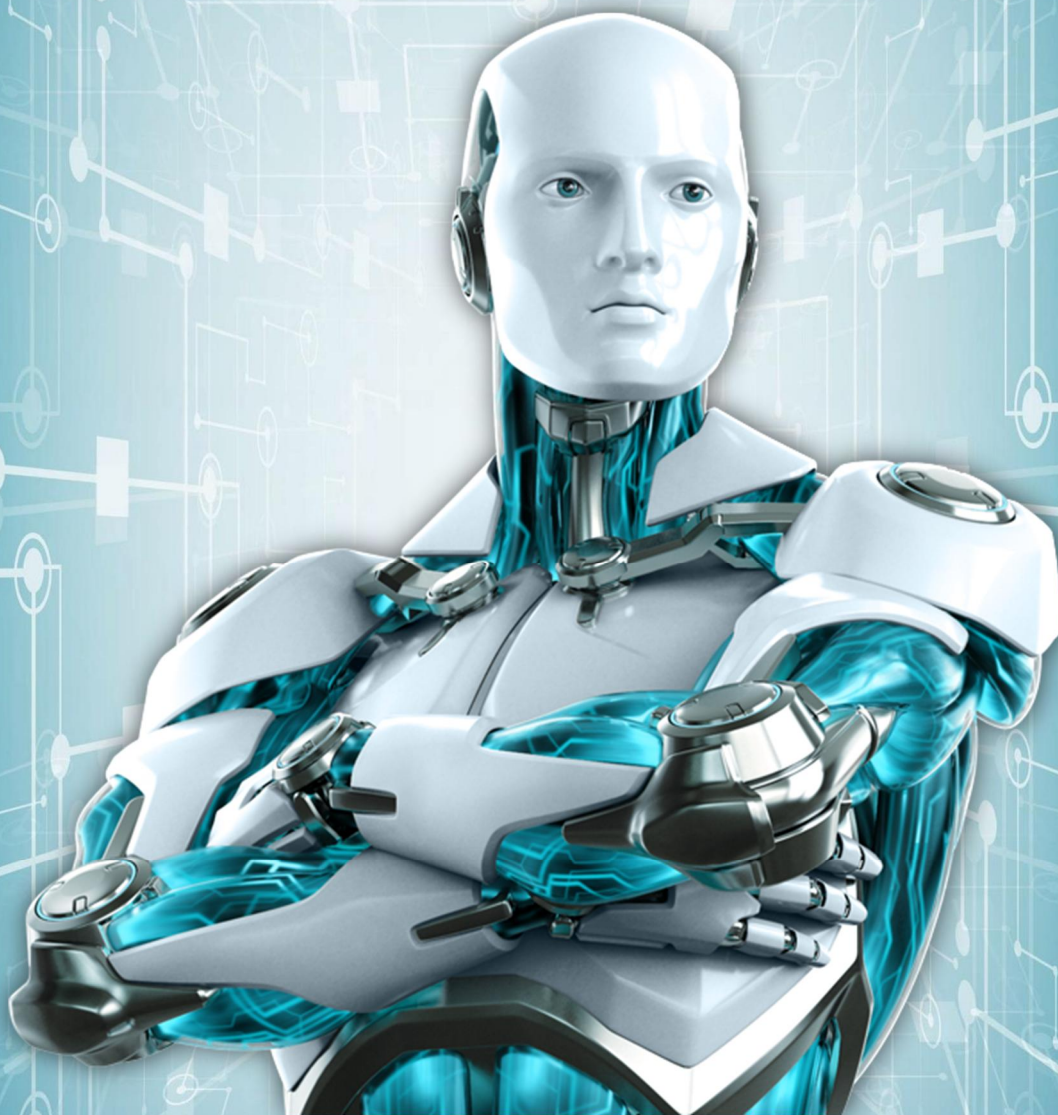


В.С. ТОЛМАЧОВ

ОСВІТНЯ РОБОТОТЕХНІКА ТА АВТОМАТИКА



Навчально-методичний посібник

**Міністерство освіти і науки України
Глухівський національний педагогічний університет
імені Олександра Довженка**

В. Толмачов

ОСВІТНЯ РОБОТОТЕХНІКА ТА АВТОМАТИКА

Навчально-методичний посібник

2023

УДК 004.89(072)

T52

*Рекомендовано Вченою радою
Глухівського національного педагогічного університету
імені Олександра Довженка протокол № 3 від 01.11.2023 року*

Рецензенти:

Луценко Г.В., доктор педагогічних наук, професор Глухівського національного педагогічного університету імені Олександра Довженка.

Шакотько В.В., кандидат педагогічних наук, доцент кафедри початкової освіти, гуманітарних дисциплін та інформатики Обласного коледжу «Кременчуцька гуманітарно-технологічна академія імені А. С. Макаренка» Полтавської обласної ради.

Толмачов В.С.

T52

Толмачов В.С. Освітня робототехніка та автоматика : навчально-методичний посібник. Глухів : Глухівський НПУ ім. О.Довженка, 2023. – 139 с.

ISBN 978-966-376-123-7

Навчально-методичний посібник охоплює теми, передбачені програмою навчальної дисципліни «Освітня робототехніка та автоматика», а також допоможе отримати необхідний базовий теоретичний матеріал, як для виконання лабораторно-практичних робіт, так і під час створення власних електронних пристроїв.

Теоретичний матеріал охоплює тематику роботи з платою Arduino UNO та підключення до неї різних модулів, датчиків і пристроїв керування.

Роботи пропонується виконувати за допомогою онлайн-сервісів Wokwi та Tinkercad. Представлений у рукопису теоретичний матеріал добре проілюстрований прикладами електронних схем і скетчами до них, має актуальний і сучасний стан та націлений на допомогу здобувачу освіти.

Розраховано на викладачів та студентів закладів вищої освіти, закладів фахової передвищої освіти, закладів професійної (професійно-технічної) освіти, а також учителів і учнів старших класів загальноосвітніх шкіл. Практичні роботи та теоретичний матеріал, представлений в навчально-методичному, посібнику можна також використовувати в межах гуртка з робототехніки та в якості довідника.

ЗМІСТ

Лабораторна робота №1 (Арифметичні основи функціонування обчислювальної техніки)	4
Лабораторна робота №2 (Дослідження роботи логічних елементів)	12
Лабораторна робота №3 (<i>Програмні середовища для створення моделей електронних схем</i>)	19
Лабораторна робота №4 (<i>Використання вимірювальних приладів</i>)	25
Лабораторна робота №5 (Підключення світлодіодів до плати Arduino)	35
Лабораторна робота №6 (Введення даних за допомогою кнопок)	39
Лабораторна робота №7 (<i>Підключення потенціометра до плати Arduino</i>)	42
Лабораторна робота №8 (Використання широтно-імпульсної модуляції (ШИМ))	45
Лабораторна робота №9 (Підключення різних елементів відображення інформації до плати Arduino)	51
Лабораторна робота №10 (Підключення до плати Arduino датчиків різного призначення)	76
Лабораторна робота №11 (Підключення до плати Arduino двигунів та сервоприводів)	118
Лабораторна робота №12 (Підключення крокових двигунів до плати Arduino)	127
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	136

Лабораторна робота №1 (Арифметичні основи функціонування обчислювальної техніки)

Тема: Арифметичні основи функціонування обчислювальної техніки.

Мета: Ознайомитись з різними системами числення.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

Арифметичні основи функціонування обчислювальних систем.

1. Системи числення.

Системою числення називають сукупність прийомів запису чисел. Розрізняють позиційні і непозиційні системи числення.

Непозиційні системи числення, прикладом якої є так звані римські цифри. У цій системі вага кожного символу не залежить від місця його розташування. Так запис IV позначає число 4, а запис VIII – позначає число 8.

У позиційній системі числення, на відміну від непозиційної, значення цифри в зображенні числа залежить від її положення (позиції) у послідовності цифр, що зображують число.

Наприклад, запис 7852 у позиційній системі числення можна описати наступним чином: число містить 2 одиниці, 5 десятків, 8 сотен і 7 тисяч, і це можна записати у вигляді виразу $7 \times 10^3 + 8 \times 10^2 + 5 \times 10^1 + 2 \times 10^0$.

Число 10 у цьому виразі називають основою системи числення, а систему в якій записано число – десятковою системою числення.

В десятковій системі для запису числа використовують числа 0,1,2,3,4,5,6,7,8,9, які називають алфавітом системи числення.

Для того, щоб вказати, що число записане у десятковій системі, його можна записати у такому вигляді: $(7852)_{10}$.

Але можна використовувати будь-яке натуральне число в якості основи., Наприклад, якщо узяти за основу число 7, то запис буде мати наступний вигляд $(321)_7$ буде означати вираз $3 \times 7^2 + 2 \times 7^1 + 1 \times 7^0$. Якщо порахувати, ми отримаємо число $147 + 14 + 1 = 162$.

Тобто $(321)_7 = (162)_{10}$.

У сімковій системі для запису чисел використовуються 7 цифр: 0,1,2,3,4,5,6.

В будь-якій системі числення число, рівне основі системи, має вигляд "10". Це означає, що множення або ділення на основу системи зводиться до перенесення коми, що розділяє цілу і дробову частину, на одну позицію праворуч (у випадку множення) або ліворуч (у випадку ділення):

$$(25)_7 \times (7)_{10} = (25)_7 \times (10)_7 = (250)_7.$$

З числами у будь-якій системі числення всі арифметичні операції виконуються за тими ж правилами, що і в десятковій системі.

2. Шістнадцяткова система числення

В комп'ютерній техніці часто використовується шістнадцяткова система числення. Ця система повинна мати 16 символів для позначення цифр. Алфавіт цієї системи має такий вид: перші десять цифр запозичуємо з десяткової

системи числення, а решта, це великі латинські літери: 10 - А, 11 - В, 12 - С, 13 - D, 14 - Е, 15 - F.

Тобто 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, А, В, С, D, Е, F.

Таким чином запис $(A2F)_{16}$ буде означати вираз $10 \times 16^2 + 2 \times 16^1 + 15 \times 16^0 = (2607)_{10}$.

Окрім шістнадцяткової системи в комп'ютерній техніці використовуються двійкова, а також вісімкова системи числення. Відповідно мають за основу числа 2 і 8.

Алфавіт двійкової системи складається з двох цифр: 0, 1.

Запис вигляду $(11011011)_2$ можна записати у вигляді виразу: $1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (219)_{10}$.

У представленій таблиці 1 подані перші шістнадцять цілих чисел, записаних у різних системах числення.

Таблиця 1

Десяткова система	Шістнадцяткова система	Вісімкова система	Двійкова система
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	8	10	1000
9	9	11	1001
10	A	12	1010
11	B	13	1011
12	C	14	1100
13	D	15	1101
14	E	16	1110
15	F	17	1111

Для запису чисел в таблиці у двійковій системі знадобилося максимум чотири біта.

Дріб у двійковій системі записується за тими ж правилами, що і дріб у десятковій системі числення, але під час підрахунку треба використовувати від'ємні степені двійки.

Запис $(0,1001)_2$ означає $1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = 1 \times 0,5 + 0 \times 0,25 + 0 \times 0,125 + 1 \times 0,0625 = (0,5625)_{10}$.

3. Перехід від десяткової до інших систем числення

Виникає питання, як записати подане десяткове число у будь-якій довільній системі числення.

Для цього, якщо число ціле, то його потрібно послідовно ділити на основу системи доти, доки частка не стане меншою ніж основа. Залишки, що будуть отримані у процесі ділення записані у зворотному порядку, починаючи з останньої частки.

Наприклад.

Записати число $(19875)_{10}$ у шістнадцятковій системі.

Для перетворення числа 19875 з десяткової системи числення в шістнадцяткову, ви можете використовувати метод послідовних ділень і ділення на 16.

Ділимо число 19875 на 16:

$$19875 \div 16 = 1242 \text{ з залишком } \mathbf{3}$$

Далі ділимо 1242 на 16:

$$1242 \div 16 = 77 \text{ з залишком } \mathbf{10}$$

Повторюємо ділення:

$$77 \div 16 = \mathbf{4} \text{ з залишком } \mathbf{13}$$

Тепер ми маємо всі залишки з ділень від останнього до першого: 4, 13, 10, 3.

Замінюємо числа, на відповідні шістнадцяткові символи:

4 – **4**

13 – **D**

10 – **A**

3 – **3**

Отже, число 19875 у шістнадцятковій системі записується як "4DA3", тобто $(4DA3)_{16}$.

Записати число $(497)_{10}$ у двійковій системі числення.

Щоб перевести число 497 з десяткової системи числення в двійкову, ви можете використовувати метод послідовних ділень і ділення на 2.

Ділимо число 497 на 2:

$$497 \div 2 = 248 \text{ з залишком } \mathbf{1}$$

Далі ділимо 248 на 2:

$$248 \div 2 = 124 \text{ з залишком } \mathbf{0}$$

Повторюємо ділення:

$$124 \div 2 = 62 \text{ з залишком } \mathbf{0}$$

І ще раз:

$$62 \div 2 = 31 \text{ з залишком } \mathbf{0}$$

Продовжуємо:

$$31 \div 2 = 15 \text{ з залишком } \mathbf{1}$$

Продовжуємо:

$$15 \div 2 = 7 \text{ з залишком } \mathbf{1}$$

Ділими ще раз:

$$7 \div 2 = 3 \text{ з залишком } \mathbf{1}$$

І останнє ділення:

$$3 \div 2 = \mathbf{1} \text{ з залишком } \mathbf{1}$$

Тепер ми маємо всі залишки з ділень від останнього до першого: 1,1,1,1,1, 0, 0, 0, 1.

Це і є двійкове представлення числа 497. Тепер об'єднайте всі ці біти в оберненому порядку, щоб отримати представлення у двійковій системі:

$$\mathbf{111110001}_{(2)}$$

Отже, число 497 у двійковій системі числення записується як "111110001", $(111110001)_2$, або $111110001_{(2)}$

Щоб конвертувати десятковий дріб в іншу систему, треба застосовувати дещо інший алгоритм.

Поданий дріб треба ділити на число обернене основі системи. Сутність алгоритму відображаємо на прикладі.

Записати число $(0,258)_{10}$ у двійковій системі числення.

Для переведення десяткового дробового числа 0,258 у двійкову систему числення, ви можете використовувати метод множення на 2 та виділення цілих та дробових частин.

Почнемо з цілої частини: 0

Потім перейдемо до дробової частини: 0,258

Тепер ми будемо множити дробову частину на 2 і виділяти цілу частину:

$$0,258 * 2 = 0,516$$

Беремо 0 як перший біт дробової частини.

$$0,516 * 2 = 1,032$$

Беремо 1 як наступний біт дробової частини.

$$0,032 * 2 = 0,064$$

Беремо 0 як ще один біт.

$$0,064 * 2 = 0,128$$

Беремо 0 як ще один біт.

$$0,128 * 2 = 0,256$$

Беремо 0 як ще один біт.

$$0,256 * 2 = 0,512$$

Беремо 0 як ще один біт.

$$0,512 * 2 = 1,024$$

Беремо 1 як ще один біт.

Таким чином, у двійковій системі 0,258 представляється як 0,01000010.

Отже, 0,258 у двійковій системі записується як 0,01000010.

Якщо число не є кратним степеню 0,5, то двійковий дріб буде нескінченим і конвертація у двійкову систему можна здійснити тільки приблизно.

Для переведення десяткового дробового числа 0,257 в двійкову систему числення, використовуємо попередній алгоритм дій:

Почнемо з цілої частини: 0

Потім перейдемо до дробової частини: 0,257

Тепер ми будемо множити дробову частину на 2 і виділяти цілу частину:
 $0,257 * 2 = 0,514$

Беремо 0 як перший біт дробової частини.

$0,514 * 2 = 1,028$

Беремо 1 як наступний біт дробової частини.

$0,028 * 2 = 0,056$

Беремо 0 як ще один біт.

$0,056 * 2 = 0,112$

Беремо 0 як ще один біт.

$0,112 * 2 = 0,224$

Беремо 0 як ще один біт.

$0,224 * 2 = 0,448$

Беремо 0 як ще один біт.

$0,448 * 2 = 0,896$

Беремо 0 як ще один біт.

Таким чином, у двійковій системі 0,257 представляється як 0,010000010111... (з повторюваним відомим блоком 0001).

Отже, 0,257 у двійковій системі має безкінечний період із повторюваними цифрами 0001.

4 Перехід від двійкової до вісімкової і шістнадцяткової системи

Для переводу цілого числа з двійкової системи до вісімкової його попередньо треба розбити на тріади, а потім кожен тріаду замінити відповідним цифрою вісімкової системи:

(1110101010111) ₂				
1	110	101	010	111
1	6	5	2	7
(16527) ₈				

Щоб перейти до шістнадцяткової системи треба зробити так само, тільки поділяти потрібно на тетради, і замінювати їх шістнадцятковою цифрою:

$(1110101010111)_2$			
1	1101	0101	0111
1	13	5	7
$(1D57)_{16}$			

Діючи у зворотному порядку, можна переходити від вісімкової і шістнадцяткової системи до двійкової. Такий перехід тут є можливим тільки завдяки тому, що основи цих систем мають спільне кратне 2.

5 Змішаний двійково-десятковий запис чисел

В комп'ютерній техніці іноді використовується двійково-десяткова система представлення чисел, коли кожен розряд десятичного числа кодується відповідним двійковим числом, наприклад, для числа 2357:

2	3	5	7
0010	0011	0101	0111
0010001101010111			

6. Використання двійкової системи числення в комп'ютерній техніці.

Будь-яка обчислювальна машина являє собою технічний пристрій, представлення чисел в якому відбувається з використанням двійкової системи числення. Алфавіт двійкової системи має тільки дві цифри – 0 та 1, а це значить, що біт можна моделювати за допомогою фізичного пристрою, який може знаходитися у двох стійких станах «увімкнуто» і «вимкнено».

ПОРЯДОК ВИКОНАННЯ РОБОТИ.

1. Запишіть подані числа у двійковій системі числення

$34_{(10)} =$	$15_{(10)} =$	$63_{(10)} =$
$18_{(10)} =$	$124_{(10)} =$	$7_{(10)} =$
$8,125_{(10)} =$	$1,34_{(10)} =$	$3,5_{(10)} =$
$14_{(16)} =$	$A_{(16)} =$	$FE_{(16)} =$
$10_{(8)} =$	$24_{(8)} =$	$7_{(8)} =$
$34_{(10)} =$	$29_{(10)} =$	$65_{(10)} =$
$18_{(10)} =$	$129_{(10)} =$	$7_{(10)} =$
$8,225_{(10)} =$	$1,14_{(10)} =$	$3,57_{(10)} =$
$13_{(16)} =$	$A0_{(16)} =$	$2FE_{(16)} =$
$11_{(8)} =$	$22_{(8)} =$	$7_{(8)} =$

2. Запишіть подані числа у десятковій системі числення

$1000111_{(2)} =$	$111111_{(2)} =$	$10001_{(2)} =$
$10111_{(2)} =$	$111_{(2)} =$	$11_{(2)} =$
$FF_{(16)} =$	$10_{(16)} =$	$B2_{(16)} =$
$AA_{(16)} =$	$1C_{(16)} =$	$D2_{(16)} =$
$15_{(8)} =$	$41_{(8)} =$	$10_{(8)} =$
$10100111_{(2)} =$	$1101111_{(2)} =$	$100001_{(2)} =$
$101011_{(2)} =$	$1011_{(2)} =$	$101_{(2)} =$
$FF_{(16)} =$	$10_{(16)} =$	$B2_{(16)} =$
$AA_{(16)} =$	$1B_{(16)} =$	$D3_{(16)} =$
$105_{(8)} =$	$41_{(8)} =$	$10_{(8)} =$

3. Запишіть подані числа у шістнадцятковій системі числення

$1000111_{(2)} =$	$111111_{(2)} =$	$10001_{(2)} =$
$10111_{(2)} =$	$111_{(2)} =$	$11_{(2)} =$
$10_{(10)} =$	$5_{(10)} =$	$12_{(10)} =$
$15_{(10)} =$	$20_{(10)} =$	$A2_{(16)} =$
$15_{(8)} =$	$41_{(8)} =$	$10_{(8)} =$
$10100111_{(2)} =$	$111000111_{(2)} =$	$1001101_{(2)} =$
$1010011_{(2)} =$	$1101_{(2)} =$	$110_{(2)} =$
$10_{(10)} =$	$3_{(10)} =$	$12_{(10)} =$
$15_{(10)} =$	$20_{(10)} =$	$C8_{(16)} =$
$15_{(8)} =$	$41_{(8)} =$	$10_{(8)} =$

4. Виконайте арифметичні операції:

$7_{(8)}+4_{(8)}=?$	$3_{(8)}+5_{(8)}=?$	$15_{(8)}+3_{(8)}=?$
$10_{(16)}+A1_{(16)}=?$	$B2_{(16)} - A1_{(16)}=?$	$E0_{(16)}+D_{(16)}=?$
$100010_{(2)}+10111_{(2)}=?$	$101010_{(2)}+111100_{(2)}=?$	$1000010_{(2)} - 10111_{(2)}=?$
$10010_{(2)} - 111_{(2)}=?$	$10010_{(2)} \times 111_{(2)}=?$	$10011_{(2)} \times 101_{(2)}=?$
$10010_{(2)} / 10_{(2)}=?$	$1000_{(2)} / 100_{(2)}=?$	$10100_{(2)} / 101_{(2)}=?$
$7_{(8)}+4_{(8)}=?$	$3_{(8)}+5_{(8)}=?$	$15_{(8)}+3_{(8)}=?$
$10_{(16)}+A1_{(16)}=?$	$B2_{(16)} - A1_{(16)}=?$	$E0_{(16)}+D_{(16)}=?$
$1010010_{(2)}+10111_{(2)}=?$	$101010_{(2)}+111100_{(2)}=?$	$10010010_{(2)} - 10111_{(2)}=?$
$10010_{(2)} - 111_{(2)}=?$	$10010_{(2)} \times 111_{(2)}=?$	$10011_{(2)} \times 1010_{(2)}=?$
$10010_{(2)} / 10_{(2)}=?$	$10000_{(2)} / 100_{(2)}=?$	$10100_{(2)} / 100_{(2)}=?$

КОНТРОЛЬНІ ЗАПИТАННЯ І ЗАВДАННЯ

1. Що таке система числення і які бувають системи числення?
2. Чим позиційної система числення відрізняється від непозиційна?
3. Запишіть числа $(423)_5$, $(102)_3$, $(21523)_8$, $(11101111)_2$, $(BF3A)_{16}$, у десятковій системі числення.
4. Конвертуйте число $(523)_{10}$ у двійкову і вісімкову системи числення.
5. Конвертуйте числа 0,25 і 0,56 у двійкову систему числення.

6. Виконати дії над числами, заданими у вісімковій системі числення:
- $(104)_8 + (301)_8$;
 - $(521)_8 - (223)_8$;
 - $(2311)_8 \times (2)_8$;
 - $(2150)_8 \times (4)_8$.
9. Виконати дії:
- $(10111)_2 + (100011)_2$,
 - $(10110)_2 \times (101)_2$,
 - $(11101101)_2 - (11001)_2$.
10. Які з чисел записані неправильно і чому?
 $(12131)_3$, $(121459)_8$, $(1)_2$, $(10111)_{16}$
11. Число $(10111011001111)_2$ помножити на 4.
12. Запишіть число $(10111011001111)_2$ у шістнадцятковій і вісімковій системі числення.
13. Запишіть числа $(5B0C)_{16}$, $(1756)_8$ у двійковій системі числення.

Лабораторна робота №2 (Дослідження роботи логічних елементів)

Тема: Дослідження роботи логічних елементів.

Мета: Ознайомитись з різними базовими логічними елементами.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Базові елементи цифрової логіки

(логічні елементи «AND», «OR», «NOT» та інші)

Логічний елемент – це пристрій, спроектований для обробки цифрової інформації, яка представлена у різних системах логічних рівнів, таких як двійкова логіка (послідовність "1" та "0"), трійкова логіка (послідовність "0", "1" та "2"), або десяткова логіка (послідовність "0", "1", "2", "3", "4", "5", "6", "7", "8" та "9").

Логічні елементи можуть мати різні фізичні реалізації, включаючи механічні, електромеханічні (з використанням електромагнітних реле), електронні (за допомогою діодів і транзисторів), пневматичні, гідравлічні, оптичні та інші методи.

З розвитком електротехніки від механічних логічних елементів поступово перейшли до електромеханічних логічних елементів, а потім до електронних логічних елементів, які використовували електронні лампи, а пізніше – транзистори.

Після підтвердження в 1946 році теореми Джона фон Неймана про економічність позиційних систем числення, стало відомо про переваги двійкової та трійкової систем числення у порівнянні з десятковою системою числення. Внаслідок цього, від логічних елементів, які працювали у десятковій системі, почали переходити на логічні елементи, які базувалися на двійковій системі числення. Використання двійкової та трійкової систем дозволило суттєво зменшити кількість операцій і компонентів, необхідних для обробки інформації, у порівнянні з десятковими логічними елементами.

Логічні елементи здійснюють логічні операції над вхідними сигналами (операндами або даними).

Для представлення чисел у двійковій системі використовуються всього дві цифри – нуль (0) і одиниця (1). Двійкові цифри можна легко виразити як два різних стани напруги - високий і низький. У логічних схемах і цифрових системах існують два можливі стани – 1 і 0. Стан 1 називається високим, щоб показати, що напруга в цьому стані вища, ніж у стані 0. Стан 0 називається низьким, щоб показати, що напруга в цьому стані нижча, ніж у стані 1.

Усе цифрове обладнання побудоване з використанням обмеженої кількості основних схем, які називаються логічними елементами. Ці логічні елементи виконують певні логічні функції з двійковими даними.

Логічні схеми можна розділити на два основних типи: схеми, призначені для прийняття рішень, і схеми пам'яті. Логічні схеми прийняття рішень визначаються станом вхідних сигналів і виробляють вихідний сигнал на основі цих станів та логіки схеми. Схеми пам'яті використовуються для зберігання

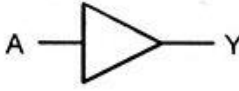
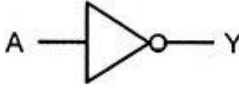
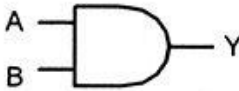
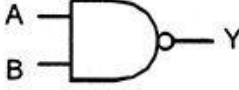
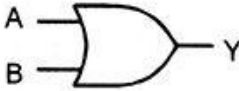



двійкових даних і можуть зберігати інформацію, яка доступна для подальшого використання.

Елемент І (AND), також відомий як логічний "І", є логічною схемою, на виході якої з'являється 1 лише в тому випадку, коли всі його входи приймають сигнал 1. Якщо на будь-який з входів подається сигнал 0, то на виході з'являється сигнал 0. Елемент І виконує операцію логічного множення, і це означає, що він видасть результат 1 лише тоді, коли всі вхідні значення рівні 1, в інших випадках він видасть 0.

Елемент АБО (OR), також відомий як логічний "АБО", є логічною схемою, на виході якої з'являється 1, якщо хоча б один із його входів має сигнал 1. На виході з'являється 0 лише в тому випадку, коли всі входи мають сигнал 0. Елемент АБО може мати два або більше входів. Він виконує логічну операцію додавання, оскільки видає результат 1, якщо принаймні один із входів має значення 1.

Елемент НІ (NOT) виконує операцію, яка називається інверсією. Основною метою інвертора є зміна стану на виході на протилежний стан входу. Якщо на вхід інвертора подається високий стан, тобто 1, то на виході з'явиться низький стан, тобто 0. І навпаки, якщо на вхід інвертора подається низький стан, тобто 0, то на виході з'явиться високий стан, тобто 1.

Елемент І-НІ (NAND) є комбінацією елементів І та НІ і є одним з найбільш широко використовуваних логічних елементів. Це через те, що ці елементи можуть бути використані для створення різних інших логічних функцій. Вихід елемента І-НІ є логічним запереченням виходу елемента І. Якщо на будь-який з входів подається 0, то на виході з'являється 1. Елемент АБО-НІ (NOR) є комбінацією елемента АБО та інвертора. Подібно до елемента І-НІ, елемент АБО-НІ також може бути використаний для створення різних логічних функцій. Вихід елемента АБО-НІ також є логічним запереченням виходу елемента АБО. На виході з'являється 1 лише тоді, коли на обох входах подається 0. Ці елементи описуються за допомогою таблиць істинності для розуміння їхньої роботи.

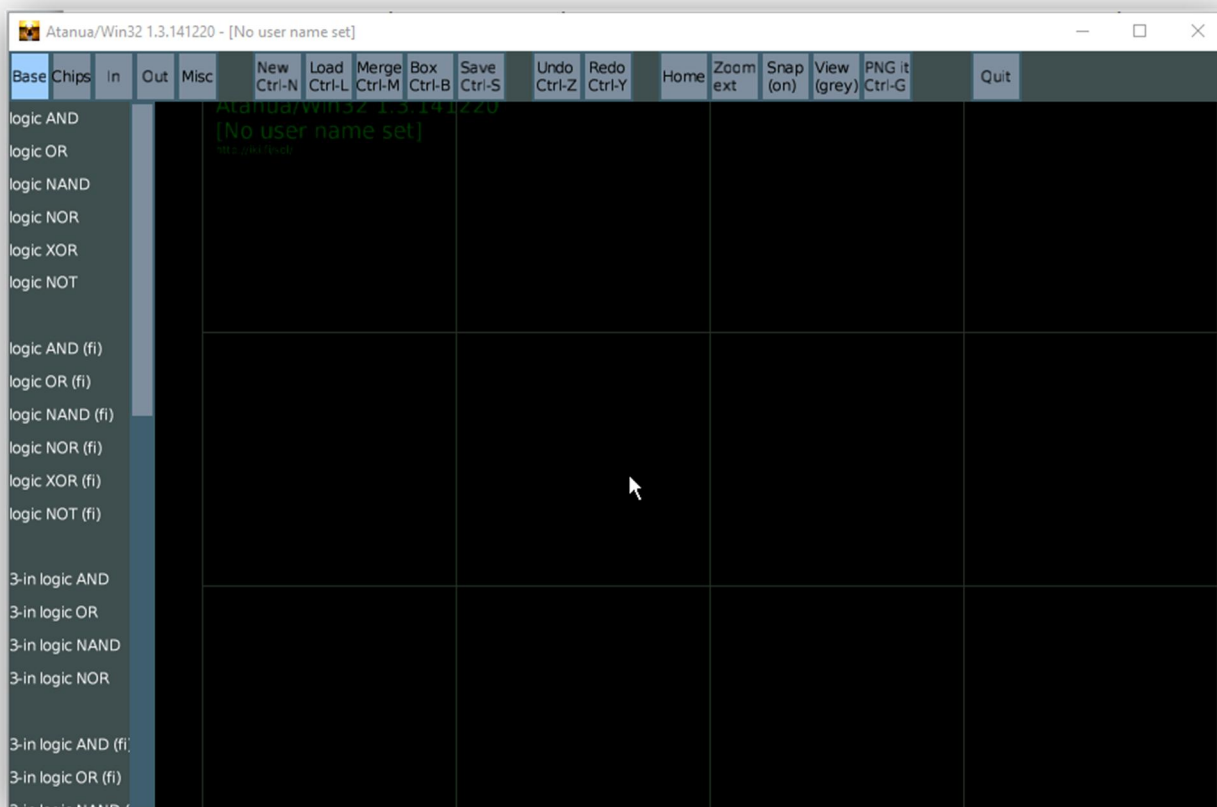
Logic function	Logic symbol	Truth table	Boolean expression															
Buffer		<table border="1" data-bbox="893 280 1021 392"> <thead> <tr><th>A</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </tbody> </table>	A	Y	0	0	1	1	$Y = A$									
A	Y																	
0	0																	
1	1																	
Inverter (NOT gate)		<table border="1" data-bbox="893 448 1021 560"> <thead> <tr><th>A</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	Y	0	1	1	0	$Y = \bar{A}$									
A	Y																	
0	1																	
1	0																	
2-input AND gate		<table border="1" data-bbox="869 593 1045 750"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	$Y = A \cdot B$
A	B	Y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
2-input NAND gate		<table border="1" data-bbox="869 795 1045 952"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	$Y = \overline{A \cdot B}$
A	B	Y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
2-input OR gate		<table border="1" data-bbox="869 996 1045 1153"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	$Y = A + B$
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
2-input NOR gate		<table border="1" data-bbox="869 1198 1045 1355"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	$Y = \overline{A + B}$
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
2-input EX-OR gate		<table border="1" data-bbox="869 1400 1045 1556"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	$Y = A \oplus B$
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
2-input EX-NOR gate		<table border="1" data-bbox="869 1601 1045 1758"> <thead> <tr><th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1	$Y = \overline{A \oplus B}$
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Логічні функції	Умовні позначення
Кон'юнкція "І" $y = x_1 \cdot x_2$	
Диз'юнкція "АБО": $y = x_1 + x_2$	
Інверсія "НІ": $y = \bar{x}$	
Заперечення диз'юнкції (Пірса) "АБО-НІ": $y = \overline{x_1 + x_2}$	
Заперечення кон'юнкції "І-НІ": (Шеффера) $y = \overline{x_1 \cdot x_2}$	

Логічна операція	Позначення логічного елемента		Таблиця істинності		
	країни СНД	Європа й США			
І (AND)			X ₁	X ₂	Y
			0	0	0
			0	1	0
			1	0	0
АБО (OR)			X ₁	X ₂	Y
			0	0	0
			0	1	1
			1	0	1
НІ (NOT)			X	Y	
			0	1	
			1	0	
І-НІ (NAND)			X ₁	X ₂	Y
			0	0	1
			0	1	1
			1	0	1
АБО-НІ (NOR)			X ₁	X ₂	Y
			0	0	1
			0	1	0
			1	0	0

ПОРЯДОК ВИКОНАННЯ РОБОТИ.

1. Завантажте програму **Atanua** (<http://sol.gfxile.net/atanua/>).
2. Ознайомтесь з інтерфейсом програми **Atanua**.

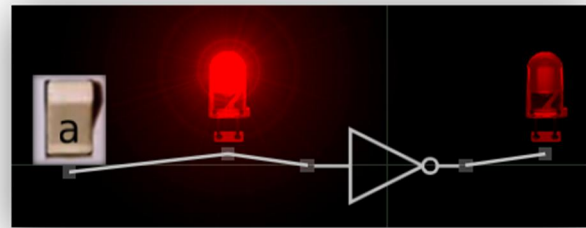
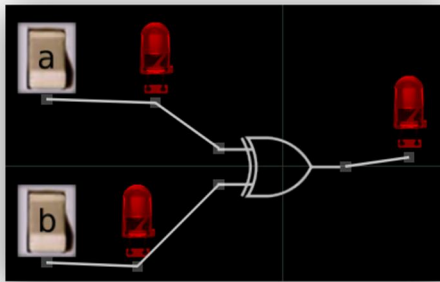
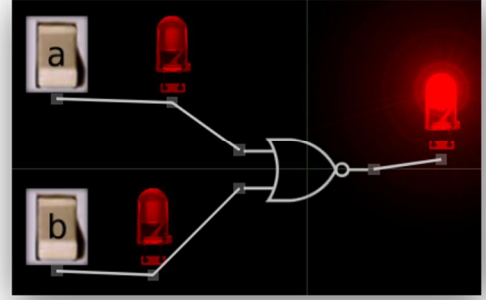
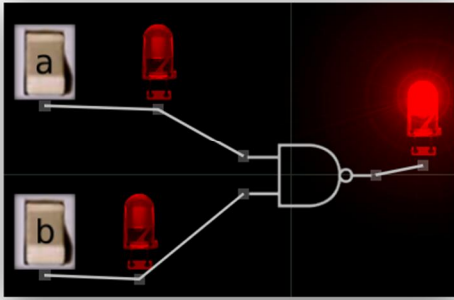
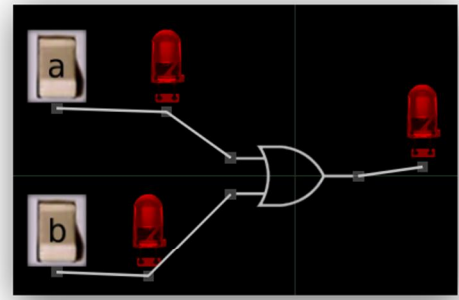
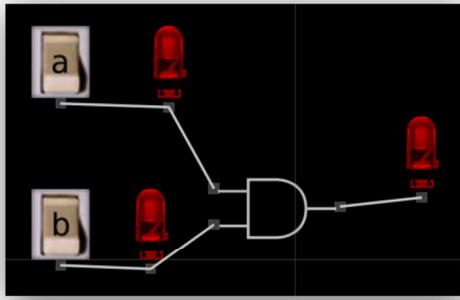


3. Зробіть знімок екрана (*англ. screenshot, скриншот*) і вставте його у звіт.
4. У вигляді таблиці опишіть призначення кнопок основного верхнього меню:



5. Використовуючи елементи з меню **Base**, **In** та **Out** складіть електричні схеми з усіма базовими логічними елементами та перевірте їх працездатність:





6. Складіть схеми подібні до попередніх, використовуючи наступні елементи:

- 3-in logic AND
- 3-in logic OR
- 3-in logic NAND
- 3-in logic NOR

До кожного з елементів побудуйте таблицю істинності.

7. Складіть подібні схеми і з'ясуйте як працюють елементи:



SR-latch (nor)
SR-latch (nand)
Synch. SR-latch
D-latch

ПИТАННЯ ДЛЯ ПІДГОТОВКИ

1. Які функції виконують логічні елементи?
2. Що означають «1» та «0» в електроніці?
3. Які основні типи логічних схем існують?
4. Які логічні операції виконують елемент І (AND), АБО (OR), НІ (NOT), НІ-І (NAND), НІ-АБО (NOR)?
5. Яку функцію виконує тригер і які існують тригери?

Лабораторна робота №3 (Програмні середовища для створення моделей електронних схем)

Тема: Програмні середовища для створення моделей електронних схем.

Мета: Ознайомитись з роботою в різних програмних середовищах для побудови електронних схем, емуляції електронних схем і написання програмного коду для Arduino-сумісних платформ.

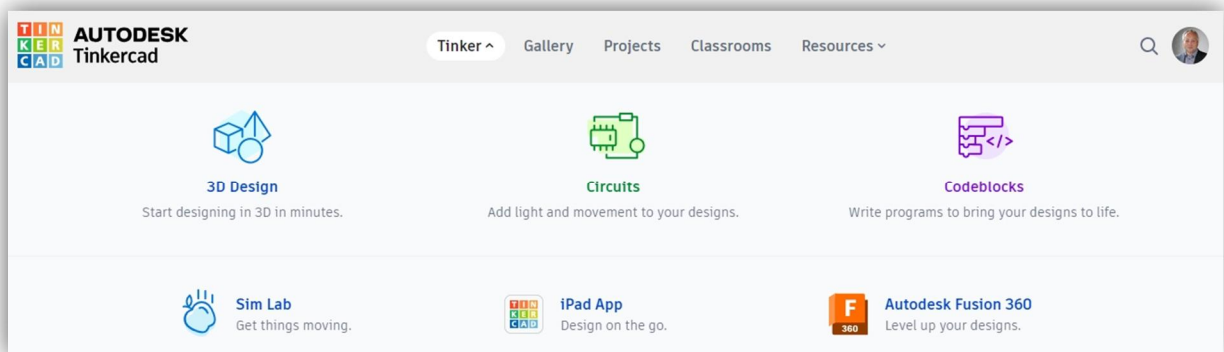
ТЕОРЕТИЧНІ ВІДОМОСТІ

Плати Arduino у всьому світі використовують для роботи з електронікою, робототехнікою та автоматикою. Такі плати можна збирати самостійно або придбати готові. Задумані ідеї можна реалізувати просто та швидко, використовуючи зручну мову програмування.

Якщо тільки почали займатися схемотехнікою чи просто є необхідність потренуватися у створенні того чи іншого пристрою, купувати купу необхідних плат, модулів та радіодеталей непрактично.

Створювати Arduino проекти без покупки самого конструктора можна через програмні симулятори, наприклад, онлайн-сервіси **Tinkercad** (<https://www.tinkercad.com>) або **Wokwi** (<https://wokwi.com/>).

Tinkercad – це безкоштовна онлайн-сервіс для створення електронних схем та 3D-моделювання, який працює у веб-браузері. З 2011 року Tinkercad став більш доступним і популярним сервісом для STEM навчання.



TinkerCAD можна використовувати навіть для тих, хто тільки починає вивчати 3D-моделювання та підготовку об'єктів для 3D-друку. У редакторі існує зручна бібліотека готових компонентів, що спрощує процес створення моделей. Для новачків доступні онлайн-підручники. Що ще важливо, всі інструменти цього сервісу безкоштовні.

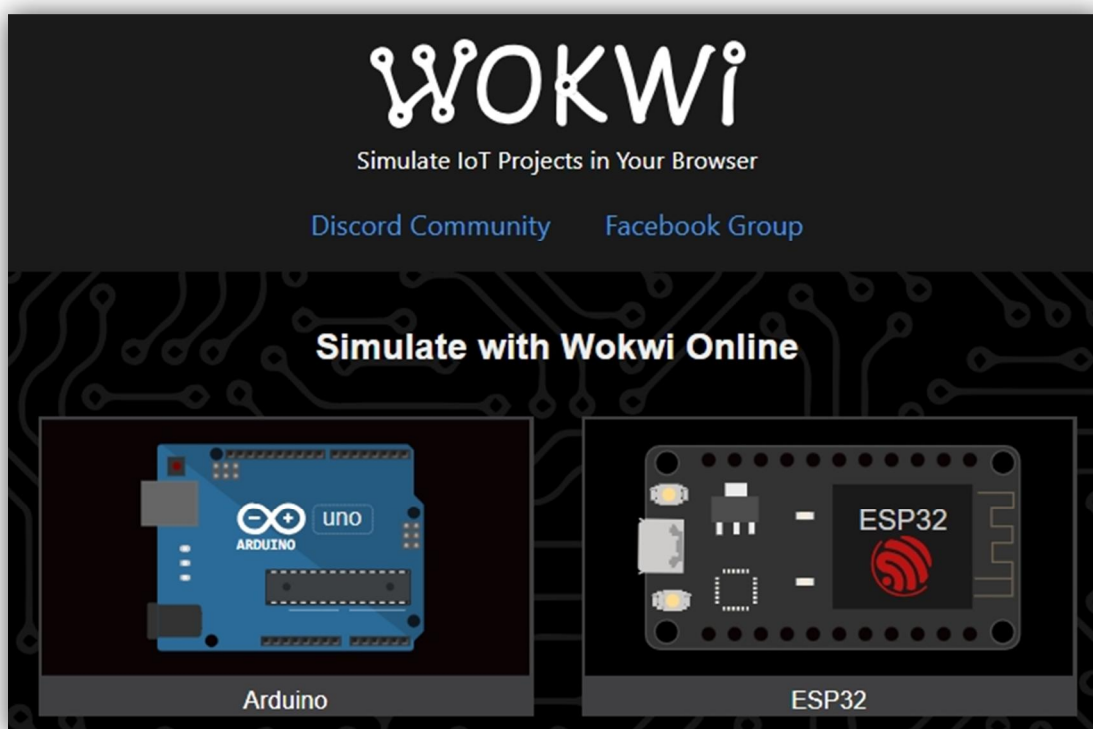
Для користування TinkerCAD необхідно створити обліковий запис Autodesk. Середовище для моделювання електронних схем включає наступні компоненти:

- Редактор електронних схем.
- Емулятор роботи основних електронних компонентів.
- Емулятор контролера Arduino.
- Редактор скетчів, включаючи візуальний редактор.

- Система налагодження та симулювання проектів з використанням Arduino.

Можливості TinkerCAD можуть бути використані для створення онлайн-курсів у галузі 3D-прототипування, основ електроніки та робототехніки. Вчителі мають можливість створити віртуальний клас і запросити учнів для спільної роботи та навчання.

Ще одним онлайн-симулятором для моделювання електронних схем з використанням Arduino є Wokwi, який підтримує багатофайлові проекти. Для початку роботи з **Wokwi Arduino Simulator**, потрібно перейти на веб-сайт wokwi.com та зареєструватися.



Wokwi виглядає як чудова платформа для навчання з підтримкою кількох плат Arduino (Uno, Mega, Nano), Raspberry Pi Pico та універсальної плати ESP32.

Wokwi можна використовувати безкоштовно, але є також програма преміум-членства під назвою «Wokwi Club», яка коштує 7 доларів на місяць або 67 доларів на рік і дозволяє користувачам додавати бібліотеки Arduino, завантажувати двійкові файли, підключатися до Wi-Fi ESP32 і зберігати свої проекти у приватному порядку. Учасники можуть також проголосувати за нові функції Wokwi.

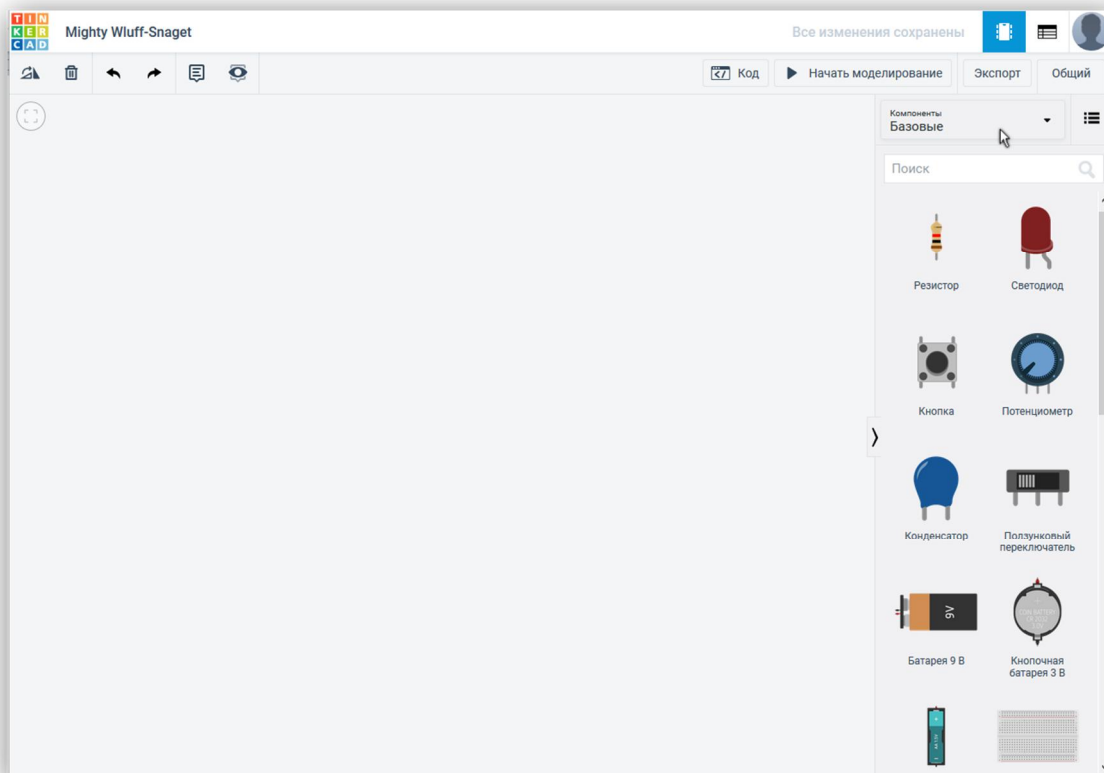
ПОРЯДОК ВИКОНАННЯ РОБОТИ.

