UM0892
User manual

STM32 ST-LINK Utility software description

Introduction

The STM32 ST-LINK Utility software facilitates fast in-system programming of the STM32 microcontroller families in development environments via the tools, ST-LINK and ST-LINK/V2.

This is the STM32 ST-LINK Utility user manual, describing its software functions. When working with the STM32 ST-LINK Utility, you are encouraged to download the *ST-LINK in-circuit debugger/programmer user manual (UM0627)* or the *ST-LINK/V2 in-circuit debugger/programmer user manual (UM1075)* for the STM8 and STM32 which provide more information about the ST-LINK tools.

Table 1. Applicable tools

Type	Part number
Development tools	ST-LINK, ST-LINK/V2

Contents

1 Getting started 3

1.1 System requirements 3

1.2 Hardware requirements 3

1.3 Installing the STM32 ST-LINK Utility 3

1.4 Uninstalling the STM32 ST-LINK Utility 3

2 STM32 ST-LINK Utility user interface 4

2.1 Main window 4

2.2 Menu bar 5

2.2.1 File menu 5

2.2.2 View menu 5

2.2.3 Target menu 6

2.2.4 ST-LINK menu 7

2.2.5 Help menu 8

3 STM32 ST-LINK Utility features 9

3.1 Device information 9

3.2 Memory display and modification 9

3.3 Flash memory erase 11

3.4 Device programming 12

3.5 Option bytes configuration 13

3.6 MCU core functions 16

3.7 Automatic mode functions 17

4 STM32 ST-LINK Utility command line interface (CLI) 18

4.1 Command line usage 18

4.1.1 Connection and memory manipulation commands 18

4.1.2 Core commands 19

4.1.3 Flash commands 20

4.1.4 Option byte commands 21

4.1.5 ST-LINK_CLI return codes 25

5 Revision history 26

1 Getting started

This section describes the requirements and procedures to install the STM32 ST-LINK Utility software.

1.1 System requirements

The STM32 ST-LINK Utility PC configuration requires as a minimum:

- PC with USB port and Intel® Pentium® processor running a 32-bit version of one of the following Microsoft® operating systems:
  - Windows® XP
  - Windows 7
- 32 Mbytes of RAM
- 10 Mbytes of hard disk space available

1.2 Hardware requirements

The STM32 ST-LINK Utility is designed to work with:

- STM32 F0, STM32 F1, STM32 F2, STM32 F3, STM32 F4, STM32 L1, and STM32W series
- ST-LINK or ST-LINK/V2 or ST-LINK/V2-ISOL in-circuit debugger/programmer probe

Note: In this document, ST-LINK/V2 refer to ST-LINK/V2 and ST-LINK/V2-ISOL which are functionally equivalent.

1.3 Installing the STM32 ST-LINK Utility

Follow these steps and the on-screen instructions to install the STM32 ST-LINK Utility.

1. Download the compressed STM32 ST-LINK Utility software from the ST website.
2. Extract the contents of the *.zip* file into a temporary directory.
3. Double-click the extracted executable, *setup.exe*, to initiate the installation, and follow the on-screen prompts to install the STM32 ST-LINK Utility in the development environment. Documentation for the utility is located in the subdirectory *\Docs* where the STM32 ST-LINK Utility is installed.

1.4 Uninstalling the STM32 ST-LINK Utility

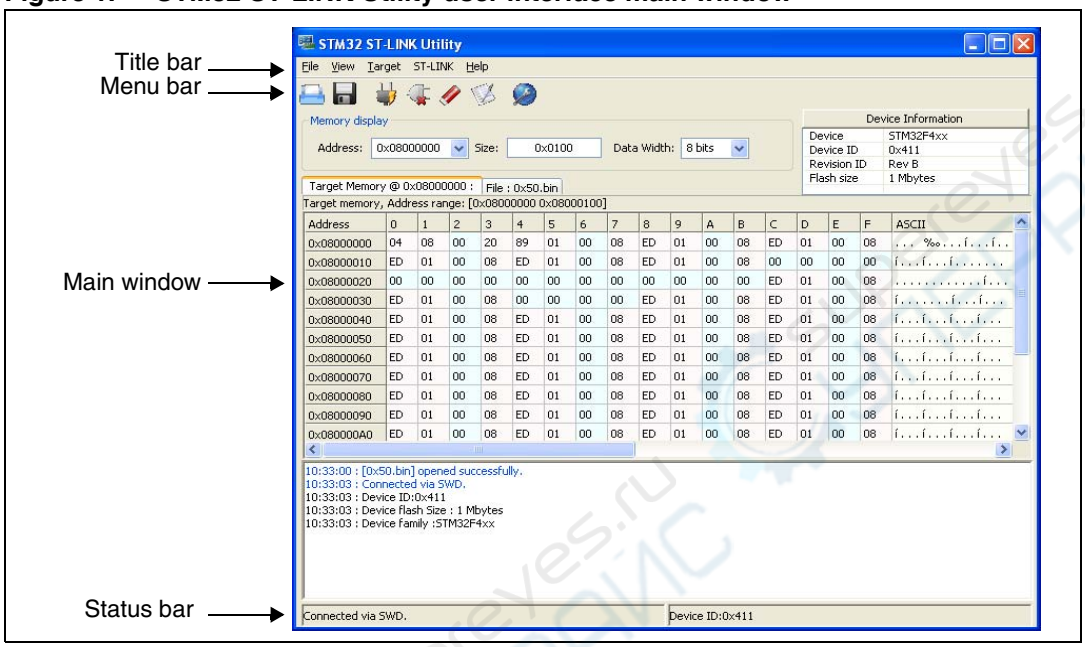
Follow these steps to uninstall the STM32 ST-LINK Utility.

1. Select **Start | Settings | Control Panel**.
2. Double-click on **Add or Remove Programs**.
3. Select **STM32 ST-LINK Utility**.
4. Click on the **Remove** button.

2 STM32 ST-LINK Utility user interface

2.1 Main window

Figure 1. STM32 ST-LINK Utility user interface main window



The main window is composed of three zones and three bars as illustrated in *Figure 1*:

- Memory display zone
- Device information zone
- Memory contents zone
- Title bar: The name of the current menu
- Menu bar: Use the menu bar to access the following STM32 ST-LINK Utility functions:
  - **File** menu
  - **View** menu
  - **Target** menu
  - **Help** menu

  (These menus are described in more detail in *Section 2.2*)
- Status window: The status bar displays:
  - Connection status and debug interface
  - Device ID

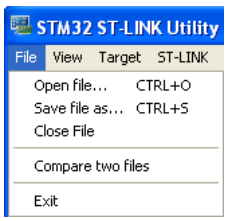
The STM32 ST-LINK Utility user interface also provides additional forms and descriptive pop-up error messages.

2.2 Menu bar

The Menu bar allows users to execute the STM32 ST-LINK Utility software features.

2.2.1 File menu

Figure 2. File menu



Open file... Opens a binary, Intel Hex or Motorola S-record.

Save file as… Saves the content of the memory panel into a binary, Intel Hex or Motorola S-record.

Close File Closes the loaded file.

Compare two files Compares two binary, hex, or srec files. The difference is colored in red in the file panel. If a file contains a section with an address range that is unavailable in the other file, this section will be colored in violet.

Exit Closes the STM32 ST-LINK Utility program.

2.2.2 View menu

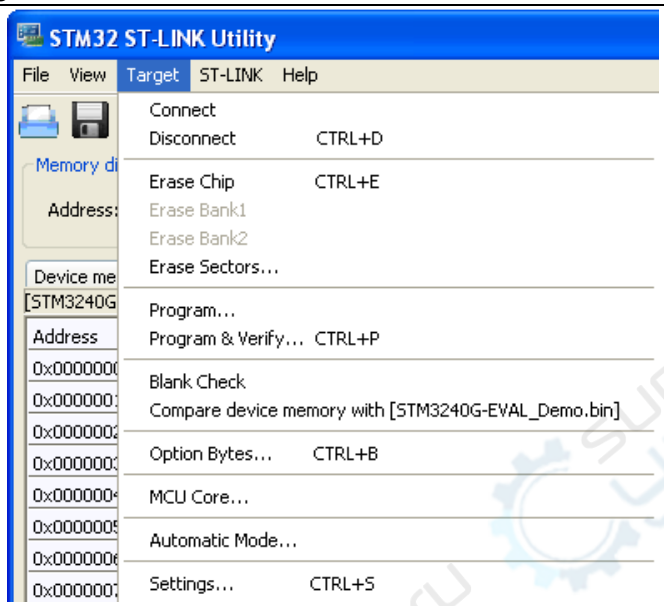
Figure 3. View menu

Binary file Displays the content of the loaded binary file.

Device memory Displays the content of the device memory.

2.2.3 Target menu

Figure 4. Target menu



Connect Connects to the target device and displays the Device Type, Device ID and Flash memory size in the device information zone.

Disconnect Disconnects from the target device.

Erase Chip Performs a Flash memory mass erase and then displays the Flash memory content in the memory panel.

Erase Bank1 Erases bank 1 of the Flash memory. This menu is enabled only when connected to an XL-density device.

Erase Bank2 Erases bank 2 of the Flash memory. This menu is enabled only when connected to an XL-density device.

Erase Sectors... Selects sector(s) to erase using the erase sectors dialog window (see *Section 3.3: Flash memory erase* for more details).

Program... Loads a binary, Intel Hex or Motorola S-record file into the device memory (Flash or RAM). To do this, select a binary, Intel Hex or Motorola S-record file, enter the start address (where to put the file in the device) in the program dialog window and then click on program button (see *Section 3.4: Device programming*).

Program & Verify... Loads a binary, Intel Hex or Motorola S-record file into the device memory (Flash or RAM) then performs a verification of the programmed data.

Blank Check Verifies that the STM32 Flash memory is blank. If the Flash memory is not blank, the first address with data is highlighted in a prompt message.

Compare device memory with file Compares the MCU device memory content with a binary, hex, or srec file. The difference is colored in red in the file panel.

