**Лабораторне заняття 1**

**Основи обчислювальної та мікропроцесорної техніки**

**Мета:** ознайомитися з основами обчислювальної техніки, а саме: системами числення, булевою алгеброю, синтезом комбінаційних схем та автоматів з пам'яттю. Отримати практичні навички.

Зміст роботи

**Завдання 1:** Згідно з вказаним викладачем варіантом переведіть десяткове число А10 (таблиця 1.1) у двійкову та шістнадцяткову системи числення



У двійкову систему:

Ціла частина:

291(10) = 100100011(2)

|  |  |
| --- | --- |
| Число | Частка(біт) |
| 291/2 | 1 |
| 145/2 | 1 |
| 72/2 | 0 |
| 36/2 | 0 |
| 18/2 | 0 |
| 9/2 | 1 |
| 4/2 | 0 |
| 2/2 | 0 |
| 1 |  |

Дробова частина:

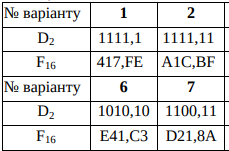
0,73(10)=0.1011101(2)

|  |  |
| --- | --- |
| 0 | 73 |
|  | \*2 |
| 1 | 46 |
|  | \*2 |
| 0 | 92 |
|  | \*2 |
| 1 | 84 |
|  | \*2 |
| 1 | 68 |
|  | \*2 |
| 1 | 36 |
|  | \*2 |
| 0 | 72 |
|  | \*2 |
| 1 | 44 |

У шістнадцяткову систему счислення:

123.BAE1

**Завдання 2**: Згідно з вказаним викладачем варіантом переведіть двійкове число D2 та шістнадцяткове число F16 (таблиця 1.2) в десяткову систему числення.



1. Переведення числа з двійкової у десяткому систему числення:

1100 => =8+4+0+0=12

0,11 => =0.5+0.25=0.75

1100,011=12.075

1. Переведення числа з шістнадцяткової у десяткому систему числення

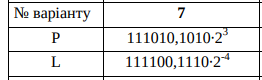
D21 =>

0.8A =

**Завдання 3:** Згідно з вказаним викладачем варіантом виконайте наступні дії над числами P2 і L2, що задані в формі з плаваючою комою (таблиця 1.3):

а) додавання;

б) віднімання



Для початку переведемо наші числа в нормальну форму, для цього виконаємо відповідне перетворення чисел. В результаті отримаємо наступне.

|  |  |  |
| --- | --- | --- |
| № варіанту | 7 | |
| P | 111010,1010∙23 | 111010101,0 |
| L | 111100,1110∙2-4 | 11,11001110 |

**Операція додавання.** Маючи відповідне представлення чисел в двійковій формі можемо провести операцію побітового додавання. Суть виконання даної опреації досить проста, під час виконання додавання варто пам’ятати декілька правил на основі яких і виконується додвання, під час додавання двох нулів в результаті отримаємо нуль, під час додавання нуля та одиниці в результат записується 1, під час додавання двох одиниць в результат записується нуль, але в старший розряд переноситься значення 1. Вище описані операції додавання зображені в таблиці №3.2.

Розпочнімо процес виконаня операції додавання над нашими числами P та L. Результат виконання операції додавання зобразимо у вигляді таблиці №3.3.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | , | 0 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 1 | 1 | , | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |  |
| = | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |  | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |  |

Виконаємо операцію віднімання на числами P та L. Виконання даної операції у нашому випадку може бути виконано у прямому коді, так як представлені числа мають однаковий знак числа. У випадку, коли числа мають протилежні знаки, операція віднімання не може бути виконано у прямому коді. Тоді така операція зводиться до виконання операції додавання над числами у прямому та доповняльному коді. Виконаємо операція віднімання над нашими числами. В такому випадку отримаємо наступний результат.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| - | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | , | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
|  |  |  |  |  |  |  |  | 1 | 1 | , | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |  |
| = | 1 | **1** | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | , | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

Як можемо побачити, операція віднімання виконана успішно, але в такому випадку отриманий результат віднімання не матиме достатньо точність для виконання деяких інших операцій. Тому така операція віднімання також можемо бути виконанна іншими способами, котрі забезпечують вищого рівня точності.

**Завдання 4**: Згідно з заданим викладачем варіантом запишіть досконалу диз'юнктивну нормальну форму (ДДНФ) для логічної функції F чотирьох змінних, яка задана таблицею істинності (таблиця 1.4).

Використовуючи карти Карно запишіть вираз для мінімальної форми і побудуйте функціональну електричну схему (перевірити роботу у Electronic Workbench(Multisim).

Реалізувати функцію логічного виразу на будь-якій мові програмування: асемблер, С, С ++, С # тощо. Вхідним параметром функції є однобайтова змінна.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | f |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
|  |  |  |  |  |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

- 0000

0010

x2 - 0100

x2x3 -0110

x1x4 -1001

x1x2x4 - 1101

x1x2x3 - 1110

x1x2x3x4 - 1111

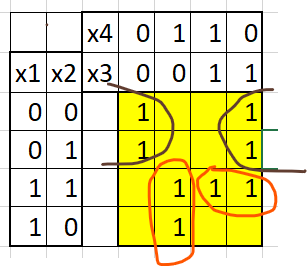
Записуємо досконалу диз`юнктивну нормальну форму (ДДНФ), за допомогою логічного додавання двійкових наборів на яких функція приймає значення, що дорівнює 1:

F36(x1x2x3x4)дднф = ++x2 +x2x3+x1x4 +x1x2x4 +x1x2x3+x1x2x3x4

¬x1¬x2¬x3¬x4 ∨ ¬x1¬x2x3¬x4 ∨ ¬x1x2¬x3¬x4 ∨ ¬x1x2x3¬x4 ∨ x1¬x2¬x3x4 ∨ x1x2¬x3x4 ∨ x1x2x3¬x4 ∨ x1x2x3x4

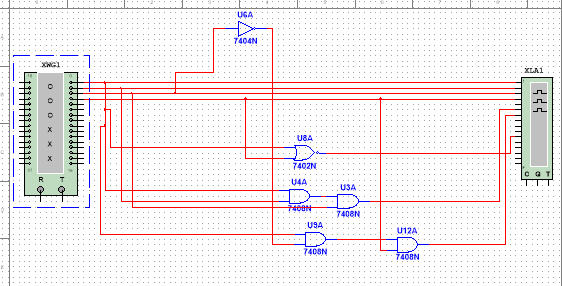
Мінімізуємо ДДНФ:

F36(x1x2x3x4)мдднф = ++x2 +x2x3+x1x4 +x1x2x4 +x1x2x3+x1x2x3x4 = x2 *+ +* x1x4 =

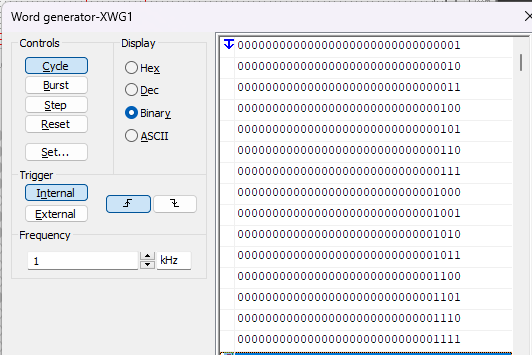


x1x2x3x4мдднф =

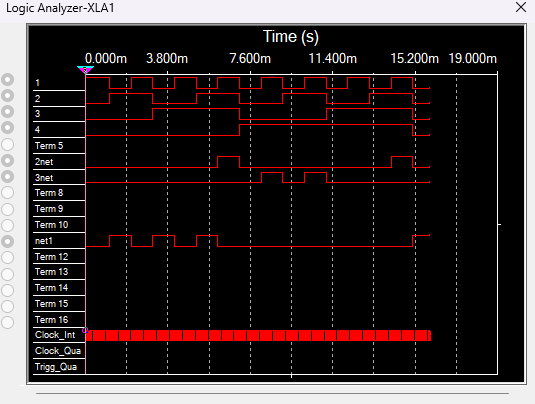
Побудуэмо схему для реалізації функції в Multisim

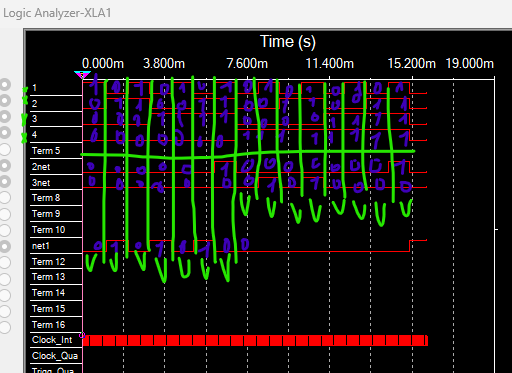


Протестуємо і тому для початку над побрабно задати набір сигналів в WordGenerator.



Подивимося на результат і як ми можемо спосперігати функія працює правильно.



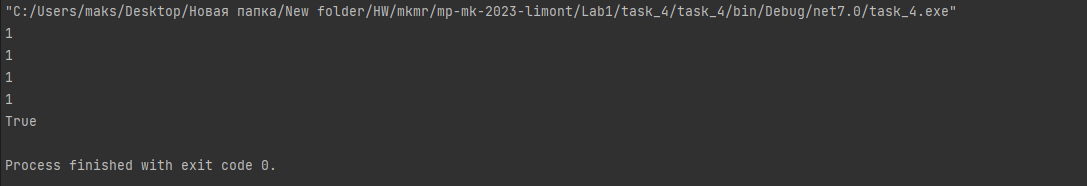


Реалізуєио цю функцію на мові програмування.

Программа реалізована на c# консольний додаток

int x1 = int.Parse(Console.ReadLine());  
int x2 =int.Parse(Console.ReadLine());  
int x3 =int.Parse(Console.ReadLine());  
int x4 = int.Parse(Console.ReadLine());  
  
if (x1 == x4 && x1 == 0)  
{  
 Console.WriteLine(true);  
}  
if (x1 == x2 && x2==x3 && x1 == 1)  
{  
 Console.WriteLine(true);  
}  
if (x1 == x4 && x3==0 && x1 == 1)  
{  
 Console.WriteLine(true);  
}

Результат програми:



**Висновки:** В ході виконання лабораторної роботи було закріплено навички та вміння з основами роботи обчислювальної техніки. Під час виконання лабораторної роботи було проведено роботу з двійковими числами, продемонстровано процес їх перетворення, додавання та вднімання. Проведено відповідні операції побудови логічних функцій на основі таблиці істинності, а також їх подальша мінімізація на основі карт Карно та методу поглинання. Також в ході виконання роботи було побудовано логічні функції для забезпечення роботи лічильника, робота якого зображена за допомогою графа.

