AN3078
Application note

STM32™ in-application programming over the I²C bus

Introduction

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

STM32F10x microcontrollers can run user-specific firmware to perform the IAP of their embedded Flash memory. This feature allows the use of any type of communication protocol for the reprogramming process (such as CAN, USART, USB, SPI, I2C, etc.).

This application note describes how to perform in-application programming using the STM32F10x's I2C peripheral. As a first step, the Aardvark tool is used as the host sending the IAP commands to the STM32F10x. As a second step, the host is another STM32F10x microcontroller that behaves as a bridge between the PC and the target (USB/I2C bridge).

The STM32F10x behaves as an I2C memory that can be programmed, erased and read via the I2C/ interface.

For further information on the STM32F10x family features, pinout, electrical characteristics, mechanical data and ordering information, please refer to the STM32F10x datasheets available from the STMicroelectronics website: *www.st.com*.

The STM32 firmware libraries are available from *www.st.com*.

The versions of the Aardvark host adapter used during the development are listed below:

- **Firmware:** Aardvark I2C/SPI Control Center v3.52 -2009.0130
- **Hardware:** Aardvark I2C/SPI v3.0

For further information on the Aardvark host adapter and its software, please refer to its documentation on *www.st.com*.

Contents

List of tables

List of figures

1 IAP overview

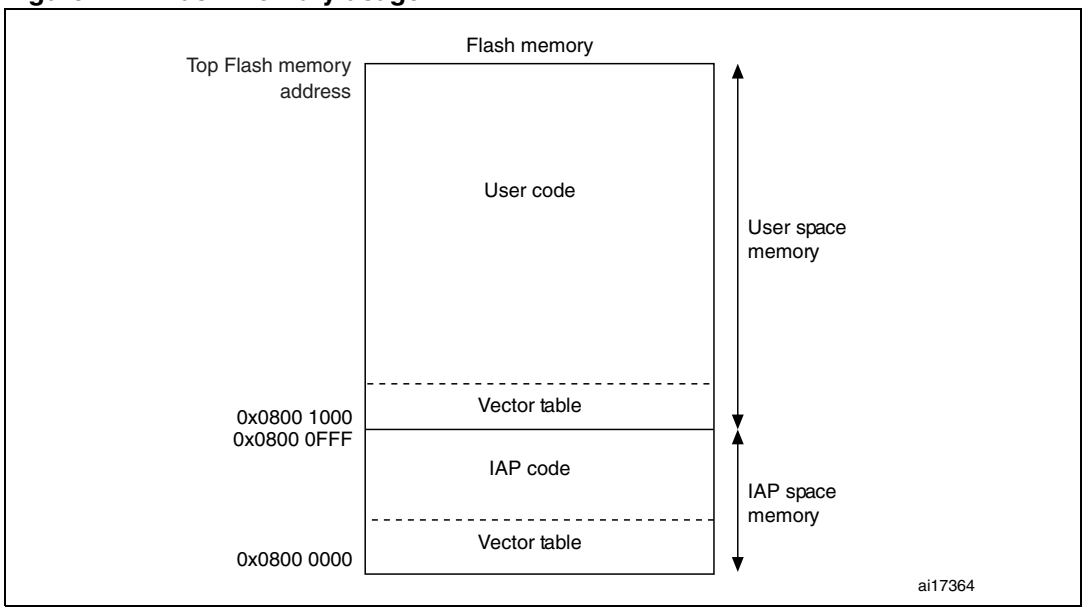
1.1 Principle

The IAP driver must be programmed from the Flash memory base address via the JTAG or SWD interface using a development toolchain. This driver uses the I2C to load a file from the host to the STM32F10x's internal Flash memory, and then executes it.

