

# ARM-МИКРОКОНТРОЛЛЕРЫ АТМЕЛ: ПРАКТИЧЕСКИЕ АСПЕКТЫ ПРИМЕНЕНИЯ

## Часть 1. ARM7

**Николай Королев**, руководитель инженерного центра, ООО «АРГУССОФТ Компани»  
**Антон Шабынин**, инженер, ООО «АРГУССОФТ Компани»

Корпорация ATMEL выпускает широкую линейку микроконтроллеров на базе ядер ARM7 и ARM9. Микроконтроллеры отличаются набором периферийных модулей и, соответственно, типом корпуса. Наличие расширенной периферии существенно увеличивает возможности микроконтроллеров, но, с другой стороны, заставляет обращать повышенное внимание на процесс инициализации кристалла, а также грамотное использование отладочных средств. В первой части статьи рассмотрена работа с контроллером на ядре ARM7.

### ВВЕДЕНИЕ

Интерфейсы микроконтроллера характеризуют возможности его «общения» с внешним миром. Поэтому разработчик, выбирая контроллер для нового проекта, внимательно изучает не только возможности ядра, но и наличие того или иного типа интерфейса на кристалле.

В таблице 1 представлены типы интерфейсов, которыми оснащены контроллеры серии AT91SAM7 и AT91SAM9. Из таблицы видно, что практически все контроллеры SAM7 имеют встроенную флэш-память, а контроллеры SAM9 отличает наличие внешней шины данных, причем микросхема AT91SAM9263 содержит две внешние шины, что естественно влечет за собой увеличение числа выводов корпуса.

Корпорация ATMEL, учитывая потребности рынка, разрабатывает микросхемы с расширенной периферией, более того, периферия микроконтроллера может быть «на лету» переинициализирована. Например, модуль USART можно запрограммировать на работу в обычном асинхронном режиме с использованием только линий TXD и RXD, а можно инициализировать его как «полноценный» 9-сигнальный интерфейс RS-232 или как синхронный интерфейс со скоростью обмена до 25 Мбит/с. Если же используется протокол RS-485 в полудуплексном режиме, можно запрограммировать дополнительный вывод микросхемы для аппаратного управления направлением передачи драйвера интерфейса. На рисунке 1 представлены основные параметры микросхем семейства

AT91SAM7S. Это семейство можно считать базовым, потому что остальные семейства содержат тот же набор периферийных модулей, но дополнены некоторыми интерфейсами, например, CAN или Ethernet.

В целях повышения производительности микроконтроллера многие периферийные модули оснащены собственными контроллерами прямого доступа к памяти (ПДП) (см. рис. 2). Следует отметить, что во время работы контроллера ПДП не происходит блокирования доступа ядра контроллера к системной шине и оперативной памяти.

Для обеспечения работы периферийного модуля в дуплексном режиме контроллер содержит два набора регистров — приемный и передающий. В состав контроллера ПДП входит счетчик адреса, кото-



Рис. 1. Основные характеристики контроллеров семейства SAM7

Таблица 1 Интерфейсы контроллеров SAM7 и SAM9

	USB-устройство	Флэш-память	CAN	Ethernet	Внешняя шина (ЕВІ)	USB-хост	Камера	ЖКИ
SAM7S	•	•						
SAM7A3	•	•	•					
SAM7X	•	•	•	•				
SAM7SE	•	•			•			
SAM7L		•						•
SAM9260	•			•	•	•	•	
SAM9XE	•	•		•	•	•	•	
SAM9RL	480 Мбит/с				•			•
SAM9261	•				•	•		•
SAM9263	•		•	•	• (x2)	•	•	•

