

# Оглавление

Предисловие .....	6
<b>Глава 1. Что нужно для работы с микроконтроллером</b> .....	<b>9</b>
1.1. Где найти минимальный набор программного обеспечения и документации для микроконтроллеров AVR .....	9
1.2. О выборе программатора .....	10
1.3. Источник питания .....	12
1.4. Дополнительные сведения .....	13
<b>Глава 2. Первый проект: контроллер сигнализации</b> .....	<b>14</b>
2.1. Постановка задачи .....	14
2.1.1. Устройства, подключаемые к контроллеру, и параметры входных и выходных сигналов .....	14
2.1.2. Логика работы контроллера .....	15
2.1.3. Схема сигнализации .....	16
2.1.4. Словесное описание алгоритма работы контроллера .....	17
2.2. Начинаем работу с AVR Studio .....	19
2.2.1. Создание первой программы на Ассемблере .....	19
2.2.2. Программа для контроллера сигнализации с использованием прерываний .....	50
2.3. Рекомендации .....	69
<b>Глава 3. Работа с внешним статическим ОЗУ</b> .....	<b>71</b>
3.1. Интерфейс микроконтроллера ATmega8515 для подключения внешней памяти ..	71
3.2. Пример схемы подключения внешней ОЗУ к микроконтроллеру ATmega8515 ..	73
3.2.1. Схема .....	73
3.2.2. Установка адреса .....	75
3.2.3. О выборе микросхемы регистра .....	75
3.2.4. Считывание данных из внешней памяти .....	76
3.2.5. Запись данных во внешнюю память .....	76
3.3. Программный доступ к оперативной памяти .....	76
3.3.1. Простая программа обращения к оперативной памяти .....	76
3.3.2. Отладка программы .....	78
3.3.3. Сохранение содержимого ОЗУ на диске .....	80

3.3.4. Запись данных в начальную область памяти данных . . . . .	80
3.4. Обращение к буферам для хранения данных как к ячейкам внешней памяти . . . . .	81
3.4.1. Электрическая схема подключения буферов . . . . .	81
3.4.2. Программа обслуживания буферов . . . . .	84
3.4.3. Отладка программы обслуживания буферов . . . . .	86
3.5. Подключение внешней памяти 512 Кбайт к микроконтроллеру ATmega8535 . . . . .	89
3.6. Схема подключения ОЗУ к микроконтроллеру ATmega8535 . . . . .	89
3.6.1. Описание схемы . . . . .	91
3.6.2. Запись в ячейку . . . . .	92
3.6.3. Считывание из ячейки . . . . .	92
3.7. Программа записи данных в ОЗУ 512 Кбайт . . . . .	93
3.7.1. Отладка программы . . . . .	96
3.7.2. Подпрограмма установки адреса SetAddr . . . . .	98
3.7.3. Подпрограмма копирования байта из внутреннего ОЗУ DataSt . . . . .	99
3.7.4. Подпрограмма копирования данных из внешней памяти во внутреннее ОЗУ DataLd . . . . .	100
<b>Глава 4. Устройство динамической индикации на 7-сегментных светодиодных индикаторах . . . . .</b>	<b>102</b>
4.1. Принцип динамической индикации . . . . .	102
4.2. Восьмиразрядное устройство отображения цифровой информации . . . . .	104
4.2.1. Схема управления восьмиразрядным индикатором . . . . .	104
4.2.2. Программа организации бегущей строки . . . . .	107
4.2.3. Описание программы . . . . .	110
4.3. Устройство управления двумя печами . . . . .	116
4.3.1. Работа устройства . . . . .	116
4.3.2. Программа управления двумя печами . . . . .	118
4.3.3. Работа с устройством управления двумя печами . . . . .	153
4.3.4. Особенности работы EEPROM микроконтроллера . . . . .	155
<b>Глава 5. Связь микроконтроллера с компьютером . . . . .</b>	<b>158</b>
5.1. Схема контроллера, обеспечивающая связь с СОМ-портом компьютера . . . . .	158
5.2. Программное обеспечение связи по каналу RS-232 . . . . .	161
5.2.1. Простая программа микроконтроллера для СОМ-порта . . . . .	163
5.3. Программное обеспечение для связи по интерфейсу RS-232 . . . . .	169
5.3.1. Протокол обмена . . . . .	169
5.3.2. Общие положения . . . . .	169
5.3.3. Структура сообщения . . . . .	170
5.3.4. Передаваемые сообщения (команды компьютера и ответы контроллера) . . . . .	170
5.3.5. Программа для микроконтроллера . . . . .	170
5.3.6. Отладка работы USART в AVR Studio . . . . .	182
5.4. Канал RS-232: программное обеспечение для компьютера . . . . .	183
5.4.1. Минимальные сведения о Delphi . . . . .	184
5.4.2. Программа обмена данными с микроконтроллером . . . . .	185
5.4.3. Описание работы программы . . . . .	191

