

ЛАБОРАТОРНА РОБОТА №1

Логічні команди. Ознайомлення з середовищем ISIS і CodeVisionAVR

Мета роботи: набути знань і теоретичних навичок при проектуванні програмних та апаратних засобів із використанням мікроконтролера

1.1 Короткі теоретичні відомості

Для розробки пристроїв з використанням мікроконтролерів використовуються програмні компілятори з мов високого рівня в машинні коди, програмні симулятори роботи мікроконтролера для перевірки працездатності логічних структур програми, програмні симулятори схеми мікроконтролера, та внутрісхемні відладчики. Найкраще використовувати внутрісхемні відладчики, оскільки саме вони працюють із реальним пристроєм та реальною схемою, а також дають можливість контролювати хід виконання програми передаючи дані з робочого процесора на відладчик. Проте такі системи вимагають спеціального обладнання і зазвичай дорожчі за програмні.

При розробці програми можна скористатися різними мовами програмування при умові, що для конкретного типу МК є такий компілятор. Для мікроконтролерів типу AVR є компілятори з мови асемблер (практично всі виробники МК мають безплатний компілятор асемблера. Проте ним користуватися важко, адже кожен із процесорів має свої команди. Прогресивнішими є мови Бейсік, Паскаль та Сі. Для контролерів AVR є німецький компілятор E-lab. Компілятор має потужне середовище розробки, відладчик та візард розробки. За своїми можливостями цей продукт практично немає рівних, проте він має багато помилок, які не дають гарантії роботоздатності програми згенерованої цим компілятором.

Компіляторів з мови С для МК з ядром AVR є декілька. Наприклад ImageCraft AVR ANSI C. Цей компілятор “правильний” і вимагає складне узгодження модулів написаних на мові асемблера та основною програмою. Більш правильний компілятор IAR System AVR. Фірми IAR та ImageCraft розробляють компілятори для різноманітних контролерів, в тому числі і для AVR.

У строю таких монстрів румунський компілятор CodeVision AVR наче іграшка. Проте в цієї програми є всі необхідні якості для успішного і швидкого написання програми для AVR. Успіху їй надають три основні моменти. По-перше генерований код правильно працює на реальних контролерах, по-друге є генератор початкового коду та макроси роботи з основними електронними вузлами (SPI, ІС, 1Wire рідкокристалічний індикатор і т.і) і по-третє в оболонку вмонтовано програматор.

Для відладки програми використовується практично єдиний (за невеликим виключенням, наприклад E-Lab та IAR мають власні відладчики) відладчик aStudio. Проте відладити програму із врахуванням схеми обв'язки за допомогою aStudio не вдасться. Для відладки програми в складі обв'язки можна скористатися програмою Proteus. Вона дає можливість відладити програму на мікроконтролері із використанням кнопок, клавіатури, індикаторів, АЦП, терміналів послідовної передачі, тощо.

Для вивчення можливостей МК треба виконати хоч якусь обдуману дію із його використанням. При цьому широко використовуються так звані плати розширення. Для AVR випускаються плати STK200, STK300 та новіші STK500. На них встановлено сам МК та різноманітні системи для простого інтерфейсу із ПЕОМ.

Мікроконтролер ATMeg64 має:

- 64кб пам'яті ;
- 4кб SRAM;
- 2кб on-chip EEPROM;
- два 8-ми бітних таймери лічильники;
- два розширених 16-ти бітних таймери-лічильники із прескалером;
- режими порівняння і захоплення та широтноімпульсну модуляцію;
- вбудований аналоговий компаратор;
- програмований сторожовий таймер із власним генератором;
- два програмовані UART;
- 8-ми каналний 10 бітний АЦП;
- головний і керований режими обміну інтерфейсу SPI.