1. Зареєструйтеся на онлайн-сервісі **tinkercad.com** (<https://www.tinkercad.com>), або приєднайтеся до наявного класу. Даний онлайн сервіс дозволяє створювати електричні схеми різного рівня складності і

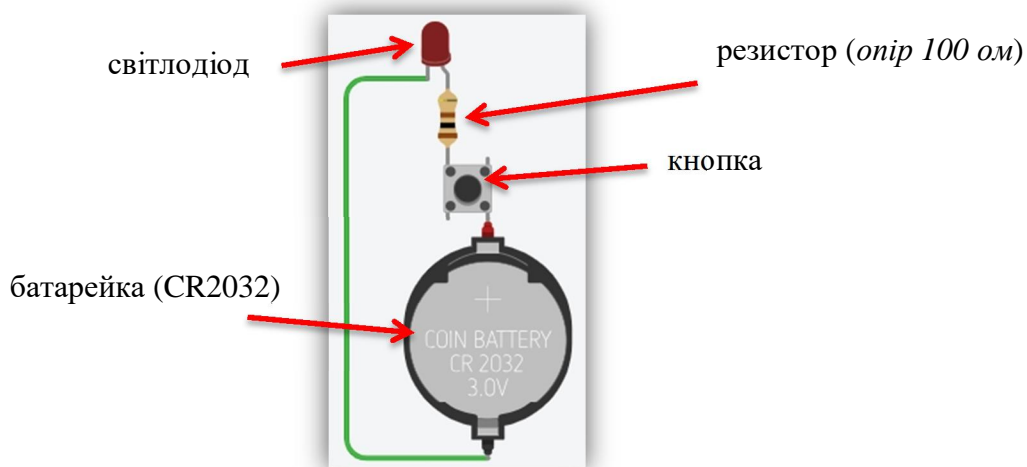
перевіряти їх працездатність. В схемах можна використовувати базові елементи, плату Arduino Uno, додаткові модулі, а також створювати і перевіряти в роботі програмне забезпечення (скетч) для Arduino.


2. У вертикальному меню з лівого боку оберіть пункт **Circuits** і створіть нову електричну схему. Ознайомтесь з інтерфейсом даного сервісу.

3. З правого боку виберіть БАЗОВІ елементи. Зробіть скріншот кожного базового елемента і помістіть його у таблицю. В таблиці треба вказати назву і призначення кожного елемента.

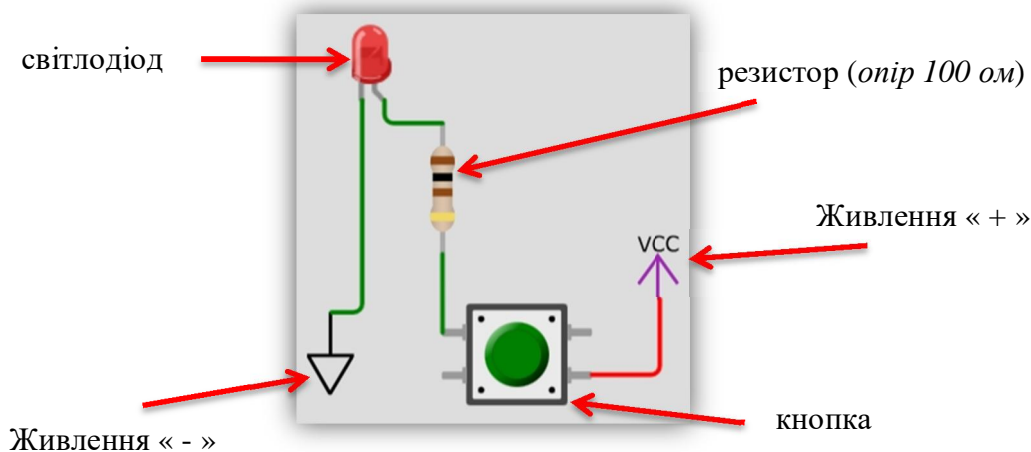


4. Зберіть запропоновану схему:



5. Запустіть симуляцію схеми  і перевірте її працездатність. При натисканні на кнопку світлодіод повинен загорятися. Необхідні скріншоти додайте у свій звіт до лабораторної роботи.

6. Зареєструйтеся на онлайн-сервісі **Wokwi** (<https://wokwi.com/>).
7. Створіть новий проєкт і зберіть наступну схему:



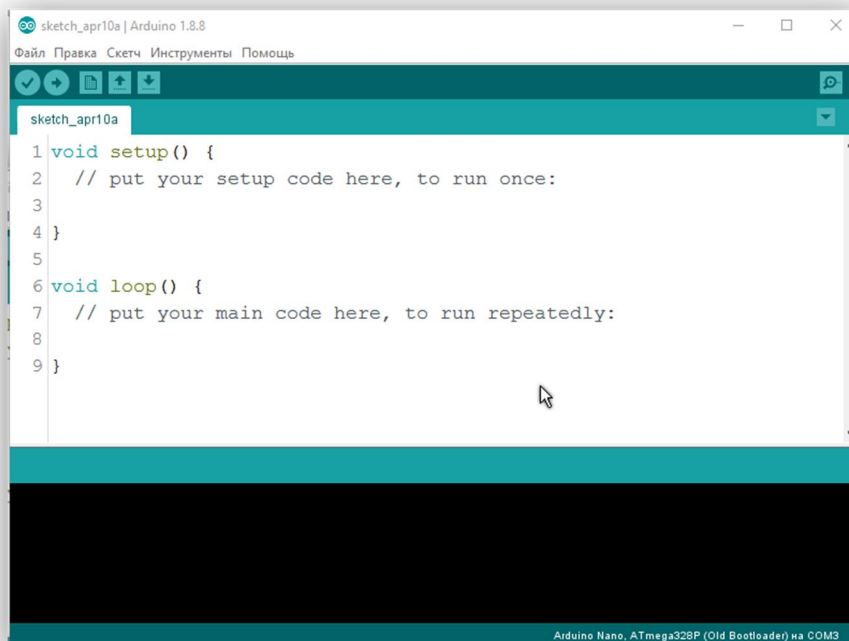
8. Запустіть симуляцію схеми і перевірте її працездатність.
9. Перейдіть на сайт <https://www.arduino.cc/en/Main/Software> для завантаження Arduino IDE.



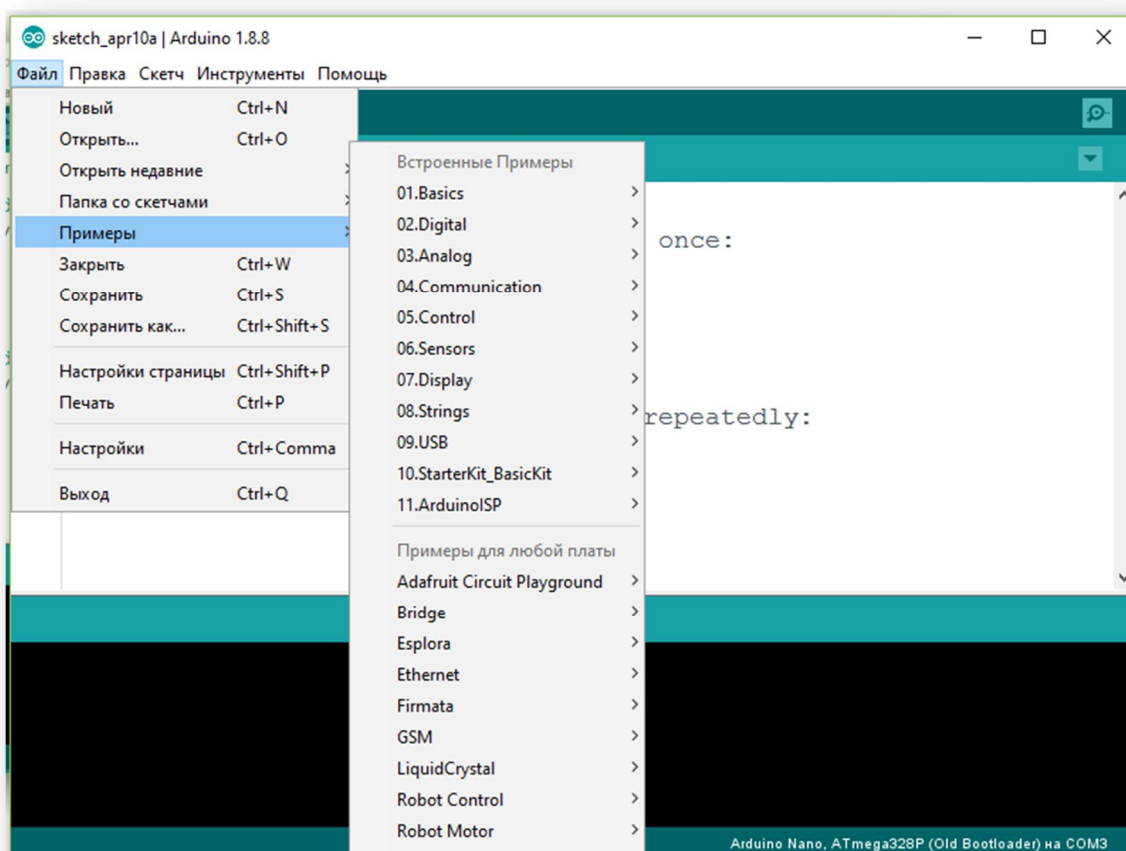
10. Виберіть той варіант Arduino IDE, що відповідає вашій операційній системі. Скачайте його і встановіть. Arduino IDE це середовище програмування плат Arduino, яке дозволяє створювати скетчі різного рівня складності,

перевіряти їх на наявність помилок і програмувати мікроконтролер на платі Arduino.

11. Запустіть Arduino IDE і ознайомтесь з інтерфейсом, опишіть призначення його елементів:



12. Для створення власного скетчу, як шаблони можна використовувати наявні приклади.



13. Для початку роботи з платою Arduino Uno або іншою, спершу підключіть її до USB-порту вашого комп'ютера. Якщо необхідно, встановіть відповідні драйвери. У меню "Інструменти" оберіть тип плати, наприклад, "Arduino Uno", та виберіть правильний послідовний порт. Номер послідовного порта можна знайти у Диспетчері пристроїв. Для прошивки плати, натисніть кнопку "Завантажити" на панелі інструментів або оберіть пункт "Завантажити" з меню "Скетч". Після цього плата буде скинута (виконано Reset), і почнеться завантаження програми в її пам'ять. У процесі завантаження ви помітите миготіння світлодіодів RX і TX на платі, що свідчить про передачу даних між платою та комп'ютером.

ПИТАННЯ ДЛЯ ПІДГОТОВКИ

1. Які радіоелектронні компоненти можна віднести до базових?
2. Які графічні позначення мають базові радіоелектронні компоненти?
3. Яке призначення компонентів: резистор, конденсатор, транзистор, реле, фоторезистор, світлодіод, діод, діодний міст, фотодіод?
4. Як підключити, налагодити та запрограмувати плату Arduino Uno?
5. Для чого використовується онлайн-сервіс **tinkercad**?
6. Для чого використовується онлайн-сервіс **Wokwi**?

Лабораторна робота №4 (Використання вимірювальних приладів)

Тема: Використання вимірювальних приладів.

Мета: Ознайомитись з основами побудови електронних схем та використання вимірювальних приладів.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

1. Опис основних функцій та технічних характеристик цифрового мультиметра



Цифровий мультиметр – **Digital MultiMeter (DMM)** – це багатофункціональний вимірювальний прилад, спеціально призначений в основному для статичних вимірювань кількох електричних (*наприклад, змінних та постійних напруг і струмів, опору, частоти*) та неелектричних (*наприклад, температури*) величин.

Структура цифрового мультиметра подібна до структури будь-якого цифрового вимірювального пристрою. Відмінність лише у вході приладу, що має кілька спеціальних перетворювачів конкретних входних фізичних величин.

Цифровий мультиметр з мінімальним набором вимірюваних величин призначений для вимірювання: постійних (**DC** – *Direct Current*) і змінних (**AC** – *Alternating Current*) напруг і струмів, а також опору **R**.

Вхідні вимірювані величини в будь-якому випадку спочатку перетворюються на пропорційну напругу постійного струму що надходить на вхід АЦП, де і перетворюється на код. Отриманий код потрапляє до мікроконтролера де обробляється за спеціальним алгоритмом, а результат обробки в цифровому вигляді відображається на дисплеї індикатора.

Конкретний режим вимірювання визначається положеннями перемикача. Наприклад для вимірювання опору перемикач встановлюється у положення **R**, а це у свою чергу включає перетворювач опору в постійну напругу.

У режимі вимірювання струму використовується внутрішній шунт (*точний резистор малого опору $R_{ш}$*). При цьому струм, що вимірюється, протікаючи по резистору $R_{ш}$, створює пропорційне струму падіння напруги яке надходить на вхід АЦП для подальшої обробки.

В даний час у практиці технічних вимірювань найбільш поширені мультиметри першої групи – компактні (*Hand-Held* – які уміщуються на долоні) і мініатюрні (*Pocket Size* – кишенькового формату) мультиметри, які дуже добре відповідають вимогам до переносних приладів для експрес-вимірювань.

Вони мають малі габаритні розміри та масу; забезпечують можливість вимірювання кількох різних величин у широких діапазонах; мають цілком задовільні точність та чутливість. У цих приладів хороші експлуатаційні характеристики, автономне живлення, вони надійні, зручні та прості у користуванні. Більшість сучасних мультиметрів забезпечує автоматичний вибір полярності та діапазону вимірювання.

Типовий набір вимірюваних мультиметрами широкого застосування величин включає постійні та змінні напруги, постійні та змінні струми, опір постійному струму. Діапазони основних вимірюваних величин такі:

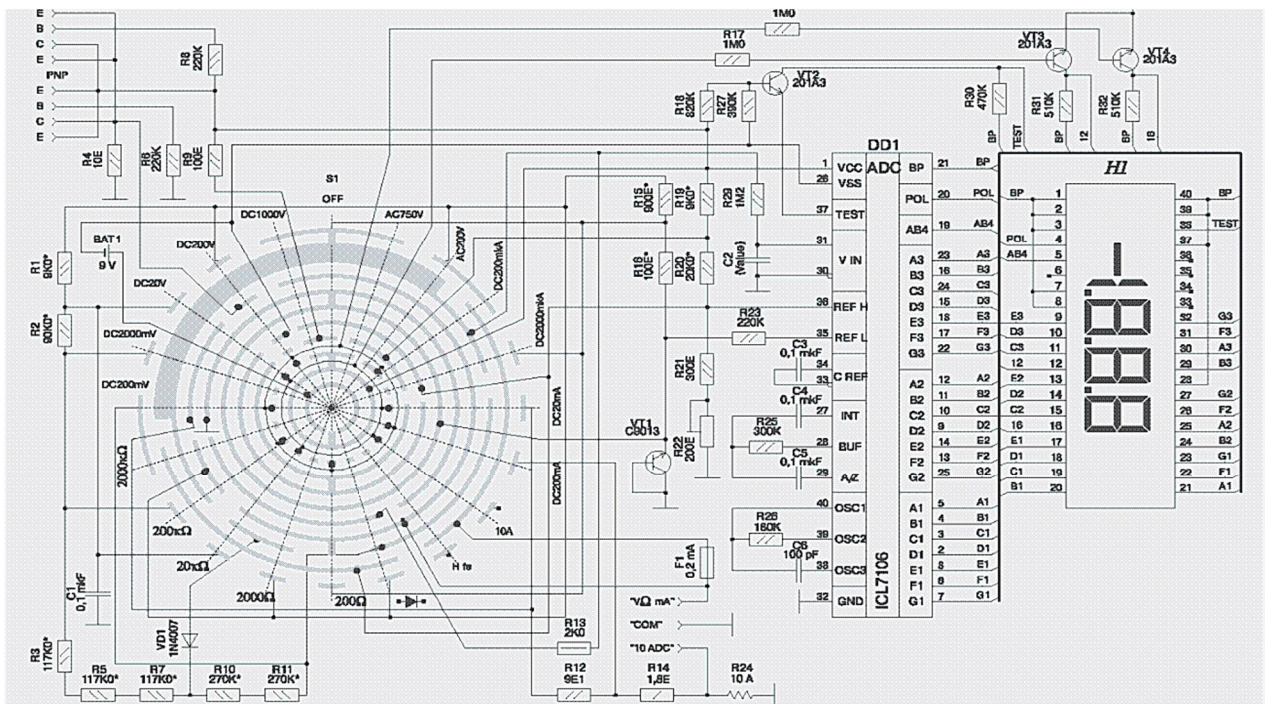
- напруга (*постійного та змінного струму*) – від часток мілівольта до кіловольта;
- струм (*без зовнішніх шунтів або трансформаторів струмів*) – від десятків міліампер до десяти ампер;
- Опір – від часток ома до десятків мегаом.

Деякі моделі мають додаткові можливості, наприклад, вимірювання температури за допомогою стандартних термопар; вимірювання електричної ємності; вимірювання частоти; режим перевірки напівпровідникових приладів; режим «продзвонювання» ланцюгів та ін.

Класи точності різних моделей мультиметрів можуть бути задані по-різному. Як правило, використовуються граничні значення основних абсолютних та відносних похибок. Класи точності звичайних мультиметрів лежать у діапазоні від 0,1 до 5,0% (*залежно від можливостей, параметрів, діапазонів, вартості приладу*). Роздільна здатність R: від 1:200 до 1:20 000. Маса приладів: 100 г... 1 кг.

Найпопулярнішим мультиметром у всьому світі є мультиметр моделі **M830B**. Прилад випускається під двома десятками різних торгових марок для різних регіонів та країн світу, але внутрішнє виконання завжди однакове. Як показала практика M830B досить надійний, але це зворотний бік простоти. Точних та різноманітних вимірів цей мультиметр не забезпечує, але в побутових умовах досить корисний і недорогий. Мультиметр M830B призначений для вимірювання постійної напруги та струму, змінної напруги, опору, коефіцієнта статичної передачі транзисторів струму, продзвінки напівпровідників. Відображення результатів вимірювань здійснюється на рідкокристалічний індикатор. Прилад забезпечує автоматичне визначення полярності вимірюваної напруги та струму. За секунду виконується 2-3 виміри.

Вибір вимірюваних величин та меж вимірювань встановлюється вручну центральним перемикачем. Прилад працює у діапазоні температур від 0 до 40°C і живиться від стандартної крони.



Працювати з мультиметром треба за допомогою кругового перемикача зі стрілкою. За умовчанням вона виставлена у положення «OFF» і це означає що прилад вимкнено.

Стрілку можна обертати в будь-якому напрямку. Працюючи з цифровим мультиметром, можна вимірювати значення змінного і постійного струму та напруги.

Під час вимірювань треба мати на увазі що у промисловості та побуті використовується змінний струм. Саме він "тече" високовольтними лініями електропроводів від генераторів електростанцій в наші будинки, "запалює" наші лампи освітлення і живить різні побутові електроприлади. Якщо узяти комп'ютер то всередині системного блоку завжди тече постійний струм, тому

що блок живлення комп'ютера перетворює змінний струм на постійну низьку напругу яка необхідна для живлення комплектуючих.

Приклад використання мультиметра для вимірювання ємності звичайної пальчикової батарейки типу AA номіналом 1,5 Вольт.

Перемикач необхідно поставити у положення вище, ніж значення, що вимірюються. Відомо, що в батарейці - 1,5V і це - постійний струм. Відповідно - виставляємо перемикач "межа" вимірювань за шкалою постійного струму 20 Вольт. Як показано на фото нижче.



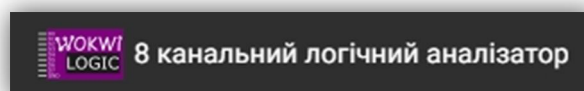
Червоний щуп треба вставити у гніздо **VΩmA**, чорний у гніздо **COM**

Потім беремо наш гальванічний елемент (батарейку) і прикладаємо до неї вимірювальні "щупи" мультиметра. Як правило, червоний на «+», а чорний на «-».

Після закінчення усіх вимірювань і завершення роботи з приладом треба перемикач поставити у положення «**OFF**»

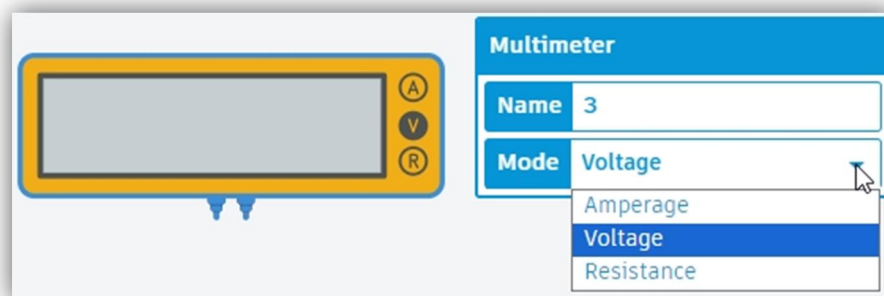
Онлайн-сервіси **Tinkercad** (<https://www.tinkercad.com>) і **Wokwi** (<https://wokwi.com/>) мають різні вимірювальні прилади.

Наприклад **Wokwi** має тільки 8-канальний логічний аналізатор, який дозволяє бачити зміну логічного стану на тому чи іншому вході.

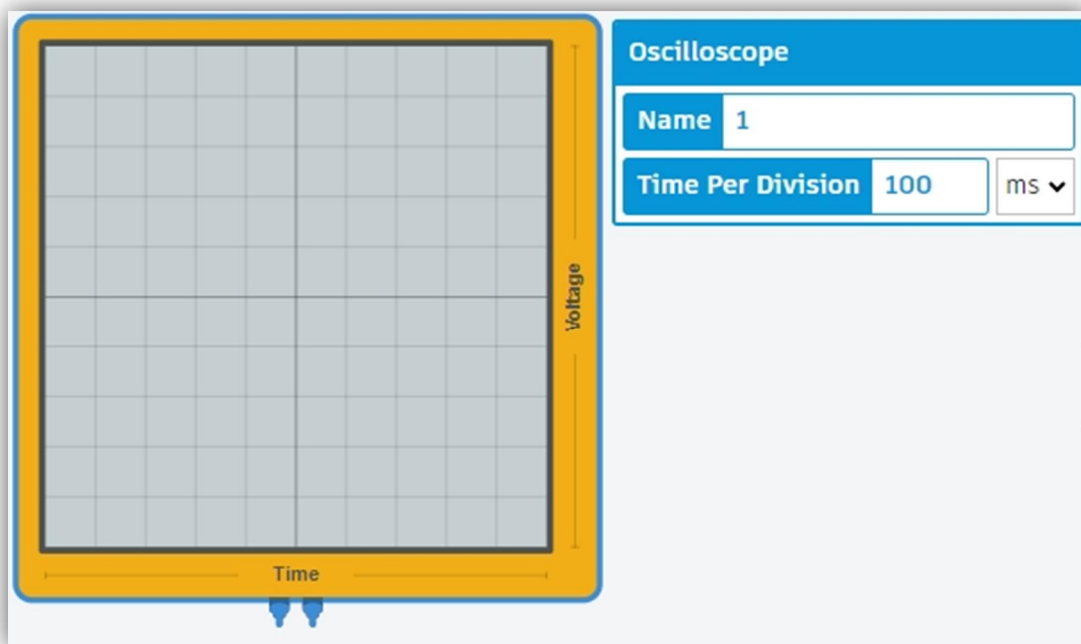


Tinkercad більш багатший на вимірювальні прилади:

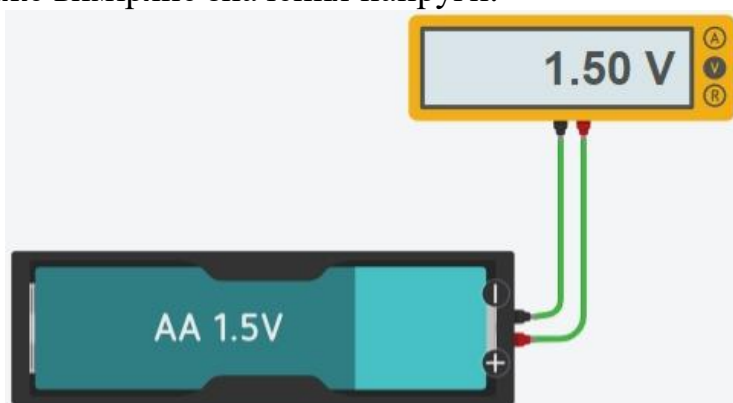
1. Мультиметр, який дозволяє вимірювати напругу (**Voltage**), струм (**Amperage**), опір (**Resistance**).



2. Осцилограф, який дозволяє бачити як змінюється форма досліджуваного сигналу в часі.



Спробуємо визначити напругу батарейки, для цього підключаємо мультиметр до батарейки, так як показано на рисунку і після запуску симуляції мультиметр покаже виміряне значення напруги.

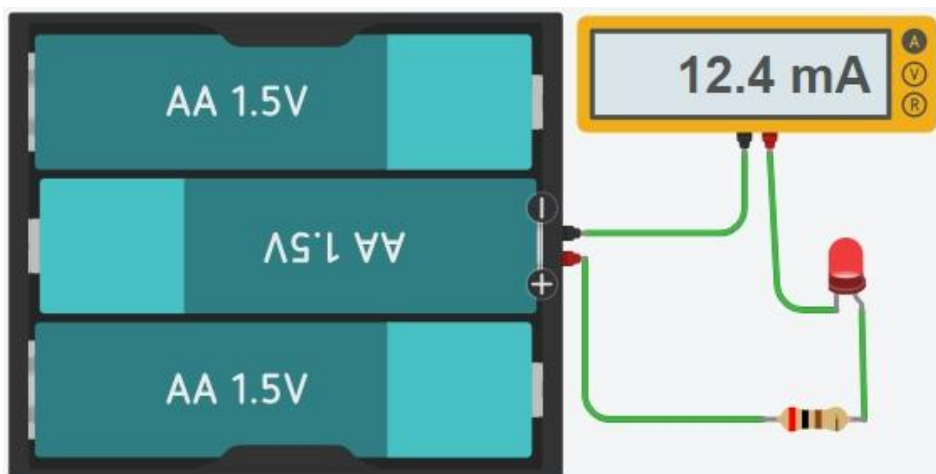


Для вимірювання опору треба мультиметр переключити у режим **R** (**Resistance**) і підключити резистор, або інший елемент до його входів.



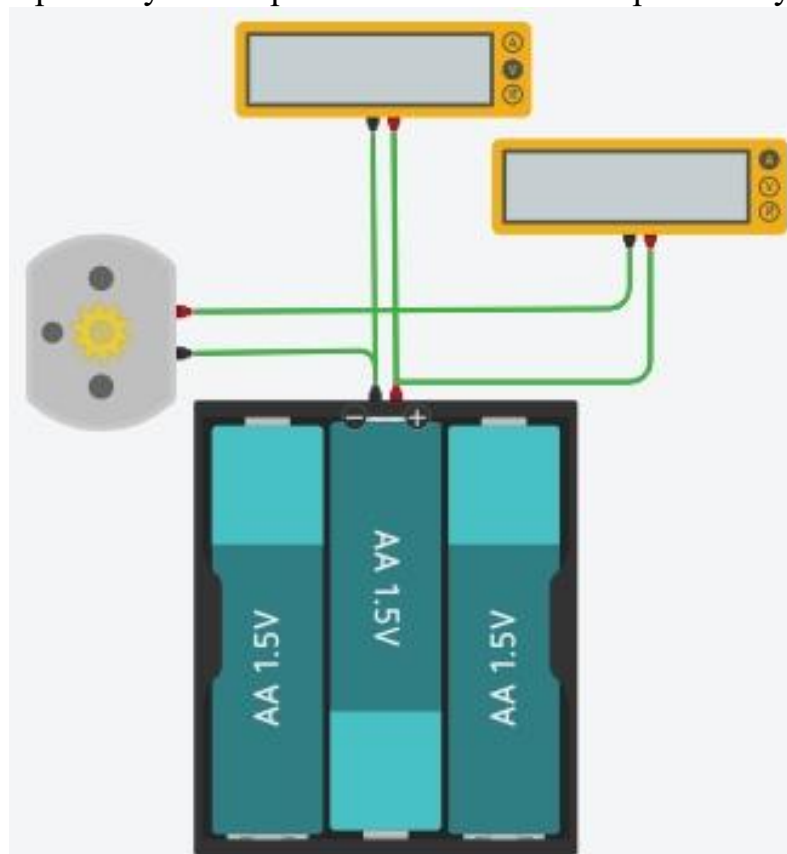
Зверніть увагу на те, що мультиметр під час вимірювання різних величин, має можливість автоматично визначати діапазон вимірювання і це дуже зручно.

Для вимірювання сили струму треба мультиметр переключити у режим А (Amperage).



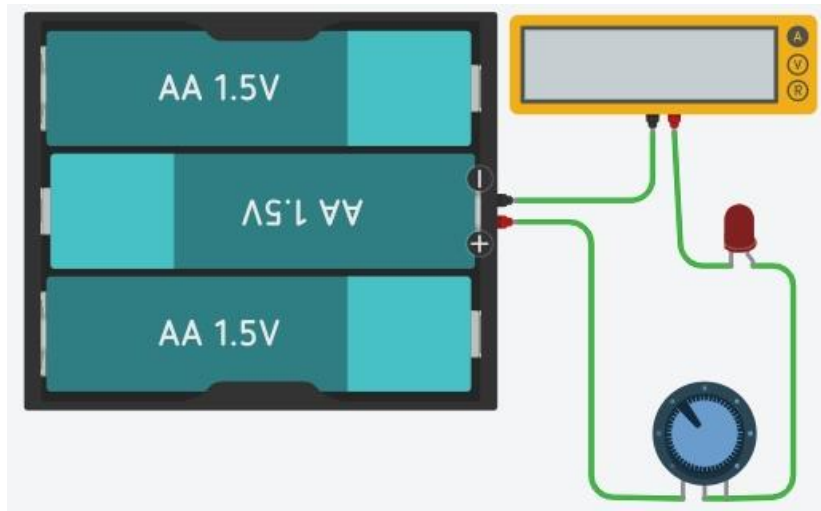
ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Використовуючи сервіс [tinkercad.com](https://www.tinkercad.com) зберіть наступну схему:



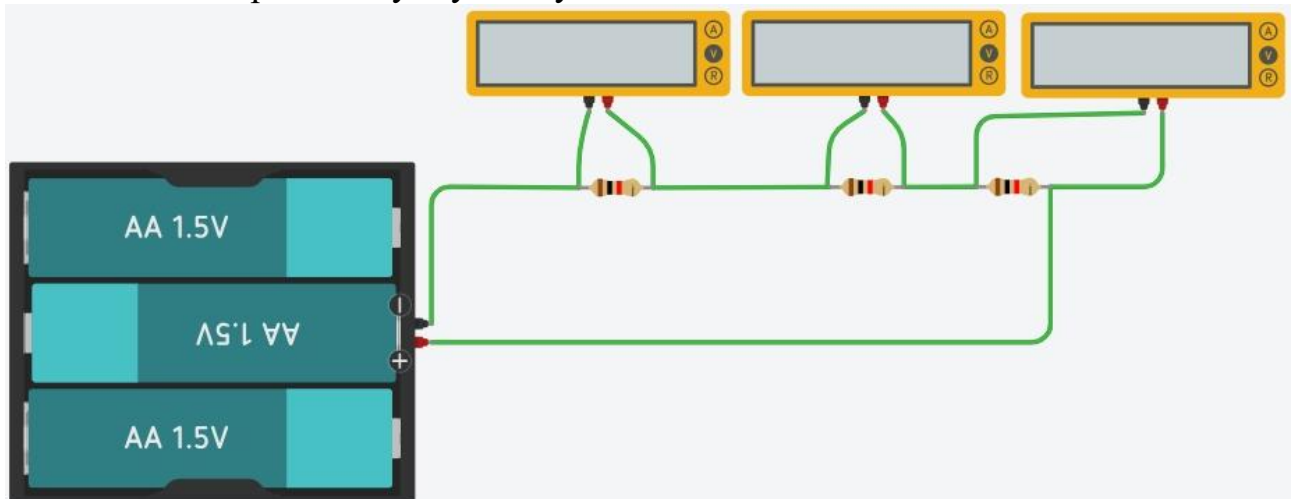
В схемі задіяні такі компоненти: батарея на 4,5 Вольт, двигун постійного струму, два мультиметра (один вимірює напругу, а другий струм).

2. Розрахуйте потужність двигуна відповідно до показань вимірювальних приладів.
3. Зберіть наступну схему:



4. Запустіть симуляцію. Обертаючи ручку змінного резистора, виставте струм на мультиметрі 10 мА. Визначте опір резистора (для цього відключіть змінний резистор від «+» батарейки і підключіть до нього ще один мультиметр, переключивши його у режим **R (Resistance)**).

5. Зберіть наступну схему:



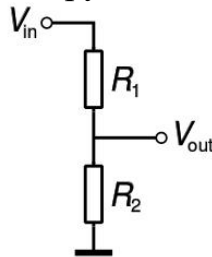
В схемі використовуються такі компоненти: батарея на 4,5 Вольт, три резистори на 1 Ком, три мультиметра (настроєні на вимірювання напруги).

Визначте падіння напруги на кожному резисторі і заповніть таблицю:

	Значення опору резисторів, Ком			Величина падіння напруги на резисторах, В.		
	R1	R2	R3	R1	R2	R3
1	1	1	1			
2	1	2	3			
3	0,1	1	10			
4	1	10	100			
5	100	150	200			
6	0,5	1,5	15			
7	5	2	12			

Дільник напруги на резисторах

Схема дільника напруги включає в себе вхідне джерело напруги і два резистора. Позначимо резистор, який знаходиться ближче до плюса вхідної напруги (V_{in}) як R_1 , а резистор знаходиться ближче до мінуса як R_2 . Падіння напруги (V_{out}) на резисторі R_2 – це знижена напруга, отримана в результаті застосування резисторного дільника напруги.



Розрахунок дільника напруги припускає, що нам відомо, принаймні, три величини з наведеної вище схеми: вхідна напруга та опір обох резисторів. Знаючи ці величини, ми можемо розрахувати вихідну напругу.

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

В радіоелектроніці є багато способів застосування дільника напруги. Ось тільки деякі приклади де ви можете виявити їх.

Потенціометри. Потенціометр являє собою змінний резистор, який може бути використаний для створення регульованого дільника напруги.

Зсередини потенціометр являє собою резистор і ковзний контакт, який ділить резистор на дві частини і пересувається між цими двома частинами. З зовнішньої сторони, як правило, у потенціометра є три висновки: два контакти під'єднані до виводів резистора, в той час як третій (центральний) підключений до ковзаючої контакту.

Якщо контакти резистора підключити до джерела напруги (один до «-», інший до «+»), то центральний вивід потенціометра буде імітувати дільник напруги.

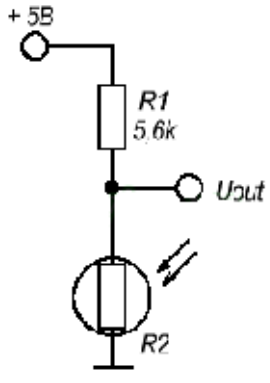
Резистивні датчики. Більшість датчиків застосовуються в різних пристроях. Фоторезистор являє собою змінний резистор, який змінює свій опір, пропорційно кількості світла, що падає на нього. Так само є й інші датчики, такі як датчики тиску, прискорення, температури та інші.

Так само резистивний дільник напруги допомагає виміряти напругу за допомогою мікроконтролера (при наявності АЦП).

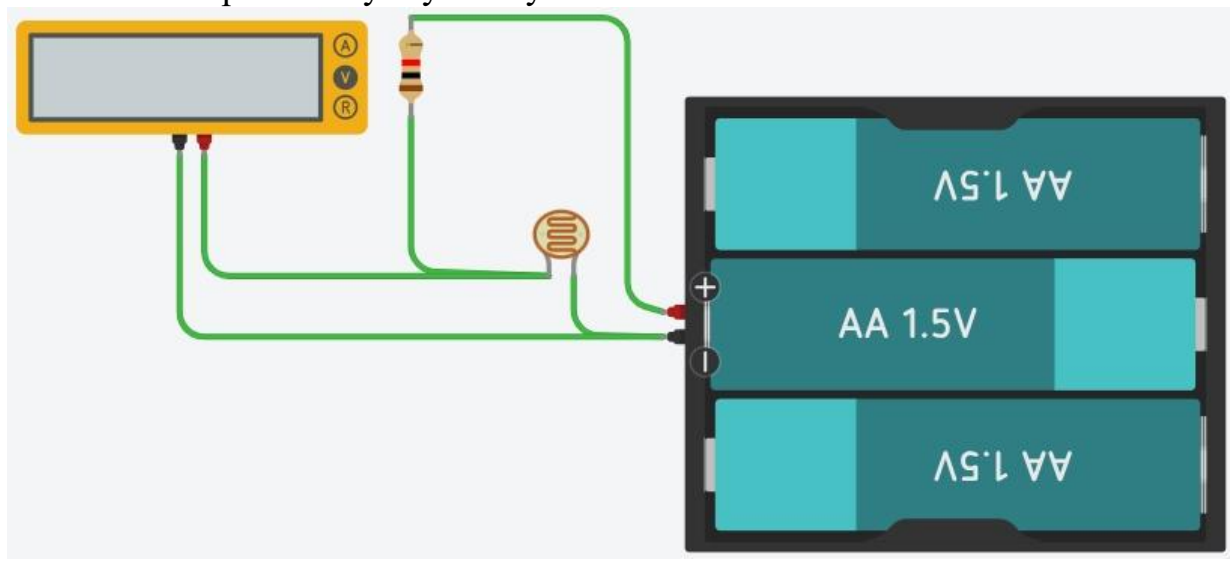
Приклад роботи дільника напруги на фоторезисторі.

Припустимо, опір фоторезистора змінюється від 1 кОм (при освітленні) і до 10 кОм (при повній темряві).

Якщо ми доповнимо схему постійним опором приблизно 5,6 кОм, то ми можемо отримати широкий діапазон зміни вихідної напруги при зміні освітленості фоторезистора.



6. Зберіть наступну схему:

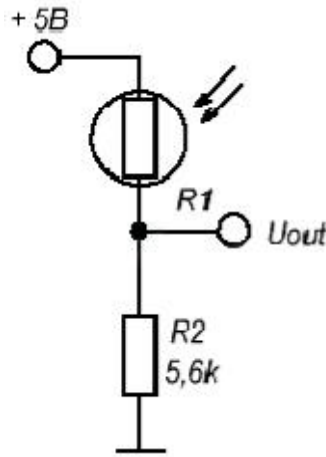


В схемі використовуються такі компоненти: батарея на 4,5 Вольт, один резистор на 1 Ком, один фоторезистор, один мультиметр (настроєний на вимірювання напруги).




Заповніть таблицю:

	Рівень освітленості	Вихідна напруга, В
1		
2		
3		

7. Зберіть наступну схему:



Заповніть таблицю:

	Рівень освітленості	Вихідна напруга, В
1		
2		
3		

8. Порівняйте показання в обох таблицях. Зробіть висновки.
9. Оформіть звіт, додайте до нього необхідні скріншоти і відправте на перевірку.

ПИТАННЯ ДЛЯ ПІДГОТОВКИ.

1. Призначення цифрових мультиметрів?
2. Які дві групи сучасних ЦМ найпоширеніші?
3. Що основою мультиметра M830B?
4. Чим визначається конкретний режим виміру?
5. Як забезпечується вимірювання напруги у кількох діапазонах?
6. Як вимірюється змінна напруга мультиметром?
7. Чим визначено нижню межу вимірювання змінної напруги?
8. Навіщо використовується внутрішній шунт?
9. Як відбувається вимірювання опору в мультиметрі M830B?
10. Індикатор якого типу та якої розрядності використовується у мультиметрі?
11. Як на індикаторі мультиметра відображається розрядність числа?
12. Переваги мультиметра M-830B.

Лабораторна робота №5 (Підключення світлодіодів до плати Arduino UNO)

Тема: Підключення світлодіодів до плати Arduino UNO.

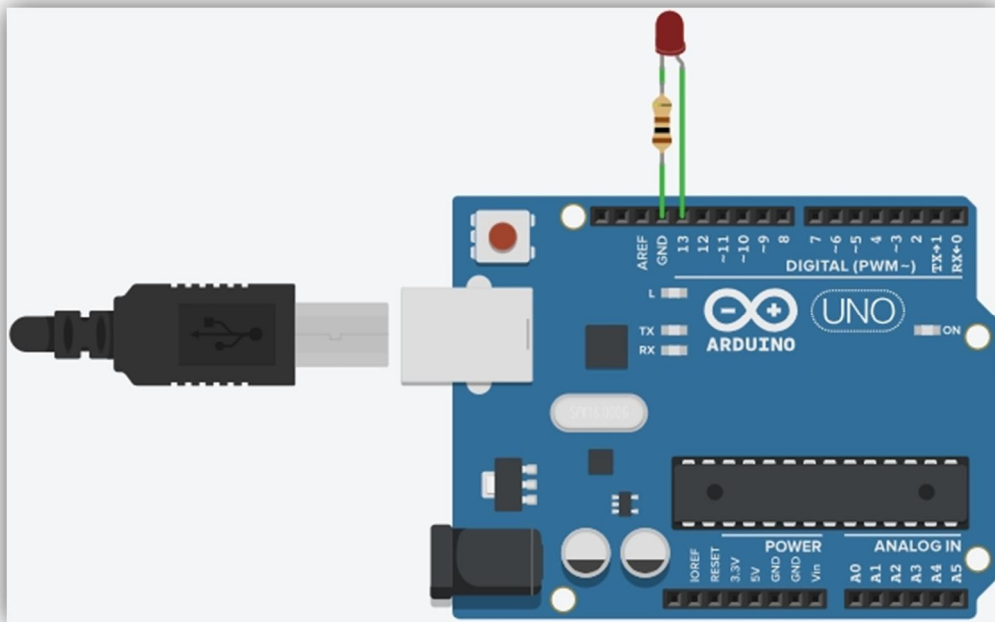
Мета: Ознайомитись з основами побудови електронних схем, написання програми (скетчу) та емуляції роботи зібраної схеми.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

Для засвоєння принципів роботи з мікроконтролером необхідно виконати декілька простих проектів. Такі проекти використовують світлодіоди і різні способи їх підключення, це дозволить ознайомитись з управлінням виходами та входами платформи Arduino. Виконання проектів також дозволить вам ознайомитись з основними компонентами, такими як світлодіоди, кнопки і резистори. Крім того, ви зможете вибирати і застосовувати резистори різного номіналу, що є важливим для забезпечення правильної роботи пристроїв введення та виведення.

Беремо перший традиційний проект **Blink**, в якому плата Arduino UNO буде блимати світлодіодом. Для цього використовується онлайн-сервіс **tinkercad.com**.

Збираємо схему:



Пишемо скетч і натискаємо кнопку «Почати моделювання». Якщо в схемі і в програмі немає помилок то ми побачимо як будуть одночасно блимати червоні світлодіоди на платі Arduino UNO і на монтажній платі.

Текст скетчу:

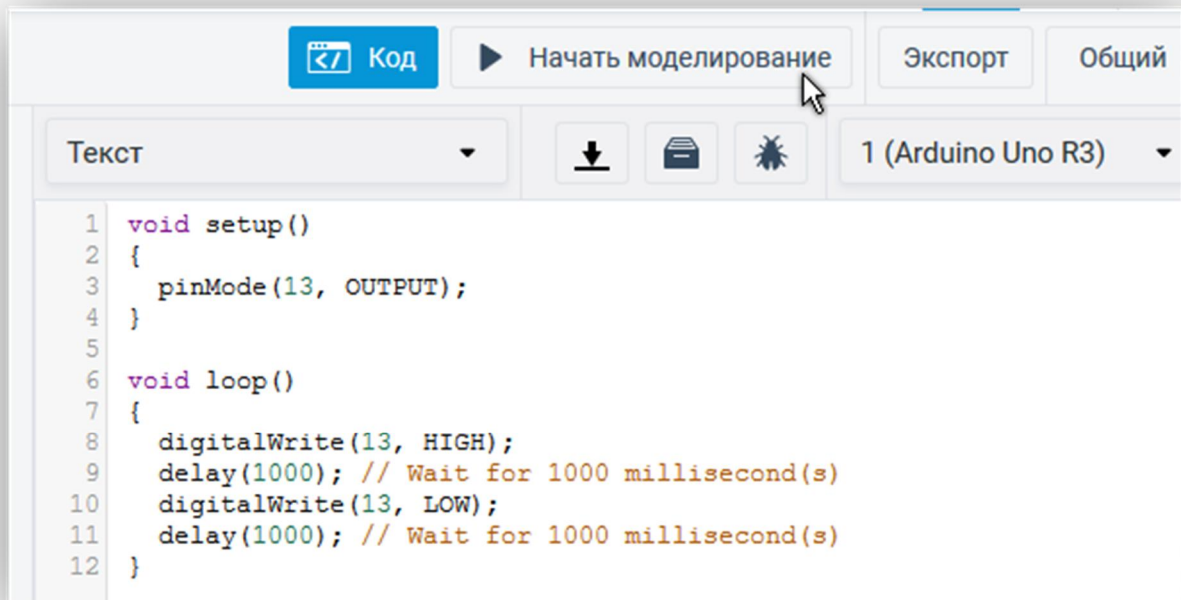
```
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
```

```

{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}

```

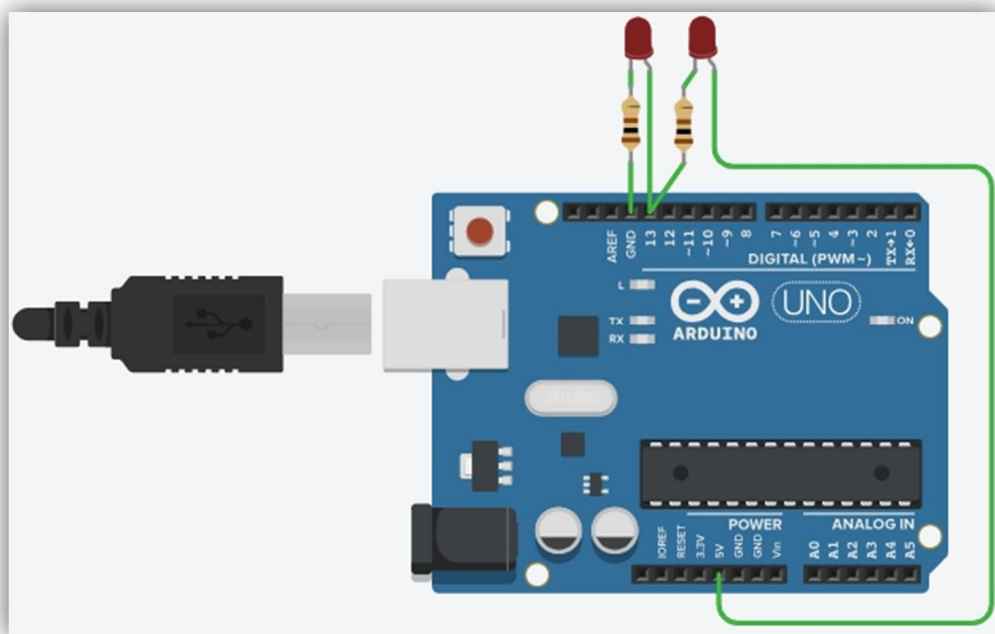
Вікно сервісу **tinkercad.com** з текстом написаної програми.