---

5.4.4. Сохранение, запуск, использование программы. . . . .	193
5.4.5. Программа с использованием функций Windows API . . . . .	193
5.4.6. Описание работы программы . . . . .	197
<b>Глава 6. Организация аналоговых выходов для микроконтроллера . . . . .</b>	<b>200</b>
6.1. Преобразование кода в ширину импульса. . . . .	201
6.1.1. ЦАП и генератор пилообразного напряжения с ШИМ . . . . .	201
6.1.2. Таймер Т1 микроконтроллера в режиме PWM . . . . .	202
6.1.3. Программа для генератора ШИМ . . . . .	203
6.2. Преобразование кода в амплитуду импульса . . . . .	205
6.2.1. Генератор пилообразного напряжения. . . . .	205
6.2.2. Программа для генератора пилообразного напряжения . . . . .	208
6.2.3. Генератор синусоидального сигнала . . . . .	212
6.2.4. Программа для генератора синусоидального сигнала . . . . .	214
6.3. Определение пространственного модуля сигнала . . . . .	218
6.3.1. Алгоритм программы . . . . .	219
6.3.2. Листинг программы вычисления модуля . . . . .	220
6.4. Цифровой фильтр . . . . .	234
6.4.1. Листинг С-программы цифрового фильтра Filter.c . . . . .	235
Приложение 1. <b>Как получить необходимые материалы через сеть INTERNET . . . . .</b>	<b>243</b>
Приложение 2. <b>Устройства, облегчающие отладку контроллера в составе системы</b>	<b>245</b>
Приложение 3. <b>Программатор . . . . .</b>	<b>251</b>
Приложение 4. <b>Работа программатора в Windows XP/2000/NT . . . . .</b>	<b>274</b>
Приложение 5. <b>Fuse-байты: выбор режима работы микроконтроллера. . . . .</b>	<b>276</b>
Приложение 6. <b>8-битные RISC-микроконтроллеры фирмы Atmel. . . . .</b>	<b>284</b>

## Предисловие

Уважаемые читатели!

Книга, предлагаемая вашему вниманию, написана на основе опыта работы с 8-битными микроконтроллерами AVR серии ATmega компании Atmel а также с учетом вопросов, поступающих на сайт <http://bvn123.boom.ru>.



**Контроллер** — законченное электронное устройство, обычно выполненное в виде платы (плата контроллера) и предназначенное для приема и обработки сигналов от датчиков, а также для управления внешними устройствами на основании результатов обработки принятых сигналов.

**Микроконтроллер** — программно управляемая интегральная микросхема, применяемая для построения различных контроллеров.

Трудности, возникающие у разработчика при проектировании пятого или десятого контроллера, меркнут на фоне проблем, с которыми сталкивается новичок. Обычно возникают следующие вопросы:

- какой микроконтроллер выбрать;
- где найти его описание;
- где взять программу, обеспечивающую написание, отладку и редактирование программ для микроконтроллера;
- где взять программатор и программное обеспечение для него;
- как приступить к работе, когда все это уже есть;
- как все это сделать с минимальными затратами средств и времени.

Вопросам получения через сеть Internet минимального набора программ и документации, достаточного для работы с микроконтроллерами AVR, посвящена первая глава книги.

Так как поиск информации представляет определенные трудности для тех, кто имеет небольшой опыт работы в сети Internet, в Приложении 1 даны рекомендации по поиску необходимой информации.

Основная цель второй главы — обучение навыкам эффективной работы в среде разработки и отладки программ для микроконтроллеров AVR Studio 4.12. В этой главе возможности AVR Studio рассматриваются очень подробно, так, чтобы разобраться в них смог читатель, не обладающий достаточным опытом работы на компьютере. Освоение AVR Studio проходит на конкретном примере полного цикла разработки устройства сигнализации — от описания требований к устройству до отладки программ. Здесь же даны рекомендации по обнаружению и исправлению ошибок в программе.

Примерно одинаково построение остальных глав книги. В каждой из них предлагаются электрические схемы контроллеров на базе микроконтроллеров AVR, а также несколько различных программ, определяющих функционирование контроллеров. Функциональные узлы микроконтроллеров описаны в объеме, достаточном для понимания программ, полное их описание можно найти в техническом описании микроконтроллеров.

В Приложении 2 вы найдете описание устройств, облегчающих тестирование контроллеров, а также цифровых схем.

В Приложении 3 приведены сведения, которые помогут вам при разработке собственного программатора.

