

AVR-микроконтроллеры: семь ярких лет становления. Что дальше?

Часть 3. Программные и аппаратные средства поддержки разработок для микроконтроллеров AVR

Мы завершаем цикл публикаций, посвященных одной из наиболее удачных восьмиразрядных микроконтроллерных платформ — AVR 8-bit RISC производства Atmel Corp. Предыдущие части статьи, опубликованные в «КиТ» №№ 1–2 '2004, содержали краткий маркетинговый обзор мирового рынка микроконтроллеров, положения в нем AVR, некоторых особенностей ядра и периферийных блоков AVR образца 2004 года. Наряду с бурным развитием и совершенствованием микроконтроллеров AVR развиваются аппаратные и программные средства поддержки разработок, которым Atmel уделяет немалое внимание. Данная публикация представляет собой обзор существующих на сегодняшний день программных и аппаратных средств поддержки разработок для микроконтроллеров AVR и их особенностей.

Алексей Курилин

ak@efo.ru

Роман Золотуха

roman@efo.ru

Виктор Березин

bvv73@mail.ru

Стартовые наборы разработчика

Применение стартовых наборов позволяет разработчику быстро, минуя этап макетирования, приступить к практической разработке приложений. На платах, входящих в состав стартовых наборов, установлены необходимые для работы микроконтроллера элементы «обвязки» (стабилизатор напряжения питания, тактовый генератор или кварцевый резонатор, цепь сброса, средства для организации внутрисхемного программирования микроконтроллера). Также устанавливаются часто применяемые на практике узлы микроконтроллерных устройств (средства ввода и индикации, формирователи интерфейса RS-232, интерфейс с внешним ОЗУ и пр.). Все порты микроконтроллеров выведены на разъемы и могут быть соединены с внешними устройствами.

В некоторых случаях стартовые наборы разработчика могут использоваться как готовые блоки в составе проектируемой аппаратуры.

Стартовый набор STK500

Компания Atmel предлагает универсальный стартовый набор STK500 для всех микроконтроллеров AVR, выпускаемых в DIP-корпусах. Для поддержки разработок на базе микроконтроллеров, выпускаемых только в корпусах TQFP64 и TQFP100, выпускаются платы расширения STK501 и STK503, а также STK502 и STK504 для микроконтроллеров в корпусах TQFP64 и TQFP100 с драйвером ЖКИ.

Помимо функций отладочной платы, STK500 выполняет функции параллельного программатора (для всех типов микроконтроллеров AVR), и последовательного внутрисхемного программатора (кроме Tiny11 и Tiny28, не поддерживающих функцию по-

следовательного программирования). При программировании микроконтроллера можно производить конфигурирование микроконтроллера установкой FUSE-битов (FUncion SEt).

Функции внутрисхемного программирования и управления STK500 реализованы на двух микроконтроллерах: 90S1200-12SC и 90S8535-8AC. Кроме них на плате STK500 смонтированы следующие элементы:

- панели для установки микроконтроллеров в корпусах DIP8, DIP20, DIP28 и DIP40;
- стабилизированный источник питания с управляемым программно выходным напряжением;
- преобразователи уровней сигналов (для случая, когда напряжение питания целевого микроконтроллера отличается от напряжения питания управляющих микроконтроллеров);
- управляемый программно источник опорного напряжения для внутреннего АЦП микроконтроллера;
- посадочное место для микросхемы DataFlash AT45DB021;
- двухканальный формирователь уровней сигналов интерфейса RS232 (один канал используется для связи STK500 с управляющим персональным компьютером, другой может быть использован в разрабатываемом приложении);
- восемь кнопок и восемь светодиодов для организации ввода и отображения информации;
- разъемы расширения для подключения внешних устройств.

Для тактирования отлаживаемых микроконтроллеров на плате STK500 предусмотрены два источника тактовых сигналов. Один из них представляет собой генератор, построенный на инвертирующих вентилях с кварцевой стабилизацией частоты. Пользователь имеет возможность задавать частоту этого генератора, устанавливая кварцевый резона-



Рис. 1. Отладочная плата STK500

тор на необходимую частоту в специальную панель (помечена CRYSTAL).

Второй источник тактовых сигналов представляет собой выход установленного на плате STK500 управляющего микроконтроллера 90S8535-8AC. Частота этого тактового сигнала может быть задана программно. Доступны значения 3,69; 1,84; 1,23 МГц, а также 32,7 кГц и полная остановка.

Переключатель OSCSEL служит для выбора одного из источников тактового сигнала для микроконтроллера. Если же в качестве тактового сигнала необходимо использовать внутренний RC-генератор микроконтроллера, то внешний тактовый сигнал должен быть отключен джампером XTAL1.

На плате STK500 не предусмотрено подключение к целевому микроконтроллеру внешнего ОЗУ.

Управление STK500 производится через COM-порт персонального компьютера. Управляющая программа является составной частью AVR Studio (более подробно о данном пакете будет написано ниже) и запускается из меню Tools / STK500/AVRISP/JTAGICE. Кроме STK500 эта управляющая программа поддерживает такие аппаратные средства отладки, как внутрисхемный программатор ATAVRISP и внутрисхемный эмулятор ATJTAGICE mkII. Подробнее возможности управляющей программы будут рассмотрены ниже, при обзоре средств внутрисхемного программирования. Здесь мы ограничим описание только функциями, уникальными для STK500. На рис. 2 представлено окно программы, управляющей настройками STK500.

