

Міністерство освіти і науки України
Житомирський державний університет імені Івана Франка

**ІНСТРУКТИВНО-МЕТОДИЧНІ МАТЕРІАЛИ
ДО ЛАБОРАТОРНИХ ЗАНЯТЬ**

з дисципліни

«РОБОТОТЕХНІЧНІ СИСТЕМИ»

для підготовки здобувачів
першого (бакалаврського) рівня вищої освіти

Житомир – 2026

УДК 004.896(076.5)

Рекомендовано до друку вченою радою Житомирського державного університету імені Івана Франка (протокол № 11 від 29 травня 2026 р.)

Рецензенти:

Юрій ТИМОНІН – кандидат технічних наук, доцент, доцент кафедри комп’ютерних технологій і моделювання систем Поліського національного університету.

Оксана НАКОНЕЧНА – кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Одеського державного аграрного університету.

Дмитро ІВАНОВ – доктор технічних наук, доцент, професор кафедри комп’ютерних наук та інформаційних технологій Житомирського державного університету імені Івана Франка.

Киселевич В. В.

Киселевич В. В. Інструктивно-методичні матеріали до лабораторних занять "Робототехнічні системи" для здобувачів першого (бакалаврського) рівня вищої освіти. Житомир : Вид-во ЖДУ ім. І. Франка, 2026. 56 с.

Інструктивно-методичні матеріали містять лабораторні роботи з дисципліни «Робототехнічні системи», які охоплюють основи платформи Arduino, роботу з сенсорами та виконавчими механізмами, 3D-моделювання і друк, а також конструювання автономних роботів. Кожна лабораторна робота включає тему, мету, план, теоретичні відомості, покрокові інструкції, контрольні запитання та завдання для самостійної роботи. Видання призначене для здобувачів вищої освіти.

© Киселевич В. В., 2026

© Житомирський державний університет
імені Івана Франка, 2026

ЗМІСТ

ЗМІСТ	3
ВСТУП.....	4
ЛАБОРАТОРНА РОБОТА № 1.....	6
ЛАБОРАТОРНА РОБОТА № 2.....	10
ЛАБОРАТОРНА РОБОТА № 3.....	13
ЛАБОРАТОРНА РОБОТА № 4.....	16
ЛАБОРАТОРНА РОБОТА № 5.....	19
ЛАБОРАТОРНА РОБОТА № 6.....	22
ЛАБОРАТОРНА РОБОТА № 7.....	25
ЛАБОРАТОРНА РОБОТА № 8.....	28
ЛАБОРАТОРНА РОБОТА № 9.....	31
ЛАБОРАТОРНА РОБОТА № 10.....	34
ЛАБОРАТОРНА РОБОТА № 11.....	37
ЛАБОРАТОРНА РОБОТА № 12.....	40
ЛАБОРАТОРНА РОБОТА № 13.....	44
ЛАБОРАТОРНА РОБОТА № 14.....	47
СПИСОК ВИКОРИСТАНИХ ТА РЕКОМЕНДОВАНИХ ДЖЕРЕЛ	51
ДОВІДКОВО-ІНФОРМАЦІЙНІ ДАНІ	53

ВСТУП

Інструктивно-методичні матеріали до лабораторних занять з дисципліни «Робототехнічні системи» розроблені відповідно до робочої програми навчальної дисципліни для підготовки здобувачів першого (бакалаврського) рівня вищої освіти Житомирського державного університету імені Івана Франка.

Метою вивчення дисципліни є формування у здобувачів вищої освіти необхідних теоретичних знань і практичних навичок з робототехніки та впровадження інженерних рішень у практичну діяльність. Лабораторні заняття є ключовою формою навчальної роботи, що забезпечує набуття практичних умінь та навичок роботи з апаратною платформою Arduino, електронними компонентами, сенсорами, виконавчими механізмами, технологіями 3D-моделювання та друку.

Матеріали побудовано за принципом послідовності та системності: від знайомства з платформою Arduino та простих електричних схем до конструювання повноцінного автономного робота. Такий підхід дозволяє здобувачам поступово опанувувати необхідні компетентності, інтегруючи знання з попередніх лабораторних робіт.

Кожна лабораторна робота містить:

- тему та мету заняття;
- план роботи;
- теоретичні відомості з необхідним мінімумом інформації;
- покрокові інструкції до виконання;
- контрольні запитання для самоперевірки.

Лабораторні роботи розраховані на 30 годин аудиторних занять (денна форма навчання) і охоплюють п'ять тематичних блоків: платформа Arduino, шлях до автоматизації, двигуни, проєктування та 3D-моделювання, автономні Arduino-роботи.

Для виконання лабораторних робіт необхідне таке обладнання та матеріали: плата Arduino UNO, макетна плата, набір з'єднувальних проводів, світлодіоди, резистори, RGB-світлодіод, п'єзовипромінювач, датчики (TCRT5000, HC-SR04, DHT11), OLED-дисплей, DC-мотори, драйвер L298N, серводвигун SG90, потенціометр, паяльна станція, 3D-принтер, комп'ютер із встановленим Arduino IDE.

Рекомендована література наведена наприкінці видання. Довідково-інформаційні дані (розпіновка плати Arduino UNO, таблиця частот нот, кольорове маркування резисторів тощо) подані в додатках для зручності студентів під час виконання лабораторних робіт.

ЛАБОРАТОРНА РОБОТА № 1

Тема: Знайомство з платою Arduino UNO та середовищем Arduino IDE.

Складання найпростішої схеми зі світлодіодом на макетній платі.

Програма «Blink»

Мета

Ознайомитися з апаратною платформою Arduino UNO, основними компонентами плати, макетною платою, середовищем розробки Arduino IDE; навчитися складати найпростішу електричну схему зі світлодіодом та завантажувати програму «Blink» на мікроконтролер.

План

1. Загальна характеристика платформи Arduino UNO.
2. Основні компоненти плати: мікроконтролер ATmega328P, цифрові та аналогові піни, роз'єм живлення, USB-порт.
3. Макетна плата (breadboard): будова, правила з'єднання.
4. Інтерфейс середовища Arduino IDE: панель інструментів, редактор коду, монітор порту.
5. Структура програми (скетчу): функції setup() та loop().
6. Складання схеми зі світлодіодом та резистором на макетній платі.
7. Написання, компіляція та завантаження програми «Blink».

Теоретичні відомості

Платформа Arduino являє собою відкриту апаратно-програмну екосистему для створення інтерактивних електронних пристроїв. Плата Arduino UNO побудована на мікроконтролері ATmega328P і є однією з найпоширеніших моделей для навчання. Вона має 14 цифрових входів/виходів (з яких 6 підтримують ШІМ), 6 аналогових входів, кварцовий резонатор 16 МГц, роз'єм USB, роз'єм живлення та кнопку скидання.

Макетна плата (breadboard) використовується для швидкого прототипування електричних схем без паяння. Вона містить контактні отвори,

з'єднані в ряди: центральна частина має ряди по 5 контактів, а бічні шини живлення з'єднують контакти горизонтально.

Середовище розробки Arduino IDE дозволяє писати програми (скетчі) мовою, що базується на C/C++, компілювати їх та завантажувати на плату. Кожен скетч складається з двох обов'язкових функцій: `setup()` – виконується один раз при старті, та `loop()` – виконується циклічно.

Світлодіод (LED) є напівпровідниковим пристрій, що випромінює світло при протіканні через нього електричного струму. Для обмеження струму послідовно зі світлодіодом підключається резистор, номінал якого розраховується за формулою: $R = (V_{\text{живлення}} - V_{\text{LED}}) / I_{\text{LED}}$. Для типового червоного LED із $V_{\text{LED}} = 2 \text{ В}$ та $I_{\text{LED}} = 20 \text{ мА}$ при живленні 5 В: $R = (5 - 2) / 0,02 = 150 \text{ Ом}$.

Функція `pinMode(pin, mode)` налаштовує режим піна (INPUT або OUTPUT). Функція `digitalWrite(pin, value)` встановлює на піні стан HIGH (5 В) або LOW (0 В). Функція `delay(ms)` зупиняє виконання програми на задану кількість мілісекунд.



Рис. 1.1. Плата Arduino UNO R3

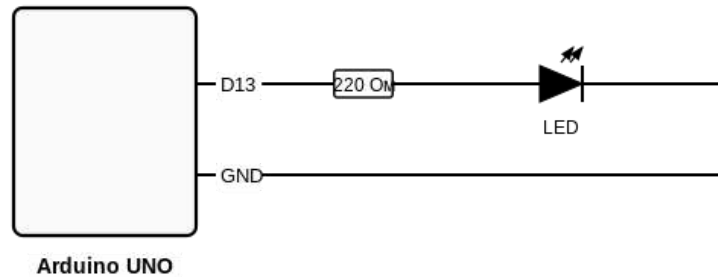


Рис. 1.2. Схема підключення світлодіода до плати Arduino UNO через резистор 220 Ом

Інструкція до виконання

1. Підключіть плату Arduino UNO до комп'ютера за допомогою USB-кабелю.

2. Запустіть середовище Arduino IDE. Оберіть плату (Інструменти → Плата → Arduino UNO) та COM-порт (Інструменти → Порт).

3. Складіть схему на макетній платі:

– з'єднайте цифровий пін 13 Arduino з анодом (+) світлодіода через резистор 220 Ом;

– катод (–) світлодіода з'єднайте із GND на платі Arduino.

