МІНІСТЕРСТВО ОСВІТИ І НАУКИ
Івано-Франківський національний технічний університет нафти і газу

Кафедра комп'ютеризованого машинобудівного виробництва

О. Р. Онисько, В. Г. Панчук, А. Г. Панчук,

МІКРОПРОЦЕСОРНЕ УПРАВЛІННЯ МЕХАНОТРОННИХ СИСТЕМ

Програмування мікроконтролерів родини PIC у середовищі MPLab

ЛАБОРАТОРНИЙ ПРАКТИКУМ

Івано-Франківськ
2018

УДК 630.01
ББК 221
П-42

Рецензент:

Райтер П.М. доктор технічних наук, завідувач кафедри технічної діагностики та моніторингу Івано-Франківського національного технічного університету

Рекомендовано методичною радою університету (протокол № від _____ р.)

Онисько О. Р., Панчук А.Г.

О-42 Мікропроцесорне управління механотронних систем. Програмування мікроконтролерів родин PIC у середовищі MPLab. Лабораторний практикум. Івано-Франківськ 2018.- 39 с.

МВ

Розроблений у відповідності до робочої програми навчальної дисципліни та навчального плану підготовки бакалаврів за спеціальністю 131 «Прикладна механіка».

Містить методичні вказівки для вивчення та застосування середовища MPLab для програмування мікроконтролерів родини PIC.

Призначений для виконання лабораторних занять з дисципліни «Мікропроцесорне управління механотронних систем» .

УДК 630.01
ББК 221
П-42
МВ

ЗМІСТ

ВСТУП

Нині мікроконтролери родини PIC займають вагому нішу на ринку мікропроцесорів і набули широкого вжитку в мережах технічного керування у томе числі механотронними системами. Для проведення лабораторних робіт вибрано один із найпоширеніших мікроконтролерів цієї родини PIC 16F84.

Лабораторна робота № 1

СЕРЕДОВИЩЕ ПРОГРАМУВАННЯ *MPLAB IDE*, ВСТУП ДО ПРОГРАМУВАННЯ НА АСЕМБЛЕРІ, **ІНІЦІАЛІЗАЦІЯ ПОРТІВ**

1.1 Мета роботи. Ознайомитися із середовищем програмування мікроконтролерів PIC — *MPLab IDE*. Створити проект програми ініціалізації портів мікроконтролера PIC16F84.

1.2 Теоретичні відомості. Компанія Microchip є розробником родини мікроконтролерів PIC і програмного продукту *MPLab IDE,* яка служить для створення і запису керуючих програм у чіпи(мікросхеми) заданої моделі мікроконтролера.

1.3 Хід роботи

1.3.1. Використовуючи *Total Comander* створіть директорію, у якій мав би розміститися проект з потрібними файлами. Наприклад, нехай це буде директорія Proect1.

1.3.2 Потрібні файли, це у першу чергу трафаретний файл, або ж інакше темеплет, який є фактично зразком–канвою файлу із розширенням *asm**. Тобто по суті це файл – заготовка програмного коду, який написаний спеціалізовано для конкретної моделі PIC. Ці темплети, як правило розміщені у директорії *Microchip*, а значить повний шлях до темплетів найімовірніше є таким: *c:\Program Files (x86)\Microchip\MPASM Suite\Template\Code*.

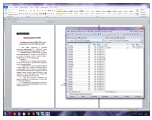

Рисунок 1.1 – Копіювання темплета 16F84TEMP *у директорію Proect1*

1.3.2.1 Серед списку темп летів із розширенням asm* вибираємо потрібний нам 16F84TEMP (рисунок 1.3) і копіюємо його до створеної директорії *Proect1* (рисунок 1.1).

1.3.2.2 Для уникнення «забруднення» трафарету рекомендують переписати скопійований файл 16F84TEMP.asm під іншою назвою, наприклад: *proect1_1.asm* (Рисунок 2).

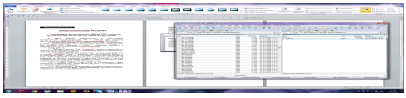

Рисунок 1.2 – Перейменування темплета 16F84TEMP на 16F84n у директорії Proect1

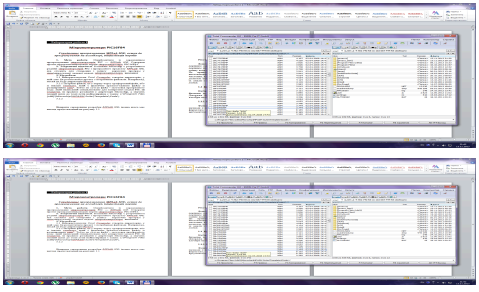

Рисунок 1.3 Фрагмент списку темплетів директрії CODE

1.3.2.3 Вибираємо у тій ж директорії Microchip файл ініціалізації потрібного мікроконтролера. Повний шлях до списку файлів ініціалізації *c:\Program Files (x86)\Microchip\MPASM Suite*

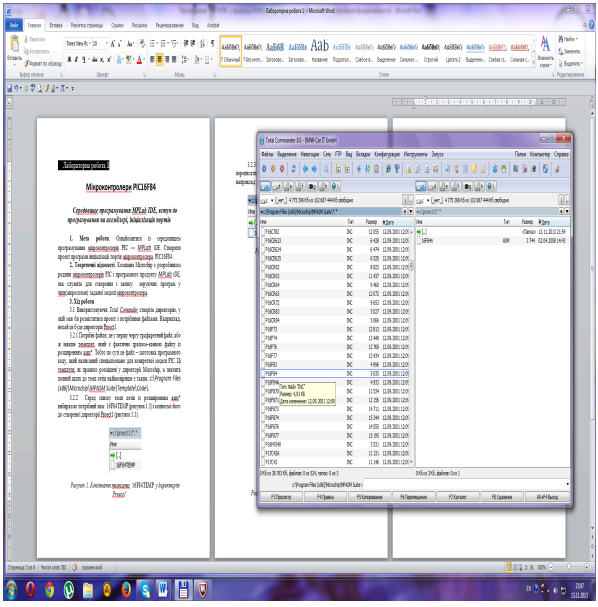

Рисунок 1.4 – Фрагмент списку файлів ініціалізації мікроконтролерів PIC

1.3.2.4 Копіюємо вибраний файл P16F84 inc до папки Proect1.

