



ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЕГО (ПОЛНОГО) ОБЩЕГО ОБРАЗОВАНИЯ

ЛИЦЕЙ ПРИ СПБГУТ

Вендор-ориентированный учебный курс в системе
«Старшая профильно-профессиональная школа-ВУЗ-Работодатель»:
«Программирование микроконтроллеров Microchip»

Богураев М.В.

«СОЗДАНИЕ ПРОЕКТА»

Методические указания к выполнению
лабораторной работы

Санкт - Петербург
2013

Богураев М.В. «СОЗДАНИЕ ПРОЕКТА». Методические указания к выполнению лабораторной работы № 1. СПб: ГОУ «Лицей при СПбГУТ», 2013.

ЛАБОРАТОРНАЯ РАБОТА №1 «СОЗДАНИЕ ПРОЕКТА»

Цель работы

Научиться создавать проекты для микроконтроллеров PIC в среде разработки MPLAB IDE, пользоваться компилятором HI-TECH C.

Теоретические основы

Работу с микроконтроллером целесообразно начинать с создания проекта. Проект создают в программе MPLAB IDE, объединяющей набор разных программ, и называемой средой разработки. В среде разработки легко запустить текстовый редактор, компилятор, отладчик или программатор. Можно подключать к проекту файлы и задавать опции компилятора. Для работы не надо каждый раз запускать отдельную программу, указывать место расположения обрабатываемых файлов, задавать опции программ: редактора, компилятора, программатора, таким образом, работа в среде MPLAB IDE облегчает труд разработчика.

Проект создают перед тем, как писать любую программу. В папке проекта помещают файлы с текстом программы (source files «исходники» – файлы с расширением *.c, например, Project1.c). После создания проекта в папке появляются новые файлы, среда разработки MPLAB IDE создаёт их сама. Это файлы workspace – *.mcw, project – *.mcp *.hex и некоторые другие. Файлы *.c нужны для программиста и компилятора – это исходный код, который пишет программист, затем обрабатывает программа-компилятор, она же генерирует файл *.hex, исполняемый микроконтроллером. Файлы с расширением mcw хранят информацию об открытых окнах и их расположении, с расширением mcp содержат информацию о подключенных файлах и настройках MPLAB IDE в этом проекте. Поэтому для компиляции исходного текста не требуется указания места расположения компилируемого файла.

Редактор текста будет настроен для работы на языке СИ, а при компиляции текста будет использована программа HI-TECH C.

Результат работы программиста – файлы с текстом программы, а также результат компиляции – файл с расширением hex. Этот файл при помощи программатора будет помещён в память команд и будет выполняться микроконтроллером.

В программе необходимо подключить библиотечный файл htc.h, который включает файл pic168xa.h. В файле pic168xa.h задаются все имена, которыми программист называет адреса ячеек памяти микроконтроллера PIC16F877A. Например, в файле указано: volatile unsigned char PORTC @ 0x07. Если программист пишет PORTC = 0x00, компилятор анализирует такую конструкцию и записывает нули по адресу 0x07 в памяти данных. Имена совпадают с документацией на микроконтроллер. Директива __CONFIG() настраивает общие параметры микроконтроллера. В программе обязательно должна быть описана функция void main (void) {}. Внутри функции main настраивают частные параметры микроконтроллера и пишут свою программу.

На языке СИ можно писать программы для разных контроллеров. Программа – компилятор автоматически переводит конструкции языка СИ в машинные коды. Компилятор отслеживает синтаксические ошибки, назначает адреса переменных и управляет памятью данных.

Программирование микроконтроллеров отличается от программирования персональных компьютеров. В микроконтроллерах нет «стандартных» устройств ввода – вывода и операционной системы. Любая программа для микроконтроллера, это бесконечный замкнутый цикл, который выполняется от начала работы и до выключения микроконтроллера.

Задание

Перед созданием проектов в корневом каталоге диска C сделайте папку Projects. В этой папке будем делать отдельные папки для каждого проекта, например Project1, Project2 и т. д. Учтите, что программа MPLAB IDE не воспринимает русские буквы в описании пути к файлам. Создайте проект на языке СИ в среде MPLAB IDE. Используйте компилятор HI-TECH C и файл Project1.c с текстом кода. Получите файл Project1.hex. Проанализируйте полученный код программы.