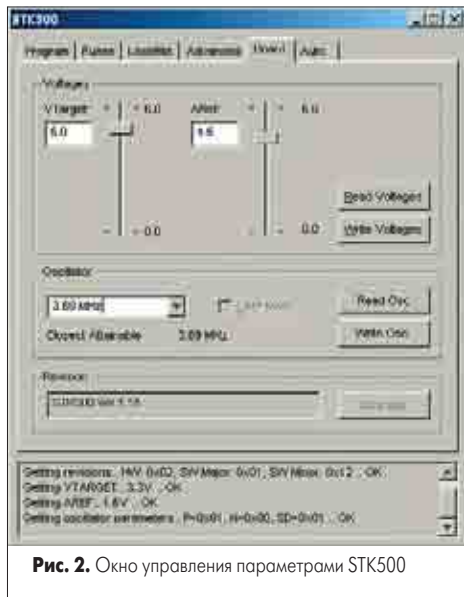


Рис. 2. Окно управления параметрами STK500

Закладка **Board** служит для установки параметров STK500:

- напряжения питания целевого микроконтроллера (**VTarget**);
- опорного напряжения АЦП микроконтроллера (**ARef**);
- частоты управляемого источника тактового сигнала (**Oscillator**);
- а также для индикации версий прошивки (firmware) управляющих микроконтроллеров (**Revision**).

STK500 поддерживает различные режимы программирования целевых микроконтроллеров: режим последовательного внутрисхемного программирования и два режима программирования с использованием повышенного программирующего напряжения — параллельный и последовательный. Выбор режима программирования осуществляется в окне управления параметрами STK500 в закладке **Program**.

Для использования STK500 в качестве внутрисхемного программатора для микроконтроллеров, установленных в целевом устройстве, необходимо соединить целевое устройство с разъемом **ISP10PIN** платы STK500, используя плоский десятижильный кабель.

Программная поддержка новых микроконтроллеров AVR в STK500 производится обновлением прошивки (firmware) памяти программ управляющих микроконтроллеров (**Revision Upgrade**). Обновленные версии firmware включаются в состав AVR Studio. Управляющая программа в момент запуска проверяет связь COM-порта персонального компьютера с STK500, и в случае его присутствия запрашивает версию firmware. Если в AVR Studio содержится более новая версия прошивки, управляющая программа предлагает обновить firmware путем перепрограммирования управляющих микроконтроллеров. Перед входом в режим перепрограммирования STK500 необходимо извлечь целевые микроконтроллеры из DIP-панелей, если они туда установлены. Для входа в режим перепрограммирования необходимо включить питание STK500 с нажатой кнопкой **Program**.

В комплекте с STK500 поставляются микроконтроллеры AT90S8515 и ATmega16, а также диск AVR Software and Technical Library, включающий AVR Studio. Также в состав набора входит комплект кабелей.

Плата расширения STK501

Для поддержки микроконтроллеров в корпусах TQFP64 (кроме микроконтроллеров с драйвером ЖКИ) компания Atmel предлагает устройство STK501, которое устанавливается в разъемы расширения STK500 как мезонинная плата. На плате STK501 размещены следующие элементы:

- панель с нулевым усилием (ZIF socket) для установки микроконтроллеров в корпусах TQFP64 (здесь и далее в разделе — кроме микроконтроллеров с драйвером ЖКИ);
- дополнительный порт RS232 с поддержкой линий RTS/CTS;
- кварцевый резонатор на 32 кГц для реализации часов реального времени (Real-Time Clock, RTC);



Рис. 3. Плата расширения STK501

- посадочное место для пайки микроконтроллера в корпусе TQFP64 либо адаптера внутрисхемного эмулятора ICE50;
- разъем JTAG-интерфейса для подключения внутрисхемного эмулятора ATJTAGICE mkII.

Кристаллы в корпусах TQFP64 имеют больше портов ввода-вывода, чем предусмотрено в STK500. Поэтому порты E, F, G, а также некоторые управляющие сигналы выведены на разъемы на плате STK501.

Комбинация STK500 и STK501 позволяет выполнять функции параллельного и последовательного программатора для всех микроконтроллеров в корпусах TQFP64.

Как уже говорилось, стартовый набор разработчика STK500 не поддерживает работу микроконтроллера с внешним ОЗУ. Разработчики STK501 исправили этот недостаток, предусмотрев на плате STK501 посадочное место под микросхему ОЗУ объемом до 128 кбайт (в корпусах TSOP32 или SOJ32) и регистр-защелку младшего байта адреса. При этом STK501 без установленного микроконтроллера может служить платой расширения для STK500, позволяющей создавать и отлаживать проекты для микроконтроллеров mega8515 или mega162 с использованием внешней памяти данных.

Для поддержки микроконтроллера AT90CAN128 с CAN-интерфейсом выпускается специальный модуль расширения ATADAPCAN01, который используется в связке с отладочными платами STK500 и STK501 (рис. 4), а также подключается непосредственно к целевому устройству.

Модуль разработан в соответствии со стандартом Bosch CAN 2.0A/B и реализует функции преобразования сигнала физического уровня CAN-интерфейса, необходимые для присоединения отлаживаемой системы к ши-

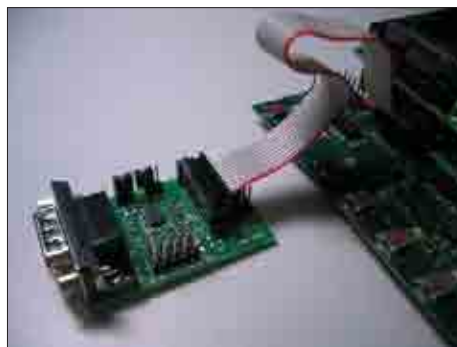


Рис. 4. Подключение ATADAPCAN01 к STK500/STK501

не CAN для разработки и отладки конечного приложения.

В комплекте с STK501 поставляется микроконтроллер ATmega128L и диск AVR Software and Technical Library. В комплекте с ATADAPCAN01 поставляется микроконтроллер AT90CAN128.