Option Bytes… Opens the Option Bytes dialog window (See *Section 3.5: Option bytes configuration* for more details).

MCU Core… Opens the MCU Core dialog window (See *Section 3.6: MCU core functions* for more details).

Automatic Mode... Opens the Automatic mode dialog window (See *Section 3.7: Automatic mode functions* for more details).

Settings... Opens the Settings dialog box to select the debug interface (JTAG or SWD).
The “connect under reset” option allows you to connect to the target before executing any instruction.
This is useful in many cases like when the target contains a code that disables the JTAG/SWD pins.

When connecting to the target via an ST-LINK/V2, the “Supply voltage” combo box displays the target voltage.
When connecting to an STM32 F2 or STM32 F4 device using an ST-LINK, the “Supply voltage” combo allows you to select the supply voltage of your target to be able to correctly program the Flash memory.

Note: The “connect under reset” option is available only with ST-LINK/V2 and in SWD mode.

For JTAG mode, “connect under reset” is available since ST-LINK/V2 firmware Version V2J15Sx.

The RESET pin of the JTAG connector (pin 15) should be connected to the device reset pin.

2.2.4 ST-LINK menu

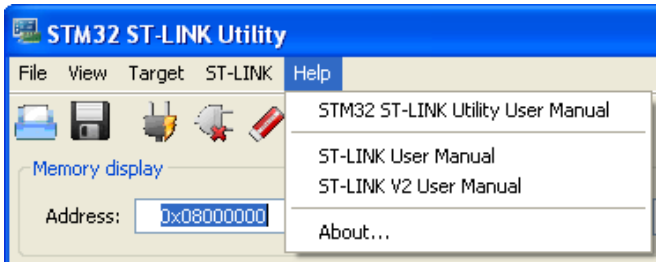
Figure 5. ST-LINK menu

Firmware update Displays the version of ST-LINK and ST-LINK/V2 firmware and updates it to the last version:

ST-LINK: V1J12S3
ST-LINK/V2: V2J16S3

2.2.5 Help menu

Figure 6. Help menu



STM32 ST-LINK Utility User Manual Opens the *STM32 ST-LINK Utility User Manual*.

ST-LINK User Manual Opens the *ST-LINK User Manual*.

ST-LINK/V2 User Manual Opens the *ST-LINK/V2 User Manual*.

About... Displays STM32 ST-LINK Utility software version and copyright information.

3 STM32 ST-LINK Utility features

This section provides a detailed description of how to use STM32 ST-LINK Utility features:

- *Device information*
- *Memory display and modification*
- *Flash memory erase*
- *Device programming*
- *Option bytes configuration*
- *MCU core functions*
- *Automatic mode functions*

3.1 Device information

The *Device information* zone displays information as shown in *Figure 7*.

Figure 7. Device information zone in the main user interface

Device Information	
Device	STM32F10xxx High-density device
Device ID	0x414
Revision ID	Rev Z
Flash size	512 Kbyte

Device: Family of the connected STM32 device. Each device type includes many devices with different characteristics such as Flash memory size, RAM size and peripherals.

Device ID: MCU device ID code located in the external PPB memory map

Revision ID: The revision ID of the connected MCU device

Flash size: Size of the on-chip Flash memory

3.2 Memory display and modification

In addition to the **Device information** zone, the main window contains 2 other zones:

- **Memory display**
- **Memory data**

Memory display: This zone contains three edit boxes:

Address: Memory start address from which you want to read.

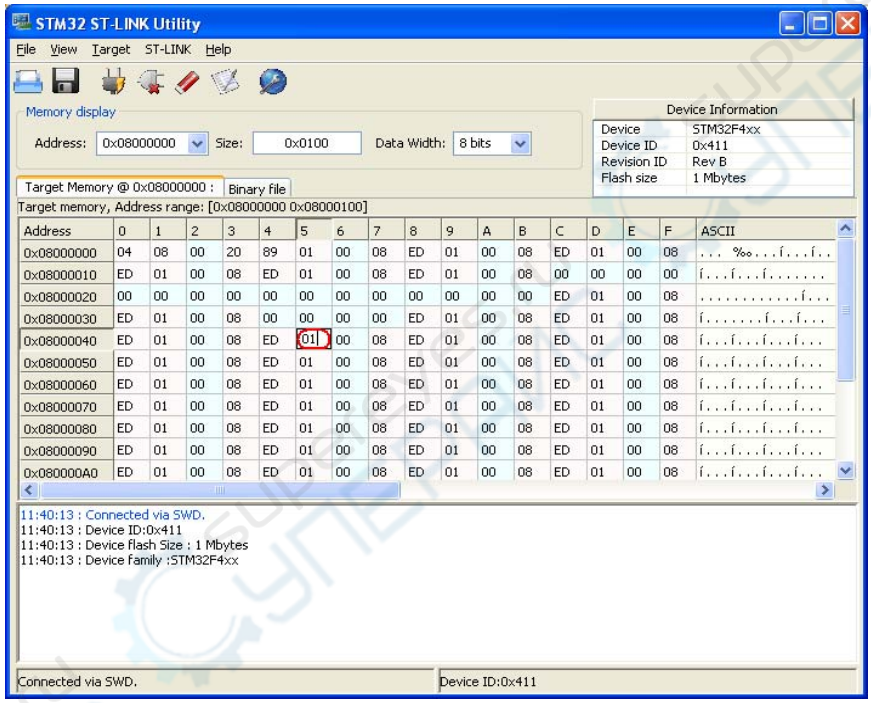
Size: Amount of data to read.

Data width: Width of the displayed data (8-bit, 16-bit or 32-bit).

Memory data: This zone displays the data read from a file or the memory content of a connected device. You can modify the content of the file before downloading.

- To use this zone to display the content of a binary, Intel Hex or Motorola S-record file, go to **File | Open file...**
- To use this zone to read and display memory content of a connected device, enter the memory start *Address*, data *Size* and the *Data Width* in the *Memory display* zone and then press **Enter**.
- After reading data, you can also modify each value merely by double-clicking on the concerned cell as illustrated by *Figure 8*. You can also save the device memory content into a binary, Intel Hex or Motorola S-record file using the menu **File | Save file as...**

Figure 8. STM32 ST-LINK Utility user interface



Note: When the Memory data zone displays device memory contents, any modification is automatically applied to the chip. You can modify user Flash memory, RAM memory and peripherals registers.

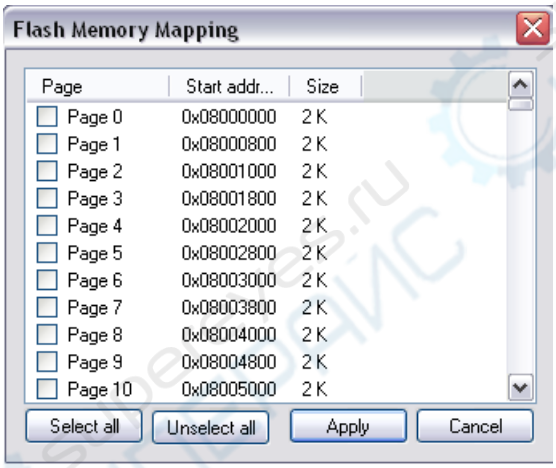
For the STM32 F2 and STM32 F4 devices, you can modify the OTP area directly from the memory data zone.

3.3 Flash memory erase

There are two types of Flash memory erase:

- **Flash mass erase**: Erase all the memory Flash memory sectors of the connected device. This is done by clicking on the menu **Target | Erase Chip**.
- **Flash Sector Erase**: Erase the selected sector(s) of the Flash memory. To select sector(s), go to **Target | Erase Sectors...** which then displays the **Flash Memory Mapping** dialog box where you select the sector(s) to erase as shown in *Figure 9*.
  - **Select all** button selects all the Flash memory pages.
  - **Deselect all** button deselects all selected page.
  - **Cancel** button discards the erase operation even if some pages are selected.
  - **Apply** button erases all the selected pages.

Figure 9. Flash Memory Mapping dialog box



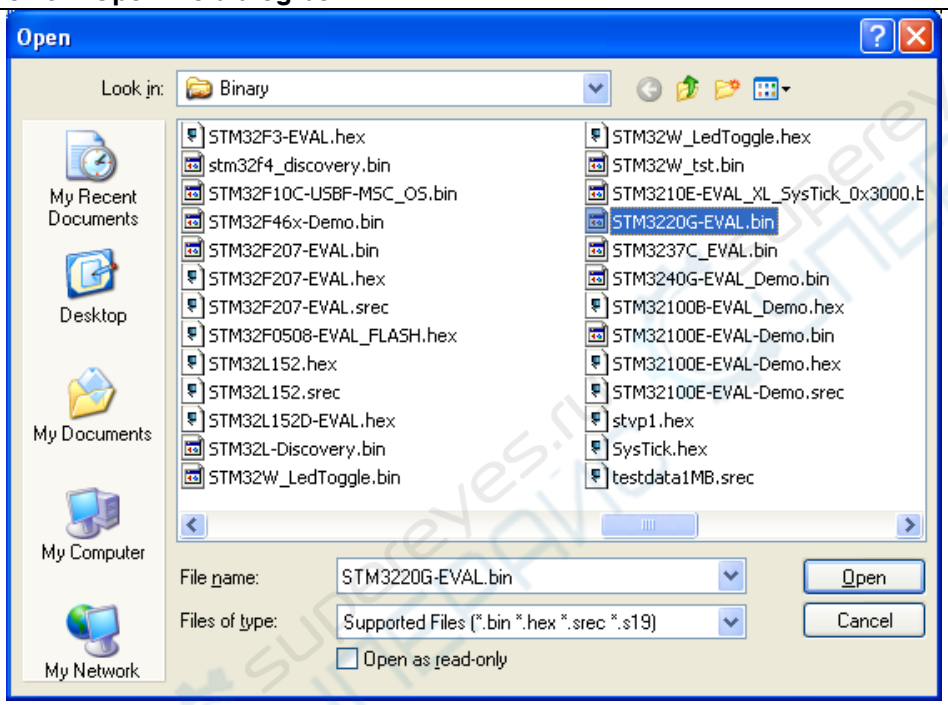
*Note: To erase the Flash data memory sector of the ultralow power STM32 L1 devices, select the data memory box at the end of the list and click **Apply**.*

3.4 Device programming

The STM32 ST-LINK Utility can download binary, Hex, or srec files into Flash or RAM memory. To do this, follow these steps:

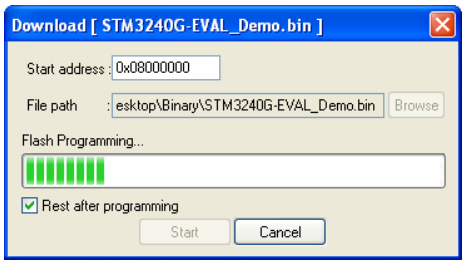
1. Click on **Target | Program...** (or **Target | Program & Verify...** if you want to verify the written data) to open the **Open** file dialog box as shown in *Figure 10*.
   If a binary file is already opened, go to step *3.*

Figure 10. Open file dialog box



2. Select a binary, Intel Hex or Motorola S-record file and click on the **Open** button.
3. Specify the address from which to start programming as shown in *Figure 11*, it may be a Flash or RAM address.

