AN3965
Application note

STM32F40x/STM32F41x in-application programming using the USART

1 Introduction

An important requirement for most Flash-memory-based systems is the ability to update firmware when installed in the end product. This ability is referred to as in-application programming (IAP). The purpose of this application note is to provide general guidelines for creating an IAP application.

The STM32F4xx microcontroller can run user-specific firmware to perform IAP of the microcontroller-embedded Flash memory. This feature allows the use of any type of communication protocol for the reprogramming process. The USART is the example used in this application note.

Contents

2 IAP overview

STM32F4xx devices are implemented in the STMicroelectronics STM324xG-EVAL evaluation board.

2.1 Principle

You should program the IAP driver to the Flash memory base address via the JTAG/SWD interface using the development toolchain of your choice or the factory-embedded boot loader in the System memory area.

The IAP driver uses the USART to:

- Download a binary file from the HyperTerminal to the STM32F4xx internal Flash memory.
- Upload the STM32F4xx internal Flash memory content (starting from the defined user application address) into a binary file.
- Execute the user program.

2.2 IAP driver description

The IAP driver contains the following set of source files:

- *main.c*: Where the USART initialization and RCC configuration are set. A main menu is then executed from the *menu.c* file.
- *menu.c*: Contains the main menu routine. The main menu gives the options of downloading a new binary file, uploading internal Flash memory, executing the binary file already loaded and disabling the write protection of the pages where the user loads his binary file (if they are write-protected).
- *flash_if.c*: Contains write, erase and disable write protection of the internal Flash memory functions.
- *common.c*: Contains functions related to read/write from/to USART peripheral
- *ymodem.c*: It is used to receive/send the data from/to the HyperTerminal application using the YMODEM protocol[a]. In case of a failure when receiving the data, the “Failed to receive the file” error message is displayed. If the data is successfully received, it is programmed into the internal Flash memory from the appropriate address. A comparison between internal RAM contents and internal Flash memory contents is performed to check the data integrity. If there is any data discrepancy, the “Verification failed” error message is displayed. Other error messages are also displayed when the image file size is greater than the allowed memory space and when the user aborts the task.
- STM32F4xx Standard Peripherals Library.

a. The Ymodem protocol sends data in 1024-byte blocks. An error check is performed in the data blocks transmitted to the STM32F4xx internal RAM to compare the transmitted and received data. Blocks unsuccessfully received are acknowledged with an NAK (Negative AcKnowledgement). For more details about the Ymodem protocol, refer to the existing documentation.

