НАЦІОНАЛЬНА АКАДЕМІЯ ПЕДАГОГІЧНИХ НАУК УКРАЇНИ
ІНСТИТУТ ЦИФРОВІЗАЦІЇ ОСВІТИ

Крамар С.С.

ARDUINO ДЛЯ ВЧИТЕЛЯ ІНФОРМАТИКИ: ВІРТУАЛЬНЕ МОДЕЛЮВАННЯ ТА ПРОГРАМУВАННЯ В TINKERCAD

Київ - 2025

УДК: 004.4:681.3.06:373.3/.5
В 43

Рекомендовано до друку
Вченою радою Інституту цифровізації освіти НАПН України
(протокол № 2 від 30 січня 2025 р.).

Рецензенти:

Мар'єнко М.В., доктор педагогічних наук, старший науковий співробітник, провідний науковий співробітник відділу хмаро орієнтованих систем і штуцчного інтелекту в освіті Інституту цифровізації освіти НАПН України.

Олексюк В.П., доктор педагогічних наук, професор, професор кафедри інформатики та методики її навчання Тернопільського національного педагогічного університету.

В 43 Arduino для вчителя інформатики: Віртуальне моделювання та програмування в TinkerCad: практ. посіб. / Крамар С.С., Київ : ІЦО НАПН України, 2025. 71 с.

ISBN 978-617-8330-51-4

Посібник створений з метою забезпечення педагогів доступними, практично орієнтованими відомостями для підвищення власної професійної компетентності в умовах цифровізації освіти. У посібнику подано комплекс теоретичних і практичних матеріалів, які дозволяють вчителю інформатики ознайомитися з основами роботи з мікроконтролерами Arduino, створенням електронних схем у середовищі Tinkercad Circuits, а також реалізувати навчальні проєкти. Матеріали посібника адаптовані для використання у неформальній освіті, самостійного опрацювання або як методичний супровід до серії тренінгових занять. Практична частина посібника подана у формі занять, які відповідають сучасним принципам проєктно-орієнтованого навчання. Окрему увагу приділено методичним підходам до організації роботи з учнями, використанню хмарних сервісів та інтеграції цифрових технологій у навчальний процес. Для педагогічних, науково-педагогічних, наукових працівників, слухачів курсів підвищення кваліфікації, всіх, хто цікавиться перспективами використання платформи Arduino в освіті.

УДК 004.4:681.3.06:373.3/.5

ISBN 978-617-8330-51-4

© Інститут цифровізації освіти НАПН України, 2025
© Крамар С.С., 2025

ЗМІСТ

ВСТУП 3

РОЗДІЛ І. ВИКОРИСТАННЯ ARDUINO ТА TINKERCAD CIRCUITS У ПІДВИЩЕННІ ЦИФРОВОЇ КОМПЕТЕНТНОСТІ ВЧИТЕЛІВ ІНФОРМАТИКИ 5

1.1 Актуальність використання Arduino в освіті 5
1.2 Освітній потенціал використання Arduino та Tinkercad для підвищення кваліфікації вчителів 7
1.3 Методичні підходи до впровадження Arduino в неформальну освіту вчителів 12

РОЗДІЛ ІІ. ОПАНУВАННЯ ОСНОВ ВИКОРИСТАННЯ ARDUINO У СЕРЕДОВИЩІ TINKERCAD CIRCUITS 15

2.1 Огляд середовища TinkerCad Circuits 15
2.1.1. Реєстрація та налаштування облікового запису 15
2. 1.2. Інтерфейс середовища, основні інструменти 17
2. 1.3. Додавання електронних компонентів 18
2.1.6. Вправи. 18

2.2 Основи електроніки для роботи з Arduino 21
2. 2.1. Базові електричні поняття (струм, напруга, опір) 21
2.2.2. Робота з резисторами, світлодіодами, кнопками 25
2.2.3. Аналогові та цифрові сигнали 30
2.2.4. Практичні вправи: складання простих схем 34

2.3 Основи програмування Arduino в TinkerCad 40
2.3.1. Структура скетча (setup(), loop()) 40
2.3.2. Основні команди для керування світлодіодами 42
2.3.3. Використання змінних та умовних операторів 45
2.3.4. Робота з циклами for, while 49
2.3.5. Практичні завдання 54

РОЗДІЛ ІІІ. ПРАКТИЧНІ ПРОЄКТИ 57

3.1. Проєкт 1. "Мигаючий світлодіод" 57
3.2. Проєкт 2. "Кнопка для керування світлодіодом" 59
3.3. Проєкт 3. Регулювання яскравості світлодіода (PWM) 61

РЕКОМЕНДАЦІЇ 64

ВИСНОВКИ 66

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 67

ВСТУП

В умовах глобалізації, євроінтеграції та стрімкої цифрової трансформації різних сфер життя виникає потреба у створенні конкурентоспроможної освітньої сфери України, розвитку сучасних цифрових компетентностей і кваліфікацій, що потребує підвищення рівня доступності та якості підготовки вчителя інформатики. Саме фахова компетентність вчителя інформатики є ключовим чинником підготовки підростаючого покоління, здатного адаптуватися до динамічних суспільно-економічних змін, критично мислити, ефективно вирішувати фахові й повсякденні задачі із залученням сучасних технічних досягнень і цифрових технологій, прагнути сталого саморозвитку та бути успішними в обраній професії.

Цифрова трансформація освіти вимагає від сучасного вчителя інформатики не лише володіння базовими цифровими навичками, а й здатності інтегрувати новітні технології у свою педагогічну практику. Робототехніка, мікроелектроніка, хмарні обчислення, віртуальне моделювання – ці технології формують нове середовище навчання, в якому учень не є пасивним споживачем інформації, а активним її творцем. Водночас вчитель має стати не лише наставником, а й фасилітатором творчого й дослідницького пошуку учнів.

Arduino – це проста у використанні, доступна платформа з відкритим кодом, яка дозволяє створювати електронні проєкти, програмувати пристрої, моделювати фізичні явища та експериментувати з автоматизованими системами. У поєднанні з онлайн-середовищем Tinkercad Circuits, ця платформа стає потужним інструментом для навчання як в аудиторії, так і в дистанційному форматі. Завдяки простоті інтерфейсу, наявності вбудованого емулятора та підтримці командної роботи, Tinkercad стає незамінним засобом у підвищенні цифрової компетентності педагогів.

Посібник, який ви тримаєте в руках, покликаний заповнити прогалину між зростаючими потребами в цифровій підготовці вчителів та відсутністю системних, методично обґрунтованих матеріалів, орієнтованих на практику. Його структура поєднує теоретичне підґрунтя, огляд сучасних підходів до використання Arduino в освіті та покрокові інструкції щодо реалізації занять. Він базується на дослідженнях у сфері педагогіки, ІКТ в освіті та практичному досвіді впровадження Arduino в тренінгових програмах для вчителів.

Ми переконані, що запропоновані матеріали стануть у пригоді як вчителям-практикам, які прагнуть розширити власні цифрові горизонти, так і

методистам, які займаються організацією курсів підвищення кваліфікації. Arduino – це не лише про техніку, це про можливість змінити освітнє середовище, зробити його живим, динамічним, орієнтованим на майбутнє. І саме вчитель є тією рушійною силою, яка здатна привести освіту до інноваційних змін.

РОЗДІЛ І. ВИКОРИСТАННЯ ARDUINO ТА TINKERCAD CIRCUITS У ПІДВИЩЕННІ ЦИФРОВОЇ КОМПЕТЕНТНОСТІ ВЧИТЕЛІВ ІНФОРМАТИКИ

1.1 Актуальність використання Arduino в освіті

У сучасному світі, де стрімко розвиваються цифрові технології, особливо актуальним є оновлення змісту освіти відповідно до вимог інформаційної епохи. Незважаючи на швидке поширення штучного інтелекту, автоматизації та роботизованих систем, рівень знань про сучасні технології у школах і закладах вищої освіти часто залишається недостатнім. Це стримує формування науково-технічного світогляду здобувачів освіти та ускладнює практичне застосування набутих знань.

Одним із ефективних засобів модернізації викладання інформатики, фізики та інших предметів природничо-математичного циклу є використання апаратно-програмного комплексу Arduino разом із онлайн-середовищем Tinkercad Circuits. Ці платформи забезпечують практичну взаємодію з сучасними технологіями, поєднуючи програмування, електроніку та елементи інженерної діяльності.

У новітньому інформаційно-освітньому середовищі формуються моделі організації навчальної діяльності, засновані на інноваційних технологічних рішеннях. Вони включають використання апаратно-програмних засобів, хмарних сервісів і технологій робототехніки, що стають важливими елементами інфраструктури освітнього процесу. Ці засоби слугують не лише інструментами передачі знань, а й каталізаторами впровадження інноваційних методів і підходів до навчання.

Актуальним є питання адаптації інформаційно-технологічного середовища до потреб користувачів — зокрема, викладачів і учнів — з метою максимального розкриття педагогічного потенціалу сучасних ІКТ. Це включає створення умов для покращення результатів навчання, розвитку дослідницьких навичок педагогів, а також стимулювання командної роботи. Досягнення таких цілей потребує впровадження інноваційних підходів до організації доступу до освітнього програмного забезпечення, зокрема хмаро орієнтованих рішень, які наразі вважаються провідними в освітній галузі.

У цьому контексті особливого значення набуває неформальна освіта вчителів, яка передбачає гнучкі моделі підвищення кваліфікації з використанням Arduino. Цей комплекс дозволяє вчителям краще зрозуміти як програмну, так і апаратну складову цифрових технологій. Поєднання навчання на платформі Tinkercad з дистанційними консультаціями, контрольними заходами та постійним оновленням змісту за допомогою хмарних сервісів створює передумови для формування ІКТ-компетентності нового рівня.

Особливої уваги потребує науково-методичне обґрунтування ефективних форм і методів навчання, які передбачають використання апаратно-програмних засобів у поєднанні з інноваційними педагогічними технологіями. Розробка спеціальних методик і сценаріїв навчання робототехніки є необхідною умовою підвищення якості освіти, зокрема у сфері STEM.

Використання Arduino в освітньому процесі особливо актуальне у зв’язку з потребою підвищення кваліфікації вже працюючих педагогів. Сучасний вчитель має бути не лише обізнаним у сфері цифрових технологій, але й здатним інтегрувати їх у свою педагогічну діяльність. Запровадження Arduino з використанням хмарних технологій є ефективним

засобом досягнення цієї мети. Воно створює підґрунтя для професійного зростання вчителя, розвитку його цифрових і інженерних компетентностей, а також готовності впроваджувати інновації у навчальний процес.

Зважаючи на зростаючий попит на навчальні програми з робототехніки, спостерігається певна суперечність: з одного боку — потреба у широкій інтеграції Arduino в освіту, з іншого — нестача методичних розробок для ефективної підготовки педагогів. Тому вкрай важливим є створення і поширення практично орієнтованих методичних матеріалів, які б допомогли вчителям ефективно використовувати Arduino у своїй діяльності.

Платформа Arduino має значний потенціал для навчання електроніки, програмування та розв'язання міждисциплінарних завдань. Вчитель, який володіє методиками використання цієї платформи, здатен зробити навчання більш захопливим і результативним, розвивати у школярів технічне мислення, креативність та навички командної роботи. Однак для цього потрібне системне навчання, адже багато педагогів стикаються з труднощами в освоєнні принципів роботи Arduino через недостатній рівень підготовки у цій сфері.

Таким чином, забезпечення високої якості підготовки вчителів до використання Arduino та хмарних платформ, зокрема Tinkercad Circuits, є важливою умовою успішної цифрової трансформації освіти. Це дозволить сформувати у педагогів належний рівень цифрової та ІКТ-компетентності, сприятиме їхній професійній самореалізації в умовах високотехнологічного суспільства і створить умови для інтеграції інноваційних освітніх підходів у повсякденну педагогічну практику.

1.2 Освітній потенціал використання Arduino та Tinkercad для підвищення кваліфікації вчителів

Arduino — це програмована електронна платформа з відкритим кодом, яка включає апаратну і програмну складові. Її перевага полягає у простоті використання, доступності, безпечності та можливості адаптації під різні цілі. У неформальній освіті вчителів Arduino виконує роль інструменту для розвитку цифрової компетентності: від розуміння принципів роботи мікроконтролерів до створення прикладних навчальних проєктів. Tinkercad Circuits, як хмарне середовище моделювання електронних схем, дозволяє вчителям без спеціального обладнання вивчати основи схемотехніки та програмування Arduino у зручному форматі.

Платформа Tinkercad вже багато років показує себе як стабільна, доступна, логічно скомпонована платформа для навчання (Рис1). За умов використання хмарних сервісів, дані зберігаються у центрі обробки даних, а не на локальному комп'ютері користувача, тоді як доступ до них забезпечується через браузер, є можливим з різних пристроїв, з яких можна вийти в Інтернет. За хмарної моделі організації доступу до ІКТ виникають необхідні умови для формування навичок командної роботи, які потрібні сучасному фахівцеві, що має бути обізнаним з використанням ІТ. Вчителі, які опановували даний комплекс, мали зареєструватися у хмаро орієнтованому середовищі, отримати акаунт, завдяки якому вони мали доступ до готових проектів з поясненням, сервісів як для індивідуального, так і колективного використання.

