гнучкий механізм конфігурування параметрів через вбудоване меню.

Етап тестування в емуляційному пакеті Proteus VSM продемонстрував повну відповідність розробленого коду поставленим задачам. Моделювання підтвердило коректність обчислювальної логіки педометра та надійність взаємодії програмних компонентів із віртуальною моделлю пристрою.

Виконання роботи дозволило оволодіти ключовими компетенціями з програмування мікроконтролерів та побудови інтелектуальних систем моніторингу активності. Набуто навичок роботи з триосьовими акселерометрами, реалізації складних обчислювальних алгоритмів мовою C/C++ та верифікації роботи апаратно-програмних комплексів.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Розрахунок енергетичних витрат при різних видах фізичної активності [Електронний ресурс]. – Доступ: <https://group-health.com/blog/post/how-many-calories-are-burned-with-physical-activit>, <https://www.dhs.wisconsin.gov/publications/p4/p40109.pdf>.
2. Методологія конвертації кількості кроків у калорії [Електронний ресурс]. – Доступ: <https://www.verywellfit.com/pedometer-steps-to-calories-converter-38-82595>.
3. Оздоровчі аспекти ходьби: досягнення цілі у 10 000 кроків [Електронний ресурс]. – Доступ: https://pwc.ottawaheart.ca/sites/pwc.ottawaheart.ca/files/wellness/getting_to_10000_steps.pdf.
4. Огляд сучасних моделей побутових крокомірів (педометрів) [Електронний ресурс]. – Доступ: <https://med-dobrik.com.ua/krokomir-tfa-hitrax-walk.html>, <https://insportline.com.ua/277121-cifrovij-krokomir-insportline-strippy-ze.html>.
5. Система автоматизованого проектування Proteus VSM: [Електронний ресурс]. – Доступ: <https://www.labcenter.com/downloads/>.
6. Програмне середовище розробки Arduino IDE [Електронний ресурс]. – Доступ: <https://www.arduino.cc/en/Main/Software>.
7. Технічні специфікації платформи ARDUINO UNO на МК ATmega328 [Електронний ресурс]. – Доступ: <https://store.arduino.cc/products/arduino-uno-rev3>, <https://components101.com/microcontrollers/arduino-uno>.
8. Datasheet МК ATmega328P: архітектура та опис виводів [Електронний ресурс]. – Доступ: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.
9. Посібник з підключення та керування символьними РК-дисплеями 1602А [Електронний ресурс]. – Доступ: <https://lastminuteengineers.com/arduino-1602-character-lcd-tutorial/>.
10. Повний технічний опис (Data Sheet) акселерометра ADXL345 [Електронний ресурс]. – Доступ: <https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf>.

11. Banzi M. Getting Started with Arduino. – 2nd Ed. O’Reilly Media, 2011.–130 p.
12. Banzi M., Shiloh M. Getting Started With Arduino: The Open Source Electronics Prototyping Platform. – 4th ed. Make Community, LLC, 2022. – 285 p.
13. Monk S. Programming Arduino: Getting Started with Sketches. – 3rd Ed. McGraw-Hill TAB, 2022. – 176 p.
14. Blum J. Exploring Arduino: Tools and Techniques for Engineering Wizardry. – 2nd ed. Wiley, 2019. – 512 p.
15. Geddes M. Arduino Project Handbook: 25 Practical Projects to Get You Started. – No Starch Press, 2016. – 272 p.
16. Margolis M. Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects. – 2nd Ed. O’Reilly Media, 2012. – 721 p.
17. Purdum J. Beginning C for Arduino: Learn C Programming for the Arduino. – 2nd ed. Apress, 2015. – 414 p.
18. Dunbar N. Arduino Software Internals: A Complete Guide. – 2nd ed. Apress, 2024. – 717 p.
19. Методика інтеграції акселерометра ADXL345 з платформною Arduino Uno [Електронний ресурс]. – Доступ: <https://circuitdigest.com/microcontroller-projects/interface-adxl345-accelerometer-with-arduino-uno>.
20. Розробка цифрового крокоміра на базі Arduino [Електронний ресурс]. – Доступ: <https://www.engineersgarage.com/arduino-based-walking-steps-distance-calculator-adxl345/>.
21. Проектування портативних пристроїв моніторингу активності [Електронний ресурс]. – Доступ: <https://electronics-projects.in/pedometersteps-counter-using-arduino-adxl345-accelerometer-sensor/>.
22. Освітні платформи та онлайн-курси з програмування мікроконтролерів [Електронний ресурс]. – Доступ: <https://www.arduino.cc/en/Tutorial/HomePage>, <https://learn.adafruit.com/category/learn-arduino>, <https://www.tutorialspoint.com/arduino/>.

