Защита интеллектуальной собственности и программного обеспечения на базе микроконтроллеров с EEPROM

Сергей Шишкин (г. Саров)

В статье автор представляет способ защиты интеллектуальной собственности и программного обеспечения в устройствах с микроконтроллерами, которые имеют встроенную EEPROM. Приведён пример устройства, состоящего из трёх составных частей, в каждой из которых реализован механизм защиты.

В некоторых случаях разработчику, владельцу устройства, патентообладателю необходимо ограничить его эксплуатацию. Например, с такой проблемой можно столкнуться, отдавая устройство в чужие руки для демонстрации его функциональных возможностей потенциальным потребителям и заказчикам, или при работе устройства в отсутствии разработчика.

Ограничить эксплуатацию устройства можно, заранее определив число включений (включения питания) или выключений. После определённого числа включений (заданного разработчиком) устройство превращается в «мёртвое железо», т.е. перестаёт работать по своему заданному рабочему алгоритму. Если устройство состоит из нескольких составных частей, то число включений задаётся для каждой части отдельно. При этом, конечно, разработчику нужно учесть вопрос безопасной эксплуатации всего устройства в целом. Вышеуказанный пример представляет собой лишь частный случай, где можно применить предлагаемую защиту.

Предлагаемый вариант защиты работает только в изделиях, разработанных на микроконтроллерах, где есть встроенная внутренняя энергонезависимая память (EEPROM). Микроконтроллеры семейства AVR со встроенной внутренней энергонезависимой памятью (EEPROM) предоставляют разработчику самый широкий спектр возможностей для создания подобных аппаратно-программных устройств. Для максимальной надёжности наиболее целесообразно, чтобы механизм защиты использовал имеющиеся в устройстве программные и аппаратные ресурсы. Целесообразно исключить функционирование алгоритма защиты (замена кодов доступа, установка и снятие

