


МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
Циклова комісія комп'ютерних систем та мереж
(повна назва циклової комісії)

Допустити до захисту

Голова випускової циклової комісії
комп'ютерних систем та мереж

 (повна назва циклової комісії)
Ірина КРАВЧУК
(ім'я, ПРІЗВИЩЕ)

« 10 » 06 2025 р.

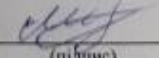
КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
ФАХОВИЙ МОЛОДШИЙ БАКАЛАВР

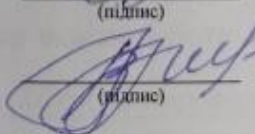
Тема: Застосування клієнт-серверної архітектури в мікроконтролерних системах

Група: 3-013 Спеціальність: 123 «Комп'ютерна інженерія»

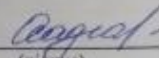
Здобувач освіти

 Максим МЕДВЕДЄВ
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник роботи

 Владислав СОБЧУК
(підпис) (ім'я, ПРІЗВИЩЕ)

Консультант з оформлення
пояснювальної записки

 Оксана ОСАДЧА
(підпис) (ім'я, ПРІЗВИЩЕ)

Кривий Ріг 2025 р.

КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Відділення комп'ютерної та програмної інженерії
Циклова комісія комп'ютерних систем та мереж
Освітній ступінь фаховий молодший бакалавр
Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Голова випускової циклової комісії
комп'ютерних систем та мереж

(прізвище, назва циклової комісії)

(підпис) Ірина КРАВЧУК
(ім'я, ПРІЗВИЩЕ)
« 01 » 03 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ОСВІТИ

Медведеву Максиму Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Застосування клієнт-серверної архітектури в мікроконтролерних системах

Керівник роботи Собчук Владислав Олегович
викладач

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по коледжу від « 04 » 04 2025 року № 50-ст

2. Строк подання здобувачем освіти роботи з 01.03.2025 по 15.06.2025

3. Вихідні дані до роботи Мікроконтролерна система Arduino Yun.

Датчики: гіроскоп-акселерометр MPU-6050, ультразвуковий датчик відстані.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Розуміння мікроконтролерних систем. Програмне забезпечення для розробки клієнт-серверних моделей. Практична реалізація: колісний робот.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Презентація Microsoft PowerPoint

6. Консультанти розділів роботи (проекту)


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	<i>Узгодження технічного завдання</i>	<i>01.03.2025</i>	
2	<i>Огляд літератури по темі кваліфікаційної роботи</i>	<i>15.03.2025</i>	
3	<i>Розуміння мікроконтролерних систем</i>	<i>28.04.2025</i>	
4	<i>Програмне забезпечення для розробки клієнт-серверних моделей</i>	<i>14.05.2025</i>	
5	<i>Практична реалізація: колісний робот</i>	<i>26.05.2025</i>	
6	<i>Оформлення пояснювальної записки</i>	<i>06.06.2025</i>	
7	<i>Захист кваліфікаційної роботи</i>		

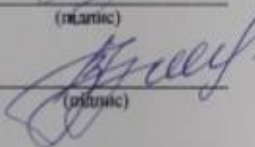
Здобувач освіти


(підпис)

Максим МЕДВЕДЄВ

(ім'я, ПРІЗВИЩЕ)

Керівник роботи


(підпис)

Владислав СОБЧУК

(ім'я, ПРІЗВИЩЕ)



Звіт подібності

Метадані

Назва організації
Ukrainian national aviation university
Заголовок
Медведєв М_3-013_2025_КП1
Автор Науковий керівник / Експерт
Медведєв МГриченко О
Ідентифікатор
Криворізький Фаховий коледж

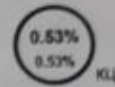
Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в рамках джерел. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрагмента для коефіцієнта подібності 2



8181

Кількість слів

62406

Кількість символів

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Застосування клієнт серверної архітектури в мікроконтролерних системах» викладена на 57 с., містить 17 рис., 12 табл. 18 використаних літературних джерел.

АРДУІНО, ВІДДАЛЕНЕ КЕРУВАННЯ, КОЛІСНИЙ РОБОТ, КЛІЄНТ СЕРВЕРНА МОДЕЛЬ, BRIDGE, ARDUINO IDE, ДРАЙВЕР ДВИГУНІВ L298N, ARDUINO YUN

Кваліфікаційна робота присвячена дослідженню мікроконтролерних систем та способів їх управління за клієнт-серверною архітектурою. Результатом роботи був створений діючий прототип клієнт-серверної моделі управління платою *Arduino YUN*. Для демонстрування її роботи був створений колісний робот, який може виконувати ряд функцій на допомогу віддаленого керування. Дану роботу рекомендовано використовувати у якості основи для подальшого вивчення технологій та розробки нових систем.

Тема кваліфікаційної роботи є актуальною і має теоретичне і практичне значення.

5

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ.....	7
ВСТУП.....	8
РОЗДІЛ 1 РОЗУМІННЯ МІКРОКОНТРОЛЕРНИХ СИСТЕМ.....	9
1.1 Що таке мікроконтролер (MCU).....	9
1.1.1 Розширювальна плата <i>NUY</i>	11
1.1.2 <i>Arduino YUN</i> : Міст між мікроконтролером та <i>Linux</i>	12
1.2 Драйвери двигунів: Невидимі керівники електроніки.....	15
1.2.1 Модуль драйвера <i>L298N</i> для керування двигунами.....	16
1.2.2 Сервоконтролер <i>Flyduino-12</i>	19
1.3 Датчики: Очі та вуха сучасних систем.....	21
1.3.1 Гіроскопічний	

акселерометр	<i>MPU6050</i>	22	1.3.2
Ультразвуковий датчик відстані	<i>HC-SR04</i>	25	
1.4 Висновок Розділу 1: Фундамент для клієнт-серверної моделі.....	26	РОЗДІЛ 2 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ КЛІЄНТ СЕРВЕРНИХ МОДЕЛЕЙ.....	27
2.1 Сучасне інтегроване середовище розробки <i>Arduino IDE</i>	27	2.2 Пропорційно-Інтегрально-Диференціальний (ПІД) регулятор: Мозок автоматичного керування.....	30
Архітектура Клієнт-Сервер: Фундамент сучасного інтернету.....	34	2.3	
2.4 Сервер: <i>Arduino YUN</i> як центр управління.....	35	2.5 Клієнт: Інтерфейс взаємодії.....	37
2.6 Варіанти підключення: Як з'єднати вашу <i>Arduino YUN</i> з мережею.....	38	2.7	
Висновок Розділу 2: Програмний фундамент та віддалене керування.....	39		
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ: КОЛІСНИЙ РОБОТ.....			
3.1 Конструкція колісного робота.....	41	3.2 Схеми підключення.....	42
3.3 Опис режимів керування роботом.....	42	3.3.1 Режим ручного керування.....	43
3.3.2 Режим переслідування.....	44	3.3.3 Режим "Кутове обертання".....	45
3.4 Безпека праці: Аналіз технології та організації виробництва.....	46	3.4.1 Планування виробничих приміщень.....	46
3.4.2 Виконувані процеси та робочі операції.....	48	3.4.3 Шкідливі та небезпечні фактори на робочих місцях.....	49
3.5 Висновок Розділу 3: Перспективи та потенціал.....	53		
ВИСНОВКИ.....	54		
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	55		

Arduino - апаратна обчислювальна платформа, основними компонентами якої є плата вводу/виводу та середовище розробки на мові *Processing/Wiring Arduino*
YUN - плата мікроконтролера на базі *ATmega32u4* і *Atheros AR9331* Процесор *Atheros* підтримує дистрибутив, заснований на *OpenWrt*. Плата має вбудований *Ethernet* і *Wi-Fi* підтримку, в *USB*-порт, слот для мікроSD карти, 20 цифрових входу / вихідних контактів

Сервер - будь-яка система, процес, комп'ютер, які володіють довільним обчислювальним ресурсом (пам'яттю, часом тощо)

Клієнт - будь-яка система, процес, комп'ютер, користувач, які запитують у сервера який-небудь ресурс, які користуються будь-яким ресурсом або обслуговуються сервером іншим способом

Клієнт-сервер - це вид розподіленої системи, в якій є сервер, що виконує запити клієнта, причому сервер і клієнт спілкуються між собою з використанням того або іншого протоколу

Скетч – в Ардуіно це програма, написана на мовах *C/C++*, що завантажується на плату

8

ВСТУП

Ми щодня користуємося комп'ютерами, але часто забуваємо про їхніх невеликих, але надзвичайно важливих родичів – мікроконтролери. Мова йде не про смартфони, а про крихітні чіпи, які є "мозком" безлічі пристроїв навколо нас. Де зустрічаються мікроконтролери?

Мікроконтролерні системи вже зараз є повсюдно: від пральних машин та автомобілів до холодильників та іншої побутової техніки. Вони також відіграють ключову роль у роботі квадрокоптерів, систем "розумного будинку" та, звісно, роботів.

Хоча багато цих пристроїв мають певні можливості для зв'язку, більшість із них все ще вимагають прямого втручання людини (наприклад, натискання кнопок) для управління.

Революція в Дистанційному Управлінні

У цій статті ми представляємо інноваційну модель дистанційного управління для мікроконтролерних систем, засновану на клієнт-серверній архітектурі. Ця модель відкриває принципово нові можливості для взаємодії з пристроями.

Для демонстрації цієї моделі, ми використали *Arduino YUN* як основну платформу. На її основі був розроблений колісний робот з дистанційним керуванням. Цей робот підтримує декілька режимів роботи, наочно ілюструючи переваги та функціональність нашої системи управління.

Безмежні Можливості для Майбутнього

Ця універсальна модель відкриває можливості для створення широкого спектра нових пристроїв, здатних суттєво полегшити роботу інженерів, медичних працівників, водіїв, будівельників і людей з інвалідністю. Окрім того, вона може слугувати основою для розробки ще більш розвинених і інтелектуальних систем «розумного будинку».

РОЗДІЛ 1

РОЗУМІННЯ МІКРОКОНТРОЛЕРНИХ СИСТЕМ

1.1 Що таке мікроконтролер (*MCU*)

Мікроконтролер (*MCU*) — це, по суті, "комп'ютер на чіпі". На відміну від звичайних комп'ютерних процесорів, які вимагають безлічі додаткових компонентів, мікроконтролер об'єднує всі ключові елементи — процесор, пам'ять (оперативну та постійну) та периферійні пристрої — на одній мікросхемі. Це дозволяє створювати надзвичайно компактні, енергоефективні та доступні електронні пристрої.

Саме мікроконтролери є основою для вбудованих систем, які ми зустрічаємо повсюди: від смартфонів до пральних машин, роблячи їх "розумними" та функціональними.

Різноманіття Мікроконтролерів

На сучасному ринку представлено величезне різноманіття мікроконтролерів, які можна умовно поділити на такі основні типи:

- 8-бітні вбудовані мікроконтролери: Зазвичай використовуються для простих, специфічних завдань.

- 16- та 32-бітні інтегральні схеми (IC): Забезпечують більшу обчислювальну потужність та функціональність.

- Цифрові сигнальні процесори (*DSP*): Спеціалізовані для обробки сигналів у реальному часі.

Вбудовані Мікроконтролери: Все-в-Одному Рішення

Більшість вбудованих мікроконтролерів розроблені за принципом "система на-чипі" (*SoC*). Це означає, що всі необхідні ресурси — пам'ять, порти введення/виведення та центральний процесор — розташовані на одному кристалі. Просто подайте живлення та тактовий сигнал, і мікроконтролер готовий до роботи. Типовий мікроконтролер включає такі базові компоненти:

- Схема скидання: Забезпечує коректний старт процесора.

10

- Генератор тактових імпульсів: Задає ритм роботи всіх компонентів. -

Центральний процесор (ЦП): Виконує команди програми.