1.3.2.5 Вибираємо із списку файлів *директорії c:\Program Files (x86)\Microchip\MPASM Suite\LKR* файл з імя'м: 16f84_g *lkr* (рисунок 1.4).

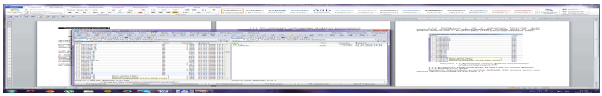

*Рисунок 1.5 – Фрагмент списку файлів lkr**

1.3.2.6 Копіюємо вибраний файл 16f84_g *lkr* у директорію *Proect1*. Відтепер список файлів цієї директорії є таким, як на рисунку 1.6

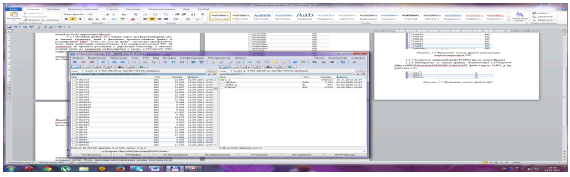

Рисунок 1.6 – Файли проекту Proect1 відібрані з директорії Microchip

1.3.3 Відкриваємо середовище розробки *MPLab IDE,* іконка якого має вигляд представлений на рисунку 1.7

Рисунок 1.7 – Етикетка програми MPLab IDE версії 8.46.

1.3.3.1 У відкритому інтерфейсі середовища *MPLab IDE* (рисунок 1.8) вибираємо із мені Proect командут Proect Wizard.

1.3.3.2 Далі натискаєм кнопку «Дальше». В вікні Step One Select a device вибираємо з набору мікроконтролерів потрібний нам PIC16F84 і знову натискаємо кнопку «Далее» (рисунок 1.9).

1.3.3.3 Наступним відкривається вікно Step Two – *з англ*. крок два, яким пропонують вибрати мову програмування. Вибираємо мову *Assembler*, саме цей момент відображений на рисунку 1.10.

1.3.3.4 Далі появляється вікно – step three в якому пропонують створити файл проекту. Створимо цей файл – наприклад *C:\proect1\pr_1*. Це можна здійснити натиснувши кнопку browz і відтак вибрати директорію *proect1*, в якій і створити файл *pr_1*.

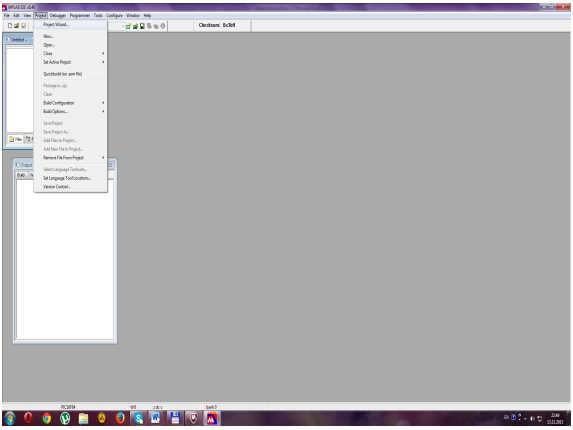

Рисунок 1.8 – Відкриття проекту із ПП MPLab IDE

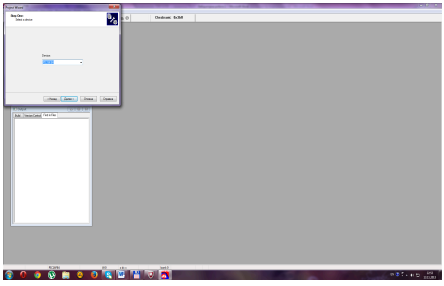

Рисунок 1.9 – Вибір моделі мікронтролера –PIC16F84

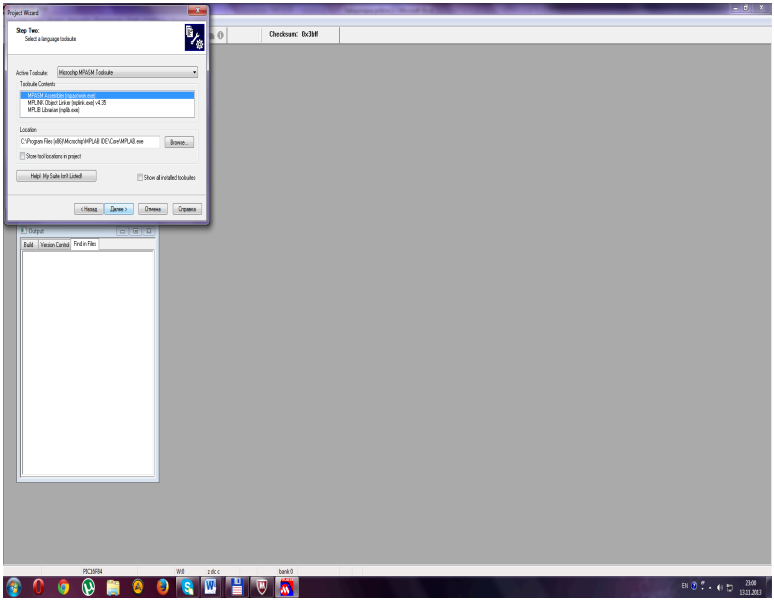

Рисунок 1.10 – Вибір мови програмування.

