

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ  
Фізико-математичний факультет  
Кафедра інформатики та прикладної математики

«Допущено до захисту»  
авідувач кафедри

\_\_\_\_\_ Моїсеєнко Н. В.  
«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

Реєстраційний № \_\_\_\_\_  
«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**РЕАЛІЗАЦІЯ STEM-ПІДХОДУ У ПРОФІЛЬНОМУ НАВЧАННІ  
ІНФОРМАТИКИ НА ПРИКЛАДІ РОЗРОБКИ ПРИСТРОЮ-ПОМІЧНИКА  
ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ**

Кваліфікаційна робота студента  
групи Ім-22  
ступінь вищої освіти «магістр»  
спеціальності 014 Середня освіта (Інформатика)  
**Командирчика Андрія Васильовича**

Керівник: доц., к. ф.-м.н.  
Мерзликін Павло Володимирович

Оцінка:  
Національна шкала \_\_\_\_\_  
Шкала ECTS \_\_\_ Кількість балів \_\_\_\_

Голова ЕК \_\_\_\_\_  
Члени ЕК \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## **ЗАПЕВНЕННЯ**

Я, Командирчик Андрій Васильович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело. Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

## Зміст

<b>ВСТУП</b> .....	4
<b>РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ВИКОРИСТАННЯ STEM-ПІДХОДУ</b> .....	6
1.1 Суть, характеристика та мета STEM-підходу .....	6
1.2 Методи та технології STEM-підходу .....	11
1.3 Реалізація STEM-підходу на уроках інформатики у профільному навчанні.....	22
<b>ВИСНОВКИ ДО РОЗДІЛУ 1</b> .....	27
<b>РОЗДІЛ 2. СТВОРЕННЯ ТА ТЕСТУВАННЯ ПРИСТРОЮ ДЛЯ ЛЮДЕЙ З ОСОБЛИВИМИ ПОТРЕБАМИ</b> .....	29
2.1 Використання апаратно-програмного комплексу Arduino для підтримки вивчення програмування у профільному навчанні .....	29
2.2 Пристрої та їх основні принципи роботи для створення пристрою для людей з особливими потребами .....	31
2.3 Програмна та фізична реалізація пристрою .....	37
<b>ВИСНОВКИ ДО РОЗДІЛУ 2</b> .....	41
<b>РОЗДІЛ 3. МЕТОДИКА STEM-ПІДХОДУ ПРИ ВИКЛАДАННІ ІНФОРМАТИКИ У ПРОФІЛЬНОМУ НАВЧАННІ</b> .....	43
3.1 Практичне використання STEM –підходу на уроках з інформатики у профільному навчанні .....	43
3.2 Оцінка результатів впровадження STEM-підходу у профільному навчанні інформатики.....	45
<b>ВИСНОВКИ ДО РОЗДІЛУ 3</b> .....	49
<b>ВИСНОВКИ</b> .....	50
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ</b> .....	52
<b>Додаток А</b> .....	57

## ВСТУП

STEM-освіта спрямована на формування у учнів компетентностей, які необхідні для успішної адаптації та самореалізації в суспільстві, заснованому на знаннях, технологіях та інноваціях. STEM-освіта також сприяє розвитку творчого та критичного мислення, розв'язанню реальних проблем, співпраці та комунікації.

Розробка пристрою-помічника для людей з вадами зору є одним із можливих прикладів застосування STEM-підходу у профільному навчанні інформатики. Такий проєкт може бути цікавим та мотивуючим для учнів, адже він має соціальну значущість, викликає емпатію та відповідальність, вимагає застосування знань та навичок з різних дисциплін, таких як математика, фізика, електроніка, програмування, дизайн тощо. Він також може бути корисним для розвитку професійних компетенцій учнів, які планують пов'язати своє майбутнє з галузями STEM.

**Об'єкт дослідження:** STEM-підхід у профільному навчанні інформатики.

**Предмет дослідження:** впровадження STEM-підходу в профільне навчання інформатики на прикладі циклу уроків з розробки пристрою для людей з вадами зору.

**Мета:** розробити цикл уроків зі створення пристрою для людей з вадами зору для впровадження у профільне навчання інформатики.

Для дослідження мети слід розв'язати такі **задачі:**

- на основі аналізу нормативної бази та інших джерел з'ясувати загальну характеристику, мету STEM-підходу, методи й технології STEM-підходу;
- проаналізувати попередні дослідження щодо реалізації STEM-підходу на уроках інформатики;
- обрати інструменти для створення навчального проєкту — прототип пристрою для людей з вадами зору;
- реалізувати апаратну й програмну складові прототипу пристрою;

- розробити цикл уроків для впровадження в профільне навчання інформатики.

**Новизна роботи** полягає в тому, що в її рамках розроблено цикл уроків, який демонструє можливість впровадження STEM-підходу в профільне навчання інформатики.

**Структура** роботи обумовлена логікою логіко дослідження. Кваліфікаційна робота складається із вступу, трьох розділів, висновків до кожного розділу, висновків, списку літератури, що нараховує 32 джерела.

## РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ВИКОРИСТАННЯ STEM-ПІДХОДУ

### 1.1 Суть, характеристика та мета STEM-підходу

Сучасні проблеми в економіці, соціальних та політичних системах потребують нових підходів до модернізації та покращення освіти в умовах глобалізації інформаційної ери. Для підвищення якості освіти необхідно розвивати освіту наукового напрямку, яка ґрунтується на дослідно-орієнтованому, проєктному навчанні, спрямована на поглиблене вивчення профільних предметів природничо-математичного профілю та формування компетентностей, необхідних для подальшої наукової дослідно-експериментальної, конструкторської, винахідницької діяльності, є пріоритетним напрямком розвитку освіти України, який сприяє зміцненню економіки держави. Сучасні глобальні соціально-економічні процеси пов'язані з виникненням і стрімким розвитком високоефективних нано- та біоматеріалів, нової енергетики й інформаційних мереж. Найближче майбутнє формуватимуть інноваційні галузі виробництва та пов'язані з ними професії, які ґрунтуються на таких основних технологіях та продуктах, як геоінженерія, розумні енергетичні мережі, синтетична біологія, персоналізована геноміка, біоінтерфейси, сонячна енергетика, препарати для підвищення когнітивних функцій, нові енергоємні акумулятори, стовбурові клітини, біопаливо, клонування, робототехніка, польоти на низькі орбіти, мемристори, мобільні мережі та засоби зв'язку, акумулятори, що заряджаються від атмосфери, розумні навігаційні системи, штучний інтелект та інші.

Формування та розвиток STEM-освіти як педагогічної інновації XXI століття було спричинено інноваційними процесами суспільно-економічного розвитку та зростанням попиту на фахівців високотехнологічних галузей, які мають здатність до комплексної наукової та інженерної діяльності.

**STEM** (від англ. Science – природничі науки; Technology – технології; Engineering – інжиніринг, проєктування, дизайн; Mathematics – математика) – термін, який означає сучасну освітню парадигму в розв'язанні питань

освітньої політики та формування навчальних програм на основі інтеграції природничоматематичних дисциплін і технологій, зокрема інформаційно-комунікаційних технологій (ІКТ).

STEM-освіта — це педагогічна технологія, яка спрямована на формування та розвиток розумово-пізнавальних і творчих якостей здобувачів освіти, які визначають їх конкурентоспроможність на сучасному ринку праці. Поняття STEM-освіти не має єдиного визначення, кожна країна адаптує його до своїх потреб. У загальному розумінні — це освітній напрям, який охоплює широкий спектр діяльності в галузях інженерії, інформатики, математики та природничих наук. До них належать, зокрема, комп'ютерна, машинобудівна, екологічна, аерокосмічна, хімічна, енергетична, атомна, біомедична, хімічна інженерія, інформаційні технології, геоматика, мехатроніка, програмування, агротехнологія, атмосферні та космічні дослідження[48].

В Україні **STEM-освіта** — це система природничої і математичної освітніх галузей, яка має на меті розвиток особистості через формування компетентностей, природничо-наукової картини світу, світоглядних позицій і життєвих цінностей. Вона базується на трансдисциплінарному підході до навчання, практичному застосуванні наукових, математичних, технічних та інженерних знань для розв'язання практичних проблем та їхнього використання у професійній діяльності [23].

Чотири основні предмети STEM-освіти мають велике значення для розуміння та перетворення навколишнього світу:

- наука досліджує природні явища, закони природи, які об'єднують фізику, хімію, біологію, а також розвиває навички застосування принципів, концепцій та фактів які стосуються цих дисциплін;
- технології охоплюють всю систему людей і організацій, знання, процеси і пристрої, які беруть участь у створенні та функціонуванні технологічних виробів, а також самі вироби, які є результатами технологічної діяльності;

- інжиніринг використовує знання про вигляд та створення продуктів і можливості вирішення проблем. Інжиніринг використовує знання з науки та математики, а також технологічні методи та засоби;
- математика досліджує правила і зв'язки між величинами, числами та фігурами. Її можна розглядати як з теоретичної сторони або так і з практичним застосуванням.

***Характеристиками STEM-освіти є:***

- учні отримують знання з різних наукових дисциплін, які допомагають їм встановлювати зв'язки між ними та використовувати їх у реальному житті;
- учні залучаються до активного та інтерактивного навчання, яке сприяє розвитку критичного, алгоритмічного та інженерного мислення, математичної логіки, цифрової грамотності, креативності та інноваційності, навичок комунікації;
- учні застосовують проєктну та проблемну методику навчання, яка стимулює їх до самостійного пошуку, дослідження, моделювання, експериментування, створення власних продуктів та рішень;
- учні користуються сучасними технологіями, які допомагають їм вивчати інформатику в контексті реального життя, як-от комп'ютерні програми, онлайн-сервіси, віртуальні лабораторії, симулятори, конструктори, робототехніка, датчики, сенсори, дрони, штучний інтелект, віртуальна та доповнена реальність та інші [23].

***Метою STEM-освіти є:***

- учнів мають підготувати до успішної адаптації до вимог сучасного світу, який вимагає швидкої зміни, високої конкуренції, глобалізації та цифровізації;
- збудити у учнів інтерес до природничих наук, технологій, інженерії та математики, а також мотивувати їх до навчання та саморозвитку в цих галузях;



- розвинути в учнів ключові компетентності, які потрібні для професійної діяльності в науково-технічних, математичних, інженерних сферах, а також для вирішення соціальних та екологічних проблем;
- покращити якість освіти та її відповідність потребам ринку праці, а також до нових досягнень науки та технологій [23].

Для реалізації STEM-підходу в Україні була розроблена Концепція розвитку природничо-математичної освіти (STEM-освіти) до 2027 року, яка була ухвалена Урядом 5 серпня 2020 року [23]. Концепція визначає основні принципи, цілі, завдання, напрями, механізми та очікувані результати розвитку STEM-освіти в Україні. Концепція також передбачає розроблення та затвердження державної програми розвитку STEM-освіти, яка має включати конкретні заходи, терміни, відповідальних виконавців, джерела фінансування та критерії оцінки ефективності.

У старшій школі програма навчання спрямована на практичне використання набутих умінь та поглиблення знань з усіх чотирьох складових STEM. Учні виконують складніші та триваліші досліди, а також розробляють проєкти, які мають на меті вирішити актуальні для людства проблеми сучасності: розвитку альтернативних джерел енергії, зниження забруднення планети, глобального потепління, раціонального використання ресурсів та інші. Учні зосереджуються переважно на можливостях, які відкриває STEM-освіта за межами школи.

STEM-освіта спрямована на досягнення таких *целей*:

- залучення школярів до технічної творчості, нових технологій, досліджень у міжпредметних суміжних галузях;
- розвиток у молодих інноваторів умінь і навичок (креативність, вміння бачити і розв'язувати проблеми. Уміння працювати в команді, комунікативні навички);
- підтримка наукової, технічної та інженерної складових в додатковій освіті школярів та розширення можливостей доручення учнів до

роботи у природничо-наукових та інженерних лабораторіях, надання їм доступу до сучасного обладнання та інноваційних програм;

- мотивація учнів старших класів до продовження освіти в науково-технічній та інженерній сферах, ознайомлення їх з новими технологіями;
- популяризація винахідницької та науково-дослідницької діяльності;
- проєктно-орієнтоване навчання школярів під керівництвом молодих вчених та інженерів і формування експертної спільноти з оцінки результатів діяльності STEM-центрів регіонального, обласного і районного рівнів;
- створення умов для адаптації і впровадження інноваційних програм, створених за участі провідних промислових і конструюючих підприємств або організацій, пов'язаних з програмами додаткової освіти для школярів [23].

STEM-освіта ставить перед собою такі основні *завдання*:

- розвиток компетенцій і навичок, які потрібні на ринку праці XXI століття, таких як:
  - готовність до розв'язання складних (комплексних ) практичних проблем;
  - критичне мислення;
  - креативність;
  - організаційні здібності;
  - вміння працювати в команді;
  - емоційний інтелект;
  - оцінювати проблеми і приймати рішення;
  - здатність до ефективної взаємодії, вміння спілкуватися з різними людьми;
- вміння домовлятися;
- когнітивна гнучкість;
- різнобічний розвиток;

- формування цілісного наукового світогляду, загальнонаукової, загальнокультурної, технологічної, комунікативної і соціальної компетентностей;
- становлення соціально-компетентної особистості, здатної здійснювати самостійний вибір і приймати відповідальні рішення у різноманітних життєвих ситуаціях;
- виховання в особистості любові до праці, забезпечення умов для її життєвого і професійного самовизначення, формування готовності до свідомого вибору і оволодіння майбутньою професією [23].

## **1.2 Методи та технології STEM-підходу**

Процес створення навчальних програм STEM розпочинається з визначення цілей, завдань, змісту та методів STEM-освіти, які відповідають потребам сучасного суспільства та ринку праці. Для цього необхідно аналізувати сучасні тенденції в науці, технологіях, інженерії та математиці, враховувати досвід країн, які успішно реалізують STEM-підходи. Особливу увагу слід приділяти формуванню STEM-компетентності, яка включає знання, уміння, навички, спосіб мислення, цінності та особистісні якості, які необхідні для інноваційної діяльності.

Процес створення навчальних програм STEM вимагає постійного моніторингу, оцінювання, коригування та удосконалення якості та ефективності STEM-освіти, з урахуванням змін у суспільстві, науці та технологіях. Для цього слід використовувати різні критерії, показники, інструменти та методи оцінювання, які відображають рівень сформованості STEM-компетентності учнів та вчителів.

Міждисциплінарні підходи в STEM-освіті — це способи навчання, які поєднують знання та методи з різних наукових та технічних дисциплін, щоб розв'язувати складні проблеми, пов'язані з реальним життям. Міждисциплінарні підходи сприяють розвитку критичного мислення, творчості, комунікації та інноваційності учнів, а також формуванню STEM-компетентності, яка визначає здатність до інноваційної діяльності.

STEM-підхід відрізняється від традиційних освітніх моделей тим, що він інтегрує знання та методи з різних наукових та технічних дисциплін, щоб розв'язувати складні проблеми, пов'язані з реальним життям, а відтак, формує та комплексно розвиває наукове й інженерне мислення.

У рамках STEM-навчання завдання побудовані відповідно до наукового методу й інженерного дизайну, які є основою будь-якого процесу досліджень незалежно від галузі пізнання.

Вчені й інженери розвивають сучасну науку різними способами, проте мають на меті розв'язання конкретних проблем. Учені використовують науковий метод, щоб перевірити теоретичні припущення та висунути прогнози щодо навколишнього світу. Науковець ставить запитання і здійснює експеримент або декілька експериментів, щоб знайти відповідь. Навпаки, інженери застосовують інженерний процес проектування, щоб знайти вирішення завдання. З цією метою інженер встановлює конкретну мету, а потім розробляє її рішення.

**Науковий метод** в STEM-освіті - це спосіб мислення та дослідження, який допомагає учням вивчати природні явища, вирішувати проблеми, перевіряти гіпотези, робити висновки та застосовувати знання в реальних ситуаціях.

Науковий метод в STEM-освіті складається з таких етапів:

- Постановка запитання або проблеми, яку потрібно дослідити або вирішити.
- Дослідження та збір інформації про запитання або проблему з різних джерел, таких як книги, інтернет, експерти, спостереження тощо.
- Висування гіпотези або припущення, яке пояснює запитання або проблему та може бути перевірене експериментально.
- Планування та проведення експерименту або діяльності, які дозволяють перевірити гіпотезу або розв'язати проблему, використовуючи наукові та технологічні інструменти та матеріали.

– Збір, аналіз та інтерпретація даних, отриманих в ході експерименту або діяльності, за допомогою математичних методів та графічних засобів.

– Формування висновків, які підтверджують або спростовують гіпотезу або розв’язують проблему, та порівняння їх з іншими джерелами інформації.

– Підбиття результатів дослідження або демонстрація розв’язання проблеми іншим людям, використовуючи різні форми та засоби вираження, такі як словесні, письмові, графічні, цифрові тощо.

**Постановка запитання або проблеми**, яку потрібно дослідити або вирішити — це перший етап наукового методу в STEM-освіті, який визначає мету та напрямок дослідження.

На цьому етапі учні формулюють запитання або проблему, які їх цікавлять, пов’язані з реальним життям, науковими явищами або технологічними викликами.

Можна виокремити наступні правила постановки запитання або проблеми:

– повинні бути чіткими, конкретними, зрозумілими, вимірюваними та досяжними;

– повинні мати наукову або технологічну значимість, тобто вони повинні сприяти розширенню знань або поліпшенню якості життя;

– можуть бути сформульовані як відкриті, так і закриті, залежно від типу дослідження. Відкриті запитання або проблеми не мають однозначної відповіді або розв’язання, а закриті запитання або проблеми мають одну або кілька відповідей або розв’язань, які можна перевірити експериментально.

**Етап дослідження та збору інформації** наукового методу в STEM-освіті — це другий етап, який допомагає учням знайти відповіді на запитання або проблеми, які вони поставили на першому етапі. На цьому етапі учні досліджують та збирають інформацію про запитання або проблему з різних джерел, таких як книги, інтернет, експерти, спостереження тощо. Учні повинні

вибирати надійні та актуальні джерела, порівнювати та аналізувати отриману інформацію, визначати її суттєвість та пов'язаність з запитанням або проблемою. Учні також повинні враховувати різні точки зору та аргументи, що стосуються запитання або проблеми, та критично оцінювати їх достовірність та обґрунтованість. Учні повинні організовувати та записувати отриману інформацію, використовуючи різні форми та засоби вираження, такі як словесні, письмові, графічні, цифрові тощо.

**Висування гіпотези** допомагає учням сформулювати можливе пояснення запитання або проблеми, які вони досліджують. На цьому етапі учні висувають гіпотезу або припущення, яке відображає їхнє розуміння причинно-наслідкових зв'язків між досліджуваними явищами, та може бути перевірене експериментально.

Гіпотеза або припущення повинні бути чіткими, конкретними, зрозумілими, вимірюваними та досяжними. Гіпотеза або припущення також повинні мати наукову або технологічну значимість, тобто вони повинні сприяти розширенню знань або поліпшенню якості життя.

Гіпотеза або припущення можуть бути сформульовані як відкриті, так і закриті, залежно від типу дослідження. Відкриті гіпотези або припущення не мають однозначної відповіді або розв'язання, а закриті гіпотези або припущення мають одну або кілька відповідей або розв'язань, які можна перевірити експериментально.

**Планування та проведення експерименту** допомагає учням перевірити гіпотезу або розв'язати проблему, які вони висунули на третьому етапі. На цьому етапі учні планують та проводять експеримент або діяльність, які дозволяють перевірити гіпотезу або розв'язати проблему, використовуючи наукові та технологічні інструменти та матеріали.

Учні повинні визначити мету, завдання, об'єкт, предмет, змінні, методи, процедури, критерії та показники експерименту або діяльності. Учні також повинні враховувати різні фактори, що впливають на результати експерименту або діяльності, та забезпечити їх контроль, відтворюваність та достовірність.

Учні повинні виконувати експеримент або діяльність відповідно до плану, дотримуючись правил безпеки, етики та екології.

*Збір, аналіз та інтерпретація даних* допомагає учням оцінити результати експерименту або діяльності, які вони провели на четвертому етапі. На цьому етапі учні збирають, аналізують та інтерпретують дані, отримані в ході експерименту або діяльності, за допомогою математичних методів та графічних засобів.

Учні повинні визначити тип, обсяг, формат, джерела та достовірність даних. Учні також повинні враховувати різні способи подання, обробки, аналізу та інтерпретації даних, такі як таблиці, діаграми, графіки, статистичні показники, кореляційний аналіз, регресійний аналіз тощо. Учні повинні використовувати відповідні програмні засоби, такі як Excel, GeoGebra, Statistica, SPSS тощо. Учні повинні висловлювати свої висновки, використовуючи наукову мову та термінологію.

