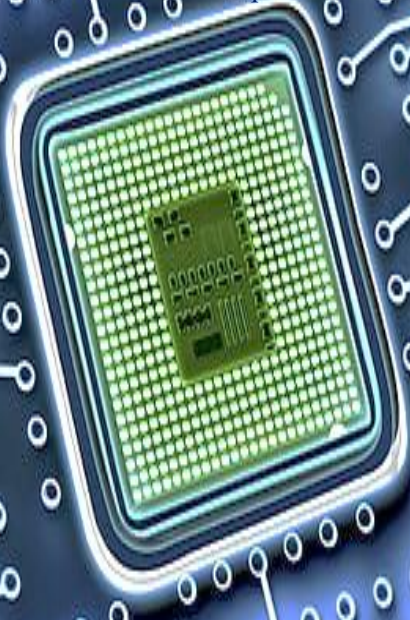


*М.Т. Стринадко
Д.А. Смерека*

**ОБМІН ДАНИМИ МІЖ МІКРОКОНТРОЛЕРАМИ
ПО ТЕЛЕКОМУНІКАЦІЙНОМУ КАНАЛУ.
ОБРОБКА ТА ВІЗУАЛІЗАЦІЯ ДАНИХ
МІКРОКОНТРОЛЕРОМ.**

Методичні рекомендації до курсової роботи



Чернівці - 2022

Міністерство освіти і науки України
Чернівецький Національний Університет
Імені Юрія Федьковича

**ОБМІН ДАНИМИ МІЖ МІКРОКОНТРОЛЕРАМИ
ПО ТЕЛЕКОМУНІКАЦІЙНОМУ КАНАЛУ.
ОБРОБКА ТА ВІЗУАЛІЗАЦІЯ ДАНИХ
МІКРОКОНТРОЛЕРОМ.**

Методичні рекомендації до курсової роботи
з навчальної дисципліни
“ Обчислювальна техніка та мікропроцесори”
для студентів спеціальностей
«Телекомунікації і радіотехніка»

Чернівці
ЧНУ ім. Юрія Федьковича 2022

УДК 006.91:378(075)

ББК 31.22я73

К66

*Рекомендовано Вченою Радою
Інституту фізико-технічних та комп'ютерних наук
Чернівецького національного університету
імені Юрія Федьковича
(Протокол No 1 від 31.09.2022)*

Укладачі: Стринадко Мирослав Танасійович, канд. фіз.-мат. наук, доц.,
Смерека Денис Анатолійович.

К66 Стринадко М.Т. Смерека Д.А. Методичні рекомендації до курсових робіт з навчальної дисципліни “Обчислювальна техніка та мікропроцесори”. Чернівці : Чернівецький національний Університет імені Юрія Федьковича, 2022. 82 с.

У методичній розробці наведені вказівки до курсових робіт з навчальної дисципліни “Обчислювальна техніка та мікропроцесори” для студентів всіх форм навчання спеціальності «Телекомунікації і радіотехніка».

УДК 006.91:378(075)

© Чернівецький національний університет
імені Юрія Федьковича, 2022

ЗМІСТ

ВСТУП	6
Вимоги до курсової роботи.....	7
Критерії оцінювання якості курсової роботи.....	9
Завдання до курсової роботи.....	12
Варіанти завдань курсової роботи	14
Приклад оформлення курсової роботи.....	15
Оформлення списку використаної літератури.....	74

ВСТУП

Дані методичні вказівки мають на меті надати студентам основні рекомендації та показати алгоритм дій щодо виконання курсової роботи з навчальної дисципліни «Обчислювальна техніка та мікропроцесори».

Основна увага приділяється розвитку у студентів навичок самостійно використовувати матеріал лекційного та лабораторного курсу для вирішення конкретних практичних задач. Продемонструвати вміння створювати у відповідності до поставленого завдання схемотехнічні рішення. Продемонструвати вміння роботи з програмними продуктами MPLAB та PROTEUS. Вміння створити діючий проект та провести його апробацію та демонстрацію роботи в програмному продукті PROTEUS.

Методичні вказівки містять перелік завдань і приклади для їх виконання, які можна використовувати при виконанні курсової роботи.

Вимоги до курсової роботи

Матеріали курсової роботи повинні включати:

- Надрукований та підшитий варіант курсової роботи, оформлений згідно правил.
- Електронний варіант у форматі pdf.
- Повний архів файлів проекту MPLAB та PROTEUS на електронному носії.
- Презентацію виступу.

Курсова робота повинна бути надрукованою на принтері з однієї сторони аркуша.

Друковану роботу оформлюють на аркушах білого паперу формату А4 (21 × 29,7 см). Розміри полів: зверху – 2 см, знизу – 2,5 см, зліва – 3 см, справа – 1,5 см. Шрифт – Times New Roman або Arial, розмір – 12-14 пунктів. Міжрядковий інтервал – 1,5.

Нумерація сторінок здійснюється на колонтитулах, номер сторінки проставляється у нижньому (або верхньому) правому куті або по центру сторінки арабськими цифрами. Кривка після цифри не ставиться. Титульний аркуш не нумерується. Кожна частина роботи (розділ) починається з нової сторінки.

Рекомендований обсяг курсової роботи – 20-25 сторінок без додатків. Цей обсяг може відхилитися в межах $\pm 10\%$. Такі ж вимоги до оформлення електронної версії курсової роботи.

Рекомендована структура курсової роботи:

- титульний аркуш;
- завдання до роботи (за загальною формою);

- реферат;
- зміст;
- список умовних позначень, термінів, скорочень
- вступ;
- основна частина;
- висновки;
- список літератури;
- додатки.

Критерії оцінювання якості курсової роботи

Відповідно до завдання, студент виконує всі розділи курсової роботи згідно змісту. Захищається курсова робота при комісії із 3 викладачів.

Остаточна оцінка за курсову роботу виставляється за результатами її захисту в присутності комісії. Захист роботи оцінюється за шкалою ECTS та національною шкалою.

При виставленні остаточної оцінки за курсову роботу члени комісії зі захисту курсової роботи обов'язково враховують такі моменти:

- Робота вважається виконаною за умови представлення діючих проектів MPLAB та PROTEUS. Використовувати електронні компоненти, інтерфейси, мікроконтролери, датчики, дисплеї, тощо виключно передбачені завданням курсової роботи.
- Відповідність роботи вимогам її написання;
- Оцінку керівника курсової роботи;
- Захист роботи її автором перед членами комісії.

Оцінка ECTS	Критерії оцінок	Оцінка в балах	Оцінка н.шкала
А	<p>Студент надав комісії оформлений згідно вимог текст курсової роботи, показав вільне і глибоке володіння матеріалом курсової роботи. Надав архів всіх файлів діючих проєктів MPLAB та PROTEUS. Вільно орієнтується у програмному коді. Продемонстрував роботу в PROTEUS.</p> <p>Точно і повно відповів на всі запитання членів комісії. Вільно володіє термінологією.</p>	90 - 100	Відмінно
В	<p>Студент надав комісії оформлений згідно вимог текст курсової роботи, показав вільне і глибоке володіння матеріалом курсової роботи. Надав архів всіх файлів діючих проєктів MPLAB та PROTEUS. Вільно орієнтується у програмному коді. Продемонстрував роботу в PROTEUS.</p> <p>При відповіді на запитання студентом були допущені незначні неточності, які він зумів повністю виправити після того, як на них було звернуто увагу з боку членів комісії. В основному володіє термінологією.</p>	89 - 90	Добре
С	<p>Студент надав комісії оформлений згідно вимог текст курсової роботи, показав вільне володіння матеріалом курсової роботи. Надав архів всіх файлів діючих проєктів MPLAB та PROTEUS. Вільно орієнтується у програмному коді. Продемонстрував роботу в PROTEUS.</p> <p>При відповіді на запитання студентом були допущені незначні неточності, які він не зумів повністю виправити після того, як на них було звернуто увагу з боку членів комісії. В основному володіє термінологією.</p>	70 - 79	Добре

D	Студент надав комісії оформлений згідно вимог текст курсової роботи, показав, що він в основному володіє змістом курсової роботи. Надав архів всіх файлів діючих проєктів MPLAB та PROTEUS. Вільно орієнтується у програмному коді. Продемонстрував роботу в PROTEUS. Відповіді на запитання членів комісії були не зовсім чітко сформульовані. Деякі терміни студент використовує не за їх призначенням.	60 -69	Задовільно
E	Студент надав комісії оформлений згідно вимог текст курсової роботи, показав, що він в основному володіє змістом курсової роботи. Надав архів всіх файлів діючих проєктів MPLAB та PROTEUS. Орієнтується у програмному коді. Продемонстрував роботу в PROTEUS. Відповіді на запитання членів комісії були не чітко сформульовані. Терміни студент використовує не за їх призначенням.	50 -59	Задовільно
FX	Студент не надав комісії оформлений згідно вимог текст курсової роботи. Відповіді на запитання членів комісії є поверхневими. Знання термінів незадовільне.	35 - 49	Незадовільно
F	Студент не надав комісії оформлений згідно вимог текст курсової роботи. Студент показав, що він не володіє змісту курсової роботи. Відповіді на запитання членів комісії є неправильними. Знання термінів незадовільне.	1 -34	Незадовільно

Завдання курсової роботи

ЗАВДАННЯ

до курсової роботи з курсу

«Обчислювальна техніка та мікропроцесори».

ПІБ студента: _____

Тема курсової роботи: Обмін даними між мікроконтролерами по телекомунікаційному каналу. Обробка та візуалізація даних мікроконтролером.

Варіант № _____

Завдання: Організувати обмін даними між двома мікроконтролерами PIC16F628 та PIC16F877 по телекомунікаційному каналу з відповідним завданням інтерфейсом. Організувати прийом даних та їх візуалізацію мікроконтролерами з відповідних завданню пристроями вводу (датчики, клавіатура, тощо) та виводу (дисплеї цифрові, дисплеї графічні, тощо). Узагальнена блок схема системи на базі мікроконтролерів на рис.1.

Компоненти схеми, а саме: мікроконтролер, датчики, дисплей, пристрої вводу та виводу, вибирають згідно варіанту завдання.

Обладнання: Емуляція в програмних продуктах.

Програмні продукти: MPLab X IDE. Proteus 8 Professional.

Результат курсової роботи:

1. Асемблерний код програми організації відповідного завданню інтерфейсу зв'язку між мікроконтролерами PIC16F628 та PIC16F877. (Діючий MP Lab проект для PIC16F628 та діючий MP Lab проект для PIC16F877).
2. Асемблерний код програми організації відповідного завданню інтерфейсу зв'язку між мікроконтролерами PIC16F628, PIC16F877 та пристроями вводу і виводу. (Окремо діючий MP Lab проект для кожного інтерфесу, а саме: пристрій вводу - PIC16F628, пристрій вводу - PIC16F877, пристрій виводу - PIC16F628, пристрій виводу - PIC16F877).
3. Діючий MP Lab проект та Proteus проект згідно блок схеми рис. 1.
4. Скріншоти (не менше трьох) різного робочого стану MP Lab проекту та Proteus проекту.
5. Оформлений відповідно вимог текст курсової роботи.



Рис. 1. Узагальнена блок схема системи на базі мікроконтролерів PIC16F628 та PIC16F877.

Дата отримання завдання: 05.09.2022 року.

Дата захисту курсової роботи: 25.11.2022 року.

_____ Підпис студента, дата: 05.09.2022 року.

Варіанти завдань курсової роботи 2022 року.