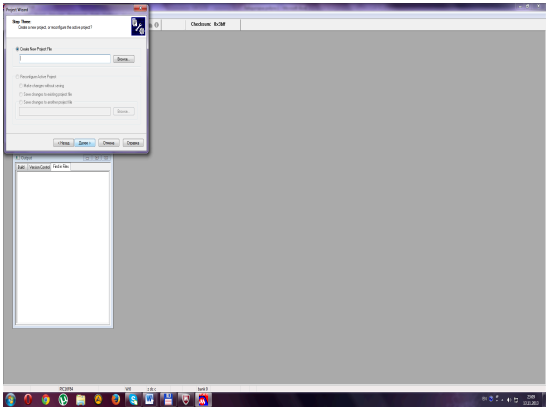

Рисунок 1.11 – Створення файлу проекта

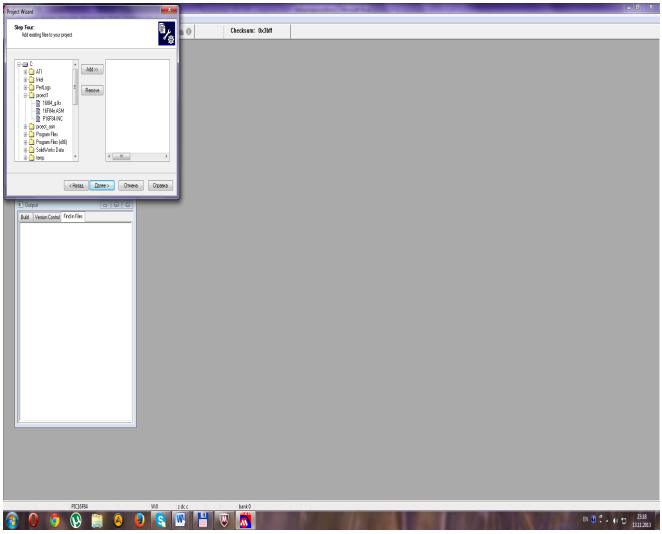

Рисунок 1.12 – Крок чотири – додавання вихідних файлів

1.3.3.5 Додаємо файли 16F84 до списку з допомогою кнопки ***add***

Результат показаний на рисунку 1.13.

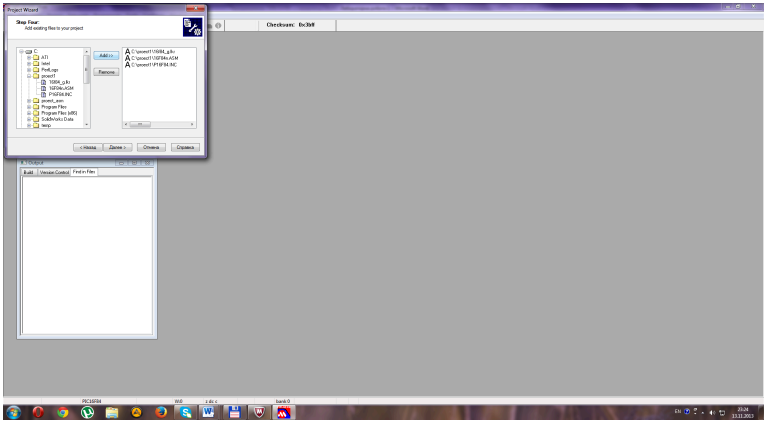

Рисунок 1.13 – Додані до списку файли.

1.3.3.6 У підсумку (Рисунок 1.14) маємо заключну інформацію про створений проект.

Тут варто звернути увагу на наступне:

Назва пристрою: *PIC 16F84*

Інструментарій: Microchip MPASM

File: C:\proect\pr_1.mcp (назва проекту із розширенням *mcp*)

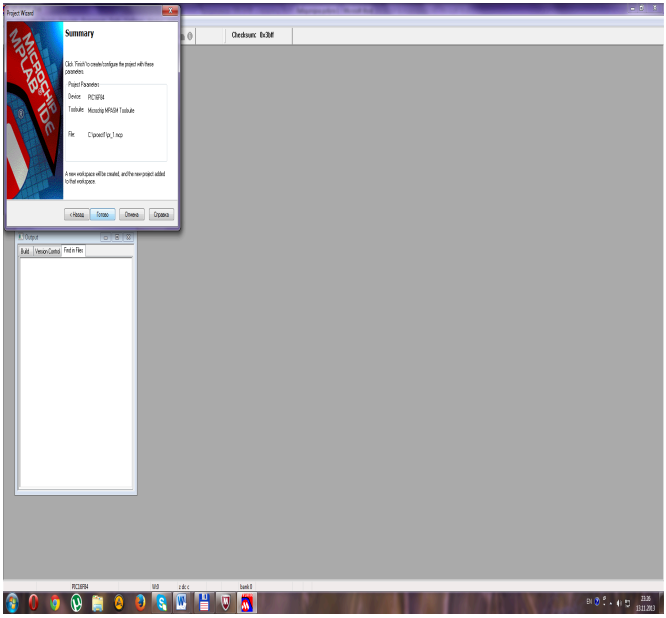

Рисунок 1.14 – Підсумкове вікна розгортання проекту.

1.3.4 Після натискання на кнопку «готово» відкриється вікно проекту *pr_1.mcp*.

1.3.5 У цьому вікні знаходимо файл з розширенням asm* і двічі клацнемо по ньому. У результаті отримаємо вікно коду заготовки програми на мові асемблер для PIC16F84 (Рисунок 1.16).

1.3.5.1 За допомогою лінійки прокручування знаходимо у вікні коду директиву END , яка розміщена наприкінці (внизу) коду і власне вказує на місце завершення програми (Рисунок 1.17).

1.3.5.2 Кількома рядками вище від директиви END знаходимо записане слово **main**, котре вказує на місце початку запису головної програми.

1.3.5.3 Виставляємо курсор після слова **main** (тобто у наступний після **main** рядок) і розпочинаємо запис тіла програми ініціалізації портів мікроконтролера PIC16F84.

