

## 5.4 Пример 1- AT91SAM7X256

Чип AT91SAM7X256 основан на ядре ARM7 фирмы ATMEL. Этот чип поставляется с 256 Кб внутренней флеш и 64 Кб внутренней SRAM. В этой секции показано, как конфигурировать и программировать его в H-Flasher.

### 5.4.1 Секция флеш

В этом пункте, выберите AT91SAM7X256 в качестве подключаемого чипа, как показано на рис. 5-2.

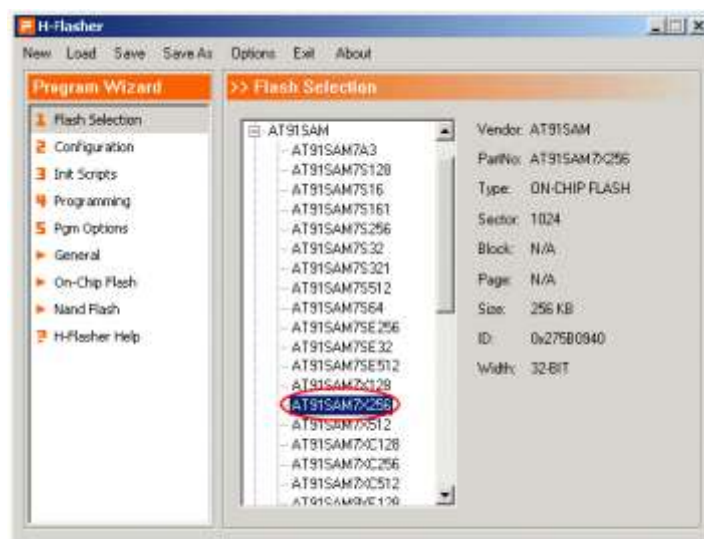


Fig 5-2 Flash Selection: AT91SAM7X256

### 5.4.2 Конфигурация

В этом пункте, пользователю необходимо предоставить основную информацию о памяти. Для AT91SAM7X256 – это ширина в битах, стартовый адрес во флеш и стартовый адрес в RAM фиксированы. Использована конфигурация по умолчанию. Для этого примера не нужно задавать XTAL.

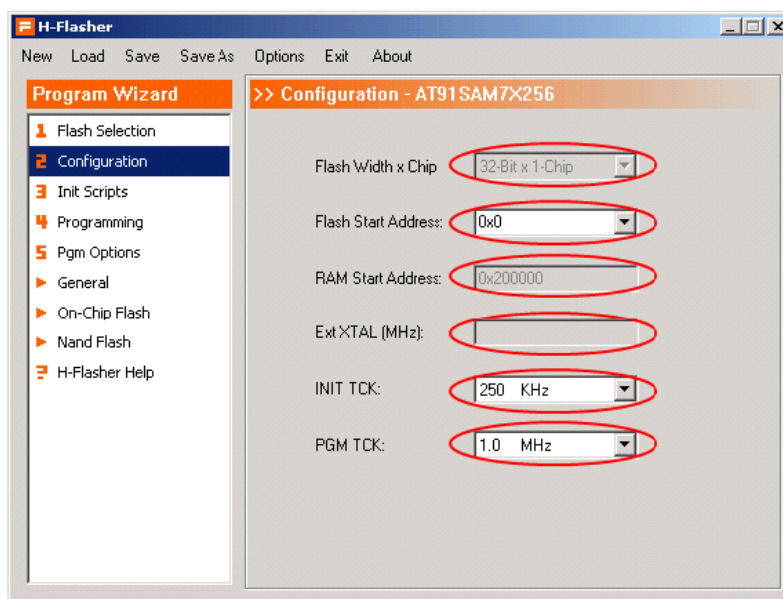


Fig 5-3 Configuration: AT91SAM7X256

### 5.4.3 Скрипт инициализации

В этом пункте, пользователю необходимо предоставить скрипт инициализации. Для AT91SAM7X256 драйвер уже содержит инициализацию. Поэтому пользователю не нужно предоставлять скрипт инициализации для AT91SAM7X256. Для этого примера, кнопки редактирования запрещены, как показано на рис. 5-4.