Создайте новый проект в новой папке Project1_2, используя тот же файл с текстом кода-заготовки. Внутри функции main впишите цикл `while(1){ }`

Проанализируйте изменения – сравните размер полученных hex файлов.

Порядок выполнения

Сделайте в корневом каталоге на диске C:\ папку Projects, в ней создайте папку Project1. В папку Project1 скопируйте файл Project1.c.

Запустите программу MPLAB IDE. Проект следует создавать при помощи Project Wizard. Для запуска помощника создания проекта нужно выбрать Project>Project Wizard (рис. 1).

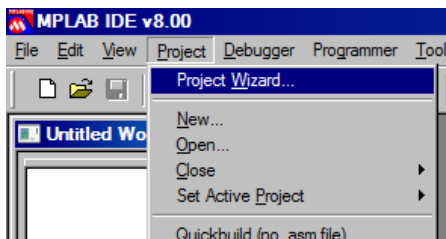


Рис. 1. Вызов помощника создания проектов.

После появления приглашения надо нажать Next> для продолжения (рис. 2).



Рис. 2. Приглашение помощника.

В следующем окне (Step One – первый шаг) выбирают тип микроконтроллера (рис. 3). В выпадающем списке Device выбирают тот микроконтроллер, который установлен в устройстве (в нашем случае PIC16F877A), затем нажимают Next>.

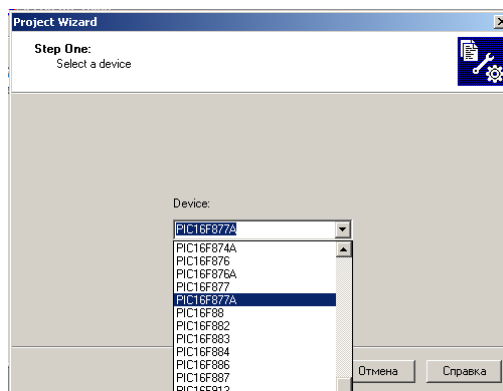


Рис. 3. Первый шаг – выбор контроллера.

В окне выбора языка программирования (Step Two) для проекта на языке Си нужно выбрать HI-TECH Universal ToolSuite (рис. 4). В окне ToolSuite Contents будет указан HI-TECH ANSI C Compiler. А в графе Location должен быть указан путь к компилятору –

C:\Program Files\HI-TECH Software\PICC\PRO\9.XX\bin\picc.exe. После того, как задана программа-компилятор, нажимаем Next>.

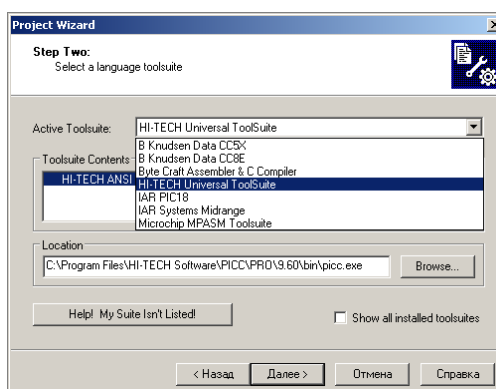


Рис. 4. Второй шаг – указание программы-компилятора.

В появившемся окне (Step Three) нажмём кнопку browse и зададим место и имя для нашего проекта в созданной заранее папке на диске C (рис. 5). Не следует использовать рабочий стол или сетевые диски для хранения подобных папок и файлов. Это связано с тем, что, MPLAB IDE не работает с файлами, путь к которым имеет русские буквы.

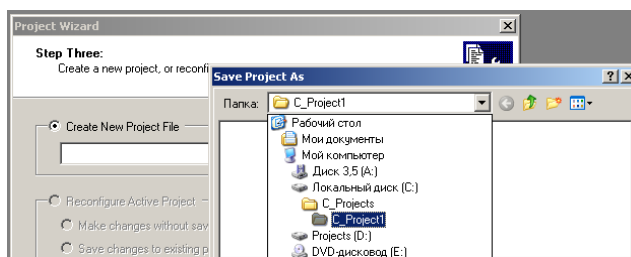


Рис. 5. Выбор места, где будет создан проект.

В папке Project1 зададим имя файла проекта Project1. Нажмём сохранить. Затем нажмём Next> (рис. 6).

