Міністерство освіти і науки України  
Національний університет «Львівська політехніка»



Записка до курсової роботи

**Виконав**:

ст. гр. ІР-22

Ганик І.О.

**Прийняв: доцент КСА**

Павельчак А.Г

Львів – 2024

**Зміст**

* Мета та завдання
* Вступ
* Аналіз питання
* Структура та функціональна схема проекту з загальним описом складових
* Проілюстровані детальні описи окремих складових
  1. апаратна частина
  2. back-end
  3. БД
  4. Web додаток
* Загальна ілюстрація працездатності проекту
* Висновки

**Мета**

Мета даної курсової роботи , це закріпити знання здобуті протягом навчання на спеціальності “Інтернет речей”. Дана робота охоплює багато предметів які викладалися за період навчання.

До таких предметів входить:

* Алгоритмізація та програмування
* Схемотехніка
* Бази даних
* Веб програмування
* Мікроконтролери

Саме ці предмети дали хороші знання для того, щоб виконати дану курсову роботу.

**Завдання на курсовий проект**

Проект має виконуватись на фізичному залізі(у разі його відсутності у програмі **Proteus**).

Основні складові курсового проекту:

* робоча симуляція
* Back-end частина
* Веб додаток
* База даних

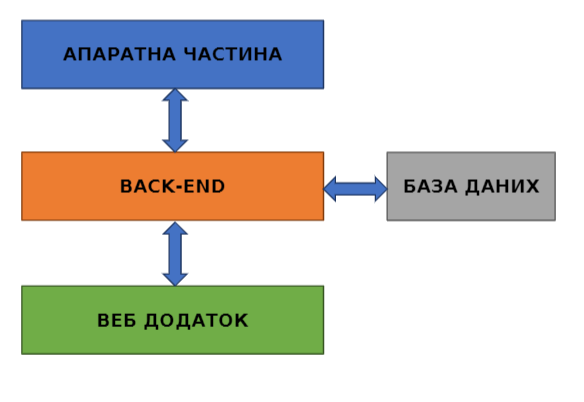


Рис. 1) Ілюстрація яка показує комунікацію між

різними складовими курсової роботи

Слід реалізувати передачу даних між апаратною частиною та одною з найголовніших back-end частиною. Натомість остання, отримує дані від веб додатку (який виступає в ролі графічного середовища для користувача) та пересилає їх на симуляцію. Та паралельно зберігає необхідну інформацію в базі даних.

**Вступ**

Ідея мого курсового проекту полягає у віддаленому та автоматизованому контролюванні імітованої гірлянди. На симуляції поставленні в ряд led-діоди які контролюються користувачем через web-аплікацію. Користувачу наданий фунціонал для контролювання гірляндою наживо та можливість задавати алгоритми для подальшого їх використання з можливістю повторного виклику

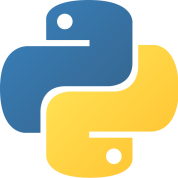
**Аналіз питання**

Нижче наведений короткий опис технічних рішень для розробки курсової роботи.

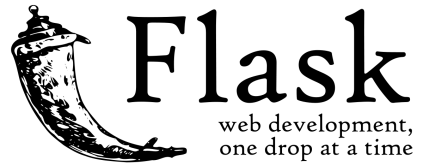
1. Симуляція зроблена у програмі **Proteus**



1. Back-end частина реалізована на мові програмування **Python**



Використовую такі бібліотеки: time, serial, flask, SQLAlchemy, Marshmallow.

Основні з них це є :

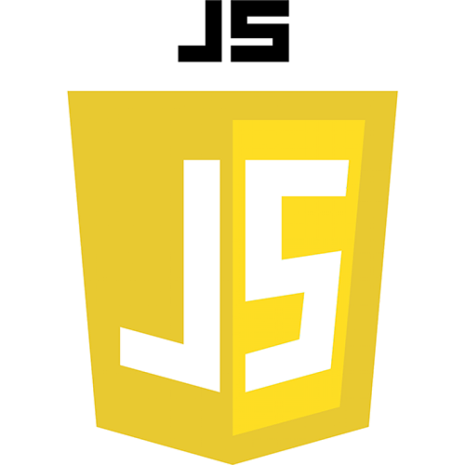
* **serial** яка слугує для того, щоб під’єднуватися до симуляції через віртуальні COM порти.
* За допомогою **Flask** приймаю дані з web частини.
* Та **SQLAlchemy**, яка надає зручний функціонал для роботи з базою даних **MySQL**.

