

**Дрогобицький державний педагогічний університет
імені Івана Франка
кафедра фізики та інформаційних систем**

Роман Лешко

**ІНФОРМАЦІЙНО-КЕРУЮЧІ СИСТЕМИ
ТА STEM-ТЕХНОЛОГІЇ**

*Методичні рекомендації до виконання
лабораторних робіт*

**Дрогобич
2023**

УДК 004.89(076.5)

Л 53

Рекомендовано до друку вченою радою Дрогобицького державного педагогічного університету імені Івана Франка (протокол № 13 від 15.11.2023 р.)

Рецензенти:

Столярчук Ігор, доктор фізико-математичних наук, професор кафедри фізики та інформаційних систем Дрогобицького державного педагогічного університету імені Івана Франка;

Попович Володимир, кандидат фізико-математичних наук, доцент кафедри технологічної та професійної освіти Дрогобицького державного педагогічного університету імені Івана Франка.

Відповідальний за випуск:

Британ Віктор, кандидат фізико-математичних наук, доцент кафедри фізики та інформаційних систем Дрогобицького державного педагогічного університету імені Івана Франка.

Лешко Р.

Л 53 Інформаційно-керуючі системи та STEM-технології : методичні рекомендації до виконання лабораторних робіт. Дрогобич : ДДПУ ім. І. Франка, 2023. 40 с.

Навчальний посібник “**Інформаційно-керуючі системи та STEM-технології**” написаний відповідно до робочих програм навчальної дисципліни “**Інформаційно-керуючі системи та STEM-технології**” для підготовки фахівців магістерського рівня вищої освіти спеціальностей 014 Середня освіта (Фізика) та 104 Фізика та астрономія, 112 Комп’ютерні науки, затверджених вченою радою Дрогобицького державного педагогічного університету імені Івана Франка.

У посібнику подано лабораторні роботи, які слугують базою для освоєння інформаційно-керуючих систем та STEM-технологій, зібрані основні методи та принципи роботи з платами Arduino. Для посібника характерне викладання теоретичних відомостей з практичною інструкцією і завданням до лабораторної роботи.

Бібліографія 12 назв.

З М І С Т

ПЕРЕДМОВА	4
Лабораторна робота № 1 Плата Arduino та її компоненти.....	6
Лабораторна робота № 2 Онлайн платформа Tinkercad	12
Лабораторна робота № 3 Цифрові виходи Arduino.	
Робота зі світлодіодом.....	16
Лабораторна робота № 4 Підключення кнопок та вимикачів.....	21
Лабораторна робота № 5 Семисегментний діодний індикатор.....	24
Лабораторна робота № 6 Рідкокристалічний дисплей.....	27
Лабораторна робота № 7 Сенсори. Резистивні сенсори	30
Лабораторна робота № 8 Аналоговий датчик температури	32
Лабораторна робота № 9 Датчик світла	35
Список використаних джерел	38

ПЕРЕДМОВА

Лабораторні роботи з дисципліни “Інформаційно-керуючі системи та STEM-технології” є важливою частиною навчального процесу, яка дає змогу студентам отримати практичні навички в розробці інформаційних систем та ознайомитися з сучасними технологіями в області STEM. Однією з ключових технологій, яка вивчатиметься під час виконання лабораторних робіт, є платформа Arduino.

Arduino – це відома в усьому світі відкрита платформа для створення вбудованих систем, що використовує мікроконтролери та забезпечує можливість створювати різноманітні інтерактивні пристрої та системи, які можуть бути використані в різних галузях науки і технологій. Arduino відкриває перед студентами безмежні можливості для реалізації інноваційних ідей та досліджень у сфері інформаційних технологій і автоматизації.

Arduino – це компактна мікроконтролерна платформа, яка стала дуже популярною у світі освіти, особливо в STEM-навчанні (наука, технології, інженерія та математика). Вона має кілька ключових переваг для вивчення інформаційно-керуючих систем і STEM технологій.

1. Легкість використання та програмування. Arduino має простий та зрозумілий інтерфейс, який дає змогу навчатися основ програмування швидко та ефективно. Використання Arduino допомагає студентам вивчати основи програмування, алгоритмів та логіки, не вдаючись до складних низькорівневих аспектів.

2. Низька вартість та доступність. Arduino відома відносно низькою вартістю та доступністю. Це робить її ідеальним засобом для навчання та використання в освітніх закладах з обмеженими бюджетами.

3. Велика спільнота та ресурси. Arduino має велику та активну спільноту, що підтримує навчання та обмін знаннями. Існує безліч

онлайн-ресурсів, від навчальних посібників до форумів, які допомагають долати труднощі та розширювати навички.

4. Широкі можливості додаткового обладнання. Arduino сумісна з великою кількістю сенсорів, модулів та іншого обладнання. Це дає змогу студентам вивчати різні аспекти STEM-технологій та інформаційно-керуючих систем, включаючи сенсори, робототехніку, автоматизацію тощо.

5. Практичне застосування. Arduino надає можливість створювати функціональні пристрої та проекти. Це сприяє кращому засвоєнню теорії та мотивує для подальших досліджень.

У цьому методичному посібнику ми надаємо інструкції та рекомендації щодо виконання лабораторних робіт з використанням платформи Arduino. Ви дізнаєтеся, як підключати компоненти, програмувати мікроконтролери, створювати програми для зчитування даних з датчиків, керувати даними та розробляти інтерактивні проекти. Ми сподіваємося, що ці лабораторні роботи допоможуть вам отримати практичні навички, які будуть корисні у вашій подальшій кар'єрі та дослідженнях.

Нехай цей посібник стане вашим провідником у світі Arduino та інформаційно-керуючих систем і STEM-технологій, де теоретичні знання поєднуються з практичними навичками, і допоможе реалізувати ваші ідеї та проекти з електроніки та програмування.

Лабораторна робота № 1

Плата Arduino та її компоненти

Мета

Вивчити компоненти плати Arduino та отримати навички її під'єднання до персонального комп'ютера. Знати й уміти користуватися середовищем Arduino IDE.

План

1. Вивчення плати Arduino і її типів.
2. Встановлення Arduino IDE і налаштування портів.
3. Під'єднання плати Arduino до персонального комп'ютера.

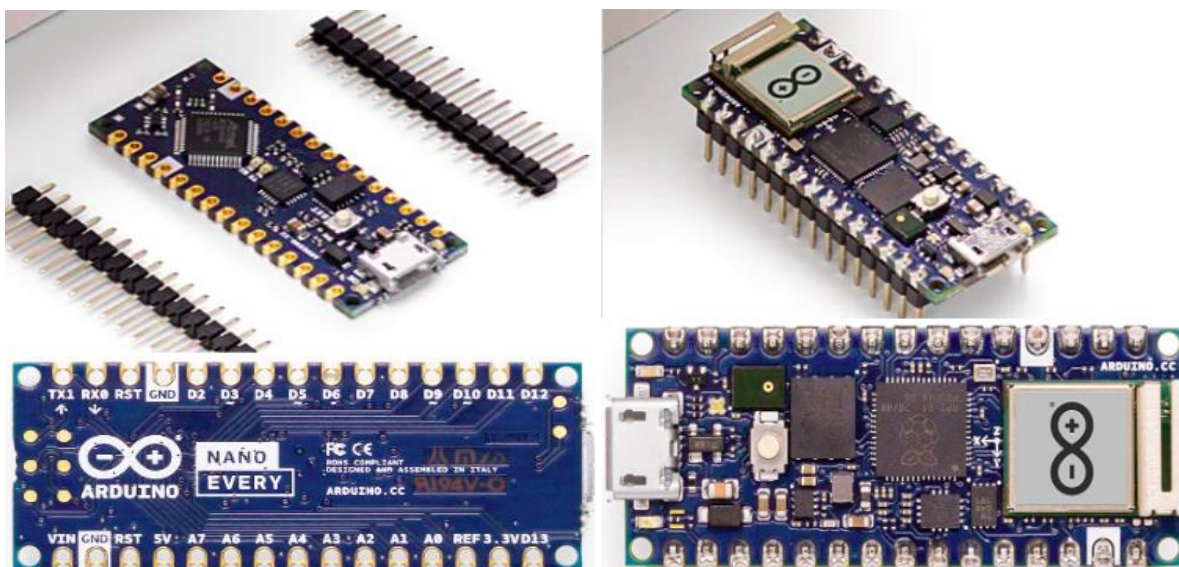
Теоретичні відомості

Arduino – це платформа з відкритим кодом, створена для швидкої і легкої розробки різноманітних електронних пристроїв. Arduino може отримувати дані про навколишній світ завдяки датчикам та реагувати, керуючи світлом, моторчиками та іншими приводами. Мікроконтролер на платі програмується за допомогою мови програмування Arduino та середовища розробки Arduino. Для програмування не потрібно програматора, програма зашивається через порт USB. Для початку роботи знадобиться лише сама плата Arduino та комп'ютер із встановленим середовищем розробки Arduino IDE. Проєкт народився у 2003 і отримав свою назву завдяки таверні біля Інституту IVREA в Італії (Bar di Re Arduino або Bar del Rey Arduino). Проєкт розв'язав проблему, як навчити швидко студентів програмувати електронні пристрої на мікроконтролерах без паяння і складання складних схем.

На сьогодні відомі різні “сімейства” плат Arduino. Перелічимо декілька з них.

1. Сімейство **Nano** – це набір мініатюрних плат із безліччю функцій. Він варіюється від недорогого базового Nano Every до багатофункціонального Nano 33 BLE Sense / Nano RP2040 Connect, який має радіомодулі Bluetooth® / Wi-Fi. Ці плати також мають

набір вбудованих датчиків, таких як температура / вологість, тиск, жести, мікрофон тощо. Їх також можна програмувати за допомогою MicroPython, вони підтримують машинне навчання.



2. У класичному сімействі є такі плати, як легендарна Arduino UNO та інші класичні плати, наприклад Leonardo & Micro.



Ці плати вважаються основою проекту Arduino і користуються успіхом протягом багатьох років (і далі в майбутньому). Надалі мова буде йти про них.

