Способ защиты программного обеспечения микроконтроллеров

Сергей Шишкин (Нижегородская обл.)

В статье представлен способ защиты программного обеспечения в устройствах на основе микроконтроллеров. В качестве примера описана система защиты с использованием микроконтроллера ATMEGA 8535.

Перед каждым разработчиком (или группой разработчиков) электронной техники рано или поздно встаёт вопрос о защите прав на интеллектуальную собственность. Когда нет возможности собственными силами продвигать проект (макет или опытный образец какого-либо устройства) на рынке, то приходится прибегать к помощи дилеров или просто посредников. Последние демонстрируют потенциальным заказчикам технические характеристики разработанного устройства. И не всегда играют честно.

Здесь и возникает вопрос о защите прав на интеллектуальную собственность разработчика. Решение заключается в том, чтобы, передавая проект в чужие руки, снабдить его программным или аппаратным механизмом защиты от копирования ПО (прошивки). Желательно, чтобы степень защиты была достаточно высокой.

В общем случае задача ставится следующим образом. Дилеру или посреднику даётся для демонстрации устройство, в микроконтроллере которого «зашита» демонстрационная версия основной программы. Устройство должно работать определённое время, в течение которого посредник демонстрирует потенциальные возможности устройства; затем оно теряет свою работоспособность. Фактор времени можно привязать к количеству включений/выключений устройства. После определённого количества включений (заданного разработчиком) устройство перестаёт работать по своему рабочему алгоритму.

Отметим, что предлагаемый вариант защиты работает только в изделиях, разработанных на микроконтроллерах, снабжённых встроенной энергонезависимой памятью (EEPROM). Микроконтроллеры семейства AVR с энергонезависимой памятью дают разработчику большую свободу действий. Для абсолютной надёжности целесообразно, чтобы механизм защиты использовал имеющиеся в изделии программные и аппаратные ресурсы и исключал взаимодействие алгоритма защиты (замена кодов доступа, установка и снятие защиты и т.д.) с каким-либо внешним интерфейсом.

Сформулируем общие технические требования к устройству, в котором можно применить предлагаемую защиту. Пусть в изделии имеются трёхразрядный семисегментный индикатор, две пользовательские кнопки и незадействованный вывод микроконтроллера для подключения дополнительного, «секретного» выключателя. Этих ресурсов более чем достаточно для реализации представляемого механизма защиты. Разработка подобной защиты фактически сводится к незначительной доработке ПО изделия, при этом задействуются свободные программные ресурсы микроконтроллера.

Принципиальная схема устройства на микроконтроллере ATMEGA8535, где реализован механизм защиты, приведена на рисунке 1. Программные и аппаратные ресурсы вышеуказанного микроконтроллера позволяют разработать достаточно надёжный механизм защиты с простым и удобным интерфейсом.

В интерфейс механизма защиты входят следующие элементы. SA1 – «секретный» выключатель, S1, S2 – пользовательские кнопки, задействованные в алгоритме управления устройством. Блок индикации (дисплей) из цифровых семисегментных индикаторов HG1, HG2 (дисплей). В принципиальной схеме (см. рис. 1) применены сдвоенные семисегментные индикаторы типа DA56-11GWA. Поэтому в трёхразрядном индикаторе в корпусе HG1 задействован один индикатор, в корпусе HG2 – два. Устройство, представленное на рисунке 1, является функционально законченным и для демонстрации механизма защиты может работать самостоятельно.

Алгоритм работы механизма защиты достаточно прост. В нём можно выделить два режима работы, которые задаются переключателем SA1. Если переключатель SA1 находится в положении «1», то механизм защиты находится в основном рабочем режиме (режим № 1), если в положении «2», то в режиме задания параметров (режим № 2).

При установленной защите, в режиме № 1, сразу после включения устройства (после подачи питания) на трёхразрядном индикаторе в течение 2 с будет отображаться некоторое число (оно может быть задано от 1 до 999, кроме кода № 3, о нём будет рассказано дальше). Вышеуказанное число будет уменьшаться с каждым последующим включением питания. Как только его значение станет равным нулю, устройство сразу после включения питания «зависнет», т.е. не будет отрабатывать заданный алгоритм работы. На дисплее устройства при этом будут индицироваться символы – – – (при каждом последующем включении питания). Увеличить количество включений или снять защиту может только разработчик. Начальное значение числа (от 1 до 999) также задаёт разработчик. Поэтому у дилера/посредника есть только ограниченная возможность работать с изделием, включая его заданное число раз.

