

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
ІНСТИТУТ ЕЛЕКТРОМЕХАНІКИ, ЕНЕРГОЗБЕРЕЖЕННЯ
І СИСТЕМ КЕРУВАННЯ



МЕТОДИЧНІ ВКАЗІВКИ
ЩОДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«МІКРОПРОЦЕСОРНІ СИСТЕМИ ВИМІРЮВАННЯ ТА ЗАХИСТУ
ЕЛЕКТРОТЕХНІЧНОГО ОБЛАДНАННЯ»
ДЛЯ СТУДЕНТІВ ДЕННОЇ ТА ЗАОЧНОЇ ФОРМ НАВЧАННЯ
ЗІ СПЕЦІАЛЬНОСТІ
141 – «ЕЛЕКТРОЕНЕРГЕТИКА, ЕЛЕКТРОТЕХНІКА ТА
ЕЛЕКТРОМЕХАНІКА»
ЗА ОСВІТНЬО-НАУКОВОЮ ТА ОСВІТНЬО-ПРОФЕСІЙНОЮ
ПРОГРАМАМИ
«ЕЛЕКТРОМЕХАНІЧНЕ ОБЛАДНАННЯ ЕНЕРГОЄМНИХ
ВИРОБНИЦТВ»
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

КРЕМЕНЧУК 2017

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Мікропроцесорні системи вимірювання та захисту електротехнічного обладнання» зі спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка» за освітньо-науковою та освітньо-професійною програмами «Електромеханічне обладнання енергоємних виробництв» освітнього ступеня «Магістр»

Укладач старш. викл. М. Ю. Юхименко

Рецензент к.т.н., доц. Д. Г. Мамчур

Кафедра систем автоматичного керування та електропривода

Затверджено методичною радою Кременчуцького національного університета імені Михайла Остроградського

Протокол «___» від «___» _____ 2017 р.

Голова методичної ради _____ проф. В. В. Костін

ЗМІСТ

Вступ	4
1 Теми та погодинний розклад лабораторних і самостійних робіт	6
2 Перелік лабораторних робіт	8
Лабораторна робота № 1 Дослідження організації й ознайомлення з особливостями роботи в інтегрованому середовищі розроблення MPLAB IDE для мікроконтролерів серії PICmicro	8
Лабораторна робота № 2 Дослідження особливостей розроблення та відлагоджування програмного забезпечення для мікроконтролерів сімейства PICmicro у середовищі MPLAB IDE	14
Лабораторна робота № 3 Дослідження організації відображення інформації у мікропроцесорних системах вимірювання та захисту	20
Лабораторна робота № 4 Дослідження організації оброблення запитів переривань мікроконтролерів сімейства PICmicro	28
Лабораторна робота № 5 Дослідження організації введення інформації з датчиків у мікропроцесорних системах вимірювання та захисту електротехнічного обладнання	33
Лабораторна робота № 6 Дослідження організації роботи з аналогово-цифровим перетворювачем. Дискретизація аналогового сигналу та передача його на ЕОМ	38
Лабораторна робота № 7 Дослідження організації роботи з аналогово-цифровим перетворювачем. Дискретизація аналогового сигналу та передача його на ЕОМ.	40
Лабораторна робота № 8 Дослідження організації роботи з периферійним модулем ССР у режимі ШІМ	42
3 Критерії оцінювання знань студентів	45
Список літератури	46
Додаток А Зразок оформлення титульної сторінки звіту з лабораторної роботи	47

ВСТУП

Пропоновані методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Мікропроцесорні системи вимірювання та захисту електротехнічного обладнання» адресовані студентам освітньо-кваліфікаційного рівня «Магістр», які засвоїли основи організації мікропроцесорних пристроїв, ознайомилися зі схмотехнікою сучасних програмованих мікроконтролерів провідних виробників і призначені для оволодіння практичною методикою самостійної підготовки та розв'язання інженерних задач з використанням засобів проектування для роботи з програмованими мікроконтролерами.

У методичних вказівках основна увага приділена технології розроблення та налагоджування програмних додатків для ОМК на всіх етапах, від постановки задачі до створення об'єктного коду. При цьому розглянуті необхідні інструментальні засоби (асемблер, симулятор, інтегроване середовище). Наведені приклади розроблення робочих програм для мікроконтролерів, які мають за мету покращення засвоєння матеріалу курсу. Основна увага приділена практичній стороні розроблення мікроконтролерних пристроїв і програмного забезпечення.

Метою вивчення дисципліни «Мікропроцесорні системи вимірювання та захисту електротехнічного обладнання» ознайомлення з основними типами архітектури однокристальних мікроЕОМ, складом і характеристиками сучасних сімейств програмувальних мікроконтролерів, загальними питаннями застосування мікропроцесорів і мікроконтролерів у вимірювальних приладах, а також системах керування та захисту електротехнічного обладнання; навчання студентів принципам побудови, функціонування та проектування апаратних і програмних засобів мікропроцесорних систем на базі однокристальних мікроЕОМ, а також способам організації інтерфейсів зв'язку з системами керування верхнього рівня.

У результаті вивчення дисципліни студент повинен

знати:

- сучасний стан і можливості елементної бази мікропроцесорних систем вимірювання, контролю та захисту;
- типову структуру, склад та основні характеристики загальноживаних однокристальних мікроЕОМ та цифрових сигнальних процесорів;
- особливості програмування мікроконтролерів PICmicro;
- принципи апаратної побудови і програмної реалізації сучасних програмованих мікроконтролерів;
- методи використання мікропроцесорної техніки для проектування засобів і систем керування процесами та об'єктами;
- організацію паралельних і послідовних інтерфейсів зв'язку на базі однокристальних мікроЕОМ на рівні можливості побудови на їх основі модулів контролю та керування;
- особливості функціонування та класифікацію цифрових сигнальних процесорів;
- організацію систем захисту електромеханічного обладнання на базі мікропроцесорів;

уміти:

- розробляти структури систем контролю, вимірювання та захисту на базі однокристальних мікроЕОМ;
- проектувати системи збору і обробки інформації з використанням мікроконтролерів і мікроЕОМ;
- розробляти алгоритми і програми типових обчислювальних операцій та операцій керування для програмування мікроконтролерів PICmicro;
- розробляти інтерфейси зв'язку з об'єктами керування на базі мікроконтролерів.

**1 ТЕМИ ТА ПОГОДИННИЙ РОЗКЛАД ЛАБОРАТОРНИХ
І САМОСТІЙНИХ РОБІТ**

№ пор.	Тема	Денна форма		Заочна форма	
		К-сть	К-сть	К-сть	К-сть
		год (лб)	год СРС	год (лб)	год СРС
Модуль 1					
1	Дослідження організації й ознайомлення з особливостями роботи в інтегрованому середовищі розроблення MPLAB IDE для мікроконтролерів серії PICmicro	2	4	0,5	2
2	Дослідження особливостей розроблення та відлагоджування програмного забезпечення для мікроконтролерів сімейства PICmicro у середовищі MPLAB IDE	2	4	0,5	2
Модуль 2					
3	Дослідження організації відображення інформації в мікропроцесорних системах вимірювання та захисту електротехнічного обладнання	2	4	1	4
4	Дослідження організації оброблення запитів переривань мікроконтролерів PICmicro	2	4	1	4
Усього за семестр		8	16	3	12

Модуль 3					
5	Дослідження організації введення інформації з датчиків у мікропроцесорних системах вимірювання та захисту електротехнічного обладнання	2	4	1	2
6	Дослідження організації роботи з аналогово-цифровим перетворювачем. Дискретизація аналогового сигналу та передача його на ЕОМ	4	6	1	2
Модуль 4					
7	Дослідження організації роботи з аналогово-цифровим перетворювачем. Дискретизація аналогового сигналу та передача його на ЕОМ	4	6	1	4
8	Дослідження організації роботи з периферійним модулем ССР у режимі широтно-імпульсного модулятора. Розрахунок заданої шпаруватості імпульсів	4	6	–	–
Усього за семестр		14	22	3	8

2 ПЕРЕЛІК ЛАБОРАТОРНИХ РОБІТ

Лабораторна робота № 1

Тема. Дослідження організації й ознайомлення з особливостями роботи в інтегрованому середовищі розроблення MPLAB IDE для мікроконтролерів серії PICmicro

Мета: отримання початкових навичок практичної роботи з системою MPLAB IDE, ознайомлення з призначенням окремих елементів системи та настроюванням її параметрів.

Короткі теоретичні відомості

Програмний комплекс MPLAB IDE, що розглянутий у методичних вказівках, являє собою інтегроване середовище розроблення програм для мікроконтролерів PICmicro. Система MPLAB IDE містить редактор тексту, симулятор і менеджер проектів, підтримує роботу емуляторів (MPLAB-ICE і PICMASTER) і програматорів (PICSTART plus, PRO MATE) та інших налагоджувальних засобів фірми Microchip, а також сторонніх виробників.

MPLAB IDE складається з декількох модулів, що забезпечують єдине середовище розроблення.

Для створення робочого проекту необхідно завантажити програму MPLAB IDE шляхом вибору відповідного пункту в стартовому меню Windows («Пуск\Програми\Microchip MPLAB IDE\MPLAB IDE»), після чого буде завантажено вікно менеджера проектів MPLAB IDE (рис. 1.1).