Короткий опис:

* **Python -** інтерпретована мова, що дозволяє заощадити значну кількість часу, що зазвичай витрачається на компіляцію. Інтерпретатор можна використовувати інтерактивно, що дозволяє експериментувати з можливостями мови, писати шаблони програм або тестувати функції при розробці “знизу-вверх”. Він також зручний як настільний калькулятор. Python дозволяє писати дуже компактні й зручні для читання програми
* **Proteus** -пакет програм для автоматизованого проєктування електронних схем.
* **Flask** - мікрофреймворк для вебдодатків, створений з використанням Python.
* **MySQL** - вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних з відкритим кодом була створена як альтернатива комерційним системам

1. Технології веб частини включають в себе **HTML**, **СSS** та **JS.**

* **HTML** - стандартизована мова розмітки документів для перегляду веб-сторінок у браузері.
* **CSS** - це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду.
* **Java Script -** це мова програмування, що дозволяє зробити web -сторінку інтерактивною та передати данні вводу користувача до back-end частни.

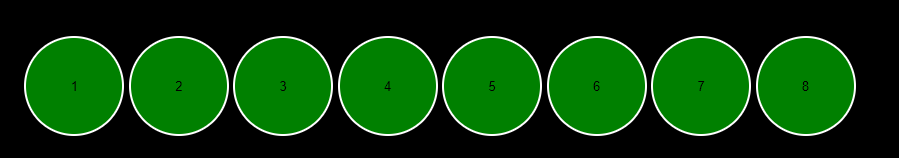


**Структура та функціональна схема проекту з загальним описом складових**

Структура моєї курсової базується відповідно до схеми зображеної на рис.1.

Web частина, якою користувач керує симуляцією.

* Вона складається з ряду кнопок які відповідають світло діодам на симуляції.



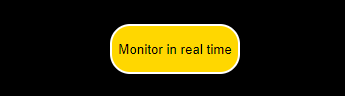
Ці кнопки є неактивні поки ви не обрали відповідну вказівку, контролювання в реальному часі чи створення алгоритму.

* Наступними йдуть кнопки які вмикають/вимикають усі світло діоди.



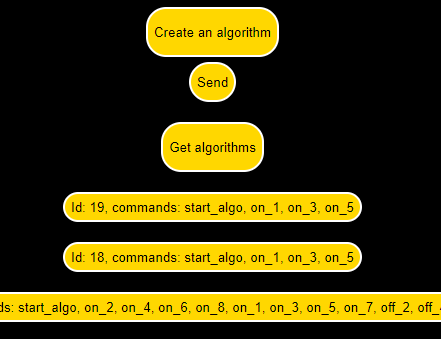
При їхньому активуванні, ви можете керувати усіма світло діодами без вибраних вказівок та у режимі реального часу або створення алгоритму

* Кнопка яка активує керування наживо.



Після її активування вона стає виділена(створення алгоритму блокується), та надається змога змінювати стан світло діодів в реальному часі.

* Створення алгоритмів з можливістю їх повторного виклику.



При активуванні “Create an algorithm”, ви обираєте послідовність світло діодів(при цьому кнопки “onAll/offal” також можуть входити в створену послідовність) та кнопкою “Send” відправити його на виконання.

Після чого кнопкою “Get algorithm” ви можете отримати створенні вами раніше алгоритми, які зберігаються в базі даних. Та при активуванні кнопок з id та командами викликати заданий алгоритм заново.

Функціонал back end частини

Приймає відповідні команди за допомогою протоколу HTTP.

* **HTTP** — протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від **H**yper**T**ext **T**ransfer **P**rotocol, протокол передачі гіпертекстових документів. Я використовую його на основі архітектури REST API.
* **REST** - REpresentational State Transfer. Це популярний архітектурний підхід для створення API у сучасному світі.

Відмінною рисою REST є те, що API-інтерфейси REST фокусуються на ресурсах та способах доступу до ресурсів. Ресурси, зазвичай, є різними типами інформації. Ви отримуєте доступ до ресурсів через URL (Uniform Resource Locators), так як перехід до URL-адреси у вашому браузері дозволяє підключитися до інформаційного ресурсу. URL-адреси супроводжуються методом, який вказує на те, як ви хочете взаємодіяти з ресурсом.

