KiT#34(8).qxd 11/15/04 1:19 PM Page 114

Внутрисхемное программирование

Метод внутрисхемного (внутрисистемного) программирования (ISP — In-System Programmability) на сегодняшний день является основным способом программирования микроконтроллеров, ПЛИС и других приборов, в состав которых входит память Flash или EEPROM. В основе внутрисистемного программирования лежит идея использования какого-либо стандартного интерфейса, интегрированного в программируемую микросхему для последовательного занесения в нее программных или конфигурационных данных.

Дмитрий Гаврилюк

info@ineltek.ru

Типичная система ISP-программирования представляет собой комплекс средств, состоящий из персонального компьютера, интерфейсной платы и специализированного программного обеспечения. В зависимости от типа программируемой микросхемы могут использоваться различные средства программирования:

- специализированные программаторы, ориентированные на работу с определенными сериями микроконтроллеров, ПЛИС, DSP или CPLD (Complex Programmable Logic Device — комплексное программируемое логическое устройство);
- интерфейсные платы или так называемые «загрузочные кабели» для программирования с использованием персонального компьютера;
- тестовое оборудование (ATE — Automated Test Equipment), позволяющее помимо тестирования платы проводить также и программирование с использованием JTAG-интерфейса.

Основные преимущества использования ISP-технологии отражены в таблице 1.

Для понимания особенностей внутрисхемной отладки и программирования по ISP-технологии далее будут рассмотрены три основных интерфейса, используемых для этих целей:

Таблица 1. Достоинства технологии ISP

Момент использования	Возможности	Преимущества
Разработка и проектирование	Программирование выполняется при рабочем напряжении микросхемы	Не нужно использовать источник на 12 В или другое напряжение, не используемое при нормальной работе микросхемы
	Микроконтроллер или ПЛИС программируется непосредственно в устройстве	Снижается риск повреждения микросхемы от электростатики и механических воздействий
	Сборка устройства осуществляется до окончания написания готовой программы	Сокращается время выхода изделия на рынок, так как сборка устройства может проходить параллельно с написанием программы
Производство	Программное обеспечение может быть загружено в микросхему с другого устройства на плате или во время производства печатной платы	Упрощается и сокращается производственный цикл
	Микросхема, поддерживающая ISP-программирование, становится одним из многих устройств на плате и не требует к себе особого внимания	
	Микросхема может программироваться тестовой программой	Расширяются возможности тестирования и внутрисхемной отладки
Эксплуатация	Микросхема перепрограммируется непосредственно в составе работающего оборудования	Возрастают функциональные возможности устройства и привлекательность для потребителя

- JTAG/IEEE 1149.1 — 4-проводной интерфейс, обеспечивающий тестирование и программирование микросхем, смонтированных на печатной плате;
- SPI — интерфейс, позволяющий реализовать высокоскоростной полнодуплексный синхронный обмен данными по 3-проводной шине и программирование в последовательном режиме;
- DebugWIRE — однопроводной интерфейс, используемый в новейших микроконтроллерах ATMEL, ориентированный на программирование встроенной памяти и отладку в составе готового устройства.

JTAG

Бурное развитие электронной техники и связанный с этим рост степени интеграции микросхем к концу 70-х годов привели к необходимости разработки механизма тестирования, способного обеспечить полноценное тестирование сложных и функционально насыщенных печатных плат. В 1980 году группой JTAG (Joint Test Automation Group — объединенная рабочая группа по автоматизации тестирования) была начата разработка спецификации граничного (периферийного) сканирования, стандартизованная в 1990 году институтом IEEE. Получившийся стандарт известен как IEEE Std.1149.1–1990. В дальнейшем было выполнено несколько пересмотров стандарта. В 1994 году в него было добавлено описание языка BSDL (Boundary-Scan Description Language — язык описания устройств, поддерживающих периферийное сканирование). С этого момента стандарт был принят ведущими электронными компаниями во всем мире. В настоящее время многие небольшие фирмы, которые по экономическим причинам не могут воспользоваться дорогими внутрисхемными тестерами, используют технологию граничного сканирования. Последний раз стандарт был пересмотрен в 2001 году [1]. Текст стандарта можно приобрести в IEEE (http://shop.ieee.org).