Вікно менеджера проектів містить наступні елементи:

1. Головне текстове меню (Меню команд).
2. Панель інструментів (Графічне меню), що містить графічні позначення (піктограми) найчастіше використовуваних функцій пакета.
3. Робоча область проекту, у якій розміщуються відкриті вікна з файлами, діалогами та іншою інформацією (Application Workspace).
4. Рядок стану, що відображає поточний стан і настройки системи.

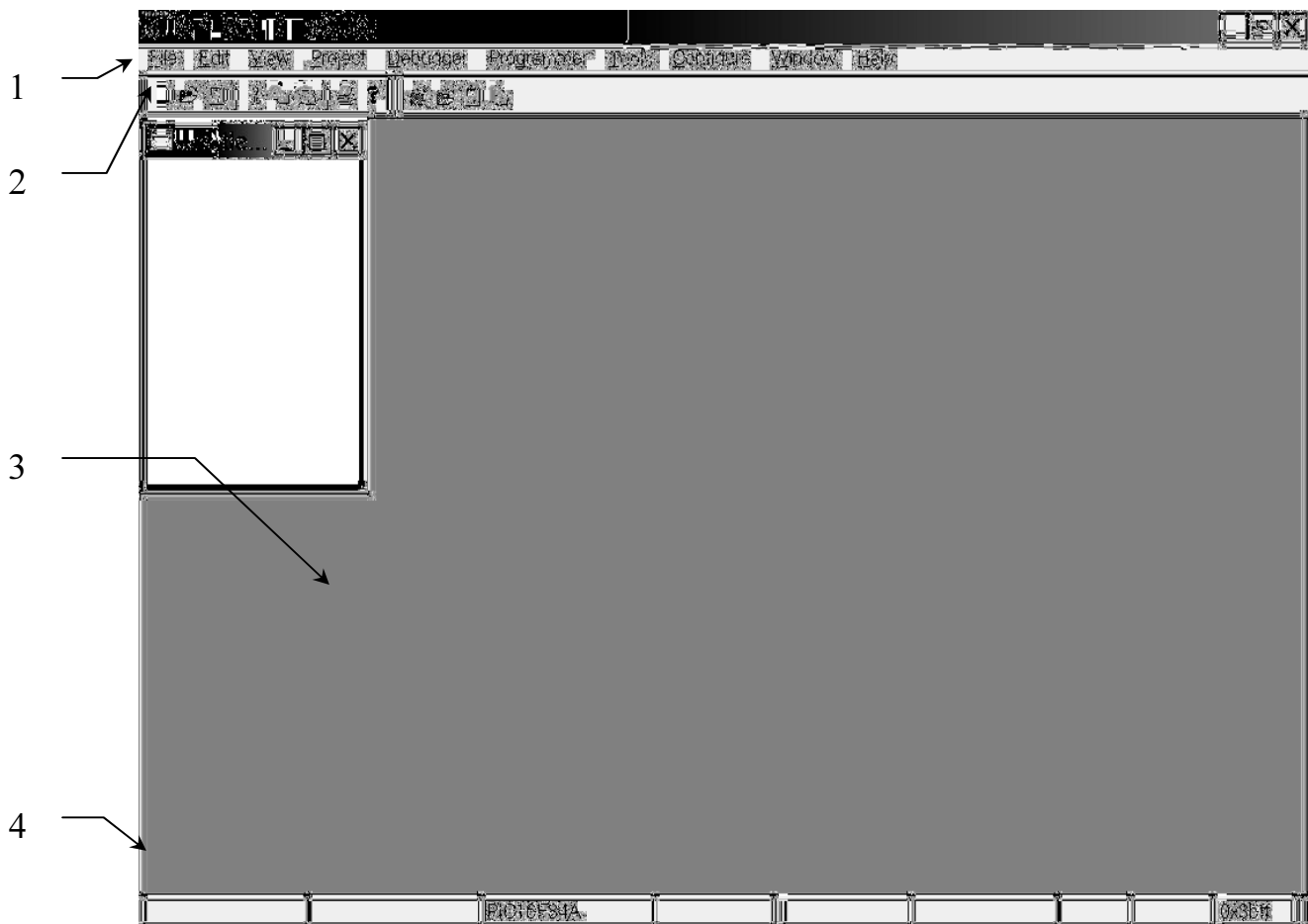


Рисунок 1.1 – Вікно менеджера проектів програмного пакета MPLAB IDE

Команди пакета активізуються за допомогою програми або за допомогою іконок на панелі інструментів.

- File – відкриває меню роботи з файлами;
- Edit – відкриває меню функцій редактора;
- View – відкриває меню налаштування параметрів відображення інформації у вікні менеджера проектів;
- Project – відкриває меню створення та налаштування параметрів проекту;
- Debugger – відкриває меню вибору та опцій налагоджування проекту;
- Programmer – відкриває меню вибору програматора та налаштування параметрів програмування й роботи з програмованим мікроконтролером;
- Configure – відкриває меню вибору пристрою, що програмується, установлення бітів конфігурації;
- Help – відкриває довідкове меню.

Хід роботи

1. Написати програму мовою асемблера для мікроконтролерів, що забезпечить періодичне миготіння світлодіода, приєднаного до лінії RA1.

Для створення вихідного коду програми використовується вбудований редактор MPLAB IDE. Для цього за допомогою команди «File\New...» потрібно створити новий файл (типу «*.asm»).

Рекомендується для кожного нового проекту створювати окремий каталог); ім'я файла має складатися лише з латинських літер і цифр і не містити спецсимволів (+, /, |, \, *, ?, =, >, < та ін.). Не допускається використовувати літери національних алфавітів. Після цього у списку «Тип файла» слід вибрати рядок «Assembly source files (*.asm; *.as, *.inc, *.s)» і натиснути кнопку «Зберегти». Створеному файлу буде автоматично присвоєне розширення *.asm. У разі, якщо змінюється ім'я вихідного файла, потрібно також змінити ім'я проекту.

2. Після успішного створення проекту виконати його компіляцію. Компіляцію вихідного файла може бути виконано за допомогою команди меню «Project > Build All».

3. Після успішної компіляції буде створений файл з об'єктним кодом, який може бути запрограмований у пам'ять програм мікроконтролера. Потрібно перевірити коректність роботи програми, використовуючи симулятор.

Настроювання параметрів середовища проектування

Після успішної компіляції проекту користувач може перевірити його функціонування. Для цього потрібно вибрати інструмент для налагодження. Ці інструменти можуть бути реалізовані як на програмному (симулятори), так і на апаратному (емулятори) рівнях. У розглянутому прикладі використовується симулятор MPLAB SIM. Потрібно вказати необхідний режим роботи в меню «Debugger > Select Tools» і вибрати симулятор MPLAB SIM.

Після цього потрібно вибрати пункт меню «Debugger > Reset > Processor Reset» для ініціалізації системи. Лічильник команд буде встановлений у нуль, що є вектором скидання для мікроконтролера PIC16F84A. У рядку стану в нижній

частині вікна менеджера проектів значення лічильника команд РС буде дорівнювати 0x00.

4. Вибором пункту меню «Debugger > Run» або натисканням на клавішу <F9> виконується запуск програми з поточного місця. Темна смуга, що вказує поточну інструкцію, змінить свій колір, жодний параметр на лінійці стану не буде змінювати свого значення до припинення роботи програми.

Зупинити роботу програми можна командою «Halt» меню «Debugger» або клавішею <F5>, показник поточної інструкції набуде первісного кольору, інформація в рядку стану зміниться.

Приклад програмного коду, що забезпечує періодичне вмикання та вимикання світлодіода.

```
*****
;
; програма миготіння світлодіода, підключеного до виводу RA1 (18)
;*****
list p=16F84                ;директива, що визначає тип процесора та
                            ;інші вихідні параметри проекту
#include <p16F84.inc>        ;директива підключення файла, що описує
                            ;специфічні параметри, які
                            ;відповідають конкретному типу
                            ;мікроконтролера
;*****
; опис використовуваних змінних (призначення адрес
; комірок для зберігання змінних користувача)
;*****
count1 EQU 0x0C            ;змінна циклу
count2 EQU 0x0D            ;змінна циклу
ledset EQU 0x0E            ;значення для виведення через порт
;*****
; *      основний текст програми      *
;*****
```



```

;якщо обидві змінні лічильника = 0 ,
goto loop ;переходимо до мітки loop для виведення
;нового значення в порт і повторення
;процесу.

end ;кінець програми

```

5. Крім того, засобами середовища MPLAB IDE слід занести до вихідного файла наступні настройки слова стану мікроконтролера:

- біт захисту коду після програмування «не встановлений»;
- сторожовий таймер (WDT) «вимкнутий»;
- таймер затримки скидання при подачі живлення «ввімкнений»;
- тип резонатора: «керамічний або кварцовий».

Розглянутий приклад програми працює так. Указавши асемблеру тип процесора та підключивши файл опису апаратних ресурсів P16F84.inc, описують, у яких чарунках ОЗП будуть зберігатися значення змінних. Потім виконують налаштування портів уведення-виведення. Спочатку обнуляють значення у вихідних засувках портів.

Потім, установивши значення «1» біта RP0 регістра STATUS, отримують доступ до регістрового банку 1. Тепер, при звертанні до регістрів PORTA і PORTB, звернення відбувається не до вихідних засувок, а до регістрів стану цих портів, що відповідає схемним підключенням у конкретному випадку. Потім слід установити значення біта RP0 «0», після чого обнулити змінну ledset.