ДОДАТКИ

Додаток 1

Програмний модуль роботи з акселерометром: Лістинг основних функцій для опитування та конфігурування MEMS-датчика ADXL345

```
/* Читання одного байта з регістра ADXL345
 * @param reg_addr - адреса регістра
 * @return отримане значення (uint8_t) */
uint8_t ADXL345_ReadByte(uint8_t reg_addr) {
    uint8_t data;
    // Фаза 1: Встановлення адреси регістра (Write sequence)
    i2c_start();
    i2c_write(ADXL345_W); // Адреса пристрою + біт запису
    i2c_write(reg_addr); // Вказуємо, який регістр хочемо прочитати
    // Фаза 2: Читання даних (Repeated Start)
    i2c_start(); // Повторний старт без звільнення шини
    i2c_write(ADXL345_R); // Адреса пристрою + біт читання
    data = i2c_read(0); // Читаємо байт з NACK (завершення транзакції)
    i2c_stop();
    return data;
}

/* Читання двобайтового значення (16-біт) з ADXL345
 * @param address - адреса першого регістра (наприклад, DATA0)
 * @return ціле число (int16_t) */
int16_t ADXL345_ReadInt(uint8_t address) {
    uint8_t lsb, msb;
    i2c_start(); i2c_write(ADXL345_W);
    i2c_write(address); // Вказуємо адресу початкового регістра
    i2c_start(); // Повторний старт
    i2c_write(ADXL345_R);
    // ADXL345 передає дані у форматі: Молодший байт (LSB), потім Старший (MSB)
    lsb = i2c_read(1); // Читаємо LSB і надсилаємо ACK (продовжуємо читання)
    msb = i2c_read(0); // Читаємо MSB і надсилаємо NACK (завершуємо читання)
    i2c_stop();
    // Збираємо 16-бітне число
    // Використовуємо знакові типи для коректної обробки від'ємних прискорень
    return (int16_t)((msb << 8) | lsb);
}

/* Запис одного байта в регістр ADXL345
 * @param reg_addr - адреса регістра
 * @param data - значення для запису */
void ADXL345_WriteByte(uint8_t reg_addr, uint8_t data) {
    i2c_start();
    // Надсилаємо адресу пристрою з бітом запису
    i2c_write(ADXL345_W);
    // Вказуємо адресу регістра, куду хочемо записати дані
    i2c_write(reg_addr);
    // Надсилаємо саме значення
    i2c_write(data);
    // Завершуємо транзакцію
    i2c_stop();
}
```

```

}

/* Ініціалізація ADXL345
 * Переводить сенсор з режиму очікування у режим активного вимірювання */
void ADXL345_Init(void) {
    // 1. Встановлюємо режим Standby (скидання бітів)
    // Це гарантує, що попередні налаштування не завадять запуску
    ADXL345_WriteByte(ADXL345_REG_POWER_CTL, 0x00);
    // Невелика пауза для стабілізації внутрішньої логіки (за бажанням)
    delay_ms(5);
    // 2. Встановлюємо біт Measure (D3) у регістрі POWER_CTL (0x2D)
    // 1 << 3 — це запис значення 0x08, що активує режим вимірювання
    ADXL345_WriteByte(ADXL345_REG_POWER_CTL, (1 << 3));
    // Додатково: рекомендується налаштувати формат даних (діапазон)
    // Наприклад, +/- 2g, 10-бітний режим (регістр DATA_FORMAT 0x31)
    // ADXL345_WriteByte(0x31, 0x00);
}
// Визначення значень для вибору діапазону (згідно з даташитом)
typedef enum {
    RES_2G = 0x00, RES_4G = 0x01, RES_8G = 0x02, RES_16G = 0x03
} ADXL_Range;

/* Налаштування діапазону вимірювання (2-4-8-16 g)
 * Вмикає Full Resolution для стабільної чутливості 4 мг/LSB */
void ADXL345_SetResolution(ADXL_Range range) {
    // 0x08 — це біт FULL_RES (D3)
    // range — це молодші два біти (D0, D1)
    uint8_t format_value = 0x08 | range;
    i2c_start(); i2c_write(ADXL345_W); i2c_write(ADXL345_REG_DATA_FORMAT);
    i2c_write(format_value); i2c_stop();
}

void readAccel(void) {
    // 1. Вказуємо початкову адресу (DATA_X0)
    Wire.beginTransaction(0x53); Wire.write(0x32); Wire.endTransmission(false); // false для repeated
    start (краще для стабільності)
    // 2. Запитуємо одразу 6 байтів (X0, X1, Y0, Y1, Z0, Z1)
    Wire.requestFrom(0x53, 6);
    if (Wire.available() <= 6) {
        x0 = Wire.read(); /* X молодший */ x1 = Wire.read(); /* X старший */
        y0 = Wire.read(); /* Y молодший */ y1 = Wire.read(); /* Y старший */
        z0 = Wire.read(); /* Z молодший */ z1 = Wire.read(); /* Z старший */
    }
}
/* Обчислення коефіцієнтів калібрування (зміщення та чутливість)
 * @param D - очікуване значення прискорення вільного падіння (1.0g або 9.81) */
void ADXL345_CalculateCalibration(float D) {
    // 1. Обчислення зміщення (Zero-g offset)
    // Це середня точка між максимумом і мінімумом.
    // В ідеалі вона має бути 0 для X та Y.
    xoffset = (xmax + xmin) / 2.0; yoffset = (ymax + ymin) / 2.0;
    zoffset = (zmax + zmin) / 2.0;
    // 2. Обчислення коефіцієнта чутливості (Sensitivity Scale Factor)
    // Використовуємо різницю (xmax - xmin).
    // Формула: ((max - min) / 2) — це амплітуда 1g у одиницях ADC.
}