Мікроконтролер Arduino на платі програмується за допомогою мови програмування Arduino, що базується на мовах (C та C++), та власного середовища розробки, яке доступне і безкоштовне для завантаження. Проекти пристроїв на основі Arduino можуть працювати самостійно та взаємодіяти з комп'ютером.

Для взаємодії Arduino з комп'ютером необхідний кабель USB і програма Arduino IDE (IDE – Integrated Development Environment / Інтегроване середовище розробки). Відповідну програму можна завантажити тут <https://www.arduino.cc/en/software> для різних платформ (Windows, Linux, Mac OS). Після встановлення і першого запуску вікно програми має вигляд:



```
2 | Arduino 1.8.19
File Edit Sketch Tools Help
[Icons]
2
// C++ code
//
int pin7 = 7;
int pin8 = 8;
byte val = 0;
void setup()
{
  pinMode(pin7, INPUT);
  pinMode(pin8, OUTPUT);
}

void loop()
{
  val = digitalRead(pin7);

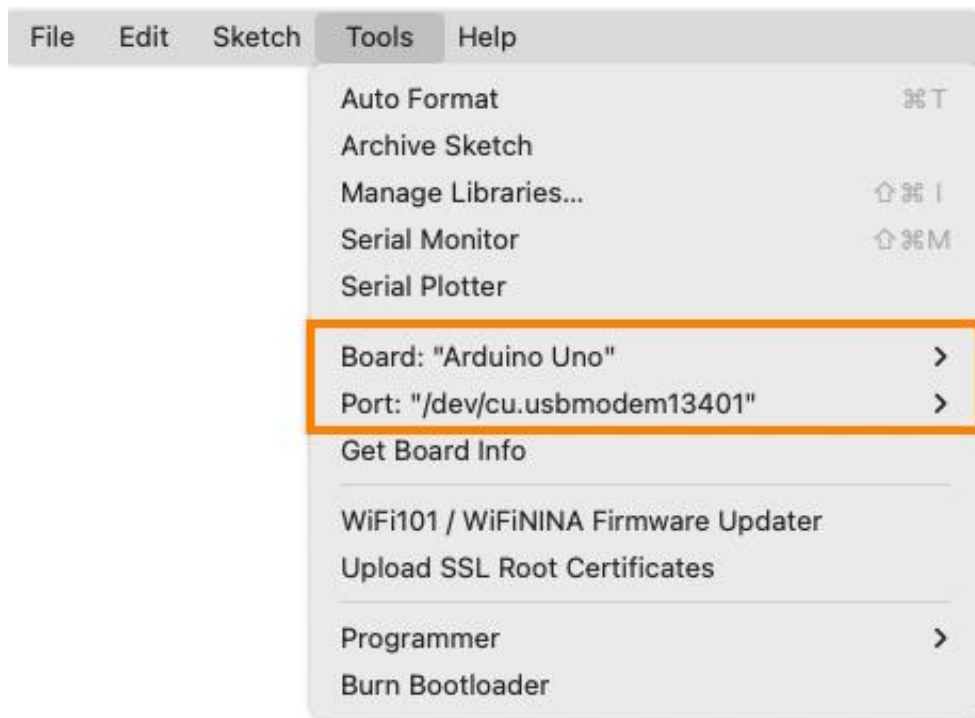
  if(val == 1){

```

1 Arduino Uno on /dev/ttyACM0

меню, кнопки керування програмою, редактор програми і консоль, де виводиться деяка інформація.

Після під'єднання Arduino до USB порту, необхідно у програмі вказати потрібний порт і вибрати тип плати. Наприклад, як на картинці.



Треба відзначити, що у кожного окремого комп'ютера назва порту може бути іншою (не обов'язково так, як на картинці вище).

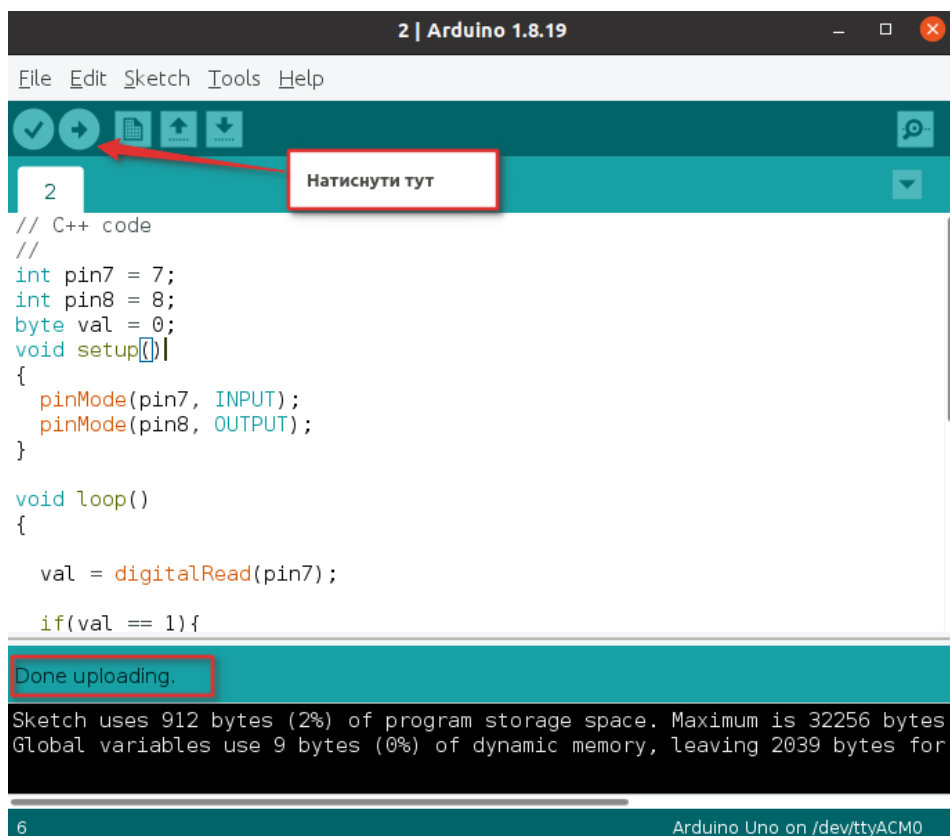
Після налаштувань можна перевірити, чи взаємодія налагоджена. Для цього необхідно написати просту порожню програму (яка нічого не робить) і завантажити у плату. Програми для Arduino називають скетчами.

```
// C++ code
void setup()
{
}
void loop()
{
}
```

Наведена структура програми є такою: у блоці **setup()** може писатися код, який виконується лише раз при вмиканні живлення

на плату; код, що додано у блок **loop()**, виконується постійно, доки є живлення.

Якщо все правильно зроблено, то після завантаження програми виведено буде повідомлення



Якщо ж щось неправильно налаштовано, то у консолі буде вказуватися помилка і посилання на інтернет-ресурс з її описом та варіантами вирішення.

Завдання

1. Завантажте та встановіть програму Arduino IDE.
2. Налаштуйте тип плати і порт.
3. Вивчіть меню та завантажте порожню програму у плату.

Контрольні питання

1. Що таке Arduino?
2. Яку інформацію можна отримати для Arduino?
3. Які типи і сімейства плат Arduino існують?

4. Які типові плати належать до сімейства Arduino Nano?
5. Які типові плати належать до класичного сімейства Arduino?
6. Що потрібно, щоб Arduino під'єднати до комп'ютера?
7. Що таке Arduino IDE і де його завантажити?
8. Що потрібно, щоб налаштувати Arduino IDE?
9. Для яких платформ можна встановити Arduino IDE?
10. Які мови програмування використовуються для написання програм для Arduino?
11. Що таке скетч?
12. Назвіть особливості блоків програми setup і loop.
13. Чи може Arduino працювати самостійно без комп'ютера?

Лабораторна робота № 2

Онлайн платформа Tinkercad

Мета

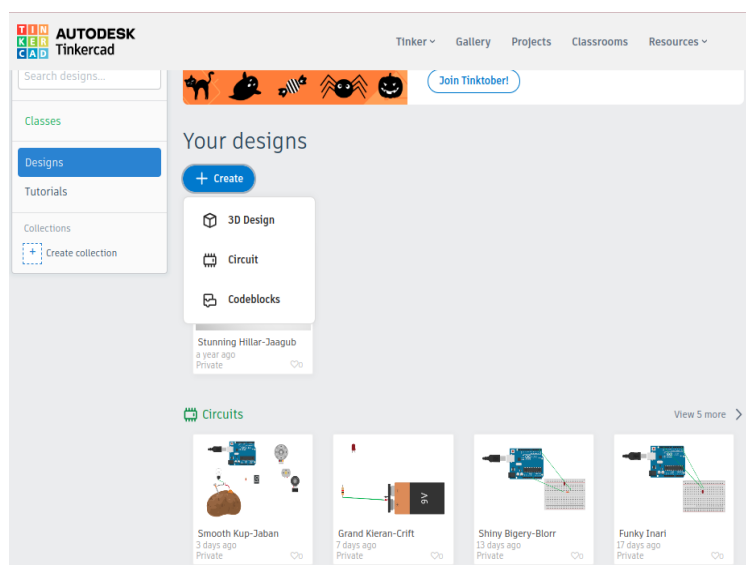
Вивчити основи роботи на платформі Tinkercad і симулювати роботу контролера Arduino та інших електронних елементів.

План

1. Вивчення платформи Tinkercad.
2. Створення облікових записів та проєктів.
3. Вивчення особливості симуляції Arduino на платформі Tinkercad.