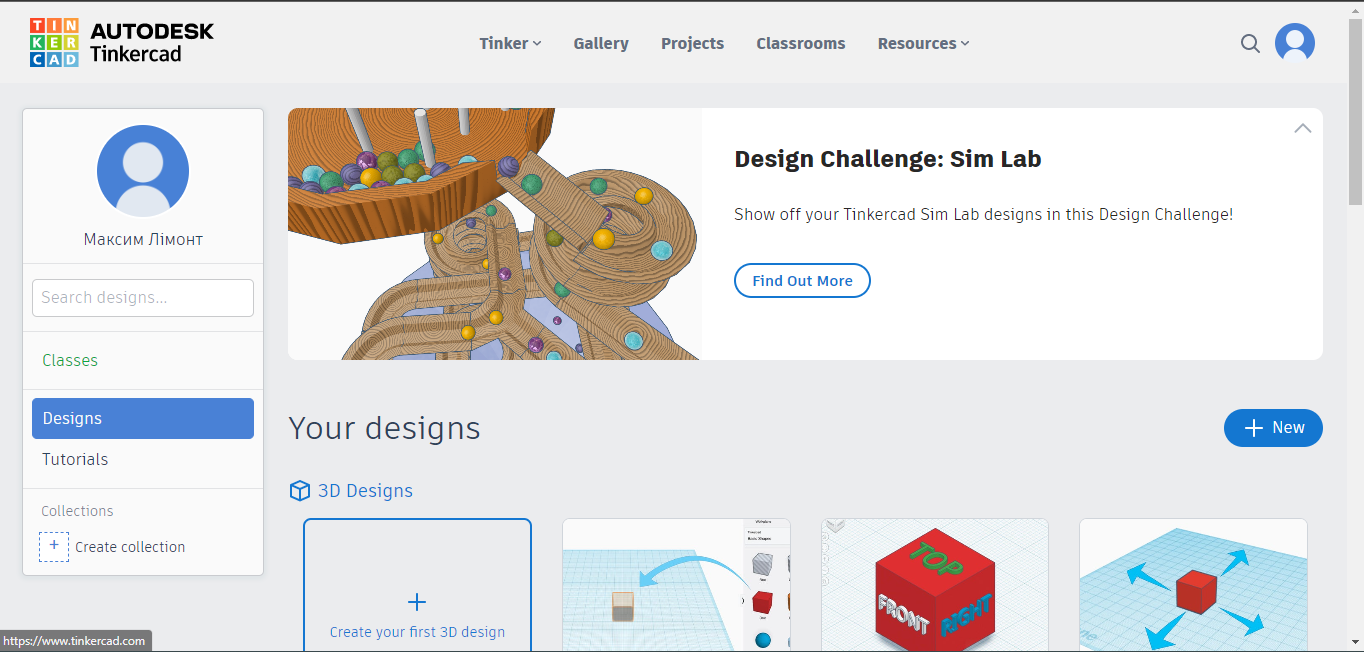
**ЛАБОРАТОРНА РОБОТА №2**

**Мета заняття:** ознайомитися з основними поняттями. Зареэструватися в емуляторы Thinkercad. Отримати практичні навички по роботі з емулятором.

**Хід роботи**

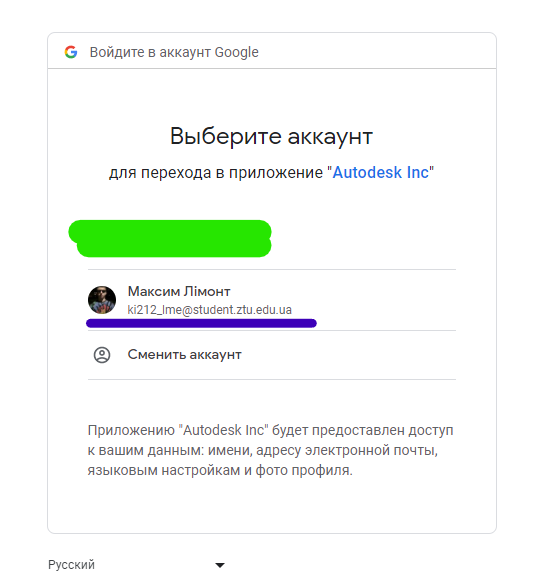
**Завдання №1**. Ознайомитися з основними поняттями.

Відповідно до отриманого завдання проведемо процес ознайомлення з онлайн сервісом Thinkercad. Tinkercad – це онлайн сервіс, який зараз належить найвідомішій компанії світу CAD-систем – Autodesk.Tinkercad безкоштовне середовище для навчання 3D-моделювання. Нещодавно Тінкеркад отримав можливість створення електронних схем і підключення їх до симулятора віртуальної плати ардуїно.

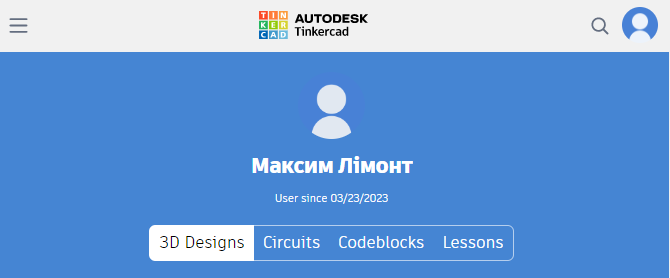


**Завдання №2.** Зареєструватися в емуляторі Tinkercad.

Відповідно до отриманого завдання, було проведено процес реєстрації на платформі Thinkercad для подальшого використання її в якості платформи для практичних робіт з радіокомпонентами. Використовуючи студентський аккаунт Google було виконано процес реєстрації

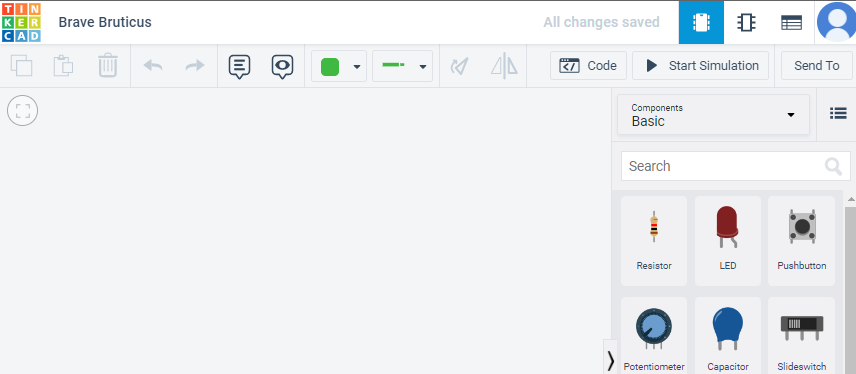


В результаті чого ми успішно виконали реєстрацію та вхід до онлайн сервісу Thinkercad

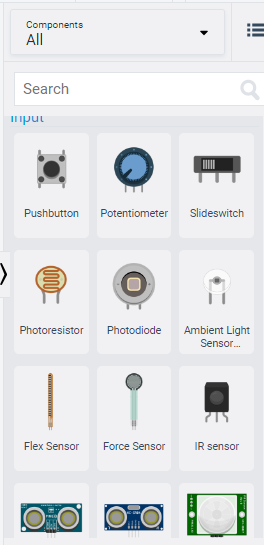
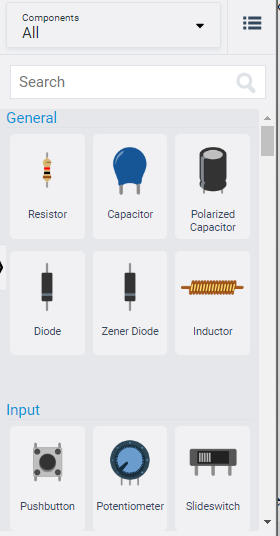


**Завдання №3.** Ознайомитися з роботою емуляторі Tinkercad. Розібратинся з візуальним редактором схем. Коротко описати елементи панелі компонентів.

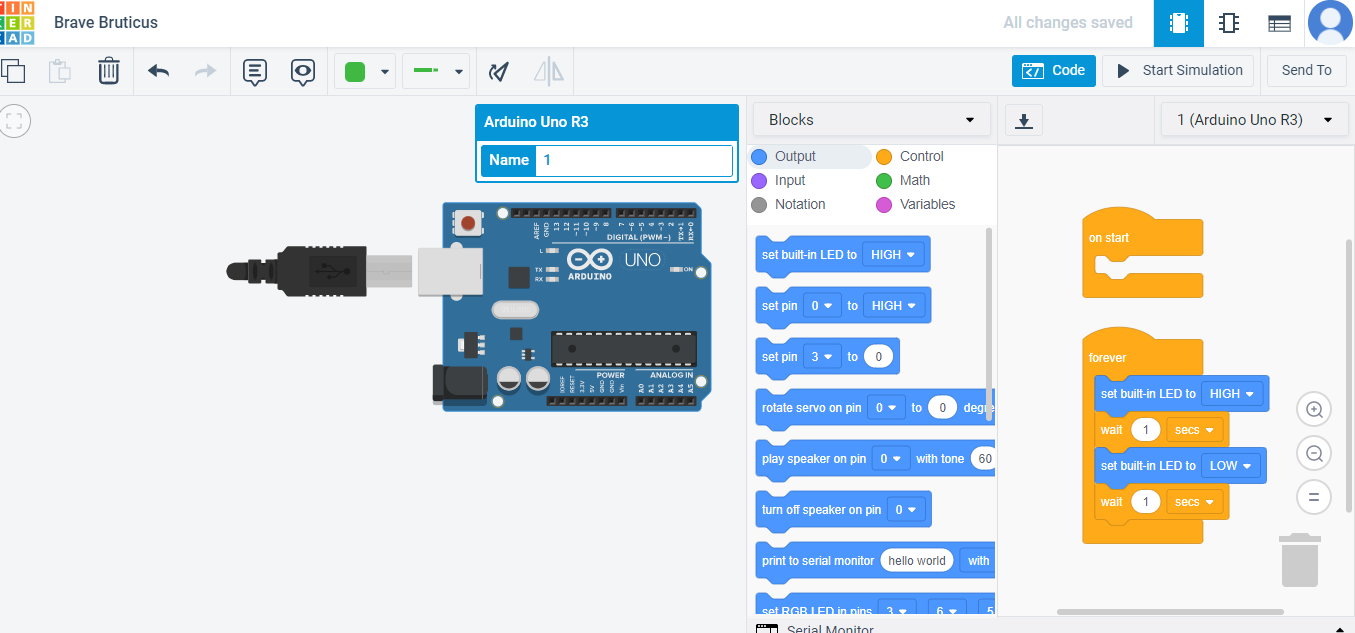
Розпочнімо виконання ознайомлення з інтерфейсом середовища. Для цього виконаємо опис основних компонентів даного середовища з якими будемо виконувати роботу



В першу чергу нас цікавитиме вікно вибору компонентів, на основі яких і будемо виконувати моделювання роботи різних комопнетів та виконувати процес керування ними за допомогою відповідних засобів. Панель вибору базових елементів розміщена у правій частині середовища, а також зображена на рисунку нижче. Для отримання більшого їх переліку варто натиснути на відповідне меню та обрати пункт «All».