- Пам'ять програм (*E(E)PROM*) та інтерфейс: Зберігає код програми. - Порти введення/виведення: Для взаємодії з зовнішніми пристроями. - Таймер: Відстежує цикли виконання команд, дозволяючи реалізовувати тимчасові затримки.

Більш складні вбудовані мікроконтролери також можуть мати розширені можливості, такі як:

- Вбудований монітор/відладчик програм: Для зручного тестування та налагодження коду.

- Внутрішні засоби програмування пам'яті: Для оновлення прошивки без додаткового обладнання.

- Обробка переривань: Швидке реагування на зовнішні події.

- Аналогові входи/виходи: Для роботи з аналоговими сигналами (наприклад, від датчиків).

- Послідовний та паралельний ввід/вивід: Для різних типів зв'язку з іншими пристроями.

- Можливість підключення зовнішньої пам'яті: Для розширення обсягу зберігання даних.

Всі ці функції значно збільшують гнучкість використання мікроконтролерів та спрощують процес розробки систем на їх основі.

Мікроконтролери із Зовнішньою Пам'яттю

Деякі мікроконтролери, особливо потужні 16- та 32-бітні моделі,

використовують зовнішню пам'ять. Це рішення застосовується у випадках, коли пристрою потрібен значний обсяг пам'яті та відносно невелика кількість портів введення/виведення.

Чудовим прикладом є контролери жорстких дисків (*HDD*). Вони використовують зовнішній буферний кеш для швидкого проміжного зберігання та обробки великих обсягів даних (до кількох мегабайт). Така архітектура дозволяє

11

мікроконтролеру працювати на значно вищих швидкостях, ніж моделі з виключно внутрішньою пам'яттю.

Цифрові Сигнальні Процесори (*DSP*)

Цифрові сигнальні процесори (*DSP*) – це окремий клас процесорів, спеціально розроблених для обробки поточних даних у реальному часі. Їхнє основне завдання — отримувати аналогові дані із зовнішніх систем, швидко їх обробляти та генерувати відповідну реакцію.

DSP зазвичай є невід'ємною частиною більших систем, виступаючи як керуючий елемент для зовнішнього обладнання, а не як самостійний пристрій. Їх можна знайти в аудіо- та відеообладнанні, телекомунікаційних системах та інших пристроях, де потрібна швидка та точна обробка сигналів.

1.1.1 Розширювальна плата *NUY*

NUY — це плата розширення, розроблена *8Devices*, яка значно розширює можливості вашого *Arduino*, додаючи підтримку *Wi-Fi* та повноцінної операційної системи *Linux*. Ця плата базується на модулі *Carambola2* з мікросхемою *Atheros AR9331 (SoC)*, що є провідним промисловим рішенням.

NUY працює під управлінням *OpenWRT-YUN* — спеціалізованого дистрибутива *Linux*, розробленого командою *Arduino* на базі популярної *OpenWRT*. Будучи апаратним продуктом з відкритим кодом, *NUY* (див рис. 1.1) повністю сумісний з *Arduino YUN* та його скетчами (програмами), що робить його чудовим доповненням для розробників.



Рисунок 1.1 - Зовнішній вигляд *NUY*

На жаль, попри свої переваги, *NUY* поки що не набув широкої популярності в Україні, тому знайти його на місцевому ринку може бути досить складно.

Таблиця 1.1 - Характеристики *NUY*

Параметр	Деталі
Бездротовий модуль	<i>Carambola2 (AR9331)</i>
Інтерфейси	- <i>Wi-Fi 802.11</i> - <i>Ethernet 100 Мбіт/с</i> - <i>USB-хост</i> - Послідовний <i>TTL</i> - Стандартний роз'єм для зовнішньої антени (<i>U.FL / IPEX</i>)
Частота <i>Wi-Fi</i>	2.4 ГГц
Максимальна вихідна потужність	21 дБм
Бездротові стандарти	<i>bgn</i>
Живлення	5 В, від плати <i>Arduino</i>
Розмір	54 x 69 мм
Програмне забезпечення	<i>OpenWrt-YUN</i>

1.1.2 *Arduino YUN*: Міст між мікроконтролером та *Linux*

Arduino YUN — це унікальна мікроконтролерна плата, що поєднує в собі дві потужні складові: мікроконтролер *ATmega32u4* (як у *Arduino Leonardo*) та

процесор *Atheros AR9331*. Саме процесор *Atheros* надає платі можливість запускати повноцінний дистрибутив *Linux – OpenWrt-YUN*, розроблений на базі популярної *OpenWrt*.

Особливості *Arduino YUN*

Плата *Arduino YUN* (рис. 1.2) має багатий набір вбудованих функцій: -
Мережеві можливості: Вбудовані адаптери *Ethernet* та *Wi-Fi* для легкого підключення до мережі.

13

- Розширення: Порт *USB-A* та слот для карт *micro SD* для підключення зовнішніх пристроїв та розширення пам'яті.

- Введення/виведення: 20 цифрових контактів вводу/виводу для взаємодії з різними сенсорами та виконавчими механізмами.

- Основні компоненти: Кварцовий генератор 16 МГц та роз'єм *micro USB* для живлення та програмування.



Рисунок 1.2 - Зовнішній вигляд *YUN*

Переваги Інтеграції *Linux*

Що відрізняє *Arduino YUN* (рис. 1.3) від інших плат *Arduino*, так це її здатність взаємодіяти з вбудованою системою *Linux*. Це перетворює плату на потужний мережевий комп'ютер, зберігаючи при цьому звичну простоту використання *Arduino*. Ви можете не лише виконувати стандартні команди *Linux*, такі як *cURL*, а й писати власні скрипти оболонки та *Python* для створення складних та надійних інтеграцій.

Як і *Leonardo*, *Arduino YUN* використовує *ATmega32u4* з вбудованим *USB* з'єднанням, що усуває потребу в додатковому процесорі для *USB*. Це дозволяє *YUN* функціонувати як миша чи клавіатура для підключеного комп'ютера, а також

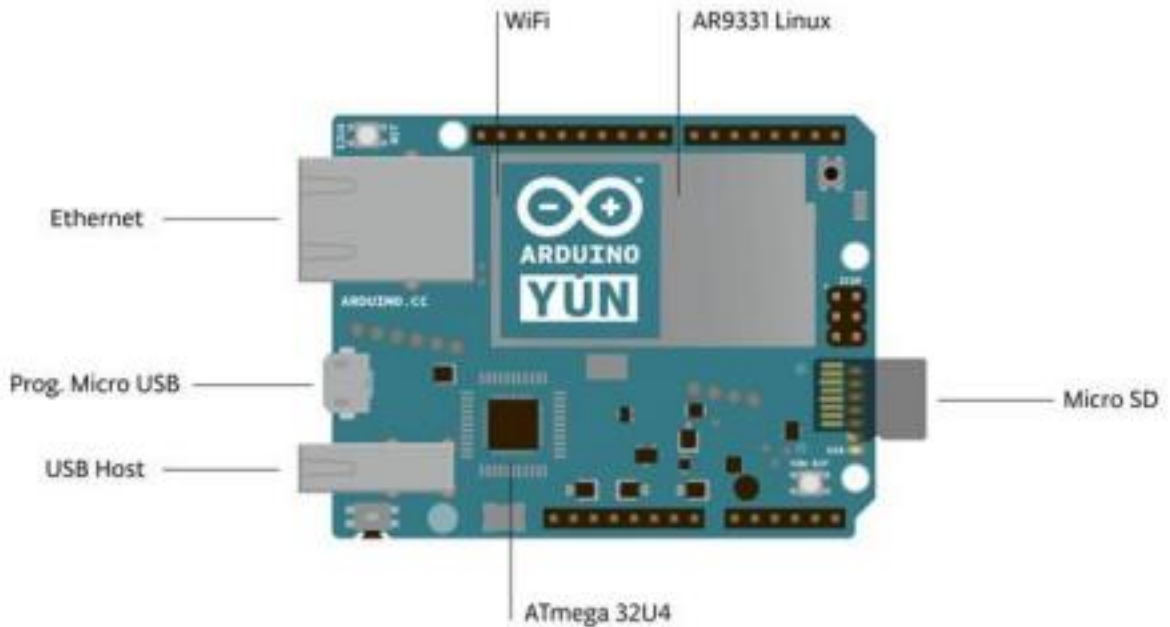


Рисунок 1.3 Розташування модулів плати *Arduino YUN* [1]

Бібліотека *Bridge*: Зв'язок між Світами

Ключовим елементом, що забезпечує взаємодію між мікроконтролером *ATmega32u4* та *Linux*-процесором *AR9331*, є бібліотека *Bridge*. Вона дозволяє скетчам *Arduino* запускати скрипти в *Linux*, взаємодіяти з мережевими інтерфейсами та отримувати інформацію від процесора *AR9331*, відкриваючи широкі можливості для складних проектів.

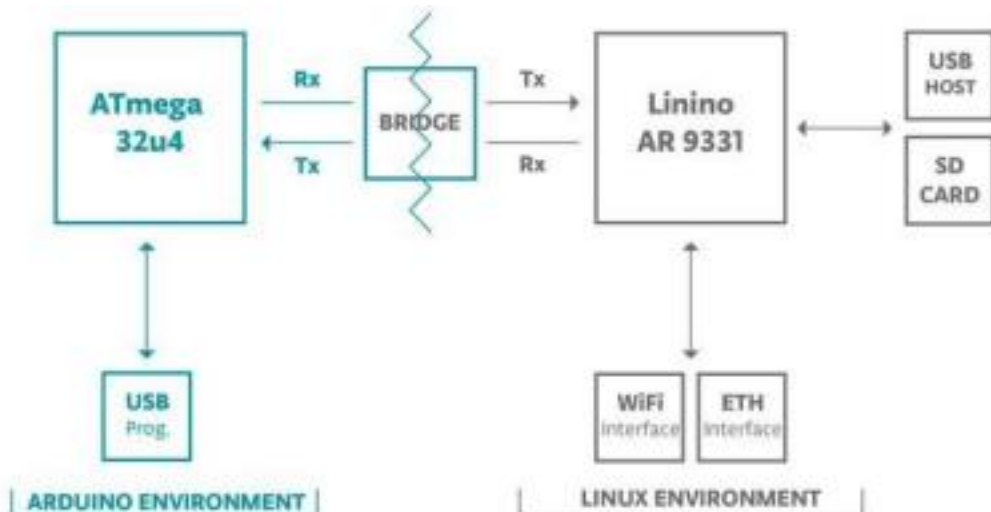


Рисунок 1.4 - Схема зв'язку плати *Arduino YUN* [2]

AR9331 (*Linux*), а не безпосередньо до *ATmega32U4*. Однак, саме бібліотека *Bridge* дозволяє *Arduino* взаємодіяти з цими периферійними пристроями *Linux*-частини плати.

Таблиця 1.2 - Актуальні технічні характеристики плати *Arduino Yún*

Параметр	Значення
Центральний мікроконтролер	<i>ATmega32u4</i>
Номінальна напруга живлення	5 В
Діапазон вхідної напруги живлення	5 В (через <i>USB</i> або <i>VIN</i>)
Кількість цифрових портів вводу/виводу	14
Кількість каналів ШІМ	7
Кількість аналогових входів	12
Максимальний струм на пін вводу/виводу	40 мА
Максимальний струм на пін 3.3 В	50 мА
Обсяг флеш-пам'яті	32 КБ (4 КБ зарезервовано під завантажувач)
Обсяг статичної оперативної пам'яті (<i>SRAM</i>)	2.5 КБ
Обсяг електрично стираної пам'яті (<i>EEPROM</i>)	1 КБ
Частота роботи мікроконтролера	16 МГц

1.2 Драйвери двигунів: Невидимі керівники електроніки

Драйвер — це електронний пристрій або мікросхема, що перетворює електричні сигнали на форму, придатну для прямого управління виконавчими механізмами або сигнальними елементами. По суті, це інтерфейс між керуючою електронікою та керованим пристроєм, що дозволяє точно контролювати його роботу. Драйвери можуть бути як окремими модулями, так і інтегрованими компонентами в складних системах.

