AN2557
Application note

STM32F10xxx 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 STM3210B-EVAL/STM3210E-EVAL board was used to validate the IAP driver.

The STM32F10xxx 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 (such as CAN, USART, USB). USART is the example used in this application note.

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 Executing the IAP driver 5

3 IAP driver menu 6

3.1 Download image to the internal Flash memory 6

3.2 Execute the new program 6

3.3 Disabling the write protection 7

4 STM32F10xxx IAP implementation summary 8

5 User program conditions 9

6 IAP restrictions 9

7 How to use the IAP driver 10

8 Revision history 12

1 IAP overview

Medium-density devices are STM32F101xx and STM32F103xx microcontrollers where the Flash memory density ranges between 32 and 128 Kbytes. Medium-density devices are implemented in the STMicroelectronics STM3210B-EVAL evaluation board.

High-density devices are STM32F101xx and STM32F103xx microcontrollers where the Flash memory density ranges between 256 and 512 Kbytes. High-density devices are implemented in the STMicroelectronics STM3210E-EVAL evaluation board.

1.1 Principle

The IAP driver must be programmed from the Flash memory base address via the JTAG or SWD interface using the development toolchain chosen by the user. This driver uses the USART to load a binary file from the HyperTerminal to the STM32F10xxx's internal Flash memory, and then executes it.

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 *common.c* file.
- *common.c*: contains display functions and the main menu routine. The main menu gives the options of loading a new binary file, 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).
- *ymodem.c* and *download.c*: they are used to receive the data from the HyperTerminal application (using the YMODEM protocol[a], and then to load them into the STM32F10xxx's internal RAM. In the event of a failure when receiving the data, 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 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.
- STM32F10xxx firmware library

To specify the device (High-density or Medium-density) on which the software is to be run, uncomment the relative definition line in the `platform_config.h` header file:

```
//#define USE_STM3210B_EVAL
//#define USE_STM3210E_EVAL
```

a. The Ymodem protocol sends data in 1024-byte blocks. An error check is performed in data blocks transmitted to the STM32F10xxx'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.

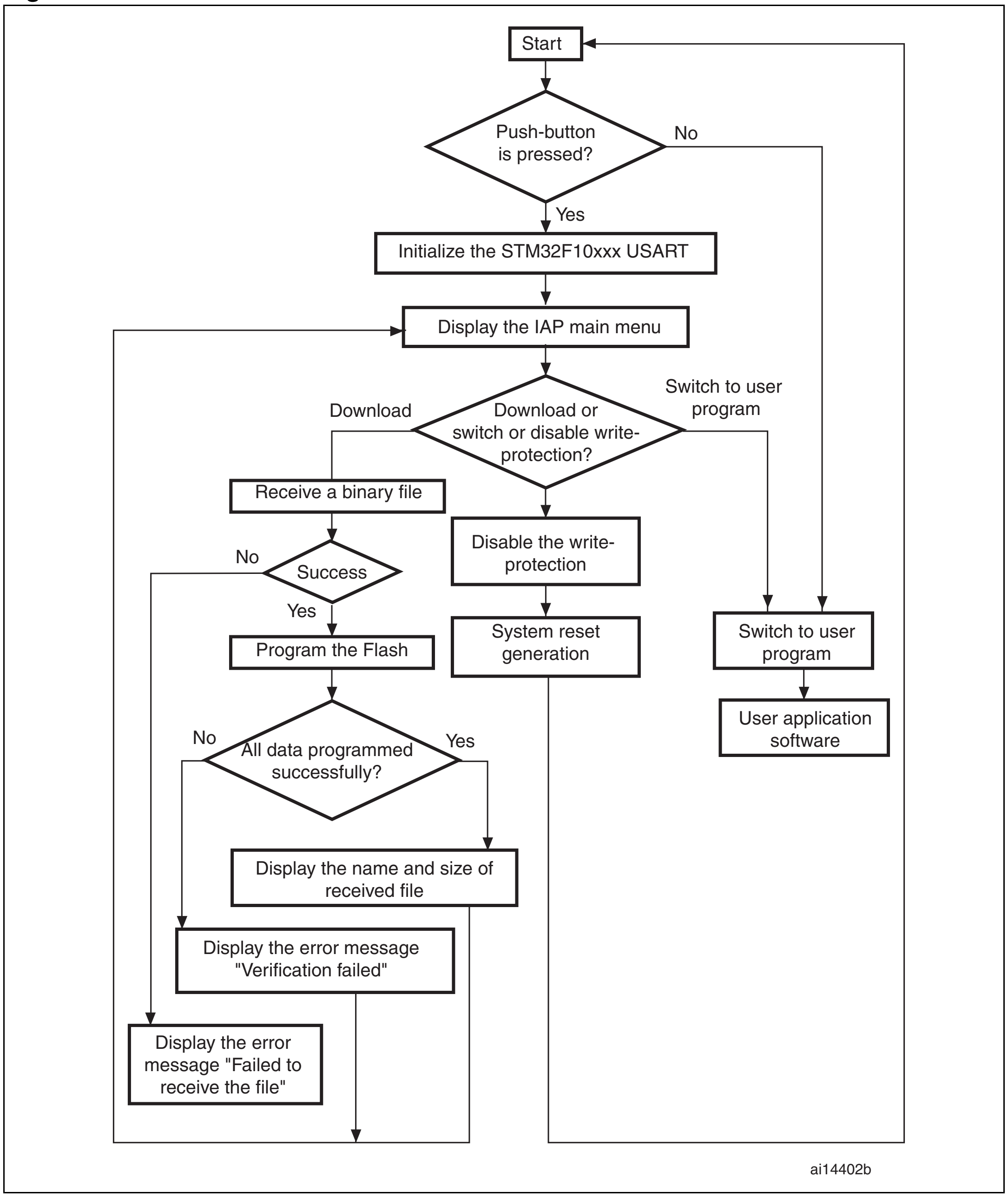
The user can choose to either go to the user application or execute the IAP for reprogramming purposes by pressing a push-button connected to a pin.