4. Введіть програму «Blink»:

```
void setup() {
  pinMode(13, OUTPUT);
}
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

5. Натисніть кнопку «Завантажити» (Upload). Спостерігайте за миганням світлодіода.

6. Змініть затримку на 500 мс, потім на 100 мс. Зафіксуйте результати.

7. Підключіть світлодіод до іншого цифрового піна (наприклад, 8). Модифікуйте програму.

Контрольні запитання

1. Назвіть основні компоненти плати Arduino UNO та їх призначення.
2. Яка структура програми (скетчу) в Arduino IDE?
3. Для чого потрібен резистор при підключенні світлодіода? Як розрахувати його номінал?
4. Поясніть різницю між функціями `setup()` та `loop()`.
5. Що відбудеться, якщо підключити світлодіод без резистора?
6. Як змінити частоту блимання світлодіода?

ЛАБОРАТОРНА РОБОТА № 2

Тема: Широтно-імпульсна модуляція. Плавне керування яскравістю світлодіода. Робота з аналоговими виходами та монітором порту

Мета

Вивчити принцип широтно-імпульсної модуляції (ШІМ); навчитися плавно керувати яскравістю світлодіода за допомогою функції `analogWrite()`; освоїти роботу з монітором послідовного порту для виведення даних.

План

1. Поняття широтно-імпульсної модуляції (ШІМ).
2. Шпаруватість (duty cycle) та її вплив на середню напругу.
3. ШІМ-піни на платі Arduino UNO (позначення ~).
4. Функція `analogWrite(pin, value)`: діапазон значень 0–255.
5. Монітор послідовного порту: `Serial.begin()`, `Serial.print()`, `Serial.println()`.
6. Плавне збільшення та зменшення яскравості світлодіода (ефект «дихання»).

Теоретичні відомості

Широтно-імпульсна модуляція (ШІМ, англ. PWM, Pulse Width Modulation) є способом керування середнього значення напруги шляхом швидкого перемикавання сигналу між увімкненим (HIGH) та вимкненим (LOW) станами. Частка часу, протягом якого сигнал перебуває у стані HIGH, називається шпаруватістю (duty cycle) і вимірюється у відсотках.

На платі Arduino UNO ШІМ підтримують піни 3, 5, 6, 9, 10, 11 (позначені символом ~). Частота ШІМ-сигналу на цих пінах становить приблизно 490 Гц (на пінах 5 і 6 – 980 Гц).

Функція `analogWrite(pin, value)` генерує ШІМ-сигнал на зазначеному піні. Параметр `value` приймає цілі значення від 0 (0 % шпаруватості, сигнал завжди LOW) до 255 (100 % шпаруватості, сигнал завжди HIGH). Наприклад,

analogWrite(9, 127) створить сигнал із шпаруватістю ~50 %, що відповідає середній напрузі ~2,5 В.

Монітор послідовного порту є вбудованим інструментом Arduino IDE для обміну даними між комп'ютером і платою. Для ініціалізації використовується Serial.begin(baudRate), де baudRate – швидкість передачі даних (типово 9600 бод). Функції Serial.print() та Serial.println() виводять дані у монітор порту.

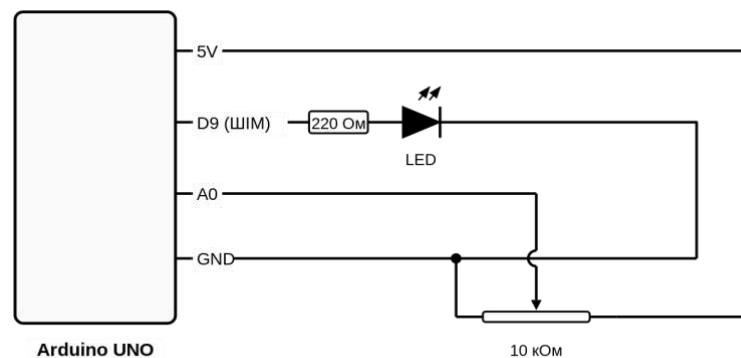


Рис. 2.1. Схема підключення світлодіода до ШІМ-піна Arduino UNO для плавного керування яскравістю

Інструкція до виконання

1. Складіть схему: підключіть світлодіод через резистор 220 Ом до ШІМ-піна 9.

2. Завантажте програму плавного збільшення яскравості:

```
void setup() {  
  pinMode(9, OUTPUT);  
  Serial.begin(9600);  
}  
void loop() {  
  for (int brightness = 0; brightness <= 255; brightness += 5) {  
    analogWrite(9, brightness);  
    Serial.print("Яскравість: ");  
    Serial.println(brightness);  
    delay(30);  
  }  
  for (int brightness = 255; brightness >= 0; brightness -= 5) {
```

```
    analogWrite(9, brightness);  
    Serial.print("Яскравість: ");  
    Serial.println(brightness);  
    delay(30);  
  }  
}
```

3. Завантажте програму на плату. Спостерігайте ефект «дихання» світлодіода.

4. Відкрийте монітор порту (Ctrl+Shift+M). Слідкуйте за значеннями яскравості.

5. Змініть крок зміни яскравості та затримку. Порівняйте результати.

6. Підключіть потенціометр до аналогового входу A0 та керуйте яскравістю за допомогою analogRead().

Контрольні запитання

1. Що таке широтно-імпульсна модуляція та для чого вона використовується?

2. Які піни на Arduino UNO підтримують ШІМ?

3. Який діапазон значень приймає функція analogWrite()?

4. Як шпаруватість впливає на середню напругу сигналу?

5. Як налаштувати та використовувати монітор послідовного порту?

6. Яка різниця між функціями digitalWrite() та analogWrite()?

ЛАБОРАТОРНА РОБОТА № 3

Тема: RGB-світлодіод. Адитивна модель кольору. Програмування плавних переходів між кольорами

Мета

Вивчити будову та принцип дії RGB-світлодіода; засвоїти адитивну модель змішування кольорів RGB; навчитися програмувати плавні переходи між кольорами.

План

1. RGB-світлодіод: типи (із спільним анодом та катодом), будова, принцип дії.
2. Адитивна модель кольору RGB: формування кольорів змішуванням червоного, зеленого та синього.
3. Порівняння колірних моделей RGB та CMYK.
4. Підключення RGB-світлодіода до ШІМ-пінів Arduino UNO.
5. Програмування статичних кольорів та плавних переходів.

Теоретичні відомості

RGB-світлодіод містить три окремих світлодіоди (червоний, зелений, синій) в одному корпусі. Існують два типи: із спільним катодом (загальний вивід – $-$) та із спільним анодом (загальний вивід $+$). У моделі із спільним катодом загальний вивід з'єднується з GND, а кожен кольоровий вивід – через окремий резистор до ШІМ-піна Arduino.

Адитивна колірна модель RGB ґрунтується на змішуванні трьох базових кольорів: червоного (Red), зеленого (Green) та синього (Blue). Кожен компонент може мати значення від 0 до 255. Наприклад: (255, 0, 0) – червоний, (0, 255, 0) – зелений, (255, 255, 0) – жовтий, (255, 255, 255) – білий, (0, 0, 0) – вимкнено.

Модель CMYK (Cyan, Magenta, Yellow, Key/Black) є субтрактивною моделлю, що використовується у друці. На відміну від RGB, де кольори

додаються до чорного фону, у СМΥК фарби віднімають світло від білого аркуша.

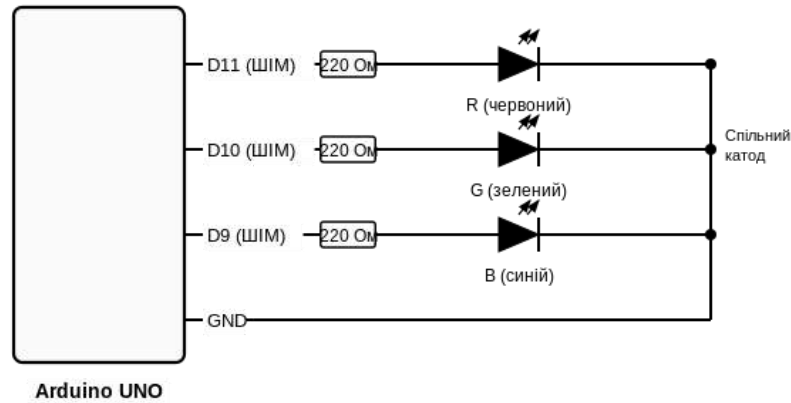


Рис. 3.1. Схема підключення RGB-світлодіода (із спільним катодом) до Arduino UNO

Інструкція до виконання

1. Підключіть RGB-світлодіод (із спільним катодом) до плати Arduino:
 - червоний вивід через резистор 220 Ом до піна 11;
 - зелений вивід через резистор 220 Ом до піна 10;
 - синій вивід через резистор 220 Ом до піна 9;
 - спільний катод – до GND.
2. Завантажте програму для перевірки кожного кольору окремо.
3. Створіть функцію `setColor(red, green, blue)` для зручного керування кольором.

4. Реалізуйте плавний перехід кольорів «веселка» за допомогою циклу:

```
void setColor(int r, int g, int b) {  
    analogWrite(11, r);  
    analogWrite(10, g);  
    analogWrite(9, b);  
}  
  
void setup() {  
    pinMode(11, OUTPUT);  
    pinMode(10, OUTPUT);  
    pinMode(9, OUTPUT);  
}
```

```
void loop() {  
  // Червоний -> Жовтий -> Зелений -> Блакитний -> Синій -> Фіолетовий ->  
  Червоний  
  for (int i = 0; i < 256; i++) { setColor(255, i, 0); delay(10); }  
  for (int i = 255; i >= 0; i--) { setColor(i, 255, 0); delay(10); }  
  for (int i = 0; i < 256; i++) { setColor(0, 255, i); delay(10); }  
  for (int i = 255; i >= 0; i--) { setColor(0, i, 255); delay(10); }  
  for (int i = 0; i < 256; i++) { setColor(i, 0, 255); delay(10); }  
  for (int i = 255; i >= 0; i--) { setColor(255, 0, i); delay(10); }  
}
```

5. Виведіть поточні значення RGB у монітор порту.

Контрольні запитання

1. Яка будова RGB-світлодіода? У чому різниця між типами із спільним анодом та катодом?
2. Поясніть принцип адитивної моделі змішування кольорів.
3. Як отримати жовтий, фіолетовий та білий кольори за допомогою RGB-світлодіода?
4. У чому різниця між колірними моделями RGB та CMYK?
5. Чому для підключення RGB-світлодіода потрібні три окремих резистори?

ЛАБОРАТОРНА РОБОТА № 4

Тема: П'єзовипромінювач. Генерація звукових сигналів. Реалізація простих мелодій

Мета

Вивчити принцип дії п'єзоелектричних елементів; навчитися генерувати звукові сигнали різної частоти за допомогою функції `tone()`; реалізувати програвання простих мелодій.

План

1. П'єзоелектрики: принцип п'єзоефекту, застосування у повсякденному житті.
2. П'єзовипромінювач: будова, принцип генерації звуку.
3. Підключення п'єзовипромінювача до плати Arduino UNO.
4. Функції `tone(pin, frequency, duration)` та `noTone(pin)`.
5. Нотний стан та частоти нот.
6. Програмування простих мелодій за допомогою масивів.

Теоретичні відомості

П'єзоефект полягає у здатності деяких кристалічних матеріалів генерувати електричний заряд при механічній деформації (прямий п'єзоефект) або деформуватися під дією електричного поля (зворотний п'єзоефект). П'єзоматеріали використовуються у запальничках, кварцових годинниках, ультразвукових датчиках та звукових випромінювачах.

П'єзовипромінювач (п'єзобузер, англ. *piezo buzzer*) є електроакустичним пристроєм, що перетворює електричний сигнал у звукові коливання завдяки зворотному п'єзоефекту. При подачі змінної напруги певної частоти п'єзоелемент вібрує та створює звукову хвилю.

Функція `tone(pin, frequency)` генерує прямокутний сигнал заданої частоти (у Гц) на вказаному піні. Необов'язковий третій параметр `duration`

задає тривалість звуку в мілісекундах. Функція `noTone(pin)` зупиняє генерацію сигналу.

Діапазон людського слуху: приблизно 20 Гц – 20 000 Гц. Нота «ля» першої октави має частоту 440 Гц. Частоти нот зростають у геометричній прогресії зі знаменником $^{12}\sqrt{2} \approx 1,0595$ (рівномірно темперований стрій).

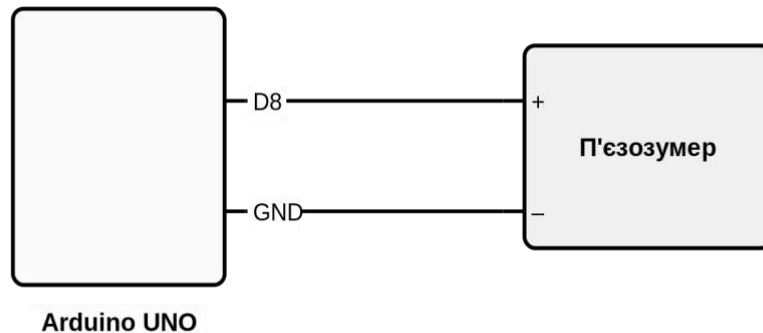


Рис. 4.1. Схема підключення п'єзовипромінювача до Arduino UNO

Інструкція до виконання

1. Підключіть п'єзовипромінювач: один вивід до цифрового піна 8, інший – до GND.

2. Завантажте програму генерації тону:

