



STM32F3xx in-application programming using the USART

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 STM32F3 microcontroller product family 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. Additionally, it allows two methods to calculate CRC (software and hardware). This possibility is new in the STM32F3 microcontroller product family.

Table 1. Applicable products

Type	Part numbers/Product subclass
Microcontrollers	STM32F3xx STM32 F3 Analog & DSP

Contents

1	IAP overview	3
1.1	Principle	3
1.2	IAP driver description	3
2	Running the IAP driver	6
2.1	Configuring the HyperTerminal	6
2.2	Executing the IAP driver	6
3	IAP driver menu	7
3.1	Download image to the internal Flash memory	7
3.2	Upload image from the internal Flash memory	8
3.3	Execute the new program	8
3.4	Disable the write protection	8
4	STM32F3xx IAP implementation summary	10
5	User program conditions	11
6	Revision history	12

1 IAP overview

STM32F3Xx⁽¹⁾ devices are implemented in the STMicroelectronics STM32373C-EVAL/STM32303C-EVAL evaluation boards.

1.1 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 STM32F3xx's internal Flash memory.
- Upload the STM32F3xx's internal Flash memory content (starting from the defined user application address) into a binary file.
- Execute the user program.

1.2 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 the internal Flash memory, executing the binary file already loaded and disabling the write protection of the pages where the user loads the binary file (if they are write-protected).
- *flash_if.c* contains write, erase and disable write protection of the internal Flash memory.
- *common.c*: contains functions related to read/write from/to the USART peripheral.
- *ymodem.c*: used to receive/send the data from/to the HyperTerminal application using the YMODEM protocol⁽²⁾. In case of data reception failure, the "Failed to receive the file" error message is displayed. If the data is received successfully, it is programmed into the internal Flash memory from the appropriate address. A comparison between the contents of the internal RAM and the contents of the internal Flash memory is performed to check the integrity of the data. 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.
- STM32F3Xx Standard Peripherals Library.

1. STM32F3Xx refers to STM32F37x or STM32F30x.

2. The Ymodem protocol sends data in 1024-byte blocks. An error check is performed in data blocks transmitted to the STM32F3xx's 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 existing documentation.

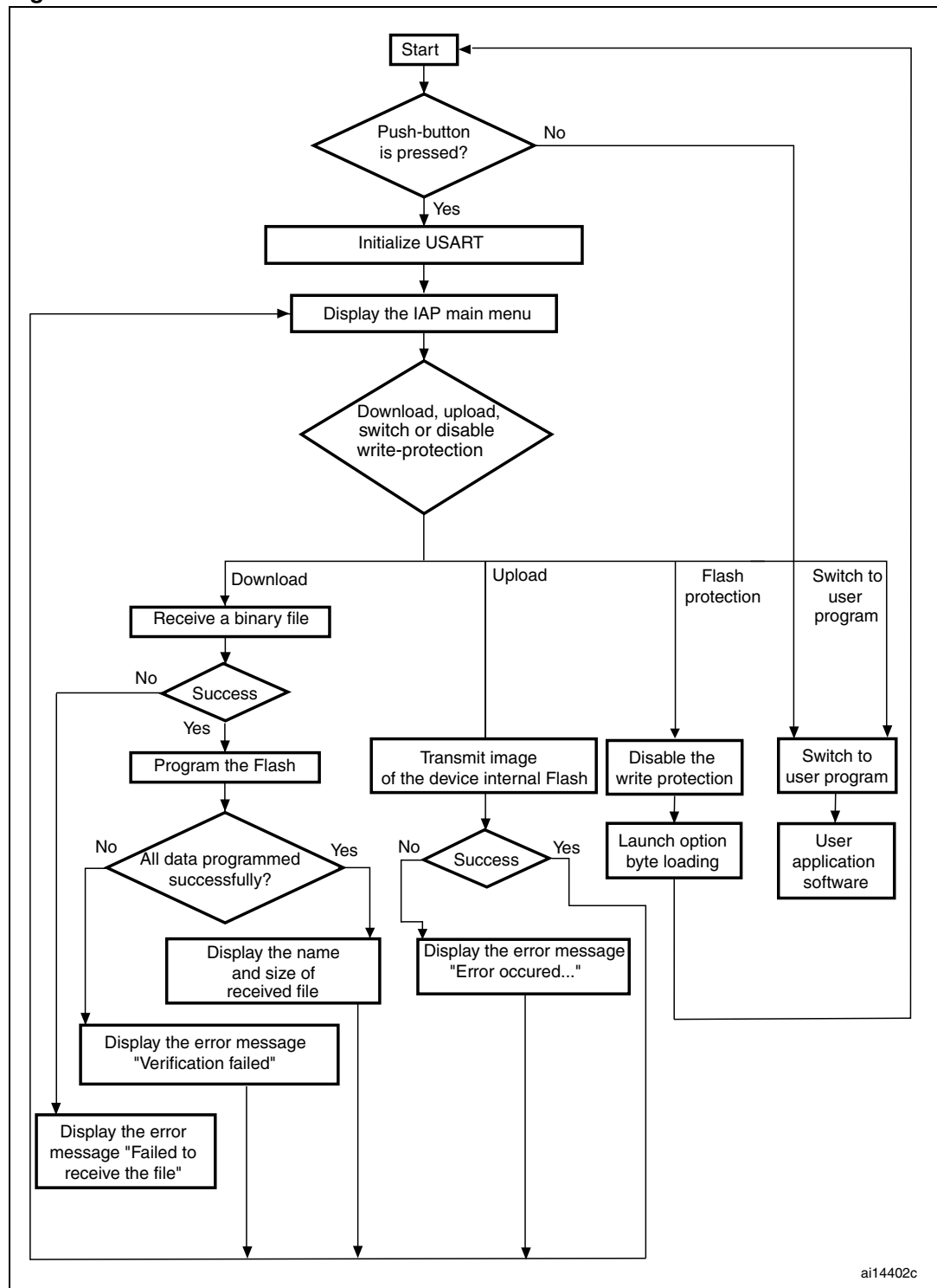
The user can choose to either go to the user application or execute the IAP for reprogramming purposes by pressing a push-button connected to a pin.

