

## AVR: программирование в среде AVR Studio

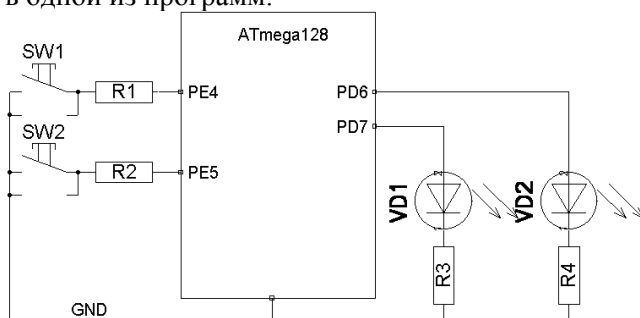
Для программирования AVR-микроконтроллеров существует немало средств разработки, однако, наиболее популярным, несомненно, следует признать пакет AVR Studio. Есть ряд причин такой популярности – это бесплатный пакет, разработанный фирмой ATMEL, он объединяет в себе текстовый редактор, ассемблер и симулятор. Пакет AVR Studio также используется совместно с аппаратными средствами отладки. В предлагаемой статье на примерах рассматриваются приемы работы с пакетом, что поможет начинающим программистам быстрее понять взаимодействие отдельных компонентов AVR Studio.

В следующей части статьи будет рассказано об отладке в среде AVR Studio программ, написанных на языке Си.

Пакет AVR Studio имеет солидную историю развития, что отражается в количестве существующих версий. В конце 2003 г. выпущена версия 4.08, которая имеет ряд полезных дополнений, а в начале 2004 г. вышло обновление (Service Pack 1), добавляющее поддержку AVR-контроллеров третьего поколения семейства ATmega48. Производство микросхем этого семейства намечено на вторую половину 2004 г.

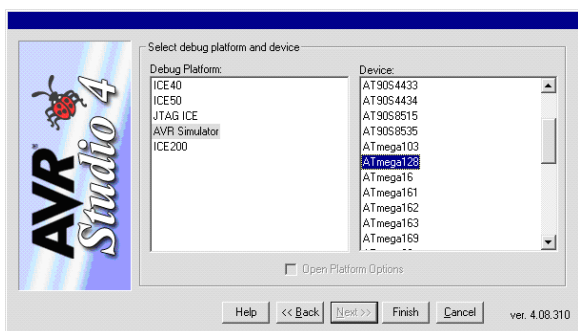
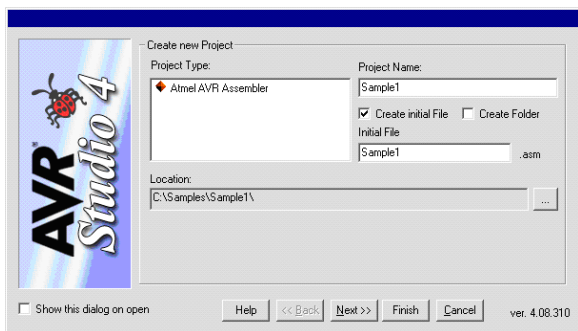
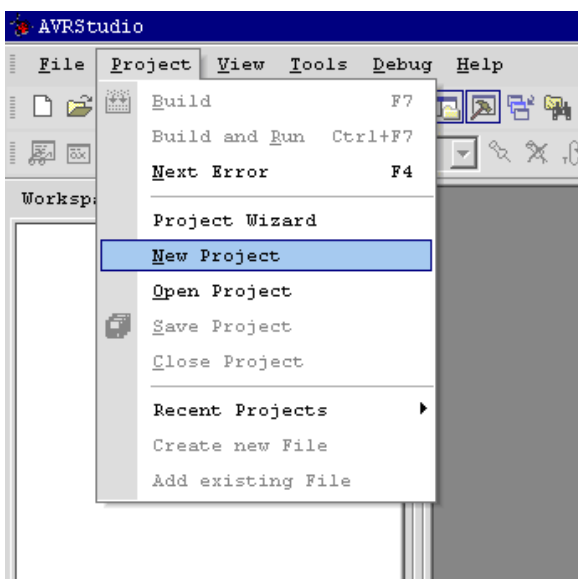
Дистрибутив пакета и Service Pack можно загрузить с сайта [www.atmel.com](http://www.atmel.com) или получить компакт-диск с этим дистрибутивом у российского дистрибьютора фирмы ATMEL.