Если число включений не равно нулю, то устройство начинает работать в соответствии со своим рабочим алгоритмом и на дисплее отображается число 555. Индицирование символов – – – и числа 555 необходимо лишь для наглядной демонстрации алгоритма работы устройства защиты. Если программа защиты встроена в программу какого-либо изделия, то переход на подпрограмму отображения символов – – – означает блокировку алгоритма работы всего изделия. Соответственно переход на отображение числа 555 в программе изделия означает выполнение рабочего алгоритма основной программы.

Защита снимается следующим образом. Перед включением питания необходимо установить выключатель SA1 в

© СТА-ПРЕСС

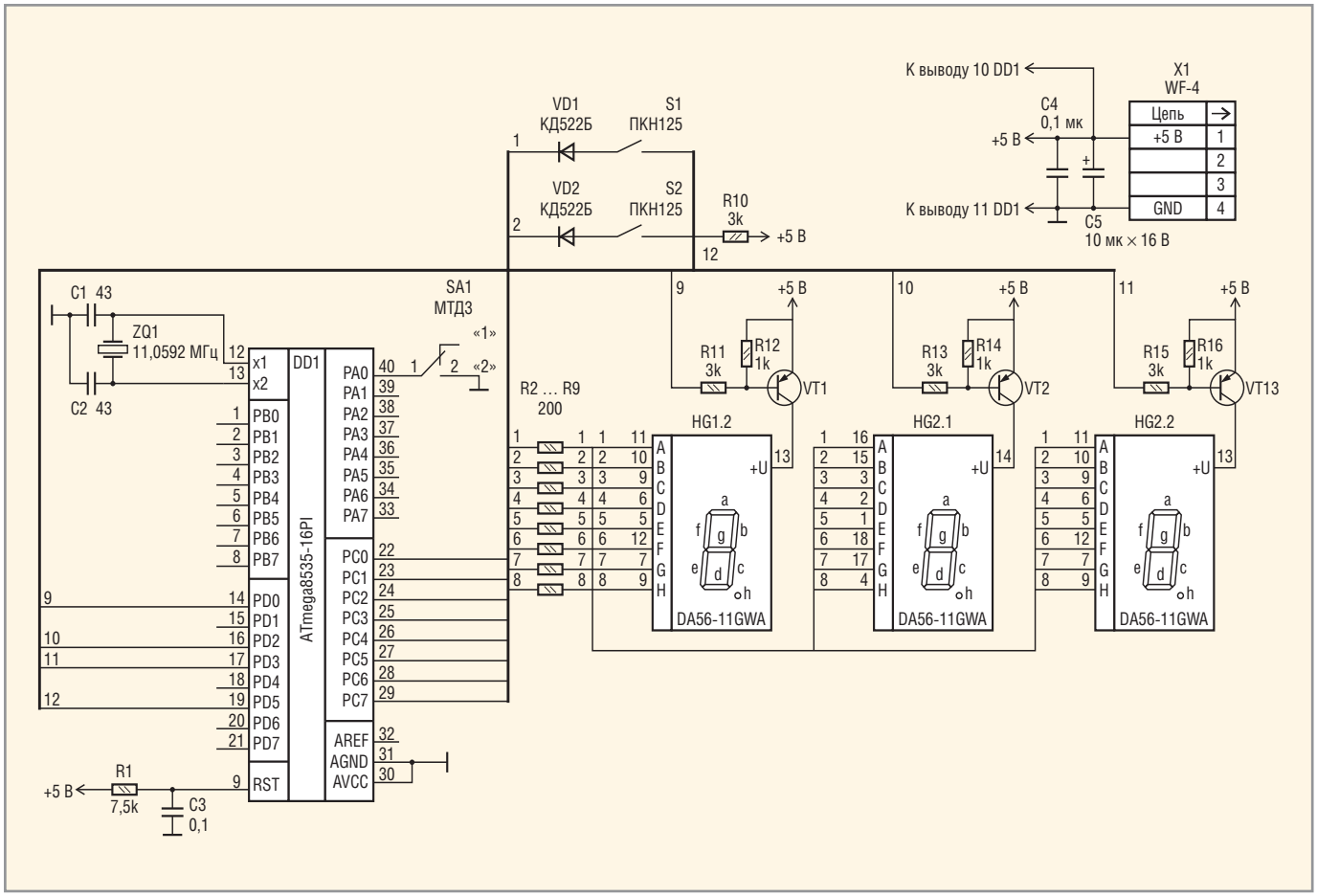

Рис. 1. Принципиальная схема устройства на микроконтроллере ATMEGA8535

положение «2» (режим № 2). При этом пользовательские кнопки будут иметь следующие назначения: S1 – увеличение (инкремент) вводимого числа, которое индицируется на дисплее; S2 – ввод (активация) набранного числа.

С помощью кнопок S1 и S2 необходимо набрать и ввести трёхразрядный код № 1. Затем с помощью кнопок необходимо набрать и ввести трёхразрядный код № 2 (фактически в два приёма вводится шестиразрядный код). После ввода каждого кода (неважно, верно или нет) индикаторы показывают нули.

Если коды введены правильно, то устройство перейдёт в режим работы, в котором можно задать число включений или снять защиту. Далее необходимо задать кнопками S1, S2 любое число от 1 до 999. Причём в этом диапазоне есть число, которое снимает защиту. Это число и есть код № 3. Набирая количество включений, равное коду № 3, мы снимаем защиту. Вводя любое другое (в диапазоне от 1 до 999), мы задаём число включений. Чтобы снять защиту, фактически необходимо знать девятиразрядный код, который «вычислить» затруднительно. Коды № 1...№ 3 знает только разработчик.

Далее следует выключить устройство, установить выключатель SA1 в положение «1» и снова включить. Если защита не установлена или снята, то устройство будет отрабатывать свой рабочий алгоритм (отображается число 555) без предварительного индицирования количества включений и намека на какую-либо установленную защиту. Кнопки S1 и S2 будут выполнять функции в соответствии с заданным алгоритмом функционирования изделия. Если за-