№	Завдання на курсову роботу 2022 рік.				
	Інтерфейс зв'язку PIC16F628 та PIC16F877	Пристрій вводу PIC16F628	Пристрій вводу PIC16F877	Пристрій виводу PIC16F628	Пристрій виводу PIC16F877
1	RS232	DS18B20	DS1822	LM016L	-
2	RS232	DS18B20	DHT11	-	PG12864F
3	RS232	DS1307	MRF-TA1	LM016L	MILFORD 2x20-BKP
4	RS232	DS1307	«Calculator Keypad»	LM016L	MILFORD 4x20-BKP
5	RS232	«Calculator Keypad»	MRF-TA1	LM016L	MILFORD 2x20-BKP
6	RS232	DS1822 DS1822	DS1822 DS1822	-	MILFORD 4x20-BKP
7	I2C	DS18B20	DS1820	LM016L_1	LM016L_2
8	I2C	DS18B20	DHT22	LM016L_1	LM016L_2
9	I2C	DS1307	MRF-TA1	LM016L	MILFORD 2x20-BKP
10	I2C	DS1307	«Calculator Keypad»	LM016L	MILFORD 4x20-BKP
11	I2C	«Calculator Keypad»	MRF-TA1	LM016L	MILFORD 4x20-BKP
12	I2C	DS1822 DS1822	DS1822 DS1822	-	MILFORD 4x20-BKP
13	SPI	DS18B20	DS1822	LM016L_1	LM016L_2
14	SPI	DS18B20	DHT11	LM016L_1	LM016L_2
15	SPI	DS1307	MRF-TA1	LM016L	MILFORD 4x20-BKP
16	SPI	DS1307	«Calculator Keypad»	LM016L	MILFORD 4x20-BKP
17	SPI	«Calculator Keypad»	MRF-TA1	LM016L	MILFORD 4x20-BKP
18	SPI	DS1822 DS1822	DS1822 DS1822	-	MILFORD 4x20-BKP

Приклад оформлення курсової роботи.

Рекомендована структура курсової роботи:

- титульний аркуш;
- завдання до роботи (за загальною формою);
- реферат;
- зміст;
- список умовних позначень, термінів, скорочень
- вступ;
- основна частина;
- висновки;
- список літератури;
- додатки (обов'язково).

Нижче наведений приклад оформленої курсової роботи.

Міністерство освіти і науки України
Чернівецький національний університет
імені Юрія Федьковича

Інститут фізико-технічних та комп'ютерних наук

(повна назва інституту)

кафедра кореляційної оптики

(повна назва кафедри)

Курсова робота
з курсу “Обчислювальна техніка та мікропроцесори”

на тему: **Обмін даними між мікроконтролерами по телекомунікаційному каналу.**

Обробка та візуалізація даних мікроконтролером.

Студента (ки) : 3 курсу 324 групи

Спеціальності : 172

“Телекомунікації та радіотехніка”

(прізвище та ініціали)

Керівник: доцент, кандидат фіз.-мат.наук

Стринадко М.Т.

(посада, вчене звання, ступінь, прізвище та ініціали)

Національна шкала : _____

Кількість балів: _____ Оцінка: ECTS _____

Стринадко М.Т.

(підпис керівника)

(прізвище та ініціали)

Члени комісії

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

Чернівці–2022

РЕФЕРАТ

В курсовій роботі розроблений асемлерний код в програмі MPLAB X, який реалізує зв'язок між мікроконтролерами по інтерфейсу SPI, обмін даними між годинником DS1307 і мікроконтролером PIC16F628 по інтерфейсу I2C, а також вивід даних, в залежності від натискання кнопки, на дисплей LM016L або Milford.

Було перевірено та змодельовано роботу схеми в програмі Proteus 8.

В додатках приведений короткий огляд мікроелектронних пристроїв, використаних в даній курсовій роботі. Описана структура портів мікроконтролерів PIC16F628, PIC16F877, описані використані інтерфейси, а саме: SPI, I2C, USART, а також опис годинника DS1307.

Сторінок - **44**, рисунків - 16, джерел літератури - **5**.

Ключові слова: мікроконтролер PIC16F628, мікроконтролер PIC16F877, годинник DS1307.

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ, СКОРОЧЕНЬ

PIC16F628	- тип мікроконтролера
PIC16F877	- тип мікроконтролера
LCD	- дисплей
DS1307	- годинник
МК	- мікроконтролер
ЗП	- запам'ятовуючий пристрій

ЗМІСТ

РЕФЕРАТ	
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВЬ	
ВСТУП.....	
1. Блок-схема системи.....	
1.1. Взаємодія компонентів блок-схеми системи	
2. Розробка асемблерного програмного коду мікроконтролерів в MPLAB	
2.1. Розробка асемблерного коду для PIC16F628	
2.1.1 Програмна реалізація інтерфейсу SPI.....	
2.1.2 Програмна реалізація інтерфейсу I2C	
2.1.3 Запис даних на дисплей LM016L	
2.1.4 Повний асемблерний код для PIC16F628.....	
2.2 Розробка асемблерного коду PIC16F877	
2.2.1 Реалізація зв'язку по SPI.....	
2.2.2 Реалізація запису на дисплей Milford по USART	
2.2.3 Повний асемблерний код для PIC16F877	
3. Демонстрація роботи в Proteus.....	
3.1. Обмін даними між DS1307 та PIC16F628	
3.2. Демонстрація обміну даними між мікроконтролерами.	
ВИСНОВКИ.....	
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	
ДОДАТКИ.....	
Основні характеристики й блок-схема мікроконтролера PIC16F628	
Основні характеристики й блок-схема мікроконтролера PIC16F877	
Загальні відомості й характеристики годинника DS1307	
Опис інтерфейсу USART	
Опис інтерфейсу I2C	
Опис інтерфейсу SPI	

ВСТУП

Завдання курсової роботи полягає в тому, щоб організувати обмін даними між двома мікроконтролерами PIC16F628 та PIC16F877 по SPI інтерфейсу. А також організувати прийом даних від годинника DS1307, і їх візуалізацію на дисплеях LM016L і Milford.

Мета курсової роботи - організувати обмін даними між мікроконтролером PIC16F628 і PIC16F877 по SPI інтерфейсу, а також візуалізувати дані з годинника на дисплеях.

Для виконання поставленої мети вирішено наступні задачі:

- Вибрано всі необхідні компоненти для успішної розробки схеми
- Проведений пошук необхідних інформаційних систем
- Розроблено відповідну схему

Також описаний код реалізації для двох мікроконтролерів. Проведено моделювання та перевірка працездатності схеми в програмі PROTEUS.

1. Блок-схема системи.

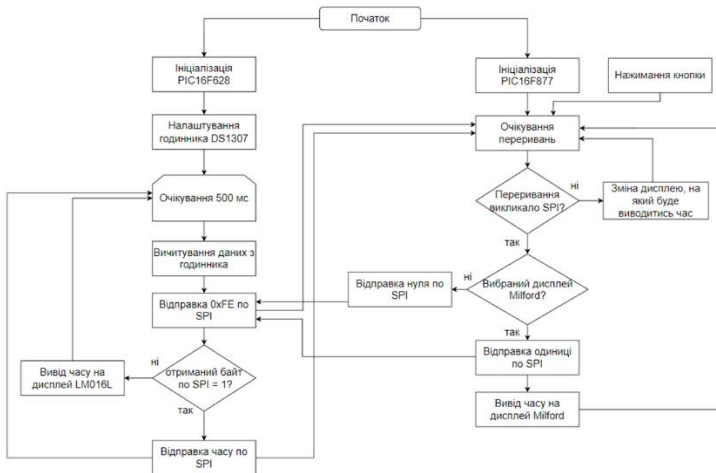


Рис. 1. Блок-схема системи

1.1. Взаємодія компонентів блок-схеми системи

На початку роботи проходить ініціалізація мікроконтролерів:

У PIC16F628 налаштовуються порти для роботи з дисплеєм LM016L, проходить налаштування портів для SPI, а також налаштування годинника DS1307 заданою датою.

У PIC16F877 порти SPI, і вибір режиму Slave, без контакту SS, налаштування UART для роботи з дисплеєм Milford, а також включення переривань від SPI і підключеної кнопки на виводі RB0.

Розглянемо наступні дії мікроконтролера PIC16F628 після ініціалізації. Мікроконтролер очікує деякий час, а далі вичитує та зберігає дані з годинника DS1307 по шині I2C, після чого відправляє перевірочний байт (0xFE) по шині SPI на мікроконтролер PIC16F877. Якщо у відповідь на відправлений байт PIC16F628 отримує одиницю (це сигналізує, що на мікроконтролері PIC16F877 був вибраний дисплей Milford), то дані, які раніше були зчитані з годинника, відправляються по шині SPI. У випадку, якщо у відповідь було отримано будь-яке інше значення крім одиниці (тобто 0x01), мікроконтролер PIC16F628 виводить раніше вчитані дані на дисплей LM016L по паралельному інтерфейсу, і повертається в початок циклу.

Тепер розглянемо роботу PIC16F877 після ініціалізації.

Контролер входить у вічний цикл, який нічого не виконує, в очікуванні на переривання. Якщо було отримано дані по шині SPI від мікроконтролера PIC16F628, або на виводі RB0 буде логічний нуль (тобто була натиснута кнопка), то мікроконтролер переходить в стан переривання. В цьому стані перевіряється, що саме його викликало. Якщо це було натискання кнопки, то в змінній display, яка була створена при ініціалізації, міняється значення з 0 на 1, або з 1 на 0. Значення 1 в змінній display означає що вибрано дисплей Milford, а значення 0 – те, що вибрано дисплей LM016L. Після цього мікроконтролер повертається в режим очікування.

Якщо це був отриманий байт по шині SPI, то спочатку мікроконтролер перевіряє змінну display. Якщо в ній записано 1, то мікроконтролер відправляє по шині SPI одиницю(0x01), яка буде оброблена PIC16F628, а також виводить раніше отриманий байт на дисплей Milford, далі переходить в режим очікування. Якщо змінна display містить 0 – то мікроконтролер відправляє по шині SPI нуль (0x00) і повертається в цикл очікування.

2. Розробка асемблерного програмного коду мікроконтролерів в MPLAB

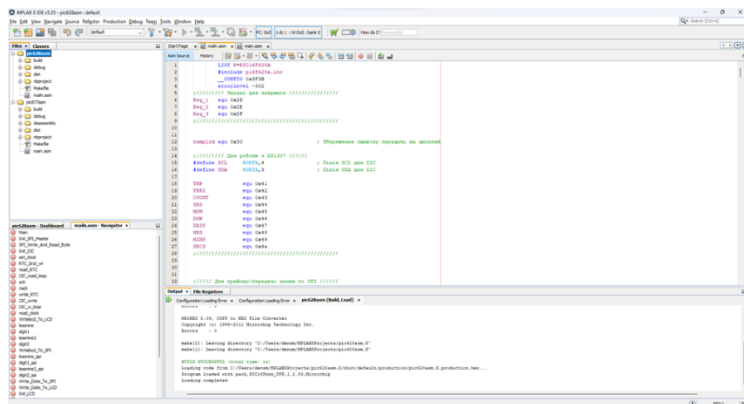


Рис. 2. Головне вікно Mplab

2.1. Розробка асемблерного коду для PIC16F628

2.1.1 Програмна реалізація інтерфейсу SPI

Ініціалізація портів для SPI:

```
Init_SPI_Master:
    bsf STATUS,RP0

    bsf TRISA,RA0           ; Порт RA0 на вхід
    bcf TRISA,RA1         ; Порт RA2 на вихід
    bcf TRISA,RA2         ; Порт RA2 на вихід

    bcf STATUS,RP0
    RETURN
```

Запис і зчитування байту по SPI:

```
SPI_Write_And_Read_Byte:
    bcf SCK
    movwf temp_byte

    movlw .8
    movwf i

new_bitbfsc temp_byte,7
    goto write_1
    goto write_0
write_1 bsf MOSI
    goto next1
write_0 bcf MOSI
    goto next1

next1  bsf SCK
    btfsc MISO
    goto read_1
    goto read_0
```

```

read_1 bsf recieved_byte,0
        goto nxt2
read_0 bcf recieved_byte,0
        goto nxt2

nxt2   call delay_500us
        bcf SCK

        bcf STATUS,C
        rlf temp_byte,1
        bcf STATUS,C
        rlf recieved_byte,0
        decfsz i,F
        goto new_bit
RETURN

```

2.1.2 Програмна реалізація інтерфейсу I2C

Ініціалізація портів для I2C

```

Init_I2C:
    SET_BANK0
    clrf  PORTA                ; Очищення PORTA
    SET_BANK1
    movlw 0x08                ; Налаштування портів для входу
i виходу
    movwf TRISA
    SET_BANK0
    RETURN