*Підбиття результатів дослідження* — це сьомий і останній етап, який допомагає учням представити та поширити свої досягнення, отримані в ході дослідження. На цьому етапі учні підбивають результати дослідження, використовуючи різні форми та засоби вираження, такі як словесні, письмові, графічні, цифрові тощо.

Учні повинні враховувати різні аудиторії та мету представлення результатів дослідження, такі як вчителі, однокласники, батьки, експерти, громадськість тощо. Учні також повинні використовувати наукову мову та термінологію, а також відповідні стилістичні та оформлювальні правила. Учні повинні бути готові до захисту своїх результатів дослідження, відповідати на запитання, аргументувати свої висновки, отримувати та враховувати зворотний зв'язок.

**Процес інженерного проєктування** в STEM-освіті — це послідовність кроків, які допомагають учням розробляти та реалізовувати рішення для реальних проблем за допомогою наукових, технологічних, інженерних та

математичних знань та навичок. Процес інженерного проєктування складається з таких етапів:

- визначення проблеми;
- дослідження;
- генерація ідей;
- проєктування;
- створення;
- тестування;
- оцінка;
- удосконалення.

Етап **визначення проблеми** — це перший крок, на якому учні виявляють та уточнюють проблему, яку потрібно вирішити. На цьому етапі учні повинні:

Зрозуміти контекст та мету проєкту, які можуть бути задані вчителем або обрані самостійно.

- сформулювати проблему у вигляді запитання, яке відображає потреби та бажання користувачів або клієнтів;
- визначити критерії та обмеження для рішення, які враховують технічні, економічні, соціальні, екологічні та етичні аспекти;
- перевірити, чи є проблема актуальною, реальною, досяжною та вимірюваною.

Етап визначення проблеми допомагає учням сфокусуватися на суті проєкту, встановити цілі та очікування, а також спланувати наступні кроки. Цей етап є важливим для успішної реалізації проєкту, адже якщо проблема не визначена чітко та коректно, то рішення може бути неефективним або непридатним.

Етап **дослідження** — це другий крок, на якому учні збирають інформацію про проблему, яку вони обрали для розв'язання. На цьому етапі учні повинні:



- використовувати різні джерела та методи для пошуку, аналізу та оцінки інформації, яка стосується проблеми, її причин, наслідків, контексту та можливих рішень;
- застосовувати наукові та математичні знання та навички для розуміння проблеми, її параметрів, обмежень та критеріїв успішності;
- вивчати існуючі рішення, які вже були запропоновані або реалізовані для подібних або пов'язаних проблем, виявляти їхні переваги та недоліки, аналізувати їхню ефективність та придатність;
- використовувати цифрові технології та інструменти для обробки, візуалізації, моделювання та презентації інформації, яка допоможе у розв'язанні проблеми.
- співпрацювати з іншими учнями, вчителями, експертами або клієнтами, обмінюватися інформацією, ідеями, думками та досвідом, отримувати та надавати зворотний зв'язок.

Етап дослідження допомагає учням підготуватися до наступного етапу генерації ідей, на якому вони будуть висувати свої варіанти рішень. Цей етап є важливим для якісного та інноваційного рішення, адже якщо учні не мають достатньо інформації про проблему, її контекст та вимоги, то вони не зможуть запропонувати оптимальне та оригінальне рішення.

Етап *генерації ідей* — це третій крок, на якому учні висувають різні варіанти рішень для проблеми, яку вони обрали для розв'язання. На цьому етапі учні повинні:

- застосовувати творчі та критичні методи мислення, такі як мозковий штурм, аналогії, синектика тощо, для генерування якомога більшої кількості ідей;
- використовувати наукові, технологічні, інженерні та математичні знання та навички для розробки концепцій, принципів, алгоритмів, схем, моделей, прототипів тощо, які відображають ідеї рішень;

- враховувати критерії та обмеження для рішення, які були визначені на попередньому етапі, а також вимоги та очікування користувачів або клієнтів;
- оцінювати переваги та недоліки кожної ідеї, порівнювати їх між собою, використовуючи різні методи, такі як матриця оцінки, діаграма Парето, дерево рішень тощо;
- співпрацювати з іншими учнями, вчителями, експертами або клієнтами, обмінюватися ідеями, думками, досвідом, отримувати та надавати зворотний зв'язок.

Етап генерації ідей допомагає учням підготуватися до наступного етапу проєктування, на якому вони будуть обирати найкраще рішення та розробляти план його реалізації. Цей етап є важливим для якісного та інноваційного рішення, адже якщо учні не мають достатньо ідей, то вони не зможуть вибрати оптимальне та оригінальне рішення.

Етап *проєктування* — це четвертий крок, на якому учні обирають найкраще рішення для проблеми, яку вони обрали для розв'язання, та розробляють план його реалізації. На цьому етапі учні повинні:

- вибрати одну або декілька ідей, які відповідають критеріям та обмеженням для рішення, а також задовольняють потреби та бажання користувачів або клієнтів;
- розробити детальний план реалізації рішення, який включає кроки, необхідні матеріали, технології, інструменти, час, бюджет, розподіл обов'язків, способи контролю та оцінки тощо;
- створити документацію для рішення, яка містить опис, мету, цілі, критерії, обмеження, концепцію, принцип, алгоритм, схему, модель, прототип тощо, які відображають рішення;
- використовувати цифрові технології та інструменти для обробки, візуалізації, моделювання та презентації рішення, яке буде створено;

- співпрацювати з іншими учнями, вчителями, експертами або клієнтами, обмінюватися ідеями, думками, досвідом, отримувати та надавати зворотний зв'язок.

Етап проектування допомагає учням підготуватися до наступного етапу створення, на якому вони будуть виготовляти прототип або модель свого рішення. Цей етап є важливим для якісного та інноваційного рішення, адже якщо учні не мають чіткого плану та документації для рішення, то вони не зможуть ефективно та своєчасно реалізувати свою ідею.

Етап **створення** — це п'ятий крок, на якому учні виготовляють прототип або модель свого рішення для проблеми, яку вони обрали для розв'язання. На цьому етапі учні повинні:

- виконувати план реалізації рішення, який був розроблений на попередньому етапі, використовуючи необхідні матеріали, технології, інструменти, час, бюджет тощо;
- створювати прототип або модель рішення, який відображає концепцію, принцип, алгоритм, схему, модель тощо, які були розроблені на попередніх етапах;
- використовувати наукові, технологічні, інженерні та математичні знання та навички для виготовлення, налаштування, збирання, з'єднання, програмування, калібрування тощо прототипу або моделі рішення;
- використовувати цифрові технології та інструменти для обробки, візуалізації, моделювання та презентації прототипу або моделі рішення, який був створений;
- співпрацювати з іншими учнями, вчителями, експертами або клієнтами, обмінюватися ідеями, думками, досвідом, отримувати та надавати зворотний зв'язок.

Етап створення допомагає учням підготуватися до наступного етапу тестування, на якому вони будуть перевіряти функціональність та ефективність свого рішення. Цей етап є важливим для якісного та

інноваційного рішення, адже якщо учні не мають готового прототипу або моделі рішення, то вони не зможуть демонструвати та оцінювати свою ідею.

Етап **тестування** — це шостий крок, на якому учні перевіряють функціональність та ефективність свого рішення для проблеми, яку вони обрали для розв’язання. На цьому етапі учні повинні:

- виконувати різні експерименти та вимірювання, щоб перевірити, чи працює прототип або модель рішення, який був створений на попередньому етапі, відповідно до очікувань та вимог;
- використовувати наукові, технологічні, інженерні та математичні знання та навички для аналізу, інтерпретації та оцінки результатів експериментів та вимірювань, виявлення помилок, недоліків, ризиків, переваг та недоліків рішення;
- використовувати цифрові технології та інструменти для обробки, візуалізації, моделювання та презентації результатів експериментів та вимірювань, які були проведені;
- співпрацювати з іншими учнями, вчителями, експертами або клієнтами, обмінюватися ідеями, думками, досвідом, отримувати та надавати зворотний зв’язок.

Етап тестування допомагає учням підготуватися до наступного етапу оцінки, на якому вони будуть аналізувати результати тестування, виявляти сильні та слабкі сторони свого рішення, порівнювати його з іншими варіантами. Цей етап є важливим для якісного та інноваційного рішення, адже якщо учні не перевіряють та не оцінюють своє рішення, то вони не зможуть виявити та усунути можливі проблеми, поліпшити та удосконалити свою ідею.

Етап **оцінки** — це сьомий крок, на якому учні аналізують результати тестування, виявляють сильні та слабкі сторони свого рішення, порівнюють його з іншими варіантами. На цьому етапі учні повинні:

- використовувати наукові, технологічні, інженерні та математичні знання та навички для аналізу, інтерпретації та оцінки результатів експериментів та вимірювань, які були проведені на попередньому етапі, виявлення помилок, недоліків, ризиків, переваг та недоліків рішення;
- порівнювати своє рішення з іншими варіантами, які були запропоновані або реалізовані для подібних або пов'язаних проблем, враховуючи критерії та обмеження для рішення, а також вимоги та очікування користувачів або клієнтів;
- використовувати цифрові технології та інструменти для обробки, візуалізації, моделювання та презентації результатів оцінки рішення, яка була проведена;
- співпрацювати з іншими учнями, вчителями, експертами або клієнтами, обмінюватися ідеями, думками, досвідом, отримувати та надавати зворотний зв'язок.

Етап оцінки допомагає учням підготуватися до наступного етапу удосконалення, на якому вони будуть вносити необхідні зміни та поліпшення до свого рішення, враховуючи отриману зворотню зв'язок та нові дані. Цей етап є важливим для якісного та інноваційного рішення, адже якщо учні не аналізують та не оцінюють своє рішення, то вони не зможуть виявити та усунути можливі проблеми, поліпшити та удосконалити свою ідею.

Етап **удосконалення** — це восьмий та останній крок, на якому учні вносять необхідні зміни та поліпшення до свого рішення, враховуючи отриманий зворотний зв'язок та нові дані. На цьому етапі учні повинні:

- використовувати наукові, технологічні, інженерні та математичні знання та навички для модифікації, вдосконалення, оптимізації, масштабування, стандартизації тощо свого рішення, яке було створено та протестовано на попередніх етапах;

- використовувати цифрові технології та інструменти для обробки, візуалізації, моделювання та презентації удосконаленого рішення, яке було створено;
- співпрацювати з іншими учнями, вчителями, експертами або клієнтами, обмінюватися ідеями, думками, досвідом, отримувати та надавати зворотний зв'язок.

Етап удосконалення є завершальним для інженерного проектування STEM-освіти, але не означає, що рішення не може бути покращене в майбутньому. Учні повинні розуміти, що інженерне проектування - це ітеративний процес, який вимагає постійного вдосконалення та адаптації до змінних умов та потреб.

Процес інженерного проектування в STEM-освіті не є жорстко фіксованим або лінійним, адже учні можуть повертатися до попередніх етапів, якщо це необхідно, або пропускати деякі кроки, якщо вони не застосовні до конкретної проблеми. Головна мета процесу інженерного проектування — навчити учнів критично та креативно мислити, співпрацювати в командах, використовувати наукові методи та технології, а також створювати інноваційні продукти, які відповідають потребам суспільства.

### **1.3 Реалізація STEM-підходу на уроках інформатики у профільному навчанні**

Потреба в переосмисленні навчального процесу направила освітні заклади в нове русло сучасної освіти. Школи поступово оновлюють своє матеріальне забезпечення, що стає основою для якісного навчання та нових ідей. Вчителі поступово відмовляються від застарілих методів навчання і переходять до сучасних провідних систем. Перехід у це русло змушує отримувати нові знання та оновлювати свої навчальні програми. Тому зараз відбувається процес появи великої кількості курсів, навчальних програм, де відбувається не лише навчання, а і не менш об'ємний творчий процес. Такі курси та програми потребують різноманіття, а отже існує місце для творчості.

Сучасні вчителі, відвідуючи курси чи певні освітні програми, не лише розглядають нові системи навчання, а й спостерігають творчий процес створення нових, та оновлення вже наявних освітніх програм. Відвідавши ці курси, вчителі повертаються в школи та починають свій творчий процес проведення уроків.

Основною проблемою реалізації нових підходів до навчання є необхідність перепланувати кожен урок, переглянути навчальний план. На це потрібен час, тому педагоги з великим досвідом не охоче переходять до нових систем. Проте стрімкий розвиток технологій поступово впливає на всі сфери життя і педагоги поступово переходять до нових концепцій та програм. Спостерігаючи цей процес вчителі, які підтримують класичну систему викладання, також починають переходити до нових концепцій та програм.

Однією із цих концепцій є STEM-освіта. Сучасні технології дозволяють швидко та без зайвих зусиль викласти будь-який матеріал. Вчителі, розуміючи, що тепер їм не треба витрачати багато часу на певні питання, починають більш детально розглядати теми. Таким чином з'являється можливість не лише детально розглядати тему, а й розглянути її зв'язок з іншими предметами. Особливо це стосується предметів природничо-математичного циклу.

Інформатика не просто входить до предметів природничо-математичного циклу, а й спрямована на пришвидшення розв'язання проблем цих наук. Так завдання з інформатики дозволяють демонструвати не лише предметні зв'язки з іншими предметами, а й її практичну значимість. Таким чином інформатика є величезним простором для реалізації STEM-освіти.

Розглядаючи інформатику в сучасній науковій діяльності, не можна не погодитись, що будь-яка наукова діяльність не обходиться без використання сучасних комп'ютерних і програмних засобів.

Так інформатика поєднує в собі всі компоненти STEM-освіти. Тому сучасні учні позитивно ставляться до вивчення інформатики і мають змогу самостійно розглядати новий матеріал, а якщо врахувати той факт, що сучасна

мережа інтернету має велику кількість навчального матеріалу, то самонавчання стає досить простим.

Таким чином сучасні десятикласники вивчаючи новий матеріал на уроках інформатики, мають змогу робити це як їм заманеться. Таким чином виконуючи певне завдання з інформатики учні завжди підходять до нього творчо, адже існує безліч його виконання і кожен варіант є правильним. При чому виконуючи завдання учні мають змогу виконати невелике дослідження, або провести певний експеримент.

Залежно від завдання на уроці інформатики учні можуть займатись науковою діяльністю чи проєктуванням, маючи великий репозитарій різних програмних засобів інформатика зводить все до бажання учнів, проте розглядаючи той творчий простір, що відкривається перед учнями, то бажання в них з'являється дуже швидко.

Так інформатика «наводить мости» з іншими науками, поєднує їх в собі, а також створює позашкільної діяльності.

Працівники майбутнього мають вирішувати проблеми, розуміючи й використовуючи наукові підходи, знаючи технології, якими можна вирішити ці проблеми. Проте зазвичай підготувати таких працівників є завданням вищих навчальних закладів. Проте, маючи великий потенціал до наукової діяльності, інформатика дає змогу почати ознайомлення з науковими підходами і технологіями вже у шкільному віці. Десятикласники вже володіють певними базовими знаннями, а отже можуть займатись науковою діяльністю. В цьому випадку інформатика грає роль певного інструменту, який має на меті допомогти у вирішенні певних питань, пришвидшити розрахунки, спростити побудову моделі.

Для реалізації STEM-підходу на уроках інформатики у профільному навчанні необхідно враховувати такі аспекти:

- визначення цілей, завдань, змісту та методів навчання з урахуванням особливостей профільного класу, інтересів та потреб учнів, а також вимог сучасного ринку праці;



- використання інтегрованого підходу до навчання, який передбачає зв'язок інформатики з іншими предметами, такими як математика, фізика, хімія, біологія, географія, історія, мистецтво тощо;
- застосування активних та інтерактивних форм навчання, які сприяють залученню учнів до пізнавальної діяльності, розвитку творчих здібностей, самостійності, критичності, співпраці та комунікації. Такі форми можуть бути: дослідницькі, проєктні, проблемні, кейс-методи, групові та індивідуальні завдання, дискусії, дебати, рольові ігри, презентації, портфоліо тощо;
- використання різноманітних ресурсів та засобів навчання, які допомагають учням вивчати інформатику в контексті реального життя, моделювати та аналізувати явища та процеси, створювати власні продукти та рішення. Такі ресурси та засоби можуть бути: комп'ютерні програми, онлайн-сервіси, віртуальні лабораторії, симулятори, конструктори, робототехніка, датчики, сенсори, дрони, штучний інтелект, віртуальна та доповнена реальність тощо.

Реалізація STEM-підходу на уроках інформатики у профільному навчанні є новим та актуальним напрямком освітньої реформи, який спрямований на формування у учнів компетентностей, необхідних для успішної адаптації до вимог сучасного світу.

Попередні дослідження [10, 30, 31] з цієї теми переважно фокусувалися на реалізації STEM-підходу на уроках інформатики у початковій та середній школі, а також на використанні різних інтернет-ресурсів, сервісів та платформ для інтеграції природничих наук, технологій, інженерії та математики в навчальний процес. Наприклад, в рамках проєкту «Інтеграція STEM-освіти в загальноосвітній процес» за підтримки Фонду ім. Фулбрайта в Україні, автори рекомендацій – Ірина Курінна та Олександр Степаненко [27] – є викладачами інформатики та математики в Київському національному університеті імені Тараса Шевченка. Рекомендації містять приклади використання різних

інтернет-ресурсів, сервісів та платформ для інтеграції природничих наук, технологій, інженерії та математики на уроках інформатики та на інших предметах. Такий та подібні проекти надають методичні рекомендації та приклади використання таких ресурсів, як Scratch, Ph.ET, Geo.Gebra, Google Документи, Padlet, Kahoot!, Google-форми та інші.

Однак, на даний момент немає достатньо досліджень, які б розглядали реалізацію STEM-підходу на уроках інформатики у профільному навчанні, а також на використанні сучасних технологій, які дозволяють учням моделювати та аналізувати реальні явища та процеси, створювати власні продукти та рішення. Такі технології можуть бути: комп'ютерні програми, онлайн-сервіси, віртуальні лабораторії, симулятори, конструктори, робототехніка, датчики, сенсори, дрони, штучний інтелект, віртуальна та доповнена реальність тощо. Лише деякі дослідження розглядають можливості реалізації STEM-проектів на уроках інформатики у профільному навчанні, але не надають конкретних прикладів та методик їх впровадження.

## ВИСНОВКИ ДО РОЗДІЛУ 1

В Україні STEM-освіта — це система природничої і математичної освітніх галузей, яка має на меті розвиток особистості через формування компетентностей, природничо-наукової картини світу, світоглядних позицій і життєвих цінностей [23].

Для реалізації STEM-підходу в Україні була розроблена Концепція розвитку природничо-математичної освіти (STEM-освіти) до 2027 року, яка була ухвалена Урядом 5 серпня 2020 року. Концепція визначає основні принципи, цілі, завдання, напрями, механізми та очікувані результати розвитку STEM-освіти в Україні. Концепція також передбачає розроблення та затвердження державної програми розвитку STEM-освіти, яка має включати конкретні заходи, терміни, відповідальних виконавців, джерела фінансування та критерії оцінки ефективності.

У старшій школі програма навчання спрямована на практичне використання набутих умінь та поглиблення знань з усіх чотирьох складових STEM. Учні мають виконувати складніші та триваліші досліди, а також розробляти проєкти, які мають на меті вирішити актуальні для людства проблеми сучасності: розвитку альтернативних джерел енергії, зниження забруднення планети, глобального потепління, раціонального використання ресурсів та інші. Учні мають зосередитись переважно на можливостях, які відкриває STEM-освіта.

В рамках STEM-навчання завдання побудовані відповідно до наукового методу й інженерного дизайну, які є основою будь-якого процесу досліджень незалежно від галузі пізнання.

Науковий метод в STEM-освіті — це спосіб мислення та дослідження, який допомагає учням вивчати природні явища, вирішувати проблеми, перевіряти гіпотези, робити висновки та застосовувати знання в реальних ситуаціях.

Процес інженерного проєктування в STEM-освіті — це послідовність кроків, які допомагають учням розробляти та реалізовувати рішення для

реальних проблем за допомогою наукових, технологічних, інженерних та математичних знань та навичок.

Попередні дослідження з цієї теми переважно фокусувалися на реалізації STEM-підходу на уроках інформатики у початковій та середній школі, а також на використанні різних інтернет-ресурсів, сервісів та платформ для інтеграції природничих наук, технологій, інженерії та математики в навчальний процес.

