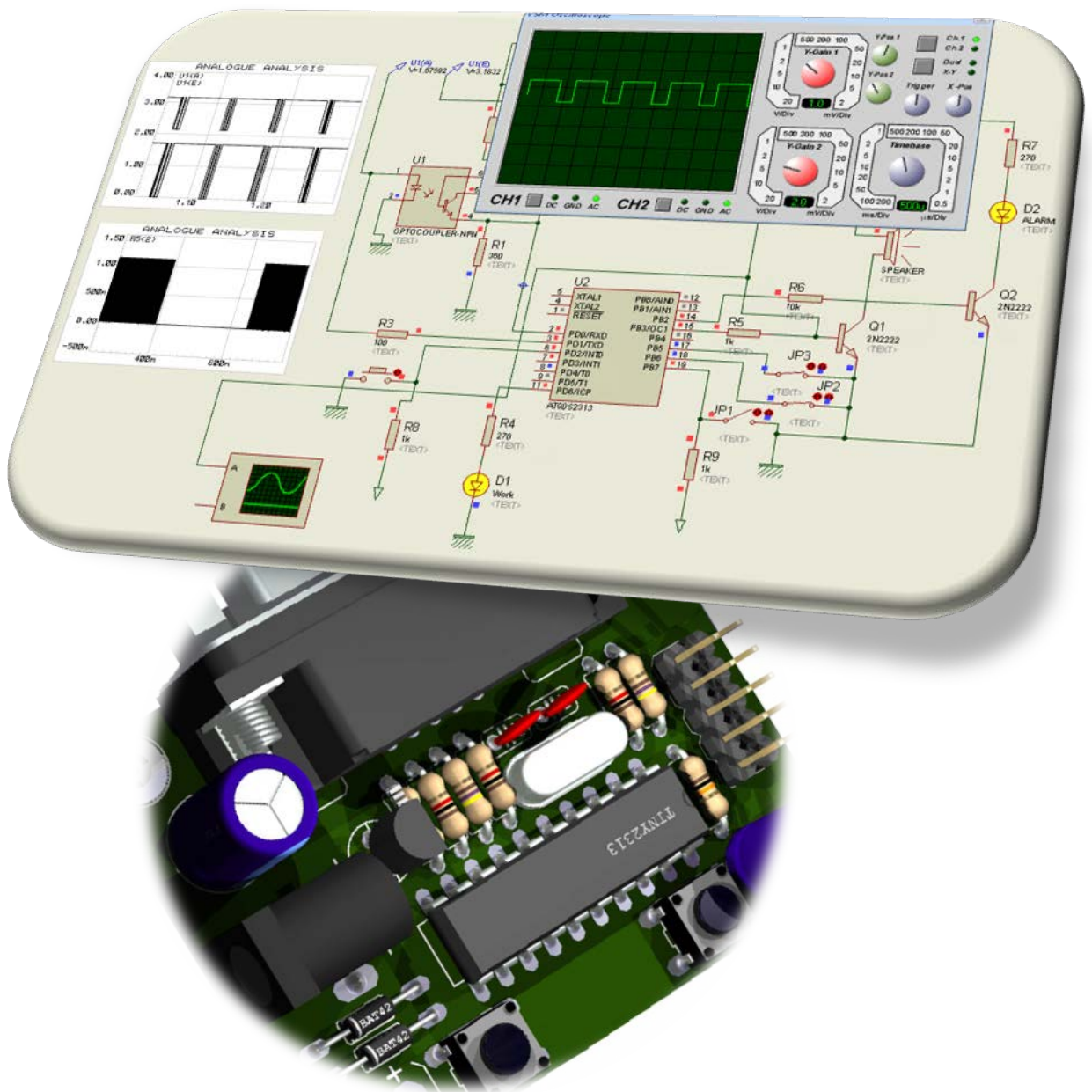


С. М. Цирульник, Г. Л. Лисенко



Проектування мікропроцесорних систем



Міністерство освіти і науки, молоді та спорту України
Вінницький національний технічний університет

ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ

Навчальний посібник

Вінниця
ВНТУ
2012

УДК 004.382.7(075.8)
ББК 32.973.26-04-018.2
Ц 71

Автори:
С. М. Цирульник, Г. Л. Лисенко

Рекомендовано Міністерством освіти і науки, молоді та спорту України як навчальний посібник для студентів вищих навчальних закладів, які навчаються за напрямом підготовки "Комп'ютерна інженерія". Лист №1/11- 6228 від 15 липня 2011 р.

Рецензенти:

А. О. Мельник, доктор технічних наук, професор
В. П. Малахов, доктор технічних наук, професор
Л. М. Карпуков, доктор технічних наук, професор

Цирульник С. М.
Ц71 Проектування мікропроцесорних систем : навчальний посібник / С. М. Цирульник, Г. Л. Лисенко. – Вінниця : ВНТУ, 2012. – 201 с.

У навчальному посібнику розглянуто особливості проектування, програмування та моделювання роботи електричних схем з використанням мікроконтролерів загального призначення виробництва Atmel, що об'єднані під загальною маркою AVR. Посібник розроблено відповідно з планом кафедри обчислювальної техніки і програмою дисципліни "Проектування мікропроцесорних систем" для студентів старших курсів та спеціалістів в галузі мікропроцесорної техніки.

УДК 004.382.7(075.8)
ББК 32.973.26-04-018.2

© С. Цирульник, Г. Лисенко, 2012

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ПЕРЕДМОВА	7
ВСТУП	9
РОЗДІЛ 1 ФУНКЦІОНАЛЬНІ ВУЗЛИ МІКРОПРОЦЕСОРНИХ СИСТЕМ	11
1.1 Виконавчі пристрої мікропроцесорних систем.....	11
1.2 Елементи індикації.....	19
1.2.1 Пристрої індикації	19
1.2.2 Організація динамічної індикації	21
1.3 Пристрої формування звукових сигналів	26
1.4 Пристрої керування двигунами постійного струму	29
1.5 Пристрої комутації.....	34
1.5.1 Аналогові ключі	34
1.5.2 Типи аналогових ключів	35
1.5.3 Застосування аналогових ключів	35
1.5.4 Аналогові мультиплектори	36
1.6 Кнопки та датчики	36
1.7 Оптичні датчики.....	44
1.7.1 Щілинний оптрон.....	44
1.7.2 Відбивальний оптрон.....	45
1.7.3 Швидкість спрацьовування оптронів	45
1.7.4 Коефіцієнт підсилення за струмом	46
1.7.5 Особливості роботи в інфрачервоному діапазоні	47
1.7.6 Дискретні оптичні датчики	48
1.8 Цифроаналогові перетворювачі.....	51
1.8.1 Загальні відомості	51
1.8.2 Інтерфейси цифроаналогових перетворювачів.....	53
1.9 Аналого-цифрові перетворювачі	56
1.9.1 Загальні відомості	56
1.9.2 Опис АЦП	58
1.9.2.1 Опорна напруга	58
1.9.2.2 Вихідне слово	59
1.9.2.3 Роздільна здатність	59
1.9.3 Типи АЦП	60
1.9.3.1 Слідкувальний АЦП	60

1.9.3.2 Паралельний АЦП	62
1.9.3.3 АЦП послідовного наближення	62
1.9.3.4 АЦП подвійного інтегрування	63
1.9.4 Інтерфейси АЦП	65
1.9.4.1 Кодування вихідного слова	65
1.9.4.2 Паралельний інтерфейс	65
1.9.4.3 Послідовний інтерфейс SPI	67
1.9.4.4 Шина I ² C	70
1.9.4.5 Шина 1-Wire	72
1.9.4.6 Спеціалізовані послідовні інтерфейси.....	83
РОЗДІЛ 2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ	
МІКРОКОНТРОЛЕРІВ.....	84
2.1 Середовища розробки програмного забезпечення для	
мікроконтролерів AVR	84
2.1.1 Компілятори мови Assembler.....	84
2.1.2 Компілятори мови C	85
2.1.3 Компілятори Basic та Pascal.....	87
2.2 Програмування мовою асемблера в середовищі AVR Studio.....	88
2.2.1 Створення програм у AVR Studio 3.0	88
2.2.2 Робота з графічним інтерфейсом користувача AVR	
STUDIO 4	91
2.3 Програмування мовою C у середовищі WinAVR.....	93
2.3.1 Створення файлу вихідного тексту програми.....	93
2.3.2 Створення makefile	94
2.3.3 Компіляція та лінування.....	96
2.4 Програмування мікроконтролерів AVR у середовищі	
MicroBasic та MicroPascal.....	98
2.4.1 Редактор коду.....	98
2.4.2 Налаштовувач (Debugger)	100
2.4.3 Вікно помилок (Error Window).....	103
2.4.4 Статистика.....	103
2.4.5 Інтегровані інструменти	104
2.4.6 Гарячі клавіші (Keyboard Sortcuts).....	105
2.4.7 Проекти (Projects)	107
2.4.8 Основні файли (Source files).....	107
2.4.9 Компіляція (Compilation).....	107
2.5 Програмування мікроконтролерів AVR у середовищі	

AVR Builder.....	107
2.5.1 Середовище розробки AVR Builder	107
2.5.2 Розробка програмного забезпечення за допомогою AVR Builder.	108
2.5.3 Приклад застосування AVR Builder.....	111
РОЗДІЛ 3 ОСОБЛИВОСТІ ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ.....	115
3.1 Технічне завдання на розробку двоканального терморегулятора	115
3.2 Розробка структурної схеми і алгоритму функціонування терморегулятора. Вибір типів датчиків і виконавчих пристроїв	116
3.3 Вибір елементної бази і розробка схеми електричної принципової терморегулятора	122
3.4 Розробка програмного забезпечення.....	136
РОЗДІЛ 4 НАЛАГОДЖЕННЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ У СЕРЕДОВИЩІ PROTEUS.....	144
4.1 Моделювання мікроконтролерів MC68HC11 фірми Motorola	162
4.2 Моделювання мікроконтролерів сімейства x51	165
4.3 Моделювання мікроконтролерів сімейства PIC.....	168
4.4 Моделювання мікроконтролерів сімейства AVR	171
ІНДИВІДУАЛЬНІ ЗАВДАННЯ НА КУРСОВЕ ПРОЕКТУВАННЯ	187
СПИСОК ЛІТЕРАТУРИ.....	193

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АЦП – аналого-цифровий перетворювач
ЕРС – електрорушійна сила
ІМС – інтегральна мікросхема
ІС – інтегральна схема
ІЧ – інфрачервоний
КПС – коефіцієнтом передачі за струмом
МБ – молодший байт вхідного слова
МДН – метал – діелектрик – напівпровідник
МК – мікроконтролер
МП – мікропроцесор
МПП – мікропроцесорний пристрій
МПС – мікропроцесорна система
ОЗП – оперативний запам'ятовувальний пристрій
ОП – операційний підсилювач
ПВЗ – пристрій вибірки-зберігання
ПЗ – програмне забезпечення
РЕП – радіоелектронний пристрій
СБ – старший байт вхідного слова
ТЗ – технічне завдання
ТТЛ – транзисторно-транзисторна логіка
ТХА – термопари хромель-алюмель
ТХК – термопари хромель-копель
ХА – хромель-алюмелевий
ХК – хромель-копелевий
ЦАП – цифроаналоговий перетворювач
ШИМ – широтно-імпульсна модуляція
АВ – Adress Bus (шина адреси)
ADC – аналого-цифровий перетворювач
ВР – Breakpoint (точка зупинки програми)
DB – Date Bus (шина даних)
EEPROM – електрично-стираний програмований постійний запам'ятовувальний пристрій
МС – мікроконтролер
MCU – мікропроцесор
PN – Programmers Notepad

ПЕРЕДМОВА

Із часів виходу на ринок першого мікропроцесора 8080 компанії Intel, що широко застосовується в промисловості, розвиток наступних поколінь пішов по двох головних напрямках. З однієї сторони, це такі високопродуктивні процесори, як Pentium компанії Intel або сімейство процесорів 68000 компанії Motorola, які знаходять застосування в персональних комп'ютерах, робочих станціях і наукових обчислювальних установках, а також реалізуються в архітектурі комп'ютерів зі складним набором команд CISC (CISC - Complex Instruction Set Computer). З іншої сторони, швидко позначилася велика потреба в більш простих мікроконтролерах, для яких компактність важливіше здатності обробити величезний потік команд.

Компанія Intel ще в 1976 році вивела на ринок однокристальний мікроконтролер 8048, у якого в одному кристалі інтегрований набір команд 1 Кбайт (ROM), 64-байтовий ОЗП (RAM), 27 портів введення/виведення, восьмирозрядний таймер, схема переривань, а також кварцовий генератор.

Для високопродуктивних процесорів, на зразок Pentium (які, втім, націлені на інший сегмент ринку), необхідно велике число таких зовнішніх компонентів, як завантажувальний ПЗП (ROM), RAM як робоча пам'ять, жорсткий диск як пристрій для зберігання програм, таймер, схема переривань і т.д., які повинні бути розміщені в одному корпусі. У випадку ж однокристального мікроконтролера для роботи не потрібні ніякі інші елементи.

За ціною також намітилася істотна різниця. У той час як процесори, на зразок Pentium, внаслідок складної побудови й дорогого процесу виготовлення, пропонуються за ціною в кілька сотень доларів, вартість однокристального мікроконтролера перебуває в межах від одного до 25 доларів.

Однокристальні мікроконтролери знаходять широке застосування в найрізноманітніших сферах: від вимірювальних приладів, фотоапаратів і відеокамер, принтерів, сканерів і копіювальних апаратів до виробів електронних розваг і всілякої домашньої техніки, у результаті чого вони одержали загальну назву "процесор для пральних машин".

З огляду на велику сферу застосування й, відповідно, великі обсяги продажів однокристальних мікроконтролерів, немає нічого дивного в тому, що практично кожний виробник напівпровідників пропонує на ринку свій власний "тип". Проте у вісімдесяті роки минулого століття на ринок зміг вийти восьмирозрядний процесор 8051 компанії Intel з усіма його варіаціями. Навіть незважаючи на те,

що згодом на ринку з'явився конкурент – 16-розрядний мікроконтролер, I8051, як і раніше втримує за собою пальму першості, хоча останнім часом його й потіснили восьмирозрядні RISC-контролери нового покоління (RISC – Reduced Instruction Set Computer – комп'ютер зі скороченим набором команд). До таких RISC-мікроконтролерів належить сімейство PIC компанії Microchip, а також сімейство AVR компанії Atmel.

Можна заздалегідь відзначити, що компанія Atmel, завдяки комбінації прогресивної архітектури RISC і програмованої Flash-пам'яті EPROM розробила дуже ефективне рішення для всіх можливих завдань внутріплатного керування та встановила нові масштаби у співвідношенні ціни й продуктивності.

Навчальний посібник призначений для студентів технічних спеціальностей усіх форм навчання, інженерів і вчених, а також для всіх, хто вже має певні основні пізнання про побудову й функціонування мікроконтролерів, бажає зрозуміти й вивчити однокристальні мікроконтролери AVR і успішно запровадити в життя власні ідеї, щоб не відстати від "поїзда", що рухається в технологічне майбутнє.

У першому розділі розглядаються схемотехнічні аспекти побудови пристроїв на мікроконтролерах.

Другий розділ присвячений огляду програмного забезпечення, що дозволяє створювати програми для мікроконтролерів AVR мовами програмування: Assembler, Basic, Pascal, C.

У третьому розділі наводиться приклад проектування пристрою для вимірювання і підтримки в заданому діапазоні температури в нагрівальній камері промислової установки на мікроконтролері AVR.

У четвертому розділі показані особливості налагодження як апаратної частини, так і програмного забезпечення мікропроцесорних систем у програмі автоматизованого схемотехнічного проектування Proteus.

ВСТУП

Мікроконтролер можна уявити у вигляді непотрібного шматка кремнію, що має проте величезний потенціал. Поки в ньому немає програми, він нічого не буде робити, однак при її наявності він зможе виконувати практично будь-які функції. Досить велика принципова схема може перетворитися у звичайну програму, зменшивши в такий спосіб цілий пристрій до однієї-єдиної мікросхеми. Мікроконтролери ліквідують розрив між апаратним і програмним забезпеченням – вони виконують програму як звичайний комп'ютер, хоча в той же час є дискретними елементами, які можуть взаємодіяти з іншими компонентами схеми. За кілька років мікроконтролери стали невід'ємною частиною інструментарію радіоінженерів і величезного числа радіоаматорів, оскільки вони чудово підходять для експериментування, дрібносерійного виробництва й реалізації проектів, що вимагають певної гнучкості виконуваних функцій.

Мікроконтролери AVR – це одні з найшвидкодійних мікроконтролерів у світі. На даний час співвідношення "ціна – продуктивність – енергоспоживання" для AVR є одним із кращих на світовому ринку 8-розрядних мікроконтролерів. Обсяги продажів AVR у світі подвоюються щорічно. У геометричній прогресії зростає число сторонніх фірм, що розробляють і випускають різноманітні програмні й апаратні засоби підтримки розробок для них.

Галузі застосування AVR мікроконтролерів різноманітні. Для сімейства "tiny" – це інтелектуальні автомобільні датчики різного призначення, іграшки, ігрові приставки, материнські плати персональних комп'ютерів, контролери доступу в мобільних телефонах, зарядні пристрої, детектори диму і полум'я, побутова техніка, різноманітні інфрачервоні системи дистанційного керування. Для сімейства "classic" – це модеми різних типів, сучасні зарядні пристрої, вироби класу Smart Cards і пристрою читання для них, супутникові навігаційні системи для визначення місця розташування автомобілів на трасі, складна побутова техніка, пульти дистанційного керування, мережеві карти, материнські плати комп'ютерів, стільникові телефони нового покоління, а також різні та різноманітні промислові системи контролю і керування. Для "mega" AVR – це аналогові (NMT, ETACS, AMPS) і цифрові (GSM, CDMA) стільникові телефони, принтери і ключові контролери для них, контролери апаратів факсимільного зв'язку і ксероксів, контролери сучасних дискових накопичувачів, DVD-ROM і т.д.

Етапи розробки програмного забезпечення мікроконтролерів AVR наведені на рисунку В.1.



Рисунок В.1 – Етапи розробки програмного забезпечення мікроконтролерів AVR

Серед мікроконтролерів AVR є величезна кількість різних моделей, починаючи від невеликих пристроїв у 8-вивідних корпусах (сімейство "tiny") і закінчуючи мікросхемами в 40-вивідних корпусах (сімейство "tiny"). Однак найголовніше полягає в тому, що можна спокійно писати програму для однієї моделі, а потім передумати й переробити цю програму під іншу модель мікроконтролера, вносячи всього лише незначні зміни. Більше того, вивчаючи один з мікроконтролерів AVR, учимося працювати з усіма моделями сімейства. Зрозуміло, кожний з мікроконтролерів має свої особливості, однак в основі всіх моделей лежить загальне ядро.

Загалом кажучи, програмування мікроконтролерів AVR полягає в різних маніпуляціях даними. Завдання програмування полягає в тому, щоб змусити мікроконтролер виконувати поставлене завдання шляхом простих переміщень чисел і здійснення операцій над ними. Існує обмежений набір операцій, які можна виконувати над даними, – ці операції називаються командами. Мікроконтролер буде послідовно перебирати ці команди, виконуючи мільйони їх щосекунди і, таким чином, виконувати поставлене завдання.

РОЗДІЛ 1 ФУНКЦІОНАЛЬНІ ВУЗЛИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

1.1 Виконавчі пристрої мікропроцесорних систем

Практично жодна мікропроцесорна система (МПС) не може обійтися без таких елементів, як виконавчі пристрої. Головне призначення будь-якої системи – це керування яким-небудь зовнішнім механізмом. Це можуть бути електродвигуни, нагрівачі, електромагнітні клапани і т.п. Тому окрім датчиків, кнопок керування і елементів індикації до мікроконтролера (МК) обов'язково доведеться підключати і виконавчі пристрої. Для керування зовнішніми пристроями використовуються ті ж самі порти введення/виведення МК, які працюють на виведення. Сигнали з будь-якої лінії будь-якого порту легко можуть бути використані для вмикання і вимикання зовнішнього пристрою. Необхідно лише підсилити сигнал керування за потужністю до необхідного рівня. Для цього застосовуються різні схеми узгодження. Вибір схеми залежить від типу виконавчого пристрою.

У найпростішому випадку можна застосувати транзисторний ключ (рис. 1.1). При використанні транзистора КТ315Г можна керувати зовнішніми колами зі струмом споживання до 100 мА і напругою $U_{\text{ж}}$ до 15 В [5, 6]. Транзистор допускає також високу напругу, проте підвищення напруги можливе при зменшенні струму.

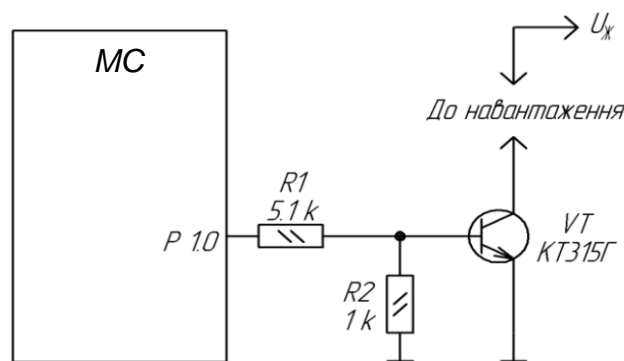


Рисунок 1.1 – Найпростіший транзисторний ключ

Для керування ланками з великим струмом потрібно застосувати потужніший транзистор. При виборі транзистора не забувайте, що максимально допустимий струм навантаження для будь-якого з виходів МК не повинен перевищувати величини 20 мА [9, 10, 20, 24]. При складанні програми потрібно не забувати, що будь-який транзисторний ключ інвертує сигнал. Якщо на виході P1.0 (див. рис. 1.1) встановити високий рівень "1", ключ відкривається і

навантаження підключається до джерела живлення. При нульовому рівні "0" на тому ж виході ключ закривається і навантаження відключається.

Якщо виконавчий механізм, яким повинна керувати МПС, живиться від мережі змінного струму 220 В, потрібно застосовувати схему керування з гальванічною розв'язкою. Один з можливих варіантів – релейна схема керування.

Типовий варіант схеми керування з використанням реле наведений на рисунку 1.2. На схемі показаний електронний ключ, в навантаження якого включено електромагнітне реле *K1*. МК за допомогою ключа може вмикати і вимикати електромагнітне реле. Контакти реле, в свою чергу, керують навантаженням. Така схема забезпечує комутацію достатньо великої напруги і струму.

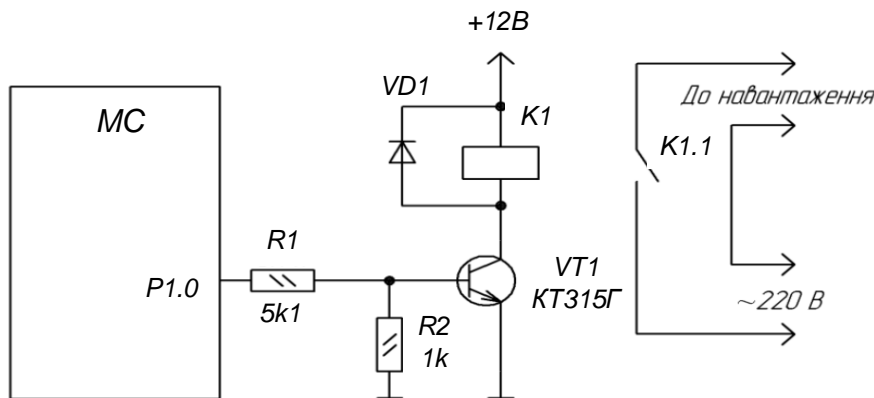


Рисунок 1.2 – Виконавчий пристрій з використанням реле

Гальванічна розв'язка між всіма колами МПС і силовою мережею 220 В забезпечує безпеку роботи з цією схемою. Діод *VD1* призначений для захисту елементів схеми від напруги ЕРС самоіндукції, що виникає в котушці реле *K1* у момент закривання ключа *VT1*. При виборі електромагнітного реле необхідно звертати увагу на такі параметри. По-перше, напруга спрацьовування реле. Для прикладу на рисунку 1.2 вона має бути рівна 12 В. По-друге, максимально допустимий струм комутації і максимально допустима напруга для виконавчих контактів реле. Вони повинні відповідати реальним значенням струму і напруги в колі навантаження.

У випадку, якщо необхідно керувати великим числом реле, або інших потужних навантажень, зручно застосовувати мікросхеми *ULN2003* або *ULN2803* [1,12]. Ці мікросхеми містять, відповідно, 7 і 8 транзисторних ключів на складених транзисторах (схема Дарлінгтона). Вони дозволяють керувати навантаженням до 500 мА при напрузі до 50 В. При цьому входи цих мікросхем можна підключати безпосередньо до ліній портів введення/виведення МК.

У середині мікросхем вже є вбудований захисний діод, який можна підключати або відключати, здійснюючи зовнішні з'єднання. На рисунку 1.3 наведений приклад схеми з використанням мікросхеми ULN2003.

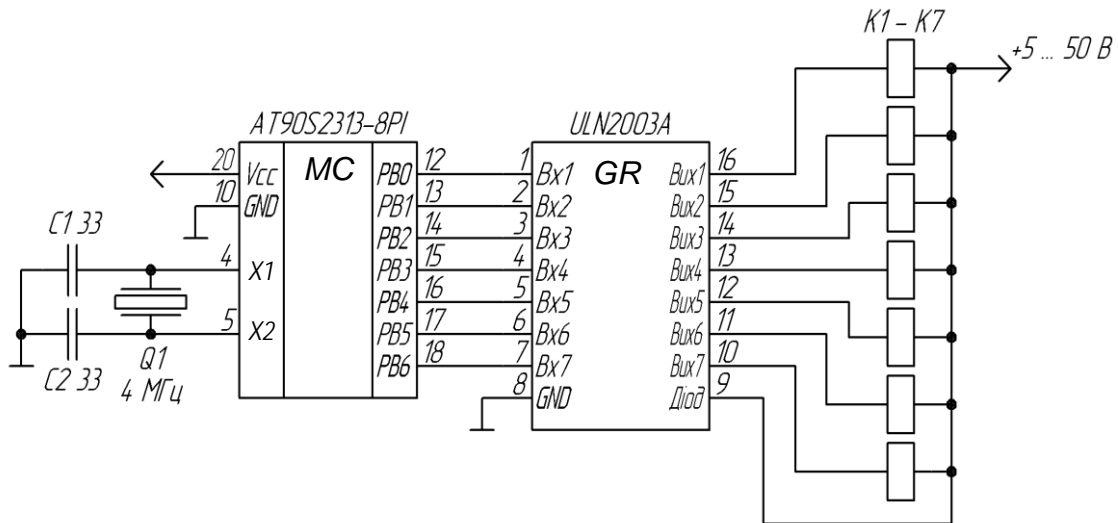
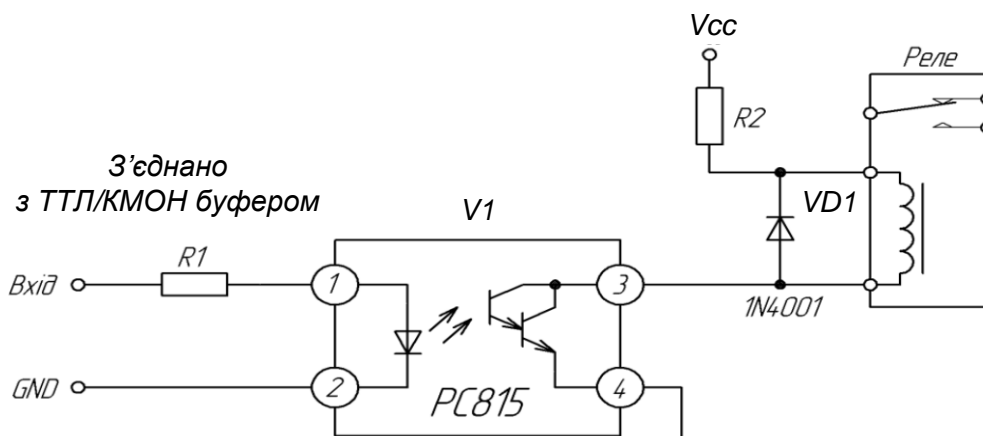


Рисунок 1.3 – Використання мікросхеми ULN2003



Максимальний струм колектора 80 мА
R2 Залежить від типу реле і напруги живлення
 На вході «1» - контакти реле замкнуті
 На вході «0» - контакти реле розімкнуті

Рисунок 1.4 – Застосування оптопары PC815 для керування реле

Оптопары можна використовувати для керування приладами з малими значеннями струмів, що вимагають гальванічної розв'язки. Максимальний робочий струм обмежений характеристиками фототранзисторів. Наприклад, оптопары Дарлінгтона серії PC815 (Sharp, RS175-198) мають максимальне значення вихідного струму 80 мА, достатнє для керування слабкострумним реле, яке, у свою

чергу, здатне працювати з потужнішими приладами (рис. 1.4). Оптопари Дарлінгтона серії *PS2502* (NEC, *RS590-424* і *RS590-430*) підтримують струми до 160 мА [1].

Коефіцієнт передачі зазвичай досягає до 2000. Напряга, що керує світлодіодом, дорівнює 1,1 В, максимальний робочий струм – 80 мА. Максимальна напряга між колектором і емітером фототранзистора складає 40 В, а час насичення – 100 мкс.

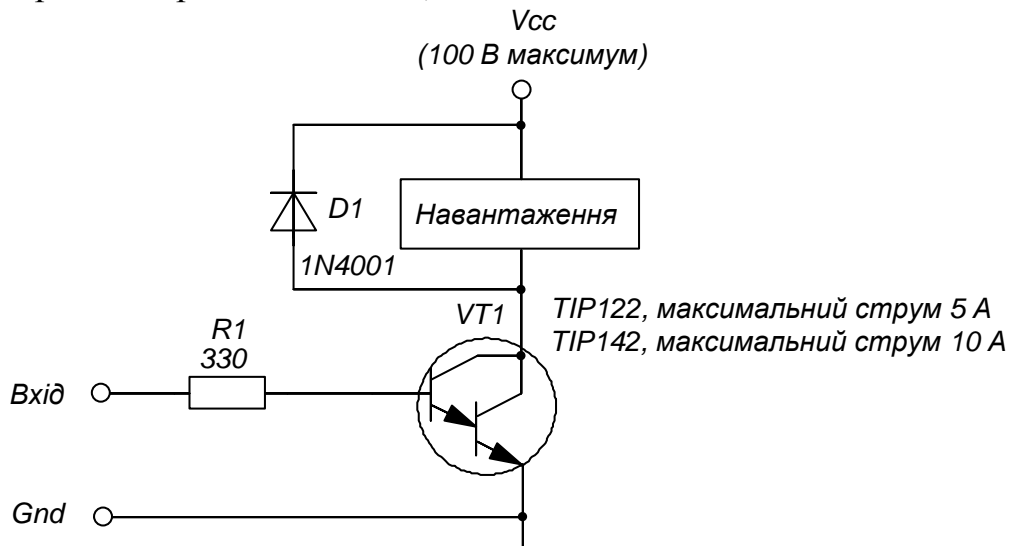


Рисунок 1.5 – Пристрій керування на транзисторі Дарлінгтона

На рисунку 1.5 зображена схема на базі потужного транзистора Дарлінгтона *TIP122* або *TIP142*. *TIP122* керує напругою до 100 В і струмами до 5 А. Максимальна потужність розсіювання – 65 Вт. Транзистори Дарлінгтона відкриваються при напрузі 1,2 В між базою і колектором і можуть підсилювати струм у 5000 разів. Отже, напряга на базі, що перевищує 1,2 В, викликає режим насичення транзистора. База з'єднується з ТТЛ входом через резистор *R1*. *TIP142* функціонує при колекторному струмі 10 А. Максимальна робоча частота цих транзисторів складає 5 МГц. Для індуктивного навантаження повинні використовуватися захисні діоди.

На рисунку 1.6 зображена схема на базі польових МДН-транзисторів *VN10KM* або *VN66AF* [1]. Щоб МДН-транзистор відкрився, на нього необхідно подати пряму напругу зсуву близько 0,8 В. Прямий зсув близько 5 В істотно впливає на провідність. Оскільки вхідний опір пристрою на польовому транзисторі дуже високий, з'єднати його безпосередньо з вихідним портом МК неможливо. Таким пристроєм слід керувати через резистор. *VN10KM* витримує максимальну напругу 60 В і струм 310 мА, *VN66AF* працює при максимальній напрузі 60 В і струмі 2 А. Час насичення – близько 15 нс.

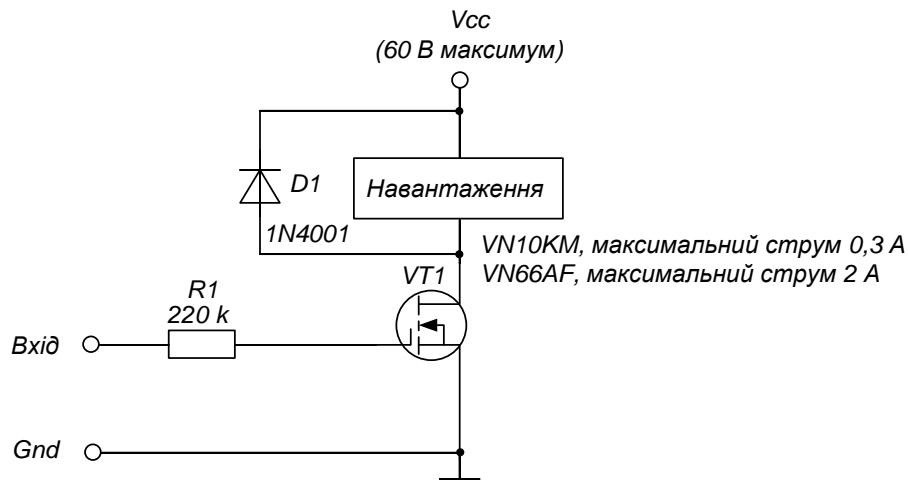


Рисунок 1.6 – Пристрій керування на польовому транзисторі

Потужні пристрої керування на базі МДН-транзисторів із захистом, які інколи називають твердотільними реле, грають роль перемикачів в силових колах цифрових систем керування. Вхідне керування сумісне з логічними рівнями 5 В. У цих елементах використовується вбудована схема термоконтролю, яка захищає їх від перегріву, короткого замикання і перевантаження. Такі пристрої, як правило, мають інформаційний вихід, який низьким рівнем інформує про спрацювання вбудованих кіл захисту.

Транзистор *BTS410* (див. рис. 1.7) здатний керувати напругою в діапазоні 4,9 - 40 В, поріг спрацювання захисту від перенапруги порядку 42 - 52 В. Максимальна робоча температура дорівнює 150 °С. Залежно від температури рівень значень струму перевантаження коливається в межах 3,1 – 21 А. Цей пристрій має низький опір у ввімкненому стані у всьому діапазоні температур. Час вмикання і вимикання складає 60 і 50 мкс відповідно. Вхідна напруга вмикання змінюється від 2 до 5 В, вимикання – від 0 до 0,8 В. Вхідний струм рівний 25 мкА при вхідній напрузі 3,5 В.

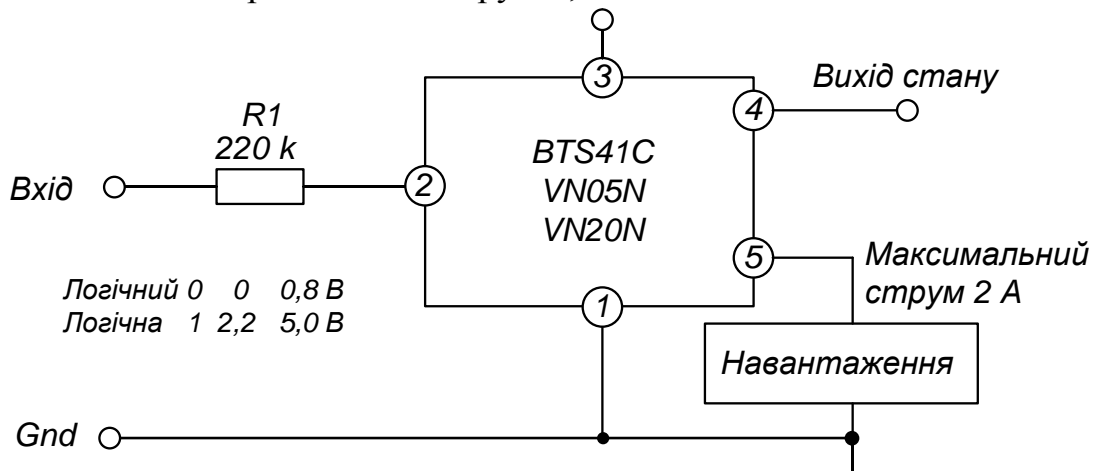


Рисунок 1.7 – Пристрій керування на МДН-транзисторі із захистом

Два інших приклади таких пристроїв – *VN05N* і *VN20N*, їх виходи і типова схема застосування зображено на рисунку 1.7 [1, 12]. *VN05N* розрахований на вихідний струм 12 А, а *VN20N* – на 28 А. У силових колах цифрових систем керування можна використовувати твердотільні реле серії *KP293* [27, 35].

Як відомо, недоліком електромагнітних реле є їх недовговічність, пов'язана з підгорянням контактів і їхнім швидким зношуванням, а також акустичні шуми в процесі роботи.

Найбільш популярні позитивні риси заміни електромагнітних реле на оптоелектронні комутаційні вузли – це малі струм і напруга керування, беззвучність і довговічність роботи, можливість роботи в середовищах постійного і змінного струмів, комутації напруги (деяких приладів) до 400...600 В і струмів до 0,5 А. На рисунку 1.8 наведена одну з таких схем.

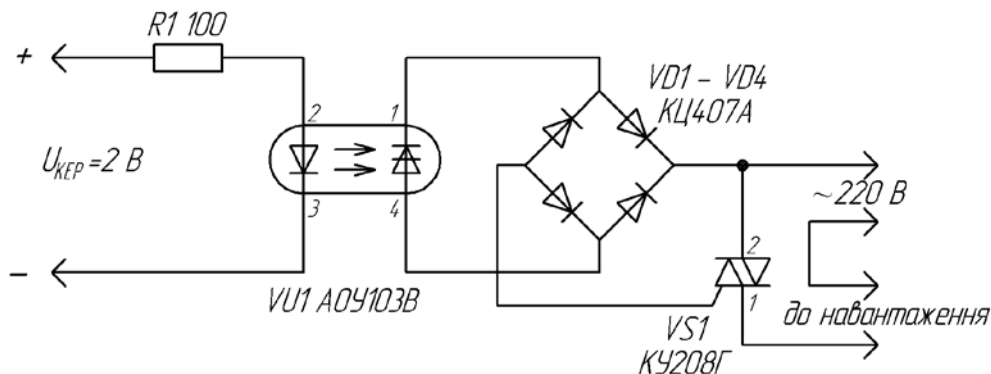


Рисунок 1.8 – Електрична схема із застосуванням оптрона

У цій схемі керування навантаженням (потужність якого може досягати 600 Вт) здійснюється симістором *KY208*. Завдяки розв'язці із живлення – застосуванню оптоелектронного приладу *AOY103B* – кола керування навантаженням в мережі 220 В і схема керування повністю розв'язані. Постійна напруга керування (або імпульси) амплітудою 2 В потрапляє від схеми керування через обмежувальний резистор *R1* на вхід оптопар *VU1*. Струм керування не перевищує 5 мА.

За наявності сигналу керування тиристор усередині оптопар відкривається (його опір в прямому напрямку зменшується до декількох десятків Ом), і він шунтує діагональ випрямного моста *VD1*. Від випрямного моста напруга проходить на електрод керування симістора *VS1*, завдяки чому він відкривається у відповідні півперіоди напруги і в навантаженні протікає струм. Використання оптопар *AOY103* залежить від напруги в електричному колі. Так, для схеми (див. рис. 1.8) та інших з напругою більше 200 В підходить лише оптопара *AOY103B*, а при меншій нарузі комутації з літерами А - до 50 В, Б – до 100 В [27].

При необхідності керування більш потужним навантаженням, наприклад до 1000 Вт, симістор як основний пристрій, що комутує навантаження в даній схемі, слід встановити на охолоджувальний радіатор.

Схожа за принципом роботи схема наведена на рисунку 1.9. Тут діагональ випрямного моста замикає оптосимістор *TO132-40* (або аналогічний *TO125-12,5*, *TO106-10* та інші). Їх основна відмінність один від одного полягає у різних струмах і потужності комутації.

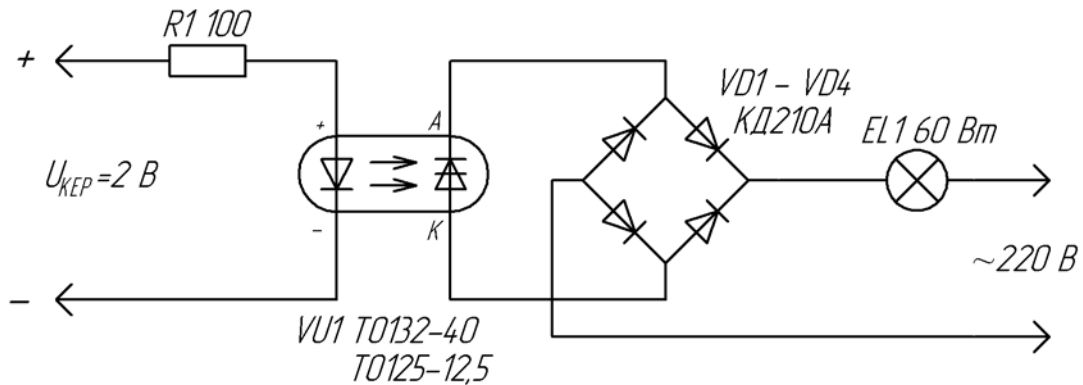


Рисунок 1.9 – Інший варіант схеми вузла керування навантаженням із застосуванням оптрона

При середньому навантаженні (більше 60 Вт) діоди слід встановити на радіатори, а при навантаженні більше 200 Вт їх слід замінити потужнішими. Струм керування оптосимістором *VU1* – 10 мА, напруга – 2...3 В.

На рисунку 1.10 показана схема керування симістором *KY208Г* за допомогою подавання на електрод *VS1* змінної напруги керування через обмежувальний резистор *R1*.

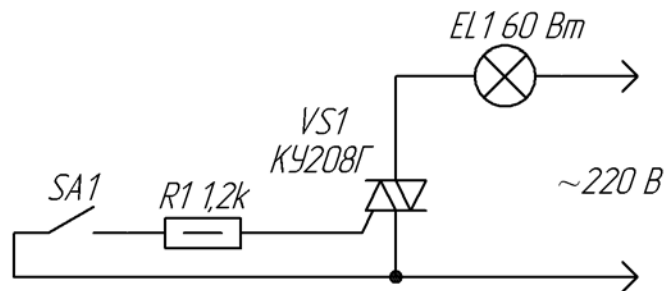


Рисунок 1.10 – Електрична схема керування симістором *KY208Г*

Поки замкнуті контакти вмикача *SA1* лампа розжарювання (навантаження *EL1*) світиться. Такий вимикач може бути корисним при дистанційному керуванні яким-небудь навантаженням або як складова частина складніших схем – адже замість вимикача *SA1*

можна використовувати електронну схему (у тому числі із застосуванням МДН-транзистора і оптореле). Симістори *KУ208Г* чутливі до "правильної" подачі напруги на елемент керування, тому, якщо вузол не працюватиме відразу, в даному випадку необхідно поміняти місцями провідники підключення в мережі 220 В (змінити фазування).

На рисунку 1.11 наведена схема гібридного керування навантаженням.

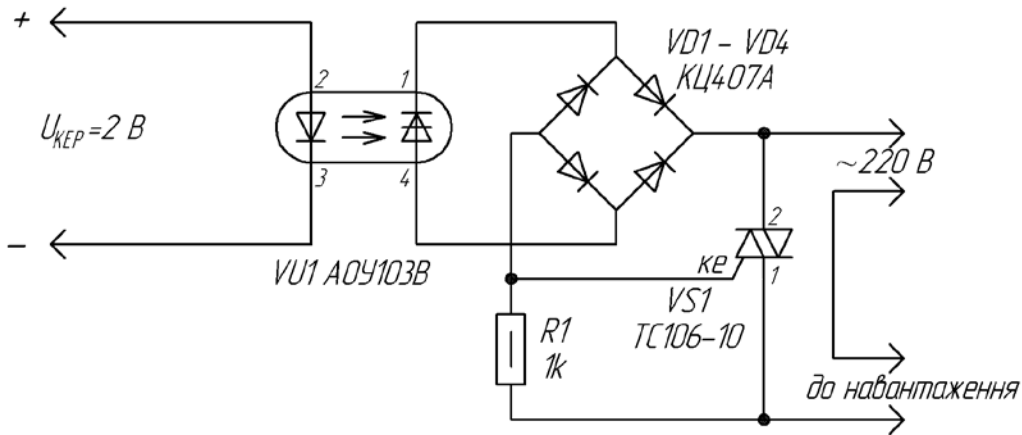


Рисунок 1.11 – Електрична схема "гібридного" керування навантаженням

Тут задіяні і оптопара тиристора *AOY103B*, і оптосимістор *TC106-10*. У результаті такого схемного рішення вузол має переваги обох схем (якщо їх розглядати окремо).

На рисунку 1.12 наведена схема для керування потужним навантаженням.

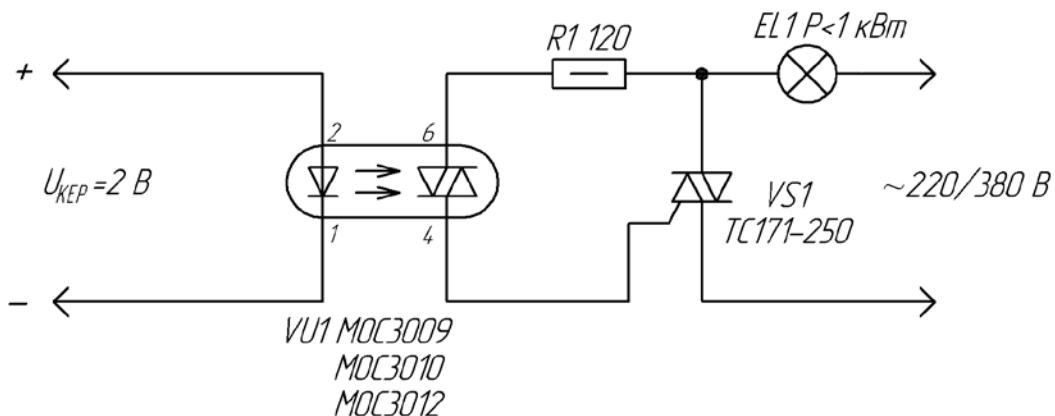


Рисунок 1.12 – Електрична схема вузла керування потужним навантаженням

Силевим елементом керування є симістор *ТС171-250*, а проміжним – оптосимістор *МОС3009* (*МОС3052*, *МОС3010*, *МОС3012* і аналогічні за електричними характеристиками). Відмінна особливість цієї схеми в тому, що напруга навантаження може складати 380...400 В, а потужність більше 1000 Вт при струмі керування до 10 мА і напрузі 2 В [1, 12].

Такий вузол без яких-небудь доповнень може керувати, наприклад, потужним двигуном.

Оптосимістори серії МОС – зарубіжні пристрої, в деяких випадках вони можуть бути замінені *АОУ103В* або оптоелектронними МДН-реле вітчизняного виробництва, наприклад, *КР293КП15В*, *К449КП1ВР*, *К449КП2ВР*, *К293КП13П* та аналогічними за електричними характеристиками.

На рисунку 1.13 показаний ще один варіант включення – поєднання оптоелектронної розв'язки із застосуванням оптопарою *АОУ103В* і симістора *КУ208Г*.

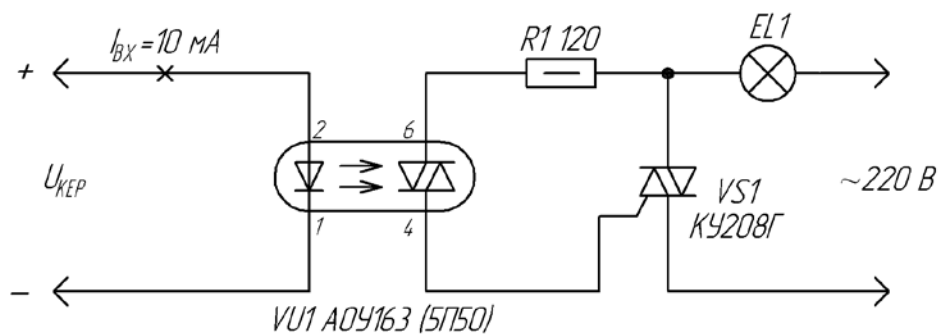


Рисунок 1.13 – Електрична схема оптоелектронної розв'язки

Керування пристроями навантаження ефективно здійснюється, якщо їх потужність не перевищує 600 Вт. Оптопара *АОУ103В* дозволяє самостійно комутувати високовольтне навантаження (з напругою до 350 В), проте струм комутації не повинен перевищувати 100 мА. Тому для керування потужним навантаженням в схему введений симістор *КУ208Г*.

1.2 Елементи індикації

1.2.1 Пристрої індикації

Практично кожен мікропроцесорний пристрій містить елементи індикації. Як індикатори в даний час найчастіше застосовуються світлодіоди. На ринку є величезний вибір світлодіодів найрізноманітніших видів і розмірів.

У МПП світлові індикатори можуть служити для відображення

різних режимів роботи: попередження про критичні ситуації, відображення ходу прийому сигналів керування тощо. Підключити одиночний світлодіодний індикатор до МК дуже просто. На рисунку 1.14 наведена схема підключення світлодіода безпосередньо до виводу порту МК.

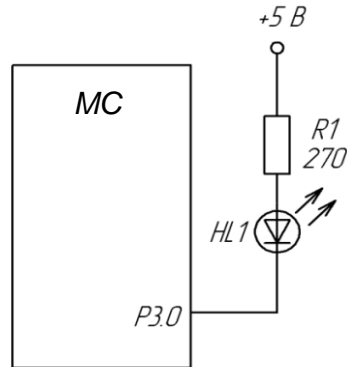


Рисунок 1.14 – Схема підключення світлодіодного індикатора

Усі вихідні каскади МК мають достатню навантажувальну здатність для того, щоб витримати підключення одного світлодіодного індикатора зі споживаним струмом у робочому режимі не більше 20 мА.

Для керування двома світлодіодами одним виходом у МК передбачено активні вихідні каскади, і для перемикання режиму роботи (введення або виведення) слугує спеціальний регістр. Таким чином, сигнал кожного виходу будь-якого порту може мати 3 значення – "0", "1" і високоімпедансний ("Z") стан. Це дозволяє керувати двома світлодіодами за допомогою одного виводу (рис. 1.15).

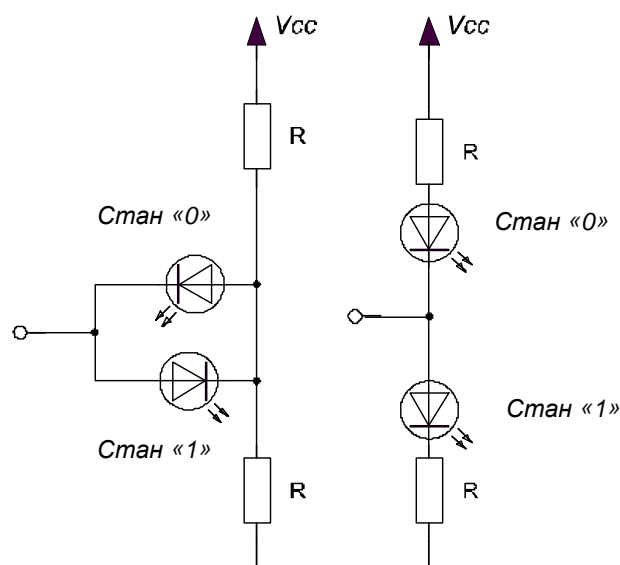


Рисунок 1.15 – Керування двома світлодіодами одним виходом МК

При роботі порту в режимі виходу, залежно від стану "0" або "1" горить відповідно верхній або нижній світлодіод. При перемиканні в Z-стан, і при відповідному виборі резисторів струм через світлодіоди дуже малий і їх світіння майже непомітно.

Цікавим є включення трьох пар зустрічно паралельних світлодіодів за схемою "зірка" (рис. 1.16).

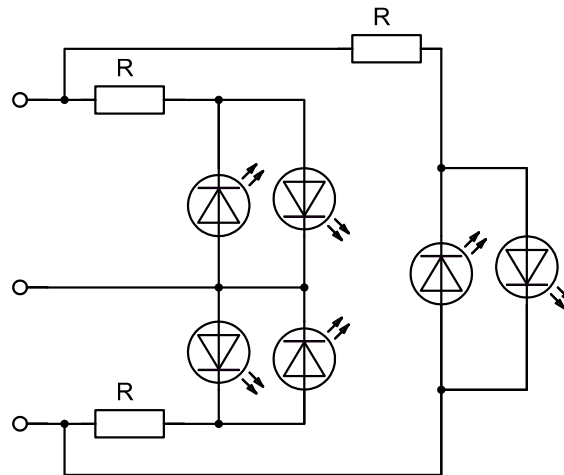


Рисунок 1.16 – Включення трьох пар зустрічно паралельних світлодіодів

Відповідними сигналами МК можна домогтись світіння будь-якого з шести світлодіодів, а також їх комбінацій. Використовуючи динамічну індикацію, можна отримати будь-які комбінації.

1.2.2 Організація динамічної індикації

Дуже часто МК використовується не тільки для керування роботою конструкції, але й для того, щоб повідомити що-небудь користувачеві і/або отримати від нього які-небудь вказівки про подальшу роботу. Наприклад, електронний годинник, крім власне відліків часу, повинен його ще відображати, а також дозволяти змінювати покази (встановлювати точний час). Якщо вся "інформація" зводиться до мигання парою світлодіодів, яких-небудь спеціальних зусиль з відображення інформації з боку розробника конструкції не вимагається, але якщо таких світлодіодів виявляється два-три десятки, тут вже потрібне застосування додаткових засобів (як апаратних, так і програмних). Як правило, в цьому випадку відображення інформації виконують у режимі динамічної індикації – це найбільш економний за кількістю використовуваних ліній спосіб.

Для організації динамічної індикації застосовується матриця, що складається з ліній рядків і ліній стовпців (рис. 1.17). На перетині

стовпця і рядка матриці розташований індикаторний елемент – світлодіод. Для того, щоб запалити той або інший елемент, необхідно подати на матрицю не один, як у звичайних індикаторах, а два сигнали: логічна 1 на відповідному рядку і логічний 0 на відповідному стовпці матриці. Через односторонню провідність світлодіода кожна комбінація сигналів на входах рядків і стовпців однозначно включає рівно один індикаторний елемент.

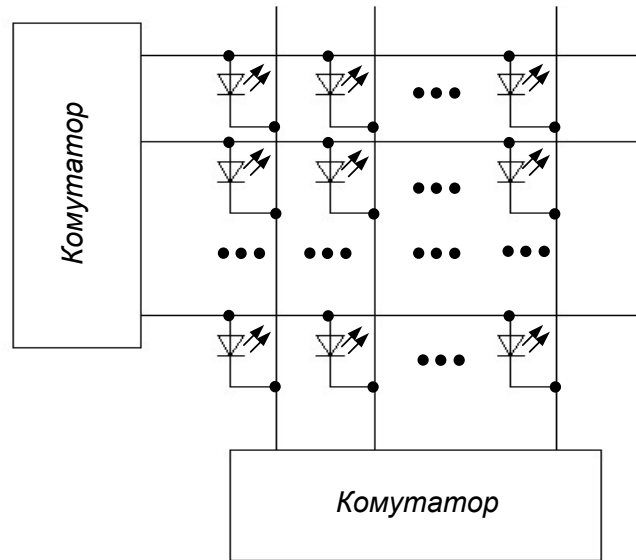


Рисунок 1.17 – Організація динамічної індикації

Головна перевага динамічної індикації – невелике число ліній, що керують: для матриці світлодіодів розміром $N \times N$ елементів потрібно всього $2N$ сигналів, що керують. За таку економію, втім, доводиться платити – справа в тому, що при почерговому виведенні інформації на кожен світлодіод матриці його яскравість світіння буде в N^2 разів нижча, ніж при безпосередньому виведенні інформації на один світлодіод, що "окремо стоїть". Тому в пристроях, що використовують динамічну індикацію, виведення інформації здійснюється не на кожен світлодіод окремо, а на один рядок або на один стовпець повністю – у цьому випадку яскравість світіння світлодіодів падає тільки в N разів.

Розглянемо декілька варіантів реалізації динамічної індикації із застосуванням МК [3, 7, 9, 26, 43, 56, 66]. Як індикаторний елемент передбачається застосування семисегментних індикаторів, але кожен такий індикатор з легкістю можна замінити і групою світлодіодів.

Схема реалізації динамічної індикація без додаткових елементів наведена на рисунку 1.18. До порту B МК підключені катоди всіх світлодіодів матриці, а до порту A – аноди кожного з індикаторів, що створюють матрицю. На лініях порту A організовується одиниця, що

"біжить". На лінії порту *B* при кожному положенні одиниці, що біжить, виводиться семисегментний код того символу, який повинен горіти в даному знакомісті. Для індикаторів із загальним катодом замість одиниці, що біжить, використовується нуль, що біжить. Перевага такого способу індикації – у відсутності яких-небудь додаткових компонентів (окрім самих світлодіодних індикаторів), головний недолік – значна перевитрата ліній портів. Таке рішення для МК може забезпечити роботу не більше 5 семисегментних індикаторів одночасно, і при цьому МК стає "глухим і сліпим", оскільки жодної вільної лінії у нього не залишається. При використанні інших МК з великою кількістю ніжок вказана проблема знімається.

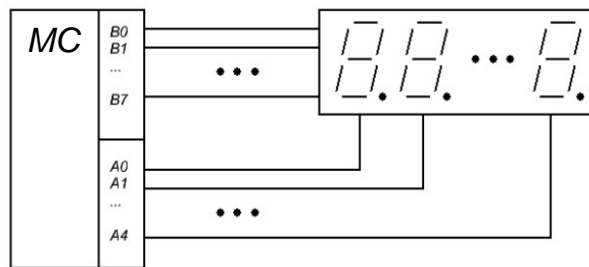


Рисунок 1.18 – Схема реалізації динамічної індикація без додаткових елементів

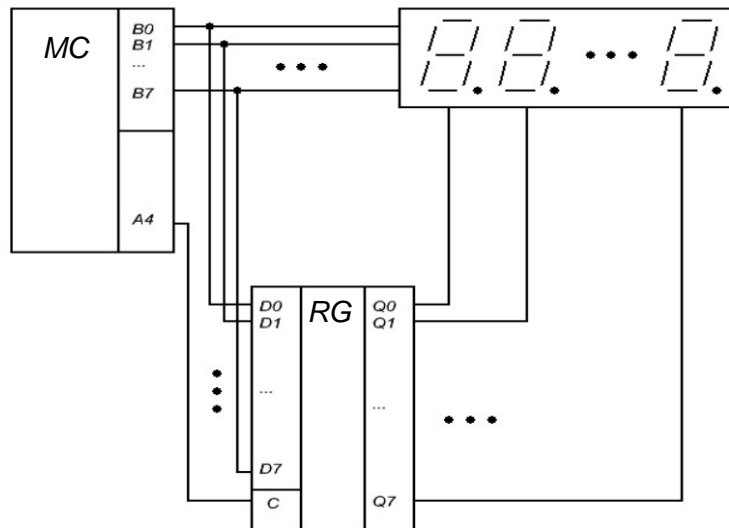


Рисунок 1.19 – Схема реалізації динамічної індикації з одним додатковим елементом

Схема реалізації динамічної індикації з одним додатковим елементом наведена на рисунку 1.19. До порту *B* МК підключені катооди всіх світлодіодів матриці, а також – входи буферного регістра. До виходів же буферного регістра підключені аноди кожного з

семисегментних індикаторів. На порт *B* спочатку виводиться одиниця, що біжить, яка записується в буферному регістрі сигналом *C*, що утворюється однією з ліній порту *A*. Потім на лінії порту *B* видається семисегментний код символу, який повинен горіти в знакомісті, визначуваному сигналом на виході буферного регістра. В даному випадку лінії порту *B* використовуються в режимі часового мультиплексування, тобто, ними по черзі передається і код символу, і номер знакомістя. Перевага такого способу індикації – менша витрата ліній портів МК (окрім порту *B* – всього одна додаткова лінія) і можливість роботи до 8 індикаторів одночасно. Ще один варіант індикації з одним додатковим елементом наведений на рисунку 1.20.

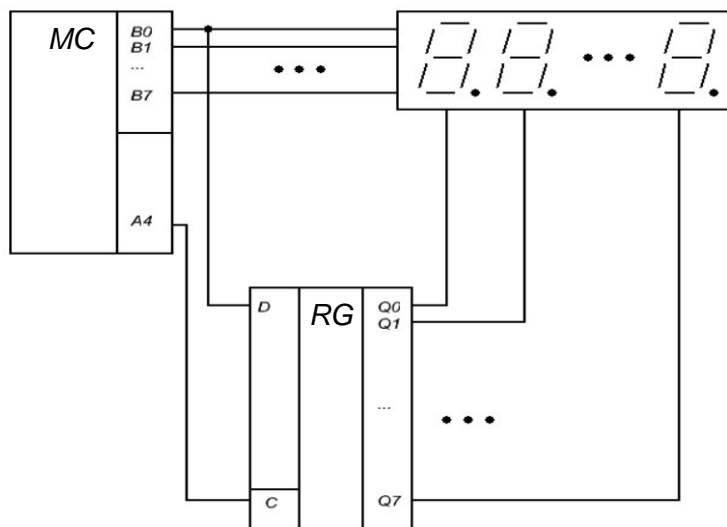


Рисунок 1.20 – Схема реалізації динамічної індикації з регістром зсуву

У цьому варіанті одиниця, що біжить, реалізується за допомогою регістра зсуву. Порт *B* у цій схемі також використовується в режимі часового мультиплексування – як для видачі символу, так і для занесення чергового біта до регістра зсуву. Схема також вимагає всього одну додаткову лінію (крім порту *B*), але за габаритами виходить трохи менше попереднього варіанта. Ще один варіант реалізації схеми з одним додатковим елементом наведений на рисунку 1.21. Така схема придатна тільки для індикаторів із загальним катодом, оскільки для організації нуля, що біжить, в ній використовується дешифратор. Схема вимагає три додаткові лінії (крім порту *B*), проте у багатьох випадках вона може виявитися цілком корисною.

Схема реалізації динамічної індикації з двома додатковими елементами наведена на рисунку 1.22. У схемі використовуються два послідовні регістри зсуву: один – для розгортки зображення по стовпцях (замінює порт *A*), інший – для розгортки зображення по

рядках (замінює порт *B*). Перевага такого рішення – всього три лінії порту МК. Крім того, при такому варіанті реалізації блок динамічної індикації легко оформити і у вигляді окремої плати. Недолік такого рішення – два додаткових компоненти.

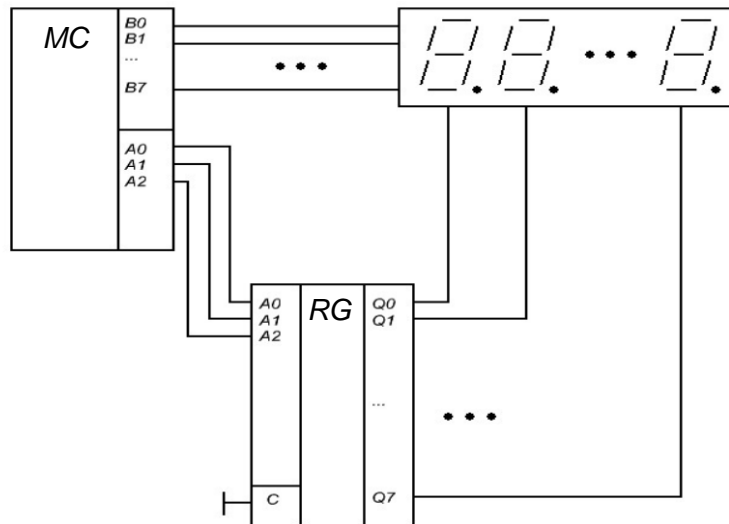


Рисунок 1.21 – Схема реалізації динамічної індикації з дешифратором

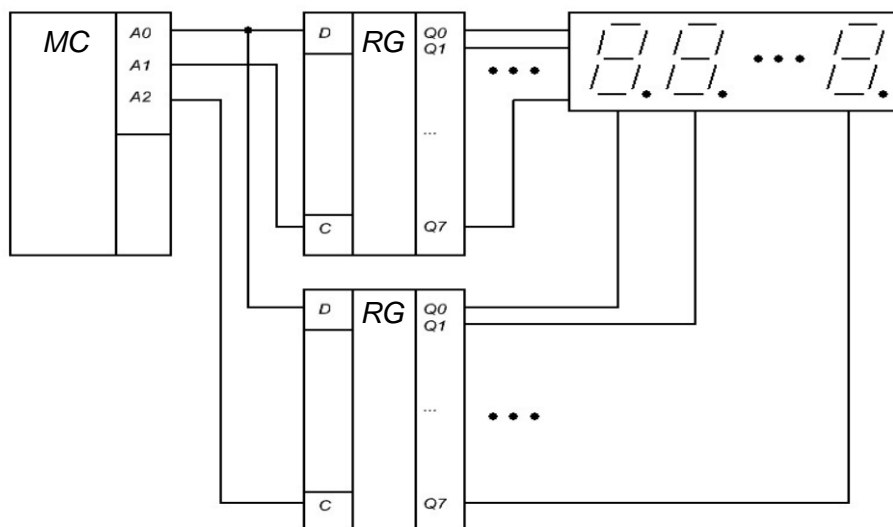


Рисунок 1.22 – Схема реалізації динамічної індикації з двома додатковими елементами

Ще одна схема реалізації наведена на рисунку 1.23. Перевага такої схеми – всього дві лінії порту. Недолік – складніша програма керування і велика тривалість формування вихідних сигналів, що викликає деяке паразитне підсвічування індикаторів (помітно лише в темряві).

Практична схема підключення динамічного індикатора на МК AT90S2313 за схемою, що наведена на рисунку 1.18, зображена на

рисунку 1.24. Символ, що буде світитись на індикаторі визначається сигналом логічної одиниці на вивід COM1, 2, 3, 4 індикатора. На шині даних (А, В, С ...) активним рівнем є рівень логічного 0.

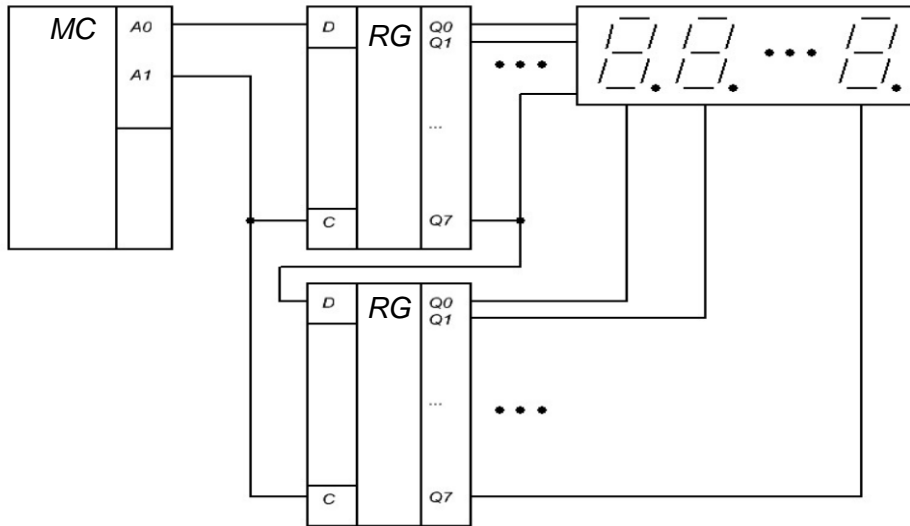


Рисунок 1.23 – Схема динамічної індикації з двома лініями керування

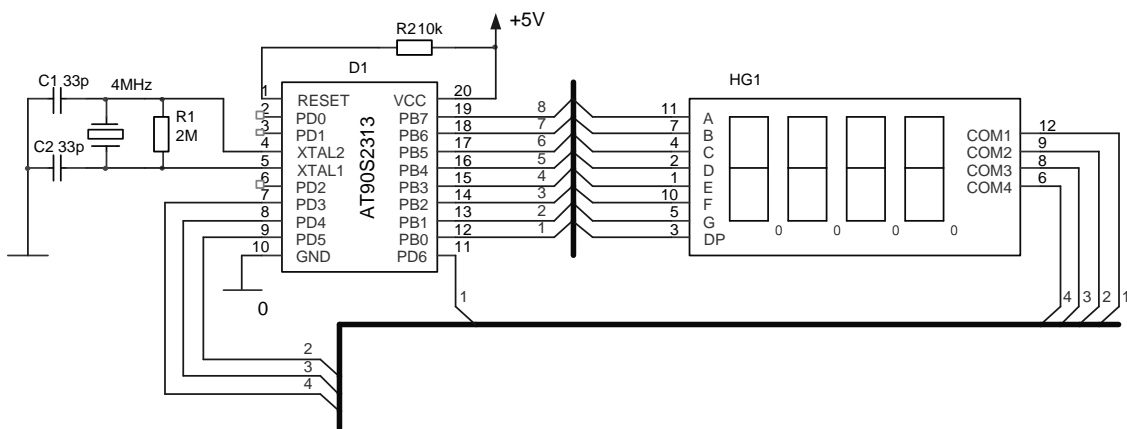


Рисунок 1.24 – Підключення динамічного індикатора до МК AT90S2313

1.3 Пристрої формування звукових сигналів

П'єзоелектричні динаміки служать для генерації звуків. Вони мають максимальну вхідну напругу 50 В і номінальний струм 10 мА. На рисунку 1.25 зображена схема, що використовує буфер КМОП/ТТЛ для керування таким динаміком. Схема пристрою керування на транзисторі ZTX300 наведена на рисунку 1.26 [1, 9]. Щоб отримати звук, необхідно подати на вхід послідовність імпульсів.

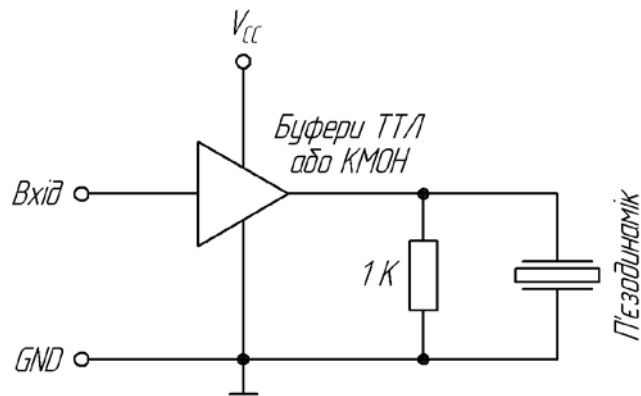


Рисунок 1.25 – Схема керування п'єзоелектричним динаміком на базі буфера TTL/КМОП

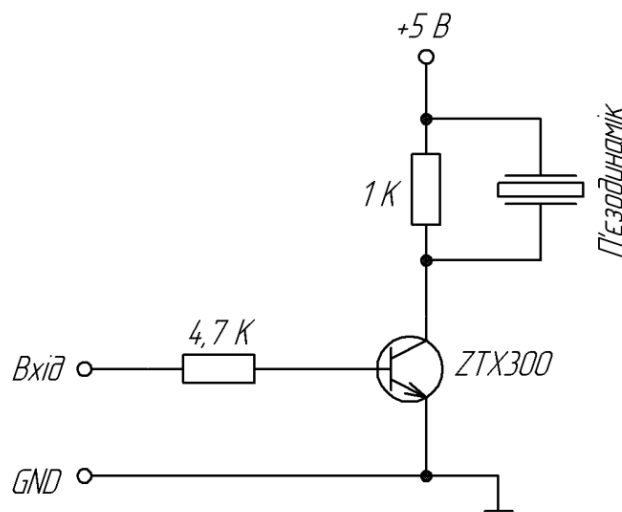


Рисунок 1.26 – Схема керування п'єзоелектричним динаміком на транзисторі

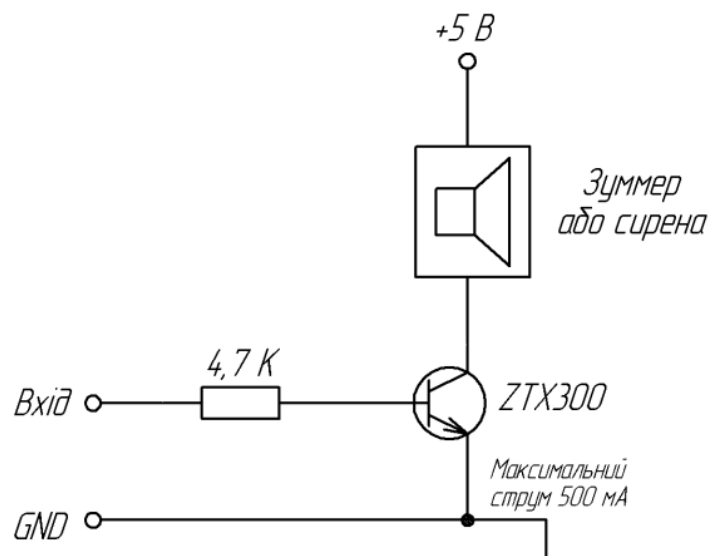


Рисунок 1.27 – Схема керування зуммером або сиреною

Напівпровідникові зумери – це автономні динаміки, здатні генерувати тон, частотою близько 450 Гц. На рисунку 1.27 наведена схема керування на транзисторі ZTX300. Для генерації звуку на базу ZTX300 необхідно подати високий рівень напруги. При керуванні сиренами можна використовувати такі ж схеми.

Ультразвукові перетворювачі призначені для генерації ультразвуку. Зазвичай вони застосовуються в пристроях дистанційного керування, вимірювання і передачі даних, наприклад в ультразвуковому вимірювачі відстаней і детекторах руху об'єкта. На рисунку 1.28 зображена схема, яка генерує сигнал частотою 38,4 кГц.

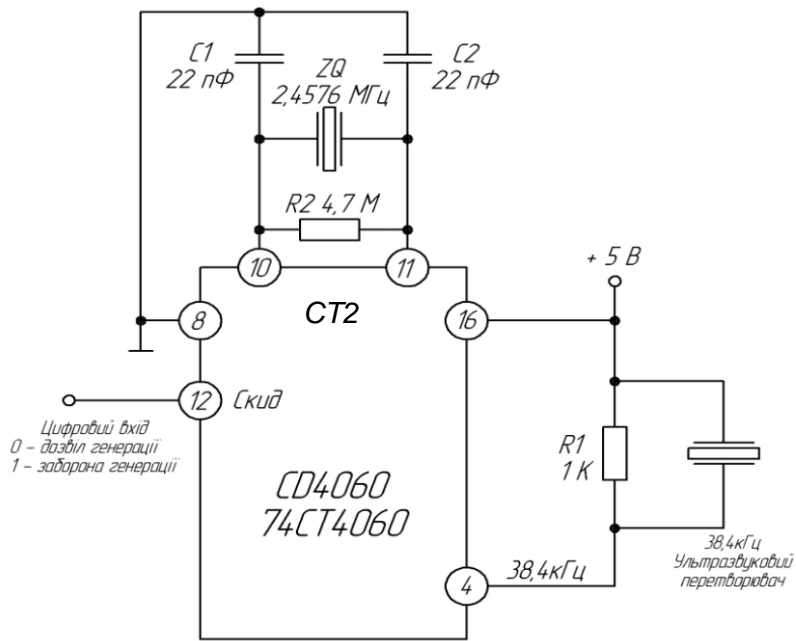


Рисунок 1.28 – Схема генератора ультразвуку

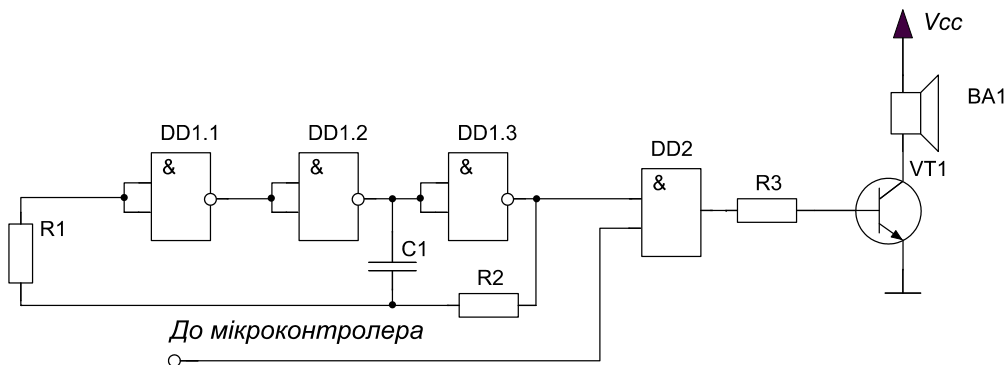


Рисунок 1.29 – Схема формування звукового сигналу апаратним способом

У даній схемі ультразвуковий сигнал генерується лише у тому

випадку, коли на контакт 12 (RESET) подається сигнал низького рівня. Якщо на цьому вході високий рівень, генерація заборонена [26, 64].