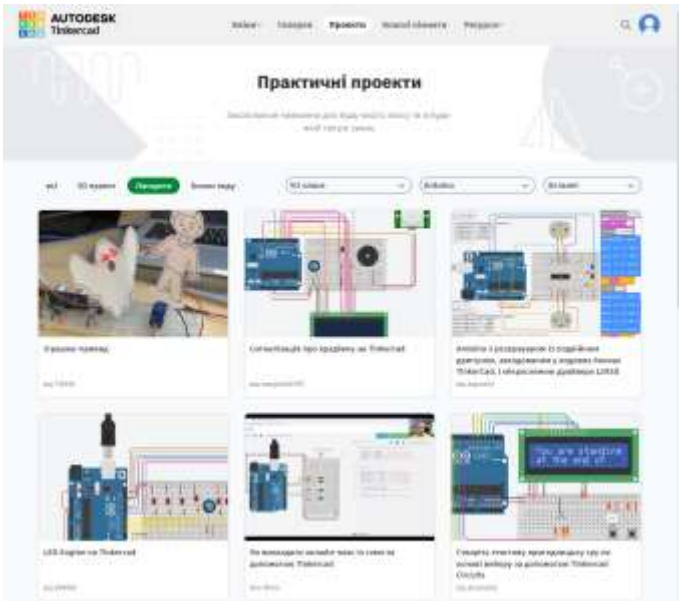


Рис 1. Хмаро орієнтоване середовище Tinkercad

Найважливішою особливістю *програмно-апаратного комплексу Arduino,* з точки зору педагогічного використання, є вільний доступ, безпека та надійність зберігання інформації, контроль прав доступу, здатність легко адаптувати під різні цілі та задачі.

В умовах формування інформаційного суспільства важливо підготувати висококваліфікованих фахівців, здатних правильно та доцільно навчити використовувати комплекс так, щоб це було змістовно та цікаво. Тому необхідно шукати нові методичні підходи до організації навчання, які б сприяли глибокому засвоєнню та розумінню

основних понять, правил, принципів і методів вивчення дисциплін, їх взаємозв'язку з суміжними дисциплінами та способів їх використання на практиці.

Для організації групової роботи з вчителями була використана платформа *Tinkercad*, але з інтегрованими класами, оскільки в даній платформі реалізовано наступні можливості: корегувати дії викладача, вести групове та індивідуальне спілкування, робити нотатки, відслідковувати прогрес успішності, зберігати роботи. Застосування програмно-апаратного комплексу не спричинить вчителям суттєвих труднощів у вирішенні поставлених завдань, для роботи потрібно лише мати доступ до мережі інтернет. Однак для користування хмарними сервісами необхідно пройти короткий інструктаж роботи з ними(Рис2).

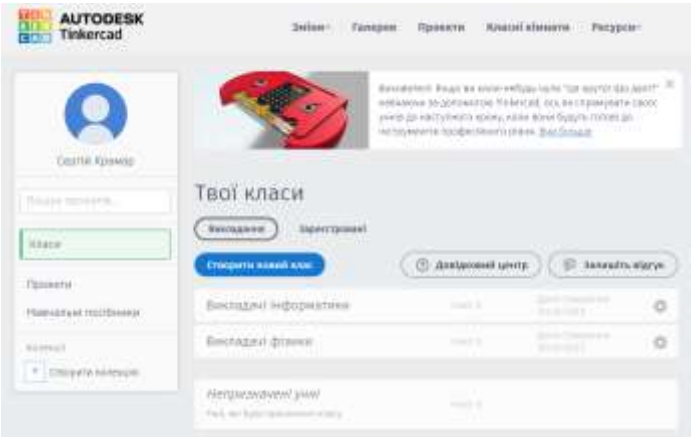


Рис 2. Приклад платформи Tinkercad з інтегрованими класами

Мета запровадження *програмно-апаратного комплексу Arduino* у процесі підготовки фахівців - формування здатності до успішного використання інформаційних технологій у своїй професійній діяльності, творчого підходу до вирішення нестандартних проблем, глибокого оволодіння основами дисципліни. З цією метою була розроблена методика використання *комплексу* у неформальній освіті вчителів, спрямована на (і) формування професійних компетентностей вчителів інформатики та фізики, що дасть можливість успішно адаптуватися до вимог інформаційного суспільства; розвиток творчого підходу до вирішення нестандартних завдань; та (iii) формування навичок використання ІКТ, необхідних для аналізу, моделювання та вирішення теоретичних та практичних задач у професійній діяльності.

Завдяки впровадженню *комплексу* у навчальний процес вчителів з'являється можливість зосередити увагу на принципах, підходах, звільнити час та зусилля, які витрачаються на створення навчального матеріалу, використання платформи значно покращує процес навчання(Рис3).

Рис 3. Приклад елемента навчального процесу на хмарній платформі Tinkercad

Можливості використання *програмно-апаратного комплексу* для вирішення навчальних задач досить широкі. Вчитель, використовуючи хмарні сервіси, вирішує поставлену перед ним задачу, і, таким чином, він не має перешкод у застосуванні сучасних засобів ІКТ, а крім того, усвідомлює потрібність корисність платформи для успішної реалізації професійних цілей. Рішення задач прикладного характеру за допомогою хмарних сервісів забезпечує можливість формування професійних компетентностей.

Основною перевагою застосування програмно-апаратного комплексу Arduino при проектуванні середовища навчання є розширення спектру засобів для роботи із сучасним обладнанням на онлайн платформі, до якої матимуть доступ всі учасники процесу навчання. Використання цього комлпексу позитивно впливає на:

– формування навичок роботи в середовищі програмування мікроконтролерів;

– розвиток знань щодо структури програми та її елементів, змінних, виразів, масивів, логічних конструкцій, функцій, бібліотек тощо;

– удосконалення навичок написання програмного коду відповідно до поставленого завдання та перенесення його у середовище мікроконтролера;

– опанування основних понять з електрики;

– формування знань про основні елементи цифрових схем;

– удосконалення навичок розуміння, модифікації та конструювання електричних схем відповідно до пройденого матеріалу;

– формування навичок роботи із датчиками, їх налаштування, опрацювання даних з них та зчитування документації [4; 5].

Використання Arduino має беззаперечні переваги, а саме:

– спрощує процес роботи з мікроконтролерами у порівнянні з іншими пристроями для вчителів;

– плати Arduino відносно дешевші в порівнянні з іншими платформами;

– плати Arduino кросплатформені (можна працювати під управлінням ОС Windows, Mac OS і Linux);

– система Arduino має просте і зрозуміле середовище програмування.

– середовище програмування Arduino є додатком, що охоплює редактор коду, компілятор і спеціальний модуль для прошивки плати;

– мова програмування, що використовується в Arduino, є реалізацією Wiring, тобто це C / C ++, доповнений деякими бібліотеками;

– можливості плат Arduino можна розширити за допомогою особливих мікросхем, які називають «шилдами» (від англ. shields). Шилди встановлюються поверх основної плати і дають нові можливості. [7]

Основні риси функціоналу та особливості Tinkercad Circuits, що роблять доцільним її запровадження у середовище навчання вчителів, наступні: це онлайн платформа, тож для роботи потрібен лише Інтернет; є у наявності зручний графічний редактор для візуальної побудови електронних схем; доступним є набір попередньо встановлених моделей найбільш популярних електронних компонентів, відсортованих за типами компонентів; платформа містить симулятор електронних схем, за допомогою якого можна підключити створений віртуальний пристрій до віртуального джерела живлення і простежити, як воно буде працювати; можна також використати симулятор датчиків та інструментів зовнішнього впливу. Є можливості змінювати покажчики датчиків, стежити за тим, як в залежності від цього змінюються параметри системи; зручно використовувати вбудований редактор Arduino з монітором порту і можливістю налагодження; є вже розроблені для розгортання проекти Arduino зі схемами і кодом; візуальний редактор коду Arduino; дає можливість інтеграції з рештою функціональністю Tinkercad і швидкого створення для вашого пристрою корпусу та інших конструктивних елементів; створена модель може бути відразу відправлена на 3D-принтер; вбудовані підручники і величезне співтовариство з колекцією готових проектів.

Головне, що не потрібно завантажувати Arduino IDE, не потрібно шукати і викачувати популярні бібліотеки і sketch, не потрібно збирати схему і підключати плату – все знаходиться відразу на одній сторінці.

Переваги використання Tinkercad: безкоштовний онлайн сервіс; найпопулярніші компоненти для Arduino вже є у симуляторі; багато схем створено (можна їх дописувати і змінювати); ділитися проектом з іншими людьми; постійне оновлення й доповнення; вікно написання коду не відрізняється від вікна Arduino IDE.

Недоліки використання Tinkercad: не вистачає компонентів, які створюються швидше, ніж встигають створити симулятор цього компонента. [7]

Запропонована методика використання програмно апаратного комплексу Arduino на платформі Tinekrcad була апробована в освітньому процесі і виявилася досить зручною у використанні. Лекційні, практичні та лабораторні роботи проводились дистанційно, за допомогою одного з месенджерів для зв’язку з викладачем та за допомогою демонстрування робочого стола для чіткості та якісності подачі інформації. Кожне з цих занять було записане і потім завантажене в GoogleDrive, на якому в будь яку хвилину можна було переглянути його, якщо щось не зрозуміло або виникли якісь питання в завданнях або лабораторних роботах.

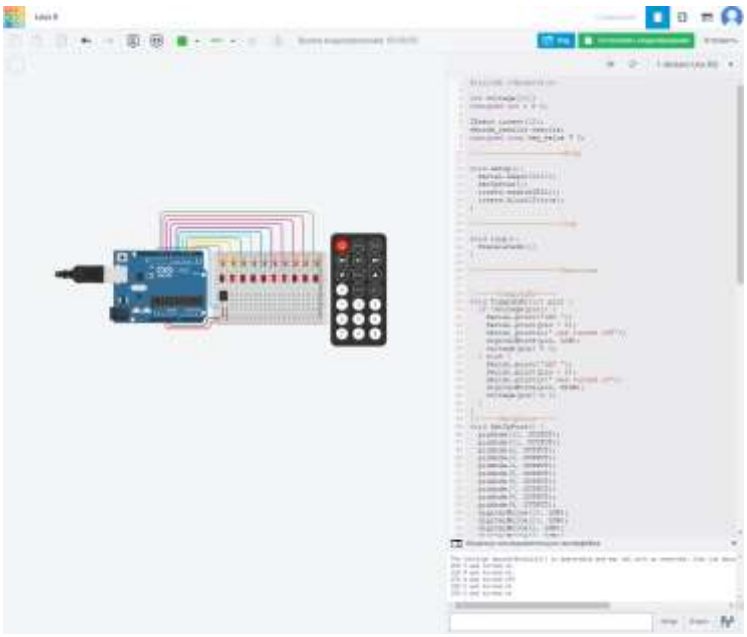


Рис 4. Приклад заняття на платформі Tinkercad

Для перевірки ефективності розробленої методики навчання було виконано порівняння навчальних досягнень вчителів за рівнями підготовки за розробленою методикою і відзначено позитивні показники успішності з опанування платформи. Експериментальна база дослідження: Дніпровський науковий ліцей інформаційних технологій, м.Дніпро. Було сформовано фокус-групу з 22 вчителів Дніпровського наукового ліцею інформаційних технологій, для якої було проведено вхідне і вихідне опитування, яке показало підвищення рівня ІКТ компетентності вчителів щодо використання засобів робототехніки, на основі комплексного показника за основними критеріями – діяльнісним, когнітивним, мотиваційним і рефлексивним.

Проведене дослідження свідчить, платформа Tinkercad є ефективним засобом для організації навчання вчителів інформатики та фізики використанню програмно-апаратного комплексу Arduino. Завдяки опануванню низки навчальних тем і проведення творчих робіт, які можна було організувати на єдиній платформі, до якої викладачі і вчителі могли отримувати доступ у будь-якому місці і у будь-який час, значно розширилися можливості організації якісного навчання; розширився доступ до електронних ресурсів; підвищився рівень організації навчального процесу завдяки структуруванню матеріалу і підтримуванню ресурсів для вивчення в актуальному стані.

Завдяки використанню хмарних технологій можна сформувати поліфункціональне навчальне середовище на єдиній основі, завдяки чому вдається досягти активізації освітнього процесу, формувати у вчителів мотивацію, орієнтування та обізнаність щодо роботи з навчальними проектами, вміння ефективно опрацьовувати значні обсяги даних і відомостей, раціонально організовувати час і наявні ресурси, технічно правильно та доцільно розповідати про датчики та їх особливості, розуміти відмінності датчиків один від одного, орієнтуватись у бібліотеках та знати як їх встановлювати. Всі ці навички є необхідними для повноцінного існування і самореалізації майбутнього фахівця в інформаційному суспільстві, що відкриває широкі можливості для особистісного розвитку і самореалізації.

Визначено наступні найбільш доцільні шляхи використання Arduino у неформальній освіті вчителів:

- *Проектна робота з Arduino:* використання Arduino для реалізації проектів, в яких поєднуються програмування та електроніка, розвиваються індивідуальні та командні комунікативні навички.
- *Наукові дослідження*: Arduino дозволяє проводити наукові експерименти, такі як аналіз впливу кліматичних умов на рослини за допомогою датчиків температури, вологості та світла.
- *Розробка ігор*: створення ігор на базі Arduino сприяє зміцненню навичок програмування та роботи з електронікою.
- *Автоматизація та робототехніка*: можливість створювати автоматизовані системи, використовуючи Arduino для керування та моніторингу, вивчаючи основи інженерії та системного аналізу.
- *Інтерактивні арт-проекти*: за допомогою Arduino можна реалізувати інтерактивні арт-інсталяції, які відгукуються на зміни у середовищі, що збагачує навчальний процес в естетичному вимірі.

1.3 Методичні підходи до впровадження Arduino в неформальну освіту вчителів

Ефективне використання Arduino в підвищенні кваліфікації вчителів ґрунтується на кількох методичних підходах: компетентнісному, конструктивістському, проєктно-орієнтованому. До основних принципів належать: «навчання через діяльність», практико-орієнтованість, гнучкість і доступність. У фокусі – організація практичних занять, хакатонів, міні-проєктів, що дозволяють інтегрувати знання з інформатики, математики, фізики. Значну увагу приділяють STEM-інтеграції, використанню віртуальних лабораторій, симуляторів та освітніх платформ.