```

Запис даних на годинник по інтерфейсу I2C

```

I2C_write:
    SET_BANK1
    bcf  TRISA,3              ; Налаштування SDA для виходу
    SET_BANK0
    movlw 08h                ; Для отримання 8 бітів

```



```

movwf COUNT
bcf   SCL
SET_BANK1
bcf   TRISA, 4           ; SCL = 0, для виходу
SET_BANK0

I2C_w_loop:
bcf   SDA                ; Записати 0
btfsc TMP, 7            ; Якщо біт для відправки = 1:
bsf   SDA                ; то запис 1
SET_BANK1
bsf   TRISA, 4
SET_BANK0
rlf   TMP, 1            ; Зміщення бітів вліво на 1
bcf   SCL
SET_BANK1
bcf   TRISA, 4
SET_BANK0
decfsz COUNT, 1        ; Повтор відправки поки COUNT
!=0
goto  I2C_w_loop

SET_BANK1
bsf   TRISA,3
SET_BANK0
bcf   SCL

SET_BANK1
bcf   TRISA, 4
SET_BANK0
SET_BANK1
bsf   TRISA, 4
SET_BANK0
bcf   SCL
SET_BANK1
bcf   TRISA, 4
SET_BANK0
return

```

Зчитування даних з годинника по шині I2C

```
read_RTC:
    SET_BANK1
    bsf    TRISA,3           ; Налаштування SDA для входу
    SET_BANK0

    movlw 08h               ; Для отримання 8 бітів
    movwf COUNT

    bcf    SCL
    SET_BANK1
    bcf    TRISA, 4         ; SCL = 0, для виходу
    SET_BANK0

    clrf   TMP              ; Очистка регістру
    rlf    TMP, 1
    clrf   TMP

I2C_read_loop:
    rlf    TMP, 1
    SET_BANK1
    bsf    TRISA, 4         ; SCL = 1, для входу
    SET_BANK0

    btfsc  SDA
    bsf    TMP, 0           ; Якщо біт = 1, запис в регістр

    bcf    SCL
    SET_BANK1
    bcf    TRISA, 4         ; SCL = 0, для виходу
    SET_BANK0
    decfsz COUNT, 1        ; Повторення зчитування, поки
COUNT != 0
    goto  I2C_read_loop
    movf   TMP, W           ; Запис зчитаного байта в
акумулятор
    return
```

2.1.3 Запис даних на дисплей LM016L

Вивід символу на дисплей

Write_Char_To_LCD:

BSF PORTB, RB3 ; Встановлення RS = 0

movwf templcd

MOVLW 0x0F ; Відправка старшої частини

ANDWF PORTB, 1

movf templcd, w

andlw 0xf0

IORWF PORTB, 1

CALL Enable ; подача імпульсу на E

MOVLW 0x0F ; Відправка молодшої частини байта

ANDWF PORTB, 1

swapf templcd, w

andlw 0xf0

IORWF PORTB, 1

CALL Enable ; подача імпульсу на E

Enable:

BSF PORTB, RB2 ; E = 1, щоб дисплей обробив

получені дані

call delay_100us

BCF PORTB, RB2 ; E = 0

call delay_100us

RETURN

Вивід числа у форматі BCD на дисплей

Writebcd_To_LCD: ; Конвертування даних з DS1307 з BCD
в ASCII

; і запис на екран LM016L

movwf TMP ; Зберігання даних з акумулятора

swapf TMP, W ; зміна старшого і меншого бітів

місцями

```

        andlw 0x0f          ; Очищення молодших бітів
        addlw 0x06         ; Додавання 6
        btfss STATUS,DC
        goto lessnine      ; Якщо число < 10, перехід до lessnine
        addlw 0x31
        goto digit1       ; Перехід до запису
lessnine:
        addlw 0x2a
digit1:
        call Write_Char_To_LCD ; Запис на дисплей LM016L
        movf TMP,W        ; Відновлення акумулятора
        andlw 0x0f        ; Очищення молодших бітів
        addlw 0x06        ; Додавання 6
        btfss STATUS,DC
        goto lessnine2    ; Якщо число < 10, перехід до lessnine2
        addlw 0x31
        goto digit2
lessnine2:
        addlw 0x2a
digit2:
        call Write_Char_To_LCD ; Запис на дисплей LM016L

        return

```

2.1.4 Повний асемблерний код для PIC16F628

```

LIST P=PIC16F628A
#include p16f628a.inc
__CONFIG 0x3F3B
errorlevel -302
;//////// Змінні для затримок //////////
Reg_1 equ 0x2D
Reg_2 equ 0x2E
Reg_3 equ 0x2F
;////////////////////////////////////

```

```

templcd equ 0x30 ; Збереження символу передачі
на дисплей

;//////// Для роботи з DS1307 //////
#define SCL PORTA,4 ; Лінія SCL для I2C
#define SDA PORTA,3 ; Лінія SDA для I2C

TMP equ 0x41
TMP2 equ 0x42
COUNT equ 0x43
YRS equ 0x44
MON equ 0x45
DOW equ 0x46
DAYS equ 0x47
HRS equ 0x48
MINS equ 0x49
SECS equ 0x4a
;////////////////////////////////////////

;//// Для прийому/передачі даних по SPI ////
temp_byte equ 0x4B ; Зберігання байта для відправки
по SPI
recieved_byte equ 0x4C ; Зберігання прийнятого
байта по SPI
i equ 0x4D ; Для циклу відправки

#define MISO PORTA, RA0 ; Лінія MISO для SPI
#define MOSI PORTA, RA1 ; Лінія MOSI для SPI
#define SCK PORTA, RA2 ; Лінія SCK для SPI
;////////////////////////////////////////

SET_BANK0 MACRO ; Вибір нульового банку PIC
    bcf STATUS, RP0
    bcf STATUS, RP1
ENDM

SET_BANK1 MACRO ; Вибір першого банку PIC

```

```
bsf STATUS, RP0
bcf STATUS, RP1
ENDM
```

I2C_START MACRO ; Макрос для початку роботи по шині I2C

```
bsf SDA
nop
SET_BANK1
bsf TRISA, TRISA4
SET_BANK0
bcf SDA
ENDM
```

I2C_STOP MACRO ; Макрос для завершення роботи по шині I2C

```
bcf SDA
nop
nop
SET_BANK1
bsf TRISA, TRISA4
SET_BANK0
bsf SDA
ENDM
```

```
ORG 0x00
goto Main ; Початок програми з Main
```

Main:

```
call Init_LCD ; Ініціалізація дисплея LM016L
call Init_SPI_Master ; Ініціалізація шини SPI
call set_clock ; Запис дати на DS1307
; (дата відповідає дню
```

написання програми)

```
loop call delay_500ms ; Основний цикл,
очікування 0.5 сек.
```

```

call read_clock ; Зчитування дати з DS1307

movlw 0xFE ; Відправка байта 0xFE по SPI
call SPI_Write_And_Read_Byte
btfsc recieved_byte,0
goto Write_Date_To_SPI
call Write_Date_To_LCD
goto loop

Init_SPI_Master:
bsf STATUS,RP0

bsf TRISA,RA0 ; Порт RA0 на вхід
bcf TRISA,RA1 ; Порт RA2 на вихід
bcf TRISA,RA2 ; Порт RA2 на вихід

bcf STATUS,RP0
RETURN

SPI_Write_And_Read_Byte:
bcf SCK
movwf temp_byte ; Запис з акумулятора в
регістр

movlw .8 ; Для виконання циклу 8
раз
movwf i

new_bitbtfsc temp_byte,7 ; Перевірка біта, який буде
відправлений
goto write_1
goto write_0
write_1 bsf MOSI ; Якщо біт = 1, запис 1 в
MOSI
goto next1
write_0 bcf MOSI ; Якщо біт = 0, запис 0 в
MOSI
goto next1

```

```

nxt1   bsf SCK                                ; Сигналізація підлеглому
що біт відправлений
       btfsc MISO                             ; Перевірка отриманий
біт
       goto read_1
       goto read_0
read_1  bsf recieved_byte,0                  ; Якщо біт = 1, зміна молодшого
біта регістру recieved_byte на 1
       goto nxt2
read_0  bcf recieved_byte,0                  ; Якщо біт = 0, зміна молодшого
біта регістру recieved_byte на 0
       goto nxt2

nxt2   call delay_500us
       bcf SCK                                ; Закінчення передачі
біту

       bcf STATUS,C                          ; Скидання біту C щоб після
наступного зміщення бітів не було проблем
       rlf temp_byte,1                       ; Зміщення бітів регістру вліво
на 1
       bcf STATUS,C
       rlf recieved_byte,0                  ; Зміщення бітів регістру
вліво на 1
       decfsz i,F                            ; Зменшення лічильника
       goto new_bit                          ; Повтор відправки біта, поки
лічильник != 0
       RETURN

Init_I2C:
       SET_BANK0
       clrf  PORTA                          ; Очищення PORTA
       SET_BANK1
       movlw 0x08                            ; Налаштування портів для входу
і виходу
       movwf TRISA
       SET_BANK0

```


RETURN

set_clock:

movlw 0x22
movwf YRS

; Запис дати та часу в регістри

movlw 0x10
movwf MON

movlw 0x14
movwf DAYS

movlw 0x06
movwf DOW

movlw 0x18
movwf HRS

movlw 0x21
movwf MINS

movlw 0x42
movwf SECS

call RTC_brst_wr
DS1307

; Запис дати та часу на

return

RTC_brst_wr:

I2C_START
movlw 0D0h
call write_RTC
movlw 0
call write_RTC
movf SECS, W
call write_RTC
movf MINS, W

; Старт шини I2C

; Відправка команди для запису

; Зміщення адреси для запису секунд

```

call    write_RTC
movf    HRS, W
call    write_RTC
movf    DOW, W
call    write_RTC
movf    DAYS, W
call    write_RTC
movf    MON, W
call    write_RTC
movf    YRS, W
call    write_RTC
I2C_STOP                                ; Завершення роботи по I2C

```

```
return
```

```
read_RTC:
```

```

SET_BANK1
bsf     TRISA,3                        ; Налаштування SDA для входу
SET_BANK0

```

```

movlw  08h                            ; Для отримання 8 бітів
movwf  COUNT

```

```

bcf     SCL
SET_BANK1
bcf     TRISA, 4                       ; SCL = 0, для виходу
SET_BANK0

```

```

clrf   TMP                            ; Очистка регістру
rlf    TMP, 1
clrf   TMP

```

```
I2C_read_loop:
```

```
rlf    TMP, 1
```

```

SET_BANK1
bsf     TRISA, 4                       ; SCL = 1, для входу
SET_BANK0

```

```

    btfsc  SDA
    bsf    TMP, 0                ; Якщо біт = 1, запис в регістр

    bcf    SCL
    SET_BANK1
    bcf    TRISA, 4             ; SCL = 0, для виходу
    SET_BANK0
    decfsz COUNT, 1           ; Повторення зчитування, поки
COUNT != 0
    goto   I2C_read_loop

    movf   TMP, W              ; Запис зчитаного байта в
акумулятор

    return

ack:
    bcf    SDA                ; Відправка команди ACK

    SET_BANK1
    bcf    TRISA,3
    SET_BANK0

    SET_BANK1
    bsf    TRISA, 4
    SET_BANK0
    nop
    bcf    SCL
    SET_BANK1
    bcf    TRISA, 4
    SET_BANK0

    return

nack:
    bsf    SDA                ; Відправка команди NACK

