

Лабораторна робота №1

Ознайомлення з мікроконтролерами сімейства Arduino та з програмним середовищем Arduino IDE

Мета роботи: Ознайомитись із контролерами сімейства Arduino, визначити їх можливості та базові принципи їх програмування для побудови систем малої автоматизації.

1.1 Теоретичні відомості

1.1.1 Короткий опис сімейства мікроконтролерів Arduino

Arduino – це контролер ATMega компанії Atmel поширеного 8-розрядного сімейства AVR. До нього додається блок живлення і мікросхеми послідовного чи USB - інтерфейсу. Через останній відбувається завантаження програм користувача і, при необхідності, обмін даними між персональним комп'ютером і платою Arduino під час виконання програми.

Типова плата Arduino надає в розпорядження користувача 14 цифрових входів чи виходів, з них типово шість можна використовувати як аналоговий вихід (із 8-розрядний ШІМ, ще шість входів можуть приймати аналогові сигнали із 10-розрядним апаратним АЦП послідовного наближення. У якості додаткових інтерфейсів передбачені шини SPI і I2C.

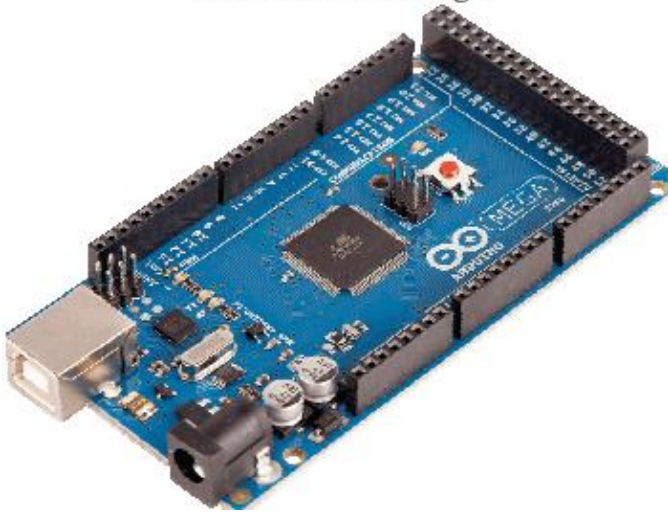


Існує багато варіантів виконання плат Arduino. Оригінальні вироби виробляє італійська фірма Smart Projects. Існують також численні клони і копії інших постачальників, нарешті, зустрічаються вільно поширювані апаратні засоби, у тому числі плати для самостійного виготовлення. Найбільш розповсюдженими платами є ArduinoMega, ArduinoUno, Arduino Mini, Arduino Nano та Freeduino.

Технічні характеристики деяких із плат наведені нижче. Для Arduino також випущено безліч плат розширення, деякі із котрих будуть використані у наступних лабораторних роботах. Серед них є плати різних провідникових і без провідних інтерфейсів, плати контролю двигунів, давачів різної складності, адаптерів різних інтерфейсів і індикаторів, тощо. Цікавою є плата Arduino ProtoShield, що

					ІКАТ.420.007.007-ЗЛ6	Арк.
						1
Змін.	Арк.	№ докум.	Підпис	Дата		

призначена для прототипування саморобних пристроїв без пайки. Усі експерименти можна проводити на панелі із контактними гніздами.

Табл. 1.1

Зображення та назва	Технічні характеристики
<p style="text-align: center;">Плата Arduino Mega.</p> 	<p>Мікроконтролер ATmega1280. Flashпам'ять 128 кбайт, RAM-пам'ять 8 кбайт, EEPROM 4 кбайт. Тактова частота 16 МГц. 54 цифрових канали вводу/виводу, із них 14 можуть використовувати ШІМ із стандартними бібліотеками. 4 апаратних порти UART. Апаратні інтерфейси I2C, SPI.</p> <p>16 аналогових входів із 10 розрядним АЦП. Інтерфейс USB</p> <p>Габаритні розміри приблизно 101 x 53 x 12 мм.</p>
<p style="text-align: center;">Arduino Mini</p> 	<p>Мікроконтролер ATmega168 з тактовою частотою 16 МГц. Порт USB, EEPROM 512 біт, ОЗП 1 кбайт, Flash 16 кбайт (2 кбайт використовуються для початкового завантаження системи). 14 цифрових каналів вводу/виводу, 6 із них можуть використовуватись для ШІМ. Вісім аналогових входів із 10 розрядним АЦП.</p>
<p style="text-align: center;">Arduino Nano</p> 	<p>ATmega328 або 168 з тактовою частотою 16 МГц. Програмування через вбудований USB. Функція автоматичного скидання.</p> <p>14 цифрових каналів вводу/виводу, 6 із них можуть використовуватись для ШІМ. Вісім аналогових входів із 10-розрядним АЦП. Flash-пам'ять об'ємом 32 кбайт або 16 кбайт. ОЗП 1 кбайт. EEPROM 512 байт або 1 кбайт.</p> <p>Габаритні розміри 18x43 мм.</p>

1.1.2. Середовище розробки Arduino

Програмування апаратних засобів Arduino зручно здійснювати у середовищі ArduinoIDE (Integrated Development Environment), проте, можливе програмування Arduino із використанням, наприклад, AVRStudio. Загальний вигляд вікна даної програми показаний на наступному рисунку.

