AN2078
Application note

STR7 In-Application Programming using UART

Introduction

An important requirement for most Flash-based systems is the ability to update firmware while the system is installed in the end product. This is referred to as In-Application Programming (IAP).

STR7xx MCUs have the capability of running user-specific firmware to perform In Application Programming of the MCU embedded Flash memory. This feature allows the use of any type of communication protocol for the reprogramming process (for example, CAN, UART, USB). In this application note, UART is used as an example.

This application note outlines general guidelines for creating an IAP application. The STR7xx Eval Boards were used to validate the IAP driver.

Software is provided and is available on *http://www.st.com/mcu*

Contents

1 IAP overview 3

1.1 Principle 3

1.2 IAP driver description 3

2 Running the IAP driver 5

2.1 HyperTerminal configuration 5

2.2 Triggering execution of the IAP driver 5

3 IAP driver menu 6

3.1 Download image to internal Flash 6

3.2 Execute the new program 6

4 STR7xx IAP implementation Summary 7

5 User program conditions 8

6 IAP restrictions 9

7 How to use the IAP driver 10

8 Revision history 12

1 IAP overview

1.1 Principle

The IAP driver must be programmed by the user in the Flash base address (0x00) via JTAG using a flasher toolset.

Using the UART, the IAP driver loads a binary file from the HyperTerminal to the STR7xx internal Flash, and then jumps to execute it.

1.2 IAP driver description

The following table shows the IAP driver code size for each STR7xx MCU.

STR71x	STR73x	STR75x
7.2 Kbytes	8.3 Kbytes	5.8 Kbytes

The IAP driver contains a set of source files:

- *main.c*: where the UART initialization and PLL configuration are set. Then, a main menu is executed from the *common.c* file.
- *common.c*: contains display functions and the main menu routine. The main menu gives the choice between loading a new binary file and executing the binary file already loaded.
- *ymodem.c* and *download.c*: they allow all data to be received from the HyperTerminal application (using the YMODEM protocol[1)]), then loaded into STR7 internal RAM. In the event of a failure in data receival, the 'Failed to receive the file' error message is displayed on the HyperTerminal window. However, if successful, the data is then programmed into internal Flash at the appropriate address. A comparison between internal RAM and internal Flash contents is performed to check the data intergrity. If there is any discrepency, the 'Verification failed' error message is displayed on HyperTerminal window. Other error messages are also displayed when the image size is higher than the allowed space memory and when there is an abort by the user.

Note: 1 Ymodem protocol sends data in 1024-byte blocks. A check for errors in data blocks transmitted to the STR7xx Internal RAM is applied: A comparison between the transmitted and received data is achieved. Blocks received unsuccessfully are acknowledged with a NAK (Negative Acknowledgement). For more details about the Ymodem protocol, refer to existing documentation.