Архитектура граничного сканирования (Boundary-Scan Test) обеспечивает тестирование соединений между интегральными схемами на плате без ис-

KiT#34(8).qxd 11/15/04 1:19 PM Page 115

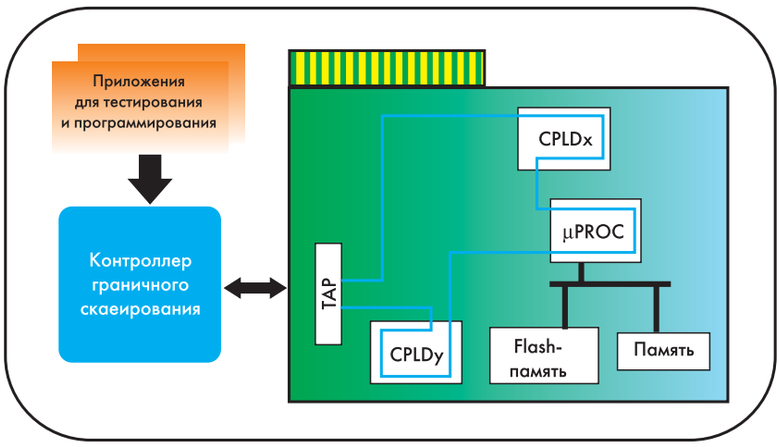

Рис. 1. Внутренняя организация микросхемы с поддержкой JTAG

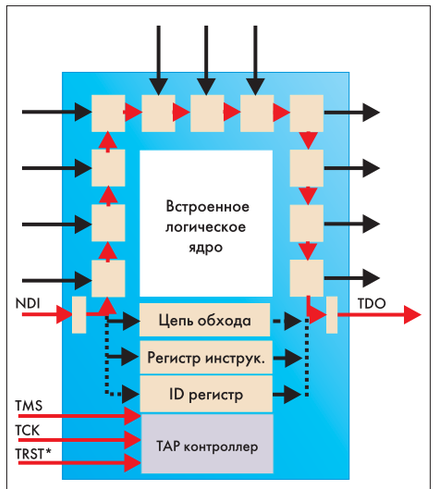

Рис. 2. Возможности JTAG-интерфейса

пользования физических тестовых пробников. Предполагается, что тестируемые микросхемы имеют поддержку JTAG-интерфейса. Стандарт IEEE 1149.1 определяет 4-проводной (опционально 5-проводной) последовательный интерфейс доступа к микропроцессорам, DSP, CPLD. Любая совместимая с JTAG-интерфейсом микросхема (рис. 1) имеет регистры сдвига и модуль поддержки функций граничного сканирования. Ячейки регистра сдвига (BS-регистра) располагаются непосредственно между внешними выводами и функциональным ядром микросхемы.

Данные, поступающие в кристалл через вывод TDI, сохраняются в регистре инструкций или в одном из регистров данных. Последовательные данные выводятся из микросхемы через вывод TDO. Логический модуль интерфейса тактируется сигналом на входе TCK, не зависимом от системной тактовой частоты. Сигналы на входе TMS управляют состоянием порта TAP (Test Access Port). Вывод TRST является необязательным и служит для сброса логики интерфейса JTAG.

Множество микросхем, совместимых со стандартом IEEE 1149.1, могут последовательно соединяться друг за другом на печатной плате, формируя цепь граничного сканирования (рис. 2).

С ее помощью появляется возможность получать информацию о состоянии каждого вывода каждой микросхемы, являющейся частью общей цепи сканирования через последовательный TAP-интерфейс. При обычной работе устройства JTAG-интерфейс не оказывает никакого влияния на его работу.

