

Практична робота 6. Знайомство з платформою Arduino

Теоретичні відомості

Arduino – торгова марка апаратно-програмних засобів для побудови простих систем автоматики і робототехніки, орієнтованих на непрофесійних користувачів.

Програмна частина являє собою безкоштовну програмну оболонку для написання програм, їх компіляції та програмування апаратних засобів.

Апаратна частина включає набір змонтованих друкованих плат, що продаються як офіційним виробником, так і сторонніми виробниками. Повністю відкрита архітектура системи дозволяє вільно копіювати або доповнювати лінійку продукції Arduino.

Під торговою маркою Arduino випускається кілька плат з мікроконтролером (Boards) та плати розширення (Shields).

Більшість плат з мікроконтролером забезпечені мінімально необхідним набором компонентів для нормальної роботи мікроконтролера (стабілізатор живлення, кварцовий резонатор, ланцюжки скидання і т.п.).

Мікроконтролери для Arduino відрізняються наявністю попередньо прошитого в них завантажувача. За допомогою цього завантажувача користувач завантажує свою програму в мікроконтролер без використання традиційних окремих апаратних програматорів.

В серії пристроїв Arduino в основному застосовуються мікроконтролери Atmel AVR. Існують також плати на базі ARM Cortex M та Intel x86.

Arduino UNO

Arduino Uno – це широко використовувана плата мікроконтролерів з відкритим кодом на базі мікроконтролера ATmega328P (рис. 1.1). У його склад входить все необхідне для зручної роботи з мікроконтролером. Для початку роботи з пристроєм досить просто подати живлення від AC/DC-адаптера чи батарейки, або підключити його до комп'ютера за допомогою USB-кабелю.

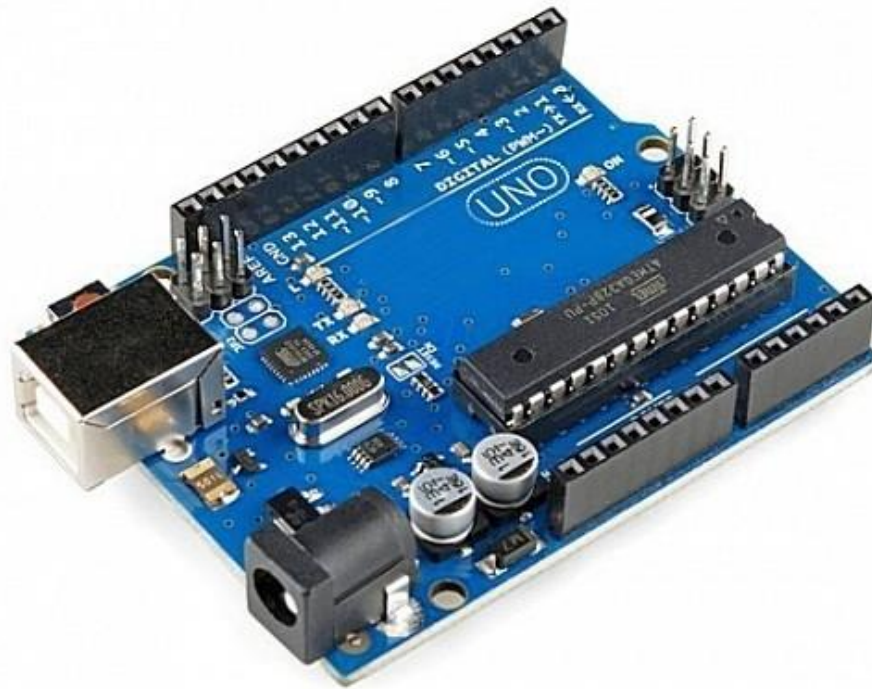


Рисунок 1.1 – Плата Arduino Uno R3 на базі мікроконтролера ATmega328P

Кожен з 14 цифрових пінів Arduino Uno (рис. 1.2) може працювати в якості входу або виходу. Рівень напруги на виходах обмежений 5В. Максимальний струм, який може віддавати або споживати один пін, становить 40 мА. Крім цього, деякі піни Arduino можуть виконувати додаткові функції:

- *Послідовний інтерфейс*: піни 0 (RX) та 1 (TX). Використовуються для отримання (RX) і передачі (TX) даних по послідовному інтерфейсу. Ці піни з'єднані з відповідними виходами мікросхеми ATmega8U2, яка виконує роль перетворювача USB-UART.
- *Зовнішні переривання*: піни 2 і 3. Чи можуть служити джерелами переривань, що виникають при високому, низькому або зміні сигналу на них.
- *ШИМ (широкоімпульсна модуляція)*: піни 3, 5, 6, 9, 10 і 11. За допомогою функції `analogWrite()` можуть виводити 8-бітні аналогові значення в вигляді ШИМ-сигналу.