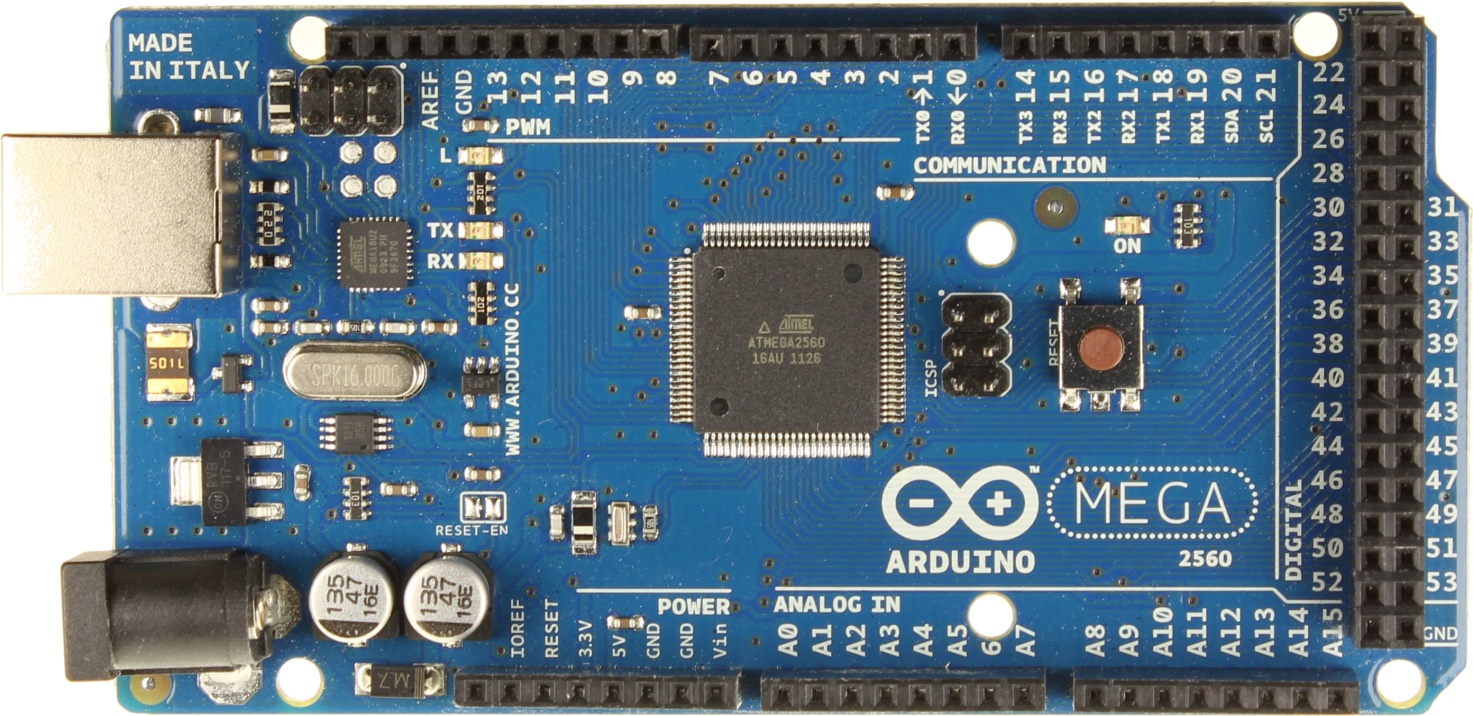
Загальні методи включають GET (читання), POST (створення), PUT (оновлення) та DELETE (видалення). Кінцева точка зазвичай включає параметри запиту, які визначають докладнішу інформацію про представлення ресурсу, який потрібно побачити.

Саме back end отримує вказівки з клієнтської аплікації та налаштовує виконання на симуляції. У пункті нижче – “Детальний опис back end частини” розписано про те як працює кожен блок коду.

**Проілюстровані детальні описи окремих складових**

**Детальний опис компонентів симуляції**

* + 1. Мікроконтролер Atmega2560



Arduino Mega побудована на мікроконтролері ATmega2560 (технічний опис). Плата має 54 цифрових входів/виходів (14 з яких можуть використовуватися як виходи ШІМ), 16 аналогових входів,4 послідовних порти UART, кварцовий генератор 16 МГц, USB коннектор, роз'єм живлення, роз'єм ICSP і кнопка перезавантаження. Для роботи необхідно підключити платформу до комп'ютера за допомогою кабелю USB або подати живлення за допомогою адаптера AC/DC, або акумуляторною батареєю. Arduino Mega 2560 сумісна з усіма платами розширення, розробленими для платформ Uno або Duemilanove.

