

# Электроника

Юрий Ревич



**Практическое  
программирование  
микроконтроллеров**

**Atmel AVR**  
**на языке ассемблера**

**3-е издание**

УДК 004.431.4  
ББК 32.973.26-018.1  
P32

**Ревич Ю. В.**

P32 Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. — 3-е изд., испр. — СПб.: БХВ-Петербург, 2014. — 368 с.: ил. — (Электроника)

ISBN 978-5-9775-3311-9

Изложены принципы функционирования, особенности архитектуры и приемы программирования микроконтроллеров Atmel AVR. Приведены готовые рецепты для программирования основных функций современной микроэлектронной аппаратуры: от реакции на нажатие кнопки или построения динамической индикации до сложных протоколов записи данных во внешнюю память или особенностей подключения часов реального времени. Особое внимание уделяется обмену данными микроэлектронных устройств с персональным компьютером, приводятся примеры программ. В книге учтены особенности современных моделей AVR и сопутствующих микросхем последних лет выпуска. Приложение содержит основные параметры микроконтроллеров AVR, перечень команд и тексты программ для них, а также список используемых терминов и аббревиатур.

*Для учащихся, инженерно-технических работников и радиолюбителей*

УДК 004.431.4  
ББК 32.973.26-018.1

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Редактор	<i>Леонид Кочин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 31.03.14.  
Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 29,67.  
Тираж 1500 экз. Заказ №  
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.  
Первая Академическая типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3311-9

© Ревич Ю. В., 2014  
© Оформление, издательство "БХВ-Петербург", 2014

# Оглавление

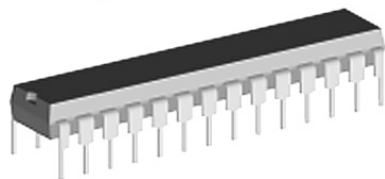
<b>ВВЕДЕНИЕ. Микроконтроллеры, их возникновение и применение.....</b>	<b>7</b>
Предыстория микроконтроллеров .....	8
Электроника в греческом стиле.....	10
Почему AVR?.....	12
Что дальше? .....	15
<b>ЧАСТЬ I. ОБЩИЕ ПРИНЦИПЫ УСТРОЙСТВА И ФУНКЦИОНИРОВАНИЯ ATMEL AVR .....</b>	<b>17</b>
<b>Глава 1. Обзор микроконтроллеров Atmel AVR.....</b>	<b>19</b>
Семейства AVR.....	21
Особенности практического использования МК AVR.....	23
<b>Глава 2. Общее устройство, организация памяти, тактирование, сброс .....</b>	<b>27</b>
Память программ.....	27
Память данных (ОЗУ, SRAM) .....	29
Энергонезависимая память данных (EEPROM).....	31
Способы тактирования .....	32
Сброс.....	34
Особенности подключения дополнительной внешней памяти данных .....	36
<b>Глава 3. Знакомство с периферийными устройствами .....</b>	<b>39</b>
Порты ввода-вывода.....	40
Таймеры-счетчики.....	42
Аналого-цифровой преобразователь.....	43
Последовательные порты.....	45
UART .....	45
Интерфейс SPI.....	49
Интерфейс TWI (I <sup>2</sup> C).....	52
Универсальный последовательный интерфейс USI.....	53
<b>Глава 4. Прерывания и режимы энергосбережения .....</b>	<b>55</b>
Прерывания.....	55
Разновидности прерываний .....	59
Режимы энергосбережения.....	61
Потребление МК AVR.....	62
Потребление МК AVR и режимы энергосбережения.....	65

<b>ЧАСТЬ II. ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ ATMEL AVR.....</b>	<b>67</b>
<b>Глава 5. Общие принципы программирования МК семейства AVR.....</b>	<b>69</b>
Ассемблер или С? .....	69
Способы и средства программирования AVR.....	73
Редактор кода .....	73
Об AVR Studio.....	74
Обустройство ассемблера .....	76
Программаторы.....	77
О hex-файлах .....	82
Команды, инструкции и нотация AVR-ассемблера .....	84
Числа и выражения .....	86
Директивы и функции .....	87
Общая структура AVR-программы .....	91
Обработка прерываний.....	93
<i>RESET</i> .....	96
Простейшая программа.....	97
Задержка .....	99
Программа счетчика .....	101
Использование прерываний.....	103
Задержка по таймеру .....	104
Программа счетчика с использованием прерываний .....	105
О конфигурационных битах.....	108
<b>Глава 6. Система команд AVR .....</b>	<b>111</b>
Команды передачи управления и регистр <i>SREG</i> .....	111
Команды проверки-пропуска.....	117
Команды логических операций .....	119
Команды сдвига и операции с битами .....	120
Команды арифметических операций .....	122
Команды пересылки данных.....	124
Команды управления системой .....	128
Выполнение типовых процедур на ассемблере.....	129
О стеке, локальных и глобальных переменных .....	131
<b>Глава 7. Арифметические операции.....</b>	<b>133</b>
Стандартные арифметические операции .....	134
Умножение многоразрядных чисел.....	135
Деление многоразрядных чисел .....	137
Операции с дробными числами.....	140
Генератор случайных чисел.....	142
Операции с числами в формате BCD.....	144
Отрицательные числа в МК.....	149
<b>Глава 8. Программирование таймеров.....</b>	<b>153</b>
8- и 16-разрядные таймеры .....	153
Формирование заданного значения частоты.....	156
Отсчет времени.....	159
Точная коррекция времени .....	164