Figure 11. Device programming dialog box (programming)

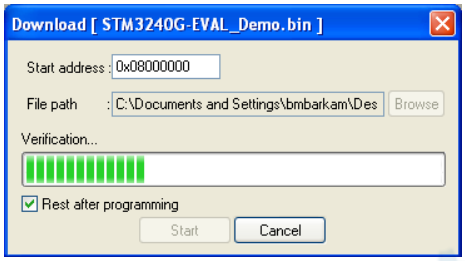


4. Finally, click on the **Start** button to start programming. If you selected **Target | Program & Verify...** in the first step, a check is done at the end of the programming operation (see *Figure 12*). If the “Reset after programming” box is checked, an MCU reset will be issued.

*Note: The STM32 F2 and STM32 F4 series supports different programming modes depending on the MCU supply voltage. When using ST-LINK, the MCU supply voltage should be specified in the **Target | Settings** Menu to be able to program the device with the correct mode. When using ST-LINK/V2, the supply voltage is detected automatically.*

If the device is read-protected, the protection will be disabled. If some Flash memory pages are write-protected, the protection will be disabled during programming and then recovered.

Figure 12. Device programming dialog box (verification)



3.5 Option bytes configuration

The STM32 ST-LINK Utility can configure all the option bytes via the **Option Bytes** dialog box shown in *Figure 13* which is accessed by **Target | Option Bytes...**.

The **Option Bytes** dialog box contains the following sections:

- **Read Out Protection**: Modifies the read protection state of the Flash memory.

  For STM32 F0, STM32 F2, STM32 F3, STM32 F4 and STM32 L1 devices, read protection levels are available:
  - Level 0: no read protection
  - Level 1: memory read protection enabled
  - Level 2: memory read protection enabled and all debug features disabled.

  For the other devices, the read protection can only be enabled or disabled.
- **BOR Level**: Brownout reset level. This list contains the supply level threshold that activates/releases the brownout reset. This option is enabled only when connected to the STM32 L1, STM32 F2 and STM32 F4 device.

  For ultralow power devices, 5 programmable VBOR thresholds can be selected:
  - BOR LEVEL 1: Reset threshold level for 1.69 to 1.8 V voltage range
  - BOR LEVEL 2: Reset threshold level for 1.94 to 2.1 V voltage range
  - BOR LEVEL 3: Reset threshold level for 2.3 to 2.49 V voltage range
  - BOR LEVEL 4: Reset threshold level for 2.54 to 2.74 V voltage range
  - BOR LEVEL 5: Reset threshold level for 2.77 to 3.0 V voltage range

  For STM32 F2 and STM32 F4 devices, 4 programmable VBOR thresholds can be selected:
  - BOR LEVEL 3: Supply voltage ranges from 2.70 to 3.60 V
  - BOR LEVEL 2: Supply voltage ranges from 2.40 to 2.70 V
  - BOR LEVEL 1: Supply voltage ranges from 2.10 to 2.40 V
  - BOR off: Supply voltage ranges from 1.62 to 2.10 V

- **User Configuration option byte:**
  - WDG_SW: If checked, the watchdog is enabled by software. Otherwise, it is automatically enabled at power-on.
  - nRST_STOP: If not checked, a reset is generated when entering Standby mode (1.8V domain powered-off). If checked, no reset is generated when entering Standby mode.
  - nRST_STDBY: if not checked, reset is generated when entering Stop mode (all clocks are stopped). If checked, no reset is generated when entering Stop mode.
  - BFB2: If not checked, and if the boot pins are set to make the device boot from the user Flash at startup, the device boots from Flash memory bank 2; otherwise, it boots from Flash memory bank 1. This option is enabled only when connected to a device containing two Flash banks.
  - nBoot1: Together with the BOOT0 pin, selects the Boot mode:

    nB00T1 checked/unchecked and BOOT0=0 => Boot from Main Flash memory.

    nB00T1 checked and BOOT0 =1 => Boot from System memory.

    nB00T1 unchecked and BOOT0 =1 => Boot from Embedded SRAM."

    nBoot1 is available only on STM32 F0 and STM32 F3.
  - VDDA_Monitor: Selects the analog monitoring on VDDA Power source:

    If checked, VDDA power supply supervisor is enabled; otherwise, it is disabled.

    VDDA is available only on STM32 F0 and STM32 F3.
  - nSRAM_Parity: This bit allows the user to enable the SRAM hardware parity check.

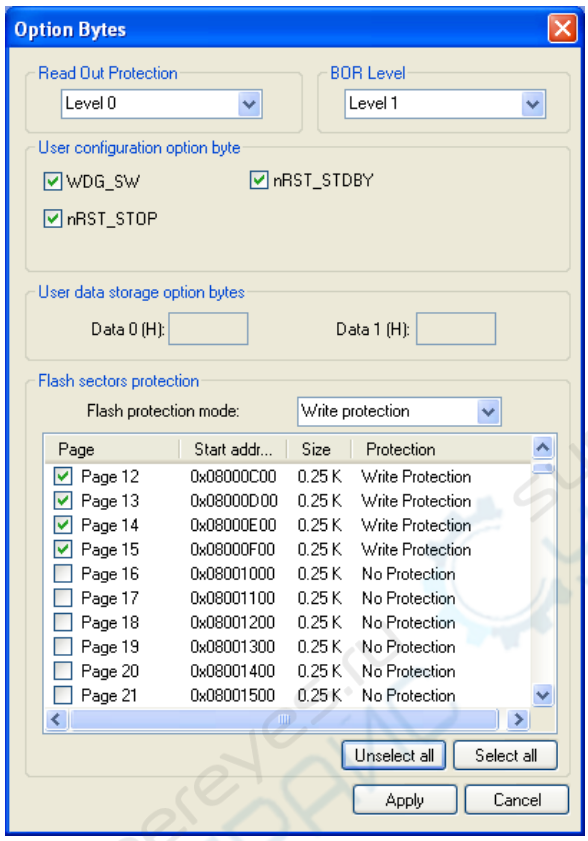
    If checked, SRAM parity check is disabled; otherwise, it is enabled.

    nSRAM_Parity is available only on STM32 F0 and STM32 F3.
  - SDADC12_VDD_Monitor:

    If checked, SDADC12_VDD power supply supervisor is enabled; otherwise, it is disabled.

    SDADC12_VDD_Monitor is available only on STM32F37x.
- **User data storage option bytes:** Contains two bytes for user storage. These two option bytes are not available in the STM32 F0, STM32 F2, STM32 F3,STM32 F4 and STM32 L1 devices.
- **Flash Sectors Protection:** Depending on the connected device, Flash sectors are grouped by a defined number of pages. You can modify the write protection of each Flash sector here.
- For devices supporting PCRop feature, it is possible to enable/disable the Read protection of each sector. The “Flash protection mode” allows you to choose between read or write protection.

Figure 13. Option Bytes dialog box



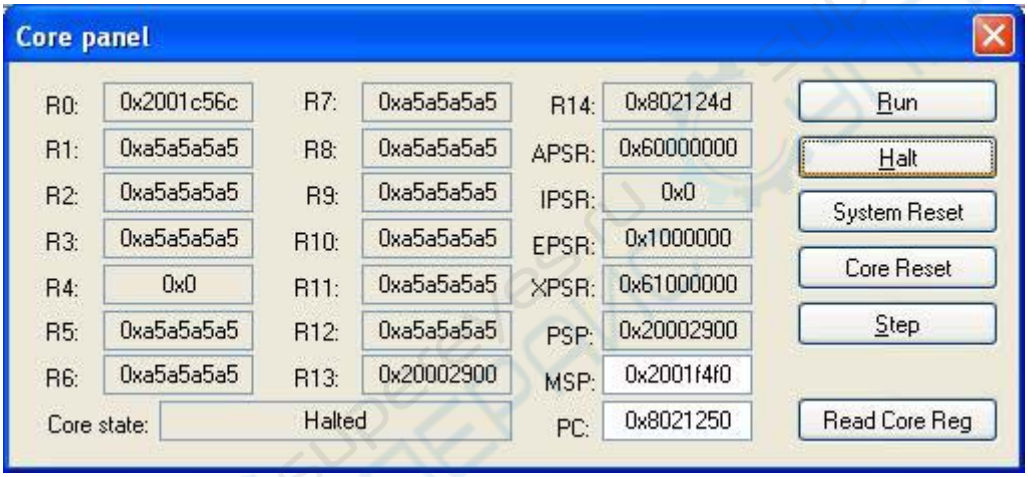
For more details, please refer to the Option Bytes section in the Flash programming manual and reference manual available from www.st.com.

3.6 MCU core functions

The **Core panel** dialog box shown in *Figure 14*, displays the Cortex™-M3 core register values. It also allows you to carry out the following actions on the MCU, using the buttons on the right:

- **Run**: Run the core.
- **Halt**: Halt the core.
- **System Reset**: Send a system reset request.
- **Core Reset**: Reset the core.
- **Step**: Execute only one step core instruction.
- **Read Core Reg**: Update the core registers values.