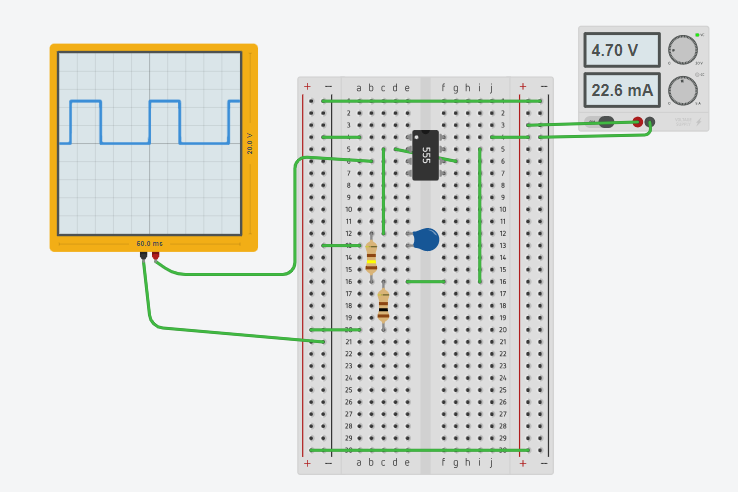


Для програмування використовується спростована мова програмування C++, на основі якиї і виконується програмування роботи мікроконтролерів, для прикладу платформи Arduino. Інтерфейс середовища програмування зображено на зображені нижче.

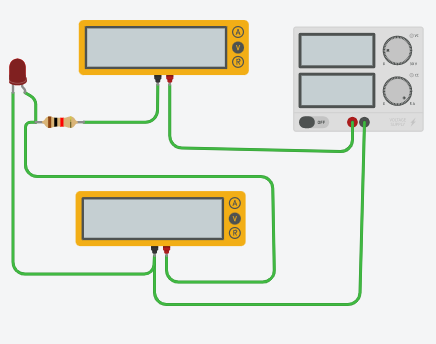


На цьому розгляд основних елементів середовища закінчено. Цього буде достатньо для проведення подальших процесів моделювання роботи різних схем

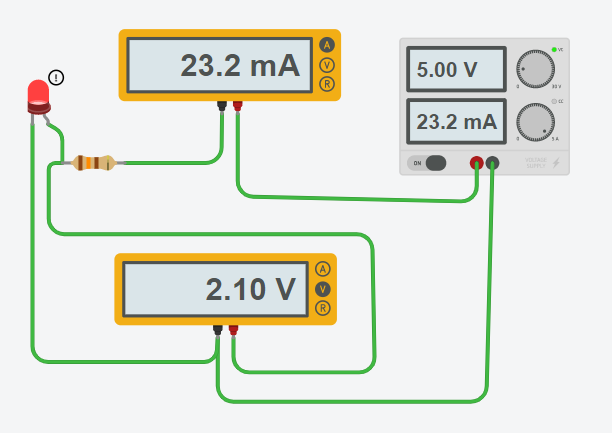
**Завдання №4.** Скласти схему генератора тактових імпульсів та дослідити її в емуляторі Tinkercad. Перейдемо до виконання практичного завдання на базі платформи Thinkercad відповідно до якої проведемо процес побудови схеми генератора тактових імпульсів та виконаємо процес дослідження схеми в емуляторі. В нашому випадку схема генератора прямокутникоподібних імпульсів базуватиметься на схемі таймера, що . наявна в програмному середовищі Tinkercad. В результаті проведених робіт в програмному середовищі було побудовано наступну схеми генератора прямокутникоподібних імпульсів.



**Завдання №5.** Розрахувати транзисторний ключ для світлодіода та перевірити роботу схеми в емуляторі Tinkercad.



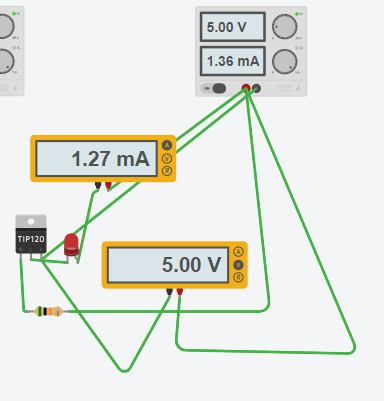
Для початку варто визначити два основних параметра для світлодіода, а саме споживана напруга та споживаний струм. Для будь-якого світлодіода при напрузі споживання 2,5В струм, що повинен протікати крізь світлодіод не повинен бути більший за 20 мА. Отже, в нашому випадку за параметри споживаного струму та напруги будемо приймати VL = 2,5В та IL = 20мА. Використовуючи отримані параметри виконаємо розрахунок параметрів опору резистора використовуючи закон Ома.



Перейдемо до процесу розрахунку значення транзисторного ключа. В нашому випадку, розрахунки будуть виконуватися для транзистора.

Тепер виконаємо розрахунок струму, що буде протікати кріз колектор та емітер транзистора знаючи опір бази транзистора

При встановлені више 125кОм було помилка в Tinkercad.тому опір був обраний майже максимально припустимий в 50000ОМ.



**Завдання №6.** Згідно з заданим викладачем варіантом запишіть досконалу диз'юнктивну нормальну форму (ДДНФ) для логічної функції Е чотирьох змінних, яка задана таблицею істинності (таблиця 2.1). Реалізувати функцію логічного виразу. Перевірити роботу схеми в емуляторі Tinkercad.



Використовуючи таблицю істинності для нашої функції виконаємо процес її мінімізації на основі метода карт Карно. Для цього виконаємо побудову карти та на її основі знайдемо ДДНФ для нашої функції

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | f |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

– 0000

-0001

– 0010

-0011

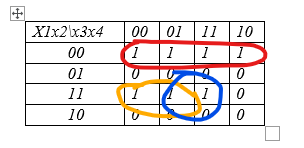
x1x2 – 1100

x1x2x4 – 1101

x1x2x3x4-1111

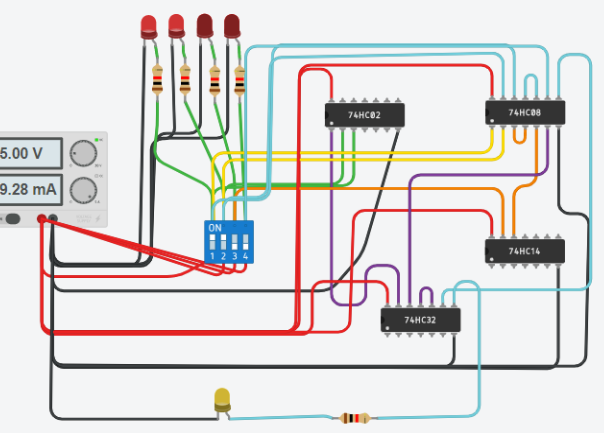
F7(x1x2x3x4)дднф = +  *+ +* x1x2+ x1x2x4 + x1x2x3x4 = + x1x2 *+* x1x2x3x4

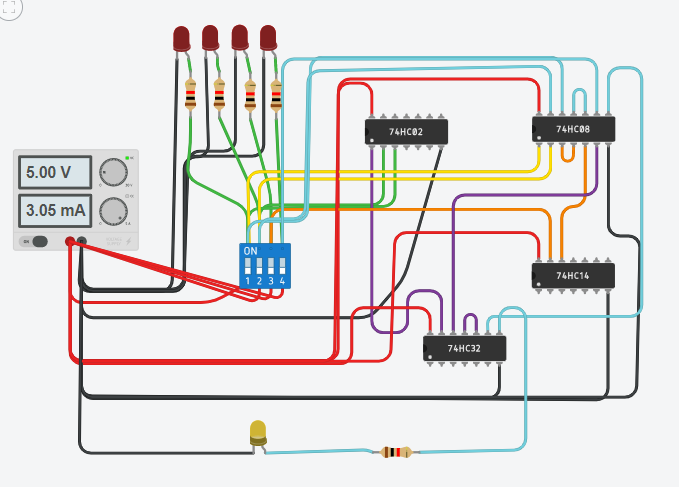
Мінімізуємо ДДНФ

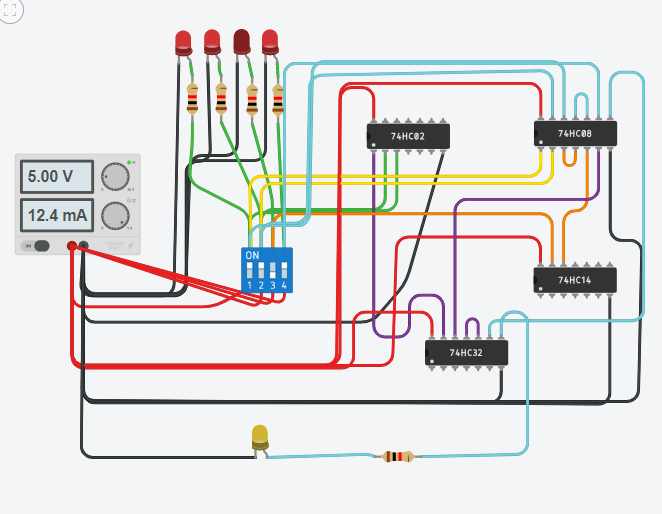


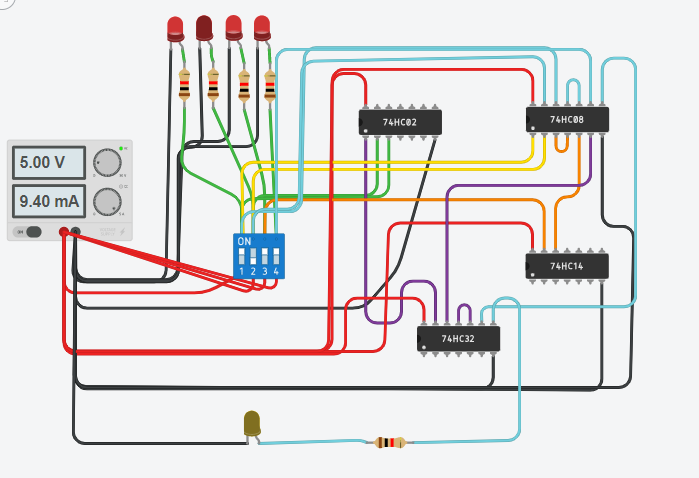
+ x1x2 *+* x1x2x4

Проведемо перевірку роботи даної схеми на базі логічних елементів. Для цього запустимо процес симуляції роботи схеми та проведемо тестування. Результати деяких логічних станів зображені на скріншотах нижче.









