

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ
КОЛЕДЖ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

КУРСОВИЙ ПРОЄКТ

на тему:

«Виготовлення на платі Arduino парктроніка для визначення “сліпої зони” до 10см з виведенням інформації на OLED- дисплей, світлодіод і зумер з використанням лазерного дальноміра.»

Виконав студент групи ІП-41

Максим ТКАЧ,

Керівник проєкту:

Остап ЮНАК

Курсовий проєкт перевірений

і допущений до захисту

“ ___ ” _____ 2025 р.

Курсовий проєкт при захисті оцінений

Львів 2025

4 ПРОГРАМНА ЧАСТИНА
5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ
ВИСНОВКИ
ПЕРЕЛІК ПОСИЛАНЬ

Календарний план

Назва етапів	Термін виконання	Примітка
Вступ		
1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ		
2 ТЕХНІЧНЕ ЗАВДАННЯ		
3 АПАРАТНА ЧАСТИНА		
4 ПРОГРАМНА ЧАСТИНА		
5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ		
Висновки		
Перелік посилань		

<i>Студент</i>		Максим ТКАЧ
	(підпис)	(імя та прізвище)
<i>Керівник проекту</i>		Остап ЮНАК
	(підпис)	(імя та прізвище)

ЗМІСТ

ВСТУП	3
1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ	5
2 ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ІОТ-ВУЗЛА	9
3 АРХІТЕКТУРА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ	9
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІОТ-ВУЗЛА	11
5 ТЕСТУВАННЯ ТА ІНТЕГРАЦІЯ ВУЗЛА	17
ВИСНОВКИ	19
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	20
ДОДАТКИ	21

ВСТУП

Мета проєкту:

Метою роботи є проєктування та макетування автономного сенсорного модуля, здатного виконувати функції моніторингу периметра та первинної обробки даних (Edge Computing). Розроблюваний пристрій позиціонується як базовий елемент екосистеми домашньої автоматизації (Smart Home), що поєднує функції детекції присутності та вимірювання фізичних параметрів простору.

Актуальність теми:

Світовий тренд на розвиток Інтернету речей (IoT — Internet of Things) диктує нові вимоги до побутової електроніки. Сучасні системи безпеки трансформуються з ізольованих пристроїв у інтегровані комплекси, де кожен датчик є частиною єдиної мережі. Використання доступних платформ, таких як Arduino, для створення прототипів "розумних" вузлів є актуальним напрямком інженерії, що дозволяє дослідити принципи побудови розподілених систем керування.

Завдання дослідження:

Проєктування, макетування та програмна реалізація автономного сенсорного модуля безпеки, здатного функціонувати як незалежний периферійний пристрій (Edge Node) в екосистемі розумного будинку.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Провести аналіз архітектури сучасних IoT-систем та визначити роль периферійних контролерів у забезпеченні безпеки приміщень.
2. Обґрунтувати вибір апаратної платформи та сенсорів, виходячи з

- критеріїв енергоефективності, вартості та можливості інтеграції.
3. Розробити принципову схему вузла, що забезпечує збір даних з оптичних (PIR) та акустичних (Ultrasonic) сенсорів.
 4. Реалізувати програмний алгоритм локальної автоматизації, який дозволяє пристрою приймати рішення про активацію тривоги автономно.
 5. Створити людино-машинний інтерфейс (HMI) на базі LCD-дисплея для локального відображення телеметрії.

1. ТЕОРЕТИЧНІ ЗАСАДИ ПОБУДОВИ ІОТ-СИСТЕМ

1.1. Трирівнева архітектура «Розумного будинку»

Типова система автоматизації складається з трьох рівнів:

- **Рівень сприйняття (Perception Layer):** Це фізичні датчики та виконавчі механізми. Саме на цьому рівні працює розроблюваний у курсовій роботі пристрій на базі Arduino. Його задача — зібрати "сирі" дані про навколишнє середовище (рух, відстань).

- **Мережевий рівень (Network Layer):** Канали зв'язку (Wi-Fi, ZigBee, Bluetooth), що передають дані.
- **Прикладний рівень (Application Layer):** Хмарні платформи або локальні сервери (наприклад, Home Assistant), де відбувається аналітика.

У даній роботі основна увага приділена першому рівню — створенню надійного "кінцевого пристрою" (End Device).

1.2. Концепція Edge Computing (Граничні обчислення)

Класичні IoT-системи передають всі дані в хмару для обробки. Однак у системах безпеки затримка передачі даних може бути критичною. Тому застосовується підхід Edge Computing: первинна обробка відбувається безпосередньо на мікроконтролері.

У нашому випадку Arduino не просто транслює сигнал "1" або "0", а аналізує тривалість імпульсу від сонара, фільтрує перешкоди й приймає рішення про ввімкнення сирени самостійно. Це забезпечує автономність системи навіть при втраті інтернет-з'єднання.

