

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Кафедра ПЗ

Методичні вказівки
до лабораторних робіт
з дисципліни
«Інженерія програмного забезпечення вбудованих систем»
для студентів ОР «бакалавр»
спеціальності 121 «Інженерія програмного забезпечення»

2023

Методичні вказівки до лабораторних робіт з дисципліни «Інженерія програмного забезпечення вбудованих систем» для студентів ОР «бакалавр» спеціальності 121 «Інженерія програмного забезпечення» / Укл.: А.В. Пархоменко, О.М. Гладкова. – Запоріжжя: НУЗП, 2023. – 84 с.

Укладачі: А.В. Пархоменко, к.т.н., доцент каф. ПЗ
О.М. Гладкова, к.т.н., доцент каф. ПЗ

Рецензент: І.Я. Зеленцова, к.т.н., доц.каф. КСМ

Відповідальний
за випуск: С.О. Субботін, д.т.н., професор, зав.каф. ПЗ

Затверджено
на засіданні кафедри
Програмних засобів
Протокол № 12
від 09.06.2023 р.

ЗМІСТ

ЛАБОРАТОРНА РОБОТА № 1 ВІРТУАЛЬНЕ ПРОЄКТУВАННЯ ЕЛЕКТРОННИХ ПРИЛАДІВ. ЗНАЙОМСТВО З PROTEUS VSM.	4
ЛАБОРАТОРНА РОБОТА № 2 ПРИНЦИПИ ВИМІРЮВАННЯ В ЕЛЕКТРОНІЦІ. РОБОТА З КНОПКОЮ ТА ЗМІННИМ РЕЗИСТОРОМ	15
ЛАБОРАТОРНА РОБОТА № 3 ПОЧАТОК РОБОТИ З ARDUINO. СИМУЛЯЦІЯ РОБОТИ ARDUINO ЗА ДОПОМОГОЮ ВЕБЗАСТОСУНКУ TINKERCAD	20
ЛАБОРАТОРНА РОБОТА № 4 СТВОРЕННЯ ПРОЄКТУ РОБОТИ ЗІ СВІТЛОДІОДАМИ З ВИКОРИСТАННЯМ СЕРЕДОВИЩ PROTEUS VSM ТА ATMEL STUDIO	29
ЛАБОРАТОРНА РОБОТА № 5 ПРИНЦИПИ РОБОТИ З RGB СВІТЛОДІОДОМ, КНОПКОЮ ТА ЗМІННИМ РЕЗИСТОРОМ	35
ЛАБОРАТОРНА РОБОТА № 6 СТВОРЕННЯ ПРОЄКТУ З ВИКОРИСТАННЯМ СЕРВО	41
ЛАБОРАТОРНА РОБОТА № 7 СТВОРЕННЯ ПРОЄКТУ З ВИКОРИСТАННЯМ РЕЛЕ	52
ЛАБОРАТОРНА РОБОТА № 8 РОБОТА З БІБЛІОТЕКОЮ ARDUINO У PROTEUS. ПРОГРАМУВАННЯ LCD ДИСПЛЕЯ	63
ЛАБОРАТОРНА РОБОТА № 9 ПРОЄКТУВАННЯ ДРУКОВАНОЇ ПЛАТИ З ВИКОРИСТАННЯМ СЕРЕДОВИЩА PROTEUS ARES	68
ДЖЕРЕЛА ІНФОРМАЦІЇ.....	84

ЛАБОРАТОРНА РОБОТА № 1

ВІРТУАЛЬНЕ ПРОЄКТУВАННЯ ЕЛЕКТРОННИХ ПРИБАДІВ. ЗНАЙОМСТВО З PROTEUS VSM.

Мета роботи: ознайомитись з середовищем Proteus VSM. Оволодіти основними принципами роботи світлодіодів та підключення світлодіодів до живлення.

1. Теоретичні відомості

1.1 Світлодіод та його використання

Світлодіоди, або світловипромінюючі діоди (СВД, в англійському варіанті LED — light emitting diode)— напівпровідниковий прилад, що випромінює некогерентне світло при пропусканні через нього електричного струму. Робота заснована на фізичному явищі виникнення світлового випромінювання при проходженні електричного струму через р-п-перехід. Колір світіння (довжина хвилі максимуму спектра випромінювання) визначається типом використовуваних напівпровідникових матеріалів, що утворюють р-п-перехід.



Переваги:

1. Світлодіоди не мають ніяких скляних колб і ниток розжарювання, що забезпечує високу механічну міцність і надійність (ударна і вібраційна стійкість);
2. Відсутність розігріву і високих напруг гарантує високий рівень електро- і пожежобезпеки;
3. Безінерційність робить світлодіоди незамінними, коли потрібна висока швидкодія;
4. Мініатюрність;
5. Довгий термін служби (довговічність);
6. Високий ККД;

7. Відносно низькі напруги живлення та споживані струми, низьке енергоспоживання;

8. Велика кількість різних кольорів світіння, спрямованість випромінювання;

9. Регульована інтенсивність.

Недоліки:

1. Відносно висока вартість. Відношення ціна/люмен для звичайної лампи накаливання порівняно зі світлодіодами становить приблизно 100 раз;

2. Малий світловий потік від одного елементу;

3. Деградація параметрів світлодіодів з часом;

4. Підвищені вимоги до живлячого джерела.

Зовнішній вигляд та основні параметри:

У світлодіодів є декілька основних параметрів:

1. Тип корпусу

2. Типовий (робочий) струм

3. Зменшення (робоче) напруги

4. Колір свічення (довжина хвилі, нм)

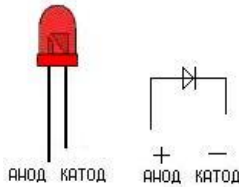
5. Кут розсіювання

Загалом, під типом корпусу розуміють діаметр та колір колби (лінзи). Як відомо, світлодіод - напівпровідниковий прилад, який необхідно живити струмом. Таким чином струм, яким слід живити той чи інший світлодіод називається типовим. При цьому на світлодіоді падає певна напруга. Колір випромінювання визначається як використовуваними напівпровідниковими матеріалами, так і легуючими домішками. Найважливішими елементами, використовуваними в світлодіодах, є: Алюміній (Al), Галій (Ga), Індій (In), Фосфор (P), що викликають свічення в діапазоні від червоного до жовтого кольору. Індій (In), Галій (Ga), Азот (N) використовують для здобуття блакитного і зеленого свічень. Крім того, якщо до кристала, що викликає блакитне (синє) свічення, додати люмінофор, то отримаємо білий колір світлодіоду. Кут випромінювання також визначається виробничими характеристиками матеріалів, а також колбою (лінзою) світлодіоду.

В даний час світлодіоди знайшли вживання в самих різних областях: світлодіодні ліхтарі, автомобільна світлотехніка, рекламні вивіски, світлодіодні панелі і індикатори, світлофори і т.п.

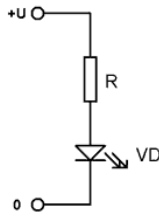
Схема підключення та розрахунок необхідних параметрів:

Оскільки світлодіод є напівпровідниковим приладом, то при включенні в електричне коло необхідно дотримувати полярність. Світлодіод має два виводи, один з яких катод ("мінус"), а інший - анод ("плюс").



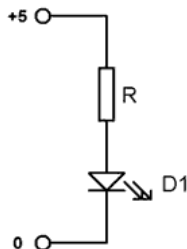
1. Маємо один світлодіод, як його підключити правильно у найпростішому випадку?

Щоб правильно підключити світлодіод у самому простішому випадку, необхідно підключити його через струмообмежувальний резистор.



Задача 1

Є блакитний світлодіод з робочою напругою 3 вольти і робочим струмом 20 мА. Необхідно підключити його до джерела з напругою 5 вольт.



У кожного світлодіода існує пряме падіння напруги (робоча напруга) при робочому струмі. Для світлодіодів одного кольору воно приблизно однакове: для червоних 1,8...2В, жовтих і зелених 2...2,4В, синіх і білих 3...3,6В. Ці данні є приблизними, для точного визначення краще дивитися у даташит (datasheet) конкретного світлодіода.

Розрахунок опору струмообмежувального резистора виконується за формулою:

$$R = U_{\text{гасяча}} / I_{\text{світлодіода}}$$

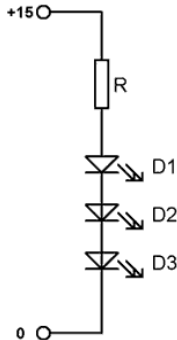
$$U_{\text{гасяча}} = U_{\text{живлення}} - U_{\text{світлодіода}}$$

2. Як підключити декілька світлодіодів?

Декілька світлодіодів підключаємо послідовно чи паралельно, розраховуючи необхідні опори.

Задача 2.

Є блакитні світлодіоди з робочою напругою 3 вольти і робочим струмом 20 мА. Треба підключити 3 світлодіоди до джерела 15 вольт.



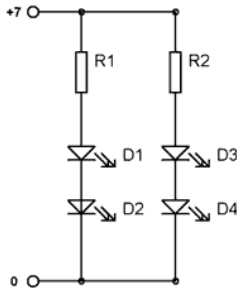
Розрахунок аналогічний попередній задачі:

$$R = U_{\text{гасяча}} / I_{\text{світлодіода}}$$

$$U_{\text{гасяча}} = U_{\text{живлення}} - N * U_{\text{світлодіода}}$$

Задача 3.

Нехай є 2 червоні світлодіоди з робочою напругою 1,8 вольти і робочим струмом 20 мА, та 2 жовті світлодіоди з робочою напругою 2,2 вольти і робочим струмом 20 мА. Треба підключити 4 світлодіоди до джерела 7 вольт.



$$R = U_{\text{гасяча}} / I_{\text{світлодіода}}$$

$$U_{\text{гасяча}} = U_{\text{живлення}} - N * U_{\text{світлодіода}}$$

1.2 Середовище віртуального проєктування електронних схем Proteus VSM

PROTEUS VSM – середовище для проєктування та симуляції роботи електронних схем. Відмінністю пакета Proteus VSM є можливість моделювання роботи програмованих пристроїв: мікроконтролерів (МК), мікропроцесорів, DSP та ін..

PROTEUS VSM дозволяє дуже достовірно моделювати і налагоджувати досить складні пристрої, в яких може міститися кілька МК одночасно і навіть різних родин в одному пристрої.

PROTEUS складається з двох основних модулів:

- ISIS – графічний редактор принципів схем служить для введення розроблених проєктів з подальшою імітацією і передачею для розробки друкованих плат в ARES. До того ж після налагодження пристрою можна відразу розвести друковану плату в ARES яка підтримує авто розміщення і трасування по вже існуючій схемі.

- ARES – графічний редактор друкованих плат з вбудованим менеджером бібліотек і автотрасувальник ELECTRA, автоматичної розстановкою компонентів на друкованій платі.

На рисунку 1.1 зображене головне вікно програми. Увесь робочий простір програми розподілений на декілька основних частин:

- вікно редактору схем (виконується синтез з окремих компонентів);
- вікно вибору об'єктів (доступні різні елементи в залежності від вибраного режиму);

- панель керування симуляцією (знаходиться у лівому нижньому куті, включає в себе такі команди: пуск; виконання одного такту, яке включає симуляцію на час Single Step Time, який задається у розділі головного меню System>Set Animation Options; пауза; стоп).

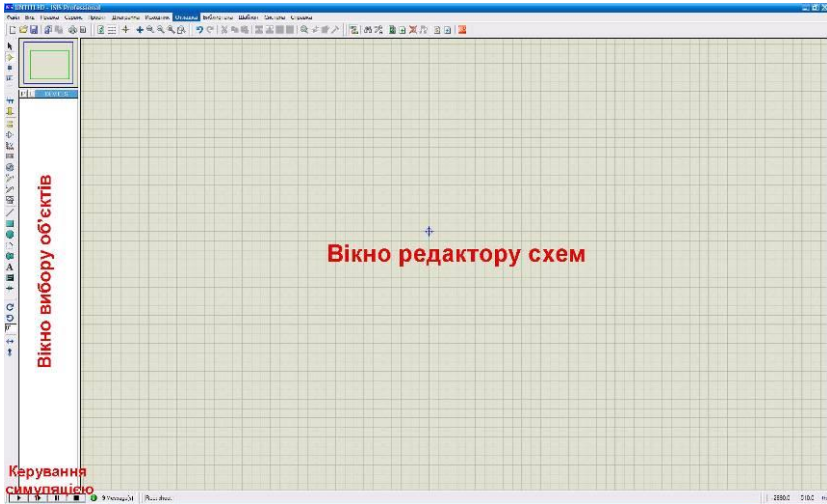
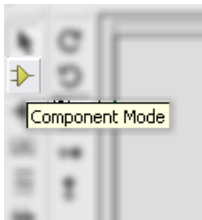


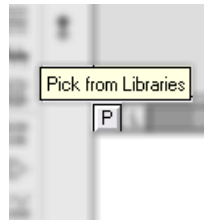
Рисунок 1.1 – Головне вікно Proteus VSM (ISIS)

1.3 Проектування віртуальної моделі електронної схеми та симуляція її роботи в Proteus VSM

Для того щоб зібрати схему будь-якого пристрою, необхідно підготувати набір елементів, з яких буде складатися ця схема. Для цього переходимо в режим компонентів або Component Mode (рис. 1.2a) і клікаємо клавішу P, яка знаходиться під вікном перегляду поряд з клавішею L (рис 1.2б).



а)



б)

Рисунок 1.2 – Панель інструментів

Перед нами з'являється менеджер компонентів, який надає на наш вибір всі елементи, які містяться в бібліотеці програми (рис 1.3).

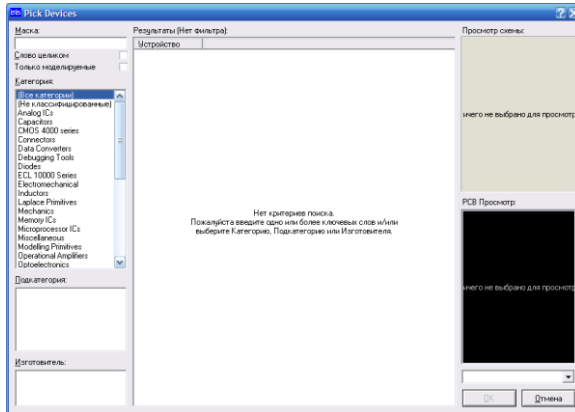


Рисунок 1.3 – Вікно менеджера компонентів

Треба користуватися рядком пошуку, який знаходиться у верхньому лівому куті. Коли потрібний компонент знайдений, двійним кліком лівої кнопки миші по його назві додасть його до переліку використовуваних компонентів. Наприклад створимо електронну схему яка буде складатися з: світлодіода, резистора і блоку живлення (батареї). У рядку пошуку (маска) вводимо перший елемент який будемо додавати до схеми: led-green. У списку елементів з'являється світлодіод з такою назвою (рис.1.4), клікаємо по ньому двійним кліком лівою кнопкою миші.

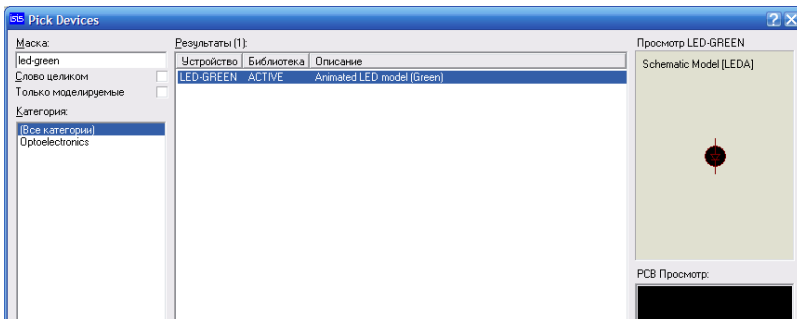


Рисунок 1.4 – Пошук компонентів

Далі так само робимо пошук за такими назвами: resistor, battery (бібліотека - Active), - і по черзі двійним кліком по цих елементах у списку додаємо їх до нашого проєкту. Після того як ми вибрали всі компоненти натискаємо на кнопку ОК. Тепер наше вікно вибору об'єктів буде містити всі чотири компоненти що ми вибрали і виглядати як зображено на рисунку 1.5.

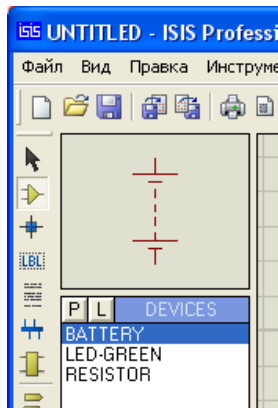
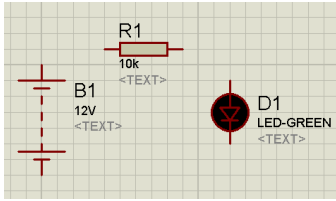


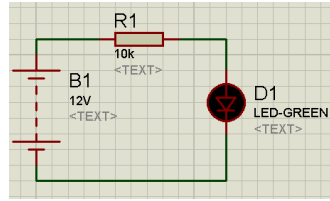
Рисунок 1.5 – Вікно вибору об'єктів

Для встановлення компоненту у вікні редактора схеми його необхідно вибрати зі списку і двійним кліком лівої кнопки миші встановити у потрібному місці. До того як встановити компонент на схемі його можна попередньо розвернути у потрібне положення, яку можна контролювати у вікні перегляду.

Встановіть всі компоненти як зображено на рисунку 1.6 (а), та з'єднайте їх (рис 1.6 б). Для того щоб з'єднати два компоненти між собою для початку виберіть інструмент Стрілка, потім на самій схемі наведіть курсор миші на кінець елемента, повинен з'явитися квадратний червоний контур, натискаємо і відпускаємо ЛКМ, і ведемо курсор до контакту іншого елемента, і на кінці іншого елемента також натискаємо ЛКМ, щоб завершити побудову з'єднання між двома елементами.



а)



б)

Рисунок 1.6 – Побудова електричної схеми

Після того як електричне коло створено, міняємо характеристики наших компонентів. Зараз батарея живить схему 12В, а нам потрібно змінити живлення схеми на 3В. Для цього клікаємо подвійним кліком на батареї і у полі Voltage встановлюємо 3V.

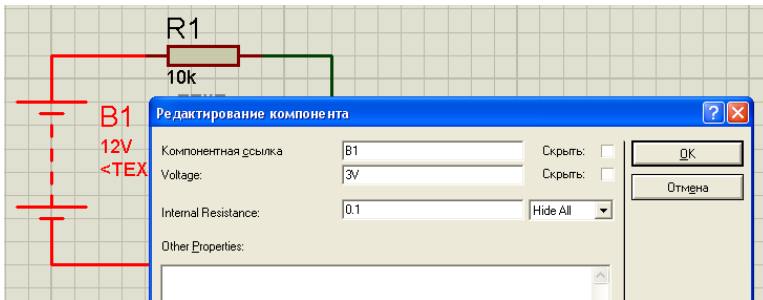


Рисунок 1.7 – Змінення параметрів живлення

Так само змінюємо параметри світлодіоду. У нашому випадку в нас зелений світлодіод, тож встановлюємо у вікні параметрів світлодіоду падіння напруги 2В та робочий струм 20mA.

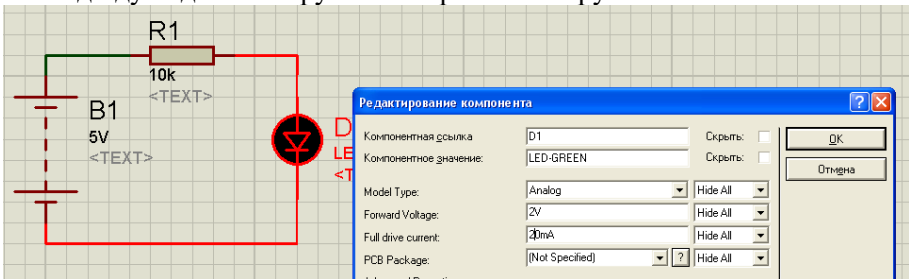


Рисунок 1.8 – Змінення параметрів опору

Далі робимо розрахунок який опір нам необхідний для нормальної роботи світлодіоду. Для цього розраховуємо скільки напруги гаситиметься на нашому світлодіоді: $U_{\text{гасяча}} = 3\text{В} - 2\text{В} = 1\text{В}$. Згідно закону Ома необхідний опір резистора: $R = U/I = 1\text{В} / 0,02\text{А} = 50 \text{ (Ом)}$. У симуляторі ми можемо написати опір 50 Ом, але коли треба спроектувати реальну схему, оскільки такого опору в стандартному ряді резисторів нема, то будемо брати найближчий у сторону зростання – 51Ом.

