

## 5 Конфигурирование H-Flasher

Эта глава рассказывает о конфигурировании и использовании H-Flasher. В конце главы приведены два примера, используемые для справки.

### 5.1 Процесс работы H-Flasher

Процесс работы H-Flasher очень прост. Как показано на рис. 5-1, процесс работы состоит из четырёх пунктов: исполнение скриптов инициализации, загрузка флеш драйвера, проверка флеш ID и работа во флеш. Все эти пункты выполняются последовательно. Если один из пунктов не проходит, операция сразу же останавливается.

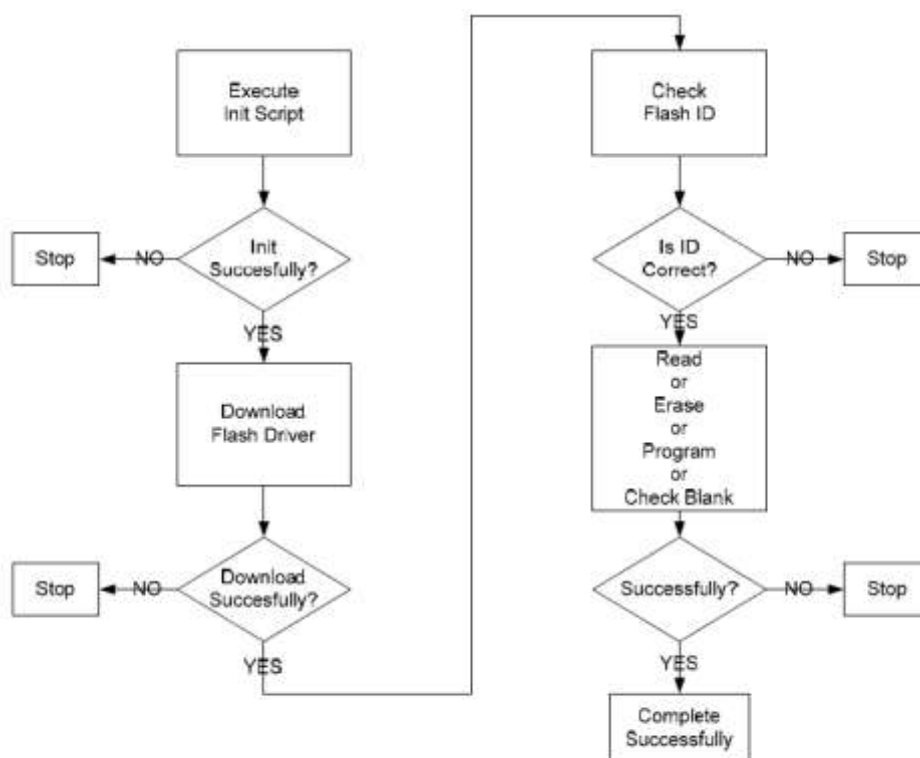


Fig 5-1 Workflow of H-Flasher

#### 5.1.1 Исполнение скрипта инициализации

Если задан скрипт инициализации, H-Flasher сначала выполняет его для инициализации подключенного устройства. Если не задан скрипт инициализации или скрипты инициализации не нужны, H-Flasher пропускает этот пункт. Если что-нибудь не проходит во время инициализации, H-Flasher сразу же останавливает процесс и извещает об этом пользователя.

#### 5.1.2 Загрузка флеш драйвера

После успешного завершения инициализации, H-Flasher ищет соответствующий флеш драйвер, подходящий для выбранной флеш и задаёт ширину в битах. Затем, H-Flasher загружает флеш драйвер в SRAM/SDRAM, основываясь на указанном стартовом адресе RAM. Если загрузка успешна, переходит к следующему пункту. Иначе, выход ошибки и предупреждение пользователя сообщением: Can't download driver to specified address.

---

**5.1.3 Проверка ID флеш**

После загрузки драйвера, H-Flasher проверяет ID флеш. Смысл в том, чтобы убедиться в правильности выбранной флеш и проверить, что флеш доступна по заданному адресу.

**5.1.4 Чтение/стирание/программирование/верификация/проверка на стирание**

После успешного завершения предыдущих трёх пунктов, H-Flasher может выполнить любую требуемую операцию над подключенным устройством. Поддерживаемые операции включая в себя: чтение, стирание, программирование, верификацию, проверку на стирание (пустоту флеш).