Короткі характеристики:

* Робоча напруга 5В
* Вхідна напруга (рекомендована) 7-12В
* Вхідна напруга (гранична) 6-20 В
* Цифрові Входи/Виходи 54 (14 з яких можуть працювати також як виходи ШІМ)
* Аналогові входи 16 Постійний струм через вхід/вихід 40 mA
* Постійний струм для виведення 3.3 50 mA
* Флеш-пам'ять 256 KB (з яких 8 КВ використовуються для завантажувача) ОЗУ 8 KB
* Енергонезалежна пам'ять 4 KB
* Тактова частота 16 Мгц.

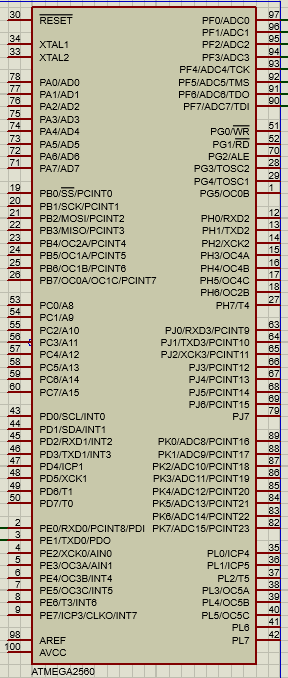
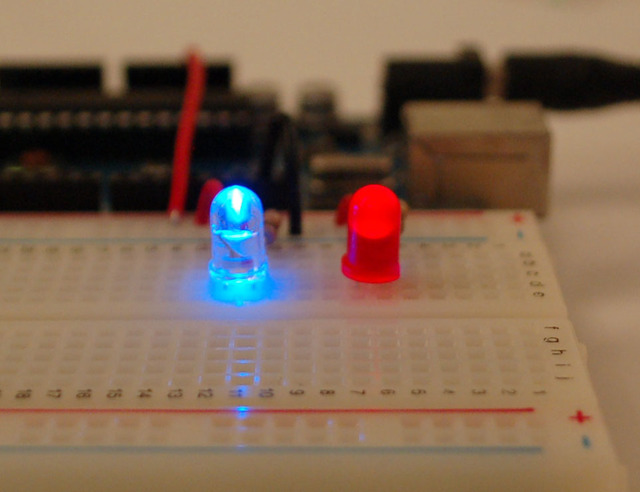


Рис. 2) Скрін мікроконтролера у стимуляційному пакеті **Proteus**

* + 1. Світлодіоди(8 штук)



Світлодіод - напівпровідниковий пристрій, що випромінює некогерентне світло, при пропусканні через нього електричного струму (ефект, відомий як електролюмінесценція). Випромінюване світло традиційних світлодіодів лежить у вузькій ділянці спектра, а його колір залежить від хімічного складу використаного у світлодіоді напівпровідника. Сучасні світлодіоди можуть випромінювати світло від інфрачервоної ділянки спектра до близької до ультрафіолету . Існують методи розширення смуги випромінювання і створення білих світлодіодів. На відміну від ламп розжарювання, які випромінюють світловий потік широкого спектра, рівномірно у всіх напрямках, звичайні світлодіоди випромінюють світло певної довжини хвилі і в певному напрямі.

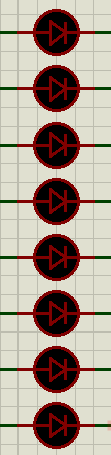
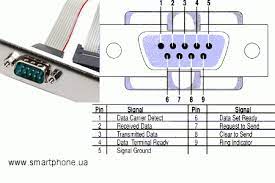


Рис.3) Скрін світло діодів на симуляції

* + 1. COM порт.

COM-порт є послідовним двонаправленим інтерфейсом, призначення якого полягає в обміні бітовою інформацією. Цей порт називається послідовним, оскільки забезпечує передачу інформації біт за битом. Раніше COM порти використовували для з'єднання комп'ютера із зовнішніми пристроями – супутниковими ресиверами, джерелами безперебійного живлення, приладами систем безпеки.



Сьогодні широко використовується емуляція порту COM. Практично кожен мобільний телефон емулює в собі модем і класичний СОМ порт, щоб забезпечити доступ до мережі Інтернет. Для фізичного підключення до комп'ютера використовується Bluetooth або USB.  
Інтерфейс COM порту RS-232 часто зустрічався в настільних комп'ютерах до появи досконаліших FireWare та USB. Цей інтерфейс підтримує два способи управління обміну даними: апаратний (за допомогою сигналів CTS та RTS), а також програмний метод.