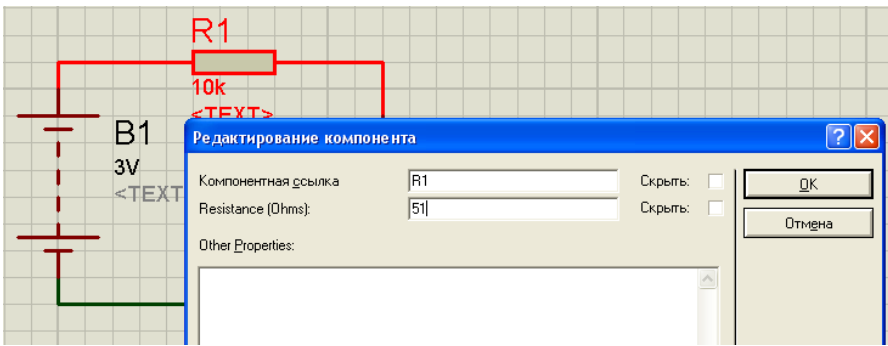


Рисунок 1.9 – Змінення параметрів опору

Після натискаємо в панелі керування симуляцією кнопку «Воспроизвести».



Рисунок 1.10 – Панель керування симуляцією

Тепер наша схема може віртуально відтворити включення світлодіоду (рис 1.11).

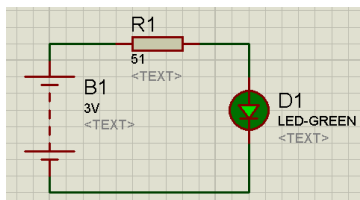


Рисунок 1.11 – Симуляція роботи світлодіоду

2. Завдання до лабораторної роботи

- 2.1 Спроекувати приклад приведений у розділі 1.3.
- 2.2 Вирішити 3 задачі які наводяться у теоретичній частині цієї роботи. Спроекувати віртуальні моделі електронних схем та симуляція їх роботи в Proteus VSM.
- 2.3 Оформити звіт

3. Зміст звіту

- 3.1 Тема та мета роботи
- 3.2 Рішення 3 задач
- 3.3 Схема електрична-принципова підключення світлодіоду у середовищі Proteus VSM
- 3.4 Висновки з роботи

4. Контрольні запитання

- 4.1 Дайте визначення закону Ома?
- 4.2 Що таке резистор?
- 4.3 Навіщо потрібен резистор у схемі підключення?
- 4.4 Запишіть формули розрахунку опору послідовно та паралельно підключених резисторів у колі.
- 4.5 З якої бібліотеки Ви використовували елементи для побудови електричної схеми у середовищі Proteus?
- 4.6 В якому режимі, у середовищі Proteus VSM, відбувається пошук елементів для схеми?
- 4.7 Які власні параметри є у світлодіоду?

ЛАБОРАТОРНА РОБОТА № 2

ПРИНЦИПИ ВИМІРЮВАННЯ В ЕЛЕКТРОНІЦІ. РОБОТА З КНОПКОЮ ТА ЗМІННИМ РЕЗИСТОРОМ

Мета роботи: навчитися користуватися різними електровимірювальними приладами, ознайомитись з основними принципами роботи та підключення змінного резистора та кнопки в середовищі Proteus.

1. Теоретичні відомості

Для того щоб переглянути, які вимірювальні прилади наявні в середовищі Proteus VSM, необхідно вибрати «Віртуальні інструменти» (рис. 2.1).

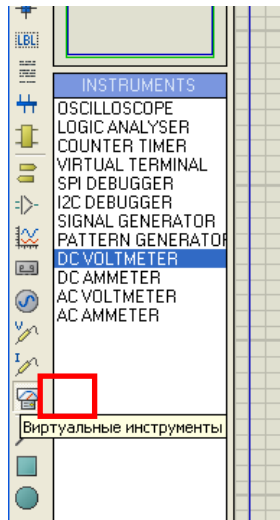


Рисунок 2.1 – Електровимірювальні прилади в середовищі Proteus VSM

У цьому курсі лабораторних робіт ми розглянемо роботу таких приладів як: DC Voltmeter (вольтметр постійного струму), DC Ammeter (амперметр), Oscilloscope (осцилограф). АС використовується в електричних колах змінного струму.

Створимо електричну схему, як зображено на рисунку 2.2.

Щоб знайти елемент кнопка у бібліотеці Proteus, введіть push button.

Кнопка – найпростіший електромеханічний пристрій, для передачі електричного сигналу різним пристроям шляхом замикання або розмикання двох і більше контактів. В залежності від стану, в якому залишається кнопка після натискування на неї, вона буває двох видів:

- кнопки (перемикачі) з фіксованим положенням;
- тактові кнопки або нефіксовані (які повертаються у вихідне положення).

Встановіть живлення батареї 5В, показники зеленого світлодіода залиште не змінними та розрахуйте необхідне значення опору резистора.

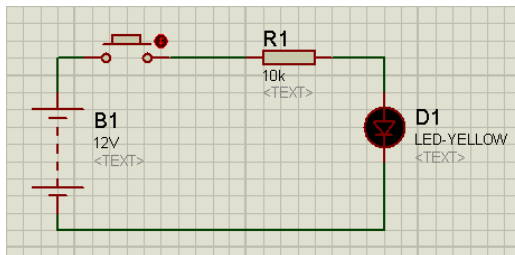


Рисунок 2.2 – Побудова електричної схеми

Після натискаємо в панелі керування симуляцією кнопку «Воспроизвести».



Рисунок 2.3 – Панель керування симуляцією

Зараз наша схема може віртуально відтворити включення та виключення світлодіода при натисканні на кнопку. Для цього просто наведіть курсор миші на елемент кнопка у схемі і натисніть ЛКМ. Для того щоб затиснути кнопку клікніть ЛКМ на круг з правої сторони елемента кнопки



Спроектуйте схему з першої роботи з додаванням кнопки. Далі перейдіть до віртуальних приладів в панелі інструментів зліва та підключіть DC Voltmeter, DC Ammeter, як зображено на рисунку 2.4. Ці

прилади показують нам значення струму в електричному колі та значення напруги на світлодіоді, відповідно.

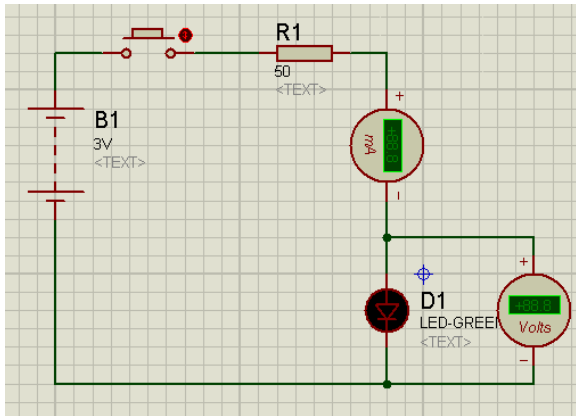


Рисунок 2.4 – Схема підключення вольтметра і амперметра

Щоб змінити відображення значень амперметра з Амперів на міліАмperi або мікроАмperi, у властивостях амперметра у полі Display Range встановить необхідне значення (рис. 2.5).

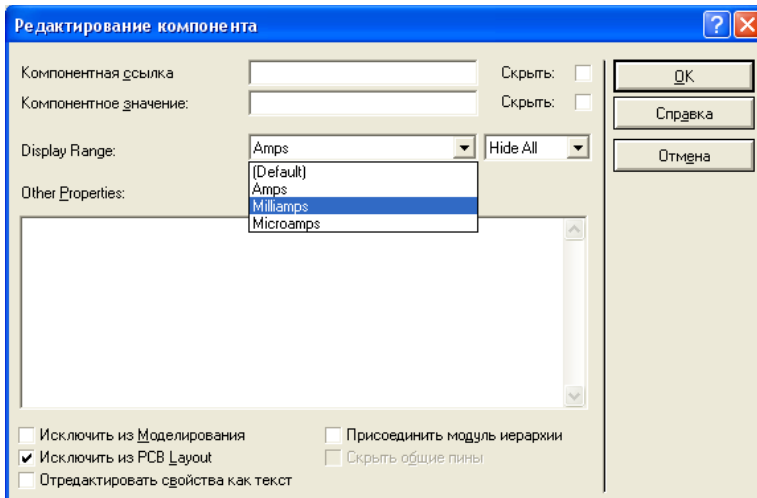
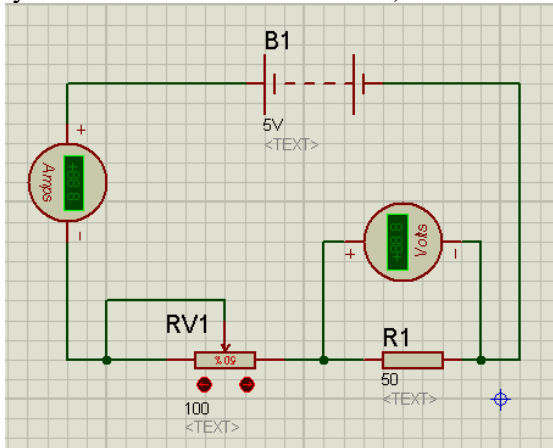


Рисунок 2.5 – Встановлення одиниць вимірювання

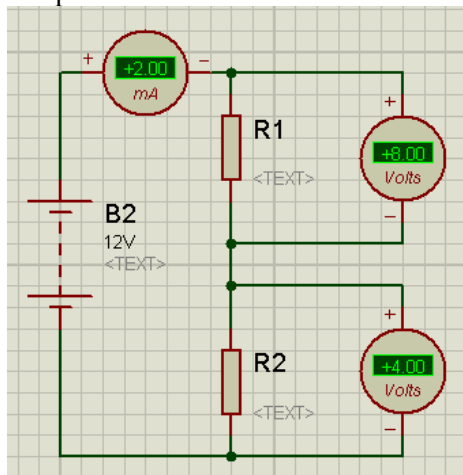
2. Завдання до лабораторної роботи

2.1 Спроектуйте електричну схему, як зображено на рисунку. Виконайте 3 сценарії: коли опір змінного резистора є нульовим, максимальним або має середнє значення (змінний резистор або потенціометр у Proteus називається POT-HG).

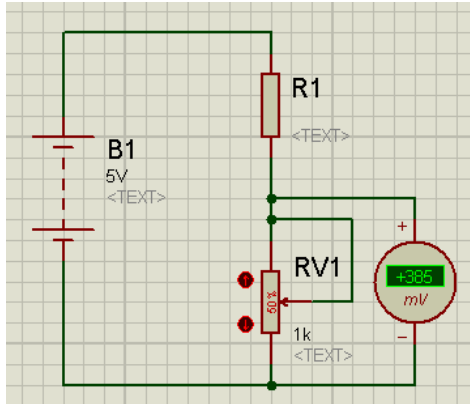


2.2 Спроектуйте схеми, що представлені нижче та визначте:

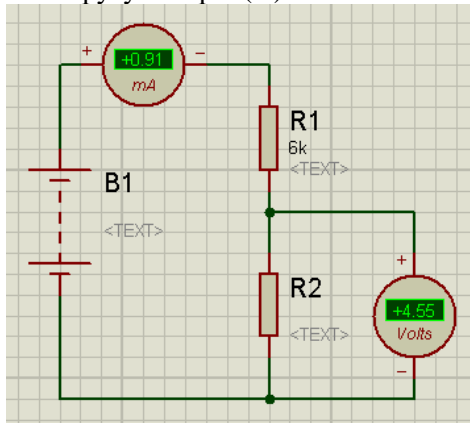
1) опір резисторів



2) опір R1 та силу струму (I)



3) опір R2 та напругу батареї (U)



2.3 Розрахуйте самостійно силу струму на рисунку 2.4?

3. Контрольні запитання

- 3.1. Що таке амперметр?
- 3.2. Що таке вольтметр?
- 3.3. Що таке осцилограф?
- 3.4. В чому різниця між постійним і змінним струмом?
- 3.5. Переведіть 1 Ампер в мікро та мілі Амperi.

ЛАБОРАТОРНА РОБОТА № 3

ПОЧАТОК РОБОТИ З ARDUINO. СИМУЛЯЦІЯ РОБОТИ ARDUINO ЗА ДОПОМОГОЮ ВЕБЗАСТОСУНКУ TINKERCAD

Мета роботи: ознайомитися з основними елементами плати Arduino Mega та вебзастосунка віртуальної імітації роботи платформи Arduino, розробити прості проекти мерехтіння світлодіодів з використанням можливостей вебзастосунка та реальної платформи.

1. Теоретичні відомості

1.1 Arduino

Arduino – це електронний конструктор і зручна платформа швидкої розробки електронних пристроїв для новачків і професіоналів. Переваги платформи: зручна у використанні, простота мови програмування, а також відкрита архітектура і програмний код. Пристрій програмується через USB без використання програматорів.

Пристрої на базі Arduino можуть отримувати інформацію про навколишнє середовище за допомогою різних датчиків, а також можуть управляти різними виконавчими пристроями.

Існує декілька версій платформ Arduino. В цьому курсі буде розглядатися платформа Arduino Uno (рис. 3.1).

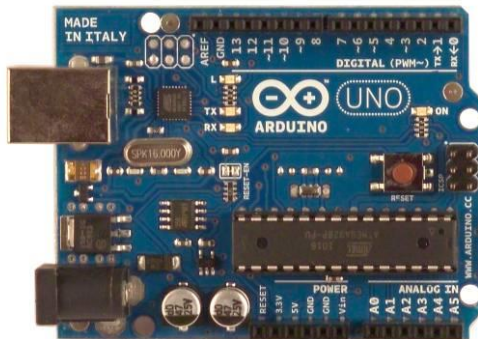


Рисунок 3.1 – Зовнішній вигляд Arduino Uno

Arduino Uno базується на мікроконтролері Atmel ATmega328. Мікроконтролер на платі програмується за допомогою мови Arduino (заснований на мові Wiring) і середовища розробки Arduino (заснована на середовищі Processing). Проекти пристроїв, засновані на Arduino, можуть працювати самостійно, або ж взаємодіяти з програмним забезпеченням на комп'ютері (напр.: Flash, Processing, MaxMSP). Плати можуть бути зібрані користувачем самостійно або куплені в зборі. Програмне забезпечення доступне для безкоштовного скачування. Вихідні креслення схем (файли CAD) є загальнодоступними, користувачі можуть застосовувати їх на свій розсуд.

Платформа має 14 цифрових вхід/виходів (6 з яких можуть використовуватися як виходи ШІМ - Широтно-імпульсна модуляція), 6 аналогових входів, кварцовий генератор 16 МГц, роз'єм USB, силовий роз'єм, роз'єм ICSP і кнопку перезавантаження. Кожен з 14 цифрових виводів Uno може налаштований як вхід або вихід, використовуючи функції `pinMode ()`, `digitalWrite ()`, і `digitalRead ()`. Кожен вивід має навантажувальний резистор (за замовчуванням відключений) 20-50 кОм і може пропускати до 40 мА. Для роботи необхідно підключити платформу до комп'ютера за допомогою кабелю USB, або подати живлення за допомогою адаптера AC/DC або батареї.

Платформа може працювати при зовнішньому живленні від 6 В до 20 В. При напрузі живлення нижче 7 В, вивід 5V може видавати менше 5 В, при цьому платформа може працювати нестабільно. При використанні напруги вище 12 В регулятор напруги може перегрітися і пошкодити плату. Рекомендований діапазон від 7 В до 12 В.

Деякі виводи мають особливі функції:

- VIN. Вхід використовується для подачі живлення від зовнішнього джерела (в відсутність 5 В від роз'єму USB або іншого регульованого джерела живлення). Подача напруги живлення відбувається через даний вивід.

- 5V. Регульоване джерело напруги, що використовується для живлення мікроконтролера і компонентів на платі. Живлення може подаватися від виводу VIN через регулятор напруги, або від роз'єму USB, або іншого регульованого джерела напруги 5 В.

- 3V3. Напруга на виводі 3.3 В генерується вбудованим регулятором на платі. Максимальне споживання струму 50 мА.

- GND. Виводи заземлення.

- Послідовна шина: 0 (RX) і 1 (TX). Виводи використовуються для отримання (RX) і передачі (TX) даних TTL. Дані виводи підключені до відповідних роз'ємів мікросхеми послідовної шини ATmega8U2 USB-to-TTL.

- Зовнішнє переривання: 2 і 3 Дані висновки можуть бути налаштовані на виклик переривання або на молодшому значенні, або на передньому чи задньому фронті, або при зміні значення. Детальна інформація знаходиться в описі функції `attachInterrupt ()`.

- ШІМ: 3, 5, 6, 9, 10, і 11. Будь-який з виводів забезпечує ШІМ з роздільною здатністю 8 біт за допомогою функції `analogWrite ()`.

- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). За допомогою даних виводів здійснюється зв'язок SPI, для чого використовується бібліотека SPI.

- LED: 13. Вбудований світлодіод, підключений до цифрового виводу 13. Якщо значення на виводі має високий потенціал, то світлодіод горить.

На платформі Uno встановлені 6 аналогових входів (позначених як A0 .. A5), кожен роздільною здатністю 10 біт (тобто може приймати 1024 різних значення). Стандартно виводи мають діапазон вимірювання до 5 В відносно землі, проте є можливість змінити верхню межу за допомогою виводу AREF і функції `analogReference ()`. Деякі виводи мають додаткові функції:

- I2C: 4 (SDA) і 5 (SCL). За допомогою виводів здійснюється зв'язок I2C (TWI), для створення якої використовується бібліотека Wire.

Додаткова пара виводів платформи:

- AREF. Опорна напруга для аналогових входів. Використовується з функцією `analogReference ()`.

- Reset. Низький рівень сигналу на виводі перезавантажує мікроконтролер. Звичайно застосовується для підключення кнопки перезавантаження на платі розширення, що закриває доступ до кнопки на самій платі Arduino.

–

1.2 Основні функції для роботи зі світлодіодами

Для роботи зі світлодіодом необхідно знати і вміти володіти такими функціями і константами:

- оператор `setup()`;
- оператор `loop()`;
- функція `pinMode()`
- функція `digitalWrite()`;
- функція `delay()`;
- константи `OUTPUT`, `HIGH`, `LOW`.

Далі наведений код програми найпростішого прикладу мерехтіння вбудованим у плату Arduino світлодіодом, який підключено до 13 виводу:

```
void setup()
{ pinMode(13, OUTPUT);
}
```

Ця функція виконується на початку роботи програми (після запуску мікроконтролеру). Тобто послідовно виконується кожна команда, яка знаходиться між фігурними скобками цієї функції. Наприкінці кожної строки необхідно поставити символ закінчення команди “;”. Тут функція `setup` містить одну єдину команду – `pinMode(13, OUTPUT)`. Ця команда налаштовує 13 порт Arduino, як вивід. Порт № 13 знаходиться на верхній колодці портів Arduino.

Після функції `setup` виконується функція `loop`.

```
void loop()
{
digitalWrite(13, HIGH); // вмикаємо світлодіод
delay(1000); // чекаємо секунду
digitalWrite(13, LOW); // вимикаємо світлодіод
delay(1000); // чекаємо секунду
}
```

На відміну від `setup`, функція `loop` постійно повторюється – як тільки послідовно виконані всі команди в скобках, функція запускається знову. Функція `loop` для цього прикладу складається з чотирьох команд:

На порт 13 подається напруга (5 В) – світлодіод вмикається.

Затримка до виконання наступної команди 1000 мілісекунд (одну секунду)

Порт 13 з'єднується з землею – світлодіод вимикається.

Ще одна затримка на 1 секунду.

Після виконання усіх чотирьох команд, знову виконується перша команда (включення світлодіоду) и так продовжується до тих пір, поки Arduino включена або поки не буде натиснута кнопка RESET.