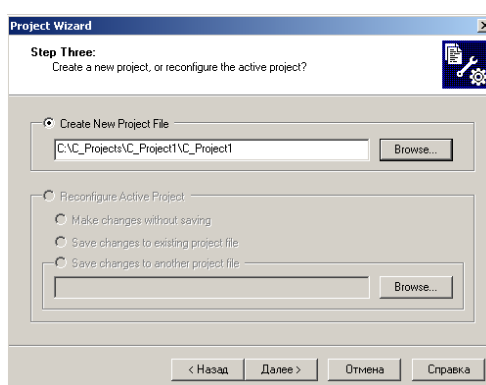


Рис. 6. Третий шаг – создание файла проекта.

Затем (Step Four) сразу нажмём Next> (рис. 7). Файл с исходным текстом добавим в проект позже из среды разработки. Убедитесь, что файл не имеет атрибута «только чтение».

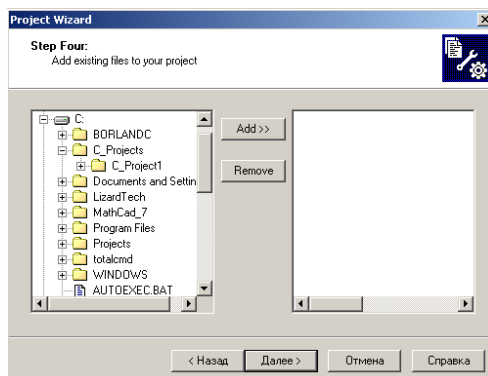


Рис. 7. Четвёртый шаг – нажимаем Next>.

Последнее окно Project Wizard – это резюме проекта. Показываются выбранный микроконтроллер, программа компилятор, имя файла проекта.

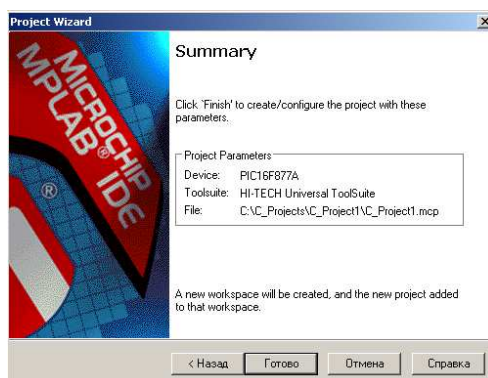


Рис. 8. Сведения о проекте.

После того, как нажата кнопка Finish, на рабочем столе MPLAB IDE должно появиться окно Project. Если окно Project не отображается надо выбрать View>Project (рис. 9).

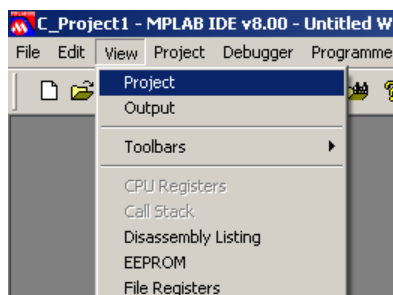


Рис. 9. Вызов окна Project.

В папке проекта должен находится файл Project1.c. Добавим файл в проект из среды разработки: наводим мышь на папку Source Files в окне Project.mcw, щёлкаем правой кнопкой и выбираем Add files (рис.10).

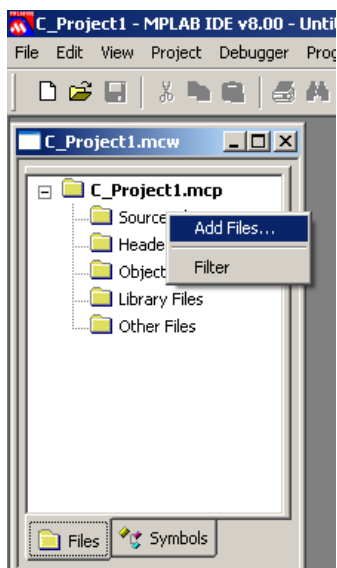


Рис. 10. Добавление файла в проект.

После чего указываем место расположения файла, в нашем случае путь к файлу выглядит следующим образом: C:\Projects\Project1\Project1.c (рис. 11).

