



# Интегрированная среда разработки, внутрисхемной отладки программ и программирования внутренней памяти микроконтроллеров. IDE1886 версия 8.8 Руководство пользователя.

## 1. Назначение и состав.

IDE1886 - это интегрированная среда разработки, внутрисхемной отладки программ на языках «С» и ассемблер и программирования внутренней памяти микроконтроллеров семейства 1886BE. Внешний вид программы приведён на рисунке 1.

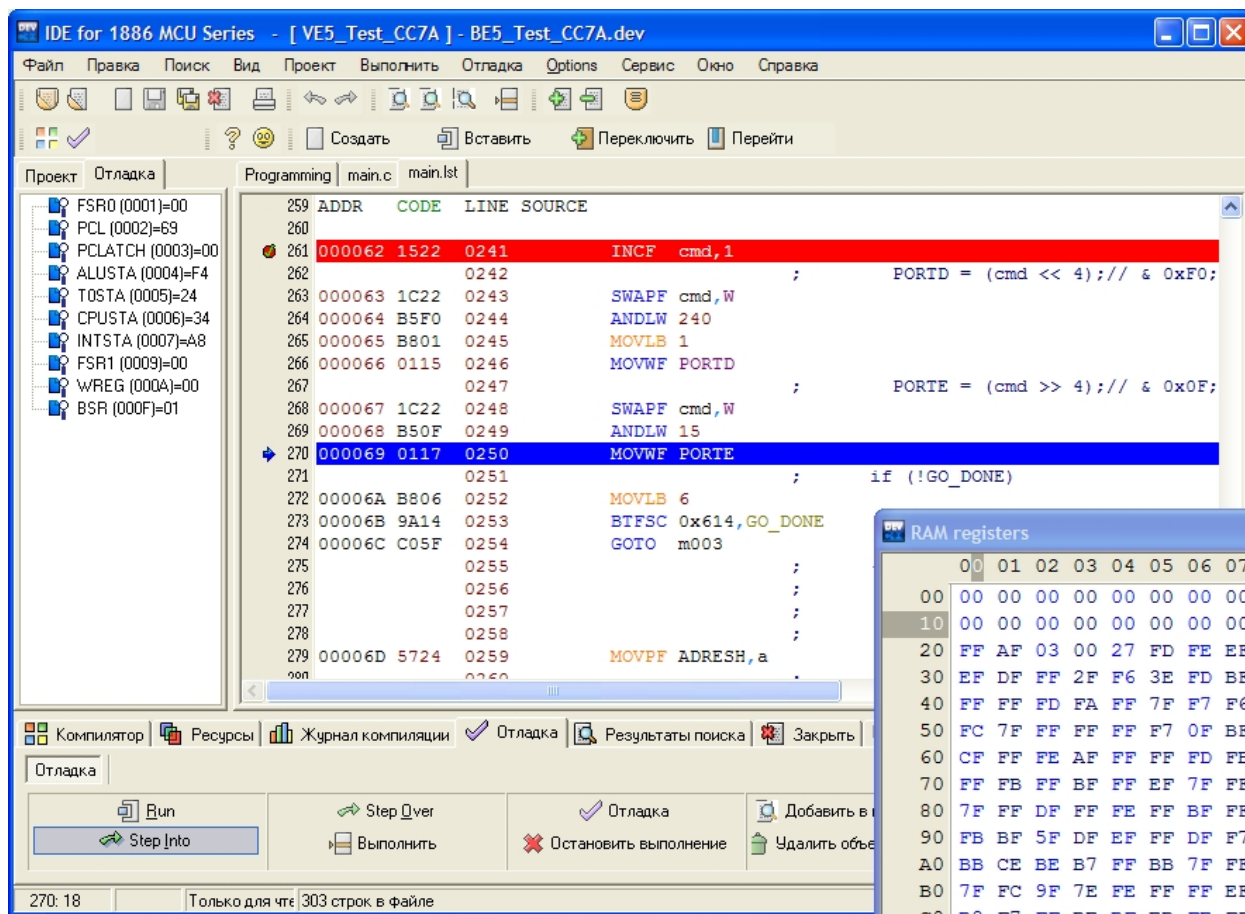


рисунок 1.

Программа поставляется на компакт-диске или с сайта фирмы АО «ПКК Миландр» <http://www.milandr.ru> и требует для работы следующие модули:

- исполняемый модуль DEVCPPEX.E;
- для компиляции программ на языке Assembler:

MPASWIN.EXE из пакета MPLAB IDE фирмы «MICROCHIP» ;

для компиляции программ на языке C: CC7A фирмы «B Knudsen Data» <http://www.bknd.com/cc7a> ;

- драйвер giveio.sys для работы с LPT-портом под ОС Windows XP и программа установки драйвера LOADDRV.EXE (При работе с USB программатором установка драйвера не требуется).