```
void setup() {  
  tone(8, 440, 1000); // Нота "ля" (440 Гц) тривалістю 1 с  
}  
void loop() {}
```

3. Змініть частоту та тривалість. Прослухайте різні ноти.

4. Реалізуйте програвання мелодії за допомогою масивів:

```
int melody[] = {262, 294, 330, 349, 392, 440, 494, 523};  
int durations[] = {250, 250, 250, 250, 250, 250, 250, 500};  
void setup() {  
  for (int i = 0; i < 8; i++) {  
    tone(8, melody[i], durations[i]);  
    delay(durations[i] * 1.3);  
    noTone(8);  
  }  
}  
void loop() {}
```

5. Створіть власну мелодію, використовуючи таблицю частот нот із довідкових даних.

Контрольні запитання

1. Що таке п'єзоефект? Наведіть приклади застосування.
2. Як працює п'єзовипромінювач?
3. Які параметри приймає функція `tone()`?
4. Який діапазон частот сприймає людське вухо?
5. Як організувати програвання мелодії за допомогою масивів?
6. Чому між нотами потрібна невелика пауза?

ЛАБОРАТОРНА РОБОТА № 5

Тема: Сенсори відстані: інфрачервоний TCRT5000 та ультразвуковий HC-SR04

Мета

Вивчити принцип дії інфрачервоного сенсора TCRT5000 та ультразвукового далекоміра HC-SR04; навчитися зчитувати покази датчиків та обробляти отримані дані.

План

1. Інфрачервоний сенсор TCRT5000: будова, принцип дії, розпізнавання поверхонь.
2. Підключення TCRT5000 до Arduino: цифровий та аналоговий режими.
3. Ультразвуковий далекомір HC-SR04: будова, принцип вимірювання відстані.
4. Підключення HC-SR04: піни Trig та Echo.
5. Формула обчислення відстані за часом проходження ультразвуку.
6. Виведення показів датчиків у монітор порту.

Теоретичні відомості

Інфрачервоний сенсор відбиття TCRT5000 складається з ІЧ-світлодіода та фототранзистора, розміщених поруч. ІЧ-світлодіод випромінює інфрачервоне світло, яке відбивається від поверхні та потрапляє на фототранзистор. Інтенсивність відбитого світла залежить від кольору та типу поверхні: біла поверхня відбиває більше світла, чорна – поглинає. Це дозволяє використовувати TCRT5000 для слідування за лінією.

Ультразвуковий далекомір HC-SR04 вимірює відстань до об'єктів за часом проходження ультразвукового імпульсу. Датчик має два елементи: передавач (Trig) та приймач (Echo). При подачі імпульсу тривалістю 10 мкс на пін Trig модуль генерує пакет із 8 ультразвукових імпульсів частотою 40 кГц.

Пін Echo переходить у стан HIGH на час, рівний часу проходження звуку до об'єкта та назад.

Відстань розраховується за формулою: $d = t \times v / 2$, де t – час у секундах, v – швидкість звуку (~ 343 м/с при 20 °C). У практичних розрахунках: d (см) = t (мкс) / 58. Діапазон вимірювання: 2–400 см.

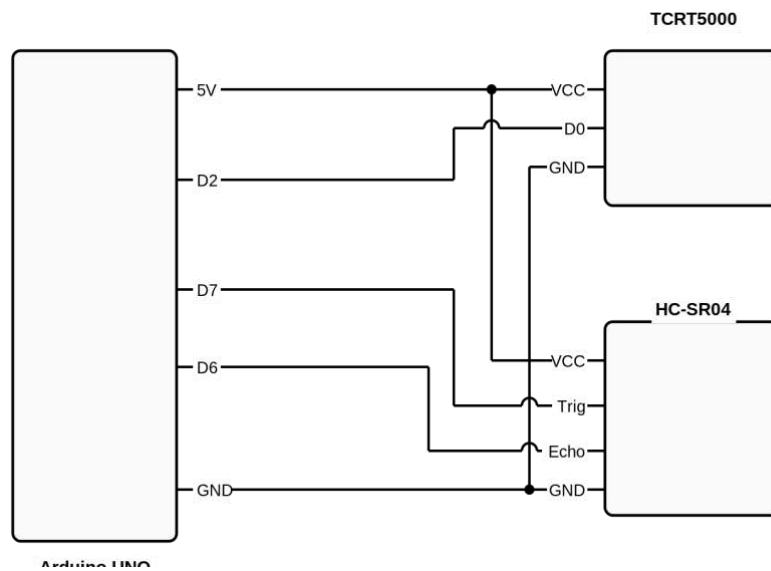


Рис. 5.1. Схема підключення TCRT5000 та HC-SR04 до Arduino UNO

Інструкція до виконання

Частина 1. Робота з TCRT5000:

1. Підключіть модуль TCRT5000 до Arduino: VCC \rightarrow 5V, GND \rightarrow GND, D0 \rightarrow пін 2.

2. Напишіть програму зчитування цифрового сигналу:

```
void setup() {  
  pinMode(2, INPUT);  
  Serial.begin(9600);  
}  
void loop() {  
  int value = digitalRead(2);  
  Serial.println(value == LOW ? "Біла поверхня" : "Чорна поверхня");  
  delay(200);  
}
```

3. Перевірте роботу датчика над білою та чорною поверхнями.

Частина 2. Робота з HC-SR04:

4. Підключіть HC-SR04: VCC → 5V, GND → GND, Trig → пін 7, Echo → пін 6.

5. Напишіть програму вимірювання відстані:

```
const int trigPin = 7;
const int echoPin = 6;
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  float distance = duration / 58.0;
  Serial.print("Відстань: ");
  Serial.print(distance);
  Serial.println(" см");
  delay(500);
}
```

6. Перевірте точність вимірювань за допомогою лінійки.

Контрольні запитання

1. Поясніть принцип дії інфрачервоного сенсора TCRT5000.
2. Як HC-SR04 вимірює відстань до об'єкта?
3. Виведіть формулу обчислення відстані за часом проходження ультразвуку.
4. Який діапазон вимірювання HC-SR04?
5. Які фактори можуть впливати на точність вимірювань ультразвукового далекоміра?
6. Як можна використовувати TCRT5000 для руху робота вздовж лінії?

ЛАБОРАТОРНА РОБОТА № 6

Тема: Цифровий датчик температури і вологості DHT11. Виведення показів на OLED-дисплей. Реалізація проєкту «домашня метеостанція»

Мета

Вивчити характеристики та принцип роботи датчика DHT11; навчитися виводити дані на OLED-дисплей; реалізувати комплексний проєкт «домашня метеостанція».

План

1. Датчик DHT11: характеристики, протокол обміну даними.
2. Підключення DHT11 та встановлення бібліотеки DHT.h.
3. OLED-дисплей (SSD1306): інтерфейс I²C, адресація.
4. Бібліотеки Adafruit_SSD1306 та Adafruit_GFX: основні функції.
5. Виведення тексту та числових показів на OLED-дисплей.
6. Збирання проєкту «домашня метеостанція».

Теоретичні відомості

DHT11 – цифровий датчик температури та вологості з однопровідним інтерфейсом. Основні характеристики: діапазон вимірювання вологості 20–90 % (точність ± 5 %), температури 0–50 °C (точність ± 2 °C), період опитування – не частіше 1 разу на 2 с.

Для роботи з DHT11 в Arduino використовується бібліотека DHT.h. Об'єкт DHT ініціалізується з вказівкою пін та типу датчика: `DHT dht(pin, DHT11)`. Функції `dht.readTemperature()` та `dht.readHumidity()` повертають відповідні значення.

OLED-дисплей на контролері SSD1306 є монохромним графічним дисплеєм з роздільною здатністю 128×64 пікселів та інтерфейсом I²C. Протокол I²C використовує дві лінії: SDA (дані) та SCL (тактовий сигнал). На Arduino UNO: SDA – пін A4, SCL – пін A5.

Бібліотека `Adafruit_SSD1306` керує дисплеєм, а `Adafruit_GFX` надає графічні примітиви. Основні функції: `display.clearDisplay()`, `display.setTextSize(n)`, `display.setCursor(x, y)`, `display.print(text)`, `display.display()`.

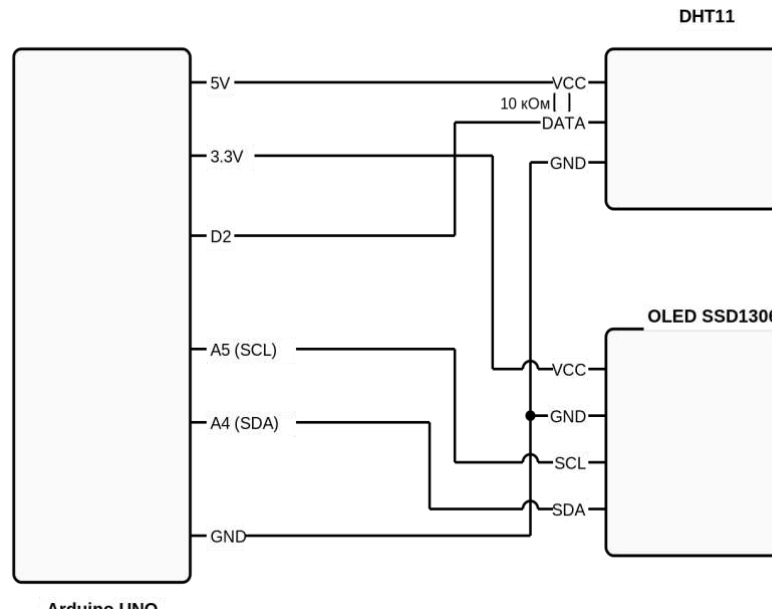


Рис. 6.1. Схема підключення DHT11 та OLED-дисплея до Arduino UNO (проект «домашня метеостанція»)

Інструкція до виконання

1. Підключіть DHT11: VCC → 5V, GND → GND, DATA → пін 2 (через підтягувальний резистор 10 кОм до 5V).
2. Встановіть бібліотеку DHT через Менеджер бібліотек Arduino IDE.
3. Перевірте зчитування показів DHT11 у монітор порту.
4. Підключіть OLED-дисплей: VCC → 3.3V, GND → GND, SDA → A4, SCL → A5.
5. Встановіть бібліотеки Adafruit SSD1306 та Adafruit GFX.
6. Створіть програму «домашня метеостанція»:

```
#include <DHT.h>
#include <Adafruit_SSD1306.h>
#define DHTPIN 2
DHT dht(DHTPIN, DHT11);
```

```

Adafruit_SSD1306 display(128, 64, &Wire, -1);
void setup() {
  dht.begin();
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
}
void loop() {
  float t = dht.readTemperature();
  float h = dht.readHumidity();
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 0);
  display.println("=== Метеостанція ===");
  display.setTextSize(2);
  display.setCursor(0, 20);
  display.print(t, 1); display.println(" C");
  display.setCursor(0, 44);
  display.print(h, 1); display.println(" %");
  display.display();
  delay(2000);
}

```

7. Завантажте програму та спостерігайте оновлення показів на дисплеї.

Контрольні запитання

1. Які характеристики та обмеження датчика DHT11?
2. Який протокол використовує OLED-дисплей SSD1306 для зв'язку з Arduino?
3. Які піни Arduino UNO відповідають за шину I²C?
4. Як вивести текст на OLED-дисплей за допомогою бібліотеки Adafruit?
5. Чому DHT11 не можна опитувати частіше ніж раз на 2 секунди?
6. Як можна розширити метеостанцію додатковими датчиками?

ЛАБОРАТОРНА РОБОТА № 7

Тема: Двигун постійного струму. Підключення через драйвер L298N.

Програмне керування швидкістю та напрямком обертання

Мета

Вивчити будову та принцип роботи двигуна постійного струму; освоїти підключення двигуна через драйвер L298N; навчитися програмно керувати швидкістю та напрямком обертання.

План

1. Двигун постійного струму (DC-мотор): будова, принцип дії.
2. Необхідність використання драйвера двигуна.
3. Драйвер L298N: будова, схема H-мосту, входи керування (IN1, IN2, ENA).
4. Підключення двигуна через L298N до Arduino.
5. Керування напрямком обертання та швидкістю (ШІМ на піні ENA).

Теоретичні відомості

Двигун постійного струму (DC-мотор) перетворює електричну енергію в механічну. Він складається зі статора (нерухомі магніти), ротора (обмотка з провідника) та колектора з щітками, які забезпечують переключення полярності обмоток ротора під час обертання.

Підключати двигун безпосередньо до пінів Arduino не можна, оскільки максимальний струм піна – 40 мА, тоді як типовий DC-мотор споживає 200–500 мА. Для цього використовують драйвери двигунів.

L298N є двоканальним драйвером двигунів на основі подвійного H-мосту. Він дозволяє керувати двома DC-моторами або одним кроковим двигуном. Основні виводи: IN1, IN2 – логічні входи керування напрямком; ENA – вхід увімкнення (ШІМ-керування швидкістю); OUT1, OUT2 – виходи на двигун; VCC – живлення двигунів (до 35 В), GND – спільна земля.

Таблиця керування: IN1=HIGH, IN2=LOW → обертання за годинниковою стрілкою; IN1=LOW, IN2=HIGH → обертання проти годинникової стрілки; IN1=LOW, IN2=LOW → зупинка; IN1=HIGH, IN2=HIGH → гальмування.

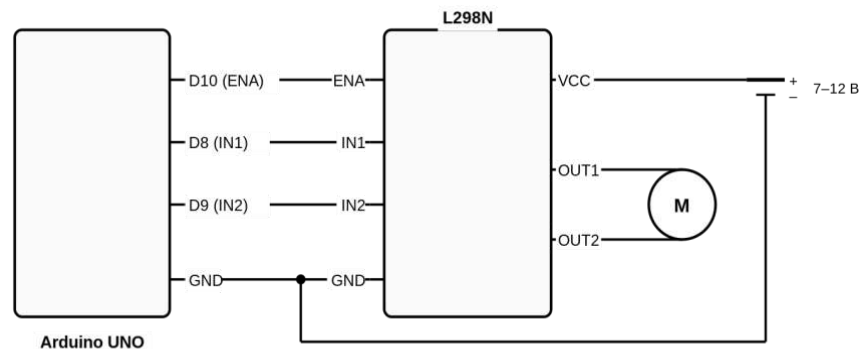


Рис. 7.1. Схема підключення DC-мотора через драйвер L298N до Arduino UNO

Інструкція до виконання

1. Підключіть L298N:

- IN1 → пін 8, IN2 → пін 9, ENA → пін 10 (ШІМ);
- OUT1, OUT2 → клеми двигуна;
- VCC → зовнішнє живлення (7–12 В), GND → GND Arduino + GND живлення.

2. Напишіть програму керування:

```
const int in1 = 8, in2 = 9, ena = 10;
void setup() {
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(ena, OUTPUT);
}
void loop() {
  // Вперед, повна швидкість
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  analogWrite(ena, 255);
  delay(2000);
  // Вперед, половина швидкості
  analogWrite(ena, 128);
```

```
delay(2000);  
// Назад  
digitalWrite(in1, LOW);  
digitalWrite(in2, HIGH);  
analogWrite(ena, 200);  
delay(2000);  
// Зупинка  
digitalWrite(in1, LOW);  
digitalWrite(in2, LOW);  
delay(2000);  
}
```

3. Завантажте програму та спостерігайте за поведінкою двигуна.
4. Додайте керування швидкістю за допомогою потенціометра.

Контрольні запитання

1. Поясніть принцип роботи двигуна постійного струму.
2. Чому не можна підключати двигун напряму до піна Arduino?
3. Що таке H-міст та як він працює?
4. Як за допомогою L298N змінити напрямок обертання двигуна?
5. Як ШІМ-сигнал на піні ENA впливає на швидкість двигуна?
6. Чому необхідне окреме живлення для двигуна?

ЛАБОРАТОРНА РОБОТА № 8

Тема: Серводвигун. Підключення до плати Arduino UNO. Керування кутом повороту за допомогою бібліотеки Servo.h і потенціометра

Мета

Вивчити будову та принцип дії серводвигуна; освоїти підключення серводвигуна та бібліотеку Servo.h; реалізувати керування кутом повороту за допомогою потенціометра.

План

1. Серводвигун: будова, принцип керування (ШІМ-сигнал).
2. Типи серводвигунів: позиційні (180°) та серводвигуни безперервного обертання.
3. Підключення серводвигуна до Arduino UNO.
4. Бібліотека Servo.h: основні функції `attach()`, `write()`, `read()`.
5. Керування кутом повороту за допомогою потенціометра та функції `map()`.

Теоретичні відомості

Серводвигун (сервопривод) поєднує двигун із вбудованим редуктором, датчиком положення (потенціометром) та схемою керування. Він дозволяє точно задавати кут повороту вихідного вала. Типовий серводвигун (наприклад, SG90) має три дроти: живлення (червоний, 5 В), земля (коричневий/чорний, GND), сигнальний (помаранчевий/жовтий).

Керування здійснюється ШІМ-сигналом із періодом 20 мс (50 Гц). Тривалість імпульсу визначає кут: 1 мс → 0°, 1,5 мс → 90°, 2 мс → 180°.

Бібліотека Servo.h спрощує керування серводвигунами. Основні функції: – `servo.attach(pin)` – прив'язує серводвигун до піна; – `servo.write(angle)` – встановлює кут від 0 до 180; – `servo.read()` – зчитує поточний кут.

Потенціометр – змінний резистор із трьома виводами. При підключенні крайніх виводів до 5 В та GND, на середньому виводі формується напруга від

0 до 5 В залежно від положення повзунка. Функція `analogRead()` зчитує значення 0–1023, а `map()` перетворює діапазон: `map(value, 0, 1023, 0, 180)`.

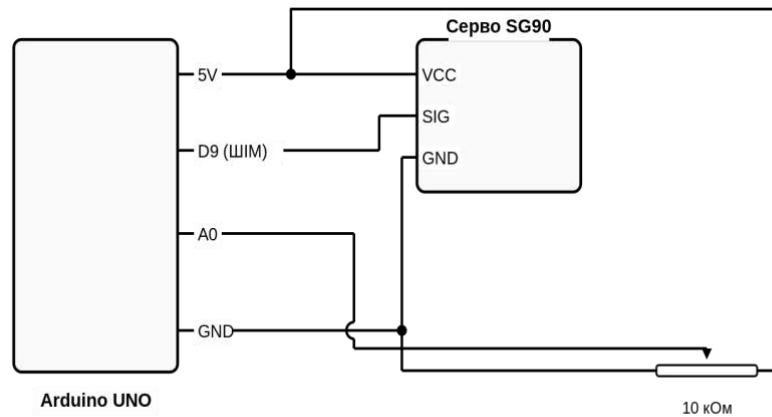


Рис. 8.1. Схема підключення серводвигуна SG90 та потенціометра до Arduino UNO

Інструкція до виконання

1. Підключіть серводвигун SG90: червоний → 5V, коричневий → GND, помаранчевий → пін 9.

2. Завантажте тестову програму:

```
#include <Servo.h>
Servo myServo;
void setup() {
  myServo.attach(9);
}
void loop() {
  for (int angle = 0; angle <= 180; angle += 10) {
    myServo.write(angle);
    delay(500);
  }
  for (int angle = 180; angle >= 0; angle -= 10) {
    myServo.write(angle);
    delay(500);
  }
}
```

3. Підключіть потенціометр: крайні виводи → 5V та GND, середній → A0.

4. Реалізуйте керування кутом від потенціометра:

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

```
Servo myServo;
void setup() {
  myServo.attach(9);
  Serial.begin(9600);
}
void loop() {
  int potValue = analogRead(A0);
  int angle = map(potValue, 0, 1023, 0, 180);
  myServo.write(angle);
  Serial.print("Кут: ");
  Serial.println(angle);
  delay(15);
}
```

5. Обертайте потенціометр та спостерігайте за рухом серводвигуна.

Контрольні запитання

1. Яка будова серводвигуна та чим він відрізняється від DC-мотора?
2. Як ШІМ-сигнал визначає кут повороту серводвигуна?
3. Які основні функції бібліотеки Servo.h?
4. Для чого використовується функція map()? Наведіть приклад.
5. Як потенціометр перетворює механічне положення в електричний сигнал?
6. Які обмеження має серводвигун SG90?

ЛАБОРАТОРНА РОБОТА № 9

Тема: Робота з паяльною станцією: підготовка робочого місця, лудіння провідників, виконання простих паяних з'єднань

Мета

Ознайомитися з будовою паяльної станції, правилами техніки безпеки при паянні; набути практичних навичок лудіння та виконання паяних з'єднань.

План

1. Паяльна станція: будова, основні характеристики (потужність, діапазон температур).
2. Матеріали для паяння: припій (олов'яно-свинцевий, безсвинцевий), флюс, каніфоль.
3. Правила техніки безпеки при роботі з паяльною станцією.
4. Підготовка робочого місця та інструментів.
5. Лудіння провідників та контактних площадок.
6. Виконання паяних з'єднань на макетній друкованій платі.

Теоретичні відомості

Паяльна станція являє собою комплекс обладнання для виконання паяних з'єднань, що складається з блока керування та паяльника з регулюванням температури. Робоча температура для олов'яно-свинцевого припою: 250–350 °С, для безсвинцевого: 300–400 °С.

Припій є легкоплавким сплавом для з'єднання металевих деталей. Найпоширеніші марки: ПОС-61 (61 % олова, 39 % свинцю, температура плавлення 183 °С), безсвинцевий SAC305 (96,5 % Sn, 3 % Ag, 0,5 % Cu, температура плавлення 217 °С).

Флюс використовується для видалення оксидної плівки з поверхні металу та покращення змочуваності припоєм. Каніфоль є природним флюс на основі соснової смоли.

Техніка безпеки: – робоче місце повинно мати витяжну вентиляцію; – не торкатися жала паяльника та нагрітих деталей; – паяльник розміщувати на підставці; – після роботи ретельно вимити руки з милом; – при використанні свинцевого припою не торкатися обличчя руками.

Лудіння полягає у покритті поверхні тонким шаром припою для подальшого паяння. Якісне паяне з'єднання має блискучу, гладку поверхню кінчної форми.

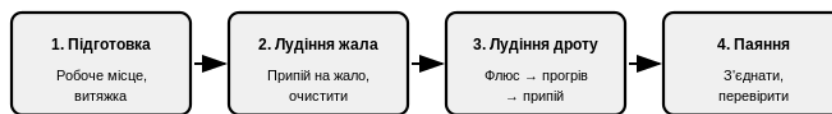


Рис. 9.1. Елементи паяльної станції та приклади якісних паяних з'єднань

Інструкція до виконання

1. Підготуйте робоче місце: увімкніть витяжку, перевірте наявність усіх інструментів (паяльна станція, припій, флюс, кусачки, «третя рука»).
2. Встановіть температуру паяльної станції 300 °С.
3. Виконайте лудіння жала паяльника: нанесіть припій на розігріте жало, очистіть вологою губкою.
4. Виконайте лудіння 5–7 відрізків мідного проводу:
 - зніміть ізоляцію (5–7 мм);
 - нанесіть флюс на оголений провідник;
 - прогрійте провідник жалом паяльника;
 - подайте припій до місця контакту жала та провідника.
5. Виконайте з'єднання двох луджених провідників скруткою з подальшим паянням.
6. На макетній друкованій платі виконайте паяння резисторів та світлодіодів до контактних площадок.

7. Перевірте якість з'єднань: візуально (блискучість, форма) та тестером (провідність).

Контрольні запитання

1. Яка будова паяльної станції та її основні характеристики?
2. Які матеріали використовуються для паяння?
3. Назвіть основні правила техніки безпеки при роботі з паяльною станцією.
4. Що таке лудіння та навіщо воно виконується?
5. За якими ознаками визначається якість паяного з'єднання?
6. У чому переваги безсвинцевого припою порівняно з олов'яно-свинцевим?

ЛАБОРАТОРНА РОБОТА № 10

Тема: Створення 3D-моделі деталі робота у програмному середовищі (Tinkercad / FreeCAD). Підготовка макета для лазерного різачка

Мета

Освоїти базові прийоми 3D-моделювання у середовищі Tinkercad; навчитися створювати параметричну 3D-модель деталі робота; підготувати векторний макет для лазерного різання.

План

1. Огляд програм 3D-моделювання: Tinkercad, FreeCAD, Fusion 360.
2. Інтерфейс Tinkercad: робоча площина, бібліотека примітивів, інструменти трансформації.
3. Базові операції: переміщення, обертання, масштабування, групування, вирізання (Hole).
4. Створення 3D-моделі шасі для робомашини.
5. Експорт моделі у формат STL для 3D-друку.
6. Підготовка 2D-креслення (SVG/DXF) для лазерного різачка.

Теоретичні відомості

3D-моделювання є процесом створення тривимірного математичного представлення об'єкта. Tinkercad є безкоштовним браузерним інструментом від Autodesk для початківців. FreeCAD – параметричний 3D-модельєр із відкритим кодом.

Основний підхід у Tinkercad базується на конструктивній блоковій геометрії (CSG): складні форми створюються шляхом об'єднання та віднімання простих примітивів (куб, циліндр, сфера тощо). Об'єкт, позначений як Hole, при групуванні вирізає свою форму з іншого тіла.

Формат STL (Standard Tessellation Language) описує поверхню 3D-моделі за допомогою трикутників і є стандартним для 3D-друку. Після експорту STL-файл завантажується у слайсер.

Лазерне різання є технологією розрізання та гравіювання листових матеріалів (акрил, фанера, картон) сфокусованим лазерним променем. Для підготовки макета використовуються векторні формати SVG або DXF. Лінії різку зазвичай позначаються червоним кольором (RGB: 255, 0, 0), гравіювання – чорним.

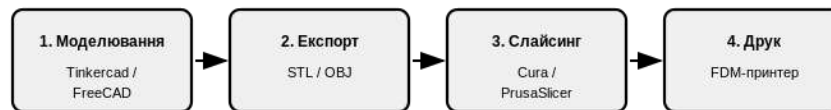


Рис. 10.1. Процес 3D-моделювання шасі робота у Tinkercad

Інструкція до виконання

Частина 1. 3D-моделювання у Tinkercad:

1. Відкрийте tinkercad.com, створіть новий 3D-проект.
2. Створіть основу шасі робота: перетягніть примітив «Вох» на робочу площину, встановіть розміри 120×80×5 мм.
3. Додайте отвори для кріплення двигунів: створіть два циліндри (діаметр 6 мм), позначте їх як Hole, розмістіть та згрупуйте з основою.
4. Додайте кріплення для кулькового колеса та Arduino.
5. Екпортуйте модель у формат STL.

Частина 2. Підготовка макета для лазерного різачка:

6. Створіть новий 2D-проект або використовуйте режим експорту SVG.
7. Підготуйте контур бічної панелі робота у векторному форматі.
8. Розмітьте лінії різку (червоний) та гравіювання (чорний).
9. Збережіть файл у форматі SVG.

Контрольні запитання

1. Які основні підходи використовуються у 3D-моделюванні?
2. Що таке конструктивна блокова геометрія (CSG)?

3. Який формат використовується для експорту моделей на 3D-друк?
4. Як у Tinkercad створити отвір у деталі?
5. Чим відрізняється підготовка макета для лазерного різання від 3D-друку?
6. Які матеріали можна різати лазерним різакон?

ЛАБОРАТОРНА РОБОТА № 11

Тема: Налаштування слайсера. FDM-друк підготовлених деталей з PLA-пластику

Мета

Ознайомитися з технологією FDM-друку та параметрами слайсингу; навчитися підготовляти 3D-модель до друку та виконувати друк деталей з PLA-пластику.

План

1. Технологія FDM (Fused Deposition Modeling): принцип роботи, переваги та обмеження.
2. Будова FDM 3D-принтера: екструдер, хот-енд, стіл, рама, крокові двигуни.
3. Матеріали для FDM-друку: PLA, PETG, ABS – порівняння характеристик.
4. Слайсер: поняття, основні програми (Cura, PrusaSlicer).
5. Параметри слайсингу: висота шару, заповнення, підтримки, адгезія до столу.
6. Підготовка та запуск друку деталей робота.

Теоретичні відомості

FDM (Fused Deposition Modeling) є технологією адитивного виробництва, при якій об'єкт формується пошаровим нанесенням розплавленого термопластику. Нитка (філамент) діаметром 1,75 мм подається в екструдер, де нагрівається до температури плавлення та продавлюється через сопло на робочий стіл.

PLA (полілактид) – біорозкладний полімер, найпопулярніший матеріал для FDM-друку. Температура екструзії: 190–220 °C, температура столу: 50–60 °C. Переваги: простота друку, відсутність запаху, екологічність. Недоліки: низька термостійкість (розм'якшується при ~60 °C), крихкість.

Слайсер – програма, що перетворює 3D-модель (STL) у G-код – послідовність інструкцій для 3D-принтера. Основні параметри: – висота шару (layer height): типово 0,2 мм (стандартна якість) або 0,1 мм (висока якість); – заповнення (infill): 15–20 % для декоративних деталей, 50–100 % для функціональних; – підтримки (supports): генеруються для нависаючих елементів (кут > 45°); – швидкість друку: 40–60 мм/с для PLA.



Рис. 11.1. Модель деталі робота, підготовлена до 3D-друку у слайсері

Інструкція до виконання

1. Запустіть слайсер (Cura або PrusaSlicer).
2. Імпортуйте STL-файл деталі робота, створений у попередній лабораторній роботі.
3. Налаштуйте параметри друку:
 - матеріал: PLA;
 - температура сопла: 210 °С;
 - температура столу: 60 °С;
 - висота шару: 0,2 мм;
 - заповнення: 20 %;
 - швидкість: 50 мм/с.
4. Розмістіть модель на віртуальному робочому столі. За потреби додайте підтримки.
5. Виконайте слайсинг. Перевірте попередній перегляд шарів. Зверніть увагу на орієнтовний час друку та витрату матеріалу.
6. Збережіть G-код на SD-карту або передайте через USB.

7. Підготуйте 3D-принтер: калібрування столу, завантаження філаменту.
8. Запустіть друк. Спостерігайте за першими шарами для контролю адгезії.
9. Після завершення друку: зніміть деталь, видаліть підтримки, за потреби обробіть.

Контрольні запитання

1. Опишіть принцип роботи FDM-технології 3D-друку.
2. Порівняйте матеріали PLA, PETG та ABS за основними характеристиками.
3. Що таке слайсер та яку функцію він виконує?
4. Як висота шару впливає на якість та час друку?
5. Коли необхідно використовувати підтримки при 3D-друку?
6. Які основні етапи підготовки 3D-принтера до друку?

ЛАБОРАТОРНА РОБОТА № 12

Тема: Збірка робомашини LineTracker. Механічна конструкція, підключення електроніки, завантаження базової прошивки

Мета

Освоїти повний цикл конструювання робомашини LineTracker: від механічної збірки до завантаження базової прошивки; набути навичок комплексної інтеграції раніше вивчених компонентів.

План

1. Огляд конструкції робомашини LineTracker: складові, принцип роботи.
2. Механічна збірка: встановлення двигунів, кулькового колеса, шасі.
3. Підключення електроніки: Arduino UNO, драйвер L298N, два датчики TCRT5000.
4. Автономне живлення: літій-іонний акумулятор або батарейний відсік.
5. Складання та завантаження базової прошивки.
6. Тестування руху: вперед, назад, повороти.

Теоретичні відомості

LineTracker є автономним мобільним роботом, що рухається вздовж чорної або білої лінії на контрастному фоні. Типова конструкція включає: – шасі (корпус) – пластикова або фанерна основа; – два DC-мотори з редукторами та колесами – забезпечують диференціальний привід; – кулькове (або ролікове) колесо – опорна точка для балансу; – два ІЧ-датчики TCRT5000 – розпізнають лінію під роботом; – драйвер двигунів L298N – керує моторами; – Arduino UNO – мікроконтролер; – автономне джерело живлення.

Диференціальний привід означає, що робот повертає, змінюючи відносну швидкість лівого та правого коліс. Якщо обидва колеса обертаються з однаковою швидкістю – робот їде прямо. Якщо одне зупиняється – робот повертає у бік зупиненого колеса.

Для автономного живлення використовується батарейний відсік (4×AA – 6 В або 6×AA – 9 В) або літій-іонний акумулятор 7,4 В (2S). Живлення подається на VCC драйвера L298N, а Arduino живиться через вбудований стабілізатор L298N (вивід 5V).

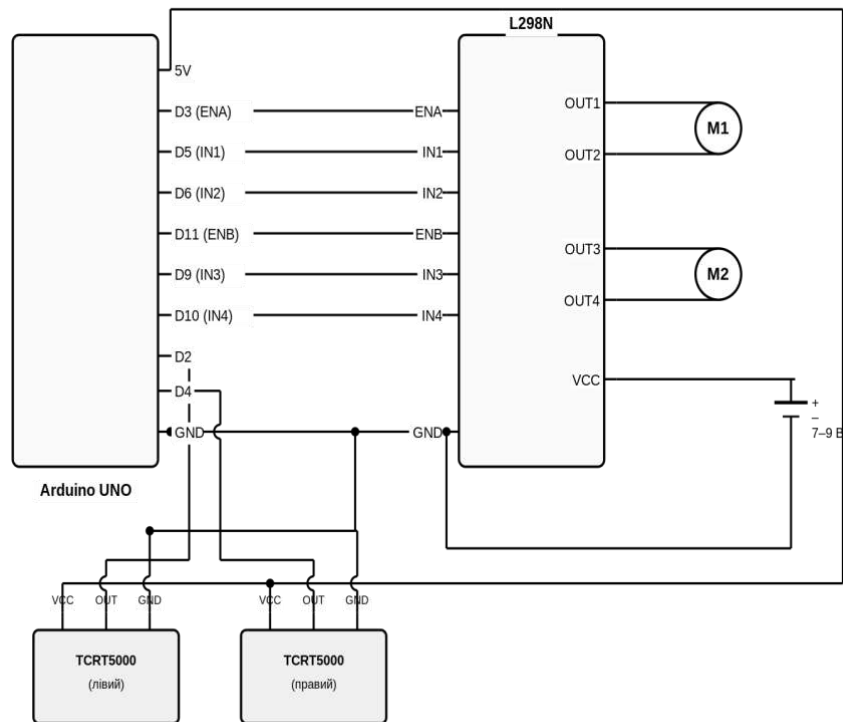


Рис. 12.1. Загальна схема підключення компонентів робота LineTracker

Інструкція до виконання

1. Підготуйте всі компоненти: шасі, 2 DC-мотори з колесами, кулькове колесо, Arduino UNO, L298N, 2× TCRT5000, батарейний відсік, з'єднувальні проводи, кріплення.

2. Виконайте механічну збірку:

- закріпіть двигуни на шасі за допомогою кріплень;
- встановіть колеса на вали двигунів;
- закріпіть кулькове колесо ззаду (або спереду) шасі;

– встановіть датчики TCRT5000 знизу шасі на відстані ~15 мм від поверхні, розташували їх симетрично відносно центру.

3. Виконайте електричне підключення згідно зі схемою:

– лівий мотор → OUT1, OUT2 драйвера L298N;

– правий мотор → OUT3, OUT4;

– IN1 → пін 5, IN2 → пін 6, IN3 → пін 9, IN4 → пін 10;

– ENA → пін 3, ENB → пін 11;

– лівий TCRT5000 D0 → пін 2, правий TCRT5000 D0 → пін 4.

4. Підключіть живлення: батарейний відсік → VCC та GND драйвера.

5. Завантажте базову прошивку для тестування руху:

```
void setup() {
  for (int i = 2; i <= 11; i++) pinMode(i, OUTPUT);
  Serial.begin(9600);
}
void moveForward(int speed) {
  digitalWrite(5, HIGH); digitalWrite(6, LOW); analogWrite(3, speed);
  digitalWrite(9, HIGH); digitalWrite(10, LOW); analogWrite(11, speed);
}
void stop() {
  analogWrite(3, 0); analogWrite(11, 0);
}
void loop() {
  moveForward(180);

  delay(2000);
  stop();

  delay(2000);
}
```

6. Перевірте напрямок обертання кожного мотора. За потреби поміняйте місцями дроти.

7. Тестуйте рух вперед, назад, поворот ліворуч та праворуч.

Контрольні запитання

1. Які основні компоненти робомашини LineTracker та їх функції?
2. Поясніть принцип диференціального приводу.
3. Як розмістити датчики TCRT5000 для ефективного розпізнавання лінії?

4. Які вимоги до автономного джерела живлення робота?
5. Як перевірити правильність підключення двигунів?
6. Чому важливо тестувати кожен компонент окремо перед загальним тестуванням?

ЛАБОРАТОРНА РОБОТА № 13

Тема: Програмування логіки руху LineTracker уздовж лінії змінної ширини та з поворотами

Мета

Розробити та оптимізувати алгоритм руху роботомашини LineTracker уздовж чорної лінії з поворотами різного радіусу; навчитися налаштовувати параметри для стабільного руху.

План

1. Алгоритм слідування за лінією: логіка прийняття рішень на основі показів двох датчиків.
2. Таблиця станів датчиків та відповідних дій робота.
3. Реалізація базового алгоритму.
4. Налаштування швидкості для стабільного проходження поворотів.
5. Обробка перехресть та розривів лінії.
6. Тестування та оптимізація на трасі.

Теоретичні відомості

Алгоритм слідування за лінією базується на аналізі показів двох ІЧ-датчиків, розміщених по обидва боки від лінії. Кожен датчик повертає два стани: 0 (біла поверхня / лінія виявлена) або 1 (чорна поверхня / лінія не виявлена). Це залежить від типу підключення та налаштувань модуля.

Таблиця станів (для білої лінії на чорному фоні): Лівий = 0, Правий = 0 → обидва на лінії → їхати прямо; Лівий = 0, Правий = 1 → лінія зміщена ліворуч → повернути ліворуч; Лівий = 1, Правий = 0 → лінія зміщена праворуч → повернути праворуч; Лівий = 1, Правий = 1 → лінія втрачена → зупинитися або шукати.

Для плавного руху використовується пропорційне керування: замість різкого повороту зменшується швидкість одного колеса. Наприклад, при

зміщенні ліворуч: ліве колесо – мінімальна швидкість, праве – повна швидкість.

Для поворотів змінної крутизни базова швидкість та різниця швидкостей коліс налаштовуються експериментально. Типові значення: базова швидкість 120–180 (з 255), швидкість повороту – 0–80.

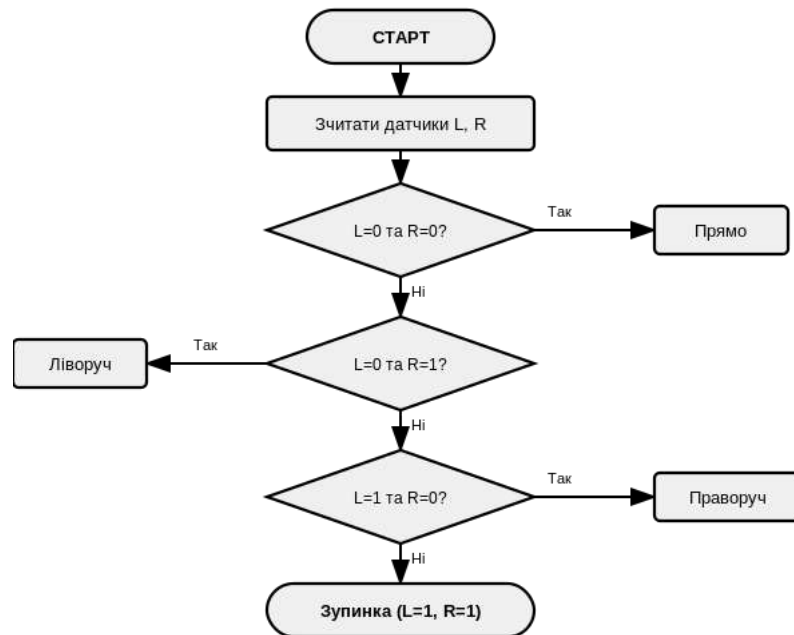


Рис. 13.1. Алгоритм прийняття рішень роботомашини LineTracker (блок-схема)

Інструкція до виконання

1. Переконайтеся, що роботомашини зібрана та базова прошивка працює (лабораторна робота № 12).

2. Завантажте алгоритм слідування за лінією:

```
const int leftSensor = 2, rightSensor = 4;
const int in1=5, in2=6, in3=9, in4=10, enA=3, enB=11;
int baseSpeed = 150;
int turnSpeed = 60;
void setup() {
```

```

    for (int i = 2; i <= 11; i++) pinMode(i, OUTPUT);
    pinMode(leftSensor, INPUT);
    pinMode(rightSensor, INPUT);
    Serial.begin(9600);
}
void setMotors(int leftSpeed, int rightSpeed) {
    digitalWrite(in1, leftSpeed >= 0 ? HIGH : LOW);
    digitalWrite(in2, leftSpeed >= 0 ? LOW : HIGH);
    analogWrite(enA, abs(leftSpeed));
    digitalWrite(in3, rightSpeed >= 0 ? HIGH : LOW);
    digitalWrite(in4, rightSpeed >= 0 ? LOW : HIGH);
    analogWrite(enB, abs(rightSpeed));
}
void loop() {
    int L = digitalRead(leftSensor);
    int R = digitalRead(rightSensor);
    if (L == 0 && R == 0) setMotors(baseSpeed, baseSpeed); // Прямо
    else if (L == 0 && R == 1) setMotors(turnSpeed, baseSpeed); // Ліворуч
    else if (L == 1 && R == 0) setMotors(baseSpeed, turnSpeed); // Праворуч
    else setMotors(0, 0); // Зупинка
}

```

3. Підготуйте трасу: наклейте чорну ізоляційну стрічку на білу поверхню (або навпаки).
4. Встановіть робота на трасу. Запустіть та спостерігайте за поведінкою.
5. Налаштуйте параметри `baseSpeed` та `turnSpeed` для стабільного руху.
6. Ускладніть трасу: додайте круті повороти, змінійте ширину лінії.
7. Виведіть покази датчиків у монітор порту для діагностики.

Контрольні запитання

1. Опишіть алгоритм слідування робота за лінією на основі двох датчиків.
2. Як таблиця станів визначає поведінку робота?
3. Чому використовується пропорційне керування замість дискретного?
4. Як налаштувати параметри швидкості для стабільного проходження поворотів?
5. Що станеться, якщо обидва датчики одночасно втратять лінію?
6. Як можна покращити алгоритм для проходження перехресть?

ЛАБОРАТОРНА РОБОТА № 14

Тема: Модифікація конструкції LineTracker: додавання ультразвукового сонара для оминання перешкод. Підсумкова презентація готового робота

Мета

Модифікувати конструкцію LineTracker, додавши ультразвуковий далекомір для виявлення та оминання перешкод; інтегрувати алгоритми слідування за лінією та оминання; підготувати та презентувати готового робота.

План

1. Встановлення ультразвукового сонара HC-SR04 на конструкцію робота.
2. Інтеграція коду вимірювання відстані в основну прошивку.
3. Алгоритм оминання перешкоди: виявлення, зупинка, об'їзд, повернення на лінію.
4. Тестування та налаштування комбінованого алгоритму.
5. Підготовка та проведення підсумкової презентації робота.

Теоретичні відомості

Модифікація LineTracker полягає у додаванні ультразвукового далекоміра HC-SR04, який встановлюється на передній частині робота та спрямований вперед по ходу руху. Це дозволяє виявляти перешкоди на шляху руху до зіткнення.

Алгоритм комбінованої поведінки: 1. Якщо відстань до перешкоди $>$ порогового значення (наприклад, 15 см) – працює звичайний алгоритм слідування за лінією. 2. Якщо відстань \leq порогового значення – активується режим оминання перешкоди: а) зупинка; б) поворот у визначеному напрямку (наприклад, ліворуч) на $\sim 90^\circ$; в) рух вперед на фіксовану відстань; г) поворот назад (праворуч) на $\sim 90^\circ$; д) рух вперед до повернення на лінію; е) поворот для відновлення руху вздовж лінії.

Для більш інтелектуального оминання можна використовувати серводвигун з установленим на ньому сонаром для сканування простору зліва та справа перед вибором напрямку оминання.

Презентація робота включає: демонстрацію конструкції, опис використаних компонентів, демонстрацію проходження траси, відповіді на запитання.

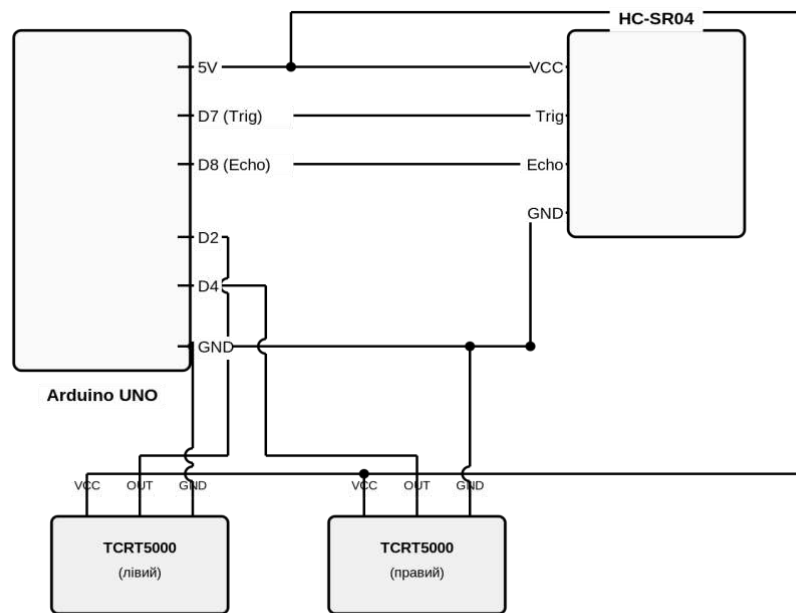


Рис. 14.1. Модифікована схема підключення LineTracker з ультразвуковим сонаром HC-SR04

Інструкція до виконання

1. Закріпіть HC-SR04 на передній частині робота, спрямувавши його вперед.
2. Підключіть: Trig → пін 7, Echo → пін 8. (За потреби перепідключіть інші компоненти.)
3. Додайте функцію вимірювання відстані до основної прошивки:

```
float getDistance() {
    digitalWrite(7, LOW); delayMicroseconds(2);
    digitalWrite(7, HIGH); delayMicroseconds(10);
    digitalWrite(7, LOW);
    long duration = pulseIn(8, HIGH, 30000);
```

```

    if (duration == 0) return 999;
    return duration / 58.0;
}
4. Модифікуйте головний цикл для оминання перешкод:

void loop() {
    float dist = getDistance();
    if (dist < 15) {
        setMotors(0, 0); delay(300);