Існує широкий спектр драйверів, кожен з яких призначений для виконання специфічних функцій у складі апаратного забезпечення:

- Формувачі шин (*Bus Transceivers*): Використовуються для забезпечення надійної передачі цифрових сигналів між окремими модулями комп'ютерних або електронних систем. Вони здійснюють узгодження рівнів напруги та покращують стабільність сигналу при обміні даними через загальні шини зв'язку.

- Інтерфейсні драйвери: Слугують для прийому, передачі й перетворення цифрових сигналів між пристроями, а також для електричного узгодження з характеристиками фізичних каналів зв'язку. Найбільш поширеними прикладами є драйвери для інтерфейсів *RS-232*, *RS-485*, *RS-422*, *CAN*, *LIN*, *Ethernet*, *USB*, *IEEE 1394* тощо.

- Драйвери виконавчих механізмів: Забезпечують керування різними виконавчими пристроями, зокрема електродвигунами (в тому числі кроковими), соленоїдами, сервоприводами, сигнальними лампами, звуковими пристроями, розподільниками та іншими елементами, що здійснюють механічні або сигнальні дії.

- Модулі живлення та керування: Призначені для забезпечення стабільної та ефективної роботи пристроїв, що потребують точного контролю параметрів живлення при запуску, роботі та вимиканні. Прикладом є драйвери для світлодіодів, які підтримують оптимальні умови живлення *LED*-компонентів.

- Драйвери силових транзисторів (*MOSFET*, *IGBT*): Виконують критично важливу функцію в системах силової електроніки, забезпечуючи швидке перемикання транзисторів за допомогою подачі достатнього струму на затвор. Це дозволяє знизити втрати енергії під час перемикання та підвищити загальну ефективність роботи системи.

1.2.1 Модуль драйвера *L298N* для керування двигунами

Модуль на базі мікросхеми *L298N* є популярним і універсальним засобом для керування електродвигунами. Він дозволяє одночасно керувати двома двигунами постійного струму або одним двофазним кроковим двигуном з чотирма виводами.

Завдяки своїй простоті та надійності, плата широко використовується в робототехніці, мобільних платформах, механічних приводах та інших проектах з

рухомими елементами.



Рисунок 1.5 – Зовнішній вигляд модуля драйвера *L298N* [2]

Функціональність перемикача живлення

На платі розміщено перемикач *S1*, який дозволяє вибрати джерело живлення логічної частини:

- *S1* замкнутий — живлення логіки здійснюється через вбудований стабілізатор модуля;
- *S1* розімкнутий — необхідне зовнішнє джерело 5 В для живлення логічної частини.

Живлення для двигунів може подаватися в діапазоні 7–20 В, при цьому доступний вбудований стабілізатор для логіки. Якщо використовується напруга менше 7 В, рекомендується подавати 5 В безпосередньо з контролера. Для напруг понад 20 В необхідно застосовувати окремий стабілізатор або зовнішнє джерело живлення логіки.



Рисунок 1.6 – Розташування ключових компонентів на платі *L298N* [3]

Основні технічні характеристики модуля *L298N*:

- Драйвер: *L298N* (подвійний *H*-мостовий драйвер)
- Напруга живлення двигунів (V_s): від +5 В до +35 В
- Максимальний вихідний струм: до 2 А на канал (пікове навантаження) -
- Живлення логіки (V_{ss}): від +5 В до +7 В
- Споживання логічної частини: до 36 мА
- Рівні логічних сигналів:
 - Низький рівень: від -0.3 В до 1.5 В
 - Високий рівень: від 2.3 В до V_{ss}
- Максимальна розсіювана потужність: до 20 Вт при температурі 75°C -
- Робочий температурний діапазон зберігання: від -25°C до +130°C -
- Габарити модуля (з радіатором): 48 × 43 × 33 мм
- Маса: 33 г

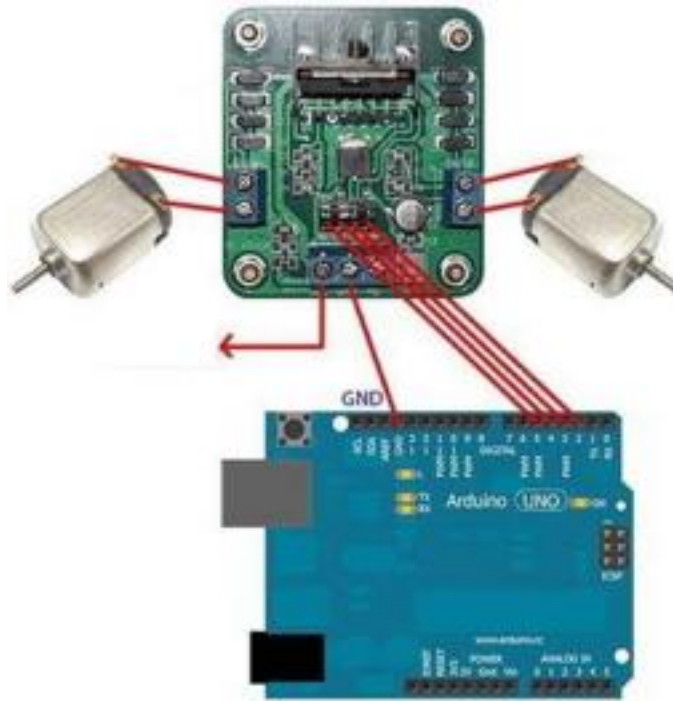


Рисунок 1.7 – Схематичне підключення модуля *L298N* [4]

Додаткові особливості:

Модуль обладнано світлодіодними індикаторами, які відображають наявність живлення та логічних сигналів управління, що значно полегшує налагодження та візуальний контроль під час роботи з платою.

1.2.2 Сервоконтролер *Flyduino-12*

Flyduino-12 — це, ймовірно, найменший *Arduino*-сумісний контролер, спеціально розроблений для управління сервоприводами. Він здатний керувати до 12 стандартних сервоприводів, що робить його ідеальним для роботизованих проектів, квадрокоптерів або дронів завдяки вбудованому інтерфейсу *XBee*.

Його надзвичайно легка вага (лише 7,5 г) робить *Flyduino-12* чудовим вибором для проектів, де кожен грам має значення. Вбудований регулятор дозволяє підключати плату до джерел живлення з напругою від 3,5 В до 8 В, що охоплює більшість стандартних акумуляторів та блоків живлення.



Рисунок 1.8 - Зовнішній вигляд контролера *Flyduino* [4]

Flyduino-12 сумісний з *Arduino IDE* та програмується як *Arduino Mini*. Для підключення до комп'ютера через *USB* вам знадобиться адаптер з вихідною напругою 3,3 В. Важливо: ніколи не використовуйте 5-вольтовий адаптер, оскільки це може пошкодити контролер! Для зручності управління сервоприводами рекомендується використовувати бібліотеку *MegaServo*.

Ключові Характеристики:

- Мікроконтролер: *Atmega328P*.
- Робоча напруга: 3,3 В.
- Вхідна напруга: 3,5 В - 8 В.
- Кількість каналів керування сервоприводами: 12.
- Роз'єм *XBee*: Для бездротового зв'язку.
- Аналогові входи: 8.
- *SRAM* (статична ОЗП): 2 КБ.
- *EEPROM* (електрично стирає ПЗП): 1 КБ.
- Вага: 7,5 г.

21

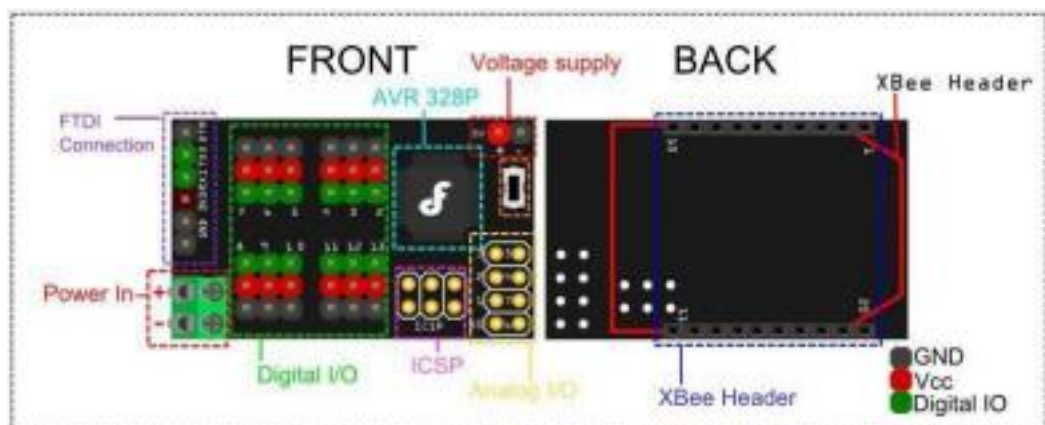


Рисунок 1.9 - Схема розташування компонентів драйвера *Flyduino* [5]

1.3 Датчики: Очі та вуха сучасних систем

Датчик (або сенсор) — це ключовий компонент, який дозволяє технічним системам сприймати навколишній світ. Його основне завдання — вимірювати та перетворювати фізичні величини (такі як тиск, температура, витрата, концентрація, швидкість, переміщення, напруга, струм тощо) на зручні для обробки сигнали (електричні, оптичні, пневматичні). Це дає змогу вимірювати, передавати, перетворювати, зберігати та реєструвати інформацію про стан об'єкта.

Історично датчики розвивалися пліч-о-пліч з вимірювальною технікою (термометрами, витратомірами, манометрами). Проте, з появою та розвитком систем автоматичного керування, термін "сенсор" став невід'ємною частиною ширшої концепції "сенсор – система керування – виконавчий механізм – керований об'єкт". Застосування датчиків у системах автоматичного запису параметрів (наприклад, у наукових дослідженнях) є одним із багатьох сценаріїв їх використання.

Еволюція Датчиків: Від "Очей" до "Розумних Очей"

Останнім часом у сфері сенсорів з'явилися такі терміни, як "багатофункціональний датчик" та "розумний датчик". Це відображає сучасні тенденції розвитку, де датчики перестають бути лише простими перетворювачами. Окрім своєї базової функції вимірювання, ці нові покоління сенсорів набувають розширених можливостей завдяки інтеграції з аналого-цифровими

22

перетворювачами та вбудованими мікроконтролерами. Це значно розширює їхній функціонал, дозволяючи:

- Попередня обробка сигналів: Виконання лінеаризації, фільтрації та корекції помилок безпосередньо в датчику, забезпечуючи більш точні та чисті дані. -

Самодіагностика: Здатність датчика перевіряти свій власний стан та виявляти несправності, підвищуючи надійність системи.

- Дистанційне налаштування: Можливість віддаленого коригування діапазону вимірювання, одиниць вимірювання та узгодження частотної характеристики, що спрощує експлуатацію.

- Індивідуальний контроль: Більш точне та адаптивне керування процесами або об'єктами на основі отриманих даних.