Однак, на даний момент немає достатньо досліджень, які б розглядали реалізацію STEM-підходу на уроках інформатики у профільному навчанні, а також на використанні сучасних технологій. Лише деякі дослідження розглядають можливості реалізації STEM-проектів на уроках інформатики у профільному навчанні, але не надають конкретних прикладів та методик їх впровадження.

## **РОЗДІЛ 2. СТВОРЕННЯ ТА ТЕСТУВАННЯ ПРИСТРОЮ ДЛЯ ЛЮДЕЙ З ОСОБЛИВИМИ ПОТРЕБАМИ**

### **2.1 Використання апаратно-програмного комплексу Arduino для підтримки вивчення програмування у профільному навчанні**

Сучасні уроки програмування зазвичай розглядають лише поняття лінійного, розгалуженого і циклічного алгоритмів. В 10 класі також розглядаються масиви та способи їх впорядкування та алгоритми пошуку в масивах [18]. Проте не розглядається більш практичне значення програмування. При цьому розгляданні теми складових комп'ютера не приділяється багато уваги, оскільки не має змоги більш детально розглядати цю тему.

Одним із способів вирішення цієї проблеми є програмно-апаратний комплекс Arduino. Він є недороговартісним і має безліч можливостей як для конструювання так і для програмування. Він відкриває великі можливості для подальшого програмування мікроконтролерів, створення різних приладів та систем розумного будинку.

Програмно-апаратний комплекс Arduino вже завоював світовий ринок завдяки доступності і простоті у використанні. Наразі ця платформа є найпопулярнішою освітньою платформою електроніки і робототехніки.

До плюсів програмно-апаратний комплекс Arduino слід віднести що мова, якою програмується контролери Arduino — це видозмінений C++, а це означає що Arduino допоможе учням освоїти одну із затребуваних мов програмування. Також слід зазначити, що швидкість проектування і розробки на Arduino набагато вища ніж у інших мікроконтролерів, це обумовленою простою але в той же час добре проробленою архітектурою.

Також одним із плюсів Arduino є комплектуючі. Нині існує безліч периферійних пристроїв і датчиків, які можна підключити до Arduino, починаючи зі звичайних кнопок і закінчуючи рідкокристалічними екранами.

Крім того існує безліч дидактичного матеріалу, який можуть використовувати як учні так і вчителі. Це безліч підручників, різноманітні статті в мережі інтернет і канали в YouTube присвячені цій тематиці.

Вся робота з Arduino зводиться до трьох задач: підключити кабель, підключити компоненти і написати програму. При цьому з Arduino можна створювати різноманітні речі серед яких створення роботів, керування іграми, системи сигналізації, проєкти автоматизації систем будинку.

Програмування на Arduino простіше ніж написання мобільного додатку чи першої гри, а тому простіше і швидше можна продемонструвати учням результати роботи. А чим швидше і простіше, тим цікавіше це для учнів.

Програмування на Arduino можливо будь-якою мовою, проте для початку існує дуже зручне інтегроване середовище програмування Arduino IDE. Його версії можна знайти для Windows, macOS, GNU/Linux та інших операційних систем. IDE підтримує C і C++ і включає в себе бібліотеки для різноманітних апаратних компонентів. Коли програма закінчена, вона завантажується до плати за допомогою USB-кабеля.

Працюючи з Arduino, учні також будуть отримувати нові навички роботи з електронікою, в залежності від того, над яким проєктом вони працюють. Оскільки учні ніколи раніше не працювали з дротами і макетними платами, це буде безпечним і цікавим методом навчання.

Учні також мають змогу попрацювати з безліччю цікавих електронних компонентів, зрозуміють принцип їх роботи. Коли учні самостійно зможуть збирати схеми і писати програмний код, то відкриють для себе великий творчий потенціал.

Як противагу Arduino було розглянуто платформу Micro:bit. Вона має такі самі переваги, що і Arduino, проте програмування відбувається на Python. Тому було створено опитування для вчителів, з метою встановлення кількості платформ Arduino і Micro:bit в закладах освіти.

Окрім Arduino та Micro:bit було розглянуто NodeMCU, Raspberry Pi, ESP32, STM32, Mindstorms, проте вони не є такими ж популярними, або взагалі відсутні в закладах освіти.

За результатами опитування було виявлено, що кількість наборів Micro:bit становить 6 штук, кількість наборів Arduino 10 штук. Тому було прийнято рішення використовувати платформу Arduino.

## 2.2 Пристрої та їх основні принципи роботи для створення пристрою для людей з особливими потребами

Смарт-пристрій для людей з вадами зору на базі програмно-апаратного комплексу Arduino, складається з таких елементів (таблиця 2.1).

Таблиця 2.1

### Перелік елементів для створення смарт-пристроїв для людей з особливими потребами на базі ПМК Arduino

№	Назва	Кількість, шт.	Вартість, грн.
1	Плата Arduino Uno	1	415
2	Ультразвуковий датчик відстані HC SR04	3	49
3	Плоский вібромотор	3	18
4	Проводи з'єднувальні	30	25
<b>Загальна вартість:</b>			641

Основою в смарт-пристрої є плата Arduino Uno, три ультразвукові датчики та 3 вібромотори. Розглянемо як вони працюють.

**Arduino** — це інструмент для створення електронних пристроїв (електронний конструктор), які взаємодіють з навколишнім фізичним середовищем. Програмна частина складається з безкоштовного програмного середовища (IDE) для написання та компіляції коду, а також завантаження його до мікроконтролера. Апаратна частина складається з вироблених друкованих плат, які можна придбати як у офіційного виробника, так і іншими

виробниками. Повністю відкриті програмний код і будова плат комплексу дозволяє вільно копіювати або поповнювати лінійку продукції Arduino [35].

Arduino надає можливість створювати автономні об'єкти автоматички або підключати їх до програмного забезпечення на комп'ютері за допомогою різних інтерфейсів. Arduino не має стандартного корпусу або монтажної конструкції. Розробник сам вирішує, як встановлювати і захищати плати. Існують набори робототехнічної електромеханіки від сторонніх виробників, які призначені для роботи з платами Arduino.

Характерною особливістю Arduino Uno (рис. 1) є конвертер USB-UART на базі ATmega16U2, в попередніх версіях плати використовувались чіпи 8U2 і FTDI.

Основні характеристики:

- Мікроконтролер: ATmega328;
- Робоча напруга: 5V;
- Вхідна напруга: (рекомендується) 7-12V;
- Вхідна напруга: (межі) 6-20V;
- Цифрові вводи / виводи: 14 (6 з яких забезпечить вихід ШІМ);
- Аналогові входи: 6;
- Постійний струм в лінії введення / виводу: 40 мА;
- Постійний струм на 3,3V Pin: 50 мА;
- Флеш-пам'ять: 32 Кб (ATmega328), з яких 0,5 Кб використовуються завантажувачем;
- SRAM 2 Кб (ATmega328);
- EEPROM 1 Кб (ATmega328);
- Тактова частота 16 МГц [37].





Рис. 2.1. Зовнішній вигляд плати Arduino Uno

**Ультразвуковий датчик.** Ультразвуковий датчик вимірює відстань до об'єкта за допомогою принципу ехолокації — відправляючи ультразвукову хвилю, і отримуючи її відбиття через певний час, пристрій виявляє наявність об'єктів і відстань до них. Ультразвукові хвилі, які створює приймач, повертаються до нього після відбивання від перешкоди через певний час. За допомогою цього проміжку часу і визначається відстань до об'єкта. (рис.2.2.) [37].

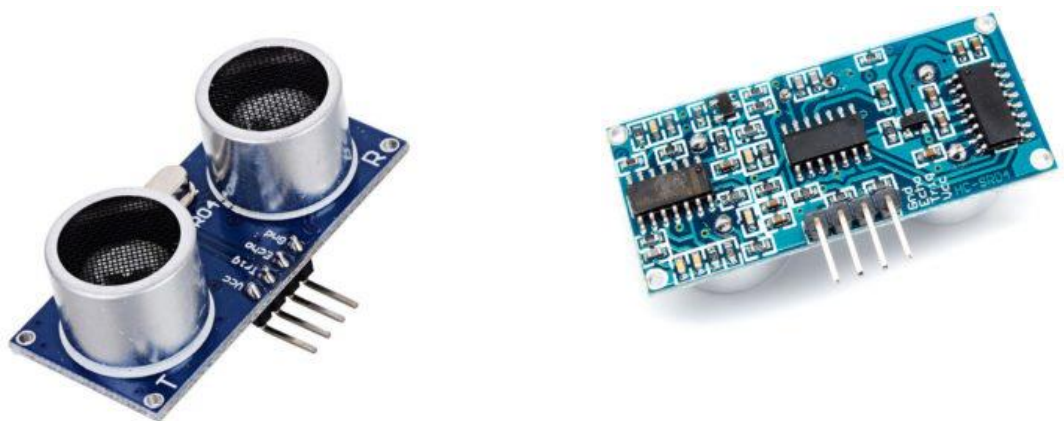


Рис. 2.2. Зовнішній вигляд ультразвукового датчика HC SR04

Arduino датчик відстані — це прилад безконтактного типу, який забезпечує точне вимірювання і стабільність. Він може вимірювати відстань

від 2 до 400 см. Електромагнітні випромінювання і сонячна енергія не впливають на його роботу. Модуль з HC SR04 Arduino містить також ресивер і трансмітер.

Технічні параметри ультразвукового далекоміру HC SR04 представлені в таблиці 2.2.

Таблиця 2.2

### Основні параметри ультразвукового датчика HC SR04

№	Назва	Параметр
1	Напруга живлення	5В
2	Робочий параметр сили струму	15 мА
3	Сила струму в пасивному стані	2 мА
4	Оглядний кут	15°
5	Сенсорний дозвіл	0,3 см
6	Вимірювальний кут	30°
7	Ширина імпульсу	10-6 с

Датчик має чотири виходи (стандарт 2, 54 мм) (рис.2.2.)

Контакт позитивного живлення – +5 В;

Trig (Т) – вхідний сигнал;

Echo (R) – вихідний сигнал;

GND – «Земля».

Для вимірювання відстані, потрібно виконати такі кроки:

1. Надіслати на вихід Trig імпульс тривалістю 10 мікросекунд;
2. У ультразвуковому датчику HC SR04, підключеному до Arduino, сигнал перетвориться на 8 імпульсів з частотою 40 кГц, які через випромінювач будуть відправлені вперед;
3. Коли сигнали дістануться до перешкоди, вони відбиваються від неї і їх ловить приймач, що створює вхідний сигнал виході Echo;
4. На самому контролері отриманий сигнал переводимо у відстань між датчиком і перешкодою за допомогою фізичної формули [37].

Щоб отримати дані в сантиметрах, потрібно поділити ширину імпульсу на 58.2, а щоб отримати дані в дюймах, потрібно поділити ширину імпульсу на 148 (рис.2.3).

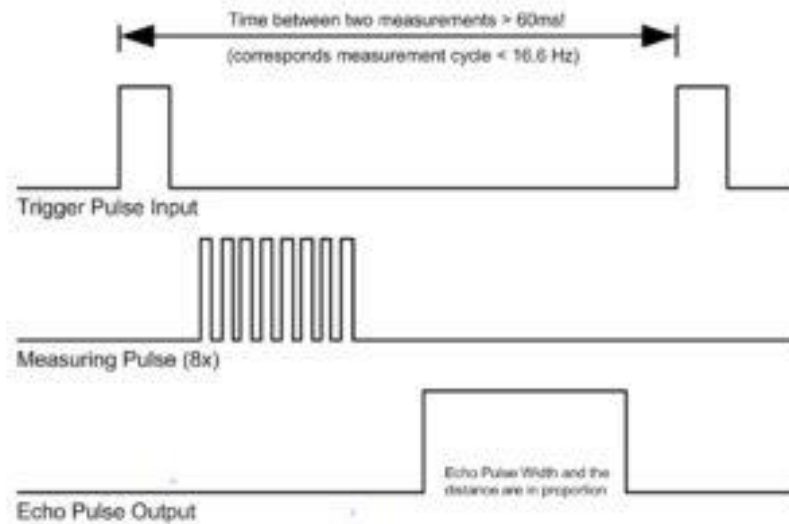


Рис. 2.3. Схема роботи ультразвукового датчика HC SR04

Стандартний спосіб з'єднання ультразвукового датчика з платою демонструється на рисунку 2.4.



а) без макетної плати

б) за допомогою макетної плати

Рис. 2.4. Підключення ультразвукового датчика HC SR04 до плати

Для з'єднання землі і живлення з Arduino використовуємо виводи GND і 5V відповідно. Виходи Trig і Echo ультразвукового датчика підключаються до цифрових Пінів Arduino.

Для роботи з датчиком HC SR0 існує бібліотека NewPing, яка вирішує проблеми зі стандартними пінговими функціями і надає додаткові функції. До особливостей бібліотеки входять:

- Сумісність з різними ультразвуковими датчиками;
- Можливість використовувати один пін для роботи з датчиком;
- Усунення затримки на 1 секунду при відсутності відбитого сигналу;
- Вбудований цифровий фільтр для зменшення помилок вимірювань;
- Найбільш точний розрахунок відстані [37].

Будь-який ультразвуковий датчик працює на принципі відбивання акустичних хвиль, які розповсюджуються в повітрі. Але, як відомо з фізики, швидкість розповсюдження звуку в повітрі залежить від його властивостей (переважно від температури). Датчик, випускаючи хвилі і вимірюючи час до їх повернення, не знає, в якому середовищі вони будуть розповсюджуватися і бере для розрахунків деяке середнє значення. В реальних умовах через фактор температури повітря HC-SR04 може давати похибку від 1 до 3-5 см. Фактор відстані до об'єкта важливий, тому що збільшується ймовірність відбиття від інших предметів, а також сигнал слабшає з відстанню. Також для підвищення точності треба правильно спрямувати датчик: зробити так, щоб предмет був в межах конуса діаграми спрямованості. Іншими словами, датчики HC-SR04 повинні дивитися безпосередньо на предмет.

Давайте розглянемо принцип роботи пристрою.

Коротким імпульсом (2-5 мікросекунди) використовуючи ультразвуковий датчик, використовуємо принцип ехолокації, при якому в простір відправляємо ультразвукові хвилі з частотою 40 КГц.

Отримавши відбитий сигнал ультразвуковий датчик відстані передасть на вхід ЕСНО імпульс, пропорційний відстані. Ми визначаємо тривалість імпульсу за допомогою функції `pulseIn`.

Отримавши час, ми переводимо його в відстань в сантиметрах шляхом ділення значення на константу (для датчика SR04 це 29.1 в одному напрямку, врахувавши обидва напрямки отримаємо 58.2).

Якщо датчик не отримає відбитий імпульс, то і не передасть на вхід ЕСНО імпульс, а відповідно рівень сигналу буде — LOW. Рекомендується при використанні скетчів виставляти його значення вручну.

НС SR04 починає погано працювати при відстані більше 3 метрів, тому час затримки краще виставляти більше 20 мс, тобто 25 або 30 мс.

До складу пристрою входять 3 плоских віброметри (рис. 2.5).



Рис. 2.5. Зовнішній вигляд віброметру

Основні технічні параметри віброметру:

- Діаметр: 10 мм;
- Висота: 3 мм;
- Напруга живлення: 3 В;
- Клейка стрічка на зворотному боці;
- 2 дроти довжиною 20 мм [37].

Віброметор можна під'єднати до пінів виходу і працювати в двох режимах: аналоговому та цифровому. При аналоговому з'єднанні ми можемо керувати силою вібрацію, на відміну від цифрового. У випадку цифрового опрацювання ми можемо отримати два стани: є і не має вібрації.

### 2.3 Програмна та фізична реалізація пристрою

В результаті етапу проектування отримано наступний зовнішній вигляд пристрою (рис.6.).

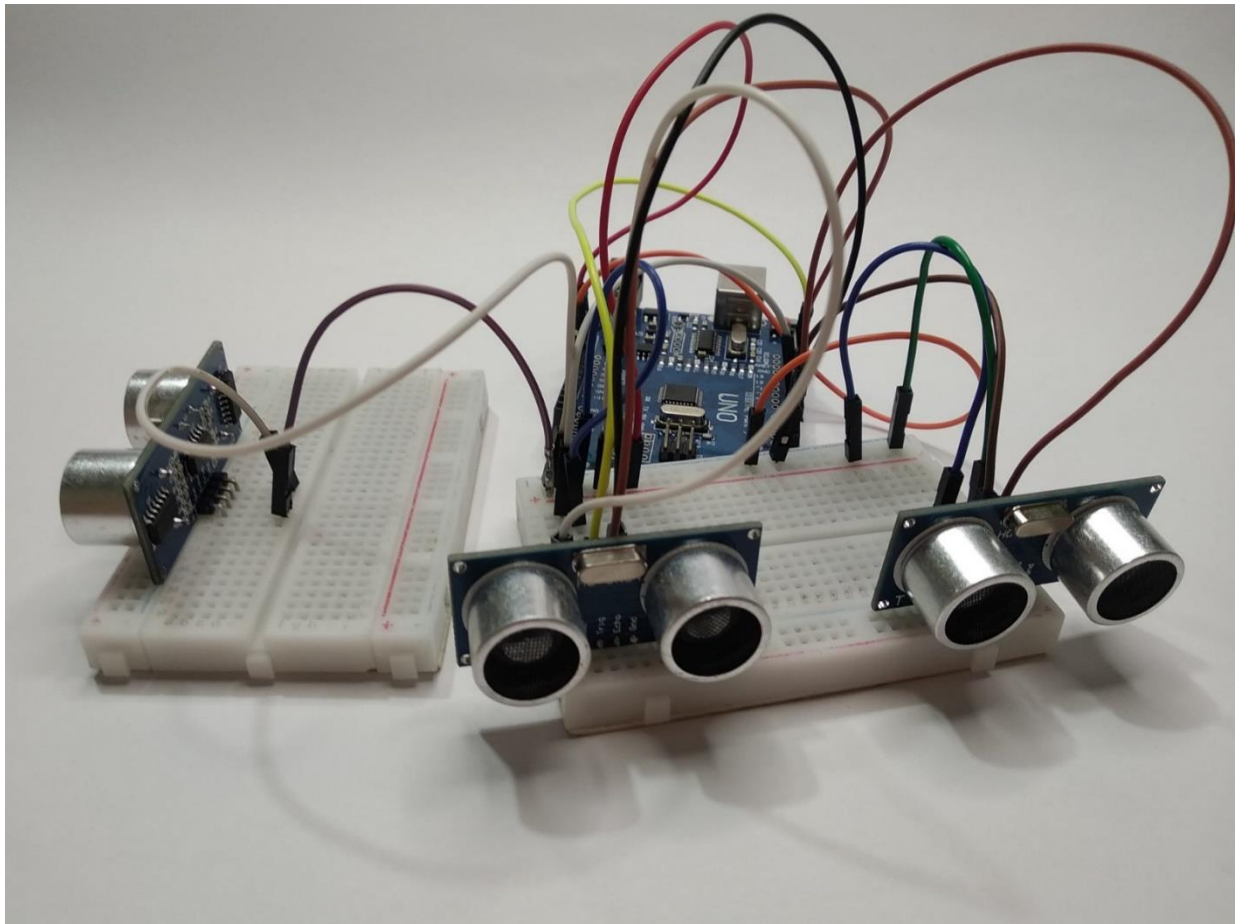


Рис 2.6. Зовнішній вигляд пристрою на проміжному етапі  
Після етапу проєктування на етапі реалізації було написано програму для отримання вібросигналів при наближенні до предмету-перешкоди.

```
#define PIN_TRIG 12
#define PIN_ECHO 11
#define PIN_VIBRO 6
#define PIN_TRIGL 3
#define PIN_ECHOL 2
#define PIN_TRIGR 5
#define PIN_ECHOR 4
long duration, cm, durationL, cmL, durationL, cmL;
void setup() {
    Serial.begin (9600); // Відкриття порту
    pinMode(PIN_TRIG, OUTPUT);
    pinMode(PIN_ECHO, INPUT);
```

```

pinMode(PIN_VIBRO, OUTPUT);
pinMode(PIN_TRIGL, OUTPUT);
pinMode(PIN_ECHOL, INPUT);
pinMode(PIN_TRIGR, OUTPUT);
pinMode(PIN_ECHOR, INPUT);
}
void loop() {
    // Генеруємо короткий імпульс 2-5 мікросекунд.
    digitalWrite(PIN_TRIG, LOW);
    delayMicroseconds(5);
    digitalWrite(PIN_TRIG, HIGH);
    // Встановивши високий рівень сигналу, чекаємо 10 мікросекунд.
    // У цей момент датчик відправляє сигнали з частотою 40 КГц.
    delayMicroseconds(10);
    digitalWrite(PIN_TRIG, LOW);

    // Одержуємо час затримки сигналу з приймача.
    duration = pulseIn(PIN_ECHO, HIGH);
    // Розраховуємо відстань (1 / швидк.звуку = 1 / 0.03435 см/мс = 29.1
мс/см)
    cm = (duration / 2) / 29.1;
    Serial.print("Відстань до об'єкту: ");
    Serial.print(cm);
    Serial.println(" см.");
    delay(250); // Пауза між вимірюваннями
    //if (cm <20) analogWrite(PIN_VIBRO, 100); else
analogWrite(PIN_VIBRO, 0);
    //if (cm<20) analogWrite(PIN_VIBRO,255-cm*10);else
analogWrite(PIN_VIBRO,0);
    // робимо аналогічні дії для лівого датчика

```

```
digitalWrite(PIN_TRIGL, LOW);
delayMicroseconds(5);
digitalWrite(PIN_TRIGL, HIGH);
delayMicroseconds(10);
digitalWrite(PIN_TRIGL, LOW);
durationL = pulseIn(PIN_ECHOL, HIGH);
cmL= (durationL / 2) / 29.1;
Serial.print("Відстань до об'єкту: ");
Serial.print(cmL);
Serial.println(" см.");
delay(250);// Пауза між вимірюваннями
// робимо аналогічні дії для правого датчика
digitalWrite(PIN_TRIGR, LOW);
delayMicroseconds(5);
digitalWrite(PIN_TRIGR, HIGH);
delayMicroseconds(10);
digitalWrite(PIN_TRIGR, LOW);
durationR = pulseIn(PIN_ECHOR, HIGH);
cmR= (durationR / 2) / 29.1;
Serial.print("Відстань до об'єкту: ");
Serial.print(cmR);
Serial.println(" см.");
delay(250);// Пауза між вимірюваннями
}
```



## ВИСНОВКИ ДО РОЗДІЛУ 2

Для створення навчального проєкту було розглянуто платформи Arduino і Micro:bit. Також розглядалися платформи NodeMCU, Raspberry Pi, ESP32, STM32, Mindstorms, проте вони не є такими ж популярними, або взагалі відсутні в закладах освіти. Серед вказаних платформ є ряд переваг:

- Низька вартість.
- Відкриті платформи.
- Наявність великої кількості бібліотек.
- Велика і активна спільнота, що дозволяє легко створювати проєкти.