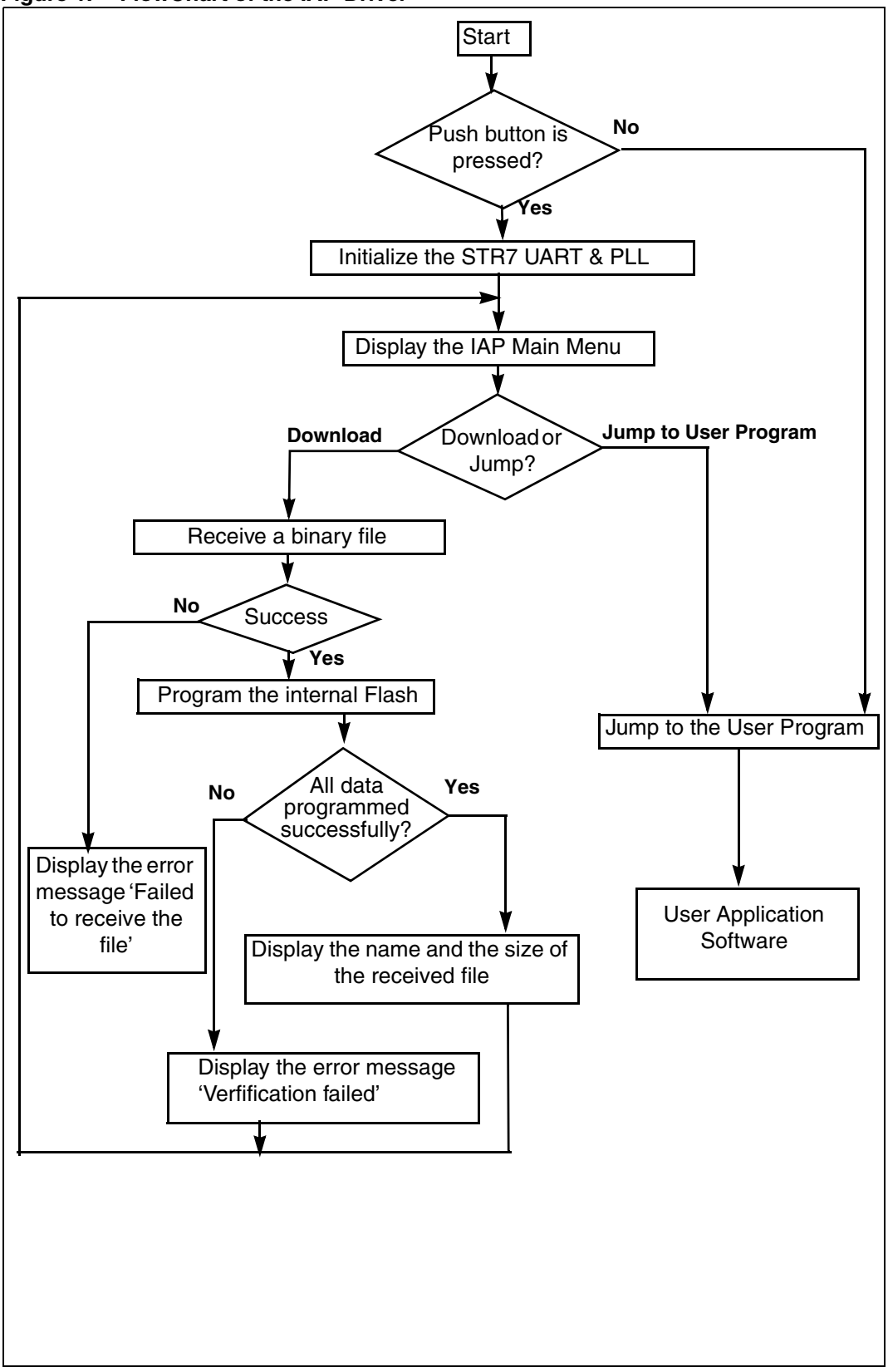
- Some STR7xx Standard library and Flash library files.

A pin connected to a push button is used to select between jumping to the user application and executing the IAP for reprogramming purpose:

- Push button is not pressed at reset: jump to the user application
- Push button is pressed at reset: the IAP main menu is displayed

Figure 1 shows the IAP flowchart.

Figure 1. FlowChart of the IAP Driver



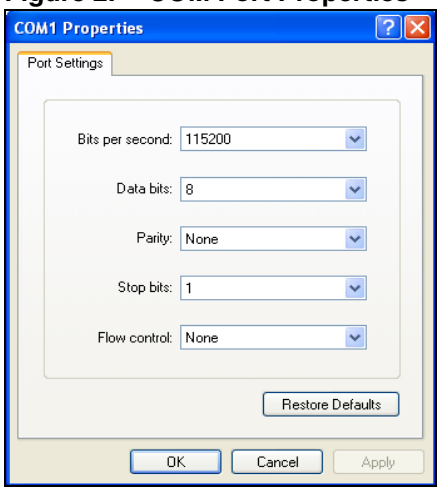
2 Running the IAP driver

To run the IAP driver, STR7xx internal Flash must be hardware remapped at address 0x0. Thus the IAP driver is programmed in Flash bank0 sector0 and the user application in the remaining memory space in bank0.

2.1 HyperTerminal configuration

The IAP requires a PC running HyperTerminal with the following settings:

Figure 2. COM Port Properties



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

Care should be taken when selecting the system clock frequency. Ensure that with the system clock frequency used in your application, you are able to generate a baud rate equal to 115200 bps in order to guarantee successful communication via UART.