Далі (мітка loop) за допомогою операції «АБО, ЩО ВИКЛЮЧАЄ» інвертують значення біта <1> змінної ledset, зберігаючи значення знову до ledset (xorwf ledset, 0 і, скопіювавши результат в акумулятор (movf ledset,w), виводять значення ledset у вихідну засувку порту A. На лінії RA1 з'являється рівень, що відповідає поточному значенню біта <1> змінної ledset.

Потім виконують ініціалізацію змінної зовнішнього циклу count_1. Усередині цього циклу вкладений ще один цикл, зі змінною count_2. Призначення цих циклів – сформувати затримку, протягом якої на виводі порту

утримується встановлене значення. Коли затримка скінчиться, до порту виводиться інверсне значення і цикл повторюється.

Спробуйте збільшити частоту миготіння світлодіода, зменшуючи початкові значення змінних count_1 і count_2. При цьому слід ураховувати, що якщо встановити частоту миготіння більшу ніж 20 – 25 Гц, то буде здаватися, що світлодіод світиться безперервно, хоча програма працюватиме правильно. Якщо ввести до внутрішнього циклу три÷чотири команди NOP, то затримка помітно збільшиться, відповідно зменшиться частота миготіння.

Зміст звіту

1. Найменування і мета лабораторної роботи.
2. Короткі теоретичні відомості
3. Отримане у викладача завдання.
4. Опис виконання кожного пункту завдання.
5. Письмові відповіді на контрольні питання.

Контрольні питання

1. Дайте загальну характеристику середовища розроблення MPLAB IDE.
2. Назвіть, з яких елементів складається вікно менеджера пакетів MPLAB IDE. Поясніть їх призначення.
3. Опишіть послідовність роботи з майстром проектів MPLAB IDE.
4. Поясніть, як виконати симуляцію роботи програми мікроконтролера. Які програмні або апаратні засоби можна при цьому використовувати?
5. Поясніть, як виконується програмування мікроконтролера за допомогою середовища розроблення MPLAB IDE.

Література: [3, 4, 7].

Лабораторна робота № 2

Тема. Дослідження особливостей розроблення та відлагоджування програмного забезпечення для мікроконтролерів сімейства PICmicro у середовищі MPLAB IDE

Мета: навчитися створювати проекти у середовищі MPLAB IDE, працювати з

проектами, проводити компіляцію та налагоджування програмних додатків.

Короткі теоретичні відомості

Мікроконтролери підгрупи PIC16F8x належать до сімейства 8-розрядних КМОП-мікроконтролерів групи PIC16Cxxx, для яких характерні низька вартість, повністю статична КМОП-технологія й висока продуктивність.

До складу підгрупи входять МК PIC16F83, PIC16CR83, PIC16F84 і PIC16CR84. Основні характеристики МК підгрупи PIC16F8x наведені в таблиці 2.1.

Таблиця 2.1 – Основні технічні характеристики мікроконтролерів PIC16F8x

Параметр	PIC16F83	PIC16CR83	PIC16F84	PIC16CR84
Максимальна частота, МГц	10	10	10	10
Flash-пам'ять програм, слів	512	–	1К	–
ПЗУ програм, слів	–	512	–	1К
Пам'ять даних, байтів	36	36	68	68
Пам'ять даних у РПЗП (EEPROM), байтів	64	64	64	64
Таймери	TMR0	TMR0	TMR0	TMR0
Число джерел переривань	4	4	4	4
Число ліній введення/ виведення	13	13	13	13
Діапазон напруг живлення, В	2,0–6,0	2,0–6,0	2,0–6,0	2,0–6,0
Число виводів і тип корпусу	18 DIP, SOIC	18 DIP, SOIC	18 DIP, SOIC	18 DIP, SOIC

Структурну схему МК підгрупи PIC16F8x показано на рис. 2.1.

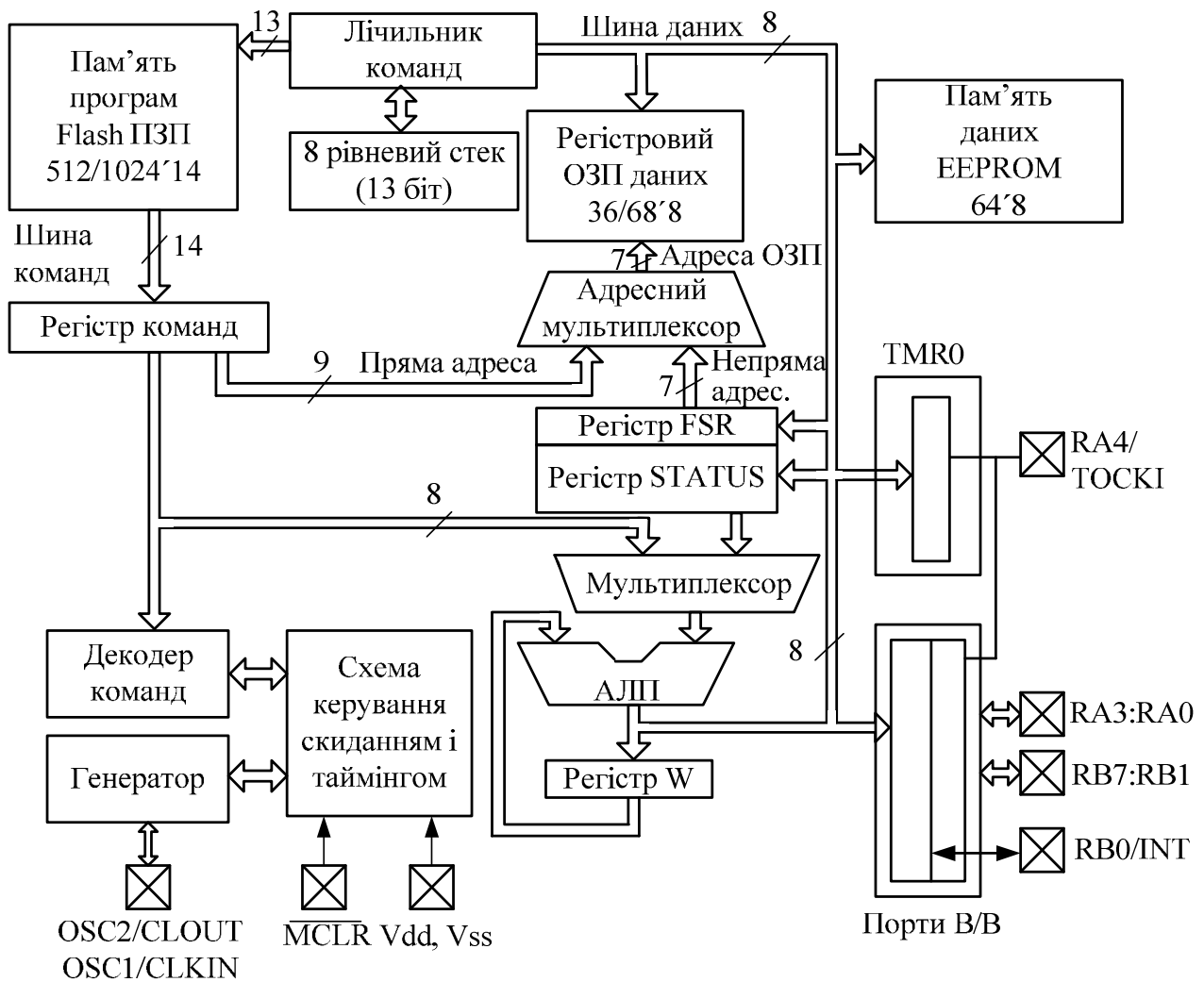


Рисунок 2.1 – Структурна схема МК підгрупи PIC16F8x

Архітектура ґрунтується на концепції роздільних шин і областей пам'яті для даних і для команд (гарвардська архітектура). Шина даних і пам'ять даних (ОЗП) – мають ширину 8 бітів, а програмна шина та програмна пам'ять (ПЗП) мають ширину 14 бітів. Така концепція забезпечує просту, але потужну систему команд, розроблену так, що бітові, байтові й регістрові операції виконуються з високою швидкістю та з перекриттям за часом вибірки команди й циклів виконання. 14-бітна шина пам'яті програм забезпечує вибірку 14-бітової команди за один цикл. Двоступінчастий конвеєр забезпечує одночасну вибірку та виконання команди. Усі команди виконуються за один цикл, виключаючи команди переходів.

Хід роботи

1. Створити новий проект за допомогою пункту меню «Project/New Project» та вказати ім'я проекту, що створюється.

2. У вікні редагування проекту в пункті меню «Debugger» вибрати пункт Select Tools, вибрати режим «MPLAB-SIM». Установити в меню «Configure» тип процесора PIC16F84.

3. У вікні редагування проекту в пункті «Project Files» вибрати пункт з указаним ім'ям проекту та розширенням *.hex. Натиснути кнопку «Node Properties».

4. У новому вікні в пункті «Language Tool» вибрати «MPLINK» і закрити вікно кнопкою «Ok».

5. У вікні редагування проекту натиснути кнопку «Add Node» і ввести ім'я файла аналогічно до імені проекту, але з розширенням *.asm.

6. Повторно натиснути кнопку «Add Node», вибрати розширення «*.lkr» (Linker Script) і знайти файл з ім'ям 16F84.lkr. Цей файл потрібен для правильного складання проекту.

7. На цьому створення проекту завершується. Натисніть кнопку «Ok».

8. За допомогою пункту меню «File/New» створюємо файл з ім'ям, що було вказане у пункті 5.

9. Увести текст програми відповідно до завдання викладача.