Посилання на схему: <https://www.tinkercad.com/things/78BysHpehRc>

В результаті проведеного тестування роботи схеми помилок в її роботі, а також в відповідності значень логічних станів до таблиці істинності функції, не було виявлено.

**Висновки**: В ході виконання лабораторної роботи було проведено ознайомлення з онлайн сервісом Thinkercad. На основі даного сервісу було проведено процес ознайомлення з можливостями цього середовища, а також на його основі було проведено моделювання роботи схеми транзисторного ключа для світлодіода, з відповідним виконання процесів розрахунку основних параметрів схеми, а також виконано моделювання роботи логічної функції у відповідно до отриманого завдання

**ЛАБОРАТОРНА РОБОТА №3**

***Мета заняття:*** ознайомитися з основними поняттями. Встаановити середовище розробки. Отримати практичні навички по роботі з відлагоджувальною платою та середовищем розробки.

**Початок роботи**

**Завдання №1.** Ознайомитися з основними поняттями.

Розпочнемо виконання лабораторної роботи. Для початку опишемо короткі теоретичні відомості про платформу Arduino.

В першу чергу Arduino апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та середовище розробки Processing/Wiring на мові програмування, що є спрощеною підмножиною C/C++. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері. Ардуїно може використовуватися для розробки інтерактивних систем, керованих різними датчиками та перемикачами. Такі системи, у свою чергу, можуть керувати роботою різних індикаторів, двигунів та інших пристроїв.Будь-яку плату Ардуїно можна зібрати вручну або купити готовий пристрій;середовище розробки для програмування такої плати має відкритий вихідний код і є повністю безкоштовним.

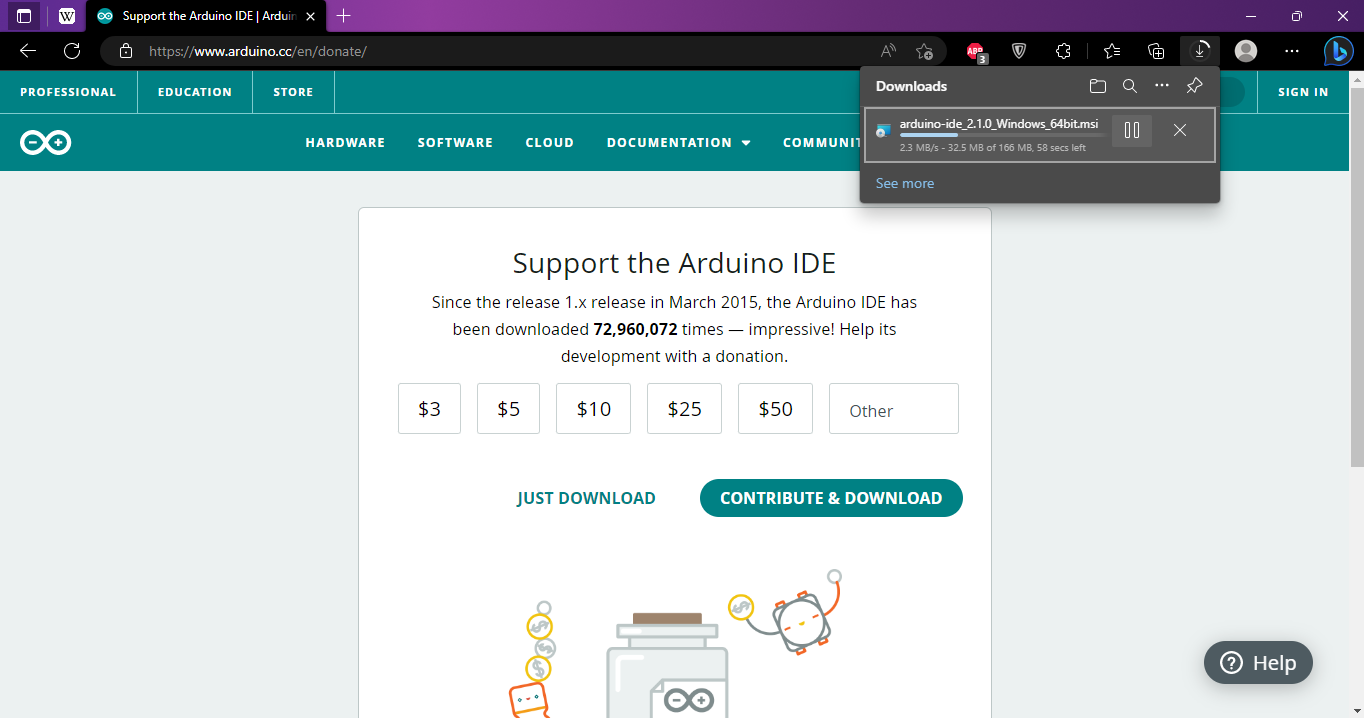
ArduinoIDE — інтегроване середовище розробки для Windows, MacOS та Linux, призначене для створення та завантаження програм на Arduino-сумісні плати, а також на плати інших виробників.

Підтримує мови *С* та *C++* з використанням спеціальних правил структурування коду.

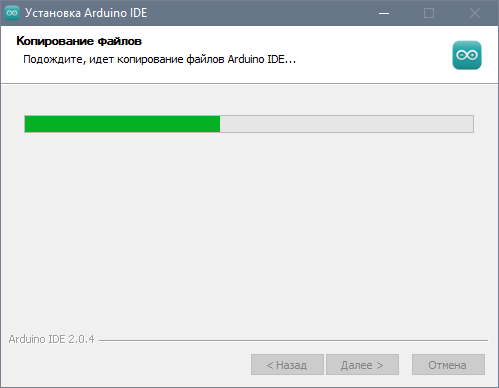
*ArduinoIDE* надає бібліотеку програмного забезпечення з проекту *Wiring*, яка надає безліч загальних процедур введення та виведення. Для написаного користувачем коду потрібні лише дві базові функції для запуску ескізу та основного циклу програми, які скомпільовані та пов'язані з заглушкою програми *main()* у виконувану циклічну програму з ланцюжком інструментів GNU, також включеною до дистрибутиву IDE.

**Завдання №2.** Встановити середовище розробки Arduino IDE.

Розпочнемо процес встановлення програмного середовища ArduinoIDE. Для виконання цього завдання виконаємо завантаження відповідного істаляційного пакету з офіційного сайту розробника даного програмного забезпечення.



Запустимо на виконання істаляційний пакет, після чого виконаємо усі необхідно налаштування для виконання подальшого процесу становлення даного ПО.



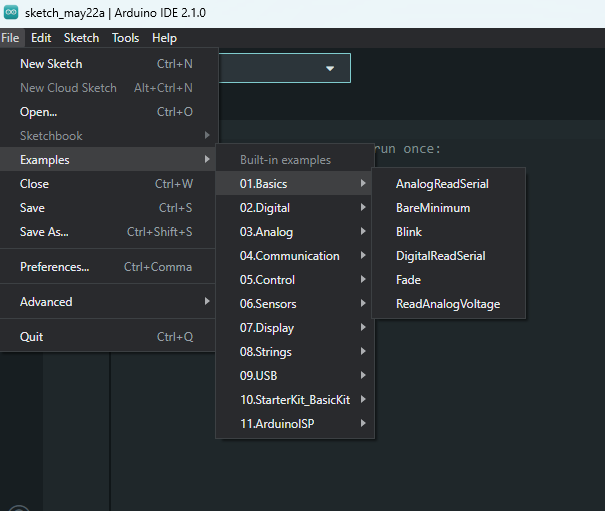
Після виконання процесу встановлення, програма являється повністю готовою до її експлуатації.



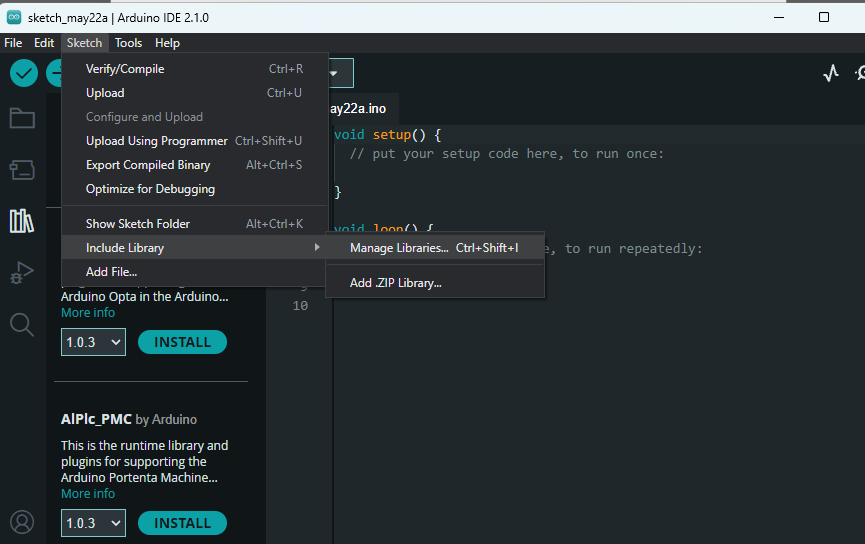
**Завдання №3.** Ознайомитися з роботою в середовищы розробки Arduino IDE та виконати приклади з лістингів 3.1 – 3.3. Розібратися з роботою прикладів.

Дане середовище програмування призначене для виконання розробки програмного забезпечення для платформи Arduino або для мікроконтролерів, що використовуються на базі цих платформ.

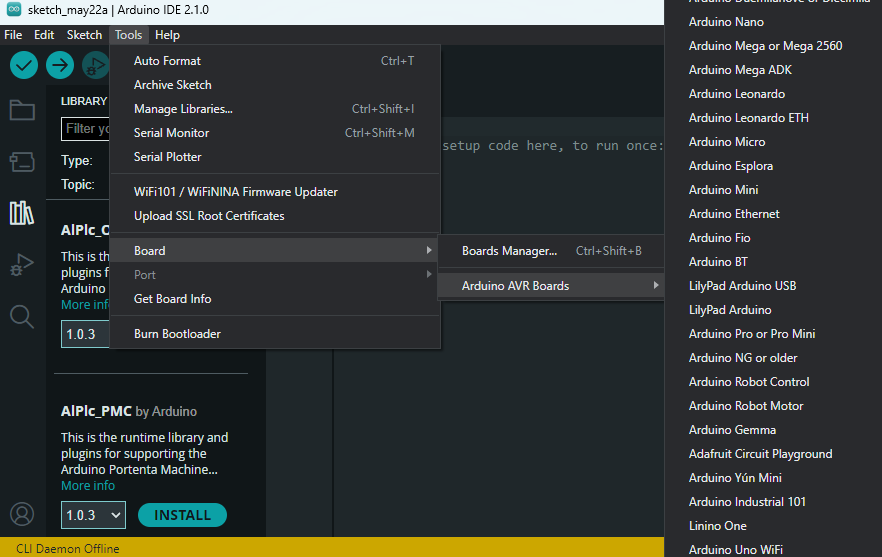
Середовище програмування ArduinoIDE працює на базі мови програмування C/C++ и скомпонований з бібліотекою AVR Libc. Для відкриття відповідних прокладів достатньо перейти до вкладки «File» та обрати пункт «Exanples» після чого у відповідному меню варто обрати, котрий з представлених прикладів цікавитиме саме вас.