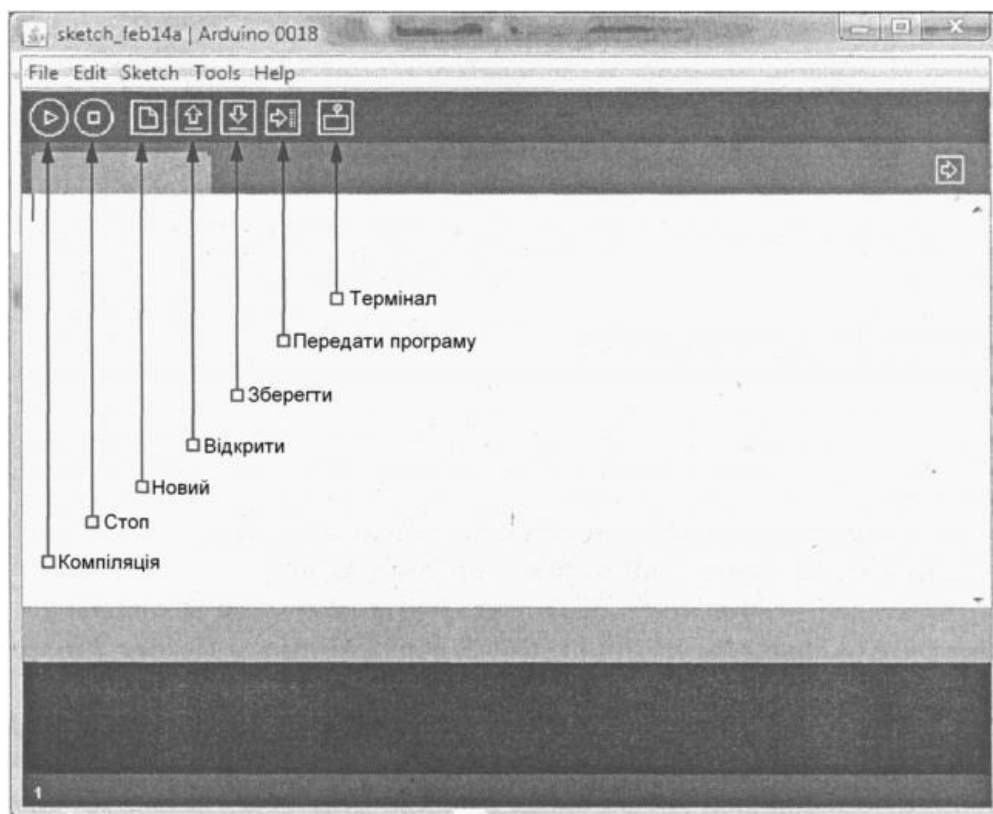


Рис.1.2 Середовище розробки Arduino

Перед тим як відкрити середовище розробки (IDE) необхідно приєднати плату мікроконтролера до персонального комп'ютера і встановити драйвери належним чином. Інакше в списку не з'явиться COM-порт. Потім слід встановити швидкість в бодах в терміналі. Для цього потрібно натиснути піктограму Terminal і задати швидкість 9600 бод. Ця швидкість є типовою для більшості програм. Для перевірки програмних і апаратних засобів, варто використати просту програму. При цьому спочатку у меню File знайдемо команду New, з'явиться нове вікно редагування. Далі слід набрати код , що приведений нижче та зберегти його. Краще за все створити спеціальну папку, наприклад,

					ІКАТ.420.011.037-ЗЛ1	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		4

Program_Arduino та вказати в налаштуваннях цю папку як каталог для збереження програм. При вказані назви, "Blink" у якості імені програми, ArduinoIDE створює підкаталог у обраному каталозі із іменем Blink і зберігає у ньому файл під іменем Blink.pde.

1.1.3. Основний цикл програми та переривання.

При автоматизації обладнання код програми звичайно виконується у нескінченному циклі. На рис.1.3 зліва показано виконання послідовності команд "ввід даних — обробка — вивід".