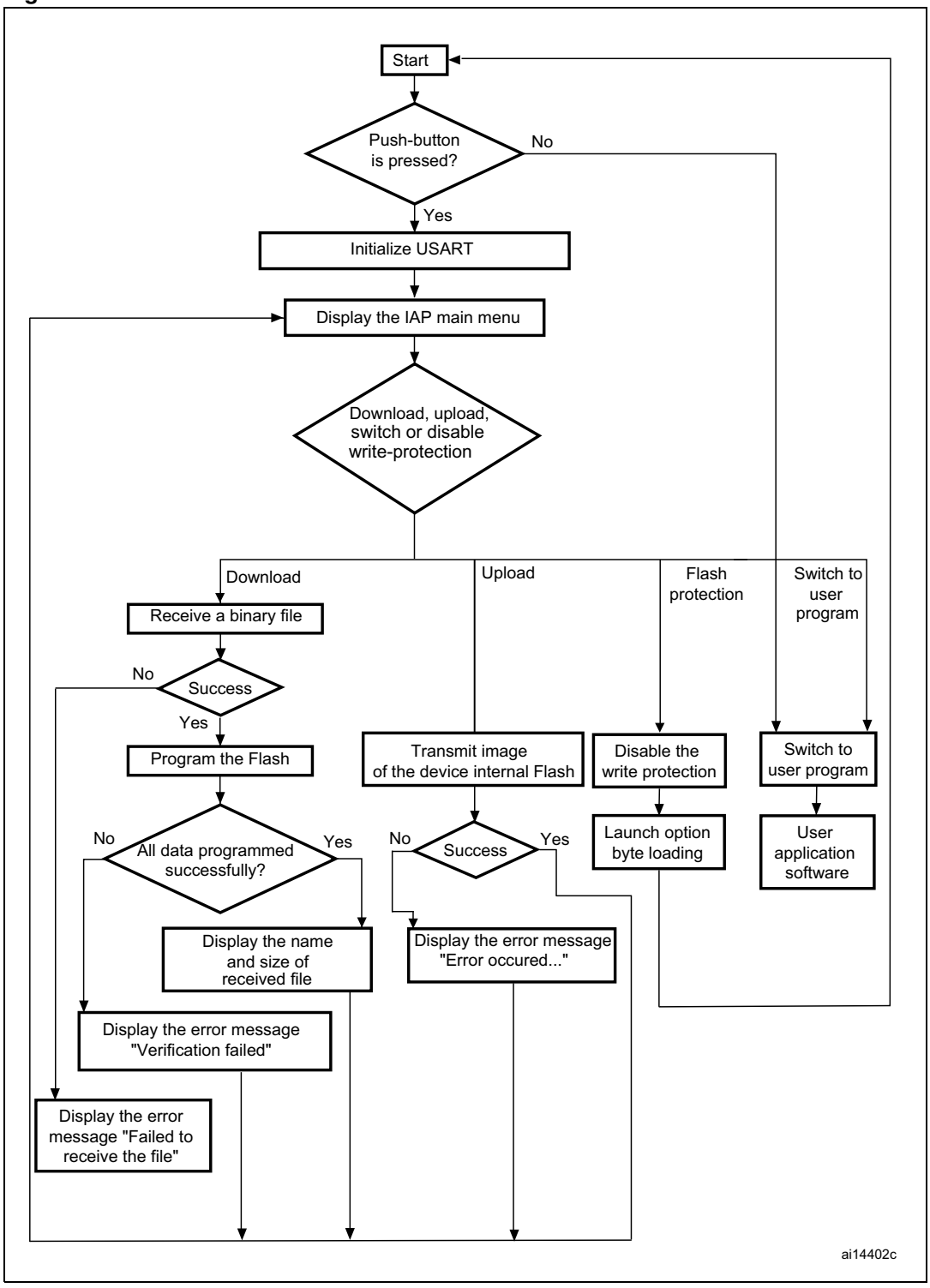
The user can choose to either go to the user application or execute the IAP for reprogramming purposes by pressing a push-button connected to a pin:

- Not pressing the push-button at reset: switches to the user application.
- Pressing the push-button at reset: displays the IAP main menu.

Refer to *Table 1: STM32F4xx IAP implementation* for more details about the STM324xG-EVAL board push-button used to enter the IAP mode.

The IAP flowchart is shown in *Figure 1*.

Figure 1. Flowchart of the IAP driver

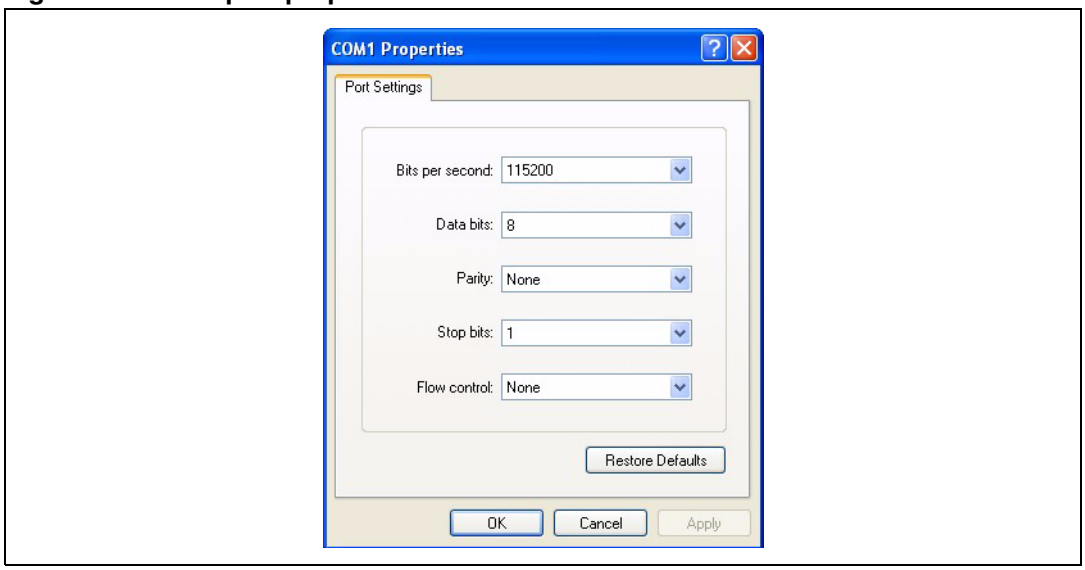


3 Running the IAP driver

3.1 HyperTerminal configuration

To use the IAP, the user must have a PC running HyperTerminal or other Terminal program that supports **ymodem protocol**. In this document the HyperTerminal is used. *Figure 2* shows the HyperTerminal configuration.