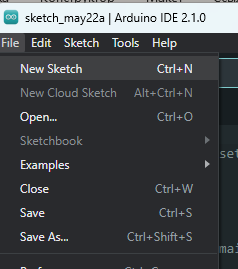
Є наявність підтримки стороніх бібліотек з наборами відповіних інструкцій.



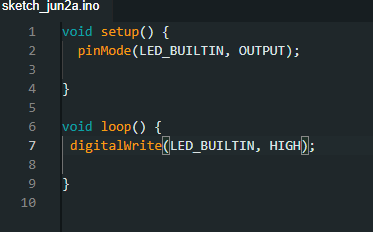
Перед початком роботи з програмним кодом впевнимося у відповіності встановлених параметрів роботи компілятора програмного середовища. Для цього перейдемо на вкладку «Tools».



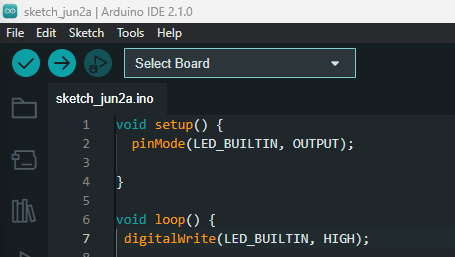
Оскільки я не маю Arduino UNO я лише можу імітувати процес роботи тому створимо новий скетч



В результаті було скетч. Лістинг програми зображений на рисунку нижче.



Виконаємо процес компілації скетч, та перевіримо роботу даного програмного коду на базі платформи Arduino.



**Завдання №4.** Згідно з заданим викладачем варіантом запишіть досконалу диз'юнктивну нормальну форму (ДДНФ) для логічної функції F чотирьох змінних, яка задана таблицею істинності (таблиця 3.1). Реалізувати програмно функцію логічного виразу. Вхідним параметром функції є однобайтова змінна. Введення виконати за допомогою програми монітора. Результат. Роботи функції відобразити на світлодіоді.

Відповідно до наданого викладачем завдання виконаємо пошук ДДНФ для логічної функції F чотирьох змінних, яка задана таблицею істинності №4.1.



*Таблиця №4.1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | F |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

– 0000

– 0010

-0100

-0110

x1x4 – 1001

x1x2x4 – 1101

x1x2x3 – 1110

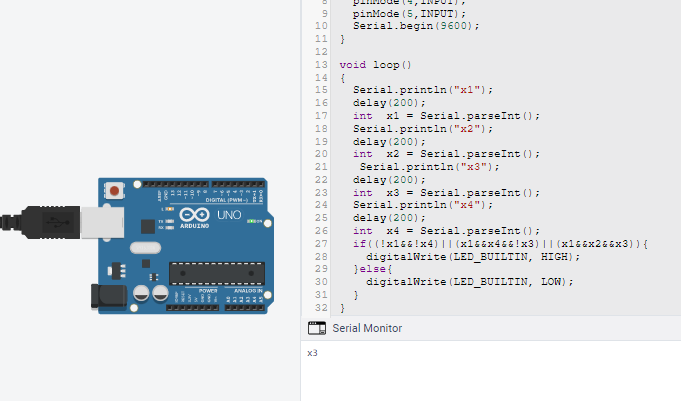
x1x2x3x4-1111

Використовуючи таблицю істинності для нашої функції виконаємо процес її мінімізації на основі метода карт Карно. Для цього виконаємо побудову карти та на її основі знайдемо ДДНФ для нашої функції.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***х1х2\х3х4*** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 0 | 1 |
| **01** | 1 | 0 | 0 | 1 |
| **11** | 0 | 1 | 1 | 1 |
| **10** | 0 | 1 | 0 | 0 |

Отримуємо наступну **мінімізовану функцію методом карт Карно**:

Таким чином, нами було отримано досконалу диз’юнктивну нормальну форму для нашої початкової функції. Перейдемо до процесу моделювання роботи даної функції на базі платформи Arduino. Для цього напишимо відповідну програмне реалізацію нашої функції, щоб виконати її моделювання. Лістинг програми зображено нижче.



Відкриваючи моніторинг послідовного порту та вводячи відповідні значення було отримано наступні результати перевірки роботи програми.

Лістинг коду:

// C++ code

//

void setup()

{

pinMode(LED\_BUILTIN, OUTPUT);

pinMode(2,INPUT);

pinMode(3,INPUT);

pinMode(4,INPUT);

pinMode(5,INPUT);

Serial.begin(9600);

}

void loop()

{

Serial.println("x1");

delay(200);

int x1 = Serial.parseInt();

Serial.println("x2");

delay(200);

int x2 = Serial.parseInt();

Serial.println("x3");

delay(200);

int x3 = Serial.parseInt();

Serial.println("x4");

delay(200);

int x4 = Serial.parseInt();

if((!x1&&!x4)||(x1&&x4&&!x3)||(x1&&x2&&x3)){

digitalWrite(LED\_BUILTIN, HIGH);

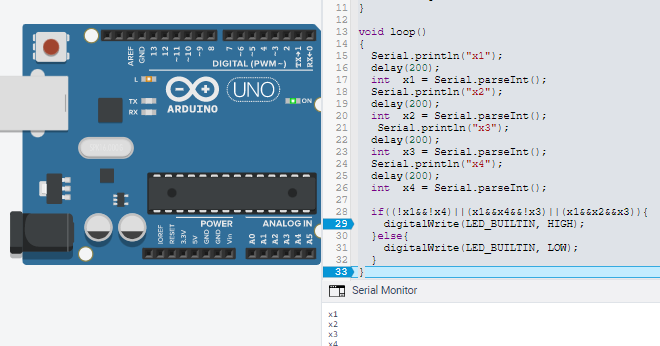
}else{

digitalWrite(LED\_BUILTIN, LOW);

}

}

Результат роботи при введені 1110:



Посилання на схему: <https://www.tinkercad.com/things/hLMCDjygvnT>

Виконавши тестування роботи побудованої нами ДДНФ, що було спрощена на основі карт Карно, помилок в її роботі виявлено не було. Побудована функція працює коректно.

**Завдання №5.** Детально пояснити схему багатофункціонального макету, схема якого зображена на рисунку №5.1, відповідно до завдання.

Почнімо виконання завдання. Для цього викладачем було надано відповідну електричну схему деякого пристрою, що зображена на рисунку №5.1.

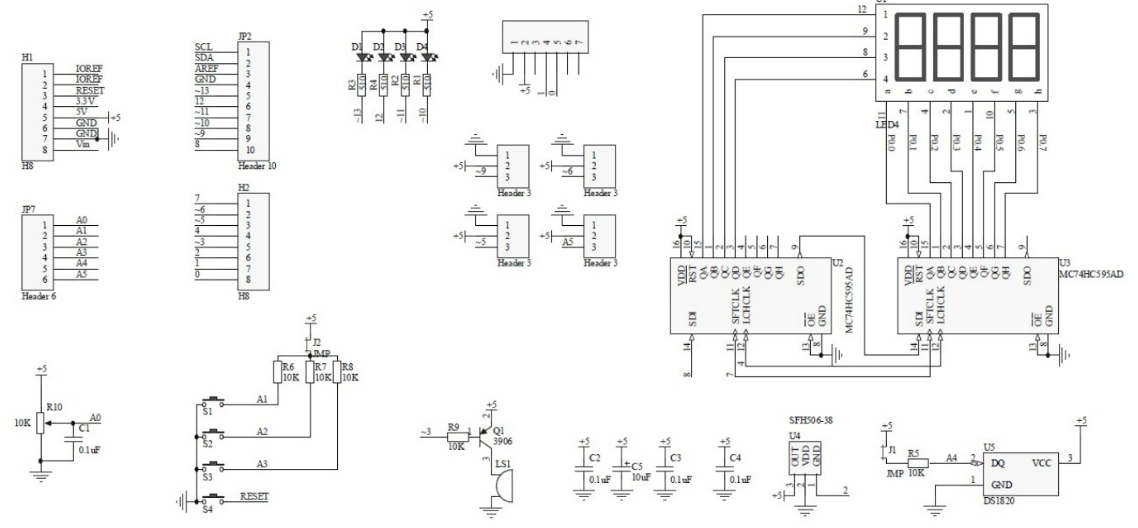
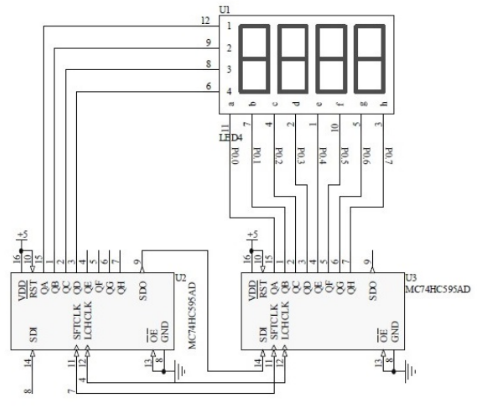


Рис. 5.2 – Схема багатофункціонального макету

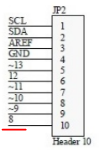
Використваши технічну документацію на мікросхему MC74HC595AD вдалося встановити, що це мікросхема восьмибітного регістра зсуву, що здатна приймати інформацію послідовному вигляді та перетворювати її у паралельний. Також вивід інформації можливий у послідовного вигляді.

Оглянемо частину схеми.

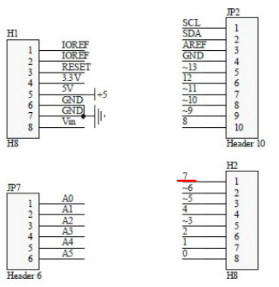


На 14 вхід мікросхеми U2 приходить послідовний код значення. Під впливом сигналів керування 11 вивід та 12 вивід в регістри виконується запис послідовної інформації. Частина бітів, застосовується для керування збіркою семисегментними індикаторам. Далі, з мікросхеми U2, з виводу 9 виконується вивід інформації на мікросхему U3, яка в свою чергу виконує керування значенням одного з чотирьох семисегментним індикатором.

