

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»
КАФЕДРА ПРИЛАДОБУДУВАННЯ

СИМУЛЯЦІЯ ТА АНАЛІЗ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ У
ПРОГРАМНОМУ СЕРЕДОВИЩІ PROTEUS

Методичні вказівки до практичних робіт з дисципліни
«Мікропроцесорні системи»
(магістерський рівень)

Ужгород – 2023

УДК 004.056.55, 003.26

Мешко Р.О., Рябощук М.М. Симуляція та аналіз мікропроцесорних пристроїв у програмному середовищі Proteus. Методичні вказівки до практичних робіт з дисципліни «Мікропроцесорні системи». Ужгород: в-во УжНУ, 2023, 34 с.

Укладачі:

Роман Мешко – старший викладач кафедри приладобудування ДВНЗ «УжНУ»

Михайло РЯБОЩУК – канд. фіз.-мат. наук, доцент кафедри приладобудування ДВНЗ «УжНУ»

Відповідальний за випуск – Михайло РЯБОЩУК., канд. фіз.-мат. наук, доцент кафедри приладобудування ДВНЗ «УжНУ».

Методичні рекомендації розглянуто та схвалено на засіданні кафедри приладобудування, протокол № 9 від 21.06.2023 року.

ЗМІСТ

ВСТУП	4
1 Освоєння принципів роботи з симулятором Proteus	5
2 Аналіз принципової та структурної схеми пристрою автоматизації на основі МК серії AVR	8
3 Побудова прототипа схеми на основі МП платформи Arduino, розробка та відладка алгоритму й програми роботи у середовищі симулятора - Proteus	12
4 Практичне освоєння основних методів дослідження МП систем в режимі симуляції в середовищі Proteus	31
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	34

ВСТУП

Методичні вказівки до виконання практичних робіт із дисципліни «Мікропроцесорні системи» із роз'ясненням, що до виконання індивідуальних завдань.

Відповідно до освітньої програми, практичні роботи із навчальної дисципліни сприяють формуванню у здобувачів вищої освіти таких компетентностей:

- здатність здійснювати автоматизацію складних технічних й технологічних об'єктів і комплексів;
- здатність проектувати та впроваджувати високонадійні технічні системи автоматизації й робототехнічні комплекси та їх прикладне програмне забезпечення.
- здатність застосовувати сучасне спеціалізоване програмне забезпечення для комп'ютерно-інтегрованих технологій та впровадження робототехніки.

1 Освоєння принципів роботи з симулятором Proteus

Мета роботи: Ознайомитися із основними інструментами програмного пакета Proteus.

Початок роботи з Proteus. Proteus - це комерційний пакет програм класу САПР (CAD), який об'єднує у собі дві основні системи:

- ISIS - засіб розробки і налагодження в режимі реального часу електронних схем;

- ARES - засіб розробки друкованих плат. Вбудованим автоматичним трасувальником в ARES, починаючи з версії 7.4, використовується програма ELECTRA

Autorouter. Відмінність від аналогічних за призначенням пакетів програм, наприклад, Electronics Workbench Multisim, MicroCap, Tina і т.п. - із розвинутою системою симуляції цифрових та аналогових схем (інтерактивним налагодженням в режимі реального часу та покроковою симуляцією) для різних сімейств мікроконтролерів: 8051, PIC (Microchip), AVR (Atmel), та інші.

Proteus має великі бібліотеки компонентів, в тому числі і периферійних пристроїв: світлодіодні і РК індикатори, температурні датчики, таймери реального часу, електромеханічні пристрої.

- RTC, інтерактивних елементів введення-виведення: кнопок, перемикачів, віртуальних портів і віртуальних вимірювальних приладів, інтерактивних графіків, які не завжди присутні в інших подібних програмах.

Proteus VSM може використовуватися двома досить відмінними способами — або для Інтерактивного Моделювання або для Моделювання на Основі Діаграм.

Як правило, інтерактивне моделювання використовують, щоб побачити, чи працює розробка взагалі, а графічне - щоб дослідити, чому вона не працює або проводити деталізовані вимірювання.

Налаштування середовища проектування

Нижче наведено основне вікно програми ISIS з поясненнями по призначенню основних елементів інтерфейсу: ліве меню, верхнє меню команд, верхнє основне меню, кнопки симуляції, селектор об'єктів.

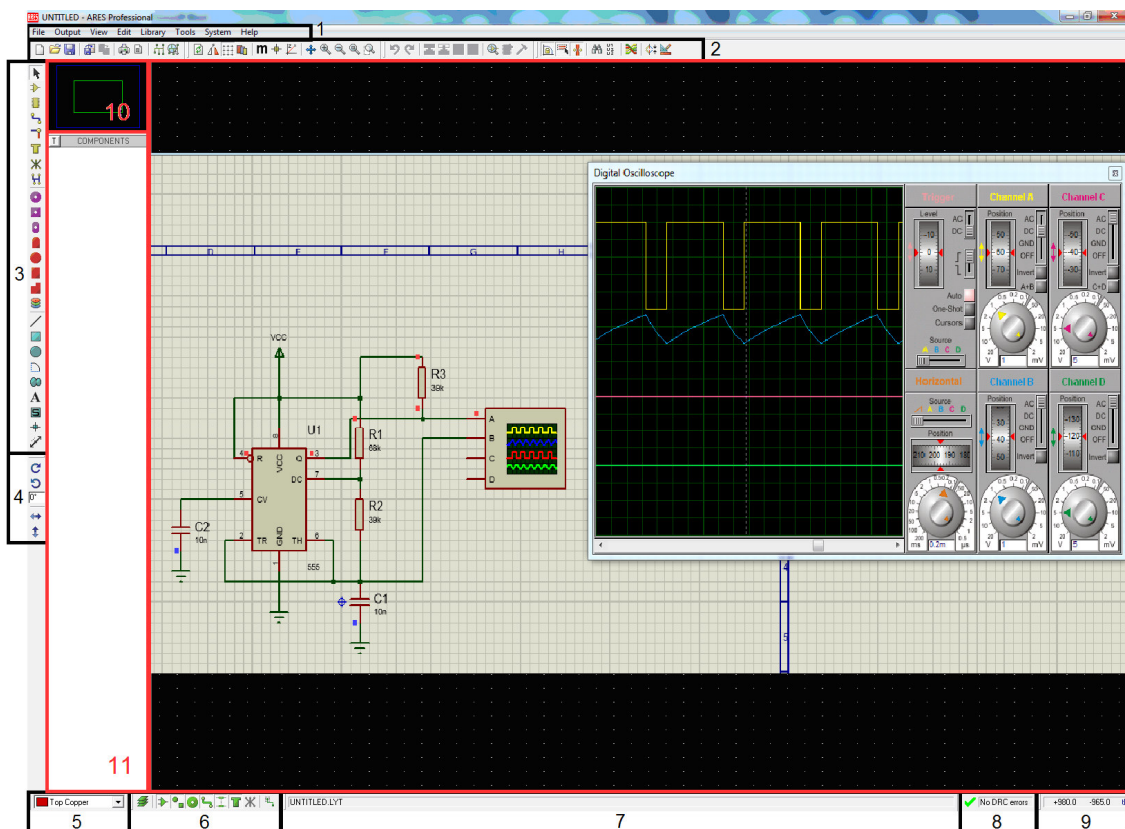


Рисунок 1.1 - Вигляд головного вікна програми ISIS

Елементи головного меню ISIS: 1 – головне меню; 2 – верхня панель інструментів; 3 – ліва панель інструментів; 4 – іконки повороту/відображення об'єкта; 5 – селектор шарів об'єкта; 6 – фільтр для вибору об'єктів; 7 – панель завдань; 8 – вікно стану програми перевірки правил проектування

друкованої плати; 9 – вікно координат курсора миші; 10 – вікно попереднього перегляду; 11 – селектор об'єктів; 12 – основне вікно редагування.

Розглянемо Основні елементи головного робочого вікна:

1. Вікно редагування Editing Window — в ньому відбуваються основні операції створення, редагування і відлагодження моделі.

2. Вікно навігації і попереднього перегляду Overview Window — допомагає побачити в вікні редагування потрібну частину схеми, якщо вона повністю в ньому не поміщається. Достатньо курсором миші вибрати потрібну точку зменшеного зображення схеми у вікні попереднього перегляду.

3. Вікно вибору об'єктів Object Selector — в процесі роботи тут розміщується список компонентів, що використовуються у створюваній моделі, символів та інших об'єктів.