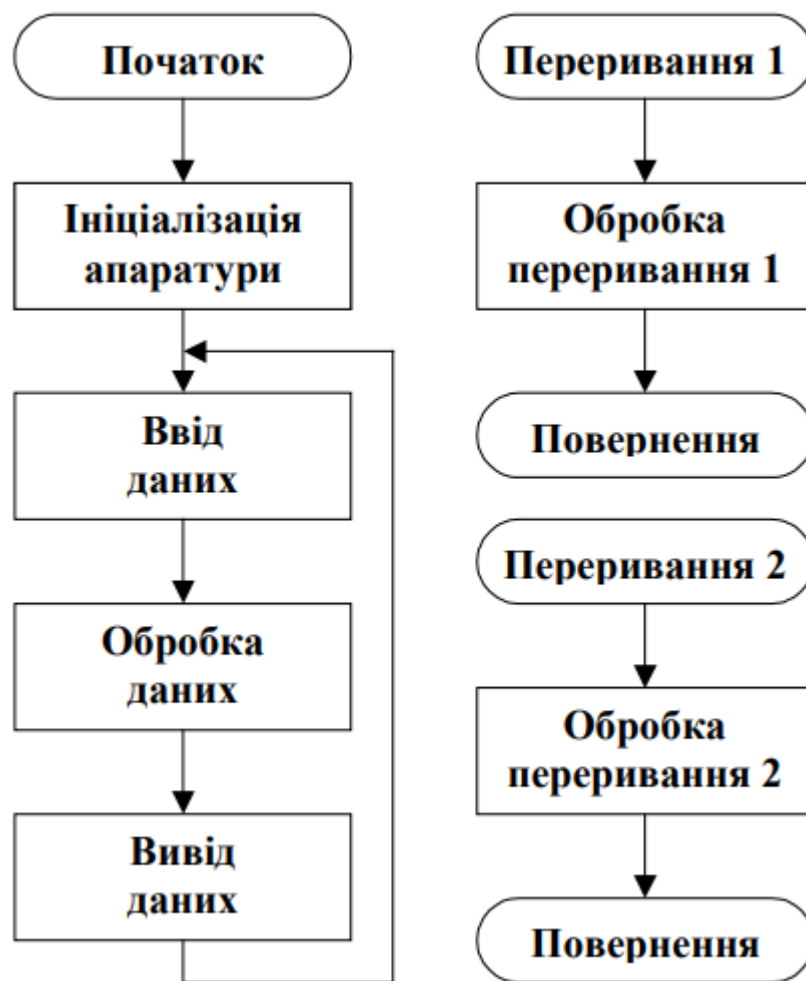


Рис.1.3 Основний цикл та обробка переривань

Іноді виникає необхідність провести швидке обслуговування того чи іншого обладнання. Це здійснюється за допомогою переривань (Interrupt). Основна частина програми виконується як і при послідовному програмуванні,

але коли відбудеться зовнішнє чи внутрішнє переривання, наприклад при натисканні кнопки, процесор перерве виконання основного циклу і відбудеться перехід на підпрограму обробки переривань (InterruptRoutine). Після завершення обробки переривання контролер знову перейде до виконання основного циклу програми. Для успішного повернення у основну програму слід на початку підпрограми обробки переривання зберегти всі регістри процесора, що можуть бути змінені підпрограмою обслуговування, а у кінці підпрограми збережені значення мають бути відновлені.

Програма Arduino складається з наступних основних частин:

- Коментарі та опис програми.
- Заголовки файлів і підключення бібліотеки.
- Оголошення глобальних змінних.
- Стандартна настройка void setup () (порти і конфігурація).
- Основний цикл void loop ().
- Власні процедури.

Операторні дужки виділяють фрагмент коду для компілятора. Блок коду завжди відкривається символом { і завершується символом }. Між ними можуть бути розміщені команди, опис змінних, тощо. Всі змінні, описи процедур, та інші об'єкти задані у дужках після закриття дужок стають недоступними для компілятора. Вважають, що операторні дужки задають область видимості для описаних параметрів. Область видимості простягається від моменту опису до закриття відповідної операторні дужки.

1.1.4. Типи даних і змінні

У середовищі розробки Arduino C є відмінність між великими та малими символами у імені змінної. Також не на першій позиції в Arduino у іменах змінних допускається вживання символу _ і цифр. Ключові слова (if, while, do і т.д.) не можуть бути іменами змінних. Імена глобальних змінних і функцій не можуть збігатися. Крім того, не допускається одночасне завдання функцій і

					ІКАТ.420.011.037-ЗЛ1	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		6

локальних змінних з одним і тим же ім'ям. Для опису у розпорядженні програміста є різні типи змінних. Тип змінної завжди потрібно ставити перед її описом.

Якщо змінна оголошується в межах функції, процедури або як аргумент функції, вона є локальною. Це означає, що змінна існує тільки всередині своєї функції. Змінна, оголошена поза функцією, є глобальною. Вона визначена для всіх функцій в межах програми.

1.2. Виконання роботи

Середовище розробки TinkerCAD

Середовище розробки TinkerCAD – це онлайн сервіс який включає в себе декілька типів програм типу CAD, САЕ та власне хмарне сховище. На базі даного сервісу можливо створювати типові проєкти Arduino, складати схеми та писати програмні коди, які в подальшому можливо записувати на плати сімейства Arduino. На рис. 1.4 показаний інтерфейс сервісу для програмування плати Arduino, також на даному зображенні можна побачити функцію моделювання, яка дозволяє перевірити створений програмний код.