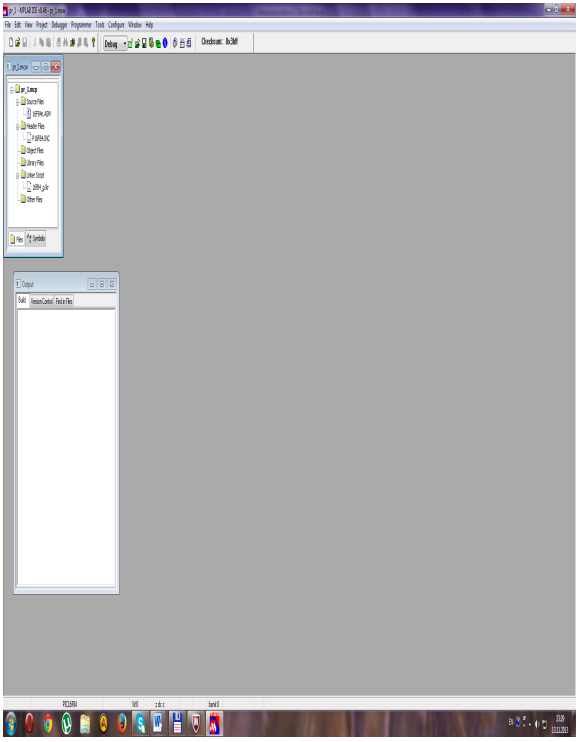

Рисунок 1.15 – Вікно проекту pr_1.mcp

1.3.6 Програма ініціалізації порту А мікроконтролера PIC16F84 представлена у лекції 5 (розділ5.7.2, ***Приклад 5.1***). Ця програма забезпечує налаштування виводів ПОРТу А на таку конфігурацію: 10000. Отже 4 молодших розряди працюватимуть як вихід, а найстарший біт — як вхід.

1.3.6.1 Запис рядків програми слід здійснювати із відступом зліва, хоч би на один пробіл.

1.3.6.2 Користуючись прикладом 5.1 записуємо власноручно, порядково, бажано з коментарями усі чотири рядки програми, яка зображена на рисунку 1.18.

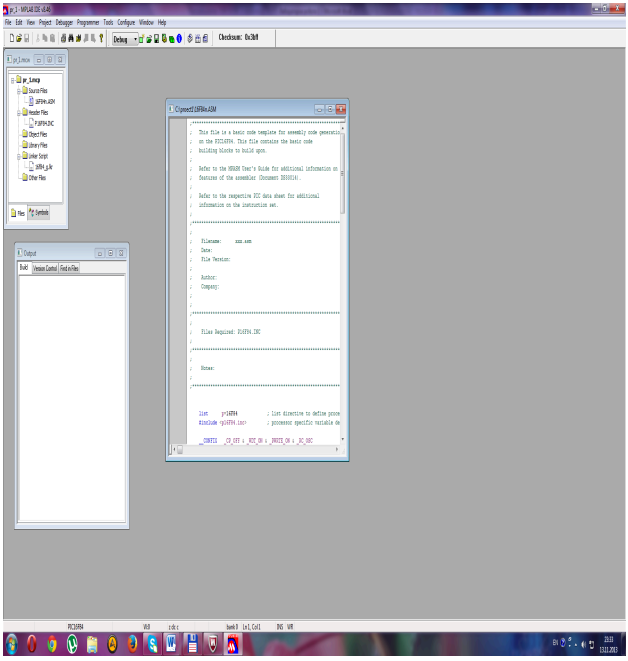

Рисунок 1.16 – Заготовка коду програми для PIC 16 F84.

Рисунок 1.17 Розміщення директиви у вікні коду програми

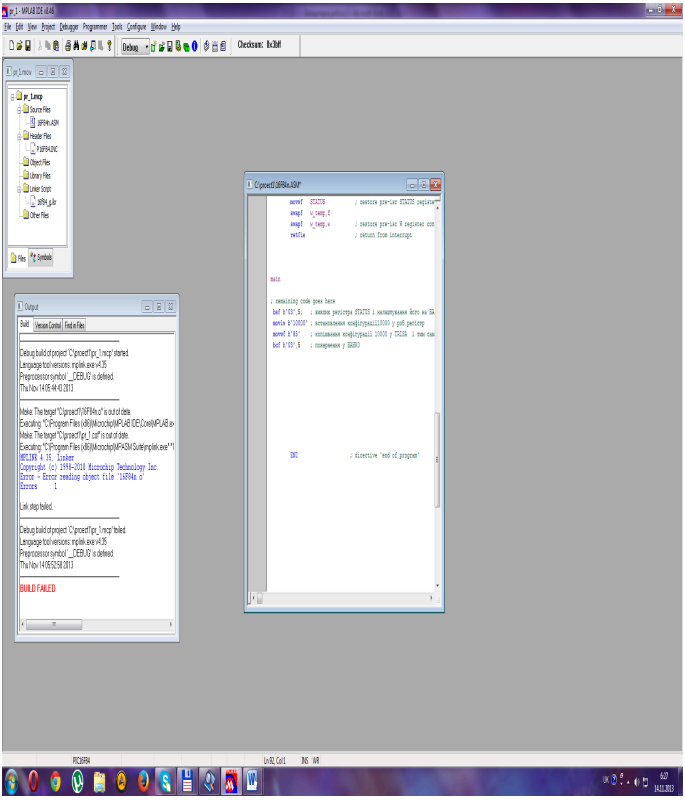

Рисунок 1.18 – Записана програма ініціалізації порту А.

1.3.7 У меню *Proect* знаходимо команду *Build All* (Ctrl+F10) що призводить до перевірки правильності запису коду і формування у вікні *Output* програми повідомлення про помилку, якщо вона знайдеться, або ж про її відсутність.

1.3.7.1 Здійснимо команду *Build All* і отримаємо повідомлення про виконання її (*Executing*). На рисунку 1.19 у вікні *Output* показане таке повідомлення.

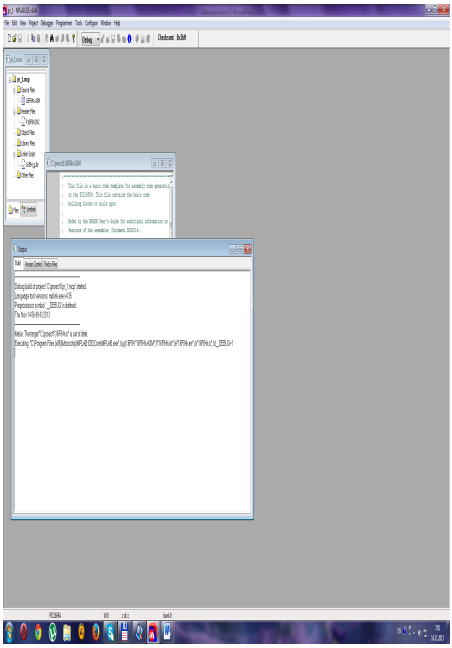

Рисунок 1.19 – Повідомлення про виконання команди Build All

1.3.7.2 Здійснимо перевірку, який вигляд матиме повідомлення у вікні *Output* в разі виявлення помилки. Для цього навмисно зробимо якусь ваду у тілі програми.. Наприклад впишемо цифру 1 попереду однієї з команд, як це показано на рисунку 1.20.

