

# **MPLAB IDE**

Интегрированная среда разработки  
для микроконтроллеров PICmicro  
компании Microchip Technology Incorporated

Перевод основывается на технической документации DS51025D  
компании Microchip Technology Incorporated, USA.

© ООО «Микро-Чип»  
Москва - 2001

Распространяется бесплатно.  
Полное или частичное воспроизведение материала допускается только с письменного разрешения  
ООО «Микро-Чип»  
тел. (095) 737-7545  
[www.microchip.ru](http://www.microchip.ru)

# **MPLAB<sup>®</sup> IDE, SIMULATOR, EDITOR USER'S GUIDE**

Information contained in this publication regarding device applications and the like is intended by way of suggestion only. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip.

© 1999 Microchip Technology Incorporated. All rights reserved.

The Microchip logo, name, PIC, PICmicro, PICMASTER, PICSTART, and PRO MATE are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. MPLAB, and Smart Serial are trademarks of Microchip Technology in the U.S.A. and other countries.

All product/company trademarks mentioned herein are the property of their respective companies.

## Содержание

<b>ГЛАВА 1. ПРЕДВАРИТЕЛЬНАЯ ИНФОРМАЦИЯ О MPLAB IDE .....</b>	<b>8</b>
1.1 ВВЕДЕНИЕ .....	8
1.2 ОСНОВНЫЕ РАЗДЕЛЫ ГЛАВЫ .....	8
1.3 ЧТО ТАКОЕ MPLAB IDE .....	8
1.4 КАК MPLAB IDE ПОМОГАЕТ ВАМ В РАБОТЕ .....	8
1.5 MPLAB IDE – ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ .....	8
1.6 СРЕДСТВА РАЗРАБОТКИ MPLAB IDE .....	9
<b>ГЛАВА 2. УСТАНОВКА MPLAB IDE .....</b>	<b>10</b>
2.1 ВВЕДЕНИЕ .....	10
2.2 ОСНОВНЫЕ РАЗДЕЛЫ ГЛАВЫ .....	10
2.3 ТРЕБОВАНИЯ К ПЕРСОНАЛЬНОМУ КОМПЬЮТЕРУ .....	10
2.4 ИСХОДНЫЕ ФАЙЛЫ .....	10
2.5 ИНСТАЛЛЯЦИЯ MPLAB IDE .....	10
2.6 УДАЛЕНИЕ MPLAB IDE .....	12
<b>ГЛАВА 3. НАЧАЛО РАБОТЫ С MPLAB IDE (С ПРИМЕРОМ) .....</b>	<b>13</b>
3.1 ВВЕДЕНИЕ .....	13
3.2 ОСНОВНЫЕ РАЗДЕЛЫ ГЛАВЫ .....	13
3.3 НАСТРОЙКА СРЕДЫ ПРОЕКТИРОВАНИЯ .....	13
3.4 СОЗДАНИЕ НОВОГО ПРОСТОГО ПРОЕКТА .....	14
3.5 СОЗДАНИЕ НОВОГО ИСХОДНОГО ФАЙЛА .....	18
3.6 ВВОД ИСХОДНОГО ТЕКСТА ПРОГРАММЫ .....	18
3.7 КОМПИЛЯЦИЯ ИСХОДНОГО ФАЙЛА .....	19
3.8 ЗАПУСК ПРОГРАММЫ .....	20
3.9 ОТКРЫТИЕ ДОПОЛНИТЕЛЬНЫХ ОКОН .....	21
3.10 ИСПОЛЬЗОВАНИЕ ОКОН С ПЕРЕМЕННЫМИ .....	21
3.10.1 Создание окна .....	21
3.10.2 Сохранение окна .....	22
3.10.3 Редактирование окна .....	22
3.11 ТОЧКИ ОСТАНОВКИ .....	23
3.12 РЕЗЮМЕ К ГЛАВЕ .....	23

<b>ГЛАВА 4. СРЕДА ПРОЕКТИРОВАНИЯ MPLAB IDE</b> .....	<b>25</b>
4.1 ВВЕДЕНИЕ .....	25
4.2 ОСНОВНЫЕ РАЗДЕЛЫ ГЛАВЫ .....	25
4.3 КРАТКИЙ ОБЗОР ПРОЕКТОВ MPLAB IDE .....	25
4.4 КОМПИЛЯЦИЯ ПРОЕКТА С ОДНИМ ИСХОДНЫМ ФАЙЛОМ MPASM.....	27
4.4.1 <i>Настройка параметров среды проектирования</i> .....	27
4.4.2 <i>Создание нового проекта</i> .....	28
4.4.3 <i>Диалоговое окно настройки проекта</i> .....	28
4.4.4 <i>Настройка параметров компиляции</i> .....	29
4.4.5 <i>Подключение исходного файла</i> .....	30
4.4.6 <i>Компиляция исходного текста программы</i> .....	31
4.4.7 <i>Поиск и устранение ошибок</i> .....	31
4.4.8 <i>Окно проекта</i> .....	32
4.4.9 <i>Заключение</i> .....	32
4.5 КОМПИЛЯЦИЯ ОДНОГО ИСХОДНОГО ФАЙЛА MPASM БЕЗ СОЗДАНИЯ ПРОЕКТА .....	33
4.5.1 <i>Настройка параметров среды проектирования</i> .....	33
4.5.2 <i>Открытие файла исходного текста программы</i> .....	33
4.5.3 <i>Компиляция исходного текста программы</i> .....	34
4.5.4 <i>Поиск и устранение ошибок</i> .....	34
4.5.5 <i>Заключение</i> .....	35
4.6 СОЗДАНИЕ ПРОЕКТА С НЕСКОЛЬКИМИ ИСХОДНЫМИ ФАЙЛАМИ MPASM .....	36
4.6.1 <i>Настройка параметров среды проектирования</i> .....	36
4.6.2 <i>Создание нового проекта</i> .....	37
4.6.3 <i>Настройка параметров компиляции</i> .....	37
4.6.4 <i>Подключение первого исходного файла</i> .....	38
4.6.5 <i>Подключение дополнительных исходных файлов</i> .....	39
4.6.6 <i>Подключение файла сценария</i> .....	39
4.6.7 <i>Компиляция</i> .....	40
4.6.8 <i>Поиск и устранение ошибок</i> .....	41
4.6.9 <i>Окно проекта</i> .....	42
4.6.10 <i>Заключение</i> .....	42
4.7 СОЗДАНИЕ ПРОЕКТА СОВМЕСТНО С HI-TECH PIC C .....	43
4.7.1 <i>Настройка параметров среды проектирования</i> .....	43
4.7.2 <i>Подключение компилятора к среде проектирования MPLAB IDE</i> .....	44
4.7.3 <i>Создание нового проекта</i> .....	45
4.7.4 <i>Настройка параметров компиляции</i> .....	46
4.7.5 <i>Подключение исходных файлов</i> .....	46
4.7.6 <i>Компиляция</i> .....	48
4.7.7 <i>Поиск и устранение ошибок</i> .....	48
4.7.8 <i>Окно проекта</i> .....	49
4.7.9 <i>Заключение</i> .....	50
4.8 СОЗДАНИЕ ПРОЕКТА СОВМЕСТНО С MPLAB-C17 ИЛИ MPLAB-C18 .....	50
<b>ГЛАВА 5. РЕДАКТОР MPLAB</b> .....	<b>51</b>
5.1 ВВЕДЕНИЕ .....	51
5.2 ОСНОВНЫЕ РАЗДЕЛЫ ГЛАВЫ .....	51
5.3 ЧТО ТАКОЕ РЕДАКТОР MPLAB.....	51
5.4 РЕДАКТОР MPLAB ПОМОГАЕТ ВАМ .....	51
5.5 ХАРАКТЕРИСТИКИ РЕДАКТОРА MPLAB .....	52
5.5.1 <i>Размер файлов</i> .....	52
5.5.2 <i>Совместимость с операционной системой Windows</i> .....	52
5.5.3 <i>Определение функций для комбинации кнопок клавиатуры</i> .....	52
5.5.4 <i>Создание файлов</i> .....	52
5.6 ФУНКЦИИ РЕДАКТОРА MPLAB .....	52
5.6.1 <i>Работа с файлами</i> .....	52
5.6.2 <i>Работа с шаблонами</i> .....	52
5.6.3 <i>Обработка текста</i> .....	53
5.6.4 <i>Редактирование параметров окна</i> .....	53
5.6.5 <i>Поддержка языка C</i> .....	54

<b>ГЛАВА 6. ОТЛАДЧИК И СИМУЛЯТОР MPLAB IDE .....</b>	<b>55</b>
6.1 ВВЕДЕНИЕ .....	55
6.2 ОСНОВНЫЕ РАЗДЕЛЫ ГЛАВЫ .....	55
6.3 ФУНКЦИИ ОТЛАДЧИКА MPLAB IDE.....	55
6.4 ВЫПОЛНЕНИЕ ПРОГРАММЫ В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ .....	55
6.4.1 Использование симулятора MPLAB SIM .....	55
6.4.2 Режим анимации .....	56
6.5 СРЕДА СИМУЛЯТОРА MPLAB SIM.....	56
6.5.1 Симуляция портов ввода/вывода .....	56
6.5.2 Скорость выполнения .....	56
6.5.3 Стоимость .....	56
6.5.4 Инструмент отладки .....	56
6.6 ХАРАКТЕРИСТИКИ СИМУЛЯТОРА MPLAB SIM.....	57
6.7 ТОЧКИ ОСТАНОВКИ И ТРАССИРОВКИ .....	58
6.7.1 Точки останова .....	59
6.7.2 Точки трассировки .....	60
6.7.3 Настройка числа проходов для точек останова и трассировки .....	61
6.8 ТОЧКИ ОСТАНОВКИ ПО УСЛОВИЮ .....	62
6.8.1 Условия .....	62
6.8.2 Данные трассировки .....	62
6.8.3 Единственный цикл .....	62
6.8.4 Многократные циклы .....	62
6.9 ФУНКЦИИ СТИМУЛА .....	63
6.9.1 Асинхронный стимул .....	63
6.9.2 Файлы стимула порта ввода/вывода .....	64
6.9.3 Файл стимула регистра .....	66
6.9.4 Стимул тактового сигнала .....	68
6.10 СИМУЛЯЦИЯ 12-РАЗРЯДНЫХ МИКРОКОНТРОЛЛЕРОВ .....	69
6.10.1 12-разрядные микроконтроллеры .....	69
6.10.2 Порты ввода/вывода .....	69
6.10.3 Модель микроконтроллера .....	69
6.10.4 Периферия .....	69
6.11 СИМУЛЯЦИЯ 14-РАЗРЯДНЫХ МИКРОКОНТРОЛЛЕРОВ .....	70
6.11.1 14-разрядные микроконтроллеры .....	70
6.11.2 Порты ввода/вывода .....	70
6.11.3 Прерывания .....	70
6.11.4 Модель микроконтроллера .....	71
6.11.5 Специальные регистры .....	71
6.11.6 Периферия .....	71
6.12 СИМУЛЯЦИЯ 16-РАЗРЯДНЫХ МИКРОКОНТРОЛЛЕРОВ PIC17CXXX .....	73
6.12.1 16-разрядные микроконтроллеры семейства PIC17CXXX .....	73
6.12.2 Порты ввода/вывода .....	73
6.12.3 Прерывания .....	73
6.12.4 Модель микроконтроллера .....	73
6.12.5 Специальные регистры .....	74
6.12.6 Периферия .....	74
6.12.7 Реализация памяти .....	74
6.13 СИМУЛЯЦИЯ 16-РАЗРЯДНЫХ МИКРОКОНТРОЛЛЕРОВ PIC18CXXX .....	75
6.13.1 16-разрядные микроконтроллеры семейства PIC18CXXX .....	75
6.13.2 Порты ввода/вывода .....	75
6.13.3 Прерывания .....	75
6.13.4 Модель микроконтроллера .....	75
6.13.5 Специальные регистры .....	76
6.13.6 Периферия .....	76

<b>ГЛАВА 7. ПАНЕЛИ ИНСТРУМЕНТОВ И МЕНЮ MPLAB IDE</b> .....	<b>77</b>
7.1 ВВЕДЕНИЕ .....	77
7.2 ОСНОВНЫЕ РАЗДЕЛЫ ГЛАВЫ .....	77
7.3 РАБОЧИЙ СТОЛ MPLAB IDE .....	77
7.3.1 <i>Графическое меню</i> .....	78
7.3.2 <i>Линейка состояния</i> .....	78
7.3.3 <i>Системное меню</i> .....	78
7.4 МЕНЮ FILE .....	79
7.4.1 <i>Новый файл (File &gt; New)</i> .....	79
7.4.2 <i>Открыть сохраненный файл на диске (File &gt; Open)</i> .....	79
7.4.3 <i>Открыть файл для просмотра (File &gt; View)</i> .....	79
7.4.4 <i>Сохранение файлов (File &gt; Save, File &gt; Save As, File &gt; Save All)</i> .....	80
7.4.5 <i>Закрыть открытые файлы (File &gt; Close, File &gt; Close All)</i> .....	80
7.4.6 <i>Импорт (File &gt; Import)</i> .....	81
7.4.7 <i>Экспорт файлов (File &gt; Export)</i> .....	83
7.4.8 <i>Печать (Ctrl + P)</i> .....	86
7.4.9 <i>Настройка параметров печати (File &gt; Print Setup)</i> .....	87
7.4.10 <i>Выход (File &gt; Exit)</i> .....	87
7.4.11 <i>Список ранее используемых файлов</i> .....	87
7.5 МЕНЮ ПРОЕКТА (PROJECT) .....	88
7.6 МЕНЮ РЕДАКТОРА MPLAB (EDIT) .....	89
7.6.1 <i>Команды меню редактора</i> .....	89
7.6.2 <i>Работа с шаблонами</i> .....	91
7.6.3 <i>Обработка текста</i> .....	95
7.7 МЕНЮ ОТЛАДКИ (DEBUG) .....	96
7.7.1 <i>Подменю Run</i> .....	96
7.7.2 <i>Подменю Execute</i> .....	97
7.7.3 <i>Подменю Simulator Stimulus</i> .....	99
7.7.4 <i>Указатель выполнения программы</i> .....	99
7.7.5 <i>Настройка точек остановки (Debug &gt; Break Settings)</i> .....	100
7.7.6 <i>Настройка точек трассировки (Debug &gt; Trace Settings)</i> .....	102
7.7.7 <i>Debug &gt; Trigger In/Out Settings – настройка триггера ввода/вывода</i> .....	106
7.7.8 <i>Debug &gt; Trigger Output Points – точка триггера</i> .....	106
7.7.9 <i>Debug &gt; Clear All Points – удаление всех точек остановки и трассировки</i> .....	106
7.7.10 <i>Debug &gt; Complex Trigger Settings – настройка сложного триггера</i> .....	107
7.7.11 <i>Debug &gt; Code Coverage – захват кода</i> .....	107
7.7.12 <i>Debug &gt; Clear Program Memory – очистка памяти программ</i> .....	107
7.7.13 <i>Debug &gt; System Reset (Ctrl + Shift + F3) – системный сброс</i> .....	108
7.7.14 <i>Debug &gt; Power-On-Reset (Ctrl + Shift + F5) – сброс по включению питания</i> .....	108
7.8 МЕНЮ ПРОГРАММАТОРОВ (PICSTART PLUS/PRO MATE).....	109
7.9 МЕНЮ НАСТРОЕК (OPTIONS) .....	110
7.9.1 <i>Options &gt; Development Mode – настройка параметров работы микроконтроллера</i> .....	110
7.9.2 <i>Подменю Window Setup</i> .....	117
7.9.3 <i>Настройка текстового редактора MPLAB IDE</i> .....	118
7.9.4 <i>Сброс настроек редактора MPLAB IDE</i> .....	118
7.9.5 <i>Настройка среды проектирования (Environment Setup)</i> .....	119
7.9.6 <i>Настройка программаторов (Options &gt; Programmer Options)</i> .....	129
7.10 МЕНЮ TOOLS.....	130
7.10.1 <i>Выполнение DOS программы</i> .....	130
7.10.2 <i>Повторное выполнение DOS программы</i> .....	130
7.10.3 <i>Проверка эмулятора PICMASTER</i> .....	130
7.10.4 <i>Проверка эмулятора MPLAB-ICE</i> .....	130

7.11	МЕНЮ WINDOW .....	131
7.11.1	Window > Program Memory – открыть окно памяти программ.....	131
7.11.2	Window > Trace Memory – открыть окно буфера трассировки.....	133
7.11.3	Window > EEPROM Memory – открыть окно EEPROM памяти.....	133
7.11.4	Window > Calibration Data. – открыть окно данных калибровки.....	134
7.11.5	Window > Absolute Listing – открыть окно листинга программы *.LST файл.....	135
7.11.6	Window > Map File. – открыть окно карты памяти.....	135
7.11.7	Window > Stack – открыть окно аппаратного стека.....	136
7.11.8	Window > File Registers – открыть окно памяти данных.....	137
7.11.9	Window > Special Function Registers – открыть окно регистров специального назначения.....	139
7.11.10	Window > Show Symbol List – открыть окно списка переменных и меток.....	140
7.11.11	Window > Stopwatch – открыть окно секундомера.....	141
7.11.12	Window > Project – открыть окно проекта.....	141
7.11.13	Окна с переменными (Window > Watch Window).....	142
7.11.14	Открыть окно изменений.....	145
7.11.15	Разместить окна горизонтально.....	146
7.11.16	Разместить окна вертикально.....	146
7.11.17	Разместить окна каскадом.....	146
7.11.18	Свернуть все окна.....	146
7.11.19	Упорядочить заголовки свернутых окон.....	146
7.12	МЕНЮ HELP.....	147
7.12.1	Release Notes (Shift+F1).....	147
7.12.2	Tool Release Notes.....	147
7.12.3	MPLAB IDE Help.....	147
7.12.4	Editor Help.....	147
7.12.5	Error Help.....	147
7.12.6	MPASM Help.....	147
7.12.7	MPLINK Help.....	147
7.12.8	About.....	148
<b>ПРИЛОЖЕНИЯ.....</b>		<b>149</b>
A.	ОПИСАНИЕ КНОПОК ГРАФИЧЕСКОГО МЕНЮ MPLAB IDE.....	149
A.1	Меню редактора MPLAB IDE.....	149
A.2	Меню отладчика.....	149
A.3	Меню проекта.....	150
A.4	Меню пользователя.....	150
B.	НАЗНАЧЕНИЕ ПОЛЕЙ ЛИНЕЙКИ СОСТОЯНИЯ.....	151
C.	НАЗНАЧЕНИЕ ФАЙЛОВ СРЕДЫ MPLAB IDE.....	152
D.	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КНОПОК КЛАВИАТУРЫ.....	153
D.1	Функциональные кнопки.....	153
D.2	Кнопки управления перемещением курсора.....	154
D.3	Управляющие кнопки.....	155
D.4	Кнопки форматирования и редактирования.....	155

# Глава 1. Предварительная информация о MPLAB IDE

## 1.1 Введение

В этой главе дана краткая характеристика интегрированной среды проектирования MPLAB IDE.

## 1.2 Основные разделы главы

Что такое MPLAB IDE  
Как MPLAB IDE помогает Вам в работе  
MPLAB IDE – интегрированная среда разработки  
Инструментальные средства MPLAB IDE

## 1.3 Что такое MPLAB IDE

MPLAB IDE – бесплатная интегрированная среда разработки для микроконтроллеров PICmicro фирмы Microchip Technology Incorporated. MPLAB IDE позволяет писать, отлаживать и оптимизировать текст программы. MPLAB IDE включает в себя редактор текста, симулятор и менеджер проектов, поддерживает работу эмуляторов (MPLAB-ICE, PICMASTER) и программаторов (PICSTART plus, PRO MATE) фирмы Microchip и других отладочных средств фирмы Microchip и третьих производителей.

## 1.4 Как MPLAB IDE помогает Вам в работе

Легко настраиваемые инструментальные средства, тематическая помощь, «выпадающие» меню и «назначение горячих» клавиш в MPLAB IDE позволяют Вам:

- получить код программы;
- наблюдать выполнение программы с помощью симулятора, или в реальном времени, используя эмулятор (требуется аппаратная часть);
- определять время выполнения программы;
- просматривать текущее значение переменных и специальных регистров;
- работать с программаторами PICSTAR и PRO MATE II;
- использовать систему помощи по MPLAB IDE.

## 1.5 MPLAB IDE – интегрированная среда разработки

MPLAB IDE – легкая в освоении и использовании интегрированная среда разработки, работающая под управлением операционных систем Microsoft Windows 3.1x, 95/98, 2000, NT.

**Примечание.** Аппаратные модули MPLAB IDE (программаторы и эмуляторы) могут не работать в некоторых операционных системах. Для детальной информации обратитесь к технической документации на конкретное устройство.

MPLAB IDE поддерживает следующие функции:

- создание и редактирование исходных текстов программы;
- объединение файлов в проект;
- отладка кода программы;
- отладка кода программы с использованием симулятора или эмулятора (требуется аппаратная часть).

MPLAB IDE позволяет Вам создавать исходный текст программы в полнофункциональном текстовом редакторе, легко выполнить исправление ошибок при помощи окна результатов компиляции, в котором указываются возникшие ошибки и предупреждения.

Используя менеджер проектов можно указать исходные файлы программы, объектные файлы, библиотеки и файлы сценария.

MPLAB IDE обеспечивает разнообразные средства симуляции и эмуляции исполняемого кода для выявления логических ошибок. Вот их основные особенности:

- большое количество сервисных окон, чтобы контролировать значения регистров памяти данных и выполнение инструкций микроконтроллера;
- окна исходного кода программы, листинга программы, кода программы - позволяют оценить качество компиляции;
- пошаговое выполнение программы, система точек остановки, трассировки, сложных условий предназначена для быстрой и удобной отладки вашей программы.

## 1.6 Средства разработки MPLAB IDE

MPLAB IDE состоит из нескольких модулей, обеспечивающих единую среду разработки.

### *Менеджер проекта MPLAB*

Используется для создания и работы с файлами, относящимися к проекту. Позволяет одним щелчком «мыши» выполнить компиляцию исходного текста, включить симулятор или внутрисхемный эмулятор и т.д.

### *Редактор MPLAB*

Предназначен для написания и редактирования исходного текста программы, шаблонов и файлов сценария линкера.

### *Отладчик MPLAB ICD*

Недорогой внутрисхемный отладчик для микроконтроллеров семейства PIC16F87X.

### *MPLAB-SIM симулятор*

Программный симулятор моделирует выполнение программы в микроконтроллере с учетом состояния портов ввода/вывода.

### *MPLAB ICE эмулятор*

Эмулирует работу микроконтроллера в масштабе реального времени непосредственно в устройстве пользователя.

### *MPASM ассемблер/ MPLINK линкер/ MPLIB редактор библиотек*

MPASM компилирует исходный текст программы. MPLINK создает заключительный код программы, связывая различные модули полученные из MPASM, MPLAB-C17, MPLAB-C18. MPLIB управляет библиотеками.

### *MPLAB-CXX компиляторы*

MPLAB-C17 и MPLAB-C18 выполняют компиляцию текста программы написанному на языке ANSI C. Сложные проекты могут состоять и частей написанных на языке C и ассемблера.

### *Программаторы PRO MATE и PICSTART plus*

Работают под управлением MPLAB IDE и предназначены для программирования микроконтроллеров кодом программы, полученной в результате компиляции исходных файлов. Программатор PRO MATE может работать самостоятельно, без использования MPLAB IDE.

### *Эмуляторы MPLAB-ICE, PICMASTER-CE и PICMASTER*

Применяются для моделирования работы микроконтроллера в устройстве пользователя в масштабе реального времени.

**Примечание.** Описание, подключения инструментальных модулей к MPLAB IDE другими производителями, смотрите в технической документации DS00104.

## Глава 2. Установка MPLAB IDE

### 2.1 Введение

В этой главе будут рассмотрены вопросы установки MPLAB IDE на Ваш компьютер.

### 2.2 Основные разделы главы

Требования к персональному компьютеру  
Исходные файлы  
Инсталляция MPLAB IDE  
Удаление MPLAB IDE

### 2.3 Требования к персональному компьютеру

Минимальная конфигурация компьютера, требуемая для установки MPLAB:

- PC совместимый компьютер с Pentium архитектурой;
- установленная операционная система Microsoft Windows 3.1x, 95/98, 2000 или NT;
- ОЗУ 16Мб (рекомендуется 32Мб);
- 45Мб свободного дискового пространства.

**Примечание.** Аппаратные модули MPLAB IDE (программаторы и эмуляторы) могут не работать в некоторых операционных системах. Для детальной информации обратитесь к технической документации на конкретное устройство.

### 2.4 Исходные файлы

MPLAB IDE поставляется в комплекте со всеми отладочными средствами Microchip.

Также MPLAB IDE можно найти на Technical Library CD-ROM, который можно получить у дистрибьюторов Microchip, либо загрузить установочные файлы из Интернет с серверов технической поддержки [www.microchip.ru](http://www.microchip.ru) или [www.microchip.com](http://www.microchip.com).

Например, для версии MPLAB 4.00 необходимы следующие файлы (номер в названии файла указывает версию программы):

- MP40000.EXE
- MP40000.W02
- MP40000.W03
- MP40000.W04
- MP40000.W05
- MP40000.W06

### 2.5 Инсталляция MPLAB IDE

Для начала установки необходимо запустить файл MPvvvvv.EXE (где vvvvv является номером версии MPLAB IDE).

Рекомендуемый порядок установки MPLAB IDE:

1. Загрузите операционную систему Microsoft Windows.
2. Если исходные файлы находятся на CD ROM Microchip, установите его.
3. Запустите файл X:\MPvvvvv.exe, где X – диск, на котором находятся исходные файлы MPLAB IDE; vvvvv – номер устанавливаемой версии. Например, d:\MP41219.exe.

**Примечание.** Пользователи Windows NT должны иметь доступ администратора для установки MPLAB-ICE.

- Следуйте указаниям, которые появляются на экране монитора во время установки MPLAB IDE. Если Вы не уверены, какое значение параметра выбрать, принимайте значение по умолчанию.

*Выбор устанавливаемых компонентов*

При ограниченном объеме свободного места на жестком диске или при отсутствии программатора, эмулятора, отладчика Microchip – Вы можете установить только часть программных инструментальных средств:

- файлы MPLAB IDE;
- файлы MPASM/ MPLINK/ MPLIB;
- файлы MPLAB SIM;
- файлы помощи.

Повторная установка позволяет дополнить среду проектирования MPLAB IDE требуемыми компонентами.

*Выбор рабочей директории*

Рекомендуется устанавливать MPLAB IDE на жестком диске компьютера, а не на сетевом диске.

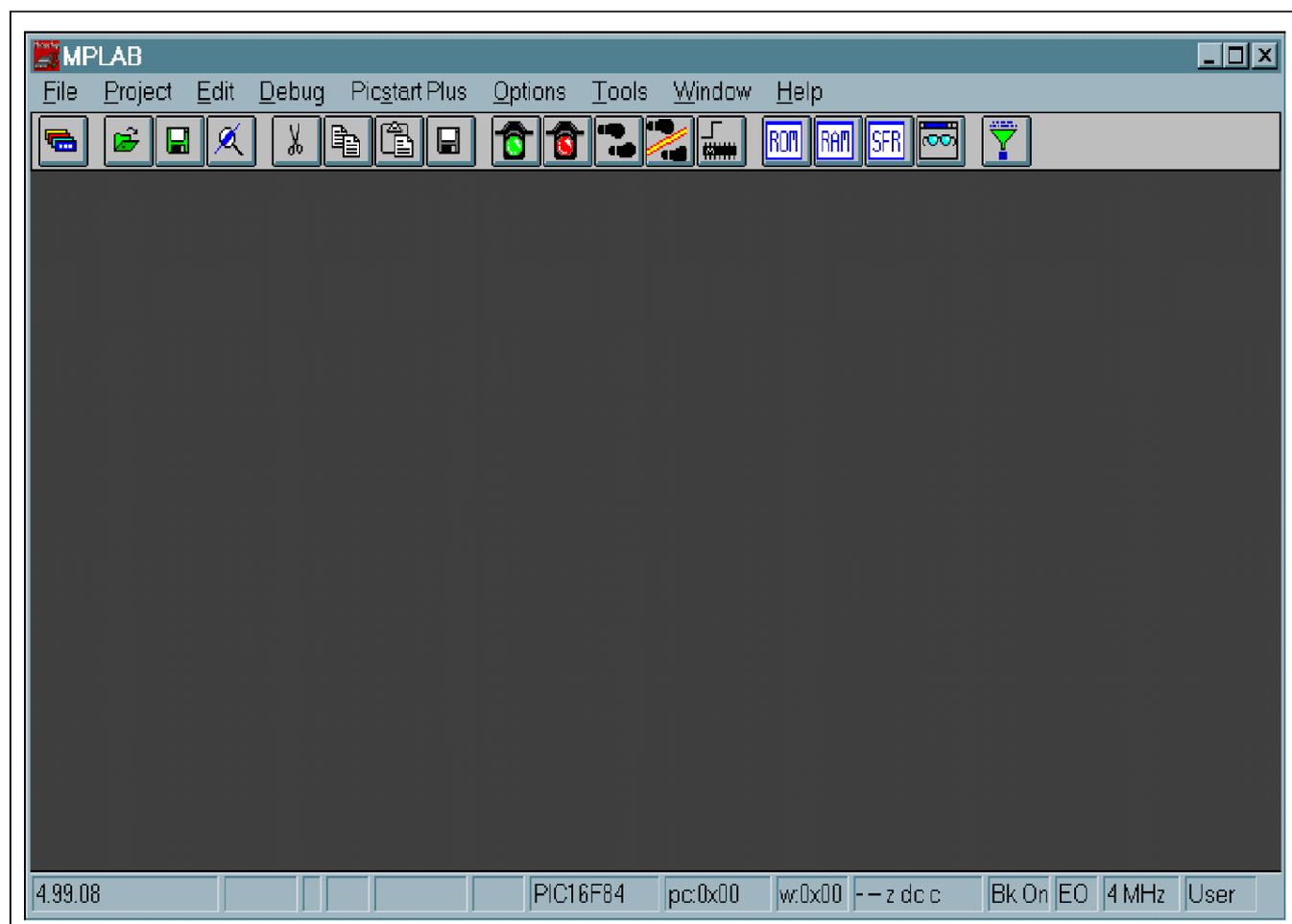
*Выбор директории размещения системных файлов*

Рекомендуется устанавливать системные файлы (DLLs) в директорию \Windows\System, для их автоматической замены при установке следующих версий MPLAB IDE.

- В то время, пока происходит копирование файлов MPLAB IDE на ваш компьютер, посмотрите дополнительную информацию, касающуюся интегрированной среды разработки MPLAB IDE.
- Внимательно просмотрите файлы Readme, в которых указаны особенности новой версии MPLAB IDE, возможные ограничения в работе и описано решение большинства проблем, связанных с установкой MPLAB IDE.

**Примечание.** Если Вы откажетесь от просмотра файлов Readme, их можно прочитать позже. Рекомендуется перед обращением в сервисную службу с проблемой установки MPLAB IDE внимательно изучить файлы Readme.

- После запуска программы (файл MPLAB.EXE) Вы увидите рабочий стол среды, показанный на рисунке.



## **2.6 Удаление MPLAB IDE**

Для удаления, запустите файл `unwise.exe` в рабочей директории MPLAB IDE. Программа автоматически удалит все рабочие и системные файлы, относящиеся к MPLAB IDE.

## Глава 3. Начало работы с MPLAB IDE (с примером)

### 3.1 Введение

Целью данной главы является познакомить Вас с основными принципами работы среды проектирования MPLAB IDE. Предполагается, что Вы потратите от 1 до 2 часов на изучения материала этой главы.

### 3.2 Основные разделы главы

Настройка среды проектирования  
Создание нового простого проекта  
Создание нового исходного файла  
Ввод текста программы  
Компиляция исходного файла  
Выполнение программы  
Открытие дополнительных окон  
Создание окон с переменными  
Сохранение окон с переменными  
Установка точек остановки

Кроме того, кратко рассмотрены другие вопросы, детальное описание которых будет дано в следующих главах. Изучая части данной главы, Вы познакомитесь с:

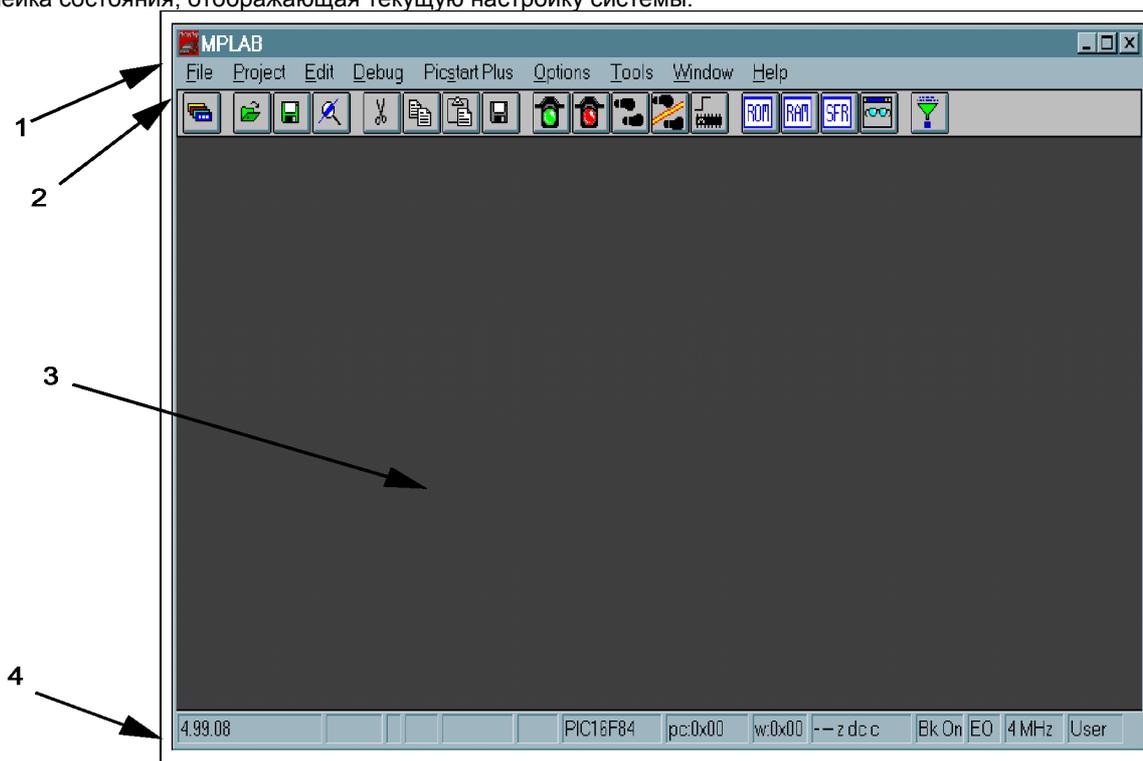
- элементами рабочего стола MPLAB IDE;
- созданием исходного файла и подключением его к новому проекту для PIC16F84;
- идентификацией и исправлением простых ошибок;
- управлением симулятором;
- установкой точек остановки;
- созданием окна с переменными;
- открытием окна секундомера и других окон отладки.

### 3.3 Настройка среды проектирования

В предыдущей главе рассматривалось как установить MPLAB IDE. Теперь рассмотрим работу в среде на основе конкретного примера.

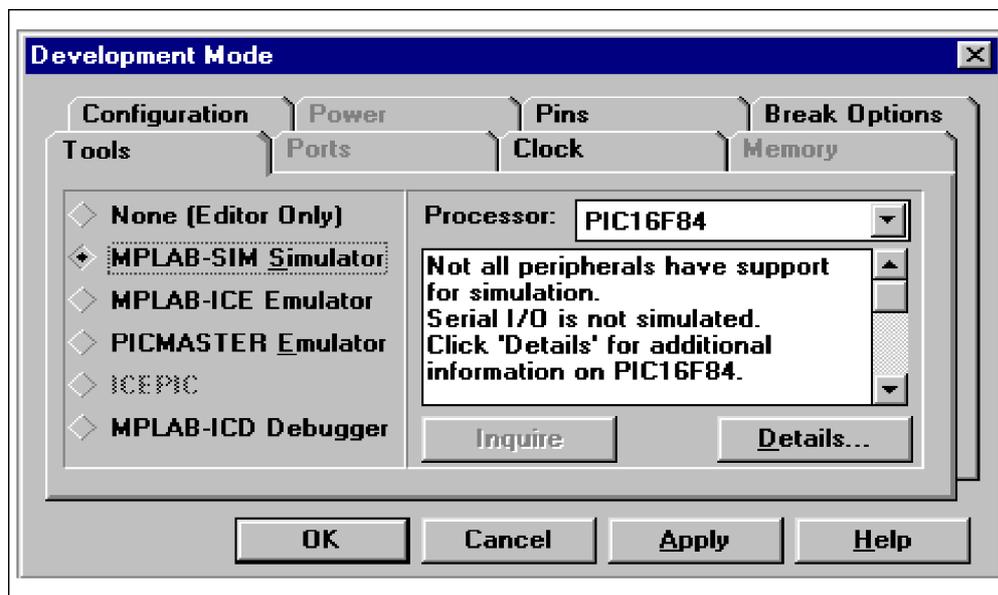
Рабочий стол среды состоит из:

1. Главное текстовое меню.
2. Графическое меню.
3. Рабочая область, в которой размещаются открытые окна с файлами, диалогами или другой информацией.
4. Линейка состояния, отображающая текущую настройку системы.



Пояснения работы интегрированной среды разработки будут производиться с использованием симулятора MPLAB SIM. Работа с эмулятором аналогично симулятору, за исключением открытия файлов кода только на чтение.

Выберите пункт *Options > Development Mode* нажмите кнопку *Tools*, для выбора инструментального средства и типа микроконтроллера, используемого в проекте.



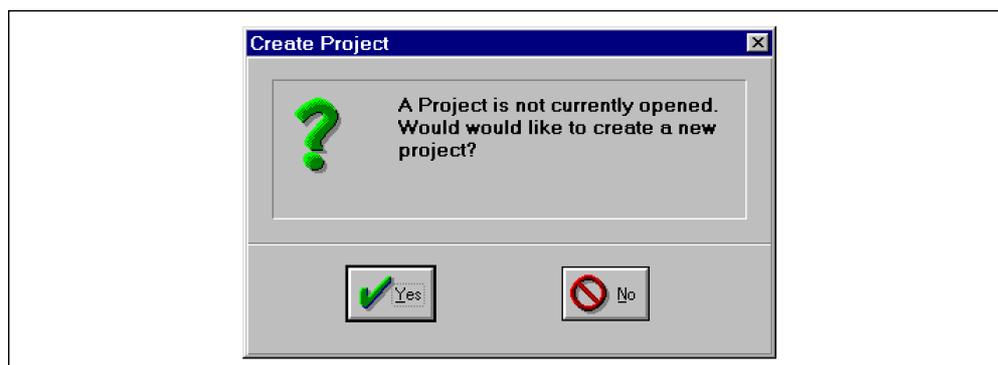
MPLAB IDE является постоянно развивающейся средой разработки, поэтому могут быть небольшие различия между меню у Вас на экране монитора и в данном руководстве.

Выберите симулятор MPLAB SIM и микроконтроллер PIC16F84, подтвердите выбор нажатием кнопки *Ok*. Симулятор инициализирован, в линейке состояния на рабочем столе появится микроконтроллер «PIC16F84» и режим «SIM». Теперь среда проектирования находится в режиме симулятора для микроконтроллера PIC16F84.

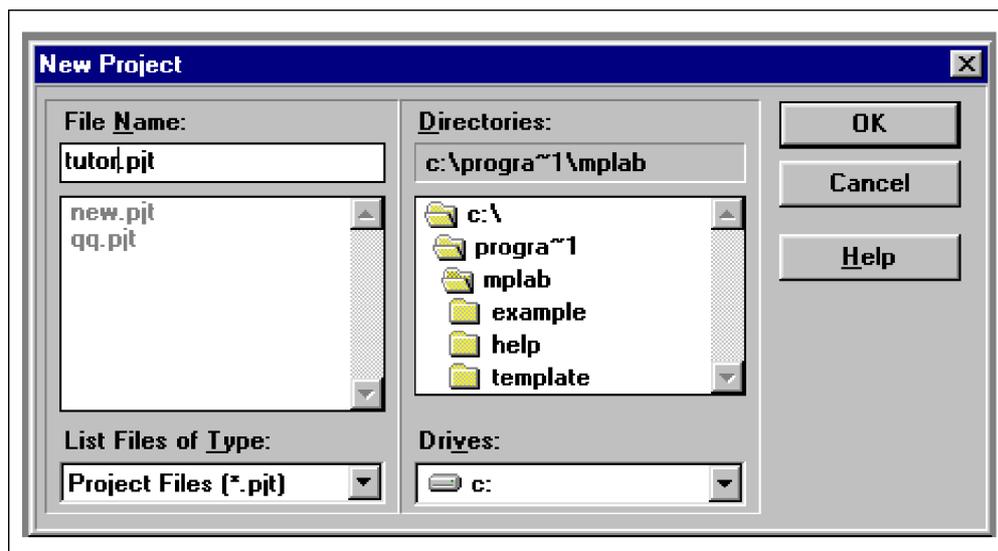
### 3.4 Создание нового простого проекта

Для работы симулятора MPLAB SIM нужен код программы (файл с расширением .HEX), который получается компиляцией исходного текста программы. В данном примере файл называется *tutor84.hex*, позже он может быть загружен непосредственно в микроконтроллер с помощью программатора.

Выберите пункт меню *File > New* и на экране появится диалоговое, окно показанное на рисунке.



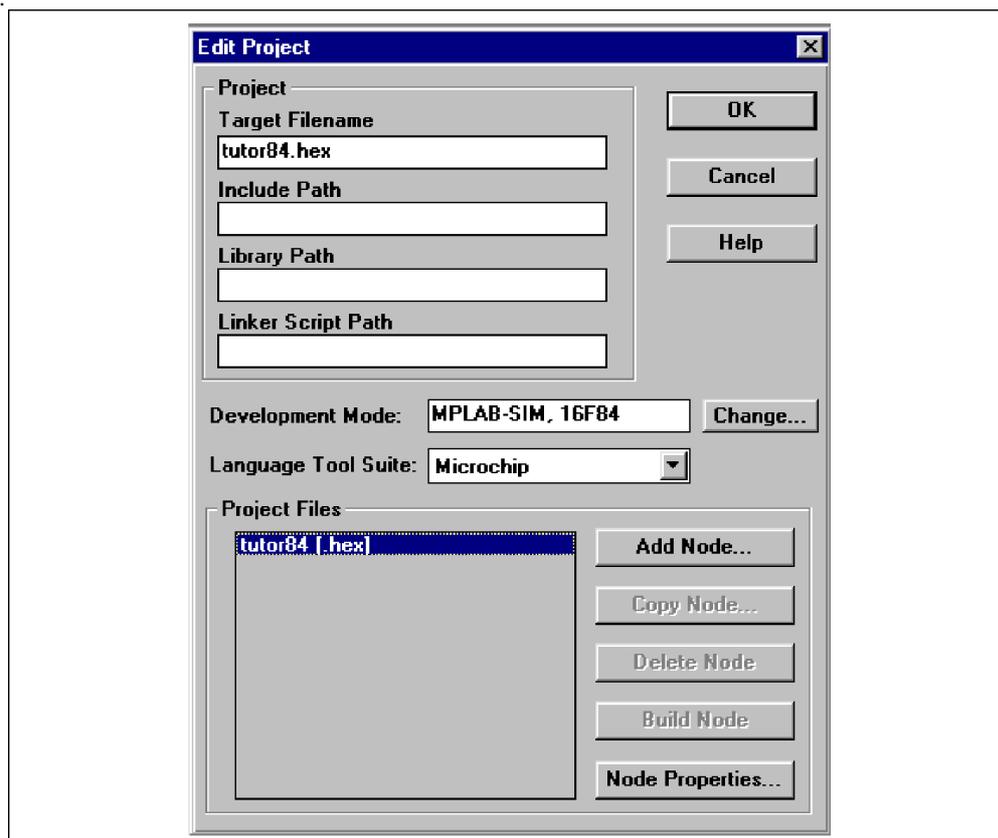
Нажмите кнопку *Yes* и увидите следующее диалоговое окно.



В этом диалоговом окне необходимо указать, где вы хотите сохранить проект. В данном случае создается файл tutor84.pjt в каталоге c:\Program Files \MPLAB.

Тип файла .PJT будет назначен автоматически. Файлы с таким расширением являются файлами проекта в среде MPLAB IDE. Имя проекта, в данном случае tutor84, станет заданным по умолчанию для многих файлов, используемых в нашем примере.

Подтверждение имени файла проекта и место его размещения приведет к переходу к следующему диалоговому окну.



Симулятор, программаторы и эмулятор среды MPLAB IDE используют файлы кода, части которого созданы различными инструментальными средствами: ассемблером, компилятором и/или линкером. Несколько различных инструментальных средств могут участвовать в создании шестнадцатеричного кода. Эти инструментальные средства являются частью каждого проекта. Создание проекта позволяет Вам определить, какие инструментальные средства будут участвовать в создании .HEX файла кода.

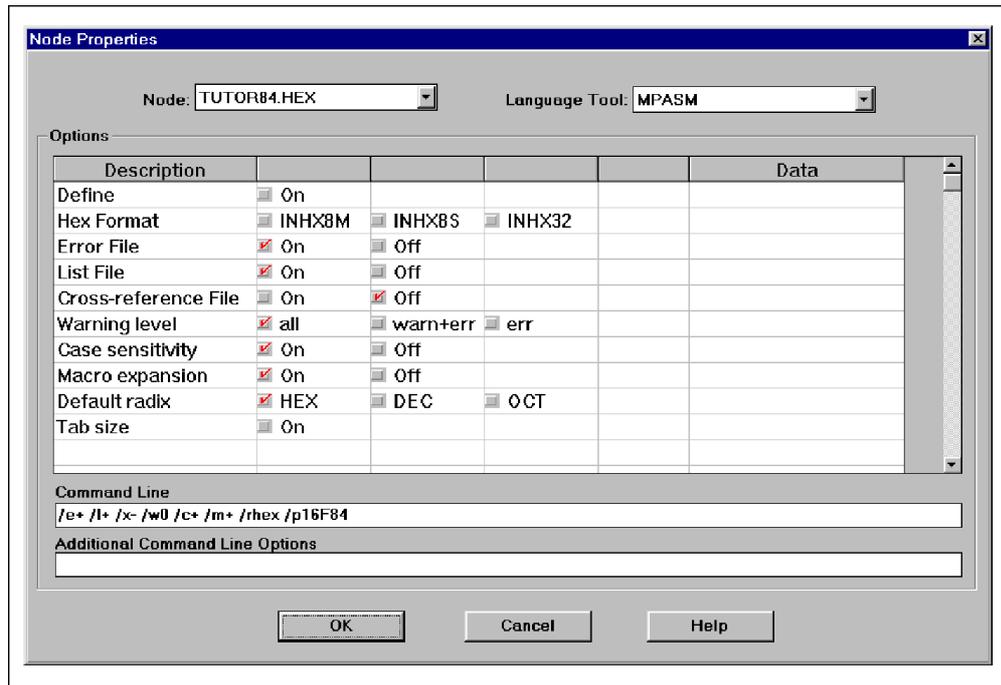
Подробное описание создания проектов с несколькими исходными файлами и правила использования других компиляторов смотрите в главе 4.

Заметьте, что имена файлов в диалоговом окне указаны правильно. Первоначальная настройка проекта соответствует предварительно установленным значениям по умолчанию.

Параметры проекта по умолчанию можно настроить в меню *Options > Environment Setup* раздел *Projects*, которые используются только при создании нового проекта.

В списках файлов проекта есть файл tutor84.hex. Выбрав этот файл (нажав на него левой кнопкой «мышки») кнопка *Node Properties* потемнеет, указывая возможность ее нажатия.

Прежде чем выполнять какие-либо действия необходимо указать правила создания шестнадцатеричного файла. Нажав на кнопку *Node Properties*, появится диалоговое окно настройки параметров компиляции, показанное на рисунке.



Это диалоговое окно содержит параметры настройки, указанные по умолчанию, для инструментального средства, показанного в правом верхнем углу (в данном случае для MPASM). В самом простом варианте, проект содержит один исходный файл и один шестнадцатеричный .HEX файл.

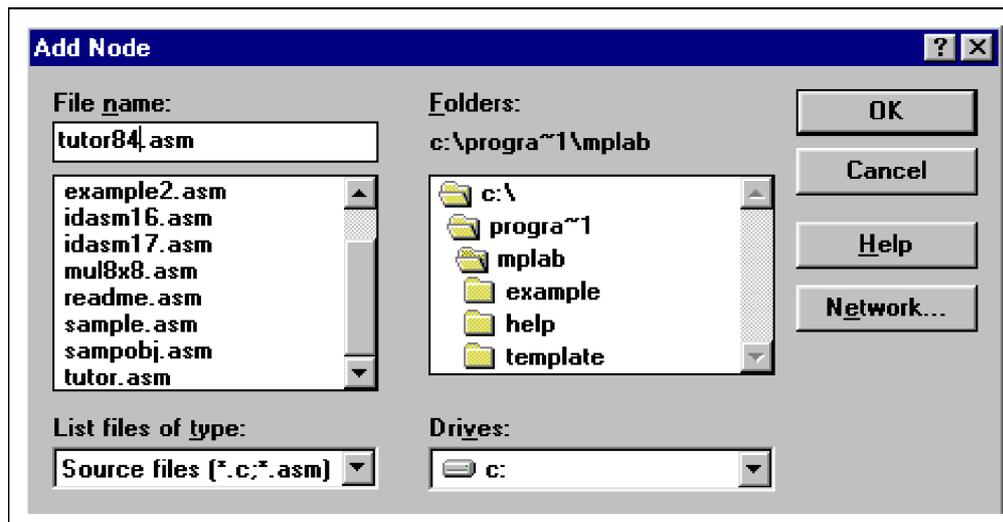
Вы видите, что диалоговое окно содержит несколько строк и столбцов. Как правило, каждая строка соответствует параметру, указываемому в командной строке при вызове инструментального средства. Установка параметров отображается в командной строке (Command Line), которая будет использоваться при вызове MPASM средой проектирования MPLAB IDE.

Пока Вы можете использовать настройки по умолчанию. Со временем, изучив среду проектирования, Вы сможете правильно изменять эти параметры.

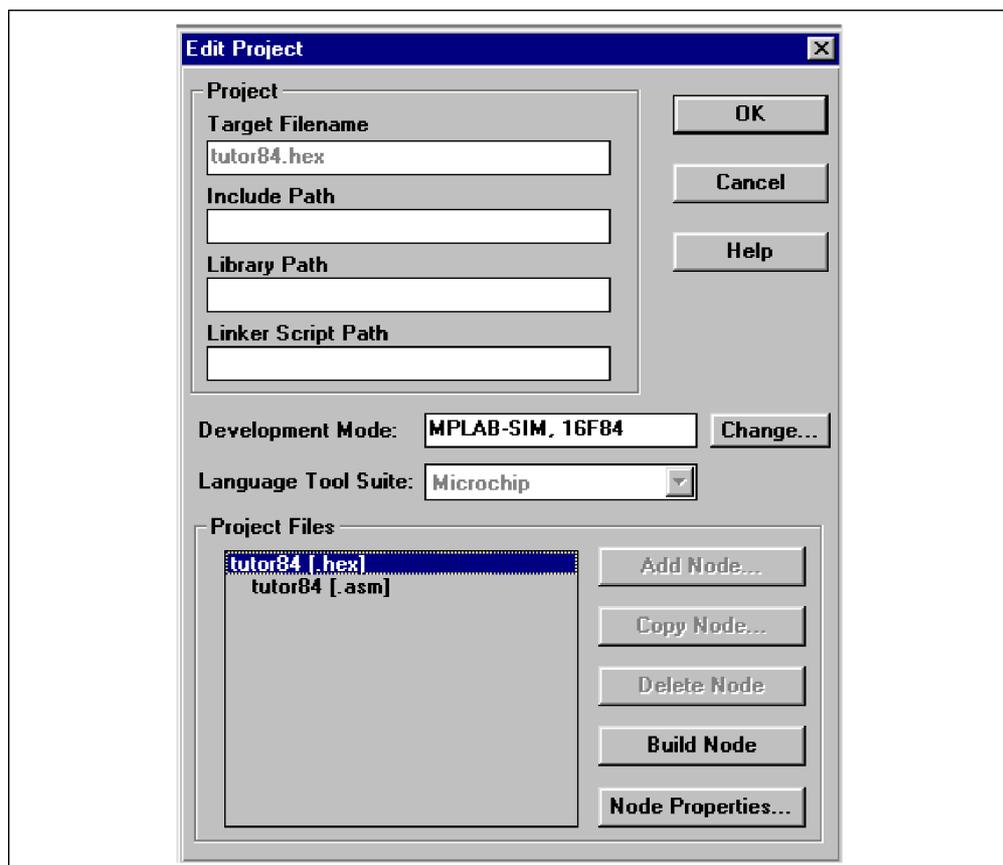
Нажмите кнопку *OK* для сохранения параметров компиляции и возврату к диалоговому окну настройки проекта.



Нажмите кнопку *Add Node*. Вы увидите стандартный диалог выбора файлов (см. рисунок) с открытой рабочей директорией проекта. Введите имя файла `tutor84.asm` и нажмите кнопку *OK*.



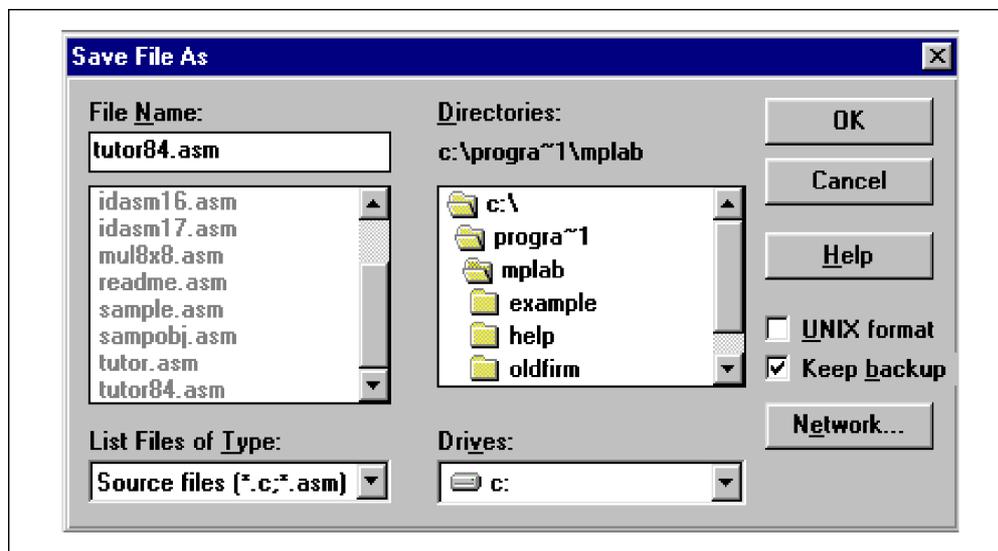
Вы вернетесь к диалоговому окну редактирования параметров проекта, ниже файла кода `tutor84.hex` должен появиться файл исходного текста программы `tutor84.asm`.



Нажатие на кнопку *OK* закроет диалоговое окно и возвратит Вас на рабочий стол среды MPLAB IDE к открытому, но еще не названному исходному файлу.

### 3.5 Создание нового исходного файла

Поставьте курсор в любое место пустого, неназванного файла, который был создан автоматически при создании нового проекта. Выберите пункт меню *File > Save As...*, укажите файл `tutor84.asm`, нужную директорию и нажмите кнопку *OK*.



На рабочем столе MPLAB IDE будет открыт пустой файл с новым именем.

Имя исходного файла должно быть такое же, как и имя проекта, в данном случае `tutor84`.

Если вы изменяете имя исходного файла, Вы также должны изменить имя проекта. Проекты, в которых используется линкер, допускают, чтобы имя файла кода отличалось от имени исходного файла (см. раздел 4.6).

**Примечание.** При использовании в проекте одного исходного файла MPASM имя файла кода программы (.HEX) будет такое же, как и у исходного файла (.ASM). Имя проекта и имя файла кода должны быть одинаковыми.

### 3.6 Ввод исходного текста программы

Используйте манипулятор «мышь», чтобы расположить курсор в начале файла `tutor84.asm`. Введите следующий текст, в точности как написано ниже. Текст комментария вводится после символа точка с запятой.

```
list p=16f84
include <p16F84.inc>
c1 equ    0x0c    ; Адрес переменной c1 - 0x0c
org      0x00    ; Установить начало программы в вектор сброса 0x00
reset
goto     start   ; Переход на начало программы
org      0x04    ; Указать начало размещения программы в памяти
start
movlw   0x09    ; Установить значение счетчика
movwf   c1      ; Записать его в регистр счетчика
loop
incfsz  c1,F    ; Инкрементировать счетчик, пропустить следующую команду
          ; если результат нуль
goto    loop    ; Продолжить инкрементирование
goto    bug     ; Переинициализировать счетчик
end
```

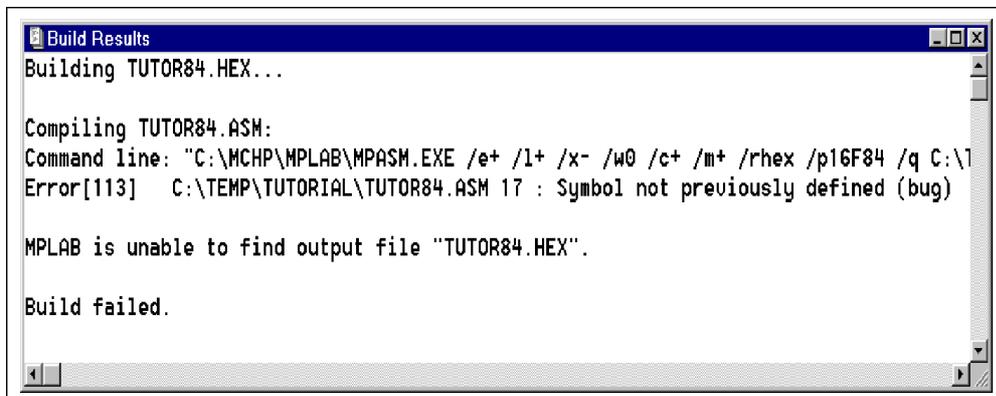
Это очень простая программа, в которой значение регистра инкрементируется, а при переполнении в него записывается значение `0x09`.

Все метки должны начинаться с начала строки, а в последней строке должна быть директива `end`. Подробное описание директив смотрите в документации: MPASM with MPLINK and MPLIB User's Guide. В технической документации на микроконтроллер смотрите список поддерживаемых инструкций.

Сохранение файла выполняется с помощью пункта меню *File > Save*.

### 3.7 Компиляция исходного файла

Компиляция исходного файла может быть выполнено несколькими способами. Описанный здесь метод использует пункт меню *Project > Build All*. После выбора указанного пункта меню, исходный текст программы сохраняется, и запускается программа MPASM. Как только компилирование будет завершено, на экране появится окно результатов.



```
Build Results
Building TUTOR84.HEX...

Compiling TUTOR84.ASM:
Command line: "C:\MCHP\MPLAB\MPASM.EXE /e+ /l+ /x- /w0 /c+ /m+ /rhex /p16F84 /q C:\1
Error[113] C:\TEMP\TUTORIAL\TUTOR84.ASM 17 : Symbol not previously defined (bug)

MPLAB is unable to find output file "TUTOR84.HEX".

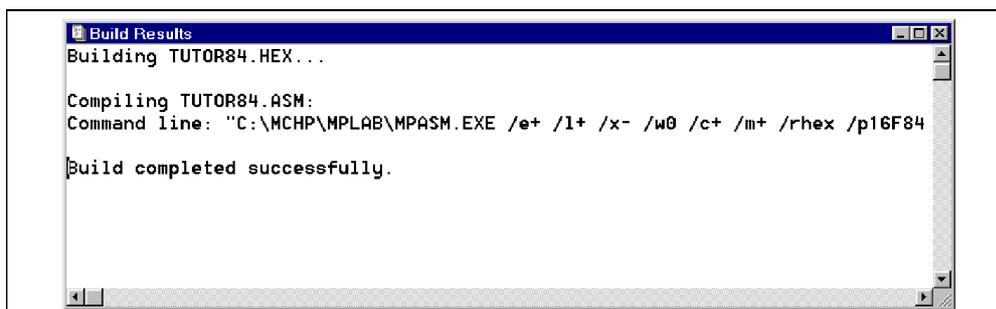
Build failed.
```

В последней строке текста программы преднамеренно была сделана ошибка (раздел 3.6). При выполнении компилирования MPASM выдаст ошибку о не существующей метке. Двойной щелчок «мыши» на сообщении об ошибке перенесет курсор на строку в исходном тексте, где была сделана ошибка.

Исправьте последнюю строку, замените слово *bug* на *start*.

Вновь дайте команду выполнить компилирование *Project > Build All*.

После исправления всех ошибок на экране появится окно результатов с сообщением об успешной компиляции "Build completed successfully". Теперь Вы можете использовать симулятор для проверки работы программы.



```
Build Results
Building TUTOR84.HEX...

Compiling TUTOR84.ASM:
Command line: "C:\MCHP\MPLAB\MPASM.EXE /e+ /l+ /x- /w0 /c+ /m+ /rhex /p16F84

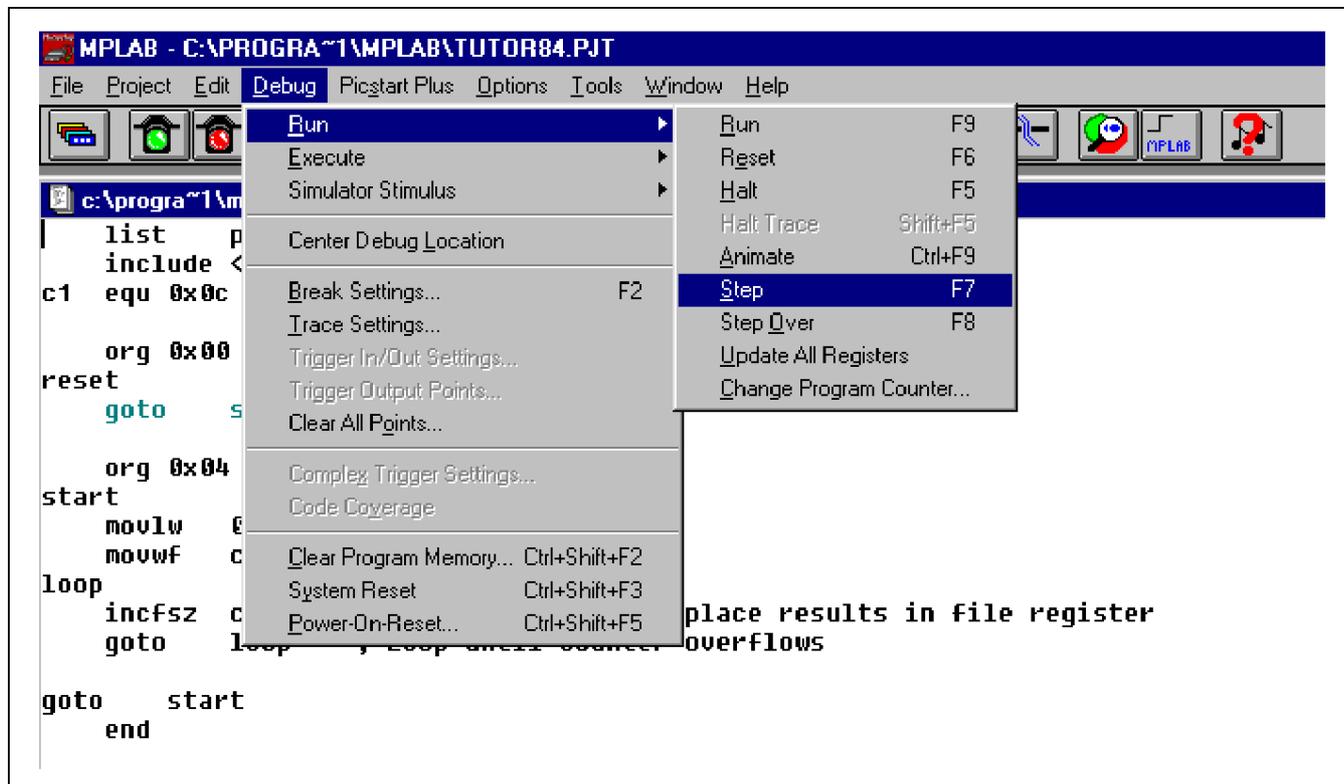
Build completed successfully.
```

**Примечание.** При старте компиляции открытые исходные файлы сохраняются на диске.

### 3.8 Запуск программы

Выберите пункт меню *Debug > Run > Reset* для инициализации системы. Счетчик команд будет установлен в нуль, что является вектором сброса для микроконтроллера PIC16F84. В линейке состояния PC будет равен 0x00.

Затем выберите пункт меню *Debug > Run > Step*: темная полоса будет указывать инструкцию в исходном тексте программы, которая будет выполнена следующей. Значение счетчика программы будет равно «PC:0x04».



Многие пункты меню дублируют кнопки на клавиатуре компьютера.

Пункт меню *Debug > Run > Step* - дублирующая клавиша <F7>. Нажмите кнопку <F7> несколько раз, наблюдая за значением счетчика команд PC и темной полосой, указывающей следующую инструкцию.

Выбором пункта меню *Debug > Run > Run* или нажатием на клавишу <F9> выполняется запуск программы с текущего места. Темная полоса, указывающая текущую инструкцию, изменит свой цвет, ни один параметр на линейке состояния не будет изменять своего значения до приостановки программы.

Остановите программу *Debug > Run > Halt* <F5>, указатель текущей инструкции примет первоначальный цвет, информация в линейке состояния обновится.

Другим способом управления работой программы является нажатие кнопок на графическом меню MPLAB IDE. Если Вы разместите курсор над кнопкой графического меню, то увидите название функции данной кнопки. Крайняя левая кнопка графического меню переключает доступные панели управления, которые могут быть настроены в соответствии с требованиями пользователя. На инструментальной панели отладки, зеленый индикатор соответствует команде RUN <F9>, а красный команде HALT <F5>.