Використання Arduino та Tinkercad Circuits у неформальній освіті вчителів дозволяє:
– формувати практичні навички з програмування та роботи з електронними пристроями;
– сприяти розвитку творчого, критичного та інженерного мислення;
– забезпечити міждисциплінарну інтеграцію знань;
– підвищити мотивацію вчителів до використання ІКТ;
– створити підґрунтя для впровадження STEM-освіти в школі.

Один з методів використання Arduino у навчальному процесі є проведення навчальних проектів. Розроблення проектів з Arduino дає можливість вчителю залучити учнів до практичного застосовувати знань з різних галузей, таких як програмування, електроніка, робототехніка. Ці проекти можуть бути як індивідуальними, так і командними, що дозволяє розвивати комунікативні навички учнів.

Крім того, Arduino можна використовувати для проведення дослідів у різних наукових галузях. Наприклад, використання датчиків температури та вологості може бути корисним для дослідження рослинного росту, датчика світла - для дослідження впливу світла на рослини. Ще одним методом використання Arduino в освіті є створення ігрових програм. Розроблення ігор є гарною галуззю для розвитку навичок програмування та роботи з електронікою. Такі ігри можуть бути як індивідуальними, так і командними, та дають можливість розвивати креативність та логічне мислення [3].

Крім того, запровадження Arduino в навчальний процес може бути досить корисним у вихованні ціннісного ставлення до навколишнього світу та екологічної свідомості.

Вчитель може створювати проекти, що дають можливість розуміти вплив технологій на довкілля та сприяти розвитку екологічної культури. Для того, щоб використання Arduino у навчальному процесі було успішним, необхідно враховувати деякі методичні підходи, а саме:

Проектно-орієнтоване навчання (Project-Based Learning, PBL). Вчитель пропонує учням реальні завдання або проблеми, які можна вирішити за допомогою Arduino. Наприклад, створення системи розумного освітлення або датчика температури. Такий підхід дозволяє учням інтегрувати знання з кількох предметів (інформатики, фізики, математики) і отримати практичний досвід.

Дослідницький підхід. Учні самостійно досліджують можливості Arduino для збору та аналізу даних і роблять досліди. Наприклад, вимірювання рівня шуму в різних приміщеннях школи або моніторинг вологості ґрунту для біологічних експериментів.

Занурення у STEM та міжпредметні зв’язки. Arduino дозволяє інтегрувати різні науки в межах одного уроку. Вчитель може організувати заняття, де учні проектують та програмують пристрої, які потребують знань з фізики (наприклад, закони руху) або математики (наприклад, розрахунок електричного опору).

Інтерактивні воркшопи та хакатони. Залучення учнів до змагань або хакатонів з використанням Arduino сприяє розвитку навичок командної роботи, креативного мислення та інновацій.

Гейміфікація. Вчитель може застосувати ігрові елементи на уроках, наприклад, запропонувати учням створити гру або робота, який виконує певні рухи. Це підвищує мотивацію до вивчення електроніки та програмування.

Робота з готовими наборами Arduino. Багато виробників пропонують навчальні комплекти з Arduino, які включають інструкції для учнів різних рівнів. Це дає можливість поступового переходу від простих проєктів до складніших.

Отже, можна стверджувати, що використання Arduino в освіті є досить ефективним інструментом для розвитку неформальної освіти учителів, що дозволить їм значно підвищити ефективність навчання учнів. Проекти з використанням Arduino дають можливість учням практично застосовувати знання з різних областей, таких як програмування, електроніка та механіка, а також розвивати комунікативні та креативні навички. Однак, для того, щоб використання Arduino було максимально ефективним, необхідно дотримуватися певних методичних вимог, таких як врахування інтересів та потреб учнів, застосування наочності та реалізації проектів, а також володіння достатнім рівнем знань в галузі програмування та електроніки. Необхідною умовою успішного використання Arduino в навчальному процесі є наявність необхідних матеріальних та технічних ресурсів. Вчителі повинні мати доступ до необхідних матеріалів для побудови проектів з використанням Arduino, таких як електронні компоненти, сенсори та інші засоби зв'язку. Також вчителі повинні мати доступ до необхідного програмного забезпечення та обладнання для програмування та налагодження проектів.

Розуміння потреб вчителів. Перш ніж розробляти методику навчання вчителів, зокрема у неформальній освіті, необхідно ретельно зрозуміти потреби та очікування вчителів, наприклад, інформатики та фізики. Важливо з'ясувати, які теми робототехніки заслуговують більшої уваги за своєї складності, можливо є теми які вони хотіли б включити до своїх курсів, які труднощі виникли. Це допоможе побудувати методику, яка б максимально відповідала потребам вчителів і забезпечувала якісне навчання робототехніки та програмування.

Важливо використовувати сучасні технології та додаткові ресурси для покращення якості навчання робототехніці. Застосування віртуальної реальності, 3D-друку, симуляційних програм дозволить отримати більш глибоке розуміння робототехнічних концепцій та забезпечує їх активну практику [7]. Також слід залучатися до роботи з позашкільними та науковими організаціями, де буде можливість отримати додаткові ресурси, ідеї та підтримку.

Використання Arduino в освіті є досить перспективним напрямом, який дає можливість розвивати творчі та інноваційні навички учнів. Вчителі мають застосовувати цей інструмент у навчальному процесі. Такий підхід сприятиме розвитку неформальної освіти учнів та формуванню їх творчої та креативної особистості, що є ключовими факторами у формуванні кваліфікованих та компетентних фахівців в галузі технологій та науки.

Також важливим фактором успіху є активна робота вчителів з учнями та їхнє зацікавлення у навчанні. Вчителі повинні бути готові відповісти на запитання учнів, пояснювати складні концепції та підтримувати інтерес до навчання. Крім того, важливо враховувати індивідуальні особливості кожного учня та розвивати їх здібності та інтереси.

Використання Arduino може бути корисним для розвитку не тільки технічних, а й соціальних навичок учнів. Наприклад, учні можуть створювати проекти, які допоможуть вирішувати соціальні проблеми у їхній громаді. Такі проекти можуть залучити учнів до вирішення реальних проблем та покращення життя людей навколо них.

Вчителі повинні мати достатній рівень знань у галузі програмування та електроніки, використовувати методичні підходи для розвитку навчання та мати доступ до необхідних матеріальних та технічних ресурсів. Активна робота вчителів, зацікавлення та розвиток соціальних навичок - ключові фактори успіху у використанні Arduino в освіті.

Крім того, використання Arduino може бути корисним і для розвитку критичного мислення. Наприклад, при створенні проектів з використанням Arduino учні повинні розуміти принципи роботи сенсорів, актуаторів, а також базові концепції програмування. Це допомагає розвивати їхнє критичне мислення та здатність до аналізу та розв'язання проблем.

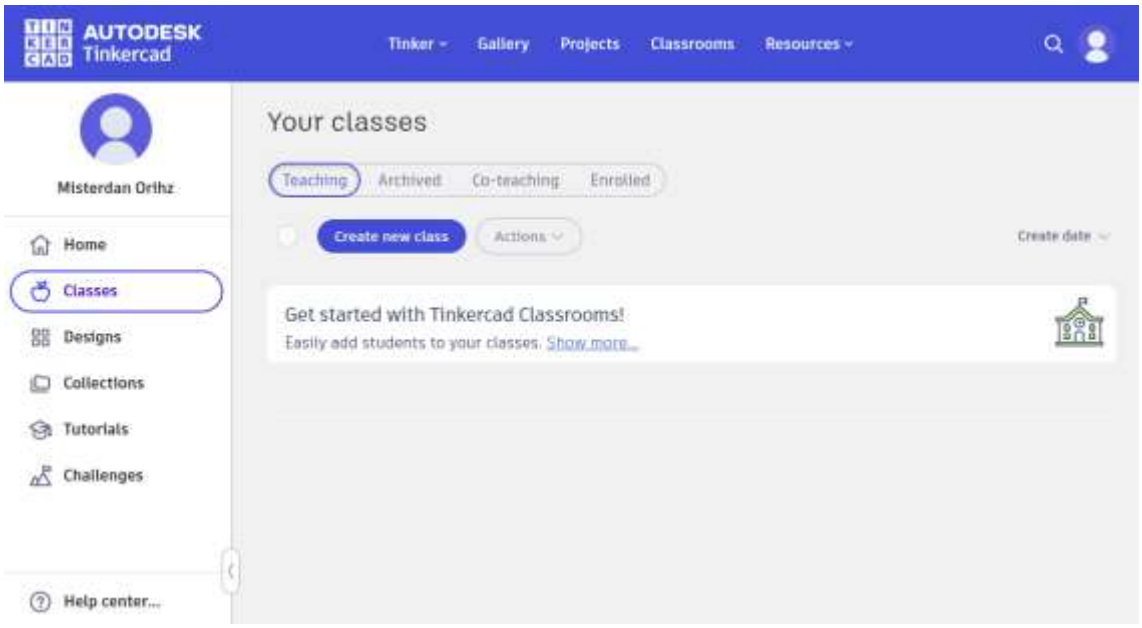
Одним із важливих аспектів використання Arduino в освіті є його відкритість та доступність. Arduino є відкритою платформою з відкритим вихідним кодом, що дозволяє розробникам створювати нові проекти та додатки[6]. Крім того, вартість компонентів та пристроїв, необхідних для розробки проектів з використанням Arduino, є доступною для більшості.

РОЗДІЛ ІІ. ОПАНУВАННЯ ОСНОВ ВИКОРИСТАННЯ ARDUINO У СЕРЕДОВИЩІ TINKERCAD CIRCUITS

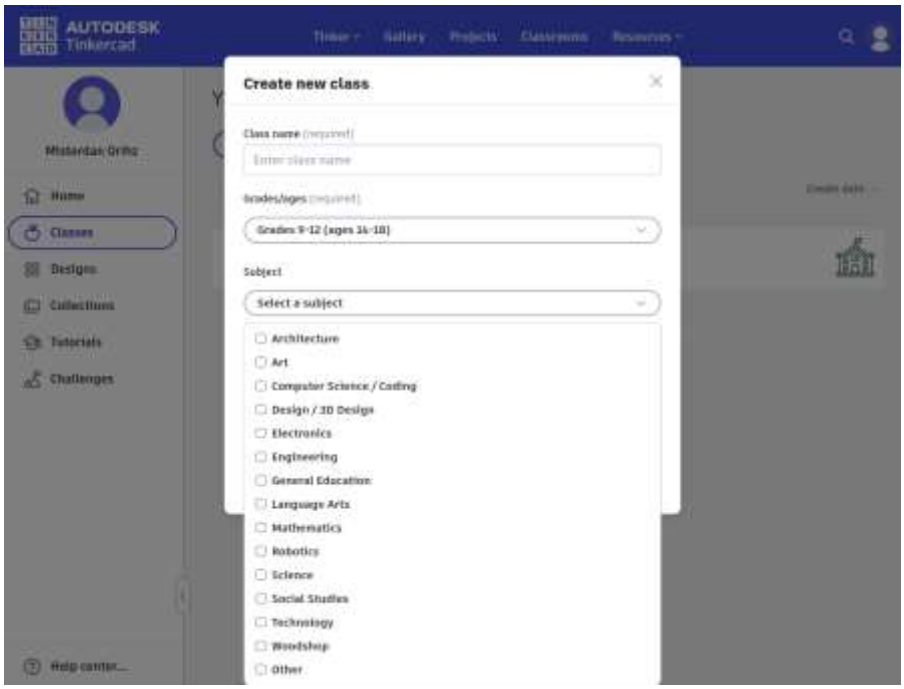
2.1 Огляд середовища TinkerCad Circuits

2.1.1. Реєстрація та налаштування облікового запису

Для початку роботи вчителю необхідно зареєструватися на платформі TinkerCad. Це можна зробити за допомогою електронної пошти або облікового запису Google. Після реєстрації обираємо меню “Classes” => “Create new class”.



У діалоговому вікні містяться меню “Class name”, “Grades/ages”, “Subject”, “Class Safe Mode setting”. Заповнюємо ці поля. Меню “Subject” видає опції “Architecture”, “Art”, “Computer Science / Coding”, “Design / 3D Design”, “Electronics”, “Engineering”, “General Education”, “Language Arts”, “Mathematics”, “Robotics”, “Science”, “Social Studies”, “Technology”, “Woodshop”, “Other”.