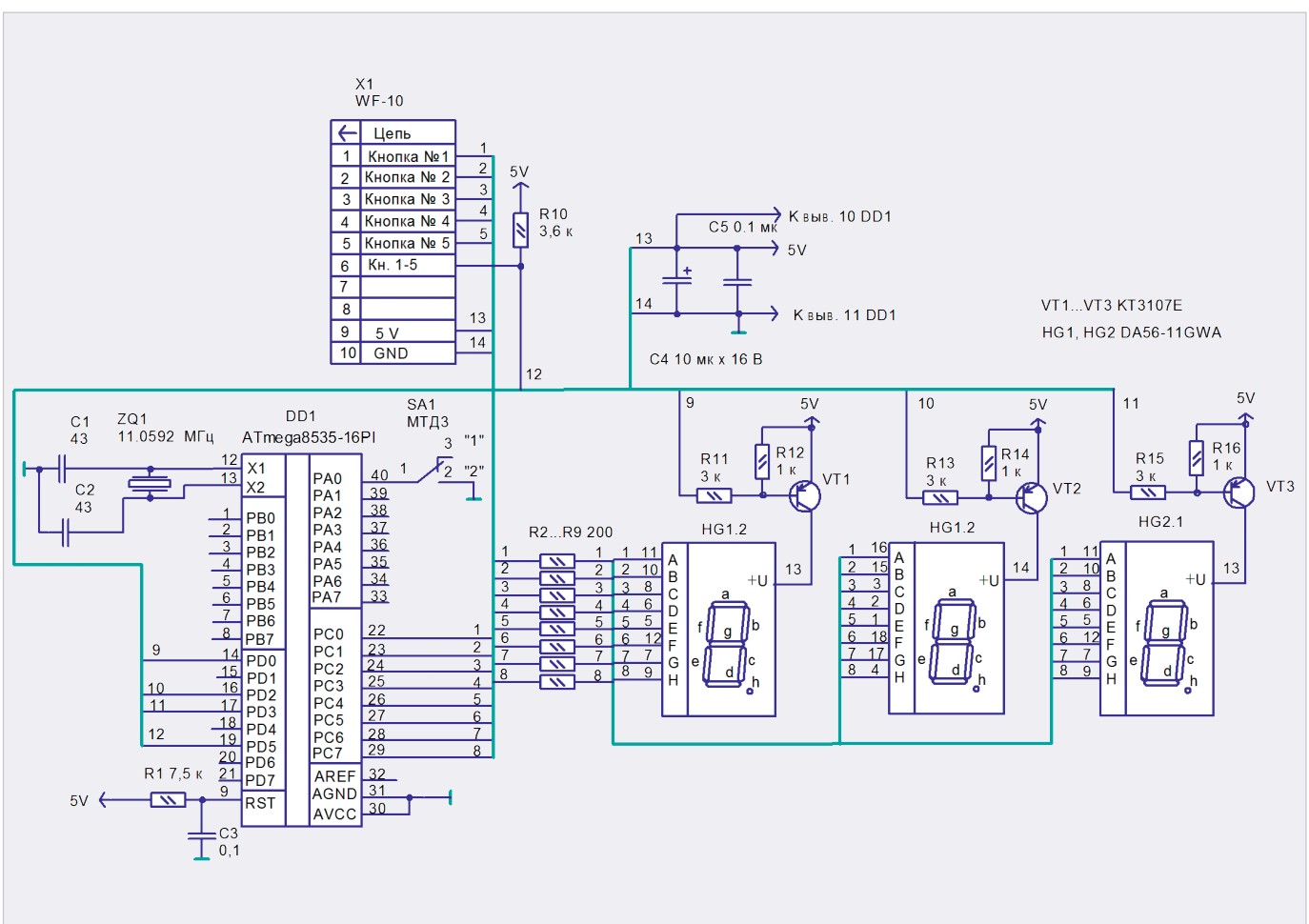

Рис. 1. Принципиальная схема аппаратной части контроллера № 1 с реализацией механизма защиты