1.3 Симуляція роботи світлодіоду та Arduino в TINKERCAD

Сервіс TINKERCAD компанії Autodesk (попередня назва 123D Circuits) дає можливість створити електронні схеми. Вебзастосунок має підтримку платформи Arduino та дозволяє редагувати код, а також дає можливість у візуальному режимі будувати схем і виконувати інтерактивну імітацію їх роботи у реальному часі. Працювати над схемами можна спільно з іншими людьми, використовуючи бібліотеку компонентів.

Для роботи з сервісом TINKERCAD перейдіть за посиланням <https://www.tinkercad.com/>.

Після проходження реєстрації в TINKERCAD ви опинитесь на головній сторінці (рис. 3.2). Домашня сторінка показує список схем, які ви створили, а також нещодавно створені/редаговані.

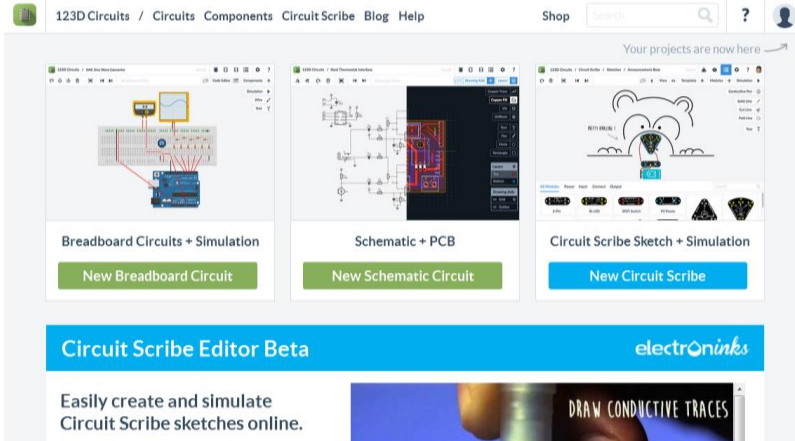


Рисунок 3.2 – Головна сторінка TINKERCAD

Щоб створити нову схему клікніть на кнопку «+New», яка знаходиться у верхньому меню. Після, натисніть кнопку "New Electronics Lab" на правій панелі вибору. Далі вам буде представлено сторінка для побудови свого проєкту з основними функціями та

інструментами для роботи зі схемою та програмою. Веб-інтерфейс програмного забезпечення, представлений на рисунку 3.3, складається з таких основних блоків:

1. робоча область створення схеми
2. панель керування проєктом
3. меню вибору відображення схеми (Breadbord View, Schematics View, PCB View)

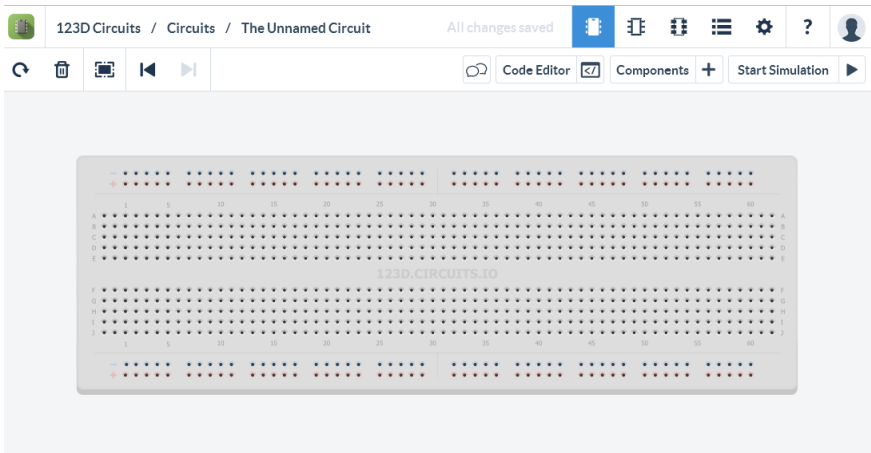


Рисунок 3.3 – Інтерфейс розробки проєкту схеми

Для розуміння роботи в середовищі TINKERCAD, розробимо простий проєкт миготіння світлодіоду. Створюємо схему за прикладом як зображено на рисунку 3.4.

Кликаємо на кнопку Components яка знаходиться на панелі керування проєктом праворуч. У вікні компонентів яке відкрилося знаходимо плату Arduino Uno, обираємо її і розміщуємо її на робочій поверхні. Оскільки даний стимулятор не має в базі платформи Arduino Mega2560 ми обираємо Arduino Uno, робота з цими платами Arduino ідентична. Далі обираємо LED (світлодіод) і розміщуємо її на Breadbord. Відкривається вікно налаштування параметрів світлодіоду, де можемо задати його ім'я та колір. Після встановлюємо резистор і також задаємо його параметри: ім'я та опір 100 Ом. Як всі компоненти встановлені, з'єднуємо їх.

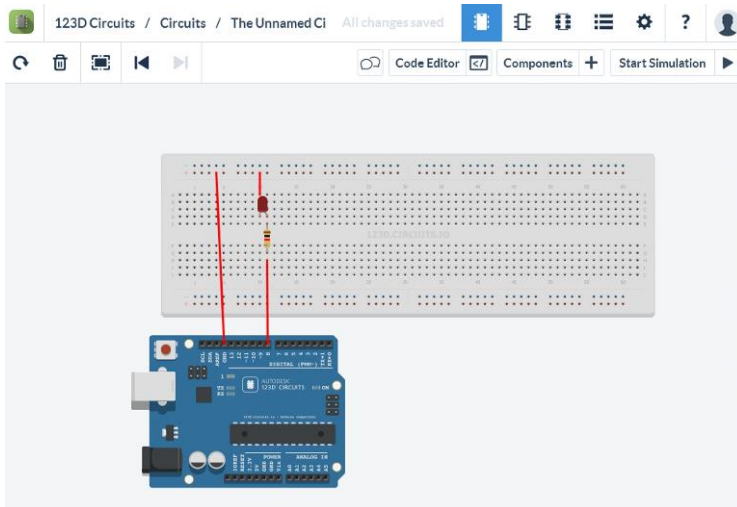


Рисунок 3.4 – Приклад схеми

Викличемо вікно редагування коду, клікнувши на кнопку «Code Editor», яка знаходиться праворуч на панелі керування проектом. У редактор копіюємо код:

```
int led = 8;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Компілюємо і завантажуюмо код, клікнувши на кнопку «Apload&Run».

2. Завдання до лабораторної роботи

2.1 Розробіть у TINKERCAD проєкт підключення світлодіода. За допомогою мультиметра зробіть вимірювання напруги, яку дає плата Arduino Uno, встановіть відповідний резистор для обраного світлодіода.

2.2 Розробіть проєкт мерехтіння 6 світлодіодів використовуючи цикл FOR.

2.3 Розробіть у TINKERCAD будь-який проєкт з підключенням осцилографа.

3. Робота з платою Arduino

3.1 Виконайте підключення плати Arduino до комп'ютера та напишіть програму мерехтіння світлодіода, який встановлений за замовчанням на 13 виході плати. Для цього:

3.2 Запустіть на комп'ютері середовище розробки Arduino

3.3 Написати код.

3.4 Підключіть плату Arduino через USB до ПК.

3.5 Натисніть кнопку Verify и переконайтесь, що у нижній частині вікна з'явився надпис Done Compiling. Це значить, що у написаній програмі не знайдено помилок.

3.6 Виберіть у Tools->Board ваш тип плати. Перевірте, чи правильно обрано USB-порт в Tools->Serial port. Після натисніть на кнопку Upload.

3.7 Якщо внизу з'явився надпис “Done uploading” – процес запису пройшов успішно.

3.8 Підключіть 3 світлодіод (послідовно червоний, жовтий і зелений) до плати Arduino. Написати код згідно з алгоритмом представленим на рисунку 3.5.

4. Зміст звіту

4.1 Тема та мета роботи

4.2 Три схеми проєкту: компонентна, принципова, друкована.

4.3 Код програми

4.4 Висновки з роботи

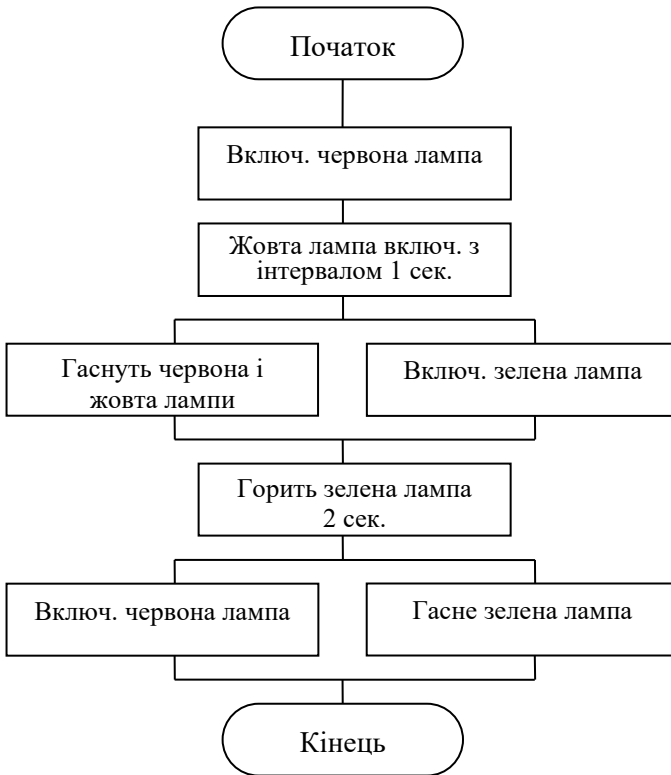


Рисунок 3.5 – Схема алгоритму роботи світлофора

5. Контрольні запитання

5.1 У якому режимі здійснюється макетне моделювання в сервісі TINKERCAD?

5.2 До якого моменту буде працювати плата Arduino?

5.3 До якого виводу підключено вбудований світлодіод на платформі Arduino Uno?

5.4 Які оператори середовища Arduino треба використовувати, щоб створити проєкт керування світло діодами?

5.5 Що таке RX та TX на платі Arduino?

5.6 Скільки усього входів/виходів має платформа Arduino Uno? Перелічте основні з них.

ЛАБОРАТОРНА РОБОТА № 4

СТВОРЕННЯ ПРОЄКТУ РОБОТИ ЗІ СВІТЛОДІОДАМИ З ВИКОРИСТАННЯМ СЕРЕДОВИЩ PROTEUS VSM ТА ATMEL STUDIO

Мета роботи: ознайомитись з середовищами Proteus VSM та Atmel Studio 6.2. Навчитися створювати проєкт віртуальної симуляції роботи світлодіоду. Ознайомитись з різновидами функцій та команд для створення прототипу світлофора.

1. Теоретичні відомості

1.1 Створення проєкту програми для мікроконтролера у Atmel Studio 6.2

Atmel Studio – інтегроване середовище розробки (IDE) для розробки 8-ми і 32-х бітних застосунків від компанії Atmel, що працює в операційних системах Windows NT/2000/XP/Vista/7. Atmel Studio містить асемблер і симулятор, що дозволяє відстежити виконання програми. Atmel Studio містить в собі менеджер проєктів, редактор вихідного коду, інструменти віртуальної симуляції та внутрішньосхемного налагодження, дозволяє писати програми на асемблері або на C / C ++.

Після того як ви встановите *Atmel Studio 6.2* на свій комп'ютер, створіть новий проєкт:

1. Відкрийте Atmel Studio 6.2 та виберіть **New Project**. У діалоговому вікні New Project, виберіть **GCC C Executable Project** як шаблон (рис.4.1). Введіть ім'я проєкту і вкажіть місце де він буде зберігатися. Назвемо наш проєкт "BlinkLED". Зніміть прапорець **Create directory for solution** для спрощення структури каталогів проєкту. Клікніть **OK**.

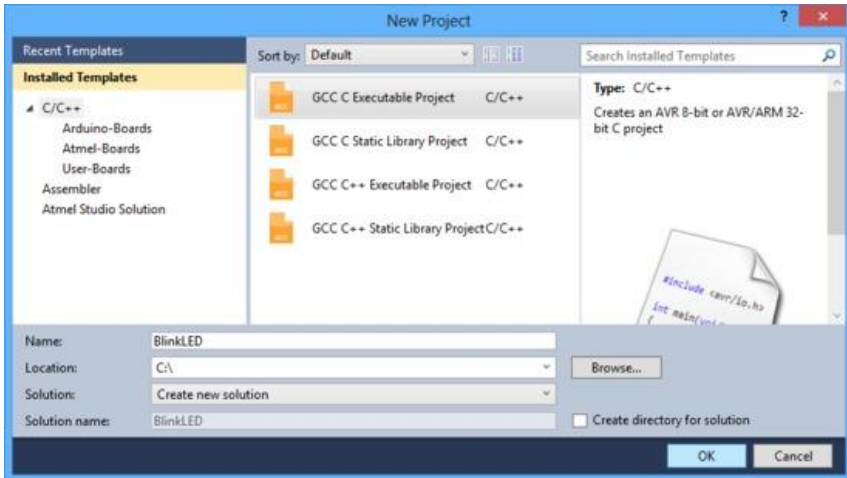


Рисунок 4.1 – Діалогове вікно New Project в Atmel Studio 6

2. У вікні Device Selection (рис. 4.2), оберіть ім'я пристрою AVR яке необхідно. Клікніть **ОК** для створення нового проєкту.

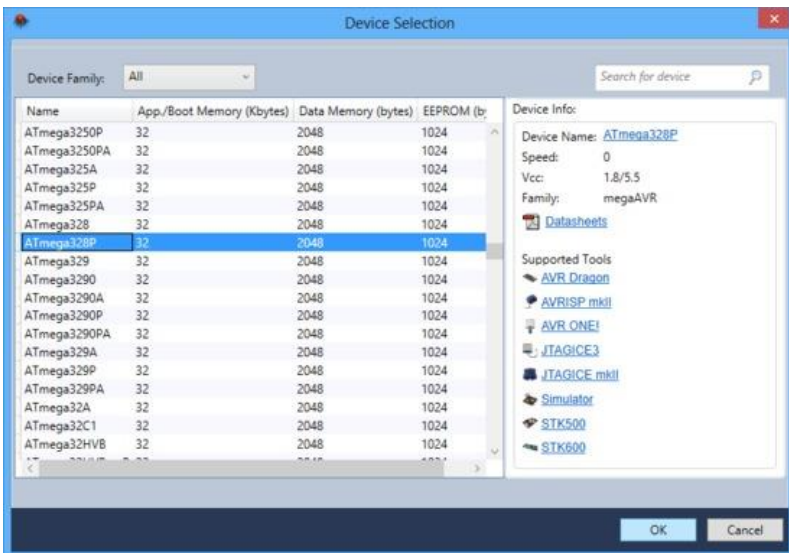


Рисунок 4.2 – Діалогове вікно Device Selection в Atmel Studio 6

3. Напишіть код у BlinkLED.cpp приведений нижче:

```
#define F_CPU 8000000      // частота AVR в Гц,
                           // використовується для util/delay.h
#include <avr/io.h>
#include <util/delay.h>

