**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ**
**Кафедра комп'ютерних наук**

# КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«ФАКУЛЬТАТИВНИЙ КУРС З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КЕРУВАННЯ РОБОТИЗОВАНИМИ ПРИСТРОЯМИ ДЛЯ ПРОФІЛЬНОЇ ШКОЛИ»**

Виконала: студентка 2 курсу, групи 8.0143-ін.з
спеціальності 014 Середня освіта
(шифр і назва спеціальності)
освітньої програми 014.09 Середня освіта (Інформатика)
(назва освітньої програми)
Н. В. Маловічко
(ініціали та прізвище)

Керівник завідувачка кафедри комп'ютерних наук, д. т. н., професор Шило Г. М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри загальної та прикладної фізики, д. пед. н., професор Андрєєв А. М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя – 2024

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти магістр

Спеціальність 014 Середня освіта

(шифр і назва)

Освітня програма 014.09 Середня освіта (Інформатика)

**ЗАТВЕРДЖУЮ**
Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Шило Г.М.
(підпис)

" 5 " березня 2024 р.

# З А В Д А Н Н Я
## НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Маловічко Наталії Валентинівни
(прізвище, ім'я та по-батькові)

1. Тема роботи Факультативний курс з розробки програмного забезпечення керування роботизованими пристроями для профільної школи

керівник роботи Шило Галина Миколаївна, д.т.н., професор
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 01 » березня 2024 року № 384-с

2. Строк подання студентом роботи 01.12.2024

3. Вихідні дані до роботи 1. Платформа Arduino
2. Середовище Arduino IDE
3. Концепції розвитку природничо-математичної освіти (STEM-освіти), затвердженої розпорядженням Кабінету Міністрів України від 05 серпня 2020року № 960-р);

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Огляд літератури та аналіз існуючих факультативних курсів з робототехніки
Основні теоретичні відомості.
Теоретичні основи програмного та апаратного забезпечення для керування роботизованими пристроями

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) презентація

6. Консультанти розділів роботи

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

7. Дата видачі завдання 02.03.2024 р.

## К А Л Е Н Д А Р Н И Й  П Л А Н

| № | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | Розробка плану роботи. | 01.04.2024 | |
| | | | |
| 2. | Збір вихідних даних. | 01.05.2024 | |
| | | | |
| 3. | Обробка методичних та теоретичних | 01.07.2024 | |
| | джерел. | | |
| | | | |
| 4. | Розробка першого та другого розділу. | 01.09.2024 | |
| | | | |
| 5. | Розробка третього розділу. | 01.11.2024 | |
| | | | |
| 6. | Оформлення та нормоконтроль | 25.11.2024 | |
| | кваліфікаційної роботи магістра. | | |
| | | | |
| 7. | Захист кваліфікаційної роботи. | 09.12.2024 | |

Студент _____ Н. В. Маловічко
(підпис) (ініціали та прізвище)

Керівник роботи _____ Г. М. Шило
(підпис) (ініціали та прізвище)

**Нормоконтроль пройдено**

Нормоконтролер _____ О. Г. Спиця
(підпис) (ініціали та прізвище)

# РЕФЕРАТ

Кваліфікаційна робота магістра «Факультативний курс з розробки програмного забезпечення керування роботизованими пристроями для профільної школи»: 96 с., 44 рис., 1 табл., 19 джерел.

ІНФОРМАТИКА, ОСВІТА, ПРОГРАМУВАННЯ, ПРОФІЛЬНИЙ РІВЕНЬ, РОБОТ, РОБОТОТЕХНІКА, ARDUINO, C++.

Об’єктом дослідження є процес інтеграції робототехніки в навчальний процес середніх шкіл України.

Метою дослідження є розробка факультативного курсу з робототехніки для учнів старших класів профільних шкіл України, що відповідає сучасним вимогам та сприяє інтеграції знань у галузі інформатики, фізики, математики та інженерії.

Методи дослідження – аналіз наукових та методичних джерел, моделювання, порівняльний аналіз та описово-теоретичний метод.

У кваліфікаційній роботі проведено дослідження, яке показує важливість інтеграції робототехніки в шкільну освіту для підготовки молоді до майбутніх професій, що пов'язані з автоматизацією, інженерією та програмуванням. В Україні існує значний дефіцит освітніх програм для старшокласників у цій галузі, що потребує розробки нових курсів, орієнтованих на поглиблену підготовку учнів в цій сфері.

Розроблений факультативний курс з розробки програмного забезпечення керування роботизованими пристроями має потенціал не лише покращити навчальний процес в школах, а й сприяти підготовці кваліфікованих кадрів для національної економіки, що активно потребує фахівців у сферах, пов'язаних з роботизацією та автоматизацією.

# SUMMARY

Master's Qualification Thesis «The Optional Course on the Software Development of Controlling Robotic Devices for High School»: 96 pages, 44 figures, 1 table, 19 references.

COMPUTER SCIENCE, EDUCATION, PROGRAMMING, PROFESSIONAL LEVEL, ROBOT, ROBOTICS, ARDUINO, C++.

The object of the study is the process of integrating robotics into the educational process of secondary schools in Ukraine.

The purpose of the study is to develop an optional course in robotics for senior students of specialized schools in Ukraine, which meets modern requirements and promotes the integration of knowledge in the field of computer science, physics, mathematics and engineering.

Research methods - analysis of scientific and methodological sources, modeling, comparative analysis and descriptive-theoretical method.

The qualification work conducted a study that shows the importance of integrating robotics into school education for preparing young people for future professions related to automation, engineering and programming. In Ukraine, there is a significant shortage of educational programs for high school students in this field, which requires the development of new courses focused on in-depth training of students in this area. The developed optional course on the development of software for controlling robotic devices has the potential not only to improve the educational process in schools, but also to contribute to the training of qualified personnel for the national economy, which actively needs specialists in areas related to robotics and automation.

**ЗМІСТ**

Завдання на кваліфікаційну роботу 2
Реферат ..... 4
Summary ..... 5
Вступ ..... 8
1 Огляд літератури та аналіз існуючих факультативних курсів з робототехніки ..... 10
1.1 Аналіз наукових та методичних джерел з впровадження робототехніки в середній освіті ..... 10
1.2 Аналіз існуючих курсів програмування та робототехніки для школярів ..... 16
1.3 Обґрунтування потреби у розробці нового курсу для профільних шкіл ..... 22
2 Теоретичні основи програмного та апаратного забезпечення для керування роботизованими пристроями ..... 24
2.1 Основні концепції розробки програмного забезпечення для роботизованих систем ..... 24
2.2 Архітектури та платформи для робототехніки ..... 34
2.3 Аналіз мов програмування, що використовуються для керування роботизованими системами ..... 41
2.4 Використання Arduino та C++ для розвитку практичних навичок у робототехніці серед старшокласників ..... 48
2.5 Вибір платформи для програмування Arduino ..... 49
3 Розробка факультативного курсу з розробки програмного забезпечення керування роботизованими пристроями для профільної школи ..... 54
3.1 Визначення цілей і завдань курсу ..... 54
3.2 Структура курсу ..... 55

3.3 Розробка навчальних матеріалів (конспекти лекцій, презентації, завдання) ..... 58
Висновки ..... 62
Перелік посилань ..... 63
Додаток А Приклади конспектів уроків 66

# ВСТУП

**Актуальність**. Сучасне суспільство знаходиться на етапі активного розвитку технологій, зокрема робототехніки, яка стає важливою частиною практично всіх сфер економіки. Сьогодні робототехніка впливає на медицину, виробництво, сільське господарство, транспорт, і є основою для нових професій і робочих місць. Для того щоб молодь могла успішно адаптуватися до цих змін, необхідно надавати їй відповідні знання та навички вже на етапі шкільної освіти. В Україні спостерігається дефіцит ефективних освітніх програм з робототехніки, що спричиняє відставання від міжнародних стандартів у підготовці майбутніх спеціалістів. Зокрема, проблема полягає в відсутності розроблених програм для старших класів, які б відповідали вимогам сучасного ринку праці та допомагали учням адаптуватися до нових технологій. Тому розробка факультативного курсу з робототехніки для профільних шкіл є надзвичайно актуальною і необхідною.

**Об’єктом дослідження** є процес інтеграції робототехніки в навчальний процес середніх шкіл України.

**Предметом дослідження** є розробка факультативного курсу з робототехніки для учнів профільних шкіл, що включає теоретичну та практичну підготовку в цій галузі.

**Метою дослідження** є розробка факультативного курсу з робототехніки для учнів старших класів профільних шкіл України, що відповідає сучасним вимогам та сприяє інтеграції знань у галузі інформатики, фізики, математики та інженерії.

Основні завдання дослідження:

- проаналізувати стан впровадження робототехніки в загальну середню освіту України та виявити основні проблеми;
- оцінити потреби учнів у навчанні робототехніці на рівні старших класів;

– розробити програму факультативного курсу, що охоплює основи програмування та створення роботизованих пристроїв;
– визначити методи інтеграції робототехніки в шкільну програму та розробити рекомендації для її ефективного впровадження.

Для вирішення задач були застосовані такі методи дослідження:

– аналіз наукових та методичних джерел – для вивчення існуючих підходів до впровадження робототехніки в освіту та розробки програм для старших класів;
– моделювання – для створення структури факультативного курсу з робототехніки;
– порівняльний аналіз – для оцінки наявних освітніх програм та курсів, що пропонуються на ринку;
– описово-теоретичний метод – для аналізу теоретичних основ робототехніки та її впливу на розвиток учнів.

Практичне значення дослідження полягає в розробці конкретної програми факультативного курсу з робототехніки, яка може бути впроваджена в ліцеях України, що дозволить учням старших класів отримати необхідні технічні знання та навички. Наукове значення полягає в розробці теоретичних засад для інтеграції робототехніки в середню освіту та створенні нової моделі навчання для профільних шкіл.

Результати дослідження можуть бути використані в освітніх установах для розробки та впровадження нових програм з робототехніки на рівні старших класів. Також вони можуть бути впроваджені в системи підвищення кваліфікації вчителів, що працюють в сфері STEM-освіти.

У кваліфікаційній роботі розглянуто проблему впровадження робототехніки в середні школи України, зокрема для учнів старших класів. Розроблено факультативний курс з робототехніки, який включає як теоретичні, так і практичні аспекти, що сприяють розвитку критичного мислення та технічних навичок у учнів. Дослідження охоплює аналіз існуючих програм та курсів, а також розробку рекомендацій щодо інтеграції новітніх технологій у шкільну освітню програму.

# 1 ОГЛЯД ЛІТЕРАТУРИ ТА АНАЛІЗ ІСНУЮЧИХ ФАКУЛЬТАТИВНИХ КУРСІВ З РОБОТОТЕХНІКИ

## 1.1 Аналіз наукових та методичних джерел з впровадження робототехніки в середній освіті

Робототехніка – одна з ключових галузей майбутнього, яка розвивається шаленими темпами і стає невід’ємною частиною багатьох сфер, таких як медицина, сільське господарство, транспорт, виробництво та навіть сфера обслуговування.

Аналіз міжнародних трендів у сфері робототехніки свідчить про такі основні моменти: суттєве збільшення виробництва промислових роботів, що за даними Міжнародної федерації робототехніки (IFR) подвоїлося в період з 2013 до 2017 років; очікуване прискорення темпів виробництва в 2019–2021 роках із середнім річним зростанням на 14% за оцінками IFR; активне впровадження роботизованих систем і автоматизації у різних галузях, включаючи промисловість, військову та космічну сфери, автомобільну галузь, медицину та обслуговування.

Розвивається концепція «розумних фабрик» (Smart Factories), яка передбачає повністю автоматизоване виробництво з мінімальним втручанням людини, використовуючи комп'ютерні системи та сенсори для зворотного зв’язку. Прискорення автоматизації очікується й у наступні шість років, що підтверджує дослідження Всесвітнього економічного форуму, яке прогнозує зміну співвідношення «людина-робот» на користь останнього до 2025 року.

Також відзначається зростаючий інтерес великих компаній до інвестицій у робототехнічні стартапи, прикладом чого є придбання Google у 2014 році восьми компаній та запуск платформи Google Cloud Robotics у 2019 році, що об'єднує робототехніку, хмарні технології та штучний інтелект. Попит на спеціаліс-

тів у сфері робототехніки зростає через необхідність розробки, конструювання та програмування роботів. Нарешті, популярність робототехніки як тренду у сфері освіти в Україні та світі збільшується, що відображає високий попит на кваліфікованих фахівців [15].

Вивчення робототехніки допоможе учням підготуватися до світу, де технології будуть оточувати їх у всіх аспектах життя. Раннє знайомство з цією галуззю дає змогу зрозуміти сучасні технологічні тренди та спонукає до активного й свідомого використання інновацій.

Беручи до уваги результати досліджень та власний досвід, можна виокремити основні цілі та завдання інтеграції освітньої робототехніки у навчальний процес навчальних закладів. Зокрема, це:

- стимулювання та розвиток інтересу учнів до природничих і точних наук, а також до науково-технічної творчості, що відповідає принципам STEM-освіти;
- набуття учнями навичок роботи з технічними пристроями та здатності вирішувати актуальні інженерно-технічні завдання на практиці;
- виховання в учнів особистісних якостей, які допомагають самостійно визначати цілі, планувати шляхи їх досягнення, контролювати й оцінювати свої результати;
- розвиток умінь працювати з різноманітними інформаційними джерелами, аналізувати їх та формулювати власні судження, висновки й створювати власні проєкти;
- забезпечення міжпредметних зв’язків між інформатикою, математикою, фізикою і технологіями;
- формування наукового світогляду як важливої складової загальної культури, що є необхідною для повноцінного життя в сучасному суспільстві;
- розвиток стабільної мотивації до навчання;
- інтелектуальний розвиток учнів, що включає формування логічного, алгоритмічного та творчого мислення для вирішення практичних зав-

дань, розвиток інформаційної культури, пам’яті, уваги та наукової інтуїції [2].

Робототехніка охоплює широкий спектр дисциплін: математику, фізику інформатику, електроніку та механіку. Заняття робототехнікою дозволяють учням отримати міждисциплінарні знання в цікавій і практичній формі, що допомагає формувати цілісне бачення науки та технологій. STEM (наука, технології, інженерія, математика)  навички є фундаментальними для багатьох професій, а також сприяють розвитку критичного мислення, аналізу та творчого підходу до вирішення проблем.

STEM-освіта є одним із засобів підготовки фахівців майбутнього, які здатні мислити творчо та створювати новаторські рішення. Навчання у сфері STEM слугує базою для підготовки спеціалістів у галузі передових технологій. У багатьох розвинутих країнах світу STEM-освіті надається підтримка на державному рівні, що зумовлює розвиток національних програм у таких країнах, як Австралія, Великобританія, Данія, Ізраїль, Китай, Корея, Сінгапур, США та Японія. Прогнозовані оцінки щодо ринку праці у США свідчать, що у період з 2017 по 2027 роки кількість робочих місць у сферах STEM зросте на 13% порівняно з 9% зростанням у сферах, що не належать до STEM. Особливо помітні ці зміни у галузях комп’ютерних наук, інженерії, машинобудування та виробництва. На сьогодні частка STEM-професій становить 7% від усіх робочих місць у США, а попит на фахівців у цих сферах продовжує зростати.

Багато європейських роботодавців також стикаються з труднощами при пошуку кваліфікованих фахівців у сфері STEM, особливо спеціалістів з ІКТ. За останніми даними PISA, значна частина 15-річних підлітків у Європі демонструє функціональну неграмотність у математиці та природничих науках. Україна також має серйозні проблеми у цій сфері: показники інновацій та технологічної забезпеченості працівників залишаються одними з найнижчих серед інших складових конкурентоспроможності країни. Згідно зі звітом про конкурентоспроможність за 2018 рік, Україна зайняла 83 місце серед 140 країн, а за рівнем

доступності інновацій – 58 місце. Ці показники є типовими для держав пострадянського простору.

Однак, інновації, модернізація системи освіти та підготовка кадрів визначені як пріоритети в стратегії "Європа 2020". Підвищення потенціалу та розвиток інноваційних зв’язків між наукою і суспільством є важливим завданням не лише для ЄС, але й для України. Школи мають ключову роль у подоланні розриву між навичками майбутніх фахівців та високим попитом на кваліфікованих спеціалістів у STEM-сферах. Важливо розвивати в дітей з раннього віку навички та зацікавленість у STEM-дисциплінах [3].

В Україні інтерес до STEM-освіти зростає, проте впровадження залишається епізодичним, без системного підходу. Важливою є роль робототехніки, яка, як складова STEM, допомагає розвивати ключові компетентності учнів, такі як критичне та системне мислення, комунікаційні навички, здатність працювати в команді та вирішувати практичні завдання. Автори підкреслюють, що заняття з робототехніки стимулюють навчальну мотивацію учнів, покращують дослідницькі здібності, надають можливість створювати власні продукти та виражати творче мислення.

Опитування серед вчителів і майбутніх учителів показало, що 92% респондентів вважають робототехніку ефективним інструментом навчання через гру, а 79% відзначають її значну мотиваційну цінність для учнів. Під час таких занять учні отримують навички 21 століття: здатність вирішувати реальні проблеми, працювати над проєктами, швидко шукати та опрацьовувати інформацію, працювати в команді, а також критично та системно мислити [2].

Підвищення значущості STEM-освіти є одним з пріоритетних напрямів реформування освіти, частиною державної політики щодо підвищення конкурентоспроможності національної економіки та розвитку людських ресурсів, а також одним з ключових чинників інноваційної діяльності в освітній сфері, що відповідає потребам суспільства й економіки. STEM-освіта спрямована на всебічний розвиток особистості через формування компетентностей, природничо-наукової картини світу, світоглядних позицій та життєвих цінностей за допомогою трансдисцип-

лінарного підходу до навчання, що базується на практичному використанні наукових, математичних, технічних і інженерних знань та умінь для вирішення реальних завдань, що застосовуються в майбутній професійній діяльності.

Використання основного принципу STEM-освіти – інтеграції – дозволяє оновити методологічні основи, зміст та обсяг навчального матеріалу природничо-математичних предметів, а також забезпечити технологізацію процесу навчання. Це допомагає формувати навички вирішення складних практичних завдань, критичне мислення, креативність, когнітивну гнучкість, організаційні та комунікативні здібності, здатність оцінювати проблеми та приймати рішення, свідомий вибір і підготовку до майбутньої професії, фінансову грамотність, цілісне наукове сприйняття, ціннісні орієнтири, загальнокультурну та технологічну компетентності, математичну і природничу грамотність. Це сприяє всебічному розвитку особистості, розкриттю її нахилів і здібностей, оволодінню засобами пізнавальної, дослідницької та практичної діяльності, а також вихованню прагнення до навчання протягом усього життя й вмінню творчо застосовувати знання на практиці. Особливу увагу в інтегративному підході реалізації STEM-освіти приділено якості викладання математики.

STEM-освіта реалізується на різних рівнях освіти: початковий рівень забезпечує стимулювання допитливості, підтримку інтересу до навчання та розвиток навичок дослідження й технічної творчості; базовий рівень сприяє стійкому інтересу до природничо-математичних предметів, технологічній грамотності та навичкам вирішення проблем, залученню до дослідницької й проєктної діяльності; профільний рівень дає можливість поглибленого опанування знань і вмінь STEM, виконання інноваційних проєктів. На вищому/професійному рівні здійснюється підготовка спеціалістів технічних і наукових напрямів у закладах вищої освіти та підвищення кваліфікації педагогів через впровадження нових методик і курсів.