Частотомер и периодомер.....	167
Частотомер .....	167
Периодомер .....	171
Управление динамической индикацией.....	174
LED-индикаторы и их подключение .....	175
Программирование динамической индикации.....	178
Таймеры в режиме PWM .....	181
<b>Глава 9. Использование EEPROM.....</b>	<b>187</b>
Еще раз о сохранности данных в EEPROM.....	187
Запись и чтение EEPROM.....	189
Хранение констант в EEPROM .....	191
<b>Глава 10. Аналоговый компаратор и АЦП.....</b>	<b>195</b>
Аналого-цифровые операции и их погрешности .....	195
Работа с аналоговым компаратором .....	198
Интегрирующий АЦП на компараторе.....	201
Принцип работы и расчетные формулы .....	202
Программа интегрирующего АЦП.....	206
Встроенный АЦП.....	209
Пример использования АЦП .....	213
Программа .....	215
<b>Глава 11. Программирование SPI.....</b>	<b>223</b>
Основные операции через SPI .....	223
Аппаратный вариант.....	224
Программный вариант.....	226
О разновидностях энергонезависимой памяти.....	227
Запись и чтение flash-памяти через SPI .....	229
Программа обмена с памятью 45DB011B по SPI .....	232
Запись и чтение flash-карт.....	232
Подключение карт MMC.....	234
Подача команд и инициализация MMC .....	235
Запись и чтение MMC .....	240
<b>Глава 12. Интерфейс TWI (I<sup>2</sup>C) и его практическое использование.....</b>	<b>245</b>
Базовый протокол I <sup>2</sup> C.....	245
Программная эмуляция протокола I <sup>2</sup> C .....	248
Запись данных во внешнюю энергонезависимую память.....	249
Режимы обмена с памятью AT24 .....	250
Программа .....	252
Часы с интерфейсом I <sup>2</sup> C.....	256
Запись данных .....	263
Чтение данных .....	267
<b>Глава 13. Программирование UART/USART .....</b>	<b>269</b>
Инициализация UART.....	270
Передача и прием данных.....	272
Пример установки часов DS1307 с помощью UART .....	275

Приемы защиты от сбоев при коммуникации.....	279
Проверка на четность .....	280
Как организовать корректный обмен.....	283
Дополнительные возможности USART .....	285
Реализация интерфейсов RS-232 и RS-485.....	286
Преобразователи уровня для RS-232 .....	290
RS-485 .....	294
<b>Глава 14. Режимы энергосбережения и сторожевой таймер .....</b>	<b>297</b>
Программирование режима энергосбережения .....	299
Пример прибора с батарейным питанием .....	300
Доработка программы .....	302
Использование сторожевого таймера .....	306
<b>ПРИЛОЖЕНИЯ .....</b>	<b>311</b>
<b>Приложение 1. Основные параметры микроконтроллеров Atmel AVR.....</b>	<b>313</b>
<b>Приложение 2. Команды Atmel AVR.....</b>	<b>321</b>
Арифметические и логические команды .....	322
Команды операций с битами .....	323
Команды сравнения .....	324
Команды передачи управления .....	325
Команды безусловного перехода и вызова подпрограмм .....	325
Команды проверки-пропуска и команды условного перехода .....	326
Команды переноса данных .....	327
Команды управления системой .....	328
<b>Приложение 3. Тексты программ .....</b>	<b>329</b>
Демонстрационная программа обмена данными с flash-памятью 45DB011B по интерфейсу SPI .....	329
Процедуры обмена по интерфейсу I <sup>2</sup> C .....	333
<b>Приложение 4. Обмен данными с персональным компьютером и отладка программ через UART .....</b>	<b>341</b>
Работа с COM-портом в Delphi .....	342
COM-порт и Windows API .....	342
Работа с COM через готовые компоненты .....	348
Установка линии RTS в DOS и Windows.....	349
Программа COM2000.....	351
Отладка программ с помощью терминальной программы .....	353
<b>Приложение 5. Словарь часто встречающихся аббревиатур и терминов .....</b>	<b>355</b>
Соответствие терминов на русском их переводу на английский.....	357
Соответствие терминов на английском их переводу на русский.....	358
<b>Литература .....</b>	<b>361</b>
<b>Предметный указатель .....</b>	<b>363</b>

## ГЛАВА 1



# Обзор микроконтроллеров Atmel AVR

Atmel AVR представляет собой семейство универсальных 8-разрядных микроконтроллеров на основе общего ядра с различными встроенными периферийными устройствами. Возможности МК AVR позволяют решить множество типовых задач, возникающих перед разработчиками радиоэлектронной аппаратуры.

Особенности микроконтроллеров Atmel AVR.

- **Производительность порядка 1 MIPS/МГц.** MIPS (Millions of Instructions Per Second, миллион команд в секунду) — старая и во многом формальная характеристика производительности процессоров, т. к. наборы команд для различных процессоров различаются, и, соответственно, одно и то же число инструкций на различных системах даст разную полезную работу. Тем не менее для простых 8-разрядных вычислительных систем, не содержащих команд, оперирующих с большими числами, числами с плавающей точкой и массивами данных, это неплохой показатель для сравнения их производительности. Вычислительное ядро AVR на ряде задач по производительности превосходит 16-разрядный процессор 80286.
- **Усовершенствованная RISC-архитектура.** Концепция RISC (Reduced Instruction Set Computing, вычисления с сокращенным набором команд) предполагает наличие набора команд, состоящего из минимума компактных и быстро выполняющихся инструкций; при этом такие более громоздкие операции, как вычисления с плавающей точкой или арифметические действия с многоразрядными числами, предполагается реализовать на уровне подпрограмм. Концепция RISC упрощает устройство ядра (в типовом ядре AVR содержится лишь 32 тыс. транзисторов, в отличие от десятков миллионов в процессорах для ПК) и ускоряет его работу: типовая инструкция выполняется за один такт (кроме команд ветвления программы, обращения к памяти и некоторых других, оперирующих с данными большой длины). В AVR имеется простейший двухступенчатый конвейер, выполняющий команду в одном такте с выборкой следующей. В отличие от Intel-архитектур, в «классическом» AVR нет аппаратного умножения/деления, однако в подсемействе Mega присутствуют операции умножения.
- **Раздельные шины памяти команд и данных.** AVR (как и большинство других микроконтроллеров) имеет так называемую *гарвардскую архитектуру*, где об-

ласти памяти программ и данных разделены (в отличие от классической архитектуры фон Неймана в обычных компьютерах, где память общая). Раздельные шины для этих областей памяти значительно ускоряют выполнение программы: данные и команды могут выбираться одновременно.