        // Об'їзд ліворуч
        setMotors(-120, 120); delay(600); // Поворот ліворуч

        setMotors(150, 150); delay(800); // Вперед

        setMotors(120, -120); delay(600); // Поворот праворуч

        setMotors(150, 150); delay(800); // Вперед

        setMotors(120, -120); delay(400); // Довернути праворуч

        // Повернутися до слідування за лінією
        while (digitalRead(leftSensor) == 1 && digitalRead(rightSensor) == 1) {
            setMotors(100, 100);
        }
    } else {
        // Стандартне слідування за лінією
        int L = digitalRead(leftSensor);
        int R = digitalRead(rightSensor);
        if (L == 0 && R == 0) setMotors(baseSpeed, baseSpeed);
        else if (L == 0 && R == 1) setMotors(turnSpeed, baseSpeed);
        else if (L == 1 && R == 0) setMotors(baseSpeed, turnSpeed);
        else setMotors(0, 0);
    }
}

```

5. Тестуйте на трасі з перешкодами (коробки, стаканчики).

6. Налаштуйте порогову відстань та часові параметри оминання.

7. Підготуйте презентацію: опис конструкції, схема підключення, демонстрація роботи.

Контрольні запитання

1. Як інтегрувати ультразвуковий далекомір у конструкцію робота?
2. Опишіть алгоритм оминання перешкоди.
3. Як визначити оптимальну порогову відстань для виявлення перешкоди?
4. Які проблеми можуть виникнути при поверненні робота на лінію після оминання?

5. Як можна вдосконалити алгоритм оминання за допомогою серводвигуна?

6. Підсумуйте, які компоненти та знання були використані при створенні робота.

СПИСОК ВИКОРИСТАНИХ ТА РЕКОМЕНДОВАНИХ

ДЖЕРЕЛ

1. Ходукін М. А. Архітектура комп'ютера та вбудовані мікропроцесорні системи з використанням Arduino : навч.-метод. посіб. Ч. 2. Кривий Ріг : ДУЕТ, 2023. 156 с.
2. Манжілевський О. Д., Іскович-Лотоцький Р. Д. Сучасні адитивні технології 3D друку. Особливості практичного застосування : навч. посіб. Вінниця : ВНТУ, 2021. 105 с.
3. Годунко М. О., Гречка А. І. Основи 3D моделювання і робототехніки : навч. посіб. для проведення практ. занять. Київ : 7БЦ, 2024. 144 с.
4. Vohall J. Arduino Workshop : a Hands-On Introduction with 65 Projects. 2nd ed. San Francisco : No Starch Press, 2021. 432 p.
5. Monk S. Programming Arduino : Getting Started with Sketches. 3rd ed. New York : McGraw-Hill Education, 2022. 178 p.
6. Цвіркун Л. І., Грулер Г. Робототехніка та мехатроніка : навч. посіб. Дніпро : Видавництво НГУ, 2017. 224 с.
7. Craig J. J. Introduction to Robotics : Mechanics and Control. 4th ed. Boston : Pearson, 2017. 480 p.
8. Цирульник С. М., Азаров О. Д., Крупельницький Л. В., Троянівська Т. І. Програмування мікроконтролерів AVR : навч. посіб. Вінниця : ВНТУ, 2018. 111 с.
9. Banzi M., Shiloh M. Getting Started with Arduino : the Open Source Electronics Prototyping Platform. 4th ed. Sebastopol : Make Community, 2022. 285 p.
10. Margolis M., Jepson B., Weldin N. R. Arduino Cookbook : Recipes to Begin, Expand, and Enhance Your Projects. 3rd ed. Sebastopol : O'Reilly Media, 2020. 800 p.

11. Horvath J., Cameron R. Mastering 3D Printing : a Guide to Modeling, Printing, and Prototyping. 2nd ed. Berkeley : Apress, 2020. 368 p.
12. Морзе Н. В., Варченко-Троценко Л. О., Гладун М. А. Основи робототехніки : навч. посіб. Кам'янець-Подільський : ПП Буйницький О. А., 2016. 184 с.
13. Corke P. Robotics, Vision and Control : Fundamental Algorithms in Python. 3rd ed. Cham : Springer Nature, 2023. 850 p.
14. Spong M. W., Hutchinson S., Vidyasagar M. Robot Modeling and Control. 2nd ed. Hoboken : John Wiley & Sons, 2020. 608 p.
15. Arduino : офіційний сайт. URL: <https://www.arduino.cc/> (дата звернення: 20.04.2026).
16. Arduino Project Hub : каталог проєктів. URL: <https://projecthub.arduino.cc/> (дата звернення: 20.04.2026).
17. Arduino Documentation : офіційна документація платформи. URL: <https://docs.arduino.cc/> (дата звернення: 20.04.2026).
18. Tinkercad Circuits : онлайн-симулятор Arduino. URL: <https://www.tinkercad.com/circuits> (дата звернення: 20.04.2026).
19. Adafruit Learning System : технічні гайди до сенсорів і модулів. URL: <https://learn.adafruit.com/> (дата звернення: 20.04.2026).

ДОВІДКОВО-ІНФОРМАЦІЙНІ ДАНІ

Д.1. Розпіновка плати Arduino UNO

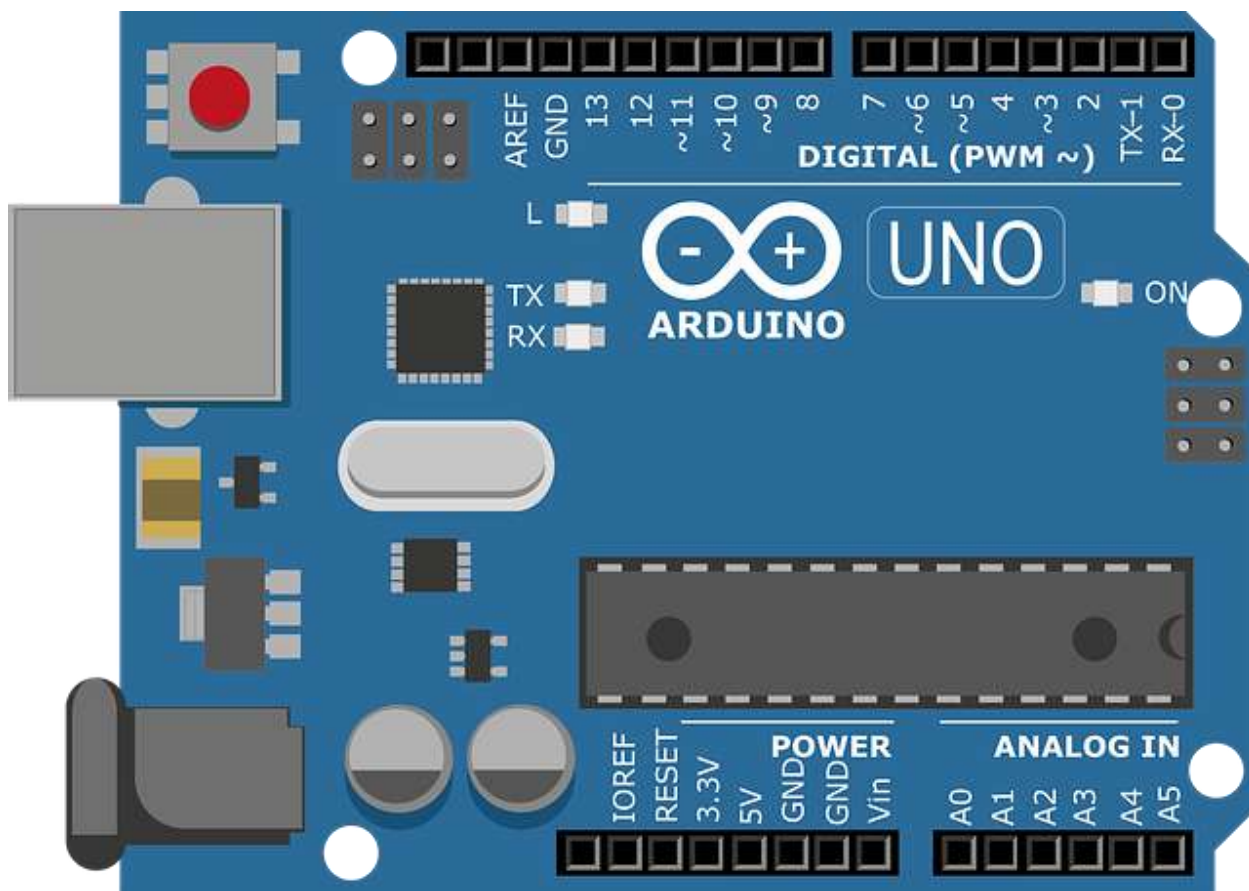


Рис. Д.1. Зовнішній вигляд та розпіновка плати Arduino UNO R3

Основні характеристики плати Arduino UNO:

- мікроконтролер: ATmega328P;
- робоча напруга: 5 В;
- вхідна напруга (рекомендована): 7–12 В;
- цифрових входів/виходів: 14 (з них 6 підтримують ШІМ);
- аналогових входів: 6;
- максимальний струм одного піна: 40 мА;
- Flash-пам'ять: 32 КБ;
- SRAM: 2 КБ;
- тактова частота: 16 МГц.

Д.2. Таблиця частот нот (перша октава)

До (C)	Ре (D)	Ми (E)	Фа (F)	Соль (G)	Ля (A)	Сі (В)	До2 (C2)
262	294	330	349	392	440	494	523

* Частоти вказані в герцах (Гц).

Д.3. Кольорове маркування резисторів



Колір	Назва	Цифра	Множник	Допуск
Чорний	Чорний	0	$\times 1$	—
Коричневий	Коричневий	1	$\times 10$	$\pm 1\%$
Червоний	Червоний	2	$\times 100$	$\pm 2\%$
Оранжевий	Оранжевий	3	$\times 1\,000$	—
Жовтий	Жовтий	4	$\times 10\,000$	—
Зелений	Зелений	5	$\times 100\,000$	$\pm 0,5\%$
Синій	Синій	6	$\times 1\,000\,000$	$\pm 0,25\%$
Фіолетовий	Фіолетовий	7	$\times 10\,000\,000$	$\pm 0,1\%$
Сірий	Сірий	8	—	$\pm 0,05\%$
Білий	Білий	9	—	—
Золотий	Золотий	—	$\times 0,1$	$\pm 5\%$
Срібний	Срібний	—	$\times 0,01$	$\pm 10\%$

Опір (Ом) = (1-ша цифра $\times 10$ + 2-га цифра) \times множник

Рис. Д.3. Таблиця кольорового маркування резисторів

Д.4. Основні функції Arduino IDE

- pinMode(pin, mode) – налаштування режиму піна (INPUT / OUTPUT / INPUT_PULLUP);
- digitalWrite(pin, value) – встановлення стану піна (HIGH / LOW);
- digitalRead(pin) – зчитування цифрового стану піна;
- analogRead(pin) – зчитування аналогового значення (0–1023);
- analogWrite(pin, value) – генерація ШІМ-сигналу (0–255);
- delay(ms) – затримка виконання програми;
- millis() – час від початку роботи програми в мілісекундах;

- `tone(pin, frequency, duration)` – генерація звукового сигналу;
- `pulseIn(pin, value)` – вимірювання тривалості імпульсу;
- `map(value, fromLow, fromHigh, toLow, toHigh)` – лінійне перетворення діапазону;
- `Serial.begin(baudRate)` – ініціалізація послідовного порту;
- `Serial.print()` / `Serial.println()` – виведення даних у монітор порту.

Д.5. Характеристики основних компонентів

Датчик DHT11: температура 0–50 °C (± 2 °C), вологість 20–90 % (± 5 %), мін. інтервал 2 с.

Сонар HC-SR04: діапазон 2–400 см, кут 15°, робоча напруга 5 В.

Сенсор TCRT5000: робоча відстань 1–25 мм, напруга 3,3–5 В.

Серводвигун SG90: кут 0–180°, момент 1,8 кг·см, швидкість 0,1 с/60°, живлення 4,8–6 В.

Драйвер L298N: 2 канали, макс. струм 2 А/канал, напруга двигунів 5–35 В.

OLED SSD1306: 128×64 пікселів, інтерфейс I²C, адреса 0x3C, напруга 3,3–5 В.

Навчально-методичне видання

КИСЕЛЕВИЧ ВОЛОДИМИР ВОЛОДИМИРОВИЧ

РОБОТОТЕХНІЧНІ СИСТЕМИ

Інструктивно-методичні матеріали до лабораторних занять

Надруковано з оригінал-макета автора
Підписано до друку __.__.26. Формат 60×90/16. Ум. друк. арк. 3,6.
Гарнітура Times New Roman. Зам. ____. Наклад 100.

Видавництво Житомирського державного університету імені Івана Франка
вул. Велика Бердичівська, 40, м. Житомир, 10008