

Роман Абраш
г. Новочеркасск
E-mail: arv@radioliga.com

Книга по работе с WinAVR и AVR Studio



Продолжение.
Начало в №1-3/2010

WinAVR + AVR Studio

Пакет WinAVR включает собственный редактор (Programmers Notepad) для написания исходных текстов программ и ряд утилит, облегчающих этот процесс. Однако значительно более удобным способом следует признать интеграцию WinAVR в бесплатную среду разработки программ для микроконтроллеров AVR фирмы Atmel – AVR Studio, самую свежую версию которой всегда можно скачать с сайта Atmel.

Не смотря на все достоинства утилит, входящих в WinAVR, они имеют один весьма существенный недостаток: не поддерживают отладку непосредственно в IDE, и слабо интегрируются с аппаратными средствами разработки и отладки, называемых Starter Kit (стартовые комплекты). AVR Studio прекрасно работает со всеми аппаратными средствами Atmel, но поставляется лишь с поддержкой ассемблера. Объединение достоинств обоих пакетов позволяет достичь желаемого результата: разработка и интерактивная отладка с поддержкой всех средств Atmel на языке высокого уровня Си.

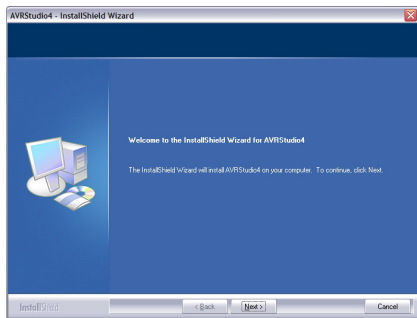
Инсталляция

Процесс инсталляции хотя и прост, но сопровождается большим количеством англоязычных сообщений, что вызывает затруднения у русскоязычных пользователей. По этой причине далее при всяком удобном случае будут приведены скриншоты окон, выводимых программами, с соответствующими пояснениями.

Установка AVR Studio

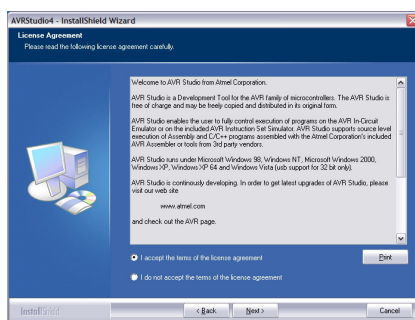
Установка AVR Studio осуществляется запуском соответствующего программного файла-инсталлятора¹⁶ – файл **AVRStudio4.18SP2.exe** [2], который в режиме мастера (Wizard) выполнит все необходимые действия, пользователю остается лишь внимательно следить за выводимыми сообщениями и давать необходимые ответы на задаваемые вопросы.

Сразу после запуска инсталлятора вы видите окно следующего вида:

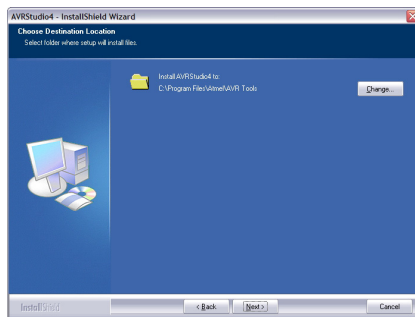


Это просто экран приветствия, следует нажать кнопку «Next» для продолжения.

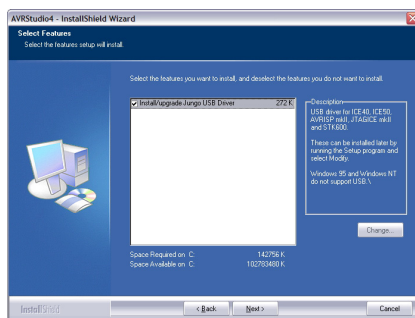
Следующий экран такой:



Традиционное лицензионное соглашение – необходимо отметить вариант «I accept...» (я принимаю соглашение) и нажать «Next» для продолжения.

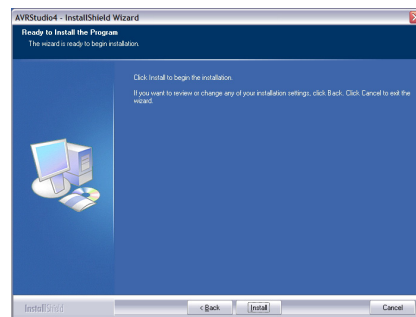


Теперь предлагается выбрать папку для установки программы. Рекомендуется избегать папок, в именах которых встречаются русские символы. Папка по умолчанию, как правило, удовлетворяет этим требованиям. Далее снова следует нажать «Next».