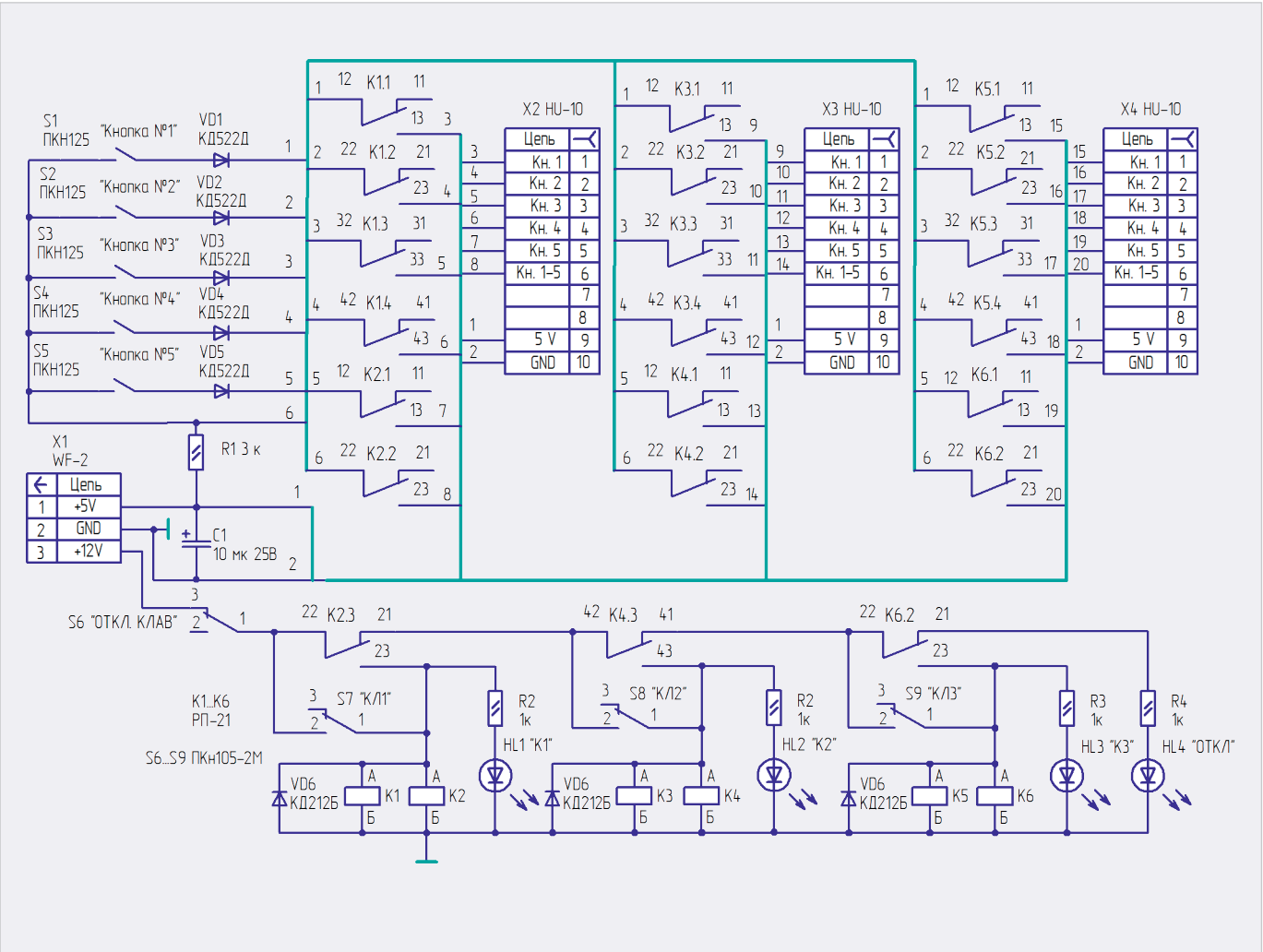

Рис. 2. Принципиальная схема платы клавиатуры с релейным подключением кнопок

защиты и т.д.) с помощью какого-либо внешнего интерфейса.

Сформируем общие технические требования к устройству, в котором можно применить предлагаемую защиту. Пусть оно состоит из четырёх составных частей трёх плат контроллеров, далее по тексту – контроллеры № 1...3, работающих по заранее заданному алгоритму, и платы клавиатуры. Аппаратная часть контроллеров № 1...3 в зависимости от решаемых задач может быть разной, но аппаратная и программная части механизма защиты – одинаковые.

В каждом контроллере имеется трёхразрядный семисегментный индикатор, две пользовательские кнопки, а также микроконтроллер семейства AVR, у которого имеется незадействованный вывод для подключения дополнительного «секретного» выключателя. Этих ресурсов более чем достаточно для реализации механизма защиты. Построение подобной защиты фактически сводится к незначительной доработке уже разработанного программного обеспечения изделия, при этом задействуются свободные программные ресурсы микроконтроллера. Принципиальная схема устройства контроллера № 1 на микроконтроллере ATMEGA8535, где реализован механизм защиты, приведена на рис. 1. Программные и аппаратные ресурсы вышеуказанного микроконтроллера позволяют разработать достаточно надёжный механизм защиты с простым и удобным интерфейсом.

Принципиальная схема платы клавиатуры с релейным подключением кнопок приведена на рис. 2.

Соединители платы клавиатуры X2...X4 подключаются к соединителям X1 контроллеров № 1...3. Кнопки S1...S5 платы клавиатуры – пользовательские кнопки, которые подключаются к контроллерам № 1...3 через группы контактов реле K1...K6. Кнопки S6...S9 управляют реле K1...K6. После подачи питания на устройство кнопки S1...S5 платы клавиатуры подключены к контроллеру № 1. При нажатии на кнопку S8 вышеуказанные кнопки подключаются к контроллеру № 2, а при нажатии на кнопку S8 – подключаются к контроллеру № 3.

Так как аппаратная и программная части механизма защиты контроллеров № 1...3 одинаковы, то далее будем рассматривать его работу в контроллере № 1. Пусть кнопки S1...S5 платы клавиатуры подключены к контроллеру № 1. В интерфейс механизма защиты входят следующие элементы. SA1 – «секретный» выключатель на плате контроллера, S1, S2 – пользовательские кнопки, задействованные в алгоритме управления контроллера, в котором необходимо установить защиту. Кнопки S3...S5 – кнопки, которые могут быть задействованы в рабочем алгоритме контроллера. Данные кнопки не задействованы в алгоритме работы механизма защиты.