STEM-освіта розвивається за умови поєднання різних форм навчання: формальної, неформальної та інформальної освіти. Її розвиток забезпечується співпрацею між освітніми установами, науковими інститутами, лабораторіями,

музеями, природничими центрами, підприємствами та громадськими організаціями, залученими до створення освітнього середовища. Першочерговими завданнями для ефективного розвитку STEM-освіти є розробка науково-методичного забезпечення, впровадження сучасних навчальних засобів, підготовка та підвищення кваліфікації педагогічних кадрів, розширення мережі регіональних STEM-центрів і лабораторій, проведення наукових досліджень, аналіз розвитку STEM-освіти та прогнозування її подальших тенденцій [4].

Конструювання та програмування роботів вчить учнів структурувати мислення і шукати рішення для складних задач. Наприклад, якщо робот не виконує команду, потрібно проаналізувати, де саме вийшла помилка: чи в коді, чи в з'єднаннях, чи в самій конструкції робота. Такі виклики розвивають в учнів логічне мислення, терпіння та вміння працювати з помилками, що надзвичайно корисно в будь-якій діяльності.

Робототехніка дозволяє дітям бути винахідниками. Завдяки доступу до конструкторів, мікроконтролерів та програмного забезпечення, учні можуть створювати власні проєкти, експериментувати і втілювати власні ідеї в життя. Така свобода сприяє розвитку творчого підходу та здатності до інновацій. Наприклад, учень може спроєктувати робота для збирання сміття, поливу рослин або навіть для гри в шахи – можливості безмежні.

Робототехніка зазвичай передбачає командну роботу, оскільки на більшість проєктів потрібен розподіл ролей: один учень може займатися дизайном, інший – програмуванням, третій – тестуванням. Це допомагає розвивати навички співпраці, вміння спілкуватися, ділитися ідеями та працювати над досягненням спільної мети, що важливо для успіху в будь-якій сучасній професії.

Проєкти з робототехніки дозволяють учням бачити результати своїх зусиль у реальному часі. Наприклад, коли учень створює робота, який дійсно виконує певні дії, це може викликати відчуття гордості та впевненості у власних силах. Така «видимість» результатів навчання часто підвищує мотивацію до навчання, особливо у тих учнів, яким традиційні методи навчання здаються менш цікавими.

Оскільки автоматизація охоплює дедалі більше галузей, знання з робототехніки можуть стати вирішальними у виборі майбутньої професії. Навіть якщо учень не планує стати інженером чи програмістом, розуміння принципів роботи автоматизованих систем та основ програмування є цінними навичками в будь-якій професії. Вміння управляти технічними пристроями, розуміти принципи роботи роботів або навіть мати базові знання програмування робить учня більш конкурентоздатним на ринку праці.

## 1.2 Аналіз існуючих курсів програмування та робототехніки для школярів

Питання аналізу існуючих курсів вже піднімалось у 2018 році Струтинською О. В. та Барановим С. С [17]. Було проаналізовано розвиток освітньої робототехніки в Україні та світі, яка на сьогодні є важливим освітнім трендом. Однак в Україні відсутнє системне викладання робототехніки в рамках шкільної освіти, що створює значну потребу у впровадженні цього напрямку в позашкільну освіту. У зв'язку з цим заклади позашкільної освіти, як державні, так і комерційні, активно розвивають освітню робототехніку, пропонуючи курси та гуртки, що сприяють розвитку технічних навичок у молоді. Це особливо важливо, оскільки робототехніка відкриває нові професії, такі як оператори роботів, програмісти, інженери, що відповідає вимогам сучасного ринку праці.

Для забезпечення ефективного навчання в цій сфері є нагальна потреба у підготовці кваліфікованих викладачів робототехніки, що потребує розробки спеціалізованих навчальних програм у педагогічних університетах. Оскільки робототехніка має важливе значення для майбутнього, запропоновано зробити її обов'язковою складовою шкільної програми, інтегруючи її в курси інформатики та технологій, а також як частину STEM-освіти [4]. Також визначено, що подальші дослідження повинні бути спрямовані на моніторинг розвитку робо-

тотехніки та оновлення навчальних програм, щоб відповідати вимогам часу та готувати педагогів до викладання цієї важливої дисципліни.

Також були детально розглянуті та проаналізовані по різним критеріям комерційні гуртки України та зроблений рейтинг популярності серед них, але з часу того дослідження багато чого змінилося. З початком пандемії COVID-19 та повномасштабного вторгнення Росії в Україну в лютому 2022 року освітній процес зазнав значних змін. Багато закладів освіти були змушені перейти на дистанційне навчання або тимчасово припинити свою діяльність через безпекову зупинку та руйнування інфраструктури. Згідно з даними, понад 10 мільйонів людей були вимушені покинути свої домівки, а понад 1 700 закладів освіти постраждали від бойових дій, з яких 144 були повністю зруйновані [1].

У таких умовах комерційні курси з робототехніки, відзначили зниження кількості відвідувачів. Пандемія та війна спричинили економічні труднощі, що вплинули на фінансові можливості населення, а також на безпеку, яка обмежила мобільність та доступ до навчальних закладів. Наприклад, багато підприємств було зруйновано або припинили свою діяльність, а працівники виїхали в інших регіонах України або за кордон, що створило дефіцит кваліфікованих кадрів [3].

Однак варто відзначити, що деякі освітні заклади адаптовані до нових умов, впровадивши онлайн-курси та дистанційне навчання, що дозволило зберегти навчальний процес навіть під час обмеження.

Виникла потреба зробити новий огляд комерційних курсів. За допомогою пошукача Google у листопаді 2024 року,був зроблений запит «Курси робототехніки для дітей» і з результатів пошуку обиралися самі популярні курси. Було відібрано сім організацій які надають послуги навчання дітей робототехніці. Після аналізу отриманої інформації були визначені переваги та недоліки кожної фірми.




Рисунок 1.1 – Послуги з навчання робототехніки Комп'ютерної академії IT Step




Рисунок 1.2 – Послуги з навчання робототехніки STEM школи INVENTOR

ними рівнями навчання

**IT Step.** Для підлітків 9–12 років пропонуються курси, що включають роботу з роботами LEGO та гаджетами на базі Arduino. Для старшокласників 13–14 років передбачено програми, де учні створюють технічні проекти та програмують за допомогою мови Pyton.

До переваг можна віднести наявність навчальних центрів в багатьох містах України. Інформація по Комп'ютерній академії IT Step відображена на рис. 1.1.

До недоліків можна віднести те, що робототехніка не виділена окремим курсом, а вписана в загальну програму навчання, на ряду з багатьма іншими напрямками (веб-дизайн, 3-d моделювання, ігровий дизайн, тощо) інформаційних технологій, при цьому розраховане навчання на декілька років.

**STEM-школа INVENTOR.** Пропонуються навчальні програми для дітей від 8 до 16 років, використовуючи конструктори LEGO Education та платформи Romeo BLE. Учні мають можливість вивчати програмування роботів на таких мовах, як Scratch, C++ та Python. Також серед переваг школи – можливість окремо обрати курс з робототехніки з різ-

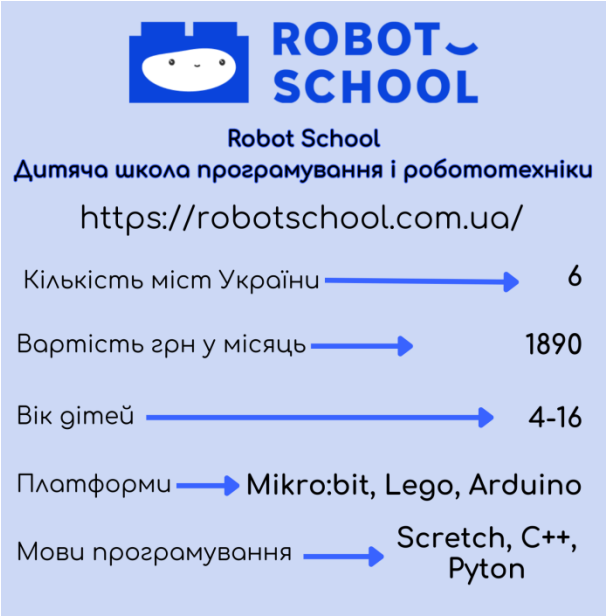


Рисунок 1.3 – Послуги з навчання робототехніки школи Robot School

До недоліків STEM-школи INVENTOR можна віднести відсутність у навчальній програмі платформи Arduino. Інформація щодо цієї компанії зображена на рис. 1.2.

**Robot School.** Ця дитяча школа програмування та робототехніки, яка навчає дітей віком від 4 до 16 років за системою STEM. Використовуючи конструктори Lego, Arduino та Mikro:bit, школа дає теоретичні та практичні знання з робототехніки та програмування за допомогою Scratch, C++ та Python. До переваг віднесемо гарний набір для навчання платформ і мов програмування та можливість обрати програму різних рівнів. Головним недоліком цієї школи програмування та робототехніки можна відзначити обмежену кількість місць для занять. Інформація по школі Robot School на рис 1.3.

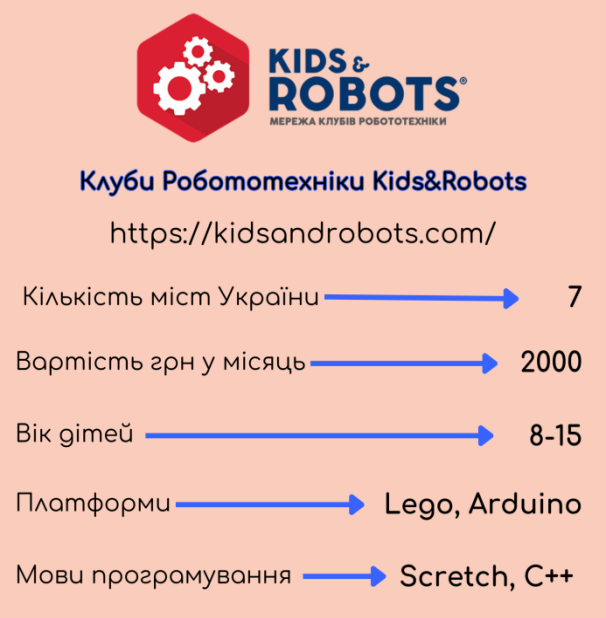


Рисунок 1.4 – Послуги з навчання робототехніки у клубі Kids&Robots

**Kids&Robots.** Клуби робототехніки Kids&Robots спеціалізуються на технічному розвитку дітей віком від 8 до 15 років. Вони пропонують різноманітні курси, які охоплюють такі напрямки, як механіка, логіка, конструювання та програмування. таким чином, діти можуть працювати з робототехнікою Lego та Arduino, що дозволяє їм вивчати програмування за допомогою таких мов, як Scratch та C++. Програми навчання клубів спрямовані на розвиток критичного

мислення, творчих здібностей та інженерних навичок у дітей. Переваги та недоліки клубів Kids&Robots співпадають з особливостями школи робототехніки Robot School. Додаткову інформацію про клуби робототехніки Kids&Robots можна переглянути на рис. 1.4.

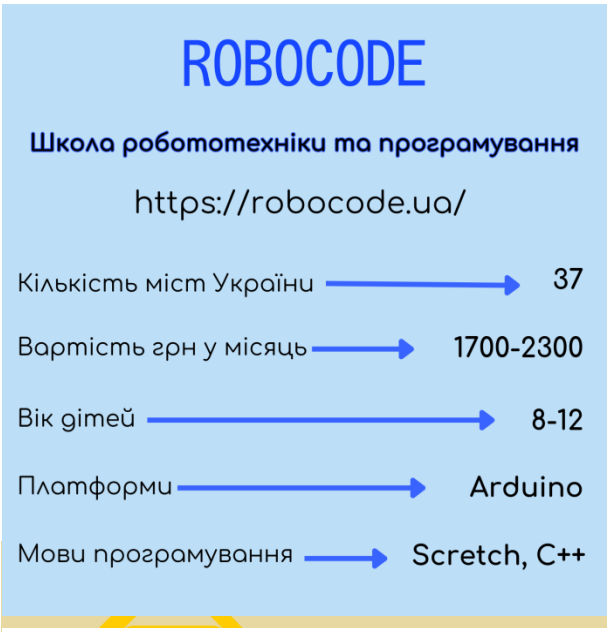


Рисунок 1.5 – Послуги з навчання робототехніки у школі ROBOCODE

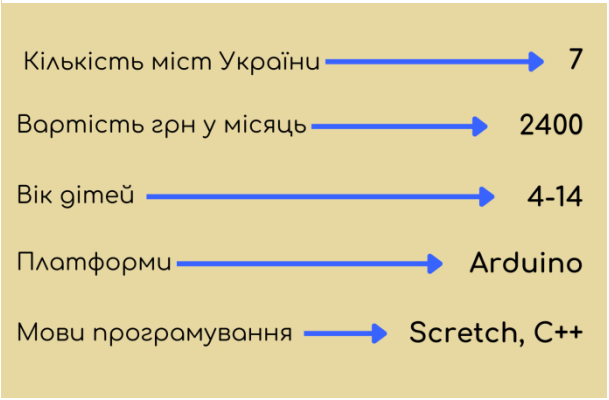


Рисунок 1.6 – Послуги з навчання робототехніки у школі ROBOCODE

**ROBOCODE.** Школа робототехніки та програмування ROBOCODE спеціалізується на навчанні дітей віком від 8 до 12 років. Вона пропонує різноманітні курси, включаючи робототехніку на платформі Arduino та програмування на мовах C++, та Scratch.

Перевагою школи ROBOCODE є велика кількість локацій для проведення занять, а недоліком можна вважати використання лише однієї платформи для вивчення робототехніки. Дані по роботі цієї компанії на рис. 1.5.

**Robo.Hous.** Пропонуються різноманітні курси з робототехніки для дітей від 4 до 14 років на платформі Arduino та програмуванні на таких мовах, як Scratch та C++.

Ці курси охоплюють основи електроніки, створення робіт та інші аспекти інженерії, які дають дітям можливість розвивати технічні та творчі навички. Компанія вибрала найбільш популярну систему Arduino, навчання на якій проходить на різних рівнях – від базового до більш складного, що дозволяє дітям розвивати нові технології. Курси включають програму використання сучасних створених комплектів та інструментів для розробки, що дозволяє дітям реалізовувати власні проекти. До недоліку віднe-

семо невелику кількість місць для навчання, що може обмежити доступ до курсів. Інформацію про цю школу можна побачити на рис. 1.6.




Рисунок 1.7 – Послуги з навчання робототехніки у Академії Професій Майбутнього

**Академія Професій Майбутнього.** Пропонуються різноманітні курси з робототехніки для дітей віком від 6 до 16 років, спрямовані на розвиток технічних навичок та інженерного мислення. Для навчання обрано декілька різних конструкторів Lego та платформа Arduino, а також для програмування цих систем дітей навчають C++ та Python, що можна віднести до переваг компанії. До недоліків віднесемо дуже маленьку кількість місць навчання.

Також було розглянуто більш детально інформацію про кількість місць навчання у містах Київ, Запоріжжя і Дніпро. На рис. 1.8 ми бачимо, що дуже не рівномірно розподілені школи для навчання.

Більш детально інші недоліки організацій, які надають послуги навчання дітей робототехніці розглянуто у наступному пункті кваліфікаційної роботи.

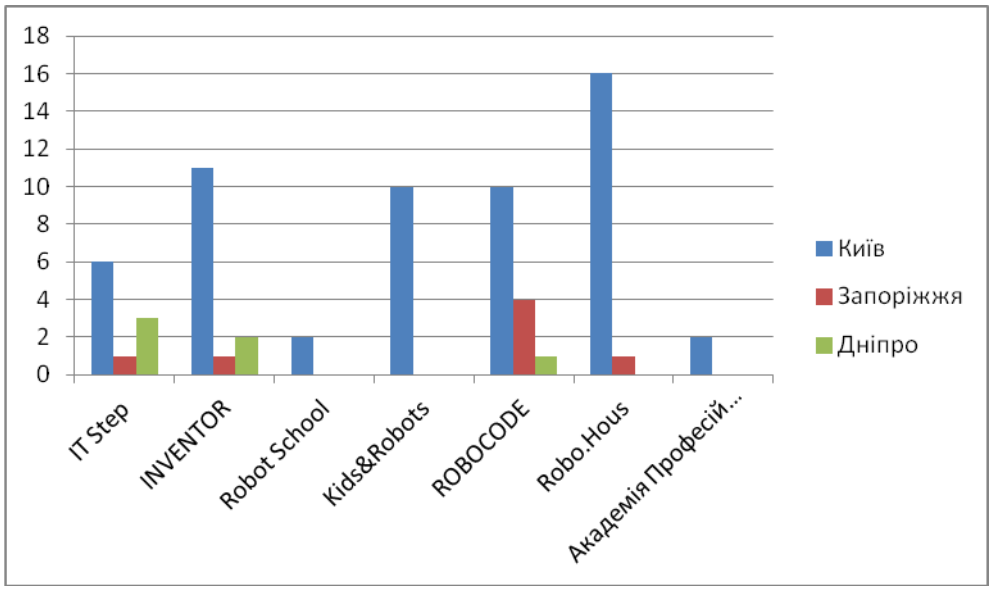

Рисунок 1.8 – Порівняння кількості шкіл з робототехніки в деяких містах України

## 1.3 Обґрунтування потреби у розробці нового курсу для профільних шкіл

На основі проведеного дослідження стало очевидним, що розвиток робототехніки в школах України є надзвичайно важливою і нагальною потребою. Це питання потребує термінового вирішення, адже сучасний розвиток технологій і автоматизації вимагає від молоді набуття відповідних навичок уже на етапі шкільного навчання. Попри наявність великої кількості комерційних курсів з робототехніки для дітей, ситуація в освітній системі України залишає бажати кращого. Найбільше викликає занепокоєння відсутність доступних програм для старших класів, особливо для учнів профільних шкіл.

Зокрема, з усіх доступних програм, єдина модельна програма, яка охоплює робототехніку, розроблена для учнів 7-9 класів, автори якої – Сокол І. М. і Ченцов О. М. [14]. Однак така програма має значні обмеження, адже вона призначена лише для учнів «Нової української школи», що в свою чергу обмежує її охоплення і доступність для більшості учнів країни. Для старших класів, а тим більше для профільних шкіл, не існує жодних програм, що робить нагальним створення факультативного курсу з робототехніки для старшокласників, зокрема для профільних навчальних закладів.

До того ж, навіть наявні комерційні курси робототехніки для дітей мають низку недоліків. Перший з них полягає в обмеженій географії їхнього проведення. Курси доступні не у всіх містах, і навіть не в кожному обласному центрі є можливість відвідувати заняття з робототехніки. Це створює нерівність у можливостях для дітей, які мешкають не в обласних центрах або навіть у селах. Тому дуже важливо створювати умови для розвитку навичок робототехніки не тільки в великих містах, а й для учнів з менших населених пунктів, де таке навчання відсутнє.

Другим суттєвим недоліком є висока вартість навчання. Більшість комерційних курсів передбачають навчання, яке триває мінімум рік, і вартість цих курсів варіюється від двох до п’яти тисяч гривень на місяць. Для багатьох сімей, особливо в умовах військового стану та економічної нестабільності, такі витрати є непосильними. Це робить доступ до таких курсів для великої частини населення України майже неможливим.