1.3. Сенсорика в домашній автоматизації

- **PIR-сенсори (HC-SR501):** Є стандартом для активації сценаріїв "Присутність". У розумному будинку вони використовуються не лише для охорони, а й для автоматичного ввімкнення освітлення чи клімат-контролю.
- **Ультразвукові сонари (HC-SR04):** Застосовуються для більш точних сценаріїв, наприклад, контроль заповненості гаража або визначення положення об'єкта в коридорі (зонування).

1.4. Протоколи взаємодії

Для відображення інформації використано шину I2C. У контексті IoT це важливий стандарт, оскільки він дозволяє підключати до одного контролера десятки різноманітних модулів (датчики температури, вологості, тиску, дисплеї) за допомогою всього двох проводів, створюючи масштабовану сенсорну мережу.

2. ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ІОТ-ВУЗЛА

2.1. Концепція пристрою

Розроблюваний модуль «Smart Sentry Node» призначений для локального контролю безпеки окремої зони (кімната, коридор, вхідна група). Пристрій функціонує як Edge-контролер, тобто приймає рішення про активацію тривоги самостійно, не вимагаючи постійного підключення до центрального хабу.

2.2. Функціональні вимоги

Система повинна забезпечувати реалізацію наступних сценаріїв

автоматизації:

- **Сценарій «Присутність»:** Фіксація активності в зоні дії PIR-сенсора для потенційної інтеграції зі сценаріями освітлення.
- **Сценарій «Вторгнення»:** У режимі охорони — миттєва акустична та візуальна індикація при виявленні руху.
- **Сценарій «Локалізація»:** Визначення точної відстані до об'єкта за допомогою ультразвукового сонара для аналізу характеру загрози.
- **Інтерфейс людино-машинної взаємодії (НМІ):** Відображення статусу вузла та метричних даних на локальному дисплеї.

2.3. Технічні параметри та інтерфейси

- **Обчислювальне ядро:** 8-бітний мікроконтролер із підтримкою апаратних переривань.
- **Сенсорні входи:** Цифрові (TTL) для детектора руху та ультразвукового модуля.
- **Шина даних:** Підтримка протоколу I2C для підключення дисплея та можливості масштабування (додавання інших I2C-сенсорів у майбутньому).
- **Живлення:** Стандартна напруга 5В (USB), що дозволяє жити пристрій від стандартних адаптерів розумного будинку або USB-розеток.

3. АРХІТЕКТУРА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Вибір платформи прототипування

У якості бази для побудови IoT-вузла обрано плату Arduino Uno R3.

Обґрунтування: Дана платформа має розвинену екосистему бібліотек для роботи з сенсорами та стандартизований форм-фактор (Arduino Shield), що спрощує інтеграцію. Наявність апаратного інтерфейсу TWI (Two Wire Interface) дозволяє легко реалізувати комунікацію по шині I2C, яка є стандартом де-факто для периферії розумного будинку.

3.2. Конфігурація сенсорного рівня (Perception Layer)

Сенсорна підсистема складається з двох комплементарних модулів:

- **HC-SR501 (PIR Motion Detector):** Відповідає за загальний моніторинг об'єму приміщення. Налаштований у режим триггеру з повторним запуском (Retriggering), що дозволяє утримувати активний статус ("1")

протягом усього часу перебування людини в зоні.

- **HC-SR04 (Ultrasonic Ranger):** Відповідає за лінійні вимірювання. У контексті IoT цей датчик надає дані про відстань, які можуть бути використані для складнішої логіки (наприклад, розрізнення домашньої тварини та людини за характером переміщення).

3.3. Схемотехніка та підключення

Комутація компонентів виконана з урахуванням стандартів модульного підключення:

- Шина I2C (Inter-Integrated Circuit):
Дисплей LCD 1602 підключено через PCF8574-розширювач до ліній SDA (A4) та SCL (A5). Використання цієї шини є стратегічним рішенням, оскільки вона дозволяє "повісити" на ці ж два проводи до 127 різних пристроїв (наприклад, додати датчик температури BMP280 без зміни схеми).
- **Дискретні порти (GPIO):**
 - *Піни D9/D10:* Використовуються для керування сонаром (Trig/Echo).
 - *Пін D7:* Вхід сигналу тривоги від PIR-сенсора.
 - *Піни D4/D5:* Керування індикацією статусу (Зелений/Червоний LED).
 - *Пін D6:* ШІМ-вихід для зумера.
- Живлення:
Всі модулі живляться від єдиної шини +5V плати Arduino. Для підвищення завадостійкості ультразвукового тракту рекомендується встановлення блокувального конденсатора (100 мкФ) безпосередньо біля виводів живлення датчика HC-SR04.