### 3.9 Открытие дополнительных окон

В MPLAB IDE существует большое количество способов контролировать ход выполнения программы. Например программа, которая используется в примере, увеличивает значение счетчика, но как можно убедиться в том, что это действительно происходит? Для просмотра текущего значения регистра воспользуйтесь пунктом меню *Window > File Registers*. На экране появится небольшое окно со всеми регистрами ОЗУ PIC1684.

При каждом шаге программы (нажатие клавиши <F7>) значение регистров в окне будет обновляться. В нашем случае изменяется значение счетчика в регистре с адресом 0x0C. Изменение значения регистра отображается в окне памяти данных, выделяя его другим цветом. Однако в сложных программах одновременно могут изменяться несколько регистров, что усложняет контроль хода выполнения программы. Проблема может быть решена использованием окон с переменными.

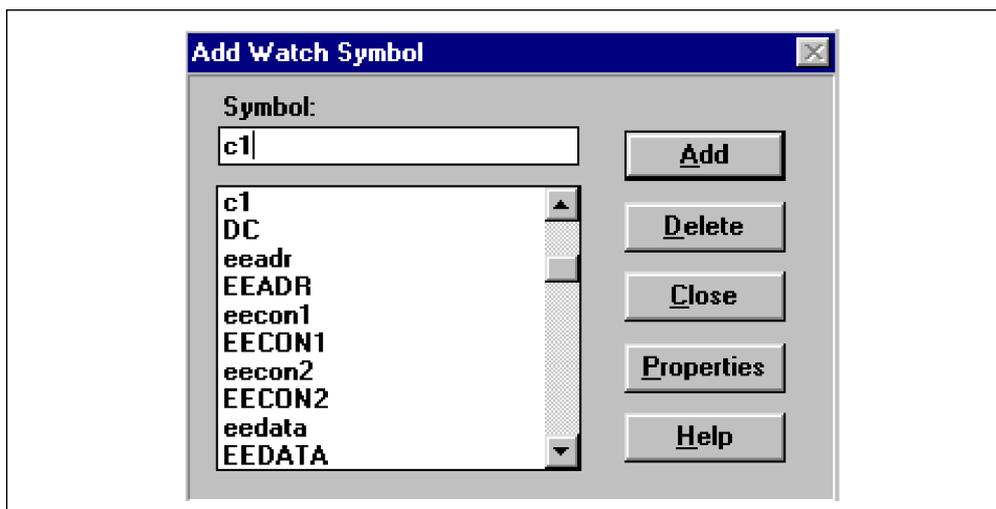
### 3.10 Использование окон с переменными

MPLAB IDE позволяет наблюдать за содержимым регистров в отдельном окне.

#### 3.10.1 Создание окна

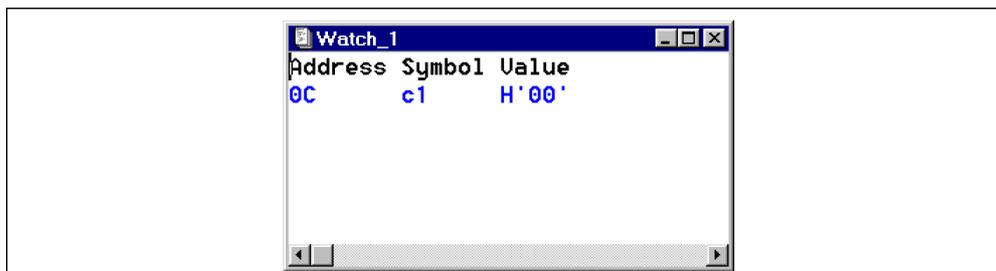
Чтобы создать окно, выберите пункт меню *Window > Watch Window > New Watch Window*. Если окно уже было создано и сохранено на диске, выберите пункт меню *Window>Watch Window>Load Watch Window*. Укажите нужный файл, подтвердив выбор нажатием кнопки *OK*, или двойным нажатием кнопки «мышки» на имени файла.

На экране появится диалоговое окно добавления переменных, показанное на рисунке.



Введите в поле 'Symbol:' новую переменную 'c1' и нажмите кнопку добавить *Add*. Переменная добавится в список. Нажмите кнопку *Close*, и на экране появится окно с переменной c1.

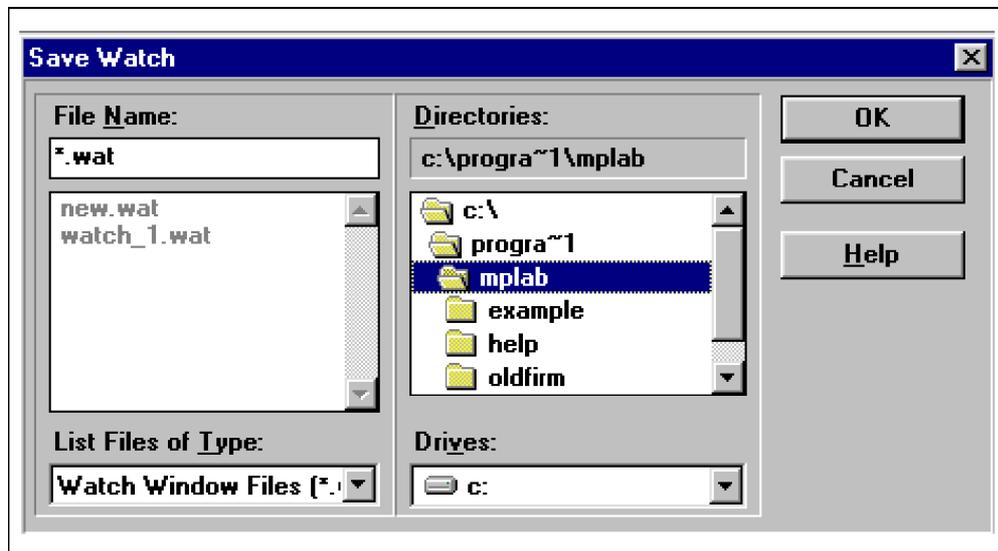
Вы можете разрешить отображение номера строки, выбрав пункт *Toggle Line Numbers* в системном меню окна.



С каждым шагом программы <F7> значение переменной в окне будет обновляться. Если Вы оставили открытым окно регистров памяти данных, то увидите изменение значения регистров и в этом окне.

### 3.10.2 Сохранение окна

Сохранение параметров окна на диске выполняется командой *Window > Watch Window > Save Active Watch* или выбором команды *Save Watch* в системном меню окна (для вызова системного меню окна нажмите на иконку в левом верхнем углу окна). На экране появится диалог сохранения, показанный на рисунке. Введите имя и нажмите кнопку **OK**.



Состояние окна, его расположение на экране сохраняется вместе с проектом. При открытии проекта окно также будет восстановлено.

### 3.10.3 Редактирование окна

Редактирование окна можно выполнять только после его создания (открытия).

Для редактирования окна Вы можете использовать подменю *Window > Watch Window* или системное меню окна.

- |                                  |  |
|----------------------------------|--|
| Добавить переменную в окно       | Выберите <i>Window &gt; Watch Window &gt; Add to Active Watch</i> в меню MPLAB IDE или <i>Add Watch</i> в системном меню окна.   |
| Удалить переменную из окна       | Укажите удаляемую переменную в окне и нажмите <i>Delete Watch</i> в системном меню окна.   |
| Выбор формата отображения данных | Выберите <i>Window &gt; Watch Window &gt; Edit Active Watch</i> в меню MPLAB IDE или <i>Edit Watch</i> в системном меню окна, а затем, нажмите кнопку <i>Propirties</i> . В диалоговом окне настройки параметров отображения данных Вы можете указать: формат, размер, порядок байт и номер выводимого бита. |

### 3.11 Точки остановки

Нажатием кнопки <F5> (*Debug > Run > Halt*) остановите выполнение программы. Щелкните левой кнопкой «мышки» на строке с текстом 'movlw 0x09', затем правой кнопкой «мышки» на той же строке, появится диалоговое окно, показанное на рисунке.



Выберите пункт *Break Point(s)*, указательная линия изменит свой цвет, показывая точку остановки.

Нажмите кнопку <F6> (*Debug > Run > Reset*) для сброса, а затем <F9> для начала выполнения программы.

Выполнение программы будет прервано в точке остановки, значение переменной 'c1' после сброса будет равно 0x00 (Вы можете увидеть значение регистра в окне переменных или в окне памяти данных). Выполнение одного шага программы изменит значение переменной 'c1' на 0x09. Нажмите кнопку <F9> несколько раз, Вы заметите изменение цвета линейки состояния во время выполнения программы.

**Примечание.** Если выполнение программы не прерывается в точке остановки, то проверьте глобальное разрешение остановки программы *Global Break Enable* в пункте меню *Options >Development Mode - Break Options*.

### 3.12 Резюме к главе

В рассмотренном примере было показано как:

- создать новый проект;
- создать и подключить исходный файл к проекту;
- выполнить компиляцию исходного текста программы;
- управлять ходом выполнения программы, используя симулятор;
- использовать точки остановки;
- наблюдать значения переменных и регистров в ходе выполнения программы.

Полностью разобравшись с темами, описанными в этой главе, переходите к изучению следующей, для получения дополнительной информации.

**Несколько полезных советов:**

**Точки остановки** – Вы можете устанавливать точки остановки в окне памяти программ (*Window > Program Memory*), в окне исходного текста программы (в данном случае *tutor84.asm*) и в окне файла листинга программы (*Window > Absolute Listing*).

**Исходные файлы** – используйте пункт меню *Window > Project Window* для просмотра списка исходных файлов в проекте. Открыть исходный файл в редакторе MPLAB можно двойным щелчком левой кнопкой «мыши» на названии файла.

**Ошибки MPASM** – в случае возникновения ошибок при компиляции исходных файлов открывается окно с полным списком ошибок. Выполнив двойной щелчок левой кнопки «мыши» на ошибке, курсор будет установлен в место ошибки исходного текста программы. Всегда сначала выбирайте первую ошибку в списке, поскольку она может быть причиной остальных ошибок.

**Биты конфигурации микроконтроллера и режим работы MPLAB IDE** – биты конфигурации, указанные в исходном тексте программы, не будут настраивать параметры работы симулятора и эмулятора. Например, в исходном тексте программы настройкой битов конфигурации разрешена работа WDT. Для настройки работы WDT при использовании симулятора или эмулятора необходимо выбрать пункт меню *Options > Development Mode* раздел *Configuration* и указать параметры. Это сделано для того, чтобы была возможность имитировать работу микроконтроллера в различных режимах без изменения исходного текста программы.

Используя пункт меню *Options > Development Mode*, выберите тип микроконтроллера. Установка типа микроконтроллера в исходном тексте программы MPASM или MPLAB-CXX не изменяет настроек MPLAB IDE.

**Параметры** – в меню *Options > Environment Setup* раздел *General* можно настроить следующие параметры:

- используемый шрифт и его размер;
- размещение графического меню на экране монитора;
- состав графического меню;
- количество символов метки, отображаемых на экране.

Прежде чем Вы закроете это диалоговое окно, выберите раздел *Key Mappings*, в котором можно назначить «горячие» клавиши на функции MPLAB IDE и специальные символы ASCII.

**Файл карты памяти** – для генерации файла карты памяти установите флажок напротив параметра MAP в диалоговом окне настройки проекта *Project > Edit Project* раздел *Node Properties*. Файл *tutor84.map* позволяет оценить использование памяти микроконтроллера.

**Серый цвет пункта меню** – если пункт меню отображается серым цветом, то он недоступен для использования. Если Вы уверены, что все настроено правильно и выключен режим «только чтение», попробуйте перезагрузить MPLAB IDE.

## Глава 4. Среда проектирования MPLAB IDE

### 4.1 Введение

В этой главе подробно рассмотрен вопрос настройки проектов. Если Вы полностью изучили вводный материал главы 3, можете приступить к рассмотрению разделов этой главы.

Менеджеры проектов MPLAB IDE версии 3.40 и выше поддерживают проекты, состоящие из нескольких файлов. Проекты, выполненные в MPLAB IDE версии 3.31 и ниже, будут автоматически конвертированы в новую версию при их открытии. Повторное открытие конвертированных проектов в ранних версиях невозможно.

### 4.2 Основные разделы главы

Краткий обзор проектов MPLAB IDE

Компиляция проекта с одним исходным файлом MPASM

Компиляция одного исходного файла MPASM без создания проекта

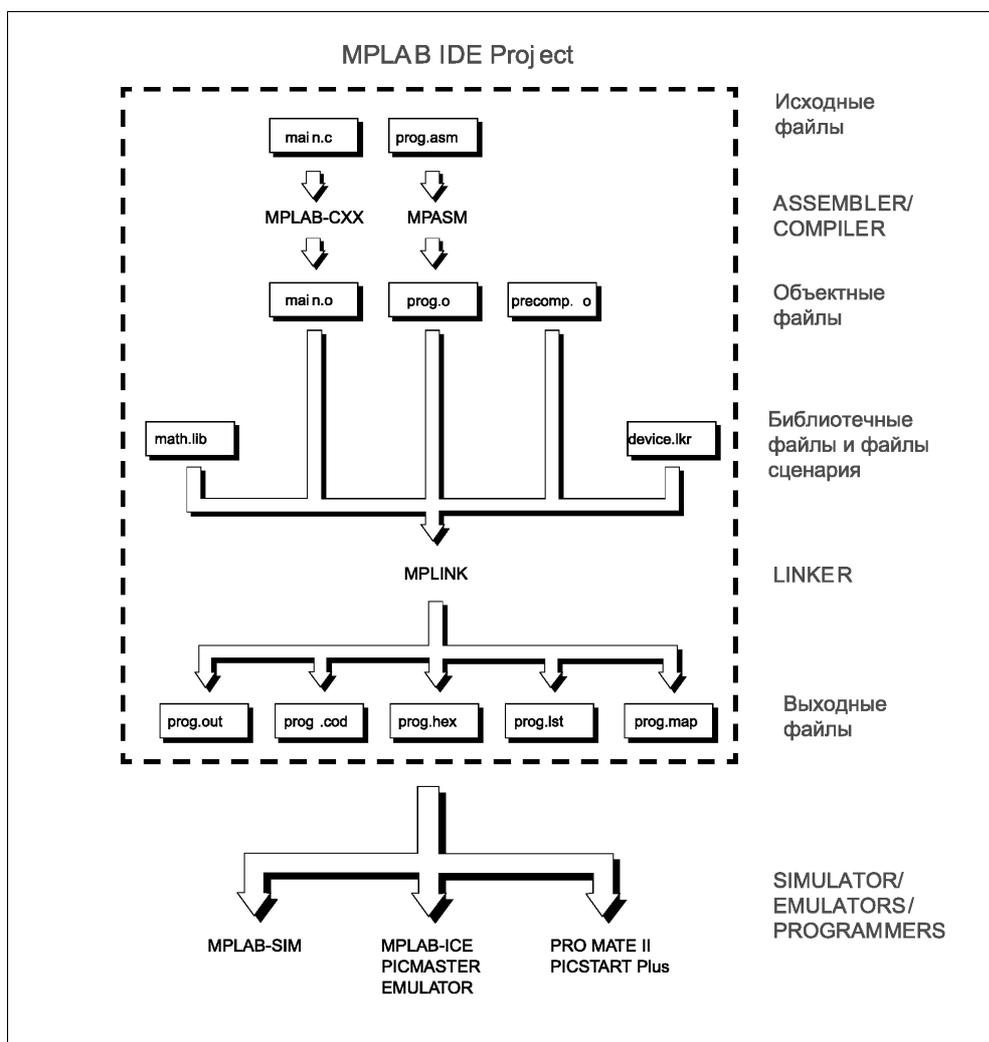
Создание проекта с несколькими исходными файлами MPASM

Создание проекта совместно с HI-TECH PIC C

Создание проекта совместно с MPLAB-C17 или MPLAB-C18

### 4.3 Краткий обзор проектов MPLAB IDE

Проект MPLAB IDE – это группа файлов, необходимых для работы различных инструментальных средств среды проектирования. Проект состоит из узла компиляции и одного или нескольких исходных узлов. Исходными узлами являются: исходные файлы, написанные на ассемблере или C, объектные файлы и файлы сценария линкера. Обычно файлы проекта размещаются в той же директории, что и исходные файлы.



В проекте, показанном на рисунке, исходный файл main.c связан с компилятором MPLAB-CXX. MPLAB IDE будет использовать эту информацию, чтобы создать файл main.o для линкера MPLINK. Дополнительную информацию смотрите в документации MPLAB-CXX User's Guide (DSXXXXXX).

Файл источника prog.asm связан с ассемблером MPASM. MPLAB IDE, будет использовать эту информацию, чтобы создать файл prog.o для линкера MPLINK. Дополнительную информацию смотрите в документации MPASM User's Guide with MPLINK and MPLIB (DS33014).

Кроме того, могут быть подключены дополнительные объектные файлы (ресомп.o), которые могут содержать:

- стартовый код;
- код инициализации;
- типовую методику обработки прерываний;
- описание переменных.

Также эти файлы могут содержать характеристики устройства и/или модель памяти.

Подробную информацию по дополнительным файлам смотрите в документации MPLAB-CXX Reference Guide (DS51224).

Некоторые файлы библиотек (math.lib) входят в состав компилятора, другие библиотечные файлы могут быть созданы с помощью программы MPLIB. Дополнительную информацию смотрите в документации MPASM User's Guide with MPLINK and MPLIB. Информацию о доступных библиотеках Microchip смотрите в описании на компиляторы MPLAB-CXX.

Для генерации выходных файлов проекта с использованием линкера MPLINK применяются объектные файлы вместе с файлами библиотек и файлами сценария (device.lkr). Дополнительную информацию смотрите в документации MPASM User's Guide with MPLINK and MPLIB.

Основным результатом работы линкера MPLINK является HEX файл (prog.hex), используемый симулятором MPLAB SIM, эмулятором MPLAB ICE и пригодный для записи в микроконтроллер программаторами.

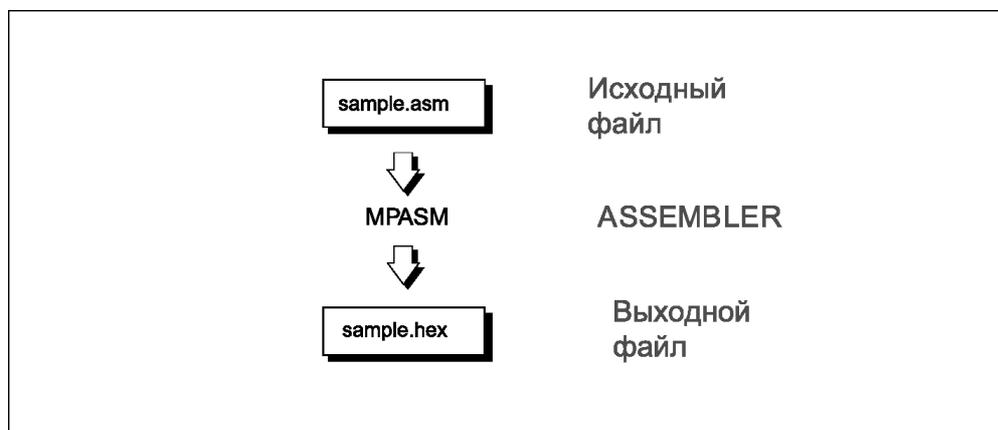
Другие файлы, формируемые линкером:

- COFF файл (.out), промежуточный файл MPLINK;
- Файл (.cod), файл отладки используемый MPLAB IDE;
- Файл карты памяти (.map), показывает использование памяти микроконтроллера;
- Файл листинга программы (.lst), первоначальный исходный текст совмещенный с кодом программы.

Инструментальные средства, описанные здесь, разработаны фирмой Microchip Technology Incorporated. Для подключения к среде проектирования MPLAB IDE инструментальных других производителей обратитесь к документации Third Party Guide (DS00104).

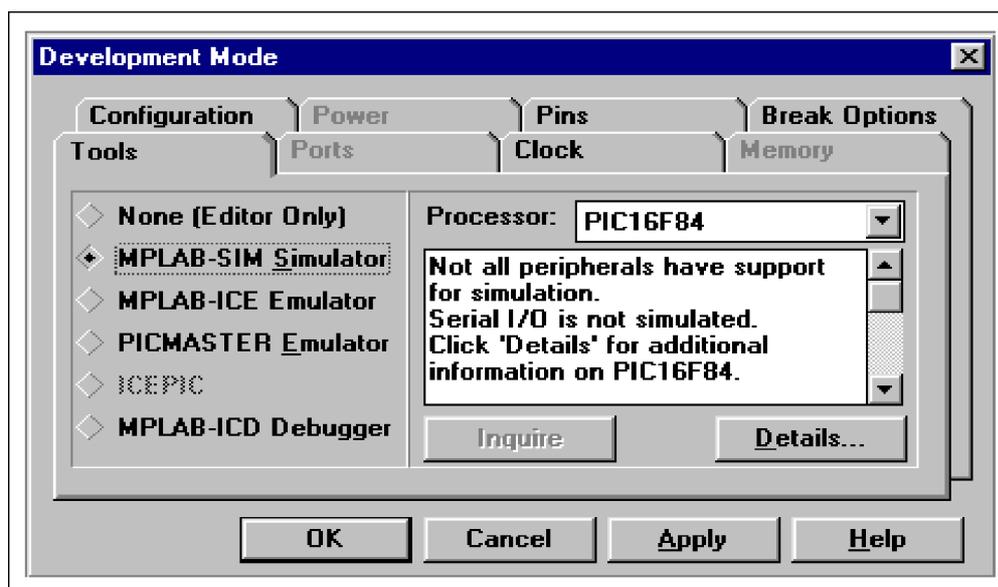
## 4.4 Компиляция проекта с одним исходным файлом MPASM

В разделе рассмотрены вопросы создания проекта с одним исходным файлом (или если Вы используете версию MPLAB IDE 3.31), подключая дополнительные файлы с помощью директивы #include.



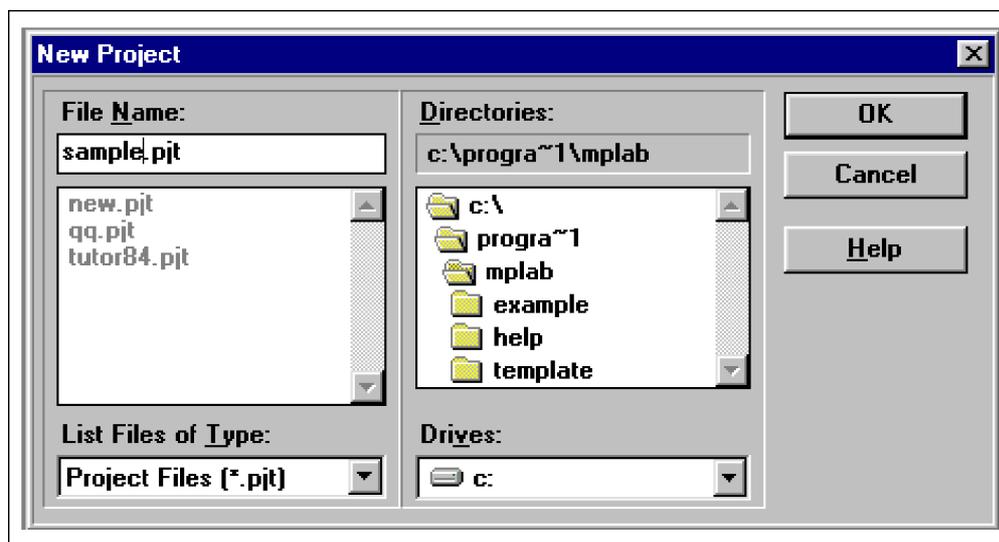
### 4.4.1 Настройка параметров среды проектирования

Укажите требуемый режим работы в меню *Options > Development Mode > Tools*, выберите симулятор MPLAB SIM и микроконтроллер PIC16F84. Подтвердите выбор кнопкой *OK*.



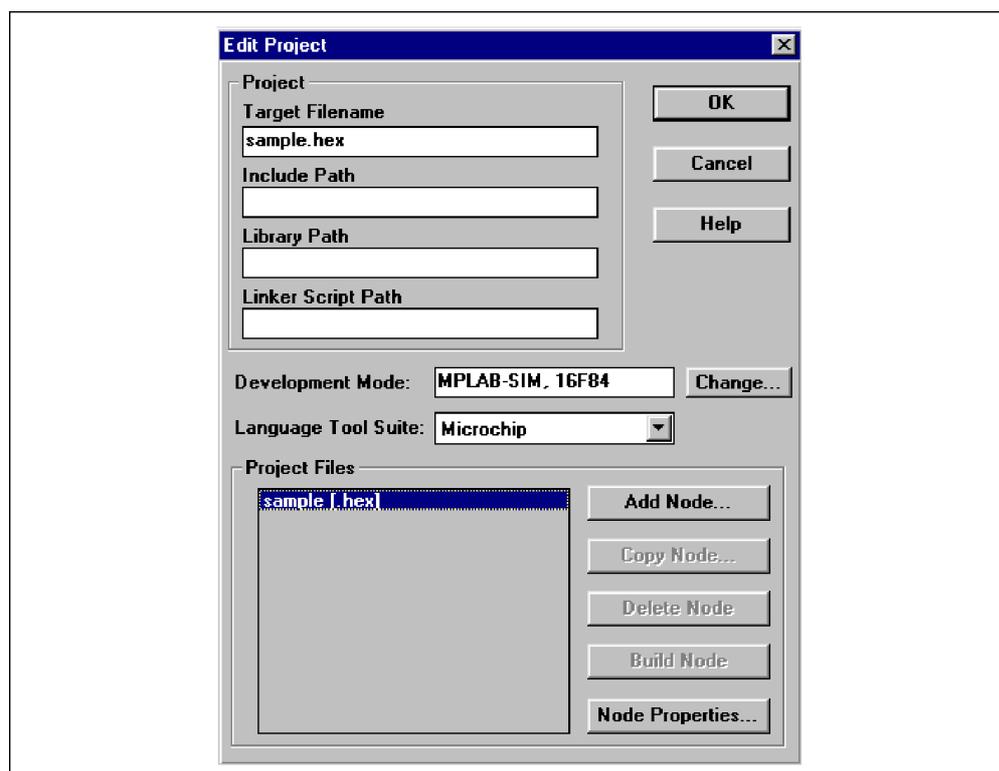
#### 4.4.2 Создание нового проекта

Создайте новый проект - *Project > New Project*, указав директорию размещения и имя. Для примера, назовите проект *sample.pjt*, сохранив его в директории *Program Files\MPLAB*.



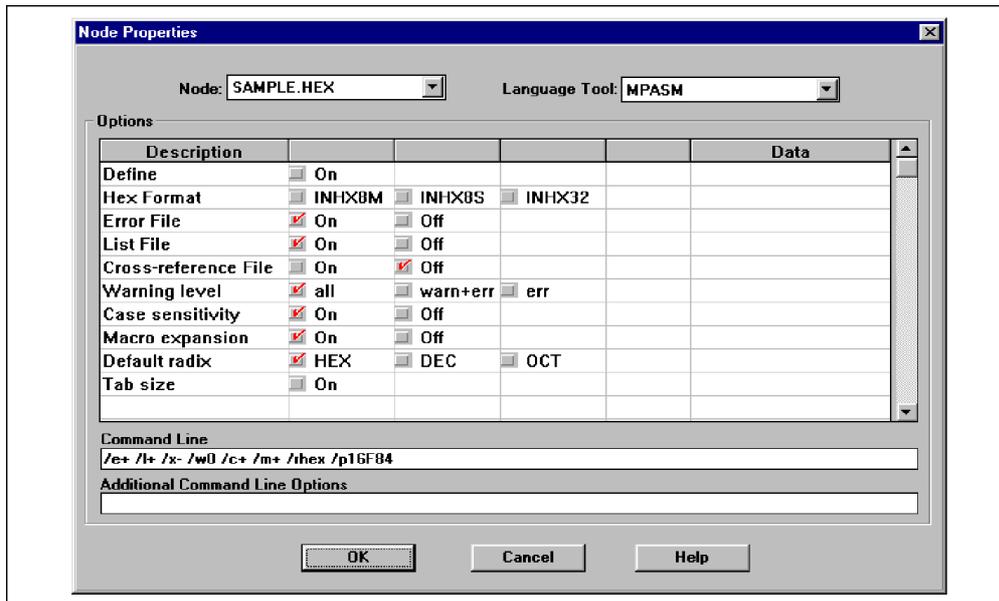
#### 4.4.3 Диалоговое окно настройки проекта

Подтвердив создание проекта на экране, появится диалоговое окно настройки параметров.



#### 4.4.4 Настройка параметров компиляции

Выберите файл `sample.hex` в списке файлов проекта и нажмите кнопку *Node Properties*.

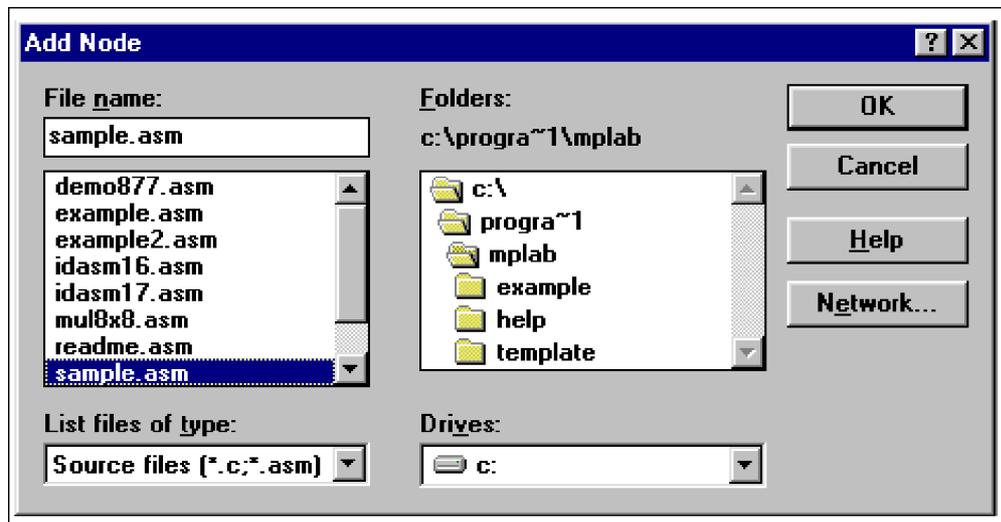


Это диалоговое окно содержит параметры настройки, указанные по умолчанию, для инструментального средства, показанного в правом верхнем углу (в данном случае для MPASM). Изменение параметров в этом случае не требуется. Подробное описание данного диалогового окна смотрите в документации MPASM with MPLINK and MPLIB User's Guide.

Нажмите кнопку *OK* для возвращения в диалоговое окно настройки проекта.

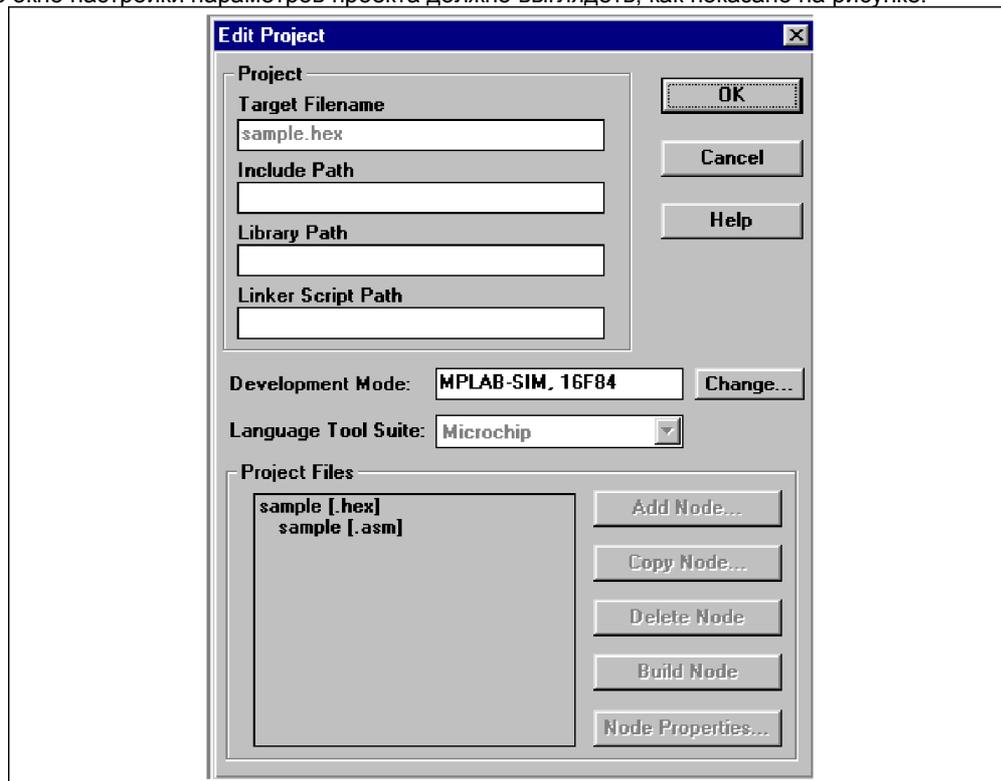
#### 4.4.5 Подключение исходного файла

Нажмите кнопку *Add Node*. Вы увидите стандартный диалог выбора файлов (см. рисунок) с открытой рабочей директорией проекта. Для этого примера выберите файл *samples.asm*.



MPASM всегда создает HEX файл с таким же именем как и исходный файл. Менеджер проектов создаст файл *sample.hex*, во время исполнения проекта.

Диалоговое окно настройки параметров проекта должно выглядеть, как показано на рисунке.



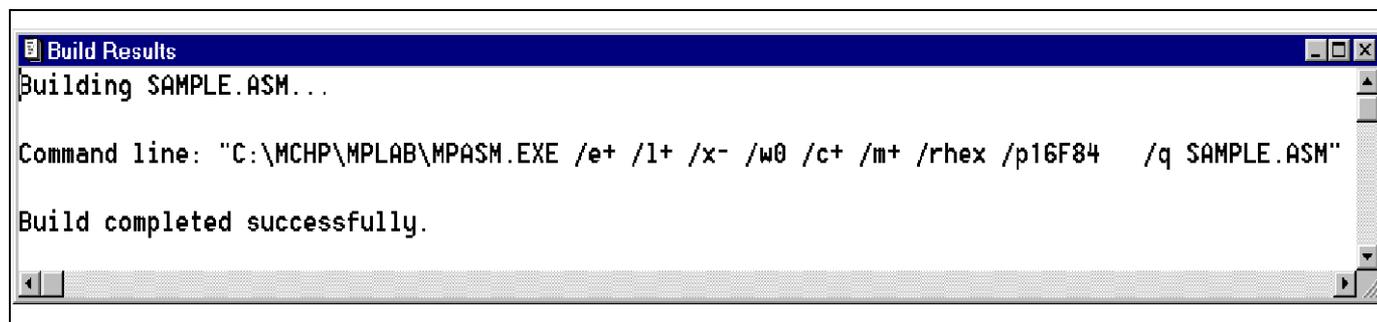
В этом простом примере не указываются директории размещения файлов в параметрах Path. По мере усложнения проекта Вам может понадобиться указать путь к подключаемым файлам, библиотекам и файлам сценария линкера.

Параметры проекта по умолчанию (при создании нового проекта) можно настроить в меню *Options > Environment Setup* раздел *Projects*.

Нажмите кнопку *OK* для закрытия диалогового окна настройки проекта.

#### 4.4.6 Компиляция исходного текста программы

Выберите пункт меню *Project > Make Project* для компиляции текста программы с помощью MPASM. Если компиляция выполнена без ошибок, на экране появится сообщение, показанное на рисунке.

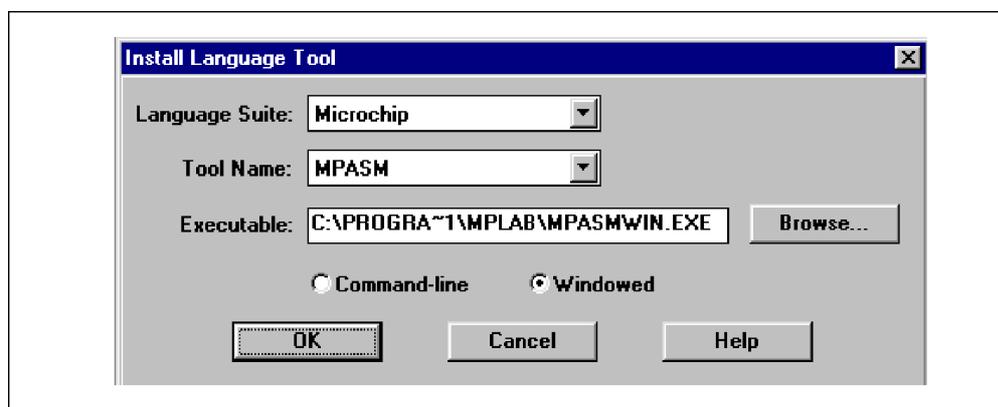


#### 4.4.7 Поиск и устранение ошибок

В случае появления ошибок при выполнении компиляции воспользуйтесь следующими рекомендациями:

1. Проверьте наличие синтаксических ошибок. Двойным щелчком «мыши» на сообщении об ошибке курсор будет перемещен в место ошибки.
2. В меню *Project > Edit Project* нажмите кнопку *Node Properties* и проверьте параметры работы MPASM.
3. В меню *Project > Edit Project* проверьте правильность указания имен файлов. Если вы неправильно указали имя файла, выберите его и нажмите кнопку *Delete Node*. Затем подключите нужный файл как описано в разделе 4.4.5.

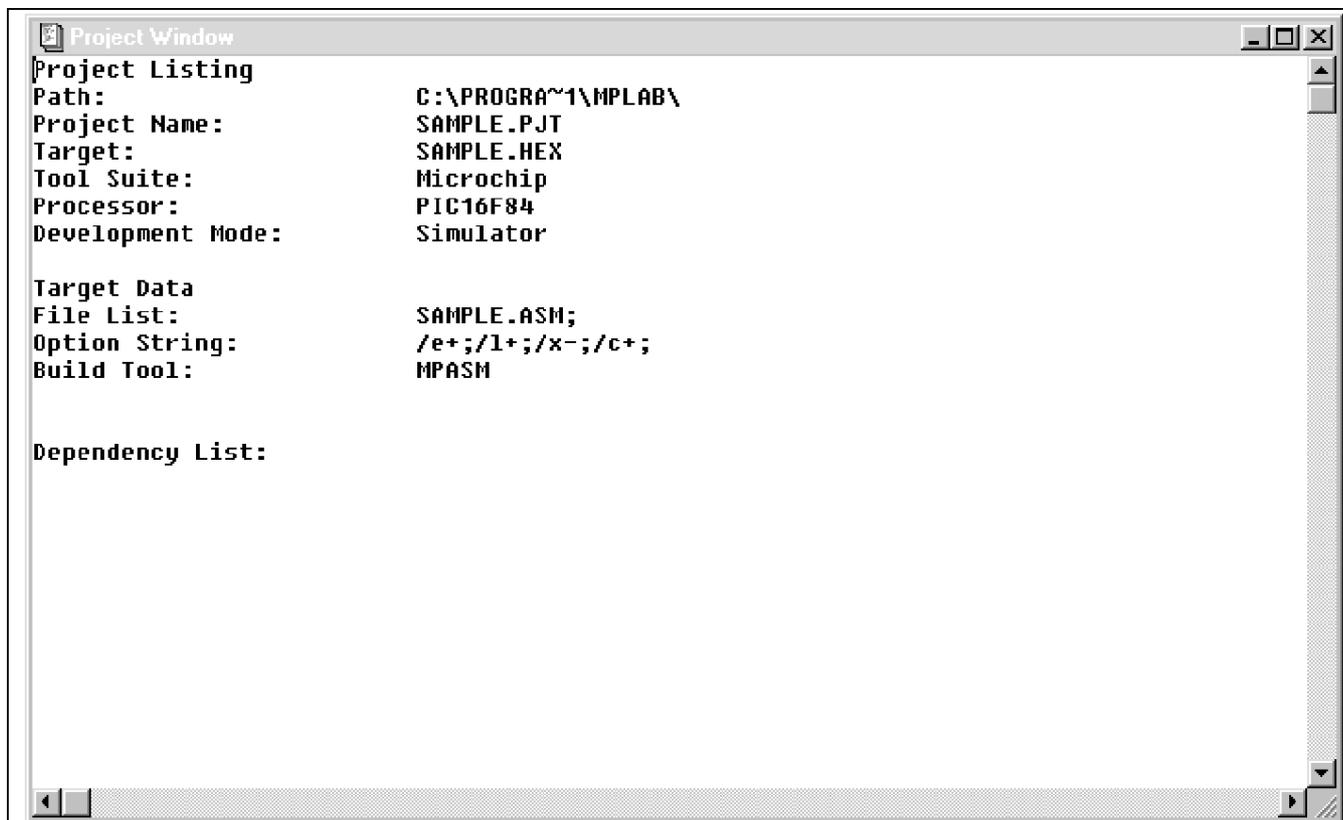
С помощью пункта меню *Project > Install Language Tool...* проверьте, чтобы файл MPASMWIN.EXE находился в директории вместе с MPLAB IDE.



Альтернативой MPASMWIN.EXE может быть программа MPASM.EXE, работающая под управлением операционной системой DOS и запускаемой из командной строки. MPASM.EXE может не работать на компьютерах Pentium 100MHz и выше.

#### 4.4.8 Окно проекта

Для проверки соответствия имен файлов и других параметров работы воспользуйтесь пунктом меню *Window > Project window*. На экране появится окно, показанное на рисунке. В нашем примере, файлы с расширением *.ASM* и *.HEX* будут иметь имя *SAMPLE*.



#### 4.4.9 Заключение

Итак, основные шаги создания нового проекта:

1. создать проект Project > New Project;
2. указать в качестве компилятора MPASM и настроить параметры компиляции;
3. подключить исходный файл к проекту.

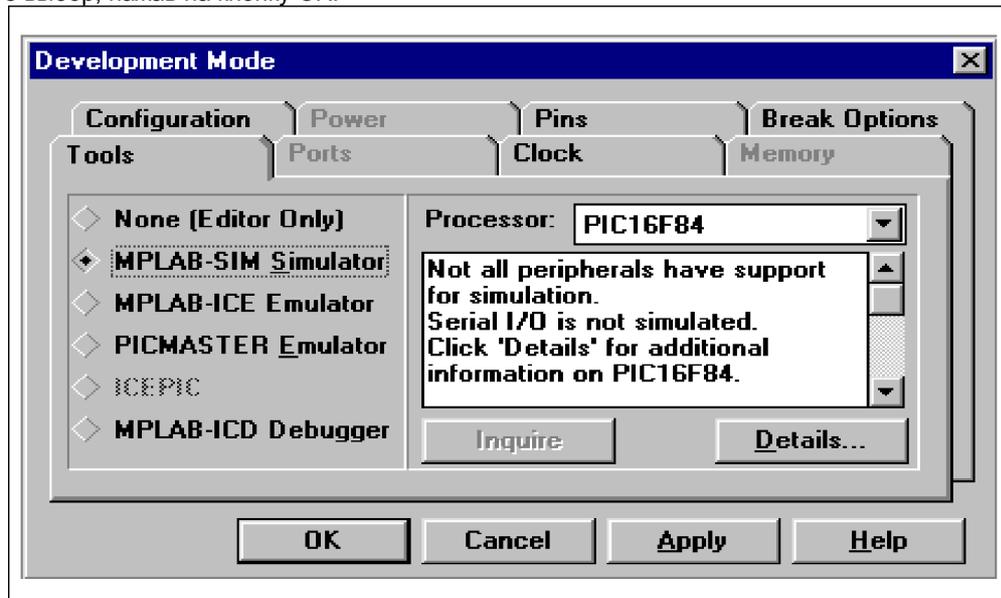
## 4.5 Компиляция одного исходного файла MPASM без создания проекта

MPLAB IDE позволяет выполнять компиляцию исходного текста программы без создания проекта. Неудобством такого метода компиляции является то, что необходимо каждый раз указывать параметры работы. В данном примере будет использоваться тот же исходный файл, что и в предыдущем случае.

Сначала закройте все открытые проекты - *Project > Close Project*.

### 4.5.1 Настройка параметров среды проектирования

Выберите симулятор MPLAB SIM и тип микроконтроллера PIC16F84 - *Options > Development Mode* раздел *Tools*. Подтвердите выбор, нажав на кнопку *OK*.



### 4.5.2 Открытие файла исходного текста программы

Откройте исходный файл, который Вы хотите компилировать. В данном примере используйте файл *sample.asm* из рабочей директории MPLAB IDE.

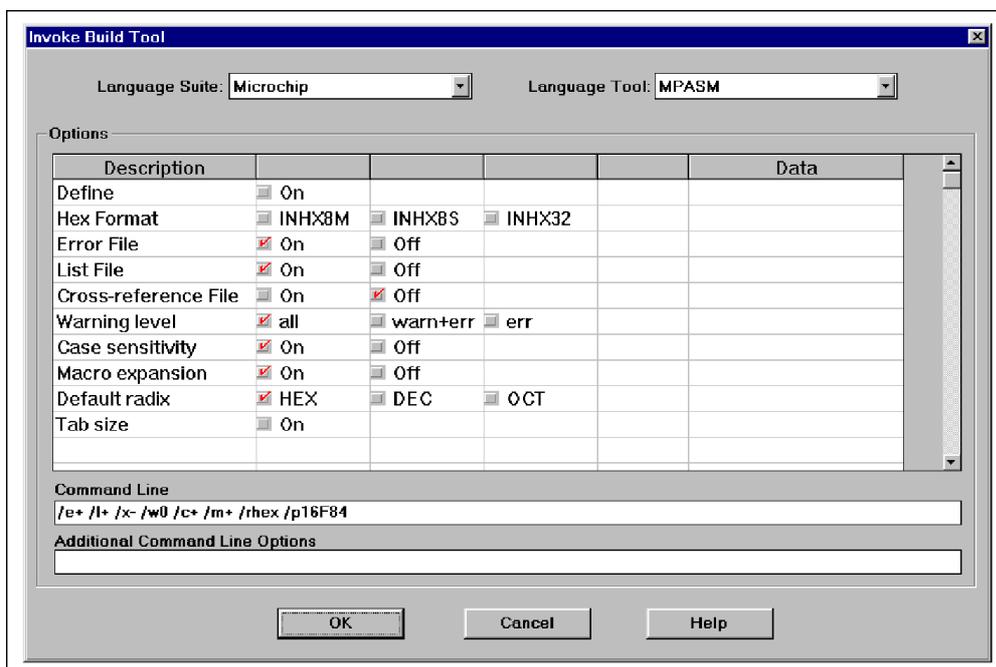
```

c:\progra~1\mplab\sample.asm
;*****
;                               SAMPLE.ASM
;                               8x8 Software Multiplier for 16Cxxx Family
;*****
;
;   The 16 bit result is stored in 2 bytes
;
; Before calling the subroutine " mpy ", the multiplier should
; be loaded in location " mulplr ", and the multiplicand in
; " mulcnd ". The 16 bit result is stored in locations
; H_byte & L_byte.
;*****
;
;   LIST    p=16F84 ; PIC16F844 is the target processor
;
; #include "P16F84.INC" ; Include header file
;
; cblock 0x10 ; Temporary storage
;     mulcnd ; 8 bit multiplicand
;     mulplr ; 8 bit multiplier, this register will be set to zero after multiply
;     H_byte ; High byte of the 16 bit result
;     L_byte ; Low byte of the 16 bit result
;     count ; loop counter
; endc
;
;     org    0

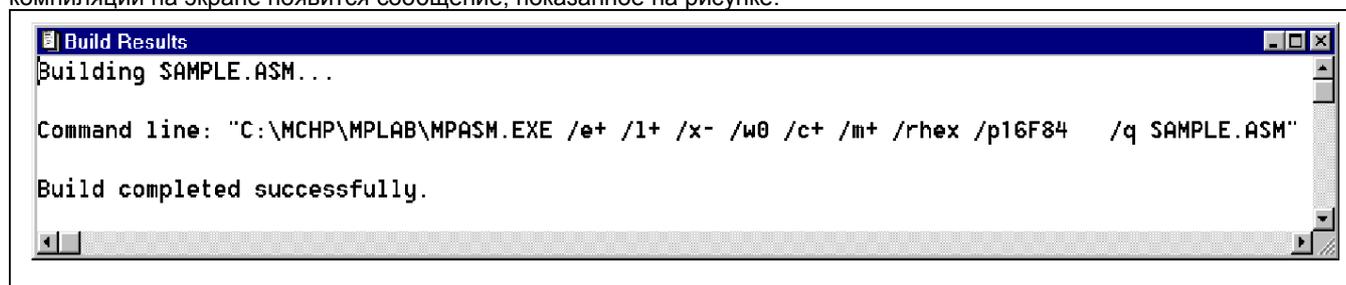
```

### 4.5.3 Компиляция исходного текста программы

Выберите пункт меню *Project > Build Node* для настройки параметров работы MPASM с файлом `sample.asm`. На экране появится диалоговое окно, показанное на рисунке.



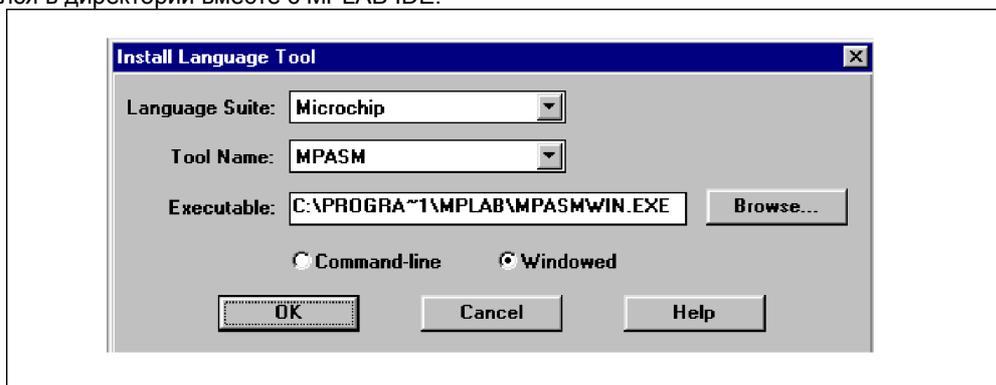
Для подтверждения выбранных параметров и начала компиляции нажмите кнопку **OK**. При успешной компиляции на экране появится сообщение, показанное на рисунке.



### 4.5.4 Поиск и устранение ошибок

В случае появления ошибок при выполнении компиляции воспользуйтесь следующими рекомендациями:

1. Проверьте наличие синтаксических ошибок. Двойным щелчком «мышки» на сообщении об ошибке курсор будет перемещен в место ошибки.
2. С помощью пункта меню *Project > Install Language Tool...* проверьте, чтобы файл `MPASMWIN.EXE` находился в директории вместе с `MPLAB IDE`.



Альтернативой `MPASMWIN.EXE` может быть программа `MPASM.EXE`, работающая под управлением операционной системой DOS и запускаемая из командной строки. `MPASM.EXE` может не работать на компьютерах Pentium 100MHz и выше.

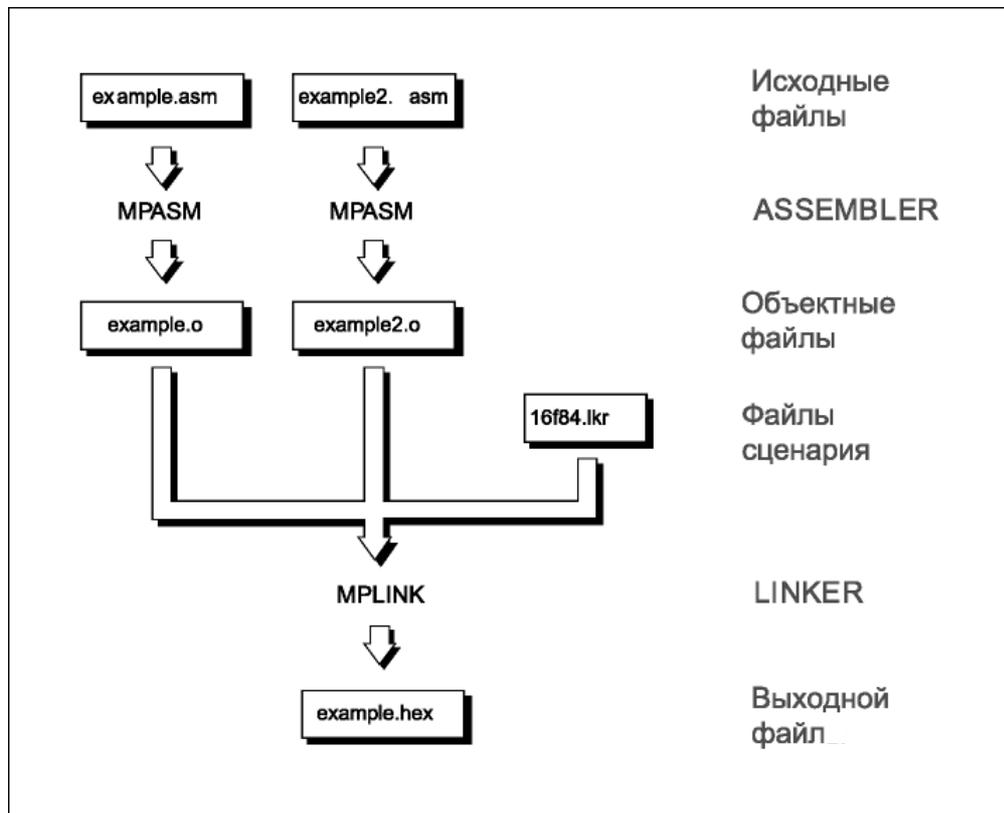
#### 4.5.5 Заключение

Итак, основные шаги компиляции исходного файла без создания проекта:

1. закрыть открытые проекты – *Project > Close Project*;
2. открыть исходный файл;
3. выберите пункт меню – *Project > Build Node*;
4. выбрать программу и параметры компиляции в диалоговом окне *Invoke Build Tool* (раздел 4.5.3).

## 4.6 Создание проекта с несколькими исходными файлами MPASM

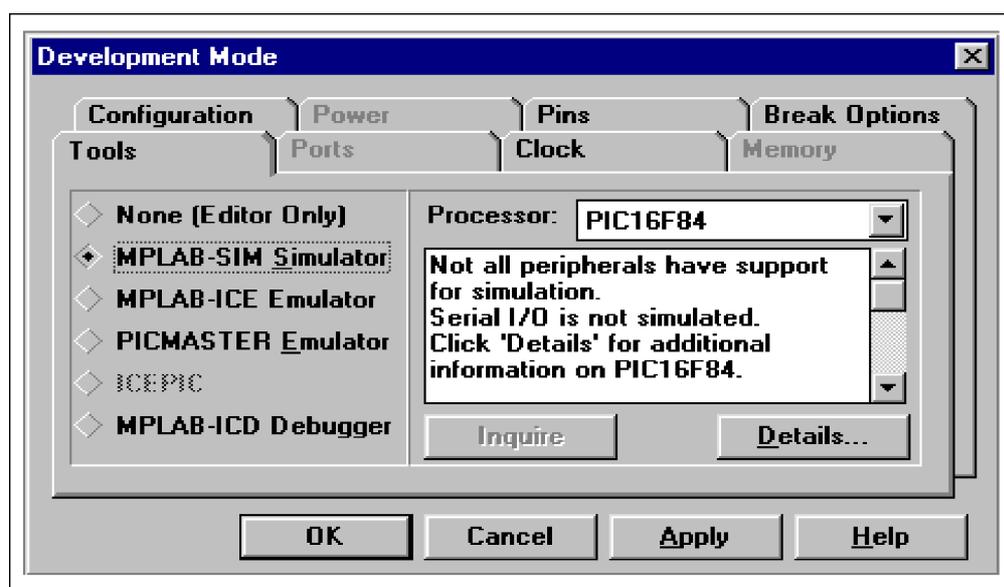
Для связи нескольких исходных файлов в один HEX файл необходимо использовать линкер MPLINK.



Если есть открытые проекты, то закройте их - *Project > Close Project*.

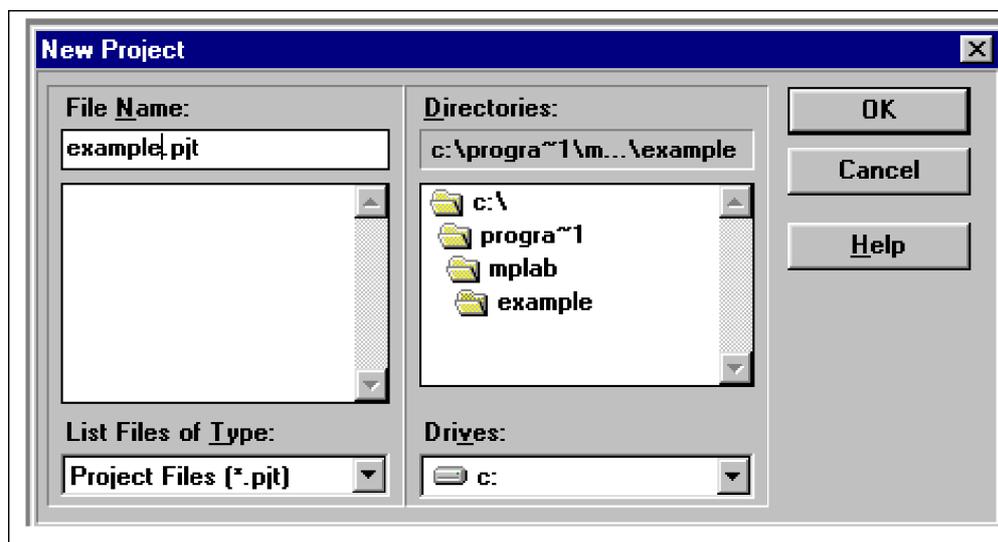
### 4.6.1 Настройка параметров среды проектирования

Выберите симулятор MPLAB SIM и тип микроконтроллера PIC16F84 - *Options > Development Mode > Tools*. Подтвердите выбор, нажав на кнопку *OK*.



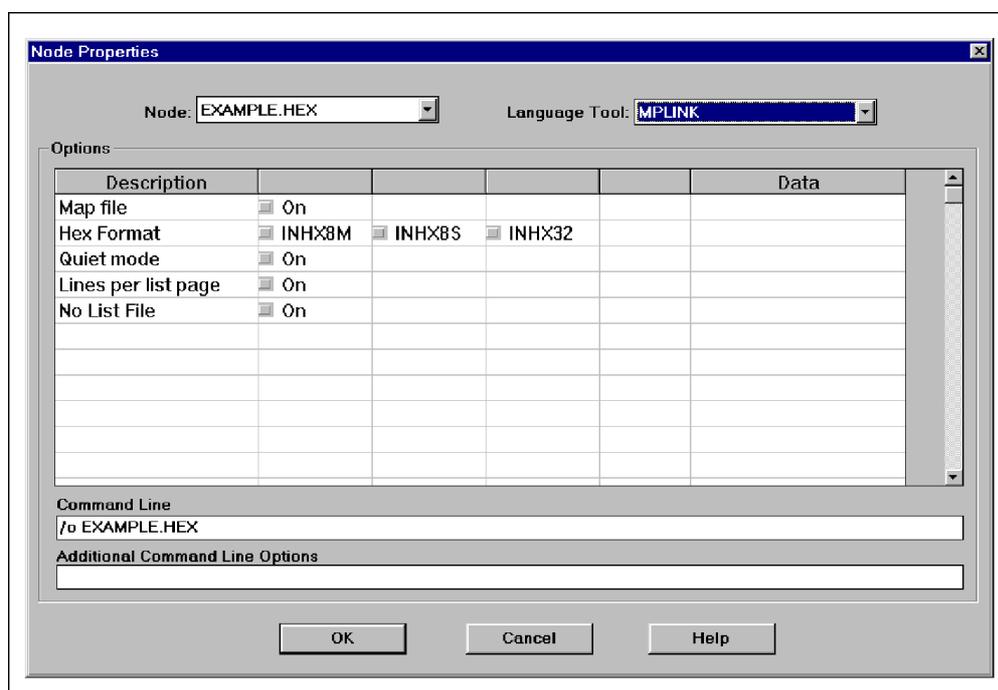
#### 4.6.2 Создание нового проекта

Создайте новый проект - *Project > New Project*, указав директорию размещения и имя. Для примера, назовите проект `example.pjt`, сохранив его в директории `\PROGRAM FILES\IMPLAB\EXAMPLE`.



#### 4.6.3 Настройка параметров компиляции

Выберите HEX файл из списка в диалоговом окне настройки параметров проекта и нажмите кнопку *Node Properties*. На экране появится следующее диалоговое окно, показанное на рисунке. Укажите в качестве компоновщика (строка `language tool`) MPLINK.



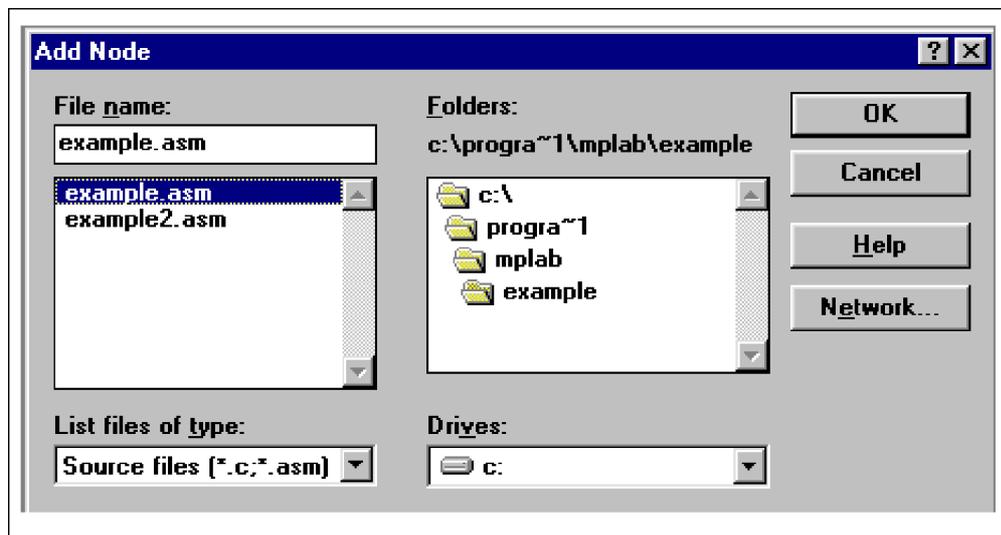
Это диалоговое окно содержит параметры настройки инструментального средства, в данном случае для MPLINK. При первом открытии диалогового окна, в нем будут указаны параметры установленные по умолчанию.

В данном примере изменять параметры не требуется. Дополнительную информацию смотрите в документации MPASM with MPLINK and MPLIB User's Guide.

Нажмите кнопку *OK* для возвращения в диалоговое окно настройки параметров проекта.

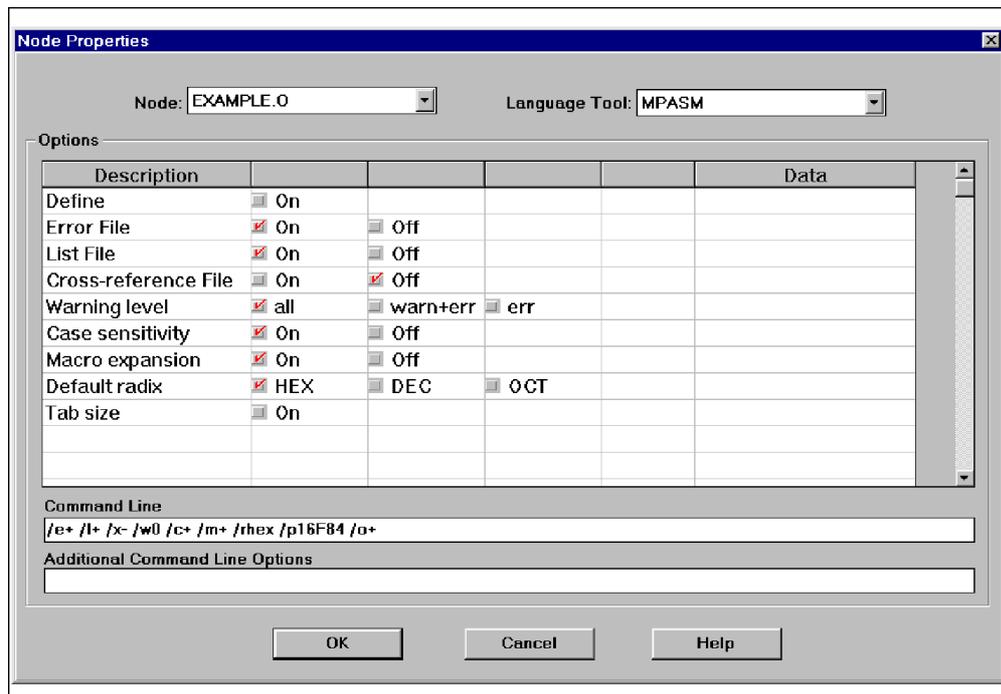
#### 4.6.4 Подключение первого исходного файла

Нажмите кнопку *Add Node* в диалоговом окне настройки проекта. Для этого примера, выберите исходный файл `example.asm` в директории `\PROGRAM FILES\IMPLAB\EXAMPLE`.



Одновременно можно выбрать несколько файлов. Используйте клавишу `<Ctrl>` для выбора каждого файла в отдельности, а клавишу `<Shift>` для выбора группы файлов.

В диалоговом окне параметров проекта, выберите файл `example.asm` и нажмите кнопку *Node Properties*. Проверьте параметры настройки MPASM.



Подробное описание данного диалогового окна смотрите в документации MPASM with MPLINK and MPLIB User's Guide.

Нажмите кнопку *OK* для возвращения в диалоговое окно настройки параметров проекта.