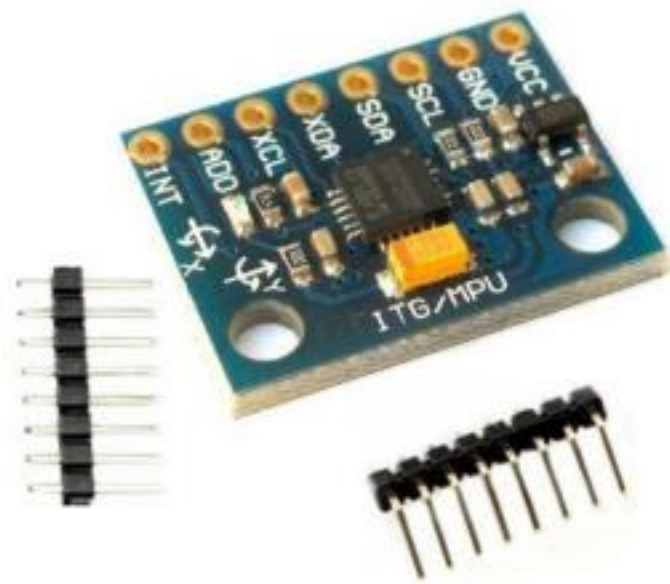
1.3.1 Гіроскопічний акселерометр MPU6050

Перш ніж заглибитись у деталі, давайте розберемося, що таке гіроскоп та акселерометр.

Гіроскоп — це пристрій, який реагує на зміни кута орієнтації об'єкта. У традиційному розумінні, це швидко обертовий інерційний об'єкт на підвісці, який зберігає свою орієнтацію, а кут відхилення вимірюється положенням підвіски. Однак, електронні гіроскопи працюють на інших, складніших принципах, часто використовуючи мікроелектромеханічні системи (*MEMS*).

Акселерометр — це пристрій, що вимірює "видиме" прискорення, тобто різницю між фактичним прискоренням об'єкта та прискоренням, спричиненим гравітацією. Простий приклад — маса, закріплена на пружинній підвісці. Коли система прискорюється (кидається або повертається), пружина реагує на рух маси, і це відхилення дозволяє визначити прискорення.

Отже, гіроскоп відстежує зміни в обертанні об'єкта в просторі, незалежно від його лінійного руху. Натомість, акселерометр вимірює лінійне прискорення і може використовуватися для штучного обчислення положення об'єкта в просторі. Кожен з цих датчиків має свої унікальні переваги та обмеження.



MPU6050 — це інтегрована мікросхема, що об'єднує в собі тривісний гіроскоп, тривісний акселерометр та датчик температури. Ця мікросхема є серцем модуля *GY-521*. Окрім самого чіпа, плата модуля також містить необхідні для *MPU6050* компоненти, включаючи:

- Підтягуючі резистори для інтерфейсу *I2C*, що забезпечують коректний зв'язок з мікроконтролером.

- Стабілізатор напруги 3,3 В з низьким падінням (*LDO*), який забезпечує стабільні 3 В на виході при живленні 3,3 В, разом з фільтруючим конденсатором [6].

- Нерозпаяний *SMD*-світлодіод з обмежувальним резистором для індикації наявності живлення.

Розмір плати модуля *GY-521* становить компактні 10 x 20 мм.

24

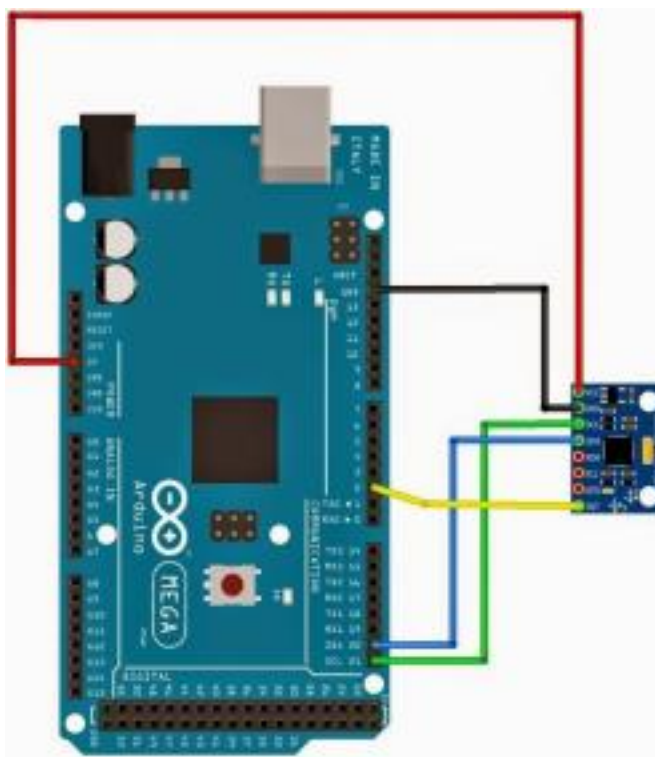


Рисунок 1.11 - Приклад підключення модуля *MPU6050* [6]

Ключові Характеристики *MPU6050*:

- Тривісний *MEMS*-гіроскоп: З 16-бітним аналогово-цифровим перетворювачем (АЦП) для точного вимірювання кутових швидкостей. -

- Тривісний *MEMS*-акселерометр: З 16-бітним АЦП для вимірювання лінійних прискорень.

- Цифровий процесор руху (*DMP*): Вбудований апаратний блок для складних

обчислень руху, що знімає навантаження з основного мікроконтролера. - *Slave I2C*: Для підключення до мікроконтролерів (наприклад, *Arduino*). - *Master I2C*: Для підключення додаткових зовнішніх датчиків безпосередньо до *MPU6050*.

- Регістри даних датчиків: Для зберігання вимірних значень. - *FIFO (First-In, First-Out)* буфер: Для ефективного збору даних. - Лінія переривання: Для сповіщення мікроконтролера про події. - Датчик температури: Для компенсації температурних коливань та додаткових вимірювань.

25

- Функція самотестування: Для гіроскопа та акселерометра, що дозволяє перевірити їхню працездатність.

- Регістр ідентифікації пристрою: Для унікальної ідентифікації *MPU6050*.

1.3.2 Ультразвуковий датчик відстані *HC-SR04*

Ультразвуковий далекомір *HC-SR04* — це потужний сенсор, що надає роботам "зір", дозволяючи їм вимірювати відстань до об'єктів. Він ідеально підходить для навігації, допомоги роботам в об'їзді перешкод або для картографування приміщень. Також його можна використовувати як датчик безпеки, що спрацьовує, коли об'єкт наближається до певного порогу.



Рисунок 1.12 - Зовнішній вигляд датчика *HC-SR04* [6]

Принцип Роботи: Як Бачать Дельфіни та Кажани

Принцип роботи *HC-SR04* імітує ехолокацію, якою користуються дельфіни та кажани. Датчик випромінює короткий ультразвуковий імпульс на частоті 40 кГц, а потім прослуховує зворотне відлуння. Шляхом вимірювання часу, необхідного для подорожі звукової хвилі туди й назад, *HC-SR04* точно розраховує відстань до об'єкта [7].

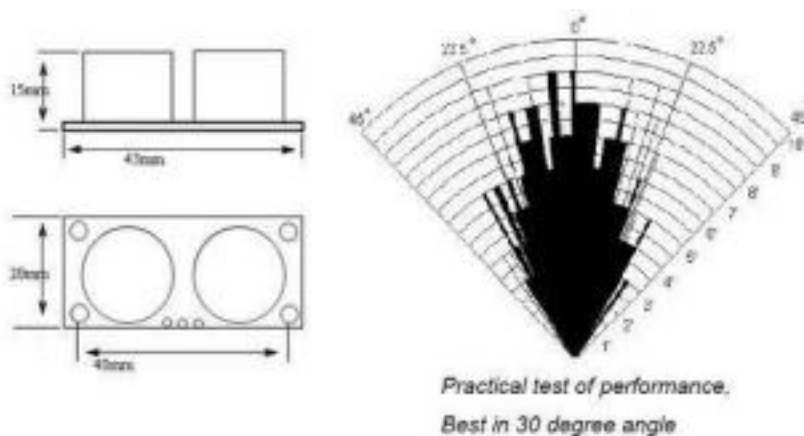


Рисунок 1.13 - Діаграма фактичної продуктивності використання *HC-SR04* [6]

На відміну від інфрачервоних далекомірів, ультразвукові датчики не схильні до впливу сонячного світла чи кольору об'єкта. Однак варто зазначити, що вони можуть мати труднощі з визначенням відстані до пухнастих або дуже тонких об'єктів, оскільки такі поверхні погано відбивають звукові хвилі.

1.4 Висновок Розділу 1: Фундамент для клієнт-серверної моделі

Цей розділ заклав основу для подальшого розуміння моделі управління, зосередившись на ключових аспектах мікроконтролерних систем. Ми детально розглянули різні платформи мікроконтролерів, які можуть слугувати базою для розробки клієнт-серверної моделі управління. Окрім цього, ми дослідили периферійні пристрої, що є невід'ємними компонентами для практичної реалізації таких систем.

РОЗДІЛ 2

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ КЛІЄНТ СЕРВЕРНИХ МОДЕЛЕЙ

2.1 Сучасне інтегроване середовище розробки *Arduino IDE*

Arduino IDE — це безкоштовне, кросплатформне середовище розробки з відкритим вихідним кодом, призначене для створення, компіляції та завантаження програмного коду (скетчів) на мікроконтролерні плати *Arduino*. Побудоване на мові *Java*, воно використовує компоненти проєктів *Processing* і *Wiring*, а також

підтримує *Windows*, *macOS* та *Linux*.



Рисунок 2.1 - Зовнішній вигляд інтегрованого середовища розробки *Arduino IDE* [5]

28

Основні компоненти інтерфейсу *Arduino IDE*:

- Редактор коду — базовий текстовий редактор із підсвічуванням синтаксису та функціями пошуку, вирізання/вставки.

- Область повідомлень — виводить інформацію про компіляцію, помилки та інші повідомлення системи.

- Консоль — детальні логи компіляції та повідомлення під час роботи з платою.

- Панель інструментів — швидкий доступ до основних функцій. - Меню — доступ до налаштувань, інструментів, довідки та інших функціональних можливостей.

Основні кнопки панелі інструментів:

- *Verify* (Перевірити) — компілює код без завантаження на плату. -

Upload (Завантажити) — компілює та передає скетч на плату. -

New (Новий) — створює новий проєкт.

- *Open* (Відкрити) — відкриває існуючий скетч.

- *Save* (Зберегти) — зберігає поточний скетч.

- *Serial Monitor* (Послідовний монітор) — відкриває інтерфейс для передачі та прийому даних через *COM*-порт.

Меню *IDE*: розширені функції

- *File* (Файл) — керування скетчами, відкриття, збереження, доступ до *Sketchbook*.

- *Edit* (Редагувати) — редагування коду, копіювання з форматуванням (*HTML*, для форумів).

- *Sketch* (Скетч) — компіляція, додавання бібліотек, перегляд розташування скетчу, додавання файлів.

- *Tools* (Інструменти):

- *Auto Format* — автоматичне форматування коду.

- *Board* — вибір моделі плати.

- *Port* — вибір послідовного порту.

- *Burn Bootloader* — запис завантажувача на мікроконтролер.

29

Робота зі скетчами

Arduino IDE реалізує концепцію *Sketchbook* — окремої директорії для зберігання скетчів. При першому запуску ця папка створюється автоматично, її можна змінити в налаштуваннях. Скетчі можуть складатися з кількох файлів (з розширеннями *.ino*, *.c*, *.cpp*, *.h*), які відкриваються у вигляді вкладок.

Перед завантаженням необхідно:

- Вибрати модель плати (*Tools > Board*).

- Вказати активний *COM*-порт (*Tools > Port*):

- *Windows*: *COM4*, *COM5* тощо.

- *macOS*: */dev/tty.usbserial-**.

- *Linux*: */dev/ttyUSB0* або подібне.

Після цього скетч завантажується кнопкою *Upload* або через меню *File > Upload*. У сучасних плат *Arduino* завантаження супроводжується автоматичним

перезавантаженням і блимаючими індикаторами *RX/TX*. У разі успіху *IDE* виводить відповідне повідомлення, або вказує на помилки.

Завантажувач *Arduino*

Arduino Bootloader — це попередньо записана в мікроконтролер програма, яка дозволяє завантажувати нові скетчі без зовнішнього програматора. Він автоматично активується після перезапуску плати та забезпечує короткий період готовності до прийому коду.