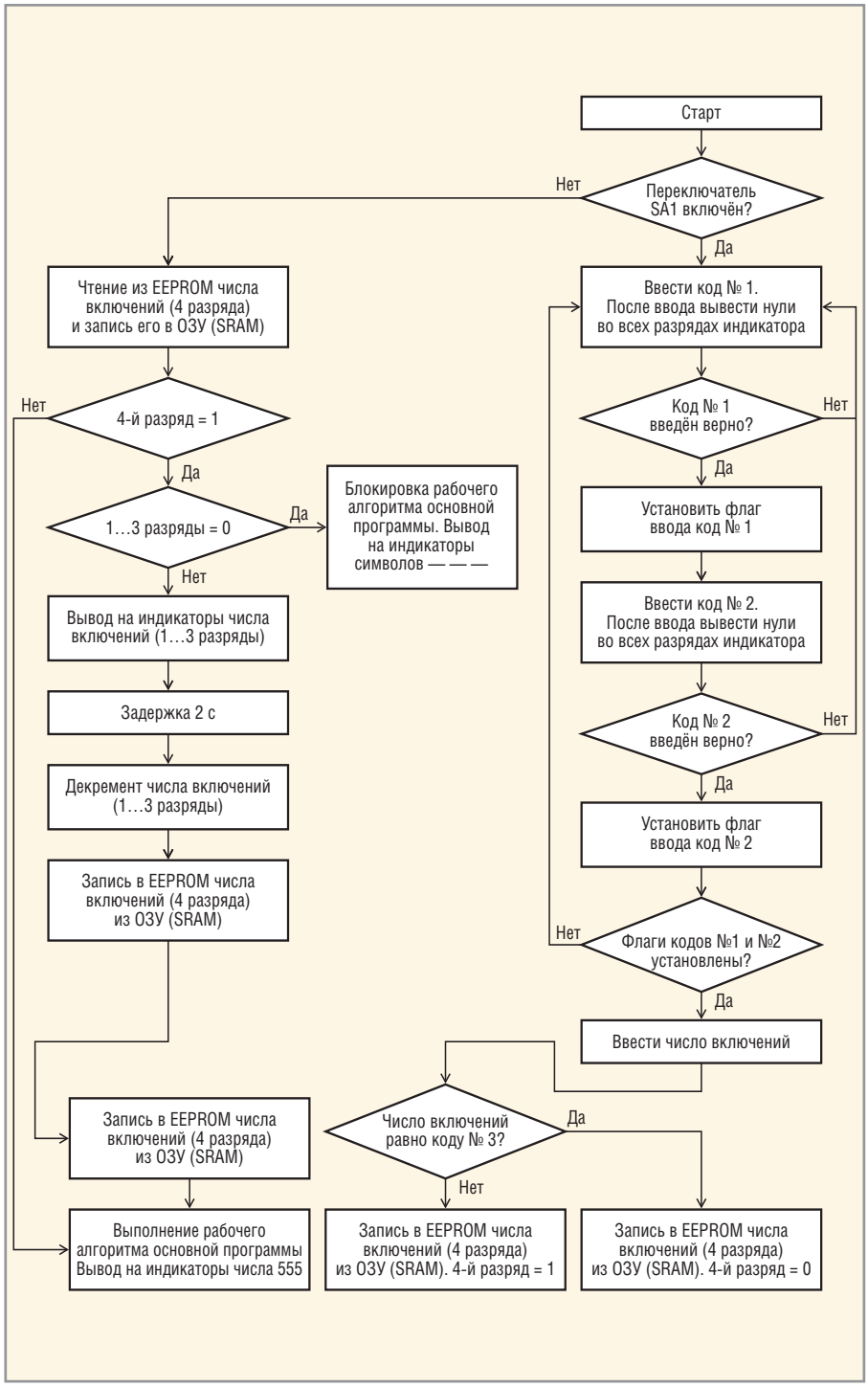

Рис. 2. Блок-схема представляемой защиты

щита установлена и задано число включений, то сразу после включения устройства в течениие 2 с будет отображаться текущее число включений, которое будет уменьшаться с каждым новым включением. Дальше устройство будет отрабатывать свой рабочий алгоритм. Разработчик, используя только аппаратные ресурсы устройства, может не один раз записать необходимое число включений и снять защиту. (В техническом описании на микроконтроллер ATMEGA8535 утверждается, что его EEPROM выдерживает 100 000 циклов записи /стирания.)

Целесообразно, чтобы доступ к выключателю SA1 был ограничен. Конструктивно это сделать не сложно. Все пересылки данных происходят внутри микроконтроллера. У «злоумышленника» нет никакой возможности их контролировать и отследить момент сравнения вводимого кода с кодами, хранящимися в памяти. Не поможет и знание рабочего алгоритма устройства. Вводимые коды находятся во внутренней памяти программ микроконтроллера, под битами защиты, про которые не следует забывать при программировании микроконтроллера.

Даже если «злоумышленник» определит выключатель SA1 – не беда. Чтобы войти в режим задания числа включений, необходимо два раза ввести трёхразрядный код, а чтобы снять защиту – три раза. В перспективе, разобравшись в программе для увеличения степени защиты, код доступа можно сделать 12- или 15-разрядным. Степень защиты можно еще увеличить, если в устройстве задействован четырёхразрядный индикатор. Перебор всех возможных комбинаций, даже при шестиразрядном коде, просто нереален.

Конструктивно выключатель SA1 можно вообще можно исключить: например, сделать два штыря в разных местах платы, замыкать их проводником, подключая соответствующий вывод микроконтроллера к общему проводу устройства.

Блок-схема представляемой защиты представлена на рисунке 2.

Программное обеспечение микроконтроллера было разработано в среде AVR Studio. В программе используются два прерывания: Reset и прерывание таймера T0, обработчик которого начинается с метки TIM0. При переходе на метку Reset инициализируются стек, таймер, порты, а также флаги и переменные, используемые в программе. Таймер T0 генерирует прерывания по переполнению (в регистре TIMSK установлен бит TOIE0). Коэффициент предварительного деления тактовой частоты таймера установлен равным 64 (в регистре TCCR0 записано число 3).