Завдання передбачає побудову програми так, щоб у черговому режимі миготів світлодіод, а при натисненні кнопки S2, підключеної до лінії RB2 (як це зображено на рис. 2.2), програма генерувала б звуковий сигнал. Під час генерації імпульсів низької частоти потрібно періодично опитувати стан кнопки, і, якщо вона натиснута, викликати підпрограму генерації звуку. Отже, отримуємо комбінацію програм, з яких одна є головною, а інша викликається як підпрограма.

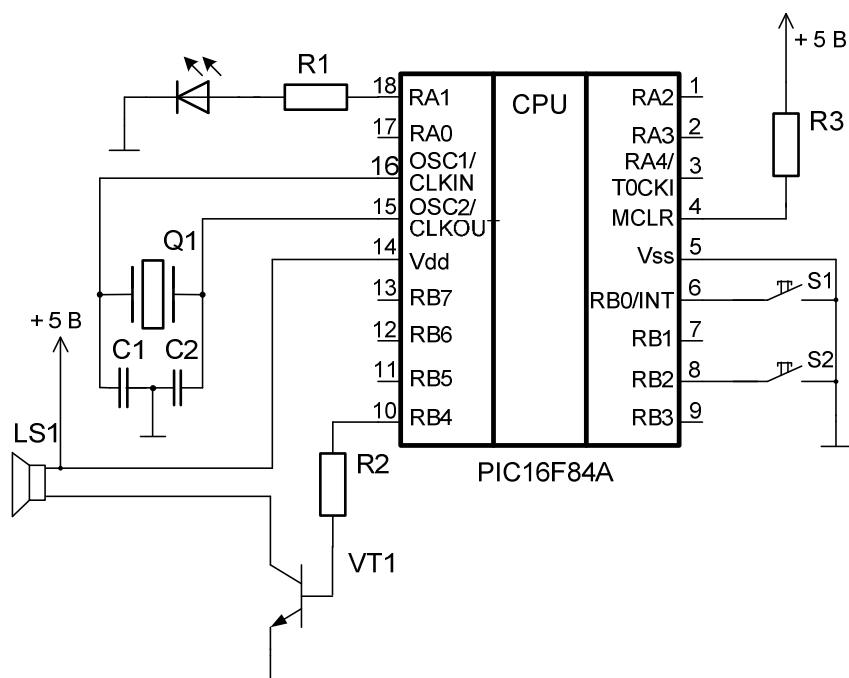


Рисунок 2.2 – Принципова схема макета для відпрацювання програм

```

list p=16F84          ;тип процесора
#include <pi6F84.inc> ;файл опису апаратних засобів МК
    _CONFIG_CP_OFF&_WDT_OFF&_PWRTE_ON &_XT_OSC
count1 EQU 0x0C      ;змінна циклу
count2 EQU 0x0D      ;змінна циклу
count3 EQU 0x0E      ;змінна циклу
ledset EQU 0x0F      ;значення для виведення через порт
    ORG 0x000        ;вектор скидання процесора – початкова
                    ;адреса програми
    clrf PORTA       ;очищення засувки порту А
    clrf PORTB       ;очищення засувки порту В
    bsf STATUS, RP0 ;перемикання на банк 1
    movlw b'1111101' ;настроювання ліній порту А
    movwf PORTA
    movlw b'00000101' ;і порту В
    movwf PORTB
    bcf OPTION_REG,7 ;увімкнути підтягувальні резистори

```

```

    bcf STATUS, RP0      ;повернутися до банку 0
    clrf ledset          ;очистити змінну ledset
loop  movlw b'00000010'  ;інвертування поточного значення біта 1
    xorwf ledset,f      ;змінної ledset
    movf ledset,w       ;і збереження значення в акумуляторі
    movwf PORTA
    movlw 0xFF          ;установлення початкового значення
    movwf count1       ;змінної зовнішнього циклу
cycle1 movlw 0xF0       ;встановлення початкового значення
    movwf count2       ;змінної зовнішнього циклу
cycle2 btfss PORTB,2   ;перевіряємо кнопку S2, якщо натиснута,
                        ;викликаємо підпрограму генерації звуку
    call beep          ;якщо не натиснута –
    decfsz count2,f    ;декремент змінної циклу count2 =
                        ;= count2 - 1
    goto cycle2        ;якщо count2 не дорівнює нулю
    decfsz count1,f
    goto cycle1
    goto loop
beep  movlw 0xA0        ;початок підпрограми генерації звуку
    movwf count3       ;ініціалізували змінну циклу
loop_b movlw b'00010000' ;інвертуємо
    xorwf ledset,f     ;рівень
    movf ledset,w      ;на виводі
    movwf PORTB        ;RB4(10)
    decfsz count3,f    ;декремент count_3, якщо не нуль
    goto loop_b        ;повторити цикл, інакше
    return             ;повернення з підпрограми
end

```

Тепер, якщо натиснута кнопка S2, світлодіод миготітиме зі зменшеною

частотою й динамік генерує звук з частотою приблизно 1 кГц. Після відпускання кнопки світлодіод почне миготіти зі звичайною частотою, а генерація звуку припиниться.

Зміст звіту

1. Найменування і мета лабораторної роботи.
2. Короткі теоретичні відомості
3. Отримане у викладача завдання.
4. Опис виконання кожного пункту завдання.
5. Письмові відповіді на контрольні питання.

Контрольні питання

1. Особливості архітектури та структури мікроконтролерів PIC16F8х.
2. Характеристика організації пам'яті.
3. Характеристика каналів уведення/виведення.
4. Режими роботи та спеціальні символи.
5. Характеристика системи команд.
6. Основні параметри й характеристики.

Література: [1, 3, 5].

Лабораторна робота № 3

Тема. Дослідження організації відображення інформації в мікропроцесорних системах вимірювання та захисту електротехнічного обладнання

Мета: набуття навичок створення програмних засобів для керування пристроями відображення інформації в мікропроцесорних системах вимірювання та захисту електротехнічного обладнання.

Короткі теоретичні відомості

У багатьох випадках у мікропроцесорних системах вимірювання та захисту електротехнічного обладнання потрібна наявність тільки простої індикації типу ТАК/НІ, УВИМКНЕНО/ВИМКНЕНО. Така індикація реалізується на основі дискретних світлодіодів.

Для відображення інформації з великим числом символів використовуються лінійні (однорядкові) дисплеї. Такі дисплеї являють собою «лінійку» індикаторів (світлодіодних або рідкокристалічних). Число знакомісць дисплея визначається відповідно до вимог до мікроконтролерних пристроїв. При цьому для відображення цифрової й літерної інформації широко використовуються семисегментні індикатори.

Існують два способи реалізації відображення символічної інформації: статичний і динамічний.

Перший вимагає наявності на входах кожного індикатора спеціальних буферних регістрів для зберігання кодів символів, що виводяться на індикатор. Природно, що зі збільшенням розрядності дисплея зростає число додаткових регістрів, а отже, і вартість мікроконтролерних пристроїв.

Другий спосіб (динамічний) ґрунтується на тому, що будь-який світловий індикатор є інерційним приладом, а людському оку інформація, що відображається на дисплеї, якщо її оновлювати з частотою більш ніж 20 разів на секунду, здається незмінною. Динамічний спосіб виведення інформації на дисплей вимагає значно менших апаратних витрат, але складнішого програмного забезпечення. Саме цей спосіб організації відображення інформації набув переважного поширення в мікроконтролерних пристроях.

Хід роботи

Розглянемо завдання, у якому необхідно запропонувати варіант мікроконтролерного пристрою для динамічної індикації 8-розрядних десяткових чисел на 7-сегментних рідкокристалічних індикаторах (РКІ). Десяткові числа, що відображаються, є результатом обчислень, виконаних програмою, що зберігаються у вигляді двійкового коду в пам'яті даних мікроконтролера типу PIC16F84.

Таку задачу можна розв'язати різними способами. Розглянемо детальніше один з можливих варіантів розв'язання поставленої задачі. Для реалізації початкового технічного завдання в цьому разі необхідно здійснити (рис. 3.1):

1. Переписати початкові двійкові коди чисел, що мають виводитися на індикацію, з реєстрів DATx, у яких вони формуються в процесі оброблення даних у МК, у реєстри зберігання INPx, відведені для цих цілей у пам'яті даних мікроконтролера.

2. Перетворити початкові двійкові коди на двійково-десяткові й записати результат до реєстрів виведення OUTx, відведені для цього також у пам'яті даних.

3. Вивести отримані дані для індикації на 8-розрядному лінійному дисплеї, що складається з 8 РКІ індикаторів.

Причому для здійснення динамічної індикації 8-розрядних десяткових чисел необхідно виконати такі дії:

1) сформувати 3-розрядний двійковий код адреси чергового знакомісця (індикатора) у 8-розрядному дисплеї;

2) перетворити отриманий код адреси на сигнали вибору чергового індикатора ВБР-І;

3) перетворити двійково-десятковий код, відповідний вибраному розряду числа, на 7-сегментний код ВБР-С, що дозволяє засвітити необхідні сегменти (забезпечити протікання струму через сегменти) у вибраному індикаторі.

4) засвічувати вибрані сегменти протягом не менше 7–10 мс. Для цього можна використовувати часову затримку вибору наступного індикатора;

5) повторити п. 1–4 для кожного знаку 8-розрядного десяткового числа. Таким чином, в одному повному циклі індикації числа п. 1–4 повторюються вісім разів;

6) повторювати п. 1–5 безперервно, поки ввімкнено живлення пристрою. Для цього необхідно організувати зовнішній цикл регенерації, під час якого за необхідності можна буде здійснювати зміну кодів чисел у реєстрах зберігання INPx.