```

```
SET_BANK1
bcf TRISA,3
SET_BANK0
```

```
SET_BANK1
bsf TRISA, 4
SET_BANK0
```

```
bcf SCL
SET_BANK1
bcf TRISA, 4
SET_BANK0
```

```
return
```

```
write_RTC:
```

```
movwf TMP ; Запис з акумулятора в регістр
```

```
I2C_write:
```

```
SET_BANK1
bcf TRISA,3 ; Налаштування SDA для
виходу SET_BANK0
```

```
movlw 08h ; Для отримання 8 бітів
movwf COUNT
```

```
bcf SCL
SET_BANK1
bcf TRISA, 4 ; SCL = 0, для виходу
SET_BANK0
```

```
I2C_w_loop:
```

```
bcf SDA ; Записати 0
btfsc TMP, 7 ; Якщо біт для відправки = 1:
bsf SDA ; то запис 1
; nop
```

```

SET_BANK1
bsf    TRISA, 4
SET_BANK0
rlf    TMP, 1           ; Зміщення бітів вліво на 1
bcf    SCL
SET_BANK1
bcf    TRISA, 4
SET_BANK0
decfsz COUNT, 1       ; Повтор відправки поки COUNT
!=0
goto   I2C_w_loop

SET_BANK1
bsf    TRISA,3
SET_BANK0

bcf    SCL
SET_BANK1
bcf    TRISA, 4
SET_BANK0
; nop
SET_BANK1
bsf    TRISA, 4
SET_BANK0
bcf           SCL
SET_BANK1
bcf    TRISA, 4
SET_BANK0

return

read_clock:           ; Зчитування дати та часу
з DS1307
    I2C_START
    movlw 0D0h       ; Налаштування DS1307 для
запису
    call   write_RTC

```

```

movlw 0 ; Задання адреси секунд
call write_RTC
I2C_START
movlw 0D1h ; Налаштування DS1307 для
зчитування
call write_RTC
call read_RTC ; Вичитка по черзі
секунд, хвилин і тд.
movwf SECS
call ack;
call read_RTC
movwf MINS
call ack;
call read_RTC
movwf HRS
call ack;
call read_RTC
movwf DOW
call ack;
call read_RTC
movwf DAYS
call ack;
call read_RTC
movwf MON
call ack;
call read_RTC
movwf YRS
call nack;
I2C_STOP
return

```

```

Writebcd_To_LCD: ; Конвертування даних з DS1307 з BCD
в ASCII

```

```

; і запис на екран

```

```

LM016L

```

```

movwf TMP ; Зберігання даних з акумулятора
swapf TMP,W ; зміна старшого і

```

```

меншого бітів місцями
    andlw 0x0f ; Очищення молодших
бітів
    addlw 0x06 ; Додавання 6
    btfss STATUS,DC
    goto lessnine ; Якщо число < 10, перехід до
lessnine
    addlw 0x31 ; Перевід в ASCII
    goto digit1 ; Перехід до запису
lessnine:
    addlw 0x2a ; Перевід в ASCII
digit1:
    call Write_Char_To_LCD ; Запис на дисплей LM016L
    movf TMP,W ; Відновлення
акумулятора
    andlw 0x0f ; Очищення молодших
бітів
    addlw 0x06 ; Додавання 6
    btfss STATUS,DC
    goto lessnine2 ; Якщо число < 10,
перехід до lessnine2
    addlw 0x31 ; Перевід в ASCII
    goto digit2 ; Перехід до запису
lessnine2:
    addlw 0x2a ; Перевід в ASCII
digit2:
    call Write_Char_To_LCD ; Запис на дисплей LM016L

return

Writebcd_To_SPI: ; Конвертування даних з DS1307
з BCD в ASCII ; і запис на екран Milford

    movwf TMP ; Зберігання даних з акумулятора
    swapf TMP,W ; зміна старшого і меншого бітів
місцями

```

```

        andlw 0x0f          ; Очищення молодших бітів
        addlw 0x06         ; Додавання 6
        btfss STATUS,DC
        goto lessnine_spi ; Якщо число < 10, перехід до
lessnine
        addlw 0x31         ; Перевід в ASCII
        goto digit1_spi   ; Перехід до запису
lessnine_spi:
        addlw 0x2a         ; Перевід в ASCII
digit1_spi:
        call SPI_Write_And_Read_Byte ; Запис на дисплей
Milford
        movf TMP,W         ; Відновлення акумулятора
        andlw 0x0f         ; Очищення молодших бітів
        addlw 0x06         ; Додавання 6
        btfss STATUS,DC
        goto lessnine2_spi ; Якщо число < 10, перехід до lessnine2
        addlw 0x31         ; Перевід в ASCII
        goto digit2_spi   ; Перехід до запису
lessnine2_spi:
        addlw 0x2a         ; Перевід в ASCII
digit2_spi:
        call SPI_Write_And_Read_Byte ; Запис на дисплей
Milford

        return

Write_Date_To_SPI:
Milford
        movlw 0x80         ; Перевід курсору на початок
        call SPI_Write_And_Read_Byte
        movf HRS,W
        call Writebcd_To_SPI
        movlw ':'
        call SPI_Write_And_Read_Byte
        movf MINS,W
        call Writebcd_To_SPI
        movlw ':'

```



```

call    SPI_Write_And_Read_Byte
movf    SECS,W
call    Writebcd_To_SPI
movlw   0xFE                                ; Перевід дисплею в режим
команд
call    SPI_Write_And_Read_Byte
movlw   0xC0                                ; Перевід курсору на другий
рядок
call    SPI_Write_And_Read_Byte
movf    DAYS,W
call    Writebcd_To_SPI
movlw   '/'
call    SPI_Write_And_Read_Byte
movf    MON,W
call    Writebcd_To_SPI
movlw   '/'
call    SPI_Write_And_Read_Byte
movf    YRS,W
call    Writebcd_To_SPI

goto loop                                    ; Повернення в Main

```

Write_Date_To_LCD: ; Запис дати та часу на дисплей
LM016L

```

movf    HRS,W
call    Writebcd_To_LCD
movlw   ':'
call    Write_Char_To_LCD
movf    MINS,W
call    Writebcd_To_LCD
movlw   ':'
call    Write_Char_To_LCD
movf    SECS,W
call    Writebcd_To_LCD

```

```

movlw 0xC0 ; Перехід на другий рядок
call Write_Command_To_LCD
movf DAYS,W
call Writebcd_To_LCD
movlw '/'
call Write_Char_To_LCD
movf MON,W
call Writebcd_To_LCD
movlw '/'
call Write_Char_To_LCD
movf YRS,W
call Writebcd_To_LCD
movlw 0x80 ; Перехід на перший рядок
call Write_Command_To_LCD

```

```

RETURN

```

```

Init_LCD:

```

```

    BSF STATUS,RP0 ; Очистка регістрів TRISB і
PORTB
    CLRF TRISB
    BCF STATUS,RP0
    CLRF PORTB
    call delay_100ms;
    BCF PORTB,RB3 ; Встановлення RS = 0

    MOVLW 0x0F ; Очистка молодших бітів
регістру PORTB
    ANDWF PORTB,1
    MOVLW 0x30 ; Запис команди в молодші біти
PORTB
    IORWF PORTB,1
    CALL Enable ; Подача імпульсу на E
    call delay_5ms

    MOVLW 0x0F
    ANDWF PORTB,1
    MOVLW 0x30

```

```
IORWF PORTB,1
CALL Enable
call delay_5ms
```

```
MOVLW 0x0F
ANDWF PORTB,1
MOVLW 0x30
IORWF PORTB,1
CALL Enable
call delay_5ms
```

```
MOVLW 0x0F
ANDWF PORTB,1
MOVLW 0x20
IORWF PORTB,1
CALL Enable
call delay_5ms
```

```
movlw 0x28 ; Налаштування функцій
дисплею
call Write_Command_To_LCD
movlw 0x0C ; Включення дисплею,
виключення курсору
call Write_Command_To_LCD
movlw 0x01 ; Очистка дисплею
call Write_Command_To_LCD
movlw 0x06 ; Режим вводу
call Write_Command_To_LCD
RETURN
```

```
Write_Char_To_LCD:
BSF PORTB,RB3 ; Встановлення RS = 0
movwf templcd
MOVLW 0x0F ; Відправка старшої частини
байта
```

```
movf templcd, w
andlw 0xf0
```

```

        IORWF PORTB,1
        CALL Enable                ; Подача імпульсу на E

байта  MOVLW 0x0F                ; Відправка молодшої частини
        ANDWF PORTB,1
        swapf templcd, w
        andlw 0xf0
        IORWF PORTB,1
        CALL Enable                ; Подача імпульсу на E
        RETURN

```

Enable:

```

        BSF PORTB,RB2                ; E = 1, щоб дисплей
обробив отримані дані
        call delay_100us
        BCF PORTB,RB2                ; E = 0
        call delay_100us
        RETURN

```

Write_Command_To_LCD:

```

        BCF PORTB,RB3                ; Встановлення RS = 0
        movwf templcd
        MOVLW 0x0F                ; Відправка старшої частини байта
        ANDWF PORTB,1
        movf templcd, w
        andlw 0xf0
        IORWF PORTB,1
        CALL Enable                ; Подача імпульсу на E

        call delay_5ms

        MOVLW 0x0F                ; Відправка молодшої частини байта
        ANDWF PORTB,1
        swapf templcd, w

```

```
andlw 0xf0
IORWF PORTB,1
CALL Enable ; Подача імпульсу на E
```

```
call delay_5ms
```

```
RETURN
```

```
;/!!!!!!!!!!!!!! Затримки (при частоті PIC = 20 Mhz) !!!!!!!!!!!!!!!
```

```
delay_5ms: ; Затримка 5 мілісекунд
```

```
movlw .118
movwf Reg_1
movlw .33
movwf Reg_2
decfsz Reg_1,F
goto $-1
decfsz Reg_2,F
goto $-3
nop
RETURN
```

```
delay_500ms: ; Затримка 500 мілісекунд
```

```
movlw .182
movwf Reg_1
movlw .175
movwf Reg_2
movlw .13
movwf Reg_3
decfsz Reg_1,F
goto $-1
decfsz Reg_2,F
goto $-3
decfsz Reg_3,F
goto $-5
nop
RETURN
```

delay_100ms:

; Затримка 100 мілісекунд

```
movlw    .85
movwf    Reg_1
movlw    .138
movwf    Reg_2
movlw    .3
movwf    Reg_3
decfsz   Reg_1,F
goto     $-1
decfsz   Reg_2,F
goto     $-3
decfsz   Reg_3,F
goto     $-5
nop
nop
RETURN
```

delay_100us:

; Затримка 100 мікросекунд

```
movlw    .166
movwf    Reg_1
decfsz   Reg_1,F
goto     $-1
nop
RETURN
```

delay_500us:

; Затримка 500 мікросекунд

```
movlw    .61
movwf    Reg_1
movlw    .4
movwf    Reg_2
decfsz   Reg_1,F
goto     $-1
decfsz   Reg_2,F
goto     $-3
nop
nop
```

RETURN

END

2.2 Розробка асемблерного коду PIC16F877

2.2.1 Реалізація зв'язку по SPI

Ініціалізація інтерфейсу SPI

SPI_Slave_Init:

movlw 0x25

movwf SSPCON

bsf STATUS,RP0

movlw 0xD0

; Включення переривань

movwf INTCON

bsf PIE1,SSPIE

; Включення переривань після

отримання даних по SPI

movlw 0x9F

movwf TRISC

; Налаштування виходів TRISC

return

2.2.2 Реалізація запису на дисплей Milford по USART

Ініціалізація інтерфейсу USART

USART_Init:

bsf STATUS,RP0

movlw .129

; по формулі $2 \cdot 10^6 / (16(X+1))$

movwf SPBRG

; Налаштування USART на 9600 бітрейт

movlw 0x24

movwf TXSTA

; Налаштування USART

bcf STATUS,RP0

movlw 0x80

; Налаштування портів запису

USART

movwf RCSTA

return

Запис символу по USART

USART_Write_Char:

```
    bcf STATUS,RP0
wthere btfss PIR1, TXIF      ; Очікування поки дисплей не буде
вільний
    goto wthere
    movf SSPBUF, W           ; Копіювання байта з регістру в
акумулятор
    movwf TXREG             ; Запис байту на дисплей
    movlw 0x01              ; Відправлення 0x01 по SPI
    movwf SSPBUF
    RETFIE
```