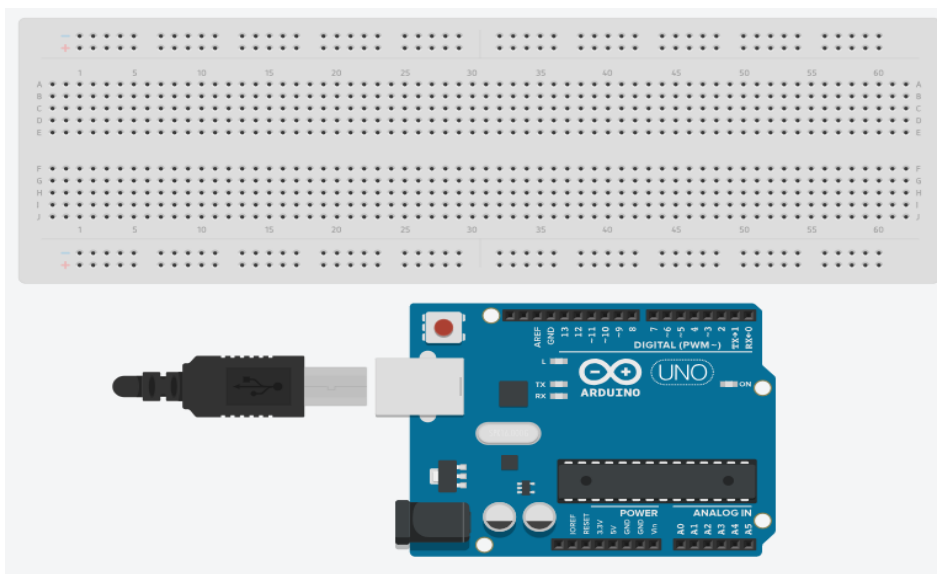
Теоретичні відомості

Для того, щоб симулювати роботу контролера Arduino, не обов'язково мати безпосередньо плату Arduino. Для симуляції її роботи достатньо зареєструватися у вебсередовищі **Autodesk Tinkercad** (<https://www.tinkercad.com/>). Є різні варіанти і способи реєстрації, однак кожен із них дає змогу отримати доступ до онлайн симулятора.

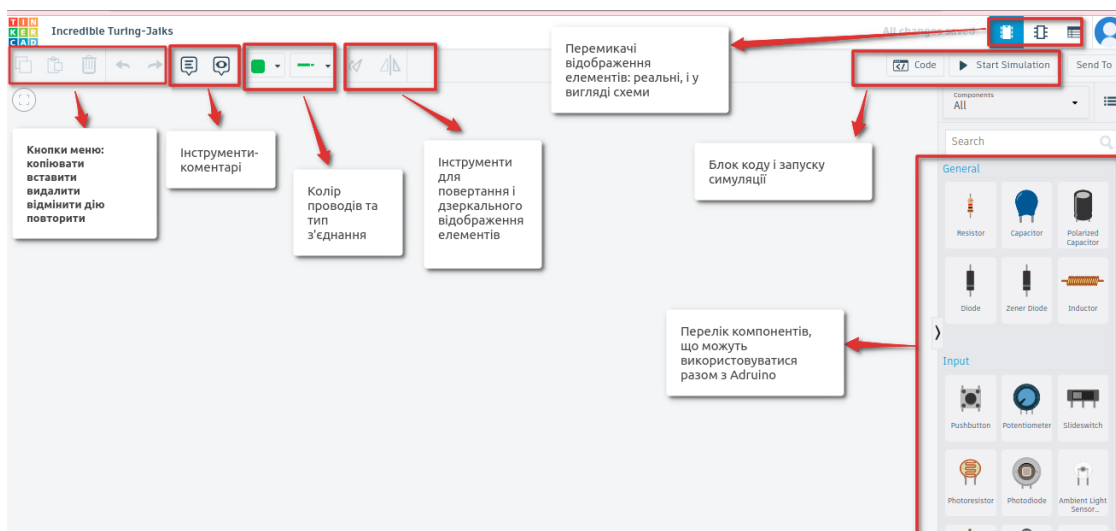


Після реєстрації треба вибрати **Designs** (конструкції). У цьому розділі можуть бути доступні створені користувачам схеми

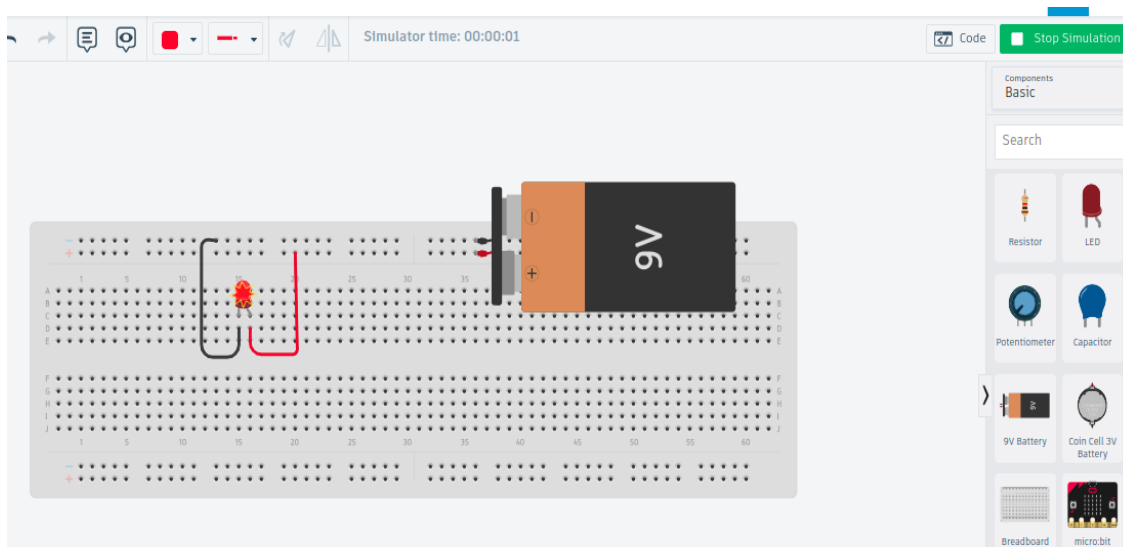
(Circuits), а також можливість створити нову схему: **Create**→**Circuits**. Після створення нової схеми відкривається вікно, що подано на рисунку нижче.



У переліку компонент є як плата Arduino, так і макетна плата, якою будемо користуватися для складання навчальних схем і моделей. Макетна плата складається з чотирьох областей. Зверху і знизу є два ряди отворів, що призначені для створення шин живлення. Усі отвори ряду пов'язані між собою. Вважаємо, що через отвір одного ряду до цих шин підводиться напруга живлення (додатний і від'ємний контакт), а потім через інші отвори до живлення можна буде підключати компоненти нашої схеми.

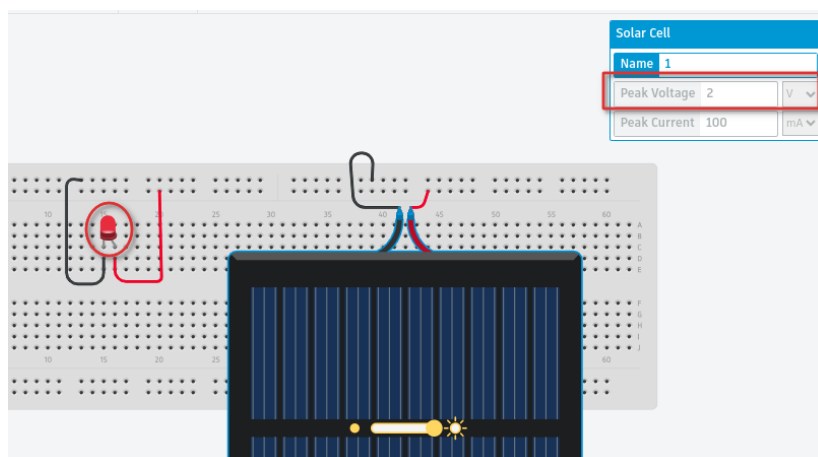


Отвори у кожній з двох вертикальних областей, навпаки, пов'язані вертикально: вони з'єднані в рядах А-В-С-D-E та F-G-H-I-J. Ці області використовуються для встановлення та підключення електронних пристроїв.



Давайте, додамо елемент світлодіод та джерело живлення на 9 вольт. З'єднаємо як показано на рисунку і включимо симуляцію.

Симуляція показала, що світлодіод згорів, бо напруга на світлодіод 9В є зовеликою. Для налагодження схеми виключимо симуляцію і додамо резистор 450 Ом, або змінимо джерело напруги на слабше напругою 2В. До речі, кожен елемент має свої властивості, де можна задавати параметри. Наприклад, джерело живлення “сонячна батарея” має можливість вказати максимальну напругу (саме там ми вказали 2В). Як видно, при вмиканні симуляції світлодіод працює і світиться.



Отже, платформа Autodesk Tinkercad дає можливість перевіряти будь-які схеми і їх підключення (у тому числі плати Arduino), імітувати їх роботу, експериментувати і не боятися спалити реальні електронні елементи.

Завдання

1. Зареєструйтеся на Autodesk Tinkercad.
2. Зберіть схему, що подана вище. Перевірте її роботу.

Контрольні питання

1. Що потрібно для симуляції роботи Arduino?
2. Що таке Autodesk Tinkercad?
3. Як зареєструватися в Autodesk Tinkercad?
4. Які області має макетна плата?
5. Як з'єднані отвори у макетній платі?
6. Як запустити симуляцію в Autodesk Tinkercad?
7. Чому світлодіод може згоріти, якщо на нього подати 9 В?
8. Чи є можливість регулювати параметри компонентів в Autodesk Tinkercad?
9. Назвіть компоненти, які є в Autodesk Tinkercad?
10. Де можна писати скетч в Autodesk Tinkercad?
11. Як задати колір проводів і тип з'єднання?
12. Де можна перемкнути зображення реальних елементів на схематичні в Autodesk Tinkercad?

Лабораторна робота № 3

Цифрові виходи Arduino. Робота зі світлодіодом

Мета

Вивчити можливості під'єднання світлодіодів до контролера Arduino. Налаштування програми для керування світлодіодом.