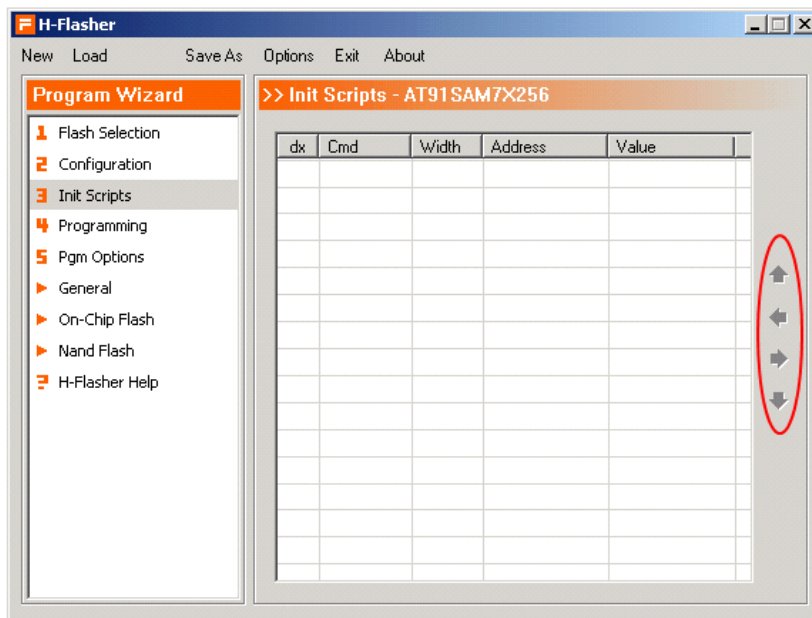


Fig 5-4 Init Script: AT91SAM7X256

### 5.4.4 Опции программирования

Перед началом операции, пользователь может выбрать различные программные опции. Для этого примера, мы выбрали выполнение дополнительной верификации и сброса подключенного устройства по завершению программирования.

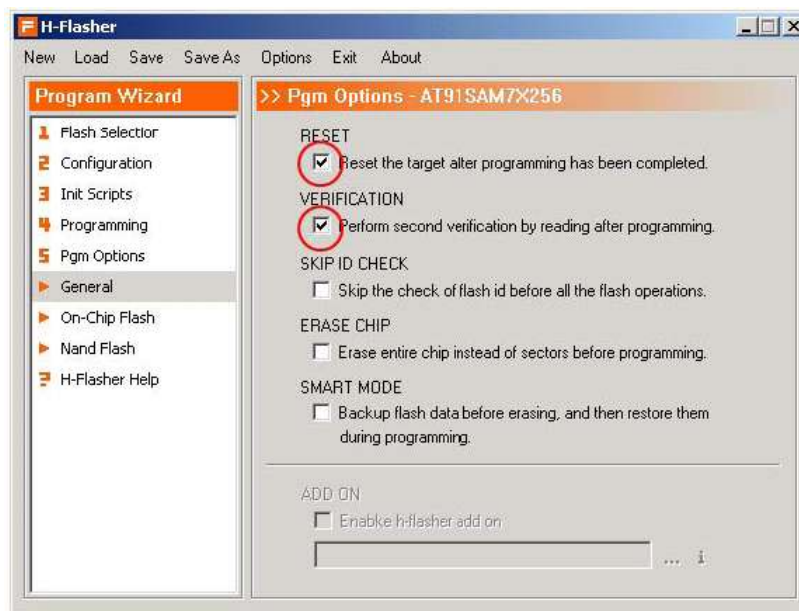


Fig 5-5 Program Options

### 5.4.5 Программирование

После завершения конфигурации в п. 4, пользователь может проводить операции с флеш. Сначала, попытайтесь получить информацию от подключенного устройства, кликнув "Check". Для этого примера, результат проверки показан на рис. 5-6. Результат демонстрирует хорошо сделанную конфигурацию.