Послідовний монітор (*Serial Monitor*)

Інструмент *Serial Monitor* дозволяє здійснювати текстовий обмін між комп'ютером і платою *Arduino*. Для передачі даних слід ввести текст у відповідне поле та натиснути *Send*. Також необхідно встановити правильну швидкість передачі (*baud rate*), що відповідає виклику *Serial.begin()* у скетчі. На *Linux/macOS* відкриття монітора може спричиняти перезапуск плати.

30

2.2 Пропорційно-Інтегрально-Диференціальний (ПІД) Регулятор: Мозок автоматичного керування

Пропорційно-інтегрально-диференціальний (ПІД) регулятор є одним із найпоширеніших та найпотужніших алгоритмів для автоматичного керування в промисловості та робототехніці. Він об'єднує в собі три ключові складові, що дозволяють точно та стабільно підтримувати бажане значення керованої величини.

Реалізація ПІД-регулятора ґрунтується на пружному зворотному зв'язку, де система постійно аналізує різницю між бажаним та фактичним станом. На рисунку 1b (хоча його не надано в тексті, уявляємо його як графік) можна побачити перехідну функцію ПІД-закону, що ілюструє вплив кожної з його складових: Диференціальної (*D*), Пропорційної (*P*) та Інтегральної (*I*).

Застосування ПІД-регулятора особливо ефективно для об'єктів з великими затримками та інерцією, оскільки він дозволяє значно покращити якість регулювання.

Математична Основа ПІД-регулятора

Закон ПІД-регулятора описується наступним рівнянням:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2.1)$$

Де:

- $u(t)$ — вихідний керуючий вплив у момент часу t .

- $e(t)$ — похибка (різниця між бажаним і фактичним значенням) у момент часу t .

- K_p — коефіцієнт пропорційної складової.

- K_i — коефіцієнт інтегральної складової.

- K_d — коефіцієнт диференціальної складової.

Це рівняння може бути реалізовано як регулятор постійної швидкості або ПІД регулятор з налаштуванням, де параметри K_p , T_i (час інтегрування) та T_d (час диференціювання) визначають тип та якість перехідних процесів у системі керування та динамічні характеристики об'єкта.

31

Приклад: Контроль Температури з ПІД-регулятором

Розглянемо практичний приклад використання ПІД-регулятора для контролю температури. Це чудовий спосіб зрозуміти його роботу.

Задача: Нам потрібно підтримувати температуру певного об'єкта на заданому рівні. Для цього ми маємо:

1. Об'єкт, температуру якого треба регулювати (наприклад, нагрівальний елемент).
2. Мікроконтролер, який буде керувати процесом.
3. Датчик температури (термометр), що передаватиме поточні дані контролеру.
4. Виконавчий пристрій (наприклад, реле або ШІМ-модулятор) для керування потужністю нагрівача.
5. Комп'ютер, підключений до мікроконтролера, для задання бажаної температури.

Вхідні дані: Поточна температура об'єкта та бажана (цільова) температура.
Вихідні дані: Значення потужності, яку необхідно подати на нагрівальний елемент. Для таких завдань, де потрібне точне та стабільне регулювання, ПІД-регулятор є ідеальним рішенням.

Компоненти ПІД-регулятора та Їхня Роль

Давайте покроково розберемо, як працює кожен компонент ПІД-регулятора для вирішення типових проблем.

Пропорційна (P) складова

Це найпростіша частина. Ми обчислюємо різницю (похибку) між бажаною температурою (задана точка) та поточною температурою. Ця похибка множиться на коефіцієнт пропорційності (Kp), і результат передається на нагрівач як керуюча потужність.

Проблема: Використання лише пропорційної складової має два суттєві недоліки:

1. Затримка: Реакція на зміни не є миттєвою.

32

2. Статична похибка (відхилення): Пропорційна складова не враховує зовнішні впливи (наприклад, тепловтрати в навколишнє середовище). Коли температура досягає бажаного значення, похибка стає нульовою, і вихідна потужність також обнуляється. Проте, через тепловіддачу, об'єкт починає охолоджуватися, і температура коливатиметься навколо бажаного значення, ніколи не досягаючи його ідеально.

Диференціальна (D) складова

Ця складова допомагає вирішити проблему затримки та передбачає майбутні зміни. Вона реагує на швидкість зміни похибки. Якщо температура швидко наближається до цільового значення, диференціальна складова зменшує керуючий вплив, запобігаючи "перельоту" (*overshoot*) через цільове значення.

Приклад: Якщо поточна температура нижча за бажану, пропорційна складова починає подавати потужність. Диференціальна складова (яка обчислюється як похідна від похибки, тобто різниця між поточною та попередньою похибкою, помножена на коефіцієнт Kd) також вносить свій внесок. Коли температура підвищується і наближається до цільового значення, швидкість зміни похибки стає від'ємною (адже похибка зменшується). Тому диференціальна складова починає поступово зменшувати потужність, "гальмуючи" нагрів до моменту досягнення цільового значення.

Інтегральна (I) складова

Ця складова вирішує проблему статичної похибки (постійного відхилення). Вона накопичує похибку з часом.

Приклад: Повернімося до температурного контролю. Якщо температура нижча за задане значення, пропорційна складова починає нагрів. Поки похибка є додатною, інтегральна складова накопичує її значення (у програмі це просто сума

всіх попередніх похибок, помножена на коефіцієнт K_i). Коли температура "досягає" бажаного значення, пропорційна та диференціальна складові можуть стати нульовими. Проте інтегральна складова зберігає накопичене значення похибки, що не дорівнює нулю. Це означає, що вона продовжує подавати невелику потужність на нагрівач, компенсуючи тепловтрати в навколишнє середовище і дозволяючи

33

системі точно підтримувати бажану температуру без постійних коливань. Це є дуже ефективним рішенням для усунення остаточного відхилення. Формула ПД-регулятора

На основі цього розуміння, отримуємо повну формулу ПД-регулятора:

(2.2) Тут $u(t)$ — керуючий вплив, а $e(t)$ — похибка. Цю формулу часто можна зустріти у різному записі, але її суть залишається незмінною.

Налаштування Коефіцієнтів ПД-регулятора

Налаштування коефіцієнтів K_p , K_i , K_d є критично важливим для ефективної роботи ПД-регулятора. В ідеалі, це повинно базуватися на математичному моделюванні системи, що вимагає глибоких знань теорії автоматичного керування. Однак на практиці часто використовуються спрощені методи.

Метод Циглера-Ніколса

Цей метод є одним з найвідоміших для емпіричного налаштування ПД регулятора.

Кроки:

1. Обнулiть усі коефіцієнти: Встановіть $K_p=0$, $K_i=0$, $K_d=0$.
2. Збільшуйте K_p : Поступово збільшуйте коефіцієнт пропорційності (K_p), спостерігаючи за реакцією системи.
3. Визначте критичні значення: Продовжуйте збільшувати K_p до моменту, коли керована величина почне безперервно коливатися (стабільні, незатухаючі коливання).
4. Запишіть параметри: Зафіксуйте значення K_p у цей момент (це буде критичний коефіцієнт $K_{\text{крит}}$) та виміряйте період цих коливань ($T_{\text{крит}}$). На основі отриманих $K_{\text{крит}}$ та $T_{\text{крит}}$ коефіцієнти ПД-регулятора обчислюються за наступними формулами:

2.3 Архітектура клієнт-сервер: Фундамент сучасного інтернету

Клієнт-серверна технологія — це перевірений часом і широко поширений підхід до організації взаємодії в розподілених системах, особливо актуальний для роботи з базами даних в інтернеті. Її історія сягає десятиліть, і вона традиційно використовувалася у великих корпоративних середовищах. Сьогодні, завдяки бурхливому розвитку інтернету та накопиченню величезних обсягів інформації, що зберігається в базах даних, клієнт-серверна архітектура залишається в центрі уваги розробників програмного забезпечення.

Процес взаємодії між клієнтом і сервером можна описати таким алгоритмом:

- Клієнт ініціює запит: Користувач, працюючи з клієнтським додатком (наприклад, веб-браузером), формує запит (заповнює форму, натискає кнопку) і надсилає його до сервера, точніше, до серверної програми, яка обробляє цей запит. - Сервер обробляє запит: Серверна програма взаємодіє з базою даних, що зберігається на сервері, виконуючи дії відповідно до отриманого запиту. Вона генерує необхідний результат (наприклад, вибірку даних або підтвердження операції) та передає його назад клієнту.

- Клієнт відображає результат: Клієнтський додаток отримує відповідь від сервера, відображає інформацію користувачеві (на моніторі або в інтерфейсі) і чекає на подальші дії. Цей цикл "запит-відповідь" продовжується доти, доки користувач не завершить роботу з сервером.

Переваги Клієнт-Серверної Архітектури

Стандартні програмні рішення, що реалізують клієнт-серверну технологію, пропонують низку важливих переваг:

- Масштабованість: Можливість ефективного розширення апаратного забезпечення для збільшення продуктивності системи.

- Стабільність: Висока надійність та стійкість до збоїв у роботі. - Безпека: Надійний захист даних від несанкціонованого доступу. - Обробка великих обсягів даних: Здатність ефективно працювати з масштабними базами даних та складними проектами.

Принцип роботи клієнт-серверної системи не залежить від фізичного розташування клієнта та сервера, або від способу підключення. Користувач може бути віддаленим, підключеним навіть через модем.

Коли клієнт (наприклад, через веб-браузер) заповнює форму або ініціює дію, його запит надходить на сервер. Серверна програма отримує ці дані, проводить їхню перевірку і звертається до системи управління базами даних (СУБД) або отримує результат. СУБД обробляє запит, і за відсутності помилок, повертає необхідні дані серверній програмі. Диски сервера, де зберігається база даних, модифікуються відповідно до запиту клієнта.

Такий розподілений режим роботи значно підвищує безпеку бази даних, забезпечує захист від апаратних та програмних збоїв, запобігає несанкціонованому доступу, а також сприяє збільшенню загальної продуктивності системи та зниженню навантаження на мережу. Проте, варто враховувати, що це також збільшує вимоги до ресурсів та потужності самого сервера.

2.4 Сервер: *Arduino YUN* як центр управління

Плата *Arduino YUN* — це справжній мікроконтролерний центр, що об'єднує два потужні процесори:

- *ATmega32u4* (архітектура *AVR*): Відповідає за низькорівневі операції, керування периферійними пристроями, обробку сигналів від датчиків та контроль діодів.

- *Linino AR9331* (*Linux*-процесор): Працює під управлінням легкої операційної системи *Linux*, що забезпечує розширені мережеві можливості, зокрема доступ до *Wi-Fi* адаптера.

Wi-Fi адаптер на *Arduino YUN* може працювати в двох режимах: - Точка доступу (*Access Point*): *Arduino YUN* виступає як повноцінний маршрутизатор, створюючи власну *Wi-Fi* мережу, до якої ви можете підключатися з інших пристроїв і надсилати команди.

- Підключення (*Client/Station*): Плата підключається до вже існуючої *Wi-Fi* мережі (наприклад, вашого домашнього роутера) і взаємодіє з іншими пристроями в локальній мережі.

Незалежно від режиму, *Arduino YUN* може успішно функціонувати як сервер, і це широко використовується дослідниками та розробниками по всьому світу.

Програмування Сервера на *Arduino YUN*

Код (скетч) для мікроконтролерної системи записується в пам'ять процесора *ATmega32u4* за допомогою *Arduino IDE*. Для забезпечення зв'язку між *ATmega32u4* та потужнішим *Linux*-процесором *Linino AR9331* використовується спеціальна бібліотека *Bridge*. Вона надає різні механізми для взаємодії з *Linux*-частиною плати.

