

К.т.н., доцент Плахотний М.В., магістрант Наконечний А.О.

**Національний технічний університет України
«Київський Політехнічний Інститут»**

ДОСЛІДЖЕННЯ КОНТРОЛЕРІВ ATmel AVR У НАВЧАЛЬНОМУ ПРОЦЕСІ

Вступ

Сімейство контролерів ATmel AVR широко використовується у різних галузях. Ці контролери користуються найбільшим попитом в автомобільній промисловості (системах захисту та безпеки, системах контролю двигуна, перемикача передач, периферійних системах) [1]. Також контролери використовуються в проміжних розробках, таких, як плата Arduino, що використовуються для проектування об'єктів з великою кількістю ліній керування та контролю і ґрунтується на використанні контролера Atmega 168 [2]. Контролери AVR використовуються також при проектуванні невеликих систем, роботів домашнього застосування [1].

Одною з головних особливостей цих контролерів є низький рівень енергоспоживання, висока надійність, а також те, що програмне забезпечення для програмування і відлагодження розповсюджується безкоштовно. Активно розвивається проект компілятора мови програмування Cі — WinAVR, наявність якого сприяє підвищенню швидкості розробки програмного забезпечення [3].

Дослідження контролерів ATmel AVR у навчальному процесі є актуальною задачею з точки зору подальшого використання навичок роботи з ними в ході професійної діяльності студентів, проектування систем керування на їх основі, та на основі інших контролерів.

Постановка задачі

Метою роботи є розробка методики дослідження контролерів Atmel AVR в навчальному процесі, з подальшим використанням набутих навичок у проектуванні систем керування промисловими об'єктами на їх основі та на основі інших контролерів.

Методика дослідження повинна охоплювати наступні аспекти роботи контролера: стандартний набір інструкцій, структуру програми на асемблері, роботу з зовнішньою і внутрішньою пам'ятю, периферійні

засоби контролера і роботу з портами введення-виведення, використання компілятора мови програмування Сі для розробки програм.

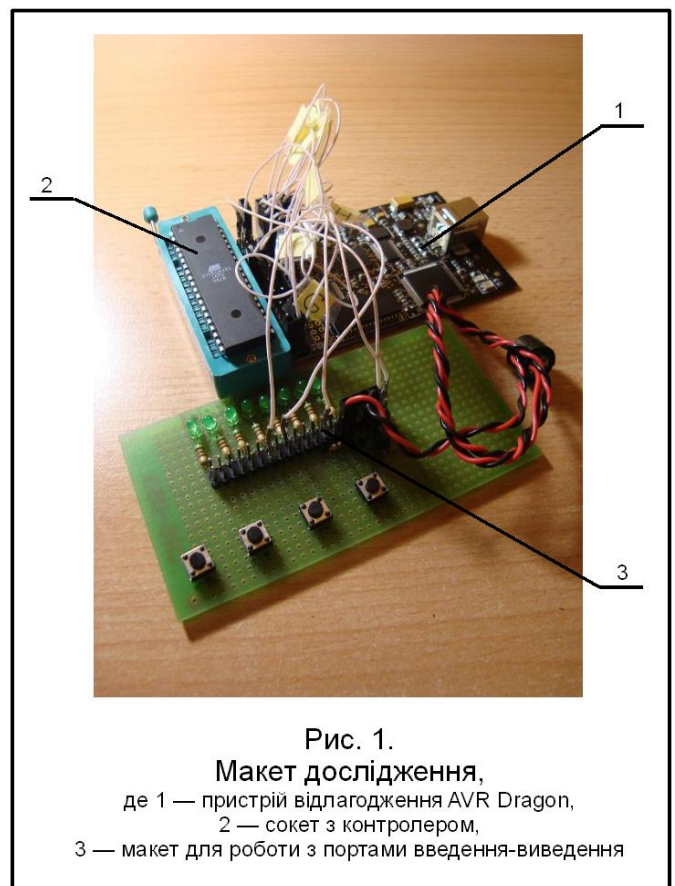
Засоби і методи вирішення

Контролери ATmel AVR є контролерами восьмибітної RISC архітектури, містять вбудовану RAM та EEPROM пам'ять та забезпечують можливість роботи із зовнішньою пам'ятю. Є можливість програмування та відлагодження контролерів по інтерфейсах: ISP, JTAG, HVPROG [3].

З периферійних засобів в контролерах звичайно є декілька лічильників/таймерів, аналогові компаратори, асинхронні прийомопередавачі, контрольний таймер. Більшість виводів контролерів можуть використовуватися, як однонаправлені порти для введення або виведення, кожний вивід яких може бути настроєно на введення або виведення незалежно від інших виводів порту [4].

В ході дослідження AVR контролерів виконувалося програмування контролера ATmega 162 з використанням пристрою вілагодження AVR Dragon, що є одним з найпростіших за функціональністю і низьковартісних пристроїв для програмування таких контролерів [5]. Для виконання програм та відлагодження роботи контролера використовувався інтерфейс JTAG.

В прототипній частині пристрою відлагодження будло змонтовано сокет (рис. 1) для контролера та 40 контактів, які на платі з'єднані з виводами контролера. Для написання програм, програмування та відлагодження контролера використовувалося графічне середовище розробки AVR Studio, яке є у відкритому безкоштовному доступі на сайті виробника. Це середовище розробки дає можливість виконувати програми покроково, після кожного оператора змінювати вручну значення в регістрах загального призначення, портах введення-виведення. Також в будь-який момент часу можна отримати дамп пам'яті контролера в текстовому вигляді, або ж навпаки –



завантажити вміст пам'яті в пам'ять контролера.