2.2.3 Повний асемблерний код для PIC16F877

LIST P=PIC16F877A

#include p16f877a.inc

__CONFIG 0x3F3B

ORG 0x00

GOTO Main ; Старт програми з Main

ORG 0x04

GOTO INTERRUPT

display equ 0x2C ; Регістр для переключення дисплею

SPI_Slave_Init:

movlw 0x25

movwf SSPCON

bsf STATUS,RP0

movlw 0xD0

movwf INTCON

bsf PIE1,SSPIE

; Включення переривань

; Включення переривань по SPI


```

movlw 0x9F
movwf TRISC          ; Налаштування виходів TRISC
return

```

USART_Init:

```

bsf STATUS,RP0
movlw .129
movwf SPBRG          ; Налаштування USART на 9600 бітрейт
movlw 0x24
movwf TXSTA

```

```

bcf STATUS,RP0
movlw 0x80           ; Налаштування портів запису

```

USART

```

movwf RCSTA
return

```

Main:

```

call SPI_Slave_Init ; Ініціалізація SPI
call USART_Init     ; Ініціалізація USART
write goto write

```

INTERRUPT:

```

bcf STATUS,RP0
BTFSC PIR1,SSPIF ; Якщо SSPIF = 1, перехід до spi
goto spi
BTFSC INTCON,INTF; Якщо INCON = 1, перехід до button
goto button
RETFIE

```

```

spi   bcf PIR1, SSPIF          ; Скидання SSPIF
      BTFSC display,0         ; Якщо display = 1, вивід даних
на дисплей
      goto USART_Write_Char
      movlw 0x00              ; В іншому випадку

```

```
відправлення 0x00 по SPI
    movwf SSPBUF
    RETFIE
```

```
USART_Write_Char:
```

```
    bcf STATUS,RP0
wthere  btfss PIR1, TXIF      ; Очікування поки дисплей не буде
вільний
    goto wthere
    movf SSPBUF, W           ; Копіювання байта з регістру в
акумулятор
    movwf TXREG             ; Запис байту на дисплей
    movlw 0x01              ; Відправлення 0x01 по SPI
    movwf SSPBUF
    RETFIE                  ; Повернення в вічний цикл
```

```
button:
```

```
    bcf INTCON, INTF      ; Скидання INTF
    BTFSC display,0       ; Якщо display = 1
    goto minus           ; Зміна на 0
    goto plus            ; Якщо display = 0, зміна на 1
```

```
minus  decf display, F
    RETFIE
```

```
plus   incf display, F
    RETFIE                ; Повернення в вічний цикл
```

```
END
```

3. Демонстрація роботи в Proteus

3.1. Обмін даними між DS1307 та PIC16F628

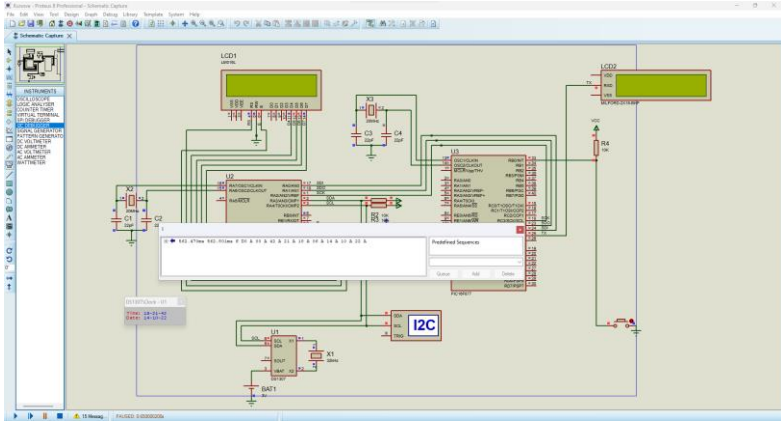


Рис. 3. Запис дати на DS1307

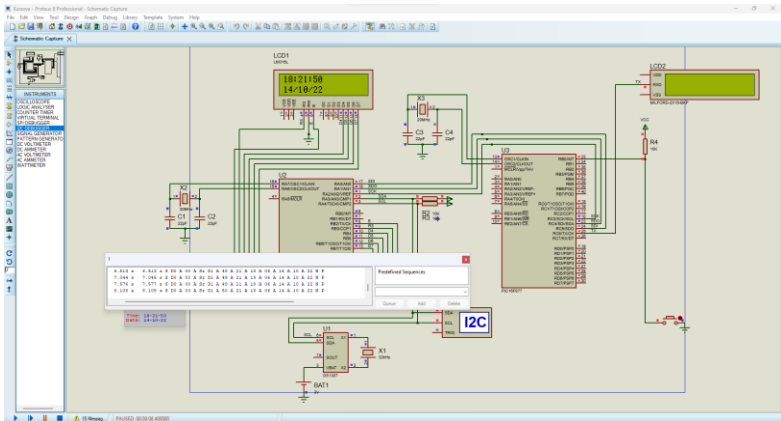


Рис. 4. Прийм даних по I2C від DS1307

3.2. Демонстрація обміну даними між мікроконтролерами.

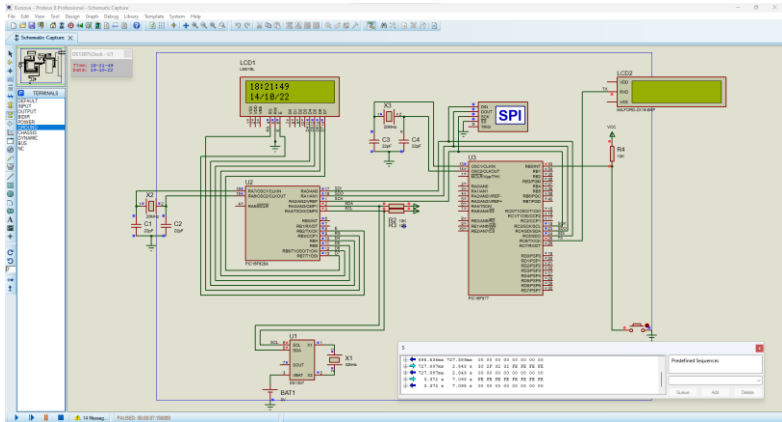


Рис. 5. Обмін даними по SPI, коли вибрано дисплей LM016L

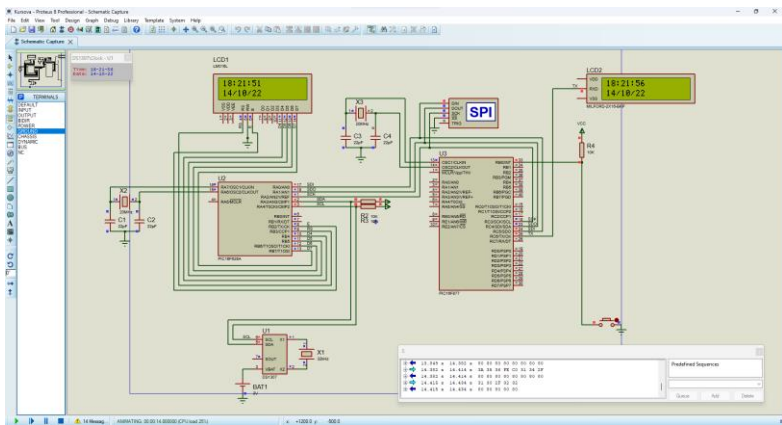


Рис. 6. Обмін даних по SPI, коли вибрано дисплей Milford

ВИСНОВКИ

1. Розроблений асемблерний код в програмі MPLAB X, що реалізує телекомунікаційний канал обміну даними між мікроконтролерами по інтерфейсу SPI.
2. Розроблений асемблерний код в програмі MPLAB X, що реалізує телекомунікаційний канал обміну даними між мікроконтролером та годинником DS1307 по інтерфейсу I2C.
3. Розроблений асемблерний код в програмі MPLAB X, що реалізує вивід даних, в залежності від натискання кнопки, на дисплей LM016L або Milford.
4. Апробована в Proteus 8 схема системи обміну даними між мікроконтролерами по телекомунікаційному каналу та обробки і візуалізації даних на дисплеях LM016L або Milford.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Мікроконтролер PIC16F628:

<https://www.microchip.com/en-us/product/PIC16F628>

2. Мікроконтролер PIC16F877:

<https://www.microchip.com/en-us/product/PIC16F877>

3. Годинник DS1307:

<https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>

4. Дисплей LM016L

<https://embeddedcenter.wordpress.com/ece-study-centre/display-module/lcd-16x2-lm016l>

5. Дисплей Milford 2x16 ВКР

<https://www.farnell.com/datasheets/11357.pdf>

Основні характеристики й блок-схема мікроконтролера PIC16F628

Основні характеристики:

- Тактова частота від DC до 20МГц
- підтримка переривань
- 8-рівневий апаратний стек
- Пряма, непряма і відносна адресація
- 35 однослівних команд

всі команди виконуються за один машинний цикл, крім команд розгалуження і умови з істинним результатом

- Зовнішній і внутрішній режими тактового генератора

Прецизійний внутрішній генератор 4МГц, нестабільність +/- 1%

Енергозберігаючий внутрішній генератор 37кГц

Режим зовнішнього генератора для підключення кварцового або керамічного резонатора

- Режим енергозбереження SLEEP
- Програмовані підтягуючі резистори на входах PORTB
- Сторожовий таймер WDT з окремим генератором
- Режим низьковольтного програмування
- Програмування на платі через послідовний порт (ICSP)
- Захист коду програми
- Скидання по зниженню напруги живлення BOR
- Скидання по включенню живлення POR

- Таймер включення живлення PWRT і таймер запуску генератора OST

Блок-Схема:

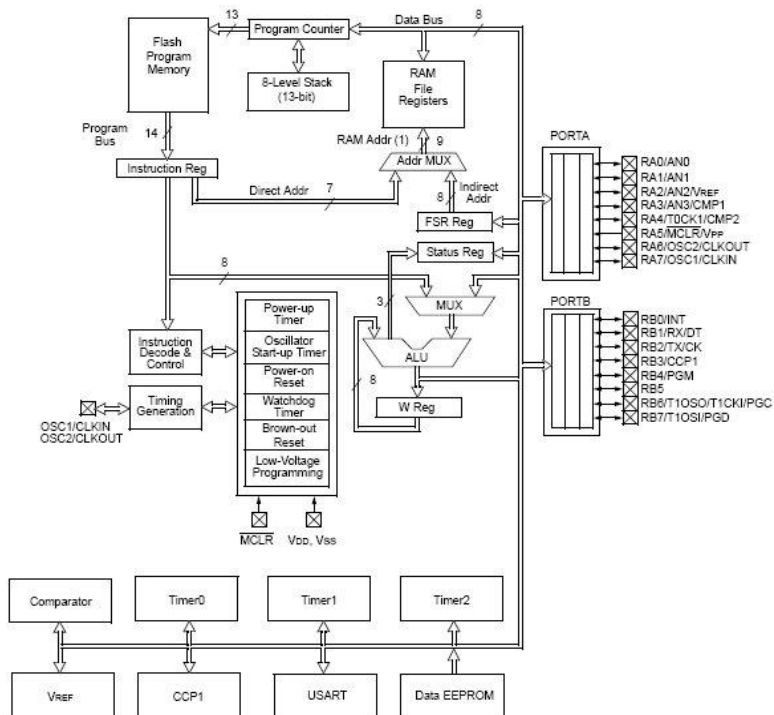


Рис.7. Блок-схема алгоритму роботи МК PIC16F628

Основні характеристики й блок-схема мікроконтролера PIC16F877

Основні характеристики:

- Високошвидкісна RISC архітектура
- 35 інструкцій, всі вони виконуються за один цикл, крім інструкцій переходів, які виконуються за 2 цикли
- Тактова частота від DC до 20 МГц
- Сумісність по виводам з PIC16C73B/74B/76/77
- Система переривань (до 14 джерел)
- 8-ми рівневий апаратний стек
- Прямий, непрямий і відносний режим адресації
- Скидання по зниженню напруги живлення BOR
- Скидання по включенню живлення POR
- Таймер включення живлення PWRT і таймер запуску генератора OST
- Режим енергозбереження SLEEP
- Вибір параметрів тактового генератору
- Повністю статична архітектура
- Програмування в готовому пристрої (використовується два виводи МК)
- Режим внутрішньосхемної відладки (використовується два виводи МК)

