Безопасное обновление исполняемого кода в микроконтроллерах Atmel с ядром Cortex-M3 и Cortex-M4

При создании отказоустойчивых устройств и систем, требующих регулярного обновления рабочей программы (которое может осуществляться по нестабильным коммуникационным каналам), модификация программного кода в микроконтроллере является важным моментом. Обычно для процесса модификации программного кода микроконтроллера необходимо остановить работу устройства на время процедуры обновления. Для повышения надежности обновления кода программы в своих новых микроконтроллерах на базе ядра Cortex-M3 и Cortex-M4 компания Atmel применила Flash-память, разделенную на два банка, с возможностью одновременной записи в один банк и продолжения выполнения программы из другого.

Елена ЛАМБЕРТ
elena@efo.ru

Традиционно в микроконтроллерах Flash-память организована в виде одного банка. В таких микроконтроллерах наиболее частым методом обновления микропрограммы является полная остановка выполнения программы микроконтроллером и передача управления процедуре загрузчика. При этом загрузчик выполняет следующие действия: стирание текущего исполняемого кода, получение нового исполняемого кода и его обработку в соответствии с заданным алгоритмом, запись нового исполняемого кода во Flash-память, верификацию нового исполняемого кода и его запуск.

В устройствах, где Flash-память организована в виде одного банка, для обеспечения надежного обновления нужно предусмотреть, чтобы часть Flash-памяти не была стерта или перезаписана. Код в этой части памяти отвечает за проверку целостности кода в остальной части Flash-памяти. Использование циклически избыточной контрольной суммы или цифровой подписи является стандартным способом определения достоверности содержания Flash-памяти после перепрошивки или сброса.

В системах, где исполняемый код загружается через беспроводные соединения или возможны сбои по питанию, возникает вероятность ошибки при обновлении и записи в память микроконтроллера неработоспособной программы. Самый неприятный случай, когда в результате перепрошивки информация была искажена и произошел запуск некорректной программы.

Для защиты от ошибок, которые могут возникнуть при разрыве канала связи (по которому передается новая версия исполняемого кода) и сбое питания, также можно сохранять копию рабочей программы микроконтроллера во внешней или встроенной памяти. Такое решение позволяет всегда иметь рабочую версию программы. Из недостатков этого решения следует отметить увеличение используемого объема памяти программ в два раза.

Рассмотрим методы работы с сохранением резервной копии во Flash-памяти программ, организованной в виде одного и двух банков.

Организация Flash-памяти программ в виде одного банка

При наличии у процессора однобанковой Flash-памяти можно разделить ее на две области плюс область загрузчика (рис. 1):

- исполняемый код (область A);
- буфер для новой прошивки (область B).

Область B используется в качестве буфера. Новая версия исполняемого кода загружается в эту область и проверяется. Если проверка кода в области B прошла успешно, то код копируется в область A. Если нет, то копирования не происходит. Такой метод обеспечивает наличие в области A работающей программы после обновления, независимо от того, было обновление кода успешным или нет.

Для защиты от случайного стирания кода загрузчика во Flash-памяти в микроконтроллерах Atmel семейства SAM3 и SAM4 код загрузчика размещается в масочном ПЗУ. Для осуществления записи во Flash-память нужно сначала буферизировать данные для записи, а затем запустить процедуру работы с Flash-памятью, располагающейся в масочном ПЗУ микроконтроллера.

Для повышения надежности процесса управления обновлением приложений в микроконтроллерах, Flash-память которых организована в виде одного банка, разработчику приходится преодолевать некоторые трудности. Например, новое приложение может потребовать внесения изменений в таблицу векторов прерываний, то есть нужно изменять содержимое Flash-памяти по младшим адресам. Это означает, что невозможно сохранить содержимое этой части памяти неизменным в течение всего срока эксплуатации устройства. В этом случае для всех последующих обновлений программного кода необходимо зафиксировать векторы прерываний по тем адресам, где находятся обработчики

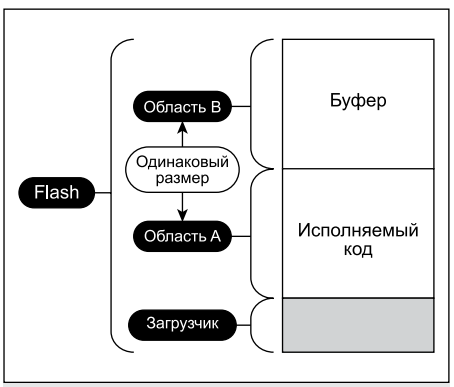

Рис. 1. Распределение памяти, организованной в виде одного банка

прерываний текущего приложения. Тогда коды обработчиков прерываний можно спокойно изменить при обновлении приложения, хотя это может привести к необходимости введения ограничений на объем кода обработчиков или к появлению во Flash-памяти неиспользуемых областей между обработчиками прерываний.