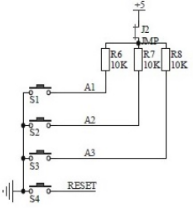
Відповідно до схеми цей послідовний сигнал буде надходити з JP2 а саме з сигналу 8.



Також було визначено, що керування мікросхемами U2 та U3 займається сигнал під номером 7, що також надходить зовні на одни з роз’ємів схеми.

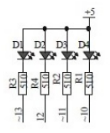


Розглянемо інші частини схеми. Оглянувши схему, було виявлено наявність перемикачів, схема яких зображена нижче на рисунку.



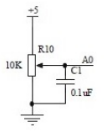
В нашому випадку кнопок на платі 4. Значення з 3 перемикачів позначено у вигляді сигналів A1, A2, A3.

Розглянемо іншу частину схеми, яка відповідає за індикацію наявності деяких сигналів.

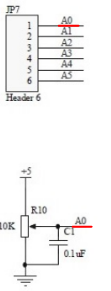
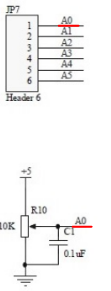


Світлодіоди D1, D2, D3 та D4. Світлодіоди підключені через навантаження, у вигляді резисторів. Керування ними відбувається на основі 4 сигналів. Дані сигнали 10, 11, 12 та 13 надходять з роз’єму JP2.

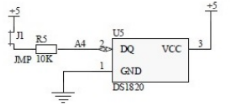
Оглянувши схему, було виявлено наявність потенціометра, який може бути застосований у вигляді елементу керування.



Значення з даного потенціометру виводиться у вигляді сигналу A0 на роз’єм JP7.



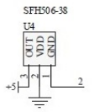
Перейдемо до огляду іншої частини схеми. Для прикладу оглянемо частину схеми з використанням комопонета під маркуванням DS1820.



Це цифровий термодатчик, який можна використовуватися як елемент керування.

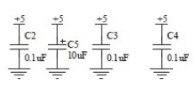
Через резистор виконається його підключення до деякої схеми, яка займається обробкою цифрового сигналу з цього датчика.

Було виявлено інфрачервоний датчкий, що представлений у вигляді інтегральної мікросхеми SFH506-38 на схемі.

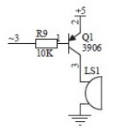


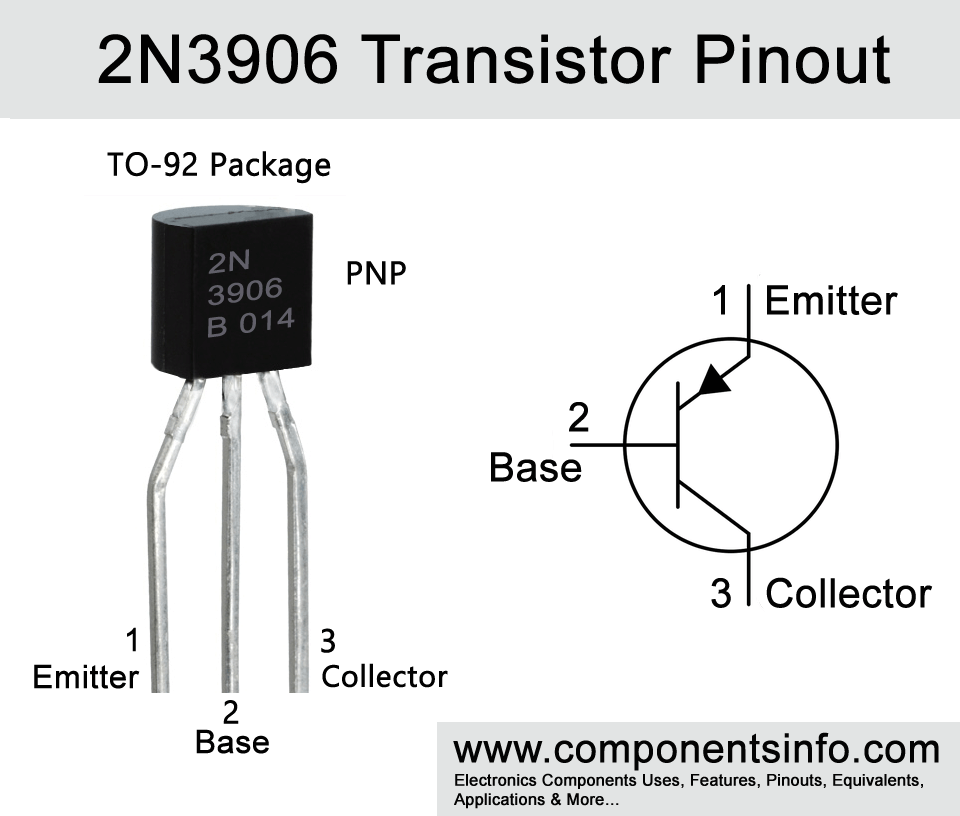
Цифровий сигнал з цього датчику подається на відповідний роз’єм на нашій схемі до H2.

Також, на схемі було виявлено конденсатори в колі живлення схеми. Конденсатори використовуються для фільтрації можливих завад.

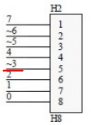


Було виявлено схему керування навантаженням через біполярний P-N-P транзистор 3906.





Керування транзистором в схемі виконується на основі знайомої нам ШИМ. Використовуючи різну сквіжність сигналу можемо виконувати контроль ступенем відкриття транзистора. Керування виконується на основі сигналу 3, що через резистор підлючений до бази транзистора. Сам сигнал керування надходить з відповідного роз’єму на схемі.



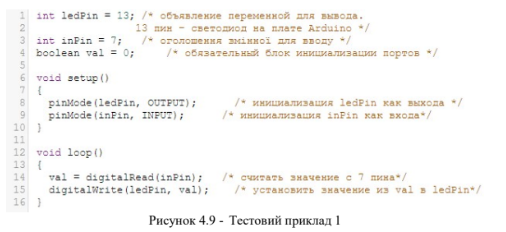
**Висновки:** В ході виконання лабораторної роботи було проведено ознайомлення з роботою програмного середовища Arduino IDE, а також безпосередньо платформою Arduino. Під час виконання роботи було проведено процес програмування мікроконтролера та тествання працеспроможності деяких програмних застосунків. Закріпив здобуті навички та вміння, щодо роботи з даною платформою, а також основними принципами програмування платформи.

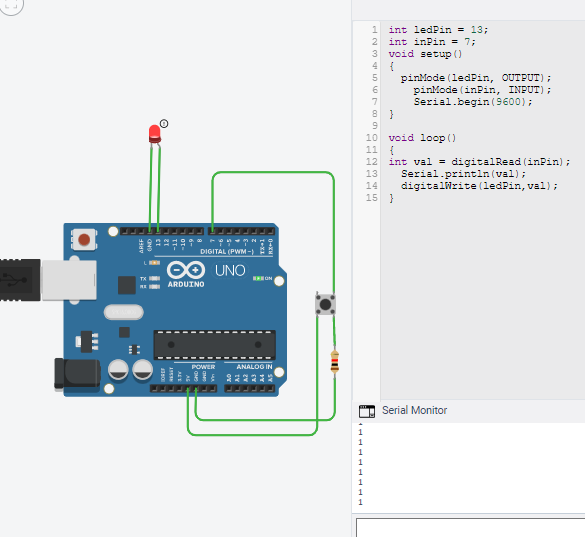
**ЛАБОРАТОРНА РОБОТА №4**

**Мета заняття**: ознайомитися з принципами введення/виведення цифрових сигналів. Вивчити роботу дискретних електронних компонентів (світлодіод, пьезовипромінювач, кнопка тощо) та використання плати Arduino. Ознайомитись з середовищем розробки ArduinoIDE.

**Початок роботи**

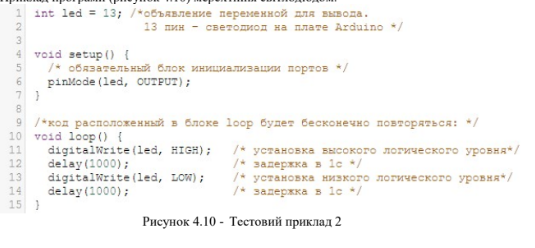
**Завдання №1**. Ввести програму на рисунку 4.9. Переконатися в її працездатності. В якості кнопки використати кнопку S1 макету.

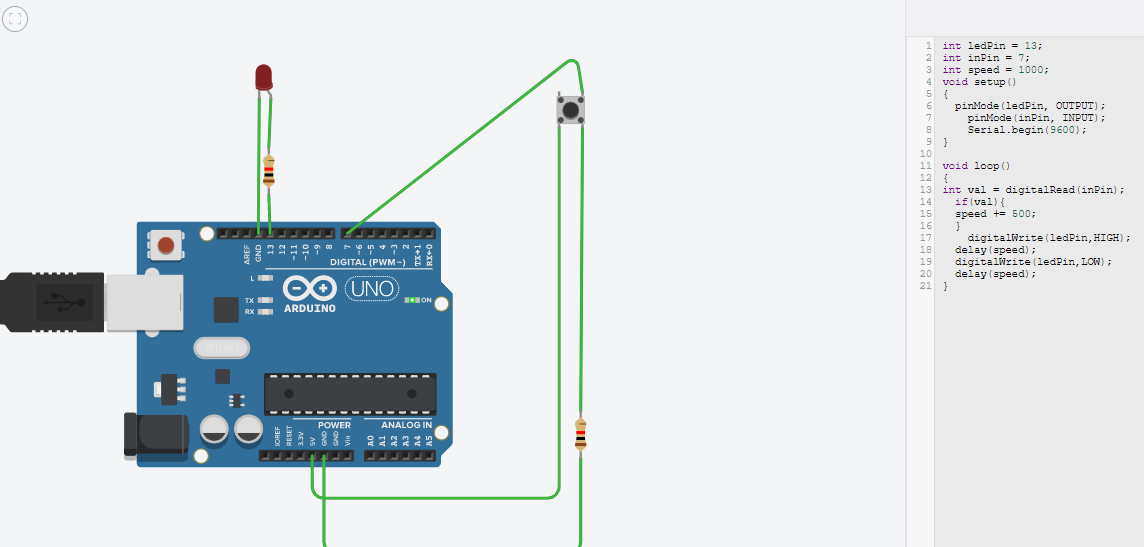




Посилання на схему: <https://www.tinkercad.com/things/gF5qAMqNWMg>

**Завдання №2.** Ввести програму на рисунку 4.10. Переконатися в її працездатності. Змінити швидкість миготіння.





**Посилання на схему:** [**https://www.tinkercad.com/things/j9JFqYrCMY5**](https://www.tinkercad.com/things/j9JFqYrCMY5)

**Висновки:** В ході виконання лабораторної роботи було проведено ознайомлення з принципами ввову та виведення цифрових сигналів на базі платформи Arduino UNO. Під час виконання роботи виконаня ряд відповідних завдань, на основі яких покращив власні навички та вміння побудови принципових електричних схем, програмування платформи ардуіно, а також проектування систем на базі логічних функцій та зображання їх у вигляді графа. Закріпив здобуті навички та вміння.