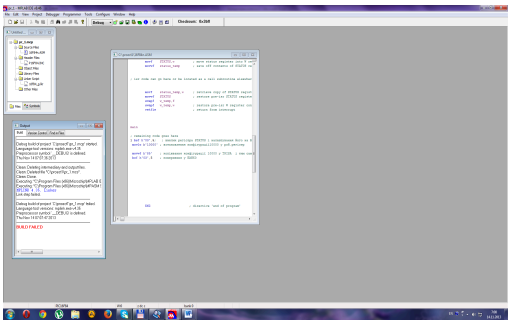

Рисунок 1.20 – Помилово записана цифра 1 перед командою bsf h'03',5

1.3.7.3 Збережемо файл з допомогою команди *File –> Save.*

1.3.7.4 Закриємо програму *MPLab IDE.*

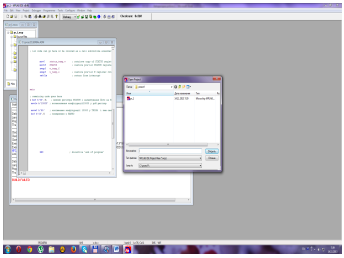

Рисунок 1.21 – Відкриття готового проекту Pr1

1.3.7.5 Здійснимо відкриття програми, як це вказано у пункті 3.3, але далі не командою *Proect–>Wizard* , а командою *Proect–>Open*. Відкриється діалогове вікно, як показано на рисунку 1.21. Вибираємо наш файл Pr1 і відкриваємо його. Бачимо у програмному коді зроблену нами помилку (див. Рисунок 1.20).

1.3.7.6 Здійснюємо команду *Build All*, у разі появи вікна попередження про недоцільність багаторазового ввімкнення симулятора клацніть по кнопці «НЕТ».

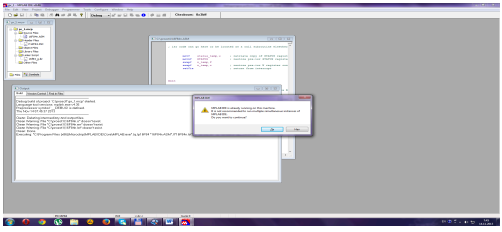

Рисунок 1.22 – Вікно попередження про недоцільність багаторазового ввімкнення симулятора

1.3.7.7 Відразу після цього у вікні Output появиться повідомлення про помилку: BUILD FAILED. Порівняйте

рисунок 1.23 з повідомленням про помилку і рисунок 1.19 про безпомилкове виконання побудови коду програми.

1.4 Індивідуальна частина роботи.

За завданням викладача здійсніть запис програми ініціалізації всіх портів мікроконтролера PIC16F84.

Варіанти завдань подано у таблиці 1.1.

Таблиця1.1 – Варіанти завдань

№ варіанту	Кофігурація портів	
	ПортА	ПортВ
1	01000	10000111
2	00001	11110000
3	11110	00001111
4	11000	11111111
5	00000	11000000
6	10111	00000000
7	11111	00000001
8	00000	11100011

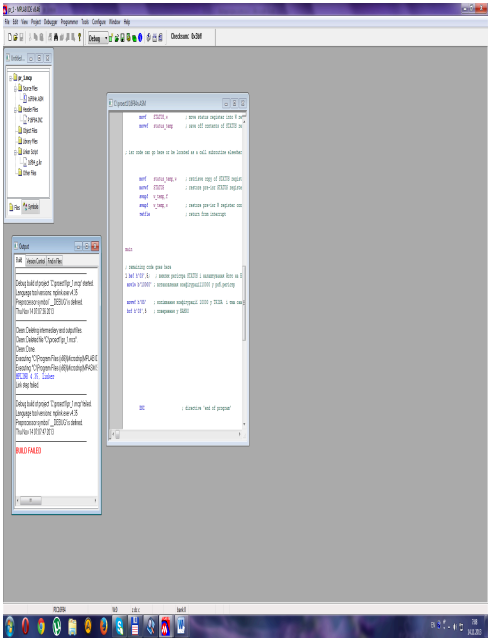

Рисунок 1.23 – Повідомлення про помилку у коді

1.5. Оформлення звіту з лабораторної роботи

1.5.1. Створену програму збережіть у індивідуальний файл з виконання лабораторних робіт.

1.5.2 Програму ініціалізації портів згідно з варіантом подайте як письмовий звіт з лабораторної роботи.

Лабораторна робота № 2

ПРЯМА І НЕПРЯМА АДРЕСАЦІЯ

2.1 Мета роботи. Створити коди програм з командами прямої і непрямої адресації і здійснити їх побудову у середовищі програмування *MPLab IDE* для мікроконтролера PIC16F84.

2.2 Теоретичні відомості. Теоретичні відомості про команди прямої і непрямої адресації подані у лекції 6 і показані на прикладах 6.1 і 6.2.

2.3 Хід роботи

2.3.1 Створіть новий проект, наприклад rp_2 і додайте до нього вихідні файли для мікроконтролера PIC16F84, як це розказано у пунктах 3.2.1-3.2.6 лабораторної роботи 1.

2.3.2 Ввімкніть запуск середовища *MPLab IDE* і здійсніть відкриття проекту rp_2 у послідовності, як це вказано у пунктах 3.3.1 – 3.3.6 лабораторної роботи 1.

2.3.3 Відкрийте темплет коду програми і знайдіть у ньому місце для запису головної програми, як це вказано у п.3.5 лабораторної роботи 1.