Figure 14. MCU Core panel dialog box



Note: The PC and MSP registers can be modified from this panel.

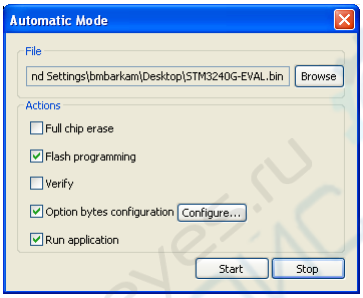
3.7 Automatic mode functions

The **Automatic Mode** dialog box shown in *Figure 15* allows programing and configuring STM32 devices in loop. It allows you to carry out the following actions on the STM32 device:

- **Full chip erase**
- **Flash programming**
- **Verify**
- **Option bytes configuration**
- **Run application**

Clicking on the Start button will execute the selected actions on the connected STM32 device and will wait to repeat the same actions after disconnecting the current device and connecting the new device.

Figure 15. Automatic mode



Note: *If the user deselects Flash programming action while the STM32 Flash memory is readout protected, it will be automatically unprotected.*

If the user deselects Flash programming action while some or all STM32 Flash memory are write-protected, they will be automatically unprotected and restored after programming operation.

Connection to the device should be established to be able to select the option Bytes configuration using the configure button.

The connected devices should be derivatives of the same STM32 family and will be all connected in the same mode (JTAG or SWD).

4 STM32 ST-LINK Utility command line interface (CLI)

4.1 Command line usage

The following sections describe how to use the STM32 ST-LINK Utility from the command line.

The ST-LINK Utility command line interface is located at the following address:

[Install_Directory]\STM32 ST-LINK Utility\ST-LINK Utility\ST-LINK_CLI.exe

4.1.1 Connection and memory manipulation commands

-c

Description: Selects JTAG or SWD communication protocol. By default, JTAG protocol is used.

Syntax: `-c [JTAG/SWD] [UR]`

`[UR]`: Connect to target under reset.

Example: `-c SWD UR`

Note: [UR] available only with ST-LINK/V2 and in SWD mode.
For JTAG mode, connect under reset is available since ST-LINK firmware Version V2J15Sx.
The RESET pin of the JTAG connector (pin 15) should be connected to the device reset pin.

-Q

Description: Enable quiet mode. No progress bar displayed.

-w8

Description: Writes 8-bit data to the specified memory address.

Syntax: `-w8 <Address> <data>`

Example: `-w8 0x20000000 0xAA`

Note: -w8 supports writing to Flash memory, OTP, SRAM and R/W registers.

-w32

Description: Writes 32-bit data to the specified memory address.

Syntax: `-w32 <Address> <data>`

Example: `-w32 0x08000000 0xAABBCCDD`

Note: -w32 supports writing to Flash memory, OTP, SRAM and R/W registers.

-r8

Description: Reads <NumBytes> memory.

Syntax: `-r8 <Address> <NumBytes>`

Example: `-r8 0x20000000 0x100`

4.1.2 Core commands

-Rst

Description: Resets the system.

Syntax: `-Rst`

-HardRst

Description: Hardware reset.

Syntax: `-HardRst`

Note: -HardRst command is available only with ST-LINK/V2. The RESET pin of the JTAG connector (pin 15) should be connected to the device reset pin.

-Run

Description: Sets the Program Counter and Stack pointer as defined at user application and performs a run operation. This is useful if the user application is loaded with an offset (e.g. 0x08003000). If the address is not specified, 0x08000000 is used.

Syntax: `-Run [<Address>]`

Example: `-run 0x08003000`

–Halt

Description: Halts the core.

Syntax: `-Halt`

–Step

Description: Executes Step core instruction.

Syntax: `-Step`

–SetBP

Description: Sets the software or hardware breakpoint at a specific address. If an address is not specified, 0x08000000 is used.

Syntax: `-SetBP [<Address>]`

Example: `-SetBP 0x08003000`

–ClrBP

Description: Clears all hardware breakpoints, if any.

Syntax: `-ClrBP`

–CoreReg

Description: Reads the Core registers.

Syntax: `-CoreReg`

–SCore

Description: Detects the Core status.

Syntax: `-SCore`

4.1.3 Flash commands

-ME

Description: Executes a Full chip erase operation.

Syntax: `-ME`

-SE

Description: Erase Flash sector(s).

Syntax: `-SE <Start_Sector> [<End_Sector>]`

Example:

```
-SE 0    => Erase sector 0
-SE 2 12 => Erase sectors from 2 to 12
*For STM32L series, the following cmd
erases data eeprom:
-SE ed1 => Erases data eeprom at 0x08080000
-SE ed2 => Erases data eeprom at 0x08081800
```

-P

Description: Load binary, Intel Hex or Motorola S-record file into device memory without verification. For hex and srec format, the address is relevant.

Syntax: `-P <File_Path> [<Address>]`

Examples:

```
-P "C:\file.srec"
-P "C:\file.bin" 0x08002000
-P "C:\file.hex"
```

Note: STM32 F2 and STM32 F4 series supports different programming modes depending on the MCU supply voltage. When using ST-LINK/V2, the supply voltage is detected automatically. Therefore, the correct programming mode is selected. When using ST-LINK, the 32-bit programming mode is selected by default.

If the device is read-protected, the protection will be disabled. If some Flash memory pages are write-protected, the protection will be disabled during programming and then recovered.

-V

Description: Verifies that the programming operation was performed successfully.

Syntax: `-V`

Example: `-P "C:\file.srec" -V`

4.1.4 Option byte commands

-rOB

Description: Displays all option bytes.

Syntax: `-rOB`

-OB

Description: Configures the option bytes. This command :

– sets the Read Protection Level to Level 0 (no protection)

– sets the IWDG_SW option to '1' (watchdog enabled by software)

– sets the nRST_STOP option to '0' (reset generated when entering Standby mode)

– sets the Data0 option byte

– sets the Data1 option byte

Syntax:

```
-OB [RDP=<Level>][BOR_LEV=<Level>][IWDG_SW=<Value>]
[nRST_STOP=<Value>][nRST_STDBY=<Value>][BFB2=<Value>]
[nBoot1=<Value>][nSRAM_Parity=<Value>][Data0=<Value>]
[Data1=<Value>][WRP=<Value>][WRP2=<Value>]
[WRP3=<Value>]
```

Example:

```
-OB RDP=0 IWDG_SW=1 nRST_STOP=0
Data0=0xAA Data1=0xBC
```

Option byte command parameter descriptions

RDP=<Level>:

RDP=<Level> sets the Flash memory read protection level.

The <Level> could be one of the following levels:

0: Protection disabled

1: Protection enabled

2: Protection enabled (debug and boot in SRAM features are DISABLED)

Note: Level 2 is available on STM32 F0, *STM32 F2,* STM32 F3, *STM32 F4 and STM32 L1 series only.*

BOR_LEV=<Level>:

BOR_LEV sets the Brownout Reset threshold level.

For STM32 L1 series:

0: BOR OFF, 1.45 to 1.55 V voltage range

1: 1.69 to 1.8 V voltage range

2: 1.94 to 2.1 V voltage range

3: 2.3 to 2.49 V voltage range

4: 2.54 to 2.74 V voltage range

5: 2.77 to 3.0 V voltage range

For STM32 F2 and STM32 F4 series:

0: BOR OFF, 1.8 to 2.10 V voltage range

1: 2.10 to 2.40 V voltage range

2: 2.40 to 2.70 V voltage range

3: 2.70 to 3.60 V voltage range

IWDG_SW=<Value>:

The <Value> should be 0 or 1:

0: Hardware-independent watchdog

1: Software-independent watchdog

nRST_STOP=<Value>:

The <Value> should be 0 or 1:

0: Reset generated when CPU enters the Stop mode

1: No reset generated.

nRST_STDBY=<Value>:

The <Value> should be 0 or 1:

0: Reset generated when CPU enters the Standby mode

1: No reset generated.

BFB2=<Value>:

The <Value> should be 0 or 1:

0: Boot from Flash bank 2 when boot pins are set in boot from user Flash position (default)

1: Boot from Flash bank 1 when boot pins are set in boot from user Flash position (default).

Note: BFB2 is available only on devices containing two Flash banks.

nBoot1=<Value>:

The <Value> should be 0 or 1:

Together with the BOOT0 pin, selects the Boot mode:

nB00T1 = 0 or 1 and BOOT0 = 0 => Boot from Main Flash memory.

nB00T1 = 1 and BOOT0 = 1 => Boot from System memory.

nB00T1 = 0 and BOOT0 = 1 => Boot from Embedded SRAM.

Note: *nBoot1 is available only on STM32 F0* and STM32 F3.

VDDA=<Value>:

Selects the analog monitoring on VDDA Power source.

The <Value> should be 0 or 1.

Note: *VDDA is available only on STM32 F0* and STM32 F3.

nSRAM_Parity=<Value>:

This bit allows the user to enable the SRAM hardware parity check.

The <Value> should be 0 or 1.

Note: *nSRAM_Parity is available only on STM32 F0* and STM32 F3.

SDADC12_VDD=<Value>:

It selects the analog monitoring (comparison with Bgap 1.2V voltage) on SDADC12_VDD Power source.

The <Value> should be 0 or 1.

Note: *SDADC12_VDD is available only on STM32F37x devices.*

Data0=<Value>:

Data0 sets the Data0 option byte.

The <Value> should be in [0..0xFF].

Note: *Not available on* STM32 F0, *STM32 F2,* STM32 F3, *STM32 F4 and STM32 L1 devices.*

Data1=<Value>:

Data1 sets the Data1 option byte.

The <Value> should be in [0..0xFF].

Note: *Not available on* STM32 F0, *STM32 F2,* STM32 F3, *STM32 F4 and STM32 L1 devices.*

SPRMOD =<Value>:

Selection of Protection Mode of WPRi bits:

The <Value> should be 0 or 1.

