AN2078
APPLICATION NOTE

STR71x In-Application Programming using UART

INTRODUCTION

In-Application Programming (IAP) has been implemented for users who want their application software to update itself by re-programming the embedded Flash memory during execution.

The STR71x embedded Flash memory has two banks (bank 0 & bank 1). Every bank contains a set of sectors which can be write-protected. The sector 0 of the bank 0 must contains the application software driver that is used to re-program all other embedded Flash sectors.

The IAP driver must be first programmed by the user in sector 0 of bank 0 using In-Circuit Programming (ICP).

1 IAP DRIVER FEATURES

The IAP Driver has the following features:

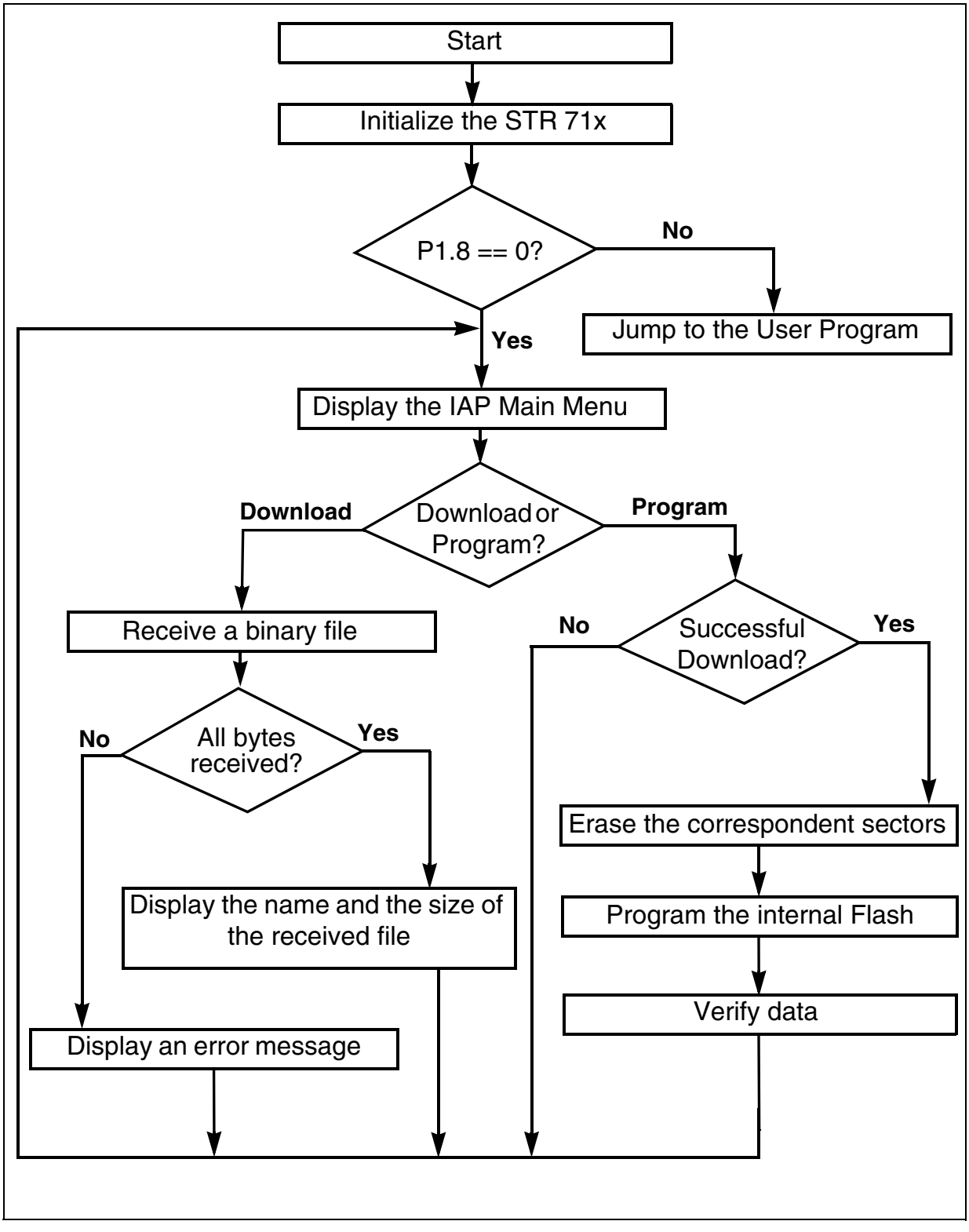
– The IAP Driver code size: 7.84Kbytes,

– The maximum size of the image to be loaded: 56 Kbytes.

2 IAP DRIVER FLOWCHART

Figure 1 shows the FlowChart of the IAP Driver.