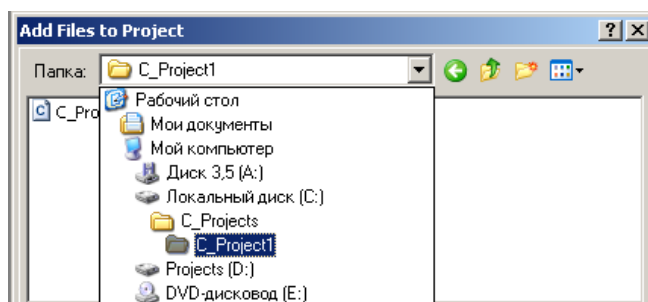


Рис. 11. Место, в котором расположен добавляемый файл.

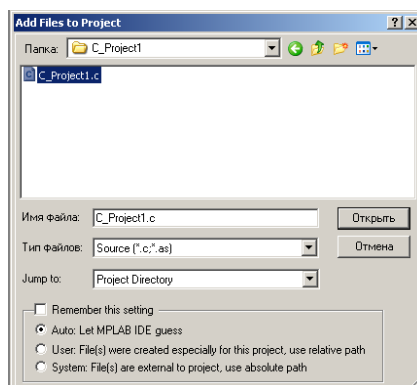


Рис. 12. Выбор добавляемого файла.

Нажимаем кнопку открыть (рис. 12), после чего файл присоединяется к проекту (рис. 13).

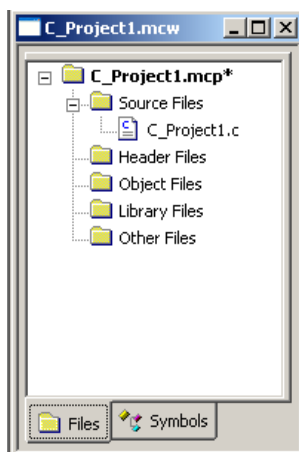


Рис. 13. Вид окна Project с присоединённым файлом.

В корневой папке с названием проекта Project1.mcp* звёздочка у названия обозначает, что были внесены изменения в проект, но проект ещё не был сохранён. Чтобы увидеть текст исходного файла надо дважды щёлкнуть правой кнопкой мыши на названии Project1.c (рис. 14). Для настройки текстового редактора нужно щёлкнуть правой кнопкой мыши на тексте программы и выбрать Properties (рис. 15).

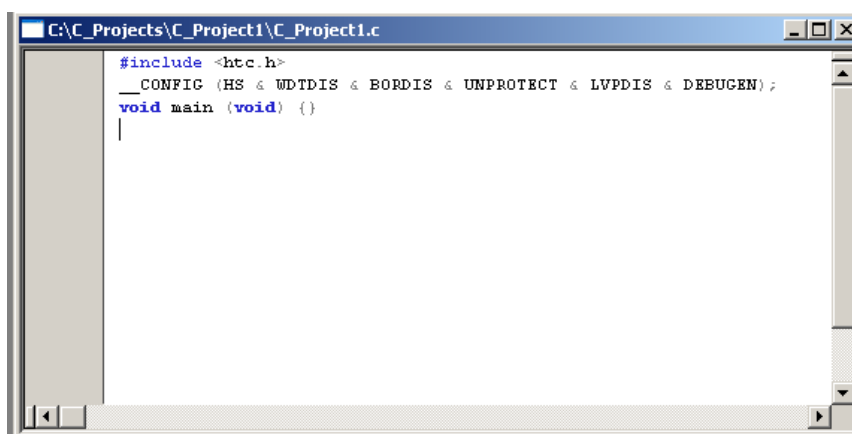


Рис.14. Окно редактора с текстом программы.

В появившемся окне выберем вкладку 'C' File Types и поставим галочку напротив Line Numbers, щёлкнув правой кнопкой мыши в квадрате (рис. 16). После этого слева от строк текста появятся их номера. Для компиляции исходного текста программы нажимаем на кнопку Rebuild (рис. 17).