Работу пакета AVR Studio удобно рассматривать на какой-либо конкретной программе. В качестве иллюстрации мы рассмотрим создание проекта для простейшей программы, которая будет по очереди зажигать два светодиода. Для определенности возьмем микросхему ATmega128 и подключим два светодиода в выводах 31 и 32 (это биты 6 и 7 порта D микросхемы ATmega128). AVR-контроллеры имеют мощные выходные каскады, типовой ток каждого вывода составляет 20 мА, максимальный ток вывода – 40 мА, причем это относится как к вытекающему, так и к вытекающему току. В нашем примере светодиоды подключены анодами к выводам контроллера, а катоды через гасящие резисторы соединены с землей. Это означает, что светодиод зажигается подачей «1» на соответствующий вывод порта. Принципиальная схема приведена на рисунке. На схеме также показаны две кнопки, которые будут использованы в одной из программ.



Здесь уместно сделать небольшое отступление о выборе типа микросхемы для простейшего примера. Действительно, с первого взгляда может показаться странным, зачем нужен такой мощный кристалл в 64-выводном корпусе там, где хватит и 8-выводной микросхемы ATtiny12? Однако, в таком подходе есть логика. Известно, что в основе практически любого AVR-контроллера лежит одинаковое ядро. По большому счету, контроллеры различаются объемом памяти, количеством портов ввода/вывода и набором периферийных модулей. Особенности каждого конкретного контроллера – привязка логических имен регистров ввода/вывода к физическим адресам, адреса векторов прерываний, определения битов портов и т.д. описаны в файлах с расширением .inc, которые входят в состав пакета AVR Studio. Следовательно, используя конкретный тип кристалла, можно отлаживать программу как собственно для него, так и для любого младшего кристалла. Далее, если использовать в качестве отладочного самый старший кристалл, на сегодня это ATmega128, можно отлаживать программу практически для любого AVR-контроллера, надо просто не использовать аппаратные ресурсы, которые отсутствуют у целевого микроконтроллера. Таким образом, например, можно отлаживать на ATmega128 программу, которая будет выполняться на ATtiny13. При этом исходный код останется практически тем же, изменится лишь имя подключаемого файла с 128def.inc на tn13def.inc. У такого подхода также есть свои преимущества. Например, «лишние» порты ввода/вывода можно использовать для подключения ЖК-индикатора, на который можно выводить отладочную информацию. Или, воспользоваться внутрисхемным эмулятором, который подключается к JTAG-порту микросхемы ATmega128 (контроллер ATtiny13 такой порт не имеет). Таким образом, можно использовать единственную отладочную плату, на которой установлен «старший» AVR-контроллер, для отладки любых вновь разрабатываемых систем,

естественно, базирующихся также на AVR-микроконтроллерах. Одна из таких плат называется AS-megaM. Именно она использовалась для создания примеров программ, приводимых в статье. Это универсальный одноплатный контроллер на базе микросхемы ATmega128, который содержит внешнее ОЗУ, два порта RS-232, порт для подключения ЖК-индикатора, внутрисхемного программатора и эмулятора AT JTAGICE. На плате также есть место для распайки микросхемы FLASH-ПЗУ серии AT45 в корпусах TSOP32/40/48 и двухканального ЦАП серии AD5302/12/22. Теперь, после объяснения причин использования AVR-монстра для зажигания пары светодиодов, можно идти дальше.



При программировании в среде AVR Studio надо выполнить стандартную последовательность действий:

- создание проекта
- загрузка файла
- компиляция
- симуляция
- загрузка hex-кода в микроконтроллер

Создание проекта начинается с выбора строки меню Project\New Project. В открывшемся окне “Create new Project” надо указать имя проекта, (в нашем случае – sample1) и имя файла инициализации. После нажатия кнопки “Next” открывается окно “Select debug platform and device”, где выбирается отладочная платформа (симулятор или эмулятор) и тип микроконтроллера.