- 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 2: STM32F3xx IAP implementation](#) for further information.

The IAP flowchart is shown in [Figure 1](#).

Figure 1. Flowchart of the IAP driver

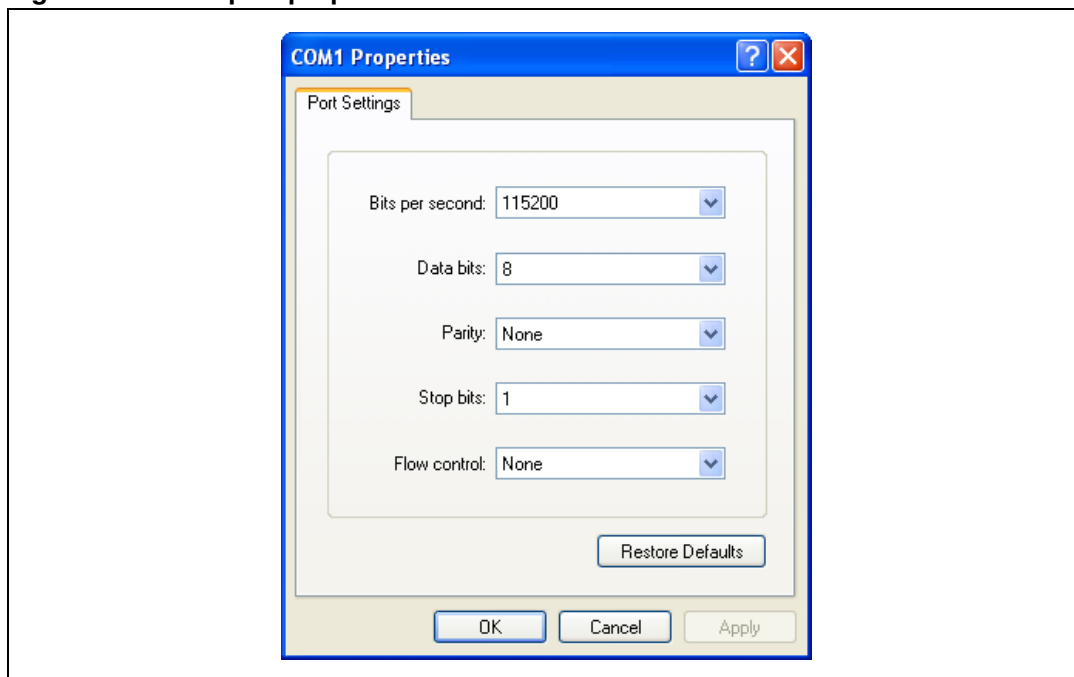


2 Running the IAP driver

2.1 Configuring the HyperTerminal

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

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.