Для того, щоб визначити, яку з платформ буде доцільніше використати, було проведення опитування серед вчителів інформатики щодо наявності платформ Arduino і Micro:bit в закладах освіти. Як виявилось платформа Arduino більш поширена, тому було вирішено використовувати саме її.

Розроблений смарт-пристрій для людей з особливими потребами на базі програмно-апаратного комплексу Arduino, складається з наступних елементів: плата Arduino Uno, три ультразвукові датчики HC SR04, три плоскі вібротвори. Загальна вартість смарт-пристрою 641 грн.

Arduino Uno — це пристрій на базі мікроконтролера ATmega328P. До його складу входить все необхідне для зручної роботи з мікроконтролером та програмування розробленого пристрою.

Ультразвуковий датчик відстані — це прилад безконтактного типу, що забезпечує високоточне вимірювання і стабільність. Діапазон дальності його вимірювання складає від 2 до 400 см. Здатність ультразвукового датчика визначати відстань до об'єкта засновано на принципі ехолокації – посилаючи ультразвукову хвилю, і отримуючи її відображення із певною затримкою, пристрій визначає наявність об'єктів і відстань до них.

Вібротвори використовуються в приладі для попередження про наявність перешкоди. Їх можна під'єднати до виводу і працювати в двох

режимах: аналоговому та цифровому. При аналоговому з'єднанні ми можемо керувати силою вібрацію, на відміну від цифрового.

## РОЗДІЛ 3. МЕТОДИКА STEM-ПІДХОДУ ПРИ ВИКЛАДАННІ ІНФОРМАТИКИ У ПРОФІЛЬНОМУ НАВЧАННІ

### 3.1 Практичне використання STEM –підходу на уроках з інформатики у профільному навчанні

#### STEM-проект «Пристрій-помічник для людей з вадами зору»

Пропоную приклад реалізації STEM-проекту для учнів 10-х класів на тему «Пристрій-помічник для людей з вадами зору», який поєднує вивчення та узагальнення знань із таких навчальних дисциплін, як інформатика, фізика та математика, окрім цього він поглиблює знання учнів з інженерії.

STEM-проект розрахований на заміну урокам із змістовного модуля «Мова програмування і структури даних» для профільного класу. Він включає в себе наступні теми:

1. Особливості середовища розробки.
2. Основні елементи мови програмування.
3. Оператори і вирази.
4. Логічні вирази. Таблиці істинності. Умовний оператор мовою програмування. Оператор вибору.
5. Послідовні та вкладені розгалуження.
6. Тестування розгалужених алгоритмів.
7. Оператори циклу.
8. Порівняння операторів циклу.
9. Покрокове виконання програм із циклами і розгалуженнями.

*Тип проекту:* навчальний, груповий, інтегрований (інформатика, фізика, математика).

*Мета:* удосконалити уміння десятикласників із добору й обробки інформації, її аналізу та систематизації; набути й узагальнити знання, уміння і навички роботи з платою Arduino Uno та мови програмування C++; усвідомити роль природничо-математичних наук у сучасному світі.

Також створені уроки можна проводити в дистанційному форматі з використанням додатку Tinkercad.

Таблиця 3.1

Етап реалізації	Діяльність педагога	Діяльність учнів
Визначення теми проєкту, його типу, кількості учасників	Тема проєкту «Пристрій-помічник для людей з вадами зору» Тип: навчально-пізнавальний, короткотривалий Кількість учасників: 15 (14–17 років)	—
Визначення проблеми в межах обраної теми	Програмно-апаратний комплекс Arduino в ролі керуючого пристроєм-помічником для людей з вадами зору	Ознайомлення з темою дослідження
Формулювання задач роботи	Формулювання попередніх задач: <ul style="list-style-type: none"> <li>• встановити можливості програмно-апаратного комплексу Arduino;</li> <li>• ознайомитись з мовою програмування C++;</li> <li>• ознайомитись з принципом збирання схем;</li> <li>• ознайомитись з принципом роботи, підключенням та програмування роботи ультразвукового датчика, вібромотору</li> </ul>	Ознайомлення з попередніми задачами

Формування умінь і навичок роботи з платою	Ознайомлення учнів з базовим теоретичним матеріалом, контроль роботи учнів, консультування учнів	Ознайомлення з базовим теоретичним матеріалом, формування умінь і навичок роботи з платою
Самостійна робота	Контроль роботи учнів, консультування учнів, оцінювання учнів	Самостійна, поетапна розробка схеми та програми для керування екосистемою за допомогою наявних пристроїв, демонстрація результатів

Проект відбувається в 2 етапи: формування умінь і навичок роботи з платою Arduino Uno, створення готового продукту.

Перший етап розбито на 8 уроків, під час яких учні ознайомляться з платою Arduino Uno, ультразвуковим датчиком, вібромотором, кнопкою потенціометром.

Другий етап є заключним, під час його проведення учні на основі отриманих знань, умінь і навичок формують готовий продукт у вигляді пристрою-помічника для людей з вадами зору.

Під час використання цього проекту слід зробити акцент на ехолокації та вимірюванні відстані за допомогою ультразвуку.

Розроблені в рамках виконання кваліфікаційної роботи уроки можна переглянути в Додатку А.

### **3.2 Оцінка результатів впровадження STEM-підходу у профільному навчанні інформатики**

На жаль впровадити розроблений STEM-проект в шкільну програму не має змоги, оскільки ліцеї не готові до випробування нових проєктів, які займають багато часу під час уроків через сучасні реалії війни. Проте впровадити розроблений проєкт можна не лише в шкільну програму з

інформатики для профільного рівня, а й в приватні школи, які займаються навчанням дітей програмуванню. Тому було проаналізовано можливі приватні школи і їх програми з навчання дітей.

Таку змогу надає Програма профільного рівня для 10-11 класів профільного рівня з предмету Інформатика, оскільки в програмі вказано, що вчитель самостійно визначає кількість навчальних годин на вивчення того чи іншого розділу (теми) програми та порядок вивчення тем.

Розроблений проєкт розрахований на профільний рівень 10 класу для модуля «Мова програмування та структури даних», тому серед критеріїв пошуку були:

1. Вік учнів від 14 до 17 років.
2. Програма навчання має містити розділи пов'язані з наступними темами: «Особливості середовища розробки», «Типи даних», «Лінійні алгоритми», «Розгалужені алгоритми» та «Циклічні алгоритми».
3. Наявність принаймні двох груп, які навчаються за однаковою програмою.

З переглянутих приватних шкіл була обрана «Міжнародна школа програмування та англійської мови «Logika»». В обраній школі є курс з розробки на мові програмування Python, який переходить в розробку комп'ютерних ігор.

Розглянемо уроки, які входять до їх програми:

1. Знайомство з середовищем розробки.
2. Змінні.
3. Вирази. Типи даних.
4. Гра Несінитниця.
5. Умовний оператор.
6. Вкладені умовні оператори.
7. Превірка знань. Цикли.
8. Цикли продовження.

Для впровадження проєкту було обрано дві групи. Перша група займалась за програмою від школи, друга — за розробленим проєктом. Кількість учнів у групах — дев'ять дітей.

Для можливості оцінки результатів навчання було створено 3 тести. Ці тести учні з обох груп мали пройти перед початком навчання і після завершення уроків проєкту.

Переглянемо результати з обох груп до та після навчання.

Таблиця 3.2

№ питання	Кількість правильних відповідей до участі у проєкті		Кількість правильних відповідей після участі у проєкті	
	1 група	2 група	1 група	2 група
Урок № 3. Оператори і вирази.				
1	7	6	9	9
2	9	9	9	9
3	6	7	9	9
4	6	8	8	9
5	7	7	9	9
6	8	7	7	8
7	9	9	9	9
8	9	9	8	8
9	4	3	8	7
10	3	3	7	8
Всього:	68	68	83	85
Урок № 6. Тестування розгалужених алгоритмів.				
1	7	6	9	9
2	7	8	9	9
3	8	9	8	9
4	9	9	8	9
5	9	9	9	9

6	8	7	9	9
7	9	9	9	8
8	9	9	8	9
9	9	7	8	9
10	6	9	8	9
11	7	6	8	7
Всього:	88	88	93	96

Урок № 8. Порівняння операторів циклу.

1	3	5	8	9
2	4	3	8	9
3	4	6	9	9
4	5	4	9	8
5	5	3	7	8
6	6	7	9	8
7	7	6	6	8
8	1	3	5	5
9	1	1	6	6
10	2	1	6	5
Всього:	38	39	73	75

Як бачимо з таблиці, то результати обох груп майже однакові, різниця в 1-2 правильних відповіді. Це демонструє, що обидва підходи однаково ефективні. Проте, якщо звернути увагу на те, що друга група окрім вивчення мови програмування, також отримала знання з інженерії, а отже учні змогли отримати більше знань, умінь і навичок саме під час навчання за програмою STEM-проєкту.



## ВИСНОВКИ ДО РОЗДІЛУ 3

Було розроблено STEM-проект для учнів 10-х класів на тему «Пристрій-помічник для людей з вадами зору», який поєднує вивчення та узагальнення знань із таких навчальних дисциплін, як інформатика, фізика та математика, окрім цього він поглиблює знання учнів з інженерії.

STEM-проект розрахований на заміну урокам із змістовного модуля «Мова програмування і структури даних» для профільного класу. Він включає в себе наступні теми:

1. Особливості середовища розробки.
2. Основні елементи мови програмування.
3. Оператори і вирази.
4. Логічні вирази. Таблиці істинності. Умовний оператор мовою програмування. Оператор вибору.
5. Послідовні та вкладені розгалуження.
6. Тестування розгалужених алгоритмів.
7. Оператори циклу.
8. Порівняння операторів циклу.
9. Покрокове виконання програм із циклами і розгалуженнями.

Впровадження проекту було реалізовано в приватній школі «Міжнародна школа програмування та англійської мови «Logika»». В обраній школі є курс з розробки на мові програмування Python.

Для впровадження проекту було обрано дві групи. Перша група займалась за програмою від школи, друга — за розробленим проектом.

Результати тестування двох груп продемонстрували, що обидва підходи однаково ефективні, проте друга група окрім вивчення мови програмування, також отримала знання з інженерії, а отже учні змогли отримати більше знань, умінь і навичок саме під час навчання за програмою STEM-проекту.

## ВИСНОВКИ

Одним із напрямків модернізації природничо-математичної освіти в Україні є впровадження STEM-підходу, який інтегрує знання та навички з науки, технологій, інженерії та математики. Для цього було розроблено та схвалено Кабінетом Міністрів Концепцію розвитку STEM-освіти до 2027 року, яка окреслює основні засади, мету, завдання, напрямки, механізми та очікувані результати впровадження STEM-освіти в Україні. Згідно з Концепцією, має бути розроблена та затверджена державна програма розвитку STEM-освіти, яка має містити специфічні заходи, терміни, відповідальних виконавців, джерела фінансування та критерії оцінки ефективності.

У профільному навчанні старшокласників програма навчання передбачає практичне застосування отриманих умінь та поглиблення знань з усіх чотирьох компонентів STEM. Учні мають реалізовувати більш складні та довготривалі експерименти, а також розробляти проєкти, які мають на меті вирішити актуальні для людства проблеми сучасності.

Наразі існує недостатньо досліджень, які б аналізували реалізацію STEM-підходу на уроках інформатики у профільному навчанні, а також на використанні сучасних технологій. Лише окремі дослідження розглядають можливості реалізації STEM-проєктів на уроках інформатики у профільному навчанні, але не надають конкретних прикладів та методик їх впровадження.

Для створення навчального проєкту було розглянуто платформи Arduino і Micro:bit. Окрім Arduino та Micro:bit було розглянуто NodeMCU, Raspberry Pi, ESP32, STM32, Mindstorms, проте вони не є такими ж популярними, або взагалі відсутні в закладах освіти. Серед вказаних платформ є ряд переваг:

- Низька вартість.
- Відкриті платформи.
- Наявність великої кількості бібліотек.
- Велика і активна спільнота, що дозволяє легко створювати проєкти.

Проте була обрана саме Arduino, оскільки вона більш поширена.

Окрім Arduino та Micro:bit було розглянуто NodeMCU, Raspberry Pi, ESP32, STM32, Mindstorms, проте вони не є такими ж популярними, або взагалі відсутні в закладах освіти.

Розроблений смарт-пристрій для людей з особливими потребами на базі програмно-апаратного комплексу Arduino, складається з наступних елементів: плата Arduino Uno, три ультразвукові датчики HC SR04, три плоскі вібромотори. Загальна вартість смарт-пристрою 641 грн.

Розроблений STEM-проєкт для учнів 10-х класів на тему «Пристрій-помічник для людей з вадами зору» поєднує вивчення та узагальнення знань із таких навчальних дисциплін, як інформатика, фізика та математика, окрім цього він поглиблює знання учнів з інженерії.

STEM-проєкт розрахований на заміну урокам із змістовного модуля «Мова програмування і структури даних» для профільного класу. Проєкт включає в себе 9 тем.

Впровадження проєкту було реалізовано в приватній школі «Міжнародна школа програмування та англійської мови «Logika»».

Результати тестування продемонстрували, що запропонований STEM-проєкт є ефективним, оскільки окрім вивчення мови програмування учні також отримала знання з інженерії, а отже учні змогли отримати більше знань, умінь і навичок саме під час навчання за програмою STEM-проєкту.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Education Thematic Brief «Closing the gender gap in STEM», August 2016. — URL: <https://www.pedrada.com.ua/article/1401-shcho-take-stem-osvta-u-navchalnomu-zaklad>.
2. Heidi Kleinbach-Sauter. STEM 2.0: An Imperative for Our Future Workforce // Heidi
3. International Journal of STEM Education 2017, March 2017.
4. Kleinbach-Sauter, Edie Fraser [Електронний ресурс]. / Режим доступу: <https://www.stemconnector.com/wp-content/uploads/2016/12/STEM-2pt0-Publication-2nd-Edition-1.pdf>.
5. Lila Carly Gilbreath. Factors Impacting Women’s Participation in STEM Fields, 2015.
6. STEM-ОСВІТА: НАУКОВО-ПРАКТИЧНІ АСПЕКТИ ТА ПЕРСПЕКТИВИ РОЗВИТКУ СУЧАСНОЇ СИСТЕМИ ОСВІТИ 18 жовтня – 26 листопада 2021 року Видавничий дім «Гельветика» 2021.
7. Азізов Р. Образование нового поколения: 10 преимуществ STEM образования [Електронний ресурс]. / Руфат Азізов — Режим доступу до ресурсу: <https://ru.linkedin.com/pulse>.
8. Альтшуллер Г. Найти идею. введение в теорию решения изобретательских задач. / Г. Альтшуллер. — Петрозаводск, — 2003 г., — С. 173–185.
9. Балик Н. Р. Підходи та особливості сучасної STEM-освіти / Н. Р. Балик, Г. П. Шмигер // Фізикоматематична освіта. – 2017. – № 2(12). – С. 26–30.
10. Барна О. В. Впровадження STEM-освіти у навчальних закладах: етапи та моделі / О. В. Барна, Н. Р. Балик // STEM-освіта та шляхи її впровадження в навчально-виховний процес: збірник матеріалів І регіональної науково-практичної веб-конференції, Тернопіль, 24 травня 2017 р. – Тернопіль: ТОКІШПО, 2017. – С. 3–8.

11. Відділ STEM-освіти [Електронний ресурс]. // Інститут модернізації змісту освіти. — Режим доступу до ресурсу: <https://imzo.gov.ua/pro-imzo/struktura/viddil-stem-osviti/>.
12. Демківський А. В. Основи методології наукових досліджень: [навч. посіб.] / А. В. Демківський, П. І. Безус. — Київ : Акад. муніцип. упр., 2012. — 276 с.
13. Для незрячих та слабозрячих. URL: <https://inkluzia.com.ua/meditsinskoe-i-korreksionnoe-oborudovanie/dlya-nezryachikh-i-slabovidyashchikh/>.
14. Доценко С. О. Прийоми активізації творчої діяльності учнів в умовах STEM-освіти / С. О. Доценко // Професійна освіта: методологія, теорія та технології. — 2016. — Вип. 4. — С. 32–46.
15. Доценко С. О. STEM-освіта як засіб активізації творчого потенціалу особистості [Електронний ресурс] / С. О. Доценко, В. В. Лебедева. — Режим доступу: <http://www.dgma.donetsk.ua/docs/konf/2017/mkonf2017/dopovidy/it/Доценко-Лебедева.pdf>.
16. Закон України «Про освіту» [Електронний ресурс]. / Режим доступу: <http://zakon3.rada.gov.ua/laws/show/2145-19>.
17. Інститут модернізації змісту освіти. Режим доступу: <https://imzo.gov.ua/tag/stem-osvita/>.
18. ІНФОРМАТИКА. Програма для 10-11-х класів ЗНЗ. — URL: <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-dlya-10-11-klasiv>
19. Кириленко С. Поліфункціональний урок у системі STEM-освіти: теоретико-методологічні та методичні сегменти./С.Кириленко,О.Кіян//Рідна школа.-2016-№4-с.50-54.
20. Коваленко О. STEM- освіта: досвід упровадження в країнах ЄС та США ./О.Коваленко, О.Сапрунова.//Рідна школа.-2016-№4-с.46-49.
21. Комплексний хмарний CAD/CAE/CAM інструмент. URL: <https://www.autodesk.com/products/fusion-360>.