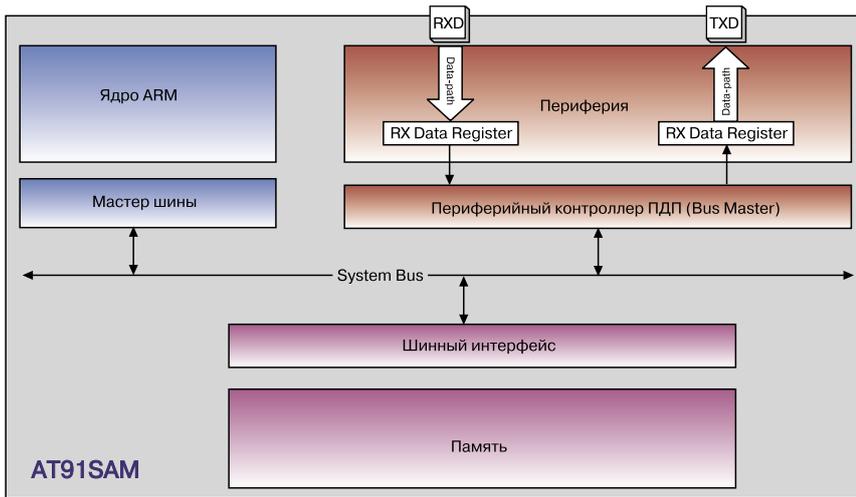


Рис. 2. Структура контроллера ПДП

рый программируется на прием/передачу пакета данных. Если длина пакета превышает разрядность счетчика адреса, необходима повторная инициализация, что может привести к нежелательной паузе в выходном потоке данных и к потере данных во входном потоке. Для предотвращения такой ситуации контроллер содержит комплект регистров-указателей адреса/счетчик адреса/буфер данных для текущего пакета данных и такой же комплект для последующего пакета. На рисунке 3 представлена структурная схема приемной части контроллера ПДП. Передающая часть контроллера устроена идентично.

### СРЕДСТВА РАЗРАБОТКИ

#### Программные средства

На сегодняшний день все программные средства разработки можно разделить на платные и бесплатные продукты с открытым кодом (см. табл. 2). Функциональные возможности у всех полноценных продуктов примерно одинаковые.

#### IAR

Компания IAR предлагает своим пользователям несколько версий среды разработки, которые значительно отличаются функциональными возможностями и сервисной поддержкой. В том числе есть бесплатная

версия IAR Kickstart, ограниченная на максимальную длину выходного бинарного файла – до 32 Кбайт. В такое ограничение помещаются небольшие пользовательские программы, активно использующие стандартные библиотеки языка C/C++. Пакет IAR Kickstart предлагает совершенно привычную IDE со ставшим классическим интерфейсом. Вполне приличный редактор, хорошие возможности по отладке с использованием JTAG-эмулятора: просмотр и правка регистров, памяти, watch, использование условных точек остановок. Есть поддержка как эмуляторов от Segger, так и более дешевых, типа Wiggler, через внешний RDI-сервер. Словом – все, что нужно для полноценной отладки пользовательской программы. Оптимизирующий компилятор IAR позволяет повысить скорость выполнения кода иногда в несколько раз.

#### GCC

Альтернативой коммерческой среде от IAR является открытая и бесплатная среда разработки на платформе Eclipse и компиляторе GNU GCC. От IAR она отличается более богатыми возможностями редактора кода, отсутствием каких-либо ограничений на размер файлов. К недостаткам этого решения необходимо отнести относительно более сложную процедуру настройки среды перед первым использованием. Набор поддерживаемых JTAG-эмуляторов примерно одинаковый. После того, как компания Segger выпустила GDB-сервер для своих эмуляторов, в среде Eclipse + GCC появилась возможность использования всей линейки эмуляторов от Segger. Сложность настройки деталей компиляции проектов (линковка, подключение библиотек, и т.д.) примерно одинаковая. Оптимизация в GCC незначительно уступает IAR.

#### Что выбрать?

Отличий у этих продуктов много, поэтому все зависит от возможностей по покупке коммерческой среды и предыдущего опыта работы. Работа в среде IAR под разными архитектурами практически не отличается. С Eclipse, на первый взгляд, все несколько сложнее, но к интуитивно понятному интерфейсу привыкнуть нетрудно. Диалекты расширений языка Си у IAR и GCC разные. В основном это касается работы с памятью и прерываниями, поэтому простого переноса проекта из одной среды в другую не получится. Однако, чисто алгоритмический код в рамках стандарта переносится без каких-либо трудностей.