2.3.4 Пряма адресація. Запишемо у вікно коду програми програму очисткм регістрів засобами команд виключно прямої адресації: Наприклад, починаючи із регістра h'21 ' по регістр h'4F':

```
clrf h'21'
clrf h'22'
clrf h'23'
.........
clrf h'4F'
```

2.3.5 Непряма адресація

2.3.5.1 Розглянемо очистку регістрів засобами непрямої адресації

```
;запишемо для зручності читання програми
STATUS  equ 03
Z       equ 02
FSR     equ 04
        ; власне сама програма
CLR_PROGRAM
 movlw h'20'  ;вставити адресу h'20' у w
 movwf FSR, f ; і скопіювати його у регістр FSR
```

```
clr_LOOP
  clrf 0     ; очистка регістра на який вказує FSR
  incf FSR, f  ; інкрементувати вміст FSR
               ;перевірити чи досяг FSR кінцевої
               ;адреси масиву h'7F'
  movf FSR, w  ; копіювати FSR  у w
  addlw -h'7F' ; порівняти вміст w з числом h'7F'
  btfss STATUS,Z ; якщо ознака Z встановлена то
               ; завершити цикл і перейти до іншої
               ;секції програми
  goto clr_LOOP;інакше перейти до наст. ітерації
 . . .         ; наступна секція програми
```

2.3.5.2 Здійснимо запис цієї програми у вікно коду прграми середовища *MPLab IDE*

2.3.5.3 Запустимо команду Build All як це показано у п.3.7 лабораторної роботи 1.

2.4. Індивідуальна частина роботи.

За завданням викладача здійсніть очистку регістрів мікроконтролера PIC16F84 засобами прямої і непрямої адресації.

Варіанти завдань

№ варіанту	Регістри	
	Початковий	Кінцевий
1	H'15'	H'25'
2	H'23'	H'35'
3	H'19'	H'F1'
4	H'02'	H'25'

5	H’25’	H’C2’
6	H’29’	H’FD’
7	H’21’	H’3A’
8	H’02’	H’25’

5. Оформлення звіту з лабораторної роботи

5.1. Створену програму збережіть у індивідуальний файл з виконання лабораторних робіт.
5.2 Програму ініціалізації портів згідно з варіантом подайте як письмовий звіт з лабораторної роботи.

Лабораторна робота № 3

АРИФМЕТИЧНІ ОПЕРАЦІЇ, ПЕРЕМІЩЕННЯ ДАНИХ

1. Мета роботи. Створити коди програм з командами переміщення даних

2. Теоретичні відомості. Теоретичні відомості про команди прямої і непрямої адресації подані у лекції 7 і показані на прикладах 7.1 і у задачі 7.1..

3. Хід роботи

3.1 Створіть новий проект, наприклад гр_2 і додайте до нього вихідні файли для мікроконтролера PIC16F84, як це розказано у пунктах 3.2.1-3.2.6 лабораторної роботи 1.

3.2 Ввімкніть запуск середовища *MPLab IDE* і здійсніть відкриття проекту гр_2 у послідовності, як це вказано у пунктах 3.3.1 – 3.3.6 лабораторної роботи 1.

3.3 Відкрийте темплет коду програми і знайдіть у ньому місце для запису головної програми, як це вказано у п.3.5 лабораторної роботи 1.

3.4 Постановка задачі.

Дано: циистерна у якій може зберігатися 255 л. палива. На дні цистерни розміщений сприймач, який реагує на величину тиску стовпа рідини, а значить кількості апвлива у цистерні.

Потрібно створити підпрограму - попередження, яка б спричиняла подачу світлового сигналу при наявності 10 літрів палива у цистерні і звукового сигналу при наявності у ній 5 л. палива. Входи до світлодіода і звуковипромінювача - це сигнали високого рівня від біта 1 порту А і біта 0 порту А відповідно.

3.5 Відобразимо алгоритм задачі (рисунок 3.1)

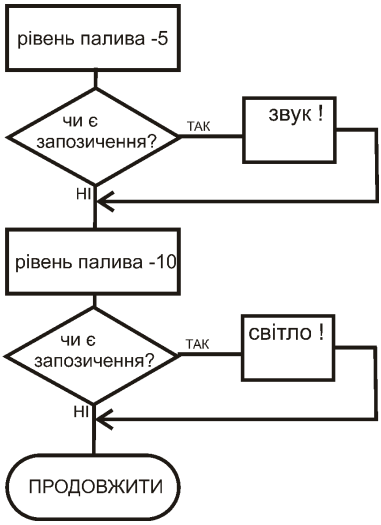

Рисунок 3.1 Алгоритм до розвязку задачі

3.6 Розглянемо алгоритм і на основі його здійснимо запис програми:

STATUS equ 3 ; регістр розміщений за адресою h'03'

Z equ 2 ; ознака нуля - це біт 2

C equ 0 ; ознака перенесення (запозичення) – біт 0

FUEL equ h'34'; регістр де зберігається величина рівня

;палива розміщений за адресою h'34'

RA1 equ 1 ; біт 1 порту А для сигналу до світлодіода

RA0 equ 0 ; біт 0 порту А для сигналу до звуку

PORTA equ 05; портА розміщений за адресою 05

; власне підпрограма

PROCEDURE

bcf PORTA,RA0 ; очистити біти RA0 і RA1 порта A

bcf PORTA,RA1

movf FUEL,w ;скопіювати вміст регістра рівня палива у

; робочий регістр

addlw -5 ; і порівняти його з числом 5

btfss STATUS,C ; перевірити чи рівень >5, якщо ні то

bsf PORTA,RA0 ; встановити біт RA0, інакше

movf FUEL,w ;скопіювати вміст регістра рівня палива у

;робочий регістр

addlw –d'10'; порівняти його з десятковим числом 10

btfss STATUS,C ; перевірити чи рівень >10, якщо ні то

bsf PORTA,RA1 ; встановити біт RA1 інакше перейти до

; на ступної секції програми

NEXT ; наступна секція програми

. . . . . .

3.6.2 Здійснимо запис цієї програми у вікно коду прграми середовища *MPLab IDE*

3.6.3 Запустимо команду Build All як це показано у п.3.7 лабораторної роботи 1.

4. Індивідуальна частина роботи.

За завданням викладача здійсніть очистку регістрів мікроконтролера PIC16F84 засобами прямої і непрямої адресації.

Варіанти завдань

№ варіанту	Рівень палива	
	початковий	Кінцевий
1	5	18
2	2	12
3	12	23
4	10	32
5	23	34
6	1	9
7	4	14
8	8	25

5. Оформлення звіту з лабораторної роботи

5.1. Створену програму збережіть у індивідуальний файл з виконання лабораторних робіт.