- **32 регистра общего назначения (РОН).** Atmel была первой компанией, далеко отошедшей от классической модели вычислительного ядра, в которой выполнение команд предусматривает обмен данными между АЛУ и запоминающими ячейками в общей памяти. Введение РОН в таком количестве (напомним, что в архитектуре x86 всего четыре таких регистра, а в x51 понятие РОН, как таковое, отсутствует) в ряде случаев позволяет вообще отказаться от расположения глобальных и локальных переменных в ОЗУ и от использования стека, операции с которым усложняют и загромождают программу. В результате структура ассемблерной программы приближается к программам на языках высокого уровня. Правда, это привело к некоторому усложнению системы команд, номенклатура которых для AVR больше, чем в других RISC-семействах (хотя значительная часть инструкций — псевдонимы).
- **Flash-память программ** (10 000 циклов стирание/запись) с возможностью внутрисистемного перепрограммирования и загрузки через последовательный канал прямо в готовой схеме. О преимуществах такого подхода, ныне ставшего общепринятым, подробно рассказано во *введении*.
- **Отдельная область энергонезависимой памяти (EEPROM, 100 000 циклов стирание/запись)** для хранения данных, с возможностью записи программным путем, или внешней загрузки через SPI-интерфейс.
- **Встроенные устройства для обработки аналоговых сигналов:** аналоговый компаратор и многоканальный 10-разрядный АЦП.
- **Сторожевой таймер**, позволяющий автоматически перезагружать контроллер через определенные промежутки времени (например, для выхода из «спящего» режима).
- **Последовательные интерфейсы SPI, TWI (I<sup>2</sup>C) и UART (USART),** реализующие аппаратный обмен данными с большинством стандартных датчиков и других внешних устройств (в том числе таких, как персональные компьютеры).
- **Таймеры-счетчики** с предустановкой и возможностью выбора источника счетных импульсов: как правило, один-два 8-разрядных и как минимум один 16-разрядный, в том числе поддерживающие режим многоканальной 8-, 9-, 10-, 16-битовой широтно-импульсной модуляции (PWM).
- **Возможность работы при тактовой частоте** от 0 Гц до 16–20 МГц.
- **Диапазон напряжений питания от 2,7 до 5,5 В** (в некоторых случаях от 1,8 или до 6,0 В).
- **Многочисленные режимы энергосбережения,** отличающиеся числом узлов, остающихся подключенными. Выход из «спящих» режимов по сторожевому таймеру или по внешним прерываниям.



□ **Встроенный монитор питания** — детектор падения напряжения (Brown-out Detection).

Здесь перечислены далеко не все особенности, характерные для различных моделей AVR. С некоторыми другими мы познакомимся в дальнейшем, а также на практике рассмотрим перечисленные подробнее. Но сначала дадим общую характеристику различных семейств AVR с точки зрения их преимущественного назначения.

## Семейства AVR

В 2002 г. фирма Atmel начала выпуск новых подсемейств 8-разрядных МК на базе AVR-ядра. С тех пор все МК этого семейства делятся на три группы (подсемейства): Classic, Tiny и Mega. В 2008 г. к ним добавилось семейство XMeta. МК семейства Classic (AT90Sxxxx) уже не выпускаются; дольше всего в производстве «задержалась» очень удачная (простая, компактная и быстродействующая модель) AT90S2313, но и она была в 2005 г. заменена на ATtiny2313. Все «классические» AVR с первыми цифрами 2 и 8 в наименовании модели (означающими объем памяти программ в килобайтах) имеют аналоги в семействах Tiny и Mega. Для Mega при программировании возможна установка специального бита совместимости, который позволяет без каких-либо изменений использовать программы, созданные для семейства Classic. Примеры в этой книге мы по возможности постарались изложить в версиях для новых модификаций МК, но в Интернете «гуляет» еще много примеров программ для семейства Classic, потому основные отличия старых и новых версий приведены в соответствующих разделах.

Примеры различных типов корпусов, в которых выпускаются микросхемы AVR, приведены на рис. 1.1. Более подробную информацию на эту тему можно найти в *приложении 1* (табл. П1.2), а также в технической документации на устройства. Отметим, что для радиолюбительских нужд и макетирования удобнее всего микросхемы в PDIP-корпусах, но в них производятся не все модели МК.

Все семейства могут иметь две модификации: буква «L» в обозначении говорит о расширенном диапазоне питания 2,7–5,5 В, отсутствие такой буквы означает диапазон питания 4,5–5,5 В. При выборе конкретного типа микросхемы нужно быть внимательным, т. к. L-версии одновременно также и менее быстродействующие, у большинства из них максимальная тактовая частота ограничена значением 8 МГц. Для «обычных» версий максимальная частота составляет 16 или 20 МГц. Хотя, как правило, при запуске L-микросхем с напряжением питания 5 В на частотах до 10–12 МГц неприятностей ожидать не следует (аналогично версии без буквы L вполне могут работать при напряжении питания около 3 В, разумеется, не на экстремальных значениях частот), тем не менее при проектировании высоконадежных устройств нужно учитывать это требование.

Микросхемы Tiny имеют Flash-ПЗУ программ объемом 1–8 кбайт и размещаются в основном в корпусах с 8–20 выводами (кроме ATtiny28), т. е. они в целом предназначены для более простых и дешевых устройств. Это не значит, что их возможности во всех случаях более ограничены, чем у семейства Mega. Так, например, ATtiny26 при цене менее 2 долларов содержит таймер с высокоскоростным ШИМ-режимом (в других моделях такого нет), а также 11-канальный АЦП с возмож-

ностью работы в дифференциальном режиме, с регулируемым входным усилителем и встроенным источником опорного напряжения, что характерно для старших моделей. Микросхема ATtiny2313, как уже говорилось, представляет собой улучшенную версию одного из наиболее универсальных и удобных «классических» AVR AT90S2313. Старший в семействе Tiny в настоящий момент МК ATtiny828 с 8 килобайтами памяти программ и корпусом с 32 выводами по некоторым параметрам даже превышает младшие Mega: например, у него уникальное число каналов АЦП — 28.