Можно выбрать один из предлагаемых внутрисхемных эмуляторов, заметим, что у каждого эмулятора свой список поддерживаемых микросхем. Для рассматриваемого примера мы выбираем в качестве отладочной платформы AVR Simulator и микросхему ATmega128. После нажатия кнопки “Finish” нашему взору предстают собственно рабочие окна пакета AVR Studio, пока пустые. Следует в правое окно поместить исходный текст программы. Это можно сделать двумя способами, либо набрать весь текст непосредственно в окне редактора, либо загрузить уже существующий файл. Ниже приведен полный текст простейшей программы с комментариями.

```

; Пример «Управление светодиодами»
; написан для отладочной платы AS-
MegaM
; Частота задающего генератора 7,37
МГц
; светодиоды подключены к выводам
PD6 и PD7 и через резисторы - на общий
провод.

; подключение файла описания ввода-
вывода микросхемы ATmega128
#include "m128def.inc"

; начало программы
begin:

; первая операция - инициализация
стека
; если этого не сделать, то вызов
подпрограммы или прерывания
; не вернет управление обратно
; указатель на конец стека
устанавливается на последний адрес
внутреннего ОЗУ - RAMEND
ldi r16, low (RAMEND)
out spl, r16
ldi
r16, high (RAMEND)
out sph, r16

```

```

; для того, чтобы управлять
светодиодами, подключенными к выводам
PD6 и PD7,
; необходимо объявить эти выводы
выходными.
; для этого нужно записать "1" в
соответствующие биты регистра DDRD (Data
DiRection)
        ldi            r16, (1<<6) |
(1<<7)
        out            DDRD, r16

; основной цикл программы
loop:
        ldi            r16, (1<<6)
; светится один светодиод
        out            PORTD, r16
        rcall         delay
; задержка
        ldi            r16, (1<<7)
; светится второй светодиод
        out            PORTD, r16
        rcall         delay ;
задержка

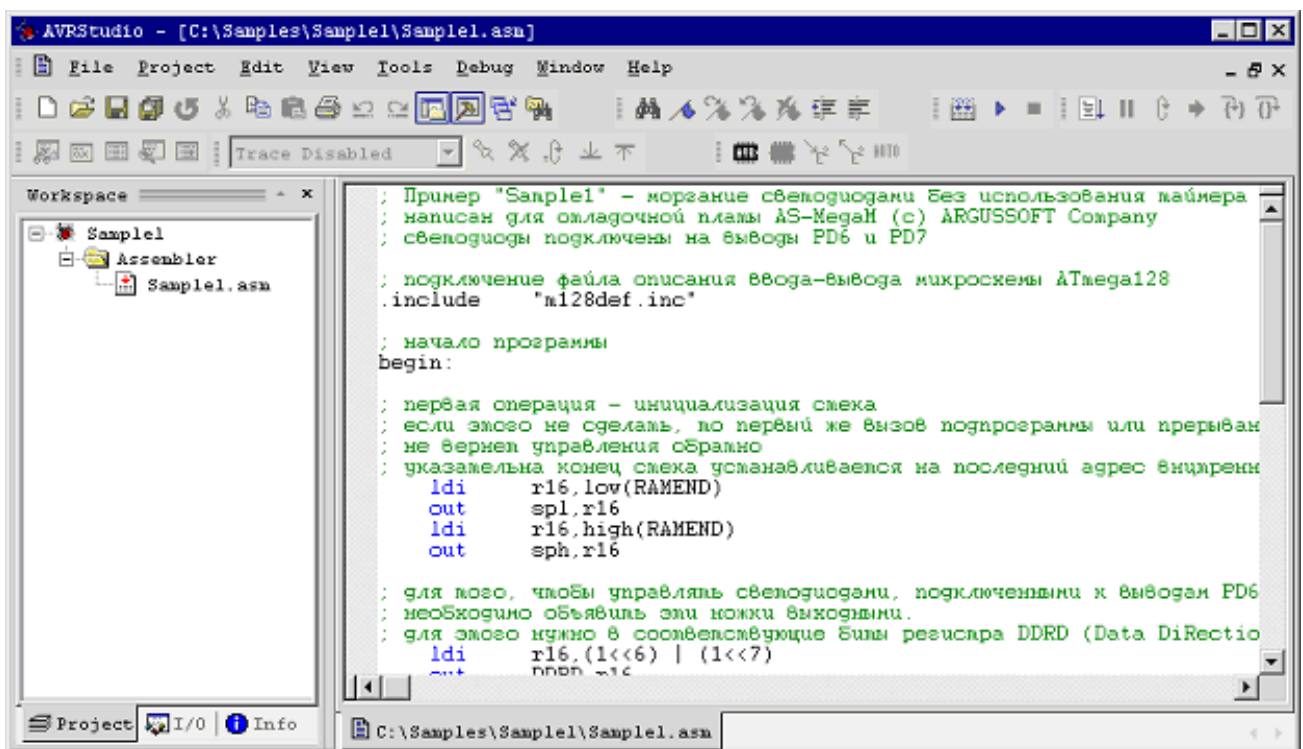
```

```

        rjmp          loop ;
повторение цикла

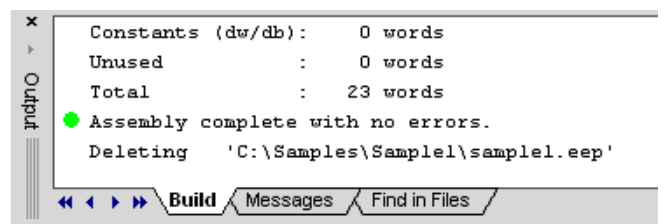
; процедура задержки
; примерно полсекунды при частоте
7,37 МГц
; три пустых вложенных цикла
соответственно
delay:
        ldi            r16, 30 ; 30
delay1:
        ldi            r17, 200
; 200
delay2:
        ldi            r18, 200
; и еще 200 итераций
delay3:
        dec            r18
        brne          delay3
        dec            r17
        brne          delay2
        dec            r16
        brne          delay1
        ret
; возврат в главную программу