В обработчике прерывания таймера T0 осуществляются: процедура опроса кнопок S1, S2, выключателя SA1, функционирование динамической индикации, запись числа включений в EEPROM микроконтроллера, чтение числа включений из EEPROM, перекодировка двоичного числа в код для отображения информации на семисегментных индикаторах устройства, а также временной интервал длительностью 2 с, необходимый для отображения числа включений на дисплее устройства. Флаги, задействованные в программе, находятся в регистрах R19 (flo) и R25 (flo1).

Число, индицируемое на дисплее устройства, имеет три разряда. Число, заносимое в EEPROM микроконтроллера, имеет четыре разряда. Каждый разряд занимает 1 байт в ОЗУ и, соответственно, в EEPROM. Первые три разряда задают количество включений. Четвёртый разряд не отображается на дисплее. Функциональное назначение данного

разряда следующее. Если разряд содержит единицу – значит, защита установлена, если ноль – защита снята (см. рис. 2). При инициализации в четвёртый разряд заносится единица. Как видно из блок-схемы, блокировка рабочего алгоритма основной программы происходит при обнулении числа включений.

В ОЗУ микроконтроллера с адреса RAM= $60 организованы четыре буфера отображения для динамической индикации. Буфер № 1 необходим для отображения чисел и кодов, которые надо инкрементировать (число включений, коды № 1...№ 3). Число включений из буфера № 1 в режиме № 2 заносится в EEPROM микроконтроллера. Функциональное назначение каждой ячейки буфера отображения № 1 следующее:

- $61 – ячейка, где хранятся «сотни» числа включений и кодов № 1...№ 3 (1-й разряд индикатора, слева направо);
- $62 – ячейка, где хранятся «десятки» числа включений и кодов № 1...№ 3 (2-й разряд индикатора);
- $63 – ячейка, где хранятся «единицы» числа включений и кодов № 1...№ 3 (3-й разряд индикатора);
- $64 – ячейка, где хранится число 0 или 1, определяющее установку или снятие защиты.

При инициализации в ячейку с адресом $64 записывается число 1, в остальные ячейки буфера № 1 заносятся нули. С адреса RAM+6 начинается буфер отображения № 2 для динамической индикации. В данный буфер в режиме № 1 из EEPROM микроконтроллера заносится (читается) число включений, которое индицируется в течение 2 с, затем декрементируется и записывается в EEPROM микроконтроллера. При инициализации в буфер № 2 заносятся нули. С адреса RAM+12 начинается буфер отображения № 3. При инициализации в каждую ячейку буфера № 3 заносится число $A, которое после перекодировки в каждом разряде индицируется как символ «–». В итоге на дисплее индицируются символы – – – (только при блокировке рабочего алгоритма основной программы).

С адреса RAM+17 начинается буфер отображения № 4. При инициализации в каждую ячейку буфера № 4 заносится число 5. В итоге на дисплее индицируется число 555. Данное число индицируется при переходе на рабочий алгоритм основной программы. Коды № 1...№ 3 в программе заданы соответственно 010, 011, 012. Метки перехода на отображение буферов № 3 и № 4 соответственно osn2 и osn3.

Информация, записанная в буферы отображения № 3 и № 4, как уже отмечалось выше, нужна лишь для наглядности во время демонстрации работы устройства.

Разработанная на ассемблере программа устройства защиты, вместе с подпрограммой динамической индикации для вывода буферов отображения № 1...№ 4 на дисплей, занимает 1 Кб памяти программ. Представляемая защита достаточно универсальна, её можно адаптировать для любого устройства с микроконтроллером с учётом вышеуказанных требований. Её легко доработать, увеличив степень защиты. Предлагаемую защиту можно встроить в тысячи приборов, изменяя при этом только коды доступа.

Литература

1. http:// www.atmel.com.
2. *Баранов В.Н.* Применение микроконтроллеров AVR: схемы, алгоритмы, программы. 2006.
3. *Белов А.В.* Создаем устройства на микроконтроллерах. Наука и техника, 2007.