Приложение 4 описывает использование дополнительного программного обеспечения для работы программатора микроконтроллеров AVR в Windows XP.

Об использовании Fuse-битов для определения режима работы микроконтроллера рассказывается в Приложении 5.

В Приложении 6 приведена сравнительная таблица выпускаемых сегодня 8-битных микроконтроллеров AVR компании Atmel.

Все описываемые в книге программы для микроконтроллеров отлаживались в AVR Studio версии 4.12. Программное обеспечение для компьютера написано в Delphi.

Часть описываемых устройств являются фрагментами больших разработок. Эти фрагменты адаптированы специально для книги так, чтобы наиболее лаконично проиллюстрировать работу изучаемой функции микроконтроллера либо предлагаемого аппаратного или программного решения. При этом все устройства с приведенными программами вполне работоспособны и могут быть повторены.

Представляется наиболее продуктивным просмотр книги с одновременной обработкой изучаемого материала на компьютере. Особенно это

касается второй главы — ознакомление с нею без работы с компьютером вряд ли принесет пользу.

В начале каждого листинга программы, размещенного в книге, указывается имя файла \*.asm, содержащего эту программу. При изучении какой-либо программы, приводимой в книге, удобно открыть соответствующий проект в AVR Studio.

Материал каждой последующей главы дается в расчете на то, что читатель ознакомился с предыдущими главами, поэтому описание уже разобранных ранее фрагментов программ или схем опускается.

Ваши замечания и пожелания направляйте по адресу [bvn123@bk.ru](mailto:bvn123@bk.ru)  
Надеюсь, что книга окажется вам полезной.

Автор.

# Глава 1. Что нужно для работы с микроконтроллером

Для работы с 8-битными микроконтроллерами AVR корпорации Atmel (далее — микроконтроллер) вам понадобится программное обеспечение для разработки и отладки проектов, документация на микроконтроллеры, программатор и источник питания.

Вероятно, компакт-диск, содержащий программное обеспечение, и документацию на различные программаторы можно приобрести в любом крупном городе.

Однако все необходимое, включая схему программатора, а также приложение (компьютерную программу), обеспечивающее его работу, можно разыскать в сети Internet.

Ниже приводятся ссылки на материалы, которые вам понадобятся. Ссылки были проверены в июне 2006 года, если же они перестали работать, вам придется заняться поиском самостоятельно.

## 1.1. Где найти минимальный набор программного обеспечения и документации для микроконтроллеров AVR

Все необходимые материалы для разработки и отладки программного обеспечения на ассемблере можно разыскать на сайте корпорации Atmel (<http://www.atmel.com>).

Поскольку сайты производителей электронных компонентов периодически обновляются, лучше разыскивать информацию самостоятельно.

Обычно достаточно перейти в раздел Products или Product selector сайта ([http://www.atmel.com/products/product\\_selector.asp](http://www.atmel.com/products/product_selector.asp)) и выбрать интересующую вас группу компонентов: AVR 8-Bit RISC. Выбор группы обеспечит доступ ко всем необходимым ресурсам: Software & Application — программное обеспечение, примеры программ, Datasheets — техническая документация на микроконтроллеры, Parametric Table — сводная таблица параметров.

Для работы понадобится загрузить с сайта:

- последнюю версию интегрированной среды разработки программного обеспечения для микроконтроллеров AVR, AVR Studio 4.12 (build 460) (45 MB, updated 11/05) [http://www.atmel.com/dyn/resources/prod\\_documents/aStudio4b460.exe](http://www.atmel.com/dyn/resources/prod_documents/aStudio4b460.exe);
- последнее обновление AVR Studio: AVR Studio 4.12 Service Pack 1 (8 MB, updated 01/06) [http://www.atmel.com/dyn/resources/prod\\_documents/aStudio4.12SP1b462.exe](http://www.atmel.com/dyn/resources/prod_documents/aStudio4.12SP1b462.exe);
- файлы документации на применяемые микроконтроллеры. Разыщите в таблице параметров микроконтроллер, например, ATtiny2313. Выбрав его в таблице, вы попадете на страничку со ссылками на техническую документацию, далее загружайте полную версию (Complete) [http://www.atmel.com/dyn/resources/prod\\_documents/doc2543.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2543.pdf).

### 1.2. О выборе программатора

Возможный вариант программатора с программным обеспечением описан в Приложении 3. Программатор не предназначался для распространения. Поэтому в нем есть функции, которые оказались удобными при эксплуатации, например, введена дополнительная кнопка, нажатие которой вызывает стирание памяти микроконтроллера, перезагрузку измененного в результате ассемблирования файла программы и загрузку программы в микроконтроллер. При этом отсутствуют некоторые функции, имеющиеся в других программаторах, например, считывание всей памяти микроконтроллера (возможно считывание отдельных ячеек).