Оскільки більшість комерційних курсів не покривають потреби старшокласників, а також мають обмеження за віковими категоріями (14 років для більшості курсів, рідше – до 16 років), з'являється необхідність у спеціалізованих програмах, що орієнтовані на старшокласників, здатних до глибшого опанування робототехнікою. Тому розробка факультативного курсу для профільних шкіл, який враховуватиме потреби учнів старших класів, є не лише бажаною, а й необхідною.

Також, враховуючи досвід дистанційного навчання, яке стало звичним унаслідок пандемії та війни, можна розглянути можливість введення елементів онлайн-занять для робототехніки. Проте, варто зазначити, що робототехніка вимагає практичних занять, і дистанційне навчання повинно бути використано лише як виняток, наприклад, у випадках відсутності учня через хворобу або неможливість відвідати заняття. Для цього можуть бути застосовані онлайн-симулятори, такі як Tinkercad, але тільки для теоретичної частини або як допоміжний інструмент для практичних завдань [11].

Таким чином, розробка факультативного курсу з робототехніки для профільних шкіл є важливим і необхідним кроком для подолання існуючих прогалин в освітній системі України. Це дозволить забезпечити рівний доступ до якісної освіти з робототехніки для учнів незалежно від їхнього місця проживання, а також знизить фінансове навантаження на родини. Водночас, курс має бути орієнтований на старшокласників і включати в себе як теоретичну частину, так і практичні заняття, що забезпечить учням можливість реалізувати свої навички в реальних умовах.

# 2 ТЕОРЕТИЧНІ ОСНОВИ ПРОГРАМНОГО ТА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КЕРУВАННЯ РОБОТИЗОВАНИМИ ПРИСТРОЯМИ

## 2.1 Основні концепції розробки програмного забезпечення для роботизованих систем

**Основи робототехніки** є ключовим аспектом розробки програмного забезпечення для роботизованих систем. Робототехніка охоплює широкий спектр систем, які мають здатність виконувати завдання з використанням механічних, електронних і програмних компонентів. Існують різні типи роботизованих систем, кожна з яких має свою специфіку та сферу застосування [5].

Промислові роботи зазвичай застосовуються на виробничих лініях для виконання завдань, таких як складання, пакування, зварювання та фарбування. Вони мають високу точність і здатність працювати в умовах високих навантажень і небезпечних або забруднених середовищах, що дозволяє автоматизувати виробничі процеси (рис. 2.1).



Рисунок 2.1 – Промисловий робот

Сервісні роботи призначені для надання різноманітних послуг людині в повсякденному житті. Вони можуть включати домашніх помічників, роботи для прибирання або доставки товарів, що забезпечують взаємодію з людьми і адаптацію до змінних умов (рис. 2.2).



Рисунок 2.2 – Сервісний робот

Мобільні роботи здатні переміщуватися по різних поверхнях, що дозволяє їм використовуватися в таких сферах, як дослідження, транспорт і безпека. Ці роботи можуть бути оснащені колесами, гусеницями або іншими системами для забезпечення рухливості, що дозволяє їм ефективно функціонувати в різноманітних середовищах (рис. 2.3).




Рисунок 2.3 – Мобільний робот

Когнітивні роботи використовують штучний інтелект для розпізнавання і обробки інформації, а також для прийняття рішень в умовах непередбачуваних ситуацій. Вони знаходять застосування в інтелектуальних системах і роботах, що потребують високої адаптивності до змінюваних умов (рис. 2.4).

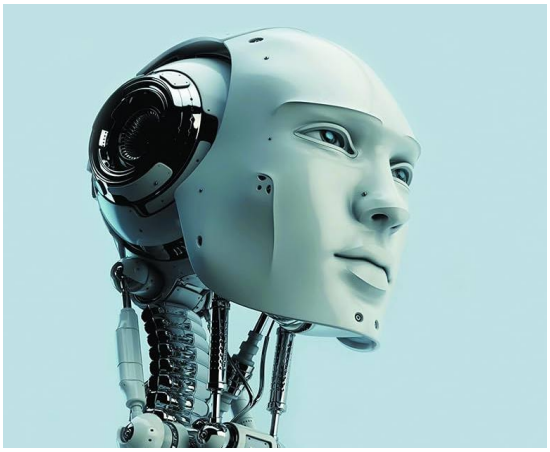

Рисунок 2.4 – Когнітивний робот

Роботи для досліджень використовуються для збору даних в екстремальних умовах, таких як космос або глибокий океан. Ці роботи здатні працювати в середовищах, де людська присутність є небезпечною або неможливою, завдяки своїм автономним функціям (рис. 2.5).

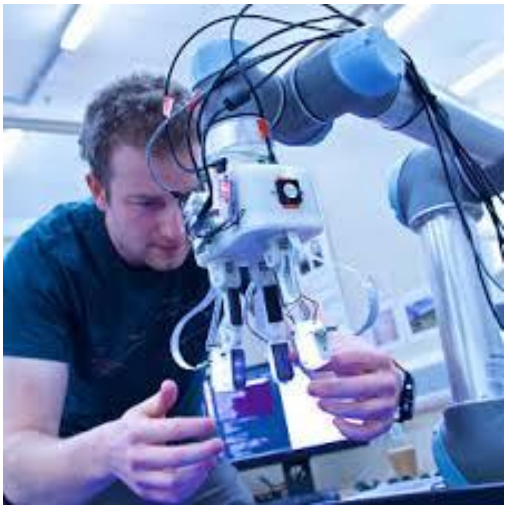

Рисунок 2.5 – Робот для досліджень

Людські роботи або екзоскелети, в свою чергу, допомагають людям підвищити їхні фізичні можливості або полегшити виконання важких фізичних завдань. Вони можуть бути використані в медицині, реабілітації або в умовах, де потрібно знизити навантаження на людину (рис. 2.6).



Рисунок 2.6 – Екзоскелет

Роботи для навчання та досліджень призначені для демонстрації і вивчення основ робототехніки та програмування. Вони використовуються в навчальних закладах і дають можливість ознайомитися з основами створення і програмування роботів (рис. 2.7).

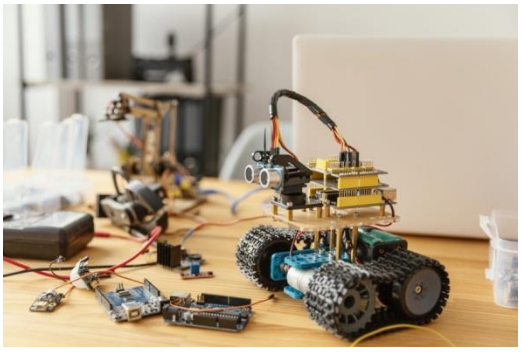

Рисунок 2.7 – Робота для навчання

Інша популярна категорія – дрони, які застосовуються для моніторингу, зйомки, доставки або дослідження важкодоступних територій (рис. 2.8).



Рисунок 2.8 – Дрон

Основними компонентами будь-якого робота є сенсори, приводи та контролери. Сенсори, або датчики, забезпечують здатність робота сприймати навколишнє середовище (рис. 2.9). Вони можуть вимірювати різні фізичні параметри, такі як температура, відстань, світло, звук чи наявність об’єктів. Наприклад, ультразвукові сенсори дозволяють вимірювати відстань до об'єктів, тоді як камери або лазерні сканери забезпечують створення карт навколишнього середовища. Інші сенсори, як-от акселерометри чи гіроскопи, використовуються для визначення положення робота або стабілізації його руху [18].

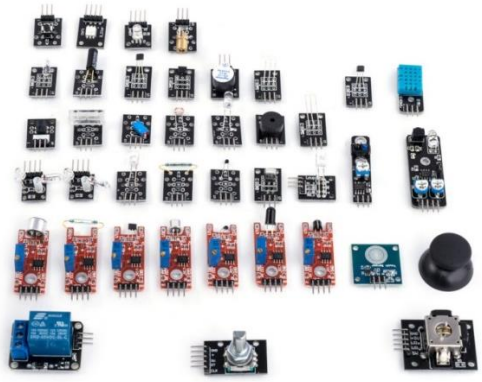

Рисунок 2.9 – Сенсори

Приводи виконують функцію «м'язів» робота. Вони забезпечують механічний рух або дію (рис. 2.10). До приводів належать двигуни, серводвигуни, пневматичні або гідравлічні приводи. Наприклад, для мобільного робота це можуть бути електродвигуни, які приводять у рух колеса або гусениці, а для маніпулятора – серводвигуни, які дозволяють точно позиціонувати маніпулятор у просторі. Правильна робота приводів залежить від алгоритмів управління, які забезпечують потрібну швидкість, силу або кут повороту, залежно від завдання [7].



Рисунок 2.10 – Приводи

Контролери є «мозком» роботизованої системи (рис. 2.11). Вони забезпечують обробку даних від сенсорів, прийняття рішень та надсилання команд до приводів. Контролери можуть бути представлені мікроконтролерами (наприклад, Arduino) або більш потужними обчислювальними системами (наприклад, Raspberry Pi). Залежно від складності завдань, контролери можуть виконувати як прості опе-

рації, такі як керування двигуном на основі сигналу з датчика, так і складні обчислення, наприклад, обробку зображень або побудову маршруту руху [9].

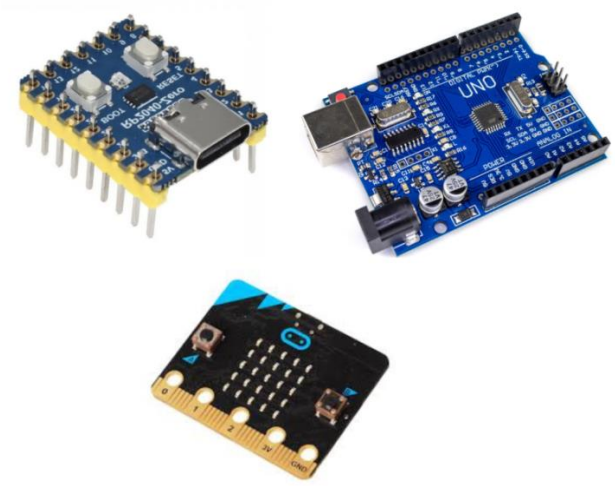


Рисунок 2.11 – Контролери

Інтеграція цих компонентів дозволяє роботам виконувати завдання, для яких вони були розроблені. Наприклад, у мобільному роботі сенсори сприймають навколишнє середовище, контролер обробляє отриману інформацію, визначає найкращий маршрут і передає команди двигунам, щоб забезпечити безпечний рух. У промисловому маніпуляторі система сенсорів може виявляти об'єкти на конвеєрі, після чого контролер визначає координати цього об'єкта і відправляє команди до серводвигунів, щоб робот міг захопити його і перемістити в потрібне місце.

Таким чином, основи робототехніки будуються на гармонійній взаємодії сенсорів, приводів і контролерів, що дозволяє створювати роботизовані системи, здатні працювати автономно, адаптуватися до змін середовища та виконувати складні завдання з високою точністю й ефективністю.

При розробці програмного забезпечення для керування роботизованими засобами важливо розглянути ключові концепції та принципи, що використовуються при створенні програмного забезпечення для роботизованих систем.

Однією з ключових концепцій є **модульність** програмного забезпечення, що передбачає розподіл програмного коду на окремі незалежні, але взаємопов'-

язані функціональні блоки або модулі. Модульна архітектура дозволяє легко масштабувати програмне забезпечення, забезпечує його підтримку, а також дає змогу швидко вносити зміни або оновлення без порушення роботи всієї системи. Наприклад, в роботизованих системах часто застосовують модульну організацію для керування окремими підсистемами, такими як управління рухом, обробка даних з сенсорів або алгоритми планування траєкторії. Програмування в стилі модульної архітектури дозволяє знизити складність, полегшити тестування та підтримку системи, а також забезпечити розширюваність у майбутньому.

Інша важлива характеристика архітектури програмного забезпечення для роботів – **реактивність**. У контексті роботизованих систем це означає, що система повинна бути здатна миттєво реагувати на зовнішні події, такі як зміни в навколишньому середовищі або сигнали від сенсорів. Реактивність є критично важливою, коли робота потрібно адаптувати до динамічно змінюваного оточення або виконувати завдання, які вимагають швидкого реагування, наприклад, ухиляння від перешкод чи виконання маніпуляцій з об'єктами.

Для реалізації реактивних систем часто використовують підходи на основі систем подій, де програма "чекає" на події, що виникають в навколишньому середовищі, і реагує на них, виконуючи необхідні дії.

**Розподіленість** є ще однією суттєвою характеристикою архітектури програмного забезпечення для роботизованих систем. Вона стосується розподілу обчислювальних завдань серед кількох пристроїв або обчислювальних одиниць. Зазвичай, робот складається з декількох підсистем, таких як центральний процесор, система управління рухом, сенсори для виявлення оточення, приводи для здійснення рухів та інші. Кожна з цих підсистем має свої власні обчислювальні ресурси і завдання, тому програмне забезпечення має бути здатне ефективно керувати комунікацією між ними.

Для цього використовують розподілені обчислювальні моделі, які можуть включати як централізовані, так і дистрибутивні підходи до обробки даних. Зазвичай це реалізується за допомогою комунікаційних протоколів, таких як TCP/IP, CAN, MQTT або DDS, що дозволяють забезпечити обмін даними між

компонентами роботизованої системи, навіть якщо вони працюють на різних обчислювальних пристроях.

Оскільки більшість роботизованих систем працюють в реальному часі, дуже важливим є **тимчасове управління**. Це означає, що програма повинна мати можливість точно керувати часом для виконання завдань, уникати затримок у критичних процесах і забезпечувати своєчасну реакцію на зміни оточення. В умовах реального часу навіть незначні затримки можуть призвести до втрати точності або навіть до помилок у виконанні завдання.

Для забезпечення належного управління часом у реальному часі використовуються операційні системи реального часу, які гарантують виконання задач у визначені терміни. Вони забезпечують передбачуваність і точність виконання операцій, що є критичним у роботизованих системах, наприклад, для управління рухом робота або взаємодії з сенсорами, які вимірюють фізичні параметри в режимі реального часу

Загалом, правильна архітектура програмного забезпечення для роботизованих систем забезпечує їхню ефективність, надійність та масштабованість. Модульність, реактивність, розподіленість та управління часом – це основні концепції, які допомагають створювати високоякісні, адаптивні та стабільні програмні рішення для роботів, що працюють в різних умовах.

**Алгоритми керування** роботизованими системами є ключовою частиною розробки програмного забезпечення для роботів, оскільки саме вони визначають, як робот реагуватиме на оточення та виконуватиме завдання. Ці алгоритми поділяються на низькорівневі та високорівневі залежно від складності управління та рівня абстракції.

**Низькорівневе управління** фокусується на прямому контролі фізичних компонентів робота, таких як двигуни та серводвигуни. Це включає програмування руху моторів для забезпечення основної функціональності, наприклад, регулювання швидкості обертання коліс, кута повороту або положення маніпуляторів. Важливою складовою є робота з низькорівневими інтерфейсами, такими як PWM (широтно-імпульсна модуляція), для точного контролю параметрів

двигунів. Наприклад, в автономному роботі ці алгоритми можуть забезпечувати стабільність руху навіть на нерівних поверхнях або при зміні навантаження.

**Високорівневе управління** охоплює більш складні аспекти роботи роботизованих систем, зокрема їхню автономність і здатність до адаптації. Один із ключових елементів високорівневого управління – це навігація, яка включає планування траєкторії, уникнення перешкод і використання карт місцевості. Технології, такі як SLAM (Simultaneous Localization and Mapping), дозволяють роботам одночасно створювати карту оточення та визначати своє місцезнаходження на цій карті. Використання таких алгоритмів робить можливим автономне пересування роботів у невідомому середовищі.

Іншою важливою складовою є системи ухилення від перешкод, які інтегрують дані від сенсорів, таких як ультразвукові, інфрачервоні або лазерні далекоміри (LiDAR). Ці системи дозволяють роботам швидко реагувати на динамічні зміни в середовищі, наприклад, обминати об'єкти або зупинятися перед несподіваними перешкодами.

Додатково, високорівневе управління часто включає використання штучного інтелекту, який забезпечує адаптивність і навчання роботів. Нейронні мережі та методи глибокого навчання дозволяють створювати моделі для розпізнавання об'єктів, аналізу поведінки та прийняття рішень у складних сценаріях. Наприклад, нейронна мережа може навчитися розпізнавати обличчя, щоб робот міг взаємодіяти з певними користувачами, або класифікувати об’єкти, щоб маніпулятор правильно захоплював потрібний предмет. Штучний інтелект також використовується для оптимізації маршрутів або прогнозування подій, що дозволяє роботам працювати ефективніше та безпечніше.

Таким чином, алгоритми управління забезпечують роботу роботизованих систем на всіх рівнях – від базового контролю апаратних компонентів до складних моделей автономної поведінки. Взаємодія між низькорівневим і високорівневим управлінням дозволяє створювати інтегровані рішення, які можуть адаптуватися до мінливих умов і виконувати широкий спектр завдань.

## 2.2 Архітектури та платформи для робототехніки

Архітектури та платформи для робототехніки є основою для розробки і інтеграції різноманітних роботизованих систем. Основна мета архітектури робота – забезпечити ефективну взаємодію між його компонентами, такими як датчики, приводи, процесори та програмне забезпечення. Вибір архітектури залежить від типу робота, його призначення та вимог до продуктивності, надійності та масштабованості. Існує кілька основних типів архітектур, зокрема монолітні, модульні та гібридні, кожна з яких має свої переваги та недоліки в контексті робототехніки.

Монолітні архітектури передбачають інтеграцію всіх функцій і компонентів робота в єдину систему. Це може бути корисно для простих роботів з обмеженими завданнями, але згодом виникають труднощі при масштабуванні або адаптації системи до нових вимог. Модульні архітектури, з іншого боку, дозволяють розділити робота на окремі блоки, кожен з яких відповідає за конкретну функцію, таку як обробка даних, керування рухом або сприйняття навколишнього середовища. Це значно полегшує розробку, тестування і оновлення системи, а також дозволяє змінювати або додавати нові модулі без впливу на інші частини системи.

Гібридні архітектури поєднують елементи обох попередніх підходів, забезпечуючи як високу адаптивність і модульність, так і централізовану координацію між різними частинами робота. Це може бути корисно для складних роботизованих систем, які виконують багатофункціональні завдання.

У освітній робототехніці використовуються різні платформи, які забезпечують доступність для учнів та студентів, простоту у навчанні, а також можливість розвивати практичні навички програмування і роботи з роботами. Найбільш популярними платформами для навчання є: LEGO Mindstorms, VEX Robotics, Arduino, Raspberry Pi, Thymio та micro:bit.

LEGO Mindstorms є однією з найпопулярніших платформ для навчання робототехніці завдяки своїй універсальності та доступності (рис. 2.12). Ця платформа дозволяє створювати роботів за допомогою стандартних LEGO-блоків, що робить її ідеальною для учнів будь-якого віку, починаючи від початкової школи до старших класів. Однією з основних переваг LEGO Mindstorms є те, що вона поєднує в собі як механічні, так і програмні компоненти, дозволяючи учням не лише збирати роботів, а й програмувати їх для виконання різних завдань. Програмування роботів на Mindstorms здійснюється через візуальне середовище, що значно полегшує процес навчання для новачків, оскільки учні можуть створювати алгоритми, перетягуючи блоки коду, а не вводячи їх вручну. Це дозволяє зосередитися на логіці програмування без необхідності глибокого знання текстових мов програмування [6].