- 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 1.: STM32F10xxx IAP implementation* for more details about the STM3210B-EVAL/STM3210E-EVAL board push-button used to enter the IAP mode.

The IAP flowchart is represented in *Figure 1*.

Figure 1. Flowchart of the IAP driver


Start
Push-button is pressed?
No
Yes
Initialize the STM32F10xxx USART
Display the IAP main menu
Download
Download or switch or disable write-protection?
Switch to user program
Receive a binary file
No
Success
Yes
Program the Flash
Disable the write-protection
System reset generation
Switch to user program
User application software
No
All data programmed successfully?
Yes
Display the name and size of received file
Display the error message "Verification failed"
Display the error message "Failed to receive the file"
ai14402b

2 Running the IAP driver

The IAP driver is programmed in Flash memory:

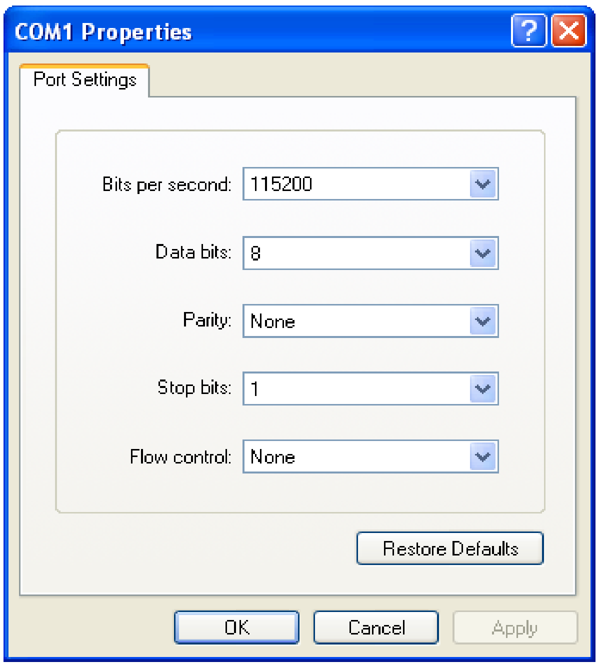
- from page 0 to page 7 on Medium-density devices
- from page 0 to page 3 on High-density devices

The user application occupies the remaining memory space.

2.1 HyperTerminal configuration

To use the IAP, the user must have a PC running HyperTerminal and configured as shown in *Figure 2*.