Figure 1. FlowChart of the IAP Driver



3 RUNNING THE IAP DRIVER

To run the IAP Driver, the internal Flash must be hardware remapped at address 0x0. Therefore, the IAP Driver must be programmed in sector 0 of the Flash bank 0.

After the Reset, the IAP Driver is executed. It checks the P1.8 pin level, if high it jumps to the user application program, otherwise it continues its execution.

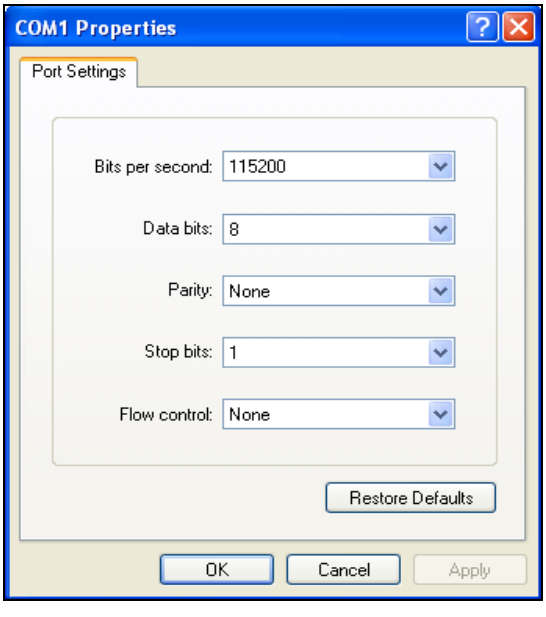
The IAP allows the user code to be loaded into internal RAM via UART0 interface, then programs this code into internal Flash.

A PC running the HyperTerminal application must be used to allow the communication with the IAP Driver.

3.1 HYPERTERMINAL CONFIGURATION

Launch HyperTerminal, create a new connection and configure the port settings according to Figure 2 below.

Figure 2. COM Port Properties

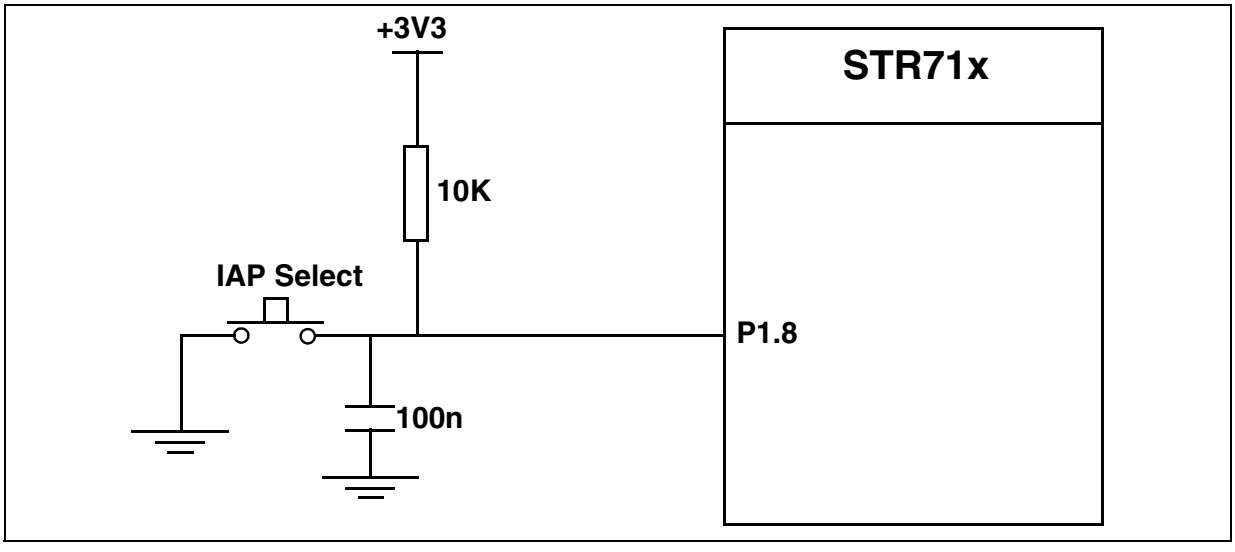


3.2 P1.8 PIN CONFIGURATION

To execute the IAP driver, the P1.8 pin must be maintained low after Reset. As a typical example, this pin must be connected to a push button.

Figure 3 illustrates how to realize this typical solution.