2.2 Triggering execution of the IAP driver

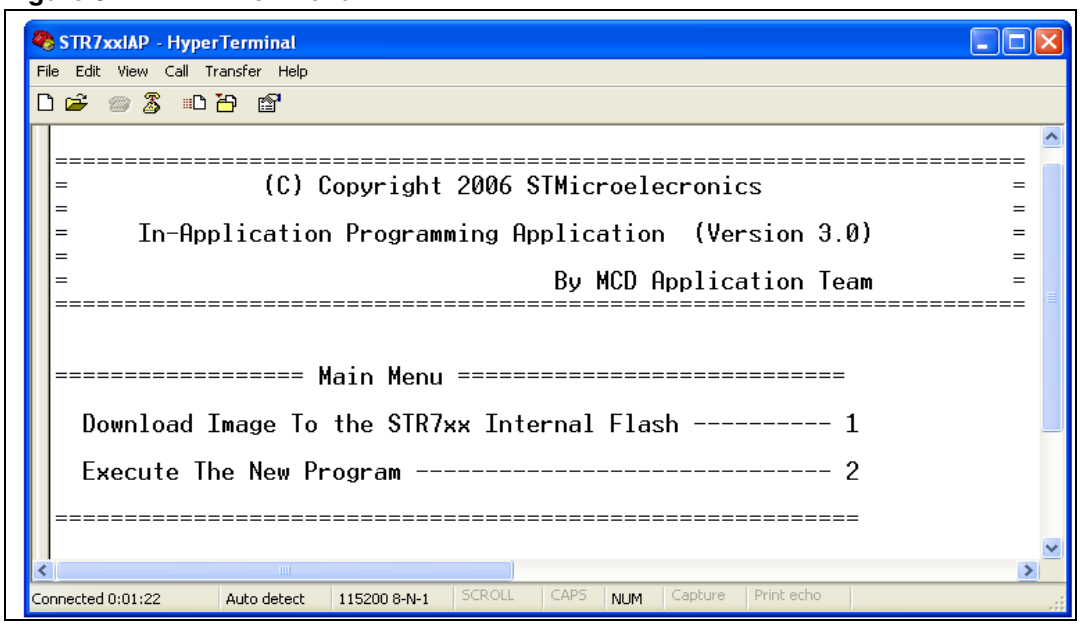
In this application note, depending on the micro eval board, pins connected to push buttons are used as an example, allowing execution of the IAP driver. By keeping the push button pressed at reset, the user is able to run the IAP driver to reprogram the STR7xx internal Flash.

It is not mandatory to use push buttons and the user can simply enter signals on these pins with respect to their corrsponding active levels. Refer to the table in *Section 4 on page 7* for the correct pins and their active levels for each MCU.

3 IAP driver menu

The execution of the IAP firmware results in the following menu displayed in the HyperTerminal window.

Figure 3. IAP Driver menu



3.1 Download image to internal Flash

To download a binary file via HyperTerminal to the STR7xx internal Flash, follow the following procedure:

- Press **"1"** on the keyboard to choose the menu **"Download Image To Internal Flash"**

Now, in the Transfer menu, select **"Send File"**:

- In the **filename** field, type the name and the path of the binary file you want to send.
- In the protocol list, choose the **Ymodem** protocol,
- Click the **"Send"** button.

Following these steps, the IAP driver loads the binary file into the STR7xx internal Flash from bank0 sector1 base address and displays the binary file name and file size in the HyperTerminal window.

3.2 Execute the new program

After loading the new program from HyperTerminal by selecting the **"Download Image To Internal Flash"** menu, you can execute your code by selecting the **"Execute The New Program"** menu by pressing **"2"** from the keyboard.