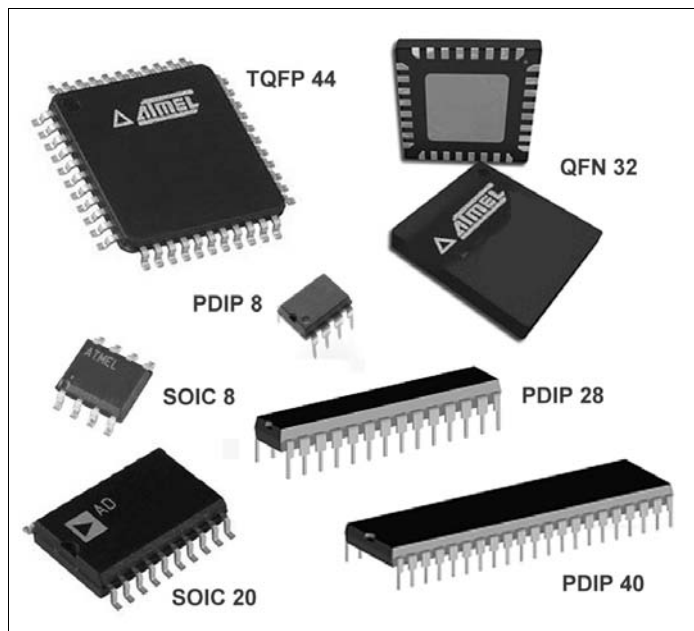


Рис. 1.1. Примеры различных типов корпусов для МК AVR

Подсемейство Mega оснащено Flash-ПЗУ программ объемом 8–256 кбайт и корпусами с 28–100 выводами. В целом МК этой группы более «навороченные», чем Tiny, имеют весьма разветвленную систему встроенных устройств с развитой функциональностью.

Таблицы с основными характеристиками некоторых моделей Tiny и Mega из числа самых ходовых приведены в *приложении 1*. Там же даны некоторые общие технические характеристики семейства AVR. Более подробные сведения можно почерпнуть из [1, 2] и фирменной технической документации, которая доступна на сайте Atmel для каждой модели.

Кроме этих трех семейств, на базе AVR-ядра выпускаются специализированные микросхемы для работы с USB-интерфейсом (AT90USBxxx), промышленным интерфейсом CAN (AT90CANxxx), для управления ЖК-дисплеями (ATmega329 и др.), с беспроводным интерфейсом IEEE 802.15.4 (ZigBee) для предприятий торговли и некоторые другие. В последнее время некоторые микроконтроллеры серий Tiny и Mega стали выпускаться в версиях со сверхмалым потреблением (технология

riCoPower с напряжением питания от 1,8 В, в конце наименования МК этой серии добавлена буква «Р») и высокотемпературных для использования в автомобильной промышленности (версии Automotive).

Из новых моделей стоит обратить внимание на ATmega32HVE2 и ATmega64HVE2. Они изначально предназначены для мониторинга состояния свинцовых и литий-ионных батарей, однако их с успехом можно применять и в других приложениях. Их особенностями являются, во-первых, регулятор напряжения питания кристалла, позволяющий подключать микросхему напрямую к источнику +12 В, во-вторых, АЦП повышенной точности — 17–18 двоичных разрядов («дельта-сигма» типа, в отличие от обычных, имеющих АЦП последовательного приближения). В дополнение к этому на кристалле размещен 32/64-разрядный математический сопроцессор, выполняющий арифметические и логические операции с 32-разрядными операндами за 1–2 такта, а с 64-разрядными операндами — за 2–6 тактов. Такое построение МК типа HVE2 позволяет в значительной степени упростить выполнение арифметических операций с многоразрядными (в том числе действительными) числами (см. главу 7).

Семейство XMeга предназначено для относительно «продвинутых» устройств: контроллеры из этой серии работают при напряжении питания 1,8–3,6 В, обладают повышенным быстродействием (тактовая частота до 32 МГц), 12-разрядным 16-канальным АЦП и 2–4 каналами ЦАП (в других семействах они отсутствуют), несколькими каналами UART и/или USB и других последовательных портов (причем с возможностью работы в автономном режиме, при остановленном ядре), встроенной поддержкой криптографии, усовершенствованным режимом riCoPower и другими «наворотами» [23]. Существует также отдельное семейство 32-разрядных МК AVR32, предназначенное для высокоскоростных приложений, таких как обработка видеопотока или распознавание образов в реальном времени.

## Особенности практического использования МК AVR

При использовании AVR возникает ряд вопросов практического характера, игнорирование которых может иногда привести к неработоспособности или сбоям устройства (а в некоторых случаях — даже к невозможности его запрограммировать). Например, одна из таких проблем — возможность потери содержимого EEPROM при выключении питания. Эту и подобные проблемы мы подробно рассмотрим в соответствующих главах. О вопросах, связанных с энергопотреблением МК, см. главу 4, раздел «Режимы энергосбережения и потребление». Здесь же остановимся на некоторых общих вопросах включения МК AVR.

У большинства выводов МК имеется встроенный подключаемый «подтягивающий» (т. е. подсоединенный к шине питания) резистор, что, казалось бы, решает одну из обычных схемотехнических проблем, когда наличие такого резистора требуется для подключения двухвыводных кнопок или выходов с «открытым коллектором». Однако в критичных случаях необходим внешний резистор сопротивлением 2–5 кОм (в случае схем, критичных для потребления — 10–30 кОм).

«Подтягивающий» резистор следует устанавливать не только на выводе /RESET (о чем пойдет речь в *главе 2*), но и в том случае, когда выводы SCK, MOSI и MISO соответствующих портов используются для программирования и подключены к программирующему разъему ISP (см. *главу 5*), а также по выводам внешних прерываний, если они задействованы. Если эти выводы не «подтягивать» к напряжению питания дополнительными резисторами (хотя это и не оговорено в технической документации), то не исключены ложные срабатывания внешних прерываний, перезапуск системы, а при очень мощных помехах — даже порча данных в памяти программ. С другой стороны, когда выводы программирования выполняют функцию обычных портов, сконфигурированных на выход, а в устройстве применяются режимы энергосбережения, наличие «подтягивающих» резисторов может привести к лишнему потреблению тока (при установке вывода в логический ноль через резистор потечет ток от источника питания на вход МК). Если реализован один из режимов энергосбережения, то нужно тщательно проанализировать схему, чтобы исключить ситуации, при которых через эти резисторы протекает ток.

