

### Getting started with STM32H7Rx/7Sx MCUs in STM32CubeIDE

### Introduction

This application note describes how to get started with projects based on STM32H7Rx/7Sx microcontrollers in the STMicroelectronics STM32CubeIDE integrated development environment.

Table 1. Applicable products

Туре	Products
Microcontrollers	STM32H7R3/7S3 line, STM32H7R7/7S7 line
Software development tool	STM32CubeIDE







Note:

### 1 General information

STM32CubeIDE supports STM32 32-bit products based on the Arm<sup>®</sup> Cortex<sup>®</sup> processor.

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

### arm

### **1.1 Prerequisites**

The following tools are prerequisites for understanding the tutorial in this document and developing an application based on the STM32H7Rx/7Sx microcontrollers:

- STM32CubeIDE 1.15.0 or newer
- STM32Cube\_FW\_H7RS\_V1.0.0 or newer
- STM32CubeMX 6.12.0 or newer

Users are advised to keep updated with the documentation evolution of the STM32H7Rx/7Sx microcontrollers at www.st.com/en/microcontrollers-microprocessors/stm32h7-series.html.

### 1.2 The use cases in this document

In the STM32CubeIDE context, users have different ways to explore and get started with the development of projects based on the STM32H7Rx/7Sx microcontrollers. From the list below, select the description that best fits the use case considered and refer to the corresponding section in this application note:

- I want to learn with and explore example projects: Refer to Section 2.3: Import a project from the STM32CubeH7RS MCU Package
- I want to start a first STM32H7Rx/7Sx project:
  - Empty project No STM32CubeMX support for maximum flexibility.
     Refer to Section 2.2: Create an empty project based on the template in the STM32CubeH7RS MCU Package
  - STM32CubeH7RS project STM32CubeMX-managed project. Refer to Section 2.1: Create a new STM32 project



## 2 Create and import projects

This chapter describes how to create or import projects for STM32H7Rx/7Sx microcontrollers.

### 2.1 Create a new STM32 project

To start a new project, go to [File]>[New]>[STM32 Project] as shown in Figure 1.

#### Figure 1. New STM32 project

File	Edit Source Refactor Navigate	Search Project	Ru	Run Window Help 💄 Hello	
	New	Alt+Shift+N >	<b>C</b> +	Makefile Project with Existing Code	
	Open File		C	C/C++ Project	
	Open Projects from File System		IDE	IDE STM32 Project	
	Recent Files	>	MX	MX STM32 Project from an Existing STM32CubeMX Configuration File (.ioc)	
	Close Editor	Ctrl+W	IDE	IDE STM32 CMake Project	
	Close All Editors	Ctrl+Shift+W		Project	
	Save	Ctrl+S	62	Source Folder	
	Save As		Ċ	😭 Folder	
	Save All	Ctrl+Shift+S	C	C Source File	
	Revert		h	h Header File	
	Move			File from Template	
P	Rename	F2	G	G Class	
\$	Refresh	F5		🕆 Other Ctrl+N	
	Convert Line Delimiters To	>	Г		
۵	Print	Ctrl+P			
2	Import				
4	Export		Ŀ		
	Properties	Alt+Enter			
	Switch Workspace	>			
	Restart				
	Exit				



Select the desired MCU or board. In the example illustrated in Figure 2, the selected board is the STM32H7S78-DK. Click on [Next >].

V32 Project <b>Selection</b> V32 target or STM32Cube example selection is re	equired						
CU/MPU Selector Board Selector Exa	ample Selector	r Cross Selector					
🛨 🔂 🔂 J		Features	Large Picture	Docs & Resources	[]] C	atasheet	📑 Buy
Commercial Part Number STM32H7S78-DK	~	*					
۹ 🔍	+ -				75		
PRODUCT INFO	~				<b>STM</b> 32		
Туре	>			TRNSK			
Supplier	>		5TM32				
MCU / MPU Series	>			Bitto			
Marketing Status	>						
Price	>		New 600 MHz	oootflash MCU v	vith more		
			performance,	scalability and	security	5//	
MEMORY	~						
Ext. Flash = 1024 (MBit) 1024 Ext. EEPROM = 0 (kBytes) 0 Ext. RAM = 32 (MBit) 32	_	Boards List: 1 item	Overview Commerc	al Part No Type S78-DK Discovery Kit	X Marketing Status	× Unit Price (f 99.0	
FEATURES	~						
Embedded Sensor	>						
User Button	>						
Camera	>						
CAN	>						
Connector	>						