2.2 Executing the IAP driver

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

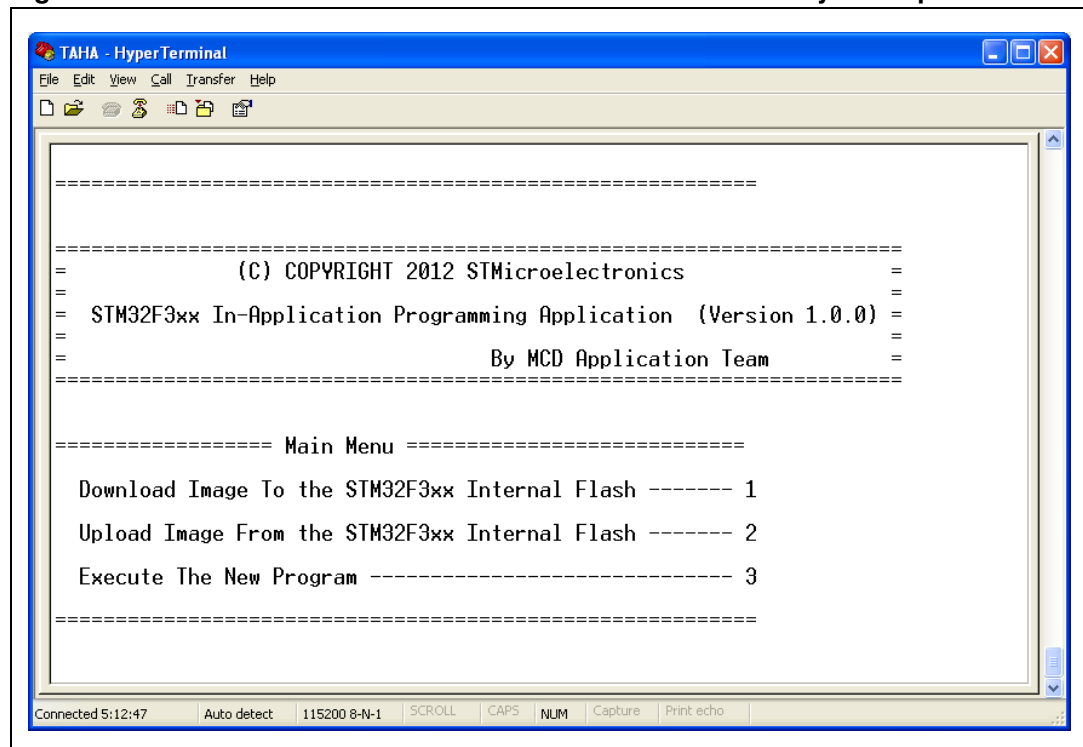
By pressing the push-button at reset, the user can run the IAP driver to reprogram the STM32F3xx's internal Flash memory. It is not mandatory to use the push-button; the user can apply a signal to this pin with respect to its active level. Refer to [Table 2: STM32F3xx IAP implementation](#).

3 IAP driver menu

When the provided IAP firmware is run, four application possibilities are displayed on the HyperTerminal.

- Download image into the internal flash memory.
- Upload image from the internal flash memory.
- Execute the new program.
- Disable the write protection (in the case where the Flash memory pages are write protected).

Figure 3. IAP driver menu when the STM32F3xx Flash memory is not protected



3.1 Download image to the internal Flash memory

Follow these steps to download a binary file via the HyperTerminal to the STM32F3xx's internal Flash memory.

1. Press **1** on the keyboard to select the **Download Image To the STM32F3xx 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 STM32F3xx's internal Flash memory from the defined base address and displays the binary file name and size in the HyperTerminal window.

3.2 Upload image from the internal Flash memory

Follow these steps to upload a copy of the internal Flash memory started from the user application address.