#### 4.6.5 Подключение дополнительных исходных файлов

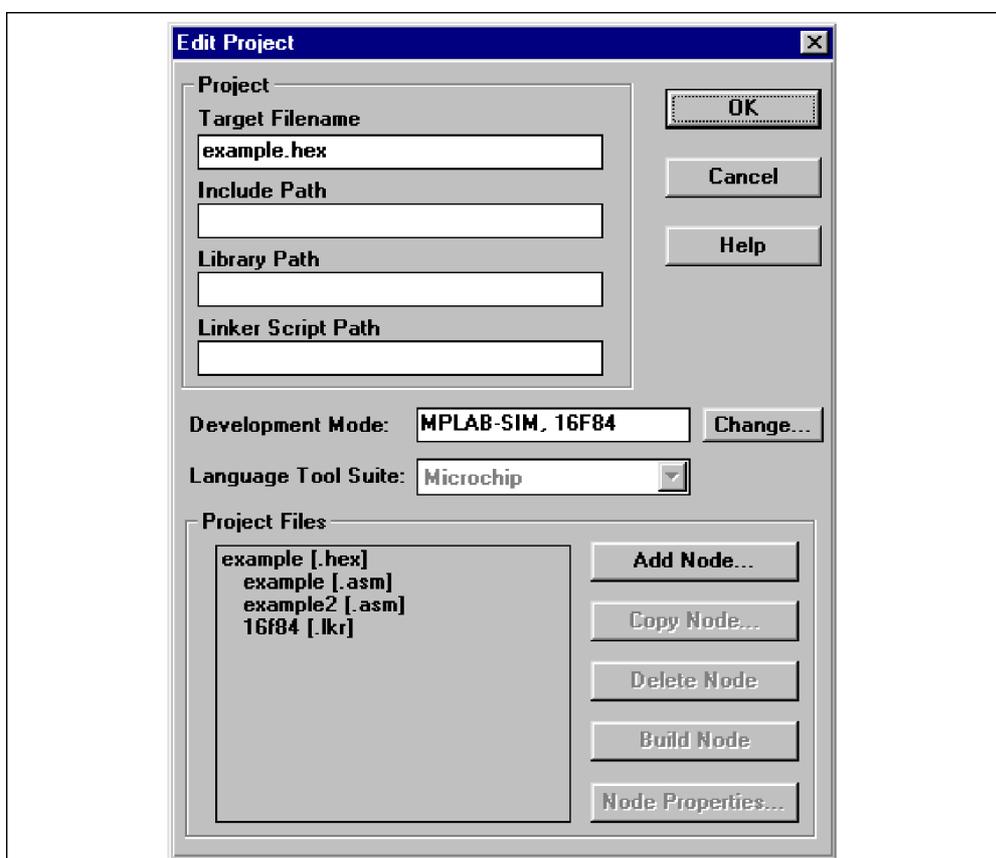
Таким же способом подключите второй исходный файл MPASM `example2.asm` в директории `\PROGRAM FILES\MPLAB\EXAMPLE`.

Вы можете воспользоваться кнопкой *Copy Node* для копирования параметров компиляции первого исходного файла и присвоения их другим файлам.

Выберите один из исходных файлов в списке диалогового окна настройки проекта и нажмите кнопку *Copy Node*. В диалоговом окне подключения исходных файлов (*Add Node*) выберите один или несколько файлов и нажмите кнопку *OK*. Это действие присвоит параметры компиляции добавленным файлам, что особенно удобно при подключении большого числа файлов.

#### 4.6.6 Подключение файла сценария

Аналогично подключите файл сценария линкера, нажав кнопку *Add Node*. Файлы сценария позволяют учитывать линкеру MPLINK объем памяти и архитектуру микроконтроллера PICmicro. Стандартные файлы сценария MPLINK размещены в рабочей директории MPLAB IDE. В нашем примере необходимо подключить файл `PIC16F84.lkr` из директории `\PROGRAM FILES\MPLAB\EXAMPLE`. Для файлов сценария параметры компиляции не указываются.



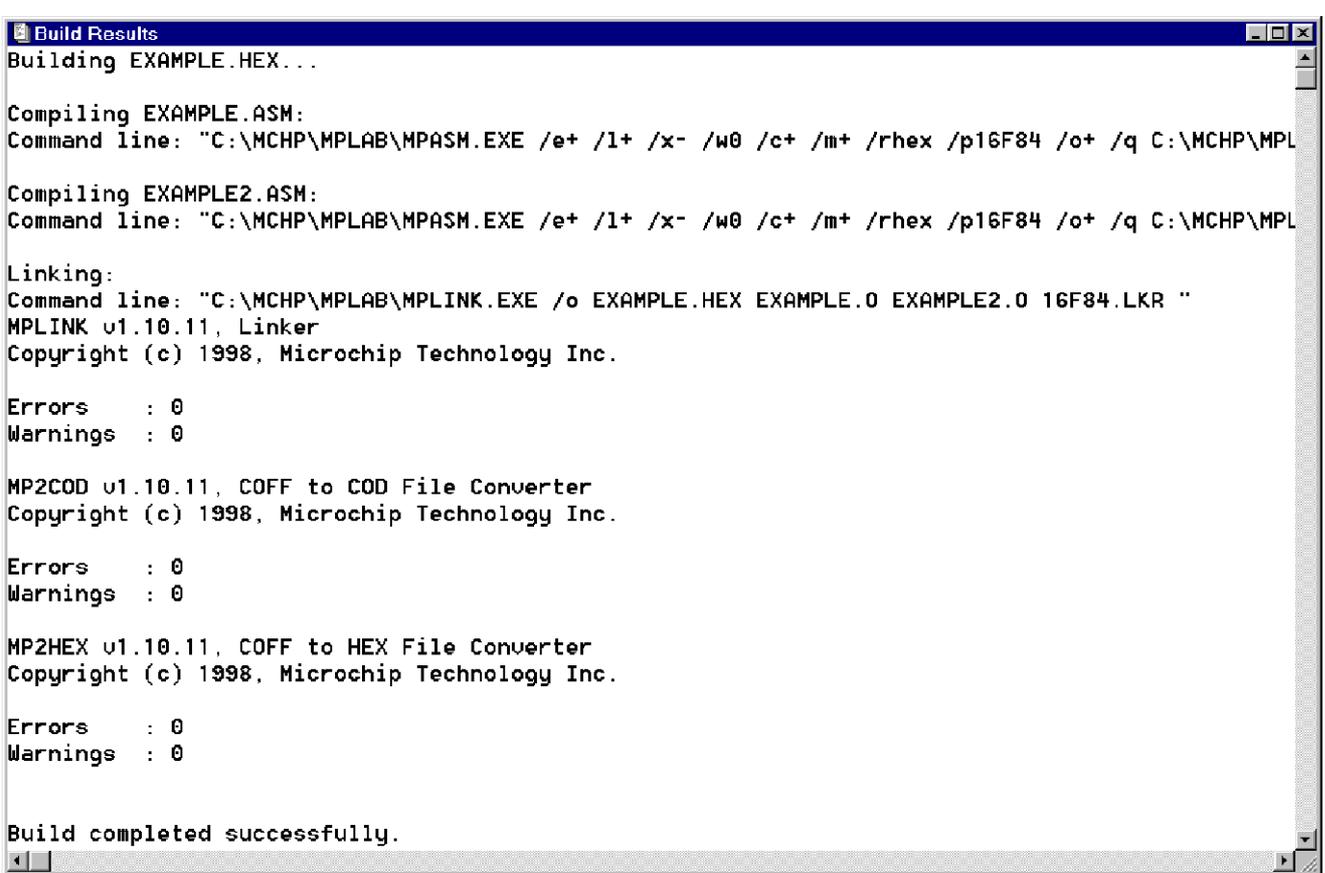
В этом простом примере не указываются директории размещения файлов в параметрах Path. По мере усложнения проекта Вам может понадобиться указать путь к подключаемым файлам, библиотекам и файлам сценария линкера.

Параметры проекта по умолчанию (при создании нового проекта) можно настроить в меню *Options > Environment Setup* раздел *Projects*.

Нажмите кнопку *OK* для закрытия диалогового окна настройки проекта.

#### 4.6.7 Компиляция

Выберет *Project > Make Project* для компиляции исходных файлов проекта с помощью MPASM и MPLINK. На экране появится окно результатов, с указанием параметров командной строки, используемых при запуске программ.



```
Build Results
Building EXAMPLE.HEX...

Compiling EXAMPLE.ASM:
Command line: "C:\MCHP\MPLAB\MPLINK.EXE /e+ /l+ /x- /w0 /c+ /m+ /rhex /p16F84 /o+ /q C:\MCHP\MPL

Compiling EXAMPLE2.ASM:
Command line: "C:\MCHP\MPLAB\MPLINK.EXE /e+ /l+ /x- /w0 /c+ /m+ /rhex /p16F84 /o+ /q C:\MCHP\MPL

Linking:
Command line: "C:\MCHP\MPLAB\MPLINK.EXE /o EXAMPLE.HEX EXAMPLE.O EXAMPLE2.O 16F84.LKR "
MPLINK v1.10.11, Linker
Copyright (c) 1998, Microchip Technology Inc.

Errors      : 0
Warnings    : 0

MP2COD v1.10.11, COFF to COD File Converter
Copyright (c) 1998, Microchip Technology Inc.

Errors      : 0
Warnings    : 0

MP2HEX v1.10.11, COFF to HEX File Converter
Copyright (c) 1998, Microchip Technology Inc.

Errors      : 0
Warnings    : 0

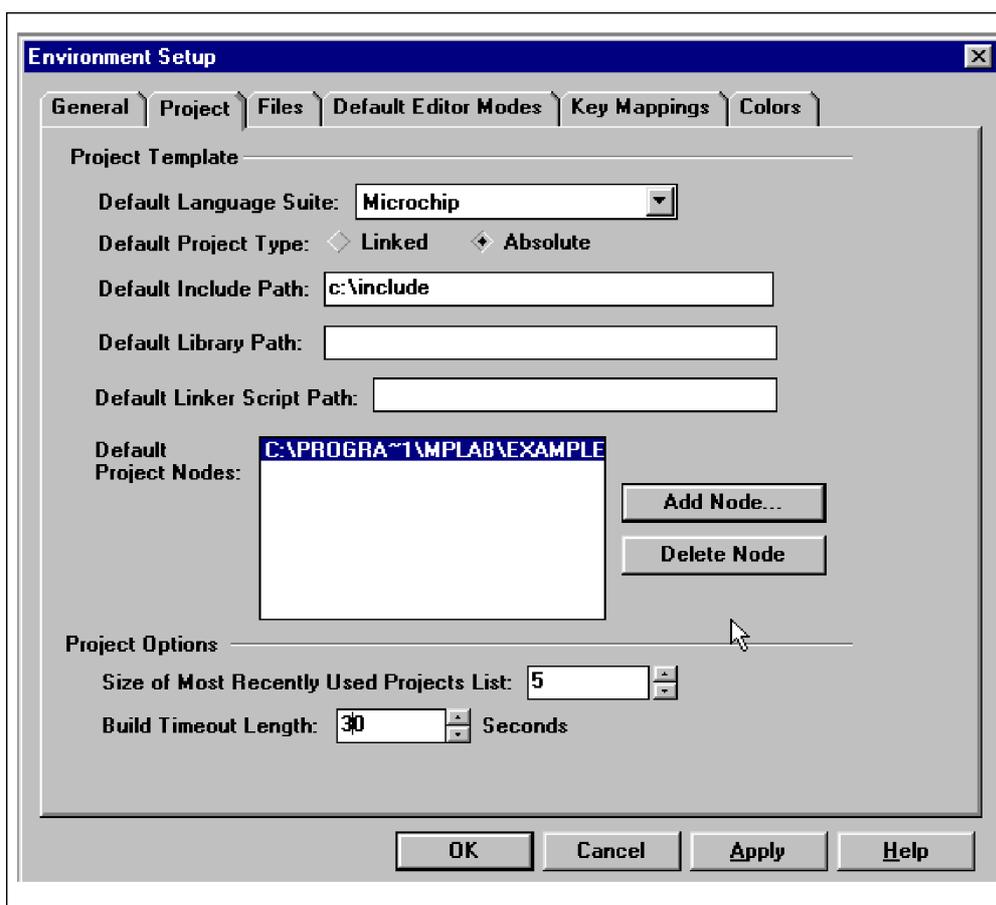
Build completed successfully.
```

## 4.6.8 Поиск и устранение ошибок

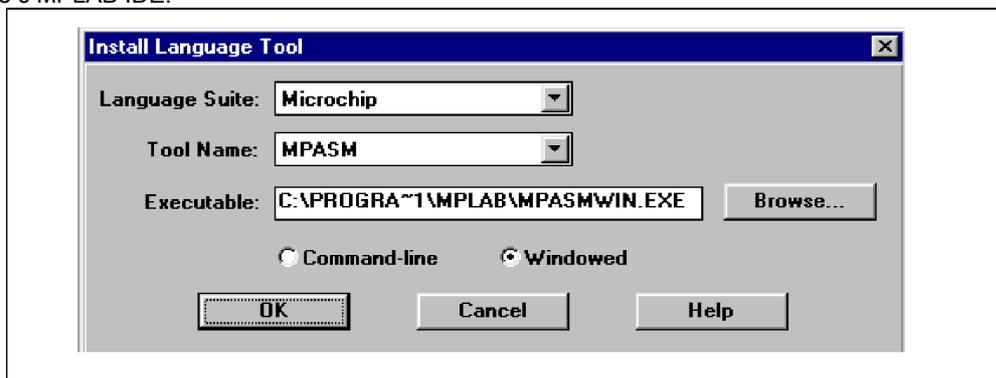
В случае появления ошибок при выполнении компиляции воспользуйтесь следующими рекомендациями:

1. Проверьте наличие синтаксических ошибок. Двойным щелчком «мыши» на сообщении об ошибке курсор будет перемещен в место ошибки.
2. В меню *Project > Edit Project* выберите HEX файл и нажмите кнопку *Node Propertie*, проверьте параметры работы MPLINK. Затем выберите исходные файлы и проверьте параметры компиляции MPASM.
3. В меню *Project > Edit Project* проверьте правильность указания имен файлов. Если вы неправильно указали имя файла, выберите его и нажмите кнопку *Delete Node*. Затем подключите нужный файл как описано в разделе 4.6.5.

Если MPLAB IDE выдаст сообщение об ошибке “Time-out.”, нажмите кнопку *OK*, для продолжения. Ошибка может быть вызвана малой производительностью компьютера. Вы можете настроить интервал времени, в течение которого MPLAB IDE не будет формировать сообщение об ошибке (меню *Options > Environment > Project*). Установите значение параметра *Build Timeout Length* равное 0, если вы хотите, чтобы это сообщение не появлялось.



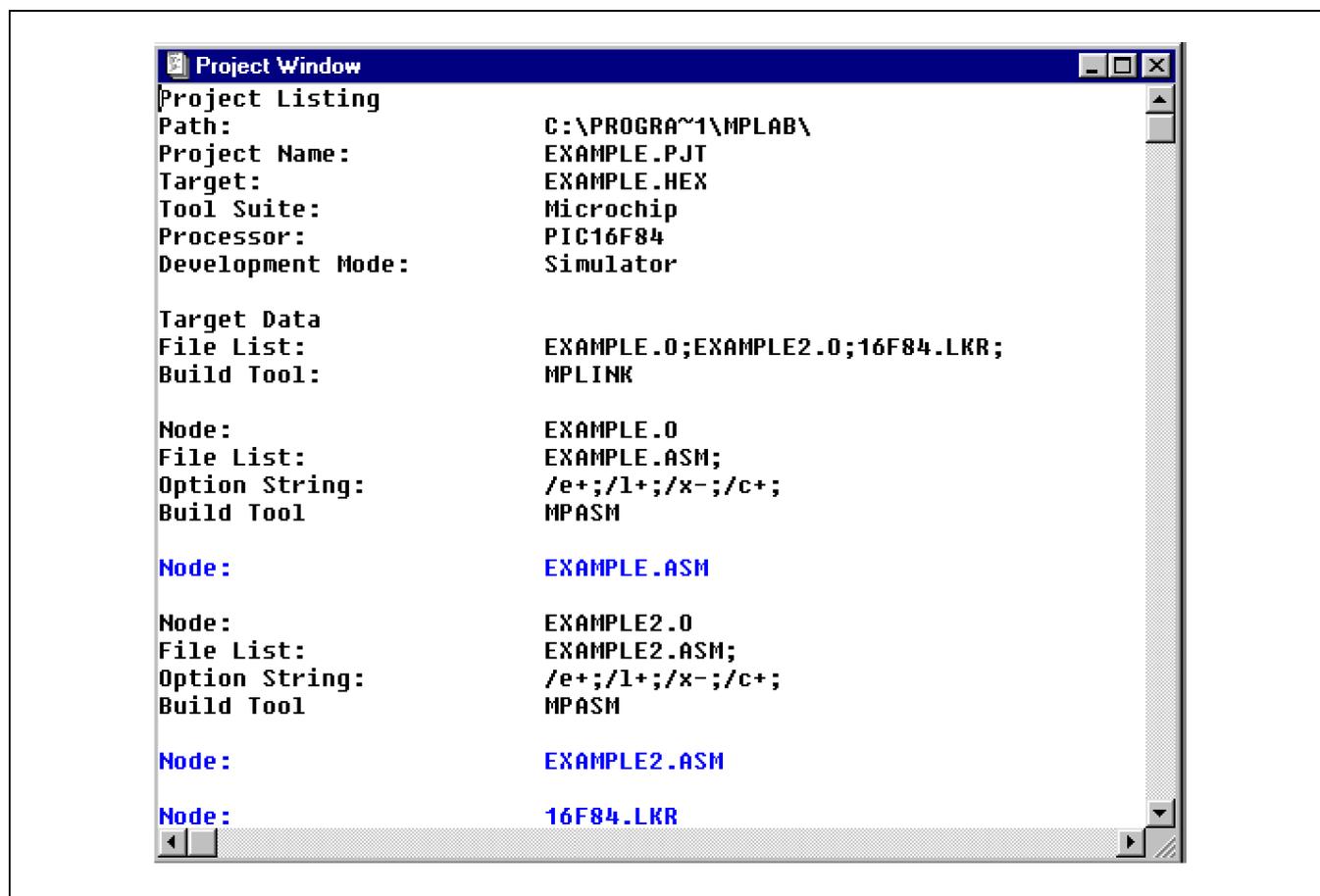
С помощью пункта меню *Project > Install Language Tool...* проверьте, чтобы файл MPASMWIN.EXE находился в директории вместе с MPLAB IDE.



Альтернативой MPASMWIN.EXE может быть программа MPASM.EXE, работающая под управлением операционной системой DOS и запускаемая из командной строки. MPASM.EXE может не работать на компьютерах Pentium 100MHz и выше.

#### 4.6.9 Окно проекта

Для проверки соответствия имен файлов и других параметров работы воспользуйтесь пунктом меню *Window > Project window*. На экране появится окно, показанное на рисунке.



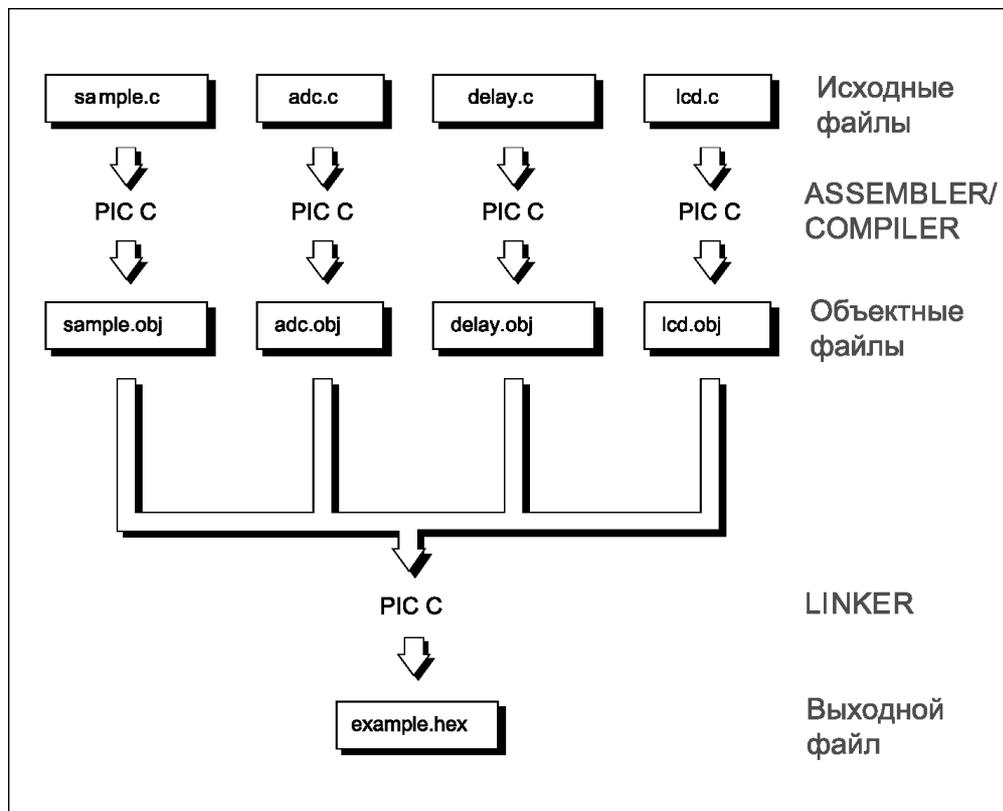
#### 4.6.10 Заключение

Основные шаги создания проекта с несколькими исходными файлами:

1. создать новый проект – *Project > New Project*;
2. указать компилятором MPLINK – *Node Properties*;
3. подключить исходные файлы;
4. подключить файл сценария.

## 4.7 Создание проекта совместно с HI-TECH PIC C

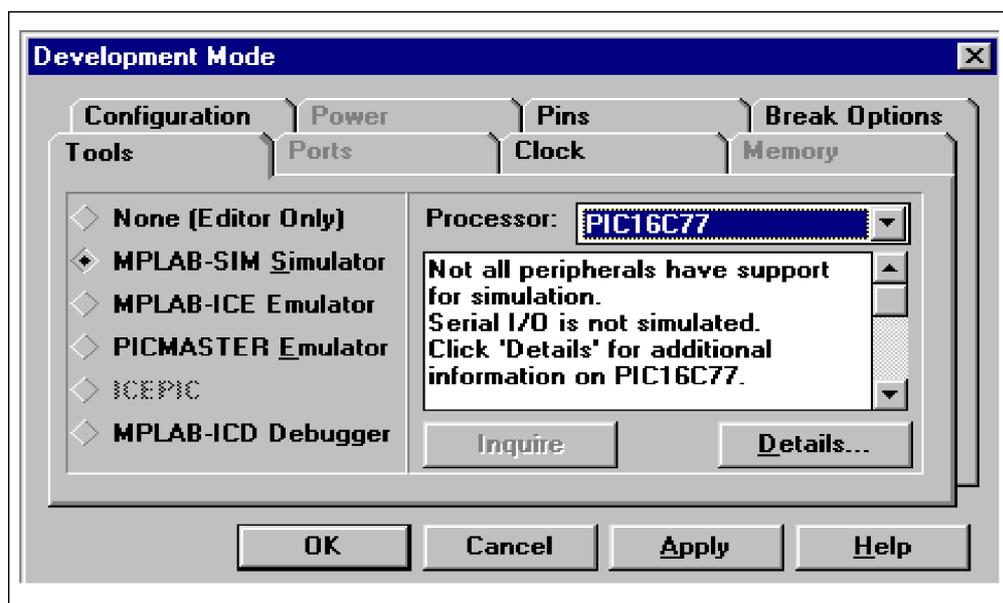
В данном примере будет показано, как использовать компилятор HI-TECH PIC C совместно со средой проектирования MPLAB IDE.



Закройте все открытые проекты - *Project > Close Project*.

### 4.7.1 Настройка параметров среды проектирования

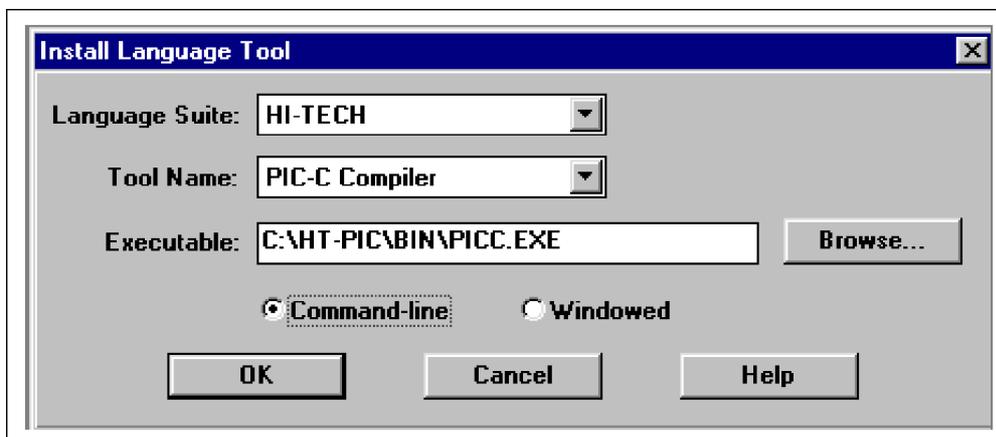
Для примера, выберите симулятор MPLAB SIM и тип микроконтроллера PIC16C77 - *Options > Development Mode > Tools*. Подтвердите выбор, нажав на кнопку *OK*.



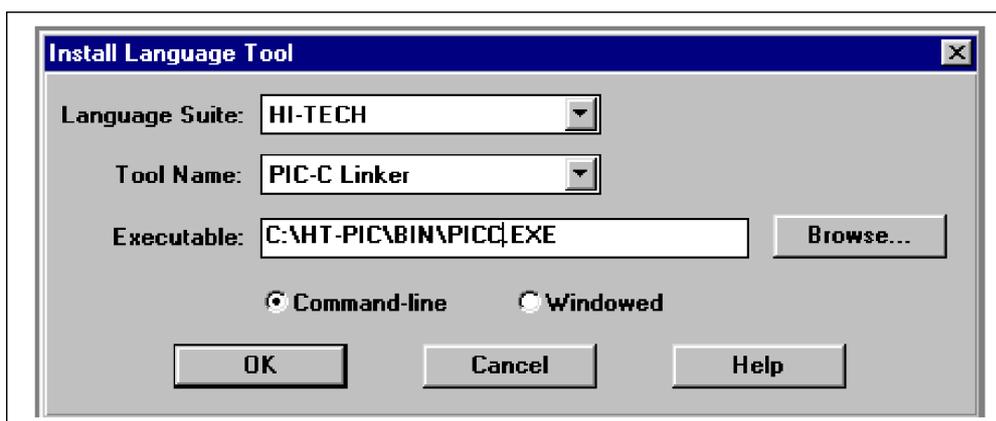
#### 4.7.2. Подключение компилятора к среде проектирования MPLAB IDE

Проверьте правильность подключения компилятора PIC C и его компонентов к MPLAB IDE - *Project > Install Language Tool*. Путь к программам HI-TECH, установленным на вашем компьютере, может отличаться от указанных этих в примерах.

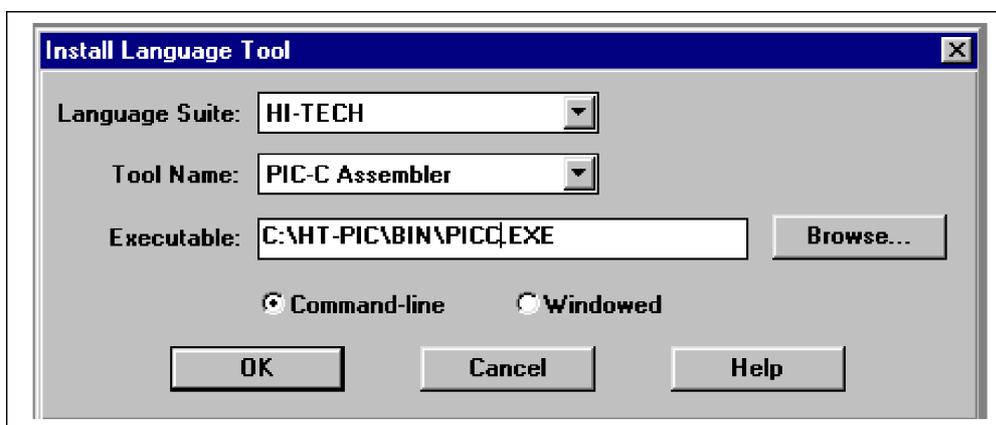
Диалоговое окно настройки PIC C компилятора.



Диалоговое окно настройки PIC C линкера.



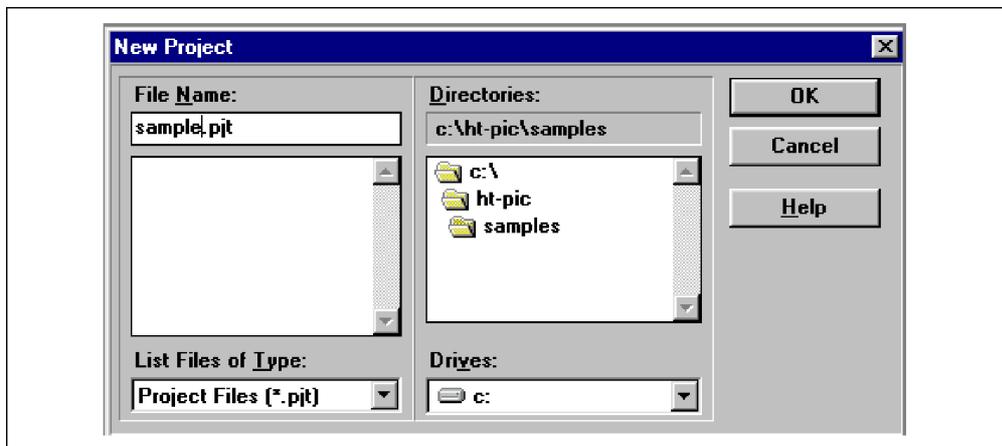
Диалоговое окно настройки PIC C ассемблера.



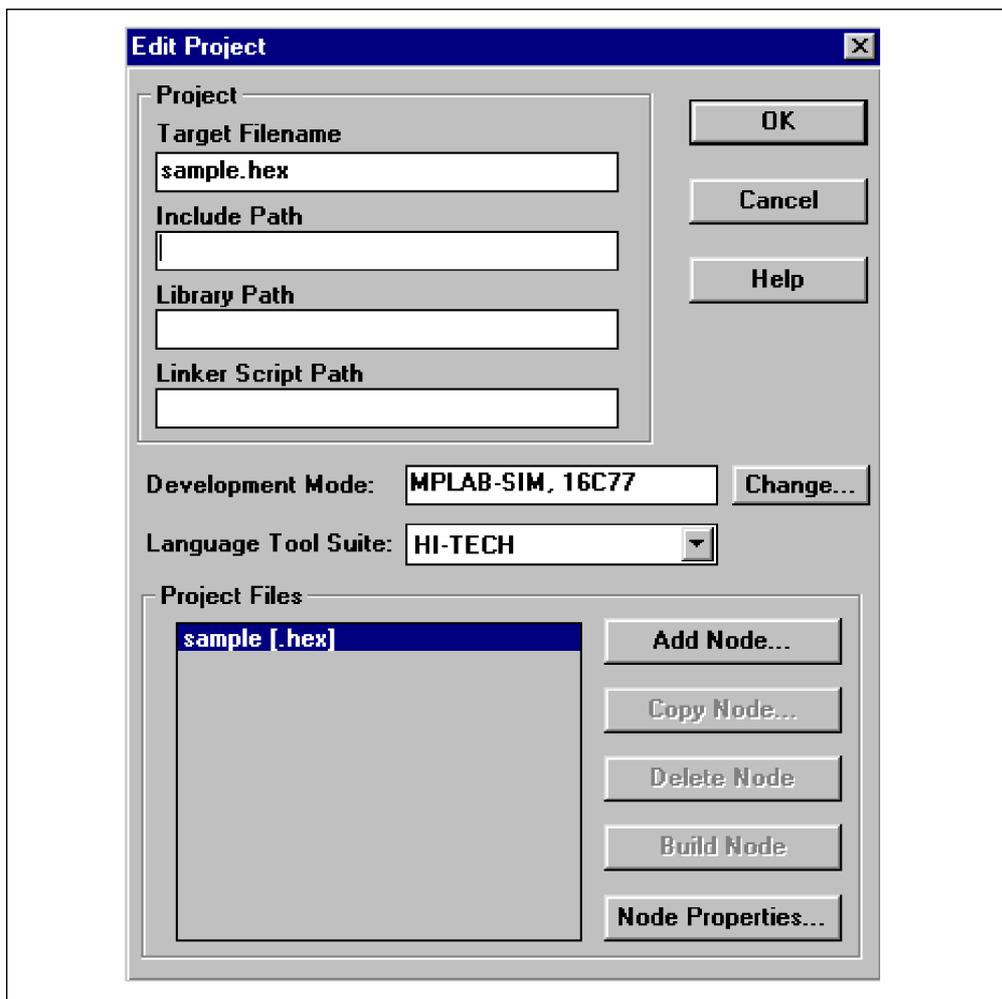
С помощью кнопки *Browse* в качестве компилятора, линкера и ассемблера укажите файл PICC.EXE.

### 4.7.3 Создание нового проекта

Создайте новый проект - *Project > New Project*, указав директорию размещения и имя. Для примера, создайте новый проект в директории \HT-PIC\SAMPLES с именем sample.pjt.



После указания директории и имени проекта, на экране появится диалоговое окно настройки параметров.

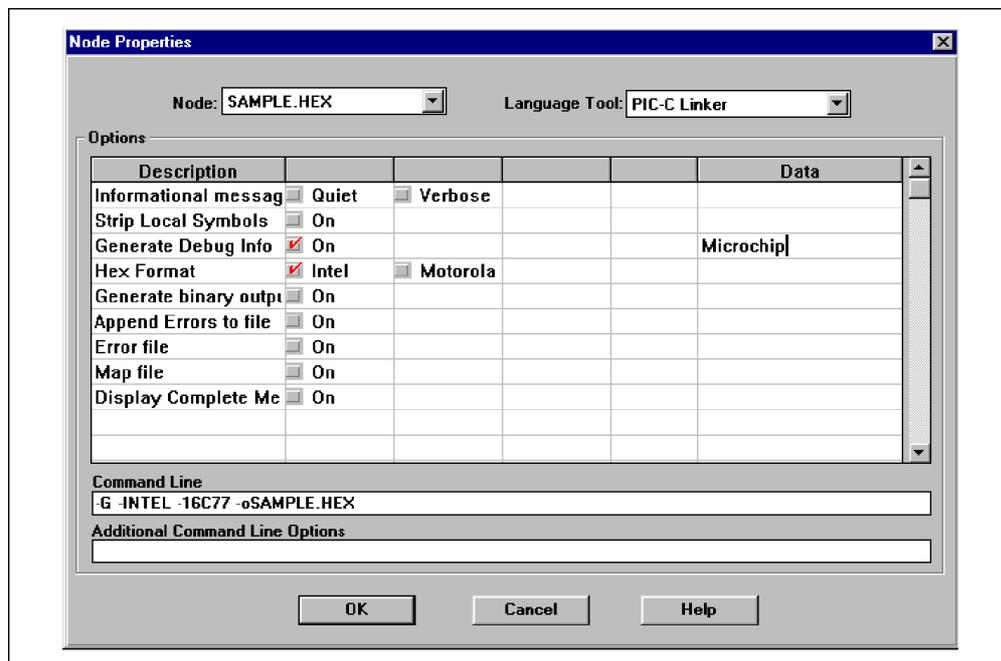


Проверьте, что инструмент HI TECH подключен.

**Примечание.** Убедитесь, что Вы указали правильно размещение компилятора. Если путь к файлам был указан неправильно, то на экране появится предупреждение сообщающее, что исходный файл не удалось открыть.

#### 4.7.4 Настройка параметров компиляции

В диалоговом окне настройки проекта нажмите кнопку *Node Properties*. Установите PIC-C Linker в строке Language Tool. В пункте Generate Debug Info в столбце Data впишите слово Microchip. Проверьте параметры настройки.



В диалоговом окне содержатся пункты параметров работы PIC C. При первом открытии этого окна в нем будут указаны параметры по умолчанию. Для данного примера необходимо внести только указанные выше изменения.

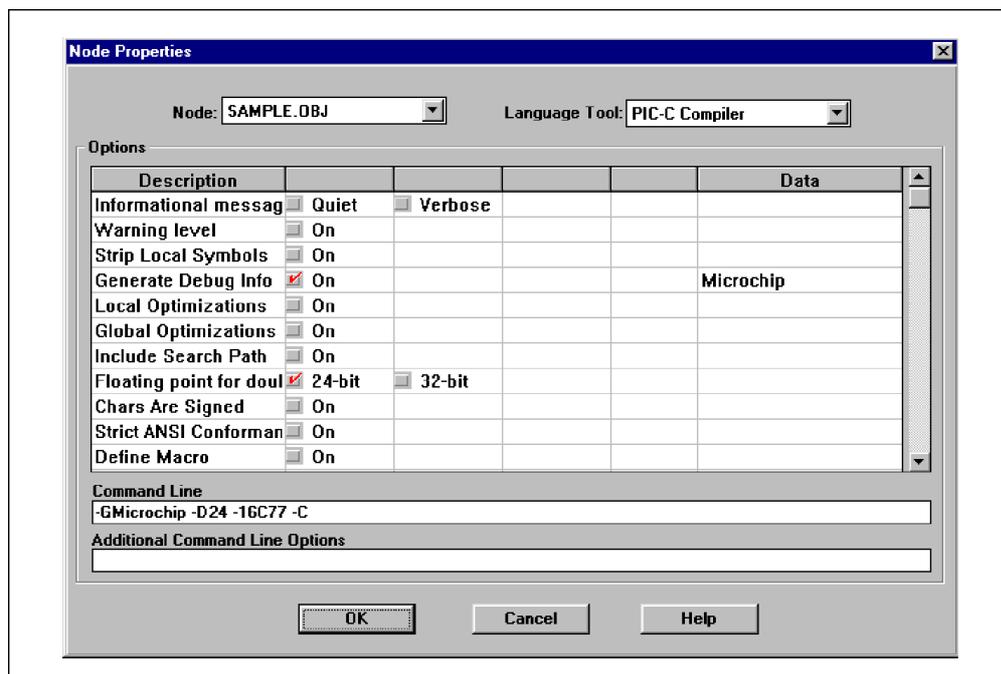
Обратитесь к технической документации HI TECH для получения дополнительной информации. Нажмите кнопку *OK* для возвращения в диалоговое окно настройки параметров проекта.

#### 4.7.5 Подключение исходных файлов

Нажмите кнопку *Add Node* и укажите исходный файл sample.c в директории \HT-PIC\SAMPLES.

Когда файл будет добавлен, выберите его и нажмите кнопку *Node Properties* для настройки параметров компилятора:

- укажите в строке Language Tool для файла sample.obj компилятор PIC C;
- выберите пункт Generate Debug Info;
- в пункте Generate Debug Info в столбце Data впишите слово Microchip.



Имя файла `sample.obj` записывается автоматически.

В диалоговом окне содержатся пункты параметров работы PIC C. При первом открытии этого окна в нем будут указаны параметры по умолчанию. Для данного примера необходимо внести только указанные выше изменения.

Обратитесь к технической документации HI TECH для получения дополнительной информации.

Нажмите кнопку **OK** для возвращения в диалоговое окно настройки параметров проекта.

Выберите файл `sample.c`, нажмите кнопку **Copy Node** и добавьте файлы `adc.c`, `delay.c` и `lcd.c`. Параметры компиляции файла `sample.c` будут назначены на подключаемые файлы.

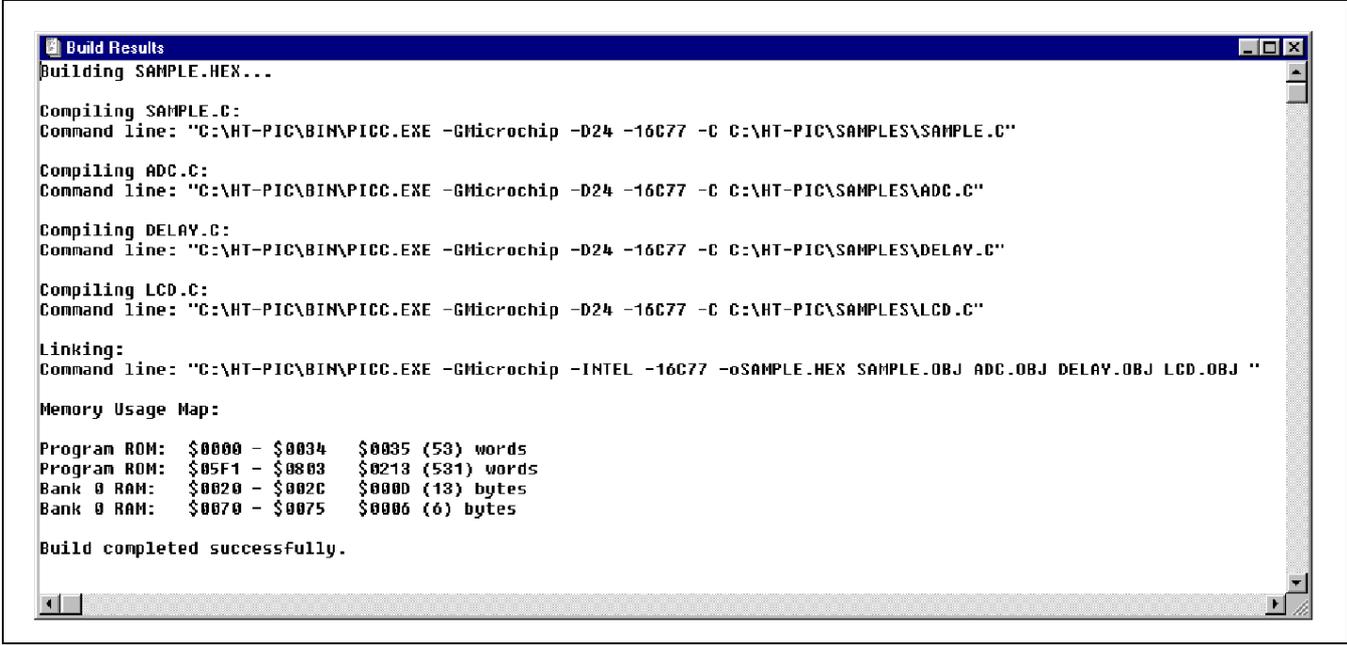
После подключения всех файлов диалоговое окно настройки проекта должно выглядеть, как показано на рисунке.



Параметры проекта по умолчанию (устанавливаемые при создании нового проекта) можно настроить в меню *Options > Environment Setup* раздел *Projects*.

## 4.7.6 Компиляция

Выберите *Project > Make Project* для начала компиляции с использованием Hi-TECH линкера и компилятора. На экране появятся окно результатов с указанием параметров командной строки вызываемых файлов, показанное на рисунке.



```
Build Results
Building SAMPLE.HEX...

Compiling SAMPLE.C:
Command line: "C:\HT-PIC\BIN\PICC.EXE -GMicrochip -D24 -16C77 -C C:\HT-PIC\SAMPLES\SAMPLE.C"

Compiling ADC.C:
Command line: "C:\HT-PIC\BIN\PICC.EXE -GMicrochip -D24 -16C77 -C C:\HT-PIC\SAMPLES\ADC.C"

Compiling DELAY.C:
Command line: "C:\HT-PIC\BIN\PICC.EXE -GMicrochip -D24 -16C77 -C C:\HT-PIC\SAMPLES\DELAY.C"

Compiling LCD.C:
Command line: "C:\HT-PIC\BIN\PICC.EXE -GMicrochip -D24 -16C77 -C C:\HT-PIC\SAMPLES\LCD.C"

Linking:
Command line: "C:\HT-PIC\BIN\PICC.EXE -GMicrochip -INTEL -16C77 -oSAMPLE.HEX SAMPLE.OBJ ADC.OBJ DELAY.OBJ LCD.OBJ "

Memory Usage Map:

Program ROM: $0000 - $0034 $0035 (53) words
Program ROM: $05F1 - $0803 $0213 (531) words
Bank 0 RAM: $0020 - $002C $0000 (13) bytes
Bank 0 RAM: $0070 - $0075 $0006 (6) bytes

Build completed successfully.
```

## 4.7.7 Поиск и устранение ошибок

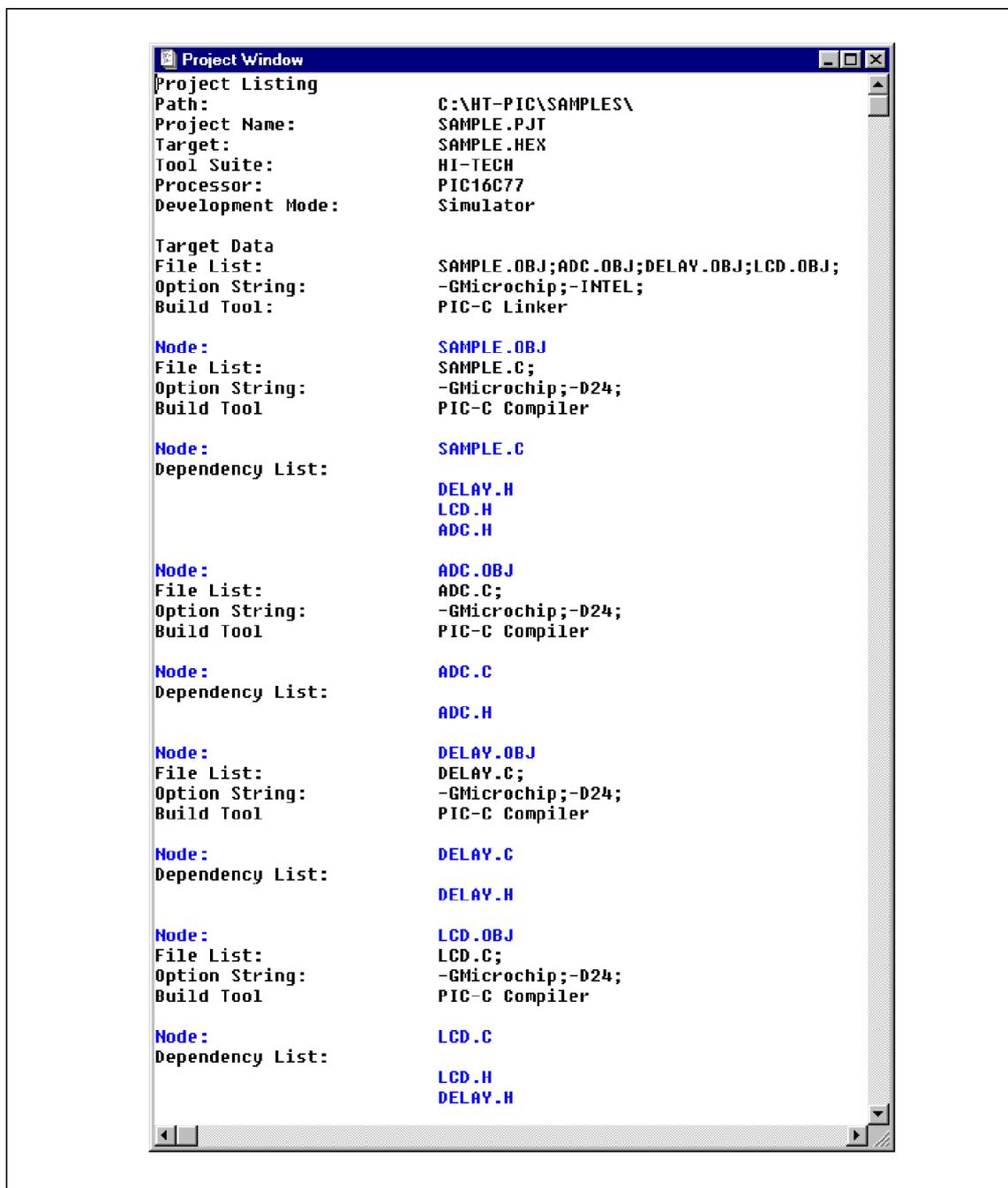
В случае появления ошибок при выполнении компиляции воспользуйтесь следующими рекомендациями:

1. Проверьте наличие синтаксических ошибок. Двойным щелчком «мыши» на сообщении об ошибке курсор будет перемещен в место ошибки.
2. В меню *Project > Edit Project* нажмите кнопку *Node Properties* и проверьте параметры работы компилятора и линкера.
3. В меню *Project > Edit Project* проверьте правильность указания имен файлов. Если вы неправильно указали имя файла, выберите его и нажмите кнопку *Delete Node*. Затем подключите нужный файл как описано в разделе 4.7.5.

С помощью пункта меню *Project > Install Language Tool...* проверьте, чтобы файл PICC.EXE находился в указанной директории и был назначен для линкера и компилятора.

#### 4.7.8 Окно проекта

Для проверки соответствия имен файлов и других параметров работы воспользуйтесь пунктом меню *Window > Project window*. На экране появится окно, показанное на рисунке.



#### 4.7.9 Заключение

Основные шаги создания проекта совместно с HI-TECH PIC C:

1. установить в меню *Project > Install Language Tool* PIC C компилятор, линкер и ассемблер;
2. указать рабочую директорию HI-TECH PIC C (или директорию PIC C, в которой он установлен на вашем компьютере);
3. создать новый проект - *Project > New Project*;
4. включить генерацию информации для отладки проекта;
5. настройте параметры PIC C линкера;
6. подключите исходные файлы к проекту;
7. настройте параметры компиляции исходных файлов;
8. установите формат файла отладки "Microchip" для каждого исходного файла.

#### **4.8 Создание проекта совместно с MPLAB-C17 или MPLAB-C18**

Информацию об использовании MPLAB-C17 или MPLAB-C18 совместно с MPLAB IDE смотрите в технической документации MPLAB-CXX User's Guide (DS51217).

## Глава 5. Редактор MPLAB

### 5.1 Введение

В этой главе будут рассмотрены основные принципы работы с редактором MPLAB. Редактор интегрирован в среду проектирования MPLAB IDE, поэтому подробное описание функций редактора смотрите в главе 7.

### 5.2 Основные разделы главы

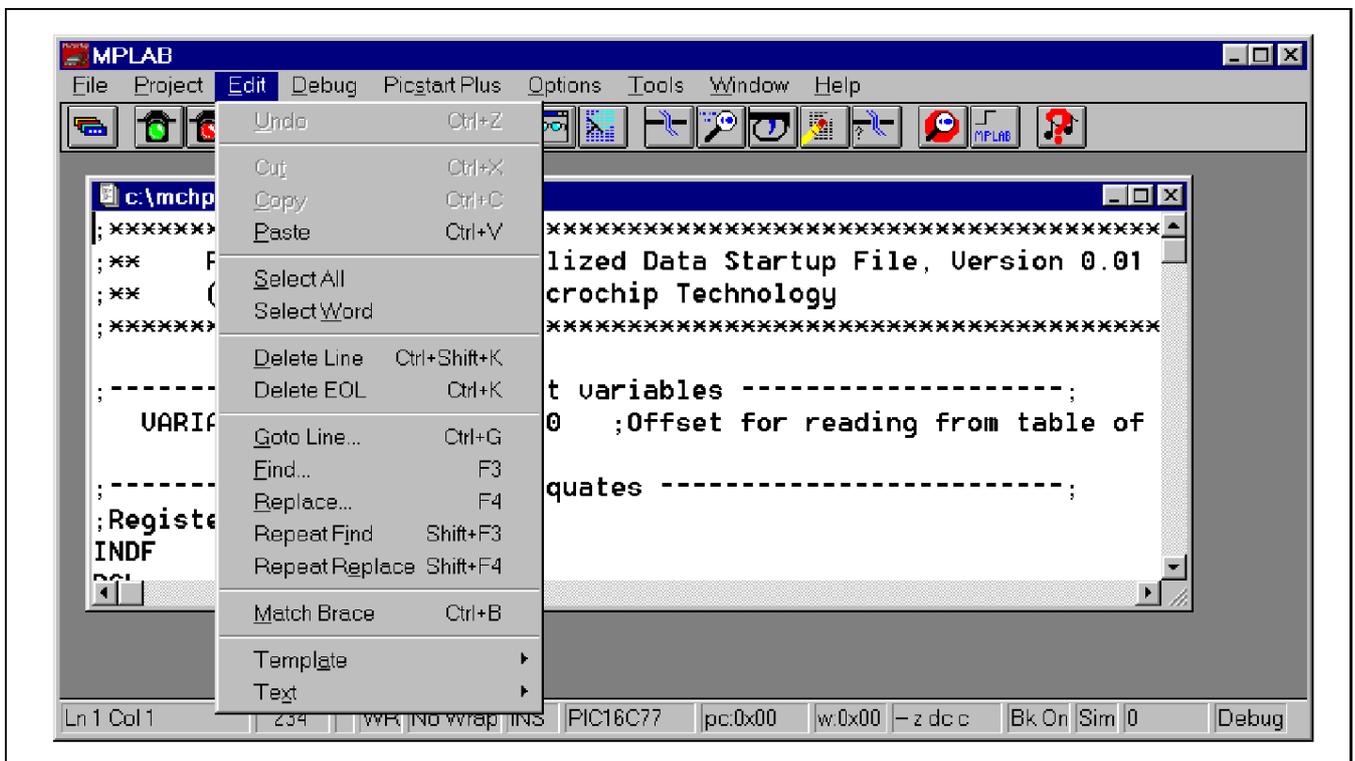
Что такое редактор MPLAB  
 Редактор MPLAB помогает Вам  
 Характеристики редактора MPLAB  
 Функции редактора MPLAB

### 5.3 Что такое редактор MPLAB

MPLAB редактор является неотъемлемой частью среды разработки MPLAB IDE. Редактор доступен для работы даже в то время, когда выполняется симуляция или эмуляция работы программы микроконтроллера. Редактор не является отдельно выполняемым файлом, а представляет набор функций в интегрированной среде разработки MPLAB IDE.

### 5.4 Редактор MPLAB помогает Вам

Редактор предоставляет разработчикам РЭА удобную среду написания программ для микроконтроллеров PICmicro.



## 5.5 Характеристики редактора MPLAB

### 5.5.1 Размер файлов

Редактор не имеет никаких ограничений на количество и размер открытых файлов, кроме доступного объема ОЗУ персонального компьютера.

### 5.5.2 Совместимость с операционной системой Windows

Редактор MPLAB поддерживает:

- работу через стандартное или графическое меню;
- использование манипулятора «мышь»;
- работу с буфером clipboard;

### 5.5.3 Определение функций для комбинации кнопок клавиатуры

Пользователь может определять функциональное назначение комбинации кнопок клавиатуры. Допускается включать в комбинацию следующие специальные кнопки Esc, Ctrl, Alt (Например - <Esc+G>, <Ctrl+K>, <Ctrl+B>).

### 5.5.4 Создание файлов

Редактор MPLAB обеспечивает следующие функции:

- определение набора шаблонов со стандартным текстом программы, позволяя их добавлять в текущий файл всего несколькими нажатиями кнопки «мышки»;
- группировка шаблонов в файлы;
- автоматическая загрузка шаблонов.

## 5.6 Функции редактора MPLAB

Основные функции редактора MPLAB:

- работа с файлами;
- работа с шаблонами;
- обработка текста;
- редактирование параметров окна;
- поддержка языка C.

### 5.6.1 Работа с файлами

Редактор MPLAB позволяет сохранять созданные файлы, загружать сохраненные файлы и переписывать файлы, указывая новое имя (подробное описание в главе 7.4). Сохранение файлов выполняется заменой старого файла на диске или созданием нового файла с другим именем.

### 5.6.2 Работа с шаблонами

Как Вы выполняете повторение текста программы при создании новых файлов? Вы можете просто вставить часть текста программы, скопировав его из другого файла. Однако этот процесс не исключает возникновение ошибок при переносе фрагментов текста программы.

MPLAB IDE обеспечивает Вас шаблонами, которые являются предварительно подготовленными текстовыми файлами. Добавляя подготовленные фрагменты программы, взамен повторного ввода, Вы уменьшаете общее время написание программы. Вы можете использовать готовые шаблоны из директории Templates MPLAB IDE или создавать собственные.

Вы можете создавать исходный файл программы (или открыть существующий), добавить текст шаблона, и по специальным маркерам найти области, в которых необходимо сделать изменения в соответствии с требованиями Вашего приложения.

Создание файлов шаблона (.tpl) позволяет, например, объединить фиксированные части программы по функциональному признаку или для каждого устройства в отдельности.

### 5.6.3 Обработка текста

Редактор MPLAB предназначен для написания текста программы, поэтому имеет некоторые особенности, что делает его более полезным при редактировании текста. Дополнительную информацию смотрите в разделе 7.6.

#### 5.6.3.1 Вставка, выбор и удаление текста

Редактор MPLAB позволяет установить два режима вставки: вставка со сдвигом <INS> или вставка с замещением <OVR>. Режим вставки отображается на линейке состояния.

Выбор текста может быть выполнен - отдельного символа, слова целиком или строки целиком. Удаление может выполняться - отдельного символа, строки целиком или от курсора до конца строки. Вы можете также использовать функцию поиска и замены текста или специального символа.

#### 5.6.3.2 Глубина текста

При редактировании программы очень удобно использовать функцию выравнивания текста. При переводе курсора редактор MPLAB автоматически устанавливает глубину текста следующей строки равную предыдущей строки.

#### 5.6.3.3 Чувствительность к регистру

Редактор MPLAB позволяет изменять регистр (верхний или нижний) выбранного фрагмента текста.

#### 5.6.3.4 Обработка скобок

Редактор MPLAB позволяет управлять характерными знаками (фигурной или круглой скобкой) для разграничения программы или текста.

#### 5.6.3.5 Отмена действия

Редактор MPLAB запоминает все выполненные действия и может их отменить в обратном порядке.

#### 5.6.3.6 Автоматическое форматирование текста

Иногда бывает удобно, чтобы была возможность ограничивать число символов в строке. Включение автоматического форматирования текста выполняется двойным щелчком левой кнопки «мыши» на области форматирования в линейке статуса. Если указано значение «No Wrap», то форматирование текста выключено. Например, значение «Wr 72» указывает, что форматирование текста включено и максимальная длина линии 72.

Параметры форматирования текста определяются для каждого окна в отдельности.

##### 5.6.3.6.1 Тип языка программирования «(none)» или «C»

Редактор MPLAB выполняет перевод курсора на следующую строку в определенную позицию на ближайшем символе пробела или тире.

##### 5.6.3.6.2 Тип языка программирования «TeXt»

Редактор MPLAB выполняет перевод курсора на следующую строку в определенную позицию на ближайшем символе пробела.

**Примечание.** Редактор MPLAB выполняет ограничение длины строки, только если курсор находится в конце строки. Если Вы установите курсор в середину строки и начнете вводить текст, редактор MPLAB не будет выполнять ограничение длины строки, даже если ее длина вышла за установленное значение.

### 5.6.4 Редактирование параметров окна

Параметры редактирования текста определяются для каждого рабочего окна в отдельности.

Редактор MPLAB поддерживает следующие настройки: формат окна, режим отображения, ввод данных, печать и режим файла. Дополнительную информацию смотрите в разделах 7.9.3 и 7.9.4.

## 5.6.5 Поддержка языка C

При редактировании текста программы, написанной на языке C, редактор MPLAB контролирует наличие специальных символов:

Характерный знак «#» всегда перемещает на пустую строку в первую колонку.

Закрывающаяся фигурная скобка «}» будет расположена на пустой строке в той же колонке, что и соответствующая открывающаяся «{».

Например:

```
//
*****
// EXAMPLE.C
//
*****
#include <PIC16C84.H>
void delay(void);
void main(void)
{
    unsigned int i,j;
    TRISB = 0xff;
    PORTB = 0;
    i = 0x1;
    while(1)
    {
        PORTB = i;
        if (i == 0x80)
            i = 0x1;
        else
            i <<= 1;
        TRISB = 0;
        delay();
        TRISB = 0xff;
        delay();
    }
}
void delay(void)
{
    int x, y;
    x = 0x3f;
    y = 0xff;
    while(x--)
    {
        while(y--)
            NOP();
    }
}
```

## Глава 6. Отладчик и симулятор MPLAB IDE

### 6.1 Введение

В этой главе будут рассмотрены вопросы отладки программы с использованием симулятора (MPLAB SIM) или эмулятора (MPLAB-ICE, MPLAB-ICD, PICMASTER или ICEPIC). Описание работы с эмулятором MPLAB-ICE смотрите в оригинальной технической документации MPLAB-ICE User's Guide.

### 6.2 Основные разделы главы

- Функции отладчика MPLAB IDE
- Выполнение программы в реальном масштабе времени
- Среда симулятора MPLAB SIM
- Характеристики симулятора MPLAB SIM
- Точки остановки и трассировки
- Точки остановки по условию
- Функции стимула
- Симуляция 12-разрядных микроконтроллеров
- Симуляция 14-разрядных микроконтроллеров
- Симуляция 16-разрядных микроконтроллеров PIC17CXXX
- Симуляция 16-разрядных микроконтроллеров PIC18CXXX

### 6.3 Функции отладчика MPLAB IDE

Поле ввода и компиляции проекта в MPLAB IDE Вы можете посмотреть, как выполняется программа. Одним из методов может быть программирование микроконтроллера и установка его в устройство, чтобы проверить работу программы. Как правило, с первого раза новая программа работать будет неправильно и Вам потребуется отладка текста программы.

Отладку теста программы можно выполнить с помощью симулятора MPLAB SIM или эмулятора MPLAB-ICE для управления устройством, в то время как вы отлаживаете программу.

В любом случае Вы будете останавливать ход выполнения программы для проверки значения регистров общего и специального назначения и состояния процессора.

Эмулятор MPLAB-ICE управляет работой устройства с фактической скоростью выполнения программы (в реальном масштабе времени) и останавливается только в указанных точках. MPLAB SIM моделирует выполнение программы любого типа PICmicro с учетом состояния портов ввода/вывода в скоростях, которые зависят от быстродействия персонального компьютера.

Эмулятор и симулятор поддерживают следующие функции:

- эмуляция памяти программ;
- прерывание выполнения программы в точках остановки;
- работа по шагам;
- контроль регистров общего и специального назначения.

Все указанные функции используют данные из проекта MPLAB IDE. Символьные метки могут использоваться для указания точек остановки и трассировки, а так же изменения значений регистров памяти данных.

### 6.4 Выполнение программы в реальном масштабе времени

Обычно термин «в реальном масштабе времени» применим только при работе с эмулятором MPLAB ICE или отладчиком MPLAB-ICD.

#### 6.4.1 Использование симулятора MPLAB SIM

В режиме симулятора инструкции выполняются с той скоростью, которую позволяют вычислительные мощности персонального компьютера. Скорость выполнения программы в симуляторе значительно медленнее, чем при реальной работе, поскольку симулятор должен следить за состоянием портов ввода/вывода, значением регистров памяти данных общего и служебного назначения, учитывать точки остановки и трассировки. Так же скорость выполнения программы в симуляторе зависит от числа программ запущенных в фоновом режиме на вашем компьютере.

**Примечание.** В реальной программе могут использоваться пустые циклы для выдержки некоторого интервала времени. Для увеличения скорости выполнения программы в симуляторе рекомендуется не использовать пустые циклы.

Вообще, когда говорят «режим реального времени» для симуляции работы программы, моделирование программы PICmicro происходит с той скоростью, с которой может позволить Ваш персональный компьютер.

## 6.4.2 Режим анимации

В эмуляторе и симуляторе предусмотрена автоматическая работа по шагам. Программа выполняется непрерывно, но значительно медленнее с обновлением всех значений в открытых окнах (окно регистров специального назначения, окно памяти данных, окно с переменными, окно секундомера) после исполнения каждой инструкции.

## 6.5 Среда симулятора MPLAB SIM

Дискретный симулятор MPLAB SIM встроен в среду разработки MPLAB IDE и предназначен для следующих семейств микроконтроллеров PICmicro:

PIC12CXXX  
PIC14000  
PIC16C5X  
PIC16CXX  
PIC16FXXX  
PIC17CXXX  
PIC18CXXX

Дискретный симулятор помогает разработчику в отладке программы для микроконтроллеров PICmicro с возможностью быстрого изменения текста программы и повторного выполнения, подключения внешних стимулов для моделируемого процесса, контроля выполнения программы.

Существует три основных отличия симулятора от эмулятора:

- временные параметры внешних сигналов на портах ввода/вывода;
- скорость выполнения;
- стоимость.

### 6.5.1 Симуляция портов ввода/вывода

Симуляция состояния портов ввода/вывода с дискретностью один машинный цикл. Импульсные сигналы длительностью меньше одного машинного цикла не моделируются, но могут быть обработаны при использовании эмулятора.

**Примечание.** Внешний стимул, указанный MPLAB SIM, будет выполнен в следующем машинном цикле.

### 6.5.2 Скорость выполнения

Максимальная скорость выполнения инструкций определяется производительностью персонального компьютера. Но симулятор позволяет программно установить время выполнения инструкций с точностью до миллисекунд, для более удобного контроля.

### 6.5.3 Стоимость

MPLAB SIM симулятор интегрирован в среду разработки MPLAB IDE, распространяется бесплатно и не требует никаких аппаратных средств. MPLAB SIM используется для нахождения и исправления большинства ошибок в программе.

### 6.5.4 Инструмент отладки

MPLAB SIM наиболее подходящий инструмент для оптимизации алгоритмов. В отличие от некоторых эмуляторов, симулятор MPLAB SIM позволяет контролировать все внутренние регистры микроконтроллера. MPLAB SIM может использоваться для полной отладки программы, если Вы не сталкиваетесь с необходимостью обработки состояния портов ввода/вывода или работы с периферийными модулями в реальном масштабе времени.

## 6.6 Характеристики симулятора MPLAB SIM

MPLAB SIM выполняет контроль и управление состоянием портов ввода/вывода на границах инструкций с периодичностью  $T_{CY}$ .  $T_{CY}$  – время выполнения одной инструкции равно  $4 T_{OSC}$ , где  $T_{OSC}$  период тактового генератора микроконтроллера. Поэтому некоторые физические события не могут быть смоделированы точно:

- асинхронные сигналы (относительно тактового генератора микроконтроллера);
- сигналы, которые имеют период меньше  $T_{CY}$ .

Все внешние события синхронизируются к границам цикла команд микроконтроллера, поэтому событие длительностью меньше одного машинного цикла смоделировано быть не может.

Следующие ограничения распространяются на симуляцию портов ввода/вывода и периферийных модулей PICmicro из-за синхронизации событий к границе цикла инструкции:

- тактовые импульсы таймеров не могут быть меньше времени  $T_{CY}$ , хотя предделитель способен обрабатывать импульсы короче времени выполнения одной инструкции;
- ШИМ импульсы меньше  $T_{CY}$  не поддерживаются;
- не поддерживается 8-разрядное сравнение;
- асинхронные сигналы (относительно тактового генератора микроконтроллера) не моделируются;
- форму сигнала генератора на выводах RC0/RC1 описывать нельзя;
- MPLAB SIM не моделирует работу последовательных интерфейсов.