```

```
// Потім ділимо D на цю амплітуду, щоб отримати множник.  
xsense = D / ((xmax - xmin) / 2.0); ysense = D / ((ymax - ymin) / 2.0);  
zsense = D / ((zmax - zmin) / 2.0);  
}  
/* Застосування калібрування до сирих даних */  
float getAccelX(int x_raw) {  
    return (x_raw - xoffset) * xsense;  
}
```

Додаток 2

Фрагмент програмної реалізації головного алгоритму роботи цифрового крокоміра

```
void setup() {
  // 1. Пряме керування регістрами напрямку (швидке налаштування портів)
  DDRD |= 0x78; // Pins 3, 4, 5, 6 як OUTPUT (LCD)
  DDRB |= 0x18; // Pins 11, 12 як OUTPUT (LCD)
  // 2. Налаштування стандартних пінів (LED та кнопки)
  pinMode(RED, OUTPUT); pinMode(BLUE, OUTPUT); pinMode(GREEN, OUTPUT);
  pinMode(KH1, INPUT); /* MENU */ pinMode(KH2, INPUT); /* Select+ */
  pinMode(KH3, INPUT); /* Select- */ pinMode(KH4, INPUT); /* EXIT */
  // 3. Читання даних з енергонезалежної пам'яті (EEPROM)
  rist = EEPROM.read(1); vaga = EEPROM.read(3); askrok = readLongFromEEPROM(10);
  long acalx = readLongFromEEPROM(20); accalor = (float)acalx;
  // 4. Ініціалізація інтерфейсів
  Serial.begin(9600); Wire.begin(); lcd.begin(16, 2);
  // Калібрування акселерометра (ваша кастомна функція)
  calibration();
  // 5. Стартове вітання та тест LED
  lcd.setCursor(0, 0); lcd.print(" Pedometr "); lcd.setCursor(0, 1);
  lcd.print(" КРОКОМІР ");
  // Тест світлодіодів (RGB тест)
  digitalWrite(RED, HIGH); digitalWrite(BLUE, HIGH); digitalWrite(GREEN, HIGH); delay(1000);
  digitalWrite(RED, LOW); digitalWrite(BLUE, LOW); digitalWrite(GREEN, LOW);
  // 6. Ініціалізація ADXL345
  Wire.beginTransmission(0x53); Wire.write(0x00); /* Запит Device ID */
  Wire.endTransmission(); Wire.requestFrom(0x53, 1); byte id = Wire.read();
  lcd.setCursor(0, 1);
  if (id == 0xE5) {
    lcd.print("ID: 0x"); lcd.print(id, HEX); lcd.print(" OK ");
    // Конфігурація акселерометра одним блоком
    setRegisterADXL(0x2C, 0x08); // BW_RATE: 25Hz
    setRegisterADXL(0x31, 0x08); // DATA_FORMAT: Full Res, +/- 2g
    setRegisterADXL(0x2D, 0x08); // POWER_CTL: Measurement Mode
  } else {
    lcd.print("ADXL Error! "); Serial.println("ADXL345 connection failed!"); }
  delay(1000); lcd.setCursor(0, 1); lcd.print(" "); // Очищення нижнього рядка
}
// Допоміжна функція для запису в регістри ADXL345
void setRegisterADXL(byte reg, byte val) {
  Wire.beginTransmission(0x53); Wire.write(reg); Wire.write(val); Wire.endTransmission(); }

void loop() {
  // --- 1. Читання та фільтрація даних акселерометра ---
  long sumX = 0, sumY = 0, sumZ = 0;
  for (int i = 0; i < num_sample; i++) {
    readAccel(); // Функція заповнює глобальні x0, x1, y0, y1, z0, z1
    // Перетворення 10-бітних значень (додатковий код)
    int16_t x = ((x1 << 8) | x0) & 0x03FF; if (x > 511) x -= 1024;
    int16_t y = ((y1 << 8) | y0) & 0x03FF; if (y > 511) y -= 1024;