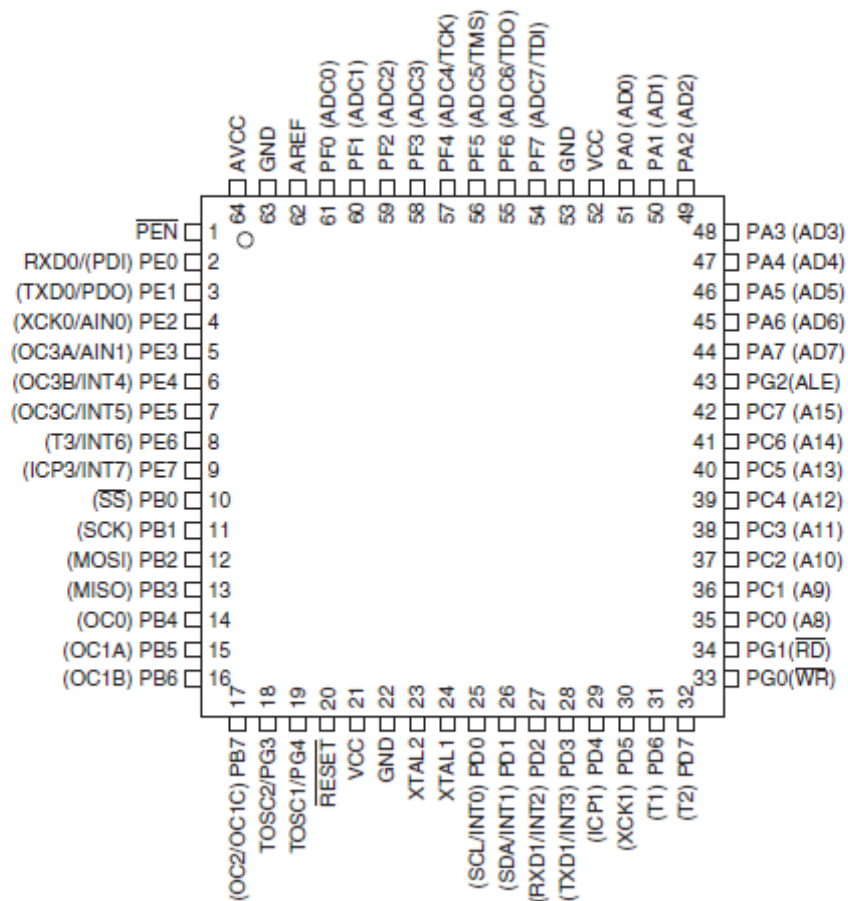


Рисунок Ошибка! Текст указанного стиля в документе отсутствует..1- Схема розташування контактів МК ATМega64

Для вивчення процесу розробки і основних схемних рішень можна обійтися і без STK500. Для початку заставимо МК запуситися і засвітити два світодіоди, приєднаних до порту D.

Для запуску МК треба приєднати кварцевий резонатор до входів XTAL та блокувати його конденсаторами, а також подати живлення на контакти Vcc та GND. В принципі цього вистачить, але варто приєднати ще й контакти аналогового живлення AVcc через RC фільтр. Також треба приєднати AGND прямо до землі і контакт опорної напруги AREF до Vcc також через RC-ланку. У простому проекті без аналогових сигналів обійдемося.

Для побудови схеми скористаємося програмним комплексом PROTEUS.