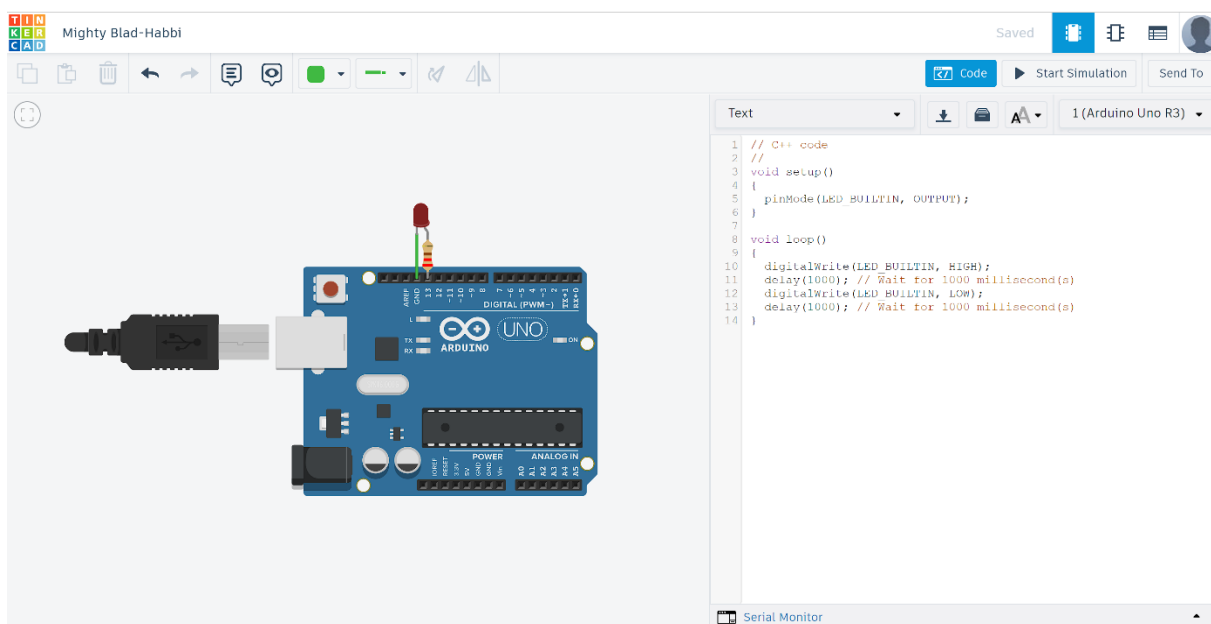


Рис. 1.4 Інтерфейс TinkerCAD

					ІКАТ.420.011.037-ЗЛ1	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		7

Нижче вказано послідовність кроків (алгоритм дій студента) для початку ознайомлення з онлайн платформою TinkerCAD.

1. Запуск онлайн сервісу TinkerCAD.

Для ознайомлення з мікроконтролерами використаємо он-лайн віртуальну лабораторію Tinkercad від Autodesk. Tinkercad дозволяє моделювати схеми на основі мікроконтролера Arduino.

Реєструємося на порталі: <https://www.tinkercad.com/>

У особистому кабінеті обираємо моделювання мікросхем: Circuits та активуємо кнопку **Створити ланцюг**.

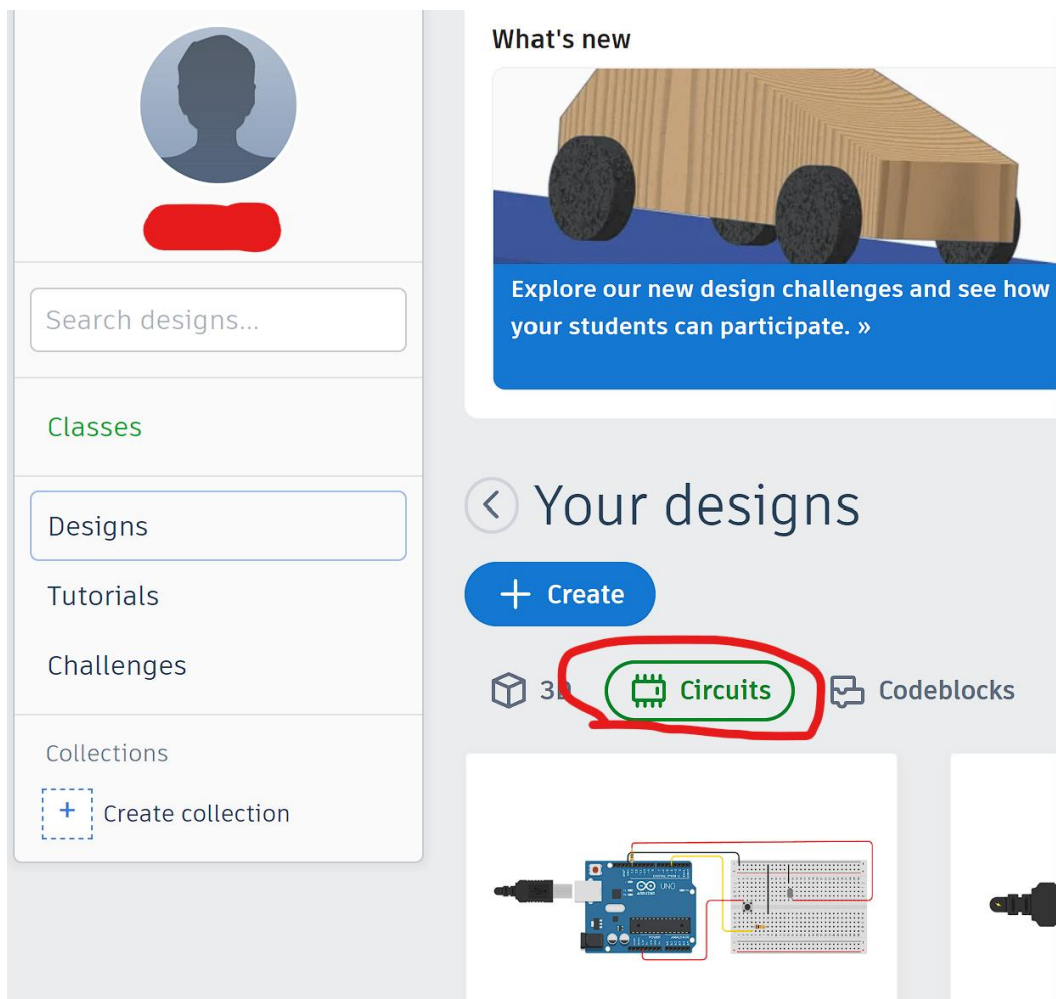


Рис. 1.5 Особистий кабінет TinkerCAD

2. Робота з компонентами. Відпрацювати роботу з основними компонентами.

Можна перетягувати на робоче поле Компоненти. Перелік компонент з правої частини вікна. Клікнувши курсором мишки на окремі компоненти, можливо виділити їх для подальшого перетягування мишкою або видалення клавішею клавіатури **delete**.