Розберіться з текстом програми і прокоментуйте кожний рядок. Спробуйте змінити паузу `delay(1000)`, вказавши інший час.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Використовуючи сервіс **tinkercad.com** зберіть наступну схему:

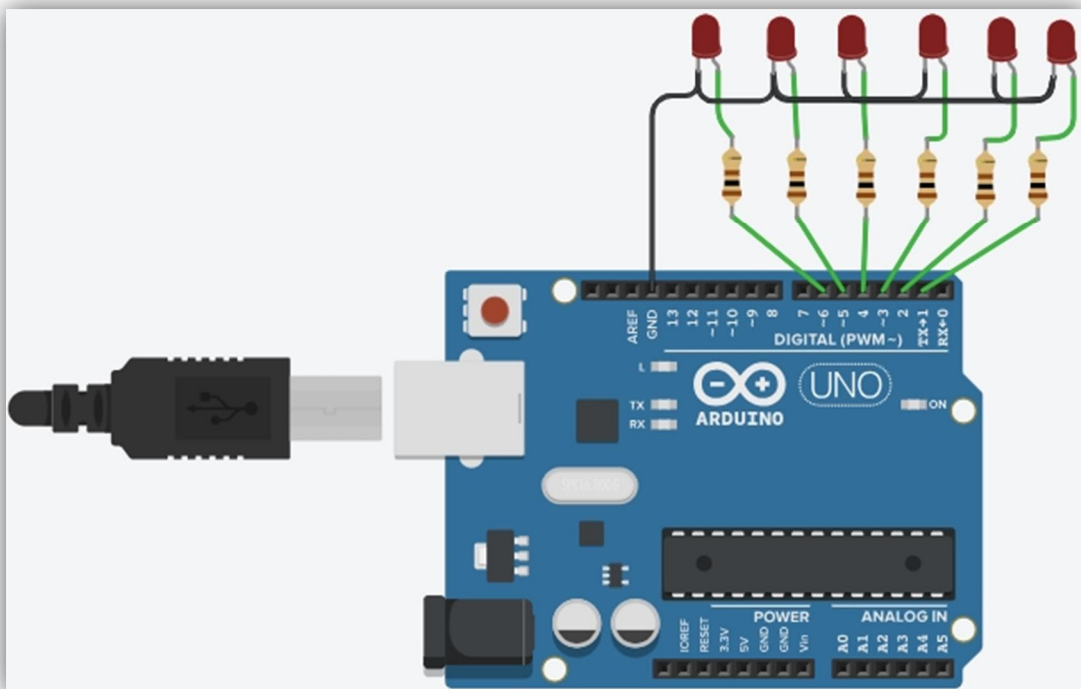


Використовуйте скетч з прикладу:

```
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Натискаємо кнопку «Почати моделювання». Якщо в схемі і в програмі немає помилок то ми побачимо як світлодіоди будуть мигати по черзі.

2. Зберіть наступну схему:



3. Побудувати алгоритм і написати скетч який виконував би наступне:

- блимати усіма світлодіодами з паузою 0,5 секунди;
- блимати світлодіодами по черзі парними і непарними;
- зробити «біжучий вогонь» із світлодіодів;

4. Самостійно зібрати схему, яка включає в себе три світлодіода (червоний, жовтий, зелений), три резистора і плату Arduino UNO. Написати скетч, який би перетворив зібрану схему на світлофор, що змінює колір із червоного на жовтий, а далі на зелений, після певної паузи дії повторюються.

5. Спробуйте поміняти схему попереднього завдання, замінивши три світлодіода на один RGB-світлодіод. Скетч залиште без змін. Спробуйте зібрати цей проєкт у сервісі **Wokwi** (<https://wokwi.com/>). Запустіть симуляцію. Зробіть необхідні скріншоти і додайте їх до звіту.

6. Запропонуйте декілька власних схем із підключенням світлодіодів до плати Arduino UNO. І напишіть до них скетч. Зберіть ці схеми у сервісі Wokwi (<https://wokwi.com/>)

ПИТАННЯ ДЛЯ ПІДГОТОВКИ.

1. З яких частин складається скетч?
2. Як конфігурувати порти (піни)?
3. Які команди керують пінами?
4. Як підключити світлодіод до плати Arduino?

Лабораторна робота №6 (Введення даних за допомогою кнопок)

Тема: Введення даних за допомогою кнопок.

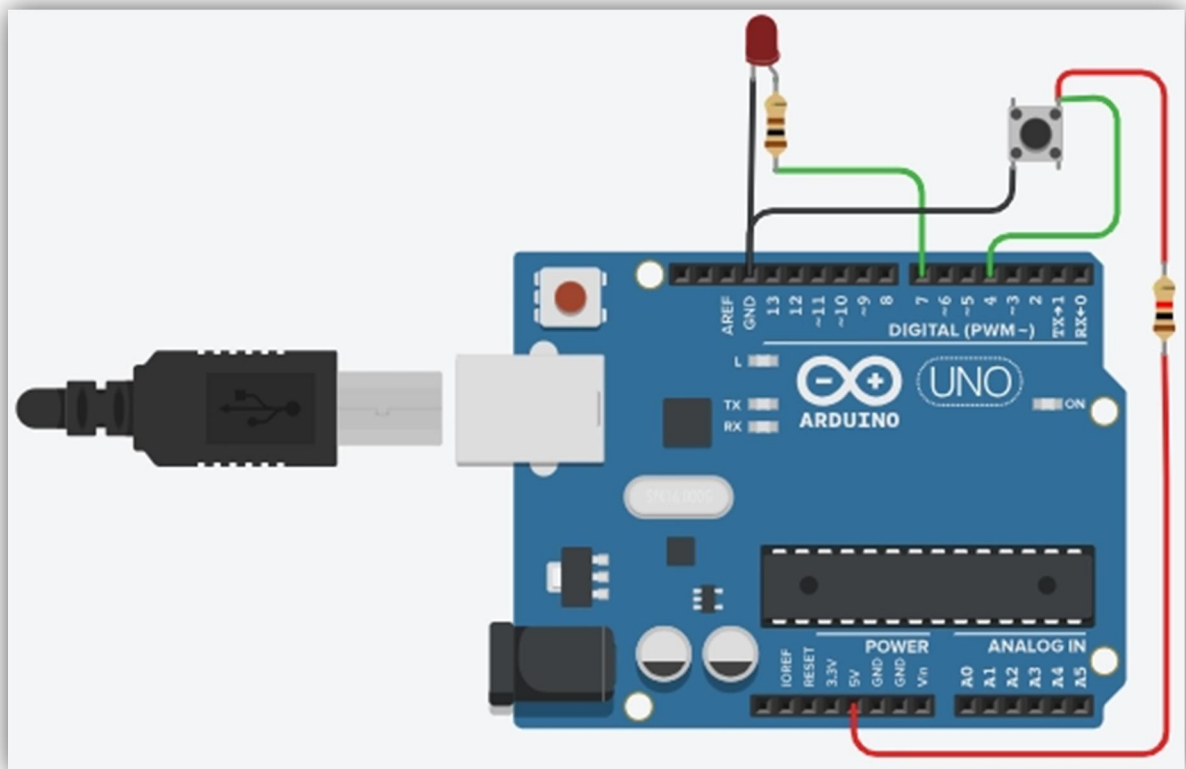
Мета: Навчитися підключати кнопки до плати Arduino.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

Для засвоєння принципів роботи з мікроконтролером необхідно виконати декілька простих проектів. Такі проекти використовують світлодіоди, кнопки, перемикачі та різні способи їх підключення, це дозволить ознайомитись з управлінням виходами та входами в Arduino.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Використовуючи сервіс **tinkercad.com** зберіть наступну схему:



В схемі використовуються такі радіоелементи: світлодіод, кнопка, резистори з наміналом 100 Ом і 1 Ком.

Використовуйте скетч з прикладу:

```
const int buttonPin = 4;
const int ledPin = 7;
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);
}}

```

Цей скетч можна трохи спростити, записавши таким чином:

```

int buttonState = 0;
void setup() {
  pinMode(7, OUTPUT);
  pinMode(4, INPUT);
}
void loop() {
  buttonState = digitalRead(4);
  if (buttonState == HIGH) {
    digitalWrite(7, HIGH);
  } else {
    digitalWrite(7, LOW);
  }
}}

```

Порівняйте обидва скетча і визначте відмінності.

Натискаємо кнопку «Почати моделювання». Якщо в схемі і в програмі немає помилок то ми побачимо, що світлодіод горить коли кнопка відпущена і гасне при натисканні на неї.

Розберіться з кожним рядком скетча і алгоритмом його роботи.

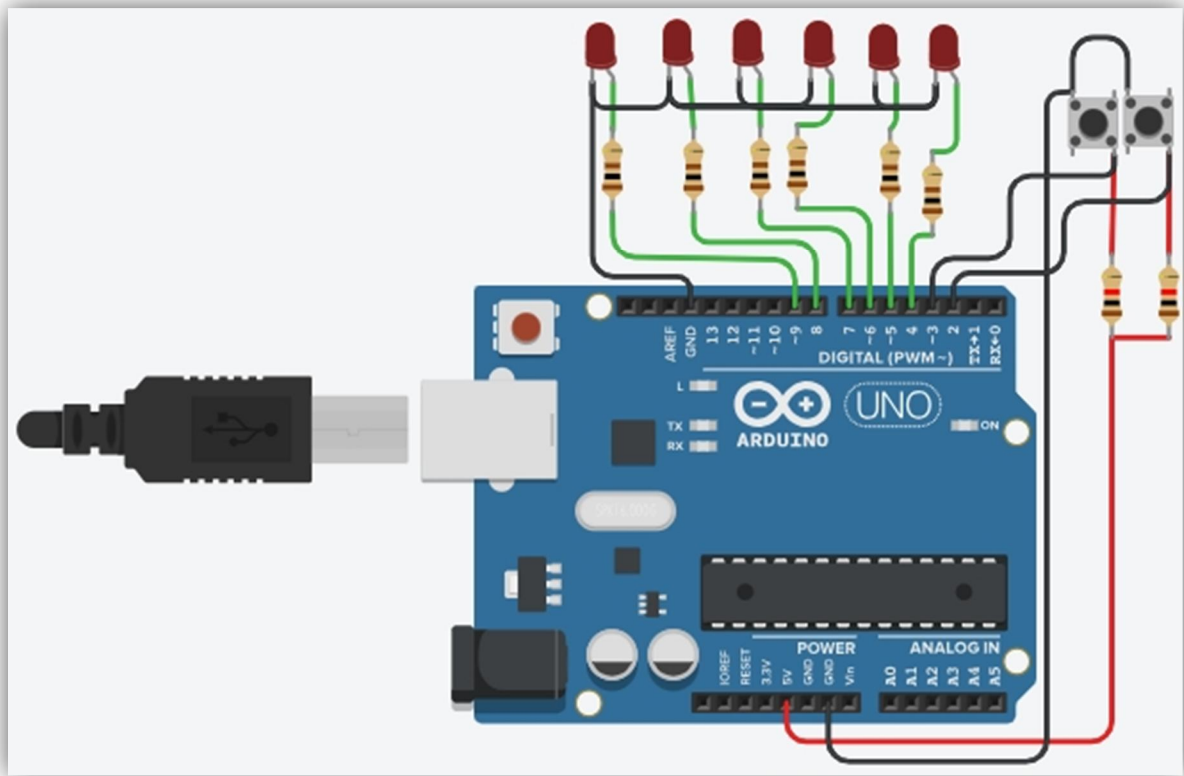
2. Складіть алгоритм і напишіть програму яка буде виконувати наступне:

- При натисканні і утримуванні кнопки засвічується світлодіод, а при відпусканні кнопки світлодіод гасне.
- При однократному натисканні на кнопку світлодіод засвічується, а при повторному натисканні світлодіод гасне.
- При однократному натисканні на кнопку світлодіод засвічується на 2 секунди, а потім гасне.
- При однократному натисканні на кнопку світлодіод блимає 5 разів, а потім гасне.

3. Спробуйте зібрати ці проєкти у сервісі Wokwi (<https://wokwi.com/>). Зробіть необхідні скріншоти і додайте їх до звіту.

4. Зібрати наступну схему:

У схемі задіяні такі адіоеlementи: 6 резисторів по 100 Ом, 6 світлодіодів, 2 кнопки, 2 резистора опором 1 Ком.



- При однократному натисканні лівої кнопки засвічуються усі світлодіоди, а при натисканні правої гаснуть.
- При однократному натисканні лівої кнопки засвічуються усі парні світлодіоди, а при натисканні правої засвічуються усі непарні.
- При натисканні на кнопки збільшується або зменшується кількість світлодіодів що світяться.
- При запуску програми світиться перший світлодіод, при натисканні на праву кнопку засвічується другий світлодіод, а перший гасне, наступне натискання правої кнопки засвітить наступний третій світлодіод. При натисканні лівої кнопки свічення світлодіодів переміщується в лівий бік.

ПИТАННЯ ДЛЯ ПІДГОТОВКИ.

1. Як конфігурувати пін на вхід?
2. Яка команда відповідає за зчитування даних з піна?
3. Як позначається кнопка на схемі?
4. Як позначається світлодіод на схемі?
5. Як правильно підключити кнопку до порта плати Arduino?

Лабораторна робота №7 (Підключення потенціометра до плати Arduino)

Тема: Підключення потенціометра до плати Arduino

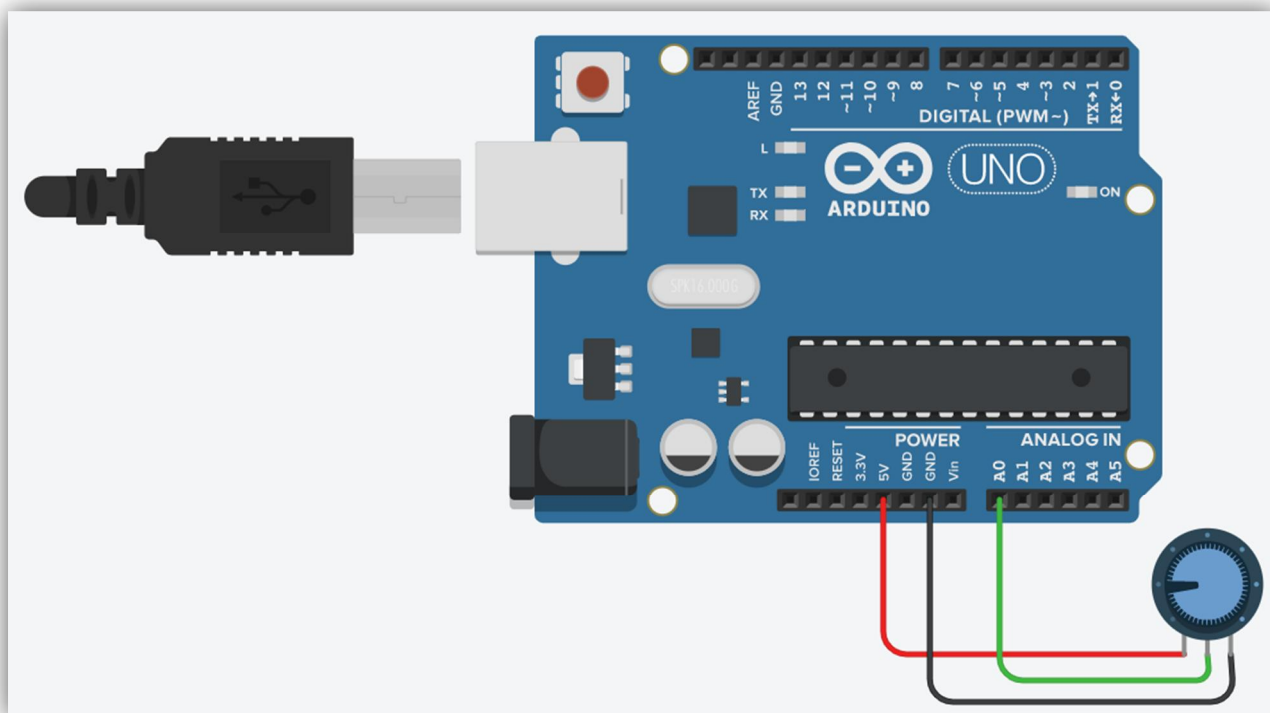
Мета: Навчитися обробляти аналогову інформацію можливостями плати Arduino.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

Для засвоєння принципів роботи з платформою Arduino необхідно виконати низку простих проектів, які розкривають базові можливості платформи. Такі проекти передбачають використання світлодіодів, кнопок, потенціометрів, аналогових і цифрових датчиків, виконання таких проектів дозволить ознайомитись з управлінням виходами та входами Arduino-сумісної платформи.

ПОРЯДОК ВИКОНАННЯ РОБОТИ.

1. Використовуючи сервіс **tinkercad.com** зберіть наступну схему:



Використайте запропонований скетч:



```
const int analogInPin = A0;
int sensorValue = 0;
int outputValue = 0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  sensorValue = analogRead(analogInPin);
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
```

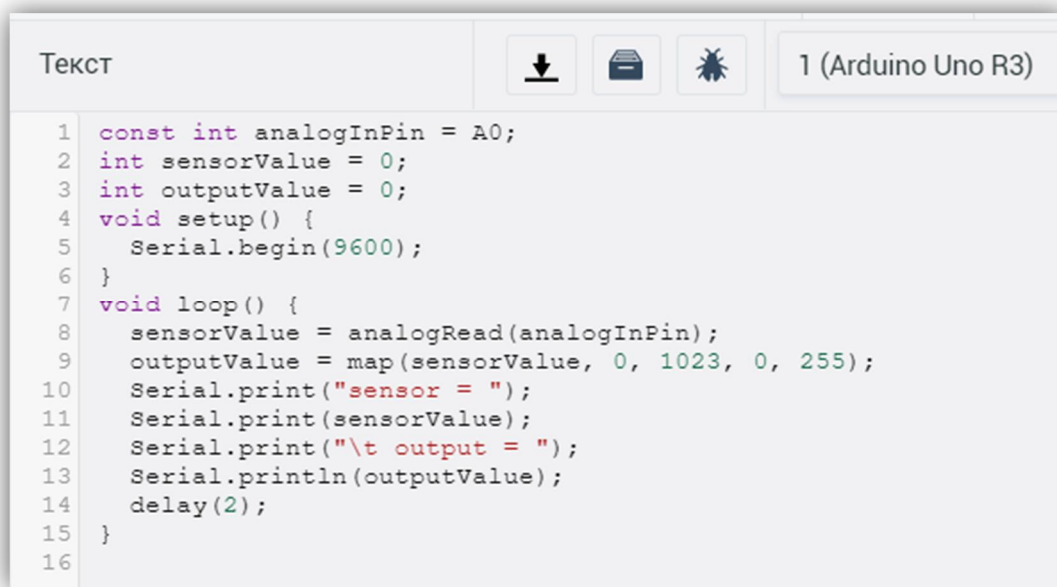
```




Serial.println(outputValue);
delay(2);
}

```

Натискаємо кнопку «Почати моделювання». Якщо в схемі і в програмі немає помилок то ми побачимо, що яскравість світлодіоду залежить від положення ручки потенціометра, яке можна змінювати за допомогою миші.

Натисніть кнопку  Код потім  Монитор послідовального інтерфейса, на екрані з правого боку побачимо текст програми і дані Монітору послідовного порта.

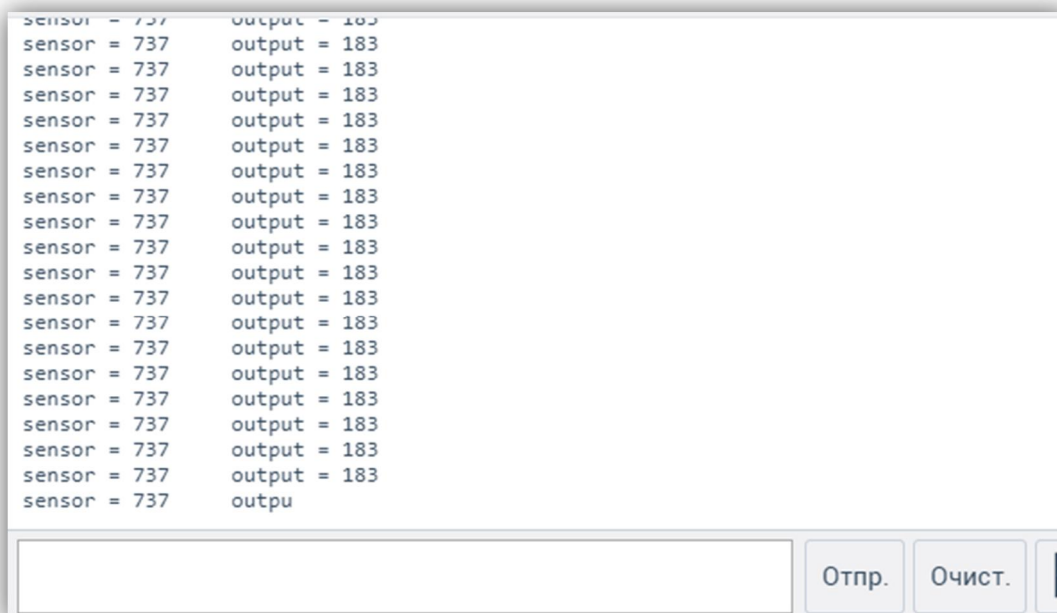


Текст    1 (Arduino Uno R3)

```

1  const int analogInPin = A0;
2  int sensorValue = 0;
3  int outputValue = 0;
4  void setup() {
5      Serial.begin(9600);
6  }
7  void loop() {
8      sensorValue = analogRead(analogInPin);
9      outputValue = map(sensorValue, 0, 1023, 0, 255);
10     Serial.print("sensor = ");
11     Serial.print(sensorValue);
12     Serial.print("\t output = ");
13     Serial.println(outputValue);
14     delay(2);
15 }
16


```



```

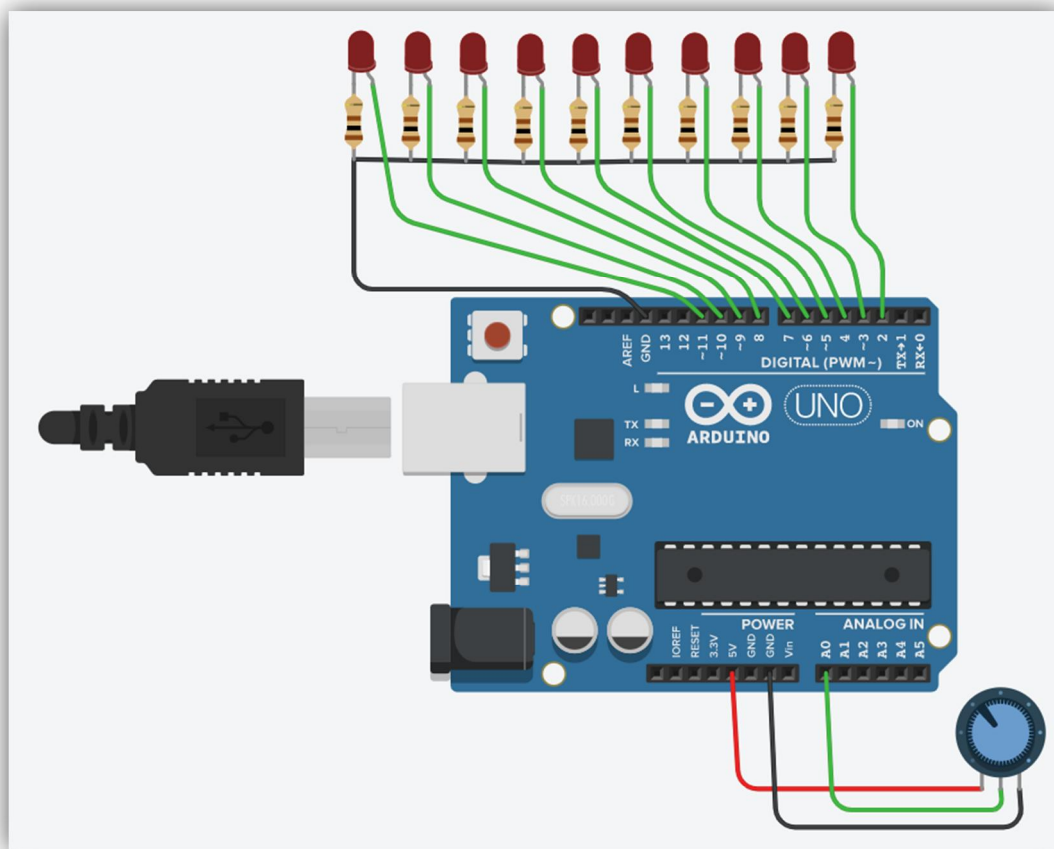
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183
sensor = 737    output = 183

```

Отпр. Очист. 

Розберіться з кожним рядком скетча і алгоритмом його роботи.

2. Зберіть наступну схему (використовуйте для схеми резистори 100 ом, світлодіоди червоного кольору, потенціометр 250 Ком), складіть алгоритм і напишіть програму яка буде виконувати наступне:



- Обертаючи ручку потенціометра змінюємо кількість світлодіодів які світяться, тобто при лівому крайньому положенні ручки потенціометра жоден світлодіод не світиться, а при крайньому правому положенні світяться усі світлодіоди;

- Обертаючи ручку потенціометра змінюємо положення засвіченого світлодіоду, тобто при лівому крайньому положенні ручки потенціометра світиться тільки лівий світлодіод, а при крайньому правому положенні світиться тільки правий світлодіод.

В усіх написаних програмах передбачити виведення даних порта A0 у послідовний порт як це показано у прикладі.

ПИТАННЯ ДЛЯ ПІДГОТОВКИ.

1. Як конфігурувати послідовний порт?
2. Як користуватися монітором послідовного порта?
3. Для чого використовують команду `map` ?
4. Що означає команда `Serial.begin(9600)` ?
5. Яка команда відповідає за зчитування аналогових даних?
6. Як позначається потенціометр на схемі?
7. Як правильно підключити світлодіод на схемі?

Лабораторна робота №8 (Використання широтно-імпульсної модуляції (ШІМ))

Тема: Використання широтно-імпульсної модуляції (ШІМ).

Мета: Навчитися використовувати широтно-імпульсну модуляцію у власних розробках на платформі Arduino.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

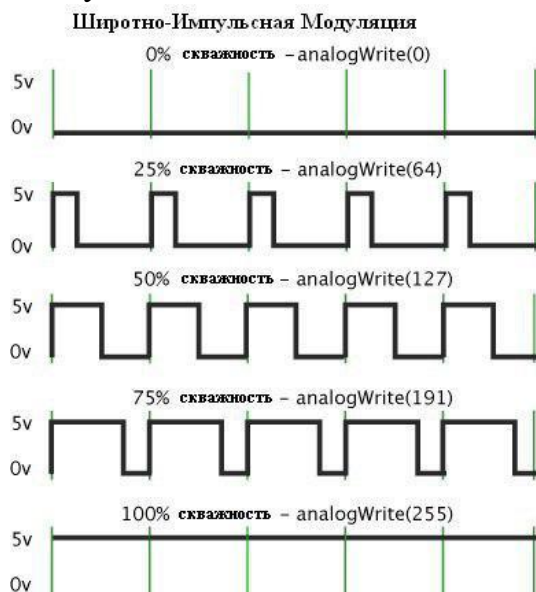
Широтно-імпульсна модуляція (*pulse-width modulation* або *PWM*), також відома як модуляція за тривалістю імпульсів (*PDM*), це процес управління шириною (тривалістю) високочастотних імпульсів відповідно до закону, який визначається низькочастотним сигналом. Цей метод дозволяє регулювати потужність або інтенсивність сигналу шляхом зміни тривалості високочастотних імпульсів відповідно до заданого сигналу низької частоти.

В електроніці, цей процес може використовуватися для контролю середнього значення вихідної напруги шляхом зміни тривалості замкнутого стану електронного ключа. Наприклад, це застосовується у схемах ключового стабілізатора напруги. На практиці, за допомогою ШІМ, можна регулювати яскравість світлодіода, швидкість обертання ротора в двигуні та виконувати інші подібні завдання.

При 100% скважності на цифровому виході Arduino напруга відповідатиме логічному 1 або 5 вольт. Якщо скважність становить 50%, то на виході буде присутня логічна 0 протягом половини часу і логічна 1 протягом іншої половини часу. У цьому випадку середнє значення напруги становитиме 2,5 вольт.

При програмуванні Arduino, скважність зазвичай вказується числами в діапазоні від 0 до 255, особливо, коли використовується 8-бітний мікроконтролер, такий як Arduino Nano або Arduino Uno.

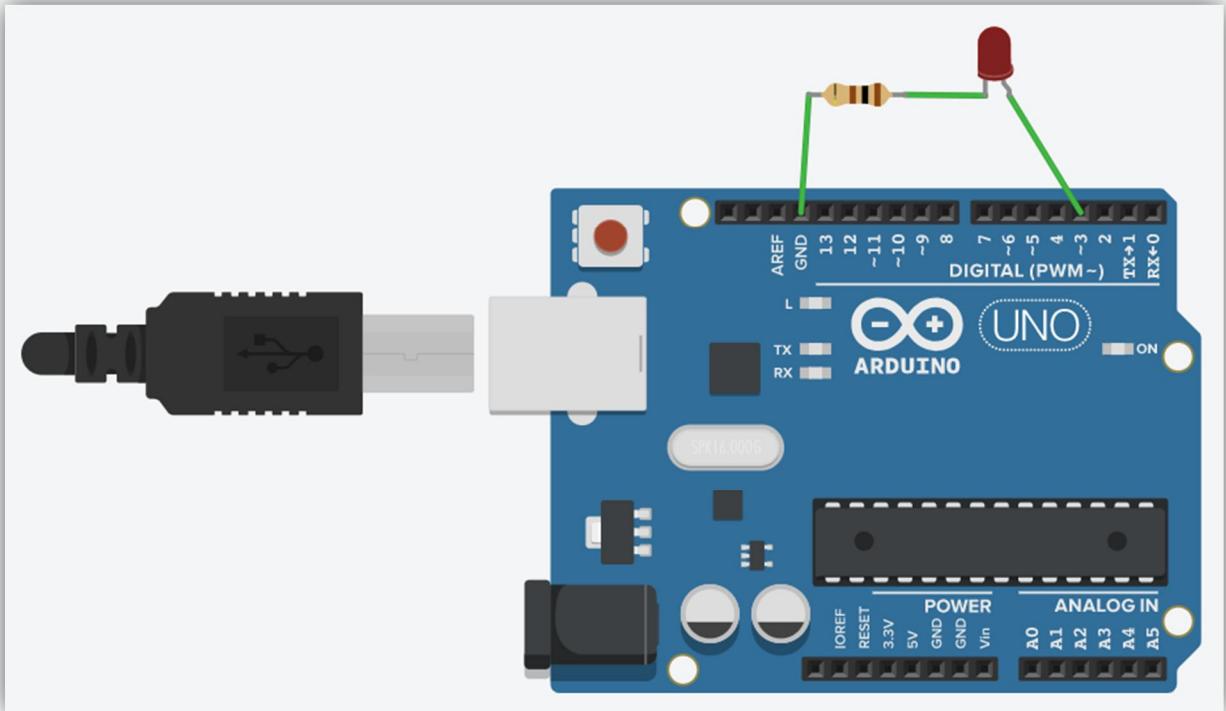
Мікроконтролер Arduino має 6 пінів, призначених для використання в режимі широтно-імпульсної модуляції (ШІМ). З них 4 працюють на частоті 500 Гц, а 2 - на частоті 1000 Гц. Період проходження імпульсів в цих режимах становить приблизно 2 мілісекунди.



Таким чином, використовуючи широтно-імпульсну модуляцію в Arduino, можна імітувати аналоговий сигнал на цифровому виході. Це дозволяє точно керувати імпульсами в часі і створювати сигнали з різною інтенсивністю, наприклад, для управління яскравістю світлодіода. Широтно-імпульсна модуляція є потужним інструментом для створення аналогових ефектів та керування різними пристроями у світі цифрової електроніки.

ПОРЯДОК ВИКОНАННЯ РОБОТИ.

1. Використовуючи сервіс **tinkercad.com** зберіть наступну схему:



Елементи, що використовуються: світлодіод червоного світіння і резистор опором 100 ом.

Наберіть запропонований скетч і виконайте його в емуляторі.

```
void setup()
{
  pinMode(3,OUTPUT);
}
void loop()
{
  analogWrite(3,255);
  delay(1000);
  analogWrite(3,128);
  delay(1000);
  analogWrite(3,50);
  delay(1000);
  analogWrite(3,0);
  delay(1000);
}
```


Функція `analogWrite()` в Arduino дозволяє генерувати аналоговий сигнал (сигнал ШІМ) на виході плати. Ця функція часто використовується для керування яскравістю підключеного світлодіода або для регулювання швидкості обертання ротора електродвигуна.

Після виклику `analogWrite()` на виході генерується прямокутна хвиля з вказаною шириною імпульсу. Ця хвиля залишається активною до того часу, поки не буде викликана нова функція `analogWrite()` з іншими параметрами або поки не буде викликана функція `digitalWrite()` або `digitalRead()` на тому ж порті.

Запропонований приклад дозволяє змінювати яскравість світіння світлодіода, яке матиме чотири значення: 255 – максимальна яскравість, 128 – яскравість в половину менша від максимальної, значення 50 – ще в половину менша і 0 – світлодіод не світиться.

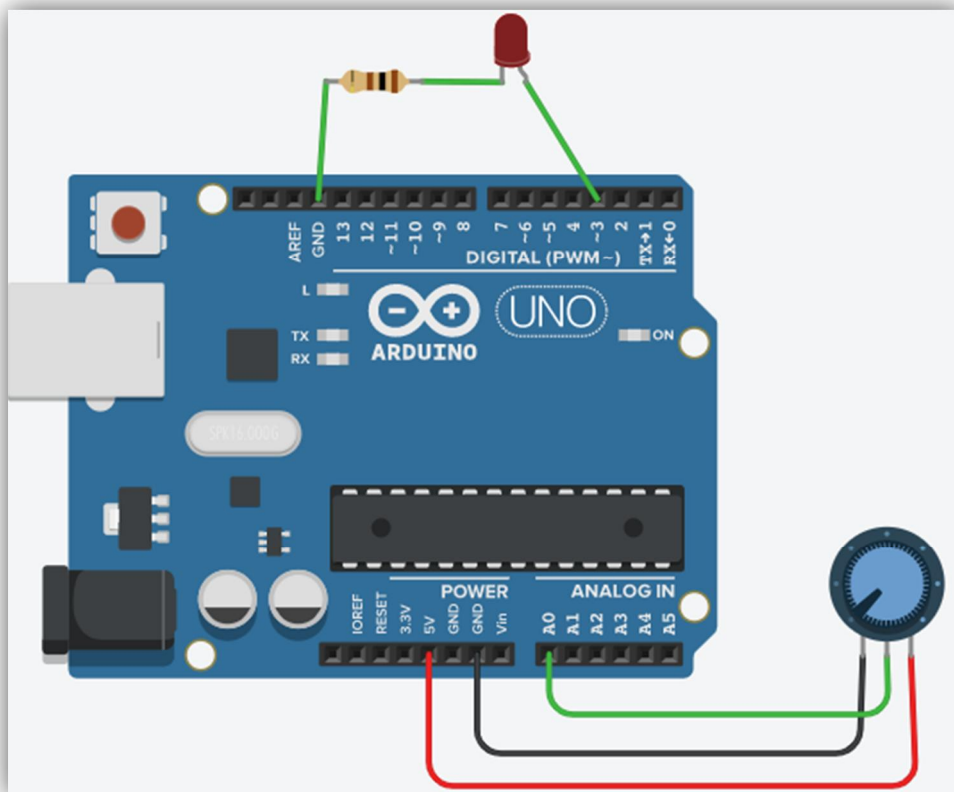
Поекспериментуйте з текстом програми, змінюючи значення, які відповідають за ШІМ.

Завдання: напишіть скетч в якому, використовуючи цикл плавно змінюється яскравість світіння світлодіода, значення ШІМ задавати у діапазоні від 0 до 255.

Завдання: напишіть скетч в якому, яскравість світіння світлодіода задається генератором випадкових чисел у діапазоні від 0 до 255.

Зберіть наступну схему:

Елементи, що використовуються: світлодіод червоного світіння, резистор опором 100 ом, резистор змінного опором на 100 ком.



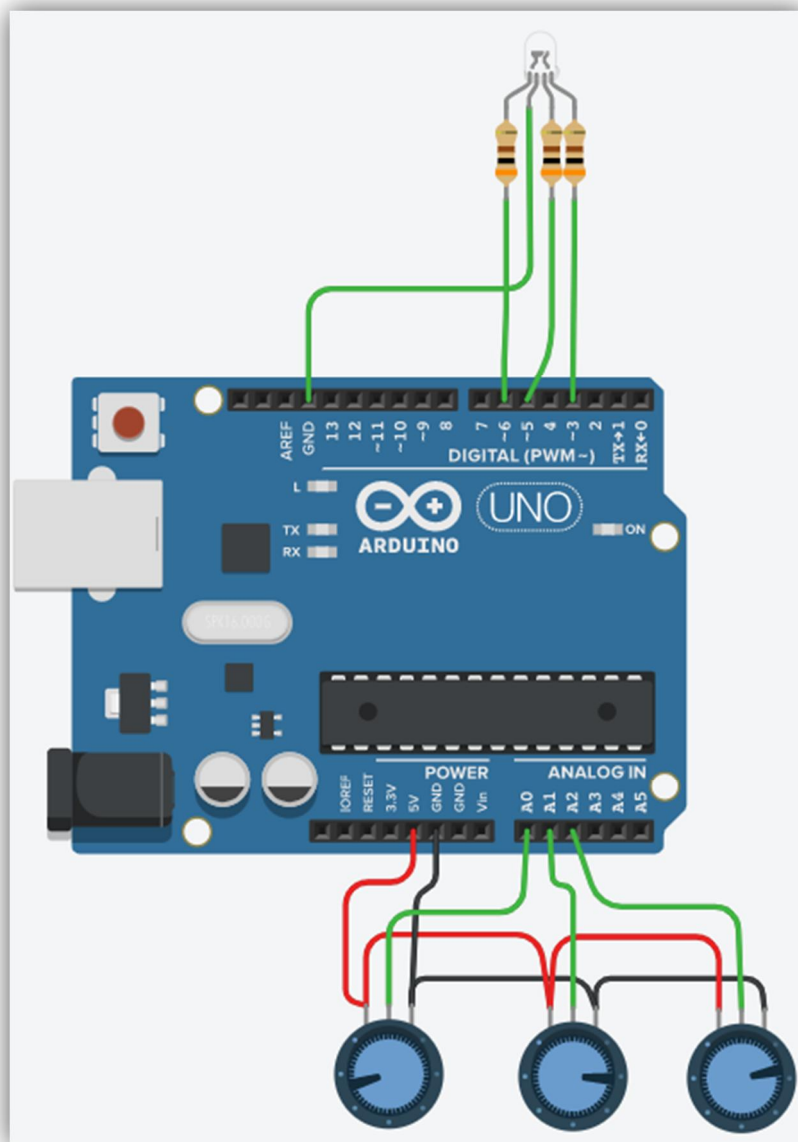
Завдання: напишіть скетч в якому, в залежності від положення ручки потенціометра (резистора зі змінним опором), буде змінюватися яскравість світіння світлодіода.

Зберіть наступну схему:

Елементи, що використовуються: RGB-світлодіод, резистор опором 300 ом кількістю 3 шт, резистор змінного опором на 100 ком у кількості 3 шт.



Призначення виводів RGB-світлодіода показано на рисунку



Завдання: проаналізуйте запропонований скетч в якому, в залежності від положення ручки потенціометра (резистора зі змінним опором), буде змінюватися яскравість світіння зеленого кольору RGB-світлодіода.

```
void setup()  
{
```

```

pinMode(3,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);