0: WPRi bits used for Write Protection on sector i (Default).

1: WPRi bits used for PCRoP Protection (read protection) on sector i (Sector)

Note: Available only on devices supporting PCRop feature.

WRP=<Value>:

WRP enables/disables the write protection of the MCU Flash sectors.
Each bit will enable/disable the write protection of one sector or more depending on the connected device.

For STM32 L1 devices, WRP[i] = 0: Flash sector(s) is protected.
For other devices, WRP[i] = 1: Flash sector(s) is protected.

This command is sufficient to enable/disable all Flash sector protection, except for STM32 L1 High density where WRP2 and WRP3 commands are needed.

For STM32 F4 series, each bit of WRP enables/disables the write protection of one sector.

The <Value> should be in [0..0xFFFFFFFF]

Note: For devices supporting PCRop feature, WRP controls the read protection of the MCU Flash sectors when SPRMOD = 1.

WRP2=<Value>:

WRP2 is available only for STM32 L1 medium density plus and high density devices to enable/disable the protection of Flash sectors from page 512 to 1023.

The <Value> should be in [0..0xFFFFFFFF].

Note: For devices supporting PCRop feature, WRP controls the read protection of the MCU Flash sectors when SPRMOD = 1.

WRP3=<Value>:

WRP3 is available only for STM32 L1 high density devices to enable/disable the protection of Flash sectors from page 1024 to 1535.

The <Value> should be in [0..0xFFFFFFFF]

Note: 1 All parameters listed above should be in hexadecimal format.

2 For more details, please refer to the Option Bytes section in the Flash programming manual corresponding to your device available at www.st.com

4.1.5 ST-LINK_CLI return codes

In case of error while executing ST-LINK_CLI commands, the return code (Errorlevel) will be greater than 0.

The following table summarizes the ST-LINK_CLI return codes:

Table 2. ST-LINK_CLI return codes

Return code	Commands	Error
1	All	Command arguments error.
2	All	Connection problem.
3	All	Command not available for the connected target.
4	-w8, -w32	Error occurred while writing data to the specified memory address.
5	-r8, r32	Cannot read memory from the specified memory address.
6	-rst, -HardRst	Cannot reset MCU.
7	-Run	Failed to run application.
8	-halt	Failed to halt the core.
9	-STEP	Failed to perform a single instruction step.
10	-SetBP	Failed to set/clear a breakpoint.
11	-ME, -SE	Unable to erase one or more Flash sectors.
12	-P, -V	Flash programming/verification error.
13	-OB	Option bytes programming error.

5 Revision history

Table 3. Document revision history

Date	Revision	Changes
22-Jan-2010	1	Initial release.
12-Feb-2010	2	Changed figures 1, 2, 3, 4, 5, 6 and 7. Added SWD support.
20-May-2010	3	Added support of XL-density devices in *Section 2.2.3* and *Section 3.5*.
27-Aug-2010	4	Added support of STM32 L1.
03-Feb-2011	5	Hex, srec format support. Command Line Interface support. Changed name and all figures. Added *Section 3.7: Automatic mode functions*
01-Aug-2011	6	Added ST-LINK/V2 support in *Section 1.2: Hardware requirements* and support of different programming modes for STM32 series in *Section 3.4: Device programming* and *Section 4.1.3: Flash commands*. Added MCU revision ID Display in *Section 3.1: Device information*.
18-Oct-2011	7	Added support of STM32W and STM32 F4 throughout the document. Added support of “Connect under reset” option in *Section 2.2.3: Target menu*. Replaced *Figure 14: MCU Core panel dialog box*. Updated Flash with modifications in *Section 3.2: Memory display and modification* via GUI commands and in *Section 4.1.1: Connection and memory manipulation commands* via CLI commands. Added *-HardRst* command in CLI mode in *Section 4.1.2: Core commands*. Added WRP2 and WRP3 for STM32 L1 high density devices in *Option byte command parameter descriptions*.

Table 3. Document revision history (continued)

Date	Revision	Changes
11-May-2012	8	Added support of STM32 F0. Added *Table 1: Applicable tools*. Replaced *Figure 1*, *Figure 8*, *Figure 11*, *Figure 12*, *Figure 13* and *Figure 15*. Added note for JTAG mode in *Section 2.2.3: Target menu*. Added nBoot1, VDDA, and nSRAM_Parity commands in *Option byte command parameter descriptions*. Updated “connect under reset” option in *Section 2.2.3: Target menu*. Updated -c command and added -Q command in *Section 4.1.1: Connection and memory manipulation commands*. Added *Section 4.1.5: ST-LINK_CLI return codes*.
05-Oct-2012	9	Added support of STM32F050, STM32 F3, STM32L medium density plus. Added “Compare two files” in *Section 2.2.1: File menu* and replaced *Figure 2*. Replaced *Figure 4*. Added “Reset after programming option” in *Section 3.4: Device programming* and replaced *Figure 10* to *Figure 12*. Added “SPRMOD” command in *Section 4.1.4: Option byte commands*. Updated Flash sector protection in *Section 3.5: Option bytes configuration* and replaced *Figure 13*. In *Section 3.6: MCU core functions*, replaced *Figure 14*. Updated “WRPx” commands in section *Section 4.1.4: Option byte commands*. Fixed typos.

UM1075
User manual

ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32

Introduction

The ST-LINK/V2 is an in-circuit debugger/programmer for the STM8 and STM32 microcontroller families. The single wire interface module (SWIM) and JTAG/serial wire debugging (SWD) interfaces, facilitate communication with any STM8 or STM32 microcontroller located on an application board.

In addition to providing the same functionalities as the ST-LINK/V2, the ST-LINK/V2-ISOL features digital isolation between the PC and the target application board. It also withstands voltages of up to 2500 VRMS.

The USB full-speed interface allows communication with a PC and:

- STM8 devices via ST Visual Develop (STVD) or ST Visual Program (STVP) software (which are available from STMicroelectronics)
- STM32 devices via Atollic, IAR, Keil, and TASKING integrated development environments.

Table 1. Applicable tools

Type	Part number	Order Code	Description
Development tools	ST-LINK/V2	ST-LINK/V2	In-circuit debugger/programmer
		ST-LINK/V2-ISOL	In-circuit debugger/programmer with digital isolation

Figure 1. ST-LINK/V2 and ST-LINK/V2-ISOL

ST-LINK/V2

ST-LINK/V2-ISOL

Contents

1 Features 5

2 Product contents 6

3 Hardware configuration 8

3.1 Connection with STM8 applications 8

3.1.1 Standard ERNI connection with SWIM flat ribbon 9

3.1.2 Low-cost SWIM connection 10

3.1.3 SWIM signals and connections 11

3.2 Connection with STM32 applications 12

3.3 ST-LINK/V2 status LEDs 14

4 Software configuration 15

4.1 ST-Link/V2 firmware upgrade 15

4.2 STM8 application development 15

4.3 STM32 application development and Flash programming 15

5 Schematics 16

6 Revision history 18

List of tables

Table 1. Applicable tools . . . 1
Table 2. SWIM flat ribbon connections for ST-LINK/V2 . . . 11
Table 3. SWIM low-cost cable connections for ST-LINK/V2-ISOL . . . 11
Table 4. JTAG/SWD cable connections . . . 12
Table 5. How third party toolchains support ST-LINK/V2 . . . 15
Table 6. Document revision history . . . 18

List of figures

Figure 1. ST-LINK/V2 and ST-LINK/V2-ISOL 1
Figure 2. ST-LINK/V2 product contents 6
Figure 3. ST-LINK/V2-ISOL product contents 7
Figure 4. Connectors of the ST-LINK/V2 and ST-LINK/V2-ISOL. 8
Figure 5. ERNI connection 9
Figure 6. Key detail on connectors 9
Figure 7. Low-cost connection 10
Figure 8. Target SWIM connector 11
Figure 9. JTAG and SWD connection 13
Figure 10. JTAG debugging flat ribbon layout 13
Figure 11. SWIM ST-LINK/V2 standard ERNI cable 16
Figure 12. SWIM ST-LINK/V2 low-cost cable 17

1 Features

- 5 V power supplied by a USB connector
- USB 2.0 full speed compatible interface
- USB standard A to mini B cable
- SWIM specific features
  - 1.65 V to 5.5 V application voltage supported on SWIM interface
  - SWIM low-speed and high-speed modes supported
  - SWIM programming speed rate: 9.7 Kbytes/s in low speed and 12.8 Kbytes/s in high speed
  - SWIM cable for connection to the application via an ERNI standard vertical (ref: 284697 or 214017) or horizontal (ref: 214012) connector
  - SWIM cable for connection to the application via a pin header or a 2.54 mm pitch connector
- JTAG/serial wire debugging (SWD) specific features
  - 1.65 V to 3.6 V application voltage supported on the JTAG/SWD interface and 5 V tolerant inputs
  - JTAG cable for connection to a standard JTAG 20-pin pitch 2.54 mm connector
  - Supports JTAG communication
  - Supports serial wire debug (SWD) and serial wire viewer (SWV) communication
- Direct firmware update feature supported (DFU)
- Status LED which blinks during communication with the PC
- 2500 VRMS high isolation voltage (ST-LINK/V2-ISOL only)
- Operating temperature 0 to 50 °C

2 Product contents

Figures *ST-LINK/V2 product contents* show the various cables delivered within the product. They include (from left to right in *Figure 2* and *Figure 3*):

- USB standard A to mini B cable (A)
- ST-LINK/V2 debugging and programming (B)
- SWIM low-cost connector (C)
- SWIM flat ribbon with a standard ERNI connector at one end (D)
- JTAG or SWD and SWV flat ribbon with a 20-pin connector (E)

Figure 2. ST-LINK/V2 product contents

Figure 3. ST-LINK/V2-ISOL product contents

3 Hardware configuration

The ST-LINK/V2 is designed around the STM32F103C8 device which incorporates the high-performance ARM®, Cortex™-M3 core. It is available in a TQFP48 package.