На рисунку 1.29 зображена схема формування звукового сигналу апаратним способом. Схема складається з генератора звукового діапазону на логічних елементах І та схеми керування звуковим пристроєм. При формуванні МК сигналу логічної "1" імпульси від генератора проходять на звуковий пристрій.

1.4 Пристрої керування двигунами постійного струму

Двигунами постійного струму можна керувати за допомогою реле або транзисторів (рисунок 1.30). Одиночне перемикальне реле вмикає і вимикає двигун, а спарене реле відповідає за напрям обертання (рисунок 1.31).

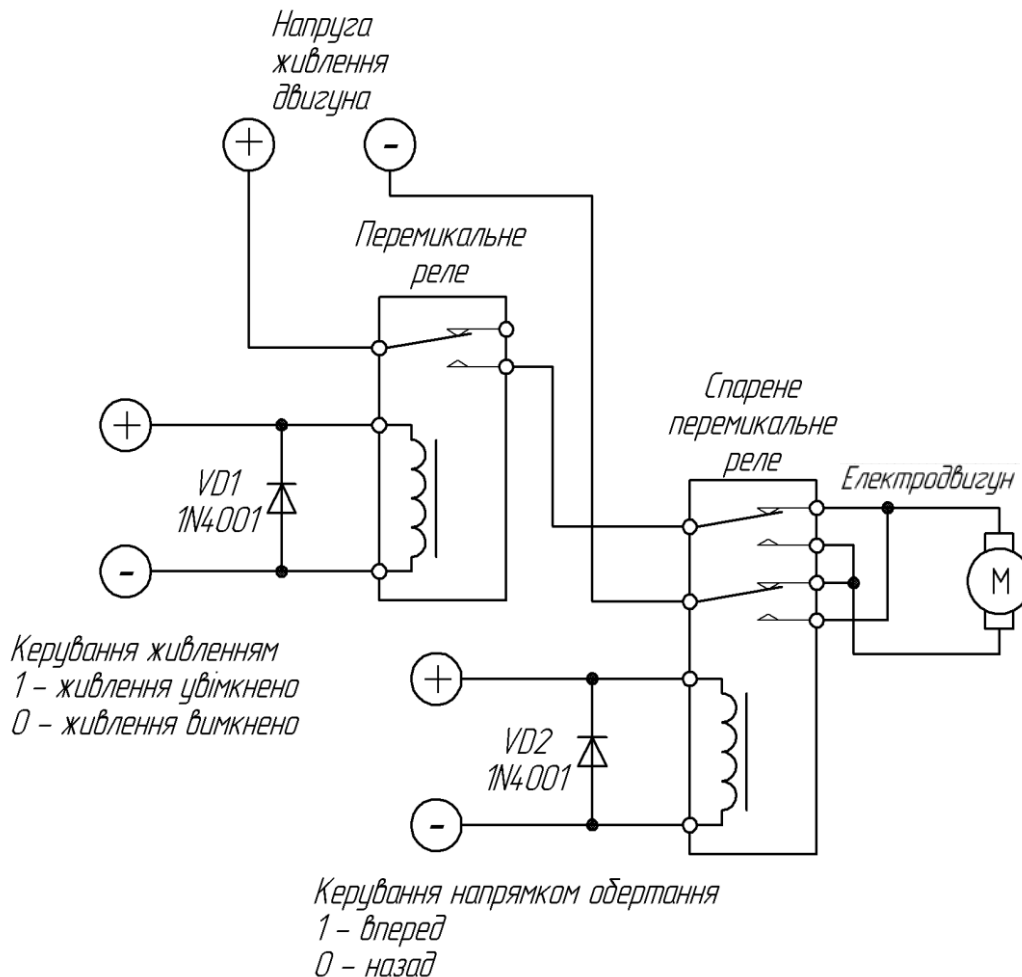


Рисунок 1.30 – Пристрій керування двигуном постійного струму на базі реле

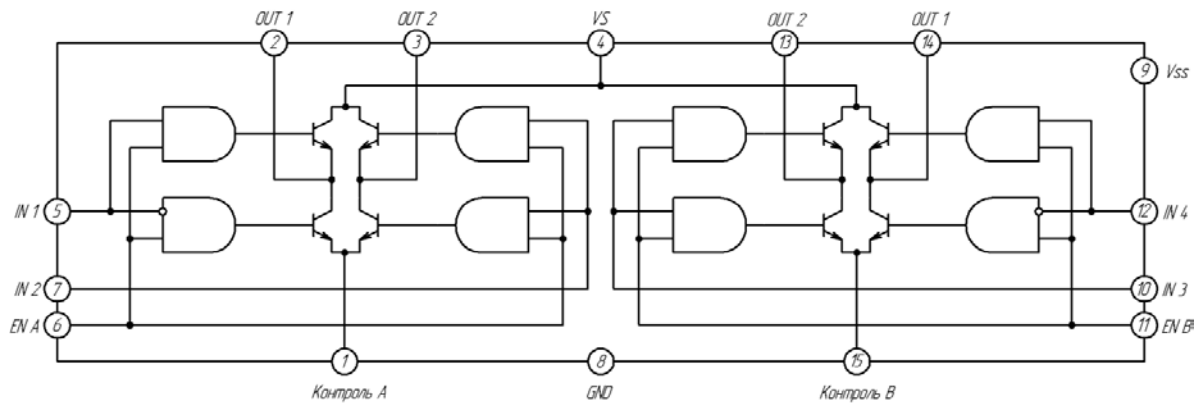


Рисунок 1.31 – Блок-схема мостового пристрою керування L298N

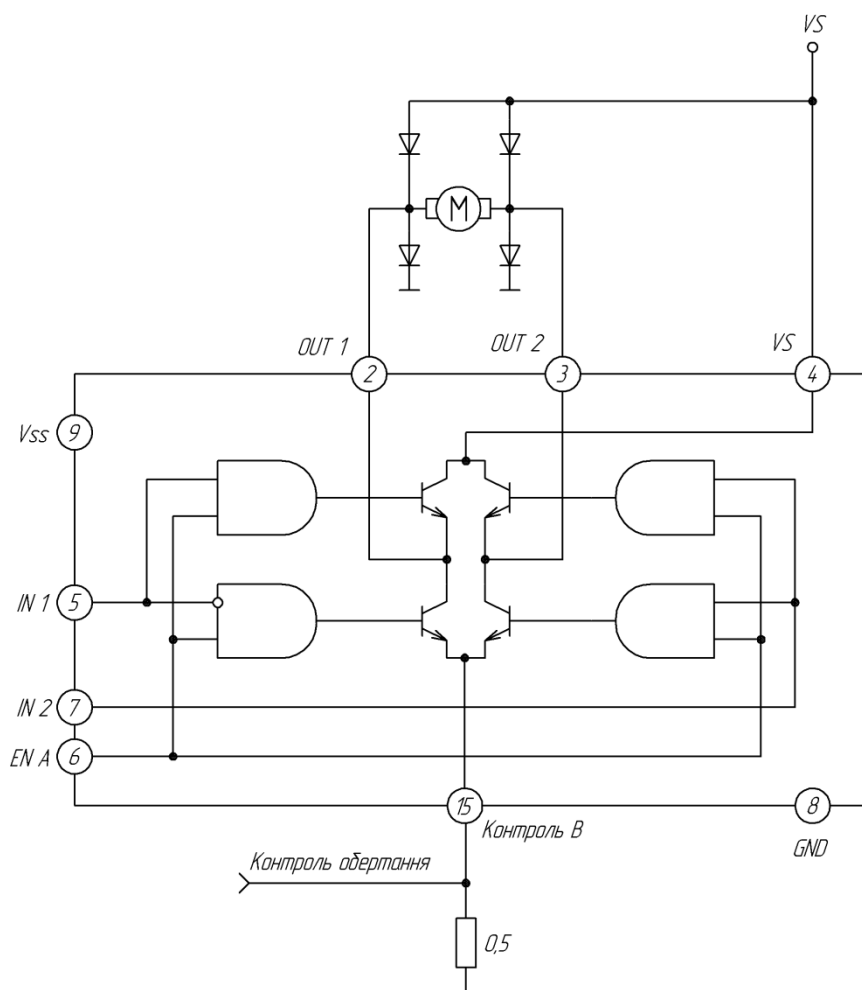


Рисунок 1.32 – Пристрій керування двигуном постійного струму

Інший спосіб керування двигунами постійного струму заснований на використанні мостових схем типу L298N (SGS-Thomson, RS636-384) [1]. Це двоканальний пристрій, в якому присутня потужна напруга до 46 В, струм до 2А на кожен канал. Такий пристрій працює від рівнів ТТЛ (див. рис. 1.31). З виводу VS

(контакт 4) подається напруга живлення для двигуна, на вивід V_{ss} (контакт 9) подається напруга живлення схеми (+5В). Виводи $EN A$ і $EN B$ (контакти 6 і 11) відкривають входи двох каналів. Входи $IN 1$ і $IN 2$ (контакти 5 і 7) керують першим каналом, а $IN 3$ і $IN 4$ – другим. Емітери транзисторів з'єднані для підключення зовнішніх датчиків контролю. Типова схема включення для одного каналу наведена на рисунку 1.32.

Коли на вході $EN A$ низький рівень, входи заблоковані і двигун не обертається. Якщо на цей вхід подати високий рівень, входи відкриваються. Входи $IN 1$ і $IN 2$ керують режимами роботи двигуна таким чином:

- $IN 1$ - "1", $IN 2$ - "0" – двигун обертається за годинниковою стрілкою;
- $IN 1$ - "0", $IN 2$ - "1" – двигун обертається проти годинникової стрілки;
- $IN 1 = IN 2$ – двигун не обертається.

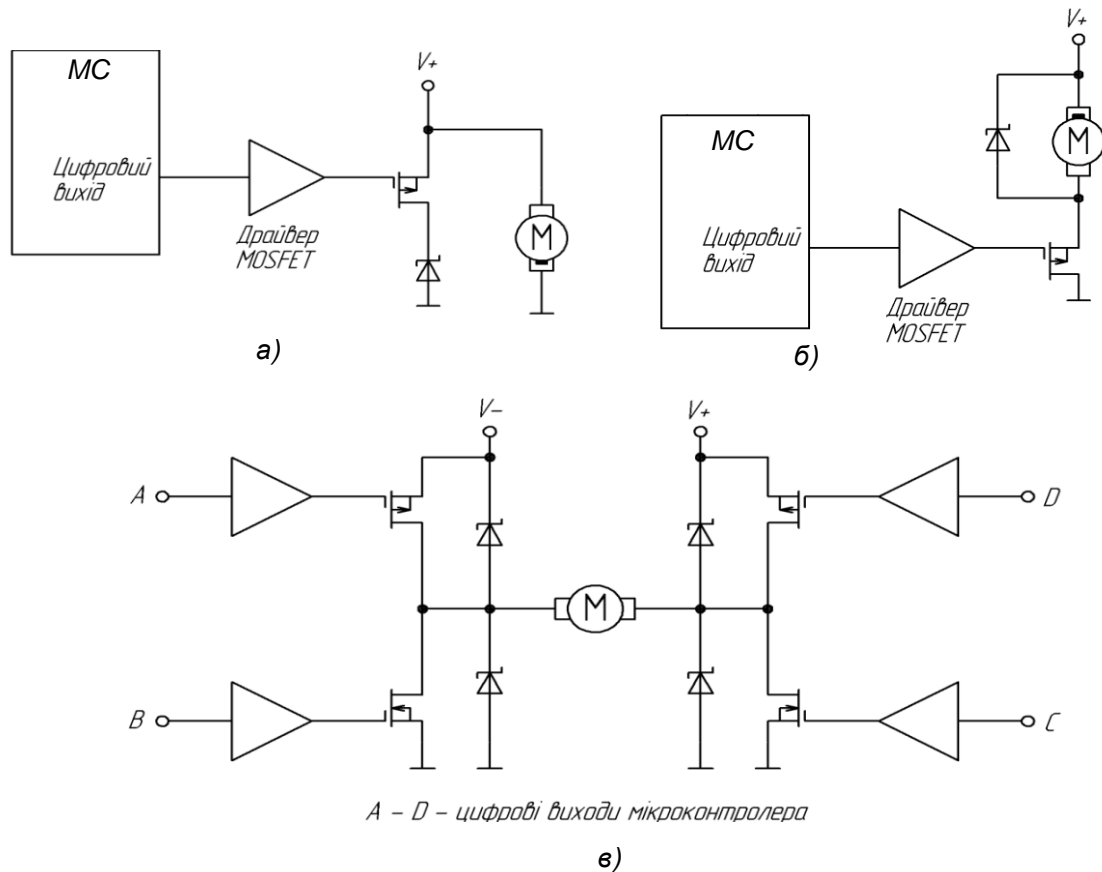


Рисунок 1.33 – Керування колекторним двигуном постійного струму: а) схема з верхнім розташуванням ключа; б) схема з нижнім розташуванням ключа; в) мостова схема

При керуванні колекторними двигунами постійного струму

потрібно регулювати струм, що проходить через обмотки двигуна. Цей процес охоплює регулювання напрямку магнітного потоку і величини струму. Проста схема керування наведена на рисунку 1.33. Дані схеми дозволяють керувати обертанням двигуна лише в одному напрямку.

Схема з верхнім розташуванням ключа часто застосовується в системах з підвищеними вимогами до безпеки – коротке замикання не призводить до увімкнення двигуна, у вимкненому стані обидва виводи обмотки підключено до спільної точки схеми. Схема з нижнім розташуванням ключа найдешевша, оскільки для керування силовим транзистором *MOSFET* досить подавати на затвор сигнал з цифрового виходу МК без використання спеціального драйвера. Для реверсивного керування двигуном потрібно використовувати мостову схему включення, наведену на рисунку 1.33. Частота обертання двигуна регулюється за допомогою зміни значення напруги на обмотці якоря. При використанні МК цю напругу можна регулювати за допомогою широтно-імпульсної модуляції.

Для вимірювання частоти обертання двигуна можна використовувати ефект зворотної ЕРС або використовувати оптоелектронний датчик положення ротора ("ромашка") (рис. 1.34).

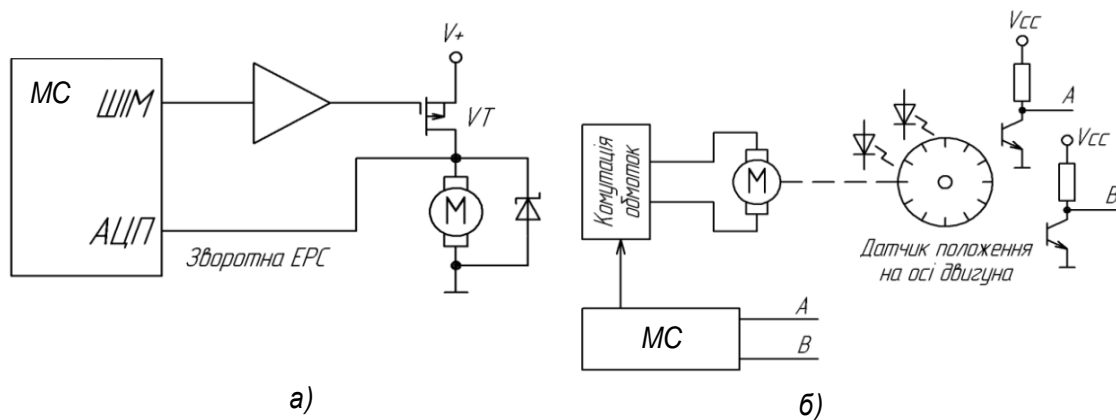


Рисунок 1.34 – Вимірювання швидкості обертання двигуна: а) з використанням зворотної ЕРС; б) з використанням датчика положення ротора

Безколекторні двигуни є прикладом спрощення конструкції з одночасним ускладненням схеми керування. Двигун не може самостійно перемикає обмотки (керувати струмом), тому схема керування повинна самостійно коректно регулювати величину струму в обмотках для забезпечення рівномірного обертання вала двигуна. Схема керування містить півмостову схему включення кожного з трьох виводів обмоток. Існують 2 основних типи керування

безколекторним двигуном: з датчиками і без датчиків. Для того, щоб включати обмотки в потрібній послідовності, необхідно використовувати різні методи визначення положення ротора. Двигун з датчиком завжди повідомляє МК про положення ротора. Кожному положенню ротора відповідає певний набір керівних дій, які подаються на мостову схему включення обмоток.

У двигунах без датчика положення ротора визначається за величиною ЕРС, що виникає в невідключеній обмотці. Двигуни без датчиків простіші у виготовленні, але складніші в управлінні. Їх застосовують в умовах, що не вимагають частих запусків і зупинок. Двигуни з датчиками – кращий вибір для умов, пов'язаних з періодичними зупинками і запусками. Схеми включення двигуна наведені на рисунку 1.35. Складність побудови схем керування не залежить від типу двигуна. Безколекторні двигуни мають кращі показники надійності, питомої потужності і економічності в порівнянні з колекторними, тому рекомендується використовувати безколекторні двигуни.

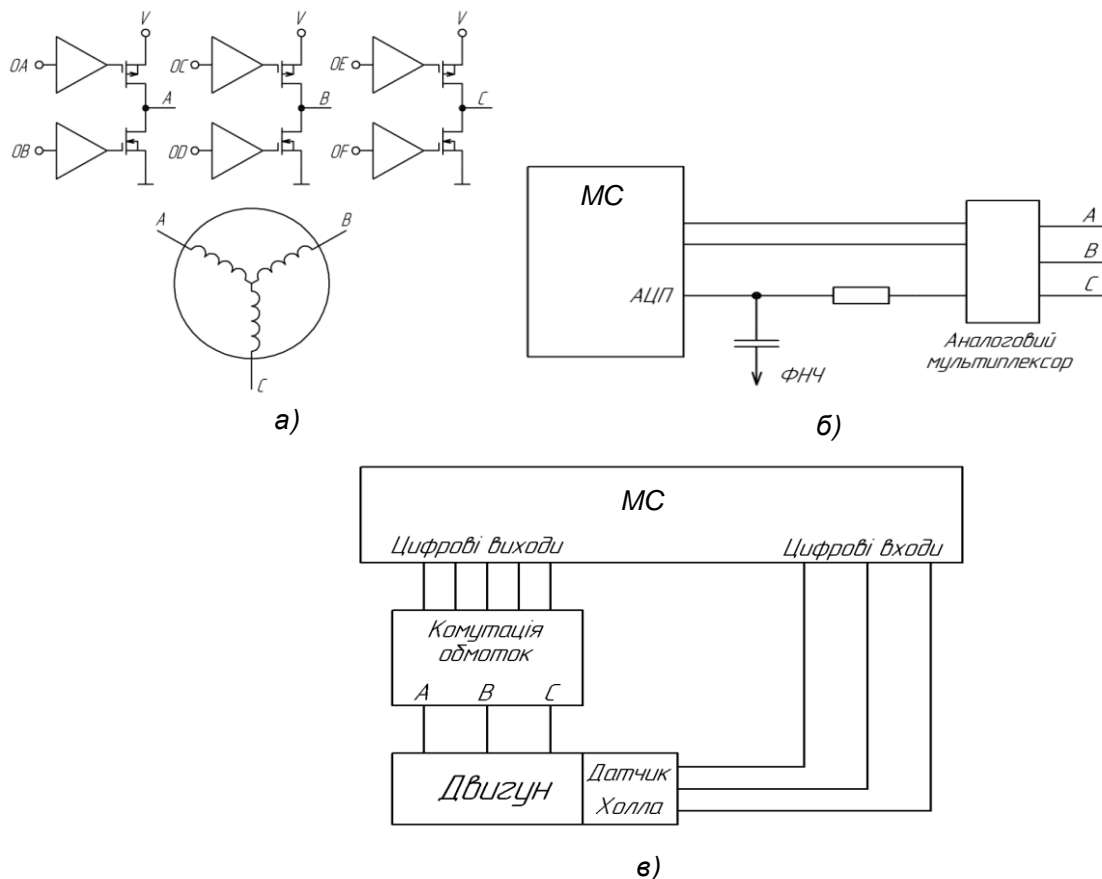


Рисунок 1.35 – Керування безколекторним двигуном постійного струму: а) спільна схема включення; б) схема включення без датчика положення; в) схема включення з датчиком положення.

1.5 Пристрої комутації

1.5.1 Аналогові ключі

Аналоговий електронний ключ можна уявити як електронний аналог реле, що дозволяє МК замикати і розмикати контакт між двома точками схеми. Аналогові ключі швидші і компактніші ніж реле, не мають тремтіння контактів і вимагають менший струм для включення.

На рисунку 1.36, а показано схематичне зображення аналогового ключа і його внутрішню структуру. Зазвичай ключ складається з МДН-транзисторів n - і p -типу, з'єднаних паралельно. Коло керування вмикає або вимикає відразу обидва транзистори. Коли транзистори відкриті, струм може протікати ними в обох напрямках однаково – як від входу аналогового ключа до виходу, так і від виходу до входу. Напрямок струму – від входу до виходу або навпаки – визначається потенціалом керування на вході.

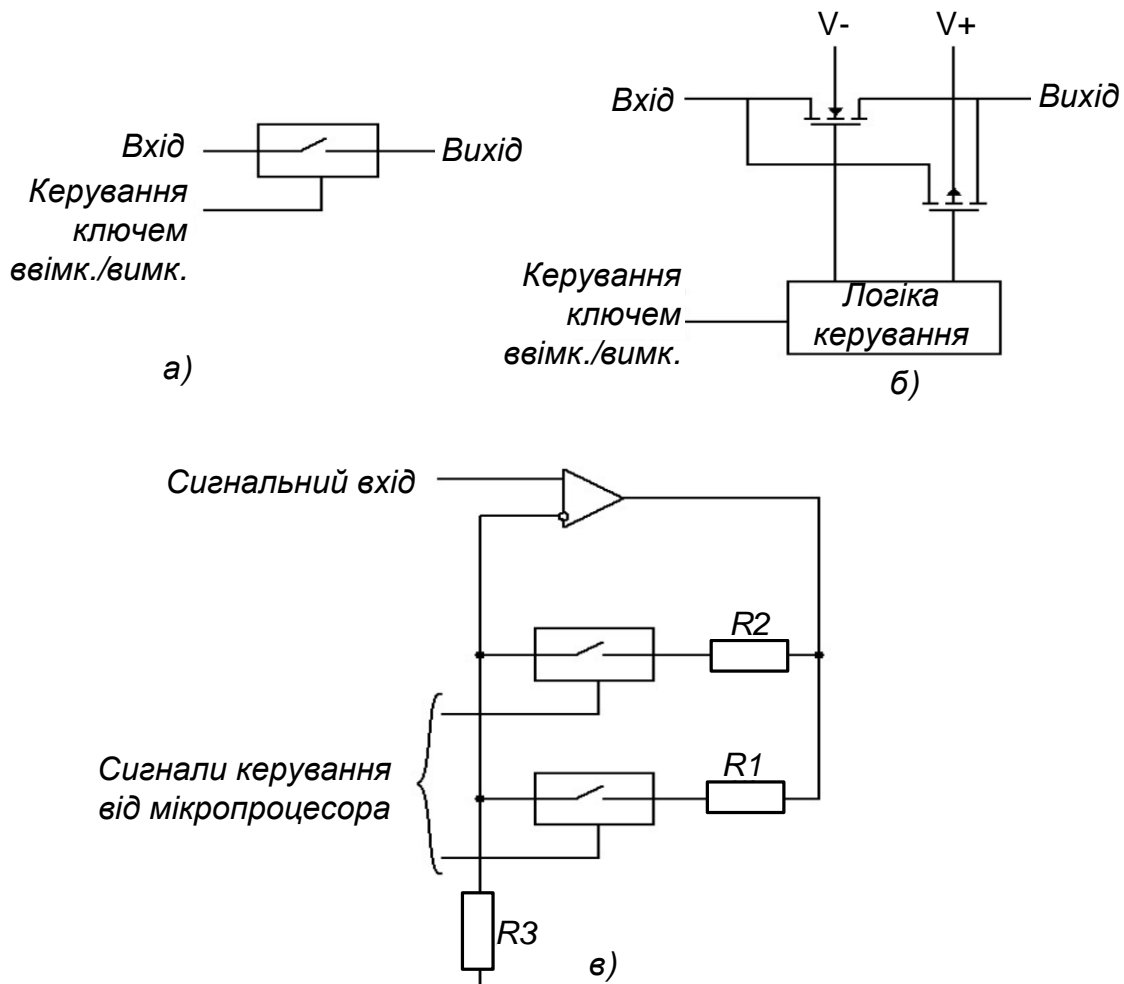


Рисунок 1.36 – Аналоговий ключ: керування та застосування

На рисунку 1.36, б показаний аналоговий ключ з двополярним

живленням $V+$ і $V-$. Насправді, деякі ключі можуть працювати і з однополярним живленням ($V-$ підключений до землі). У деяких ключах потрібний третій вхід напруги (див. рис. 1.36, в), еквівалентний напрузі живлення логічних кіл.

1.5.2 Типи аналогових ключів

Хоча аналоговий ключ і можна уявити аналогом твердотільного реле, між ними існують значні відмінності. Наприклад, контакти реле абсолютно ізольовані від обмотки реле. Це дозволяє перемикає напругу в сотні вольт без небезпеки пошкодити МП кола, тобто реле працює як гальванічна розв'язка, аналогічно оптрону з закритим оптичним каналом. У аналоговому ключі необхідно подавати живлення на перемикальні транзистори, тому амплітуда напруги сигналу не може вийти за межі $V+$ і $V-$. Якщо напруга вхідного або вихідного сигналу вийде за ці межі, ключ може вийти з ладу через значний струм між сигнальним провідником і шиною живлення.

Деякі аналогові ключі забезпечені захистом від перенапруги. Проте вони не можуть перемикає напруги вище напруги живлення, але і не виходять з ладу. Ця особливість передбачена для пристроїв, сигнал на які може надходити навіть тоді, коли напруга живлення відключається. Типовим прикладом є ІС MAX4511/4512 фірми Maxim/Dallas [1, 11].

Оскільки аналогові ключі побудовані з використанням МДН-транзисторів, існує деякий опір контакту в замкнутому стані, рівний опорі стік-витік транзистора. У перших модифікаціях ключів цей опір вимірювався сотнями Ом, проте в останніх розробках понижено до часток Ома.

Час вмикання і вимикання аналогових ключів позначається як T_{on} і T_{off} і змінюється від наносекунд до мікросекунд.

1.5.3 Застосування аналогових ключів

Аналогові ключі можуть бути використані для регулювання коефіцієнта підсилення операційного підсилювача (ОП), як показано на рисунку 1.36, в. Два резистори $R1$ і $R2$ перемикаються аналоговим ключем під дією зовнішнього МК. У даному колі встановленням сигналів керування МК (код 01, 10 і 11) можна вибрати один з трьох коефіцієнтів підсилення, пропорційних величинам $R1$, $R2$ і $R1$ паралельно $R2$. Відмітимо, що напруга живлення аналогових ключів має бути вище за амплітуду вхідних і вихідних сигналів, тобто, у загальному випадку, напруги живлення ОП. Насправді, якщо напруга

живлення ОП стабілізована, то напруга живлення аналогового ключа має бути лише трохи більша максимальної напруги на неінвертувальному вході ОП.

1.5.4 Аналогові мультиплектори

На рисунку 1.37 зображений 4-1 (4 входи/1 вихід) аналоговий мультиплектор. Аналоговий мультиплектор складається з двох або більше аналогових ключів з одним спільним виходом (позначеним Y на схемі). За допомогою входів керування можна вибрати, який з контактів буде замкнений, і, таким чином, який з входів $X1...X4$ буде з'єднаний з виходом Y .

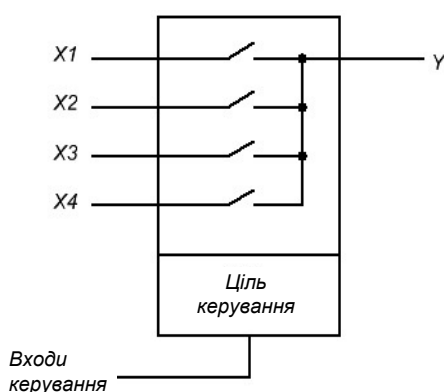


Рисунок 1.37 – Аналоговий мультиплектор

Як і аналоговий ключ, мультиплектор характеризується такими параметрами, як діапазон вхідної напруги, опір у відкритому і закритому станах, час перемикачання. Типове використання аналогових мультиплекторів – вибір джерела звукового сигналу в аудіопристроях.

Інтегральна схема MAX349 фірми Maxim/Dallas - це один 8-1 аналоговий мультиплектор з інтерфейсом SPI. Інтегральна схема (IC) MAX350 складається з двох 4-1 аналогових мультиплекторів, також з керуванням за допомогою інтерфейсу SPI [1, 9, 11, 12].

1.6 Кнопки та датчики

Практично жоден мікропроцесорний пристрій не обходиться без кнопок і простих датчиків. За допомогою цих периферійних елементів в МПП надходить різна інформація, яка використовується для зміни алгоритму роботи програми. Схема підключення контактного датчика до МК наведена на рисунку 1.38. У наведеному прикладі датчик підключений до лінії PDO порту D МК. Через цей вхід МК проводить зчитування стану датчика. Датчик можна підключити і до будь-якої іншої лінії будь-якого з портів МК.

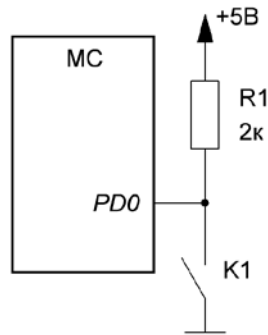


Рисунок 1.38 – Підключення контактного датчика до МК

У початковому стані контакти датчика розімкнені. На вхід МК через резистор R1 прикладається напруга від джерела живлення + 5 В. МК сприймає цю напругу як сигнал логічної одиниці. При спрацьовуванні датчика контакти замикаються і з'єднують вивід МК із загальним дротом. Тепер мікросхема сприймає вхідний рівень сигналу як логічний нуль. Резистор R1 при цьому служить струмообмежувальним елементом, запобігаючи короткому замиканню між шиною живлення і загальним дротом. Деякі МК мають свої внутрішні резистори навантаження, які можуть замінити зовнішній резистор. Схема підключення декількох датчиків або кнопок до МК зображена на рисунку 1.39.

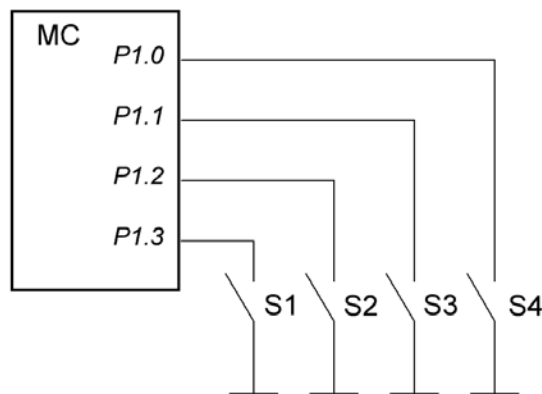


Рисунок 1.39 – Підключення кнопок або простих датчиків до МК

У схемі, що зображена рисунку 1.40, при натисканні однієї з клавіш змінюється постійна напруга на відповідному вході процесора, яка розпізнається процесором і дешифрується в певну команду. Ця напруга максимальна (приблизно 5 В), коли кнопки не натиснуті, і мінімальна (0 В) при натиснутій клавіші S1 [6, 9, 56, 65, 66, 68].

Існує два види клавіатур, що підключаються до МК: зі скануванням і з кодуванням.

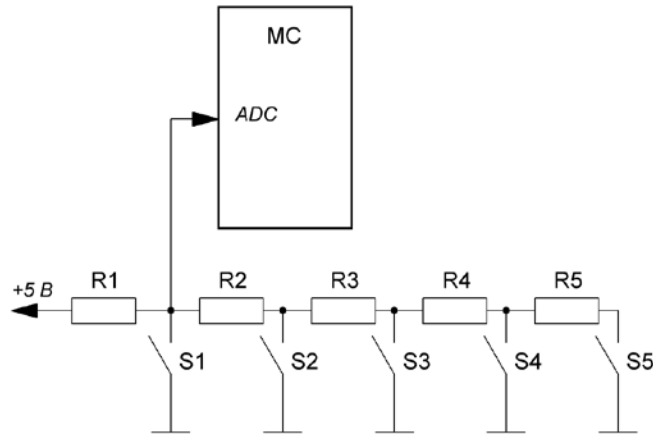


Рисунок 1.40 – Підключення кнопок зміною напруги на аналоговому вході МК

Блок-схема 12-клавiшної клавіатури зі скануванням показана на рисунку 1.41.

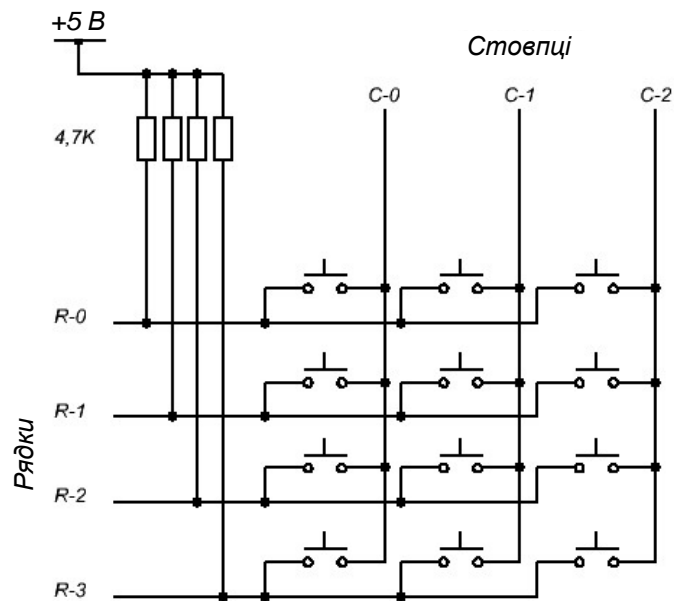


Рисунок 1.41 – Матрична клавіатура 4×3

Клавiші розташовані у вузлах матриці, у якої чотири лінії рядків і три лінії стовпців. На лінії стовпців по черзі подається негативний імпульс (логічний "0"). У цей момент перевіряється стан чотирьох ліній рядків. Якщо натиснутих клавiш немає, всі лінії рядків мають високий рівень (вони підключені до напруги +5 В через резистори). Якщо ж клавiша натискається, і на лінії стовпця, відповідного натиснутій клавiші, все ще нуль, то адекватна лінія рядка також стає рівною нулю. Знаючи номери стовпця і рядка, можна отримати позицію натиснутої клавiші.

У клавіатурі з кодуванням застосовують спеціалізовані мікросхеми, які виявляють натискання клавіші і передають її код. Прикладом такого пристрою є мікросхема MM74C922 (National Semiconductors) [12, 14, 57].

Розглянемо організацію клавіатури 5×3, що зображена на рисунку 1.42. Горизонтальні лінії X0...X4 (ряди або рядки) є вихідними, а вертикальні Y0...Y2 (колонки) є входними. У неактивному перебуванні на виходах X0... X4 присутній рівень логічної 1, а входи Y0...Y2 зазвичай підтягуються до живлення резисторами. Для цієї мети, як правило, використовуються вбудовані резистори (pullup), які є у деяких портів МК. При скануванні на горизонтальних лініях матриці формується сигнал нуля, що "біжить". На решті виходів зберігається рівень логічної 1. Для кожної фази нуля, що "біжить", проводиться зчитування і аналіз коду Y0...Y2. При замиканні якої-небудь з клавіш матриці програмно встановлюється код даної клавіші. Окрім розпізнавання положення натиснутої клавіші слід програмно захиститися від "тремтіння" контактів, тобто від впливу перехідних процесів, а також від ситуацій, пов'язаних з одночасним натисканням декількох клавіш.

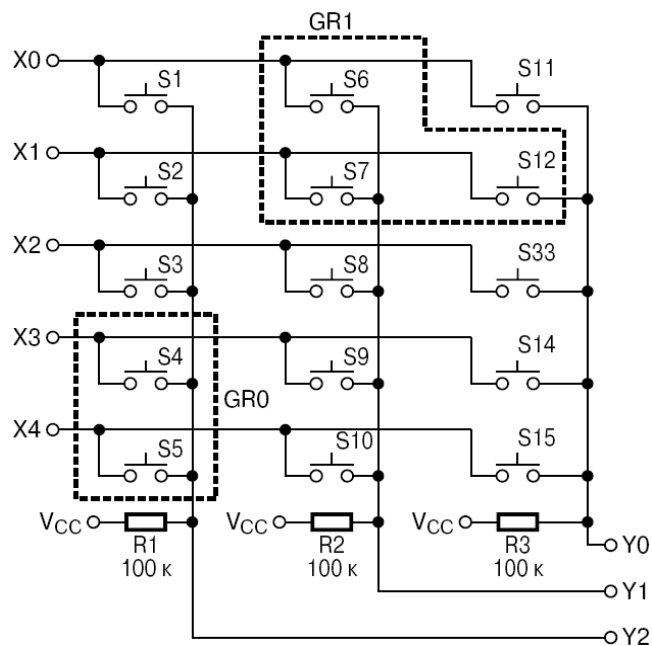


Рисунок 1.42 – Структура матричної клавіатури

Для того, щоб запобігти протіканню небезпечних струмів при одночасному замиканні декількох клавіш у одному стовпці, в лініях X0...X4 зазвичай встановлюють послідовно резистори або діоди. З цією ж метою можна використовувати і інший метод сканування, при якому всі неактивні горизонтальні шини, окрім шини нуля, що

"біжить", програмно призначаються входами. Впливу перехідних процесів можна уникнути, якщо повторно зчитувати стан входів матриці сканування через певну часову затримку.

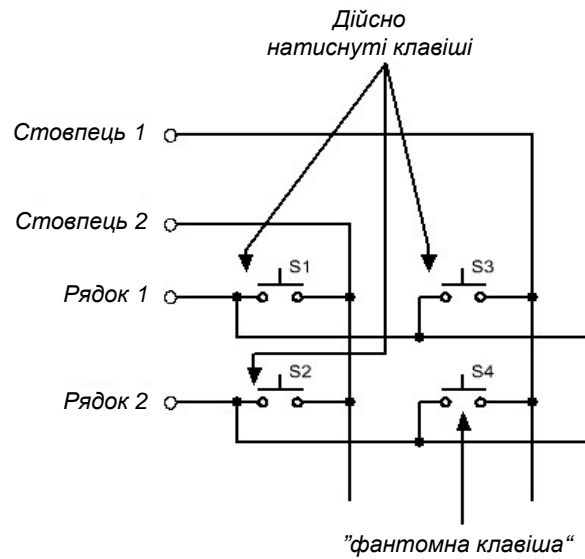


Рисунок 1.43 – Помилкова клавіша при скануванні

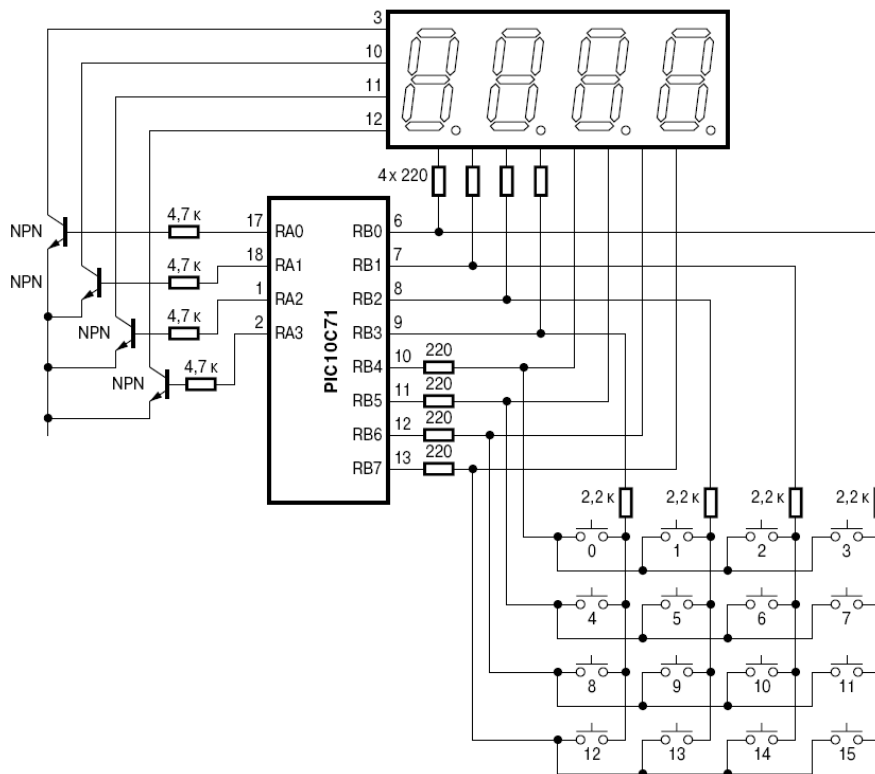


Рисунок 1.44 – Мультиплексорне керування матричною клавіатурою та семисегментним світлодіодним табло

У невеликих клавіатурних матрицях комбінації декількох натиснутих клавіш не використовуються. Але якщо це все-таки буде потрібно, то для сканування краще використовувати метод з

призначенням входами неактивних рядків. У цьому випадку можна впевнено визначати дві одночасно натиснуті клавіші в одному рядку і навіть визначати позиції трьох одночасно натиснутих клавіш. Але це справедливо, тільки якщо топологія трьох натиснутих кнопок не утворює прямого кута, як показано на рисунку 1.43.

Для раціонального використання ресурсу портів їх виводи, які використовуються для сканування клавіатури, можна використовувати і для інших функцій, наприклад, для підтримки динамічної світлодіодної індикації. На рисунку 1.44 показаний приклад такого мультиплексування портів МК.

Схема матричної клавіатури на 32 клавіші наведена на рисунку 1.45.

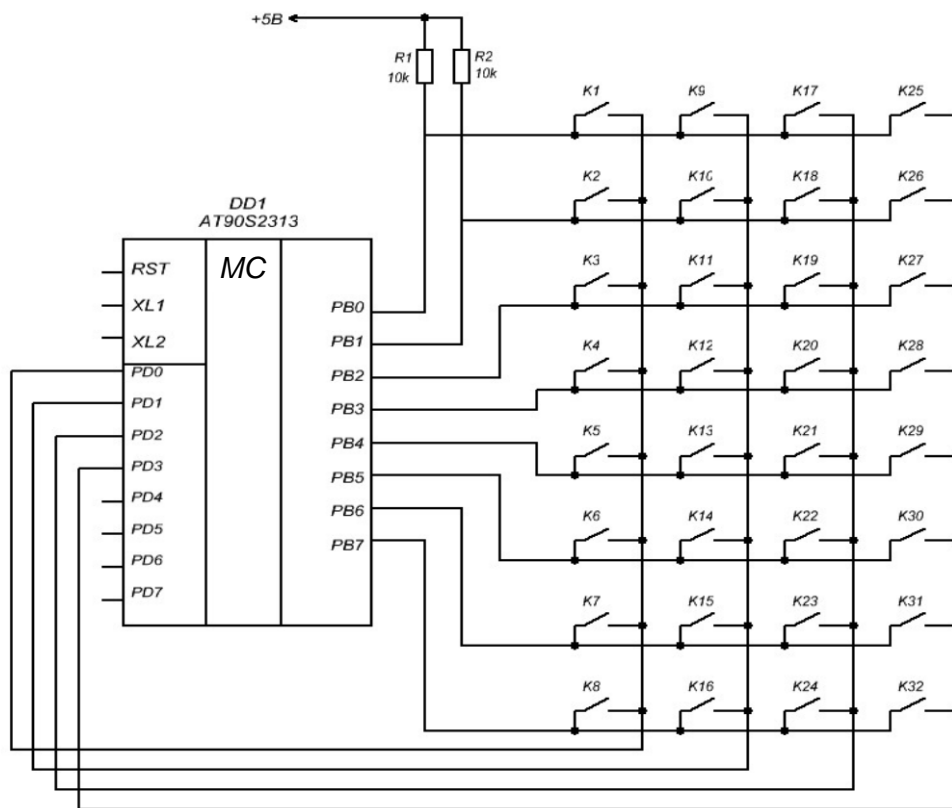


Рисунок 1.45 – Схема підключення клавіатури у вигляді матриці клавіш

Додавши один дешифратор, можна заощадити дві лінії порту PD. Схема клавіатури з дешифратором наведена на рисунку 1.46.

У цій схемі для вибору одного з чотирьох стовпців клавіатури використовується дешифратор DD2 типу K555ИД4. У такій схемі для сканування стовпців МК повинен подавати на виходи PD0 і PD1 двійковий код номера стовпця. Код надходить на входи A0 і A1 дешифратора. В результаті один з його виходів (той, номер якого

відповідає коду, що надійшов) прийме нульове значення. На решті виходів буде одиниця. Так, при коді 00 на вході дешифратора вихід $Q0$ (вивід 9) приймає нульове значення. Для коду 01 – нуль буде на виході $Q1$ і так далі. Таким чином, МК може перебирати всі чотири стовпці, використовуючи всього два розряди.

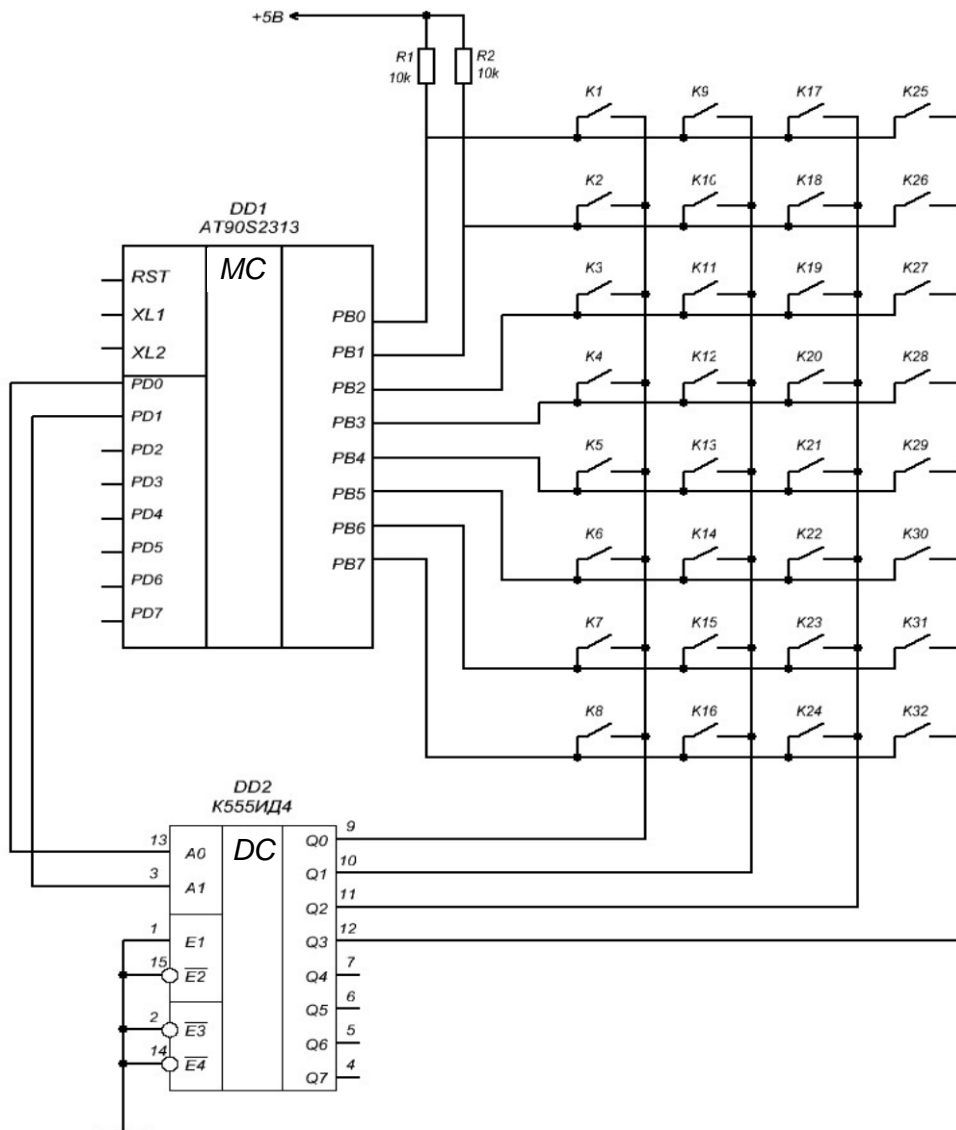


Рисунок 1.46 – Підключення клавіатури з використанням дешифратора

Існують й інші способи зменшення числа використання виводів МК, призначених для сканування клавіатури. Реалізація одного з таких способів наведена на рисунку 1.47. У звичайних матричних клавіатурах при замиканні кожного контакту утворюється електричне з'єднання між вихідною і вхідною ланками схеми сканування. Але якщо змінити топологію розташування клавіш так, щоб при замиканні

контакту забезпечувалася зміна потенціалів групи шин, то число кодованих клавіш можна значно збільшити. Така конструкція клавіатури забезпечує кодування більшого числа клавіш, ніж звичайна матрична, при однаковому числі шин, що використовуються для кодування. Збільшення досягнуте за рахунок того, що додаткові кодові комбінації утворюються парами шин кодування.

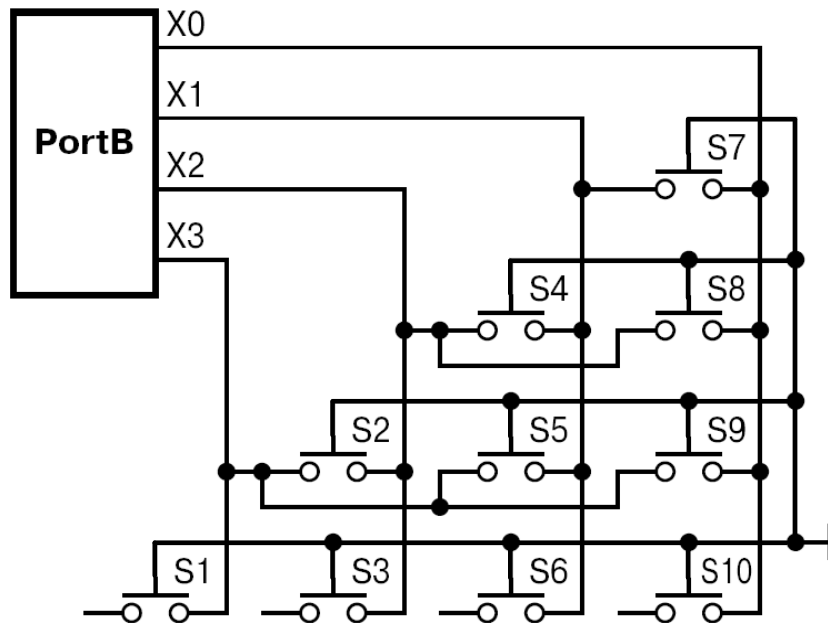


Рисунок 1.47 – Топологія клавіатури з одним контактором

Для подібного варіанта топології сканування не потрібне. Всі шини є входами з підтягуванням до напруги живлення резисторами. Контакттор з'єднаний з загальною шиною. При замиканні контакту потенціал контактора передається на одну або дві кодові шини, на перетині яких він знаходиться. Якщо ввести другий замикальний контакттор з потенціалом живлення, то можна вдвічі збільшити число сканованих клавіш. Для цього варіанту необхідно проводити сканування, наприклад, з періодом 20-30 мс. При скануванні використовується ефект пам'яті на паразитній ємкості шин. У першій фазі сканування всі шини визначаються виходами, і на них подається потенціал логічного 0. Потім виводи порту перевизначаються як входи, і проводиться зчитування їх стану. Факт замикання фіксується за зміною потенціалу, після чого проводиться друга фаза сканування. Цього разу на шини подається потенціал живлення, проводиться аналіз можливих станів шин і визначаються позиції замкнутих контактів. Наприклад, при використанні чотирьох шин із звичайною матричною топологією можна сканувати всього 4 кнопки, а при використанні вищеописаного методу вже 20 (рис. 1.48).

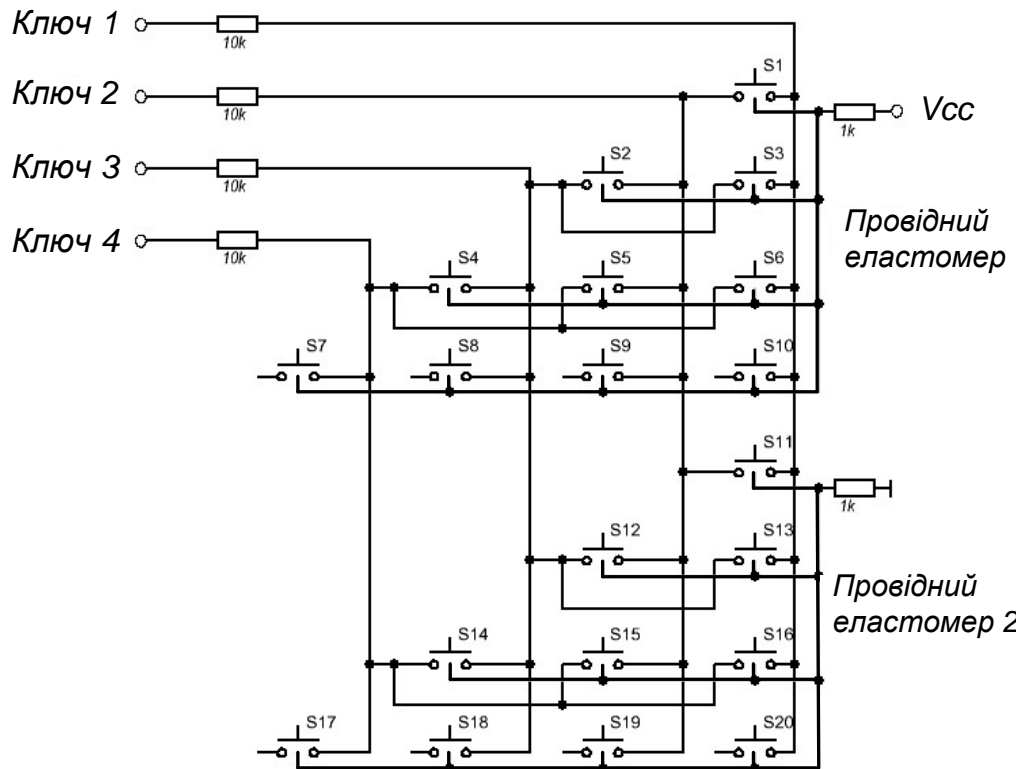


Рисунок 1.48 – Топологія клавіатури на 20 клавіш з 4-ма лініями сканування

1.7 Оптичні датчики

Як оптичний датчик найчастіше виступає світлодіод та фотоприймач, який називається оптроном. Випускаються оптрони з закритим (optoisolator) оптичним каналом (у монолітному виконанні) і відкритим оптичним каналом (щілинні і відбивальні оптрони).

1.7.1 Щілинний оптрон

На рисунку 1.49 показаний оптичний датчик – щілинний оптрон (slotted optical switch). Фототранзистор і направлений на нього світлодіод закріплені на пластиковій підставці і розділені проміжком так, що коли якийсь предмет рухається в зазорі, він перекриває світло між світлодіодом і датчиком. Щілинні оптрони часто використовуються для вимірювання швидкості двигуна за допомогою диска з прорізами, розміщеного на осі двигуна. Коли вісь обертається, диск перекриває світловий промінь. Інше застосування щілинного оптрона – це індикація того, відкриті чи закриті двері або, наприклад, кожух охоронного приладу. Прапорець на дверях, потрапляючи в

щілину, блокує світло, коли двері закриваються. Механічна комп'ютерна миша також використовує щілинні оптрони.

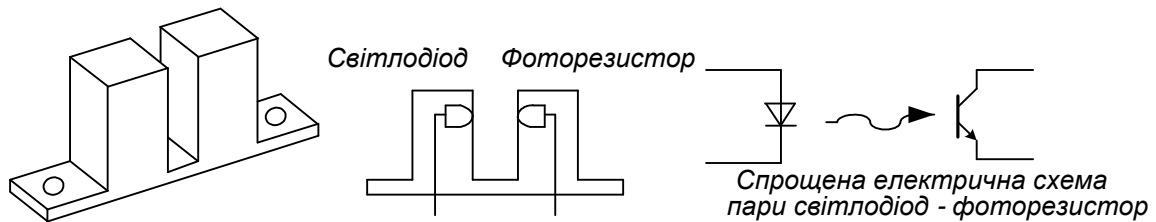


Рисунок 1.49 – Щілинний оптрон

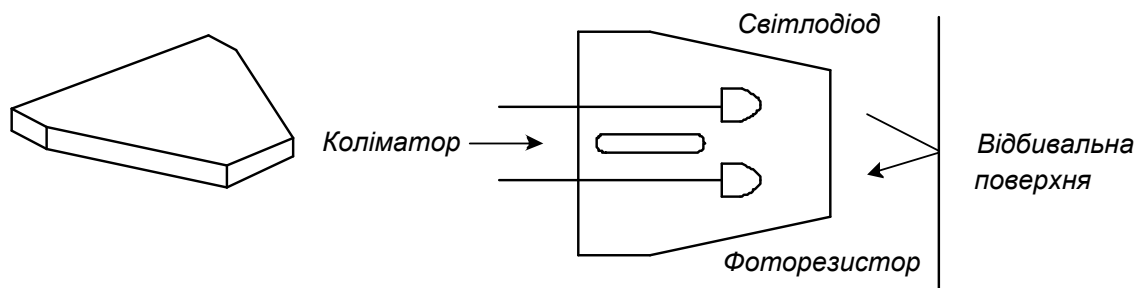


Рисунок 1.50 – Відбивальний оптрон

1.7.2 Відбивальний оптрон

На рисунку 1.50 показаний інший тип оптичного датчика – відбивальний оптрон (reflective sensor). Принцип роботи цього датчика такий же, як і щілинного, з тією різницею, що фототранзистор приймає відбите, а не пряме світло. Більшість датчиків відбиття характеризується фокусною відстанню – оптимальною відстанню, на якій має бути розміщений відбивальний об'єкт від датчика. Ця відстань дорівнює 0,254... 1,270 см (від 0,1 до 0,5 дюйма). Типове застосування відбивальних оптронів – це реєстрація обертання двигуна за нанесеними на його вісь темними мітками [1, 3, 16, 34, 35].

Коли вісь обертається, датчик реєструє зміну темних і відбивальних ділянок. Обидва типи оптронів мають схожі характеристики, які необхідно враховувати при проектуванні систем.

1.7.3 Швидкість спрацьовування оптронів

Оскільки фототранзистор є оптичним приладом, що діє досить повільно, то це обмежує максимальну швидкість реєстрації. Типовий час включення фототранзистора 8 мкс, а вимкнення 50 мкс. Ці часові параметри визначаються швидкістю носіїв під дією світла в переході база-емітер транзистора.

1.7.4 Коефіцієнт підсилення за струмом

Оптопара світлодіод-фототранзистор має обмежений коефіцієнт підсилення, зазвичай, менше одиниці. Відношення струму, що протікає в колекторі фото транзистора, до струму через світлодіод називається коефіцієнтом передачі за струмом (КПС) (Current Transfer Ratio, CTR). Типове значення КПС для щільних оптронів складає 0,1. Це означає, що при струмі 10 мА, що протікає через світлодіод, струм колектора фототранзистора складе 1 мкА. КПС інколи задається як коефіцієнт, інколи наводиться у вигляді таблиці, що показує різні значення струмів колектора для різних величин струму світлодіода. КПС залежить від характеристик світлодіода і фототранзистора. КПС слід брати до уваги при створенні інтерфейсу між оптронем і мікропроцесорною системою. По-перше, якщо оптрон приєднується до цифрового входу (рис. 1.51), вихід транзистора необхідно узгоджувати за логічними рівнями з входом цифрового пристрою.

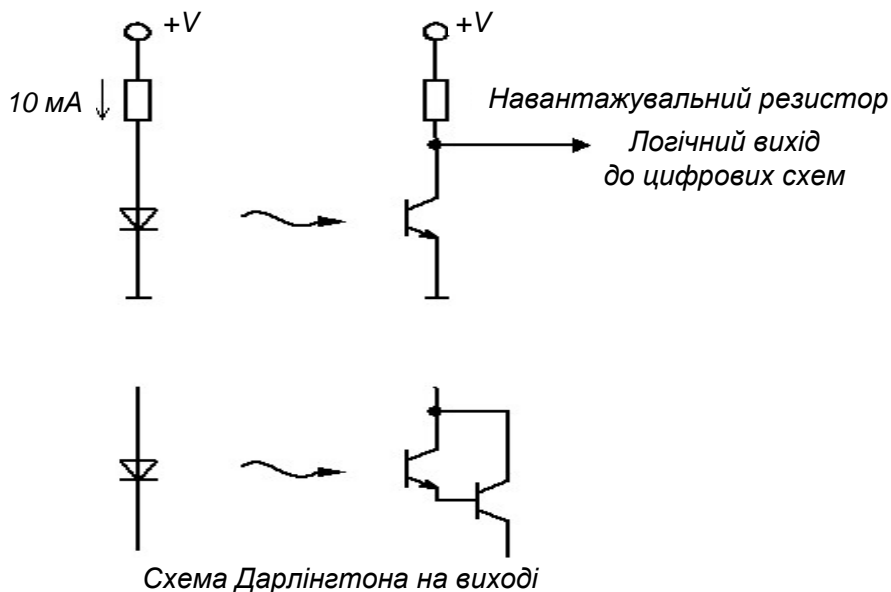


Рисунок 1.51 – Оптопари з цифровим виходом

Для швидкого насичення фототранзистора величину резистора навантаження слід обмежити. Наприклад, якщо через світлодіод протікає струм 10 мА, а мінімальна величина КПС дорівнює 0,1; то величину резистора навантаження слід вибрати 5 кОм [5, 26, 34]. Менша величина резистора забезпечить кращу стійкість до шумів (через менший імпеданс) і, можливо, велику швидкість перемикання, але не гарантує сумісності зі всіма пристроями, оскільки транзистор не зміг би пропускати достатній струм для забезпечення низького логічного рівня. Щоб знизити величину навантажувального резистора,

можна використовувати оптопару з більш високим КПС або жити світлодіод великим струмом.

Випускаються оптрони зі складеним транзистором Дарлінгтона на виході, що забезпечує КПС > 1 , але швидкість перемикавання такого оптрона нижча і складає 20% від швидкості поодиноких транзисторів. До того ж напруга насичення транзистора Дарлінгтона більша, ніж поодинокого транзистора.

Відбивальні датчики також характеризуються КПС. Оскільки датчик розрахований на прийом відбитого світла, то КПС залежатиме від властивостей відбивальної поверхні і відстані до неї. КПС відбивального датчика задається відносно деякої стандартної відбивальної поверхні, розміщеної на характерній фокальній відстані.

1.7.5 Особливості роботи в інфрачервоному діапазоні

У більшості щілинних і відбивальних датчиків використовуються інфрачервоні світлодіоди і фототранзистори. Це означає, що реакція датчика може відрізнитися від реакції у видимому діапазоні спектра. Об'єкти, що добре відбивають у видимому світлі, можуть відбивати менш ефективно в інфрачервоних (ІЧ) променях. Слід також враховувати завади, що вносяться природним і штучним освітленням.

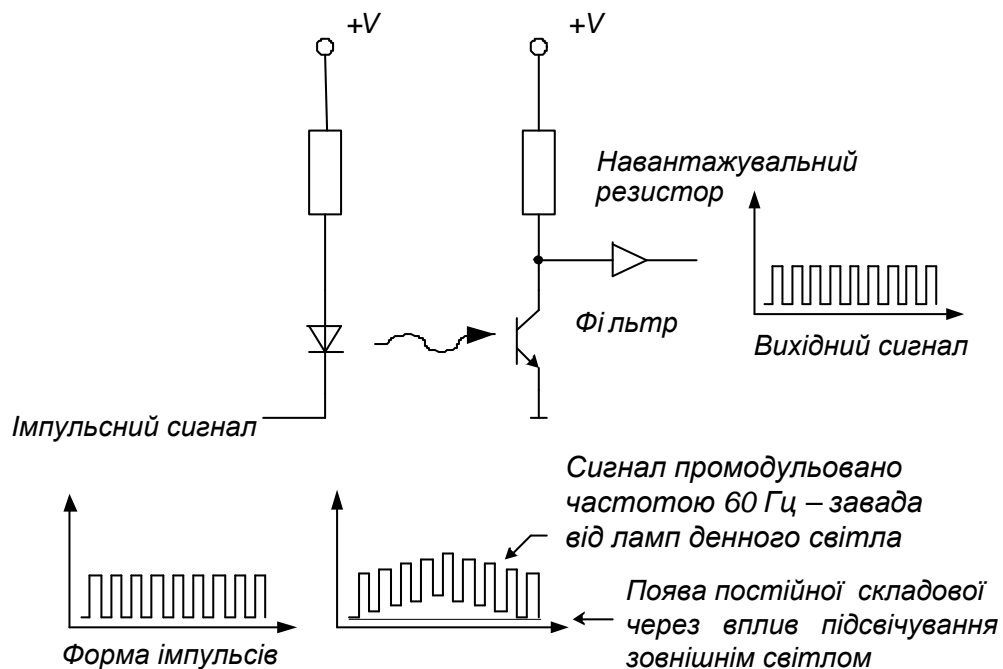


Рисунок 1.52 – Фільтрація сигналу оптичного датчика

На рисунку 1.52 показано, що оптрон, який працює з прямокутними імпульсами, може бути обладнаний фільтром для усунення подібних завад. У даному прикладі освітлювальний прилад додає до вихідного сигналу постійний зсув і коливання з частотою мережі 50 Гц. Після проходження сигналу через фільтр, налаштований на частоту паразитної модуляції мережі, ці завади вирізаються фільтром, і відфільтрований від завад сигнал може бути перетворений в цифровий. Фільтр може бути реалізований як за допомогою електроніки, так і програмно. ІЧ-промені широко застосовуються в побутовій техніці. У пульті дистанційного керування телевізора на ІЧ-променях використовується модуляція сигналу з частотою 40 кГц. Для забезпечення роботи з такою частотою в датчику застосовується високошвидкісний фотодіод.

Подібна фільтрація має і недоліки. Один з недоліків – обмеження швидкості. Через значний час вмикання і вимикання фототранзистора частота модуляції, з якою пристрій ще працюватиме, обмежена величиною порядку 10 кГц.

Недолік також в тому, що в результаті фільтрації з'являється деяка затримка в часі, що досягає декількох циклів роботи механічної частини. Так, якщо навіть датчик розрахований на роботу з частотою 10 кГц, в цілому система зможе підтримувати не вище 1 кГц, бо розраховується за найповільнішою частотою. Крім того, якщо зовнішнє освітлення має сильний вплив, то фототранзистор насичується і фільтрація не в змозі відновити вихідний сигнал. У даному випадку необхідно захистити фототранзистор від зовнішнього засвічування.

1.7.6 Дискретні оптичні датчики

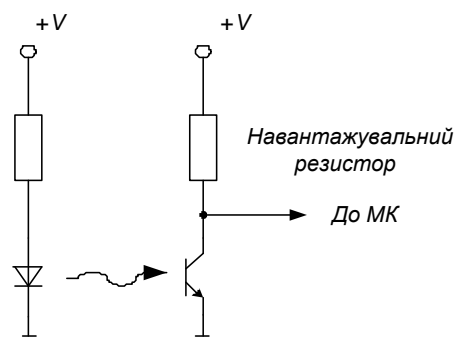
У деяких пристроях інколи потрібне застосування дискретних оптичних датчиків: світлодіода і фототранзистора. Вони розраховані на роботу в інфрачервоному діапазоні, як і ті, що застосовуються в щілинних і відбивальних оптронах. Дискретні оптопари використовуються в системах, де великий об'єкт заступає шлях світловому променю між світлодіодом і фото транзистором, або там, де відстань дуже велика для використання оптрона.

Дискретні оптопари мають аналогічні схеми включення і використовуються так само, як розглянуті раніше оптрони. Проте існує декілька додаткових зауважень. Оскільки відстань між світлодіодом і фототранзистором у такій системі більша, ніж в стандартній ІС, КПС, відповідно, нижчий. Часто з'являється необхідність підстроїти струм світлодіода або поріг чутливості для

надійної роботи схеми. У деяких випадках може знадобитися лінза (об'єктив) на один з елементів оптопари для фокусування світла.

Необхідність фокусування – досить поширена проблема, що виникає при використанні дискретних елементів. Ця проблема буде особливо відчутна, якщо світлодіод і фототранзистор розташовані на різних механічних підставках.

У оптроні світлодіод і фототранзистор узгоджені за довжиною світлової хвилі. Хоча велика частина фототранзисторів і світлодіодів ІЧ-діапазону можуть працювати спільно, але узгодження максимуму їх спектральної чутливості все ж може відрізнятись. Отже, при використанні дискретних елементів вибирають світлодіод і фототранзистор, що працюють в одній і тій же області ІЧ-спектра.



Може бути необхідне застосування тригера Шмітта.
Навантажувальні резистори в колі світлодіода і фоторезистора вибираються з урахуванням КПС для встановлення необхідних логічних рівнів.

Датчики деяких систем мають достатньо низький КПС для застосування такої схеми.

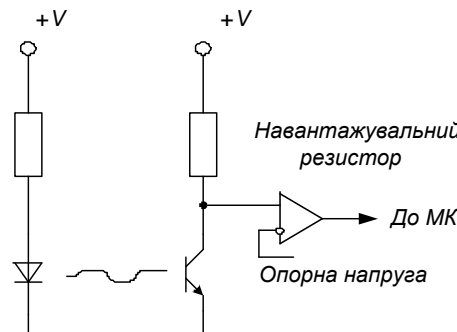
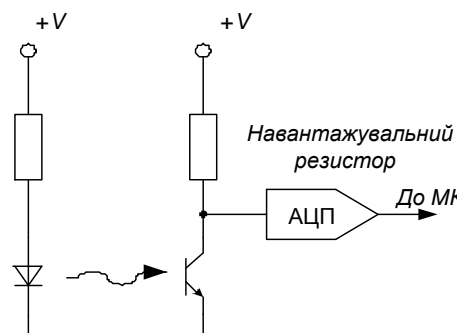


Схема забезпечує роботу оптичних приладів навіть у тому випадку, коли вихідний сигнал фоторезистора менший необхідного логічного рівня.

Рівень опорної напруги може бути змінений за допомогою потенціометра для встановлення напруги спрацьовування.



Найповільніша, оскільки кожний вимір потребує окремого аналого-цифрового перетворення.

Дана схема призводить до ускладнення всієї системи, оскільки програмне забезпечення повинно обробляти результати перетворення.

Така схема універсальна і може працювати з датчиками з різними КПС і оптичними системами.

Рисунок 1.53 – Схеми підключення оптичних датчиків до МК

На рисунку 1.53 показано три основні способи підключення оптичного датчика до МК. У всіх випадках світлодіод підключений через обмежувальний резистор, емітер фототранзистора з'єднується з землею, а колектор підключений до живлення через резистор навантаження і з'єднаний також з МК чи іншим пристроєм – компаратором або АЦП. Всі ці способи також ефективні і при інших підключеннях пристроїв керування, як, наприклад, з використанням МК для вмикання і вимикання світлодіода. Оптрони з закритим оптичним каналом працюватимуть і при іншій схемі підключення фототранзистора, наприклад, не за наведеною вище схемою зі спільним колектором, а за іншою – зі спільним емітером (живлення позитивної полярності подається на колектор, і сигнал знімається через резистор, що з'єднує емітер з землею).

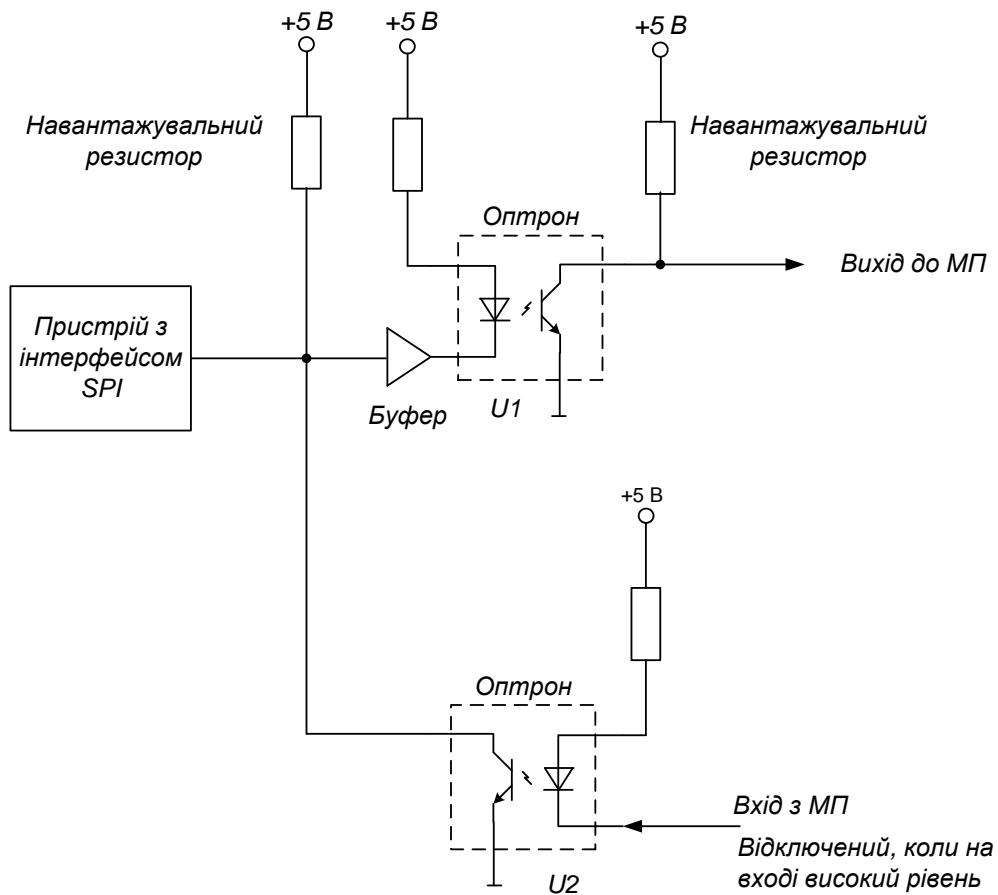


Рисунок 1.54 – Двонаправлена розв'язка з використанням оптронів із закритими оптичними каналами

На рисунку 1.54 показано, як оптрони з закритими оптичними каналами можуть бути використані для розв'язки двонаправленого сигналу між двома системами. На рисунку 1.54 показано, що вивід пристрою, що використовує інтерфейс *SPI*, потрібно ізолювати за постійним струмом від МП. Для забезпечення необхідної розв'язки

використовуються два оптрони з закритими оптичними каналами. Вихід *SPI*-пристрою забезпечений буфером (для забезпечення необхідного струму світлодіода), і вихід оптрона *U1* переходить до стану низького рівня, коли *SPI*-пристрій встановлює на виводі низький логічний рівень. Вихід *U1* може бути підключений до порту МП.

Другий оптрон (*U2*) встановлює вивід порта в нуль, коли МП формує на світлодіоді сигнал низького рівня. Якщо МП не передає дані на вивід I/O, він повинен залишити світлодіод у вимкненому стані (OFF state), так, щоб *SPI*-пристрій міг подавати сигнали на цей вивід. Коли МП встановлює на виводі I/O низький логічний рівень, вихід оптрона *U2* також встановиться в низький рівень, щоб МП ігнорував передачі на виході *U2* в той час, коли він керує *SPI*-пристроєм. З іншого боку, може бути використаний у відповідь сигнал (*return*) для перевірки коректного проходження даних через *SPI*-пристрій.