Плата расширения STK502

Для поддержки микроконтроллеров в корпусах TQFP64 с драйвером ЖКИ (ATmega169/329) компания Atmel выпустила устройство STK502, которое устанавливается в разъемы расширения STK500 как мезонинная плата, также как и STK501. Состав платы STK502 аналогичен составу STK501 (за исключением дополнительного порта RS232 и площадки для ОЗУ), она имеет панель с нулевым усилением (ZIF socket) для установки микроконтроллеров, кварцевый резонатор на 32 кГц, разъем JTAG-интерфейса, посадочное место для пайки микроконтроллера в корпусе TQFP64 либо адаптера внутрисхемного эмулятора.

Также на плате STK502 содержится жидкокристаллический индикатор (ЖКИ) со 120 сегментами, который подключается к выводам микроконтроллера плоским 34-жильным кабелем, датчик температуры (NTC-термистор), используемый в демонстрационной программе «Измерение температуры с выводом на ЖКИ», посадочное место для пайки микроконтроллера. Разъем расширения для подключения внешнего ЖКИ — Segment pins from ATmega169 — дублирует порты A, C, D и G для более удобного подключения жидкокристаллического дисплея.



Рис. 5. Плата расширения STK502

Комбинация STK500 и STK502 также позволяет выполнять функции параллельного программатора для всех микроконтроллеров в корпусах TQFP64, включая микроконтроллеры с драйвером ЖКИ. Программирование в параллельном режиме производится с использованием портов В и D при повышенном напряжении, поэтому необходимо отключать ЖК-дисплей от микроконтроллера для исключения выхода дисплея из строя. При использовании портов А, С, D и G в качестве портов ввода-вывода ЖК-дисплей также должен быть отключен.

Разработчики платы STK502 не предусмотрели возможность программирования в последовательном режиме микроконтроллеров в корпусах TQFP64, которые не имеют драйвера ЖКИ. Но при необходимости, можно обеспечить режим последовательного программирования для этих микроконтроллеров. Это делается небольшим изменением в соединении

разъемов SPROG (на плате STK502) и ISP6PIN (на плате STK500). Стандартно для обеспечения режима последовательного программирования производится соединение вывода MISO и MOSI разъема ISP6PIN с выводами PB2 и PB3 микроконтроллера. Для последовательного программирования микроконтроллеров в корпусах TQFP64, которые не имеют драйвер ЖКИ, необходимо произвести соединение выводов MOSI и MISO разъема ISP6PIN с выводами PE0 и PE1 микроконтроллера.

В комплекте с платой STK502 поставляется микроконтроллер ATmega169V и диск AVR Software and Technical Library, в состав которого входит AVR Studio.

Плата расширения STK503

Мезонинная плата STK503, на сегодняшний день официально не представленная для широкого круга потребителей, предназначена для поддержки многовыводных микроконтроллеров в корпусах TQFP100 (не содержащих драйвер ЖКИ). Плата STK503, аналогично вышеописанным продуктам, устанавливается в разъемы расширения STK500 как мезонинная плата. На плате STK503 размещены:

- панель с нулевым усилением (ZIF socket) для установки микроконтроллеров в корпусах TQFP100;
- два преобразователя интерфейсов TTL — RS232;
- кварцевый резонатор на 32 кГц для реализации часов реального времени (Real-Time Clock, RTC);
- разъем JTAG-интерфейса для подключения внутрисхемного эмулятора JTAGICE.

В отличие от платы STK501, выполняющей аналогичные функции для 64-выводных микроконтроллеров, на плате STK503 выведено большее число портов ввода-вывода, что обусловлено наличием большего числа портов ввода-вывода на 100-выводных микроконтроллерах.



Рис. 6. Плата расширения STK503

Аналогично плате STK501, плата STK503 содержит посадочное место под микросхему ОЗУ.

Отличие состоит в том, что тактирование, помимо источников тактового сигнала с платы STK500 и встроенного 32 кГц кварцевого резонатора, может осуществляться также от высокочастотного кварца, устанавливаемого в имеющийся на плате STK503 разъем. Выбор источника производится аппаратными переключателями, установленными на плате.

Для реализации интерфейса RS232 на плате имеется двоякий преобразователь уровней TTL — RS232. Выводы интерфейса RS232 выведены на восьмиконтактный разъем.

Комбинация STK500 и STK503 позволяет выполнять функции параллельного и последовательного программатора для всех микроконтроллеров в корпусах TQFP100. Возможно, исключение составят микроконтроллеры с драйвером ЖКИ.

Отсутствие посадочного места для пайки микроконтроллера в STK503 является отличительной особенностью платы от остальных представителей линейки мезонинных плат.

В комплект набора STK503, предположительно, будет входить микроконтроллер ATmega2560.

Плата расширения STK504

Аналогично плате STK502 для поддержки многовыводных микроконтроллеров в корпусах TQFP100 с драйвером ЖКИ компания Atmel выпустила устройство STK504, которое устанавливается в разъемы расширения STK500 в качестве мезонинной платы.



Рис. 7. Плата расширения STK504

На плате установлен разъем ZIF для установки микроконтроллера в корпусе TQFP100, жидкокристаллический дисплей на 160 сегментов. Дисплей, аналогично STK502, подключается с помощью плоского кабеля и может быть отключен при необходимости. Также может быть использован внешний ЖК-дисплей.



Рис. 8. Набор сегментов 160-сегментного ЖКИ платы STK504

Тактирование микроконтроллера в STK504 может осуществляться внешним тактовым сигналом, поступающим с платы STK500, кварцевого резонатора на 32кГц, установленного на плате STK504, а также от высокочастотного кварца, устанавливаемого в имеющийся на плате STK504 сокет. Выбор источника производится аппаратными переключателями, установленными на плате.