```

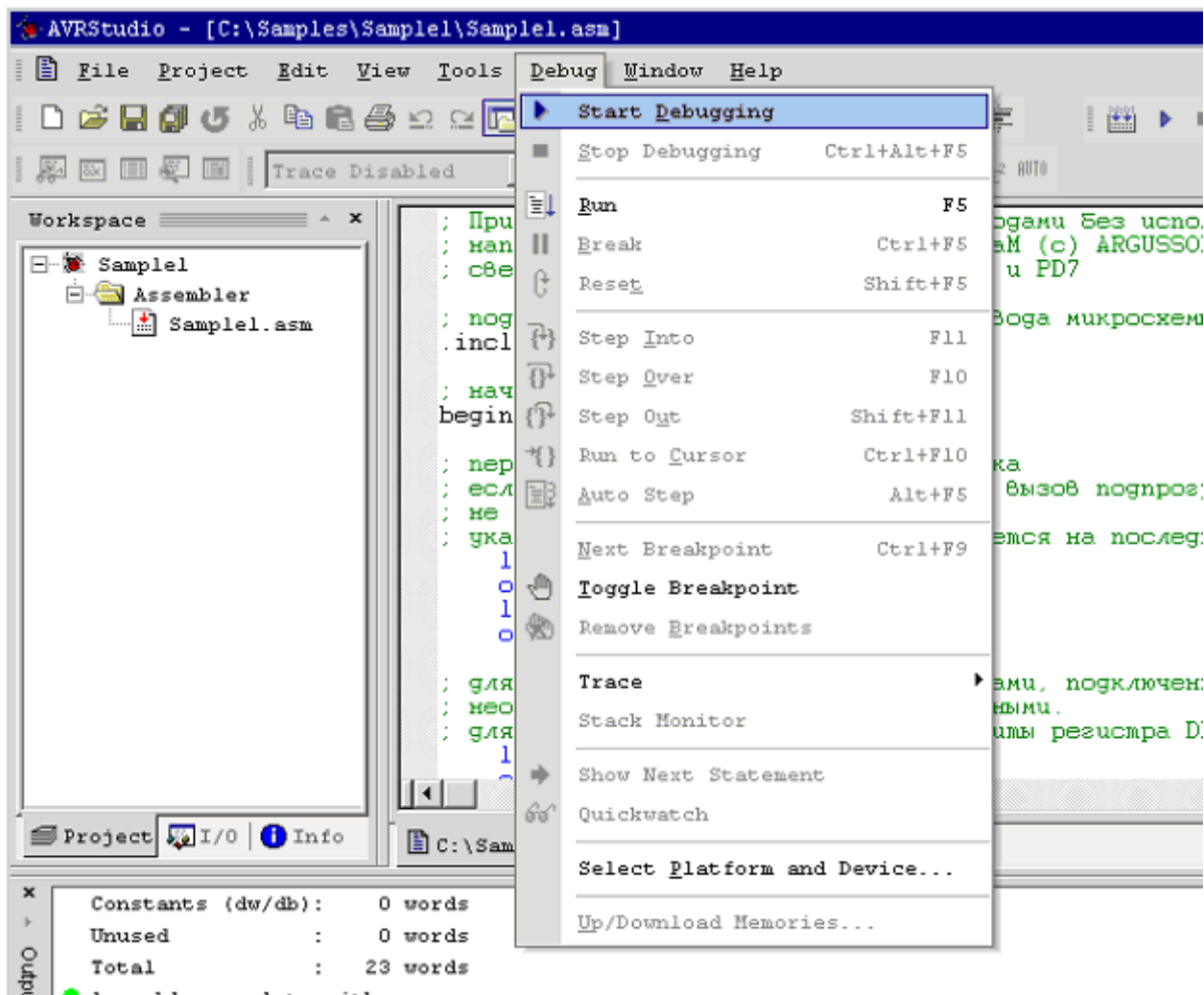


Проект может состоять из нескольких файлов, при этом один файл назначается основным. Все операции удобно производить, используя контекстную кнопку мыши. После подключения исходного файла окна имеют следующий вид.

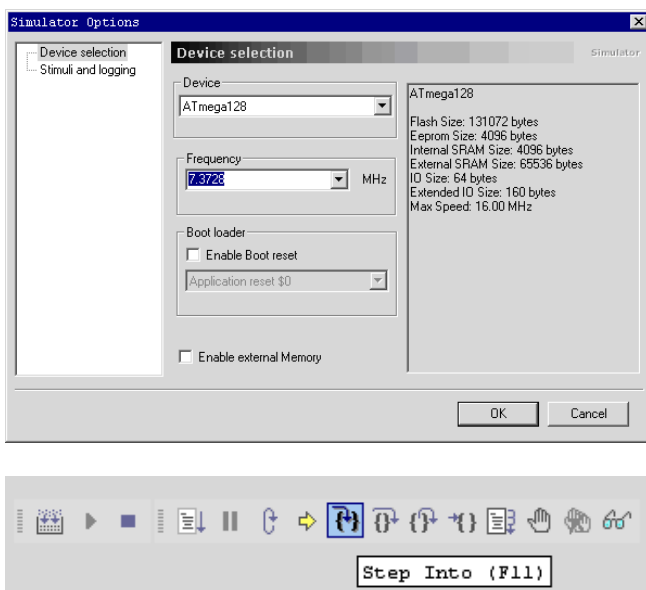
Компиляция проекта производится командой \Project\Build или нажатием кнопки F7. Процесс компиляции отображается в окне "Output". Это окно можно «вытащить» командой \View\Output.



В принципе, мы уже получили выходной файл в формате .hex, который уже можно загружать в микросхему и наблюдать перемигивание светодиодов. Однако, цель статьи –



показать полный цикл работы в среде AVR Studio, поэтому мы переходим к стадии отладки. Это делается командой \Debug\Start Debugging.



Теперь устанавливаем в окне “Simulator Options” частоту кварца 7,3728 МГц для точного измерения времени выполнения программы .

Остальные опции следует оставить без изменения. Теперь можно выполнять программу в пошаговом режиме при помощи мыши или кнопки F11.

Пакет AVR Studio содержит мощные средства для просмотра и редактирования состояния внутренних регистров и портов ввода/вывода отлаживаемого микроконтроллера, а также время, выполнения программы . Доступ к ним осуществляется через окно “I/O”.

На самом деле, количество информации, доступное через окна просмотра пакета AVR Studio настолько велико, что для получения максимального комфорта нужно использовать компьютер в двухмониторной конфигурации.

Для отладки нашего примера, чтобы получить доступ к битам порта D, надо раскрыть строку I/O ATMEGA128 и затем строку PORTD. Теперь видны все три регистра этого порта, PORTD, DDRD и PIND. Чтобы увидеть поля Value,

Bits и Address, придется расширить правую границу окна, потеснив при этом окно с исходным текстом программы.

Теперь, проходя программу в пошаговом режиме, можно видеть изменение текущих состояний этих регистров в поле Bits. Есть возможность оперативного изменения состояния любого бита регистров порта, причем это можно делать либо записью нового кода в поле Value, либо непосредственно, щелкнув мышью на нужном бите регистра.