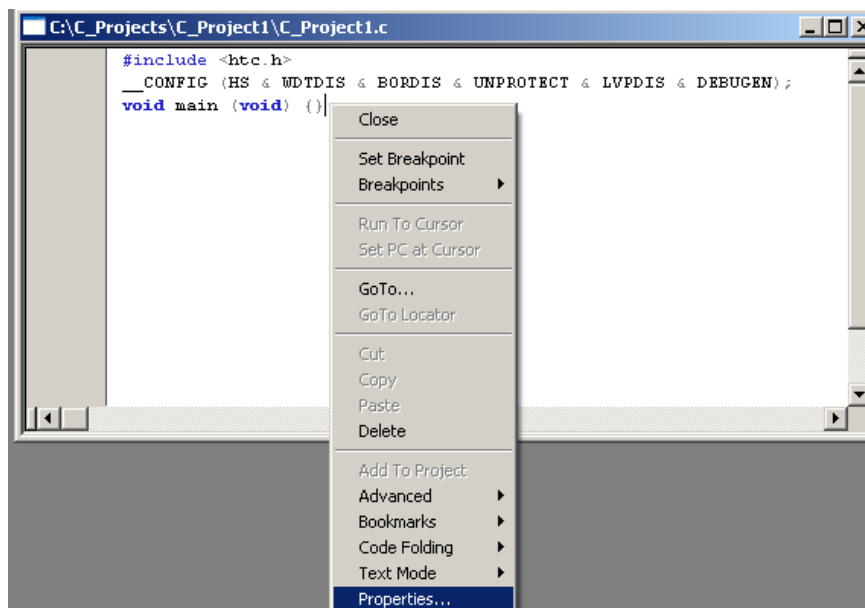


Рис. 15. Меню настроек текстового редактора.

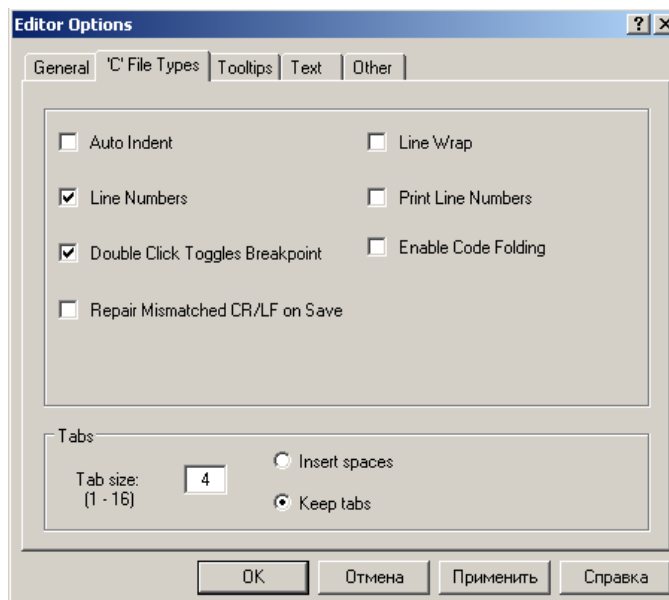


Рис. 16. Меню настройки редактора текстов.

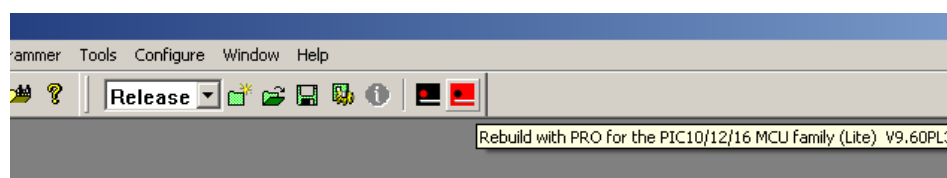


Рис. 17. Кнопка Rebuild для запуска компилятора.

Если компиляция прошла успешно, то в окне output не будет сообщений об ошибках (рис. 18). Теперь посмотрим содержимое памяти команд: откроем окно Program Memory (рис. 19). В окне Program Memory выберем вкладку Symbolic (рис. 20).

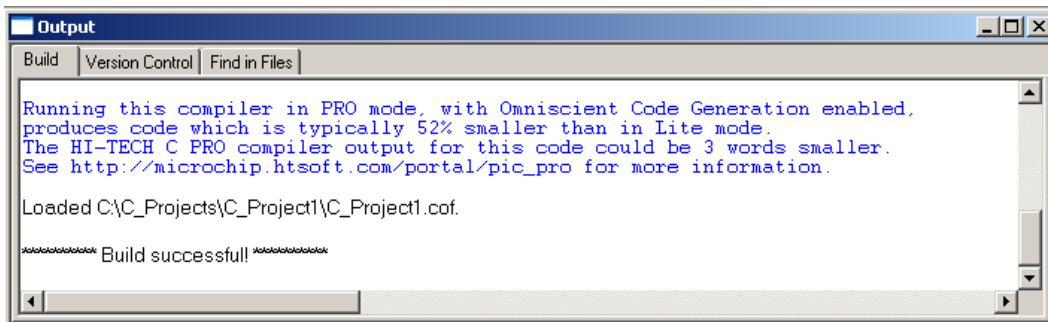


Рис. 18. Окно Output после успешной компиляции.