При тестировании или внутрисхемном программировании с помощью JTAG-интерфейса можно воздействовать на логическое ядро микросхемы, выдавать логические сигналы на выводы микросхемы, считывать выходные сигналы от внутренней логики или значение логических уровней на выводах, впаянных в печатную плату.

Технология граничного сканирования позволяет испытывать плату на наличие структурных производственных ошибок и выполнять ISP-программирование почти всех устройств на плате, независимо от типа и размера встроенной памяти, через стандартный JTAG TAP-интерфейс. В результате происходит значительное сокращение необходимых точек тестирования на плате. Это приносит много выгод: упрощается разводка печатной платы, снижаются затраты и время тестирования [2], сокращается время выхода готового изделия на рынок.

Файлы, содержащие описание поддерживаемых инструкций и конфигурационных параметров регистров сдвига JTAG-совместимых микросхем на языке BSDL, бесплатно доступны на сайтах фирм-производителей.

SPI

Интерфейс SPI (Serial Peripheral Interface — последовательный периферийный интерфейс) обеспечивает высокоскоростную передачу информации в синхронном режиме между ведущим и ведомыми устройствами. На рис. 3 показано включение ведущего и ведомого устройств с использованием SPI.

Ведущее устройство содержит 8-разрядный регистр сдвига и генератор синхроимпульсов. Инициирование цикла обмена данными происходит, когда ведущий устанавливает на выводе $\overline{SS}$ ведомого устройства лог. «0». Ведущее устройство генерирует необходимую тактовую последовательность на выводе SCK для обмена данными между сдвиговыми регистрами устройств. Данные побитно сдвигаются от ведущего к ведомому по линии MOSI (Master Out — Slave In) и от ведомого к ведущему по линии MISO (Master In — Slave Out). В итоге содержимое регистра сдвига ведущего оказывается в регистре сдвига ведомого и наоборот. После передачи каждого пакета данных ведущий синхронизируется с ведомым установкой лог. «1» на линии $\overline{SS}$.

Следует учитывать, что если устройство сконфигурировано как ведущее, модуль SPI-интерфейса автоматически не управляет линией $\overline{SS}$. Передача нового байта данных происходит сразу же после его записи в регистр сдвига. При необходимости использования линии $\overline{SS}$ управлять ее состоянием должно программное обеспечение пользователя.

В AVR-микроконтроллере ATmega8 фирмы Atmel модуль SPI-интерфейса обладает следующими возможностями [3]:

- полнодуплексная 3-проводная синхронная передача данных;
- работа в режиме ведущего или ведомого;
- возможность определения порядка передачи: первым передается либо младший, либо старший бит;
- семь программируемых величин скорости передачи;
- флаг прерывания окончания передачи;
- флаг защиты от коллизий записи;
- система «пробуждения» из режима Idle;
- максимальная тактовая частота интерфейса в два раза ниже тактовой частоты процессорного ядра.

С помощью SPI легко организуется внутрисхемное программирование микроконтроллеров Atmel (рис. 4).

Для этого используется три линии интерфейса: MISO, MOSI и SCK. Вывод $\overline{RESET}$ подключается к «земле» у AVR-серии или к положительному источнику — у серии AT89Sx. В этом режиме обеспечивается программирование (запись и чтение) памяти программ и данных, а также конфигурационных битов кристалла.