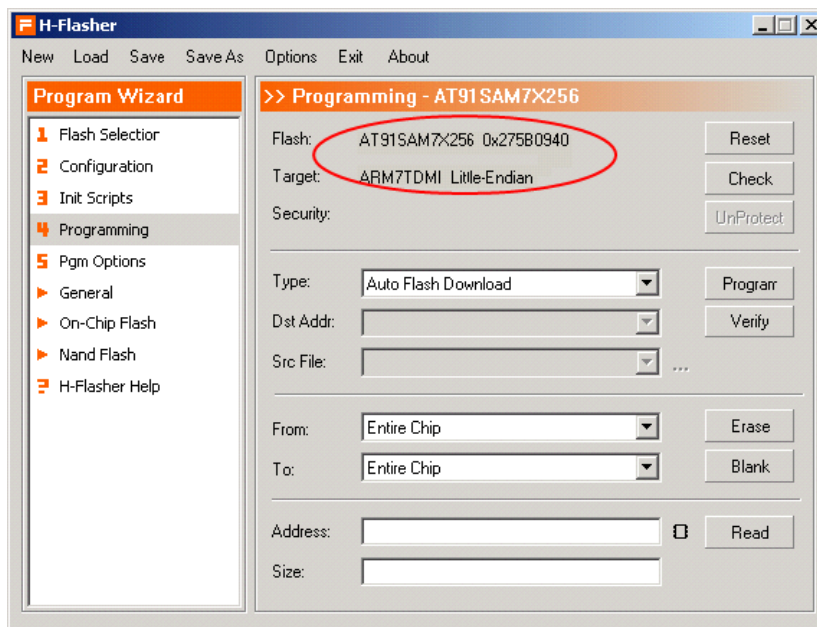


Fig 5-6 Check Result: AT91SAM7X256

Далее попытайтесь запрограммировать бинарный файл. Для этого примера – это настройки, показанные на рис. 5-7. Выбран простой бинарный формат, исходный файл TEST.bin расположен в C:\, а адрес назначения – 0x0 (базовый адрес флеш).

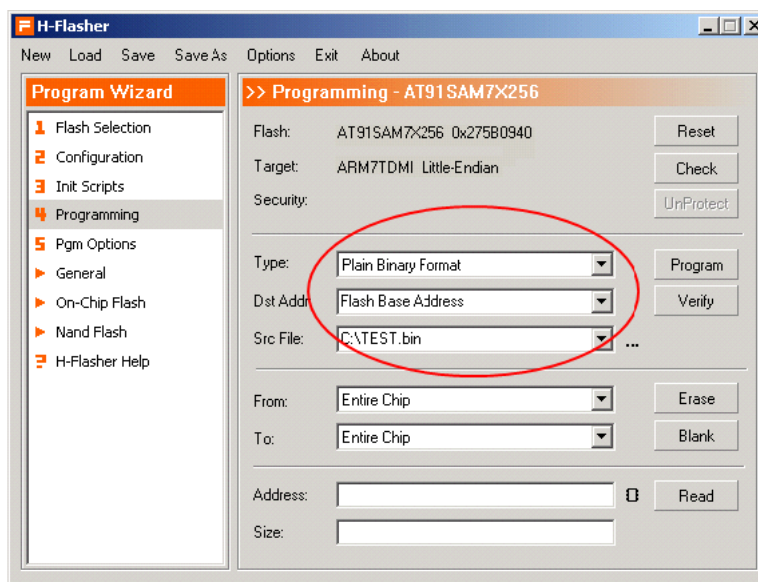


Fig 5-7 Programming Settings

Затем кликните "Program" для запуска программирования. Вы можете видеть прогресс процесса, среднюю скорость и затраченное время. Когда программирование успешно завершится, H-Flasher известит об этом пользователя сообщением, показанным на рис. 5-8.