Попробуйте разыскать в сети другие варианты программаторов. Воспользуйтесь для этого сайтами, обеспечивающими поиск в сети по ключевым словам, например, <http://www.google.ru>.

В качестве ключевой можно использовать фразу «программатор avr».

Предлагаемые для микроконтроллеров AVR программаторы можно классифицировать по способу загрузки программ в память микроконтроллера и по подключению к компьютеру.

Программаторы с параллельной (побайтной) загрузкой программ практически не предлагаются. Их преимущества — более высокая скорость программирования и некоторые дополнительные возможности (в частности, они позволяют установить в микроконтроллере защиту от работы с последовательным программатором). К недостаткам можно отнести необходимость извлечения микроконтроллера из системы для перепрограммирования (то есть из платы контроллера, в составе которой работает микроконтроллер). Вероятно, основное назначение таких программаторов — массовое программирование микроконтроллеров перед их установкой в систему. Однако, используя программатор с последо-



вательной загрузкой, например, совместно с микроконтроллером АТМega8, можно перепрограммировать его линию RESET (системный сброс микроконтроллера) так, что в дальнейшем она будет функционировать как обычная линия ввода/вывода. После этого перепрограммировать микроконтроллер тем же программатором с последовательной загрузкой становится невозможным. Объясняется это тем, что для его работы требуется функционирование упомянутой линии именно в режиме RESET. Восстановить же работу линии в режиме RESET можно только с помощью параллельного программатора.

Для разработчика более удобен программатор с последовательной (побитной) загрузкой программ в микроконтроллер. При соблюдении некоторых мер на этапе проектирования схемы контроллера, о которых будет упомянуто позже, с помощью последовательного программатора микроконтроллер можно программировать, не извлекая его из системы.

Программаторы подключаются либо к LPT-порту, либо к COM-порту, либо к USB компьютера.

Простейший программатор представляет собой несколько проводников, подключаемых с одной стороны к LPT-порту, а с другой — к программируемому микроконтроллеру. Но такой программатор работает неустойчиво, а проводники должны быть как можно короче.

Вполне приемлемы программаторы, подключаемые к LPT-порту разъемом DB25. В корпусе такого разъема размещается печатная плата программатора. При длине кабеля до 1 м, напряжении питания +5 В и отсутствии посторонних цепей, подключаемых к линиям программатора, сбои при программировании не обнаруживаются. Основой такого программатора обычно является микросхема 74НС244, а при питании +5 В ее успешно заменяет микросхема КР1533АП5. Питается такой программатор вместе с программируемым микроконтроллером от одного источника напряжения.

Очень удобно, если после программирования микроконтроллера компьютерная программа, обслуживающая программатор, переводит все его выходные линии, связанные с микроконтроллером, в высокоимпедансное состояние — в этом случае нет необходимости каждый раз после программирования микроконтроллера отключать программатор.

Поскольку многие микроконтроллеры имеют встроенный универсальный асинхронный приемопередатчик, обмен данными с компьютером удобно организовывать через COM-порт. А так как не каждый компьютер снабжен двумя COM-портами, несомненным достоинством программатора, подключаемого к LPT-порту компьютера, является возможность отлаживать связь микроконтроллера с компьютером через COM-порт без отключения программатора и микроконтроллера от компьютера. При использовании программатора, подключаемого к COM-порту компьютера, вы будете вынуждены постоянно подсоединять то программатор, то мик-

роконтроллер к COM-порту, разъем которого к тому же находится на задней панели компьютера.

Если вы решили сделать программатор самостоятельно по материалам, найденным в сети Internet, нужно учитывать, что:

- с выбранной схемой программатора будет работать приложение, написанное именно для этой схемы;
- если приложение для программатора, подключаемого к LPT-порту, было написано под Windows 98, для его работы под Windows XP понадобится устанавливать дополнительное программное обеспечение, так как в таких приложениях обычно использовалось прямое обращение к портам ввода/вывода, а в Windows XP такое обращение блокируется;
- приложение должно поддерживать программирование как памяти программ, так и электрически стираемой памяти (EEPROM) всех выпускающихся микроконтроллеров группы;
- должно быть обеспечено считывание служебных битов микроконтроллера (Signature byte — информация о типе микроконтроллера, Calibration byte — индивидуальный параметр микроконтроллера, обеспечивает программную подстройку тактовой частоты в режиме работы с внутренним тактовым генератором, Fuse bytes — параметры, определяющие режим работы микроконтроллера);
- должна быть обеспечена запись Fuse bytes, битов защиты памяти программ от считывания Lock Bits;
- удобные функции — диагностика подключения питания и программатора к компьютеру, счетчик программирований, возможность перевода линий программатора в высокоимпедансное состояние, выполнение при нажатии одной кнопки последовательности действий: загрузка последней редакции файла программы в буфер — стирание памяти — загрузка программы в память микроконтроллера;
- дополнительные функции — расчет контрольной суммы, считывание программы из памяти, просмотр и сохранение в файл, проверка очистки памяти.