**5.2 Мастер программ H-Flasher**

H-Flasher поставляется с мастером программ для упрощения конфигурирования. Пользователь может следовать мастеру для выполнения конфигурации.

**5.2.1 Секция флеш**

В первом пункте пользователь может выбрать из списка подключаемую флеш. Пользователю нужно проверить информацию и проследить за корректностью выбора. Корректность выбора очень важна, поскольку разные флеш чипы задают разный набор команд.

**5.2.2 Конфигурация**

В этом пункте пользователю необходимо предоставить информацию о подключаемой системе, включая ширину флеш в битах, стартовый адрес флеш, стартовый адрес RAM, XTAL и TCK.

**Ширина в битах и количество чипов**

Некоторые внешние флеш чипы могут работать с различной шириной в битах, например, 8, 16 и 32 бита. Обычно драйверы для различной ширины также различаются. Отсюда, пользователю необходимо задать ширину в битах для этого типа чипов. Для чипов, поддерживающих только одну ширину в битах, используется только значение по умолчанию и пользователь не может его изменить. В некоторых проектах используются несколько чипов. В этом случае, пользователю необходимо задать правильное количество чипов.

**Стартовый адрес флеш**

Для работы с подключенной флеш, H-Flasher необходимо знать её стартовый адрес. Т.о., пользователю необходимо задать стартовый адрес флеш. Для H-Flasher правильное адресное пространство начинается с (стартового адреса флеш) до (стартового адреса флеш + размер – 1). Любой адрес, выходящий за этот диапазон считается неверным. Главное, флеш может быть доступна с адреса 0x0 без инициализации. Но некоторые системы поддерживают переадресацию и могут разместить флеш по другим адресам. В этом случае, пользователю нужно проследить, чтобы заданный стартовый адрес флеш согласовывался с предлагаемыми скриптами инициализации. Одним словом, предоставьте нужные скрипты инициализации и проследите за тем, чтобы H-Flasher имел доступ к флеш по заданному адресу.

---

**Стартовый адрес RAM**

Пользователю нужно разработать пространство в RAM, которое должно быть более 4 Кб, поскольку H-Flasher необходимо 4 Кб. Пространство RAM используется драйвером. Правильное адресное пространство с (стартового адреса RAM) по (стартовый адрес RAM + 4Кб – 1). H-Flasher загружает флеш драйвер в этом месте. Флеш драйвер может быть загружен в SRAM и SDRAM. Если подключенная система имеет on-chip SRAM, он вынуждает использовать on-chip SRAM вместо внешней SDRAM. Потому что доступ к SRAM много быстрее, чем к SDRAM. Предоставьте необходимые скрипты инициализации для инициализации системы памяти и проследите за тем, чтобы разработанное пространство памяти было доступно.

**XTAL**

Для некоторых чипов, H-Flasher необходимо знать частоту внешнего кварцевого генератора. H-Flasher использует эту частоту для генерирования системного тактового сигнала. Когда задан флеш чип, вход XTAL может быть запрещён или разрешён соответственно. Когда он разрешён, задайте XTAL, иначе проигнорируйте его.

**INIT TCK & PGM TCK**

INIT TCK задаёт скорость TCK, используемую на стадии инициализации, а PGM TCK задаёт скорость TCK, используемую на стадии программирования. Обычно, поддерживает более высокую скорость TCK после соответствующей инициализации. Поэтому пользователь может задать медленную скорость TCK для успешного выполнения инициализации и высокую скорость TCK для ускорения программирования. Внимание, INIT TCK и PGM TCK активны только в USB версии H-JTAG эмулятора.

**5.2.3 Скрипт инициализации**

В этом пункте пользователю необходимо предоставить скрипты инициализации для инициализации подключенной системы. Пользователь может редактировать скрипты в редакторе, поставляемым с H-Flasher. За дополнительной информацией обратитесь к главе 6.

Для on-chip флеш не требуется скриптов инициализации, поскольку драйвер уже содержит его. Для внешней флеш, скрипты инициализации необходимы. Назначение скриптов инициализации в конфигурировании системного тактового сигнала и системы памяти. Последнее более важное, поскольку корректная инициализация системы памяти – это необходимое условие. Иначе H-Flasher не будет иметь доступ к флеш и SDRAM.

Если H-Flasher не может загрузить драйвер в RAM, он предупреждает об этом сообщением об ошибке: Can't download driver to specified address. В большинстве случаев, это происходит из-за отсутствия скрипта инициализации или из-за некорректности скрипта инициализации. Чтобы предоставить правильные скрипты, пользователь должен знать подключенную систему. Мы рекомендуем обратиться к технической документации, особенно к разделу о конфигурации памяти.