Запустимо програму редактора ISIS

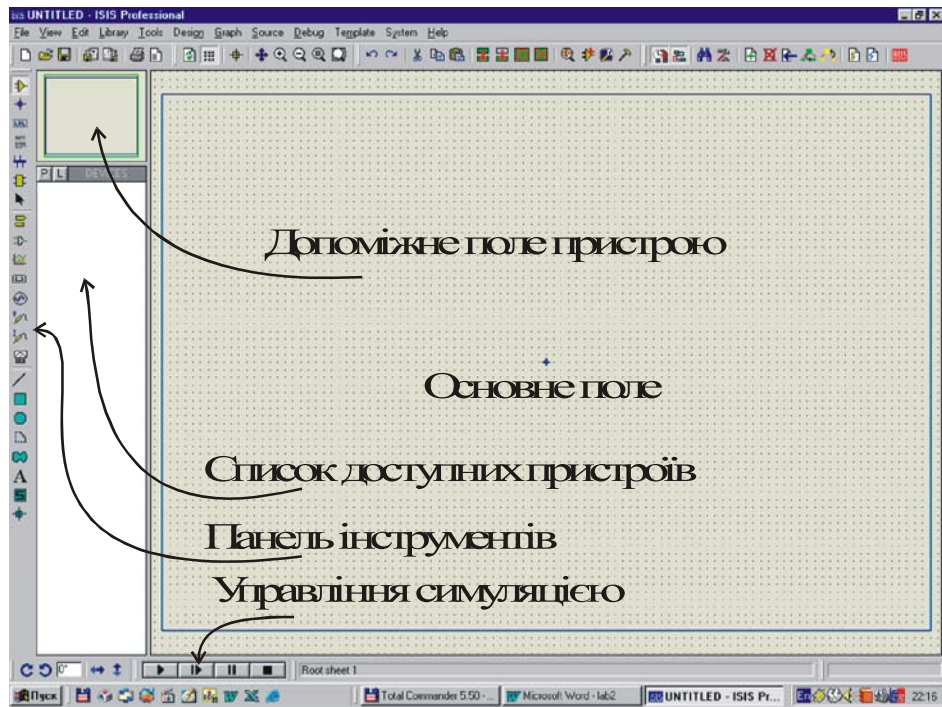
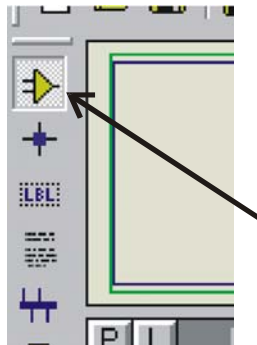
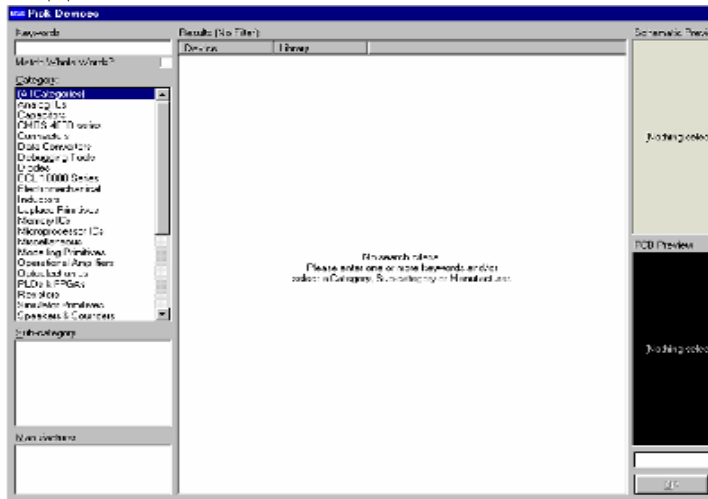


Рисунок Ошибка! Текст указанного стиля в документе отсутствует..2-Экран програми Proteus

На панелі інструментів виберемо іконку пристроїв



У списку доступних пристроїв двічі клацнемо лівою кнопкою мишки (ЛКМ). Випаде вікно зі списком бібліотек.



У полі Keywords наберемо АТМega64 у вікні Result клацнути ЛКМ і на кнопку Ok. У вікні доступних пристроїв (Device) з'явиться наша АТМega64.

Так само двічі клацнувши ЛКМ шукаємо пристрій CRYSTAL, CAP, LED-GREEN. Для набору пристроїв у проект не обов'язково повсякчас натискати Ok. Можна в полі Result двічі клацнути ЛКМ на вибраних компонентах.

Тепер виберемо МК і ЛКМ поставивши її на середину робочої області помістимо мікросхему. Правою кнопкою миші (ПКМ) помітимо МК і ЛКМ викличемо вікно параметрів МК