int main() {
    DDRD |= (1<<DDD1);      // установка LED виходу
    PD1 на вихід
    while (1) {
        PORTD |= (1<<PORTD1); // PD1 включення
        _delay_ms(50);        // затримка на 50 мс
        PORTD &= ~(1<<PORTD1); // PD1 виключення
        _delay_ms(100);       // затримка 100 мс
    }
}
```

4. На панелі інструментів клікніть кнопку **Build Solution** (або натисніть **F7**), щоб скомпілювати код (рис.4.3).

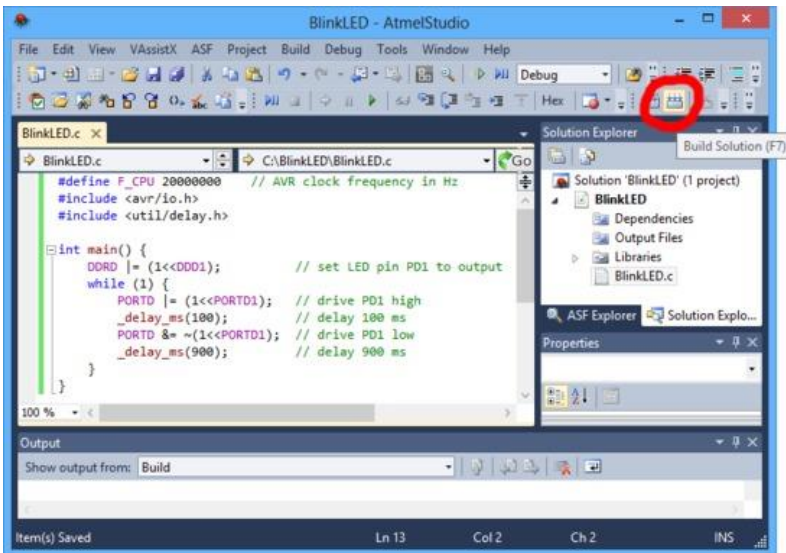


Рисунок 4.3 – Побудова проєкту

Після компіляції перейдіть в директорію проєкту і знайдіть файл BlinkLED.hex – це файл, який представляє собою довільні двійкові данні у текстовому вигляді. По історичним причинам є стандартом де-факто при прошивці різноманітних мікросхем. Саме його необхідно буде завантажити у мікроконтролер схеми у Proteus VSM, яка буде побудована далі.

1.2 Симуляція роботи мікроконтролера за допомогою Proteus VSM

На рисунку 4.4 показане розведення виводів мікроконтролера ATmega8. Світлодіод ми будемо підключати до третього виводу мікроконтролера (PD1).

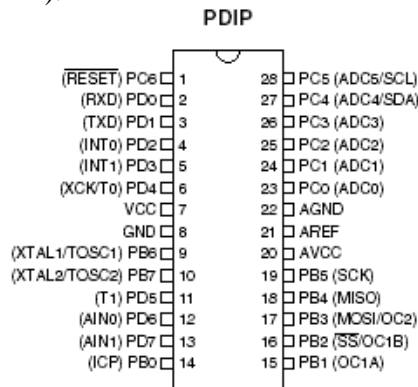



Рисунок 4.4 – Розташування виводів мікроконтролера ATmega8

Створіть новий проєкт у Proteus VSM. І розкресліть схему, як зображено на рисунку 4.5. У менеджері компонентів знайдіть нижче приведені елементи:

1. світлодіод
2. мікроконтролер ATmega328P
3. резистор
4. заземлення

Для того щоб створити елемент «заземлення» (GROUND) перейдіть у панелі інструментів (знаходиться вертикально зліва) на вкладку Terminals Mode ().

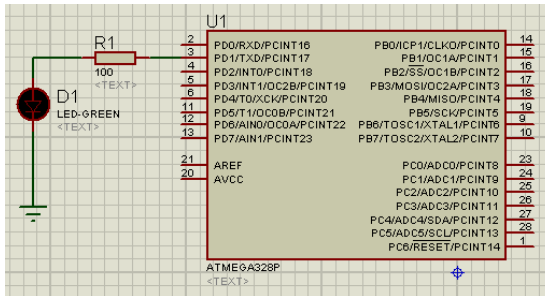


Рисунок 4.5 – Схема підключення

Після з'єднання елементів у електричне коло, подвійним кліком клікніть на зображені мікроконтролера. Відкриється вікно Edit Component (рис. 4.6), в якому у полі Program File вказуємо шлях до файлу BlinkLED.hex. також перевірте щоб була встановлена частота мікроконтролера така сама як і у програмному коді, тобто 8МГц.

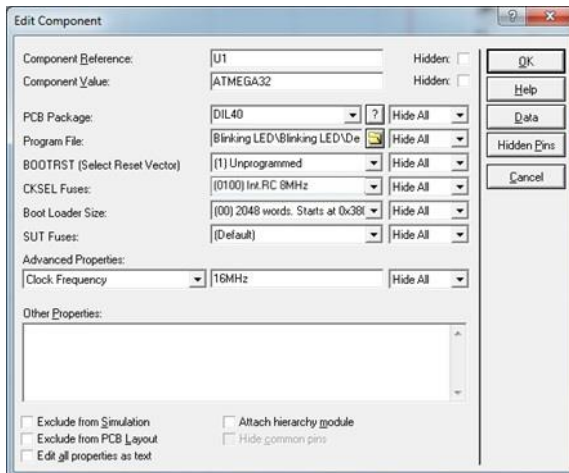
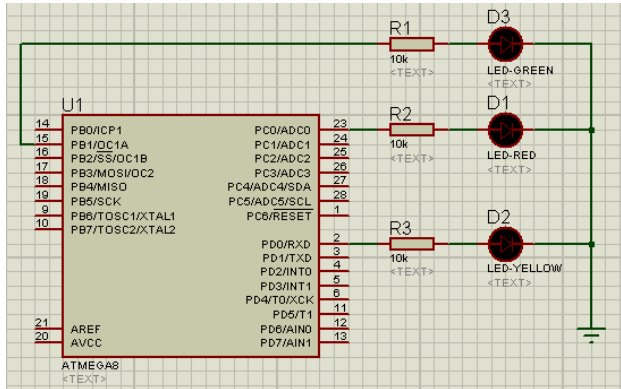


Рисунок 4.6 – Вікно Edit Component

Після на панелі керування симуляцією клікаємо на Play, і якщо все правильно було зроблено, споглядаємо роботу схеми.

2. Завдання до лабораторної роботи

3.9 Розробіть програму і виконайте симуляцію мерехтіння трьох різних світлодіодів, підключених до мікроконтролера ATmega8. Схему підключення у середовищі Proteus VSM наведено нижче:



3. Зміст звіту

- 3.1. Тема та мета роботи
- 3.2. Схема електрична-принципова
- 3.3. Код програми
- 3.4. Висновки з роботи

4. Контрольні запитання

- 4.1 Що таке Нех-файл?
- 4.2 Назвіть послідовність етапів створення нового проєкту в середовищі Atmel Studio
- 4.3 Яке сімейство мікроконтролерів використовувалося при розробці схеми у цій роботі?
- 4.4 При розробці керуючої програми, у яких одиницях вимірювання задається затримка виконання команди мікроконтролера?

ЛАБОРАТОРНА РОБОТА № 5

ПРИНЦИПИ РОБОТИ З RGB СВІТЛОДІОДОМ, КНОПКОЮ ТА ЗМІННИМ РЕЗИСТОРОМ

Мета роботи: ознайомитись з основними принципами роботи та підключення RGB світло діоду, кнопки та змінного резистора до плати Arduino.

1. Теоретичні відомості

Розглянемо роботу RGB світлодіоду з платою Arduino. Для управління кольором світлодіоду буде використовуватись функція `analogWrite()`. Якщо задіяти на платі контакти, з відміткою «~», ми можемо регулювати напругу, яке подається на відповідний світлодіод.

RGB світлодіод має 4 виводи.

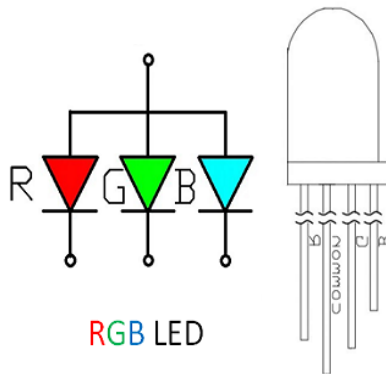


Рисунок 5.1 – RGB світлодіод

На перший погляд, RGB(Red, Green, Blue) світлодіод виглядає як звичайний регулярний світлодіод, однак всередині RGB насправді 3 світлодіоди: один червоний, один зелений і один синій.

3 виводи, кожний окремо, підключено до позитивного виводу кожного одиночного світлодіоду усередині RGB світлодіоду та один з виводів підключається до негативного, який є загальним для виводу

усіх трьох світлодіодів. До загального негативного виводу підключається заземлення (GROUND).

Шляхом контролю яскравості кожного індивідуального світлодіоду можна отримувати різні кольори, це схоже на змішування трьох кольорів фарби на палітрі. Жорсткий спосіб змінювати колір RGB світлодіоду – це використовувати різне значення резисторів.

Розглянемо невеликий проєкт з RGB світлодіодом. Для кожного світлодіоду потрібен відповідний резистор на 270 Ом, щоб запобігти можливості протікання занадто великих токів. Ці резистори встановлюються в коло між катодами (червоний, зелений и синій) та керуючими пінами на Arduino (рис. 5.2).

Якщо використовується RGB світлодіод с загальним анодом, замість загального катода, тоді загальний пін світлодіоду підключається до піна +5 V замість піна gnd.

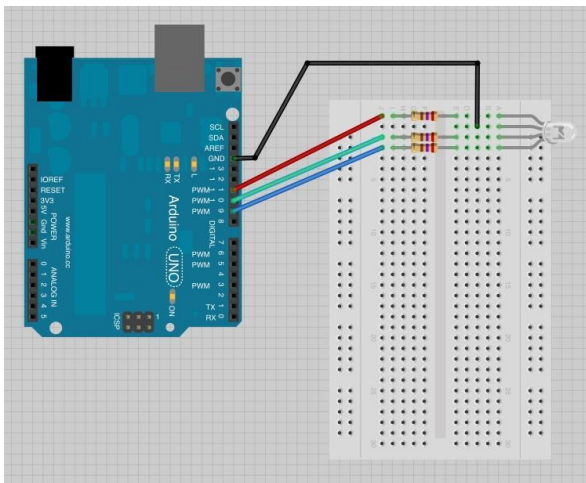


Рисунок 5.2 – Схема відключення

Але більш зручний спосіб – це використання змінного резистора.

Змінний резистор (або потенціометр) – це змінний резистор з регульованим опором. Потенціометри використовуються в робототехніці як регулятори різних параметрів – гучність звуку, потужності, напруги тощо.

Потенціометр має три контакти (рис. 5.3). Середній контакт йде на аналоговий вихід. Зовнішні контакти йдуть до піна із живленням и до землі.



Рисунок 5.3 – Зображення потенціометра

Для детального розуміння роботи потенціометра розглянемо приклад його роботи.

На рисунку 5.4 зображена схема підключення потенціометра до плати. У цьому випадку потенціометр підключений між землею і +5 V потенціалами (оскільки плата Arduino живиться +5 V), а движок сполучений з каналом аналого-цифрового перетворювача мікроконтролера. У такому разі можна регулювати вихідну напругу потенціометра в межах від 0 до 5 V.

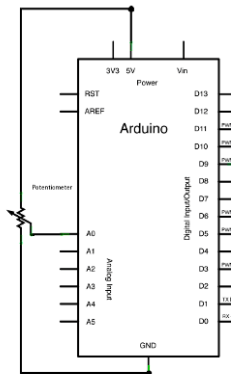


Рисунок 5.4 – Схема підключення потенціометра до плати Arduino

Код програми для роботи потенціометра:

```
void setup() {
    Serial.begin(9600);
}
```

```
void loop() {
    int sensorValue = analogRead(A0); // отримуємо
    поточне значення
    Serial.println(sensorValue, DEC); // виводимо
    результат на монітор
}
```

При ініціалізації встановлюємо потрібну швидкість зв'язку: **Serial.begin(9600);**. Далі в циклі ми постійно зчитуємо дані, що поступають з потенціометра за допомогою методу `analogRead()`. Оскільки значення знаходитимуться в діапазоні від 0 до 1023, можемо використовувати тип `int` для змінної `sensorValue`.

Отриманий результат будемо виводиться у вікно послідовного монітора в десятковому форматі.

Далі ми розглянемо роботу з кнопкою. В одній з попередніх робіт вже було розглянуте загальне визначення кнопки.

На рисунку 5.5 зображена схема підключення кнопки до піна плати Arduino.

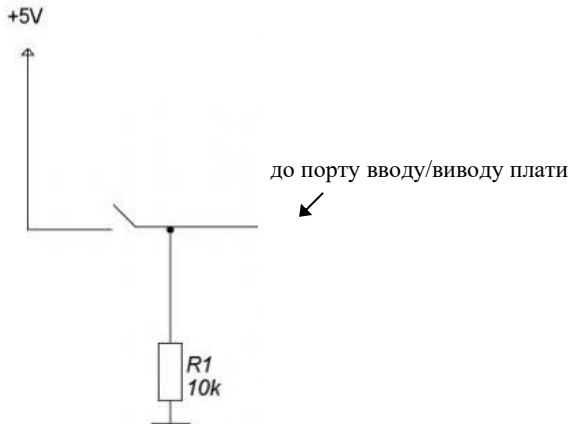


Рисунок 5.5 – Схема підключення кнопки

Пін плати на вході повинен мати стан 0 або 1. Коли стан «підвішений», тобто не визначений, на вході будуть збиратися різні зовнішні наводки (статичні, електричні, електромагнітні), що буде призводити до хибного спрацювання кнопки. Щоб підвести пін до 0 або 1 використовують підтягуючі резистори. Вони бувають двох видів,

верхньої або нижньої підтяжки. Верхні підключаються до плюса живлення, а нижні до мінуса.

Після підключення схеми переходимо до написання програми. Необхідно пін, до якого буде підключатися кнопка, ініціалізувати на вхід за допомогою команди:

`pinMode(buttonPin, INPUT);` //де buttonPin – це номер піна до якого підключена кнопка

Також необхідно об'явити змінну в яку буде записуватися статус кнопки.

2. Завдання до лабораторної роботи

2.1 Виконайте підключення до 5 порту звичайного світлодіоду. Підключіть до 6 порту кнопку. Напишіть програму, щоб при натисканні кнопки світлодіод, підключений до 5 порту, вмикався, а вбудований світлодіод до 13 порту вимикався.

2.2 При повороті ручки потенціометра змінювати яскравість світлодіоду. Схема підключення зображена на рисунку 5.6.

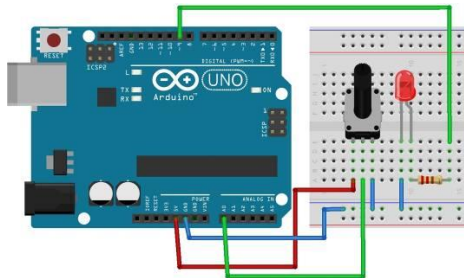


Рисунок 5.6 – Схема підключення

2.3 Підключіть RGB світлодіод до плати та по чергово змінюйте значення опору резисторів.

3. Зміст звіту

- 3.1. Тема та мета роботи
- 3.2. Схема електрична-принципова
- 3.3. Код програми

3.4. Висновки з роботи

4. Контрольні запитання

4.1 Як працює потенціометр?

4.2 До яких виводів підключається потенціометр в схемі керування на платформі Arduino

4.3 Які виводи має RGB світло діод?

4.4 Для управління кольором RGB світло діоду які використовуються функції Arduino?

4.5 З якою метою в схемі підключення кнопки до Arduino Uno використовуються підтягуючі резистори

ЛАБОРАТОРНА РОБОТА № 6

СТВОРЕННЯ ПРОЄКТУ З ВИКОРИСТАННЯМ СЕРВО

Мета роботи: ознайомитись з особливістю будови та різновидами серводвигунів, та отримати навички роботи Arduino з серводвигунами.

1. Теоретичні відомості

Сервопривід – привід з управлінням через від’ємний зворотній зв’язок, що дозволяє точно керувати параметрами руху. Сервопривід з мотором (серводвигун), призначений для приведення в рух пристроїв керування через поворот вихідного валу.

Сімейство серводвигунів різноманітне, його можна розподілити на декілька груп (рис.6.1). Відміні особливості обумовлені наступними факторами:

- конструкція двигунів (статор, ротор);
- необхідні системи регулювання;
- система зворотного зв’язку (датчики).

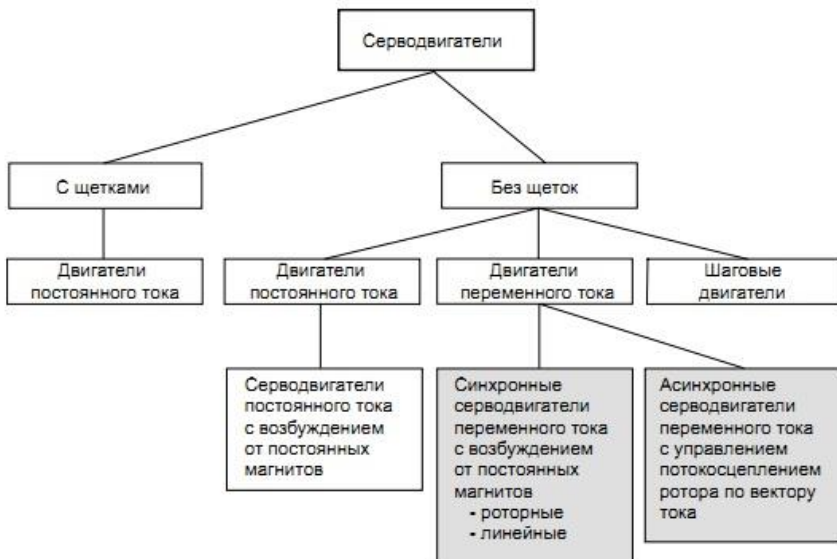


Рисунок 6.1 – Огляд серводвигунів

Будова сервоприводу для радіокерованих моделей зображена на рисунку 6.2. Він складається з електродвигуна, укладеного в один корпус з редуктором і керуючою електронікою, яка найчастіше складається з потенціометра зворотного зв'язку та плати управління. Серводвигун використовує три приводи для роботи. Плюсовий привід для живлення, зазвичай 4.8В-6В, мінусовий привід та сигнальний привід. Керуючий сигнал передає інформацію щодо положення вихідного валу. Вал пов'язаний з потенціометром, який визначає його положення. Контролер згідно опору потенціометра і значенням керуючого сигналу визначає, в який бік потрібно обертати мотор, щоб отримати потрібна положення вихідного валу.



Рисунок 6.2 – Конструкція сервоприводу

Виводи сервоприводу наступні:

Три проводи: чорний(коричневий), червоний, білий(жовтий) – відповідно, земля, живлення та керуючий сигнал (ШІМ) .

Керування сервоприводом відбувається шляхом подачі на нього прямокутних імпульсів по сигнальному виводу з частотою біля 50Гц, амплітудою не менше 3.3В (часто не менше 4.8В), шириною стандартно від 1000 до 2000мкс, що відповідає стандартним крайнім положенням (див. рис. 6.3). Зазвичай реальний діапазон може бути трохи ширше (наприклад, 900-2100мкс), але це вже залежить конкретно від виробника сервоприводів.

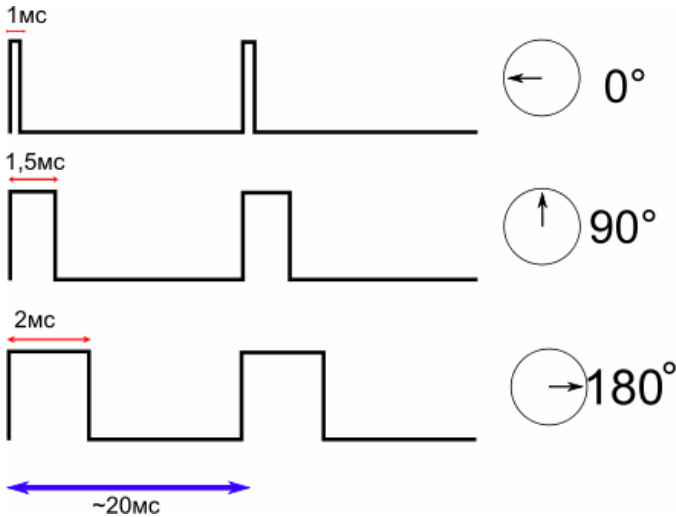


Рисунок 6.3 – Керуючі сигнали

2. Виконання роботи

2.1 Створення сервоприводу у Proteus

Ми перевіримо, як спроектувати сервопривід у Proteus ISIS, а також спроектуємо керуючий привід постійного струму за допомогою логічних елементів. З початку ми створимо двигун постійного струму, який присутній в Proteus і дуже простий у використанні. Ми будемо керувати їм шляхом подачі напруги на його обидві сторони, тобто прямим методом.

Двигун постійного струму являє собою простий двигун, який потребує різної полярності на своїх двох виводах. Якщо ця полярність в прямому напрямку, то двигун постійного струму рухається в одному напрямку, і якщо ми змінимо полярність то двигун буде рухатися в протилежному напрямку. Отже, створимо DC Motor Drive Circuit в Proteus ISIS.

Простий DC Motor Drive Circuit в Proteus ISIS

Виберіть компоненти з бібліотеки Proteus:

1. Motor
2. Logic State (рис. 6.4). Logic State має два стани 1 і 0.

Коли 0, то це означає 0 V і коли, то 1 - 5V.

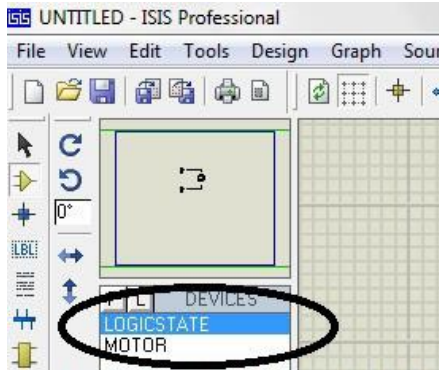


Рисунок 6.4 – Вибір компонентів

Тепер розробіть схему, як показано на рисунку 6.5.

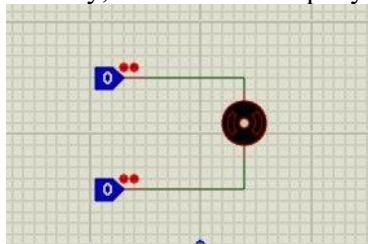


Рисунок 6.5 – Схема з однаковими логічними станами

Як бачимо на схемі додано два логічні стан на обох сторонах двигуна. Напрямок двигуна буде залежати від цієї логіки. Таким чином, їх буде всього чотири стани:

1. Коли обидва стани 0, двигун не буде рухатися і залишатися нерухомими.
2. Коли обидва стани знаходяться на 1, також двигун не буде рухатися і залишатися нерухомими.
3. Двигун буде рухатися за годинниковою стрілкою, коли верхній стан 1, а нижній 0.
4. Двигун буде рухатися проти годинникової стрілки, коли верхній стан 0, а нижній 1 (рис. 6.6).

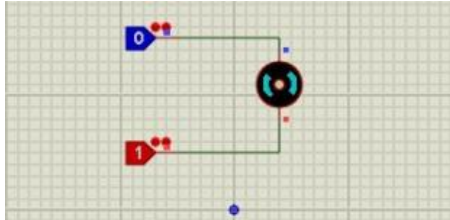


Рисунок 6.6 – Схема з різними логічними станами

Керування сервоприводом у Proteus ISIS за допомогою ШІМ

Для керування сервоприводом нам потрібно формувати **ШІМ** з частотою 50 Герц. При цьому для положення “0” довжина імпульсу повинна складати 1000 мікросекунд, а для положення “максимум” – 2000 мікросекунд. Середнє положення – 1500 мікросекунд (рис. 6.3). Розглянемо це конкретно на прикладі.

Відкриваємо Proteus ISIS та створюємо новий проект. У бібліотеці елементів знаходимо сервопривід та розміщуємо так як зображено на рисунку 6.7. На ці три сервоприводи ми будемо відповідно надсилати сигнали різної довжини: 1мс, 1.5мс та 2 мс.

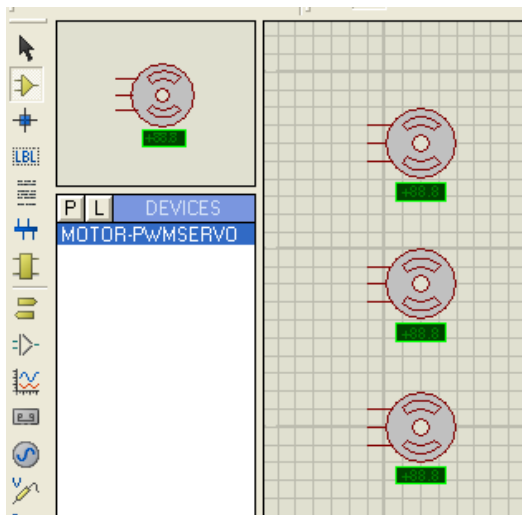


Рисунок 6.7 – Схема з декількома сервоприводами

Встановлюємо елемент живлення «POWER» та у його властивостях задаємо живлення «+12V» (рис.6.8).

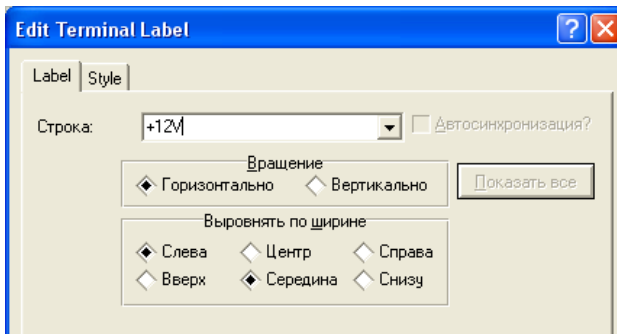


Рисунок 6.8 – Встановлення живлення

Після встановлення заземлення електрична схема буде виглядати так як зображено на рисунку 6.9.

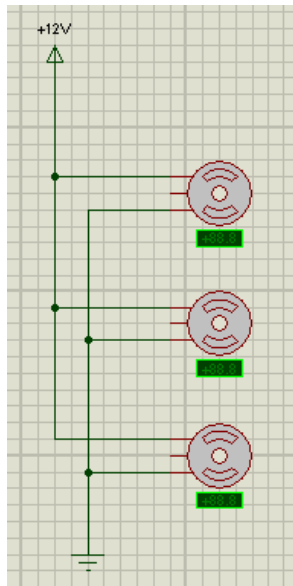


Рисунок 6.9 – Встановлення заземлення

Для генерування сигналів потрібної частоти та довжини встановимо елементи, які будуть генерувати три різні сигнали на кожен з сервоприводів. Перейдіть у лівій панелі інструментів у режим Generator Mode та виберіть компонент PULSE. Встановіть три компоненти PULSE до схеми та з'єднайте їх як зображено на рисунку 6.10.

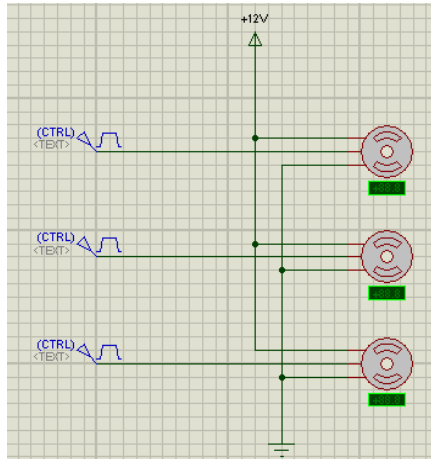


Рисунок 6.10 – Схема з компонентами PULSE

Для першого зверху генератора імпульсу встановіть параметр довжини (ширина) імпульсу 1мс та частота 50Гц (див. рис. 6.11).

Для двох наступних за аналогією з попереднім встановіть довжину імпульсу 1.5мс та 2мс відповідно. Як схема буде завершена, запустіть симуляцію. У результаті симуляції верхній сервопривід повернеться у крайнє ліве положення, середній встановить нульове положення, а нижній повернеться у крайнє праве положення, як зображено на рисунку 6.12.

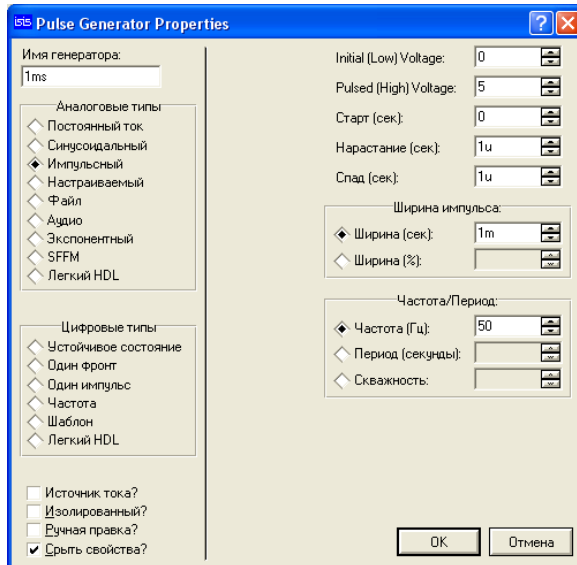


Рисунок 6.11 – Встановлення параметрів для першого генератора імпульсів

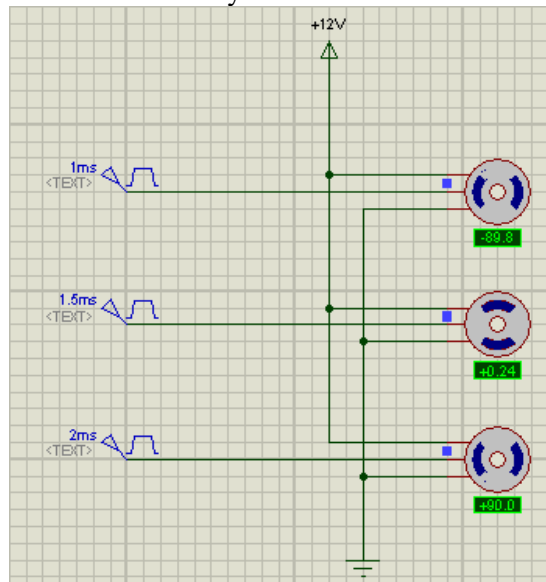


Рисунок 6.12 – Результат симуляції

2.2 Створення проєкту з сервоприводом на Arduino

Для розглядання прикладу керування положенням сервоприводу за допомогою Arduino необхідна плата та сервопривід. Схема підключення зображена на рисунку 6.13. Як вже було сказано вище серводвигун має 3 приводи: живлення, заземлення, сигнальний. Дріт живлення, як правило, червоний, і повинен бути з'єднаний з піном 5V на платі Arduino. Дріт заземлення, як правило, чорного або коричневого кольору і повинен бути підключений до заземлення на платі Arduino. Сигнальний контакт, як правило, жовтого, білого або помаранчевого кольору, і повинен бути підключений до контакту 9 на платі Arduino.

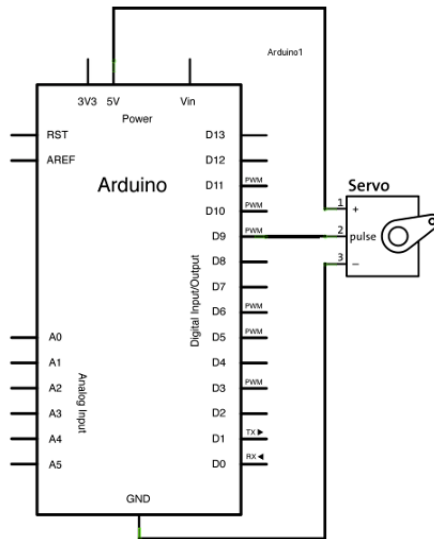


Рисунок 6.13 – Схема підключення сервоприводу до Arduino

Для написання програми у середовищі Arduino є спеціальна бібліотека для роботи з сервоприводом «Servo.h». Далі наведено простий приклад програми роботи з сервоприводом:

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

Servo myservo; // створення об'єкта сервоприводу для курування серво; максимум можна створити 8 об'єктів сервоприводів

int pos = 0; // змінна для збереження положення сервоприводу

void setup()