4. Панель інструментів і вибору режимів:

Назва	Група інструментів
Основні режими	
Пристрої	
2D графіка	

Рисунок 1.2 - Панель інструментів і вибору режимів

4.1. Основні режими:

						
безпосереднє редагування	компоненти проекту	з'єднання провідників	маркування провідників	додавання опису	Шини провідників	підсхеми

4.2. Пристрої:

							
Термінали	Виводи пристроїв	Графи	Магніт фон	Генератори	Пробник напруги	Пробник струму	Вимірні пристрої

Назва	Група інструментів
Файл/друк, команди	
Команди відображення	
Команди редагування	
Керування проектом	

Рисунок 1.3 - Панель основних інструментів команд

2 Аналіз принципової та структурної схеми пристрою на основі МК серії AVR

Мета роботи: Ознайомитися с особливостями роботи з пакетом моделювання електронних схем ISIS та змоделювати схему електричну принципову МПСУ згідно схеми електричної функціональної приведеної на рис. 2.1.

Теоретичні відомості

Для того щоб запустити ISIS, натиснути на кнопку ПУСК і вибрати програми, Proteus, а потім ISIS. При запуску програми з'явиться основне вікно. Найбільша частина простору відводиться під вікно редагування EDIT WINDOW.

Саме в ньому відбуваються основні процеси створення, редагування та налагодження схеми пристрою.

Зліва зверху знаходиться маленьке вікно попереднього перегляду OverView Window з його допомогою можна переміщуватись по вікну редагування.

Переміщувати вікно редагування по схемі можна утримуючи натиснутою кнопку SHIFT переміщувати курсор миші, не натискаючи її кнопок, по вікну редагування.

Наближати та віддаляти схему у вікні можна відповідно кнопками F6 і F7, або колесом миші, F5 центрує схему в вікні, а натискання F8 підганяє розмір схеми під вікно редагування.

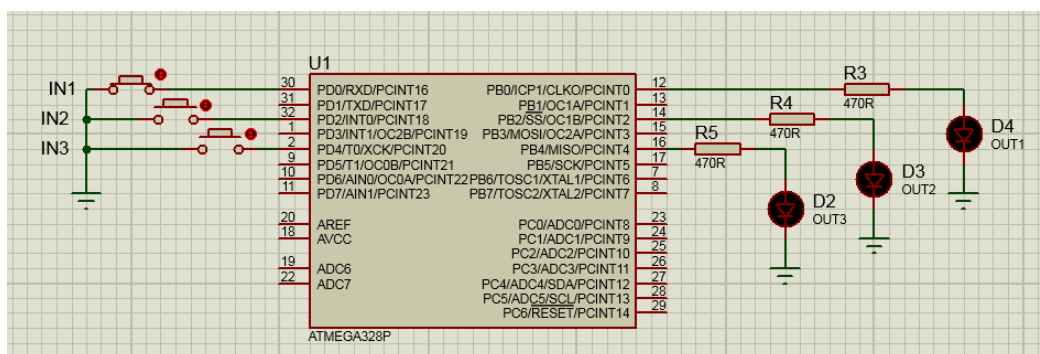


Рисунок 2.1 - Схема електрична функціональна МПС

Під вікном перегляду знаходиться Object Selector список вибраних в даний момент компонентів, символів та інших елементів. Виділений в списку об'єкт відображається в вікні попереднього перегляду.

Всі можливі функції і інструменти Proteus доступні через меню розташоване вгорі основного вікна програми, через іконки, які розташовані під меню і в лівому куті основного вікна, і через гарячі клавіші, які можуть пере призначатися користувачем.

Знизу головного вікна знаходиться: зліва направо кнопки обертання і розвороту об'єкта, панель управління симуляцією, рядок статусу і координати курсору.

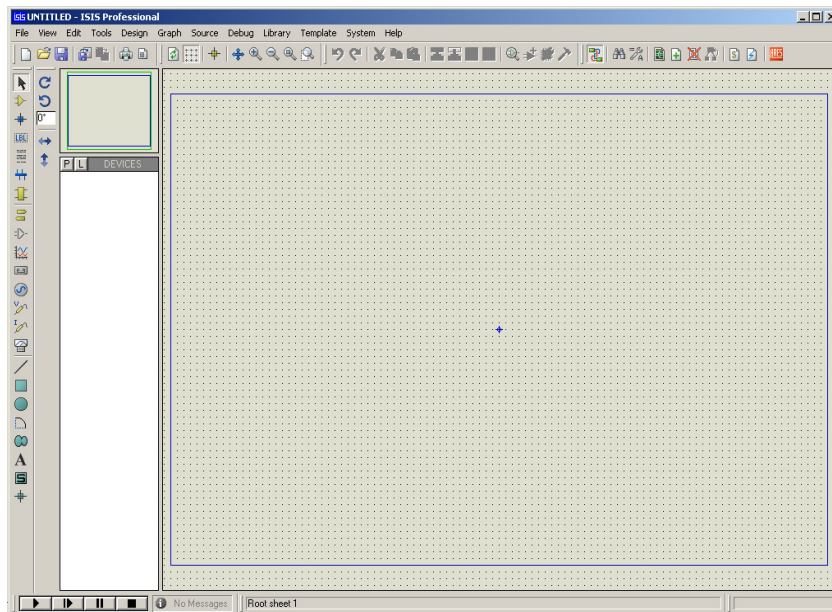


Рисунок 2.2 - Головне вікно програми ISIS

Для додавання компонентів необхідно перейти в режим COMPONENT(компоненти), нажав на відповідну іконку. Далі натиснувши на іконку P (Pick Devices) або двічі натиснути лівою кнопкою в полі вибору компонентів Object Selector, ми потрапляємо в бібліотеку компонентів. Використовуючи пошук знаходимо необхідний компонент.

Для редагування компоненту необхідно натиснути на нього двічі лівою кнопкою миші.

Для того щоб додати компонент “Земля” необхідно вибрати режим Terminals Mode.

Для того щоб завантажити програму в мікроконтролер необхідно:

- Вибрати пункт меню Source -> Add/Remove Source Code Files;

- Вибрати компілятор AVRASM2;

- Додати код програми за допомогою клавіші New;

- Вибрати пункт меню Source -> Build All.

Якщо не було знайдено помилок, то перейти до перевірки схеми.

Для відладки схеми використовується пункт меню Debug.

Завдання для практичної роботи

Використовуючи пакет моделювання електронних схем Proteus створити наступну схему:

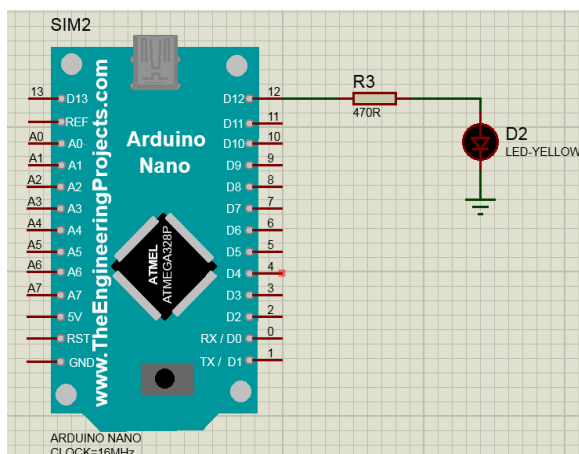


Рисунок 2.3 - Схема електрона

Дана схема вмикає світлодіод.

Завдання для лабораторної роботи:

1. Побудувати схему електричну принципову МПС
2. Записати алгоритм управління з врахуванням схемотехніки
3. Завантажити програму управління у МК
4. Змоделювати функціонування системи на моніторі комп'ютера
5. Відобразити алгоритм функціонування

Порядок виконання роботи:

- Запустити програму ISIS,
- Додати необхідні компоненти,
- Завантажити програму в мікроконтролер,
- Перевірити роботу схеми.

Звіт про лабораторну роботу має містити:

- Титульний аркуш з П.І.Б., номером групи;
- Завдання на лабораторну роботу;
- Лістинг програми, яка реалізує завдання.

Опис об'єкта

Сигнал готовності формують датчики об'єкта: IN1 самостійно запускає систему (датчик являється ведучим), при відсутності сигналу IN1 систему запускають сумісно датчики IN2, IN3.

Час очікування сигналу готовності задається константою таймера T1. По закінченню вказаного часу система вимикається (перехід на кінець процедури).

Управління лінією Control виконується по принципу пріоритету (IN1-IN2-IN3) інформаційними сигналами OUT1, OUT2, UUT3.

Час очікування OUT1, OUT2, UUT3. задається (константа T2). По закінченню вказаного часу система вимикається (перехід на кінець).

Затримка на спрацювання (захист від дребезгу контактів кнопок) влк./вимкн. стану лінії Control (константа T3)

3 Побудова прототипа схеми на основі МП платформи – Arduino, розробка та відладка алгоритму роботи у середовищі симулятора - Proteus 8.6.