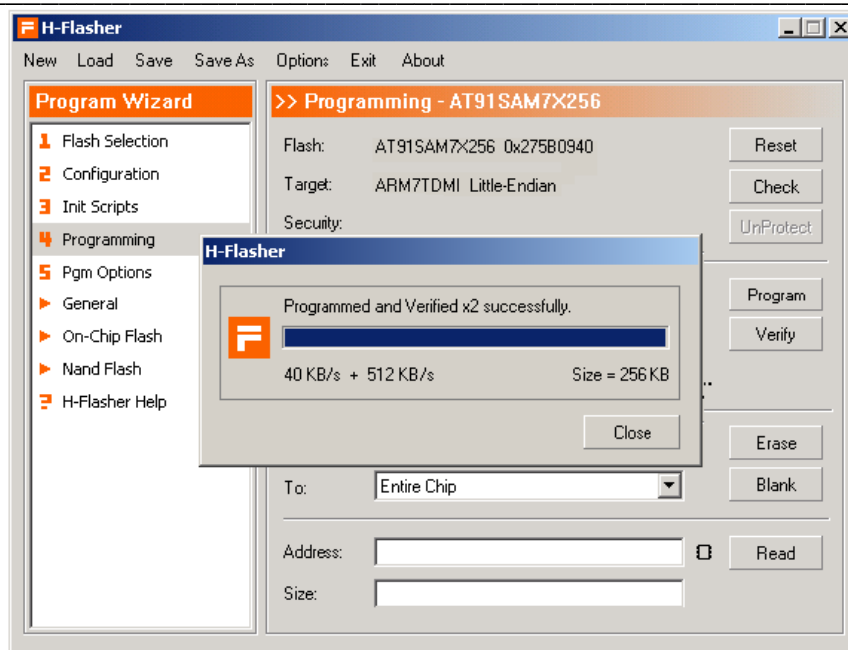


Fig 5-8 Programming Is Completed

#### 5.4.6 Сохранение конфигурации

Пользователь может сохранить эту конфигурацию в HFC файле. Позже, он сможет загрузить файл HFC прямо в H-Flasher, чтобы не конфигурировать заново.

## 5.5 Пример 2 - LPC2210 + SST39VF1601

Чип LPC2210 основан на ядре ARM7 от NXP (бывший PHILIPS Semiconduct). Этот чип имеет 16 Кб внутренней SRAM и не имеет внутренней флеш. Он также имеет четыре внешних банка памяти, которые могут быть использованы под расширение внешней флеш и SDRAM. В этом примере, мы допускаем, что BANK0 используется под внешнюю флеш (SST39VF1601). Диапазон адресов от 0x80000000 до 0x80FFFFFF. BANK1 используется под внешнюю SDRAM с диапазоном адресов от 0x81000000 до 0x81FFFFFF. Далее мы должны подготовить к программированию H-Flasher + SST39VF1601 с H-Flasher.

### 5.5.1 Секция флеш

В первом пункте выбирается флеш подключаемого устройства. Для этого примера выбирается SST39VF1601 как показано на рис. 5-9.

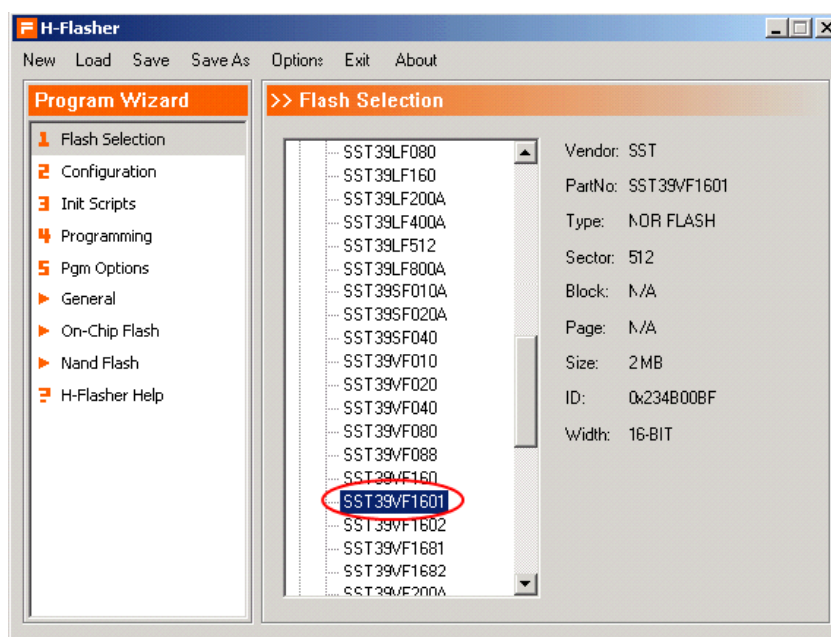


Fig 5-9 Flash Selection: SST39VF1601

### 5.5.2 Конфигурация

В этом пункте мы предоставляем информацию о системе памяти. SST39VF1601 поддерживает только режим 16 бит. Поэтому мы можем использовать ширину в битах по умолчанию. SST39VF1601 расширяется через BANK0, поэтому стартовый адрес флеш 0x80000000. Несмотря на то, что LPC2210 имеет внутреннюю SRAM, мы используем в качестве примера внешнюю SRAM. Внешняя SRAM расширяется через BANK1. Поэтому стартовый адрес RAM - 0x81000000. К тому же, мы должны задать некоторую область 4 Кб внутри SDRAM. XTAL не нужно задавать. Окончательные настройки для этого примера показаны на рис. 5-10.