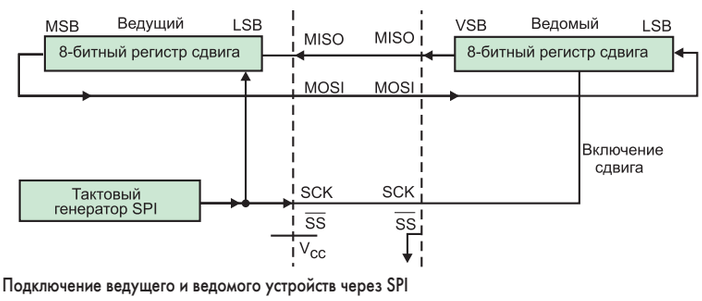

Рис. 3. Подключение ведущего и ведомого устройств через SPI

KiT#34(8).qxd 11/15/04 1:19 PM Page 116

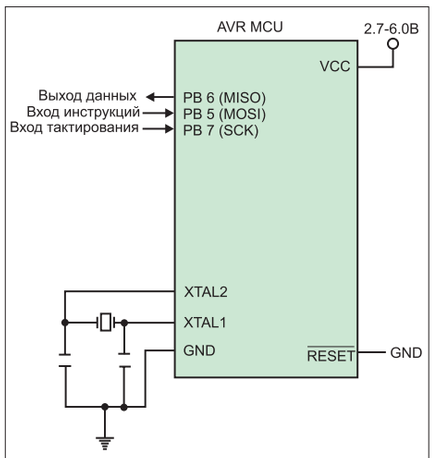

Рис. 4. Внутрисхемное программирование микроконтроллеров Atmel через SPI-интерфейс

DebugWIRE

В мае 2003 года Atmel анонсировала интегрированный в устройства ATtiny13 [4] и ATtiny2313 интерфейс DebugWIRE (One-Wire Debug Interface — однопроводной отладочный интерфейс). Он представляет собой встроенный в кристалл механизм отладки на основе одной двунаправленной линии передачи информации, позволяющий управлять программным потоком, выполнять AVR-инструкции ядра микроконтроллера и осуществлять программирование интегрированной энергонезависимой памяти.

Основные возможности интерфейса DebugWIRE:

- комплексное управление программным потоком;
- эмулирование всех функций кристалла, включая цифровые и аналоговые, за исключением вывода RESET;
- работа в реальном времени;
- поддержка символьной отладки (на основе исходных кодов ассемблера, C или других языков высокого уровня);
- неограниченное число программных точек останова;
- работа в «теневом» режиме (интерфейс не препятствует работе основной программы);
- электрические характеристики идентичны реальному устройству;
- автоматическое конфигурирование системы;
- высокая скорость работы;
- программирование энергонезависимой памяти.

Режим отладки в устройстве активируется установкой бита DWEN (debugWIRE Enable) и сброса Lock-битов микроконтроллера. Физическое подключение эмулятора интерфейса к микроконтроллеру производится через вывод $\overline{\text{RESET}}$, который работает как двунаправленный вывод с открытым стоком с подтягивающим к шине питания резистором. На рис. 5 показана схема подключения отлаживаемого микроконтроллера в режиме debugWIRE к эмулятору.

Тактовая частота определяется установками битов CKSEL и не зависит от debugWIRE.

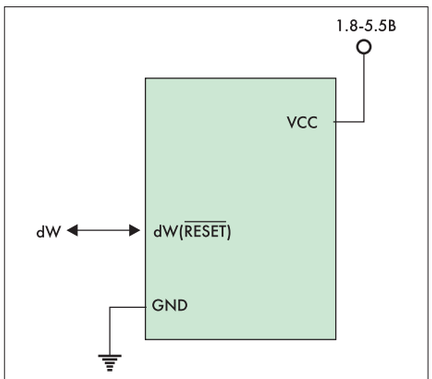

Рис. 5. Подключение микроконтроллера к интерфейсу DebugWIRE

Когда разрабатывается система, в которой будет использоваться debugWIRE, необходимо соблюдать следующие требования:

- если используется «подтягивающий» резистор на выводе dW($\overline{\text{RESET}}$), величина его сопротивления должна лежать в диапазоне от 10 до 20 кОм;
- соединение вывода RESET напрямую с источником питания V_{cc} недопустимо;
- конденсаторы, подключенные к выводу RESET, должны быть отключены при использовании debugWIRE;
- все внешние источники сброса микроконтроллера должны быть отключены.