При использовании резервной копии во Flash-памяти с одним банком происходит увеличение времени процедуры обновления, а также отсутствует защита от ошибок, возникших при сбое питания.

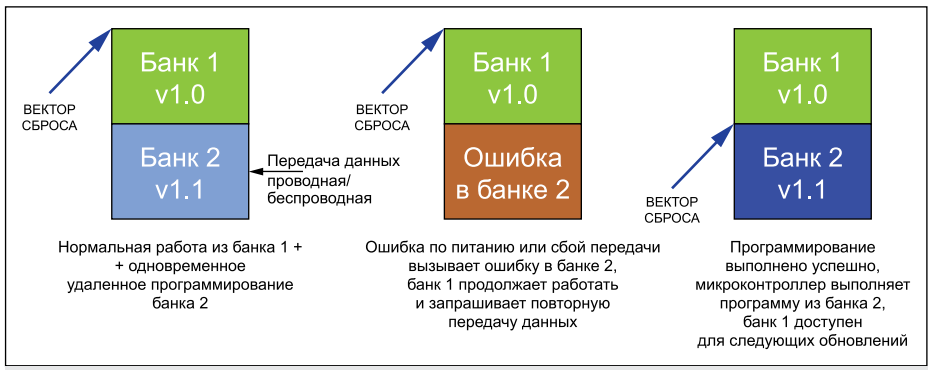

Рис. 2. Работа Flash-памяти с двумя банками

Организация Flash-памяти программ в виде двух банков

При организации Flash-памяти в виде одного банка во время обновления кода и сохранения данных во Flash-памяти процессор ожидает окончания операции записи и не осуществляет выборку команд из Flash-памяти, так как нельзя одновременно осуществлять две процедуры: чтение и запись. Происходит остановка в функционировании устройства, которая не всегда приемлема для конечного приложения.

Наличие двух банков позволяет программировать один банк, пока из другого банка осуществляется выборка исполняемого кода, что позволяет обеспечить непрерывное функционирование устройства во время стирания, записи и проверки корректности информации, записанной во второй банк.

Такое решение позволяет проводить процедуру обновления программного обеспечения, не останавливая работу устройства. Переключение микроконтроллера на выполнение новой версии программы можно осуществить после успешного выполнения процедуры обновления и проверки целостности полученного кода.

В новых микроконтроллерах Atmel, где Flash-память уже разделена на два банка, каждый из них может быть отображен на область загрузки. При распределении памяти используются преимущества такой организации: каждый банк содержит одинаковые копии загрузчика в начальных адресах, далее следует исполняемый код. При загрузке из одного банка другой банк используется в качестве буфера, где сохраняется новая прошивка. После получения новой версии программы и ее верификации для последующей загрузки можно использовать другой банк. Таким образом, в памяти находятся одновременно две рабочие программы. Для того чтобы иметь возможность исполнения микроконтроллером разных версий программного кода, можно менять отображение банков памяти на область загрузки.

Банк, который содержит требуемый исполняемый код, отображается на область загрузки (рис. 2). Алгоритм работы может выглядеть так:

- Сначала банк 1 является загрузочным, исполняется программа (v1).
- Банк 2 используется в качестве буфера для записи новой версии программы (v2).
- После загрузки новой версии программы (v2) в банк 2 и положительной ее верификации банк 2 отображается на область загрузки, и при следующем старте системы (или формировании сигнала сброса) выборка кода будет осуществляться из этого банка.
- После этого банк 1 используется в качестве буфера для получения новой программы либо области хранения прежней версии программы для обеспечения возможности возврата к предыдущему варианту программы.

Преимущество этого метода в том, что в памяти может содержаться две рабочие версии программы, и поэтому нет необходимости выполнять дополнительное копирование программы из одной области в другую. Просто меняется отображение банков на область загрузки. Но так как код загрузчика должен присутствовать в каждом банке, доступный для приложения объем памяти будет чуть меньше, чем размер самого банка. Таким образом, Flash-память, разделенная на два банка, обеспечивает безопасное для функционирования устройства удаленное обновление кода.

У микроконтроллеров Atmel Flash-память организована в виде двух банков в сериях SAM3SD, SAM3U4, SAM3X, SAM3A на базе ядра Cortex-M3 и SAM4SD на базе ядра Cortex-M4.

Flash-память с двумя банками позволяет надежно осуществлять обновление программы микроконтроллера благодаря возможности исполнения кода и перепрограммирования одновременно. Также наличие прежней версии кода дает возможность устройству в любой момент времени вернуться к ее исполнению.

Литература

1. http://atmelcorporation.wordpress.com/2013/06/21/the-value-of-microcontrollers-mcus-with-dual-bank-flash/
2. Atmel Application Note AT02333: Safe and Secure Bootloader Implementation for SAM3/4.