**Совет:** Когда используется USB H-JTAG эмулятор, пользователь может конфигурировать системный тактовый сигнал через скрипты инициализации, чтобы добиться лучших характеристик.

**5.2.4 Программирование**

В этом пункте, пользователь выполняет операции во флеш. Поддерживаемые операции включают: проверку флеш и информации об устройстве, программирование флеш, верификация флеш, стирание флеш, проверка на стирание (пустоту флеш).

---

- **Reset**

Операция сброса используется для выполнения сброса подключенной системы.

- **Check**

Операция проверки читает ID флеш и другую основную информацию. Пользователь может использовать эту информацию для тестирования конфигурации на корректность.

- **UnProtect**

Эта операция используется для снятия защиты с защищённых флеш чипов. Когда выбранная флеш не поддерживает эту операцию, кнопка автоматически запрещается. В настоящий момент поддерживается серия STM32F.

- **Program**

H-Flasher предлагает три типа программирования для различных сценариев: автоматическую загрузку во флеш, формат Intel HEX и обыкновенный бинарный формат. Для программирования во флеш, необходимы данные для записи и адрес назначения. Различия в этих трёх типах в источнике информации.

- A - Auto Flash Download**

Для автоматической загрузки во флеш не требуется исходный файл и адрес назначения. Вся информация исходит от H-Jtag.

- B - Intel HEX Format**

Файл HEX содержит информацию о данных/программе и адресе. Однако пользователю необходимо лишь указать исходный файл формата Intel HEX. H-Flasher автоматически извлекает адрес из HEX файла и использует его в качестве адреса назначения.

- C - Plain Binary Format**

Обыкновенный бинарный файл содержит только данные/программу. Для обыкновенного бинарного файла пользователь должен задать исходный файл и адрес назначения.

- **Verify**

Эта операция используется для верификации программирования путём чтения содержимого подключенной флеш и сравнения его с исходным файлом.

- **Erase & Check Blank**

Операции стирания и проверки на стёртость используются для стирания флеш и проверки её опустошения. Для обеих операций пользователь должен задать диапазон, использования окна списка.

- **Read**

Операция чтения предлагает дамп содержимого памяти по заданному адресу. Для чтения содержимого памяти, задайте стартовый адрес и размер в байтах (для Nor и On-Chip флеш) или стартовый сектор, стартовую страницу, конечный сектор и конечную страницу для Nand флеш.

---

**5.2.5 Программные опции**

H-Flasher предлагает некоторые полезные опции, например, сброс устройства после программирования, дополнительную верификацию и шифрование. Пользователь может выбрать эти опции по потребностям.

**RESET**

Когда разрешена, H-Flasher сбросит устройства после программирования.

**VERIFICATION**

Когда разрешена, H-Flasher прочитает данные из флеш и сравнит их с исходным файлом через секунду, после завершения программирования.

**SKIP ID CHECK**

Когда разрешена, H-Flasher пропустит операцию проверки ID флеш.

**ERASE CHIP**

Когда разрешена, H-Flasher сотрёт весь флеш чип, содержащий выбранный сектор, перед программированием флеш.

**SMART MODE**

Когда разрешена, H-Flasher делает резервное копирование содержимого флеш перед программированием и восстанавливает его во время программирования. Эта опция предохраняет от изменений данные, которые не покрываются исходным файлом.

**NXP LPC1700/2000**

Эта опция определяет, когда H-Flasher настраивает контрольную сумму таблицы векторов перед программированием.

**ATMEL AT91SAM**

Когда разрешена, H-Flasher установит защитный бит для разрешения защиты флеш по окончании программирования.

**ST STM32F**

Когда разрешена, H-Flasher установит байт опции RDP для разрешения защиты флеш по окончании программирования.

**NAND FLASH PGM OPTIONS**

Для программирования NAND флеш, пользователю необходимо предоставить следующую информацию для H-Flasher:

- **Programming Mode:**

Задаёт режим программирования NAND флеш, пропуск плохих блоков или перемещение плохих блоков.

- **Scan of Bad Blocks:**

Задаёт способ сканирования плохих блоков в режиме перемещения, стирает весь чип или стирает задействованные блоки, резервирует область под перемещение и резервирует блоки под перемещение.