Рисунок 3.1 – Загальний алгоритм програми реалізації динамічної індикації

Перераховані вище функції можна реалізувати або чисто програмним способом, або із залученням додаткових апаратних засобів. Другий спосіб більш прийнятний, оскільки вимагає меншої кількості ліній уведення/виведення і звільняє мікроконтролер від виконання дій з п. 2 і п. 3 запропонованого вище алгоритму індикації. Для цього необхідно використовувати два дешифратори DC1 і DC2, перший з яких призначений для реалізації п.2, а другий – п. 3 розглянутого алгоритму. Підключення DC1, DC2 і РКІ до виводів портів А і В зображено на рис. 3.2.

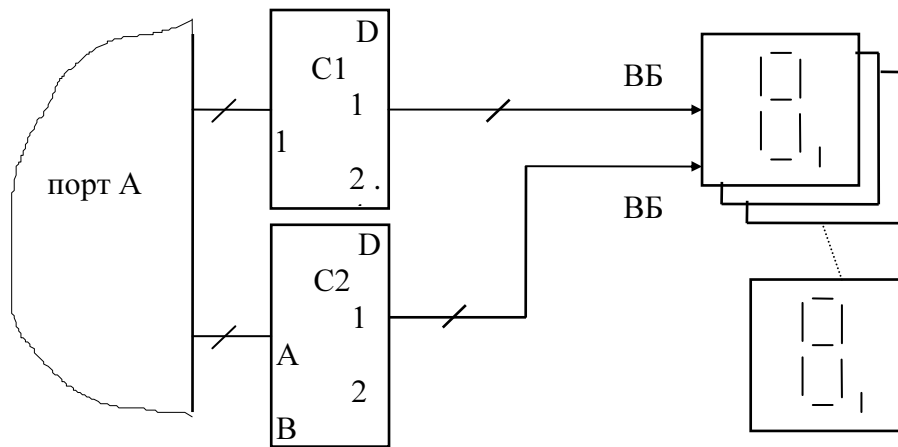


Рисунок 3.2 – Підключення додаткових апаратних засобів до виводів мікроконтролера

На підставі розглянутих вище принципів побудови пристрою для динамічної індикації, алгоритму його функціонування та розподілу пам'яті даних запропоновано варіант тексту програми, який наведений нижче:

```
list p=16F84           ;тип процесора
#include <pi6F84.inc>   ;файл опису апаратних засобів МК
_CONFIG_CP_OFF&_WDT_OFF&_PWRTE_ON &_XT_OSC
                        ;Опис регістрів пам'яті даних
DELAY0 EQU 0x0C        ;Регістри-лічильники для організації
DELAY1 EQU 0x0D        ;затримки (DELAY) більшої тривалості
                        ;(7-10 мс)
DAT0 EQU 0x0E          ;Регістри, у яких формуються двійкові
DAT1 EQU 0x0F          ;коди чисел
DAT2 EQU 0x10          ;дані, що виводяться на індикацію
INP1 EQU 0x11          ;Регістри зберігання двійкового
INP2 EQU 0x12          ;коду числа, що відображається
INP3 EQU 0x13
OUT0 EQU 0x14          ;Регістри зберігання двійково-десятькового
OUT1 EQU 0x15          ;коду числа, що відображається
OUT2 EQU 0x16
```


OUT3 EQU 0x17

```
                                ;Робоча секція
                                ;Початок основної програми
ORG 0x000                       ;Наступна команда буде
                                ;розташована за адресою 00h

goto begin

indication                       ;вивід на індикатор (Процедура
                                ;динамічної індикації)

    bsf STATUS, RP0             ;установлюємо активною
                                ;другу сторінку пам'яті

    movlw b'00000000'          ;установлюємо напрямки роботи
    movlw TRISA                 ;усіх виводів порту А
                                ;на виведення інформації

    movlw b'00000000'          ;установлюємо напрям роботи
    movlw TRISB                 ;усіх виводів порту В
                                ;на виведення інформації

    bcf STATUS, RP0           ;знову робимо активною першу
                                ;сторінку пам'яті

next    clrw                    ;очищуємо регістр W
    movlw PORTA                ;записуємо адресу сегмента В у порт А
    movf OUT0, 0               ;записуємо дані для сегмента в порт В
    movwf PORTB                ;
    call DELAY                 ;викликаємо процедуру затримки
    incf PORTA,1               ;решта частини програми виконує
    swapf OUT0, 0              ;аналогічні дії
    movwf PORTB
    call DELAY
    incf PORTA,1
    movf OUT1, 0
    movwf PORTB
```

```
call delay
incf PORTA, 1
swapf OUT1, 0
movwf PORTB
call delay
incf PORTA,1
movf OUT2, 0
movwf PORTB
call delay
incf PORTA,1
swapf OUT2, 0
movwf PORTB
call delay
incf PORTA, 1
movf OUT3, 0
movwf PORTB
call delay
incf PORTA, 1
swapf OUT3, 0
movwf PORTB
call delay
goto next
```

delay ;Процедура затримки на 200000 тактів

```
movlw 0xFF
movwf DELAY0
```

delay_0

```
movlw 0xFF
movwf DELAY1
```

delay_1 decfsz DELAY1, 1

```
goto delay_1
```

```

    decfsz DELAY0, 1
    goto delay_0
    return
begin
    movf DAT0, 0           ;запис сформованого в
    movwf INP1            ;регістрах DAT0, DAT1 і DAT2
    movf DAT1, 0         ;двійкового коду числа в регістри
    movwf INP2           ;зберігання INP1,INP2,INP3
    movf DAT2, 0
    movwf INP3
    call bin-dec          ;Дана процедура переводу двійкового коду,
;що зберігається в регістрах INP1, INP2, INP3 (причому старший
;байт у регістрі INP1, а молодший – у регістрі INP3), у двійково-
;десятковий код по одній тетрадї в регістри OUT0, OUT1, OUT2, OUT3.
;При цьому старша тетрада регістра OUT0 містить двійковий код
;старшого розряду десятичного числа, а молодша тетрада регістра
;OUT3 містить двійковий код молодшого розряду десятичного числа.
    goto indication      ;Процедура індикації
end                       ;Кінець програми

```

Цей фрагмент можна оформити як стандартну підпрограму для динамічної індикації. Тому, а також для спрощення прикладу, підпрограма перетворення двійкового коду числа, що відображається, на двійково-десятковий код не включена до тексту наведеного фрагмента програми.

Зміст звіту

1. Найменування і мета лабораторної роботи.
2. Короткі теоретичні відомості
3. Отримане у викладача завдання.
4. Опис виконання кожного пункту завдання.
5. Письмові відповіді на контрольні питання.

Контрольні питання

1. Розкрити структуру портів уведення/виведення мікроконтролерів сімейства PIC16F8x, дати характеристику їх функціональних вузлів і програмно-доступних регістрів.
2. Назвати програмно-доступні регістри мікроконтролерів сімейства PIC16F8x і дати характеристику організації паралельного обміну.
3. Охарактеризувати режими функціонування таймера-лічильника в мікроконтролерах PIC16F8x.
4. Дати характеристику організації системи стеження за нормальним функціонуванням МК PIC16F8x.
5. Дати коротку характеристику системи переривань МК сімейства PIC16F8x.
6. Охарактеризувати мікроконтролери сімейства PICmicro. Дати аналіз програмної моделі та функціональних можливостей МК сімейства PIC16F8x.

Література: [1, 3, 4, 5].

Лабораторна робота № 4

Тема. Дослідження організації оброблення запитів переривань мікроконтролерів сімейства PICmicro

Мета: отримання початкових навичок створення програмних додатків для мікроконтролерів PICmicro, що реалізують обмін інформацією за перериваннями.

Короткі теоретичні відомості

Мікроконтролери PIC16Fx мають два зовнішні та два внутрішні джерела переривань. Як приклад розглянемо зовнішнє переривання за входом RB0/INT. Зрозумівши роботу за цим прикладом, нескладно організувати опрацювання зовнішнього переривання за зміною стану ліній RB4-RB7.

Переривання за входом RB0/INT може відбуватися як за наростанням, так і за спаданням фронту вхідного сигналу. Зробимо так, щоб переривання викликалося натисненням кнопки S1. У початковому перебуванні на вході

підтримується високий логічний рівень за допомогою внутрішнього підтягувального резистора. З натисненням кнопки відбувається перехід від високого рівня до низького. Тому для генерації запиту переривання у момент замикання контактів кнопки біт INTEDG регістра OPTION_REG має бути встановлений у нуль.

Окрім цього, слід урахувати, що вектор переривання мікроконтролера всього один і знаходиться за адресою 0004h. Тому з адреси 0004h починається підпрограма оброблення переривання, а за адресою вектора скидання 0000h розміщують команду переходу на мітку, розташовану після закінчення підпрограми оброблення переривання.

