
STM32Cube command-line toolset quick start guide

Introduction

This document is a brief guide for users to get started quickly with **STM32CubeCLT**, the STMicroelectronics command-line toolset for STM32 MCUs.

STM32CubeCLT offers all the **STM32CubeIDE** facilities packaged for command-prompt use by third-party IDEs, or continuous integration and continuous development (CD/CI).

The streamlined single STM32CubeCLT package includes:

- CLI (command-line interface) versions of ST tools like toolchain, probe connection utility, and flash memory programming utility
- Up-to-date system view descriptor (SVD) files
- Any other IDE relevant metadata

STM32CubeCLT allows:

- Building a program for STM32 MCU devices using an enhanced GNU toolchain for STM32
- Programming STM32 MCU internal memories (flash memory, RAM, OTP, and others) and external memories
- Verifying the programming content (checksum, verification during and after programming, comparison with file)
- Automating the STM32 MCU programming
- Debugging applications through the interface of STM32 MCU products, which provides access to MCU internal resources using basic debug features



1 General information

The [STM32CubeCLT](#) command-line toolset for STM32 MCUs provides tools to build, program, run, and debug applications targeting STM32 microcontrollers based on the Arm® Cortex®-M processor.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Reference documents

- [Command-line toolset for STM32 MCUs \(DB4839\)](#), STM32CubeCLT data brief
- [STM32CubeCLT installation guide \(UM3089\)](#)
- [STM32CubeCLT release note \(RN0132\)](#)

Screenshots in this document

The screenshots provided in [Section 2](#), [Section 3](#), and [Section 4](#) are only examples of the tool usage from a command prompt.

The integration in third-party IDEs or the use in CD/CI scripts is not illustrated in this document.

2 Building

The `STM32CubeCLT` package contains the GNU tools for STM32 toolchain to build a program for an STM32 microcontroller. A Windows® console window example is shown in [Figure 1](#).

1. Open a console in the project folder.
2. Execute the following command to build the project:
`> make -j8 all -C .\Debug`

Figure 1. Build output

```

PS C:\STM32CubeCLT_TGA_0114> make -j8 all -C .\Debug
make: Entering directory 'C:\STM32CubeCLT_TGA_0114\Debug'
arm-none-eabi-size <LT prog valid.elf
  text  data  bss   dec   hex filename
 7916   20   1572  9508  2524 <LT prog valid.elf
'Finished building: default.size.stdout'
make: Leaving directory 'C:\STM32CubeCLT_TGA_0114\Debug'
PS C:\STM32CubeCLT_TGA_0114>
  
```

DTT1096V1

Note: *The `make` utility might require a separate installation step.*

3 Board programming

The STM32CubeCLT package contains the STM32CubeProgrammer (STM32CubeProg), which is used to program the build obtained previously into the target STM32 microcontroller.

1. Make sure that the ST-LINK connection is detected
2. Select the project folder location in the console window
3. Optionally, erase all the flash memory content (refer to Figure 2):
`> STM32_Programmer_CLI.exe -c port=SWD freq=4000 -e all`
4. Upload the program file to the 0x08000000 flash memory address (refer to Figure 3):
`> STM32_Programmer_CLI.exe -c port=SWD freq=4000 -w .\Debug\YOUR_PROGRAM.elf 0x08000000`

Figure 2. Flash memory erase output

```

Windows PowerShell
PS C:\Users\user> STM32_Programmer_CLI.exe -c port=SWD freq=4000 -e all

-----
STM32cubeProgrammer v2.12.0-B03
-----

ST-LINK SN : 005300263137510133333639
ST-LINK FW : V3J10M3
Board      : NUCLEO-G474RE
Voltage    : 3.29V
SWD freq   : 3300 KHZ
Connect mode: Normal
Reset mode : Software reset
Device ID  : 0x469
Revision ID: Rev B
Device name: STM32G47x/G48x
Flash size : 512 KBytes
Device type: MCU
Device CPU : Cortex-M4
BL version : 0xD3

Mass erase ...
Mass erase successfully achieved
  