Figure 2. COM port properties



Bits per second: 115200
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

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

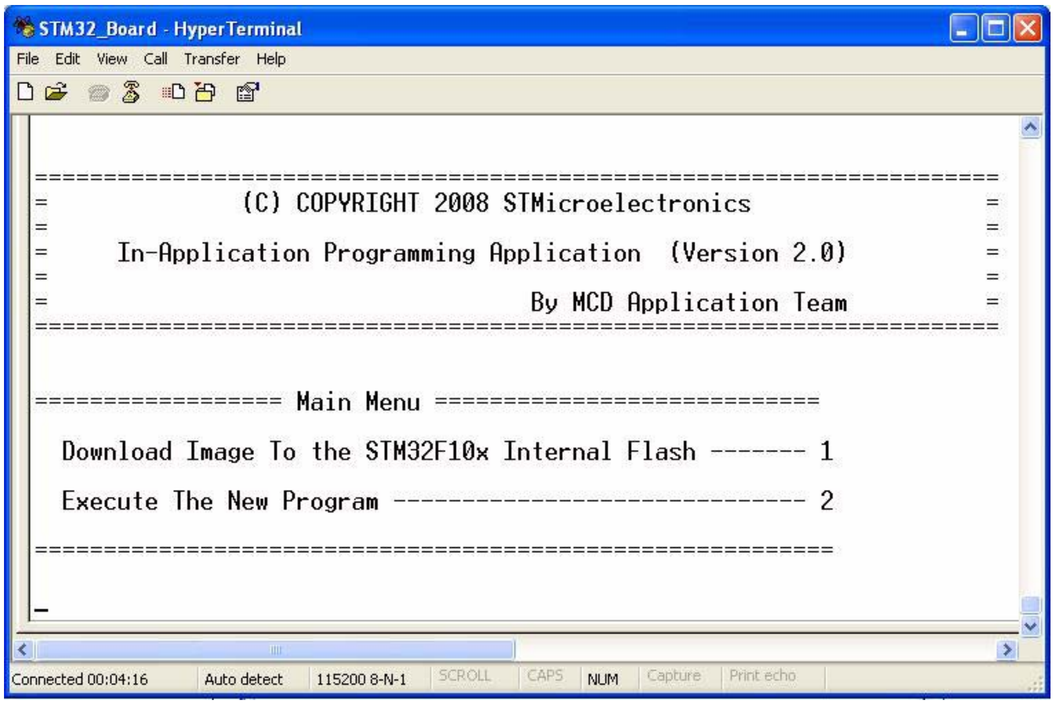
As an example in this application note, pressing the pin connected to the push-button allows the IAP driver to run.

By pressing the push-button at reset, the user can run the IAP driver to reprogram the STM32F10xxx'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 1: STM32F10xxx IAP implementation on page 8*.

3 IAP driver menu

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

Figure 3. IAP Driver menu when the STM32F10xxx Flash memory is not protected



```
=====================================================================
=              (C) COPYRIGHT 2008 STMicroelectronics                =
=                                                                   =
=     In-Application Programming Application  (Version 2.0)         =
=                                                                   =
=                                  By MCD Application Team          =
=====================================================================

================== Main Menu ============================

  Download Image To the STM32F10x Internal Flash ------- 1

  Execute The New Program ------------------------------ 2

==========================================================

_
```

3.1 Download image to the internal Flash memory

To download a binary file via HyperTerminal to the STM32F10xxx's internal Flash memory, do as follows:

1. Press **1** on the keyboard to select the **Download Image To the STM32F10x 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 STM32F10xxx's internal Flash memory from the defined base address and displays the binary file name and size in the HyperTerminal window.