**LEGO Mindstorms**



Рисунок 2.12 – Роботи LEGO Mindstorms

Іншою перевагою є велика кількість додаткових компонентів і аксесуарів, таких як мотори, датчики, сенсори та інші, що дозволяє створювати складніші роботи і виконувати різноманітні завдання. LEGO Mindstorms також підтримує інтеграцію з іншими платформами, що дає можливість розширювати можливості проектів. Крім того, наявність великої спільноти користувачів і безлічі навчальних матеріалів дозволяє учням і викладачам швидко знайти необхідну допомогу та ресурси.

Однак, LEGO Mindstorms має і деякі недоліки. Один з головних – це висока вартість. Платформи LEGO Mindstorms можуть бути досить дорогими для шкіл, що

обмежує їх доступність для широкого кола учнів. Крім того, хоча візуальне програмування є дуже зручним для початківців, з часом воно може обмежувати учнів у розвитку навичок програмування на більш складних мовах, таких як Python чи C++. Це може створити певні труднощі для тих, хто хоче перейти до розробки більш складних проектів після початкового знайомства з робототехнікою.

**VEX Robotics** – це потужна платформа для навчання робототехніці, яка пропонує великий вибір наборів для створення роботів різних типів і складності (рис. 2.13). Платформа підтримує програмування як у візуальному середовищі Blockly, що підходить для початківців, так і на більш складних мовах програмування, таких як C++ та Python, що дозволяє студентам і учням поступово розвивати свої навички від основ до більш глибоких знань [2].



Рисунок 2.13 – Роботи VEX Robotics

Серед переваг VEX Robotics – це висока якість і довговічність компонентів, що дозволяє використовувати їх для створення робочих прототипів у школах і університетах, а також в умовах навчальних проектів і змагань. Крім того, наявність великої спільноти користувачів і ресурсів для навчання сприяє ефективному освоєнню робототехніки.

Проте, є й певні недоліки. По-перше, вартість комплектів VEX Robotics може бути досить високою, що обмежує доступність цієї платформи для деяких навчальних закладів, особливо в порівнянні з іншими платформами, такими як Arduino. По-друге, хоча VEX Robotics пропонує потужні можливості для створення складних роботів, для новачків може знадобитися більше часу для осво-

єння більш технічних аспектів, таких як програмування на C++ чи Python, що може бути складним для учнів без попереднього досвіду в програмуванні.

**Arduino** – це одна з найпопулярніших і доступних платформ для навчання робототехніці, електроніці та програмуванню (рис. 2.14). Вона складається з мікроконтролера, який можна програмувати для управління різними електронними компонентами, такими як датчики, мотори, світлодіоди, дисплеї тощо. Arduino є відкритою платформою з великою кількістю доступних бібліотек, що дозволяє легко інтегрувати різні пристрої й компоненти.



Рисунок 2.14 – Роботи Arduino

Платформа має простий інтерфейс для програмування, який підтримує мову C++, з додатковими можливостями для використання інших мов, таких як Python або JavaScript. Це дозволяє учням і початківцям легко освоїти основи програмування, а також дає змогу працювати з реальними, фізичними об'єктами, що значно підвищує цікавість до навчання [12].

Однією з головних переваг Arduino є її низька вартість, що робить її доступною для широкого кола користувачів, від учнів до професіоналів. Крім того, платформа має велику кількість навчальних матеріалів, посібників і форумів, що дозволяє легко знайти допомогу при вирішенні різних задач. Величезною перевагою є також велика спільнота користувачів, яка постійно ділиться своїми проектами, ідеями та рішеннями. Це створює не лише можливості для навчання, але й для реалізації реальних проектів, від простих роботів до складних автоматизованих систем.

Іншою перевагою є висока гнучкість Arduino. Завдяки підтримці численних модулів і сенсорів, таких як ультразвукові датчики, датчики температури, руху і навіть камери, користувачі можуть створювати проекти на основі своїх інтересів і вимог. Платформа також дозволяє працювати з різними типами пристроїв, починаючи від малих роботів і закінчуючи промисловими автоматизованими системами.

Проте, є й деякий недолік. Хоча платформа підтримує багато компонентів, для роботи з ними іноді потрібно знати основи електроніки, а також мати навички пайки та монтажу, що може бути не завжди зручним для початківців.

**Raspberry Pi** – це малий одноплатний комп'ютер, який став дуже популярним у робототехніці, освіті та проектуванні завдяки своїй потужності, низькій вартості та універсальності (рис. 2.15). Raspberry Pi дозволяє створювати роботизовані системи з використанням повноцінної операційної системи, зазвичай Linux, що дає можливість програмувати на широкому спектрі мов, таких як Python, C++, Java та інших. Ця платформа особливо корисна для навчання програмуванню, робототехніці, а також для розробки складних проектів, таких як автономні роботи, роботи для обробки зображень або системи розпізнавання об'єктів [10].

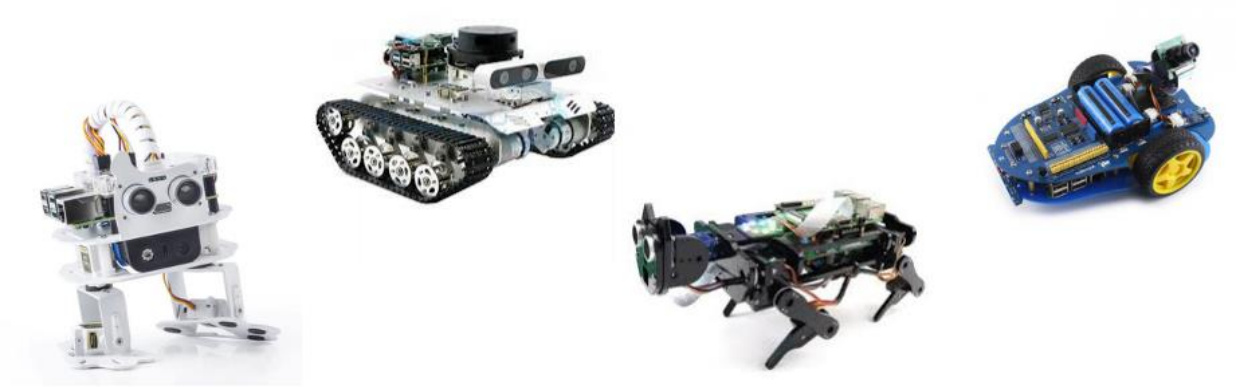

Рисунок 2.15 – Роботи Raspberry Pi

Одна з основних переваг Raspberry Pi – це її потужність і можливості, що виходять за межі простих мікроконтролерів, як, наприклад, Arduino. Raspberry Pi має достатньо ресурсів для запуску повноцінних операційних систем і виконання більш

складних обчислювальних завдань. Вона підтримує підключення до Інтернету, працює з камерами та іншими периферійними пристроями, що робить її відмінним вибором для створення роботів, які потребують обробки великих обсягів даних або інтеграції з мережею. Крім того, Raspberry Pi має велику кількість додаткових модулів і аксесуарів, таких як сенсори, камери, датчики температури та інші, що дозволяє створювати роботів для різних застосувань.

Ще однією перевагою є підтримка великої кількості програмних інструментів і бібліотек для роботи з робототехнічними проєктами. Програмне забезпечення для Raspberry Pi включає в себе такі інструменти, як OpenCV для комп'ютерного зору, TensorFlow для машинного навчання та ROS для створення роботизованих систем, що дає змогу реалізовувати складні проекти в навчанні та дослідженнях.

Проте, є й кілька недоліків. По-перше, Raspberry Pi може бути складнішим у використанні для початківців у порівнянні з платформами, такими як Arduino, оскільки для роботи з нею потрібно мати певні знання в області програмування, операційних систем та налаштування периферії. Крім того, хоча Raspberry Pi і має потужний процесор, вона не завжди підходить для дуже вимогливих додатків, таких як обробка складних графічних чи відео даних у реальному часі. У зв'язку з цим, для деяких проектів можуть знадобитися додаткові потужності або використання іншого апаратного забезпечення.

Ще одним недоліком є обмеження за ресурсами: хоча Raspberry Pi є дуже потужною для своєї ціни, її оперативної пам'яті та процесорної потужності недостатньо для деяких складних завдань без додаткових апаратних ресурсів. Вона також вимагає підключення до джерела живлення і часто використовується з монітором або клавіатурою для налаштування, що не завжди зручно в робототехнічних проектах, де автономність є важливою.

**micro:bit** – це компактна програмована плата, розроблена для навчання дітей та початківців основам програмування і робототехніки (рис. 2.16). Вона оснащена різноманітними вбудованими сенсорами, такими як акселерометр, компас, кнопки для вводу, а також світлодіодним екраном для відображення інформації. micro:bit підтримує просте програмування за допомогою візуальних

блоків, що робить її ідеальною для початкового рівня, а також має можливість програмування за допомогою текстових мов, таких як JavaScript або Python, що дозволяє переходити до більш складних проектів [19].

Однією з основних переваг micro:bit є її доступність і низька вартість, що робить її дуже популярною в освітніх установах, особливо для навчання дітей і молодших школярів. Плата має достатньо функцій, щоб реалізовувати широкий спектр проектів, таких як прості роботи, інтерактивні ігри, датчики температури і вологості, а також інші базові робототехнічні рішення. Крім того, micro:bit підтримує бездротовий зв'язок через Bluetooth, що дозволяє створювати більш складні інтерактивні проекти, включаючи роботів, що можуть взаємодіяти з іншими пристроями.



Рисунок 2.16 – Роботи micro:bit

Ще однією перевагою є величезна спільнота користувачів та наявність безлічі онлайн-ресурсів, навчальних посібників і прикладів проектів, що робить платформу дуже зручною для навчання. Платформа також підтримує інтеграцію з іншими популярними інструментами, такими як LEGO, що дає можливість розширити функціональність роботів і створювати ще більш складні проекти.

Проте є і деякі недоліки. По-перше, хоча micro:bit має багато вбудованих функцій, її апаратні можливості обмежені порівняно з більш потужними платформами, такими як Arduino або Raspberry Pi. Вона не підходить для дуже складних або ресурсомістких проектів, таких як розпізнавання образів або обробка великих обсягів даних. Крім того, для виконання більш просунутих завдань може бути необхідна додаткова апаратура, що додає складності.

Також, хоча micro:bit дозволяє програмувати як візуальними блоками, так і текстовими мовами, для деяких користувачів перехід від блочного до текстового програмування може бути не таким інтуїтивно зрозумілим, особливо для дітей і початківців без попереднього досвіду програмування.

Ці платформи підтримують широкий спектр освітніх програм і дозволяють учням і студентам реалізовувати практичні проекти в галузі робототехніки. Кожна з платформ має свої особливості і підходить для різних вікових груп та рівнів підготовки, що робить їх ідеальними для використання в освітньому процесі.

## 2.3 Аналіз мов та середовищ для програмування, що використовуються для керування роботизованими системами

Мови програмування, що використовуються для керування роботизованими системами, відіграють ключову роль у розробці алгоритмів, що забезпечують ефективне функціонування роботів. Вибір мови залежить від багатьох факторів, таких як складність проекту, вимоги до швидкодії, необхідність взаємодії з апаратним забезпеченням, а також рівень кваліфікації розробників. Ось кілька основних мов програмування, які використовуються в робототехніці, їхні особливості, переваги та недоліки [13].

**C/C++** є одними з найпоширеніших мов програмування для робототехніки завдяки їхній потужності, гнучкості та здатності ефективно працювати з апаратними ресурсами робота. Вони дають розробникам великий контроль над процесами, які відбуваються на низькому рівні, що важливо для розробки систем, що працюють в реальному часі, таких як управління рухом роботів, обробка сигналів від датчиків і контроль приводів. C/C++ активно використовуються в більшості сучасних робототехнічних платформ і є основою для багатьох програмних пакетів, що забезпечує взаємодію між різними компонентами роботизованих систем.

Для учнів старших класів вивчення C/C++ відкриває можливість глибше розуміти принципи роботи комп'ютерних систем, управління апаратним забезпеченням і оптимізацію програм для досягнення максимальної ефективності.

Однією з переваг C/C++ є можливість безпосередньо працювати з пам'яттю, що дозволяє розробникам оптимізувати код для максимального використання обмежених ресурсів, таких як пам'ять і процесорний час. Ця мова також дозволяє використовувати численні бібліотеки, які полегшують розробку складних алгоритмів і взаємодію з різними типами датчиків і приводів, що є важливими в робототехніці. C++ дає можливість використовувати об'єктно-орієнтоване програмування, що дозволяє створювати більш модульні і масштабовані програми.

Для старшокласників, які вже мають певні базові навички програмування, C/C++ стане хорошим кроком до вивчення більш складних концепцій програмування, таких як вказівники, управління пам'яттю та багатозадачність. Це дозволяє створювати не тільки прості програми, а й складніші алгоритми для обробки великих обсягів даних або складних обчислень.

Однак, C/C++ може бути складною мовою для початківців, особливо для учнів, які не мають досвіду в програмуванні. Потрібно ретельно працювати з пам'яттю, оскільки неправильне управління нею може призвести до помилок, таких як витоки пам'яті або переповнення буфера. Крім того, хоча C/C++ пропонують велику гнучкість, вони не є настільки зручними для швидкого розроблення програм, як деякі високорівневі мови, такі як Python. Це означає, що для розробки роботизованих систем можуть знадобитися додаткові ресурси і більше часу для налаштування і тестування коду. Водночас, володіння C/C++ значно підвищує кваліфікацію учнів і дає їм міцну основу для вивчення інших мов програмування та розуміння більш складних концепцій комп'ютерних наук, що стане в нагоді при подальшому навчанні або в професійній діяльності в галузі програмування, робототехніки і автоматизації. C/C++ також активно використовуються в індустрії для розробки програмного забезпечення для автономних систем, автомобілів, дронів, а також для вбудованих систем, що дозволяє учням зрозуміти реальні застосування робототехніки в сучасному світі.

**Python** є однією з найпопулярніших мов програмування, що активно використовується в багатьох сферах, включаючи робототехніку, завдяки своїй простоті, читабельності та широкій підтримці бібліотек. Для учнів старших класів Python є чудовим вибором, оскільки він дозволяє швидко навчитися основам програмування і створювати робочі програми без необхідності глибоко занурюватися в складні концепції. Його синтаксис простий і зрозумілий, що значно полегшує процес навчання. Python дозволяє зосередитися на логіці програмування, а не на складностях синтаксису, що робить його доступним навіть для початківців.

У робототехніці Python використовується для написання програм, які управляють роботами, обробляють дані з сенсорів, виконують алгоритми штучного інтелекту і комп'ютерного зору. Зокрема, Python часто використовують для створення складних систем, що вимагають обробки великих обсягів інформації, таких як автономні роботи або роботи з інтегрованими камерами і сенсорами.

Однією з основних переваг Python є величезна кількість бібліотек, які дозволяють реалізувати різноманітні функції без необхідності писати складний код з нуля. Наприклад, бібліотеки OpenCV для комп'ютерного зору, TensorFlow для машинного навчання або Pyaudio для роботи зі звуком дозволяють учням працювати з сучасними технологіями та досліджувати цікаві аспекти робототехніки, що значно спрощує розробку проектів.

Для старшокласників Python є також чудовим інструментом для розуміння основ штучного інтелекту, оскільки мова широко використовується для створення нейронних мереж, алгоритмів машинного навчання та глибокого навчання. Вивчення Python дозволяє учням розвивати навички, які будуть корисні в майбутньому для роботи з великими даними та програмування інтелектуальних систем. Крім того, Python має підтримку для багатьох роботизованих платформ, таких як Raspberry Pi, що робить його ідеальним для створення простих проектів, наприклад, роботів, які використовують сенсори для взаємодії з навколишнім середовищем або автономно виконують завдання.

Одним із значних недоліків Python є те, що він не є так ефективним у реальному часі, як мови низького рівня, такі як C або C++, через те, що Python є

інтерпретованою мовою. Це означає, що виконання програм може бути повільнішим, що інколи є обмеженням для систем, які потребують високої продуктивності та миттєвої реакції. Однак для більшості освітніх проектів, включаючи робототехнічні, Python забезпечує достатню швидкість і продуктивність. Незважаючи на це, Python є ідеальним вибором для тих, хто хоче швидко втілювати ідеї в робототехніці, вивчаючи при цьому основи програмування та складніші концепції штучного інтелекту і обробки даних. Його простота, численні ресурси для навчання та багатий вибір бібліотек роблять Python чудовою мовою для старшокласників, які прагнуть освоїти робототехніку та інші сучасні технології.

**MATLAB/Simulink** – це програмне середовище, яке широко використовується для математичних розрахунків, моделювання та симуляції складних систем, зокрема в робототехніці. MATLAB є мовою програмування, орієнтованою на обчислювальні задачі, і дозволяє зручно працювати з матрицями, векторами, числами з плаваючою комою, а також виконувати чисельні та символічні обчислення. Simulink, у свою чергу, є візуальним інструментом для моделювання та симуляції різноманітних динамічних систем, включаючи роботизовані, де можна створювати модель роботизованої системи, не написавши жодного рядка коду.

Для учнів старших класів MATLAB/Simulink є чудовим вибором, оскільки дозволяє зрозуміти теоретичні аспекти робототехніки, одночасно даючи можливість втілювати ці знання на практиці. MATLAB і Simulink дозволяють створювати математичні моделі руху роботів, планування траєкторій, управління і обробку сигналів від датчиків, а також інтеграцію роботів у складні середовища.

У контексті робототехніки, MATLAB забезпечує доступ до потужних інструментів для аналізу, оптимізації та тестування алгоритмів управління, таких як моделі зворотного зв’язку, методи оптимізації, алгоритми для контролю руху та багато іншого. Simulink доповнює MATLAB, даючи можливість створювати графічні моделі систем, використовуючи блоки для різних елементів роботизованої системи, таких як мотори, датчики та алгоритми обробки сигналів, що полегшує проектування і тестування.

Для учнів старших класів важливою перевагою є наявність інтегрованих функцій для моделювання, які допомагають розуміти складні системи управління та алгоритми, не необхідно занурюватися в деталі програмування низького рівня. MATLAB/Simulink дозволяють симулювати поведінку роботів у різних умовах до того, як їх програмне забезпечення буде впроваджено в реальні апаратні системи, що значно знижує ризик помилок і дає можливість протестувати безліч варіантів рішень. Крім того, MATLAB широко використовується в індустрії, особливо для розробки роботизованих систем у таких галузях, як аерокосмічна, автомобільна та медична. Це дозволяє учням старших класів не тільки отримати базові знання з робототехніки, але й ознайомитися з інструментами, які використовуються професіоналами.

Проте є й недоліки. MATLAB і Simulink можуть бути дорогими для навчальних закладів, оскільки вони є комерційними програмами, що може обмежити їх доступність. Крім того, хоча ці інструменти дуже потужні, для їх ефективного використання потрібно знання математичних і фізичних принципів, що може бути складним для учнів без попереднього досвіду в цих областях. Тим не менш, MATLAB/Simulink є дуже корисним для старшокласників, які хочуть поглибити свої знання з робототехніки, оскільки ця платформа забезпечує потужні інструменти для моделювання, тестування та розробки алгоритмів управління роботами.

**Java** є однією з найпопулярніших і широко використовуваних мов програмування, яка також має потужне застосування в робототехніці. Вона є об'єктно-орієнтованою мовою, що дозволяє створювати ефективні, масштабовані та модульні програми для різних систем.