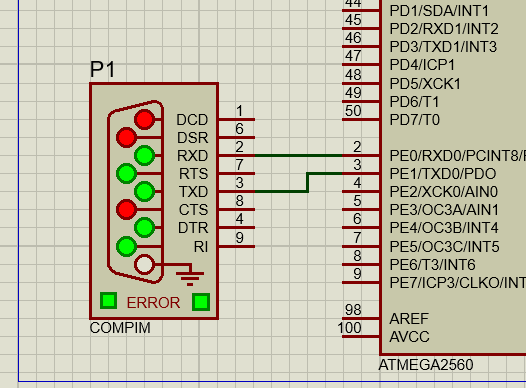
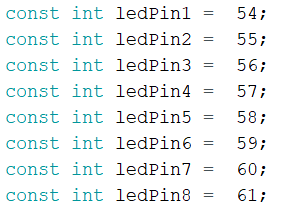


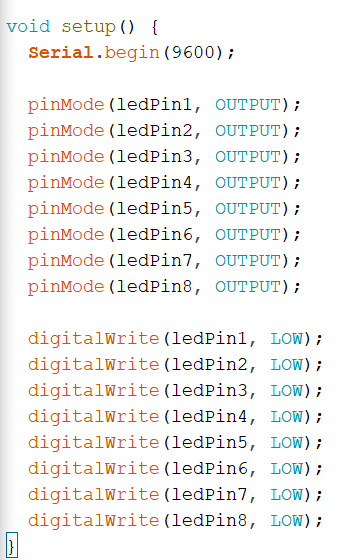
Рис. 4) Скрін компорту на симуляції

**Опис програми для контролю світло діодами на фізичному рівні.**

* Змінні які відповідають за піти на Atmega2560

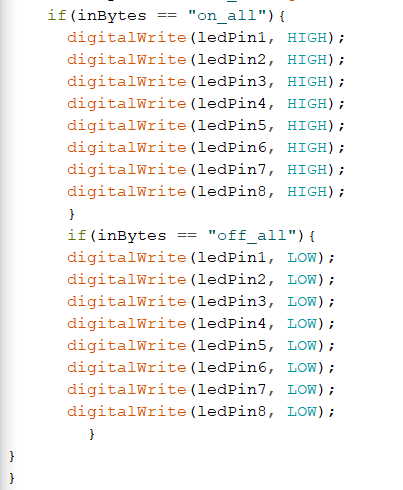
****

* Функція setup() яка виконується одразу при запуску симуляції. Спочатку налаштовується serial. Після чого встановлюються піни які були оголошенні у попередньому пункті. Та подання на піни сигналу LOW, щоб при запуску симуляції вони завжди були вимкненні.



* Функція loop() - працює у постійному циклі. У змінну inBytes передаються команди які надходять з back end частини і при спів падінні умов виконується певний код. Команди які приймає симуляція – on\_(ідентифікатор світло діода) та off\_( ідентифікатор світло діода).

Наприклад on\_1, off\_3 і т.д. Також є дві додаткові команди які вмикають/вимикають всі світло діоди – on\_all/off\_all.





Дана програма написана на мові програмування **С**.

**C** — універсальна, процедурна, імперативна мова програмування загального призначення, розроблена у 1972 році Деннісом Рітчі у Bell Telephone Laboratories з метою написання нею операційної системи UNIX.

Середовище яким я користувався – **Arduino IDE 1.18.19**.

**Arduino IDE** - це програмне середовище розробки, що використовує мови програмування **С/C++** та призначене для програмування всіх плат ряду Arduino.



Arduino IDE дозволяє складати програми зручному текстовому редакторі, компілювати їх у машинний код, та завантажувати на всі версії плати Arduino.