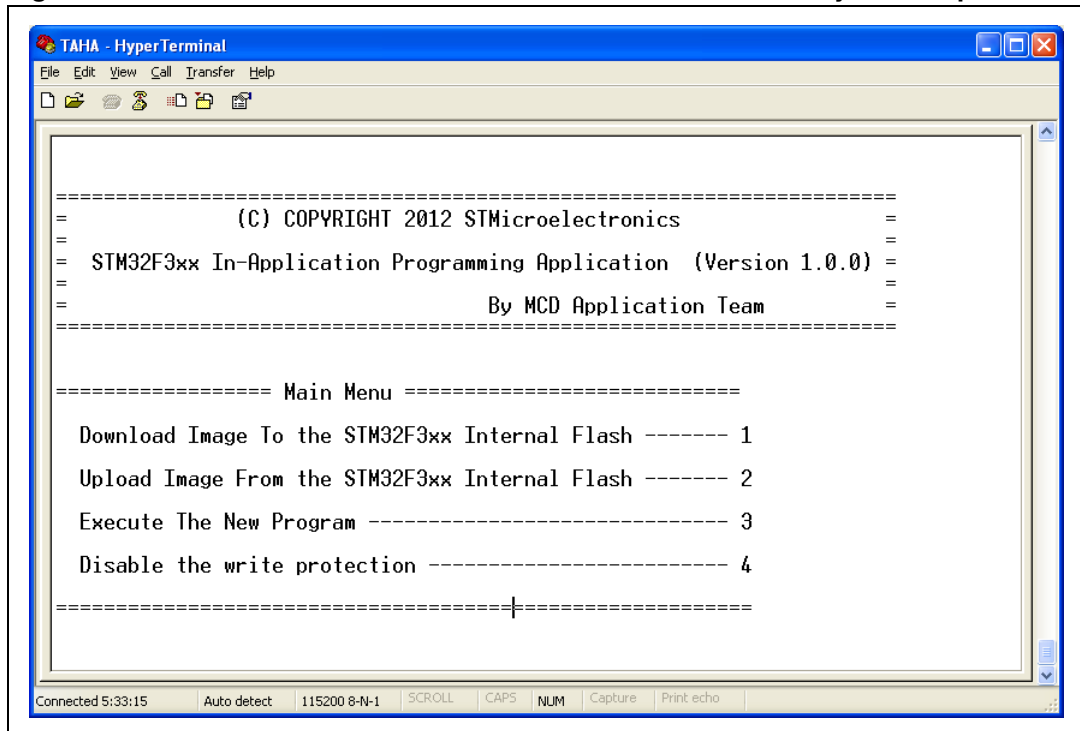
1. Press 2 on the keyboard to select **Upload image from the STM32F3xx internal Flash** menu.
2. Select **Receive File** in the **Transfer** menu.
3. Select the directory in which the binary file will be saved.
4. From the protocol list, select the **Ymodem** protocol.
5. Click on the **Receive** button.

3.3 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.

3.4 Disable the write protection

When the IAP starts, a write protection check is performed on the Flash memory pages where the user program is to be loaded. In case of write protection, the menu option allowing to disable it appears on the HyperTerminal window ([Figure 4](#)).

Figure 4. IAP driver menu when the STM32F3xx's 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, therefore the user has to verify that the Flash memory is not read-protected.*

4 STM32F3xx IAP implementation summary

[Table 2](#) provides a summary of the STM32F3xx IAP implementation.

Table 2. STM32F3xx IAP implementation

	Firmware			Hardware	
	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:	The user application (image to be downloaded with the IAP) will be programmed starting from address 0x8003000 ⁽¹⁾ The maximum size of the image to be loaded is:	The image is uploaded with the IAP from the STM32F3xx internal Flash. The size of the image to be uploaded is:	Push-button (active level: high)	USART used
STM32F3XX devices	page 0 - page 5	244 Kbytes (page 6 - page 127)	244 Kbytes (page 6 - page 127)	Key push-button connected to pin PA2 (STM32F37x) and PE6 (STM32F30x)	USART2

1. User application location address is defined in the *flash_if.h* file as: `#define APPLICATION_ADDRESS 0x8003000`. To modify it, change the default value to the desired one.

The STM32F3xx IAP package comes with the following.

- Source files and pre-configured projects for the IAP program (in the **Project\STM32F3xx_IAP** directory)
- Source files and pre-configured projects that build the application to be loaded into the Flash memory using the IAP (in the **Project\STM32F3xx_IAP\binary_template** directory).

The **readme.txt** file provided with this package describes, step-by-step, how to execute this IAP application.