## **2. Установка программы.**

Для работы программы требуются:

- PC с процессором, аналогичным Intel Pentium;
- ОС Windows XP или старше;
- от 256 MB оперативной памяти;
- от 10 MB пространства на жестком диске.

Для установки программы необходимо скопировать все файлы с установочного диска в одну директорию, вложенность которой составляет менее 62 символов (в случае использования компилятора фирмы MICROCHIP). После этого запустите исполняемый модуль DEVCPP.EXE на выполнение.

## **3. Основные характеристики программы.**

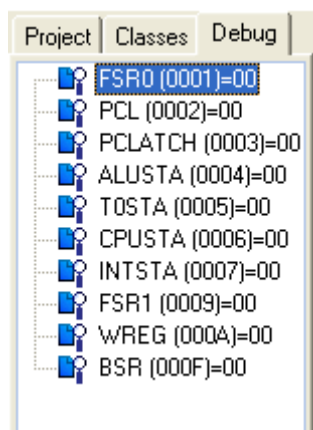
- Полноэкранный многооконный интерфейс со свободной конфигурацией окон пользователем.
- Создание, изменение, отображение исходного кода программ и ведение проектов с помощью менеджера проектов.
- Компиляция программ в двоичный машинный код компиляторами фирм MICROCHIP, B Knudsen Data.
- Цветовая подсветка ключевых слов «C» и ассемблерных команд, точек останова программы, ошибок в программе.
- Загрузка файла двоичных команд в память программ микроконтроллера.
- Ведения отладки программ по шагам, с точками останова, блочным выполнением процедур и в реальном масштабе времени.
- Ведение отладки с помощью меню и кнопок быстрого доступа.
- Отображение регистров и регистрового файла в процессе отладки.
- Программирование микроконтроллеров.
- Возможность загрузки программы из файла и сохранения в файл.

## **4. Интерфейс программы.**

Программа состоит из главного меню, панели инструментов и четырёх диалоговых окон: «**Project Manager**», «**Report Window**», «**Project Window**» и «**RAM registers**». В окне «**Project Window**» отображается отлаживаемая программа на языке «C» или ассемблер. В окне «**Report Window**» отображается информация о результатах компиляции программы и ошибках на закладках «**Compile Log**» и «**Compiler**» соответственно. Также доступны средства отладки и программирования программ на закладках «**Debug**» и «**Programming**» соответственно. С помощью окна «**RAM registers**» происходит интерактивное отображение регистрового файла в процессе отладки программы.

Все файлы проекта отображаются на закладке «**Project**» диалогового окна «**Project Manager**» что позволяет осуществлять быстрое переключение и поиск по файлам, добавлять или удалять файлы из проекта.

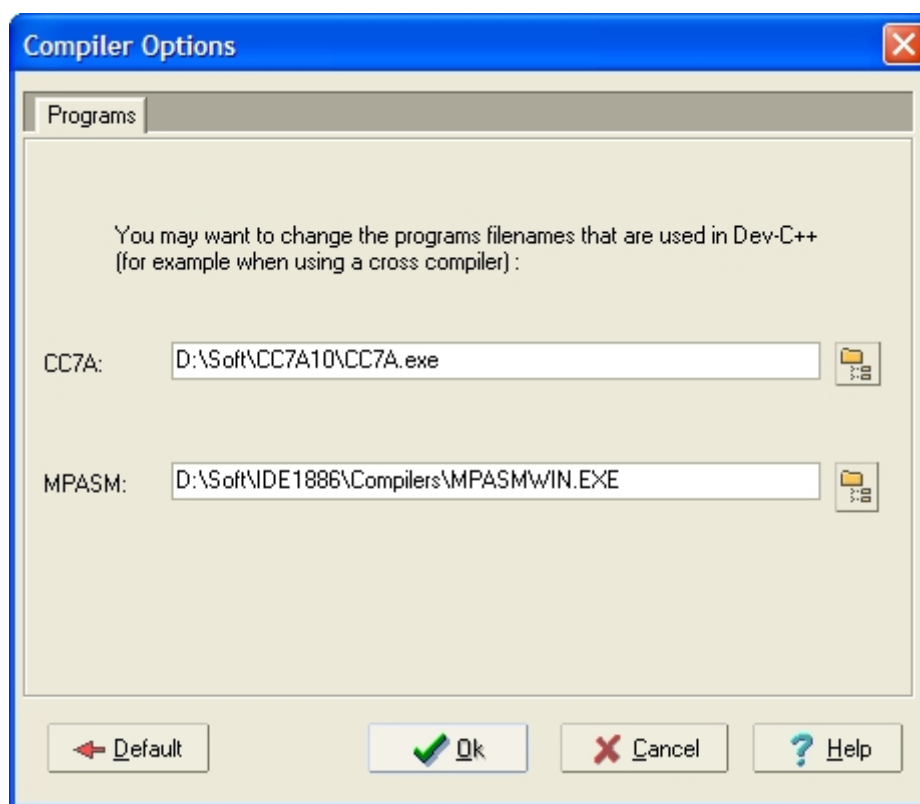
С помощью закладки «**Debug**» (рисунок 2) диалогового окна «**Project Manager**» происходит отображение регистров в процессе отладки программы. Для добавления и удаления переменных в этом окне используются кнопки «**Add watch**» и «**Remove watch**» закладки «**Debug**» диалогового окна «**Report Window**».