As shown in *Figure 4*, the ST-LINK/V2 provides two connectors:

- an STM32 connector for the JTAG/SWD and SWV interface
- an STM8 connector for the SWIM interface

The ST-LINK/V2-ISOL provides one connector for the STM8 SWIM, STM32 JTAG/SWD and SWV interfaces.

Figure 4. Connectors of the ST-LINK/V2 and ST-LINK/V2-ISOL

1. A = STM32 JTAG and SWD target connector
2. B = STM8 SWIM target connector
3. C = STM8 SWIM, STM32 JTAG and SWD target connector
4. D = Communication activity LED

3.1 Connection with STM8 applications

For STM8 developments, the ST-LINK/V2 can be connected to the target board by two different cables, depending on the connector available on the application board.

These cables are:

- SWIM flat ribbon with a standard ERNI connector at one end
- SWIM cable with two 4-pin, 2.54 mm connector or SWIM separate-wires cable

3.1.1 Standard ERNI connection with SWIM flat ribbon

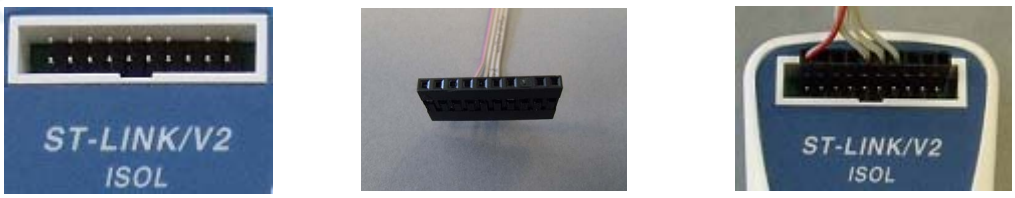
Figure 5 shows how to connect the ST-LINK/V2 if a standard ERNI 4-pin SWIM connector is present on the application board.

Figure 5. ERNI connection

1. A = Target application board with ERNI connector
2. B = Wire cable with ERNI connector at one end
3. C = STM8 SWIM target connector
4. See *Figure 11: SWIM ST-LINK/V2 standard ERNI cable*.

Figure 6 shows that pin 16 is missing on the ST-LINK/V2-ISOL target connector. This missing pin is used as a safety key on the cable connector to guarantee connection of the SWIM cable in the correct position on the target connector even pins, used for both SWIM and JTAG cables.

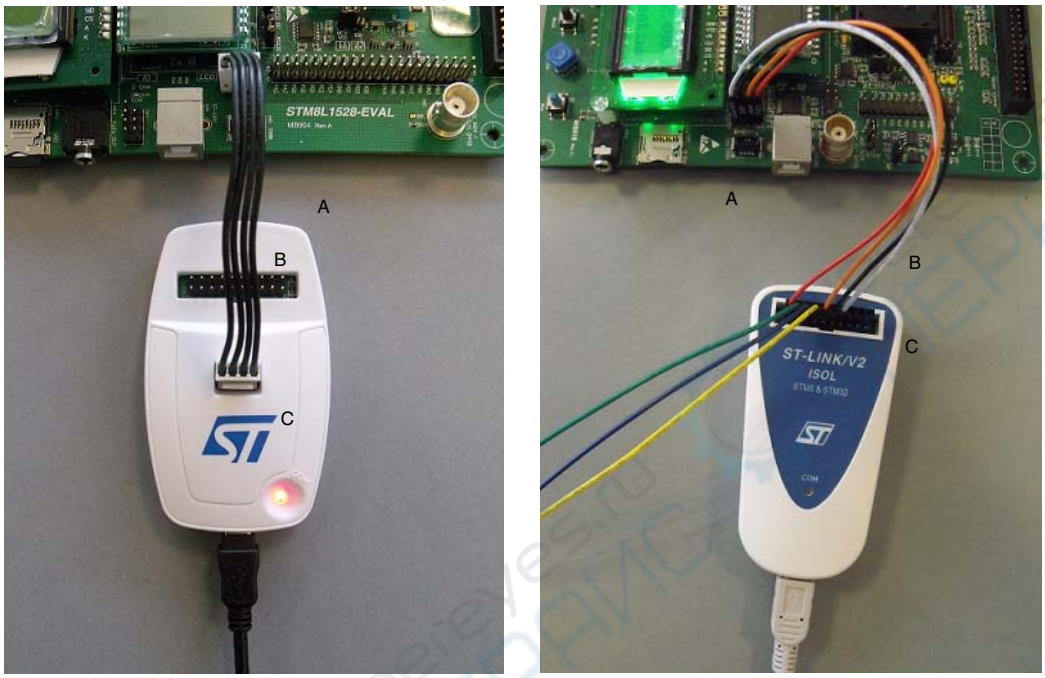
Figure 6. Key detail on connectors



3.1.2 Low-cost SWIM connection

Figure 7 shows how to connect the ST-LINK/V2 if a 4-pin, 2.54 mm, low-cost SWIM connector is present on the application board.

Figure 7. Low-cost connection



1. A = Target application board with 4-pin, 2.54 mm, low-cost connector
2. B = Wire cable with a 4-pin connector or separate-wires cable
3. C = STM8 SWIM target connector
4. See *Figure 12: SWIM ST-LINK/V2 low-cost cable*

3.1.3 SWIM signals and connections

Table 2 summarizes the signal names, functions, and target connection signals using the wire cable with a 4-pin connector.

Table 2. SWIM flat ribbon connections for ST-LINK/V2

Pin no.	Name	Function	Target connection
1	VDD	Target VCC(1)	MCU VCC
2	DATA	SWIM	MCU SWIM pin
3	GND	GROUND	GND
4	RESET	RESET	MCU RESET pin

1. The power supply from the application board is connected to the ST-LINK/V2 debugging and programming board to ensure signal compatibility between both boards.

Figure 8. Target SWIM connector

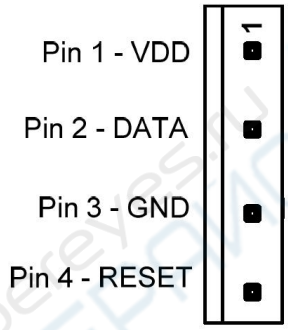


Table 3 summarizes the signal names, functions, and target connection signals using the separate-wires cable.

As the SWIM separate-wires cable has independent connectors for all pins on one side, it is possible to connect the ST-LINK/V2-ISOL to an application board without a standard SWIM connector. On this flat ribbon, all signals are referenced by a specific color and a label to ease the connection on target.

Table 3. SWIM low-cost cable connections for ST-LINK/V2-ISOL

Color	Cable pin name	Function	Target connection
Red	TVCC	Target VCC(1)	MCU VCC
Green	UART-RX	Unused	Reserved (2) (not connected on the target board)
Blue	UART-TX		
Yellow	BOOT0		
Orange	SWIM	SWIM	MCU SWIM pin
Black	GND	GROUND	GND
White	SWIM-RST	RESET	MCU RESET pin

1. The power supply from the application board is connected to the ST-LINK/V2 debugging and programming board to ensure signal compatibility between both boards.
2. BOOT0, UART-TX and UART-RX are reserved for future developments.

TVCC, SWIM, GND and SWIM-RST can be connected to a low-cost 2.54 mm pitch connector or to pin headers available on the target board.

3.2 Connection with STM32 applications

For STM32 developments, the ST-LINK/V2 needs to be connected to the application using the standard 20-pin JTAG flat ribbon provided.

Table 4 summarizes the signals names, functions, and target connection signals of the standard 20-pin JTAG flat ribbon.

Table 4. JTAG/SWD cable connections

Pin no.	ST-LINK/V2 connector (CN3)	ST-LINK/V2 function	Target connection (JTAG)	Target connection (SWD)
1	VAPP	Target VCC	MCU VDD(1)	MCU VDD(1)
2				
3	TRST	JTAG TRST	JNTRST	GND(2)
4	GND	GND	GND(3)	GND(3)
5	TDI	JTAG TDO	JTDI	GND(2)
6	GND	GND	GND(3)	GND(3)
7	TMS_SWDIO	JTAG TMS, SW IO	JTMS	SWDIO
8	GND	GND	GND(3)	GND(3)
9	TCK_SWCLK	JTAG TCK, SW CLK	JTCK	SWCLK
10	GND	GND	GND(3)	GND(3)
11	NC	Not connected	Not connected	Not connected
12	GND	GND	GND(3)	GND(3)
13	TDO_SWO	JTAG TDI, SWO	JTDO	TRACESWO(4)
14	GND	GND	GND(3)	GND(3)
15	NRST	NRST	NRST	NRST
16	GND	GND	GND(3)	GND(3)
17	NC	Not connected	Not connected	Not connected
18	GND	GND	GND(3)	GND(3)
19	VDD	VDD (3.3V)(5)	Not connected	Not connected
20	GND	GND	GND(3)	GND(3)

1. The power supply from the application board is connected to the ST-LINK/V2 debugging and programming board to ensure signal compatibility between both boards.
2. Connect to GND for noise reduction on the ribbon.
3. At least one of this pin must be connected to the ground for correct behavior (connecting all of them is recommended).
4. Optional: for Serial Wire Viewer (SWV) trace.
5. Available on ST-LINK/V2 only and not connected on ST-LINK/V2/OPTO.

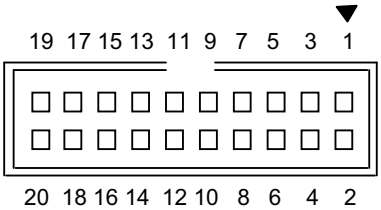
Figure 9 shows how to connect the ST-LINK/V2 to a target using the JTAG cable.

Figure 9. JTAG and SWD connection

1. A = Target application board with JTAG connector
2. B = JTAG/SWD 20-wire flat cable
3. C= STM32 JTAG and SWD target connector

The reference of the connector needed on the target application board is:
2x10C header wrapping 2x40C H3/9.5 (pitch 2.54) - HED20 SCOTT PHSD80.