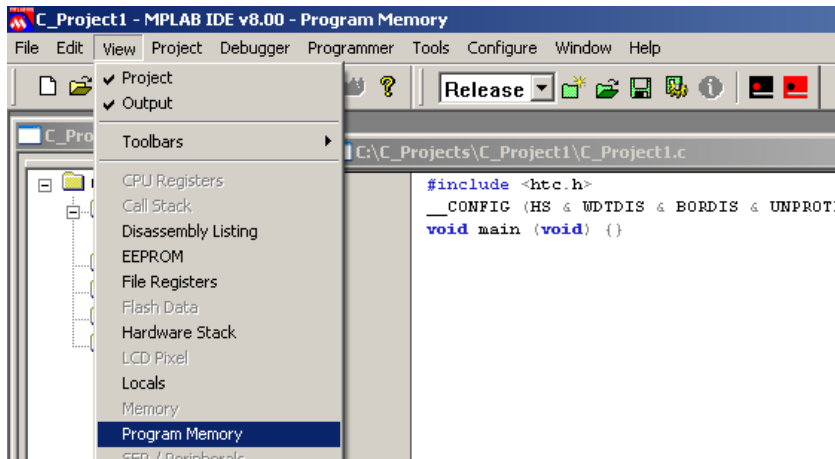


Рис. 19. Просмотр памяти команд.

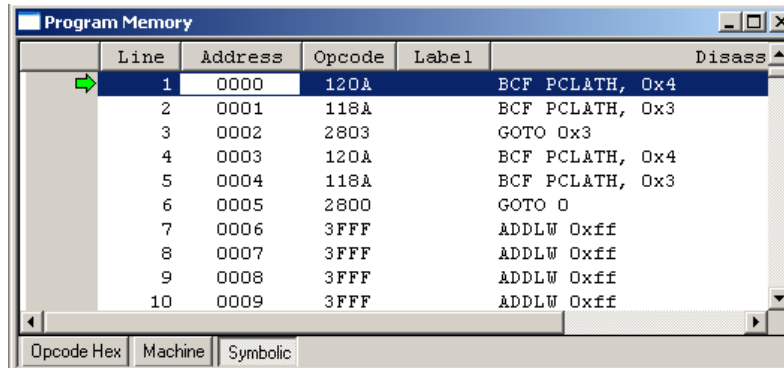


Рис. 20. Окно Program Memory.

В окне Program Memory виден машинный код, который получился в результате компиляции программы.

В папке Projects создайте папку Project1_2, в эту папку скопируйте файл Project1.c и переименуйте его в Project1_2.c; создайте проект Project1_2. Откройте файл с текстом программы и впишите цикл while(1){} внутрь функции main (между фигурными скобками). Откомпилируем (рис. 19) новую программу и получим новый машинный код в окне Program Memory (рис. 21).