Figure 2. COM port properties



Note: The baud rate value of 115200 bps is used as an example.

Care must be taken when selecting the system clock frequency. To guarantee successful communication via the USART, the system clock frequency in the end application must be such that a baud rate equal to 115200 bps can be generated.

3.2 Executing the IAP driver

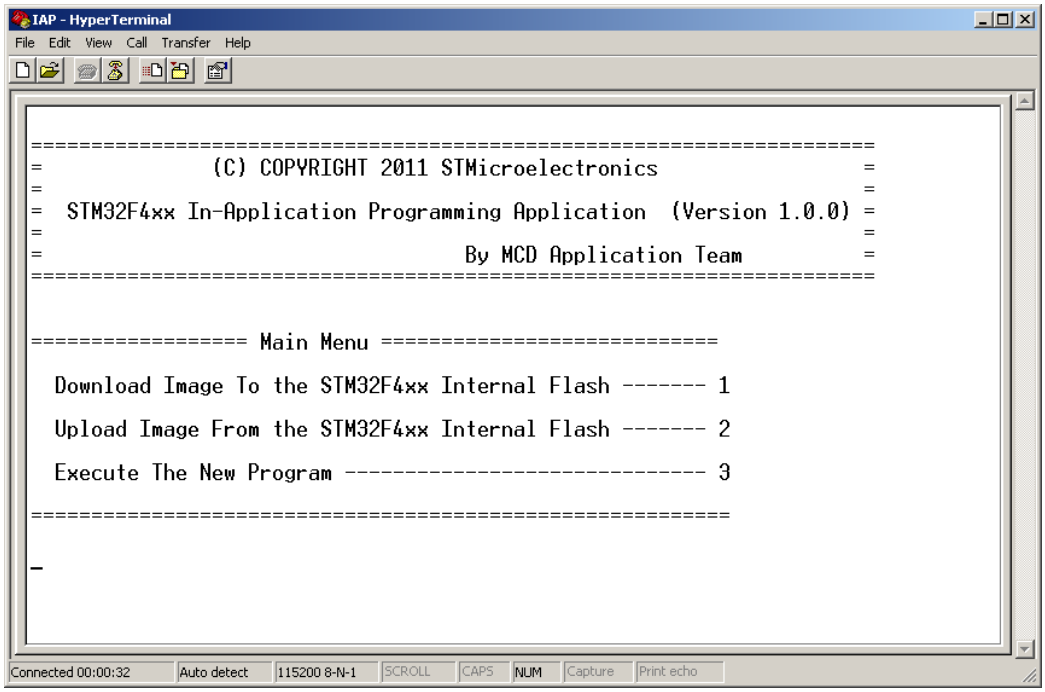
As an example, in this application note, pressing the push-button connected to the pin launches the IAP driver.

By pressing the push-button at reset, the user can run the IAP driver to reprogram the STM32F4xx internal Flash memory. It is not mandatory to use the push-button; the user can apply a signal to this pin with respect to its active level. Refer to *Table 1: STM32F4xx IAP implementation*.

4 IAP driver menu

Running the IAP displays the following menu in the HyperTerminal window.

Figure 3. IAP driver menu when the STM32F4xx Flash memory is not protected



4.1 Download image to the internal Flash memory

To download a binary file via HyperTerminal to the STM32F4xx's internal Flash memory, do as follows:

1. Press **1** on the keyboard to select the **Download Image To the STM32F4xx Internal Flash** menu.
2. Select **Send File** in the **Transfer** menu.
3. In the **Filename** field, type the name and the path of the binary file you want to download.
4. From the protocol list, select the **Ymodem** protocol.
5. Click on the **Send** button.

As a result, the IAP driver loads the binary file into the STM32F4xx's internal Flash memory from the defined base address and displays the binary file name and size in the HyperTerminal window. for further information on base address settings, refer to *Chapter 6: User program conditions*.