- Широкий діапазон напруги живлення: 2 - 5.5 В
- Підвищена навантажуюча здібність портів вводу/виводу (25mA)

Блок-Схема:

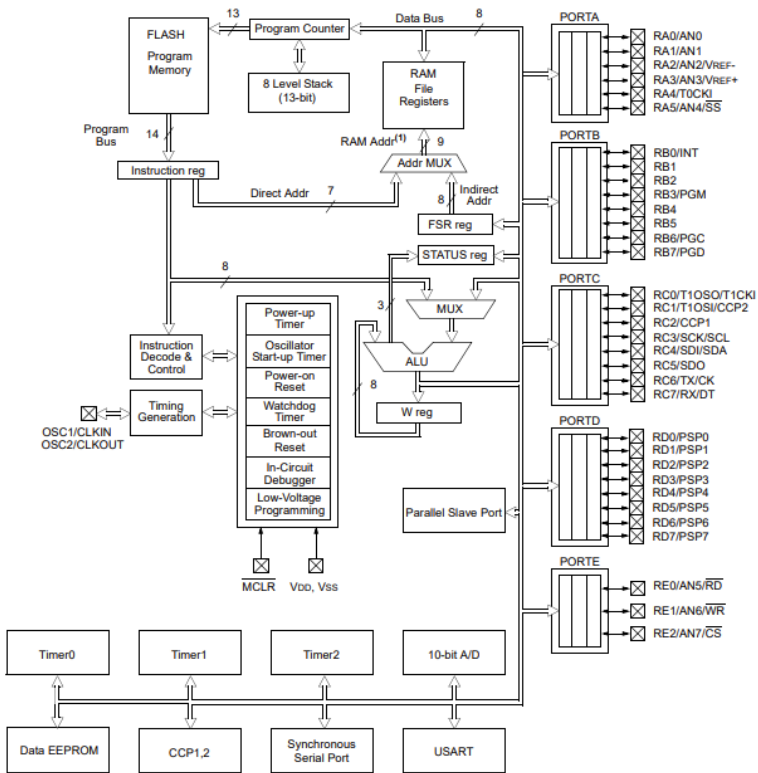


Рис.8. Блок-схема алгоритму роботи МК PIC16F877

Загальні відомості й характеристики годинника DS1307

Годинник реального часу DS1307 може протягом тривалого часу живитись від однієї батарейки типу CR20xx. Даний годинник має 56 байт енергонезалежної пам'яті 24C32, яка доступна для користувача. Він може зберігати та видавати інформацію, таку як день тижня, день, місяць, рік, хвилини, секунди. Формат часу 12 або 24 годин. Корекція календаря автоматична, і буде працювати аж до 2100 року. Час зберігається з точністю до секунди у форматі BCD.

Літієва батарея CR2032 гарантує, що дані зберігатимуться навіть без зовнішнього живлення, та навіть у разі збою. При виявленні збою, мікросхема переходить на резервне джерело, літієву батарею 3В і далі продовжує відраховувати час. Можна запрограмувати мікросхему таким чином, щоб вона працювала в режимі генератора, і генерувала на виході прямокутні імпульси частотою 1, 4096, 8192, або 32768 Гц.

Характеристики DS1307:

- Напруга живлення: 4.5 - 5.5 В
- Струм в режимі очікування: 200 мкА
- Струм споживання: 5 - 50 нА
- Температурний(робочий) діапазон: -40 °С - +85 °С
- Напруга батареї не менше: 2 - 3.5 В

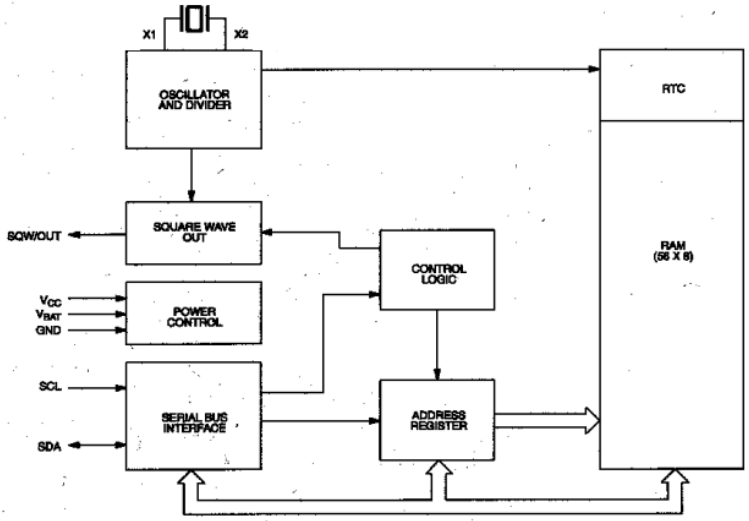


Рис.9. Блок-схема алгоритму роботи годинника DS1307

Опис інтерфейсу USART

USART (Universal Synchronous-Asynchronous Receiver/Transmitter) – інтерфейс, для зв'язку цифрових пристроїв, призначений для передачі даних в послідовній формі. На відміну від UART, підтримує режим синхронної передачі даних з використанням додаткової лінії тактового сигналу. В даній курсовій роботі використовується режим асинхронної передачі, тобто в режимі UART.

Інтерфейси USART/UART, не дивлячись на їх солідний вік, потепер дуже поширені і достатньо затребувані. Вони мають апаратну реалізацію в багатьох мікроконтролерах, включаючи PIC16F628 і PIC16F877, які використовуються в даній курсовій роботі.

В UART передача даних відбувається в послідовній формі, тобто по одному біту. Тому для передачі в одному напрямку потрібен один провідник, а для повнодуплексного двохнапрявленого зв'язку потрібно два провідника.

Вихід позначають TX (transmitted data) , а вхід – RX (received data).

Вигляд підключення двох пристроїв зображений нижче:

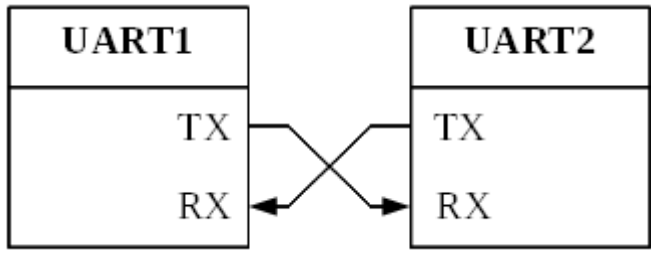


Рис. 10. Підключення двох пристроїв по інтерфейсу UART

Швидкість передачі вимірюється в бодах (відповідає бітам в секунду), в даній курсовій роботі швидкість передачі дорівнює 9600 бод.

Окрім інформаційних бітів, в потоці є синхронізуючі мітки, тобто старт і стоп-біти. При прийомі даних ці біти не враховуються. Дані інформаційні біти займають приблизно 20% потоку, тобто якщо заявлена швидкість передачі 9600 бод, то на частку корисної передачі припадає 7680 бод.

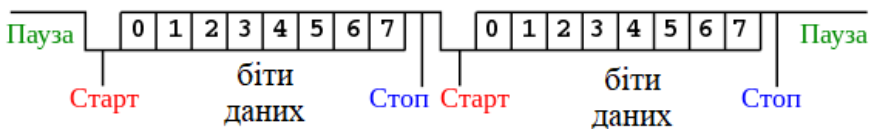


Рис. 12. Структура повідомлення по UART

Опис інтерфейсу I2C

I2C – послідовна асиметрична шина, призначена для зв'язку між інтегральними схемами всередині пристроїв, а також підключення периферійних пристроїв. Використовує дві двонаправлені лінії зв'язку (SDA і SCL)

Серед недоліків даного інтерфейсу є обмежена ємність шини в 400 пФ, важкість програмування і труднощі з виявленням несправностей з ситуаціями низького рівня.

На початку швидкість шини була всього 100 кбіт в секунду, а підключити до неї було можливо до 120 пристроїв. Але в 90-их роках стандарти змінились, і швидкість передачі даних виросла в 4 рази, і появилась можливість підключати до 1000 пристроїв.

На відміну від UART, I2C являється протоколом синхронного зв'язку, тобто обмін даними протікає по спільному, для всіх зв'язаних пристроїв, сигналу синхронізації. Генерацією цього сигналу займається тільки єдиний головний пристрій – ведучий. Підлеглі пристрої мають свій унікальний номер(адресу).

Запит до підлеглих починається з падіння рівня на шині даних SDA, що являється стартовим сигналом для підлеглих. Підвищення рівня SDA при високому тактовому сигналі являється для них стоп-командою. А все, що відбувається між цими двома сигналами, називається повідомленням, тобто передачею даних.

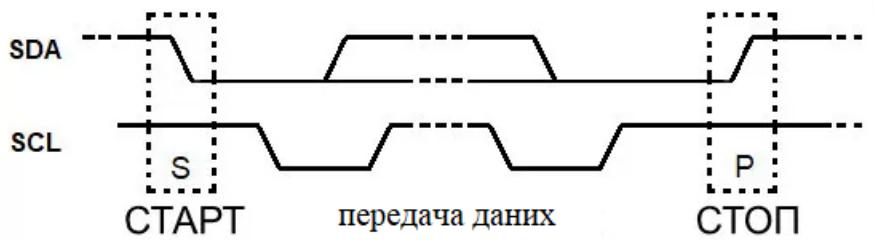


Рис. 13. Вигляд повідомлення I2C

Опис інтерфейсу SPI

SPI (Serial Peripheral Bus) – послідовний синхронний стандарт передачі даних в режимі повного дуплекса. Призначений для забезпечення простого і недорогого високошвидкісного з'єднання мікроконтролерів і периферії.

Схеми з'єднань по SPI:

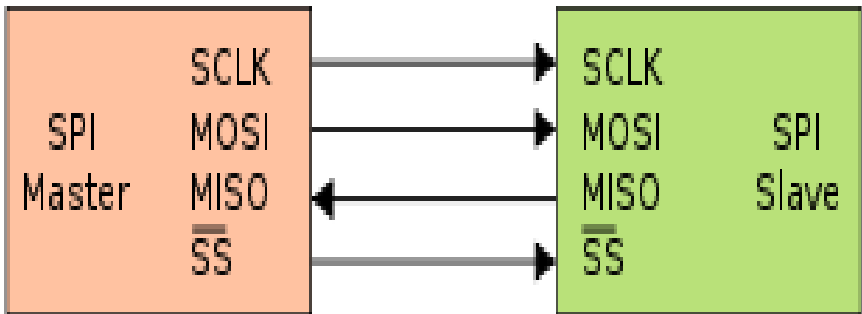


Рис. 14. З'єднання ведучого і підлеглого шиною SPI

В цьому випадку лінію \overline{SS} можна не використовувати, потрібно тільки заземлити \overline{SS} на підлеглому пристрої.