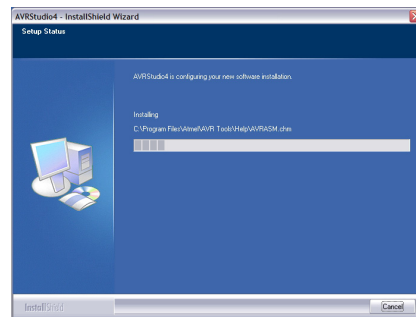


На этом экране приводятся сведения об элементах, установка которых не обязательна. В приведенном примере предлагается установить USB драйвер для аппаратных средств отладки и программирования микроконтроллеров ICE40, ICE50, AVRISP mkII, JTAGICE и STK600. Если вы не имеете указанных аппаратных средств – можно отказаться от установки драйвера, тем более что при необходимости всегда можно «доустановить» его. Для некоторых опциональных компонентов может активироваться кнопка «Change», при помощи которой можно изменить состав выбранной компоненты (в рассматриваемой версии таких компонентов нет).

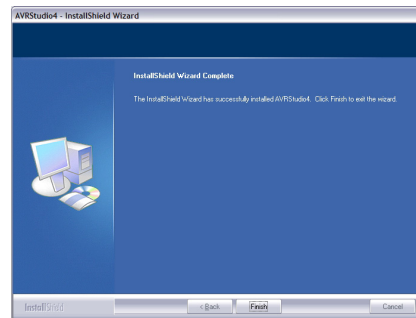
Следующий экран уведомляет, что все готово для начала инсталляции, которая начнется после нажатия кнопки «Next»:



После нажатия «Next» начинается собственно установка, т.е. распаковка и копирование файлов IDE AVR Studio в выбранную папку. Процесс довольно долгий, сопровождается «индикацией»:



Если решено отменить инсталляцию – можно нажать кнопку «Cancel», однако работоспособность программы не будет обеспечена. После завершения процесса выводится последнее окно следующего вида:



Здесь сообщается об успешной установке AVR Studio, следует нажать кнопку «Finish» для завершения работы мастера установки. После этого в меню «Пуск» Windows появится пункт для запуска AVR Studio – программа установлена, и можно начинать работу с ней.

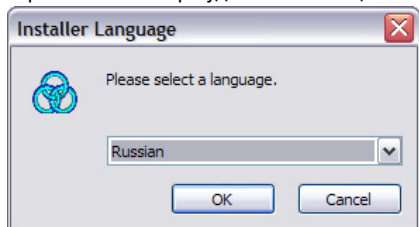
Но пока что возможности работы с Си нет – AVR Studio поставляется только с двумя версиями компиляторов ассемблера. Если у вас есть желание освоить Си (а оно должно быть, если вы читаете эту книгу), следует приступить к инсталляции WinAVR.

¹⁶ На момент верстки статьи последняя версия была датирована февралем 2010 г.

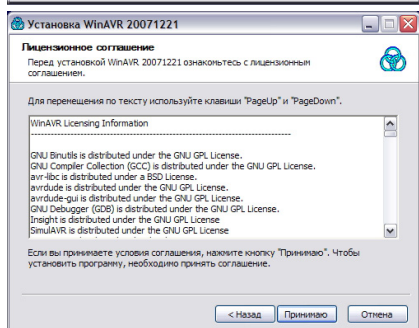
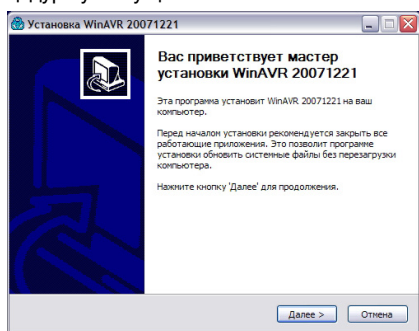
Установка WinAVR

Инсталлятор WinAVR для Windows последней (на момент верстки статьи) версии можно скачать по ссылке [3]. Файл инсталлятора имеет имя вида **WinAVR-yyyymmdd-install.exe**, где **yyyy** – год, **mm** – месяц и **dd** – день релиза.