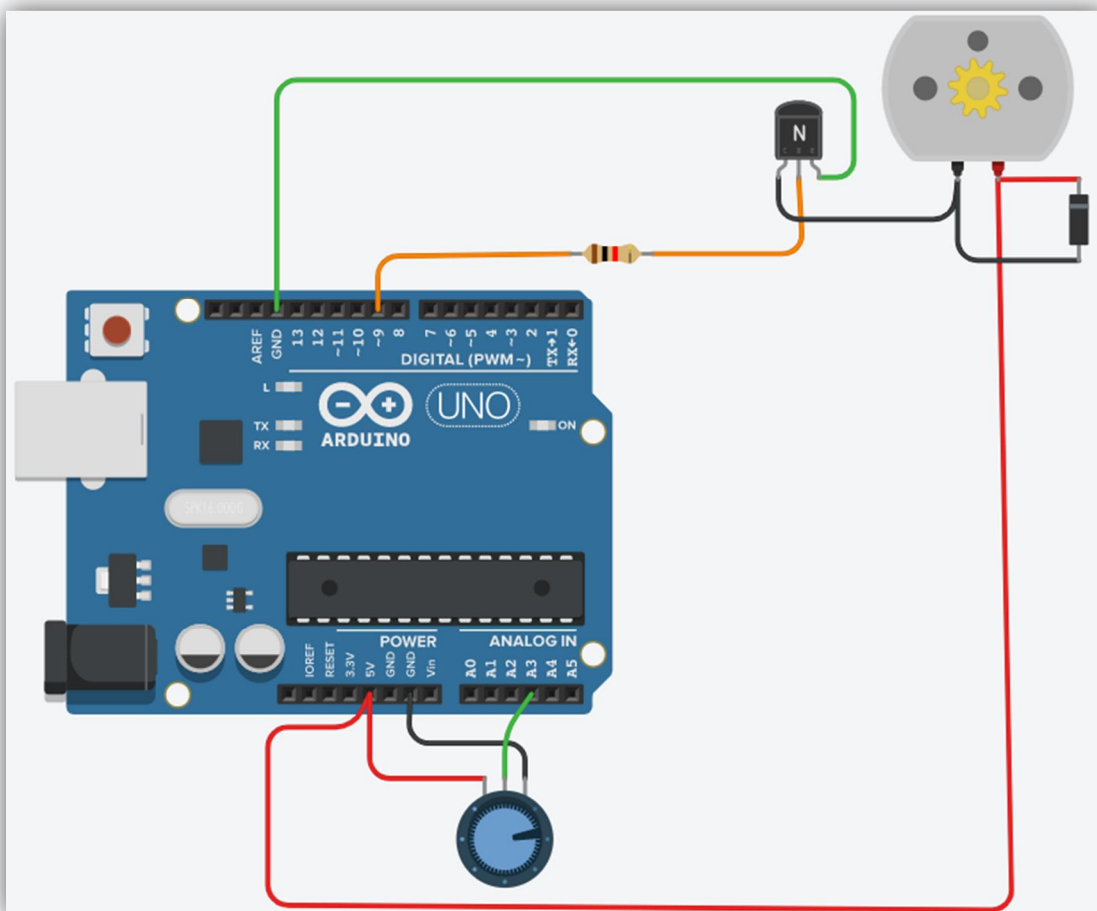
pinMode(A0,INPUT);
pinMode(A1,INPUT);
pinMode(A2,INPUT);
}
void loop()
{
  int g=analogRead(A0);
  g = map(g, 0, 1023, 0, 255);
  analogWrite(3,g);
  ...
  ...
}

```

Необхідно дописати скетч таким чином, щоб можна було іншими потенціометрами змінювати яскравість світіння RGB-світлодіода синім і червоним кольором.

Зберіть наступну схему:

Елементи, що використовуються: двигун постійного струму, діод, резистор змінного опору на 100 Ком, резистор з опором 1 Ком, N-P-N транзистор.



Завдання: проаналізуйте скетч з попереднього завдання і зменіть його так, щоб можна було керувати швидкістю обертання ротора в двигуні, використовуючи ШІМ.

ПИТАННЯ ДЛЯ ПІДГОТОВКИ.

1. Як конфігурувати порти входу/виходу?
2. Для чого використовують команду `map` ?
3. Що таке ШІМ?
4. Що собою являє транзистор?
5. Для чого використовується діод?
6. Для чого використовують ШІМ?
7. Як позначається потенціометр на схемі?
8. Як правильно підключити світлодіод на схемі?

Лабораторна робота №9 (Підключення різних елементів відображення інформації до плати Arduino)

Тема: Підключення різних елементів відображення інформації до плати Arduino.

Мета: Навчитися використовувати різні способи відображення інформації у власних розробках на платформі Arduino.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

Для того щоб відображати інформацію існує велика кількість електронних компонентів: лампи, світлодіоди, RGB- світлодіоди, 7-сегментні індикатори, світлодіодні матриці, адресні світлодіоди NeoPixel, рідкокристалічні індикатори. Елементи індикації вибирають в залежності від поставлених задач. Використовують електронні індикатори в різних приладах і пристроях для індикації станів, будь-якої інформації. Наприклад світлодіодні індикатори часто використовуються в якості елементів відображення стану для візуального контролю, світлодіодні табло для індикації тексту або картинок, рідкокристалічні індикатори – для відображення знаково-текстової інформації.

Індикатори за принципом створення зображення можна розділити на **активні і пасивні**.

Активні індикатори генерують видиме випромінювання при наявності на їх електродах електричної енергії. До них відносяться газорозрядні, катодолюмінесцентні, електролюмінесцентні порошкові і тонкоплівкові, розжарювані і напівпровідникові індикатори.

Пасивні - рідкокристалічні і електрохромні - модулюють тільки падаючий або проходячий крізь них зовнішній світловий потік при наявності на електродах електричного поля.

Активні індикатори всіх типів для забезпечення гарного контрасту зображення при великому зовнішньому освітленні витрачають електричну енергію значно більше, ніж пасивні, у яких яскравість і контраст зображення не залежить від зовнішньої освітленості при невеликому споживанні електроенергії.

Незалежно від характеру інформації, що відображається, індикатори за своїм призначенням можна розділити на три групи:

- індивідуального (*напівпровідникові і рідкокристалічні*);
- групового (*вакуумні розжарювані і люмінесцентні*);
- колективного користування.

За характером інформації, що відображається індикатори всіх типів діляться на:

- поодинокі (*окрема точка*);
- шкальні (*дискретно-аналогові прилади*);
- цифрові - однорозрядні (*з можливістю відображення арабських цифр від 0 до 9*);
- цифрові багаторозрядні (*з одночасним відображенням декількох цифрових і службових розрядів*),

- букви і цифри - однорозрядні і багаторозрядні;
- матричні (*відображають на одному знакомісць будь-який текст, графіки, символи*)
- мнемонічні.

Основні стандартизовані визначення, які носять загальний характер і застосовні до різних типів індикаторних приладів.

Знакосинтезуючий індикатор – це пристрій, який використовує один або кілька окремих елементів для візуального відображення інформації, призначеної для сприйняття органами зору.

Інформаційне поле – конструктивна частина знаковосинтезуючих індикатора, в межах якої можуть бути відображені інформації.

Елемент відображення інформації – конструктивна частина інформаційного поля індикатора, що має самостійне управління.

Сегмент – елемент відображення інформації знаковосинтезуючого індикатора, контуром якого є прямі або криві лінії.

Знакоместо – інформаційне поле знаковосинтезуючого індикатора або його частина, необхідна або достатня для відображення одного знака. Під знаком розуміється умовне позначення букв алфавіту, математичних знаків, знаків пунктуації, предметів, явищ, подій та ін.

Види знаковосинтезуючих індикаторів

Активний індикатор – індикатор, принцип дії якого заснований на перетворенні енергії електричного поля в світловий потік.

Пасивний індикатор – індикатор, принцип дії якого заснований на модуляції зовнішнього світлового потоку під дією електричного поля.

Розжарюваний вакуумний індикатор – активний індикатор, в якому використовується явище світіння тіл розжарювання у вакуумі.

Люмінесцентний вакуумний індикатор – активний індикатор, в якому використовується явище катодолюмінесценції.

Напівпровідниковий індикатор – активний індикатор, в якому використовується явище інжекційної електролюмінесценції.

Рідкокристалічний індикатор – пасивний індикатор, в якому використовується явище електрооптичного ефекту в рідкому кристалі.

Сегментний індикатор – індикатор, елементи відображення якого є сегментами, згрупованими в одне або кілька знакомісць.

Матричний індикатор – індикатор, елементи відображення якого згруповані по рядках і стовпцях.

Знакосинтезуючий екран – матричний знаковосинтезуючий індикатор без фіксованих знакомісць з числом елементів відображення не менше 10 000.

Одиничний індикатор – індикатор, що складається з одного елемента відображення і призначений для відображення інформації у вигляді точки або іншої геометричної фігури.

Цифровий індикатор – індикатор, призначений для відображення інформації у вигляді цифр.

Буквено-цифровий індикатор – індикатор, призначений для відображення інформації у вигляді букв, математичних знаків, знаків пунктуації.

Шкальний індикатор – індикатор, призначений для відображення інформації у вигляді рівнів або значень величин.

Мнемонічний індикатор – індикатор, призначений для відображення інформації у вигляді мнемосхеми або частини мнемосхеми.

Однорозрядний індикатор – індикатор, що має одне знакомісце. Однорозрядні індикатори можуть бути цифровими і буквено-цифровими.

Багаторозрядний індикатор – індикатор, що має кілька фіксованих знакомісць. Багаторозрядні індикатори можуть бути цифровими і буквено-цифровими.

Знакосінтезуючий модуль – індикатор, конструктивно виконання якого забезпечує можливість створення з'єднаних між собою індикаторів без втрати інформації в місцях стикування. Залежно від видів знакосінтезуючих індикаторів розрізняють шкальний, матричний, мнемонічний модулі та модуль екрану.

Складений індикатор – індикатор, конструктивно виконаний як єдине ціле з окремих модулів. Залежно від видів знакосінтезуючих індикаторів розрізняють складений шкальний, матричний і мнемонічний індикатор, складений екран.

Знакосінтезуючий індикатор з вбудованим керуванням – індикатор, конструктивно виконаний спільно з частиною елементів системи управління.

Електронний індикатор - це керований електронний прилад в закінченому корпусі, призначений для візуального інформування.

Класифікація електронних індикаторів.

Промисловість випускає безліч індикаторів різного типу і різноманітність постійно збільшується. Індикатори можна розділити за різними однаковими ознаками:

За принципом утворення зображення:

- Активні (*напівпровідникові, газорозрядні, катодолюмінісцентному, електролюмінісцентні, розжарюються*)
- Пасивні (*рідкокристалічні і електрохромніе*)

За призначенням:

- Індивідуальні;
- Групові;
- Колективного користування;

За характером інформації, що відображається:

- Поодинокі;
- Шкальні;
- Мнемонічні;
- Цифрові;
- Буквено-цифрові;

- Графічні.

За способом формування зображення:

- Знакомодельючі;
- Знакосинтезуючі (*сегментні і матричні*);

По виду інформаційного поля:

- Поодинокі;
- Шкальні;
- Мнемонічні;
- Однорозрядні знаковосинтезуючі (*сегментні і матричні*);
- Багаторозрядні знаковосинтезуючі (*сегментні і матричні*);
- Матричні без фіксованих знакомісць і графічні;

За способом управління:

- Мультиплексні (динамічні);
- Статичні;

За живлячої напруги:

- низьковольтні
- високовольтні

За кольором:

- одноколірні;
- багатобарвні;
- повнокольорові;

Також, індикатори розрізняються за способом передачі інформації:

- аналогові (яскравістю або відтінком кольору);
- дискретні (кількістю і сукупністю елементів);

Індикатори також мають різні характеристики:

- Висота знака
- Ширина знака
- Кут огляду
- діаграма спрямованості
- Яскравість елемента відображення інформації
- Яскравість (середня по площі)
- Нерівномірність яскравості елемента відображення інформації
- нерівномірність яскравості
- Яскравість власного фону
- Сила світла
- Середня сила світла елемента відображення
- Нерівномірність сили світла
- Ширина спектра випромінювання
- Довжина хвилі в максимумі випромінювання
- Час реакції пасивного індикатора
- Час релаксації пасивного індикатора

- час готовності
 - Гранична напруга рідкокристалічного індикатора
 - Напруга насичення рідкокристалічного індикатора
 - Критична частота напруги, що управляє рідкокристалічного індикатора
- Семисегментний індикатор.**

Цифрові 7-сегментні індикатори - інтегральні мікросхеми з діодних напівпровідникових світлодіодів, призначені для відображення цифрової та буквеної інформації.

Основними характеристиками семисегментних цифрових індикаторів є кількість розрядів в одному корпусі (*однорозрядні, двохрозрядні, трьохрозрядні і чотирирозрядні*), колір, що відображається і висота відображуваного знака.

У деяких цифрових індикаторах, окрім сегментів, які відображають цифри і букви, може бути одна або дві десяткові точки, які використовуються для відображення дробових чисел. Ці точки відомі як "дільник дробових чисел".

Цифрові індикатори зазвичай мають пластмасовий корпус, задня частина якого заливається епоксидним складом. Дисплей може бути чорним або сірим. Деякі індикатори додатково маркуються на корпусі з вказівкою серії та основних характеристик виробу.

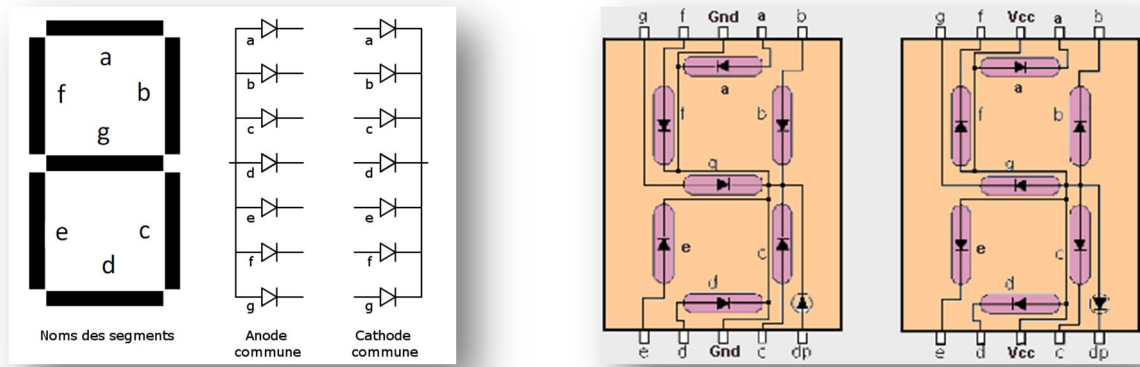


Для управління цифровими індикаторами необхідні спеціальні інтегральні мікросхеми, які перетворюють двійковий або двійковий-десятковий код в відповідний код індикатора, або підключаються до виводів мікроконтролера чи плати Arduino і керуються програмно.

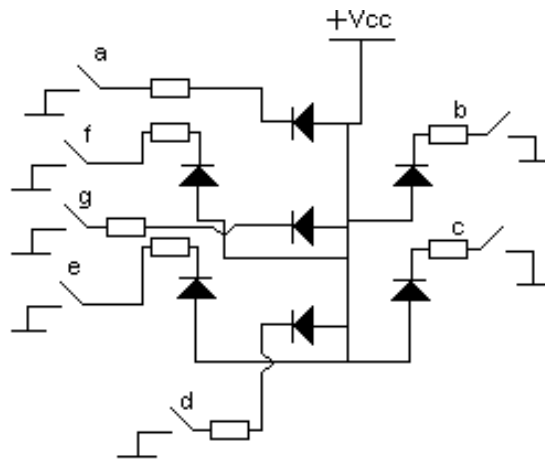
Цифрові індикатори застосовуються в якості світловипромінюючих джерел в різних вимірювальних приладах, пристроях автоматики і обчислювальної техніки та приладах побутової техніки.

Цифрові 7-сегментні індикатори виготовляються із загальним катодом та загальним анодом. При розробці приладу вибираються в залежності від схемотехніки і особливостями керування.

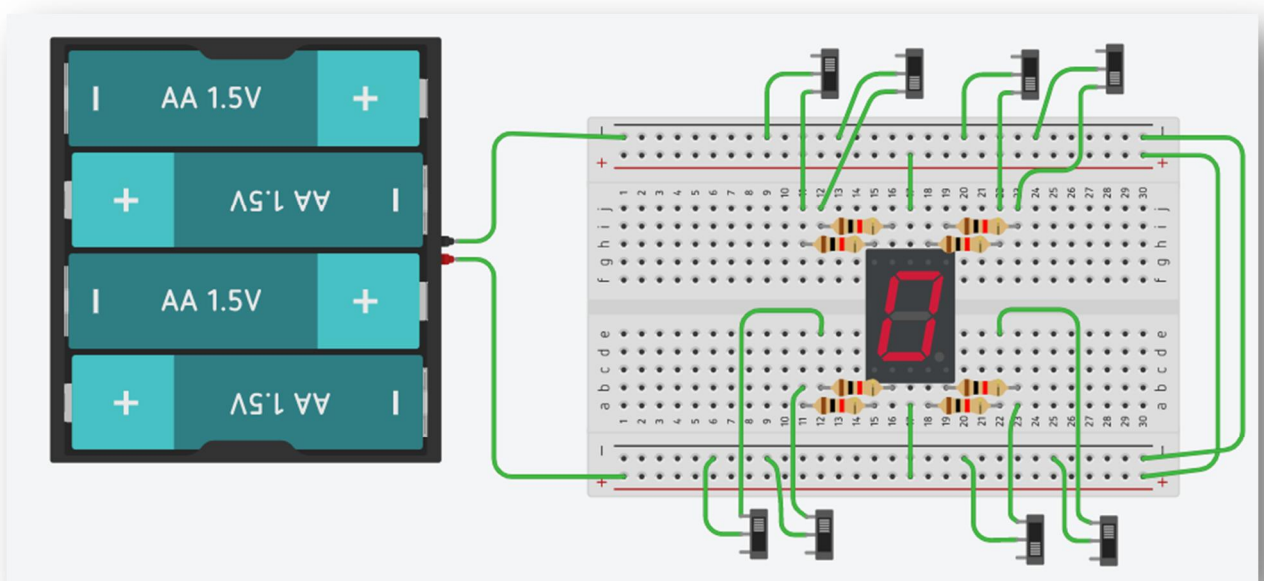
Як ми бачимо, семисегментний індикатор - це просто набір звичайних світлодіодів в одному корпусі та розташовані вісімкою і мають форму палички-сегмента.



Проста схема включення індикатора для керування його сегментами може виглядати так:

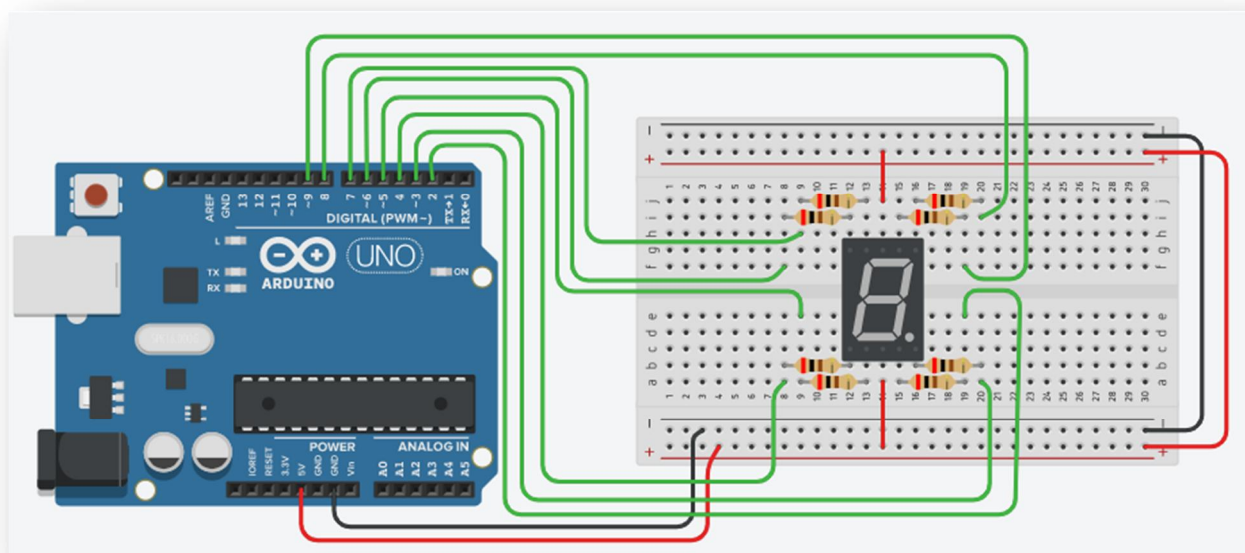


Для побудови подібної схеми (з використанням *крапки*) нам знадобяться: резистори опором 200 Ом у кількості 8 шт., перемикачі або кнопки – 8 шт., семисегментний індикатор (*загальний анод*) і джерело живлення.

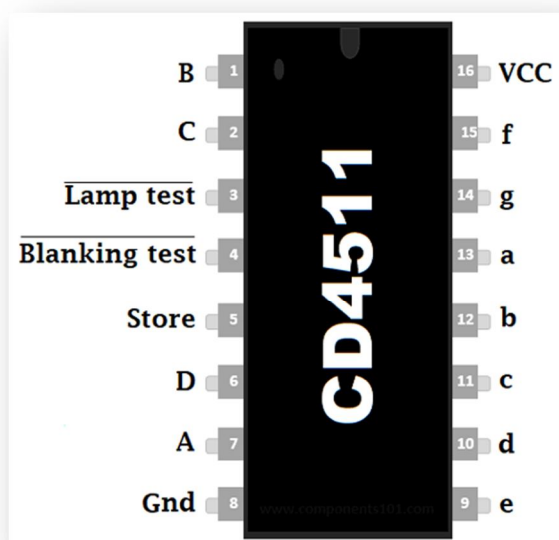


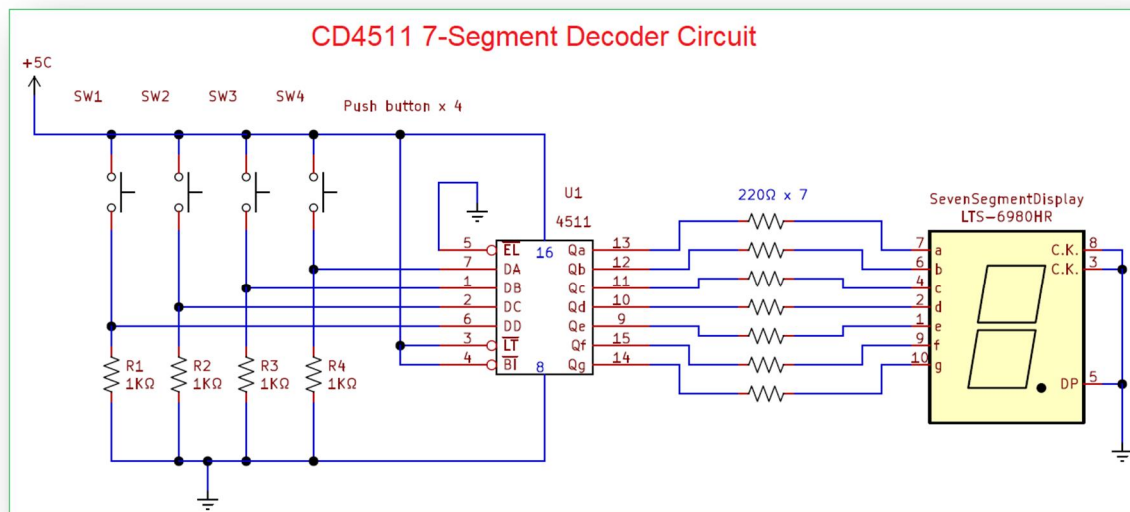
Зібравши дану схему можна керувати усіма сегментами індикатора, використовуючи відповідні перемикачі.

Семисегментний індикатор можна підключитися безпосередньо до Arduino, але тоді буде зайнято 7 контактів, а в програмі буде необхідно реалізувати алгоритм перетворення числа із двійкового подання до відповідних сигналів.

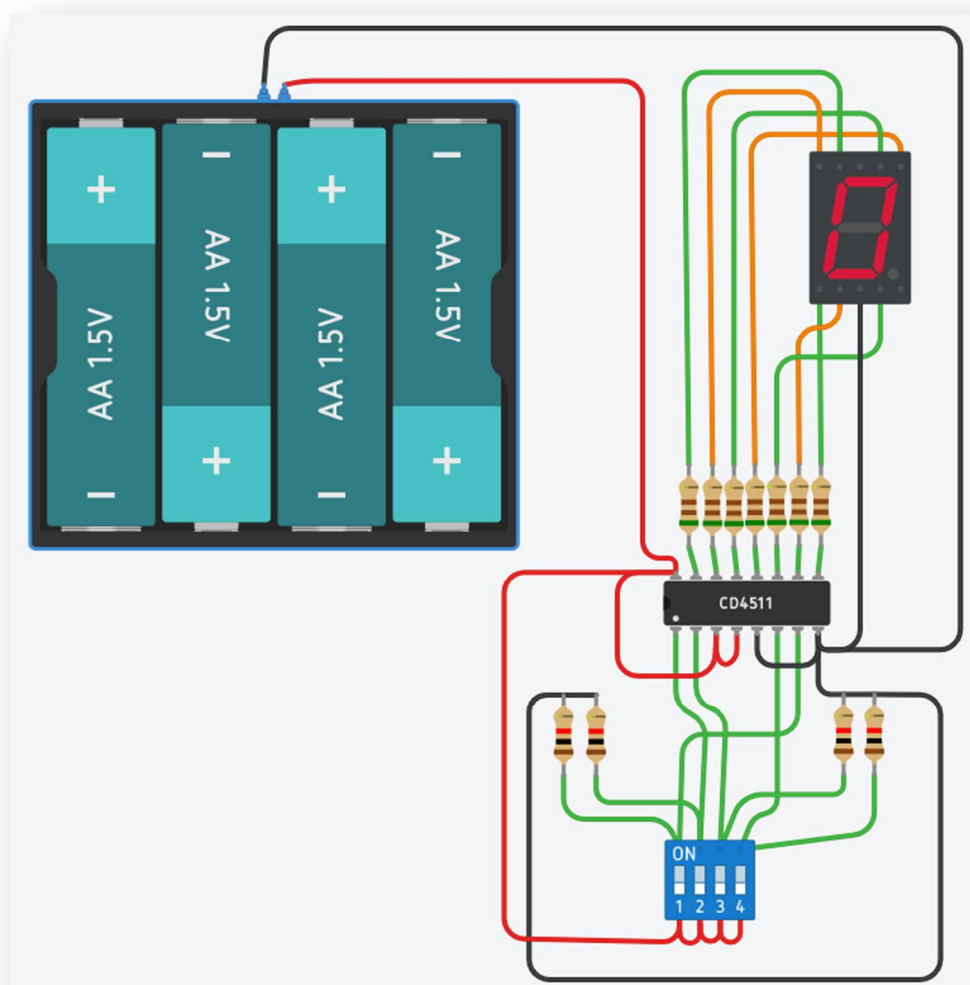


Для спрощення цієї задачі існують 7-сегментні драйвери, що дозволяють зменшити кількість пінів при підключенні до плати Arduino. Це проста мікросхема яка має виходи для підключення усіх сегментів (*a, b, c, d, e, f, g pins*) і входи для керування ними.





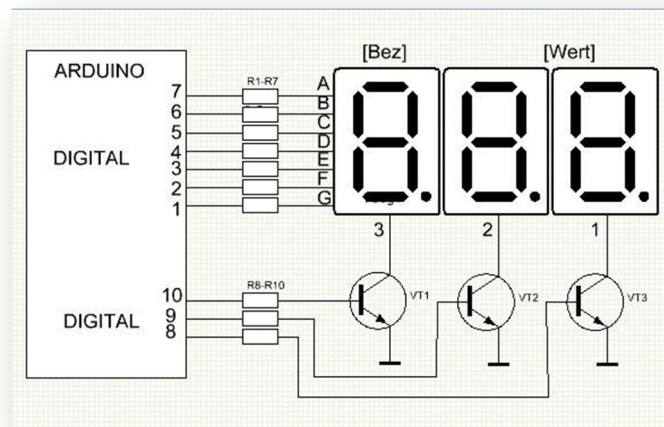
Підключення семисегментного індикатора відбувається за наступною схемою:



Для побудови цієї схеми необхідно: резистори 510 Ом – 7 шт., резистори 1 Ком – 4 шт., 7-сегментний індикатор (із загальним катодом), блок перемикачів, мікросхема драйвер CD 4511, джерело живлення на 6В. Після

того, як схема зроблена, можна спробувати за допомогою перемикачів відобразити на семисегментному індикаторі цифру від 0 до 9.

Для підключення двох і більше семисегментних індикаторів роблять наступне: одноімennі сегменти індикаторів з'єднують різом, а загальними виводами керують транзистори, так як зображено на схемі:



При роботі з семисегментними індикаторами виводити інформації можна як статично, так і динамічно.

1. Статична індикація.

Статична індикація - це найпростіший вид індикації, при якому кожен сегмент індикатора постійно перебуває в одному з двох станів - включений або виключений. Основною перевагою цього методу є те, що після виведення інформації, наприклад, в зсувний регістр, стан індикатора залишається незмінним, доки дані в цих регістрах не будуть змінені. Оскільки напруга на сегментах постійна, яскравість індикатора завжди максимальна.

Також існує декілька обмежень та недоліків, пов'язаних зі статичною індикацією. По-перше, для її реалізації потрібна велика кількість регістрів, де кожний розряд вимагає використання окремої мікросхеми. Крім того, потрібно встановлювати окремий резистор для кожного сегмента, що може призвести до значної кількості резисторів. Зазвичай використовують спеціальні драйвери, які не потребують окремих резисторів і мають більше виходів, але це може підвищити вартість проекту через витрати на ці драйвери.

І по-друге всі збірки, в яких від 3 і більше розрядів, йдуть із з'єднаними сегментами і статично їх використовувати просто не вийде. Тому використовують динамічну індикацію.

2. Динамічна індикація.

Принцип дії динамічної індикації полягає в послідовному включенні кожного семісегментного блоку посредством загальної схеми перетворення даних. У такому випадку всі дані надходять всього на один дешифратор, але в певній черговості. Такий дешифратор підключається до всіх елементів відразу,

а сигнал про те, який блок сегментів зараз потрібно використовувати надходить по керуючому виводу.

Тобто при динамічній індикації в певний момент часу працює тільки один семисегментний блок, проте, якщо перемикання робити з достатньою частотою, то мерехтіння індикаторів буде непомітно для людини.

При динамічній індикації сегменти запалюються по черзі. А за рахунок інерції очей здається, що індикатор горить постійно. З основних плюсів динамічної індикації – потрібно набагато менше зовнішніх елементів. Основний мінус - для неї постійно потрібна увага процесора.

Частота зміни сегментів вибирається зазвичай не нижче 50Гц. Краще використовувати частоти не кратні 50, інакше при штучному освітленні може спричинити мерехтіння.

Частота переривань розраховується як «Кількість розрядів» x «Частота оновлення». Так для 8 розрядів з частотою 60 Гц потрібно викликати переривання з $F = 8 \times 60 = 480$ Гц.

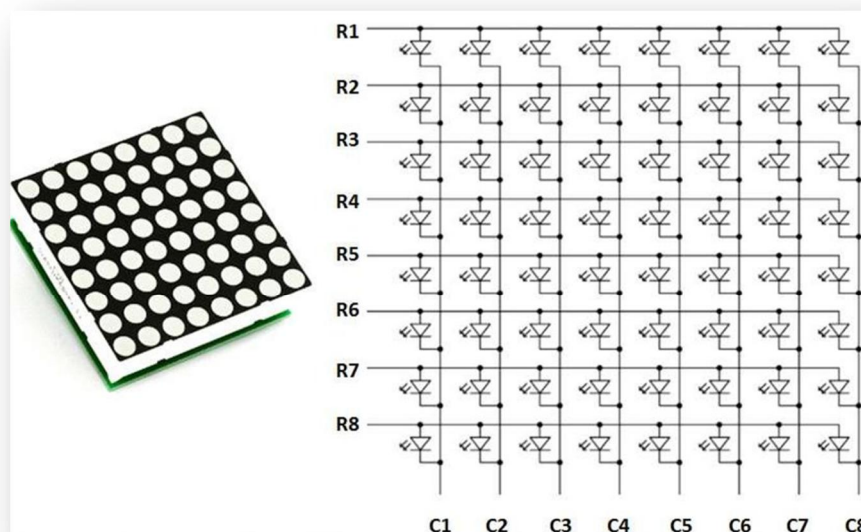
Є два види такої індикації – **поразрядна** і **посегментна**. Перша найбільш відома і популярна, друга краще підходить коли потрібно обробити велику кількість розрядів.

Поразрядна динамічна індикація це найвідоміший і популярний вид індикації. Вона полягає у тому, що в кожен момент часу горить тільки один розряд.

Посегментна динамічна індикація відрізняється від поразрядної індикації тим, що вона перебирає не розряди, а сегменти індикатора. Це означає, що в цьому методі кожен сегмент вмикається та вимикається окремо, незалежно від інших сегментів, і частота, з якою кожен сегмент загоряється, не залежить від кількості розрядів в індикаторі.

Матричний індикатор.

Матричний індикатор (*англ. dot-matrix display*) є певним видом знаковинтезуючого індикатора, де елементи відображення організовані у вигляді матриці, де є рядки та стовпці.



Цей тип індикатора використовується для відображення символів, спеціальних знаків та графічних об'єктів у різноманітних пристроях.

Матричний індикатор представляє собою пристрій, що має закінчений конструктивний корпус та обмежену кількість елементів індикації, організованих у вигляді матриці з рядків та стовпців. Ця назва виникла через схожість з математичним поняттям матриці.

Матричний індикатор складається з багатьох однотипних елементів, які можуть бути пікселями, і ці елементи організовані у рядки та стовпці. Такі індикатори можуть використовувати різні типи елементарних індикаторів-пікселів, такі як рідкокристалічні, світлодіодні, люмінесцентні, блінкерні тощо.

Матричні індикатори можуть бути розділені на два основних види залежно від принципу формування зображення та керування: статичні і динамічні, також відомі як мультиплексні.

У статичних матричних індикаторах, кожен піксель або елемент індикації має свій власний індивідуальний драйвер для керування напругою або струмом.

Для випадків, коли індикатор має невелику роздільну здатність, наприклад, 4×4 пікселі, використання статичного методу є економічно вигіднішим варіантом. Крім того, статичний метод корисний у пристроях, де необхідно знизити електромагнітні перешкоди, оскільки динамічні індикатори, через свій імпульсний спосіб керування, можуть створювати електромагнітні завади.

У випадку індикаторів, що використовують динамічний метод формування зображення, керування відбувається за допомогою рядків та стовпців. Виводи, які відповідають рядкам та стовпцям елементів індикації, пов'язані між собою. Для активації певного пікселя потрібно подати напругу на вивід рядка та вивід стовпця. При цьому взаємодіючи на перетині цих двох координат, формується відображення точки на екрані.

Також можливо одночасно увімкнути кілька пікселів, які належать до одного і того ж стовпця або рядка. Наприклад, якщо на виводи одного стовпця світлодіодного матричного індикатора подається робоча напруга, а на контакти певних рядків подається необхідний керуючий струм, то одночасно активуються декілька конкретних пікселів, що належать цьому стовпцю. За рахунок швидкої динамічної зміни активованих рядків (стовпців) та інерції людського зору, або інерції самих елементів індикації, формується зображення, яке сприймається як ціла картинка.

Такий метод дозволяє значно зменшити кількість необхідних виводів для керування світлодіодами. Наприклад, якщо у світлодіодній матриці розміром 8x8 є 64 світлодіоди, то для їх керування потрібно було б 64 порти вводу/виводу, по одному для кожного пікселя індикатора. Але якщо аноди всіх світлодіодів у рядках об'єднати разом, а катоди в стовпцях, то кількість необхідних портів вводу/виводу зменшиться до 16.

Управління багатобарвними індикаторами, такими як напівпровідникові двоколірні індикатори з червоним і зеленим світлодіодами, потребує врахування особливостей кожного кольору світіння. В кожному пікселі такого індикатора є як мінімум два світлодіоди – червоний і зелений. Керування

кольором світіння виконується шляхом зміни співвідношення або середнього струму, який подається через червоний і зелений світлодіоди, при фіксованому часі горіння або часу світіння при стабільному струмі.

Управління матричним індикатором відбувається за принципом мультиплексії для формування зображення. Цей процес полягає у виборі одного із рядків і подачі сигналу (коду), який активує конкретні пікселі в цьому рядку. Потім вибирається наступний рядок і процес повторюється. Напруга на вивід індикатора подається за допомогою схеми керування – драйвера.

Так, односимвольні матричні індикатори широко використовуються у різних пристроях, і найпоширеніші розміри піксельних матриць складають 5x7, 5x8 та 8x8 пікселів. Вони надають можливість виводити текст, числа та інші символи на екрані, і їх роздільна здатність дозволяє зображати символи чітко та читабельно.

Індикатори можуть бути як монохромними, що відображають зображення використовуючі елементи одного кольору, так і кольоровими. У кольорових індикаторів кожен піксель складається з двох або трьох окремих кольорових елементів (зазвичай червоного, зеленого та синього, *RGB*), і керування кожним кольором здійснюється окремо. Це дозволяє створювати барвисті та більш динамічні зображення та індикацію на дисплеях.

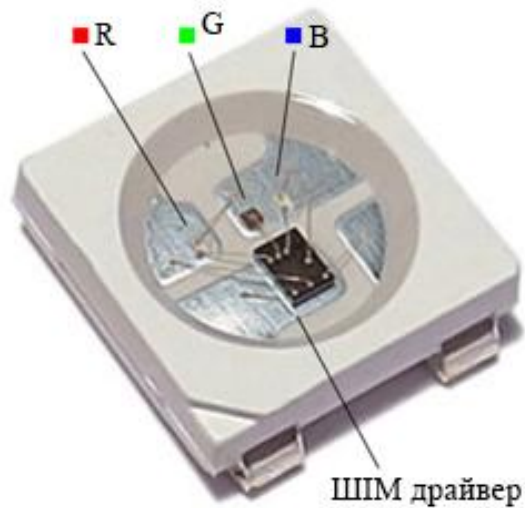
Адресний світлодіод та адресна стрічка.

Адресний світлодіод – це джерело світла з інтелектуальним керуванням, в який вбудовані схема управління і мікросхема *RGB* в одному корпусі. Має вбудований інтелектуальний цифровий порт для передачі даних і підсилювач сигналу. Фізично в адресному світлодіоді є 3 випромінюючих світлодіода (червоний, синій і зелений) і ШІМ-драйвера, які керують їх яскравістю.

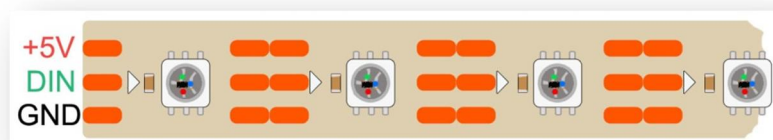
Драйвери ШІМ мають роздільну здатність в 8 біт, що означає, що для кожного з трьох кольорів доступно 256 різних рівнів яскравості. Отже, для встановлення яскравості кожного світлодіода потрібно передавати 24 біти інформації (або 3 байти). Це надає можливість створити індивідуальний та унікальний відтінок кольору на кожному світлодіоді.

Це дозволяє отримувати широкий спектр кольорів і створювати деталізовані та різноманітні зображення на таких дисплеях, оскільки можна контролювати яскравість і кольори кожного пікселя окремо. Такий підхід особливо важливий у відеоіграх, графічних додатках, візуалізаціях та інших сценаріях, де важлива точна кольорова і яскравісна інформація.

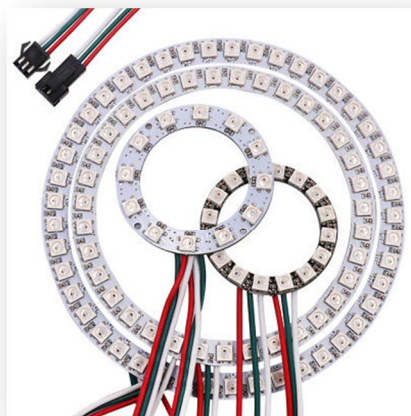
Протокол передачі інформації для керування світлодіодом є однолінійним і використовує фіксовану швидкість передачі. Інформацію про рівень яскравості кодують за допомогою тривалості високого та низького рівнів сигналу в одному каналі передачі.

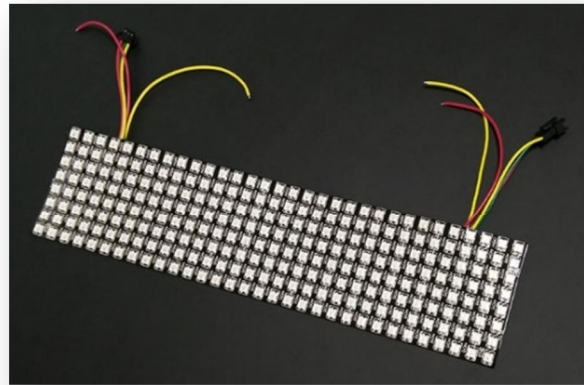
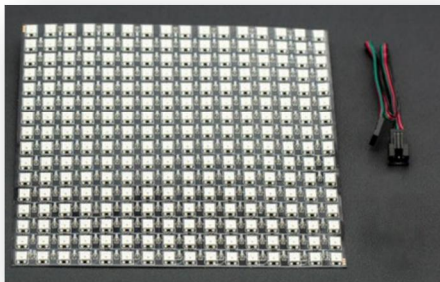
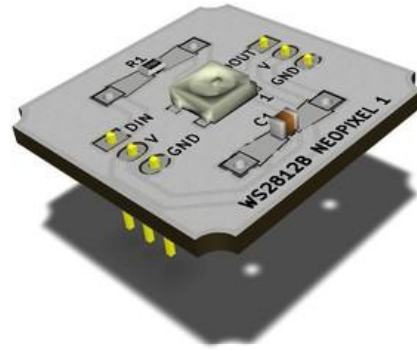
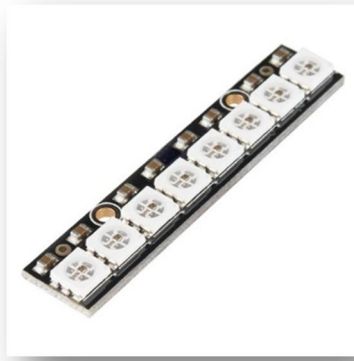


Адресна світлодіодна стрічка, яка складається із адресних світлодіодів, на відміну від звичайної RGB стрічки, має можливість засвітити будь-який світлодіод в стрічці будь-яким кольором завдяки того, що кожен світлодіод в стрічці має свою унікальну адресу, по якій драйвер звертається до нього за допомогою трехбітної команди. Команди подаються у лінію послідовно, для цього слугує третій провід на стрічці «DATA INPUT».

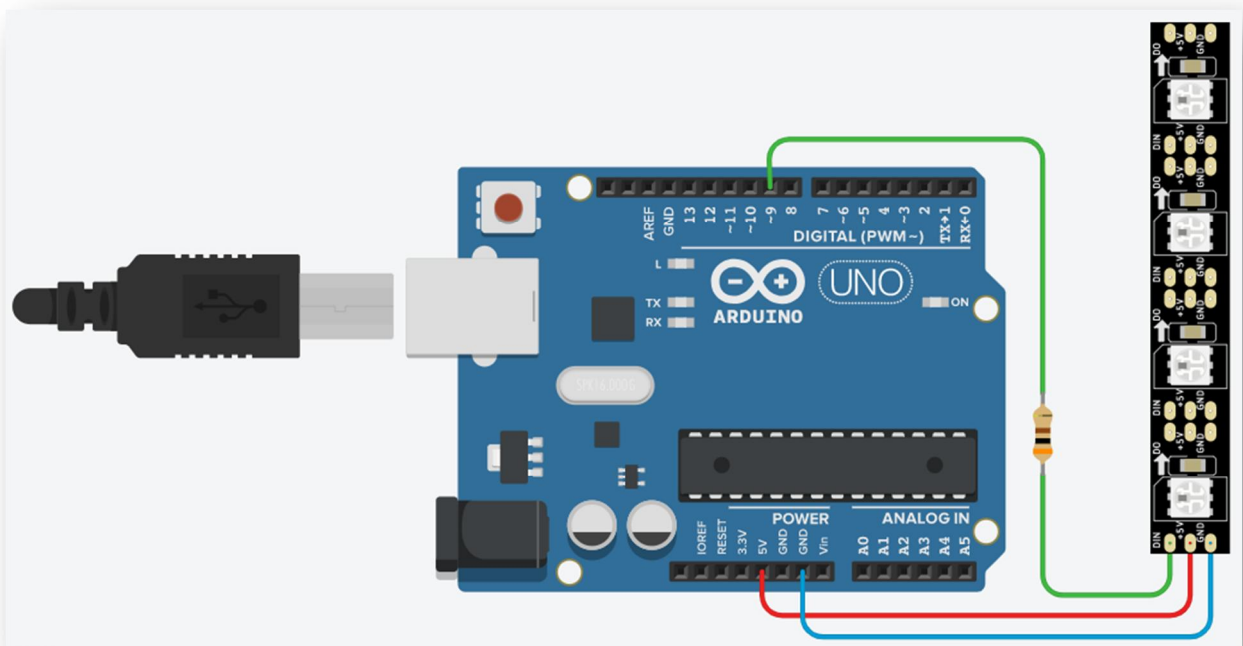


На основі адресних світлодіодів WS2812 и WS2811 випускається багато різних модулів:

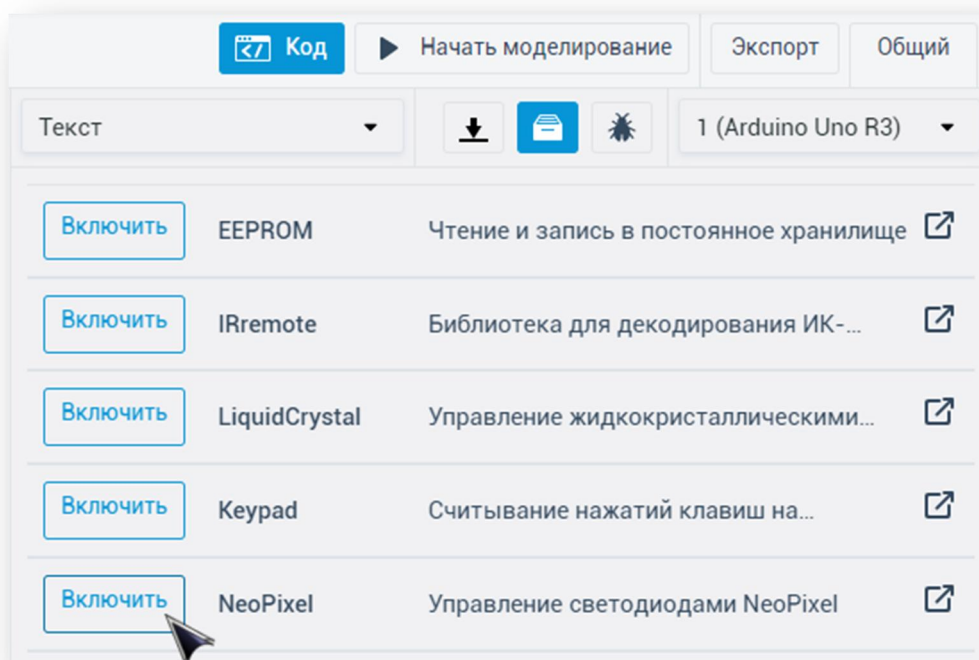




Тема-модуль NeoPixel – це 4 RGB-світлодіоди з вбудованими драйверами в одному корпусі, встановлені на одному модулі. Ці модулі можна з'єднувати в стрічки NeoPixel, використовуючи всього одну лінію даних, а їх «адресність» дозволяє керувати кольором кожного світлодіода на одній лінії.



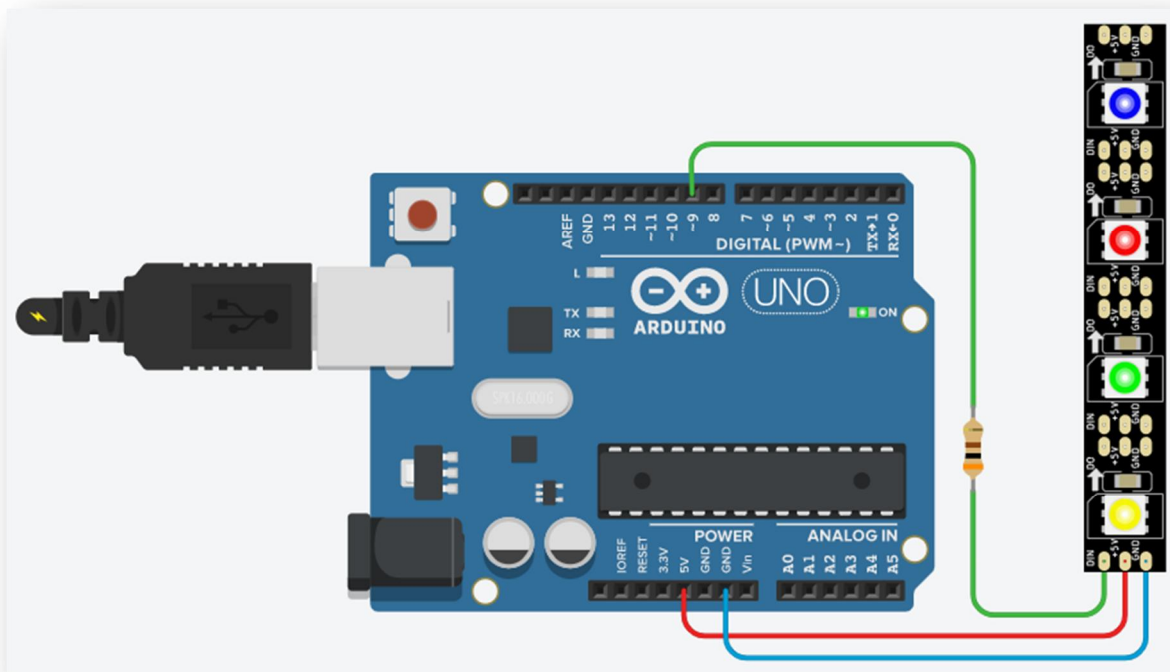
Для того, щоб можна було працювати з модулями NeoPixel в онлайн-сервісі **tinkercad.com** треба встановити бібліотеку **Adafruit_NeoPixel.h**



Приклад скетча роботи з модулями NeoPixel

```
#include <Adafruit_NeoPixel.h>
#define PIN 9 // номер пина до якого підключено модуль
#define count_led 4 // кількість світлодіодів в модулі
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(count_led, PIN,
NEO_GRB + NEO_KHZ800);
void setup() {
  pixels.begin();
  pixels.show(); // вибираємо для усіх світлодіодів стан -
  // вимкнено
}
void loop() {
  pixels.setPixelColor(0, pixels.Color(250,250,0)); // назначаемо
  // для нульового світлодіода колір «жовтий»
  pixels.setPixelColor(1, pixels.Color(0,250,0)); // назначаемо для
  // першого світлодіода колір «зелений»
  pixels.setPixelColor(2, pixels.Color(250,0,0)); // назначаемо для
  // другого світлодіода колір «червоний»
  pixels.setPixelColor(3, pixels.Color(0,0,250)); // назначаемо для
  // третього світлодіода колір «синій»
  pixels.show();
  delay(1000);
}
```

Після запуску скетча отримаємо світіння світлодіодів таку, як на малюнку:



Рідкокристалічний дисплей.

Рідкокристалічний дисплей (англ. *liquid crystal display (LCD)*), є електронним засобом візуального відображення інформації, який використовує рідкі кристали для контролю прохідності світла. LCD-дисплеї використовуються в різноманітних пристроях, таких як телевізори, монітори, смартфони, ноутбуки та інші.

Рідкокристалічний дисплей складається зі шарів рідкого кристалу, які можуть бути впорядкованими за допомогою електричних полів. Коли рідкі кристали впорядковані, вони можуть керувати прохідністю світла через дисплей, що призводить до відображення зображення або тексту на екрані.

Одні з переваг рідкокристалічних дисплеїв включають високу якість зображення, ефективність енергоспоживання та тонку конструкцію, що дозволяє їх використовувати у різних типах пристроїв.

Рідкокристалічні комірки керуються за допомогою транзисторів тонкої підкладки (*TFT* — *аббревіатура англійського виразу «Thin Film Transistors»*), які вмикають або вимикають кожну комірку шляхом регулювання напруги, що подається на цю комірку. Цей метод керування дозволяє точно регулювати прозорість кожної комірки, що робить можливим відображення зображень або тексту на дисплеї.

Однією з головних переваг рідкокристалічних дисплеїв є їх низьке споживання енергії, що робить їх ідеальними для використання в різних портативних пристроях, таких як годинники, мобільні телефони, кишенькові комп'ютери, а також у великих пристроях, включаючи комп'ютерні монітори і телевізори.

Символьні монохромні LCD дисплеї широко використовуються і доступні в різних конфігураціях, з різною кількістю символів у рядку і рядків. Виготовляються дисплеї з розмірами: 8x2, 12x2, 16x1, 16x2, 16x4, 20x2, 20x4,

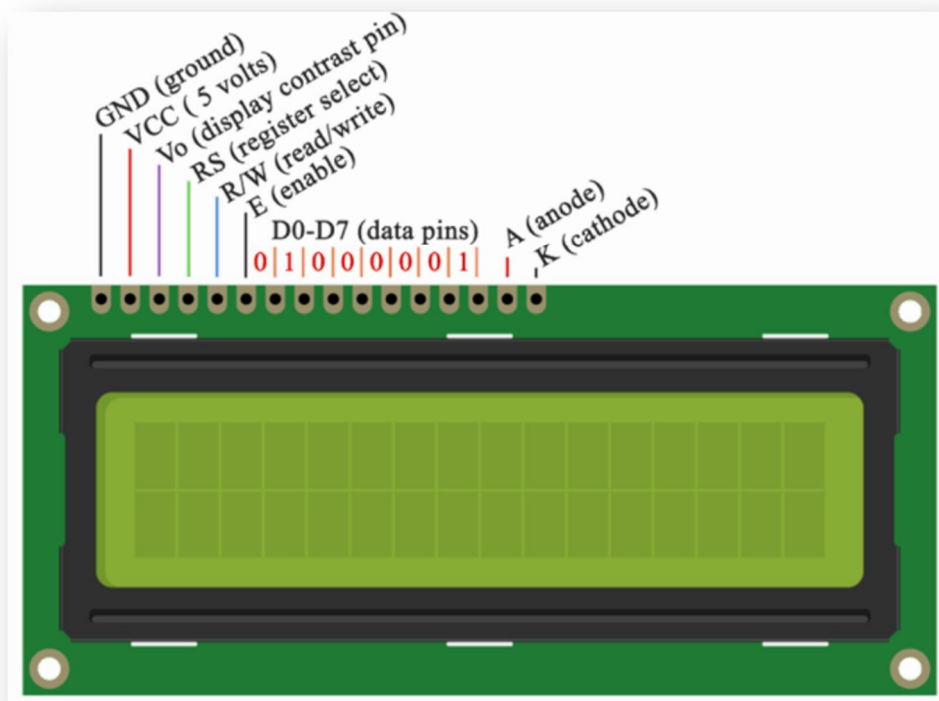
24x2 до 40x4. Перша цифра в їхній специфікації вказує на кількість символів у рядку, а друга – на кількість рядків. Це дозволяє вибрати дисплей, який найкраще відповідає конкретним потребам.

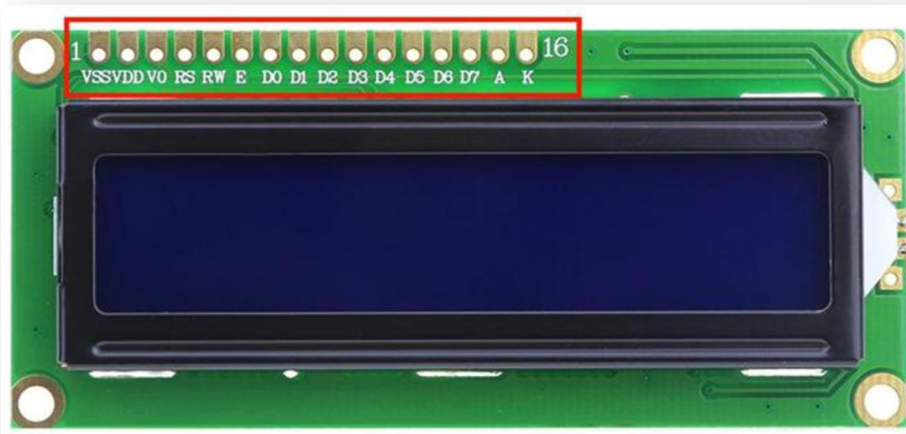
Крім того, існують різні варіанти LED підсвічування для цих дисплеїв, такі як білий, червоний, синій, зелений, жовтий, RGB та варіанти без підсвічування. Вибір кольору підсвічування може додати ефектності та функціональності вашому проекту.



Існують LCD дисплеї з різними кутами огляду та різними варіантами шрифтів. Символьні LCD модулі можуть використовуватися в промислових і побутових приладах, включаючи охоронні системи, медичне обладнання, автомобільне і домашнє аудіо, ігрові автомати, іграшки та інше.

Підключення будемо розглядати на дисплеї типу 1602. У першу чергу треба звернути увагу на підписи виводів LCD дисплея. Зустрічається два варіанти, нумерації. На двох малюнках нижче все нормально - від 1 до 16 виводу.

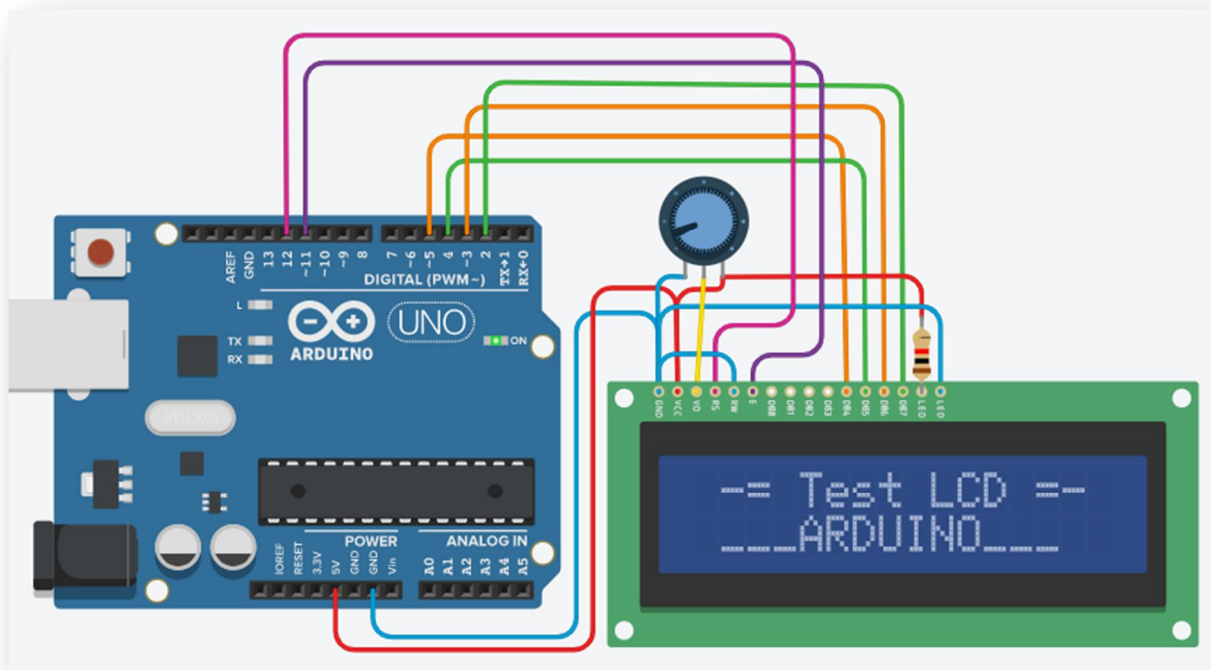
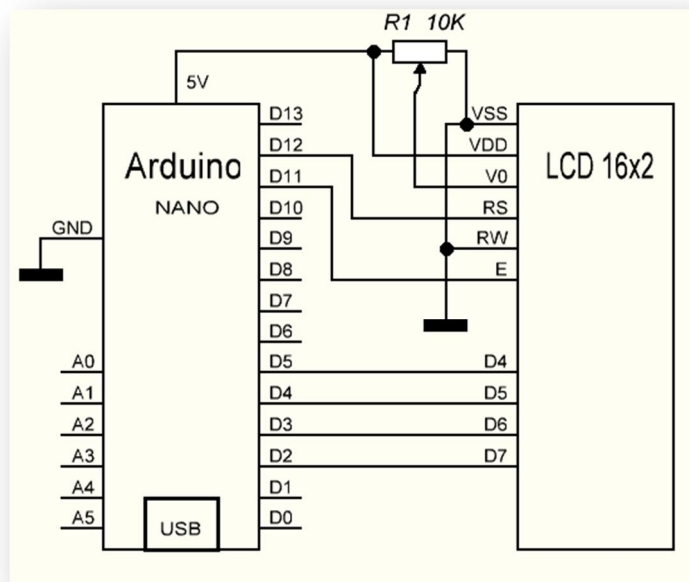




Вивід VSS означає Ground (GND). В іншому призначення виводів ідентичні. Але часто можна зустріти і нестандартну нумерацію:



Стандартна схема підключення LCD дисплея безпосередньо до плати Ардуіно виглядає так:



Для роботи роботи з LCD дисплеєм необхідно підключити спеціалізовану бібліотеку **LiquidCrystal.h**, після цього будуть доступні процедури і функції для роботи з LCD.

Приклад скетча роботи з LCD дисплеєм:

```
#include <LiquidCrystal.h> /* підключаємо бібліотеку для дисплея
LCD 16x2 */
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); /* вказуємо піни до яких
підключено дисплей: RS(12),E(11),D4(5),D5(4),D6(3),D7(2) */
```

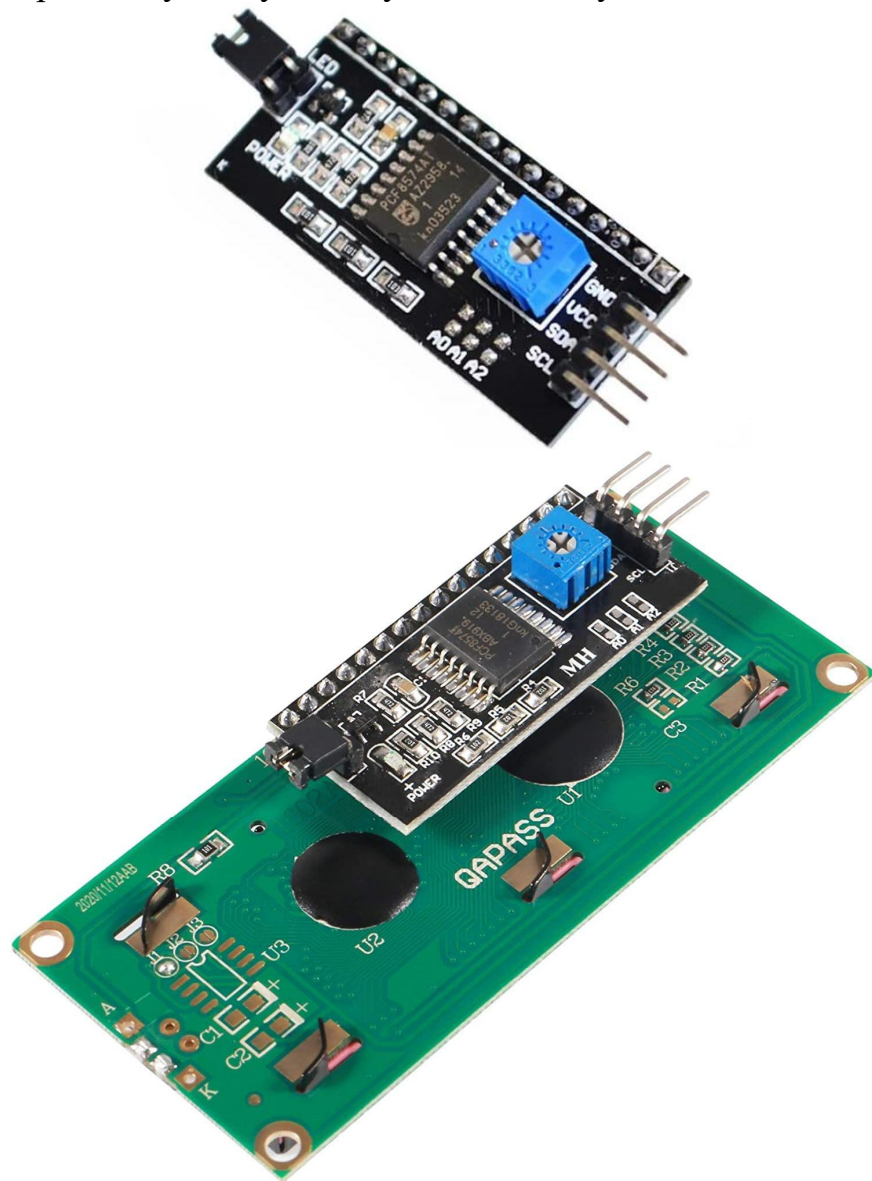
```
void setup() {
  lcd.begin(16, 2); // вказуємо тип дисплея LCD 16X2
```

```

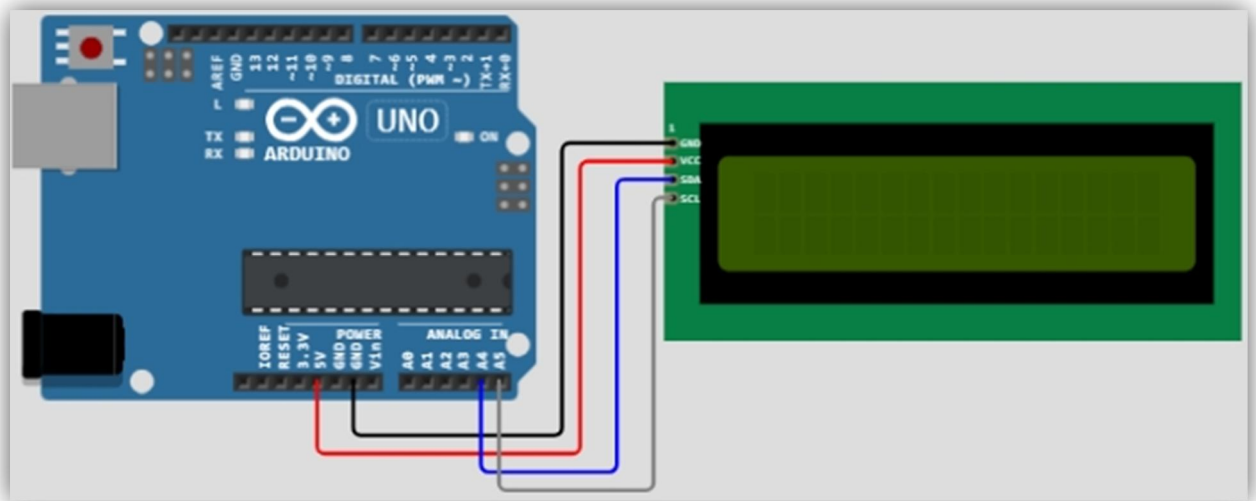
}
void loop() {
  lcd.setCursor(1,0); // встановлюємо курсор - нульовий рядок,
перший символ
  lcd.print("-- Test LCD =="); // виведення тексту на екран LCD
  lcd.setCursor(1,1); // встановлюємо курсор - перший рядок,
перший символ
  lcd.print("__ARDUINO__"); // виведення тексту на екран LCD
  delay(500); // пауза 500 мс
}

```

При звичайному підключенні LCD дисплея необхідно використовувати багато портів на платі Arduino і це не зовсім зручно під час створення складних проєктів. Але вихід є. LCD дисплей можна підключити по шині I2C за допомогою додаткового модуля. Такий модуль підключається до дисплея, перетворюючи паралельну шину зв'язку в послідовну I2C.



Підтримка такого модуля є в онлайн-сервісі **Wokwi** (<https://wokwi.com/>). Перед написанням скетчу нам необхідно завантажити бібліотеку LCD (LiquidCrystal_I2C). Підключення дисплея відбувається за наступною схемою:



```
#include <LiquidCrystal_I2C.h>

#define I2C_ADDR    0x27
#define LCD_COLUMNS 16
#define LCD_LINES   2

LiquidCrystal_I2C lcd(I2C_ADDR, LCD_COLUMNS, LCD_LINES);

void setup() {
  // Init
  lcd.init();
  lcd.backlight();

  // Print something
  lcd.setCursor(3, 0);
  lcd.print("Hello!");
  lcd.setCursor(2, 1);
  lcd.print("Test I2C_LCD");
}

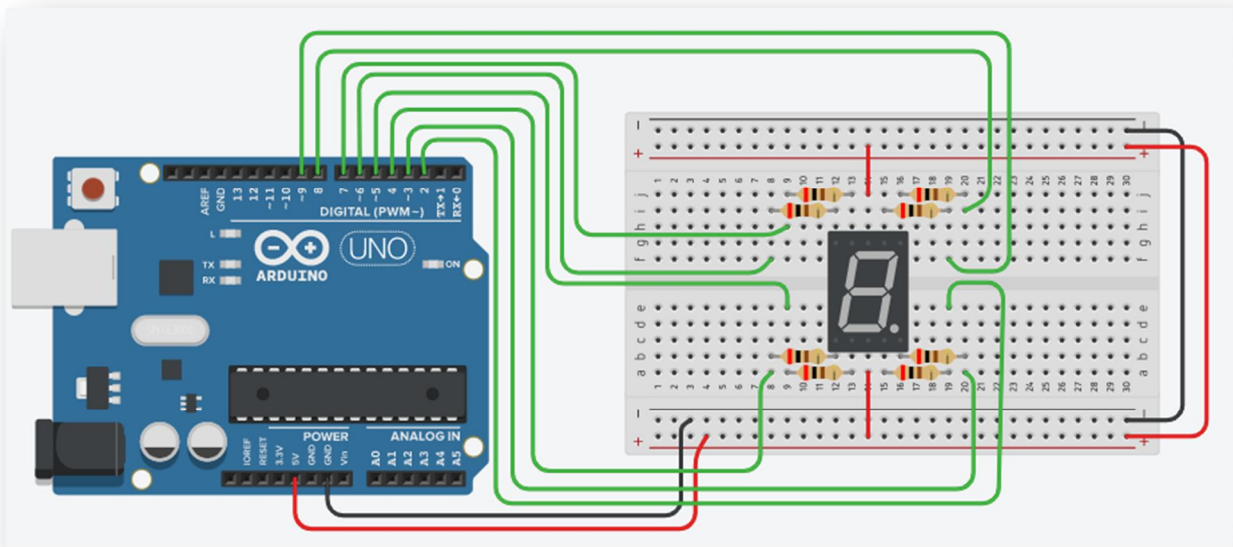
void loop() {
}
```

ПОРЯДОК ВИКОНАННЯ РОБОТИ.

1. Використовуючи пошукові сервіси інтернету знайдіть більш детальну інформацію, щодо різних елементів відображення інформації та індикації (знайдіть відповіді на наступні запитання):

- Як підключити RGB світлодіод до плати Arduino?
- Як підібрати резистор для підключення світлодіода?
- Як підключити світлодіодну матрицю до плати Arduino?
- Які бувають LCD дисплеї?
- Як підключаються LCD дисплеї до плати Arduino?
- Яка використовується бібліотека для роботи з LCD дисплеями?
- Як можна зменшити кількість проводів при підключенні LCD?
- Для чого можна використовувати модулі NeoPixel?
- Яка бібліотека використовується для роботи з модулями NeoPixel?
- Чим відрізняється звичайна RGB стрічки від стрічки із адресними світлодіодами?

2. Використовуючи сервіс **tinkercad.com** зберіть наступну схему:

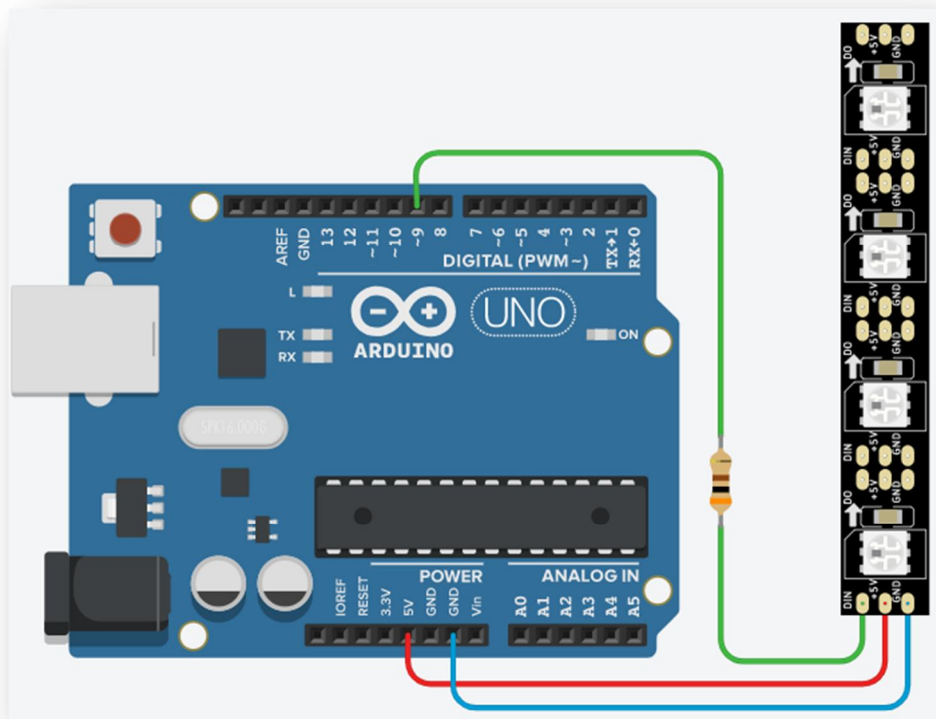


3. Напишіть скетч, при запуску якого на семисегментному індикаторі по черзі зображаються числа від 0 до 9.

4. Додайте до схеми дві кнопки і змініть скетч таким чином, щоб при натисканні однієї кнопки число на індикаторі збільшувалося, а при натисканні другої кнопки – зменшувалося.

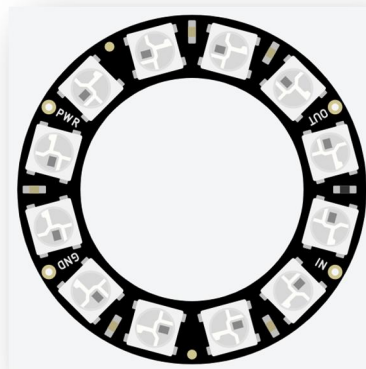
5. Підключіть семисегментний індикатор до плати Arduino, використовуючи мікросхему драйвер CD 4511 і напишіть скетч, при запуску якого на семисегментному індикаторі по черзі зображаються числа від 0 до 9.

6. Зберіть наступну схему:



7. Напишіть скетч, який при запуску по черзі засвічує усі світлодіоди основними кольорами – червоним, зеленим, синім. Написаний скетч прокоментуйте.

8. Заменіть лінійний модуль NeoPixel на кільцевий:



9. Використайте запропонований скетч:

```
#include <Adafruit_NeoPixel.h>
#define PIN 9
#define NUMPIXELS 12
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN,
NEO_GRB + NEO_KHZ800);
int delayval = 10;
byte LEDx = -1;
int redColor = 0;
int greenColor = 0;
int blueColor = 0;
bool F = 1;
```

```

void setup() {
  pixels.begin();
  setColor();
}
void loop() {
  LEDx++;
  for(int i=0;i<NUMPIXELS;i++){
    F = 1;
    if (LEDx > NUMPIXELS){LEDx = 0; setColor();}
    if(i==LEDx && F == 1){
      F = 0;
      pixels.setPixelColor(i, pixels.Color(redColor, greenColor,
blueColor));
pixels.show();
}
if (i == LEDx-1){
  F = 0;
  pixels.setPixelColor(i, pixels.Color(redColor-2, greenColor-2,
blueColor-2));
}
if (i == LEDx-2){
  F = 0;
  pixels.setPixelColor(i, pixels.Color(redColor-5, greenColor-5,
blueColor-5));
}
if(F == 1){
  F = 0;
  pixels.setPixelColor(i, pixels.Color(0, 0, 0));
}
delay(delayval);