1.2 User program conditions

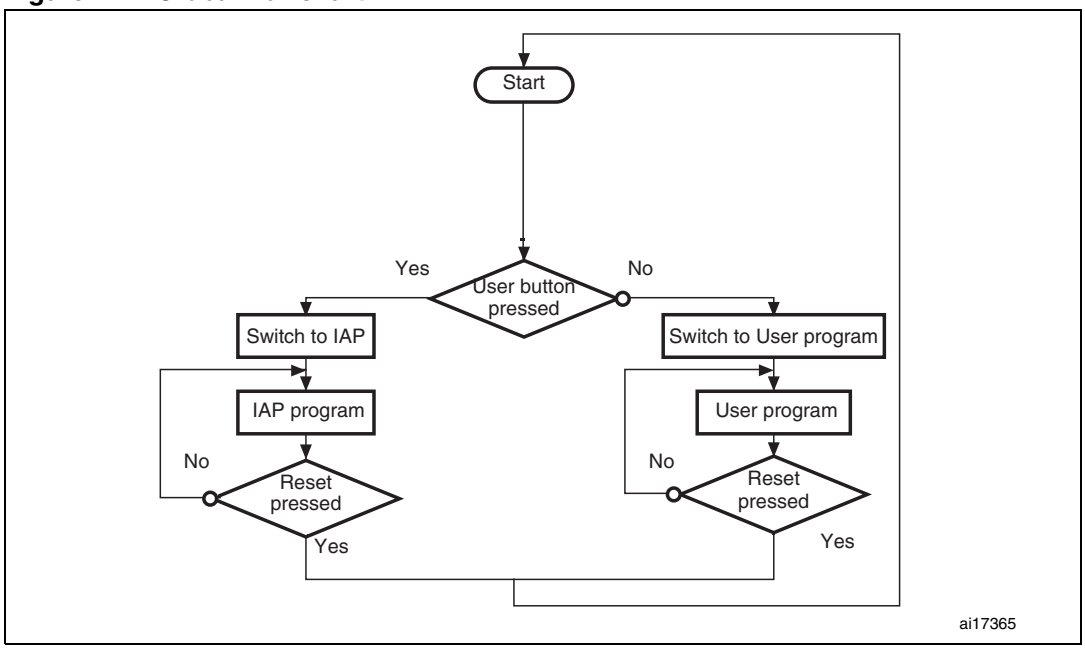
The IAP application is loaded from the Flash memory base address (0x0800 0000) to the 0x0800 0FFF address. The user application therefore has to be loaded from the 0x0800 1000 address to the top Flash memory address.

Figure 1. Flash memory usage



1.3 Global flowchart

Figure 2. Global flowchart



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

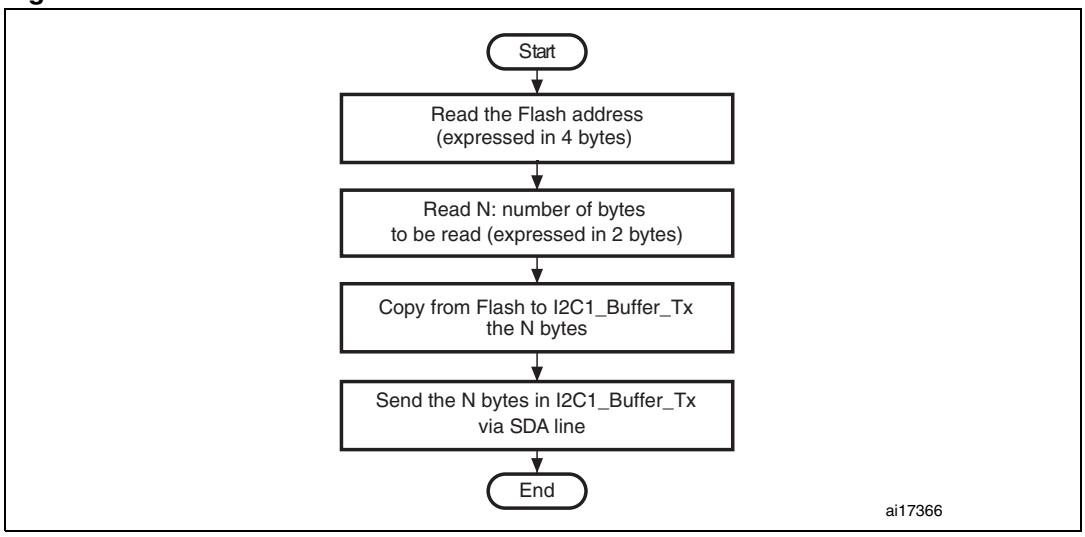
- at reset, when the push-button is pressed IAP is executed
- otherwise, a jump to the user program is performed

1.4 IAP commands

1.4.1 Read command

The read command is used to read data from the STM32's Flash memory, starting from any Flash address in the user space memory.