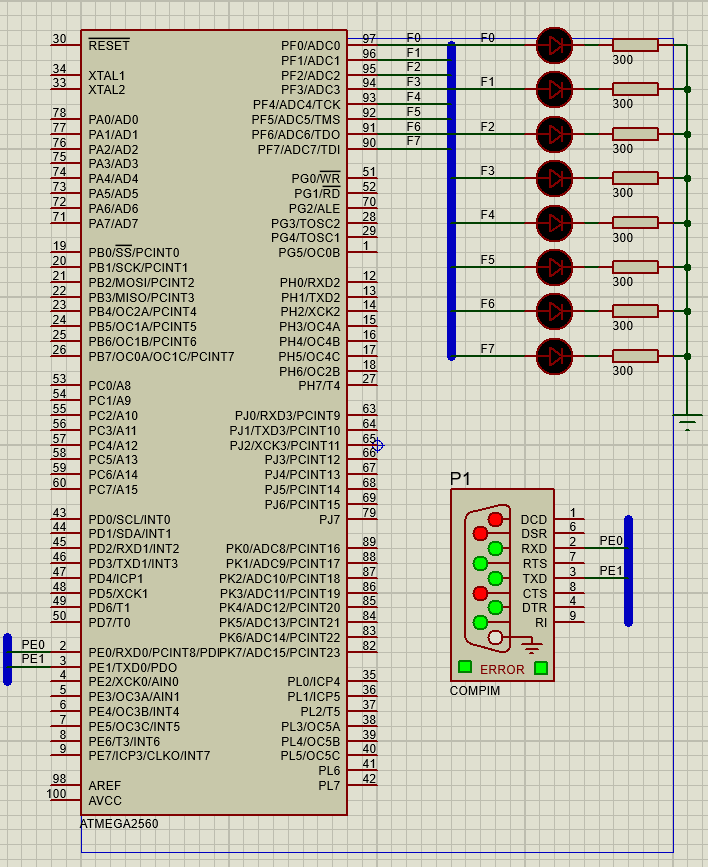
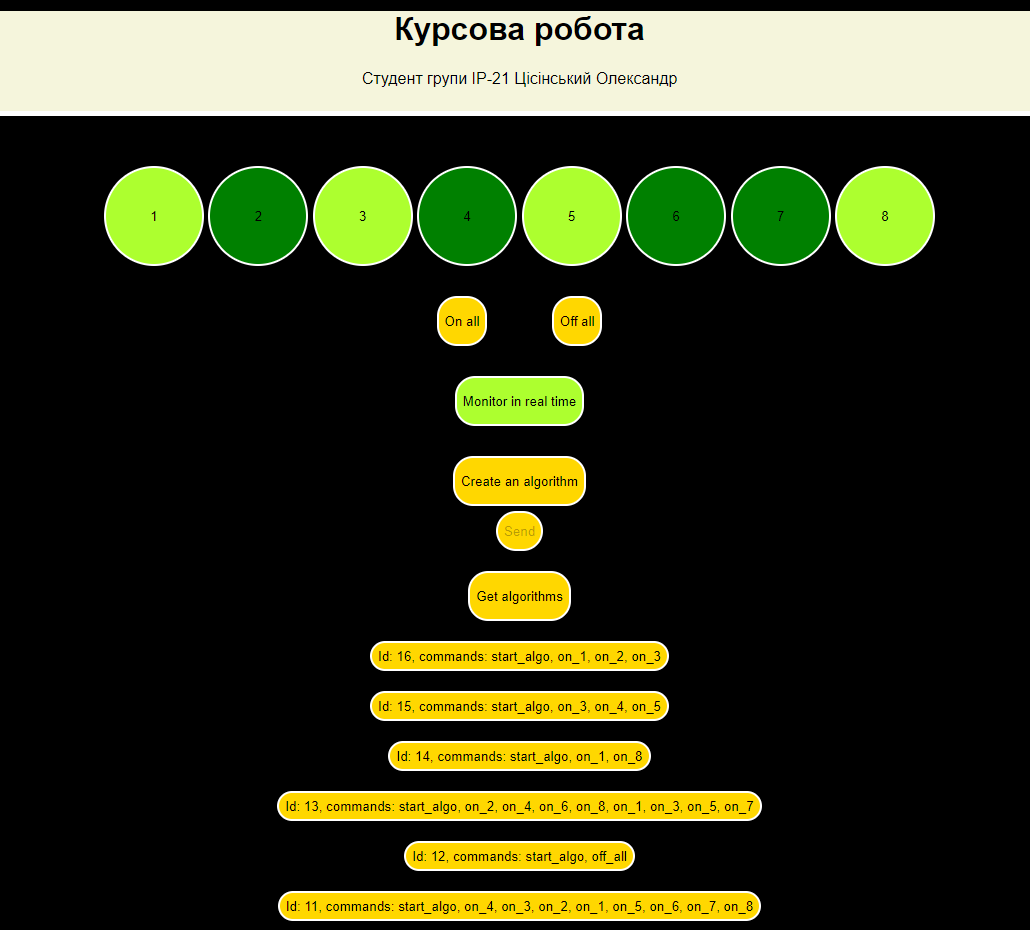
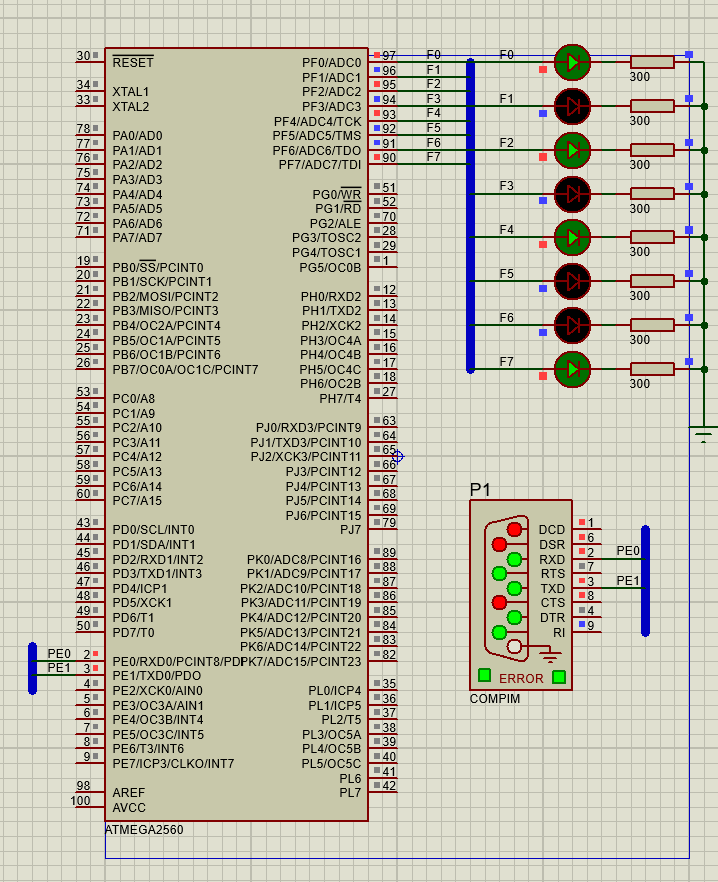
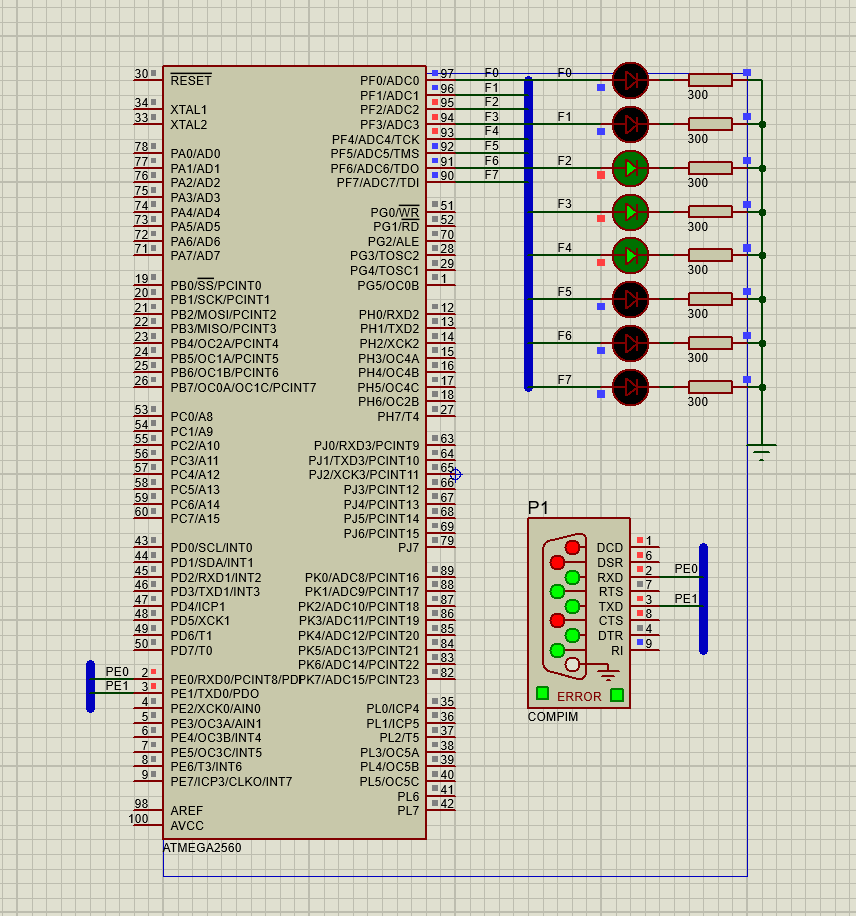
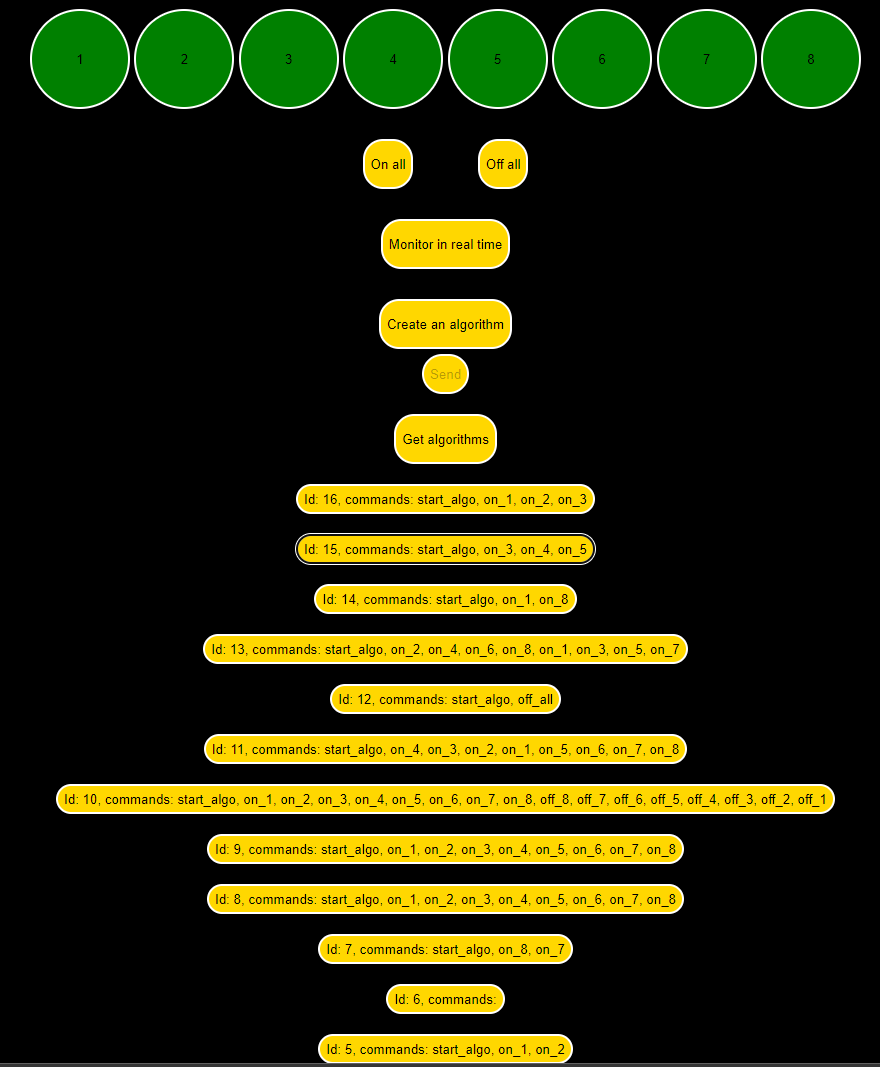


Рис. 5) Скрін загальної симуляції

**Загальна ілюстрація працездатності проекту**



Як видно з попередніх двох зображень, включенні світло діоди співпадають з світло діодами на симуляції(увімкнений режим monitor in real time).



На фото вибраний алгоритм з ID 15, який запускає алгоритм на включення 3, 4 та 5 світло діодів. На фото справа можна побачити виконання його алгоритму.

Посилання на репозиторій **GitHub**

* <https://github.com/igorka2421/final_work_for_the_2nd_semester/pulls>

**Висновки**

Процес виконання цього завдання надав мені можливість значно розширити свої знання та краще зрозуміти взаємодію між клієнтським інтерфейсом та фізичним обладнанням, в моєму випадку симуляції. Окрім нових знань, виконання курсової роботи допомогло систематизувати та закріпити матеріал, вивчений протягом навчання. Особливо під час виконання цієї роботи я удосконалив навички програмування на Arduino та створення схем у Proteus.

Виконання таких завдань цікаве, оскільки це чудова практика для подальшого вивчення інформаційних технологій.