*рисунок 2.*

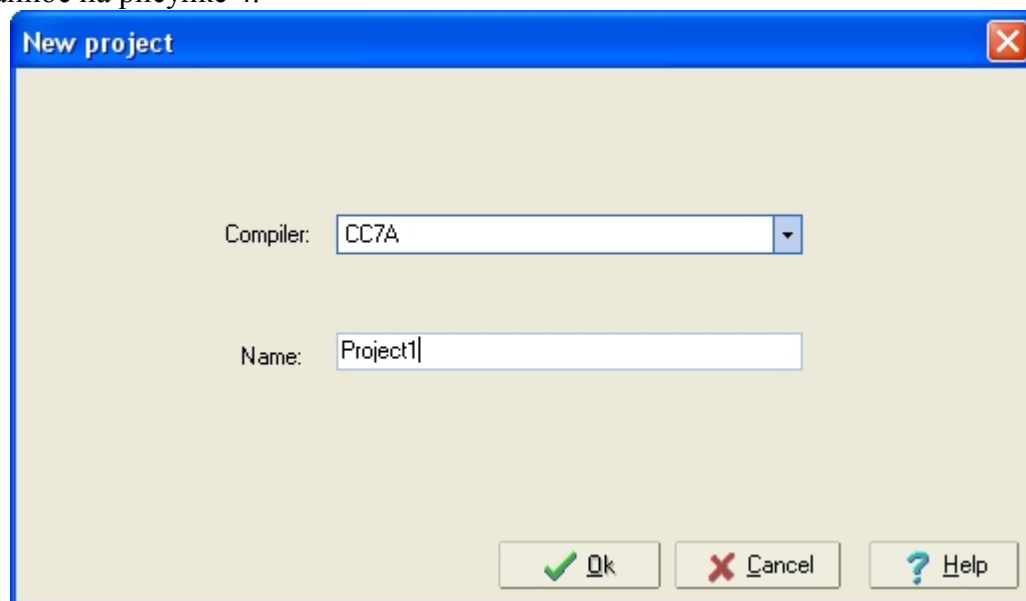
Главное меню состоит из одиннадцати пунктов «**File**», «**Edit**», «**Search**», «**View**», «**Project**», «**Execute**», «**Debug**», «**ComPort**», «**Tools**», «**Window**» и «**Help**».

Для правильной работы IDE1886 надо выбрать из главного меню пункт «**Tools->Compiler Options**» и указать пути к файлам компиляторов (рисунок 3).





*рисунок 3.*

При выборе из главного меню пункта **“File->New->Project...”** или нажатии кнопки быстрого доступа  панели инструментов вызывается диалоговое окно “New project”, показанное на рисунке 4.




*рисунок 4.*

В диалоговом окне нужно выбрать компилятор и ввести название проекта, затем нажать кнопку «ОК» и сохранить проект по выбранному пути. После этого содержимое проекта отображается в диалоговом окне **«Project Manager»**.

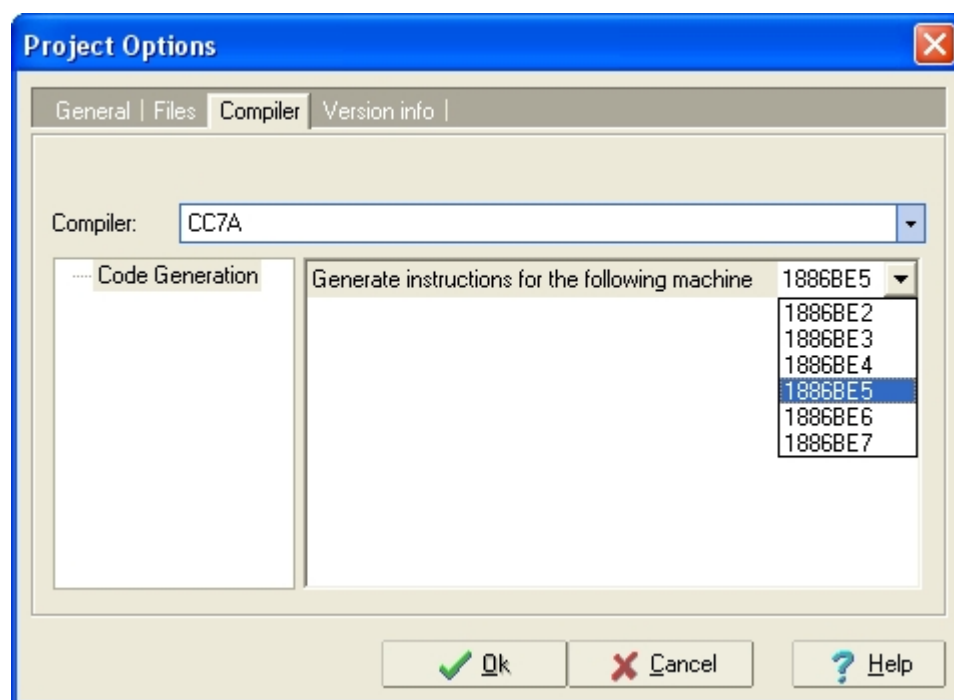
Для добавления к проекту или удаления из проекта существующих файлов необходимо выбрать из главного меню пункты **«Project->Add to Project»** или **«Project->Remove from Project»** соответственно или нажать кнопки быстрого доступа   панели инструментов.