Широтно-імпульсна модуляція – процес керування середнім значенням вихідної напруги шляхом зміни тривалості замкнутого стану електронного ключа (пульсуючого увімкнення і вимкнення споживача енергії).

- *Світлодіод*: 13. Вбудований світлодіод, приєднаний до 13 піна. При відправці значення HIGH світлодіод вмикається, при відправці LOW – вимикається.

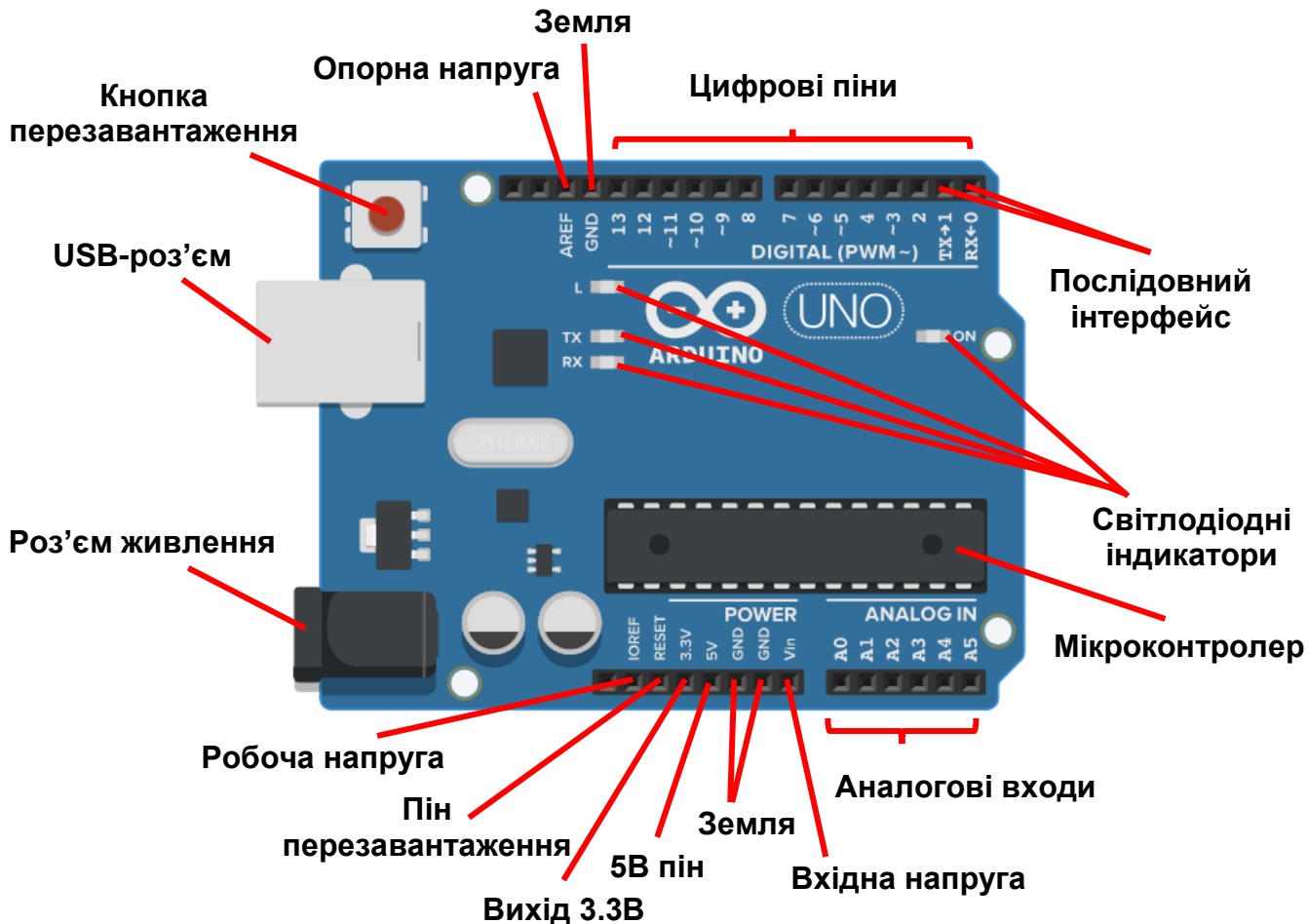


Рисунок 1.2 – Схема розміщення виходів Arduino Uno R3

В Arduino Uno також є 6 аналогових входів (A0 – A5), кожен з яких може представити аналогову напругу у вигляді 10-бітного числа (1024 значення). За замовчуванням, вимір напруги здійснюється в діапазоні від 0 до 5 В. Проте, верхню межу цього діапазону можна змінити, використовуючи вихід AREF. Крім цього, деякі з аналогових входів мають додаткові функції:

- *I²C*: вихід A4 або SDA і вихід A5 або SCL. З використанням бібліотеки `Wire` дані виходи можуть здійснювати зв'язок по інтерфейсу I²C.

Крім перерахованих на платі існує ще кілька виходів:

- *AREF*. Опорна напруга для аналогових входів.
- *Reset*. Формування низького рівня (LOW) на цьому виході призведе до перезавантаження мікроконтролера. Зазвичай цей вихід служить для функціонування кнопки скидання на платах розширення.

Arduino Uno може житися від USB або від зовнішнього джерела живлення – тип джерела вибирається автоматично.

В якості зовнішнього джерела живлення (не USB) може використовуватися мережевий AC/DC-адаптер або акумулятор/батарейка. Штекер адаптера (діаметр – 2.1 мм, центральний контакт – позитивний) необхідно вставити у відповідний роз'єм живлення на платі. В разі живлення від акумулятора/батарейки, її необхідно під'єднати до виходів GND та VIN.

Напруга зовнішнього джерела живлення може бути в межах від 6 до 20 В. Однак, зменшення напруги живлення нижче 7В призводить до зменшення напруги на виході 5V, що може стати причиною нестабільної роботи пристрою. Використання напруги більше 12В може призводити до перегріву стабілізатора напруги і виходу плати з ладу. З огляду на це, рекомендується використовувати джерело живлення з напругою в діапазоні від 7 до 12В.

Нижче перераховані виходи живлення, розташовані на платі:

- *VIN*. Напруга, що надходить в Arduino безпосередньо від зовнішнього джерела живлення. Через цей вихід можна як подавати зовнішнє живлення, так і споживати струм, коли пристрій живиться від зовнішнього адаптера.
- *5V*. На вихід подається напруга 5В від стабілізатора напруги на платі, незалежно від того, як живиться пристрій: від адаптера (7 – 12В), від USB (5В) або через вихід VIN (7 – 12В). Живити пристрій через вихід 5V або 3V3 не рекомендується, оскільки в цьому випадку не використовується стабілізатор напруги, що може привести до виходу плати з ладу.
- *3V3*. 3.3В, що надходять від стабілізатора напруги на платі. Максимальний струм становить 50 мА.

- *GND*. Вихід землі.
- *IOREF*. Цей вихід надає платам розширення інформацію про робочу напругу мікроконтролера Arduino. Залежно від напруги, отриманої з виходу IOREF, плата розширення може перейти на відповідне джерело живлення або задіяти перетворювачі рівнів, що дозволить їй працювати як з 5В, так і з 3.3В-пристроями.

Об'єм флеш-пам'яті ATmega328 становить 32 КБ (з яких 0.5 КБ використовуються завантажувачем). Мікроконтролер також має 2 КБ пам'яті SRAM і 1 КБ EEPROM.

На відміну від всіх попередніх плат Arduino, Uno в якості перетворювача інтерфейсів USB-UART використовує мікроконтролер ATmega16U2 (ATmega8U2 до версії R2) замість мікросхеми FTDI.

Концепція програмування

Мова програмування Arduino є стандартною мовою C++ (використовується компілятор AVR-GCC) з деякими особливостями, які полегшують новачкам написання програм.

Найпростіша Ардуіно-програма складається з двох функцій:

- *setup()* – функція викликається одноразово при старті мікроконтролера.
- *loop()* – функція викликається після *setup()* в нескінченному циклі весь час роботи мікроконтролера.

Прикладом простої програми для платформи Arduino може слугувати код, для мигання вбудованим світлодіодом (лістинг 1.1).

Контролер Arduino вже містить резистор і LED-світлодіод, підключений до 13 виходу, тому ніяких інших зовнішніх радіоелементів не знадобиться.

В середовищі програмування Arduino напишемо поданий нижче код програми (скетч) та завантажимо його на пристрій.