Дополнительные порты (порт E, порт F, порт G, порт H, порт J) выведены на отдельные разъемы на плате STK504. На разъеме порта G по умолчанию установлен джампер, соединяющий вывод 6 и вывод 8 разъема, что соответствует соединению сигнала сброса (/RESET) микроконтроллера (PG.5) и сигнала сброса (/RESET), поступающего с платы STK500. При использовании внутрисхемного эмулято-

ра JTAGICE2 (будет описан ниже) джампер должен быть извлечен, чтобы не блокировались сигналы сброса, поступающие с эмулятора.

Комбинация STK500 и STK504 позволяет выполнять функции параллельного и последовательного программатора для микроконтроллеров в корпусах TQFP64 с драйвером ЖКИ. Как и в случае с STK502, не следует забывать отключать ЖКИ от портов ввода-вывода микроконтроллера во время программирования в параллельном режиме, при этом, напряжение питания, подаваемое с STK500 на микроконтроллер, должно быть не ниже 4,5 В.

В комплекте с STK504 поставляется микроконтроллер ATmega3290.

Внутрисхемный программатор ATAVRISP

Как уже говорилось, микроконтроллеры AVR имеют несколько режимов программирования: параллельный режим с использованием повышенного программирующего напряжения, последовательный режим с использованием повышенного программирующего напряжения и режим последовательного внутрисхемного программирования.

Возможность внутрисхемного программирования микроконтроллеров AVR по последовательному синхронному интерфейсу SPI позволяет создавать простые и недорогие программаторы. Такие программаторы называют внутрисхемными (In-System Programmer, ISP).

Для реализации внутрисхемного последовательного программирования компания Atmel выпускает и поддерживает внутрисхемный программатор ATAVRISP. ATAVRISP поддерживает все выпускаемые на сегодняшний день микроконтроллеры AVR, имеющие функцию последовательного внутрисхемного программирования. Управление этим внутрисхемным программатором осуществляется через COM-порт персонального компьютера. Питание на ATAVRISP подается от целевого устройства.

Подобно стартовому набору разработчика STK500, внутрисхемный программатор ATAVRISP реализован на двух микроконтроллерах — 90S1200-12SC и 90S8535-8AC, и работает под управлением AVR Studio. Управляющая програм-



Рис. 9. Внутрисхемный программатор ATAVRISP

ма вызывается в меню Tools / STK500/AVRISP/JTAGICE / STK500/AVRISP/JTAGICE.

Выбрав одну из закладок в окне управляющей программы, пользователь получает доступ к различным функциям программирования:

- **Program.** Окно управляющей программы в этом режиме представлено на рис. 10. Здесь пользователь указывает тип целевого микроконтроллера (**Device**), режим программирования (в нашем случае **ISP**), а также файлы, содержащие данные для программирования Flash-ПЗУ и EEPROM микроконтроллера. Здесь же пользователь может указать управляющей программе на необходимость автоматического стирания памяти программ микроконтроллера перед очередным программированием, а также верификацию записанного кода — после программирования.
- **Fuses.** Позволяет считать состояние управляющих битов микроконтроллера и при необходимости запрограммировать их. Операцию установки FUSE-битов нужно завершать нажатием кнопки **Program** на вкладке Fuses.
- **LockBits.** Предоставляет доступ к битам защиты памяти программ микроконтроллера от несанкционированного считывания или модификации. Операцию установки LOCK-битов нужно завершать нажатием кнопки **Program** на вкладке LockBits.
- **Advanced.** Позволяет считать сигнатуру целевого микроконтроллера. Здесь же может быть задано значение калибровочного байта для внутреннего RC-генератора микроконтроллера.
- **Board.** Эта вкладка предназначена для управления STK500 и была описана выше.
- **Auto.** Позволяет выбрать несколько функций программирования и верификации и запускать их в пакетном режиме.

При каждом переключении с одной закладки на другую управляющая программа проверяет связь компьютера с внутрисхемным программатором. Если внутрисхемный программатор не отвечает на запрос компьютера, в поле сообщений окна управляющей программы выдается соответствующее предупреждение.

При работе с целевым микроконтроллером программатор ATAVRISP помимо линий внутрисхемного программирования (MOSI, MISO, CLK) использует вывод /RESET, GND, а также линию питания (VTG), по которой осуществляется питания программатора.

Все описанные функции программирования микроконтроллера также могут быть выполнены на плате STK500 в последовательном, а также, в отличие от ATAVRISP, в параллельном режиме.

ATAVRISP поставляется в комплекте с диском AVR Software and Technical Library, кабелем для подключения к COM-порту компьютера, двумя кабелями (с 10- и 6-контактными разъемами) для подключения к целевой плате. Питание программатора осуществляется от целевого устройства по линиям VTG и GND.

Вся необходимая информация по подключению и использованию программатора AVRISP включена во встроенный файл справки AVR Studio.



Рис. 10. Окно управляющей программы внутрисхемного программатора ATAVRISP

Внутрисхемные эмуляторы

Принцип внутрисхемной эмуляции состоит в замене целевого микроконтроллера неким устройством, поведение которого с точки зрения отлаживаемой системы соответствует поведению целевого микроконтроллера. Это устройство, называемое внутрисхемным эмулятором, должно не только адекватно отражать поведение целевого микроконтроллера, но и предоставлять разработчику возможность управления процессом отладки и контроля состояния реальной отлаживаемой микропроцессорной системы. Для управления процессом отладки при использовании современных внутрисхемных эмуляторов, как правило, используется персональный компьютер с соответствующим программным обеспечением.

В настоящий момент компания Atmel предлагает два типа внутрисхемных эмуляторов.

Первый тип подразумевает полную замену микроконтроллера на некое устройство, которое эмулирует поведение реального микроконтроллера на целевой плате. Данный тип на сегодняшний день представлен двумя внутрисхемными эмуляторами ICE40 и ICE50.