Блок индикации (дисплей) выполнен на цифровых семисегментных индикаторах HG1, HG2. В принципиальной схеме (см. рис. 1) применены сдвоенные семисегментные индикаторы DA56-11GWA. В связи с этим в трёхразрядном индикаторе в корпусе HG1

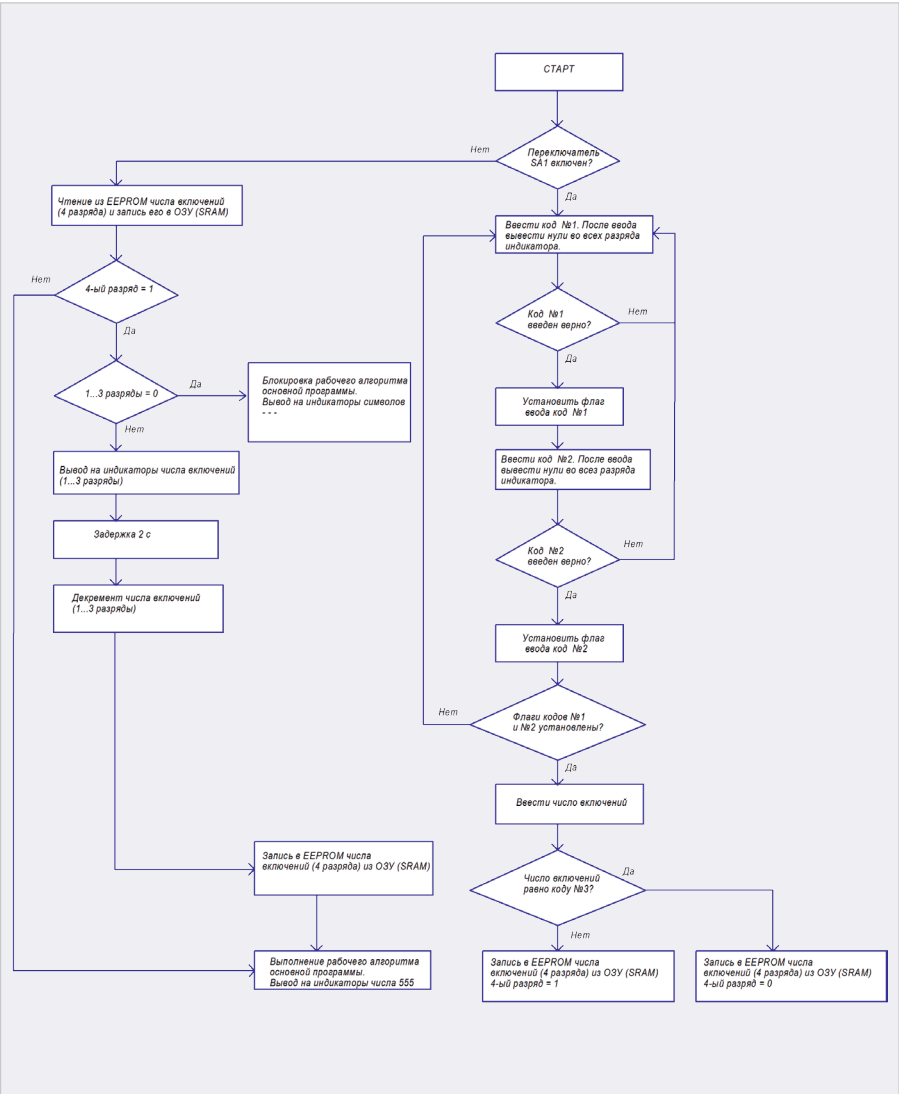

Рис. 3. Блок-схема устройства защиты

задействован один индикатор, а в корпусе HG2 – два. Устройство, представленное на рис. 1, является функционально законченным и для демонстрации алгоритма работы механизма защиты может работать самостоятельно.

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

При установленной защите в режиме № 1 сразу после включения устройства на трёхразрядном индикаторе в течение 2 с будет отображаться число, которое может быть задано любым в интервале от 1 до 999, кроме кода № 3 (см. далее). Данное число будет декрементироваться с каждым последующим включением питания. И как только его значение станет равным нулю, устройство сразу после включения питания не будет отрабатывать заданный алгоритм работы. На дисплее устройства при этом (и последующих включениях) будут отображаться символы «---». Увеличить количество включений или вообще снять защиту может только разработчик. Начальное значение числа (от 1 до 999) также задаёт разработчик. Таким образом, у дилера (посредника) есть только ограниченная по числу включений возможность работы с изделием.

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

Снятие защиты организовано следующим образом. Перед включением питания необходимо установить выключатель SA1 в положение «2» (режим № 2). При этом пользовательские кнопки будут иметь следующие назначения:

- S1 – увеличение (инкремент) вводимого числа, которое при этом индицируется на дисплее;
- S2 – ввод (активация) набранного числа.