Инсталляция происходит немного приятнее, так как инсталлятор русифицирован. В начале предлагается выбрать язык, на котором инсталлятор будет с вами общаться:

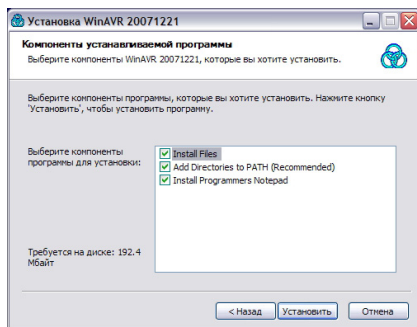
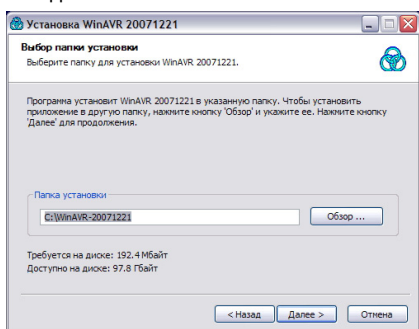


Рекомендуется выбрать «**Russian**» - русский (если нет иных предпочтений, разумеется) и нажать «**OK**». Дальнейшая процедура уже существенно более понятна:



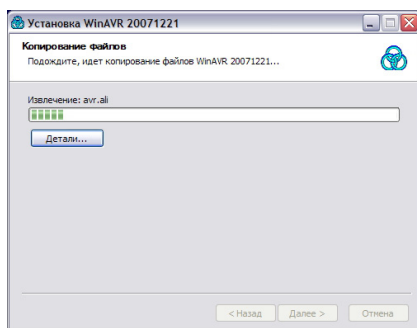
После окна приветствия предлагается согласиться с условиями лицензии – следует согласиться, разумеется.

Затем предлагается указать папку для инсталляции. Обязательно следует выбрать папку, не содержащую в своем пути русских символов и пробелов¹⁷! Рекомендуется сократить предлагаемое по умолчанию имя до «**C:\WinAVR**».



В следующем окне предлагается выбрать компоненты для установки. Единственное, без чего вы вполне можете обойтись – это редактор **Programmers Notepad**, остальные опции рекомендуется оставить включенными.

После нажатия кнопки «**Установить**» начинается весьма долгая процедура инсталляции, сопровождаемая индикатором:



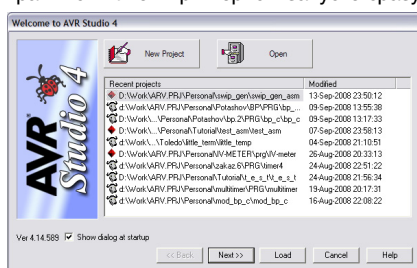
Вариантов нет – следует дождаться завершения процесса инсталляции.

Завершится инсталляция нажатием в появившемся окне кнопки «**Готово**». Сразу после этого откроется файл со сведениями о релизе – он на английском, можно смело его закрывать.

На этом подготовительные операции следует считать завершенными – теперь все готово для работы.

Работа Мастер проектов

Запустите только что установленную AVR Studio, и на вашем дисплее появится интегрированная среда разработки программ от Atmel. При первом запуске сразу



же появится окно мастера проектов:

На рисунке показан вид мастера с некоторым числом проектов, с которыми уже велась работа – в вашем случае при

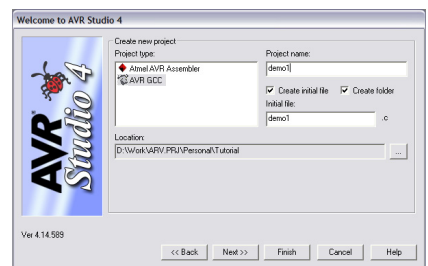
первом запуске список проектов будет, конечно, пуст. Кнопка «**New Project**» в верхней части позволит создать новый проект, а расположенная рядом кнопка «**Open**» – открыть любой имеющийся.

Если список предыдущих проектов (**Recent projects**) не пуст – можно выбрать любой из них и продолжить работу над ним, нажав кнопку «**Load**» (загрузить). Кнопка «**Next >>**» позволит изменить параметры выбранного проекта.

Но в первый раз список предыдущих проектов будет пуст, и ничего иного, кроме как создать новый проект или открыть существующий, сделать не получится, разве что завершить работу мастера проектов кнопкой «**Cancel**».

Мастер проектов автоматически стартует при каждом запуске AVR Studio. Если такое его поведение вас не устраивает – деактивируйте опцию «**Show dialog at startup**» (показать при запуске) – в следующий раз мастер не запустится автоматически.

Нажмем «**New Project**», чтобы пройти все этапы создания проекта с нуля.