Створимо модель пристрою, схема якого показана на рис. 3.1. Це своєрідний мікропроцесорний генератор з функцією зміни частоту спалахів світлодіода D1 за допомогою потенціометра R2.

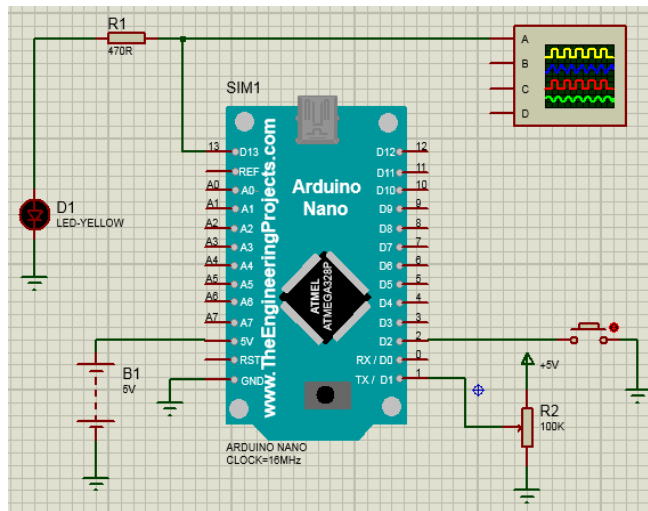


Рисунок 3.1 - Схема пристрою для моделювання

Запуск програми ISIS створює новий проект автоматично. В іншому випадку, для створення нового проекту необхідно вибрати в меню File пункт New Design, або натиснути відповідну піктограму.

Виберіть на панелі інструментів піктограму Компоненти (Component Mode).

Натисніть на піктограмі з літерою "P", що знаходиться над вікном вибору об'єктів (Object Selector), щоб відкрити вікно Браузера Бібліотек (Library Browser).

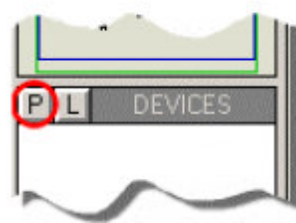


Рисунок 3.2 - Виклик Браузера Бібліотек

У вікні Браузера Бібліотек в полі Keywords введіть назву мікроконтролера (типа платформи - Arduino). У аншому випадку ми досліджуємо - Arduino Nano. Щоб перенести знайдений компонент у вікно вибору об'єктів, натисніть два рази лівою кнопкою миші на рядку з його найменуванням, що з'явиться у вікні Results.

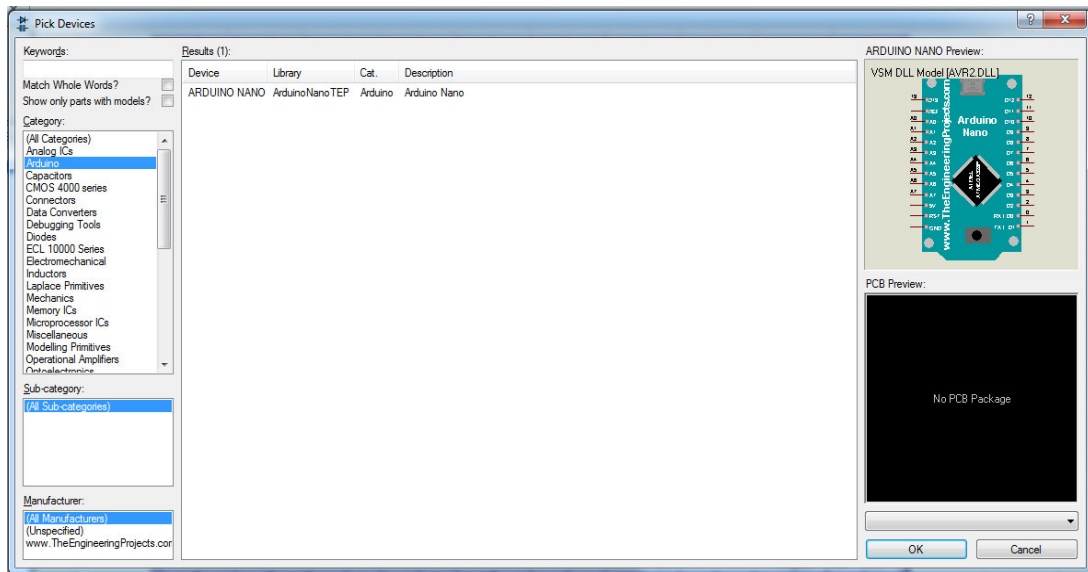


Рисунок 3.3 - Вікно браузера бібліотек

Знайдіть в бібліотеці резистор (RES), змінний резистор (POT-LIN), конденсатор (CAP) і світлодіод червоного свічення (LED-RED).

Компоненти вибираються в одному екземплярі, розмножити їх можна згодом, просто вибираючи їх у вікні вибору об'єктів.

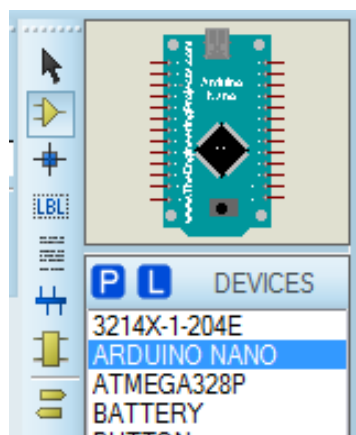


Рисунок 3.4 - Вибір компонентів у вікні вибору об'єктів

Щоб розмістити елемент на схемі, виберіть його у вікні вибору об'єктів (елемент буде показаний в вікні попереднього перегляду в правому верхньому куті екрану),

перемістіть курсор у потрібне місце в вікні редагування та натисніть ліву кнопку миші.

З'явиться контур об'єкту, який буде переміщуватися разом із курсором. При натисненні лівої кнопки ще раз, компонент буде встановлено.

Перед тим, як помістити елемент на схему, його можна попередньо повернути на певний кут чи відобразити дзеркально, використовуючи панель повороту і дзеркального відображення.

Встановіть на схему мікроконтролер. Тепер виберіть компонент RES, натисніть на піктограмі повороту проти годинникової стрілки. У вікні попереднього перегляду побачите резистор, розвернутий на 90°. Встановіть його відповідно до схеми. Тепер, аналогічно, встановіть на схему конденсатор, змінний резистор та світлодіод.

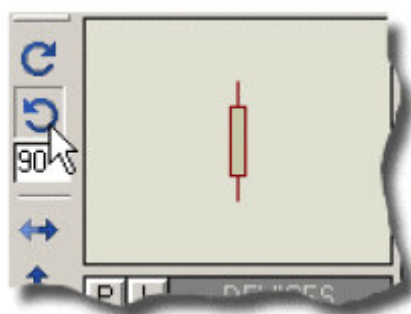


Рисунок 3.5 - Поворот компонента

Щоб змінити розміщення елемента, виберіть його (права кнопка миші), натисніть ліву кнопку миші і перетягніть елемент на необхідну позицію. Натиснення правої кнопки миші над попередньо вибраним об'єктом видалить його. Повернути ж компонент після його розташування на схему можна, виділивши його правою кнопкою миші і вибравши відповідний пункт в контекстному меню.

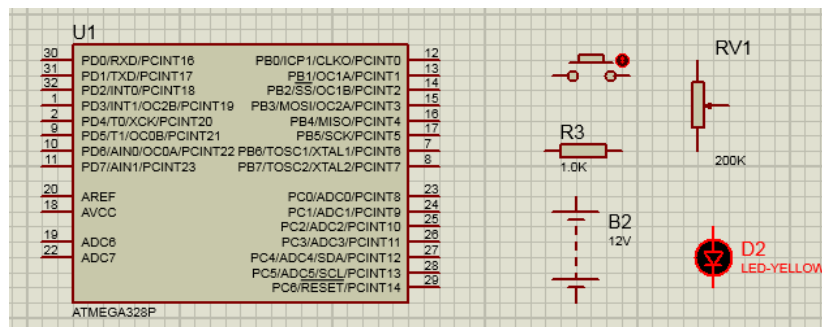


Рисунок 3.6 - Попереднє розташування компонентів на схемі

Маркування і зміна властивостей моделей

При додаванні компонентів на схему, одразу ж відбувається автоматична анотація цих компонентів. Якщо уважно роздивитися резистор, то можна помітити, що ISIS маркує його величиною по замовчуванню (R) і певним унікальним значенням (порядковим номером). Дану властивість можна знайти в меню Tools, пункт Real Time Annotation, по замовчуванню вона ввімкнена.

Щоб змінити маркування і інші властивості, відкриємо вікно редагування компоненту.

Виберіть режим безпосереднього редагування () на панелі інструментів, виділіть резистор R1 правою кнопкою миші та натисніть ліву, або, після натиснення правої кнопки, виберіть пункт Edit Properties (або комбінація клавіш CTRL+“E”) в контекстному меню.