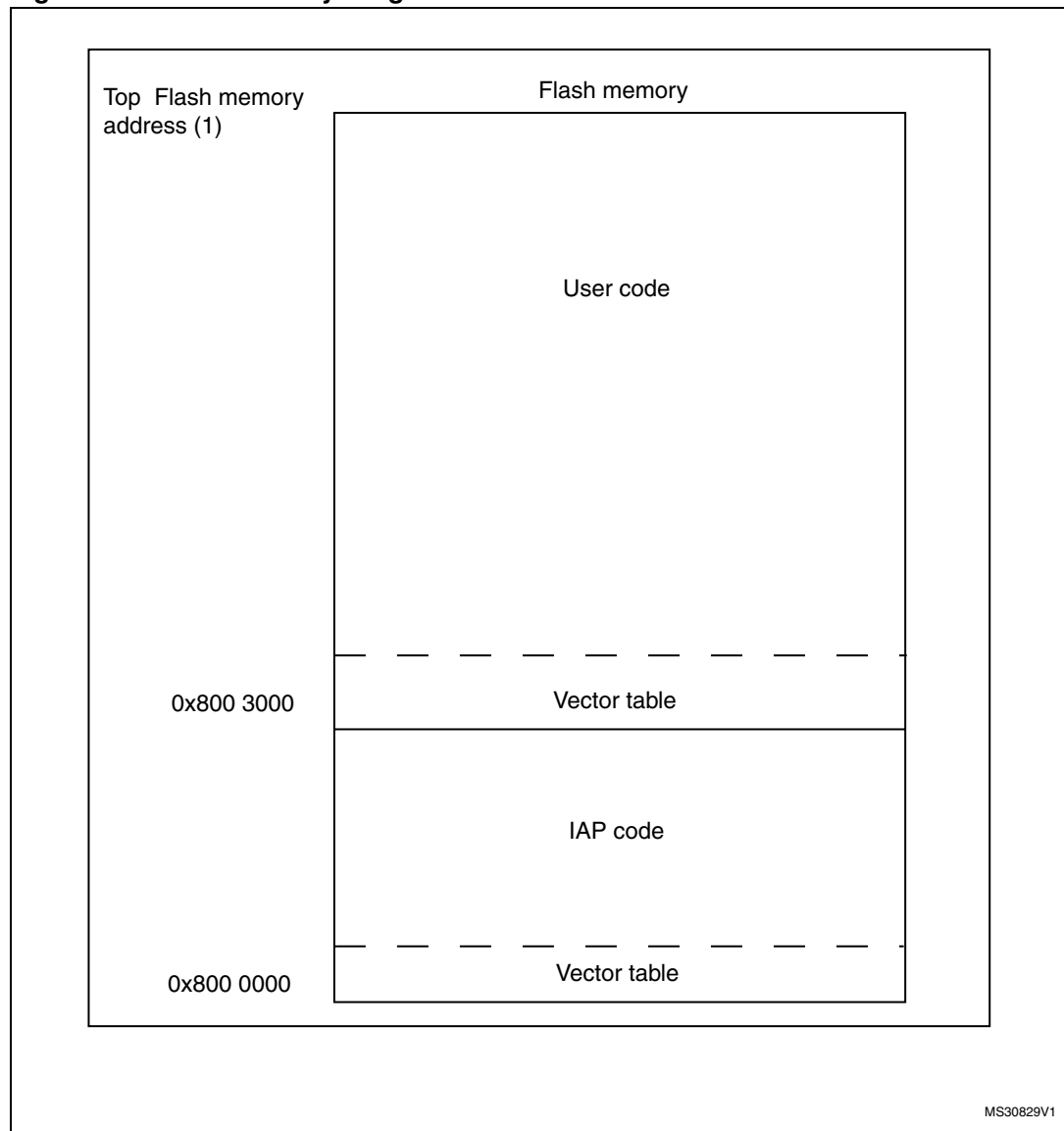
5 User program conditions

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 0x08003000, using your toolchain linker file
2. Relocate the vector table at address 0x08003000, using the "NVIC_SetVectorTable" function from the stm32f3xx_misc.h/.c driver."

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:
0x0803 FFFF for STM32F37x (or STM32F30x) devices

6 Revision history

Table 3. Document revision history

Date	Revision	Changes
20-Sep-2012	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2012 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com