Хід роботи

1. Скласти програму за завданням викладача.
2. Увести програму в текстовому редакторі інтегрованого середовища MPLAB IDE.
3. Перевірити програму на відсутність синтаксичних помилок. Виконати асемблерування вихідного тексту програми.
4. Перевести середовище MPLAB IDE в режим стимулятора. Виконати програму в покроковому режимі, пересвідчитися в правильності логіки функціонування програми, проконтролювати значення змінних, які змінюються з кожним кроком та проконтролювати правильність обчислень.
5. У стані очікування програма буде керувати вмиканням і вимиканням світлодіода, а натиснення кнопки S1 викликає підпрограму генерації звукового сигналу тривалістю приблизно 0,5 с.

```
;*****
```

```
;оброблення переривання за спадом рівня за лінією RB0/INT
```

```
list p=16F84
```

```
#include <pl6F84.inc>
```

```
_CONFIG _CP_OFF&WDT_OFF&_PWRTE_ON&_XT_OSC
```

```
count1 EQU 0x0C ;змінна циклу
```

```
count2 EQU 0x0D ;змінна циклу
```

```

count3 EQU 0x0E                ;змінна циклу
ledset EQU 0x0F                ;значення для виводу через порт
st_tmp EQU 0x10                ;чарунка для зберігання STATUS
w_tmp EQU 0x11                ;чарунка для тимчасового зберігання w
    ORG 0x000                  ;вектор скидання процесора
    goto main                  ;перехід на початок основної програми
;*****початок обробки переривання*****
    ORG 0x004                  ;вектор переривання
    movwf w_tmp                ;збереження поточного значення
                                ;акумулятора W
    movf STATUS,w              ;перенесення значення регістра
                                ;STATUS до W
    movwf st_tmp                ;збереження значення регістра
    STATUS
;-----формування звукового сигналу-----
    movlw 0x10
    movwf count 1
beep1 movlw 0x10
    movwf count2
beep2 movlw 0xA0
    movwf count3
loop_b movlw b'00010000'        ;інвертуємо значення на лінії RB4
    xorwf ledset,f              ;формування затримки на тривалість
                                ;одного
    movf ledset, w              ;півперіоду звукової частоти
    movwf PORTB
    decfsz count3, f
    goto loop_b
    decfsz count!, f
    goto beep2

```

```

decfsz countl,f
goto beepl
;*****закінчення звукового сигналу*****
movf st_tmp,w           ;перенесення копії регістра STATUS
movwf STATUS           ;відновлення попереднього значення
                        ;регістра STATUS
swapf w_tmp,f
swapf w_tmp,w           ;відновлення колишнього значення w
bcf INTCON, INTF       ;скидання прапора переривання
retfie                  ;повернення з підпрограми оброб-
                        ;лення переривання

```

```
main clrf PORTA
```

```
clrf PORTB
```

```
bsf STATUS, RP0
```

```
movlw b'1111101'
```

```
movwf PORTA
```

```
movlw b'00000101'
```

```
movwf PORTB
```

```
movlw b'00111111'
```

```
;включаємо підтягувальні резистори
```

```
movwf OPTION_REG
```

```
;та переривання за спадом рівня
```

```
movlw b'10010000'
```

```
;вирішуємо переривання за входом
```

```
movwf INTCON
```

```
;RB0/INT, решту переривань
```

```
;забороняємо
```

```
bcf STATUS, RP0
```

```
clrf ledset
```

```
loop movlw b'00000010'
```

```
xorwf ledset, f
```

```
movf ledset, w
```

```
movwf PORTA
```

```
movlw 0xFF
```

```
    movwf count 1
cycle1 movlw 0xFF
    movwf count1
cycle2 decfsz count2, f
    goto cycle2
    decfsz count 1, f
    goto cycle1
    goto loop
end
```

Після директиви «ORG 0000h» записана команда переходу за міткою main, а потім – директива «ORG 0004h». Починаючи з цієї адреси, розташовуватиметься підпрограма оброблення запиту переривання. Вона починається стандартним фрагментом коду, що зберігає поточні значення акумулятора W і регістра STATUS. Потім іде програма генерації звукового сигналу. Коли всі цикли завершені, відновлюються збережені значення W і STATUS, очищається прапорець-біт переривання і відбувається повернення з підпрограми обробки переривання.

У основному фрагменті підпрограми додалася настроювання службових регістрів OPTION_REG і INTCON. Настроювання портів і керування світлодіодом відбувається так само, як і в попередніх програмах.

Зміст звіту

1. Найменування і мета лабораторної роботи.
2. Короткі теоретичні відомості
3. Отримане у викладача завдання.
4. Опис виконання кожного пункту завдання.
5. Письмові відповіді на контрольні питання.

Контрольні питання

1. Дайте коротку характеристику системи переривань мікроконтролерів PICmicro.
2. Пояснити відмінності оброблення зовнішніх і внутрішніх запитів

переривань PIC-контролерів.

3. Дайте характеристику регістра INTCON. Розкрийте призначення та функції окремих розрядів цього регістра.

4. Як налаштувати мікроконтролер на оброблення запиту зовнішнього переривання за зміною рівнів на лініях RB4-RB7?

5. Як реалізувати оброблення запиту на переривання за переповненням таймера-лічильника TMR0?

6. Поясніть особливості організації програми мікроконтролера в разі реалізації оброблення запитів на переривання.

7. Як здійснюється вихід з підпрограми оброблення переривань, і як змінюється при цьому вміст стека та лічильника команд?

Література: [4, 5, 6].

Лабораторна робота № 5

Тема. Дослідження організації введення інформації з датчиків у мікропроцесорних системах вимірювання та захисту електротехнічного обладнання

Мета: отримання початкових навичок практичної роботи із зовнішніми портами мікроконтролера та використовувати цифрові сигнали від датчиків і формувати дискретні сигнали керування.

Короткі теоретичні відомості

У технічних системах різного призначення події в об'єкті керування фіксуються за допомогою різноманітних датчиків цифрового та аналогового типів.

Значне поширення мають двійкові датчики типу «ТАК/НІ», наприклад, кінцеві вимикачі. Сигнали на виходах таких датчиків є звичайними дискретними статичними сигналами, що змінюються або з «0» на «1», або з «1» на «0».

Даний тип сигналів є також найпоширенішим і серед сигналів керування, що формуються мікроконтролерами.

Найчастіше мікроконтролери мають справу не з одиночним датчиком, а з групою автономних (логічно незалежних) або взаємопов'язаних (що формують паралельний двійковий код) датчиків.

Вихідні сигнали мікроконтролерів також часто є паралельними двійковими кодами (наприклад, у разі індикації даних, передачі їх до ПЕВМ і так далі).

Імпульсні сигнали та їх послідовності також достатньо часто використовуються як керувальні сигнали. Різновидом таких сигналів є сигнали з широтно-імпульсною модуляцією (ШІМ). У такій послідовності сигналів можуть змінюватися в часі як тривалість імпульсів (t_i), так і пауза між ними (t_n).

Датчики, що формують дискретні й дуже рідко – імпульсні сигнали можуть бути безконтактні й контактні. У останньому випадку виникає необхідність усунення паразитних замикань контактів апаратними або програмними засобами. Крім того, рівні сигналів з виходів датчиків можуть істотно відрізнятися від рівнів ТТЛ-логіки, прийнятих у мікроконтролерах РІС 16/17. У такому разі необхідно на входах мікроконтролера встановлювати спеціальні перетворювачі рівнів сигналів.

Аналогові сигнали можуть використовуватися також як керувальні. У такому разі до виходів мікроконтролера необхідно підключати цифро-аналогові перетворювачі (ЦАП) і далі, якщо необхідно, підсилювачі потужності й оптронні роздільники сигналів.

Хід роботи

За допомогою мікроконтролера РІС16F84 здійснити опитування двійкового датчика і, залежно від його стану, або організувати процедуру «очікування події», або сформувати й видати керуючий сигнал у вигляді позитивного імпульсу заданої тривалості.

Для розв'язання завдання необхідно:

- 1) вибрати лінії (розряди) портів А і В для введення сигналів від датчика й виведення імпульсних сигналів (наприклад, RB0 – на введення, а RA0 – на виведення);

2) заборонити всі переривання;

3) здійснити опитування стану датчика;

4) проаналізувати сигнал (наприклад, перевіривши біт, що сигналізує про стан датчика, тобто визначити, відбулося в поточному циклі опитування події в об'єкті керування або ні – чи був зафіксований черговий перехід з «0» в «1»);

5) організувати процедуру передачі керування на дві підпрограми залежно від рівня сигналу, отриманого від датчика;

6) сформувати позитивний імпульсний сигнал (IMPULS) заданої тривалості у випадку, якщо, наприклад, стан датчика дорівнює «1» ($D = 1$). Для організації затримки (DELAY) можна використовувати:

- одну або декілька команд порожньої операції NOP;
- програмний лічильник, що працює, наприклад, у режимі декремента.

При цьому до лічильника заноситься константа, лічильник декрементується. Обнуління лічильника є сигналом закінчення часового інтервалу;

- внутрішній таймер-лічильник (TMR0);
- зовнішній (додатковий) таймер-лічильник у разі, якщо TMR0 з якої-небудь причини використовувати неможливо;

7) організувати процедуру очікування події, яка полягає в безперервному опитуванні датчика та аналізі його стану.

На підставі викладеного вище можна запропонувати варіант функціональної схеми мікроконтролерного пристрою, наведену на рис. 5.1.