## 6.7 Точки останковки и трассировки

Функции отладки, влияющие на выполнение инструкций программы, основаны на следующих элементах:

- точки останковки;
- точки трассировки;
- назначение счетчика проходов.

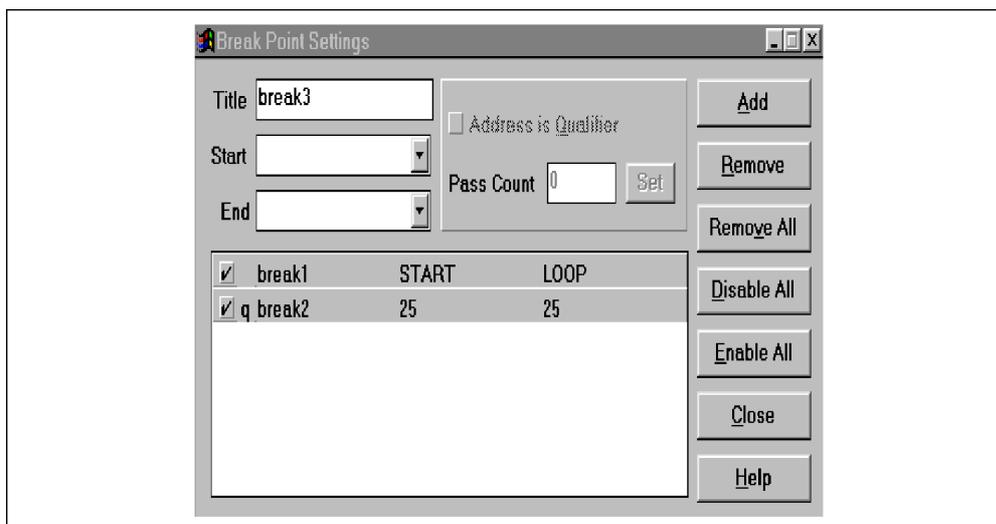
Максимальное число именованных адресов точек каждого типа 16. Точки трассировки и останковки полностью независимы друг от друга и могут быть установлены в любой части программы.

На следующих рисунках показаны диалоговые окна точек останковки и трассировки.

Настройка точек останковки – *Debug > Break Settings*.

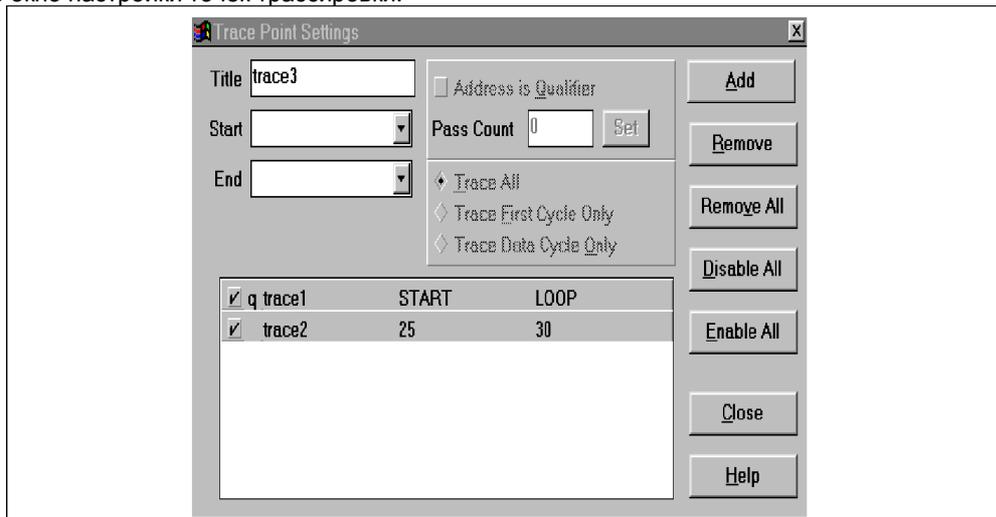
Настройка точек трассировки - *Debug > Trace Settings*.

Диалоговое окно настройки точек останковки.



**Примечание.** MPLAB ICD позволяет установить только одну точку останковки.

Диалоговое окно настройки точек трассировки.



**Примечание.** Диалог настройки точек трассировки не доступны для MBLAB ICE и MPLAB ICD.

## 6.7.1 Точки остановки

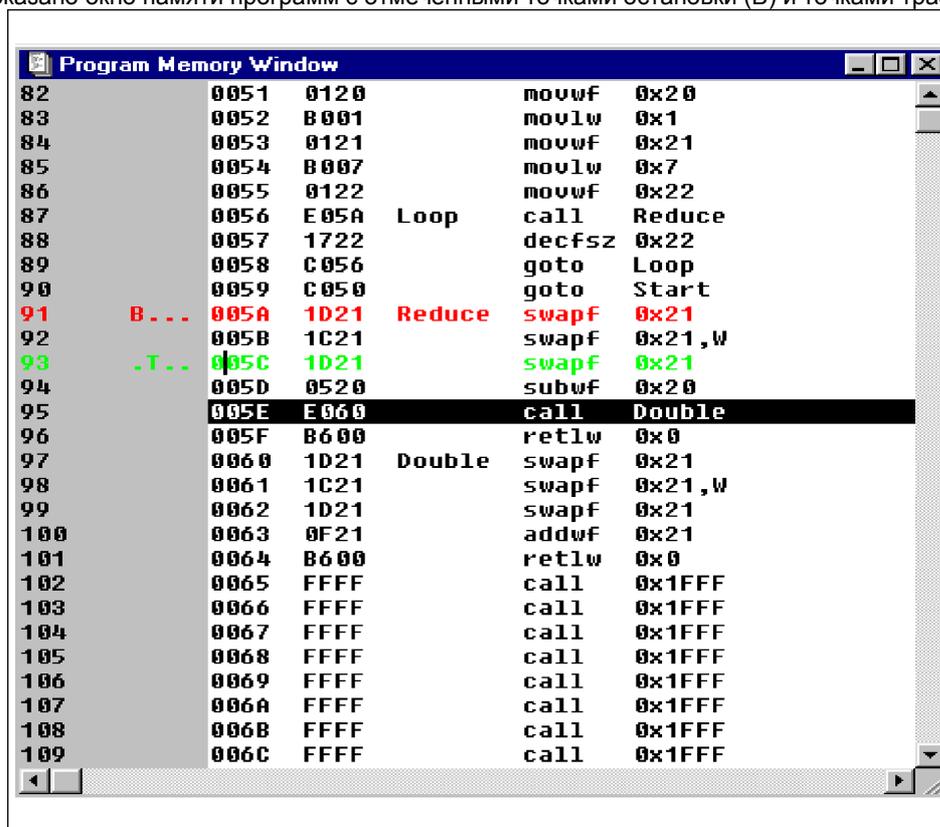
Точка остановки – условие, при котором микроконтроллер прекращает выполнение программы.

**Примечание.** Если выполнение программы не прерывается в точке остановки, проверьте глобальное разрешение остановки программы *Global Break Enable* в пункте меню *Options >Development Mode - Break Options*.

MPLAB IDE обеспечивает следующие условия остановки:

- при совпадении адреса точки остановки с программным счетчиком PC;
- при полном буфере трассировки;
- выполнено указанное число проходов точки остановки;
- переполнение стека;
- переполнение счетчика сторожевого таймера WDT;
- остановка пользователем.

На рисунке показано окно памяти программ с отмеченными точками остановки (B) и точками трассировки (T).



### 6.7.1.1 Остановка при совпадении адреса точки остановки со счетчиком команд PC

Прерывает ход выполнения программы на инструкции, адрес которой соответствует адресу точки остановки. Например, если точка остановки назначена на адрес 5Ah, то остановка будет выполнена на адресе 5Ah до выполнения инструкции.

### 6.7.1.2 Остановка при полном буфере трассировки

MPLAB IDE может быть настроен таким образом, чтобы остановить ход выполнения программы при заполнении выделенного 8Кбайтного буфера трассировки.

### 6.7.1.3 Остановка, после выполнения указанного числа проходов

MPLAB IDE позволяет указывать число проходов точки остановки. Счетчик проходов используется для того, чтобы остановить выполнение программы после определенного числа проходов точки остановки. Например, если разрешен счет числа проходов, то программа будет приостановлена в точки остановки, когда значение счетчика обнулится.

### 6.7.1.4 Остановка при переполнении стека

Когда происходит переполнение аппаратного стека микроконтроллера, MPLAB IDE останавливает ход выполнения программы.

### 6.7.1.5 Остановка при переполнении счетчика сторожевого таймера WDT

MPLAB IDE может быть настроен таким образом, что при выполнении сброса микроконтроллера от сторожевого таймера выполнение программы будет остановлено.

### 6.7.1.6 Остановка пользователем

Пользователь может остановить выполнение программы следующими действиями:

- выбрать пункт меню *Debug > Run > Halt*;
- нажать кнопку <F5> на клавиатуре;
- нажать кнопку остановки на графическом меню MPLAB IDE.

## 6.7.2 Точки трассировки

Дополнительным средством контроля, за ходом выполнения программы, являются точки трассировки. Симулятор MPLAB SIM поддерживает 8Кб-атный буфер трассировки, в котором сохраняются данные о состоянии микроконтроллера. Допускается запись в буфер при переполнении, удаляя более старые значения (если не выбран параметр остановки программы при переполнении буфера).

### 6.7.2.1 Буфер трассировки

MPLAB IDE непрерывно фиксирует состояние микроконтроллера в указанных точках трассировки. Структура информации, записываемой в буфер:

- 16-разрядный адрес точки трассировки;
- 16-разрядный код и данные инструкции;
- время и измененные регистры.

Окно буфера трассировки

Address	Instruction	Comment	Time
1	CALL	Reduce ; Perform the inner portion of	117608.00us
2	0056 E05A Loop	call Reduce ; 117608.00us 58804	
3	DECFSZ	OuterLoop,f ; the loop.	117612.00us
4	0057 1722	decfsz 0x22 ; 117612.00us 58806  [ 57 ]	
5	GOTO	Loop	117642.00us
6	0058 C056	goto Loop ; 117642.00us 58821  22:01 04:I	
7	CALL	Reduce ; Perform the inner portion of	117646.00us
8	0056 E05A Loop	call Reduce ; 117646.00us 58823	
9	DECFSZ	OuterLoop,f ; the loop.	117650.00us
10	0057 1722	decfsz 0x22 ; 117650.00us 58825  [ 57 ]	
11	GOTO	Loop	117682.00us
12	0058 C056	goto Loop ; 117682.00us 58841  22:00 04:I	
13	CALL	Reduce ; Perform the inner portion of	117686.00us
14	0056 E05A Loop	call Reduce ; 117686.00us 58843	
15	DECFSZ	OuterLoop,f ; the loop.	117688.00us
16	0057 1722	decfsz 0x22 ; 117688.00us 58844  0A:FF	
17	GOTO	Loop	117690.00us
18	0058 C056	goto Loop ; 117690.00us 58845  20:FF 04:I	
19	CALL	Reduce ; Perform the inner portion of	117692.00us
20	0056 E05A Loop	call Reduce ; 117692.00us 58846  0A:01	
21	DECFSZ	OuterLoop,f ; the loop.	117694.00us
22	0057 1722	decfsz 0x22 ; 117694.00us 58847  21:01 04:I	
23	GOTO	Loop	117702.00us
24	0058 C056	goto Loop ; 117702.00us 58851  [ 57 ]	
25	CALL	Reduce ; Perform the inner portion of	117732.00us
26	0056 E05A Loop	call Reduce ; 117732.00us 58866  22:06 04:I	
27	DECFSZ	OuterLoop,f ; the loop.	117736.00us

### 6.7.2.2 Остановка записи в буфер трассировки

Остановка записи в буфер трассировки (*Halt Trace*) позволяет просмотреть состояние микроконтроллера в точках трассировки без остановки выполнения программы. Когда запись в буфер следа остановлена, нажмите кнопку для продолжения записи *Halt Trace*.

### 6.7.2.3 Просмотр состояния микроконтроллера

Окно буфера трассировки используется для того, чтобы проследить состояние микроконтроллера в ходе выполнения программы в симуляторе MPLAB SIM. Каждая запись буфера трассировки содержит: адрес в памяти программ, выполняемый код, время записи и измененные регистры. При записи времени в буфер трассировки используются часы MPLAB IDE. Вы можете сбросить время записи, сбросив системные часы MPLAB IDE.

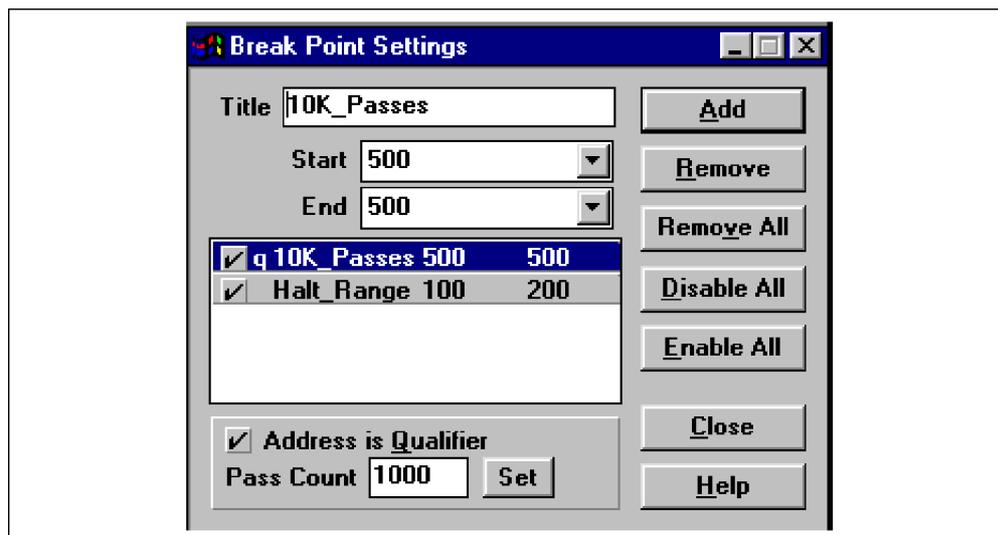
### 6.7.3 Настройка числа проходов для точек остановки и трассировки

MPLAB IDE позволяет назначить один 16-разрядный счетчик прохода для любого адреса памяти программ.

Когда выполнение программы приостановлено, Вы можете изменить число проходов точки остановки/трассировки в диалоговых окнах настройки точек остановки/трассировки соответственно. Сначала указывается желаемый адрес (или диапазон адресов), а затем записывается 16-разрядное значение счетчика проходов. Когда счетчик проходов будет декрементирован до нуля, произойдет остановка программы (запись в буфер трассировки).

#### 6.7.3.1 Настройка числа проходов для точки остановки

Если указан счетчик проходов, то остановка программы произойдет при обнаружении другого условия остановки или при обнулении счетчика проходов.



Пример последовательности настройки точек остановки (показанный на рисунке). Имейте в виду, что настраиваемые точки остановки независимы друг от друга.

Точки остановки:

1. При выполнении любых инструкций в пределах диапазона адресов от 100 до 200.
2. После выполнения 1000 проходов адреса 500.

1. Настроить диапазон адресов 100-200 для точки остановки:

- введите имя *Halt\_Range* в пункте *Title*;
- введите начала диапазона адресов *100* в строке *Start* и конец диапазона *200* в строке *End*;
- нажмите кнопку *Add* для сохранения настроек.

2. Настроить вторую точку остановки с адресом 500:

- введите имя *10K\_Passes* в пункте *Title*;
- введите начала диапазона адресов *500* в строке *Start* и конец диапазона *500* в строке *End*;
- нажмите кнопку *Add* для сохранения настроек.

3. Настроить остановку после выполнения 1000 проходов адреса 500:

- выберите точку остановки *10K\_Passes*;
- отметьте пункт *Address is Qualifier*;
- введите значение *1000* в пункте *Pass Count*;
- нажмите кнопку *Set* для сохранения настроек.

#### 6.7.3.2 Настройка числа проходов для точки трассировки

Если указан счетчик проходов, то пока его значение не станет равно нулю, данные в буфер трассировки записываться не будут. Когда значение счетчика проходов будет равно нулю, данные записываются в буфер трассировки при проходе через точку трассировки.

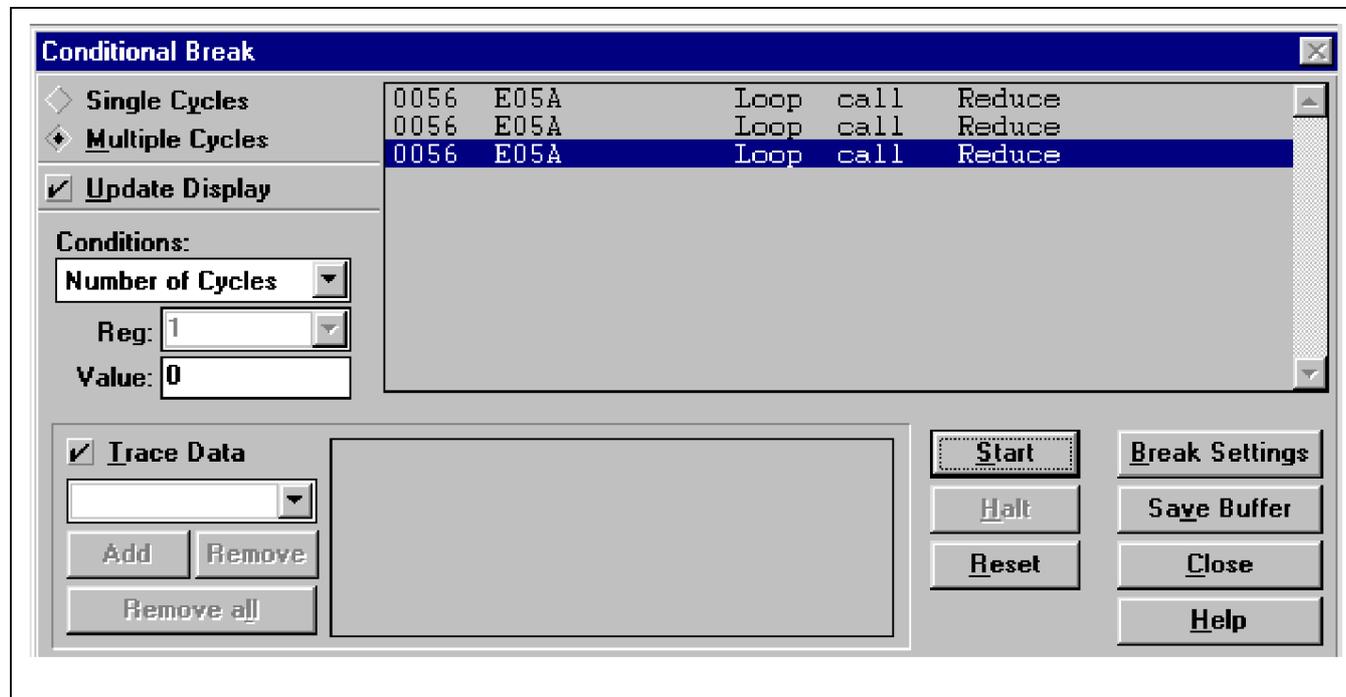
#### 6.7.3.2 Альтернативное использование счетчика проходов

Счетчик проходов декрементируется каждый раз при выполнении команды с указанным адресом. Вы можете использовать эту особенность, чтобы определить число раз исполнения команды.

## 6.8 Точки остановки по условию

Если настроена точка остановки по условию, MPLAB IDE прервет выполнение программы, когда значение внутреннего регистра микроконтроллера будет удовлетворять указанному условию.

Подробное описание работы с точками остановки по условию смотрите в главе «Панели инструментов и меню MPLAB IDE» меню *Debug > Execute > Conditional Break*.



**Примечание.** Данная функция не поддерживается в режиме эмуляции MPLAB ICD или MPLAB ICE. Для эмулятора MPLAB ICE предназначены функции комплексного триггера.

### 6.8.1 Условия

MPLAB IDE остановит ход выполнения программы в одном из следующих случаев:

- остановка пользователем – MPLAB IDE продолжает выполнять программу, пока Вы не нажмете кнопку *Halt* в диалоговом окне точки остановки по условию;
- выполнение указанного числа циклов микроконтроллера – MPLAB IDE остановит программу, после выполнения указанного числа циклов;
- удовлетворение условиям остановки.

### 6.8.2 Данные трассировки

Данные трассировки позволяют Вам контролировать значения регистров в диалоговом окне точек остановки по условию.

### 6.8.3 Единственный цикл

MPLAB IDE выполняет отдельные циклы микроконтроллера, пока не встретится условие остановки.

### 6.8.4 Многократные циклы

В режиме многократных циклов:

- Инструкции выполняются в режиме реального времени (для эмулятора), проверяются условия выбранных точек остановки и условной остановки. Симулятор и эмулятор останавливается только при выполнении указанного условия.
- Точки остановки и условия остановки по значению регистров проверяются только в точках определенных в диалоговом окне настройки параметров точек остановки.

## 6.9 Функции стимула

Стимул подготавливает сигналы для симулятора MPLAB SIM. Вам предоставляется возможность моделировать состояние портов ввода/вывода или записывать значения непосредственно в регистры.

Существует четыре вида стимулов:

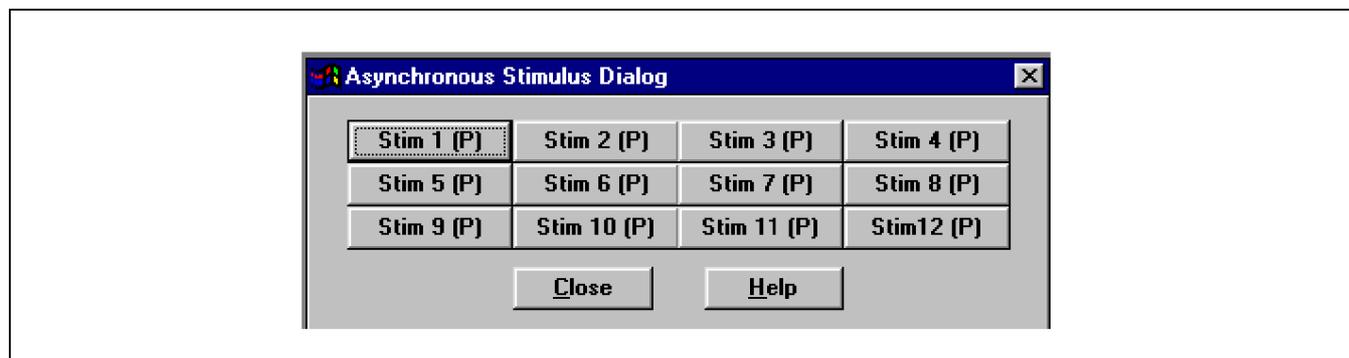
- Асинхронный стимул – непосредственное управление состоянием портов ввода/вывода.
- Файл состояния порта ввода/вывода – текстовый файл, описывающий состояние порта ввода/вывода.
- Файл стимула регистра – текстовый файл, содержащий 8-разрядное значение регистра.
- Стимул тактового сигнала – регулярный, программируемый, периодический источник тактового сигнала.

### 6.9.1 Асинхронный стимул

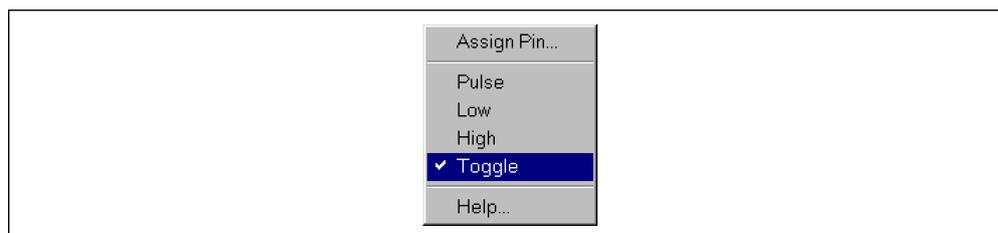
Используется для моделирования логического состояния порта ввода/вывода настроенного на вход (установка значений +5В или 0В). Вы можете нажатием кнопки в диалоговом окне асинхронного стимула указывать входной сигнал на портах ввода/вывода.

Для примера будем управлять состояние сигнала на входе PORTB микроконтроллера PIC16F84.

Выберите пункт меню *Debug > Simulator Stimulus > Asynchronous Stimulus* на экране появится диалоговое окно, показанное на рисунке.



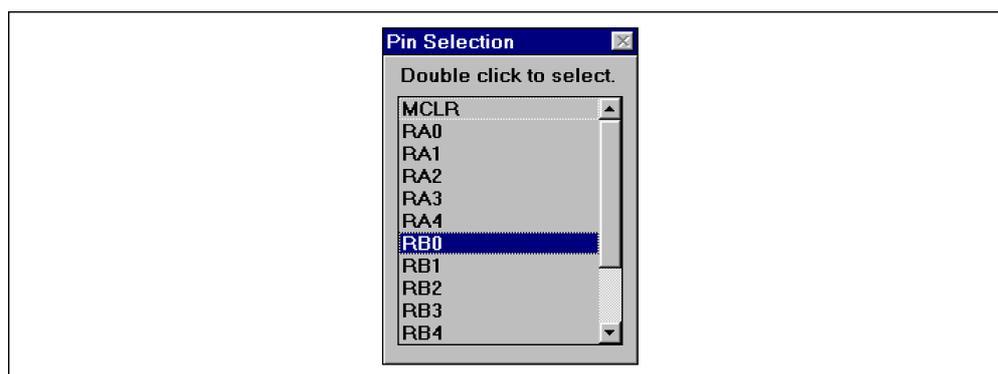
Поместите курсор на кнопке «Stim1 (P)» и нажмите правой кнопкой «мыши», на экране появится сокращенное меню.



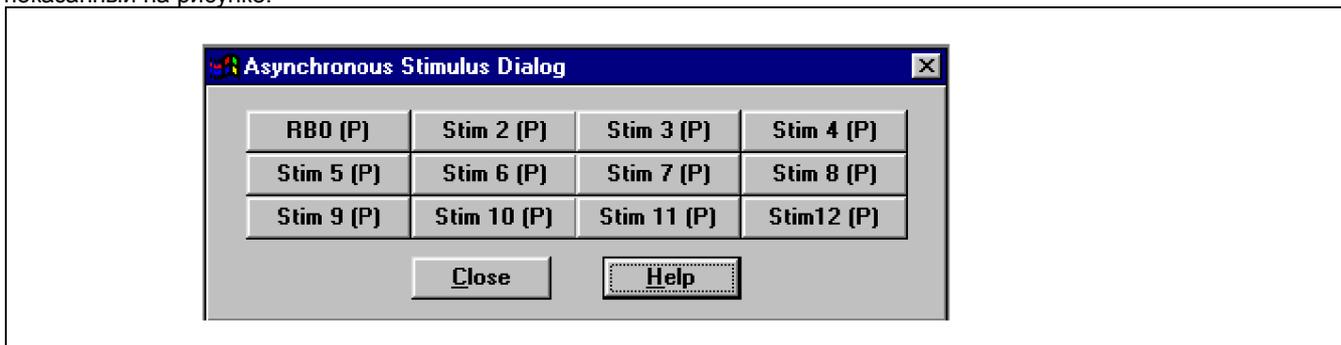
Выберите пункт *Toggle*.

Снова поместите курсор на кнопку «Stim1 (T)» (символ «P» был заменен на «T» значение Toggle) и нажмите правой кнопкой «мыши», на экране появится сокращенное меню.

Список портов ввода/вывода PIC16F84.



Двойным щелчком левой кнопкой «мыши» выберите порт RB0. Асинхронный диалог теперь должен иметь вид, показанный на рисунке.



Обратите внимание, что значение «Stim1 (T)» заменено на «RB0 (T)».

Выберите пункт меню *Debug > Run > Animate* для запуска быстрого выполнения программы по шагам.

Нажмите кнопку «RB0 (T)» в диалоговом окне асинхронного стимула. Изменение состояния PORTB вы можете увидеть в окне регистров специального назначения.

## 6.9.2 Файлы стимула порта ввода/вывода

Текстовый файл стимула портов ввода/вывода состоит из нескольких колонок. Первая колонка определяет номер цикла при выполнении программы, в котором будут изменяться значения портов ввода/вывода.

### 6.9.2.1 Порядок создания файла стимула

1. Создайте новый текстовый файл - *File > New File*. Вы будете вводить текст файла стимула в этом окне.
2. Впишите первое слово в первой линии CYCLE.

**Примечание.** Для совместимости с более ранними версиями MPLAB IDE, первое слово в первой строке всегда должно быть CYCLE или STEP. Первый столбец определяет номер цикла (в соответствии с секундомером MPLAB IDE), в котором будет изменяться состояние порта ввода/вывода.

3. Справа от слова CYCLE укажите имена выводов микроконтроллера, которые должны соответствовать типу моделируемого микроконтроллера.

**Примечание.** Чтобы увидеть список поддерживаемых портов ввода/вывода, воспользуйтесь пунктом меню *Debug > Simulator Stimulus > Asynchronous Stimulus* и нажмите правой кнопкой «мышки» на одной из кнопок окна.

4. Заполните таблицу, указав номер цикла и состояние вывода. После знака «;» или «!» можно вписать комментарии.
5. Сохраните файл стимула - *File > Save As...* Укажите имя и директорию сохраняемого файла. Файл стимула должен иметь расширение .sti. Теперь файл готов к использованию в проекте.

### 6.9.2.2 Порядок использования файла стимула

1. Подключите файл стимула к проекту - *Debug > Simulator Stimulus > Pin Stimulus > Enable*.
2. Откройте дополнительные окна. Например, окно секундомера *Window > Stopwatch* и окно специальных регистров *Window > Special Function Registers*. В окне секундомера будет показано прошедшее время при выполнении каждой команды. Если секундомер сбрасывается, файл стимула портов ввода/вывода тоже сбрасывается в начальное состояние.
3. Сбросьте микроконтроллер и наблюдайте за состоянием порта ввода/вывода, выполняя программу по шагам.

## 6.9.2.3 Пример создания и работы с файлом стимула портов ввода/вывода

**Примечание.** Этот пример подразумевает, что Вы создали проект, описанный в главе 3.

1. Для создания файла стимула порта ввода/вывода, выберите пункт меню *File > New File*, на рабочем столе среды MPLAB IDE появится безымянный файл. Введите текст указанный ниже. После символов «;» или «!» можно вписать комментарии.

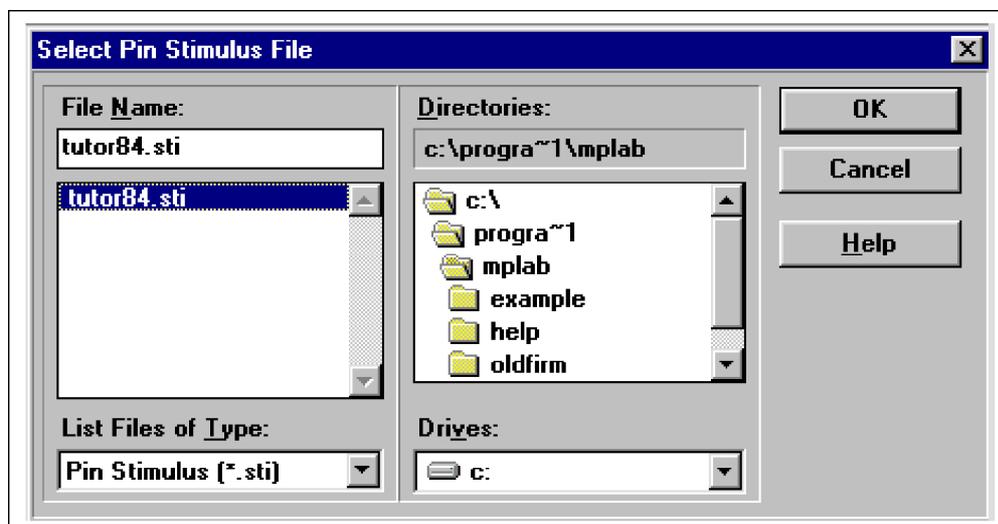
CYCLE	RB1	RB0	
20	0	0	
41	1	0	; Установить высокий уровень сигнала на RB1
52	0	1	; Установить высокий уровень сигнала на RB0, низкий на RB1
55	1	1	
60	0	0	
65	1	0	; Переключить RB1, тогда ....
76	0	1	; .... переключить RB0.

Введите слово CYCLE на первой строке нового файла. Справа от слова CYCLE введите имя выводов порта ввода/вывода PICmicro RB1 и RB0.

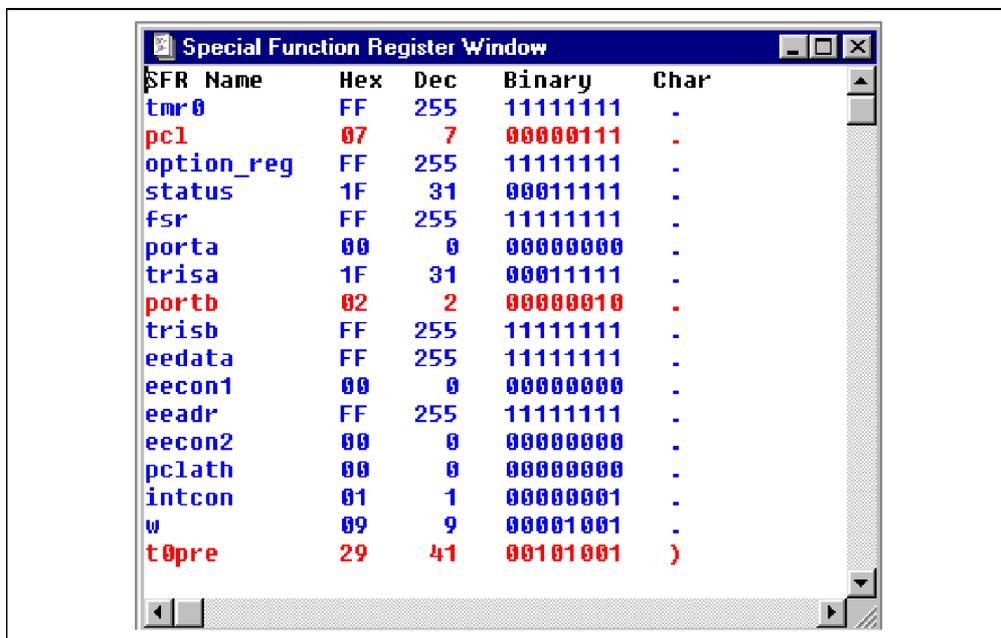
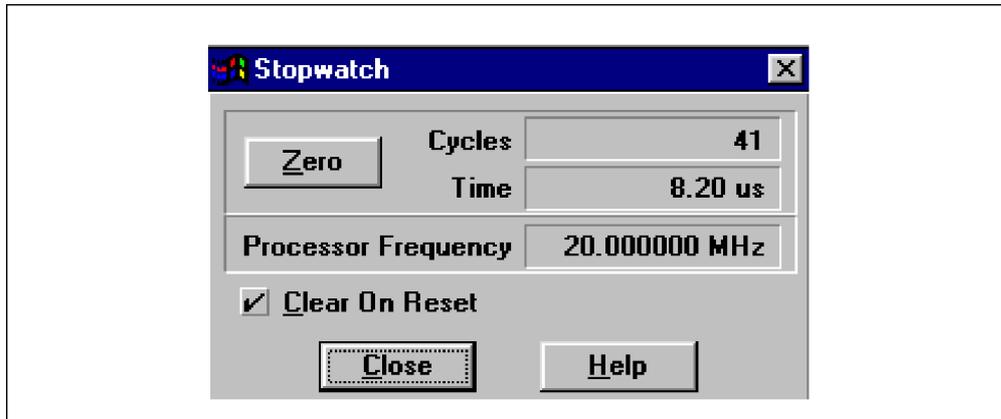
**Примечание.** Для совместимости с более ранними версиями MPLAB IDE, первое слово в первой линии всегда должно быть CYCLE или STEP. Первый столбец определяет номер цикла (в соответствии с секундомером MPLAB IDE), в котором будет изменяться состояние порта ввода вывода.

В этом файле второй столбец содержит значения для вывода RB1 (PORTB бит 1), третий столбец содержит значения для RB0 (PORTB бит 0). Названия выводов должны соответствовать типу моделируемого микроконтроллера PICmicro. Чтобы увидеть список поддерживаемых портов ввода/вывода, воспользуйтесь пунктом меню *Debug > Simulator Stimulus > Asynchronous Stimulus* и нажмите правой кнопкой «мышки» на одной из кнопок окна.

2. Выберите пункт меню *File > Save As...* и сохраните подготовленный текст в файле tutor84.sti.
3. Выберите пункт меню *Debug > Simulator Stimulus > Pin Stimulus > Enable* для подключения файла стимула к проекту.



- Откройте окно секундомера *Window > Stopwatch* и окно регистров специальных функций *Window > Special Function Registers* для просмотра состояния PORTB. В окне секундомера можно увидеть текущий номер цикла, время выполнения программы и тактовую частоту микроконтроллера.
- На 41 цикле, считая от сброса секундомера, PORTB изменит свое значение.



### 6.9.3 Файл стимула регистра

Текстовый файл стимула регистра состоит из одной колонки значений, которые будут переданы регистру, когда адрес в счетчика команд PC равен значению, указанному в диалоговом окне стимула регистра. Это может быть полезно при моделировании работы АЦП.

#### 6.9.3.1 Порядок создания файла стимула регистра

- Создайте новый текстовый файл - *File > New File*. Вы будете вводить текст файла стимула в этом окне.
- В безымянном окне введите список значений в одну колонку, которые Вы хотите назначить регистру.
- Сохраните созданный файл - *File > Save As...* Выберите директорию, в которой хотите сохранить файл, введите имя файла с расширением *.reg*.

#### 6.9.3.2 Порядок использования файла стимула регистра

- Выберите пункт меню *Debug > Simulator Stimulus > Register Stimulus > Enable* для открытия диалогового окна настройки стимула регистра.
- Укажите адрес в памяти программ, в котором значения из файла должны быть подставлены в регистр.
- Укажите адрес регистра в памяти данных, в который будут подставляться значения из файла.
- Нажмите кнопку *Browse* и выберите файл стимула регистра.
- Откройте окно памяти данных - *Window > File Registers*.
- Сбросьте микроконтроллер и выполните шаги программы. Каждый раз, когда значения адреса счетчика команд PC совпадет с указанным адресом в диалоговом окне, в регистр будет занесено значение из файла стимула. Значения из файла стимула будут подставляться циклически, пока выполняется программа в симуляторе MPLAB-SIM.

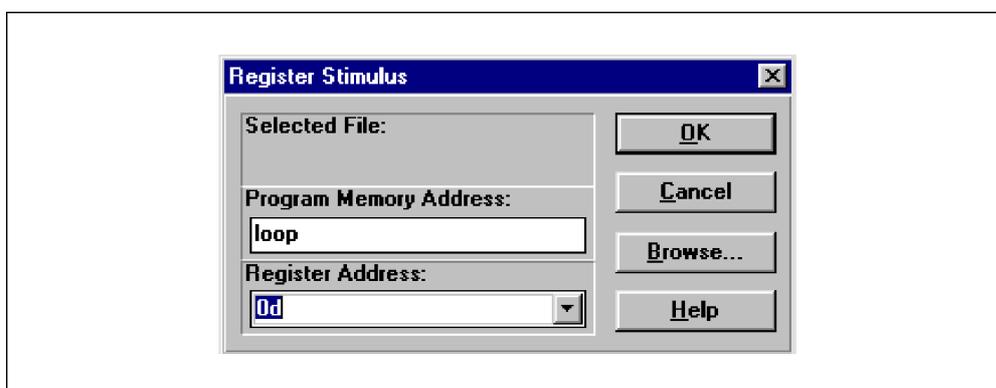
## 6.9.3.3 Пример работы с файлом стимула регистра

**Примечание.** Этот пример подразумевает, что Вы создали проект, описанный в главе 3.

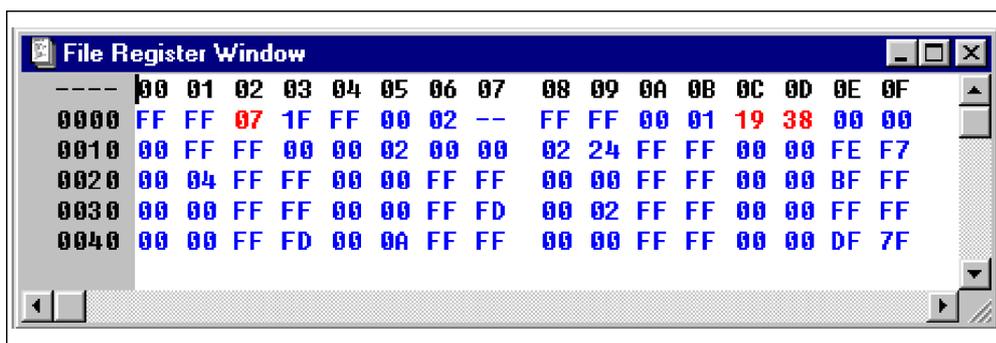
1. Создайте новый файл - *File > New File* и введите следующий текст:

```
10
2E
38
41
50
7A
99
A0
FD
```

2. Сохраните файл *File > Save As...* с именем *tutor84.reg*. Этот файл будет использоваться для последовательного ввода значений в регистр.
3. Выберите пункт меню *Debug > Simulator Stimulus > Register Stimulus > Enable* для открытия диалогового окна стимула регистра. Укажите в качестве адреса памяти программ (*Program Memory Address*) метку *Loop*, адрес регистра памяти данных (*Register Address*) *0x0D* и нажмите кнопку *Browse* для указания файла стимула *tutor84.reg*.



4. Откройте окно с регистрами *Window > File Registers* для наблюдения эффекта стимула регистра.

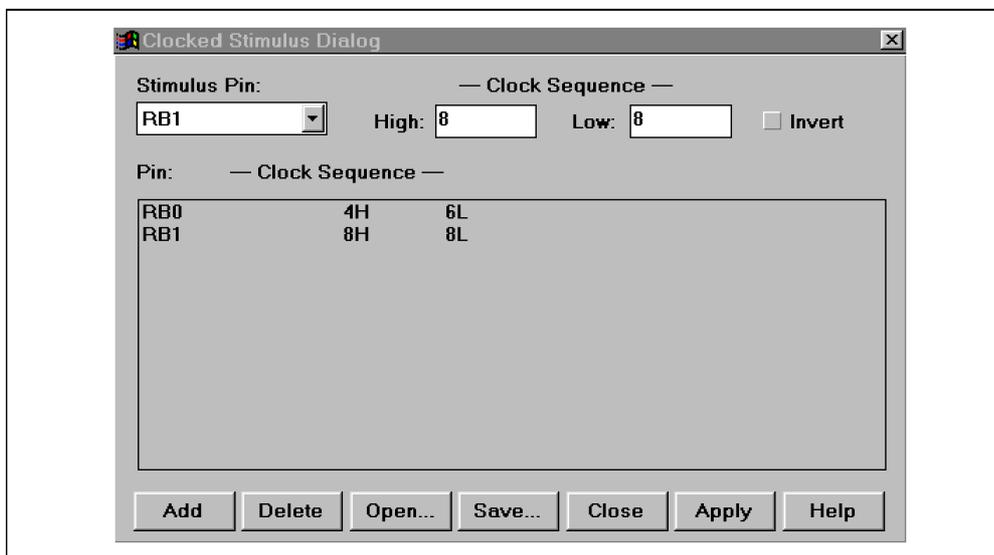


5. Сбросьте микроконтроллер и выполните шаги программы. Каждый раз, когда значения адреса счетчика команд PC совпадет с указанным адресом *Loop*, в регистр *0x0D* будет занесено значение из файла стимула *tutor84.reg*. Значения *0x10*, *0x2E* будут внесены в регистр, выбранный в диалоговом окне стимула регистра, при проходе через метку *Loop*. После того как подставлено последнее значение (*0xFD* из файла *tutor84.reg*), при следующей подстановке, будет использоваться первое значение (*0x10*), т.е. MPLAB-SIM выполняет подстановку циклически.

## 6.9.4 Стимул тактового сигнала

Стимул тактового сигнала формирует на входе порта ввода/вывода регулярную форму сигнала с периодичностью кратной тактовым циклам микроконтроллера. Параметры настройки стимула тактового сигнала сохраняются, пока Вы не закроете MPLAB IDE или не удалите их.

Выберите пункт меню *Debug > Simulator Stimulus > Clock Stimulus* для открытия диалогового окна стимула тактового сигнала, показанного на рисунке.



Настройки, показанные на рисунке следующие:

RB0 – 4 цикла высокий уровень, 6 циклов низкий;

RB1 – 8 цикла высокий уровень, 8 циклов низкий;

Для добавления стимула тактового сигнала – укажите имя порта ввода/вывода, длительность высокого и низкого уровня сигнала и нажмите кнопку *Add*.

Удалить существующий стимул тактового сигнала можно следующим образом – выберите удаляемый стимул из списка и нажмите кнопку *Delete*.

## 6.10 Симуляция 12-разрядных микроконтроллеров

В этом разделе рассмотрена симуляция портов ввода/вывода, прерываний, периферийных модули и режимов 12-разрядных микроконтроллеров.

### 6.10.1 12-разрядные микроконтроллеры

Список поддерживаемых микроконтроллеров (на момент написания данного документа):

- PIC12C508/509
- PIC12CE518/519
- PIC16C52/54/55/56/57/58
- PIC16HV540
- PIC16C505

Вышеуказанный список подразумевает версии микроконтроллеров PIC16CR5X и PIC16C5XA.

### 6.10.2 Порты ввода/вывода

Список выводов микроконтроллера, для которых может быть изменен входной уровень сигнала с помощью файла стимула или асинхронного стимула, исключая несуществующие выводы для конкретного микроконтроллера (например, вывод RC0 не существует в PIC16C54).

- MCLR
- T0CKI
- RA0-RA3
- RB0-RB7
- RC0-RC7

### 6.10.3 Модель микроконтроллера

#### 6.10.3.1 Сброс и режим SLEEP

Все условия сброса поддерживаются симулятором MPLAB SIM. Сброс -MCLR в нормальном или SLEEP режиме микроконтроллера, может быть смоделирован с помощью файла стимула или командой *Debug > Run > Reset*.

Сброс от сторожевого таймера WDT происходит (если он разрешен) через 18мс после последнего сброса счетчика WDT при условии, что предварительный делитель не используется перед WDT.

Причина сброса определяется битами -TO – сброс от сторожевого таймера и -PD – сброс по включению питания в регистре STATUS. Эта функция полезна при отладке алгоритмов обработки причин сброса.

#### 6.10.3.2 Сторожевой таймер WDT

Работа сторожевого таймера WDT полностью моделируется симулятором MPLAB SIM. Включение WDT выполняется в меню *Options > Development Mode* раздел *Configuration*. Коэффициент предделителя определяется битами в регистре OPTION. Период WDT при коэффициенте деления 1 примерно равен 18мс (синхронизируется относительно циклов команд).

### 6.10.4 Периферия

В симуляторе MPLAB SIM полностью реализована работа модуля TMR0 как от внутреннего, так и от внешнего источника тактового сигнала. Значение предделителя TMR0 указано в переменной 'T0PRE' для удобства отладки программы.

**Примечание.** Источник тактового сигнала с длительностью логического уровня меньше 1 T<sub>CY</sub> моделироваться не может.

## 6.11 Симуляция 14-разрядных микроконтроллеров

В этом разделе рассмотрена симуляция портов ввода/вывода, прерываний, периферийных модули и режимов 14-разрядных микроконтроллеров.

### 6.11.1 14-разрядные микроконтроллеры

Список поддерживаемых микроконтроллеров (на момент написания данного документа):

- PIC12C671/672
- PIC12CE673/674
- PIC14000
- PIC16C62/62A/62B/63/63A/64A/65A/65B/66/67
- PIC16C71/72/72A/73A/73B/74A/74B/76/77
- PIC16C554/558
- PIC16C620/621/622
- PIC16C642/662
- PIC16C710/711/715
- PIC16C712/716
- PIC16C717
- PIC16C770/771
- PIC16C773/774
- PIC16C923/924
- PIC16CE623/624/625
- PIC16F83/84/84A
- PIC16F872/873/874/876/877

Вышеуказанный список подразумевает версии микроконтроллеров PIC16CRXX и PIC16CXXA.

### 6.11.2 Порты ввода/вывода

Список выводов микроконтроллера, для которых может быть изменен входной уровень сигнала с помощью файла стимула или асинхронного стимула, исключая несуществующие выводы для конкретного микроконтроллера, представлен ниже.

- MCLR
- T0CKI
- RA0-RA5
- RB0-RB7
- RC0-RC7
- RD0-RD7
- RE0-RE7

### 6.11.3 Прерывания

Симулятор MPLAB SIM поддерживает все виды прерываний 14-разрядных микроконтроллеров (исключая несуществующие прерывания для конкретного микроконтроллера).

- TMR0 переполнение
- TMR1 переполнение
- TMR2
- CCP1
- CCP2
- SSP (только в режиме SPI)
- Изменение состояние на портах RB<7:4>
- Внешнее прерывание RB0/INT
- Компаратор
- АЦП окончание преобразования
- EEPROM окончание записи

## 6.11.4 Модель микроконтроллера

### 6.11.4.1 Сброс

Все условия сброса поддерживаются симулятором MPLAB SIM. Сброс -MCLR в нормальном или SLEEP режиме микроконтроллера, может быть смоделирован с помощью файла стимула или командой *Debug > Run > Reset*.

Сброс от сторожевого таймера WDT происходит (если он разрешен) через 18мс после последнего сброса счетчика WDT при условии, что предварительный делитель не используется перед WDT.

Причина сброса определяется битами -TO – сброс от сторожевого таймера и -PD – сброс по включению питания в регистре STATUS. Эта функция полезна при отладке алгоритмов обработки причин сброса.

### 6.11.4.2 Режим SLEEP

Симулятор MPLAB-SIM модулирует выполнение команды SLEEP, переводя микроконтроллер в режим SLEEP до возникновения условия «пробуждения». Например, если разрешена работа сторожевого таймера WDT, то после переполнения WDT (время переполнения зависит от коэффициента предделителя установленного в регистре OPTION ) микроконтроллер выйдет из режима SLEEP.

Другой пример, выход микроконтроллера из режима SLEEP по переполнению таймера TMR1. Пока микроконтроллер находится в режиме SLEEP, TMR1 инкрементируется до переполнения. Если прерывания от TMR1 разрешены, то микроконтроллер «проснется» и выполнит переход по вектору прерывания.

### 6.11.4.3 Сторожевой таймер WDT

Работа сторожевого таймера WDT полностью моделируется симулятором MPLAB SIM. Включение WDT выполняется в меню *Options > Development Mode* раздел *Configuration*. Коэффициент предделителя определяется битами в регистре OPTION. Период WDT при коэффициенте деления 1 примерно равен 18мс (синхронизируется относительно циклов команд).

## 6.11.5 Специальные регистры

Для более удобной отладки программы, некоторые элементы, которые не могут быть явно прочитаны, были объявлены как специальные регистры. Предделители и выходные делители таймеров не могут быть объявлены в вашем тексте программы, но в окне регистров специального назначения Вы можете увидеть значения этих регистров.

Список специальных регистров, доступных для 14-разрядных микроконтроллеров (обратитесь к технической документации на микроконтроллер для определения реализованных регистров).

- T0PRE – предделитель таймера 0
- T1PRE – предделитель таймера 1
- T2PRE – предделитель таймера 2
- T2POS – выходной делитель таймера 2
- CCP1PRE – предделитель CCP1
- SPIPRE – предделитель SPI
- SSPSR – регистр изменения SSP

## 6.11.6 Периферия

### 6.11.6.1 Поддерживаемые периферийные модули

Следующие периферийные модули поддерживаются симулятором MPLAB SIM для 14-разрядных микроконтроллеров (обратитесь к технической документации на микроконтроллер для определения реализованных периферийных модулей):

- Таймер TMR0
- Таймер TMR1
- Таймер TMR2
- CCP1
- CCP2
- Параллельный Slave порт
- SSP (только в SPI режиме)
- Компаратор
- АЦП (ограничено)

### 6.11.6.2 Таймер TMR0

В симуляторе MPLAB SIM полностью реализована работа модуля TMR0 как от внутреннего, так и от внешнего источника тактового сигнала (с поддержкой прерываний при переполнении). Значение предделителя TMR0 указано в переменной 'T0PRE' для удобства отладки программы. Источник тактового сигнала с длительностью логического уровня меньше 1 T<sub>cy</sub> моделироваться не может.

#### 6.11.6.3 Таймер TMR1

Поддерживаются все режимы работы таймера TMR1, кроме использования внешнего керамического резонатора. Симулятор MPLAB-SIM обрабатывает прерывания от TMR1 при переполнении, с выводом микроконтроллера из режима SLEEP. Значение предделителя таймера TMR1 можно увидеть в окне регистров специального назначения - T1PRE регистр. Внешний тактовый генератор не моделируется, но стимул тактового сигнала может быть назначен на выводы RC0/RC1.

#### 6.11.6.4 Таймер TMR2

Полностью моделируется работа таймера TMR2 симулятором MPLAB-SIM, с обработкой прерываний при переполнении. Значение предделителя и выходного делителя записывается в переменные T2PRE и T2POS соответственно.

#### 6.11.6.5 CCP1 и CCP2

##### *Режим захвата*

Симулятор полностью моделирует режим захвата, с обработкой прерываний. Значение предделителя сохраняется в переменной CCP1PRE.

##### *Режим сравнения*

В режиме сравнения реализуются прерывания, специальный спусковой механизм, сброс таймера TMR1.

##### *ШИМ*

Выходной сигнал ШИМ формируется с дискретностью в один машинный цикл микроконтроллера ( $1 T_{CY}$ ).

#### 6.11.6.6 SSP

Реализуется только режим SPI. Значение сдвигового регистра (SSPSR) может быть проверено и изменено. В настоящее время симулятор MPLAB SIM не поддерживает режим I2C.

#### 6.11.6.7 АЦП

Полностью моделируется работа модуля АЦП с формированием прерываний. Однако симулятор не записывает никакого результата в регистр ADRES в конце преобразования.

#### 6.11.6.8 EEPROM

Полностью моделируется работа EEPROM памяти (для микроконтроллеров PIC16F8X), с обработкой всех циклов чтения/записи. Время записи в EEPROM память примерно равно 10 мс (синхронизируется относительно циклов команд). Имитируется работа битов WREER и WREN в регистре EECON1.

## 6.12 Симуляция 16-разрядных микроконтроллеров PIC17CXXX

В этом разделе рассмотрена симуляция портов ввода/вывода, прерываний, периферийных модули и режимов 16-разрядных микроконтроллеров семейства PIC17CXXX.

### 6.12.1 16-разрядные микроконтроллеры семейства PIC17CXXX

Список поддерживаемых микроконтроллеров (на момент написания данного документа):

- PIC17C42/43/44
- PIC17C752/756
- PIC17C762/766

Вышеуказанный список подразумевает версии микроконтроллеров PIC17CRXX и PIC17CXXA.

### 6.12.2 Порты ввода/вывода

В 16-разрядных микроконтроллерах выводы портов ввода/вывода мультиплицируются с периферийными модулями (поэтому имеют более одного обозначения). Список выводов микроконтроллера, для которых может быть изменен входной уровень сигнала с помощью файла стимула или асинхронного стимула, исключая несуществующие выводы для конкретного микроконтроллера, представлен ниже.

- MCLR
- T0CKI
- RA0-RA5
- RB0-RB7
- RC0-RC7
- RD0-RD7
- RE0-RE2

### 6.12.3 Прерывания

Симулятор MPLAB SIM поддерживает все виды прерываний 16-разрядных микроконтроллеров (исключая несуществующие прерывания для конкретного микроконтроллера).

- Внешнее прерывание INT
- Переполнение таймера TMR0
- Внешнее прерывание с входа RA0
- Изменение состояние на входах порта RB
- Таймер/счетчик 1
- Таймер/счетчик 2
- Таймер/счетчик 3
- Прерывание захвата 1
- Прерывание захвата 2

### 6.12.4 Модель микроконтроллера

#### 6.12.4.1 Сброс

Все условия сброса поддерживаются симулятором MPLAB SIM. Сброс -MCLR в нормальном или SLEEP режиме микроконтроллера, может быть смоделирован с помощью файла стимула или командой *Debug > Run > Reset*.

Сброс от сторожевого таймера WDT происходит (если он разрешен) через 12мс после последнего очистки счетчика WDT, при условии, что предварительный делитель не используется перед WDT.

Причина сброса определяется битами -TO – сброс от сторожевого таймера и -PD – сброс по включению питания в регистре ALUSTA. Эта функция полезна при отладке алгоритмов обработки причин сброса.

#### 6.12.4.2 Режим SLEEP

Симулятор MPLAB-SIM модулирует выполнение команды SLEEP, переводя микроконтроллер в режим SLEEP до возникновения условия «пробуждения». Например, если разрешена работа сторожевого таймера WDT, то после переполнения WDT микроконтроллер выйдет из режима SLEEP.

Другой пример, выход микроконтроллера из режима SLEEP по изменению уровня сигнала на входе PORTB. Если выполнено условие возникновения прерывания и бит GLINTD=1, то микроконтроллер выйдет из режима SLEEP и продолжит выполнять инструкции после команды SLEEP. Если бит GLINTD=0, то будет выполнена обработка с переходом по вектору прерываний.

#### 6.12.4.2 Сторожевой таймер WDT

Работа сторожевого таймера WDT полностью моделируется симулятором MPLAB SIM. Включение WDT выполняется в меню *Options > Development Mode* раздел *Configuration*. Период WDT при коэффициенте деления 1 примерно равен 12мс (синхронизируется относительно циклов команд).

### 6.12.5 Специальные регистры

Для более удобной отладки программы, некоторые элементы, которые не могут быть явно прочитаны, были объявлены как специальные регистры. Предделители и выходные делители таймеров не могут быть объявлены в вашем тексте программы, однако в окне регистров специального назначения Вы можете увидеть значения этих регистров.

Список специальных регистров, доступных для 16-разрядных микроконтроллеров семейства PIC17CXXX (обратитесь к технической документации на микроконтроллер для определения реализованных регистров).

- T0PRE – предделитель таймера 0
- WDTPRE – предделитель WDT

### 6.12.6 Периферия

#### 6.12.6.1 Поддерживаемые периферийные модули

Следующие периферийные модули поддерживаются симулятором MPLAB SIM для 16-разрядных микроконтроллеров семейства PIC17CXXX (обратитесь к технической документации на микроконтроллер для определения реализованных периферийных модулей):

- Таймер TMR0 с внутренним и внешним тактовым сигналом
- Таймеры TMR1 и TMR2 (и их регистры периода)
- Таймер TMR3
- Два модуля захвата
- Два ШИМ модуля

#### 6.12.6.2 Таймер TMR0

В симуляторе MPLAB SIM полностью реализована работа модуля TMR0 как от внутреннего, так и от внешнего источника тактового сигнала (с генерацией прерывания при переполнении). Также моделируется задержка от активного фронта тактового сигнала до приращения таймера. Значение предделителя TMR0 указано в переменной 'T0PRE' с возможностью его изменения. Источник тактового сигнала с длительностью логического уровня меньше 1 T<sub>cy</sub> моделироваться не может (согласно параметрам файла стимула).

#### 6.12.6.3 Таймеры TMR1 и TMR2

TMR1 и TMR2 полностью моделируются симулятором MPLAB SIM во всех режимах. Также моделируется задержка от активного фронта тактового (заднего, переднего в режиме с внешним тактовым сигналом) сигнала до приращения таймера. Источник внешнего тактового сигнала с длительностью логического уровня меньше 1 T<sub>cy</sub> моделироваться не может.

#### 6.12.6.4 Таймер 3 и захват

Полнофункциональная симуляция работы модуля TMR3 и модуля захвата во всех режимах. Также моделируется задержка от активного фронта тактового (в режиме с внешним тактовым сигналом) сигнала до приращения таймера. Источник внешнего тактового сигнала с длительностью логического уровня меньше 1 T<sub>cy</sub> моделироваться не может.

#### 6.12.6.5 ШИМ

Выходной сигнал ШИМ формируется с дискретностью в один машинный цикл микроконтроллера (1 T<sub>cy</sub>).

### 6.12.7 Реализация памяти

Симулятор MPLAB SIM поддерживает следующие режимы работы микроконтроллера:

- режим микроконтроллера;
- расширенный режим микроконтроллера;
- режим микропроцессора.

Для установки режима работы микроконтроллера выберет пункт меню *Options > Development Mode*.

## 6.13 Симуляция 16-разрядных микроконтроллеров PIC18CXXX

В этом разделе рассмотрена симуляция портов ввода/вывода, прерываний, периферийных модули и режимов 16-разрядных микроконтроллеров семейства PIC18CXXX.

### 6.13.1 16-разрядные микроконтроллеры семейства PIC18CXXX

Список поддерживаемых микроконтроллеров (на момент написания данного документа):

- PIC18C242
- PIC18C252
- PIC18C442
- PIC18C452

Вышеуказанный список подразумевает версии микроконтроллеров PIC18CRXXX и PIC18CXXXA. Обратите внимание, что в этом семействе микроконтроллеров PIC18C байтовая организация памяти программ, а не словная, как в других семействах. Имеются некоторые ограничения по обращению к памяти программ, особенно при использовании записи в память программ. В некоторых ситуациях, симулятор не может учитывать все ограничения. Воспользуйтесь технической документацией на микроконтроллер для правильного выполнения действий с памятью программ.

### 6.13.2 Порты ввода/вывода

В 16-разрядных микроконтроллерах выводы портов ввода/вывода мультиплицируются с периферийными модулями (поэтому имеют более одного обозначения). Список выводов микроконтроллера, для которых может быть изменен входной уровень сигнала с помощью файла стимула или асинхронного стимула, исключая несуществующие выводы для конкретного микроконтроллера, представлен ниже.

- MCLR
- T0CKI
- RA0-RA5
- RB0-RB7
- RC0-RC7
- RD0-RD7
- RE0-RE2

### 6.13.3 Прерывания

Симулятор MPLAB SIM поддерживает все виды прерываний 16-разрядных микроконтроллеров (исключая несуществующие прерывания для конкретного микроконтроллера).

- Внешнее прерывание INT
- Переполнение таймера TMR0
- Внешнее прерывание с входа RA0
- Изменение состояние на входах порта RB
- Таймер/счетчик 1
- Таймер/счетчик 2
- Таймер/счетчик 3
- Прерывание захвата 1
- Прерывание захвата 2

### 6.13.4 Модель микроконтроллера

#### 6.13.4.1 Сброс

Все условия сброса поддерживаются симулятором MPLAB SIM. Сброс -MCLR в нормальном или SLEEP режиме микроконтроллера, может быть смоделирован с помощью файла стимула (формированием импульса низкого уровня на выводе MCLR), нажатием кнопки сброса на графическом меню или командой *Debug > Run > Reset*.

Сброс от сторожевого таймера WDT произойдет, если он разрешен аппаратно (*Options > Development Mode* раздел *Configuration*) или если запрещен аппаратно, но разрешен программой пользователя в регистре WDTCON.

Когда произойдет пополнение WDT, будет выполнен сброс микроконтроллера. После сброса микроконтроллера выполнение программы может быть приостановлено (*Options > Development Mode* раздел *Break Options*). Период пополнения WDT при выключенном делителе примерно равен 18мс (синхронизируется относительно циклов команд).

#### 6.13.4.2 Режим SLEEP

Симулятор MPLAB-SIM модулирует выполнение команды SLEEP, переводя микроконтроллер в режим SLEEP до возникновения условия «пробуждения». Например, если разрешена работа сторожевого таймера WDT, то после переполнения WDT микроконтроллер выйдет из режима SLEEP.

Другой пример, выход микроконтроллера из режима SLEEP по изменению уровня сигнала на входе PORTB. Если выполнено условие возникновения прерывания и бит GIE=0, то микроконтроллер выйдет из режима SLEEP и продолжит выполнять инструкции после команды SLEEP. Если бит GIE=1, то будет выполнена обработка с переходом по вектору прерываний.

#### 6.13.4.3 Сторожевой таймер WDT

Работа сторожевого таймера WDT полностью моделируется симулятором MPLAB SIM. Включение WDT выполняется в меню *Options > Development Mode* раздел *Configuration*. Период WDT при коэффициенте деления 1 примерно равен 18мс (синхронизируется относительно циклов команд).

### 6.13.5 Специальные регистры

Для более удобной отладки программы, некоторые элементы, которые не могут быть явно прочитаны, были объявлены как специальные регистры. Предделители и выходные делители таймеров не могут быть объявлены в вашем тексте программы, однако в окне регистров специального назначения Вы можете увидеть значения этих регистров.

Список специальных регистров, доступных для 16-разрядных микроконтроллеров семейства PIC18CXXX (см. техническую документацию на микроконтроллер для определения реализованных регистров).

- TOPRE – предделитель таймера 0
- WDTPRE – предделитель WDT

### 6.13.6 Периферия

#### 6.13.6.1 Поддерживаемые периферийные модули

Следующие периферийные модули поддерживаются симулятором MPLAB SIM для 16-разрядных микроконтроллеров семейства PIC18CXXX (см. техническую документацию на микроконтроллер для определения реализованных периферийных модулей):

- Таймер TMR0 с внутренним и внешним тактовым сигналом
- Таймеры TMR1 и TMR2 (и их регистры периода)
- Таймер TMR3
- Два модуля захвата
- Два ШИМ модуля

#### 6.13.6.2 Таймер TMR0

В симуляторе MPLAB SIM полностью реализована работа модуля TMR0 как от внутреннего, так и от внешнего источника тактового сигнала (с генерацией прерывания при переполнении). Также моделируется задержка от активного фронта тактового сигнала до приращения таймера. Значение предделителя TMR0 указано в переменной 'TOPRE' с возможностью его изменения. Источник тактового сигнала с длительностью логического уровня меньше 1 T<sub>CY</sub> моделироваться не может (согласно параметрам файла стимула).

#### 6.13.6.3 Таймеры TMR1 и TMR2

TMR1 и TMR2 полностью моделируются симулятором MPLAB SIM во всех режимах. Также моделируется задержка от активного фронта тактового (заднего, переднего в режиме с внешним тактовым сигналом) сигнала до приращения таймера. Источник внешнего тактового сигнала с длительностью логического уровня меньше 1 T<sub>CY</sub> моделироваться не может.