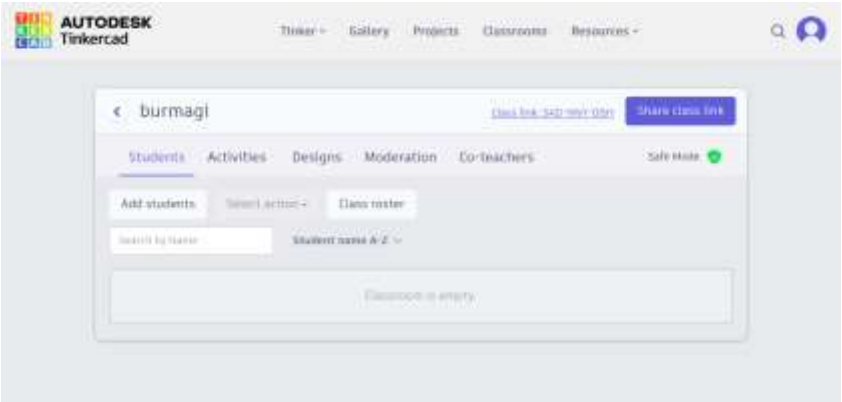


Якщо вчитель створює клас для роботи з Tinkercad Circuits і Arduino, найкращими варіантами будуть:

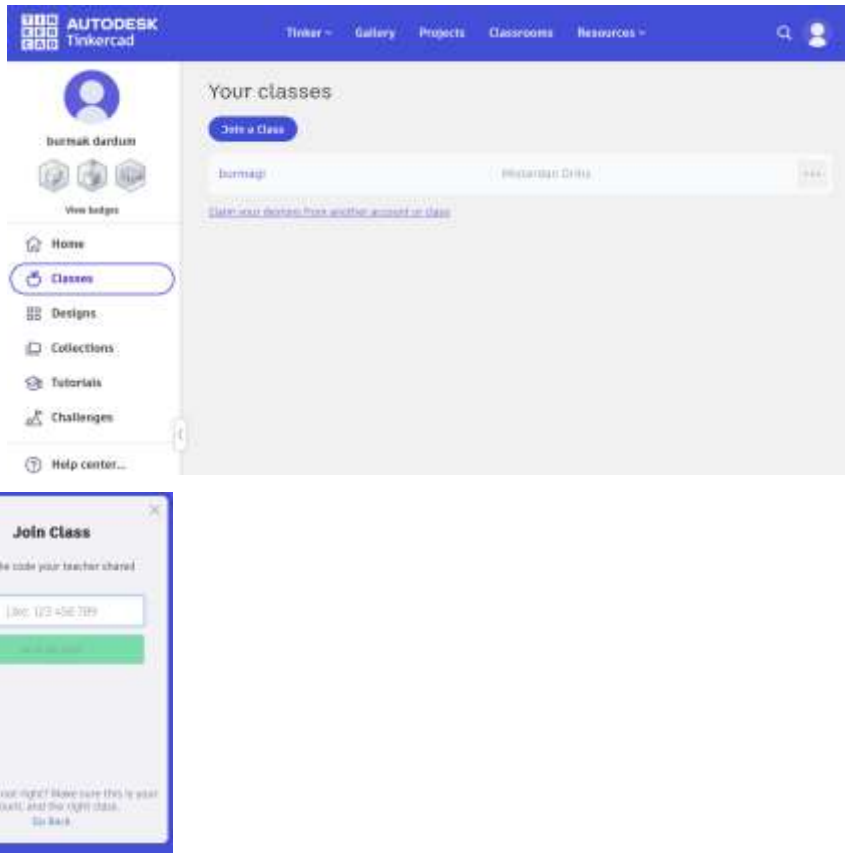
- Electronics – якщо фокус на електроніці та схемах
- Computer Science / Coding – якщо планується програмування Arduino
- Robotics – якщо використовується Arduino для робототехніки
- Technology або Engineering – якщо курс ширший і включає різні технічні аспекти

Якщо основна тема – віртуальне моделювання і програмування в Tinkercad та Arduino, тоді найкращим вибором буде Electronics або Computer Science / Coding.

Створений клас слід наповнити навчальним матеріалом і отримати Class link, тобто ідентифікаційний код класу, за яким учень приєднається до навчання.



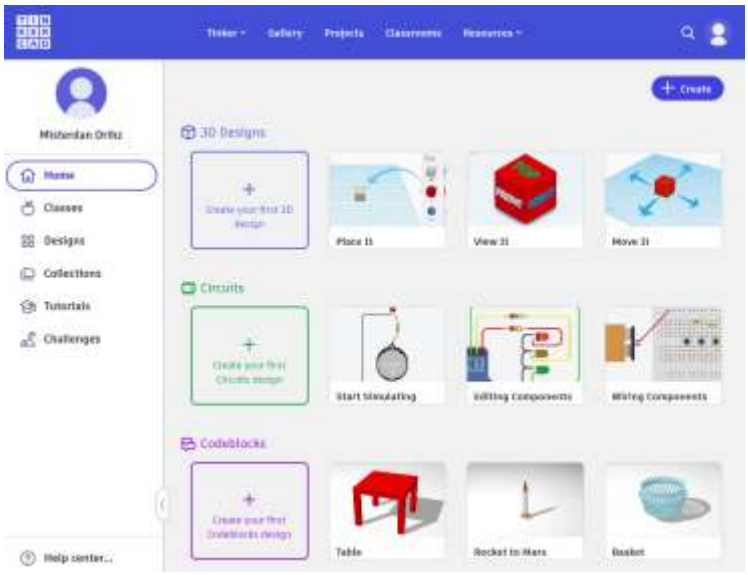
Учень також створює власний акаунт. Зайшовши у свій акаунт, учень обирає опцію меню “Classes”. У вікні “Your Classes” обираємо “Join a Class”.



В діалоговому вікні “Join Class” учень повинен занести код класу.

2. 1.2. Інтерфейс середовища, основні інструменти

Коли клас створено викладач створює курс (tutorial), який наповнює матеріалом, наявним в середовищі TinkerCad. Для цього викладач заходить в меню “Home” і в розділі “Circuits”:



Головний екран TinkerCad Circuits містить такі основні розділи:

Панель інструментів – кнопки для додавання компонентів, керування симуляцією.

Область проектування – місце, де створюється схема.

Список компонентів – набір доступних електронних елементів, таких як резистори, світлодіоди, кнопки, мікроконтролери.

2. 1.3. Додавання електронних компонентів

Щоб додати компонент до схеми, потрібно:

Відкрити список доступних елементів.

Обрати необхідний елемент та перетягнути його в область проектування.

Підключити компоненти між собою за допомогою провідників.

2.1.4. Використання блокового та текстового програмування

TinkerCad підтримує два режими програмування:

Блокове програмування – візуальне створення програм за допомогою готових блоків, що підходить для початківців.

Текстове програмування – написання коду мовою Arduino (C++), що дозволяє створювати складніші програми.

2.1.5. Запуск симуляції та перевірка роботи схеми

Після створення схеми та написання коду можна протестувати її роботу:

Натиснути кнопку "Start Simulation".

Перевірити правильність роботи схеми та коригувати помилки.

За потреби змінити код або з’єднання між компонентами.

Симуляція дозволяє зрозуміти, як працює схема без потреби у фізичному обладнанні, що значно полегшує навчальний процес.

2.1.6. Вправи.

Наочні матеріали:

Схема простого кола (LED + резистор).

Готовий файл проекту в TinkerCad.

Домашнє завдання:

Повторити створення простої схеми (LED + резистор). Вправи для учня, які допоможуть засвоїти основні функції Tinkercad Circuits:

1. Вправа для "Create your first Circuits design"

Завдання: Створіть новий електронний проєкт у Tinkercad Circuits, у якому лампочка (LED) світиться від батарейки.

Ознайомитися з блоковим кодуванням.

Натисніть "Create your first Circuits design".

Додайте 9V батарейку (Battery 9V) і світлодіод (LED).

З’єднайте позитивний (червоний) полюс батарейки з анодом (довга ніжка) LED.

З’єднайте катод (коротка ніжка) LED із негативним (чорним) полюсом батарейки.

Додайте резистор 330Ω між батарейкою та LED, щоб уникнути перевантаження.

Переконайтеся, що схема працює правильно.

2. Вправа для "Start Simulating"

Завдання: Перевірте, як працює схема із кнопкою (Button).

Відкрийте ваш проєкт або створіть новий.

Додайте батарейку, кнопку (Button) та світлодіод.

З’єднайте їх так, щоб LED засвітився лише при натисканні кнопки.

Натисніть "Start Simulating" і перевірте, чи працює кнопка.

Якщо LED не загоряється, перевірте правильність підключень.

3. Вправа для "Editing Components"

Завдання: Змініть опір резистора і спостерігайте, як це впливає на яскравість LED.

Відкрийте проєкт із батарейкою, LED і резистором.

Натисніть на резистор, щоб змінити його опір.

Змініть значення, наприклад, з 330Ω на 1kΩ (1000Ω).

Запустіть симуляцію і порівняйте яскравість світлодіода.

Спробуйте поставити занадто низький опір (наприклад, 10Ω) і подивіться, що станеться.

4. Вправа для "Wiring Components"

Завдання: Правильно з'єднайте компоненти на макетній платі (Breadboard).

Додайте батарейку, світлодіод, резистор і макетну плату.

З’єднайте батарейку з живильними шинами макетної плати (червоний дріт – до "+", чорний – до "-").

Встановіть LED у вертикальному ряду макетної плати.

Додайте резистор між катодом LED та "-" шиною.

Використовуйте правильне маркування дротів (червоний для "+", чорний для "-").

Перевірте, чи правильно ви з’єднали схему, і запустіть симуляцію.

2.2 Основи електроніки для роботи з Arduino

2. 2.1. Базові електричні поняття (струм, напруга, опір)

1. Тема: Визначення електричних величин: струм, напруга, опір

2. Мета:

Ознайомити учнів із основними електричними величинами (струм, напруга, опір).

Навчити застосовувати закон Ома в електричних розрахунках.

Пояснити вплив опору на роботу електронних схем.

3. Обладнання:

Комп'ютери з доступом до Tinkercad Circuits

Віртуальні компоненти: батарейка, резистори, світлодіоди, амперметр, вольтметр

Презентація або плакати з теоретичним матеріалом

Робочі зошити

4. Організаційний момент (5 хв)

Привітання, перевірка присутніх

Ознайомлення з темою та цілями уроку

5. Актуалізація знань (5 хв)

Дайте визначення таким термінам, як "струм", "напруга", "опір".

Де у повсякденному житті ми стикаємося з електрикою?

Що відбувається, коли заряджається телефон або вмикається лампочка?

(Коротке обговорення, формування початкових уявлень)

6. Теоретична частина (15 хв)

6.1 Визначення електричних величин

- **Струм (I)** – це потік електричних зарядів (електронів) у провіднику. Вимірюється в амперах (А).
- **Напруга (U)** – це різниця електричних потенціалів між двома точками кола. Вимірюється у вольтах (В).
- **Опір (R)** – це властивість матеріалу обмежувати потік електричного струму. Вимірюється в омах (Ω).

Порівняння з водопровідною системою:

- Струм – це як потік води в трубі.
- Напруга – це тиск води, що проштовхує її крізь трубу.

- Опір – це ширина труби: чим вона вужча, тим складніше воді текти.

6.2 Закон Ома та його застосування

Закон Ома:

Струм у провіднику прямо пропорційний напрузі та обернено пропорційний опору.

Формула:

U = I × R

де:

- U – напруга (В)
- I – струм (А)
- R – опір (Ω)

Закон Ома пояснює:

- Якщо **збільшити напругу** (U) при сталому опорі, **струм зростає** (I↑).
- Якщо **збільшити опір** (R) при сталій напрузі, **струм зменшується** (I↓).

Приклад для обговорення:

Як зміниться яскравість лампочки, якщо збільшити опір у колі?

6.3 Вплив опору на роботу електронних схем

- Якщо опір **занадто малий** – струм може стати надто великим → можливе **перегрівання** або **коротке замикання**.
- Якщо опір **занадто великий** – струм може бути занадто малим, і пристрій **не працюватиме належним чином**.

Роль резисторів у схемах з Arduino:

- Захищають компоненти (наприклад, світлодіоди) від надмірного струму.
- Дозволяють точно контролювати електричні величини в колі.

6.4 Основи мови С для Arduino

Програмування в Arduino базується на C/C++.

Основні оператори: *pinMode(), digitalWrite(), digitalRead()*.

Структура програми: функції setup() (налаштування) і loop() (основний цикл виконання).

6.5 Робота з віртуальними компонентами в Tinkercad Circuits

Використання бредборда для складання схем без паяння.

З’єднання компонентів за допомогою провідників.

Запуск і перевірка роботи схеми через симуляцію.

6.7 Теоретичні основи: призначення резисторів, світлодіодів і кнопок

Резистори обмежують струм у схемі (наприклад, перед світлодіодом, щоб уникнути перегорання).

Світлодіоди мають полярність: анод (+) і катод (-).

Кнопки працюють як перемикачі (замикають або розмикають електричний ланцюг).

6.8 Робота з цифровими сигналами

HIGH (1) – напруга 5V (увімкнено), *LOW (0)* – 0V (вимкнено).

Кнопки можуть працювати у режимі підтягувального резистора (INPUT_PULLUP).

Взаємодія світлодіодів і кнопок через логічні оператори (if-else).

7. Практична робота в Tinkercad Circuits (20 хв)

Завдання 1: "Дослідження струму та напруги"

Створити просту схему (батарейка 9 В + резистор + світлодіод).

Додати амперметр і вольтметр, виміряти значення.

Запитання: Що зміниться, якщо збільшити опір резистора?

Завдання 2: "Закон Ома в дії"

Використати Ohm’s Law Calculator у Tinkercad.

Розрахувати, який резистор потрібен для безпечного підключення світлодіода.

Завдання 3: "Вплив опору на яскравість світлодіода"

Змінювати значення резистора та спостерігати, як змінюється яскравість світлодіода.

Обговорення результатів: Чому при малому опорі світлодіод може перегоріти?

8. Аналіз та обговорення (10 хв)

Які величини ми сьогодні вивчили?

Як можна пояснити закон Ома простими словами?

Чому важливо контролювати рівень опору у схемах?

9. Підсумок уроку (5 хв)

Коротке повторення ключових понять.

Оцінювання роботи учнів.

10. Домашнє завдання: дослідити, які пристрої вдома містять резистори та який їхній вплив на роботу електроніки.

Завдання 1. Визначення електричних величин: струм, напруга, опір

"Електричні величини в повсякденному житті"

Інструкція:

Учням надається таблиця з побутовими електроприладами (наприклад, ліхтарик, фен, зарядний пристрій для телефону).

Завдання: знайти вказані в характеристиках пристрою напругу (V), силу струму (A) та опір (Ω) або розрахувати відсутню величину за формулою Ома.

Приклад таблиці:

Пристрій	Напруга (V)	Струм (A)	Опір (Ω)
Ліхтарик	3 В	?	10 Ом
Фен	220 В	5 А	?
Зарядний блок	5 В	2 А	?

➖ Запитання для обговорення:

Чому для різних пристроїв потрібна різна напруга?

Що відбувається, якщо змінити напругу, яку отримує пристрій?

Завдання2. Закон Ома та його застосування

"Обчислюємо параметри електричного кола"

➖ Інструкція:

Учням надається схема електричного кола в Tinkercad Circuits: батарейка 9 В, резистор, світлодіод.