Figure 3. Read command flowchart

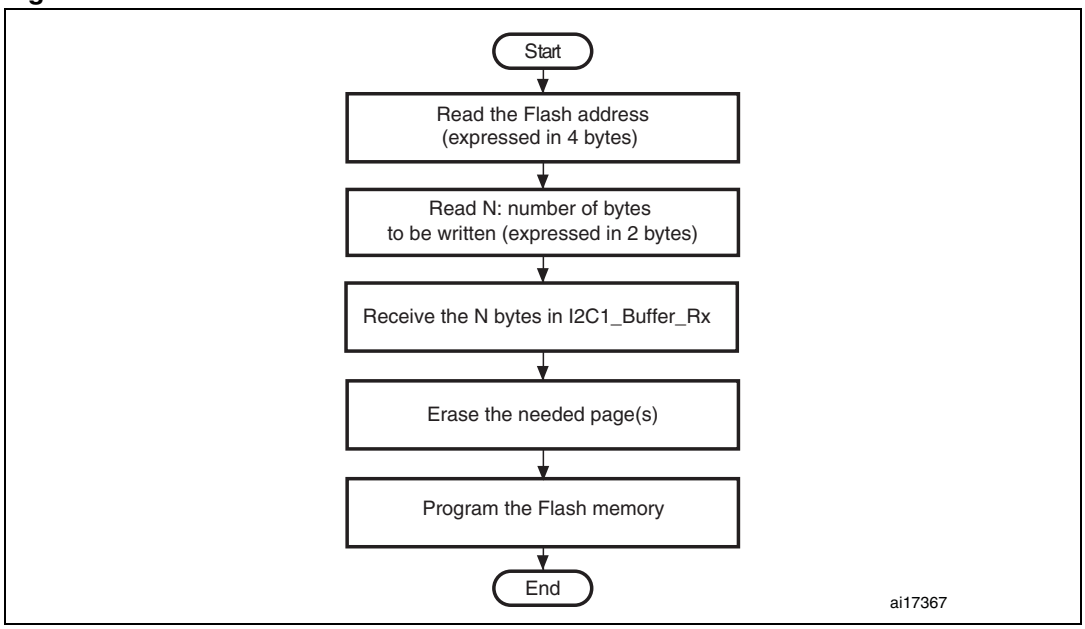


If the opcode value is **OPC_READ**, the Read command is performed. The target device considers the first four bytes sent by the host master after the opcode as the Flash memory address to be read from. The next two bytes after the Flash memory address give the size in bytes of the data to be read. The contents of the Flash memory are buffered into **I2C1_buffer_Tx** and sent by bytes via the **SDA line**.

1.4.2 Write command

The write command is used to write data into the STM32 Flash memory, starting from any Flash memory address in the user space memory.

Figure 4. Write command flowchart



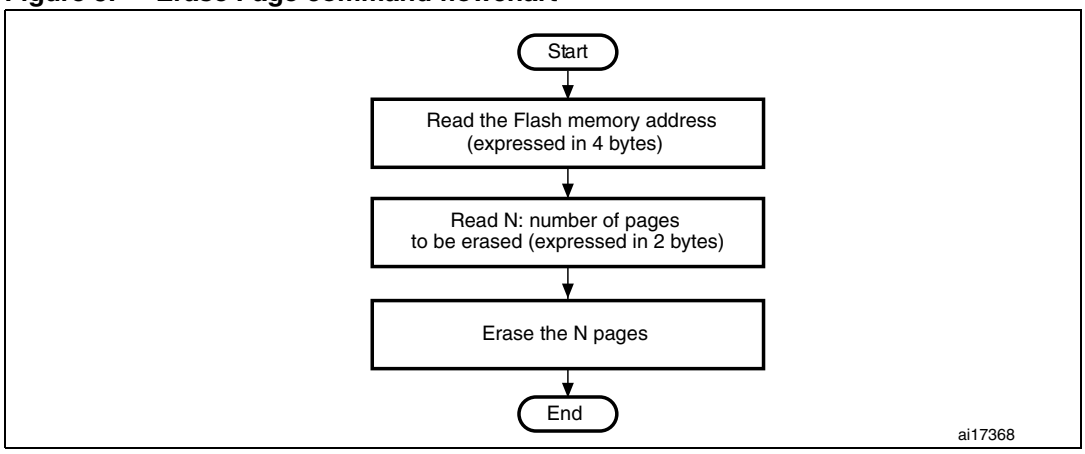
If the opcode value is **OPC_WREN**, the Write command is performed. The target device considers the first four bytes sent by host master after the opcode as the Flash memory address to be written to. The two next bytes after the Flash memory address give the size in

bytes of the data to be written. The N bytes received from the host master are buffered into **RAM_BufferRx**. The corresponding Flash memory page(s) is (are) erased before the data are written.

1.4.3 Erase page command

The Erase Page command erases pages from any Flash memory base address in the user space memory. The numbers of the pages to be erased are between **0x0001** and **0x007B** (1 to 124 pages) for medium-density devices and between **0x0001** and **0x00FE** (1 to 254 pages) for high-density devices.

Figure 5. Erase Page command flowchart

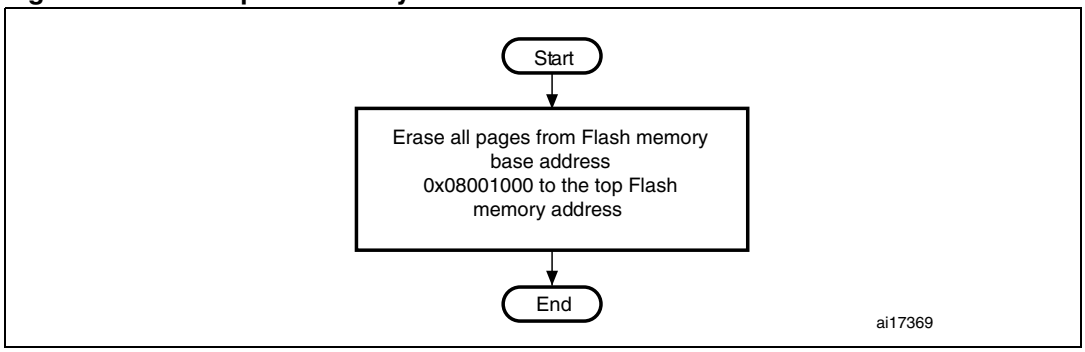


If the opcode value is **OPC_ERPG**, the Erase Page command is performed. The target device considers the first four bytes sent by host master after the opcode as the Flash memory address. The next two bytes after the Flash memory address give the numbers of the pages to be erased. Once the address and page numbers have been received, the target STM32 device erases the corresponding Flash memory pages.