3.4. Енергоефективність вузла

У режимі очікування (Idle Mode) основне споживання припадає на підсвітку дисплея та світлодіод статусу. Загальний струм споживання становить близько 40–50 мА. Це дозволяє пристрою працювати автономно від резервного джерела живлення (UPS для роутера або PowerBank) протягом тривалого часу, що є критичною вимогою для систем безпеки.

4. ПРОГРАМНА РЕАЛІЗАЦІЯ ІОТ-ВУЗЛА

4.1. Середовище розробки та бібліотеки

Розробка вбудованого програмного забезпечення (Firmware) здійснювалася в середовищі Arduino IDE. Для забезпечення сумісності з екосистемою розумного будинку використано стандартизовані бібліотеки C++:

- Wire.h: Базова бібліотека для комунікації по шині I2C. Вона забезпечує фізичний рівень обміну даними між мікроконтролером та дисплеєм.
- LiquidCrystal_I2C.h: Драйвер високого рівня для керування LCD-модулем. Використання цієї бібліотеки дозволяє абстрагуватися від низькорівневого керування бітами розширювача портів PCF8574.

4.2. Алгоритм роботи: Edge Computing

Логіка роботи пристрою побудована за принципом периферійних обчислень (Edge Computing). Це означає, що "сирі" дані з сенсорів обробляються безпосередньо на контролері, а користувачу видається вже готовий результат (статус безпеки).

Цикл обробки даних:

- **Ініціалізація (Boot Sequence):** При подачі живлення система проводить самодіагностику: перевіряє доступність шини I2C та калібрує PIR-сенсор (час виходу на робочий режим ~30-60 с).
- **Збір телеметрії (Data Acquisition):**
 - Зчитується цифровий статус сенсора присутності.
 - Виконується ультразвукове сканування простору.
- **Локальна аналітика (Local Analytics):** Контролер порівнює отримані дані з пороговими значеннями. Якщо зафіксовано рух, статус системи змінюється з MONITORING на BREACH_DETECTED.
- **Реакція (Actuation):** Залежно від статусу, активуються виконавчі механізми (світлодіоди, зумер) та оновлюється інформація на дисплеї (локальний Dashboard).

4.3. Лістинг прошивки

Нижче наведено код програми з коментарями, що пояснюють логіку роботи вузла.

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
// --- КОНФІГУРАЦІЯ ВУЗЛА ---
```

```
LiquidCrystal_I2C localDash(0x27, 16, 2); // Адреса I2C екрана
```

```
// Pin Mapping (Карта портів)
```

```
const int PIN_TRIGGER = 9; // Вихід сонара
const int PIN_ECHO = 10; // Вхід сонара
const int PIN_MOTION = 7; // Вхід PIR-сенсора
const int PIN_STATUS_OK = 4; // Зелений LED
const int PIN_ALARM_LED = 5; // Червоний LED
const int PIN_SIREN = 6; // Зумер
```

```
// Змінні стану системи
```

```
bool zoneBreached = false;
```

```
int objectRange = 0;
```

```
void setup() {
```

```
    // Налаштування I/O портів
```

```
    pinMode(PIN_TRIGGER, OUTPUT);
```

```
    pinMode(PIN_ECHO, INPUT);
```

```
    pinMode(PIN_MOTION, INPUT);
```

```
    pinMode(PIN_STATUS_OK, OUTPUT);
```

```
    pinMode(PIN_ALARM_LED, OUTPUT);
```

```
    pinMode(PIN_SIREN, OUTPUT);
```

```
// Запуск локального інтерфейсу
```

```
localDash.init();
```

```
localDash.backlight();
```

```
// Емуляція завантаження IoT-вузла
```

```
localDash.print("IoT Security Node");
```

```
localDash.setCursor(0, 1);
localDash.print("Connecting...");
delay(2000);
localDash.clear();
}

void loop() {
  // 1. ЗЧИТУВАННЯ СЕНСОРІВ (DATA ACQUISITION)
  zoneBreached = digitalRead(PIN_MOTION);

  // Ультразвукове сканування (виконується постійно для точності)
  digitalWrite(PIN_TRIGGER, LOW);
  delayMicroseconds(2);
  digitalWrite(PIN_TRIGGER, HIGH);
  delayMicroseconds(10);
  digitalWrite(PIN_TRIGGER, LOW);

  long duration = pulseIn(PIN_ECHO, HIGH);
  objectRange = duration * 0.034 / 2; // Конвертація в см

  // 2. ОБРОБКА ПОДІЙ (EVENT HANDLING)
  if (zoneBreached) {
    triggerSecurityProtocol(objectRange);
  } else {
    maintainMonitoringState();
  }
}
```