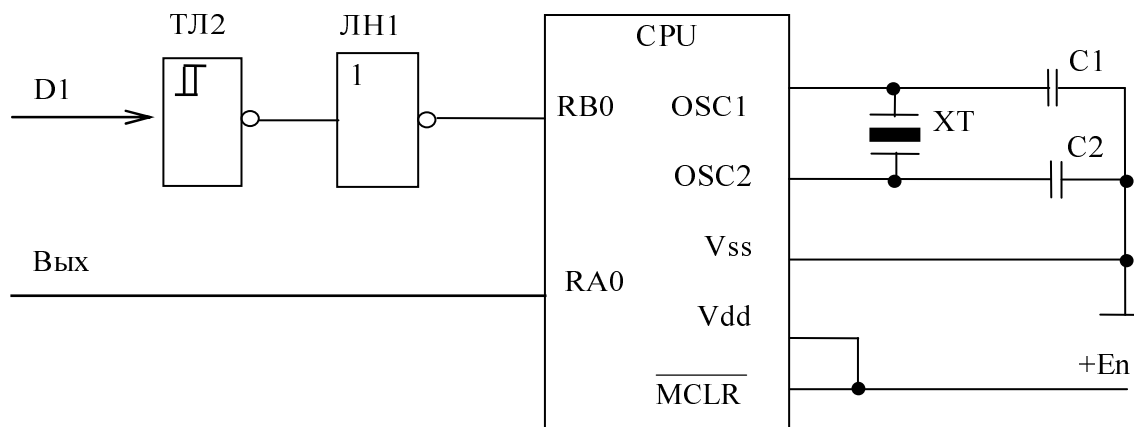


Рисунок 5.1 – Функціональна схема пристрою введення інформації з датчиків

8) на підставі блок-схем алгоритмів і розподілу пам'яті даних створений початковий текст робочої програми. Варіант тексту програми, яка дозволяє реалізувати всі функції, закладені в початковому технічному завданні, наведений нижче:

;Текст програми

LIST p=16F84

;Секція заголовка

;Опис чарунок ОЗП

ctd EQU 0Dh

;Лічильник циклу в підпрограмі

;DELAY

;Робоча секція

;Точка входу в програму після RESET

ORG 0

goto inmk

ORG 100h

;Процедура ініціалізації МК

;Заборона переривань

inmk bcf INTCON, 7

;Скидання 7-го біта в регістрі

;INTCON.

;Завантаження в регістр керування

;портом В

movlw B'11111111'

;Переслати константу в регістр

TRISB

;W і призначити всі лінії порту В

;на введення.

;Завантаження регістра керування

;портом А

movlw B'00000000'

;Переслати константу в регістр

TRISA

;W і призначити всі лінії порту

;А як вихідні

bcf PORTA, 0

;Скидання 0-го біта в регістрі

```

;порту А
;Опитування стану датчика і процедура «очікування події»
M1  btfss PORTB, 0          ;Аналіз 0-го біта в регістрі
                                ;порту В. Якщо сигнал датчика
                                ;дорівнює «1»
                                ;то пропустити наступну
                                goto M1          ;команду, інакше – очікувати
                                ;Видача імпульсу заданої тривалості
                                bsf PORTA,0
                                ;Затримка (DELAY)
                                movlw B'00001111' ;Завантаження лічильника затримки
                                movwf CTD          ;Цикл затримки
M2  decfsz CTD             ;Декремент CTD, пропустити
                                ;команду, якщо 0 (якщо регістр
                                goto M2          ;CTD очистився)
                                bcf PORTA, 0     ;Припинити видачу сигналу
                                goto M1
                                end

```

9) увести програму за допомогою вбудованого редактора інтегрованого середовища MPLAB IDE.

10) за допомогою засобів налагодження перевірити правильність виконання програми.

Зміст звіту

1. Найменування і мета лабораторної роботи.
2. Короткі теоретичні відомості
3. Отримане у викладача завдання.
4. Опис виконання кожного пункту завдання.
5. Письмові відповіді на контрольні питання.

Контрольні питання

1. У чому полягає різниця у структурі портів введення/виведення для

мікроконтролерів серії PIC16F8x?

2. Пояснити принципи завдання напряму введення/виведення для зовнішніх ліній контролера.

3. Пояснити особливості організації введення дискретних сигналів від датчиків через порти введення/виведення.

4. Пояснити послідовність операцій з формування на портах уведення/виведення імпульсних сигналів.

5. Пояснити особливості роботи послідовного порту введення/виведення мікроконтролерів PIC16.

Література: [1, 2, 5, 7].

Лабораторна робота № 6

Тема. Дослідження організації роботи з аналогово-цифровим перетворювачем. Дискретизація аналогового сигналу та передача його на ЕОМ

Мета: отримання навичок практичної реалізації протоколу повнодуплексного режиму обміну інформацією за інтерфейсом RS-232.

Короткі теоретичні відомості

USART – це один з модулів послідовного порту вводу/виводу, що може працювати в повнодуплексному асинхронному режимі для зв'язку з терміналами, персональними комп'ютерами або в синхронному напівдуплексному режимі для зв'язку з мікросхемами ЦАП, АЦП, послідовними EEPROM і т. д.

USART може працювати в одному із трьох режимів:

- Асинхронний, повний дуплекс;
- Ведучий синхронний, напівдуплекс;
- Ведений синхронний, напівдуплекс.

Біти SPEN (RCSTA<7>) і TRIS повинні бути встановлені в «1» для використання ліній TX/CK і RX/DT як порту універсального синхронно-

асинхронного приймача-передавача. Модуль USART підтримує режим детектування 9-розрядної адреси для роботи в мережному режимі.

Хід роботи

1. Ознайомитися з теоретичними відомостями.
2. Створити проект для програми мікроконтролера.
3. Набрати програму демонстраційного прикладу, що приводиться.
4. Скласти проект (компілювати) та виправити синтаксичні помилки в коді програми.
5. Розібрати приклади організації масивів констант та даних, циклічної роботи з масивами даних, як створюються програмні розгалуження, методи прямої та непрямой адресації.
6. Отримати індивідуальне завдання на захист роботи від викладача
7. Дати відповідь на контрольні питання самоперевірки.
8. Оформити звіт по роботі.

Зміст звіту

1. Найменування і мета лабораторної роботи.
2. Короткі теоретичні відомості.
3. Отримане у викладача завдання.
4. Опис виконання кожного пункту завдання.
5. Письмові відповіді на контрольні питання.

Контрольні питання

1. Охарактеризуйте асинхронний режим USART.
2. Поясніть особливості функціонування генератора швидкості обміну SPBRG.
3. Назвіть послідовність дій під час налагоджування приймача USART.
4. Які дії виконуються для налагоджування передавача USART.
5. Організація роботи з приймачем-передавачем інформації за допомогою прапорців подій і переривань.
6. Стандарт інформаційного обміну USART.

Література: [3, 4, 7].

Лабораторна робота № 7

Тема. Дослідження організації роботи з аналогово-цифровим перетворювачем. Дискретизація аналогового сигналу та передача його на ЕОМ

Мета: отримання навичок практичної реалізації опитування аналогових входів аналогово-цифрового перетворювача.

Короткі теоретичні відомості

Модуль аналого-цифрового перетворення має до восьми вхідних каналів. Вхідний аналоговий сигнал через комутатор каналів заряджає внутрішній конденсатор АЦП C_{hold} . Модуль АЦП перетворить напругу, утримувану на конденсаторі C_{hold} у відповідний 10-розрядний цифровий код методом послідовного наближення. Джерело опорної напруги може бути програмно обране з ліній V_{dd} , V_{ss} , V_{ref+} або V_{ref-} . Допускається робота модуля АЦП в SLEEP режимі мікроконтролера, при цьому як джерело тактових імпульсів для АЦП повинен бути обраний RC генератор.

Для керування АЦП у мікроконтролері використовується 4 регістри:

- Старший регістр результату ADRESH;
- Молодший регістр результату ADRESL;
- Регістр керування ADCON0;
- Регістр керування ADCON1.

Регістр ADCON0 використовується для налаштування роботи модуля АЦП, а за допомогою регістра ADCON1 встановлюється, які входи мікроконтролера будуть використовуватися модулем АЦП і в якому режимі.

Робота модуля АЦП

У регістрах ADRES x зберігається 10-розрядний результат аналого-цифрового перетворення. Коли перетворення завершено, результат перетворення записується в регістри ADRES x , після чого скидається біт GO/DONE (ADCON0<2>) і встановлюється прапор переривання ADIF.

Після включення й налаштування АЦП необхідно вибрати робочий аналоговий канал. Відповідні біти TRI аналогових каналів мають набудувати

канал порту вводу/виводу на вхід. Перед початком перетворення необхідно витримати тимчасову паузу, розрахунок якої наведений у розділі.

Хід роботи

1. Ознайомитися з теоретичними відомостями
2. Створити проект для програми мікроконтролера.
3. Набрати програму демонстраційного прикладу, що приводиться.
4. Скласти проект (компілювати) і виправити синтаксичні помилки в кодї програми.
5. Розібрати приклади організації масивів констант і даних, циклічної роботи з масивами даних, як створюються програмні розгалуження, методи прямої та непрямої адресації.
6. Отримати індивідуальне завдання на захист роботи від викладача.
7. Дати відповідь на контрольні питання.
8. Оформити звіт по роботі.

Зміст звіту

1. Найменування і мета лабораторної роботи.
2. Короткі теоретичні відомості.
3. Отримане у викладача завдання.
4. Опис виконання кожного пункту завдання.
5. Письмові відповіді на контрольні питання.

Контрольні питання

1. Поясніть призначення регістру ADCON0.
2. Поясніть призначення регістру ADCON1.
3. Поясніть послідовність дій під час налагоджування роботи АЦП.
4. Дайте характеристику організації роботи з аналогово-цифровим перетворювачем сигналу за допомогою прапорців подій і переривань.
5. Дайте характеристику організації роботи з АЦП за допомогою таймерів.