На рисунку 1.54 не показане коло, доповнене ще однією оптопарою і ще одним портом МП, які знадобляться для формування тактового сигналу інтерфейсу *SPI*. Керування двонаправленим виводом вимагає, щоб МП використовував два виводи порту (один вхід і один вихід), що дозволяє забезпечити ізоляцію за постійним струмом периферійного блока або системи. У більшості випадків необхідні високошвидкісні оптрони з закритими оптичними каналами. Цілком можливо, що підійдуть діодно-транзисторні оптрони чи оптрони з логічним виходом, але слід пам'ятати, що оптрон *U2* (див. рис. 1.54), безпосередньо з'єднаний з двонаправленим виводом, повинен мати вихід з відкритим колектором. Для забезпечення достатньої швидкості перемикання слід ретельно підібрати резистор навантаження, щоб уникнути надмірного часу наростання, без перевищення навантажувальної здатності виводу. Якщо як *U2* застосовується оптрон з діодно-транзисторним виходом, то він повинен забезпечувати достатній струм світлодіода для гарантованого встановлення виводу катода світлодіода в низький логічний рівень.

1.8 Цифроаналогові перетворювачі

1.8.1 Загальні відомості

Цифроаналоговий перетворювач (ЦАП) призначений для перетворення числа, визначеного, як правило, у вигляді двійкових кодів, у напругу або струм пропорційно значенню цифрового коду.

Дуже часто ЦАП входить до складу МПС. У цьому випадку, якщо не потрібна висока швидкодія, цифроаналогове перетворення

може бути дуже просто здійснено за допомогою ШІМ. Схема ЦАП з ШІМ наведена на рисунку 1.55.

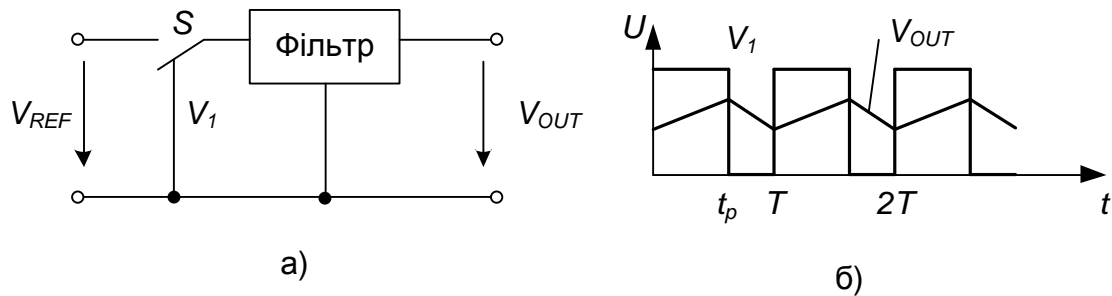


Рисунок 1.55 – ЦАП з широтно-імпульсною модуляцією:
а) структурна схема; б) часова діаграма

Найпростіше організується цифроаналогове перетворення в тому випадку, якщо МК має вбудовану функцію широко-імпульсного перетворення (наприклад, AT90S8515 фірми Atmel або 87C51GB фірми Intel) [24, 58, 59]. Вихід ШІМ керує ключем S . У залежності від заданої розрядності перетворення (для МК AT90S8515 можливі режими 8, 9 і 10 розрядів) контролер за допомогою власного таймера/лічильника формує послідовність імпульсів, відносна тривалість яких $\gamma = t_p / T$ визначається співвідношенням (формула 1.1):

$$\gamma = \frac{D}{2^N}, \quad (1.1)$$

де N – розрядність перетворення, а D – код перетворення.

Фільтр нижніх частот згладжує імпульси, виділяє середнє значення напруги. У результаті вихідна напруга перетворювача (формула 1.2):

$$V_{OUT} = \gamma V_{REF} = \frac{DV_{REF}}{2^N} \quad (1.2)$$

Розглянута схема забезпечує ідеальну лінійність перетворення, не містить прецизійних елементів (за винятком джерела опорної напруги). Основний її недолік – низька швидкодія.

Паралельні ЦАП мають більшу швидкодію і тому вони можуть застосовуватися для більш широкого кола задач. Більшість схем паралельних ЦАП реалізовано на додаванні струмів, що пропорційні вазі цифрових двійкових розрядів, причому повинні додаватися тільки струми тих розрядів, значення яких дорівнює 1.

1.8.2 Інтерфейси цифроаналогових перетворювачів

Важливу частину ЦАП складає цифровий інтерфейс, тобто схеми, що забезпечують зв'язок входів керування ключів з джерелами цифрових сигналів. Структура цифрового інтерфейсу визначає спосіб підключення ЦАП до джерела вхідного коду, наприклад, МП чи МК. Властивості цифрового інтерфейсу безпосередньо впливають і на форму вихідної характеристики ЦАП. Так, неодноразовість надходження розрядів вхідного слова на входи керування ключів перетворювача приводить до появи вузьких викидів ("голок") у вихідному сигналі при зміні коду.

При керуванні ЦАП від цифрових пристроїв з жорсткою логікою входів керування ключів, ЦАП можуть бути безпосередньо підключені до виходів цифрових пристроїв, тому в багатьох ІС ЦАП, особливо ранніх (572ПА1, 594ПА1, 1108ПА1, AD565) [11, 12, 26, 41, 59], будь-яка істотна цифрова частина відсутня. Якщо ж ЦАП входить до складу МПС і отримує вхідний код від шини даних, то він повинен бути забезпечений схемами, що дозволяють приймати вхідне слово від шини даних, зберігати його до отримання іншого слова, комутувати відповідно до цього слова ключі ЦАП. Для керування процесом завантаження вхідного слова ЦАП повинен мати відповідні входи керування і схему керування. Залежно від способу завантаження вхідного слова в ЦАП розрізняють перетворювачі з послідовним і паралельним інтерфейсами вхідних даних.

ЦАП з послідовним інтерфейсом вхідних даних, крім власне ЦАП, містить на кристалі послідовний регістр завантаження, паралельний регістр зберігання та логіку керування (рис. 1.56, а). При активному рівні сигналу \overline{CS} (в даному випадку низькому) вхідне слово довжиною N (рівною розрядності ЦАП) завантажується по лінії DI в регістр зсуву під керуванням тактової послідовності CLK . Після закінчення завантаження, виставивши активний рівень на лінію \overline{LD} , вхідне слово записують в регістр зберігання, виходи якого безпосередньо керують ключами ЦАП. Для того, щоб мати можливість передавати по одній лінії вхідні коди на декілька ЦАП, останній розряд регістра зсуву з'єднується з виводом ІС. Цей вивід підключається до входу DI наступного ЦАП і так далі. Коди вхідних слів передаються, починаючи з коду найостаннішого перетворювача.

Як приклад, на рисунку 1.56, б наведена часова діаграма, що відображає процес завантаження вхідного слова в ЦАП AD7233 [11, 12]. Мінімально допустимі значення інтервалів часу (порядку 50 нс), позначених на епюрах, вказуються в технічній документації на ІС.

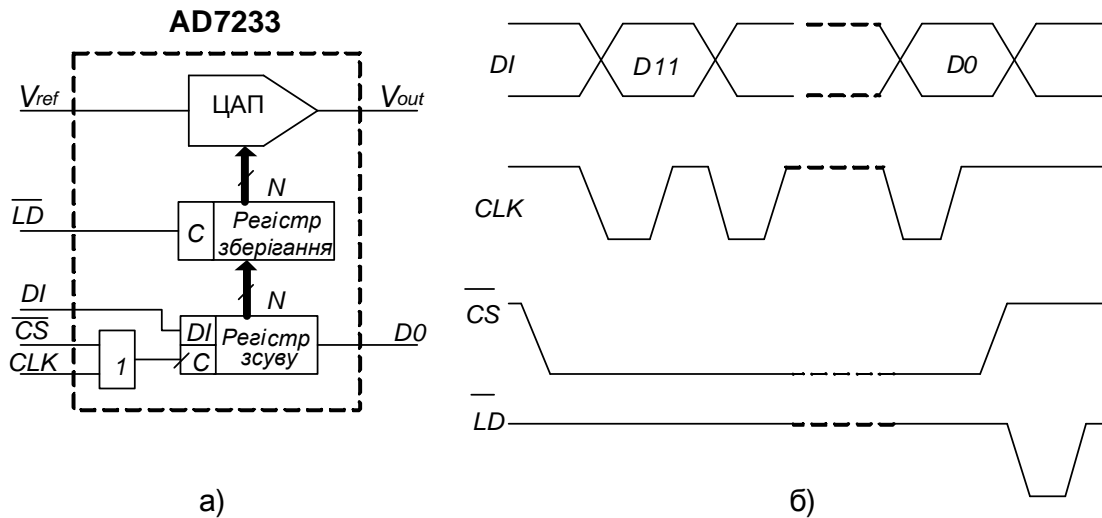


Рисунок 1.56 – ЦАП з послідовним інтерфейсом:
 а) функціональна схема; б) часова діаграма

Після закінчення завантаження МК змінює рівень на вході \overline{CS} , як це показано на рисунку 1.56, б. Виставивши активний рівень на вході \overline{LD} , ЦАП забезпечує пересилання вхідного коду з регістра зсуву ЦАП в регістр зберігання. Час завантаження залежить від тактової частоти МК і зазвичай складає одиниці мікросекунд. У випадку, якщо коливання вихідного сигналу ЦАП під час завантаження допустимі, вхід \overline{LD} можна з'єднати з загальною точкою схеми.

На рисунку 1.57 наведений варіант схеми підключення перетворювача з послідовним інтерфейсом до МК. На час завантаження вхідного слова в ЦАП через послідовний порт МК, до якого можуть бути також підключені і інші приймачі, на вхід \overline{CS} (вибір кристала) подається активний рівень з однієї з ліній введення/виведення МК.

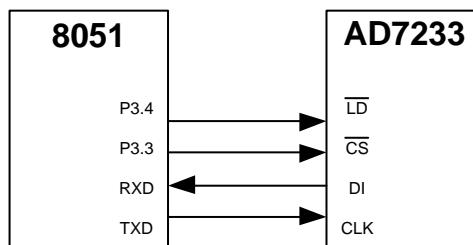


Рисунок 1.57 – Підключення ЦАП з послідовним інтерфейсом до МК MCS51

Стосовно ЦАП з паралельним інтерфейсом вхідних даних можна сказати, що частіше використовуються два варіанти. У

першому варіанті на N входів даних N -розрядного ЦАП подається вхідне слово повністю. Інтерфейс такого ЦАП містить два регістри зберігання і схему керування (рис. 1.58, а) Два регістри зберігання потрібні, якщо пересилання вхідного коду до ЦАП і установлення вихідного аналогового сигналу, відповідного цьому коду, повинні бути розділені в часі.

Подача на вхід асинхронного скидання \overline{CLR} сигналу низького рівня приводить до скидання першого регістра і, відповідно, вихідної напруги ЦАП.

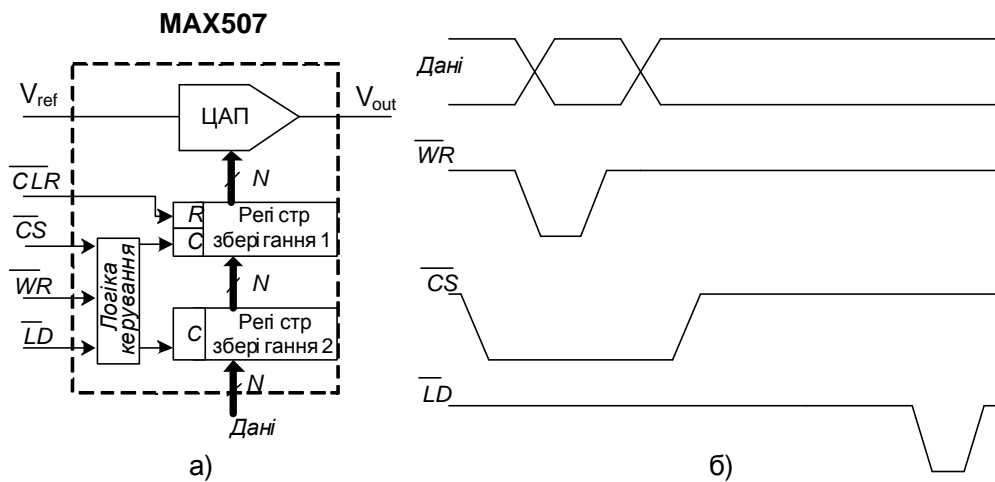


Рисунок 1.58 – ЦАП з паралельним інтерфейсом:
а) структурна схема; б) часова діаграма

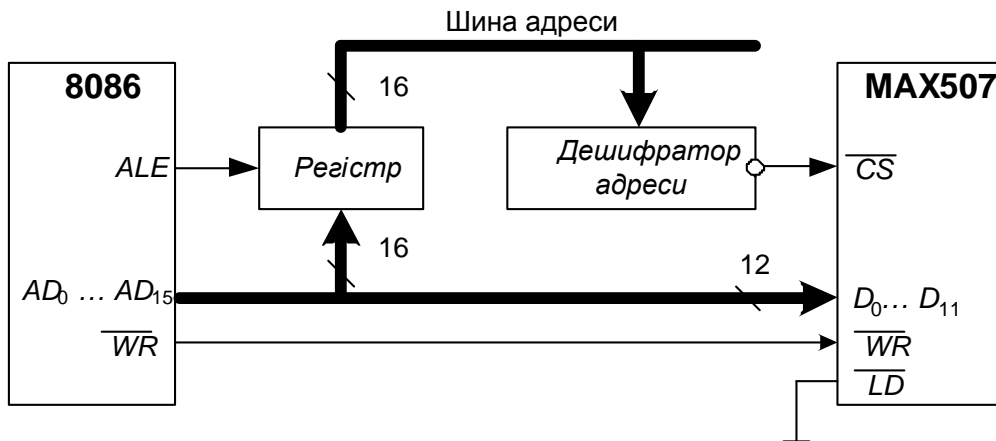


Рисунок 1.59 – Підключення ЦАП з паралельним інтерфейсом до МП Intel 8086

Приклад блок-схеми підключення 12-розрядного ЦАП MAX507 до 16-розрядного МП наведений на рисунку 1.59. Процесор посилає вхідний код в ЦАП як в елемент пам'яті даних. Спочатку з шини адреси/даних AD надходить адреса ЦАП, яка фіксується регістром за

командою з виходу ALE МП і, після дешифрування, активізує вхід \overline{CS} ЦАП. Зразу після цього МП подає на шину AD вхідний код ЦАП і потім сигнал запису на вхід \overline{WR} .

Для підключення багаторозрядних ЦАП до восьмирозрядних МП і МК використовується другий варіант паралельного інтерфейсу. Він передбачає наявність двох паралельних завантажувальних регістрів для прийому молодшого байта (МБ) вхідного слова і старшого байта (СБ) (рис. 1.60). Пересилання байтів вхідного слова до завантажувальних регістрів може відбуватися у будь-якій послідовності.

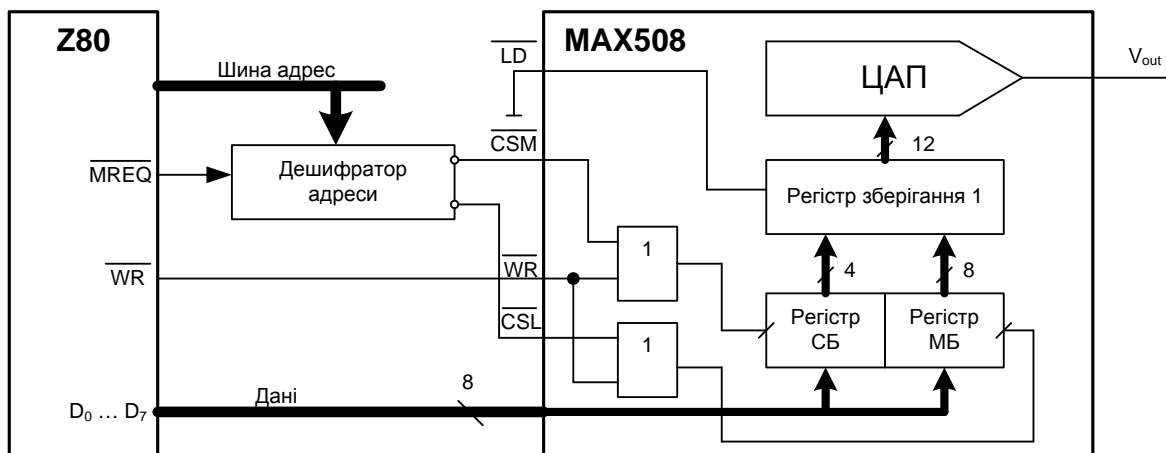


Рисунок 1.60 – Підключення ЦАП з паралельним інтерфейсом другого типу до восьмирозрядного МП

1.9 Аналого-цифрові перетворювачі

1.9.1 Загальні відомості

На рисунку 1.61 показаний простий резистивний АЦП східчастого типу з трьома ключами. Резистори складені в $R/2R$ конфігурації. Номінали резисторів не важливі; опір може бути 10 кОм, 100 кОм і тому подібне. Кожен з ключів $S_0...S_2$ може підключати один вивід одного резистора номіналом $2R$ між землею і вхідною опорною напругою, V_{REF} . На рисунку 1.61 показано, що відбувається, коли S_2 замкнутий "ON" (з'єднаний із V_{REF}), а S_0 і S_1 розімкнені "OFF" (з'єднані із землею). У результаті падіння напруги на послідовно-паралельній резистивній ділянці остаточна вихідна напруга (V_0) стає рівною $0,5V_{REF}$. Можна так само обчислити V_0 для всіх інших комбінацій ключів (табл. 1.1).

Якщо положення (замкнутий-розімкнутий) трьох ключів уявити,

як цифрове слово з трьох бітів, тоді можемо переписати таблицю, використовуючи позначення ON = 1 (замкнутий), OFF = 0 (розімкнутий) (табл. 1.2).

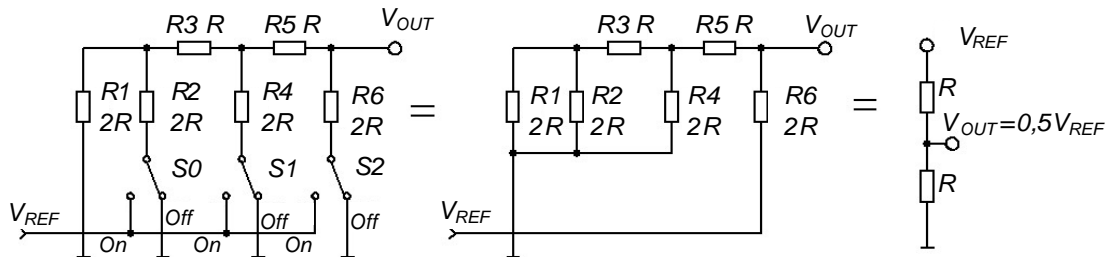


Рисунок 1.61 – Бітові АЦП

Таблиця 1.1 – Залежність вихідної напруги відносно стану ключів

Положення			Вихідна напруга V_{OUT}
S2	S1	S0	
OFF	OFF	OFF	0
OFF	OFF	ON	$0,125 \cdot V_{REF}(1/8 - V_{REF})$
OFF	ON	OFF	$0,25 \cdot V_{REF}(2/8 - V_{REF})$
OFF	ON	ON	$0,375 \cdot V_{REF}(3/8 - V_{REF})$
ON	OFF	OFF	$0,5 \cdot V_{REF}(4/8 - V_{REF})$
ON	OFF	ON	$0,625 \cdot V_{REF}(5/8 - V_{REF})$
ON	ON	OFF	$0,75 \cdot V_{REF}(6/8 - V_{REF})$
ON	ON	ON	$0,875 \cdot V_{REF}(7/8 - V_{REF})$

Таблиця 1.2 – Змінена таблиця залежності вихідної напруги відносно стану ключів

Положення ключа			Логічний еквівалент			S0...S2 чисельний еквівалент
S2	S1	S0	S2	S1	S0	
OFF	OFF	OFF	0	0	0	0
OFF	OFF	ON	0	0	1	1
OFF	ON	OFF	0	1	0	2
OFF	ON	ON	0	1	1	3
ON	OFF	OFF	1	0	0	4
ON	OFF	ON	1	0	1	5
ON	ON	OFF	1	1	0	6
ON	ON	ON	1	1	1	7

Таким чином, вихідна напруга є поданням комбінації положень ключів. Кожен додатковий біт у таблиці додає $V_{REF}/8$ до загальної напруги. Або, іншими словами, вихідна напруга дорівнює двійковому числу комбінації $S_0...S_2$, помноженому на $V_{REF}/8$. Такий 3-бітовий АЦП має 8 можливих станів, і кожен крок напруги складає $V_{REF}/8$.

Якщо додати ще одну $R/2R$ пару і ще один ключ до схеми, отримаємо схему з чотирма ключами і шістнадцятьма кроками по $V_{REF}/16$ вольт кожна. Схема з вісьмома ключами має 256 рівнів кожний по $V_{REF}/256$ вольт. Замінивши механічні ключі в схемі на електронні, можна створити повноцінний інтегральний АЦП.

1.9.2 Опис АЦП

Для передачі аналогового сигналу до МПС використовують АЦП. АЦП сприймає аналоговий сигнал, напругу або струм і перетворює його в цифрове слово, зрозуміле МП (рис. 1.62, а). На рисунку 1.62, б наводиться умовне зображення АЦП, який має вхід опорної напруги V_{REF} (reference) та вхід, куди подається сигнал. АЦП має один вихід – цифрове слово розмірністю 8 бітів, що являє в цифровій формі вхідну величину.

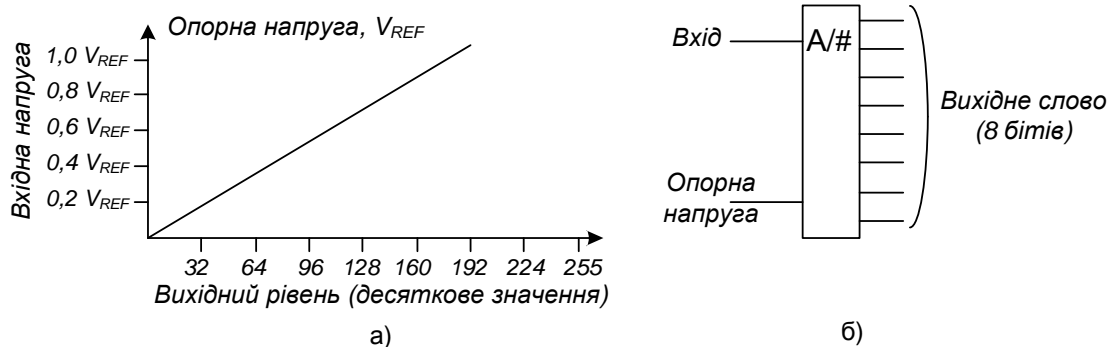


Рисунок 1.62 – До розгляду роботи АЦП

1.9.2.1 Опорна напруга

Опорна напруга, V_{REF} – максимальна величина напруги, яку АЦП може перетворити. АЦП, що зображений на рисунку 1.62, може перетворити напругу з величинами від 0 до V_{REF} . Цей діапазон напруги поділений на 256 рівнів або кроків. Розмір одного кроку задається так:

$$\frac{V_{REF}}{256} = \frac{5B}{256} = 0,0195B = 19,5 мВ \quad (1.3)$$

1.9.2.2 Вихідне слово

8-бітовий АЦП подає аналоговий сигнал цифровим словом. Старший значущий біт цього слова (біт 7) показує, чи вища вхідна напруга, ніж $V_{REF}/2$ (2,5 В при $V_{REF} = 5$ В). Кожен наступний біт (від біта 6 до біта 0, який буде молодшим значущим бітом – least significant bit, LSB) є половиною значення попереднього, як показано в таблиці 1.3.

Таблиця 1.3 – Подання аналогового входу цифровим словом

Біт	Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0
Напруга [В]	2,5	1,25	0,625	0,3125	0,156	0,078	0,039	0,0195

Так, маючи цифрове слово 0010 1100, можна визначити величину напруги (таблиця 1.4).

Таблиця 1.4 – Відповідність цифрового слова певним значенням

Біт	Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0
Напруга[В]	2,5	1,25	0,625	0,3125	0,156	0,078	0,039	0,0195
Вихідна	0	0	1	0	1	1	0	0

Додавши напругу, що відповідає кожному одиничному біту, отримуємо реальну напругу (формула 1.4).

$$0,625 + 0,156 + 0,078 = 0,859V \quad (1.4)$$

1.9.2.3 Роздільна здатність

Роздільна здатність АЦП визначається опорною напругою і розрядністю слова. Роздільна здатність – це найменша напруга, яка може бути виміряна АЦП і визначає точність перетворення. Роздільна здатність – це найменший крок, може бути визначена діленням V_{REF} на число можливих величин перетворення.

Наприклад, для 8-бітного АЦП з $V_{REF} = 5$ В роздільна здатність дорівнює 0,0195 В (19,5 мВ). Це означає, що будь-яка вхідна напруга, менша 19,5 мВ, дасть на виході 0. Вхідна напруга між 19,5 і 39 мВ приведе до 1 на виході. Між 39 і 58,6 мВ – до 2 і так далі. Роздільна здатність може бути покращена зменшенням V_{REF} . Зміна V_{REF} з 5 до 2,5 В дасть роздільну здатність 2,5/256 або 9,7 мВ. Проте максимальна напруга вимірювання тепер складе не 5, а 2,5 В.

Є лише один шлях збільшення роздільної здатності без зміни опорної напруги – використання АЦП з більшою розрядністю слова.

Так 10-бітовий АЦП з опорною напругою 5 В має 2^{10} , або 1024 можливих вихідних двійкових кодів. Роздільна здатність складе 5 В/1024 або 4,88 мВ.

1.9.3 Типи АЦП

Існує декілька типів АЦП, що працюють з різними швидкостями, мають різні інтерфейси і забезпечують різну точність. Розглянемо їх особливості [11, 26, 41].

1.9.3.1 Слідкувальний АЦП

Слідкувальний АЦП складається з компаратора, реверсивного лічильника, керованого компаратором, і ЦАП. Компаратор порівнює вхідну напругу з вихідною напругою ЦАП і керує напрямком лічби реверсивного лічильника (+/-). Якщо вхідна напруга більша, ніж напруга ЦАП, лічильник під дією тактових імпульсів, що надходять на його вхід (рис. 1.63, *a*) рахує на додавання (+), якщо ж менше напруги ЦАП, лічильник рахує на віднімання (-).

Вхід ЦАП з'єднаний з виходом лічильника. Припустимо $V_{REF} = 5$ В. Це означає, що перетворювач зможе працювати в діапазоні вхідної напруги 0...5 В. Якщо старший значущий біт вхідного слова ЦАП дорівнює 1, то $V_{ЦАП} = 2,5$ В. Якщо наступний біт дорівнює 1, додається 1,25 В, в результаті встановлюється 3,75 В. Кожен наступний біт додає половину вихідної напруги, що відповідає попередньому біту. Таким чином, вхідні біти ЦАП відповідають певній напрузі (табл. 1.3, 1.4).

На рисунку 1.63, *a* показано, як слідкувальний АЦП обробляє постійний сигнал з рівнем $V_{BX} = 0,37$ В. Лічильник стартує з нуля, при цьому вихід компаратора знаходиться у високому логічному рівні. Лічильник рахує на додавання з кожним тактовим імпульсом, ступінчасто піднімаючи $V_{ЦАП}$. Коли лічильник проходить двійкову величину, яка є V_{BX} , то компаратор перемикається, і лічильник починає рахувати на віднімання. У результаті, лічильник хитається коло величини, яка є V_{BX} .

Головний недолік слідкувальних АЦП – низька швидкість перетворення. Перетворення може зайняти до 256 тактів 8-бітового виходу, 1024 такти для 10-бітового значення і так далі. До того ж, швидкість перетворення змінюється залежно від V_{BX} . Якби напруга в даному прикладі складала 0,18 В, перетворення зайняло лише половину тактів у порівнянні з $V_{BX} = 0,37$ В.

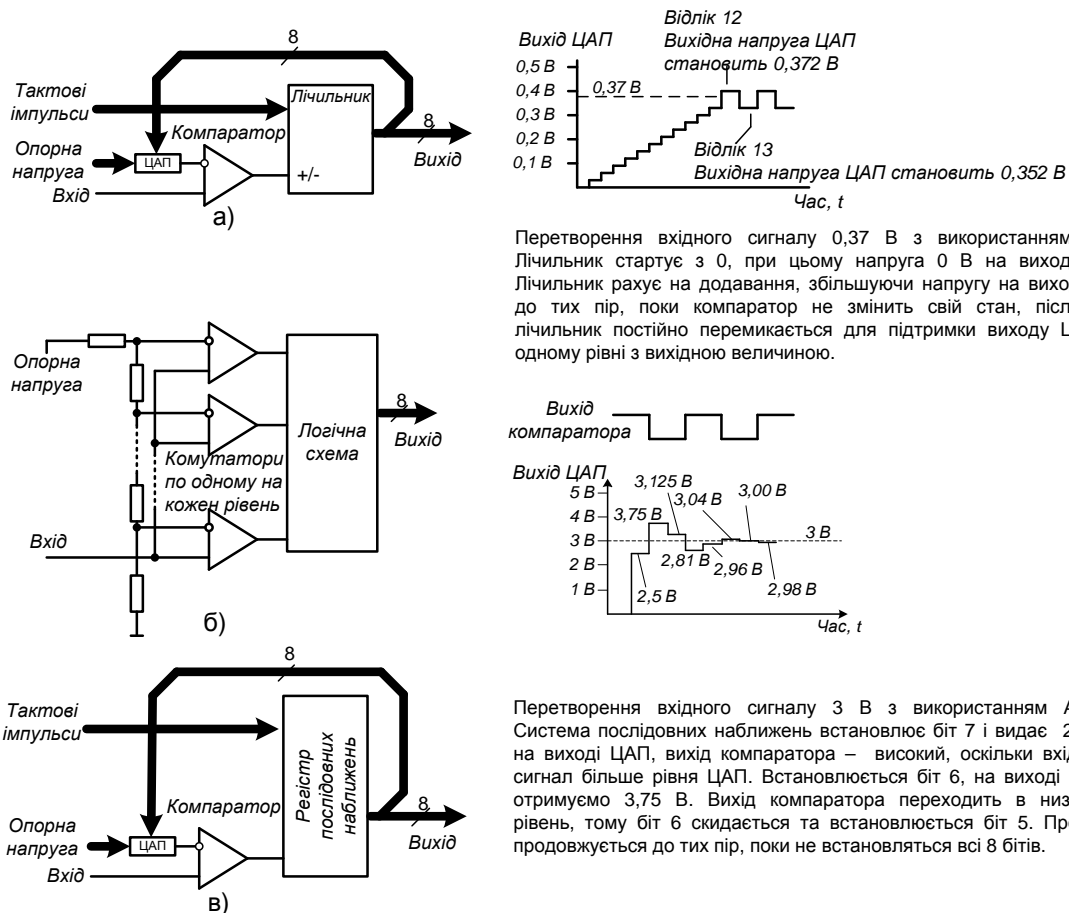


Рисунок 1.63 – Три типи АЦП: а) слідкувальний; б) паралельний; в) послідовних наближень

Максимальна тактова частота АЦП слідкувального типу залежить від затримки поширення сигналу в ЦАП і компараторі. Після кожного тактового імпульсу вихід лічильника має бути перетворений ЦАП, а результат перетворення з виходу потрапити на інверсний вхід компаратора. Компаратор віднімає деяку частину часу для реакції на зміну $V_{ЦАП}$, створюючи новий сигнал керування на додавання чи віднімання для реверсивного лічильника. Всі затримки сигналу роблять слідкувальний АЦП найповільнішим з усіх представлених типів АЦП. Слідкувальні АЦП широкого поширення не отримали; серед інтегральних схем, що випускаються такими фірмами, такими як Analog Devices, Maxim, Burr-Brown, неможливо знайти слідкувальних АЦП. Треба відзначити, що АЦП послідовних наближень працює з великою швидкістю з такою розрядністю слова. Проте зустрічаються такі випадки, коли слідкувальний АЦП буде корисним. Наприклад, якщо сигнал змінюється повільно відносно частоти дискретизації, слідкувальний АЦП встановлюватиме вихідне число за менше число тактів, ніж АЦП послідовних наближень.

1.9.3.2 Паралельний АЦП

Паралельний АЦП (рис. 1.63, б) – найшвидший з тих, що випускаються виробниками. Паралельний АЦП містить по компаратору на кожен крок напруги. На один із входів кожного компаратора подається вимірювана напруга. Інші входи компараторів підключаються до лінійки резисторів. Якщо рухатися вгору по цій лінійці, то можна відмітити, що кожен наступний компаратор підключений до вищої напруги з постійним кроком приросту напруги. Залежно від зміни вхідної напруги будуть переключені у стан 1 лише ті компаратори, в яких було перевищено опорну напругу. Сигнали з виходів всіх компараторів надходять в логічний блок – шифратор, який і задає вихідне цифрове m -бітове слово АЦП залежно від того, які компаратори в стані 0, а які в стані 1. Число компараторів визначається числом кодових комбінацій $2^m - 1$, де m – роздільна здатність АЦП. Тоді для 3-бітового АЦП буде потрібно 7 компараторів, для 4-бітового – 15 компараторів і так далі. Роздільна здатність паралельного АЦП лежить в межах, як правило, від 6 до 12 бітів. Нескладно підрахувати, що для 6 бітного АЦП потрібно 63 вхідних компаратора.

Швидкість перетворення паралельного АЦП складається з часу затримки одного компаратора і логічного блока. Паралельні АЦП найбільш швидкодійні серед основних типів АЦП. Їх швидкодія складає десятки частки мікросекунд. До недоліків паралельних АЦП можна віднести різке збільшення вхідних компараторів разом з збільшенням роздільної здатності. З цієї причини для них недосяжний бар'єр у 12 бітів, навіть з збільшенням міри інтегрування мікросхем. Такі величезні об'єми інтегральних елементів (компараторів і шифраторів) в одній мікросхемі реально впливають на робочі характеристики. Використання великого числа компараторів веде до збільшення споживаного струму. Так, наприклад, 10-бітовий паралельний АЦП може споживати струм до 0,5 А. У свою чергу, струм споживання визначає потужність розсіювання корпусом мікросхеми, тому 8-12-бітові АЦП повинні мати на платі ефективний тепловідвід від корпусу АЦП.

1.9.3.3 АЦП послідовного наближення

АЦП послідовного наближення (рис. 1.63, в) подібний до слідкувального АЦП в тому, що система ЦАП/лічильник формує напругу, яка надходить на один вхід компаратора, а вхідний сигнал надходить на інший вхід. Відмінність полягає в тому, що регістр послідовного наближення виконує двійковий пошук, замість лічби вгору або вниз по одиниці. Візьмемо для прикладу, що початкова вхідна напруга складає 3 В, а опорна – 5 В (див. рис. 1.63). Регістр

послідовного наближення виконає перетворення таким чином:

Встановити MSB, напруга на виході ЦАП дорівнює 2,5 В
Вихід компаратора - високий логічний рівень, MSB залишається встановленим.
Результат: 1000 0000 Встановити біт 6, напруга на виході ЦАП дорівнює 3,75 В (2,5 + 1,25)
Вихід компаратора - низький логічний рівень, скидається біт 6.
Результат: 1000 0000 Встановити біт 5, напруга на виході ЦАП дорівнює 3,125 В (2,5 + 0,625)
Вихід компаратора - низький логічний рівень, скидається біт 5.
Результат: 1000 0000 Встановити біт 4, напруга на виході ЦАП дорівнює 2,8125 В (2,5 + 0,3125)
Вихід компаратора - високий, залишається біт 4 встановленим.
Результат: 1001 0000 Встановити біт 3, напруга на виході ЦАП рівна 2,968 В (2,8125 + 0,15625)
Вихід компаратора - високий, залишається біт 3 встановленим.
Результат: 1001 1000 Встановити біт 2, напруга на виході ЦАП дорівнює 3,04 В (2,968 + 0,07125)
Вихід компаратора - низький, скидається біт 2.
Результат: 1001 1000 Встановити біт 1, напруга на виході ЦАП дорівнює 3,007 В (2,8125 + 0,039)
Вихід компаратора - низький, скидається біт 1.
Результат: 1001 1000 Встановити біт 0, напруга на виході ЦАП дорівнює 2.988 В (2.8125 + 0.0195)
Вихід компаратора - високий, залишається біт 0 встановленим.
Результат: 1001 1001

При використанні 8-бітового ЦАП з вихідною напругою 0...5 В, цьому результату відповідає напруга, розрахована у формулі 1.5.

$$2,5 + 0,3125 + 0,15625 + 0,0195 = 2,988 \text{ В} \quad (1.5)$$

Це не точно 3 В, але настільки близько, наскільки можна отримати з 8-бітовим перетворенням і опорною напругою 5 В.

8-бітовий АЦП послідовного наближення може завершити перетворення за 8 тактів, незалежно від вхідної напруги. Потрібно більше логічних кіл, ніж для слідкувального АЦП, але швидкість перетворення буде вища.

1.9.3.4 АЦП подвійного інтегрування

АЦП подвійного інтегрування (рис. 1.64) використовує інтегратор, за яким ставлять компаратор і лічильник. Вхід інтегратора підключається до вхідного сигналу, і ємність інтегратора заряджається до рівня вхідної напруги тієї ж полярності. Після певного числа тактів вхід інтегратора перемикається до джерела опорної напруги (V_{REF1} на рис. 1.64), і ємність інтегратора розряджається до величини цієї напруги. У той момент, коли ключ замикається на V_{REF1} , лічильник відраховує стільки ж тактів, скільки займав час першого інтегрування.

Коли напруга на виході інтегратора падає нижче за величину другої опорної напруги (V_{REF2} на рис. 1.64), вихід компаратора переходить у стан високого логічного рівня, лічильник зупиняється, а значення лічильника відповідає величині вхідної напруги. Більша вхідна напруга дозволяє ємності інтегратора заряджатися до більшої величини протягом часу первинного інтегрування, що приводить до більшого часу розрядження до V_{REF2} , і до більшого вихідного значення лічильника. Менше значення напруги на вході приводить до меншого потенціалу на ємності інтегратора і, відповідно, до меншого вихідного числа.

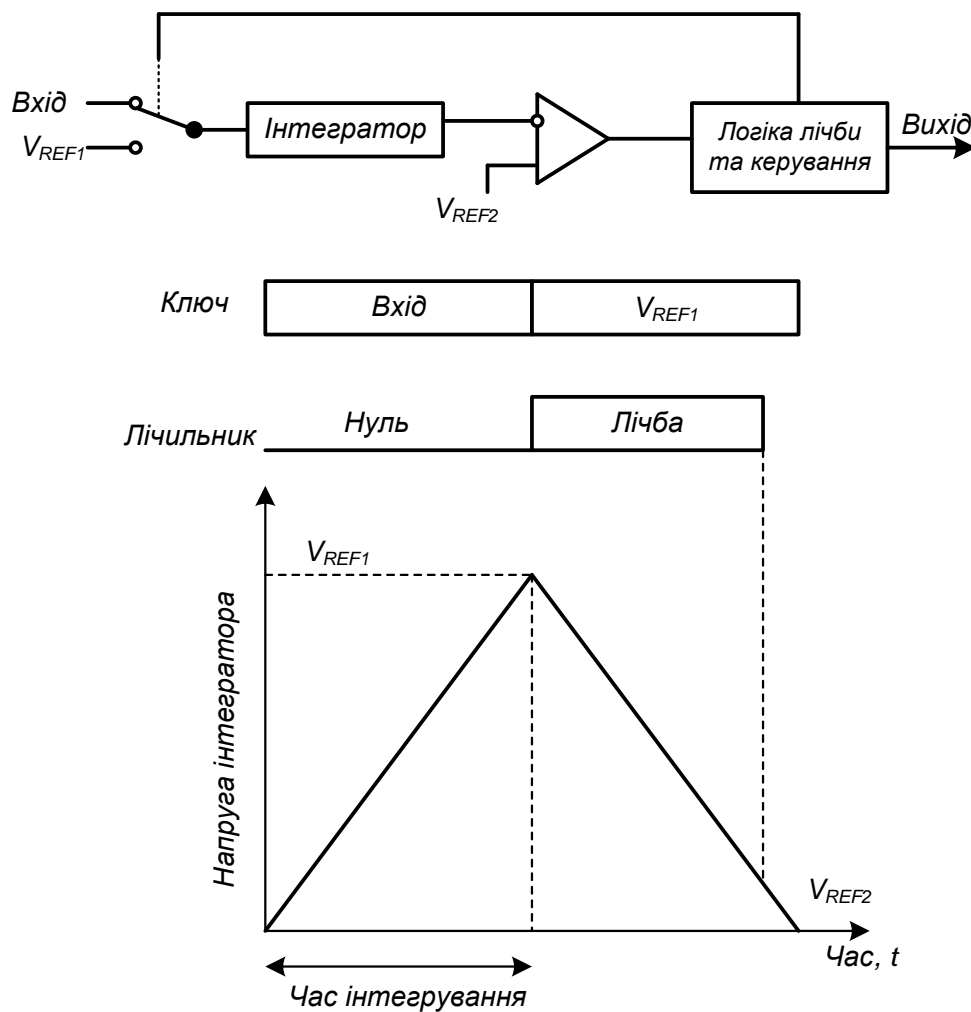


Рисунок 1.64 – АЦП подвійного інтегрування

Схема з подвійним інтегруванням усуває проблеми точності синхронізації, оскільки один і той же генератор тактових імпульсів застосовується для задання часу заряду ємності і для приросту вмісту лічильника. Перетворювач з подвійним інтегруванням витрачає відносно великий час на виконання перетворення, але властива інтегратору фільтрація усуває шум.

1.9.4 Інтерфейси АЦП

1.9.4.1 Кодування вихідного слова

Наведені до цих пір приклади базувалися на двійкових кодах, де кожен біт являв собою певну величину напруги, відповідну вазі біта, а вся напруга відповідала сумі ваг кожного одиничного біта у вихідному слові і була величиною вхідної напруги. Основні типи АЦП подають вихідне слово в прямому коді числа. Деякі АЦП подають вихідне слово в додатковому коді числа, де негативна напруга має цифровий еквівалент у вигляді доповнення до двох прямого коду числа. Деякі АЦП подають вихідне слово в двійково-десятковому коді (Binary Coded Decimal, BCD). Але таке подання простіше сприймається людиною, а не обчислювальними машинами. Очевидно, що таке подання вимагає більшої розрядності для даного діапазону; 12-бітовий двійковий вихід подаватиме величини від 0 до 4095, а в двійково-десятковому 12-бітовому виході діапазон величин зменшиться (0...3999). Даний код (BCD) використовується в АЦП з виходом на цифрові індикатори.

1.9.4.2 Паралельний інтерфейс

АЦП, що випускаються промисловістю, забезпечуються різними вихідними паралельними і послідовними інтерфейсами для роботи з багатьма типами МК. Деякі ІС містять більше одного інтерфейсу для забезпечення сумісності з якомога більшою кількістю МК.

ІС MAX151 фірми MAXIM/Dallas є типовим 10-бітовим АЦП з 8-бітовим універсальним периферійним паралельним інтерфейсом для зв'язку з МК [11, 41]. Як показано на рисунку 1.65, інтерфейс Max151 містить 8 бітів даних, виходи сигналів: \overline{CS} (chip select) – "вибір мікросхеми", \overline{RD} (read strobe) – "строб-імпульс читання", \overline{BUSY} – "зайнято". Також Max151 містить вбудований пристрій вибирання-зберігання (ПВЗ).

На спаді імпульсу на входах \overline{RD} і \overline{CS} ПВЗ переходить у режим зберігання, і стартує аналого-цифрове перетворення. У більшості систем вихід \overline{CS} з'єднується з декодером адреси і тому сигнал на ньому переходить в низький логічний рівень перед сигналом на вході \overline{RD} . Як тільки перетворення почалося, АЦП встановлює сигнал на виводі \overline{BUSY} у низький (активний) рівень. Вивід \overline{BUSY} залишається в низькому логічному рівні до тих пір, поки не завершиться перетворення.

На першій стадії роботи, яка в специфікації фірми MAXIM називається режимом повільної пам'яті (Slow Memory Mode), МК чекає, утримуючи \overline{RD} і \overline{CS} на низькому рівні доки перетворення не

завершиться. Максимальний час перетворення для MAX151 складає 2,5 мкс.

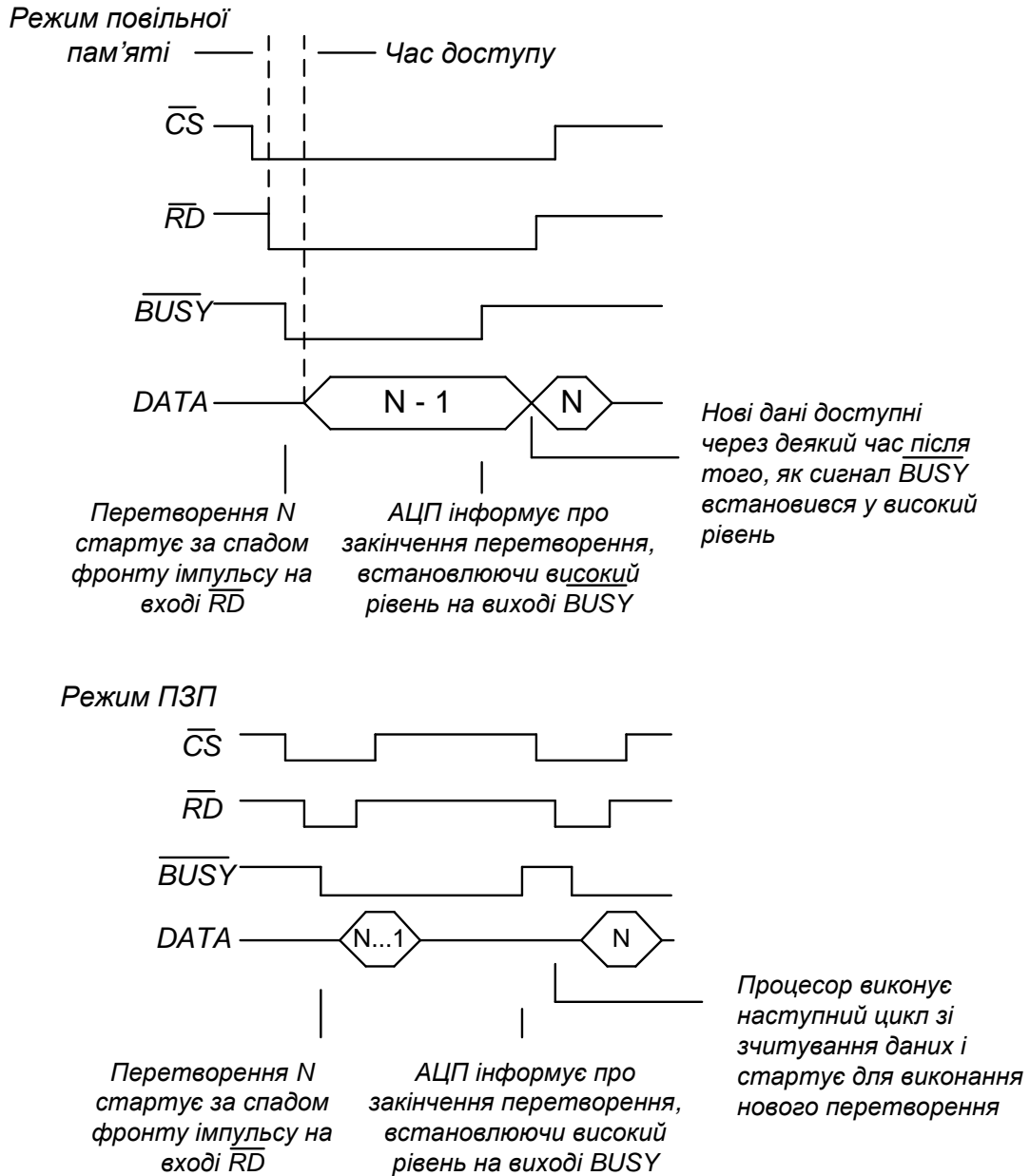


Рисунок 1.65 – Інтерфейс ІМС MAX151

Друга стадія називається режим ПЗП (ROM-mode). На цій стадії процесор виконує цикл читання, в якому ПЗП переходить у режим фіксації рівня, і АЦП починає перетворення. Під час цієї стадії МК зчитує результати попереднього перетворення. Сигнал \overline{RD} не використовується на стадії читання даних. Замість цього вивід \overline{BUSY} з'єднується з виводами МК "запит переривань" або "почерговий опит пристроїв" (polling) для індикації завершення перетворення. Коли

рівень \overline{BUSY} стає високим, МК проводить наступне читання, щоб отримати наступний результат і почати наступне перетворення. Хоча в специфікаціях (data sheets) [11, 36] ці операції відносять до двох різних стадій роботи, АЦП працює аналогічним чином в обох випадках:

- за спадом імпульсів на входах \overline{RD} і \overline{CS} починається цикл перетворення;
- поточний результат стає доступний на шині після того, як закінчився час доступу читання;
- поки \overline{RD} і \overline{CS} залишаються в стані низького логічного рівня, поточний результат залишається доступним на шині;
- після завершення циклу перетворення дані перетворення стають доступні процесору; якщо \overline{RD} і \overline{CS} все ще залишаються в низькому рівні, ці дані заміщають на шині результат попереднього перетворення.

1.9.4.3 Послідовний інтерфейс SPI

Периферійний послідовний інтерфейс SPI застосовується як для з'єднання МК між собою, так і МК з периферійними пристроями. У одному сеансі зв'язку беруть участь лише 2 пристрої, з яких один обов'язково МК, а інший або МК, або периферійний пристрій з інтерфейсом SPI (АЦП, датчик, пам'ять, виконавчий пристрій). У периферійному послідовному інтерфейсі SPI використовуються цифрові сигнали "такти" (clock), «вибір кристала (мікросхеми)» (chip select), "вхід даних" (data input) і "вихід даних" (data output), але немає адресних сигналів. Дані зчитуються з виходу послідовного АЦП по біту за такт (рисунок 1.66).

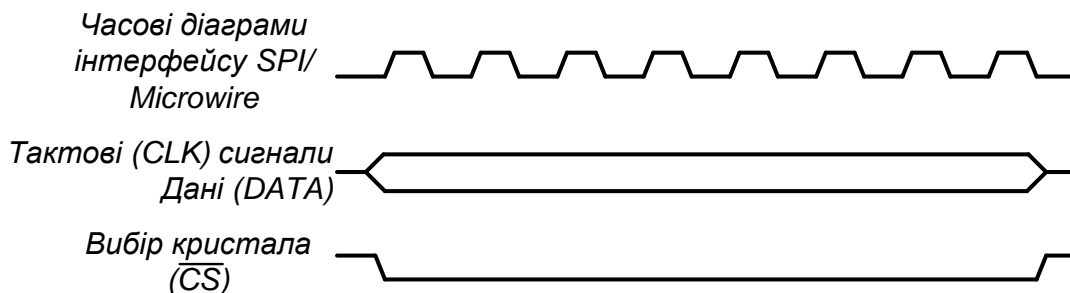


Рисунок 1.66 – Шина SPI

Кожен з пристроїв, підключених до шини SPI, вимагає наявності окремого сигналу \overline{CS} , яким він вибирається. Два пристрої працюють в режимі "ведучий-ведений". Дані, які потрібно передати,

завантажуються в 8-бітові регістри портів і вхід даних ведучого пристрою з'єднується з виходом даних веденого. У свою чергу, вихід даних ведучого пристрою з'єднується з входом даних веденого. Також підключені до МК (ведучого пристрою) останні вище перелічені ведені пристрої, які включаються за окремими сигналами "вибір кристала". Хоча регістри мають по 8 бітів, але при такому підключенні входів і виходів утворюється спільний 16-бітовий регістр зсуву і, крім того, одночасно передаються по двох лініях дані в обох напрямках. Зсув проводиться тактовими сигналами від тактового генератора ведучого пристрою.

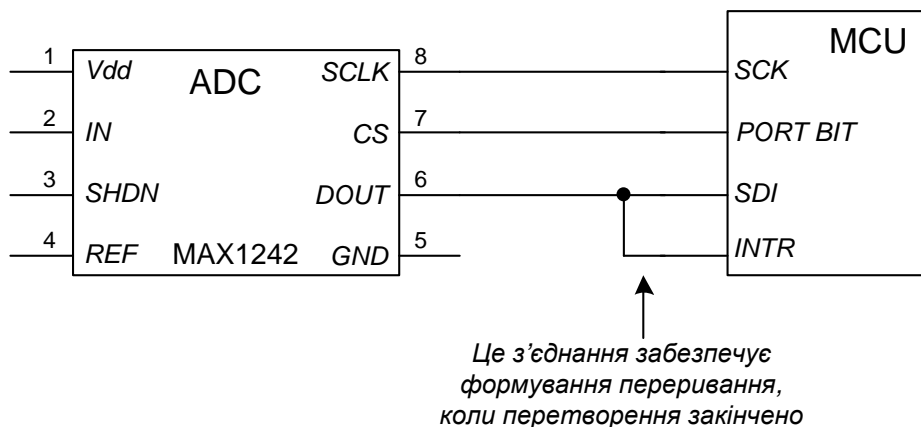
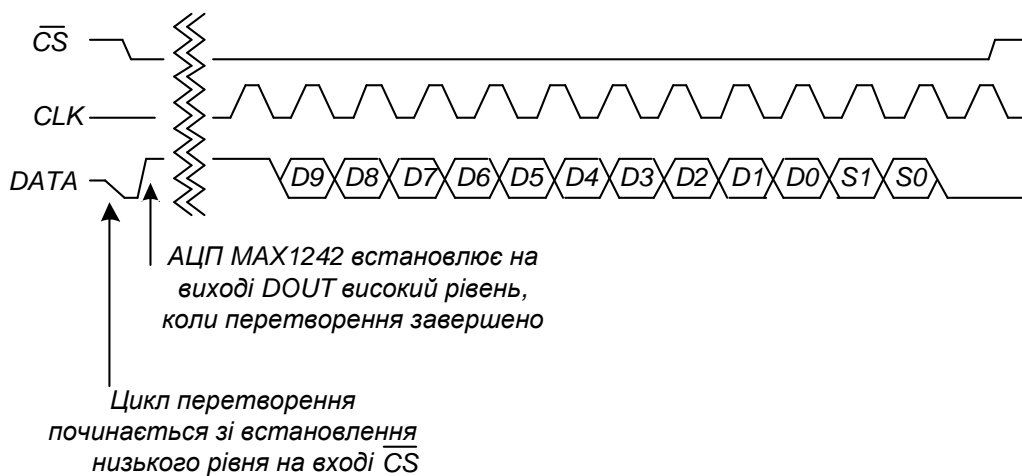


Рисунок 1.67 – Інтерфейс АЦП MAX1242

ІС MAX1242 фірми MAXIM/Dallas є типовим прикладом АЦП з інтерфейсом SPI. MAX1242 – 10-бітний АЦП послідовного наближення з ПВЗ в одному 8-вивідному корпусі [11, 41]. На рисунку 1.66 наведені часові діаграми роботи послідовного інтерфейсу ІС MAX1242. За спадом сигналу \overline{CS} починається перетворення, яке триває максимум 7,5 мкс. Після завершення перетворення на виводі (data output)

встановлюється високий логічний рівень. Процесор потім може зчитати дані біт за бітом, синхронізуючи читання сигналами "такти" на лінії тактових імпульсів (clock line) і керувати виведенням даних на виводах *DOUT* (data output) мікросхеми MAX1242. Після того, як 10 бітів даних прочитані, IC MAX1242 формує два службових біти (sub-bits) *S1* і *S0*. Після проходження 13 тактів синхронізації IC встановлює вихід в 0, що означає закінчення видачі даних з виводу *DOUT* (data output).

На рисунку 1.67 показано, як АЦП MAX1242 може бути з'єднаний з МК за допомогою вбудованого інтерфейсу SPI.

Сигнал *SCLK* з АЦП Max1242 надходить на *SCK* SPI МК, а вихід *DOUT* АЦП MAX1242 з'єднується з *SDI* входом даних МК. Один з виводів порту МК генерує сигнал \overline{CS} для вибору IC MAX1242. Шина SPI буде недоступна для з'єднання з іншими периферійними пристроями до тих пір, поки не закінчиться перетворення і не буде прочитаний результат. Якщо є підпрограми обробки переривань, що використовують SPI-пристрої, вони мають бути заборонені під час перетворення. Для уникнення пов'язаних з перериваннями проблем АЦП MAX1242 міг би звертатися до МК за допомогою спеціалізованої (нестандартної) SPI-шини. Такий підхід вимагає використання додатково трьох виводів МК, оскільки більшість МК якщо і мають, то лише один SPI-інтерфейс. Усі необхідні часові діаграми спеціалізованої SPI-шини мають бути виконані програмно і виведені на інший порт (вивід) МК.

Можна генерувати переривання МК після того, як аналого-цифрове перетворення завершиться. Спеціальне з'єднання для виконання такої функції показано на рисунку 1.67. Вивід *DOUT* АЦП MAX1242 з'єднується з виводом *INTR* ("переривання") МК. Коли на \overline{CS} низький рівень і перетворення завершено, сигнал *DOUT* перейде у високий рівень, що викликає переривання на МК. Для використання даного методу програмне забезпечення повинно заборонити або ігнорувати переривання під час перетворення.

Іншим прикладом АЦП зі SPI-інтерфейсом є MCAD7823 фірми Analog Devices [11, 12, 14, 41]. На відміну від MAX1242, АЦП AD7823 не змінює логічний рівень на виводі даних, поки МК не почне зчитувати результат, тому шина інтерфейсу SPI може використовуватися іншими пристроями доки йде перетворення. Оскільки немає зміни рівня на виводі даних, то немає й спеціального сигналу для МП про те, що перетворення завершилося. Процесор сам повинен дати старт перетворенню, потім почекати деякий час до завершення перетворення і, нарешті, зчитати результат. Один з методів керування цим процесом – використання регулярних переривань таймера. З кожним перериванням результат зчитується, і починається нове перетворення.

1.9.4.4 Шина I²C

У двонаправленій шині I²C використовується лише 2 лінії: послідовна лінія синхронізації SCL (SCLock) і послідовна лінія даних SDA (SDAta) [1, 6, 12]. Сигнал на лінії SCL формується процесором для синхронізації даних периферійного пристрою. Обидва виводи (SDA і SCL) виконуються з відкритим колектором або відкритим стоком, що визначається типом мікросхеми: біполярна і КМОП. Вони з'єднані з позитивним джерелом живлення через резистор навантаження за схемою "Монтажне I" і, таким чином, декілька пристроїв можуть одночасно використовувати шини SCL і SDA.

При передачі даних сигнал SDA можна змінювати лише доти, доки на SCL встановлений низький рівень. Коли на SCL високий рівень, перепади на лінії SDA з одного рівня в інший інтерпретуються як умови "СТАРТ" і "СТОП".

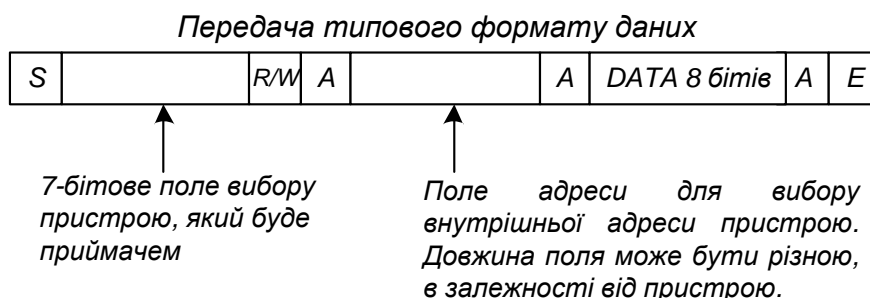
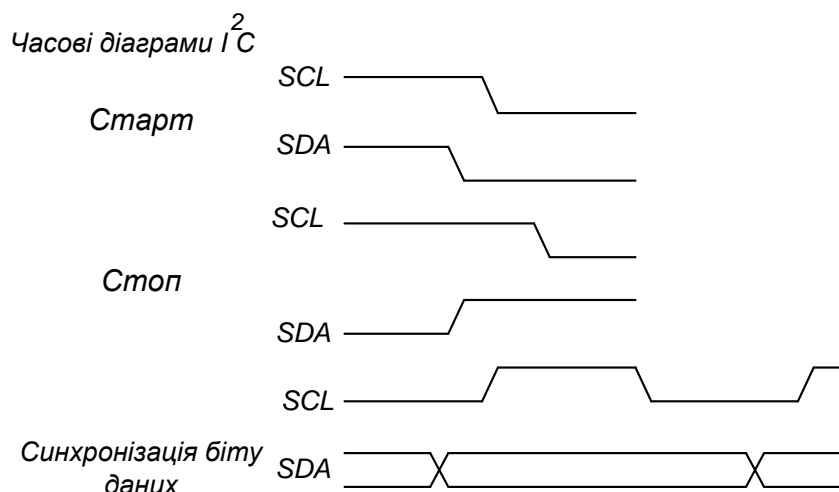


Рисунок 1.68 – Часові діаграми роботи шини I²C

Якщо SDA переходить у низький рівень, тоді як на лінії SCL високий, усі периферійні пристрої на шині сприйматимуть цю подію як старт-умову. Якщо SDA переходить у високий рівень, коли на SCL –

високий, генерується стоп-умова. На рисунку 1.68 показаний приклад типової передачі даних по шині I²C. Процесор генерує старт-умову, а потім посилає одночасно всім периферійним пристроям адресу довжиною 7 бітів, повідомляючи, який з них вибраний, і восьмий біт читання/запису (0 – запис, 1 – читання). 7-бітова адреса дозволяє підключати по I²C-шині 2⁷, або 128, периферійних пристроїв за однієї умови: ємність шини не повинна перевищувати 400 пФ.

Під час передачі в першому байті восьмого біта читання/запису (*R/W*) процесор встановлює напрям передачі: програмує *R/W* на запис (передачу) даних від МП до периферійного пристрою, якщо цей біт дорівнює 0. Інакше МП налаштовується на читання (прийом) даних від периферійного пристрою з адресою, вказаною в 7 бітах. Слід відмітити, що перші 7 бітів передаються, починаючи з старшого біта і закінчуючи молодшим бітом, а восьмим бітом передається біт *R/W*. Після прийому кожного байта, включаючи адресний, вибраний периферійний пристрій (приймальна сторона) по лінії *SDA* посилає сигнал підтвердження на виконання функції прийому переведенням рівня на лінії *SDA* в низький, щоб показати, що він прийняв адресу і умову читання/запису.

Після прийому біта підтвердження процесор встановлює адресу іншого периферійного пристрою, з яким він хоче встановити зв'язок. Довжина цього поля залежить від периферійного пристрою. Потім приймається біт підтвердження і передаються дані. При виконанні операції запису процесор синхронізує вихідні 8 бітів, а при читанні МП встановлює виведення *SDA* як вхід і синхронізує вхідну 8-бітову послідовність. Дані завершуються бітом підтвердження.

Деякі периферійні пристрої дозволяють зчитувати або записувати декілька байтів за одну передачу. Процесор повторює послідовність команд "Дані/біт підтвердження" (*data/acknowledge*) до тих пір, поки всі байти не будуть передані. Периферійний пристрій збільшуватиме свою внутрішню адресу після кожної передачі.

Відомо, що недолік шини I²C – це низька швидкість. Тактова частота шини обмежена на рівні близько 100 кГц. Швидкість прийому/передачі складає 100 Кбіт/с. Проте друга модифікація інтерфейсу I²C, що називається швидкісною (*Fast*), працює зі швидкістю до 400 Кбіт/с. У третій модифікації – високошвидкісній (*High-speed*) – швидкість на шині I²C доходить до 3,4 Мбіт/с. Обидві *Fast* і *High-speed* шини підтримують 10-бітовий адресний простір, таким чином може бути адресовано до 1024 пристроїв. Ці *High-speed*- і *Fast*-пристрої здатні функціонувати і в старих специфікаціях шини I²C, проте старі периферійні пристрої не можна використовувати в системах з підвищеною швидкістю.

Типовий АЦП, що використовує інтерфейс I²C, – це ІС PCF8591

фірми Philips [11, 36]. Ця мікросхема містить і АЦП, і ЦАП. ІС PCF8591 містить три адресні виводи: A0, A1 і A2. Вони можуть бути з'єднані з високим або з низьким рівнем для вибору пристрою, що адресується. Коли декодується адреса периферійного пристрою, ІС PCF8591 відповідатиме на адресу 1001xxx, де xxx узгоджується з рівнями на A2, A1 і A0. Це дозволяє розмістити до восьми мікросхем PCF8591 на одній I²C-шині.

1.9.4.5 Шина 1-Wire

Шина 1-Wire є основою мереж MicroLAN і розроблена наприкінці XX століття фірмою Dallas Semiconductor [8].

Шина 1-Wire побудована за технологією Master/Slave. На шині повинен бути хоча б один ведучий пристрій (Master). Всі інші пристрої повинні бути веденими (Slave). Ведучий пристрій ініціює всі процеси передачі інформації в межах шини. Master може прочитати дані з будь-якого Slave-пристрою або записати їх туди. Передача інформації від одного Slave-пристрою до іншого безпосередньо неможлива. Для того, щоб Master міг звертатися до кожного з ведених пристроїв по шині, кожний ведений пристрій містить у собі індивідуальний код (ID-код).

Протокол 1-Wire містить у собі спеціальну команду пошуку, за допомогою якої ведучий пристрій (Master) може здійснювати автоматичний пошук ведених пристроїв. У процесі пошуку Master визначає ID-коди для всіх підключених до мережі мікросхем. Пошук відбувається шляхом поступового відсіювання неіснуючих адрес. Тому для того, щоб знайти всі пристрої, що підключені до шини, потрібен досить значний час. Середня швидкість пошуку елементів у мережі MicroLAN становить близько 75 вузлів за секунду.

Обмін інформацією по шині 1-Wire відбувається за такими правилами.

1. Обмін завжди ведеться з ініціативи одного ведучого пристрою, що у більшості випадків є мікроконтролером.

2. Будь-який обмін інформацією починається з подачі імпульсу скидання ("Reset Pulse" або просто RESET) у лінію 1-Wire ведучим пристроєм.

3. Для інтерфейсу 1-Wire у загальному випадку передбачається "гаряче" підключення й відключення пристроїв.

4. Будь-який пристрій, підключений до 1-Wire, після одержання живлення видає в лінію DQ імпульс присутності, який називається "Presence pulse". Цей же імпульс пристрій завжди видає в лінію, якщо

виявить сигнал RESET.

5. Поява в шині 1-Wire імпульсу PRESENCE після видачі RESET однозначно інформує про наявність хоча б одного підключеного пристрою.

6. Обмін інформації ведеться так званими тайм-слотами: один тайм-слот служить для обміну одним бітом інформації.

7. Дані передаються побайтово, біт за бітом, починаючи з молодшого біта. Ймовірність переданих/прийнятих даних (перевірка відсутності спотворень) гарантується шляхом підрахунку циклічної контрольної суми.

На рисунку 1.69 показана діаграма сигналів RESET і PRESENCE, з яких завжди починається будь-який обмін даними.

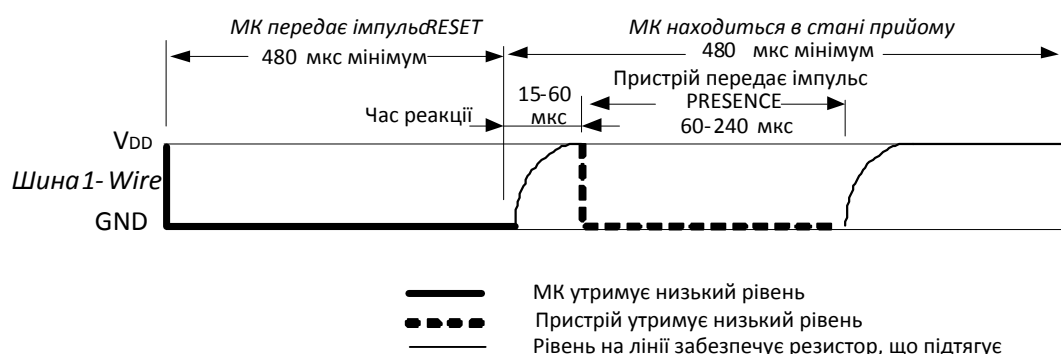


Рисунок 1.69 – Діаграма сигналів RESET і PRESENCE, з яких завжди починається обмін даними

Імпульс RESET формує МК, що переводить в низький логічний рівень шину 1-Wire і втримує її в цьому стані мінімум 480 мікросекунд. Далі МК повинний "відпустити" шину. Через якийсь час, що залежить від ємності лінії й опору резистора, що підтягує, у лінії встановиться високий логічний рівень. Протокол 1-Wire обмежує цей час "релаксації" діапазоном від 15 до 60 мікросекунд, що і є визначальним для вибору резистора, що підтягує.

Виявивши імпульс RESET, ведений пристрій (Slave) приводить свої внутрішні вузли у вихідний стан і формує відповідний імпульс PRESENCE не пізніше 60 мікросекунд після завершення імпульсу RESET. Для цього пристрій переводить у низький рівень лінію і втримує її в цьому стані від 60 до 240 мікросекунд. Після цього пристрій так само "відпускає" шину.

Але після завершення імпульсу PRESENCE пристрою дається ще якийсь час для завершення внутрішніх процедур ініціалізації, таким чином, МК повинен приступити до будь-якого обміну з пристроєм не раніше, ніж через 480 мікросекунд після завершення імпульсу RESET.

Отже, процедура ініціалізації інтерфейсу, з якої починається будь-який обмін даними між пристроями, триває мінімум 960 мікросекунд, складається з передачі від МК сигналу RESET і прийому від пристрою сигналу PRESENCE. Якщо сигнал PRESENCE не виявлений – значить на шині 1-Wire немає готових до обміну пристроїв.

Обмін бітами інформації здійснюється певними тайм-слотами. Тайм-слот – це певна, досить жорстко лімітована за часом послідовність зміни рівнів сигналу в лінії 1-Wire. Розрізняють 4 типи тайм-слотів: передача "1" від МК, передача "0" від МК, прийом "1" від Slave-пристрою, прийом "0" від Slave-пристрою.

Будь-який тайм-слот завжди починає МК шляхом переведення шини 1-Wire у низький логічний рівень. Тривалість будь-якого тайм-слоту повинна перебувати в межах від 60 до 120 мікросекунд. Між окремими тайм-слотами завжди повинен передбачатися інтервал не менший 1 мікросекунди.

Тайм-слоти передачі відрізняються від тайм-слотів прийому поведінням МК: при передачі він тільки формує сигнали; при прийманні, крім того, ще й опитує рівень сигналу в лінії 1-Wire. Рисунок 1.70 демонструє часові діаграми тайм-слотів всіх 4-х типів: угорі показані тайм-слоти передачі від МК, унизу – прийому від Slave-пристрою.

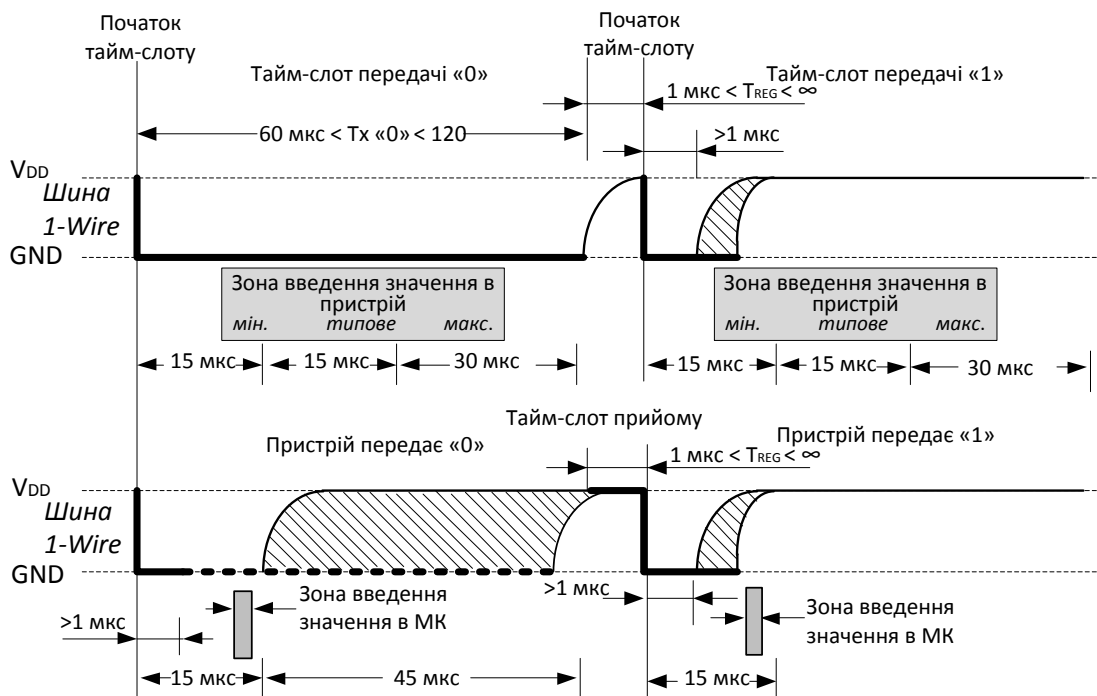


Рисунок 1.70 – Часові діаграми тайм-слотів на лінії 1-Wire

Тайм-слот передачі "0" полягає просто в утриманні шини 1-Wire

у низькому рівні протягом всієї тривалості тайм-слоту. Передача "1" здійснюється шляхом "відпускання" шини 1-Wire з боку МК не раніше 1 мікросекунди після початку тайм-слоту, але не пізніше 15 мікросекунд. Ведений пристрій опитує рівень у шині 1-Wire протягом часового інтервалу, умовно показаного у вигляді сірого прямокутника, тобто починаючи з 15-ї мікросекунди від початку тайм-слоту й закінчуючи 60-ю мікросекундою від початку. Типовий момент уведення рівня в пристрій – біля 30-ї мікросекунди від початку тайм-слоту.

Заштрихована область – це область "наростання" рівня в шині 1-Wire, що залежить від ємності лінії й опору резистора, що підтягує. Тайм-слоти прийому інформації відрізняються тим, що МК формує тільки початок тайм-слоту (абсолютно так само, як при передачі "1"), а потім керування рівнем шини 1-Wire бере на себе пристрій, а МК здійснює введення цього рівня так само в певній зоні часових інтервалів. Заштрихована область – область невизначеності, тому для введення мікроконтролеру залишається навіть не проміжок, а скоріше конкретний момент, коли він повинен увести рівень сигналу з лінії. Цей момент часу – 14-а або 15-а мікросекунда від початку тайм-слоту.

Таким чином МК починає тайм-слот з видачі в шину 1-Wire "0" протягом 1 мікросекунди. Наступний рівень залежить від типу тайм-слоту: для прийому й передачі "1" рівень повинен стати високим, а для передачі "0" – залишатися низьким аж до кінця тайм-слоту, тобто не менш 60 і не більше 120 мікросекунд. Якщо МК приймає дані, то опитування рівня в шині він повинен зробити на проміжку від 13-ї до 15-ї мікросекунди тайм-слоту. МК повинен забезпечити інтервал між окремими тайм-слотами не менше 1 мікросекунди (краще – більше, максимальне значення не обмежене).

Кожний пристрій 1-Wire має унікальний ідентифікаційний 64-бітовий номер, який програмується на етапі виробництва мікросхеми.

Мережний рівень. На цьому рівні відбувається адресація елементів на однопроводовій шині. Перебуваючи на цьому рівні, всі Slave пристрої очікують одну з команд адресації. Така команда повинна вибрати один або кілька пристроїв, з якими Master буде працювати на транспортному рівні. Всі команди мережного рівня наведені в таблиці 1.5

Таблиця 1.5 – Команди мережного рівня

Команда	Значення байта	Опис
SEARCH ROM	0×00	Пошук адреси – використовується при універсальному алгоритмі визначення кількості й адрес підключених пристроїв

Продовження таблиці 1.5

Команда	Значення байта	Опис
READ ROM	0×33	Читання адреси пристрою – використовується для визначення адреси єдиного пристрою на шині
MATCH ROM	0×55	Вибір адреси – використовується для звертання до конкретної адреси пристрою з багатьох підключених
SKI ROM	0×CC	Ігнорувати адресу – використовується для звертання до єдиного пристрою на шині, при цьому адреса пристрою ігнорується (можна звертатися до невідомого пристрою)

Після того, як МК видасть команду READ ROM, від пристрою надійде 8 байтів його власної унікальної адреси – МК повинен їх прийняти. Будь-яка процедура обміну даними з пристроєм повинна бути завершена повністю або перервана посиленням сигналу RESET.