Также всегда следует устанавливать внешние резисторы при работе выводов МК на общую шину, как в интерфейсе I<sup>2</sup>C (или просто при подсоединении входа МК к выходу другого устройства с открытым коллектором, например, мониторов питания, описанных в *главе 3*), при подключении к двухвыводным кнопкам (особенно при наличии внешнего прерывания, см. *главы 4 и 5*). Сопротивление встроенного резистора (на самом деле представляющего собой, разумеется, полевой транзистор) в таких случаях слишком велико для того, чтобы электромагнитные помехи («наводки») на нем эффективно «садились».

Все без исключения МК AVR работают при напряжении питания в диапазоне 4,5–5,5 В, обеспечивая при этом заданное быстродействие. Максимально допустимое напряжение составляет 6 В и превышать его не рекомендуется. Для подавляющего большинства моделей нижний порог питания составляет 2,7 В, для многих он еще ниже — 1,8 или даже 1,7 В. Однако при выборе напряжения источника нужно быть внимательным: как правило, при пониженном питании максимальная тактовая частота тоже оказывается сниженной, потому следует ориентироваться не только на цифры в таблице характеристик на сайте Atmel, но и заглянуть в документацию. Например, для ATtiny2313, который мы будем часто упоминать далее, максимальная тактовая частота равна 20 МГц, однако это действительно только в диапазоне напряжения 4,5–5,5 В. При минимально допустимом напряжении 1,8 В возможная частота составляет всего 4 МГц, а при 2,7 В — 10 МГц. Кроме того, следует обращать внимание на букву в конце наименования модели: так, ATmega8535 работает лишь в диапазоне питающего напряжения 4,5–5,5 В (частота до 16 МГц), а для диапазона 2,7–5,5 В (частота до 8 МГц) нужна модель ATmega8535L.

Микросхемы AVR, как и всякая КМОП-логика, благодаря высокому порогу срабатывания эффективно защищены от помех по шине «земли»<sup>1</sup>. Однако они сильно

---

<sup>1</sup> Как и в других книгах автора (например, [8]), здесь термин «земля» в кавычках означает то же самое, что «общий провод» (обозначается в нашей литературе значком  $\perp$ , а в западной треугольником  $\blacktriangledown$ ). Его следует отличать от заземления («настоящей земли» в электротехническом смысле), обозначаемся значком  $\perp$ .

подвержены помехам по шине питания. Поэтому не забывайте о развязывающих конденсаторах, которые нужно устанавливать непосредственно у выводов питания (керамические 0,1–0,5 мкФ), а также про качество сетевых выпрямителей и стабилизаторов.

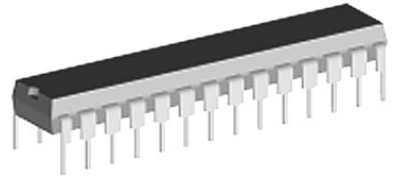
Принадлежность к семейству КМОП сказывается и еще в одном отношении: МК AVR плохо переносят изменение полярности питания. По входу КМОП всегда (за исключением некоторых специальных кристаллов) имеются защитные диоды, через которые при неправильном подключении питания начинает течь ничем не ограниченный прямой ток. Опыт показывает, что, в зависимости от мощности источника, МК могут выдержать такой режим на протяжении нескольких секунд, но потом бесповоротно «сгорают». Зато выходы AVR легко выдерживают перегрузки и режим КЗ в течение длительного времени — как и любая КМОП-логика, они переходят в режим ограничения тока.

### **ЗАМЕТКИ НА ПОЛЯХ**

Из-за этого свойства к выходам AVR иногда напрямую — без токоограничивающего резистора — подключают светодиоды, однако я бы такое рекомендовать не стал. Стандартная КМОП-логика (серии 4000В) такое «издевательство» еще выдерживает, но быстродействующая (серий АС и НС, а также МК AVR) работает гораздо хуже. При напряжении 2,7–3 В ток короткого замыкания выхода AVR (порядка 30 мА) будет близок к предельно допустимому для светодиодов, а при питании 5 В возрастет до 70–80 мА и значительно превысит любые нормы для обычных светодиодов в корпусах 3–5 мм диаметром. Кроме того, большинство светодиодов выдают нормальную яркость уже при токах около 3–5 мА, а в режиме прямого подключения, даже если ток и не превышает максимально допустимого, будут сверкать, как боевой лазер. Потому такой режим не только опасен для сохранности светодиодов, но и просто нецелесообразен.

При проектировании устройств на МК AVR необходимо помнить про общее ограничение тока через выводы питания, которое для абсолютного большинства моделей составляет 200 мА. Для многоконтактных корпусов это может вырасти в серьезную проблему: например, в модели ATmega640/1280/2560 среди 100 контактов 86 выводов портов общего назначения, каждый из которых можно нагрузить отдельно от других. Зная хорошие нагрузочные свойства этих выводов, возникает соблазн применить их непосредственно для управления какими-нибудь матричными светодиодными табло или шаговыми двигателями. Но при этом нужно всегда тщательно подсчитать суммарный ток через вывод питания, чтобы не превысить допустимое значение. Расчет показывает, что всего двух-трех таких выводов, оказавшихся в режиме КЗ, хватит для превышения максимального значения — наверное, с первого раза МК не сгорит, но при систематическом превышении общего тока едва ли проработает долго.

## ГЛАВА 2



# Общее устройство, организация памяти, тактирование, сброс

Все основные компоненты МК AVR (за исключением модуля JTAG) показаны на общей структурной схеме (рис. 2.1). В отдельных моделях некоторые составляющие могут отсутствовать или различаться по характеристикам, неизменным остается только общее 8-разрядное процессорное ядро (GPU, General Processing Unit). Кратко опишем наиболее важные компоненты, большинство из которых мы подробно будем рассматривать в дальнейшем.

Начнем с памяти. В структуре AVR имеются три разновидности памяти: flash-память программ, ОЗУ (SRAM) для временных данных и энергонезависимая память (EEPROM) для долговременного хранения констант и данных. Рассмотрим их по отдельности.

## Память программ

Объем встроенной flash-памяти программ в AVR-контроллерах составляет от 1 кбайта у ATtiny11 до 256 кбайт у ATmega2560. Первое число в наименовании модели соответствует величине этой памяти из ряда: 1, 2, 4, 8, 16, 32, 64, 128 и 256 кбайт. Память программ, как и любая другая flash-память, имеет страничную организацию (размер страницы, в зависимости от модели, составляет от 64 до 256 байт). Страница может программироваться только целиком. Число циклов перепрограммирования достигает 10 тыс.