Figure 10. JTAG debugging flat ribbon layout



Viewed from above PCB

ai18744

Note: For low cost applications or when the standard 20 pins pitch 2.54mm connector footprint is too big, it is possible to implement the Tag-Connect solution to save cost and space on the application board. The Tag-Connect adapter and cable provide a simple reliable means of connecting ST-LINK/V2 or ST-LINK/V2-ISOL to your PCB without requiring a mating

component on application PCB. For more details on this solution and application PCB footprint information, please visit www.Tag-Connect.com.

The references of components compatible with JTAG and SWD interfaces are:

a) TC2050-ARM2010 adapter (20pins to 10pins interface board)

b) TC2050-IDC or TC2050-IDC-NL (No Legs) (10pins cable)

c) TC2050-CLIP retaining clip for use with TC2050-IDC-NL (optional)

3.3 ST-LINK/V2 status LEDs

The LED labeled 'COM' on top of the ST-LINK/V2 shows the ST-LINK/V2 status (whatever the connection type).

When the:

- LED is blinking RED: the first USB enumeration with the PC is taking place.
- LED is RED: communication between the PC and ST-LINK/V2 is established (end of enumeration).
- LED is blinking GREEN/RED: data are being exchanged between the target and the PC.
- LED is GREEN: the last communication has been successful.
- LED is ORANGE: ST-LINK/V2 communication with the target has failed.

4 Software configuration

4.1 ST-Link/V2 firmware upgrade

The ST-Link/V2 embeds a firmware upgrade mechanism for in-situ upgrade through the USB port. As the firmware might evolve during the whole life of the ST-Link/V2 product (new functionality, bug fixes, support for new microcontroller families …), it is recommended to visit www.st.com/stlinkv2 periodically in order to stay up-to-date with the latest firmware version.

4.2 STM8 application development

Please refer to ST toolset Pack24 with Patch 1 or more recent, which includes ST Visual Develop (STVD) and ST Visual Programmer (STVP).

4.3 STM32 application development and Flash programming

Third party toolchains, Atollic TrueSTUDIO, IAR EWARM, Keil MDK-ARM, and TASKING VX-toolset support ST-LINK/V2 according to the versions given in *Table 5* or the most recent version available.

Table 5. How third party toolchains support ST-LINK/V2

Third party	Toolchain	Version
Atollic	TrueSTUDIO	2.1
IAR	EWARM	6.20
Keil	MDK-ARM	4.20
TASKING	VX-toolset for ARM Cortex-M	4.0.1

The ST-LINK/V2 requires a dedicated USB driver. If the toolset installed it automatically, file *stlink_winusb.inf* is installed in *<WINDIR>/inf* (where <WINDIR> is typically C:/Windows).

If the toolset setup did not install it automatically, the driver can be found on www.st.com:

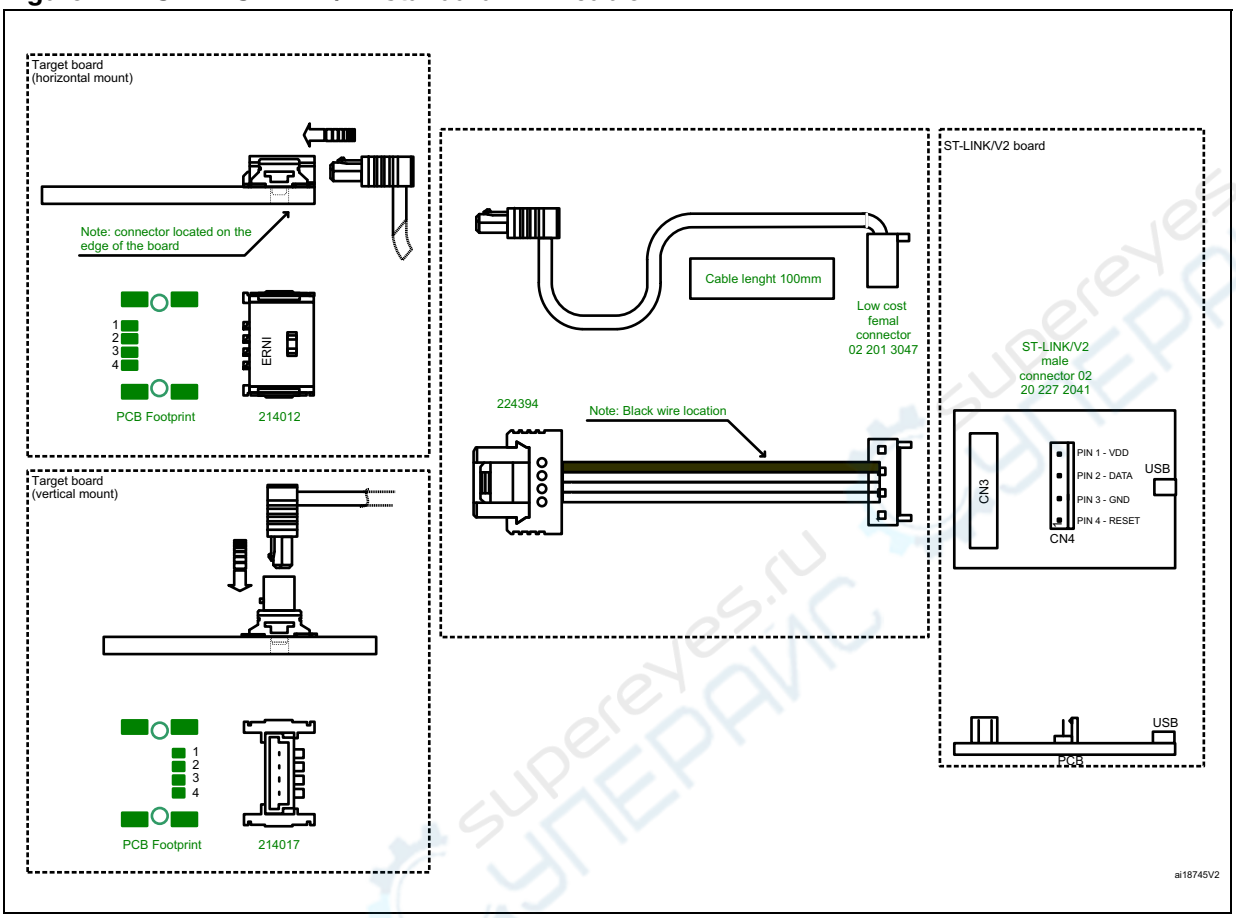
1. Connect to www.st.com.
2. In the search tab, part number field, look for **ST-Link/V2**.
3. Click on the **Generic Part Number** column's hyperlink to ST-Link/V2.
4. In the **Design support** tab, **SW drivers** section, click on the icon to download st-link_v2_usbdriver.zip.
5. Unzip and run ST-Link_V2_USBdriver.exe.

For more information on third party tools, please visit:

- www.atollic.com
- www.iar.com
- www.keil.com
- www.tasking.com

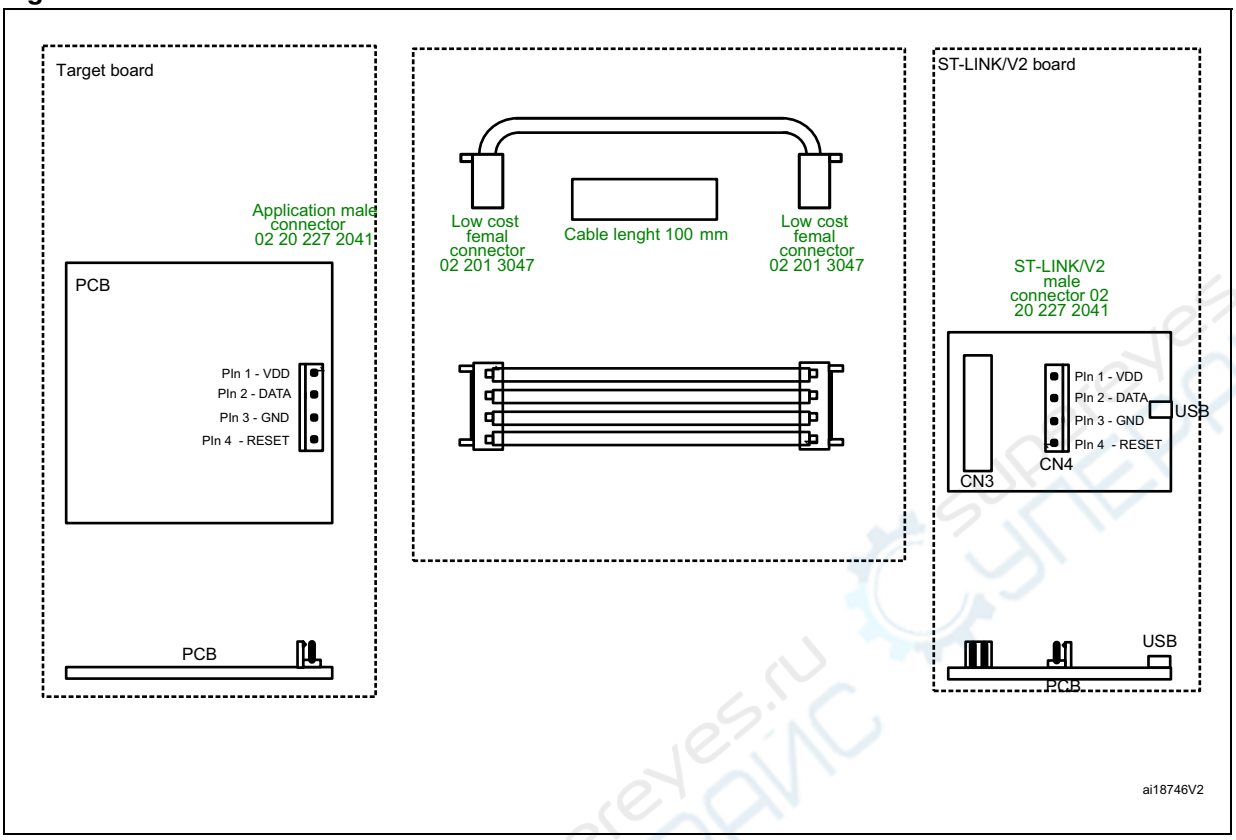
5 Schematics

Figure 11. SWIM ST-LINK/V2 standard ERNI cable



1. Legend for pin descriptions:
   VDD = Target voltage sense
   DATA = SWIM DATA line between target and debug tool
   GND = Ground voltage
   RESET = Target system reset