Якщо відправлено команду MATCH ROM, то після неї МК повинен передати так само й 8 байтів конкретної адреси пристрою, з яким буде здійснюватися наступний обмін даними. За цією командою кожний пристрій порівнює передану адресу зі своєю власною. Всі пристрої, адреси яких не збіглися, припиняють аналіз і видачу сигналів у лінії 1-Wire, а пристрій, що пізнав адресу, продовжує роботу. Тепер всі дані, що передані МК, будуть потрапляти лише до цього пристрою. Які саме дані треба послати в пристрій або одержати від нього після його адресації, залежить від конкретного пристрою.

Якщо пристрій один на шині, то можна прискорити процес взаємодії з ним за допомогою команди SKI ROM. Отримавши цю команду, пристрій відразу вважає, що адреса збіглася, хоча ніякої адреси для цієї команди не треба. Деякі процедури не вимагають прийому від пристрою ніяких даних, у цьому випадку команду SKI ROM можна використовувати для передачі деякої інформації відразу всім пристроям.

Транспортний рівень. Всі мікросхеми мережі MicroLAN переходять на цей рівень після будь-якої команди мережного рівня. Набір команд транспортного рівня для різних мікросхем дещо відрізняється. Приклади команд транспортного рівня для різних типів мікросхем наведені в таблиці 1.6. Основні команди транспортного рівня – це команди "Запис пам'яті" і "Читання пам'яті". Як приклад команди транспортного рівня розглянемо команду "Читання пам'яті".

Таблиця 1.6 – Приклади команд транспортного рівня

Мікросхема	Код	Команда
DS1994 (4096-бітова перезаписовна енергонезалежна пам'ять)	0FH	Запис в блокнотну пам'ять
	CAAH	Читання блокнотної пам'яті
	55H	Копіювання блокнотної пам'яті
	0F0H	Читання основної пам'яті
DS2417 Годинник реального часу	66H	Читання часу
	99H	Запис часу
DS2450 4-канальний АЦП	0AAH	Читання з пам'яті результатів перетворення напруги
	55H	Запис до пам'яті команд керування
	3CH	Запуск процесу перетворення напруги в код
DS18S20/DS18B20/DS1822 Електронні датчики температури	44H	Запуск процесу конвертування температури в код
	4EH	Запис у блокнотну пам'ять (меж регулювання температури)
	0BEH	Читання блокнотної пам'яті
	48H	Копіювання блокнотної пам'яті в спеціальні осередки внутрішньої EEPROM
	0B8H	Читання інформації з EEPROM у блокнотну пам'ять
	0B4H	Читання інформації про режим живлення

На рисунку 1.71 наведена часова діаграма виконання команди "Читання пам'яті". Для адресації мікросхеми в наведеному прикладі використана команда мережного рівня – "Збіг ПЗП". Як видно з рисунка, виконання команди починається з імпульсу скидання. Далі всі ведені пристрої виробляють сигнал присутності на лінії. Витримавши паузу Master-пристрій видає на лінію команду "Збіг ПЗП". Команда складається з восьми бітів коду операції й 64-бітового ID-коду. Одержавши цю команду, всі ведені мікросхеми переходять на транспортний рівень. Але тільки одна з них залишиться активною. Інші перейдуть у пасивний режим очікування.

Далі Master-пристрій передає на шину команду "Читання пам'яті" (8 бітів). Одержавши цю команду, обрана мікросхема переходить у режим видачі даних. Master-пристрій читає ці дані з веденої мікросхеми. Обсяг даних, що зчитуються, залежить від типу мікросхеми. Найчастіше він дорівнює 8 байтам (64 біти) і обов'язково містить контрольну суму в останніх вісьми бітах.

Після того, як всі дані прийняті, виконання команди можна вважати закінченим. Для того, щоб виконати наступну команду, потрібно все починати з початку. Тобто, з імпульсу початкового скидання.

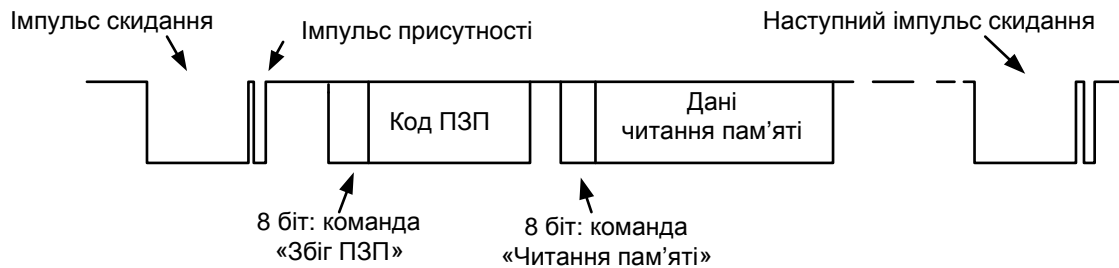


Рисунок 1.71 – Часова діаграма виконання команди "Читання пам'яті"

Структура індивідуального коду, записаного в спеціальний ПЗП будь-якої мікросхеми, що підтримує інтерфейс 1-Wire, умовно зображена на рисунку 1.72. Код складається з трьох основних елементів:

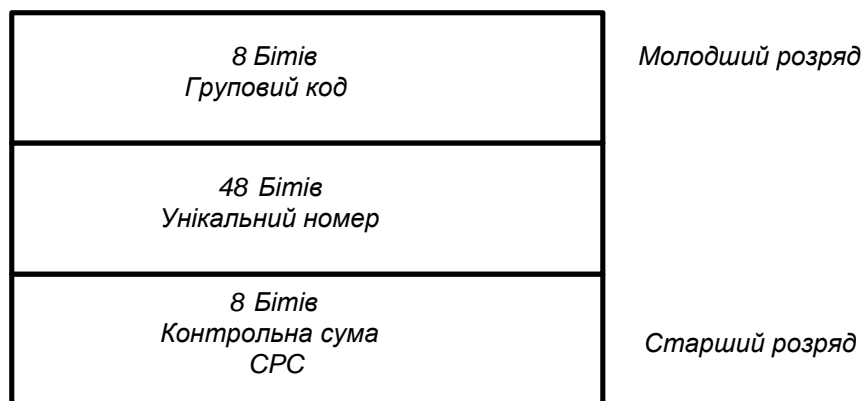


Рисунок 1.72 – Структура індивідуального коду мікросхеми 1 -Wire

Перші 8 бітів, починаючи з молодшого розряду, – це груповий код мікросхеми. Груповий код однозначно визначає її вид. Приклади групового коду для різних видів мікросхем наведені в таблиці 1.7. Наступні 48 бітів – це унікальний серійний номер мікросхеми. Останні 8 бітів – це контрольна сума (CRC).

Таблиця 1.7 – Групові коди деяких видів мікросхем

Вид мікросхеми	Опис	Груповий код
DS18S20	Електронний термометр	10Н
DS18B20	Електронний термометр	28Н
DS18B22	Електронний термометр	22Н
DS1992	Незалежна пам'ять	08Н
DS1993	Незалежна пам'ять	0ЕН
DS1994	Незалежна пам'ять	04Н
DS2401	Кремнієвий серійний номер	01Н
DS2417	Таймер реального часу	27Н
DS2450	4-канальний АЦП	20Н
DS2890	Електронний змінний резистор	2СН

Контрольна сума або CRC – це байт, значення якого передається найостаннішим і обчислюється за спеціальним алгоритмом на основі значення усіх семи попередніх байтів. Алгоритм підрахунку такий, що якщо усі байти, що передаються-приймаються, без спотворень (а спотворення цілком можливі, якщо згадати характер апаратної реалізації інтерфейсу), то прийнятий байт контрольної суми обов'язково збіжиться з розрахованим в МК (чи пристрої) значенням. Тобто при реалізації програмного алгоритму обміну інформацією необхідно при передачі і прийомі байтів підраховувати їх контрольну суму за певним алгоритмом, а потім або передати отримане значення, або порівняти розрахункове значення з отриманим значенням CRC. Тільки при збігу обох CRC МК або пристрій вважають прийняті дані достовірними. Інакше продовження обміну неможливе.

Очевидно, що алгоритм підрахунку CRC має бути однаковим як для МК, так і для будь-якого пристрою. Він "стандартизований" і описаний в документації [8].

DS2450 – чотириканальний 16-розрядний аналого-цифровий перетворювач послідовного наближення. Він містить на вході високоімпедансний аналоговий комутатор, що підключає по черзі, відповідно до команд, що надходять по однопроводовому інтерфейсу, вхідні сигнали від чотирьох аналогових датчиків на вхід АЦП. Рівень вхідних напруг сигналів може бути встановлений програмно для двох діапазонів: 2,56 В або 5,12 В. Роздільність приладу також програмується від 1 до 16 розрядів. Швидкість перетворення становить близько 80 мкс/біт. Максимальна швидкість передачі даних 16,3 Кбіт/с.

Як і всі прилади MicroLAN, DS2450 містить ПЗП з унікальним серійним номером, груповим кодом і контрольною сумою.

"Паразитного" живлення вистачає тільки на роботу в режимі зв'язку з мікроконтролером. Для аналого-цифрового перетворення необхідне підключення шини до додаткового джерела напруги 5 В або подача живлення на вивід V_{cc} (рисунок 1.73).

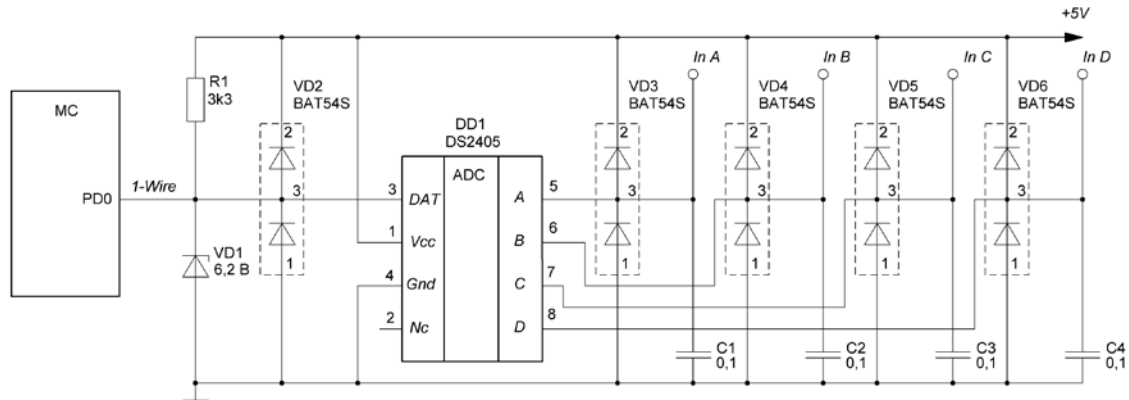


Рисунок 1.73 – Схема підключення DS2450 по шині 1-Wire

Таблиця 1.8 – Сторінка читання-перетворення (сторінка 0)

Адреса	Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0
00	A	A	A	A	A	A	A	молодший біт
01	старший біт	A	A	A	A	A	A	A
02	B	B	B	B	B	B	B	молодший біт
03	старший біт	B	B	B	B	B	B	B
04	C	C	C	C	C	C	C	молодший біт
05	старший біт	C	C	C	C	C	C	C
06	D	D	D	D	D	D	D	молодший біт
07	старший біт	D	D	D	D	D	D	D

Всі регістри DS2450 розташовані в 24 сусідніх байтах і організовані у вигляді трьох сторінок по 8 байтів. Перша сторінка (сторінка 0), названа сторінкою читання-перетворення, містить область пам'яті, де розташовуються результати аналого-цифрового перетворення для наступного читання їхнім мікроконтролером. Починаючи з молодшої адреси порту A, кожному порту відведено 16 бітів для зберігання результату перетворення (таблиця 1.8). При подачі живлення в усі регістри читання-перетворення за замовчуванням записуються нулі. Старший значущий біт результату

перетворення завжди перебуває на фіксованому місці, незалежно від необхідної розрядності. При розрядності менше 16 бітів для формування 16 бітового результату молодші значущі біти будуть доповнюватися нулями. Якщо в конкретному випадку використовуються не всі чотири порти, то для першого порту буде застосований вхід D, для другого – C й так далі. При цьому мінімізується завантаження однопроводової шини й скорочується час перевірки правильності передачі даних.

Керування портами й інформація про стан цих портів розташована на сторінці 1 (табл. 1.9). Як і у випадку сторінки читання-перетворення, на кожний порт відведено по 16 бітів. Чотири молодших біти RC3-RC0 – двійкове подання числа перетворених бітів. Код 1111_2 формує 15-бітовий результат. У випадку повного 16 бітового перетворення повинен бути код 0000_2 .

Таблиця 1.9 – Сторінка керування й стану портів (сторінка 1)

Адреса	Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0
08	0E-A	0C-A	0	0	RC3-A	RC2-A	RC1-A	RC0-A
09	POR	0	AFH-A	AFL-A	AEH-A	AEL-A	0	IR-A
0A	0E-B	0C-B	0	0	RC3-B	RC2-B	RC1-B	RC0-B
0B	POR	0	AFH-B	AFL-B	AEH-B	AEL-B	0	IR-B
0C	0E-C	0C-C	0	0	RC3-A	RC2-A	RC1-A	RC0-A
0D	POR	0	AFH-C	AFL-C	AEH-C	AEL-C	0	IR-C
0E	0E-D	0C-D	0	0	RC-3	RC2-D	RC1-D	RC0-D
0F	POR	0	AFH-D	AFL-D	AEH-D	AEL-D	0	IR-D

Біти 0C (керування виходом) і 0E (дозвіл виходу) керують роботою порту на вихід. При роботі порту як аналоговий вхід (прийом даних) біт 0E повинен бути встановлений в "0", при цьому стан біта 0C ігнорується. Якщо біт 0E встановлений в "1", то при встановленні 0C в "0" вихідний транзистор даного порту буде відкритий, а при 0C=1 – замкнений. Наприклад, при підключенні резистора навантаження на "плюс" живлення, біт 0C буде безпосередньо перетворюватися в напругу, що еквівалентна його логічному стану.

За допомогою біта IR у другому байті керування й стану вибирається діапазон вхідних напруг. Установлення IR в "0" дозволяє одержати максимально можливий результат перетворення при напрузі 2,2 В. При IR = 1 одержання такого ж результату можливо при вхідній напрузі 5,1 В. Бітами AEL і AEH установлюється дозвіл відповіді приладу на команду "Умовного пошуку" у випадку, коли результати

перетворення вищі (AEN) або нижчі (AEL) граничних значень. Прапорці AFL і AFH указують мікроконтролеру на перевищення встановлених порогів при останньому перетворенні. Прапорці скидаються автоматично, якщо результат нового перетворення не виходить за встановлені межі.

Біт POR (скидання після включення живлення) автоматично встановлюється в "1" на весь час виконання циклу скидання установок приладу при включенні живлення. Поки цей біт встановлений, прилад буде відповідати на команду "Умовного пошуку" для того, щоб повідомити мікроконтролер про те, що дані керування й пороги не дійсні. Після завершення циклу скидання біт POR скидається мікроконтролером. Це може виконуватися одночасно з встановленням даних керування й порогів, при цьому мікроконтролер записує в POR "1". У цьому випадку прилад відповідає на команду "Умовного пошуку", але не генерує скидання.

Регістри граничних напруг кожного порту розташовані на сторінці пам'яті 2, починаючи з молодшого біта нижнього порога. При включенні живлення за замовчуванням встановлюються: нижній поріг – 00h, верхній поріг – FFh. Розмір порогів завжди становить 8 бітів. Коли дозвіл більше або дорівнює 8 бітів, то прапорець перевищення порога буде встановлений, якщо 8 старших значущих біт результату перетворення дають число, більше вмісту регістра AFH або менше вмісту регістра AFL. При дозволі менше 8 бітів молодші значущі біти регістрів граничних напруг ігноруються (сторінка 2 (табл. 1.10)).

Таблиця 1.10 – Сторінка граничних значень (сторінка 2)

Адреса	Біт 7	Біт 6	Біт 5	Біт 4	Біт 3	Біт 2	Біт 1	Біт 0
10	СБМБ	А	А	А	А	А	А	МБМБ
11	СБСБ	А	А	А	А	А	А	МБСБ
12	СБМБ	В	В	В	В	В	В	МБМБ
13	СБСБ	В	В	В	В	В	В	МБСБ
14	СБМБ	С	С	С	С	С	С	МБМБ
15	СБСБ	С	С	С	С	С	С	МБСБ
16	СБМБ	Д	Д	Д	Д	Д	Д	МБМБ
17	СБСБ	Д	Д	Д	Д	Д	Д	МБСБ

СБМБ – старший біт молодшого байта; СБСБ – старший біт старшого байта; МБМБ – молодший біт молодшого байта; МБСБ – молодший біт старшого байта

Аналого-цифрове перетворення для одного або декількох портів виконується за командою "Перетворення" (3Ch). Перетворення займає

від 60 до 80 мкс на біт плюс час виходу (160 мкс) після завершення команди перетворення. Наприклад, для чотирьох портів з роздільністю 12 бітів час виконання команди перетворення складе $4 \times 12 \times 80 \text{ мкс} + 160 \text{ мкс} = 4 \text{ мс}$. Якщо DS2450 підключений до зовнішнього джерела живлення через вивід V_{cc} , то мікроконтролер в цей час може працювати з іншими приладами мережі. У випадку живлення приладу від однопроводової шини мікроконтролер не може зв'язуватися з іншими приладами, тому що він повинен забезпечити живлення DS2450 на весь час перетворення.

1.9.4.6 Спеціалізовані послідовні інтерфейси

Деякі АЦП оснащені своїм власним інтерфейсом. Типовим прикладом є ІС АЦП MAX1101 фірми MAXIM/Dallas [11]. Це 8-бітовий АЦП, що призначений для роботи з приладами із зарядовим зв'язком (ПЗЗ). АЦП MAX1101 для забезпечення інтерфейсу використовує 4 виводи: MODE ("режим роботи"), LOAD ("навантаження"), DATA ("дані") і SCLK ("такти"). Сигнал на виводі MODE визначає, записуються чи зчитуються дані (1 – читання, 0 – запис). Вивід DATA – двонаправлений, сигнал SCLK синхронізує дані, що зчитуються або записуються в пристрій, а вивід LOAD використовується після процедури запису для синхронізації запису даних у внутрішні регістри. Синхронний послідовний інтерфейс ІС MAX1101 подібний до SPI, але, оскільки відсутній сигнал "вибір кристала" (chip select), на шині не може бути використано багато периферійних пристроїв. Кожен АЦП MAX1101 потребує чотирьох ліній зв'язку з МП.

Проблема використання будь-якого послідовного інтерфейсу для АЦП в тому, що послідовний інтерфейс обмежує швидкість перетворення. Окремі види послідовних інтерфейсів вносять додаткові часові обмеження. Оскільки прилади обмінюються даними по шині І²С розрядністю слова, принаймні, 20 бітів, 1-Wire, І²С-пристрій не в змозі досягти швидкості, подібний SPI-пристрою. З цієї причини випускається набагато більше АЦП з інтерфейсом SPI, ніж І²С.

Необхідна пропускна здатність послідовного інтерфейсу диктує умови вибору інтерфейсу. Якщо необхідно досягти швидкості перетворення 100000 вибірок/с по 8 бітів при використанні програмно сформованого SPI-інтерфейсу, тоді слід врахувати, що МК не зможе витратити більш ніж $1/(100000 \times 8)$, або 1,25 мкс, на кожен біт. Може виявитися неможливим виконання процесором яких-небудь інших операцій, після чого перевага може буде надана АЦП з паралельним інтерфейсом або МК з апаратною підтримкою SPI.

РОЗДІЛ 2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МІКРОКОНТРОЛЕРІВ

2.1 Середовища розробки програмного забезпечення для мікроконтролерів AVR

Процес написання програм для МК AVR, як і для будь-яких інших МК, складається з декількох етапів:

- підготовка вихідного тексту програми на будь-якою мовою програмування;
- компіляція програми;
- налагодження й тестування програми;
- остаточне програмування й підготовка до серійного виробництва.

Мікропрограма пристрою повинна бути написана однією з мов програмування.

На даний час для МК AVR існує декілька мов програмування, а також різних засобів підтримки розробки, що використовують одну мову, але різняться за функціональністю.

На кожному з етапів необхідне застосування спеціальних програмних й апаратних засобів. Варто відзначити, що базовий набір програмного забезпечення (компілятор асемблера, ПЗ для програмування) поширюється фірмою Atmel безкоштовно. Однак за досить довгий період часу, що пройшов з моменту появи цих МК, з'явилася велика кількість програмного забезпечення сторонніх виробників.

2.1.1 Компілятори мови Assembler

Вибір компілятора асемблера є найменш принциповим питанням, оскільки сам процес компіляції (перетворення вихідного тексту програми в машинний код) виконується досить однозначно. Власне, всі відмінності між асемблерами полягають у можливостях процесора, що обробляє макрокоманди, наявності текстового редактора або середовища розробки й типу операційної системи, що підтримується.

AVR Studio

Досить вдалим вибором при розробці програмного забезпечення для МК сімейства AVR може служити інтегроване середовище розробки AVR Studio фірми Atmel [3, 9, 19, 20, 24, 58, 68], що містить

у собі текстовий редактор з підсвічуванням синтаксису, компілятор асемблера, симулятор, налагоджувач й інтерфейс із апаратними емуляторами. Програма розрахована на роботу під керуванням операційних систем Windows 9x, 2000, XP, Vista. До недоліків AVR Studio можна віднести деяку нестабільність роботи налагоджувача, а також неповну симуляцію периферійних пристроїв (зокрема, відсутня симуляція АЦП). До плюсів належить, у першу чергу, підтримка практично всіх МК сімейства AVR.

IAR AVR Assembler

Atmel безкоштовно розповсюджує асемблер фірми IAR. Даний компілятор є частиною IAR Embedded Workbench [19, 52] – середовища розробки, що містить менеджер проектів, редактор, лінкер і менеджер бібліотек. Цей компілятор відрізняється розширеними можливостями роботи з макросами.

GNU/Linux AVR Assembler

Цей асемблер, що повністю сумісним зі компілятором фірми Atmel, має в порівнянні з ним кілька переваг. По-перше, він поставляється з відкритими текстами, так що користувач, якщо буде потреба, може легко додати нові можливості. По-друге, це один з декількох асемблерів для МК фірми AVR, що працюють під керуванням операційної системи Linux (крім того, наявність вихідних текстів дозволяє легко перенести його на будь-яку іншу Unix-систему). У GNU AVR Assembler значно розширені, у порівнянні з асемблером фірми Atmel, можливості роботи з макрокомандами (зокрема, допускаються макроси всередині макросів). Програма поширюється за умовами GNU Public License (GPL) [3, 17, 45].

2.1.2 Компілятори мови C

Останнім часом усе популярніше стає використання компіляторів мов високого рівня при написанні програм для МПС. Найбільше поширення при цьому одержали компілятори мови C, оскільки в цій мові найбільш просто реалізуються всі необхідні можливості з керування апаратними засобами МК.

IAR C Compiler

Компілятор фірми IAR, як загально визнано, є одним із кращих компіляторів C для МК сімейства AVR. Пов'язано це з наявністю в

ньому можливостей оптимізації коду. Істотним його недоліком є те, що у демонстраційній версії накладаються значні обмеження на максимальний обсяг коду. Компілятор поставляється у складі інтегрованого середовища розробки IAR Embedded Workbench (EWB), що містить також компілятор асемблера, лінкер, менеджер проектів і бібліотек, а також налагоджувач. Він може працювати сумісно з AVR Studio.

Image Craft C Compiler

Image Craft C Compiler [19, 43, 47, 58, 68] є другим за популярністю компілятором мови C для МК сімейства AVR. До переваг цього компілятора можна віднести можливість підтримки мікросхем сімейства FPSLIC фірми Atmel, непоганий рівень оптимізації коду й досить низьку ціну. Демо-версія компілятора не містить функціональних обмежень і лімітована тільки часом роботи (30 днів).

CodeVision AVR C Compiler

Іншим популярним компілятором мови C для МК сімейства AVR є Code Vision AVR C Compiler [47, 68]. Компілятор підтримує мікроконтролери ATTiny22, AT90S2313, AT90S4433 та багато інших. Демонстраційна версія компілятора так само, як і компілятор фірми IAR, має обмеження на максимальний обсяг коду програми. Компілятор поставляється разом з інтегрованим середовищем розробки, у яку, крім стандартних можливостей, внесена досить цікава функція – CodeWizardAVR Automatic Program.

Фактично це генератор стандартних блоків програми, що виконує такі функції:

- налагоджування доступу до зовнішньої пам'яті;
- ініціалізація портів введення/виведення, зовнішніх переривань, таймерів, сторожового таймера;
- ініціалізація асинхронного порту послідовної передачі даних (UART) і прийом, передача даних через UART;
- настроювання аналогового компаратора й АЦП;
- ініціалізація шини I²C і підключених до неї температурного датчика LM75 або годинника реального часу PCF8583, DS1302, DS1307;
- реалізація протоколу 1-Wire Bus й ініціалізація температурного датчика DS1820;
- керування LCD-індикатором.

Наявність цієї можливості спрощує написання програм. Наявність у середовищі розробки послідовного терміналу дозволяє робити налагодження програм з використанням послідовного порту UART МК.

2.1.3 Компілятори Basic та Pascal

Крім C при розробці програмного забезпечення для МПС застосовуються й інші мови високого рівня. Так, для МК сімейства AVR існують також компілятори мов Basic, Pascal і Forth [19, 47, 52].

BASCOM-AVR

Компілятор мови Basic розроблений фірмою MCS Electronics. Є демо-версія з обмеженням на обсяг коду програми. За вхідним кодом компілятор практично повністю сумісний з компіляторами Visual Basic/Quick Basic фірми Microsoft. У синтаксис мови додано кілька нових команд для забезпечення підтримки LCD-індикаторів, I²C й 1-Wire інтерфейсів, обробки переривань й інших специфічних для МК можливостей.

ABC Basic Compiler

Розроблений фірмою Investments Technologies PTY, дозволяє писати й налагоджувати програми мовою Basic в інтегрованому середовищі розробки. Існують версії компілятора як під Windows, так і під DOS.

Microbasic і Mikropascal

Microbasic і Mikropascal є потужними середовищами програмування для МК AVR.

Дані середовища створені для полегшення роботи користувача і поліпшення можливостей роботи з МК сімейства AVR.

Вони дозволяють:

- писати код Basic та Pascal за допомогою редактора коду;
- використовувати бібліотеки для прискорення роботи з даними, пам'яттю, індикаторами, периферійними пристроями;
- бачити структуру програми, змінні та функції у редакторі коду, генерувати прокоментовану, зрозумілу людині програму і сумісну з HEX-файлами для будь-яких програм.

2.2 Програмування мовою асемблера в середовищі AVR Studio

2.2.1 Створення програм у AVR Studio 3.0

Для створення нового проекту необхідно вибрати у головному меню **Project (Проект)|New... (Новий)**. У вікні, що з'явилося, в рядку **Project name (Ім'я проекту)** треба ввести ім'я нового проекту, наприклад "new". Далі, натиснувши лівою кнопкою мишки на рядку **AVR Assembler**, вибрати тип проекту (рисунок 2.1). Натиснути кнопку **OK**.

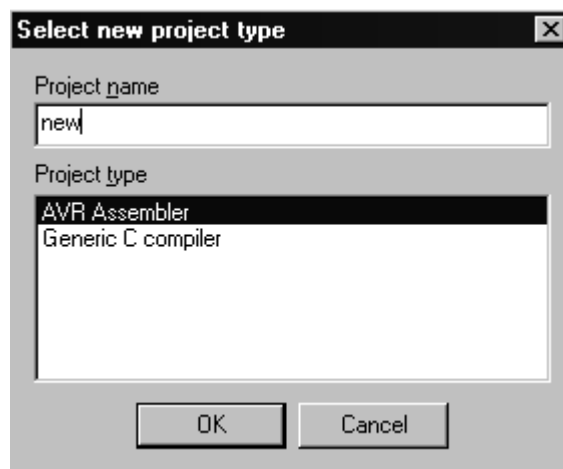


Рисунок 2.1 – Вікно вибору типу проекту

У вікні, що утворилось, необхідно натиснути правою кнопкою мишки на рядку **Assembler Files**; у меню, що з'явилося (рисунок 2.2) натиснути лівою кнопкою мишки на рядку **Add File...** (Додати файл).

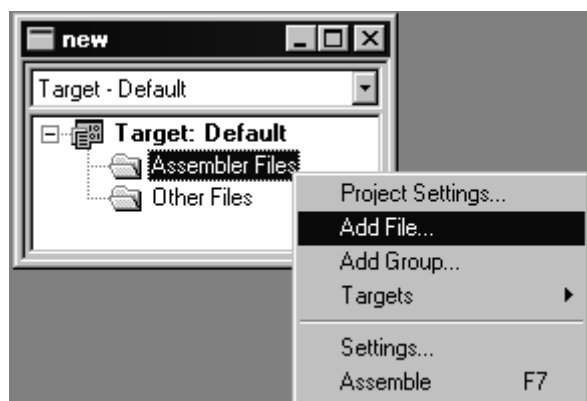


Рисунок 2.2 – Список файлів проекту

У вікні **Open file** необхідно створити порожній текстовий файл, змінити ім'я файлу майбутньої програми та надати йому розширення **.asm**. Таким чином буде створений порожній файл робочої програми з

розширенням **.asm**. Якщо у вас вже існує файл програми (з розширенням **.asm**), то у вікні **Open file** треба вибрати цей файл.

Для того, щоб зробити створений файл основним, необхідно натиснути правою кнопкою мишки на імені створеного файлу в вікні проекту, а потім у меню, що з'явилося, натиснути лівою кнопкою мишки на рядку **Assembler entry file** (рис. 2.3). Тепер необхідно відкрити створений файл, двічі натиснувши на ньому лівою кнопкою мишки у вікні проекту. У вікні, що з'явилося, можна набирати текст програми.

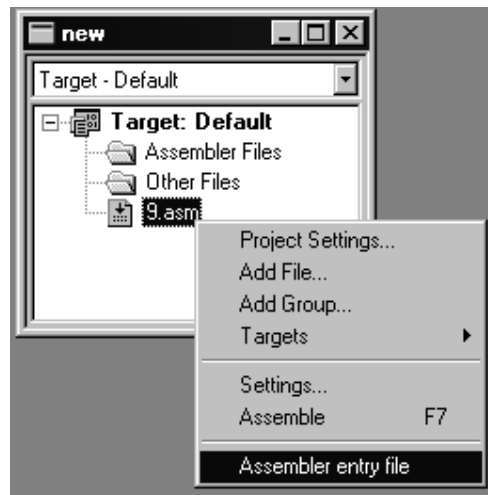


Рисунок 2.3 – Вибір основного файлу

Після введення тексту програми необхідно натиснути клавішу **F7**, щоб провести компіляцію програми. Після цього з'являється вікно (рис. 2.4), в якому міститься інформація про правильність написаної програми. При виявленні помилок повідомляється про всі рядки, в яких виявлено помилки. Натиснувши на рядок, в якому повідомляється про помилку, потрапимо в те місце програми, де виявлено помилку.

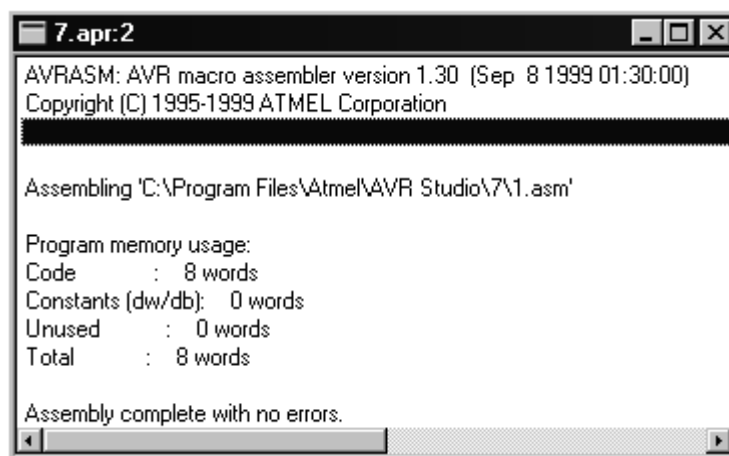


Рисунок 2.4 – Вікно результатів компіляції коду програми

Після завершення компіляції без помилок, тобто в вікні (див. рис. 2.4) нижній рядок буде мати вигляд **Assembly complete with no errors** (програма відкомпільована без помилок), можна приступити до відлагодження роботи програми.

Після натискання клавіші **F11** почнеться процес відлагодження. Для перегляду результатів роботи програми з меню **View** (Вид) треба викликати вікна **Registers** (Регістри), **Processor** (Процесор), **New Memory View** (Новий вид пам'яті). Вікно **Registers** відображає поточний зміст регістрів загального призначення **R0-R31**. Вікно **Processor** відображає поточне значення регістрових пар (X, Y, Z), значення програмного лічильника, значення покажчика стека та всіх восьми прапорців регістра стану.

Вікно **New Memory View**, залежно від обраного режиму, відображає поточний зміст пам'яті даних, пам'яті програм, пам'яті введення-виведення, пам'яті EEPROM. Робоче вікно програми набуде вигляду, показаного на рисунку 2.5.

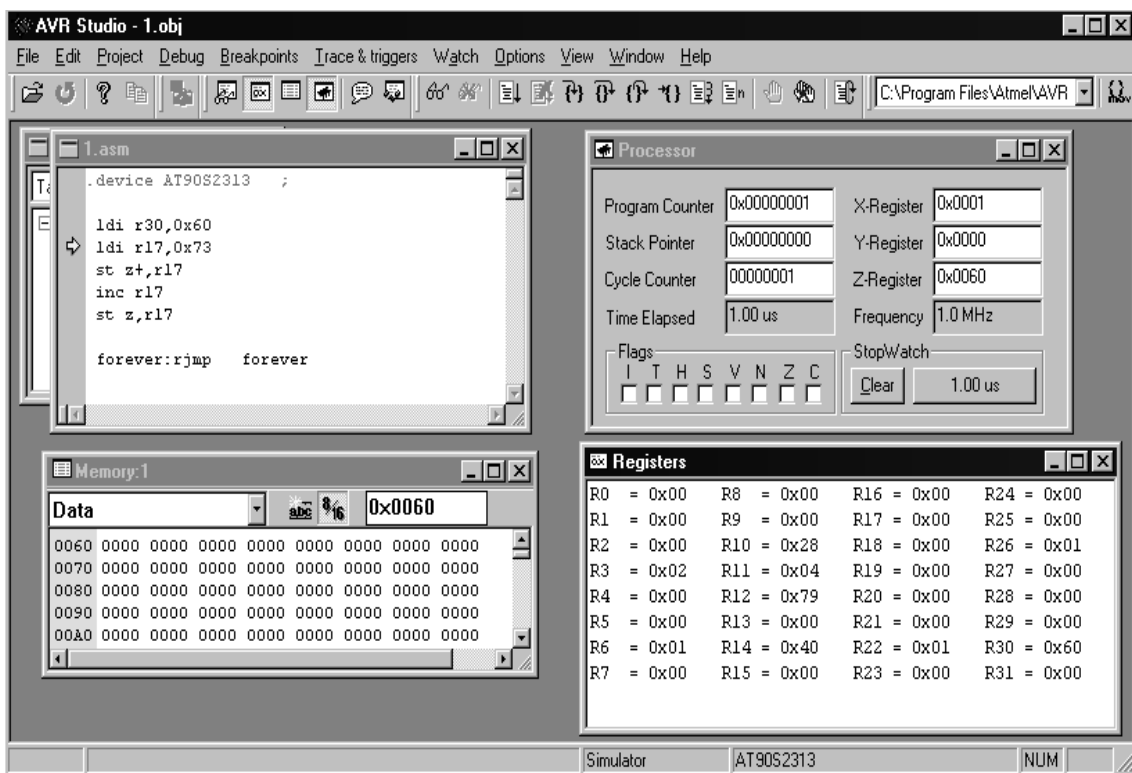


Рисунок 2.5 – Робоче вікно програми

При натисканні клавіші **F11 (Trace into)** відбувається виконання поточних команд, що відображається відповідними змінами у вікнах.

Компілятор працює з вихідними файлами, що містять інструкції, мітки і директиви. Інструкції та директиви, як правило, мають один чи декілька операндів.

Примітка. Рядок коду не повинен бути довшим 120 символів. Будь-який рядок може починатися з мітки, що є набором символів та закінчується двокрапкою. Мітки використовуються для вказання місця, у яке передається керування при переходах, а також для задання імен змінних.

Вхідний рядок може мати одну з чотирьох форм:

- [мітка:] директива [операнди] [Коментар];
- [мітка:] інструкція [операнди] [Коментар];
- коментар;
- порожній рядок.

Коментар починається після крапки з комою (;) і до кінця рядка ігнорується компілятором.

2.2.2 Робота з графічним інтерфейсом користувача AVR STUDIO 4

Після завантаження AVR STUDIO 4 з'явиться діалогове вікно. Для створення нового проекту треба натиснути [**Create New Project**] (Створити новий проект).

Другим етапом є встановлення налаштувань проекту. На даному етапі задаються розширення, назва й адреса файлу, що створюється (рис. 2.6).

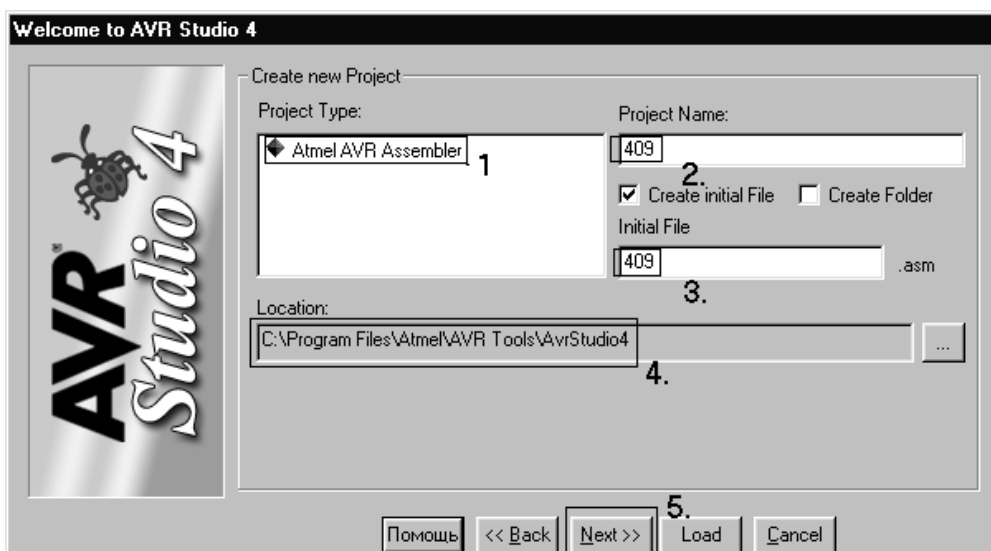


Рисунок 2.6 – Встановлення налаштувань проекту

Для цього необхідно виконати такі дії:

- у вікні **Project type** необхідно обрати Atmel AVR Assembler;
- у вікні **Project Name** необхідно ввести ім'я проекту та для створення нового файлу встановити прапорець **Create initial File**;
- вказати шлях збереження файлів проекту.

Третім етапом є вибір системи налагодження.

Програмне забезпечення **AVR STUDIO 4** має у своєму складі програми налагодження для широкого спектру задач (рис. 2.7), що передбачає:

- вибір вбудованого симулятора-налагоджувача;
- вибір пристрою, для якого створюється програма.

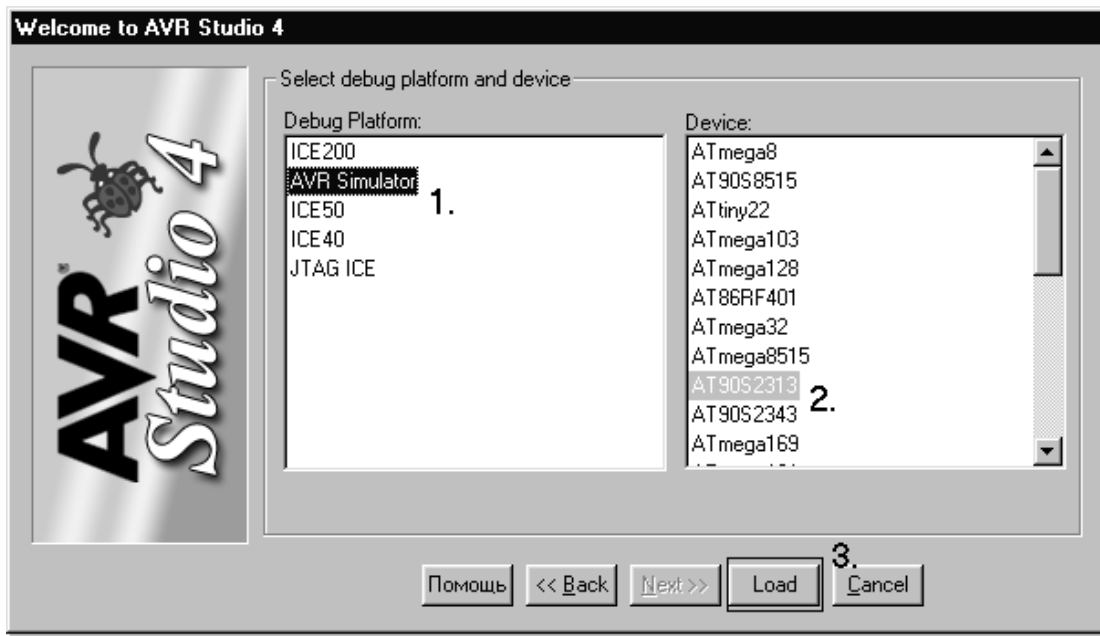


Рисунок 2.7 – Вибір симулятора та мікроконтролера

Четвертим етапом є написання програми.

На початку даного етапу необхідно зберегти створений файл кнопкою “**Save**” у файловому меню.

Після введення тексту програми необхідно натиснути клавішу **F7**, щоб провести компіляцію програми. Після цього з’являється вікно, у якому міститься інформація про правильність написаної програми. При виявленні помилок повідомляється про всі рядки, у яких виявлено помилки.

Натиснувши на рядок, у якому повідомляється про помилку, в робочій програмі буде показано місце, де виявлено помилку.

Після завершення компіляції без помилок, тобто у вікні (див. рис. 2.4), нижній рядок буде мати вигляд **Assembly complete with no errors** (програма відкомпільована без помилок). Тепер можна приступати до відлагодження роботи програми (аналогічно як для **AVR STUDIO 3**).

2.3 Програмування мовою C у середовищі WinAVR

2.3.1 Створення файлу вихідного тексту програми

Відкрийте програму Programmers Notepad (PN), що входить до складу WinAVR, та виберіть новий C/C++ файл (рис. 2.8).

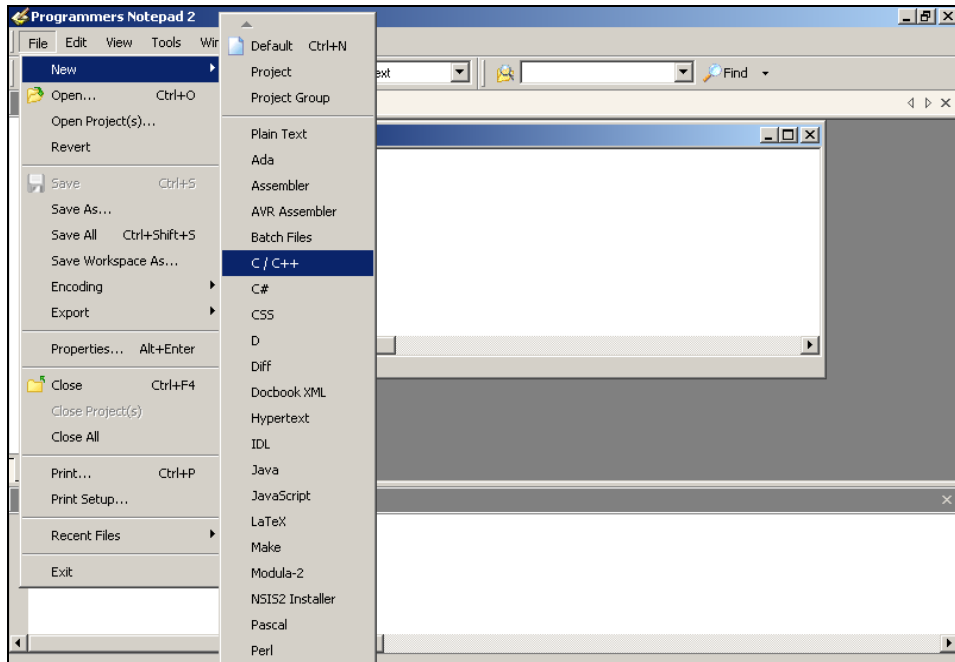


Рисунок 2.8 – Створення нового файлу в редакторі Programmers Notepad

Створіть папку проекту, наприклад **D:\prj**. У PN наберіть код і збережіть файл у папці проекту під ім'ям **simple.c**.

```
#include <avr/io.h>
#include <stdint.h>
// Підключення необхідних бібліотек
void main(void)
{
    MCUCR = 0xA0; //Дозволити роботу з зовнішньою
пам'яттю
    unsigned char* const pLeftInd = 0xA000;
    //Покажчик на комірку пам'яті за адресою 0xA000
    *pLeftInd = 0x66;
    while (1)
    {
        // Нескінченний цикл
    }
}
```

2.3.2 Створення makefile

Для компіляції програми в середовищі WinAVR використовується один з найпотужніших і найгнучкіших компіляторів – GCC. І саме через свою універсальність він має дуже багато налаштувань, котрі для компіляції програми можна кожний раз вводити через командний рядок, а можна один раз записати у спеціальний makefile.

Після компіляції буде створено об'єктний файл у форматі операційної системи Linux (ELF). Цей об'єктний файл необхідно перетворити в формат HEX. Ця дія забезпечується за допомогою лінковника, який в свою чергу може бути викликаний з командного рядка, або керуватися налаштуваннями makefile. Тож найкращий вихід побудови файлу програми – це побудова з використанням makefile.

Створити makefile можна або вручну, або за допомогою спеціального генератора.

Мейкфайли для мікроконтролерів AVR є специфічними, і тому для успішного налаштування необхідно змінити лише декілька параметрів.

У ролі генератора мейкфайлів буде використовуватись опція **Mfile**.

Для створення makefile проекту виконуються такі кроки.

1. Завантажуємо програму **Mfile** (входить до складу WinAvr) і вибираємо тип МК (рис. 2.9).

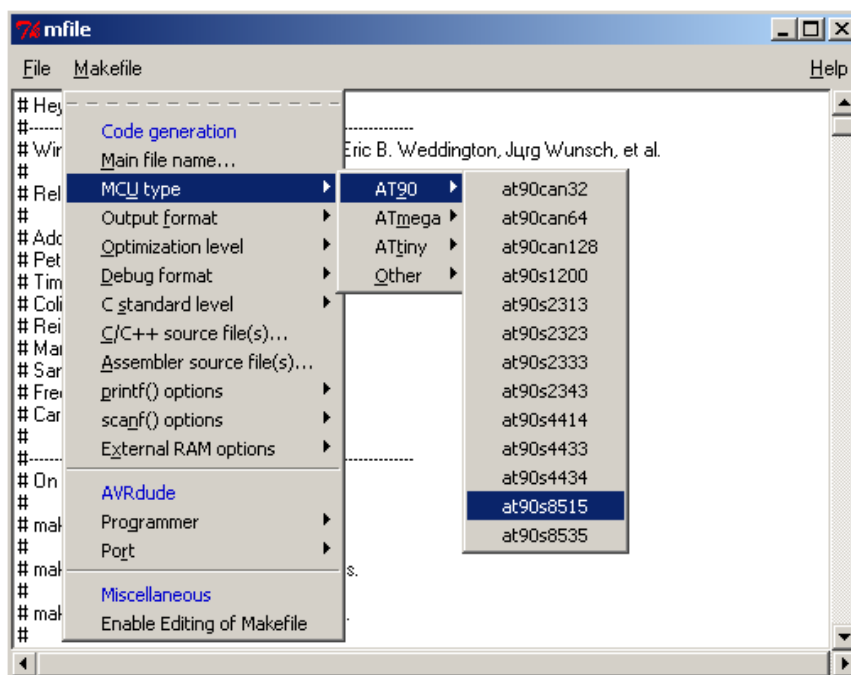


Рисунок 2.9 – Вибір типу мікроконтролера для створення makefile

2. У меню **Output format** виберіть **ihex** (Intel HEX)
3. У меню **C standard level** – **gnu99**.
4. Зберігаємо файл у директорію проекту. Файл обов'язково повинен мати ім'я **makefile**
5. Додаємо до **makefile** запис про вихідний текст програми, що знаходиться в файлі **simple.c**.

Зробити це можна за допомогою діалогу, що викликається при виборі пункту меню **C/C++ source file(s)**. Але ця частина програми працює не завжди коректно, і тому, якщо змін файлу не сталося (зміни у **makefile** підсвічуються жовтим кольором), цей параметр потрібно встановити вручну. Для цього поставте прапорець на пункті меню **Makefile → Enable Editing of Makefile**.

Знайдіть параметр **TARGET** та встановіть його значення:

TARGET = simple.

Знайдіть параметр **SRC** і встановіть його значення:

SRC = simple.c.

Прибираємо значення параметра **CPPSRC**.

6. Зберігаємо вміст **makefile**.

Необхідні налаштування зображені на рисунку 2.10.

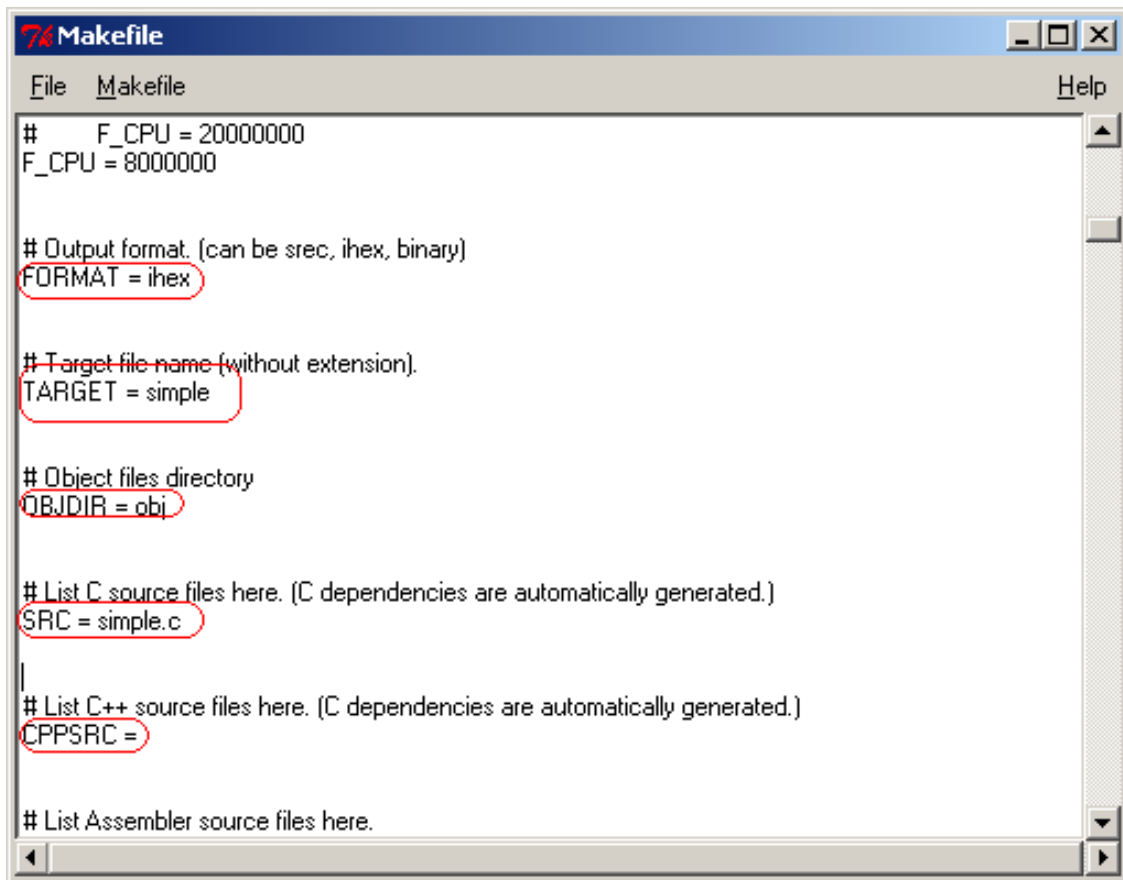


Рисунок 2.10 – Налаштування **makefile**

2.3.3 Компіляція та лінування

Після виконаних дій у директорії проекту з'являться 2 потрібних файли (рис. 2.11): **simple.c** і **makefile**.

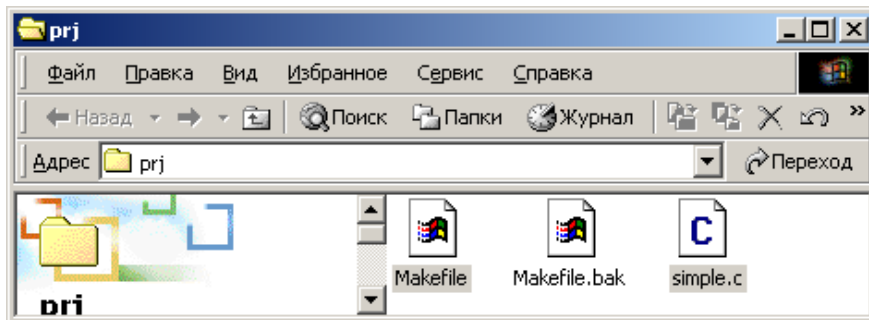


Рисунок 2.11 – Файли проекту

Тепер потрібно створити виконуваний файл програми (ihex), що зрозумілий МК:

1. Відкривається вихідний текст програми в редакторі **PN**;
2. Вибираємо пункт меню **Tools** → **[WinAVR] Make Clean** (рис. 2.12).

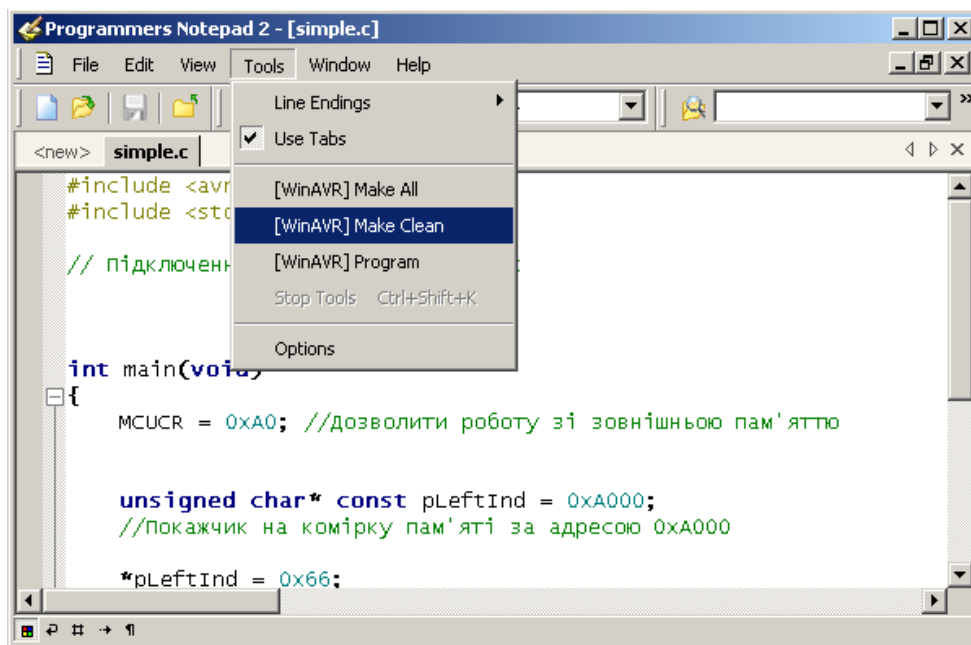


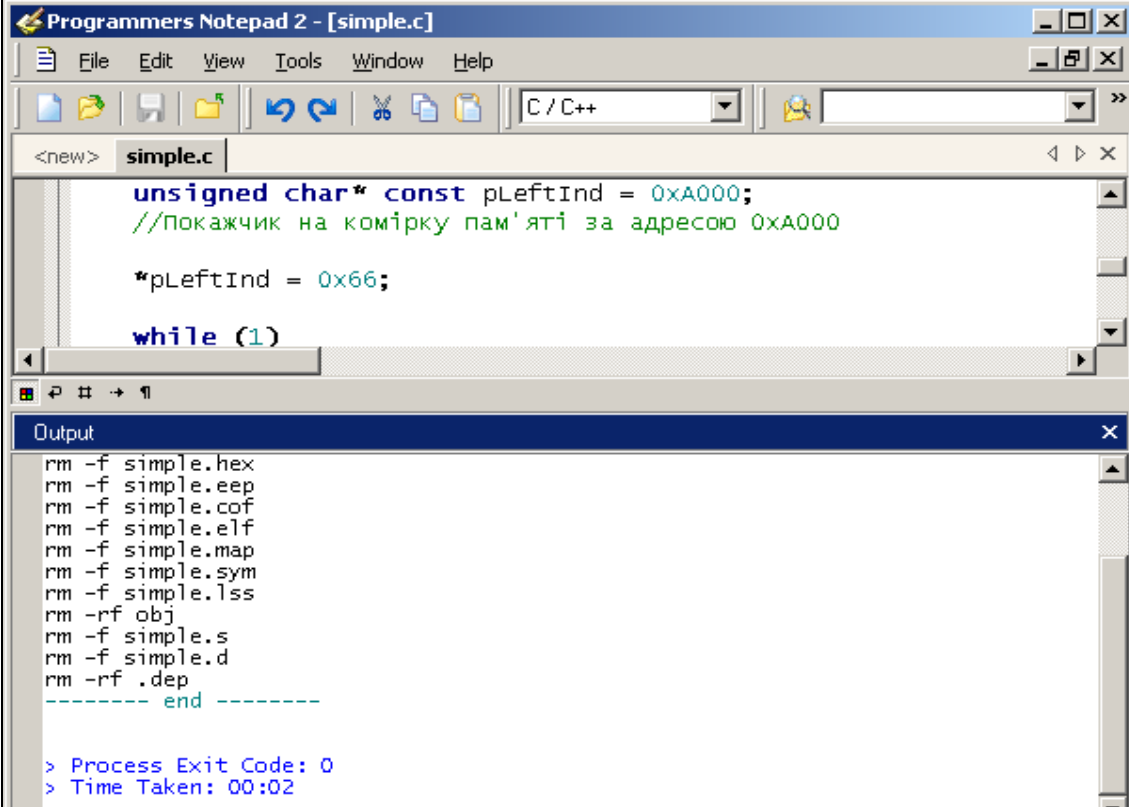
Рисунок 2.12 – Виконання make clean

Ця дія запускає спеціальну утиліту **make**, яка, згідно з параметрами, повинна очистити всі зайві файли, що могли залишитися від попередньої компіляції.

Під час виконання команди, в нижній частині **PN**, з'явиться вікно "Output" (рис. 2.13), що буде інформувати про хід виконання команди.

Операція вважається успішною, якщо ви побачите внизу рядок:
"> **Process Exit Code: 0**".

Якщо значення **Process Exit** відмінне від нуля – це означає, що сталася помилка. Для операції "**make.exe**" **clean** помилку потрібно шукати у неправильній конфігурації мейкфайлу.



The screenshot shows a window titled "Programmers Notepad 2 - [simple.c]". The main text area contains the following C code:

```
unsigned char* const pLeftInd = 0xA000;
//Показчик на комірку пам'яті за адресою 0xA000

*pLeftInd = 0x66;

while (1)
```

Below the code is an "Output" window showing the results of a "make.exe clean" operation:

```
rm -f simple.hex
rm -f simple.eep
rm -f simple.cof
rm -f simple.elf
rm -f simple.map
rm -f simple.sym
rm -f simple.lss
rm -rf obj
rm -f simple.s
rm -f simple.d
rm -rf .dep
----- end -----

> Process Exit Code: 0
> Time Taken: 00:02
```

Рисунок 2.13 – Результати роботи операції "make.exe" clean

Якщо операція пройшла успішно, то виконуємо побудову виконуваного файлу. Для цього вибираємо пункт **Tools** → **[WinAVR] Make all**. Якщо результат компіляції успішний, тобто "Process Exit Code: 0" – це означає, що необхідний *hex* файл був побудований без помилок.

Інакше – помилки знаходяться у тексті програми. Всю необхідну інформацію про місце виникнення помилки можна дізнатися з вікна "Output".

Компіляція та лінування програми завершено. Якщо результат операції був успішним, то в директорії проекту з'являться такі нові файли:

- *simple.eep* – містить інформацію, що буде скопійована в EEPROM;
- *simple.elf* – об'єктний файл. Саме на основі інформації цього файлу буде створений виконуваний файл;

- simple.hex – виконуваний файл. Програма в кодах МК;
- simple.lss – містить інтерпретацію відкомпільованої програми мовою Асемблер;
- simple.map, simple.sym – допоміжні файли, що описують хід компіляції та лінування проекту.

2.4 Програмування мікроконтролерів AVR у середовищі MicroBasic та MicroPascal

2.4.1 Редактор коду

Редактор коду являє собою звичайний текстовий редактор. Особливостями редактора є: синтаксична система виділення; кодовий "Помічник"; кодові шаблони; автокорегування; система закладок і переходів за мітками.

Ці опції можна налаштувати через вікно *Редактора Налаштувань*. Tools → options from the Drop-down menu, або натиснути іконку Tools (рис. 2.14).

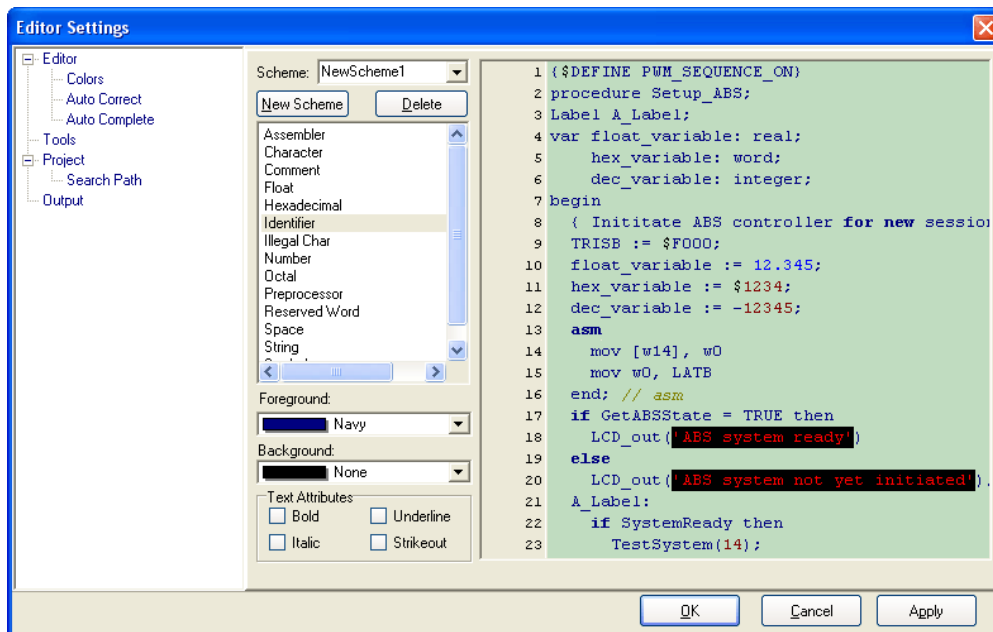


Рисунок 2.14 – Налаштування редактора коду

Кодовий "Помічник" [CTRL+SPACE]

Якщо надрукувати кілька перших літер слова і натиснути комбінацію [CTRL+SPACE], то всі можливі комбінації продовження будуть виведені на екран (рис. 2.15). Можна вибрати потрібну команду, за допомогою клавіш стрілок і клавіші ENTER, замість того, щоб її набирати на клавіатурі.

```

LCD_]
procedure LCD_Config(Port, RS, EN, RW, D7, D6, D5, D4);
procedure LCD_Out(var PORT: byte; Row: byte; Column: byte; var text: char)
procedure Lcd_Init(var PORT: byte)
procedure Lcd_Chr(var port: byte; Row: byte; Column: byte; Out_Char: byte)
procedure Lcd_Cmd(var port: byte; Out_Char: byte)
const LCD_FIRST_ROW = 128;
const LCD_SECOND_ROW = 192;
const LCD_THIRD_ROW = 148;

```

Рисунок 2.15 – Кодовий "Помічник"

"Помічник" параметра [CTRL+SHIFT+SPACE]

"Помічник" параметра буде автоматично з'являтися коли відкриваються дужки або натискається комбінація [CTRL+SHIFT+SPACE]. Якщо назва функції іде поперед дужки, то на висхідній панелі будуть запропоновані очікувані параметри. Якщо вводиться потрібний параметр, то наступні запропоновані параметри зникнуть.

```

ADC_Read channel : word;

```

Рисунок 2.16 – Помічник параметра

Шаблони коду [CTR+J]

Можна вставляти шаблони, написавши їх назву, а потім натиснути [CTR+J], і редактор автоматично згенерує потрібний код. Можна натиснути кнопку меню Code і вибрати потрібний шаблон.

Можна додавати власні шаблони. Для цього виберіть Tools > Options або Tools з Settings Toolbar, а потім виберіть Auto Complete Tab, де можна ввести ключове слово, опис і сам Код шаблону.

Автокорегування

Система автокорегування виправляє деякі помилки при наборі коду. Щоб зайти до списку розпізнавання помилок, виберіть Tools > Options або Tools з Settings Toolbar, а потім виберіть Auto Correct Tab. Можна додати свої зміни до списку.

Коментування

Редактор коду дає можливість коментувати окремі частини коду за допомогою простого натискання на кнопку Comment/Uncomment з Code toolbar (рис. 2.17).

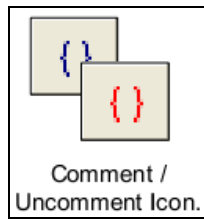


Рисунок 2.17 – Кнопка Comment/Uncomment



Рисунок 2.18 – Кнопка для згорання/розгорання дерева

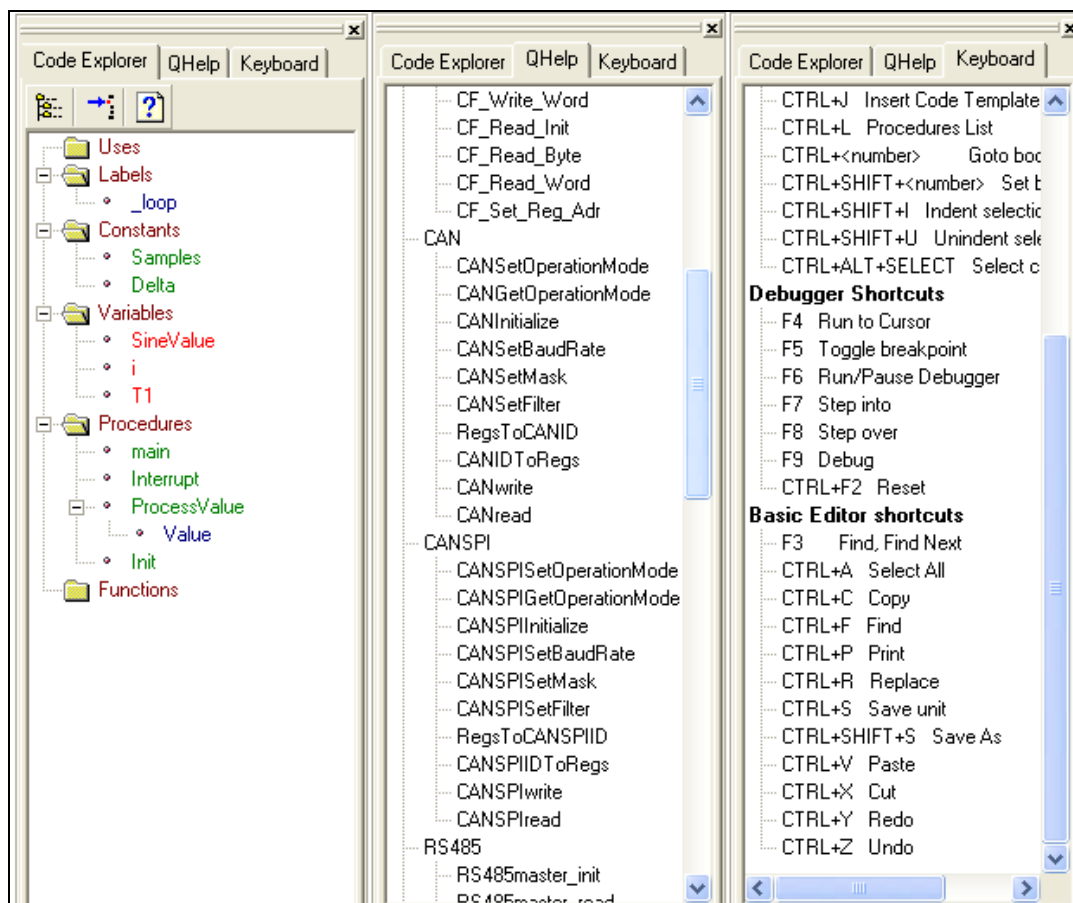


Рисунок 2.19 – Закладки Qhelp та Keyboard

Закладки

Закладки дозволяють полегшити навігацію у великій програмі. Для зручності використовуються закладки: CTRL+<номер> – перейти до закладки (Goto bookmark); CTRL+SHIFT+ <номер> – встановити закладку (Set bookmark)

Диспетчер коду (Code Explorer)







Диспетчер коду розміщено (за замовчуванням) у лівій частині головного вікна, він показує кожну оголошену частину коду, можна перейти до будь-якої його частини, або скористатись системою пошуку оголошених частин. Щоб згорнути або розгорнути "дерево" диспетчеру коду потрібно скористатись відповідними клавішами (рис. 2.18).

Також можна викликати два і більше вікна у Диспетчері коду. Закладка QHelp містить усі вбудовані та бібліотечні функції, подвійний "клік" відкриває вікно Help. Закладка Keyboard містить список усіх "Гарячих" комбінацій клавіш (рис. 2.19).

2.4.2 Налagodжувач (Debugger)

Налagodжувач (Debugger) – це інтегрований компонент середовища програмування MicroBasic/MicroPascal. Його призначення симулювати операції, що їх виконує МК AVR, і відладити програмне забезпечення, написане для цих пристроїв.

Таблиця 2.1 – Призначення кнопок роботи з налагоджувачем

<i>Кнопка</i>	<i>Призначення</i>
 Start Debugger.	Налagodження запускається клавішею F9
 Pause Debugger.	Пауза/Стоп налагодження, F6
 Step Into.	Виконує одну операцію, записану в рядку, і призупиняє подальше виконання програми, F7.
 Step Over.	Виконує перехід через цикл і зупиняється на першій команді, що слідує за ним, F8.
 Run to Cursor.	Виконує всі інструкції до поточного положення курсора, F4.
 Toggle Breakpoint.	Установлення точки переривання на даній позиції курсора, для перегляду всіх точок переривань виберіть Run>View Breakpoints, подвійний "клік" на вибраній точці показує її місцезнаходження.

Після написання програми можна запустити налагоджувач натисканням кнопки Run >Debug або натисканням на іконку Debug.

Запуск налагоджувача дає доступ до опцій: крок вперед, крок назад та інших. Рядок, що виконується, виділяється.

Вікно перегляду

Вікно перегляду налагоджувача – це вікно, що дозволяє відслідковувати роботу програми.

Для вибору Вікна виберіть View → Debug Windows → Watch Window (рис. 2.20).

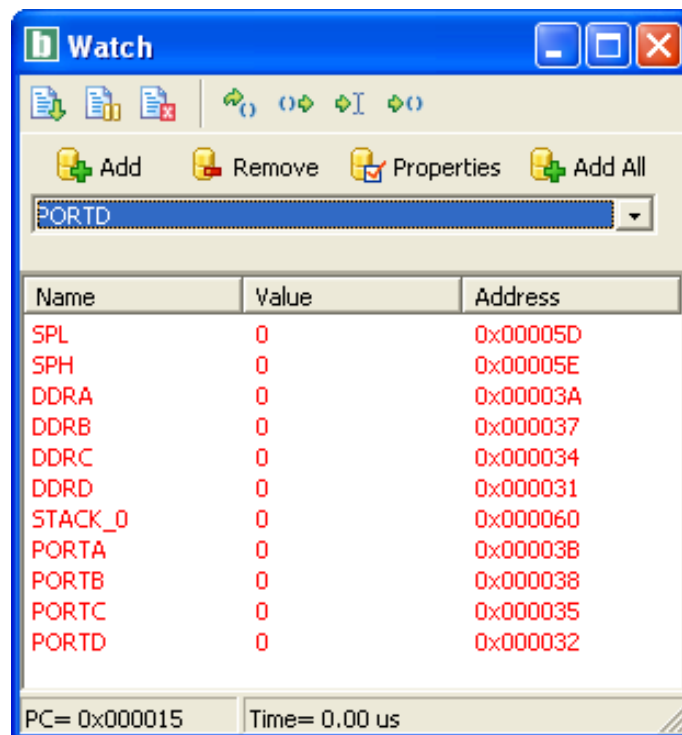


Рисунок 2.20 – Вікно перегляду

Вікно перегляду показує змінні, регістри МК, з їхніми адресами і змістом, зміст оновлюється протягом роботи програми. Можна використовувати меню для виведення інших даних, змінені дані виділяються червоним.

Подвійний "клік " на значенні виводить вікно у якому можна змінити його, також змінити на інші системи числення.

Вікно таймера

Вікно таймера налагоджувача можна викликати, вибравши View → Debug Windows → View Clock (рис. 2.21). Вікно таймера відображає час від останньої дії. Таймер відображає час виконання

(кількість циклів) з моменту запуску налагоджувача і може бути скинутим у будь-який момент. Delta показує кількість циклів між попереднім рядком і поточним.

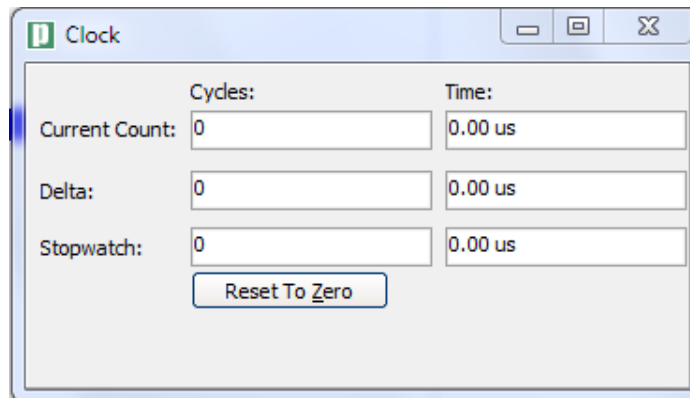


Рисунок 2.21 – Вікно таймера

Можна змінювати частоту роботи МК, але це ніяк не відобразиться на роботі програми.

2.4.3 Вікно помилок (Error Window)

У випадку виникнення помилок під час компіляції компілятор інформує про них і не створюється *hex* файл.

Вікно помилок розміщено внизу головного вікна (рис. 2.22). Воно вказує розміщення і тип помилки, що її знайшов компілятор.

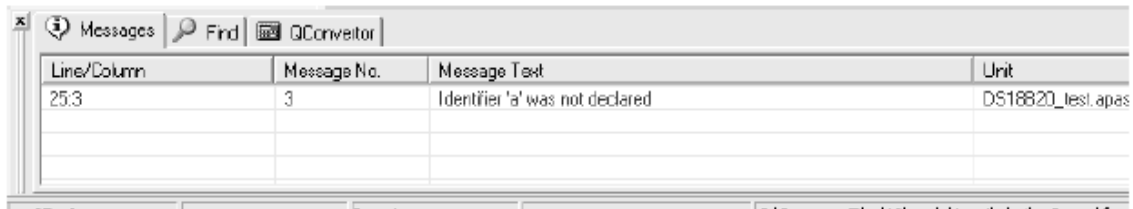


Рисунок 2.22 – Вікно помилок

Подвійний "клік" на повідомленні показує рядок де знаходиться помилка.