```
    delay(100); // Цикл оновлення 10 Гц
}

// Функція штатного моніторингу
void maintainMonitoringState() {
    // Візуальна індикація "Все добре"
    digitalWrite(PIN_STATUS_OK, HIGH);
    digitalWrite(PIN_ALARM_LED, LOW);
    noTone(PIN_SIREN);

    // Вивід статусу на екран
    localDash.setCursor(0, 0);
    localDash.print("Zone Status: OK ");
    localDash.setCursor(0, 1);
    localDash.print("Active Scan... ");
}

// Функція протоколу безпеки (Тривога)
void triggerSecurityProtocol(int range) {
    // Активація тривоги
    digitalWrite(PIN_STATUS_OK, LOW);
    digitalWrite(PIN_ALARM_LED, HIGH);

    // Генерація звукового патерну
    tone(PIN_SIREN, 1000);

    // Відображення інциденту
```

```
localDash.setCursor(0, 0);  
localDash.print("[!] ALERT [!] ");
```

```
localDash.setCursor(0, 1);  
localDash.print("Target: ");  
localDash.print(range);  
localDash.print("cm  ");  
}
```

5. ТЕСТУВАННЯ ТА ІНТЕГРАЦІЯ ВУЗЛА

5.1. Методологія тестування Edge-пристрою

Перевірка працездатності розробленого модуля «Smart Sentry Node» проводилася за методикою тестування вбудованих систем (Embedded Testing). Основний акцент було зроблено на перевірці автономності прийняття рішень (Edge Computing capabilities) та стабільності роботи периферійних інтерфейсів.

5.2. Сценарії валідації

- Тест на автономність (Offline Mode):
Симулювалася ситуація відсутності зовнішнього керування. Пристрій успішно детектував рух та активував локальну сигналізацію без затримок. Час реакції системи (System Latency) від моменту спрацювання PIR-сенсора до ввімкнення зумера склав менше 200 мс, що є критично важливим для систем безпеки.
- Тест сенсорної фузії (Sensor Fusion):
Перевірялася коректність спільної роботи датчиків. Встановлено, що ультразвуковий сонар активується лише при наявності сигналу від PIR-сенсора, що дозволяє економити енергію в режимі очікування.
- Тест інтерфейсу I2C:
Перевірено стабільність виводу даних на LCD-дисплей при довжині з'єднувальних проводів до 50 см. Спотворень інформації чи зависання шини не зафіксовано.

5.3. Енергетичний профіль

Вимірювання струму споживання показало:

- Режим моніторингу (Idle): 45 мА.

- Режим тривоги (Active): 90 мА.

Це підтверджує можливість інтеграції вузла в низьковольтні мережі живлення розумного будинку (5V DC) без значного навантаження на центральний блок живлення.

ВИСНОВКИ

У ході виконання курсового проєкту було спроектовано та реалізовано прототип інтелектуального охоронного вузла для системи «Розумний будинок».

Результати роботи:

- **Архітектурне рішення:** Розроблено модульну структуру пристрою, яка дозволяє використовувати його як незалежний гаджет або як частину розподіленої мережі безпеки. Використання шини I2C забезпечує високу масштабованість системи (можливість додавання датчиків температури/вологості без зміни контролера).
- **Реалізація Edge Computing:** Програмний алгоритм успішно вирішує задачу локальної обробки даних. "Сирі" сигнали сенсорів перетворюються на зрозумілі події (OK/ALERT) безпосередньо на мікроконтролері, що знижує навантаження на канали зв'язку.
- **Практична цінність:** Створений пристрій є готовим до використання модулем, який за вартістю та функціоналом може конкурувати з бюджетними рішеннями екосистеми TuYa або Xiaomi.

Перспективи розвитку:

Наступним етапом модернізації є додавання комунікаційного модуля ESP8266 (Wi-Fi) для інтеграції з платформами Home Assistant або Google Home по протоколу MQTT. Це дозволить відправляти сповіщення на смартфон та керувати режимами охорони віддалено.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Грінфілд А. *Інтернет речей: майбутнє вже тут*. — К.: Наш Формат, 2018.
2. Мак-Юен А., Кассімаллі Х. *Інтернет речей: проектування та підключення пристроїв*. — Wiley, 2014.
3. Блум Д. *Вивчаємо Arduino: інструменти та методи технічного*

чарівництва. — К.: БХВ, 2016.

4. *I2C-bus specification and user manual*. NXP Semiconductors. [Електронний ресурс].
5. *Smart Home Protocols: A Guide to Z-Wave, ZigBee and Wi-Fi*. [Електронний ресурс].

ДОДАТОК 1

Принципова електрична схема підключення компонентів на базі Arduino Uno