В приложении вы найдете схему, рисунок печатной платы и листинг компьютерной программы последовательного программатора, разработанного автором.

### 1.3. Источник питания

Обычно микроконтроллер и программатор питаются от одного источника напряжения (+5 В). Источник должен быть гальванически развязан от сети переменного тока 220 В 50 Гц, так как общий провод источника соединяется через программатор с общим проводом компьютером.

Желательно использовать источник с регулируемым током срабатывания защиты, что позволит обезопасить как микроконтроллер, так и программатор при ошибках в монтаже контроллера, при неверном подключении питающих проводников, а также при коротких замыканиях.

Следует предварительно рассчитывать ток потребления системы и устанавливать соответствующий ток срабатывания защиты перед включением.

Указанным условиям соответствуют источники питания постоянного тока серии Б5-хх (например, Б5-44). Впрочем, вы можете сделать источник питания самостоятельно или даже питать микроконтроллер и программатор от батареек.

Если блок питания вашего компьютера имеет внешний выход с напряжением +12 В, достаточно сделать внешний стабилизатор напряжения +5 В, разместив на небольшом радиаторе микросхему 7805 и пару конденсаторов.

## 1.4. Дополнительные сведения

Статьи о микроконтроллерах AVR на русском языке, можно найти на сайте <http://www.atmel.ru>.

На этом же сайте можно быстро получить консультацию специалистов на русском языке.

Получить специфическую информацию, которой не располагают специалисты русского сайта, можно обратившись в службу технической поддержки корпорации Atmel. О какой информации идет речь? Например, о параметрах надежности микроконтроллеров или, скажем, о допустимых токах через диоды, которые используются внутри микроконтроллера для защиты входов АЦП.

Полезно просматривать материалы телеконференций по микроконтроллерам, их адреса можно найти с помощью поисковых машин, введя в строку запроса фразу «конференция по микроконтроллерам».

# Глава 2. Первый проект: контроллер сигнализации

Основные цели этой главы:

- ознакомить читателя с основными этапами разработки контроллера и максимально быстро обучить приемам работы в среде разработки программ для микроконтроллеров AVR Studio 4.12;
- разобраться с организацией и работой портов ввода/вывода;
- понять принцип работы аппаратных прерываний микроконтроллера;
- пройти этапы разработки контроллера от схемы до отладки программы.

Разработка простого устройства сигнализации, подходящего для защиты помещений от вторжения — удобный пример, не перенасыщенный техническими деталями.

## 2.1. Постановка задачи

Пусть требуется изготовить простой контроллер сигнализации, работающей в следующих режимах:

- ожидание;
- вторжение;
- штатное отпирание двери;
- отпирание изнутри.

### 2.1.1. Устройства, подключаемые к контроллеру, и параметры входных и выходных сигналов

Питание сигнализации производится от аккумулятора (+12 В). Аккумулятор подзаряжается от сети переменного тока, через зарядное устройство, которое здесь не рассматривается. Исполнительные устройства сигнализации питаются непосредственно от аккумулятора.

На дверь устанавливается кнопка. При запертой двери кнопка нажата, а ее контакты разомкнуты, при отпирании двери или взломе контакты замкнуты.

Контроллер должен управлять электрическим замком. Открывание производится подачей на соленоид замка напряжения +12 В. При отсутс-

твии напряжения на соленоиде, замок запирается автоматически под действием пружинного механизма.

Пара контактов кодового устройства замыкается на короткое время, после того, как кнопки устройства нажаты в определенной последовательности.

При нажатии кнопки открывания замка внутри помещения замок открывается без ввода кода.

Для сигнализации используется светодиод на удаленном пульте, а также сирена. Для включения sireны на нее должна быть подана импульсная последовательность, частота повторения импульсов 1 кГц, амплитуда импульсов 12 В.

При отказе или разрядке аккумулятора замок открывается изнутри вручную, сирена и светодиод не включатся, так как питания на них не будет.

### 2.1.2. Логика работы контроллера

#### Режим ожидания

В этом режиме контакты кнопки, установленной на двери, замкнуты, контакты кодового устройства разомкнуты, сирена и светодиод выключены, ток через соленоид замка протекать не должен.