Затиснувши **Shift**, утримуючи праву кнопку мишки, протягуючи вказівником мишки по робочі області, можливо виділяти сукупності компонентів.

При виборі компонентів можливо залишити тип **Базові**, але оберемо **Arduino**.

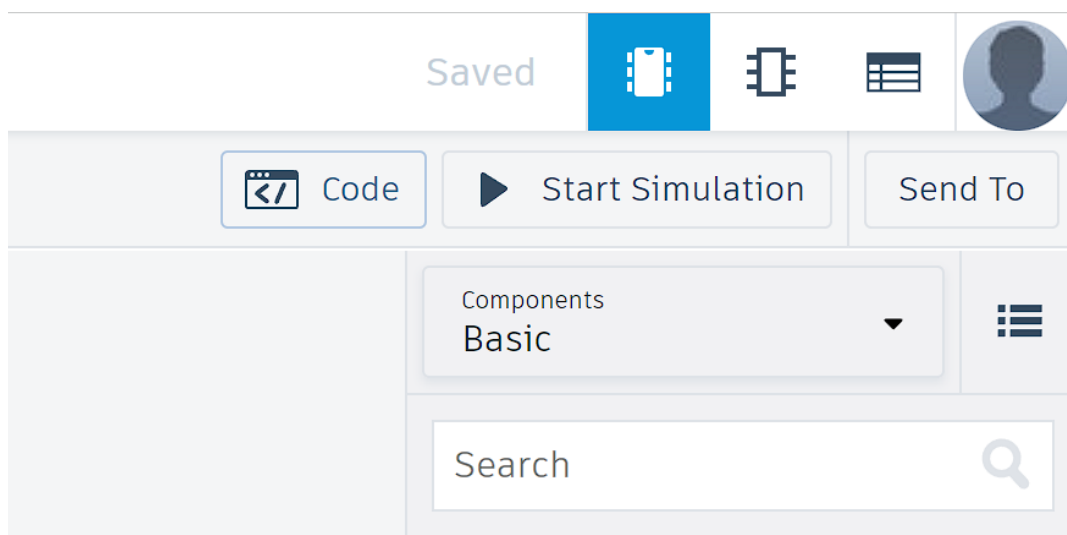


Рис. 1.6 Панель компонентів Tinkercad для Arduino

Почати з пустого arduino: тобто заготовку **Blink**, див. рис. 1.7.

					ІКАТ.420.011.037-ЗЛ1	Арк.
						9
Змін.	Арк.	№ докум.	Підпис	Дата		

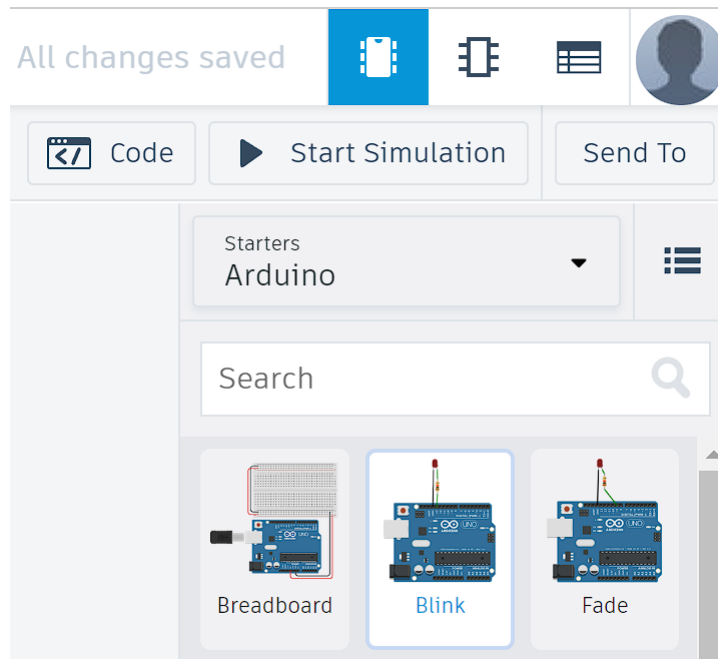


Рис. 1.7 Підключення LED до плати Arduino UNO

3. Запуск моделювання.


Можливо одразу запустити проект кнопкою меню  Почати моделювання. Під час виконання програми в Arduino, її можна перезапустити, натиснувши на платці arduino на кнопку reset:



Рис. 1.8 Кнопка reset на платі Arduino UNO

Після натиску кнопки початку моделювання світло діод має засвітитись, можна зупинити проект кнопкою **Зупинити моделювання**.

Наведемо мишку на виходи Arduino, до яких підключено світлодіод, та покрутити коліщатко мишки, щоб наблизити їх. Бачимо, що світлодіод

					<i>IKAT.420.011.037-3Л1</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		10

Звичайний однорядковий коментар:

```
// коментар 3
```

Функція **setup()**, яка виконується лише раз, на початку запуску програми **arduino**. Її назва підказує, що у ній краще розташовувати початкові налагодження: ініціалізацію змінних, активацію портів, налагодження виводів **arduino**.

```
void setup()
```

```
{
```

```
  Установка 1
```

```
}
```

Функція **loop()**, яка буде виконуватись одразу після функції, **setup()**, і буде продовжувати повторне (циклічне) виконання до зупинки самого **arduino**.

```
void loop() // !! це автоматичний нескінченний цикл
```

```
{
```

```
  Код циклу
```

```
}
```

Налагодження виводу з **arduino**, для його використання як виходу (входу), тобто такий вивід можна буде вважати дискретним виходом (входом).

```
pinMode(3, OUTPUT);
```

```
pinMode(3, INPUT);
```

					ІКАТ.420.011.037-ЗЛ1	Арк.
						12
Змін.	Арк.	№ докум.	Підпис	Дата		

Як видно на платі 13 виводів у arduino. Кожен з них можна налагодити як вихід для видачі з **arduino** сигналів на інші пристрої, так і як вхід для подачі сигналів **на arduino** з датчиків та інших пристроїв.