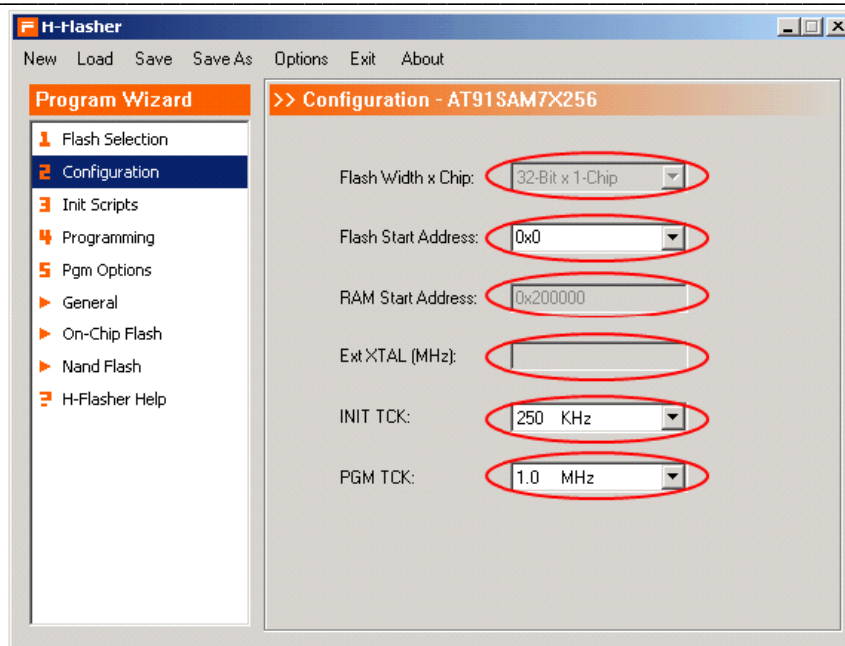


Fig 5-10 Configuration: LPC2210 + SST39VF1601

### 5.5.3 Скрипт инициализации

В соответствии с технической документацией на LPC2210, вам необходимо сконфигурировать три регистра. PINSEL2@0xE002C014, BCFG0@0xFFE00000 и BCFG1@0xFFE00004. PINSEL2 – это вывод выбора регистров, который используется для конфигурирования каких-либо выводов мультиплексирования. BCFGx – регистр конфигурации банка, который используется для задания ширины шины и циклов ожидания чтения/записи. За подробной информацией обратитесь к технической документации. Для этого примера нам потребуются три скрипта, показанные на рис. 5-11, которые сделают доступной SDRAM.