План

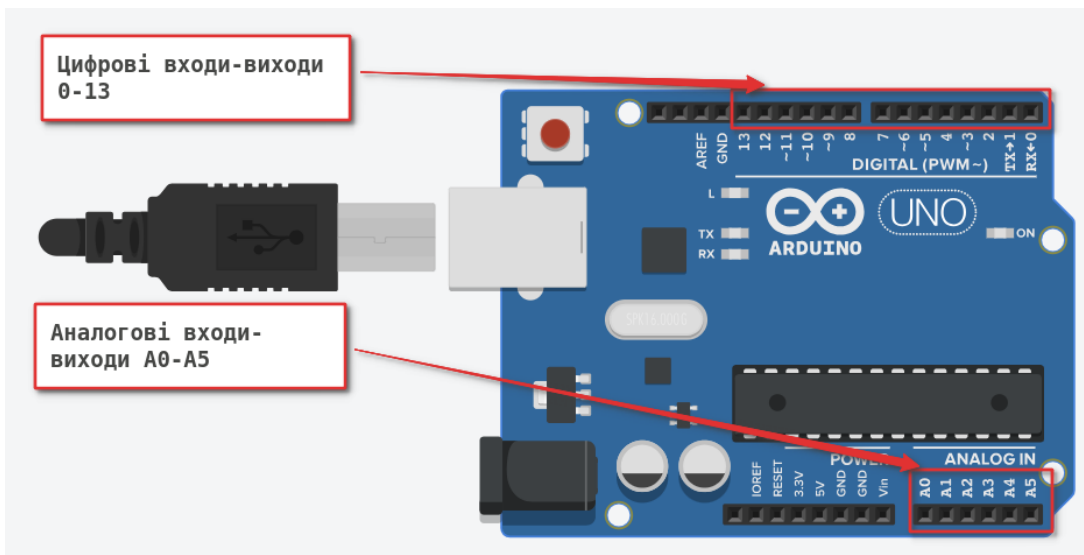
1. Вивчення цифрових виходів плати Arduino.
2. Дослідження особливостей світлодіодів та електричних схем їх вмикання.
2. Підключення світлодіода до Arduino налаштування його світіння через цифрові порти.
3. Проведення віртуальної симуляції та реальної роботи світлодіода з контролером Arduino.
4. Під'єднання декількох світлодіодів.

Теоретичні відомості

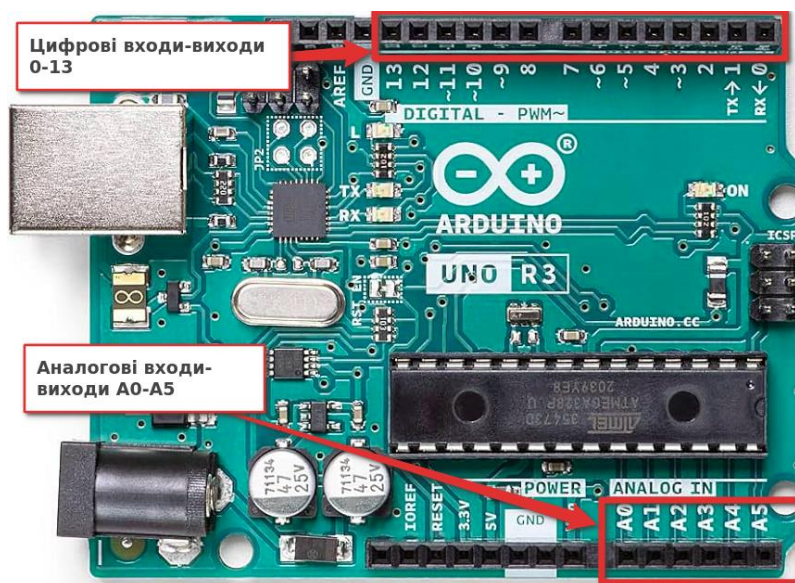
Одна з найпростіших дій, що може робити контролер Arduino, – це перемикання станів контактів цифрових входів-виходів. Будь-який із цифрових входів-виходів мікроконтролера Arduino можна програмним способом перемикати між високим рівнем (+5В) і низьким (0В). За допомогою такого перемикання відбувається керування простими пристроями. Перемикання між рівнями здійснюється за допомогою команди:

digitalWrite(номер_контакту, рівень_сигналу);

Пам'ятаємо, що команди у C++ закінчуються символом “;” (крапка з комою).



Параметр сигналу (**рівень_сигналу**) може набувати два значення: **HIGH, LOW** (високий і низький). А **номер_контакту** – це число, що відповідає номеру контакту плати Arduino.



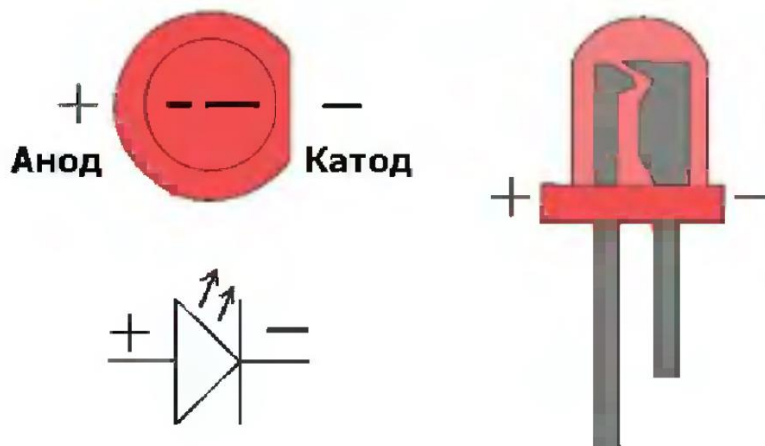
Зверніть увагу, що одні і ті ж цифрові контакти можуть виступати у ролі входу і виходу сигналу. Тому перед їх використанням необхідно вказати у якій ролі будуть використовуватися відповідні контакти (вхід чи вивід). Це робиться за допомогою функції:

pinMode(номер_контакту, режим_контакту);

де **режим_контакту** – це два значення: **OUTPUT** (вихід) або **INPUT** (вхід). Отже, для того, щоб встановити на виводі № 2 високий рівень сигналу, досить виконати таку програму:

```
const int pin = 2;
void setup()
{
  pinMode(pin, OUTPUT);
  digitalWrite(pin, HIGH)
}
void loop()
{
}
```

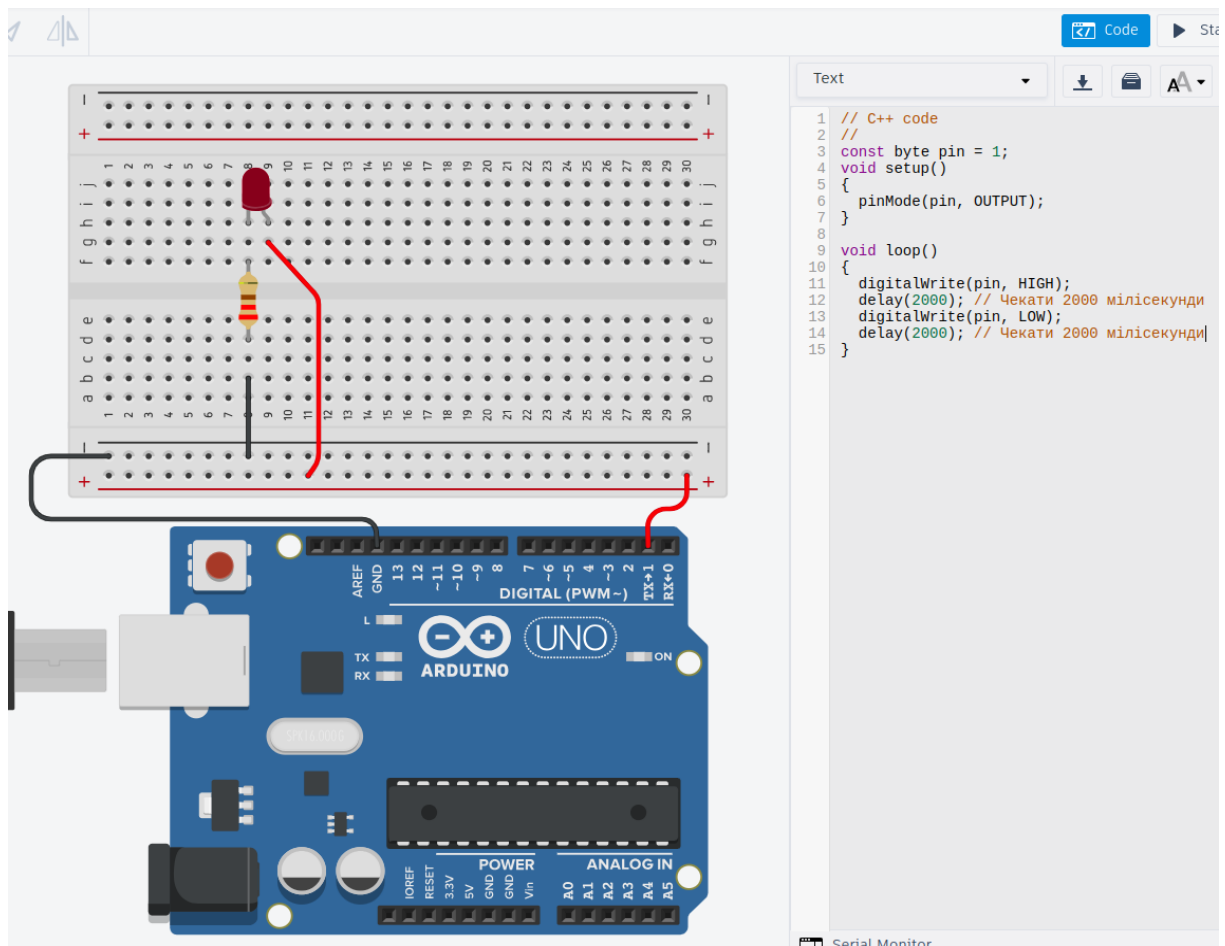
Розглянемо одне з найпростіших підключень виконавчих пристроїв, якими можна керувати за допомогою цифрового контакту.



Це – підключення *світлодіода*, тобто, напівпровідникового пристрою, що може випромінювати світло при пропусканні через нього електричного струму у прямому напрямку (від катода до анода). Для того, щоб правильно включити світлодіод в електричне коло, необхідно відрізнити катод від анода. Це можна зробити за трьома ознаками:

- 1) анод має довший провід;
- 2) зі сторони катода корпус світлодіода;
- 3) у середині світлодіода катод ширший.