Для самостоятельных упражнений, предлагается следующая программа, которая отличается от предыдущей тем, что зажиганием светодиодов управляют две кнопки.

```
; Пример «Управление светодиодами
от кнопок»
; написан для отладочной платы AS-
MegaM
; светодиоды подключены к выводам
PD6 и PD7 и через резисторы - на общий
провод.
; кнопки - на PE4 и PE5

.include      "m128def.inc"

; основная программа
begin:
; инициализация стека
    ldi      r16,low(RAMEND)
    out      spl,r16
    ldi
r16,high(RAMEND)
    out      sph,r16

; инициализация светодиодов
    ldi      r16,(1<<6) |
(1<<7)
    out      DDRD,r16

; инициализация выводов, к которым
подключены кнопки (на вход)
; внутренние подтягивающие
резисторы подключены
; для этого в PORTE нужно
установить соответствующие биты в единицы
(1<<5)
    out      PORTE,r16
; а в DDRE - в нули
    ldi      r16,0
    out      DDRE,r16

; бесконечный цикл
forever:
    in       r16,PINE      ;
теперь в r16 находится текущее "состояние"
кнопок
    com      r16          ;
кнопка "нажимается" нулем, поэтому
инвертируем регистр
    lsl      r16          ;
переносим биты 4,5 в позиции 6,7
    lsl      r16          ;
и обновляем "показания" светодиодов
```

```
andi      r16,(1<<6) | (1<<7)
    out      PORTD,r16
rjmp      forever        ; цикл
выполняется бесконечно
```

Таким образом, на примере простейших программ показаны некоторые возможности пакета AVR Studio. Надо понимать, что это лишь первое знакомство, позволяющее быстрее освоиться с базовыми командами пакета. Между тем, возможности рассматриваемого пакета намного шире. Например, здесь можно отлаживать программы написанные на языках высокого уровня. В частности, Си-компилятор фирмы ImageCraft пользуется отладчиком AVR Studio «как родным». Для этого при компиляции исходного кода надо установить опцию генерации выходного файла в формате, совместимом с AVR Studio. При этом появляется возможность производить отладку в исходных кодах.

Еще одна из многих характеристик пакета AVR Studio - возможность подключения внешних программ. Например, для обеспечения вызова оболочки внутрисхемного программатора AS2 нужно выполнить несколько простых операций.

В меню *Tools* главного окна AVR Studio надо выбрать пункт *Customize*;

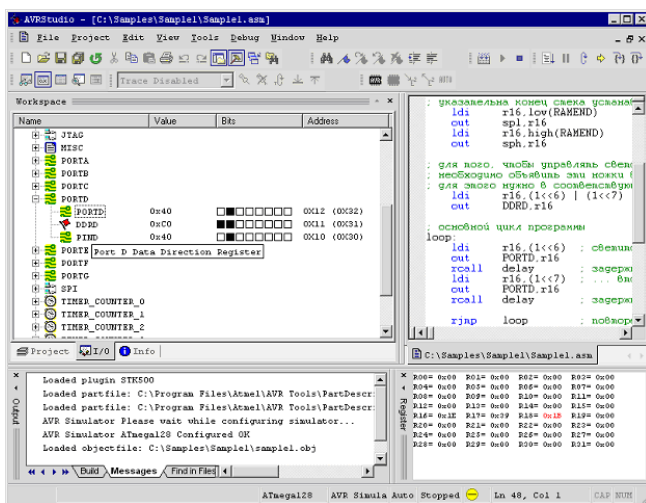
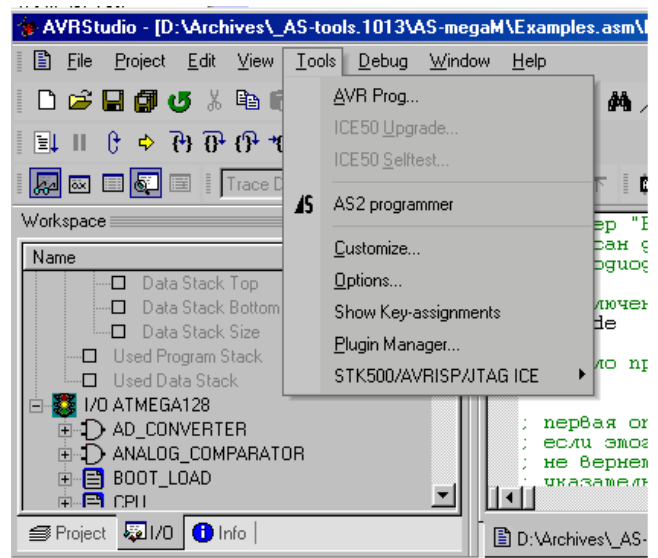
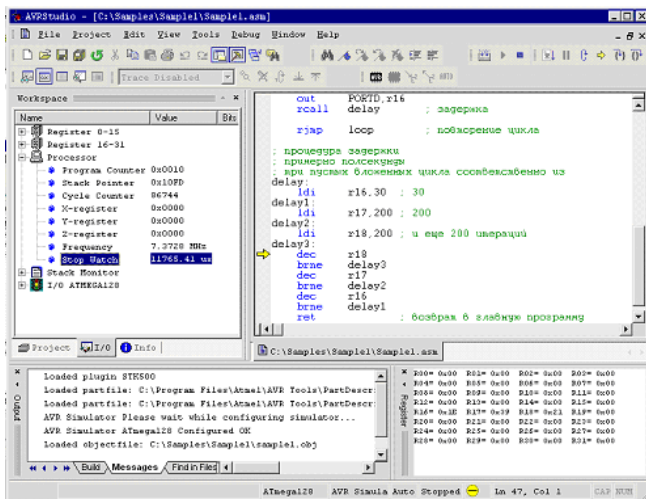
В окне *Customize* выбрать пункт *Tools*;