Figure 2. Target selection



After the target selection comes the project setup step shown in Figure 3. The *Targeted Project Type* setting determines whether the project gets generated by STM32CubeMX or not. An *Empty* project is a skeleton of a project that needs building upon while *STM32Cube* indicates an STM32CubeMX-managed project.

	tt _ □ ×
Setup STM32 pr	oject IDE
Project	
Project Name:	STM32H7S78-DK
🗹 Use default	location
Location:	C:/Users/ /STM32CubelDE/workspace_1.16.1_ne Browse
	and and Annull Duringthe different and an Duringth
⊡ Boot P Targeted Bi	roject 🗹 Appli Project 🗹 ExtMemLoader Project
✓ Boot P Targeted Bi ● Executal	rroject ⊡Appli Project ⊡ExtMemLoader Project
Boot P Targeted Bi Executal Targeted Pr Targeted Pr STM32C	roject ☑ Appli Project ☑ ExtMemLoader Project nary Type ble ○ Static Library oject Type ube ○ Empty

#### Figure 3. Projet setup (STM32CubeMX-managed project creation)

The created project contains up to three subprojects:

- STM32H7S78-DK\_Appli: dedicated to the main application development.
- STM32H7S78-DK\_Boot: dedicated to the bootloader development.
- STM32H7S78-DK\_ExtMemLoader (optional and not generated by default): to be used when the user needs an own external flash loader.



# 2.2 Create an empty project based on the template in the STM32CubeH7RS MCU Package

To start a new project, go to [File]>[New]>[STM32 Project] as shown in Figure 1. New STM32 project. Select the desired MCU or board. In the example illustrated in Figure 2. Target selection, the selected board is the

STM32H7S78-DK. Click on [Next >].

After the target selection comes the project setup step shown in Figure 4. The *Targeted Project Type* setting determines whether the project gets generated by STM32CubeMX or not. An *Empty* project is a skeleton of a project that needs building upon while *STM32Cube* indicates an STM32CubeMX-managed project.

DE STM32 Projec	t	_		×
Setup STM32 pr	oject		I	DE
Project				
Project Name:	STM32H7S78-DK			
🗹 Use default	ocation			
Location:	C:/Users/ /STM32CubelDE/workspace_1	.16.1_ne	Brows	se
Project Stru	· cture roject ☑ Appli Project ☑ ExtMemLoader Proj	ect		
Targeted Bin	nary Type Ie ○ Static Library			
Targeted Pr	oject Type Jbe OEmpty			
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>		Cance	1

#### Figure 4. Projet setup (empty project creation)



### 2.3 Import a project from the STM32CubeH7RS MCU Package

To import the STM32Cube firmware project into STM32CubeIDE, go to [File]>[Open Projects from File System...].

Then, select the corresponding project. A project example is by default located at \$Home\STM32Cube\Repository
\STM32Cube\_FW\_H7RS\_V1.0.0\Projects\STM32H7S78-DK\Applications\USB\_Device\MSC\_Standalone\STM32CubeIDE.

#### Figure 5. Firmware project selection

IDE Import Projects from File System or Archive		– 🗆 X
Import Projects from File System or Archive		<u> </u>
This wizard analyzes the content of your folder or archive file to find proje	ects and import them in the IDE.	
Import source: C:\Users\ \STM32Cube\Repository\STM32Cube	e_FW_H7RS_V1.0.0\Projects\STM32H7S78-DK\Applications\U v	Directory
type filter text		Select All
Folder	Import as	Deselect All
STM32CubeIDE	Eclipse project	Entrettant
STM32CubelDE\Appli	STM32CubeIDE project	
	S INIS2CUBEIDE project	
		3 of 3 selected
		<u>H</u> ide already open projects
Close newly imported projects upon completion		
Use installed project configurators to:		
Search for hested projects		
Detect and configure project natures		
Working sets		
Add project to working sets		Ne <u>w</u>
Western and		Caluat
working sets:		Select
	Sh	ow other specialized import wizards
0	Dark	Finish Cancel
	<u>Dack</u> <u>Next</u> >	Cancel

After selecting the project, click on [Finish] to import and build the project.

# \_\_\_\_

57/

### 3 Debugging

This chapter highlights some of the points to bear in mind while debugging an STM32H7Rx/7Sx microcontroller. In the next two sections, this application note covers the configurations needed to start debug sessions with ST-LINK GDB server and OpenOCD.