Налаштування зв'язку починається з ініціалізації системи *Bridge*:

```
C++
```

```
Bridge.begin();
```

Бібліотека також пропонує серверну функціональність через клас *YunServer*. Його необхідно налаштувати на прослуховування:

- *YunServer::listenOnLocalhost()*: Для взаємодії з клієнтами в межах власної мережі *YUN*.

- *YunServer::noListenOnLocalhost()*: Для взаємодії через існуючу точку доступу (наприклад, ваш роутер).

Після конфігурації, сервер запускається методом *YunServer::begin()*:

```
C++
```

```
myServer.begin();
```

У головній функції програми *void loop()* необхідно постійно перевіряти спроби підключення клієнтів. Це робиться за допомогою класу *YunClient*, екземпляр якого отримується від сервера через метод *YunServer::accept()*. Якщо немає активних підключень, метод поверне *NULL*; інакше ви отримаєте об'єкт *YunClient*, що представляє нового клієнта.

```
C++
```

```
YunClient client = myServer.accept();
```

```
if (client) {
```

```
// Обробка запиту від клієнта
```

```
}
```

Взаємодія з Клієнтами: *REST*-Запити та Обмін Даними

Основний механізм передачі даних до *Arduino YUN* через мережу — це *REST* запити, які можна надсилати через адресний рядок будь-якого веб-браузера. Для

зчитування даних, отриманих від клієнта, використовується базовий метод `YunClient::read()`, який зчитує один байт інформації. Для зручності роботи з рядками та числовими даними існують також допоміжні методи, такі як `YunClient::readStringUntil(char delimiter)`, `YunClient::parseInt()`, `YunClient::parseFloat()` тощо. Ці методи дозволяють ефективно парсити вхідні дані та видобувати з них параметри.

Для відправки даних у відповідь клієнту використовуються методи `YunClient::print()` та `YunClient::println()`. Вони дозволяють надсилати текстові повідомлення, числа або будь-яку іншу інформацію.

Після обробки запиту та відправлення відповіді, критично важливо завершити з'єднання за допомогою методу `YunClient::stop()`, щоб звільнити ресурси сервера: C++

```
client.stop();
```

2.5 Клієнт: Інтерфейс взаємодії

Клієнтська програма — це застосунок, який встановлює зв'язок із сервером через визначений мережевий протокол. Основною її функцією є надсилання запитів до сервера з подальшою обробкою отриманих відповідей. Залежно від завдань, клієнт може отримувати, змінювати або запускати певні дії на стороні сервера. Результати обробки можуть бути виведені користувачеві або використані програмою для подальших операцій. Клієнт і сервер можуть працювати як на одному пристрої, так і на різних, але для їх взаємодії необхідне наявне мережеве з'єднання (локальне або через Інтернет).

38

Як згадувалося, для комунікації із сервером (*Arduino YUN*) клієнт використовує *REST*-запити. Це означає, що будь-який веб-браузер теоретично може виступати в ролі клієнта. Проте, для проектів, що вимагають частої та інтерактивної взаємодії, зручніше використовувати спеціалізоване клієнтське програмне забезпечення.

У даному випадку, клієнтська програма була розроблена на мові програмування *Java* 1.6 з використанням середовища розробки *NetBeans* 8.0.2. Для створення графічного інтерфейсу користувача (*GUI*) використовувалася стандартна графічна бібліотека *Java* — *Swing*.

Оскільки система передбачає керування за допомогою джойстика (детальніше про це буде пояснено далі), для підтримки взаємодії з ігровими контролерами, клавіатурами та іншими пристроями введення, була використана бібліотека *Jinput*. Це забезпечує гнучкість та сумісність клієнтської програми з різними пристроями керування, підключеними до комп'ютера.

2.6 Варіанти підключення: Як з'єднати вашу *Arduino YUN* з мережею

Як ми вже згадували, *Arduino YUN* оснащена вбудованими можливостями для мережевого зв'язку, що значно спрощує її інтеграцію у віддалені системи керування. Ось основні способи підключення:

- Вбудований *Wi-Fi*: Це найзручніший та найпопулярніший варіант. *Arduino YUN* має інтегрований *Wi-Fi* адаптер, який працює "з коробки" без необхідності встановлення додаткових драйверів чи бібліотек. За умови правильного налаштування, плата може легко підключатися до будь-якої доступної точки доступу в межах зони покриття, забезпечуючи бездротову передачу даних.

- Вбудований *Ethernet*: Другий спосіб – використання вбудованого *Ethernet* адаптера. Цей метод забезпечує більш стабільне та швидке дротове з'єднання, але він менш гнучкий, ніж *Wi-Fi*, оскільки обмежує мобільність платформи. Ви не зможете вільно переміщувати пристрій, якщо він підключений кабелем.

39

- *GSM*-модуль (додатково): Хоча цей модуль не входить до базової комплектації *Arduino YUN*, його можна придбати окремо та налаштувати за допомогою відповідних бібліотек. Використання *GSM*-модуля надає найвищий рівень автономності, дозволяючи пристрою працювати в місцях, де немає *Wi-Fi* або *Ethernet*-інфраструктури, використовуючи мобільну мережу. Однак варто враховувати, що швидкість зв'язку та його надійність через *GSM* можуть бути нижчими порівняно з *Wi-Fi* чи *Ethernet*.

2.7 Висновок Розділу 2: Програмний фундамент та віддалене керування

У цьому розділі ми детально розглянули програмні засоби, необхідні для створення ефективних клієнт-серверних систем на базі *Arduino*. Було обговорено

середовище розробки *Arduino IDE*, яке є основним інструментом для написання та завантаження коду на плату.

Особлива увага приділялася алгоритму роботи пропорційно-інтегрально диференціального (ПІД) регулятора, що є ключовим для точного та стабільного керування різними процесами, і його реалізація була детально пояснена.

Описана вище клієнт-серверна модель відкриває широкі можливості для віддаленого керування мікроконтролерними системами. Основною вимогою як для клієнта, так і для сервера є стабільне мережеве з'єднання, яке може бути забезпечене одним із розглянутих способів. Ця архітектура дозволяє створювати гнучкі та потужні рішення для автоматизації та робототехніки.

40

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ: КОЛІСНИЙ РОБОТ

Після детального вивчення апаратного та програмного забезпечення, ми перейшли до створення прототипу колісного робота, що використовує описані технології. Перш ніж зануритись у його опис, давайте з'ясуємо, що ж таке робот. Сучасні роботи бувають різних форм і призначень:

- Промислові роботи: Зазвичай це маніпулятори, схожі на "руки", закріплені на основі, що виконують рутинні завдання, такі як збирання, переміщення предметів або зварювання.

- Мобільні роботи: Пристрої, що працюють у небезпечних для людини середовищах або віддалено керовані. Це можуть бути роботи для глибоководних досліджень, космічних операцій, військові роботи (для розвідки, розмінування, транспортування вантажів) та інші.

- Домашні роботи: Хоча поки не дуже поширені, футурологи прогнозують їхнє широке використання для допомоги в повсякденному житті.
- Роботи-іграшки: Пристрої, призначені для розваг та навчання. Робот може бути електромеханічним, пневматичним, гідравлічним або комбінацією цих типів, керованим складною системою. Його основна мета — замінити людину у виробництві, будівництві, роботі з важкими або небезпечними матеріалами, а також у повсякденному житті.

Роботи можуть виконувати команди:

- Безпосередньо від оператора.
- За заздалегідь складеними програмами.
- Використовувати технології штучного інтелекту для автономного виконання завдань на основі загальних інструкцій.

Ці можливості дозволяють значно спростити або повністю замінити людську працю в умовах, які є складними, небезпечними або монотонними для людини.

41

3.1 Конструкція колісного робота

Основою розробленого робота є мобільна двоповерхова платформа. Вона складається з двох рівнів (палуб), що забезпечують зручне кріплення друкованих плат, датчиків та іншого периферійного обладнання завдяки спеціальним отворах.

Рух платформи забезпечують два сервоприводи, встановлені в нижній частині та з'єднані з колесами. Для забезпечення стійкості та маневреності, як третя точка опори використовується паразитне колесо, яке не потребує приводу і може вільно обертатися в будь-якому напрямку.

Рисунок 3.1 - Зовнішній вигляд двошарової платформи робота [7]

Платформа робота інтегрує наступні ключові компоненти:

- Плата *Arduino YUN*: Виступає як центральний мозок робота, забезпечуючи управління та зв'язок.

- Два драйвери двигунів *L298N*: Контролюють обертання сервоприводів. -

Чотири первинні батареї по 1.5 В: Джерело живлення для двигунів.

42

- Ультразвуковий датчик відстані *HC-SR04*: Використовується для визначення відстані до об'єктів та уникнення перешкод.

- Гіроскопічний акселерометр *MPU6050*: Надає дані про орієнтацію та рух робота у просторі.

- Блок живлення для *Arduino YUN*: Забезпечує стабільне живлення плати. -
Вимикач живлення двигунів: Для безпечного ввімкнення/вимкнення живлення рушіїв.

3.2 Схема підключення

На рисунку 3.2 (схематичне зображення, що не надається в тексті) відображено поточну робочу схему підключення всіх периферійних пристроїв до плати *Arduino YUN*. Ця схема є ключовою для розуміння взаємодії апаратних компонентів робота.

Рисунок 3.2 - Схема підключення компонентів робота [8]

3.3 Опис режимів керування роботом

Керування роботом здійснюється через *REST API*, що дозволяє надсилати команди віддалено. Кожен режим роботи робота має власну унікальну назву

(ключове слово) та може приймати специфічні параметри для активації та налаштування.

Усі команди починаються з префікса *arduino*, що вказує на звернення до процесора AVR. Після цього йде ключове слово *drive*, яке активує модель поведінки колісного робота. Наприклад, команда може виглядати як *http://<IP_ARDUINO>/arduino/drive/<режим_роботи>?<параметри>*.

Цей підхід забезпечує гнучке та структуроване керування роботом через мережевий інтерфейс.

3.3.1 Режим ручного керування

3.3.1.1 Опис

Цей режим активується ключовим словом "*manual*" у REST-запиті. Приклад запиту:

http://<IP_адреса_Arduino>/arduino/drive/manual/p:<P_параметр>,y:<Y_параметр>,t:<T_параметр>,r:<R_параметр>/

Параметри, що передаються:

- *p (Pitch)*: Параметри для керування обертанням коліс.
- *y (Yaw)*: Швидкість руху вперед (значення > 0) або назад (значення < 0) у діапазоні від -1000 до 1000.
- *t (Throttle)*: Швидкість повороту ліворуч (значення > 0) або праворуч (значення < 0) у діапазоні від -1000 до 1000.
- *r (Roll)*: Швидкість обертання навколо своєї осі за годинниковою стрілкою (значення > 0) або проти годинникової стрілки (значення < 0), у діапазоні від -1000 до 1000.

Оператор використовує спеціально розроблену клієнтську програму, що інтегрується з джойстиком. Джойстик дозволяє керувати рухом робота вперед/назад, нахилами вправо/вліво, а також обертанням навколо власної осі для імітації повороту.

3.3.1.2 Практичне застосування

Цей механізм дистанційного керування може бути використаний у широкому спектрі застосувань, де потрібен точний та віддалений контроль над пристроями: -

Керування будівельними кранами: Забезпечує підвищену безпеку та точність операцій.

- Дистанційне керування транспортними засобами: Дозволяє здійснювати точне маневрування, наприклад, у небезпечних зонах.

- Керування електричними інвалідними візками: Надає користувачам більшу мобільність та незалежність.

3.3.2 Режим переслідування

3.3.2.1 Опис

Цей режим активується ключовим словом "*pursuit*". Приклад запиту:

http://<IP_адреса_Arduino>/arduino/drive/pursuit/d:<D_відстань>/

Єдиний параметр:

- *d (Distance)*: Бажана відстань до перешкоди в сантиметрах.