Як дискретні входи можливо налагодити також аналогові входи arduino:

```
pinMode(A3, INPUT);
```

Коли налагоджені виходи, на них можливо надсилати високі та низькі рівні напруг (тобто включати/виключати):

```
digitalWrite(13, HIGH);
```

```
digitalWrite(13, LOW);
```

Для збереження певних значень можливо оголосити змінні. В arduino змінні оголошуються на початковому рівні часто глобальними (і робиться це перед **void setup()**):

```
int dd = 0;
```

```
float gg = 0.0;
```

```
void setup()
```

```
{
```

```
}
```

```
void loop()
```

```
{
```

```
}
```

					ІКАТ.420.011.037-ЗЛ1	Арк.
						13
Змін.	Арк.	№ докум.	Підпис	Дата		

Коли налагоджені входи, з них можливо у змінні зчитувати 1 або 0 при отриманні arduino високого чи низького рівнів сигналу на вході:

```
dd = digitalRead(0);  
dd = analogRead(A0);
```

Команда затримки виконання

```
delay(1000); // тут 1000 будь-яка кількість мілісекунд
```

Команди перевірки умови такі ж як і в c++:

```
if (dd == 1) {  
    // певні дії  
}
```

або

```
if (dd == 0) {  
  
    // певні дії  
  
} else {  
  
    // певні інші дії  
  
}
```

Наприклад, можна перевірити чи натиснута кнопка, підключена до 6 входу arduino:

					ІКАТ.420.011.037-ЗЛ1	Арк.
						14
Змін.	Арк.	№ докум.	Підпис	Дата		


```

dd = digitalRead(6);
if (dd == HIGH) {

    // певні дії
}
або
dd = digitalRead(6);
if (dd == 1) {
    // певні дії
}

```

Порядок виконання роботи

1. Створіть новий проект circuits у TinkerCAD;
2. Переходимо в базові компоненти;

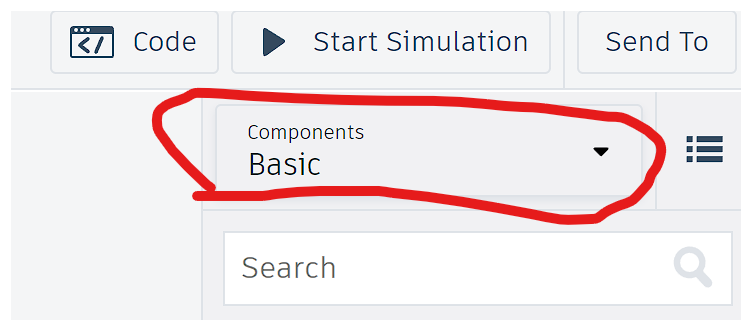


Рис. 1.10 Вибір типу компонентів

3. Обрати плату Arduino UNO. Щоб додати плату на робочий простір потрібно натиснути її та перетягнути з панелі компонентів на вільне місце;

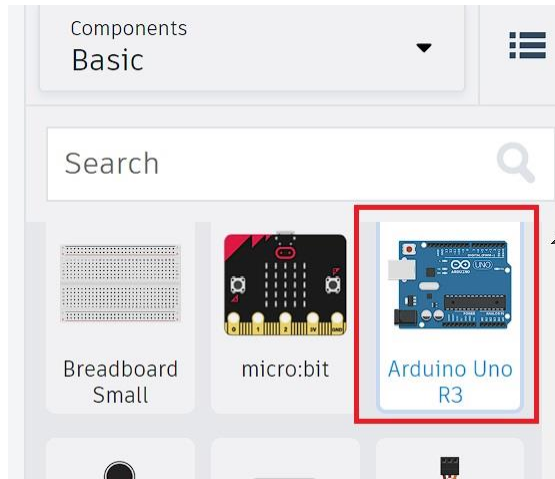


Рис. 1.11 Умовне зображення плати Arduino UNO

4. Обрати LED, або інша його назва світлодіод. Додати до робочого простору;

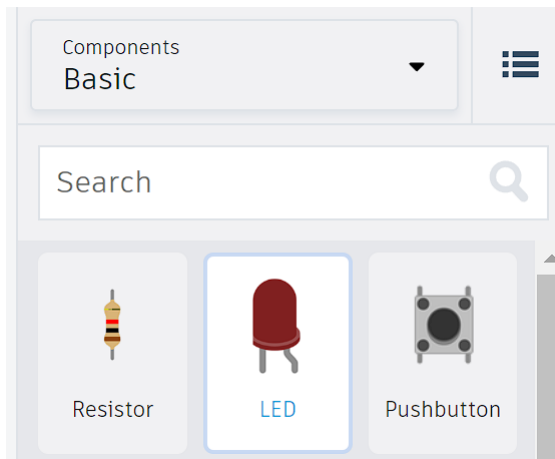


Рис. 1.12 Умовне зображення плати LED

5. З'єднати світлодіод та плату Arduino UNO. Катодом до входу GND, а анодом до 13 входу.