Figure 12. SWIM ST-LINK/V2 low-cost cable



1. Legend for pin descriptions:
   VDD = Target voltage sense
   DATA = SWIM DATA line between target and debug tool
   GND = Ground voltage
   RESET = Target system reset

6 Revision history

Table 6. Document revision history

Date	Revision	Changes
22-Apr-2011	1	Initial release.
03-Jun-2011	2	*Table 2: SWIM flat ribbon connections for ST-LINK/V2*: added footnote *1* to the function “Target VCC”. *Table 4: JTAG/SWD cable connections*: added footnote to the function “Target VCC”. *Table 5: How third party toolchains support ST-LINK/V2*: updated the “Versions” of IAR and Keil.
19-Aug-2011	3	Added USB driver details to *Section 4.3*.
11-May-2012	4	Added SWD and SWV to JTAG connection features. Modified *Table 4: JTAG/SWD cable connections*.
13-Sep-2012	5	Added ST-LINK/V2-ISOL order code. Updated *Section 4.1: STM8 application development on page 15* Added Note *5.* in *Table 4.* Added Note “For low cost applications...” before *Section 3.3: ST-LINK/V2 status LEDs on page 14*
18-Oct-2012	6	Added *Section 4.1: ST-Link/V2 firmware upgrade on page 15*.

ST-LINK/V2

ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32

Data brief

Features

- 5 V power supplied by a USB connector
- USB 2.0 full speed compatible interface
- USB standard A to mini B cable
- SWIM specific features
  - 1.65 V to 5.5 V application voltage supported on SWIM interface
  - SWIM low-speed and high-speed modes supported
  - SWIM programming speed rate: 9.7 Kbytes/s in low speed and 12.8 Kbytes/s in high speed
  - SWIM cable for connection to the application via an ERNI standard vertical connector (ref: 284697 or 214017) or horizontal connector (ref: 214012)
  - SWIM cable for connection to the application via a pin header or a 2.54 mm pitch connector
- JTAG/serial wire debugging (SWD) specific features
  - 1.65 V to 3.6 V application voltage supported on the JTAG/SWD interface and 5 V tolerant inputs
  - JTAG cable for connection to a standard JTAG 20-pin pitch 2.54 mm connector
  - JTAG supported
  - SWD and serial wire viewer (SWV) communication supported
- Direct firmware update feature supported (DFU)
- Status LED which blinks during communication with the PC
- Operating temperature 0 to 50 °C
- 2500 VRMS high isolation voltage (ST-LINK/V2-ISOL only)

Table 1. Device summary

Part number	Order Code	Description
ST-LINK/V2	ST-LINK/V2	In-circuit debugger/programmer
	ST-LINK/V2-ISOL	In-circuit debugger/programmer with digital isolation

ST-LINK/V2

ST-LINK/V2-ISOL

1 Description

The ST-LINK/V2 is an in-circuit debugger and programmer for the STM8 and STM32 microcontroller families. The single wire interface module (SWIM) and JTAG/serial wire debugging (SWD) interfaces are used to communicate with any STM8 or STM32 microcontroller located on an application board.

In addition to providing the same functionalities as the ST-LINK/V2, the ST-LINK/V2-ISOL features digital isolation between the PC and the target application board. It also withstands voltages of up to 2500 VRMS.

STM8 applications use the USB full speed interface to communicate with STMicroelectronic's ST Visual Develop (STVD) or ST Visual Program (STVP) software.

STM32 applications use the USB full speed interface to communicate with Atollic, IAR, Keil or TASKING integrated development environments.

2 Revision history

Table 2. Document revision history

Date	Revision	Changes
21-Apr-2011	1	Initial release.
07-May-2012	2	Added SWD to JTAG connection features.
14-Sep-2012	3	Added ST-LINK/V2-ISOL.

RN0078
Release note

STM32 ST-LINK Utility release 2.4

About this release note

This release note contains information about the latest version of the STM32 ST-LINK Utility software:

- New features
- Improvements
- Program corrections
- Known problems in current version

Table 1. Applicable tools

Type	Part number
Development tools	ST-LINK, ST-LINK/V2

New features

Version 2.4

- Added support of STM32 F3 series Cortex™-M4.
- Added support of STM32L medium density plus Cortex™-M3.
- Added support of STM32F050 Entry-level Cortex™-M0 microcontrollers
- Comparison between two files (Hex, SREC or Bin files).
- Support of "Reset after programming" option.

Version 2.3

- Added support of STM32 F4 series Cortex™-M4 revision Z.
- Added support of STM32L Cortex™-M3 EEPROM data memory programming.
- Added support of STM32 F0 series Entry-level Cortex™-M0 microcontrollers
- Save file in Intel Hex and Motorola S-record file format.
- Address TextBox replaced by a ComboBox with history saving.

Version 2.2

- Added support of STM32 F4 series of high-performance microcontrollers with DSP and FPU instructions
- Added support of STM32W 32-bit RF microcontrollers
- Added support of “Connect under reset” option.
- Flash modification via GUI and CLI commands.
- Added support of hardware reset in CLI mode.

Version 2.1

- ST-LINK/V2 support.
- Added support for STM32 F2 series microcontroller programming modes.
- Connected MCU revision ID Display.
- ST-LINK and ST-LINK/V2 firmware upgrade to V1J12S3 and V2J14S3, respectively.

Version 2.0

- Support of Intel Hex file and Motorola S-record file format.
- Command line interface.
- Automatic Mode: Programming and configuring STM32 devices in loop.
- Flash memory Blank Check.
- Compare the contents of the Flash memory with files.
- ST-LINK firmware upgrade to V1J11S3.

Version 1.3

- Added support for STM32 L1 ultralow power 32-bit microcontroller devices.
- Added support for STM32 F2 series microcontroller devices.

Version 1.2

- Added support of XL-density devices.

Version 1.1

- Added support of Serial Wire Debug (SWD) interface.

Version 1.0

- Initial release.

Improvements

Version 2.4

- Enhanced display of File/Flash comparison.

Version 2.3

- Added Tooltip in the option bytes window.
- Enable/Disable progress bar display in ST-LINK CLI (STM32 ST-LINK Utility command line interface).
- Added support of programming at unaligned addresses.
- Different error codes return for each type of problem in CLI mode.
- “Connect under reset” option is now available in JTAG mode.

Version 2.2

- -w8 and -w32 CLI commands are able to program Flash memory.
- The Flash memory can be modified from the GUI even when it is not erased.

Version 2.1

- Automatic MCU voltage check with ST-LINK/V2.

Version 2.0

- Faster switch between device memory display and file display.
- Connection management enhancement.
- Core status displayed in the MCU Core window.

Version 1.3

- Flash mass erase and Option bytes programming progress display.

Version 1.2

- Connection management enhancement.

Version 1.1

- Connection error management enhancement.
- Debug interface and connection status display.

Version 1.0

- Initial release.

Program corrections

Version 2.4

- Fix STM32W108CC WRP granularity.
- Fix loading Hex file adjacent segments.
- Display R0 register in Core window.

Version 2.3

- CLI return code issue fixed.

Version 2.2

- None.

Version 2.1

- Binary file save issue resolved with Windows7.

Version 2.0

- Flash size is ignored for STM32 F2 series 32-bit microcontroller devices.

Version 1.3

- None.

Version 1.2

- None.

Version 1.1

- Corrected “STM32 ST-LINK Utility user manual.pdf” name format at start Menu.

Version 1.0

- Initial release.

Known problems in current version

Copy and paste of multiple data in the device memory window can be slow.

Customer support

Software released previously is documented internally and available upon request.

For more information, see the *STM32 ST-LINK utility user manual (UM0892)* and visit our website at *www.st.com*.

Revision history

Table 2. Document revision history

Date	Product Revision	Document Revision	Changes
01-Aug-2011	2.1	1	Initial release.
18-Oct-2011	2.2	2	Added *Version 2.2* with support of STM32 F4 and STM32W series and attributes in *New features* and *Improvements*.
11-May-2012	2.3	3	Added *Table 1: Applicable tools*. Added *Version 2.4* with support of STM32 F0 series and attributes in *New features* and *Improvements*.
29-May-2012	2.3	4	Updated *New features on page 1*, *Improvements on page 2* and *Known problems in current version on page 4*.
05-Oct-2012	2.4	5	Updated *New features on page 1*, *Improvements on page 2* and *Program corrections on page 3*.

RN0084
Release note

ST-LINK/V2 firmware upgrade

About this release note

This release note contains information about the latest version of the ST-LINK/V2 firmware. It can be used to identify the version of the firmware currently running and, if necessary, to upgrade the firmware through the USB port.

Table 1 summarizes the tools covered by this document.

Table 1. Applicable tools

Type	Part number
Development tools	ST-LINK/V2

1 Host PC system requirements

PC and compatibles running with:

- Windows XP, Vista® and Windows Seven 32-bit operating systems
- Windows Vista® and Windows Seven 64-bit operating systems.

The application requires an ST-Link/V2 dedicated USB driver to be installed. In case the driver was not installed by the used toolset, it can be found at *www.st.com/stlinkv2*.

2 Changes in version 2.16.4

On ST-Link/V2, upgrade firmware to V2J16S4.

New features:

- Added command for FPU registers access (STM32F3xxx and STM32F4xxx)
- Firmware compatible for boards with isolation
- Added command for board identification
- Added possibility of hot plug JTAG connection (without target hard reset)
- Bugfix Power not activated in debug domain during JTAG initialization sequence

3 Customer support

For more information or help concerning ST-LINK/V2, please contact the nearest sales office. For a complete list of ST offices and distributors, please refer to *www.st.com*.

4 Revision history

Table 2. Document revision history

Date	Document Revision	Changes
16-Oct-2012	1	Initial release.