Підключимо світлодіод до плати Arduino і “змусимо” його блимати з інтервалом 2 секунди. Для цього зберемо схему, як подано на рисунку. Опір резистора встановимо 220 Ом.



Також впишемо програму у відповідне вікно. Після запуску симуляції відбудеться блимання світлодіода. Таку ж схему можна скласти реально, під'єднати плату Arduino до комп'ютера і завантажити програму через середовище Arduino IDE.

Завдання

1. Реалізуйте схему, що описана вище.
2. Під'єднайте ще один синій світлодіод та організуйте чергове блимання (синій-червоний), наче проблискові маячки.

Контрольні питання

1. Які є входи і виходи в платі Arduino?
2. Яка напруга можна подаватися на цифровий контакт?
3. Скільки цифрових контактів є у платі Arduino UNO?
4. У якому режимі можуть працювати контакти плати Arduino?
5. Як задати режим роботи контакту плати Arduino?
6. Що таке контакт Ground?
7. Скільки контактів має світлодіод?
8. Як розпізнати катод і анод?
9. Який знак напруги треба подати на катод?
10. Який знак напруги треба подати на анод?
11. Яка команда “каже” платі Arduino зробити паузу у роботі?

Лабораторна робота № 4

Підключення кнопок та вимикачів

Мета

Вивчити можливості під'єднання кнопок, вимикачів і зумера до контролера Arduino. Налаштування програми для керування світлодіодом через вимикачі.

План

1. Вивчення особливостей вимикачів і кнопок та їх вмикання.
2. Підключення світлодіода, кнопок, вимикачів і зумера до Arduino, налаштування їх роботи.
3. Проведення віртуальної симуляції та реальної роботи світлодіода і кнопок з контролером Arduino.

Теоретичні відомості

Вимикач – це пристрій, що дає змогу замикати чи розмикати електричне коло. Зміна стану вимикача може відбуватися різними способами залежно від типу пристрою. У цій лабораторній роботі ми будемо використовувати простий механічний вмикач-вимикач – (**pushbutton**), який замикає свої два контакти при натисканні. Для зручності монтажу їх часто роблять у вигляді 4-х контактів, у яких виходи з'єднані попарно.

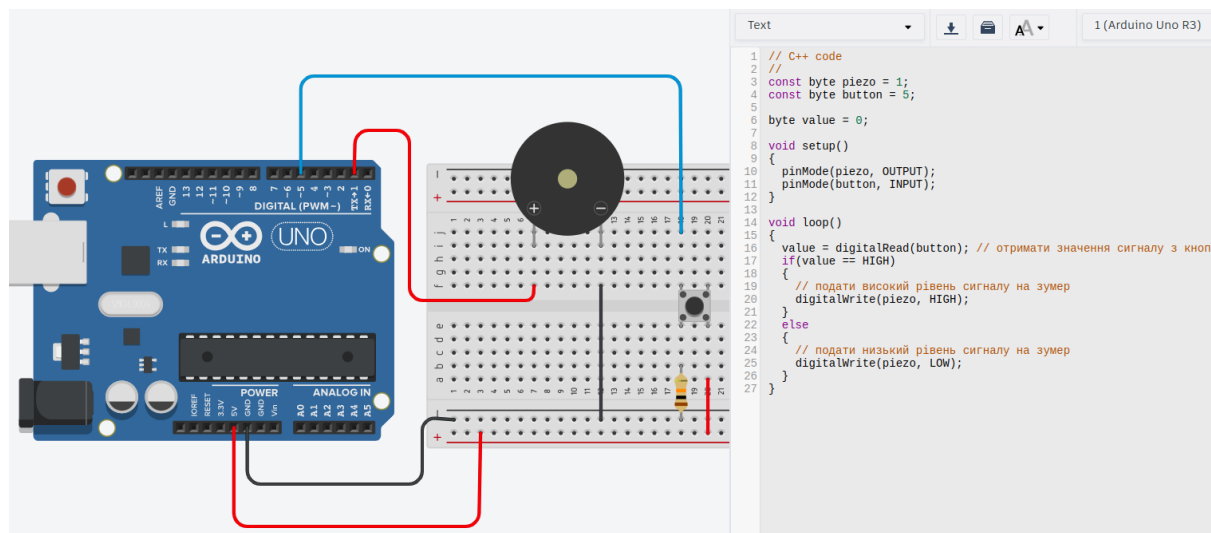
Для зчитування сигналу з одного з контактів Arduino, необхідно скористатися функцією

res = digitalRead(номер_контакту).

Після виклику цієї функції змінна **res** буде зберігати рівень цифрового сигналу, що детектується на відповідному контакті.

Щоб читати стан кнопки, її треба підключити так, щоб при її натисканні на вході був сигнал високий (+5 В), а при її відпусканні – низький (0 В). Для цього підключимо послідовно між шиною живлення і землею (GROUND) кнопку і резистор, а вхід мікроконтролера підключимо між ними. Тоді, якщо кнопка НЕ натиснута, то вихід мікроконтролера буде з'єднаний через резистор із землею і

буде мати низький рівень сигналу. Якщо кнопку нажати, то вхід буде з'єднаний із живленням і буде мати високий рівень сигналу. При такій схемі підключення через резистор потече струм, що залежить від його опору. І щоб схема не споживала багато енергії, візьмемо резистор з великим опором, наприклад 10 кОм.



Під'єднаємо також до плати ще **Piezo** зумер (пристрій, що відтворює звуковий сигнал, коли на нього поступає високий рівень напруги). Організуємо його роботу так: коли кнопка натиснута, то зумер відтворює звук, а коли кнопку відпустили, то звук не йде. У такий спосіб можна організувати вивід звукового сигналу.

Завдання

1. Відтворіть схему, яку подано в інструкції на симуляторі і на реальному пристрої.

2. Додайте до попередньої схеми ще світлодіод. Організуйте, що при НЕ натиснутій кнопці світлодіод світиться, і зумер виключений, а при натиснутій – світлодіод згасає і зумер відтворює звук.

Контрольні питання

1. Як зчитати рівень сигналу з цифрового контакту Arduino?
2. Що таке pushbutton?
3. Як з'єднані контакти у 4 контактній кнопці?
4. Як під'єднують зумер до плати Arduino?

5. У чому полягає ідея підключення резистора до схеми, що подана в інструкції?
6. Який зміст констант HIGH і LOW?
7. Як можна організувати постійне звучання зумера протягом 10 секунд?
8. Як можна “запам’ятати” у програмі, що кнопку була натиснута?
9. Чи можна регулювати напругу на контактах, що позначені 5V і 3.33V?

Лабораторна робота № 5

Семисегментний діодний індикатор

Мета

Вивчити семисегментний індикатор. Навчитися налаштовувати індикатор і організувати вивід інформації через нього.

План

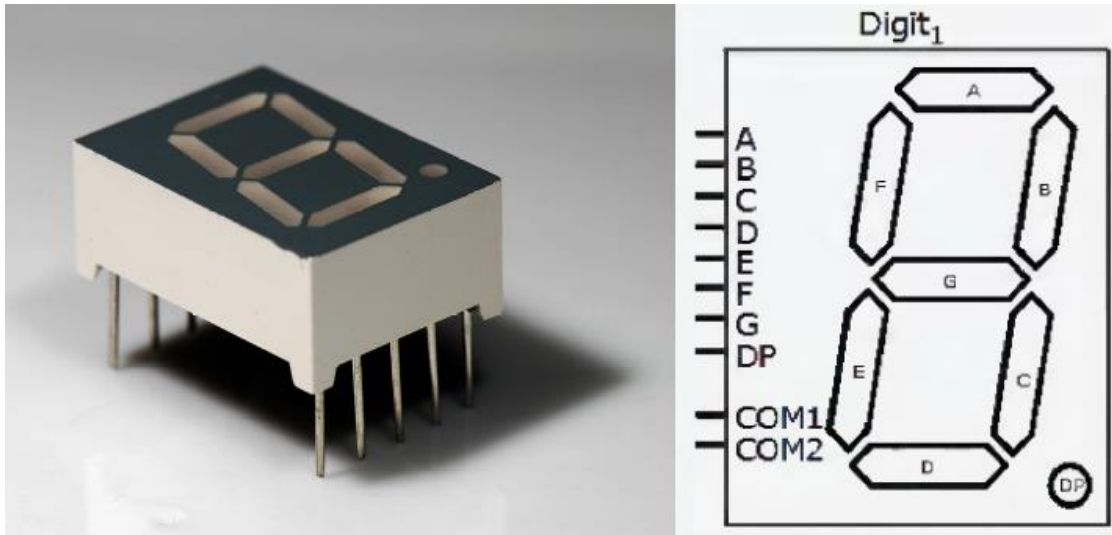
1. Вивчення особливостей семисегментного індикатора.
2. Підключення семисегментного індикатора до Arduino, налаштування його роботи.
3. Проведення віртуальної симуляції та реальної роботи семисегментного індикатора контролером Arduino.
4. Організувати таймер.

Теоретичні відомості

Звичайним світлодіодом можна легко інформувати користувача про які-небудь бінарні події, тобто про події, про які важливо знати, чи вони відбулися, чи ні. Наприклад, вмикання або вимикання якого-небудь пристрою, про перевищення допустимого значення, тощо.