Второй тип подразумевает использование микроконтроллера AVR, имеющего все необходимые для внутрисхемной отладки аппаратные блоки и управляемого внешним отладчиком. На данный момент этот тип отладчиков представлен устройством ATJTAGICE mkII (читается «Марк два»).

Использование внутрисхемных эмуляторов позволяет отлаживать приложения, в которых задействованы периферийные узлы и режимы микроконтроллеров, поддержка которых отсутствует в программном симуляторе.

Внутрисхемные эмуляторы ICE40, ICE50

ICE40 и ICE50 представляют собой классические внутрисхемные эмуляторы. Различие эмуляторов состоит в количестве поддерживаемых микроконтроллеров.

Список поддерживаемых эмулятором ICE50 микроконтроллеров очень широк: ATmega128/



Рис. 11. Внешний вид внутрисхемных эмуляторов ICE40/ICE50

CAN128, ATmega165/325, ATmega169/329/649, ATmega48/88/168, ATmega162, ATmega8/16/32/64, ATmega8515/8535, ATtiny26, ATtiny13, ATtiny2313.

Эмулятор ICE40 поддерживает гораздо меньшее количество микроконтроллеров, а именно микроконтроллеры ATtiny26, ATmega8, ATtiny13.

Для поддержки различных микроконтроллеров имеется набор выносных плат (адаптеров) для пайки либо установки в DIP-панель для микроконтроллера на целевой плате (см. рис. 9). Для микроконтроллеров AVR, которые не выпускаются в DIP-корпусах, адаптер (ATADAP64BOT) предназначен для припаивания на площадку под корпус TQFP. По типу подключенного адаптера ICE40/50 автоматически определяет, какой микроконтроллер будет эмулироваться. Адаптеры для поддержки микроконтроллеров ATtiny13, ATmega169/329/649, ATmega165/325 в комплект эмулятора ICE40/50 не входят, а заказываются отдельно.



Рис. 12. Внешний вид выносных плат эмулятора ICE40/50

Эмуляторы ICE40 и ICE50 выполняют следующие функции:

- эмуляция аналоговой и цифровой периферии микроконтроллера;
- работа в диапазоне напряжений от 2,2 до 5,5 В;
- просмотр значений регистров ввода-вывода;
- просмотр регистрового файла, памяти программ, данных и EEPROM;
- запись трассы;
- неограниченное количество точек останова;
- отладка по тексту программы на языке высокого уровня;
- счетчик времени исполнения программы.

Управление отладкой производится в интегрированной среде разработки AVR Studio. В качестве программы для работы микроконтроллера может использоваться отладочный объектный код в формате UBROF8 либо COFF, а также программа на ассемблере. Подключение эмулятора к компьютеру производится через COM-порт.

Одной из удобных функций, реализованных в ICE40/50, является возможность записи трас-

сы, то есть сохранение последовательности исполнения инструкций микроконтроллером. Анализ записанной трассы позволяет выявить участки кода, исполнение которых по тем или иным причинам не происходит (Code Coverage Analysis). Такой анализ очень полезен при отладке программ, содержащих несколько процедур обработки прерываний, особенно если вложенность прерываний не разрешена.

В ICE50 реализован режим диагностики узлов внутрисхемного эмулятора. Для осуществления диагностики необходимо установить тестовую панель на плату отладочного кристалла и в AVR Studio выбрать пункт меню Tools / ICE50 Selftest. После запуска процесса диагностирования (кнопка **Run**) последовательно проверяются все узлы и режимы внутрисхемного эмулятора. Если при этом определяется, что на плате управления содержится устаревший вариант внутреннего микрокода (firmware), то программа диагностики предложит автоматически обновить конфигурацию ПЛИС и памяти программ микроконтроллера. Также возможно произвести ручное обновление микрокода ICE50, выбрав пункт меню Tools / ICE50 Upgrade.

Внутрисхемный эмулятор ATJTAGICE mkII (ATJTAGICE2)

Альтернативой внутрисхемной эмуляции является режим фоновой отладки. В англоязычной литературе этот режим обозначается термином «On-Chip Debugging», или OCD.

Разница между внутрисхемной эмуляцией и фоновой отладкой заключается в следующем. Внутрисхемный эмулятор с той или иной степенью достоверности имитирует поведение целевого микроконтроллера в отлаживаемой системе, в то время как в режиме фоновой отладки программный код выполняется самим целевым микроконтроллером. Таким образом, достигается полное совпадение временных и электрических параметров системы в отладочном и штатном режимах.

Для поддержки режима «On-Chip Debugging» (OCD) блок управления отладкой должен быть частью самого отлаживаемого микроконтроллера. Управление фоновой отладкой в микроконтроллерах AVR осуществляется по JTAG-интерфейсу, совместимому со стандартом IEEE1149.1, либо однопроводному интерфейсу debugWIRE, являющемуся собственной разработкой компании Atmel.

Блок OCD с управлением по интерфейсу JTAG содержится в микроконтроллерах ATmega128/CAN128, ATmega16/32/64, ATmega162, ATmega165/325, ATmega169/329, ATmega3250/3290. Блок OCD с управлением по интерфейсу debugWIRE содержится в микроконтроллерах ATmega48/88/168, ATtiny13, ATtiny2313, ATtiny45.

Под управлением блока OCD в микроконтроллерах AVR производится исполнение программного кода и доступ ко всем регистрам микроконтроллера в режиме фоновой отладки.

Для работы микроконтроллера в режиме внутрисхемной отладки необходимо разрешить (по умолчанию запрещен) интерфейс JTAG либо debugWIRE. Для этого необходимо установить соответствующие FUSE-биты. При разрешении интерфейса debugWIRE

следует помнить, что интерфейс debugWIRE использует вывод /RESET микроконтроллера и при разрешении интерфейса стандартная функция вывода (сброс микроконтроллера) отключается, а значит, становится недоступной функция последовательного внутрисхемного программирования, так как вывод /RESET используется для ввода микроконтроллера в режим последовательного программирования. Перепрограммирование FUSE-битов микроконтроллера можно произвести по интерфейсу debugWIRE либо в режиме параллельного программирования.