Для учнів старших класів Java є чудовим вибором, оскільки вона дозволяє розвивати важливі навички програмування, такі як проектування програм, робота з об'єктами, а також розуміння важливості абстракції і інкапсуляції. У робототехніці Java активно використовується для розробки програм для роботів, автоматизації процесів, а також для створення інтерфейсів і систем, які потребують обробки великої кількості даних або взаємодії з іншими програмами.

Важливою перевагою Java є її платформонезалежність, що означає, що програми, написані на Java, можуть працювати на різних операційних системах без необхідності змінювати код. Це дуже корисно в робототехніці, де зазвичай необхідна інтеграція різних компонентів, які можуть працювати на різних платформах. Крім того, Java має хорошу підтримку багатозадачності та потоків, що дозволяє ефективно розподіляти ресурси при роботі з багатьма завданнями одночасно, що є важливим у робототехнічних системах, де роботу робота потрібно координувати з іншими процесами, такими як обробка даних від сенсорів, управління рухом і взаємодія з іншими пристроями. Java також підтримує інтеграцію з іншими бібліотеками і платформами для робототехніки, такими як LEGO Mindstorms, VEX Robotics або ROS, що дає можливість програмувати роботи з використанням Java.

Для старшокласників, які мають базові знання в програмуванні, Java є ідеальним інструментом для розвитку логічного мислення, а також навичок розв'язання складних задач. Вона дає учням можливість працювати з реальними проектами і розвивати професійні навички, які знадобляться при подальшому навчанні або в кар'єрі.

Однак, для деяких учнів Java може здаватися менш доступною, оскільки її синтаксис є більш складним порівняно з іншими мовами програмування, такими як Python чи Scratch. Зокрема, вона вимагає чіткої структури коду і більшого обсягу коду для вирішення деяких завдань, що може бути складним для початківців. Крім того, хоча Java підтримує високу продуктивність і масштабованість, її час компіляції та виконання може бути дещо повільнішим у порівнянні з мовами низького рівня, такими як C або C++. Водночас, Java є потужною і універсальною мовою програмування, яка дозволяє старшокласникам створювати складні і ефективні роботизовані системи та розвивати важливі навички для майбутньої кар'єри в галузі програмування і робототехніки.

**LabVIEW** (Laboratory Virtual Instrument Engineering Workbench) – це графічне програмне середовище, яке активно використовується для розробки програмних застосунків у таких галузях, як автоматизація, вимірювання та контро-

лю, а також робототехніка. LabVIEW відрізняється від традиційних мов програмування тим, що використовує візуальне програмування, де замість написання коду програміст створює програми, використовуючи графічні блоки, що представляють різні функції та операції. Ця особливість робить LabVIEW особливо підходящим для учнів старших класів, які тільки починають вивчати програмування або мають обмежений досвід роботи з традиційними текстовими мовами. У LabVIEW всі програми побудовані як графічні схеми, де користувач перетягує і з'єднує блоки, що відповідають за різні функції, такі як вимірювання сигналів, обробка даних, управління пристроями та взаємодія з іншими компонентами. Такий підхід значно спрощує вивчення принципів програмування та дозволяє зосередитися на логіці роботи системи, а не на складностях текстового коду [8].

У робототехніці LabVIEW використовують для створення інтерфейсів управління, розробки алгоритмів для контролю руху роботів, обробки сигналів від датчиків і управління актуаторами. Завдяки інтеграції з апаратними платформами, такими як NI Robotics та myRIO, LabVIEW дозволяє створювати повноцінні роботизовані системи, що виконують складні завдання, включаючи автономне пересування, обробку сенсорних даних і взаємодію з оточенням. Крім того, LabVIEW підтримує зв'язок з іншими технологіями, такими як MATLAB або Simulink, що дозволяє комбінувати переваги обох середовищ для більш складних наукових і інженерних розрахунків.

Однією з головних переваг LabVIEW є його здатність швидко і наочно створювати прототипи складних систем, що дозволяє учням зрозуміти принципи роботи роботизованих технологій, не заглиблюючись у складний код. Це дозволяє фокусуватися на практичному застосуванні і тестуванні ідей в реальному часі.

Для старшокласників LabVIEW стане чудовим інструментом для освоєння основ робототехніки, автоматизації та наукових досліджень, оскільки воно дозволяє вивчати принципи керування та обробки даних без необхідності глибоких знань у програмуванні.

Однак, LabVIEW має і свої недоліки. Оскільки це комерційний продукт, для отримання доступу до повної версії може знадобитися ліцензія, що може бути до-

рогим для шкіл або учнів, що займаються самостійним навчанням. Крім того, LabVIEW не є настільки гнучким, як текстові мови програмування, і може бути обмеженим при розробці складних програм або алгоритмів, які вимагають більш детального налаштування. Однак для учнів старших класів, які хочуть створювати прототипи роботизованих систем і працювати з реальними фізичними пристроями, LabVIEW є дуже корисним і доступним інструментом, який поєднує в собі простоту вивчення і потужність для створення робототехнічних рішень.

Вибір мови програмування для робототехніки значною мірою залежить від специфіки проекту: від простих освітніх роботів до складних автономних систем, від інтеграції з іншими програмами до обробки даних у реальному часі. Кожна з мов має свої переваги і обмеження, тому важливо вибирати ту, яка найкраще підходить для конкретних завдань, з урахуванням ефективності, гнучкості та простоти використання.

## 2.4 Використання Arduino та C++ для розвитку практичних навичок у робототехніці серед старшокласників

Вибір платформи і мови програмування для застосування у навчанні учнів старших класів робототехніці зупинився на платформі Arduino та мові C++.

Arduino – це доступна і популярна платформа, що дозволяє створювати різноманітні роботизовані та електронні проекти. Вона оснащена мікроконтролером, який легко програмується, має просту конструкцію та підтримує безліч додаткових модулів і сенсорів. Це дозволяє учням реалізувати проекти від простих до складних, розвиваючи практичні навички в робототехніці, електроніці та програмуванні.

Використання C++ на платформі Arduino дозволяє учням ознайомитися з потужною і універсальною мовою програмування, яка широко використовується в індустрії для розробки програмного забезпечення для вбудованих систем, роботів і автоматизації. C++ дає змогу працювати з апаратним забезпеченням

на низькому рівні, що дозволяє краще зрозуміти, як працюють мікроконтролери та інші електронні компоненти, що є основою для розвитку навичок системного мислення та проблемно-орієнтованого підходу. Ця мова дає високий рівень контролю за пам'яттю та ресурсами, що важливо для ефективної роботи в умовах обмежених апаратних ресурсів, які характерні для робототехнічних проектів. Вивчення C++ дозволяє учням зрозуміти основи об'єктно-орієнтованого програмування, що є важливим для подальшого розвитку в програмуванні. Крім того, Arduino має велику спільноту користувачів, що дозволяє учням швидко знаходити допомогу, ресурси, навчальні посібники та приклади проектів, що полегшує процес навчання. Завдяки великій кількості доступних бібліотек та готових рішень, учні можуть зосередитися на логіці програмування та реалізації своїх ідей, а не на написанні низькорівневого коду. Arduino також дозволяє інтегрувати різноманітні датчики, мотори, дисплеї та інші компоненти, що відкриває широкі можливості для практичних проектів, таких як автоматизація, робототехніка, Інтернет речей тощо. Платформа має низьку вартість, що робить її доступною для шкіл та учнів. Вивчення C++ на платформі Arduino також дозволяє учням перейти до більш складних програмних середовищ та проектів, оскільки ця мова є основою для багатьох інших технологій і програмних рішень у галузі робототехніки та програмування вбудованих систем.

## 2.5 Вибір платформи для програмування Arduino

Програмна складова охоплює потужне середовище розробки, яке дає змогу учням створювати та налагоджувати прошивки без зайвих труднощів. Крім того, вона містить набір попередньо створених і зручних бібліотек та використовує доступну для розуміння мову програмування, що полегшує опанування матеріалу.

Апаратна частина включає широкий асортимент мікроконтролерів та сумісних з ними готових модулів, забезпечуючи педагогів і школярів багатим інструментарієм для створення різноманітних проектів.

Не менш важливим є наявність численних платформ для програмування Arduino. Кожне програмне середовище має свої унікальні особливості та простоту у використанні, що дозволяє враховувати індивідуальні запити користувачів. Обрання оптимальної платформи здатне значно спростити процес навчання й вдосконалення, роблячи його більш результативним і цікавим для учнів.

**Arduino IDE** застосовує спрощений варіант програмної мови C++, доповнений заздалегідь визначеними функціями. Подібно до інших мов програмування сімейства C, існують певні норми щодо написання коду, серед яких ключові включають:

– постановку крапки з комою в кінці кожного оператора;
– зазначення типу даних перед визначенням функції, що вказує, яке значення вона повертає, або використання ’void’, якщо функція не має результату;
– обов’язкове вказування типу даних під час оголошення змінних.

Інтегроване середовище розробки (IDE), створене для Windows, MacOS і Linux, базується на мовах програмування C та C++ і призначене для розробки та завантаження застосунків на плати, сумісні з Arduino, а також на пристрої інших виробників. Arduino IDE містить бібліотеку програмного забезпечення, засновану на проекті Wiring, яка надає широкий набір універсальних функцій для вводу-виводу.

Для написання коду розробником необхідні лише дві основні функції: одна для ініціалізації ескізу, а інша для виконання основного циклу програми. Ці функції компілюються та інтегруються із заготовкою програми main(), після чого код конвертується у текстовий файл у форматі Hex. Завантаження отриманого файлу на плату Arduino здійснюється через завантажувач, що є складовою частиною вбудованого програмного забезпечення плати.

Зі збільшенням популярності Arduino інші розробники почали створювати власні компілятори та інструменти з відкритим вихідним кодом для розробки і завантаження ескізів у мікроконтролери, які не належать до офіційної лінійки Arduino.

**Programino IDE** – це середовище розробки програмного забезпечення для Arduino та сумісних мікроконтролерів. Воно створене для того, щоб полегшити процес написання, компіляції та завантаження коду на Arduino або інші мікроконтролери без використання офіційного Arduino IDE.

Основні особливості Programino: підтримка багатьох платформ (Windows, macOS, Linux), можливість роботи з різними платами (як стандартними Arduino, так і ESP32, ESP8266 тощо), вбудований компілятор, що спрощує робочий процес, вбудований серійний монітор для реального зв'язку між мікроконтролером і комп'ютером, зручне керування бібліотеками безпосередньо в IDE, а також підтримка таких функцій, як підсвітка синтаксису, автодоповнення та зручна навігація між функціями та файлами, що допомагає прискорити програмування. Програмне забезпечення має простий та зрозумілий інтерфейс, що підходить як для початківців, так і для досвідчених користувачів.

Однак у Programino є й недоліки. Це, зокрема, менший розвиток спільноти порівняно з офіційним Arduino IDE, що може ускладнити пошук рішень для специфічних питань. Також підтримка документації є обмеженою, і для деяких користувачів може бути важко знайти достатньо навчальних матеріалів. Крім того, не всі функції, доступні в офіційному Arduino IDE, підтримуються в Programino, що може бути обмеженням для більш складних проектів. Іноді також можуть виникати проблеми з оновленнями, оскільки нові версії програмного забезпечення з'являються рідше, ніж у офіційного IDE, що може призвести до затримок у підтримці нових моделей плат чи функцій. А саме головне, це середовище платне, а значить для навчання у школі нам не підходить.

**CodeBlocks Arduino IDE** – це версія популярного середовища розробки Code::Blocks, адаптована для роботи з платами Arduino. Вона пропонує більш потужний та гнучкий інтерфейс для програмування мікроконтролерів, ніж стандартне Arduino IDE, і надає можливість розробки з підтримкою C++ та інших мов програмування. CodeBlocks Arduino IDE поєднує зручність стандартного Arduino IDE з додатковими функціями, що робить його привабливим для більш досвідчених користувачів. Серед його переваг можна виділити потужний реда-

ктор коду з автозавершенням, підтримку налагодження коду через GDB, що дає можливість краще відстежувати та виправляти помилки, а також інтеграцію з такими додатковими бібліотеками та плагінами, як wxWidgets для створення графічних інтерфейсів користувача. Крім того, CodeBlocks Arduino IDE підтримує всі можливості стандартного Arduino IDE, такі як компіляція та завантаження коду на мікроконтролери безпосередньо з середовища розробки.

Однак, у CodeBlocks Arduino IDE є й певні недоліки. По-перше, ця IDE може бути складною для новачків, оскільки вона має значно більше налаштувань та параметрів порівняно з класичним Arduino IDE. Для деяких користувачів процес налаштування середовища може бути заплутаним і вимагати додаткових зусиль для налаштування компілятора та інтеграції з конкретними платами. Крім того, хоча CodeBlocks підтримує багато функцій, це може бути обмеженням, якщо мова йде про менш популярні або нестандартні плати. У порівнянні з більш простими та специфічно орієнтованими середовищами, такими як Arduino IDE, CodeBlocks Arduino IDE може здатися занадто громіздким для простих проектів, де не потрібно так багато налаштувань і додаткових можливостей. Окрім цього, підтримка спільноти в CodeBlocks Arduino IDE менша, а ресурси для навчання можуть бути обмеженими, що ускладнює пошук вирішення специфічних проблем.

Arduino IDE обираємо з кількох ключових причин, зокрема через його простоту, доступність та функціональність, що робить його ідеальним для початківців та користувачів з різним рівнем досвіду. По-перше, Arduino IDE має дуже простий та зрозумілий інтерфейс, що дозволяє легко зосередитися на програмуванні без зайвих складнощів. Це особливо важливо для новачків, оскільки IDE не перевантажена складними функціями, що робить процес навчання більш зручним. Крім того, Arduino IDE має інтегрований компілятор та можливість безпосереднього завантаження коду на мікроконтролери, що значно спрощує робочий процес. Середовище підтримує широкий спектр плат Arduino, а також деякі інші сумісні пристрої, що дає користувачам величезний вибір у виборі апаратного забезпечення для проектів.

Однією з основних причин, чому обираємо Arduino IDE, є величезна спільнота користувачів та розробників, яка забезпечує доступ до великої кількості бібліотек, прикладів коду, навчальних матеріалів і рішень для різноманітних проблем. Це дозволяє користувачам швидко знаходити допомогу або готові рішення для своїх проектів. Бібліотеки в Arduino IDE постійно оновлюються та додаються нові, що робить середовище дуже гнучким і дозволяє швидко інтегрувати нові функції в свої проекти.

Ще однією перевагою Arduino IDE є його доступність. Це безкоштовне програмне забезпечення з відкритим кодом, що дозволяє будь-кому скачати, встановити і використовувати IDE без жодних фінансових витрат. З часом Arduino IDE також зростала разом з розвитком апаратного забезпечення, що зробило її оптимальним вибором для будь-яких проектів, від простих до складних, завдяки стабільності та постійним оновленням.

Важливою причиною вибору Arduino IDE є також підтримка налагодження і відлагодження коду, можливість використання серійного монітора для взаємодії з пристроєм в реальному часі, що є необхідним для ефективного тестування та відстеження роботи проекту.

# 3 РОЗРОБКА ФАКУЛЬТАТИВНОГО КУРСУ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КЕРУВАННЯ РОБОТИЗОВАНИМИ ПРИСТРОЯМИ ДЛЯ ПРОФІЛЬНОЇ ШКОЛИ

## 3.1 Визначення цілей і завдань курсу

Факультативний курс «Розробка програмного забезпечення для управління роботизованими пристроями на платформі Arduino» розрахований на учнів 10 класу та призначений для формування у них основних технічних і програмних компетентностей у галузі робототехніки та автоматизації. Курс спрямований на розвиток навичок програмування на мові C++, ознайомлення з принципами роботи з мікроконтролерами Arduino та створення програмного забезпечення для керування роботизованими пристроями. Окрім того, курс надає учням можливість застосовувати отримані знання в реальних проектах, розвиваючи їх критичне мислення та творчі здібності, що сприяє підготовці до подальшого навчання та кар'єри в сфері технологій.

**Мета курсу:**

- розширення і поглиблення знань учнів про основи розробки програмного забезпечення для роботизованих систем;
- формування практичних навичок програмування для управління роботами і автоматизованими пристроями;
- створення умов для перевірки можливостей самореалізації учнів у сфері сучасних технологій;
- ознайомлення з процесами проектування, розробки і тестування програмного забезпечення для робототехнічних комплексів.

**Завдання курсу:**

- ознайомити учнів із основами робототехніки, принципами роботи роботизованих пристроїв;

– формувати знання з програмування мікроконтролерів та розробки алгоритмів для керування роботами;
– розвивати практичні навички використання сучасних інструментів для програмування роботів і автоматизованих систем;
– сформувати вміння працювати з основними бібліотеками та платформами Arduino для розробки програмного забезпечення на C++;
– ознайомити учнів з методами тестування і налаштування роботизованих пристроїв.

## 3.2 Структура курсу

**Розділ 1 Вступ. Основи робототехніки та автоматизації**

– ознайомлення з поняттями робототехніки та автоматизації;
– вступ до платформи Arduino та мови програмування C++;
– огляд основних компонентів роботизованих систем (сенсори, мотори, мікроконтролери).

**Розділ 2 Операційна система для платформи Arduino та налаштування середовища розробки**

– встановлення та налаштування середовища Arduino IDE;
– основи програмування на платформі Arduino;
– перші кроки в програмуванні на мові C++ для Arduino;
– визначення простих входів/виходів, робота з цифровими і аналоговими пінами.

**Розділ 3 Основи програмування на мові C++ для мікроконтролерів**

– основи мови C++: змінні, типи даних, оператори, цикли, умови;
– програмування для роботи з простими сенсорами та приводами;
– використання бібліотек Arduino для роботи з датчиками та сервоприводами.

**Розділ 4 Створення та програмування простих роботизованих систем**

– створення базових роботизованих пристроїв (наприклад, мобільних роботів);
– програмування для руху робота за допомогою моторів і датчиків;
– використання датчиків для навігації: ультразвукові датчики для вимірювання відстані.

**Розділ 5 Робота з сенсорами та приводами**

– ознайомлення з різними типами сенсорів (ультразвукові, температурні, світлові);
– програмування роботів для використання сенсорних даних;
– застосування сервоприводів і крокових моторів для створення рухомих частин роботів.

**Розділ 6 Інтерфейси та комунікації: Arduino і інші пристрої**

– основи комунікацій між Arduino та іншими пристроями (зчитування даних, відправка команд);
– використання послідовних портів для обміну даними з ПК або іншими платами;
– ознайомлення з Bluetooth, Wi-Fi та іншими бездротовими технологіями для інтеграції роботизованих систем.

**Розділ 7 Створення проекту роботизованої системи**

– планування та реалізація проекту роботизованої системи;
– виконання практичних завдань по створенню роботів для специфічних цілей (наприклад, робот, що рухається по лабіринту);
– тестування та налагодження роботи створеного проекту.

**Розділ 8 Професійне становлення та можливості для розвитку в галузі робототехніки**

– ознайомлення з професіями в сфері робототехніки, інженерії та програмування;
– роль робототехніки в сучасному світі та ринку праці;
– перспективи розвитку кар’єри в сфері робототехніки та автоматизації.