1.4.4 Erase User Space memory command

The Erase User Space Memory command erases all the pages of the user space memory from Flash memory address **0x0800 1000** to the top Flash memory address.

Figure 6. User Space Memory Erase command flowchart



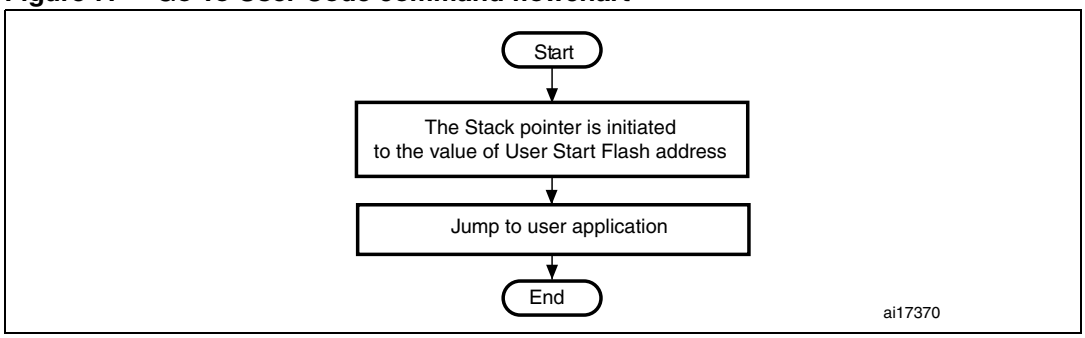
If the opcode value is **OPC_ERUSM**, the Erase User Space Memory command is performed. As its name suggests, this command erases the contents of all pages from Flash

memory base address 0x0800 1000 (the user space start address) to the top Flash memory address.

1.4.5 Go To User Code command

The Go To User Code command allows the program to jump to the user code.

Figure 7. Go To User Code command flowchart



2 STM32 IAP over the I²C peripheral

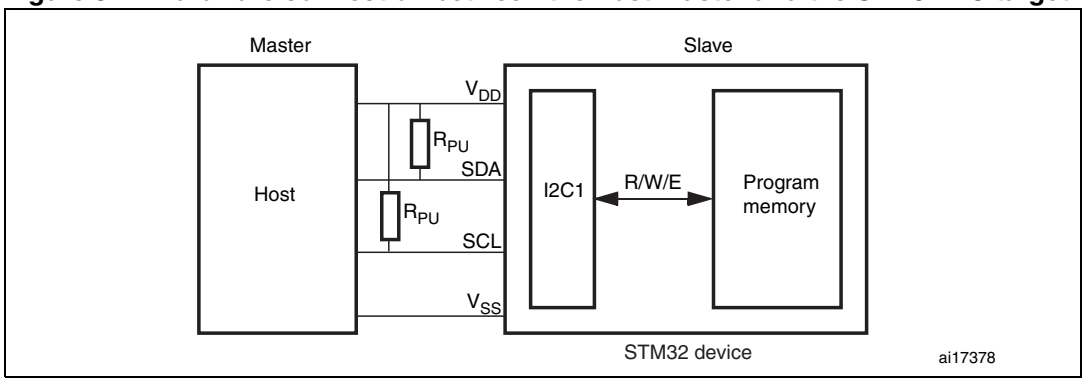
This section describes how to perform IAP using the STM32 I²C peripheral.

2.1 Hardware description

Figure 15 shows a typical hardware connection between the host master and the STM32 I²C slave target. The master and the STM32 target are connected together via the data (SDA) and clock (SCL) pins.

A 4.7 kΩ pull-up resistor has to be connected to both the SDA and SCL lines.

Figure 8. Hardware connection between the host master and the STM32 I²C target



2.2 Software description

2.2.1 I²C packet description

The different packets in an I²C command are:

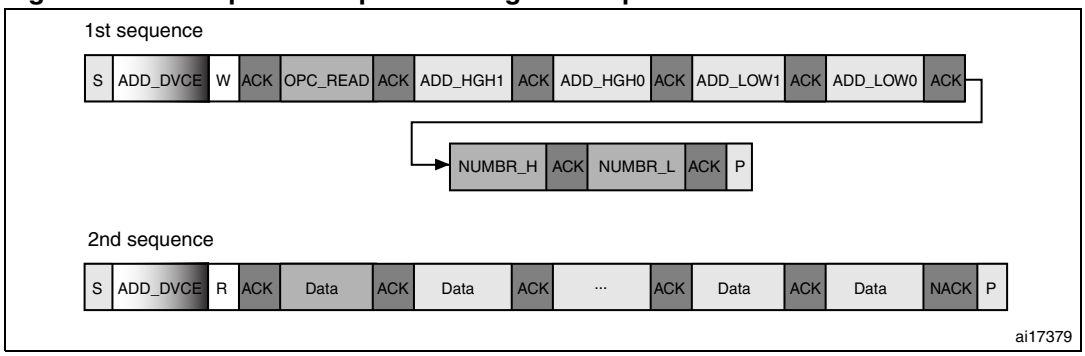
1. **ADD_DVCE:** Target address
2. **R:** READ BIT: the master receives the message from the target
3. **W:** WRITE BIT: the master will send the message to the target
4. **ACK:** Acknowledge
5. **NACK:** Not acknowledge
6. **S:** I2C Start condition
7. **P:** I2C Stop condition
8. **DUM_BYTE:** Dummy byte (0xFF)

2.2.2 IAP I²C Read command

The Read command consists of two sequences:

1. **I2C_write sequence:** the master sends the opcode, the Flash memory address and the number of bytes to be read.
2. **I2C_read sequence:** the target sends the contents of the Flash memory.

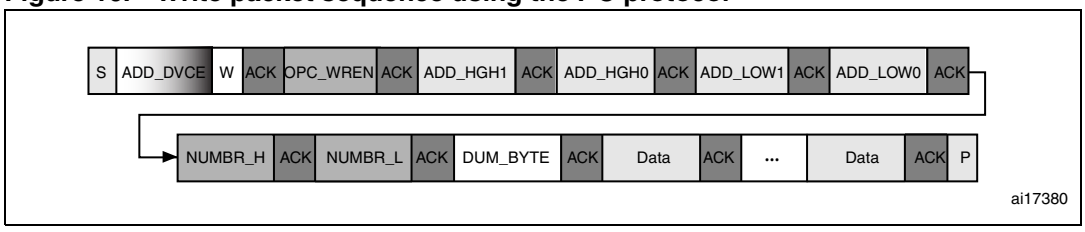
Figure 9. Read packet sequence using the I²C protocol



2.2.3 IAP I²C Write command

The master successively sends the slave address, the write command opcode, the Flash memory address, the numbers of bytes to be written, a dummy byte and the bytes to be written to the user Flash memory.

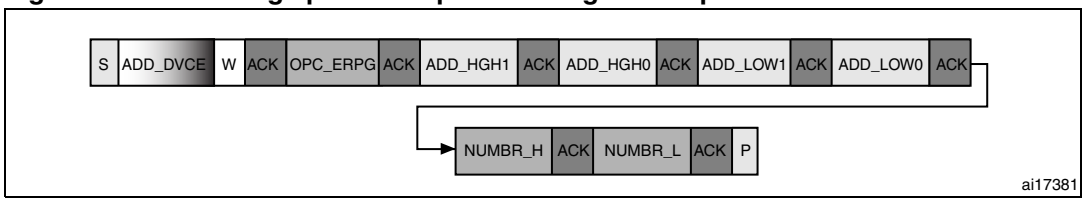
Figure 10. Write packet sequence using the I²C protocol



2.2.4 IAP I²C Erase Page command

The master successively sends the target address, the Erase Page command opcode, the Flash memory address and the numbers of the pages to be erased.

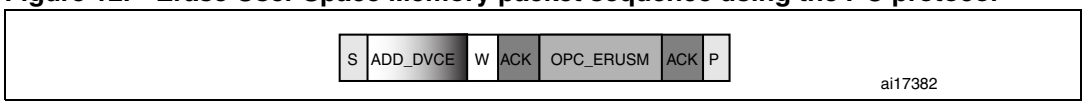
Figure 11. Erase Page packet sequence using the I²C protocol



2.2.5 IAP I²C Erase User Space Memory command

The master successively sends the target address and the Erase User Space Memory command opcode.

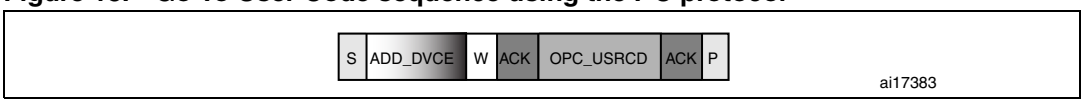
Figure 12. Erase User Space Memory packet sequence using the I²C protocol