5.2 Програму ініціалізації портів згідно з варіантом подайте як письмовий звіт з лабораторної роботи.

Лабораторна робота № 4

ПІДПРОГРАМА ФОРМУВАННЯ ПОКАЗНИКІВ 7-СЕГМЕНТНОГО ІНДИКАТОРА

1. Мета роботи. Створити коди програм з командами переміщення даних

2. Теоретичні відомості. Теоретичні відомості про команди прямої і непрямої адресації подані у лекції 7 і показані на прикладах 7.1 і у задачі 7.1..

3. Хід роботи

3.1 Створіть новий проект, наприклад rp_2 і додайте до нього вихідні файли для мікроконтролера PIC16F84, як це розказано у пунктах 3.2.1-3.2.6 лабораторної роботи 1.

3.2 Ввімкніть запуск середовища *MPLab IDE* і здійсніть відкриття проекту rp_2 у послідовності, як це вказано у пунктах 3.3.1 – 3.3.6 лабораторної роботи 1.

3.3 Відкрийте темплет коду програми і знайдіть у ньому місце для запису головної програми, як це вказано у п.3.5 лабораторної роботи 1.

3.4 Розглянемо таблицю перетворення кодів 7-сегментної індикаці\(Таблця 1).

Таблиця 4.1 Таблиця перетворення 7-сегментного коду

PC	Table[n]	зображення
+0	**retlw b'00111111'**	
+1	**retlw b'00000110'**	

+2	**retlw b’01011011’**	
+3	**retlw b’01001111’**	
+4	**retlw b’01100110’**	
+5	**retlw b’01101101’**	
+6	**retlw b’01111101’**	
+7	**retlw b’00000111’**	
+8	**retlw b’00111111’**	
+9	**retlw b’01101111’**	

3.6 Розглянемо алгоритм і на основі його здійснимо запис програми для реалізації вказаної таблиці:

```
        ; ВИХІД : n - елемент таблиці у W
;-------------------------------------------------------------------------
-------------------------
        PCL equ h’02’
        N equ 5         ;замість «5» можна вставити будь-
                        ;яке потрібне одноцифрове число
;-ОСНОВНА ПРОГРАМА -----------------------------
        ; Код основної
        ; програми. . . . . .
        movlw N         ; завантаження числа N у W
                        ; щоб його дешифрувати для
                        ; відображення символу «N» на
                        ; 7-сегменьному індикаторі
        call SVN_SEG    ; виклик підпрограми дешифрації
        ; . . . . . .
;-ПІДПРОГРАМА ДЕШИФРАЦІЇ SVN_SEG ---------------
```

SVN_SEG

addwf PCL,f ;додавання N до PCL,отримано PC=PC+N

retlw b'00111111'; код для 0, повертається при N=0

retlw b'00000110'; код для 1, повертається при N=1

retlw b'01011011'; код для 2, повертається при N=2

retlw b'01001111'; код для 3, повертається при N=3

retlw b'01100110'; код для 4, повертається при N=4

retlw b'01101101'; код для 5, повертається при N=5

retlw b'01111101'; код для 6, повертається при N=6

retlw b'00000111'; код для 7, повертається при N=7

retlw b'00111111'; код для 8, повертається при N=8

retlw b'01101111'; код для 9, повертається при N=9

. . . . . .

3.6.2 Здійснимо запис цієї програми у вікно коду прграми середовища *MPLab IDE*

3.6.3 Запустимо команду Build All як це показано у п.3.7 лабораторної роботи 1.

4. Індивідуальна частина роботи.

За завданням викладача здійсніть очистку регістрів мікроконтролера PIC16F84 засобами прямої і непрямої адресації.

Варіанти завдань

№ варіанту	Вказані шифри	
	початковий	Кінцевий
1	7	18

2	2	23
3	12	23
4	17	32
5	23	34
6	1	9
7	4	14
8	8	25

5. Оформлення звіту з лабораторної роботи

5.1. Створену програму збережіть у індивідуальний файл з виконання лабораторних робіт.
5.2 Програму ініціалізації портів згідно з варіантом подайте як письмовий звіт з лабораторної роботи.

Лабораторна робота № 5

ОБРОБКА ПЕРЕРИВАНЬ

1. Мета роботи. Створити коди програм з командами переміщення даних

2. Теоретичні відомості. Теоретичні відомості про команди прямої і непрямої адресації подані у лекції 7 і показані на прикладах 7.1 і у задачі 7.1..

3. Хід роботи

3.1 Створіть новий проект, наприклад rp_2 і додайте до нього вихідні файли для мікроконтролера PIC16F84, як це розказано у пунктах 3.2.1-3.2.6 лабораторної роботи 1.

3.2 Ввімкніть запуск середовища *MPLab IDE* і здійсніть відкриття проекту rp_2 у послідовності, як це вказано у пунктах 3.3.1 – 3.3.6 лабораторної роботи 1.

3.3 Відкрийте темплет коду програми і знайдіть у ньому місце для запису головної програми, як це вказано у п.3.5 лабораторної роботи 1.

3.4 Розглянемо логіку системи переривань (рисунок 1)

3.5 Постановка задачі

При перетині деталлю променя на мікроконтролер PIC поступає запит у вигляді імпульса, котрий сформований фотоелементом і тригером Шмідта. Мікроконтролер, котрий у цей момент є зайнятий якоюсь фоновою програмою, повинен перйти на обробку цієї зовнішньої події EVENT. Потрібно отримати на певному регістрі величину, що відповідає кількості деталей, що перетнули лазерний пучок упродовж робочого дня, під кінець якого мікроконтролер скидається. Максимальна кількість деталей – 256.