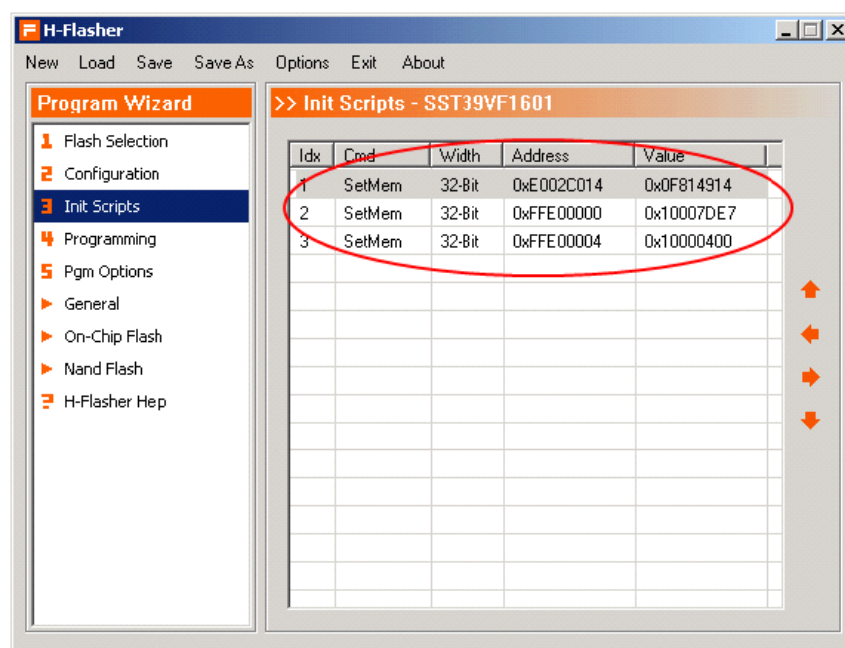


Fig 5-11 Init Script: LPC2210 + SST39VF1601



#### 5.5.4 Опции программирования

Перед началом операции, пользователь может выбрать различные программные опции. Для этого примера, мы выбрали выполнение дополнительной верификации и сброса подключенного устройства по завершению программирования

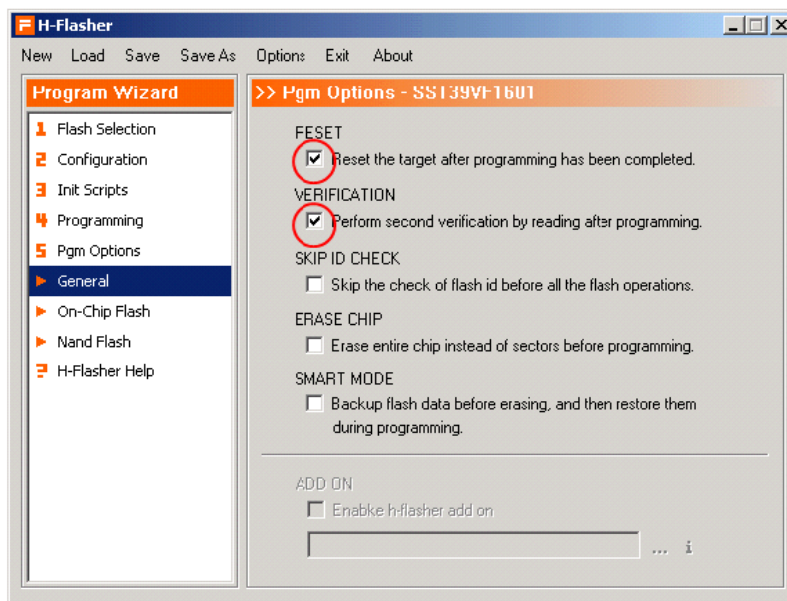


Fig 5-12 Program Options

#### 5.5.5 Программирование

После завершения конфигурации в п. 4, пользователь может проводить операции с флеш. Сначала, попытайтесь получить информацию от подключенного устройства, кликнув "Check". Для этого примера, результат проверки показан на рис. 5-13. Результат демонстрирует хорошо сделанную конфигурацию.

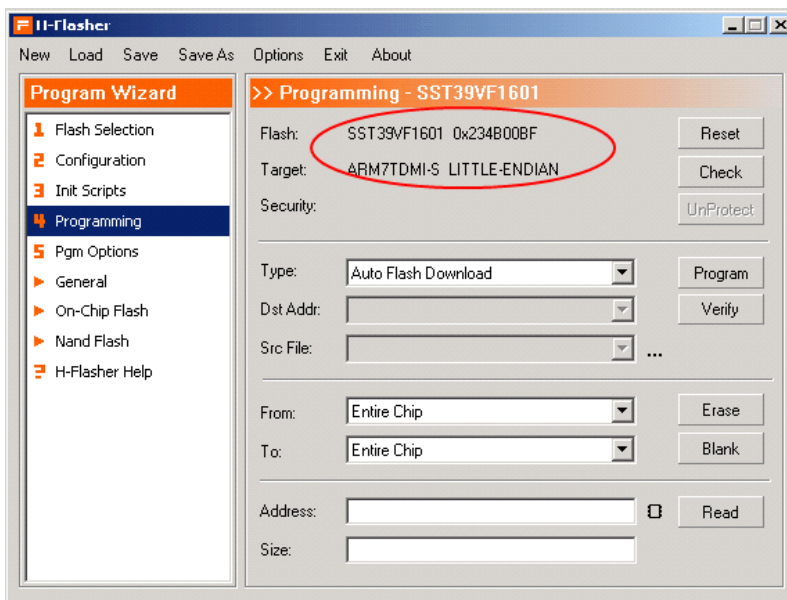


Fig 5-13 Check Result: LPC2210 + SST39VF1601

Далее попытайтесь запрограммировать бинарный файл. Для этого примера – это настройки, показанные на рис. 5-14. Выбран простой бинарный формат, исходный файл TEST.bin расположен в C:\, а адрес назначения – 0x80000000.