2.2.6 IAP I²C Go To User Code command

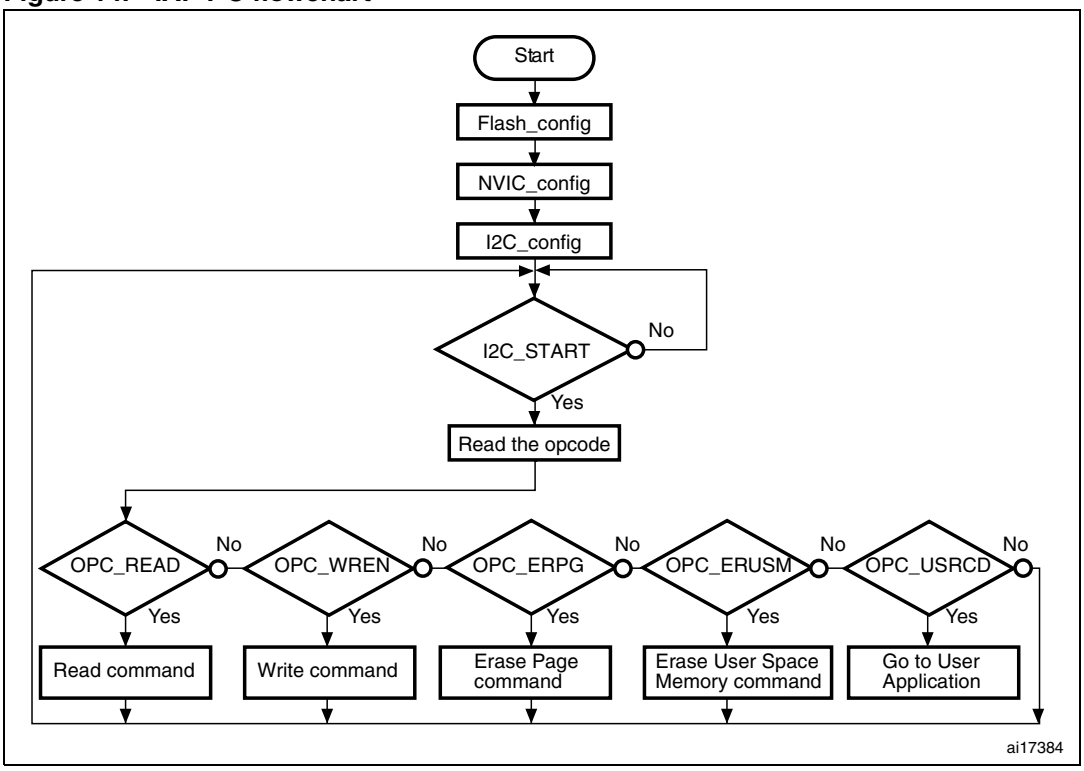
The master successively sends the target address and the opcode of the Go To User Code command.

Figure 13. Go To User Code sequence using the I²C protocol



2.2.7 IAP I²C flowchart

Figure 14. IAP I²C flowchart



3 Sending IAP commands to a target

3.1 Sending IAP I²C commands using the Aardvark I²C host adapter

3.1.1 Aardvark I²C interface mode

To use the I²C interface, select the "**I2C + GPIO**" or the "**I2C + SPI**" operational mode.

The following steps explain how to configure the Aardvark I²C interface to match the previous configuration.

1. Select the 'Master' tabsheet.
2. Set the bitrate to 100 kHz using the 'Bitrate' field and the 'set' button.
3. Set the slave address to **0x18** in the 'Slave Addr' field.
4. Uncheck the '10-Bit Addr' option (10-bit addressing mode) because the firmware uses the 7-bit addressing mode only.
5. Uncheck the 'No Stop' option.

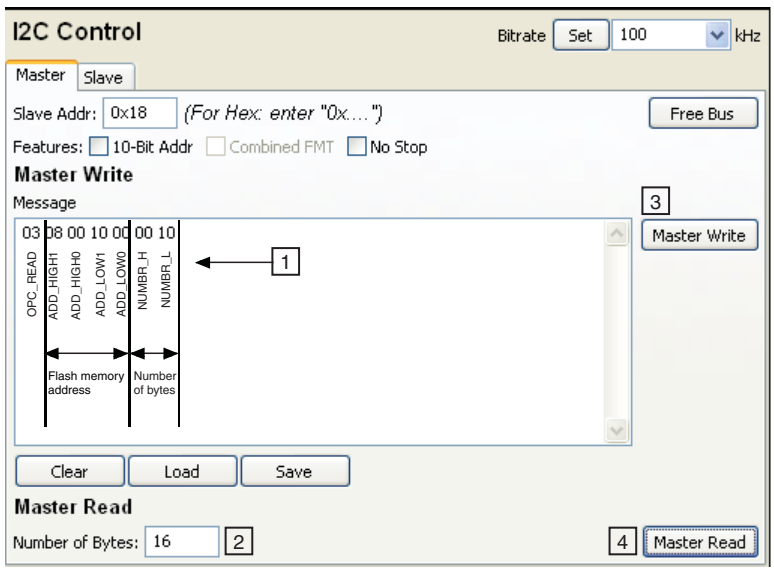
Figure 15. Configuration of Aardvark I²C host adapter in interface modet

Read command

The following steps explain how to read the contents of the user Flash memory:

- Step 1: Enter the byte sequence of the Read command in a Master Write message. Refer to *Section 2.2.2: IAP I2C Read command* for a description of the sequence.
- Step 2: Enter the number of bytes to be read into the "Number of Bytes" field.
- Step 3: Press the 'Master Write' button.
- Step 4: Press the 'Master Read' button.