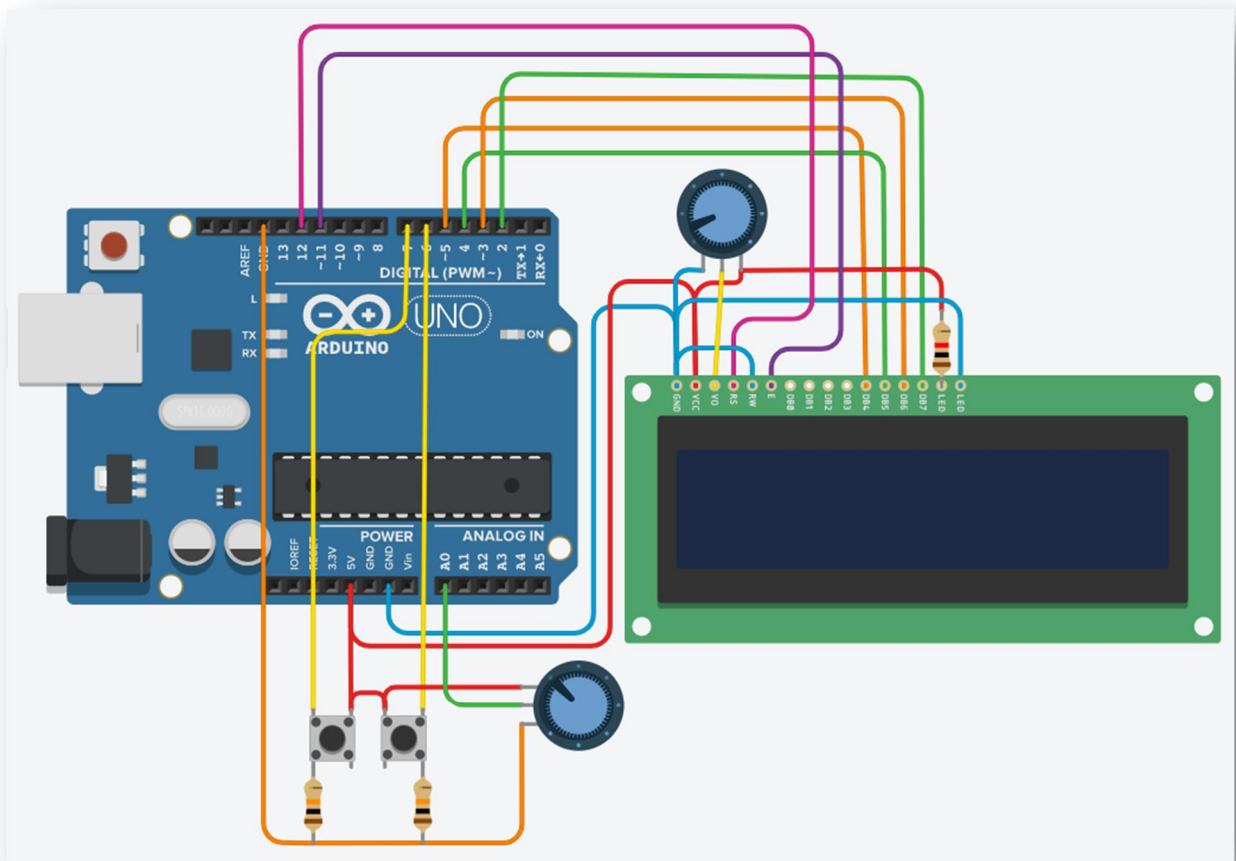
if (i == NUMPIXELS){
  i = 0;
  setColor();
}
}
void setColor(){
  redColor = random(0, 255);
  greenColor = random(0,255);
  blueColor = random(0, 255);
}

```

10. Прокоментуйте запропонований скетч і на його основі напишіть власний, змінивши його таким чином, щоб світлодіоди засвічувалися по черзі і по одному, кожне коло різним кольором.

11. Зберіть наступну схему:

В схемі використовуються такі деталі: плата Arduino, резистор 10 Ком – 2 шт., резистор 1 Ком – 1 шт., змінний резистор 1 Ком - 1 шт., змінний резистор 250 Ком (*підключений до піна A0*) – 1 шт.



12. Напишіть скетч, який виводить на LCD дисплей ваше ім'я у першому рядку, а прізвище у другому (для написання тексту використовуйте латинські літери).

13. Напишіть скетч, що виводить у першому рядку LCD дисплея, значення считане з вналогового піна A0, яке залежить від положення ручки змінного резистора.

14. Напишіть скетч, що виводить у другому рядку LCD дисплея, число яке змінюється при натисканні кнопок. При натисканні лівої кнопки число зменшується на 1, а при натисканні правої – збільшується на 1.

15. Використовуючи онлайн-сервісі **Wokwi** (<https://wokwi.com/>), підключіть LCD дисплей по шині I2C Напишіть скетч, який виводить на LCD дисплей ваше ім'я у першому рядку, а прізвище у другому (для написання тексту використовуйте латинські літери). .

ПИТАННЯ ДЛЯ ПІДГОТОВКИ.

1. Для чого використовують бібліотеку **LiquidCrystal.h**?
2. Як підключити LCD дисплей до плати Arduino?
3. Яка команда виводить інформацію на LCD дисплей?
4. З яких частин складається скетч?

Лабораторна робота №10 (Підключення до плати Arduino датчиків різного призначення)

Тема: Підключення до плати Arduino датчиків різного призначення.

Мета: Навчитися використовувати аналогові датчики у власних розробках на платформі Arduino.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

Датчики використовують в найрізноманітніших схемах. Жодна автоматизація не обходиться без них. Датчики є очима, вухами і іншими органами чуття мікроконтролера або іншого автоматизованого пристрою. Датчики можна класифікувати залежно від різних характеристик. Ось кілька основних критеріїв поділу:

За родом сигналу:

Аналогові датчики: видають аналоговий сигнал, який зазвичай представлений у вигляді змінної напруги чи струму, і залежить від вимірюваного параметра. Наприклад, термістори для вимірювання температури або аналогові фоторезистори для вимірювання освітленості.

Цифрові датчики: Видають цифровий сигнал, який може бути оброблений мікроконтролером чи іншими цифровими пристроями. Вони мають вбудований аналого-цифровий перетворювач (АЦП). Приклади цифрових датчиків включають датчики тиску з цифровим виводом або датчики руху.

За призначенням:

Датчики температури: Вимірюють температуру об'єкта або середовища. Приклади включають термістори, термопари, інфрачервоні датчики.

Датчики тиску: Використовуються для вимірювання атмосферного тиску, тиску рідини чи газу. Наприклад, барометри, п'єзорезистивні датчики.

Датчики вологості: Вимірюють вологість повітря або інших середовищ. Датчики вологості можуть бути резистивними або оптичними.

Датчики кислотності: Використовуються для вимірювання рівня кислотності (рН) рідини.

Датчики освітленості: Вимірюють рівень світла у навколишньому середовищі. Фоторезистори та фотодіоди є типовими прикладами.

Датчики рівня води: Використовуються для вимірювання рівня рідини або інших матеріалів. Ультразвукові датчики, поплавкові вимірювачі, тощо.

Датчики вібрації: Вимірюють вібрації або коливання. П'єзоелектричні датчики та акселерометри є прикладами датчиків вібрації.

Інші спеціалізовані компоненти: Це можуть бути датчики газу, диму, руху, магнітні датчики, датчики звуку та багато інших видів датчиків, які призначені для вимірювання конкретних параметрів або подій.

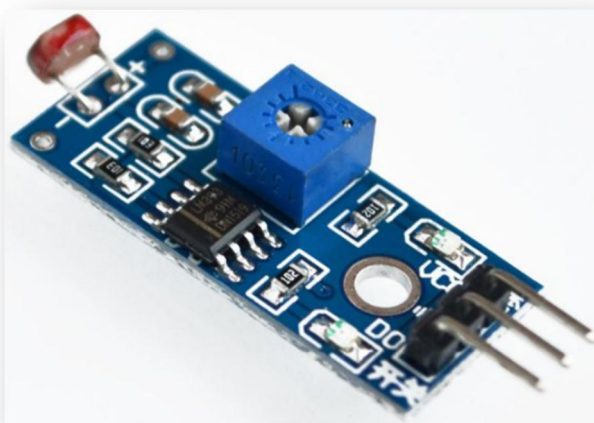
Класифікація датчиків може також ґрунтуватися на багатьох інших параметрах, таких як засіб вимірювання, принцип дії, точність тощо.

Датчик світла або освітленості

Найпростішим способом визначити освітленість чого-небудь – використати фоторезистор, фотодіод або фототранзистор.



Можна підключити до Ардуіно один з перерахованих варіантів або купити спеціальну плату - датчик освітленості.



У чому переваги готового рішення? По-перше, для визначення змін освітленості одного фотоелемента недостатньо, потрібен ще й звичайний або підстроювальний резистор, можливо наявність компаратора для ступеневого спрацьовування типу "так/ні".

По-друге, друкована плата, виготовлена в заводських умовах буде надійніше ніж навісний монтаж або макетна плата.

Так на платі розміщений підстроювальний резистор для регулювання моменту спрацьовування компаратора, який може видавати цифровий сигнал.

Приклади використання: датчик освітленості для фотореле, сигналізація (в парі з випромінювачем), лічильник об'єктів які перетинають світловий промінь і т.д.

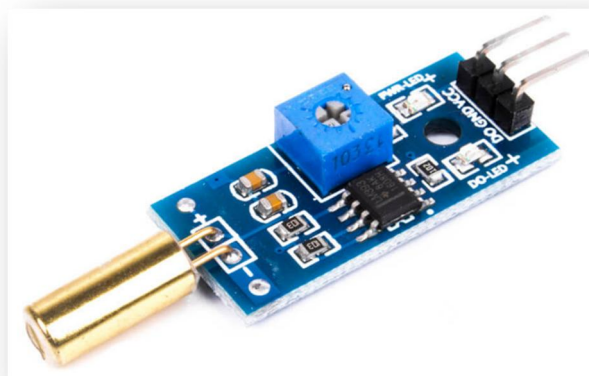
Точних значень отримати складно, так як для отримання точних і коректних значень освітленості знадобиться повноцінний люксметр. Фоторезистори більше підходять для визначення абстрактних величин типу «темно або світло».

Крім такої плати можна зустріти модуль GY-302. Це датчик освітленості на базі інтегральної мікросхеми BH-1750. Його особливістю є те, що це цифровий модуль, у нього розрядність в 16 біт, з'єднується з мікроконтролерами по шині I²C. 16 біт дозволяють вимірювати освітленість від 1 до 65356 Люкс (Лк).



Датчик нахилу і вібрації SW-520D

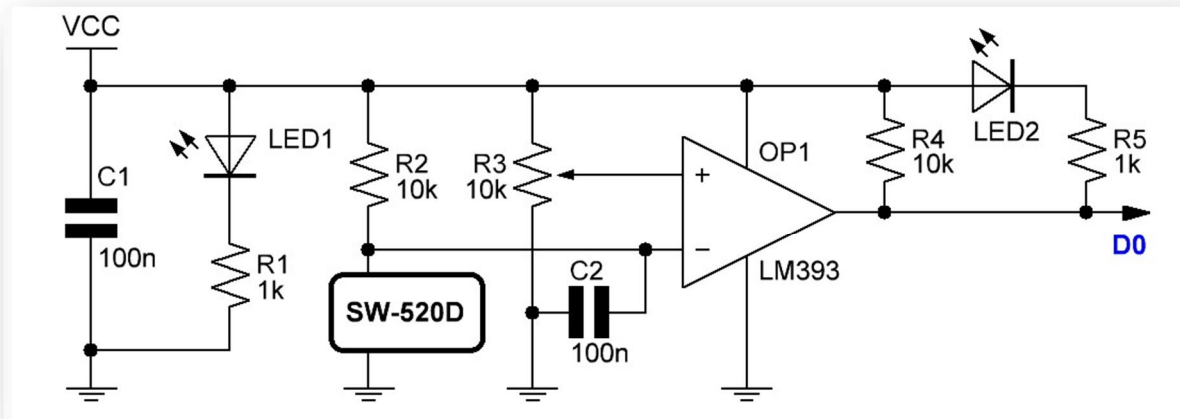
Модуль, який дозволяє фіксувати вібрацію і зміну положення свого чутливого елемента в просторі. Виробники випускають даний девайс під назвою SW-520D. Його зовнішній вигляд показаний на малюнку:



Дивлячись на малюнок, можна побачити датчик SW-520D, розпаяний на платі з електронної обв'язкою, основні елементи якої розглянуті нижче. Підлаштування резистор, включений в парі з операційним підсилювачем LM393, дозволяє регулювати чутливість датчика до механічних впливів. Індикатор живлення у вигляді SMD-світлодіоди загоряється при подачі на модуль 5В. Індикатор спрацювання має аналогічне виконання і допомагає візуально відстежити момент спрацювання датчика.

Модуль має три виводи для підключення, два з яких призначені для подачі живлення, а третій є сигнальним.

Принципова схема модуля показана на малюнку:

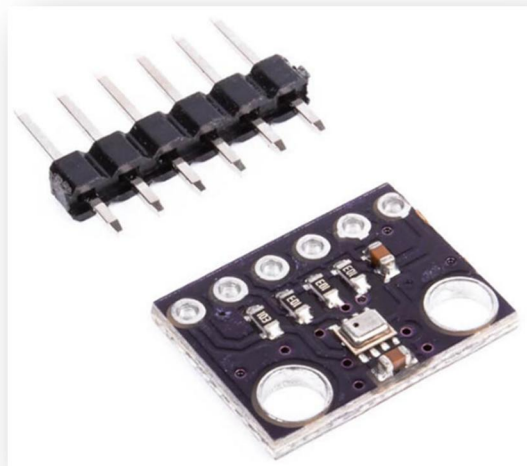


Конструктивно датчик SW-520D (циліндр жовтого кольору) являє собою порожнистий циліндр в якому вільно переміщуються дві металеві кульки, замикаючи і розмикаючи чутливі електроди.

Модуль атмосферного тиску BMP280

Модуль BMP280 є високоточним цифровим вимірювачем атмосферного тиску, який використовує мікросхемі BMP280 від фірми Bosch. Кожен такий датчик після виготовлення проходить індивідуальне калібрування на заводі.

Основні характеристики модуля BMP280, які зробили його популярним серед розробників проектів на Arduino, включають малі розміри, низьке енергоспоживання та високу точність вимірювань. Ця модифікація є технологічно більш розвиненою версією свого попередника, BMP180.



Модуль BMP280 надає користувачеві можливість вибору між двома інтерфейсами обміну даними: SPI і I2C. Також він пропонує три режими роботи:

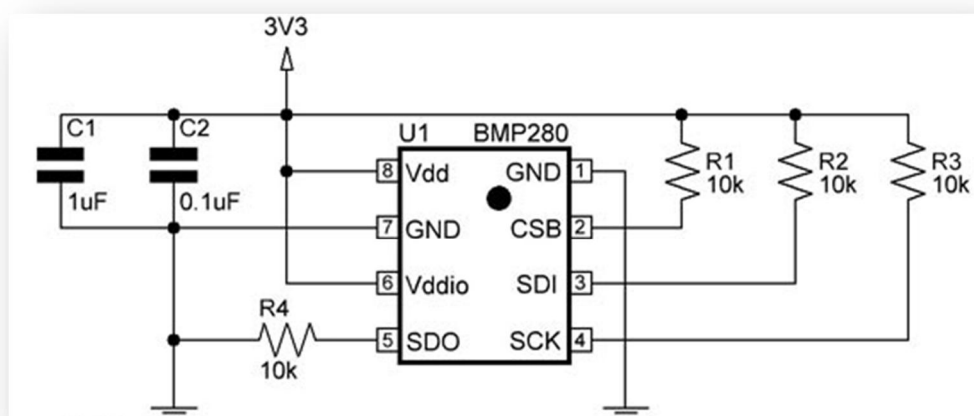
NORMAL: У цьому режимі модуль прокидається періодично, виконує вимірювання і переходить в режим сну. Користувач може налаштувати частоту вимірів, і дані зчитуються за запитом.

SLEEP: Це найекономічніший режим з найнижчим споживанням електроенергії, коли модуль перебуває в сплячому стані.

FORCED: Цей режим дозволяє моделі активувати за допомогою зовнішнього керуючого сигналу. Після вимірювань модуль автоматично повертається в режим зниженого споживання енергії.

Модуль BMP280 використовується для вимірювання атмосферного тиску, температури і може бути корисним в різних проектах, які потребують моніторингу кліматичних параметрів чи визначення висоти над рівнем моря.

Електрична схема модуля показана на малюнку:



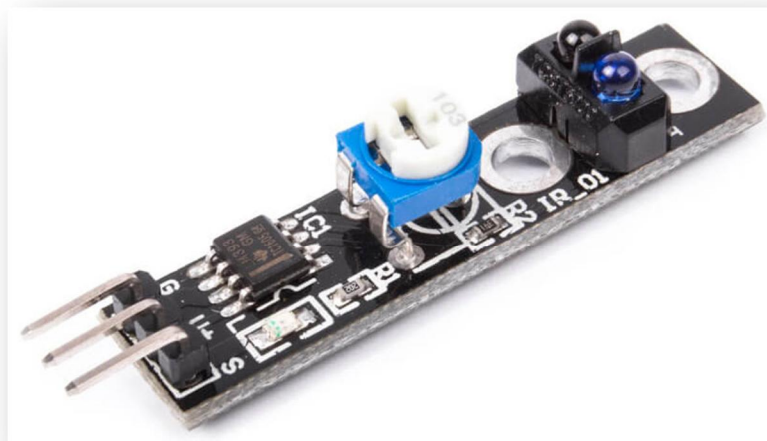
Датчик лінії KY-033

Датчик лінії – це пристрій, який використовується для виявлення білих або чорних ліній на певній поверхні. Ці пристрої, як правило, використовують оптичні методи для визначення наявності або відсутності ліній. Це дозволяє їм виконувати функції в різних застосуваннях, таких як навігація роботів, автономні вакуумні прибирачі, лінійні конвеєри, іграшки та багато інших.

Датчики ліній можуть визначати, наприклад, чи є чорні лінії на білому фоні або навпаки. Вони генерують сигнал або вивід, коли лінія знаходиться в їх зоні спостереження. Це дозволяє роботам або іншим пристроям слідувати за лінією або уникати її, в залежності від конкретної задачі.

Оптичні датчики ліній використовують світлодіоди та фотодетектори для створення та вимірювання світлових властивостей ліній і можуть бути корисними компонентами для багатьох автоматизованих систем і роботів.

Основним елементом датчика KY-033 є оптопара TCRT5000, що складається з інфрачервоного світлодіода і фототранзистора. Згідно з документацією, інфрачервоний діод випромінює світло з довжиною хвилі 950nm, що дозволяє достовірно визначати перешкоди на відстані від 1мм до 25мм. Зовнішній вигляд датчика KY-033 показаний на малюнку:



Також до складу модуля входять компаратор, який налаштовується змінним резистором і контрольний світлодіод. Змінним резистором вибирається поріг спрацьовування датчика на різний відтінок чорного кольору. Факт спрацьовування супроводжується загорянням контрольного світлодіода і подачею логічного нуля на сигнальний вивід датчика. Слід зазначити, що для зручності монтажу, по обидва боки оптичного елемента розташовані два отвори. Це дозволяє більш точно позиціонувати датчик або групу датчиків на необхідній відстані від перешкоди. Для виключення взаємного впливу світлодіода на фототранзистор їх конструктивно розділили невеликою перегородкою.

Що стосується принципу роботи, то він дуже простий. При подачі живлення на модуль, інфрачервоний світлодіод починає випромінювати світло, який відбиваючись від білої поверхні потрапляє на фототранзистор. У такому режимі на виводу **OUT (SIGNAL)** буде встановлена логічна одиниця. Як тільки в зону видимості датчика потрапляє чорний об'єкт, світловий потік, поглинаючись цим самим об'єктом, перестає доходити до фототранзистора і компаратор перемикає вивід **OUT (SIGNAL)** в логічний нуль.

Датчик кольору TCS230.

Модуль датчика TCS230 призначений для визначення кольору предмета. Мікросхема TCS230 перетворює інтенсивність колірної спектра в сигнал різної частоти. Інтенсивність колірної спектра обернено пропорційна частоті вихідного сигналу.

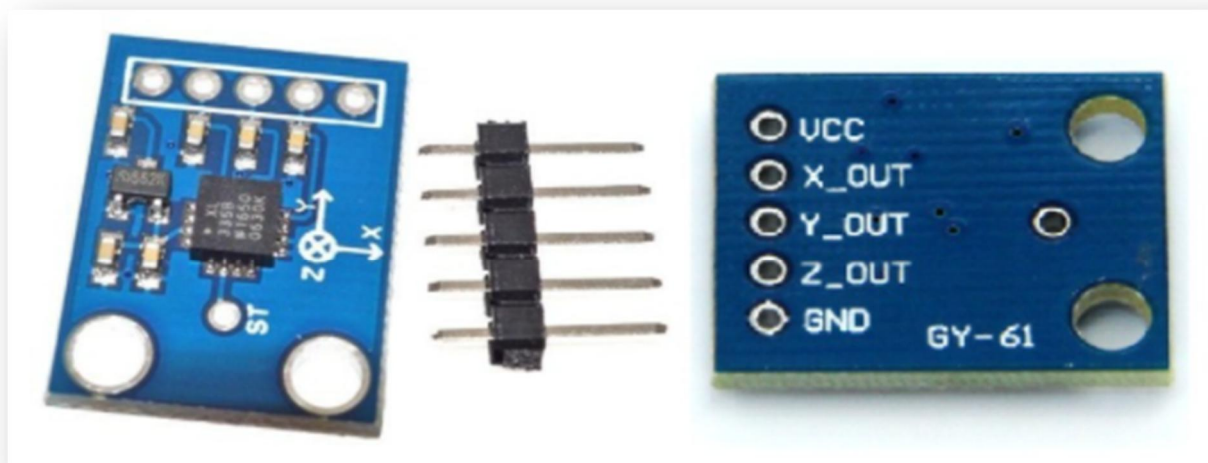
Мікросхема TCS230 для визначення кольору вимірює три спектра: червоний, синій, зелений. Мікросхема складається з 64 (4 * 16) фотодіодів, що визначають вибір фільтра: синій, зелений, червоний або без фільтра.



Вимірюваний об'єкт необхідно встановлювати паралельно до об'єкта вимірювання на відстані не більше 10 мм. На корпусі датчика розташовано чотири світлодіода, які використовуються для підсвічування місця вимірювання.

Датчик простору GY-61 ADXL335

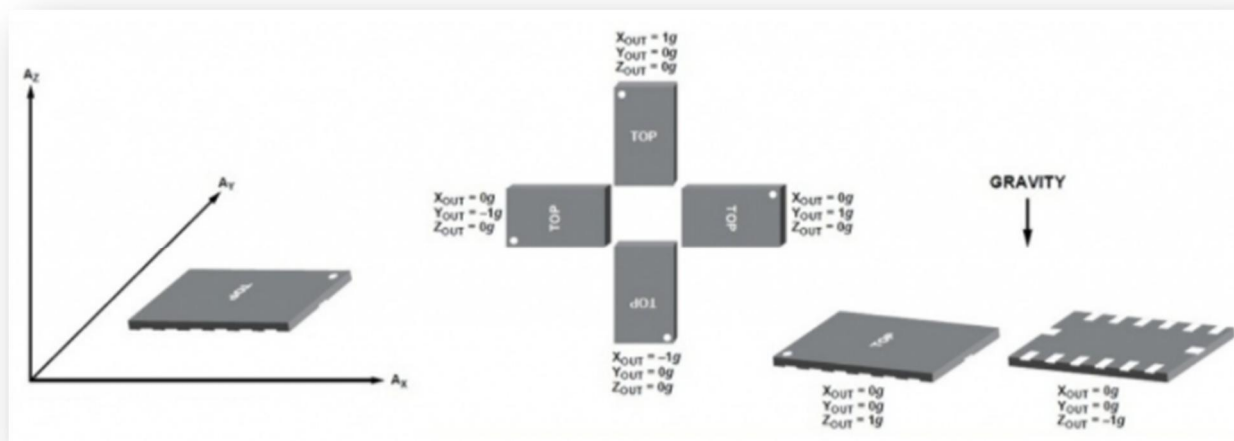
Акселерометр модуль GY-61 використовує мікросхему ADXL335 для вимірювання прискорення. Ця мікросхема має три осі, що дозволяє визначати вектор прискорення в тривимірному просторі. Прискорення, яке вимірюється цим акселерометром, включає силу земного тяжіння, яка діє вниз відносно центру Землі.



Завдяки можливості вимірювання вектора прискорення по трьох взаємно перпендикулярних осях, акселерометр дозволяє визначити орієнтацію пристрою в просторі. Це може бути корисним, наприклад, для визначення нахилу або повороту пристрою.

Модуль GY61 на базі ADXL335 може бути використаний в різних додатках, включаючи навігаційні системи, робототехніку, іграшки, прилади з контролем за рухом та інші сфери, де важливо вимірювати прискорення та визначати орієнтацію.

Координатні осі чутливості акселерометра по відношенню до геометричного розміщення корпусу пристрою в просторі показано на зображенні:



Якщо датчик сильно труснути уздовж вертикальної осі Z, то значення Z_{out} буде більше, ніж $1g$. Максимальна вимірювана прискорення - $3g$ по кожній з осей в будь-якому з напрямів.

Якщо сильно трусити датчик уздовж вертикальної осі Z, то значення Z_{out} може перевищити $1g$ у величині, що вказує на збільшене прискорення в цьому напрямку. При цьому, якщо датчик встановлений вертикально і відповідає гравітаційному полю, то значення Z_{out} буде близьким до $1g$, де g - середнє значення прискорення вільного падіння (приблизно 9.81 м/с^2 на поверхні Землі).

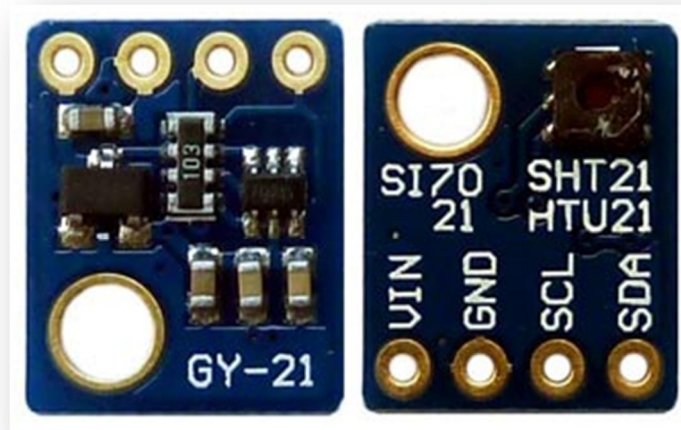
Датчик температури і вологості GY-21.

Модуль датчика температури і вологості GY21, який базується на датчику SHT21, є високоточним і надійним засобом для вимірювання температури та вологості в промислових умовах. Він володіє дуже низькою похибкою для свого класу, що робить його корисним для точних вимірів у різних застосуваннях. Дана похибка становить 0.4% для вимірювання температури і 2% для вимірювання вологості, що гарантує високу точність вимірів.

Модуль також підтримує інтерфейс I2C, що робить його легким у використанні з мікроконтролерами і одноплатними комп'ютерами, такими як Arduino або Raspberry Pi. Завдяки цим характеристикам, він ідеально підходить для вимірювання температури та вологості в промислових приміщеннях, де важлива точність та надійність даних.

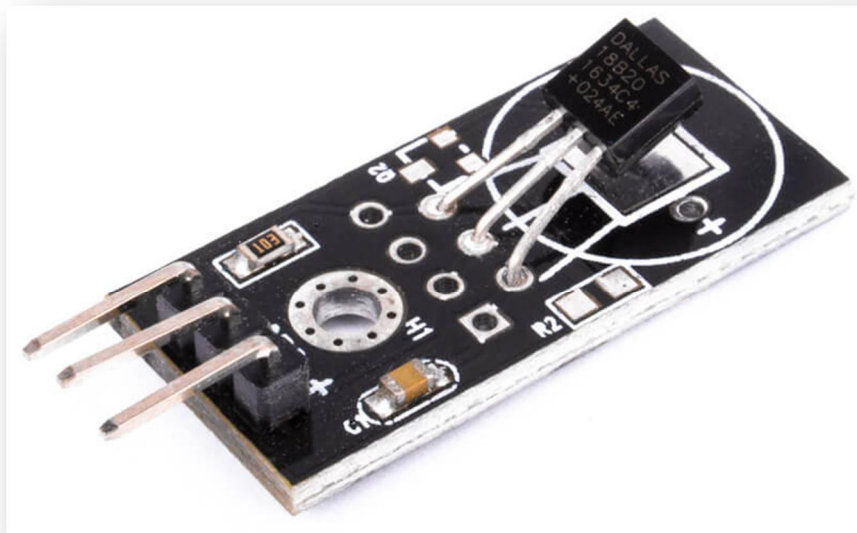
Цей модуль може бути корисним у багатьох застосуваннях, включаючи системи контролю клімату, моніторингу умов зберігання, а також в

дослідницьких та інженерних проектах, де необхідно виміряти температуру та вологість з високою точністю.



Датчика температури DS18B20.

Цифровий датчик температури повітря на мікросхемі DS18D20, яка перетворює температуру корпусу сенсора в інформацію передану по послідовній двохпровідній шині даних 1-Wire. Даний модуль призначений для роботи спільно з мікроконтролерами, зокрема на базі Arduino. Модуль датчика температури DS18B20 застосовується для вимірювання температури повітря як в приміщенні, так і на відкритому повітрі.



DS18B20 – це цифровий термометр, який здатний точно вимірювати температуру в діапазоні від -55°C до $+125^{\circ}\text{C}$ з програмованою точністю від 9 до 12 біт. Кожному датчику DS18B20 присвоюється унікальна 64-бітова адреса під час виробництва. Для обміну інформацією з пристроєм, таким як мікроконтролер або плата Arduino, використовується спеціальна шина 1-wire.

Однією з основних переваг DS18B20 є можливість підключення до однієї лінії багатьох таких датчиків, навіть до 264 одиниць. Це робить їх ідеальними

для вимірювання температури в різних точках або різних приміщеннях, в яких потрібно знати температурні параметри.

Ці датчики знайшли широке застосування в багатьох проектах, включаючи системи контролю клімату, метеостанції, промислові системи моніторингу і багато інших додатків, де дуже важливе точне вимірювання температури.

Датчик руху HC-SR501

Модуль датчика руху HC-SR501 використовує піроелектричний ефект і складається з наступних компонентів:

- PIR-датчик 500BP: це піроелектричний датчик, який здатний реагувати на зміни теплового випромінювання від рухомих об'єктів. Він є основним сенсором для виявлення руху і присутності.
- Мікросхема BISS0001: ця мікросхема відповідає за обробку сигналу від PIR-датчика і керування виходами модуля. Вона дозволяє налаштовувати параметри датчика, такі як чутливість і тривалість вимірювань.
- Лінза Френеля: лінза Френеля встановлена перед PIR-датчиком і призначена для збільшення радіусу огляду датчика та підвищення чутливості до інфрачервоного сигналу. Вона допомагає збільшити точність виявлення руху.

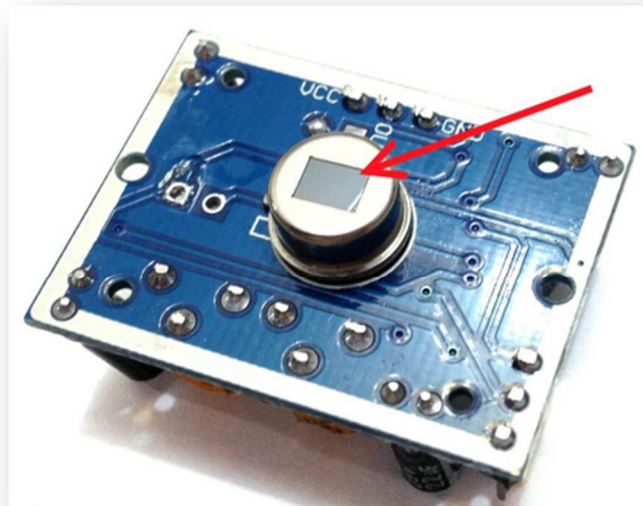
Цей модуль датчика руху використовується в різних додатках, таких як системи безпеки, автоматичні системи освітлення, системи енергозбереження та інші, де важливо виявлення руху або присутності особи.



Лінзи Френеля



Модуль використовується для виявлення руху об'єктів, випромінюючих інфрачервоне випромінювання. Чутливий елемент модуля – PIR-датчик 500BP.



Модуль датчика руху HCSR501 використовує принцип роботи на основі піроелектрика та керується мікросхемою BISS0001. Ось основні особливості його роботи:

- Принцип роботи: Модуль виявляє рух і присутність об'єктів завдяки змінам теплового випромінювання, які виникають при русі об'єктів в полі зору датчика. Піроелектричний ефект допомагає перетворювати зміни температури на електричний сигнал.
- Налаштування дистанції і затримки: Модуль має два потенціометри, які дозволяють налаштувати дистанцію виявлення об'єктів (від 3 до 7 метрів) та затримку після спрацювання датчика (від 5 до 300 секунд).
- Режими роботи: Модуль має два режими роботи: режим **L** і режим **H**. У режимі **L (Low)** датчик встановлюється на одноразовий режим спрацювання. При виявленні руху виходить сигнал **HIGH** на визначений час затримки. Це може бути використано, наприклад, в системах охорони для активації сирени після виявлення руху.

У режимі **H (High)**, датчик спрацьовує при кожному виявленні руху і видає відповідний сигнал. Цей режим може бути використаний для управління освітленням.

- Калібрування: Після включення модуль проводить калібрування, яке триває приблизно одну хвилину. Після калібрування модуль готовий до роботи.
- Розміщення: Рекомендується розміщувати датчик далеко від джерел світла, оскільки світло може впливати на його роботу.

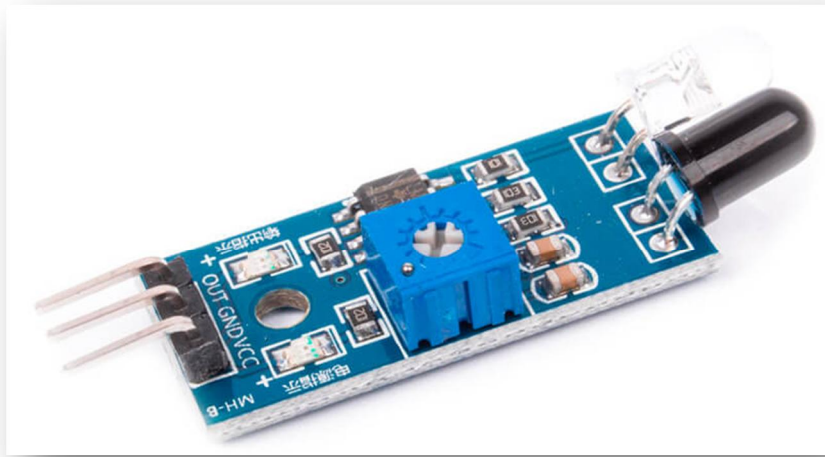
Цей модуль датчика руху може бути використаний для автоматизації різних систем і додатків, де важливо виявлення руху та об'єктів.

Інфрачервоний датчик перешкод YL-63

Датчик обходу перешкод YL-63 є цифровим інфрачервоним пристроєм, призначеним для виявлення наявності об'єкта перед ним. Ось деякі основні особливості цього датчика:

- Складові: Датчик включає в себе інфрачервоний випромінювач і фотоприймач. Інфрачервоний випромінювач генерує інфрачервоний сигнал, який відбивається від об'єкта і приймається фотоприймачем.
- Робочий принцип: Датчик вимірює кількість інфрачервоного світла, яке повертається від об'єкта. Якщо об'єкт перешкоджає шляху інфрачервоного променя між випромінювачем і фотоприймачем, то виходить інший сигнал, що свідчить про наявність об'єкта.
- Дистанція і точність: Цей датчик призначений переважно для виявлення наявності об'єкта, а не для точного вимірювання відстані до нього. Точність визначення відстані може бути обмеженою.
- Застосування: Цей тип датчика часто використовується в системах автоматизації для виявлення руху або наявності об'єктів, таких як пасажирів, які підходять до автоматичних дверей, об'єкти на конвеєрах та інші подібні застосування.
- Режим роботи: Датчик може працювати в режимі перешкоди (якщо об'єкт виявлений) або в режимі відсутності перешкоди (якщо об'єкт відсутній).
- Підключення: Зазвичай цей датчик має цифровий вихід і може бути легко інтегрований в мікроконтролерні системи або інші пристрої автоматизації.

Датчики обходу перешкоди, такі як YL-63, грають важливу роль в різних автоматизованих системах, сприяючи виявленню об'єктів та управлінню подіями на основі їх наявності.



Датчик перешкоди, побудований на основі компаратора LM393, призначений для виявлення перешкод в діапазоні відстаней від нуля до певної граничної відстані. Ось декілька ключових характеристик цього датчика:

- Оптичний принцип: Датчик використовує оптичний метод для виявлення перешкод. Він випромінює інфрачервоне світло і вимірює відбитий сигнал. Якщо об'єкт перешкоджає шлях цього світла, то компаратор LM393 переводить вихідний сигнал в логічний "HIGH" (1), що показує наявність перешкоди.
- Регульований діапазон: Встановлений потенціометр дозволяє регулювати граничну відстань, на яку реагує датчик. Це дозволяє користувачам налаштувати датчик під свої потреби і умови.
- Індикація стану: Датчик має світлодіодну індикацію. Червоний світлодіод вказує на наявність перешкоди, і зелений світлодіод індикатора показує статус живлення.
- Застосування: Такі датчики перешкоди застосовуються в робототехніці для виявлення наявності перешкод перед колісними або гусеничними роботами. Це допомагає роботам уникати зіткнень з перешкодами та забезпечує безпеку їхньої рухомості.
- Підключення: Датчик може бути підключений до мікроконтролерних систем або інших пристроїв для обробки сигналу і прийняття рішень на основі виявлення перешкод.

Цей тип датчиків допомагає автоматизованим системам рухомості реагувати на найближчі перешкоди та приймати необхідні дії для уникнення зіткнень.

Датчик рівня води

Датчики води використовуються для визначення рівня рідини в ємностях, де немає можливості проводити візуальний контроль. Головна функція цих датчиків полягає в тому, щоб вчасно виявити надмірний рівень рідини, що дозволяє уникнути переповнення ємності.

Декілька ключових особливостей та принципів роботи датчиків води:

- Контактний та безконтактний спосіб вимірювання: Датчики води можуть бути контактними, коли їхні сенсори фізично контактують з рідиною, або безконтактними, коли вони використовують інші технології, наприклад ультразвукову.
- Спостереження за рівнем: Датчики води відстежують рівень рідини і зазвичай активують сигнал або сповіщення, коли рівень досягає певної критичної позначки.
- Застосування: Вони використовуються в багатьох областях, включаючи системи водопостачання та каналізації, басейни, акваріуми, машини для кави, пральні машини, бойлери, ванни, водні баки для тварин, системи автоматичного поливу і багато інших сфер.
- Керування пристроями: Датчики води можуть бути інтегровані в системи керування, щоб автоматично вимикати або включати пристрої, коли рівень води досягає визначеної точки. Наприклад, датчик води в пральній машині може припинити процес прання, коли бак заповнюється водою.
- Різноманітні технології: Існує багато різних технологій для вимірювання рівня води, включаючи поплавкові датчики, оптичні сенсори, ультразвукові датчики та інші. Вибір конкретного датчика залежить від конкретних потреб та умов використання.

Датчики води грають важливу роль в забезпеченні безпеки та надійності систем, де важливо вимірювати та контролювати рівень рідини.



Датчик вологості ґрунту.

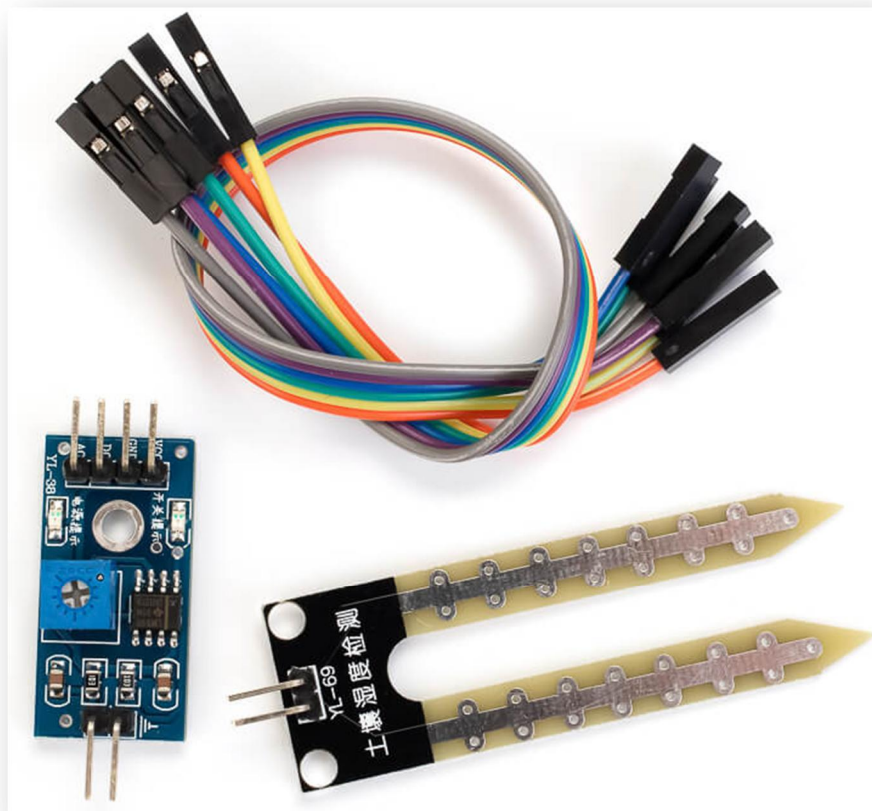
Датчик вологості ґрунту є корисним пристроєм для садівників та аграріїв, які бажають відстежувати та контролювати вологість ґрунту для оптимального зростання рослин. До ключових характеристик та застосувань цих датчиків можна віднести:

- Визначення вологості ґрунту: Датчики вологості ґрунту можуть точно вимірювати рівень вологості в ґрунті. Вони використовуються для

визначення, наскільки вологим є ґрунт, що вказує на потребу в поливі рослин або на недоліки водопостачання.

- Автоматизація поливу: Підключення датчиків вологості ґрунту до систем автоматичного поливу дозволяє регулювати полив рослин в залежності від рівня вологості. Це допомагає зекономити воду і забезпечити оптимальні умови для росту рослин.
- Захист від переповнення або пересушування: Вологості ґрунту також використовуються для запобігання переповненню рослин і надмірному поливу, що може бути шкідливим для кореневої системи.
- Моніторинг ґрунтового середовища: Ці датчики використовуються також в наукових дослідженнях для моніторингу вологості ґрунту в природних екосистемах, сільському господарстві та інших галузях.
- Підтримка садівництва та аграрної справи: Датчики вологості ґрунту допомагають садівникам та фермерам підтримувати відповідні умови для росту рослин, забезпечуючи їх оптимальними ресурсами.
- Моніторинг росту рослин: Ці датчики також можуть бути використані для вивчення взаємозв'язку між вологості ґрунту і зростанням рослин, що може бути корисним для наукових досліджень та агрономічних досліджень.

Датчики вологості ґрунту дозволяють забезпечити кращий догляд і управління рослинами, а також сприяють збереженню водних ресурсів та екологічно відповідному сільському господарству.

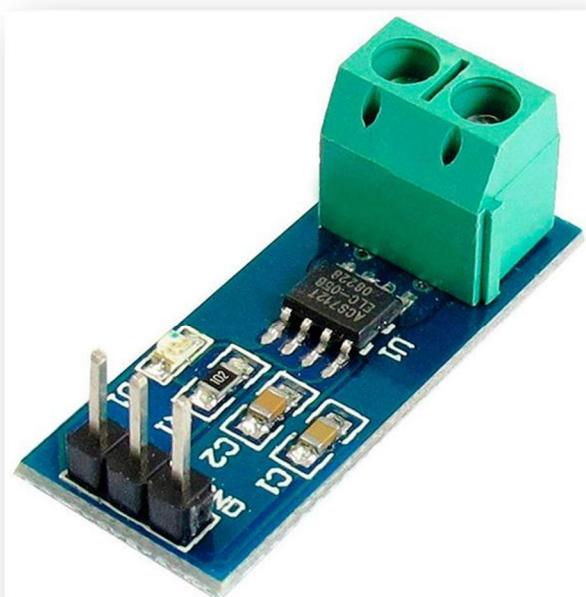


Датчик вологості ґрунту складається з контактної щупи YL-69 і датчика YL-38, призначений для вимірювання рівня вологості в ґрунті. Контактний щуп YL-69 має два електроди, які стикаються з ґрунтом. Він створює невелику напругу між цими електродами. Датчик YL-38 підключений до щупа YL-69 за допомогою двох проводів і призначений для зчитування ступеня вологості в ґрунті. Рівень вологості в ґрунті впливає на опір між електродами контактної щупи YL-69. Якщо ґрунт сухий, то опір великий, і струм, що проходить через нього, менший. У вологому ґрунті опір менший, і струм більший. Датчик YL-38 зчитує аналоговий сигнал, який відображає рівень вологості в ґрунті. Цей сигнал може бути переданий контролеру для подальшого аналізу.

Отже, цей датчик допомагає визначити, наскільки вологим є ґрунт і дозволяє вчасно вживати заходи для забезпечення потрібного рівня вологості для рослин.

Датчик струму ACS712

Датчик струму ACS712 - це електронний пристрій, призначений для вимірювання електричного струму, який протікає через нього. Датчик ACS712 містить елемент Холла, який реагує на зміни магнітного поля, що виникають через проходження струму. Цей елемент генерує аналоговий сигнал, який пропорційний інтенсивності струму.



Датчики ACS712 доступні в різних варіантах з різними вимірювальними діапазонами, наприклад, 5 А, 20 А, тощо. Вони призначені для вимірювання струму у відповідному діапазоні.

Датчик генерує аналоговий сигнал, який можна зчитати за допомогою мікроконтролера або іншого електронного пристрою. Цей сигнал вказує на поточний струм, який проходить через датчик. Деякі моделі цих датчиків мають вбудований захист від перевантаження, що робить їх надійними для застосувань в системах безпеки і керування струмом.

Датчик струму ACS712 широко використовується в різних галузях, включаючи вимірювання енергоспоживання, контроль струму в електричних схемах, а також для вимірювання струму в інших електронних пристроях.

ACS712 забезпечує вимірювання АС і DC струму в промислових, автомобільних, комерційних системах і системах зв'язку.

Датчик інтенсивності світла GY-302 (BH1750)

Датчик інтенсивності світла GY-302 або BH1750, є електронним пристроєм для вимірювання освітленості в навколишньому середовищі.



Датчик BH1750 має вбудований фотодіод, який реагує на світло від навколишнього середовища. Цей фотодіод здатний конвертувати світловий потік в електричний сигнал. Датчик має широкий діапазон вимірювань освітленості в люксах. Деякі моделі дозволяють налаштовувати точність вимірювань для різних застосувань. Датчик BH1750 підтримує різні інтерфейси для підключення до мікроконтролерів або інших електронних пристроїв, такі як I2C або аналоговий вихід. Результати вимірювань освітленості виводяться у люксах (lx) і можуть бути зчитані та використані для подальшого керування освітленням або іншими системами домашньої автоматики.

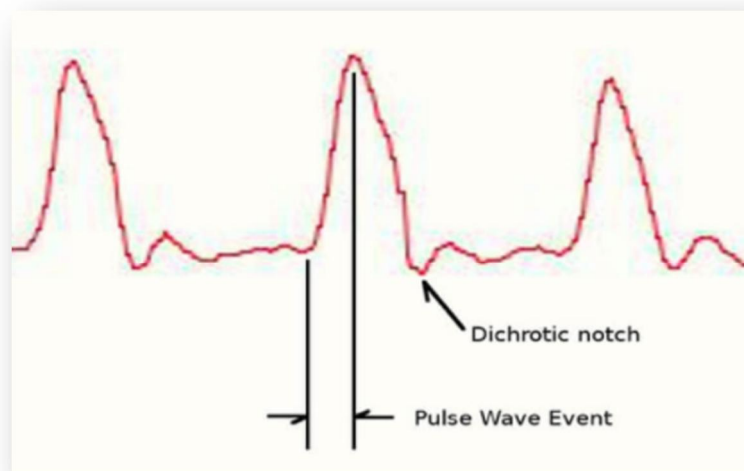
Датчик BH1750 широко використовується в домашніх автоматизованих системах для контролю освітлення, регулювання яскравості освітлення в залежності від умов освітленості, а також для ефективного використання електроенергії.

Датчик пульсу

Пульс – це ритмічні коливання стінок кровоносних судин, що відбуваються під час скорочень серця. Вимірювання пульсу дуже важливі для діагностики серцево-судинних захворювань. Важливо стежити за змінами серцевого ритму, щоб не допустити перевантаження організму, особливо під час занять спортом. Один з зрозумілих параметрів пульсу – частота пульсу. Вимірюється в кількості ударів за хвилину.

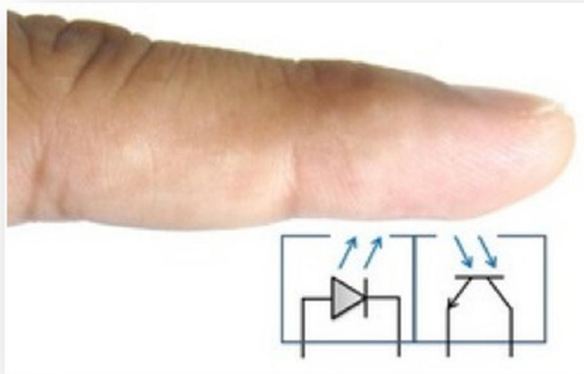


Датчик пульсу Pulse Sensor є доступним і простим у використанні засобом для вимірювання серцевого ритму. Цей аналоговий датчик використовує метод фотоплетизмографії для вимірювання пульса. Фотоплетизмограма – метод вимірювання пульсації крові за допомогою інфрачервоного світла та фотодетекторів. Графік фотоплетизмограми має такий вид:



Датчик Pulse Sensor використовує світлодіод і фотодетектор для вимірювання змін оптичної щільності обсягу крові в біологічній тканині, де проводиться вимір. Зміни цієї щільності відбуваються внаслідок кровотоку, який змінюється відповідно до фаз серцевого циклу. Датчик накладається на палець руки або іншу відповідну область, де є доступ до кровотоку. Світлодіод випромінює світло через шкіру, і фотодетектор вимірює інтенсивність світла, яке відбивається від крові. Датчик Pulse Sensor генерує аналоговий сигнал, який відображає зміни інтенсивності світла в часі. Цей сигнал може бути зчитаний за

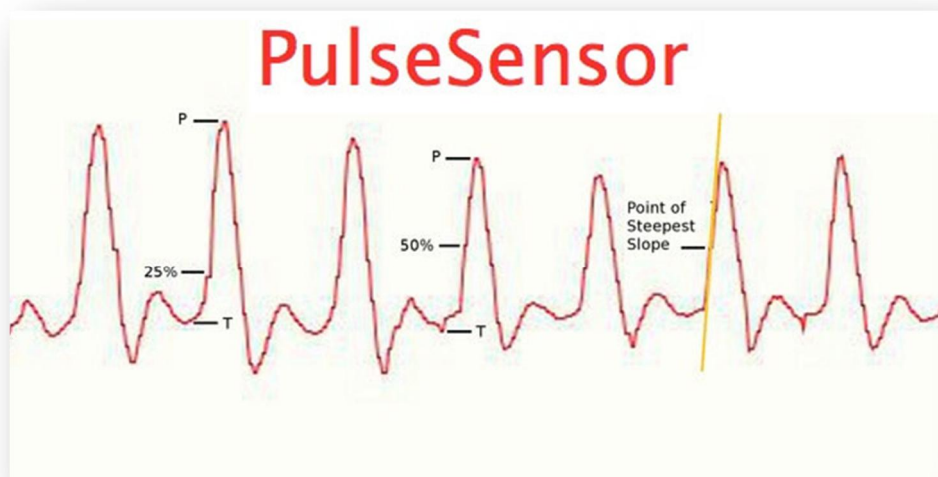
допомогою мікроконтролера або іншого аналогового пристрою для подальшого аналізу пульса.



Pulse Sensor використовується для вимірювання серцевого ритму, вивчення пульсу та відслідковування серцевого здоров'я. Його можна використовувати у різних проектах, таких як моніторинг пульсу під час фізичних вправ, біометричних систем або для реагування на пульсові зміни в реальному часі.

Датчик Pulse Sensor є популярним засобом для швидкого та недорогого вимірювання серцевого ритму та відстеження здоров'я. Датчик пульсу підсилює аналоговий сигнал і нормалізує щодо точки середнього значення напруги живлення датчика ($V/2$).

Датчик реагує на відносні зміни інтенсивності світла. Якщо кількість світла, що падає на датчик залишається постійним, величина сигналу буде залишатися на рівні середини діапазону АЦП. Якщо реєструється велика інтенсивність, то крива сигналу йде вгору, якщо інтенсивність менша, то крива йде вниз.



Датчик пульсу використовуємо для вимірювання частоти пульсу, фіксуючи проміжок між точками графіка, коли сигнал має значення 50% від амплітуди хвилі під час початку імпульсу.

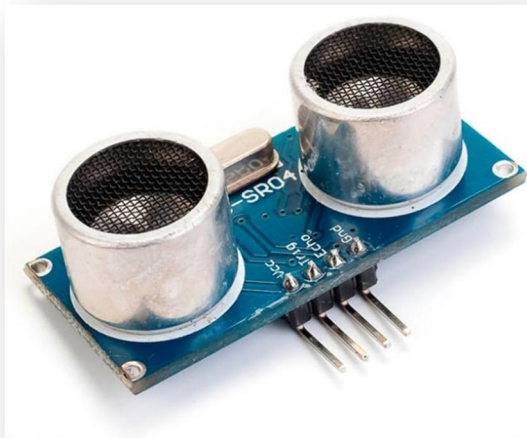
Ультразвуковий далекомір HC-SR04

Ультразвуковий далекомір HC-SR04 – це датчик, який використовує ультразвук для вимірювання відстані до об'єкта.

Датчик HC-SR04 – це пристрій для вимірювання відстані, який використовує ультразвукові хвилі. Датчик HC-SR04 генерує ультразвукові хвилі. Цей сигнал відправляється до об'єкта, до якого вимірюється відстань, потім приймає відбиті від об'єкта ультразвукові хвилі і визначає час який затребував сигнал на подолання відстані до об'єкта і повернення їх відбиття. Відстань визначається на основі вимірюваного часу і швидкості поширення звуку в повітрі.

Датчик включає в себе мікроконтролер, який контролює відправку та отримання ультразвукових сигналів. Результат вимірювання може бути зчитаний через цифровий інтерфейс, як, наприклад, відстань в сантиметрах або дюймах.

Датчик HC-SR04 досить популярний у робототехніці, автономних автомобілях та інших розробках, де необхідно точно виміряти відстань до об'єктів.



Послідовність дій для вимірювання відстані за допомогою ультразвукового датчика HC-SR04:

- Подаємо імпульс тривалістю 10 мікросекунд на вивід **Trig**.
- На платі модуля вхідний імпульс перетворюється на 8 імпульсів з частотою 40 кілогерц і відправляється через випромінювач.
- Послані імпульси відбиваються від об'єкта та приймаються приймачем, і, в результаті, ми отримуємо вихідний сигнал на виводі **Echo**.
- За допомогою контролера ми перетворюємо отриманий сигнал на відстань, використовуючи такі формули:

Ширина імпульсу (мікросекунди) / 58 = відстань (сантиметри)

Ширина імпульсу (мікросекунди) / 148 = відстань (дюйми)

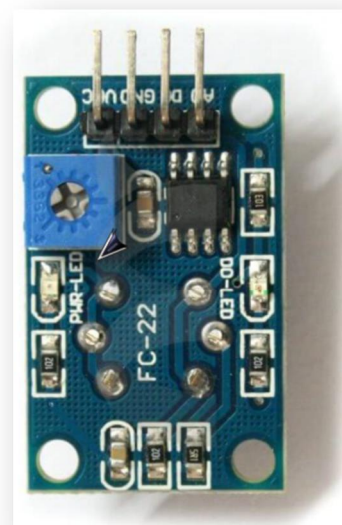
Цей процес дозволяє точно визначити відстань до об'єкта, використовуючи час, який ультразвуковому сигналу необхідно для подолання

цієї відстані і повернення назад. Цей метод широко використовується в проектах робототехніки, безпілотних літальних апаратах та інших сферах, де важливо знати точну відстань до об'єктів.

Датчик чадного газу MQ7

Датчик MQ7 – це прилад для виявлення чадного газу (CO), який виникає під час згоряння вуглецевого палива при недостатньому доступі кисню. Цей газ є надзвичайно отруйним, але він не має кольору або запаху, тому його важко виявити без спеціального обладнання.

Датчик MQ7 призначений для раннього виявлення наявності чадного газу у приміщенні, що допомагає запобігти отруєнню в разі небезпеки.



Даний датчик має широке застосування в системах автоматизації та безпеці, і його використання дозволяє вчасно реагувати на витіки чадного газу і приймати необхідні заходи для забезпечення безпеки в приміщенні.

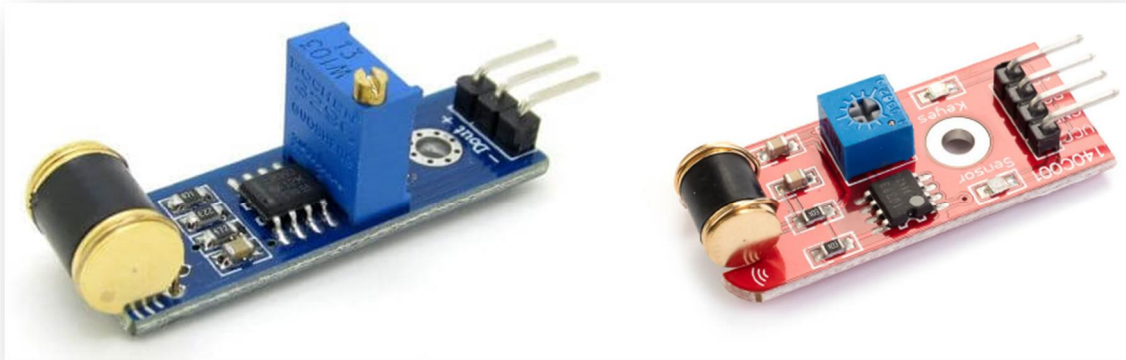
Датчик MQ7 використовує нагрівальний елемент для ініціювання хімічної реакції, яка дає інформацію про концентрацію чадного газу. Цей нагрівальний елемент гарячий під час роботи, і він вимагає попереднього прогріву (приблизно 48 годин) для отримання стабільних показань. Після прогріву датчик стабілізується за хвилину після включення.

Датчик генерує аналоговий сигнал, який пропорційний концентрації чадного газу. Важливо враховувати, що на покази датчика можуть впливати температура та вологість навколишнього середовища.

Датчик вібрації Arduino.

Датчик вібрації Arduino використовується для визначення зовнішніх вібраційних впливів. Вони можуть бути застосовані при створенні різних сигналізацій. Датчики вібрації, такі як Logo Sensors v1.5 і 140C001, базуються на гнучких металевих пружинках, розташованих всередині пластикової трубки. Ці пружинки коливаються у відповідь на будь-які впливи або вібрації, які діють на них. Сигнал, який генерується цими коливаннями, потім посилюється за

допомогою операційного підсилювача, такого як LM386. Нарешті, оброблений сигнал надсилається на аналоговий вихід, де його можна використовувати для вимірювання або контролю вібраційних подій.

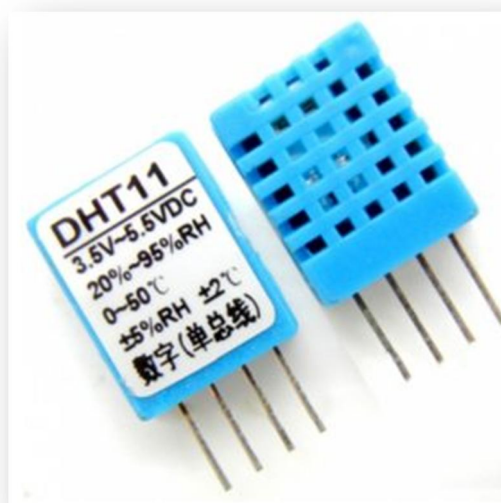


Обидва датчики вібрації (Logo Sensors v1.5 і 140C001) мають спільні виводи: GND для заземлення, Vcc для живлення та вивід A0 для аналогового сигналу. Вони також обладнані світлодіодами, які показують наявність живлення. Крім цього, датчик 140C001 має додатковий цифровий вивід D0, який видає логічний нуль, якщо величина вібрації перевищує налаштований поріг. Цей поріг можна регулювати за допомогою потенціометра.

Цифровий вивід D0 та світлодіод D0 на датчику 140C001 дозволяють використовувати його самостійно, без підключення до контролера. Датчики також мають монтажний отвір для кріплення до поверхні, що робить їх зручними для встановлення.

Датчик вологості і температури.

Датчик DHT11 є цифровим сенсором, призначеним для вимірювання вологості і температури. Він містить термістор для вимірювання температури і ємнісний датчик для вимірювання вологості. Додатково, внутрішній аналого-цифровий перетворювач (АЦП) використовується для перетворення аналогових вимірювань у цифрові значення для подальшого використання.



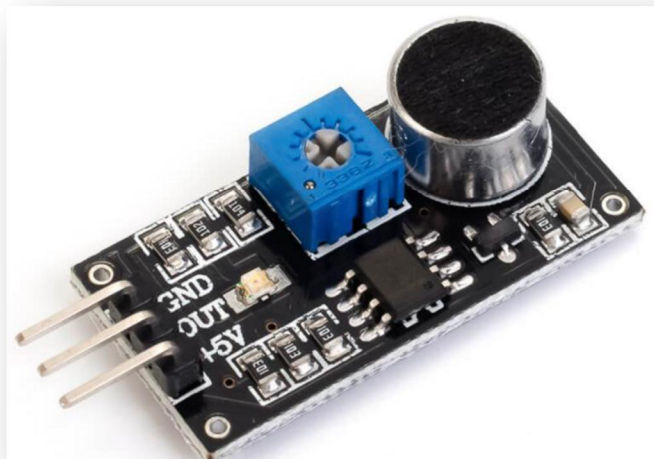
Незважаючи на те, що DHT11 має деякі обмеження у швидкодії та точності, він є простим у використанні, доступним та добре підходить для навчальних цілей та контролю рівня вологості в приміщенні.

Датчик рівня звуку

Даний датчик включає в себе мікрофон, мікрофонний підсилювач і змінний резистор для зміни чутливості. Мікрофон перетворює звукові коливання на електричний сигнал. Цей сигнал потім підсилюється за допомогою компаратора L293.

Датчик складається з невеликої плати, на якій розташовані наступні компоненти:

- Датчик має вбудований мікрофон, який використовується для перетворення звукових коливань в електричний сигнал.
- Мікрофонний підсилювач: Після перетворення звуку в електричний сигнал, сигнал підсилюється за допомогою мікрофонного підсилювача, що допомагає збільшити амплітуду сигналу.
- Регулятор чутливості: Для налаштування чутливості датчика встановлений змінний резистор, який дозволяє користувачу налаштовувати чутливість на свій розсуд.



Після підсилення сигналу він подається на вхід компаратора L293, який може порівнювати сигнал з встановленим порігом чутливості і видає вихідний сигнал в залежності від того, чи сигнал перевищує цей поріг. Регулятором чутливості можна вибирати, від якого звуку буде спрацьовувати датчик – від слабкого, гучного або дуже гучного звуку.

Цей датчик використовується для вимірювання звуку та відслідковування наявності звукових подій. Він може бути використаний в різних додатках, де необхідно реагувати на звукові сигнали або події.

Джойстик


Джойстик складається з двох потенціометрів з номіналом 10 кОм кожен, які визначають положення вісей X та Y. Кожен потенціометр має рухливий контакт, який обертається при нахилі ручки джойстика. Середні контакти цих потенціометрів підключені до роз'ємів VRX і VRY, тоді як крайні контакти підключені до живлення і землі. Джойстик може використовуватися як аналогове пристрій управління у вашому проекті.



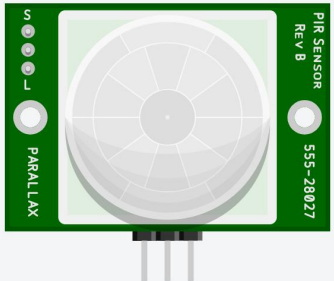


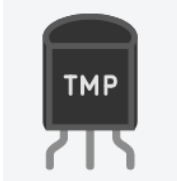
Додатково, на джойстику присутня тактова кнопка, яка може бути натиснута вертикально на ручку. Показання кнопки зчитуються з контакту SW. Після відпускання кнопки, джойстик автоматично повертається у центральне положення.

Джойстик використовується для керування різними пристроями, надає можливість керувати рухом в горизонтальному та вертикальному напрямках, а також виконувати дії за допомогою кнопки.

Беручи за увагу той факт, що в сервісі **tinkercad.com** на сьогодні у наявності є тільки такі елементи, які можна використовувати у якості датчиків:

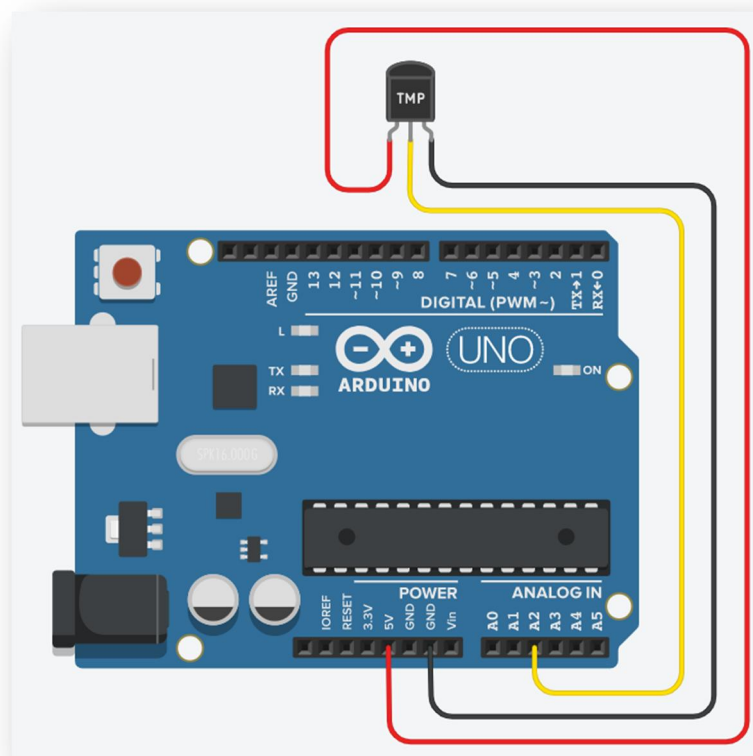
№	Зображення	Назва елемента	Призначення
1	2	3	4
1		Фоторезистор	Фоторезистор – це напівпровідниковий пристрій, опір якого змінюється в залежності від інтенсивності освітлення, до якої він викладений. Іншими словами, опір фоторезистора зменшується при збільшенні освітленості і збільшується при зменшенні освітленості. Цей принцип роботи дозволяє використовувати фоторезистори для вимірювання і контролю

			освітленості в різних додатках, таких як автоматичні системи освітлення, фотодетектори та інші пристрої.
2		Фотодіод	Основа дії фотодіодних елементів - внутрішній фотоэффект. Він полягає у виникненні в напівпровіднику під впливом світлового потоку нерівноважних електронів і дірок, які формують фотоелектрорушійну силу.
3		Фототранзистор (датчик освітленості)	Фототранзистори є твердотільними пристроями з внутрішнім підсиленням, які використовуються для сприймання світла і перетворення його на електричний сигнал. Цей пристрій має конструкцію транзистора і внутрішній фоточутливий матеріал, що забезпечує підсилення світлового сигналу.
4		Інфрочервоний датчик	Використовується для роботи з пультом керування.
5		Ультразвуковий датчик HC-SR04 (далекомір)	Далекомір. Цифровий датчик, використовується для визначення відстані до перешкоди або об'єкта.
6		Ультразвуковий датчик (далекомір)	Далекомір. Датчик, використовується для визначення відстані до перешкоди або об'єкта.

7		Піроелектричний інфрачервоний датчик (датчик руху)	Датчик руху.
8		Датчик газу	Датчик газу
9		Датчик нахилу	Датчик нахилу
10		Датчик температури	Аналоговий датчик температури (LM35).

Приклади використання наявних датчиків у [tinkercad.com](https://www.tinkercad.com):

1. Приклад використання аналогового датчика температури:

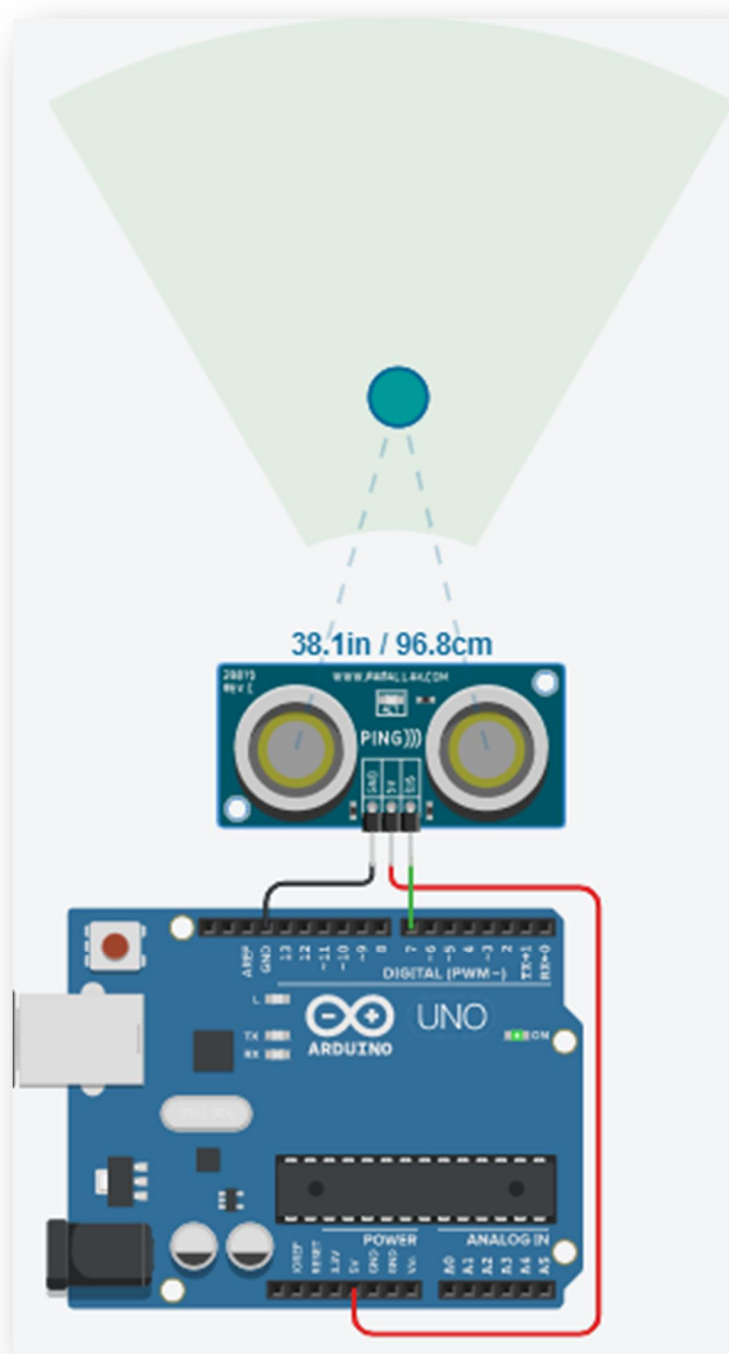


Приклад скетчу, який перетворює сигнал з датчика в значення температури і виводить це значення у послідовний порт.

```
int n;
float mv;
float tmp;
void setup()
{
  pinMode(A2, INPUT);
  Serial.begin(9600);
}
void loop()
{
  n = analogRead(A2);
  mv = n * 5000.0 / 1024.0;
  tmp = (mv - 500.0) / 10.0;
  Serial.println(tmp);

  delay(200);
}
```

2. Приклад використання ультразвукового далекоміру:



Приклад скетчу, який перетворює сигнал з ультразвукового далекоміру і виводить у послідовний порт значення відстані від датчика до перешкоди.

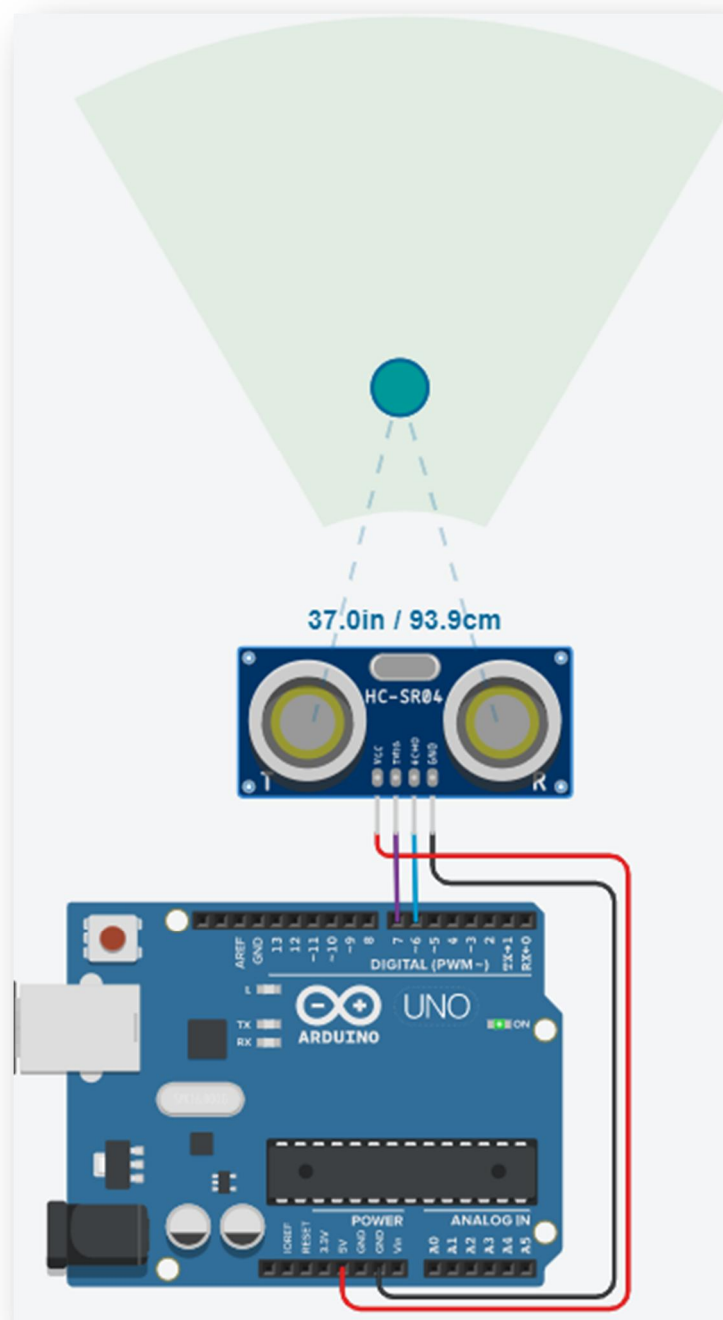
```
long n; //значення яке зчитуємо
int cm; //значення відстані в см
void setup()
{
  Serial.begin(9600);
}
void loop()
{
```

```

pinMode(7, OUTPUT);
digitalWrite(7, LOW);
delayMicroseconds(2);
digitalWrite(7, HIGH);
delayMicroseconds(10);
digitalWrite(7, LOW);
pinMode(7, INPUT);
n = pulseIn(7, HIGH);
cm = 0.01723 * n;
Serial.print(cm);
Serial.println(" cm");
delay(200);
}

```

3. Приклад використання ультразвукового далекоміру HC-SR04:

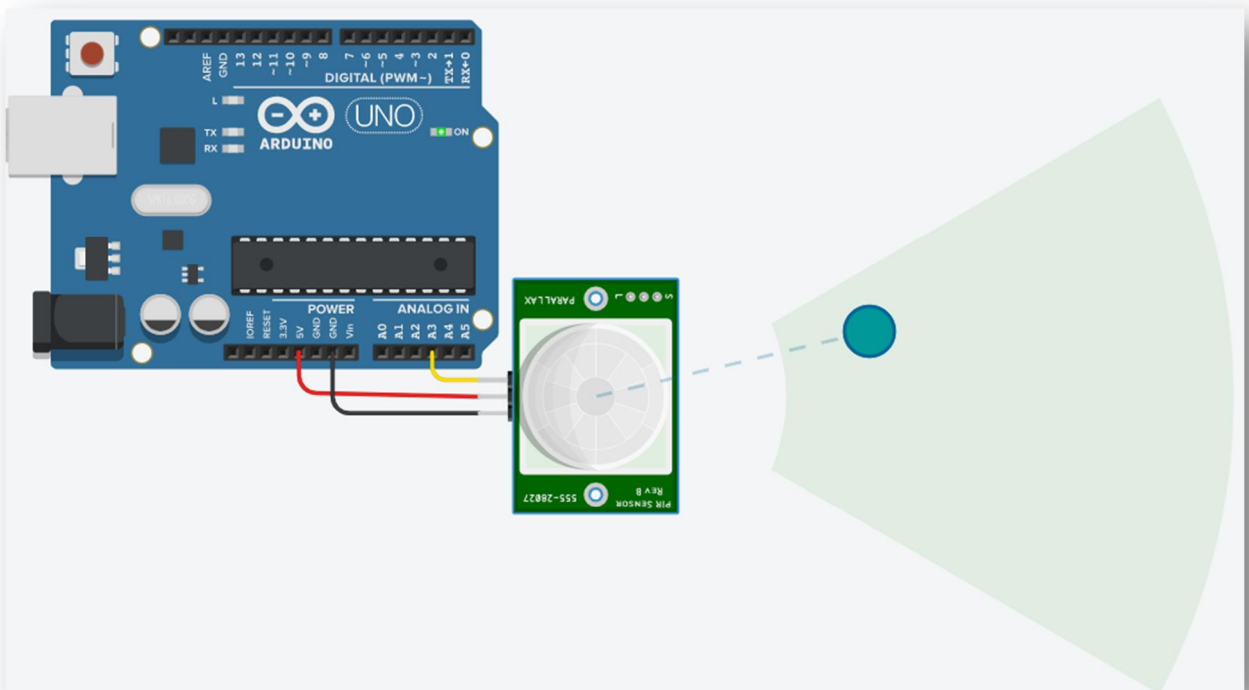


Приклад скетчу, який перетворює сигнал з ультразвукового далекоміру HC-SR04 і виводить у послідовний порт значення відстані від датчика до перешкоди.