4.2 Upload image from the internal Flash memory

To upload a copy of the internal Flash memory started from the user application address, do as follows:

1. Press 2 on the keyboard to select **Upload image from the STM32F4xx internal Flash** menu.
2. Select **Receive File** in the **Transfer** menu.
3. Select the directory to save the binary file.
4. From the protocol list, select the **Ymodem** protocol.
5. Click on the **Receive** button.

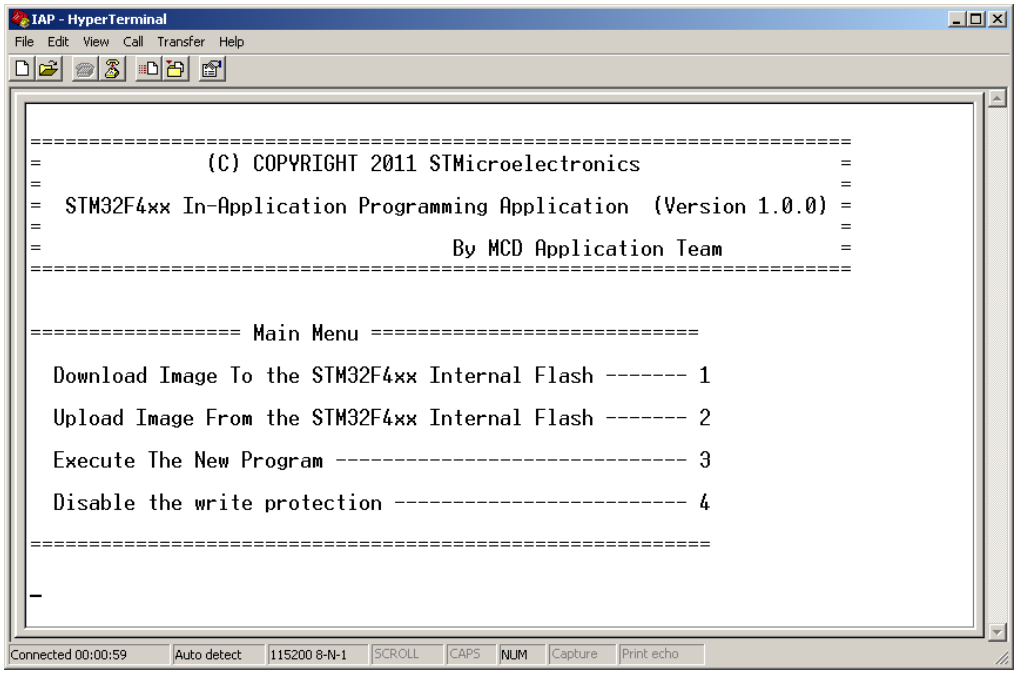
4.3 Execute the new program

Once the new program has been loaded, press **3** on the keyboard to select the **Execute The New Program** menu and execute the code.

4.4 Disabling the write protection

When the IAP starts, it checks the Flash memory pages where the user program is to be loaded to see if any are write-protected. If this instance, the menu shown in *Figure 4* appears.

Figure 4. IAP driver menu when the STM32F4xx Flash memory is write-protected



Prior to downloading the new program, the write protection must be disabled. To do so, press 4 (**Disable the write protection**) on the keyboard. The write protection is disabled and a system reset is generated to reload the new option byte values. After resuming from reset, the menu shown in *Figure 3* is displayed if the key push-button is pressed.

Note: In this application, the read protection is not supported, so the user has to verify that the Flash memory is not read-protected.

5 STM32F4xx IAP implementation summary

Table 1 provides a summary of the STM32F4xx IAP implementation.

Table 1. STM32F4xx IAP implementation

Platform	Implementation	Configuration
Firmware	The IAP program is located at 0x8000000. The Flash routines (program/erase) are executed from the Flash memory. The size of this program is about 8 Kbytes and programmed on:	Sector 0
	The user application (image to be downloaded with the IAP) will be programmed starting from address 0x8004000(1) The maximum size of the image to be loaded is:	1008 Kbytes (Sector 1 - Sector 11)
	The image is uploaded with the IAP from the STM32F4xx internal Flash. The size of the image to be uploaded is:	1008 Kbytes (Sector 1 - Sector 11)
Hardware	Push-button (active level: high)	Key push-button connected to pin PG15
	USART used	USART3

1. User application location address is defined in the *flash_if.h* file as: `#define APPLICATION_ADDRESS 0x8004000`. To modify it, change the default value to the desired one.