22. Концепція нової української школи [Електронний ресурс]. / Режим доступу:  
<http://mon.gov.ua/%D0%9D%D0%BE%D0%B2%D0%B8%D0%BD%D0%B8%202016/12/05/konczepczya.pdf>.
23. Концепція розвитку природничо-математичної освіти (STEM-освіти) – URL: <https://zakon.rada.gov.ua/laws/show/960-2020-p#n8>
24. Мартинюк І. Творчий потенціал і самореалізація особистості // Психологія і педагогіка життєтворчості. — К., 1996. — 792 с
25. Методичні рекомендації щодо впровадження STEM-освіти у загальноосвітніх та позашкільних навчальних закладах України на 2017-2018 навчальний рік.(Лист ІЗМО № 21. 1/10-1470 від 13.07.17 року).
26. Методичні рекомендації щодо розвитку STEM-освіти в закладах загальної середньої та позашкільної освіти у 2019/2020 навчальному році : лист ІМЗО № 22.1/10- 2876 від 22 серп. 2019 р. – URL: [https://osvita.ua/legislation/Ser\\_osv/65463](https://osvita.ua/legislation/Ser_osv/65463).
27. Методичні рекомендації щодо розвитку STEM-освіти в закладах загальної середньої та позашкільної освіти у 2020/2021 навчальному році: лист ІМЗО від 19.08.2020 № 22.1/10-1646
28. Методичні рекомендації щодо розвитку STEM-освіти в закладах загальної середньої та позашкільної освіти у 2021/2022 навчальному році: лист ІМЗО від 11.08.2021 № 22.1/10-1775
29. Навчальні програми для загальноосвітніх навчальних закладів України + опис ключових змін. Математика. Інформатика. 5-9 класи.-К.: Вид. дім «Освіта», -2017-56 с.
30. Патрикєєва О. STEM-освіта: умови впровадження у навчальних закладах України / О. Патрикєєва, О. Лозова, С. Горбенко // Управління освітою. – 2017. – № 1. – С. 28–31.
31. Поліхун Н. І., Сліпучіна І. А., Чернецький І. С. Педагогічна технологія STEM як засіб реформування освітньої системи України / Н. І.

Поліхун, І. А. Сліпухіна, І. С. Чернецький // Освіта та розвиток обдарованої особистості. – 2017. – № 3(58). – С. 5–9.

32. Применко Л.Л. Використання технології проєктів./Л.Л. Применко//Педагогічна Житомирщина.-2009-№1(49)-с.68-70.

33. Пристрої для інвалідів. URL: <http://ipkey.com.ua/uk/teg-disabilities-devices.html>.

34. Проблеми інвалідів важливі для суспільства. URL: <https://ukurier.gov.ua/uk/articles/problemi-invalidiv-vazhlivi-dlya-suspilstva/>

35. Програма обміну для студентів вищих навчальних закладів (Global UGRAD) з додатковим компонентом «STEM та підприємництво» — URL: <https://ua.usembassy.gov/uk/education-culture-uk/exchange-programs-uk/academic-exchanges-uk/global-ugrad-uk/>

36. Проєкт Концепції STEM – освіти в Україні[Електронний ресурс].mk-kor.at.ua/STEM/ STEM\_2017.pdf

37. Розробка смарт-пристрою для людей з особливими потребами на основі програмно-апаратного комплексу Arduino URL: [https://ontu.edu.ua/download/konfi/2021/all-ukrainian\\_student\\_scientific\\_works\\_ce/Works-of-winners-VKSNR-2021.pdf](https://ontu.edu.ua/download/konfi/2021/all-ukrainian_student_scientific_works_ce/Works-of-winners-VKSNR-2021.pdf)

38. Соціально-педагогічна робота з дітьми та молоддю з функціональними обмеженнями. URL: [http://www.ebooktime.net/book\\_156\\_glava\\_91](http://www.ebooktime.net/book_156_glava_91).

39. Стрижак О. Є., Сліпухіна І. А., Поліхун Н. І. STEM-освіта: основні дефініції / О. Є. Стрижак, І. А. Сліпухіна, Н. І. Поліхун, І. С. Чернецький // Інформаційні технології і засоби навчання. – 2017. – Т. 62. – № 6. – С. 16–33. – URL: <https://journal.iitta.gov.ua/index.php/itlt/article/view/1753/1276>.

40. Трушкова Т.С. Використання міжпредметних зв'язків для розвитку пізнавальних інтересів учнів./Т.С. Трушкова//Фізика в школах України. Позакласна робота.-2016-№2-с.23-28.

41. Трушкова Т.С. Міжпредметні зв'язки та розвиток пізнавальних інтересів учнів./С.Т. Трушкова//Фізика в школах України.-2016-№13-14-с.35-39.
42. Упровадження STEM-освіти в умовах інтеграції формальної і неформальної освіти обдарованих учнів: методичні рекомендації / Н. І. Поліхун, К. Г. Постова, І. А. Сліпухіна, Г. В. Онопченко, О. В. Онопченко. – Київ : Інститут обдарованої дитини НАПН України, 2019. – 80 с
43. Чуйко О.В. Використання методу проєктів на уроках та в позаурочний час./О.В. Чуйко//Фізика в школах України.-2008-№11-12(111-112)-с.2-5.
44. Шоман Н.Ф. Психолого-педагогічна суть поняття «навчальний преєкт»./Н.Ф. Шоман//Педагогічна майстерня.-2012-№7(19)-с.23-26.
45. Шубіна Т. Проєктна діяльність як важливий аспект компетентнісної освіти учнів./Т.Шубіна//Педагогічна Житомирщина.-2013-№4(72)-с.45-48.
46. Шулікін Д. STEM-освіта[Електронний ресурс]./Д. Шулікін. - Режим доступу: <http://iteach.com.ua/news/mass-media/?pid=2621/>.
47. Що таке STEM-освіта у навчальному закладі [Електронний ресурс]. / Режим доступу:
48. Як надати нашим дітям STEM-освіту. 8 кроків до успішного майбутнього. Режим доступу:<http://vynahidnyk.org/arhiv-novyn-ta-podiy/STEM.html>.-Назва з екрана.
49. Як стати дослідником : посіб. для учнів / Н. І. Поліхун. – Київ : Інформ. сист., 2010. – 224 с.
50. Якушенко Л. М. Актуальні проблеми соціального захисту людей з інвалідністю. Аналітична записка Серія «Соціальна політика», № 9, 2019, С. 1-11. URL: <https://niss.gov.ua/sites/default/files/2019-12/analit-yakushenko-social-policy-9-2019.pdf>.



**Додаток А****Клас 10****Дата проведення уроку** \_\_.\_\_.20\_\_**Урок № 1/9.****Тема:** Особливості середовища розробки.**Мета:**

**навчальна:** познайомити учнів програмно-апаратним комплексом Arduino;

**розвивальна:** розвивати алгоритмічне мислення, увагу, пам'ять;

**виховна:** виховувати дисципліну, зібраність, уміння сконцентруватися, позитивне ставлення учнів до навчально-пізнавальної діяльності.

**Тип уроку:** комбінований.**Програмне забезпечення:** середовище програмування Arduino IDE.**Апаратне забезпечення:** комп'ютер, плата Arduino Uno, USB-кабель.**План уроку (45 хв.)**

1. Організаційний етап (2 хв.).
2. Повідомлення нового матеріалу (20 хв.).
3. Розв'язування задач на комп'ютері (20 хв.).
4. Підсумки уроку (2 хв.).

**Хід уроку:****I.Організаційний етап (2 хв.).****II.Повідомлення нового матеріалу (20 хв.).**

Arduino (Ардуіно) — апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та середовище розробки Processing/Wiring на мові програмування, що є спрощеною підмножиною C/C++.

Інформація про плату (рисунок друкованої плати, специфікації елементів, програмне забезпечення) знаходяться у відкритому доступі і можуть бути використані тими, хто воліє створювати плати власноруч.

Плата Arduino складається з мікроконтролера Atmel AVR, а також елементів обв'язкових для програмування та інтеграції з іншими пристроями. У мікроконтролер записаний завантажувач (bootloader), тому зовнішній програматор не потрібен.

Плати Arduino дозволяють використовувати значну кількість виводів мікроконтролера як вхідні/вихідні контакти у зовнішніх схемах. Наприклад, у платі Decimila доступно 14 цифрових входів/виходів, 6 із яких можуть генерувати ШІМ сигнал, і 6 аналогових входів. Ці сигнали доступні на платі через контактні площадки або штирові роз'єми. Ці сигнали доступні на платі через контактні площадки або штирові роз'єми.

Ми з вами будемо працювати з платою Arduino Uno (рис. 3.1), яка складається з USB-порту, джерела живлення, кнопки скидання (перезапуску), мікроконтролера, цифрових портів (пінів), аналогових входів, живлення зовнішніх компонентів та вбудованого світлодіоду (13 пін).



Під час роботи з платою слід дотримуватись наступних правил:

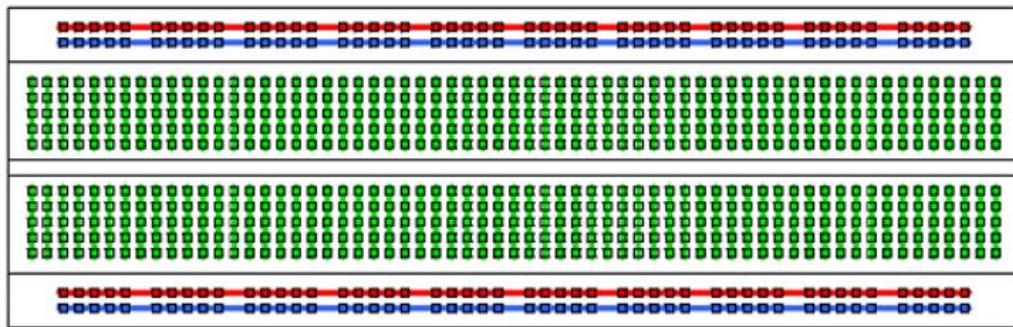
1. Збирання й розбирання електричної схеми здійснюється при вимкненому живленні.
2. Дотримуйтеся правильної полярності підключення компонентів.
3. Перед вмиканнями живлення продемонструйте зібрану схему викладачу.
4. Уникайте контакту всіх електронних компонентів з рідинами та не жививайте їжу й напої на робочому місці.
5. Не торкайтеся контактів при увімкненому живленні.

6. Не кладіть плату на металеві предмети.
7. Не замикайте «піни» між собою.
8. Заборонено підключати плату до невідомих джерел живлення.
9. У разі аварійної ситуації або виникнення сумнівів щодо підключення того чи іншого компонента негайно зверніться до викладача.

Написання програмного коду здійснюється за допомогою середовища Arduino IDE. Для цього слід у програмному середовищі обрати Файл > Новий. Після написання коду треба зберегти програму і натиснути кнопку Завантаження.

Для роботи з Arduino Uno буде потрібно будь-який персональний комп'ютер з вільним USB-портом. У разі підключення Arduino до ПК зовнішнє живлення пристрою не потрібне.

Для включення мікроконтролера, достатньо подати на нього живлення або під'єднати до ПК через кабель USB, або безпосередньо через спеціальний роз'єм зовнішнього живлення. При цьому, напруга зовнішнього живлення може змінюватись від +7В до +12В.



Макетна плата дозволяє здійснювати складання електричних ланцюгів без паяння.

По краях плати йдуть довгі провідники, які найчастіше називають шиною живлення. Провідники ж розміщені в середині плати, що з'єднуються в напрямку, перпендикулярному до напрямку шин живлення.



Рис. 3.3. Вікно програми Arduino.

У пункті Файл можна знайти команди, які відповідають за створення нової програми, читання старої, збереження її змін, а також команди для завантаження програми на мікроконтролер.

- Створити — створити нову програму (скетч);
- Відкрити — відкрити програму;
- Папка зі скетчами — відкрити програму із заданої папки;
- Приклади — відкрити приклад програми;
- Закрити — закрити поточне вікно.
- Зберегти — зберегти зміни у раніше збереженій програмі;
- Зберегти як — зберегти нову програму із зазначенням імені;
- Завантажити — завантажити програму в Arduino;
- Завантажити за допомогою програматора — завантажити програму за допомогою програматора;
- Налаштування друку — Налаштування принтера;
- Друк — друк коду програми;
- Налаштування — налаштування редактора;

- Вихід — вихід з Arduino IDE

Пункт меню Правка містить команди, пов'язані з редагування тексту програми, включаючи копіювання, вставку, налаштування відступів та пошук за ключовим словом.

У розділі Скетч розміщуються команди для керування компіляцією програми.

- Перевірити/компілювати — компілювати програму;
- Показати папку скетчів — відкрити системну папку з програмами;
- Додати файл — додати до проекту файл із даними або програмою;
- Імпортувати бібліотеку — підключити до програми бібліотеку зі списку встановлених.

Пункт меню Сервіс включає допоміжні функції для роботи з самим мікроконтролерів.

- Автоформатування — автоматичне розміщення відступів, переносів рядків тощо;
- Архівувати скетч — архівація папки з програмою та збереження архіву у вказане місце;
- виправити кодування та перезавантажити
- Монітор порту — відкрити вікно обмінюватись даними з мікроконтроллером;
- Плата — вибір поточної плати (у разі Arduino Uno);
- Послідовний порт — вибір порту, до якого підключено пристрій;
- Програматор — вибір програматора;
- Записати — завантажувач запис програмизавантажувача в мікроконтролер (не використовується в даному курсі).

Нарешті меню Довідка містить докладний опис всіх функцій самого редактора Arduino IDE, а також всілякі команди та прийоми роботи з платформою Arduino.

Кожна програма Arduino може складатися з декількох файлів. Для перемикання між цими файлами служить система вкладок у редакторі. Там же можна створити нову вкладку, і асоціювати з нею файл у папці з проектом.

Безпосередньо текст програми створюється і редагується в головному вікні редактора. По суті, вікно редактора є типовим текстовим редактором, з підсвічуванням конструкцій коду.

У самому низу редактора Arduino IDE є невелике вікно, яке служить для виведення повідомлень про помилки, що виникають у процесі компіляції програми, або під час завантаження програми в мікроконтролер.

Програмування Arduino виконується мовою C++.

Кожний вираз закінчується символом, точка з комою. Наприклад:

```
a + b = c;
```

Тіло функцій та складових операторів (if, else, for, while) відокремлюється фігурними дужками. Наприклад:

```
if (a > 0)
{
    b = a + 1;
}
```

Рядки відокремлюються звичайними подвійними лапками”. Приклад:

```
println(“some text”);
```

Символи відокремлюються одинарними лапками:

```
symbol = 'a';
```

Підключення бібліотек здійснюється за допомогою конструкції:

```
#include <math.h>
```

Рядкові коментарії у програмі починаються із символів // два прямих слеша. Приклад:

```
// коментар
```

Багато рядкові коментарії у програмі починаються із символів // два прямих слеша. Приклад:

```
/*
```

```
Коментар
```

```
*/
```

### **III. Розв'язування задач на комп'ютері (20 хв.).**

Перша програма.

Давайте розглянемо першу програму. В ній ви будете бачити вбудованим світлодіодом. Для цього Оберіть Файл > Приклади > 01. Basics > Blink.

```
/*
```

```
Це багаторядковий коментар.
```

```
Він ніяк не впливає на виконання програми,  
але полегшує її розуміння людиною.
```

```
*/
```

```
//Це однорядковий коментар
```

```
// Функція setup() виконається один раз при запуску програми
```

```
// Перезапуск програми відбувається при подачі живлення або щоразу при  
натисканні кнопки Reset на платі.
```

```
void setup() {
```

```
// встановлення режиму роботи "піну" вбудованого світлодіоду для  
виведення (OUTPUT)
```

```
pinMode(LED_BUILTIN, OUTPUT);
```

```
}
```

```
// Функція loop (англ. "цикл") багатократно повторюється під час виконання  
програми
```

```
void loop() {
```

```
digitalWrite(LED_BUILTIN, HIGH); // Увімкнути світлодіод (HIGH - подати  
високий рівень напруги)
```

```
delay(1000); // Чекати одну секунду (1000мс=1с)
```

```
digitalWrite(LED_BUILTIN, LOW); // Вимкнути світлодіод, подавши
низький (LOW) рівень напруги
delay(1000); // Чекати одну секунду
}
```

Як ви вже побачили, програма складається з двох функцій `setup()` і `loop()`. `setup()` виконується лише один раз на початку програми, а `loop()` виконується багаторазово, як нескінченний цикл. В програмі написані коментарі, тому ваше завдання самостійно розібратись як працює ця програма, а після чого запустити програму і подивитись на її виконання. Для цього спочатку треба підключити плату до комп'ютера за допомогою USB-кабеля, а потім натиснути кнопку Завантаження.

Після того як запустите програму змініть код так, щоб світлодіод почергово вмикався на 1, 2 та 3 секунди і гас на такий же час.

#### IV. Підсумки уроку (2 хв.).

**Клас 10**

**Дата проведення уроку** \_\_.\_\_.20\_\_

**Урок № 2/9.**

**Тема:** Основні елементи мови програмування.

**Мета:**

**навчальна:** познайомити учнів з типами даних в мові програмування C++, математичними операторами, навчити створювати змінні різних типів;

**розвивальна:** розвивати алгоритмічне мислення, увагу, пам'ять;

**виховна:** виховувати дисципліну, зібраність, уміння сконцентруватися, позитивне ставлення учнів до навчально-пізнавальної діяльності.

**Тип уроку:** комбінований.

**Програмне забезпечення:** середовище програмування Arduino IDE.



**Апаратне забезпечення:** комп'ютер, плата Arduino Uno, USB-кабель, потенціометр, з'єднувальні проводи.

### **План уроку (45 хв.)**

5. Організаційний етап (2 хв.).
6. Повідомлення нового матеріалу (10 хв.).
7. Розв'язування задач на комп'ютері (30 хв.).
8. Підсумки уроку (2 хв.).

### **Хід уроку:**

#### **IV. Організаційний етап (2 хв.).**

#### **V. Повідомлення нового матеріалу (15 хв.).**

Типи даних в мові програмування C++ — це спосіб визначення, які значення може приймати змінна, які операції з нею можна виконувати і скільки пам'яті вона займає. В C++ існує кілька базових типів даних, які можна розділити на наступні групи:

**Логічний** тип `bool`: може мати одне з двох значень — `true` (істина) або `false` (хиба). Займає 1 байт пам'яті.

**Цілочисельні** типи: представляють цілі числа в десятковій системі числення. Залежно від модифікатора `signed` або `unsigned`, можуть бути знаковими або беззнаковими. Займають від 1 до 8 байт пам'яті. До цієї групи належать такі типи:

`char`: представляє один символ в кодуванні ASCII. Може бути знаковим (`signed char`) або беззнаковим (`unsigned char`). Займає 1 байт пам'яті. Діапазон значень від -128 до 127 для знакового типу і від 0 до 255 для беззнакового типу.

`short`: представляє коротке ціле число. Може бути знаковим (`signed short`) або беззнаковим (`unsigned short`). Займає 2 байта пам'яті. Діапазон значень від -32768 до 32767 для знакового типу і від 0 до 65535 для беззнакового типу.

`int`: представляє ціле число. Може бути знаковим (`signed int`) або беззнаковим (`unsigned int`). Займає 2 або 4 байта пам'яті, в залежності від архітектури процесора. Діапазон значень від -32768 до 32767 (для 2 байт) або

від -2147483648 до 2147483647 (для 4 байт) для знакового типу і від 0 до 65535 (для 2 байт) або від 0 до 4294967295 (для 4 байт) для беззнакового типу.

`long`: представляє довге ціле число. Може бути знаковим (`signed long`) або беззнаковим (`unsigned long`). Займає 4 або 8 байт пам'яті, в залежності від архітектури процесора. Діапазон значень від -2147483648 до 2147483647 (для 4 байт) або від -9223372036854775808 до 9223372036854775807 (для 8 байт) для знакового типу і від 0 до 4294967295 (для 4 байт) або від 0 до 18446744073709551615 (для 8 байт) для беззнакового типу.

`long long`: представляє дуже довге ціле число. Може бути знаковим (`signed long long`) або беззнаковим (`unsigned long long`). Займає 8 байт пам'яті. Діапазон значень від -9223372036854775808 до 9223372036854775807 для знакового типу і від 0 до 18446744073709551615 для беззнакового типу.

Типи з **плаваючою комою**: представляють дійсні числа з десятковою точкою. Займають від 4 до 16 байт пам'яті. До цієї групи належать такі типи:

`float`: представляє число з плаваючою комою одинарної точності. Займає 4 байта пам'яті. Має приблизно 7 значущих цифр і діапазон експоненти від -38 до +38.

`double`: представляє число з плаваючою комою подвійної точності. Займає 8 байт пам'яті. Має приблизно 15 значущих цифр і діапазон експоненти від -308 до +308.

`long double`: представляє число з плаваючою комою розширеної точності. Займає 10 або 16 байт пам'яті, в залежності від архітектури процесора. Має приблизно 19 значущих цифр і діапазон експоненти від -4932 до +4932.