```
{
  myservo.attach(9); // призначити серво який підключений до 9 піна до
  об'єкту сервоприводу
}
```

void loop()

```
{
  for(pos = 0; pos < 180; pos += 1) // рух від 0 до 180 градусів
  {
    // с кроком в 1 градус
    myservo.write(pos); // дає команду сервоприводу перейти на
    позицію 'pos'
    delay(15); // затримка 15мс для того щоб серво досягла
    необхідної позиції
  }
  for(pos = 180; pos >= 1; pos -= 1) // рух від 180 до 0 градусів
  {
    myservo.write(pos); // дає команду сервоприводу перейти на
    позицію 'pos'
    delay(15); // затримка 15мс для того щоб серво досягла
    необхідної позиції
  }
}
```

3. Завдання до лабораторної роботи

3.1 Підключити кнопку і сервопривід до плати Arduino. Написати програму, щоб при натисканні кнопки сервопривід рухався за годинниковою стрілкою, а при відпусканні кнопки сервопривід зупинявся.

3.2 Підключити кнопку і сервопривід до плати Arduino. Написати програму, щоб при натисканні кнопки сервопривід рухався проти

годинникової стрілки, а при відпусканні кнопки сервопривід зупинявся.

4. Зміст звіту

- 4.1. Тема та мета роботи
- 4.2. Схема електрична-принципова
- 4.3. Код програми
- 4.4. Висновки з роботи

5. Контрольні запитання

- 5.1 Що таке сервопривід і які різновиди сервоприводів існують?
- 5.2 За якими ознаками відрізняються сервоприводи?
- 5.3 Як відбувається керування сервоприводом керуючими сигналами?
- 5.4 До яких виводів підключається сервопривід до платформи Arduino?
- 5.5 Назвіть основні команди Arduino для керування сервоприводом?

ЛАБОРАТОРНА РОБОТА № 7

СТВОРЕННЯ ПРОЄКТУ З ВИКОРИСТАННЯМ РЕЛЕ

Мета роботи: ознайомитись з особливістю будови та реле, та отримати навички роботи Arduino з серводвигунами.

1. Теоретичні відомості

Реле є найкращим способом управління будь-яким двигуном. У цій роботі ми будемо розбиратися як контролювати реле в Proteus ISIS. Реле є одним з ключових компонентів будь-якого електронного або електричного кола і, як правило, є проблемою при проєктуванні.

Спочатку змодельуємо реле в простій схемі, в якій при запуску моделювання, реле автоматично активується, а після цього будемо керувати реле за допомогою логіки, тобто, коли ви подаєте + 5V до нього, то реле буде активуватися і коли ви даєте Gnd, то він буде знеструмлений. Крім того, розглянемо, що таке реле і як його використовувати?

Для управління різними виконавчими пристроями, комутації ланцюгів, управління приладами в електроніці активно застосовуються електромагнітні реле (рис.7.1).



Рисунок 7.1 – Зовнішній вигляд електромагнітного реле

Будова реле досить проста. Його основою є котушка, що складається з великої кількості витків ізольованого проводу. Всередину котушки встановлюється стрижень з м'якого заліза. В результаті виходить електромагніт. Також в конструкції реле присутній якір. Він закріплений на пружному контакті. Сам же пружний контакт закріплений на ярмі. Разом зі стрижнем і якорем ярмо утворює магнітопровід.

Якщо котушку підключити до джерела струму, то магнітне поле, що утворилося намагнічує сердечник. Він в свою чергу притягує якір. Якір укріплений на пружному контакті. Далі пружний контакт замикається з іншим нерухомим контактом. Залежно від конструкції реле, якір може по-різному механічно управляти контактами.

Внутрішня будова електромагнітного реле. У більшості випадків реле монтується в захисному корпусі. Він може бути як металевим, так і пластмасовим. Подивімося на те, що знаходиться всередині цього реле (рис.7.2).



Рисунок 7.2 – Внутрішня будова реле

Це реле без захисного корпусу. Як бачимо, реле має котушку, стрижень, що пружний контакт, на якому закріплений якір, а також виконавчі контакти.

На принципових схемах електромагнітне реле позначається наступним чином (рис.7.3).

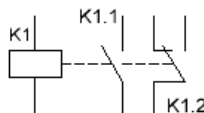


Рисунок 7.3 –Принципова схема електромагнітного реле

Умовне позначення реле на схемі складається наче з двох частин. Одна частина (**K1**) – це умовне позначення електромагнітної котушки. Вона позначається у вигляді прямокутника з двома виводами. Друга частина (**K1.1**; **K1.2**) – це групи контактів, якими управляє реле. Залежно від своєї складності реле може мати досить велику кількість комutowаних контактів. Вони розбиваються на групи. Як бачимо, на позначенні зображені дві групи контактів (**K1.1** і **K1.2**).

Як тільки ми замкнемо керуючий ланцюг вимикачем **SA1**, струм від батареї живлення **G1** надійде на реле **K1**. Реле спрацює, і його контакти **K1.1** замкнуть виконавчий ланцюг. На навантаження надійде напруга живлення від батареї **G2** і лампа **HL1** засвітиться. Якщо розімкнути ланцюг вимикачем **SA1**, то з реле **K1** буде знята напруга живлення та контакти реле **K1.1** знову розімкнуться і лампа **HL1** вимкнеться.

Комutowані контакти реле можуть мати своє конструктивне виконання. Так, наприклад, розрізняють нормально-розімкнуті контакти, нормально-замкнуті контакти та контакти на перемикання (перекидні). Розберемося з цим детальніше.

Нормально розімкнуті контакти – це контакти реле, які знаходяться в розімкнутому стані до тих пір, поки через котушку реле не потече струм. Говорячи простіше, коли реле вимкнено, контакти теж розімкнуті. На схемах реле з нормально-роз'єднаними контактами позначається як на рисунку 7.4.

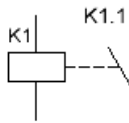


Рисунок 7.4 – Схема реле з нормально розімкнутими контактами

Нормально замкнуті контакти – це контакти реле, що знаходяться в замкнутому стані, поки через котушку реле не почне текти струм. Таким чином, виходить, що при вимкненому реле контакти замкнуті. Такі контакти на схемах зображують як на рисунку 7.5.

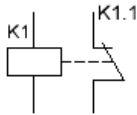


Рисунок 7.5 – Схема реле з нормально зімкнутими контактами

Перекидні контакти – це комбінація з нормально-замкнутих і нормально-розімкнутих контактів. У перекидних контактів є спільний провід, який переключається з одного контакту на інший (рис.7.6).

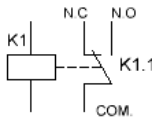


Рисунок 7.6 – Схема реле з перекидними контактами

Сучасні широко розповсюдженні реле, як правило, мають перекидні контакти, але можуть зустрічатися і реле, які мають у своєму складі тільки нормально-розімкнуті контакти.

У імпортних реле нормально-розімкнуті контакти реле позначаються скороченням **N.O.** А нормально-замкнуті контакти **N.C.** Загальний контакт реле має скорочення **COM.** (від слова *common* – «загальний»).

Параметри електромагнітних реле. Як правило, розміри самих реле дозволяють наносити на корпус їх основні параметри. Як приклад, розглянемо імпортне реле Bestar BS-115C. На його корпусі нанесено параметри реле (рис. 7.7).



Рисунок 7.7 – Вигляд реле

COIL 12VDC – це номінальна напруга спрацьовування реле (12V). Оскільки це реле постійного струму, то вказано скорочене позначення постійної напруги (скорочення DC позначає постійний струм/напруга). Англійське слово **COIL** перекладається як «котушка», «соленоїд». Воно вказує на те, що скорочення 12VDC має відношення до котушки реле.

Далі на реле вказані електричні параметри його контактів. Зрозуміло, що потужність контактів реле може бути різною. Це залежить як від габаритних розмірів контактів, так і від використовуваних матеріалів. При підключенні навантаження до контактів реле потрібно знати потужність, на яку вони розраховані. Якщо навантаження споживає потужність більше тієї, на яку розраховані контакти реле, то вони будуть нагріватися, іскрити, “залипати”. Природно, це призведе до швидкого виходу з ладу контактів реле.

Для реле, як правило, зазначаються параметри змінного і постійного струму, які здатні витримати контакти.

Так, наприклад, контакти реле Bestar BS-115C здатні комутувати змінний струм в 12A і напругу 120V. Ці параметри зашифровані в написі **12A 120VAC** (скорочення AC позначає змінний струм).

Також реле здатне комутувати постійний струм силою 10A і напругою 28V. Про це свідчить напис **10A 28VDC**. Це були силові характеристики реле, точніше його контактів.

Споживана потужність реле. Тепер звернемося до потужності, яку споживає реле. Як відомо, потужність постійного струму дорівнює добутку напруги (U) на струм (I): $P = U * I$. Візьмемо значення номінальної напруги спрацьовування (12V) і споживаного струму (30 mA) реле Bestar BS-115C і отримаємо його споживану потужність (англ. – **Power consumption**).

Таким чином, потужність реле Bestar BS-115C становить 360 міліВат (mW).

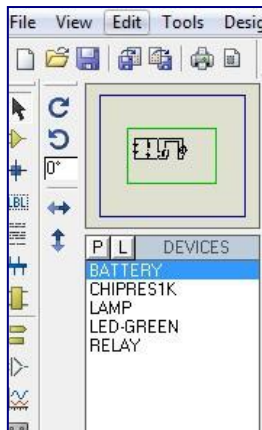
Є ще один параметр – це чутливість реле. За своєю суттю, це і є потужність споживання реле у включеному стані. Зрозуміло, що реле, якому потрібно менше потужності для спрацьовування, є більш чутливим у порівнянні з тими, які споживають велику потужність. Такий параметр, як чутливість реле, особливо важливий для пристроїв з автономним живленням, так як включене реле витрачає заряд батареї. Приміром, є два реле з споживаною потужністю 200 mW і 360 mW.

Таким чином, реле потужністю 200 mW має більшу чутливість, ніж реле потужністю 360 mW.

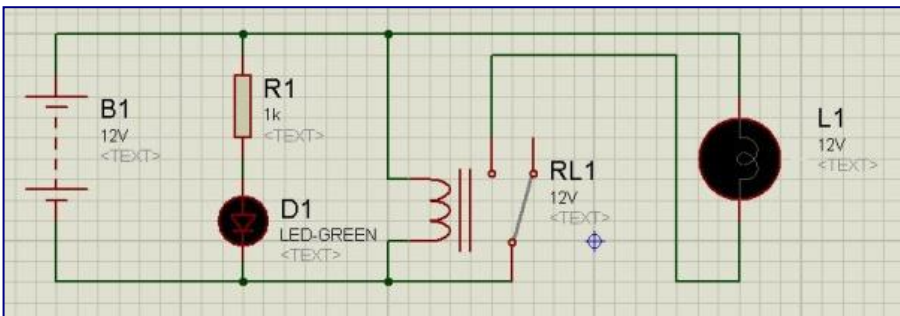
2. Виконання роботи

Проста схема підключення Реле в Proteus ISIS

- відкрити Proteus ISIS и вибрати компоненти, які зображені на рисунку нижче, з бібліотеки компонентів Proteus



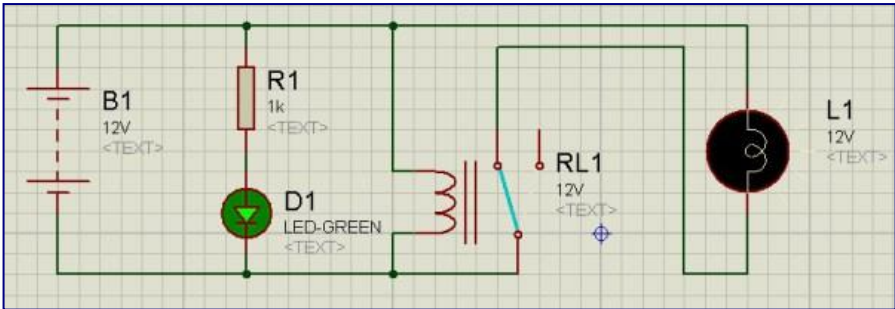
- встановить та з'єднайте компоненти у схему, яка зображена нижче:



- коли реле буде під напругою, лампа L1 буде світитись і коли

реле знеструмлене, лампа залишиться вимкненою. Світлодіод D1 виконує роботу індикатора, який свідчитиме, що напруга в колі є.

- після розробки схеми, натисніть на кнопку запуску, і якщо все зроблено правильно, то лампа буде світитись:



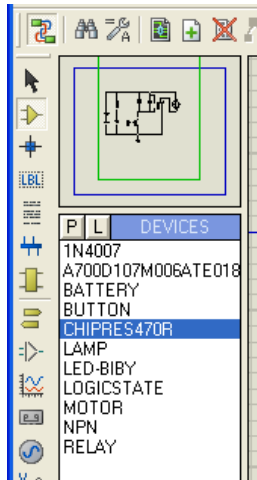
- Отже, тепер ви можете побачити світлодіодний індикатор також загоряється, та бачимо, що реле тепер з'єднаний з другим терміналом і, таким чином замикає ланцюг для лампи і лампи тепер також світиться.

Керування реле за допомогою логіки в Proteus ISIS

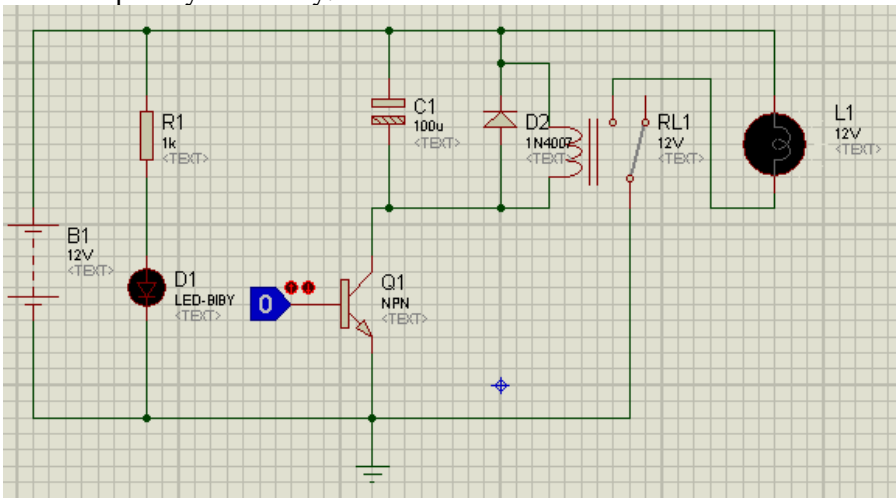
- У попередньому розділі ми бачили просту схему, яка управляється вручну це означає, що для того, щоб включити або виключити його потрібно включити або виключити живлення, але, як правило, потрібно, щоб реле знаходилося під автоматичним контролем якогось мікроконтролера.

- Зважаючи на те що мікроконтролери зазвичай працюють від 5V для того, щоб управляти реле 12V з використанням 5V мікроконтролера, ми повинні використовувати транзистор. У цьому випадку, коли ви даєте + 5V реле приводиться в дію, а коли ви даєте GND реле вимикається.

- Таким чином, в першу чергу знайдемо вказані компоненти:



• Проектуємо схему:

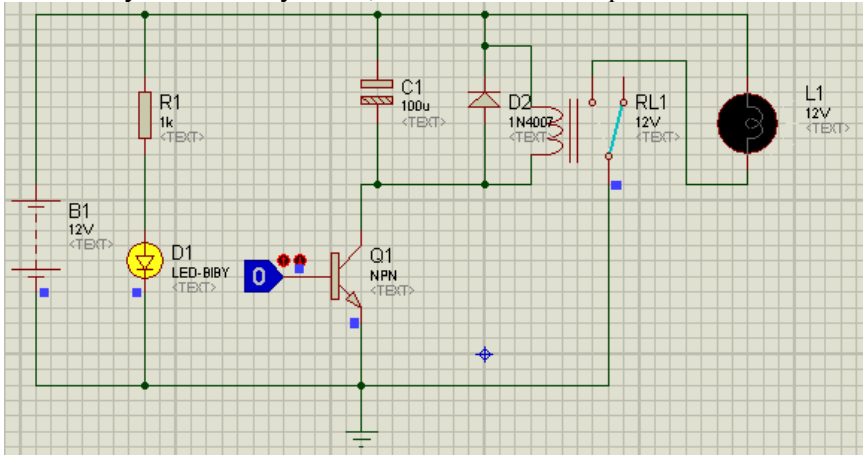


• Оскільки ми не використовуємо мікроконтролер, замість нього будемо використовувати логічний стан, він буде працювати так само, як мікроконтролер. Таким чином, схема дуже схожа на просту схему, яку ми вже робили в попередньому розділі. Єдина відмінність тут присутній n-p-n транзистор.

• У цьому випадку ми не забезпечуємо подачу живлення безпосередньо до реле, замість цього ми забезпечуємо його за

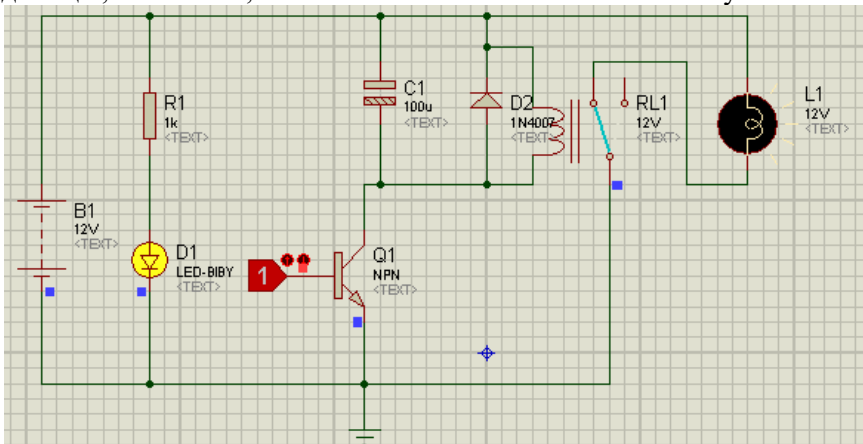
допомогою цього транзистора. Так що, коли логічний стан дорівнює нулю (це означає землю), транзистор не працюватиме, і живлення не може досягти реле і коли ми робимо логічну 1, що означає + 5V на вході транзистора, то схема реле буде завершена і реле буде запитана.

- Запускаємо симуляцію, вимкнений стан реле виглядає так:



• На наведеному вище рисунку можемо побачити, що світлодіодний індикатор працює тому, що живлення подається на схему, але лампа все ще вимкнена і на реле також не подається живлення, так як логічне стан 0.

• Тепер натисніть на логічному стані та зробимо логічну одиницю, тобто +5V, включений стан показано на малюнку нижче:



Робота Arduino з реле

Реле модуль має 3 виводи:

VCC: "+"

GND: "-"

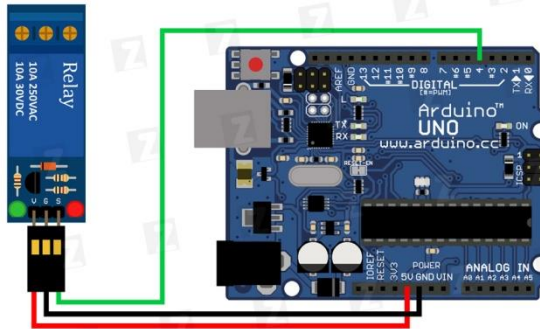
IN: вивід вхідного сигналу

Підключення модулю дуже просте:

VCC на + 5 вольт на Arduino.

GND на любий з GND пінів Arduino.

IN на любий з цифрових входів/виходів Arduino.



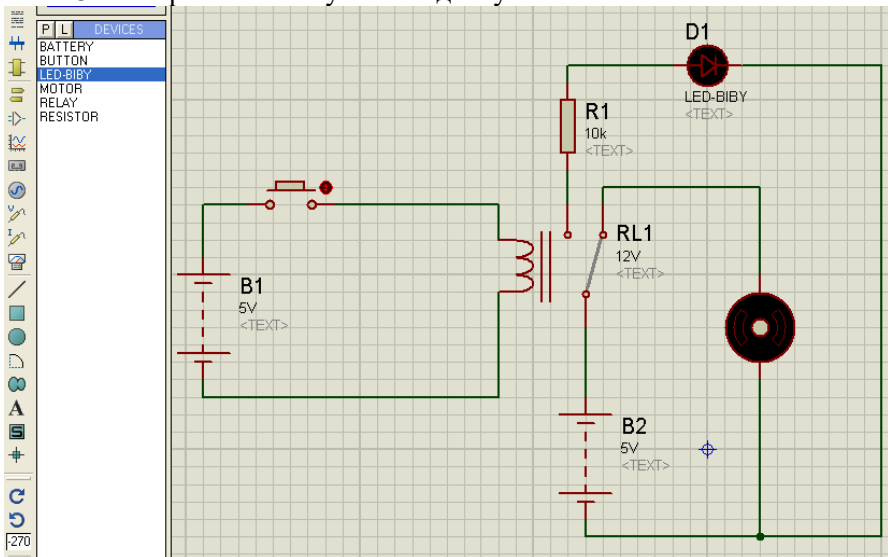
Переходимо до створення програми. У даному прикладі реле буде включатись и виключатись з інтервалом у 2 секунди.

Приклад програмного коду:

```
// Реле модуль підключено до цифрового виводу 4
int Relay = 4;
void setup()
{
    pinMode(Relay, OUTPUT);
}
void loop()
{
    digitalWrite(Relay, LOW);    // реле вкл.
    delay(2000);
    digitalWrite(Relay, HIGH);  // реле викл.
    delay(2000);
}
```

3. Завдання до лабораторної роботи

3.1 Накресліть схему та знайдіть у ній помилки



3.2 Створіть проєкт на Arduino з використанням реле у TINKERCAD.

4. Зміст звіту

- 4.1. Тема та мета роботи
- 4.2. Схема електрична-принципова
- 4.3. Код програми
- 4.4. Висновки з роботи

5. Контрольні запитання

- 5.1 Що таке реле? Яка будова реле?
- 5.2 Які бувають комутативні контакти реле?
- 5.3 Принципи роботи транзистора?

ЛАБОРАТОРНА РОБОТА № 8

РОБОТА З БІБЛІОТЕКОЮ ARDUINO У PROTEUS.

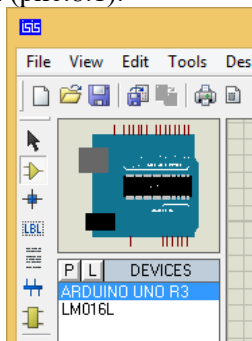
ПРОГРАМУВАННЯ LCD ДИСПЛЕЯ

Мета роботи: розробити проєкт схеми з підключенням плати Arduino у середовищі Proteus VSM. Ознайомитись з принципами програмування LCD дисплея.

1. Виконання роботи

У папці Arduino Lib знаходяться два бібліотечних файли Arduino (ARDUINO.IDX, ARDUINO.LIB). Вставте ці два файли у бібліотечну папку Proteus, яка знаходиться: C: -> Program Files (x86) -> Labcenter Electronics -> Proteus 7 Professional -> LIBRARY

Відкриваємо Proteus ISIS та знаходимо у бібліотеці плату Arduino та LCD дисплей LM016L (рис.8.1).



Рисунк 8.1 – Вибір компонентів

З'єднуємо виходи Arduino з дисплеєм таким чином:

- LCD RS вихід до 12 виходу Arduino
- LCD Enable до 11
- LCD D4 до 5
- LCD D5 до 4
- LCD D6 до 3
- LCD D7 до 2
- LCD RS до терміналу GROUND.

Далі відкриваємо Arduino IDE. Для того щоб бачити більш детально куди зберігаються файли з розширенням *.hex, який нам знадобиться для симуляції роботи Arduino у Proteus, перейдіть Файл->Налаштування. Та налаштуйте Arduino IDE, щоб показувало докладний звіт при компіляції, поставивши прапорець у потрібному місті (рис.8.2).

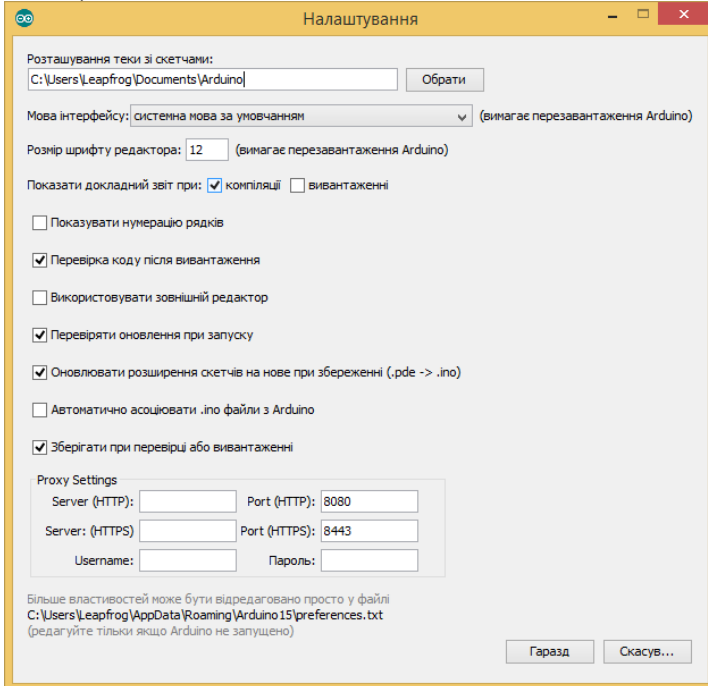


Рисунок 8.2 – Налаштування Arduino IDE

Копіюємо код приведений нижче, зберігаємо та компілюємо:

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the
// interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
```



```

// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting
  begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}

```

У вікні результатів компіляції (рис.8.3) знаходимо папку, до якої зберігається файл з розширенням *.hex.

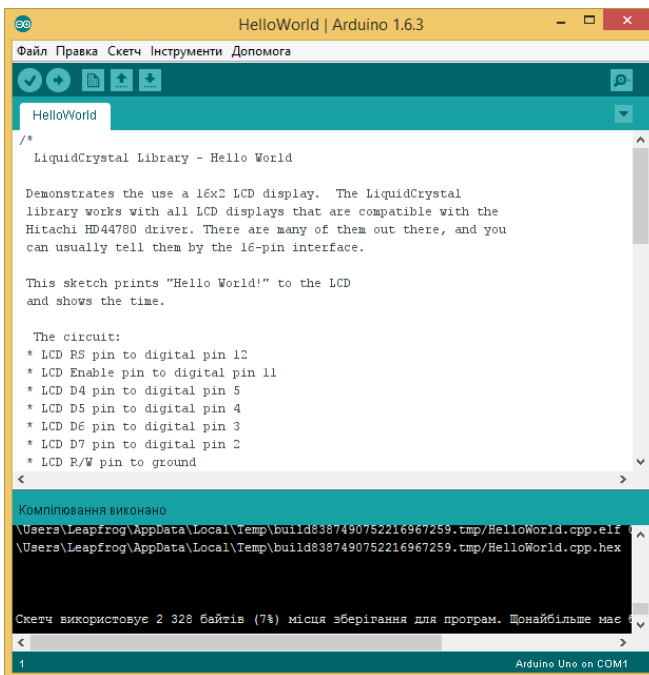


Рисунок 8.3 – Вікно результатів компіляції

Переходимо до цієї папки та копіюємо два файли, що приведені нижче до своєї папки (рис. 8.4).

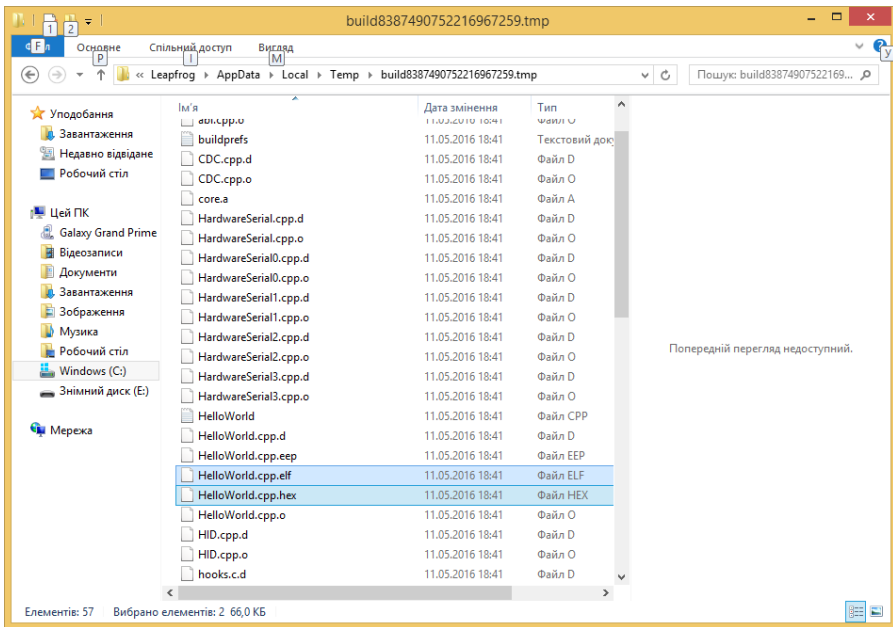


Рисунок 8.4 – Копіювання двох файлів

Для Arduino Uno у Proteus вказуємо шлях до файлу з розширенням *.hex. Запускаємо симуляцію і дивимось результат.

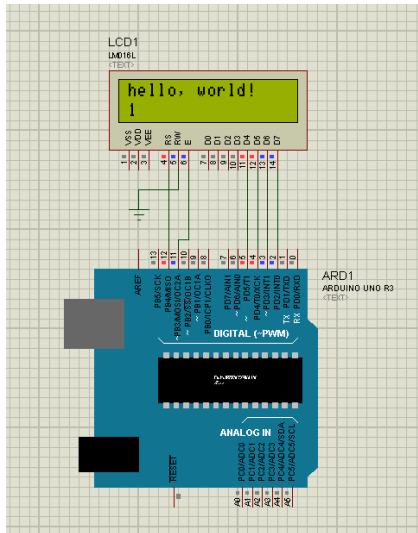


Рисунок 8.5 – Результат компіляції

2. Завдання до лабораторної роботи

2.1 Зробити симуляцію роботи з дисплеєм в TINKERCAD <https://www.tinkercad.com/>.

2.2 Зареєструватися на <http://swed.zntu.edu.ua> та провести віддалений експеримент з дисплеєм.

3. Зміст звіту

- 3.1. Тема та мета роботи
- 3.2. Схема електрична-принципова
- 3.3. Код програми
- 3.4. Висновки з роботи

4. Контрольні запитання

- 4.1 Основні функції Arduino для роботи з дисплеєм?
- 4.2 Які піни на платі Arduino використовуються для створення проєкту роботи з дисплеєм?
- 4.3 Що таке LCD I2C модуль?

ЛАБОРАТОРНА РОБОТА № 9 ПРОЄКТУВАННЯ ДРУКОВАНОЇ ПЛАТИ З ВИКОРИСТАННЯМ СРЕДОВИЩА PROTEUS ARES

Мета роботи: спроектувати схему таймера 555, зробити розводку доріжок для РСВ плати використовуючи стандартні функції ARES та згенерувати 3D візуалізацію плати.

1. Виконання роботи

555 це серія таймера, яка стала однією з перших інтегральних мікро складань. Вона несе в собі близько 20 транзисторів і використовується для роботи в двох режимах. У режимі безпосередньо таймера і генератора прямокутних імпульсів.

На 555 серії існує величезна кількість схем як для новачків радіоаматорів, так і для фахівців. На основі цього таймера можна зробити саморобні сигналізації, датчики, сирени, генератори, перетворювачі напруги, високовольні пристрої підсилювачі потужності звукової частоти і т.п.

У цій лабораторній роботі буде створюватися приклад таймера, який генерує імпульси з заданою частотою. Діапазон частот, що генерується таймером, досить широкий: від найнижчої частоти, період якої може досягати кількох годин, до частот в кілька десятків кілогерц.

Відкриваємо Proteus ISIS і починаємо створювати проєкт таймера. Виберіть з бібліотеки елементи (рис. 9.1).

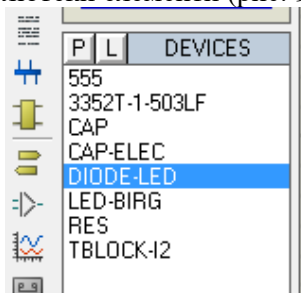


Рисунок 9.1 – Вибір елементів

Спроектуйте схему, як показано на рис. 9.2, і проведіть симуляцію її роботи. Світлодіод буде відображати поява імпульсу на виході таймера.