#### Режим вторжения

При взломе двери контакты кнопки, установленной на двери, замкнуты. Если предшествующего замыкания контактов кодового устройства не было, это должно вызвать включение sireны и светодиода на удаленном пульте. Сирена должна работать в прерывистом режиме: продолжительность звука и продолжительность пауз равны и должны составлять 0.5 с. Выключение sireны и светодиода должно производиться внутри помещения отдельной кнопкой, расположенной на плате контроллера.

#### Режим штатного отпирания двери

После ввода кода в правильной последовательности кратковременно замыкается пара контактов кодового устройства, к соленоиду замка должно быть приложено напряжение 12 В в течение 2 с, замок отперется. Если дверь после этого будет открыта, замыкание контактов кнопки, установленной на двери, не приведет к срабатыванию sireны и светодиода.

#### Режим отпирания двери изнутри

При замыкании контактов кнопки отпирания изнутри к соленоиду замка должно быть приложено в течение 2 с напряжение 12 В, замок отперется. Если дверь после этого будет открыта, замыкание контактов кнопки, установленной на двери, не приведет к срабатыванию sireны и светодиода.

### 2.1.3. Схема сигнализации

На Рис. 1 приведена схема, удовлетворяющая перечисленным условиям.

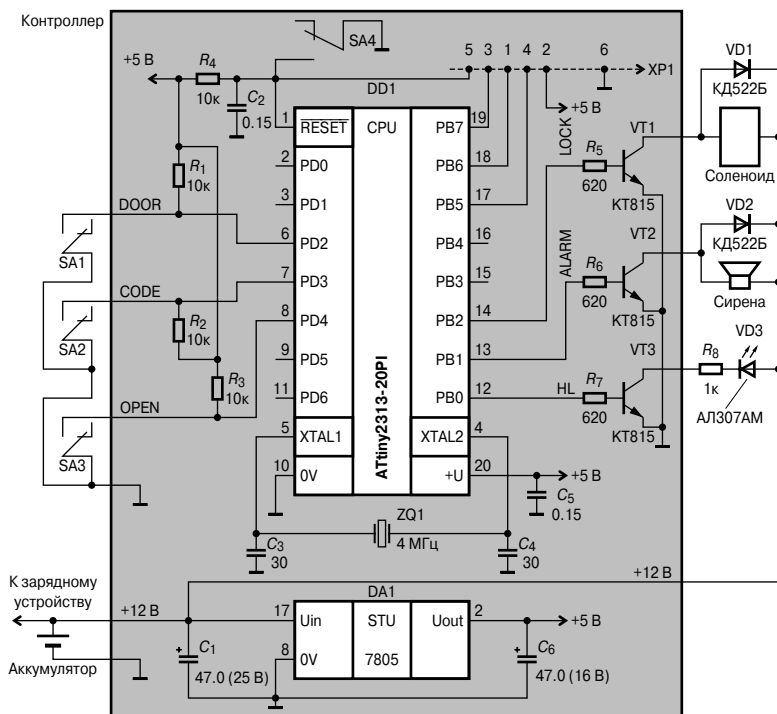


Рис. 1. Схема контроллера сигнализации

Кнопка SA1 установлена на двери, кнопка SA2 представляет пару контактов в кодовом устройстве, кнопка SA3 установлена внутри помещения. Положение контактов всех трех кнопок на схеме соответствует режиму ожидания (дверь заперта).

Параллельно соленоиду и сирене (индуктивные нагрузки) подключены диоды, они защищают транзисторы VT1, VT2 и VT3 контроллера от токов, возникающих в индуктивных нагрузках после снятия с них напряжения.

Питание схемы осуществляется от аккумулятора.

Штриховой линией окружены компоненты, размещаемые на плате контроллера. 10-контактный разъем XP1 предназначен для подключения к микроконтроллеру программатора.

В устройстве используется микроконтроллер ATtiny2313, для питания которого необходимо напряжение, не превышающее 6 В. Для получения стабилизированного напряжения питания +5 В применена микросхема 7805 (DA1).

Если цепь DOOR замкнута на общий провод кнопкой SA1, то в цепи устанавливается НИЗКИЙ логический уровень; при отсутствии такого замыкания в цепи поддерживается ВЫСОКИЙ логический уровень через резистор R1, соединенный с источником напряжения +5 В. То же справедливо для кнопок SA2, SA3, резисторов R2, R3 и цепей CODE и OPEN.

НИЗКИЙ логический уровень близок к 0 В, ВЫСОКИЙ логический уровень близок к напряжению питания микроконтроллера, составляющему для этой схемы +5 В.

Резистор R4 с конденсатором C2 обеспечивают аппаратный сброс микроконтроллера при включении питания. При отсутствии напряжения или его пропадании конденсатор C2 оказывается разряженным. После появления напряжения питания на контакте  $\overline{\text{RESET}}$  микроконтроллера удерживается НИЗКИЙ логический уровень до тех пор, пока конденсатор C2 не зарядится через резистор R4.