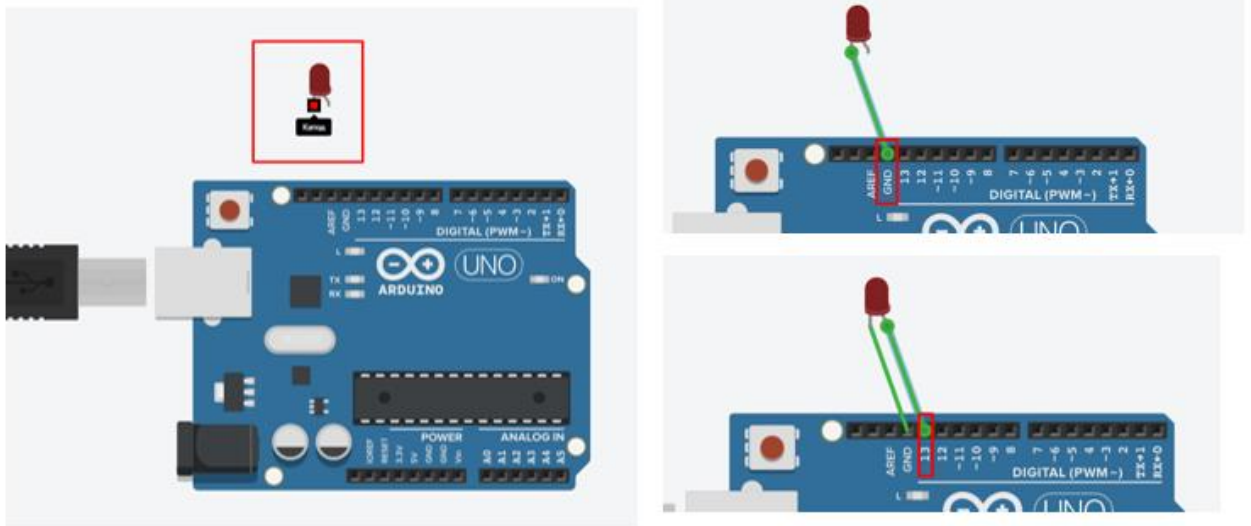


Рис. 1.13 З'єднання світлодіода до плати Arduino UNO

6. Відкрити вікно програмування, обрати тип відображення текст;

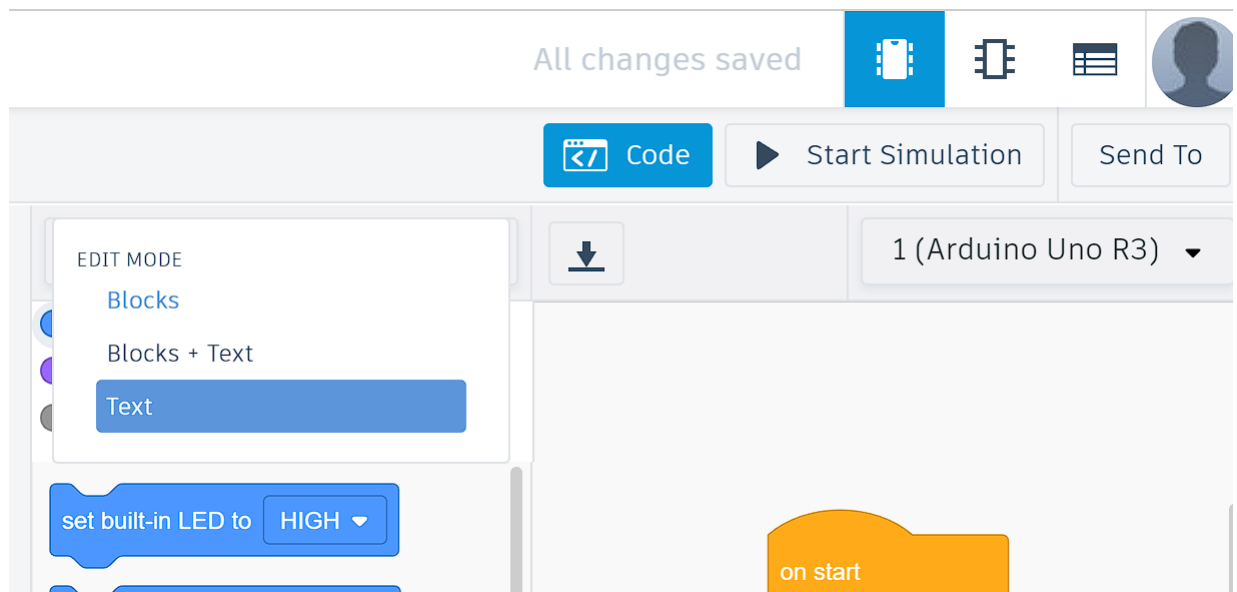


Рис. 1.14 Режим програмування плати

7. Написати коду управляючої програми для світлодіода (у випадку, якщо код приклад не активувався автоматично);

					ІКАТ.420.011.037-ЗЛ1	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		17

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(LED_BUILTIN, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(LED_BUILTIN, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14 }
```

Рис. 1.15 Код управляючої програми

8. Виконати моделювання коду;

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(LED_BUILTIN, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(LED_BUILTIN, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14 }
```

Рис. 1.16 Кнопка моделювання програми

9. Реалізувати представлену схему на рис. 1.17.