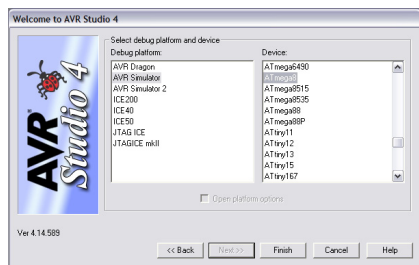
На этом этапе мастер проектов предлагает выбрать тип проекта (**Project type**) - будет ли это программа на ассемблере (**Atmel AVR Assembler**), или же на Си (**AVR GCC**)¹⁸. Проекты на ассемблере нас не интересуют, поэтому работу мастера в этом режиме мы рассматривать не станем, а сразу выберем AVR GCC. Кроме этого следует указать имя проекта (**Project name**) – это будет особый файл, в котором будут сохранены все последующие настройки нашего проекта. Если отмечены опции «**Create initial file**» (создать файл главного модуля) и «**Create folder**» (создать папку проекта) – по мере ввода имени проекта автоматически будут заполняться поля «**Initial file**» (файл главного модуля) и «**Location**» (местоположение). Расширения для файлов указывать не надо – они присваиваются автоматически.

Разумеется, назначенные мастером имена файлов и папок вы всегда можете изменить вручную. Рекомендуется все же придерживаться следующего правила: имя папки для проекта должно совпадать с именем самого проекта так будет меньше путаницы в дальнейшей работе. Так же рекомендуется всегда размещать каждый новый проект в отдельной папке, а для всех проектов вообще выделить другую, «корневую» папку.

¹⁷ В последних версиях WinAVR жесткость этого требования уменьшена – пробелы допустимы, хотя если есть намерение пользоваться утилитами комплекта WinAVR для командной строки – следует его соблюдать.

¹⁸ Как уже было сказано, WinAVR – это всего лишь «сборка» утилит для разработки программ на Си, в которую включен мультиплатформенный компилятор AVR GCC.

После того, как имя проекта задано – активируется кнопка «Next >>», позволяющая перейти к следующей странице мастера:



Здесь предлагается выбрать отладочную платформу¹⁹ и используемый микроконтроллер (**Select debug platform and device**). В окне слева перечислены все доступные платформы для отладки (**Debug platform**), все, кроме тех, что содержат в своем названии слово «**Simulator**», требуют наличия аппаратных средств от Atmel. Если вы имеете одно из них – следует выбрать из списка его, а если нет – придется довольствоваться эмуляцией отладки, для чего следует выбрать **AVR Simulator**.

Для некоторых платформ могут появляться дополнительные опции для их настройки. В этом случае активируется опция «**Open platform option**» (открыть настройки платформы), при активации которой вам будет предложено настроить специфические для выбранной платформы параметры после завершения работы мастера. Для каждой платформы эти параметры могут быть различны, и об этом следует читать в сопроводительной документации.

В зависимости от выбранной платформы меняется содержимое окна выбора микроконтроллера (**Device**) – те модели, которые не поддерживаются выбранной системой, становятся недоступными для выбора (затеняются). Эмулятор (**AVR Simulator**)²⁰ поддерживает практически все существующие микроконтроллеры, поэтому для начала следует пользоваться им.

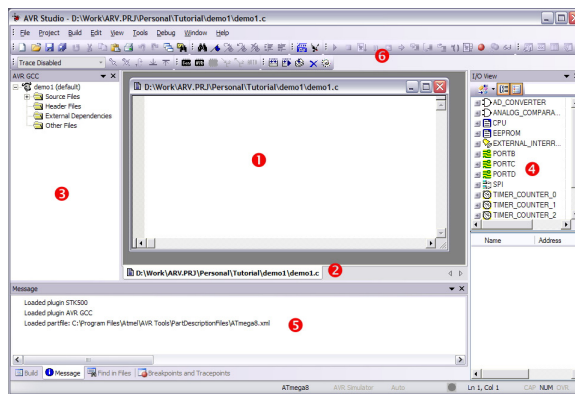
Выбрав платформу и модель микроконтроллера, вы можете завершить работу мастера нажатием кнопки «**Finish**», в результате чего будут сгенерированы и открыты необходимые файлы. На рисунке показан выбор модели **Atmega8** – все дальнейшее рассмотрение среды AVR Studio будет сопровождаться скриншотами именно для этого случая (для других моделей могут быть незначительные отличия, связанные с различием в архитектуре микроконтроллеров).

Рабочее пространство

Когда работа мастера проектов завершена, внешний вид рабочего пространства AVR Studio становится таким:

¹⁹ Об отладке будет сказано в следующих главах.

²⁰ Более «продвинутый» AVR Simulator 2 по заявлению Atmel реализует новые (более адекватные реальности) алгоритмы моделирования периферии микроконтроллеров, однако на момент написания книги поддерживал очень ограниченный ассортимент микроконтроллеров, что не позволяет рекомендовать его как основное средство эмуляции.