Для сброса программы (выключения сирены и светодиода после вторжения) в схему введена кнопка SA4, при замыкании контактов которой на линии RESET микроконтроллера устанавливается НИЗКИЙ уровень, что вызывает аппаратный сброс микроконтроллера.

Для управления мощной нагрузкой, питающейся большим напряжением, служат транзисторы VT1, VT2 и VT3.

Если на выходе микроконтроллера, соединенного через резистор с базой транзистора, сформируется ВЫСОКИЙ уровень, транзистор откроется. Тогда подключенная к его коллектору цепь нагрузки соединится с общим проводом через малое сопротивление транзисторного перехода коллектор-эмиттер, почти все напряжение аккумулятора будет приложено к нагрузке.

Когда на выходе микроконтроллера НИЗКИЙ уровень, транзистор закрыт, сопротивление его перехода коллектор-эмиттер очень велико, напряжение на нагрузке оказывается близким к нулю.



Таким образом, для срабатывания одной из нагрузок, подключенных к схеме, на соответствующем выходе микроконтроллера надо сформировать ВЫСОКИЙ уровень, для прекращения работы нагрузки — НИЗКИЙ уровень.

### 2.1.4. Словесное описание алгоритма работы контроллера

После включения питания по аппаратному сбросу

- линии порта D микроконтроллера DD1, соединенные с контактами кнопок SA1, SA2 и SA3, должны быть сконфигурированы как входные;

- линии порта В, управляющие нагрузками, необходимо сконфигурировать как выходные;
- на линиях порта В установить НИЗКИЕ уровни (нагрузки — соленоид замка, сирена и светодиод — выключены).

### Режим ожидания

Контроллер находится в состоянии ожидания, необходимо организовать циклическую проверку состояния линий порта D.

### Режим штатного отпирания двери

При появлении НИЗКОГО уровня в цепи CODE (введен правильный код, переход в режим штатного открывания двери) сформировать ВЫСОКИЙ уровень на линии OPEN, управляющей соленоидом замка; удерживать уровень в течение 2 с для отпирания замка.

Через пять секунд после запираания замка проверить уровень в цепи DOOR. Если в цепи ВЫСОКИЙ уровень (дверь не открывали или успели закрыть), контроллер должен вернуться в режим ожидания. Если уровень НИЗКИЙ (дверь еще открыта) — ожидать появления ВЫСОКОГО уровня (запираания двери), после чего перейти в режим ожидания.

### Режим отпирания двери изнутри

При появлении НИЗКОГО уровня в цепи INSIDE сформировать ВЫСОКИЙ уровень на линии OPEN, управляющей соленоидом замка; удерживать уровень в течение 2 с для отпирания замка.

Через пять секунд после запираания замка проверить уровень в цепи DOOR. Если в цепи ВЫСОКИЙ уровень (дверь не открывали или успели закрыть), контроллер должен вернуться в режим ожидания. Если уровень НИЗКИЙ (дверь еще открыта) — ожидать появления ВЫСОКОГО уровня (запираания двери), после чего перейти в режим ожидания.

### Режим вторжения

При появлении НИЗКОГО уровня в цепи DOOR (дверь открыта) сформировать ВЫСОКИЙ уровень на линии HL для включения светодиода, на линии ALARM сформировать импульсную последовательность, вызывающую прерывистый звуковой сигнал сирены. Для этого чередовать импульсные последовательности с частотой повторения 1 кГц в течение 0.5 с (паузы также 0.5 с).

Возвращение в режим ожидания должно происходить при временном появлении НИЗКОГО уровня на контакте RESET микроконтроллера, что обеспечивается нажатием кнопки SA4 или выключением-включением питания.



## 2.2. Начинаем работу с AVR Studio

На прилагаемом компакт-диске в папке \avr\ размещены проекты, рассматриваемые в книге. Проекты выполнены в предположении, что AVR Studio будет установлена в папку c:\Program Files\.

Установите AVR Studio (aStudio4b460.exe) в папку c:\Program Files\, а затем выполните ее обновление (aStudio4.12SP1b462.exe).

### 2.2.1. Создание первой программы на Ассемблере

#### Подготовка к работе

При установке AVR Studio 4 в папке c:\Program Files\Atmel\AVR Tools\AvrAssembler2\Appnotes размещаются файлы \*.def.inc, в которых хранятся predetermined имена регистров и констант микроконтроллеров AVR. Это имя папки для файлов \*.def.inc будет использоваться в примерах программ, приведенных в книге. Именно эти файлы совместимы с Assembler Version 2, который рекомендован для новых проектов и будет использоваться для ассемблирования программы. Выбор версии ассемблера будет рассмотрен ниже.