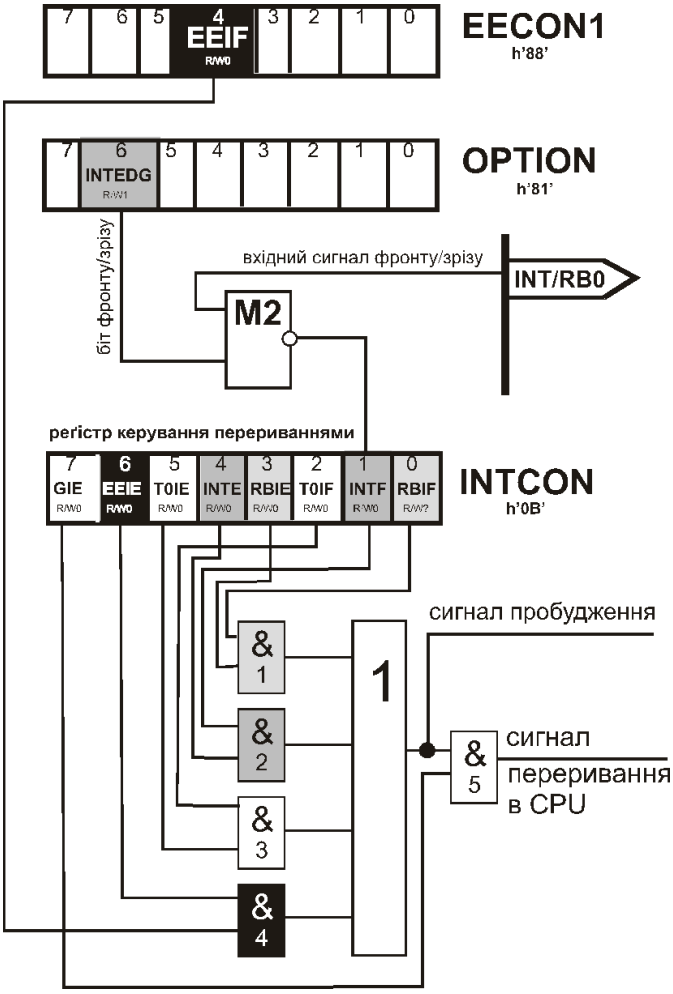

Рисунок 5.1 – Логіка системи переривань

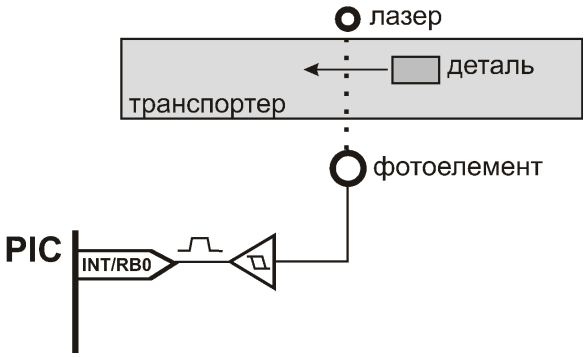

Рисунок 5.2 – Контроль проходження деталі на стрічковому транспортері

3.6 Розглянемо алгоритм і на основі його здійснимо запис програми для реалізації вказаної таблиці:

STATUS equ 03 ; регістр **STATUS**

INTCON equ h'0B'; регістр керування перериваннями

INTF equ 1 ; ознака зовнішнього переривання

INTE equ 4 ; дозвіл зовнішнього переривання

GIE equ 7 ; дозвіл глобального переривання

_status equ h'34'; комірка для тимчасового
; зберігання копії регістра STATUS

_work equ h'35' ; комірка для тимчасового
; зберігання копії регістра W

EVENT equ h'21' ; лічильник денної кількості деталей

;----ВЕКТОР СКИДАННЯ----------

```
        org 000     ;при скиданні у PC заноситься адреса h'000'
        goto MAIN  ; перехід до початку фонової програми
      ;----ВЕКТОР ПЕРЕРИВАННЯ--------------------------
        org 004; при перериванні PIC перехід на адресу h'004'
        goto INT_COUNT ; перехід до процедури обробки
                       ; переривань
      ;----ІНІЦІАЛІЗАЦІЯ У ФОНОВІЙ ПРОГРАМІ-------
      MAIN
          bsf INTCON, INTE ; дозвіл зовнішнього переривання
          bsf INTCON, GIE  ; дозвіл глобального переривання
        clrf EVENT      ;онулення лічильника кількості деталей
      ;----НЕСКІНЧЕНИЙ ЦИКЛ ФОНОВОЇ ПРОГРАМИ-----------------
      M_LOOP          ; виконання певної команди
                      ; виконання певної команди
        goto M_LOOP

      ;----ФУНКЦІЯ ОБРОБКИ ЗОВНІШНЬОЇ ;ПОДІЇ
      ;---- ДОДАЄ ДО ЛІЧИЛЬНИКА КОЖНУ ДЕТАЛЬ-
      INT_COUNT
        movwf _work     ; збереження W у памяті даних
        swapf STATUS,W  ; зчитування STATUS без зміни ознак
        movwf _status   ; і збереження його у памяті даних
      ;-----------------------------------------------------
```

bsf INTCON, INTF ; скинення ознаки переривання

inc EVENT, f ; реєстрація (додавання)деталі

;----------

swapf _status, W ; відновлення початкового стану

movwf STATUS ; регістра **STATUS**

swapf _work, f **;** відновлення початкового стану

swapf _work, W **;** регістра **W** без впливу на ознаки

;----------

3.6.2 Здійснимо запис цієї програми у вікно коду прграми середовища *MPLab IDE*

3.6.3 Запустимо команду Build All як це показано у п.3.7 лабораторної роботи 1.

4. Індивідуальна частина роботи.

За завданням викладача здійсніть очистку регістрів мікроконтролера PIC16F84 засобами прямої і непрямої адресації.

Варіанти завдань у таблиці 5.1

Таблиця 5.1 – Варіанти завжань

№ варіанту	Кількість деталей	
	мінімальна	Максимальна
1	7	1200
2	2	2300

3	12	2073
4	17	324
5	23	3400
6	10	900
7	4	144
8	8	253

5. Оформлення звіту з лабораторної роботи

5.1. Створену програму збережіть у індивідуальний файл з виконання лабораторних робіт.
5.2 Програму ініціалізації портів згідно з варіантом подайте як письмовий звіт з лабораторної роботи.

Використана література

1. Онисько О.Р. Основи програмного керування/ Посібник// О.Р.Онисько, В.Г. Панчук. Івано-Франківськ, 2014.