```

DTT1087V1

Figure 3. Program upload output

```

Windows PowerShell
PS C:\Users\user> STM32_Programmer_CLI.exe -c port=SWD freq=4000 -w .\Debug\YOUR_PROGRAM.elf 0x08000000

-----
STM32cubeProgrammer v2.12.0-B03
-----

ST-LINK SN : 005300263137510133333639
ST-LINK FW : V3J10M3
Board      : NUCLEO-G474RE
Voltage    : 3.29V
SWD freq   : 3300 KHZ
Connect mode: Normal
Reset mode : Software reset
Device ID  : 0x469
Revision ID: Rev B
Device name: STM32G47x/G48x
Flash size : 512 KBytes
Device type: MCU
Device CPU : Cortex-M4
BL version : 0xD3

Warning: Write address is ignored for hex, sec, s19 and elf files

Memory programming ...
Opening and parsing file: C:\_proj\your_program.elf
File      : C:\_proj\your_program.elf
Size     : 7.75 KB
Address   : 0x08000000

Erasing memory corresponding to segment 0:
Erasing internal memory sectors [0..3]
Download in progress: 100%

File download complete
Time elapsed during download operation: 00:00:00.767
PS C:\Users\user>
  
```

DTT1088V1

4 Debugging

4.1 Start debugging through ST-LINK GDB server

In addition to the GNU tools for STM32 toolchain, the **STM32CubeCLT** package also contains the ST-LINK GDB server. Both are needed to start a debug session.

1. Start the ST-LINK GDB server in another Windows® PowerShell® window (refer to **Figure 4**):
 > ST-LINK_gdbserver.exe -d -v -t -cp C:\ST\STM32CubeCLT\STM32CubeProgrammer\bin
2. Use the GNU tools for STM32 toolchain to start the GDB client in the PowerShell® window:
 > arm-none-eabi-gdb.exe
 > (gdb) target remote localhost:port (use the port indicated in the GDB server opened connection)

The connection is established and GDB server session messages are displayed as shown in **Figure 5**. It is then possible to run GDB commands in the debug session, for instance to reload an `.elf` program using GDB:

- > (gdb) load YOUR_PROGRAM.elf

Figure 4. GDB server output

```

Windows PowerShell
PS C:\ST\STM32CubeCLT\bin> ST-LINK_gdbserver.exe -d -v -t -cp C:\ST\STM32CubeCLT\STM32CubeProgrammer\bin

STMicroelectronics ST-LINK GDB server, Version 22w41_DEVELOPMENT_VERSION
Copyright (c) 2022, STMicroelectronics. All rights reserved.

starting server with the following options:
  Persistent Mode      : Disabled
  Logging Level       : 31
  Listen Port Number  : 61234
  Status Refresh Delay: 15s
  Verbose Mode        : Enabled
  SSO Debug           : Enabled

COM frequency = 24000 kHz
target connection mode: default
Reading ROM table for AP 0 @0x000fffd0
Hardware watchpoint supported by the target
ST-LINK Firmware version = V324043
Device ID: 0x469
PCI: 0x80003a8
ST-LINK device status: HALT_MODE
ST-LINK detects target voltage = 3.29 V
ST-LINK device status: HALT_MODE
ST-LINK device initialization OK
stm32device, pollAndNotify running...
GDB2SV state change: 0 -> 1
waiting for connection on port 61235...
waiting for debugger connection...
waiting for connection on port 61234...
Accepted connection on port 61234...
Debugger connected
waiting for debugger connection...
waiting for connection on port 61234...
GDB session thread running
gdbSessionManager, session started: 1
  
```

DTT1089V1

Figure 5. GDB client output

```

Windows PowerShell
PS C:\ST\STM32CubeCLT\bin> arm-none-eabi-gdb.exe
GNU gdb (GNU tools for STM32 10.1-2021.10.20/11105-1100) 10.2.90.20210623-g1t
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "--host=x86_64-w64-mingw32 --target=arm-none-eabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) target remote localhost:61234
Remote debugging using localhost:61234
warning: No executable has been specified and target does not support
determining executable automatically. Try using the "file" command.
(gdb)
  