С точки зрения программиста память программ можно считать построенной из отдельных ячеек — слов по два байта каждое. Устройство памяти программ (и только этой памяти) по двухбайтовым словам — очень важный момент, который нужно твердо усвоить. Такая организация обусловлена тем, что любая команда в AVR имеет длину ровно два байта. Исключения составляют команды `JMP`, `CALL` и некоторые другие (например, `LDS`), которые оперируют с 16-разрядными и более длинными адресами, длина этих команд равна четырем байтам и они применяются лишь в моделях с памятью программ объемом свыше 8 кбайт (подробнее см. главу 5). Во всех остальных случаях счетчик команд сдвигается при выполнении очередной команды на два байта (одно слово), поэтому необходимую емкость памяти легко подсчитать, зная число используемых команд. Абсолютные адреса в памяти про-

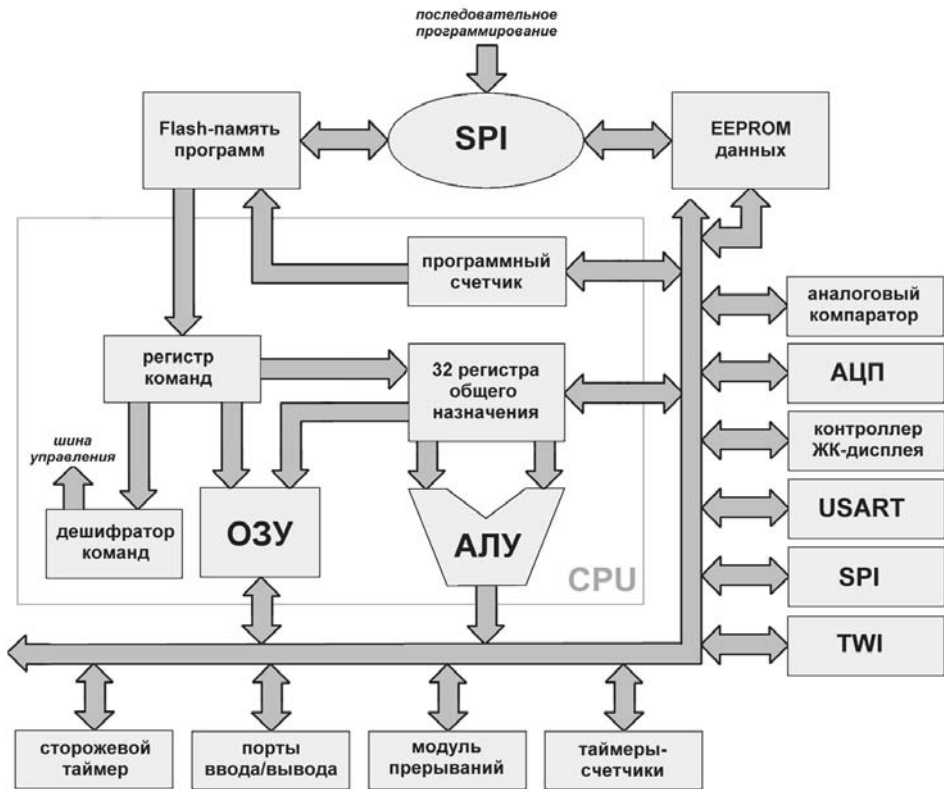


Рис. 2.1. Общая структурная схема микроконтроллеров AVR

грамм (указываемые, например, в таблицах векторов прерываний в техническом описании МК) также отсчитываются в словах.

### ЗАМЕТКИ НА ПОЛЯХ

Приведем пример интересного случая адресации, который представляет команда для чтения констант из памяти LPM (а также ELPM в МК с памятью программ 128 кбайт и более). Эта команда подразумевает чтение по *байтовому* адресу, указанному в двух старших РОН (образующих так называемый регистр Z, см. *далее*). Однако чтобы не нарушать «чистоту» концепции организации памяти программ по словам, разработчики запутали этот простой вопрос, указав в описании, что при вызове команды LPM старшие 15 разрядов регистра Z адресуют *слово* в памяти, а младший разряд выбирает младший или старший байт (при равенстве разряда 0 или 1 соответственно) этого слова. Легко, однако, заметить, что байтовая и пословная организации памяти при таком подходе эквивалентны.

Последний адрес существующего объема памяти программ для конкретной модели обозначается константой FLASHEND. По умолчанию все контроллеры AVR всегда начинают выполнение программы с адреса \$0000. Если в программе нет прерываний, то с этого адреса может начинаться прикладная программа. В противном случае по данному адресу располагается так называемая *таблица векторов прерываний*, подробнее о которой мы будем говорить в *главах 4 и 5*. Здесь укажем лишь,

что первым в этой таблице (по тому же адресу \$0000) всегда размещается вектор сброса RESET, который указывает на процедуру, выполняющуюся при сбросе МК (в том числе и при включении питания).

### **ПРИМЕЧАНИЕ**

В ассемблере AVR можно обозначать шестнадцатеричные числа в «паскалевском» стиле, предваряя их знаком \$, при этом стиль языка C (0x00) тоже действителен, а вот «интеловский» способ (00h) не работает. Подробнее об обозначениях чисел различных систем счисления в AVR-ассемблере см. главу 5.

В последних адресах памяти программ контроллеров семейства Mega может располагаться так называемый *загрузчик* — специальная программа, которая управляет загрузкой и выгрузкой прикладных программ из основного объема памяти. В этом случае положение вектора сброса и всей таблицы векторов прерываний (т. е. фактически начального адреса, с которого начинается выполнение программы) может быть изменено установкой специальных конфигурационных ячеек (см. главу 5).

## **Память данных (ОЗУ, SRAM)**

В отличие от памяти программ, адресное пространство памяти данных адресуется побайтно (а не пословно). Адресация полностью линейная, без какого-то деления на страницы, сегменты или банки, как это принято в некоторых других системах. Младшие МК семейства Tiny (включая Tiny1x и Tiny28) памяти данных, как таковой, не имеют, ограничиваясь лишь регистровым файлом (РОН) и регистрами ввода-вывода (РВВ). В других моделях объем встроенной SRAM колеблется от 128 байт в представителях семейства Tiny (например, у ATtiny2313) до 4–8 кбайт у старших моделей Mega.