2.4.4 Статистика

Після закінчення компіляції можна переглянути статистику вашої програми. Виберіть Project → view Statistics, або натисніть на іконку Statistics. Вікно містить шість закладок, що відображають використання динамічної та постійної пам'яті; дерево переривань відповідно до кожної процедури і функції: розмір, початкова і кінцева

адреси; зміст усіх регістрів, їхні адреси, назви змінних і їх адреси; зміст постійної пам'яті МК.

2.4.5 Інтегровані інструменти

До складу MikroBasic/MicroPascal входить синхронно-асинхронний приймач-передавач. Його можна запустити за допомогою меню Tools → Terminal (рис. 2.23).

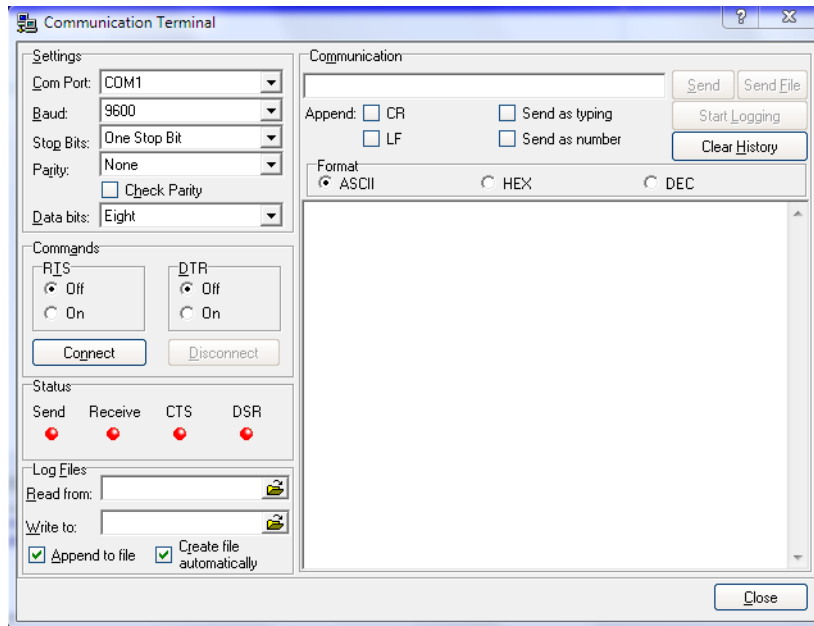


Рисунок 2.23 – Синхронно-асинхронний приймач-передавач

До складу MikroBasic/MicroPascal входить таблиця ASCII II, яка корисна для роботи з рідинно-кристалічним індикатором. Викликається таблиця через меню Tools → ASCII chart (рис. 2.24).

ASCII chart						
0 NUL	1 SOH	2 STX	3 ETX	4 EOT	5 ENQ	6 ACK
10 LF	11 VT	12 FF	13 CR	14 SO	15 SI	16 DLE
20 DC4	21 NAK	22 SYN	23 ETB	24 CAN	25 EM	26 SUB
30 RS	31 US	32	33 !	34 "	35 #	36 \$
40 (41)	42 *	43 +	44 ,	45 -	46 .
50 2	51 3	52 4	53 5	54 6	55 7	56 8
60 <	61 =	62 >	63 ?	64 @	65 A	66 B
70 F	71 G	72 H	73 I	74 J	75 K	76 L
80 P	81 Q	82 R	83 S	84 T	85 U	86 V

Рисунок 2.24 – Таблиця ASCII

До складу MikroBasic/MicroPascal входить декодер для 7-сегментного індикатора, що показує цифровий еквівалент коду (рис. 2.25).

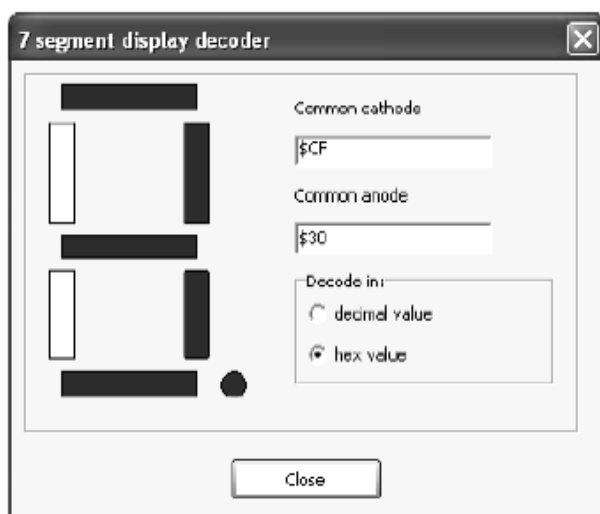


Рисунок 2.25 – Вікно декодера 7-сегментного індикатора

2.4.6 Гарячі клавіші (Keyboard Sortcuts)

Основні комбінації гарячих клавіш, які використовуються у MikroBasic/MicroPascal, наведені у таблицях 2.2 – 2.5.

Таблиця 2.2 – Гарячі клавіші інтегрованого середовища розробки (IDE Shortcuts)

<i>Комбінація клавіш</i>	<i>Призначення</i>
F1	Допомога (Help)
CTRL+N	Новий елемент (New Unit)
CTRL+O	Відкрити (Open)
CTRL+F9	Компілювати (Compile)
CTRL+F11	Ввімкнути/вимкнути провідник коду (Code Explorer on/off)
CTRL+SHIFT+F5	Показати мітки (View breakpoints)

Таблиця 2.3 – Гарячі клавіші базового редактора (Basic Editor shortcuts)

<i>Комбінація клавіш</i>	<i>Призначення</i>
F3	Знайти, знайти наступне (Find, Find Next)
CTRL+A	Виділити все (Select All)
CTRL+C	Копіювати (Copy)
CTRL+F	Знайти (Find)

Продовження таблиці 2.3

<i>Комбінація клавіш</i>	<i>Призначення</i>
CTRL+P	Друкувати (Print)
CTRL+R	Перемістити (Replace)
CTRL+S	Зберегти елемент (Save unit)
CTRL+SHIFT+S	Зберегти як (Save As)
CTRL+V	Вставити (Paste)
CTRL+X	Вирізати (Cut)
CTRL+Y	Повторити (Redo)
CTRL+Z	Відмінити (Undo)

Таблиця 2.4 – Гарячі клавіші покращеного редактора (Advanced Editor shortcuts)

<i>Комбінація клавіш</i>	<i>Призначення</i>
CTRL+SPACE	Помічник коду (Code Assistant)
CTRL+SHIFT+SPACE	Помічник параметрів (Parameters Assistant)
CTRL+D	Знайти декларування (Find declaration)
CTRL+G	Перейти до рядка (Goto line)
CTRL+J	Вставити шаблон коду (Insert Code Template)
CTRL+<номер>	Перейти до закладки (Goto bookmark)
CTRL+SHIFT+<номер>	Встановити закладку (Set bookmark)
CTRL+SHIFT+1	Вирізати виділене (Indent selection)
CTRL+SHIFT+U	Вставити виділене (Unindent selection)
CTRL+ALT+SELECT	Виділити колонки (Select columns)

Таблиця 2.5 – Гарячі клавіші налагоджувача (Debugger shortcuts)

<i>Комбінація клавіш</i>	<i>Призначення</i>
F4	Перейти до курсора (Run to Cursor)
F5	Встановити мітку (Toggle breakpoint)
F6	Старт/Пауза налагоджувача (Run/Pause Debugger)
F7	Крок в (Step into)
F8	Крок над (Step over)
CTRL+F8	Крок з (Step out)
F9	Налагоджувач (Debug)
CTRL+F2	Скидання (Reset)
CTRL+F5	Додати до списку спостереження (Add to watch list)
CTRL+F6	Показати лічильник (View Clock)
SHIFT+CTRL+F6	Показати ОЗП (View RAM)
SHIFT+F6	Змінити ОЗП (Edit RAM)

2.4.7 Проекти (Projects)

MicroBasic/MicroPascal створює проекти, які складаються з файлу проекту (.abr) і одного або більше основних файлів (.abas/.apas). Можна компілювати основні файли лише тоді, коли вони є частиною проекту.

Проект несе таку інформацію: назву проекту і опис налаштувань, робочий пристрій, таймер пристрою, список основних файлів.

2.4.8 Основні файли (Source files)

Основні файли повинні мати розширення .abas/.apas. Список основних файлів зберігається у файлі проекту.

При підключенні нового основного файлу програма шукатиме його спочатку в папці де встановлено програму, а потім у директорії введеної користувачем.

2.4.9 Компіляція (Compilation)

Коли створений проект і написаний основний файл, його необхідно скомпілювати. Процес компіляції може бути перерваний у випадку, якщо компілятор виявив помилки у коді. Основні відомості, що стосуються компіляції, наведені у таблиці 2.6.

Таблиця 2.6 – Вихідні файли

<i>Тип файлу та розширення</i>	<i>Призначення</i>
HEX-файл (.hex)	Файл для програмування МК
Двійкова бібліотека (.mc1)	Може підключатись до інших проектів
Список (list-file) (.lst)	Опис розподілу пам'яті
Файл асемблера (.asm)	Код мовою асемблер

2.5 Програмування мікроконтролерів AVR у середовищі AVR Builder

2.5.1 Середовище розробки AVR Builder

Середовище призначене для створення повного циклу розробки. Розробка програми може виконуватись як на рівні асемблера, так і на макрорівні з маніпуляцією багатобайтовими величинами зі знаком.

На відміну від класичного асемблера програма вводиться у вигляді алгоритму з деревоподібними розгалуженнями і відображається на площині у двох вимірах. Мережа умовних і безумовних переходів відображається графічно у зручній векторній формі. Вся логічна структура програми стає наочною.

Візуальність логічної структури зменшує ймовірність помилок і скорочує строки розробки.

За оцінкою користувачів, у порівнянні з класичним асемблером, час на розробку програмного забезпечення скорочується в 3-5 разів.

Для налаштування периферійних пристроїв (таймери, UART, ADC, SPI і т.п.) передбачений спеціальний елемент алгоритму – "налагоджувач" з віконним інтерфейсом. У ньому досить вибрати необхідні параметри роботи пристрою, а набір інструкцій, що забезпечують ці параметри, сформує компілятор (у правій частині вікна).

Підтримується автоматичне перекодування рядків ANSI-кодів Windows в коди русифікованого буквено-цифрового рідинно-кристалічного індикатора.

Середовище поєднує в собі графічний редактор, компілятор алгоритму, симулятор МК та програматор. При використанні програматора МК підключається до COM-порту комп'ютера через нескладний адаптер. Algorithm Builder забезпечує моніторне відлагодження на кристалі (On Chip debug), що дозволяє спостерігати вміст реального кристала в заданій точці зупинки. При цьому, для зв'язку МК з комп'ютером використовується тільки один вивід, на вибір користувача. Моніторне відлагодження може бути застосовано до будь-якого типу кристала, що має SRAM.

Середовище призначене для роботи з операційною системою Windows 98/2000/NT/ME/XP.

2.5.2 Розробка програмного забезпечення за допомогою AVR Builder

Будь-яке програмне забезпечення можна розбити на окремі логічно завершені фрагменти. Як правило, кінцевим оператором цих фрагментів є такі оператори, як безумовний перехід або повернення з підпрограми, тобто оператори, після яких лінійне виконання програми однозначно припиняється. Розробка програмного забезпечення в середовищі Algorithm Builder зводиться до формування таких блоків, розміщення їх на площині й встановлення між ними векторних зв'язків з умовних і безумовних переходів.

Основний інтерфейс програми зображено на рисунку 2.26.

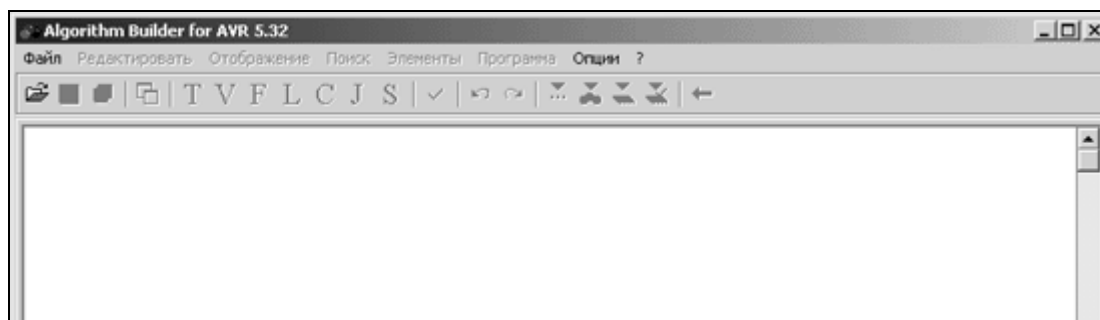


Рисунок 2.26 – Вікно програми AVR Builder

Для побудови алгоритму в Algorithm Builder передбачено сім елементів:

- TEXT (рядок локального текстового редактора);
- VERTEX (вершина блока);
- FIELD (поле);
- LABEL (мітка);
- CONDITION (умовний перехід);
- JMP Vector (відносний безумовний перехід);
- SETTER (налагоджувач периферійних пристроїв).

Усі елементи знаходяться на панелі інструментів для швидкого доступу до них та зручного використання (рис. 2.27).

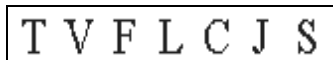


Рисунок 2.27 – Елементи AVR Builder

Елемент "TEXT" являє собою текстовий рядок, що починається від лівого краю поля алгоритму. Сукупність із декількох таких рядків утворює локальний текстовий редактор, обведений пунктирними лініями. Правила роботи в ньому, аналогічні іншим текстовим редакторам.

Рядки призначені для запису в них ряду директив компілятора, а також для коментарів. Для додавання нового рядка текстового редактора потрібно вибрати меню "Элементы > Текст", або натиснути клавіші "Alt+T". Також можна натиснути кнопку "T" на панелі інструментів. Коментарі повинні позначатися двома похилими рисочками: "//".

Елемент "LABEL" являє собою вертикальний штрих, розташований всередині блоку операторів і необов'язкове ім'я, що розташовується ліворуч або праворуч від штриха. Мітка призначена для позначення місць в алгоритмі, куди можливе здійснення умовних і безумовних переходів. Така реалізація значно спрощує розробку програми, оскільки код стає більш зрозумілим, ніж у звичайній програмі мовою Assembler.

Для додавання мітки в блок використовується пункт меню "Элементы → Метка", або ж натисканням клавіші "Alt+L", або кнопкою "L" на панелі інструментів.

Елемент «VERTEX» за своїм змістом й призначенням повністю ідентичний мітці, але, на відміну від неї, задає розташування блоку на робочій площині й завжди є його початком. Створення нової вершини виконується з пункту меню "Элементы → Вершина", або ж викликом гарячої клавіші "Alt+V", або натисканням кнопки "V" на панелі інструментів.

Елемент "FIELD" являє собою рядок у блоці. Об'єкт призначений для запису більшості операторів МК. Щоб додати поле, потрібно вибрати пункт меню "Элементы → Поле", або натиснути кнопку "F" на панелі інструментів, або комбінацією клавіш "Alt+F", або клавішею "Enter" у випадку, якщо курсор перебуває поза локальним текстовим редактором.

Елемент "CONDITION" конструктивно найбільш складний і призначений для реалізації умовних переходів. Являє собою овальний контур, усередині якого вписується умова переходу й можливий вектор у виді ламаної лінії зі стрілкою на кінці, біля якої можливо необов'язкове ім'я вектора. Кінець вектора повинен:

- закінчуватися на якій-небудь мітці;
- закінчуватися на вершині блоку;
- закінчуватися на відрізьку іншого вектора;
- мати ім'я адресованої мітки.

Проведення вектора до потрібного місця відбувається натисканням лівої кнопки миші при натиснутій клавіші "Alt". Для редагування вектора використовуються клавіші напрямку в комбінації із клавішею "Alt". Щоб ввести новий об'єкт, слід обрати пункт меню "Элементы→Условие", натиснути клавіші "Alt+C" або кнопку на "C" панелі інструментів.

Елемент "JMP Vector" призначений для реалізації коротких безумовних переходів (у базовому асемблері це оператор "RJMP"). Являє собою ламану лінію, що виходить із середини блоку зі стрілкою на кінці, аналогічну вектору об'єкта "CONDITION". Щоб додати новий безумовний перехід, потрібно вибрати пункт меню "Элементы → Вектор б/у перехода", натиснути комбінацію клавіш "Alt+J" або кнопку "J" на панелі інструментів.

Елемент "SETTER" являє собою сірий прямокутник, усередину якого вписане ім'я периферійного компонента МК, такого як таймер, АЦП, регістр маски переривань та ін. Налагоджувач призначений для формування послідовності операцій МК, які забезпечують завантаження необхідних констант у відповідні регістри керування

введенням/виведенням у відповідності з обраними властивостями.

Перед використанням цього елемента потрібно визначити тип МК. Для додавання налагоджувача в алгоритм потрібно обрати пункт меню "Элементы → Настройщик", натиснути клавіші "Alt+S" або кнопку "S" на панелі інструментів.

Об'єкт "SETTER" є макро оператором. Після компіляції він перетвориться в послідовність команд МК, які забезпечують завантаження необхідних констант у відповідні регістри керування. У цих операціях буде використаний регістр R16.

Для ряду компонентів, наприклад ADC, налагоджувач може впливати на декілька регістрів керування. У цьому випадку при необхідності вплив на кожний конкретний регістр можна заблокувати.

2.5.3 Приклад застосування AVR Builder

Для створення нового проекту в програмному середовищі AVR Builder потрібно вибрати пункт меню "Файл → Новый". При цьому в робочій області програми з'явиться елемент "Текст", "Вершина блоку" та елемент "Поле" з пустим оператором NOP (рис. 2.28).

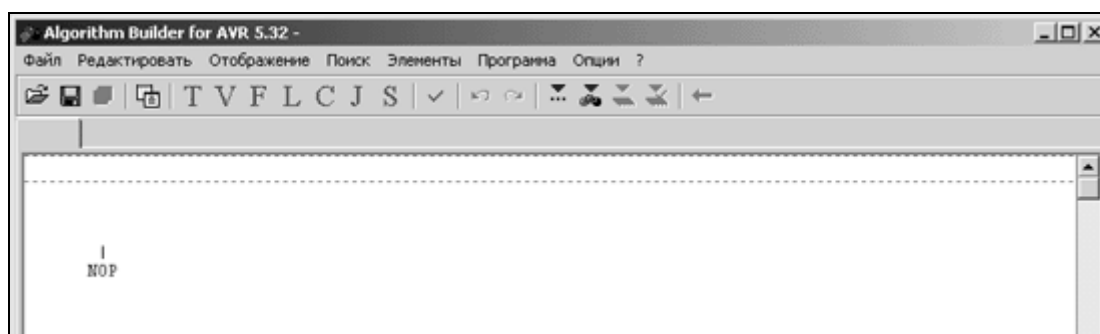


Рисунок 2.28 – Порожній проект AVR Builder.

Розглянемо застосування AVR Builder на прикладі програми "Вогонь, що біжить". Алгоритм програми зображений на рисунку 2.29. Даний алгоритм почергово виводить логічні одиниці в біти порту В. З лівої сторони відображено алгоритм основної програми, а з правої – алгоритм підпрограми затримки. Підпрограма затримки використовується для того, щоб мерехтіння "вогню, що біжить" було помітним для людського ока.

Запис даних до регістрів відбувається за допомогою вказівки ">". Для зручності її можна вводити за допомогою натискання однієї клавіші "`", що знаходиться під клавішею ESC.

Для створення нового поля використовується клавіша "Enter", а для його видалення комбінація клавіш "Ctrl+Del". Створення всіх інших елементів наводиться в пункті 2.5.2.

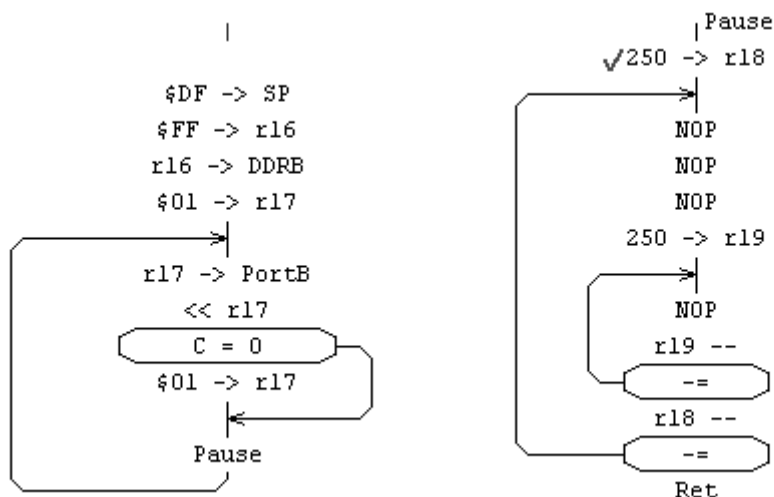


Рисунок 2.29 – Алгоритм програми "Вогонь, що біжить" у AVR Builder

Рожева помітка біля оператора "250 -> r18" означає точку зупинки для симулятора. Додається вона за допомогою клавіші "F5" або ж відповідною піктограмою на панелі інструментів.

Після завершення роботи над алгоритмом програми необхідно визначити тип МК, після чого можна буде скомпілювати програму. Для цього потрібно скористатися пунктом меню "Опции → Опции проекта..." та на закладці "Кристалл" обрати потрібний МК. Вибираємо МК AT90S2313 (рис. 2.30).

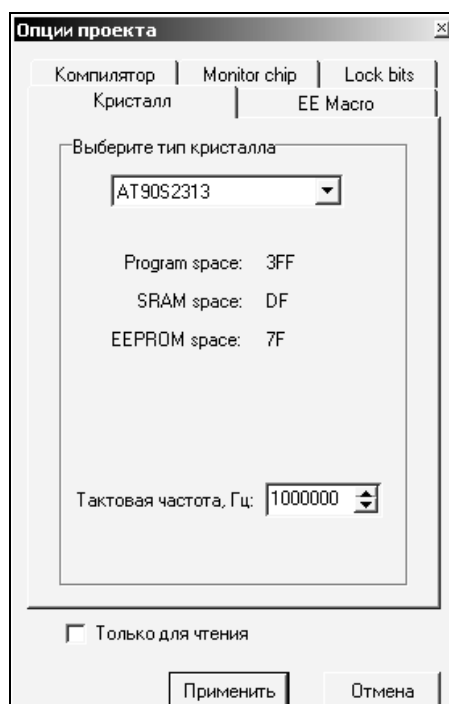


Рисунок 2.30 – Опції проекту AVR Builder

Після виконання цих дій можна скомпілювати програму та запустити на виконання в симуляторі. Для цього використовується клавіша "F9". Якщо в програмі відсутні помилки, то запуститься симулятор. Якщо ж помилки є, то про це буде повідомлено.

Для спостереження за станом МК під час виконання програми в AVR Builder передбачені вікна моніторингу (рис. 2.31 – 2.33). Їх можна викликати через пункт меню "Открыть". Для відстеження стану МК у даному алгоритмі достатньо відкрити вікна "Processor", "Working Registers" та "I/O Registers... → Port B".

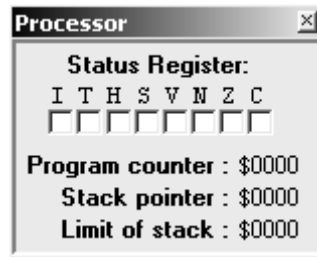


Рисунок 2.31 – Вікно стану процесора

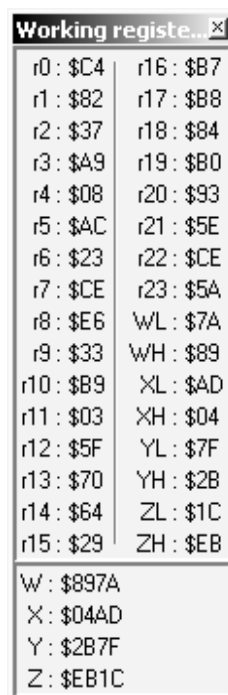


Рисунок 2.32 – Вікно стану регістрів

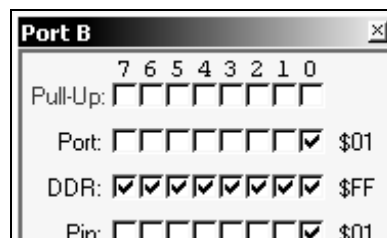


Рисунок 2.33 – Вікно стану порту B

При запуску симулятора вміст регістрів SRAM автоматично заповнюється випадковими числами, тому що в реальному часі їх вміст неможливо передбачити одразу ж після подачі живлення.

Мітка біля оператора (рис. 2.34) – це поточний стан програмного лічильника. Вона вказує на оператор, який буде виконаний у наступному кроці. Після компіляції програми та запуску симулятора вона автоматично встановлюється біля першого оператора.

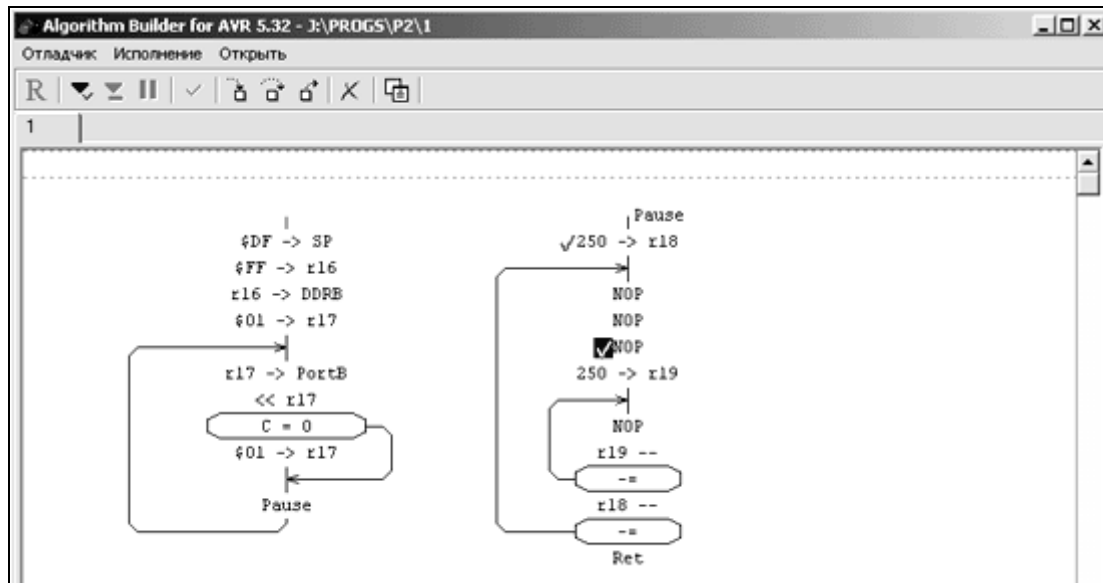


Рисунок 2.34 – Вікно симулятора AVR Builder

Для покрокового виконання програми використовується клавіша "F7". Покрокове виконання без входу в підпрограму виконується за допомогою клавіші "F8". Виконання алгоритму до виходу з підпрограми виконується клавішею "F6".

Запуск на необмежене виконання до точки зупинки виконується клавішею "F9", а виконання до виділеного елемента – клавішею "F4". При цьому виконання алгоритму може бути призупинено натисканням клавіші "F2".

Для більш зручного використання програми передбачені спеціальні кнопки на панелі інструментів симулятора, які відповідають вищезазначеним функціям (рис. 2.35).

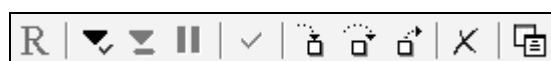


Рисунок 2.35 – Панель інструментів симулятора AVR Builder

Під час виконання алгоритму у вікні моніторингу порту В спостерігається почергова поява логічних одиниць на виходах.

РОЗДІЛ 3 ОСОБЛИВОСТІ ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ

3.1 Технічне завдання на розробку двоканального терморегулятора

Загальний опис розроблюваного виробу. Пристрій призначений для вимірювання і підтримки в заданому діапазоні температури в нагрівальній камері промислової установки термостатування. Функціональна схема установки термостатування наведена на рисунку 3.1. Камера обладнана двома електронагрівальними елементами (ТЕН1, ТЕН2), що перемикаються за допомогою електромеханічних реле змінного струму (Р1, Р2) і двома датчиками температури (ДТ1, ДТ2). Регулювання температури здійснюється шляхом нагріву, до заданої температури, повітря нагрівальної камери за допомогою електронагрівальних елементів, які включаються подачею напруги живлення, і подальшого природного його охолодження. Швидкість зміни температури повітряного середовища нагрівальної камери складає від 10 до 15 °С/хв.

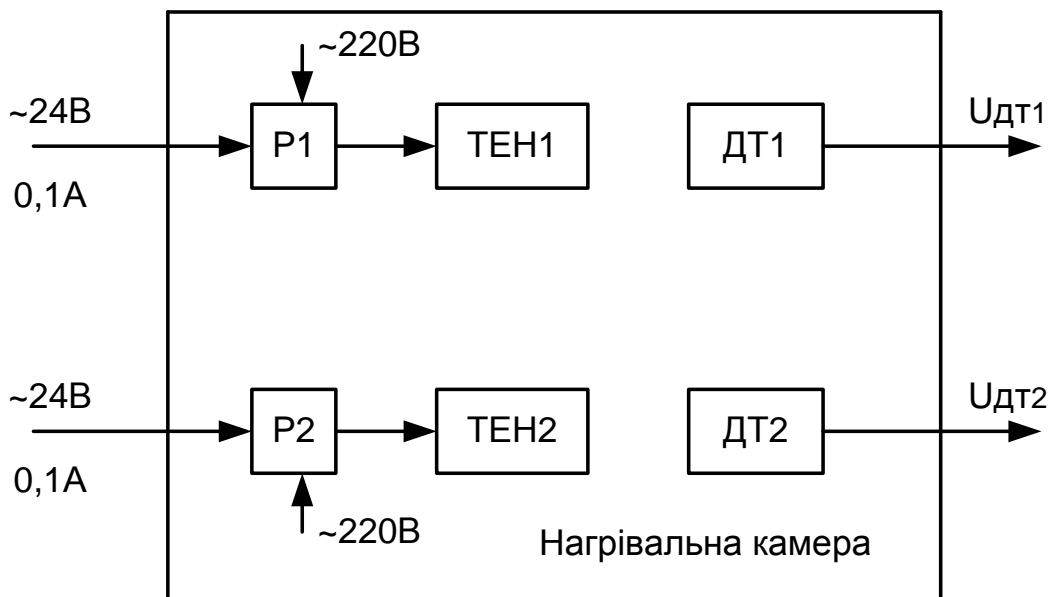


Рисунок 3.1 – Функціональна схема установки термостатування

Характеристики сигналу керування електромеханічними реле:

- номінальна напруга – ~ 24 В;
- частота сигналу керування – 50 Гц;
- номінальний струм вмикання – 0,1 А.

Вимоги до технічних характеристик терморегулятора:

- діапазон робочих температур – від 0 до +300 °С;

- кількість каналів вимірювання і регулювання – 2;
- точність вимірювання температури – ± 1 °С;
- точність підтримки заданої температури – ± 3 °С;
- дискретність устанавлення заданої температури – 1 °С.

Вимоги до функціональних характеристик терморегулятора. Виріб повинен виконувати такі функції:

- вимірювання і цифрову індикацію двох каналів поточної температури в нагрівальній камері;
- незалежне цифрове встановлення двох значень заданої температури;
- зберігання значень встановленої температури;
- індикацію вмикання електронагрівальних елементів;
- звукову сигналізацію входу/виходу в режим устанавлення заданої температури;
- індикацію вмикання пристрою (наявність напруги живлення).

Вимоги до конструкції. Пристрій повинен являти собою моноблок з роз'ємами для підключення зовнішніх датчиків температури і електромеханічних реле керування електронагрівальними елементами. Габаритні розміри пристрою не повинні перевищувати 70×70×130 мм.

Вимоги до живлення. Живлення терморегулятора здійснюється від мережі змінного струму загального користування ~220В.

Умови експлуатації. Пристрій призначений для роботи в діапазоні робочих температур навколишнього середовища від +0 до +50 °С, при відносній вологості повітря до 85% (при температурі +25 °С).

Спеціальні вимоги. Необхідно передбачити гальванічну розв'язку між частиною керування і виконавчими органами системи регулювання температури.

3.2 Розробка структурної схеми і алгоритму функціонування терморегулятора. Вибір типів датчиків і виконавчих пристроїв

Основне призначення розроблюваного пристрою – регулювання температури нагрівальної камери установки термостатування. Розроблювана система є електронною системою регулювання (керування), яка, як і більшість систем керування є системою замкнутого типу (рис. 3.2). Вона містить об'єкти керування – електронагрівальні елементи (ТЕН1, ТЕН2), виконавчі органи – ключові каскади, датчики інформаційних параметрів – датчики температури (ДТ1, ДТ2) і пристрій керування (ПК). Особливістю даної системи є наявність двох незалежних каналів керування.

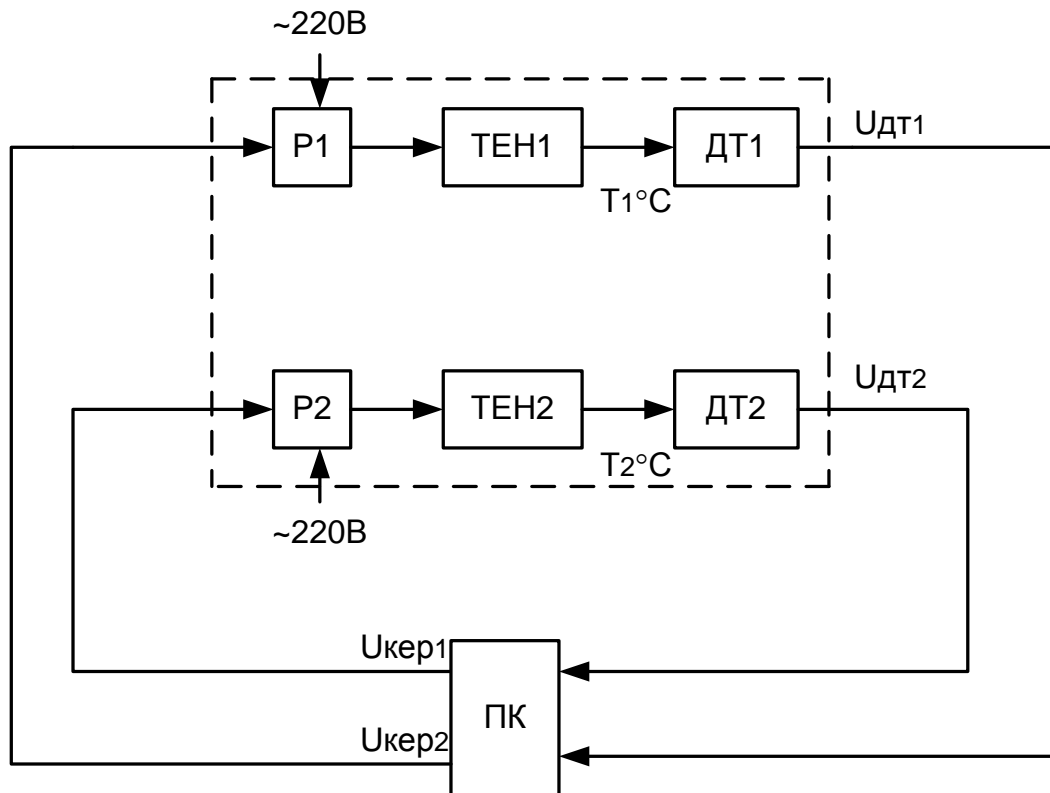


Рисунок 3.2 – Узагальнена структура системи регулювання температури нагрівальної камери

Принцип дії системи регулювання полягає в тому, що: датчики температури ДТ1 і ДТ2 вимірюють поточну температуру в певних областях нагрівальної камери; сигнал з ДТ1 і ДТ2 надходить на пристрій керування, який порівнює поточні значення із заданим інтервалом температур. При цьому повинна виконуватися умова:

$$\left(T_i^{yсм} - \Delta T\right) \leq T_i \leq \left(T_i^{yсм} + \Delta T\right), \quad (3.1)$$

де T_i – поточна температура ($i=1,2$);

$T_i^{yсм}$ – значення встановленої температури;

ΔT – точність підтримки температури.

Якщо поточне значення за будь-яким каналом вимірювання нижче значення $(T_i^{yсм} - \Delta T)$, то пристрій керування виробляє сигнал, який відкриває відповідний ключовий каскад, внаслідок чого напруга живлення подається на електронагрівальний елемент. Відбувається нагрів повітряного середовища нагрівальної камери до тих пір, поки не виконається умова $T_i \leq (T_i^{yсм} + \Delta T)$. При виконанні вказаної умови пристрій керування, виробляє сигнал замикання ключового каскаду і здійснюється природне охолодження нагрівальної камери.

Датчики температури за матеріалом, з якого вони виготовляються, діляться на металеві і напівпровідникові. За принципом дії металеві датчики діляться на терморезистивні датчики і термопари [21, 38].

У основі роботи терморезистивних датчиків лежить залежність електричного опору від температури. У різних діапазонах робочих температур ця залежність апроксимується інтерполяційними рівняннями різного ступеня, але в першому наближенні її можна вважати лінійною:

$$R_T = R_0 [1 + \alpha(T - T_0)], \quad (3.2)$$

де T – поточна температура;

R_0 – значення електричного опору при температурі

$T_0 = 0 \text{ } ^\circ\text{C}$ (273K);

R_T – значення опору при температурі $T \text{ } ^\circ\text{C}$;

α – температурний коефіцієнт опору (ТКС).

Прецизійні терморезистивні датчики виготовляють з нікелю (ТКС=5,4*10⁻³ K⁻¹) або платини (ТКС=3,9910⁻³ K⁻¹) [21]. Їх застосовують у діапазоні температур від -260 °С до +1300 °С.

Для вимірювання опору та відповідно, і температури на основі терморезистивних датчиків широко застосовують вимірювальну схему – міст Уїтсона (рис. 3.3).

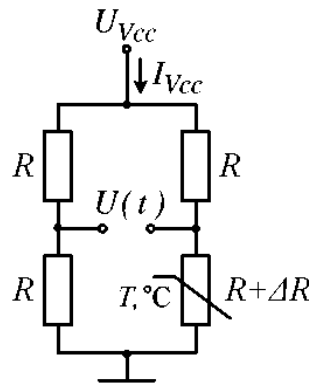


Рисунок 3.3 – Міст Уїтсона

При $\Delta R = 0$ міст знаходиться в нульовому (збалансованому) стані $U(t) = 0$. Вихідна напруга моста пов'язана з зміною опору ΔR нелінійно:

$$U(t) = \frac{U_{Vcc}}{4} \left[\frac{\Delta R}{R + \frac{\Delta R}{2}} \right]. \quad (3.3)$$

Нелінійність схеми моста складає 0,05% від абсолютного значення. Дана нелінійність стосується тільки схеми вимірювання, і оскільки вона описується аналітично, то може бути компенсована при цифровій обробці сигналу вимірювання.

Додаткова похибка вимірювань вноситься за рахунок саморозігріву опору, внаслідок чого струм живлення моста I_{Vcc} повинен бути дуже малим.

Таким чином, недоліками методу вимірювання температури на основі терморезистивних датчиків є нелінійність вимірювань, похибки за рахунок саморозігрівання терморезистора, залежність вихідного сигналу від напруги живлення моста.

Крім того, як показує практика, через конструктивні особливості виготовлення терморезисторів, вони мають більшу інерційність, ніж термопари.

Термопари належать до датчиків генераторного типу. В основі їх роботи лежить ефект генерації ТЕРМО-ЕРС у точці контакту двох різнорідних металів під дією температури. Залежно від виду металів ТЕРМО-ЕРС може змінюватися від 7 мкВ/°С до 75 мкВ/°С. Діапазон робочих температур складає від 0°С до +2300°С [21].

При підключенні датчика на основі термопари з'являється додаткова паразитна ТЕРМО-ЕРС, яка може бути компенсована шляхом реалізації повного датчика. Повний датчик утворює дві термопари (див. рис. 3.4), одна з яких постійно знаходиться при температурі 0°С (температура танення льоду).

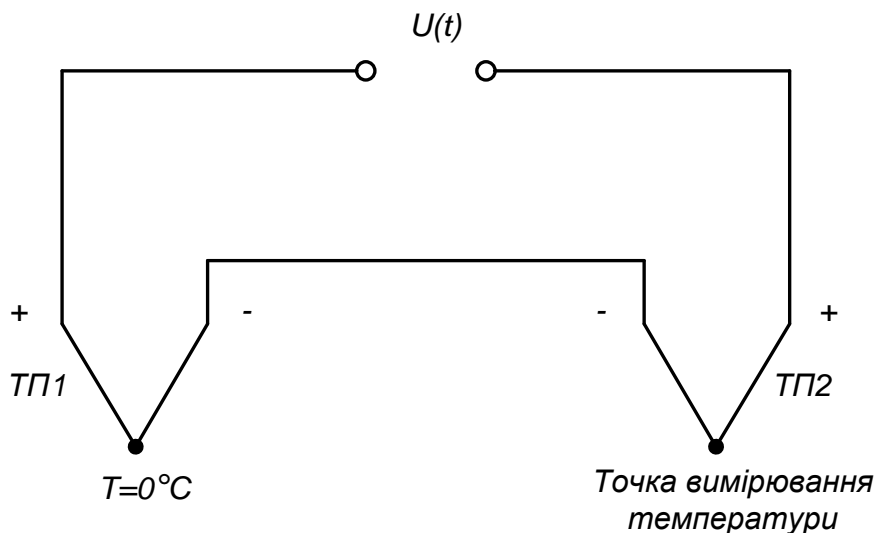


Рисунок 3.4 – Схема вимірювання температури з використанням термопар

Перевага методу вимірювання температури з використанням термопар – високі точність і роздільна здатність. Недоліком є низька

повторюваність і виникнення паразитних ТЕРМО-ЕРС, унаслідок чого виникає необхідність налаштування приладу.

Напівпровідникові датчики температури засновані на залежності опору кристала від температури. ТКС таких датчиків у 3-5 разів більше ТКС металевих датчиків, але, оскільки їх діапазон робочих температур складає від -50°C до $+180^{\circ}\text{C}$, що не задовольняє вимоги технічного завдання, розглядати можливість застосування датчиків вказаного типу у даній системі терморегулювання не має сенсу.

Грунтуючись на наведеному вище аналізі, можна зробити висновок, що для розроблюваної системи необхідно застосувати датчики на основі термопар.

Ключовими каскадами системи терморегулювання є реле змінного струму, що комутують напругу живлення ($\sim 220\text{ В}$) до нагрівальних елементів (ТЕН1, ТЕН2). Спеціальною вимогою при їх реалізації, згідно з технічним завданням (ТЗ), є гальванічна розв'язка з пристроєм керування.

Гальванічна розв'язка традиційно здійснюється з використанням електромеханічних приладів (трансформаторів і реле). Той самий ефект може бути досягнутий використанням приладів з оптичним зв'язком (оптоелектронних реле або оптронів). Принцип дії приладів з оптичним зв'язком заснований на подвійному перетворенні енергії. У передавачі електричний сигнал перетворюється в світловий пучок, а в приймачі здійснюється обернене перетворення, внаслідок чого приймач і передавач зв'язані тільки оптичним зв'язком, що дає можливість здійснювати гальванічну розв'язку приймача і передавача.

Оптоелектронні прилади перевершують електромагнітні аналоги за надійністю, довговічністю, перехідними і частотними характеристиками.

Можна відзначити такі переваги цих приладів:

- можливість реалізації безконтактного оптичного керування електронними об'єктами та обумовлені цим різноманітність і гнучкість конструкторських рішень кіл керування;
- однонаправленість розповсюдження інформації оптичним каналом, відсутність зворотної реакції приймача на випромінювач;
- широка частотна смуга пропускання оптрона, відсутність обмеження з боку низьких частот (що властиве імпульсним трансформаторам);
- можливість передачі оптичною лінією як імпульсного сигналу, так і постійної складової;
- оптичні канали зв'язку захищені від дії завад електромагнітних полів;

- для оптронів не існує принципових фізичних або конструктивних обмежень з досягнення високої напруги і опорів розв'язки, а також малої прохідної ємності.

Використання оптронів не вимагає спеціальних засобів погодження з цифровими ІМС.

Згідно з ТЗ система терморегулювання повинна здійснювати цифрову індикацію поточної температури та цифрове встановлення заданої температури. Очевидно, що у такому разі переважним є цифровий варіант реалізації пристрою керування. При цьому для відображення значень поточної температури необхідно застосувати семисегментні світлодіодні індикатори (по три десяткових розряди на канал), оскільки відображення текстової інформації не потрібне.

Пристрій керування системи терморегулювання як цифровий обчислювальний пристрій може бути реалізований як на основі наборів ІС цифрової логіки, так і на основі МК. Другий варіант є кращим, оскільки дозволяє мінімізувати габарити розроблювального пристрою, скоротити терміни розробки і залишає можливість реконфігурації системи. Структурна схема системи терморегулювання, що розробляється, наведена на рисунку 3.5.

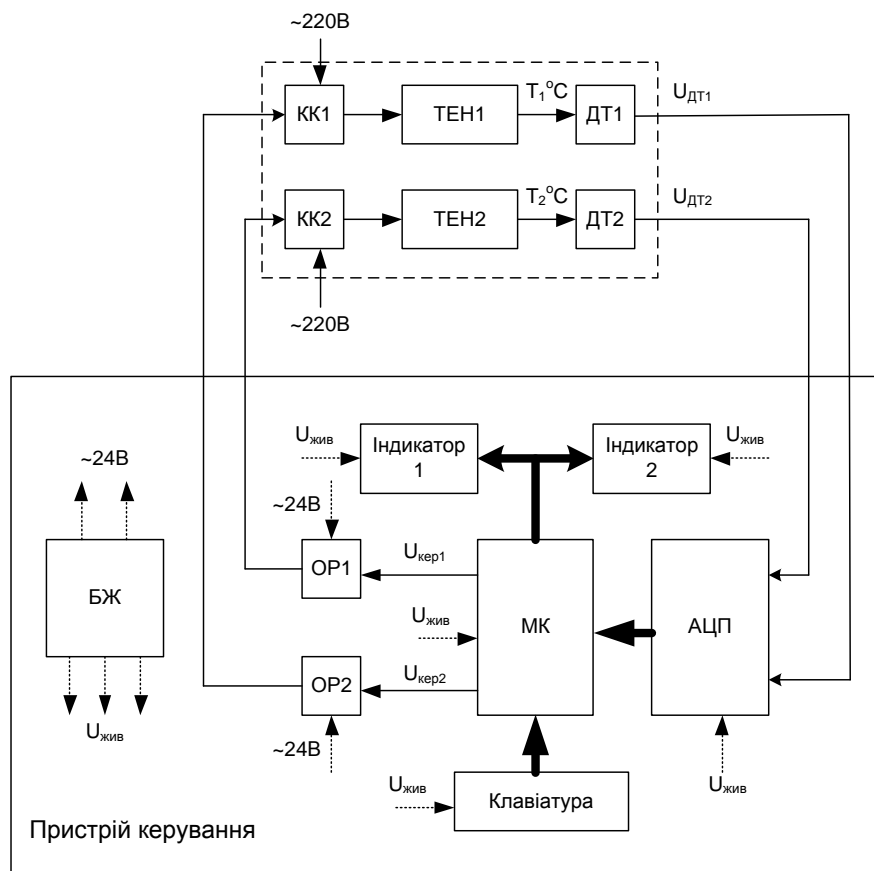


Рисунок 3.5 – Структурна схема системи терморегулювання, що розробляється

Для перетворення сигналів аналогових датчиків температури до форми, зручної для обробки в обчислювальних пристроях, в схемі передбачений двоканальний аналого-цифровий перетворювач (АЦП). Індикатори 1 і 2 служать для безперервної цифрової індикації задаваної поточної температури нагрівальної камери. Клавіатура призначена для встановлення заданої температури стабілізації. Оптоелектронні реле ОР1 і ОР2 здійснюють гальванічну розв'язку керувальної та силової частин системи терморегулювання.

Усі активні пристрої контролера керування живляться від блока живлення пристрою, на вхід якого подається напруга живлення мережі змінного струму загального користування ~220 В.

3.3 Вибір елементної бази і розробка схеми електричної принципової терморегулятора

Синтез принципової схеми почнемо з вибору основних елементів системи.

Датчики температури. Як датчики температури вибрані датчики на основі термоелектричного ефекту (термопари). Залежно від діапазону вимірюваних температур їх підрозділяють на низькотемпературні (до 300 °С), середньотемпературні (до 600 °С) і високотемпературні (понад 1800 °С), а залежно від способу застосування – на занурювані в робоче середовище і поверхневі. Занурювані термопари використовують для вимірювання температури в газоподібних і рідких неагресивних і агресивних середовищах. Очевидно, що найповніше умовам ТЗ задовольняють середньотемпературні занурювані термопари, оскільки для низькотемпературних верхня межа діапазону робочих температур розмірна з вказаним в ТЗ значенням. Серед термопар вказаних типів, що випускаються промислово, найбільшого розповсюдження отримали хромель-алюмелеві (ХА) або хромель-копелеві (ХК) термопари. Основні характеристики цих термопар наведені у таблиці 3.1 [38].

Таблиця 3.1 – Основні характеристики ХА і ХК термопар

Матеріал термопари	Тип	Діапазон вимірюваних температур, °С	Значення ТЕРМО-ЕРС, мВ, при температурі робочого спаю, °С						
			0	50	100	150	200	250	300
ХА	ТХА	-200...+1000	0	2,44	4,10	6,14	8,14	10,15	12,21
ХК	ТХК	-50...+600	0	4,10	6,90	10,62	14,57	18,69	22,88

Як видно з таблиці 3.1, значення ТЕРМО-ЕРС термопари хромель-копель (ТХК) приблизно в два рази вищі, ніж у термопари хромель-алюмель (ТХА) в одному і тому ж діапазоні робочих температур, хоча діапазон вимірюваних температур у ТХА ширший. Оскільки діапазон вимірюваних температур ТХК задовольняє задане в ТЗ значення, як датчики температури розроблюваної системи терморегулювання виберемо термопари типу ТХК.

Аналого-цифровий перетворювач. Аналого-цифровий перетворювач – пристрій, призначений для перетворення вхідного аналогового сигналу у вихідний цифровий код. Мікросхеми АЦП випускаються рядом виробників: Analog Devices, Maxim, Burr-Brown, STMicroelectronics, Hitachi, National Semiconductor, Texas Instruments і інші [11, 41]. Основними характеристиками АЦП є розрядність, швидкодія, діапазон вхідної напруги, значення опорної напруги, тип інтерфейсу, характеристики живлення, тип корпусу.

Розрядність АЦП визначає роздільну здатність (точність) перетворення. В даний час випускаються АЦП з розрядністю від 8 до 24. Роздільну здатність АЦП можна оцінити, знаючи значущість молодшого розряду як величину зворотну 2^n , де n – кількість розрядів АЦП. Слід зазначити, що реальна похибка перетворення дещо вища за значущість молодшого розряду за рахунок нелінійності характеристики АЦП, а також залежить від можливого "пропуску коду".

У деяких застосуваннях дуже важливе значення має швидкодія АЦП. За швидкодією сучасні АЦП можна умовно розділити на три групи: АЦП низької швидкодії (до 100 ksp/s), АЦП середньої швидкодії (від 100 до 500 ksp/s), АЦП високої швидкодії (більше 500 ksp/s). Для технологічних контролерів найчастіше використовуються АЦП середньої і низької швидкодії.

Опорну напругу АЦП визначає діапазон можливих вхідних значень напруги. Вхідній напрузі, величина якої рівна значенню опорної напруги, відповідає максимальний цифровий код. Опорна напруга задається спеціальним високостабільним джерелом опорної напруги, яке може бути як зовнішнім по відношенню до АЦП, так і внутрішнім (вбудованим).

У разі, якщо АЦП реалізований у вигляді окремої ІМС, вирішальне значення може мати швидкість обміну АЦП з обчислювальним пристроєм. Вона залежить від використовуваного інтерфейсу передачі даних. Як правило, в АЦП реалізований один з уніфікованих послідовних або паралельних інтерфейсів. Разом з тим в багатьох моделях сучасних МК АЦП присутній у вигляді периферійного пристрою.

У розроблюваній системі терморегулювання може бути використаний АЦП як реалізований у вигляді окремої ІМС, так і інтегрований у МК. Як правило, АЦП, реалізовані у вигляді окремої ІМС мають вищі показники роздільної здатності і швидкодії, але при цьому ускладнюється реалізація обміну даними між АЦП і обчислювальним пристроєм (потрібне використання програмної або апаратної реалізації послідовного чи паралельного інтерфейсів обміну даними). Як показує досвід практичної реалізації систем подібного типу, для вимірювання температури із заданою в ТЗ точністю цілком достатньо інтегрованого АЦП. Тому наявність АЦП буде одним з критеріїв вибору МК пристрою керування.

Індикатор. Використання семисегментних світлодіодних індикаторів у вбудованих системах дозволяє відображати десяткові цифри, деякі букви російського і латинського алфавітів, а також деякі спеціальні знаки. Семисегментні індикатори за схемою з'єднання в них світлодіодних сегментів діляться на два види:

- із загальним анодом (рис. 3.6, а);
- із загальним катодом (рис. 3.6, б).

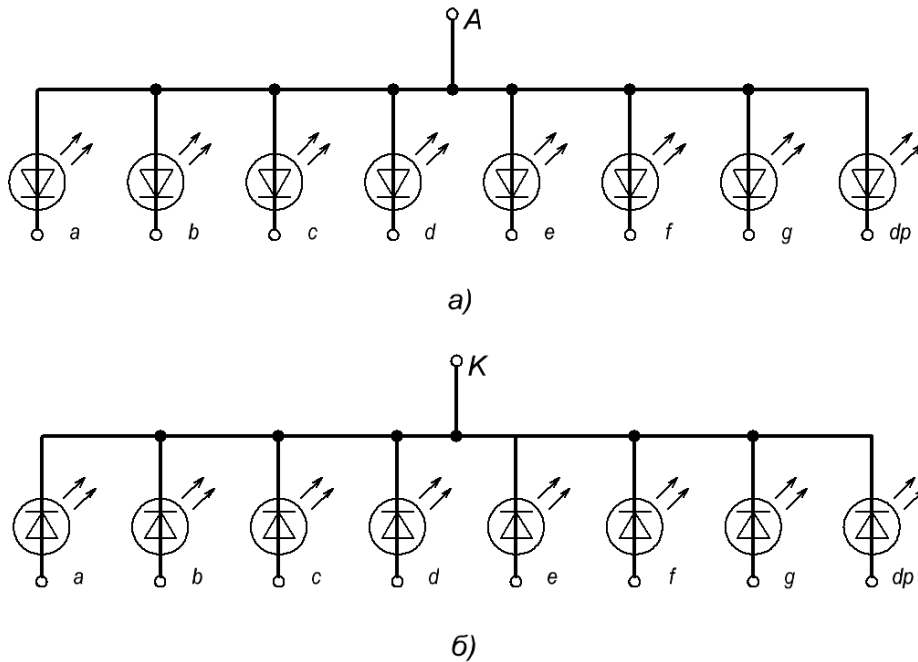


Рисунок 3.6 – Типи з'єднань сегментів в індикаторах

"Запалення" світлодіодного сегмента для індикатора першого типу здійснюється шляхом подачі логічного "0", другого – логічної "1" на відповідний вивід індикатора.

Для відображення цифри на семисегментному індикаторі, її

машинне подання перетвориться в байт індикації (двійково-десятковий код), який безпосередньо подається на виводи індикатора.

Для відображення багатосимвольної інформації окремі семисегментні індикатори об'єднуються в лінійні (однорядкові) дисплеї. Існує два способи організації інтерфейсу МК з лінійним дисплеєм: статичний і динамічний. Перший вимагає наявності на входах кожного індикатора спеціальних буферних регістрів для зберігання коду символів, що виводяться. Це збільшує число додаткових мікросхем, а також підвищує споживану потужність.

Динамічний заснований на тому, що людське око є інерційною ланкою; інформація, що відображається на дисплеї, якщо її оновлювати з частотою не менше 20 разів за секунду, здається незмінною. Динамічний спосіб індикації вимагає значно менших апаратних витрат, але складнішого програмного забезпечення. Саме цей спосіб знайшов переважне розповсюдження у вбудованих системах.

При динамічній індикації байт індикації надходить одночасно на входи всіх семисегментних індикаторів, що створюють лінійний дисплей, а вибір знакомісця здійснюється байтом вибірки. Для забезпечення яскравої і рівномірної індикації необхідно забезпечити: по-перше, заборону вибірки знакомісця на час зміни байта індикації, по-друге, регенерацію зображення на кожному знакомісці з частотою не менше 20 Гц.

У даний час промисловістю випускаються спеціалізовані ІМС, що реалізують не тільки апаратне перетворення числа в двійково-десятковий код, але й здійснюють керування лінійним семисегментним індикатором. Прикладом такого індикаторного контролера (драйвера) служить ІМС MAX7219/MAX7221 [67], що випускається фірмою MAXIM. Драйвери MAX7219/MAX7221 використовуються для керування лінійним 8-розрядним (до 8 десяткових розрядів числа) семисегментним індикатором із загальним катодом. Вони містять у собі апаратний декодер, який здійснює перетворення з двійкового у двійково-десятковий код, і ОЗП для зберігання восьми байтів індикації. Драйвер MAX7221, окрім іншого, має вбудований апаратний інтерфейс SPI для обміну даними з процесорним пристроєм, що полегшує програмну реалізацію обміну даними.

Таким чином, у разі використання MAX7221 як індикаторів можуть бути вибрані будь-які 3-розрядні семисегментні індикатори із загальним катодом, наприклад, BT-N325RD [81]. При цьому схема підключення індикаторів системи терморегулювання матиме вигляд, наведений на рисунку 3.7.

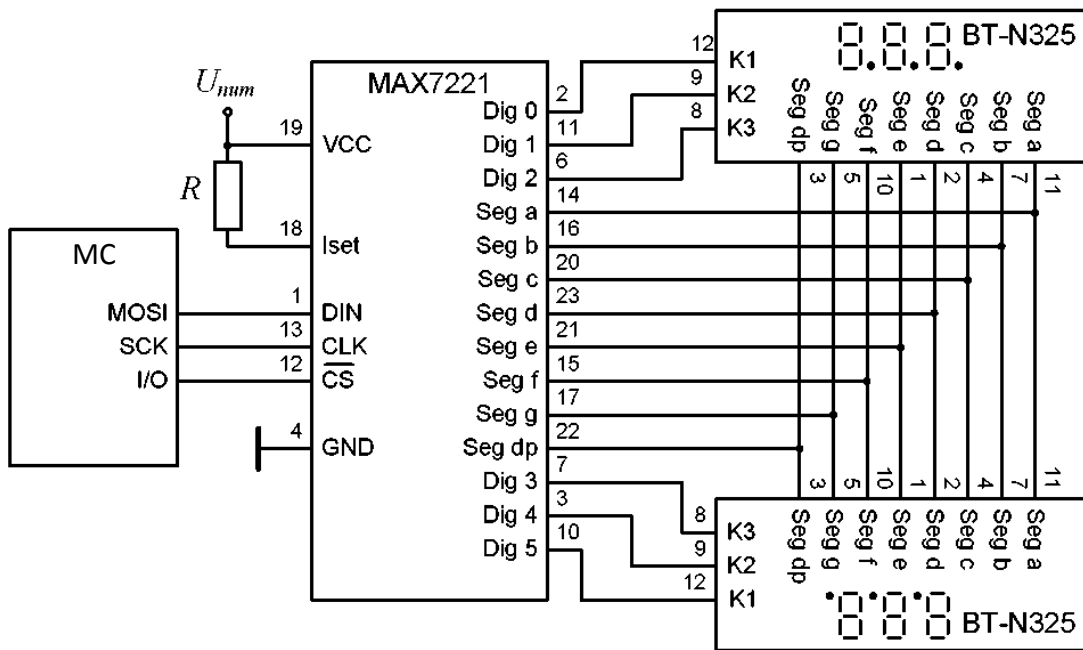


Рисунок 3.7 – Схема підключення індикаторів температури

Резистор R дозволяє задати струм для всіх сегментів одночасно. Мінімальне значення $R = 9,53$ кОм відповідає струму розряду 40 мА. Середній струм, а отже, і середня потужність, що розсіюється розрядом, визначається не тільки струмом розряду, але й частотою регенерації зображення на ньому. Виберемо $R = 10$ кОм.

При виборі мікроконтролера слід звернути увагу на наявність у ньому вбудованого інтерфейсу SPI.

Для встановлення заданої температури і вибору режиму роботи системи терморегулювання використовуємо трипозиційну клавіатуру. Призначення клавіш буде детальніше описано в підрозділі, що стосується розробки програмного забезпечення мікроконтролера.

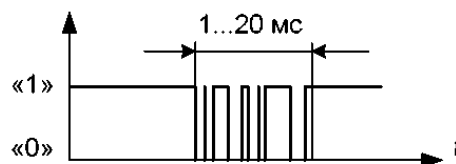


Рисунок 3.8 – До пояснення ефекту тремтіння контактів

При роботі мікроконтролера з датчиками, що мають механічні або електромеханічні контакти (кнопки, клавіші, реле, клавіатури і так далі), виникає явище, яке називають тремтінням контактів. Це призводить до перехідного процесу, при якому сигнал з контакту може бути прочитаний як випадкова послідовність нулів і одиниць (рис. 3.8). Усунути це небажане явище можна апаратними або програмними способами.

Апаратні способи полягають у використанні схеми одновібратора (рис. 3.9, а), або фільтра нижніх частот (рис. 3.9, б) для згладжування пульсацій з періодом меншим, ніж постійна часу фільтра $\tau = RC$.

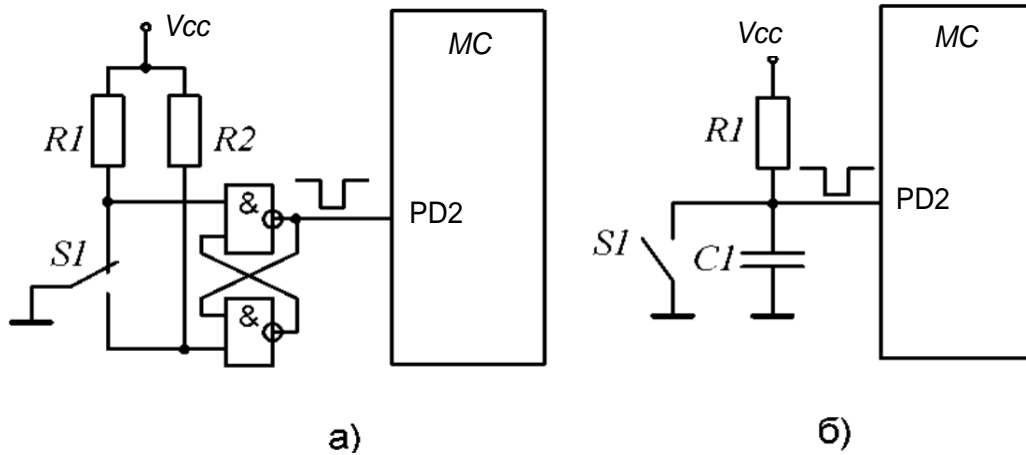


Рисунок 3.9 – Способи апаратного усунення тремтіння контактів

Серед програмних способів найбільше розповсюдження отримали два:

- підрахунок заданого числа значень сигналу, що збігаються;
- часова затримка.

Суть першого способу полягає в багатократному зчитуванні сигналу з контакту. Підрахунок вдалих опитувань (тобто опитувань, які виявили, що контакт замкнутий) ведеться програмним лічильником. Якщо після серії вдалих опитувань зустрічається невдале, то підрахунок починається спочатку. Контакт вважається стійко замкнутим (тремтіння усунено), якщо пройшло N вдалих опитувань. Число N підбирається експериментально для кожного типу використовуваних датчиків і лежить в межах від 10 до 100.

Усунення тремтіння контактів шляхом введення часової затримки полягає в тому, що програма, виявивши замикання контакту, забороняє опитування стану цього контакту на якийсь час, свідомо більший тривалості перехідного процесу. Час затримки підбирається експериментально для кожного типу датчиків в межах 1...20 мс і реалізується, зазвичай, за допомогою одного з таймерів/лічильників.

При проектуванні системи терморегулювання зупинимось на способі усунення тремтіння контактів з використанням RC-ланки, оскільки він вимагає мінімальних апаратних витрат і не ускладнює програмної реалізації процедур обробки натиснення клавіш. Електрична принципова схема підключення клавіатури до МК наведена на рисунку 3.10. Розрахуємо параметри RC-ланок. Для всіх RC-ланок на рисунку 3.10 параметри встановимо однаковими.

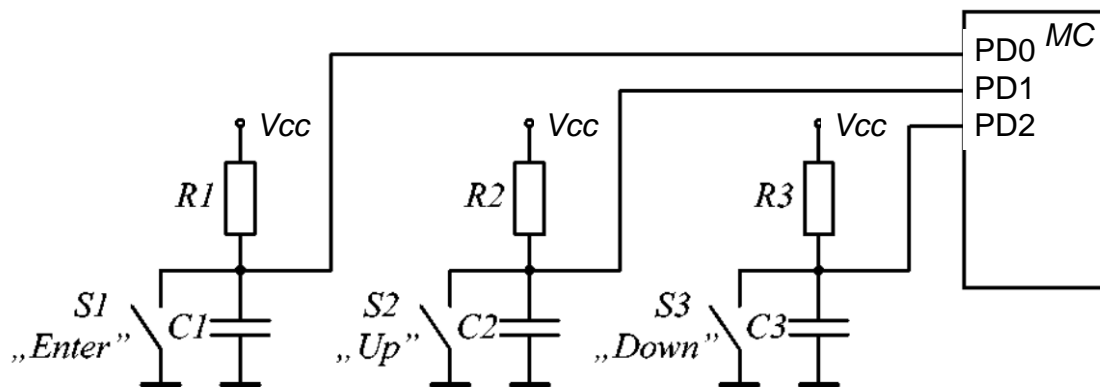


Рисунок 3.10 – Схема підключення клавіатури до мікроконтролера пристрою керування

Хай час перехідних процесів (час тремтіння контактів) у використуваних клавішних перемикачах $S1...S3$ дорівнює $t_{др} = 1$ мс. Умовою стабільної фіксації логічного рівня на вході МК при усуненні тремтіння контактів за допомогою RC-ланки є нерівність

$$t_{др} = \tau_{RC}, \quad (3.4)$$

де $\tau_{RC} = RC$ – постійна часу RC-ланки.

Задано значенням резисторів $R1 = R2 = R3 = 10$ кОм. Тоді згідно з (3.4) $C1 = C2 = C3 \geq 1 \cdot 10^{-7}$ Ф. Виберемо значення з стандартного ряду номіналів $C1 = C2 = C3 = 0,1$ мкФ.

Вибір мікроконтролера системи регулювання температури. Для системи регулювання може бути використаний один з 8-бітових МК, широко вживаних в даний час для розв'язання подібного роду завдань.

Вибір МК є питанням неоднозначним, таким, що має декілька варіантів рішення. Як правило, завдання розв'язується евристичними методами. Основними характеристиками, які при цьому беруться до уваги, є:

- продуктивність, MIPS (MIPS - Million Instructions per Second - мільйон інструкцій в секунду);
- кількість ліній введення/виведення;
- склад периферійних блоків;
- тактова частота, МГц;
- енергоспоживання, мВт/МГц;
- ціна.

Важливість того чи іншого параметра визначається у кожному конкретному випадку індивідуально. Наприклад, перший параметр – продуктивність (кількість виконуваних операцій за одиницю часу).

Точне значення необхідної продуктивності можна визначити, знаючи кількість операцій, потрібну для реалізації алгоритму роботи пристрою за допомогою даного конкретного МК. Визначити це значення можна в тому випадку, якщо однозначно встановлений алгоритм роботи програмного забезпечення МК, а алгоритм може бути однозначно визначений тільки тоді, коли вибраний конкретний МК, розроблена принципова схема, визначені використовувані периферійні вузли і так далі. Виходить замкнуте коло.

Ще складніше розв'язувати задачу вибору МК при розробці принципово нового пристрою, що не має аналогів, оскільки в процесі розробки алгоритм роботи пристрою може зазнавати істотних змін.

Деяко скоротити область вибору можуть обмеження щодо інших, наведених вище критеріям.

Розрахуємо необхідну кількість ліній введення/виведення, а також визначимо необхідні периферійні вузли МК, для реалізації пристрою системи терморегулювання, що управляє.

Для підключення клавіатури використовуються три лінії введення/виведення. Дві лінії використовуються для керування виконавчими механізмами. Ще дві лінії введення/виведення використовуються для відображення стану ТЕН1 і ТЕН2. Одна лінія – для керування звуковою сигналізацією вибору режимів роботи пристрою.

Таким чином, МК повинен мати як мінімум 8 ліній введення/виведення загального призначення. Крім того, для спрощення обміну даними з індикаторами, потрібна наявність інтерфейсу SPI; для переведення значень температури, що надходять з аналогових датчиків температури, у цифровий код – АЦП, а для зберігання встановлених значень температури – незалежна пам'ять даних EEPROM ємністю не менше 4 байтів (по 2 байти для кожного значення температури).

Розрахуємо розрядність АЦП мікроконтролера.

Згідно з ТЗ діапазон вимірюваних температур складає $T_{\Delta} = 300^{\circ}\text{C}$, а точність вимірювання – $\delta T = 1^{\circ}\text{C}$. Для забезпечення заданої точності необхідна кількість рівнів квантування електричного сигналу, відповідного вимірюваній фізичній величині, повинна складати:

$$N_{\text{кв}} = \frac{T_{\Delta}}{\delta T}. \quad (3.5)$$

Тоді необхідна кількість розрядів АЦП:

$$n = E[\log_2(N_{кв})], \quad (3.6)$$

де E – оператор округлення до найближчого більшого цілого.

Для даного випадку згідно з (3.5), (3.6) $n = 9$, отже, для аналого-цифрового перетворення сигналів внутрішній АЦП МК повинен мати розрядність не меншу 9.

Зі всієї різноманітності моделей 8-бітових МК можуть бути вибрані МК фірми Atmel – AVR, Microchip – PIC16, Intel – 80x51 та інші. Оптимальним за приведеними вище критеріями є МК ATmega8 [19, 20, 24]. Його основні характеристики:

- об'єм пам'яті програм – 8 кБ;
- об'єм ОЗП – 1 кБ;
- об'єм EEPROM – 512 байтів;
- кількість ліній введення/виведення – 23;
- вбудований 10-розрядний АЦП (8 вхідних каналів);
- вбудований інтерфейс SPI;
- два 8-бітових і один 16-бітовий таймер/лічильник;
- тактова частота – до 16 МГц;
- напруга живлення – від 2,7 В до 5,5 В.

Розробка принципової схеми. Напруга живлення драйвера МАХ7221 і мікроконтролера складає +5 В, тому аналогічна напруга живлення може бути вибрана для решти блоків системи терморегуляції (див. рис. 3.5).

Оскільки живлення терморегулятора згідно з ТЗ, повинно здійснюватися від мережі змінного струму загального користування $\sim 220\text{В}$, для отримання $U_{жив} = +5\text{В}$ потрібно використовувати вторинне джерело живлення.

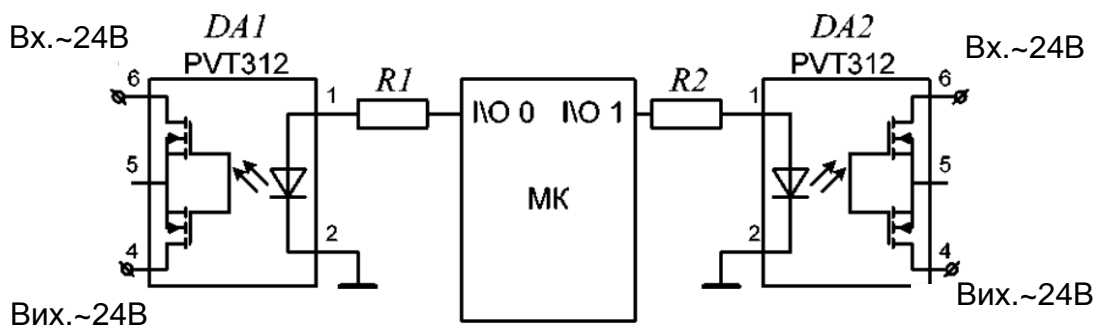


Рисунок 3.11 – Схема підключення оптронів

Як оптрони системи терморегулювання можуть бути вибрані будь-які оптрони для комутації змінного струму не менше 0,1 мА, з комутованою напругою не менше $\sim 24\text{В}$ (діюче значення) і струмом

керування не більше 20 мА (максимальний вихідний струм МК). Наприклад, PVT312 [80]. Схема підключення оптронів до МК наведена на рисунку 3.11.

Опори $R1$ і $R2$ встановлюють величину струму керування оптронів. Розрахуємо їх значення.

Мінімальне значення струму керування оптрона – $I_{кер}^{min} = 2 \text{ мА}$. Для надійного спрацьовування номінальний струм керування оптронів вибирається рівним:

$$I_{кер}^{ном} = K_3 \cdot I_{кер}^{min}, \quad (3.7)$$

де K_3 - коефіцієнт запасу ($K_3 = 1,5 \div 3$). Задамо $K_3 = 2$, тоді згідно (3.7) $I_{кер}^{ном} = 5 \text{ мА}$.

На вольт-амперній характеристиці оптрона, при вибраному струму керування, визначаємо падіння напруги на діоді $U_{VD} = 1,2 \text{ В}$ [80]. Напряга високого логічного рівня на виході МК (при $U_{жив} = +5 \text{ В}$) складає $U_{лог.1}^{min} \geq 3,4 \text{ В}$. Таким чином на обмежувальному опорі $R1$ ($R2$) повинна падати напряга $U_{R1(R2)} = 2,2 \dots 3,8 \text{ В}$. Відповідно $R1$ ($R2$) для забезпечення заданого струму керування повинні знаходитися в межах $R1 = R2 = \frac{U_{R1(R2)}}{I_{кер}^{ном}} = 440 \dots 760 \text{ Ом}$. Виберемо номінальне значення опорів $R1 = R2 = 560 \text{ Ом}$.

Коли одночасно включені обидва тени (тобто, подані сигнали керування на обидва оптрони), то максимальний струм буде дорівнювати:

$$I_{опт}^{max} = 2 \cdot \frac{U_{лог.1}^{max} - U_{VD}}{R} = 2 \cdot \frac{5 - 1,2}{560} = 13,6 \text{ мА}. \quad (3.8)$$

Максимальний струм споживання МК – $I_{МК}^{max} = 100 \text{ мА}$, а максимальний струм споживання драйвера спільно з індикаторами $I_{дрв+інд}^{max} = 500 \text{ мА}$.

Розрахуємо енергетичні показники терморегулятора. У разі, коли одночасно натиснуті всі (три) клавіші, струм споживання клавіатури складає:

$$I_{кл}^{max} = 3 \cdot \frac{U_{жив}}{R} = 3 \cdot \frac{5}{10 \cdot 10^3} = 1,5 \text{ мА} \quad (3.9)$$

Таким чином, максимальний струм, споживаний частиною керуючого пристрою становить:

$$I_{max} = I_{кл}^{max} + I_{опт}^{max} + I_{МК}^{max} + I_{дрв+інд}^{max} = 1,5 + 13,6 + 100 + 500 = 615,1 \text{ мА} \quad (3.10)$$

Принципова схема вторинного джерела живлення наведена на рисунку 3.12. Воно містить знижувальний трансформатор, діодний міст і стабілізатор напруги L7805 або його аналог.