Figure 3. Typical hardware solution

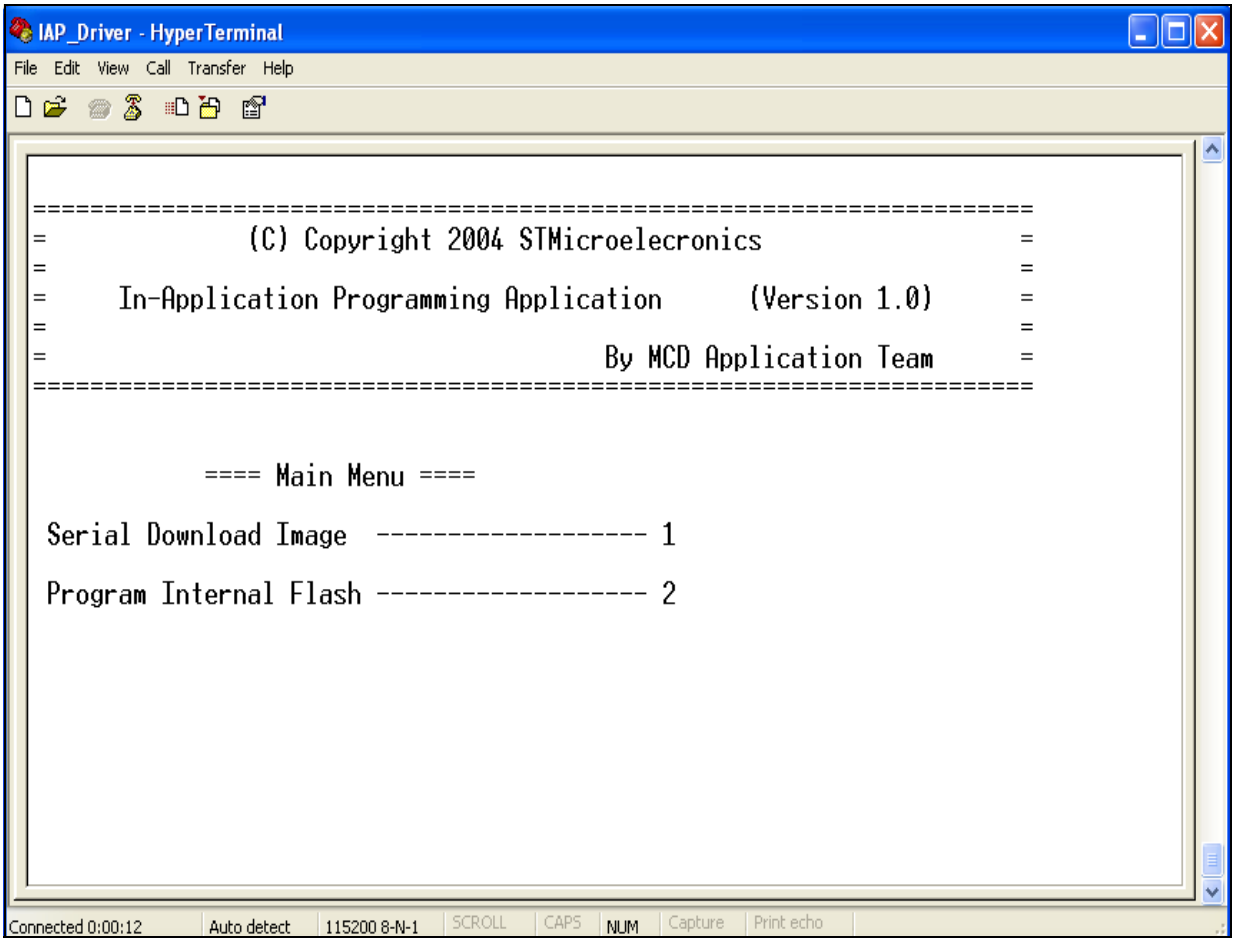


4 IAP DRIVER MENU

After the HyperTerminal and the P1.8 pin configurations, you can execute the IAP driver.

Figure 4 illustrates the menu of the IAP driver which will be displayed in the HyperTerminal window.

Figure 4. IAP driver menu



5 DOWNLOAD A BINARY FILE

To download a binary file via HyperTerminal to internal RAM of the STR71x, you must follow the following procedure:

– Press the **"1"** key from the keyboard to choose the menu **"Serial Download Image"**

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.

The binary file will be loaded into the internal RAM from the address 0x20002000 and the IAP driver will display its name and size in the HyperTerminal window.

6 PROGRAM INTERNAL FLASH

To program the STR71x embedded Flash, follow the following procedure:

– Press the **“2”** key from the keyboard to choose the menu **“Program Internal Flash”**,

After pressing **“2”**, the IAP driver will copy the correspond image from internal RAM to the STR71x embedded Flash from the address 0x40002000.

7 IAP MESSAGES

Status and Errors messages are managed by the IAP driver allowing you to follow the status and the result of the Programming process.