Для створення змінної в C++ потрібно вказати її тип, ім'я і необов'язково початкове значення. Наприклад, так можна створити змінну типу `int` з ім'ям `x` і значенням 10:

```
int x = 10;
```

Або так можна створити змінну типу `double` з ім'ям у без початкового значення:

```
double y;
```

Арифметичні оператори мови C++ - це спеціальні символи, які використовуються для виконання математичних операцій над змінними, константами, виразами та об'єктами. Арифметичні оператори C++ можуть бути розділені на наступні категорії:

Оператори додавання (+) і віднімання (-): використовуються для виконання операцій додавання і віднімання над числами, вказівниками, векторами, матрицями та іншими типами даних. Наприклад,  $a + b$ ,  $c - d$ ,  $e + f * g$ ,  $h - i / j$ .

Оператори множення (\*) і ділення (/): використовуються для виконання операцій множення і ділення над числами, векторами, матрицями та іншими типами даних. Наприклад,  $a * b$ ,  $c / d$ ,  $e * f / g$ ,  $h / i * j$ .

Оператори інкременту (++) і декременту (--): використовуються для збільшення або зменшення значення змінної на одиницю. Можуть бути префіксними або постфіксними. Наприклад, ++a, b++, --c, d--.

Оператори остачі від ділення (%): використовуються для виконання операції отримання остачі від ділення цілих чисел. Наприклад,  $a \% b$ ,  $c \% d$ ,  $e \% f \% g$ ,  $h \% i \% j$ .

Оператори присвоєння (=, +=, -=, \*=, /=, %=): використовуються для присвоєння значень змінним або об'єктам. Можуть бути простими або складеними. Наприклад,  $a = b$ ,  $c += d$ ,  $e -= f$ ,  $g *= h$ ,  $i /= j$ ,  $k \% = 1$ .

Для розв'язання сьогоденішньої задачі нам знадобиться потенціометр.

Змінний резистор або потенціометр — це електричний пристрій, значення рівня опору якого можна задати в певних межах. Таким чином ми можемо змінювати параметри електричних схем, гнучко підлаштовуючи їх під певні умови: наприклад, регулювати чутливість датчика або гучність звуку в

динаміці. Потенціометри набули широкого поширення в схемах регулювання гучності, напруги, контрастності і т.д., за свою простоту і практичність.

Потенціометр має три піни: живлення(5V), аналоговий пін та земля(GND). Він складається з 3-х пнів, резисторної речовини, ротора з движком та ручки. Принцип його роботи наступний (рис. 2.2): між пінами 5V та GND по резисторній речовині протікає електричний струм, резисторна речовина має свій опір, а одже на кожній її ділянці опір поступово збільшується, так біля піну 5V опору майже не буде, а біля GND опір буде максимальний; ротор приєднаний до ротору і ручки, за допомогою якої ми можемо переміщувати його по резисторній речовині, ротор в свою чергу буде передавати напругу, отриману з движка до аналогово піну. Плата Arduino Uno зчитає значення напруги з датчика і поверне значення від 0 до 1023.

Підключення потенціометра (рис. 2.3):

- Чорний — GND.
- Червоний — живлення 5V.
- Зелений — аналоговий вхід A0.

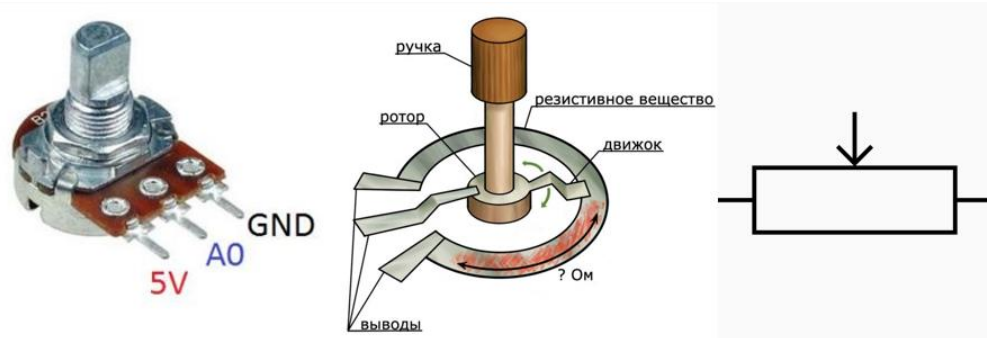


Рис. 2.2 Потенціометр. Підключення та будова потенціометра

## VI. Розв'язування задач на комп'ютері (20 хв.).

Перша програма.

Напишіть програму, яка зчитує значення потенціометра, який підключений до аналогового піна A0, і виводить його на серійний порт. Для цього потрібно використати функції `setup()` і `loop()`, як і в попередній задачі. В функції `setup()` потрібно встановити швидкість передачі даних за допомогою

функції `Serial.begin()`. В функції `loop()` потрібно зчитати значення потенціометра за допомогою функції `analogRead()` і вивести його на серійний порт за допомогою функції `Serial.println()`.

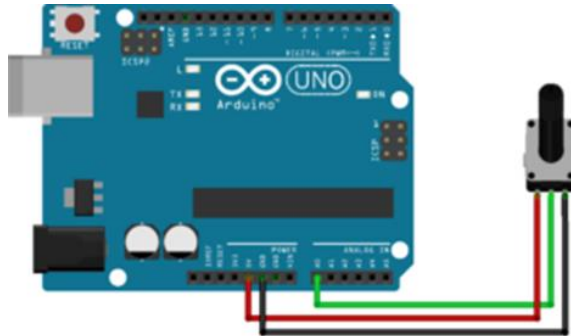


Рис. Схема підключення потенціометра до плати Arduino Uno

Значення з потенціометра будемо виводити в послідовний порт, який знаходиться на вкладці Інструменти. Ось код програми:

```
#define PIN_POT  A0

void setup()
{
  //Відкриття послідовного порту
  Serial.begin(9600);
  // Пін, до якого підключено змінний резистор, налаштовуємо на введення
  pinMode(PIN_POT, INPUT);
}

void loop(){
  // Змінна для зберігання значення опору
  int rotat;
  // Зчитування напруги зі змінного резистора
  // Мікроконтроллер поверне число від 0 до 1023
  rotat = analogRead(PIN_POT);
  Serial.print("Rotat: "); // Виведення в порт рядку "Rotat: "
```

```

Serial.write(rotat); //Виведення значення в порт
Serial.println(); // Перехід на новий рядок
delay(1000); // Затримка 1000 мс
}

```

Третя програма.

Зчитати значення потенціометра, який підключений до аналогового піна A1, і встановити яскравість світлодіода, який підключений до піна 10, пропорційно до цього значення.

Для розв'язання цієї задачі на ардуіно потрібно використати функції `setup()` і `loop()`, як і в попередніх задачах. В функції `setup()` потрібно встановити режим роботи піна 10 як вихідний за допомогою функції `pinMode()`. В функції `loop()` потрібно зчитати значення потенціометра за допомогою функції `analogRead()`, перетворити його в діапазон від 0 до 255 за допомогою функції `map()`, і встановити яскравість світлодіода за допомогою функції `analogWrite()`.

```

void setup() {
    // встановлюємо режим роботи піна 10 як вихідний
    pinMode(10, OUTPUT);
}

void loop() {
    // зчитуємо значення потенціометра з піна A1
    int value = analogRead(A1);
    // перетворюємо значення в діапазон від 0 до 255
    int brightness = map(value, 0, 1023, 0, 255);
    // встановлюємо яскравість світлодіода на піні 10
    analogWrite(10, brightness);
}

```

#### IV. Підсумки уроку (2 хв.).

**Клас 10**

**Дата проведення уроку** \_\_.\_\_.20\_\_

**Урок № 3/9.**

**Тема:** Оператори і вирази.

**Мета:**

**навчальна:** закріпити знання учнів про типи даних та арифметичні операції в мові c++;

**розвивальна:** розвивати алгоритмічне мислення, увагу, пам'ять;

**виховна:** виховувати дисципліну, зібраність, уміння сконцентруватися, позитивне ставлення учнів до навчально-пізнавальної діяльності.

**Тип уроку:** практична робота.

**Програмне забезпечення:** середовище програмування Arduino IDE.

**Апаратне забезпечення:** комп'ютер, плата Arduino Uno, USB-кабель, потенціометр, з'єднувальні проводи.

#### **План уроку (45 хв.)**

1. Організаційний етап (4 хв.).
2. Розв'язування задач на комп'ютері (30 хв.).
3. Перевірка раніше засвоєних знань (10 хв.).
4. Підсумки уроку (1 хв.).

#### **Хід уроку:**

##### **I. Організаційний етап (2 хв.).**

##### **II. Розв'язування задач на комп'ютері (20 хв.).**

Перша програма.

Напишіть програму, яка виводить на екран повідомлення “Привіт, світ!” за допомогою функції `Serial.println()`. Для цього потрібно використати функції `setup()` і `loop()`, які є основними функціями в ардуіно. Функція `setup()` виконується один раз при запуску програми, а функція `loop()` виконується безкінечно. В функції `setup()` потрібно встановити швидкість передачі даних за допомогою функції `Serial.begin()`. В функції `loop()` потрібно вивести повідомлення за допомогою функції `Serial.println()`.

```

void setup() {
    // встановлюємо швидкість передачі даних 9600 біт/с
    Serial.begin(9600);
}

void loop() {
    // виводимо повідомлення на екран
    Serial.println("Привіт, світ!");
}

```

Друга програма.

Напишіть програму, яка обчислює і виводить на екран площу трикутника за формулою  $S = (a * h) / 2$ , де  $a$  - довжина основи,  $h$  - висота. Значення  $a$  і  $h$  вводяться з клавіатури.

Для розв'язання цієї задачі потрібно використати функції `setup()` і `loop()`. В функції `setup()` потрібно встановити швидкість передачі даних за допомогою функції `Serial.begin()`. В функції `loop()` потрібно запитати у користувача значення  $a$  і  $h$  за допомогою функцій `Serial.print()` і `Serial.parseInt()`, обчислити площу трикутника за формулою  $S = (a * h) / 2$  і вивести її на екран за допомогою функції `Serial.println()`.

```

void setup() {
    // встановлюємо швидкість передачі даних 9600 біт/с
    Serial.begin(9600);
}

void loop() {
    // запитуємо у користувача значення a
    Serial.print("Введіть значення a: ");
    // зчитуємо значення a з клавіатури
    int a = Serial.parseInt();
}

```



```

// запитуємо у користувача значення h
Serial.print("Введіть значення h: ");
// зчитуємо значення h з клавіатури
int h = Serial.parseInt();
// обчислюємо площу трикутника за формулою S = (a * h) / 2
int S = (a * h) / 2;
// виводимо площу трикутника на екран
Serial.print("Площа трикутника: ");
Serial.println(S);
}

```

Третя програма.

Напишіть програму, яка переводить градуси Цельсія в градуси Фаренгейта за формулою  $F = (C * 9 / 5) + 32$ , де  $C$  - температура в градусах Цельсія,  $F$  - температура в градусах Фаренгейта. Значення  $C$  вводиться з клавіатури.

Для розв'язання цієї задачі на ардуіно потрібно використати функції `setup()` і `loop()`. В функції `setup()` потрібно встановити швидкість передачі даних за допомогою функції `Serial.begin()`. В функції `loop()` потрібно запитати у користувача значення  $C$  за допомогою функцій `Serial.print()` і `Serial.parseInt()`, перевести його в градуси Фаренгейта за формулою  $F = (C * 9 / 5) + 32$  і вивести його на екран за допомогою функції `Serial.println()`.

```

void setup() {
    // встановлюємо швидкість передачі даних 9600 біт/с
    Serial.begin(9600);
}

void loop() {
    // запитуємо у користувача значення C
    Serial.print("Введіть значення C: ");

```

```

// зчитуємо значення C з клавіатури
int C = Serial.parseInt();

// переводимо значення C в градуси Фаренгейта за формулою F = (C *
9 / 5) + 32
int F = (C * 9 / 5) + 32;

// виводимо значення F на екран
Serial.print("Значення F: ");
Serial.println(F);
}

```

Четверта програма.

Зчитати значення потенціометра, який підключений до аналогового піна A1, і встановити яскравість світлодіода, який підключений до піна 10, пропорційно до цього значення.

Схема підключення:

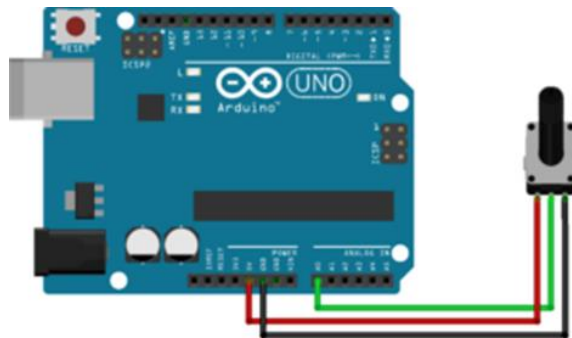


Рис. 3.4. Підключення потенціометра

Для розв'язання цієї задачі на ардуіно потрібно використати функції `setup()` і `loop()`, як і в попередніх задачах. В функції `setup()` потрібно встановити режим роботи піна 10 як вихідний за допомогою функції `pinMode()`. В функції `loop()` потрібно зчитати значення потенціометра за допомогою функції `analogRead()`, перетворити його в діапазон від 0 до 255 за допомогою функції `map()`, і встановити яскравість світлодіода за допомогою функції `analogWrite()`.

```

void setup() {
    // встановлюємо режим роботи піна 10 як вихідний

```

```
pinMode(10, OUTPUT);  
}  
  
void loop() {  
    // зчитуємо значення потенціометра з піна A1  
    int value = analogRead(A1);  
    // перетворюємо значення в діапазон від 0 до 255  
    int brightness = map(value, 0, 1023, 0, 255);  
    // встановлюємо яскравість світлодіода на піні 10  
    analogWrite(10, brightness);  
}
```

### **III. Перевірка раніше засвоєних знань (10 хв.).**

Необхідно виконати наступний тест:

1. Який з наступних типів даних в C++ може приймати значення true або false?

- a) int
- b) char
- c) bool
- d) double

Відповідь: c).

2. Який з наступних операторів в C++ використовується для виконання операції додавання?

- a) +
- b) -
- c) \*
- d) /

Відповідь: a).

3. Який з наступних операторів в C++ використовується для виконання операції присвоєння?

- a) ==
- b) +=
- c) =
- d) :=

Відповідь: c).

4. Який з наступних операторів в C++ використовується для виконання операції інкременту (збільшення на одиницю)?

- a) ++
- b) -
- c) +=
- d) -=

Відповідь: a).

5. Який з наступних операторів в C++ використовується для виконання операції декременту (зменшення на одиницю)?

- a) ++
- b) -
- c) +=
- d) -=

Відповідь: b).

6. Який з наступних операторів в C++ використовується для виконання операції остачі від ділення?

- a) /
- b) //
- c) %
- d) \

Відповідь: c).

7. Який з наступних операторів в C++ використовується для виконання операції множення?

- a) +
- b) -

- c) \*
- d) /

Відповідь: c).

8. Який з наступних операторів в C++ використовується для виконання операції ділення?

- a) +
- b) -
- c) \*
- d) /

Відповідь: d).

9. Який з наступних типів даних в C++ займає 4 байти пам'яті і може представляти дійсні числа з одинарною точністю?

- a) int
- b) char
- c) bool
- d) float

Відповідь: d).

10. Який результат виконання коду:

```
void setup() {  
  Serial.begin(9600);  
  int a = 10;  
  int b = 3;  
  int c = a / b;  
  int d = a % b;  
    Serial.println(c);  
    Serial.println(d);  
}  
void loop() {  
  int a = 3;
```

```
int b =10;
}
```

- a) c = 1.0, d = 3.0
- b) c = 0.33, d = 0
- c) c = 0, d = 0.33
- d) c = 3, d = 1

Відповідь: d).

#### IV. Підсумки уроку (2 хв.).

**Клас 10**

**Дата проведення уроку** \_\_.\_\_.20\_\_

**Урок № 4/9.**

**Тема:** Логічні вирази. Таблиці істинності. Умовний оператор мовою програмування. Оператор вибору.

**Мета:**

**навчальна:** познайомити учнів з синтаксисом умовного оператора та оператором вибору в мові програмування C++, операторами порівняння, таблицею істинності;

**розвивальна:** розвивати алгоритмічне мислення, увагу, пам'ять;

**виховна:** виховувати дисципліну, зібраність, уміння сконцентруватися, позитивне ставлення учнів до навчально-пізнавальної діяльності.

**Тип уроку:** комбінований.

**Програмне забезпечення:** середовище програмування Arduino IDE.

**Апаратне забезпечення:** комп'ютер, плата Arduino Uno, USB-кабель, кнопка, з'єднувальні проводи.

#### **План уроку (45 хв.)**

1. Організаційний етап (2 хв.).
2. Повідомлення нового матеріалу (10 хв.).
3. Розв'язування задач на комп'ютері (30 хв.).

## 4. Підсумки уроку (2 хв.).

**Хід уроку:****I. Організаційний етап (2 хв.).****II. Повідомлення нового матеріалу (10 хв.).**

Порівняльні оператори мови C++ — це спеціальні символи, які використовуються для порівняння двох значень або виразів за певним критерієм, таким як рівність, нерівність, більше, менше тощо. Порівняльні оператори повертають логічне значення true або false в залежності від результату порівняння. У C++ існують такі порівняльні оператори:

- == — рівність. Повертає true, якщо обидва операнди рівні, інакше повертає false. Наприклад,  $a == b$  дорівнює true, якщо  $a$  і  $b$  мають однакове значення.
- != — нерівність. Повертає true, якщо обидва операнди не рівні, інакше повертає false. Наприклад,  $a != b$  дорівнює true, якщо  $a$  і  $b$  мають різні значення.
- > — більше. Повертає true, якщо перший операнд більший за другий, інакше повертає false. Наприклад,  $a > b$  дорівнює true, якщо  $a$  більше за  $b$ .
- < — менше. Повертає true, якщо перший операнд менший за другий, інакше повертає false. Наприклад,  $a < b$  дорівнює true, якщо  $a$  менше за  $b$ .
- >= — більше або рівне. Повертає true, якщо перший операнд більший або рівний другому, інакше повертає false. Наприклад,  $a >= b$  дорівнює true, якщо  $a$  більше або рівне  $b$ .
- <= — менше або рівне. Повертає true, якщо перший операнд менший або рівний другому, інакше повертає false. Наприклад,  $a <= b$  дорівнює true, якщо  $a$  менше або рівне  $b$ .

Порівняльні оператори мають нижчий пріоритет, ніж арифметичні оператори, але вищий, ніж логічні оператори.

Логічні оператори мови C++ - це спеціальні символи, які використовуються для вираження логічних зв'язків між значеннями або виразами, що мають логічний тип (bool) або можуть бути перетворені на нього. Логічні оператори повертають значення true або false в залежності від результату логічної операції. У C++ існують такі логічні оператори:

**&&** — логічне І (кон'юнкція). Повертає true, якщо обидва операнди є true, інакше повертає false. Наприклад, true && true дорівнює true, а true && false дорівнює false.

**||** — логічне АБО (диз'юнкція). Повертає true, якщо хоча б один з операндів є true, інакше повертає false. Наприклад, true || false дорівнює true, а false || false дорівнює false.

**!** — логічне НІ (негація). Повертає протилежне значення операнду. Наприклад, !true дорівнює false, а !false дорівнює true.

Логічні оператори мають нижчий пріоритет, ніж арифметичні та порівняльні оператори, але вищий, ніж оператори присвоєння. Логічне І має вищий пріоритет, ніж логічне АБО, тому вираз  $a \ || \ b \ \&\& \ c$  еквівалентний виразу  $a \ || \ (b \ \&\& \ c)$ . Для зміни порядку виконання логічних операцій можна використовувати дужки. Наприклад,  $(a \ || \ b) \ \&\& \ c$  означає, що спочатку виконується логічне АБО між  $a$  та  $b$ , а потім логічне І між результатом та  $c$ .

Логічні оператори також мають властивість короткого замикання, що означає, що другий операнд вираховується тільки тоді, коли це необхідно для визначення результату. Наприклад, вираз  $a \ \&\& \ b$  не буде обчислювати  $b$ , якщо  $a$  є false, оскільки в цьому випадку результат завжди буде false незалежно від  $b$ . Аналогічно, вираз  $a \ || \ b$  не буде обчислювати  $b$ , якщо  $a$  є true, оскільки в цьому випадку результат завжди буде true незалежно від  $b$ . Це може бути корисно для уникнення помилок, пов'язаних з невизначеними значеннями або діленням на нуль.

Розглянемо таблицю істинності.