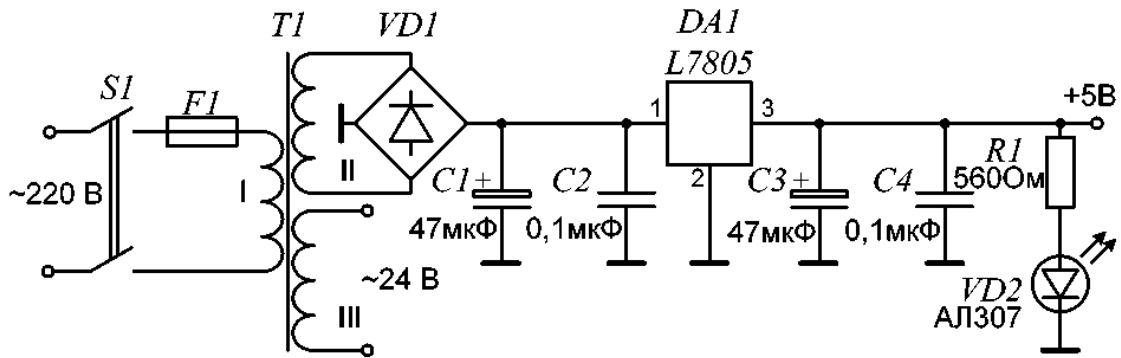


Рисунок 3.12 – Схема джерела живлення терморегулятора

Вторинна обмотка трансформатора II, від якої живиться частина пристрою керування повинна бути розрахована на вихідний струм $I_{вих.II} = K_2 \cdot I_{max} = 2 \cdot 615,1 \text{ mA} = 1230,2 \text{ mA}$ і напругу $U_{вих.II} = \sim 7,5 \text{ V}$ (амплітудне значення). Тобто, вихідна потужність обмотки II трансформатора повинна дорівнювати:

$$P_{вих.II} = U_{вих.II} \cdot I_{вих.II} = 7,5 \text{ V} \cdot 1230,2 \text{ mA} = 9,23 \text{ B} \cdot \text{A}.$$

Обмотка III використовується для керування електромеханічними реле, її вихідна потужність повинна дорівнювати:

$$P_{вих.III} = U_{вих.III} \cdot K_2 \cdot I_{вих.III} = 24 \text{ V} \cdot 2 \cdot 0,2 \text{ mA} = 9,6 \text{ B} \cdot \text{A}.$$

Вибір діодного моста визначається максимальним вихідним струмом вторинної обмотки II трансформатора $I_{вих. II}$. Вибирається SB201 [75], розрахований на максимальний струм 2 А і вхідну напругу 35 В (значення, що діє).

Номінальні значення ємності блокувальних конденсаторів C1–C4 вибрані згідно з рекомендаціями виробника стабілізатора L7805 [77].

Резистор R1 обмежує струм, що протікає через світлодіод VD2, який сигналізує наявність живлення приладу (індикацію вмикання пристрою). На обмежуючому резисторі повинна падати напруга $U_{R1} = U_{жив} - (1,8 \div 1,9) \text{ V}$. При номінальному струмі $I_{VD} = 10 \text{ mA}$ і напрузі живлення $U_{жив} = +5 \text{ V}$, згідно закону Ома, опір $R1 = 310\text{--}320 \text{ Ом}$. Як показує практичний досвід, без помітного зниження яскравості, опір R1 може бути збільшено в 1,5-3 рази (особливо для надяскравих

світлодіодів). З ряду номінальних значень виберемо номінальне значення $R1 = 560 \text{ Ом}$.

Точність аналого-цифрового перетворення визначається не тільки розрядністю АЦП, але і динамічним діапазоном вхідного аналогового сигналу.

Розрахуємо величину рівня квантування сигналу температурного датчика. З таблиці 3.1 при температурі 300°C (верхня межа діапазону вимірюваних температур) величина напруги на виході термопарі рівна $U_{max} = 22,88 \text{ мВ}$. Абсолютна величина напруги, відповідає рівню квантування, при кількості розрядів АЦП n дорівнює:

$$\Delta U = U_{max}/2^n = 22,88 \text{ мВ}/2^{10} = 22,3 \text{ мкВ}.$$

Чим більша ΔU у порівнянні з дисперсією шуму на вході АЦП, тим вища точність вимірювання рівня сигналу (менше позначається вплив шумів на точність вимірювання). Абсолютна величина напруги, що відповідає рівню квантування ΔU , пропорційна величині U_{max} , яка визначає динамічний діапазон вихідних сигналів термодатчиків.

Очевидно, що максимальна точність буде досягнута тоді, коли динамічний діапазон сигналу буде рівний динамічному діапазону АЦП, який визначається величиною опорної напруги. Величина опорної напруги АЦП мікроконтролера АТmega8 складає $U_{жив.} = +5 \text{ В}$.

Для збільшення динамічного діапазону сигналу температурних датчиків системи терморегулювання в кожному каналі вимірювання застосовуються масштабувальні підсилювачі (рис. 3.13).

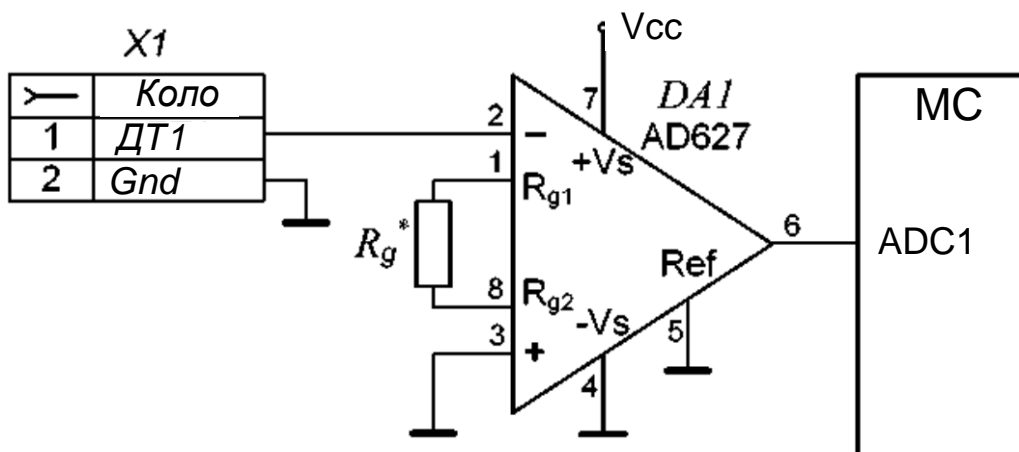


Рисунок 3.13 – Схема вхідного підсилювача терморегулятора
Схема розроблюваного пристрою наведена на рисунках 3.14-3.15.

Для підсилення малих диференціальних сигналів (сигнали з виходу термопар) застосовуються інструментальні підсилювачі AD621. Терміном "інструментальний підсилювач" ("вимірювальний підсилювач") позначають диференціальний підсилювач з зв'язками за постійним струмом, високим коефіцієнтом підсилення, високим вхідним повним опором і великим коефіцієнтом ослаблення синфазного сигналу.

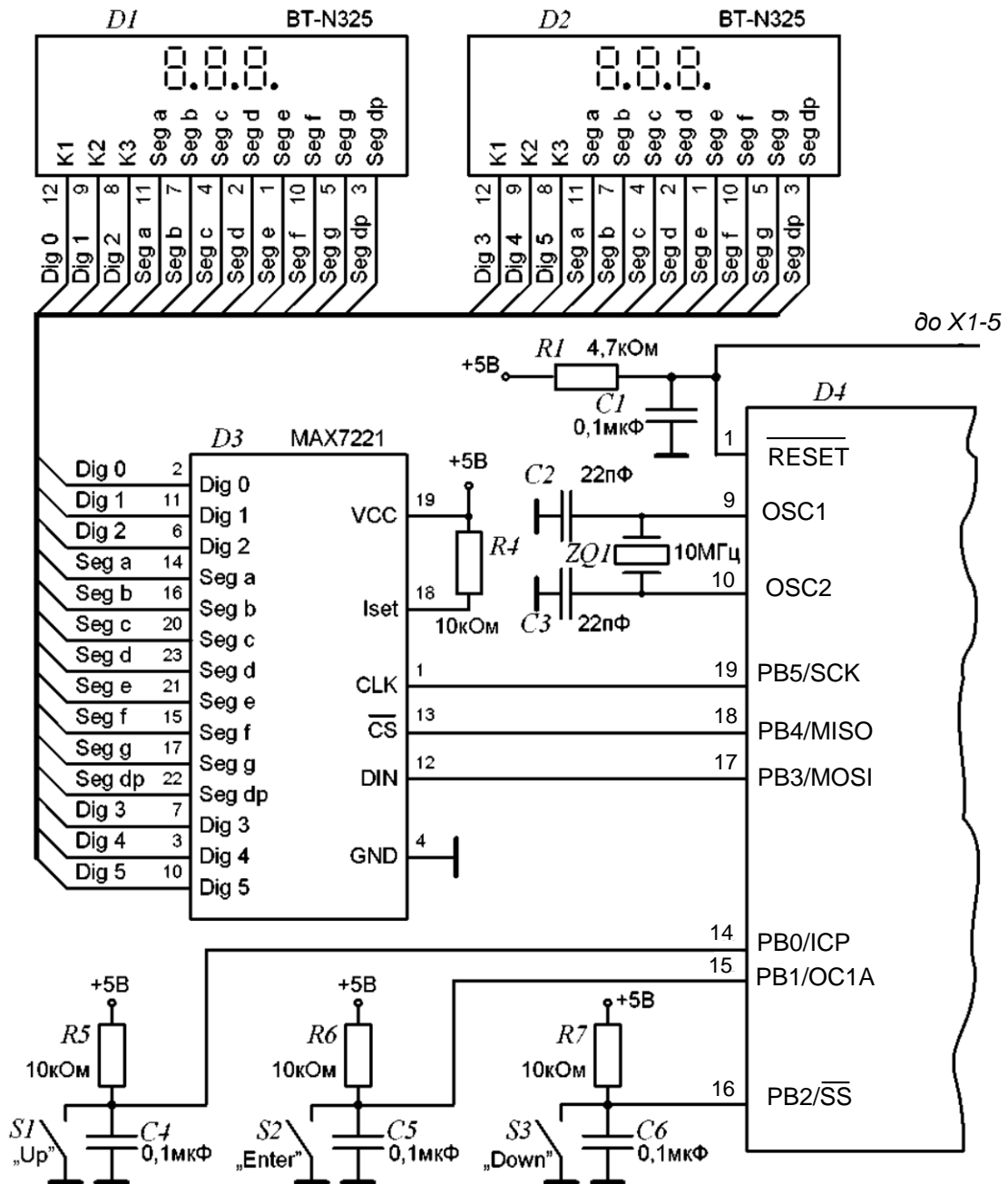


Рисунок 3.14 – Схема електрична принципова терморегулятора

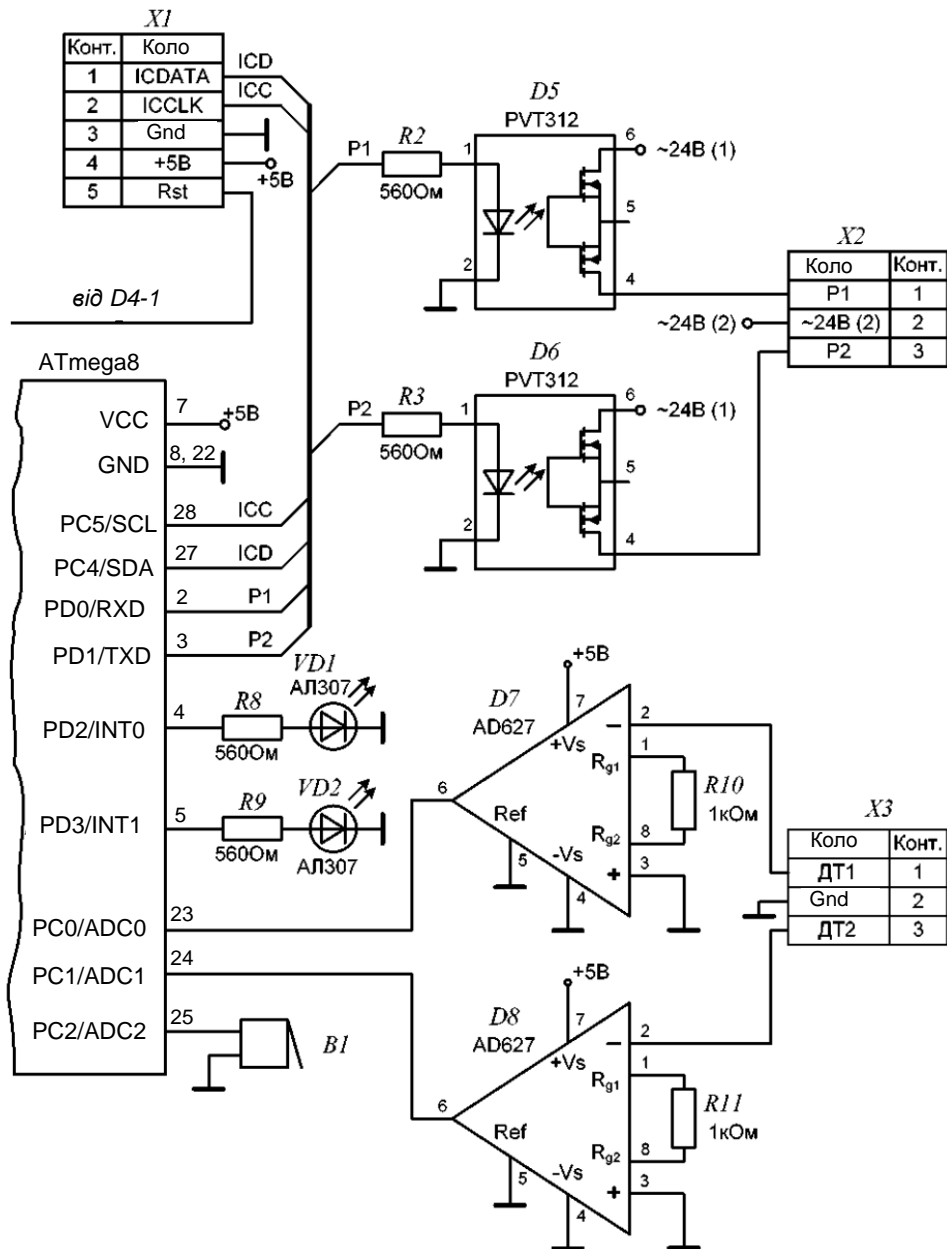


Рисунок 3.15 – Продовження схеми електричної принципової терморегулятора

Необхідний коефіцієнт підсилення визначається співвідношенням:

$$K_{yc} \leq \frac{U_{on}}{U_{max}} = \frac{5B}{22,88 \cdot 10^{-3} B} = 218,5. \quad (3.11)$$

Для AD621 коефіцієнт підсилення задається зовнішнім резистором R_g , опір якого визначається:

$$Rg = \frac{200 \text{кОм}}{K_{yc} - 5} \quad (3.12)$$

Згідно з (3.12) для $K_{yc} = 218,5$, опір $Rg = 937$ Ом. Виберемо найближче номінальне значення $Rg = 1$ кОм. Тоді реальний коефіцієнт посилення складе $K_{yc} = 205$, що задовольняє умові у (3.11).

Для звукової сигналізації входу/виходу до режиму встановлення заданої температури використаний п'єзоелектричний випромінювач *В1*.

Обмежувальні резистори *R8*, *R9* розраховуються так, як і *R1* блоку живлення.

Роз'єм *X1* служить для підключення внутрішньосхемного програматора, *X2*, *X3* – виконавчих реле і датчиків температури, відповідно.

Для забезпечення продуктивності МК, близької до максимальної, вибраний кварцовий резонатор *ZQ1*, розрахований на частоту 10 МГц.

3.4 Розробка програмного забезпечення

Відповідно до логіки роботи, описаної вище, програмне забезпечення МК повинно реалізовувати два основні режими роботи пристрою – режим індикації поточної температури і режим встановлення заданої температури. У обох режимах пристрій повинен виконувати основну функцію – регулювання температури шляхом вмикання/вимикання ТЕН1, ТЕН2.

Програмне забезпечення розробляється з урахуванням системи команд МК ATmega8 мовою Асемблер. На рисунку 3.16 наведена схема алгоритму роботи основної програми.

Після ініціалізації МК з EEPROM зчитуються значення температури $T_{уст1}$ (Т1Н:Т1Л), $T_{уст2}$ (Т2Н:Т2Л), встановленої (заданої) для кожного з каналів у попередньому сеансі роботи.

У підпрограмі обчислення порогових значень (рис. 3.17) обчислюється нижнє значення температури в нагрівальній камері ($T_{вкл.1} = T_{уст1} - \Delta T$), досягши якого відбувається вмикання термоелектронагрівача, і верхнє значення, при якому він вимикається ($T_{вкл.1} = T_{уст1} + \Delta T$). Оскільки встановлені температури для першого і другого каналів можуть відрізнятися, то можуть відрізнятися і порогові значення для ТЕН1 і ТЕН2. Алгоритм обчислення порогових значень другого каналу аналогічний. Величина точності підтримки заданої температури постійна і, згідно з ТЗ, складає $\Delta T = \pm 3$ °С.



Рисунок 3.16 – Схема алгоритму основної програми мікроконтролера

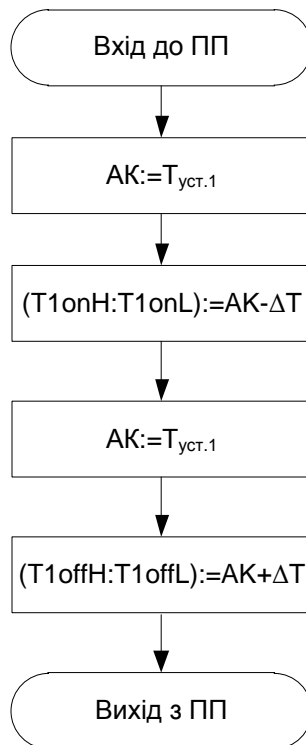


Рисунок 3.17 – Схема алгоритму підпрограми обчислення порогових значень першого каналу

Основна функція пристрою – регулювання температури – буде реалізована з використанням переривання таймера/лічильника. У оброблювачу цього переривання також відобразатиметься поточна температура, тому в основній програмі після встановлення глобального дозволу переривань, здійснюється перехід до опитування клавіатури. Натискання клавіші S2 "Enter" (див. рис. 3.14, 3.15) приводить до переходу в режим встановлення заданої температури. У цьому режимі індикація поточної температури тимчасово припиняється, відображаються значення встановлюваної температури. Після встановлення заданої температури пристрій повертається до опитування клавіші S2 "Enter" і відновлює індикацію поточної температури. Визначимо часовий інтервал між перериваннями таймера/лічильника.

Згідно з теоремою відліків Котельникова (теоремою Найквіста) [4] інтервал дискретизації сигналів ДТ1, ДТ2 повинен складати:

$$T_0 \leq \frac{1}{2F_6}, \quad (3.13)$$

де F_6 – верхня гранична частота в спектрі сигналу.

Швидкість зміни температури в нагрівальній камері, задана в ТЗ, складає від 10 до 15 °С/хв. Отже, верхня гранична частота в

спектрі сигналу температурного датчика буде рівна $Fb = \frac{15}{60} c^{-1} = 0,25 Гц$, а інтервал дискретизації (інтервал, який відраховується таймером), згідно з (3.13) – $T_{\delta} \leq \frac{1}{2 \cdot 0,25 Гц} = 2 c$.

Оскільки в підпрограмі, що обробляє переривання, також повинна відбуватися індикація значень поточної температури, то нижнє значення часового інтервалу визначатиметься частотою оновлення розрядів семисегментного індикатора. Для драйвера MAX7221 частота регенерації зображення розрядів дорівнює:

$$F_{рег} = \frac{8 \cdot f_{osc}}{N}, \quad (3.14)$$

де f_{osc} – частота сканування розрядів MAX7221 ($f_{osc} = 800$ Гц [78]);

N – кількість розрядів індикації.

При необхідних $N = 6$ розрядах індикації, згідно з (3.14):

$$F_{рег} \approx 1067 \text{ Гц} \quad (T_{рег} = \frac{1}{F_{рег}} \approx 0,94 \text{ мс}).$$

Таким чином, період дискретизації сигналів датчиків температури (часовий інтервал, що відраховує таймер) повинен знаходитися в межах

$$0,94 \text{ мс} \leq T_{\delta} \leq 2 c \quad (3.15)$$

Виберемо значення T_{δ} на порядок більше інтервалу регенерації: $T_{\delta} = 10$ мс.

Для формування часового інтервалу може бути використаний один з таймерів: $TC1$ або $TC0$. Скориставшись програмою AVRCalc [76] визначаємо, що при коефіцієнті перерахункової схеми 1024 $TC1H = FFH$, $TC1L = 9FH$, а $TC0 = 9FH$; при коефіцієнті перерахункової схеми 256 $TC1H = FEH$, $TC1L = 7AH$. Вибираємо для створення часового інтервалу таймер $TC1$. Підпрограма обробки переривання за переповненням таймера $TC1$ наведена на рисунку 3.18.

Напруга на виході датчика при температурі 300 °C складає 22,88 мВ (див. табл. 3.1). Коефіцієнт перетворення напруги датчика в температуру K_{np} з урахуванням коефіцієнта підсилення вхідного підсилювача розраховується за формулою:

$$K_{np} = \frac{300^{\circ}C}{K_{yc} \cdot 22,8 \text{ мВ}} = \frac{300^{\circ}C}{205 \cdot 22,8 \text{ мВ}} = 0,064^{\circ}C / \text{мВ} = 64^{\circ}C / B \quad (3.16)$$

У основній програмі при натисканні клавіші S2 "Enter" виконується перехід до меню користувача (рис. 3.19).

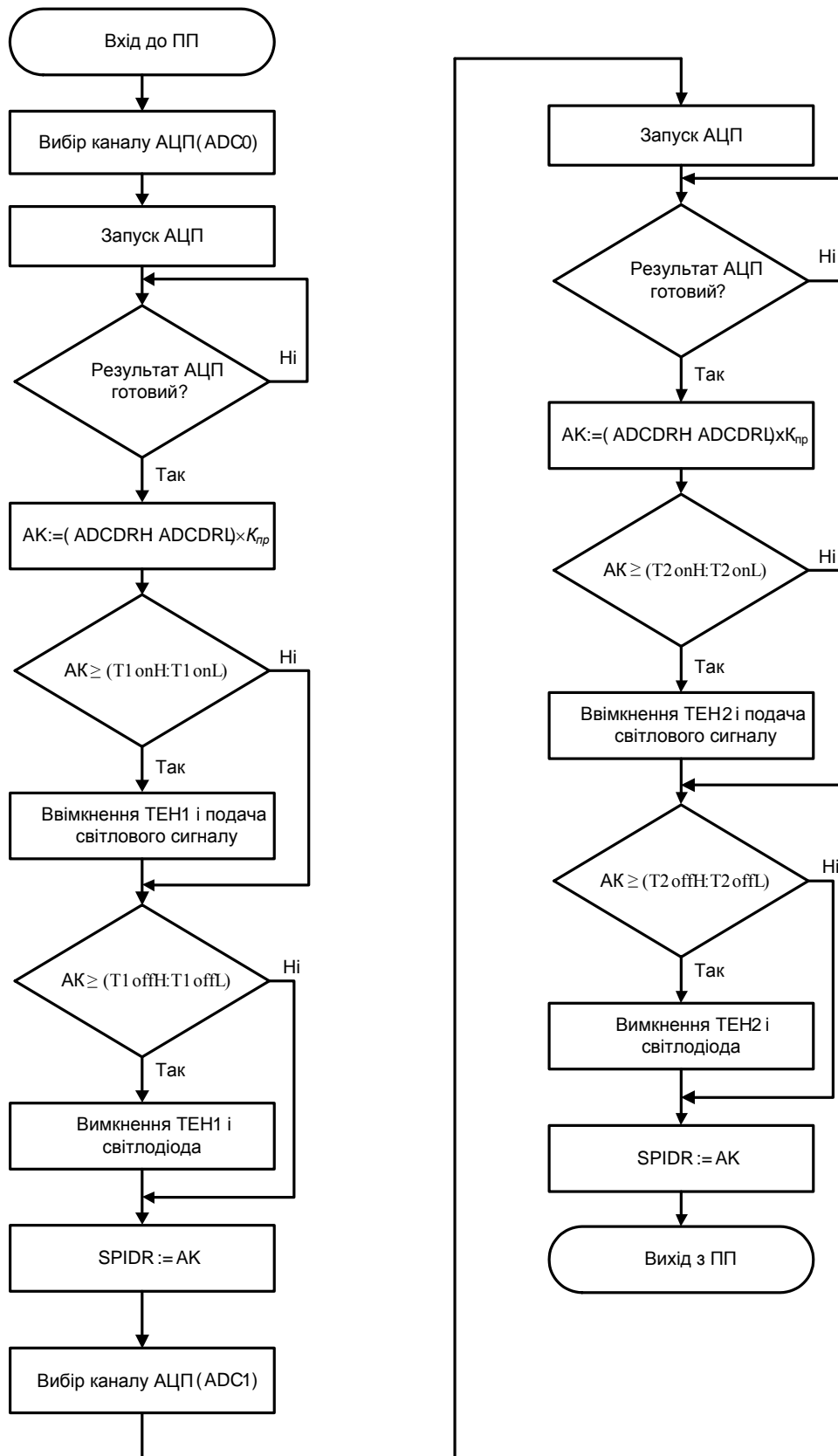


Рисунок 3.18 – Схема алгоритму підпрограми-обробника переривання за переповненням таймера/лічильника ТС1

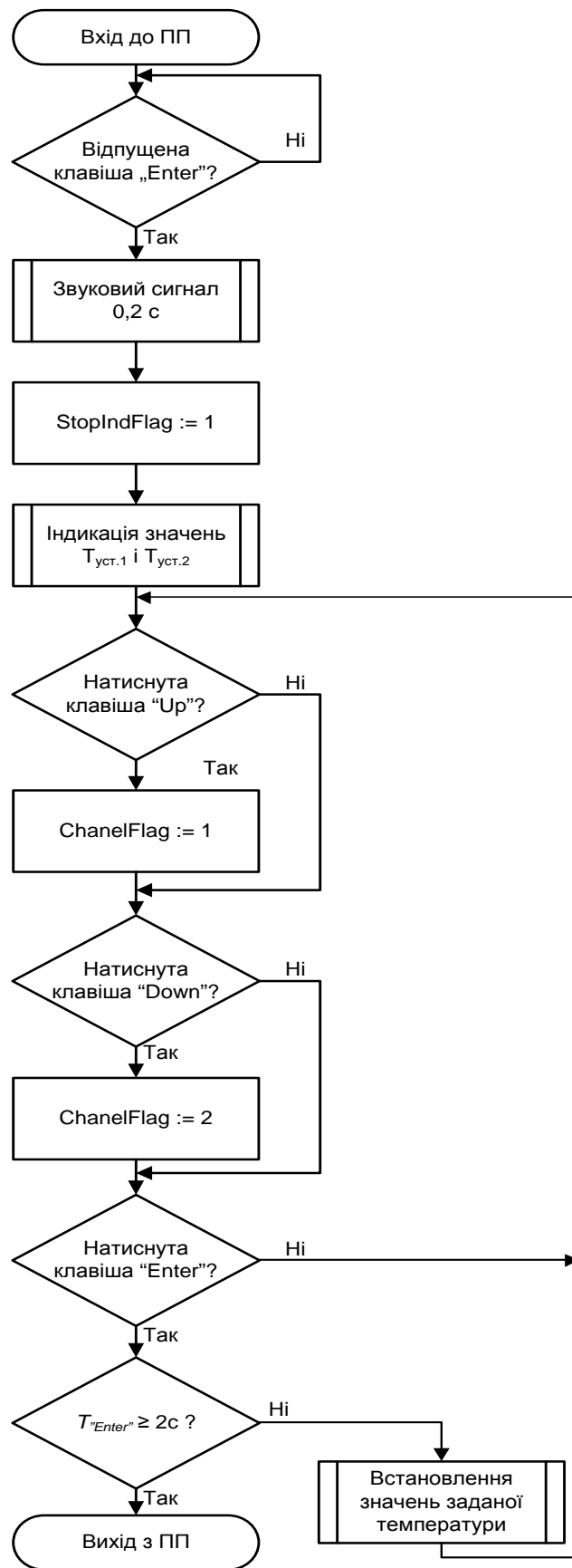


Рисунок 3.19 – Схема алгоритму підпрограми меню користувача

У меню, за допомогою клавiш S1 "Up" i S3 "Down" спочатку вибирається канал, для якого встановлюється температура, а потiм за натисканням клавiшi S2 "Enter" здiйснюється перехiд до режиму встановлення заданої температури вибраного каналу (рис. 3.20).

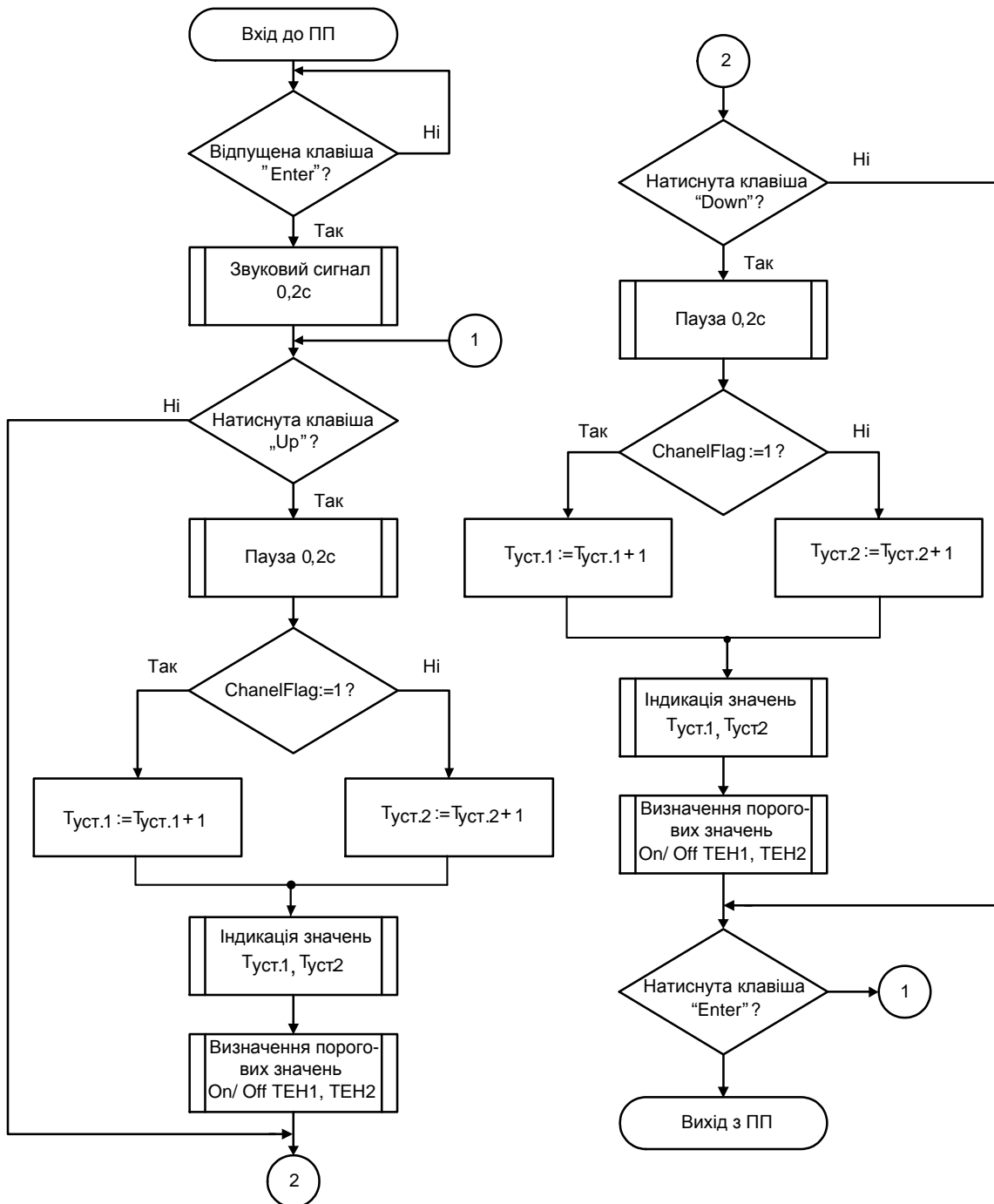


Рисунок 3.20 – Блок-схема пiдпрограми встановлення заданої температури

У цьому режимi iндикацiя поточної температури тимчасово припиняється, вiдображаються значення температури, що встановлюється у першому i другому каналах. Для заборони iндикацiї

поточної температури використовується призначений для користувача Flag.

Встановлення температури вибраного каналу здійснюється за допомогою клавіш S1 "Up" і S3 "Down". При натисненні клавіші S2 "Enter" пристрій повертається в призначене для користувача меню, вихід з якого до основного режиму можливий тільки при утримуванні клавіші протягом 2 с. Цей часовий інтервал формується методом програмних циклів. Перемикання між режимами супроводжується звуковою сигналізацією п'єзозвукознімача В1.

РОЗДІЛ 4 НАЛАГОДЖЕННЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ У СЕРЕДОВИЩІ PROTEUS

У основі автоматизованого схемотехнічного проектування лежить адекватне моделювання електричних процесів, що протікають в схемі при її функціонуванні. Впровадження в інженерну практику методів автоматизації схемотехнічного проектування дозволяє перейти від макетування апаратури до її моделювання на персональному комп'ютері за допомогою відповідного програмного забезпечення.

Існуюче програмне забезпечення для моделювання електричних процесів дозволяє розв'язувати ряд проектних завдань, до яких належать:

- оцінювання принципової електричної схеми, що реалізовується, і дотримання заданих у технічному завданні вимог до вихідних характеристик;
- вибір найкращого варіанта з декількох електричних схем;
- відпрацювання електричної схеми за вихідними характеристиками і електричними режимами роботи радіоелементів.

Системи програм, що призначені для автоматизації проектування радіоелектронних пристроїв (РЕП), можна розділити на дві основні групи: системи схемотехнічного проектування і конструкторського проектування. У останнє десятиліття намітилася явна тенденція інтеграції пакетів програм для проектування схемотехнічного проектування і конструкторського проектування РЕП.

Окрім цих основних груп пакетів програм у системах автоматизованого проектування широко використовуються різні допоміжні пакети: математичні пакети, бази даних, графічні і текстові редактори, електронні таблиці та ін.

У даний час існує велике число пакетів програм, призначених для виконання схемотехнічного проектування РЕП: DesignLab [2], Altera MAX+plus II [53], Orcad [31], Electronic Work Banch (MultiSim) [29], Micro Cap [29, 50], Proteus VSM [40, 49, 69, 70].

Автономне налагодження МПС полягає у налагодженні як самої апаратури, так і програмного забезпечення.

Налагодження апаратури припускає тестування окремих пристроїв МПС (процесора, ОЗП, контролерів, блока живлення, генератора тактових імпульсів) шляхом подачі текстових вхідних дій і знімання реакцій у відповідь. Потім перевіряється їх взаємодія шляхом аналізу сигналів на шинах: адреси, даних і керування.

Налагодження програм МПС проводиться, зазвичай, на тих же персональних комп'ютерах, на яких велася розробка програм, і тією же мовою програмування. Налагодження може бути розпочате на комп'ютері навіть за відсутності апаратури МПС. Для цього в системному програмному забезпеченні повинні знаходитися програми (інтерпретатори і емулятори), що моделюють функції відсутніх апаратурних засобів. Перевірка коректності програм здійснюється тестуванням двома способами: покроковим режимом і трасуванням програм. Засоби налагодження програм мають: керувати виконанням програм, збирати інформацію про хід виконання програми, забезпечувати обмін інформацією (діалог) між програмістом і комп'ютером на рівні мови програмування, моделювати роботу відсутніх апаратурних засобів.

Як правило, МПС – це система реального часу, тобто коректність її функціонування залежить від часу виконання окремих програм і швидкості роботи апаратури. Тому система вважається працездатною, якщо робочі програми правильно функціонують на реальній апаратурі системи в реальних умовах. Тенденція розвитку засобів налагодження МПС полягає в об'єднанні властивостей декількох приладів в одному комплексі, у створенні універсальних засобів, придатних для автономного налагодження апаратури, генерації та автономного налагодження програм і комплексного налагодження системи. У комплексному налагодженні разом з детермінованим використовується статистичне тестування, при якому МПС перевіряється при змінюванні вхідних змінних відповідно до статистичних законів роботи джерел інформації.

Для розробки і налагодження МПС потрібні прилади, які здатні: вимірювати напругу і струм, відтворювати форму сигналу, подавати імпульси певної форми і т. д.; подавати послідовність сигналів одночасно на декілька входів відповідно до заданої часової діаграми або заданого алгоритму функціонування; збирати значення сигналів багатьох ліній протягом одного і того ж проміжку часу, який визначається подіями, що задаються комбінацією або послідовністю сигналів на лініях; обробляти і подавати зібрану інформацію або у вигляді часової діаграми, або у вигляді таблиці логічних станів, або мовою високого рівня. Для автономного налагодження широко використовуються осцилографи, вольтметри, амперметри, частотоміри, генератори імпульсів і кодів, що дозволяють проводити це для апаратури на схемному рівні. Для проведення комплексного налагодження МПС використовують логічні аналізатори, налагоджувальні і діагностичні комплекси.

Однак організація експериментальних досліджень викликає

серйозні ускладнення. Навчальна лабораторія повинна мати сучасне вимірювальне обладнання й кваліфікований персонал, здатний підтримувати його в робочому стані. Якщо навчальному закладу утримувати таку лабораторію складно, то розв'язання подібних задач індивідуальним користувачем неможливо.

Застосування персонального комп'ютера створює альтернативу навчальній лабораторії – віртуальну лабораторію, яка, є по суті, програмою числового розрахунку схем з інтерфейсом, який імітує діяльність дослідника в реальній лабораторії. Розширити коло потенційних користувачів можна, якщо процес моделювання максимально наближений до реальності. У цьому випадку користувач здійснює природну послідовність таких дій, як складання схеми, підключення вимірювальних приладів, установлення режимів роботи вимірювальних приладів, отримання режимів роботи в звичній для нього формі. Таку можливість надає програма Proteus VSM [36, 40, 43, 69, 70].

На відміну від багатьох інших ця програма здатна моделювати пристрої не тільки на дискретних компонентах, звичайних аналогових і цифрових мікросхемах, але і на МК серії AVR, 8051, PIC12, PIC16, PIC18, Z80, 68000. Програма містить велику кількість бібліотек напівпровідникових пристроїв, пасивних компонентів, ламп, індикаторів (світлодіоди, семисегментні, рідинно-кристаличні), кнопок, клавіатур, динаміків, мікрофонів, джерел струму, напруги; генератори спеціальних сигналів, серво- та крокові двигуни.

Як відомо, налагодження МПС традиційними засобами, за допомогою лише вольтметра і осцилографа, є складним. Порівняно легко вдається лише встановити, що в програмі є помилка. Але практично неможливо знайти серед множини однаково правдоподібних пояснень "неадекватної" поведінки МК єдине правильне. Тому налагодження ведуть, зазвичай, методом спроб і помилок, вносячи до програми більш-менш обґрунтовані зміни. Емулятор дає можливість "зазирнути всередину" МК, зіставивши форму і характер сигналів на його виводах з ходом виконання програми і змінами стану внутрішніх регістрів. До цих регістрів моделі (на відміну від реальної мікросхеми) завжди є доступ. Можливість перевіряти роботу МК всіх популярних сімейств у реальному масштабі часу і у взаємодії з моделями реальних джерел сигналу і навантажень вигідно відрізняє Proteus VSM від простих симуляторів, що є в системах розробки програм і часто дозволяють лише стежити за ходом покрокового виконання програми.

Пакет Proteus складається з двох програм: ISIS – моделювання електронних схем та ARES – програма створення друкованих плат. У

ISIS передбачене покрокове налагодження МПС, можливість анімації елементів схем.

Розглянемо налагодження пристрою на МК AVR AT90S8515 та з інтерфейсом 1-Wire, яким підключаються DS1990, DS18S20 та інші 1-Wire прилади [6, 7, 9, 12]. Запускаємо PROTEUS і у меню *File* → *Load Design...* перейдемо на один рівень нагору – у папку, де встановлений PROTEUS, і відкриваємо папку *SAMPLES*. У ній знаходимо і відкриваємо файл проекту *One-Wire* → *NETWORK* → *1WIRE_NET.DSN* (рис. 4.1).

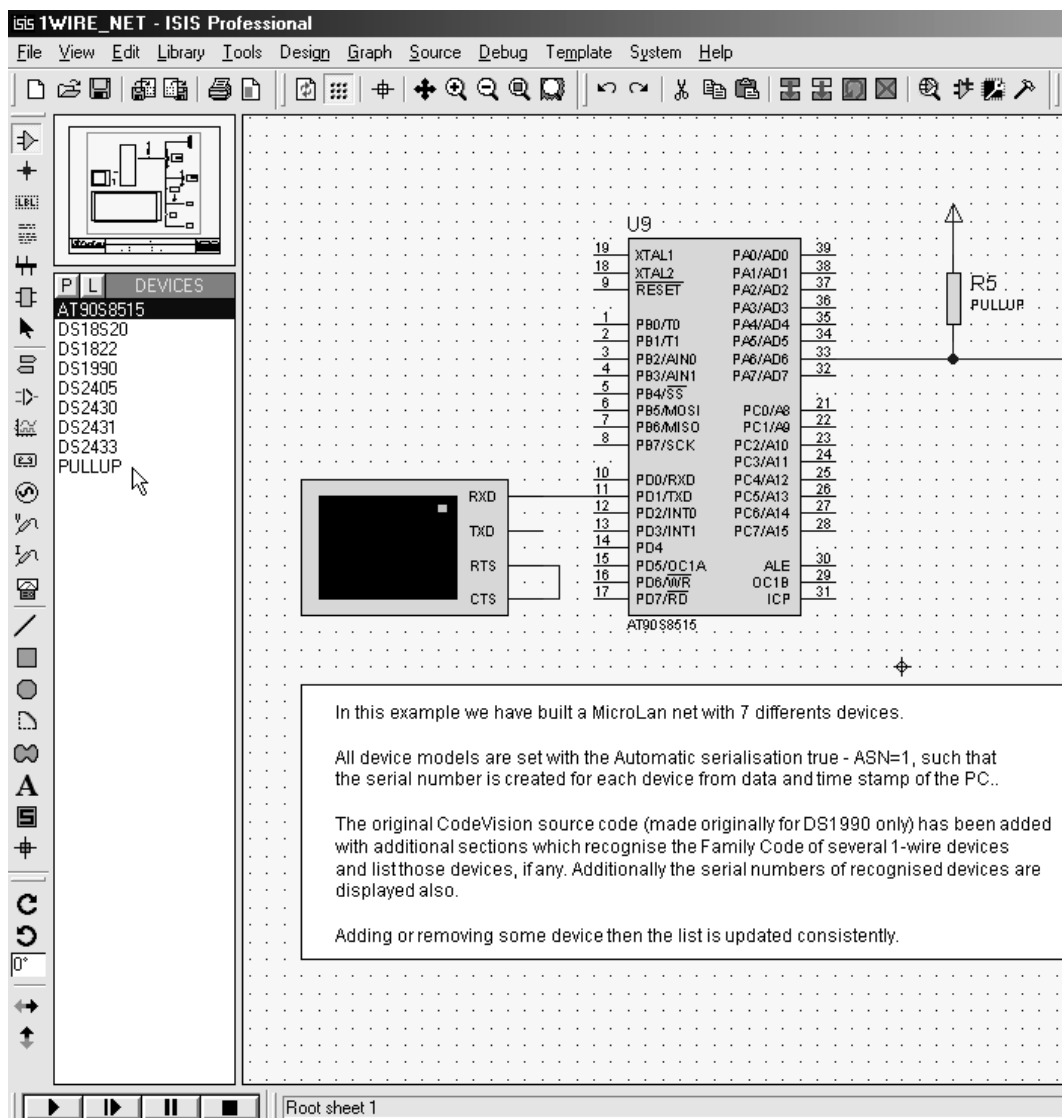




Рисунок 4.1 – Вікно проекту 1WIRE_NET у середовищі Proteus

Інтерфейс PROTEUS не має смуг прокручування. Масштаб зображення можна змінювати за допомогою колеса на миші або за допомогою інструментів "Лупа"  та  у верхній панелі

інструментів. *Переміщати зображення* можна вказанням нового центра зображення інструментом "Хрест" ✚. *Весь робочий аркуш* можна побачити, натиснувши кнопку 📏. *Показати певну область схеми* можна виділивши її за допомогою інструмента виділення області 📐.

У лівій верхній області екрана відображається міні макет сторінки й трохи нижче – панель DEVICES (компоненти проекту) (рис. 4.2).

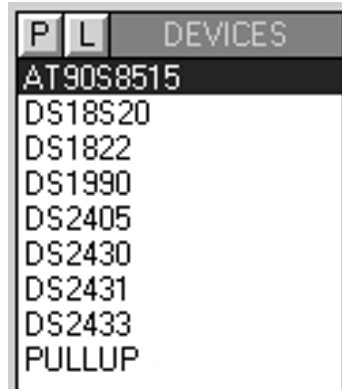


Рисунок 4.2 – Панель DEVICES

У цьому вікні відображаються всі елементи, що використовуються в схемі. Кнопка з буквою "P" відкриває форму пошуку компонента в бібліотеках PROTEUS для додавання в схему. Кнопка з буквою "L" відкриває менеджер бібліотек – з його допомогою можна підключати нові бібліотеки компонентів. На рисунку 4.2 бачимо МК AVR AT90S8515, кілька компонентів компанії DALLAS-MAXIM (вони підключені за схемою інтерфейсу 1-Wire). Внизу списку є компонент PULLUP – це резистор, що підтягує напругу до + живлення МК і приладів DSXXXX (звичайно це +5 В).

Натисніть кнопку з буквою "P". Відкриється форма-меню пошуку і вибору компонентів "Pick Devices" (рис. 4.3). У полі "Keywords" (ключові слова) введіть DS2 і виберіть верхній з 4-х знайдених компонентів – DS2405. У полі "Description" (опис) спостерігаємо "адресовий перемикач" – тобто до цього приладу можна звернутись за його адресою і «наказати» йому зробити на виході РІО логічний "1" або "0" і можна перевести вихід у високоімпедансний Z-стан – вивід з дуже великим опором, що не впливає на те, що до нього підключено. У правій частині форми можна побачити назву моделі компонента, його зображення на схемі, а нижче нього "FootPrint" – його корпус. Ще нижче назва корпусу компонента – TO92.

Інша частина 1-Wire приладів на схемі починається на DS1 (рис. 4.4) – введіть ці символи у поле ключових слів. Тепер знайдено 8 приладів. Причому вони розташовані в 2-х категоріях. Вибираємо

мишкою DS18S20 – в описі написано: "Точний 1-Wire цифровий термометр". На схемному зображенні (рис. 4.5) цього компоненту видно деяке поле, що нагадує дисплей, – так і є, у ньому будуть виводиться дані в процесі симуляції.

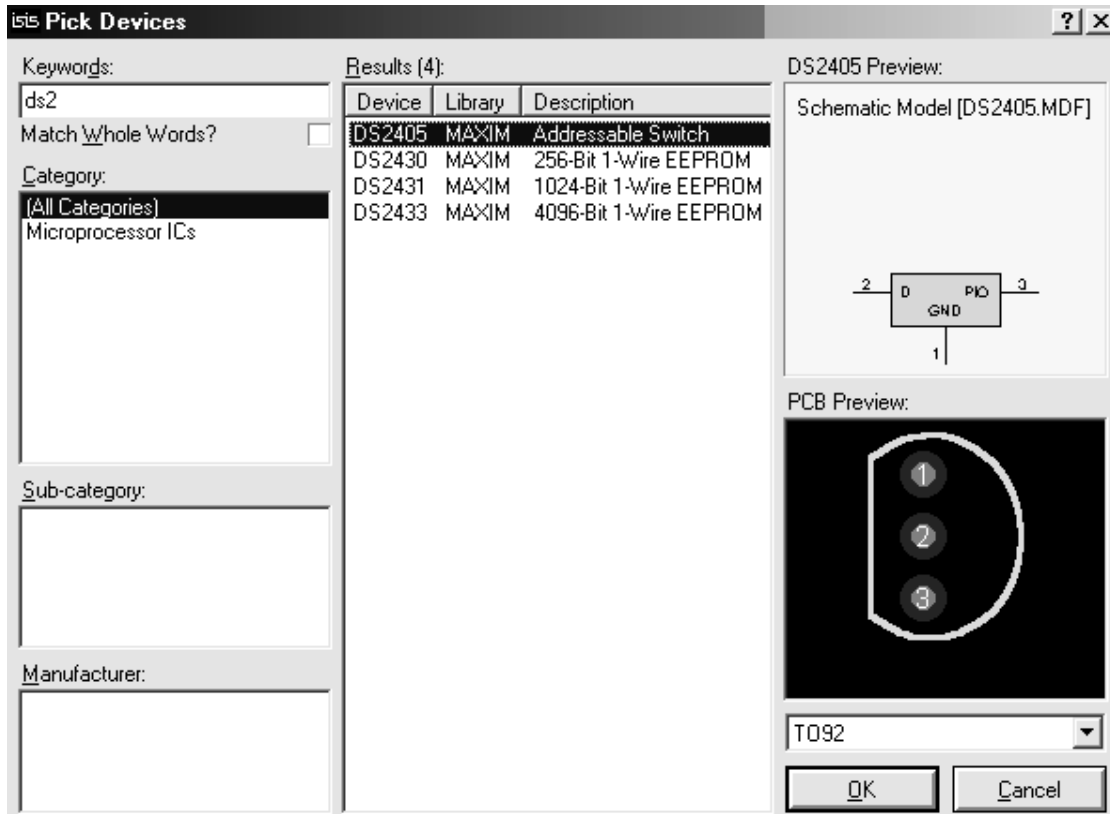


Рисунок 4.3 – Форма-меню пошуку і вибору компонентів "Pick Devices". Введений ключ пошуку DS2

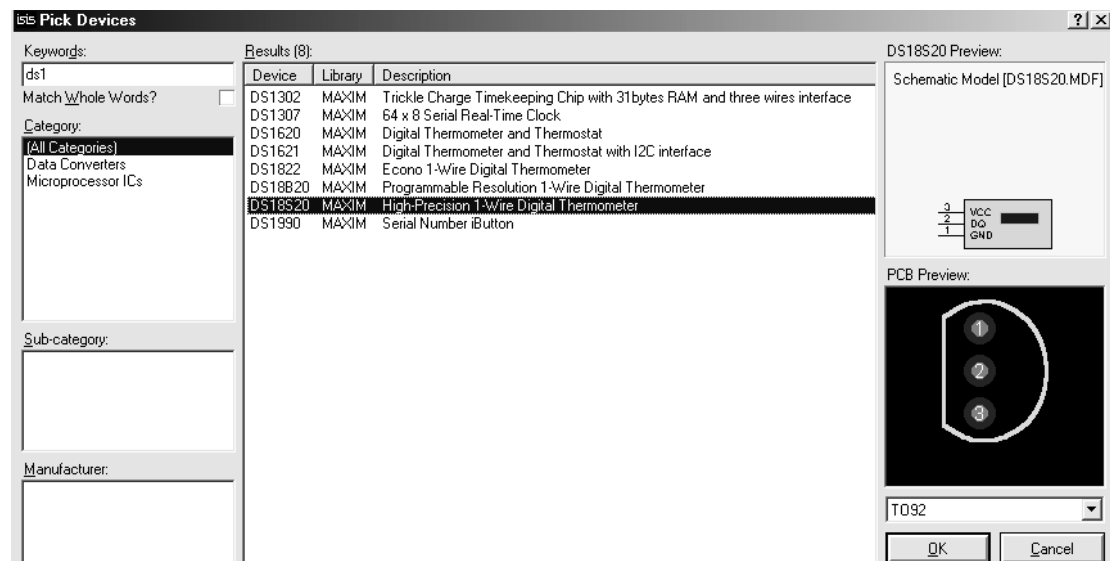
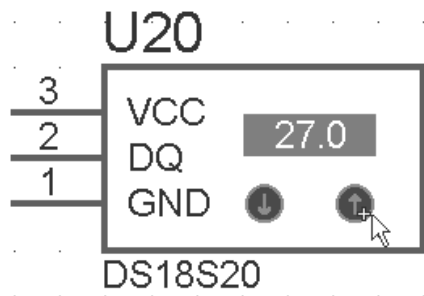


Рисунок 4.4 – Форма-меню пошуку і вибору компонентів "Pick Devices". Введений ключ пошуку DS1



Стрілками у ході симуляції можна міняти температуру корпуса датчика – натиснувши на них мишею.

VCC – + живлення датчика,
GND – "земля",
DQ – лінія даних

Рисунок 4.5 – Умовне позначення датчика DS18S20 у вікні проекту 1-WIRE_NET у середовищі Proteus










Помістити компонент на схему можна, натиснувши "ОК", потім помістите покажчик миші у потрібне місце на листі схеми і натиснути мишею. Компонент опиниться на схемі.

Компонент на схему можна додати і так – виділити, скопіювати і вставити, тільки в цьому випадку доведеться вручну надати йому порядковий номер замість знака питання – "?". Виділити компонент або будь-який елемент схеми можна, натиснувши на ньому правою кнопкою миші. При цьому він стане червоним і одночасно будуть виділені всі провідники, що надходять до компонента. Відмінити виділення всіх виділених компонентів можна, натиснувши правою кнопкою миші в порожньому місці схеми. Видалити компонент або будь-який елемент з схеми можна, двічі натиснувши правою кнопкою миші на ньому.






Таблиця 4.1 – Ліва панель інструментів (верхня частина)

Умовне позначення	Назва	Призначення
	L1	відкрити панель DEVICES - компоненти проекту і пошук нових
	L2	поставити точку з'єднання провідників уручну
	L3	дати назву дроту (дроти, що мають однакову назву, електрично з'єднані)
	L4	додати текст в довільне місце схеми
	L5	прокласти шину
	L6	створення підсхеми (блок містить в собі свою схему і з'єднувачі)
	L7	редагування властивостей компонента

Таблиця 4.2 – Ліва панель інструментів (середня частина)

Умовне позначення	Назва	Призначення
	L8	TERMINALS – живлення, земля, міжблокові з'єднання, виводи
	L9	додати вивід до компонента, що створюється
	L10	графічне відображення, збереження і аналіз результатів симуляції
	L11	"магнітофон" для запису у файл і відтворення даних
	L12	генератори будь-якої напруги, струмів та виведення їх з файлів даних
	L13	вказати точку вимірювання напруги на провіднику
	L14	вказати точку вимірювання струму на провіднику
	L15	Virtual Instruments – вимірювальні прилади
	L16	прокладання провідників на схемі

Таблиця 4.3 – Ліва панель інструментів (нижня частина)

Умовне позначення	Призначення
	обертання на 90 градусів виділеного компонента на схемі за годинниковою стрілкою
	обертання на 90 градусів виділеного компонента на схемі проти годинникової стрілки
	поле для вказання довільного кута повороту
	дзеркальний переворот компонента по вертикалі
	дзеркальний переворот компонента по горизонталі

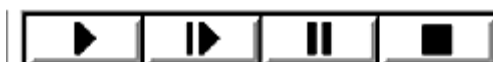


Рисунок 4.6 – Панель керування симуляцією

Панель керування симуляцією (рис. 4.6) складається з 4 кнопок (зліва направо): "Пуск" – запуск симуляції або продовження припиненої симуляції; "Крок" – виконати мінімальний крок за програмою МК, звичайно, це одна інструкція на асемблері (цією кнопкою теж можна почати симуляцію); "Пауза" – пауза симуляції

(можна продовжити кнопками "Пуск" або "Крок"); "Стоп" – зупинка симуляції.

Виділіть МК (див. рис. 4.1), натиснувши на ньому правою кнопкою миші. МК і підключені до нього провідники стануть червоними.

Відкрийте панель редагування властивостей компоненту (*Edit Component*) натиснувши на виділеному МК лівою кнопкою миші. Натисніть кнопку "Hidden Pins" (приховані виводи) – відкриється додаткове меню (рис. 4.7), в якому показано як за замовчуванням називаються вузли (провідники) схеми, до яких підключено живлення МК: VCC і GND. Зміна цих назв інколи необхідна при живленні МК або інших компонентів різною напругою або від різних джерел.

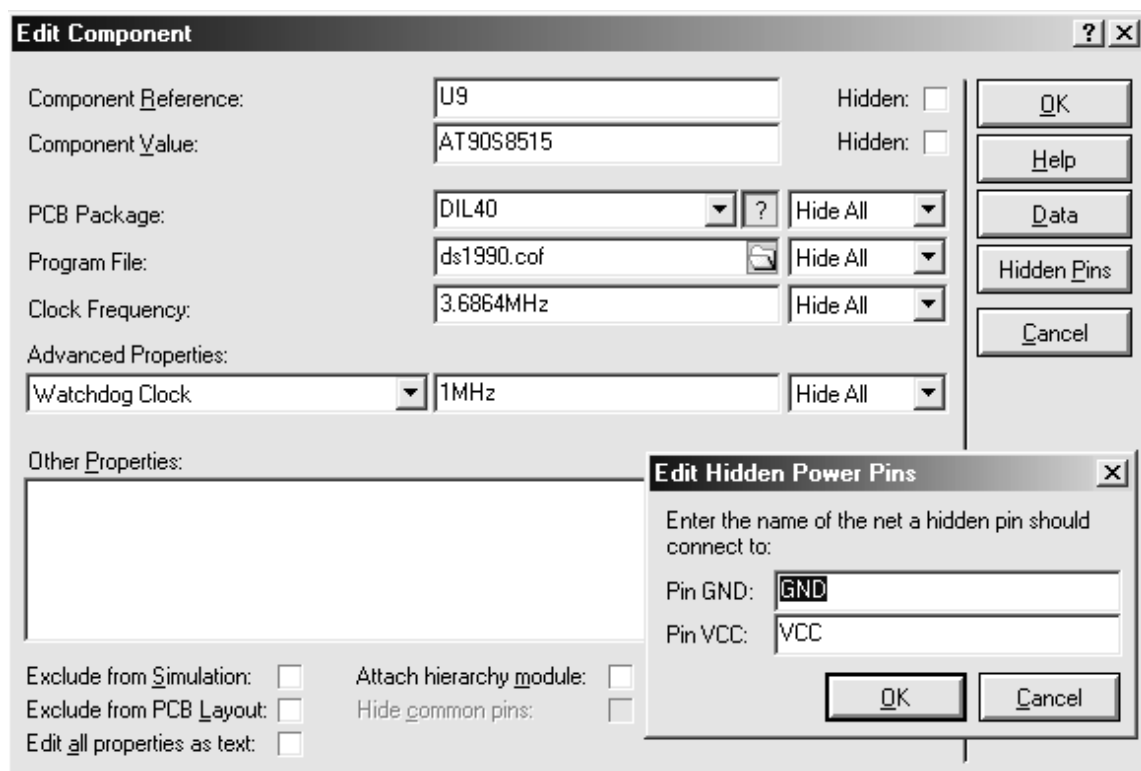


Рисунок 4.7 – Панель редагування властивостей компонента

Натисніть "ОК" щоб закрити панель прихованих виводів і подивіться уважно на вміст панелі редагування компонента. Головне для МК – це програма за якою він працює, тому у полі "*Program File*" потрібно вказати: *.cof* – файл, якщо ведеться налагодження програми (початкова програма) мовою C; *.hex* – файл "прошивки", якщо немає ніякої початкової програми ні на C, ні на Асемблері. Якщо є початкова програма на Асемблері *.asm*, то потрібно вказати назву *.hex* файлу і через меню: *Source* → *Add/remove Source File...* додати назву файлу з текстом програми на асемблері і вибрати потрібний асемблер (рис. 4.8).

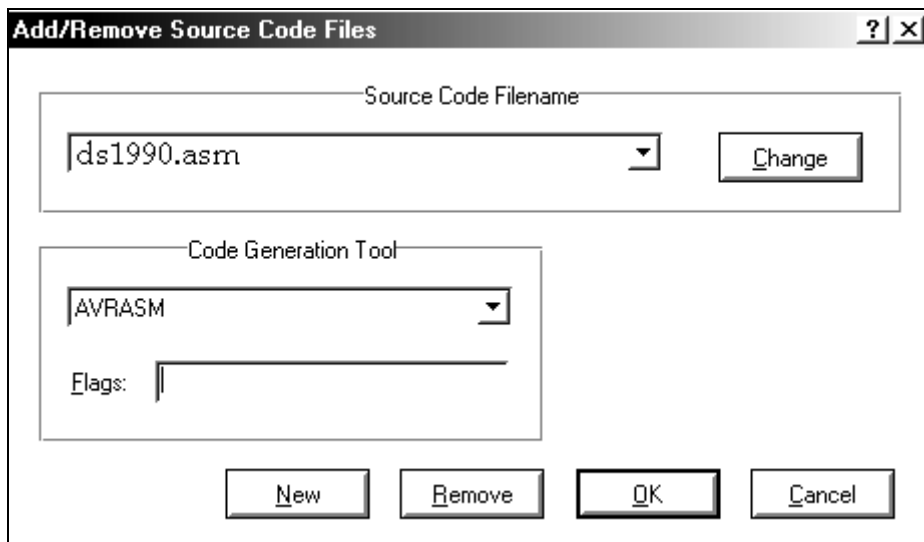


Рисунок 4.8 – Вікно Add/remove Source File

Щоб симулювати у Proteus роботу МК, необхідно знайти його в бібліотеках і помістити на схему, вказати яку програму він повинен виконувати, вказати тактову частоту МК.

Щоб побачити всі установки компонента в панелі редагування (див. рис. 4.7) натисніть *Edit All Prop. As Text* (рис. 4.9)

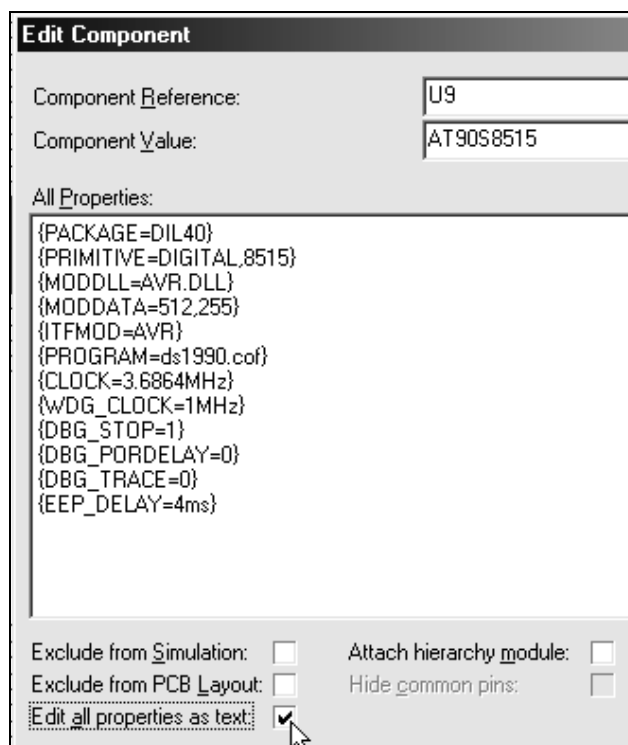


Рисунок 4.9 – Редагування властивостей МК у текстовому режимі

{CLOCK} – однозначно визначає тактову частоту МК при симуляції. Кварц і конденсатори не потрібні для симуляції, їх

встановлюють на схему тільки для того, щоб врахувати при трасуванні друкованої плати пристрою.

{Wdg_clock} – показує частоту роботи генератора сторожового таймера. Хоча вона і позначена як 1MHz – сторожовий таймер не включений, поки не додамо у властивості МК рядок: {Wdgon=1} (для даного проекту не потрібно її вносити).

{Program} – показує, що МК працюватиме за програмою ds1990.cof (створено компілятором CVAVR).

Значення інших параметрів можна дізнатися в довідці за допомогою кнопки Help.

Керувати процесом симуляції програми на асемблері можна за допомогою клавіш F10, F11, F12 і їх комбінацією з клавішами Alt і Ctrl і через меню *DEBUG* (рис. 4.10).

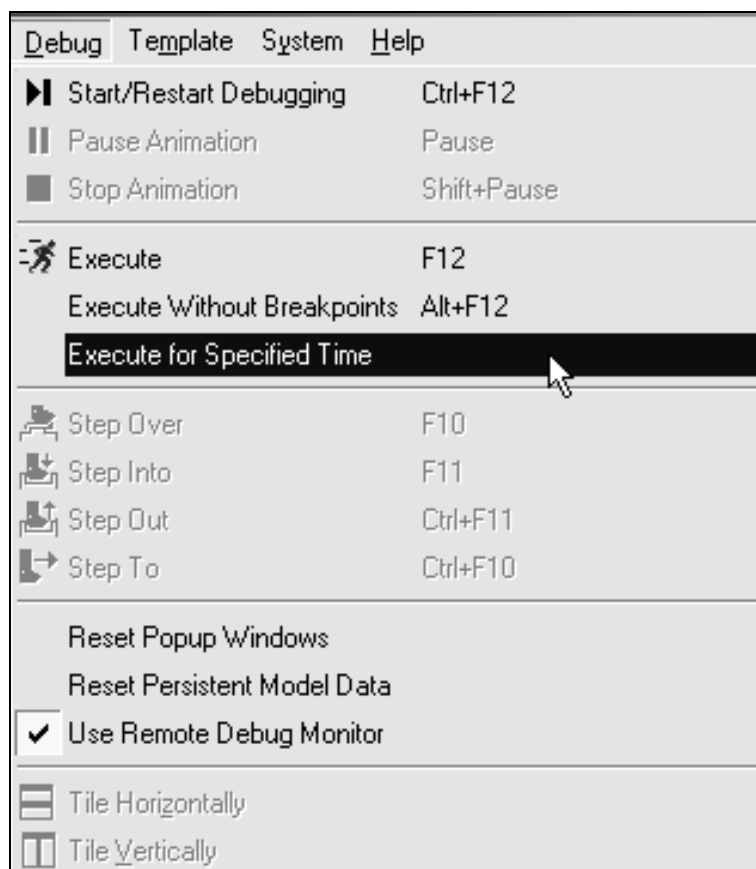


Рисунок 4.10 – Меню *DEBUG*

При виборі команди *Execute for Specified Time* виводиться поле введення для вказівки скільки часу повинна відпрацювати схема (n – нано секунди, μ – мікро секунди, m – милі секунди, секунди можна не позначати). Вказаний час буде використовуватись, якщо до його закінчення в програмі МК не зустрінеться точка зупинки (BP – Breakpoint) – симуляція зупиниться на ній.

Для меню *DEBUG* корисні такі команди:

- F12 еквівалентна кнопці "Пуск" – запускає або продовжує симуляцію до ВР;
- Alt+F12 запускає або продовжує симуляцію, що не зупиняється на ВР;
- F10 – симулювати, не використовуючи процедури на асемблері або функцій на С (вони будуть виконані як 1 крок);
- F11 – симулювати крок за програмою з входом у процедуру або функцію;
- Ctrl+F11 – закінчити процедуру або функцію, в якій знаходимося;
- Ctrl+F10 – симулювати до виділеного мишею рядка в програмі на асемблері.

Допомога щодо симуляції МК знаходиться в розділі *Proteus VSM Help* → SOURCE LEVEL DEBUGGING WITHIN PROTEUS VSM.

При симуляції пристрою на екрані спостерігаємо анімаційну картинку. Режим анімації можна встановити через меню *System* → *Set Animation Options...*(рис. 4.11).

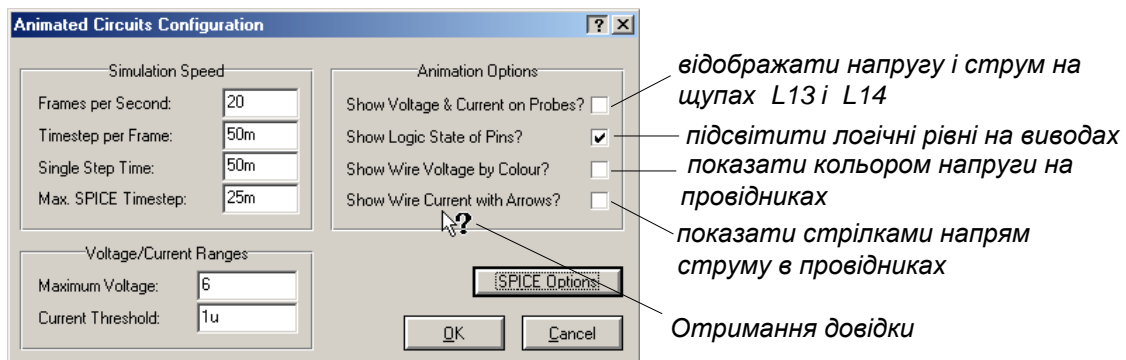


Рисунок 4.11 – Установлення опцій анімації

Запускаємо симуляцію проекту 1WIRE_NET.DSN кнопкою "Пуск" (див. рис. 4.6). З'явиться вікно віртуального термінала ПК (рис. 4.12), і приблизно за 1 секунду програма МК зробить все, що від неї було потрібно. При анімації логічні рівні на виводах указуються кольоровими квадратами: червоний – логічний рівень "1", синій – логічний рівень "0", сірий – Z стан.

У вікні віртуального термінала (рис. 4.12) виводиться інформація з UART МК на віртуальний COM-порт ПК списку розпізнаних пристроїв на шині 1-Wire і їх номери: серійний номер DS1990, серійні номери приладів DSXXXX, що надаються на заводі і незмінні (у Proteus ці номери задаються у вікні редагування властивостей компонента).

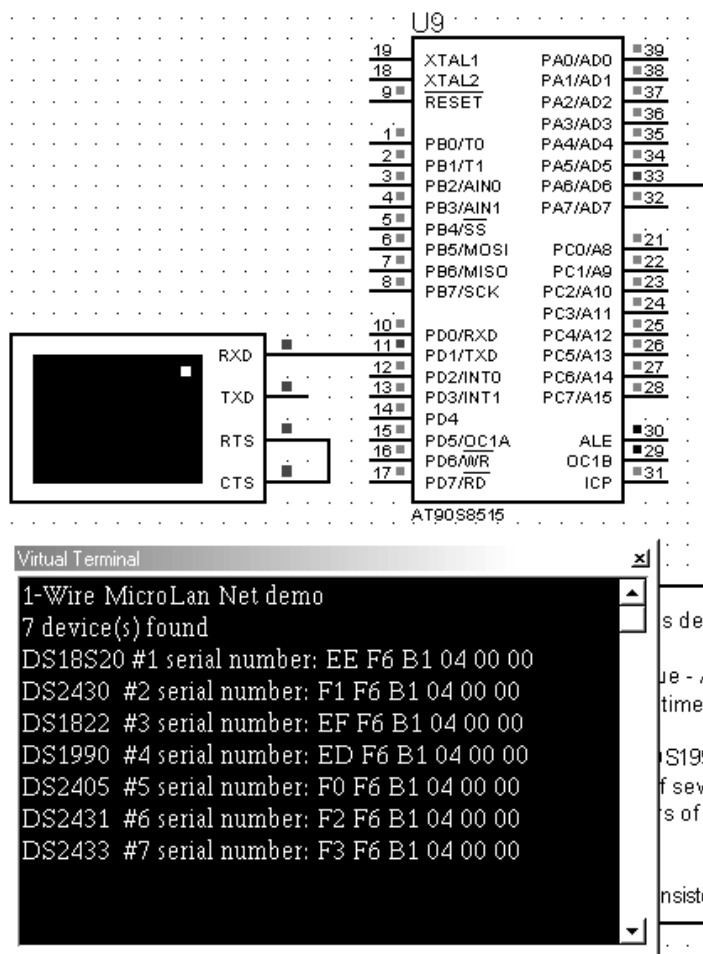


Рисунок 4.12 – Вікно симуляції проекту 1WIRE_NET.DSN. Відкрито вікно Virtual Terminal

Термінал (Virtual Terminal) знаходиться під кнопкою L15 (див. табл. 4.2). У Proteus можна не використовувати узгоджувач рівнів RS232 з UART типу MAX232, а підключати термінал безпосередньо до МК. Термінал підтримує службові символи ASCII – керування виведенням тексту: CR (0DH), BS (0X08H) і BEL (0X07H). LF (0X0AH) і решта службових кодів не підтримуються. У режимі паузи симуляції можна за натисканням правої кнопки миші на екрані терміналу відкрити меню, в якому можна скопіювати, вставити інформацію, можна змінити формат виведення числа (символи або шістнадцяковий код), можна очистити екран даних, можна встановити режим "луна" (прийняті на RXD терміналу символи будуть виводитися на його ніжку TXD).

Виводи RTS і CTS можна не з'єднувати і не підключати взагалі – хоча вони працюють як у справжньому COM-порту ПК. Термінал може передавати щось тільки при рівні "0" на виводі CTS. Можна використовувати декілька незалежних терміналів.

Якщо натиснути мишею у вікні терміналу, то він почне

передавати символи, що їх набирають на клавіатурі ПК, на ніжку TXD. Можна вставити з буфера обміну інформацію, яку термінал так само виводитиме на ніжку TXD. Передавана інформація не відображається у вікні термінала. Курсор залишається на місці. У властивостях термінала в полі додаткові властивості (*Other Properties*) можна ввести текст, який посилатиметься з термінала при натисканні кнопки "Пуск" із затримкою на час передачі одного символу.

Компонент *СOMPIM* дозволяє віртуальному пристрою підключитися до реального *СOM*-порту ПК.

Натисніть "Стоп", потім "Старт" і ... "Пауза". На екрані монітора будуть відображатись вікна з налагоджувальною інформацією, перелік цих вікон задається у меню *DEBUG* (рис. 4.13).

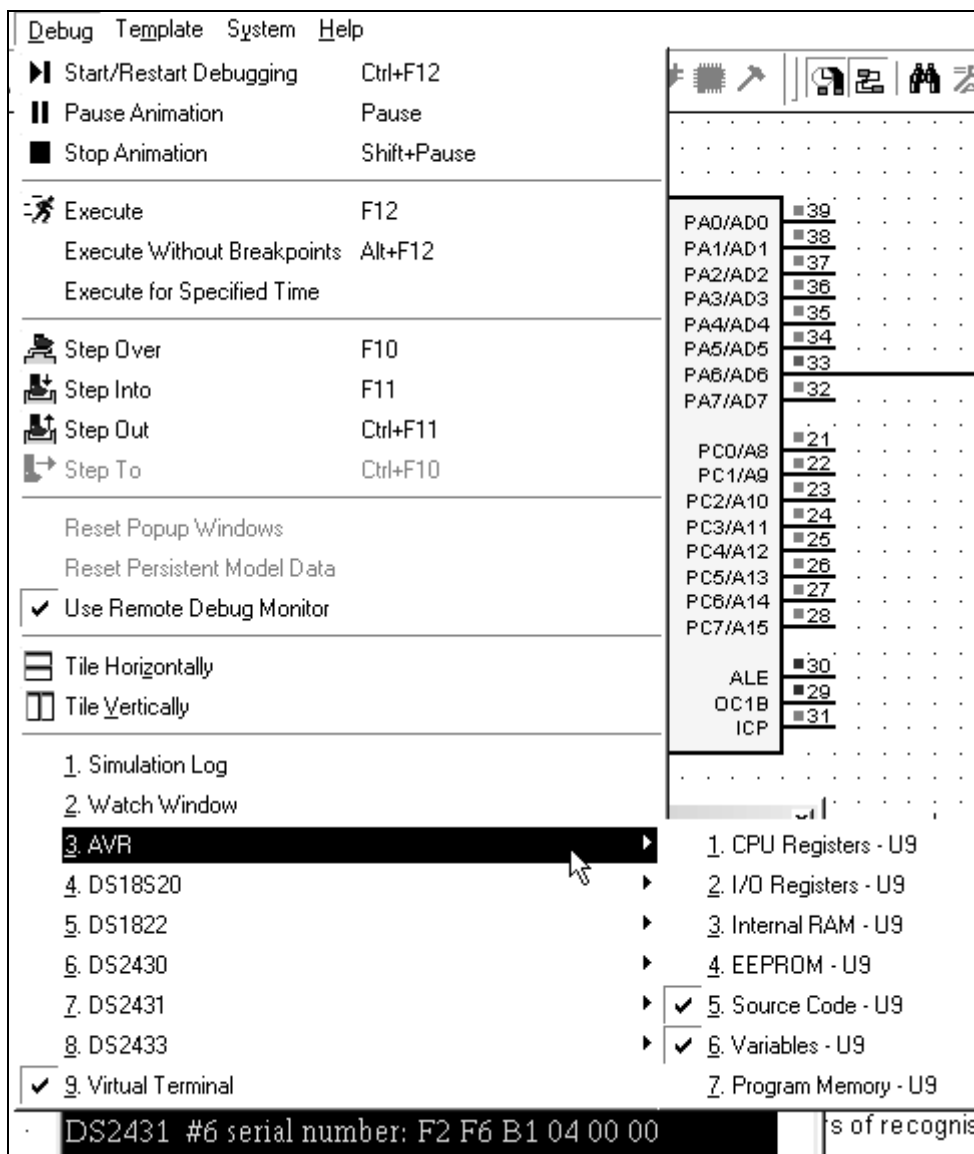


Рисунок 4.13 – Вибір коду програми та змінних у меню *DEBUG* для відображення їх під час налагодження роботи програми

Інші елементи схеми, у яких є цифрові дані або пам'ять, мають свої підменю з можливістю виведення інформаційних вікон. Наприклад, цифровий термометр DS18S20 має Scratch RAM і EEPROM. Якщо в схемі декілька МК, то для кожного буде своє підменю.