During the two operations, download and program, the status messages are displayed in the HyperTerminal window to inform you about the current operation status. These messages are classified into two types of message:

– Status messages

– Error messages

7.1 STATUS MESSAGES

These messages keep you informed of the status of the current operation. For example after a download operation, these parameters are displayed in the HyperTerminal window:

– The status of the transfer,

– The name of the received file,

– The size in bytes of the received file.

As an example of a status message which summarizes the precedent information:

“Transfer Completed!”

“Name: Pin_Toggling.bin”

“Size: 1380 Bytes”.

7.2 ERROR MESSAGES

The error messages keep you informed if the current operation fails by describing the cause, if known.

The messages are:

– **“Failed to receive the file!”**: This message is generated when the file is not correctly downloaded into the internal RAM,

– **“Invalid Command Arguments!”**: This message is generated when an invalid command argument is detected,

– **"No File to be Programmed!"**: This message is generated when trying to program internal Flash and the file is not correctly downloaded into the internal RAM,
– **"Flash Program Failed!"**: This message is generated when the program operation from internal RAM to the STR71x embedded Flash has failed,
– **"Data Verification Failed!"**: This message is generated when the content of the RAM and the content of the Flash are different,
– **"Unknown failure!"**: This message is generated when an unknown failure is produced.

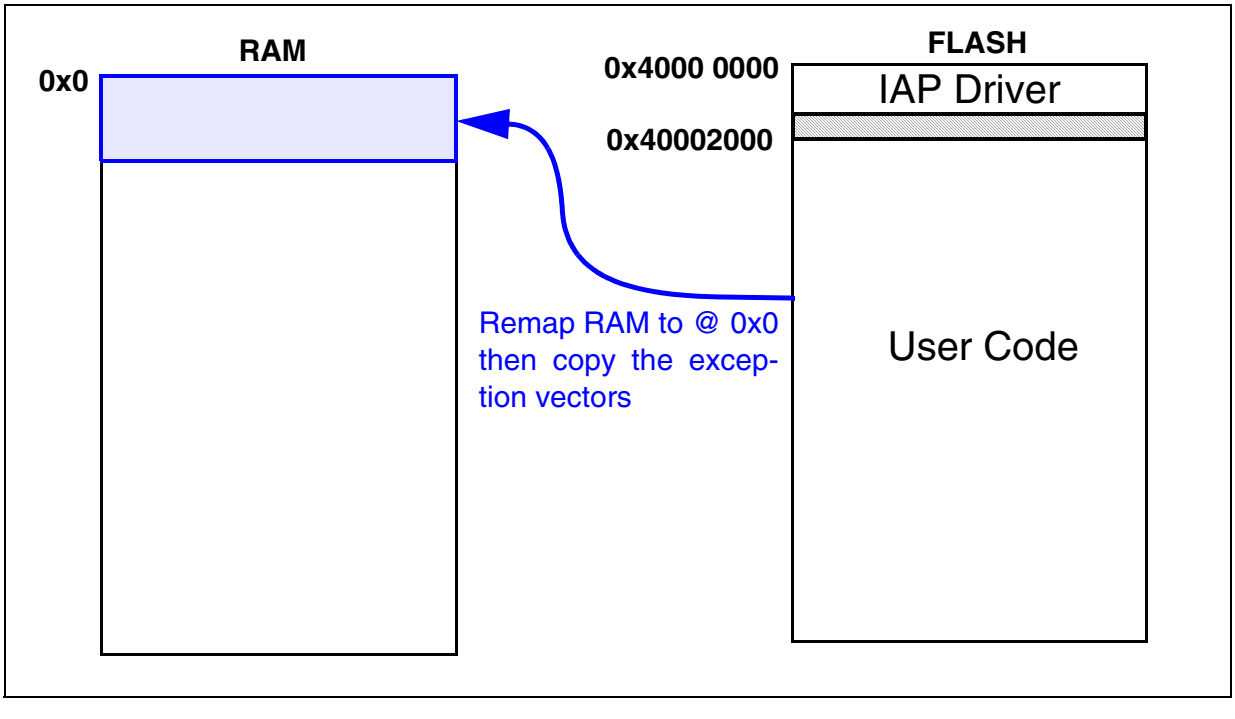
8 USER PROGRAM CONDITIONS

The IAP driver must be programmed at the address 0x40000000 and the user program must be programmed from address 0x40002000.

If the code to be programmed to the internal Flash does not use interrupts, you can program it to the internal Flash without any conditions, otherwise, the user code must provide a software remap of the RAM to the address 0x0 and copy the exception vectors to this same address.

The Figure 5 illustrates the precedent conditions.

Figure 5. Memory remapping conditions



“THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.”

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics.

All other names are the property of their respective owners

© 2005 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