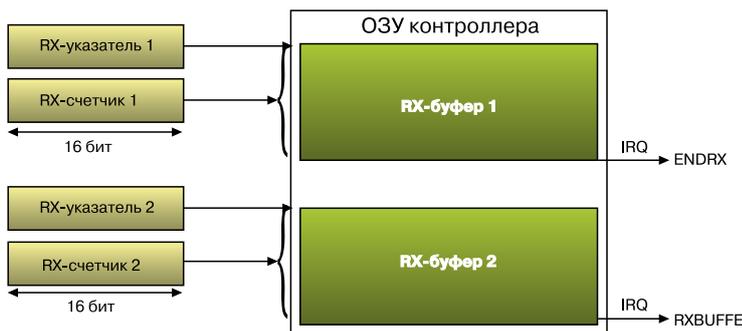


Рис. 3. Контроллер ПДП (приемная часть)

Таблица 2. Сравнительные характеристики компиляторов

	Симулятор	Поддержка эмуляторов	Языки и стандарты
IAR	+	RDI, Wiggler, AT91SAM-ICE, J-LINK ...	ANSI CExtended ANSI CExtended C++ (частичная поддержка C++)
GNU GCC + Eclipse	-	RDI, Wiggler, AT91SAM-ICE, J-LINK ...	ANSI CC99 частичная поддержка C++
ICC Imagecraft	-	-	ANSI C

Для своих микроконтроллеров Atmel предлагает пользователям библиотеку для работы с периферией. Существует версия библиотеки IAR и для работы с GCC.

### Как всем этим пользоваться?

Пример проекта AT91SAM7S-Interrupt\_SAM7S находится в каталоге ARM\examples\SAM7S256 с примерами, входящими в состав установки IAR. Для рассмотрения был выбран МК AT91SAM7S256, установленный на плату AS-SAM7S-256 производства АРГУССОФТ. Пример показывает работу с прерываниями и портом USART0.

### Установка IAR Kickstart

Среда разработки (см. рис. 4) доступна в интернете на сайте [www.iar.com](http://www.iar.com). Для загрузки и получения лицензионного номера требуется бесплатная регистрация. После загрузки и установки среды разработки в директории с IDE можно найти примеры исходных кодов. Для этого надо открыть файл проекта examples\SAM7S256\AT91SAM7S-Interrupt\_SAM7S\Compile\BasicInterrupt\_SAM7S.eww.

Стандартный пример имеет три набора настроек: для отладки из ОЗУ, для загрузки и отладки во флэш-память с помощью AT91SAMICE и для компиляции бинарного файла. Микроконтроллеры серии AT91SAM7, имеют встроенный в ПЗУ загрузчик, который позволяет программировать МК по интерфейсу USB или RS-232.

### Установка AT91 ISP

Для использования этой возможности (программирования по USB или RS-232) требуется установить пакет AT91 ISP. В версию 1.10 (текущая на момент написания статьи) входят две программы: SAM-BA (SAM Boot Assistant – помощник загрузки) и SAM-PROG. SAM-BA позволяет программировать и настраивать МК семейств AT91SAM7/AT91SAM9 по интерфейсу RS232 или USB. Для работы по USB необходим кварцевый резонатор 18, 432 МГц и соответствующая цепь ФАПЧ. Таблица с частотами кварцевых резонаторов для UART имеется в файле документации на микросхему. Далее необходимо, чтобы в микроконтроллере выполнялась программа-загрузчик. Для этого в микросхемах семейства SAM7S после стирания микросхемы замыканием ножки ERASE нужно скопировать загрузчик из ПЗУ во флэш-память, замкнув ножку TEST (более подробно этот алгоритм описан в документации на