<b>a</b>	<b>b</b>	<b>a &amp;&amp; b</b>	<b>  </b>	<b>!a</b>	<b>!b</b>
----------	----------	-----------------------	-----------	-----------	-----------



false	false	false	false	true	true
false	true	false	true	true	false
true	true	false	true	false	false
true	false	true	true	false	true

Оператор `if` є одним з основних засобів управління потоком виконання програми, який дозволяє виконувати певні дії в залежності від результату логічного виразу. Оператор `if` має наступний синтаксис:

```
if (умова) {
    // оператори для виконання, якщо умова виконується
}
```

Де умова — це будь-який вираз, який повертає значення типу `bool` або може бути перетворений на нього. Якщо умова дорівнює `true`, то виконуються оператори в фігурних дужках. Якщо умова дорівнює `false`, то ці оператори пропускаються.

Оператор `if` може бути також розширений за допомогою оператора `else`, який дозволяє виконати альтернативні дії, якщо умова не виконується. Оператор `else` має наступний синтаксис:

```
if (умова) {
    // оператори для виконання, якщо умова виконується
}
else {
    // оператори для виконання, якщо умова не виконується
}
```

Оператор `else` повинен слідувати безпосередньо за оператором `if`, з яким він пов'язаний. Якщо умова дорівнює `true`, то виконуються оператори в першому блоку, а оператори в другому блоку пропускаються. Якщо умова дорівнює `false`, то виконуються оператори в другому блоку, а оператори в першому блоку пропускаються.

Оператор `if` також може бути вкладений в інший оператор `if`, що дозволяє перевіряти додаткові умови. Наприклад:

```
if (умова1) {  
    if (умова2) {  
        // оператори для виконання, якщо умова1 і умова2 істинні  
    }  
    else {  
        // оператори для виконання, якщо умова1 істинна, а умова2 хибна  
    }  
}  
else {  
    // оператори для виконання, якщо умова1 хибна  
}
```

Оператор `if` також може бути поєднаний з логічними операторами `&&` (і), `||` (або) та `!` (ні), що дозволяє створювати складні умови.

Оператор вибору (`switch`) в мові програмування C++ — це конструкція, яка дозволяє вибрати один з декількох варіантів дій в залежності від значення виразу. Таким чином, забезпечується багатонаправлене розгалуження в програмі. Оператор `switch` може бути замінений оператором `if`. Однак, в деяких випадках використання оператора `switch` може бути більш ефективним, ніж використання оператора `if`.

Оператор `switch` має наступний синтаксис:

```
switch (вираз) {  
    case константа1:  
        // оператори, що виконуються, якщо вираз дорівнює константі1  
        break;  
    case константа2:  
        // оператори, що виконуються, якщо вираз дорівнює константі2  
        break;
```

```

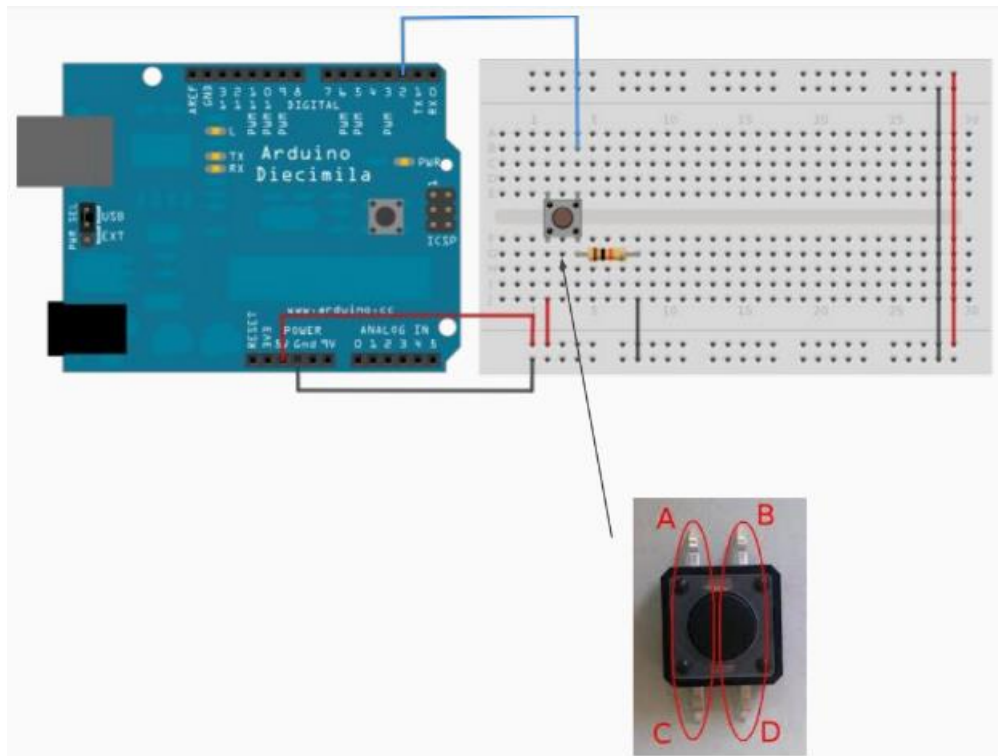
// ...
case константаN:
    // оператори, що виконуються, якщо вираз дорівнює константіN
    break;
default:
    // оператори, що виконуються, якщо вираз не дорівнює жодній з
констант
}

```

Де вираз — це будь-який вираз, який повертає значення цілочисельного (int) або символьного типу (char). Значення вираз послідовно порівнюється з константами з заданого списку. Як тільки знайдено співпадіння для однієї з умов порівняння (гілка case), виконуються оператори в фігурних дужках. Якщо не знайдено жодного співпадіння, то виконуються оператори, які слідують після ключового слова default.

Ключове слово case використовується для того, щоб вказати значення константи, при співпадінні з якою значення виразу, буде виконуватися відповідна послідовність операторів. Оператор break використовується для того, щоб припинити (закінчити) виконання коду, який визначений інструкцією switch. Без оператора break виконуватиметься кожен оператор з відповідної гілки case до кінця switch, включаючи оператор default. Оператори, які розміщені після ключового слова default, виконуються в тому випадку, якщо жодна з case-констант не співпадає з результатом обчислення switch-виразу. Ключове слово default та відповідна послідовність операторів можуть бути пропущені. В цьому випадку, якщо не виконується жодна з гілок після слова case, виконання передається наступному оператору, який слідує за оператором switch.

Для розв'язання сьогоденішньої задачі знадобиться кнопка.



Кнопка — це простий елемент, який дозволяє вводити сигнал в мікроконтролер за допомогою механічного натискання. Кнопка може мати два стани: натиснута або відпущена. Коли кнопка натиснута, вона замикає контакт між двома ножками, які підключені до пінів Arduino. Коли кнопка відпущена, вона розмикає контакт. Для того, щоб визначити стан кнопки, потрібно використовувати функцію `digitalRead()`, яка повертає значення HIGH або LOW в залежності від того, чи є напруга на піні.

Однак, для правильної роботи кнопки, потрібно також враховувати деякі особливості:

Потрібно використовувати підтягуючий резистор, який створює постійне з'єднання між піном Arduino і землею або живленням. Це необхідно для того, щоб уникнути «плаваючого» стану піна, коли кнопка відпущена, і потрібно забезпечити чітке значення HIGH або LOW.

### **III. Розв'язування задач на комп'ютері (30 хв.).**

Перша програма.

Під'єднайте кнопку та самостійно розгляньте код програми.

```
const int buttonPin = 2; // номер контакту, до якого підключена кнопка
```

```

const int ledPin = 13; // номер контакту вбудованого світлодіоду
int buttonState = 0; // змінна для зчитування стану кнопки
void setup() {
  pinMode(ledPin, OUTPUT); // встановити режим піну світлодіоду output
  pinMode(buttonPin, INPUT); // встановити режим піну кнопки input
}
void loop() {
  // читання стану кнопки
  buttonState = digitalRead(buttonPin);
  // Перевірити, чи натиснена кнопка
  if (buttonState == HIGH) {
    // увімкнути світлодіод
    digitalWrite(ledPin, HIGH);
  } else {
    // вимкнути світлодіод
    digitalWrite(ledPin, LOW);
  }
}

```

#### IV. Підсумки уроку (2 хв.).

**Клас 10**

**Дата проведення уроку** \_\_.\_\_.20\_\_

**Урок № 5/9.**

**Тема:** Послідовні та вкладені розгалуження.

**Мета:**

**навчальна:** закріпити знання учнів з теми розгалуження, познайомити з конструкцією else if в мові програмування C++;

**розвивальна:** розвивати алгоритмічне мислення, увагу, пам'ять;

**виховна:** виховувати дисципліну, зібраність, уміння сконцентруватися, позитивне ставлення учнів до навчально-пізнавальної діяльності.

**Тип уроку:** комбінований.

**Програмне забезпечення:** середовище програмування Arduino IDE.

**Апаратне забезпечення:** комп'ютер, плата Arduino Uno, USB-кабель вібро мотор, ультразвуковий датчик, з'єднувальні проводи.

### План уроку (45 хв.)

1. Організаційний етап (2 хв.).
2. Повідомлення нового матеріалу (10 хв.).
3. Розв'язування задач на комп'ютері (30 хв.).
4. Підсумки уроку (2 хв.).

### Хід уроку:

#### I. Організаційний етап (2 хв.).

#### II. Повідомлення нового матеріалу (10 хв.).

Оператор ? — це оператор умовного вираження, який дозволяє вибрати одне з двох значень в залежності від результату логічного виразу. Оператор ? має наступний синтаксис:

логічний\_вираз ? значення\_якщо\_істина : значення\_якщо\_хиба

Оператор ? повертає значення\_якщо\_істина, якщо логічний\_вираз є істинним, або значення\_якщо\_хиба, якщо логічне\_вираження є неправдивим. Оператор ? може бути використаний для скорочення коду, коли потрібно виконати просту умовну дію.

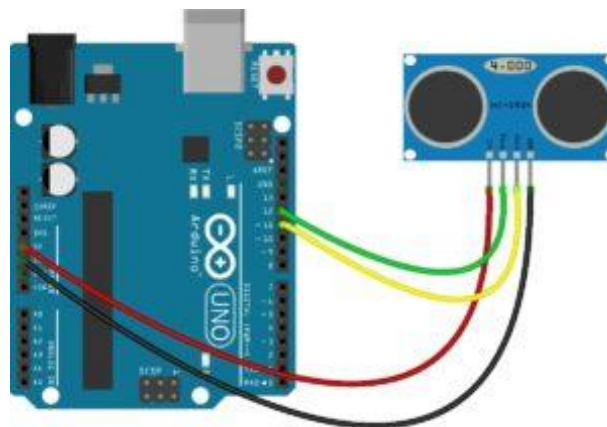
else if в мові C++ є частиною умовної конструкції if-else, яка дозволяє перевіряти декілька умов послідовно. Синтаксис else if такий:

```
if (умова1) {
// код, який виконується, якщо умова1 істинна
} else if (умова2) {
// код, який виконується, якщо умова1 хибна, а умова2 істинна
} else {
// код, який виконується, якщо усі умови хибні
}
```

Сьогодні ми познайомимось з вібромотором та ультразвуковим датчиком відстані.



Вібромотор — це пристрій, який використовується для створення вібрації або тактильного зворотного зв'язку. Вібромотор складається з електродвигуна з ексцентриком на валу, який при обертанні створює нерівномірний момент сили. Це призводить до того, що вібромотор коливається і передає вібрацію до поверхні, до якої він прикріплений. Вібромотори часто використовуються в мобільних пристроях, ігрових контролерах, роботах та інших застосуваннях, де потрібно симулювати дотик або оповістити користувача про якусь подію.



Ультразвуковий датчик — це пристрій, який дозволяє вимірювати відстань до об'єктів за допомогою ультразвукових хвиль. Ультразвуковий датчик працює за принципом ехолокації, який використовують деякі тварини, такі як кажани або дельфіни, для орієнтування в просторі. Ехолокація - це процес, коли тварина випромінює звукові хвилі і слухає їх відбиття від

об'єктів. За часом і напрямком відбиття тварина може визначити відстань, розмір, форму та рух об'єктів.

Ультразвуковий датчик використовує той самий принцип, але замість звичайних звукових хвиль він використовує ультразвукові хвилі, які мають високу частоту і не чутні для людського вуха. Ультразвуковий датчик складається з передавача і приймача, які випромінюють і отримують ультразвукові хвилі відповідно. Коли датчик висилає ультразвукову хвилю в напрямку об'єкта, то частина хвилі відбивається від об'єкта і повертається до датчика. Датчик зчитує час, який потрібен для того, щоб хвиля дісталася до об'єкта і повернулася назад, і за допомогою формули обчислює відстань до об'єкта. Формула має такий вигляд:  $d = v * t / 2$ , де  $d$  - відстань,  $v$  - швидкість звуку, а  $t$  - час.

### **III. Розв'язування задач на комп'ютері (30 хв.).**

Перша програма.

Підключіть ультразвуковий датчик за наступним принципом:

1. Контакт “земля” (GND) — до виходу GND плати Arduino.
2. Контакт живлення (Vcc) — до виходу 5V на платі Arduino.
3. Trig – 12 пін.
4. Echo – 11 пін.

Та розгляньте роботу коду:

```
#define PIN_TRIG 12 //Константа PIN_TRIG дорівнює 12, пін, до якого
підключено trig
#define PIN_ECHO 11 //Константа PIN_ECHO дорівнює 11, пін, до якого
підключено echo
long duration, cm ; //duration - час, cm - відстань у см.
void setup() {
  Serial. begin (9600); //Відкриття порту
  pinMode (PIN_TRIG, OUTPUT ); //Пін 12 налаштовується на виведення
  pinMode (PIN_ECHO, INPUT ); //Пін 11 налаштовується на введення
```



```

pinMode(9, OUTPUT); // Встановлюємо пін 9 як вихідний
}
void loop() {
// Генеруємо короткий імпульс 2-5 мікросекунд.
digitalWrite (PIN_TRIG, LOW );
delayMicroseconds (5);
digitalWrite (PIN_TRIG, HIGH );
// Встановивши високий рівень сигналу, чекаємо 10 мікросекунд.
// У цей момент датчик відправляє сигнали з частотою 40 КГц.
delayMicroseconds (10);
digitalWrite (PIN_TRIG, LOW );
// Одержуємо час затримки сигналу з приймача.
duration = pulseIn(PIN_ECHO, HIGH );
// Розраховуємо відстань (1 / швидк.звуку = 1 / 0.03435 см/мс = 29.1
мс/см)
cm = (duration / 2) / 29.1;
Serial. print("Відстань до об'єкту: ");
Serial. print(cm);
Serial. println(" см.");
delay (250); // Пауза між вимірюваннями
}

```

Друга програма.

До попереднього завдання додайте підключення вібротору. Передбачте щоб вібротор вмикався, коли перешкода знаходиться близько.

```

#define PIN_TRIG 12 //Константа PIN_TRIG дорівнює 12, пін, до якого
підключено trig
#define PIN_ECHO 11 //Константа PIN_ECHO дорівнює 11, пін, до якого
підключено echo
long duration, cm ; //duration - час, cm - відстань у см.

```

```

void setup() {
  Serial. begin (9600); //Відкриття порту
  pinMode (PIN_TRIG, OUTPUT ); //Пін 12 налаштовується на виведення
  pinMode (PIN_ECHO, INPUT ); //Пін 11 налаштовується на введення
  pinMode(9, OUTPUT); // Встановлюємо пін 9 як вихідний
}

void loop() {
  // Генеруємо короткий імпульс 2-5 мікросекунд.
  digitalWrite (PIN_TRIG, LOW );
  delayMicroseconds (5);
  digitalWrite (PIN_TRIG, HIGH );
  // Встановивши високий рівень сигналу, чекаємо 10 мікросекунд.
  // У цей момент датчик відправляє сигнали з частотою 40 КГц.
  delayMicroseconds (10);
  digitalWrite (PIN_TRIG, LOW );
  // Одержуємо час затримки сигналу з приймача.
  duration = pulseIn(PIN_ECHO, HIGH );
  // Розраховуємо відстань (1 / швидк.звуку = 1 / 0.03435 см/мс = 29.1
мс/см)
  cm = (duration / 2) / 29.1;
  if (cm > 10)
  {
    digitalWrite(9, HIGH); // Вмикаємо вібромотор, подаючи високий
сигнал на пін 9
  }
  Else
  {
    digitalWrite(9, LOW); // Виключаємо вібромотор, подаючи низький
сигнал на пін 9
  }
}

```

```
delay (250);// Пауза між вимірюваннями
}
```

#### IV. Підсумки уроку (2 хв.).

**Клас 10**

**Дата проведення уроку** \_\_.\_\_.20\_\_

**Урок № 6/9.**

**Тема:** Тестування розгалужених алгоритмів.

**Мета:**

**навчальна:** закріпити знання учнів з теми «Розгалужені алгоритми»;

**розвивальна:** розвивати алгоритмічне мислення, увагу, пам'ять;

**виховна:** виховувати дисципліну, зібраність, уміння сконцентруватися, позитивне ставлення учнів до навчально-пізнавальної діяльності.

**Тип уроку:** практична робота.

**Програмне забезпечення:** середовище програмування Arduino IDE.

**Апаратне забезпечення:** комп'ютер, плата Arduino Uno, USB-кабель, потенціометр, кнопка, вібромотор, ультразвуковий датчик, з'єднувальні проводи.

#### **План уроку (45 хв.)**

1. Організаційний етап (2 хв.).
2. Розв'язування задач на комп'ютері (30 хв.).
3. Перевірка раніше засвоєних знань (10 хв.).
4. Підсумки уроку (2 хв.).

#### **Хід уроку:**

##### **I. Організаційний етап (2 хв.).**

##### **II. Розв'язування задач на комп'ютері (30 хв.).**

Перша програма.

Створити програму, яка буде використовувати потенціометр для регулювання інтенсивності вібрації вібромотора. Якщо потенціометр повернутий до крайнього лівого положення, то вібромотор не працюватиме.

Якщо потенціометр повернутий до крайнього правого положення, то вібромотор працюватиме з максимальною силою. Якщо потенціометр знаходиться в проміжному положенні, то інтенсивність вібрації буде пропорційна значенню потенціометра. Для реалізації цієї задачі потрібно використати функції `analogRead()`, `analogWrite()` та `delay()`.

```
// Встановлюємо пін A0 як аналоговий вхідний пін, до якого
підключений потенціометр
const int potPin = A0;

// Встановлюємо пін 9 як аналоговий вихідний пін, до якого підключений
вібромотор
const int vibPin = 9;

// Створюємо змінну для зберігання значення потенціометра
int potValue;

// Функція, яка викликається один раз при запуску програми
void setup() {
  // Встановлюємо пін 9 як вихідний
  pinMode(vibPin, OUTPUT);
}

// Функція, яка викликається безперервно після функції setup()
void loop() {
  // Зчитуємо значення потенціометра в діапазоні від 0 до 1023
  potValue = analogRead(potPin);
  // Перевіряємо, чи значення потенціометра менше за 512
  if (potValue < 512) {
    // Якщо так, то виключаємо вібромотор, подаючи нульове значення на
пін 9
```

```

    analogWrite(vibPin, 0);
}
// Інакше, якщо значення потенціометра більше або дорівнює 512
else {
    // Обчислюємо значення для вібромотора, множачи значення
потенціометра на 0.5
    int vibValue = potValue * 0.5;
    // Подаємо значення для вібромотора на пін 9, який контролює
інтенсивність вібрації вібромотора
    analogWrite(vibPin, vibValue);
}
// Чекаємо 10 мілісекунд
delay(10);
}

```

Друга програма.