4 STR7xx IAP implementation Summary

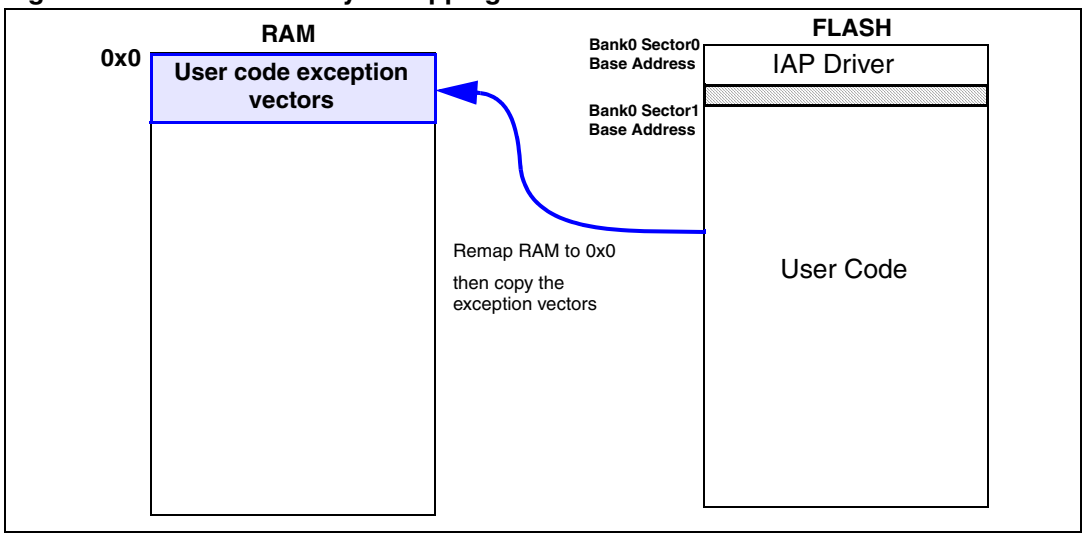
A summary of the STR7xx IAP implementation is described in the below table :

	STR71x	STR73x	STR75x
IAP driver location	0x40000000	0x80000000	0x20000000
User application location	0x40002000	0x80002000	0x20002000
Flash remapping	For STR7xx devices, the Flash must be hardware remapped at address 0x00.		
Push button Active Level	P1.8 (SEL) 0	P5.10 (SEL) 1	P1.05 (KEY) 0
Flash routines	For STR7xx devices, Flash routines (program/ erase) must be executed from RAM given that read-while-write operations in one Bank are not allowed. So, program/erase operations are performed from RAM.		
UART used	UART0		

5 User program conditions

On STR7xx MCUs, the IAP driver should be programmed at the Flash base address (sector0 base address) and the user application should be programmed from bank0 sector1 base address. If the application to be programmed to the STR7xx internal Flash uses interrupts, it should provide software remapping of RAM to address 0x0 and also the exception vectors should be copied to this same address as illustrated in *Figure 4*.

Figure 4. STR7xx memory remapping conditions



6 IAP restrictions

The STR7xx IAP driver code is downloaded to the bank0 sector0. So, the maximum size of the image to be loaded is 248 Kbytes (bank 0 size - sector 0 size).