Лістинг 1.1 – Мигання вбудованим світлодіодом

```
int ledPin = 13;
```

```
// LED приєднаний до виходу 13
```


```
void setup()
{
  pinMode(ledPin, OUTPUT);    // встановлюємо контакт 13 як вихід
}
void loop()
{
  digitalWrite(ledPin, HIGH); // вмикаємо LED
  delay(1000);                // пауза 1 секунда
  digitalWrite(ledPin, LOW);  // вимикаємо LED
  delay(1000);                // пауза 1 секунда
}
```

кінець лістингу 1.1

Для налаштування середовища Arduino IDE з конкретною платформою Arduino – необхідно вибрати назву моделі Arduino та номер присвоєного платі COM-порту.

Для встановлення моделі плати Arduino необхідно зайти в меню: *Інструменти – Плата* і вибрати плату (наприклад, «Arduino Uno», рис. 1.3).

Для вибору номера COM-порту потрібно перейти в меню: *Інструменти – Порт* і вибрати потрібний порт (рис. 1.4). Тепер середовище Arduino IDE налаштовано для прошивки плати.

Щоб завантажити прошивку, потрібно натиснути кнопку Вивантажити на верхній панелі інструментів .

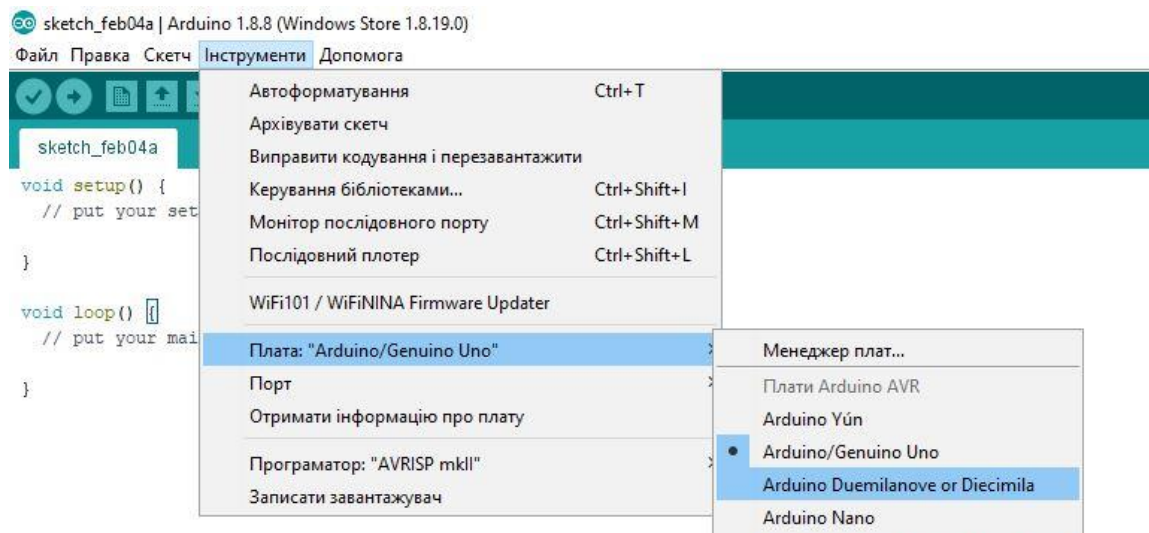


Рисунок 1.3 – Вибір плати Arduino

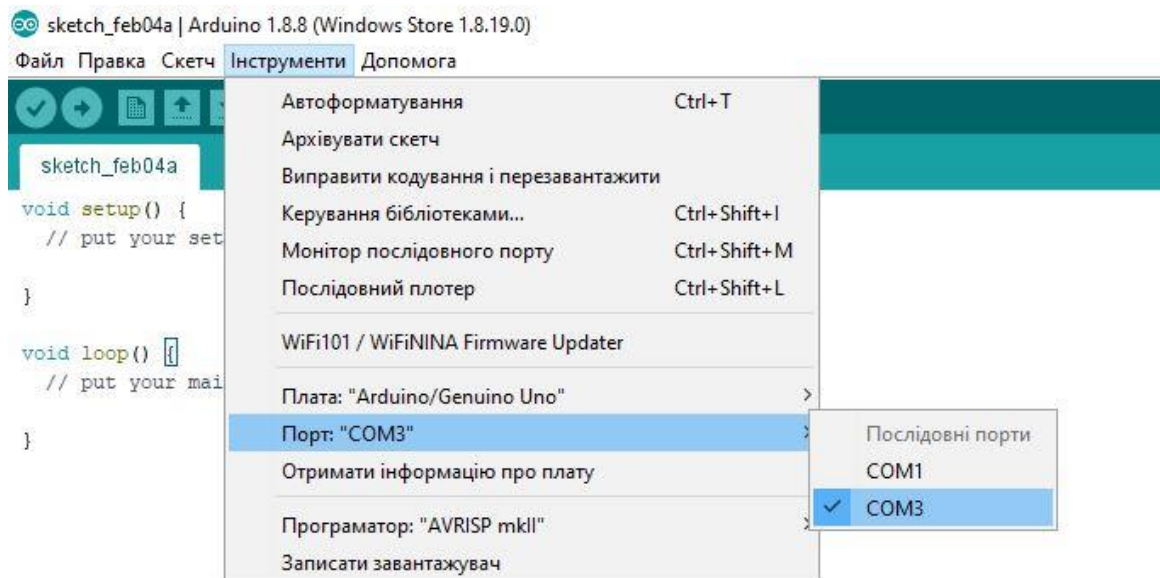


Рисунок 1.4 – Вибір порту підключення

Функція pinMode

pinMode в Arduino допомагає встановити режим роботи піна для виконання операцій зчитування або запису. Як правило, ця функція використовується всередині методу setup() і виконується один раз при запуску програми. Неправильне встановлення режиму входів і виходів Arduino може позначитись на працездатності проекту та плати.

Синтаксис функції:

pinMode(pin, type)

Функція значень не повертає.

В параметрі *pin* вказується порт, який потрібно конфігурувати. Як правило, функція використовується для налаштування цифрових пінів, тому що аналогові піни найчастіше використовуються як вхідні, а вхідний режим використовується Arduino за замовчуванням.

В параметрі *type* вказується тип режиму роботи піна: INPUT, OUTPUT, INPUT_PULLUP.

Для встановлення режиму роботи Arduino в основному використовується два варіанти параметрів: INPUT та OUTPUT. Але іноді при роботі з датчиками потрібно виставити пін в режим з неявно підключеним внутрішнім резистором, тому використовується ще одна константа, яка визначає тип піна INPUT_PULLUP.