Необхідно розрахувати силу струму (I), знаючи напругу (U = 9 В) і опір резистора (наприклад, R = 470 Ом), використовуючи закон Ома I = U / R.

➖ Покроковий процес:

Визначити відомі величини.

Розрахувати силу струму.

Запустити симуляцію у Tinkercad і перевірити, чи працює світлодіод.

Змінити значення резистора (наприклад, 220 Ом) і передбачити, як зміниться струм і яскравість LED.

➖ Запитання для обговорення:

Як зміниться струм, якщо збільшити опір?

Що станеться, якщо опору немає?

Завдання 3. Вплив опору на роботу електронних схем

"Чи згорить лампочка?"

➖ Інструкція:

Учням пропонується змоделювати електричне коло у Tinkercad Circuits з батарейкою 9 В і лампочкою.

Потрібно додати різні резистори (100 Ом, 500 Ом, 1 кОм) та передбачити, як зміниться яскравість лампочки.

➖ Що потрібно зробити:

Запустити симуляцію без резистора – що станеться з лампочкою?

Додати резистор 100 Ом – як зміниться яскравість?

Додати резистор 1 кОм – що змінилося?

Записати висновки.

➖ Запитання для обговорення:

Чому надто маленький опір може спричинити перегрів або вихід з ладу елемента?

Як можна регулювати яскравість лампочки, використовуючи змінний резистор?

2.2.2. Робота з резисторами, світлодіодами, кнопками

1. Тема: Основи електроніки для роботи з Arduino

2. Мета: Ознайомити учнів із принципами роботи резисторів, світлодіодів і кнопок та навчити їх створювати прості електронні схеми на платформі Arduino.

3. Віртуальні компоненти:

Arduino Uno – мікроконтролерна плата для керування схемою.

Резистори (220 Ом, 10 кОм) – для обмеження струму через світлодіоди та кнопки.

Світлодіоди (LED, різні кольори) – для індикації роботи схеми.

Кнопка (Pushbutton) – для керування світлодіодами.

З’єднувальні дроти (Jumper Wires) – для підключення компонентів.

Бредборд (Breadboard) – для зручного компонування схеми без паяння.

4. Організаційний момент (5 хв)

Привітання учнів, перевірка присутніх.

Ознайомлення з темою та метою уроку.

Мотивація: демонстрація роботи простої схеми (наприклад, миготливого світлодіода).

5. Актуалізація знань (5 хв)

Що таке резистори і світлодіоди?

Де знаходяться, як підключити віртуальні компоненти кнопки, резистора, світлодіода?

Структура програми.

6. Теоретична частина (15 хв)

6.1. Призначення та функціонал резисторів

Визначення резистора та його роль у схемах.

Опис одиниць вимірювання (Ом, кілоом, мегаом).

Демонстрація впливу опору на яскравість світлодіода.

6.2. Використання світлодіодів у схемах

Принцип роботи світлодіода.

Полярність світлодіода (анод і катод).

Правильне підключення світлодіода через резистор.

6.3. Підключення та використання кнопок

Різниця між нормально відкритими та нормально закритими кнопками.

Використання кнопки як елемента управління схемою.

Проблема "дребезгу контактів" і її вирішення в програмуванні.

7. Практична частина (20 хв)

Завдання 1: "Запалюємо світлодіод"

Підключити світлодіод через резистор до виходу Arduino.

Написати простий код для ввімкнення світлодіода:

```
int ledPin = 9;
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  digitalWrite(ledPin, HIGH);
}
```

Завдання 2: "Миготливий світлодіод"

Додати до коду затримку для миготіння світлодіода:

```
void loop() {
  digitalWrite(ledPin, HIGH);
  delay(500);
  digitalWrite(ledPin, LOW);
  delay(500);
}
```

Завдання 3: "Управління світлодіодом кнопкою"

Підключити кнопку до цифрового входу Arduino.

Написати код для ввімкнення світлодіода при натисканні кнопки:

```
int ledPin = 9;
int buttonPin = 2;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  if (digitalRead(buttonPin) == LOW) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Обговорення впливу резисторів на роботу кнопок.

8. Аналіз та обговорення (10 хв)

Основи мови С для Arduino

Яка структура типового скетчу (програми) для Arduino?(Очікувана відповідь: скетч складається з двох основних функцій – setup() для налаштувань і loop() для повторюваного коду.)

Для чого використовується функція *pinMode()*?

Як за допомогою *digitalWrite()* змінити стан виводу на високий (HIGH)?

Робота з віртуальними компонентами в Tinkercad Circuits

Що таке віртуальні компоненти в TinkerCad Circuits, і навіщо вони потрібні? (Очікувана відповідь: це програмне моделювання електронних компонентів для тестування схем без фізичного обладнання.)

Що таке бредборд і для чого він використовується?

Як у Tinkercad Circuits можна з'єднати компоненти між собою?

Яка функція дозволяє перевірити роботу схеми у віртуальному середовищі?

Теоретичні основи: призначення резисторів, світлодіодів і кнопок

Як працює кнопка в електронному колі і як її можна використати для керування світлодіодом? (Очікувана відповідь: кнопка може замикати або розмикати коло, що дозволяє вмикати або вимикати світлодіод, зчитуючи її стан за допомогою digitalRead().)

Яке призначення резистора у схемі з світлодіодом?

Як визначити полярність світлодіода?

Що відбувається у схемі, коли натискається кнопка?

Робота з цифровими сигналами

Які значення може приймати цифровий сигнал в Arduino?

Що означає параметр INPUT_PULLUP у функції *pinMode()* при налаштуванні кнопки?

Як за допомогою оператора if-else зробити так, щоб світлодіод загорявся при натисканні кнопки?

9. Підсумки уроку (5 хв)

Обговорення отриманих знань.

Які функції виконують резистори, світлодіоди та кнопки в електронних схемах?

Які труднощі виникли під час виконання завдань?

Відповіді на запитання учнів.

10. Домашнє завдання:

Самостійно зібрати схему з двома світлодіодами та кнопкою, де при натисканні кнопки один світлодіод гасне, а інший загоряється.

Завдання 1. Призначення та функціонал резисторів

"Знайди правильний резистор"

➖ Інструкція:

Уявіть, що ви хочете створити схему, де світлодіод працює без перегріву.

У Tinkercad Circuits створіть коло з 9 В батарейкою та світлодіодом.

Вам пропонуються резистори з різними номіналами: 220 Ом, 470 Ом, 1 кОм, 10 кОм.

Виберіть правильний резистор, щоб світлодіод світився, але не перегорів.

➖ Що потрібно зробити:

Перевірити схему без резистора. Що відбувається?

Додати резистор 220 Ом та перевірити яскравість.

Додати резистор 1 кОм та порівняти.

Зробити висновки: як опір впливає на струм і яскравість світлодіода?

➖ Запитання для обговорення:

Чому світлодіод може згоріти без резистора?

Як вибрати правильний резистор для захисту електронних компонентів?

Завдання 2. Використання світлодіодів у схемах

"Світлодіодний індикатор"

➖ Інструкція:

У Tinkercad Circuits створіть схему з трьома світлодіодами різних кольорів (червоний, жовтий, зелений).

Для кожного світлодіода використовуйте резистор 330 Ом.

Підключіть всі світлодіоди до батарейки 9 В через резистори.

Перевірте, чи працює схема у симуляції.

➖ Що потрібно зробити:

Використати окремі кнопки або перемикачі для вмикання кожного світлодіода.

Перевірити, як працює схема при паралельному та послідовному з'єднанні світлодіодів.

Записати висновки про яскравість світлодіодів у різних випадках.

➖ Запитання для обговорення:

Чому потрібно використовувати резистор для кожного світлодіода?

Чим відрізняється послідовне та паралельне з’єднання?

***Завдання 3*. Підключення та використання кнопок**

"Кнопка – вимикач світла"

➖ Інструкція:

У Tinkercad Circuits створіть електронну схему з кнопкою та світлодіодом.

Додайте батарейку 9 В, резистор 330 Ом та кнопку.

Підключіть кнопку так, щоб вона замикала коло при натисканні, вмикаючи світлодіод.

➖ Що потрібно зробити:

Переконатися, що кнопка працює правильно.

Спробувати змінити тип кнопки (тумблер, кнопка з фіксацією).

Додати ще одну кнопку і перевірити, як можна керувати світлодіодом.

➖ Запитання для обговорення:

Чим відрізняється кнопка з фіксацією від звичайної кнопки?

Як можна використати кнопку у більш складних схемах?

2.2.3. Аналогові та цифрові сигнали

Тема: Основи електроніки для роботи з Arduino

1. Мета уроку:

Ознайомити учнів з поняттям аналогових і цифрових сигналів та їх відмінностями.

Навчити визначати, які пристрої використовують аналогові або цифрові сигнали.

Пояснити, як Arduino працює з аналоговими та цифровими входами і виходами.

Закріпити навички використання функцій analogRead() і analogWrite() для роботи з датчиками та керування електронними компонентами.

Розвинути практичні навички побудови простих схем, що використовують аналогові та цифрові сигнали.

2. Віртуальні компоненти:

Arduino Uno – мікроконтролер, який дозволяє працювати з аналоговими та цифровими сигналами.

Світлодіоди (LEDs) – для демонстрації роботи цифрових виходів.

Резистори (220 Ом, 1 кОм, 10 кОм) – для обмеження струму в схемах з світлодіодами та датчиками.

Кнопка (Push Button) – як джерело цифрового сигналу (HIGH/LOW).

Фоторезистор (LDR, Light Dependent Resistor) – для демонстрації аналогового сигналу (зміна опору в залежності від освітлення).

Потенціометр – змінний резистор для регулювання аналогового сигналу.

П'єзоелемент (Buzzer) – для демонстрації роботи аналогового виходу через PWM (широтно-імпульсну модуляцію).

З'єднувальні провідники (Jumper Wires) – для підключення компонентів у схемі.

3. Організаційний момент (2 хв)

Привітання, перевірка присутності.

Оголошення теми та мети уроку.

4. Актуалізація знань (5 хв)

Що таке сигнал у фізиці?

Чим відрізняється двійкова система числення від десяткової?

Як працюють логічні рівні в електроніці?

Які датчики можуть передавати аналогові та цифрові сигнали?

5. Теоретична частина (10 хв)

Пояснення різниці між аналоговими та цифровими сигналами. Підкреслюється, що цифрові сигнали мають лише два стани — ввімкнено (HIGH) або вимкнено (LOW), тоді як аналогові можуть приймати безперервні значення в діапазоні, наприклад, від 0 до 5 Вольт.

Як Arduino зчитує аналогові сигнали за допомогою вбудованого аналогово-цифрового перетворювача (ADC), і як за допомогою широтно-імпульсної модуляції (PWM) мікроконтролер може «імітувати» аналоговий вихід.

Демонстрація прикладів використання аналогових датчиків (наприклад, фоторезисторів або потенціометрів) та цифрових входів (наприклад, кнопок).

6. Практична робота (15 хв)

Вправа 1: Підключити потенціометр до Arduino та зчитати значення через analogRead().

Вправа 2: Використати analogWrite() для регулювання яскравості світлодіода.

Вправа 3: Дослідити роботу фоторезистора, змінюючи рівень освітлення.

7. Аналіз, обговорення, закріплення матеріалу (10 хв)

Порівняння тереження щодо роботи цифрових та аналогових сигналів у власних схемах. Учитель ставить навідні запитання, наприклад: *Чому сигнал з потенціометра є аналоговим? Яку роль відіграє PWM при керуванні яскравістю світлодіода? У чому переваги цифрового сигналу в керуванні пристроями?* Учні формулюють висновки про особливості роботи кожного типу сигналу та способи їх реалізації в середовищі Arduino. Наприкінці проводиться коротке тестування або міні-вікторина для перевірки засвоєного матеріалу. Учитель заохочує учнів ставити власні запитання та ділитися прикладами, як ці сигнали можуть застосовуватися в реальних проектах.

8. Закріплення матеріалу (5 хв)

Обговорення результатів практичної роботи.

Дискусія: де використовуються аналогові та цифрові сигнали в реальному житті?

9. Підсумок уроку та домашнє завдання (3 хв)

Підсумок:

Аналогові та цифрові сигнали – основа електроніки.

Arduino підтримує як аналогові, так і цифрові входи/виходи.

Використання функцій analogRead() та analogWrite().

10. Домашнє завдання:

Придумати проєкт, у якому використовуються як цифрові, так і аналогові сигнали (наприклад, світлодіод, що змінює яскравість залежно від рівня освітлення).

Завдання 1: Різниця між аналоговими та цифровими сигналами - Розпізнаємо сигнали

Мета: зрозуміти різницю між аналоговими та цифровими сигналами.

Опис:

Учням надаються графіки сигналів (наприклад, квадратний сигнал для цифрового та плавна хвиля для аналогового).

Завдання: визначити тип сигналу та пояснити чому.

Додаткове практичне завдання (в Tinkercad Circuits):

Створити схему з кнопкою та світлодіодом.

Вмикати/вимикати світлодіод кнопкою та пояснити, чому це цифровий сигнал.

Порівняти з аналоговим потенціометром, підключеним до світлодіода, і пояснити, чому це аналоговий сигнал.

Завдання 2: Робота з аналоговими сигналами - Використання аналогових входів і виходів Arduino

Мета: навчитися використовувати аналогові входи та виходи Arduino.

Опис:

Використати потенціометр (змінний резистор), підключити його до аналогового входу A0 на Arduino.

Вивести значення потенціометра на серійний монітор у Arduino IDE.

Код для учнів:

```
int potPin = A0; // Підключаємо потенціометр до A0
int value;

void setup() {
  Serial.begin(9600); // Запускаємо серійний монітор
}

void loop() {
  value = analogRead(potPin); // Зчитуємо значення з потенціометра
  Serial.println(value); // Виводимо у монітор
  delay(500); // Чекаємо 500 мс
}
```

Запитання:

Як змінюються значення при повороті потенціометра?

Чому вони не лише 0 або 1023, а змінюються поступово?

Завдання 3: Читаємо температуру - Приклади зчитування даних із датчиків

Мета: навчитися працювати з аналоговими датчиками.

Опис:

Використати датчик температури LM35, підключити його до Arduino.

Вивести значення температури на серійний монітор.

Схема підключення:

VCC → 5V Arduino

GND → GND Arduino

VOUT → A0 (Аналоговий вхід)

Код для учнів:

```
int tempPin = A0;

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

```
void loop() {
  int value = analogRead(tempPin);
  float voltage = value * 5.0 / 1023.0;
  float temperature = voltage * 100; // Формула для LM35
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");
  delay(1000);
}
```

Запитання:

Що відбувається з показниками, якщо нагріти датчик?