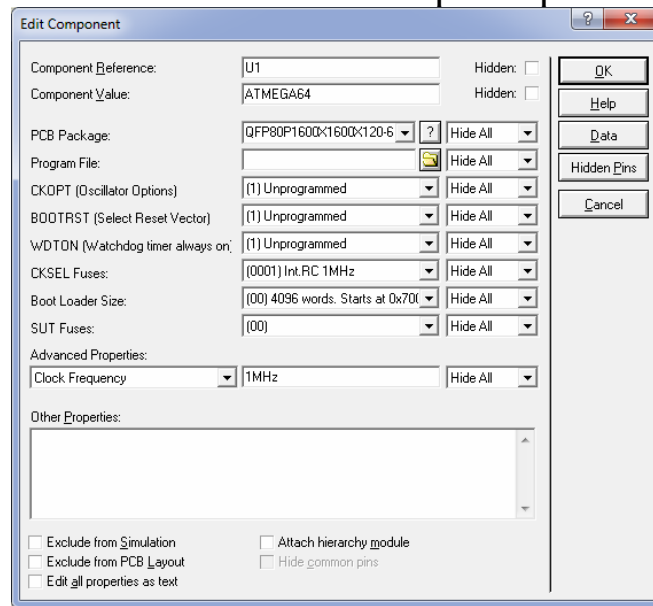


Рисунок Ошибка! Текст указанного стиля в документе отсутствует..3- Вікно параметрів компонента в Proteus

і вставимо позначення компонента схеми DD1 як на рис.1.3.

Далі виберемо резонаторі поставимо його на схема. Далі конденсатор. Для повороту конденсатора виберемо інструмент об'єкт



і ПКМ виберемо конденсатор, а далі поклацаємо в нижньому лівому куті на кнопку повороту. За допомогою інструмента BUS



протягнемо лінії між компонентами. За допомогою інструменту BUS Terminal вкажемо загальний провід GROUND та POWER. Далі намагаємось поставити світодіоди і приєднати їх до контактів порту PORTD.4 та PORTD.5. Аналогічно підключаємо через підтягуючий резистор кнопку до порту PIND.0.

У результаті повинна вийти схема

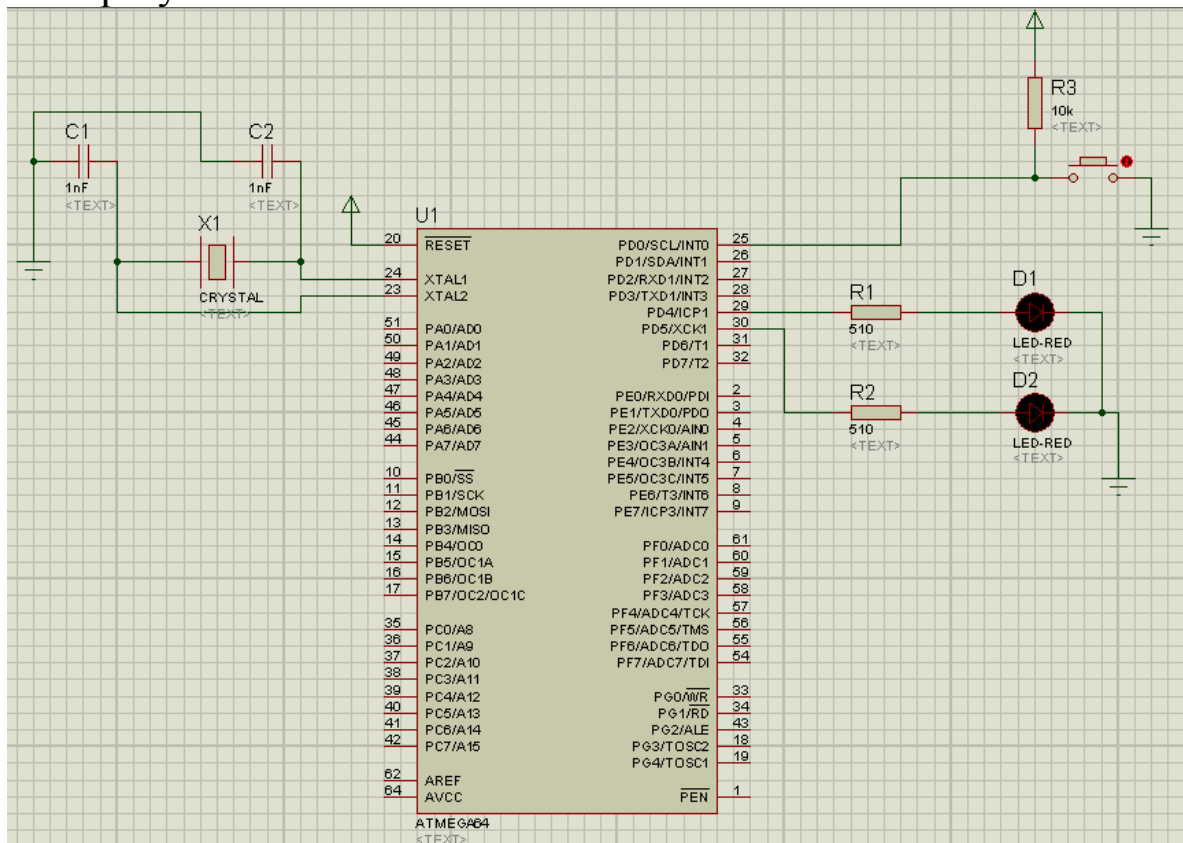
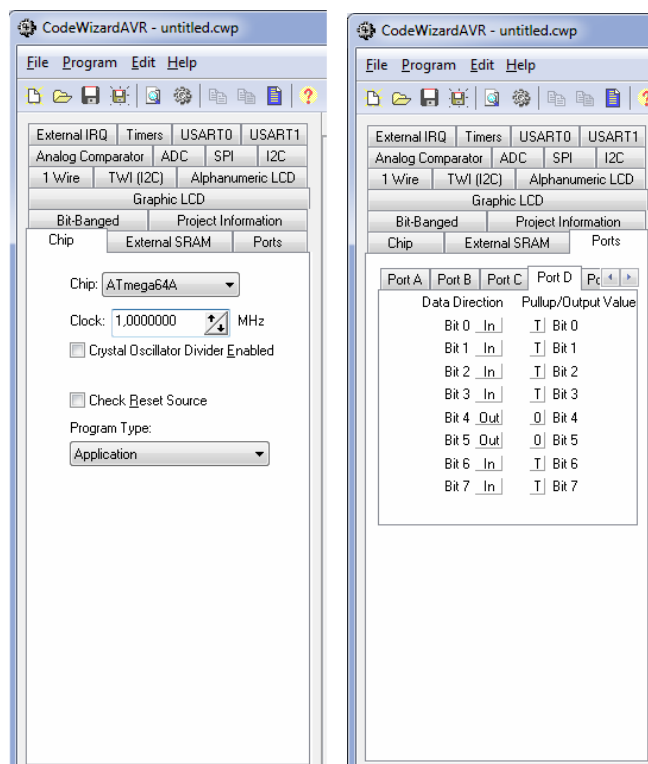