Устройством, реализующим протокол управления, является ATJTAGICE mkII (ATJTAGICE2).



Рис. 13. Внешний вид внутрисхемного эмулятора ATJTAGICE mkII

Внутрисхемный эмулятор ATJTAGICE mkII подключается к персональному компьютеру по интерфейсам COM либо USB. При подключении по интерфейсу USB не требуется использование внешнего источника питания. Производить соединение эмулятора с компьютером и целевым микроконтроллером необходимо при выключенном питании всех трех устройств (при подключении по интерфейсу USB достаточно установить выключатель питания ATJTAGICE mkII в положение «Выключено»). В противном случае можно вывести из строя выходные каскады эмулятора.

При использовании интерфейса USB для подключения ATJTAGICE mkII к компьютеру передача команд производится существенно быстрее, чем при использовании COM-порта.

В качестве управляющей программы для работы ATJTAGICE mkII и OCD микроконтроллера используется интегрированная среда разработки AVR Studio. Более подробно использование AVR Studio для внутрисхемной отладки будет описано ниже. Также более подробную информацию по возможностям и использованию ATJTAGICE mkII можно найти во встроенном файле справки AVR Studio.

Кроме обеспечения режима фоновой отладки, эмулятор ATJTAGICE mkII может быть использован в качестве внутрисхемного программатора для микроконтроллеров AVR. Программирование может осуществляться по интерфейсам JTAG либо SPI. Для программирования, а также отладки по интерфейсу JTAG одновременно нескольких устройств нужно соединить устройства в дейзи-цепочку. После этого необходимо указать AVR Studio наличие нескольких микроконтроллеров в меню Tools / STK500/AVRISP/JTAGICE / STK500/AVRISP/JTAGICE / Advanced / JTAG Daisy Chain Settings.

Альтернативой описанным выше внутрисхемным эмуляторам и отладчику может служить интегрированный в AVR Studio программный симулятор, который производит эмуляцию программным способом.

Интегрированная среда разработки AVR Studio

AVR Studio — это интегрированная отладочная среда разработки приложений (IDE) для микроконтроллеров AVR компании Atmel. IDE AVR Studio содержит:

- средства создания и управления проектом;
- редактор кода на языке ассемблера;
- транслятор языка ассемблера (Atmel AVR macroassembler);
- отладчик (Debugger);
- программное обеспечение верхнего уровня для поддержки внутрисхемного программирования (In-System Programming, ISP) с использованием стандартных отладочных средств Atmel AVR.

AVR Studio распространяется бесплатно. Его последняя версия всегда доступна на сайте компании Atmel (<http://www.atmel.com/atmel/products/prod203.htm>). На момент написания статьи выпущена AVR Studio версии 4.10. Ожидается появление версии 4.11.

Работа с AVR Studio начинается с создания проекта. При создании проекта необходимо указать используемый микроконтроллер и платформу, на которой будет производиться отладка программы.

Написание программы производится в окне редактора текста программы. Для использования символических имен регистров специального назначения вместо их адресов необходимо подключить (директива `.include`) к проекту файл определения регистров специального назначения (например, `m16def.inc` для ATmega16). Включаемые файлы входят в прикладное программное обеспечение AVR Studio и при установке помещаются в папку `Appnotes` в директории установки AVR Studio. Примеры программ доступны в большом ко-

личестве в качестве приложений к руководствам по применению микроконтроллеров AVR (см. раздел Application Notes на сайте <http://www.atmel.ru/Disks/AVR%20Technical%20Library/index.html>).

Написание программы в AVR Studio производится на языке ассемблера. Система команд микроконтроллера описана в упомянутых выше руководствах по применению в документе **AVR Instruction Set** либо в файле справки, встроенном в AVR Studio (меню **Help / AVR Tools User Guide / AVR Assembler**), в котором содержатся достаточно подробные комментарии к каждой команде.

Последние версии AVR Studio содержат тестовую версию AVR-ассемблера второй версии, который в дополнение к стандартному ассемблеру поддерживает новые директивы ассемблера, Си-подобные директивы препроцессора и создание переменных определенного типа. Более подробную информацию можно найти в файле справки.

Перед трансляцией программы можно задать установку проекта (меню **Project / AVR Assembler Setup**), указать необходимый формат выходного файла. Там же возможно установить использование AVR-ассемблера версии 2. Так как вторая версия ассемблера проходит стадию тестирования, то по умолчанию он отключен. Если не требуется каких-либо особых настроек, то можно использовать установку по умолчанию.

В результате трансляции создается выходной файл в указанном формате. Если исходный ассемблерный текст содержал сегмент энергонезависимых данных (объявленный директивой `.eseg`), то при трансляции будет создан также файл с расширением `.eep`. Этот файл содержит данные для внутренней EEPROM микроконтроллера и имеет тот же формат, что и выходной файл.

Если в результате трансляции не выдается сообщений об ошибках, можно приступить к отладке проекта.

Отладчик AVR Studio поддерживает все типы микроконтроллеров AVR и имеет два режи-

ма работы: режим программной симуляции и режим управления различными типами внутрисхемных эмуляторов (In-Circuit Emulators) производства фирмы Atmel. Важно отметить, что интерфейс пользователя не изменяется в зависимости от выбранного режима отладки.

Отладочная среда поддерживает выполнение программ как в виде ассемблерного текста, так и в виде исходного текста языка C, загруженного в объектном коде.