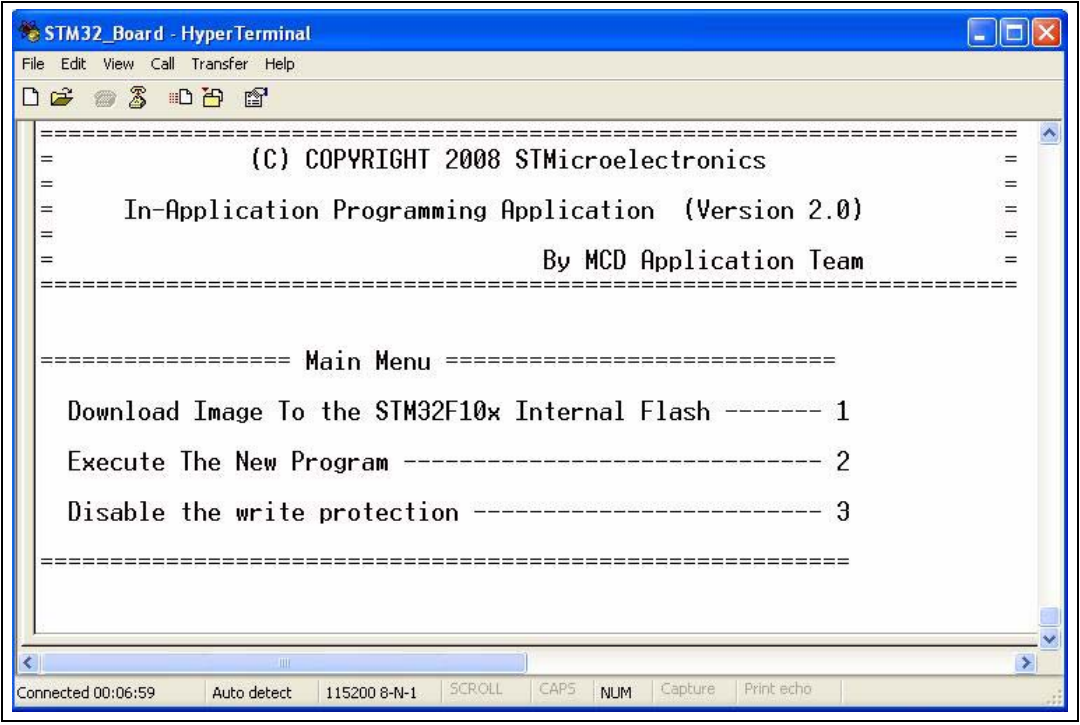
3.2 Execute the new program

Once the new program has been loaded, press **2** on the keyboard to select the **Execute The New Program** menu and execute the code.

3.3 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 it is the case, the menu shown in *Figure 4* appears.

Figure 4. IAP driver menu when the STM32F10xxx Flash memory is write-protected


STM32_Board - HyperTerminal
File Edit View Call Transfer Help

```
=======================================================================
=              (C) COPYRIGHT 2008 STMicroelectronics                  =
=                                                                     =
=     In-Application Programming Application  (Version 2.0)           =
=                                                                     =
=                                By MCD Application Team              =
=======================================================================


================== Main Menu ============================

  Download Image To the STM32F10x Internal Flash ------- 1

  Execute The New Program ------------------------------ 2

  Disable the write protection ------------------------- 3

==========================================================
```

Prior to downloading the new program, the write protection must be disabled. To do so, press **3** (**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.

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

4 STM32F10xxx IAP implementation summary

Table 1 provides a summary of the STM32F10xxx IAP implementation.

Note: The information is identical for Medium-density (STM3210B-EVAL) and High-density devices (STM3210E-EVAL) unless otherwise specified.

Table 1. STM32F10xxx IAP implementation

STM32F10xxx			
IAP application	Location		0x8000000
	Number of Flash memory pages used	STM3210B-EVAL board	8 (page size: 1 KByte)
		STM3210E-EVAL board	4 (page size: 2 KByte)
User application location			0x8002000(1)
Push-button	Assigned pin	STM3210B-EVAL board	PB.09 (KEY push-button)
		STM3210E-EVAL board	PG.08 (KEY push-button)
	Active level		Low
Flash routines			For STM32F10xxx devices, Flash routines (program/erase) are executed from the Flash memory
USART used			USART1

1. User application location address is defined in the *common.h* file as: `#define ApplicationAddress 0x8002000`. To modify it, change the default value to the desired one.