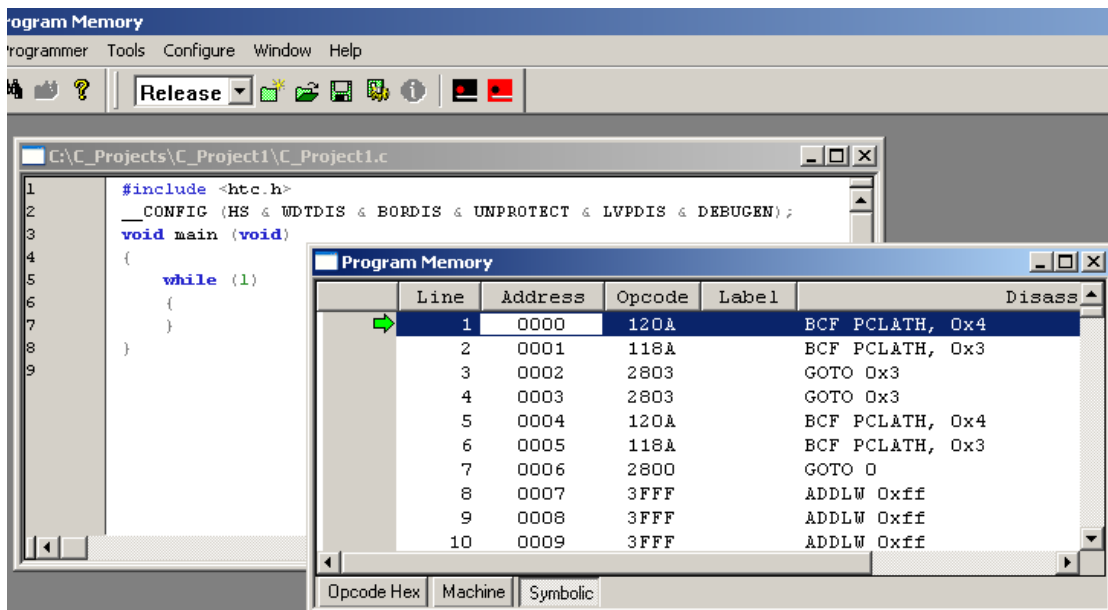


Рис. 21. Машинный код изменённой программы.

Сравните полученные файлы *.hex. Проанализируйте изменения. Попробуйте составить алгоритм полученных программ.

Аппаратное обеспечение

Эта работа выполняется на компьютере в среде разработки MPLAB IDE, макет не требуется.

Программное обеспечение

Текст файла Project1.c

```
#include <htc.h>
__CONFIG (HS & WDTDIS & BORDIS & UNPROTECT & LVPDIS & DEBUGEN);
void main (void) {}
```

Индивидуальные задания

Создайте проект с другим файлом текста программы.

В главном меню выберите View и откройте окно Program Memory. Попробуйте пояснить содержимое памяти команд.

Контрольные вопросы

1. Каким образом работа в среде MPLAB IDE облегчает труд разработчика?
2. Какие новые файлы появляются в папке после создания проекта?
3. Для чего нужны файлы *.c *.hex *.mcw *.mcp?
4. Почему при компиляции исходного текста программы в среде разработки не требуется указания места расположения компилируемого файла?
5. Что является результатом работы программиста?
6. Зачем подключают файл htc.h?
7. Что задаётся в файле pic168xa.h?
8. Какая функция должна быть обязательно описана в программе?
9. Что делают внутри функции main?
10. Опишите алгоритм программы Project1.c.
11. Опишите алгоритм программы Project1.c после ввода цикла while(1){ } в функцию main.

Оглавление:	
ЛАБОРАТОРНАЯ РАБОТА №1 «СОЗДАНИЕ ПРОЕКТА»	3
Цель работы	3
Теоретические основы	3
Задание	4
Порядок выполнения	5
Аппаратное обеспечение	13
Программное обеспечение	13
Индивидуальные задания	13
Контрольные вопросы	13
Список рисунков:	
Рис. 1. Вызов помощника создания проектов	5
Рис. 2. Приглашение помощника	5
Рис. 3. Первый шаг – выбор контроллера	5
Рис. 4. Второй шаг – указание программы-компилятора	6
Рис. 5. Выбор места, где будет создан проект	6
Рис. 6. Третий шаг – создание файла проекта	6
Рис. 7. Четвёртый шаг – нажимаем Next>	7
Рис. 8. Сведения о проекте	7
Рис. 9. Вызов окна Project	7
Рис. 10. Добавление файла в проект	8
Рис. 11. Место, в котором расположен добавляемый файл	8
Рис. 12. Выбор добавляемого файла	8
Рис. 13. Вид окна Project с присоединённым файлом	9
Рис. 14. Окно редактора с текстом программы	9
Рис. 15. Меню настроек текстового редактора	10
Рис. 16. Меню настройки редактора текстов	10
Рис. 17. Кнопка Rebuild для запуска компилятора	10
Рис. 18. Окно Output после успешной компиляции	11
Рис. 19. Просмотр памяти команд	11
Рис. 20. Окно Program Memory	11
Рис. 21. Машинный код изменённой программы	12