Необхідно приєднати світлодіод до плати Arduino UNO, а також написати програму, що буде керувати ввімкненням та вимкненням світлодіоду. Для підключення світлодіоду до плати Arduino необхідно на анод світлодіоду під'єднати резистор 220 Ом. Загальна схема підключення зображена на рис.1.17.

Світлодіоди мають вмикатись по черзі через деякий проміжок часу.

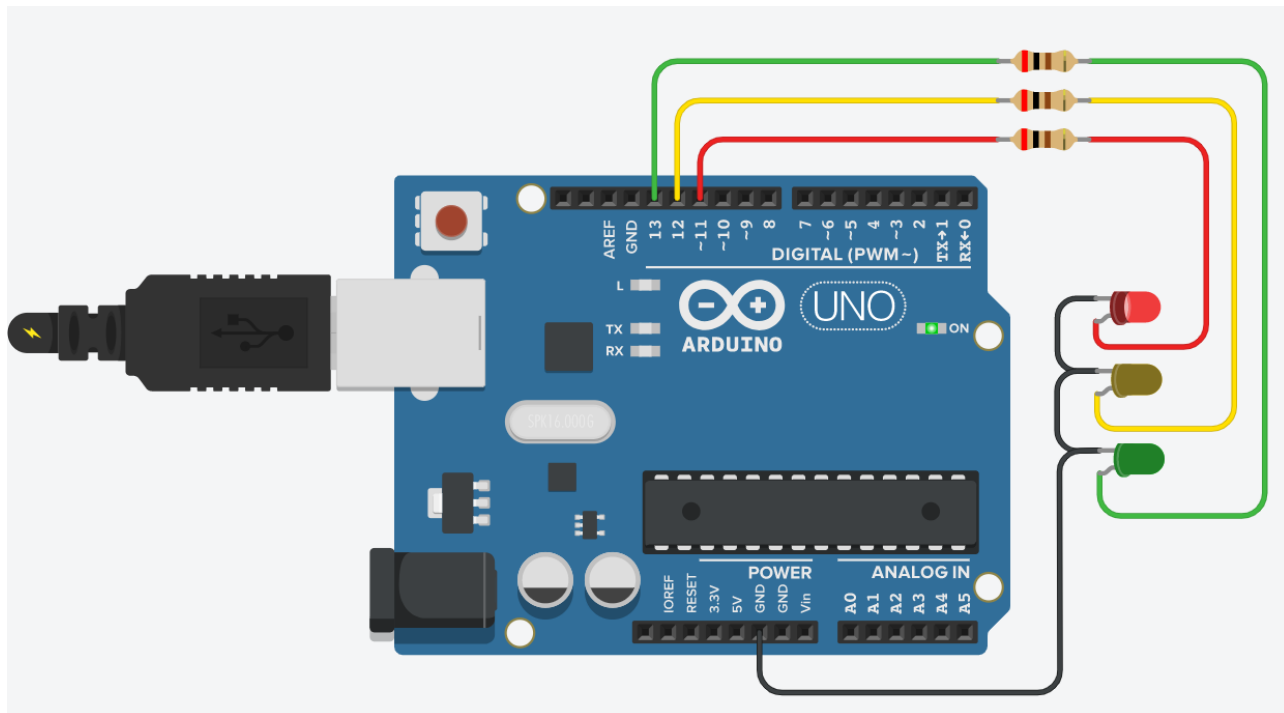


Рис.1.17 Загальна схема підключення світлодіодів

10. Виконання додаткового завдання.

Додаткове завдання 1 – підключити *RGB-світлодіод* та *кнопку* (рис. 1.18) до плати Arduino UNO. Розробити програмний код для Arduino UNO в ході роботи якого колір випромінюваного світла *RGB-світлодіода* буде змінюватись від натиску *кнопки*.

										Арк.
										19
Змін.	Арк.	№ докум.	Підпис	Дата						

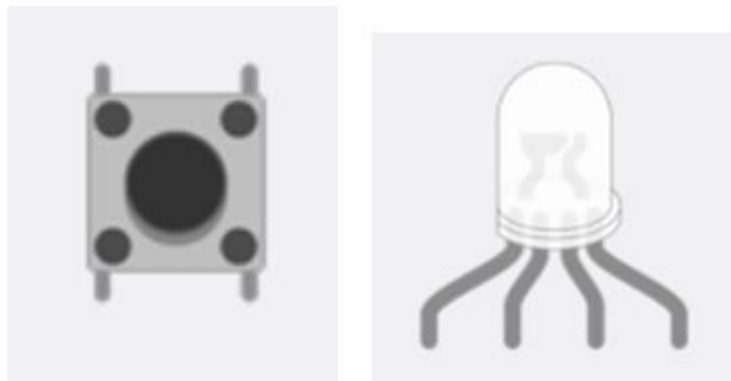


Рис. 1.18 Зображення кнопки та RGB-світлодіода

11. Оформлення звіту.

Описати результати виконаної роботи та написати висновок.

Контрольні питання

1. Які виходи на платі Arduino UNO приймають аналогові сигнали?
2. Яку функцію виконує вхід Vin на платі Arduino UNO?
3. Що означає позначення ~ PWM на цифрових входах/виходах?
4. Що виконує команда PinMode?
5. За що відповідає void setup?
6. Як організувати коментар в програмному коді?

					ІКАТ.420.011.037-ЗЛ1	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		20