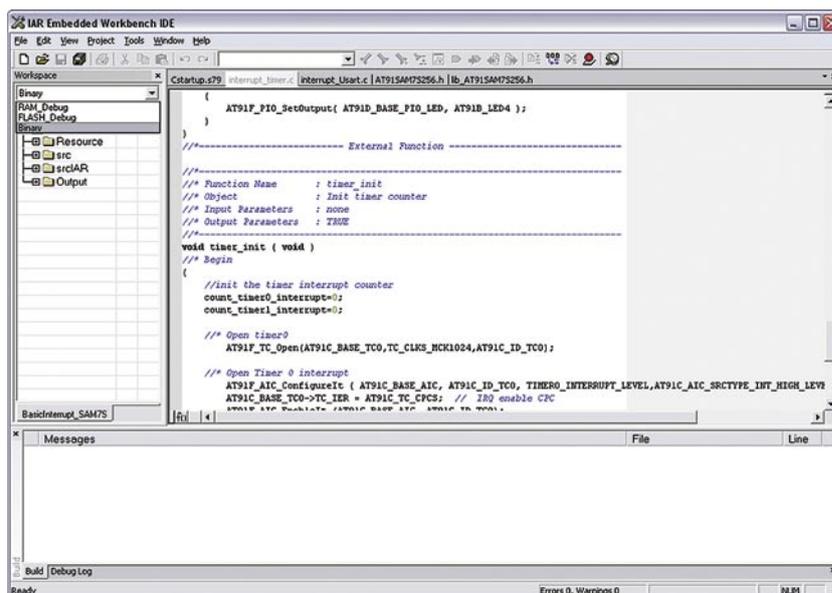


Рис. 4. Среда IAR

SAM7S). Микросхемы SAM7X могут сразу загружаться из ПЗУ. После подключения к компьютеру микросхема готова к программированию. При запуске SAM-BA необходимо правильно указать способ соединения и тип программируемой микросхемы. При первом подключении платы к USB Windows установит для нее драйверы из пакета AT91 ISP.

### Компиляция.

#### Загрузка через SAM-BA

Для загрузки через SAM-BA требуется файл в двоичном формате (bin, raw-binary в IAR). Для этого нужно выбрать binary вместо RAM\_Debug в настройках Workspace, после этого можно компилировать прошивку командой Project – Make. В случае удачной компиляции в директории AT91SAM7S-Interrupt\_SAM7S\Compile\Binary\Exe появится файл BasicInterrupts\_SAM7S.bin, пригодный для загрузки с помощью SAM-BA. После запуска SAM-BA необходимо выбрать файл для загрузки в микросхему и нажать кнопку Send File (см. рис. 5). Процесс программирования занимает несколько секунд. Есть возможность проверить целостность прошивки после программирования.

На этом работа с SAM-BA завершена, следует закрыть приложение и перезагрузить контроллер. При подключении по USB требуется переподключить плату, поскольку определение USB-устройства происходит только при их подключении.

Если подключить UART0 (на плате AS-SAM7S-256 маркирован RS-232; не путать с портом DEBUG!) к порту

RS-232 компьютера, то в стандартной терминальной программе можно увидеть сообщение, которое пишет плата при загрузке.

### JTAG-эмулятор

Каждый раз стирать флэш-память, записывать загрузчик, программировать через SAM-BA – при интенсивной отладке это утомляет и отнимает много времени. В микроконтроллерах семейства SAM7 есть специальный модуль для отладки на кристалле. С его помощью можно устанавливать точки останова (breakpoint), программировать внутреннюю память, иметь доступ ко всем регистрам ядра и периферии. С его помощью возможно, например, установить точки останова на доступ к данным и увидеть, при выполнении какой команды происходит переполнение стека. Друг от друга эмуляторы разных типов отличаются в основном скоростью работы, программной поддержкой, списком поддерживаемых микросхем. Для работы с микроконтроллерами SAM7 Atmel предлагает JTAG-эмулятор AT91SAM-ICE.

В качестве недорогой альтернативы можно рассмотреть Wiggler-совместимый гальванически развязанный эмулятор AS-JTAG производства АРГУССОФТ. Этот эмулятор подключается к компьютеру по LPT-порту и через RDI-сервер H-JTAG подключается к IAR в качестве RDI-эмулятора. При настройке необходимо указать библиотеку `hjtga.dll`. Скорость работы незначительно снижается по сравнению с AT91SAM-ICE, но дискомфорта не вызывает.

### Вносим изменения «Hello, World!»

Через JTAG-эмулятор можно загрузить программу в микроконтроллер и выполнить ее в режиме отладки. Для этого надо подключить эмулятор к плате, выставить RAM\_Debug в поле Workspace, скомпилировать, начать отладку (Project – Debug), запустить программу (Debug – Go).

Программа принимает данные с порта, возвращает их обратно, проверяет статус ошибки, возвращает его в терминал. Чтобы при вводе буквы 'z' в терминале появилось сообщение «Hello, world» в файле interrupt\_Usart.c функция Uart\_c\_irq\_handler приводится к виду:

```

... ранее без изменений
/* get Usart status register
status = USART_pt->US_CSR;
if ( status & AT91C_US_RXRDY) {
/* Get byte and send
char tmp =
AT91F_US_GetChar(USART_pt);
const char msg[] =
«Hello, world!»;
if (tmp == 'z')
{
AT91F_US_SendFrame(USART_pt
(char *)msg, sizeof(msg), 0, 0);
}
else
AT91F_US_PutChar (USART_pt, tmp);
}
if (status & AT91C_US_OVRE)
{
... далее без изменений.