**Загальні вимоги**

– програма курсу передбачає поєднання теоретичних занять та практичних робіт, виконання лабораторних і проектних завдань;
– в процесі засвоєння матеріалу необхідно використовувати сучасні інформаційно-комунікаційні технології та наочні матеріали;
– учні виконують індивідуальні та групові проекти, що дозволяє глибше зануритися в тематику та розвивати навички командної роботи;
– оцінка успіхів учнів включає аналіз виконаних проектів, рольових завдань і тестування знань.

**Резерв часу.** Учитель має можливість використовувати резервний час для поглибленого вивчення окремих тем, обговорення складних питань, або для виконання додаткових проектів з урахуванням інтересів учнів.

**Очікувані результати курсу.** По завершенню курсу учні повинні мати:

– базові знання про програмування мікроконтролерів на платформі Arduino та використання мови C++ для робототехнічних задач;
– навички створення та програмування простих роботизованих систем;
– досвід виконання індивідуальних та колективних проектів, а також здатність працювати з різними сенсорами та актуаторами.

Тематичний план факультативного курсу «Розробка програмного забезпечення для управління роботизованими пристроями» представлений у таблиці 3.1.

Таблиця 3.1 – Тематичний план 35 годин

| № з/п | Назва | Кількість годин | |
|---|---|---|---|
| | | Всього | З них на ЛПР |
| 1 | 2 | 3 | 4 |
| 1 | Розділ 1 Вступ. Основи робототехніки та автоматизації | 1 | |
| 2 | Розділ 2 Операційна система для платформи Arduino та налаштування середовища розробки | 4 | 3 |
| 3 | Розділ 3 Основи програмування на мові C++ для мікроконтролерів | 5 | 3 |

Продовження ьаблиці 3.1

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| 4 | Розділ 4 Створення та програмування простих роботизованих систем | 5 | 3 |
| 5 | Розділ 5 Робота з сенсорами та приводами | 5 | 3 |
| 6 | Розділ 6 Інтерфейси та комунікації: Arduino і інші пристрої | 5 | 3 |
| 7 | Розділ 7 Створення проекту роботизованої системи | 5 | 3 |
| 8 | Розділ 8 Професійне становлення та можливості для розвитку в галузі робототехніки | 1 | |
| 9 | Підсумковий проект | 3 | |
| 10 | Резерв часу | 1 | |
| | Разом | 35 | 18 |

### 3.3 Розробка навчальних матеріалів (конспекти лекцій, презентації, завдання)

**План-конспект уроку 1**

**Тема уроку:** *Основи програмування на платформі Arduino для створення роботизованих систем.*

**Мета уроку:** *навчальна* – ознайомлення з основами програмування на платформі Arduino, навчитися створювати базові програми для керування роботизованими пристроями; *розвивальна* – розвивання логічне мислення учнів, їх здатність до програмування та розв'язання технічних задач; *виховна* – виховувати інтерес до робототехніки, технічної творчості та інженерії.

**Тип уроку**: ознайомлення з новим матеріалом, засвоєння основних знань і навичок програмування на платформі Arduino.

**Хід уроку**

*І Організаційні моменти заняття*

**1 Привітання.** Доброго дня, учні! Сьогодні ми розпочинаємо знайомство з основами робототехніки та програмування на платформі Arduino. Для цього на

першому етапі ми розглянемо базові поняття робототехніки та автоматизації, а також ознайомимося з компонентами, що використовуються для створення роботів.

**2 Ознайомлення з планом уроку:**

1. Що таке робототехніка та автоматизація?
2. Як працює платформа Arduino?
3. Які компоненти необхідні для створення роботизованих систем?

*ІІ Актуалізація знань, умінь та навичок*

**1 Обговорення попереднього досвіду:**

*Запитання до учнів*. Що вам відомо про робототехніку та автоматизацію? Які пристрої ви знаєте, що можуть працювати без участі людини (роботи, автоматизовані системи)?

Визначення загальних понять, обговорення поширених прикладів роботів у повсякденному житті (наприклад, роботи-пилососи, промислові роботи).

*ІІІ Вивчення нового навчального матеріалу*

**1 Ознайомлення з поняттям робототехніки та автоматизації:**

Робототехніка – це галузь науки і техніки, яка займається розробкою та використанням роботів. Роботи можуть виконувати різні завдання, які є важкими або небезпечними для людей.

Автоматизація – це процес заміни людської праці технічними засобами, які здатні виконувати завдання автоматично, без безпосереднього втручання людини.

Приклади роботів та автоматизованих систем у сучасному світі:

– Промислові роботи на заводах.

– Системи автоматизації в сільському господарстві.

– Розумні будинки, де пристрої автоматично реагують на зміни у середовищі (температура, освітленість тощо).

**2 Вступ до платформи Arduino.**

Arduino – це відкрита апаратно-програмна платформа, яка дозволяє створювати інтерактивні проекти, підключати різні датчики, моторчики та інші компоненти. Платформа включає мікроконтролер, який можна програмувати для виконання завдань.

Arduino дає можливість працювати з реальними фізичними пристроями, такими як світлодіоди, серводвигуни, датчики, що робить її ідеальною для початку роботи в робототехніці.

Мова програмування для Arduino. Платформа використовує мову програмування, схожу на C++, що дозволяє контролювати поведінку роботизованих систем.

**3 Огляд основних компонентів роботизованих систем.**

Мікроконтролери – центральний компонент, який виконує програми та керує іншими пристроями в роботизованих системах. Наприклад, плата Arduino є одним із найпопулярніших мікроконтролерів для створення роботів.

Сенсори – пристрої, що дозволяють роботу отримувати інформацію з навколишнього середовища. Наприклад, ультразвукові сенсори вимірюють відстань до перешкод, температурні датчики вимірюють температуру.

Мотори та серводвигуни – використовуються для руху роботів або для виконання механічних операцій. Мотори можуть рухати колеса роботів, а серводвигуни дозволяють точно контролювати позицію окремих частин робота.

Актуатори – механізми, що виконують фізичні дії, такі як рухи чи зміна положення, підключені до мікроконтролера.

Плата Arduino – це основний елемент, який з'єднується з іншими компонентами, і на основі програм, написаних за допомогою Arduino IDE, керує всіма пристроями.

**4 Робота з компонентами.**

Огляд основних пінів та портів на платі Arduino:

– цифрові піні для підключення компонентів, таких як кнопки, світло діоди;

– аналогові піні для підключення датчиків, які вимірюють величини, що змінюються (наприклад, температура чи вологість);

– додаткові компоненти, що можуть бути використані для побудови роботизованих систем.

*IV Підсумок уроку*

**1 Узагальнення навчального матеріалу.**

– Що таке робототехніка та автоматизація?

– Які основні компоненти потрібні для створення роботизованих систем?

– Як Arduino використовується для програмування та керування роботами?

**2 Закріплення знань:**

– обговорення важливості сенсорів, моторів і мікроконтролерів для роботи роботизованих систем.

*V Повідомлення домашнього завдання*

Підготуватися до наступного уроку, де ми будемо встановлювати Arduino IDE на комп'ютери та готуватися до практичних вправ з програмування, а також передивіться відео про основи робототехніки на YouTube:

https://youtube.com/playlist?list=PLWuPdIzAqBUhAhy855qnmX0E2wLJl7zNR&si=RnIdxsakpzgF9KUE

*Завершення уроку*

Подякувати учням за активну участь. Нагадати про важливість роботи з реальними пристроями та інструментами для освоєння робототехніки та програмування.

Конспекти до всіх інших лекцій розроблені за аналогічною структурою і представлені у Додатку А.

# ВИСНОВКИ

Розробка факультативного курсу з робототехніки для старших класів профільних шкіл є актуальною та необхідною в контексті сучасного розвитку технологій та потреб ринку праці. Це дослідження показує важливість інтеграції робототехніки в шкільну освіту для підготовки молоді до майбутніх професій, що пов'язані з автоматизацією, інженерією та програмуванням. В Україні існує значний дефіцит освітніх програм для старшокласників у цій галузі, що потребує розробки нових курсів, орієнтованих на поглиблену підготовку учнів в цій сфері.

Практичне значення дослідження полягає в розробці програми факультативного курсу, що охоплює як теоретичні, так і практичні аспекти робототехніки. Це дозволить учням отримати базові знання з програмування мікроконтролерів, створення роботизованих систем та використання сучасних технологій, таких як Arduino та C++, що відповідатиме вимогам сучасного ринку праці. Крім того, курс сприятиме розвитку критичного мислення, творчих здібностей та інженерних навичок у школярів, що є важливою складовою STEM-освіти.

Наукове значення роботи полягає в розробці теоретичних засад для інтеграції робототехніки в середню освіту та створенні нової моделі навчання для профільних шкіл, що включає рекомендації для ефективного впровадження цієї дисципліни. Результати дослідження можуть бути використані в освітніх установах для створення та адаптації програм з робототехніки в школах, що дозволить не тільки підготувати учнів до нових викликів технологічного розвитку, але й покращити загальний рівень технічної освіти в Україні.

Загалом, факультативний курс з робототехніки має потенціал не лише покращити навчальний процес в школах, а й сприяти підготовці кваліфікованих кадрів для національної економіки, що активно потребує фахівців у сферах, пов'язаних з роботизацією та автоматизацією.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Cedos – незалежний центр досліджень. URL : https://cedos.org.ua/researches/osvita-i-vijna-v-ukrayini-24-lyutogo-1-kvitnya-2022/ (дата звернення:10.11.2024).

2. Morze, N., Ghladun, M.. & Dziuba, S. Formation of key and subject competences of students by robotic means of STEM-education. *Informacijni tekhnologhiji i zasoby navchannja*. 2008. Т. 65. № 3. P. 37–52. URL : https://journal.iitta.gov.ua/index.php/itlt/article/view/2041/1348 (дата звернення:10.11.2024) (in Ukrainian).

3. Peers Int. – експериментальна платформа відкритого рецензування та міжнародна віртуальна спільнота рецензентів. URL : https://peers.international/paper/rinok-praci-v-ukraini-pid-chas-viini-ta-scho-bude-pislya (дата звернення:10.11.2024).

4. STEM-освіта – Інститут модернізації змісту освіти. URL : http://imzo.gov.ua/stem-osvita (дата звернення:10.11.2024).

5. Види роботів і класифікація в робототехніці: всі області застосування і провідні виробники розумних машин. URL : https://worldbank.org.ua/3720-vidi-robotiv-i-klasifikatsiya-v-robototekhnitsi.html (дата звернення:17.11.2024)

6. Глазова В. В., Сурков М. І. Використання робототехніки під час уроків інформатики. *Збірник наукових праць фізико-математичного факультету ДДПУ*. 2022. Вип. 12. С. 65–69.

7. Електродвигуни в робототехніці - прокладаючи шлях до автоматизації. URL : https://ukrprommotor.com.ua/blog/elektrodviguni-v-robototexnitsi-prokladayuchi-shlyax-do-avtomatizatsii?srsltid=AfmBOoqxWIRfrEcwr6q1koZHb9TD6ibyaL-fT5RPC1cEyb2oHhhA2vZn (дата звернення:17.11.2024).

8. Кривонос О., Жуковський С., Кривонос М. Порівняння середовищ візуального програмування роботів для потреб навчального процесу. *Наука і техніка сьогодні*. 2022. № 6(6). С. 175-187.

9. Лабораторія програмування мікроконтролерів та робототехніки. URL : https://phm.cuspu.edu.ua/facultet/laboratoriia-prohramuvannia-mikrokontroleriv-ta-robototekhniky.html (дата звернення:17.11.2024).

10. Міні-комп'ютери – Raspberry Pi Україна. URL : https://www.raspberrypi.org/ (дата звернення:17.11.2024).

11. Оніщенко Д., Єфименко В. Класи в CIRCUIT TINKEDCAD: інноваційний підхід до вивчення та викладання програмування робототехнічних систем. *Збірник наукових праць Міжнародна науково-практична конференція «Сучасна наука та освіта: новітня соціокультурна проекція»*. Київ, 2024. С. 205-209.

12. Остапчук С. А., Садовий М. І. До проблеми використання платформи Arduino у вивченні робототехніки. *Наукові записки ЦДПУ*. *Серія: Педагогічні науки*. Кропивницький: РВВ ЦДПУ ім. В. Винниченка, 2018. Вип. 168. С. 178–181.

13. Програмування для робототехніки: мови та інструменти. URL : https://itproger.com/ua/news/programmirovanie-dlya-robototehniki-yaziki-i-instrumenti (дата звернення:17.11.2024).

14. Сокіл І. М., Ченцов О. М. Робототехніка. 7-9 класи (міжгалузевий інтегрований курс). Модельна навчальна програма для закладів загальної середньої освіти. «Рекомендовано Міністерством освіти і науки України» (наказ Міністерства освіти і науки України від 10.09.2024 № 1279). 2024. 14 с.

15. Струтинська О. В. Актуальність впровадження освітньої робототехніки в українську школу. *Електронне наукове фахове видання "Відкрите освітнє е-середовище сучасного університету", спецвипуск "Нові педагогічні підходи в STEAMосвіті"*. 2019. С. 324-344. DOI: https://doi.org/10.28925/2414-0325.2019s30. URL : http://openedu.kubg.edu.ua/journal/index.php/openedu/article/view/254 (дата звернення:10.11.2024) (in Ukraine).

16. Струтинська О. В. Навчання освітньої робототехніки в українських школах: напрями впровадження. *Інженерні та освітні технології*. 2019. Т.7. №3. С. 122–138. DOI : https://doi.org/10.30929/2307-9770.2019.07.03.011. URL : http://eetecs.kdu.edu.ua/2019_03/EETECS2019_007(3)_11.pdf (дата звернення:17.11.2024)

17. Струтинська О. В., Баранов С. С. Тенденції розвитку освітньої робототехніки в закладах позашкільної освіти. *Фізико-математична освіта*. 2019. Випуск 1(19). С. 196-204. DOI : 10.31110/2413-1571-2019-019-1-031. URL : https://fmo-journal.fizmatsspu.sumy.ua/journals/2019-v1-19/2019_1-19-Strutynska_Baranov_FMO.pdf (дата звернення:17.11.2024)

18. Типи сенсорів у робототехніці. URL : https://www.evsint.com/uk/types-of-sensors-in-robotics/ (дата звернення:17.11.2024)

19. Ченцов О. М. Модельна навчальна програма «робототехніка. 5–6 класи» для закладів загальної середньої освіти. *Освітня робототехніка* : зб.наук.пр.за матеріалами ІІ Всеукраїнської науково-практичної конференції «Освітня робототехніка» (14 квітня 2022 р.). Дніпро, 2022. С. 101-104.

# ДОДАТОК А

## Приклад конспектів уроків

### План-конспект уроку 2

**Тема уроку:** *Операційна система для платформи Arduino та налаштування середовища розробки*

**Мета уроку:** *навчальна* – Ознайомлення з операційною системою для платформи Arduino, налаштування середовища розробки Arduino IDE, основи програмування на мові C++ для Arduino; *розвивальна* – розвиток логічного мислення учнів, знайомство з основними поняттями програмування для створення роботизованих систем; *виховна* – виховання інтересу до технічної творчості та робототехніки, розвиток умінь з програмування.

**Тип уроку:** Ознайомлення з новим матеріалом, налаштування середовища розробки, виконання перших програм.

### Хід уроку

*І Організаційні моменти заняття*

**1 Привітання.** Доброго дня, учні! Сьогодні ми зробимо важливий крок у нашому навчанні – налаштуємо середовище для програмування на платформі Arduino і почнемо писати перші програми для роботизованих систем.

**2 Ознайомлення з планом уроку**

1. Встановимо і налаштуємо середовище Arduino IDE для програмування.
2. Познайомимося з основами програмування на мові C++ для Arduino.
3. Розглянемо, як визначати прості входи/виходи на платі Arduino та працювати з цифровими і аналоговими пінами.

*II Актуалізація знань, умінь та навичок*

**1 Обговорення попереднього досвіду**

*Запитання до учнів.* Хто з вас працював з програмуванням раніше? Які середовища розробки вам знайомі?

Коротка підготовка до встановлення та налаштування середовища Arduino IDE, зокрема обговорення важливості цього кроку для подальшої роботи з роботами.

*III Вивчення нового навчального матеріалу*

**1 Встановлення та налаштування середовища Arduino IDE**

*Що таке Arduino IDE?*

Arduino IDE (Integrated Development Environment) – це середовище, яке дозволяє писати, компілювати та завантажувати програми на мікроконтролери Arduino. В ньому ми можемо створювати проекти для роботів, сенсорних систем і автоматизованих пристроїв.

*Як встановити Arduino IDE?*

1. Перейдіть на офіційний сайт Arduino: https://www.arduino.cc/en/software.
2. Завантажте версію для вашої операційної системи (Windows, Mac або Linux).
3. Інсталюйте середовище, дотримуючись інструкцій.
4. Підключіть плату Arduino до комп'ютера через USB кабель.
5. У середовищі Arduino IDE виберіть модель плати (наприклад, Arduino Uno) і правильний порт для підключення.

**2 Основи програмування на платформі Arduino**

Програмування для Arduino здійснюється за допомогою мови, що є похідною від C++. Ось основні елементи програми:

Функція *setup()*: Виконується один раз на початку програми. Використовується для налаштування початкових параметрів (наприклад, встановлення пінів як виходи чи входи).

Функція *loop()*: Виконується безкінечно і відповідає за постійне виконання основної логіки програми (наприклад, вмикання світлодіода або зчитування значення з сенсора).

Приклад коду для вмикання світлодіода на піні 13 на рис. А.1

```
void setup() {
  pinMode(13, OUTPUT); // Встановлюємо пін 13 як вихід
}

void loop() {
  digitalWrite(13, HIGH); // Вмикаємо світлодіод
  delay(1000);             // Затримка 1 секунда
  digitalWrite(13, LOW);  // Вимикаємо світлодіод
  delay(1000);             // Затримка 1 секунда
}
```

Рисунок А.1 – Код для вмикання світло діода

**3 Перші кроки в програмуванні на мові C++ для Arduino**

**Змінні та типи даних**. Як у будь-якому програмуванні, ми працюємо з змінними для збереження даних. Наприклад:

```
int ledPin = 13; // Змінна для піну світлодіода
```

**Команди** *pinMode(), digitalWrite(), digitalRead()*: Для роботи з цифровими пінами використовуються такі команди:

*pinMode(pin, mode)* – встановлює пін як вхід або вихід;

*digitalWrite(pin, value)* – встановлює значення піну (вмикає або вимикає пристрій);

*digitalRead(pin)* – зчитує значення з цифрового піну (наприклад, зчитує стан кнопки).

**4 Робота з цифровими і аналоговими пінами**

**Цифрові піни.** Використовуються для підключення пристроїв, які мають два стани – вмикання/вимикання. Наприклад, підключення кнопки або світлодіода.

**Аналогові піни.** Використовуються для зчитування змінних сигналів, таких як значення температури або вологості. Наприклад, для зчитування значення з температурного датчика.

Приклад роботи з аналоговим піном на рис. А.2.

```
int sensorValue = 0;
void setup() {
  Serial.begin(9600); // Налаштовуємо серійну комунікацію
}

void loop() {
  sensorValue = analogRead(A0);  // Зчитуємо значення з піну A0
  Serial.println(sensorValue);    // Виводимо значення на серійний монітор
  delay(1000);                    // Затримка 1 секунда
}
```

Рисунок А.2 – Код для роботи з аналоговим піном

*IV Підсумок уроку*

**1 Узагальнення навчального матеріалу**

– Як ми налаштовуємо середовище Arduino IDE для програмування?