Среда разработки – многооконная, т.е. все доступное пространство разделено на несколько областей-окон изменяемых размеров, каждая из которых предназначена для просмотра различной информации (на рисунке помечены номерами). В процессе работы число и расположение этих областей может изменяться в зависимости от текущего режима IDE, так же можно самостоятельно настроить рабочее пространство по своему усмотрению. В данной главе представлен краткий обзор всех областей и общих для всех особенностей управления рабочим пространством, далее каждая область будет рассмотрена отдельно более подробно.

Центральная область ① содержит окно редактирования исходного текста. В этой области каждый файл может либо быть в отдельном окне (как на рисунке), либо может заполнять всю область целиком (если нажать кнопку «Развернуть»). В нижней части ② центральной области размещаются закладки-ярлычки для быстрого переключения между открытыми файлами.

Область ③ – это окно проекта, в котором в виде древовидной структуры показано содержимое всего проекта. Корнем дерева является файл проекта «demo1», ветви дерева – это группы файлов, составляющих проект:

- **Source Files** – исходные тексты модулей программы. Сюда входят все файлы с исходными текстами модулей, составляющих проект.
- **Header Files** – заголовочные файлы, подключаемые пользователем.
- **External Dependencies** – внешние зависимости, т.е. файлы, от которых зависят прочие файлы проекта. В этой группе обычно автоматически размещаются заголовочные файлы, подключенные внутри других заголовочных файлов, т.е. неявно задействованные в проекте.
- **Other Files** – прочие файлы. В этой группе помещаются файлы, генерируемые компилятором в процессе работы, т.е. выходные файлы, а так же некоторые вспомогательные файлы, например, справочные сведения или документация.

Область ④ – окно периферийных устройств микроконтроллера. Оно имеет название «**I/O View**» - просмотр ввода-вывода, т.к. любое периферийное устройство для программиста представлено в виде набора портов ввода-вывода (или регистров

ввода-вывода). В этом окне в виде раскрывающихся списков перечислены все аппаратные устройства выбранного микроконтроллера – таймеры, порты, АЦП и т.п. Во время написания исходного текста программы это окно поможет ориентироваться в назначении битов тех или иных регистров управления периферией, но гораздо более важную роль оно играет при отладке.

Окно ⑤ – это многофункциональное окно вспомогательной информации. По умолчанию оно включает 4 закладки, каждая из которых отвечает за свои сведения:

- **Build** – информация о результатах работы компилятора, здесь выводятся сообщения о ходе компиляции программы.
- **Message** – сообщения. В этой области можно видеть все сообщения IDE: от сведений о загрузке модулей программы до сообщений об ошибках компилятора.
- **Find in Files** – область результатов поиска по файлам.

• **Breakpoints and Tracepoints** – точки останова и трассировки. В этой области перечислены все точки останова исполнения и трассировки программы, которые программист установил для отладки.

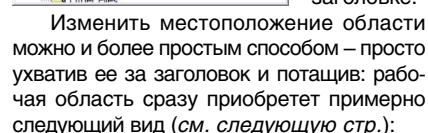
Наконец, область ⑥ – это традиционная область панелей кнопок, которые облегчают выполнение многих команд. Выше находится строка меню.

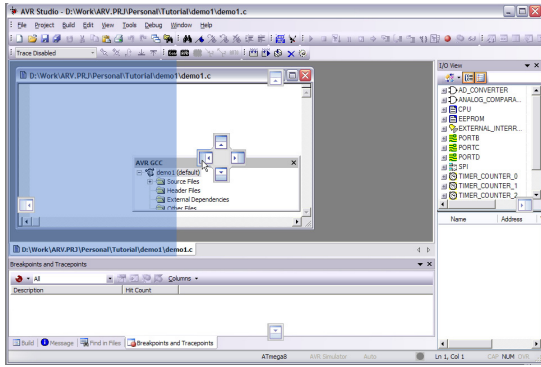
Все рассмотренные области, кроме ①, могут менять свое расположение, размеры и даже отображаться в виде плавающих отдельных окошек. Внешний вид и расположение областей можно изменить при помощи вспомогательного меню, которое открывается при нажатии на кнопку с треугольником в заголовке области (см. рисунок).

Это вспомогательное меню содержит следующие пункты:

- **Floating** – сделать область «плавающей»
- **Docking** – «прилепить» область к одному из краев рабочего пространства
- **Tabbed Document** – оформить область в виде закладки другой области
- **Auto Hide** – прятать автоматически, т.е. в режимах, когда область не требуется, она будет автоматически убрана с экрана
- **Hide** – спрятать, т.е. убрать область с экрана. Эта команда равносильна нажатию на кнопку с крестиком в заголовке.