Для проектов на ассемблере или для CC7A надо добавить только один файл, остальные файлы подключаются с помощью директив «include».

Далее необходимо установить компилятор и тип микроконтроллера. Для этого выбрать из главного меню пункт **«Project -> Project Options»** и на закладке **«Compiler»**, установить тип компилятора и тип микроконтроллера (рисунок 5).

После того как эти опции настроены правильно, можно скомпилировать проект. Для компиляции проекта необходимо выбрать из главного меню пункт **«Execute->Compile»** или нажать кнопку быстрого доступа  панели инструментов.

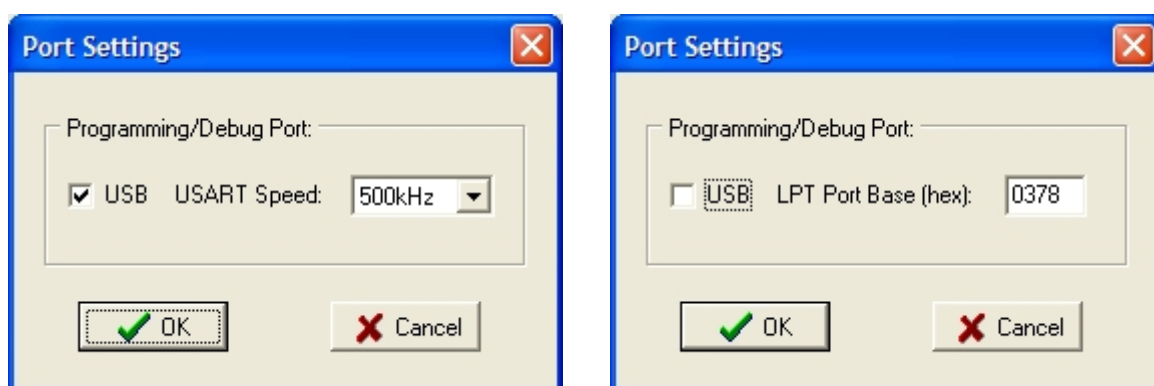
В случае отсутствия ошибок становятся доступны средства отладки в пункте главного меню **«Debug»** и на закладке **«Debug»** диалогового окна **«Report Window»**.



*рисунок 5.*

Все аппаратные настройки для LPT и USB программаторов сведены в меню «**Options**» (рисунок 6). Для USB программатора есть возможность выбора скорости обмена с программируемым контроллером. Для устойчивой работы при большой длине кабеля от программатора, возможно, потребуется выбрать меньшую скорость.

**При работе с USB программатором у пользователя должны быть административные права.**



*рисунок 6.*

### ***Замечания по режиму отладки.***

Величина тока по питанию от программатора не должна превышать определенной величины, т.е. вся периферия должна питаться от отдельного источника.

В процессорах 1886BE5 и выше появился специальный механизм, позволяющий отлаживаться через подключенный программатор. Для этого требуется запрограммировать процессор с включенной конфигурацией "Debug mode".

### **Ограничения при работе в режиме отладки для BE5-BE7:**

- Отлаживаемая программа не должна использовать USART, который занят программатором.
- Не рекомендуется ставить точку останова в подпрограмму обработки прерываний.







- Ячейки ОЗУ по адресам 0xFF во всех банках и 0xFA-0xFE в третьем банке используются отладчиком.
- Память программ по адресам 0xF90 – 0xFFF (0x790-0x7FF для BE7) занята отладчиком.
- Точка останова может быть только одна.
- **Внимание! При отладке в режиме реального времени нельзя ставить точки останова сразу после команд, могущих выполняться более одного цикла, или команд, изменяющих программный счетчик, т. к. из-за наличия конвейера в микроконтроллере это может привести к неправильному отображению данных.**

Для BE2-BE4 программа должна быть оттранслирована с подключенным модулем Debug.c.