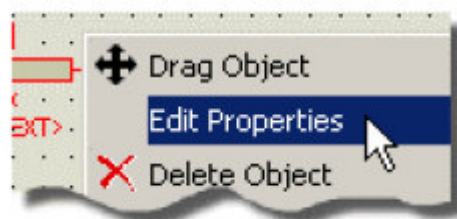


Рисунок 3.7 - Виклик вікна редагування компонента

Змініть його позиційне позначення (Component Reference) з промаркованого R1 на R3 (для більшої відповідності початковій схемі) і опір (Resistance) на 1 кОм (1k).

У випадках, коли важливим є лише логічний сигнал, аналогові елементи намагаються замінити цифровими, що виконують аналогічну функцію. Цим комп'ютер звільняється від рутинного розрахунку перехідних процесів в аналогових елементах.

Моделі багатьох компонентів мають “цифровий” і “аналоговий” варіанти. Змінюйте типи моделей, проте лише там, де це дозволяється схемою.

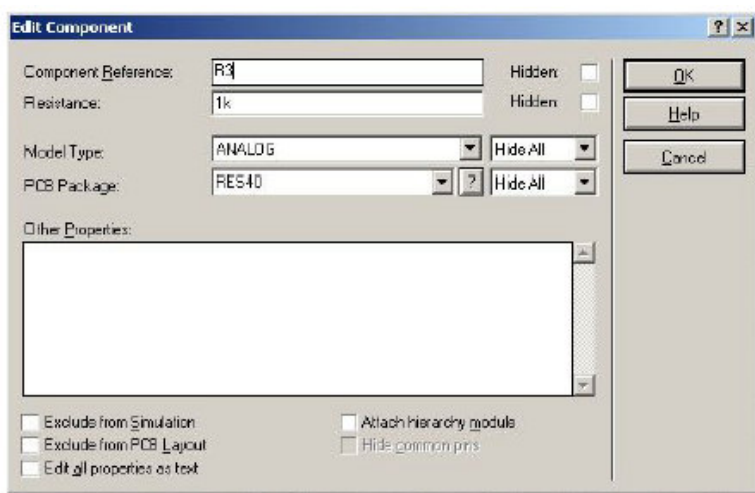


Рисунок 3.8 - Вигляд вікна редагування компонента

Виберіть конденсатор, змініть його позначення на C4, а ємність (Capacitance) – на 100 мкФ (100 uF). Змініть також позначення світлодіоду на HL1 та його номінальний струм (Full drive current) до 2 мА (mA), позначення змінного резистора на R2 і номінал до 100 кОм.

Засоби масштабування і прив'язки. Трасування з'єднань

Як правило, при розводці схеми необхідно змінювати масштаб певної території. Натиснення клавіші <F6> або піктограми Збільшити (Zoom In) збільшить масштаб навколо поточного положення курсору миші, або, в якості альтернативи, затисніть клавішу SHIFT, і, затиснувши ліву кнопку миші, виділіть область, яку необхідно збільшити.

Щоб зменшити масштаб, натисніть клавішу <F7> або піктограму Зменшити (Zoom Out), або, якщо ви хочете побачити робочий аркуш повністю, натисніть клавішу <F8> або використовуйте коліщатко миші, щоб зменшувати або збільшувати необхідну область схеми. Відповідні команди можуть бути також доступні в меню View.

Найпростіший спосіб з'єднати компоненти схеми — використовувати функцію автотрасування (Wire Auto Router), меню Tools. Переконайтесь, що вона ввімкнена, підведіть курсор миші на кінець виводу 6 мікроконтролера (RB0/INT), вигляд курсору зміниться, натисніть ліву кнопку миші, щоб почати з'єднання, і перемістіть курсор до верхнього виводу змінного резистора R2. Коли курсор знову змінить свій вигляд, натисніть ліву кнопку миші, щоб завершити з'єднання.

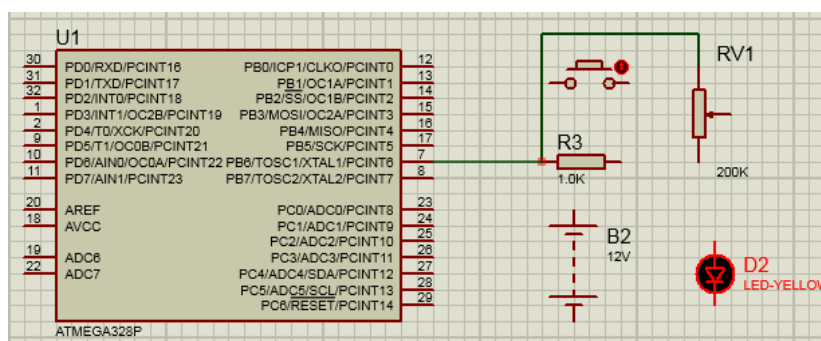


Рисунок 3.9 - З'єднання компонентів

Якщо ж вам не сподобається трасування провідників, зроблене автоматично, то можете відредагувати його вручну. Якщо ж хочете прокласти провідник без автотрасування, то можете просто натиснути лівою кнопкою миші на перший вивід деталі, і далі натискати в тих точках, де ви хочете зробити кут, і, нарешті в кінцевій точці.

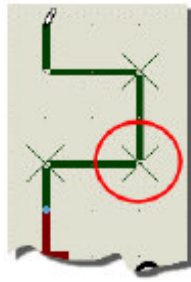


Рисунок 3.10 - Прокладення з'єднань вручну

З'єднайте всі компоненти відповідно до схеми.

Тепер потрібно “заземлити” виводи конденсатора і світлодіоду. Елементи такого типу (термінали) вибираються у відповідному режимі Terminals Mode. Виберіть елемент GROUND (земля) і розмістіть його на схемі під конденсатором і світлодіодом.

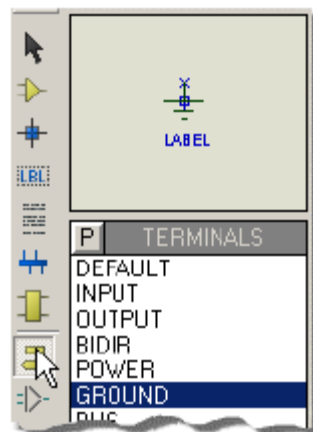


Рисунок 3.11 - Вибір терміналів

Також виберіть режим “Віртуальні вимірювальні пристрої” (Virtual Instruments Mode) та додайте до зібраної моделі вольтметр (DC VOLTMETER). Підключіть вольтметр паралельно конденсатору C4.

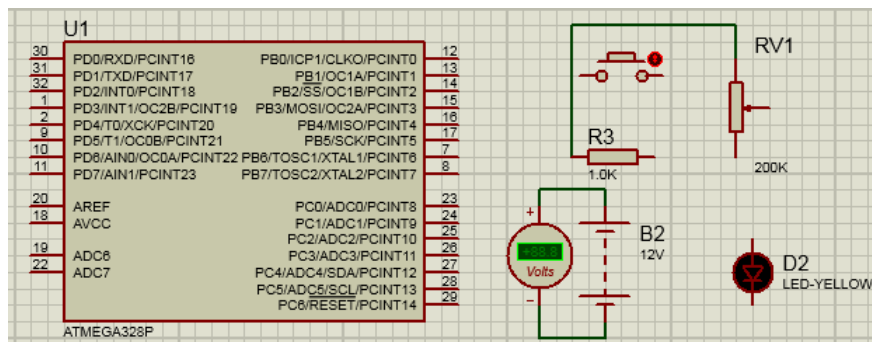


Рисунок 3.12 - Вигляд побудованої схеми

Підключати до моделі мікроконтролера живлення, встановлювати резистор R1, конденсатори C1-C3 і кварцевий резонатор не потрібно. В Proteus ці елементи емулюються програмно і вважаються вже підключеними. Деякі властивості, такі, наприклад, як частота, задаються безпосередньо в параметрах процесора.

Відкрийте вікно редагування мікроконтролера і натисніть на кнопку “Hidden Pins” (приховані виводи) — відкриється додаткове меню, в якому показано, як по замовчуванню називаються вузли (провідники) схеми, до яких підключені живлення мікроконтролера — VDD і його загальний провід — VSS. Зміна цих параметрів може знадобитися при живленні мікроконтролера або інших компонентів різними напругами, або від різних джерел.

Збережіть створений проект. Виберіть пункт меню File > Save Design. В наступному діалоговому вікні вкажіть, де ви хочете зберегти проект та назву проекту.

Назвіть проект Mult. Тип файлу .DSN буде призначений автоматично. Файли з таким розширенням є файлами проектів Proteus VSM.