С помощью кнопок S1 и S2 необходимо набрать и ввести трёхразрядный код № 1. Затем совершенно аналогично с помощью этих кнопок необходимо набрать и ввести трёхразрядный код № 2 (фактически в два приёма вводится шестиразрядный код). После ввода каждого кода индикаторы отображают нули.

Если коды введены верно, то устройство перейдёт в режим работы, в котором можно задать число включений или снять защиту. Далее следует задать кнопками S1, S2 любое число, как уже отмечалось выше, от 1 до 999. Причём в этом диапазоне есть число, которое снимает защиту. Это число и есть код № 3. То есть набирая количество включений, равное коду № 3, мы снимаем защиту. Любое другое число задаёт количество допустимых включений. Таким образом, для того чтобы снять защиту, нужно знать 9-разрядный код, который «вычислить» практически нереально. Коды № 1...3 знает только разработчик.

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

число включений, то, как отмечалось выше, сразу после включения устройства в течение 2 с будет индицироваться текущее число включений, которое декрементируется с каждым новым включением. Далее устройство будет отрабатывать свой рабочий алгоритм. Разработчик, используя только аппаратные ресурсы устройства, может бесконечное число раз записать необходимое число включений и снять защиту. В техническом описании на микроконтроллер ATMEGA8535 утверждается, что EEPROM выдерживает 100 000 циклов записи/стирания.

Целесообразно, чтобы доступ к выключателю SA1 был ограничен. Конструктивно это сделать не так уж и сложно. Все пересылки данных происходят внутри микроконтроллера. У злоумышленника нет никакой возможности их контролировать и отследить момент сравнения вводимого кода с хранящимися в памяти. Не поможет и знание рабочего алгоритма устройства. Вводимые коды находятся во внутренней памяти программ микроконтроллера, под битами защиты. Понятно, что нужно не забыть установить при программировании микроконтроллера биты защиты.

Даже если злоумышленник обнаружит выключатель SA1, то это не представляет большой опасности. Для того чтобы войти в режим задания числа включений, необходимо два раза ввести трёхразрядный код. А для того чтобы снять защиту – три раза трёхразрядный. Доработав программное обеспечение микроконтроллера, код доступа легко можно сделать и 12- или 15-разрядным. Понятно, что степень защиты можно ещё более увеличить, если в устройстве задействован четырёхразрядный индикатор. Таким образом, перебор всех возможных комбинаций даже при шестиразрядном коде просто невозможен. Конструктивно выключатель SA1 можно вообще исключить. Вместо него можно сделать, например, просто два штыря в разных местах платы, замыкать их проводником, подключая тем самым соответствующий вывод микроконтроллера к общему проводнику устройства. Алгоритм работы (блок-схема) представляемой защиты продемонстрирован на рис. 3.

Совсем коротко о программном обеспечении механизма защиты. 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 осуществляется: 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 две секунды, необходимый для индицирования числа включений на дисплее устройства. Флаги, задействованные в программе, находятся в регистрах R19 (flo) и R25 (flo1). Число, отображаемое на дисплее устройства, имеет три разряда. Число, заносимое в EEPROM микроконтроллера, имеет четыре разряда. Каждый разряд занимает один 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, начиная с адреса RAM = $60, организованы четыре буфера отображения для динамической индикации. Буфер № 1 необходим для отображения чисел и кодов, которые необходимо инкрементировать (число включений, коды № 1...3). Число включений из буфера № 1 в режиме № 2 заносится в EEPROM микроконтроллера.

Функциональное назначение ячеек 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 и 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 выводятся символы «---». Данные символы отображаются 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 отображается при переходе на рабочий алгоритм основной программы. Коды № 1...3 в программе заданы как 010, 011, 012 соответственно. 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, часть принципиальной схемы которого приведена на рис. 1.

Написанная на ассемблере программа устройства защиты вместе с подпрограммой динамической индикации для вывода буферов отображения № 1...4 на дисплей занимает всего 1 кБ памяти программ. Представляемая защита достаточно универсальна, её можно адаптировать для любого устройства с микроконтроллером и/или доработать для увеличения степени защиты. Предлагаемую защиту можно встроить в различные приборы, меняя при этом только коды доступа.