Рисунок Ошибка! Текст указанного стиля в документе отсутствует..4-Схема лабораторної установки

Такої схеми достатньо для керування роботою світлодіода.

Далі треба написати програму яка буде керувати роботою МК.

Запускаємо компілятор CodeVisionAVR. Створюємо новий проект вибираючи у віконечку Source/Project пункт Project. На пропозицію скористатися помічником при створенні проекту погоджуємося.



*Рисунок **Ошибка!** Текст указанного стиля в документе отсутствует..5- Вікно помічника (Wizard) CodeVision AVR*

У полі Chip вибираємо ATmega64, в полі Clock 1МГц. Далі вибираємо закладку Ports і ще одну підзакладку Port D. Навпроти бітів 4 та 5 клацаємо ЛКМ змінюючи стан In на стан Out. На цьому все. Записуємо стан помічника в папку свого проекту із своєю назвою. Далі генеруємо код Generate, Save and Exit. Всі назви починаються із номера лабораторії та свої позивні наприклад: Lab1NR.

Буде згенеровано проект і до нього підключено файл вихідного тексту програми. Після команди

```
#include <mega64.h>
вставимо
#include <delay.h>
```

У цьому файлі містяться описи функцій формування затримки виконання програми. Код програми побудований так, що спочатку ініціалізуються всі пристрої мікросхема, а потім запускається нескінчений цикл

```
while (1)
{
}
```

між фігурними дужками і має бути програма. Наша задача написати код, який буде по чергово засвічувати то один світлодіод то інший.

Наші діоди на PORTD.4 та POTRD.5, тому записуємо таку програму.

```
while (1)
{
    PORTD.4=1;
    PORTD.5=0;
    delay_ms(500);
    PORTD.4=0;
    PORTD.5=1;
    delay_ms(500);
}
```

Далі компілюємо проект Shift+F9.