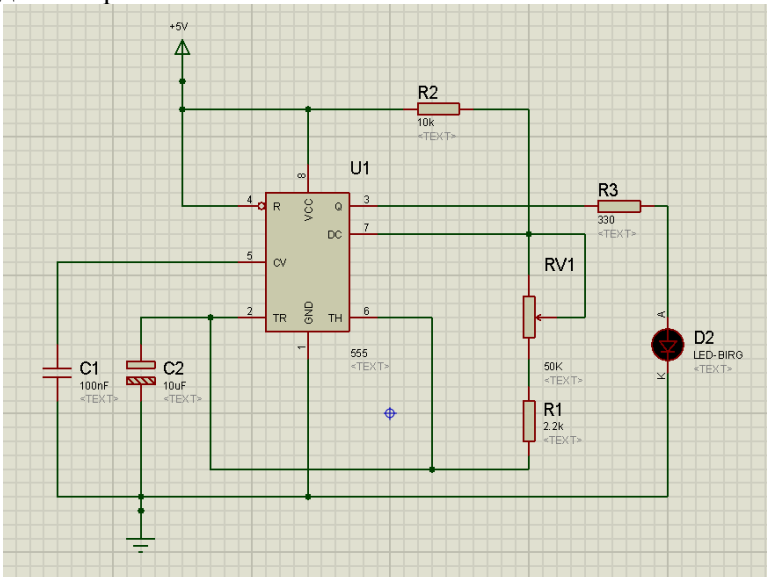


Рисунок 9.2 – Початкова схема для симуляції

Після того як переконаєтесь, що схема працює, замініть світлодіод (DIODE-LED) та встановіть роз'єм (TBLOCK-I2) (рис. 9.3).

Перевіримо наявність посадкових місць (footprint) для всіх елементів плати. Для цього подвійним кліком натискаємо по елементу і в рядку PCB Footprint дивимося, щоб була вказана бібліотека, в якій знаходиться цей footprint.

Якщо в полі PCB Footprint написано (Not specified), як показано на рисунку 9.4, то тоді натискаємо на «знак питання».

У вікні вводимо ключове слово **led**, за яким буде виконуватись пошук всіх посадкових місць в усіх бібліотеках, підключених до Proteus.

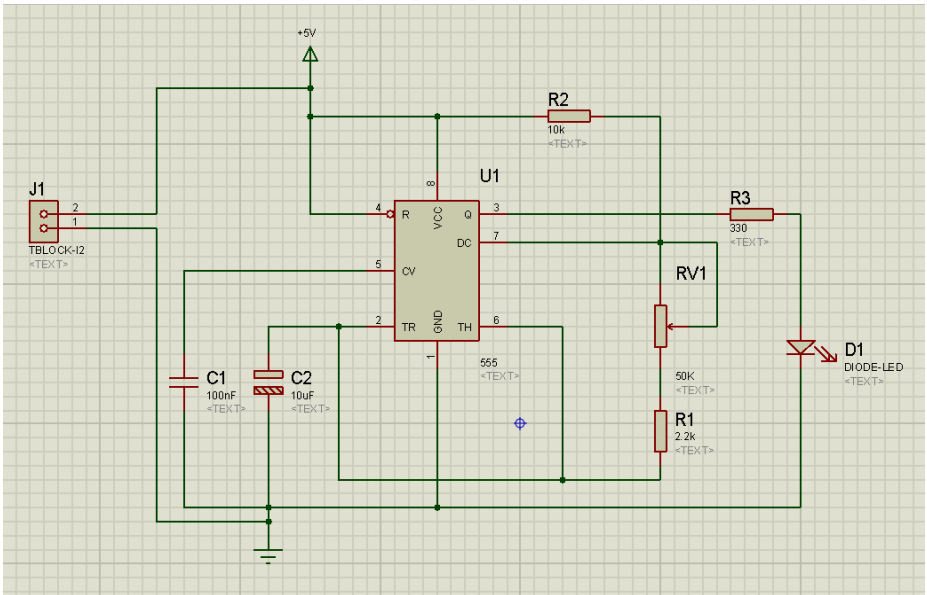


Рисунок 9.3 – Модифікована схема для симуляції

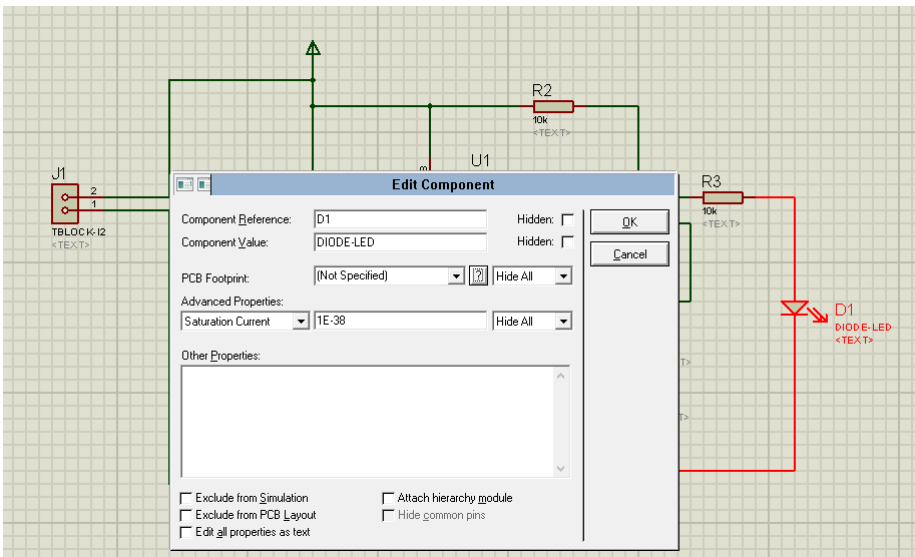


Рисунок 9.4 – Перевірка наявності посадкових місць

Вводимо в поле ключових слів елемент **LED** як показано на рисунку 9.5. Клікаємо **OK**.

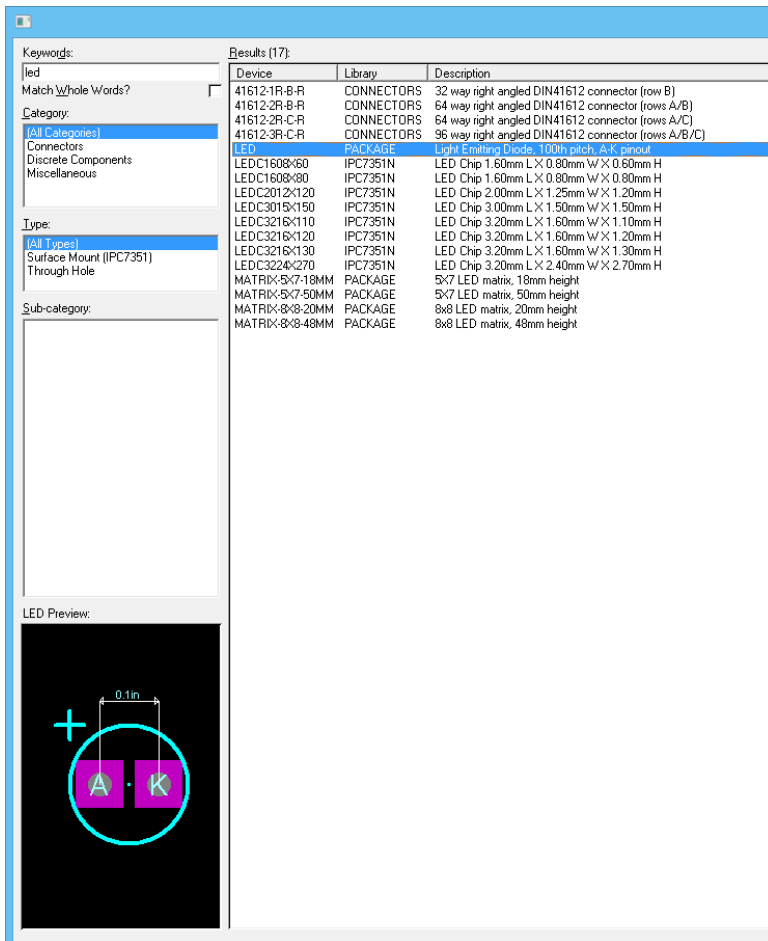


Рисунок 9.5 – Введення елемента **LED**

Далі в електричній схемі клікаємо ПКМ по елементу світлодіоду, і з випадючого списку вибираємо **Packaging Tool** (див.рис. 9.6).

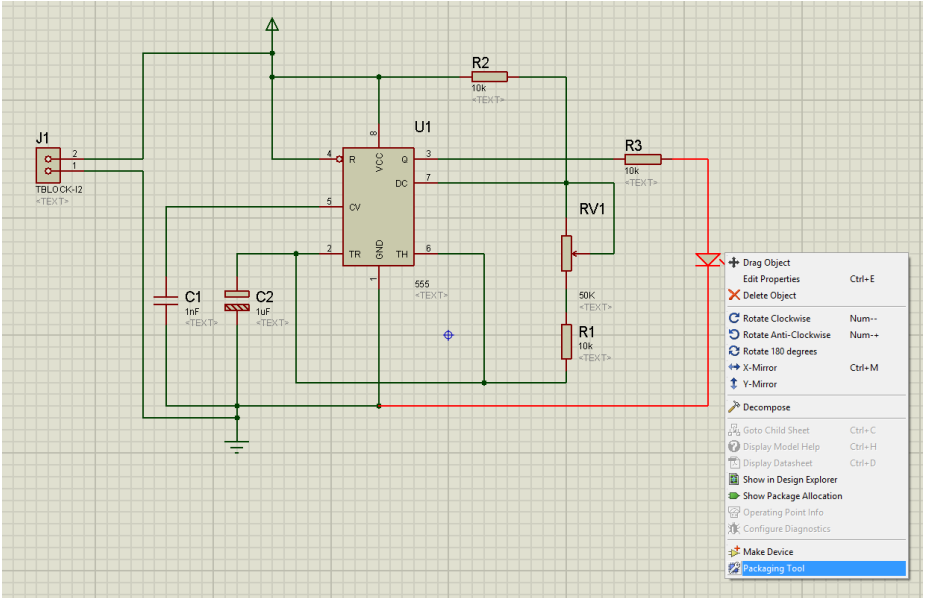


Рисунок 9.6 – Вибір інструменту Packaging Tool

У вікні **ISIS Information** (рис.9.7) клікаємо **OK**.

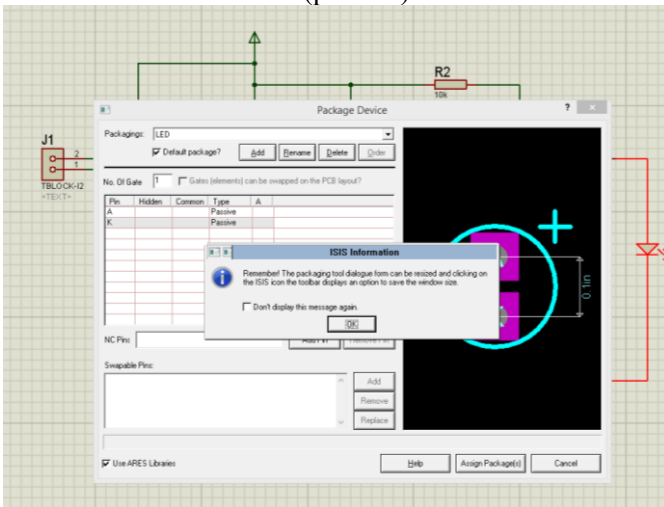


Рисунок 9.7 – Вікно ISIS Information

У вікні Package Device вказуємо відповідність виводів світлодіоду і пінів посадкового місця як показано на рис. 9.8. Після натискаємо **Assign Package(s)**. У вікні що відкрилось (рис. 9.9) обираємо **Save Package(s)**. YES

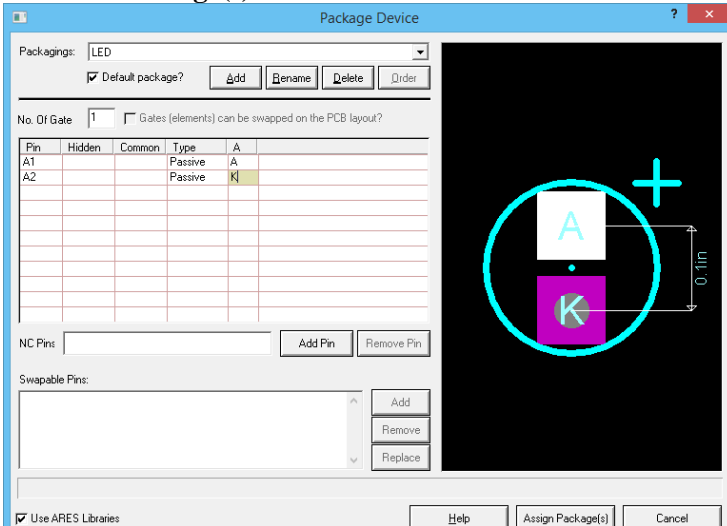


Рисунок 9.8 – Вікно Package Device

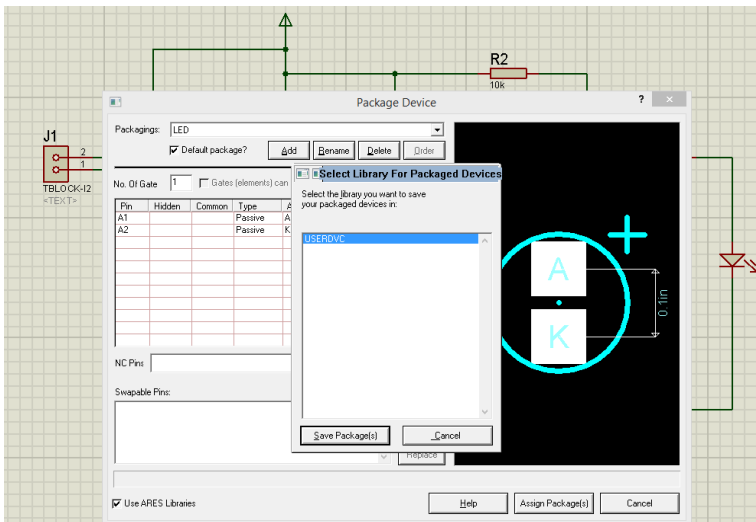


Рисунок 9.9 – Вибір Save Package(s)

На верхній панелі інструментів натискаємо **Tools->Netlist to ARES**. У лівій панелі інструментів обираємо **Component Mode**.

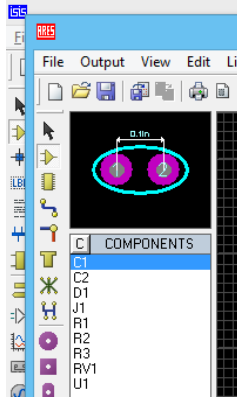


Рисунок 9.10 – Вибір Component Mode

Щоб задати контури самої плати в лівій панелі інструментів клікаємо по елементу **2D Graphics Box Mode** і після цього в нижній панелі інструментів обираємо **Board Edge** (рис. 9.11).

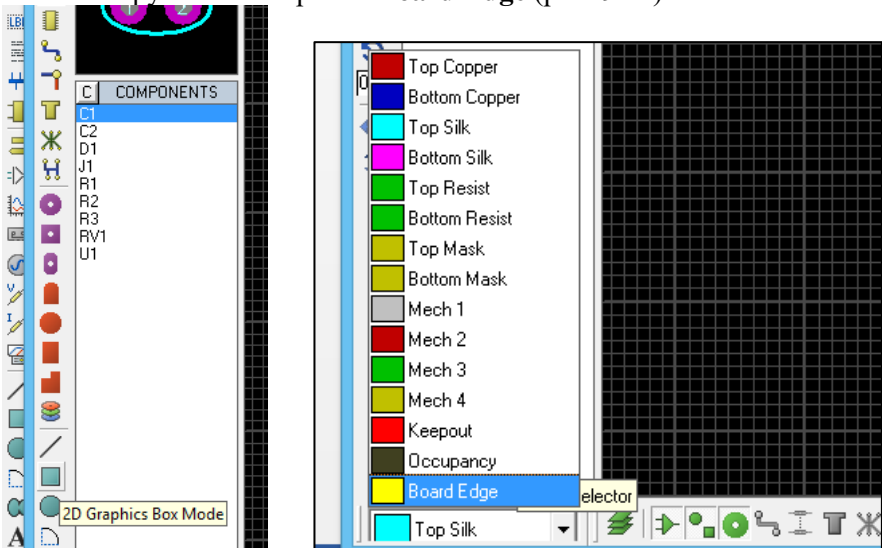


Рисунок 9.10 – Вибір Board Edge

Креслимо прямокутник, невеликий, довільної форми. Це у нас буде сама плата, на яку ми будемо розмішувати посадкові місця елементів і робити розводку (рис. 9.11).

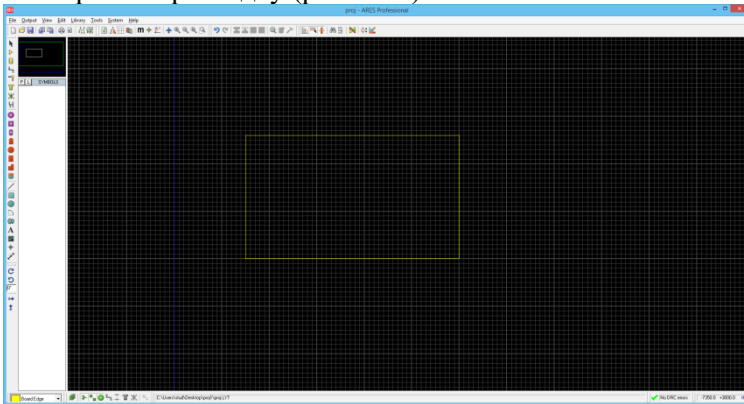


Рисунок 9.11 – Контур плати

Щоб подивитись, як вона буде виглядати в 3D, обираємо **Output -> 3D Visualisation** (рис. 9.12).

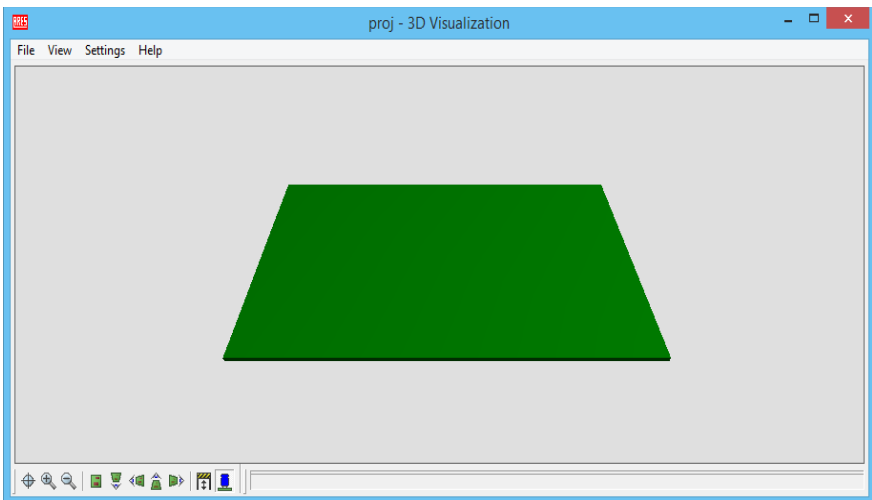


Рисунок 9.12 – Плата в 3D

Далі по черзі розміщуємо елементи, як зображено на рис. 9.13.

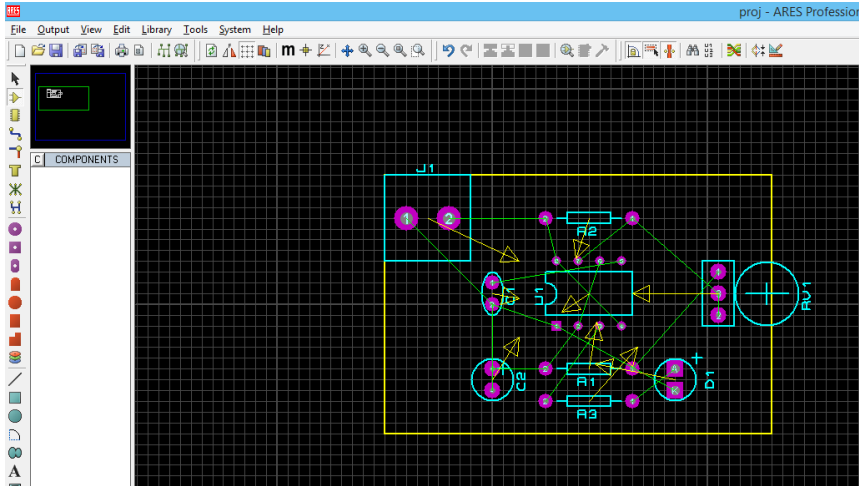


Рисунок 9.13 – Розміщення елементів на платі

У верхній панелі інструментів вибираємо іконку і встановлюємо параметри поля **Pair 1**, як показано на рисунку 9.14, в **Bottom Copper**. Що означає розміщення доріжок на нижній поверхні плати.