Watch Window – вікно стеження (рис. 4.14). Після налагоджування воно не зникає з екрана при симуляції і знаходженні в паузі. У цьому вікні можна розмістити регістри МК і не тільки відстежувати їх вміст за ходом програми, але й задавати деякі умови та дії, при досягненні яких, можна, наприклад, зупинити симуляцію. При виборі *Add Items (By Address)...* з'являється панель спостереження. Натиснувши на ній правою кнопкою миші з'являється меню для додавання регістрів, що будуть спостерігатись (за ім'ям або за їх адресою). Подвійний натиск мишею на назві регістра додає його у вікно спостереження.

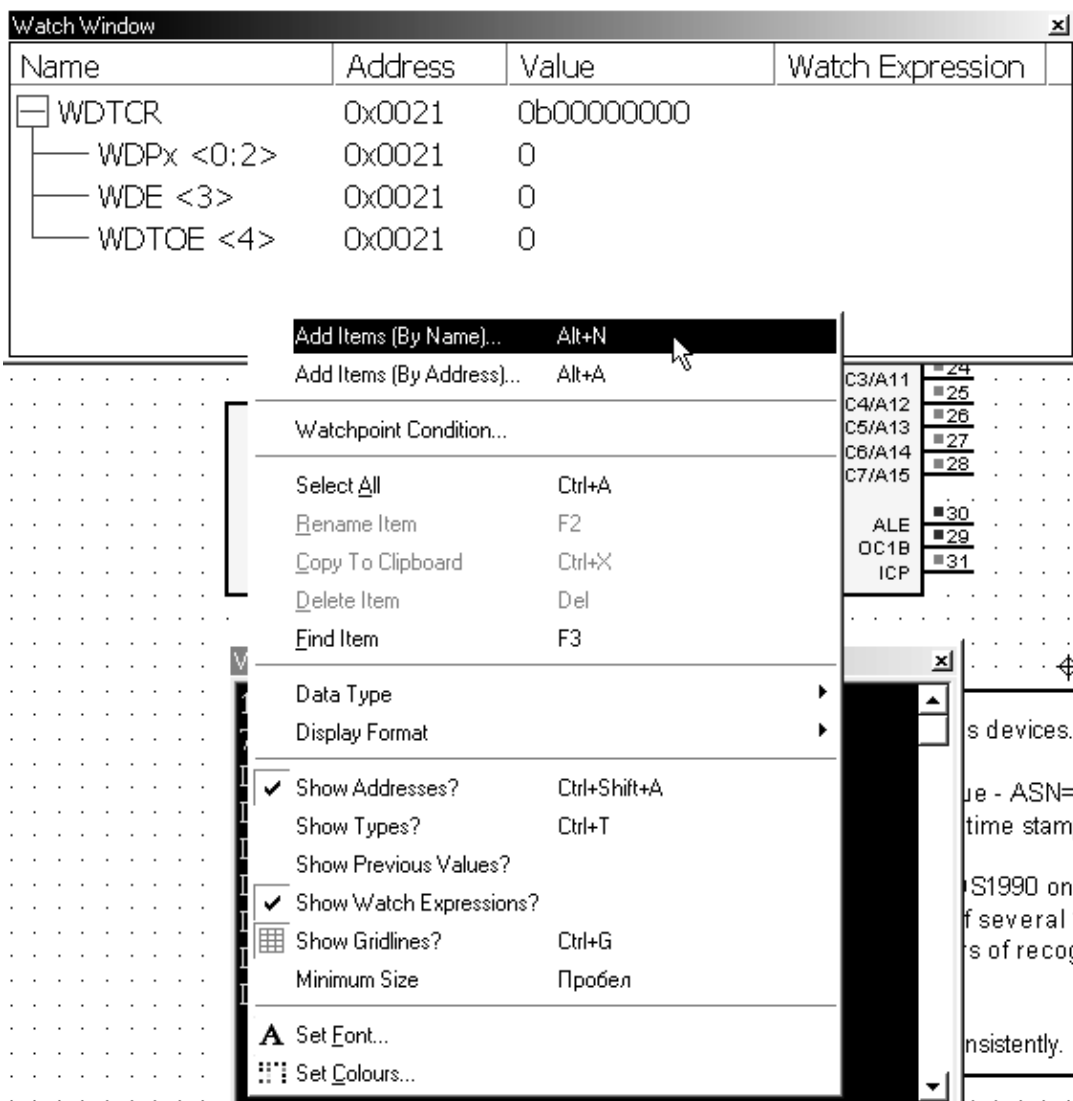


Рисунок 4.14 – Вікно Watch Window та його властивості

Якщо задамо умову у *Watch Expression* то те, що робити симулятору при її виникненні, визначається у вікні (рис. 4.15):

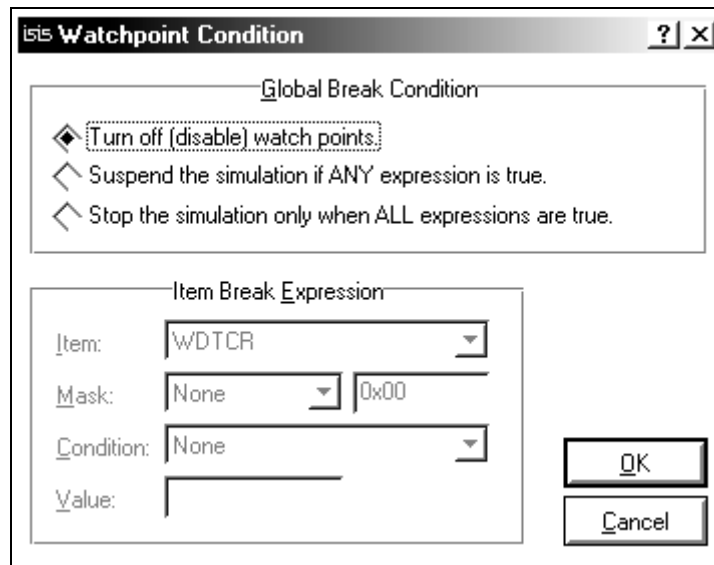


Рисунок 4.15 – Вікно, що задає як працює симулятор при виникненні умов у *Watch Expression*

Подивіться уважно й інші пункти меню **DEBUG** (див. рис. 4.13). Вікно з текстом програми – *AVR Source Code - U9*. На жаль, при симуляції МК AVR в Proteus це вікно не пам'ятає, яку програму показувати, якщо в ній немає активних ВР. Тому потрібно вказати, яку програму використовувати (рис. 4.16).

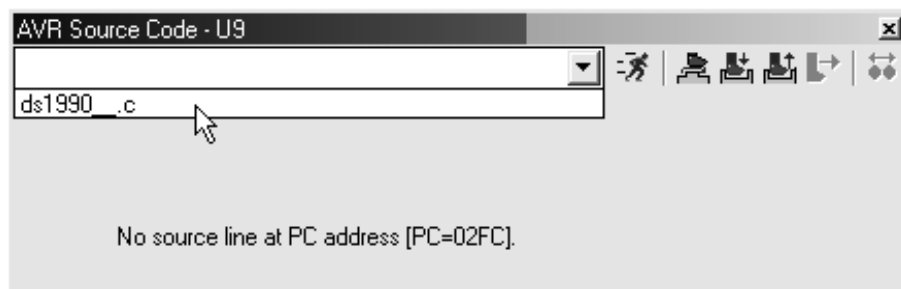


Рисунок 4.16 – Вибір тексту програми для режиму налагодження

При виборі файлу програми відкривається вікно з текстом програми (рис. 4.17). Зелені цифри показують рядки програми, які можна виділити мишею, і потім кнопкою F9 поставити ВР (точку зупинки) – червоний кружок. Програма, дійшовши до цієї точки, зупиниться в режимі паузи – на екрані монітора з'являться вікна з налагоджувальною інформацією. Якщо натиснути F9 ще раз, то ВР дезактивується – стане червоним колом і програма не зупиниться на ній. Наступне натискання кнопки F9 прибере ВР з виділеного рядка.

Розставимо ВР як на рисунку 4.17. Натискаємо кнопку "Стоп" і

потім "Пуск". Програма зупиниться на 1-ій точці ВР і з'явиться вікно з текстом програми мовою С (рис. 4.18).

```

AVR Source Code - U9
ds1990__c
---- main() {
---- unsigned char i,j,devices;
---- unsigned char n=1;
---- // init UART
0198 UCR=8;
019E UBRR=23; // Baud=9600 @ 3.6864MHz
---- // print welcome message
01A2 printf("1-Wire MicroLan Net demo\n\r");
----
---- // detect how many 1 Wire devices are present on th
01AE devices=w1_search(SEARCH_ROM,&rom_code[0,0]);
● 01BE printf("%u device(s) found\n\r",devices);
● 01CE for (i=0;i<devices;i++)
----
----     // Acknowledge DS1990 family code.
01D6     if (rom_code[i,0]==DS1990_FAMILY_CODE)
01E8         printf("DS1990  #%u serial number:",n++);
----     // Acknowledge DS2405s family code.
01F6     else if (rom_code[i,0]==DS2405_FAMILY_CODE)
0208         printf("DS2405  #%u serial number:",n++);
----     // Acknowledge DS1822s family code.
0216     else if (rom_code[i,0]==DS1822_FAMILY_CODE)
0228         printf("DS1822  #%u serial number:",n++);
----     // Acknowledge DS2430s family code.
0236     else if (rom_code[i,0]==DS2430_FAMILY_CODE)
0248         printf("DS2430  #%u serial number:",n++);

```

Рисунок 4.17 – Вікно з текстом програми

```

AVR Source Code - U9
ds1990__c
01A2 printf("1-Wire MicroLan Net demo\n\r");
----
---- // detect how many 1 Wire devices are present on th
01AE devices=w1_search(SEARCH_ROM,&rom_code[0,0]);
● 01BE printf("%u device(s) found\n\r",devices);
● 01CE for (i=0;i<devices;i++)
----
----     {
----     // Acknowledge DS1990 family code.
01D6     if (rom_code[i,0]==DS1990_FAMILY_CODE)
01E8         printf("DS1990  #%u serial number:",n++);
----     // Acknowledge DS2405s family code.
01F6     else if (rom_code[i,0]==DS2405_FAMILY_CODE)
0208         printf("DS2405  #%u serial number:",n++);
----     // Acknowledge DS1822s family code.
0216     else if (rom_code[i,0]==DS1822_FAMILY_CODE)
0228         printf("DS1822  #%u serial number:",n++);
----     // Acknowledge DS2430s family code.
0236     else if (rom_code[i,0]==DS2430_FAMILY_CODE)
0248         printf("DS2430  #%u serial number:",n++);
0256     else if (rom_code[i,0]==DS18S20_FAMILY_CODE)
0268         printf("DS18S20 #%u serial number:",n++);

```

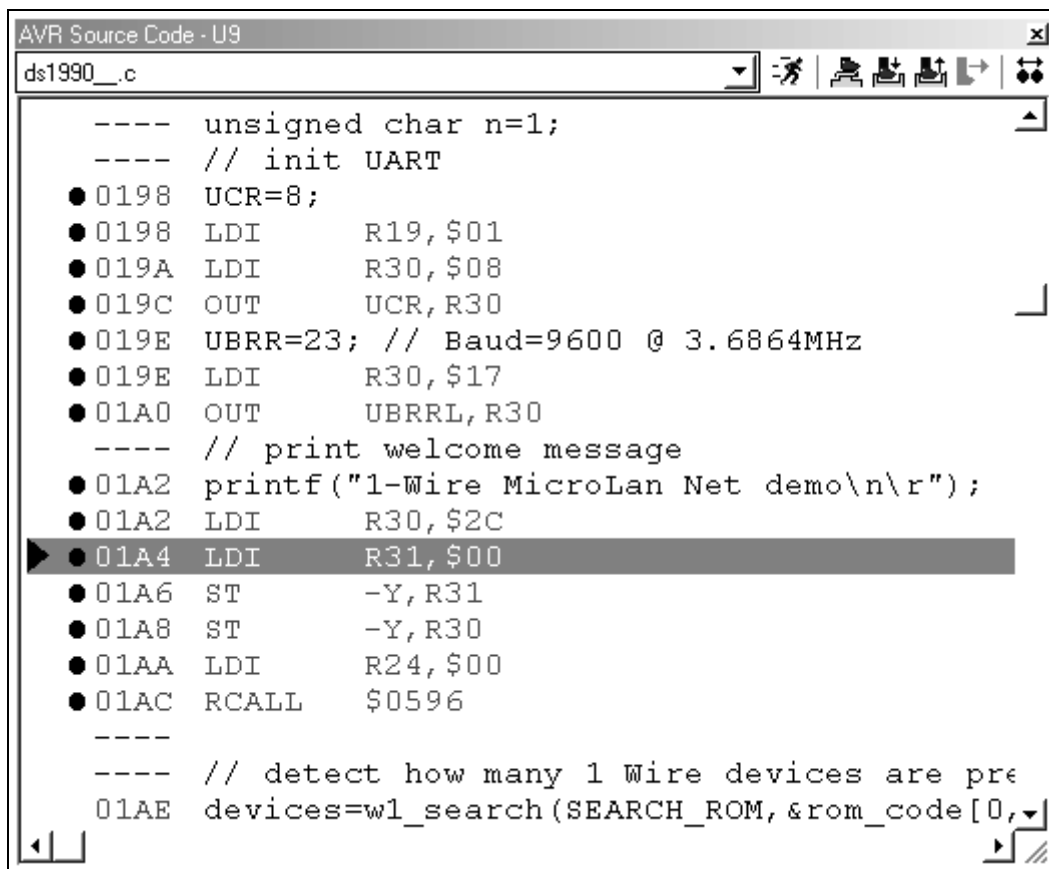
Virtual Terminal

1-Wire MicroLan Net demo

Рисунок 4.18 – Процес налагодження програми. Зупинка на першій точці ВР

Рядок, на якому зупинилася програма, підсвічується сірим кольором і на нього вказує червоний трикутник зліва від ВР. Натисніть "Старт", програма зупиниться на наступній точці ВР – тобто на наступному рядку програми. У вікні термінала видно, що відбулося виконання попереднього рядка програми.

При натисканні правою кнопкою миші у вікні програми відкривається меню, в якому можна: побачити код мовою Assembler безпосередньо в тексті програми мовою C (рис. 4.19); включити нумерацію рядків програми; ввімкнути або вимкнути усі ВР; знайти щось у програмі; перейти на потрібний рядок програми; змінити шрифт у вікні.



```
AVR Source Code - U9
ds1990_.c
---- unsigned char n=1;
---- // init UART
● 0198 UCR=8;
● 0198 LDI R19, $01
● 019A LDI R30, $08
● 019C OUT UCR, R30
● 019E UBRR=23; // Baud=9600 @ 3.6864MHz
● 019E LDI R30, $17
● 01A0 OUT UBRR, R30
---- // print welcome message
● 01A2 printf("1-Wire MicroLan Net demo\n\r");
● 01A2 LDI R30, $2C
▶ ● 01A4 LDI R31, $00
● 01A6 ST -Y, R31
● 01A8 ST -Y, R30
● 01AA LDI R24, $00
● 01AC RCALL $0596
----
---- // detect how many 1 Wire devices are pre
01AE devices=w1_search(SEARCH_ROM, &rom_code[0,
```

Рисунок 4.19 – Відображення у вікні програми на мові C еквівалентного коду програми мовою Assembler

Розставивши ВР у потрібному місці програми можна здійснити покрокове налагодження за кожним рядком програми кнопкою F12 або "Пуск".

Навіть при покроковому налагодженні всі віртуальні прилади продовжують показувати вимірювані параметри і на віртуальному осцилографі або логічному аналізаторі можна побачити після виконання якого рядка коду яка зміна відбулася на ніжках МК або в інших вузлах схеми. Детально про налагодження за початковим кодом

програми написано у розділі допомоги: "SOURCE LEVEL DEBUGGING WITHIN PROTEUS VSM".

Дані симуляції можна зберегти у файл через меню *Graph* → *Export Data*. Отримаємо файл в стандартному форматі для вимірювальних приладів – CSV (дані розділені комами). Цей файл можна потім імпортувати до Excel або до Matlab. Графіки результатів симуляції можна експортувати до графічних та CAD-форматів через меню *File* → *Export Graphics*.

4.1 Моделювання мікроконтролерів MC68HC11 фірми Motorola

Все зростаюче застосування знаходять в промислових контролерах мікропроцесори фірми Motorola. Це, перш за все, 8/16-бітові моделі серії MC68HC05, MC68HC08, MC68HC11, MC68HC12, MC68HC16. На сьогоднішній день компанією Motorola випущено декілька мільярдів мікропроцесорів цієї серії, що знайшли і знаходять до цього дня застосування практично в усіх сферах діяльності людини: від побутових приладів до складних мікропроцесорних систем.

У даний час багато фірм спеціалізуються на розробці систем програмування, орієнтованих на створення програмного забезпечення мікропроцесорів, яке використовується при проектуванні промислових мікроконтролерів. Серед них багато фірм орієнтується на вирішення цієї задачі для 8/16-бітових мікропроцесорів фірми Motorola. Одна з найбільш відомих фірма HiWare є лідером в області графічних крос-засобів розробки, відлагодження і тестування програмного забезпечення промислових мікроконтролерів [13].

Найцікавішою і найкориснішою особливістю пакета Proteus VSM є можливість симулювати роботу пристроїв введення/виведення мікроконтролера. В основному симулятори емулюють роботу регістрів мікропроцесора і виконання його команд. Використовуючи пакет Proteus VSM, розробник промислового мікроконтролера може емулювати роботу не лише мікропроцесора, а й таких пристроїв введення/виведення, як регістрові пристрої, аналого-цифрові перетворювачі, таймери, а також інші компоненти системи.

З використанням пакета Proteus VSM процес відлагодження можна проводити так:

- симуляція створеного коду на персональному комп'ютері в середовищі Windows (із збереженням реального часу виконання команд емульованої мікропроцесорної системи). У

цьому випадку відлагодження програмного забезпечення мікроконтролера може бути проведено до створення мікропроцесорної системи;

- відлагодження створеного коду в середовищі мікроконтролера (програмне забезпечення відлагоджується шляхом завантаження виконуваного коду в його пам'ять). В цьому випадку персональний комп'ютер використовується як пристрій підготовки кодів програми, завантаження їх в мікроконтролер, а також як пристрій керування процесом відлагодження і тестування. Для цих цілей використовується послідовний канал, що з'єднує персональний комп'ютер і мікроконтролер. У середовищі мікроконтролера процес завантаження і відлагодження підтримує спеціальний монітор, що зберігається в його ПЗП.

На рисунку 4.20 зображений проект, що демонструє роботу MC68HC11, який керується через послідовний порт RS232. Для цього використовується програмне середовище BASIC11.

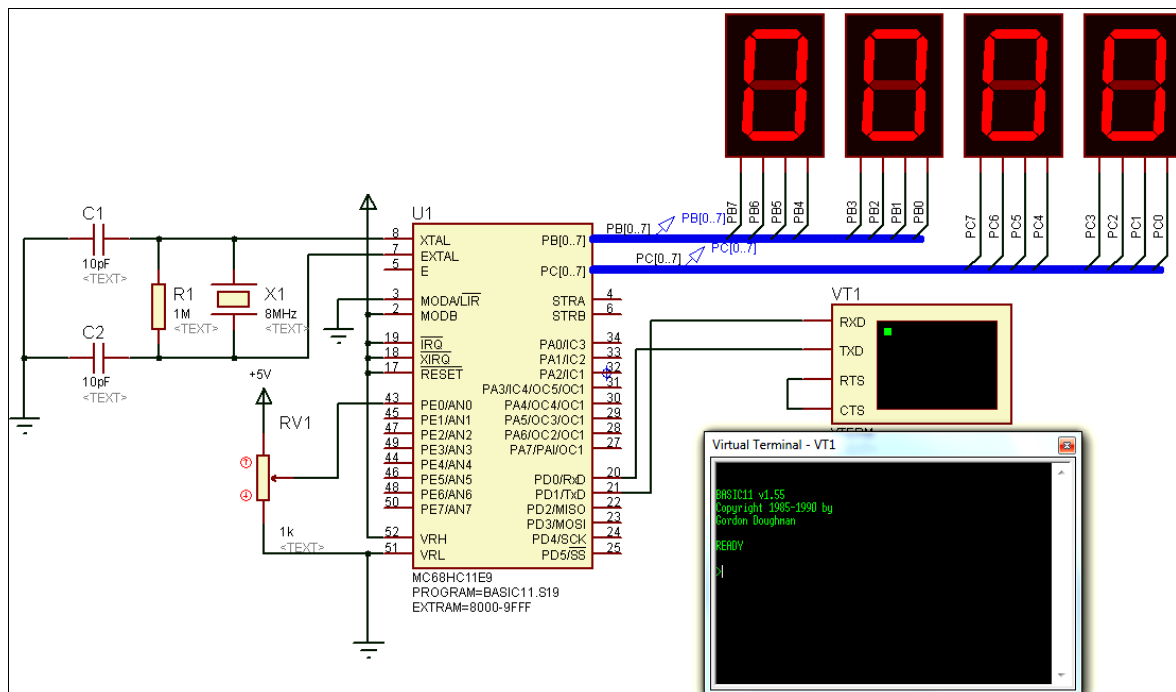


Рисунок 4.20 – Моделювання роботи мікроконтролера MC68HC11 у середовищі Proteus

На екрані віртуального терміналу з'являється запрошення ввести команду. Виведемо на індикаторі число 6811. Для цього введемо команди: PortB=\$68, PortC=\$11 (рис. 4.21).

На рисунку 4.22 показано моделювання роботи LCD-індикатора PG12864F, який програмно керується мікроконтролером MC68HC11.

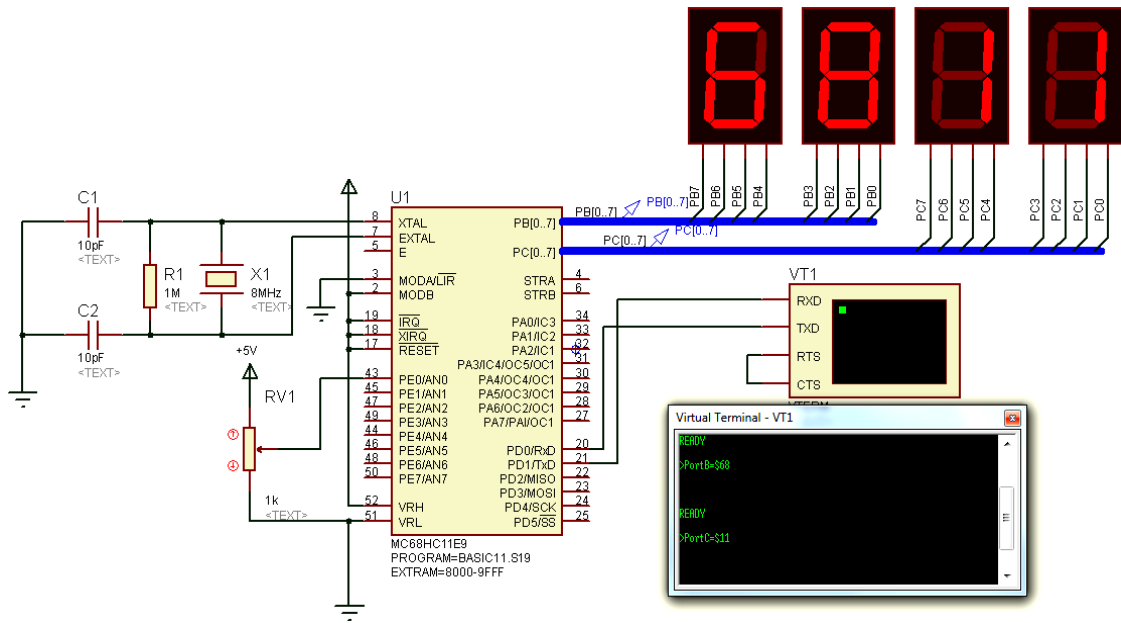


Рисунок 4.21 – Моделювання роботи мікроконтролера MC68HC11 у середовищі Proteus (виведення числа 6811H)

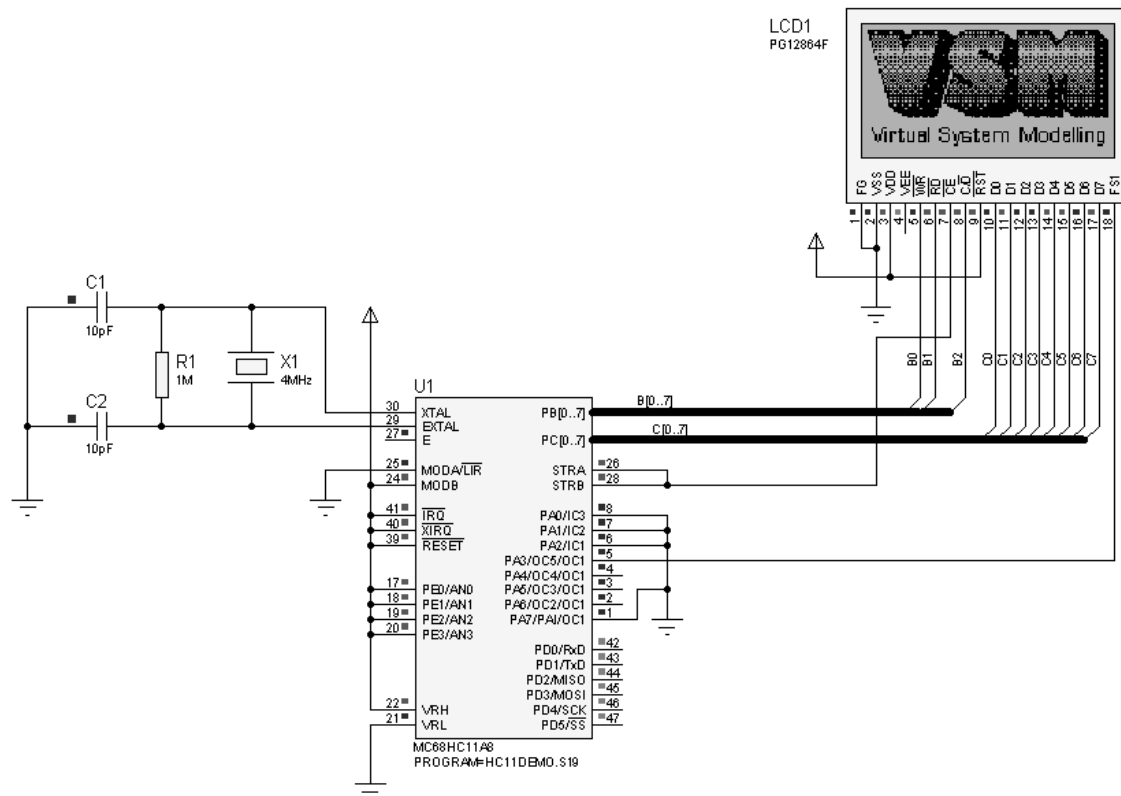


Рисунок 4.22 – Моделювання роботи LCD-індикатора PG12864F, що керується мікроконтролером MC68HC11, у середовищі Proteus

4.2 Моделювання мікроконтролерів сімейства x51

Розробка систем керування й контролю з використанням однокристальних мікроконтролерів на сьогодні переживає справжній бум. Системи на базі мікроконтролерів використовуються практично в усіх сферах життєдіяльності людини, і щодня з'являються все нові й нові галузі застосування цих пристроїв. Останнім часом у зв'язку з бурхливим розвитком електроніки й схемотехніки розширилися можливості й самих мікроконтролерів, що дозволяють виконувати багато завдань, наприклад, таких, раніше недоступних для реалізації, як обробка аналогових сигналів. Одним з найбільш ранніх мікроконтролерів, що з'явилися на ринку, є мікроконтролер 8051, розроблений фірмою Intel більше двадцяти років тому. Незважаючи на настільки пристойний вік, класичний 8051 й сьогодні залишається одним з найбільш популярних при розробці систем керування й контролю. Добре продумана архітектура й інтуїтивно зрозуміла система команд впливають на вибір багатьох розробників апаратно-програмних систем [65].

Для систем з мікроконтролерами 8051 розроблені потужні інструментальні засоби, що дозволяють знизити до мінімуму час розробки й налагодження програмного забезпечення. Провідні позиції займають програмні засоби, розроблені фірмою Keil. Інструментальні засоби цієї фірми охоплюють цілий ряд таких потужних додатків, як компілятор мови C для мікроконтролерів 8051 (Keil C51), макроасемблер A51, сумісний з ASM-51, і зручна графічна оболонка для розробки й налагодження програм Keil uVision [39]. Середовище розробки Keil дозволяє створювати складні проекти, що складаються з різних модулів, написаних як на C, так і мовою асемблера, підключати бібліотеки функцій. Крім того, це середовище містить зручний відлагоджувач (симулятор), що дозволяє оцінити й швидко перевірити працездатність програми.

Проведемо моделювання у середовищі Proteus роботи кабельного тестера [15]. Спочатку у середовищі Keil ASM-51 була відлагоджена програма Test.asm і отримано файл Test.hex, а в пакеті Proteus 7 Professional (Isis 7 Professional) проводилась перевірка працездатності схеми. Схема кабельного тестера наведена на рисунку 4.23. У властивостях мікроконтролера AT89C2051 (рис. 4.24) підключаємо отриманий файл test.hex. У початковий момент світиться світлодіод D1, який інформує, що схема готова до роботи.

При підключенні в режимі перевірки Pathcord світлодіоди по черзі загораються ("біжучий вогник"), це означає, що монтаж кабелю виконаний правильно (рис. 4.25). В процесі роботи можна перевірити

стан основних ресурсів мікроконтролера (рис. 4.26). На рисунку 4.27 наведений фрагмент покрокового дослідження роботи схеми.

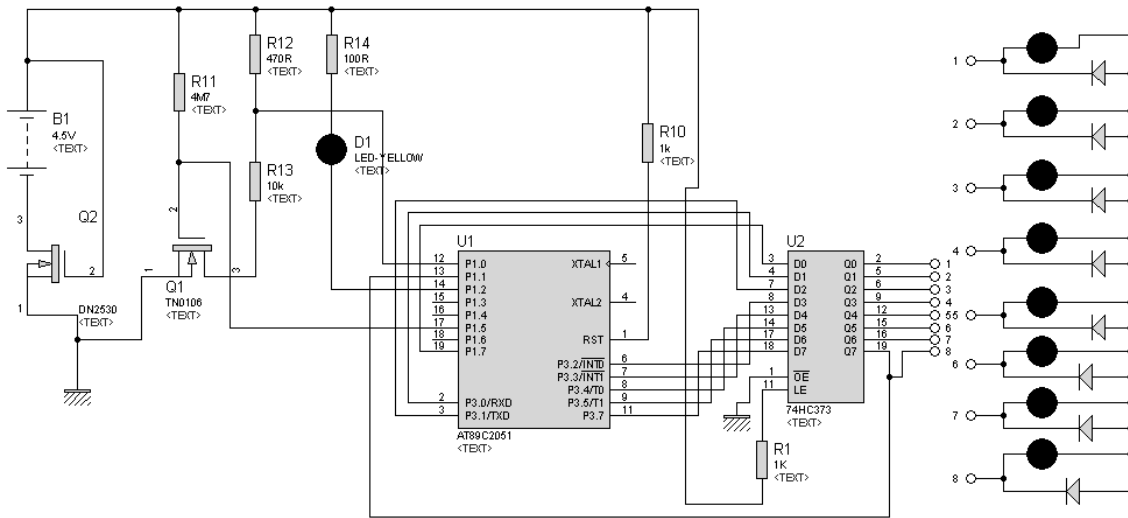


Рисунок 4.23 – Дослідження роботи мережного тестера в Proteus 7.6

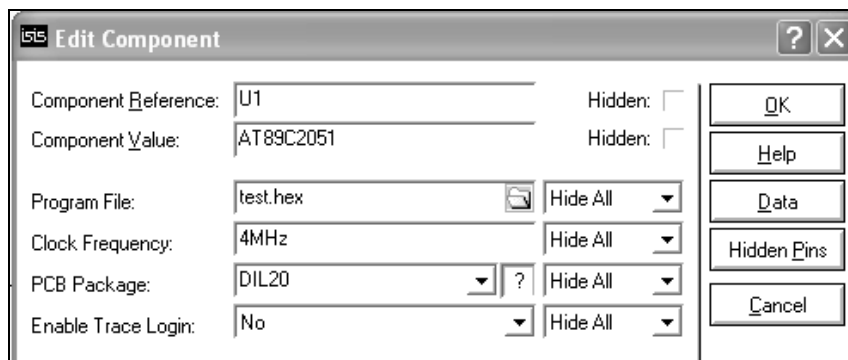


Рисунок 4.24 – Підключення програмного забезпечення test.hex

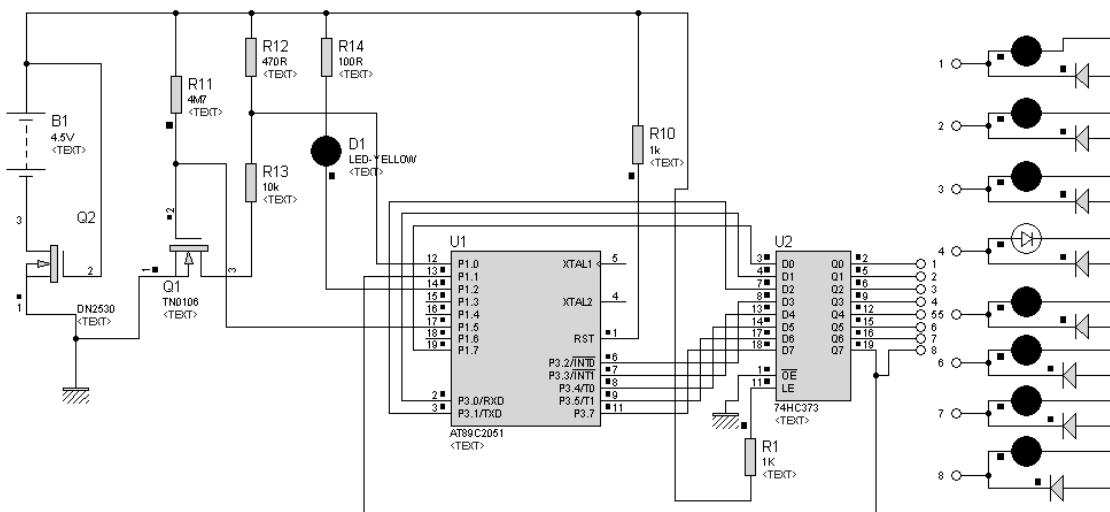


Рисунок 4.25 – Роботи схеми в режимі перевірки Pathcord

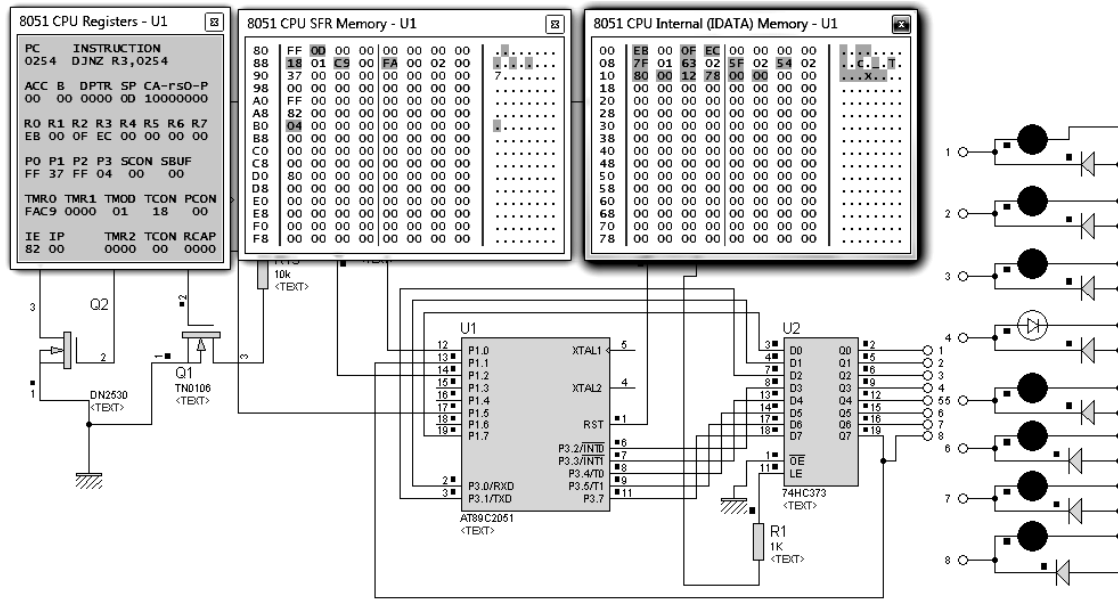


Рисунок 4.26 – Режим покрокового відлагодження програми в режимі перевірки Pathcord

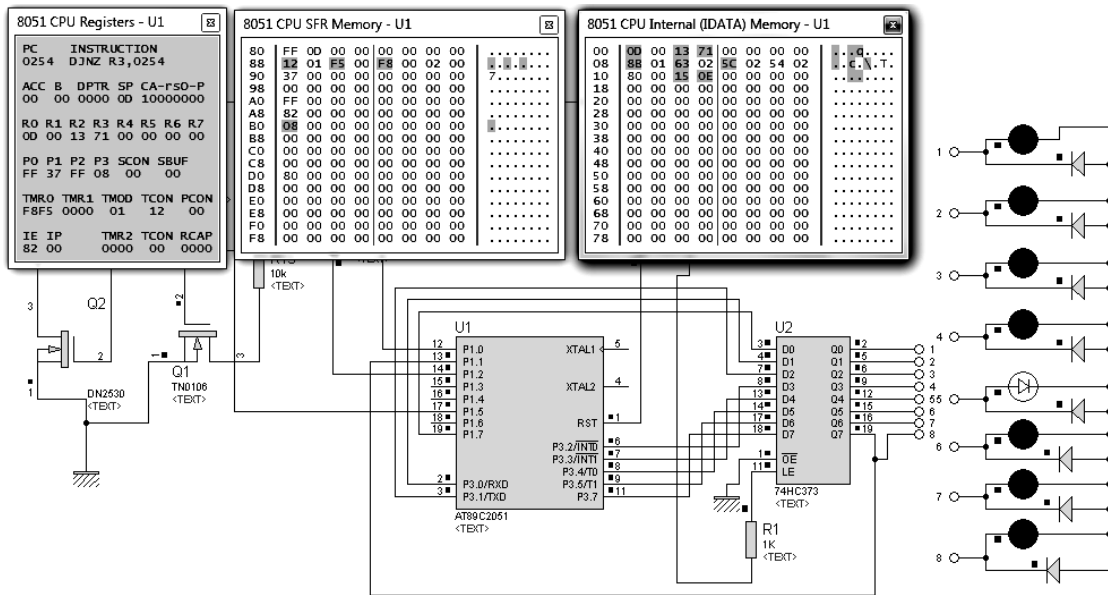


Рисунок 4.27 – Режим покрокового відлагодження (продовження)

Виконаємо моделювання роботи електронного калькулятора, що реалізований на мікроконтролері 80C51. Для того, щоб виконати дію, необхідно спочатку обнулити калькулятор натисканням кнопки ON/C. Для виконання дії ділення потрібно зробити такі дії: набрати число – ділене, наприклад 456 (рис. 4.28, а); натиснути кнопку ділення; набрати дільник, нехай 3 (рис. 4.28, б); натиснути дорівнює і на екран виведеться відповідь (рис. 4.28, в).

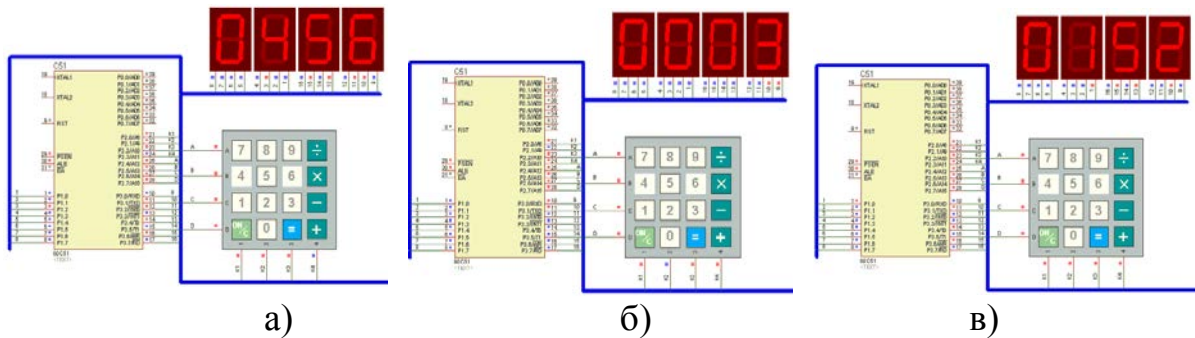


Рисунок 4.28 – Моделювання роботи калькулятора: а) введення числа 456 (ділене), б) введення числа 3 (дільник), в) результат обчислення

На рисунку 4.29 зображений процес моделювання двоканального блока живлення з цифрою індикацією на мікроконтролері 80C51.

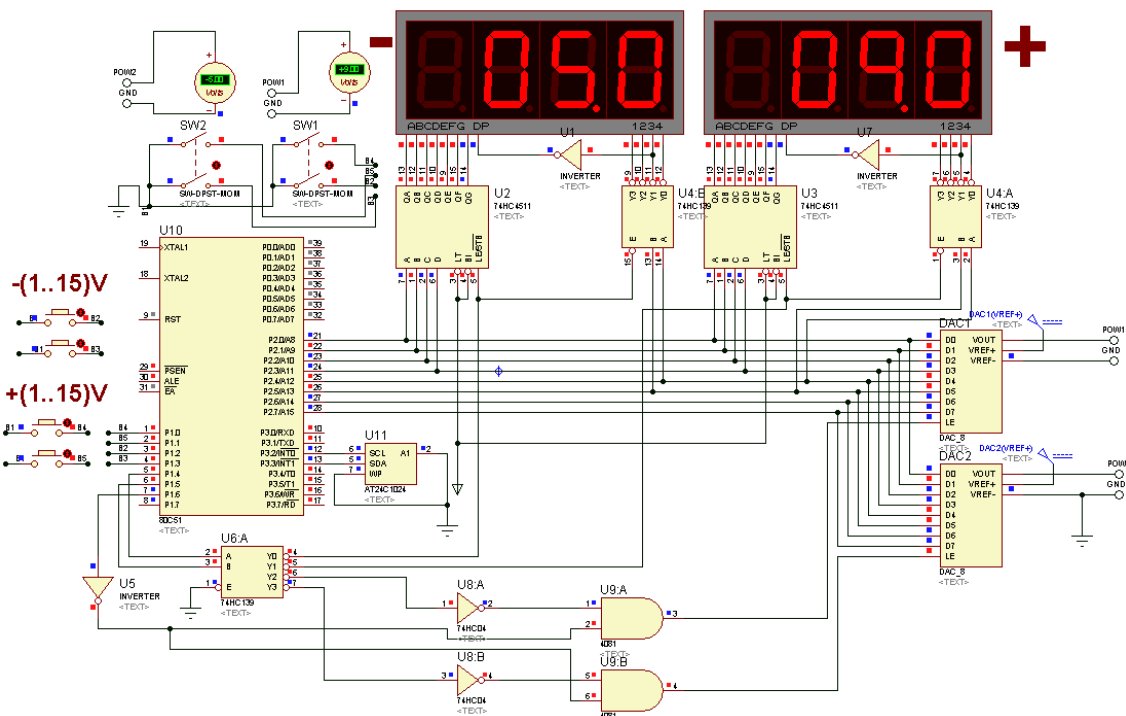


Рисунок 4.29 – Моделювання роботи блока живлення з цифрою індикацією

4.3 Моделювання мікроконтролерів сімейства PIC

Мікроконтролери фірми Microchip – це одне з найбільш популярних сімейств, що використовуються при вивченні мікроконтролерів на різних етапах навчального процесу. Освоєння мікроконтролерів цієї фірми, зокрема, полегшується невеликим набором команд і відносно простою передовою

архітектурою. Крім використання в промисловості і освітньому процесі, мікроконтролери сімейства PIC застосовуються в більшості аматорських пристроїв, в чому можна переконатися, відкривши будь-який журнал, присвячений радіоаматорству [30, 62].

Розробка електронної схеми на мікроконтролері охоплює написання і відлагодження програми на асемблері або мовою високого рівня. Для створення і відлагодження програм для мікроконтролерів сімейства PIC використовується середовище розробки MPLAB IDE. Воно об'єднує у собі редактор коду, мови програмування `masm` та `masm30`, інтерфейс програмування (PICkit, PICSTAR Plus, MPLAB ICD2, MPLAB PM3), інтерфейс відлагодження програмного коду (програмна емуляція MPLAB SIM, внутрішньосхемне відлагодження MPLAB REAL ICE, MPLAB ICD2, PICkit2), підтримку мов програмування (MPLAB C18, MPLAB C30, MPLAB C32), а також підтримку сторонніх розробників (IAR, HI-Tech, CCS, RTOS, MATLAB, Proteus) [51]. Програма ISIS, складова частина Proteus, дозволяє перевірити роботу не лише мікроконтролера, а й всього пристрою. Наведемо приклади моделювання електронних схем з використанням мікроконтролерів сімейства PIC.

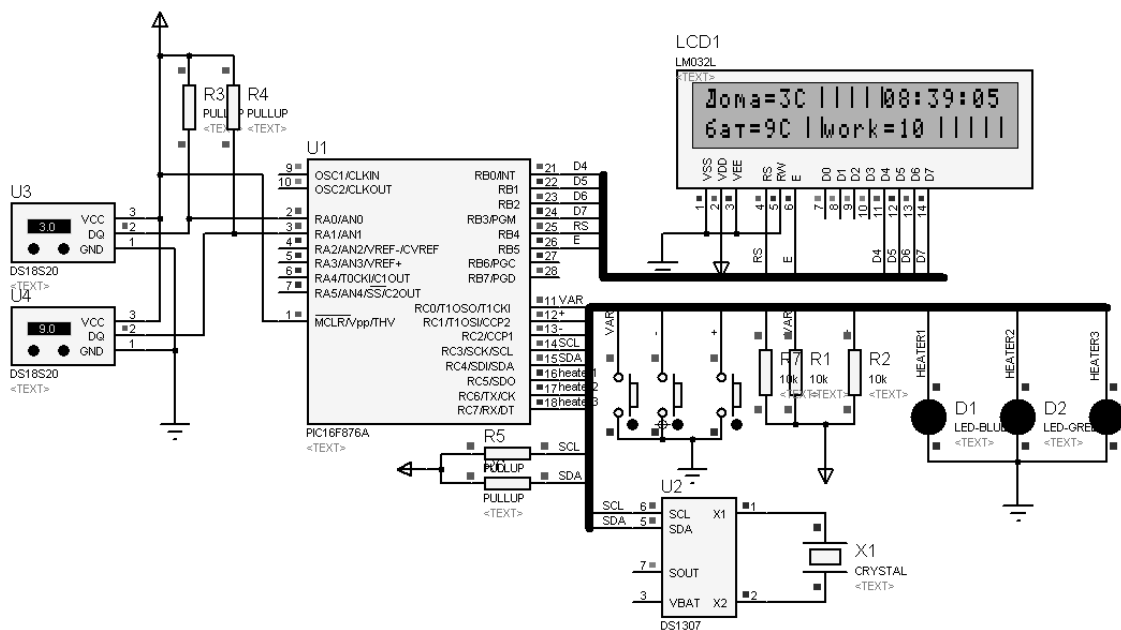


Рисунок 4.30 – Моделювання роботи блока керування системою опалення

На рисунку 4.30 наводиться моделювання роботи блока керування трьома тенами, що вбудовані в опалення, на

мікроконтролері PIC16F876 [72]. При падінні температури на 2 градуси включається перший тен (світлодіод D1), падіння ще на 2 градуси включає другий тен (світлодіод D2) та якщо ще на два градуси, то включається третій тен (світлодіод D3). Регулювання ведеться відносно датчика на батареї. На індикаторі відображається поточний час, температура у приміщенні та температура батареї, режим роботи.

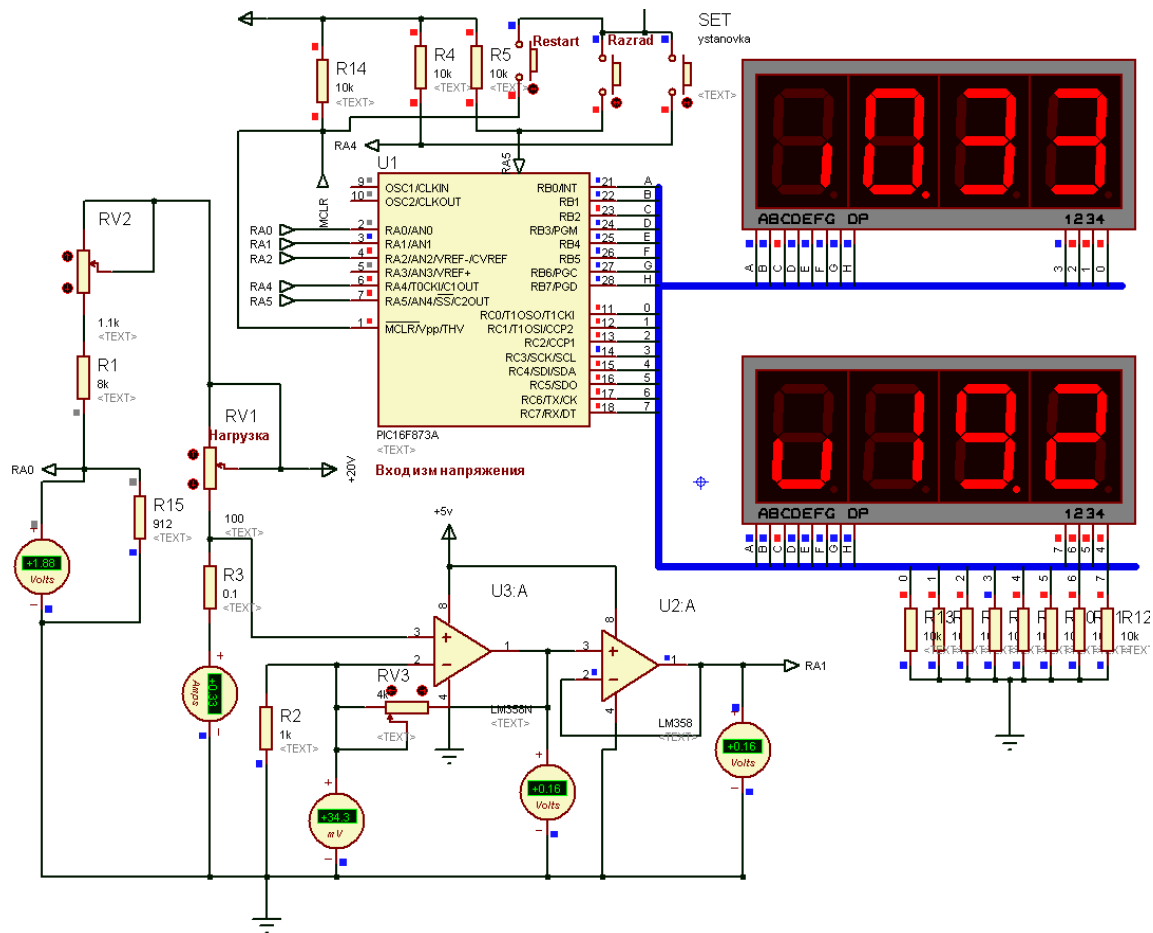


Рисунок 4.31 – Моделювання роботи блока індикації напруги та струму лабораторного блока живлення

На рисунку 4.31 наводиться моделювання роботи блока індикації напруги та струму лабораторного блока живлення [25] на мікроконтролері PIC16F873, що містить 10-розрядний АЦП. Результати вимірювань відображаються двома чотирирозрядними індикаторами.

На рисунку 4.32 наводиться моделювання "електронної книги", що реалізована на базі мікроконтролера PIC16F84, Multi Media Card (MMC), LCD індикатора стільникового телефону Nokia7110 [23].

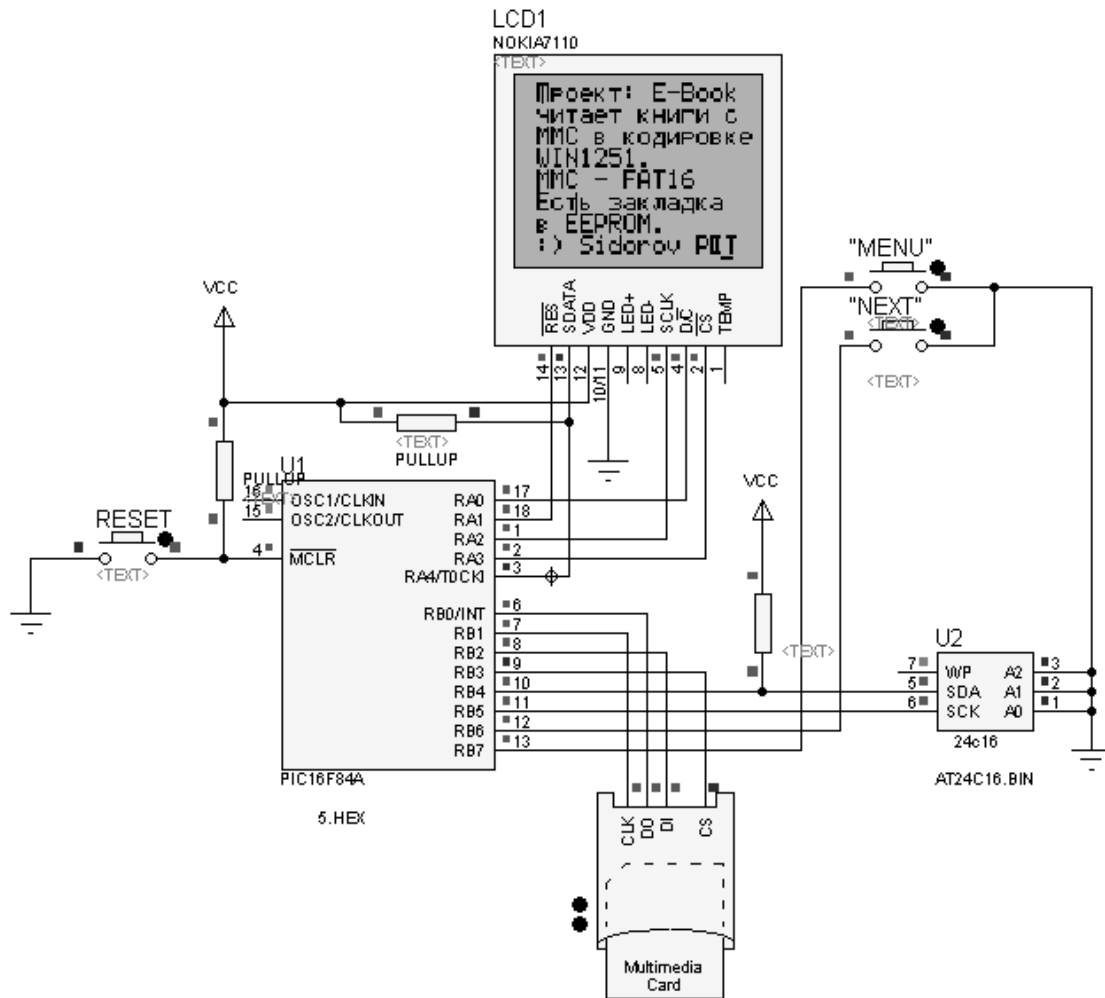


Рисунок 4.32 – Моделювання роботи E-Book

4.4 Моделювання мікроконтролерів сімейства AVR

На сьогодні дуже поширені мікроконтролери фірм Dallas Semiconductor, Microchip або Hitachi. Широко розвинені лінії PIC-контролерів фірми Microchip і мікроконтролерів H8/300 фірми Hitachi забезпечують досить високу продуктивність при невеликому енергоспоживанні. Ефективність роботи мікроконтролерів Dallas Semiconductor, що мають базову архітектуру MCS51, в середньому перевищує стандартну в 2,5-3 рази. Нові процесорні платформи MSP430 фірми Texas Instruments і XE8000 фірми Xemics, що з'явилися останнім часом, також заслуговують на увагу, особливо якщо основним критерієм є мінімальне енергоспоживання.

Мікроконтролери фірми Atmel є потужним інструментом для створення сучасних високопродуктивних і економічних багатоцільових контролерів. На сьогодні співвідношення "ціна - продуктивність - енергоспоживання" для AVR є одним з кращих на

світовому ринку 8-розрядних мікроконтролерів.

Розглянемо моделювання мікропроцесорних пристроїв на мікроконтролерах AVR.

1. Контролер сигналізації на мікроконтролері AT90S2313 [3]. Основні режими роботи контролера сигналізації на мікроконтролері AT90S2313: режим очікування (контакти кнопки, встановленої на дверях, замкнуті, контакти кодового пристрою розімкнені, сирена і світлодіод вимкнені, струм через соленоїд замка протікати не повинен); режим вторгнення (при зломі дверей контакти кнопки, що встановлена на дверях, замкнуті. Якщо попереднього замикання контактів кодового пристрою не було, це повинно викликати включення сирени і світлодіода на віддаленому пульті. Сирена повинна працювати в режимі: тривалість звуку і тривалість пауз рівні і складають 0,5. Виключення сирени і світлодіода повинно здійснюватись усередині приміщення окремою кнопкою, розташованою на платі контролера); режим штатного відмикання дверей (після введення коду в правильній послідовності короткочасно замикається пара контактів кодового пристрою, до соленоїда замка повинна бути прикладена напруга 12 В протягом 2 с, замок відімкнеться. Якщо двері після цього будуть відкриті, замикання контактів кнопки, встановленої на дверях, не приведе до спрацьовування сирени і світлодіода); режим відмикання дверей зсередини (при замиканні контактів кнопки відмикання зсередини до соленоїда замка повинна бути прикладена протягом 2 с напруга 12 В, замок відімкнеться. Якщо двері після цього будуть відкриті, замикання контактів кнопки, встановленої на дверях, не приведе до спрацьовування сирени і світлодіода). Схема пристрою, опис роботи охоронної сигналізації та програмне забезпечення наводяться у [3]. Моделювання роботи схеми, що наведена на рисунку 4.33, проводиться у пакеті Proteus 7.6 Professional [69]. Схема, що зображена на рисунку 4.33, *а*, відповідає режиму очікування, коли мікроконтролер здійснює циклічну перевірку стану ліній порту D. Режим звичайного відкриття зображений на рисунку 4.33, *б*. Коли введений правильний код (SW2 замкнутий), на лінії PB2 формується рівень "1", який приводить до спрацьовування реле (світиться світлодіод D4). Рівень "1" утримується на виході PB2 орієнтовно 2 секунди. Після 5 секунд, коли електромагніт замка закритий, здійснюється перевірка стану датчика DOOR (SW3). Якщо SW3 не замкнутий, то це еквівалентно тому, що двері закриті й мікроконтролер переходить до режиму охорони.

Режим відкриття дверей зсередини зображений на рисунку 4.34, *а*. Коли натиснута кнопка SW1 (OPEN) на лінії PB2 формується

рівень "1", який приводить до спрацьовування реле (світиться світлодіод D4). Рівень "1" утримується на виході PB2 орієнтовно 2 секунди. Після 5 секунд, як електромагніт замка закритий, здійснюється перевірка стану датчика DOOR (SW3). Якщо SW3 не замкнутий, то це еквівалентно тому, що двері закриті й мікроконтролер переходить до режиму охорони. На рисунку 4.34, б зображений режим вторгнення. З появою низького рівня в колі DOOR (SW3) на лінії PB0 утворюється сигнал логічної "1", що приводить до світіння індикатора D3, та утворюється імпульсна послідовність з частотою 1кГц на виводі PB1, яку при наявності гучномовця можна почути.

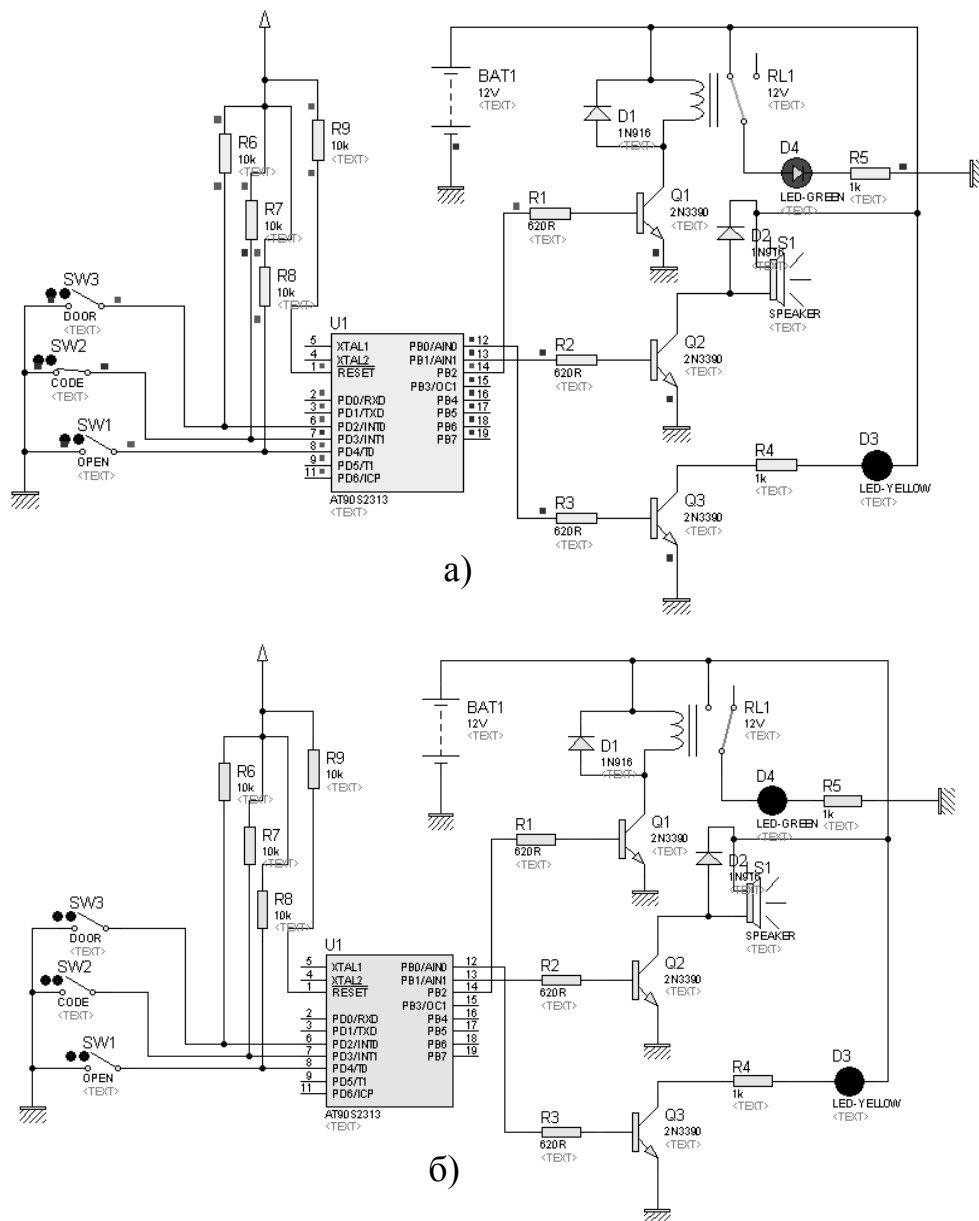
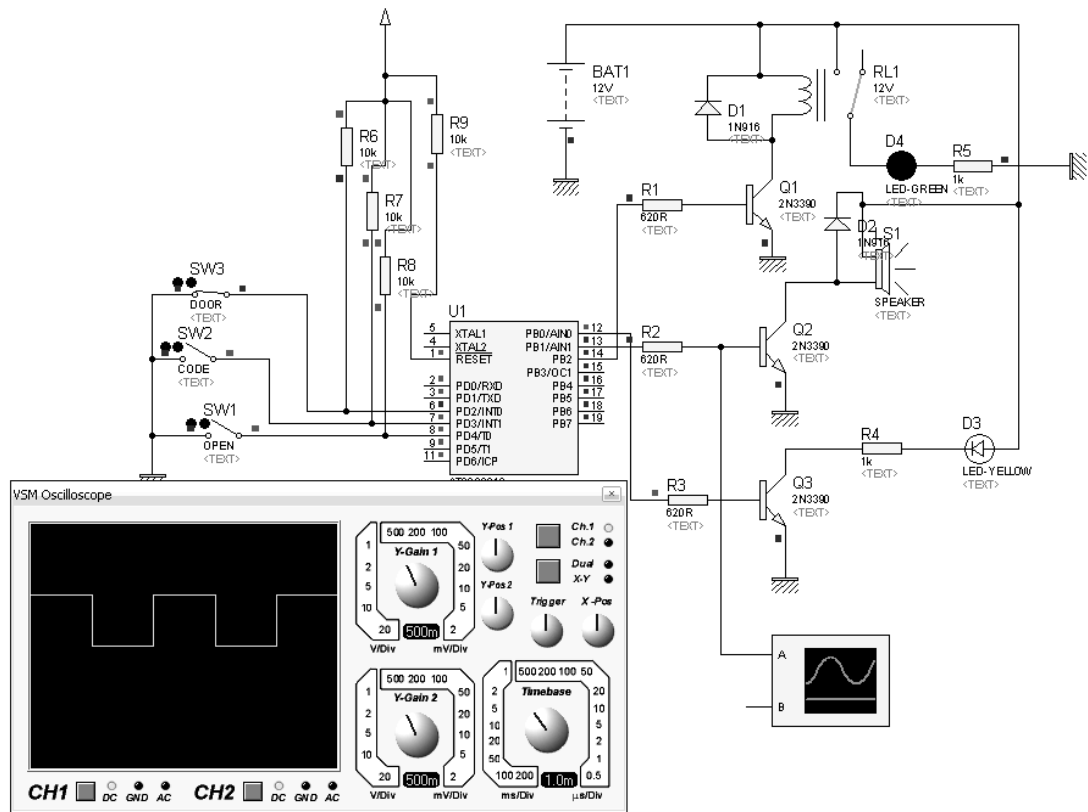
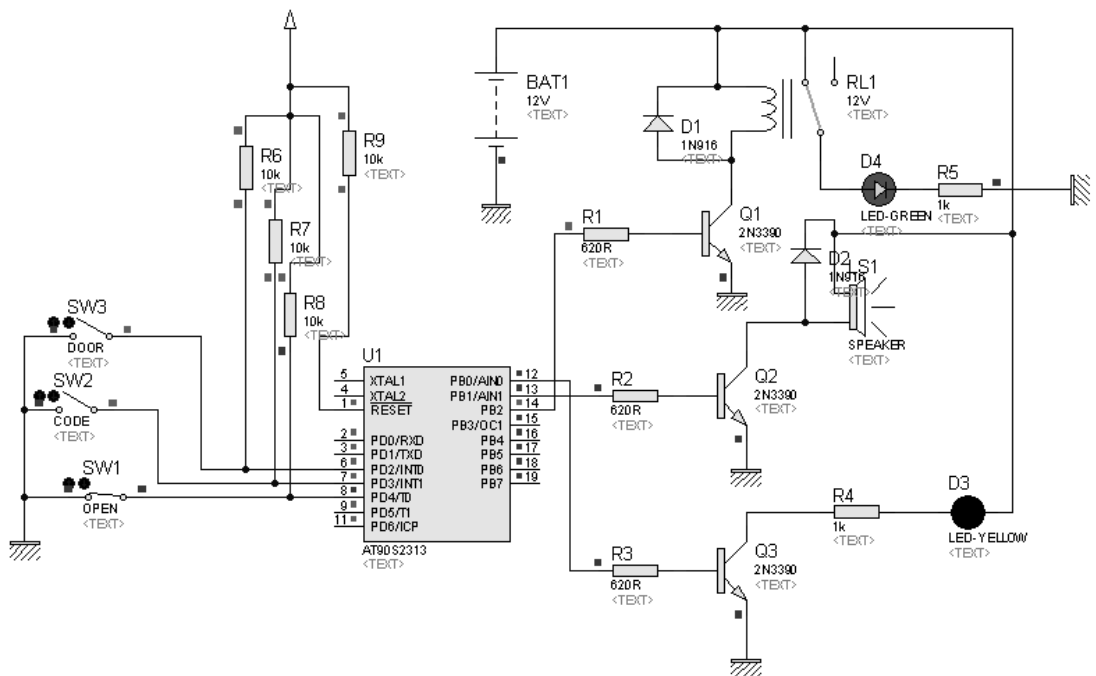


Рисунок 4.33 – Схема охоронної сигналізації (а) та режим звичайного відкриття дверей (б)



a)



b)

Рисунок 4.34 – Охоронна сигналізація в режимі відкриття дверей зсередини (а) та в режимі вторгнення (б)

2. Моделювання роботи охоронної сигналізації з інфрачервоним бар'єром [46]. Схема моделювання у пакеті Proteus 7.02 Professional наведена на рисунку 4.35. Під час моделювання замість інфрачервоного світлодіода VL07J46 та фотоприймача TSOP1736 з бібліотеки програми вибрана оптопара. Перемичка JP1 (Бар'єр/відбиття) знаходиться в режимі – "відбиття", перемичка JP2 (короткий/довгий звуковий сигнал) – в режимі "довгий сигнал", перемичка JP3 (довга/коротка пауза між імпульсами) – в режимі "коротка пауза".

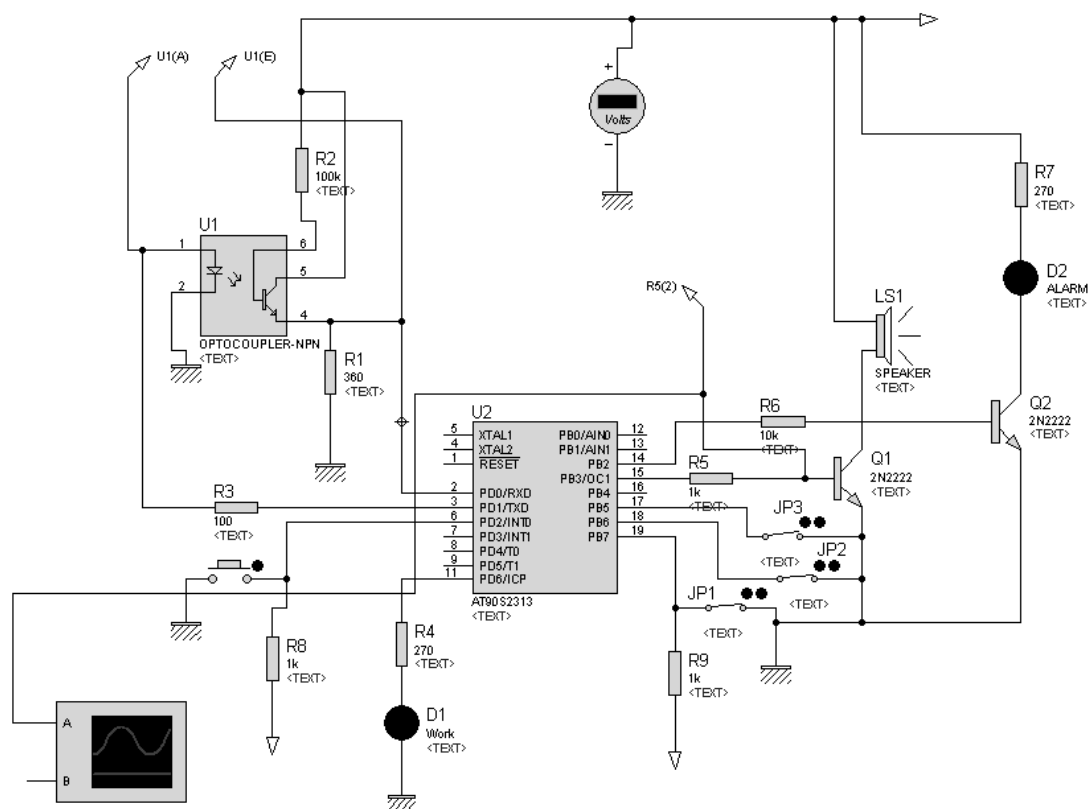


Рисунок 4.35 – Схема охоронної сигналізації в програмі Proteus 7 Professional

На рисунку 4.36 зображений режим охорони. Імпульси передаються та приймаються без помилок. Світлодіод D1 періодично спалахує, що інформує про режим охорони. Світлодіод D2 погашений, звуковий сигнал тривоги не утворюється.

На рисунку 4.37 наведений режим "Тривога". Світлодіод D1 світиться, що інформує про режим "Тривога", світлодіод D2 світиться, утворюється звуковий сигнал тривоги. При наявності звукової картки в комп'ютері можливо почути звуковий сигнал, який випромінює гучномовець схеми.

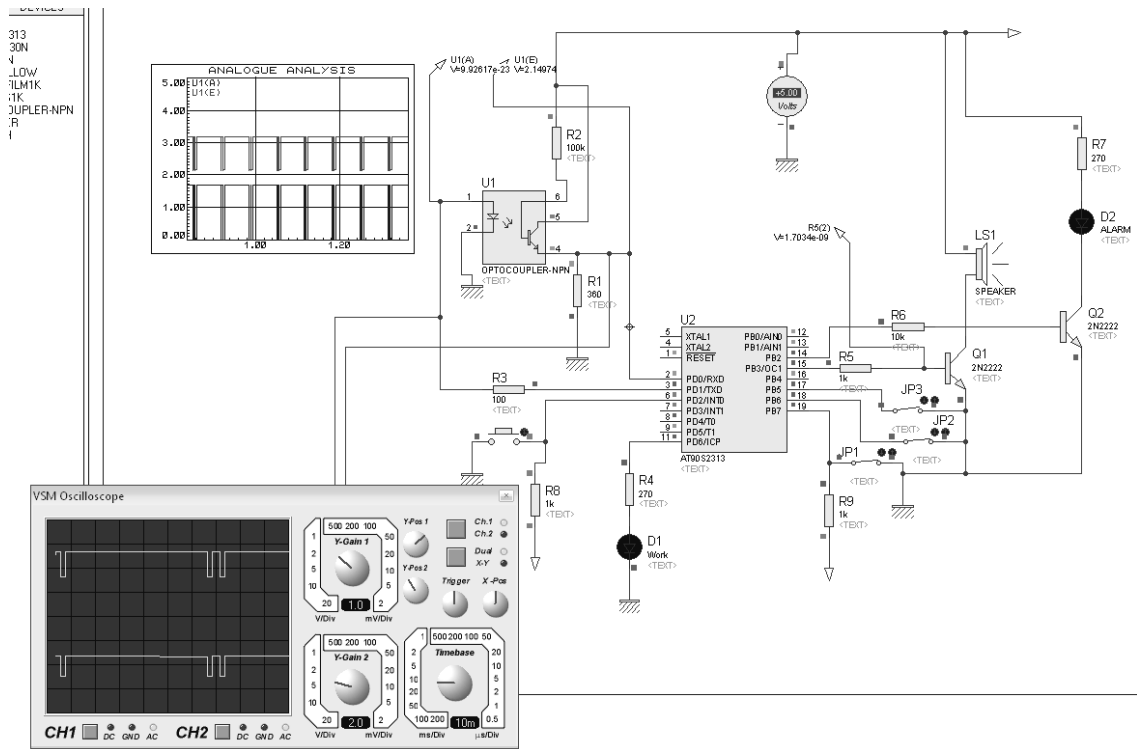


Рисунок 4.36 – Охорона сигналізація у режимі "Охорона"

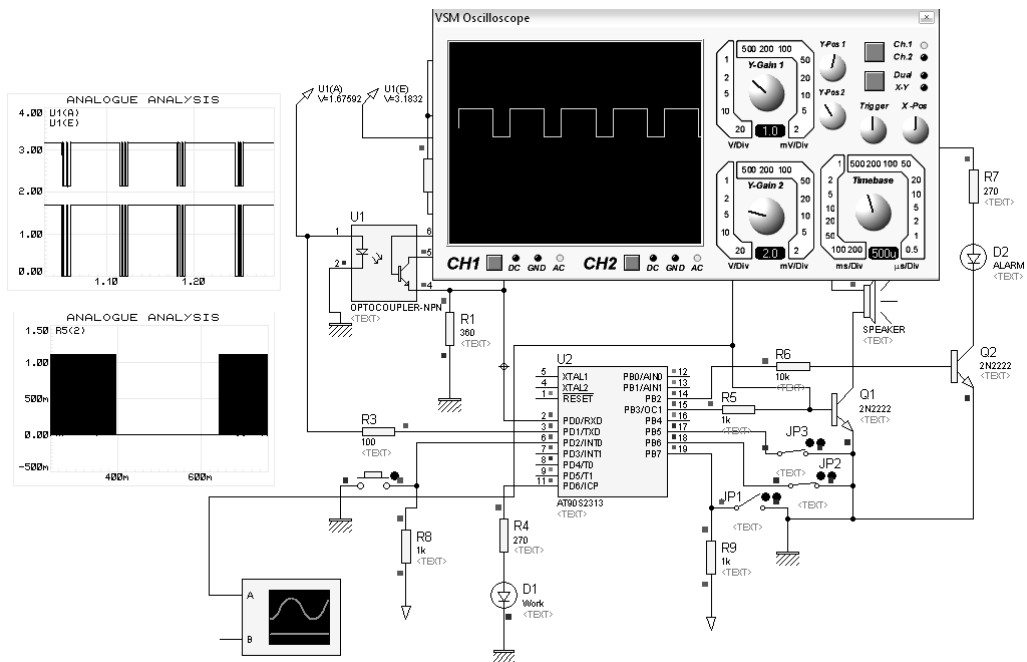


Рисунок 4.37 – Моделування режиму "Тривога"

3. Віртуальний стенд EV8031/AVR на мікроконтролері AT90S8515 [71]. Віртуальний стенд, що зображений на рисунку 4.38, є аналогом реального стенда EV8031/AVR, що розроблений фірмою Open System [79]. Стенд забезпечує обмін інформацією між

мікроконтролером AT90S8515 та периферійними пристроями: статична індикація, динамічна індикація, клавіатура, кнопки, лінійка світлодіодів, цифроаналоговий перетворювач, світлодіодна матриця.