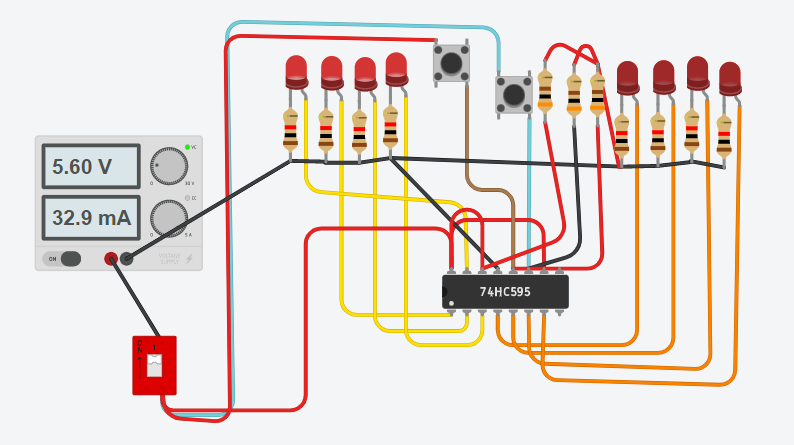
**ЛАБОРАТОРНА РОБОТА №5**

**Мета заняття:** ознайомитися з принципами введення/виведення цифрових сигналів. Вивчити роботу дискретних електронних компонентів (світлодіод, пьезовипромінювач, кнопка тощо) та використання плати Arduino. Ознайомитись з середовищем розробки ArduinoIDE

**Початок роботи**

**Завдання 1: Зібрати схему (рисунок 5.8) та вивчити роботу регістру зсуву 74HC595**

Розпочнемо виконання лабораторної роботи. Для початку, відповідно до отриманого завдання, проведемо ознайомлення з принципом роботи мікросхеми 74HC595.Зазначена вище мікросхема являється простим восьмирозрядним регістром зсуву. Дана схема займається зберігання та виконання зсуву бітів даних, котрів в межах нашої роботи будуть застосовуватися для керування семисигментним індикатором. Для керування схемою використовуються три основні контакти, а саме контакт «ввод», контакт «вивод», контакт «зсув».



**Посилання на проект:**

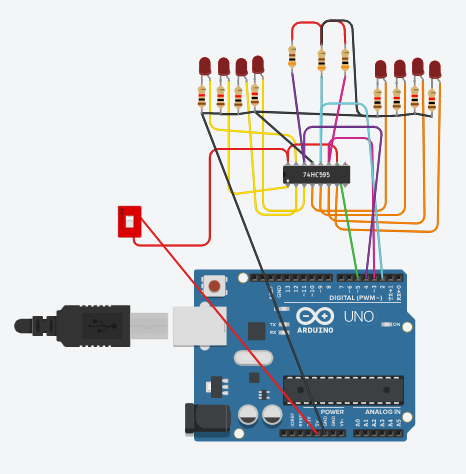
[**https://www.tinkercad.com/things/ldAuBhGpDyl**](https://www.tinkercad.com/things/ldAuBhGpDyl)

**Завдання №2**. Замінити управління перемикачами на управління за допомогою Ардуіно. Реалізувати “Мерехтливі вогні” у відповідності варіанту (див. таблицю 5.1).





Виконаємо модифікацію проекту. Для цього додамо платформу Arduino UNO, на основі мікроконтролера якої будемо виконувати керування схеми для побудови мерехтливих вогнів. В результаті проведених модифікацій, схема матиме наступний вигляд.



**Код Arduino**

// C++ code

int level = 3;

int output = 2;

int input = 4;

int delTime = 300;

int del = 5;

int step = 1;

/\*відкинув зайві нулі\*/

int a[8] = {1,0,0,0,1};

int a2[8] = {1,0,1,1,1,0,1,1};

int a3[8] = {1,1,0,1,1,1,0,1};

int a4[8] = {0,1,1,0,0,1,1};

void setup()

{

  pinMode(LED\_BUILTIN, OUTPUT);

  pinMode(3,OUTPUT);

  pinMode(4,OUTPUT);

  pinMode(2,OUTPUT);

    pinMode(5,OUTPUT);

}

void loop()

{

  if(step==5){

  step =1;

  }

  if(step){

    for(int i=(sizeof(a)/sizeof(a[0]))-1;0<=i;i--){

    int d = a[i];

    digitalWrite(input, a[i]);

    digitalWrite(level,1);

    digitalWrite(level,0);

  }

 }

    if(step==2){

     for(int i=(sizeof(a2)/sizeof(a2[0]))-1;0<=i;i--){

    digitalWrite(input, a2[i]);

    digitalWrite(level,1);

    digitalWrite(level,0);

  }

    }

    if(step==3){

   for(int i=(sizeof(a3)/sizeof(a3[0]))-1;0<=i;i--){

    digitalWrite(input, a3[i]);

    digitalWrite(level,1);

    digitalWrite(level,0);

  }

    }

  if(step==4){

     for(int i=(sizeof(a4)/sizeof(a4[0]))-1;0<=i;i--){

    digitalWrite(input, a4[i]);

    digitalWrite(level,1);

    digitalWrite(level,0);

  }

  }

  digitalWrite(output, 1);

  digitalWrite(output, 0);

  step++;

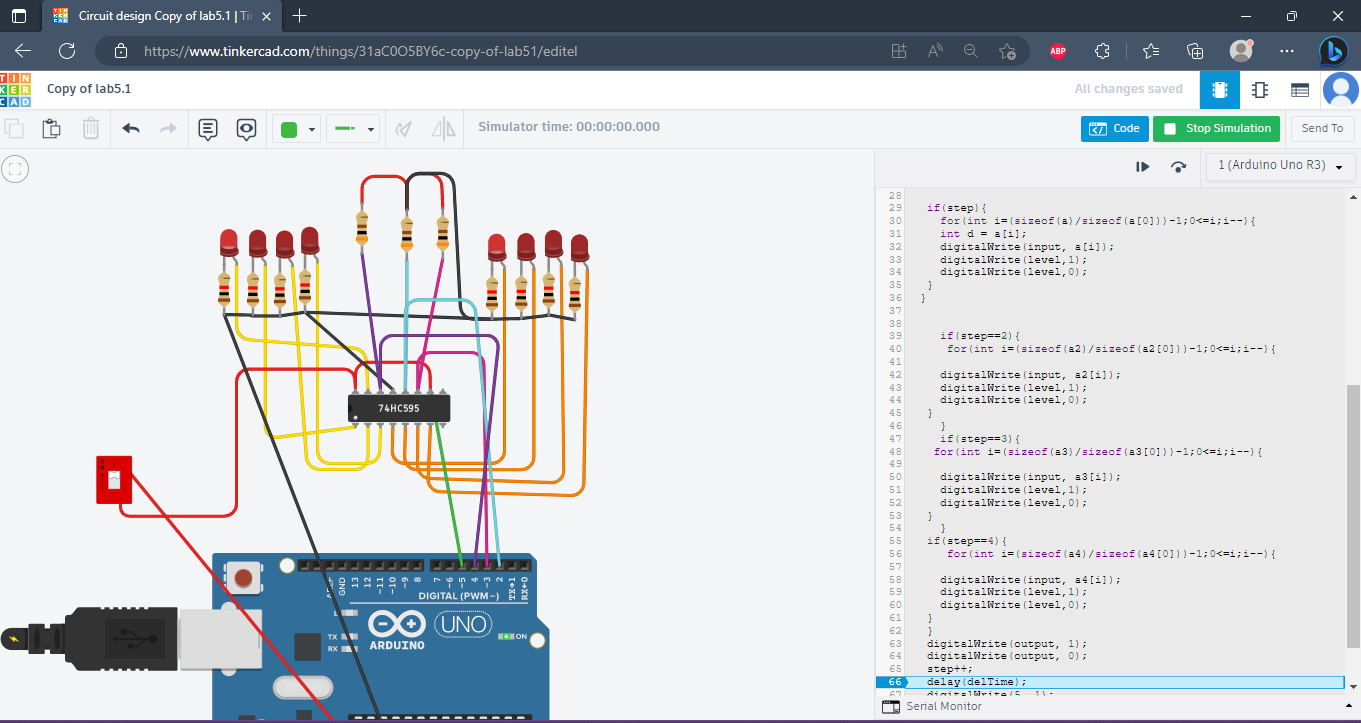
  delay(delTime);

  digitalWrite(5, 1);

  digitalWrite(5, 0);

}

**Результат роботи:**

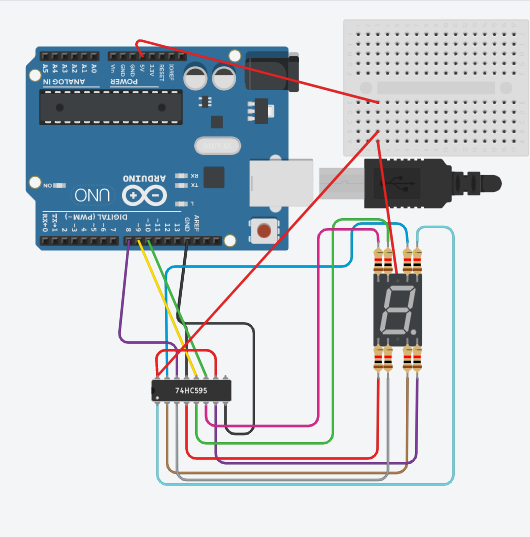


**Посилання на проект:**

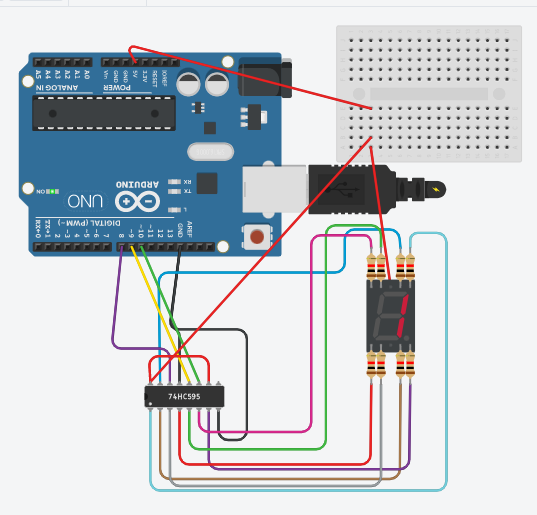
[**https://www.tinkercad.com/things/31aC0O5BY6c**](https://www.tinkercad.com/things/31aC0O5BY6c)

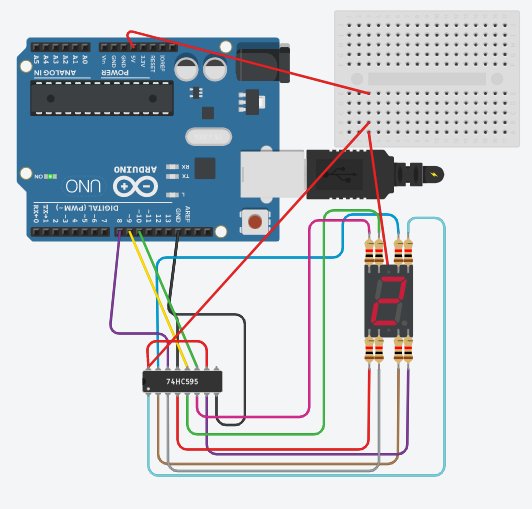
**Завдання №3.** Замінити світлодіоди на семисегментний індикатор та реалізувати функцію відображення цифр.