Поддержка программных точек останова в debugWIRE реализуется при работе с интегрированной средой разработки AVR Studio. При установке точки останова в исходной программе в AVR Studio в программную память микроконтроллера будет вставлена инструкция BREAK. Инструкция, замещенная инструкцией BREAK, будет сохранена. Когда выполнение программы будет продолжено, сначала будет выполнена инструкция, сохраненная из памяти программ. Команду останова можно вставить вручную, поместив инструкцию BREAK в программную память микроконтроллера. Следует учитывать, что Flash-память микроконтроллера перепрограммируется при каждом изменении точки останова. Этот процесс автоматически выполняет AVR Studio через интерфейс debugWIRE. Использование точек останова уменьшает ресурс Flash-памяти. Устройства, применявшиеся в отладке системы, не должны передаваться конечным пользователям.

Использование интерфейса debugWIRE накладывает некоторые ограничения на устройство, в котором он применяется. Коммуникационный вывод интерфейса dW физически совмещен с выводом внешнего сброса. Поэтому источники внешнего сброса не поддерживаются при работе debugWIRE.

Система debugWIRE точно эмулирует все входные и выходные функции на полной скорости, когда выполняется программа в CPU. Если процессорное ядро остановлено, необходимо проявлять осторожность при доступе к некоторым регистрам ввода-вывода через отладчик AVR Studio.

При установке бита DWEN (активации debugWIRE), некоторые части системы тактирования остаются в активном состоянии при переходе в режим «sleep», что приводит к увеличению потребляемой мощности. Поэтому необходимо отключать DWEN, когда debugWIRE не используется. При работе с debugWIRE микроконтроллер использует 8-разрядный регистр DWDR, образующий коммуникационный канал между исполняемой в микроконтроллере программой и отладчиком. Этот регистр доступен только интерфейсу debugWIRE и не может быть использован как регистр общего назначения при нормальной работе.

Дальнейшее развитие идея внутрисхемного программирования получила в 8-разрядной серии AVR микроконтроллеров Atmel. В ней реализован механизм самопрограммирования Flash-памяти программ через интегрированные интерфейсы SPI, UART или TWI. Все операции по программированию и перепрограммированию процессорное ядро выполняет по программе, находящейся в загрузочной области программной памяти. Когда происходит самопрограммирование, данные считываются в оперативную память через внешний интерфейс и затем записываются в программную память блоками по 10, 128 или 256 байт. Особенности такого подхода состоят в следующем:

- возможно перепрограммирование готового изделия непосредственно в условиях его применения;
- снижаются затраты, поскольку нет необходимости использовать дополнительные компоненты;
- программирование может выполняться значительно быстрее, поскольку каждый сектор памяти стирается и программируется индивидуально;
- перепрограммирование осуществляется через различные периферийные интерфейсы;
- работа в режиме самопрограммирования осуществляется в полном температурном диапазоне и при любом допустимом напряжении питания.

Комплексы внутрисхемного программирования

Практическое применение ISP-технологии возможно с использованием аппаратно-программных комплексов самых разных производителей. Наиболее развитым направлением в этой сфере является производство разнообразных модулей и программных средств для решений на основе технологии JTAG/IEEE 1149.1. Гамма предлагаемых устройств включает контроллеры, подключаемые к портам ПК, а также устанавливаемые в слоты ISA, PCI, USB, PXI. Программное обеспечение этих приборов функционирует на платформе Windows (некоторые версии адаптированы для UNIX) и позволяет проводить тестирование, отладку модулей и перепрограммирование пользовательских микросхем на плате через стандартный TAP-интерфейс. Оборудование для работы с технологией граничного сканирования производят:

1. Corelis Inc., США (www.corelis.com).
2. JTAG Technologies, Нидерланды (www.jtag.com).

KiT#34(8).qxd 11/15/04 1:19 PM Page 117

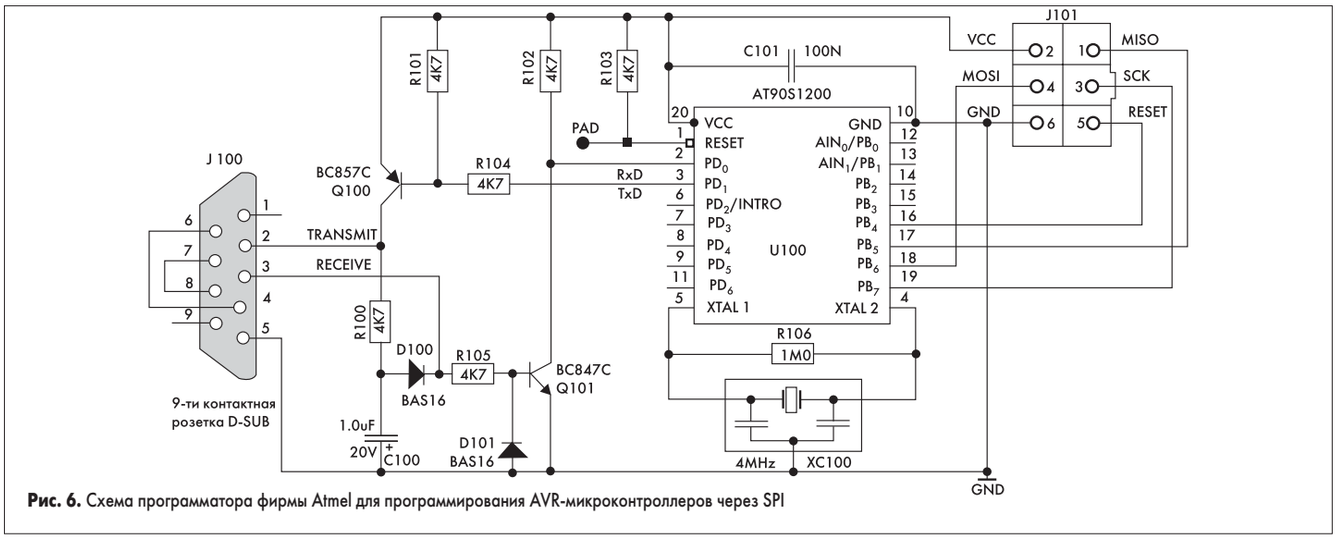

Рис. 6. Схема программатора фирмы Atmel для программирования AVR-микроконтроллеров через SPI

3. GOEPEL Electronic, Германия (www.goepel.com).
4. ASSET InterTech Inc., США (www.asset-intertech.com).
5. XJTAG Limited, Великобритания (www.xjtag.com).
6. Hewlett-Packard, США (www.hp.com).
7. Teradyne, США (www.teradyne.com).

Важно отметить, что профессиональное оборудование для граничного сканирования имеет достаточно высокую стоимость. Для оценки возможностей JTAG-технологии можно воспользоваться бесплатным программным обеспечением Дмитрия Кузнецова VEC4JTAG, которое доступно по адресу http://www.orc.ru/~dkuzn/index.htm. Пакет предназначен для получения доступа к средствам контроля и управления схемами ввода-вывода BS-чипов через порт JTAG при помощи кабеля ByteBlaster фирмы Altera. Пакет включает в себя:

- утилиту для трансляции текстового файла назначений векторов и параметров цепи JTAG в более удобную форму — байт-код;
- библиотеку для обеспечения доступа к назначенным векторам;
- файлы примеров.

Файл назначений векторов (с расширением .V4J) содержит список определений векторов в качестве объектов доступа к BS, а также определяет конфигурацию цепи JTAG. Вектор доступа к BS-чипам определяет совокупность (из одного или нескольких) выводов микросхем, объединенных в цепи JTAG.

Библиотека V4J поддерживается в следующих операционных системах:

- DOS;
- Windows 9x (в DOS-сесии).

Аппаратные требования:

- свободный LPT-порт;
- загрузочный кабель Altera ByteBlaster (его конструкция достаточно проста для самостоятельного повторения и бесплатно доступна на сайте www.altera.com).