Изменить местоположение области можно и более простым способом – просто ухватив ее за заголовок и потащив: рабочая область сразу приобретет примерный следующий вид (см. следующую стр.):





На рисунке показан момент, когда область проекта «перетаскивается» в центр рабочего пространства, причем курсор попадает в крестообразную область-«маркер» (которая появляется в момент начала перетаскивания). Кроме крестовидного маркера в центре, появляются еще 4 квадратных маркера по краям рабочего пространства. Все это должно способствовать по замыслу разработчиков AVR Studio упрощению процесса размещения перетаскиваемой области в нужном месте. Когда при перетаскивании области курсор попадает в один из этих появившихся «маркеров», на рабочем пространстве голубым цветом подсвечивается зона, в которую «прилипнет» перетаскиваемое окно. К сожалению, описать словами этот процесс непросто, гораздо интереснее попробовать потаскать окошки самостоятельно, наблюдая за возникающими эффектами и достигаемыми результатами.

В процессе перетаскивания окон обратите внимание на подсвечиваемые голубым области – они могут показывать и вариант, когда окно станет новой закладкой в уже имеющемся окне. Если бросить окно вне одного из «маркеров» – окно останется плавающим. Если область «закрыть» – ничего страшного, при помощи команд главного меню любую область всегда можно снова вывести на дисплей (см. далее).

Главное меню

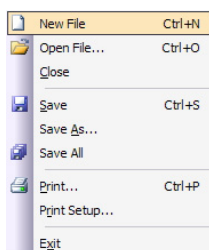
Рассмотрим кратко содержание главного меню AVR Studio, останавливаясь более подробно на заслуживающих того пунктах. Часть пунктов меню более подробно будут раскрыты далее, при описании соответствующих режимов работы.

Для многих команд в меню назначены вспомогательные изображения. Эти же изображения могут находиться и на кнопках, размещенных в области кнопок, т.е. легко сориентироваться в командах: одинаковые изображения – одинаковые команды. В дальнейшем отдельного описания кнопок не будет.

File - Файл

Традиционное меню, содержащие традиционные команды:

- **New File** – Новый файл. Создает новый пустой текстовый документ



для редактирования. Тип файла задается при сохранении.

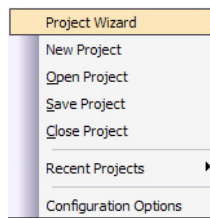
- **Open File** – Открыть файл для редактирования.
- **Close** – закрыть текущий редактируемый файл.
- **Save** – сохранить текущий редактируемый файл
- **Save As** – сохранить файл под другим именем
- **Save All** – сохранить все открытые файлы
- **Print** – напечатать текущий редактируемый файл
- **Print Setup** – настроить режим печати

- **Exit** – завершение работы с AVR Studio

Project - управление проектами

Если вы отказались от автоматического запуска при старте мастера проектов – это меню поможет вам выполнить все необходимые операции, как с мастером, так и без него.

- **Project Wizard** – мастер проекта. Команда запускает рассмотренный ранее мастер проектов.



- **Open Project** – открыть проект, ранее сохраненный на диске.

- **Save Project** – сохранить проект. Обратите внимание, что эта команда не сохраняет редактируемые файлы, составляющие проект – она сохраняет различные настройки проекта.

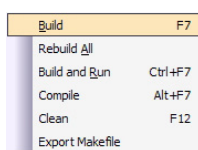
- **Close Project** – закрыть текущий проект. Работа возможна лишь с единственным проектом, поэтому перед тем, как начать новый, следует закрыть текущий. Обычно автоматическое закрытие проекта происходит и без вызова этой команды – при открытии или создании нового.

- **Recent Projects** – предыдущие проекты. Эта команда открывает список нескольких проектов, с которыми вы работали последнее время, позволяя вернуться к одному из них. По действию аналогична выбору предыдущего проекта на первой странице мастера проектов, но «помнит» не более 10 проектов (в порядке увеличения даты).

- **Configuration Options** – параметры конфигурации проекта. Важная команда, позволяющая произвести тонкую настройку режима компиляции. Подробно рассматривается далее.

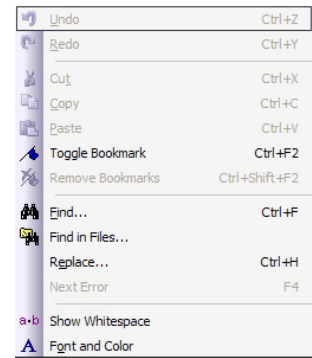
Build – сборка проекта

Подробно рассматривается далее.



Edit – правка

Меню содержит ряд традиционных команд, а также несколько дополнительных:



- **Undo** – стандартные команды вырезать, копировать и вставить.