Двойным нажатием кнопки мыши или нажав Insert на клавиатуре, добавить новую команду в список и назвать ее "Программатор AS2";

Указать путь к исполняемому файлу программатора, введя его непосредственно в поле для ввода "Command", или нажав на кнопку "..." справа от этого поля;

Теперь в меню Tools появился пункт "Программатор AS2".

Средства пакета AVR Studio 4.08 позволяют подключать вспомогательные программы – plugins. Первый plugin для AVR Studio – это программа графического редактора, упрощающая процесс инициализации ЖК-индикатора, которым может непосредственно управлять AVR-контроллер ATmega169. Максимальный логический размер ЖК-индикатора составляет 100 сегментов, каждому элементу индикатора ставится в соответствие бит в специальном регистре контроллера. Чтобы упростить рутинную процедуру привязки определенных битов к каждому сегменту, можно использовать вышеупомянутую программу.



версию (4.09) - будет включен интерфейс для нового внутрисхемного эмулятора – JTAGICE mkII (он называется также AT JTAGICE2), который во второй половине года придет на смену AT JTAGICE. У этого эмулятора есть два существенных отличия. С одной стороны, добавлена поддержка нового однопроводного отладочного интерфейса для младших AVR-контроллеров, debugWIRE. Этот интерфейс интересен тем, что он не занимает для своей работы дополнительные выводы микроконтроллера, так как использует для обмена вывод Reset микроконтроллера! С другой стороны (можно понимать это выражение буквально), у эмулятора AT JTAGICE2 появится, наконец, интерфейс USB для связи с компьютером.

Во время посещения «родины AVR» - норвежского офиса фирмы ATMEL, один из авторов статьи беседовал с Ларсом Квенилдом, руководителем группы программистов, которая создала и поддерживает пакет AVR Studio. Этот человек, классический программист, с бородой, в свитере и обутый в сандали на носки, рассказал о перспективах развития пакета. В следующую

Исходные коды этих и других примеров программ можно переписать с сайта официального дистрибьютора фирмы ATMEL в России АРГУССОФТ Компани: <http://atmel.argussoft.ru>

## Литература

1. Материалы технического семинара AVR Technical Training. Atmel. Norway. December 2003.
2. Николай Королев, Дмитрий Королев AVR-микроконтроллеры второго поколения: средства разработчика. // Компоненты и технологии, 2003 № 7
3. Николай Королев, Дмитрий Королев AVR-микроконтроллеры второго поколения: новые аппаратные возможности. // Компоненты и технологии, 2003 № 4

4. Николай Королев, Дмитрий Королев  
AVR-микроконтроллеры: большое в малом.  
//Схемотехника», 2001, №5

5. Николай Королев, Дмитрий Королев  
AVR-микроконтроллеры: программные средства  
// Компоненты и технологии, 2000 № 4, г.

6. Николай Королев  
AVR: аппаратные средства разработчика  
// Компоненты и технологии, 1999 № 1

7. Николай Королев  
RISC- микроконтроллеры фирмы ATMEL  
//Chip-News 1998, №2

8. Николай Королев, Дмитрий Королев  
AVR: новые 8-разрядные RISC-микроконтроллеры фирмы ATMEL  
//Микропроцессор Ревю, 1998, №1