Точки останова ставятся строкой «DebugBreakPoint;» с последующей трансляцией и перепрограммированием контроллера. Количество точек останова не ограничено.

- Отлаживаемая программа не должна использовать USART, занятый программатором.
- Ячейки ОЗУ по адресам 0x1F, 0x1E и 0xFA-0xFF в третьем банке используются отладчиком.
- Память программ по адресам 0x1FB8 – 0x1FFF занята отладчиком.

Пункты главного меню «**Debug**» и соответствующие им кнопки на закладке «**Debug**» диалогового окна «**Report Window**», предназначены для различных режимов отладки программ и включает в свой состав:

-  **Debug** – осуществляется переход на начало программы;
-  **Step Into** или «F7» – выполняется одна команда и происходит останов на команде следующей, за выполненной. Строка со следующей командой подсвечивается синим фоном.
-  **Step Over** или «F8» – выполняется блок команд вызываемый командами «call» или «lcall». Останов происходит на следующей команде. Если данный режим отладки выбирается для других команд, их выполнение происходит аналогично предыдущему режиму.
-  **Run** или «F9» – циклический запуск на выполнение команд до точки останова или до останова отладки пользователем.
-  **Run real time** или «Shift+F4» – циклический запуск на выполнение команд в реальном масштабе времени до точки останова или до останова отладки пользователем.
- **Toggle Breakpoint** или «F5» – для контроллеров BE5-BE7 установка точки останова на строке с курсором. Строка с командой подсвечивается красным.
-  **Stop Debug** или «Ctrl+F2» – остановка процесса отладки.
- **Power Off** – отключение питания микроконтроллера от программатора.

## **5. Основные режимы работы программы.**

### **5.1 Режим отладки.**

1. Создать проект и добавить в него существующие или создать новые «C(asm)» файлы.
2. Настроить опции компилятора и скомпилировать программу с помощью пункта главного меню «**Execute->Compile**». Если в процессе компиляции обнаружены ошибки, то они выводятся на закладке «**Compiler**» диалогового окна «**Report Window**». Двойным

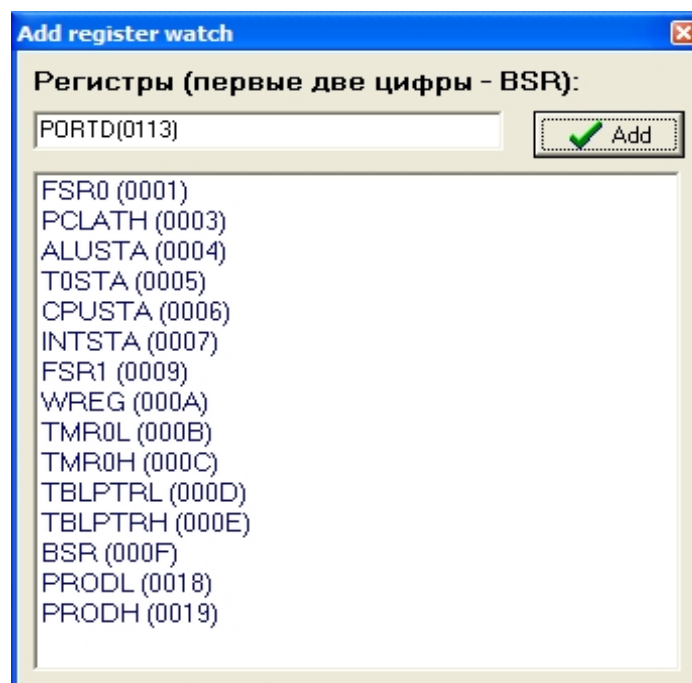


щелчком мыши на сообщении об ошибке можно перейти на место в программе, где она была обнаружена.

3. Нажать клавишу «F7» или «F8». Запустится процесс отладки. Первоначальное состояние периферийных регистров и регистрового файла отобразится в диалоговых окнах «**Debug**» и «**RAM registers**». Для перехода к этим окнам выберете из главного меню «**Debug->Watch Variables**» или «**Debug->View CPU Window**» соответственно. По мере дальнейшего нажатия клавиш «F7», содержимое диалоговых окон будет меняться в зависимости от кода выполняемой программы.

4. Для того чтобы выполнить программу до указанного места для контроллеров BE5-BE7 поставьте на строку точку останова (нажмите клавишу «F5»), а затем нажмите «F9».

5. Для того чтобы добавить новую переменную для наблюдения, в диалоговом окне «**Debug**» нажать на правую кнопку мыши, затем в падающем меню щёлкнуть на пункт «**Add Watch**», выбрать регистр и нажать кнопку «Add». Если регистра нет в списке, то ввести имя переменной, в скобках его адрес и нажать кнопку «Add» (рисунок 7).