- **Toggle Bookmark** – закладка. Команда позволяет сделать закладку на строке, в которой сейчас находится курсор. Подробно работа с закладками рассматривается в главе «Редактирование исходных текстов».

- **Remove Bookmarks** – удалить закладки.

- **Find** – найти. Выполняет поиск текста в текущем редактируемом файле.

- **Find in Files** – найти в файлах. Выполняет поиск текста по нескольким файлам, помещая результаты поиска в отдельное окно. Дополнительно о поиске упоминается в главе «Редактирование исходных текстов».

- **Replace** – найти и заменить. Выполняет автоматический поиск и замену.

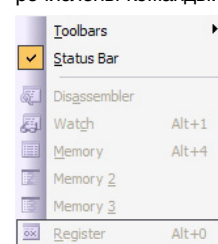
- **Next Error** – перемещает курсор в строку, содержащую очередную ошибку.

- **Show Whitespace** – показать пробелы. Команда позволяет обозначить точкой все пробелы в тексте программы.

- **Font and Color** – команда позволяет настроить внешний вид редактора текста. Подробно рассмотрена в главе «Подсветка синтаксиса».

View – вид

Управляет видимостью вспомогательных областей-окон рабочего пространства. Большинство этих областей необходимы при отладке программ, и рассматриваются подробно далее, здесь же только кратко перечислены команды:



- **Toolbars** – инструментальные панели. Открывает дополнительное меню (см. далее), позволяющее оперативно управлять видимостью различных панелей кнопок и т.п.

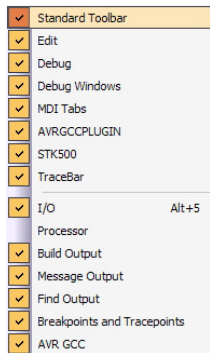
- **Status Bar** – панель состояния. Команда управляет видимостью традиционного элемента окон программ Windows – нижняя строка состояния.

- **Disassembler** – окно дизассемблированного кода программы.

- **Watch** – окно просмотра переменных

- **Memory, Memory 2 и Memory 3** – три окна просмотра содержимого памяти микроконтроллера

- **Register** – окно просмотра содержимого регистров микроконтроллера.



Под меню **Toolbars** содержит следующие опции:

- **Standard Toolbar** – панель стандартных кнопок
- **Edit** – панель кнопок, соответствующих командам меню Edit

- **Debug** – панель кнопок, соответствующая командам меню Debug

- **Debug Windows** – окна с отладочной информацией

- **MIDI Tabs** – закладки многодокументного интерфейса. Если эта опция отмечена, то в области редактирования файлов будут присутствовать закладки быстрого переключения между файлами. Если опция не активна – закладки не будут видны.

- **AVRGCCPLUGIN** и **AVR GCC** – опции, управляющие видимостью панелей и окон, внедренных в AVR Studio модулями WinAVR. Настоятельно рекомендуется не отключать эти опции.

- **STK500** – управление панелями аппаратного комплекса отладки STK500. Эта опция может отсутствовать, если STK500 не поддерживается.

- **TraceBar** – панель трассировки

- **I/O** – панель периферии

- **Processor** – панель состояния ядра процессора

- **Build Output** – панель (закладка) вывода результатов компиляции

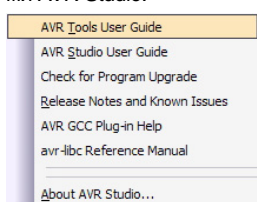
- **Message Output** – панель (закладка) вывода сообщений

- **Find Output** – панель (закладка) вывода результатов поиска текста в файлах

- **Breakpoints and Tracepoints** – панель (закладка) со списком точек остановки или трассировки

Tools – инструменты

Содержит ряд команд для работы с различными дополнительными утилитами или аппаратными средствами, поддерживаемыми AVR Studio.



Большинство команд этого меню предназначено для работы с аппаратными средствами программирования и отладки микроконтроллеров, и, в случае отсутствия этих самых аппаратных средств, не функционируют.

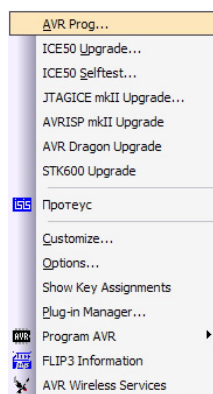
Реальный интерес представляют только следующие команды:

- **Customize** – настройка по своему вкусу внешнего вида IDE, подробно рассматривается далее.

- **Options** – параметры. Команда позволяет настроить параметры работы AVR Studio. Подробно рассматривается далее.