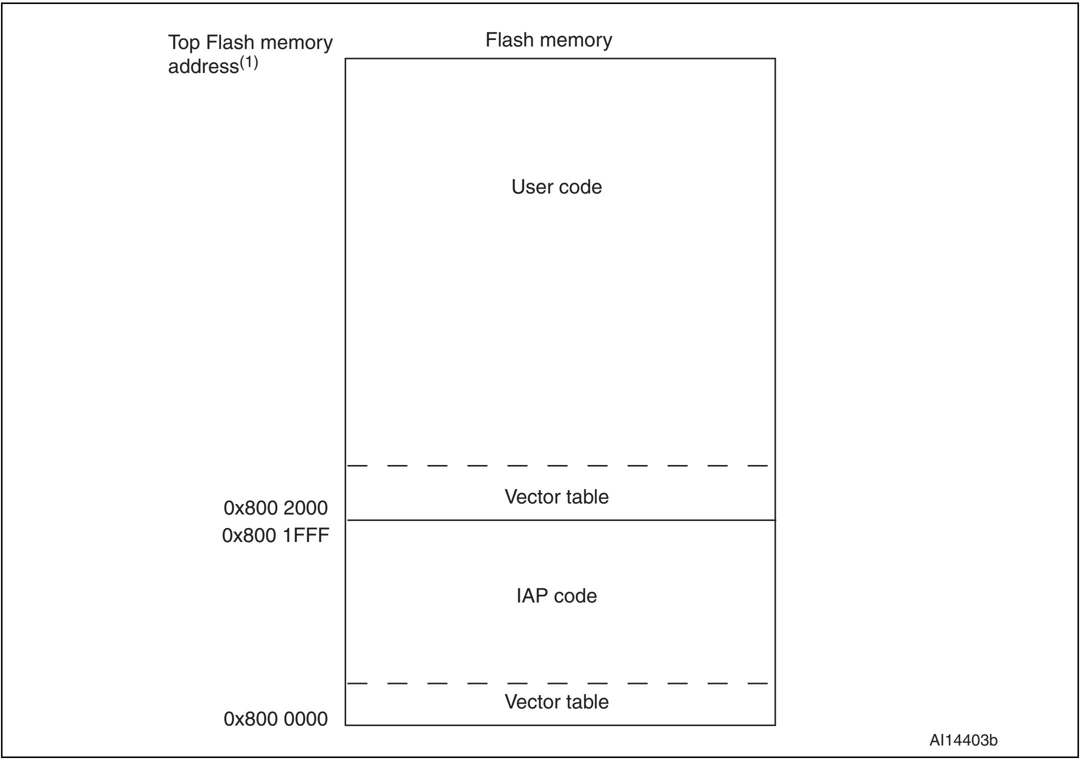
5 User program conditions

On STM32F10xxx microcontrollers, the user must program the IAP application from the Flash memory base address (0x0800 0000), and the user application from address 0x8002000.

As shown in *Figure 5*, the user application vector table must be located at address 0x8002000.

An example application program to be loaded with the IAP application is provided with preconfigured projects. Refer to *Section 7: How to use the IAP driver* for more details.

Figure 5. Flash memory usage



1. Top Flash memory address is equal to 0x0801 FFFF for Medium-density devices, and to 0x0807 FFFF for High-density devices.

6 IAP restrictions

The STM32F10xxx IAP driver code is downloaded:

- from page 0 to page 7 for Medium-density devices
- from page 0 to page 3 for High-density devices

Therefore, the maximum size of the image to be loaded is:

- 120 Kbytes (page 8 - page 127) for Medium-density devices.
- 504 Kbytes (page 4 - page 255) for High-density devices.

7 How to use the IAP driver

The IAP package is supplied in a single zip file. The extraction of the zip file generates a folder that contains the subfolders shown in *Figure 6.*

FWLib folder

The **FWlib** folder contains all the subdirectories and files that make up the core of the STM32F10xxx firmware library:

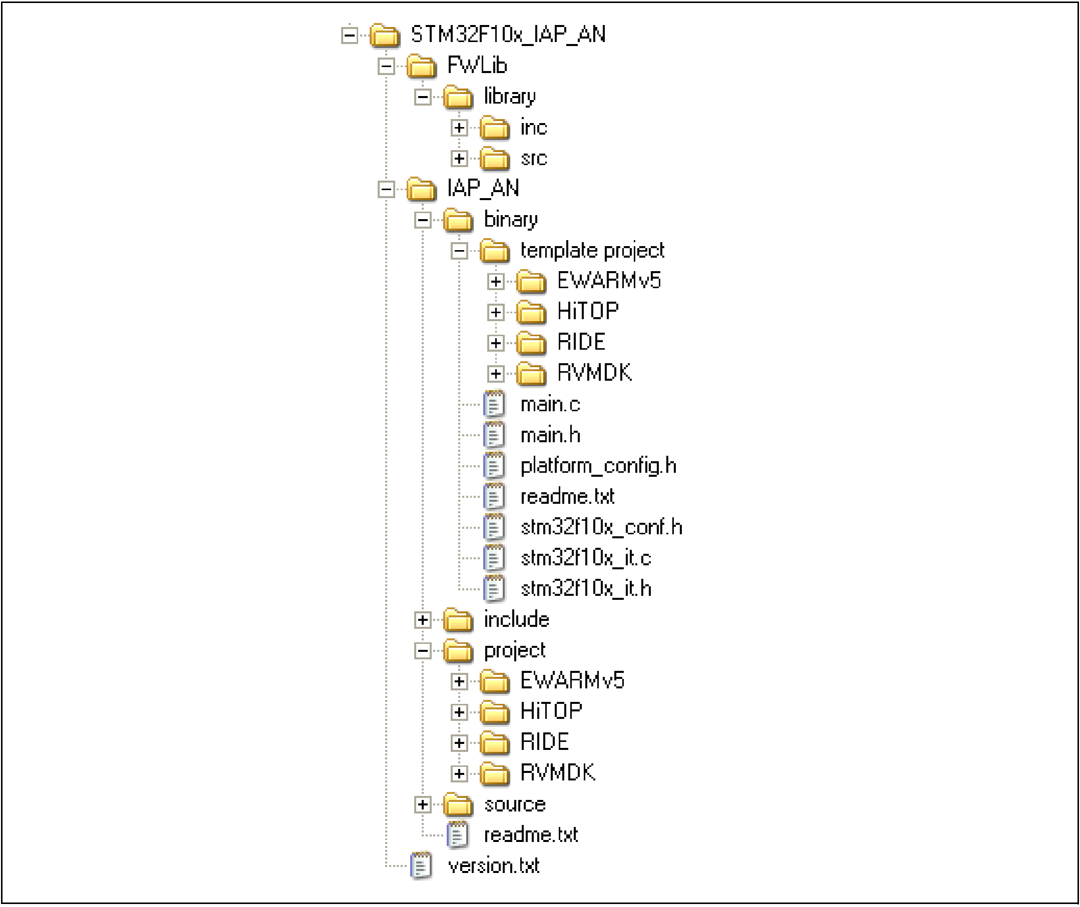
- ***inc*** subfolder contains the firmware library header files
- ***src*** subfolder contains the firmware library source files