Объектный файл может быть загружен в различных форматах, а именно:

- В формате **UBROF** (расширение файла `.d90`). Формат UBROF — это патентованный формат компании IAR. На текущий момент среда разработки AVR Studio поддерживает все версии формата до восьмой версии (UBROF5/6/7/8).
- В формате **ELF/DWARF** (расширение файла `.elf`). Формат разработан сообществом открытого программного обеспечения для поддержки бесплатно распространяемых компиляторов (GNU GCC, для микроконтроллеров AVR последние версии данного компилятора имеют название WinAVR). На текущий момент среда разработки AVR Studio поддерживает формат версии DWARF2.
- В формате **AVR COFF** (расширение файла `.cof`). Это открытый формат, введенный в среду разработки AVR Studio для поддержки сторонних разработчиков компиляторов. Отладочный код в данном формате может быть сформирован с использованием таких компиляторов, как, например, CodeVision AVR.
- В формате **Extended Intel HEX** (расширение `.hex`). Данный формат не приспособлен для отладки и используется для этих целей в крайнем случае.

После загрузки объектного кода производится выбор отладочной платформы и используемого микроконтроллера. При отладке с использованием внутрисхемных эмуляторов микроконтроллеры, поддержка которых не осуществляется, автоматически подсвечиваются серым. После загрузки проекта система готова к старту отладки.

Управление отладкой производится командами меню **DEBUG** либо соответствующими иконками на панели управления AVR Studio.

Пользователь может выполнять программу полностью в пошаговом режиме, трассируя блоки функций или выполняя программу до того места, где стоит курсор. В дополнение можно определять неограниченное число точек останова, каждая из которых может быть включена или выключена. Точки останова сохраняются между сессиями работы.

В AVR Studio для отладки программы предусмотрены две команды пошагового режима: **Step Over** и **Trace Into**. Разница между ними в том, что при выполнении команды **Step Over** выполнение подпрограмм происходит до полного окончания без отображения процесса выполнения. К командам шагового режима также относится команда **Auto Step**.

С помощью команд пошагового режима можно проследить изменения значений в переменных, в регистрах ввода-вывода, в памяти и регистровом файле. Для этого предназначен раз-

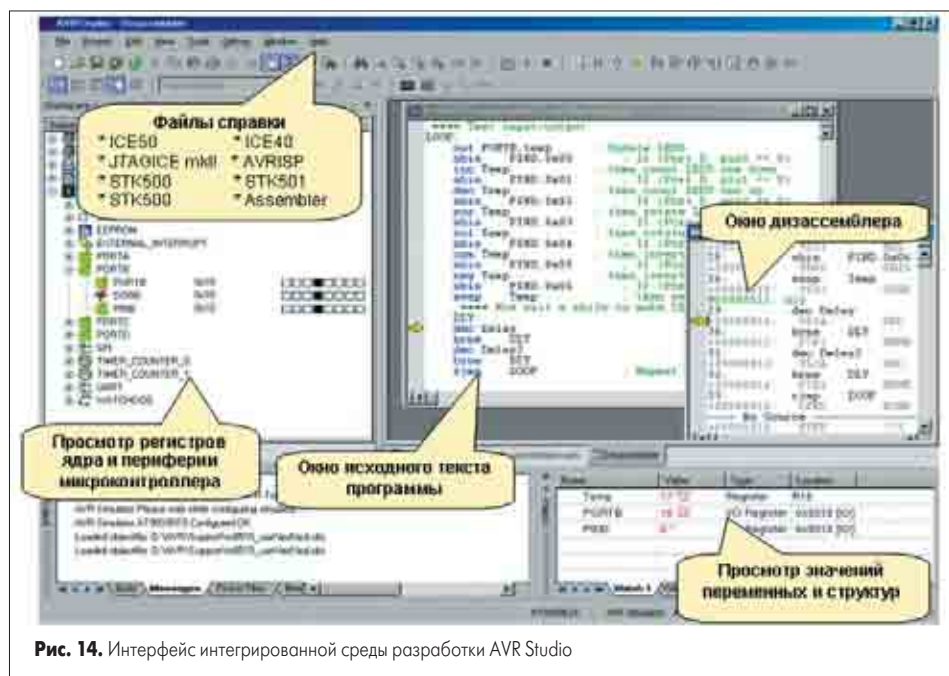


Рис. 14. Интерфейс интегрированной среды разработки AVR Studio

дел I/O рабочей области AVR Studio (см. рис. 14) и окно Watch (меню **Debug / Quickwatch**).

Интегрированная среда разработки AVR Studio также поддерживает просмотр (меню **View / Memory**) ячеек памяти программ, памяти данных, EEPROM и регистров портов ввода-вывода в ходе исполнения. Выпадающее меню диалогового окна позволяет выбрать один из четырех массивов ячеек памяти: Data, IO, Eeprom, Program Memory. Для одновременного просмотра нескольких областей окно **Memory** может быть открыто несколько раз. Информация в диалоговом окне может быть представлена в виде байтов или в виде слов в шестнадцатеричной системе счисления, а также в виде ASCII-символов.

В процессе отладки пользователь может инициализировать внутреннее ОЗУ или EEPROM микроконтроллера (например, данными, содержащимися в полученном при трансляции файле .eep) или сохранить содержимое ОЗУ и EEPROM в виде файлов в формате Intel Hex (меню **File / Up/Download Memory**).

Помимо шагового режима возможна отладка программы с использованием точек останова (меню **Breakpoints / Toggle Breakpoint**). Командой **Start Debugging** запускается исполнение программы. Программа будет выполняться до остановки пользователем или до обнаружения точки останова.

Работая с программным симулятором пакета AVR Studio, следует помнить, что он пока не поддерживает некоторые режимы работы микроконтроллеров AVR и их периферийные узлы:

- аналого-цифровой преобразователь;
- аналоговый компаратор;
- режим часов реального времени;
- режим пониженного энергопотребления (инструкция «sleep» интерпретируется программным симулятором как «пор»).