Написання програми

Proteus можна використовувати як автономний налагоджувач, або як віртуальний схемний емулятор (Virtual

ICE). Ми будемо розглядати Proteus в якості автономного налагоджувача.

Система управління початковим кодом

Система управління початковим кодом виконує дві основні функції:

❑ Реєстрація файлів початкового коду в ISIS, з можливістю їх виклику для редагування без ручного переключення до інших прикладних програм.

❑ Встановлення правил для компіляції початкового коду в об'єктний код.

Створення початкового файлу

Початковий текст програми на мові асемблера MPASM, по якій повинен працювати “генератор”, приведений нижче. Наберіть його в будь-якому текстовому редакторі і збережіть в папці проекту (Mult) під назвою Mult.asm.

Всі мітки повинні починатися з початку рядка, а в останньому рядку повинна бути директива end.

Приєднання початкового файлу до проекту

Щоб додати початковий файл до проекту:

1. Виберіть команду Add/Remove Source Files в меню Source.

2. Натисніть на кнопку New та в наступному діалоговому вікні виберіть або введіть назву файлу початкового коду (якщо ввести назву не існуючого файлу, ISIS запропонує його створити).

3. В полі Code Generation Tool виберіть засіб трансляції програми. В нашому випадку, це асемблер MPASM. Версія цього асемблера, що поставляється з Proteus, не є найновішою. В разі необхідності, її можна замінити більш свіжою.

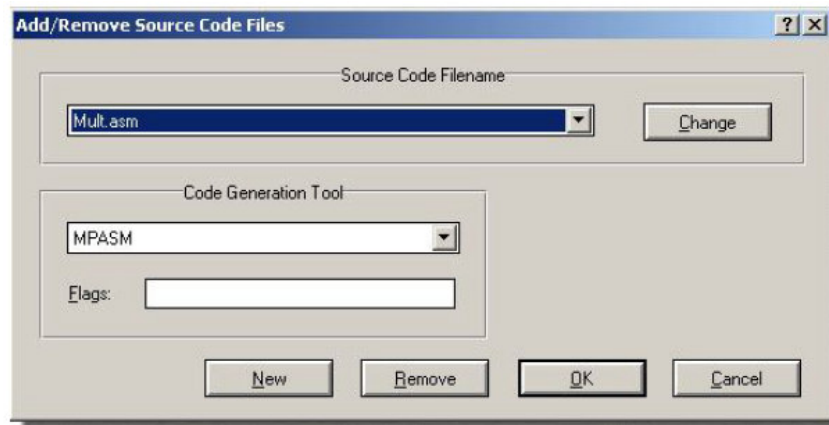


Рисунок 3.13 - Приєднання початкового файлу до проекту

4. Натисніть ОК, щоб додати початковий файл до проекту. Відредагувати вже додані файли можна в меню Source, вибравши потрібний файл.

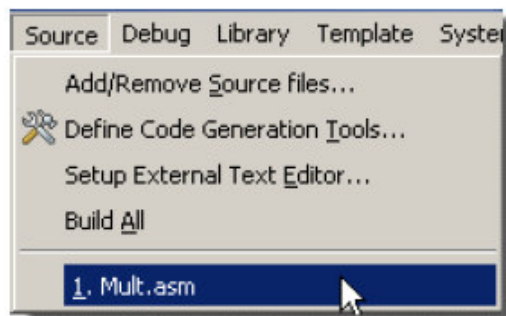


Рисунок 3.14 - Виклик початкового файлу для редагування

Proteus VSM поставляється з простим текстовим редактором — SRCEDIT, який використовується для редагування початкових файлів. SRCEDIT — це розширена версія редактора NOTEPAD, з можливостями відкриття складних початкових файлів, і відповіді на DDE запити. Змінити редактор на будь-який інший можна в меню Source > Setup External Text Editor.

Компіляція початкового файлу

Для того, щоб скомпілювати початковий файл, виберіть пункт меню Source > Build All. Після вибору вказаного пункту, меню початковий текст програми зберігається і запускається

програма MPASM. Після завершення компіляції, на екрані з'явиться вікно з повідомленнями компілятора про знайдені помилки або їх відсутність.

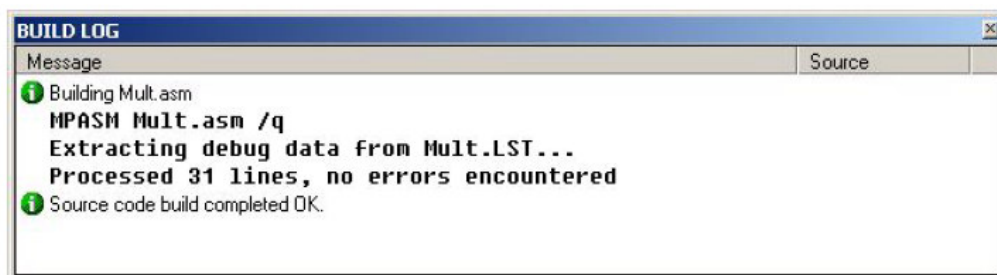


Рисунок 3.15 - Вигляд вікна звіту компіляції

Якщо в тексті були допущені помилки, можете виправити їх, відредагувавши початковий файл. Після редагування, збережіть зміни і повторіть компіляцію (Build All).

Встановлення додаткових засобів трансляції

В поставку Proteus вже входять деякі компілятори (MPASM, MPASMWIN, AVRASM та ін.). Проте, існує можливість додавати нові:

❑ Виберіть пункт Define Code Generation Tools в меню Source.

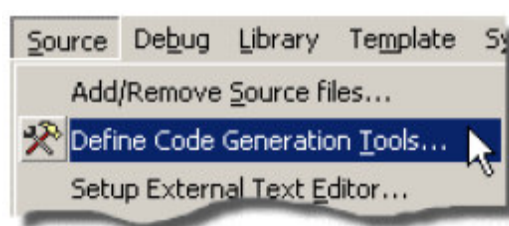


Рисунок 3.16 - Виклик вікна встановлення додаткових засобів трансляції

❑ Натисніть New і вкажіть розміщення виконавчого файлу транслятора. Ви також можете реєструвати командні файли як засоби трансляції.

❑ Введіть розширення для файлів початкового і об'єктного кодів. Вони визначають типи файлів, які ISIS буде шукати,

вирішуючи, використати засіб для специфічного початкового файлу, чи ні. Якщо ввімкнете опцію Always Build, то цей засіб буде завжди викликатися і тоді розширення об'єктного коду не потрібне.

❑ Визначіть орієнтаційний командний рядок (Command Line) для транслятора.

Використовуйте %1, щоб представити файл початкового коду і %2 для файлу об'єктного коду; %\$, щоб вказати шлях до директорії Proteus і %~ для директорії розміщення DSN файлу.

Останній пункт використовується для встановлення параметрів командного рядка, які необхідні для більш злагодженого (тобто без пауз на введення додаткових параметрів користувачем) запуску засобів трансляції, і для визначення шляхів до директорій спеціальних заголовочних файлів і т. і.

В останньому блоці налаштувань (Debug Data Extraction) вибирається спеціальна програма — Debug Data Extractor (DDX), яка зчитує інформацію з файлу лістинга (*.LST), що створюється асемблером або компілятором і генерує файл, який використовується згодом для покрокового налагодження (*.SDI, якщо засобом трансляції є асемблер).

Якщо компілятор, який ви використовуєте, генерує COF, ELF/DWARF та ін. файли, тоді використовувати DDX немає необхідності.

Налагодження програми

Використання Proteus як автономного налагоджувача вимагає, щоб символічний формат налагодження, вироблений вашим компілятором, підтримувався одним із спеціальних завантажувачів, доступних в Proteus. Завантажувач відслідковує адреси кожного рядка початкового тексту в програмі мови високого рівня, і, де це можливо, розміщення програмних змінних.

Формати налагодження, що підтримуються Proteus — COD (підтримується багатьма компіляторами, використовується при налагодженні PIC), UBROF — використовується всіма IAR - компіляторами, ELF/DWARF (підтримується багатьма компіляторами), COFF (підтримується багатьма компіляторами) та OMF (генерується компіляторами Keil). SDI формат застосовується лише для програм, написаних на асемблері та проасембльованих безпосередньо в ISIS.

Якщо використовувався асемблер або компілятор, який підтримує Proteus, то він створить вікно початкового коду для кожного початкового файлу, задіяного в проекті.

Ці вікна з'являться в меню Debug, коли симуляція буде призупинена (поставлена на паузу).