Режим INPUT визначає високоімпедансний стан піна для роботи з зовнішніми джерелами сигналів (як правило, це різні варіанти датчиків, від яких надходять покази). Іншими словами, в цьому стані ви можете підключити практично будь-яке навантаження, тому що до входу неявним чином підключається високоомний резистор.

В режимі OUTPUT Arduino підтримує пін в низькоімпедансному стані, при якому на зовнішній пристрій видається максимально можливий струм (плата стає джерелом струму). В цьому режимі зазвичай працюють піни Arduino з підключеними світлодіодами, малопотужними сервоприводами, реле, драйверами двигунів та іншими зовнішніми пристроями.

За замовчуванням, всі піни Arduino встановлені в режим INPUT, тому вказувати це явно не потрібно.

Функція digitalWrite

Функція digitalWrite() – одна з найпопулярніших функцій Arduino. Вона зустрічається практично в будь-якому проекті, де потрібно подати сигнали на підключені пристрої.

Функція дозволяє керувати підключеним обладнанням шляхом подачі або зняття робочої напруги (в більшості випадків – 5В) на піни. Подаючи напругу на підключені пристрої, ми змушуємо їх щось робити: наприклад, вмикати світло, видавати звуки, вмикати двигун. Крім того, за допомогою `digitalWrite` можна подавати імпульси і передавати інформацію закодованими сигналами і повідомленнями.

Дана функція може сформувати тільки цифровий сигнал в двійковому коді (1 – висока напруга, 0 – низька напруга). На відміну від аналогового сигналу, не можна точно виставити на виході, наприклад, 4,15 вольт. Можливі тільки два значення напруги: мінімальне і максимальне робоче (наприклад, 0 В і 5 В відповідно).

Незважаючи на свою назву, функція може керувати також і аналоговими пінами, виставляючи в них на виході 0 або 5 вольт.

Час виконання функції на платі Arduino Uno складає близько 5 мікросекунд.

Синтаксис функції:

digitalWrite(pin, value)

pin – будь-який з пінів підключеної плати Arduino.

value – 1 або 0 (краще використовувати константи HIGH і LOW відповідно).

Функція значень не повертає.

Використання функції `delay`

Функція `delay` являється найпростішою командою і її найчастіше використовують новачки. По суті вона робить затримку, яка призупиняє роботу програми, на вказане в дужках число мілісекунд. Максимальне значення може бути рівне 4294967295 мс, що приблизно дорівнює 50 діб.