У цьому режимі робот автоматично підтримує задану відстань до об'єктів. Завдяки ПІД-контролеру та показанням ультразвукового датчика відстані *HC SR04*, швидкість робота динамічно регулюється:

- Якщо робот наближається до перешкоди, його швидкість зменшується. -

Якщо відстань до перешкоди перевищує задане *D*, робот збільшує швидкість.

- Якщо відстань дорівнює *D*, робот підтримує поточну швидкість, рухаючись паралельно або зупиняючись.

3.3.2.2 Практичне застосування

Цей режим створює основу для розробки інтелектуальних систем: - Системи допомоги водієві (*ADAS*): Може бути використаний для адаптивного круїз-контролю або систем запобігання зіткненням в автомобілях.

45

- Автономні транспортні засоби: Є ключовим елементом для навігації та уникнення перешкод у безпілотних автомобілях.

- Системи автоматичного паркування: Дозволяє роботу точно маневрувати на обмеженому просторі.

3.3.3 Режим "Кутове обертання"

3.3.3.1 Опис

Цей режим активується ключовим словом "*angle*". Приклад запиту: `http://<IP_адреса_Arduino>/arduino/drive/angle/a:<A_кут>/`

Єдиний параметр:

- *a (Angle)*: Кут обертання в градусах.

У цьому режимі робот виконує обертання навколо своєї осі на заданий кут — за або проти годинникової стрілки, залежно від знака кута. Для контролю повороту використовується гіроскопічний сенсор *MPU6050*, який фіксує зміни орієнтації в реальному часі. Плавність і точність обертання забезпечуються завдяки *PID* регулятору, який динамічно регулює швидкість руху залежно від залишкового кута. Це дозволяє уникнути різких змін і гарантує контрольоване завершення маневру.

3.3.3.2 Практичне застосування

Ця функція є важливою для роботів, що вимагають точного позиціонування: - Операційні роботи: Забезпечує плавне та скоординоване виконання завдань, що вимагають точних поворотів, наприклад, у промислових маніпуляторах або сервісних роботах.

- Навігація мобільних платформ: Дозволяє роботам точно орієнтуватися у просторі та виконувати запрограмовані маршрути.

46

3.4 Безпека праці: Аналіз технології та організації виробництва

3.4.1 Планування виробничих приміщень

Ефективне планування виробничої зони має вирішальне значення для забезпечення безпечних та комфортних умов праці. Це включає в себе оптимізацію об'єму та площі приміщення, визначення кількості працівників, оптимальне розміщення обладнання та робочих місць, організацію природного та штучного освітлення, ефективну вентиляцію, а також характеристики підлоги та стін. Ці аспекти детально відображаються на кресленнях планування.

Розглянемо офісне приміщення дослідницького центру, розташоване на третьому поверсі 16-поверхової офісної будівлі. Офіс розрахований на 7 робочих

місце. Основні параметри приміщення:

- Ширина: 6 метрів
- Довжина: 9 метрів
- Висота: 3 метри

Виходячи з цих даних, розрахуємо загальні показники:

- Площа кімнати: $6 \text{ м} \times 9 \text{ м} = 54 \text{ м}^2$
- Об'єм кімнати: $54 \text{ м}^2 \times 3 \text{ м} = 162 \text{ м}^3$
- Площа, зайнята обладнанням: 7.75 м^2
- Об'єм, зайнятий обладнанням: 5.996 м^3

План робочої зони цього офісного приміщення показано на Рисунку 3.3.

47

Рисунок 3.3 - Робоче місце користувача ПК [11]

Елементи робочого місця:

1. Комп'ютерний стіл: 75 см (висота) \times 65 см (глибина) \times 140 см (ширина)
 2. Комп'ютерне крісло: 37.5 см (ширина сидіння) \times 45 см (глибина сидіння) \times 100 см (висота спинки)
 3. Принтер: 35.4 см (ширина) \times 29.7 см (глибина) \times 24.6 см (висота)
4. Проектор: 34.5 см (ширина) \times 27 см (глибина) \times 11.3 см (висота) Далі проведено розрахунок площі та об'єму приміщення, що припадає на одного працівника:

- Об'єм на одного працівника:

($V_{\text{приміщення}} - V_{\text{обладнання}}$)/кількість

працівників= $(162 - 5.996)/7 \approx 22.3 \text{ м}^3$ - Площа на одного працівника:

($S_{\text{приміщення}} - S_{\text{обладнання}}$)/кількість працівників= $(54 - 7.75)/7 \approx 6.6 \text{ м}^2$ У

таблиці 3.1 наведено порівняння фактичних характеристик робочого місця з нормативними вимогами щодо площі, об'єму та ергономічних параметрів.

48

Таблиця 3.1 – Параметри робочого місця користувача ПК

Назва параметра	Нормативне	Фактичне
	значення	значення
Площа на одного працівника	Не менше 6 м ²	6.6 м ²
Висота приміщення	Щонайменше 3 м	3 м
Об'єм на одного працівника	Не менше 20 м ³	22.3 м ³
Висота робочої поверхні столу	700 мм	700 мм
Рекомендовані розміри робочої поверхні столу	1900 × 900 мм	1900 × 900 мм
Відстань від клавіатури до краю столу	Не більше 300 мм	300 мм
Відстань від очей користувача до монітора	400–800 мм	600 мм
Висота сидіння офісного крісла (регульована)	400–500 мм	450 мм
Глибина сидіння	380 мм	380 мм
Ширина сидіння	400 мм	400 мм
Висота опорної поверхні спинки	Не менше 300 мм	380 мм
Ширина опорної поверхні спинки	Не менше 380 мм	380 мм

Як видно з таблиці, усі ключові параметри робочого простору відповідають або перевищують встановлені нормативи.

3.4.2 Виконувані процеси та робочі операції

Розробка колісного робота здійснюється з використанням системи дистанційного керування та повністю координується відповідно до заданого технічного процесу. До виконання робіт залучено команду з 7 спеціалістів, кожен з яких виконує визначену роль відповідно до своєї професійної компетенції. Розподіл обов'язків між учасниками команди подано нижче:

- 1 інженер-конструктор
- 1 інженер-технолог
- 2 інженери-схемотехніки
- 2 оператори (ймовірно, для керування та тестування робота) -
- 1 тестувальник

49

Для виконання своїх завдань співробітники використовують комп'ютерне обладнання, перелік якого наведено в Таблиці 3.2.

Таблиця 3.2 - Перелік обладнання для розробки проєкту

Назва обладнання Кількість Сучасна модель / Технічні характеристики		
Персональний комп'ютер (системний блок)	7	<i>Intel Core i5</i> 12-го покоління / 16 ГБ RAM / SSD 512 ГБ / Відеокарта <i>Intel UHD</i>
Монітор	7	24-дюймовий <i>LED IPS</i> -дисплей <i>Samsung S24R350</i>
Принтер	1	Лазерний принтер <i>HP LaserJet M211dw</i> (ч/б, <i>Wi-Fi</i> , автоматичний двосторонній друк)

3.4.3 Шкідливі та небезпечні фактори на робочих місцях

Під час роботи, особливо в офісних умовах, можуть виникати різні шкідливі та потенційно небезпечні фактори, які негативно впливають на здоров'я та

самопочуття працівників. У таблиці 3.3 наведено основні джерела цих ризиків у відповідному відділі.

Таблиця 3.3 - Основні джерела небезпеки та їх вплив на співробітників відділу

Основна Небезпека (Шкідливість) Фактори, що Впливають на Співробітників Відділу	
Електронна безпека	Вплив електромагнітного випромінювання, ризик ураження електричним струмом
Низька (висока) температура	, порушення терморегуляції організму
Висока відносна вологість	Дискомфорт, погіршення самопочуття
Нейроемоційне перевантаження	Стрес, втома, зниження концентрації
Перевантаження візуального аналізатора	Напруження очей, погіршення зору
Пожежна безпека	Ризик виникнення пожежі від електрообладнання

3.4.3.1 Несприятливі мікрокліматичні умови

Особливу увагу слід приділити мікрокліматичним умовам, оскільки вони безпосередньо впливають на комфорт та продуктивність праці. Таблиця 3.4

50

деталізує шкідливі фактори, пов'язані з мікрокліматом, їхній вплив на організм та пропоновані заходи для мінімізації негативних наслідків.

Таблиця 3.4 - Шкідливі фактори – Несприятливі мікрокліматичні умови

Шкідливий Фактор Вплив на Організм Людини		Заходи щодо Усунення або Зменшення Впливу на Працівників
Висока температура та вологість	Перегрів, прискорене серцебиття, зниження артеріального тиску	Розумне розташування обладнання, встановлення ефективної вентиляційної системи, забезпечення доступу до чистої питної води
Низька температура	Переохолодження, підвищений ризик застудних захворювань	Встановлення належної системи опалення, забезпечення доступу до теплої питної води (за потреби)

3.4.3.2 Недостатнє та надмірне освітлення

Освітлення робочого місця має величезне значення для комфорту, зорового здоров'я та продуктивності співробітників. Важливо знайти баланс між недостатнім та надмірним освітленням.

Таблиця 3.5 - Особливості візуальної роботи

Сфера Застосування		Значення
Роздільна здатність екрана		1920x1080
Розмір 1 пікселя		0.3 мм
Розмір об'єктів на дисплеї (середнє значення)		
Мінімальний розмір об'єкта		3 (0.9 мм)
Контраст		Чудово
Фон		Світлий
Візуальна точність		Середня точність

Таблиця 3.6 - небезпечні фактори освітлення

Шкідливі фактори освітлення		Вплив на Організм Людини	Заходи щодо усунення або зменшення впливу на працівників
Недостатнє освітлення на робочому місці	Передчасна зорова втома, зниження ефективності роботи, підвищена ймовірність помилок		Збільшення кількості світла, рівномірний розподіл освітлення по всьому робочому простору.

Продовження таблиці 3.6

Надмірна яскравість світла (блискучість) Зниження	світлочутливості ока, зниження працездатності, підвищена ймовірність помилки Перепланування системи освітлення, встановлення	жалюзі на вікнах для контролю природного світла.
--	---	--

3.4.3.3 Шум і вібрація

Надмірний рівень шуму на робочому місці може негативно впливати на слух, концентрацію та загальну продуктивність працівників.

Таблиця 3.7 – Шкідливі фактори: Шум

Шкідливий фактор		
Вплив на організм людини		Заходи для зменшення впливу
Підвищений рівень шуму на робочому місці	Погіршення слуху, зниження концентрації та продуктивності, підвищення ймовірності помилок	Використання тихих та енергоефективних комплектуючих; застосування звукоізоляційних матеріалів та акустичних панелей

3.4.3.4 Випромінювання від комп'ютерного обладнання

Під час роботи з комп'ютерною технікою працівники можуть зазнавати впливу різних видів електромагнітного випромінювання, що потенційно можуть негативно позначатися на здоров'ї.

Таблиця 3.8 – Небезпеки при використанні електронно-обчислювального обладнання

Небезпечні та шкідливі фактори		
Вплив на організм людини		Заходи для зниження впливу
Статичні електричні поля, магнітні поля, іонізуюче, ультрафіолетове, м'яке рентгенівське та змінне електромагнітне випромінювання	Зорова втома, головний біль, дратівливість, порушення сну, зниження працездатності, загальна втома	Дотримання оптимальної відстані між очима і екраном (60-80 см); використання захисних екранів; організація регулярних перерв і дотримання режиму роботи та

		відпочинку
--	--	------------

3.4.3.5 Небезпека ураження електричним струмом

Для захисту електричного обладнання від коротких замикань використовуються запобіжники. На всіх щитах керування та розподільних коробках обов'язково розміщуються таблички з попередженням про напругу. Таблиця 3.9 деталізує аспекти контакту людини з електричним струмом та відповідні заходи безпеки [12].