Література: [3, 4, 7].

Лабораторна робота № 8

Тема. Дослідження організації роботи з периферійним модулем ССР в режимі ШІМ

Мета: отримання навичок практичної реалізації реалізовувати режимів широтно-імпульсної модуляції.

Короткі теоретичні відомості

Кожний модуль ССР містить 16-розрядний регістр, що може використовуватися в якості:

- 16-розрядного регістра захоплення даних;
- 16-розрядного регістра порівняння;
- Двох 8-розрядних (ведучий і ведений) регістрів ШІМ.

Режим ШІМ

У режимі ШІМ модуля ССР лінія ССРх використовується як вихід 10-розрядного ШІМ. Лінія ССРх об'єднана з цифровим каналом порту вводу/виводу, тому відповідний біт напрямку TRIS повинен бути скинутий в «0».

Період ШІМ визначається значенням у регістрі PR2.

Коли значення TMR2 порівнюється з PR2, виконуються такі дії:

- TMR2 скидається в 00h;
- Установлюється високий рівень сигналу на лінії ССРх (якщо шпаруватість дорівнює 0 %, то сигнал у високий рівень установлюватися не буде);
- Модуль ШІМ починає новий цикл, завантажуючи значення з регістра ССРхL в ССРх.

Тривалість імпульсу ШІМ

Тривалість імпульсу ШІМ визначається бітами в регістрах ССРхL і ССРхCON<5:4>. Для 10-розрядного ШІМ старші вісім біт зберігаються в регістрі ССРх, а молодші два біти – у регістрі ССРхCON<5:4> (ССРх:ССРхCON<5:4>). 10-розрядне значення представляється бітами DCx9:DCx0.

Біти в регістрі CCPRx і CCPxCON<5:4> можуть бути змінені в будь-який час, але значення в регістрі CCPRx не змінюється, поки не відбудеться відповідність PR2 і TMR2. У режимі ШІМ регістр CCPRx доступний тільки для читання.

Регістр CCPRx і внутрішній двохранний проміжний регістр утворюють буфер ШІМ. Ефект буферизації необхідний для запису нового значення тривалості імпульсу ШІМ.

Коли значення CCPRx і 2-розрядної внутрішньої проміжного регістру відповідає значенню TMR2 і внутрішньому 2-розрядному лічильнику, у такті Q2 на лінії CCPx буде встановлений низький рівень сигналу (кінець імпульсу ШІМ).

Примітка. Якщо тривалість імпульсу ШІМ більша періоду ШІМ, лінія PCP1 не буде мати низький рівень сигналу, що дозволяє реалізувати шпаруватість вихідного сигналу 100 %.

Мінімальна роздільна здатність

Мінімальна роздільна здатність кожного біта імпульсу ШІМ залежить від режиму роботи попереднього дільника.

Таблиця 8.1 – Мінімальна тривалість біту імпульсу ШІМ

Коефіцієнт попереднього ділення	T2CKPS1:T2CKPS0	Мінімальна роздільна здатність
1	0	Tosc
4	1	Tcy
16	1x	4Tcy

Послідовність налаштування модуля PCP у режимі ШІМ

Рекомендована послідовність включення модуля PCP у режимі ШІМ:

1. Установити період ШІМ у регістрі PR2;
2. Установити тривалість імпульсу в бітах DCxB9:DCxB0;
3. Установити лінію CCPx як вихідну, скинувши відповідний біт TRIS;
4. Налаштувати попередній дільник і включити TMR2 у регістрі T2CON;

5. Включити ССР у режимі ШІМ.

Хід роботи

1. Ознайомитися з теоретичними відомостями
2. Створити проект для програми мікроконтролера.
3. Набрати програму демонстраційного прикладу, що приводиться.
4. Скласти проект (компілювати) і виправити синтаксичні помилки в коді програми.
5. Розібрати приклади організації масивів констант і даних, циклічної роботи з масивами даних, як створюються програмні розгалуження, методи прямої та непрямой адресації.
6. Отримати індивідуальне завдання на захист роботи від викладача
7. Дати відповідь на контрольні питання самоперевірки.
8. Оформити звіт по роботі.

Зміст звіту

1. Найменування і мета лабораторної роботи.
2. Короткі теоретичні відомості.
3. Отримане у викладача завдання.
4. Опис виконання кожного пункту завдання.
5. Письмові відповіді на контрольні питання.

Контрольні питання

1. Назвіть керуючі регістри ШІМ.
2. Опишіть послідовність дій під час налагоджуванні роботи модуля ШІМ.
3. Як здійснюється обробка запиту переривання від модуля ШІМ.
4. Дайте оцінку зміни довжини імпульсу та роздільної здатності широтно-імпульсної модуляції відповідно до індивідуального завдання.

Література: [1, 2, 7].

3 КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

Кількість лабораторних занять – 18 годин (9 лабораторних занять).

Поточний контроль на лабораторних заняттях протягом змістових модулів:

- відвідування лабораторних занять – 0,5 бала за заняття (максимум 6 балів за семестр);
- виконання лабораторного завдання – 0,5 бала за виконане лабораторне завдання (максимум 6 балів за семестр);
- захист лабораторної роботи – захист на оцінку «відмінно» – 3 бали за лабораторну роботу, захист на оцінку «добре» – 2 бали за лабораторну роботу, захист на оцінку «задовільно» – 1 бал за лабораторну роботу (максимум 27, 18 та 9 балів за семестр);
- своєчасність захисту всіх лабораторних робіт – 0,5 бала за лабораторну роботу (максимум 5 балів за семестр).

СПИСОК ЛИТЕРАТУРИ

1. Александров, В. А. Электронные и микропроцессорные системы управления автомобилей: Уч. Пособие / В. А. Александров, С. Ф. Козьмин, Н. Р. Шоль и др. – СПб.: Лань, 2012. – 624 с.
2. Баев, Б. П. Микропроцессорные системы бытовой техники / Б. П. Баев. – М.: ГЛТ, 2012. – 480 с.
3. Барабанов, Ю. А. Микропроцессорные устройства релейной защиты и автоматики распределительных сетей / Ю. А. Барабанов. – Вологда: Инфра-Инженерия, 2015. – 172 с.
4. Батоврин, В. К. LabVIEW: практикум по электронике и микропроцессорной технике: Учебное пособие / В. К. Батоврин, А. С. Бессонов, В. В. Мошкин. – М.: ДМК, 2014. – 182 с.
5. Гуревич, В. И. Микропроцессорные реле защиты. Устройство, проблемы, перспективы / В. И. Гуревич. – М.: Инфра-Инженерия, 2011. – 336 с.
6. Гусев, В. Г. Электроника и микропроцессорная техника / В.Г. Гусев. – М.: Высшая школа, 2008. – 798 с.
7. Калабегов, Б. А. Цифровые устройства и микропроцессорные системы / Б. А. Калабегов. – М.: Горячая линия-Телеком, 2007. – 336 с.
8. Калашников, В. И. Электроника и микропроцессорная техника: Учебник для студ. учреждений высш. проф. обр. / В.И. Калашников, С.В. Нефедов. – М.: ИЦ Академия, 2012. – 368 с.
9. Микушин, А. В. Цифровые устройства и микропроцессоры / А. В. Микушин. – СПб.: ВНУ, 2010. – 832 с.
10. Микушин, А. В. Программирование микропроцессорных систем на языке С-51. / А. В. Микушин. – М.: Горячая линия -Телеком, 2009. – 216 с.
11. Новиков, Ю. В. Основы микропроцессорной техники: курс лекций / Ю. В. Новиков, П. К. Скоробогатов. – М.: ИНТУИТ.РУ, 2003. – 440 с.
12. Шонфелдер, Г. Измерительные устройства на базе микропроцессора ATmega / Г. Шонфелдер, К. Шнайдер. – СПб.: БХВ-Петербург, 2012. – 288 с.

Зразок оформлення титульної сторінки звіту з лабораторної роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
ІНСТИТУТ ЕЛЕКТРОМЕХАНІКИ, ЕНЕРГОЗБЕРЕЖЕННЯ
І СИСТЕМ КЕРУВАННЯ

КАФЕДРА СИСТЕМ АВТОМАТИЧНОГО КЕРУВАННЯ ТА
ЕЛЕКТРОПРИВОДА

НАВЧАЛЬНА ДИСЦИПЛІНА:
**«МІКРОПРОЦЕСОРНІ СИСТЕМИ ВИМІРЮВАННЯ ТА ЗАХИСТУ
ЕЛЕКТРОТЕХНІЧНОГО ОБЛАДНАННЯ»**

ЗВІТ
З ЛАБОРАТОРНОЇ РОБОТИ № ____

Виконав:
студент групи
ПБ

Перевірив:
ПБ

Кременчук 20__ р.

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Мікропроцесорні системи вимірювання та захисту електротехнічного обладнання» зі спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка» за освітньо-науковою та освітньо-професійною програмами «Електромеханічне обладнання енергоємних виробництв» освітнього ступеня «Магістр»

Укладач старш. викл. М. Ю. Юхименко

Відповідальний за випуск зав. кафедри САУЕ проф. Д. Й. Родькін

Підп. до др. _____. Формат 60x84 1/16. Папір тип. Друк ризографія.
Ум. друк. арк. _____. Наклад _____ прим. Зам. № _____. Безкоштовно.

Видавничий відділ
Кременчуцького національного університету
імені Михайла Остроградського
вул. Першотравнева, 20, м. Кременчук, 39600