IAP_AN folder

The IAP_AN folder contains all the subdirectories and files that make up the core of the IAP:

- ***binary*** subfolder contains a set of sources files and preconfigured projects, that describe how to build an application to be loaded into Flash memory using IAP. A "*readme.txt*" file is provided, describes how to use the directory contents.
- ***include*** subfolder contains the IAP firmware header files
- ***source*** subfolder contains the IAP firmware source files
- ***project*** subfolder contains three projects that compile all the IAP files:
  - **EWARMv5:** contains the project for the EWARM version 5 toolchain
  - **HiTOP:** contains the project for the HiTOP toolchain
  - **RIDE**: contains the project for the RIDE toolchain
  - **RVMDK**: contains the project for the RVMDK toolchain

Figure 6. IAP driver directory structure



```
STM32F10x_IAP_AN
    FWLib
        library
            inc
            src
    IAP_AN
        binary
            template project
                EWARMv5
                HiTOP
                RIDE
                RVMDK
                main.c
                main.h
                platform_config.h
                readme.txt
                stm32f10x_conf.h
                stm32f10x_it.c
                stm32f10x_it.h
        include
        project
            EWARMv5
            HiTOP
            RIDE
            RVMDK
        source
        readme.txt
    version.txt
```

To efficiently use the IAP drive:

1. Download the IAP driver into the STM32F10xxx's internal Flash memory via the JTAG or SWD interface using the desired development toolchain.
2. Open a HyperTerminal window using the settings already defined in *Section 2.1*.
3. To run the IAP driver, keep the push-button (Key push-button on STM3210B-EVAL/STM3210E-EVAL board) pressed at Reset. The IAP main menu is then displayed on the HyperTerminal window.
4. To download an application, press **1** and use the Ymodem protocol as described in *Figure 3: IAP Driver menu when the STM32F10xxx Flash memory is not protected*.

8 Revision history

Table 2. Revision history

Date	Revision	Changes
05-Jun-2007	1	Initial release.
05-Oct-2007	2	The IAP driver can also be programmed from the SWD interface (see *Section 1.1: Principle*). Modified: – *Figure 3: IAP Driver menu when the STM32F10xxx Flash memory is not protected* – *Figure 4: IAP driver menu when the STM32F10xxx Flash memory is write-protected* – *Figure 6: IAP driver directory structure*. Flash routines modified in *Table 1: STM32F10xxx IAP implementation*. *Section 5: User program conditions* updated. *Section 7: How to use the IAP driver* modified, RIDE project added.
30-May-2008	3	Application note updated to apply to High-density devices (implemented on the STM3210E-EVAL evaluation board). Small text changes. *Figure 1: Flowchart of the IAP driver on page 4* corrected.
13-Jun-2008	4	*Figure 6: IAP driver directory structure* modified.