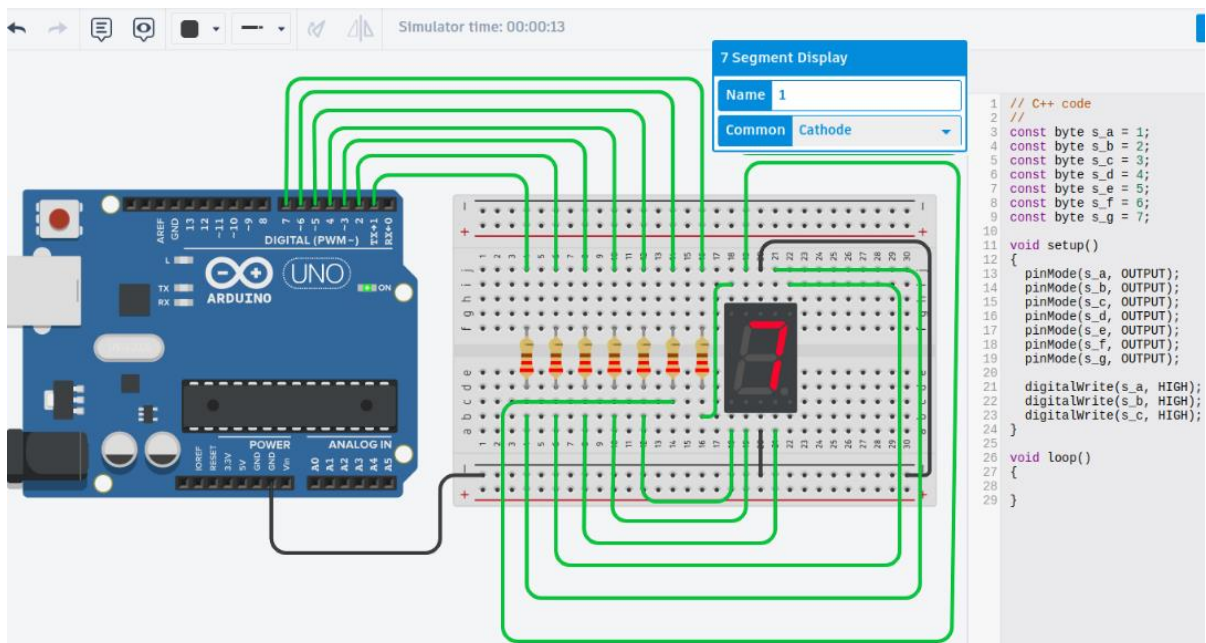
Що ж робити, якщо ж потрібно надати користувачеві складнішу інформацію, наприклад, числову. Для цих цілей в електроніці часто використовується семисегментний світлодіодний індикатор. Цей прилад складається з набору світлодіодів, що розташовані у вигляді числа “8” так, що, вмикаючи деякі з них, можна отримати контури різних фігур.

За конструкцією семисегментні світлодіоди поділяються на групи зі спільним анодом і спільним катодом. Усі сегменти марковані латинськими літерами від “A” до “G”. Як правило відлік ведеться з верхнього сегмента за годинниковою стрілкою. Точка маркується окремо, словом “dot”. Крім контактів, що з'єднані з сегментами, на схемі видно ще два контакти COM (це спільні катод чи анод) і один DP (точка).



Підключення семисегментного індикатора здійснюється так само як і світлодіода. Оскільки в індикаторі є 7 контактів (крім точки і спільних катода чи анода), то потрібно 7 цифрових виходів Arduino та 7 резисторів по 220 Ом.

Подамо схему підключення і програму, що виведе число “7” на індикаторі. Використаємо індикатор зі спільним катодом. Нижче на рис подано схему підключення і програмний код.



Як видно з рисунка на індикаторі відтворено число “7”.

Завдання

1. Відтворити схему подану вище.
2. Реалізувати лічильник, що виводить числа від 0 до 9 з інтервалом 1 секунда.

Контрольні питання

1. З чого складається семисегментний діодний індикатор?
2. Як нумеруються контакти семисегментного індикатора?
3. Як маркується точка в індикаторі?
4. Скільки контактів має одноцифровий семисегментний діодний індикатор?
5. Яких типів можуть бути семисегментні діодні індикатори?
6. Як треба підключати семисегментний діодний індикатор до Arduino?
7. Що потрібно, щоб відобразити цифру на індикаторі?
8. Який тип одноцифрового семисегментного діодного індикатора використано у лабораторній роботі?
9. Як зміниться схема підключення, якщо змінити тип одноцифрового семисегментного діодного індикатора?
10. Що потрібно, щоб реалізувати лічильник?

Лабораторна робота № 6

Рідкокристалічний дисплей

Мета

Вивчити особливості рідкокристалічного дисплея і його використання.

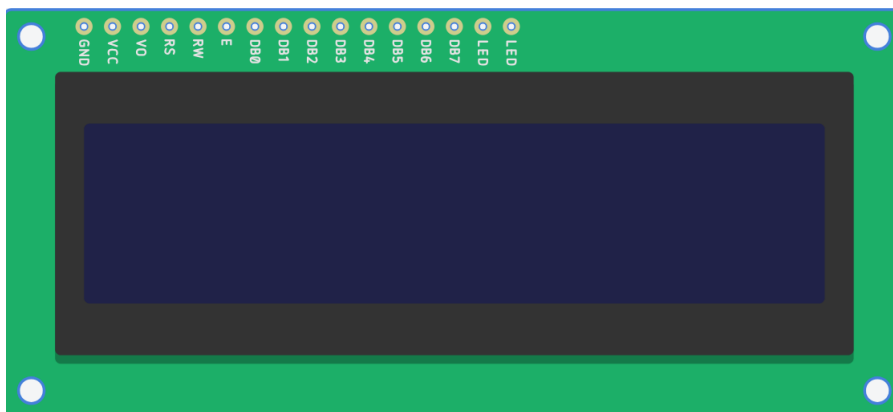
План

1. Вивчення особливостей рідкокристалічного дисплея.
2. Підключення рідкокристалічного дисплея до Arduino, налаштування його роботи і виведення даних з нього.
3. Проведення віртуальної симуляції та реальної роботи щодо виведення текстів на екран.

Теоретичні відомості

На попередній роботі ви ознайомилися з семисегментним діодним індикатором. Однак більш інформативний і водночас більш складний тип індикатора – це дисплей. У цій лабораторній роботі будемо працювати з рідкокристалічним індикатором, представники якого є у багатьох пристроях: електронний годинник, калькулятор, екрани старих (часто кнопочних) телефонів. Усі вони певною мірою використовують рідкокристалічні дисплеї.

У нас буде використовуватися дисплей 16x2. Це означає, що він “уміє” показувати 2 рядки по 16 символів. Зовнішній вигляд дисплея і його позначення подано на рисунку нижче.



Позначення індикаторів таке:

GND – земля (мінус живлення);

VCC – живлення (+5V);

V0 – контраст (регулятор контрасту дисплея);

RS – вибір регістра;

RW – напрямок передачі даних (читання чи запис);

E – (enabled) синхронізація;

DB0-DB7 – шина даних;

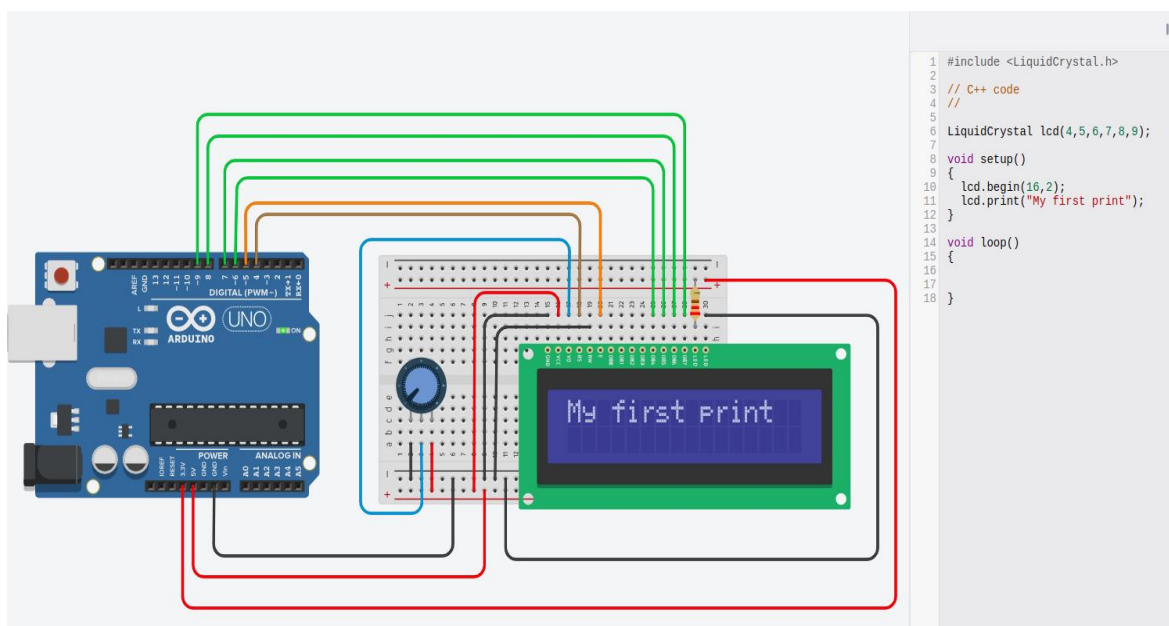
Led- – катод підсвічування;

Led+ – анод підсвічування.

Схему підключення дисплея подано на рисунку нижче. Крім підключення інформаційних сигналів і живлення, треба забезпечити регуляцію контрасту дисплея. Для цього до схеми під'єднують ще потенціометр 10 кОм. Якщо ж ще потрібне підсвічування дисплея, то його підключають за допомогою LED контактів через резистор мінімум 100 Ом.

Для роботи з рідкокристалічним дисплеєм в Arduino є спеціальна бібліотека LiquidCrystal. Вона підключається директивою **#include <LiquidCrystal.h>**. Після цього треба ініціалізувати дисплей, вказавши які саме контакти використовуються для Arduino:

LiquidCrystal lcd(4,5,6,7,8,9).



Тут перший контакт – RS, другий – Enabled, з третього до шостого DB4-BD7. Інші контакти дисплея не потрібно підключати до Arduino. У бібліотеці також передбачена функція, що дає змогу вказати позицію курсора: **lcd.setCursor(0,1)** (колонка з індексом 0 і рядок з індексом 1). Для очищення екрану використовується **lcd.clear()**.

На рисунку вище подано схему і програму, що виводить на екрані повідомлення “My first print”. Також там підключено підсвічування через резистор 220 Ом до живлення +3В.