– Які основні функції використовуються для програмування на Arduino?

– Як працювати з цифровими та аналоговими пинами для керування пристроями?

**2 Закріплення знань:**

– обговорення важливості правильної настройки середовища розробки і правильної ініціалізації пінів для коректної роботи проектів.

*V Повідомлення домашнього завдання*

**Завдання на практичну роботу 1** Напишіть програму для Arduino, яка зчитує значення з температурного датчика (підключеного до аналогового піну) і виводить його на серійний монітор. Також для перегляньте відео YouTube.

Встановлення Arduino IDE та драйвера CH 341:

https://www.youtube.com/watch?v=_5-MrRr1t9k

Мова програмування та налаштування Arduino IDE:

https://www.youtube.com/watch?v=JGJompskGDw

Перша програма. Вбудований світлодіод. Підключення світлодіоду:

https://www.youtube.com/watch?v=vmoDkrI-4pg

**Завершення уроку**

Подякувати учням за активну участь у занятті. Нагадати, що на наступному уроці ми будемо практично застосовувати отримані знання для створення простих проектів.

**Резерв часу**

У разі потреби додатковий час можна використати для допомоги учням, які мають труднощі з налаштуванням середовища або розумінням коду.

## План-конспект уроку 3

**Тема уроку:** *Основи програмування на мові С++ для мікроконтролерів*

**Мета уроку:** *навчальна* – ознайомити учнів з основами програмування на мові С++ для мікроконтролерів, навчити працювати з простими сенсорами та приводами, а також використовувати бібліотеки Arduino для роботи з датчиками і сервоприводами; *розвивальна* – розвиток логічного мислення учнів, їх здатність до програмування та розв’язання технічних задач; виховна – виховання інтересу до робототехніки, технічної творчості та інженерії.

**Тип уроку:** Ознайомлення з новим матеріалом, вивчення основ мови С++ для програмування на платформі Arduino, практичні вправи.

**Хід уроку:**

*І Організаційні моменти заняття*

**1 Привітання.** Доброго дня, учні! Сьогодні ми продовжуємо вивчати програмування на платформі Arduino, зокрема розглянемо основи мови С++, яка використовується для написання програм для мікроконтролерів.

Під час уроку ми також познайомимося з основами роботи з сенсорами та приводами, навчимося використовувати бібліотеки Arduino для зчитування даних з датчиків і керування сервоприводами.

**2 Ознайомлення з планом уроку**

1. Основи мови С++: змінні, типи даних, оператори, цикли, умови.
2. Програмування для роботи з простими сенсорами та приводами.

3. Використання бібліотек Arduino для роботи з датчиками та сервоприводами.

*II Актуалізація знань, умінь та навичок*

**1 Обговорення попереднього досвіду**

*Запитання до учнів*. Як ви вважаєте, що таке змінні та як вони використовуються в програмуванні? Що ви знаєте про умовні оператори та цикли з попередніх уроків?

Коротке повторення основ програмування, які ми вивчали раніше: синтаксис мови C++, робота з Arduino IDE.

*II. Вивчення нового навчального матеріалу*

**1 Основи мови C++**

*Змінні та типи даних*

**Змінні** – це місця в пам'яті, де зберігаються значення, з якими працює програма.

**Типи даних** визначають, який тип значення може бути збережено в змінній. Основні типи даних:

*int* – ціле число (наприклад, 10, -5, 100);

*float* – дійсне число з плаваючою комою (наприклад, 3.14, -0.5);

*char* – одиничний символ (наприклад, 'A', 'b');

*boolean* – логічне значення (true/false).

Приклад оголошення змінних зображена на рис. А.3.

```
int ledPin = 13;      // змінна для піну світлодіода
float temperature = 25.5;  // змінна для температури
bool buttonState = false;  // змінна для зчитування стану кнопки
```

Рисунок А.3 – Оголошення змінних даних

*Оператори*

Оператори використовуються для виконання математичних або логічних операцій. Основні оператори:

– математичні: «+», «-», «*», «/», «%» (залишок від ділення).

– логічні: «&&» (І), «||» (АБО), «!» (НІ).

– порівняння: «==», «!=», «>», «<», «>=», «<=».

*Цикли*

Цикли використовуються для повторення певних дій кілька разів.

**Цикл for** (рис. А.4).

```
for (int i = 0; i < 10; i++) {
  // виконується 10 разів
}
```

Рисунок А.4 – Цикл for

**Цикл while** (рис. А.5).

```
while (condition) {
  // виконується, поки умова є істинною
}
```

Рисунок А.5 – Цикл while

*Умови*

Умовні оператори дозволяють виконувати код в залежності від певних умов.

**Оператор if** (рис. А.6).

```
if (condition) {
  // виконується, якщо умова істинна
}
else {
  // виконується, якщо умова хибна
}
```

Рисунок А.6 – Оператор if

## 2 Програмування для роботи з простими сенсорами та приводами

Для роботи з сенсорами та приводами (мотори, серводвигуни) ми використовуємо піні на платі Arduino, які можуть бути налаштовані як входи або виходи.

*Зчитування даних з сенсора*

Приклад зчитування температури з датчика через аналоговий пін зображено на рис. А.7.

```
int sensorValue = analogRead(A0);  // Зчитуємо значення з піну A0
float voltage = sensorValue * (5.0 / 1023.0); // Перетворюємо значення вольтаж
```

Рисунок А.7 – Зчитування температури з датчика

*Управління сервоприводами*

Для керування сервоприводами використовуємо бібліотеку Servo.h. (рис. А.8).

```
#include <Servo.h>
Servo myServo;

void setup() {
  myServo.attach(9); // Підключаємо сервопривод до піну 9
}

void loop() {
  myServo.write(90);  // Поворот на 90 градусів
  delay(1000);         // Затримка 1 секунда
  myServo.write(0);   // Поворот на 0 градусів
  delay(1000);         // Затримка 1 секунда
}
```

Рисунок А.8 – Код для керування сервоприводами

## 3 Використання бібліотек Arduino для роботи з датчиками та сервоприводами

**Бібліотеки Arduino** дозволяють значно спростити роботу з різними пристроями, такими як сенсори або сервоприводи. Бібліотеки надають готові фун-

кції для керування цими пристроями, що значно знижує складність програмування.

Приклад використання бібліотеки для роботи з датчиком температури (DHT11) зображено на рис. А.9.

*IV Підсумок уроку*

**1 Узагальнення навчального матеріалу**

Що таке змінні, типи даних та як ми їх використовуємо в програмах для Arduino?

Як працювати з умовами, циклами та операторами в мові C++ для Arduino?

Як використовувати бібліотеки Arduino для роботи з сенсорами та сервоприводами?

```
#include <DHT.h>

DHT dht(2, DHT11);  // Датчик підключено до піну 2

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" *C  Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");
  delay(2000);
}
```

Рисунок А.9 – Код для роботи з датчиком температури

**2 Закріплення знань**

Обговорення основних принципів програмування: робота з піном, зчитування даних з сенсорів, керування сервоприводами.

*V Повідомлення домашнього завдання*

**Завдання на практичну роботу 2** Напишіть програму для зчитування даних з температурного датчика DHT11 і відображення їх на серійному моніторі. Додатково використайте умовний оператор для порівняння температури з певним значенням і виведення відповідного повідомлення. Як завжди пропоную подивитися пізнавальне відео:

https://www.youtube.com/watch?v=ikoJaZ6wnTw

Якщо немає можливості працювати з платою Arduino, можна використовувати Tinkercad:

https://www.youtube.com/watch?v=b2mt1PmwQi0&list=PLE0mrIIEVMb23-7mxUkyv32IICNF8lOp5&index=4

І ще трохи про програмування Arduino:

https://www.youtube.com/watch?v=o0xKODzRLAI&list=PLE0mrIIEVMb23-7mxUkyv32IICNF8lOp5&index=5

**Завершення уроку**

Подякувати учням за активну участь, нагадати про важливість правильної роботи з сенсорами і приводу для створення роботизованих систем.

## План-конспект уроку 4

**Тема уроку:** *Створення та програмування простих роботизованих систем*

**Мета уроку:** *навчальна* – ознайомити учнів з процесом створення та програмування базових роботизованих пристроїв, таких як мобільні роботи; *розвивальна* – розвиток навичок програмування для керування рухом роботів, використання датчиків для навігації та інтеракції з навколишнім середовищем; *виховна* – виховання інтересу до робототехніки та інженерії, розвиток творчого підходу до вирішення технічних задач.

**Тип уроку:** ознайомлення з новим матеріалом, практичне застосування знань для створення та програмування роботів.

**Хід уроку:**

*І Організаційні моменти заняття*

**1 Привітання.** Доброго дня, учні! Сьогодні ми почнемо створювати та програмувати прості роботизовані системи. Це дуже важливий крок на шляху до створення мобільних роботів, які можуть рухатися, взаємодіяти з оточенням і навіть виконувати прості задачі.

**2 Ознайомлення з планом уроку**

1. Як створювати базові роботизовані пристрої, наприклад, мобільних роботів.

2. Як програмувати рух роботів за допомогою моторів і датчиків.

3. Як використовувати датчики для навігації, зокрема ультразвукові датчики для вимірювання відстані.

*ІІ Актуалізація знань, умінь та навичок*

**1 Обговорення попереднього досвіду**

*Запитання до учнів.* Які компоненти, на вашу думку, потрібні для створення мобільного робота? Які типи датчиків можна використовувати для навігації?

*Коротке повторення.* Як ми працюємо з двигунами, сервоприводами і сенсорами для створення рухомих систем.

*ІІІ Вивчення нового навчального матеріалу*

**1 Створення базових роботизованих пристроїв (мобільних роботів)**

**Мобільний робот** – це пристрій, який може переміщатися в просторі за допомогою коліс або інших приводних механізмів. Для створення мобільного робота зазвичай використовують:

**мотори** – для руху коліс;

**мікроконтролер** – для управління роботами та взаємодії з датчиками.

**датчики** – для отримання інформації про навколишнє середовище (наприклад, датчики відстані, ультразвукові датчики).

*Приклад мобільного робота.* Мобільний робот на основі Arduino зазвичай має два або чотири колеса, два мотори для руху і, можливо, додаткові датчики для уникнення перешкод.

**2 Програмування для руху робота за допомогою моторів і датчиків**

Для програмування руху робота використовуються мотори, які підключаються до вихідних пінів на платі Arduino. Рух залежить від того, як управляються ці мотори за допомогою команд у програмі.

*Основи програмування для руху робота*

Використовуємо бібліотеки для керування моторами. Призначаємо піні для підключення моторів, керуємо напрямком їх обертання. Приклад програмування руху робота на рис. А.10.

```
int motorPin1 = 3; // Пін для першого мотора
int motorPin2 = 4; // Пін для другого мотора

void setup() {
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
}

void loop() {
  digitalWrite(motorPin1, HIGH); // Мотор рухається вперед
  digitalWrite(motorPin2, LOW);
  delay(1000);                   // Рухаємося 1 секунду

  digitalWrite(motorPin1, LOW);  // Мотор зупиняється
  digitalWrite(motorPin2, LOW);
  delay(1000);                   // Затримка 1 секунда
}
```

Рисунок А.10 – Програмування руху робота

*Принцип роботи*

1. Датчик відправляє ультразвукові хвилі.

2. Хвилі відбиваються від перешкоди і повертаються назад.

3. Вимірюється час, за який хвилі проходять до перешкоди і назад, і обчислюється відстань.

Приклад програми для ультразвукового датчика HC-SR04 на рис. А.11.

*IV Підсумок уроку*

**1 Узагальнення навчального матеріалу**

– Як створюються прості роботизовані системи, зокрема мобільні роботи?

– Як програмувати рух робота за допомогою мотора та датчиків?

– Як використовувати ультразвукові датчики для вимірювання відстані та навігації?

**2 Закріплення знань**

Обговорення важливості правильного використання датчиків для навігації та уникнення перешкод. Як правильно програмувати рухи робота в залежності від даних з датчиків.

```
int triggerPin = 9;  // Пін для відправки сигналу
int echoPin = 10;    // Пін для прийому сигналу
long duration;
int distance;

void setup() {
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0344 / 2; // Вираховуємо відстань в см

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);  // Затримка 0.5 секунди
}
```

Рисунок А.11 – Програма для ультразвукового датчика

*V Повідомлення домашнього завдання*

**Завдання на практичну роботу 3** Напишіть програму для робота, який буде рухатися вперед, поки не наблизиться до перешкоди на відстань менше 20 см, після чого змінить напрямок і поїде в іншу сторону. Перед виконанням практичної прогляньте відео

https://www.youtube.com/watch?v=uEFH9w2vOHQ&list=PLiiM1MtRdEuXAyALthxwIQs3ZI8DYSF3d&index=14

Гайд по двигунам

https://www.youtube.com/watch?v=XEZJZ3v5kQs&list=PLy54-DJf-RYfH0aX51egA1e2DZyvjV07n&index=7

**Завершення уроку**

Подякувати учням за активну участь. Нагадати, що на наступному уроці ми будемо створювати складніші роботизовані системи та тестувати різні типи датчиків.

**Резерв часу**

У разі потреби додатковий час можна використати для допомоги учням, які мають труднощі з програмуванням або підключенням датчиків.

## План-конспект уроку 5

**Тема уроку:** *Робота з сенсорами та приводами*

**Мета уроку:** *навчальна* – ознайомити учнів з різними типами сенсорів, такими як ультразвукові, температурні, світлові, та їх використанням для збору даних і навігації роботів. Навчити програмуванню роботів для роботи з сенсорами та використання сервоприводів і крокових моторів для створення рухомих частин роботів. *Розвивальна* – розвиток навичок програмування для зчитування сенсорних даних і керування роботами; виховна – виховання інтересу до робототехніки, розвитку технологій і технічної творчості.

**Тип уроку:** ознайомлення з новим матеріалом, практичне застосування знань для роботи з сенсорами та приводами.

### Хід уроку:

*І Організаційні моменти заняття*

**1 Привітання**

Доброго дня, учні! Сьогодні ми розглянемо дуже важливі аспекти роботи роботів – роботу з сенсорами та приводами. Сенсори дозволяють роботам взаємодіяти з навколишнім середовищем, а приводи дають можливість рухати частинами робота. Це знання є основою для створення складних роботизованих систем.

**2 Ознайомлення з планом уроку**

1. Ознайомимося з різними типами сенсорів: ультразвуковими, температурними, світловими.

2. Навчимося програмувати роботи для використання сенсорних даних.

3. Розглянемо застосування сервоприводів і крокових моторів для створення рухомих частин роботів.

*ІІ Актуалізація знань, умінь та навичок*

**1 Обговорення попереднього досвіду**

*Запитання до учнів.* Які типи сенсорів ви знаєте? Як вони можуть допомогти роботам взаємодіяти з навколишнім середовищем?

*Підсумки.* Поговоримо про типи сенсорів, які часто використовуються в робототехніці для збору даних, а також про приводні системи, які дозволяють роботу виконувати фізичні дії.

*ІІІ Вивчення нового навчального матеріалу*

**1 Ознайомлення з різними типами сенсорів**

Сенсори – це пристрої, які дозволяють роботам отримувати інформацію з навколишнього середовища. Розглянемо кілька основних типів сенсорів.

Ультразвукові датчики вимірюють відстань до об'єктів, використовуючи принцип відбиття ультразвукових хвиль. Вони широко використовуються для визначення перешкод в навігації робота.

Ультразвуковий датчик (HC-SR04) має два піні: для відправки ультразвукових хвиль і для прийому відбитих хвиль. Вимірюється час, за який сигнал проходить до перешкоди і повертається назад, і обчислюється відстань. Приклад програми для роботи з ультразвуковим датчиком (HC-SR04) на рис А.12.

Температурні датчики вимірюють температуру навколишнього середовища. Одним із популярних датчиків є DHT11. Програма для роботи з температурним датчиком DHT11 на рис. А.13.

Світлові датчики (фотоелементи) використовуються для вимірювання інтенсивності світла. Вони дозволяють роботам реагувати на зміни освітлення. Програма для роботи зі світловим датчиком (фотоелементом) на рис А.14.

```
int triggerPin = 9;
int echoPin = 10;
long duration;
int distance;

void setup() {
  Serial.begin(9600);
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0344 / 2;
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(500);
}
```

Рисунок А.12 – Програма для роботи з ультразвуковим датчиком HC-SR04

```
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  float temp = dht.readTemperature();
  float humidity = dht.readHumidity();

  if (isnan(temp) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.print(" C ");
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");
  delay(2000);
}
```

Рисунок А.13 – Програма для роботи з температурним датчиком

```arduino
int lightPin = A0;  // Пін для світлового датчика
int lightValue = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  lightValue = analogRead(lightPin);  // Зчитуємо значення світла
  Serial.print("Light level: ");
  Serial.println(lightValue);
  delay(1000);  // Затримка 1 секунда
}
```

Рисунок А.14 – Програма для роботи зі світловим датчиком

**2 Програмування роботів для використання сенсорних даних**

Роботи можуть використовувати дані з сенсорів для здійснення різних дій, наприклад, для уникнення перешкод, контролю температури або взаємодії з навколишнім середовищем.

Програмування для роботи з сенсорами зазвичай включає зчитування даних і виконання певних дій, залежно від результату. Наприклад, якщо робот виявляє перешкоду на відстані менше ніж 10 см, він може змінити напрямок. Приклад програми для уникнення перешкод за допомогою ультразвукового датчика на рис. А.15.

**3 Застосування сервоприводів і крокових моторів для створення рухомих частин роботів**

Сервоприводи дозволяють точно контролювати положення частини робота (наприклад, маніпуляторів або камери). Вони зазвичай мають обмежений кут повороту (від 0 до 180 градусів).

Крокові мотори дозволяють точно контролювати кут повороту і рухати роботом по точно заданому шляху. Вони використовуються для створення механізмів з точною позицією.

*IV Підсумок уроку*

**1 Узагальнення навчального матеріалу**

– Що таке сенсори і як вони використовуються для збору даних з навколишнього середовища?

– Як програмувати роботи для використання сенсорних даних для навігації та уникнення перешкод?

– Як застосовувати сервоприводи і крокові мотори для створення рухомих частин роботів?

**2 Закріплення знань**

Обговорення прикладів використання різних типів сенсорів для конкретних завдань: від ультразвукових датчиків до світлових датчиків.

```
int motorPin1 = 3;
int motorPin2 = 4;
int triggerPin = 9;
int echoPin = 10;
long duration;
int distance;

void setup() {
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0344 / 2;

  if (distance < 10) {
    // Якщо відстань менша за 10 см, зупинити рух
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
  } else {
    // Інакше рухатися вперед
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
  }

  delay(500);
}
```

Рисунок А.15 – Програми для уникнення перешкод за допомогою ультразвукового датчика

Приклад використання сервоприводу для руху маніпулятора на рис. А.16

```
#include <Servo.h>

Servo myServo;

void setup() {
  myServo.attach(9);  // Підключаємо сервопривод до піну 9
}

void loop() {
  myServo.write(0);   // Поворот на 0 градусів
  delay(1000);         // Затримка 1 секунда
  myServo.write(90);  // Поворот на 90 градусів
  delay(1000);         // Затримка 1 секунда
}
```

Рисунок А.16 – Програма для використання сервоприводу для руху маніпулятора

*V Повідомлення домашнього завдання*

**Завдання на практичну роботу 4** Створіть програму для робота, який використовує ультразвуковий датчик для уникнення перешкод, а також зчитує рівень освітленості і змінює рух залежно від рівня світла. Перегляньте пізнавальне відео https://www.youtube.com/watch?v=mWdbUS_t1_E&list=PLy54-DJf-RYfH0aX51egA1e2DZyvjV07n&index=19

https://www.youtube.com/watch?v=sJoqMboqxvA&list=PLy54-DJf-RYfH0aX51egA1e2DZyvjV07n&index=29

**Завершення уроку**

Подякувати учням за активну участь у занятті. Нагадати, що на наступному уроці ми будемо розглядати складніші задачі, пов'язані з роботами, які здатні реагувати на більше кількість сенсорних даних.