Адресное пространство статической памяти данных (SRAM) условно делится на несколько областей, показанных на рис. 2.2. Темной заливкой выделена часть, относящаяся к собственно встроенной SRAM, до нее по порядку адресов расположено адресное пространство регистров (первые 32 байта занимает РОН, еще 64 — РВВ). Для старших моделей Mega со сложной структурой (например, ATmega128) 64-х регистров ввода-вывода может оказаться недостаточно, поэтому в них для дополнительных РВВ выделяется отдельное адресное пространство (от \$60 до максимально возможного в байтовой адресации значения \$FF, итого таких регистров может быть всего 160).

### **ЗАМЕТКИ НА ПОЛЯХ**

В архитектуре МК AVR понятие «ввода-вывода» употребляется в двух смыслах: во-первых, имеются «порты ввода-вывода» (I/O ports), которые мы рассмотрим в главе 3. Во-вторых, «регистры ввода-вывода» (РВВ) в структуре AVR называются регистры, которые обеспечивают доступ к дополнительным компонентам, внешним по отношению к GPU, за исключением ОЗУ (в том числе и к портам ввода-вывода). Такое подразделение приближает структуру МК AVR к привычной конфигурации персонального компьютера, где доступ к любым внешним по отношению к центральному процессору компонентам, кроме памяти, осуществляется через порты ввода-вывода.



Для некоторых моделей Mega (ATmega8515, ATmega162, ATmega128, ATmega2560 и др.) предусмотрена возможность подключения внешней памяти объемом до 64 кбайт, которая может быть любой статической разновидностью (SRAM, Flash или EEPROM) с параллельным интерфейсом. В настоящее время гораздо удобнее применять для хранения данных внешнюю память с последовательным интерфейсом (см. главы 11 и 12), потому что вопрос с дополнительной памятью мы кратко рассмотрим в конце главы, чтобы не отвлекаться от основного изложения.

Отметим, что адреса POH и PBV не отнимают пространство у ОЗУ данных (за исключением подключаемой внешней памяти у старших моделей Mega, максимальный адрес которой ограничен значением \$FFFF): так, если в конкретной модели МК имеется 512 байт SRAM, а пространство регистров занимает первые 96 байт (до адреса \$60), то адреса SRAM займут адресное пространство от \$0060 до \$025F (т. е. от 96-й до 607-й ячейки включительно). Конец встроенной памяти данных обозначается константой RAMEND.



Рис. 2.2. Адресное пространство статической памяти данных (SRAM) микроконтроллеров AVR

Операции чтения/записи в память одинаково работают с любыми адресами из доступного пространства, и при работе с SRAM нужно быть внимательным: вместо записи в память вы легко можете «попасть» в какой-нибудь регистр. Например, команда загрузки значения регистра r16 в регистр r0 (`mov r0, r16`) равносильна записи в SRAM по нулевому адресу (`sts $0000, r16`). Адрес в памяти для POH совпадает с его номером. В то же время для непосредственной записи в PBV по его адресу в памяти к номеру регистра следует прибавить \$20: так регистр флагов SREG, который для большинства моделей располагается в конце таблицы PBV по адресу \$3F, в памяти имеет адрес \$5F. Устанавливать POH и PBV прямой адресацией памяти неудобно: такая запись всегда отнимает два такта вместо одного, характерного для большинства других команд, хотя иногда это позволяет обойти ограничения на манипуляции с некоторыми PBV. Но если имеется готовая программа, работающая с SRAM, то при замене моделей процессоров на более старшие нужно быть внимательным, поскольку в них младшие адреса SRAM могут перекрываться дополнительными PBV.

## Энергонезависимая память данных (EEPROM)

Все модели МК AVR (кроме снятого с производства ATtiny11) имеют встроенную EEPROM для хранения констант и данных при отключении питания. В разных моделях объем ее варьируется от 64 байт (ATtiny1x) до 4 кбайт (старшие модели Mega). Конец EEPROM обозначается константой `EEPROMEND` (это обозначение введено только для более поздних моделей AVR, потому при использовании этой константы иногда ее придется определять самому). Число циклов перепрограммирования EEPROM может достигать 100 тыс.

Напомним, что EEPROM отличается от Flash возможностью выборочного программирования побайтно (в принципе даже побитно, но этот способ недоступен пользователю). Однако в старших моделях семейства EEPROM, как и flash-память программ, имеет страничную организацию, правда, страницы эти невелики — до 4 байт каждая. На практике, как при программировании EEPROM по последовательному каналу (т. е. через SPI-интерфейс программирования), так и при записи и чтении EEPROM из программы, эта особенность не имеет значения, и доступ осуществляется побайтно.

Чтение из EEPROM осуществляется в течение одного машинного цикла (правда, на практике оно растягивается на четыре цикла, но программисту следить за этим специально не требуется). А вот запись в EEPROM протекает значительно медленнее, и к тому же с точно не определенной скоростью: цикл записи одного байта может занимать от 2 до ~4 мс и более. Процесс записи регулируется встроенным RC-генератором, частота которого нестабильна (при более низком напряжении питания можно ожидать, что время записи будет больше). За такое время при обычных тактовых частотах МК успевает выполнить несколько тысяч команд, потому программирование процедуры записи требует аккуратности: например, нужно следить, чтобы в момент записи не «вклинилось» прерывание (подробнее об этом см. главы 4 и 9).

Главная же сложность при работе с EEPROM — возможность повреждения ее содержимого при недостаточно быстром снижении напряжения питания в момент выключения. Обусловлено это тем, что при уменьшении напряжения питания до некоторого порога (ниже порога стабильной работы, но недостаточного для полного выключения) из-за колебаний напряжения МК начинает выполнять произвольные команды, в том числе может осуществить процедуру записи в EEPROM. Если учесть, что типовая команда МК AVR выполняется за десятые доли микросекунды, то ясно, что никакой реальный источник питания не может обеспечить снижение напряжения до нуля за нужное время. По опыту автора при питании от обычного стабилизатора типа LM7805 с рекомендованными значениями емкости конденсаторов на входе и на выходе содержимое EEPROM будет неизбежно испорчено примерно в половине случаев.