Симуляція схеми

Після успішної трансляції в папці Mult (папка проекту) з'явиться файл Mult.HEX, який необхідно “завантажити” в модель мікроконтролера. Для цього відкриємо вікно його властивостей (виділимо правою кнопкою миші та виберемо Edit Properties в контекстному меню). В полі Program File за допомогою піктограми справа від поля вкажіть шлях до файлу Mult.HEX. Встановіть тактову частоту процесора в полі

Processor Clock Frequency — 4 МГц (4MHz). Слово конфігурації (Program configuration word), якщо воно, як в нашому випадку, не було задане в початковому тексті програми, можна ввести в шістнадцятковій формі або прийняти встановлене по замовчуванню. За допомогою меню Advanced Properties встановлюються додаткові параметри мікроконтролера.

Для початку симуляції натисніть на кнопку “Play” на анімаційній панелі в правому нижньому куті екрану. Програма ISIS почне симуляцію спроектованого пристрою.

Біля виводів компонентів з'являться помітки у вигляді квадратів, кольори яких відповідають поточним логічним рівням: червоний — високий, синій — низький, сірий — невизначений (проміжний). Покази вольтметра зростають — йде зарядка конденсатора C4. Світлодіод HL1 ввімкнений.

Як тільки напруга на конденсаторі досягне значення, що сприймається мікроконтролером як логічна "1", програма змінить високий рівень на виводі 6 мікроконтролера на низький, світлодіод погасне і почнеться розрядка конденсатора C4.

Як тільки напруга на ньому знизиться до рівня, який сприймається як низький (вхід мікроконтролера має характеристику тригера Шмітта), на виводі 6 знову буде встановлений високий рівень, що приведе до ввімкнення світлодіоду і зарядки конденсатора.

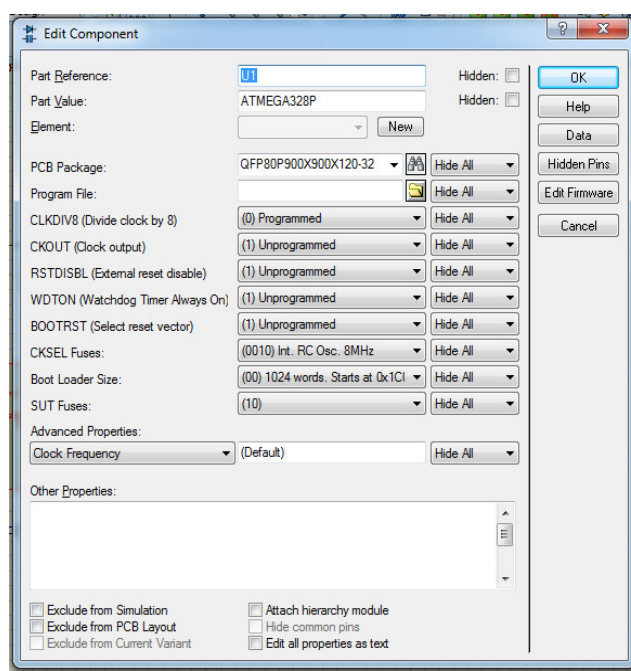


Рисунок 3.17 - Вікно властивостей компонента (МК)

Далі описаний процес повторюється періодично.

Переконайтеся, що частоту спалахів світлодіоду можна регулювати змінним резистором R2. Положення движка

цього резистора змінюють, натискаючи на кнопки зі стрілками біля його умовного позначення.

Вікно початкового коду

Вікно початкового коду є первинним інструментом для налагодження проектів в Proteus. У вікно налагодження початкового коду програма завантажується через налагоджувальний файл, згенерований компілятором, або через SDI-файл, згенерований системою управління початковим кодом.

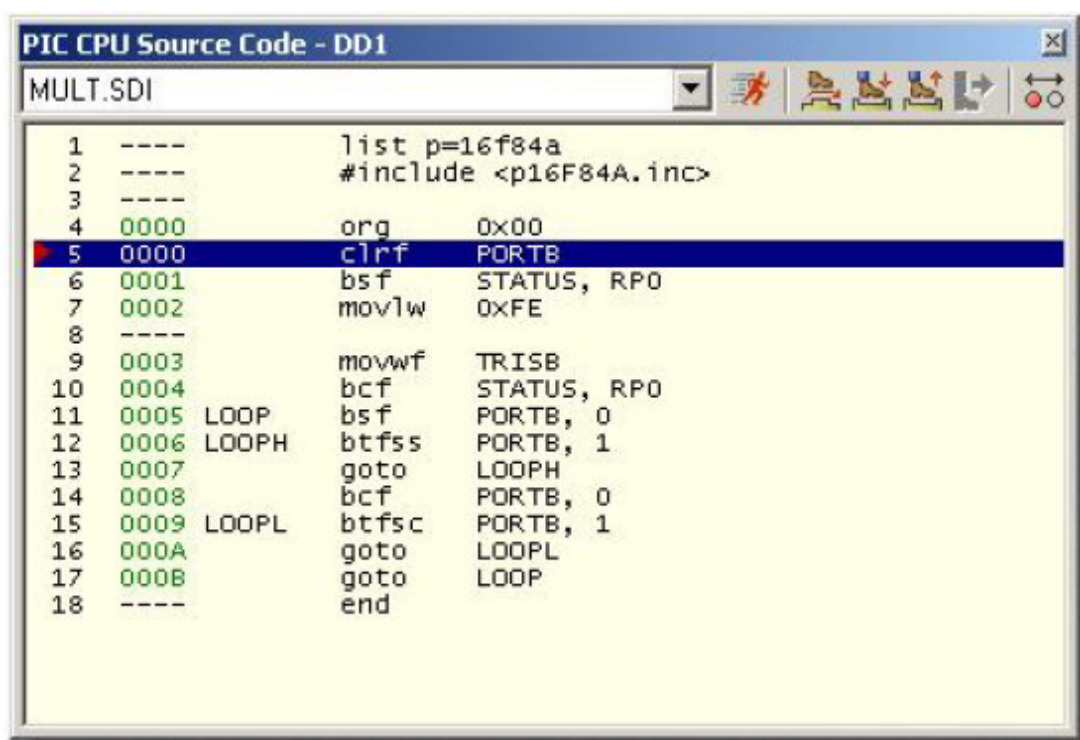


Рисунок 3.18 - Вікно налагодження

Зупиніть симуляцію, натиснувши на кнопку "Stop", а потім знову запустіть, на цей раз кнопкою "Step". В результаті, модель буде ініціалізована, але зупинена в початковому стані. На екрані з'явиться вікно початкового коду (вікно налагодження) з виведенням в ньому текстом початкової програми. Якщо таке вікно не з'явилося або ви його випадково закрили, воно може бути активоване в меню

Debug (пункт CPU Source Code) разом з іншими інформаційними вікнами.

Рядок програми, який відповідає поточному значенню програмного лічильника (в даному випадку — нульовому), відмічений червоним трикутником.

Випадаючий список у верхній частині вікна дозволяє вибрати необхідний файл налагодження (для програм, проасембльованих в ISIS) або необхідний початковий файл, якщо їх є декілька.

Основні команди налагодження (доступні також в меню Debug):

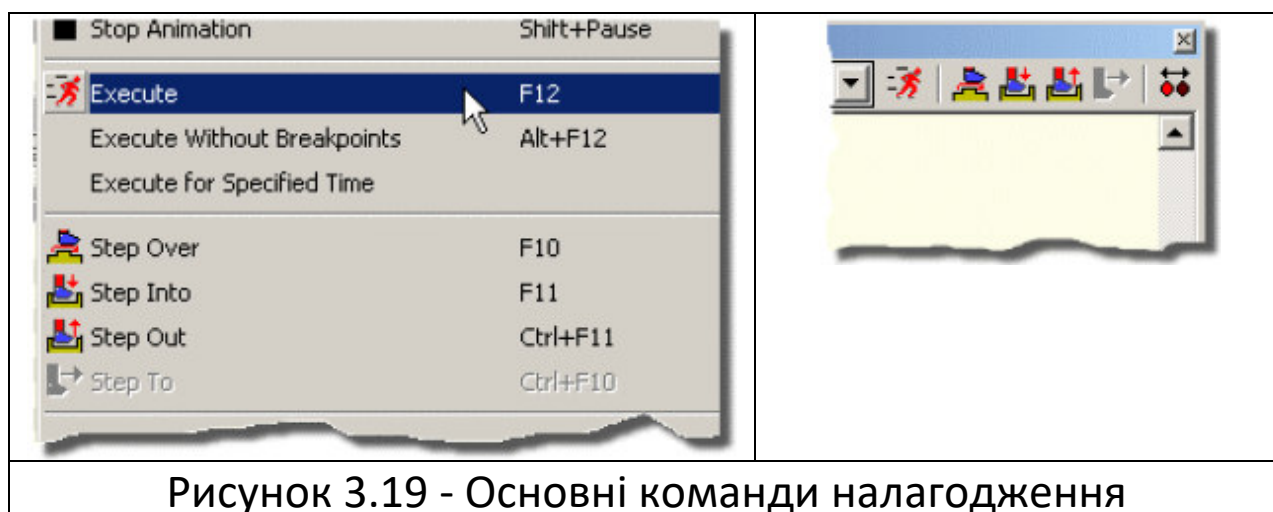
❑ Execute — запуск або продовження виконання програми (еквівалент кнопки “Start”).