Завдання

1. Відтворити схему, що наведено вище.
2. Реалізуйте біжучий рядок із довільним текстом.

Контрольні питання

1. Для чого використовують рідкокристалічний дисплей?
2. Яку приклади рідкокристалічного дисплея вам відомі?
3. Який тип дисплея використано у лабораторній роботі?
4. Як позначаються контакти у рідкокристалічному дисплеї?
5. Який контакт регулює контраст дисплея?
6. Чи обов’язково підключати контакти Led? А якщо підключити, то для чого?
7. Яка бібліотека використовується для роботи з рідкокристалічним дисплеєм?
8. Як ініціалізується бібліотека для роботи з рідкокристалічним дисплеєм?
9. Якою директивою підключається бібліотека для роботи з рідкокристалічним дисплеєм?
10. Які контакти достатньо під’єднати, щоб мати змогу щось виводити на екран?
11. Як встановити позицію курсора?
12. Як очистити екран?
13. Яка команда виводить повідомлення на екран?
14. Яку ідею закласти у створення біжучого рядка?

Лабораторна робота № 7

Сенсори. Резистивні сенсори

Мета

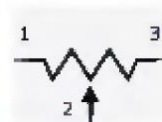
Вивчити особливості роботи сенсорів. Навчитися опрацьовувати аналогові сигнали.

План

1. Вивчення особливостей аналогових сигналів.
2. Підключення реостата чи потенціометра до Arduino, налаштування його роботи і знімання даних з нього.
3. Проведення віртуальної симуляції та реальної роботи щодо виведення значення напруги з реостата.

Теоретичні відомості

У попередніх лабораторних роботах для обмеження сили струму через світлодіод використовувалися резистори. Крім звичайних резисторів, є ще аналогічні пристрої зі змінним опором, які називаються потенціометрами або реостатами. На рисунку вище подано зображення реостатів і їх позначення на схемах.



Для чого потрібен потенціометр? По-перше, виходячи з опису пристрою, його можна використовувати для надання значення потрібного опору у випадку, коли нема потрібного або коли потрібне точне регулювання уже готового пристрою. По-друге, за

допомогою потенціометра можна створити регульовальний дільник напруги – пристрій, що може передавати на навантаження лише частину напруги від джерела. У цій лабораторній роботі ми будемо використовувати реостат як дільник напруги, змінюючи значення на одному із аналогових контактів Arduino.

Під час роботи з кнопками ми уже ознайомилися з функцією `digitalRead`, яка може зчитувати цифровий сигнал з відповідного контакту. У цієї функції існує також аналогова версія `analogRead`, яка дає змогу робити те саме тільки для аналогового сигналу:

`res = analogRead(номер_контакту).`

Після її виклику у програмі змінна `res` буде зберігати значення аналогового сигналу, який зчитаний з відповідного контакту. При цьому функція `analogRead` повертає число від 0 до 1023, де 0 означає 0В, а 1023 означає +5В. Проміжним значенням напруг будуть відповідати проміжні значення чисел з діапазону 1-1022. Підключати потенціометр треба до контактів А0-А5.

Завдання

1. Під'єднайте потенціометр до Arduino (аналогічно як у попередній роботі).
2. Напишіть програму, що буде кожні 2 секунди визначати значення напруги на потенціометрі та виводити його на рідкокристалічний дисплей.

Контрольні питання

1. Що таке потенціометр?
2. Для чого потрібен потенціометр?
3. Що таке дільник напруги?
4. Як працює функція `analogRead`?
5. Які значення можуть поступати з аналогових контактів?
6. Як позначаються аналогові контакти?
7. Скільки аналогових контактів є на платі Arduino UNO?
8. Яка ідея лежить в основі реалізації виводу значення напруги на дисплей?

Лабораторна робота № 8

Аналоговий датчик температури

Мета

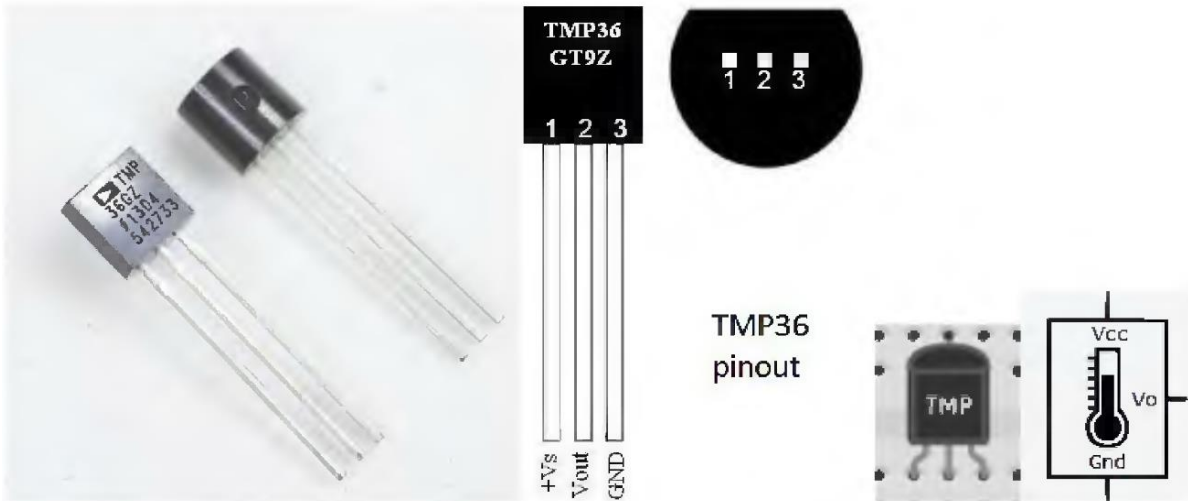
Вивчити особливості роботи з датчиком температури. Навчитися опрацьовувати дані про температуру.

План

1. Вивчення датчика температури і його параметрів.
2. Калібрування датчика температури.
3. Проведення віртуальної симуляції та реальної роботи щодо виведення значення температури у приміщенні.

Теоретичні відомості

Поширеним варіантом датчиків з аналоговим виходом є датчики температури. Позначення контактів таких датчиків таке:



V_s – живлення напругою від 2.7 до 5.5 В (для Arduino 5В);

V_{out} – вихідна напруга, що залежить від температури;

Gnd – земля.

Вихідна напруга на сенсорі залежить тільки від температури і не залежить від живлення, що подається на датчик.

Для того, щоб визначити температуру, треба визначити максимальне значення T_{max} і мінімальне значення T_{min} температури, що

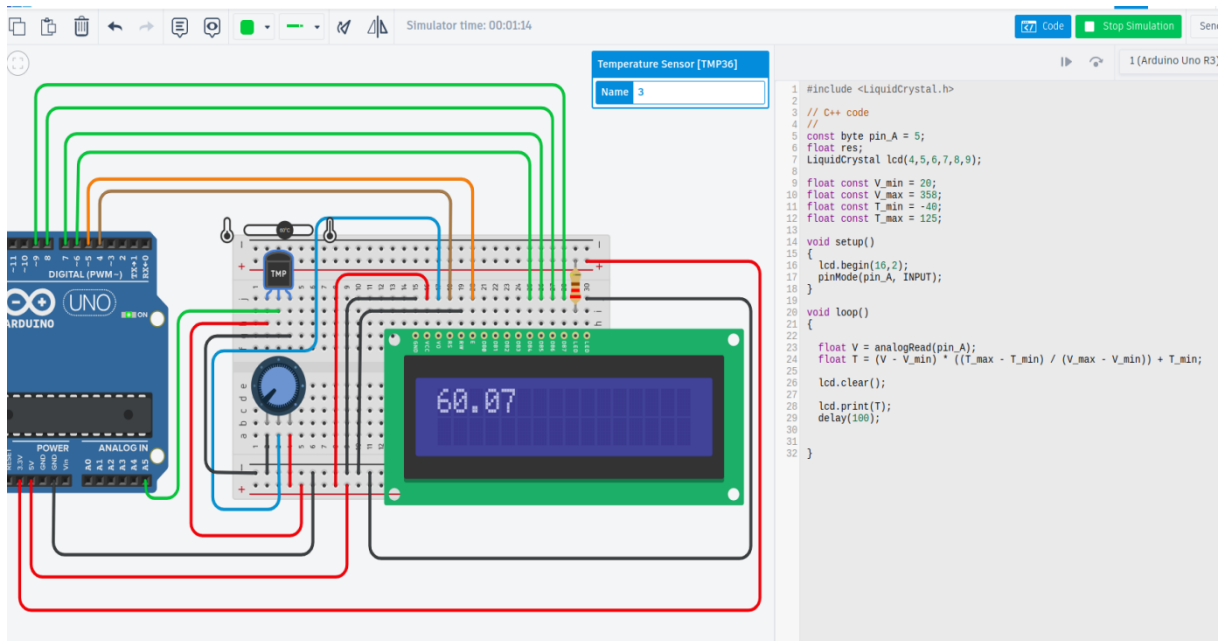
може вимірювати датчик. Інакше кажучи треба градувати шкалу. Для мінімального і максимального значення температури є мінімальне V_{min} і максимальне V_{max} значення напруги, що поступає на контакт Arduino. Тоді зважаючи на рівномірну шкалу, знайдемо зв'язок між напругою, що поступає на контакт Arduino, і температурою:

$$T = (V - V_{min}) \frac{T_{max} - T_{min}}{V_{max} - V_{min}} + T_{min}$$

Тут V – вихідна напруга у вольтах, T – температура у градусах Цельсія. V – це також значення, що отримано у результаті виклику функції

$$V = \text{analogRead}(\text{номер_контакту}).$$

Приклад підключення такого датчика подано на рисунку нижче.



У прикладі на симуляторі максимальна і мінімальна напруга є 20 і 385 В, а діапазон температур – [-40...125] °С.

Завдання

1. Відтворіть схему, подану вище.
2. Під'єднайте до схеми рідкокристалічний дисплей, на який виводиться значення температури. Відкалібруйте шкалу.