```
long n;  
int cm;  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(7, OUTPUT);  
  pinMode(6, INPUT);  
}  
void loop()  
{  
  digitalWrite(7, LOW);  
  delayMicroseconds(2);  
  digitalWrite(7, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(7, LOW);  
  n = pulseIn(6, HIGH);  
  cm = 0.01723 * n;  
  Serial.print(cm);  
  Serial.println(" cm");  
  delay(200);  
}
```

4. Приклад використання датчика руху:

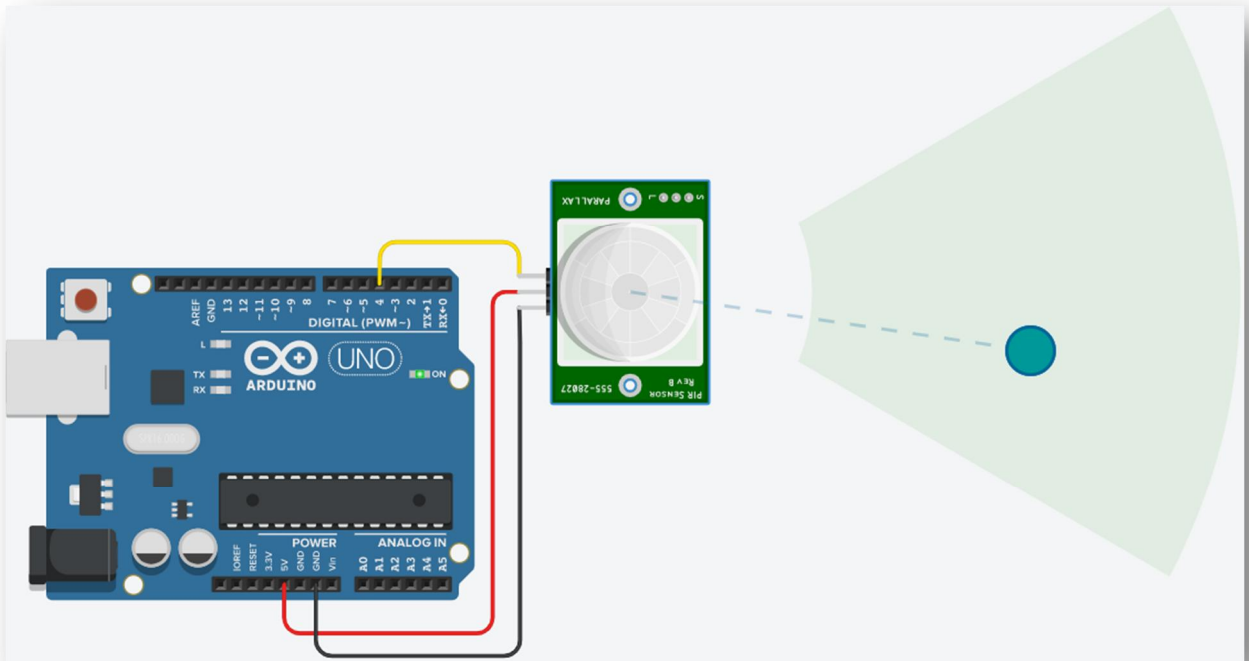
Підключення датчика руху до аналогового входу Arduino.



Приклад скетчу, який дозволяє зчитати сигнал з датчика руху і вивести його реакцію на рух об'єкта у послідовний порт.

```
int n;
void setup()
{
  pinMode(A3, INPUT);
  Serial.begin(9600);
}
void loop()
{
  n = analogRead(A3);
  Serial.println(n);
  delay(200);
}
```

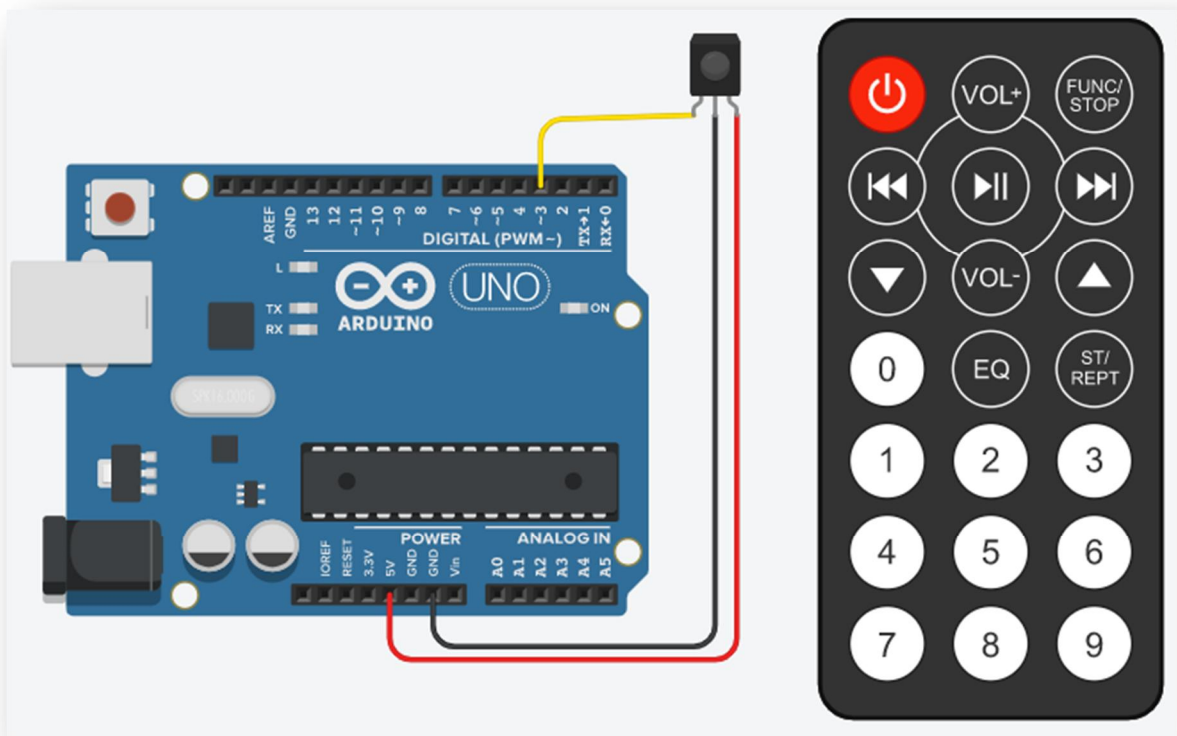
Підключення датчика руху до цифрового входу Arduino.



```
int n;

void setup()
{
  pinMode(4, INPUT);
  Serial.begin(9600);
}
void loop()
{
  n = digitalRead(4);
  Serial.println(n);
  delay(200);
}
```

5. Приклад використання інфрачервоного датчика і пульта керування:



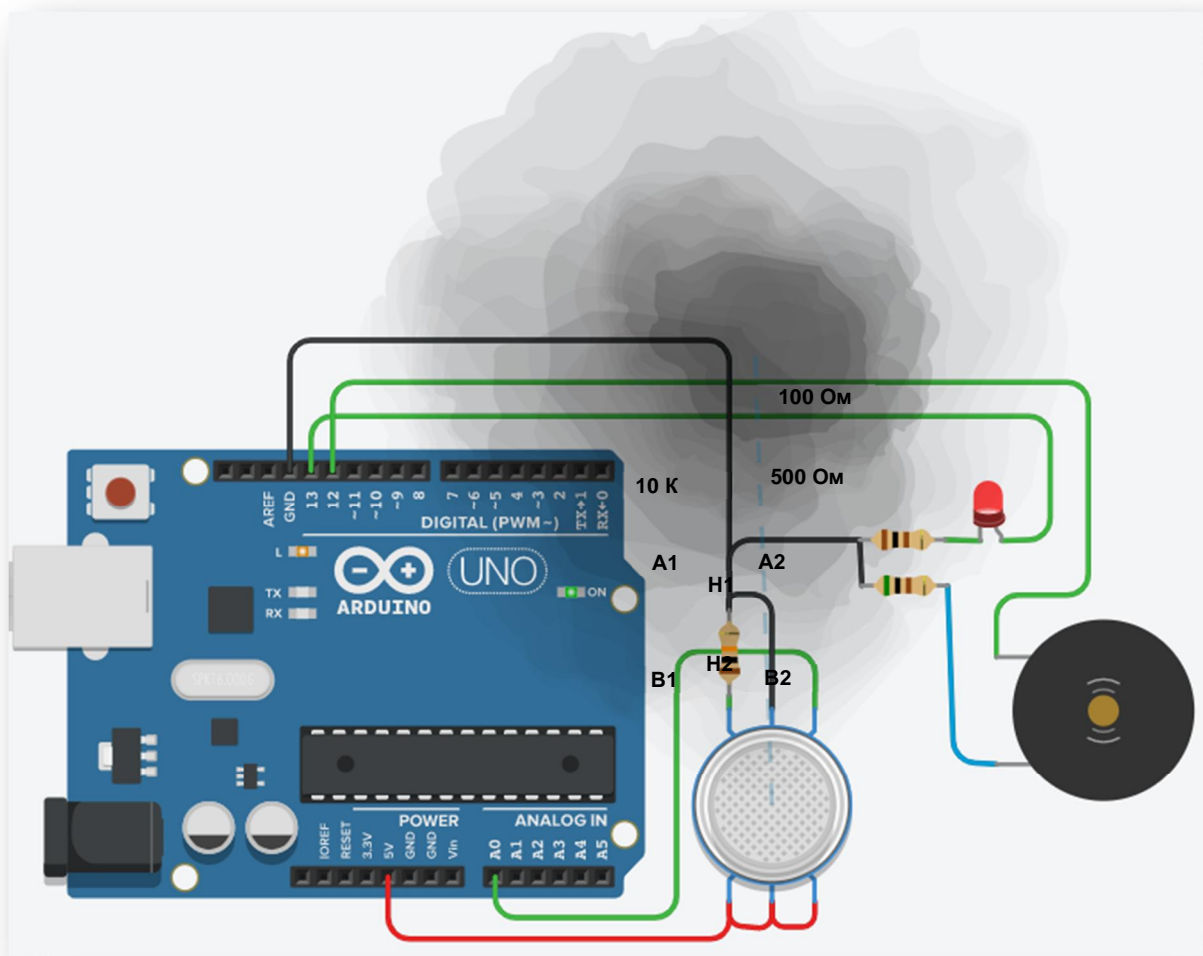
Приклад скетчу, який дозволяє зчитати сигнал з інфрачервоного датчика, який був переданий за допомогою інфрачервоного пульта керування і вивести код натиснутої кнопки пульта у послідовний порт.

```
#include <IRremote.h> //бібліотека для роботи з пультом
IRrecv IR(3);
decode_results RES;

void setup()
{
  Serial.begin(9600); // встановлюємо швидкість послідовного порта
  IR.enableIRIn(); // запуск зчитування
}

void loop()
{
  if ( IR.decode( &RES ) ) { // якщо отримали дані
    Serial.println( RES.value, HEX ); // виводимо отримані дані
    IR.resume(); // приймаємо наступну команду з пульта
  }
}
```

6. Приклад використання датчика газу:



Приклад скетчу, який дозволяє зчитати сигнал з датчика газу, і при збільшенні його концентрації до певного значення вмикається звукова і світлова індикація. Умовне значення концентрації газу передається в послідовний порт.

Для зміни концентрації газу необхідно змінити місце розташування чорної хмари.

```
int smokeA0 = A0;
int Led_Pin = 13;
int buzzer = 12;
int sensorThres = 500;

void setup() {
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop() {
  int analogSensor = analogRead(A0);
```



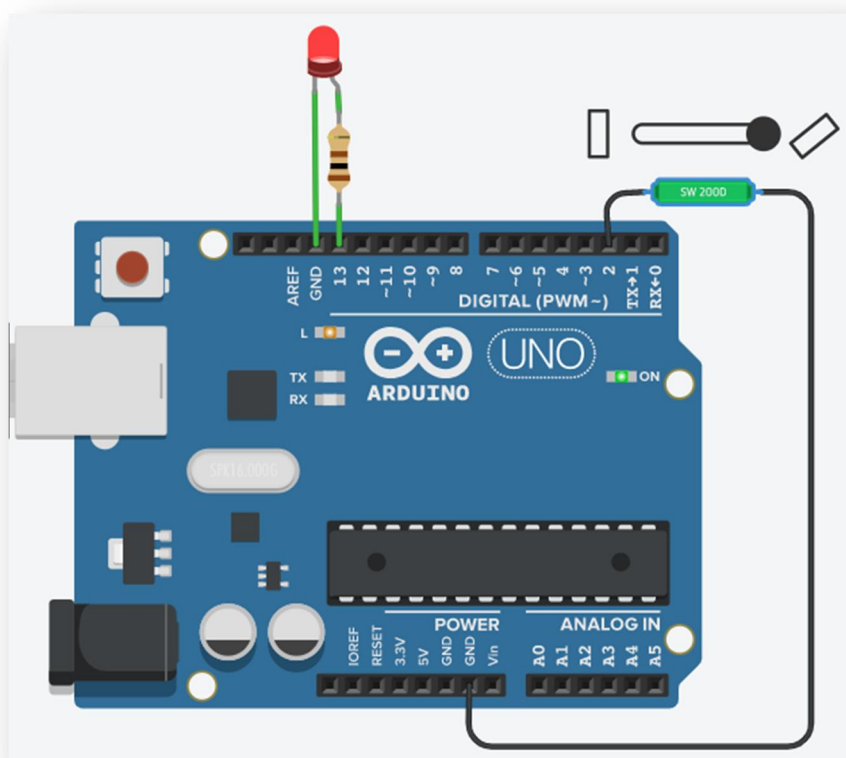
```

Serial.print(A0);
Serial.println(analogSensor);

if (analogSensor > sensorThres)
{
  tone(buzzer, 1000, 1000);
  digitalWrite(13, HIGH);
}
else
{
  noTone(buzzer);
  digitalWrite(13, LOW);
}
delay(100);
}

```

7. Приклад використання датчика нахилу:



Приклад скетчу, який дозволяє зчитати сигнал з датчика, і при його нахилі увімкнути світлодіод.

У схемі використовуються: датчик нахилу, світлодіод, резистор 100 Ом.

```

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(2, INPUT_PULLUP);
}
void loop()
{

```

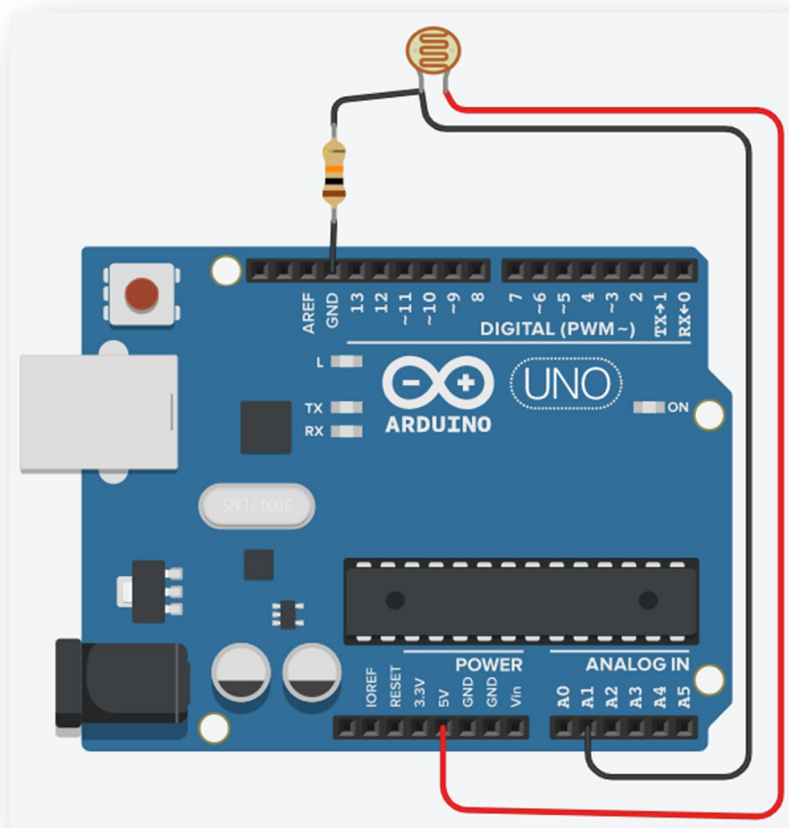
```

if(digitalRead(2)==LOW) digitalWrite(13, HIGH);
else digitalWrite(13, LOW);    }

```



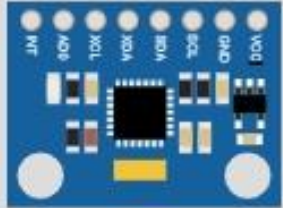

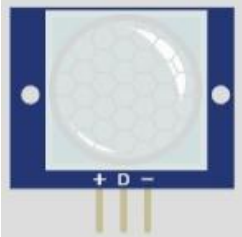
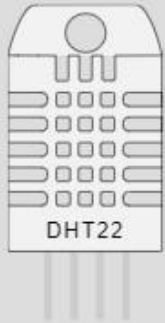
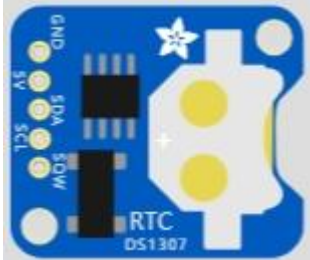
8. Приклад використання фоторезистора:

У схемі використовується фоторезистор у якості датчика. До фоторезистора послідовно підключено резистор на 10 Ком, до вільних виводів резистора і фоторезистора підведена напруга, таким чином ці два елементи створюють ділянку напруги. Значення напруги у середній точці буде залежати від освітленості фоторезистора. При збільшенні освітленості фоторезистора, його опір буде зменшуватися і відповідно напруга у центральній точці з'єднання буде збільшуватися.



Онлайн сервіс **Wokwi** (<https://wokwi.com/>) на відмінність від сервісу **tinkercad.com** має дещо інші датчики у своєму наборі:

№	Зображення	Назва елемента	Призначення
1	2	3	4
1		ntc-temperature-sensor	Аналоговий датчик температури. Модуль датчика температури включає термістор 10K NTC послідовно з резистором 10K

2		Ультразвуковий датчик HC-SR04 (далекомір)	Далекомір. Цифровий датчик, використовується для визначення відстані до перешкоди або об'єкта.
3		photoresistor-sensor	Модуль датчика фоторезистора містить LDR (світлозалежний резистор), з'єднаний послідовно з резистором 10К. Вивід АО підключається між LDR і резистором 10К.
4		mpu6050 6-Axis Accel & Gyro Sensor	Інтегрований датчик із 3-осьовим акселерометром, 3-осьовим гіроскопом і датчиком температури з інтерфейсом I2C.
5		DS18B20	Цифровий датчик температури.
6		pir-датчик руху	Пасивний інфрачервоний (PIR) датчик руху
7		DHT-22	Цифровий датчик вологості та температури
8		Годинник реального часу	Модуль RTC (Годинник реального часу) з інтерфейсом I ² C і 56 байтами NV SRAM

Розглянемо приклади роботи з тими датчиками, які частіше всього використовуються в різних проєктах.

Датчик DHT22.

Датчик DHT22 є комбінованим пристроєм, який включає в себе два основних компонента: ємнісний датчик температури та гігрометр для вимірювання вологості. Цей датчик може перетворювати оточуючі параметри в цифрові вимірювання, які легко інтерпретувати за допомогою мікроконтролера.

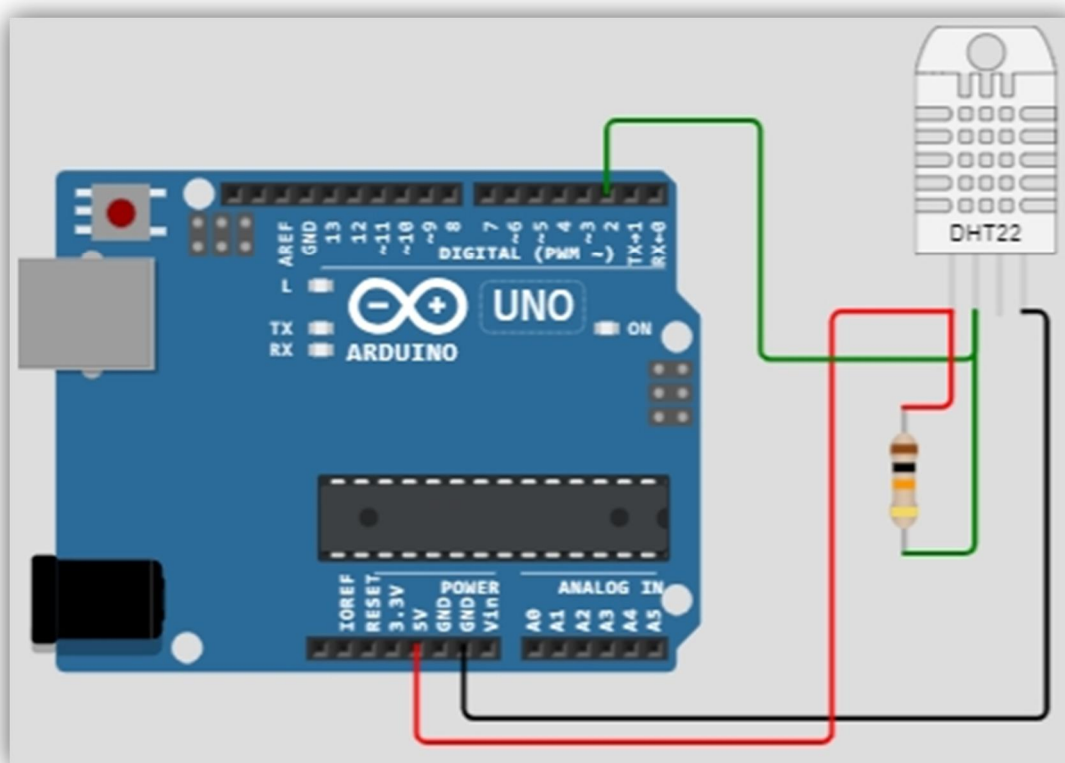
Живлення датчика від 3 до 5 вольт, що робить його сумісним з більшістю сучасних мікроконтролерів. Датчик може вимірювати вологість в інтервалі від 0% до 100%. Точність вимірів коливається від 2% до 5%, залежно від конкретної моделі. Діапазон вимірюваної температури: від мінус 40 до плюс 125 градусів за Цельсієм, з точністю вимірювань приблизно 0,5 градусів.

Швидкість вимірювань – одне вимірювання за 2 секунди з частотою до 0,5 герца.

Під час підключення датчика, між 1 і 2 пінами, рекомендується встановити резистор з опором 10 Ком. Цей датчик забезпечує можливість вимірювати як температуру, так і вологість, що робить його корисним у багатьох додатках, таких як системи контролю клімату, погодні станції та інші області, де важливо знати ці параметри.

Збираємо схему у **Wokwi** (<https://wokwi.com/>)

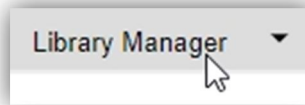
9. Приклад використання датчика DHT22



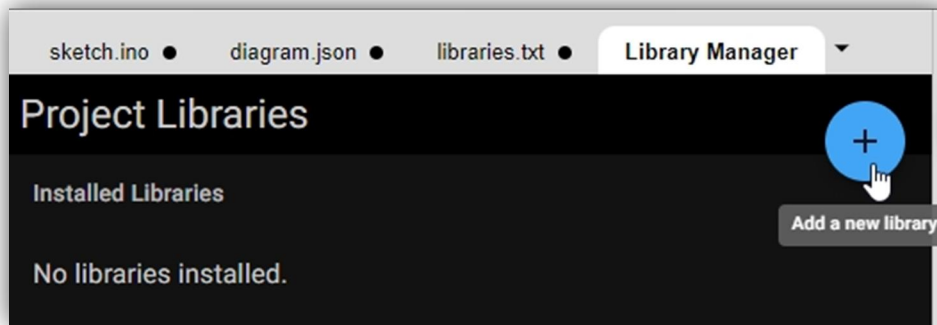
Для перевірки роботи датчика можна скористатися наступним скетчем (для роботи скетча треба бібліотека **DHT.h**, яку можна встановити у

Wokwi за допомогою *менеджера бібліотек*, а якщо ми набираємо текст скетча в *Arduino IDE*, то бібліотеку можна скачати за посиланням <https://github.com/adafruit/DHT-sensor-library>):

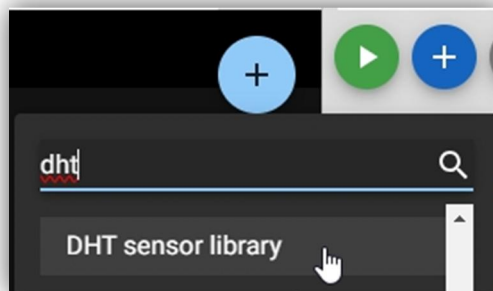
Для того щоб встановити бібліотеку у *Wokwi* треба:



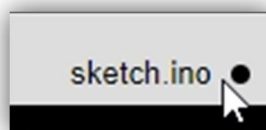
1. У меню вибираємо меню:
2. Натискаємо на плюс:



3. У рядку пошуку набираємо назву бібліотеки, і з представленого списку вибираємо **“DHT sensor library”**



4. Після встановлення бібліотеки повертаємося до нашого скетча, для цього на вибираємо пункт меню:



```
#include "DHT.h"

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("test - DHT22");
}
```

```

dht.begin();
}

void loop() {

float h = dht.readHumidity();
float t = dht.readTemperature();

if (isnan(t) || isnan(h)) {
Serial.println("Failed to read from DHT");
}
else {
Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.println(" *C");
}
}

```

Після запуску, у послідовний порт будуть виводитися значення вологості і температури:

Humidity: 64.50 %	Temperature: 32.60 *C
Humidity: 64.50 %	Temperature: 32.60 *C
Humidity: 64.50 %	Temperature: 32.60 *C

Клацнувши мишкою на датчик, можна змінити параметри температури і вологості.

Датчик DS18B20

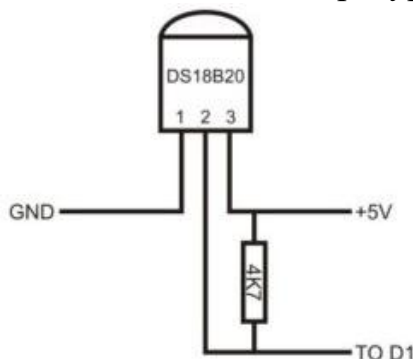
DS18B20 – це цифровий температурний датчик, який представляє собою досить потужний пристрій, оснащений різними корисними функціями. Фактично, DS18B20 є мікроконтролером, який має здатність зберігати вимірювані значення температури, сповіщати про виходження температури за певні межі (*при цьому межі можна налаштувати*), змінювати точність вимірювань, налаштувати спосіб взаємодії з контролером та виконувати багато інших функцій. Усі ці можливості реалізовані в досить невеликому корпусі, і доступні водонепроникні варіанти цього датчика.

DS18B20 може бути використаний для точного вимірювання температури в різних застосуваннях, таких як системи контролю клімату, моніторинг погодних умов, вимірювання температури обладнання та інше. До плюсів цього датчика слід віднести його гнучкість та можливість налаштування параметрів вимірювань відповідно до потреб конкретного приладу.

DS18B20 – це цифровий температурний датчик, який має кілька особливостей. Він використовує цифровий інтерфейс "1-Wire", що дозволяє підключати багато датчиків до одного контролера через один пін. Датчик має широкий діапазон вимірювання температури від -55°C до $+125^{\circ}\text{C}$, що робить його універсальним для різних застосувань. Точність вимірювань зазвичай становить $\pm 0.5^{\circ}\text{C}$ при 25°C , і її можна регулювати залежно від потреб. Існує водонепроникний варіант датчика для використання у вологих умовах. Важливою перевагою є заводська калібровка, завдяки чому багато датчиків готові до використання без додаткового налаштування. DS18B20 доступний в корпусі TO-92, який зручно використовувати в різних додатках. Цей датчик застосовується в системах контролю клімату, промислового вимірюванні температури, метеостанціях, контролі температури в приміщеннях та інших сферах. DS18B20 має зручний інтерфейс, який спрощує підключення та отримання даних.

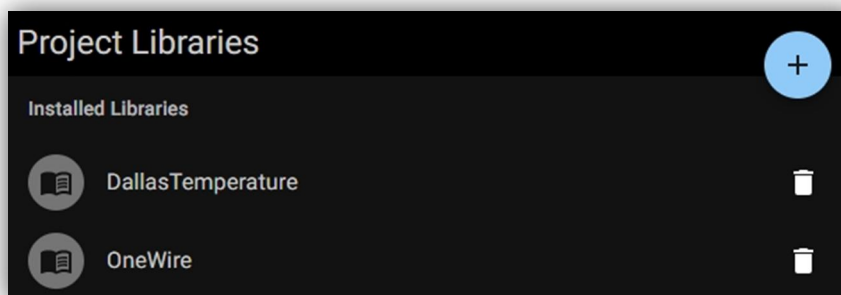


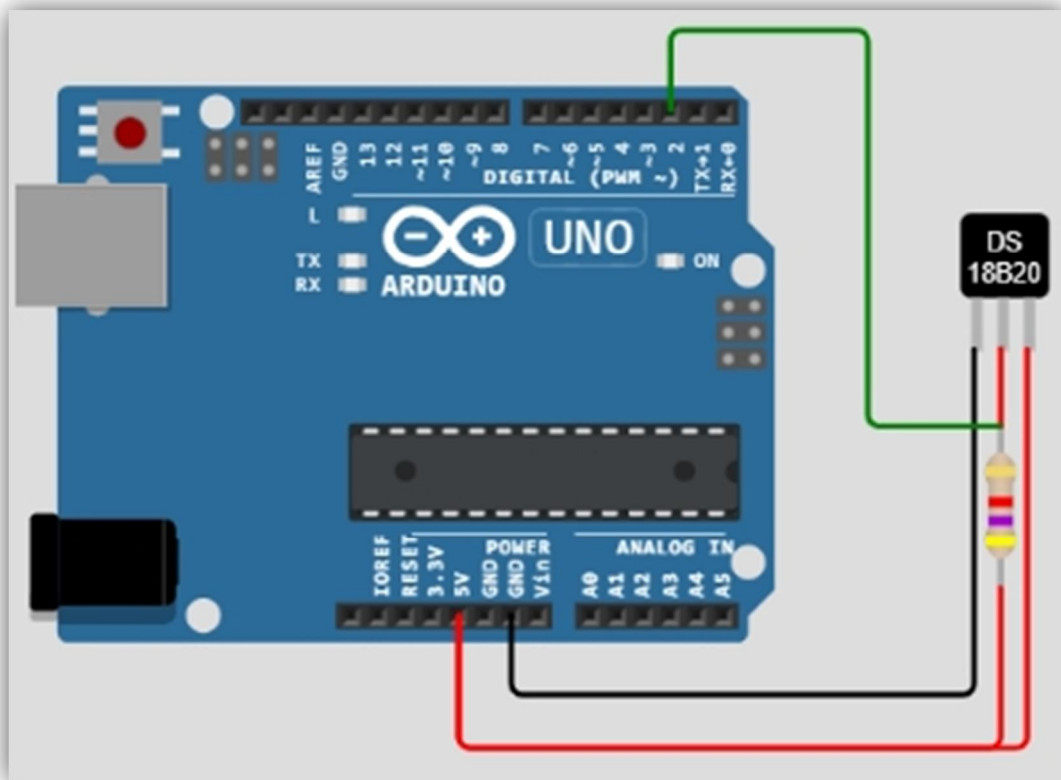
Схема підключення даного датчика температури зображена на малюнку:



10. Приклад використання датчика DS18B20

Для роботи з датчиком DS18B20 необхідно за допомогою менеджера бібліотек встановити такі бібліотеки: `OneWire.h` і `DallasTemperature.h`





```

#include "OneWire.h"
#include "DallasTemperature.h"

OneWire oneWire(2); // порт підключення датчика
DallasTemperature ds(&oneWire);

void setup() {
  Serial.begin(9600); // ініціалізація монітора порта
  ds.begin(); // ініціалізація датчика ds18b20
}

void loop() {
  ds.requestTemperatures(); // зчитуємо температуру з датчика

  Serial.print(ds.getTempCByIndex(0)); // виводимо температуру на
монітор
  Serial.println("C");
}

```

Після запуску, у послідовний порт будуть виводитися значення температури. А якщо клацнути мишкою на датчик, можна змінити параметри температури.

ПОРЯДОК ВИКОНАННЯ РОБОТИ.

1. Використовуючи сервіс tinkercad.com перевірте працездатність усіх схем і скетчів наданих у прикладах.

2. До схеми «**Приклад використання датчика температури**» додайте рідкокристалічний дисплей. Виведіть на екран дисплея у першому рядку ваше прізвище англійськими літерами, а у другому рядку значення температури. Зробіть скріншот схеми, таким чином щоб було видно інформацію на дисплеї і додайте це зображення разом з текстом програми у звіт.

3. До схеми «**Приклад використання ультразвукового далекоміру HC-SR04**» додайте рідкокристалічний дисплей, два резистори на 100 Ом, один зелений світлодіод, один червоний світлодіод. Змініть скетч таким чином, щоб програма виводила на екран у першому рядку ваше прізвище англійськими літерами, а у другому рядку - відстань до перешкоди. Якщо відстань більше 50 см то світиться зелений світлодіод, а якщо об'єкт знаходиться ближче – то світиться червоний світлодіод. Зробіть відповідні скріншоти, додайте їх разом з текстом програми у звіт.

4. До схеми «**Приклад використання інфрачервоного датчика і пульта керування**» додайте два резистори на 100 Ом, один зелений світлодіод, один червоний світлодіод. Змініть скетч таким чином, щоб використовуючи пульт керування можна було вмикати і вимикати світлодіоди різними кнопками. Зробіть відповідні скріншоти, додайте їх разом з текстом програми у звіт.

5. До схеми «**Приклад використання датчика руху**» додайте світлодіод і резистор на 100 Ом. Змініть скетч таким чином, щоб при спрацюванні датчика руху вмикався світлодіод і світився протягом 5 секунд.

6. До схеми «**Приклад використання датчика DS18B20**» підключіть LCD дисплей і виведіть значення температури на другий рядок.

7. Запропонуйте власну схему, у якій використовуються декілька датчиків і різні елементи індикації.

ПИТАННЯ ДЛЯ ПІДГОТОВКИ.

1. Які існують аналогові датчики?
2. Коли застосовують датчики руху?
3. Який принцип дії датчиків прискорення?
4. Як користуватися монітором послідовного порта?

Лабораторна робота №11 (Підключення до плати Arduino двигунів та сервоприводів)

Тема: Підключення до плати Arduino двигунів та сервоприводів.

Мета: Навчитися підключати та запускати різні двигуни та сервоприводи до платформи Arduino і використовувати їх у власних розробках.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

Мотор постійного струму з редуктором 1:48 є популярним компонентом у світі робототехніки, особливо в проектах, де потрібно забезпечити рух робота. Він здатен забезпечувати достатню силу тяги для руху навіть у невеликих роботах. Цей мотор часто використовується у поєднанні з платформою Arduino, що робить його ідеальним для початківців у робототехніці.



Споживання даного мотора становить 250 мА (при напрузі 3,6 В). Це означає, що пряме управління з виводів Arduino в даному випадку є недоречним. А якщо врахувати, що в більшості проектів необхідно мінімум два таких мотора, то завдання стає зовсім неможливим.

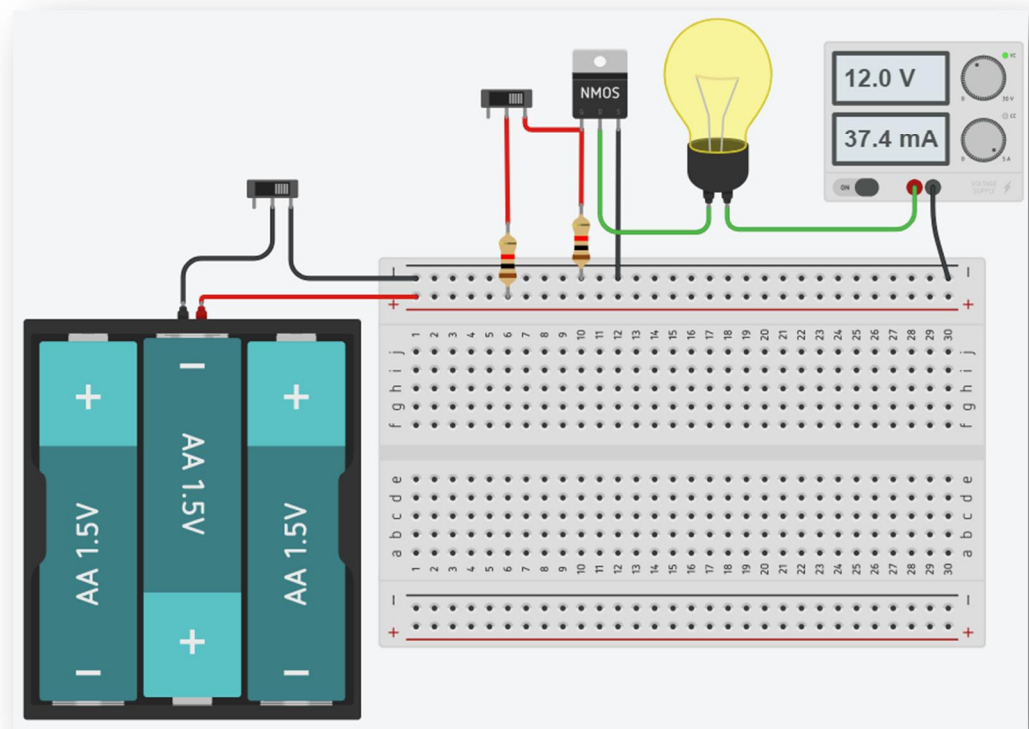
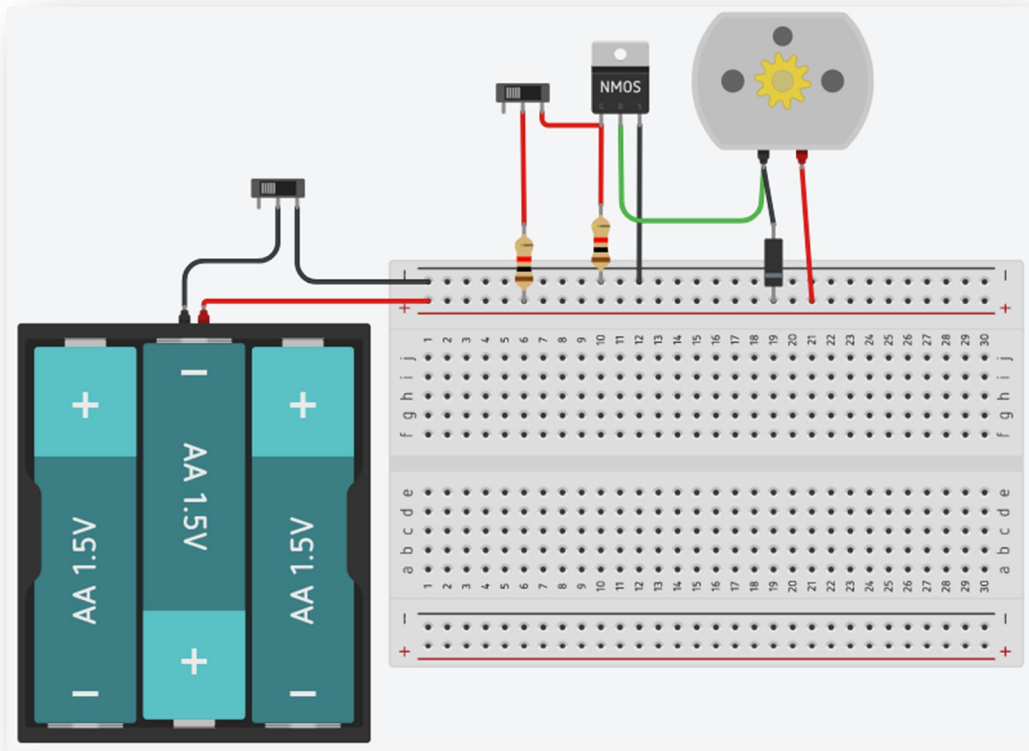
Реалізувати роботу моторів можна за допомогою польового транзистора (MOSFET) з логічним рівнем управління затвором, наприклад IRL540N. MOSFET це транзистор, але особливого типу.

Транзистори мають 3 виводи для підключення і мають дві прості функції, перша - перемикання, друга - підсилення. Виходи називаються наступним чином: Вхід (Source), Вихід (Drain), Управління (Gate).

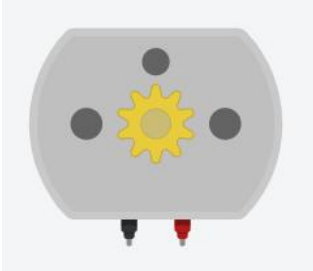
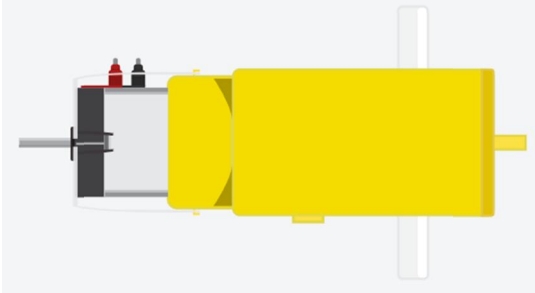
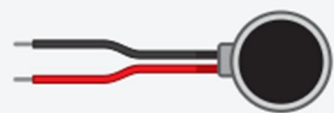
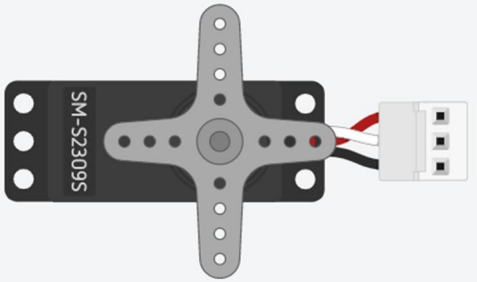
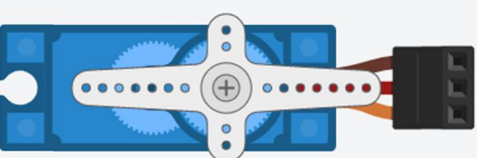
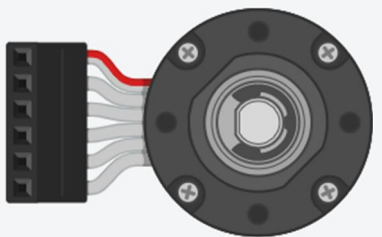
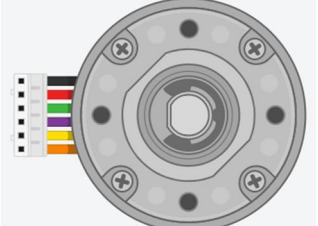
При наявності сигналу високого рівня на Gate, транзистор включається і дозволяє струму проходити від Source до Drain.

Таким чином можна підключити мотор, соленоїд, лампу або нагрівач до виводів Arduino. Пристрій підключається одним виводом до V+, а інший не до землі (V-), а до транзистора. до виводу Drain.

Коли Arduino посилає сигнал високого рівня на Gate транзистора, він з'єднує Source і Drain тим самим замикає коло для двигуна, соленоїда або лампи.

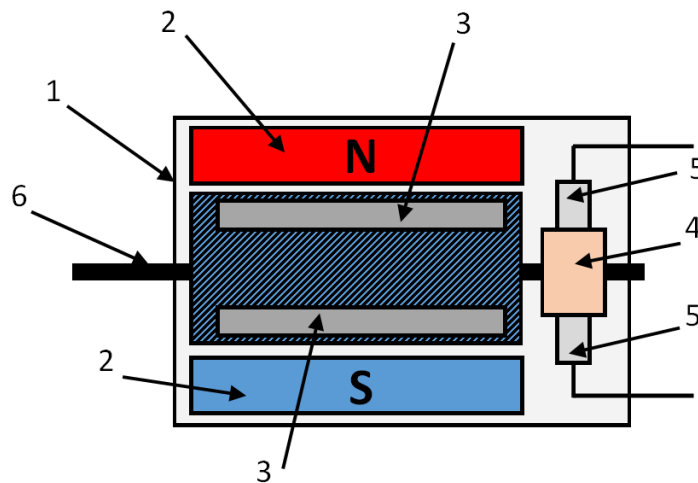


Беручи за увагу той факт, що в сервісі **tinkercad.com** на сьогодні у наявності є тільки такі двигуни і сервоприводи, що можна використовувати у власних розробках:

№	Зображення	Назва елемента та його призначення
1		Двигун постійного струму.
2		Двигун постійного струму з редуктором.
3		Вібродвигун.
4		Сервопривод SM-S2309S
5		Сервопривод
6		Двигун постійного струму з енкодером (макс. 2737 об/хв)
7		Двигун постійного струму з енкодером (макс. 1621 об/хв)

Як працює колекторний двигун:

Двигун постійного струму складається з корпусу (1), двох постійних магнітів (2), які приклеєні до корпусу двигуна (1), котушки (3), яка знаходиться на валу двигуна (6), колектора (4) і двох щіток (5).



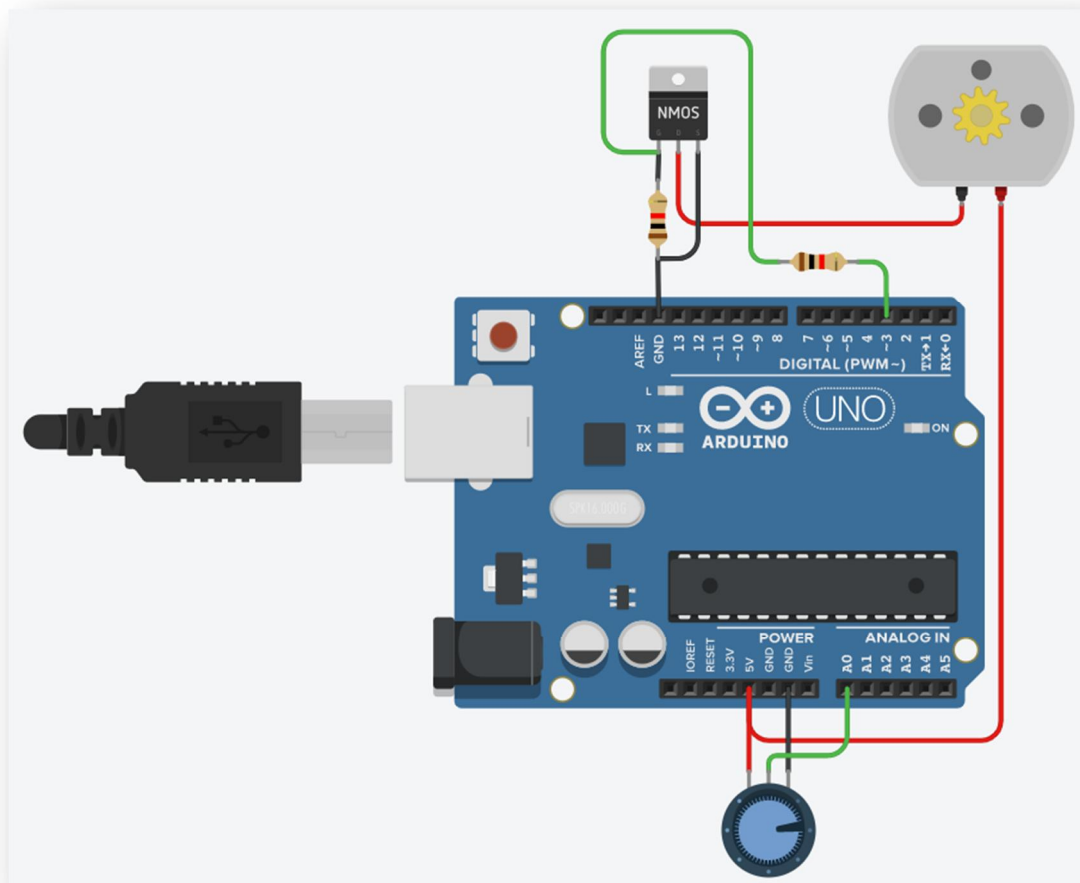
Колектор складається не з однієї суцільної металевої пластини, а з декількох паралельних пластин, розділених ізоляторами, завдяки цьому при включенні двигуна по мірі обертання ротора змінюється полярність підключення котушки і за рахунок чого відбувається рух.



Тобто спочатку до магніту притягується одна сторона котушки, а потім вона відштовхується.

Для забезпечення більш плавного руху збільшують кількість катушок на роторі.

Приклад підключення колекторного двигуна до плати Arduino показано на малюнку:



Приклад скетчу, який дозволяє змінювати швидкість обертання ротора двигуна за допомогою змінного резистора:

```
#define motorPin 3
void setup(){
  pinMode(motorPin, OUTPUT);
  pinMode(A0, INPUT);
}
void loop(){
  int sensorValue = analogRead(A0);
  int outputValue = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(motorPin, outputValue);
  delay(100);
}
```

Сервопривод.

Сервопривод – це механізм з вбудованим електромотором, який забезпечує точне керування певними параметрами руху. Він працює на основі негативного зворотного зв'язку, що дозволяє утримувати задані значення параметрів руху. Сервоприводи можуть бути використані з будь-яким механічним приводом і зазвичай мають в собі датчик для вимірювання параметрів, таких як положення, швидкість або зусилля.

Принцип роботи сервопривода полягає в тому, що він отримує зовнішнє значення параметра, який потрібно керувати, наприклад, кут повороту. Блок керування порівнює це значення зі значенням на своєму внутрішньому датчику. На основі результату порівняння сервопривід виконує відповідну дію, таку як поворот, прискорення або уповільнення, щоб забезпечити якнайближче співпадіння внутрішнього параметра зі значенням зовнішнього керуючого параметра.

Серед найпоширеніших видів сервоприводів є ті, що утримують заданий кут або позицію, і ті, що підтримують задану швидкість обертання.



Сервоприводи розрізняють за розмірами. Офіційної класифікації немає, але виробники дотримуються кількох розмірів із загальноприйнятим розташуванням кріпильних елементів.

Їх можна розділити на:

- маленькі
- стандартні
- великі

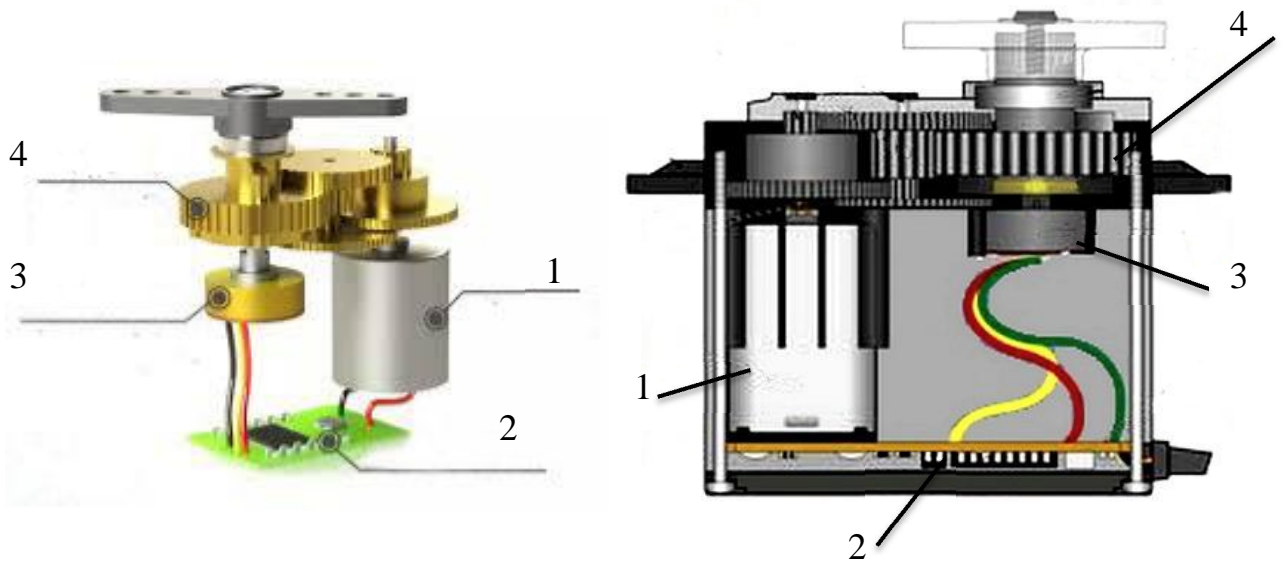
Мають вони при цьому такі характерні габарити:

	Розмір	Вага	Лінійні розміри
1	маленькі	8-25 г	22×15×25 мм
2	стандартні	40-80 г	40×20×37 мм
3	великі	50-90 г	49×25×40 мм

Сервопривід (сервомотор) є важливим елементом при конструюванні різних роботів та механізмів. Це точний виконавець, який має зворотний зв'язок, що дозволяє точно керувати рухами механізмів. Іншими словами, отримуючи на вході значення сигналу, що управляє, сервомотор прагне підтримувати це значення на виході свого виконавчого елемента.

Сервоприводи широко використовуються для моделювання механічних рухів роботів. Сервопривід складається з датчика (швидкості, положення тощо), блоку керування приводом з механічної системи та електронної схеми. Редуктори (шестірні) пристрою виконують із металу, карбону або пластику. Пластикові шестерні сервомотора не витримують сильних навантажень і ударів.

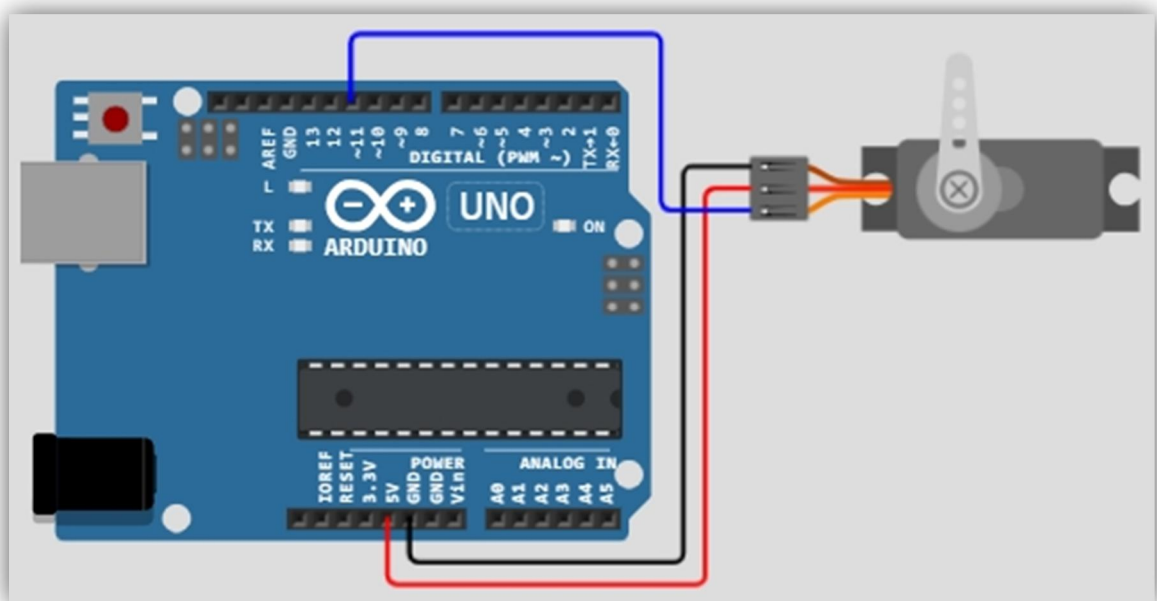
Сервомотор має вбудований потенціометр (3), який з'єднаний із вихідним валом. Поворотом валу, сервопривід змінює значення напруги на потенціометрі. Плата керування (2) аналізує напругу вхідного сигналу і порівнює його з напругою на потенціометрі (3), виходячи з отриманої різниці, мотор (1) плавно обертається до тих пір, поки не вирівняє напругу на виході і на потенціометрі.



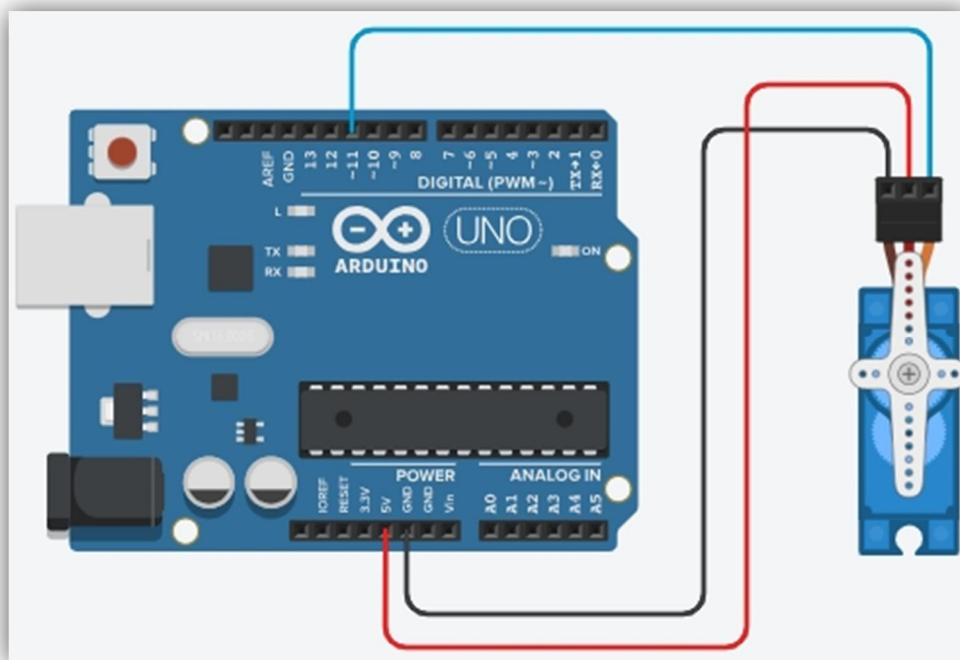
1 – двигун, 2 – плата керування, 3 – потенціометр, 4 – шестерні редуктора.

Схема підключення сервоприводу Arduino зазвичай наступна: чорний провід приєднуємо до GND, червоний провід приєднуємо до 5V, помаранчевий/жовтий провід до цифрового висновку загального призначення. Управління сервоприводом на Ардуїно досить просто, але по кутах повороту сервомотори діляться на 180 і 360, що слід враховувати при розробці проектів із застосуванням бібліотеки **servo**.

Приклад підключення сервоприводу у **Wokwi** (<https://wokwi.com/>):



Приклад підключення сервопривода у **Tinkercad** (<https://tinkercad.com/>):



Під час підключення сервопривода до плати Arduino у **Wokwi** або **Tinkercad** треба уважно дивитися на назви пінів на сервоприводі, бо послідовність може бути різною.

	Назва піна	Призначення
1	PWM	Servo control signal
2	V+	Positive voltage (5V)
3	GND	Ground

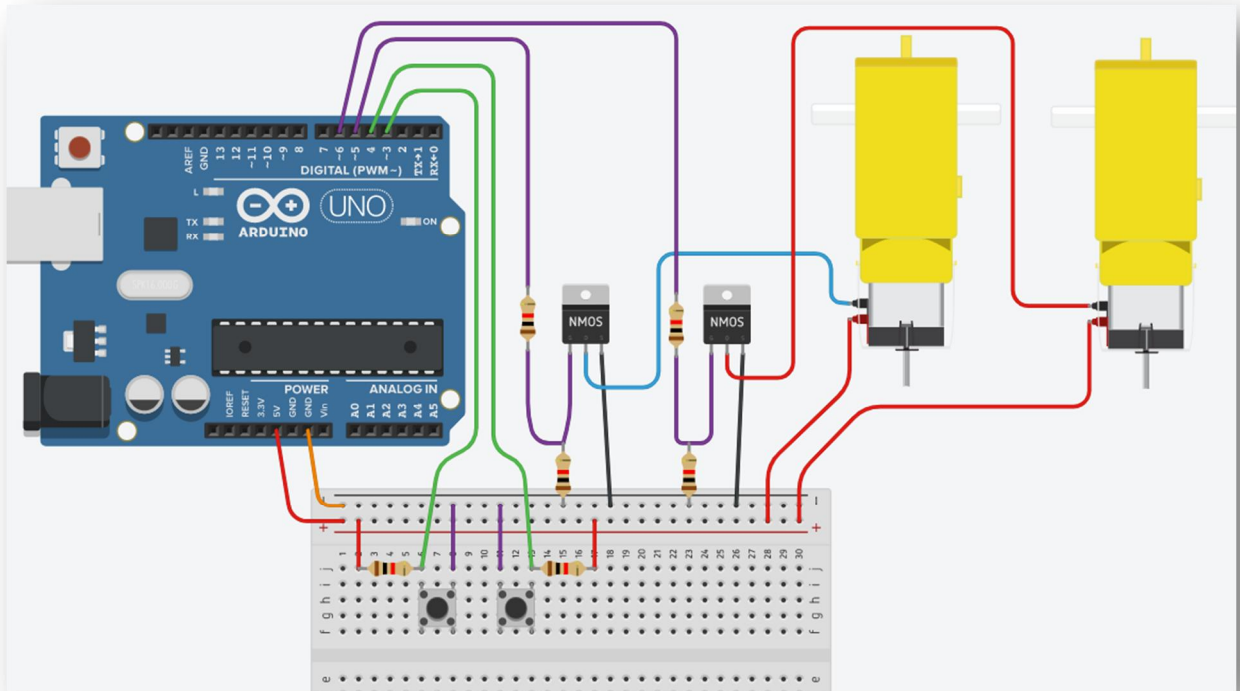
Приклад скетчу, який дозволяє прокрутити вал сервопривода на 180 градусів, а потім повернути назад:

```
#include <Servo.h>
Servo myservo;
int pos = 0;

void setup() {
  myservo.attach(11); //підключення сервопривода до 11 піна плати
  Arduino UNO
}
void loop() {
  for (pos = 0; pos <= 180; pos += 1) {
    myservo.write(pos);
    delay(15); }
  for (pos = 180; pos >= 0; pos -= 1) {
    myservo.write(pos);
    delay(15); }
}
```

ПОРЯДОК ВИКОНАННЯ РОБОТИ.

1. Використовуючи сервіс **tinkercad.com** перевірте працездатність усіх схем і скетчів наданих у прикладах.
2. Підключіть до плати Arduino колекторний двигун і дві кнопки. Напишіть скетч який повинет виконувати такі дії: при натисканні першої кнопки вмикається двигун, а при натисканні другої кнопки двигун вимикається.
3. Приберіть з попередньої схеми одну кнопку. Напишіть скетч який при натисканні кнопки вмикає двигун, а при повторному натисканні кнопки двигун вимикає.
4. Зберіть наступну схему:



Напишіть скетч, який виконує наступні дії: при натисканні лівої кнопки вмикається перший двигун, а при натисканні правої кнопки вмикається другий двигун.

5. Зберіть нову схему. Підключіть до плати Arduino сервопривод і дві кнопки. Напишіть скетч, таким чином, щоб при натисканні одної кнопки вал сервопривода обертався в один бік, а при натисканні другої – в інший. Підключіть до плати LCD дисплей і виведіть числове значення кута на скільки повернувся вал сервопривода.

ПИТАННЯ ДЛЯ ПІДГОТОВКИ.

1. Яка будова двигуна постійного струму?
2. Як підключити колекторний двигун до плати Arduino?
3. Що таке сервопривод?
4. Як підключити сервопривод до плати Arduino?
5. Чим відрізняється сервопривод від колекторного двигуна?
6. Як керувати сервориводом за допомогою плати Arduino?

Лабораторна робота №12 (Підключення крокових двигунів до плати Arduino)

Тема: Підключення крокових двигунів до плати Arduino.

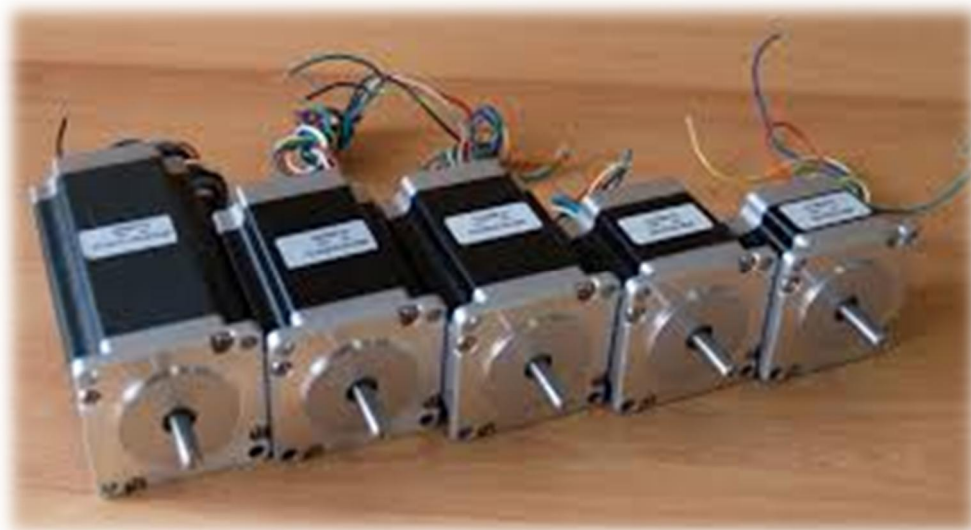
Мета: Навчитися підключати та запускати крокові двигуни до платформи Arduino і використовувати їх у власних розробках.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

Кроковий двигун – це вид електричного двигуна, призначеного для перетворення електричної енергії в механічну енергію. Він складається зі статора та ротора. Однією з важливих особливостей крокових двигунів є їхні можливості виконувати дискретні обертання, при яких певні імпульси відповідають конкретній кількості кроків або кутовому переміщенню.

Крокові двигуни знайшли широке застосування у різних галузях, включаючи верстати з чисельно-програмованим управлінням (ЧПУ), робототехніку, системи зберігання та зчитування інформації. Вони використовуються там, де потрібна висока точність і контроль руху, особливо в автоматизованих системах.

Крокові двигуни можуть керуватися за допомогою спеціального електронного обладнання, яке подає послідовні імпульси для керування їхнім рухом. Залежно від кількості кроків і послідовності імпульсів можна досягти точного переміщення та позиціонування ротора.



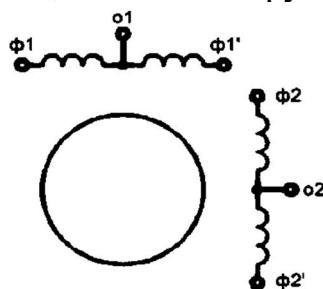
Так, кроковий двигун отримав свою назву через той факт, що він виконує рух не безперервно, а кроками. Кожен крок представляє собою частину повного обертання вала. Кількість кроків, необхідних для завершення повного обертання вала, може варіюватися в залежності від типу крокового двигуна, його схеми з'єднання та методу керування.

Такий дискретний рух дозволяє кроковим двигунам досягати високої точності та контролю над позиціонуванням. Вони широко використовуються в пристроях, де необхідно точне переміщення, таких як принтери, верстати з чисельно-програмованим управлінням (ЧПУ), робототехніка та інші застосування, де точність та контроль руху мають важливе значення.

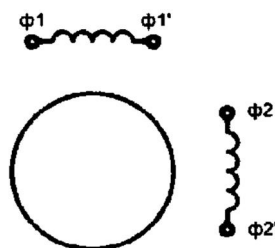
На практиці кроковий двигун є багатофазним мотором. Плавність роботи в якому безпосередньо залежить від кількості обмоток – чим їх більше, тим плавніше відбувається обертання, але від цього і більша вартість. Але при цьому момент, що крутить, від числа фаз не збільшується, хоча для нормальної роботи їх мінімальне число на статорі електродвигуна повинно становити хоча б дві. Кількість фаз не визначає числа обмоток, тому двофазний кроковий двигун може мати чотири і більше обмотки.

Бувають крокові двигуни біполярні та уніполярні.

Уніполярний кроковий двигун відрізняється тим, що у схемі підключення обмотки є вивід від середньої точки. Завдяки чому легко змінюються магнітні полюси. Недоліком такої конструкції є використання лише однієї половини доступних витків, через що досягається менший крутний момент. Тому вони відрізняються великими габаритами, або малим крутним моментом.



Біполярний кроковий двигун підключається до контролера через 4 виводи. При цьому обмотки можуть з'єднуватися як послідовно, так і паралельно.



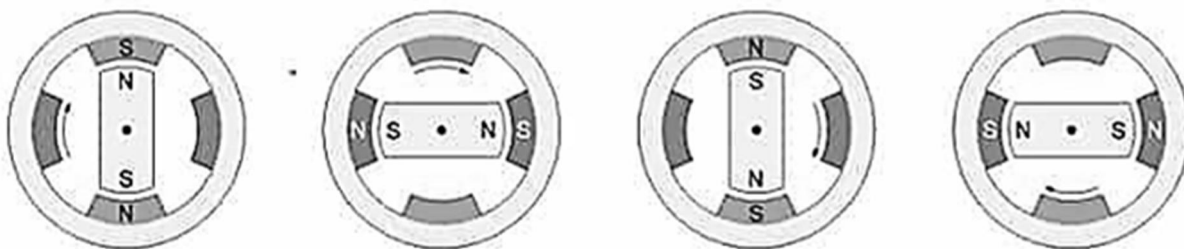
У конструктивній схемі такого двигуна є тільки одна обмотка збудження у кожній фазі. Через це зміна напрямку струму вимагає використання в електронній схемі спеціальних драйверів (*електронних чіпів, призначених для керування*). Домогтися такого ефекту можна за допомогою включення Н-моста. Порівняно з попереднім, біполярний пристрій забезпечує той самий обертальний момент при набагато менших габаритах.

Управління кроковим двигуном

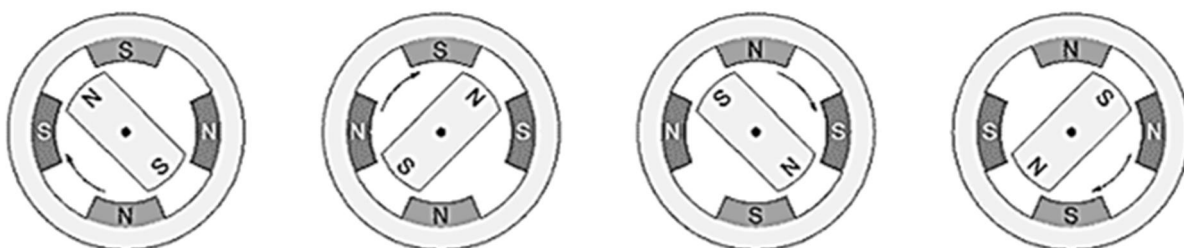
Виконання операцій кроковим агрегатом може здійснюватись кількома методами. Кожен із яких відрізняється способом подачі сигналів на пари полюсів. Усього виділяють три методи активації обмоток.

Хвильовий – у такому режимі відбувається збудження лише однієї обмотки, до якої і притягуються роторні полюси. При цьому кроковий двигун

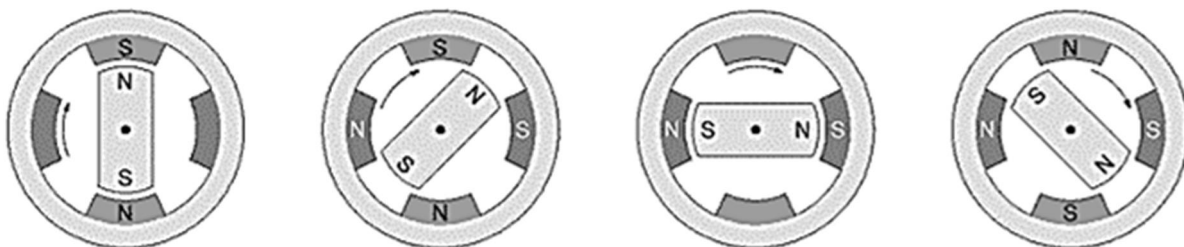
не здатний витягувати велике навантаження, оскільки видає лише половину моменту.



Повнокроковий – у такому режимі відбувається одночасна комутація фаз, тобто збуджуються відразу обидві. Через це забезпечується максимальний момент, у разі паралельного з'єднання або послідовного включення обмоток створюватиметься максимальна напруга або струм.



Напівкроковий – є комбінацією двох попередніх методів комутації обмоток. Під час реалізації якого в кроковому двигуні відбувається почергова подача напруги спочатку одну котушку, а потім відразу дві. Завдяки чому забезпечується краща фіксація на максимальних швидкостях та більша кількість кроків.




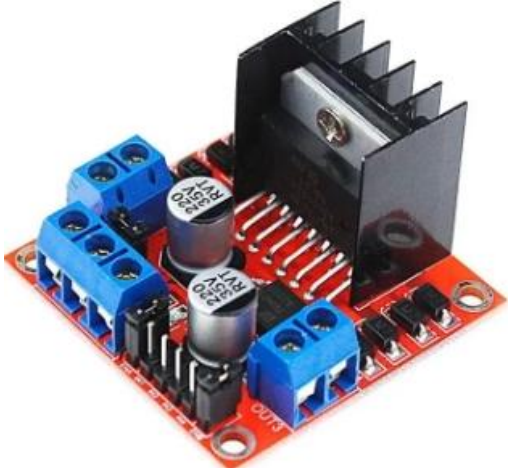

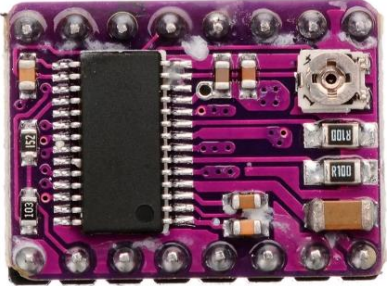
Для більш м'якого керування та подолання інерції ротора використовується мікрокрокове керування, коли синусоїда сигналу здійснюється мікроступеневими імпульсами. За рахунок чого сили взаємодії магнітних ланцюгів у кроковому двигуні набувають більш плавну зміну і переміщення ротора між полюсами. Дозволяє значною мірою знизити ривки крокового двигуна під час роботи.


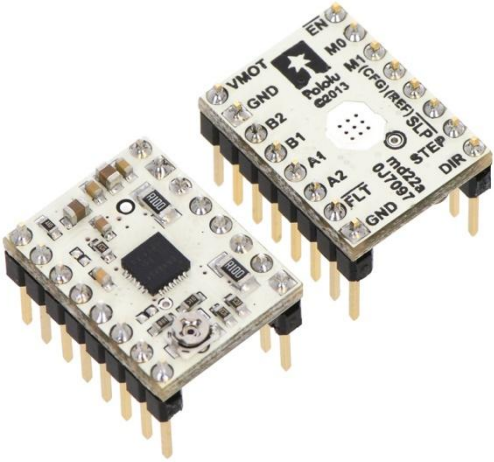
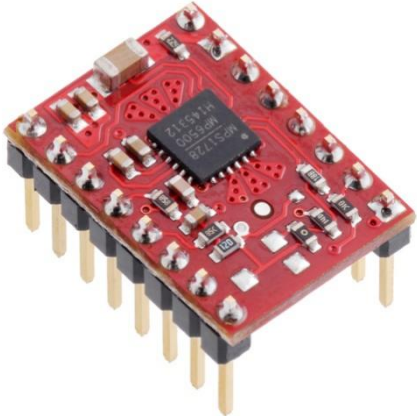
Драйвер крокового двигуна (контролер)

Драйвера моторів і крокових двигунів – це спеціальні модулі, які забезпечують управління пристроями, сприяють їхній ефективній роботі та відстежують працездатність. Мотори та крокові двигуни є непростими конструкціями, яким необхідний особливий контроль у вигляді драйвера. Основне завдання, яке виконує драйвер крокового двигуна – ефективна зміна

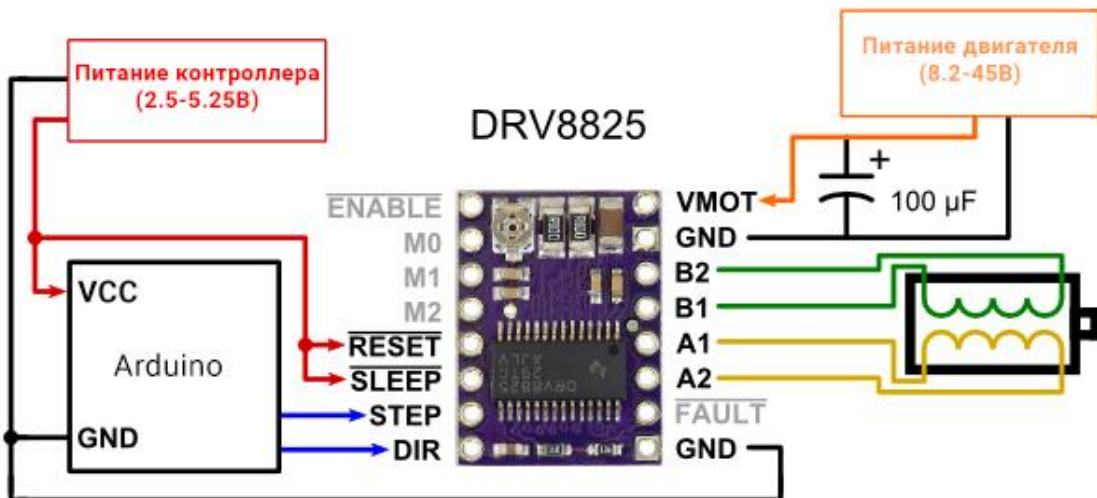
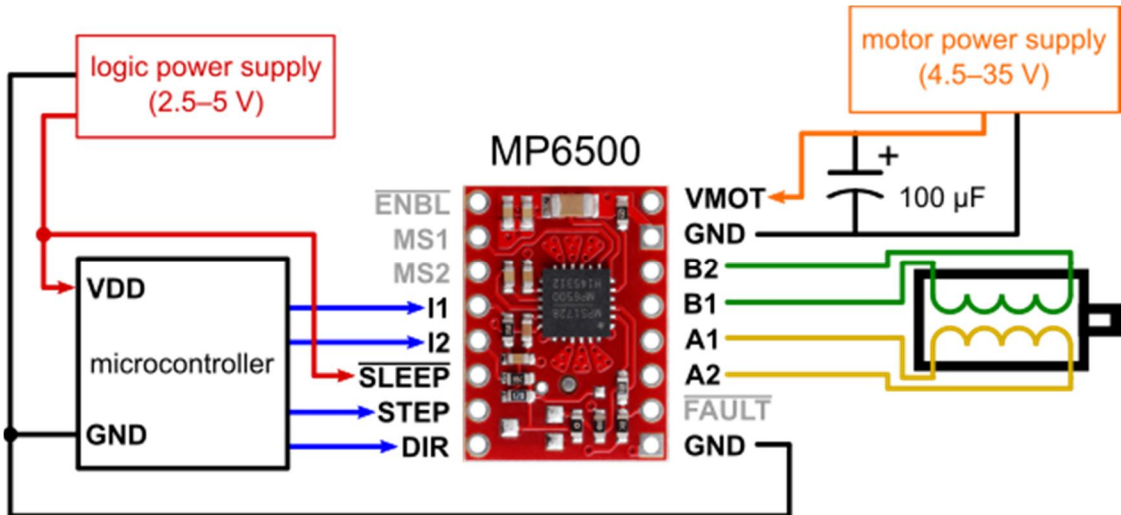
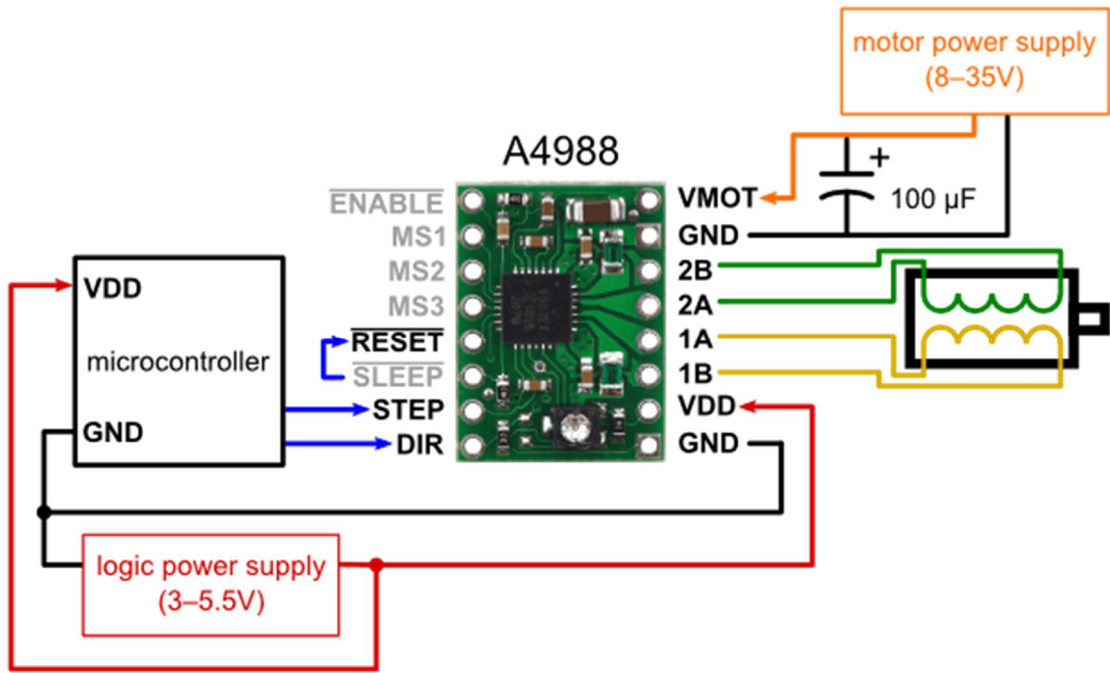
струму в обмотках. Існує кілька типів таких драйверів, які відрізняються залежно від можливостей, функцій та характеристик.

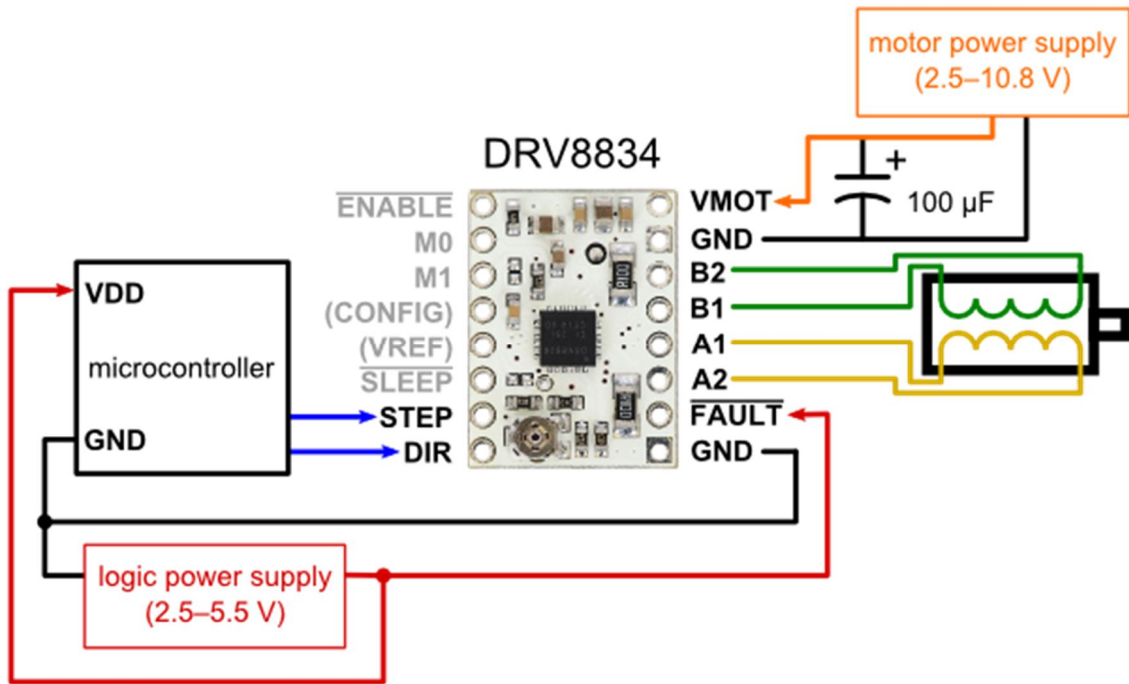
	Назва і призначення	Зображення
1.	<p>Драйвер крокового двигуна ULN2003. Використовуються для підключення уніполярних крокових двигунів. Драйвер може бути зібраний з використанням DIP або SMD компонентів.</p> <p>ULN2003 є інтегральною мікросхемою, що містить сім транзисторів з відкритим колектором та вбудованими резисторами. Він забезпечує ізоляцію між кроковим двигуном та мікроконтролером або іншими цифровими пристроями.</p>	
2.	<p>Модуль MX1508 - компактний драйвер для керування двома малопотужними колекторними двигунами постійного струму або одним кроковим мотором.</p>	
3.	<p>Модуль драйвера керування двома колекторними двигунами на мікросхемі DRV8833 містить усі необхідні компоненти для керування двома колекторними двигунами та не вимагає використання додаткових компонентів. Подвійний Н-міст дозволяє керувати двома двигунами постійного струму або одним біполярним кроковим двигуном, що робить цей драйвер чудовим рішенням для роботи з малопотужними електродвигунами.</p>	

<p>4.</p>	<p>LV8729 драйвер крокового двигуна. Розроблений для 32-бітної електроніки. Обмеження за струмом 1,5 А для крокових двигунів. Характеристики Напруга для двигунів 9-32В Напруга логіки 2-5 В Робочий ток 1,5А Робоча температура до 85 °С Температура вимкнення 150 °С Регулювання мікрошага FULL — 1/128</p>	
<p>5.</p>	<p>L298N Motor Controller Module міні – компактна версія драйвера для двох колекторних двигунів постійного струму або одного крокового двигуна.</p>	
<p>6.</p>	<p>Драйвер крокового двигуна BTS7960. Драйвер колекторного двигуна великої потужності. 43А, 6 ... 27 В.</p>	
<p>7.</p>	<p>DRV8825 – компактний, але при цьому потужний драйвер крокових двигунів з простим керуванням. DRV8825 дозволяє регулювати вихідний струм, має захист від перенавантаження і перегріву, шість варіантів мікрокроку</p>	

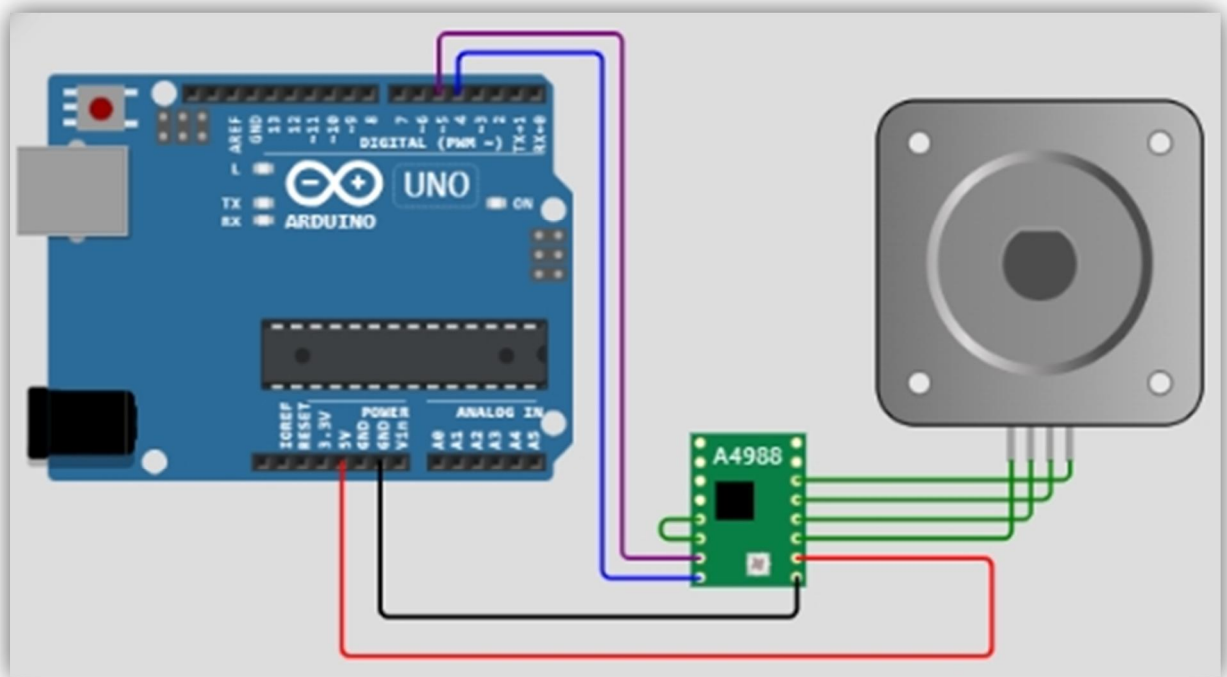
8.	<p>A4988 – компактний та простий в керуванні драйвер крокових двигунів з максимальним струмом 2А. Часто використовується спільно з 3D принтерами.</p>	
	<p>DRV8834 – компактний та простий в керуванні драйвер крокових двигунів</p>	
	<p>MP6500 - драйвер крокового двигуна 35В/2,5А - з аналоговим керуванням струмом</p>	

Схеми підключення крокового двигуна до різних драйверів.





Підключення плати драйвера A4988 та крокового двигуна до плати Arduino у **Wokwi** (<https://wokwi.com/>):



Приклад скетча керування кроковим двигуном:

```
#include <AccelStepper.h>

// визначення пінів підключення драйвера до плати Arduino
#define STEP_PIN 5
#define DIR_PIN 4

AccelStepper stepper(AccelStepper::DRIVER, STEP_PIN, DIR_PIN);
```

```

int i;

void setup() {
  // встановлення швидкості крокового двигуна
  stepper.setMaxSpeed(1000);
  // встановлення напрямлення обертання ротора крокового двигуна
  stepper.setPinsInverted(false, false, true);
}

void loop() {

  for (i = 0; i < 200; i++){
    stepper.setSpeed(-100);
    stepper.runSpeed();
    delay(10);
  }
  delay (1000);
  // змінюємо напрям обертання
  for (i = 0; i < 200; i++){
    stepper.setSpeed(100);
    stepper.runSpeed();
    delay(10);
  }
  delay (1000);

}

```

ПОРЯДОК ВИКОНАННЯ РОБОТИ.

1. Використовуючи сервіс **Wokwi**, підключіть кроковий двигун до плати Arduino використовуючи драйвер A4988 і дві кнопки.
2. Напишіть скетч, який виконував би наступні дії: при натисканні однієї з кнопок вал крокового двигуна обертається в одному напрямку, а при натисканні другої кнопки напрям обертання змінюється на протилежний.
3. Підключіть у схему другий кроковий двигун і передбачте його окреме керування, внесіть відповідні зміни до скетчу.
4. Додайте у схему LCD індикатор, підключений по шині I2C і будьякий термодатчик (який контролює температуру одного з двигунів). Виведіть на екран LCD індикатора інформацію о крокових двигунах і температуру одного з двигунів..

ПИТАННЯ ДЛЯ ПІДГОТОВКИ.

1. Яка будова крокового двигуна?
2. Що таке драйвер крокового двигуна і для чого він використовується?
3. Як підключити кроковий двигун до плати Arduino?
4. Яка бібліотека використовується для керування кроковим двигуном?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основи автоматики та автоматизації : навч. посіб. / Є. П. Пістун, І. Д. Стасюк; Нац. ун-т «Львів. політехніка». – Львів, 2014. – 336 с.
2. Всеукраїнський фестиваль робототехніки Robotica [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <http://robotica.in.ua>
3. Технічна студія «Винахідник»: робототехніка та програмування для школярів (6-16 років) [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <http://vynahidnyk.org>
4. Robot School – дитяча школа робототехніки [Електронний ресурс] : [Вебсайт]. – Режим доступу: <http://www.robotschool.com.ua>
5. Огляд та перспективи використання платформи Arduino Nano 3.0 у вищій школі / Кривонос О. М., Кузьменко Є. В., Кузьменко С. В. // Інформаційні технології і засоби навчання [Електронний ресурс] / Ін-т інформ. технологій і засобів навчання НАПН України, Ун-т менеджменту освіти НАПН України; гол. ред. : В. Ю. Биков. - 2016.