- **Reserved Area for User Table:**

Задаёт резервируемую область под плохие блоки, таблицу перемещения или пользовательские данные.

---

H-Flasher поддерживает два режима программирования NAND: режим пропуска и режим замещения. В режиме пропуска, H-Flasher будет пропускать плохие блоки и перескакивать на следующие хорошие. В режиме перемещения, H-Flasher будет замещать плохие блоки хорошими в зарезервированной области. Когда выбран режим пропуска, H-Flasher просто пропускает плохой блок и программирует данные непрерывно в следующий хороший блок. Когда выбран режим замещения, H-Flasher сначала сканирует флеш, чтобы собрать информацию о плохих блоках путём стирания всего чипа или путём стирания задействованных блоков, резервирует область под замещение и резервирует область под пользовательские таблицы. Во время программирования, если встречен плохой блок, H-Flasher будет программировать данные в первый хороший блок из резервированной области под замещение. Одновременно, H-Flasher будет записывать информацию о замещении. Когда программирование завершится, H-Flasher передаст информацию о таблице плохих блоков и таблице замещения драйверу NAND флеш. В драйвере флеш, функция `nand_info_table()` примет всю информацию, посланную H-Flasher. Пользователь может модифицировать эту функцию, чтобы создать таблицу плохих блоков и таблицу замещения в зарезервированной области, как ему нужно.

**Примечание 1:** В режиме пропуска, H-Flasher не посылает информацию о таблице плохих блоков и таблице замещения драйверу NAND флеш по завершении программирования.

**Примечание 2:** В зависимости от характеристик NAND флеш, становится невозможным обеспечить общим драйвером флеш все аппаратные платформы. Когда вам необходима таблица плохих блоков или таблица замещения, вы можете модифицировать функцию `nand_info_table()`, опираясь на собственные потребности.

**Примечание 3:** Некоторые опции работают только в обычном режиме. Для автоматической загрузки во флеш, H-Flasher игнорирует все эти опции.

## **PRODUCTION MODE**

Когда эта опция разрешена, H-Flasher вводит код продукции. В режиме продукция, программа работает в упрощённом виде, чтобы увеличить эффективность. После ввода кода продукции, H-Flasher продолжает обнаруживать подключенное устройство. Если устройство обнаружено, H-Flasher автоматически запускает процесс программирования. Когда программирование завершено, пользователь извещается об отключении устройства и подключает следующее. Т.о. пользователю нужно лишь подключать устройство и отключать его по завершении программирования, затем подключать следующее.

---

## 5.3 Полезные советы

**Совет 1:** H-Flasher и H-Flasher Lite одинаковы, за исключением того, что H-Flasher Lite не поддерживает автоматическую загрузку во флеш. Чтобы использовать автоматическую загрузку во флеш, запустите и сконфигурируйте H-Flasher взамен H-Flasher Lite.

**Совет 2:** Во время конфигурации, если редактируемый блок или список блоков серые, это означает, что им доступна только одна опция. Пользователю ничего не нужно делать.

**Совет 3:** В H-Flasher введенная конфигурация может быть сохранена в HFC файле для дальнейшего использования.

**Совет 4:** Перед началом программирования, H-Flasher автоматически сотрёт всю часть флеш. Пользователю не требуется стирать флеш вручную.

**Совет 5:** Операция стирания применяется к сектору. Чтобы избежать потери данных, H-Flasher предлагает механизм резервного копирования и восстановления. H-Flasher копирует все данные из флеш перед стиранием и восстанавливает их при программировании. С помощью этого механизма, содержимое флеш памяти остаётся неизменным, за исключением перекрываемой текущим программированием.

**Совет 6:** Когда пользователь видит сообщение об ошибке: "Destination flash address is out of range", он должен задать адрес назначения, или адрес назначения, полученный из HEX файла, не попадает в правильный диапазон адресов. Пожалуйста, проверьте адрес назначения и убедитесь в его корректности.

**Совет 7:** В некоторых чипах ID может измениться после обновления версии. В этом случае свяжитесь с нами. Мы предоставим обновлённый флеш драйвер.

**Совет 8:** Если что-то происходит во время операции, пожалуйста, проверьте конфигурацию и проследите, чтобы предоставляемая конфигурация была корректной. Если ошибки повторяются при корректной конфигурации, свяжитесь с нами. Мы проанализируем эти проблемы и предоставим вам обновлённый драйвер, если нужно.