Файлы \*.def.inc с такими же именами, но иным содержимым хранятся и в папке c:\Program Files\Atmel\AVR Tools\AvrAssembler\Appnotes. Эти файлы \*.def.inc нужно использовать с Assembler Version 1, сам ассемблер версии 1 предназначен для обеспечения совместимости со старыми проектами.

Запустите AVR Studio. Автоматически появившееся окно Project Wizard предоставляет возможность открыть существующий или создать новый проект. Если флажок «Show this dialog on open», расположенный в левом нижнем углу, оставить установленным, окно будет появляться при каждом запуске программы. Так как в течение одного сеанса может понадобиться работать с разными проектами, закроем это окно, щелкнув по кнопке Cancel, а затем создадим новый проект, пользуясь меню AVR Studio. В оставшемся на экране окне AVR Studio в первой строке находятся заголовки меню File, Project, View, Tools, Debug и Help. Для ознакомления с возможностями AVR Studio откройте меню Help/AVR Studio User Guide.



**Не пренебрегайте меню Help, обращайтесь к нему при любой возможности.** Многие разработчики осваивают лишь некий набор возможностей используемой ими программы, позволяющий выполнять текущие задачи. Это касается использования любого программного пакета. Периодический просмотр тем из меню Help, которым раньше не уделялось внимание, поможет существенно повысить уровень знаний, а в итоге облегчить и ускорить работу.

### Создание проекта

Наш проект назовем Alarm, программу на ассемблере — alarm.asm. Откройте меню Project/New Project. На экране появится окно Create new project (Рис. 2).



Рис. 2. Создание нового проекта

Введите имя проекта Alarm в строку ввода Project name, это же имя автоматически копируется в окошко Initial file — по умолчанию имя будет присвоено и файлу, который будет содержать программу на ассемблере (Alarm.asm).

Для определения папки, в которой будет находиться проект, щелкните по кнопке, находящейся справа от окошка Location. В открывшемся окне Select folder (Рис. 3) перейдите в окошко Folder Name и введите имя папки, в которой разместится проект: c:\avr\Alarm. После щелчка по кнопке Select подтвердите создание новой папки, нажав кнопку Да в появившемся окне, затем произойдет возврат в окно Create new project, а в окошке Location появится имя папки, в которой будут храниться файлы нашего проекта.

В окошке Project type выберите тип проекта Atmel AVR Assembler, щелкните мышкой по кнопке Next. В открывшемся окне Select debug platform and device (Рис. 4) выберите в окошке Debug Platform строку AVR Simulator, затем в окошке Device строку ATtiny2313 — микроконтроллер, на котором выполнено наше устройство.

Теперь на экране активно окно C:\AVR\Alarm\Alarm.asm. В нем набирается и редактируется текст программы на ассемблере.



Рис. 3. Определение папки проекта

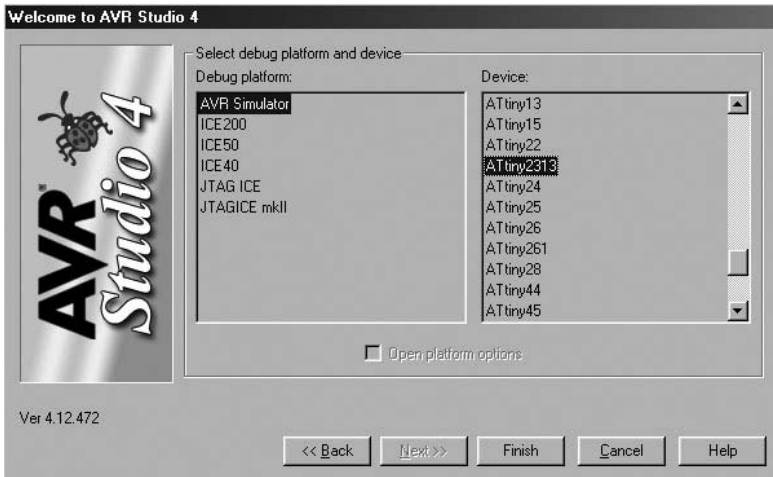


Рис. 4. Определение типа микроконтроллера

Текст программы можно скопировать с прилагаемого компакт-диска из файла `\avr\Alarm\Alarm.asm`.

### Ввод программы

Если программа вводится вручную, не опускайте символ точки и символ точки с запятой в начале строк: символом точки начинаются директивы ассемблера, символом точки с запятой начинаются комментарии.

Если комментарий не помещается на одной строке, его продолжение на следующей строке надо начинать символом точки с запятой.