Этой проблемы не должно существовать, если константы записывают в EEPROM при программировании МК, а процедура записи в программе отсутствует (о том, как сформировать файл с данными для EEPROM, см. раздел «Директивы и функции» главы 5). Большая сохранность данных в таких случаях подтверждается и эм-

пирическими наблюдениями, и тем, что разрешение записи в EEPROM — процедура двухступенчатая (см. главу 9). Во всех же остальных случаях (а их, очевидно, абсолютное большинство — в EEPROM чаще всего хранят пользовательские установки и текущую конфигурацию при выключении питания) приходится принимать специальные меры. Наиболее кардинальная и универсальная из них — установка внешнего монитора питания, удерживающего МК в состоянии сброса при уменьшении напряжения питания ниже пороговой величины. Той же цели служит встроенный детектор падения напряжения (Brown-out Detection, BOD), имеющийся практически во всех моделях Tiny и Mega, но техническая документация не исключает при этом для надежности дублирование его и внешним монитором питания. Подробнее о схеме BOD и режимах сброса МК см. далее в этой главе, а о программировании EEPROM и мерах предосторожности при ее использовании см. главу 9.

## Способы тактирования

Канонический способ тактирования МК — подключение кварцевого резонатора к соответствующим выводам (рис. 2.3, а). Емкость конденсаторов С1 и С2 в типовом случае должна составлять 15–22 пФ (может быть увеличена до 33–47 пФ с одновременным повышением потребления). В большинстве моделей Tiny и Mega имеется специальный конфигурационный бит СКРОТ, который позволяет регулировать потребление. При установке этого бита в 1 (незапрограммированное состояние) размах колебаний генератора уменьшается, однако при этом сужается возможный диапазон частот и общая помехоустойчивость, поэтому задействовать этот режим не рекомендуется. Может быть также выбран низкочастотный кварцевый резонатор (например, «часовой» 32 768 Гц), при этом конденсаторы С1 и С2 могут отсутствовать, т. к. при установке СКРОТ в значение 0 подключаются имеющиеся в составе МК внутренние конденсаторы емкостью 36 пФ.

Кварцевый резонатор можно заменить керамическим. Автору этих строк удавалось запускать МК на нестандартных частотах, подключая вместо кварца миниатюрную индуктивность (при ее значении 4,7 мкГн и емкостях конденсаторов 91 пФ частота получается около 10 МГц), что заодно позволяет немного уменьшить габариты схемы.

Естественно, тактировать МК можно и от внешнего генератора (рис. 2.3, б). Особенно это удобно, когда требуется либо синхронизировать МК с внешними компонентами, либо получить очень точную частоту тактирования, выбрав соответствующий генератор (например, серии SG-8002 фирмы Epson).

Наоборот, когда точность не требуется, можно подключить внешнюю RC-цепочку (рис. 2.3, в). В этой схеме емкость С1 должна быть не менее 22 пФ, а резистор R1 выбирается из диапазона 3,3–100 кОм. Частота при этом определяется по формуле  $F = 2/3 RC$ . С1 можно не устанавливать вообще, если записать лог. 0 в конфигурационную ячейку СКРОТ, подключив тем самым внутренний конденсатор 36 пФ.

Наконец, можно вообще отказаться от внешних компонентов и обойтись встроенным RC-генератором, который способен работать на четырех приблизительных

значениях частот (1, 2, 4 и 8 МГц). В ряде моделей предусмотрена возможность подстройки частоты этого генератора (подробнее см. [1, 2] или техническое описание конкретных моделей). Такой способ тактирования наиболее целесообразен в младших моделях *Tiny*, выпускающихся в 8-контактном корпусе, тогда выводы, предназначенные для подключения резонатора или внешнего генератора, можно задействовать для других целей, как обычные порты ввода-вывода.

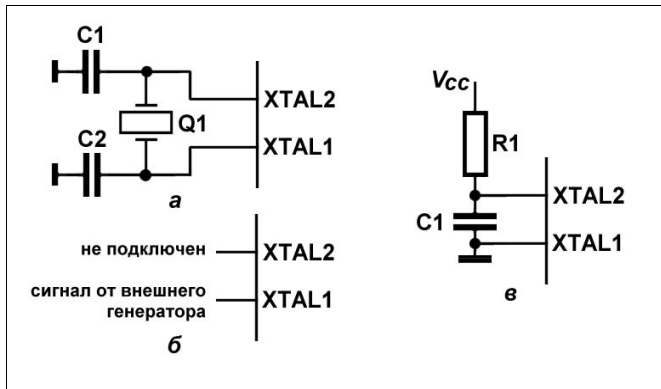


Рис. 2.3. Способы тактирования МК AVR с использованием: а — кварцевого резонатора; б — внешнего генератора; в — RC-цепочки

Семейство *Classic* встроенного RC-генератора не имело, а специальных конфигурационных ячеек у этих МК значительно меньше, и в общем случае на них можно было не обращать внимания. Сегодня это уже не так. По умолчанию МК семейств *Tiny* и *Mega* установлены в состояние для работы со встроенным генератором на частоте 1 МГц ( $CKSEL = 0001$ ), поэтому для других режимов нужно соответствующим образом установить конфигурационные ячейки  $CKSEL$  (табл. 2.1). При этом следует учитывать, что состояние ячеек  $CKSEL = 0000$  (зеркальное по отношению к наиболее часто употребляемому значению для кварцевого резонатора 1111) переводит МК в режим тактирования от внешнего генератора, и при этом его нельзя даже запрограммировать без подачи внешней частоты. О рекомендуемых установках конфигурационных ячеек и об особенностях их программирования см. также главу 5.

Таблица 2.1. Установка конфигурационных ячеек  $CKSEL$  в зависимости от режимов тактирования

$CKSEL3 \dots 0$	Источник тактирования	Частота
0000	Внешняя частота	0...16 МГц
0001	Встроенный RC-генератор	1 МГц
0010	То же	2 МГц
0011	То же	4 МГц
0100	То же	8 МГц
0101	Внешняя RC-цепочка	< 0,9 МГц