Знову повертаємося до проекту схеми ISIS. У вікні параметрів МК в полі Program.file вибираємо шлях до об'єктного файлу вашого проекту із розширенням .cof (тут можуть бути проблеми і треба буде переписати файл *.cof та *___.c в папку із проектом Proteus).

Далі ми можемо стартувати відладчик. У пункті Debug вибираємо Start. У вікні яке з'явилося ми вибираємо вихідний текст програми і можемо натискаючи кнопку F10 виконати команди одну за другою. А Можемо натиснути на запуск (бігун на верхньому полі) і наш контролер почне виконувати програму в реальному часі. Процес виконання можна прослідкувати по сигнальних крапках червоного та синього кольору на контактах МК. Синя крапка– лог.0, а червона– лог.1. Крім того буде змінюватися зображення світлодіодів.

Щоб задіяти кнопку у роботі схеми, модифікуємо алгоритм програми. Наприклад, щоб один із світлодіодів (підключений до PORTD.5) засвічувався при натиснутій кнопці, і гаснув при вимкненій кнопці. Інший світлодіод мигає без змін. При цьому можна не враховувати затримки на переключення світлодіоду підключеного до PORTD.5. Для реалізації даного алгоритму можна використати команду if(...)... else...

Приклад програми.

```
while (1)
{
    if (PIND.0==0) PORTD.5=1; else PORTD.5=0;
    PORTD.4=1;
    delay_ms(500);
}
```



```

PORTD.4=0;
delay_ms(500);
}

```

Самостійно дослідити роботу програми і схеми в цілому.

1.2 Порядок виконання роботи

Отримати допуск до виконання роботи, який передбачає наявність коротких теоретичних відомостей до роботи, їх знання та усвідомлення того, що треба робити.

Побудувати проект для управління трьома світлодіодами, кнопкою і відладити його згідно отриманого варіанту.

1.3 Форма звітності по роботі

Звіт по роботі повинен містити:

- тему і мету роботи,
- короткі теоретичні відомості,
- завдання,
- алгоритм роботи програми,
- текст програми,
- скріншоти коду і симулятора.

Таблиця 1.1 – Варіанти індивідуальних завдань

№ п/п	Якщо кнопка натиснута	Якщо кнопка не натиснута
1	Мигати СД1 (СД2, СД3, виключені)	Мигати СД2 і СД3 (загоряються і гаснуть одночасно), СД1 виключений.
2	Мигати СД1 і СД2 (загораються і гаснуть одночасно) СД3 виключений	Мигати СД3 (СД1, СД2 виключений)
3	Мигати СД1, СД2 і СД3 одночасно	Всі СД виключені
4	Мигати по чергово СД1 і СД2 (коли СД1 загорається, СД2 гасне і навпаки), СД3 виключений	СД3 включений, СД1, СД2 в довільному стані
5	Мигати по чергово СД2 і СД3 (коли СД2 загорається, СД3 гасне і навпаки), СД1 включений	Мигати СД1 (СД2, СД3 в довільному стані)
6	СД1, СД2 ключені, СД3 виключений	Мигати СД1, СД2, СД3 одночасно
7	Мигати СД2 (СД1 включений, СД3 виключений)	Мигати СД3 (СД1 виключений, СД2 включений)
8	Мигати СД2 і СД3 (загораються і гаснуть одночасно), СД1 включений	СД2, СД3 включені, СД1 виключений
9	Мигати СД2, при цьому СД1, СД3 в довільному стані	Мигати по чергово СД1 і СД3 (коли СД1 загорається, СД3 гасне і навпаки), СД2 включений
10	Мигати СД1 , СД2, СД3 одночасно	Мигати СД1 і СД2 (загораються і гаснуть одночасно), СД3 включений

СД1 – зелений світлодіод, СД2 – червоний світлодіод, СД3 – синій світлодіод.

1.4 Контрольні запитання до роботи

Яка відмінність між мікропроцесором та мікроконтролером?

Який об'єм пам'яті даних в АТmega64?

Які периферійні пристрої вмонтовано в МК АТmega64?

Які програмні засоби використовуються при проектуванні МК пристроїв?

Що таке UART, SPI?

Тривалість заняття: 2 год.