    int16_t z = ((z1 << 8) | z0) & 0x03FF; if (z > 511) z -= 1024;
```

```

    sumX += x; sumY += y; sumZ += z; delay(10);
}
// Обчислення середнього та реального прискорення (m/s^2)
float xr = ((float)(sumX / num_sample) - xoffset) / xsense;
float yr = ((float)(sumY / num_sample) - yoffset) / ysense;
float zr = ((float)(sumZ / num_sample) - zoffset) / zsense;
axyz = sqrt(xr * xr + yr * yr + zr * zr);
// --- 2. Логіка підрахунку кроків (Тільки якщо flag == 1 - START) ---
if (flag == 1) {
    if (axyz > threshold && flagstep == 0) {
        krok++;
        flagstep = 1;
        digitalWrite(GREEN, HIGH); // Індикація кроку
        delay(10); digitalWrite(GREEN, LOW);
    }
    else if (axyz < (threshold - 0.2)) { // Гістерезис для стабільності
        flagstep = 0;
    }
}
// --- 3. Розрахунок параметрів (Виконуємо лише за потреби) ---
// Визначення коефіцієнта довжини кроку (lk)
if (rist >= 182) lk = lakrok; else if (rist >= 167) lk = lbckrok;
else lk = lddkrok;
// Визначення індексу ваги (tv)
const int w_limits[] = {45, 55, 64, 73, 82, 91, 100, 114, 125}; tv = 0;
for (int i = 0; i < 9; i++) {
    if (vaga >= w_limits[i]) tv = i + 1; }
// Розрахунок калорій та дистанції
byte height_offset = (rist >= 182) ? 0 : (rist >= 167 ? 9 : 18);
cal = masuvcal[height_offset + (tv > 0 ? tv - 1 : 0)];
calor = (cal * 0.001) * krok; lm = (unsigned int)krok * lk;
// --- 4. Оновлення інтерфейсу (LCD та Serial) ---
static unsigned long lastUpdate = 0;
if (millis() - lastUpdate > 300) { // Оновлення кожні 300 мс (проти мерехтіння)
    lcd.clear(); lcd.setCursor(0, 0); lcd.print("N:"); lcd.print(krok);
    lcd.setCursor(8, 0); lcd.print("C:"); lcd.print(calor, 1);
    lcd.setCursor(0, 1); lcd.print("L:"); lcd.print(lm); lcd.print("m");
    lcd.setCursor(10, 1); lcd.print(mode); // 'W' або 'S'
    if (flags == 1) lcd.print(" AS");
    if (flags == 2) lcd.print(" AC");
    // Вивід у консоль
    Serial.print("Steps:"); Serial.print(krok); Serial.print(" Dist:"); Serial.print(lm); Serial.print(" Cal:");
    Serial.println(calor, 1);
    lastUpdate = millis();
}
// --- 5. Обробка тривоги (Alarm) ---
if (flags == 1 && krok >= askrok) {
    digitalWrite(RED, (millis() / 200) % 2); // Блимання червоним
    if (flag == 1) zvyk();
} else {
    digitalWrite(RED, LOW);
}
if (flags == 2 && calor >= accalor) {

```

```
    digitalWrite(BLUE, (millis() / 200) % 2); // Блимання синім
  } else {
    digitalWrite(BLUE, LOW);
  }
  // --- 6. Обробка кнопок (Спрощено) ---
  checkButtons();
}
// Винесена функція перевірки кнопок
void checkButtons() {
  if (digitalRead(KH1) == HIGH) { delay(150); mainMenu(); }
  if (digitalRead(KH2) == HIGH) { mode = 'W'; flag = 1; } // START
  if (digitalRead(KH3) == HIGH) { mode = 'S'; flag = 0; } // STOP
  if (digitalRead(KH4) == HIGH) { showInfo(); }
}
```