The STM32F4xx IAP package comes with:

- Source files and pre-configured projects for the IAP program (under Project\STM32F4xx_IAP directory)
- Source files and pre-configured projects that build the application to be loaded into Flash memory using the IAP (under Project\STM32F4xx_IAP\binary_template directory).

The `readme.txt` files provided within this package describes step by step how to execute this IAP application.

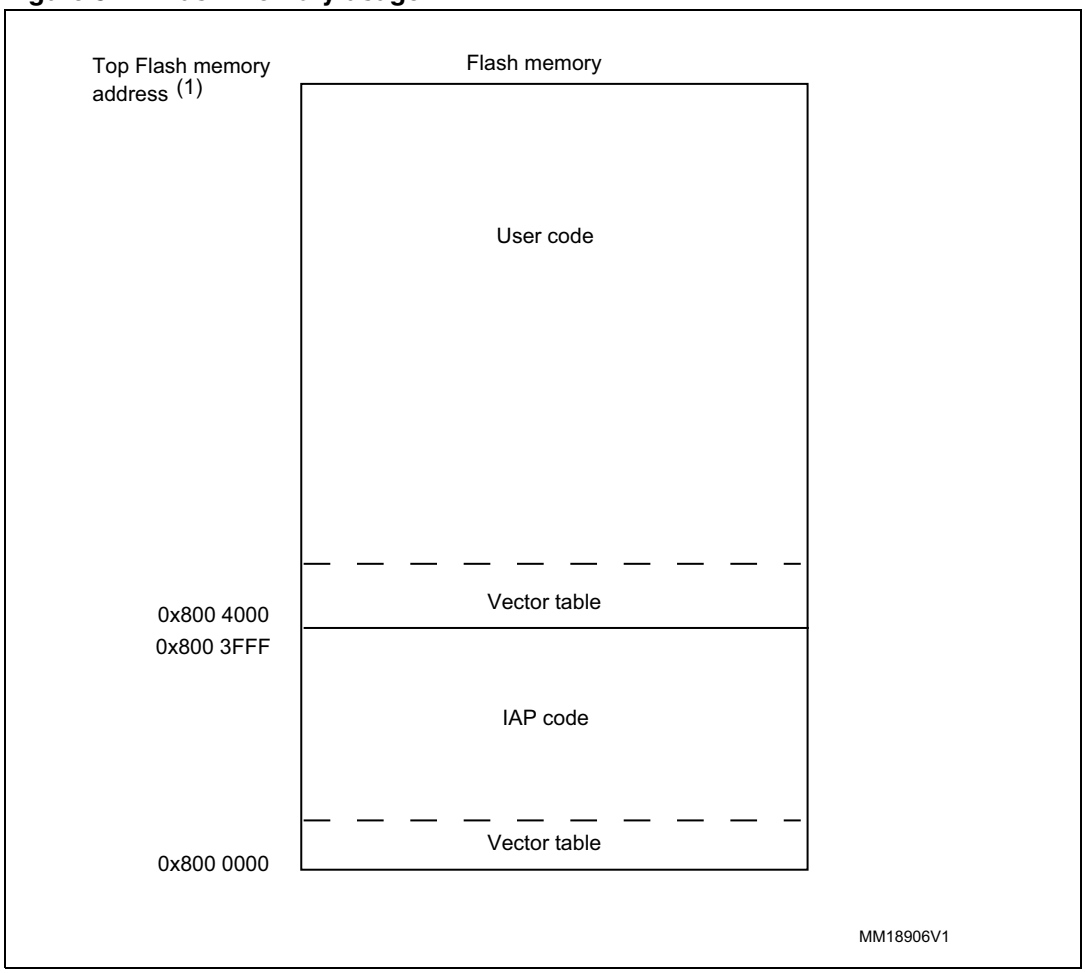
6 User program conditions

The user application to be loaded into the Flash memory using IAP should be built with these configuration settings:

1. Set the program load address at 0x08004000, using your toolchain linker file
2. Relocate the vector table at address 0x08004000, using the “NVIC_SetVectorTable” function from the `misc.h/.c` driver (under STM32F4xx_StdPeriph_Driver\inc) or by modifying the value of the constant “VECT_TAB_OFFSET” defined in the `system_stm32f4xx.c` file.”

An example application program to be loaded with the IAP application is provided with preconfigured projects.

Figure 5. Flash memory usage



1. Top Flash memory address is equal to 0x080F FFFF for STM32F4xx devices

7 Revision history

Table 2. Document revision history

Date	Revision	Changes
12-Oct-2011	1	Initial release.