#### 6.13.6.4 Таймер 3 и захват

Полнофункциональная симуляция работы модуля TMR3 и модуля захвата во всех режимах. Также моделируется задержка от активного фронта тактового (в режиме с внешним тактовым сигналом) сигнала до приращения таймера. Источник внешнего тактового сигнала с длительностью логического уровня меньше 1 T<sub>CY</sub> моделироваться не может.

#### 6.13.6.5 ШИМ

Выходной сигнал ШИМ формируется с дискретностью в один машинный цикл микроконтроллера (1 T<sub>CY</sub>).

## Глава 7. Панели инструментов и меню MPLAB IDE

### 7.1 Введение

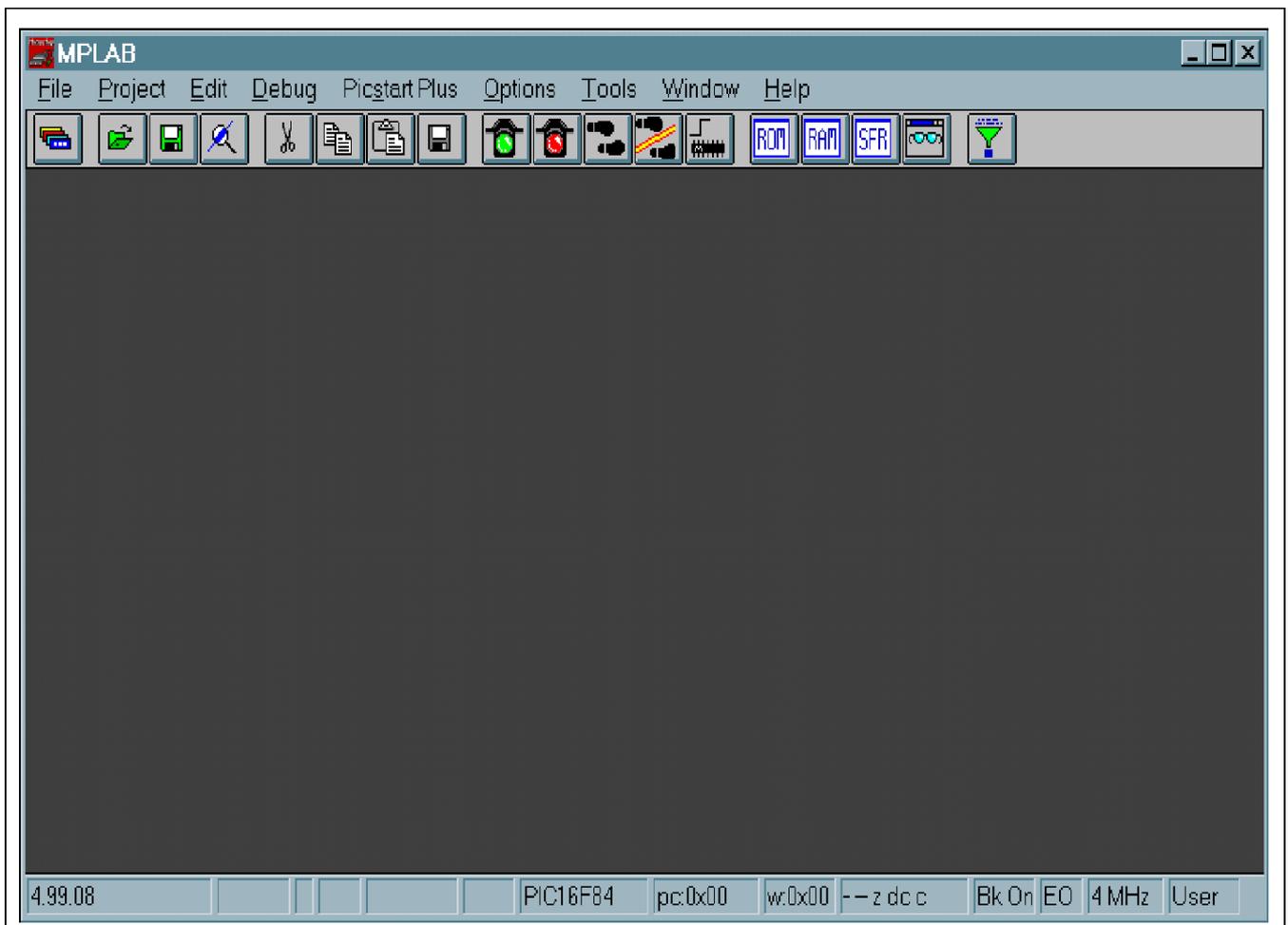
В главе описано использование рабочего стола MPLAB IDE, работа с графическим меню и структура текстового меню.

### 7.2 Основные разделы главы

- Рабочий стол MPLAB IDE
- Меню File
- Меню проекта (Project)
- Меню редактора MPLAB (Edit)
- Меню отладки (Debug)
- Меню программаторов (Picstart Plus/PRO MATE)
- Меню настроек (Options)
- Меню Tools
- Меню Window
- Меню Help

### 7.3 Рабочий стол MPLAB IDE

Размер рабочего стола назначается независимо от пунктов меню MPLAB IDE. Чтобы уменьшить размер окна MPLAB IDE, нажмите кнопку «развернуть» в правом верхнем углу окна. Открыть окно на весь экран монитора можно повторным нажатием на кнопку «развернуть». При следующих запусках MPLAB IDE размер окна будет такой же, как и при закрытии среды проектирования. Если Вы не используете MPLAB IDE, но необходимо оставить его загруженным в памяти компьютера, нажмите кнопку «свернуть», рабочий стол операционной системы освободится для работы.



Диалоговые, рабочие и информационные окна среды MPLAB IDE работают как отдельные приложения, предоставляя возможность управления их отображением на экране монитора через стандартные кнопки и элементы (кнопки размера окна, вертикальные и горизонтальные линии и кнопки прокрутки) операционной системы Windows Microsoft в правом верхнем углу окна.

Все функции MPLAB IDE доступны через текстовое меню, расположенное сверху рабочего стола. Подчеркнутые символы в опускающемся меню обозначают клавишу в «горячей комбинации». Чтобы вызвать пункт меню, нажмите и удерживайте кнопку «Alt», а затем нажмите кнопку с подчеркнутым символом в меню. Например, комбинация клавиш <Alt+F> откроет меню File, а комбинация клавиш <Alt+O> откроет диалоговое окно «Открыть файл».

### 7.3.1 Графическое меню

В графическом меню кнопки управления объединены в четыре логических группы:

- меню редактирования;
- меню отладки программы;
- меню проектирования;
- меню пользователя.

Переключение между группами кнопок осуществляется нажатием крайней левой кнопки графического меню. Стандартный состав и назначение кнопок графического меню смотрите в приложении А.

В каждой группе графического меню состав кнопок управления может быть настроен в соответствии с вашими требованиями (см. раздел 7.9.5.1.2).



### 7.3.2 Линейка состояния

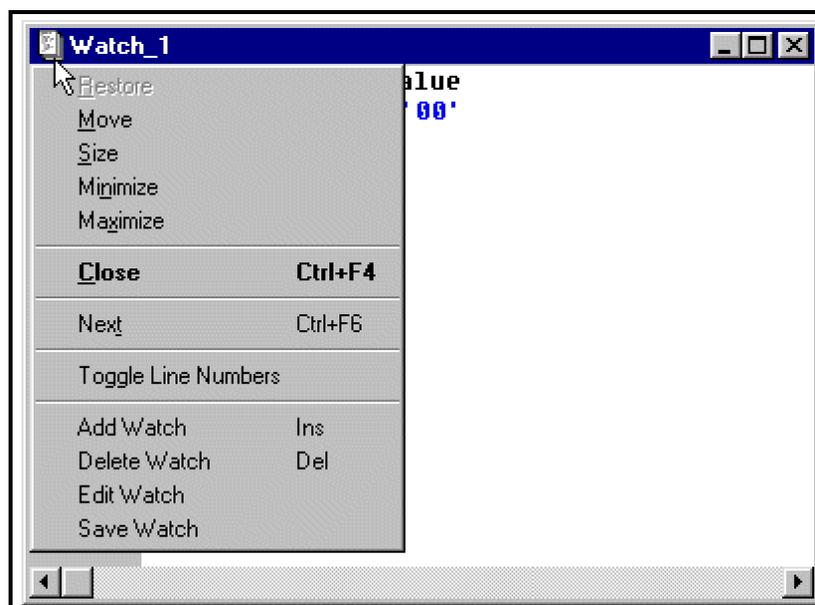
В нижней части рабочего стола MPLAB IDE отображается линейка состояния.



Состав и назначение полей линейки состояния смотрите в приложении В.

### 7.3.3 Системное меню

Для каждого окна среды проектирования MPLAB IDE может быть вызвано системное меню настройки параметров отображения информации.



## 7.4 Меню File

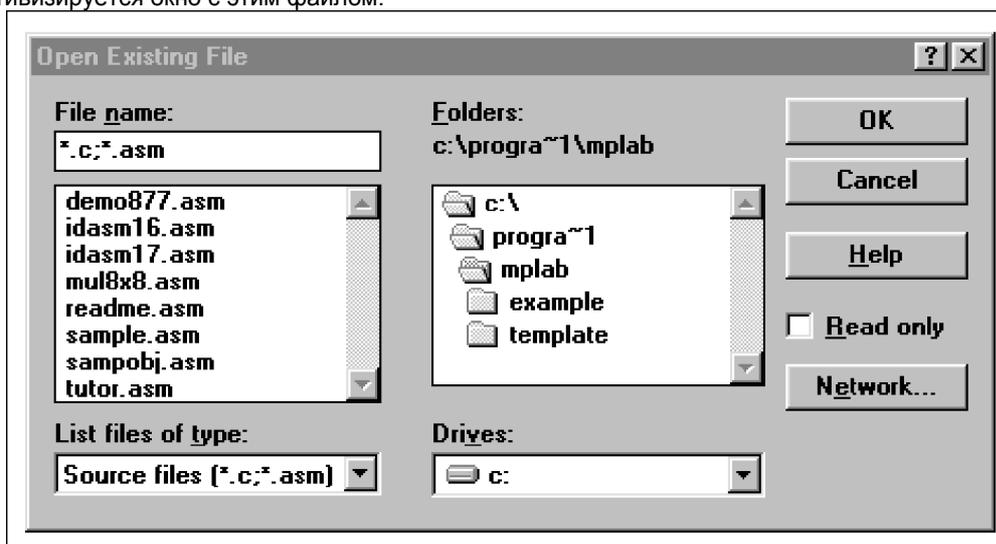
### 7.4.1 Новый файл (File > New)

Командой *File > New* создается новый пустой, безымянный файл. Настройки файла соответствуют параметрам для новых файлов в меню *Options > Environment Setup*.

Сохранить новый файл на диске можно командой *File > Save As*.

### 7.4.2 Открыть сохраненный файл на диске (File > Open)

Чтобы открыть для редактирования один или более сохраненных файлов, воспользуйтесь командой *File > Open*. На экране появится диалоговое окно, показанное на рисунке. Если Вы выберете уже открытый файл, то на рабочем столе активизируется окно с этим файлом.

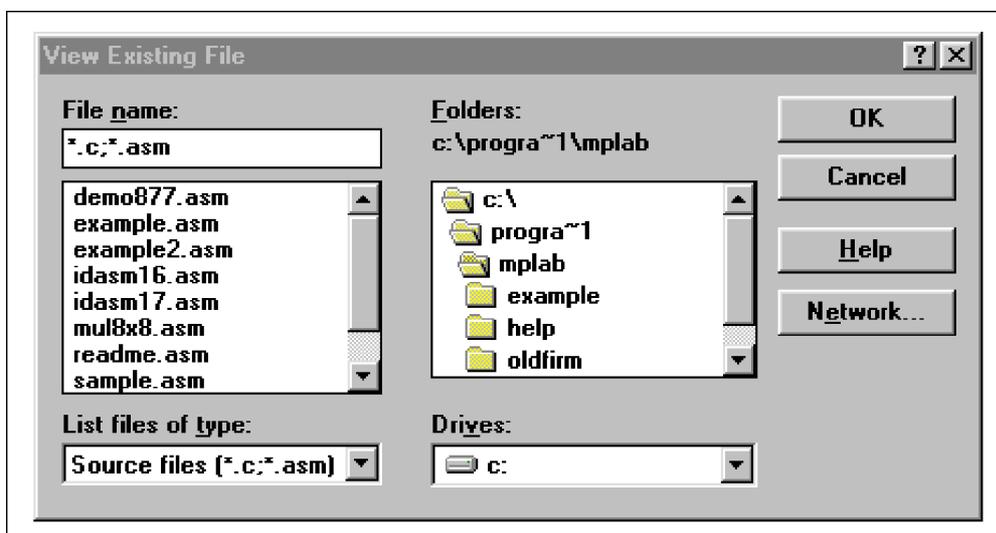


1. Укажите диск (*Drives:*), директорию (*Directories:*), тип файла (*List Files of Type:*).
2. Выберите открываемый файл из списка *File Name*. Для открытия нескольких файлов сразу используйте кнопку <Ctrl> (выделение отдельных файлов) и <Shift> (выделение группы файлов) при выборе.
3. Для запрещения редактирования файла, установите флажок *Read Only* (только чтение).
4. Подтвердите выбор файлов нажатием кнопки *OK*.

Ограничить список файлов в выбранной директории, можно указав расширения файла (*List Files of Type*). Например, если выбрать «\*.ASM», то в список будут включены файлы только с расширением .ASM. Закрывая диалоговое окно кнопкой *OK* или *Cancel*, MPLAB IDE запоминает выбранную директорию.

### 7.4.3 Открыть файл для просмотра (File > View)

Аналогично команде *File > Open*, но с установленным флагом «только чтение».



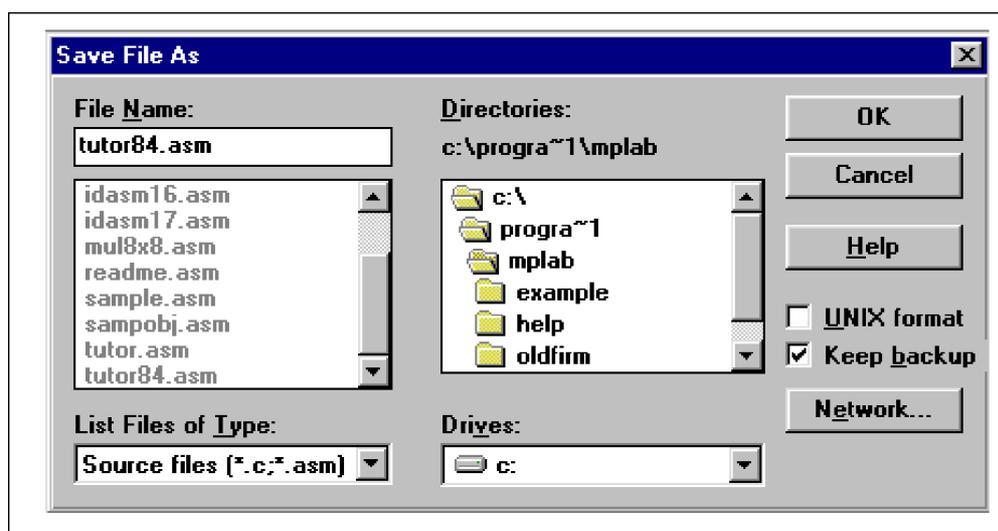
## 7.4.4 Сохранение файлов (File > Save, File > Save As, File > Save All)

Сохранить открытые файлы можно тремя способами:

Выберите *File > Save* для сохранения файла на диске из активного окна MPLAB IDE. Редактор заменяет текст в файле на новый. Если активный файл безымянный, редактор MPLAB потребует указать имя. Если файл на диске существует, редактор MPLAB делает резервную копию и сохраняет текущий файл.

Выберите *File > Save As* для сохранения файла с указанием имени. Редактор MPLAB поддерживает замену существующих файлов.

1. Укажите диск (*Drives:*) и директорию (*Directories:*), в которой хотите сохранить файл.
2. Введите имя и расширение файла (*File Name*) или выберите из списка для замены.
3. Нажмите кнопку *OK* для сохранения файла. Если Вы выбрали имя существующего файла, редактор MPLAB выдаст предупреждение о замене файла.
4. Ограничить список файлов в выбранной директории, можно указав расширения файла (*List Files of Type*).



Выберите пункт меню *File > Save All* для сохранения всех открытых файлов.

**Примечание.** Нажмите кнопку *Cancel* для закрытия диалогового окна без сохранения файлов.

## 7.4.5 Закрывать открытые файлы (File > Close, File > Close All)

Для закрытия активного окна выберите пункт меню *File > Close*.

Чтобы закрыть все открытые файлы, выберите пункт меню *File > Close All*.

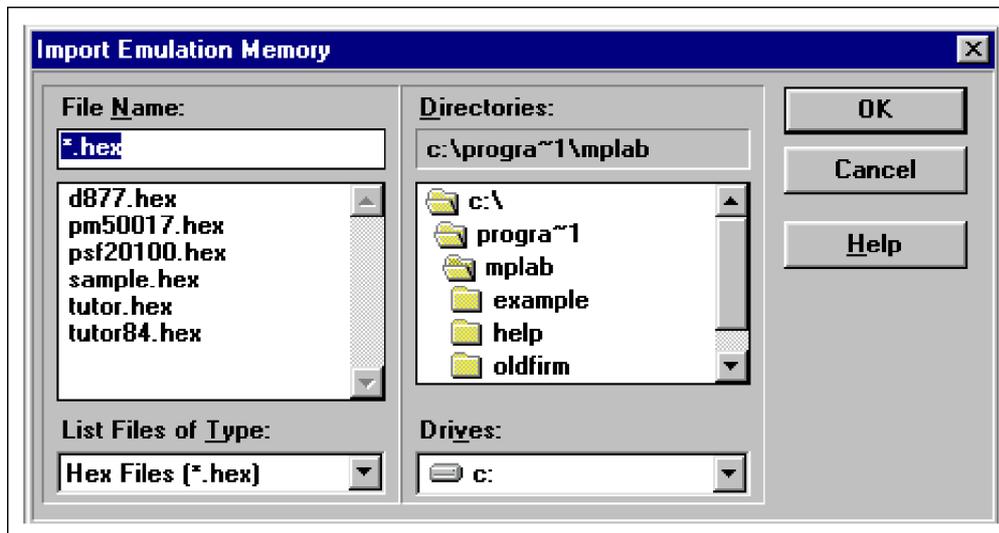
Если какой-либо файл был отредактирован, но не был сохранен, редактор MPLAB IDE выдаст предупреждение с возможностью отказа от закрытия файлов.

## 7.4.6 Импорт (File > Import)

Функция импорта позволяет перемещать данные из файла PC в память эмулятора или симулятора.

### 7.4.6.1 Импорт в память

Выполнив команду *File > Import > Import to Memory*, на экране появится диалоговое окно загрузки файла в память эмулятора или симулятора, показанное на рисунке. Загружаемый файл должен иметь шестнадцатеричный формат. Файл, подготовленный для программирования PICmicro, может быть загружен в память программ MPLAB IDE. Чтобы создать шестнадцатеричный файл, необходимо исполнить проект MPLAB IDE или выполнить экспорт памяти программ.



Для микроконтроллеров семейства PIC17CXXX и PIC18CXXX HEX файл должен быть в расширенном формате Intel INHX32. Для всех остальных семейств микроконтроллеров HEX файл должен быть в 8-разрядном формате Intel INHX8M. Если к .HEX файлу есть соответствующий файл объектного кода \*.COD, то будут автоматически загружены все символы.

Если у Вас нет соответствующего \*.COD файла, то можно отключить прослеживание в меню *Options > Environment Setup* раздел *General* пункт *Track Source Code*.

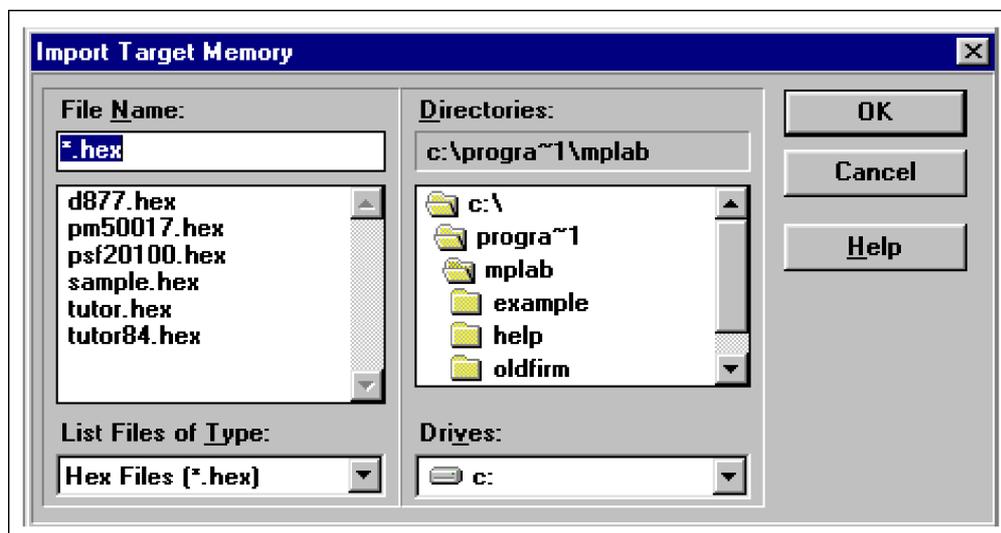
**Примечание.** Если объем загружаемого файла более 64Кбайт (32Кслов для PIC17CXXX) или он содержит биты конфигурации, то необходимо использовать INHX32 формат .HEX файла.

#### 7.4.6.2 Импорт во внешнюю память

Микроконтроллеры семейства PIC17CXXX и PIC18CXXX, работающие в режимах расширенного микроконтроллера или микропроцессора, используют внешнюю память. Настройка работы микроконтроллера выполняется в меню *Options > Development Mode* раздел *Configuration*, выберите режим микропроцессора (Processor Mode). В разделе *Memory* укажите только внешнюю память (Off-Chip Memory).

Для данного режима указанных микроконтроллеров возможно импортирование файла формата INHX32 во внешнюю память.

Выберите пункт меню *File > Import > Import to Target Memory*, на экране появится диалоговое окно импортирования файла, показанное на рисунке.



#### 7.4.6.3 Копировать из внешней памяти

Для микроконтроллеров PIC17CXXX и PIC18CXXX, работающих с эмулятором в режиме расширенного микроконтроллера или микропроцессора, возможно копирование внешней памяти из устройства в память компьютера командой *File > Import > Copy from Target Memory*. Настройка работы микроконтроллера выполняется в меню *Options > Development Mode* раздел *Configuration*, выберите режим микропроцессора (Processor Mode). В разделе *Memory* укажите только внешнюю память (Off-Chip Memory).

Загруженный файл может быть откорректирован с помощью MPLAB IDE.

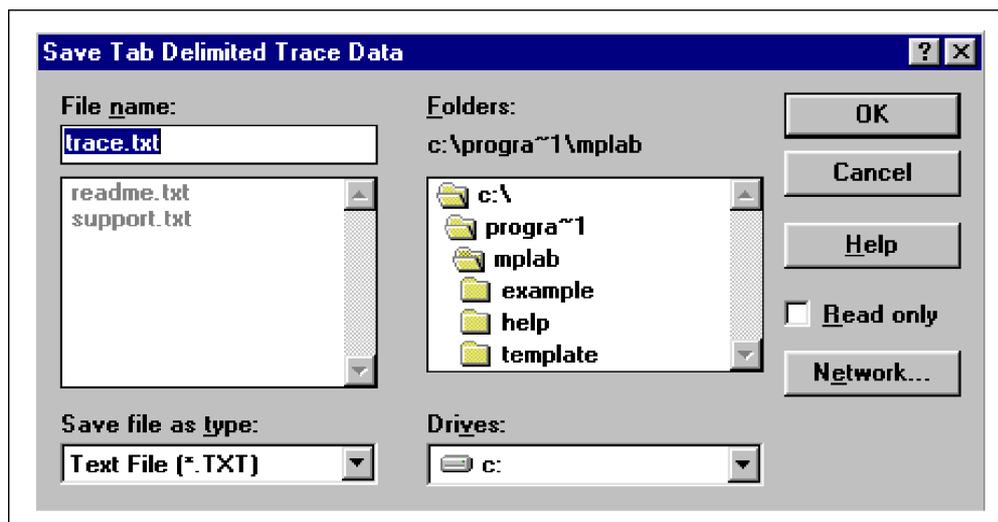
## 7.4.7 Экспорт файлов (File > Export)

### 7.4.7.1 Экспорт буфера трассировки

Командой *File > Export > Export Trace Buffer* можно сохранить буфер трассировки, полученный от эмулятора или симулятора. На экране появится диалоговое окно сохранения буфера трассировки.

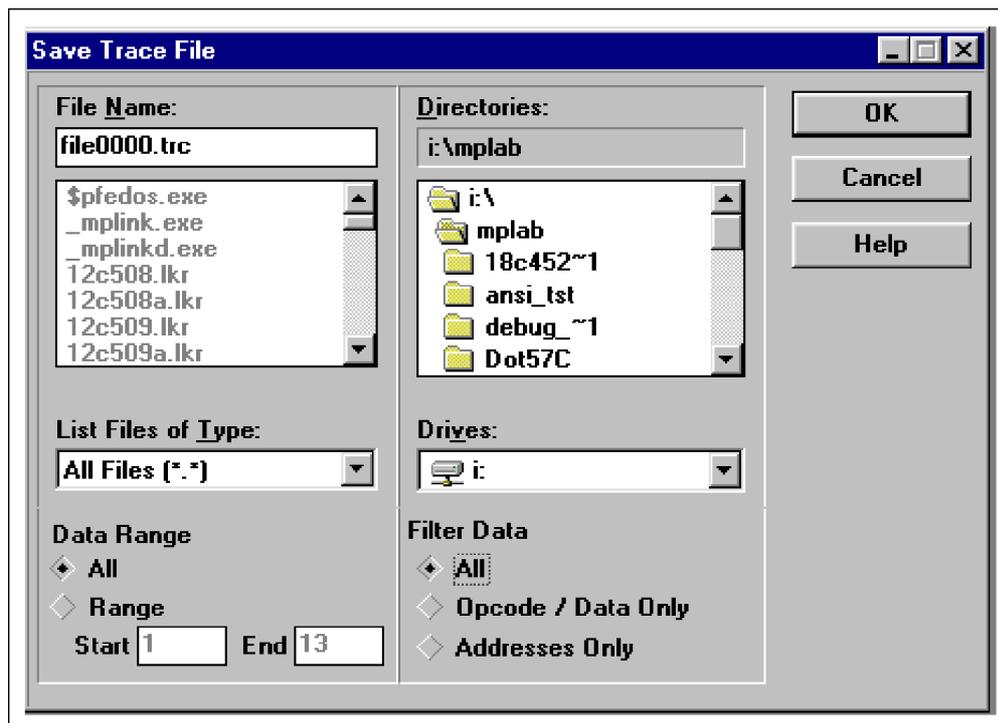
**Примечание.** Для эмулятора MPLAB ICE файл буфера трассировки может быть размером более 7Мб. Для симулятора MPLAB SIM файл буфера трассировки может быть размером более 1 Мб (сохраняется адрес, данные, дизассемблированный код и внешние логические указатели).

#### 7.4.7.1.1 Окно сохранения буфера трассировки для эмулятора MPLAB ICE



<i>File name</i>	Введите имя файла или выберите из списка, если хотите заменить существующий файл.
<i>Save file as type</i>	Укажите расширение файла буфера трассировки. В список будут включены только файлы с указанным расширением.
<i>Folders</i>	Выберите директорию, в которой хотите сохранить файл.
<i>Drives</i>	Выберите диск, на котором хотите сохранить файл.
<i>OK</i>	Нажмите <i>OK</i> для сохранения файла. Если вы заменяете существующий файл, редактор MPLAB выдаст предупреждение.
<i>Cancel</i>	Нажмите <i>Cancel</i> для отмены сохранения файла.
<i>Read Only</i>	Установите флаг <i>Read Only</i> , если Вы хотите запретить редактирование сохраняемого файла.

## 7.4.7.1.2 Окно сохранения буфера трассировки для симулятора MPLAB SIM.



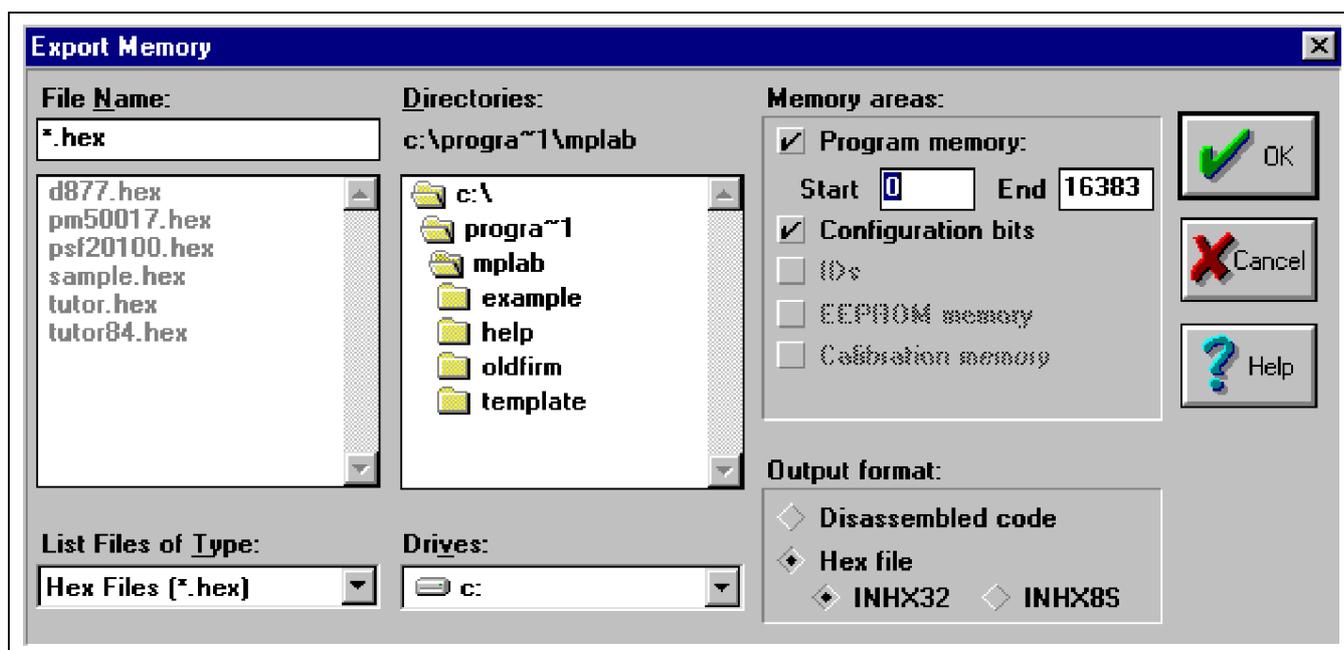
- Range* Для сохранения только части буфера выберите пункт *Range* и укажите диапазон (*Start – End*) от 0 до 8191.
- Filter Data: All* Сохранить весь буфер трассировки в выбранный файл.
- Filter Data: Opcode /Data only* Сохранять только код и данные (для PIC17CXXX циклы внешней записи/чтения). Доступно только с эмулятором MPLAB-ICD.
- Filter Data: Address only* Сохранять только адреса.

#### 7.4.7.2 Экспорт памяти

Для экспортирования памяти выберет пункт меню *File > Export > Export Memory*, на экране появится диалоговое окно, показанное на рисунке.

Если у Вас есть запрограммированный микроконтроллер PICmicro, и Вы хотите его повторить в других устройствах, прочитайте память программ микроконтроллера с помощью программатора (выберите пункт Read Device в меню программатора), а затем сохраните память программ в шестнадцатеричном файле - *File > Export > Export Memory*.

**Примечание.** Когда экспортируется память программ, в файл включаются пустые места. При создании HEX файла во время компиляции пустые места игнорируются.



#### Memory areas

В этой части диалогового окна необходимо определить, какую область памяти вы хотите сохранить в файле.

##### *Program Memory*

Сохранить память программ. Можно указать начальный (Start) и конечный (End) адрес сохраняемой части. По умолчанию сохраняется вся память программ.

##### *Configuration Bits*

Сохранить биты конфигурации

##### *IDs*

Сохранить ID информацию, если возможно.

##### *EEPROM Memory*

Сохранить EEPROM память данных, если возможно.

##### *Calibration Memory*

Сохранить калибровочную информацию, если возможно.

#### Output format

Указать формат сохраняемого файла.

##### *Disassembled Code*

Сохраняется дизассемблированный код в текстовом формате.

##### *Hex Code*

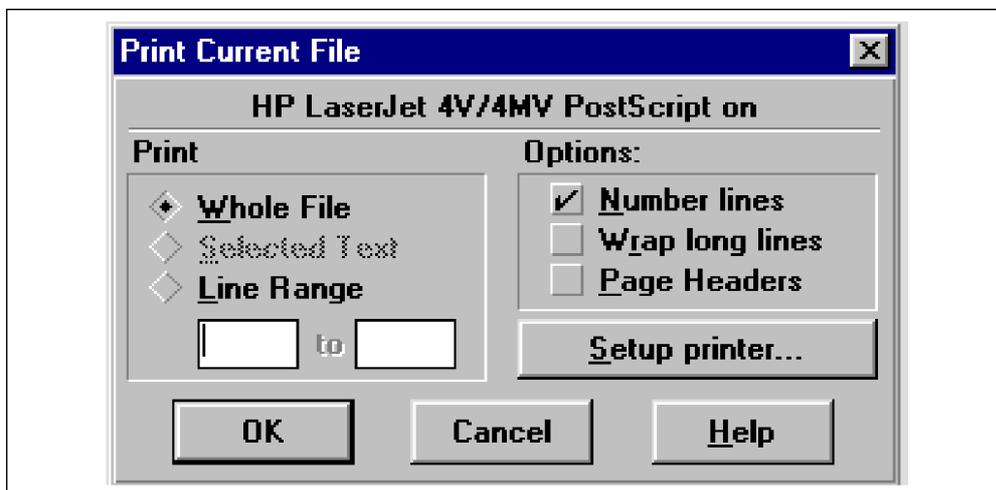
Сохранить в формате INHX32 или INHX8X .HEX файл.

## 7.4.8 Печать (Ctrl + P)

Команда *File > Print* позволяет распечатать целиком или частично файл из активного окна. На экране появится диалоговое окно, в котором можно выбрать требуемый принтер и объем печати.

По умолчанию MPLAB IDE использует принтер, который был выбран при последней печати. Параметры печати по умолчанию (длина строки, колонтитулы, число пробелов в символе табуляции и нумерация строк) соответствуют настройкам текущего окна. Вы можете изменить эти настройки, выбрав пункт меню *Options > Current Editor Modes*.

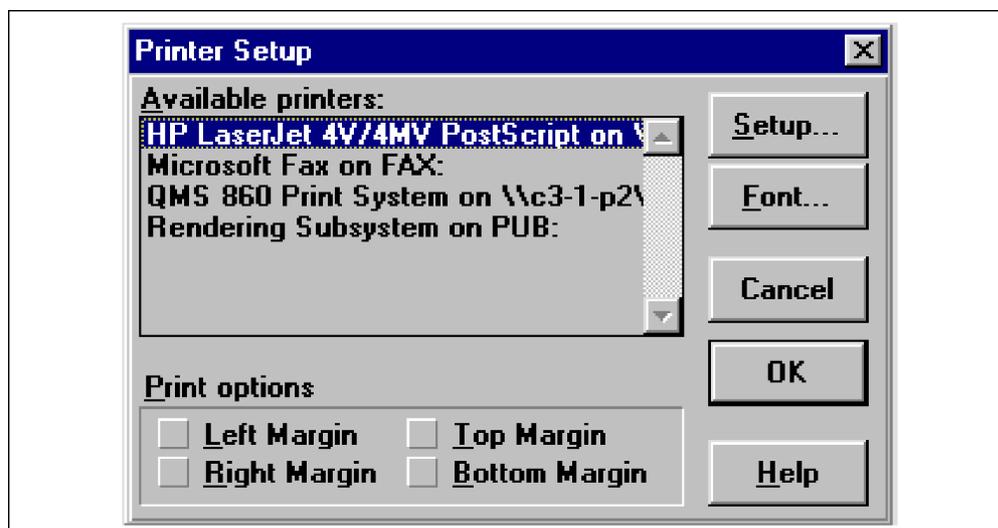
В диалоговом окне указывается имя принтера, используемого MPLAB IDE. Если Вы не определили ни один принтер, то будет использоваться принтер, назначенный по умолчанию в операционной системе.



<i>Setup Printer</i>	Нажмите <i>Setup Printer</i> для выбора принтера, шрифта печати и полей страницы.
<i>Whole File</i>	Печатать файл целиком (по умолчанию)
<i>Selected Text</i>	Печатать выделенный фрагмент текста. Эта опция доступна, если Вы выделили фрагмент текста.
<i>Line Range</i>	Печатать фрагмент файла. Необходимо указать первую ( <i>Start</i> ) и последнюю ( <i>End</i> ) строку фрагмента текста.
<i>Number Lines</i>	Нумеровать строки при печати.
<i>Wrap Long Lines</i>	Переносить не помещившиеся части длинной строки на другую строку.
<i>Page Headers</i>	Показывать меню настройки пред печатью следующего листа.

### 7.4.9 Настройка параметров печати (File > Print Setup)

Выберите пункта меню *File > Print Setup* для настройки параметров печати по умолчанию. MPLAB IDE сохраняет параметры настройки, указанные в окне Print Setup, для выбранного принтера. Таким образом, Вы можете иметь собственные настройки каждого используемого принтера.



В этом диалоговом окне можно установить:

- какой использовать принтер;
- поля печати;
- назначить шрифт печати.

<i>Available Printers</i>	Для настройки доступного принтера, выберите его нажатием левой кнопкой «мышки» на названии.
<i>Print Options</i>	Установите соответствующий флаг, чтобы заменять размер полей каждой страницы (верхнего, нижнего, левого и правого).
<i>Setup</i>	Открыть диалоговое окно настройки принтера операционной системы.
<i>Font</i>	Выбрать шрифт принтера, который будет использовать MPLAB IDE при печати.

Поскольку в MPLAB IDE встроен текстовый редактор (а не текстовый процессор), пользователи ограничены выбором только одного шрифта.

Если Вы выполняете настройку принтера в операционной системе, после указания шрифта в диалоговом окне MPLAB IDE, Вы можете обнаружить, что указанный шрифт больше не существует, а операционная система автоматически подставляет другой шрифт при печати файла.

### 7.4.10 Выход (File > Exit)

Выберите пункт меню *File > Exit* для прекращения работы с MPLAB IDE. Если существуют открытые, не сохраненные файлы, MPLAB IDE выдаст предупреждение. Вы можете выбрать: сохранять изменения в файлах, не сохраняя изменения в файлах, отказаться от закрытия MPLAB IDE. Перед выходом из среды проектирования будет предложено сохранить открытый проект.

### 7.4.11 Список ранее используемых файлов

В конце меню *File* сохраняется список нескольких файлов, которые использовались последним редактором MPLAB. Выбрав нужный файл, он автоматически откроется в окне среды проектирования MPLAB IDE.

Настройку количества файлов в списке смотрите в разделе 7.9.5.3.

## 7.5 Меню проекта (Project)

Следующие пункты доступны в этом меню:

<i>New Project</i>	- создать новый проект;
<i>Open Project</i>	- открыть сохраненный проект на диске;
<i>Close Project</i>	- закрыть открытый в настоящее время проект;
<i>Save Project</i>	- сохранить проект;
<i>Edit Project</i>	- редактировать параметры проекта;
<i>Make Project</i>	- выполнить проект
<i>Build All</i>	- компилировать все исходные файлы
<i>Build Node</i>	- компилировать исходный файл
<i>Install Language Tool</i>	- подключение модулей к проекту

Список последних открываемых проектов

Подробное описание работы с пунктами данного меню смотрите в главе 3.

## 7.6 Меню редактора MPLAB (Edit)

### 7.6.1 Команды меню редактора

Редактор может работать в режиме вставки текста (INS) или замены (OVR). Текущий режим отображается в линейке состояния.

#### 7.6.1.1 Отменить последнее действие – Ctrl + Z

Выберите *Edit > Undo*, чтобы отменить последнее действие. Если редактирование текста не было, команда не может быть выполнена, а ее выбор не доступен в меню.

Настройку количества действий, которые можно отменить смотрите в разделе 7.9.5.4.

#### 7.6.1.2 Вырезать (Cut) – Ctrl + X

Удаляет выделенный фрагмент текста, перемещая его в буфер обмена. После этой операции, Вы можете вставлять удаленный фрагмент текста в другом окне редактора MPLAB или в любой другой программе Windows.

#### 7.6.1.3 Копировать (Copy) – Ctrl + C

Копирует выделенный фрагмент текста в буфер обмена. После этой операции, Вы можете вставлять скопированный фрагмент текста в другом окне редактора MPLAB или в любой другой программе Windows.

#### 7.6.1.4 Вставить (Paste) – Ctrl + V

Вставляет содержимое буфера обмена в позицию курсора текущего окна. Вы можете выполнять эту операцию, если буфер обмена содержит фрагмент текста. Редактор MPLAB не поддерживает вставку рисунков и других форматов буфера обмена.

#### 7.6.1.5 Выделить все (Select All)

Выделяет весь текст в активном окне.

#### 7.6.1.6 Выделить слово (Select Word) - двойное нажатие левой кнопки «мыши»

Выделяет слово, в котором находится курсор.

#### 7.6.1.7 Удалить строку (Delete Line) - Ctrl + Shift + K

Удаляет строку, в которой находится курсор, перемещая курсор в начало следующей строки.

#### 7.6.1.8 Удалить текст до конца строки (Delete EOL) - Ctrl + K

Удаляет текст от курсора до конца строки. Если курсор находится в начале строки, то строка удаляется целиком, перемещая курсор в начало следующей строки.

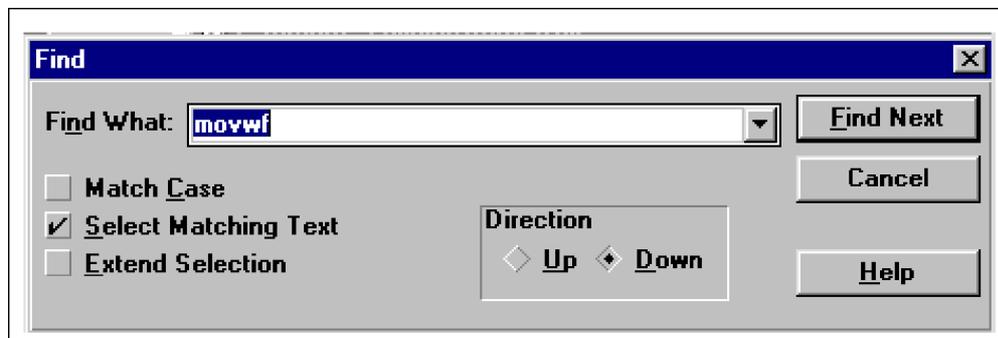
#### 7.6.1.9 Перейти на строку (Goto Line) – Ctrl + G

Перемещает курсор в начало строки, указанной косвенно или абсолютно.

#### 7.6.1.10 Найти (Find) - F3

Выполняет поиск текстовой последовательности в текущем окне редактора с параметрами, указанными в диалоговом окне. Если перед вызовом поиска выделить текстовую строку, то будет выполняться поиск этой строки.

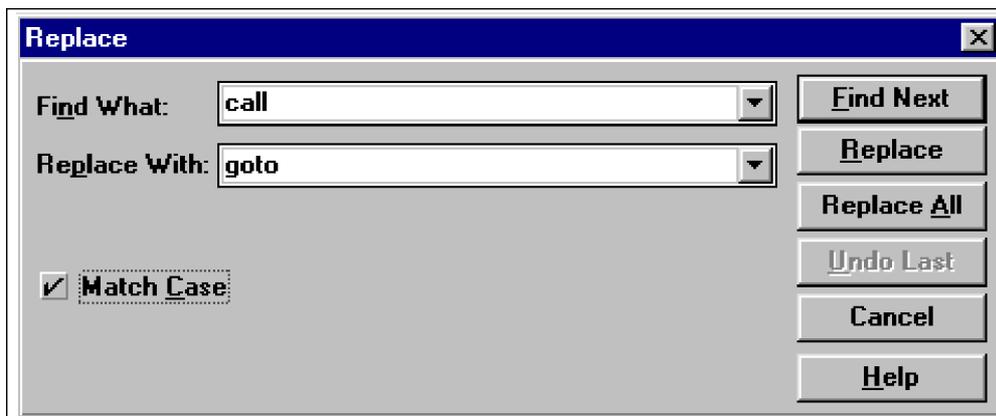
Чтобы выполнить поиск текста, установите курсор в место начала поиска и выберите пункт меню *Edit > Find*.



Используя данную функцию, Вы можете выполнять поиск специальных символов: «\n» перевод каретки, «\t» символ табуляции и «\» черта с наклоном влево.

#### 7.6.1.11 Замена (Replace) - F4

Команда выполняет поиск текстовой последовательности в текущем окне редактора и заменяет найденное другой текстовой последовательностью, указанной в диалоговом окне замены. Если перед вызовом замены выделить текстовую строку, то будет выполняться поиск этой строки.



Чтобы выполнить замену текста, установите курсор в место начала замены и выберите пункт меню *Edit > Replace*.

Используя данную функцию, Вы можете выполнять поиск и замену специальных символов: «\n» перевод каретки, «\t» символ табуляции и «\» черта с наклоном влево.

#### 7.6.1.12 Повторный поиск (Repeat Find) - Shift + F3

Выберите *Edit > Repeat Find* для начала повторно поиска с параметрами, указанными в диалоговом окне поиска.

#### 7.6.1.13 Повторная замена (Repeat Replace) - Shift + F4

Выберите *Edit > Repeat Replace* для начала повторной замены с параметрами, указанными в диалоговом окне замены текста.

#### 7.6.1.14 Установить уровень скобки – Ctrl + B

Редактор MPLAB позволяет управлять символами скобок (круглая скобка, фигурная скобка), которые часто разграничивают разделы исходного текста программы.

Символ фигурной скобки изменяется в зависимости от выбранного языка программирования для окна редактора. Для языка C фигурные скобки имеют синтаксическое значение: открывающиеся скобки {[ или (, закрывающиеся скобки ]} или ).

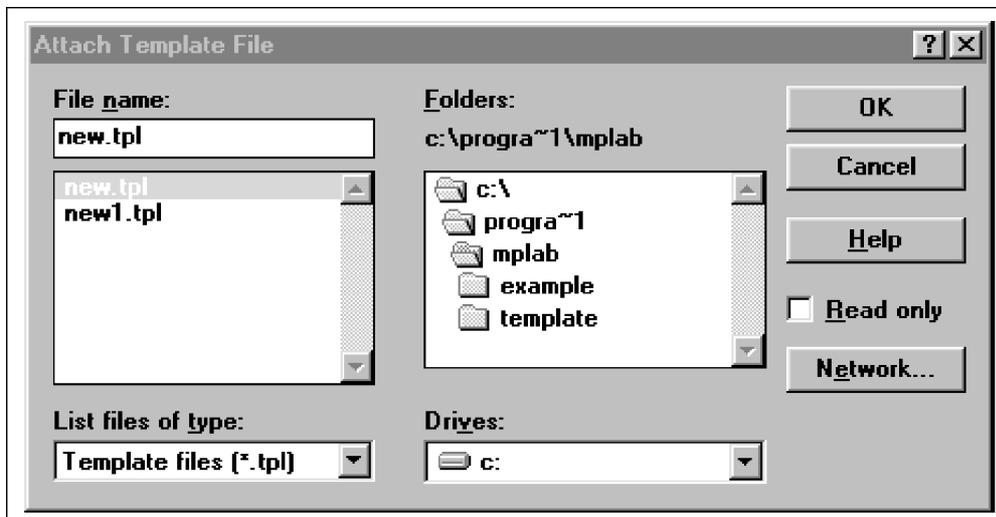
Если установлен тип языка программирования «поне», то символ фигурной скобки не определен. В других языках программирования могут использоваться открывающиеся скобки {{{ или <, закрывающиеся }}} или >.

Чтобы установить уровень фигурной скобки, установите курсор около скобки и выберите пункт меню *Edit>Match Brace*. Скобка будет перемещена на соответствующий уровень вложенности текста программы.

## 7.6.2 Работа с шаблонами

### 7.6.2.1 Подключение файла шаблона к проекту

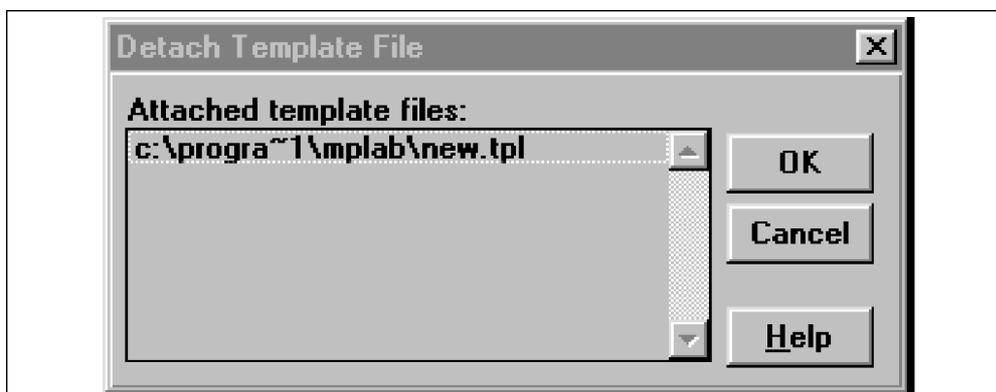
*Edit > Template > Attach File* - подключает существующий файл шаблона (.TPL), загружая его в память, что дает возможность использовать шаблоны в проекте. Выберите подключаемый файл и нажмите **OK**.



### 7.6.2.2 Отключение файла шаблона от проекта

Пункт меню *Edit > Template > Detach File* позволяет удалить файла шаблона из памяти. Как только файл шаблона отключен, Вы не сможете использовать шаблоны, содержащиеся в этом файле.

Выберите *Edit > Template > Detach File*, укажите файл шаблона, который Вы хотите отсоединить и нажмите кнопку **OK**. Вы не сможете отсоединить файл шаблона, если редактируется один из его шаблонов.



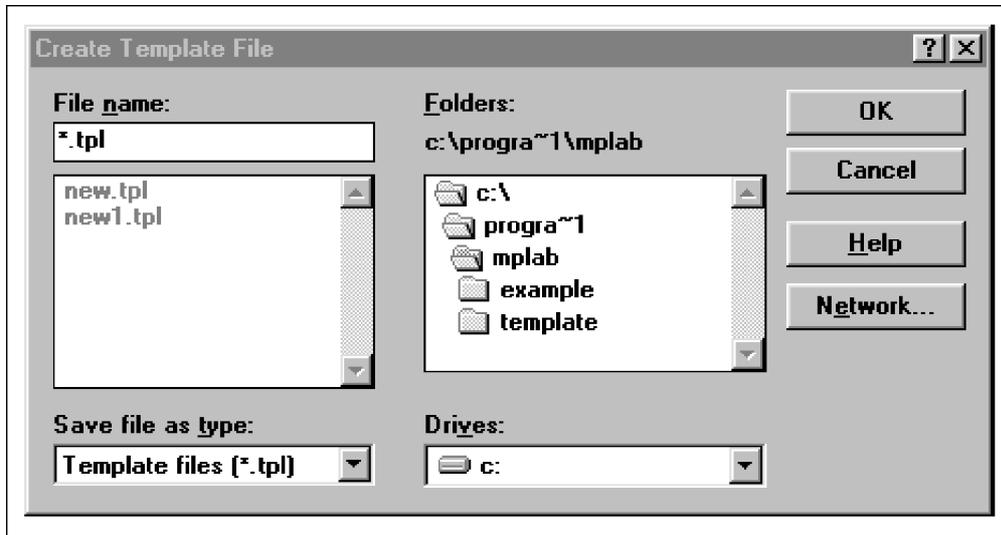
### 7.6.2.3 Создание файла шаблона

Первым шагом в подготовке шаблонов должно быть создание файла, в котором будут храниться шаблоны.

*Edit > Template > Create File* – создать файл шаблона с расширением .TPL.

Введите имя файла, укажите директорию и нажмите кнопку *OK*. Будет создан двоичный файл шаблона.

Если необходимо, чтобы шаблоны были доступны во всех проектах MPLAB IDE автоматически, назовите файл шаблона *auto.plt*.



### 7.6.2.4 Сохранение файла шаблона

Когда Вы редактируете, создаете или удаляете шаблон, редактор MPLAB IDE изменяет копию файла шаблона в памяти компьютера, которая будет потеряна при выходе из MPLAB IDE.

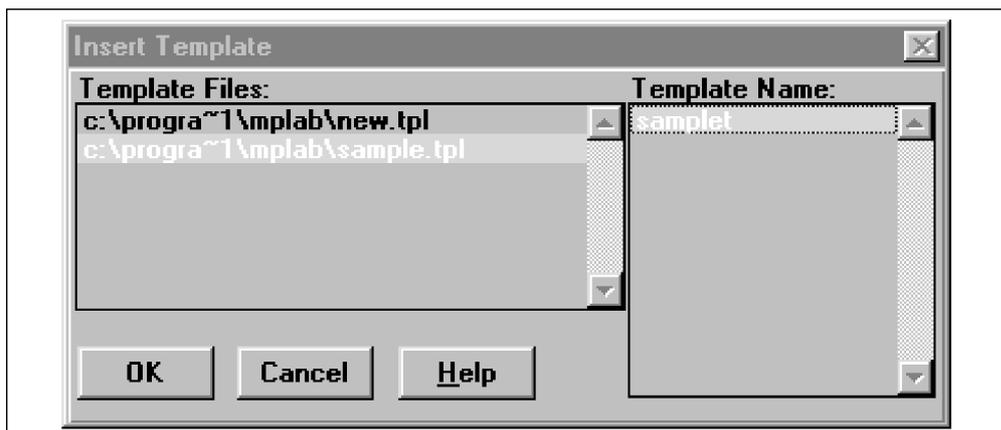
Изменения сохраняются на диске после выполнения команды *Edit > Template > Save File*.

Если Вы закрываете MPLAB IDE, а в памяти остались не сохраненные файлы шаблонов, на экране появится предупреждение. Нажмите кнопку *YES* для сохранения файлов шаблона .TPL.

### 7.6.2.5 Вставить шаблон

*Edit > Template > Insert* – вставить шаблон в исходный текст программы. Команда имеет смысл, после того как Вы подключили хотя бы один файл шаблона *Edit > Template > Attach File*.

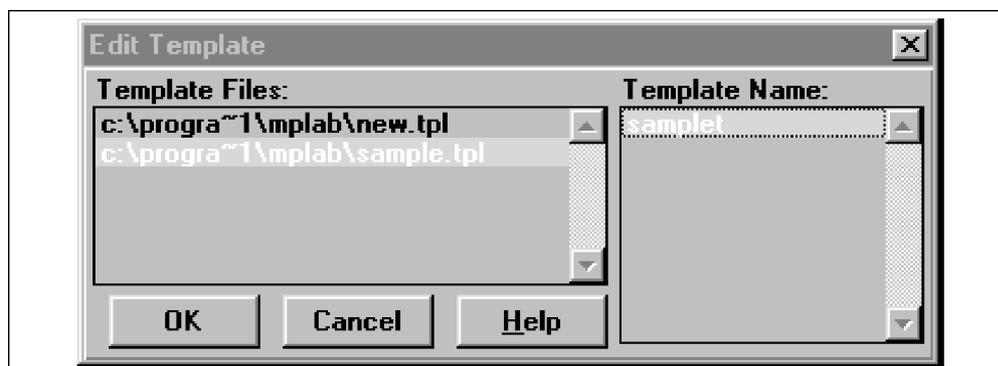
Установите курсор в исходном файле, где Вы хотите вставить шаблон. Выберите пункт меню *Edit > Template > Insert*, укажите нужный шаблон и нажмите кнопку *OK*.



#### 7.6.2.6 Редактирование шаблона

Чтобы изменить шаблон в файле шаблонов, воспользуйтесь пунктом меню *Edit > Template > Edit*, выберите нужный шаблон и нажмите кнопку *OK*. Команда имеет смысл, после того как Вы подключили хотя бы один файл шаблона *Edit > Template > Attach File*.

После выбора шаблона, он открывается в окне редактора MPLAB. В заголовке окна редактора будет указано имя шаблона. Выполнив редактирование шаблона, сохраните его командой *Edit > Template > Store*.



#### 7.6.2.7 Создание нового шаблона

Как только Вы создали файл шаблона .TPL, в котором будут храниться шаблоны, необходимо добавить текстовые шаблоны. Вы можете использовать готовые шаблоны вместо написания собственных. Выберите пункт меню *Edit > Template > New*, чтобы создать новый шаблон. Редактор MPLAB откроет новое окно, в котором введете текст шаблона.

Вы можете воспользоваться готовыми шаблонами, которые включаются в пакет MPLAB IDE и находятся в директории MPLAB\TEMPLATES\CODE или MPLAB\TEMPLATES\OBJECT, перенеся текст шаблона командами *Edit > Select All*, *Edit > Copy* и *Edit > Paste*.

В текст шаблона можно добавить маркеры, показывая текст программы, который должен быть доработан пользователем при вставке шаблона (см. раздел 7.6.2.11). После создания шаблона убедитесь, что Вы его сохранили командами *Edit > Template > Store* или *Edit > Template > Store As* (см. разделы 7.6.2.8 и 7.6.2.9).

#### 7.6.2.8 Сохранение шаблона

*Edit > Template > Store* – сохранение шаблона в файле шаблона .TPL.

Этой командой записывается новая версия шаблона в файл шаблона.

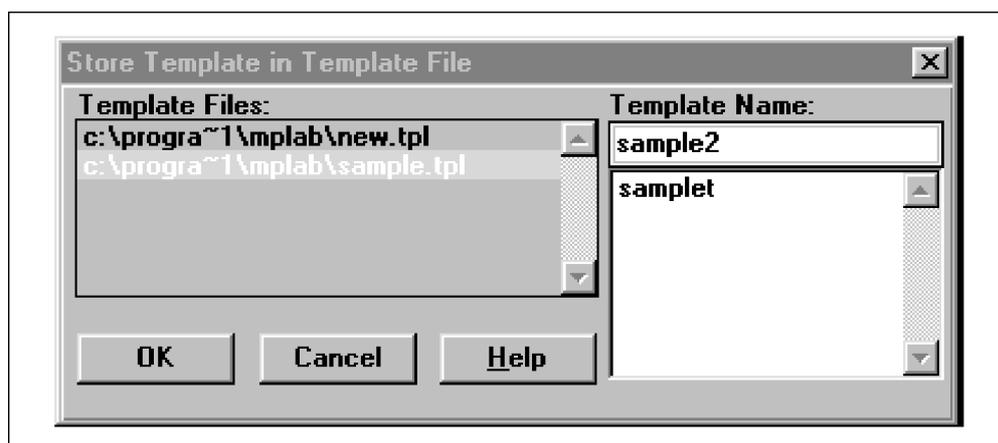
**Примечание.** Когда Вы редактируете, создаете или удаляете шаблон, редактор MPLAB IDE изменяет копию файла шаблона в памяти компьютера, которая будет потеряна при выходе из MPLAB IDE. Изменения сохраняются на диске после выполнения команды *Edit > Template > Save File*. Если Вы закрываете MPLAB IDE, а в памяти остались не сохраненные файлы шаблонов, на экране появится предупреждение. Нажмите кнопку *YES* для сохранения файлов шаблона .TPL.

#### 7.6.2.9 Сохранить шаблон как...

После ввода текста шаблона и добавления маркеров, сохраните его в файле шаблонов командой *Edit > Template > Store As*, указав имя шаблона. Вы можете использовать эту команду для сохранения отредактированного шаблона с новым именем в файле шаблонов.

Выберите пункт меню *Edit > Template > Store As*, укажите имя шаблона и нажмите кнопку *OK*. Если Вы заменяете существующий шаблон, редактор MPLAB выдаст предупреждение перед сохранением.

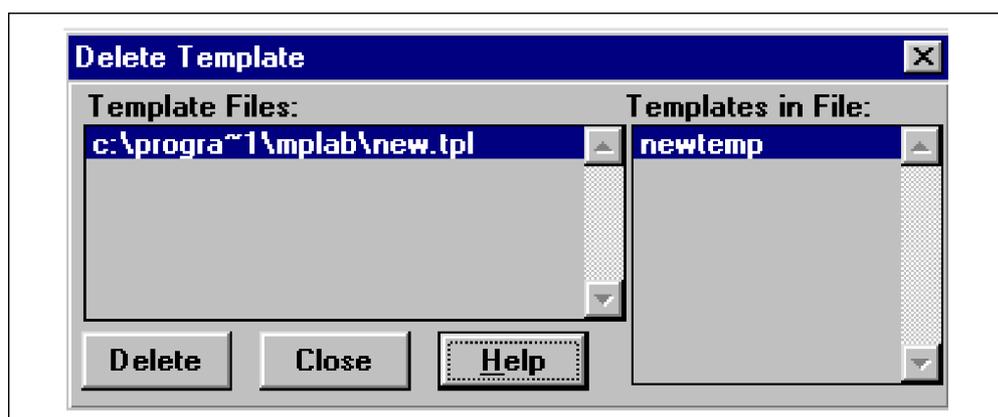
**Примечание.** Когда Вы редактируете, создаете или удаляете шаблон, редактор MPLAB IDE изменяет копию файла шаблона в памяти компьютера, которая будет потеряна при выходе из MPLAB IDE. Изменения сохраняются на диске после выполнения команды *Edit > Template > Save File*. Если Вы закрываете MPLAB IDE, а в памяти остались не сохраненные файлы шаблонов, на экране появится предупреждение. Нажмите кнопку *YES* для сохранения файлов шаблона .TPL.



#### 7.6.2.10 Удаление шаблона

Чтобы удалить шаблон, выберите пункт меню *Edit > Template > Delete*, укажите удаляемый шаблон и нажмите кнопку *OK* (файл шаблона должен быть уже подключен, см. раздел 7.6.2.1).

**Примечание.** Когда Вы редактируете, создаете или удаляете шаблон, редактор MPLAB IDE изменяет копию файла шаблона в памяти компьютера, которая будет потеряна при выходе из MPLAB IDE. Изменения сохраняются на диске после выполнения команды *Edit > Template > Save File*. Если Вы закрываете MPLAB IDE, а в памяти остались не сохраненные файлы шаблонов, на экране появится предупреждение. Нажмите кнопку *YES* для сохранения файлов шаблона .TPL.



#### 7.6.2.11 Добавить маркер в шаблон

В текст шаблона можно добавить маркеры, показывая текст программы, который должен быть доработан пользователем при вставке шаблона.

Поместите курсор в требуемом месте шаблона, выберите пункт меню *Edit > Template > Insert Mark*, чтобы вставить символы маркера <???.>. Эти маркеры появятся в исходном тексте программы, облегчая работу разработчика.

#### 7.6.2.12 Поиск маркеров в шаблоне

После того как Вы вставили шаблон (раздел 7.6.2.5), по команде *Edit > Template > Find Mark* производится поиск маркеров шаблона. Как уже было написано в предыдущем разделе, маркеры идентифицирует место в шаблоне, где необходимо вписать текст программы. Введите текст программы вместо каждого маркера.

### 7.6.3 Обработка текста

Это подменю содержит набор команд для работы с текстом.

#### 7.6.3.1 *Переместить (Transpose) – Ctrl + T*

*Edit > Text > Transpose (Ctrl + T)* – поменять местами символы, расположенные слева и справа от курсора. Команда не имеет действие, если курсор расположен в начале или в конце строки.

#### 7.6.3.2 *Верхний регистр (Uppercase)*

*Edit > Text > Uppercase* – в выделенном фрагменте текста заменить, все символы нижнего регистра, на соответствующие символы верхнего регистра.

#### 7.6.3.3 *Нижний регистр (Lowercase)*

*Edit > Text > Lowercase* – в выделенном фрагменте текста заменить, все символы верхнего регистра, на соответствующие символы нижнего регистра.

#### 7.6.3.4 *Отступ вправо (Text Indent)*

*Edit > Text Indent* – сместить начало текста выделенных строк (или строки, в которой установлен курсор) на один символ табуляции вправо.

#### 7.6.3.5 *Отступ влево (Text UnIndent)*

*Edit > Text UnIndent* – сместить начало текста выделенных строк (или строки, в которой установлен курсор) на один символ табуляции влево. Команда не распространяется на текст, который расположен в начале строки.

## 7.7 Меню отладки (Debug)

В меню Debug расположены все необходимые команды для отладки текста программы.

### 7.7.1 Подменю Run

В подменю Run находятся команды, управляющие работой микроконтроллера.

#### 7.7.1.1 Запуск программы (Run) – F9

*Debug > Run > Run* – запускает выполнение программы. Программа выполняется до тех пор, пока не будет удовлетворено одно из условий остановки.

Выполнение программы начинается с текущего значения счетчика команд, отображаемого в линейке состояния. Текущее значение счетчика команд также указывается в окне памяти программ. Во время выполнения программы кнопки Step и Run заблокированы.

#### 7.7.1.2 Сброс (Reset) – F6

*Debug > Run > Reset* – сброс микроконтроллера, аналогично сбросу по сигналу -MCLR (в счетчик команд записывается адрес вектора сброса). Если во время сброса программа исполнялась, то она продолжит выполняться с адреса вектора сброса.

#### 7.7.1.3 Остановить (Halt) – F5

*Debug > Run > Halt* – остановить ход выполнения программы (счетчик команд остановлен). Информация о состоянии микроконтроллера будет обновлена.

#### 7.7.1.4 Остановить запись в буфер трассировки (Halt Trace) – Shift + F5

*Debug > Run > Halt Trace* – прекратить запись в буфер трассировки (выполнение программы продолжается). Дополнительную информацию смотрите в технической документации на эмулятор.