Фирма BLIC (www.blic.ru) предлагает бесплатную программу PasByteBlaster, которая работает в среде Windows 9x и при использовании производимого фирмой загрузочного кабеля BLIC ByteBlaster обеспечивает следующие возможности:

- внутрисистемное программирование микроконтроллеров, поддерживающих программирование через SPI-интерфейс, например, фирмы Atmel;
- ISP-программирование микросхем серий MAX9000 и MAX7000A, S фирмы Altera через LPT-порт в режиме JTAG-интерфейса;
- конфигурирование микросхем серии FLEX 10K, KA, KE; FLEX 8000 и FLEX6000 фирмы Altera в пассивном последовательном режиме PS;
- поддерживается программирование микросхем на разные напряжения: 3,3 и 5 В;
- поддерживается загрузка данных непосредственно из среды программирования Altera MAX+PLUS II.

Бесплатный программный продукт PonyProg от Claudio Lanconelli (www.LancOS.com) ориентирован на программирование в последовательном режиме большого количества перепрограммируемых микросхем памяти и микроконтроллеров различных фирм: Atmel, Microchip и пр. Версия PonyProg2000 V2.06a поддерживает следующие семейства:

- I²C 8- и 16-разрядные EEPROM серий 24x, и AT17x;
- MicroWire8 и MicroWire16 EEPROM серии 93x;
- SPI EEPROM серии 25x;
- микроконтроллеры Atmel AVR-серии и AT89Sx;
- микроконтроллеры Microchip серий PIC12 и PIC16;
- ImBUS EEPROM MDA2062, NVM3065;
- SDE2506 EEPROM;
- S2430, X2444, X2445.

Конструктивно программатор PonyProg состоит из базового модуля, подключаемого к последовательному порту компьютера, и набора адаптеров для различных интерфейсов: SPI, I²C, MicroWire и ICSP (для поддержки PIC Microchip). Схема программатора очень проста и может быть реализована на доступных компонентах, ее версия 2.2 свободно доступна на сайте www.LancOS.com.

Корпорация Atmel производит большой ассортимент средств ISP-программирования и отладки для своих изделий. Для микроконтроллеров AVR-серии на сайте фирмы (www.atmel.com) бесплатно предлагается интегрированная среда разработки AVR Studio, работающая в среде Windows 98/2000/XP. С ее помощью возможно программирование и отладка в исходных кодах на ассемблере, а также внутрисхемное программирование микроконтроллеров с помощью программаторов JTAG, ICE, STK500, ICE50 и AVR-ISP (работает через SPI).

Программатор AVR-ISP можно изготовить самостоятельно, на основе информации, содержащейся в Application Note «AVR910: In-System Programming» (находится на www.atmel.com). Схема этого программатора представлена на рис. 6.

Он обеспечивает согласование COM-порта персонального компьютера с SPI-портом микроконтроллера. Программатор питается от того же источника, что и программируемый микроконтроллер. Когда программирование не происходит, все выводы программатора, подключенные к SPI-порту микроконтроллера, переводятся в третье состояние, благодаря чему программатор может быть постоянно подключен к плате с микроконтроллером.

В целом, использование бесплатной программы AVR Studio и самостоятельно изготовленный вариант программатора AVR-ISP представляется идеальным решением для разработки и программирования устройств на базе линейки AVR-микроконтроллеров фирмы Atmel без каких-либо дополнительных затрат.

Литература

1. 1149.1-2001 IEEE Standard Test Access Port and Boundary-Scan Architecture 2001; ISBN 0-7381-2945-3, IEEE Product No. SS94949-TBR. IEEE Standard No.: 1149.1-2001. www.ieee.org.
2. Texas Instruments, Application Report SCTA029: JTAG/IEEE 1149.1 Design Considerations, Advanced System Logic Products, 1996. www.ti.com.
3. ATMEL Data Sheet 8-bit AVR with 8K Bytes In-System Programmable Flash ATmega8, Rev.2486J-AVR-02/03. www.atmel.com.
4. ATMEL Data Sheet 8-bit AVR Microcontroller with 1K Bytes In-System Programmable Flash ATtiny13, Rev.2535A-AVR-06/03. www.atmel.com.