```

Функция AT91F\_US\_SendFrame передает через PDC данные из буфера msg в количестве, равном длине буфера. Ядро процессора при этом остается свободным. Аналогично данные можно принимать. Два нуля обозначают отсутствие второго буфера. Эта функция уже есть в библиотеке контроллера. Если щелкнуть правой кнопкой мыши на названии функции, а затем выбрать пункт Go to definition of... (перейти к определению функции), то откроется описание функции. Для AT91F\_US\_SendFrame откроется файл lib\_AT91SAM7S256.h, в котором находятся функции для работы с контроллером. Перед тем как писать свою функцию для работы с модулем, стоит посмотреть, нет

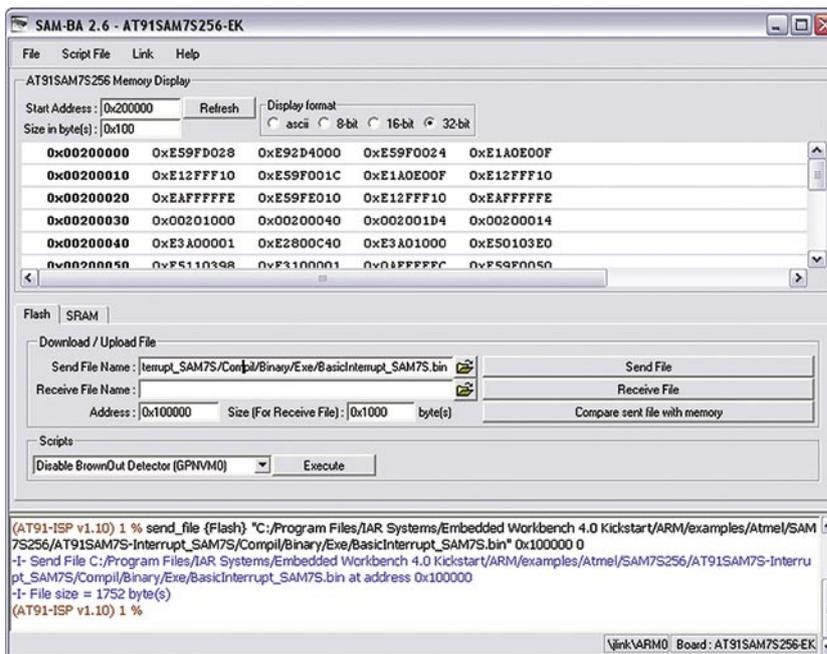


Рис. 5. Работа с SAM-BA

ли аналогичной готовой функции в библиотеке.

### Совместная работа двух периферийных модулей

Чтобы показать, как можно использовать контроллеры SSC и PWM с использованием режима прямого доступа к памяти, рассмотрим систему, которая в строго определенных моменты передает данные по последовательному интерфейсу. Как известно, на вход в прерывание требуется несколько тактов, при этом не может быть прервано выполнение текущей инструкции. В наборе команд ARM не все команды выполняются за один такт, поэтому прерывание «в чистом виде» использовать не получается – момент начала выдачи очередного пакета будет «плавать» на несколько тактов процессора. Естественное решение - использование PWM для формирования временных импульсов. Для передачи используется контроллер SSC, который в одном из режимов начинает передачу по внешнему фронту. Внешним фронтом будет служить сигнал с PWM. Если PDC запрограммирован до прихода фронта, то передача каждо-

го пакета данных будет начинаться в строго определенное импульсом PWM время.

### ВМЕСТО ЗАКЛЮЧЕНИЯ

При программировании микроконтроллеров постоянно приходится работать с периферийными модулями. Не всегда понятное описание управляющих регистров может стать гораздо яснее, если параллельно рассматривать практику работы с ними на основе библиотеки, а непонятные названия становятся осмысленными после внимательного прочтения документации на кристалл. Также полезно изучить примеры исходных кодов, которые прилагаются к дистрибутивам компиляторов.

Во второй части статьи будет рассмотрена работа с микроконтроллерами на ядре ARM9.

### ЛИТЕРАТУРА

1. Королев Н. 32-разрядные микроконтроллеры ATMEL на базе ядра ARM7 // Компоненты и технологии, 2006, №6.
2. Королев Н. Современные микроконтроллеры ATMEL: акцент на 32-разряда // Электронные компоненты, 2006, №12.