#### 7.7.1.5 Запустить программу в режиме анимации (Animate)

*Debug > Run > Animate* – симулятор работает непрерывно, обновляя значение регистров после выполнения каждой инструкции программы. Выполнение программы в режиме анимации значительно медленнее, чем по команде *Debug > Run > Run*, но позволяет контролировать состояние регистров микроконтроллера и значения секундомера.

#### 7.7.1.6 Выполнить одну инструкцию программы (Step) – F7

*Debug > Run > Step* – по этой команде выполняется одна инструкция микроконтроллера (одна или несколько инструкций цикла), а затем программа останавливается, обновляя значение регистров. Во время работы в «режиме реального времени» кнопка Step игнорируется.

#### 7.7.1.7 Выполнить одну инструкцию программы (Step Over) – F8

*Debug > Run > Step Over* – выполнить текущую инструкцию программы. По команде Step Over инструкции CALL выполняются за один шаг (полностью выполнив подпрограмму). Затем выполнение программы останавливается в следующем за инструкцией CALL адресе.

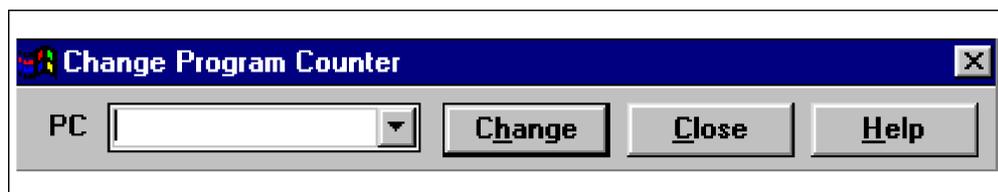
#### 7.7.1.8 Обновить значение всех регистров (Update All Registers)

*Debug > Run > Update All Registers* – обновить значение всех регистров для текущей инструкции.

#### 7.7.1.9. Изменить значение счетчика команд PC (Change Program Counter)

*Debug > Run > Change Program Counter* – изменить значение счетчика команд PC.

При вызове команды на экране появляется окно, показанное на рисунке.



<i>PC</i>	Введите значение счетчика команд PC.
<i>Change</i>	Записать в счетчик команд новое значение. Для изменения значения счетчика команд микроконтроллер должен быть остановлен.
<i>Close</i>	Закреть диалоговое окно.

## 7.7.2 Подменю Execute

Пункты подменю Execute позволяют Вам управлять выполнением программы микроконтроллера.

### 7.7.2.1 Выполнить отдельную инструкцию (Execute an Opcode)

Выберите *Debug > Execute > Execute an Opcode* для исполнения одной или нескольких инструкций без изменения текста и кода программы или объектного кода. После выполнения отдельной инструкции вы можете запустить программу с текущего места. Исполнение отдельной инструкции не является выполнением шага программы, значение секундомера не изменяется, счетчик команд не инкрементируется.

**Примечание.** Этот пункт меню не может использоваться для выполнения двух-словных инструкций микроконтроллеров PIC18CXXX.



*Opcode*

Введите код инструкции (в шестнадцатеричном виде) или мнемонику инструкции (например, ADDWF 0x19). В списке сохраняются 8 последних выполненных инструкций. MPLAB IDE автоматически выделяет выполненную инструкцию, чтобы была возможность сразу ввести новую (MPLAB IDE контролирует выполняемые инструкции, поэтому в списке не будет две одинаковых инструкции).

*Execute*

Выполнить инструкцию без изменения счетчика команд PC.

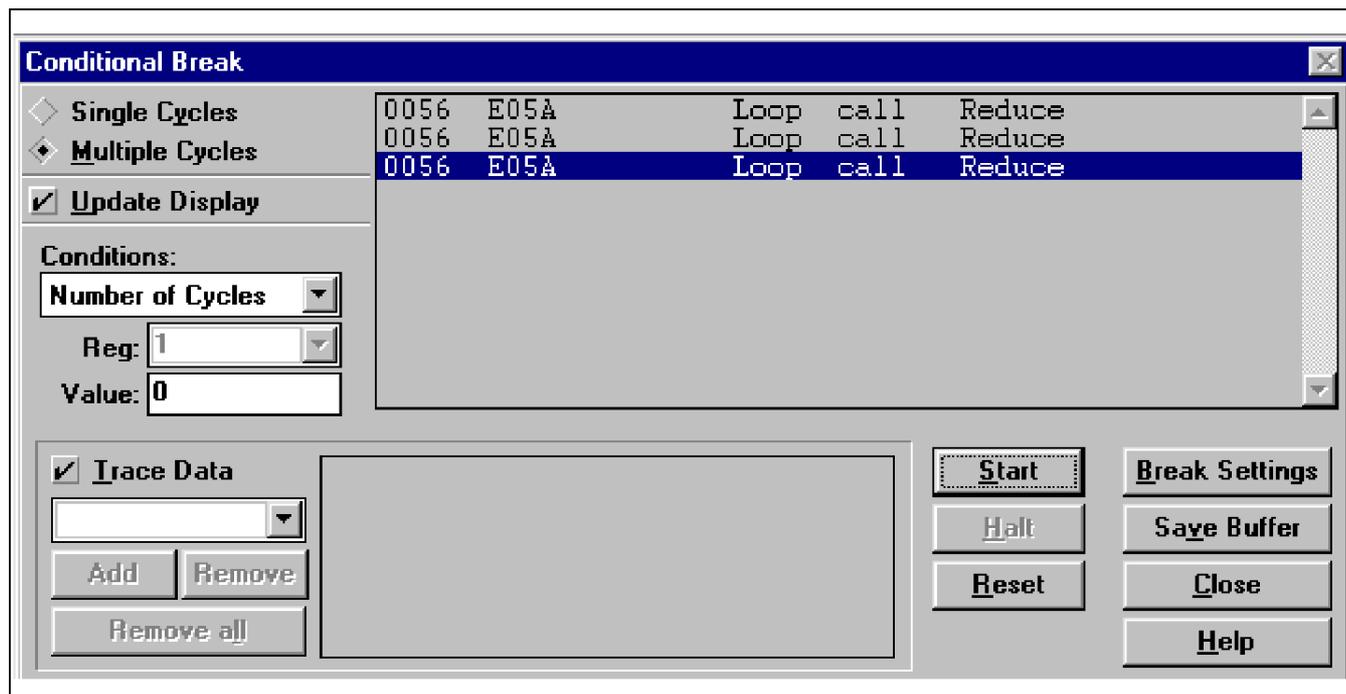
### 7.7.2.2 Условная остановка программы

Выберите *Debug > Execute > Conditional Break* на экране появится диалоговое окно, показанное на рисунке.

Пошаговое выполнение программы начинается после нажатия кнопки Start, а останавливается, когда удовлетворено условие остановки или нажата кнопка Halt.

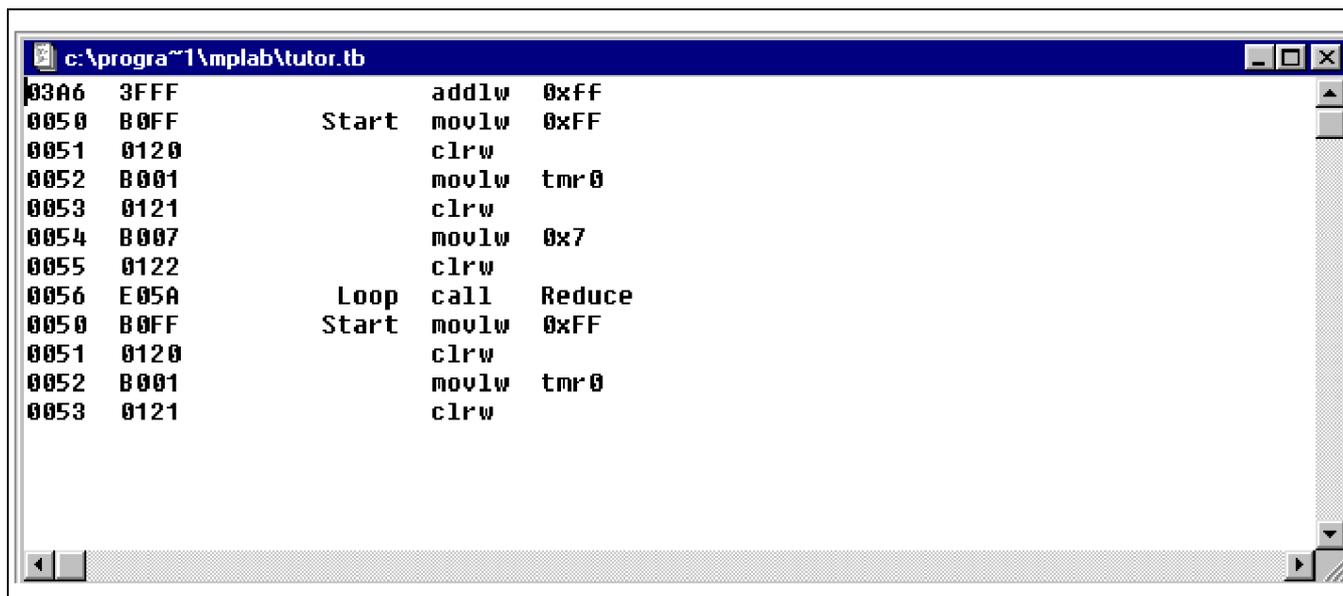
**Примечание.** Диалоговое окно условной остановки не доступно для MPLAB ICE и MPLAB ICD. Для эмулятора MPLAB-ICE предусмотрено диалоговое окно сложного триггера (Complex Trigger).

Дополнительную информацию по условным точкам остановки смотрите в разделе 6.8.



<i>Single Cycle</i>	Проверять условие остановки после выполнения каждой инструкции программы, что позволяет Вам обнаружить специфическое состояние микроконтроллера.
<i>Multiple Cycle</i>	Программа выполняется в масштабе реального времени с учетом установленных точек остановки.
<i>Update Display</i>	Выполнять остановку по условию, но не изменять дизассемблированный код в окне. MPLAB IDE сохраняет последнюю 1000 строк.
<i>Conditions</i>	<p>Условие остановки выполнения программы. Вы можете назначить 8-разрядное значение, которое будет использоваться при проверке содержимого указанного регистра. Возможны следующие условия остановки:</p> <ol style="list-style-type: none"> <li>1. Остановка пользователем. Когда программа выполняется, нажмите кнопку <i>Halt</i> для ее остановки.</li> <li>2. Выполнение указанного числа циклов. Введите требуемое количество циклов в строку <i>Value</i>.</li> <li>3. Контроль значения регистра: <ul style="list-style-type: none"> <li><i>RAM Addr Data Value ==</i> равняется указанному значению;</li> <li><i>RAM Addr Data Value &lt; &gt;</i> не равняется указанному значению;</li> <li><i>RAM Addr Data Value &gt;</i> больше указанного значения;</li> <li><i>RAM Addr Data Value &lt;</i> меньше указанного значения;</li> <li><i>RAM Addr Data Value &gt; =</i> больше или равно указанному значению;</li> <li><i>RAM Addr Data Value &lt; =</i> меньше или равно указанному значению.</li> </ul> </li> </ol> <p>Если указанное условие удовлетворено, выполнение программы останавливается перед следующей инструкцией, выделяя ее в окне памяти программ.</p> <p><b>Внимание.</b> Проверка условия производится с 8-разрядным знаковым значением, поэтому условие <math>&lt; 0</math> никогда не будет истинным.</p> <p><i>Req</i> – адрес регистра в реализованной памяти данных, значение которого будет проверяться.</p> <p><i>Value</i> – введите 8-разрядное значение.</p>
<i>Trace Data</i>	Производить запись значений указанных регистров в буфер трассировки.
<i>Add, Remove, Remove All</i>	Кнопки редактирования списка переменных. Доступны при остановке программы.
<i>Start</i>	Выполнять программу с текущего места, пока не будет выполнено условие остановки или нажата кнопка <i>Halt</i> .
<i>Halt</i>	Остановить выполнение программы.
<i>Reset</i>	Сбросить микроконтроллер.
<i>Break Settings</i>	Открыть диалоговое окно настройки точек остановки.
<i>Save Buffer</i>	Открыть диалоговое окно сохранения файлов, чтобы сохранить на диске буфер в *.TB файле.
<i>Close</i>	Закреть диалоговое окно условной остановки.

На рисунке показан пример файла \*.ТВ.



```
c:\progra~1\mplab\tutor.tb
03A6 3FFF          addlw 0xFF
0050 B0FF          Start movlw 0xFF
0051 0120          clrw
0052 B001          movlw tmr0
0053 0121          clrw
0054 B007          movlw 0x7
0055 0122          clrw
0056 E05A          Loop  call  Reduce
0050 B0FF          Start movlw 0xFF
0051 0120          clrw
0052 B001          movlw tmr0
0053 0121          clrw
```

### 7.7.3 Подменю Simulator Stimulus

MPLAB SIM позволяет моделировать состояние входов микроконтроллера с дискретностью в один машинный цикл. Текстовый файл стимула может быть написан в редакторе MPLAB или в любом другом подходящем редакторе.

Пункты подменю:

- Asynchronous Stimulus (асинхронный стимул);
- Pin Stimulus (стимул порта ввода/вывода);
- Clock Stimulus (стимул тактового сигнала);
- Register Stimulus (стимул регистра).

Подробное описание работы с подменю стимула смотрите в разделе 6.9.

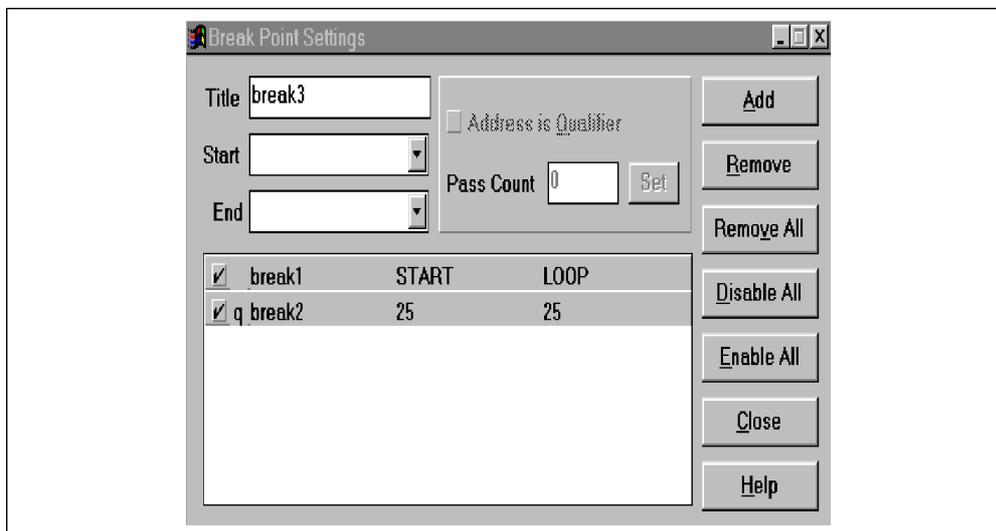
### 7.7.4 Указатель выполнения программы

*Debug > Center Debug Location* – переместить указатель выполнения программы в центр окна исходного текста программы, памяти программ и листинга программы.

## 7.7.5 Настройка точек останова (Debug > Break Settings)

Выберите *Debug > Break Settings* для настройки именованных точек останова. Дополнительную информацию по точкам останова смотрите в разделе 6.7.

**Примечание.** Если выполнение программы не прерывается в точке останова, проверьте глобальное разрешение останова программы *Global Break Enable* в пункте меню *Options > Development Mode - Break Options*.



### 7.7.5.1 Основные параметры точек останова

Вы можете назначить 16 именованных точек останова. Введите название точки останова, начальный адрес, конечный адрес (необязательно) и нажмите кнопку *Add*, чтобы запомнить параметры точки останова.

**Примечание.** При использовании эмулятора MPLAB-ICD Вы можете назначить только одну точку останова.

### 7.7.5.2. Сохранение настроек точек останова

Настройки точек останова сохраняются вместе с проектом MPLAB IDE.

<i>Title</i>	Введите символьное имя точки останова (максимальная длина 32 символа). MPLAB IDE допускает использование символов подчеркивания в имени точки останова (пробелы запрещены). MPLAB IDE автоматически назначит имя точки останова, если Вы его не указали. Если Вы назначаете диапазон адресов для точки останова, то имя указывается обязательно.
<i>Start, End</i>	Введите начальный и конечный адрес диапазона в шестнадцатеричном виде или символьной меткой. Возможные адреса ограничены физически реализованной памятью программ микроконтроллера. Вы можете указать один и тот же адрес для начала и конца диапазона, что позволяет назначить конкретный адрес в памяти программ, а не диапазон. Если Вы указываете адреса символьными метками, то можно задавать смещение относительно метки (например, "MAIN+2" или "EXESTIMR-10"). Когда Вы используете символьные метки и выполняете повторную компиляцию проекта, адреса точек останова автоматически переопределяются. Вы можете использовать настройки существующей точки останова для создания новой. Выберите нужную точку из списка, введите новое имя и нажмите кнопку <i>Add</i> .

**Break Point List**

Список позволяет Вам определить до 16 точек остановки. При выборе существующей точки остановки из списка, значения полей Title, Start и End будут доступны для редактирования. Список содержит следующие элементы:

	Title	Start	End
Enable/Disable	<input checked="" type="checkbox"/> break1	start	start
	<input checked="" type="checkbox"/> break3	ReduceADoubleB	
	<input checked="" type="checkbox"/> q end break	65	65
Qualifier	<input checked="" type="checkbox"/> middle break	start+2	loop-2

<b>Enable/Disable</b>	Включение/выключение точки остановки в памяти программ.
<b>Qualifier</b>	Если отображается символ 'q', то для диапазона адресов точки остановки назначен счетчик проходов.
<b>Title, Start, End</b>	Нажмите на элемент списка Title (имя), Start (начало диапазона) или End (конец диапазона) для редактирования.
<b>Add</b>	Добавить новую точку остановки или принять изменения. Необходимо указывать только физически реализованные адреса памяти программ. Введенное значение адреса не реализованной памяти программ очищено не будет.
<b>Remove</b>	Удалить выбранную точку остановки. Если не одна точка не выбрана, действие выполнено не будет.
<b>Remove All</b>	Удалить все точки остановки из списка.
<b>Disable All</b>	Отключить все точки остановки.
<b>Enable All</b>	Включить все точки остановки.
<b>Close</b>	Закрыть диалоговое окно настройки точек остановки.
<b>Help</b>	Показать дополнительную информацию.

**7.7.5.3 Дополнительные настройки точек остановки**

Описанные в этом разделе функции доступны только в эмуляторе PICMASTER и симуляторе MPLAB-SIM.

1. Установите флаг *Qualifier* для точки остановки из списка.
2. Введите значение счетчика проходов для точки остановки.

MPLAB IDE декрементирует 16-разрядный счетчик проходов при совпадении значения счетчика команд PC указанному диапазону адресов. Когда выполнение программы остановлено, Вы можете изменять значение счетчика прохода в диалоговом окне настройки точек остановки. Укажите начальное значение счетчика проходов и диапазон адресов. Когда значение счетчика проходов будет равно нулю, выполнение программы остановиться.

Счетчик декрементируется при каждом проходе адреса. Эта особенность может быть использована для определения числа проходов.

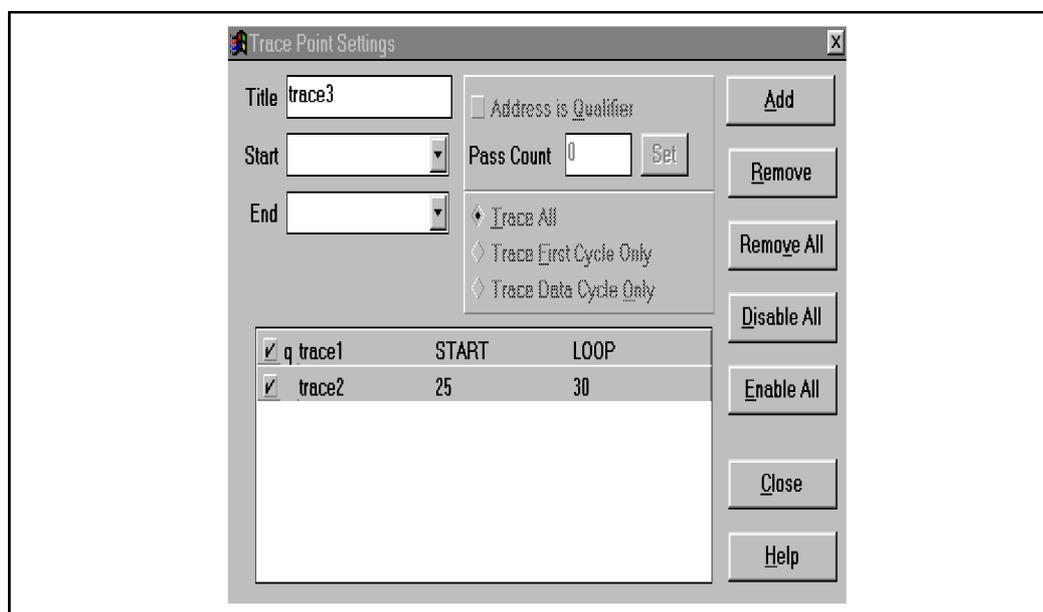
**Address is Qualifier**

Включить/выключить логику счетчика проходов. Выберите точку остановки из списка и установите флаг *Qualifier*. Рекомендуется указывать один адрес в памяти программ (а не диапазон) при использовании счетчика проходов. После того как установлен указанный флаг, MPLAB IDE допускает редактировать число проходов и нажимать кнопку *Set*. Максимальное число проходов, которое может быть назначено 65 534.

<i>Pass Count</i>	Введите число проходов. Значение «число проходов» определяет количество выполнения инструкции по указанному адресу перед остановкой программы. Каждый раз, выполняя инструкцию с указанным адресом, счетчик проходов декрементируется. Когда счетчик проходов обнулится, выполнение программы будет остановлено.
<i>Set</i>	Нажмите кнопку <i>Set</i> , чтобы сохранить число проходов. Отображаемое значение счетчика проходов не будет загружено в эмулятор, пока Вы не нажмете кнопку <i>Set</i> .

### 7.7.6 Настройка точек трассировки (Debug > Trace Settings)

Выберите *Debug > Trace Settings*, для открытия окна настройки 16 именованных точек трассировки. Настройка точек трассировки не доступна при работе с эмулятором MPLAB-ICE и MPLAB-ICD. Дополнительную информацию по точкам трассировки смотрите в разделе 6.7.



#### **Примечания.**

1. Для точки трассировки может быть назначен счетчик проходов.
2. Диалоговое окно настройки точек трассировки не доступно при работе с эмулятором MPLAB-ICE и MPLAB-ICD.

#### **7.7.6.1 Основные параметры точек трассировки**

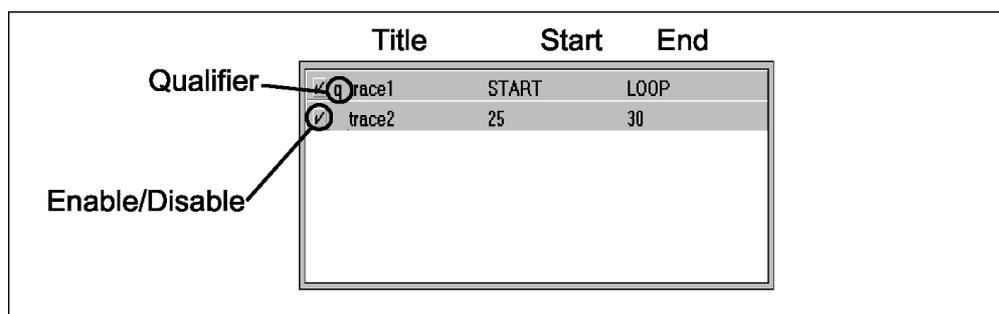
Вы можете назначить 16 именованных точек трассировки. Введите название точки трассировки, начальный адрес, конечный адрес (необязательно) и нажмите кнопку *Add*, чтобы запомнить параметры точки трассировки.

**Примечание.** MPLAB-ICD не имеет функций трассировки.

## 7.7.6.2. Сохранение настроек точек трассировки

Настройки точек трассировки сохраняются вместе с проектом MPLAB IDE.

<i>Title</i>	Введите символьное имя точки трассировки (максимальная длина 32 символа). MPLAB IDE допускает использование символов подчеркивания в имени точки трассировки (пробелы запрещены). MPLAB IDE автоматически назначит имя точки трассировки, если Вы его не указали. Если Вы назначаете диапазон адресов для точки трассировки, то имя указывается обязательно.
<i>Start, End</i>	Введите начальный и конечный адрес диапазона в шестнадцатеричном виде или символьной меткой. Возможные адреса ограничены физически реализованной памятью программ микроконтроллера. Вы можете указать один и тот же адрес для начала и конца диапазона, что позволяет назначить конкретный адрес в памяти программ, а не диапазон. Если Вы указываете адреса символьными метками, то можно задавать смещение относительно метки (например, "MAIN+2" или "EXESTIMR-10"). Когда Вы используете символьные метки и выполняете повторную компиляцию проекта, адреса точек трассировки автоматически переопределяются. Вы можете использовать настройки существующей точки трассировки для создания новой. Выберите нужную точку из списка, введите новое имя и нажмите кнопку <i>Add</i> .
<i>Break Point List</i>	Список позволяет Вам определить до 16 точек трассировки. При выборе существующей точки трассировки из списка, значения полей <i>Title</i> , <i>Start</i> и <i>End</i> будут доступны для редактирования. Список содержит следующие элементы:



<i>Enable/Disable</i>	Включение/выключение точки трассировки в памяти программ.
<i>Qualifier</i>	Если отображается символ 'q', то для диапазона адресов точки трассировки назначен счетчик проходов.
<i>Title, Start, End</i>	Нажмите на элемент списка <i>Title</i> (имя), <i>Start</i> (начало диапазона) или <i>End</i> (конец диапазона) для редактирования.
<i>Add</i>	Добавить новую точку трассировки или принять изменения. Необходимо указывать только физически реализованные адреса памяти программ. Введенное значение адреса не реализованной памяти программ очищено не будет.
<i>Remove</i>	Удалить выбранную точку трассировки. Если не одна точка не выбрана, действие выполнено не будет.
<i>Remove All</i>	Удалить все точки трассировки из списка.
<i>Disable All</i>	Отключить все точки трассировки.
<i>Enable All</i>	Включить все точки трассировки.
<i>Close</i>	Закрыть диалоговое окно настройки точек трассировки.
<i>Help</i>	Показать дополнительную информацию.

### 7.7.6.3 Дополнительные настройки точек трассировки

Описанные в этом разделе функции доступны только в эмуляторе PICMASTER и симуляторе MPLAB-SIM.

1. Установите флаг *Qualifier* для точки трассировки из списка.
2. Введите значение счетчика проходов для точки трассировки.

MPLAB IDE декрементирует 16-разрядный счетчик проходов при совпадении значения счетчика команд PC, указанному диапазону адресов. Когда выполнение программы остановлено, Вы можете изменять значение счетчика прохода в диалоговом окне настройки точек трассировки. Укажите начальное значение счетчика проходов и диапазон адресов. Когда значение счетчика проходов обнулится, произойдет запись в буфер трассировки.

Счетчик декрементируется при каждом проходе адреса. Эта особенность может быть использована для определения числа проходов.

<i>Address is Qualifier</i>	Включить/выключить логику счетчика проходов. Выберите точку трассировки из списка и установите флаг <i>Qualifier</i> . Рекомендуется указывать один адрес в памяти программ (а не диапазон) при использовании счетчика проходов. После того как установлен указанный флаг, MPLAB IDE допускает редактировать число проходов и нажимать кнопку <i>Set</i> . Максимальное число проходов, которое может быть назначено 65 534.
<i>Pass Count</i>	Введите число проходов. Значение «число проходов» определяет количество выполнения инструкции по указанному адресу перед записью в буфер трассировки. Каждый раз, выполняя инструкцию с указанным адресом, счетчик проходов декрементируется. Когда счетчик проходов обнулится, произойдет запись в буфер трассировки.
<i>Set</i>	Нажмите кнопку <i>Set</i> , чтобы сохранить число проходов. Отображаемое значение счетчика проходов не будет загружено в эмулятор, пока Вы не нажмете кнопку <i>Set</i> .

#### 7.7.6.4 Глобальные настройки точки трассировки

Данные настройки относятся к эмулятору PICMASTER и симулятору MPLAB SIM.

<i>Trace All</i>	Доступно для всех семейств PICmicro. Сохранять в буфере трассировки состояние микроконтроллера после выполнения каждой инструкции.
<i>Trace First Cycle Only</i>	Доступно для всех семейств PICmicro. Сохранение в буфере трассировки происходит только при первоначальном исполнении инструкции, исключая вложенные пустые циклы (NOP). Такой метод позволяет записать в буфер больше полезной информации.
<i>Trace Data Cycle Only</i>	Предназначено для микроконтроллеров семейства PIC17CXXX. Запись в буфер трассировки происходит на втором цикле выполнения команд табличного чтения/записи (TABLRD и TABLWR). Используйте эту особенность для чтения таблицы данных или для контроля значений регистров памяти данных.

Сохранить буфер трассировки можно командой *File > Export > Export Trace Buffer*.

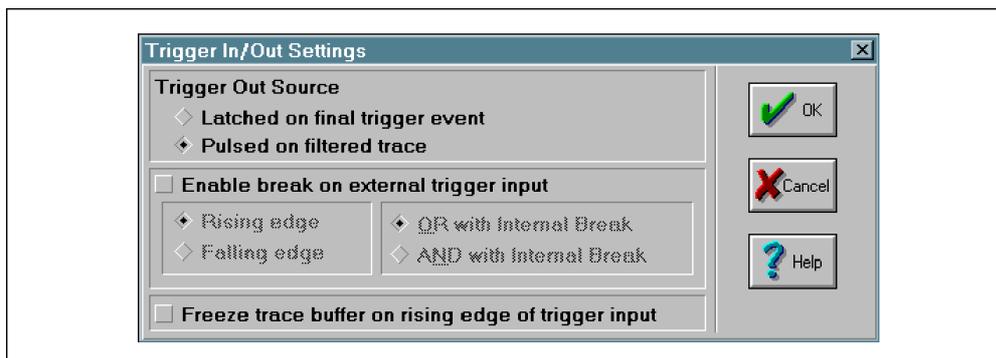
На рисунке показан пример буфера точек трассировки для симулятора MPLAB SIM. Вид окна буфера трассировки для эмулятора смотрите в технической документации на эмулятор.

```

Trace Memory Window
1 CALL Reduce ; Perform the inner portion of 117608.00us
2 0056 E05A Loop call Reduce ; 117608.00us 58804 |
3 DECFSZ OuterLoop,f ; the loop. 117612.00us
4 0057 1722 decfsz 0x22 ; 117612.00us 58806 |[ 57 ]
5 GOTO Loop 117642.00us
6 0058 C056 goto Loop ; 117642.00us 58821 |22:01 04:I
7 CALL Reduce ; Perform the inner portion of 117646.00us
8 0056 E05A Loop call Reduce ; 117646.00us 58823 |
9 DECFSZ OuterLoop,f ; the loop. 117650.00us
10 0057 1722 decfsz 0x22 ; 117650.00us 58825 |[ 57 ]
11 GOTO Loop 117682.00us
12 0058 C056 goto Loop ; 117682.00us 58841 |22:00 04:I
13 CALL Reduce ; Perform the inner portion of 117686.00us
14 0056 E05A Loop call Reduce ; 117686.00us 58843 |
15 DECFSZ OuterLoop,f ; the loop. 117688.00us
16 0057 1722 decfsz 0x22 ; 117688.00us 58844 |0A:FF
17 GOTO Loop 117690.00us
18 0058 C056 goto Loop ; 117690.00us 58845 |20:FF 04:I
19 CALL Reduce ; Perform the inner portion of 117692.00us
20 0056 E05A Loop call Reduce ; 117692.00us 58846 |0A:01
21 DECFSZ OuterLoop,f ; the loop. 117694.00us
22 0057 1722 decfsz 0x22 ; 117694.00us 58847 |21:01 04:I
23 GOTO Loop 117702.00us
24 0058 C056 goto Loop ; 117702.00us 58851 |[ 57 ]
25 CALL Reduce ; Perform the inner portion of 117732.00us
26 0056 E05A Loop call Reduce ; 117732.00us 58866 |22:06 04:I
27 DECFSZ OuterLoop,f ; the loop. 117736.00us

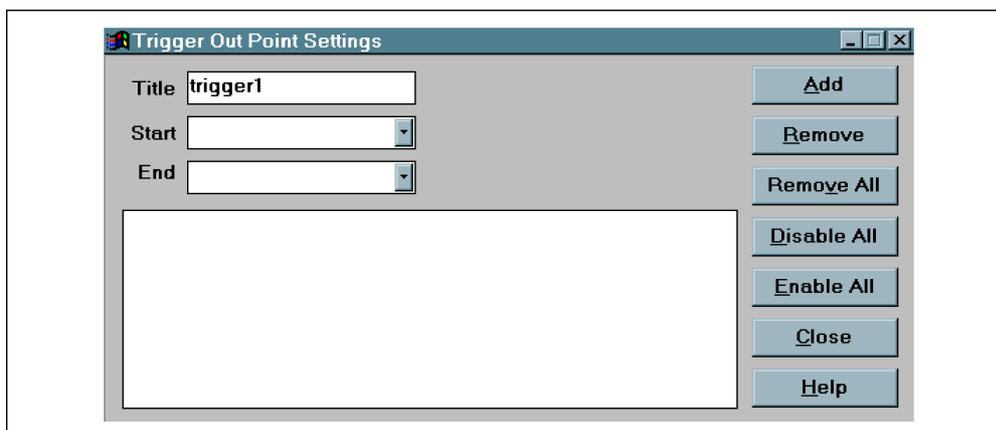
```

## 7.7.7 Debug &gt; Trigger In/Out Settings – настройка триггера ввода/вывода.



**Примечание.** Работает только с MPLAB ICE и эмулятором PICMASTER.  
Подробное описание смотрите в документации MPLAB-ICE User's Guide or the PICMASTER User's Guide.

## 7.7.8 Debug &gt; Trigger Output Points – точка триггера.

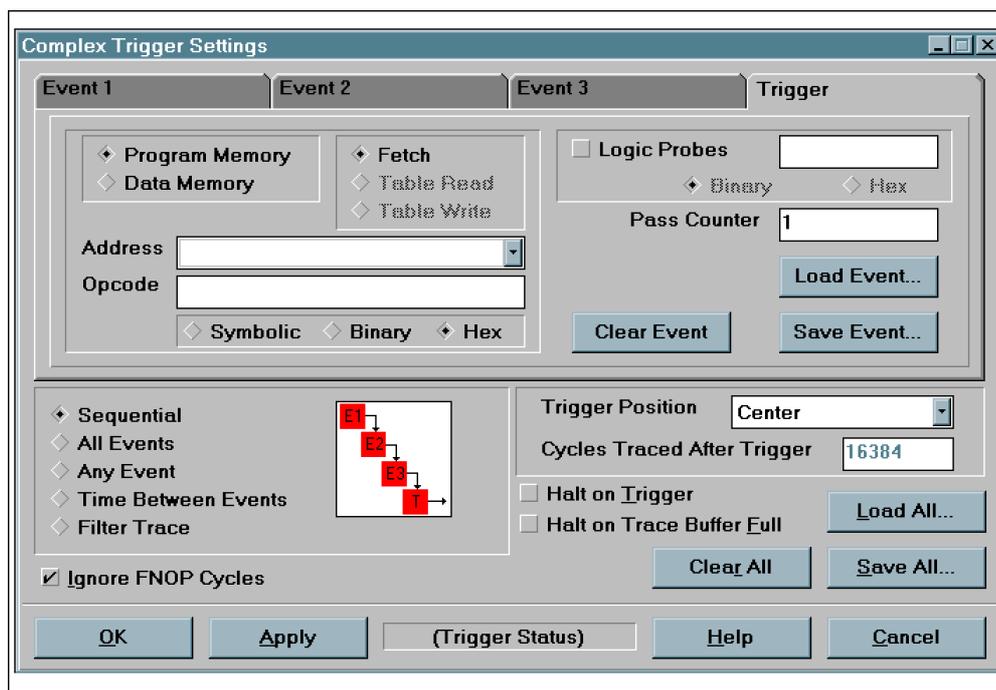


**Примечание.** Работает только с эмулятором PICMASTER.  
Подробное описание смотрите в документации на эмулятор PICMASTER.

## 7.7.9 Debug &gt; Clear All Points – удаление всех точек остановки и трассировки.

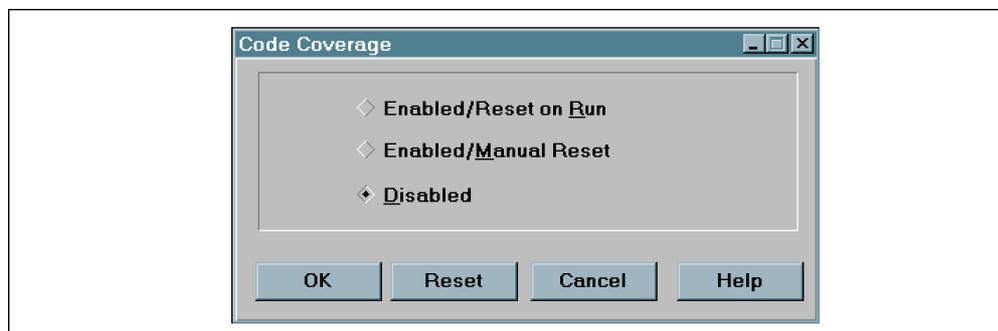


## 7.7.10 Debug &gt; Complex Trigger Settings – настройка сложного триггера.



**Примечание.** Работает только с эмулятором MPLAB ICE.  
 Подробное описание смотрите в документации на эмулятор MPLAB ICE.

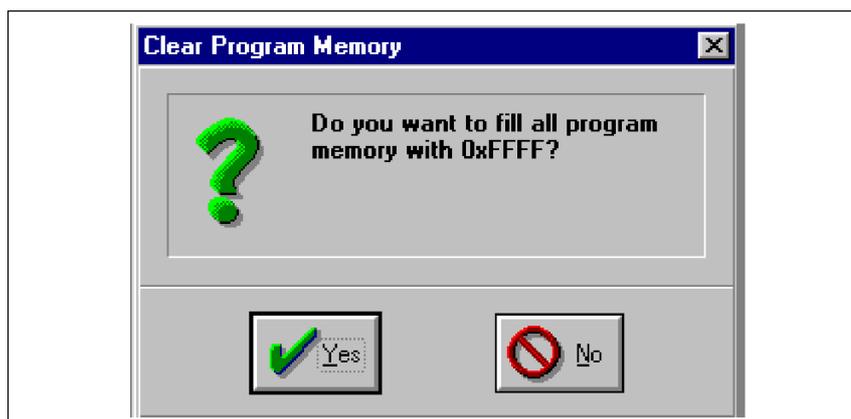
## 7.7.11 Debug &gt; Code Coverage – захват кода.



**Примечание.** Работает только с эмулятором MPLAB ICE.  
 Подробное описание смотрите в документации на эмулятор MPLAB ICE.

## 7.7.12 Debug &gt; Clear Program Memory – очистка памяти программ

Устанавливает все биты памяти программ в единицу.



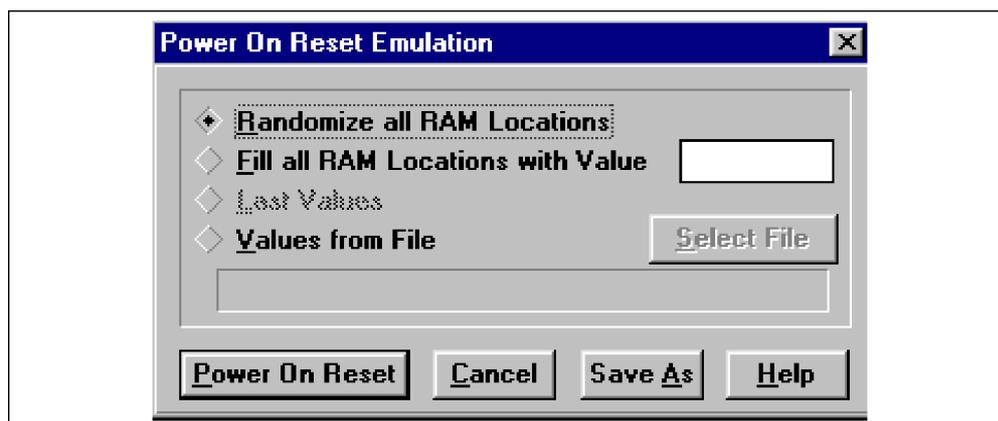
### 7.7.13 Debug > System Reset (Ctrl + Shift + F3) – системный сброс.

Системный сброс эмулятора, для установки новых параметров, включая аппаратные средства. Аналогично инициализации при запуске MPLAB IDE. Если в это время выполнялась программа, то происходит сброс микроконтроллера по сигналу -MCLR.

**Примечания.**

1. Для сброса микроконтроллера (-MCLR) используйте команду *Debug > Run > Reset*.
2. Всегда выключайте питание эмулятора при замене эмуляционного модуля, а затем выполняйте системный сброс. Если Вы не выполните системный сброс, MPLAB IDE не будет должным образом настроен для нового эмуляционного модуля.

### 7.7.14 Debug > Power-On-Reset (Ctrl + Shift + F5) – сброс по включению питания.



Сброс по включению питания позволяет записывать в ОЗУ случайные или предварительно подготовленные данные. Использование предварительно подготовленных данных поможет обнаружить нестабильные ошибки, которые возникают при включении питания.

**Примечание.** В эмуляторах при выполнении сброса по включению питания POR выполняется соединением вывода MCLR с VDD. POR сбрасывает не все регистры специального назначения.

Сброс по включению питания имеет следующие функции:

- в регистры, которые имеют неизвестное значение после POR, записывается случайные числа;
- заполнение регистров определенным значением или их очистка;
- запись в регистры значений, описанных в технической документации на микроконтроллер (только для MPLAB SIM). Регистры, значение которых определено при сбросе микроконтроллера, не будут изменяться функцией POR;
- сохранить текущее значение в POR файл;
- загружать значения POR из файла.

**Примечание.** Память программ и точки остановки деактивированы во время работы в диалоговом окне POR.

<i>Randomize</i>	В регистры, которые имеют неизвестное значение после POR, записывать случайное число.
<i>Fill with Value</i>	Введите значение, которым будут заполняться регистры при сбросе POR.
<i>Last Values</i>	Установите флаг <i>Last Values</i> , если хотите использовать предыдущие, случайные значения регистров в последующих сбросах POR.
<i>Power-On-Reset</i>	Выполнить сброс по включению питания, с изменением значений регистров.
<i>Cancel</i>	Закреть диалоговое окно без выполнения сброса POR.
<i>Save As</i>	Открыть диалоговое окно сохранения файла *.POR.
<i>Values from File</i>	Загрузить данные из файла. Установите флаг <i>Values from File</i> и нажмите кнопку <i>Select</i> . Откроется диалоговое окно загрузки файла, укажите требуемый файл *.POR.

## **7.8 Меню программаторов (Picstart Plus/PRO MATE)**

Откройте диалоговое окно выбора программатора командой *Options > Programmer Options > Select Programmer*. Как только Вы выберете другой программатор, среда проектирования MPLAB IDE будет закрыта. Изменения вступят в силу только поле перезагрузки MPLAB IDE. В области текстового меню появится название выбранного программатора.

Описание работы с программаторами PICSTART и PRO MATE в среде проектирования MPLAB IDE смотрите в технической документации на соответствующий программатор.

## 7.9 Меню настроек (Options)

### 7.9.1 Options > Development Mode – настройка параметров работы микроконтроллера

На экране появится диалоговое окно со следующими закладками:

- инструменты (Tools);
- порт компьютера (Ports);
- тактовый сигнал (Clock);
- память (Memory);
- конфигурация (Configuration);
- питание (Power);
- выводы микроконтроллера (Pins);
- точки остановки (Break options).

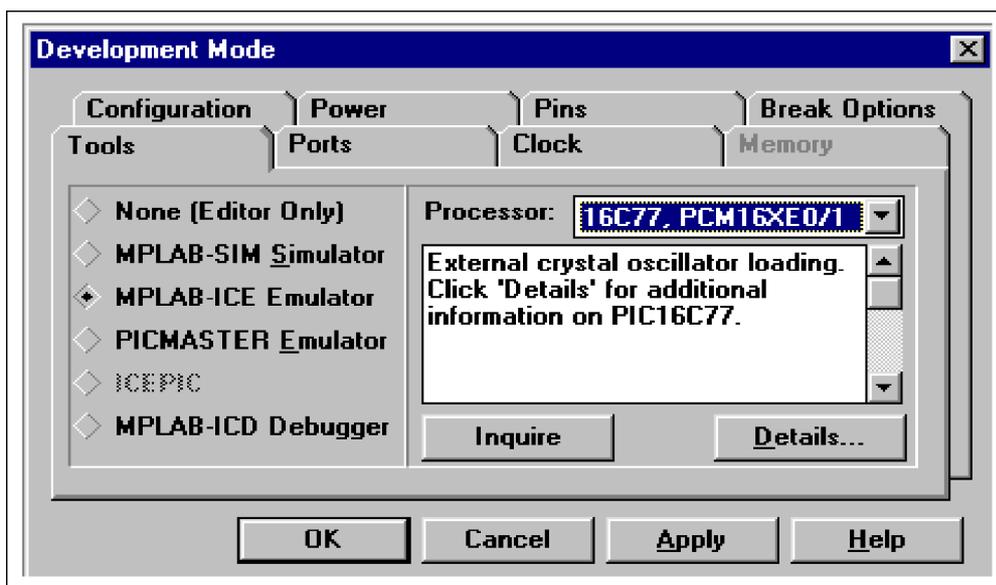
#### 7.9.1.1 Инструменты (Tools)

Выберите пункт меню *Options > Development Mode* раздел *Tools*, для настройки среды проектирования и выбора типа микроконтроллера.

Двойным щелчком «мышки» нажмите кнопку *Details* для получения дополнительной информации по ограничениям работы с конкретным типом микроконтроллера в выбранной среде.

Чтобы проверить, для какого микроконтроллера в настоящее время настроен эмулятор (MPLAB-ICE или PICSTART), нажмите кнопку *Inquire*.

Если Вы сделали изменения в настройках среды проектирования и хотите их сохранить, не закрывая диалогового окна, нажмите кнопку *Apply*.



<i>None (Editor Only)</i>	Только редактор. Симулятор и эмулятор выключен. Вы можете только редактировать и компилировать текст программы. На линейке состояния отображается знак EO
<i>Processor</i>	Выберите микроконтроллер, используемый в разработке.
<i>MPLAB-SIM Simulator</i>	Режим симулятора MPLAB-SIM.
<i>Processor</i>	Выберите симулируемый микроконтроллер.
<i>MPLAB-ICE Emulator</i>	Режим эмулятора MPLAB-ICE . На линейке состояния отображается знак ICE (если эмулятор подключен).
<i>Processor</i>	Выберите эмулируемый микроконтроллер.
<i>PICMASTER Emulator</i>	Режим эмулятора PICMASTER. На линейке состояния отображается знак Em (если эмулятор подключен).
<i>Processor</i>	Выберите эмулируемый микроконтроллер.
<i>ICEPIC</i>	Режим эмулятора ICEPIC (если эмулятор подключен).
<i>Processor</i>	Выберите эмулируемый микроконтроллер.
<i>MPLAB-ICD Debugger</i>	Режим отладчика MPLAB -ICD. На линейке состояния отображается знак ICD (если отладчик подключен).
<i>Processor</i>	Выберите микроконтроллер.
<i>Cancel</i>	Отменить изменения и закрыть диалоговое окно.
<i>OK</i>	Сохранить изменения и закрыть диалоговое окно.

### 7.9.1.2 Порт компьютера (Ports)

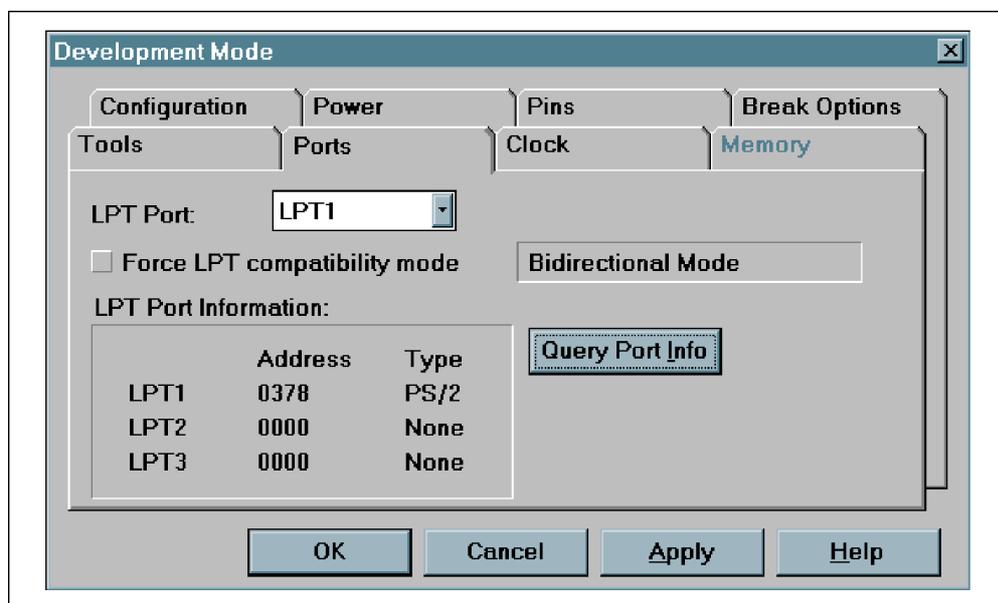
Выберите пункт меню *Options > Development Mode* раздел *Ports*.

Настройка параметров соединения эмуляторов с компьютером.

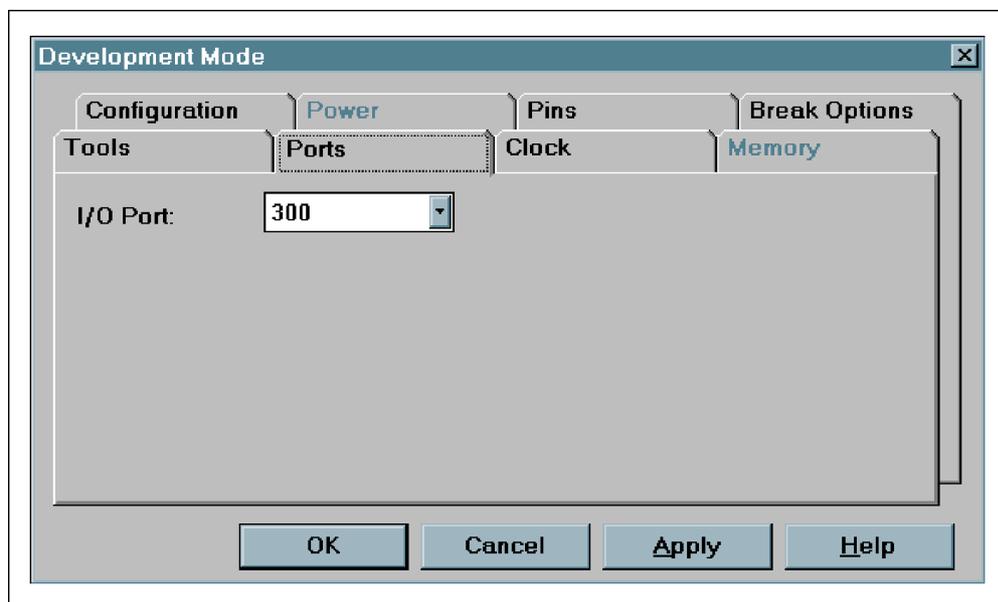
Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.

Подробное описание настроек смотрите в технической документации на соответствующий тип оборудования.

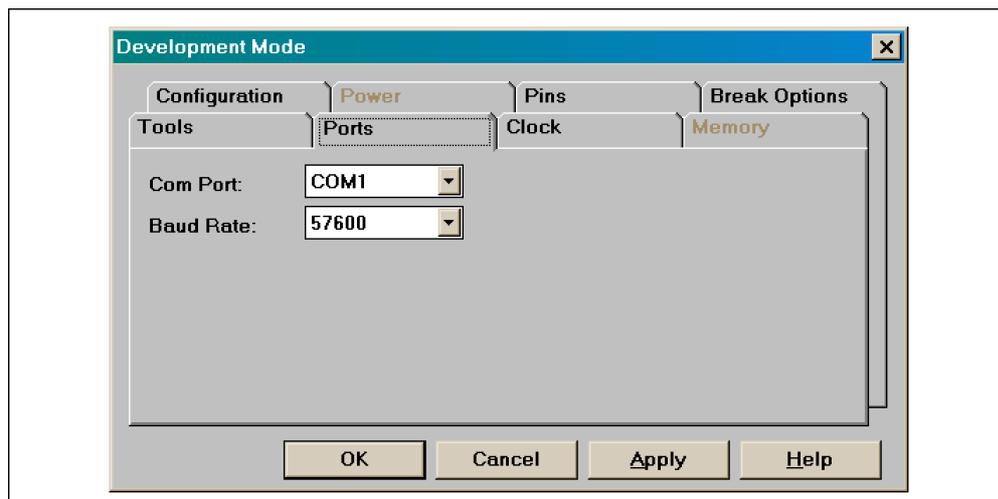
Эмулятор MPLAB-ICE



Эмулятор PICMASTER



## Эмулятор ICEPIC

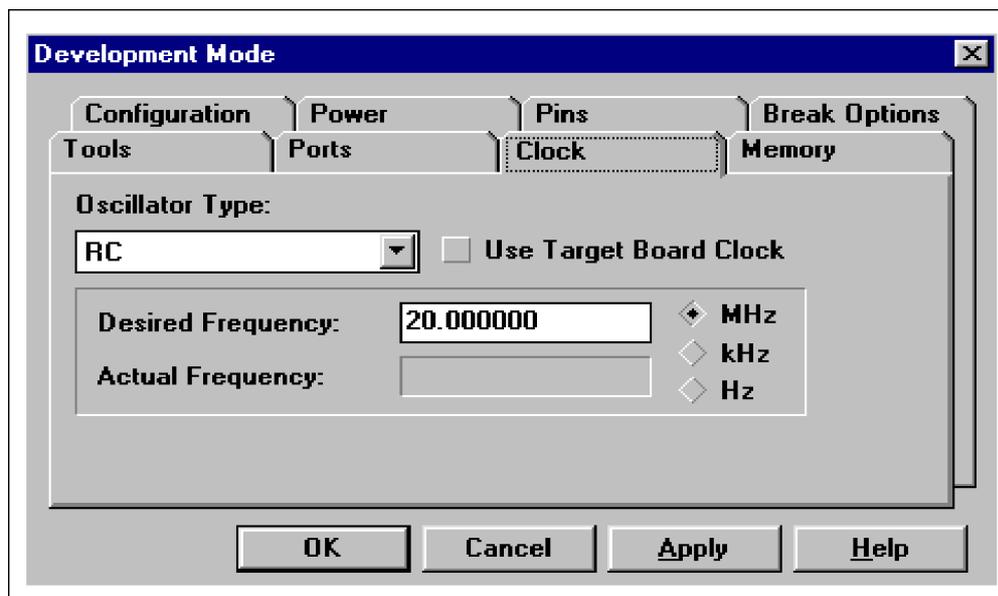


## 7.9.1.3 Тактовый сигнал (Clock)

Выберите пункт меню *Options > Development Mode* раздел *Clock*.

Вы можете указать тактовую частоту микроконтроллера в МГц(MHz), кГц(kHz) или Гц(Hz). Обратитесь к технической документации на микроконтроллер для определения диапазона поддерживаемых частот тактового сигнала и типа тактового генератора.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.



Рекомендуемая последовательность выбора тактового сигнала:

1. Выберите тип генератора тактового сигнала. Доступные варианты зависят от типа выбранного микроконтроллера.
2. Установите флаг *Use Target Board Clock*, чтобы использовался тактовый сигнал устройства, в которое включен эмулятор (доступно только с эмулятором MPLAB ICE).
3. Укажите требуемую частоту тактового сигнала (*desired frequency*) и диапазон (MHz, kHz, Hz).
4. Нажмите *Apply* для сохранения изменений.

**Примечание.** Фактическая частота (*Actual Frequency*) может быть использована только с эмулятором MPLAB ICE.

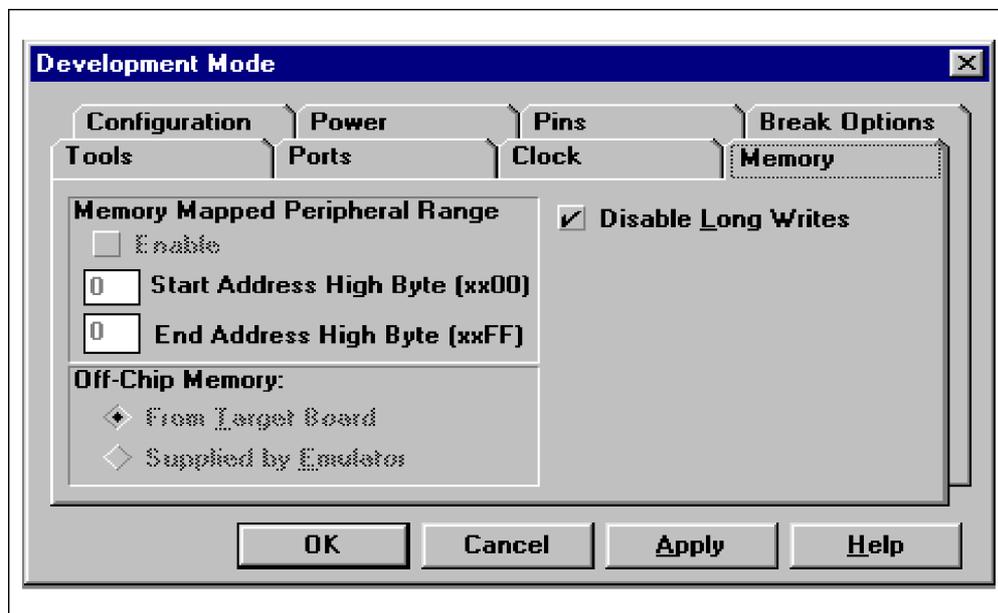
#### 7.9.1.4 Память (Memory)

Выберите пункт меню *Options > Development Mode* раздел *Memory*.

Настройка конфигурации памяти.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.

**Примечание.** Данный раздел доступен не для всех типов микроконтроллеров.



#### *Memory Mapped Peripheral Range*

Включены/выключение карты памяти и указание младшего и старшего байтов диапазона адресов. В режиме расширенного микроконтроллера, если старший байт счетчика команд равен 0xFF, то счетчик команд остановится в адресе 0xFFFFE. Максимальное допустимое значение старшего байта счетчика команд – 0xFE. Установить значение старшего байта счетчика команд меньше 0xFF, можно выключив питание эмуляционного модуля или выполнив системный сброс *Debug>System Reset*.

#### *Off-Chip Memory*

Выбор источника данных при режиме расширенного микроконтроллера или микропроцессора. *From Target Board* – данные подставляются из эмулятора. *Supplied By Emulator* – данные берутся из устройства.

#### *Disable Long Writes*

Включение/отключение длинной табличной записи в микроконтроллерах PIC17C42. Если запись производится во внутреннюю память, Вы должны выключить «длинную» запись.

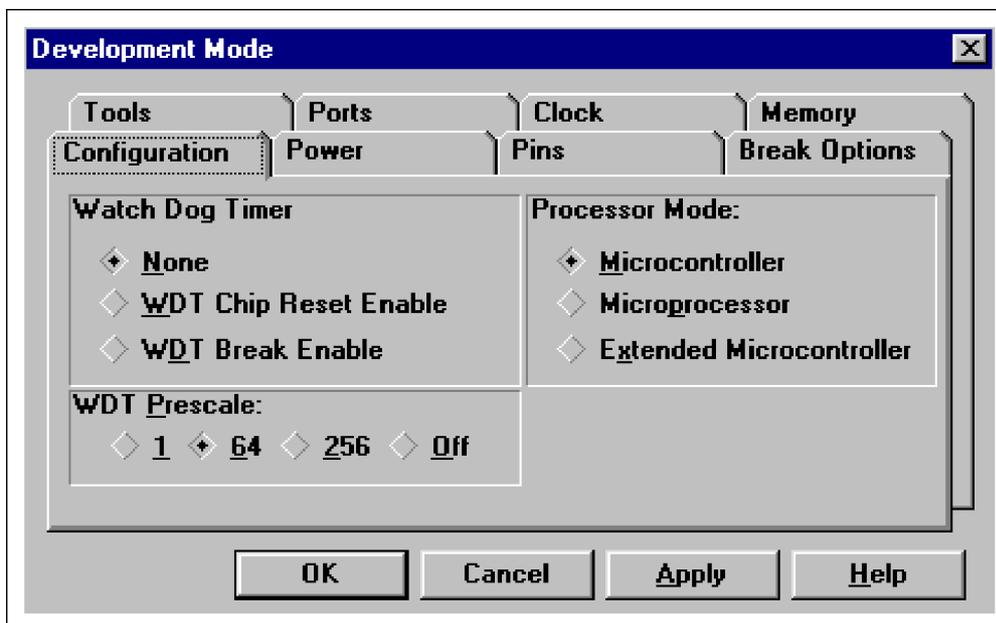
### 7.9.1.5 Конфигурация (Configuration)

Выберите пункт меню *Options > Development Mode* раздел *Configuration*.

Настройка сторожевого таймера WDT и режима работы микроконтроллера.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.

Некоторые параметры данного раздела могут быть недоступными в зависимости от типа микроконтроллера.



#### Watchdog Timer

- None*
- WDT Chip Reset Enable*
- WDT Break Enable*

Управление работой сторожевого таймера WDT.

*None* WDT выключен.

*WDT Chip Reset Enable* WDT включен, при переполнении происходит сброс микроконтроллера.

*WDT Break Enable* WDT включен, при переполнении происходит остановка микроконтроллера.

*Дополнительно.* Проверьте, что Вы правильно настроили параметры работы WDT в регистре OPTION.

#### WDT Prescale

Коэффициент предделителя WDT для микроконтроллеров PIC17CXXX и PIC18CXXX.

Для остальных типов микроконтроллеров (PIC12CXXX, PIC14000 и PIC16CXXX) коэффициент предделителя WDT устанавливается программно в регистре OPTION. Смотрите техническую документацию на соответствующий микроконтроллер.

#### Processor Mode

- Microcontroller*
- Microprocessor*
- Extended Microcontroller*

Выберите режим работы микроконтроллера, если доступно.

Дополнительную информацию смотрите в технической документации на микроконтроллер.

*Microcontroller* Режим микроконтроллера, только внутренняя память программ.

*Microprocessor* Режим микропроцессора, только внешняя память программ.

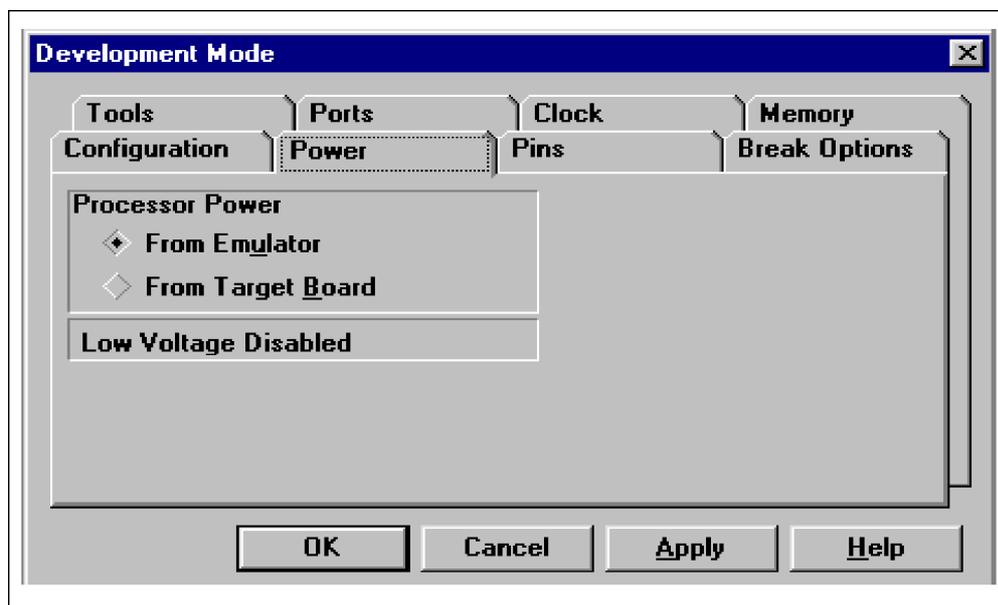
*Extended Microcontroller* Режим расширенного микроконтроллера, внутренняя и внешняя память программ.

#### 7.9.1.6 Питание (Power)

Выберите пункт меню *Options > Development Mode* раздел *Power*.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.

Настройка источника питания, только для режима эмуляции.



#### *Processor Power*

Устройство, в котором эмулируется работа микроконтроллера, получает питание от эмулятора (*From Emulator*) или от другого источника (*From Target Board*).

#### *Low Voltage Enabled/Disabled*

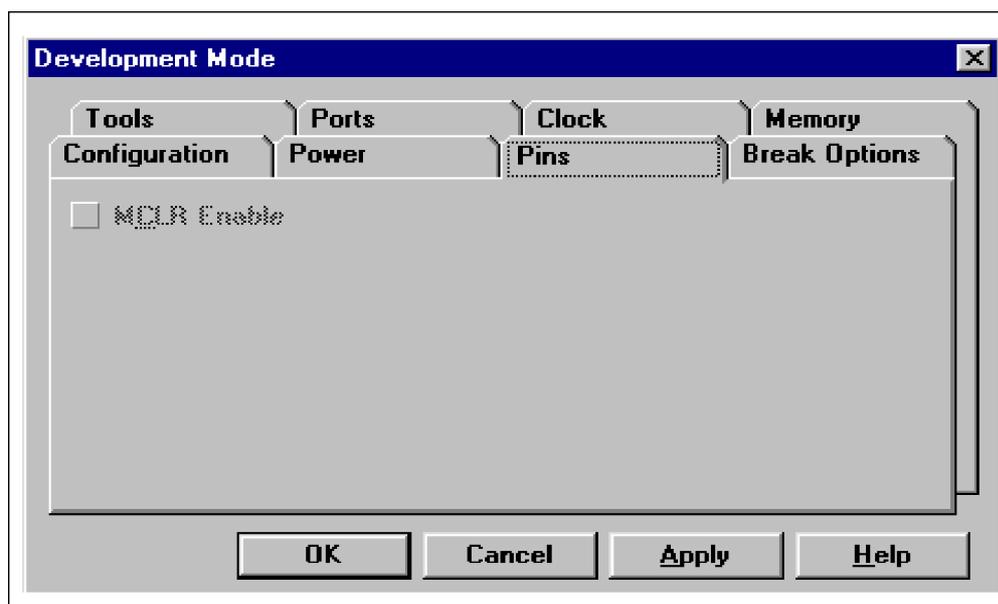
Включение/отключение контроля, за снижением напряжения питания.