*рисунок 7.*


6. Для того чтобы удалить регистр из списка окна «**Debug**» надо выбрать левой кнопкой мыши соответствующий регистр и нажать на правую кнопку мыши, затем в выпадающем меню щёлкнуть на пункт «**Remove watch**». Регистр будет удалён из списка.

7. Для того чтобы прекратить отладку нажать «Ctrl+F2».

8. После того как программа скомпилирована, машинный код может быть запрограммирован. Для этого надо перейти на закладку «**Programming**».

### **5.2 Режим запуска программы в реальном масштабе времени.**

Для того, чтобы запустить программу в реальном масштабе времени, остановить её в любом месте и просмотреть результаты её выполнения реализован данный режим. Для запуска программы в этом режиме выполните следующие действия:

1. После того, как программа скомпилирована без ошибок, нажмите кнопку  на панели инструментов.

2. После этого можно (для контроллеров BE5-BE7) с помощью клавиши «F5» или главного меню «**Debug->Toggle Breakpoint**» поставить точку останова программы (смотри замечания по режиму отладки выше). Для контроллеров BE2-BE4 подключите модуль Debug.c, вставьте строку «**DebugBreakPoint;**» в том месте программы, где требуется посмотреть переменные, затем запустите трансляцию и запрограммируйте контроллер.

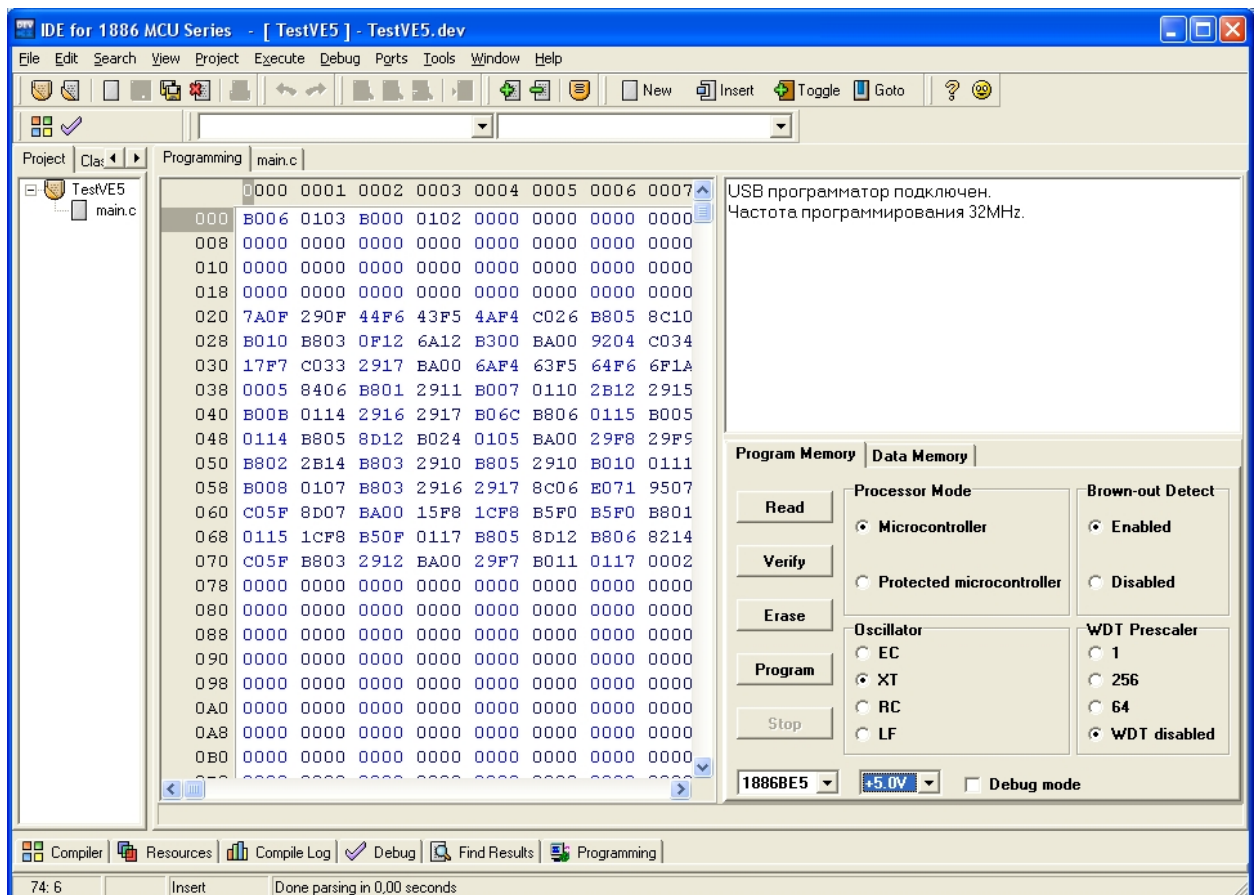
3. Затем можно нажать кнопку  **Run real time**.