Контрольні питання

1. Як працює датчик температури?
2. До якого типу належить датчик температури?
3. Скільки контактів і які саме має датчик температури?
4. Від чого залежить вихідна напруга на датчику температури?
5. Яка шкала у датчика температури?
6. У яких одиницях вимірюється сигнал, що йде від датчика температури?
7. Як пов'язана напруга та температура на датчику температури?
8. З яких міркувань можна отримати формулу для зв'язку напруги та температури на датчику температури?
9. Яку напругу живлення можна подавати на датчик температури?
10. Як можна регулювати температуру на симуляторі?
11. До якого контакту плати Arduino треба під'єднувати контакт датчика температури Vout?

Лабораторна робота № 9

Датчик світла

Мета

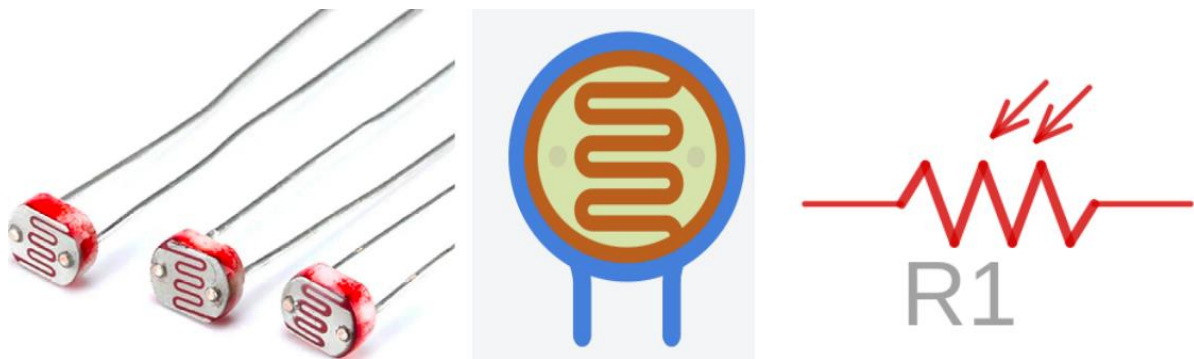
Вивчити особливості роботи з датчиком світла. Навчитися опрацьовувати дані про освітлення.

План

1. Вивчення датчика світла (фоторезистора) і його параметрів.
2. Калібрування датчика освітлення.
3. Проведення віртуальної симуляції та реальної роботи пристрою, який автоматично вмикає / вимикає світло залежно від зовнішнього освітлення.

Теоретичні відомості

У цій лабораторній роботі ми розглянемо ще один тип аналогового датчика – датчик світла, який ще також називають фоторезистором.



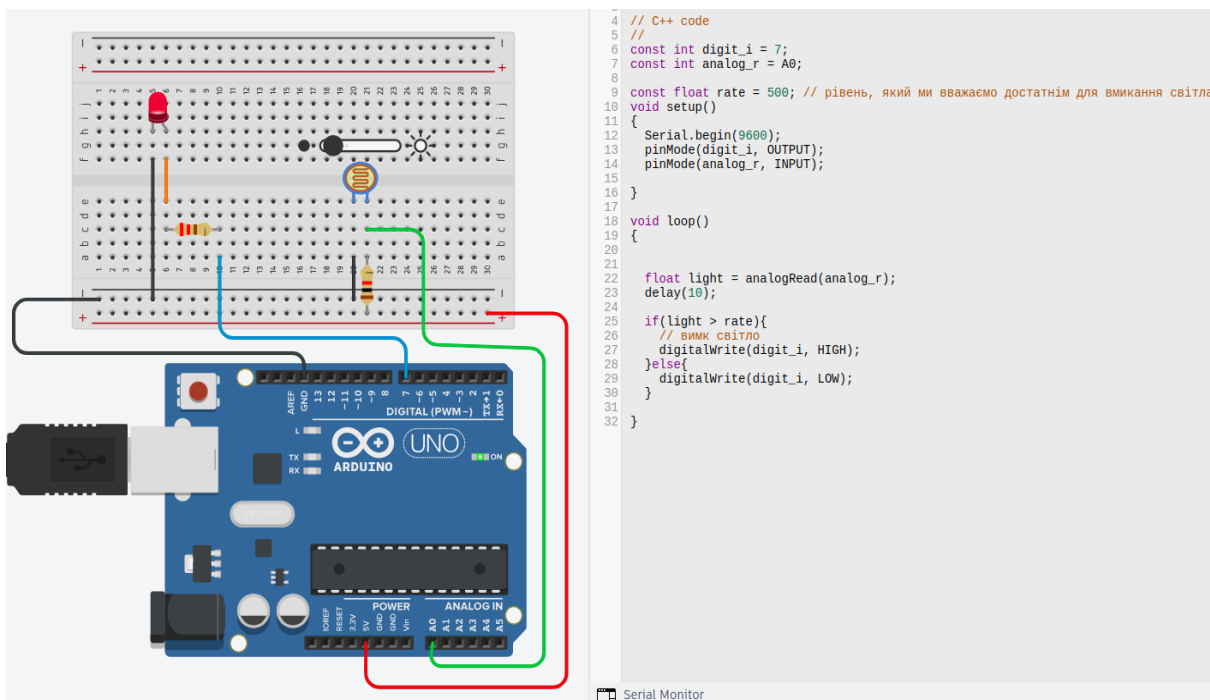
Це простий компонент з двома контактами, який змінює свій опір залежно від рівня освітлення. На відміну від датчика температури, фоторезистор не видає напругу, а лише змінює свій опір при зміні освітлення.

Щоб зчитати покази фоторезистора треба спочатку зміню опору перетворити на зміню напруги. Для цього використовують схему, “дільник напруги”. Під’єднаємо один контакт фоторезистора з напругою живлення, а до другого контакту підключимо звичайний

резистор, що з'єднаний із землею. Напруга на резисторах буде пропорційною їх опорам. Наприклад, якщо опір фоторезистора 20 кОм, а звичайного резистора 10 кОм, то напруга на фоторезисторі буде у два рази більшою, ніж на звичайному. У нас напруга живлення 5В, тоді ці значення будуть 3.333 і 1.666 вольт. При зміні опору фоторезистора напруги будуть змінюватися. З'єднавши середню точку між резисторами з аналоговим контактом Arduino, можна вимірювати напругу на звичайному резисторі та робити висновки про освітленість.

Фоторезистори, що йдуть у комплекті з платами Arduino, як правило, змінюють опір від 200 кОм (повна темнота) до 1 кОм (яскравість 10 люкс). Експериментуючи з датчиком, можна помітити, що показники на аналоговому контакті змінюються не лінійно. У цьому і є особливість датчика. Для температурного датчика залежність була лінійна.

Складемо схему, де є резистор, фоторезистор і світлодіод. Налаштуємо її так, щоб при мінімальному освітленні вмикався світлодіод.



Завдання

1. Відтворіть схему подану вище.
2. Підключіть світлодіод до аналогового контакту і порегулюйте його світіння залежно від напруги на фоторезисторі.

Контрольні питання

1. Що таке фоторезистор?
2. До яких типів аналогових чи цифрових належить фоторезистор?
3. Скільки контактів має фоторезистор?
4. Як підключати контакти фоторезистора? Чи має значення де “+” і “-”?
5. Що змінює фоторезистор при зміні освітлення?
6. Як реалізують підключення “дільник напруги”?
7. Коли опір фоторезистора більший? У повній темноті чи при яскравому світлі?
8. Яка залежність опору фоторезистора? Лінійна чи нелінійна?
9. Як на симуляторі можна регулювати освітлення, що потрапляє на фоторезистор?
10. Чи відрізняється світлодіод від фоторезистора?
11. До якого контакту Arduino треба під'єднувати фоторезистор?
12. Що таке константа “rate”, що наведена у програмі?

Список використаних джерел

1. Arduino.ua. URL: <https://arduino.ua/>.
2. Arduino blog. URL: <https://blog.arduino.cc/>.
3. Arduino-DIY. URL: <http://arduino diy.com/>.
4. Margolis M. Arduino Cookbook. O'Reilly Media, 2011. 662 p.
5. Perea F. Arduino Essentials. Packt Publishing, 2015. 206 p.
6. Tinkercad. URL: <https://www.tinkercad.com/>.
7. Microsoft C++. URL: <https://learn.microsoft.com/uk-ua/cpp/?view=msvc-170>.
8. Грабко В.В., Розводюк М.П., Грабко Вал.В. Мікропроцесорні системи керування електроприводами. Вінниця : ВНТУ, 2012. 97 с.
9. Користувацькі посібники по роботі із платформою Arduino. URL: <https://www.arduino.cc/en/Tutorial/HomePage>.
10. Методичні вказівки до лабораторних робіт з дисципліни “Електротехніка та електроніка” для студентів спеціальності 122 “Комп’ютерні науки і інформаційні технології” денної форм навчання / укл.: А.В. Пархоменко, О.М. Гладкова. Запоріжжя : ЗНТУ, 2016. 41 с.
11. Програма технічного конструювання. Програми з поза-шкільної освіти науково-технічний напрям (інформаційно-технічний профіль). Київ, 2014. С. 15–32.
12. Ресурс habr.com. URL: <https://habr.com/search/?q=Arduino#h>.

Електронне навчально-методичне видання

Роман Лешко

**ІНФОРМАЦІЙНО-КЕРУЮЧІ СИСТЕМИ
ТА STEM-ТЕХНОЛОГІЇ**

**Дрогобицький державний педагогічний університет
імені Івана Франка**

Редактор

Ірина Невмержицька

Технічний редактор

Ольга Лужецька

Коректор

Ірина Артимко

Здано до набору 22.11.2023 р. Формат 60x90/16. Гарнітура Times.
Ум. друк. арк. 2,5. Зам. 70.

Дрогобицький державний педагогічний університет імені Івана Франка.
(Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру
видавців, виготівників та розповсюджувачів видавничої продукції ДК № 5140
від 01.07.2016 р.). 82100, Дрогобич, вул. Івана Франка, 24, к. 31.