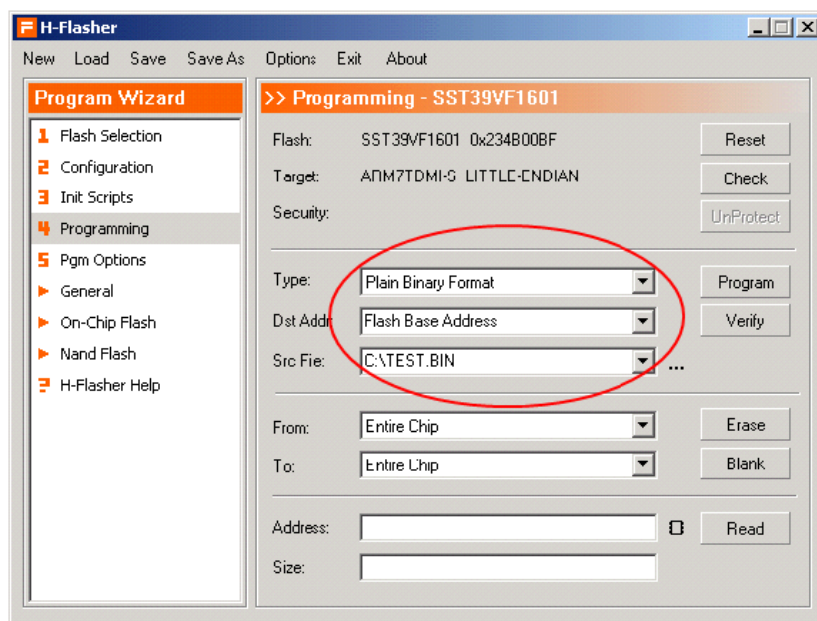


Fig 5-14 Programming Settings

Затем кликните “Program” для запуска программирования. Вы можете видеть прогресс процесса, среднюю скорость и затраченное время. Когда программирование успешно завершится, H-Flasher известит об этом пользователя сообщением, показанным на рис. 5-15.

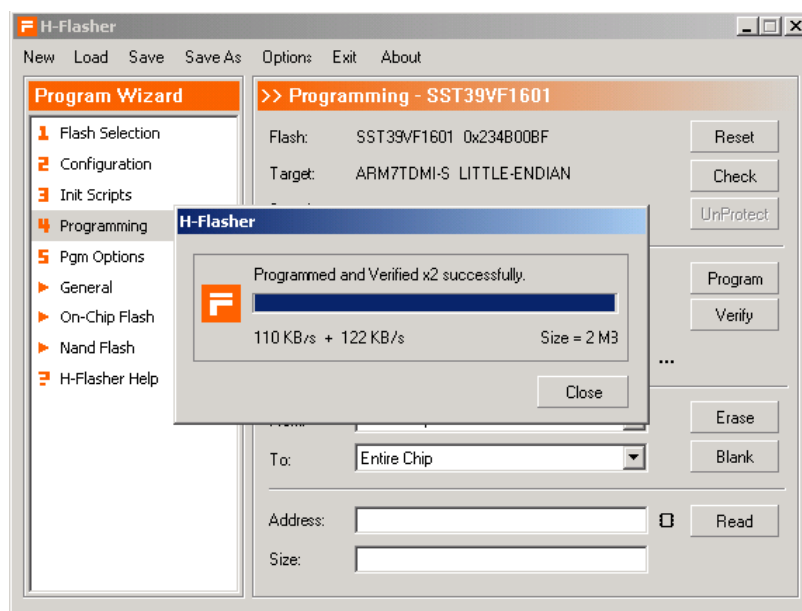


Fig 5-15 Programming Is Completed

### 5.5.6 Сохранение конфигурации

Пользователь может сохранить эту конфигурацию в HFC файле. Позже, он сможет загрузить файл HFC прямо в H-Flasher, чтобы не конфигурировать заново.