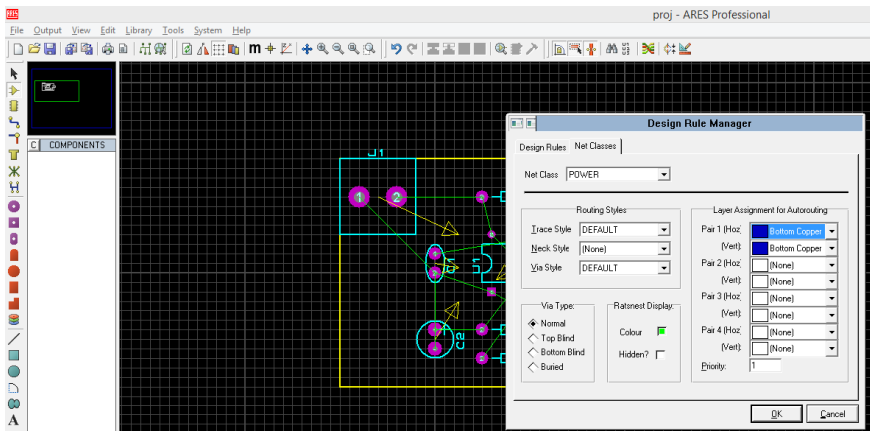


Рисунок 9.14 – Встановлення параметрів поля

Після в цьому ж вікні в полі **Net class** встановлюємо значення **SIGNAL** а також змінюємо значення Pair 1 на Bottom Copper. Кликаємо OK.

У лівій панелі інструментів обираємо Track Mode. Подвійним кліком по DEFAULT. У вікні змінюємо значення Width на 30th (рис. 9.15). Клікаємо OK.

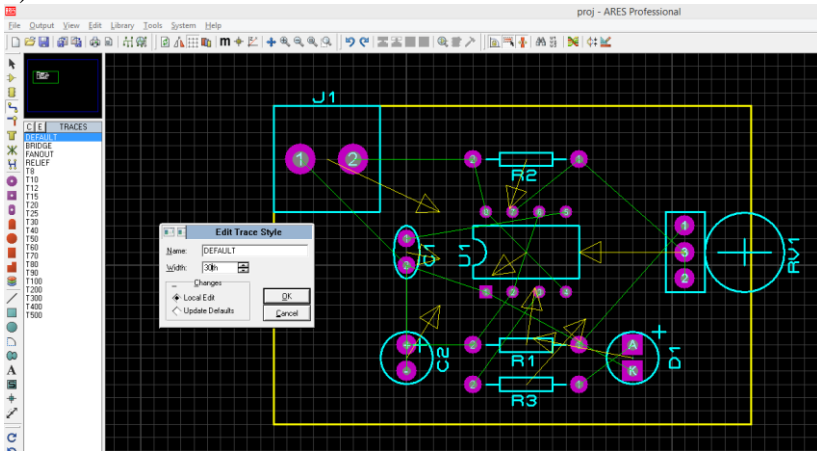


Рисунок 9.15 – Встановлення ширини доріжки

Далі розведемо доріжки на платі засобами ARES. Програма пропонує автоматично обчислити і зробити розведення доріжок (але також можна це робити вручну). У верхній панелі інструментів натискаємо **Auto-router**. У вікні, натискаємо **Begin Routing**. Клікаємо **OK**. І система сама автоматично генерує нам розводку (рис. 9.16).

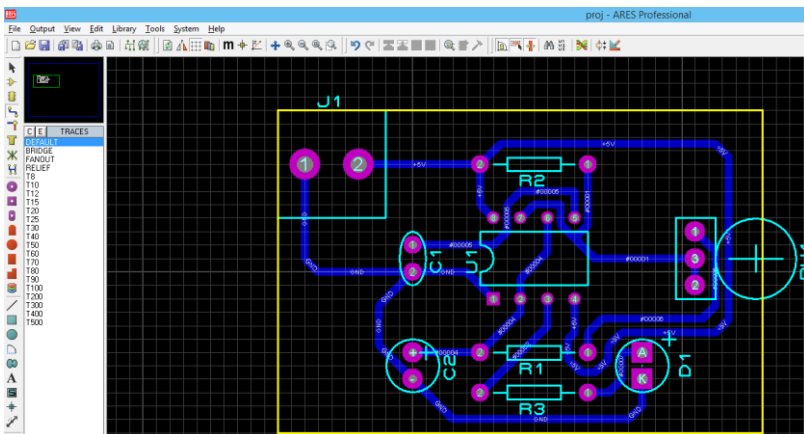


Рисунок 9.16 – Результат трасування 1

Обираємо **Tools -> Power Plane Generator...** Змінюємо значення у вікні як показано на рис. 9.17. Клікаємо **OK**.

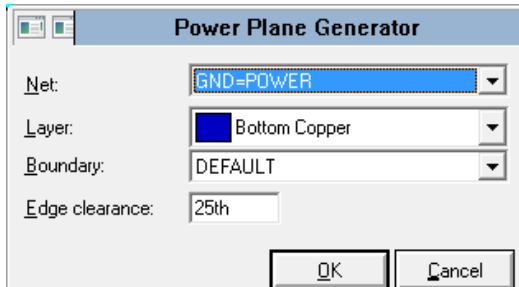


Рисунок 9.17 – Встановлення параметрів

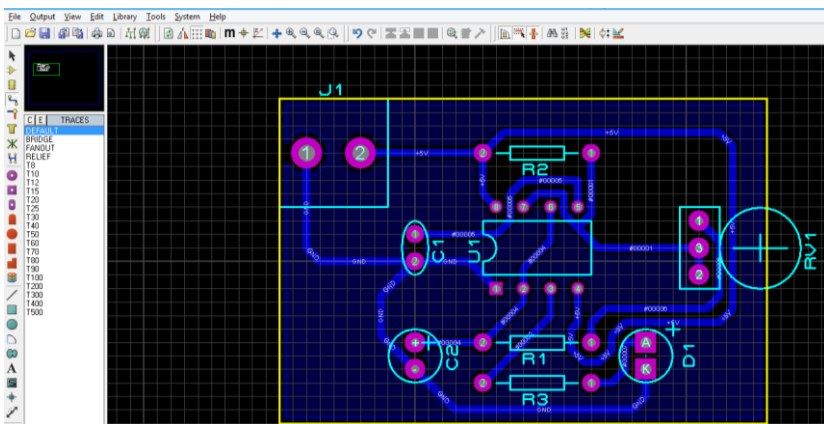


Рисунок 9.18 – Результат трасування 2

Для перегляду візуалізації плати і описаних доріжок вибираємо **Output -> 3D Visualization** (рис.9.19).

Для створення і витравлення плати в реальних умовах для розробника необхідно ч/б зображення розведених доріжок. Proteus дозволяє це зробити: вибираємо **Output -> Print**. Відкривається вікно (рис. 9.20).

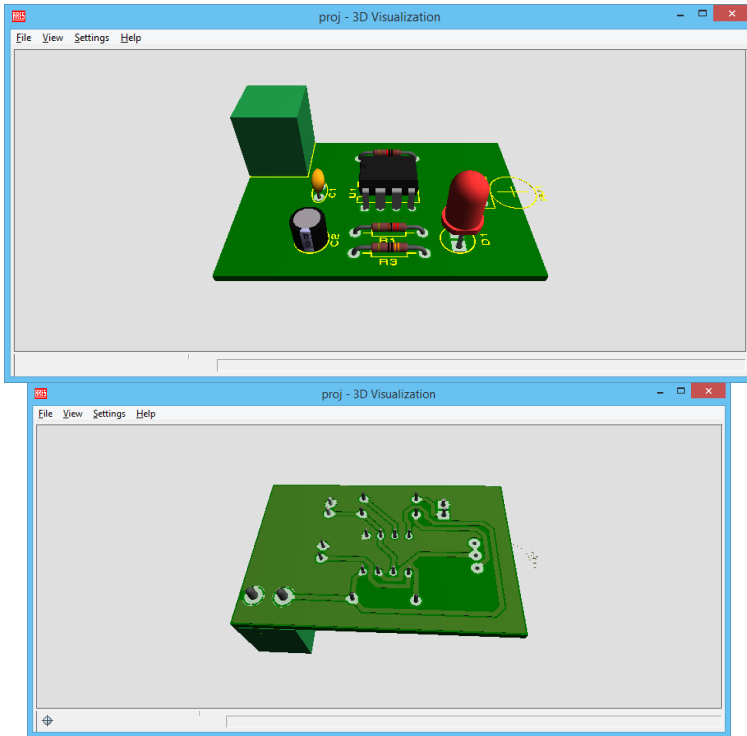


Рисунок 9.19 – Результат візуалізації

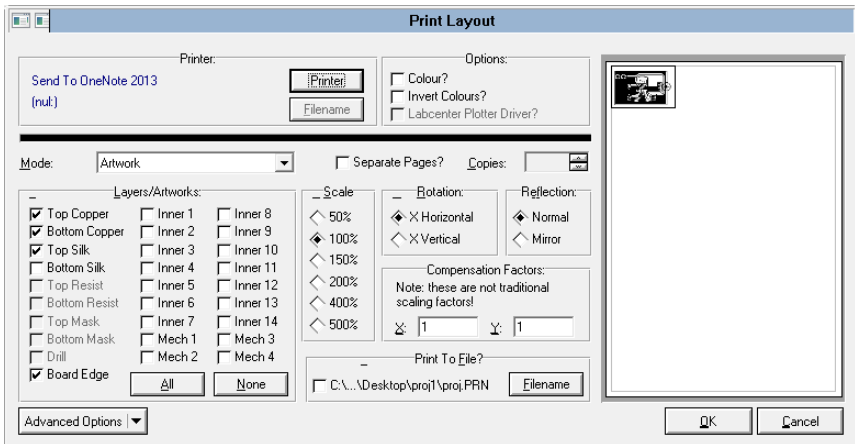


Рисунок 9.20 - Ч/б зображення розведених доріжок

Клікаємо на кнопку **Printer** й обираємо налаштування друку як зображено на рисунку 9.21.

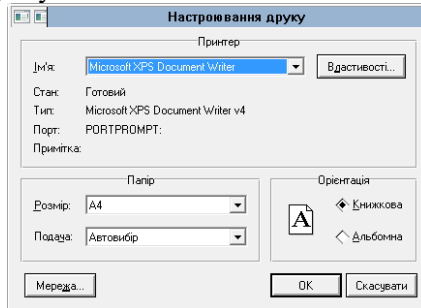


Рисунок 9.21 – Налаштування друку 1

Після цього міняємо деякі налаштування в самому вікні **Print Layout**.

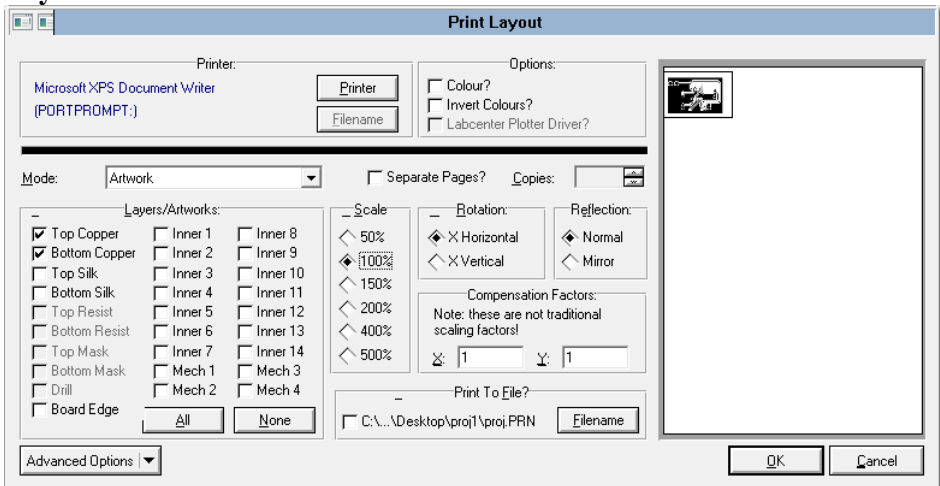


Рисунок 9.21 – Налаштування друку 2

Натискаємо **OK**. Та зберігаємо файл у форматі ***.xps**.

Далі встановимо розміри отриманої плати. Переходимо на лівій панелі інструментів в режим **Dimension Mode** (рис. 9.22) і малюємо лінії, які дорівнюють ширині і довжині плати.

Щоб змінити одиниці вимірювання клікаємо ПКМ на розмірі та з випадючого списку обираємо **Edit Properties...**(рис. 9.23).

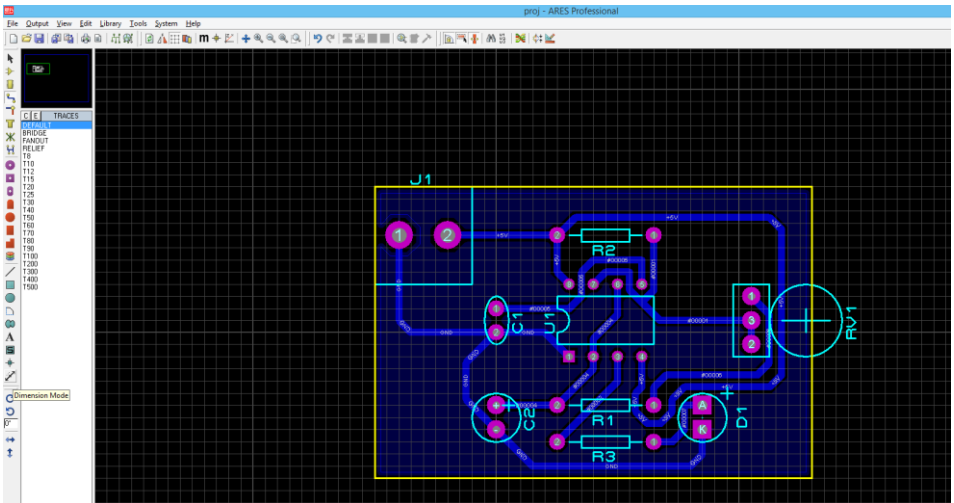


Рисунок 9.22 – Вибір Dimension Mode

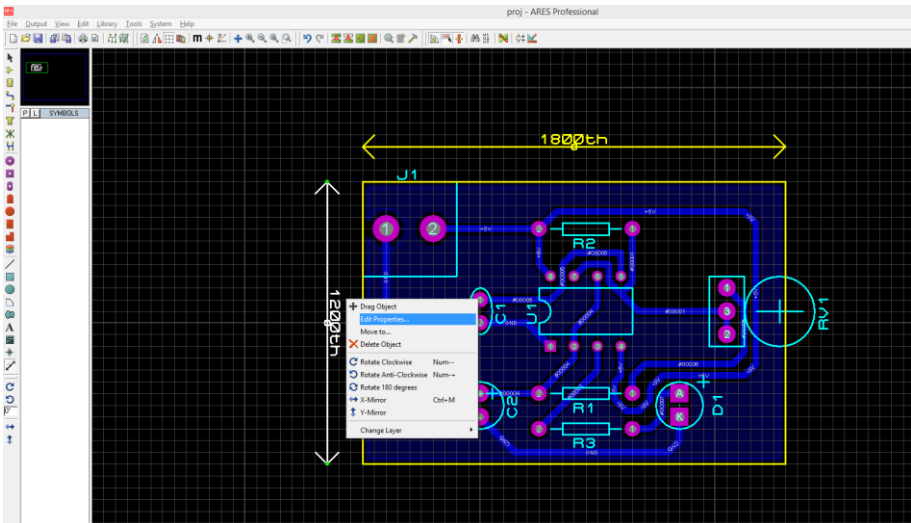


Рисунок 9.22 – Вибір пункту Edit Properties

У вікні, що відкрилося, змінюємо значення (рис. 9.24).

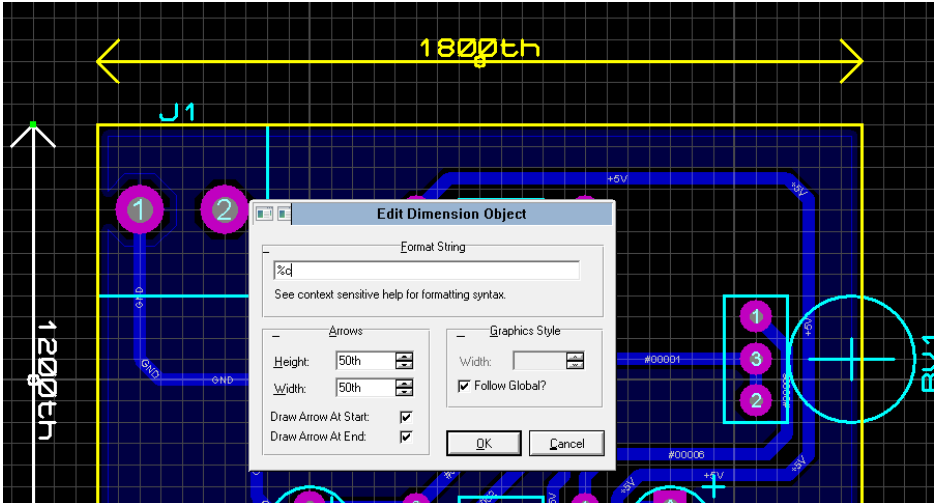


Рисунок 9.24 – Зміна одиниць

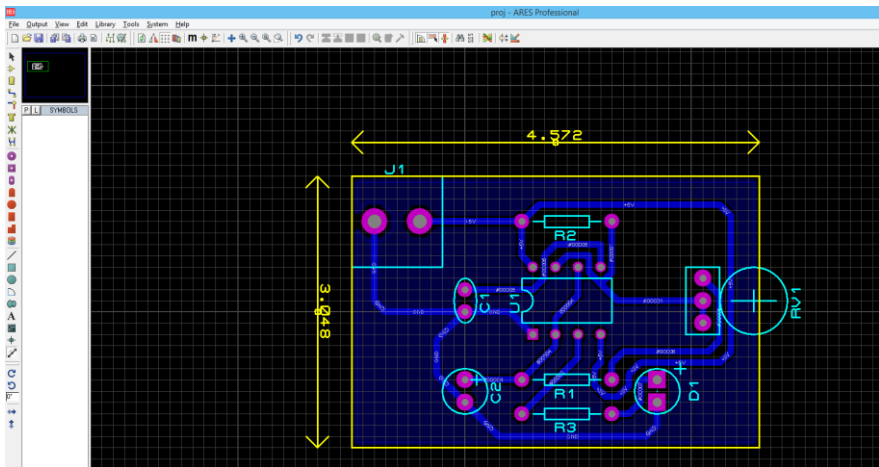


Рисунок 9.25 – Результат роботи

2. Завдання до лабораторної роботи

Спроектуйте друковану плату у середовищі Proteus з лабораторної роботи №4 (мікроконтролер Atmega8, 3 світлодіоди та 4 резистори) та №7 (реле з логічним перемикачем, замість перемикача

встановить мікроконтролер), виконуючи розводку доріжок вручну, задаючи напрям, ширину доріжки, та на якій поверхні вона буде знаходитись.

3. Зміст звіту

- 3.1 Тема та мета роботи
- 3.2 Схема електрична-принципова
- 3.3 Код програми
- 3.4 Висновки з роботи

4. Контрольні запитання

- 4.1 Що таке Footprint?
- 4.2 Які етапи створення друкованої плати?
- 4.3 Як відбувається процес витравлювання плати?

ДЖЕРЕЛА ІНФОРМАЦІЇ

1. TINKERCAD [Electronic source]. – Access mode:
<https://www.tinkercad.com/>
2. Atmel Studio [Electronic source]. – Access mode:
<https://www.microchip.com/en-us/tools-resources/develop/microchip-studio>
3. Arduino IDE [Electronic source]. – Access mode:
<https://www.arduino.cc/en/main/software>
4. Processing [Electronic source]. – Access mode:
<https://processing.org/download/?processing>
5. Proteus [Electronic source]. – Access mode:
<https://www.labcenter.com/education/>.