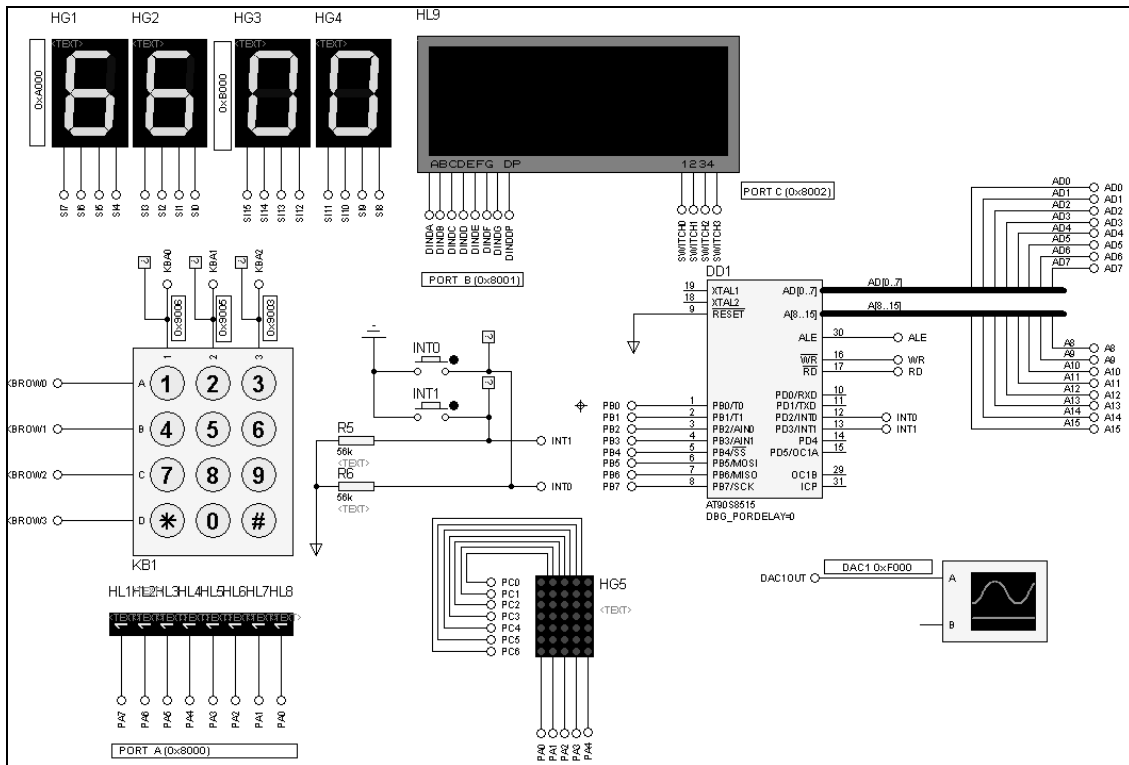


Рисунок 4.38 – Фрагмент віртуального стенду у середовищі Proteus VSM

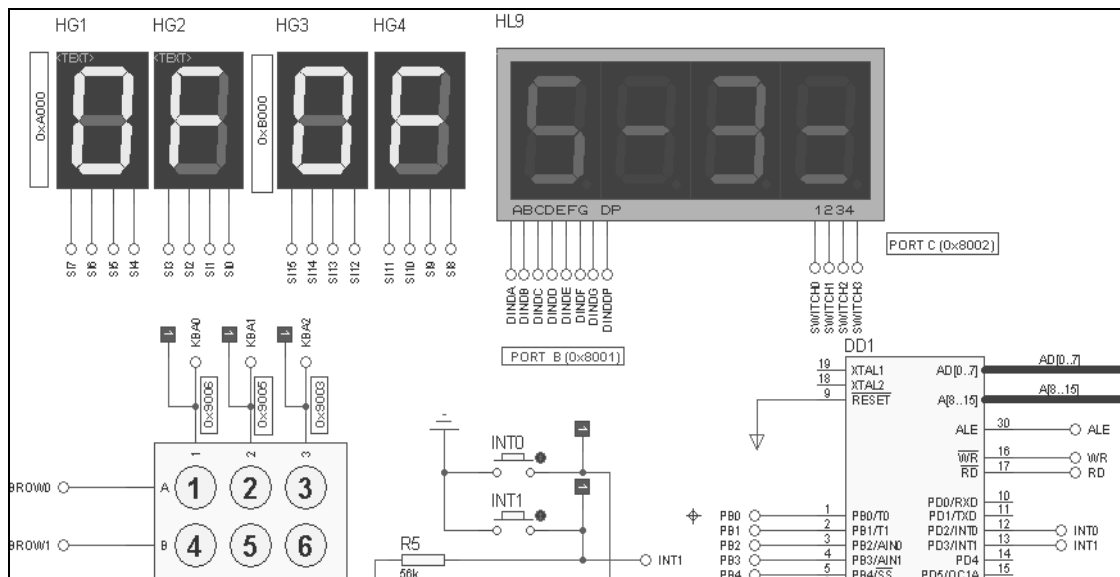


Рисунок 4.39 – Демонстрація роботи статичної та динамічної індикації на віртуальному стенді у середовищі Proteus VSM

У програмному середовищі Proteus VSM віртуальний стенд реалізований у вигляді схеми, що складається з 5 аркушів. На перший аркуш (рис. 4.38) винесені: мікроконтролер, статичні та динамічні індикатори, клавіатура, кнопки, 8 світлодіодів, світлодіодна матриця, вихід ЦАП. Це дозволяє оперативно аналізувати роботу мікропроцесорної системи та програмного забезпечення.

Особливості застосування віртуального стенда наводяться у [71]. На рисунку 4.39 наводиться демонстрація роботи статичної та динамічної індикації, а на рисунку 4.40 – робота зі світлодіодною матрицею.

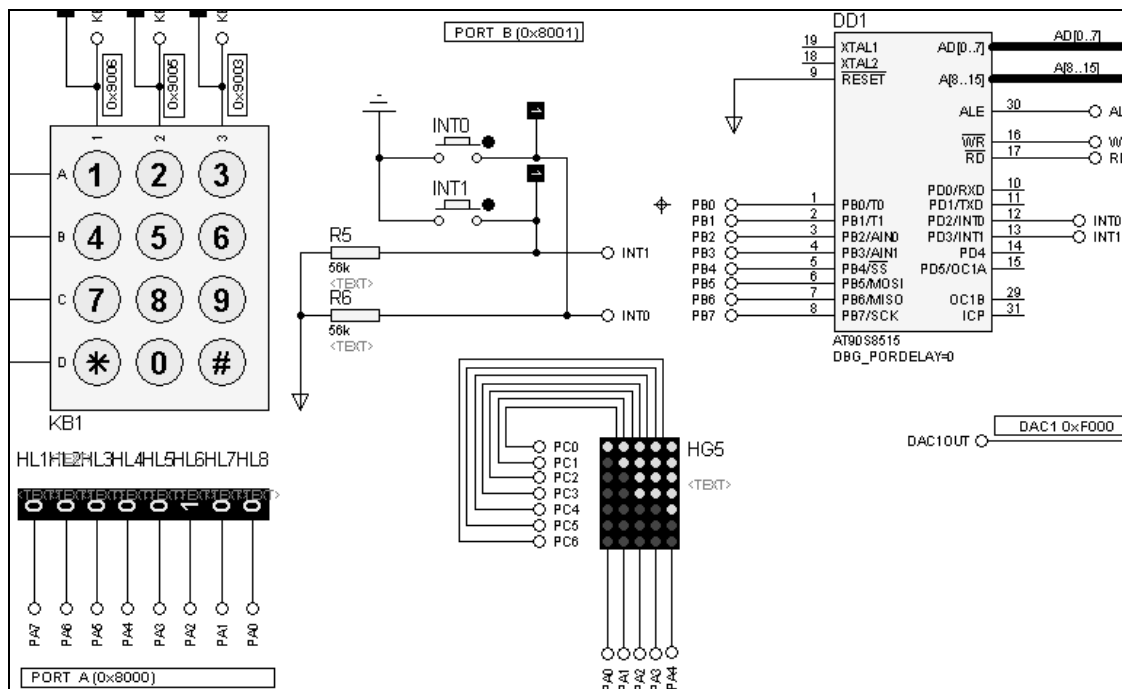


Рисунок 4.40 – Демонстрація роботи з світлодіодною матрицею на віртуальному стенді у середовищі Proteus VSM

4. Блок керування паяльною станцією [74]. Схема блока керування паяльною станцією в програмі ISIS пакета Proteus VSM 7.6 наведена на рисунку 4.41. На схемі, що зображена на рисунку 4.41, як нагрівач працює OV1, з виходу якого знімається напруга, що відповідає напрузі з виходу термопари, підсиленої каскадом на AD8551. На рисунку 4.42 показано моделювання роботи блока керування паяльною станцією у режимі P1. Робоча температура жала повинна бути 180⁰С. Включений нагрівач, поточна температура жала 98⁰С. Світлодіод D1 показує, що блок керування працює на нагрівання. На рисунку 4.43 показаний режим роботи паяльної станції, коли жало нагріто до заданої температури. Світлодіод D1 горить зеленим кольором. Це означає, що нагрівач виключений.

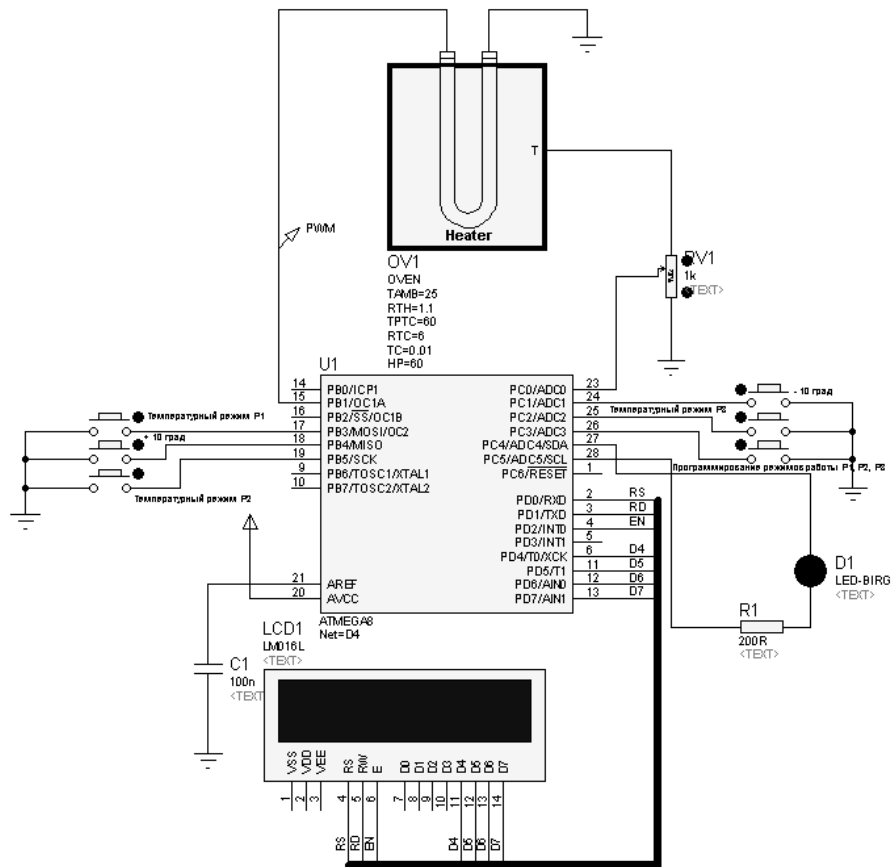


Рисунок 4.41 – Схема блоку керування паяльною станцією в програмі ISIS пакета Proteus VSM7.6

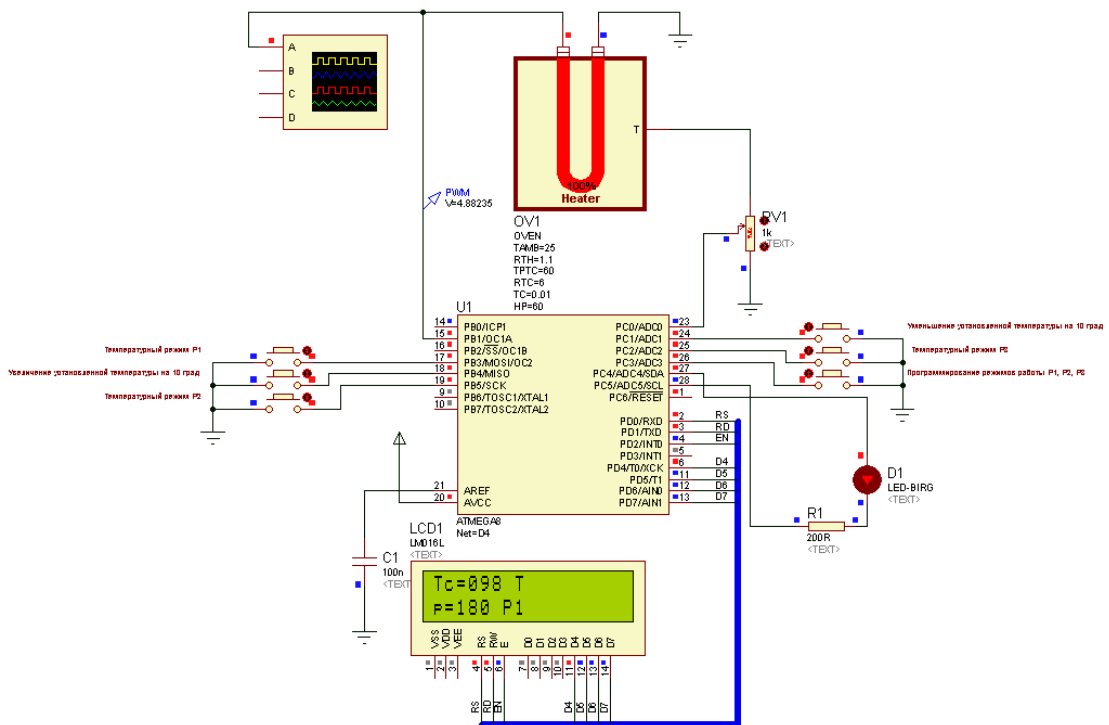


Рисунок 4.42 – Моделювання роботи паяльної станції (режим P1)

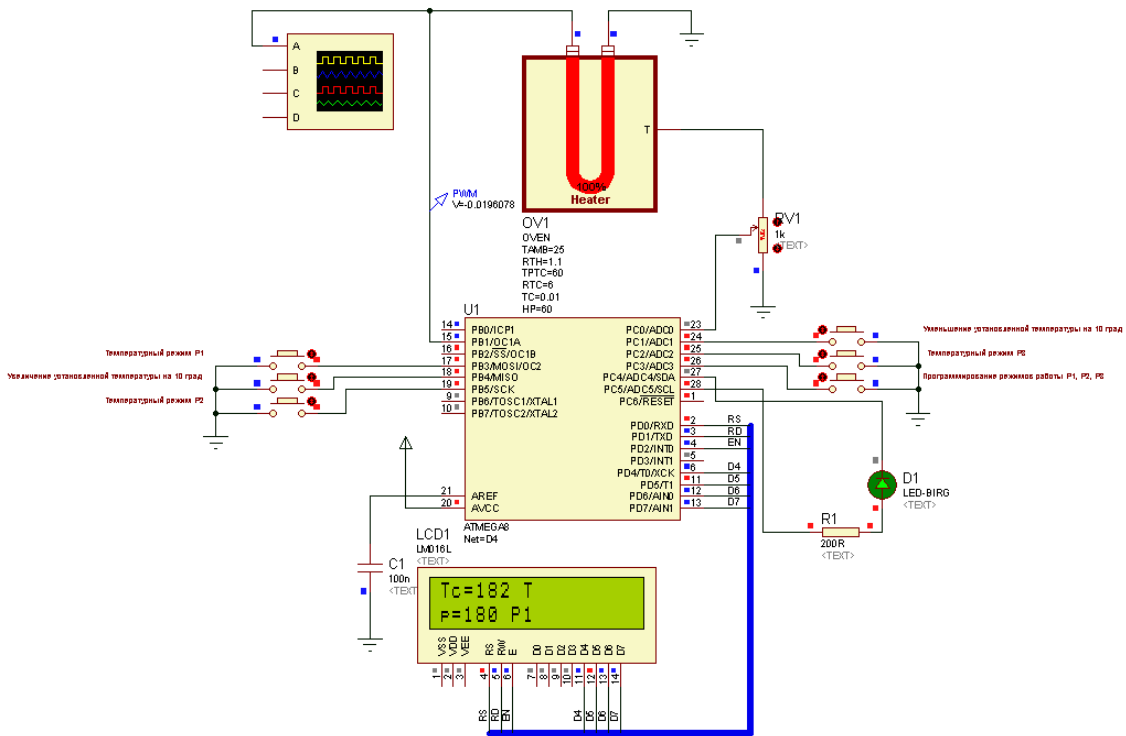


Рисунок 4.43 – Моделювання роботи паяльної станції (режим P1. Жало паяльника нагріто до заданої температури)

5. Вимірювач ємності на мікроконтролері ATtiny2313 [73]. Схема цифрового вимірювача ємності в програмі ISIS пакета Proteus VSM 7.6 наведена на рисунку 4.44. Після запуску моделювання на індикаторах відображається паразитна ємність. Стан схеми показано на рисунку 4.44.

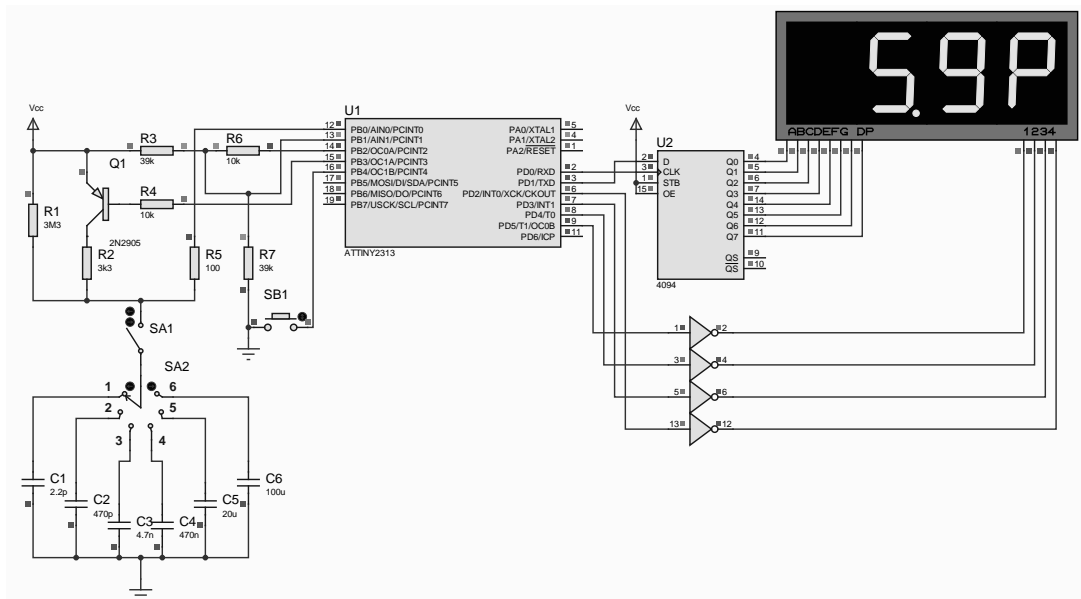


Рисунок 4.44 – Моделювання роботи схеми вимірювача ємності у програмі Proteus (початок роботи)

Після натискання кнопки SB1 вимірювач обнуляється та готовий до роботи (рис.4.45).

Далі за допомогою тумблера SA1 до вимірювача підключається блок конденсаторів C1 – C6 і перемикачем SA2 обирається необхідний конденсатор із блока. На рисунку 4.46 SA2 в положенні "1" (C1=2,2 pF).

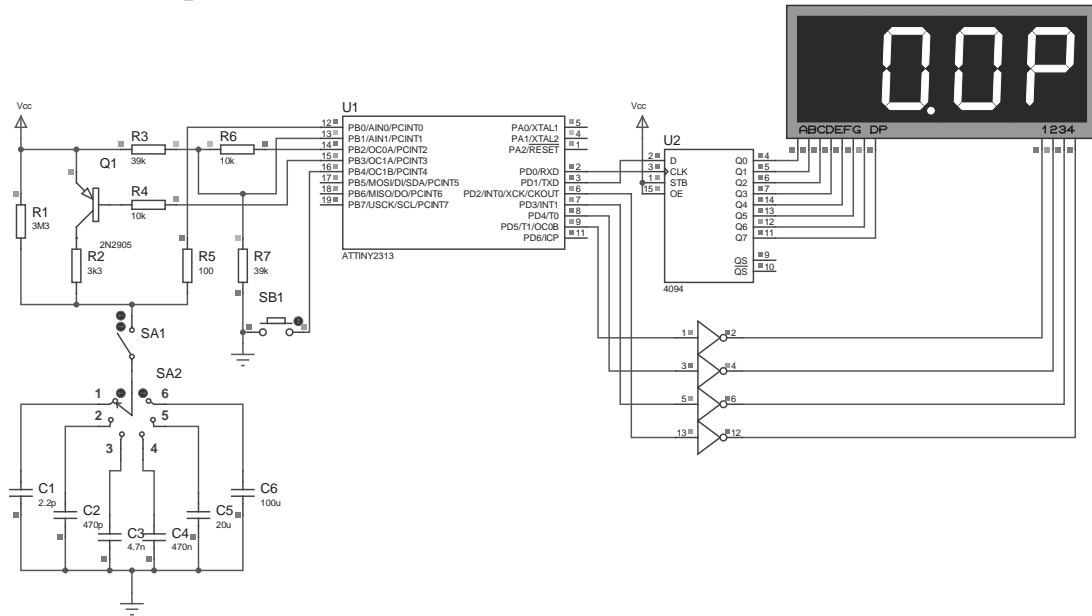


Рисунок 4.45 – Моделювання роботи схеми вимірювача ємності у програмі Proteus (пристрій готовий до роботи)

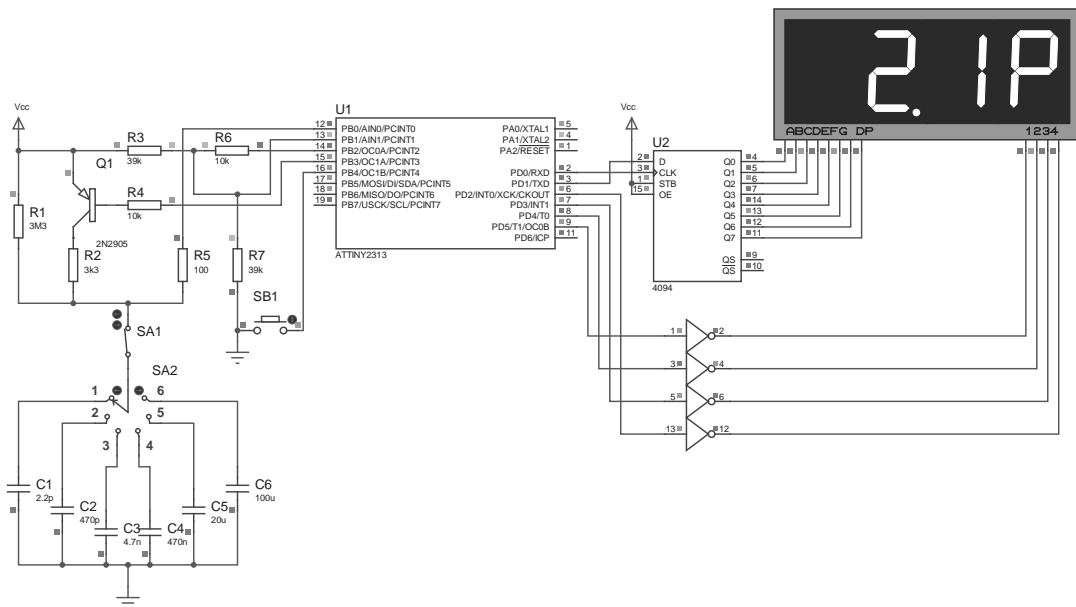


Рисунок 4.46 – Моделювання роботи схеми вимірювача ємності у програмі Proteus (вимірювання C1)

На рисунку 4.47 стан схеми, що відповідає положенню SA2 = "3" (C3=4,70 nF).

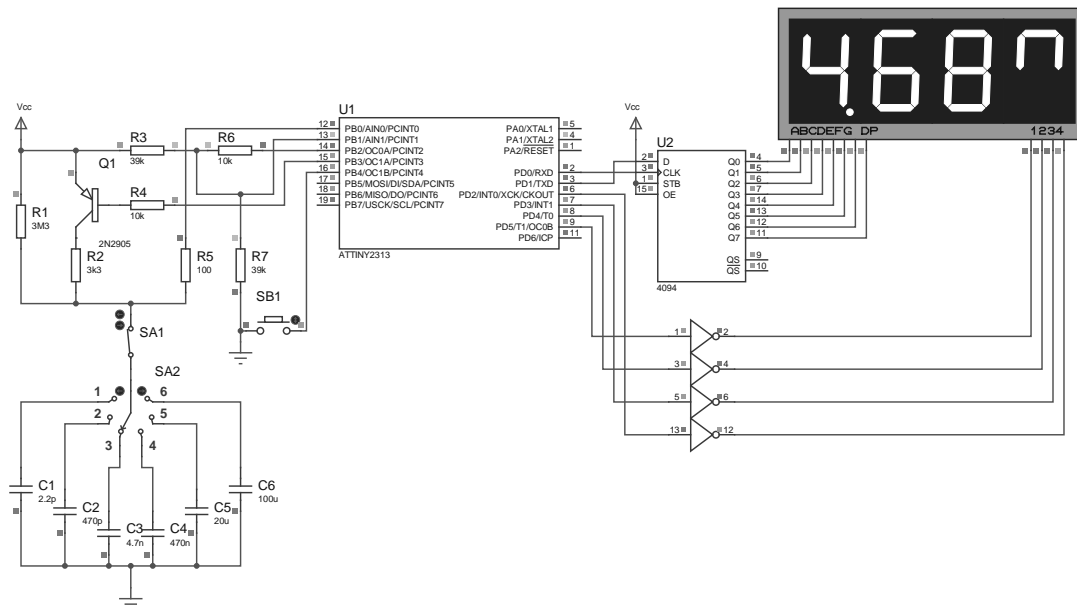


Рисунок 4.47 – Моделювання роботи схеми вимірювача ємності у програмі Proteus (вимірювання C3)

На рисунку 4.48 відображений стан схеми, що відповідає положенню SA2 = "5" (C5=20 μ F).

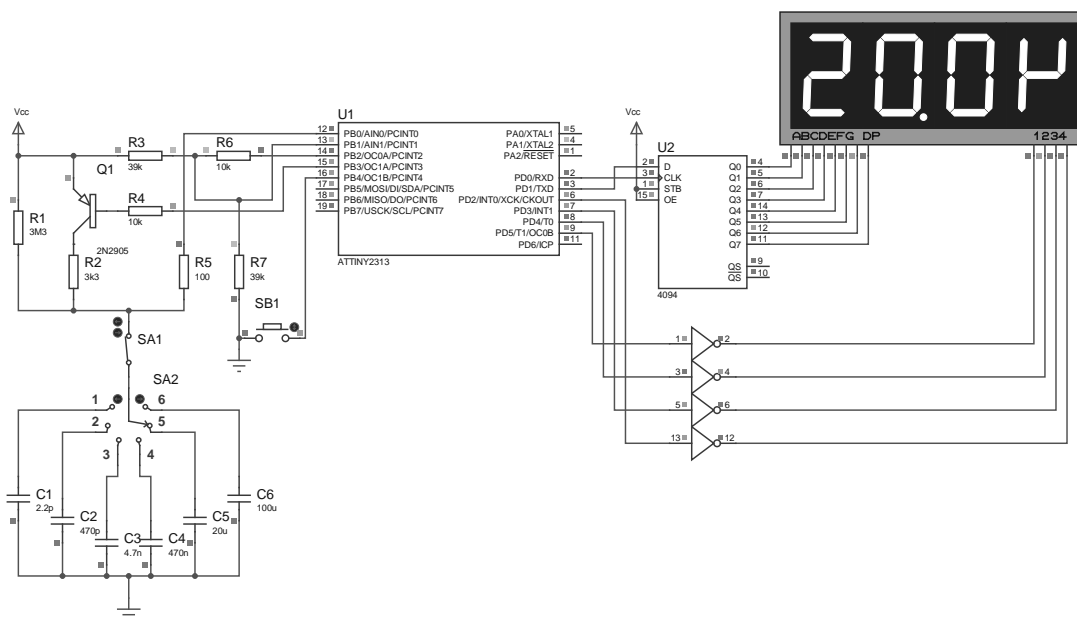


Рисунок 4.48 – Моделювання роботи схеми вимірювача ємності у програмі Proteus (вимірювання C5)

6. Моделювання роботи цифрового термометра/термостата на мікроконтролері AT90S2313 [72]. Схеми цифрового термометра/термостата в програмі ISIS пакета Proteus VSM 7.6 наведена на рисунку 4.49.

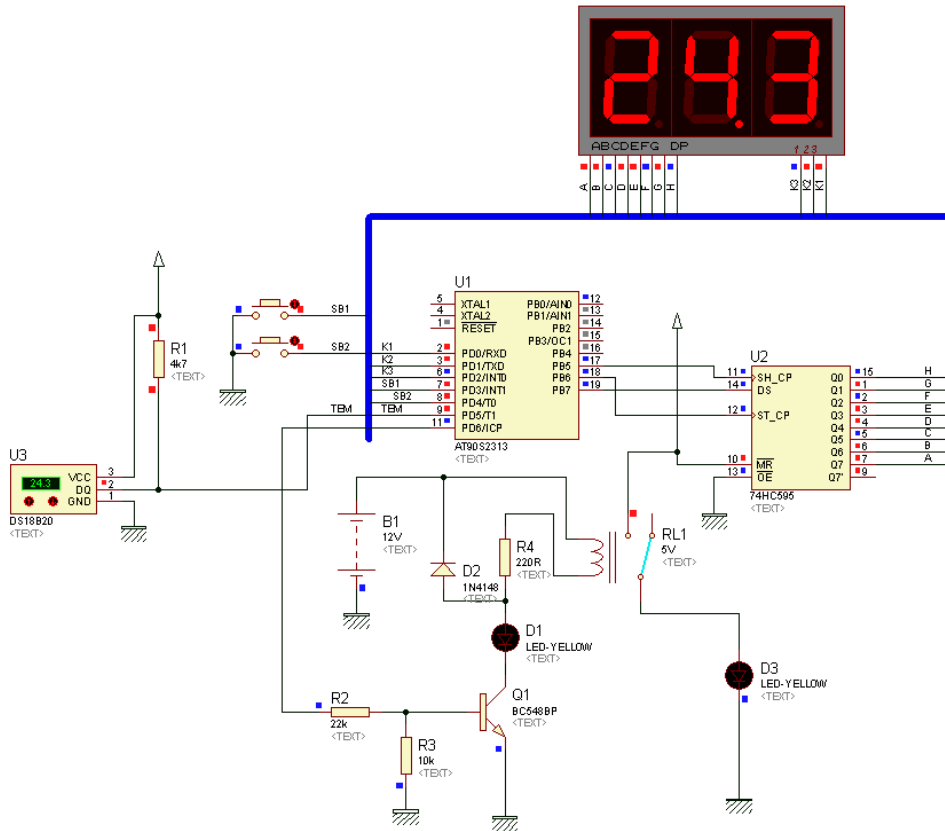


Рисунок 4.49 – Схема цифрового термометра/термостата в програмі ISIS пакета Proteus VSM7.6

На схемі, що зображена на рисунку 4.49, світлодіод D3 є імітатором навантаження, що відображає момент, коли спрацювало реле.

На рисунку 4.50 наведена осцилограма обміну між мікроконтролером та датчиком температури (передається виміряне значення температури), що отримана у середовищі Proteus VSM7.6 під час моделювання роботи схеми.

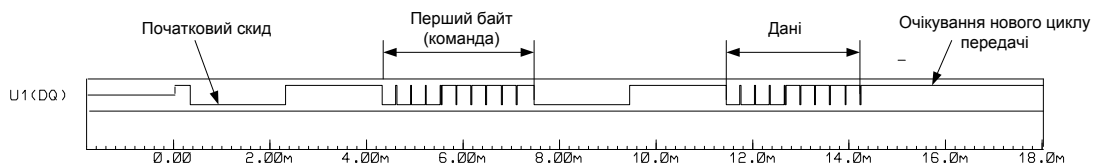


Рисунок 4.50 – Протокол обміну між мікроконтролером та датчиком температури DS1820 шиною 1-WARE

Натиснувши кнопку SA2 можна продивитись температуру терморегулятора. На дисплеї відображається значення температури терморегулятора (рисунок 4.51) до тих пір, поки утримується кнопка. Після відпускання кнопки SA2 прилад повертається в режим термометра.

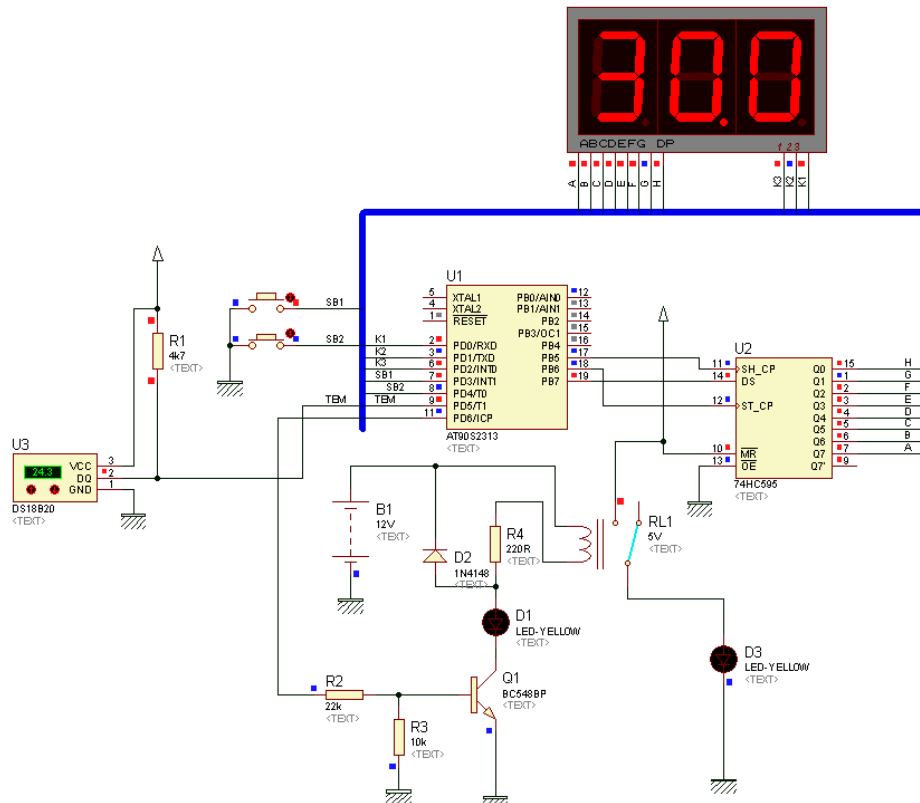


Рисунок 4.51 – Перевірка значення температури терморегулятора, що встановлена (натиснута кнопка SA2)

Для того, щоб встановити температуру терморегулятора, необхідно натиснути та утримувати кнопки SA1 та SA2 одночасно. На індикаторі з'явиться значення температури терморегулятора, що мигає, й прилад перейде у режим встановлення температури. Кнопками SA1 (+) та SA2 (-) встановлюється необхідне значення температури. Повторне одночасне натискання SA1 та SA2 приведе до запису температури терморегулятора до енергонезалежної пам'яті мікроконтролера. На рисунку 4.52 показаний процес встановлення температури терморегулятора 27 градусів. Для переведення приладу в режим термостата необхідно натиснути кнопку SA1. У правому розряді індикатора температури почне мигати десяткова точка, що інформує про включення режиму терморегулятора. Для відключення даного режиму потрібно ще раз натиснути на кнопку SA1. Коли включений режим терморегулятора, прилад на індикатор виводить значення температури датчика, як і в режимі термометра. У залежності від встановленого режиму "Нагрів/Охолодження" відбувається контроль температури й включається/виключається навантаження. На рисунку 4.53 відображений режим, коли температура, що вимірюється (26,8 C⁰), нижча запрограмованого значення 27 C⁰. Мікроконтролер дає команду на включення нагрівача

(світиться D1, спрацьовує реле RL1 й його контакти замикаються, що відображає світіння D3).

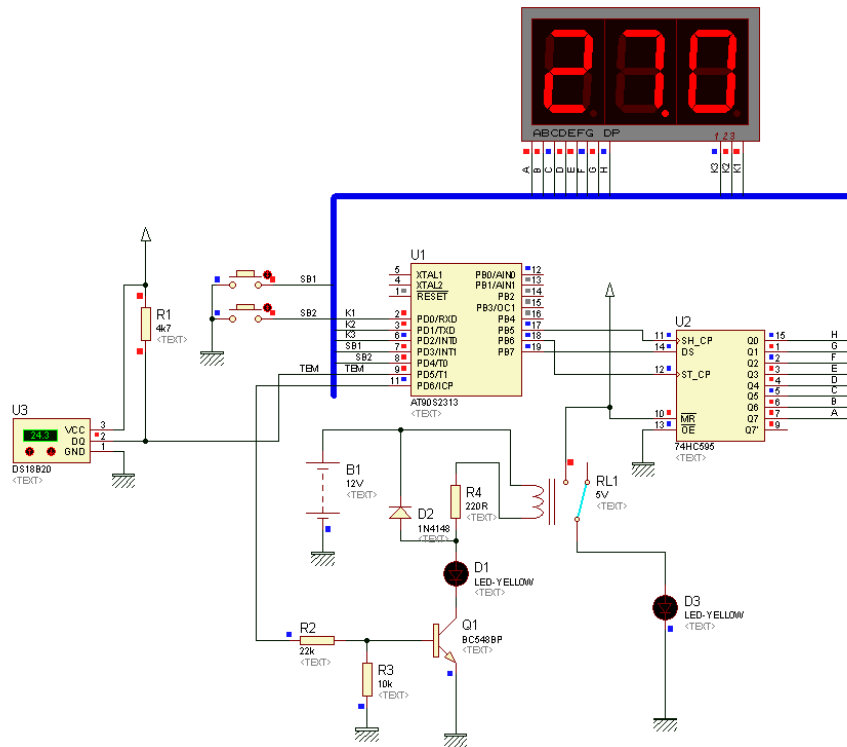


Рисунок 4.52 – Режим програмування терморегулятора. Встановлення температури 27 C⁰ градусів

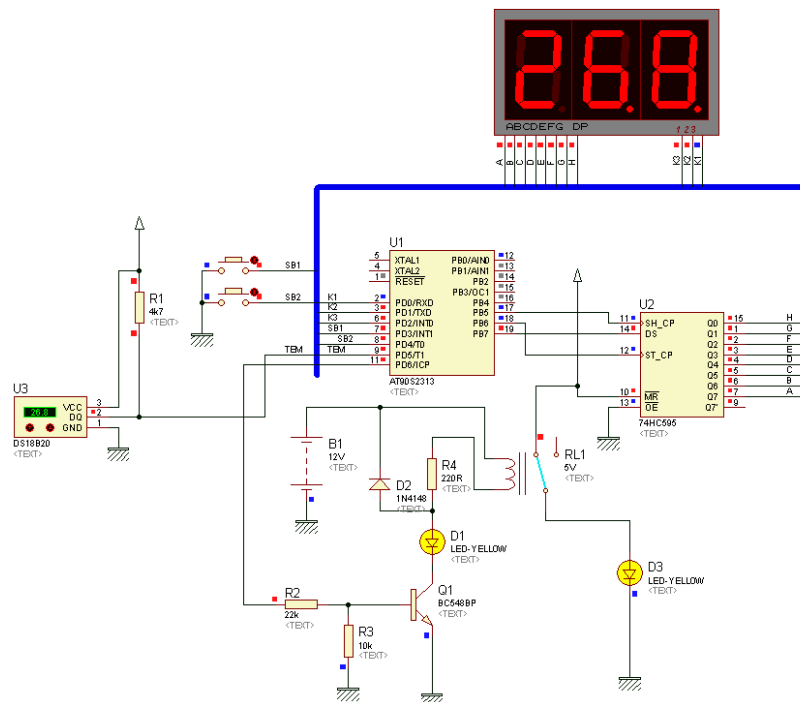


Рисунок 4.53 – Робота термостата в режимі нагріву. Температура середовища нижча запрограмованого значення

На рисунку 4.54 показана робота термостата, коли температура, що вимірюється ($27,1\text{ }^{\circ}\text{C}$) вище запрограмованого значення $27\text{ }^{\circ}\text{C}$. Мікроконтролер відключає навантаження (D1 та D3 не світяться).

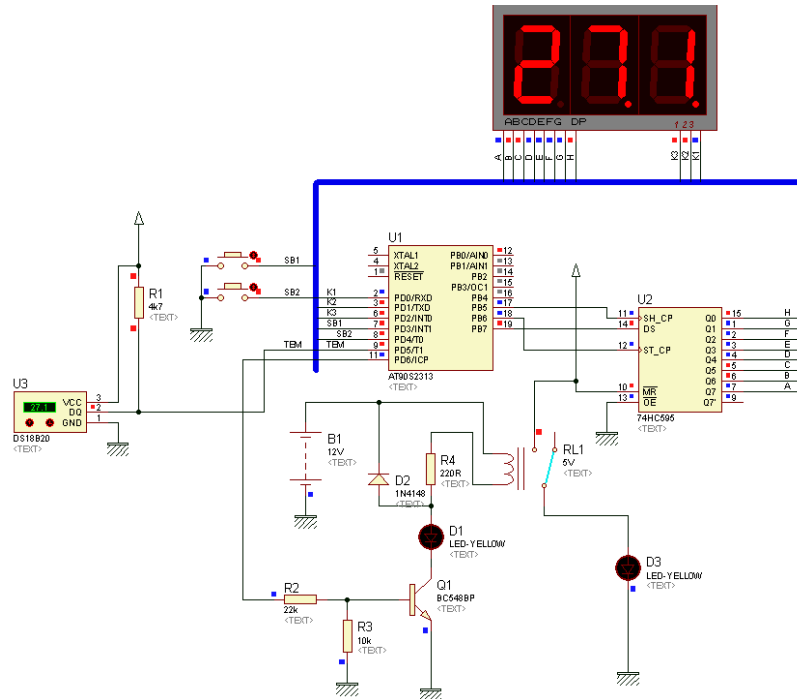


Рисунок 4.54 – Робота термостата в режимі нагріву. Температура середовища вища запрограмованого значення

ІНДИВІДУАЛЬНІ ЗАВДАННЯ НА КУРСОВЕ ПРОЕКТУВАННЯ

Основною задачею індивідуальних завдань є формування практичних навичок розробки мікропроцесорної системи на базі мікроконтролера ATtiny2313 (ATmega8).

За даними технічного завдання розробляється структурна, функціональна та принципова схеми пристрою, розробляється алгоритм роботи мікропроцесорної системи та програмне забезпечення (Assembler, C++, Basic, Pascal). За результатами курсового проекту проводиться імітаційне моделювання роботи мікропроцесорної системи у середовищі Proteus VSM.

Варіанти індивідуальних завдань:

1. Багатофункціональний генератор

Формує залежно від кнопки, що натиснута, 5 видів періодичних сигналів: гармонічний, прямокутний, трикутний, пилкоподібний, трапецеїдальний; для будь-якого сигналу можливо формування декількох (не менше 5) значень періоду.

2. Статистична обробка сигналів

Пристрій реалізує накопичення вхідного сигналу з аналогового входу АЦП, проводить обчислення і передає до персонального комп'ютера інформацію про максимум, мінімум і середнє арифметичне значення сигналу за час спостереження. Обчислена інформація відображається на LCD-індикаторі. Команда запуску чергового вимірювання формується з клавіатури пристрою.

3. Калькулятор

Пристрій реалізовує арифметичне обчислення цифрової інформації, що вводиться з клавіатури, виведення початкових даних і результатів обчислень на LCD-індикатор. Формат подання чисел – знаковий, з фіксованою точкою. Діапазон: – 32768...+32767. Додаткові функції: збереження в пам'яті, округлення, логічні функції, взаємне перетворення кодів в 2, 8, 10, 16-кову системи числення.

4. Частотомір

Пристрій вимірює частоту періодичного сигналу і відображає його на LCD-індикаторі. Передбачити можливість вимірювання високочастотного і низькочастотного сигналів найбільш точними методами.

5. Ігровий тренажер уваги

По екрану LCD-індикатора зліва направо переміщуються символи, відповідні певним клавішам. Завдання гравця – відреагувати на черговий символ натисканням відповідної йому клавіші. Темп появи символів періодично змінюється. Передбачити виведення статистики і музичного оформлення тренажера.

6. Набір і відображення числа за допомогою однієї кнопки

Для введення числа використовується кнопка А. Число набирається в десятковій системі числення (0 – 9). Кожен цифровий розряд вводиться після відповідної кількості коротких натискань. Введення цифри починається з "0", після досягнення "9" – наступна цифра "0". Поточна введена цифра відображається в молодшому розряді. Тайм-аут розпізнавання введення цифри – 1 с. При наступному натисканні – цифра, що вводилась, зміщується вліво і починається набір наступної цифри. Натискання кнопки супроводжується короткочасним звуковим сигналом. Момент закінчення введення цифри (таймаут) також супроводжується короткочасним звуковим сигналом. Введені дані виводяться на статичний або динамічний цифровий індикатор (4 розряди).

7. Набір і відображення тексту на матричному індикаторі

Користувач набирає рядок символів латинського алфавіту за допомогою телефонної клавіатури (за аналогією з набором SMS на мобільному телефоні). Кожній кнопці на клавіатурі може відповідати до 3-х букв. Набрана буква залежить від кількості натискань на кнопку. При кожному натисканні на кнопку циклічно вибирається наступна буква. Набрана буква відображається на матричному індикаторі. Тайм-аут для введення символу 2 с. При натисканні будь-якої іншої кнопки на клавіатурі – символ вважається введеним. Після введення символу він безперервно відображається на індикаторі. Введення рядка закінчується після натискання кнопки # або після введення 10 символів.

8. Гра "Змійка"

Пристрій відображає ефект "повзучої змійки" зі змінною швидкістю, що змінюється, по контуру динамічного індикатора. Швидкість руху змінюється дискретно, залежно від набраної на цифровій клавіатурі цифри: від 1 до 9 сегментів/секунду. Момент введення швидкості супроводжується короткочасним звуковим сигналом.

9. Датчик температури

Пристрій відображає покази цифрового датчика температури DS1621 (інтерфейс I²C) або DS1820 (інтерфейс 1-Wire). Значення температури відображаються на LED або LCD-індикаторі (точність – одиниці °C).

10. Синтезатор звуку

Користувач вводить за допомогою цифрової клавіатури частоту від 100 до 3000 Гц. Значення відображаються на статичному індикаторі. При введенні 5 і більше цифр здійснити зрушення введеного значення ліворуч. При введенні значення частоти використовується кнопка "*". Кнопка "#" дозволяє зафіксувати значення частоти і генерувати звук із заданою частотою. Звук заданої частоти формується до моменту задання нового значення.

11. Тест реакції

Тест складається з 10 кроків. На початку кожного кроку на матричному індикаторі засвічується N точок (від 1 до 9 включно), розташування точок – випадкове. Користувач підраховує кількість N і натискає відповідну кнопку (одноразове натискання) на цифровій клавіатурі. Номер кроку відображається на статичному індикаторі в молодшому розряді. У старшому розряді відображається кількість неправильних відповідей користувача. На динамічному індикаторі відображається час введення останньої відповіді. Вимір часу (від початку тесту) відобразити в секундах з точністю до десятих. Після отримання останньої, 10-ої, відповіді матричний індикатор гасне. Користувач бачить сумарний час проходження тесту, кількість кроків і кількість помилок. Після чого перезапуск програми тестування здійснюється за натисканням будь-якої клавіші на цифровій клавіатурі.

12. Тест швидкості передачі даних

Програма визначає швидкість передачі даних послідовним портом. Користувач з комп'ютера посилає на COM-порт пристрою один з символів "1", "3", "5", "7" або "9" (ASCII коди 0x31, 0x33, 0x35, 0x37, 0x39). Програма визначає, який з символів переданий, і виводить його на статичний індикатор. Швидкість передачі може бути одна з ряду (бодів): 300, 600, 1200, 2400, 4800, 9600 або 19200. Якщо символ не розпізнаний або не є одним з дозволених, на статичний індикатор виводяться прочерки.

13. Генератор випадкових чисел

Після запуску на статичному індикаторі відображається випадкове число. Діапазон чисел: 0 – 255 включно.

14. Терморегулятор

Пристрій повинен забезпечувати керування швидкістю обертання вентилятора залежно від температури. Встановлення частоти обертання виконується за допомогою регулювання подаваної потужності через широтно-імпульсну модуляцію. Реальна частота обертання контролюється датчиком обертання. Температура вимірюється цифровим датчиком температури. Швидкість обертання вентилятора довільна, але повинна змінюватися у бік збільшення при наростанні температури і у бік зменшення при падінні температури. Дисплей має два режими індикації: режим індикації температури в градусах, режим індикації частоти обертання в оборотах в хвилину. При натисканні на кнопку 1 режим індикації перемикається.

15. Блок керування технологічним процесом

Пристрій підраховує кількість перетинів деяким предметом певної межі в одну чи іншу сторони, запам'ятовує та відображає їх на індикаторі. Передача поточної інформації на персональний комп'ютер. Кнопкою "Пуск" починається робота пристрою. Вибір режиму здійснюється кнопкою "Режим".

16. Автомобільний світлодинамічний пристрій

Пристрій призначений для дублювання сигналу лівого та правого поворотів, дублювання габаритних вогнів, дублювання сигналу "Стоп". Режим – динамічна індикація.

17. Бортовий комп'ютер автомобіля

Пристрій має такі функціональні можливості: індикація поточного часу; індикація температури в чотирьох точках; звукова сигналізація при підвищенні температури; індикація напруги в бортовій мережі автомобіля; звукова сигналізація при нарузі бортової мережі нижче 10 В.

18. Охоронна сигналізація

Після включення сигналізації відбувається затримка у 15 секунд, і пристрій входить до режиму охорони (миготіння світлодіоду). У режимі охорони МК знаходиться в енергозберігаючому режимі. При спрацьовуванні одного з чотирьох датчиків включається сирена, включається світлова сигналізація. Через хвилину відключаються сирена, світлова індикація й перевіряється чи усунута причина

тривоги. Якщо так, то МК переходить у режим очікування, якщо ні, то через 30 с включається сирена, світлова індикація. Цикли роботи повторюються.

19. Кодовий замок

Код п'ятизначний й складається з цифр, що не повторюються. Правильно набраний код супроводжується звуковим сигналом. При неправильно набраній цифрі відбувається скид набраного коду. При неправильному наборі п'ять раз спрацьовує звуковий сигнал "тривоги" на протязі 5 хвилин. Контролер сприймає код через 10 хвилин. Користувач може встановлювати свій код (режим програмування). Цифрова індикація режиму роботи та дзвінок.

20. Таймер

Режим роботи задається розкладом. Розклад попередньо набирається в спеціальній програмі й відправляється в мікроконтролер через послідовний порт персонального комп'ютера. Кількість навантажень – 7, допустима потужність – 1кВт. Точність задання часу – 1секунда.

21. Тестер транзисторів

Пристрій автоматично виконує тестування біполярних транзисторів та відображає провідність транзистора (n-p-n або p-n-p), розташування колектора емітера, бази транзистора.

22. Тестер кабелів UTP

Пристрій тестує кабелі за стандартом 568В (комп'ютер–концентратор), кабелі за стандартом 568А (комп'ютер–комп'ютер). На LED або LCD-індикаторі відображаються результати.

23. Цифровий тахометр

Пристрій вимірює частоту обертання вала двигуна і відображає на LED або LCD-індикаторі. Діапазон вимірювання – 100 до 7000 об/хв⁻¹.

24. Цифровий термостат

Термостат керує нагрівником і підтримує стаке значення температури в діапазоні +5 - +95 °С з точністю 1 °С. Значення температури вводиться з клавіатури пристрою і відображається на цифровому індикаторі.

25. Велокомп'ютер

Пристрій вимірює і відображає на індикаторі швидкість, шлях, проміжний шлях. Діапазон вимірювання швидкості – 3...60 км/год. з точністю 1км/год.; діапазон вимірювання відстані, км (з кроком 1км) – 250.

26. Електронний годинник

Функціональні можливості: годинник має 3 кнопки для встановлення годин (1), хвилин (2), секунд (3); режим роботи встановлюється при натисканні на відповідну кнопку.

27. Лічильник витрат стрічки магнітофона

Функціональні можливості блока лічби: лічба відбувається в прямому та зворотному напрямках, запис в EEPROM поточного числа, лічба при повторній подачі живлення починається з останнього значення, максимальне число – 9999, відображення поточних значень на LED-індикаторі.

28. Ігровий автомат

Алгоритм роботи пристрою такий: автомат знаходиться в режимі очікування (на індикаторі відображається демонстраційний режим); при натисканні кнопки "Start" розпочинається ігровий режим, при опусканні монети на індикаторі відображається число, яке формує генератор випадкових чисел. Автомат видає таку кількість монет, якщо на індикаторі відображаються числа: 777 – 200 шт, 999 – 100 шт, 000 – 20 шт, 555 – 10 шт, 707 – 7 шт, 00X – 5 шт, 077 – 3 шт, 0XX – 2 шт, 77X – 1 шт; через 1 с після закінчення гри автомат переходить в режим очікування. Блок керування ігровим автоматом відображає кількість опущених монет та кількість виграних монет.

29. Автомат світлових ефектів

Режими роботи: рухомий вогонь, реверс рухомого вогню, рухома тінь, реверс рухомої тіні, накопичування, реверс накопичування, мерехтіння. Кількість каналів – 8.

30. Система кодового доступу

Система використовує 9 ключів iButton, плюс один майстер-ключ. Коди ключів заносяться до EEPROM під номерами 1 до 9; код майстер-ключа, занесений до EEPROM мікроконтролера, не може бути змінений; програмування нових ключів або стирання старих може бути проведено тільки при наявності майстер-ключа; майстер-ключ може використовуватись для відкриття замка.

СПИСОК ЛІТЕРАТУРИ

1. Ан П. Сопряжение ПК с внешними устройствами: Пер. с англ. / П. Ан. – М. : ДМК Пресс, 2001. – 320 с: ил. – ISBN 5-94074-076-6.
2. Антипенский Р. В. Схемотехническое проектирование и моделирование радиоэлектронных устройств / Р. В. Антипенский, А. Г. Фалин – М. : Техносфера, 2007. – 128 с. – ISBN 978-5-94836-130-7.
3. Баранов В. Н. Применение микроконтроллеров AVR: схемы, алгоритмы, программы / В. Н. Баранов. – М. : Издательский дом "Додека-XXI", 2004. – 288 с. (серия "Мировая электроника"). – ISBN 5-94120-075-7.
4. Барышев И. В. Прикладные вопросы цифровой обработки информации. Часть 1. Применение микроконтроллеров в РТС сбора, обработки и передачи информации / И. В. Барышев, А. В. Мазуренко, О. А. Горбуненко. – Харьков : ХАИ, 2006. – 112 с.
5. Баширов С. Р. Бытовая электроника. Занимательные устройства своими руками / С. Р. Баширов, А. С. Баширов. – М. : Эксмо, 2008. – 176 с. – ISBN 978-5-699-23966-5.
6. Белов А. В. Конструирование устройств на микроконтроллерах / А. В. Белов. – СПб. : Наука и Техника, 2005. – 256 с. – ISBN 5-94387-155-1.
7. Белов А. В. Микроконтроллеры AVR в радиолюбительской практике / А. В. Белов. – СПб. : Наука и Техника, 2007. – 352 с. – ISBN 978-5-94387-365-2.
8. Белов А. В. Самоучитель по микропроцессорной технике / А. В. Белов. – СПб. : Наука и Техника, 2003. – 224 с. – ISBN 5-94387-084-9.
9. Белов А. В. Самоучитель разработчика устройств на микроконтроллерах AVR / А. В. Белов. – СПб. : Наука и Техника, 2008. – 544 с. – ISBN 978-5-94387-363-8.
10. Белов А. В. Создаем устройства на микроконтроллерах / А. В. Белов. – СПб. : Наука и Техника, 2007. – 304 с. – ISBN 978-5-94387-364-3.
11. Богатырев Е. А. Энциклопедия электронных компонентов. Большие интегральные схемы / Е. А. Богатырев, В. Ю. Ларин,

- А. Е. Лякин; под ред. А. Н. Еркина. – Т. 1. – М. : ООО "МАКРОТИМ", 2006. – 224 с. – ISBN 5-9900833-1-9 (978-5-9900833-1-8).
12. Болл С. Р. Аналоговые интерфейсы микроконтроллеров / С. Р. Болл. – М. : Издательский дом "Додэка-XXI", 2007. – 360 с. – ISBN 978-5-94120-142-6.
 13. Буданов А. Н. Средства разработки и отладки программного обеспечения промышленных контроллеров на базе 8/16-разрядных микропроцессоров фирмы Motorola / А. Н. Буданов // Мир компьютерной автоматизации. [Электронный ресурс]. – Режим доступа до журн.: <http://www.mka.ru/>, вільний. – Загл. з екрана. – Мова рос.
 14. Вальпа О. Д. Полезные схемы с применением микроконтроллеров и ПЛИС / О. Д. Вальпа – М. : Издательский дом "Додэка-XXI", 2006. – 416 с. – (Серия "Программируемые системы"). – ISBN 5-94120-129-X.
 15. Василенко В. Кабельный пробник со звуковой индикацией / В. Василенко. // Схемотехника. – 2006, №10. – С. 38-42.
 16. Виноградов Ю. А. Радиоловительские конструкции в системах контроля и защиты / Ю. А. Виноградов – М. : Солон-Р, 2001. – 192 с. – ISBN 5-93455-070-5.
 17. Волович Г. И. Схемотехника аналоговых и аналого-цифровых электронных устройств / Г. И. Волович. – М. : Издательский дом "Додэка-XXI", 2005. – 528 с. – ISBN 5-94120-074-9.
 18. Гнатек Ю. Р. Справочник по цифроаналоговым и аналогоцифровым преобразователям; пер. с англ. / Ю. Р. Гнатек; под ред. Ю. А. Рюжина. – М. : Радно и связь, 1982. – 552 с.
 19. Голубцев М. С. Микроконтроллеры AVR: от простого к сложному / М. С. Голубцев. – М. : Солон-Пресс, 2003. – 288 с. – ISBN 5-98003-034-4.
 20. Гребнев В. В. Микроконтроллеры семейства AVR фирмы Atmel / В. В. Гребнев. – М. : ИП Радиософт, 2002. – 176с. – ISBN 5-93037-091-5.
 21. Датчики : Справочник / Под ред. З. Ю. Готра, О. И. Чайковского. – Львов : Каменяр, 1995. – 312 с. – ISBN 5-7745-0233-3.
 22. Джонсон Г. Конструирование высокоскоростных цифровых устройств: начальный курс черной магии; пер. с англ. / Г. Джонсон, М. Грэхем. – М. : Издательский дом "Вильямс",

2006. – 624 с. – ISBN 5-8459-0807-8.
23. Дмитриев С. Сотовый телефон – "электронная книга" / С. Дмитриев // Радио. – 2005. – №11. – С. 26–28.
 24. Евстифеев А. В. Микроконтроллеры AVR семейств Tiny и Mega фирмы "ATMEL" / А. В. Евстифеев. – М. : Издательский дом "Додэка-XXI", 2004. – 560 с. – ISBN 5-94120-081-1.
 25. Заец Н. И. Радиолобительские конструкции на PIC-микроконтроллерах. С алгоритмами работы программ и подробными комментариями к исходным текстам. Книга 1 / Н. И. Заец. – М. : СОЛОН-Пресс, 2005. – 192 с. – ISBN 5-98003-238-X.
 26. Зубчук В. И. Справочное пособие по цифровой схемотехнике / В. И. Зубчук, В. П. Сигорский, А. Н. Шкуро. – К. : Техника, 1990. – 448 с. – ISBN: 5-335-00584-X.
 27. Иванов В. И. Полупроводниковые оптоэлектронные приборы. Справочник / В. И. Иванов, А. И. Аксенов, А. М. Юшин. – М. : Энергоатомиздат, 1984. – 311 с.
 28. Иванов Ю. И. Микропроцессорные устройства систем управления: Учебное пособие / Ю. И. Иванов, В. Я. Югай. – Таганрог : Изд-во ТРТУ, 2005. – 133 с. – ISBN 5-8327-0206-9.
 29. Кардашев Г. А. Цифровая электроника на персональном компьютере Electronics Workbench и Micro-Cap / Г. А. Кардашев. – М. : Горячая линия-Телеком, 2003. – 311с. – ISBN 5-93517-140-6.
 30. Катцен С. PIC-микроконтроллеры. Все, что вам необходимо знать / С. Катцен; пер. с англ. Евстифеева А. В. – М. : Издательский дом "Додэка-XXI", 2008. – 656 с. : ил. (Серия "Программируемые системы"). – ISBN 978-5-94120-134-1.
 31. Кеон Д. (Кеон, Дж.). OrCAD Pspice. Анализ электрических цепей (+ DVD) / Д. Кеон. – СПб. : Издательство "Питер", 2008. – 640 с. – ISBN 978-5-388-00023-1.
 32. Контроллеры для организации USB-связи // Рынок Микроэлектроники [Электронный ресурс]. – Режим доступа до журн.: <http://www.gaw.ru/html.cgi/txt/ic/Atmel/micros/usb/start.htm>, вільний. – Загл. з екрана. – Мова рос.
 33. Корнеев В. В. Современные микропроцессоры / В. В. Корнеев, А. В. Киселев. – СПб. : БХВ-Петербург, 2003. – 448 с. – ISBN 5-94157-385-5.

34. Кравченко А. В. 10 практических устройств на AVR-микроконтроллерах. Книга 1 / А. В. Кравченко. – М. : Издательский дом "Додэка-XXI, К. : МК-Пресс, 2008. – 224 с. – ISBN 978-5-94120-205-8, ISBN 978-966-8806-41-4.
35. Кравченко А. В. 10 практических устройств на AVR-микроконтроллерах. Книга 2 / А. В. Кравченко. – К. : МК-Пресс, СПб. : Корона-Век, 2009. – 320 с. – ISBN 978-966-8806-58-2, ISBN 978-5-7931-0532-3.
36. Краткий учебный курс PROTEUS [Электронный ресурс] / Русское руководство для начинающих. – Режим доступа : <http://proteus123.narod.ru>, вільний. – Загл. з екрана. – Мова рос.
37. Лебедев М. Б. CodeVisionAVR : пособие для начинающих / М. Б. Лебедев. – М. : Додэка-XXI, 2008. – 592 с. – ISBN 978-5-94120-192-1.
38. Левшина Т. С. Электрические измерения физических величин: Измерительные преобразователи / Т. С. Левшина, П. В. Новицкий. – Л. : Энерго-атомиздат, 1983. – 320 с.
39. Магда Ю. С. Микроконтроллеры серии 8051: практический подход. / Ю. С. Магда. – М. : ДМК Пресс. 2008. – 228 с. – ISBN 5-94074-394-3
40. 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.
41. Микросхемы АЦП и ЦАП. Справочник. – М. : Издательский дом "Додэка-XXI", 2005. – 432 с. – ISBN 5-94120-091-9.
42. Микросхемы и средства отладки USB // СМД Компонент [Электронный ресурс]. – Режим доступа: <http://www.smd-component.ru/catalog/16usb.htm>, вільний. – Загл. з екрана. – Мова рос.
43. Микушин А. В. Занимательно о микроконтроллерах / А. В. Микушин. – СПб. : БХВ-Петербург, 2006. – 432 с. – ISBN 5-94157-571-8.
44. Мікропроцесорна техніка : Підручник / Ю. І. Якименко, Т. О. Терещенко, Є. І. Сокол та інш. – К. : ІВЦ Видавництво "Політехніка"; "Кондор", 2004. – 440 с. – ISBN 966-622-135-7.
45. Мортон Дж. Микроконтроллеры AVR. Вводный курс /

- Дж. Мортон; пер. с англ. – М.: Издательский дом "Додека-XXI", 2006. – 272 с. – ISBN 5-94120-096-X.
46. Потапчук М. ИК-барьер на микроконтроллере / М. Потапчук // Радиомир. – 2005. – №6. – С. 11–13.
 47. Программирование на языке С для AVR и PIC микроконтроллеров / Сост. Ю. А. Шпак – К : МК-Пресс, 2006. – 400 с. – ISBN 966-8806-16-6.
 48. Проектирование встроенных систем на микроконтроллерах STMicroelectronics / [Бабешко Е. В., Желтухин А. В., Куланов В. А. и др.]; под ред. В. С. Харченко, А. А. Орехова. – Харьков : Национальный аэрокосмический университет "ХАИ", 2007. – 197 с. – ISBN 978-966-2982-32-9.
 49. Радиокот [Электронный ресурс] / Proteus – первое знакомство. – Режим доступа: <http://radiokot.ru/start/soft/proteus/01>, вільний. – Загл. з екрана. – Мова рос.
 50. Разевиг В. Д. Схемотехническое моделирование с помощью Micro-Cap 7 / В. Д. Разевиг. – М. : Горячая линия-Телеком, 2003. – 368 с. – ISBN 5-93517-127-9
 51. Рахно Э. Микроконтроллеры Microchip – с чего начать? / Э. Рахно // CHIP NEWS Украина. – 2008. – №8(78). – С. 20–21.
 52. Ревич Ю. В. Практическое программирование микроконтроллеров Atmel AVR на языке асемблера / Ю. В. Ревич. – СПб. : БХВ-Петербург, 2008. – 384 с. (Аппаратные средства). – ISBN 978-5-9775-0277-1.
 53. Системы автоматизированного проектирования фирмы Altera MAX+plus II и QuartusII. Краткое описание и самоучитель / [Комолов Д. А., Мьяльк Р. А., Зобенко А. А., Филиппов А. С.]. – М. : ИП РадиоСофт, 2002. – 352 с. – ISBN 5-93037-098-2.
 54. Схемотехніка електронних систем : У 3 кн. Кн. 1. Аналогова схемотехніка та імпульсні пристрої : Підручник / [В. І. Бойко, А. М. Гуржій, В. Я. Жуйков та ін.]. – К. : Вища шк., 2004. – 366 с. – ISBN 966-642-192-5.
 55. Схемотехніка електронних систем : У 3 кн. Кн. 2. Цифрова схемотехніка : Підручник / [В. І. Бойко, А. М. Гуржій, В. Я. Жуйков та ін.]. – К. : Вища шк., 2004. – 423 с. – ISBN 966-642-200-X.
 56. Схемотехніка електронних систем : У 3 кн. Кн. 3. Мікропроцесори та мікроконтролери : Підручник / [В. І. Бойко, А. М. Гуржій,

- В. Я. Жуйков та інш.]. – К. : Вища шк., 2004. – 399 с. – ISBN 966-642-193-3.
57. Точки Р. Д. Цифровые системы. Теория и практика / Р. Д. Точки, Н. С. Уидмер ; пер. с англ. – М. : Издательский дом "Вильямс", 2004. – 1024 с. – ISBN 5-8459-0586-9.
58. Трамперт В. AVR-RISC микроконтроллеры ; пер. с нем. / В. Трамперт. – К. : МК-Прес, 2006. – 464 с. – ISBN 966-8806-07-7, 3-7723-5476-9.
59. Трамперт В. Измерение, управление и регулировка с помощью AVR микроконтроллеров / В. Трамперт ; пер. с нем. – К. : МК-Пресс, 2006. – 208 с. – ISBN 966-8806-14-X.
60. Угрюмов Е. П. Цифровая схемотехника / Е. П. Угрюмов. – СПб. : БХВ-Петербург, 2001. – 528 с. – ISBN 5-8206-0100-9.
61. Уилкинсон Б. Основы проектирования цифровых схем; пер. с англ. / Б. Уилкинсон. – М. : Издательский дом "Вильямс", 2004. – 320 с. – ISBN 5-8459-0685-7.
62. Уилмсхерст Т. Разработка встроенных систем с помощью микроконтроллеров PIC. Принципы и практические примеры ; пер. с англ. / Т. Уилмсхерст. – К. : МК-Пресс, СПб. : КОРОНА-ВЕК, 2008. – 544 с. – ISBN 970-5-903383-61-0 (рус.), ISBN 978-966-8806-46-9 (укр.).
63. Фрике К. Вводный курс цифровой электроники / К. Фрике; пер. с нем. – М. : Техносфера, 2003. – 432 с. – ISBN 5-94836-015-6.
64. Фромберг Э. М. Конструкции на элементах цифровой техники / Э. М. Фромберг. – М. : Горячая линия-Телеком, 2002. – 264 с. – ISBN 5-93517-077-9.
65. Фрунзе А. В. Микроконтроллеры? Это же просто! Т. 1 / А. В. Фрунзе. – М. : ООО "ИД СКИМЕН", 2002. – 336 с. – ISBN 5-94929-002-X (Т.1).
66. Фрунзе А. В. Микроконтроллеры? Это же просто! Т. 2 / А. В. Фрунзе. – М. : ООО "ИД СКИМЕН", 2002. – 392 с. – ISBN 5-94929-003-8 (Т.2).
67. Фрунзе А. В. Микроконтроллеры? Это же просто! Т. 3 / А. В. Фрунзе. – М. : ООО "ИД СКИМЕН", 2003. – 224 с. – ISBN 5-94929-003-7 (Т.3).
68. Хартов В. Я. Микроконтроллеры AVR. Практикум для начинающих / В. Я. Хартов. – М. : Изд-во МГТУ им.

- Н. Э. Баумана, 2007. – 240 с. – ISBN 978-5-7038-3051-2.
69. Цирульник С. М. Автоматизация проектирования микропроцессорных систем контролю доступа та охорони / С. М. Цирульник, С. І. Перевозніков, В. С. Озеранський // Вісник Вінницького політехнічного інституту. – 2009, №1. – С. 10–15.
 70. 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 автоматичного керування (Автоматика 2006). – Вінниця : Універсум-Вінниця. – 2007. – С. 526–530. – ISBN 978-966-641-210-5.
 71. Цирульник С. М. Комп'ютеризований лабораторний віртуальний стенд / С. М. Цирульник, В. І. Роптанов // Збірник тез доповідей IV МНТК "Сучасні проблеми радіоелектроніки, телекомунікацій та приладобудування (СПРТП-2009)". Частина 1. – Вінниця, 2009. – С. 103.
 72. Электронный портал KAZUS.RU / Проекты PROTEUS. [Електронний ресурс]. – Режим доступу: <http://kazus.ru/forums/showthread.php?t=8259>, вільний. – Загл. з екрана. – Мова рос.
 73. Электронный портал Радиодед/ Цифровой измеритель ёмкости. [Електронний ресурс]. – Режим доступу: <http://radioded.ru/content/view/77/49/>, вільний. – Загл. з екрана. – Мова рос.
 74. Электронный портал Радиокот / Цифровая паяльная станция своими руками [Електронний ресурс]. – Режим доступу: <http://radiokot.ru/lab/controller/10>, вільний. – Загл. з екрана. – Мова рос.
 75. 2 AMPS Silicon bridge rectifiers. [Електронний ресурс] // Datasheet. No. BRSB-200-1C-Diotec Electronics corp. – 2 р. – Режим доступу: <http://www.diotec-usa.com>, вільний. – Загл. з екрана. – Мова англ.
 76. AVRCalc [Електронний ресурс] / Calculator for the AVR Microcontroller. – Режим доступу: <http://clsq1.med-info.com>, вільний. – Загл. з екрана. – Мова англ.
 77. LM78XX/LM78XXA. 3-Terminal 1A Positive Voltage Regulator. [Електронний ресурс] // Datasheet. Rev. 1.0.1; 6'06. – Fairchild Semiconductor corp. – 28 р. – Режим доступу: www.fairchildsemi.com/ds/LM/LM7805.pdf, вільний. – Загл. з екрана. – Мова англ.

78. MAX7219/MAX7221. Serially Interfaced, 8-Digit LED Display Drivers [Електронний ресурс] // Datasheet. Rev. 4; 7/03. – Maxim corp. – 16 р. – Режим доступу: <http://datasheets.maxim-ic.com/en/ds/MAX7219-MAX7221.pdf>, вільний. – Загл. з екрана. – Мова англ.
79. Open System [Електронний ресурс] / Учебно-отладочный стенд EV8031/AVR. – Режим доступу: <http://opensys.com.ua>, вільний. – Загл. з екрана. – Мова рос.
80. Series PVT312. Microelectronic Power IC. [Електронний ресурс] // Datasheet. No. PD 10038D; 8/00. – International Rectifier. – 5 р. – Режим доступу: <http://www.irf.com/product-info/datasheets/data/pvt312.pdf>, вільний. – Загл. з екрана. – Мова англ.
81. Three Digit LED Displays [Електронний ресурс] // Yellow Stone corp. – Datasheet. – 1 р. – Режим доступу: <http://www.ystone-led-capacitor-manufacturer.com>, вільний. – Загл. з екрана. – Мова англ.

Навчальне видання

**Цирульник Сергій Михайлович
Лисенко Геннадій Леонідович**

ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ

Навчальний посібник

Редактор В. Дружиніна

Оригінал-макет підготовлено С. Цирульник

Підписано до друку 29.06.2010 р.
Формат 29,7×42 1/4. Папір офсетний.
Гарнітура Times New Roman.
Друк різнографічний. Ум. друк. арк.
Наклад. 300 (1-й запуск 1–100) прим. Зам. №

Вінницький національний технічний університет,
навчально-методичний відділ ВНТУ.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к.114.
Тел. (0432) 59-85-32
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к.114.
Тел. (0432) 59-87-38
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.