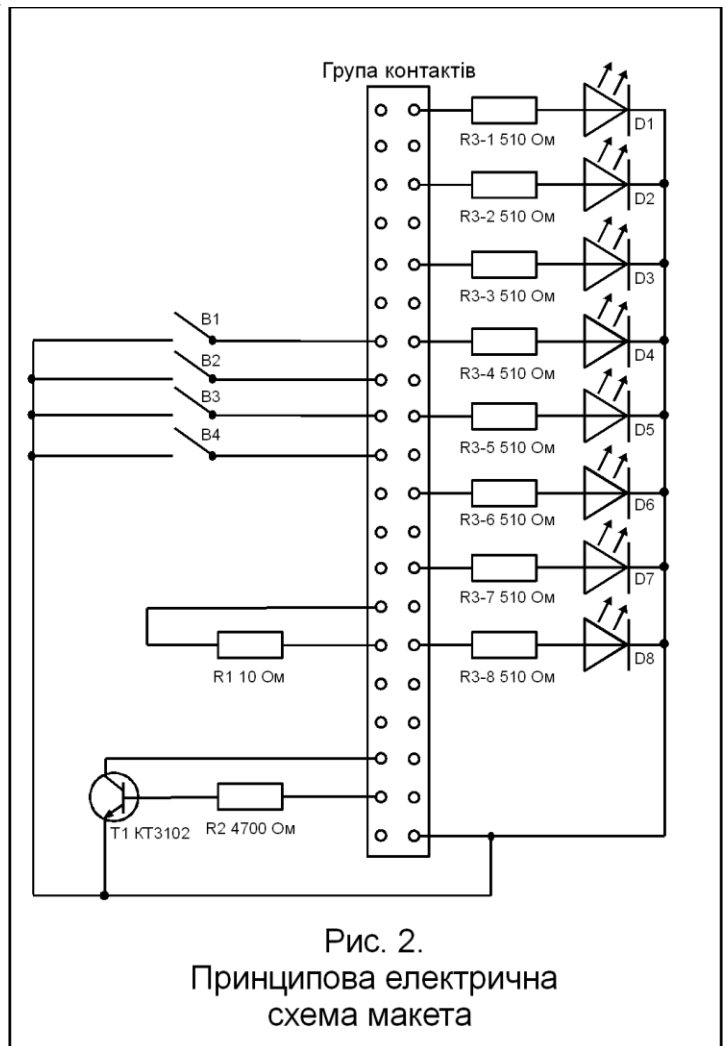
Дослідження можна умовно розбити на дві частини: робота з пам'ятю EEPROM та робота з портами введення-виведення. Для демонстрації роботи портів введення-виведення було розроблено і виготовлено макет (рис. 1) на монтажній платі, який містить світлодіоди, кнопки та РС-спікер.

Завдяки наявності опорної напруги на портах введення контролера нема потреби додавати в електричне коло резистори (рис. 2), достатньо вивести «землю» на другий вивод кнопки, а перший приєднати до контролера. Очевидно, що керуючим рівнем у такому випадку буде логічний нуль. Для запобігання пошкодження світлодіода в електричне коло з діодом було додано резистори.

Напруга на РС-спікер подається через керуючий транзистор. Це робиться з метою уникнення пошкодження контролера у випадку різкої зміни напруги під час перехідних процесів у спікері. Також теоретично це дає можливість подати на спікер більшу напругу, збільшуючи таким чином потужність, тобто гучність звучання спікера.

Виконання програмної частини першого етапу дослідження полягає у написанні невеликої програми на асемблері [6], яка читає число з області пам'яті, виконує над ним перетворення, що полягає у підрахунку кількості одиниць у його двійковому поданні, та записує результат в іншу область пам'яті [7].

Виконання програмної частини другого етапу дослідження полягає у встановленні компілятора WinAVR та написанні програми на мові програмування Сі. Програма циклічно оброблює сигнали портів введення, фіксує натиснення на кнопки та підсвічуючи світлодіоди. За прериванням таймера виконується виведення напруги на порт керування РС-спікером, що дає можливість отримувати звуковий



сигнал заданої частоти. Таймер працює в режимі Waveform generator.

Висновки

В ході виконання роботи було розроблено методику дослідження контролерів у навчальному процесі. Студентам для вивчення контролерів пропонується виконання завдань двох типів.

Перший тип завдань полягає у формуванні сегменту EEPROM пам'яті, читанні й виконанні перетворення над числом, розташованим в деякій області пам'яті, і запису результату в іншу область пам'яті, визначену варіантом. В якості перетворень можна використовувати наступні задачі: підрахунок кількості одиниць в двійковому поданні числа, інвертування порядку бітів в двійковому поданні числа, перестановка тетрад числа тощо.

Другий тип завдань полягає у написанні програми роботи з портами введення-виведення, використання системних таймерів та переривань.

Література

1. Atmel AVR // http://en.wikipedia.org/wiki/Atmel_AVR
2. Arduino board // <http://en.wikipedia.org/wiki/Arduino>
3. Atmel AVR 8-Bit RISC // <http://www.atmel.com/products/avr/>
4. Atmega162 Datasheet // http://www.atmel.com/dyn/products/product_card.asp?part_id=2023
5. AVR Dragon Documentation // <http://support.atmel.no/knowledgebase/avrstudiohelp/mergedProjects/AVRDragon/AVRDragon.htm>
6. AVR assembler beginners help // http://www.avr-asm-tutorial.net/avr_en/beginner/index.html
7. Work with EEPROM // http://www.avr-asm-tutorial.net/avr_en/TESTEEP.html