**Резерв часу**

У разі потреби, додатковий час можна використати для допомоги учням з налаштуванням датчиків або програмуванням їх використання в роботах.

## План-конспект уроку 6

**Тема уроку:** *Інтерфейси та комунікації Arduino і інших пристроїв*

**Мета уроку:** *навчальна* – ознайомити учнів з основами комунікацій між Arduino та іншими пристроями, зокрема зчитуванням даних і відправкою команд. Розглянути використання послідовних портів для обміну даними між Arduino та ПК або іншими платами. *Розвивальна* – розвиток навичок використ-

тання різних типів комунікаційних інтерфейсів для взаємодії з периферійними пристроями та іншими платами; *виховна* – виховання інтересу до сучасних бездротових технологій і їх застосування в робототехніці для інтеграції роботизованих систем.

**Тип уроку:** ознайомлення з новим матеріалом, практичне застосування знань для налаштування і програмування інтерфейсів Arduino.

**Хід уроку:**

*І Організаційні моменти заняття*

**1 Привітання**

Доброго дня, учні! Сьогодні ми будемо вивчати основи комунікацій між платформою Arduino та іншими пристроями. Це дуже важливий аспект для створення роботизованих систем, які можуть обмінюватися даними, а також взаємодіяти з ПК і іншими платами.

**2 Ознайомлення з планом уроку**

1. Ознайомимося з основами комунікацій між Arduino і іншими пристроями.

2. Розглянемо використання послідовних портів для обміну даними з ПК або іншими платами.

3. Ознайомимося з Bluetooth, Wi-Fi та іншими бездротовими технологіями для інтеграції роботизованих систем.

*ІІ Актуалізація знань, умінь та навичок*

**1 Обговорення попереднього досвіду**

*Запитання до учнів.* Чи працювали ви з передачою даних між комп'ютером і мікроконтролерами раніше? Які інтерфейси для передачі даних ви знаєте?

*Коротке повторення.* Раніше ми використовували Arduino для програмування роботів і сенсорів. Сьогодні ми будемо вивчати, як взаємодіяти з іншими пристроями через різні інтерфейси.

*ІІІ Вивчення нового навчального матеріалу*

**1 Основи комунікацій між Arduino та іншими пристроями**

Комунікації між Arduino і іншими пристроями дозволяють зчитувати дані (наприклад, з сенсорів) та відправляти команди на зовнішні пристрої (наприклад, на серводвигуни або інші плати).

Для цього Arduino використовує різні інтерфейси, такі як:

– послідовний порт (Serial Communication);

– I2C (Inter-Integrated Circuit);

– SPI (Serial Peripheral Interface).

Найпоширенішим є *послідовний порт*, який дозволяє обмінюватися даними між Arduino і іншими пристроями через дві лінії: TX (передача) та RX (прийом).

При програмування послідовної комунікації використовуємо функції *Serial.begin(), Serial.print(), Serial.println(), Serial.read()* для обміну даними між Arduino та ПК.

Приклад коду для послідовного зчитування та відправки даних наведений на рис. А.17

```
void setup() {
  Serial.begin(9600);  // 9600 – швидкість передачі даних
}

void loop() {
  if (Serial.available() > 0) {  // Перевіряємо наявність вхідних даних
    int incomingByte = Serial.read();  // Зчитуємо вхідні дані
    Serial.print("I received: ");
    Serial.println(incomingByte);  // Виводимо отримані дані на серійний монітор
  }
}
```

Рисунок А.17 – Код для послідовного зчитування та відправки даних

**2 Використання послідовних портів для обміну даними з ПК або іншими платами**

Arduino підключається до комп'ютера через USB, що дозволяє використовувати послідовний порт для обміну даними. Це зручно для відладки, тестування, а також для інтеграції з іншими програмами на ПК.

Послідовний монітор в Arduino IDE дозволяє виводити дані на екран комп'ютера для моніторингу роботи роботів або сенсорних систем.

Приклад передачі даних з Arduino на ПК через послідовний монітор представлено на рис. А.18.

```
void setup() {
  Serial.begin(9600); // Налаштування серійної передачі
}

void loop() {
  Serial.println("Hello, World!");  // Виводимо повідомлення на ПК
  delay(1000);  // Затримка 1 секунда
}
```

Рисунок А.18 – Передача даних з Arduino на ПК через послідовний монітор

## 3 Ознайомлення з Bluetooth, Wi-Fi та іншими бездротовими технологіями для інтеграції роботизованих систем

У сучасних роботизованих системах важливою частиною є бездротові технології для дистанційного управління та передачі даних. Розглянемо основні технології.

*Bluetooth* використовується для бездротової передачі даних на невеликій відстані. Більшість Bluetooth-модулів, таких як HC-05, можуть бути підключені до Arduino для забезпечення комунікації між роботом і смартфоном чи комп'ютером. Приклад підключення Bluetooth модуля HC-05 до Arduino на рис. А.19.

```
#define BT_RX 10
#define BT_TX 11

void setup() {
  Serial.begin(9600);
  Serial.println("Bluetooth Serial Started");
}

void loop() {
  if (Serial.available()) {
    char data = Serial.read();
    Serial.print("Received: ");
    Serial.println(data);
  }
}
```

Рисунок А.19 – Підключення Bluetooth модуля HC-05 до Arduino

*Wi-Fi* для бездротової комунікації на більші відстані зазвичай використовуються модулі **ESP8266** або **ESP32**, які підтримують Wi-Fi з'єднання. Вони дозволяють робити роботи частиною IoT (Інтернету речей), обмінюватися даними через інтернет і керувати роботами віддалено. Приклад підключення ESP8266 до Arduino для підключення до Wi-Fi мережі наведено на рис. А.20.

*RFID і IR (Інфрачервоні технології)* також використовуються для бездротової ідентифікації і комунікації, наприклад, для створення безконтактних систем або для керування роботами через пульти.

*IV Підсумок уроку*

**1 Узагальнення навчального матеріалу**

– Як здійснюється комунікація між Arduino та іншими пристроями через послідовний порт?

– Як підключити Arduino до ПК і передавати/отримувати дані?

– Як використовувати Bluetooth і Wi-Fi для бездротового управління роботами?

```
#include <ESP8266WiFi.h>

const char* ssid = "your_SSID";  // Назва вашої мережі Wi-Fi
const char* password = "your_PASSWORD";  // Ваш пароль Wi-Fi

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
}

void loop() {
  // Код для обміну даними через Wi-Fi
}
```

Рисунок А.20 – Підключення ESP8266 для підключення до Wi-Fi мережі

**2 Закріплення знань**

Обговорення важливості комунікаційних інтерфейсів для інтеграції роботизованих систем та взаємодії з іншими пристроями.

*V Повідомлення домашнього завдання*

**Завдання на практичну роботу 5.** Створіть програму, яка передає дані з Arduino на ПК за допомогою Bluetooth. Для цього підключіть модуль HC-05 до плати Arduino та відправте прості повідомлення через послідовний монітор. Також перегляньте пізнавальні відео

https://www.youtube.com/watch?v=emJTkQO8-Wk&list=PLy54-DJf-RYfH0aX51egA1e2DZyvjV07n&index=20

https://www.youtube.com/watch?v=YIT20E0DC8o&list=PLy54-DJf-RYfH0aX51egA1e2DZyvjV07n&index=21

**Завершення уроку**

Подякувати учням за активну участь. Нагадати, що на наступному уроці ми будемо використовувати ці технології для створення роботизованих систем, що здатні працювати бездротово.

**Резерв часу**

У разі потреби додатковий час можна використати для допомоги учням з налаштуванням Bluetooth або Wi-Fi модулів, або для виконання практичних вправ з комунікаціями між пристроями.

## План-конспект уроку 7

**Тема уроку:** *Створення проекту роботизованої системи*

**Мета уроку:** *навчальна* – ознайомити учнів з процесом планування, реалізації та тестування проекту роботизованої системи; *розвивальна* – розвиток практичних навичок створення роботів для виконання конкретних завдань та тестування роботизованих систем; виховна – виховання в учнів творчого підходу до розв'язання технічних завдань, уміння працювати в команді, а також розвитку інженерного мислення.

**Тип уроку:** практичне заняття з розробки, реалізації та тестування роботизованих систем.

**Хід уроку:**

*І Організаційні моменти заняття*

**1 Привітання**

Доброго дня, учні! Сьогодні ми займемося створенням проектів роботизованих систем. Це важливий етап у розвитку робототехніки, коли ми застосовуємо все, що вивчили раніше, і створюємо робочі системи для вирішення реальних завдань.

**2 Ознайомлення з планом уроку**

1. Розглянемо процес планування та реалізації проекту роботизованої системи.

2. Виконаємо практичні завдання з розробки роботів для специфічних цілей.

3. Пройдемо етап тестування і налагодження роботи створених проектів.

*ІІ Актуалізація знань, умінь та навичок*

**1 Обговорення попереднього досвіду**

*Запитання до учнів*. Які етапи створення робота вам вже знайомі? Як ви вважаєте, що є важливим при плануванні роботизованого проекту?

*Підсумки.* Розгляд основних етапів створення робота, таких як планування, розробка, тестування та налагодження.

*ІІІ Вивчення нового навчального матеріалу*

**1 Планування та реалізація проекту роботизованої системи**

Планування проекту – це важливий крок, який дозволяє чітко визначити мету роботи та ресурсні вимоги. Кожен проект складається з кількох основних етапів.

*Етап 1 – Визначення мети проекту.* Що робитиме робот? Яку проблему вирішує система? Наприклад, це може бути робот для навігації по лабіринту або робот, який перевіряє рівень вологи в ґрунті.

*Етап 2 – Вибір компонентів.* На цьому етапі ми визначаємо, які компоненти будуть використовуватися в проекті: плата Arduino, моторчики, датчики, сервоприводи та інші компоненти.

*Етап 3 – Складання схеми.* Схема підключення компонентів дозволяє зрозуміти, як будуть з'єднуватися всі частини роботизованої системи, і на що треба звернути увагу при розробці проекту.

*Етап 4 – Написання програми.* Програмування робота для виконання необхідних завдань: рух, взаємодія з датчиками, виконання команд.

*Етап 5 – Тестування і налагодження.* Після створення робота важливо перевірити його роботу, провести тестування і налагодити систему, щоб вона працювала без збоїв.

**2 Практичні завдання по створенню роботів для специфічних цілей**

Після планування проекту наступним кроком є практичне створення роботів. Ось декілька варіантів тем для завдань, які можуть бути реалізовані:

– *Робот для навігації по лабіринту.* Завдання полягає в тому, щоб створити робота, який може рухатися по лабіринту, використовуючи ультразвукові датчики для виявлення перешкод. Цей робот має вміти знаходити шлях, уникати зіткнень і досягати мети.

– *Робот для вимірювання рівня вологості в ґрунті.* Це робот, який використовує датчики вологості для визначення рівня вологості в ґрунті. Якщо рівень вологи занадто низький, робот може активувати насос для поливу.

– *Робот-пилосос.* Цей робот використовує сенсори для визначення об'єктів, що перешкоджають його руху, і відправляє команди для зміни напрямку руху. Крім того, може бути доданий датчик для вимірювання рівня заряду батареї.

**3 Тестування та налагодження роботи створеного проекту**

Після того як робот побудований і програма написана, важливим етапом є *тестування.* Це дозволяє перевірити, чи виконується проект за призначенням.

Тестування датчиків – перевірка правильності зчитування даних із сенсорів (ультразвукових, температурних, вологості тощо).

Тестування рухів робота – перевірка правильності роботи мотора і сервоприводів (чи рухається робот у потрібному напрямку).

Налагодження помилок – виправлення помилок в програмі, таких як неправильна логіка руху чи некоректні дані з сенсорів.

Після того як всі помилки усунено, робот має бути готовий до роботи у заданих умовах.

*IV Підсумок уроку*

**1 Узагальнення навчального матеріалу**

– Як правильно планувати і реалізовувати проект роботизованої системи?

– Як виконати практичне завдання по створенню робота для специфічної мети?

– Як тестувати і налаштовувати роботизовану систему після її створення?

**2 Закріплення знань**

Обговорення важливих аспектів створення роботів для реальних завдань: планування, вибір компонентів, програмування і тестування.

*V Повідомлення домашнього завдання*

**Завдання на практичну роботу 5.** Створити свій власний проект, реалізувати його і протестувати.

**Завершення уроку.** Подякувати учням за активну участь у занятті. Нагадати, що на наступному уроці ми будемо працювати над більш складними проектами роботів та інтегрувати різні сенсори для поліпшення їхніх функцій.

**Резерв часу**

У разі потреби додатковий час можна використати для допомоги учням, які мають труднощі з тестуванням своїх проектів або налаштуванням роботів.

## План-конспект уроку 8

**Тема уроку:** *Професійне становлення та можливості для розвитку в галузі робототехніки*

**Мета уроку:** *навчальна* – ознайомити учнів з професіями, які існують у галузі робототехніки, інженерії та програмування, а також з перспективами кар'єри в цій сфері; *розвивальна* – підвищити зацікавленість учнів у розвитку своїх технічних навичок, розвивати розуміння важливості робототехніки для сучасного світу та ринку праці; *виховна* – Виховання прагнення до професійного

розвитку в галузі новітніх технологій та робототехніки, стимулювання вибору кар'єри в цій перспективній і динамічній галузі.

**Тип уроку:** завершальний урок курсу, на якому підбиваються підсумки вивченого матеріалу і розглядаються можливості для професійного становлення в галузі робототехніки.

**Хід уроку:**

*І Організаційні моменти заняття*

**1 Привітання**

Доброго дня, учні! Сьогодні ми завершуємо наш курс з робототехніки, і на цьому уроці поговоримо про можливості розвитку кар'єри в цій галузі. Робототехніка – це одна з найбільш перспективних та динамічно розвиваються галузей, і для багатьох з вас вона може стати основною професією в майбутньому.

**2 Ознайомлення з планом уроку**

1. Ознайомимося з професіями в сфері робототехніки, інженерії та програмування.

2. Розглянемо роль робототехніки в сучасному світі та ринку праці.

3. Обговоримо перспективи розвитку кар'єри в сфері робототехніки та автоматизації.

*ІІ Актуалізація знань, умінь та навичок*

**1 Обговорення попереднього досвіду**

*Запитання до учнів.* Які професії в галузі робототехніки та програмування вам вже відомі? Які навички, на вашу думку, потрібні для роботи в цій галузі?

*Короткий підсумок.* Раніше ми вивчали основи програмування, роботу з сенсорами, створення роботизованих систем. Це лише частина необхідних знань для розвитку кар'єри в галузі робототехніки.

*ІІІ Вивчення нового навчального матеріалу*

**1 Ознайомлення з професіями в сфері робототехніки, інженерії та програмування**

У галузі робототехніки є безліч спеціалізацій, що охоплюють різні аспекти проектування, створення та підтримки роботизованих систем.

*Інженер з робототехніки* – спеціаліст, який проектує, розробляє і тестує роботизовані системи. Це можуть бути як промислові роботи, так і мобільні роботи, медичні пристрої тощо.

*Програміст робототехніки* – розробляє програмне забезпечення для управління роботами, інтеграції сенсорних систем, обробки даних. Програмісти використовують мови програмування, такі як C++, Python, Java.

*Інженер з автоматизації* – працює над створенням автоматизованих виробничих ліній, розробкою систем для автоматичного управління та моніторингу.

*Спеціаліст з інтерфейсів і комунікацій* – розробляє системи, що дозволяють роботам і пристроям взаємодіяти через різні інтерфейси (наприклад, через Wi-Fi, Bluetooth, USB або інші мережі).

Консультант з робототехніки – спеціаліст, який допомагає компаніям впроваджувати робототехнічні рішення для автоматизації різних процесів.

**2 Роль робототехніки в сучасному світі та ринку праці**

*Робототехніка в промисловості.* Роботи активно використовуються на виробничих підприємствах для автоматизації процесів, таких як складання, пакування, обробка матеріалів. Це дозволяє підвищити продуктивність і знизити витрати.

*Робототехніка в медицині.* Роботи допомагають в хірургії, діагностиці, реабілітації пацієнтів, полегшуючи роботу медиків і покращуючи точність процедур.

*Робототехніка в сільському господарстві.* Автоматизація збору врожаю, обробки ґрунту, поливу, управління тваринами – все це дозволяє зробити сільське господарство більш ефективним.

*Роботи для побуту.* Роботи-пилососи, роботи для прибирання та інших домашніх завдань стають все більш популярними.

*Технології майбутнього.* Із розвитком штучного інтелекту, машинного навчання та самонавчальних алгоритмів робототехніка має потенціал для створення автономних систем, які можуть самостійно працювати в складних і небезпечних умовах.

Ринок праці для робототехніків і інженерів є дуже перспективним і з кожним роком зростає. Попит на фахівців в цій галузі постійно збільшується, оскільки робототехніка займає все більше місця в нашому повсякденному житті.

**3 Перспективи розвитку кар'єри в сфері робототехніки та автоматизації**

У міру того, як робототехніка стає все важливішою в промисловості та інших сферах, зростає попит на фахівців з робототехніки, інженерії та програмування.

Професіонали, які бажають кар'єрно зростати, можуть пройти додаткове навчання або сертифікацію у спеціалізованих інститутах або онлайн-курсах. Ось кілька варіантів для розвитку кар'єри: стати експертом з робототехнічних систем, які використовуються в специфічних сферах (медицина, промисловість, сільське господарство), працювати в стартапах, що займаються розробкою нових роботизованих технологій, сприяти розвитку інноваційних рішень у галузі автоматизації, якщо ви хочете поєднувати дослідження і викладання, ви можете стати викладачем або науковим співробітником в університетах, що спеціалізуються на робототехніці та автоматизації.

*V Підсумок уроку*

**1 Узагальнення навчального матеріалу**

– Які професії є в галузі робототехніки, інженерії та програмування?

– Як робототехніка змінює сучасний світ і які можливості для кар'єри відкриваються в цій галузі?

– Які перспективи для кар'єри в робототехніці та автоматизації?

**2 Закріплення знань**

Обговорення важливості розвитку навичок, необхідних для роботи в галузі робототехніки: програмування, інженерія, комунікації, дослідження та інновації.

*V Повідомлення домашнього завдання*

**Домашнє завдання.** Підготуйте короткий звіт або презентацію про одну з професій в галузі робототехніки, яка вам найбільше цікава. Опишіть основні вимоги до цієї професії, ключові навички і можливості для розвитку кар'єри.

**Завершення уроку.** Подякувати учням за активну участь у курсі. Підсумувати основні досягнення і відзначити важливість вивченого матеріалу для подальшого розвитку кар'єри в сфері робототехніки та автоматизації.

**Резерв часу**

У разі потреби додатковий час можна використати для розгляду додаткових тем або відповідей на запитання учнів.