```

DTT2000V1

4.2 ST-LINK firmware upgrade

To upgrade the ST-LINK firmware, follow the steps below:

- Connect the ST-LINK probe or the ST-LINK interface of the development board.
- Run the `STLinkUpgrade` command from anywhere in the shell:
`> ./STLinkUpgrade`

Figure 6. ST-LINK upgrade

```

Windows PowerShell
PS C:\Users\hikari> ./STLinkUpgrade
No ST-Link detected
PS C:\Users\hikari> ./STLinkUpgrade
Detected firmware is up to date; use -force_prog to enforce reprogramming.
Firmware version detected: V3J12M3
PS C:\Users\hikari>
  
```

DT72072V2

4.3 Metadata files location

4.3.1 Display location information

To retrieve the location of the various metadata files and STM32CubeCLT package components already installed in the system, use the `STM32CubeCLT_metadata` command:

`> ./ STM32CubeCLT_metadata`

Figure 7. Display of location information (first option)

```

Windows PowerShell
PS C:\Users\hikari> STM32CubeCLT_metadata.bat
=====
STM32Cube Command Line Tools
-- This is a help to show the location of CubeCLT component STM32 -----
=====
STM32CubeTargetRepo = C:\ST\STM32CubeCLT\STM32target-mcu
STM32CubeSVDRepo    = C:\ST\STM32CubeCLT\STMicroelectronics_CMSTIS_SVD
GNUToolsForSTM32   = C:\ST\STM32CubeCLT\GNU-tools-for-STM32\bin
STLinkGDBServer    = C:\ST\STM32CubeCLT\STLink-gdb-server\bin
STM32CubeProgrammer = C:\ST\STM32CubeCLT\STM32CubeProgrammer\bin
=====
PS C:\Users\hikari>
  
```

DT72074V1

4.3.2 Generate JSON for location information

The `STM32CubeCLT_metadata` command can also be used with the `-j` option as its first argument to display the location information in the JSON format:

`> ./ STM32CubeCLT_metadata -j`

Figure 8. Display of location information (second option)

```

Windows PowerShell
PS C:\Users\hikari> STM32CubeCLT_metadata.bat -j
{
  "STM32CubeCLT_MetadataFileLocation" : {
    "STM32CubeTargetRepo" : "C:\ST\STM32CubeCLT\STM32target-mcu",
    "STM32CubeSVDRepo" : "C:\ST\STM32CubeCLT\STMicroelectronics_CMSTIS_SVD",
    "GNUToolsForSTM32" : "C:\ST\STM32CubeCLT\GNU-tools-for-STM32\bin",
    "STLinkGDBServer" : "C:\ST\STM32CubeCLT\STLink-gdb-server\bin",
    "STM32CubeProgrammer" : "C:\ST\STM32CubeCLT\STM32CubeProgrammer\bin" }
}
  
```

DT72075V1

To generate a JSON file containing the location information, add the filepath and filename as the second argument:

`> ./ STM32CubeCLT_metadata -j filepath/filename`

Revision history

Table 1. Document revision history

Date	Revision	Changes
16-Feb-2023	1	Initial release.
20-Jul-2023	2	Updated <i>Debugging</i> : <ul style="list-style-type: none">• Added <i>ST-LINK firmware upgrade</i>• Added <i>Metadata files location</i>
21-Nov-2023	3	Updated <i>ST-LINK firmware upgrade</i> .

Contents

1	General information	2
2	Building	3
3	Board programming	4
4	Debugging	5
4.1	Start debugging through ST-LINK GDB server	5
4.2	ST-LINK firmware upgrade	6
4.3	Metadata files location	6
4.3.1	Display location information	6
4.3.2	Generate JSON for location information	6
	Revision history	7
	List of figures	9

List of figures

Figure 1.	Build output	3
Figure 2.	Flash memory erase output	4
Figure 3.	Program upload output	4
Figure 4.	GDB server output	5
Figure 5.	GDB client output	5
Figure 6.	ST-LINK upgrade	6
Figure 7.	Display of location information (first option).	6
Figure 8.	Display of location information (second option)	6

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2023 STMicroelectronics – All rights reserved