Як можна використовувати цей принцип у реальних пристроях?

2.2.4. Практичні вправи: складання простих схем

Тема: Практичні вправи: складання простих схем

1. Мета уроку:

Закріпити знання учнів про роботу електронних компонентів (резистори, світлодіоди, кнопки, потенціометри).

Навчити створювати та аналізувати прості електронні схеми в середовищі Tinkercad Circuits.

Ознайомити з основами програмування Arduino для керування електронними компонентами.

Розвивати навички логічного мислення та роботи з електронними схемами.

2. Додаткове віртуальне обладнання:

- Arduino Uno
- Світлодіоди (LED)
- Резистори (220 Ом, 1 кОм)
- Кнопка (Push Button)
- Потенціометр
- З'єднувальні дроти

3. Організаційні моменти (5 хв)

Привітання, перевірка присутності.

Ознайомлення учнів із планом уроку та очікуваними результатами.

Підготовка робочого середовища: відкриття Tinkercad Circuits.

4. Актуалізація знань (5 хв)

Учням ставляться запитання для повторення основ електроніки та програмування Arduino:

Які електричні величини визначають роботу електронних схем?

Як підключається світлодіод? Яку функцію виконує резистор у схемі?

Яка роль кнопки у схемах з Arduino? Як зчитати її стан у коді?

Чим аналогові сигнали відрізняються від цифрових? Як можна зчитати аналогове значення на Arduino?

Обговорення відповідей та усунення можливих прогалин у знаннях.

5. Теоретична частина (10 хв)

Коротке пояснення:

Як працює електронна схема на Arduino.

Основні принципи з'єднання компонентів у Tinkercad Circuits.

Використання змінних у коді для керування світлодіодами та зчитування кнопок.

Основи використання analogRead() для потенціометра та digitalRead() для кнопки.

Приклад коду для виведення стану кнопки в монітор порту:

```
int buttonPin = 2; // Пін кнопки
void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  int buttonState = digitalRead(buttonPin);
  Serial.println(buttonState);
  delay(200);
}
```

6. Практична робота (20 хв)

Учні виконують три практичні завдання у Tinkercad Circuits:

1) Створення простої схеми з резистором і світлодіодом. Підключити світлодіод через резистор до Arduino. Написати код для його блимання.

Дослідити, як змінюється яскравість при різних значеннях резистора.

2) Використання кнопки для управління світлодіодом. Додати до схеми кнопку. Написати код, у якому світлодіод загоряється при натисканні кнопки.

3) Зчитування значень із потенціометра. Додати потенціометр. Зчитувати значення analogRead() та виводити їх у монітор порту.

7. Аналіз, обговорення, закріплення матеріалу (10 хв)

Обговорення труднощів, які виникли під час виконання вправ.

Обговорення можливих варіантів розширення схем (наприклад, додавання другого світлодіода або використання PWM).

Розбір найбільш поширених помилок у коді.

8. Закріплення матеріалу (5 хв)

Як експортувати електричну схему, створену в Tinkercad Circuits?

Навіщо потрібен потенціометр?

Що потрітно для роботи з монітором порту?

9. Підсумки заняття (5 хв)

 Учні відповідають на запитання:

Що нового вони дізналися сьогодні?

Як вони можуть застосувати ці знання у майбутніх проектах?

 Викладач дає рекомендації щодо самостійного опрацювання та подальшого вивчення Arduino.

10. Домашнього завдання (якщо передбачено).

1. Створення простої схеми з резистором і світлодіодом

Завдання 1: "Запалюємо світлодіод"

Мета: навчитися складати базову електричну схему з LED і резистором.

Необхідні компоненти:

Arduino

Світлодіод (LED)

Резистор (330 Ом)

Дроти

Схема підключення:

Катод (коротка ніжка) LED → GND

Анод (довга ніжка) LED → Один кінець резистора

Інший кінець резистора → Цифровий вихід D9 на Arduino

Код для учнів:

```
int ledPin = 9; // Підключаємо світлодіод до 9-го піну

void setup() {
  pinMode(ledPin, OUTPUT); // Налаштовуємо пін як вихід
}

void loop() {
  digitalWrite(ledPin, HIGH); // Вмикаємо світлодіод
  delay(1000); // Чекаємо 1 секунду
  digitalWrite(ledPin, LOW); // Вимикаємо світлодіод
  delay(1000); // Чекаємо 1 секунду
}
```

Запитання:

Чому ми використовуємо резистор?

Що станеться, якщо прибрати резистор?

2. Використання кнопки для управління світлодіодом

Завдання 2: "Кнопковий перемикач"

Мета: навчитися використовувати кнопку для керування світлодіодом.

Необхідні компоненти:

Arduino

Світлодіод

Резистор 330 Ом (для LED)

Кнопка

Резистор 10 кОм (pull-down)

Дроти

Схема підключення:

Катод LED → GND

Анод LED → Один кінець резистора (330 Ом) → D9

Один контакт кнопки → GND

Інший контакт кнопки → D2

D2 також підключаємо через 10 кОм резистор до GND

Код для учнів:

```
int ledPin = 9;
int buttonPin = 2;
int buttonState = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Запитання:

Що зміниться, якщо прибрати pull-down резистор?

Як можна змінити код, щоб світлодіод змінював свій стан кожного разу при натисканні кнопки?

3. Зчитування значень із потенціометра

Завдання 3: "Світлодіодний диммер"

Мета: навчитися зчитувати аналогові значення та використовувати їх для керування яскравістю світлодіода.

Необхідні компоненти:

Arduino

Світлодіод

Резистор 330 Ом

Потенціометр

Дроти

Схема підключення:

Катод LED → GND

Анод LED → Резистор 330 Ом → D9

Потенціометр:

Ліва ніжка → 5V

Середня ніжка → A0

Права ніжка → GND

Код для учнів:

```
int potPin = A0;  // Підключення потенціометра
int ledPin = 9;   // Підключення LED

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int potValue = analogRead(potPin);
  int brightness = map(potValue, 0, 1023, 0, 255);
  // Перетворення діапазону

  analogWrite(ledPin, brightness); // Регулюємо яскравість
  Serial.println(brightness); // Виводимо значення у монітор
  delay(100);
}
```

Запитання:

Чому використовується функція map()?

Що зміниться, якщо підключити потенціометр до цифрового входу?

2.3 Основи програмування Arduino в TinkerCad

2.3.1. Структура скетча (setup(), loop())

Огляд основних частин програми Arduino.

Призначення функцій setup() і loop().

Приклад простого скетча.

Тема: Структура скетча (setup(), loop())

1. Мета уроку:

Ознайомити учнів із основною структурою програми (скетча) для Arduino.

Розібрати принцип роботи функцій setup() та loop().

Навчити учнів писати базові програми для керування віртуальними компонентами в Tinkercad Circuits.

Закріпити навички роботи з цифровими входами та виходами в Arduino IDE.

2. Додаткове віртуальне обладнання:

Arduino Uno

Світлодіод (LED)

Резистор (220 Ом)

Кнопка (Push Button)

З'єднувальні дроти

3. Організаційні моменти (5 хв)

Привітання, перевірка присутності. Ознайомлення учнів із темою та очікуваними результатами. Перевірка готовності до роботи у Tinkercad Circuits.

4. Актуалізація знань (5 хв)

Учням ставляться запитання для повторення основ програмування Arduino:

5. Теоретична частина (10 хв)

Основні поняття:

1) Які типи даних використовуються у мовах програмування C/C++? 2) Що таке цифрові входи та виходи в Arduino? 3) Як підключити світлодіод до плати Arduino? 4) Як працює функція pinMode()?

Учні дають відповіді, а викладач коригує помилки та пояснює складні моменти.

Програма для Arduino складається з двох обов’язкових функцій:

void setup() – виконується один раз при запуску.

void loop() – виконується постійно у циклі.

Функція setup() використовується для:

Налаштування режимів роботи входів і виходів.

Ініціалізації змінних, бібліотек тощо.

Функція loop() містить основну логіку програми.

 Приклад простого скетча для блимання світлодіода:

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

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

☝ Пояснення:

setup() задає режим OUTPUT для піна 13.

У loop() виконується увімкнення світлодіода, пауза, вимкнення світлодіода, пауза.

6. Практична робота (20 хв)

Учні виконують три завдання у Tinkercad Circuits:

1) Завдання 1: Блимання світлодіода

Підключити світлодіод через резистор до піна 13.

Написати програму, яка змушує його блимати кожні 0,5 секунди.

2) Завдання 2: Керування світлодіодом кнопкою

Додати до схеми кнопку.

Написати код, у якому світлодіод вмикається при натисканні кнопки.

3) Завдання 3: Розширення функціоналу

Додати ще один світлодіод.

Написати код, у якому один світлодіод блимає постійно, а другий вмикається лише при натисканні кнопки.

7. Аналіз, обговорення, закріплення матеріалу (10 хв)

Учні діляться враженнями про виконання завдань.
Викладач аналізує типові помилки та пояснює, як їх виправити.
Обговорюється, як змінити скетч для інших сценаріїв (наприклад, використання датчиків).

8. Підсумки заняття (5 хв)

Підсумовуємо основні моменти уроку.
Учні відповідають на запитання:

Що нового вони дізналися сьогодні?

Чи зрозуміли вони структуру скетча Arduino?

Як можна розширити отримані знання для майбутніх проєктів?

Викладач дає рекомендації щодо подальшого вивчення теми.

9. Домашнє завдання

Завдання 1: Створити скетч, у якому три світлодіоди по черзі вмикаються на 1 секунду.

Завдання 2: Написати програму, де кнопка керує двома світлодіодами – один вмикається, коли кнопка натиснута, а інший вимикається.

Завдання 3: Дослідити функцію millis() і написати коротке пояснення, як її можна використовувати для блимання світлодіода без delay().

2.3.2. Основні команди для керування світлодіодами

Команда digitalWrite() для керування виходами.

Використання delay() для створення пауз.

Приклад коду для миготіння світлодіода.

Тема: Основні команди для керування світлодіодами

1. Мета уроку:

Ознайомити учнів із командами для керування світлодіодами в Arduino.

Вивчити основні команди: pinMode(), digitalWrite(), delay().

Навчити створювати програми для управління світлодіодами у середовищі Tinkercad Circuits.

Розвинути навички підключення компонентів до плати Arduino та роботи з кодом.

2. Додаткове віртуальне обладнання:

Arduino Uno

Світлодіоди (LED) – 3 шт.

Резистори (220 Ом) – 3 шт.

Кнопка (Push Button)

З’єднувальні дроти

3. Організаційні моменти (5 хв)

Привітання, перевірка присутності.

Ознайомлення учнів із темою, метою та завданнями уроку.

Перевірка готовності до роботи у Tinkercad Circuits.

4. Актуалізація знань (5 хв)

Учням пропонуються запитання для повторення попередніх тем:

1) Які типи сигналів (аналогові та цифрові) використовує Arduino?
2) Як правильно підключати світлодіоди до Arduino?
3) Що таке цифровий вихід у мікроконтролерах?
4) Чим відрізняються функції setup() і loop()?

Обговорення відповідей, пояснення викладача у разі потреби.

5. Теоретична частина (10 хв)

Основні поняття:

1) Команда pinMode()

Визначає, чи працюватиме пін як вхід (INPUT) або вихід (OUTPUT).

Приклад:

```
pinMode(13, OUTPUT); // Пін 13 працює як вихід
```

2) Команда digitalWrite()

Передає сигнал HIGH (1, ВКЛ) або LOW (0, ВИКЛ) на цифровий вихід.

Приклад:

```
digitalWrite(13, HIGH); // Увімкнути світлодіод
delay(1000); // Затримка 1 секунда
digitalWrite(13, LOW); // Вимкнути світлодіод
delay(1000); // Затримка 1 секунда
```

3) Команда delay()

Використовується для затримки виконання програми (в мілісекундах).

Приклад: delay(500); – затримка на 0,5 секунди.

Висновок: ці команди дозволяють вмикати, вимикати та керувати світлодіодами на платі Arduino.

6. Практична робота (20 хв)

Учні виконують три завдання у Tinkercad Circuits:

1) Завдання 1: Блимання одного світлодіода

Підключити світлодіод через резистор до піна 13.

Написати код для його блимання кожні 1 секунду.

2) Завдання 2: Керування двома світлодіодами

Підключити два світлодіоди до різних пінів.

Написати програму, у якій один світлодіод блимає, а інший світить постійно.

3) Завдання 3: Керування світлодіодом кнопкою

Додати до схеми кнопку.

Написати програму, у якій світлодіод вмикається тільки при натисканні кнопки.

7. Аналіз, обговорення, закріплення матеріалу (10 хв)

Учні обговорюють складнощі, з якими зіткнулися під час виконання завдань.

Викладач аналізує помилки та пояснює, як їх виправити.

Обговорення, як можна розширити програму, наприклад, використати більше світлодіодів або затримки різної тривалості.

8. Підсумки заняття (5 хв)

 Повторюємо основні моменти уроку.

 Обговорюємо, що дізналися нового.

 Учні відповідають на запитання:

Які основні команди використовуються для керування світлодіодами?

Як реалізувати керування кнопкою?

Чи можна змінити програму так, щоб світлодіоди вмикалися у певній послідовності?

9. Домашнє завдання

Завдання 1: Написати програму, у якій три світлодіоди вмикаються по черзі, кожен на 1 секунду.

Завдання 2: Реалізувати керування двома світлодіодами за допомогою двох кнопок (кожна кнопка керує своїм світлодіодом).

Завдання 3: Дослідити команду millis() та пояснити, як її можна використати для блимання світлодіодів без delay().

2.3.3. Використання змінних та умовних операторів

Оголошення та ініціалізація змінних.

Використання if-else для умовного виконання команд.

Приклади коду зі змінними.

Тема: Використання змінних та умовних операторів в Arduino

1. Мета уроку:

Ознайомити учнів з поняттям змінних у мові C/C++ та їх використанням в Arduino.

Навчити оголошувати та ініціалізувати змінні різних типів (int, float, bool).

Розглянути принцип роботи умовних операторів if-else.

Навчити використовувати змінні та умовні оператори у практичних проектах в Tinkercad Circuits.

2. Додаткове віртуальне обладнання:

Arduino Uno

Світлодіод (LED)

Резистор (220 Ом)

Кнопка (Push Button)

Фоторезистор

З’єднувальні дроти

3. Організаційні моменти (5 хв)

Привітання, перевірка присутності.

Ознайомлення учнів із метою та завданнями уроку.

Перевірка готовності до роботи в Tinkercad Circuits.

4. Актуалізація знань (5 хв)

Учням пропонуються запитання для повторення попередніх тем:

1) Які типи даних ви знаєте у мові C++?

2) Як використовується функція pinMode() у скетчах Arduino?

3) Чим відрізняється цифровий сигнал від аналогового?

4) Які команди використовуються для керування світлодіодами?

Обговорення відповідей, пояснення викладача у разі потреби.

5. Теоретична частина (15 хв)

Основні поняття:

1) Оголошення та ініціалізація змінних

Змінні дозволяють зберігати дані для подальшого використання в коді.

Основні типи змінних у C/C++ для Arduino:

```
int ledPin = 13;  // Ціле число (номер піна)
float voltage = 5.0;  // Дійсне число
bool isOn = false;  // Логічний тип (true/false)
```

2) Умовні оператори if-else

Використовуються для прийняття рішень у програмі.

Основний синтаксис:

```
if (умова) {
  // Код виконується, якщо умова істинна
```

```
} else {
  // Код виконується, якщо умова хибна
}
```

Приклад використання if-else для керування світлодіодом:

```
int buttonState = digitalRead(2);
if (buttonState == HIGH) {
  digitalWrite(13, HIGH);  // Увімкнути світлодіод
} else {
  digitalWrite(13, LOW);   // Вимкнути світлодіод
}
```

3) Приклади змінних в реальних програмах

Використання змінної для збереження стану світлодіода:

```
int ledState = 0;
if (digitalRead(2) == HIGH) {
  ledState = 1;
} else {
  ledState = 0;
}
digitalWrite(13, ledState);
```

6. Практична робота (20 хв)

Учні виконують три завдання в Tinkercad Circuits:

1) Завдання 1:

Керування світлодіодом за допомогою кнопки

Підключити кнопку та світлодіод до Arduino Uno.

Написати код, у якому світлодіод вмикається при натисканні кнопки та вимикається при відпусканні.

Використати змінну для збереження стану кнопки.

2) Завдання 2:

Автоматичне вмикання світлодіода при низькому освітленні

Підключити фоторезистор для зчитування рівня освітленості.

Якщо освітлення нижче 300, вмикати світлодіод.

Використати умовний оператор if-else.

3) Завдання 3:

Реалізація перемикача (Toggle Switch)

Написати код, у якому при одному натисканні кнопки світлодіод увімкнеться, а при наступному вимкнеться.

Використати змінну для збереження стану світлодіода.

Приклад роботи:

```
bool ledState = false;  // Початковий стан

void loop() {

  if (digitalRead(2) == HIGH) {

    ledState = !ledState;  // Інверсія стану

    digitalWrite(13, ledState);

    delay(300); // Антидребезг кнопки

  }

}
```

7. Аналіз, обговорення, закріплення матеріалу (10 хв)

Учні аналізують виконані завдання.

Обговорення проблем та складних моментів.

Викладач пояснює альтернативні варіанти розв’язання задач.

Обговорюємо можливі покращення коду (наприклад, використання millis() замість delay()).

8. Підсумки заняття (5 хв)

Повторюємо основні моменти уроку:

Що таке змінні і як вони використовуються?

Як працює if-else в Arduino?

Як зберігати стан кнопки та керувати світлодіодами?

Учні діляться своїми враженнями.

9. Домашнє завдання

Завдання 1:

Написати програму, у якій два світлодіоди вмикаються почергово кожні 2 секунди, використовуючи змінні.

Завдання 2:

Реалізувати логіку, при якій світлодіод блимає швидше, якщо кнопка натиснута.

Завдання 3:

Дослідити оператор switch-case та пояснити, коли його краще використовувати замість if-else.

2.3.4. Робота з циклами for, while

Призначення та використання циклів.

Приклади використання циклів у коді Arduino.

Завдання для самостійного опрацювання.

Тема: Використання циклів for та while в Arduino

1. Мета уроку:

Ознайомити учнів із призначенням та структурою циклів for та while в мові C++.

Пояснити, як цикли спрощують код при повторюваних діях.

Навчити використовувати цикли для керування компонентами в Arduino.

Закріпити знання через практичні завдання в Tinkercad Circuits.

2. Додаткове віртуальне обладнання:

Arduino Uno

Світлодіоди (3-5 шт.)

Резистори (220 Ом)

Кнопка (Push Button)

З’єднувальні дроти

3. Організаційні моменти (5 хв)

Привітання, перевірка присутності.

Ознайомлення учнів із метою та завданнями уроку.

Перевірка готовності до роботи в Tinkercad Circuits.

4. Актуалізація знань (5 хв)

Учням пропонуються запитання для повторення попередніх тем:

Що таке змінна та які її основні типи в C++?

Як працює оператор if-else?

Як керувати світлодіодом за допомогою digitalWrite()?

Що відбудеться, якщо в коді багаторазово повторюється одна і та ж команда?

Обговорення відповідей, пояснення викладача у разі потреби.

5. Теоретична частина (15 хв)

Основні поняття:

1. Призначення циклів

Цикли дозволяють автоматично повторювати певний набір команд без дублювання коду.

Використовуються для повторюваних процесів, наприклад, керування кількома світлодіодами або зчитування сенсорних даних.

2. Цикл for

Використовується, коли кількість повторень заздалегідь відома.

Структура:

```
for (ініціалізація; умова; крок) {
  // Код виконується у кожній ітерації
}
```

Приклад:

```
for (int i = 0; i < 5; i++) {
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

3. Цикл while

Використовується, коли кількість повторень невідома заздалегідь.

Структура:

```
while (умова) {
  // Код виконується, поки умова істинна
}
```

Приклад:

```
while (digitalRead(2) == LOW) {
  digitalWrite(13, HIGH);
```

```
}
```

4. Різниця між for і while

for використовується, коли відома кількість повторень.

while – коли цикл виконується, поки виконується певна умова.

6. Практична робота (20 хв)

Учні виконують три завдання в Tinkercad Circuits:

1. Завдання 1: Блимання світлодіода за допомогою циклу for

Запрограмувати світлодіод, щоб він 5 разів бликав із затримкою 500 мс.

Використати цикл for.

2. Завдання 2: Бігуче світло з 3-5 світлодіодів

Підключити 3-5 світлодіодів до Arduino.

Використати цикл for для послідовного ввімкнення та вимкнення кожного світлодіода.

Приклад коду:

```
int leds[] = {3, 4, 5, 6, 7};  // Масив з номерами пінів
void setup() {
  for (int i = 0; i < 5; i++) {
    pinMode(leds[i], OUTPUT);
  }
}
void loop() {
  for (int i = 0; i < 5; i++) {
    digitalWrite(leds[i], HIGH);
    delay(200);
    digitalWrite(leds[i], LOW);
  }
```

```
}
```

3. Завдання 3: Кнопка як перемикач (цикл while)

Коли натиснута кнопка, світлодіод блимає безперервно.

Використати цикл while.

Приклад коду:

```
while (digitalRead(2) == HIGH) {
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

7. Аналіз, обговорення, закріплення матеріалу (10 хв)

Учні аналізують виконані завдання.

Обговорення проблем та складних моментів.

Викладач пояснює альтернативні варіанти розв’язання задач.

Обговорюємо ефективність використання циклів у коді.

8. Підсумки заняття (5 хв)

Повторюємо основні моменти уроку:

Як працює цикл for?

Як працює цикл while?

У яких випадках зручніше використовувати for, а в яких while?

Учні діляться своїми враженнями.

9. Домашнє завдання

Завдання 1: Написати програму, у якій два світлодіоди блиматимуть почергово 10 разів (цикл for).

Завдання 2: Реалізувати бігучий вогник із 6 світлодіодів (цикл for).

Завдання 3: Написати код, у якому світлодіод блимає безперервно, поки натиснута кнопка (цикл while).

2.3.5. Практичні завдання

Створення програми для керування світлодіодами.

Використання кнопок у програмі.

Додаткові вправи на закріплення матеріалу.

Тема: Практичні завдання з використанням циклів у програмуванні Arduino.

1. Мета уроку: Закріпити знання про використання циклів for та while в програмуванні Arduino.

Навчитися застосовувати цикли для управління електронними компонентами.

Розвинути навички аналізу та налагодження програмного коду.

Підготувати учнів до самостійного застосування отриманих знань у проєктах.

2. Додаткове віртуальне обладнання:

Arduino Uno

Світлодіоди (5 шт.)

Резистори (220 Ом)

Кнопка (Push Button)

Потенціометр

З’єднувальні дроти

3. Організаційні моменти (5 хв):

Привітання, перевірка присутності.

Оголошення теми та мети уроку.

Перевірка готовності до роботи в Tinkercad Circuits.

4. Актуалізація знань (5 хв):

Учням пропонуються запитання для повторення:

Які типи циклів використовуються в Arduino?

Чим відрізняються цикли for та while?

Як використання циклів може оптимізувати програмний код?

Як керувати світлодіодами за допомогою циклу for?

5. Теоретична частина (10 хв):

Призначення та використання циклів:

Автоматизація повторюваних дій у коді.

Оптимізація коду для керування компонентами.

Приклади використання циклів у коді Arduino:

Цикл for: керування кількома світлодіодами.

Цикл while: очікування натискання кнопки.

6. Практична робота (20 хв):

Учні виконують наступні завдання у Tinkercad Circuits:

Завдання 1: Блимаючий світлодіод із використанням циклу for

Запрограмувати світлодіод, щоб він бликав 10 разів із затримкою 500 мс.

Завдання 2: Бігучий вогник

Підключити 5 світлодіодів до Arduino.

Використати цикл for для їх послідовного вмикання та вимикання.

Завдання 3: Кнопка як перемикач

Реалізувати схему, у якій світлодіод блиматиме лише при натисканій кнопці.

Використати цикл while для перевірки стану кнопки.

7. Аналіз, обговорення, закріплення матеріалу (10 хв):

Аналіз виконаних завдань, обговорення помилок.

Обговорення варіантів реалізації коду.

Викладач демонструє оптимізовані рішення.

8. Підсумки заняття (5 хв):

Повторення основних понять: цикл for, цикл while.

Дискусія щодо використання циклів у складніших проектах.

Оцінювання виконаних завдань.

9. Домашнє завдання:

Створити програму, у якій два світлодіоди блимають почергово 10 разів (цикл for).

Реалізувати бігучий вогник із 6 світлодіодів (цикл for).

Написати код, у якому світлодіод блимає безперервно, поки натиснута кнопка (цикл while).

РОЗДІЛ ІІІ. ПРАКТИЧНІ ПРОЄКТИ

3.1. Проєкт 1. "Мигаючий світлодіод"

Тема заняття: Практичний проєкт: "Мигаючий світлодіод" в рамках теми "Практичні проєкти".

1. Мета заняття

Ознайомити учнів із принципами роботи світлодіода в електронних схемах.

Навчити основам програмування Arduino для керування світлодіодами.

Закріпити навички роботи з базовими електронними компонентами та кодом.

Розвинути вміння налагодження та тестування простих схем у Tinkercad.

2. Необхідні матеріали та програмне забезпечення

Віртуальне середовище: Платформа Tinkercad (режим Circuits), Arduino Uno, Світлодіод (LED), Резистор 220 Ом, Дроти для з’єднання.

3. Організаційні моменти

Переконатися, що всі учні мають доступ до Tinkercad.

Перевірити, чи знайомі учні з основами електрики та програмування.

Визначити рівень підготовки учнів та розподілити їх на пари (за потреби).

4. Актуалізація знань (5 хв)

Що таке світлодіод? Яка його полярність?

Що таке резистор і навіщо він використовується в схемі?

Як працює плата Arduino?

Які основні команди для роботи з цифровими виходами?

5. Теоретична частина (10 хв)

Принцип роботи світлодіода: анод (+) і катод (-).

Роль резистора в схемі (обмеження струму).

Огляд основних команд для керування світлодіодом у середовищі Arduino IDE:

pinMode(pin, OUTPUT); – налаштування піна як виходу.

digitalWrite(pin, HIGH/LOW); – подача або відключення напруги.

delay(ms); – затримка виконання команди.

6. Практична робота (20 хв)

Завдання 1: Створення схеми в Tinkercad

Додати плату Arduino Uno.

Додати світлодіод та резистор (220 Ом).

Підключити анод світлодіода до цифрового виходу (наприклад, 13) через резистор.

Підключити катод до GND (землі).

Завдання 2: Написання коду