❑ Step Over — крок без входу в підпрограму.

❑ Step Into — крок з входом в підпрограму.

❑ Step Out — виконувати симуляцію до виходу з підпрограми (зупинка відбувається на наступній команді після команди повернення).

❑ Step To — виконувати програму до виділеного рядка.



Натиснувши праву кнопку миші в полі вікна налагодження, можна налаштувати деякі його параметри (ввімкнути або вимкнути відображення номерів рядків, адрес

команд, опкодів; вибрати шрифт відображення інформації; колір відображення елементів), знайти потрібний рядок програми по заданій адресі чи порядковому номеру, управляти контрольними точками.

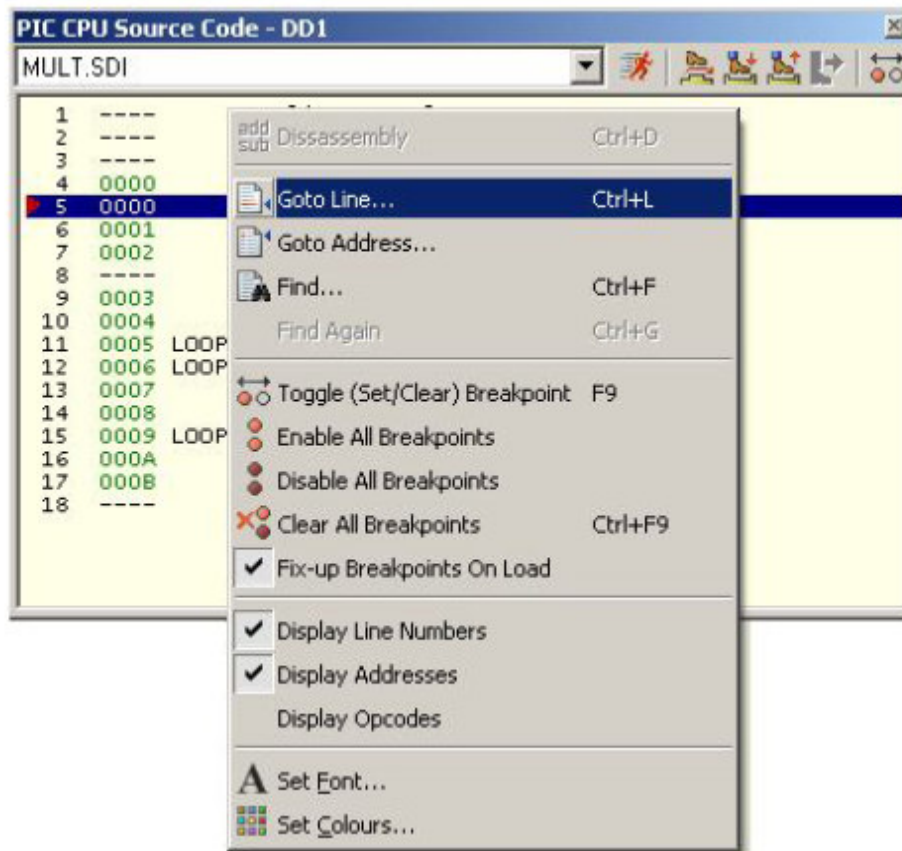


Рисунок 3.20 - Контекстне меню вікна налагодження

Початок симуляції

Почніть симуляцію заново (натисніть "Stop", а потім "Start"). Вікно спостереження з'явиться з початком симуляції. Значення регістра I/O status (TRISB) одразу ж зміниться (вивід RB0 встановиться для роботи на вихід, всі інші — на вхід).

Далі значення цього регістра змінюватись не буде.

Значення двох молодших розрядів порту (PORTB) будуть відображати зміну логічних рівнів на відповідних виводах мікроконтролера.

Watch Window				
Name	Address	Value	Previous Value	Watch Expression
PORTB	0x0006	0b00000001	0b00000010	
I/O status	0x0086	0b11111110	0b11111111	

Рисунок 3. 21 - Вікно спостереження.

Зміна значення регістрів в процесі симуляції

Вікно спостереження дозволяє зупиняти симуляцію, коли значення певного його елементу задовольнятиме визначену заздалегідь умову. Найпростіший приклад — зміна значення елементу. Проте, в разі необхідності, можна встановлювати значно складніші умови зупинки.



Рисунок 3.22 - Встановлення зупинки за умовою

Визначення умов зупинки:

1. Натисніть CTRL+F12 (команда “Step” анімаційної панелі), або призупиніть симуляцію, якщо вона вже виконується.
 2. Виберіть команду Watchpoint Condition з контекстного меню вікна спостереження.
 3. Встановіть глобальну умову зупинки (Global Break Condition).
- ☐ Turn off (disable) watch points — вимкнути зупинку за умовою.

❑ Suspend the simulation if ANY expression is true — зупинити симуляцію, якщо виконується будь-яка умова.

❑ Stop the simulation only when ALL expressions are true — зупинити симуляцію, тільки коли виконуються всі умови.

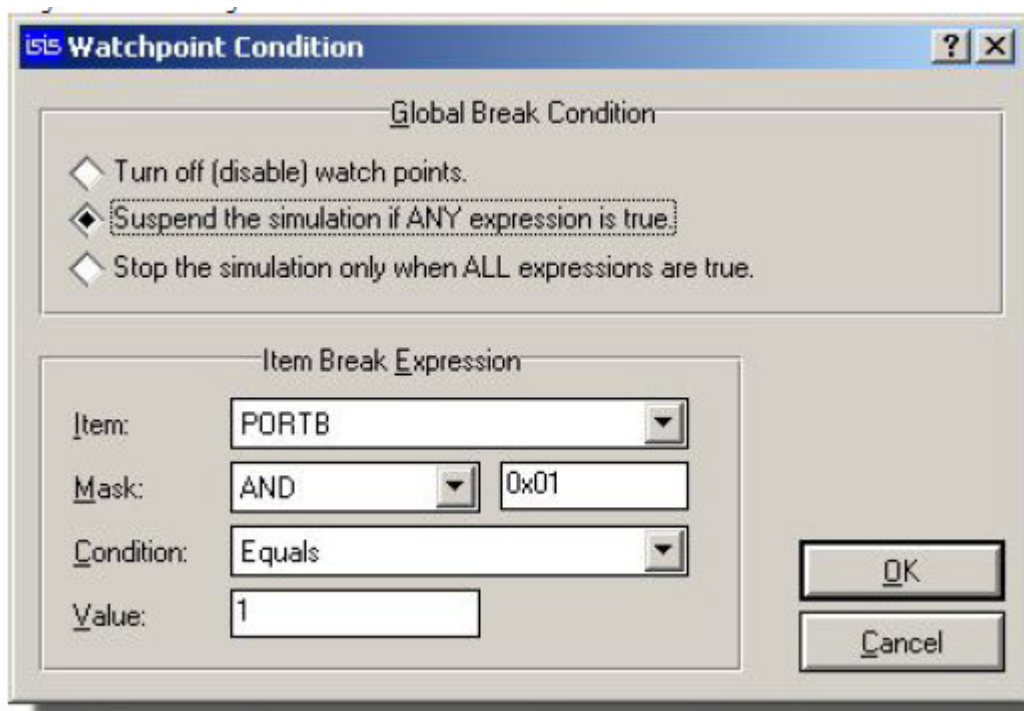


Рисунок 3.23 - Вибір умов зупинки

Освоєння модульного принципу побудови мікропроцесорних автоматизованих систем на основі платформи Arduino, та основні методи їх дослідження в режимі симуляції в середовищі - Proteus 8.6 SP1.

4 Практичне освоєння основних методів дослідження МП систем в режимі симуляції в середовищі Proteus

1. Провести «компіляцію/ перевірку» (зкладка меню - Скетч) створеного скетча у середовищі Arduino IDE.

2. У закладці (Скетч), провести «Експорт бінарного файлу». Результатом є створення папки, яка містить файл із

розширенням (*.hex), вказаний тип файлу використовується для завантаження у МК.