#### 7.9.1.7 Выводы микроконтроллера (Pins)

Выберите пункт меню *Options > Development Mode* раздел *Pins*.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.

Разрешение/запрещение обработки сигнала с вывода MCLR. Только для режима эмуляции.

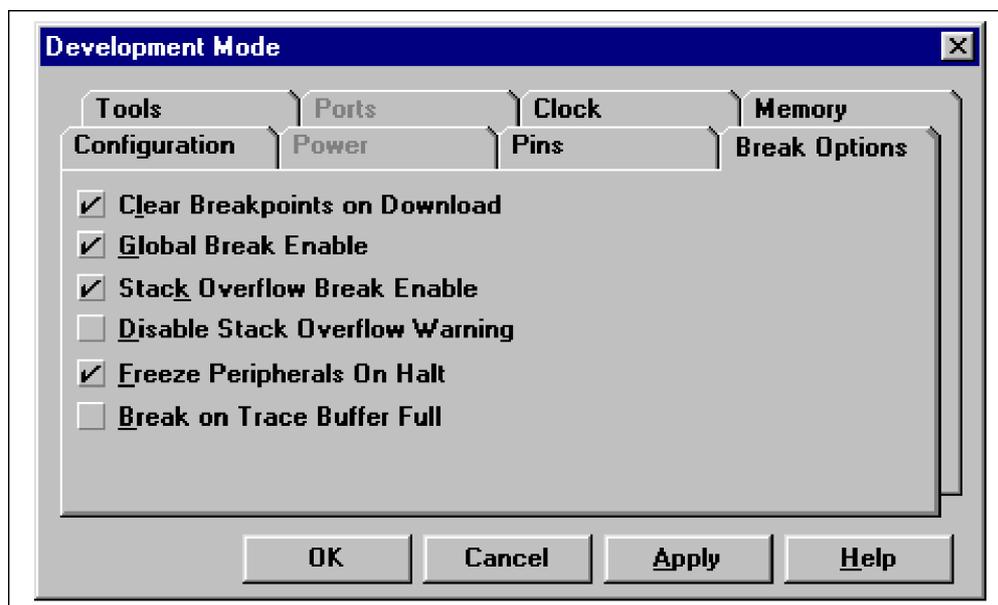


### 7.9.1.8 Точки остановки (Break options)

Выберите пункт меню *Options > Development Mode* раздел *Break options*.

Настройка параметров точек остановки и трассировки.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.



Следующие параметры доступны для всех режимов среды проектирования, кроме Edit Only.

- |  |   |
|--|---|
| <i>Clear Break Points on Download</i>  | Удаление всех точек остановки, точек трассировки, точек триггера и адреса прохода при загрузке в эмулятор.  |
| <i>Global Break Enable</i>   | Глобальное разрешение работы с точками остановки. Если флаг не установлен, все точки остановки отключены. Параметр доступен на линейке состояния.   |
| <i>Stack Overflow Break Enable</i>   | Разрешение остановить выполнение программы при переполнении аппаратного стека. Если флаг установлен, то при переполнении (исчерпании) стека останавливается выполнение программы. При отключении данной функции допускается установить флаг формирования предупреждения о переполнении/исчерпании стека.  |
| <i>Disable Stack Overflow Warning</i>  | Отключение предупреждения о переполнении аппаратного стека микроконтроллера. Когда флаг не установлен, сообщение формироваться не будет.  |
| <i>Freeze Peripherals on Halt</i>  | Остановка таймеров периферийных модулей при остановке микроконтроллера. Когда периферийные модули остановлены, Вы можете записывать значения в порт ввода/вывода, но не можете прочитать, пока не будет выполнен один шаг программы. Эта опция облегчает отладку программного обеспечения, если необходимо, чтобы таймеры продолжали работать при остановке микроконтроллера. |
| <br>   |   |
| <b>Примечание.</b> Если выбрана остановка периферийных модулей, то при выполнении одного шага программы значение регистров SFR портов ввода/вывода и показания секундомера обновляться не будут. |   |
| <i>Break on Trace Buffer Full</i>  | Остановка хода выполнения программы при переполнении буфера трассировки (объем буфера трассировки 8Кб). Пункт недоступен для эмуляторов MPLAB ICE и ICEPIC.   |

## 7.9.2 Подменю Window Setup

### 7.9.2.1 Сохранение настроек

*Options > Window Setup > Save Setup* – сохранить текущую конфигурацию в файле \*.CFG.

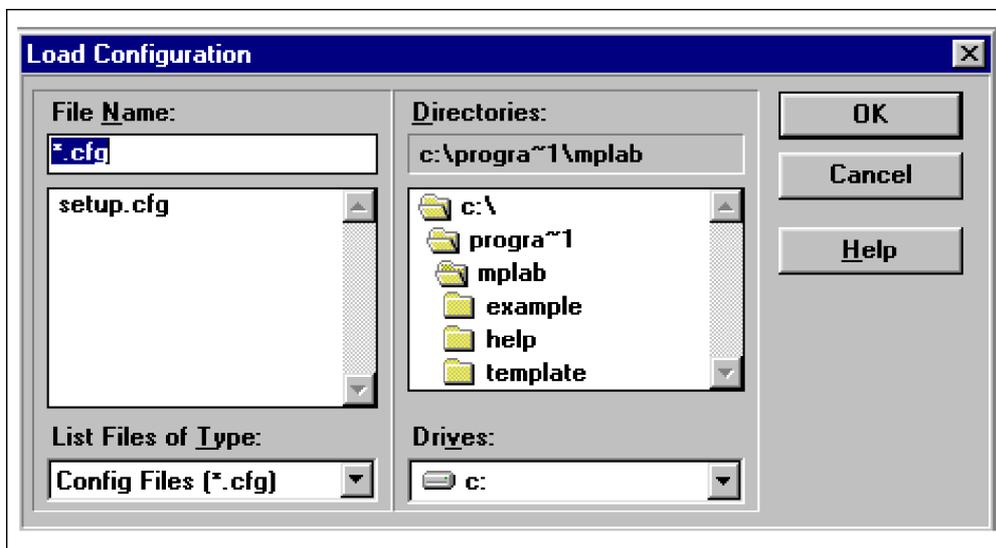


Сохраняются параметры настроек открытых окон, но не точек остановки, трассировки или триггера.

<i>Yes</i>	Если Вы нажмете кнопку <i>YES</i> на вопрос “Do you want to save the current configuration as the default?”, MPLAB IDE сохранит текущую конфигурацию и будет ее использовать при последующих запусках.
<i>No</i>	Откроется стандартное диалоговое окно сохранения файла.
<i>Cancel</i>	Отменить сохранение.

### 7.9.2.2 загрузить сохраненные настройки

*Options > Window Setup > Load Setup* – загрузить сохраненную в файле \*.CFG конфигурацию по команде *Options > Window Setup > Save Setup*. Выберите нужный файл и нажмите кнопку *OK*. Расширение файла конфигурации по умолчанию .CFG.



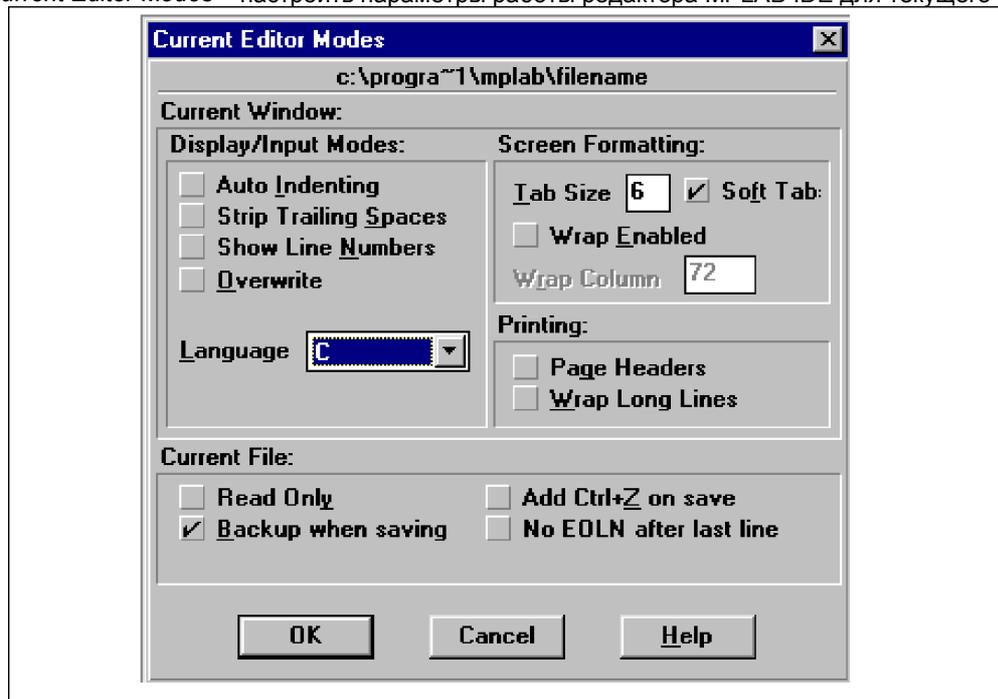
**Примечание.** Файл Setup.cfg используется программатором PRO MATE II при программировании микросхем ограничения доступа. Не записывайте собственную конфигурацию в этот файл.

### 7.9.2.3 Восстановить настройки по умолчанию

*Options > Window Setup > Default Configuration* – загрузить конфигурацию по умолчанию (как при загрузке MPLAB IDE).

### 7.9.3 Настройка текстового редактора MPLAB IDE

*Options > Current Editor Modes* – настроить параметры работы редактора MPLAB IDE для текущего окна.



#### **Display/Input Modes**

Эти параметры управляют отображением текста в окне и действиями редактора MPLAB при вводе текста.

<i>Auto Indenting</i>	При переводе курсора на новую строку установить уровень предыдущей строки.
<i>Strip Trailing Spaces</i>	Удалить пробелы в конце строки при нажатии Enter.
<i>Show Line Numbers</i>	Показывать номер строк в окне.
<i>Overwrite</i>	Режим замены текста при вводе.
<i>Language</i>	Выберете язык программирования из списка, что определит параметры форматирования текста при печати.

#### **Screen Formatting**

Настройка параметров табуляции и ограничения длины строки.

<i>Tab Size</i>	Длина символа табуляции.
<i>Soft Tabs</i>	Вставлять пробелы вместо символа табуляции.
<i>Wrap Enabled</i>	Разрешить ограничивать длину строки.
<i>Wrap Column</i>	Максимальное число символов в строке.

#### **Printing**

Эти настройки являются параметрами по умолчанию для команды File > Print.

<i>Page Headers</i>	Печатать в начале каждой страницы имя файла, страницу, число страниц, дату.
<i>Wrap Long Lines</i>	Ограничивать длину строки.

#### **Current File**

Параметры файла.

<i>Read Only</i>	Запрещение редактирования файла.
<i>Backup When Saving</i>	Создавать резервную копию при сохранении файла.
<i>Add Ctrl+Z On Save</i>	Записывать в конец файла символ Ctrl+Z при сохранении.
<i>No EOLN after last line</i>	Не записывать атрибуты конца файла (CR-LF или LF) при сохранении.

### 7.9.4 Сброс настроек редактора MPLAB IDE.

Выберете *Options > Reset Editor Modes* для восстановления параметров текущего окна по умолчанию.

## 7.9.5 Настройка среды проектирования (Environment Setup)

*Options > Environment Setup* – на экране появится диалоговое окно со следующими разделами:

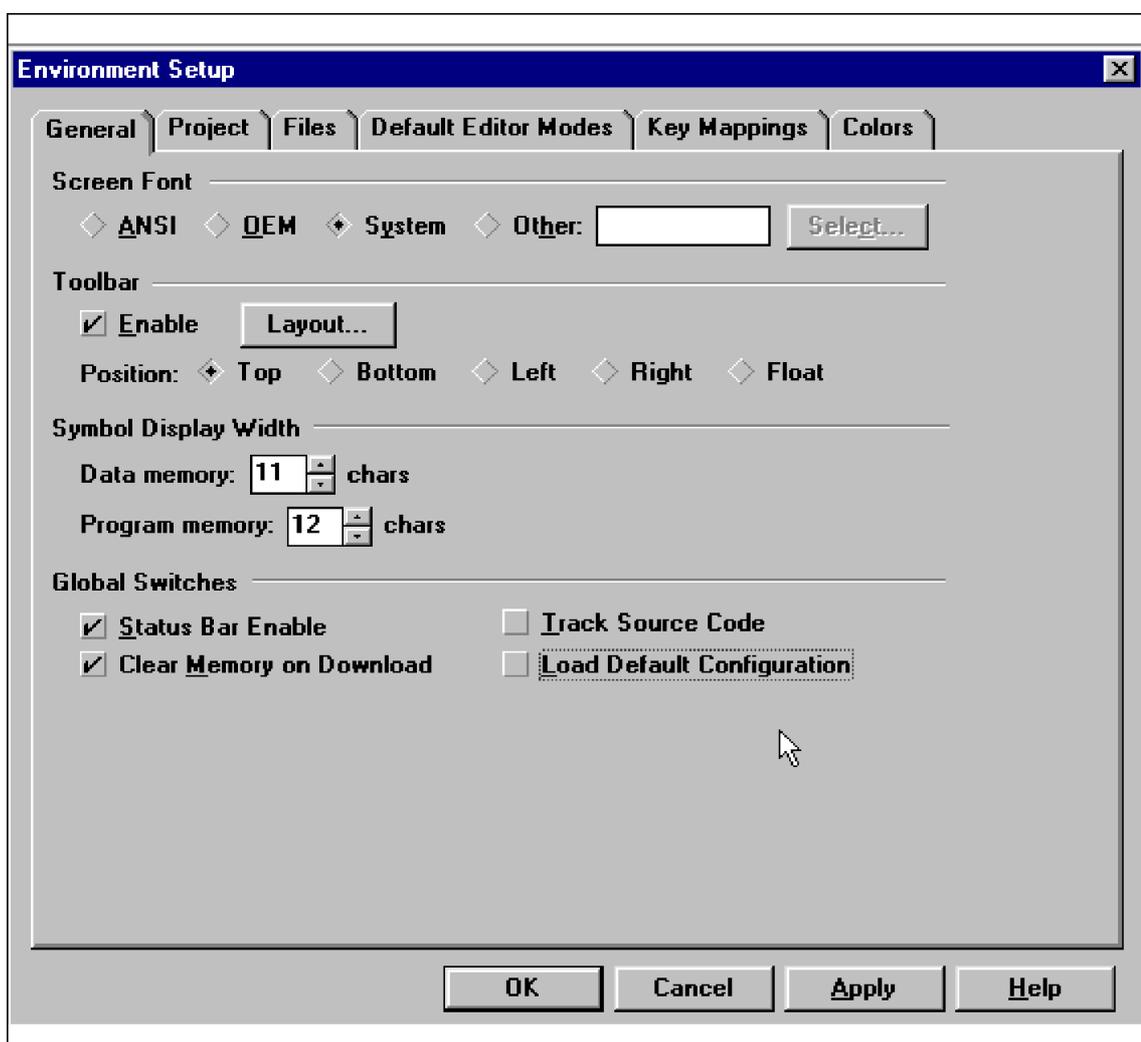
General	- общие параметры;
Project	- параметры проекта;
Files	- файлы;
Default Editor Modes	- параметры редактора по умолчанию;
Key Mappings	- назначение клавиш;
Colors	- настройка цвета.

### 7.9.5.1 Общие параметры (General)

Выберите пункт меню *Options > Environment Setup* раздел *General*.

В диалоговом окне находятся настройки шрифта на экране монитора, настройки графического меню и глобальные параметры.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.



#### 7.9.5.1.1 Настройка шрифта (Screen Font)

Настройка шрифта, который будет использоваться в окнах MPLAB IDE.

1. Выберите тип шрифта – ANSI, OEM, системный (*System*) или другой (*Other*).
2. В случае указания шрифта *Other*, нажмите *Select* и выберите конкретный шрифт.
3. Нажмите *OK* для сохранения изменений и выхода из диалогового окна.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.

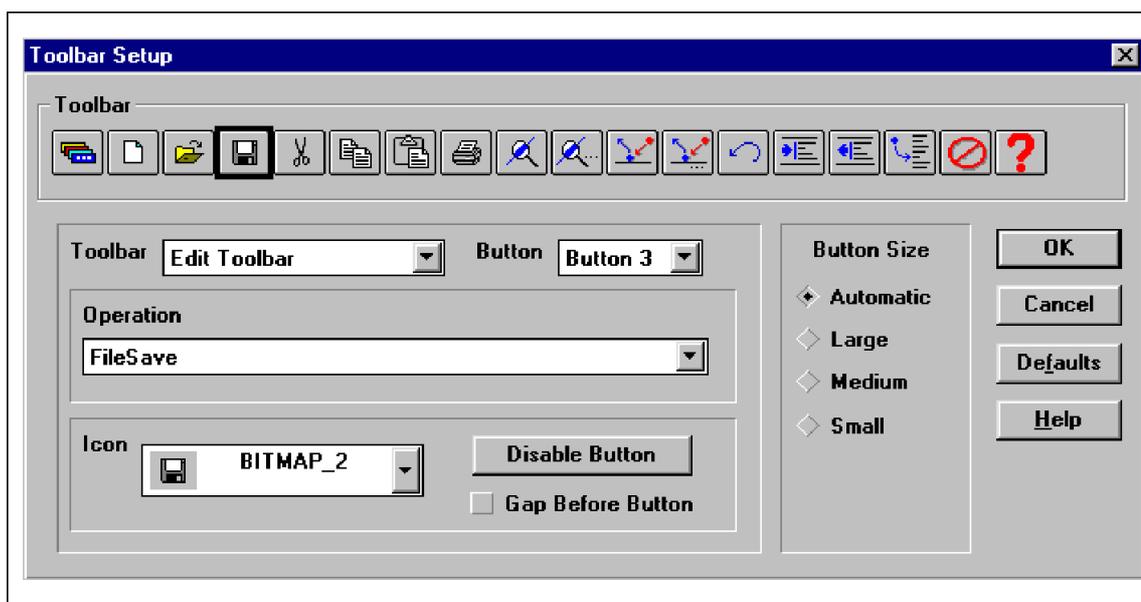
## 7.9.5.1.2 Настройка графического меню (Toolbar)

*Enable* – включить графическое меню на рабочем столе MPLAB IDE;

Разместить графическое меню на рабочем столе MPLAB IDE:

- Top* - сверху;
- Bottom* - снизу;
- Left* - слева;
- Right* – справа;
- Float* – свободное размещение.

Для дополнительных настроек графического меню нажмите кнопку *Layout*.  
На экране появится диалоговое окно, показанное на рисунке.



MPLAB IDE позволяет выполнять следующие настройки кнопок графического меню:

- добавить кнопку в графическое меню или редактировать существующую;
- удалить кнопку из графического меню;
- группировать кнопки.

<i>Toolbar</i>	Выбор группы кнопок из списка <i>Edit</i> – редактора MPLAB IDE; <i>Debug</i> – отладчика; <i>Project</i> – проекта; <i>User</i> – пользователя.
<i>Button</i>	Расположение кнопки от 1 до 16.
<i>Operation</i>	Функциональное назначение кнопки, выбирается из списка.
<i>Icon</i>	Графическое изображение кнопки, выбирается из списка.
<i>Disable Button</i>	Запрещение отображения кнопки в выбранной позиции.
<i>Gap Before Button</i>	Промежуток перед кнопкой.
<i>Button Size</i>	Размер кнопки: <i>Automatic</i> – автоматически; <i>Large</i> – крупная; <i>Medium</i> – средняя; <i>Small</i> – маленькая.
<i>Ok</i>	Сохранить изменения.
<i>Cancel</i>	Отмена изменений.
<i>Defaults</i>	Восстановление настроек по умолчанию.

**Добавление и редактирование кнопки графического меню**

Если Вы желаете добавить кнопку графического меню или изменить функциональное назначение существующей кнопки, необходимо выполнить следующие действия:

1. Выберите пункт меню *Options > Environment Setup* раздел *General* и нажмите кнопку *Layout*.
2. Из списка *Toolbar* выберите требуемую группу кнопок графического меню.
3. Из списка *Button* выберите позицию кнопки в графическом меню.

**Примечание.** Если на выбранной Вами позиции кнопки изображен красный круг, значит в этой позиции в графическом меню никакой кнопки не назначено. Если кнопка уже определена, то последующие действия вызовут изменение настроек выбранной кнопки графического меню.

4. Из списка *Icon* выберите графическое изображение кнопки в графическом меню.
5. Из списка *Option* выберите функциональное назначение кнопки.
6. Нажмите кнопку *OK* для сохранения настроек и возвращения в главное диалоговое окно.

**Удаление кнопок из графического меню**

1. Выберите пункт меню *Options > Environment Setup* раздел *General* и нажмите кнопку *Layout*.
  2. Из списка *Toolbar* выберите требуемую группу кнопок графического меню.
  3. Из списка *Button* выберите позицию кнопки в графическом меню.
  4. Нажмите *Disable Button* для удаления кнопки. Удаленная кнопка будет отображаться в виде красного круга.
  5. Нажмите кнопку *OK* для сохранения настроек и возвращения в главное диалоговое окно.
- После закрытия диалогового окна на месте удаленной кнопки будет пусто.

**Объединение кнопок графического меню в логические группы**

Допускается объединение кнопок графического меню в логические группы, разделяя их небольшим отступом.

1. Выберите пункт меню *Options > Environment Setup* раздел *General* и нажмите кнопку *Layout*.
2. Из списка *Toolbar* выберите требуемую группу кнопок графического меню.
3. Из списка *Button* выберите позицию первой кнопки группы в графическом меню.
4. Установите флаг *Gap Before*.
5. Нажмите кнопку *OK* для сохранения настроек и возвращения в главное диалоговое окно.

**7.9.5.1.3 Длина строки с символами (Symbol Display Width)**

Определяет количества символов метки при отображении в символьном формате.

<i>Data Memory</i>	Длина символьной метки от 6 до 32 в памяти данных.
<i>Program Memory</i>	Длина символьной метки от 6 до 32 в памяти программы.

**7.9.5.1.4 Глобальные параметры (Global Switches)**

<i>Status Bar Enable</i>	Показывать линейку состояния.
<i>Clear Memory on Download</i>	Устанавливать все биты памяти в единицу перед загрузкой в эмулятор MPLAB IDE.
<i>Load Default Configuration</i>	Если флаг установлен, то при запуске MPLAB IDE будет загружаться конфигурация по умолчанию. Заменить настройки по умолчанию можно командой <i>Options &gt; Window Setup &gt; Save Setup</i> , в открывшемся сообщении нажать кнопку <i>Yes</i> .
<i>Track Source Code</i>	Если флаг установлен, MPLAB IDE изменяет текущую строку в исходном тексте программы при выполнении одного шага программы. Вы может использовать эту функцию, если есть файл *.HEX, но нет *.COD.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.

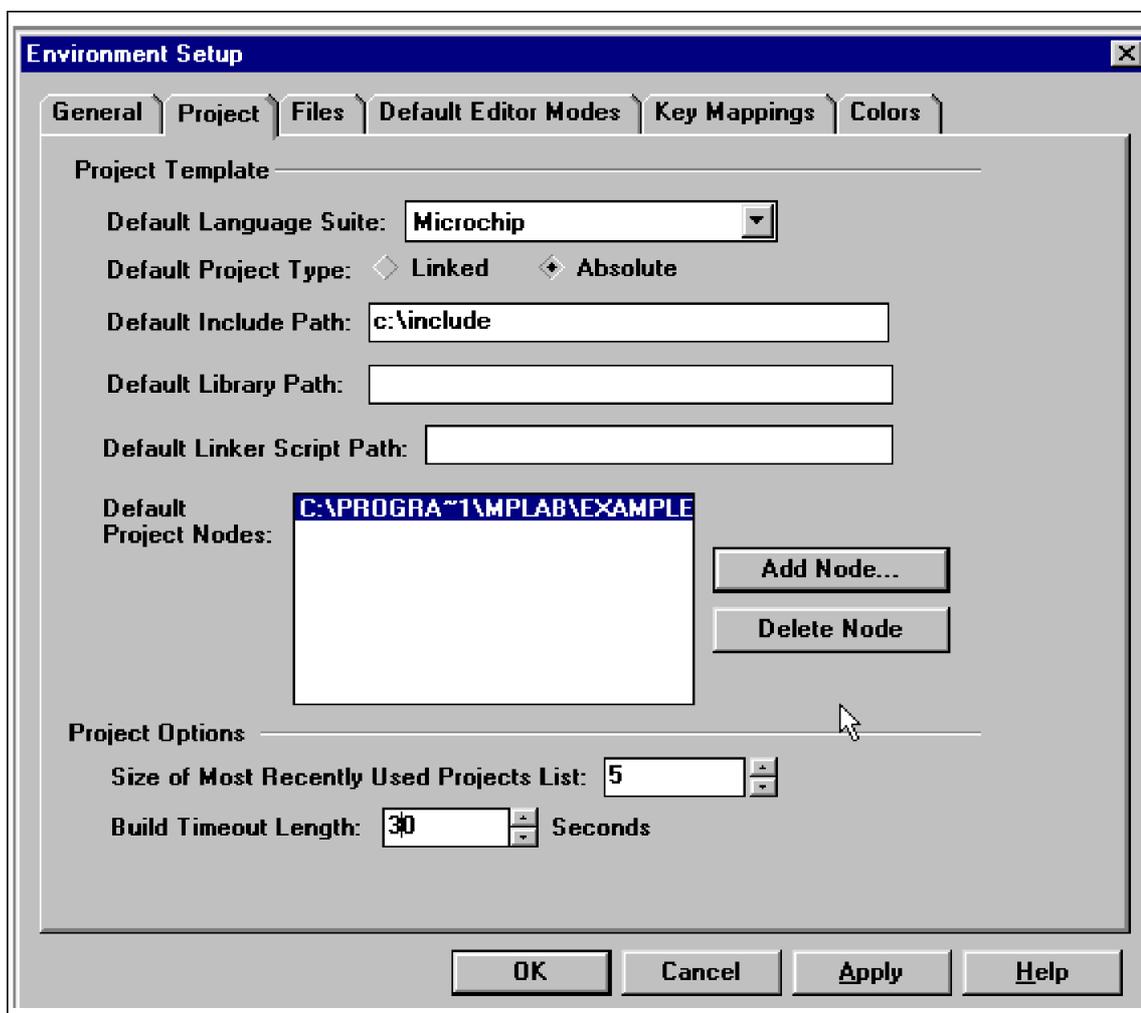
### 7.9.5.2 Параметры проекта (Project)

Выберите пункт меню *Options > Environment Setup* раздел *Project*.

Настройки, указанные в данном пункте меню будут использоваться при создании новых проектов. На существующие проекты не влияет.

**Примечание.** Для изменения параметров существующего проекта выберите пункт меню *Project > Edit Project*.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.



#### Default Language Suite

Из списка Вы можете выбрать набор программ. Затем Вы должны установить инструментальные средства для этого набора. Если выбран набор, который не установлен на вашем персональном компьютере, проекты исполняться не будут. Если Вы установили программное обеспечение до установки MPLAB IDE, Вам необходимо внести информацию относительно нового инструментального средства в диалоговом окне *Install Language Tool*.

#### Default Project Type

Если Вы выбрали набор языка Microchip (по умолчанию), то необходимо указать несколько исходных файлов (*linked*) или один исходный файл (*absolute*) используется в проекте.

#### Default Include Path

В строке впишите путь размещения подключаемых файлов в проект. Если путь для индивидуального проекта не указан, то при исполнении проектов подключаемые файлы найти не удастся. Допускается указание пути к файлам в AUTOEXEC.BAT вписав переменную MCC\_INCLUDE, однако этот способ не эффективен.

#### Default Library Path

В строке впишите путь по умолчанию к библиотечным файлам. Строка доступна для ввода, если Вы указали, что в проекте используется несколько исходных файлов.

---

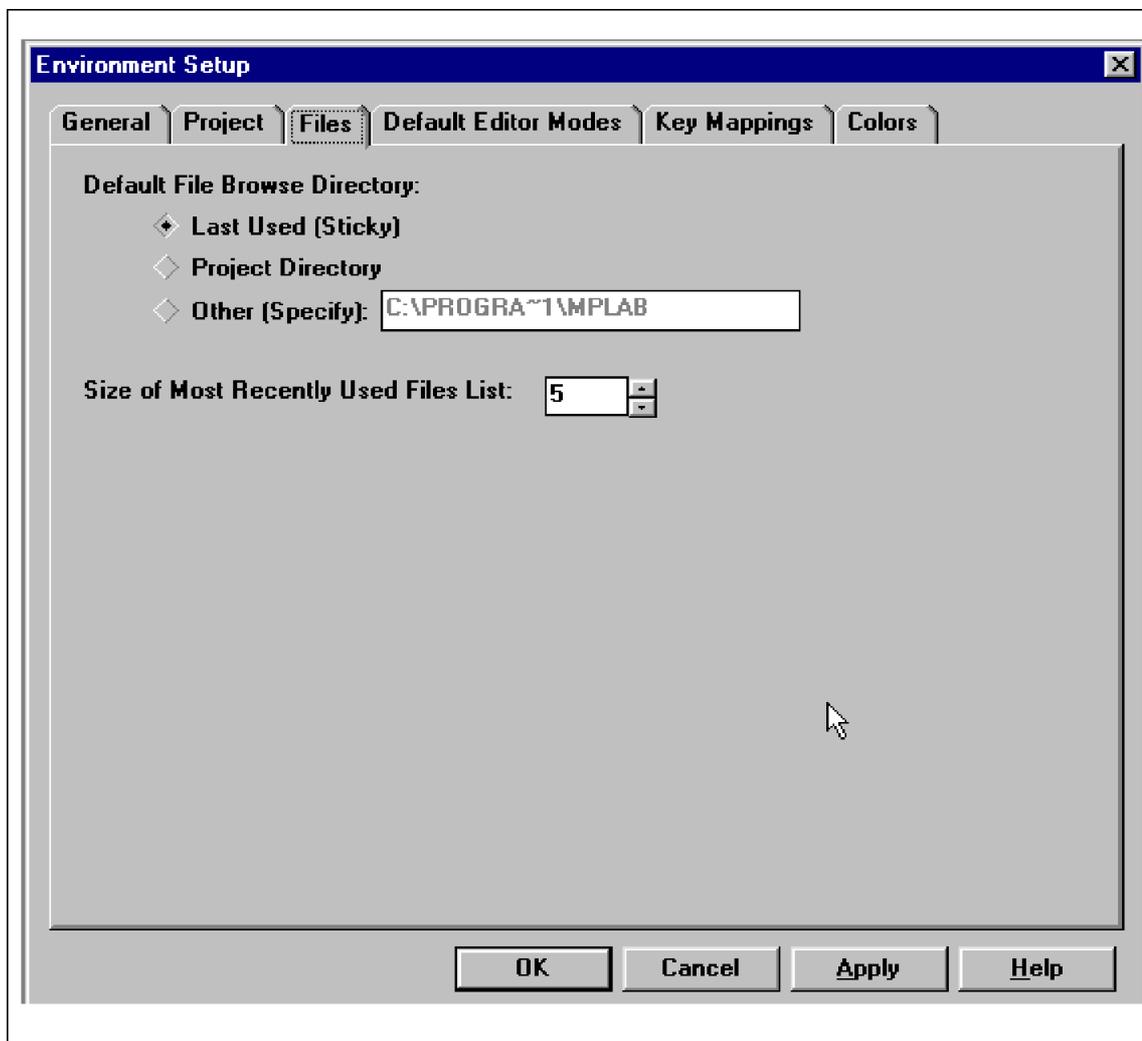
<i>Default Linker Script Path</i>	<p>В строке впишите путь по умолчанию к файлам сценария линкера. Строка доступна для ввода, если Вы указали, что в проекте используется несколько исходных файлов. Например:</p> <pre>c:\mplab\projects\mproj\include; c:\include\h; ..\sys</pre>
<i>Default Project Nodes</i>	<p>Подключение не компилируемых узлов (объект, библиотека и файл сценария), которые Вы хотите включить в Ваш проект. Нажмите кнопку <i>Add Node</i>, чтобы указать диск, директорию и имя подключаемого файла. Чтобы отключить узел, выберите его из списка и нажмите кнопку <i>Delete Node</i>.</p>
<i>Size of Most Recently Used Projects List</i>	<p>Максимальное количество имен последних используемых проектов в меню Project MPLAB. Когда Вы уменьшаете этот параметр, имена файлов будут удалены из списка. Удаленные названия файлов не будут восстановлены при увеличении числа файлов в списке. Имена файлов, которые Вы использовали после увеличения параметра, будут появляться в списке. Длина списка по умолчанию – пять.</p>
<i>Build Timeout Length</i>	<p>Максимальное время компиляции в секундах, после истечения указанного времени будет выдано соответствующее предупреждение. Если установлено значение "off", время ожидания не вычисляется.</p>

### 7.9.5.3 Файлы (Files)

Выберите пункт меню *Options > Environment Setup* раздел *Files*.

Настройка директории хранения списка открываемых файлов и максимальное количество файлов в списке.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.



#### 7.9.5.3.1 Default File Browse Directory

Указать директорию по умолчанию, с которой MPLAB IDE начинает при открытии файлов.

##### *Last Used*

Начинать с директории последнего открываемого файла. Если директория больше не существует, открывается рабочая директория MPLAB IDE.

##### *Project Directory*

Начинать с директории текущего проекта (по умолчанию). Если директория не существует, открывается директория предыдущего уровня. Если ни один проект не открыт, начинается обзор файлов с рабочей директории MPLAB IDE.

##### *Other*

Начинать с директории указанной в строке.

#### 7.9.5.3.2 Size of Most Recently Used Files List

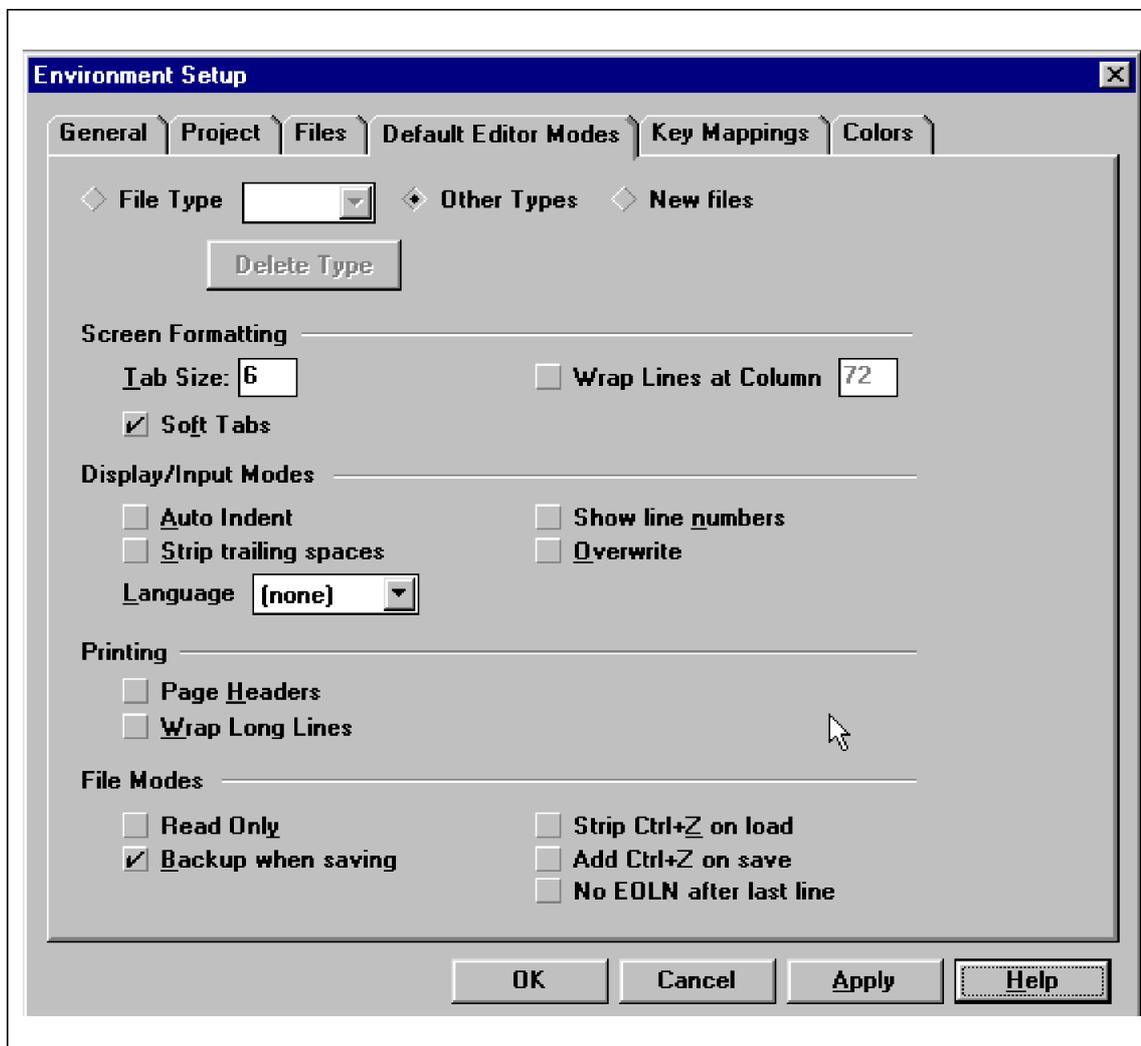
Максимальное количество имен последних используемых файлов в меню *File* MPLAB. Когда Вы уменьшаете это параметр, имена файлов будут удалены из списка. Удаленные названия файлов не будут восстановлены при увеличении числа файлов в списке. Имена файлов, которые Вы использовали после увеличения параметра, будут появляться в списке. Длина списка по умолчанию – пять.

#### 7.9.5.4 Параметры редактора по умолчанию (Default Editor Modes)

Выберите пункт меню *Options > Environment Setup* раздел *Default Editor Modes*.

Настройка параметров работы редактора MPLAB IDE по умолчанию, используется при создании новых и для существующих файлов.

Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*.



Чтобы установить параметры редактора по умолчанию для конкретного типа файлов установите флаг File Type, выберите тип файла из списка или впишите новое расширение файла. Установленные параметры применяются, когда Вы используете команды *File > Open Source*, *File > View* и *File > Save As*.

Последовательность действий при настройке параметров редактора MPLAB по умолчанию:

1. Установите флаг *New File*, для настройки параметров редактора при создании файлов командой *File > New*. Или, установите флаг *Other Types* для настройки параметров редактора других типов файлов.
2. Проверьте параметры отображения, формата, ввода, печати и сохранения файла.
3. Нажмите кнопку *Apply* для сохранения настроек.
4. Повторите шаги 1-3 для настройки других типов файлов.

Список параметров редактора MPLAB:

#### **Display/Input Modes**

Эти параметры управляют отображением текста в окне и действиями редактора MPLAB при вводе текста.

<i>Auto Indenting</i>	При переводе курсора на новую строку установить уровень предыдущей строки.
<i>Strip Trailing Spaces</i>	Удалить пробелы в конце строки при нажатии Enter.
<i>Show Line Numbers</i>	Показывать номер строк в окне.
<i>Overwrite</i>	Режим замены текста при вводе.
<i>Language</i>	Выберете язык программирования из списка, что определит параметры форматирования текста при печати.

**Screen Formatting**

Настройка параметров табуляции и ограничения длины строки.

<i>Tab Size</i>	Длина символа табуляции.
<i>Soft Tabs</i>	Вставлять пробелы вместо символа табуляции.
<i>Wrap Enabled</i>	Разрешить ограничивать длину строки.
<i>Wrap Column</i>	Максимальное число символов в строке.

**Printing**

Эти настройки являются параметрами по умолчанию для команды File > Print.

<i>Page Headers</i>	Печатать в начале каждой страницы имя файла, страницу, число страниц, дату.
<i>Wrap Long Lines</i>	Ограничивать длину строки.

**Current File**

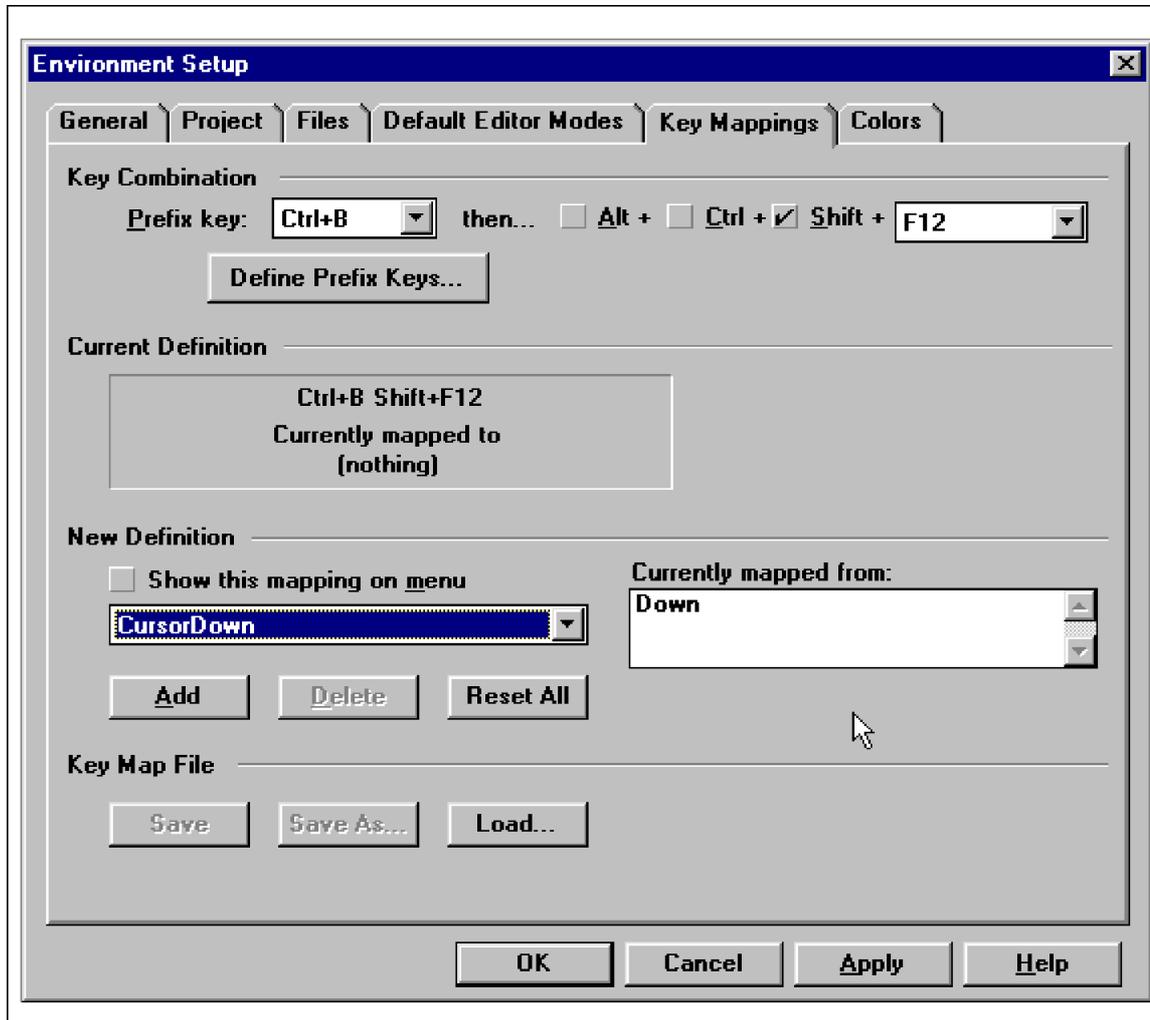
Параметры файла.

<i>Read Only</i>	Запрещение редактирования файла.
<i>Backup When Saving</i>	Создавать резервную копию при сохранении файла.
<i>Add Ctrl+Z On Save</i>	Записывать в конец файла символ Ctrl+Z при сохранении.
<i>No EOLN after last line</i>	Не записывать атрибуты конца файла (CR-LF или LF) при сохранении.

### 7.9.5.5 Назначение клавиш (Key Mappings)

Выберите пункт меню *Options > Environment Setup* раздел *Key Mappings*.

Для каждой функции среды проектирования MPLAB IDE можно назначить соответствующую клавишу или комбинацию клавиш на клавиатуре компьютера. MPLAB IDE использует двоичный файл MPLAB.KEY (в каталоге MPLAB IDE) для сохранения назначений комбинаций клавиш.



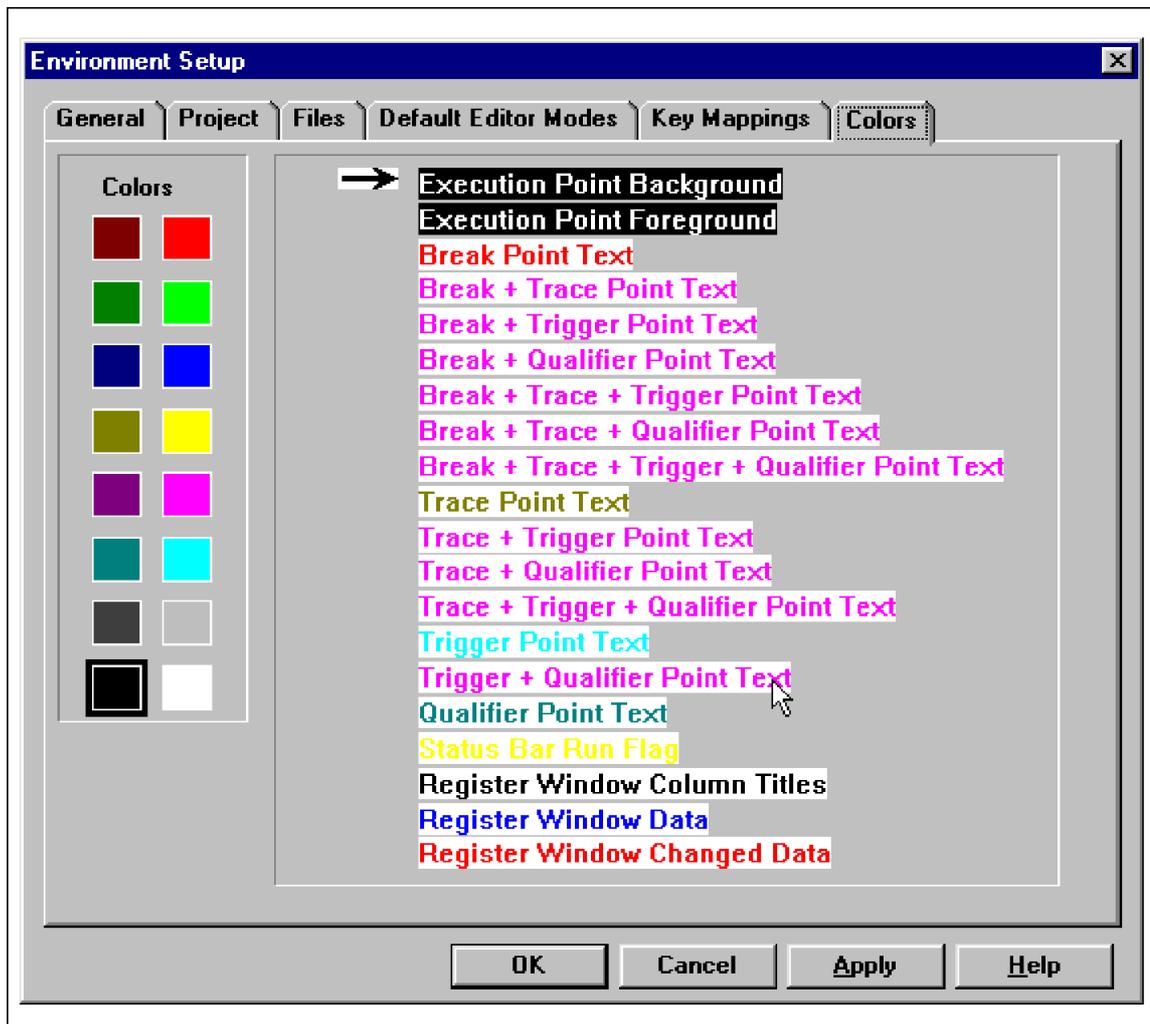
Для сохранения изменений, без закрытия диалогового окна, нажмите кнопку *Apply*. Никакие префиксы комбинаций клавиш не допускаются (по умолчанию).

Порядок назначения клавиш:

1. Откройте сохраненный файл настроек клавиатуры (*Load*).
2. Укажите необязательную приставку (*prefix key*).
3. Выберите дополнительную и основную клавишу (например, Shift + F12).
4. Установите флажок *Show this mapping on menu* если хотите, чтобы комбинация клавиш отображалась в меню возле команды.
5. Выберите функцию из списка в разделе *New Definition list*.
6. Нажмите *Add* для добавления комбинации.
7. Чтобы убрать комбинацию клавиш с функции, выберите соответствующую комбинацию и нажмите кнопку *Delete*.
8. Сохранить изменения в файле *Save* или *Save As*.

### 7.9.5.6 Настройка цвета (Colors)

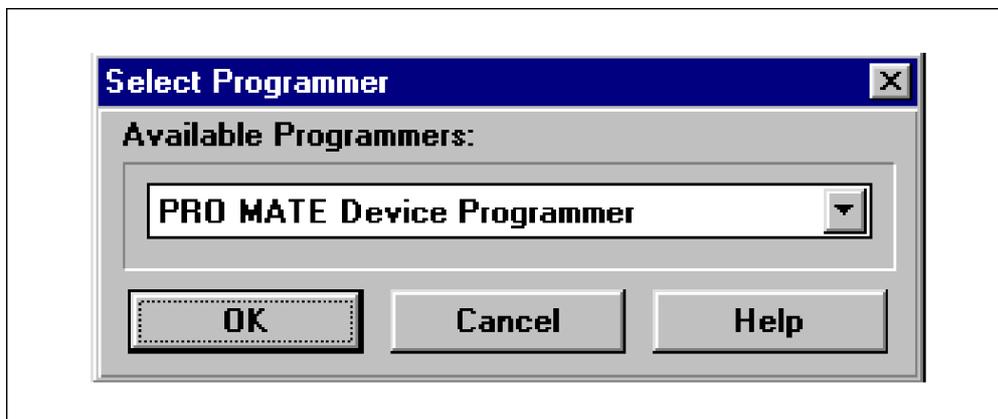
Выберите пункт меню *Options > Environment Setup* раздел *Colors*.  
Настройка цвета отображаемых данных.



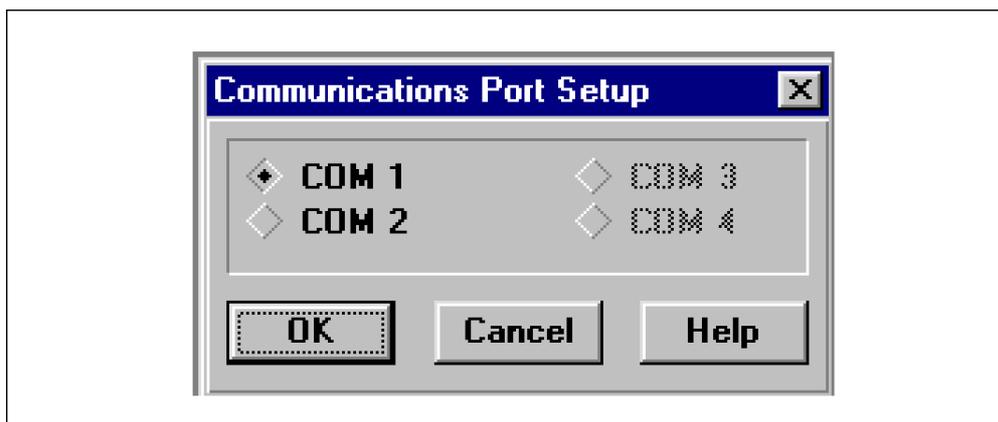
Для изменения цвета выберите текст, а затем цвет. Сохранить изменения *Apply*.

## 7.9.6 Настройка программаторов (Options > Programmer Options)

7.9.6.1 *Options > Programmer Options > Select Programmer* – выбор типа программатора. Выберите тип программатора из списка и нажмите кнопку *OK*.



7.9.6.2 *Options > Programmer Options > Communications Port Setup* – настройка порта соединения с компьютером.



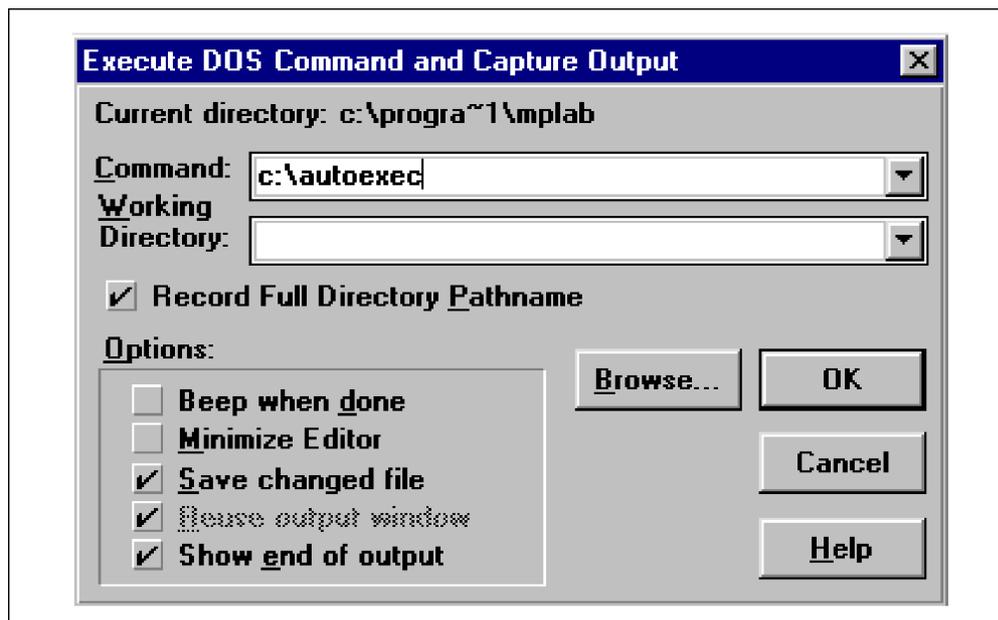
## 7.10 Меню Tools

### 7.10.1 Выполнение DOS программы

Выберите пункт меню *Tools > DOS Command to Window* (F11), на экране появится диалоговое окно, показанное на рисунке.

В диалоговом окне будут сохранены настройки предыдущего выполнения программы DOS. В строке *Command* впишите путь и имя программы DOS, которую Вы хотите выполнить. Нажав кнопку *Browse*, Вы можете определить рабочий каталог, в котором должна выполняться программа DOS. Одновременно не может быть запущено несколько программ DOS. При попытке запуска еще одной программы DOS, MPLAB IDE игнорирует запуск.

**Примечание.** Нельзя запустить Windows приложения, используя данный пункт меню.



1. Установите параметры (*options*) работы DOS программы:

<i>Beep when done</i>	Звуковой сигнал о завершении выполнения DOS программы.
<i>Minimize Editor</i>	Минимизировать окно редактора на время выполнения DOS программы.
<i>Save changed files</i>	Сохранить измененные файлы в редакторе MPLAB перед запуском DOS программы.
<i>Reuse output window</i>	Передавать результаты работы в окно редактора (если флаг не установлен, новое окно не будет создано).
<i>Show end of output</i>	Автоматически показывать конец данных полученных из DOS программы.

2. Нажмите кнопку *OK* для запуска DOS программы.

Командная строка, рабочий каталог и параметры запуска DOS программы сохраняются для следующего открытия диалогового окна.

### 7.10.2 Повторное выполнение DOS программы

*Tools > Repeat DOS Command to Window* (Ctrl + F11) – повторное выполнение DOS программы (без открытия диалогового окна).

Если еще ни одна программа DOS не была выполнена, то эта команда откроет диалоговое окно пункта меню *Tools > DOS Command to Window*.

### 7.10.3 Проверка эмулятора PICMASTER

*Tools > Verify PICMASTER* – проверить эмулятор PICMASTER.

Дополнительную информацию смотрите в технической документации на эмулятор.

### 7.10.4 Проверка эмулятора MPLAB-ICE

*Tools > Verify MPLAB-ICE* – проверить эмулятор MPLAB-ICE.

Дополнительную информацию смотрите в технической документации на эмулятор.

## 7.11 Меню Window

Все пункты меню доступны в режимах симулятора и эмулятора.  
В режиме Edit Only окна Absolute Listing и Show Symbol List не доступны.

С помощью данного меню можно открыть следующие окна:

Program Memory	– память программ;
Trace Memory	– буфер трассировки;
EEPROM Memory	– EEPROM память;
Calibration Data	– данные калибровки;
Absolute Listing	– листинг программы;
Map File	– карта памяти;
Stack	– стек;
File Registers	– память данных;
Special Function Registers	– специальные регистры;
Show Symbol List	– список переменных и меток;
Stopwatch	– секундомер;
Project	– проект;
Watch Window	– окно с переменными;
Modify	– изменения.

Управление расположением окон на рабочем столе:

Tile Horizontal	– горизонтально;
Tile Vertical	– вертикально;
Cascade	– каскадом;
Iconize All	– свернуть все окна;
Arrange Icons	– упорядочить названия свернутых окон;

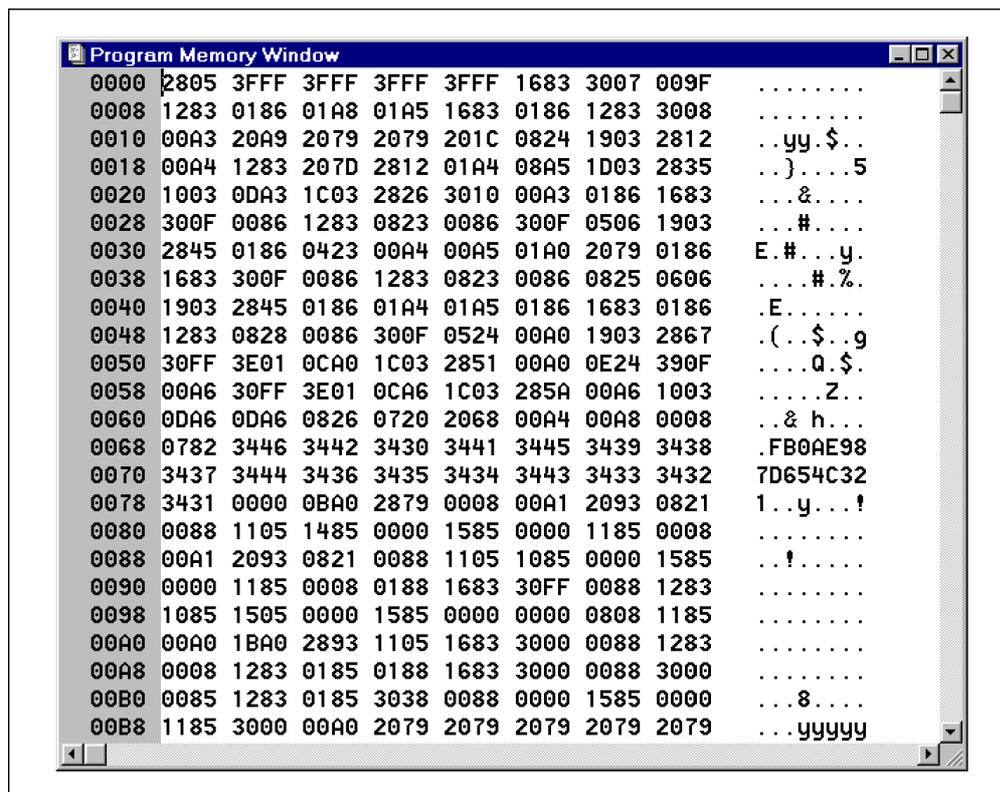
### 7.11.1 Window > Program Memory – открыть окно памяти программ.

Открытие окна памяти программ возможно только в режиме эмуляции или симуляции микроконтроллера, а адресное пространство, которое отображается в окне, соответствует типу выбранного микроконтроллера.

Окно памяти программ может быть постоянно открыто, его можно перемещать по экрану и изменять его размер.

С помощью системного меню памяти программ можно выбрать режим вывода информации.

*Hex Code Display* – показывать память программ в кодах.



*Disassembly Display* – отображать память программ в виде дисассемблированного кода.

*Machine Code Display* – отображать память программ в машинных кодах.

При режиме отображения памяти программ *Disassembly Display* или *Machine Code Display* указатель следующей выполняемой инструкции расположен в середине окна (в случае остановки выполнения программы).

```

Program Memory Window
82      0051  0120      movwf  0x20
83      0052  8001      movlw  0x1
84      0053  0121      movwf  0x21
85      0054  8007      movlw  0x7
86      0055  0122      movwf  0x22
87      0056  E05A  Loop      call   Reduce
88      0057  1722      decfsz 0x22
89      0058  C056      goto   Loop
90      0059  C050      goto   Start
91      B... 005A  1D21  Reduce  swapf  0x21
92      005B  1C21      swapf  0x21,W
93      .T...005C  1D21      swapf  0x21
94      005D  0520      subwf  0x20
95      005E  E060      call   Double
96      005F  8600      retlw  0x0
97      0060  1D21  Double  swapf  0x21
98      0061  1C21      swapf  0x21,W
99      0062  1D21      swapf  0x21
100     0063  0F21      addwf  0x21
101     0064  8600      retlw  0x0
102     0065  FFFF      call   0x1FFF
103     0066  FFFF      call   0x1FFF
104     0067  FFFF      call   0x1FFF
105     0068  FFFF      call   0x1FFF
106     0069  FFFF      call   0x1FFF
  
```

*Toggle Line Numbers* – отображать в окне атрибуты точки остановки (трассировки) и номер строки.

Окно памяти программ содержит следующую информацию:

- Первая колонка – адрес в памяти программ;
- Вторая колонка – код команды;
- Третья колонка – метки переходов в программе;
- Четвертая колонка – мнемоника команды.

Количество символов метки в окне можно настроить в меню *Options > Environment Setup* раздел *General*.

Для обозначения условных точек в памяти программ применяется следующая кодировка.

Символ	Тип точки	Выбор меню (RBM = Правая кнопка «мышки»)
B	Точка остановки (красный)	RBM > Break Point(s)
T	Точка трассировки (зеленый)	RBM > Trace Point(s)
O	Точка триггера	RBM > Trigger Point(s)
Q	Назначен счетчик проходов	Настройка в диалоговом окне именованных точек остановки/трассировки.

Доступность пунктов меню зависит от режима работы среды проектирования.

MPLAB IDE применяет цветовую кодировку установленных точек в окне памяти программ. Если никаких точек не было назначено, текст отображается в обычном формате. Если на адрес назначена какая-либо точка, изменится цвет текст и ширина строки, чтобы указать тип назначенной точки.

Чтобы назначить временную точку остановки, дважды щелкните левой кнопкой «мышки» на требуемом адресе. Программа выполняется в режиме реального времени, пока не будет удовлетворено одно из условий:

- счетчик команд PC достиг выбранного адреса;
- найдена другая точка остановки;
- остановка пользователем.

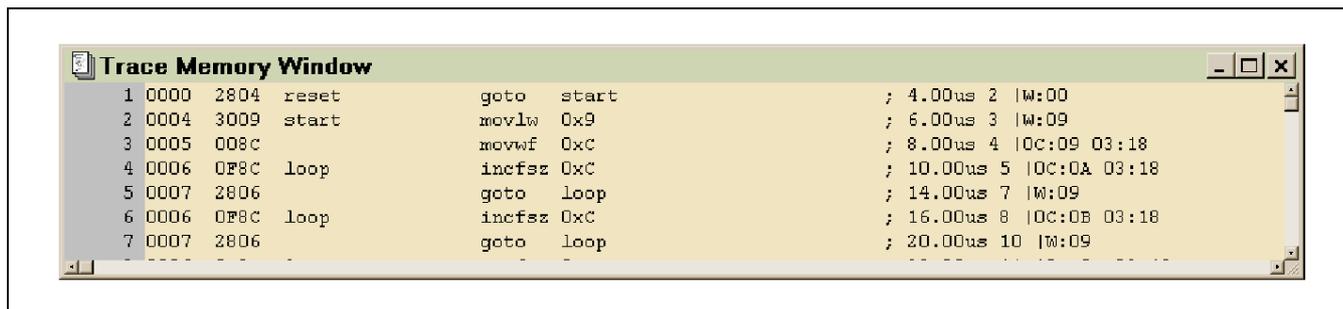
### 7.11.2 Window > Trace Memory – открыть окно буфера трассировки.

В окне отображается состояние микроконтроллера при проходе точки трассировки.

Для эмуляторов, которые поддерживают точки трассировки, сохраняются параметры микроконтроллера при работе в режиме реального времени. Некоторые приложения не могут выполнять приостановку записи в буфер трассировки. В технической документации на эмулятор проверьте его характеристики.

В режиме симулятора буфер трассировки полезен для анализа хода выполнения программы. Информация, сохраняемая в буфере трассировки симулятором, отличается от информации эмулятора. Для быстрой установки точки трассировки:

- нажмите правой кнопкой «мыши» в окне памяти программ;
- на экране появится сокращенное меню;
- выберите пункт *Trace Point(s)*.



Окно буфера трассировки содержит следующую информацию:

Первая колонка – адрес в памяти программ;

Вторая колонка – код команды;

Третья колонка – метки переходов в программе;

Четвертая колонка – мнемоника команды;

Пятая колонка:

*Эмулятор.* Состояние портов ввода/вывода.

*Симулятор.* Время от сброса и измененный регистр.