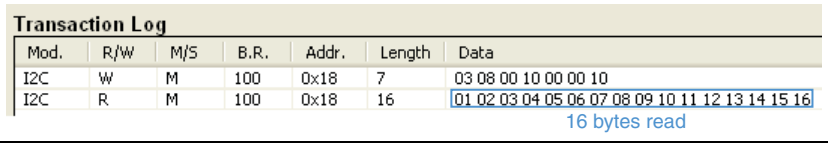
Figure 16. Read command using the Aardvark I²C interface



In this example, the Read command reads 16 bytes from the Flash memory address **0x0800 1000**.

The figure below shows the result of the Read command transaction.

Figure 17. Read command I²C transaction

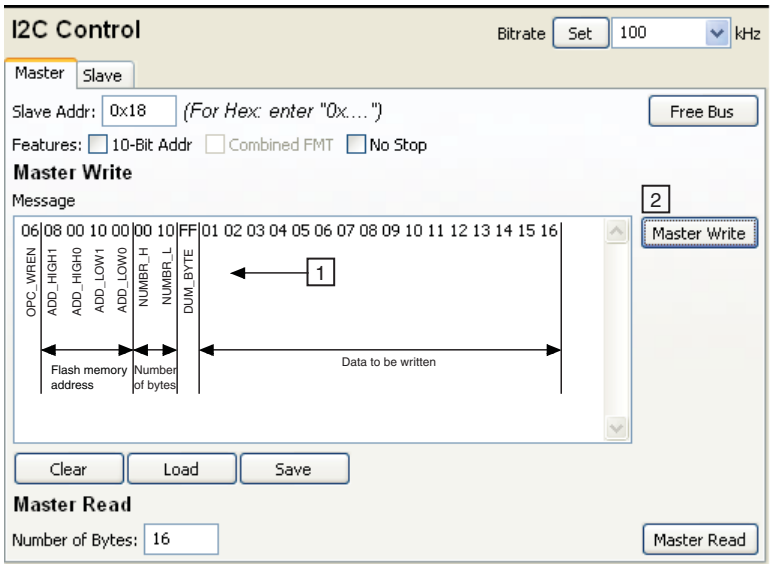


Write command

The following steps explain how to write to the user Flash memory:

- Step 1: Enter the byte sequence of the Write command. Refer to *Section 2.2.3: IAP I2C Write command* for a description of the sequence.
- Step 2: Press the 'Master Write' button.

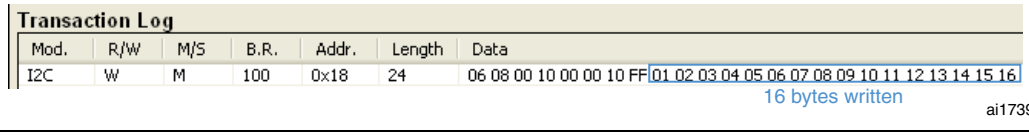
Figure 18. Write command using the Aardvark I²C interface



In this example, the Write command writes 16 bytes to the Flash memory from address **0x0800 1000**.

Figure 19 gives the result of the Write command transaction.

Figure 19. Write command I²C transaction



The other IAP commands (Erase page, Erase Full User Space Memory and Go To User Application) are issued in the same way.

3.1.2 Aardvark I²C batch mode

The Aardvark I²C host adapter is configured using the batch mode as shown in the figure below.

Figure 20. Aardvark I²C configuration using the Batch mode

```
<aardvark>
<!-- ++++++++++++++++++++++++++CONFIGURATION++++++++++++++++++++++++ -->
  <configure i2c="1" spi="0" gpio="1" tpower="0" pullups="1"/>
  <i2c_bitrate khz="100"/>


</aardvark>
```

This configuration has to be inserted into the <head> of all script I²C commands.

Read command

The figure below shows an example of a Read command, where 4096 bytes are read from Flash memory address **0x0800 1000**.

Figure 21. IAP I²C Read command in batch script

```
<aardvark>
<!-- ++++++++++++++++++++++++++CONFIGURATION++++++++++++++++++++++++ -->
    <configure i2c="1" spi="0" gpio="1" tpower="0" pullups="1"/>
    <i2c_bitrate khz="100"/>
<!-- ++++++++++++++++++++++++++READ COMMAND ++++++++++++++++++++++++ -->
<i2c_write addr="0x18" count="7"  radix="16">03 08 00 10 00  10 00 </i2c_write>
<i2c_read addr="0x18" count="4096"  radix="16"/>

</aardvark>
```

Write command

The figure below shows an example of a Write command, where 16 bytes are written to the Flash memory from address **0x0800 1000**.