Виконаємо модифікацію схеми, додамо до неї семисигментний індикатор та напишемо функцію керування ним, для відораження числових символів. Схема матиме наступний вигляд.



**Результат роботи програми**





**Код:**

// C++ code

int level = 10;

int output = 9;

int input = 8;

int delTime = 300;

int del = 5;

int zero[8] = {0,0,1,1,1,1,1,1};

int one[8] = {0,0,0,0,0,1,1,0};

int two[8] = {0,1,0,1,1,0,1,1};

int three[8] = {0,1,0,0,1,1,1,1};

int four[8] = {0,1,1,0,0,1,1,0};

int five[8] = {0,1,1,0,1,1,0,1};

int six[8] = {0,1,1,1,1,1,0,1};

int seven[8] = {0,0,0,0,0,1,1,1};

int eight[8] = {0,1,1,1,1,1,1,1};

int nine[8] = {0,1,1,0,1,1,1,1};

int step = 1;

/\*відкинув зайві нулі\*/

void setup()

{

  pinMode(LED\_BUILTIN, OUTPUT);

  pinMode(level,OUTPUT);

  pinMode(output,OUTPUT);

  pinMode(input,OUTPUT);

  pinMode(del,OUTPUT);

}

void loop()

{

 setNumber(one);

 delay(1000);

  setNumber(two);

  delay(1000);

  setNumber(three);

  delay(1000);

  setNumber(four);

  delay(1000);

  setNumber(five);

  delay(1000);

  setNumber(six);

  delay(1000);

  setNumber(seven);

   delay(1000);

  setNumber(eight);

     delay(1000);

  setNumber(nine);

}

void setNumber(int a[]){

  for(int i=0;i<8;i++){

  digitalWrite(input,!a[i]);

  digitalWrite(level, 1);

   digitalWrite(level, 0);

}

 digitalWrite(output, 1);

 digitalWrite(output, 0);

}

**Посилання на схему:**

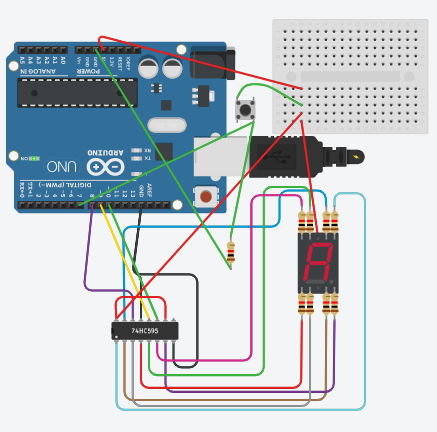
[**https://www.tinkercad.com/things/eSDSOOZy8hn**](https://www.tinkercad.com/things/eSDSOOZy8hn)

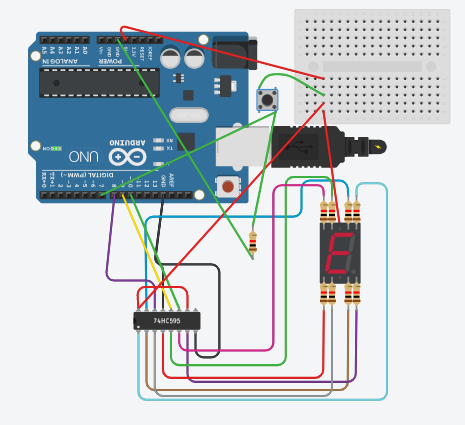
**Завдання 4.** Розробити програму дешифрування літерів A, B, C, D, E, F азбуки Морзе та відображення їх на семисегментному індикаторі. Телеграфний ключ реалізувати за допомогою кнопки. Тривалість: однієї точки – 100 мс; тире – 300 мс; паузи між елементами азбуки Морзе – 100 мс; паузи між літерами – більше 300 мс. Допустима похибка затримок – 10 % (відображати символ “П” на індикаторі, якщо похибка затримки більша). Відображати символ “0”, якщо символ не розпізнано.

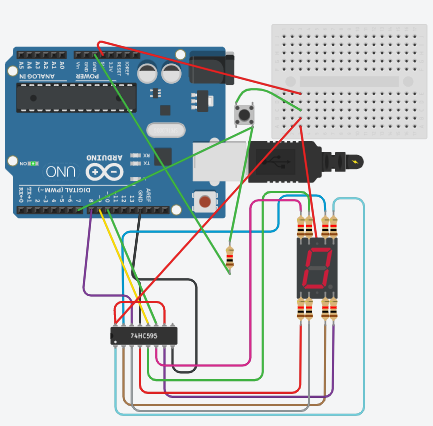
Реалізуємо роботи програми дешифрування літер згідно азбуки Морезе. В нашому випадку, реалізуємо програмний код для дешифрування зазначених символів. Для цього виконаємо модифікацію проекту з семисегментним індикатором додавши до нього кнопку. Результат зображений на рисунку нижче.

Посилання на схему:

<https://www.tinkercad.com/things/1BVeCztGtTE>







Лістинг коду:

// Значення символів Морзе для літер A, B, C, D, E, F

const char\* morseCode[] = {".-", "-...", "-.-.", "-..", ".", "..-."};

// Змінні для дешифрування символу Морзе

String currentSymbol = "";

int decodedCharacter = 0;

int a[8] = {0,1,1,1,0,1,1,1};

int b[8] = {0,1,1,1,1,1,0,0};

int c[8] = {0,0,1,1,1,0,0,1};

int d[8] = {0,1,0,1,1,1,1,0};

int e[8] = {0,1,1,1,1,0,0,1};

int f[8] = {0,1,1,1,0,0,0,1};

int zero[8] = {0,0,1,1,1,1,1,1};

// C++ code

int level = 10;

int output = 9;

int input = 8;

int inputButton = 7;

const int dotDuration = 100;  // 100 мс

const int dashDuration = 300;  // 300 мс

const int elementPause = 100;  // 100 мс

const int letterPause = 300;  // 300 мс

const int delayTolerance = 10;  // 10% похибка затримки

int buttonState = LOW;

int prevButtonState = LOW;

void setup()

{

  pinMode(level,OUTPUT);

  pinMode(output,OUTPUT);

  pinMode(input,OUTPUT);

  pinMode(inputButton,INPUT);

  Serial.begin(9600);

}

void loop()

{

  buttonState = digitalRead(inputButton);

 if (buttonState == HIGH && prevButtonState == LOW) {

    decodeMorse();

  }

 int inputSignal  = digitalRead(inputButton);

}

void decodeMorse() {

  while (buttonState == HIGH) {

    // Отримання тривалості сигналу

    int signalDuration = getSignalDuration();

    // Якщо тривалість сигналу менша за дотирання, вважаємо його крапкою

    if (signalDuration <= dotDuration + dotDuration \* delayTolerance / 100) {

      currentSymbol += ".";

    }

    // Якщо тривалість сигналу більша за дотирання, вважаємо його тире

    else if (signalDuration >= dashDuration + dashDuration \* delayTolerance / 100) {

      currentSymbol += "-";

    }

    // Якщо тривалість сигналу в межах паузи між елементами Морзе,

    // дешифруємо поточний символ і очищуємо його значення

    else if (signalDuration >= elementPause - elementPause \* delayTolerance / 100) {

      decodeSymbol();

    }

    // Затримка перед наступним вимірюванням тривалості сигналу

    delay(10);

    // Оновлення стану кнопки

    buttonState = digitalRead(inputButton);

  }

   Serial.println(currentSymbol);

  decodeSymbol();

  // Відображення розшифрованого символу на семисегментному індикаторі

  printLetter();

  delay(letterPause);

  // Очищення значень для наступного дешифрування

  currentSymbol = "";

  decodedCharacter = 0;

  }

void printLetter(){

  if(decodedCharacter==-1){

  set(zero);

  }

  if(decodedCharacter==0){

   set(a);

  }

   if(decodedCharacter==1){

   set(b);

  }

   if(decodedCharacter==2){

   set(c);

  }

   if(decodedCharacter==3){

   set(d);

  }

  if(decodedCharacter==4){

   set(e);

  }

   if(decodedCharacter==5){

   set(f);

  }

}

void decodeSymbol() {

  if (currentSymbol == "") {

    return;

  }

  // Дешифрування символу Морзе

  for (int i = 0; i < sizeof(morseCode) / sizeof(morseCode[0]); i++) {

    if (currentSymbol==morseCode[i]) {

      decodedCharacter = i;

    break;

    }

    else{

    decodedCharacter = -1;

    }

  }

}

int getSignalDuration() {

  int duration = 0;

  // Вимірювання тривалості сигналу, поки кнопка утримується

  while (buttonState == HIGH) {

    delay(1);

    duration++;

    buttonState = digitalRead(inputButton);

  }

  return duration;

}

void set(int a[]){

  for(int i=0;i<8;i++){

  digitalWrite(input,!a[i]);

  digitalWrite(level, 1);

   digitalWrite(level, 0);

}

 digitalWrite(output, 1);

 digitalWrite(output, 0);

}

**Висновки:** В ході виконання лабораторної роботи було проведено роботу з восьмирозрядним регістром зсуву 74HC595 та семисегментним індикатором, на базі яких було проведено ряд досліджень та тестувань. В результаті виконання роботи було проведено закріпленно знання та навички основних принципів керування семисигментним індикатором, а також покращено знання про роботу з регістрами зсуву. Також, в ході виконання роботи було проведено процес побудови різних схем керування, на основі яких було продемонстровано основні принципи керування, а також реалізовано декілька програм на базі мікроконтролера Arduino для керування керування роботою семисигментного індикатора на базі регістра зсуву.