- **Show Key Assignments** – настройка «горячих клавиш». Большинству команд меню уже назначены горячие клавиши, что хорошо видно на соответствующих рисунках. Вы можете самостоятельно настроить соответствие клавиш так, как вам удобно.

- **Plug-in Manager** – менеджер плагинов. Плагин – это дополнительный модуль, расширяющий функциональность AVR Studio. Обычно плагины поставляются сторонними разработчиками аппаратного обеспечения и т.п. Например, поддержка WinAVR реализуется при помощи соответствующего плагина. Данная команда позволяет подключить или отключить имеющиеся плагины, изменив тем самым функционирование IDE.



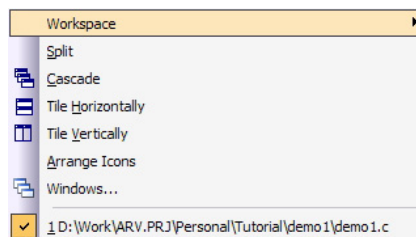
- **Program AVR** – программирование микроконтроллера AVR. Команда позволяет, не выходя из AVR Studio, выполнить программирование, т.е. «прошивку» микроконтроллера результатами компиляции программы. Для этого необходимо лишь наличие программатора, поддерживаемого AVR Studio.

Debug – отладка

Отладка в AVR Studio будет рассмотрена более детально в следующих главах, а так же, по мере возможности, при рассмотрении примеров.

Windows – окна

Стандартное меню для всех Windows-программ.



- **Workspace** – открывает дополнительное меню управления рабочим пространством (см. далее)

- **Split** – разделить. Позволяет разделить текущее окно редактирования файла на 2 или 4 части, чтобы иметь возможность одновременно наблюдать участки текста, находящиеся на большом удалении друг от друга (например, начало и конец большого файла). Режим актуален для работы с дисплеями высокого разрешения и большого размера.

- **Cascade** – расположить окна каскадом

- **Tile Horizontally** и **Tile Vertically** – расположить окна мозаикой по горизонтали или вертикали

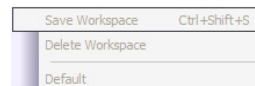
- **Arrange Icons** – упорядочить значки свернутых окон

- **Windows** – выбор текущего окна из списка (если открыто очень много файлов) Дополнительно в этом меню добавляются пункты, соответствующие всем открытым для редактирования файлам.

Следует отметить, что работа с окнами становится неактуальной благодаря системе закладок – все окна распахиваются (максимизируются), а переключение между ними осуществляется при помощи ярлычков.

Пункт Workspace

открывает следующие команды:

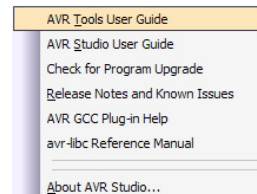


- **Save Workspace** – сохранить рабочее пространство

- **Delete Workspace** – удалить рабочее пространство

- **Default** – включить рабочее пространство по умолчанию

Эти команды позволяют запомнить расположение всех областей, чтобы потом воспользоваться им одной командой.



пользоваться им одной командой.

Help – мощь

Меню традиционное, но

содержит ряд особых команд, сильно облегчающих работу с AVR Studio (если вы хорошо владеете английским языком).

- **AVR Tools User Guide** – руководство пользователя по аппаратным устройствам фирмы Atmel. Позволяет ознакомиться с предлагаемым Atmel ассортиментом аппаратных средств поддержки разработчика, причем руководство весьма полное, вплоть до принципиальных схем некоторых устройств.

- **AVR Studio User Guide** – англоязычный справочный файл по работе с AVR Studio. Надеюсь, читателям этой книги этот пункт меню не потребуется.

- **Check for Program Upgrade** – проверить, не вышла ли более новая версия AVR Studio, доступная для загрузки с сайта Atmel.

- **Release Notes and Known Issues** – открывает документ, в котором перечислены особенности и «недоработки» текущей версии AVR Studio. Для рассматриваемой версии 4.14 содержимое этого документа приведено в переводе в приложениях.

- **AVR GCC Plug-in Help** – справка по настройке плагина WinAVR (все аспекты настройки рассматриваются в этой книге).

- **avr-libc Reference Manual** – открывает справочник по стандартным функциям библиотеки avr-libc. Полноценное руководство по этим функциям имеется в этой книге (см. раздел «AVR-LIBC»).

- **About AVR Studio** – выводит сведения о версии AVR Studio



Продолжение в №5/2010

Ресурсы

2 http://www.atmel.com/dyn/resources/prod_documents/AVRStudio4.18SP2.exe

3. <http://sourceforge.net/projects/winavr/files/WinAVR/20100110/WinAVR-20100110-install.exe/download>