Figure 22. IAP I²C Write command in batch script

```
<aardvark>
<!-- ++++++++++++++++++++++++++CONFIGURATION++++++++++++++++++++++++ -->
    <configure i2c="1" spi="0" gpio="1" tpower="0" pullups="1"/>
    <i2c_bitrate khz="100"/>
<!-- ++++++++++++++++++++++++++WRITE COMMAND++++++++++++++++++++++++ -->
    <i2c_write addr="0x18" count="24"  radix="16">
    06                          <!-- opcode of WRITE COMMAND -->
    08 00 10 00                 <!-- add of flash memory  -->
    00 10                       <!-- number of bytes  to be written -->
    FF                          <!-- dummy byte -->
    01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16  <!-- write 16 bytes -->
    </i2c_write>
</aardvark>
```

The batch scripts of the other IAP commands (Erase Page, Erase Full User Space Memory and Go To User Application) are written in the same way.

3.2 Sending IAP commands using another STM32 device as a bridge

3.2.1 Principle

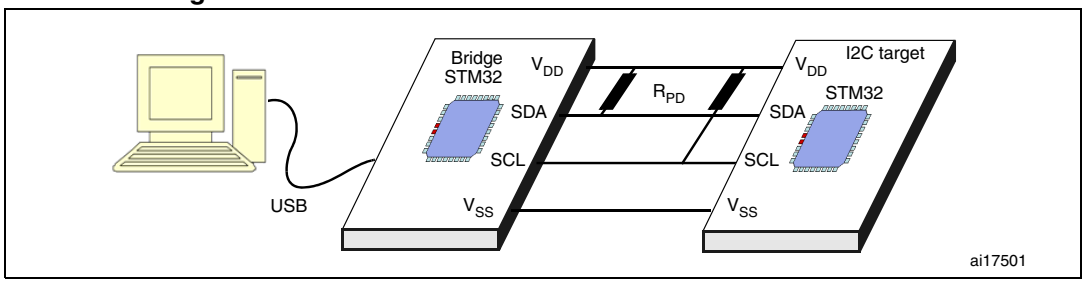
The device firmware upgrade (DFU) project is extended to support the I²C module. The DFU process is similar to any IAP process, so, in the case of this project, instead of upgrading the internal Flash memory of the bridge device, the commands are transferred via I2C peripheral to upgrade the target device.

For further information on device firmware upgrade (DFU), please refer to the dedicated user, UM0424, available from www.st.com.

3.2.2 Hardware description

Figure 23 shows the hardware connection between the STM32 bridge device and the STM32 target. The bridge is connected to the PC via USB.

Figure 23. Hardware connection between the PC, the STM32 bridge and the I²C target



3.2.3 Firmware description

The purpose of this section is not to provide detailed information on device firmware upgrade (for that purpose refer to user manual UM0424), but to highlight the extension and changes made to realize the USB-I²C bridge.

1. DFU mode entry mechanism:

   The bridge STM32F10x DFU mode is entered directly after an MCU reset.

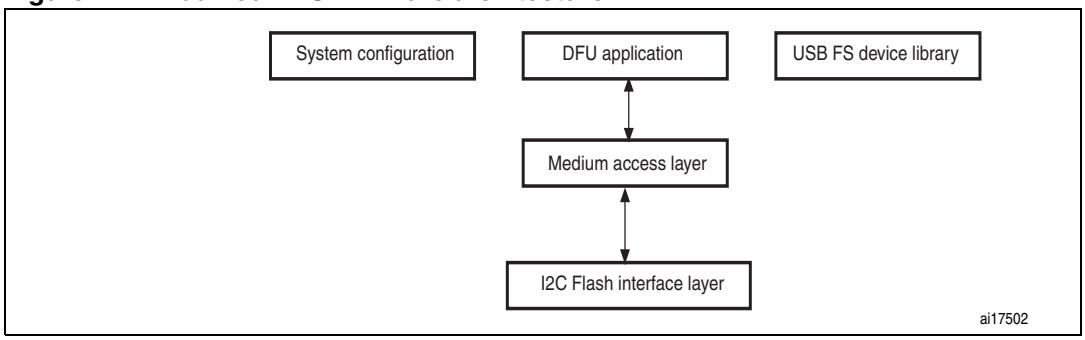
2. DFU firmware architecture:

   The Flash interface layers (internal, SPI and NOR) are omitted and a new interface layers is added: **I²C Flash interface layer**.

The address of the I²C Flash interface layer is **0x000 1000**.

Figure 24 shows the new DFU firmware architecture adopted for the USB-I²C bridge.

Figure 24. Modified DFU firmware architecture



4 IAP timings

This section gives an idea of the time required to program a file into the STM32's internal Flash memory.

Table 1. IAP timings

Settings	IAP via I^2C
Communication protocol settings	400 kHz, 7-bit addressing mode
Download time(1) (508 Kbytes)	358 s
Download time(1) (124 Kbytes)	105 s
Speed (bits/s)	11980

1. Download time = erase time + program time + verify time.

5 Revision history

Table 2. Document revision history

Date	Revision	Changes
12-Jul-2010	1	Initial release.