Створити програму, яка буде використовувати кнопку для включення або виключення ультразвукового датчика. Якщо кнопка натиснута, то датчик вимірюватиме відстань до об'єктів і виводитиме її на серійний порт. Якщо кнопка відпущена, то датчик не працюватиме. Для реалізації цієї задачі потрібно використати функції `digitalRead()`, `digitalWrite()`, `Serial.begin()`, `Serial.println()`, `delay()`.

```

// Встановлюємо пін 2 як цифровий вхідний пін, до якого підключена
кнопка
const int buttonPin = 2;
// Встановлюємо пін 9 як цифровий вихідний пін, до якого підключений
тригерний вивід датчика
const int trigPin = 9;
// Встановлюємо пін 10 як цифровий вхідний пін, до якого підключений
ехо-вивід датчика

```

```
const int echoPin = 10;

// Створюємо змінні для зберігання значення кнопки, часу і відстані
int buttonState = 0;
long duration, distance;

// Функція, яка викликається один раз при запуску програми
void setup() {
    // Встановлюємо піни 9 і 10 як вихідний і вхідний відповідно
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    // Встановлюємо пін 2 як вхідний
    pinMode(buttonPin, INPUT);
    // Ініціалізуємо серійний порт зі швидкістю 9600 бод
    Serial.begin(9600);
}

// Функція, яка викликається безперервно після функції setup()
void loop() {
    // Зчитуємо значення з входу кнопки
    buttonState = digitalRead(buttonPin);

    // Перевіряємо, чи натиснута кнопка
    // Якщо натиснута, то buttonState буде HIGH
    if (buttonState == HIGH) {
        // Подаємо високий сигнал на тригерний вивід датчика на 10
        мікросекунд
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        // Подаємо низький сигнал на тригерний вивід датчика
```

```
digitalWrite(trigPin, LOW);  
// Зчитуємо час, який потрібен для того, щоб ехо-сигнал повернувся  
назад  
duration = pulseIn(echoPin, HIGH);  
// Обчислюємо відстань за формулою, враховуючи, що швидкість  
звуку дорівнює 343 м/с  
distance = (duration / 2) * 0.0343;  
// Виводимо відстань на серійний порт  
Serial.print("Distance: ");  
Serial.print(distance);  
Serial.println(" cm");  
}  
// Інакше, якщо кнопка не натиснута  
else {  
    // Нічого не робимо  
}  
// Чекаємо півсекунди  
delay(500);  
}
```

### **III. Перевірка раніше засвоєних знань (10 хв.).**

Необхідно виконати наступний тест:

1. Яким символом або символами в мові C++ позначається початок і кінець умовного виразу в умовному операторі?

- a) Круглі дужки ()
- b) Квадратні дужки []
- c) Фігурні дужки {}
- d) Круглі дужки () і фігурні дужки {}

Відповідь: d).

2. Як називається умовний оператор, який дозволяє виконати один з двох блоків коду в залежності від результату умовного виразу?

- a) Оператор if
- b) Оператор if-else
- c) Оператор switch
- d) Оператор if-else

Відповідь: d).

3. Як називається умовний оператор, який дозволяє виконати один з декількох блоків коду в залежності від значення певної змінної або виразу?

- a) Оператор if
- b) Оператор if-else
- c) Оператор switch
- d) Оператор case

Відповідь: c).

4. Яким символом або символами в мові C++ позначається початок і кінець блоку коду в операторі switch?

- a) Круглі дужки ()
- b) Квадратні дужки []
- c) Фігурні дужки {}
- d) Круглі дужки () і фігурні дужки {}

Відповідь: c).

5. Яким словом або словами в мові C++ позначається кожен з варіантів в операторі switch?

- a) If
- b) Else
- c) Case
- d) Default

Відповідь: c).



6. Яким словом або словами в мові C++ позначається варіант за замовчуванням в операторі switch, який виконується, якщо жоден з інших варіантів не співпадає зі значенням виразу?

- a) If
- b) Else
- c) Case
- d) Default

Відповідь: d).

7. Яким словом або словами в мові C++ позначається завершення кожного варіанту в операторі switch, щоб уникнути переходу до наступного варіанту?

- a) End
- b) Stop
- c) Break
- d) Return

Відповідь: c).

8. Яким оператором або операторами в мові C++ можна створити умовний вираз, який перевіряє, чи виконується одна або декілька умов одночасно?

- a) Оператором або
- b) Оператором не
- c) [Оператором і
- d) Оператором або, оператором не

Відповідь: c).

9. Яким оператором або операторами в мові C++ можна створити умовний вираз, який перевіряє, чи виконується хоча б одна з декількох умов?

- a) Оператором або
- b) Оператором не
- c) Оператором і
- d) Оператором або, оператором не

Відповідь: а).

10. Яким оператором або операторами в мові C++ можна створити умовний вираз, який перевіряє, чи не виконується певна умова?

- a) Оператором або
- b) Оператором не
- c) Оператором і
- d) Оператором або, оператором не

Відповідь: b).

11. Який буде результат виконання наступного коду в мові C++?

```
int x = 10;
int y = 5;
if (x > y) {
    Serial.println("x is greater than y");
    if (x % y == 0) {
        Serial.println(" and x is divisible by y");
    }
} else if (x < y) {
    Serial.println("x is less than y");
    if (y % x == 0) {
        Serial.println(" and y is divisible by x");
    }
} else {
    Serial.println("x is equal to y");
}
```

- a) x is greater than y
- b) x is less than y
- c) x is equal to y
- d) x is greater than y and x is divisible by y

Відповідь: d).

#### IV. Підсумки уроку (2 хв.).

Клас 10

Дата проведення уроку \_\_.\_\_.20\_\_

Урок № 7/9.

Тема: Оператори циклу.

Мета:

**навчальна:** познайомити учнів з операторами циклу в мові програмування C++, математичними операторами, навчити створювати змінні різних типів;

**розвивальна:** розвивати алгоритмічне мислення, увагу, пам'ять;

**виховна:** виховувати дисципліну, зібраність, уміння сконцентруватися, позитивне ставлення учнів до навчально-пізнавальної діяльності.

Тип уроку: комбінований.

Програмне забезпечення: середовище програмування Arduino IDE.

Апаратне забезпечення: комп'ютер, плата Arduino Uno, USB-кабель.

#### План уроку (45 хв.)

1. Організаційний етап (2 хв.).
2. Повідомлення нового матеріалу (10 хв.).
3. Розв'язування задач на комп'ютері (30 хв.).
4. Підсумки уроку (2 хв.).

#### Хід уроку:

##### I. Організаційний етап (2 хв.).

##### II. Повідомлення нового матеріалу (10 хв.).

Оператори циклу в C++ дозволяють повторювати певний блок коду декілька разів, поки виконується певна умова. В C++ існують три види операторів циклу: for, while і do-while.

Оператор for має таку форму:

```
for (ініціалізація; умова; ітерація)
{
```

```

        // тіло циклу
    }

```

Він дозволяє встановити початкове значення змінної циклу, перевіряти умову на кожній ітерації і змінювати значення змінної циклу за правилом ітерації. Оператор `for` зручний для використання, коли відома кількість повторень циклу.

Оператор `while` має таку форму:

```

while (умова)
{
    // тіло циклу
}

```

Він виконує тіло циклу, поки умова істинна. Оператор `while` зручний для використання, коли кількість повторень циклу не відома заздалегідь, а залежить від даних.

Оператор `do-while` має таку форму:

```

do
{
    // тіло циклу
}
while (умова);

```

Він виконує тіло циклу один раз, а потім перевіряє умову. Якщо умова істинна, то виконує тіло циклу знову, і так далі. Оператор `do-while` зручний для використання, коли потрібно виконати тіло циклу принаймні один раз, незалежно від умови.

### **III. Розв'язування задач на комп'ютері (30 хв.).**

Перша програма.

Знайти суму перших  $n$  чисел за допомогою циклу `while`.

```

// змінна для зберігання кількості чисел
int n = 10;

```

```
// змінна для зберігання поточного числа
int i = 1;
// змінна для зберігання суми чисел
int sum = 0;

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

void loop() {
  // поки поточне число не перевищує кількість чисел
  while (i <= n) {
    // додати поточне число до суми
    sum = sum + i;
    // збільшити поточне число на одиницю
    i = i + 1;
  }
  // вивести суму чисел
  Serial.println(sum);
}
```

Друга програма.

Знайти суму перших n чисел за допомогою циклу do-while.

```
// змінна для зберігання кількості чисел
int n = 10;
// змінна для зберігання поточного числа
int i = 1;
// змінна для зберігання суми чисел
int sum = 0;
```

```
void setup() {  
    // встановити швидкість передачі даних  
    Serial.begin(9600);  
}  
  
void loop() {  
    // виконати один раз  
    do {  
        // додати поточне число до суми  
        sum = sum + i;  
        // збільшити поточне число на одиницю  
        i = i + 1;  
        // поки поточне число не перевищує кількість чисел  
    } while (i <= n);  
    // вивести суму чисел  
    Serial.println(sum);  
}
```

Третя програма.

Знайти суму перших n чисел за допомогою циклу do-while.

```
// змінна для зберігання кількості чисел  
int n = 10;  
  
// змінна для зберігання суми чисел  
int sum = 0;  
  
void setup() {  
    // встановити швидкість передачі даних  
    Serial.begin(9600);  
}
```

```
void loop() {  
    // виконати цикл від 1 до n  
    for (int i = 1; i <= n; i++) {  
        // додати поточне число до суми  
        sum = sum + i;  
    }  
    // вивести суму чисел  
    Serial.println(sum);  
}
```

#### **IV. Підсумки уроку (2 хв.).**

**Клас 10**

**Дата проведення уроку** \_\_.\_\_.20\_\_

**Урок № 8/9.**

**Тема:** Порівняння операторів циклу.

**Мета:**

**навчальна:** пзакріпити знання учнів з теми «Оператори циклу»;

**розвивальна:** розвивати алгоритмічне мислення, увагу, пам'ять;

**виховна:** виховувати дисципліну, зібраність, уміння сконцентруватися, позитивне ставлення учнів до навчально-пізнавальної діяльності.

**Тип уроку:** комбінований.

**Програмне забезпечення:** середовище програмування Arduino IDE.

**Апаратне забезпечення:** комп'ютер, плата Arduino Uno, USB-кабель, вібромотор, кнопка, потенціометр, ультразвуковий датчик, з'єднувальні проводи.

#### **План уроку (45 хв.)**

1. Організаційний етап (2 хв.).
2. Розв'язування задач на комп'ютері (30 хв.).

3. Перевірка раніше засвоєних знань (10 хв.).
4. Підсумки уроку (2 хв.).

### Хід уроку:

#### I. Організаційний етап (2 хв.).

#### II. Розв'язування задач на комп'ютері (30 хв.).

Перша програма.

Після натискання кнопки вібромотор має увімкнутись і нарощувати вібрацію в залежності від положення потенціометра, коли рівень вібрації буде рівний положенню потенціометра вібромотор має вимкнутись.

```
// пін, до якого підключений вібромотор
const int vibPin = 9;

// пін, до якого підключений потенціометр
const int potPin = A0;

// пін, до якого підключена кнопка
const int buttonPin = 2;

// змінна для зберігання значення потенціометра
int potValue = 0;

// змінна для зберігання значення вібрації
int vibValue = 0;

// змінна для зберігання стану кнопки
int buttonState = 0;

// змінна для зберігання попереднього стану кнопки
int lastButtonState = 0;

// змінна для зберігання стану вібромотора
bool vibState = false;

void setup() {
  // встановити пін вібромотора як вихідний
  pinMode(vibPin, OUTPUT);
```



```
// встановити пін кнопки як вхідний з підтягуючим опором
pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
    // зчитати значення потенціометра
    potValue = analogRead(potPin);
    // перетворити значення потенціометра на значення вібрації
    vibValue = map(potValue, 0, 1023, 0, 255);
    // зчитати стан кнопки
    buttonState = digitalRead(buttonPin);
    // якщо стан кнопки змінився
    if (buttonState != lastButtonState) {
        // якщо кнопка була натиснута
        if (buttonState == LOW) {
            // змінити стан вібромотора на протилежний
            vibState = !vibState;
        }
    }
    // зберегти попередній стан кнопки
    lastButtonState = buttonState;
    // якщо вібромотор увімкнений
    if (vibState) {
        // встановити початкове значення вібрації
        int currentVib = 0;
        // виконувати цикл, поки вібрація не буде рівна потенціометру або
        // кнопка не буде натиснута
        while (currentVib < vibValue && buttonState == HIGH) {
            // збільшити вібрацію на одиницю
            currentVib++;
        }
    }
}
```

```

// включити вібромотор з поточною вібрацією
analogWrite(vibPin, currentVib);
// зробити затримку 10 мілісекунд
delay(10);
// зчитати стан кнопки
buttonState = digitalRead(buttonPin);
}
// виключити вібромотор
analogWrite(vibPin, 0);
// змінити стан вібромотора на вимкнений
vibState = false;
}
}

```

Друга програма.

```

// піни, до яких підключений ультразвуковий датчик
const int trigPin = 3;
const int echoPin = 4;
// пін, до якого підключений вібромотор
const int vibPin = 9;
// пін, до якого підключений потенціометр
const int potPin = A0;
// пін, до якого підключена кнопка
const int buttonPin = 2;
// змінна для зберігання значення потенціометра
int potValue = 0;
// змінна для зберігання значення частоти вібрації
int freqValue = 0;
// змінна для зберігання значення відстані

```

```
int distance = 0;
// змінна для зберігання стану кнопки
int buttonState = 0;
// змінна для зберігання попереднього стану кнопки
int lastButtonState = 0;
// змінна для зберігання стану вібромотора
bool vibState = false;

void setup() {
    // встановити піни ультразвукового датчика як вихідні і вхідні
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    // встановити пін вібромотора як вихідний
    pinMode(vibPin, OUTPUT);
    // встановити пін кнопки як вхідний з підтягуючим опором
    pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
    // зчитати значення потенціометра
    potValue = analogRead(potPin);
    // обчислити значення частоти вібрації за формулою
    // freqValue = 1 + potValue / 10
    freqValue = 1 + potValue / 10;
    // зчитати стан кнопки
    buttonState = digitalRead(buttonPin);
    // якщо стан кнопки змінився
    if (buttonState != lastButtonState) {
        // якщо кнопка була натиснута
        if (buttonState == LOW) {
```

```
// змінити стан вібромотора на протилежний
vibState = !vibState;
}
}
// зберегти попередній стан кнопки
lastButtonState = buttonState;
// якщо вібромотор увімкнений
if (vibState) {
    // виконати цикл від 0 до freqValue
    for (int i = 0; i < freqValue; i++) {
        // змінна для зберігання часу проходження сигналу
        long duration = 0;
        // змінна для зберігання відстані
        int distance = 0;
        // встановити пін тригера в низький рівень
        digitalWrite(trigPin, LOW);
        // зробити затримку 2 мікросекунди
        delayMicroseconds(2);
        // встановити пін тригера в високий рівень
        digitalWrite(trigPin, HIGH);
        // зробити затримку 10 мікросекунд
        delayMicroseconds(10);
        // встановити пін тригера в низький рівень
        digitalWrite(trigPin, LOW);
        // зчитати час проходження сигналу з піна echo
        duration = pulseIn(echoPin, HIGH);
        // перетворити час на відстань
        distance = duration * 0.034 / 2;
        // якщо відстань менше 10 сантиметрів
        if (distance < 10) {
```

```
// включити вібромотор
digitalWrite(vibPin, HIGH);
}
// інакше
else {
    // виключити вібромотор
    digitalWrite(vibPin, LOW);
}
// зробити затримку 10 мілісекунд
delay(10);
// зчитати стан кнопки
buttonState = digitalRead(buttonPin);
// якщо кнопка була натиснута
if (buttonState == LOW) {
    // зупинити цикл
    break;
}
}
// виключити вібромотор
digitalWrite(vibPin, LOW);
// змінити стан вібратора на вимкнений
vibState = false;
}
}
```

### **III. Перевірка раніше засвоєних знань (10 хв.).**

1. Який оператор циклу використовується для повторення деякого коду, поки його умова істинна, тобто повертає true?

- a) for
- b) while

- c) do...while
- d) if

Відповідь: b).

2. Який оператор циклу називається циклом з параметром і має таку форму: for (ініціалізація; умова; приріст)?

- a) for
- b) while
- c) do...while
- d) if

Відповідь: a).

3. Який оператор циклу виконує набір інструкцій хоча б один раз, а потім перевіряє умову в кінці кожної ітерації?

- a) for
- b) while
- c) do...while
- d) if

Відповідь: c).

4. Як називається змінна, яка використовується для контролю за циклом і змінює своє значення після кожної ітерації?

- a) суматор
- b) лічильник
- c) індекс
- d) параметр

Відповідь: b).

5. Як називається кожний окремий прохід циклу?

- a) ітерація
- b) інкремент
- c) ініціалізація
- d) інструкція

Відповідь: a).

6. Як можна зупинити виконання циклу передчасно, незалежно від умови?

- a) використати оператор `break`
- b) використати оператор `continue`
- c) використати оператор `return`
- d) використати оператор `goto`

Відповідь: a).

7. Як можна пропустити поточну ітерацію циклу і перейти до наступної, незалежно від умови?

- a) використати оператор `break`
- b) використати оператор `continue`
- c) використати оператор `return`
- d) використати оператор `goto`

Відповідь: b).

8. Як можна реалізувати цикл `for` за допомогою циклу `while`?

- a) `while (ініціалізація; умова; приріст) {тіло циклу;}`
- b) `{ініціалізація; while (умова) {тіло циклу; приріст;}}`
- c) `for (while (умова) {тіло циклу;}) {ініціалізація; приріст;}`
- d) ніяк, це неможливо

Відповідь: b).

9. Як можна реалізувати цикл `while` за допомогою циклу `for`?

- a) `for (; умова;) {тіло циклу;}`
- b) `for (умова; ; ) {тіло циклу;}`
- c) `for (тіло циклу; умова; ) {}`
- d) ніяк, це неможливо

Відповідь: a).

10. Як можна реалізувати цикл `do...while` за допомогою циклу `for`?

- a) `for (bool b = true; b; b = умова) {тіло циклу;}`
- b) `for (тіло циклу; умова; ) {}`
- c) `for (; ; тіло циклу; if (умова) break;)`

d) ніяк, це неможливо

Відповідь: с).

**V. Підсумки уроку (2 хв.).**

**Клас 10**

**Дата проведення уроку** \_\_.\_\_.20\_\_

**Урок № 9/9.**

**Тема:** Покрокове виконання програм із циклами і розгалуженнями.

**Мета:**

**навчальна:** закріпити знання учнів про створення лінійних, розгалужених та циклічних алгоритмів;

**розвивальна:** розвивати алгоритмічне мислення, увагу, пам'ять;

**виховна:** виховувати дисципліну, зібраність, уміння сконцентруватися, позитивне ставлення учнів до навчально-пізнавальної діяльності.

**Тип уроку:** практична робота.

**Програмне забезпечення:** середовище програмування Arduino IDE.

**Апаратне забезпечення:** комп'ютер, плата Arduino Uno, USB-кабель, ультразвукові датчики, вібромотори, кнопки, потенціометри, з'єднувальні проводи.

**План уроку (45 хв.)**

1. Організаційний етап (1 хв.).
2. Розв'язування задач на комп'ютері (28 хв.).
3. Захист своїх проєктів (15 хв.).
4. Підсумки уроку (2 хв.).

**Хід уроку:**

**I. Організаційний етап (2 хв.).**

**II. Розв'язування задач на комп'ютері (28 хв.).**

За даними Всесвітньої організації охорони здоров'я (ВООЗ), в усьому світі понад 285 мільйонів людей страждає від порушень зору, з них 45



мільйонів незрячі. Приблизно 90% людей, які страждають від порушень зору, мешкають в країнах що розвиваються.

В Україні точної офіційної статистики щодо кількості людей з порушеннями зору немає, за неофіційними даними, – це приблизно 100 тисяч людей, з них понад 10 тисяч – діти.

Сліпота — це нездатність бачити. Провідні причини хронічної сліпоти включають катаракту, глаукому, вікову дегенерацію жовтої плями, помутніння рогівки, діабетичну ретинопатію, трахому і захворювання очей у дітей. У світі все ширше зустрічається сліпота, що розвивається з віком, а також сліпота, викликана неконтрольованим діабетом.

Інтернет магазини пропонують великий вибір товарів, призначених для навчання і комфортного життя людей з порушеннями зору. Це і мовні пристрої для незрячих; навчальні засоби для сліпих; засоби навчання, письма та друку шрифтом Брайля; друкарські машинки зі шрифтом Брайля; лінзи Френеля; тифлоприлади; ігри для незрячих та слабозрячих.

Усі товари є високоякісними, зручними у використанні, відповідають міжнародним вимогам, але є дороговартісними.

Вашою задачею буде створити смарт пристрій для людей з вадами зору. Основними елементами в пристрої є плата Arduino Uno, три ультразвукові датчики та три вібротори.

Готовий пристрій виглядає наступним чином: три ультразвукові датчики розташовані по колу з переду, з лівої і правої сторін, з їх внутрішньої сторони під'єднані вібротори. Пристрій одягається на голову і у при винекненні перешкоди вмикається вібротор з тієї сторони, з якої знаходиться прешкода.

Створіть вказаний пристрій і продумайте можливі модифікації з використанням кнопки або потенціометра як вже робили на минулих уроках.

Об'єднайтесь в групи по 3-4 чоловіки. В кінці уроку кожна команда має розказати як саме організували роботу пристрою.

**III. Захист проєктів (15 хв.).**

**IV. Підсумки уроку (2 хв.).**