Таблиця 3.9 - Заходи захисту від електричного струму

Тип Електричного Впливу	Проектні Заходи Зменшення Небезпеки		Організаційні та Технічні Заходи Зниження Ризику
Ураження електрострумом (через електрообладнання)	Захисний кожух		Проведення навчання з безпечних методів роботи з електрообладнанням.
Ураження електрострумом (через внутрішню проводку)	Прихована проводка (наприклад, APV, 5.25 мм ²)		Проведення інструктажів з електробезпеки.
Ураження електрострумом (через заземлення)	Ізоляція (наприклад, лінолеум)		Проведення інструктажів з електробезпеки.

Загалом, рекомендується дотримуватися регулярного технічного обслуговування електроустановок та залучати кваліфікованих електриків для проведення всіх комутаційних, ремонтних, монтажних та налаштувальних робіт.

3.4.3.6 Пожежна небезпека

Будівля, де розташовані приміщення, побудована із залізобетонних плит і відповідає II класу вогнестійкості [13]. Межа вогнестійкості такої конструкції становить від 0.5 до 2.5 годин. Приміщення містить переважно негорючі тверді та волокнисті матеріали. За класифікацією вибухо- та пожежонебезпеки, приміщення належить до класу Б, враховуючи наявність дерев'яних меблів та елементів (полиці, підлога, касові апарати).

53

Таблиця 3.10 наводить потенційні джерела загоряння та відповідні заходи безпеки.

Таблиця 3.10 - Джерела пожежної небезпеки та заходи зменшення ризику

Джерело Загоряння	Проектні Заходи	Організаційні та Технічні
	Зменшення Небезпеки	Заходи Зниження Ризику
Перевантаження електромережі / Коротке замикання / Підвищений перехідний опір	2 переносні вуглекислотні вогнегасники CO2 (ВВ-3 / ВВК-2) з об'ємом балона 3 літри (2 кг) — призначені для гасіння пожеж різних речовин, що не горять без доступу повітря. Кожен вогнегасник покриває площу 107 м ² . Наявність датчиків пожежної безпеки.	Дотримання технічних норм електропроводки; використання термометра <i>LaCrosse WS8005</i> для контролю температурних параметрів; використання кондиціонера <i>LG G24LHT</i> для регулювання клімату та вентиляції. Розвантаження електрообладнання та огляд проводки після робочого дня; ознайомлення персоналу з інструкціями з експлуатації електроприладів; наявність узгодженого плану евакуації.

3.5 Висновок Розділу 3: Перспективи та потенціал

Розроблена система демонструє значний потенціал для створення десятків, а то й сотень різноманітних пристроїв. Ці рішення можуть суттєво допомогти як окремим особам, так і компаніям в автоматизації рутинних, складних або нецікавих завдань.

Створений прототип колісного робота є не лише успішною реалізацією концепції, але й цінною основою для майбутніх досліджень у цій галузі. Він забезпечує багатий матеріал для аналізу та подальших вдосконалень, відкриваючи

ВИСНОВКИ

В кваліфікаційній роботі ми детально розглянули інноваційну модель дистанційного керування для мікроконтролерних систем, яка базується на надійній та гнучкій клієнт-серверній архітектурі. Цей підхід відкриває абсолютно нові горизонти для інтерактивного та віддаленого керування різноманітними пристроями, забезпечуючи розширені можливості та більшу ефективність.

Центральним елементом нашого дослідження стала платформа *Arduino YUN*. Ця унікальна плата, що поєднує мікроконтролер *AVR* та потужний процесор *Linux*, є ідеальною основою для створення складних мережевих пристроїв. На її базі був розроблений та реалізований колісний робот з дистанційним керуванням, який слугує наочним прикладом функціональності та потенціалу запропонованої моделі. Робот оснащений декількома режимами роботи, що дозволяє продемонструвати гнучкість і адаптивність клієнт-серверного керування в реальних умовах.

Запропонована модель віддаленого керування має величезний потенціал для практичного застосування. На її основі можна проектувати та створювати десятки, якщо не сотні, нових пристроїв, які значно полегшать життя та роботу в багатьох сферах. Уявіть, як ця технологія може трансформувати:

- Інженерію: Інженери зможуть керувати промисловими роботами, відстежувати дані з віддалених датчиків або налаштовувати обладнання, не перебуваючи фізично поруч.

- Медицину: Лікарі отримають можливість дистанційно моніторити пацієнтів, керувати медичним обладнанням або навіть виконувати певні діагностичні процедури за допомогою роботизованих систем.

- Транспорт: Водії зможуть використовувати просунуті системи допомоги, а розробники автономних транспортних засобів отримають потужний інструмент для керування та тестування.

- Будівництво: Будівельники зможуть дистанційно керувати важкою технікою, контролювати хід робіт або інспектувати важкодоступні ділянки.

- Допомогу людям з інвалідністю: Ця модель може стати основою для створення адаптивних пристроїв, що значно покращать якість життя людей з обмеженими можливостями, дозволяючи їм легше взаємодіяти з навколишнім світом.

- Розумні будинки: Модель ідеально підходить для розробки комплексних систем "розумного" будинку, де освітлення, клімат, безпека та побутові прилади можуть бути централізовано керовані зі смартфона або комп'ютера, створюючи комфортне та енергоефективне середовище.

Ця технологія є не просто теоретичною концепцією, а практичним рішенням, що має потенціал змінити підходи до автоматизації, зробивши керування пристроями більш доступним, гнучким та ефективним.

56

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Куксенко С. П. Мікроконтролери родини *PIC*: теорія та практика. – Львів : Видавництво Львівської політехніки, 2018. – 412 с.
2. Білоусов В. А. Мікропроцесорні системи та мікроконтролери. – Київ : Центр навчальної літератури, 2019. – 384 с.
3. Робертс М. Основи програмування мікроконтролерів *PIC*. – Харків : Ранок, 2017. – 256 с.
4. Степаненко О. С. Цифрова схемотехніка та мікропроцесорні пристрої. – Одеса : Астропринт, 2020. – 300 с.
5. Григорович В. В. Проектування вбудованих систем на мікроконтролерах. – Дніпро : ДНУ, 2021. – 280 с.
6. Таненбаум Е. С., Уезерілл Д. Комп'ютерні мережі. – 6-те вид. – Санкт Петербург : Пітер, 2019. – 960 с. (Переклад англійською: *Tanenbaum, A. S., & Wetherall, D. Computer Networks*).
7. Іванов В. В. Основи *IoT* (Інтернету речей) та їх застосування. – Київ : Видавничий дім "Києво-Могилянська академія", 2020. – 350 с.
8. Коваленко Р. В. Розробка програмного забезпечення для вбудованих систем з мережевими можливостями. – Львів : Магнолія, 2022. – 290 с.
9. Петров А. Б. Реалізація *HTTP*-сервера на базі мікроконтролерів *ESP32/ESP8266*. – *Збірник наукових праць Військової академії*. – 2023. – № 1. – С. 88–95.

10. Сидоренко Л. М. Застосування протоколу *MQTT* для передачі даних у мікроконтролерних системах. – *Вісник Харківського національного університету радіоелектроніки*. – 2022. – № 2. – С. 60–67.

11. Чумак О. С. Розробка клієнт-серверних рішень для моніторингу параметрів навколишнього середовища. – *Інформаційні технології та системи управління*. – 2021. – № 3. – С. 45–52.

57

12. Гнатюк П. В. Побудова розподілених систем керування на основі мікроконтролерів та клієнт-серверної архітектури. – *Комп'ютерні науки та інженерія*. – 2023. – № 4. – С. 101–108.

13. *RFC 793. Transmission Control Protocol*. – 1981. – [Електронний ресурс]. – Режим доступу: <https://datatracker.ietf.org/doc/html/rfc793> (дата звернення: 07.06.2025).

14. *RFC 2616. Hypertext Transfer Protocol -- HTTP/1.1*. – 1999. – [Електронний ресурс]. – Режим доступу: <https://datatracker.ietf.org/doc/html/rfc2616> (дата звернення: 07.06.2025).

15. *Microchip Technology Inc.* Офіційна документація на мікроконтролери та мережеві модулі. – [Електронний ресурс]. – Режим доступу: <https://www.microchip.com> (дата звернення: 07.06.2025).

16. *EETech. All About Circuits: Networking and Communications*. – [Електронний ресурс]. – Режим доступу: <https://www.allaboutcircuits.com/technical-articles/category/networking-and-communications/> (дата звернення: 07.06.2025).

17. *Hackaday*. Вбудовані системи та мережі. – [Електронний ресурс]. – Режим доступу: <https://hackaday.com/category/embedded-systems/> (дата звернення: 07.06.2025).

18. *PlatformIO. Development Ecosystem for Embedded Systems*. – [Електронний ресурс]. – Режим доступу: <https://platformio.org/> (дата звернення: 07.06.2025).

РЕЦЕНЗІЯ
на кваліфікаційну роботу

випускника спеціальності: 123 «Комп'ютерна інженерія»

відділення: комп'ютерної та програмної інженерії

циклова комісія: комп'ютерних систем та мереж

Максим МЕДВЕДСВ

(ім'я, прізвище)

1. Актуальність теми: Обрана тема кваліфікаційної роботи Застосування клієнт-серверної архітектури в мікроконтролерних системах є актуальною.
2. Кваліфікаційна робота відповідає темі, затвердженій наказом.
3. Завдання на виконання кваліфікаційної роботи виконано у повному обсязі.
4. В результаті виконання кваліфікаційної роботи було створено діючий прототип клієнт-серверної моделі управління платою Arduino YUN.
5. Якість виконання пояснювальної записки та ілюстративного (графічного) матеріалу відповідає вимогам Державних стандартів.
6. В кваліфікаційній роботі зроблений акцент на дані отримані на практиці («живі» експерименти).
7. Кваліфікаційна робота заслуговує оцінку «добре».

Рецензент _____ викладач, «спеціаліст вищої категорії»
(науковий ступінь, посада)

« 30 » 05 _____ 2025 р.

(підпис)

Сергій РУДИЙ
(ім'я, прізвище)

З рецензією ознайомлений _____

(підпис)

Максим МЕДВЕДСВ
(ім'я, прізвище)

ВІДГУК
керівника кваліфікаційної роботи

випускника спеціальності: 123 «Комп'ютерна інженерія»

відділення: комп'ютерної та програмної інженерії

циклова комісія: комп'ютерних систем та мереж

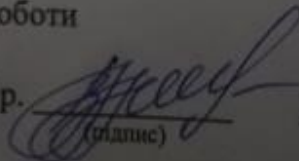
Максим МЕДВЕДСВ

(ім'я, прізвище)

1. Кваліфікаційна робота на тему «Застосування клієнт-серверної архітектури в мікроконтролерних системах» виконана в ініціативному порядку.
2. Метою кваліфікаційної роботи є дослідження мікроконтролерних систем та способів їх управління за клієнт-сервальною архітектурою.
3. Кваліфікаційна робота відповідає темі, затвердженій наказом начальника коледжу.
4. Кваліфікаційна робота виконана здобувачем освіти самостійно.
5. Здобувач освіти показав високі вміння роботи з літературними джерелами, аналіз теоретичного та практичного матеріалу, приймання обґрунтованих рішень, застосування сучасних комп'ютерних інформаційних технологій.
6. Максим МЕДВЕДСВ показав достатній рівень дотримання вимог державних стандартів при виконанні кваліфікаційної роботи в цілому та оформленні пояснювальної записки.
7. Рівень виконаної кваліфікаційної роботи заслуговує оцінку «добре», відповідає набутим випускником знань, умінь та навичок, вимогам освітньої характеристики фахівця і можливість присвоєння йому кваліфікації фахівця освітнього ступеня «бакалавр» спеціальності 123 «Комп'ютерна інженерія».

Керівник кваліфікаційної роботи

« _____ » _____ 2025 р.


(підпис)

Владислав СОБЧУК
(ім'я, прізвище)