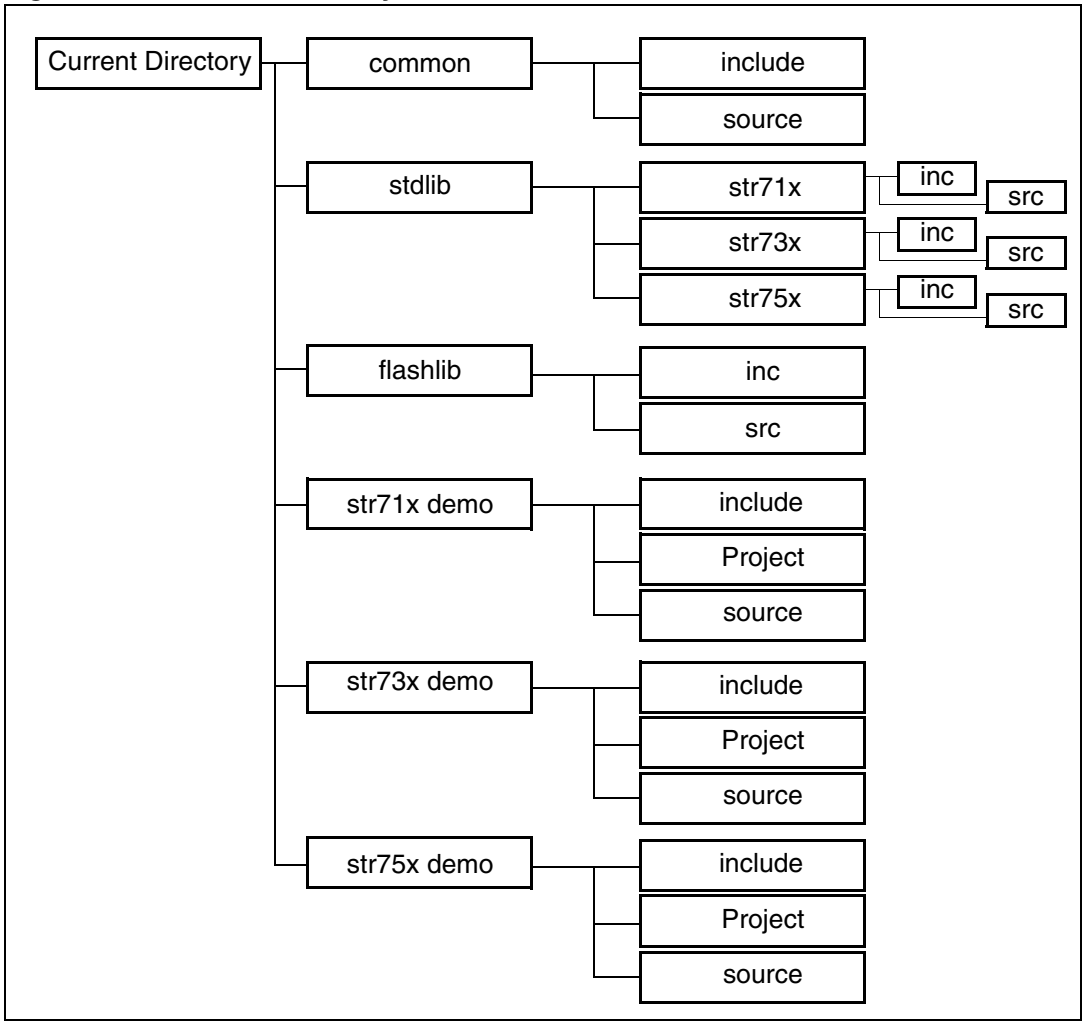
7 How to use the IAP driver

The IAP package consists of an archived file containing 6 subfolders:

- **common**: It contains the files commonly used by the STR7xx MCUs.
- **stdlib**: It contains the STR7xx standard libraries.
- **flashlib**: The STR7xx Flash library which is a set of C routines containing everything needed to program the embedded Flash of STR7xx devices.
- **str71x demo**: It contains the main application and a project.
- **str73x demo**: It contains the main application and a project.
- **str75x demo**: It contains the main application and a project.

The file structure is described in the following figure:

Figure 5. IAP driver directory structure



To efficiently use the IAP driver, please follow the different steps below:

- Download the IAP driver into the STR7xx internal Flash via JTAG using a Flasher toolset.
- Open an HyperTerminal Window with the settings already defined in *Section 2.1*
- To run the IAP driver, keep the push button (according to the used STR7xx Eval board) pressed at Reset. The IAP main menu is then displayed on the HyperTerminal window.
- To download an application, press 1 and use the YMODEM protocol as described in *Section 3.1*.

8 Revision history

Table 1. Revision history

Date	Revision	Description of changes
31-Jan-2005	1	First release
28-Feb-2006	2	Updated *Introduction on page 1* Updated *Figure 1 on page 4* Updated *Section 2: Running the IAP driver on page 5* Updated *Figure 3 on page 6* Updated *Section 3.1: Download image to internal Flash on page 6* Added new section: *Section 3.2: Execute the new program on page 6* Updated *Section 5: User program conditions on page 8* IAP Driver updated to load up to 248 KBytes Updated *Section 1: IAP overview on page 3*
12-Jan-2007	3	Document expanded to encompass STR71x, STR73x, STR75x.

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 AN AUTHORIZED ST REPRESENTATIVE, 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.

© 2007 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 - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com