Підключення світлодіода

Світлодіод – це напівпровідниковий прилад, який створює оптичне випромінювання при пропусканні через нього електричного струму (рис. 1.5). Світлодіод випромінює світло тільки при проходженні струму в прямому напрямку, завдяки чому при його підключенні необхідно дотримуватись полярності. При цьому сила випромінювання безпосередньо залежить від сили струму. Однак не можна перевищувати максимально допустимий струм, встановлений для світлодіода його характеристиками, інакше світлодіод вийде з ладу.

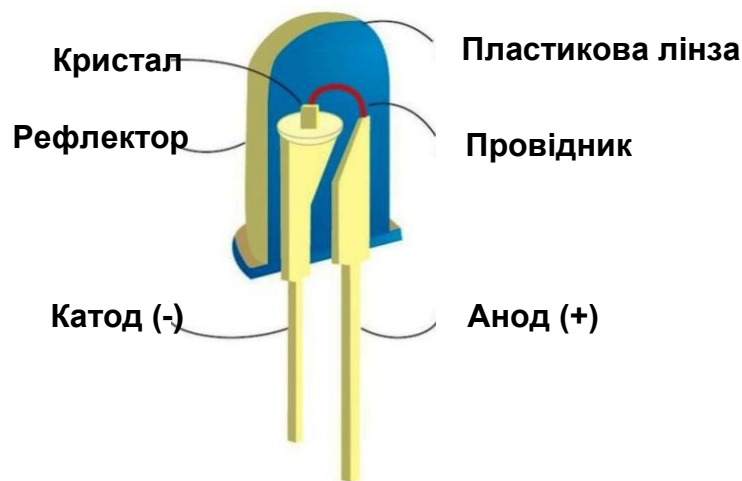


Рисунок 1.5 – Будова світлодіода

Також світлодіод характеризується падінням напруги при проходженні струму в прямому напрямку. Падіння напруги істотно вище ніж у звичайних діодів і становить 2-3 вольт, в залежності від кольору світлодіода. Це означає, що світлодіод не буде світитися якщо подана на нього напруга менша цього значення. Але якщо на світлодіод подати більшу напругу, то через нього потече максимально можливий струм, і світлодіод вийде з ладу. Щоб цього не сталося світлодіод необхідно підключати через струмообмежуючий резистор.

Розрахунок опору струмообмежувального резистора проводиться за такою формулою:

$$R_{св} = (V_{кер} - V_{св})/I_{св};$$

де:

$V_{кер}$ – напруга керування, це напруга на виході мікроконтролера, рівна 5 В;

$V_{св}$ – падіння напруги на світлодіоді, вказане в характеристиках;

$I_{св}$ – сила струму світлодіода, зазначена в характеристиках, зазвичай в межах 10-20 мА.

Для звичайних світлодіодів падіння напруги становить 2 В і рекомендований струм світіння 15 мА. При підключенні до виходу Arduino, живиться від напруги 5 В:

$$R_{св} = (5 - 2) / 0.015 = 200 \text{ Ом}$$

З урахуванням допуску можна використовувати резистори номіналом 220 Ом.

Хід роботи

1. Побудуйте схему підключення світлодіодів до Arduino згідно варіанту завдання.
2. Запрограмуйте Arduino згідно завдання.
3. Оформити звіт по роботі. Звіт повинен містити тему та мету роботи, короткі відомості про функції, що розглядаються в роботі, рисунок побудованої схеми підключення та код програми Arduino.

Індивідуальне завдання

Варіант 1, 4, 7, 10, 13, 16, 19, 22, 25

До плати під'єднуються два світлодіоди.

Алгоритм роботи програми: засвічується світлодіод 1; засвічується світлодіод 2; гасне світлодіод 1; гасне світлодіод 2. Інтервал між виконанням дій – 1 секунда.

Варіант 2, 5, 8, 11, 14, 17, 20, 23

До плати під'єднуються два світлодіоди.

Алгоритм роботи програми: засвічується світлодіод 1; засвічується світлодіод 2; гаснуть світлодіоди 1 та 2. Інтервал між виконанням дій – 1 секунда.

Варіант 3, 6, 9, 12, 15, 18, 21, 24

До плати під'єднуються два світлодіоди.

Алгоритм роботи програми: засвічується світлодіод 1; засвічується світлодіод 2; гасне світлодіод 2; гасне світлодіод 1. Інтервал між виконанням дій – 1 секунда.