4. После того как программа завершает своё выполнение на точке останова, фон следующей строки изменяет цвет на синий. В окне «**Debug**» все значения обновляются. Процесс обновления регистрового файла занимает несколько секунд. Если не требуется содержимое регистрового файла, то закройте окно «**RAM registers**».

5. Если программа не останавливается на точке останова, можно нажать «**Debug**» или «**Stop Execution**» (нажать «**Ctrl+F2**»). Будет произведен перезапуск микроконтроллера.

### **5.3 Режим работы с внутренней памятью микроконтроллера.**

В случае успешной компиляции и отладки программы, можно запрограммировать её для дальнейшего использования микроконтроллера в аппаратуре пользователя. Для работы с внутренней памятью программ микроконтроллера необходимо перейти на закладку «**Programming**» (рисунок 8).



**рисунок 8.**

Данный режим позволяет осуществлять следующие операции:

- Запись/чтение/верификацию любого объема информации в диапазоне адресов 0x0000-0x7FE0.



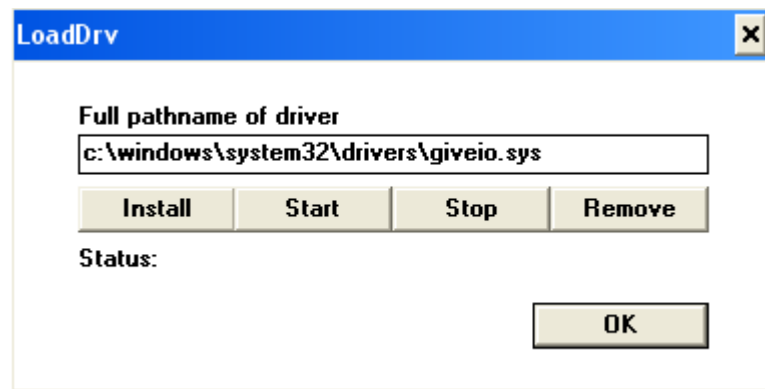
- Стирание всей памяти.
- Представление данных в шестнадцатеричном символьном формате.
- Функции редактирования в шестнадцатеричном формате.
- Возможности чтения и записи конфигурационных битов.
- Возможность выбора напряжения питания программируемой микросхемы (только для операций чтения и верификации).
- Для микроконтроллеров 1886BE5 и выше и для новых серий 1886BE3,4 доступно чтение и запись EEPROM памяти данных.

Далее приведена информация, относящаяся только к программатору, подключаемому к LPT порту.

Если при нажатии на кнопки на экран выводится сообщение «LPT port access denied», необходимо правильно установить адрес порта в меню **Options**.

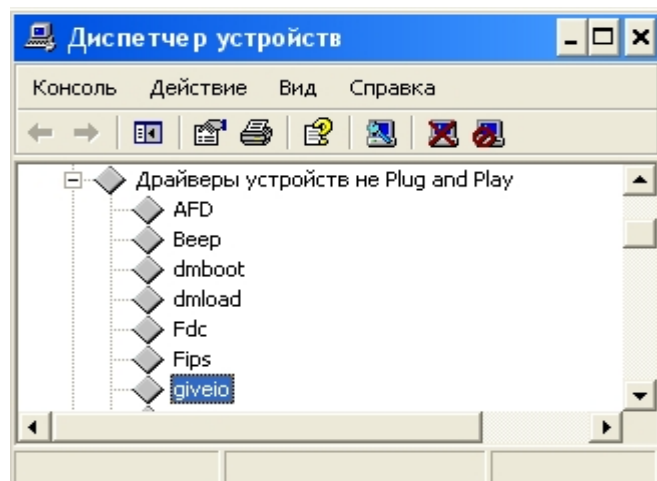
Если на этой закладке все кнопки в неактивном состоянии, необходимо установить и запустить драйвер работы с параллельным портом GIVEIO.SYS.

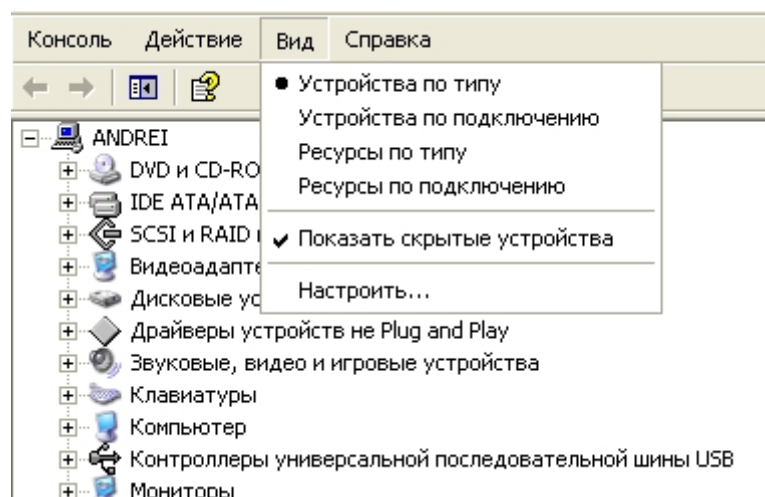
Для установки драйвера необходимо запустить программу LOADDRV.exe на выполнение (рисунок 9).