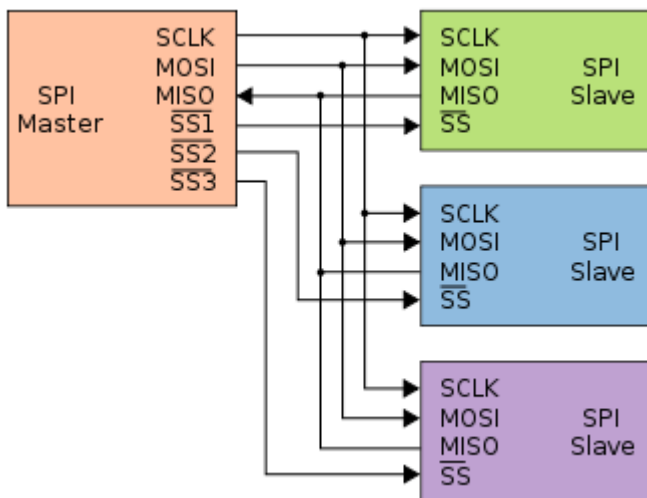


Рис. 15. Радіальне з'єднання декількох підлеглих з ведучи, шиною SPI

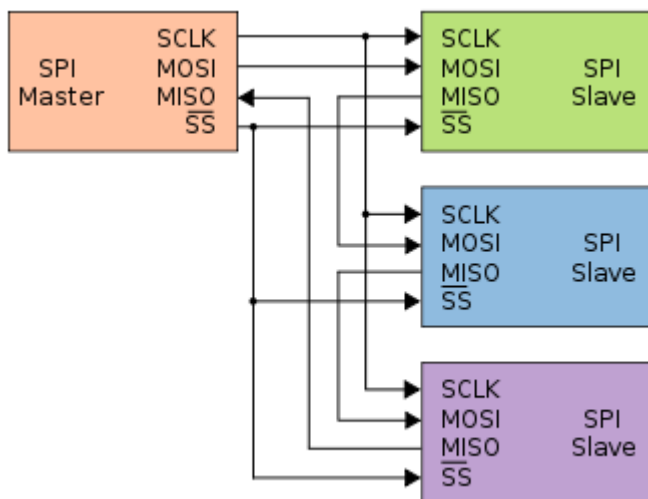


Рис. 16. Кільцеве з'єднання підлеглих з ведучим, шиною SPI

Підключення клавіатури KEYPAD-SMALLCALC до мікроконтролера.

Для розуміння алгоритму роботи та підключення клавіатури KEYPAD-SMALLCALC до мікроконтролера, наприклад PIC16F628, розглянемо наступну схему:

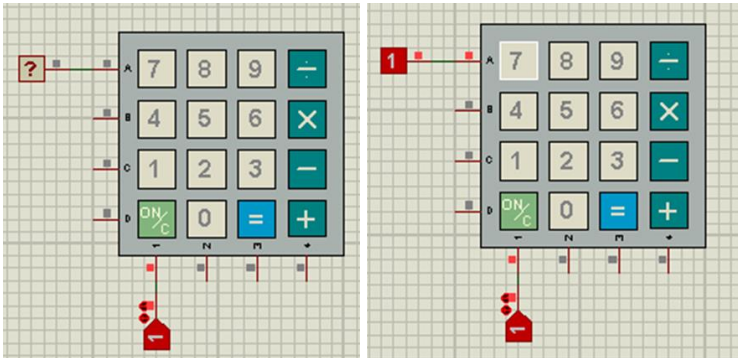


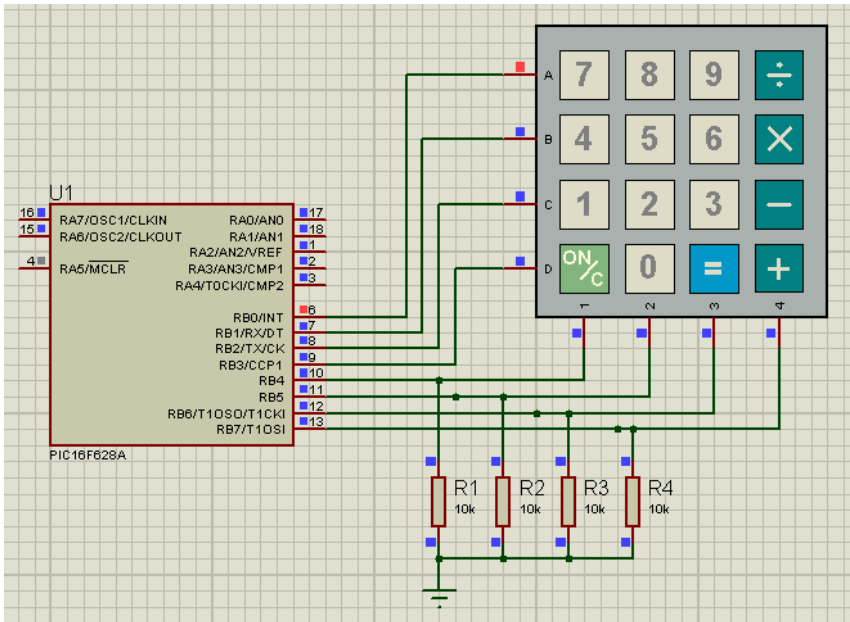
Рис. 1. Клавіатура KEYPAD-SMALLCALC.

а - стан кнопки «7» не натиснуто, б - стан кнопки «7» натиснуто.

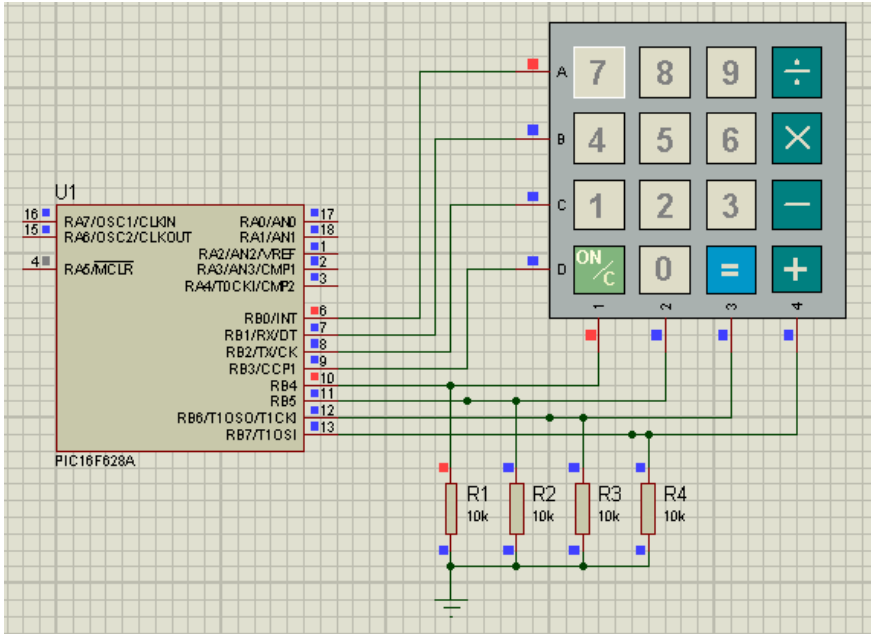
Клавіатура KEYPAD-SMALLCALC – це набір 16 вимикачів. Для організації роботи клавіатури необхідно щоб чотири входи (1,2,3,4) або (a,b,c,d) знаходились в активному стані (логічної одиниці, чи логічного нуля). Інші чотири виводи підключаються до аналізаторів логічного стану.

На рис. 1 приведено приклад роботи клавіатури. Підключений аналіз лише однієї кнопки «7». При натисканні кнопки «7» замикаються відповідна вертикальна 1 та горизонтальна А лінії. Наприклад на рис. 1б кнопк «7» натиснута, відповідно лінія 1 та лінія А замкнуті і сигнал з лінії 1 проходить на лінію А. У випадку відпущеної кнопки лінія А

залишається у невизначеному стані. Це може приводити до хибного спрацювання і отримання недостовірних даних про натискання кнопок клавіатури. Для уникнення такої ситуації рекомендується на відповідні виводи подати або логічну одиницю, або логічний нуль, тобто забезпечити активний сигнал на вході мікроконтролера. На рис. 2 приведена схема підключення клавіатури KEYPAD-SMALLCALC до мікроконтролера PIC16F628. Для підключення використано порт В мікроконтролера PIC16F628. Зручно конфігурувати піни порта В (RB4, RB5, RB6, RB7) як входи, а піни (RB0, RB1, RB2, RB3) як виходи. Таким чином можна організувати опитування стану пінів RB4, RB5, RB6, RB7 програмно, або використати переривання при зміні сигналу на пінах RB4, RB5, RB6, RB7 і перейти на обробку переривання. Для прикладу розглянемо програмне опитування станів пінів RB4, RB5, RB6, RB7.



a)



b)

Рис. 2. Схема підключення KEYPAD-SMALLCALC до PIC16F628.

а - стан кнопки «7» не натиснуто, б - стан кнопки «7» натиснуто.

Таким чином, алгоритм зчитування даних з клавіатури полягає у наступному: циклічно опитуємо стани пінів RB4, RB5, RB6, RB7 і виявляємо натиснуту кнопку по зміні стану пінів RB4, RB5, RB6, RB7. (Використання переривань не потребує організації постійного циклу опитування). Забезпечуємо у циклі опитування всіх 16 кнопок клавіатури. Тобто фрагмент коду програми мікроконтролера виконує наступні дії для вибраного підключення на рис. 2:

1. На виході RB0 встановлюємо логічну одиницю. (RB0=1, RB1=0, RB2=0, RB3=0).

2. Опитуємо стани RB4, RB5, RB6, RB7. (RB4=1 натиснута кнопка 7).
3. На виході RB1 встановлюємо логічну одиницю. (RB0=0, RB1=1, RB2=0, RB3=0).
4. Опитуємо стани RB4, RB5, RB6, RB7. (RB4=1 натиснута кнопка 4).
5. На виході RB2 встановлюємо логічну одиницю. (RB0=0, RB1=0, RB2=1, RB3=0).
6. Опитуємо стани RB4, RB5, RB6, RB7. (RB4=1 натиснута кнопка 1).
7. На виході RB3 встановлюємо логічну одиницю. (RB0=0, RB1=0, RB2=0, RB3=1).
8. Опитуємо стани RB4, RB5, RB6, RB7. (RB4=1 натиснута кнопка On\C).
9. *****
10. На виході RB0 встановлюємо логічну одиницю. (RB0=1, RB1=0, RB2=0, RB3=0).
11. Опитуємо стани RB4, RB5, RB6, RB7. (RB5=1 натиснута кнопка 8).
12. На виході RB1 встановлюємо логічну одиницю. (RB0=0, RB1=1, RB2=0, RB3=0).
13. Опитуємо стани RB4, RB5, RB6, RB7. (RB5=1 натиснута кнопка 5).
14. На виході RB2 встановлюємо логічну одиницю. (RB0=0, RB1=0, RB2=1, RB3=0).

15. Опитуємо стани RB4, RB5, RB6, RB7. (RB5=1 натиснута кнопка 2).
16. На виході RB3 встановлюємо логічну одиницю. (RB0=0, RB1=0, RB2=0, RB3=1).
17. Опитуємо стани RB4, RB5, RB6, RB7. (RB5=1 натиснута кнопка 0).
18. На виході RB0 встановлюємо логічну одиницю. (RB0=1, RB1=0, RB2=0, RB3=0).
19. Опитуємо стани RB4, RB5, RB6, RB7. (RB6=1 натиснута кнопка 9).
20. На виході RB1 встановлюємо логічну одиницю. (RB0=0, RB1=1, RB2=0, RB3=0).
21. Опитуємо стани RB4, RB5, RB6, RB7. (RB6=1 натиснута кнопка 6).
22. На виході RB2 встановлюємо логічну одиницю. (RB0=0, RB1=0, RB2=1, RB3=0).
23. Опитуємо стани RB4, RB5, RB6, RB7. (RB6=1 натиснута кнопка 3).
24. На виході RB3 встановлюємо логічну одиницю. (RB0=0, RB1=0, RB2=0, RB3=1).
25. Опитуємо стани RB4, RB5, RB6, RB7. (RB6=1 натиснута кнопка =).
26. На виході RB0 встановлюємо логічну одиницю. (RB0=1, RB1=0, RB2=0, RB3=0).

27. Опитуємо стани RB4, RB5, RB6, RB7. (RB7=1 натиснута кнопка ÷).
28. На виході RB1 встановлюємо логічну одиницю. (RB0=0, RB1=1, RB2=0, RB3=0).
29. Опитуємо стани RB4, RB5, RB6, RB7. (RB7=1 натиснута кнопка x).
30. На виході RB2 встановлюємо логічну одиницю. (RB0=0, RB1=0, RB2=1, RB3=0).
31. Опитуємо стани RB4, RB5, RB6, RB7. (RB7=1 натиснута кнопка -).
32. На виході RB3 встановлюємо логічну одиницю. (RB0=0, RB1=0, RB2=0, RB3=1).
33. Опитуємо стани RB4, RB5, RB6, RB7. (RB7=1 натиснута кнопка +).
34. Переходимо на пункт 1.