Возможно, в последующих версиях AVR Studio поддержка этих узлов и режимов будет реализована, но в настоящий момент при работе программы с аналоговой периферией необходимо производить ручной ввод данных в регистры результата.

Для наблюдения за работой программы можно открыть несколько окон, отображающих состояние различных узлов микроконтроллера (см. рис. 14). Окна открываются нажатием соответствующих кнопок на панели инструментов или при выборе соответствующего пункта меню **View**. Если в процессе выполнения программы в очередном цикле значение какого-либо регистра изменится, то этот регистр будет выделен красным цветом. При этом если в следующем цикле значение регистра останется прежним, то цветовое выделение будет снято. Такое же цветовое выделение реализовано в окнах устройств ввода-вывода, памяти и переменных.

Состояние встроенных периферийных устройств микроконтроллера, а также состояния программного счетчика, указателя стека, содержимого регистра статуса SREG и индексных регистров X, Y и Z отображено в окне I/O Window. В этом окне отражаются все функциональные блоки микроконтроллера. Любой блок может быть раскрыт нажатием на его зна-

чок. При раскрытии блока в окне отражаются адреса и состояния всех его регистров и отдельных, доступных для модификации, битов. Каждый доступный для модификации бит может быть установлен или сброшен как программой по ходу ее исполнения, так и пользователем вручную (указав курсором мыши нужный бит и щелкнув левой кнопкой мыши, пользователь может изменить значение бита на обратное), а в режиме программной симуляции это является способом имитации входного воздействия на микроконтроллер.

Другим способом задания входного воздействия на микроконтроллер в режиме симулятора является использование внешних файлов входных воздействий. Формат файла входного воздействия очень прост:

```
000000000:00
000000039:01
000000040:00
999999999:FF
```

Здесь значение, указанное после разделителя «:» — это шестнадцатеричное представление сигналов, воздействующих на порт микроконтроллера. Значение, указанное до разделителя — это десятичный номер цикла (с момента сброса микроконтроллера), в котором указанное воздействие поступает на вывод порта микроконтроллера. Файл входного воздействия должен заканчиваться строкой с заведомо большим номером цикла, в противном случае будет выдано сообщение об ошибке. Для подключения файла входного воздействия служит пункт меню **Debug / Options** в разделе **Stimuli and Logging**. В открывшемся окне нужно указать порт микроконтроллера, на который нужно подавать воздействие, и файл этого воздействия. Пользователь может создавать файлы воздействий, а также записывать изменения значений на выходах портов микроконтроллера в файл (формат этого файла тот же, что и у файла входных воздействий).

Для записи служит функция **Logging**. В открывшемся окне нужно указать порт микроконтроллера и имя файла для записи. Записываемый файл будет удаляться и создаваться вновь при каждом выполнении сброса микроконтроллера (**Debug / Reset**). Подключать файл входного воздействия или задавать имя файла для записи пользователь должен сам при каждом запуске симулятора.

Следует отметить, что AVR Studio имеет очень мощную встроенную документацию как по использованию AVR Studio, так и по использованию стандартных отладочных средств производства компании Atmel, а также по системе команд и всему, что касается программирования с использованием AVR Studio. Таким образом, использование встроенной справочной информации избавляет разработчика от накопления разобоченной информации, учитывая, что встроенная информация обновляется с выходом новых версий AVR Studio.

Как уже говорилось, кроме программного симулятора интегрированная среда разработки AVR Studio содержит программное обеспечение верхнего уровня для управления аппаратными средствами поддержки разработок, которое было описано выше.

Программирование на языках высокого уровня

Программное обеспечение для микроконтроллеров AVR, помимо ассемблера, может быть разработано на языке C, так как платформа AVR поддерживается продуктами сторонних разработчиков компиляторов языка C.

Наиболее мощным, с высокой оптимизацией генерируемого кода, является компилятор IAR Embedded Workbench производства компании IAR — разработчика системы команд для микроконтроллеров AVR. Ориентировочная стоимость данного компилятора \$3500.

В качестве альтернативы компилятору IAR широко используются компиляторы CodeVision и ImageCraft, которые являются менее мощными с точки зрения оптимизации кода, но более легкими в освоении. Ориентировочная стоимость данных компиляторов \$200–300.

Для тех, кто не готов платить за программное обеспечение, существует бесплатный компилятор WinAVR (ранее AVR GCC), который в совокупности с бесплатной графической средой VMLab обеспечит разработчика необходимыми средствами для разработки программы на языке C.

Для сторонников программирования на Паскале существует компилятор AVRco (<http://www.w.e-lab.de/AVRco/index.html>).

Выводы

Подводя итог, можно отметить, что микроконтроллеры AVR поддерживаются сбалансированным набором аппаратных и программных средств отладки. Немаловажным является тот факт, что появление новых микроконтроллеров всегда поддерживается как аппаратными, так и программными средствами для разработки с момента появления инженерных образцов.

Несмотря на то что отладочные средства производства компании Atmel доступны по цене, существует множество альтернативных аппаратных решений.

Наличие бесплатной интегрированной среды разработки AVR Studio и недорогих Си-компиляторов является существенным плюсом при выборе платформы для проектируемого прибора, так как программные средства разработки для AVR в совокупности с аппаратными средствами являются законченным аппаратно-программным комплексом для разработки, отладки и программирования любых микроконтроллеров AVR.

Литература

1. Водовозов А. М. Микроконтроллеры для систем автоматизации. Учебное пособие. Вологда: ВоГТУ. 2002.
2. Гребнев В. В. Микроконтроллеры семейства AVR фирмы Atmel. М.: ИП Радиософт. 2002.
3. AVR Technical Training. Atmel Corp. Norway. 2004.
4. AVR Software and Technical Library. Atmel Corp. Norway. December 2004.
5. www.atmel.com.