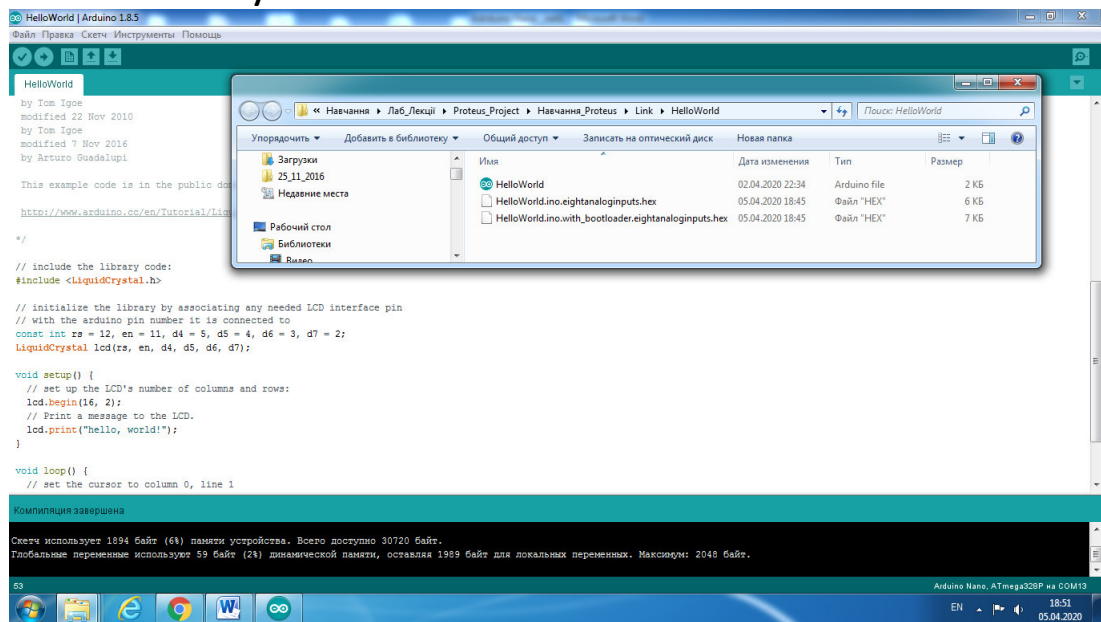


Рисунок 4.1 - вигляд інтерфейсу середовища Arduino IDE.
Генерація Нех файла.

3. Створити використовуючи елементну базу програмного середовища Proteus 8.6. принципову електричну схему об'єкту згідно завдання, наведеного нижче.

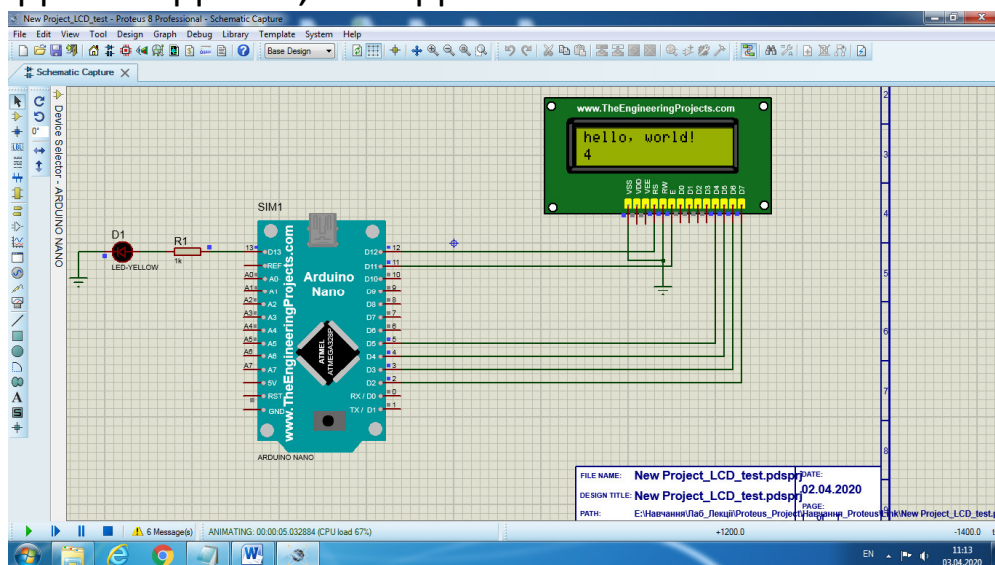


Рисунок 4.2 - Схема підключення LCD та Arduino Nano

Для реалізації повноцінної емуляції роботи пристрою є необхідність підключення бібліотеки модуля – Arduino Nano, та знако- символного дисплея (1602).

Бібліотеки можна завантажити на відкритих ресурсах за посиланнями:

<https://www.theengineeringprojects.com/2015/12/arduino-nano-library-proteus.html> - бібліотека Arduino Nano для Proteus

<https://www.theengineeringprojects.com/Downloads/downloads.php> - бібліотека символьного дисплея (16x2), паралельний тип інтерфейсу (4, 8 біт інтерфейс).

Інсталяція вищезгаданих бібліотек здійснюється шляхом копіювання у папку Proteus 8.6, зі шляхом –

C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\LIBRARY

Файлів двох типів: (*.IDX), та (*.LIB).

Після копіювання, необхідно перезавантажити Proteus 8.6. для прийняття у дію скопійованих бібліотек.

У властивостях елемента Arduino Nano, провести підключення бінарного файлу (*.hex) який було згенеровано попередньо. Шлях до необхідного бінарного файлу знаходиться у вкладці «Program Files».

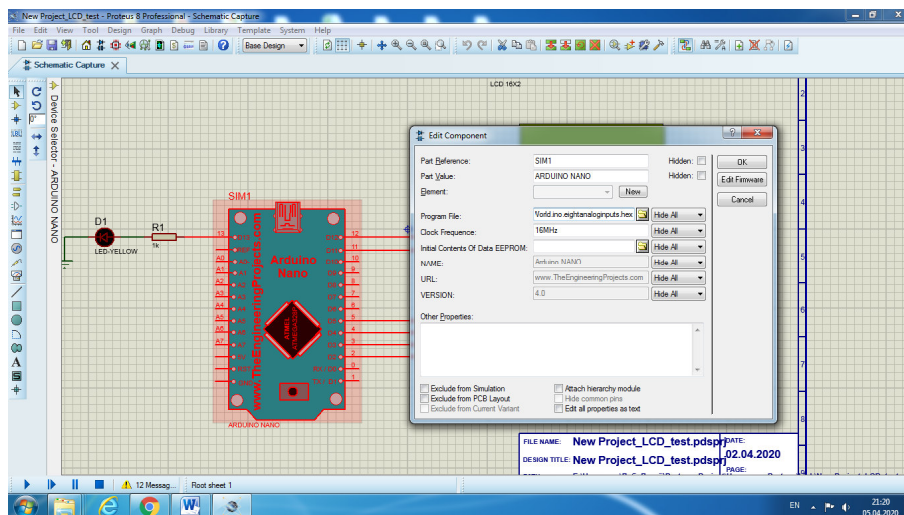


Рисунок 4.3 - Вікно налаштування властивостей модуля Arduino Nano.

Результатом успішної роботи симульованого пристрою є відображення на дисплеї запланованого повідомлення «hello World!»

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Максимов А. Моделирование устройств на микроконтроллерах с помощью программы ISIS из пакета PROTEUS VSM / А. Максимов // Радио. — 2005. — № 4, 5, 6. — С. 30—33, 31—34, 30—32.
2. Цирульник С. М. Застосування програми ISIS пакету Proteus VSM при вивченні курсу «Мікропроцесорна техніка» // С. М. Цирульник, В. К. Задорожний // Матеріали XIII міжнародної конференції з автоматики управління (Автоматика 2006). — Вінниця: Універсум-Вінниця. — 2007. — С. 526—530. — ISBN 978-966-641-210-5.
3. Проектування та аналіз електричних схем в програмному середовищі Proteus VSM. Методичні вказівки до самостійної роботи студентів курсу "Проектування мікропроцесорних систем керування технологічними процесами". Медвідь В.Р., Пісьціо В.П., Тернопіль: ТНТУ, 2018 - 26 с.
4. Навчальний посібник призначено для студентів інженерно-технічного факультету ДВНЗ «УжНУ» спеціальності 151- «Автоматизація та комп'ютерно-інтегровані технології».
5. Проектування вбудованих мікропроцесорних систем: Метод. Вказівки до викон. Лаборатор. Робіт / Уклад.: - К.: НТУУ «КПІ», 2009. - 59 с.

*Формат 60x84/16. Умовн. друк. арк. 2,55. Зам. № 27. Наклад 50 прим.
Видавництво УжНУ «Говерла».
88000, м. Ужгород, вул. Капітульна, 18. E-mail: goverla-print@uzhnu.edu.ua*

Свідоцтво про внесення до державного реєстру
видавців, виготівників і розповсюджувачів видавничої продукції –
Серія 3т № 32 від 31 травня 2006 року