Количество символов метки в окне можно настроить в меню *Options > Environment Setup* раздел *General*.  
Сохраняется буфер трассировки командой *File > Export > Export Trace Bufer*.

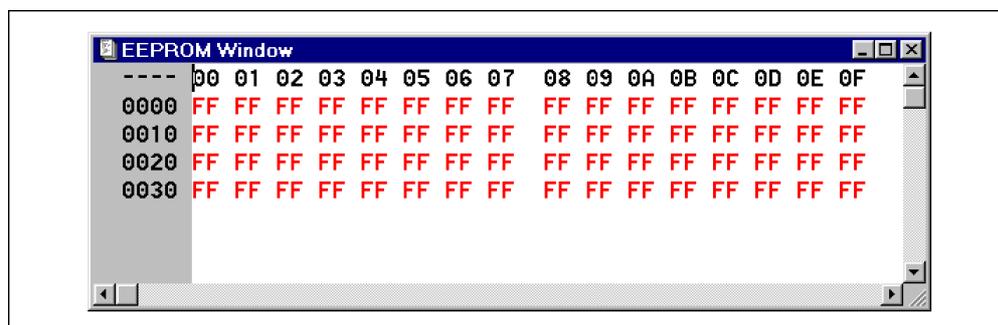
### 7.11.3 Window > EEPROM Memory – открыть окно EEPROM памяти.

Для микроконтроллеров с EEPROM памятью (например, PIC16F84).

Окно может быть постоянно открыто, его можно перемещать по экрану и изменять его размер.

Для изменения значений данного окна выберите пункт меню *Window > Modify*.

Информация в окне обновляется после остановки выполнения программы.



В системном меню окна можно указать:

*Toggle Line Numbers* – показывать номер линии;

*Hex Display* – отображать память EEPROM в HEX формате;

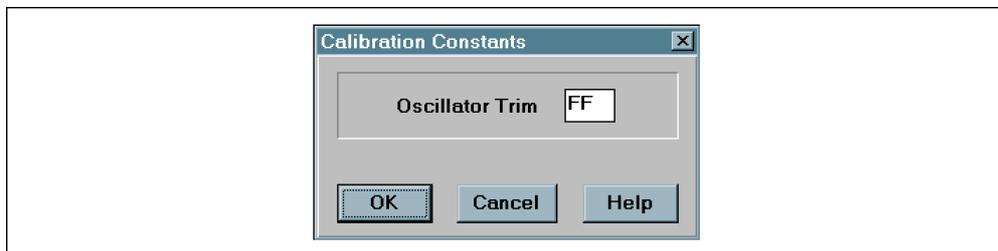
*ASCII Display* – отображать память EEPROM в ASCII формате.

#### 7.11.4 Window > Calibration Data. – открыть окно данных калибровки.

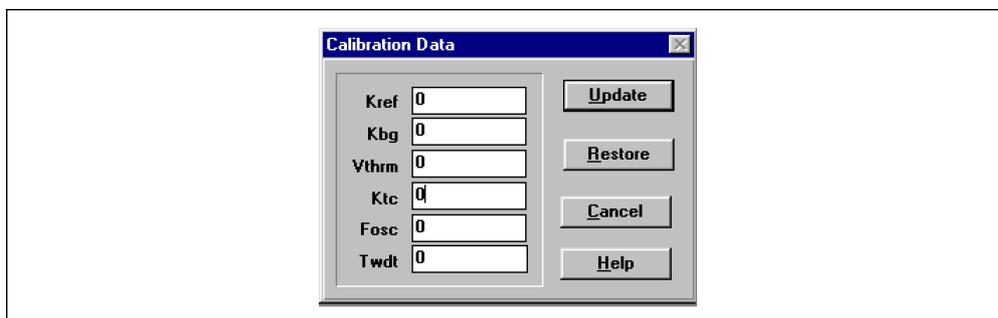
Применяется для микроконтроллеров PIC12CXXX и PIC14000.

Для загрузки сохраненной калибровочной информации используйте пункт меню *File>Import*.

Окно данных калибровки PIC12CXXX.



Окно данных калибровки PIC14000.



*Update*

Значения, указанные в этом диалоговом окне, преобразуются из формата IEEE 754 в Microchip формат IEEE754 и сохраняются в памяти программ. Первые четыре параметра (KREF, KBG, VTHRM и KTC) сохраняются в формате с «плавающей точкой». Последние два значения (FOSC и TWDT) являются незнаковыми 8-разрядными числами от 0 до 255.

*Restore*

Восстановить первоначальную калибровочную информацию.

*Cancel*

Закреть диалоговое окно без изменения памяти.

### 7.11.5 Window > Absolute Listing – открыть окно листинга программы \*.LST файл.

Просмотр абсолютного листа дает возможность оценить качество компиляции программы на языке С или просмотреть размещение кода в объектном файле. Если Вы используете только MPASM, то будет сформирован подробный файл листинга программы.

```

c:\progra~1\mplab\tutor.lst
00070
00071 ;-----
00072 ; Main Program
00073 ;-----
00074
0050 00075          ORG    H'50'
00076
0050 00077 Start
0050 30FF 00078          MOULW 255          ; Initialize the variables to
0051 00A0 00079          MOVWF CountDown      ; their starting values.
0052 3001 00080          MOULW 1
0053 00A1 00081          MOVWF Doubler
0054 3007 00082          MOULW 7
0055 00A2 00083          MOVWF OuterLoop
0056 00084 Loop
0056 205A 00085          CALL  Reduce          ; Perform the inner portion of
0057 00A2 00086          DECFSZ OuterLoop,f ; the loop.
0058 2856 00087          GOTO   Loop
0059 2850 00088
00089          GOTO   Start          ; Repeat the whole thing.
00090
00091 ;-----
005A 00092 Reduce
005A 0EA1 00093          SWAPF  Doubler,f      ; Reduce CountDown by the
005B 0E21 00094          SWAPF  Doubler,w      ; value of Doubler. Then
005C 0EA1 00095          SWAPF  Doubler,f      ; call the doubling routine.
005D 02A0 00096          SUBWF  CountDown,f
005E 2060 00097          CALL  Double
005F 3400 00098          RETLW 0
00099

```

### 7.11.6 Window > Map File. – открыть окно карты памяти.

Позволяет оценить использование памяти микроконтроллера.

```

c:\mchp\projects\17c756\leds\led756.map
MPLINK v2.00.01, Linker
Linker Map File - Created Mon Jan 3 10:54:02 2000

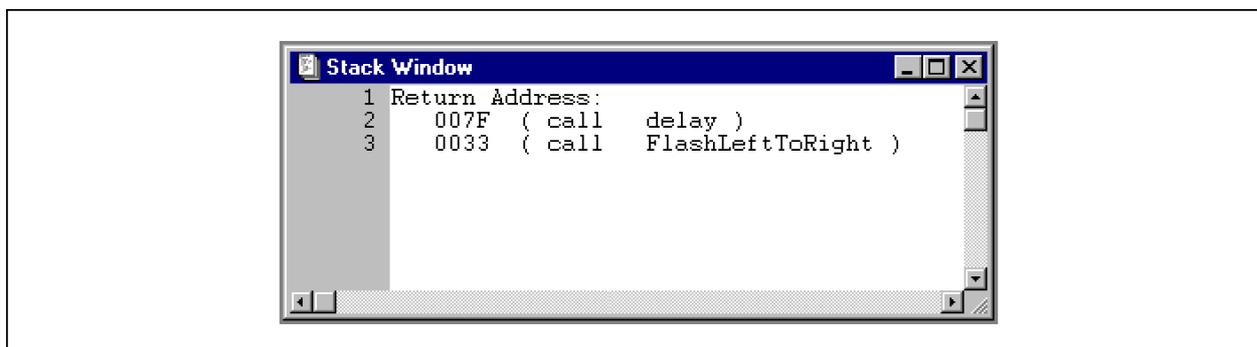
```

Section	Type	Address	Location	Size(Bytes)
.cinit	romdata	0x000028	program	0x00000e
.code_LED756.o	code	0x00002f	program	0x0001be
.code_DELAY.o	code	0x00010e	program	0x000022
_startup	code	0x00011f	program	0x000008
.idata_LED756.o_i	romdata	0x000123	program	0x000000
.romdata_LED756.o	romdata	0x000123	program	0x000000
.idata_DELAY.o_i	romdata	0x000123	program	0x000000
.romdata_DELAY.o	romdata	0x000123	program	0x000000
SFR_UNBANKED0	udata	0x000000	data	0x000010
SFR_BANKED0	udata	0x000010	data	0x000008
SFR_UNBANKED1	udata	0x000018	data	0x000002
.stack	udata	0x000020	data	0x000020
.udata_LED756.o	udata	0x000040	data	0x000006
_tmpstore	udata	0x000046	data	0x000003

Для того чтобы файл карты памяти был создан во время компиляции текста программы, необходимо в диалоговом окне настройки параметров компиляции (*Project > Edit Project - Node Properties*) установить флаг возле строки *Map file*.

### 7.11.7 Window > Stack – открыть окно аппаратного стека.

Глубина стека зависит от конкретного типа микроконтроллера. В окне аппаратного стека указывается адрес возврата из подпрограммы или обработки прерываний.



**Примечание.** Если установлен флаг *Stack Overflow Break (Options>Development Mode>Break Options)*, то в случае переполнения стека выполнение программы будет остановлено.

Содержание стека может быть отображено с указанием номеров строк. Желаемый формат выбирается в системном меню.

**Примечание.** Системное меню открывается нажатием левой кнопки «мыши» в левом верхнем углу окна.

#### 7.11.7.1 Стек 12-разрядных микроконтроллеров

Аппаратный стек 12-разрядных микроконтроллеров PIC12CXXX и PIC16C5X имеет два уровня.

Симулятор MPLAB SIM полностью моделирует работу стека микроконтроллеров PIC12CXXX и PIC16C5X с формированием предупреждающих сообщений в случае переполнения и исчерпания стека.

Запись в стек PC+1 происходит при выполнении инструкции CALL или возникновении прерываний.

Чтение из стека происходит при выполнении инструкции RETLW.

#### 7.11.7.2 Стек 14-разрядных микроконтроллеров

Аппаратный стек 14-разрядных микроконтроллеров PIC14000 и PIC16XXX имеет восемь уровней.

Симулятор MPLAB SIM полностью моделирует работу стека микроконтроллеров PIC14000 и PIC16XXX с формированием предупреждающих сообщений в случае переполнения и исчерпания стека.

Запись в стек PC+1 происходит при выполнении инструкции CALL или возникновении прерываний.

Чтение из стека происходит при выполнении инструкций RETLW, RETURN и RETFIE.

#### 7.11.7.3 Стек 16-разрядных микроконтроллеров PIC17CXXX

Аппаратный стек 16-разрядных микроконтроллеров PIC17CXXX имеет 16 уровней.

Симулятор MPLAB SIM полностью моделирует работу стека микроконтроллеров PIC17CXXX с формированием предупреждающих сообщений в случае переполнения и исчерпания стека.

Запись в стек PC+1 происходит при выполнении инструкций CALL, LCALL или возникновении прерываний.

Чтение из стека происходит при выполнении инструкций RETLW, RETURN и RETFIE.

В случае переполнения стека устанавливается бит STAKAVL, который сбрасывается при аппаратном сбросе микроконтроллера.

#### 7.11.7.4 Стек 16-разрядных микроконтроллеров PIC18CXXX

Аппаратный стек 16-разрядных микроконтроллеров PIC18CXXX имеет 31 уровень.

Симулятор MPLAB SIM полностью моделирует работу стека микроконтроллеров PIC18CXXX с формированием предупреждающих сообщений в случае переполнения и исчерпания стека.

Запись в стек PC+1 происходит при выполнении инструкций CALL, LCALL или возникновении прерываний.

Чтение из стека происходит при выполнении инструкций RETLW, RETURN и RETFIE.

В случае переполнения стека устанавливается бит STAKAVL, который сбрасывается при аппаратном сбросе микроконтроллера.

**Примечание.** Если флаг не установлен *Stack Overflow Break (Options>Development Mode>Break Options)*, то в случае переполнения стека MPLAB IDE сформирует предупредительное сообщение (только для 14/16-разрядных микроконтроллеров).

### 7.11.8 Window > File Registers – открыть окно памяти данных.

В окне отображаются значения всех регистров памяти данных. Изменение содержимого регистра обозначается изменением цвета соответствующего регистра. При выполнении программы в режиме реального времени значения регистров в окне не обновляется.

**Примечание.** Если выполнение программы остановлено и изменилось состояние порта ввода/вывода, то значение регистров SFR изменится только после выполнения шага программы.

Окно памяти данных может быть постоянно открыто, его можно перемещать по экрану и изменять его размер.

Содержание регистра памяти данных можно изменить несколькими способами:

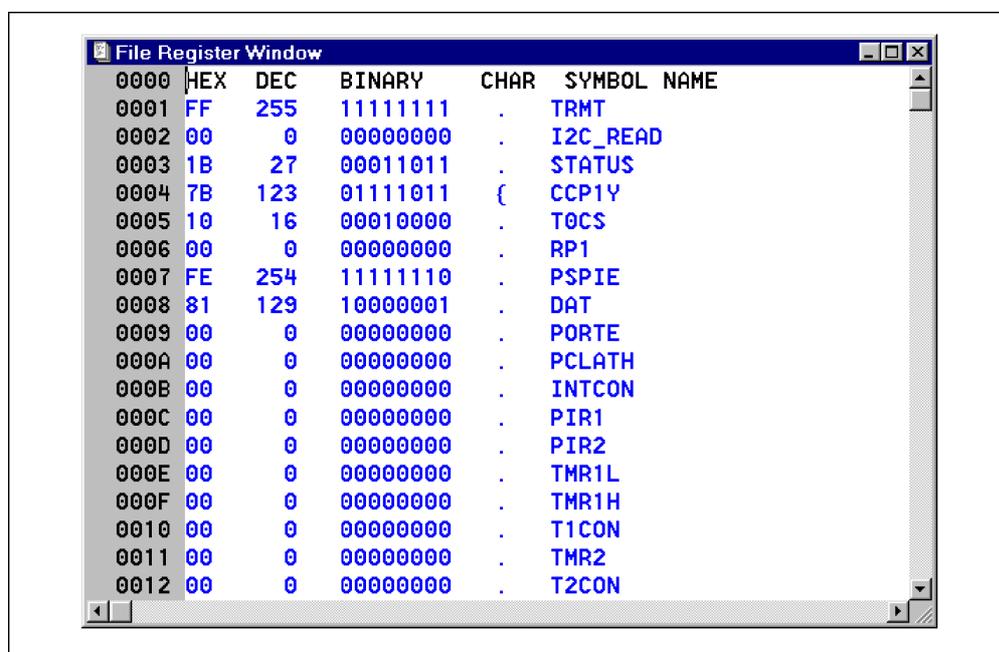
- двойным нажатием левой кнопкой «мыши» на нужном регистре и вводом нового значения;
- в окне изменений (*Window > Modify*).

В системном меню окна можно указать параметры вывода данных:

**Примечание.** Системное меню открывается нажатием левой кнопкой «мыши» в левом верхнем углу окна.

*Toggle Line Numbers* – показывать номер строки;

*Symbolic Display* – отображать данные в символьном виде;

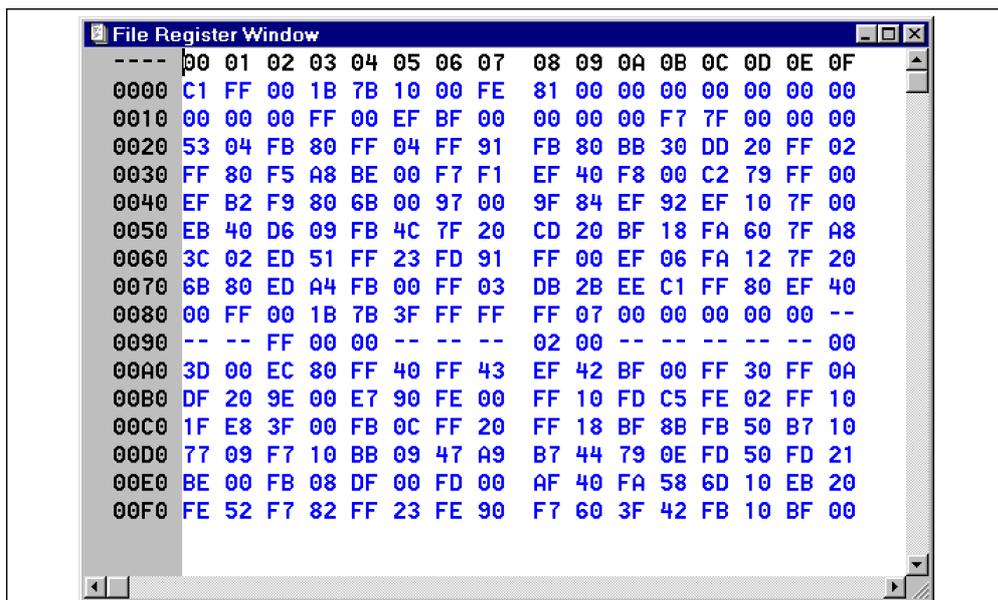


0000	HEX	DEC	BINARY	CHAR	SYMBOL NAME
0001	FF	255	11111111	.	TRMT
0002	00	0	00000000	.	I2C_READ
0003	1B	27	00011011	.	STATUS
0004	7B	123	01111011	{	CCP1Y
0005	10	16	00010000	.	T0CS
0006	00	0	00000000	.	RP1
0007	FE	254	11111110	.	PSPIE
0008	81	129	10000001	.	DAT
0009	00	0	00000000	.	PORTE
000A	00	0	00000000	.	PCLATH
000B	00	0	00000000	.	INTCON
000C	00	0	00000000	.	PIR1
000D	00	0	00000000	.	PIR2
000E	00	0	00000000	.	TMR1L
000F	00	0	00000000	.	TMR1H
0010	00	0	00000000	.	T1CON
0011	00	0	00000000	.	TMR2
0012	00	0	00000000	.	T2CON

Окно содержит следующую информацию:

- Первая колонка – шестнадцатеричное значение;
- Вторая колонка – десятичное значение;
- Третья колонка – двоичное значение;
- Четвертая колонка – соответствующий символ;
- Пятая колонка – имя регистра.

*Hex Display* – отображать память в HEX формате;



*ASCII Display* – отображать память в ASCII формате.

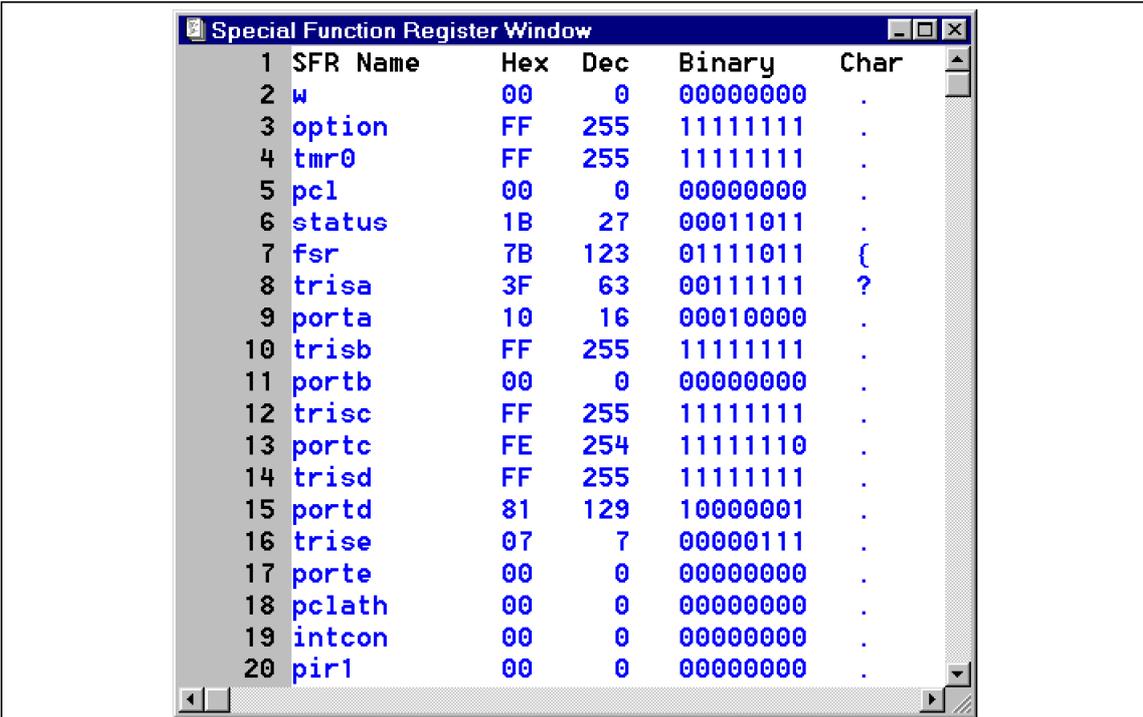
### 7.11.9 Window > Special Function Registers – открыть окно регистров специального назначения.

В окне отображаются регистры специального назначения, состав которых зависит от типа выбранного микроконтроллера. Формат окна регистров специального назначения более удобен, чем окно памяти данных, т.к. в него включены все регистры SFR и их значения представлены в различных вариантах. Обновление значения регистров происходит при каждой остановке выполнения программы.

**Примечание.** Если выполнение программы остановлено и изменилось состояние порта ввода/вывода, то значение регистров SFR изменится только после выполнения шага программы.

Окно памяти данных может быть постоянно открыто, его можно перемещать по экрану и изменять его размер.

**Примечание.** Системное меню открывается нажатием левой кнопкой «мыши» в левом верхнем углу окна.



	SFR Name	Hex	Dec	Binary	Char
1	w	00	0	00000000	.
3	option	FF	255	11111111	.
4	tmr0	FF	255	11111111	.
5	pcl	00	0	00000000	.
6	status	1B	27	00011011	.
7	fsr	7B	123	01111011	{
8	trisa	3F	63	00111111	?
9	porta	10	16	00010000	.
10	trisb	FF	255	11111111	.
11	portb	00	0	00000000	.
12	trisc	FF	255	11111111	.
13	portc	FE	254	11111110	.
14	trisd	FF	255	11111111	.
15	portd	81	129	10000001	.
16	trise	07	7	00000111	.
17	porte	00	0	00000000	.
18	pclath	00	0	00000000	.
19	intcon	00	0	00000000	.
20	pir1	00	0	00000000	.

Окно содержит следующую информацию:

- Первая колонка – имя регистра;
- Вторая колонка – шестнадцатеричное значение;
- Третья колонка – десятичное значение;
- Четвертая колонка – двоичное значение;
- Пятая колонка – соответствующий символ.

Изменить значение регистра SFR можно несколькими способами:

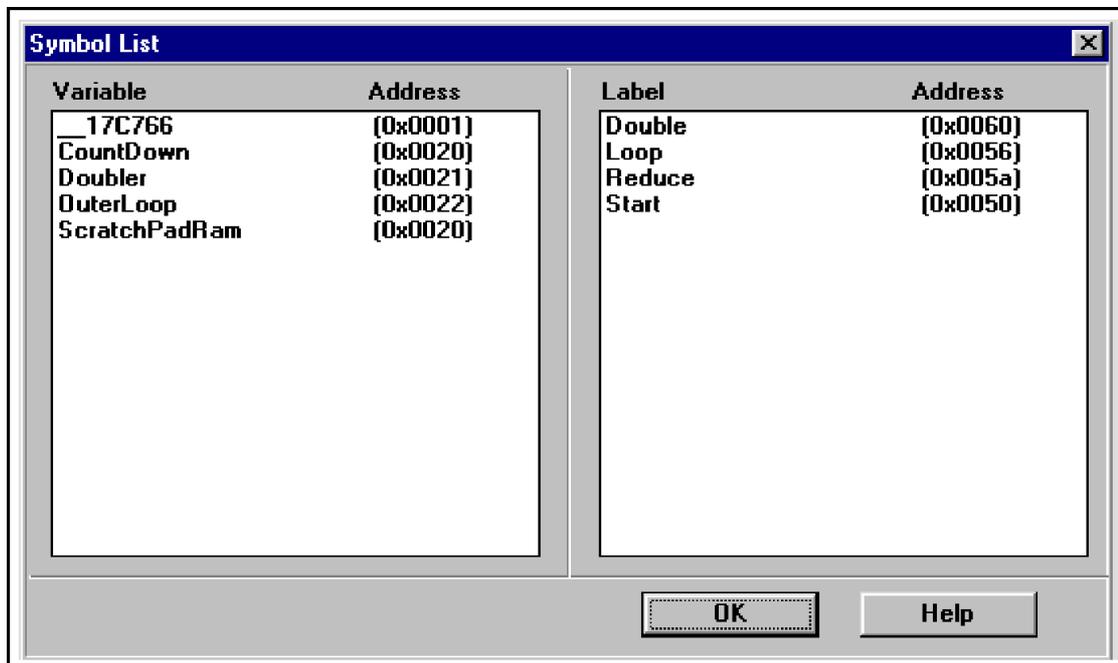
- двойным нажатием левой кнопкой «мыши» на нужном регистре и вводом нового значения;
- в окне изменений (*Window > Modify*).

**Примечание.** Название регистров SFR и их адреса соответствуют выбранному микроконтроллеру.

### 7.11.10 Window > Show Symbol List – открыть окно списка переменных и меток.

В окне показан список переменные (*Variable*), констант и меток программы (*Label*), импортированные в файл \*.COD после компиляции исходного текста программы.

**Примечание.** Для вызова данного окна проект должен быть открыт, а исходные файлы откомпилированы.



*Variable, Address*

Отображаются имена переменных и их адреса в памяти данных.

*Label, Address*

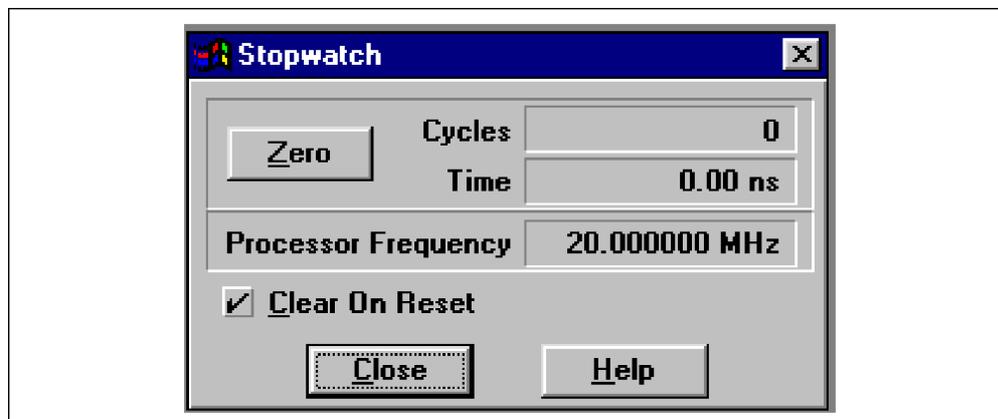
Отображаются имена меток и их адреса в памяти программ.

*Constants*

Константы, определенные в исходном тексте программы.

### 7.11.11 Window > Stopwatch – открыть окно секундомера.

Системный секундомер считает число циклов и время работы микроконтроллера от последнего сброса. Подсчет происходит при работе в режиме реального времени, анимации и пошаговом выполнении программы. Секундомер позволяет Вам измерить время выполнения программы. Вычисление времени производится на основе тактовой частоты микроконтроллера, которую можно установить в меню *Options > Development Mode* раздел *Clock*.



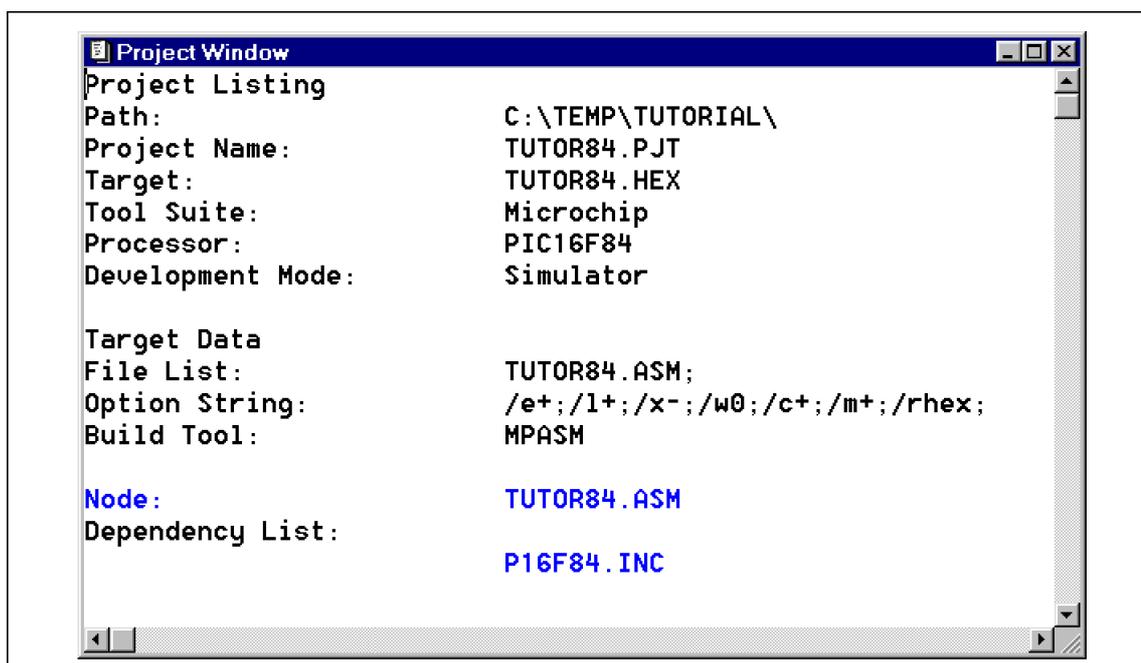
<i>Cycles</i>	Номер цикла от последнего сброса.
<i>Time</i>	Прошедшее время от последнего сброса. Соответствует числу выполненных циклов.
<i>Zero</i>	Сброс значений окна.
<i>Processor Frequency</i>	Тактовая частота микроконтроллера, установленная в пункте меню <i>Options &gt; Development Mode</i> .
<i>Clear on Reset</i>	Сброс значений при сбросе микроконтроллера.

**Пример.** Измерение времени выполнения подпрограммы.

Установите точки останова в начале и конце подпрограммы. Запустите выполнение программы, в точке останова начала подпрограммы сбросьте секундомер. После останова в конце подпрограммы на секундомере будут данные времени выполнения.

### 7.11.12 Window > Project – открыть окно проекта.

Окно доступно только при открытом проекте. В окне отображаются текущий список используемых файлов. После компиляции текста программы в список будут включены и подключаемые файлы. Двойной щелчок левой кнопки «мыши» на имени файла в окне, откроет файл в окне редактора MPLAB.



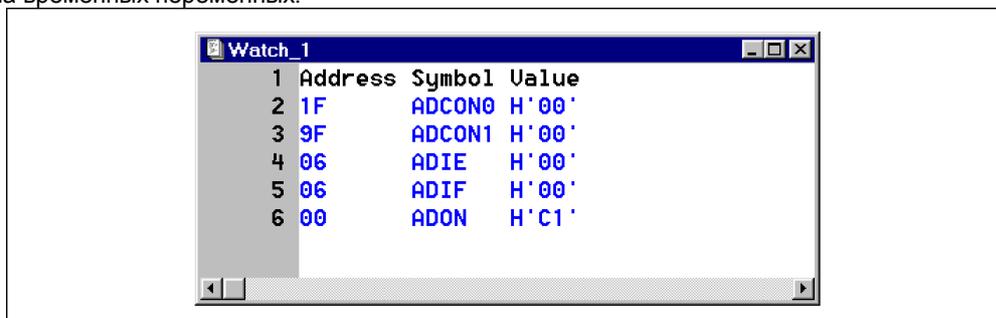
### 7.11.13 Окна с переменными (Window > Watch Window)

MPLAB IDE позволяет создавать окна с составом регистров определяемым пользователем. Используя системное меню окна, можно управлять форматом вывода данных на экран.

#### 7.11.13.1 Создание окна с переменными

*Window > Watch Window > New Watch Window* – создать новое окно с переменными. Описание настройки параметров окна смотрите в разделе 7.11.13.5.

Пример окна временных переменных.



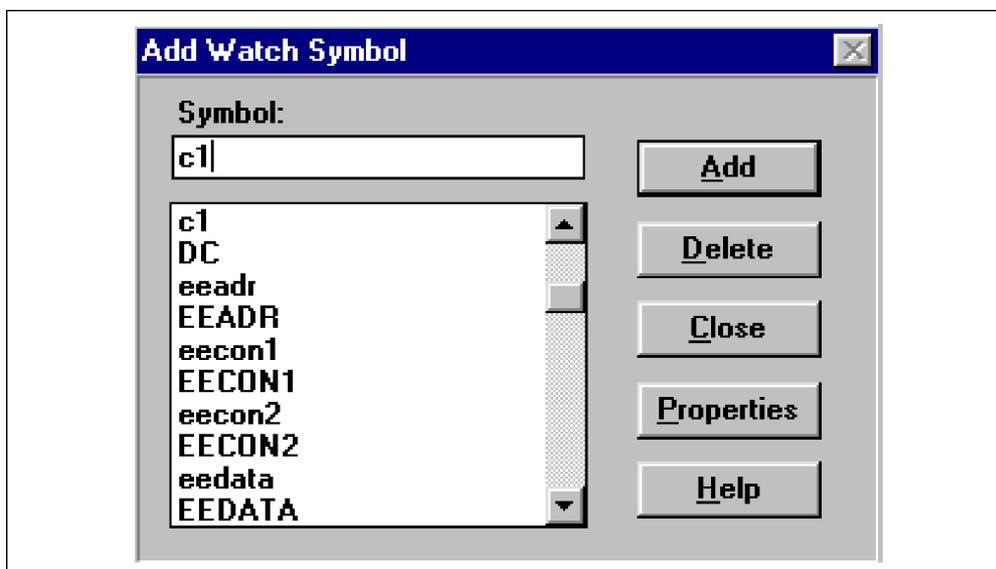
#### 7.11.13.2 Загрузка сохраненного окна

*Window > Watch Window > Load Watch Window* – загрузить сохраненное окно временных переменных. В открывшемся диалоговом окне укажите диск, директорию, выберите нужный файл и нажмите кнопку ОК.

В системном меню окна выберите *Add Watch Symbol* для добавления новой переменной в окно. Описание настройки параметров окна смотрите в разделе 7.11.13.5.

#### 7.11.13.3 Добавить переменную в активное окно

*Window > Watch Window > Add to Active Watch* – добавить переменную в активное окно.



Выберите пункт меню *Window > Watch Window > Add* или в системном меню окна *Add Watch Symbol*.

Впишите имя переменной (или адрес, например 0x40) в строке *Symbol* и нажмите кнопку *Add*.

Для настройки параметров вывода данных в окне нажмите кнопку *Properties* (описание смотрите в разделе 7.11.13.5).

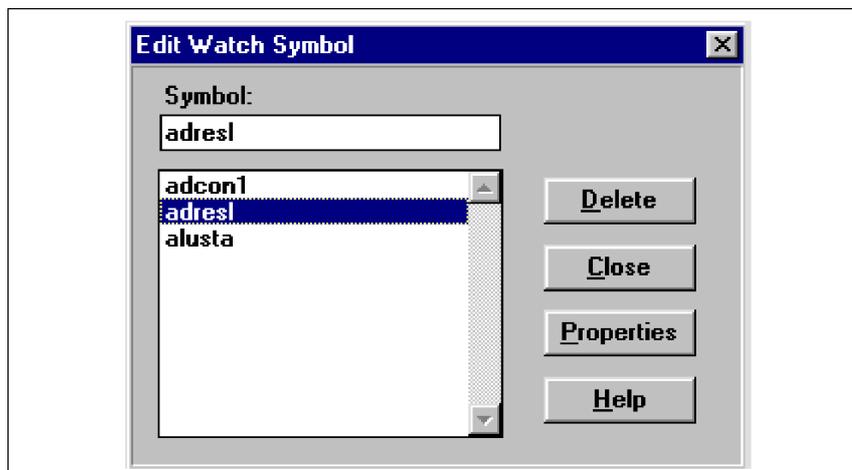
После того как Вы добавили все нужные переменные, нажмите кнопку *Close* для закрытия диалогового окна.

**Примечание.** Переменная с именем W(w) отображает значение регистра W (имеет адрес 0).

#### 7.11.13.4 Редактирование состава переменных в активном окне

Используя системное меню окна, можно управлять форматом вывода данных на экран.

Выберите *Window > Watch Window > Edit Active Watch* или в системном меню окна *Edit Watch* для открытия диалогового окна, показанного на рисунке.



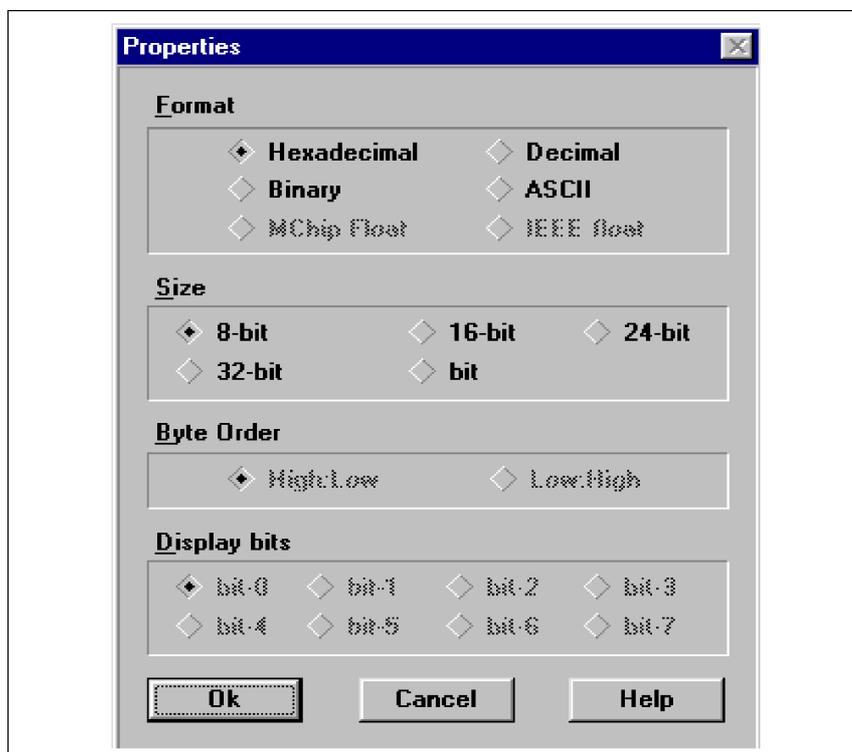
Для удаления символа из активного окна, выберите его из списка и нажмите кнопку *Delete*. Или, укажите переменную и в системном меню окна выберите *Delete Watch*.

Настроить параметры вывода данных в окне, можно нажав кнопку *Properties* (описание смотрите в разделе 7.11.13.5).

После того как Вы отредактировали состава переменных, нажмите кнопку *Close* для закрытия диалогового окна.

### 7.11.13.5 Настройка параметров отображения значений

В диалоговом окне *Properties* Вы можете настроить формат отображения переменной. Вызвать это окно можно, нажав кнопку *Properties* в диалоговом окне редактирования или добавления переменной, после выбора переменной из списка.



<i>Format</i>		Формат отображения данных:
	<i>Hexadecimal</i>	Шестнадцатеричный
	<i>Binary</i>	Двоичный
	<i>Mchip Float</i>	Mchip с плавающей точкой. Формат доступен только с 32-разрядными переменными.
	<i>Decimal</i>	Десятичный
	<i>ASCII</i>	Символы ASCII
	<i>IEEE Float</i>	IEEE с плавающей точкой. Формат доступен только с 32-разрядными переменными.
<i>Size</i>		Разрядность выводимых данных.
<i>Byte Order</i>		Старший или младший байт, только для 16, 24, 32 –разрядных данных
<i>Display Bits</i>		Номер бита, только когда установлен флаг <i>bit</i> в разделе <i>Size</i> .
<i>Ok</i>		Закреть диалоговое окно с сохранением изменений.
<i>Cancel</i>		Закреть диалоговое окно без сохранения изменений.

### 7.11.13.6 Сохранение окна

Выберите *Window > Watch Window > Save Watch Window* для сохранения активного окна на диске. В диалоговом окне сохранения выберите диск, директорию, укажите имя файла и нажмите кнопку *Ok*.

### 7.11.14 Открыть окно изменений

Выберите *Window > Modify* для просмотра и/или изменения содержимое памяти программ, памяти данных, стека или EEPROM памяти.

В окне *Modify* можно прочитать/записать значение по определенному адресу, прочитать/записать значение с приращением адреса или заполнить определенный диапазон памяти указанным значением. Окно может быть постоянно открыто на рабочем столе среды проектирования.



Открыть окно изменений можно несколькими способами:

- выбрать пункт меню *Window > Modify*;
- двойным щелчком «мыши» на выбранном регистре в окне регистров специальных функций;
- двойным щелчком «мыши» в окне секундомера;
- установить курсор на нужном регистре в окне памяти данных, нажать правую кнопку «мыши», выбрать пункт *Fill Register(s)*.

<i>Address</i>	Адрес в памяти микроконтроллера. Вы можете ввести числовой адрес или символьную метку.
<i>Data/Opcode</i>	Нажмите <i>Read</i> для чтения данных с указанного адреса. Нажмите <i>Write</i> для записи данных по указанному адресу.
<i>Radix</i>	Шестнадцатеричный или десятичный формат.
<i>Memory Area</i>	Тип памяти: <i>Data Memory</i> Память данных <i>Program Memory</i> Память программ <i>Stack</i> Стек в микроконтроллере <i>EEPROM</i> EEPROM память данных
<i>End Address</i>	Адрес конца области памяти для изменения.
<i>Fill Range</i>	Заполнять диапазон значением, введенным в <i>Data/Opcode</i> .
<i>Auto Increment</i>	Авто-инкремент адреса при чтении/записи.

**Примечание.** Авто-инкремент выполняет приращение к текущему значению адреса и отображает содержание следующего адреса. В случае чтения диапазона адресов необходимо указать первый адрес меньше на единицу требуемого, потому что первое чтение увеличит этот адрес.

<i>Write</i>	Введите значение в строку <i>Data/Opcode</i> и нажмите кнопку <i>Write</i> (Вы можете вводить данные в символьном виде). После записи значения информация во всех окнах обновится.
<i>Read</i>	Прочитать данные из памяти.
<i>Close</i>	Закрыть диалоговое окно.

**Примечание.** Изменения значения стека или памяти программ не будут приняты до тех пор, пока выполнение программы не будет приостановлено.

#### 7.11.15 Разместить окна горизонтально

Команда *Window > Tile Horizontal* размещает открытые окна в горизонтальном, не перекрывающем формате друг над другом с максимальной шириной. Если открытых окон очень много, то, начиная с нижней части экрана, окна размещаются по несколько штук в одной линии.

Окна, содержащие команды инструментальных средств (*Tools > DOS Command*) размещаются в верхней части экрана.

#### 7.11.16 Разместить окна вертикально

Команда *Window > Tile Vertical* размещает открытые окна в вертикальном, не перекрывающем формате рядом друг с другом с максимальной глубиной. Если открытых окон очень много, то, начиная с нижней части экрана, окна размещаются по несколько штук в одной колонке.

#### 7.11.17 Разместить окна каскадом

Команда *Window > Cascade* размещает открытые окна каскадом.

#### 7.11.18 Свернуть все окна

Команда *Window > Iconize All* сворачивает все открытые окна.

#### 7.11.19 Упорядочить заголовки свернутых окон

Команда *Window > Arrange Icons* упорядочивает заголовки свернутых окон так, чтобы они были видны внизу рабочего стола. На открытые окна команда не распространяется.

## 7.12 Меню Help

### 7.12.1 Release Notes (Shift+F1)

*Help > Release Notes* – показать отличия от предыдущей версии MPLAB IDE (содержание файла *Readme.lab*).

### 7.12.2 Tool Release Notes

Если Вы используете дополнительные инструментальные средства, то в меню Help может быть пункт помощи по этому инструментальному средству.

### 7.12.3 MPLAB IDE Help

*Help > MPLAB Help* – помощь к среде проектирования MPLAB IDE.

### 7.12.4 Editor Help

*Help > Editor Help* – помощь к редактору MPLAB IDE.

### 7.12.5 Error Help

*Help > Error Help* – описание сообщений об ошибках MPLAB IDE.

### 7.12.6 MPASM Help

*Help > MPASM Help* – открыть интерактивную версию документации MPASM User's Guide with MPLINK and MPLIB (содержит информацию касающуюся MPASM). В этом пункте меню содержится описание директив MPASM и инструкций микроконтроллеров.

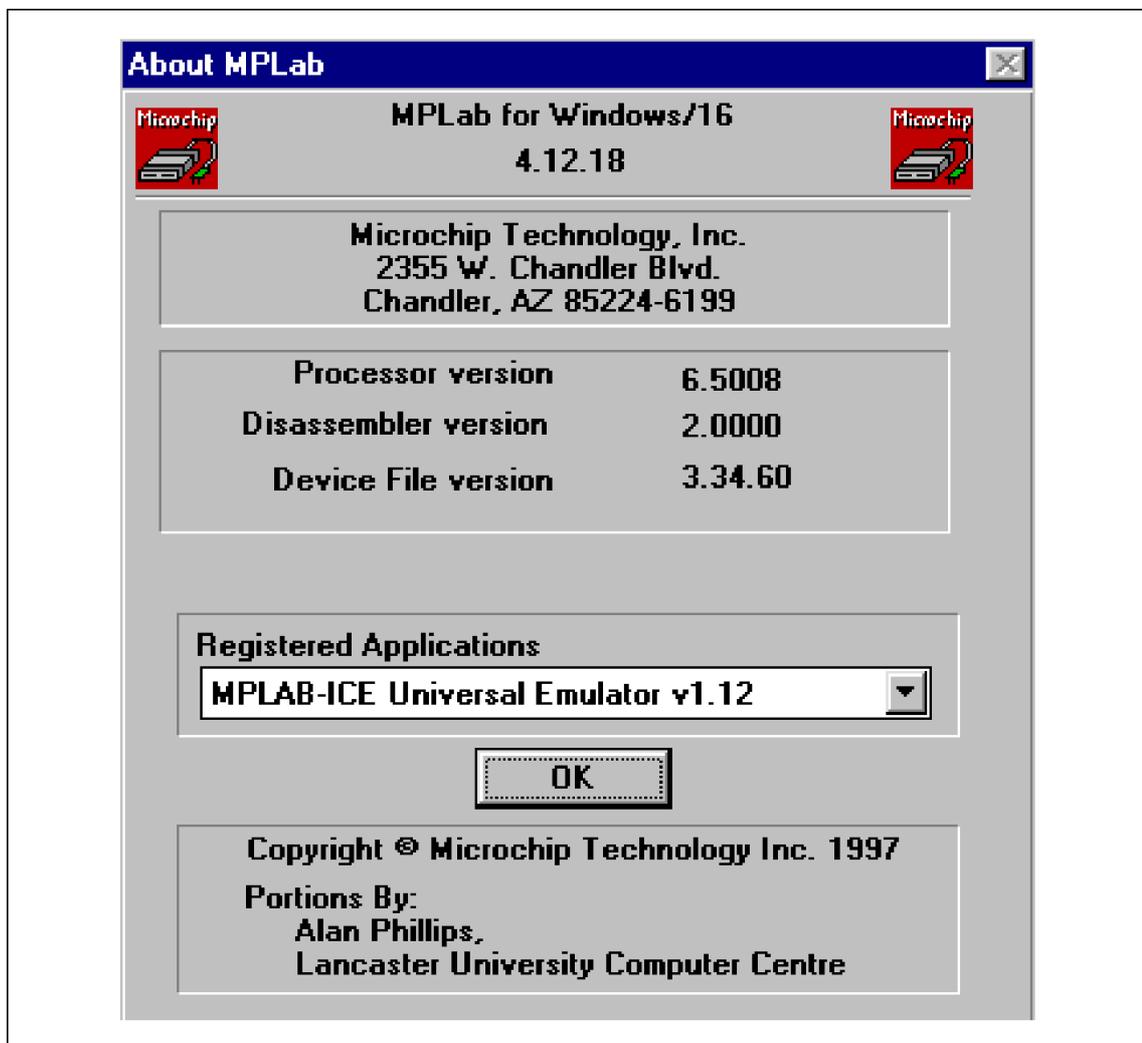
### 7.12.7 MPLINK Help

*Help > MPLINK Help* – открыть интерактивную версию документации MPASM User's Guide with MPLINK and MPLIB (содержит информацию касающуюся MPLINK).

## 7.12.8 About

*Help > About* – выводит следующую информацию о MPLAB IDE:

- Версия MPLAB IDE;
- Адрес компании Microchip;
- Версия процессора;
- Версия дизассемблера;
- Информация, касающаяся других зарегистрированных программ.

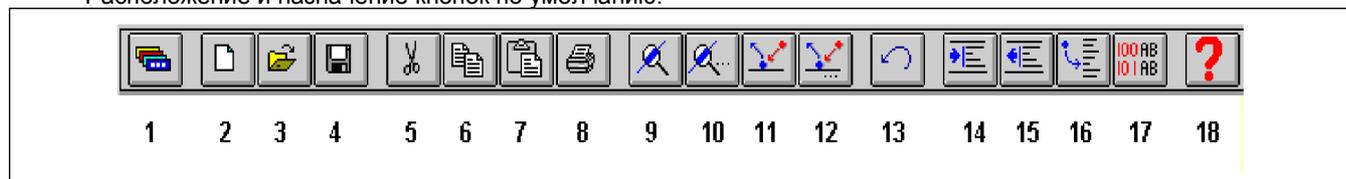


## Приложения

### A. Описание кнопок графического меню MPLAB IDE

#### A.1 Меню редактора MPLAB IDE

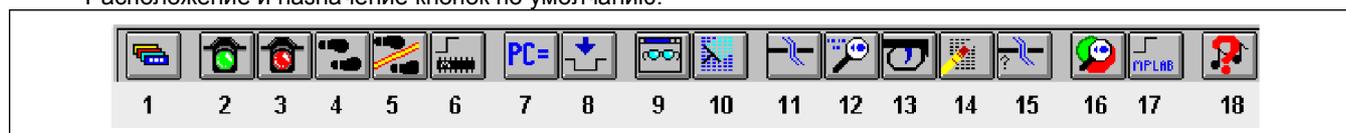
Расположение и назначение кнопок по умолчанию.



1. Выбор графического меню;
2. Создать новый файл;
3. Открыть файл;
4. Сохранить файл;
5. Вырезать в буфер;
6. Копировать в буфер;
7. Вставить из буфера;
8. Печатать;
9. Найти;
10. Повторить поиск;
11. Заменить;
12. Повторить замену;
13. Отменить действие;
14. Увеличить отступ;
15. Уменьшить отступ;
16. Перейти на строку;
17. Показать нумерацию строк;
18. Контекстная помощь.

#### A.2 Меню отладчика

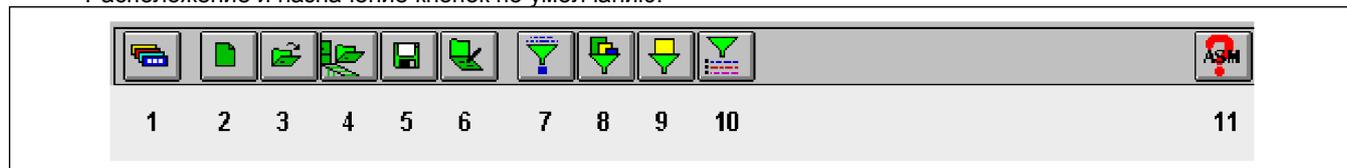
Расположение и назначение кнопок по умолчанию.



1. Выбор графического меню;
2. Запустить выполнение программы с текущего места;
3. Остановить выполнение программы;
4. Выполнить текущую инструкцию программы;
5. Выполнить текущую инструкцию программы (инструкции CALL выполняются за один шаг с полным выполнением подпрограммы);
6. Сброс микроконтроллера;
7. Выбрать значение счетчика команд PC;
8. Добавить инструкцию;
9. Новое окно с переменными;
10. Окно изменений;
11. Точка остановки;
12. Точка трассировки;
13. Триггер;
14. Сбросить все точки остановки;
15. Условная остановка;
16. Прекратить запись в буфер трассировки;
17. Системный сброс;
18. Отличия версии.

### А.3 Меню проекта

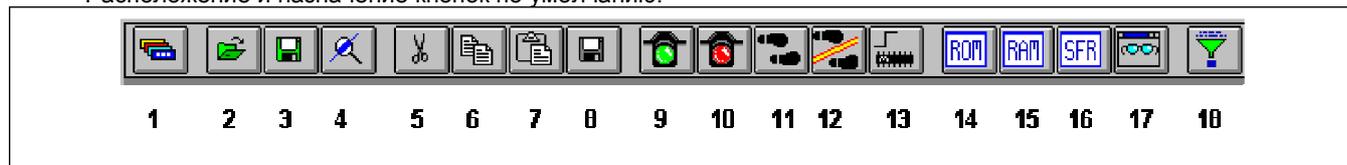
Расположение и назначение кнопок по умолчанию.



1. Выбор графического меню;
2. Создать новый проект;
3. Открыть проект;
4. Закрыть проект;
5. Сохранить проект;
6. Редактировать параметры проекта;
7. Выполнить проект;
8. Компиляция всех исходных файлов;
9. Компиляция текущего исходного файла;
10. Подключение модулей к проекту;
11. Помощь.

### А.4 Меню пользователя

Расположение и назначение кнопок по умолчанию.



1. Выбор графического меню;
2. Открыть проект;
3. Закрыть проект;
4. Найти;
5. Вырезать в буфер;
6. Копировать в буфер;
7. Вставить из буфера;
8. Сохранить файл;
9. Запустить выполнение программы с текущего места;
10. Остановить выполнение программы;
11. Выполнить текущую инструкцию программы;
12. Выполнить текущую инструкцию программы (инструкции CALL выполняются за один шаг с полным выполнением подпрограммы);
13. Сброс микроконтроллера;
14. Окно памяти программ;
15. Окно памяти данных;
16. Окно регистров специального назначения;
17. Новое окно с переменными;
18. Выполнить проект.

**В. Назначение полей линейки состояния**

Ln 100 Col 2	2048	RO	No Wrap	INS	PIC16C622	pc:0x58	w:0x00	-- z dc c	Bk On	Em 0	Debug
--------------	------	----	---------	-----	-----------	---------	--------	-----------	-------	------	-------

Формат поля	Пример отображения	Описание	Результат двойного нажатия левой кнопкой «мыши»
Строка №, Колонка № в активном окне или Версия MPLAB IDE	Ln 1 Col 1  5.00.00	Номер строки и колонки размещения курсора в активном окне редактора  Номер версии MPLAB IDE	Открытие диалога перехода на определенную строку  Не активно
Строк в файле	72	Отображает количество строк в активном окне редактора MPLAB IDE	Не активно
Модификация файла	#	Символ «#» появляется, если редактированный файл не был сохранен	Не активно
Защита от редактирования	WR	Включение/отключение защиты от редактирования: WR – редактирование разрешено RD – только чтение	Включает/отключает защиту от редактирования
Выравнивание текста	No Warp	Отображаются параметры выравнивания текста No Warp – выравнивание нет WR 72 – выравнивание текста, максимальное число символов в линии 72	Включает/выключает выравнивание текста
Параметр ввода текста	INS	Отображается параметр ввода текста INS – вставка OVR - замещение	Переключает параметр ввода текста вставка/замещение
Тип микроконтроллера	PIC16C61	Отображается тип микроконтроллера используемого в проекте	Не активно
Счетчик команд	PC:0x5f	Текущее значение счетчика команд PC	Открывает диалог выбора значения счетчика команд
Регистр W	W:0x00	Значение регистра W	Не активно
Биты статуса	ov Z dc c	Состояние битов ALU Верхний регистр – бит установлен Нижний регистр – бит сброшен	Не активно
Флаг разрешения прерываний	Bk On	Глобальное разрешение прерываний	Разрешает/запрещает прерывания
Среда проектирования	Sim	EO – только редактор Sim – симулятор MPLAB SIM Si – симулятор SIMICE ICE – эмулятор MPLAB-ICE Em – эмулятор PICMASTER	Открывает окно настройки проекта
Частота	4MHz	Тактовая частота микроконтроллера	Открывает окно настройки проекта
Режим	Edit	Режим среды проектирования MPLAB IDE	Не активно

### **С. Назначение файлов среды MPLAB IDE**

*.ASM	исходный файл на ассемблере
*.C	исходный файл на C
*.CFG	файл конфигурации
*.CSV	файл трассировки, только для MPLAB-ICE 2000
*.COD	объектный код
*.DAT	файл данных симулятора
*.ERR	список ошибок при компиляции
*.H	дополнительный файл C
*.HEX	код для микроконтроллера PICmicro
*.HLP	файл помощи
*.INC	настройка MPASM
*.INI	настройка MPLAB IDE
*.KEY	назначение клавиш в MPLAB IDE
*.LKR	файл сценария для MPLINK
*.LST	файл листинга программы
*.PJT	файл проекта
*.REG	файл стимула регистра
*.STI	файл стимула вывода
*.TB	файл настройки точек остановки
*.TBR	настройки графического меню
*.TPL	файл шаблонов
*.TRC	файл буфера трассировки
*.WAT	файл окна с переменными

## D. Функциональное назначение кнопок клавиатуры

В этом приложении представлены таблицы функционального назначения кнопок и комбинации кнопок в среде проектирования MPLAB IDE настроенные по умолчанию.

### D.1 Функциональные кнопки

<b>F1</b>	Вызвать диалоговое окно помощи
<b>F2</b>	<i>Debug &gt; Break Settings</i> - открыть диалоговое окно настройки точек останова
<b>F3</b>	<i>Edit &gt; Find</i> – поиск текста
<b>F4</b>	<i>Edit &gt; Replace</i> – поиск и замена текста
<b>F5</b>	<i>Debug &gt; Halt</i> – остановить выполнение программы
<b>F6</b>	<i>Debug &gt; Reset</i> – сброс микроконтроллера в эмуляторе или симуляторе
<b>F7</b>	<i>Debug &gt; Step</i> – выполнить одну инструкцию программы
<b>F8</b>	<i>Debug &gt; Step Over</i> – выполнить текущую инструкцию программы (инструкции CALL выполняются за один шаг с полным выполнением подпрограммы);
<b>F9</b>	<i>Debug &gt; Run</i> – запустить программы на выполнение
<b>F10</b>	<i>Project &gt; Make Project</i> – выполнить проект
<b>F11</b>	<i>Execute &gt; DOS Command To Window</i> – выполнит DOS программу
<b>Shift + F3</b>	<i>Edit &gt; Repeat Find</i> – повторный поиск текста
<b>Shift + F4</b>	<i>Edit &gt; Repeat Replace</i> – повторный поиск и замена текста
<b>Shift + F5</b>	<i>Debug &gt; Halt Trace</i> – остановить запись в буфер трассировки
<b>Shift + F9</b>	<i>File &gt; Close All</i> – закрыть все файлы и окна
<b>Ctrl + F2</b>	<i>Project &gt; Open Project</i> – открыть сохраненный проект
<b>Ctrl + F3</b>	<i>Project &gt; Edit Project</i> – открыть диалоговое окно настройки параметров проекта
<b>Ctrl + F7</b>	<i>Options &gt; Environment Setup</i> – открыть диалоговое окно настройки параметров среды проектирования
<b>Ctrl + F8</b>	<i>Windows &gt; Show Symbol List</i> – открыть окно списка переменных и меток проекта
<b>Ctrl + F10</b>	<i>Project &gt; Build All</i> – компилировать все исходные файлы
<b>Alt + F4</b>	<i>File &gt; Exit</i> – завершить работу с MPLAB IDE
<b>Alt + F10</b>	<i>Project &gt; Build Node</i> – компилировать исходный файл
<b>Shift + Ctrl + F2</b>	<i>Debug &gt; Clear Program Memory</i> – очистить память программ
<b>Shift + Ctrl + F3</b>	<i>Debug &gt; System Reset</i> – выполнить системный сброс
<b>Shift + Ctrl + F5</b>	<i>Debug &gt; POR</i> – выполнить сброс по включению питания

## D.2 Кнопки управления перемещением курсора

<b>Up</b>	Переместить курсор на одну строку вверх
<b>Shift+Up</b>	Переместить курсор на одну строку вверх с выделением фрагмента текста
<b>Down</b>	Переместить курсор на одну строку вниз
<b>Shift+Down</b>	Переместить курсор на одну строку вниз с выделением фрагмента текста
<b>Left</b>	Переместить курсор на один символ влево
<b>Shift+Left</b>	Переместить курсор на один символ влево с выделением фрагмента текста
<b>Ctrl+Left</b>	Переместить курсор на одно слово влево
<b>Ctrl+Shift+Left</b>	Переместить курсор на одно слово влево с выделением фрагмента текста
<b>Right</b>	Переместить курсор на один символ вправо
<b>Shift+Right</b>	Переместить курсор на один символ вправо с выделением фрагмента текста
<b>Ctrl+Right</b>	Переместить курсор на одно слово вправо
<b>Ctrl+Shift+Right</b>	Переместить курсор на одно слово вправо с выделением фрагмента текста
<b>PgDn</b>	Переместить курсор на одну страницу вниз
<b>Shift+PgDn</b>	Переместить курсор на одну страницу вниз с выделением фрагмента текста
<b>Ctrl+PgDn</b>	Переместить курсор к первому символу последней строки окна
<b>Ctrl+Shift+PgDn</b>	Переместить курсор к первому символу последней строки окна с выделением фрагмента текста
<b>PgUp</b>	Переместить курсор на одну страницу вверх
<b>Shift+PgUp</b>	Переместить курсор на одну страницу вверх с выделением фрагмента текста
<b>Ctrl+PgUp</b>	Переместить курсор к первому символу первой строки окна
<b>Ctrl+Shift+PgUp</b>	Переместить курсор к первому символу первой строки окна с выделением фрагмента текста
<b>Home</b>	Переместить курсор в начало строки
<b>Shift+Home</b>	Переместить курсор в начало строки с выделением фрагмента текста
<b>Ctrl+Home</b>	Переместить курсор в начало файла
<b>Ctrl+Shift+Home</b>	Переместить курсор в начало файла с выделением фрагмента текста
<b>Alt+Home</b>	Переместить курсор к первому символу текущей строки
<b>Alt+Shift+Home</b>	Переместить курсор к первому символу текущей строки с выделением фрагмента текста
<b>End</b>	Переместить курсор в конец строки
<b>Shift+End</b>	Переместить курсор в конец строки с выделением фрагмента текста
<b>Ctrl+End</b>	Переместить курсор в конец файла
<b>Ctrl+Shift+End</b>	Переместить курсор в конец файла с выделением фрагмента текста

## D.3 Управляющие кнопки

<b>Ctrl+Shift+B</b>	Выделить фрагмент текста внутри фигурных скобок соответствующего уровня
<b>Ctrl+C</b>	<i>Edit &gt; Copy</i> – копировать выделенный фрагмент текста в буфер
<b>Ctrl+G</b>	<i>Edit &gt; Goto Line</i> – перейти на строку №...
<b>Ctrl+H</b>	Удалить символ слева от курсора
<b>Ctrl+I</b>	Вставить символ табуляции
<b>Ctrl+K</b>	<i>Edit &gt; Delete To End Of Line</i> – удалить от курсора до конца строки
<b>Ctrl+Shift+K</b>	<i>Edit &gt; Delete Line</i> – Удалить строку, в которой расположен курсор
<b>Ctrl+N</b>	<i>File &gt; New</i> – создать новый файл
<b>Ctrl+O</b>	<i>File &gt; Open</i> – открыть файл
<b>Ctrl+Shift+O</b>	Разделить текущую строку без перемещения курсора
<b>Ctrl+P</b>	<i>File &gt; Print</i> – открыть диалоговое окно печати файла
<b>Ctrl+Q</b>	<i>Edit &gt; Text Insert ASCII Code</i>
<b>Ctrl+S</b>	<i>File &gt; Save</i> – сохранить файл
<b>Ctrl+T</b>	<i>Edit &gt; Text Transpose Characters</i>
<b>Ctrl+V</b>	<i>Edit &gt; Paste</i> – вставить фрагмент текста из буфера, начиная с текущего расположения курсора
<b>Ctrl+W</b>	<i>Window &gt; Select</i> – переключение между открытыми окнами
<b>Ctrl+Shift+W</b>	Выделить пару фигурных скобок, наиболее близких к курсору
<b>Ctrl+X</b>	<i>Edit &gt; Cut</i> – вырезать выделенный фрагмент текста в буфер
<b>Ctrl+Z</b>	<i>Edit &gt; Undo</i> – отменить последнее действие

## D.4 Кнопки форматирования и редактирования

<b>Enter</b>	Переход на новую строку
<b>BackSpace</b>	Удалить символ справа от курсора
<b>Del</b>	Удалить курсор слева от курсора
<b>Shift+Del</b>	<i>Edit &gt; Cut</i> – вырезать выделенный фрагмент текста в буфер
<b>Ins</b>	Переключение режима вставка/замещение текста
<b>Shift+Ins</b>	<i>Edit &gt; Paste</i> – вставить фрагмент текста из буфера, начиная с текущего расположения курсора
<b>Ctrl+Ins</b>	<i>Edit &gt; Copy</i> – копировать выделенный фрагмент текста в буфер
<b>Tab</b>	Вставить символ табуляции

Перевод основывается на технической документации DS51025D  
компании Microchip Technology Incorporated, USA.

## Уважаемые господа!

ООО «Микро-Чип» поставляет полную номенклатуру комплектующих фирмы **Microchip Technology Inc** и осуществляет качественную техническую поддержку на русском языке.

С техническими вопросами  
Вы можете обращаться по адресу  
[support@microchip.ru](mailto:support@microchip.ru)

По вопросам поставок комплектующих Вы можете обращаться к нам по телефонам:  
**(095) 963-9601**  
**(095) 737-7545**  
и адресу [sales@microchip.ru](mailto:sales@microchip.ru)

На сайте  
[www.microchip.ru](http://www.microchip.ru)

Вы можете узнать последние новости нашей фирмы, найти техническую документацию и информацию по наличию комплектующих на складе.