```
void setup() {
  pinMode(13, OUTPUT); // Налаштування піна 13 як виходу
}
void loop() {
  digitalWrite(13, HIGH); // Увімкнути світлодіод
  delay(500); // Затримка 500 мс
  digitalWrite(13, LOW); // Вимкнути світлодіод
  delay(500); // Затримка 500 мс
}
```

Завдання 3: Запуск та тестування

Завантажити код у віртуальну Arduino в Tinkercad.

Запустити симуляцію та перевірити, чи світлодіод блимає.

Внести зміни в код (наприклад, змінити швидкість миготіння).

7. Аналіз, обговорення, закріплення матеріалу (10 хв)

Чому використовується резистор?

Що відбувається, якщо змінити значення затримки?

Як можна розширити цей проєкт (наприклад, використати кілька світлодіодів)?

8. Підсумки заняття (5 хв)

Учні створили схему та програму для керування світлодіодом.

Обговорено можливі помилки та шляхи їх виправлення.

Учні зрозуміли принцип використання цифрових виходів Arduino.

9. Домашнє завдання

Модифікувати програму: змінити швидкість миготіння світлодіода.

Додати ще один світлодіод: реалізувати їх почергове миготіння.

Розширити проєкт: використати кнопку для керування світлодіодом.

3.2. Проєкт 2. "Кнопка для керування світлодіодом"

Тема: Практичні проєкти

1. Мета заняття:

Ознайомити учнів із принципом роботи кнопки у схемах Arduino.

Навчити підключати кнопку до мікроконтролера та використовувати її для керування світлодіодом.

Закріпити навички написання простих скетчів на C/C++ для керування електронними компонентами.

2. Додаткове віртуальне обладнання:

Arduino Uno

Кнопка (Button)

Світлодіод (LED)

Резистор 220 Ом (для світлодіода)

Резистор 10 кОм (для кнопки)

З’єднувальні дроти

Віртуальна макетна плата (Breadboard) у Tinkercad

3. Організаційні моменти

Переконатися, що всі учні мають доступ до платформи Tinkercad.

Перевірити, чи знайомі учні з основними командами мови C/C++ в Arduino.

Коротко повторити принцип роботи кнопки як вхідного пристрою.

4. Актуалізація знань

Що таке цифровий вхід і вихід в Arduino?

Як працює світлодіод у схемах з Arduino?

Що таке pull-down і pull-up резистори, для чого вони потрібні?

Як оголосити змінну та присвоїти їй значення в Arduino Sketch?

5. Теоретична частина

Принцип роботи кнопки в електронних схемах.

Використання кнопки для керування цифровими пристроями.

Поняття "відпружиненої" кнопки та її станів (натиснута/відпущена).

Важливість використання підтягуючих резисторів.

6. Практична робота

Створення схеми в Tinkercad

Розміщення Arduino Uno, кнопки, світлодіода та резисторів.

З'єднання компонентів за допомогою дротів.

Написання скетчу

```
int buttonPin = 2; // Пін для кнопки
int ledPin = 13;   // Пін для світлодіода
int buttonState = 0;

void setup() {
   pinMode(buttonPin, INPUT);
   pinMode(ledPin, OUTPUT);
}

void loop() {
   buttonState = digitalRead(buttonPin);
   if (buttonState == HIGH) {
      digitalWrite(ledPin, HIGH);
   } else {
      digitalWrite(ledPin, LOW);
   }
}
```

Запуск симуляції та тестування

Перевірка роботи кнопки та світлодіода.

Аналіз можливих помилок у коді та схемі.

7. Аналіз, обговорення, закріплення матеріалу

Чому необхідний резистор для кнопки?

Як змінити код, щоб світлодіод залишався ввімкненим після натискання кнопки?

Які ще пристрої можна керувати таким способом?

8. Підсумки заняття

Учні навчилися підключати кнопку та керувати світлодіодом.

Засвоєно основні команди для роботи з цифровими входами Arduino.

Отримано навички роботи у Tinkercad для віртуального моделювання схем.

9. Домашнє завдання

Модифікувати скетч так, щоб світлодіод змінював стан (ON/OFF) при кожному натисканні кнопки.

Дослідити функцію INPUT_PULLUP та застосувати її у своїй схемі.

Створити власний проєкт з використанням двох кнопок: одна вмикає світлодіод, інша вимикає.

3.3. Проєкт 3. Регулювання яскравості світлодіода (PWM)

Тема: Практичні проєкти

1. Мета проєкту:

Ознайомити учнів із широтно-імпульсною модуляцією (PWM).

Навчити використовувати аналогові виходи Arduino для регулювання яскравості світлодіода.

Закріпити навички роботи з потенціометром та функцією analogWrite().

2. Необхідні компоненти:

Плата Arduino (наприклад, Arduino Uno)

Світлодіод

Резистор 220 Ом

Потенціометр (10 кОм)

Провідники

Макетна плата

3. Теоретичні відомості:

Широтно-імпульсна модуляція (PWM) використовується для зміни середнього значення напруги, що подається на навантаження. В Arduino функція analogWrite(pin, value) дозволяє встановлювати значення від 0 до 255, де 0 – це мінімальна яскравість (вимкнено), а 255 – максимальна яскравість.

4. Покроковий план роботи:

Підключення компонентів:

Підключити катод світлодіода до GND через резистор 220 Ом.

Анод підключити до PWM-виходу (наприклад, D9).

Підключити потенціометр:

Ліва ніжка – до GND.

Права ніжка – до 5V.

Центральна ніжка – до аналогового входу A0.

Написання коду:

```
int ledPin = 9;  // Пін для світлодіода
int potPin = A0; // Пін для потенціометра

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int sensorValue = analogRead(potPin);  // Зчитуємо значення з потенціометра (0-1023)
  int brightness = map(sensorValue, 0, 1023, 0, 255);  // Масштабуємо до 0-255
  analogWrite(ledPin, brightness);  // Регулюємо яскравість світлодіода
}
```

Запуск та тестування:

Завантажити скетч на Arduino.

Обертати потенціометр і спостерігати, як змінюється яскравість світлодіода.

5. Аналіз і обговорення:

Як впливає зміна значень PWM на яскравість світлодіода?

Чому ми використовуємо функцію map()?

Як ще можна використовувати PWM у проектах?

6. Додаткові завдання:

Замініть потенціометр на фоторезистор і реалізуйте керування яскравістю світлодіода залежно від освітлення.

Реалізуйте плавне автоматичне збільшення і зменшення яскравості світлодіода.

Додайте кнопку для перемикання між режимами постійного світіння та регулювання яскравості.

РЕКОМЕНДАЦІЇ

Використання Arduino в навчальному процесі відкриває широкі можливості для розвитку в учнів технічного мислення, творчих здібностей та практичних навичок у програмуванні й електроніці. Даний підрозділ містить рекомендації щодо організації навчального процесу, підходів до викладання та ефективного використання платформи Arduino в освітньому середовищі.

1. Планування навчального процесу

Перед початком роботи з Arduino необхідно:

Визначити навчальні цілі та компетенції, які мають бути сформовані в учнів.

Розробити навчальну програму, що включає теоретичні та практичні заняття.

Підготувати необхідні матеріали: методичні посібники, презентації, відеоуроки та інструкції.

Визначити рівень підготовки учнів і, за потреби, провести вступне заняття з основ електроніки та програмування.

2. Організація навчальних занять

Для ефективного навчання слід дотримуватися наступних принципів:

Поступове ускладнення завдань. Розпочати з простих проєктів, таких як «Мигаючий світлодіод», і поступово переходити до складніших схем та алгоритмів.

Поєднання теорії та практики. Кожне заняття має містити коротке пояснення теоретичних основ та обов’язкову практичну частину.

Інтерактивне навчання. Використовувати групові проєкти, змагання та творчі завдання для підвищення мотивації учнів.

Застосування реальних проєктів. Створення корисних пристроїв на основі Arduino допомагає учням побачити практичне застосування отриманих знань.

3. Використання обладнання та програмного забезпечення

Підготувати необхідне обладнання: плати Arduino, макетні плати, світлодіоди, кнопки, резистори, потенціометри, датчики тощо.

Використовувати середовище програмування Arduino IDE або онлайн-симулятор Tinkercad, якщо фізичне обладнання недоступне.

Навчити учнів основам мови C/C++ для написання скетчів та роботи з бібліотеками.

4. Формування навичок самостійної роботи

Стимулювати учнів до самостійного пошуку інформації та аналізу помилок у коді.

Заохочувати використання документації та відкритих ресурсів, таких як офіційний сайт Arduino, форуми та відеоуроки.

Організовувати міні-проєкти, у яких учні самі визначають технічні завдання та реалізують їх.

5. Оцінювання та зворотний зв’язок

Використовувати поєднання формувального (самооцінка, взаємооцінка, обговорення) та підсумкового оцінювання (проєктні роботи, тестові завдання).

Оцінювати не лише правильність виконання проєкту, а й креативність підходу, логіку коду, командну роботу та здатність до самостійного вирішення проблем.

Надавати індивідуальний зворотний зв’язок кожному учню з рекомендаціями для подальшого розвитку.Робота з Arduino і Tinkercad Circuits у системі підвищення кваліфікації сприяє реалізації сучасних освітніх стратегій, забезпечує розвиток цифрової компетентності вчителів інформатики, сприяє інноваційному переосмисленню змісту шкільної освіти та надає нові можливості для формування професійних навичок у цифровому середовищі.

ВИСНОВКИ

Активна участь вчителів у процесі створення та розробки програмно-апаратного комплексу на базі Arduino дозволяє збільшувати свої технічні знання та розвивати практичні навички.

Застосування Arduino в освіті може мати широкі перспективи для розвитку технологічного сектору та підготовки кваліфікованих фахівців. Однак, необхідно враховувати важливість розвитку методичних підходів до використання цієї технології в навчальному процесі, а також створення необхідних умов для здійснення практичних занять з використанням Arduino. Розвиток неформальної освіти вчителів з використанням Arduino є важливим напрямом підвищення їх кваліфікації та підготовки до викладання сучасних технологій у навчальному процесі. Для успішного впровадження цієї технології в освіту необхідно продовжувати дослідження в галузі педагогіки та розвитку технологій, розробляти методичні матеріали та забезпечувати належний рівень технічного оснащення навчальних закладів.

Більш глибоке вивчення можливостей використання платформи Arduino в неформальній освіті поліпшить її вплив на розвиток творчих та інженерних навичок вчителів інформатики та фізики. Також важливо провести порівняльний аналіз різних підходів до використання Arduino в освітньому процесі та вивчити можливості поєднання цієї платформи з іншими технологіями для створення інтегрованих та мультимедійних навчальних курсів. Важливим є дослідження питань безпеки при використанні Arduino в навчальному процесі та розроблення відповідних методик та правил для уникнення можливих негативних наслідків. До інших невирішених питань належать також розроблення інтегрованих методичних підходів до використання Arduino, визначення найбільш ефективних стратегій навчання з використанням цієї платформи, а також вдосконалення засобів оцінювання. Результати подальших досліджень можуть бути корисні для вчителів, які бажають використовувати Arduino в своїх уроках, а також для науковців, які працюють у галузі інноваційних технологій в освіті.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

Алєксєєва Г. М., Бабич П. М. Використання платформи Arduino для професійної підготовки майбутніх інженерів-педагогів. Фізико-математична освіта. 2018. Вип. 4(18). С. 12–16.

2. Гриневич Л. М., Морзе Н. В., Бойко М. А. Scientific education as the basis for innovative competence formation in the conditions of digital transformation of the society. Інформаційні технології і засоби навчання. 2020. Вип. 77, № 3. С. 1–26. URL: https://journal.iitta.gov.ua/index.php/itlt/article/view/3704 (дата звернення: 14.11.2024).

3. Павлюс В. П. Використання платформи Arduino для організації курсу «Основи робототехніки» в навчальних закладах. URL: http://conf.fizmat.tnpu.edu.ua/article/9/ (дата звернення: 14.11.2024).

4. Морзе Н., Струтинська О., Умрик М. Освітня робототехніка як перспективний напрям розвитку STEM-освіти. Відкрите освітнє е-середовище сучасного університету. 2018. Вип. 5. С. 178–187. URL: https://doi.org/10.28925/2414-0325 (дата звернення: 14.11.2024).

5. Цаплан І. Перспективи підготовки майбутніх фахівців комп'ютерної та інженерної техніки на базі принципів STEM-освіти, та розробки навчально-методичного комплексу «Основи програмування, на базі датчиків Arduino». Магістерський науковий вісник. 2019. Вип. 33. С. 291–306.

6. Шерман М. І., Самчинська Я. Б., Кужелюк Н. І. Проєктування веб-ресурсу з вивчення платформи Arduino для інженерів-програмістів з рівнем вищої освіти «Магістр». Наукові нотатки. 2019. Вип. 67. С. 168–175.

7. Кривонос О. М., Кузьменко Є. В., Кузьменко С. В. Огляд та перспективи використання платформи Arduino Nano 3.0 у середній школі. Інформаційні технології і засоби навчання. 2016. Том 56, № 6. С. 77–87. URL:

https://journal.iitta.gov.ua/index.php/itlt/article/view/1506 (дата звернення: 14.11.2024).

МЕТОДИЧНО-ПРАКТИЧНЕ ВИДАННЯ

ARDUINO ДЛЯ ВЧИТЕЛЯ ІНФОРМАТИКИ: ВІРТУАЛЬНЕ МОДЕЛЮВАННЯ ТА ПРОГРАМУВАННЯ В TINKERCAD

Практичний посібник

Автор: Крамар Сергій Сергійович

Інститут цифровізації освіти
Національної академії педагогічних наук України
м. Київ, вул. Максима Берлінського, 9
Свідоцтво про державну реєстрацію:
серія ДК №7609 від 23.02.22 р.
електронна пошта (E-mail): iitzn_apn@ukr.net