*рисунок 9.*

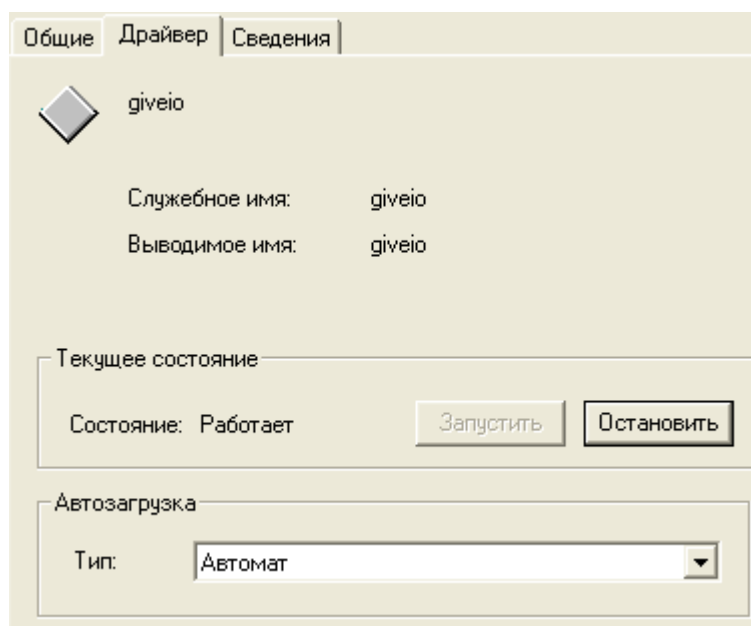
В программе необходимо указать путь к директории, где хранятся файлы драйверов ОС, и нажать кнопку Install. Затем кнопку Start. Если в строке Status выводится сообщение об ошибке, то необходимо вручную скопировать файл giveio.sys в директорию драйверов ОС. Затем снова нажать кнопку Start. После этого компьютер необходимо перезагрузить. После перезагрузки зайти в Панель управления -> Система-> Диспетчер устройств. Затем выбрать Вид -> Показать скрытые устройства. Внешний вид этой панели представлен на рисунке 10.





**рисунок 10.**

Затем выбрать свойства драйвера giveio.sys и перейти на закладку Драйвер. Тип Автозагрузки должен быть «Автомат». Текущее состояние «Работает». Внешний вид окна приведён на рисунке 11.



**рисунок 11.**

### **5.3.1 Режим программирования.**

Для того чтобы запрограммировать микроконтроллер, необходимо на закладке «Programming» загрузить файл (bin или hex) или скомпилировать программу без ошибок, правильно установить конфигурационные слова и нажать кнопку “PROGRAM” (контроллер перед этим должен быть стерт). Выводится подтверждающее сообщение, а в нижней части шкала времени сигнализирует о текущем статусе процесса. В случае неудачного завершения процесса выводится соответствующее сообщение и адрес, по которому не прошла запись. Производится также запись конфигурационных слов. Следует обратить внимание, что в новых микроконтроллерах прерывание процесса записи при установленной конфигурации «Microprocessor» (контроллер полностью стерт) устанавливается защищенный режим.

### **5.3.2 Режим верификации.**

При нажатии кнопки “VERIFY” происходит запуск процесса сравнения содержимого окна “Memory Map” с содержимым внутренней памяти микроконтроллера. Выводится подтверждающее сообщение, а в нижней части шкала времени сигнализирует о текущем статусе процесса. При несовпадении выводится сообщение об ошибке и адрес, по которому не прошла верификация.

Производится также сравнение конфигурационных слов.

### **5.3.3 Режим стирания.**

Для стирания всей внутренней памяти микроконтроллера необходимо нажать кнопку “ERASE”. Выводится подтверждающее сообщение. В случае неудачного завершения выводится сообщение об ошибке и номер нестертого сектора.

### **5.3.4 Режим чтения.**

Для чтения содержимого внутренней памяти необходимо нажать кнопку “READ”. Содержимое памяти будет отображаться в окне “Memory Map”. Производится также чтение конфигурационных слов. Считанное содержимое памяти программ и памяти данных можно сохранить в двоичный файл.

## **6. Лицензионное соглашение.**

Внимательно прочтите лицензионное соглашение в файле license.txt. Вопросы по работе с программой вы можете задать на форуме по адресу: <http://forum.milandr.ru/viewforum.php?f=17>