Таким чином отримуємо інформацію про натиснуті кнопки клавіатури.

Оформлення списку використаної літератури

Посилання на літературу можна оформлювати по-різному. Найбільш зручними є номерні посилання. Такі посилання містять тільки номер джерела (за списком), який беруть у прями дужки – [28]. Якщо виникає потреба зробити посилання на певну сторінку, то пишуть – [28, с.45]. Посилання розміщують у тому місці, де це зручно за змістом.

Приклад:

...що збігається з результатами дослідів [4; 21; 28].

...Дослідженнями ряду авторів показано [2; 7; 15], що...

При посиланні на праці, які в списку стоять підряд, їх наводять через тире [6; 24-28; 32]. При наведені в роботі цитат обов'язково посилаються на сторінку – [28, с.45].

При посиланні на багатотомні видання треба наводити том – [28, т.2], або том і сторінку – [28, т.2, с.24].

Бувають випадки, коли прізвище автора включають до структури речення. У такому разі ініціали передують прізвищу. При посиланні ініціали ставлять позаду прізвища. Прізвище автора наводять на мові оригіналу, тобто так, як вказано у списку літератури.

Приклад:

При вивченні даного явища М. М. Марчуком досліджено...

...отримано подібні результати (Марчук М. М., 1999).

R. Bowker (1967) і D. Devis із співробітниками (1978) показа ли, що...

Підрядкові посилання в курсових і кваліфікаційних роботах робити не рекомендується.

Головна вимога до укладання списку використаних джерел – однотипне оформлення та дотримання чинного державного стандарту, що стосується бібліографічного опису друкованих та електронних видань.

Усі літературні джерела, на які є посилання в тексті роботи, повинні бути включені до рубрики СПИСОК ЛІТЕРАТУРИ. Їх можна розміщувати в списку одним із таких способів: у порядку першого посилання в тексті, в алфавітному порядку прізвищ перших авторів, у хронологічному порядку. Найбільш доцільним у курсових і кваліфікаційних роботах є перший спосіб. Нумерація джерел повинна бути наскрізною. 13

Відомості про джерела, які включають до списку, необхідно подавати тільки згідно з вимогами державного стандарту з обов'язковим наведенням назв праць. Зразок оформлення відомостей подано в додатку З. 14

1. Козаков О.М. Оформлення курсових і кваліфікаційних робіт з інженерно-технічних наук : методичні рекомендації. – Чернівці : Чернів. нац. ун-т, 2010. – 72 с.

2. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання : (ГОСТ 7.1–2003, ІДТ) : ДСТУ ГОСТ 7.1:2006. – [Чинний з 2007–07–01]. – К. : Держспоживстандарт України, 2007. – 47 с. – (Національний стандарт України).
3. Інформація та документація. Скорочення слів в українській мові у бібліографічному описі. Загальні вимоги та правила : ДСТУ 3582–97. [Чинний з 1998–07–01]. – К. : Держстандарт України, 1998. – 27 с. – (Національний стандарт України). 01.01.99
4. Калиниченко И.М. Оформление курсовых и дипломных работ : учеб. пособие / И.М.Калиниченко. – М. : Изд. Моск. ун-та, 1982. – 184 с.
5. Бібліографічний опис у списку використаних джерел. Загальні вимоги і правила складання : СТУ 73.1–02070987.02:2009. [Чинний з 2009–12–10]. – Львів. : Львів. нац. ун-т, 2009. – 21 с. – (Стандарт Львівського національного університету імені Івана Франка).

Приклади оформлення бібліографічного опису у списку літератури .

Книги

Один автор

1. Кураев А. Мощные приборы СВЧ. Методы анализа и оптимизации параметров / А. Кураев. – М. : Радио и связь, 1986. – 208 с.
2. Матюх Н. Д. Що дорожче срібла-золота / Наталія Дмитрівна Матюх. – К. : Асамблея діл. кіл: Ін-т соц. іміджмейкінгу, 2006. – 311 с. – (Ювеліри України; т. 1).

Два автори

1. Фалькович С. Е. Основы статистической теории радиотехнических систем : учеб. пособие / С. Е. Фалькович, П. Ю. Костенко. – Харьков : Нац. аэрокосмич. ун-т ХАИ, 2005. – 390 с.
2. Суберляк О. В. Технологія переробки полімерних та композиційних матеріалів : підруч. [для студ. вищ. навч. закл.] / О. В. Суберляк, П. І. Баштанник. – Львів : Растр, 2007. – 375 с.

Три автори

1. Баб'як О. С. Екологічне право України : навч. посібник для вузів / Олексій Степанович Баб'як, Петро Дмитрович Біленчук, Юрій Онисимович Чирва . – К. : Атіка, 2000. - 216 с.
2. Меликов А. З. Математические модели многопоточковых систем обслуживания / А. З. Меликов, Л. А. Пономаренко, П. А. Рюмшин. – К. : Техніка, 1991. – 265 с.

Чотири, п'ять і більше авторів

1. Методика нормування ресурсів для виробництва продукції рослинництва / [Вітвіцький В. В., Кисляченко М. Ф., Лобастов І. В., Нечипорук А. А.]. – К. : НДІ «Украгропромпродуктивність», 2006. – 106 с. – (Бібліотека спеціаліста АПК. Економічні нормативи).
2. Гелле К. Історія Норвегії / Кнут Гелле, Столе Дюрвік, Ральф Даніельсен [та ін.] ; пер. з норв. Н. Іваничук, І. Сабор, М. Красавіна. – Львів : Літопис, 2001. – 351 с.
3. Дифракція волн на решетках / В. П. Шестопапов, Л. Н. Литвиненко, С. А. Масалов, В. Г. Сологуб. – Харків : Изд-во Харьк. ун-та, 1973. – 222 с.
4. Механізація переробної галузі агропромислового комплексу : [підруч. для учнів проф.-техн. навч. закл.] / О. В. Гвоздєв, Ф. Ю. Ялпачик, Ю. П. Рогач, М. М. Сердюк. – К. : Вища освіта, 2006. – 478 с. – (ПТО: Професійно-технічна освіта).
5. Психологія менеджмента / [Власов П. К., Липницький А. В., Луцихина И. М. и др.] ; под ред. Г. С. Никифорова. – [3-є изд.]. – Х. : Гуманитар. центр, 2007. – 510 с.
6. Формування здорового способу життя молоді : навч.-метод. посіб. для працівників соц. служб для сім'ї, дітей та молоді / [Т. В. Бондар, О. Г. Карпенко, Д. М. Дикова-Фаворська та ін.]. – К. : Укр. ін-т соц. дослідж., 2005. – 115 с. – (Серія «Формування здорового способу життя молоді»: у 14 кн., кн. 13).

Без автора

1. Тіло чи особистість? Жіноча тілесність у вибраній малій українській прозі та графіці кінця XIX – початку XX століття : [антологія / упоряд.: Л. Таран, О. Лагутенко]. – К. : Грані-Т, 2007. – 190 с.
2. Проблеми типологічної та квантитативної лексикології : [зб.наук.праць / наук. ред. Каліущенко В. та ін.]. – Чернівці : Рута, 2007. – 310 с.
3. Вопросы подповерхностной радиолокации : коллективная монография / под ред. А. Ю. Гринева. – М. : Радиотехника, 2005. – 416 с.
4. Межфазные взаимодействия и механизмы деградации в структурах металл-InP и металл-GaAs / под общ. ред. Р. В. Конаковой, Г. С. Коротченкова. – Киев : ИО ИФП НАНУ, 1999. – 233 с.

Електронні ресурси

Віддалені ресурси

1. УкрМАРК: Національний формат представлення бібліографічних даних : (Проект) / НБУ ім. Вернадського; НПБ України; Наук. б-ка ім. М. Максимовича; Київ. нац. ун-ту ім. Т.Г. Шевченка. – Електрон. дан. (13 файлів). – 2002-2003. – Режим доступу: www.nbuv.gov.ua/library/ukrmarc.html. – Назва з домашньої сторінки Інтернету.
2. Swanson E. Editing ISBD (SR): approach, scope, definitions // 68th IFLA Council and General Conference, August 18-24, 2002 : Proceedings. – Mode of access: WWW.URL: www.ifla.org/IV/ifla68/papers/148-162e.pdf. – Last access : 2002. – Title from the screen.

3. Прокопенко Л. С. Бібліографічна секція Міжнародної федерації бібліотечних асоціацій та закладів як осередок дослідження національної бібліографії (1965–2002 рр.) : Автореф. дис. ... канд. іст. наук: 07.00.08/ Київ нац. ун-т культури і мистецтв. – Електрон. дан. (1 файл). – К., 2004. – 18 с. – Режим доступу: www.nbuv.gov.ua/ard/2004/04plsdnb.zip. – Назва з екрана. 1. Богомольний Б. Р. Медицина екстремальних ситуацій : навч. посіб. для студ. мед. вузів III–IV рівнів акредитації / Б. Р. Богомольний, В. В. Кононенко, П. М. Чуєв. – 80 Min / 700 MB. — Одеса : Одес. мед. ун-т, 2003. – (Бібліотека студента-медика) – 1 електрон. опт. диск (CD-ROM) ; 12 см. – Систем. вимоги: Pentium ; 32 Mb RAM ; Windows 95, 98, 2000, XP ; MS Word 97-2000. – Назва з контейнера.

2. Розподіл населення найбільш численних національностей за статтю та віком, шлюбним станом, мовними ознаками та рівнем освіти : за даними Всеукр. перепису населення 2001 р. / Держ. ком. статистики України ; ред. О. Г. Осауленко. – К. : CD-вид-во «Інфодиск», 2004. – 1 електрон. опт. диск (CD-ROM) : кольор. ; 12 см. – (Всеукр. перепис населення, 2001). –

Локальні ресурси 20

Систем, вимоги: Pentium-266 ; 32 Mb RAM ; Windows 98/2000/NT/XP. – Назва з титул. екрана.

3. Технологии информационного общества и культура : международные конференции и проекты / Центр ПИК. – Электрон. дан. – М., 2004. – 1 CD-ROM. – Загл. с этикетки диска.

4. Библиотеки и информационные ресурсы в современном мире науки, культуры, образования и бизнеса : материалы междунар. конф. «Крым–2004», г. Судак, 5–13 июня 2004 г. / 11–я междунар. конф. «Библиотеки и ассоциации в меняющемся мире: новые технологии и новые формы сотрудничества», ГПНТБ России, Ассоциация ЭБНИТ. – Электрон. дан. – М.: ГПНТБ России, 2004. – 1 электрон. опт. диск (CD - ROM). – Загл. с этикетки диска.

5. Нежурбеда Г. Г. Роль национальных библиотек в сохранении Internet-ресурсов // Программа ЮНЕСКО : «Информация для всех»: Всеобщий доступ к информации : материалы междунар. конф., г. Санкт–Петербург, 23–25 июня 2004 г. – Электрон. дан. – СПб., 2004. – 1 CD-ROM. – Загл. с этикетки диска.

Примітки:

1) Бібліографічний опис оформлюється згідно з ДСТУ ГОСТ 7.1:2006 «Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання».

2) Проміжки між знаками та елементами опису є обов'язковими і використовуються для розрізнення знаків граматичної і присяної пунктуації.

Навчальне видання

**ОБМІН ДАНИМИ МІЖ МІКРОКОНТРОЛЕРАМИ ПО
ТЕЛЕКОМУНІКАЦІЙНОМУ КАНАЛУ. ОБРОБКА ТА
ВІЗУАЛІЗАЦІЯ ДАНИХ МІКРОКОНТРОЛЕРОМ.**

Методичні рекомендації до курсової роботи
з навчальної дисципліни

“ Обчислювальна техніка та мікропроцесори”

для студентів спеціальностей
«Телекомунікації і радіотехніка»

Укладачі: Стринадко Мирослав Танасійович
Смерека Денис Анатолійович.

Відповідальний редактор: Максимяк Петро Петрович