To start configuring the launch of the debug session, right-click the subproject (in this example, the STM32H7S78-DK\_Appli) and select [**Debug As**]>[**Debug Configurations...**] as shown in Figure 6.

Figure 6. ST-LINK GDB server debug configuration (1 of 4) and OpenOCD debug configuration (1 of 2)

Project Explo	rer >	<	🖻 😫 🍸 🔋				
V DE STM32H7	S78-D	к					
> 🥟 Drivers							
> IDE STM32		New	>				
> IDE STM32		Go Into					
mx.scr		Open in New Window					
MX STM32		Show In	Alt - Chift - 14/ N				
		Show in	AIL+Shirt+W /				
		Сору	Ctrl+C				
	Ē	Paste	Ctrl+V				
	×	Delete	Delete				
		Source	>				
		Move					
		Rename	F2				
	res.	Import					
	4	Export					
		Build Project					
		Clean Project					
	\$ <u>]</u>	Refresh	F5				
		Close Project					
		Close Unrelated Projects					
		Build Configurations	>				
		Build Targets	>				
		Index	>				
	0	Run As	>				
	*	Debug As	>	IDE	1	STM32 C/C++ Applicati	on
		Restore from Local History				ebug Configurations	
	ф.	Generate Code		-	Ľ	coug coningunations	
	C &	Convert to C++					
	*	Run C/C++ Code Analysis					
		Team	>				
		Compare With	>				
		Configure	>				
		Properties	Alt+Enter				

Note:

For the rest of this chapter, examples are based on the STM32H7S78-DK board.

DT73577V1



### 3.1 Setting up with ST-LINK GDB server

By default, the internal flash memory is used to load the program binary. However, due to the small size of the internal flash memory (64 Kbytes), the external memory can be used instead. Multiple linker files are generated for use according to the memory type desired. If the external memory is used, an external flash loader must be set in the debug configuration as shown in Figure 7.

To be able to debug and run the application, the boot must be programmed first in the device memory. In the debug configuration, switch to the *Startup* tab and select [**Add**...].

Debug Configurations				- Π X
Debug configurations				4.
Create, manage, and run configurations				100
				~~~~
🗋 🖻 🗫 🗎 🗶 🖻 🏹 🗸	Name: STM32H7S78-DK Appli Debug			
type filter text	Main 🕸 Debugger 🍉 Startup 🤤 Source 🥅 Common			
C/C++ Application	GDB Connection Settings			^
C/C++ Attach to Application	Autostart local GDB server     Host name or IP address	calhost		
C/C++ Postmortem Debugger	O Connect to remote GDB server Port number 61	234		
C GDB Hardware Debugging	Debug probe ST-LINK (ST-LINK GDB server)			
Launch Group     View STM32 C/C++ Application	GDB Server Command Line Options			
IDE STM32H7S78-DK_Appli Debug	Show Command Line			
	Interface			
	SWD OJTAG			
	ST-LINK S/N		✓ Scan	
	Frequency (kHz): Auto		~	
	Access part: 1 - Cortex-M7			
	Reset behaviour			
	Type: Connect under reset ~			
	Debug Authentication			
	Enable			
	Key path:			Browse
	Certificate path:			Browse
	Permissions:			
	Device settings			
	Debug in low power modes: Enable			~
	Suspend watchdog counters while balted: No configuration			
	Superior witchilding counters while holded. Ho configuration			
	Serial Wire Viewer (SWV)		RIOS Kernel Awareness	
	Care Clash (Mile)		Driver settings	
	Limit SWO clock		Driver: ThreadX	~
	Maximum SWO clock (kHz): auto detect		Port: cortex m0	
	Port number: 61235		Tork Concerno	
			Port number: 60000	
	Euternal lander			
	Lorder	Enabled	Initializa	Add
	STM32H7S78-DK ExtMemLoader.stldr	so true	3 false	- In
				Edit
				Remove
	Misc			
	Verify flash download			
	Enable live expressions	1 11C1 107100		
	Shared ST LINK	workspace_1.10.1_new\511V132	2H7578-DK\Appli\Debug\st-link_gdbserver_log.txt	Browse
	Max halt timeout(s): 2			
				~
Filter matched 8 of 9 items				Re <u>v</u> ert Appl <u>v</u>
(?)				<u>D</u> ebug Close



The STM32H7S78-DK\_Appli configuration is responsible for loading both the STM32H7S78-DK\_Appli and STM32H7S78-DK\_Boot images. Go to the *Startup* tab to do this as shown in Figure 8:

Debug Configurations					- 🗆 X
Create, manage, and run configurations					TO.
📑 🖻 ն 🗮 🗮 🖻 🏹 🗸	Name: STM32H7S78-DK_Appli Debug				,
type filter text	📄 Main 🏇 Debugger 🍺 Startup 🤤 Source 🔲 Common				
C/C++ Application     C/C++ Attach to Application     C/C++ Attach to Application     C/C++ Postmortem Debugger     C/C++ Remote Application     GOB Hardware Debugging	- Initialization Commands				×
🚭 Launch Group	Load Image and Symbols				
V IDE STM32 C/C++ Application	File	Build	Download	Load symbols	Add
Since and since and approved g	Debug/STM32H7S78-DK_Appli.elf [STM32H7S78-DK_Appli]	See Main tab	🗸 true	✓ true	Edit
	Debug\SIM32H/S/8-DK_Boot.eff[SIM32H/S/8-DK_Boot]		V true	V true	Remove
					Move up
					Move down
	Runtime Options				
	Start Address				
	Set program counter (hex):				
	O Specify vector table (hex):				
	Set breakpoint at: main				
	Exception on divide by zero				
	Halt on exception				
	Resume				
	Run Commands				
					^
Filter matched 8 of 9 items					Revert Apply
?					Debug Close

#### Figure 8. ST-LINK GDB server debug configuration (3 of 4)

DT73579V1



To download also the STM32H7S78-DK\_Boot image, click on [Add...], browse the correct project, and build the configuration. The result is shown in Figure 9.

Figure 9. ST-LINK GDE	server debug	configuration	(4	of 4	4)
-----------------------	--------------	---------------	----	------	----

Project:	STM32H7S78-DK_Boot ~
Build configuration:	Use Active $\vee$
Program path:	Debug\STM32H7S78-DK_Boot.elf
	File system
and a second sec	
✓ Perform build ✓ Download ✓ Use download offse	t (hex)
✓ Perform build ✓ Download Use download offse ✓ Load symbols	t (hex)
Perform build     Download     Use download offse     Load symbols     Use symbol address	t (hex)

DT73580V1

The order in the load list is very important. The last entry in the list, marked by a green arrow (refer to Figure 8), is the image debugged with this debug configuration. Consequently, the debugger fetches the program counter value (PC) from this image.

The configuration is complete.

#### 3.1.1 Launching the configurations (ST-LINK GDB server)

To debug the project, launch the STM32H7S78-DK\_Appli debug session. The execution starts at the beginning of the main function from the boot project. Then, it is possible to jump to the application.

Figure 10. ST-LINK GDB server debug configuration launch

脊	• 🛈 • 💁 • 🍅 🛷 • 🏄	R			
IDE	1 STM32H7S78-DK_Appli Debug				
	Debug As	>			
	Debug Configurations				
	Organize Favorites				



### 3.2 Setting up with OpenOCD

Select ST-LINK (OpenOCD) as the [Debug probe] in the debug configuration of the STM32H7S78-DK\_Appli.

Debug Configurations				- 0		
eate, manage, and run configurations				-		
				) (		
i 🖻 🖚 🗎 🗶 🖻 🏹 🕶	Name: STM32H7S78-DK_Appli Debug					
ype filter text	📄 Main 🕸 Debugger 🕨 Startup 🧤 Source 🔲 Common					
C/C++ Application	GDB Connection Settings					
C/C++ Attach to Application	Autostart local GDB server Host name or IP address loc	alhost				
C/C++ Remote Application	Connect to remote GDB server Port number 333	3				
GDB Hardware Debugging	Debug probe ST-LINK (OpenOCD)					
INF STM32 C/C++ Application	GDB Server Command Line Options					
IDE STM32H7S78-DK_Appli Debug	Show Command Line					
	OpenOCD Setup					
	OpenOCD Command:					
	"\${stm32cubeide_openocd_path}\openocd.exe" Browse.					
	OpenOCD Options :			ĉ		
	Configuration Script					
	Automated Generation O User Defined			Show generator options		
	Script File: \${ProjDirPath}\STM32H7S78-DK_Appli Debug.cf	1		Browse Reload		
	Debug Authentication					
	Enable					
	Key path:			Browse		
	Certificate path: Brows					
	Permissions:					
	Serial Wire Viewer (SWV)		RTOS Kernel Awareness			
	Enable		Enable RTOS Proxy			
	Core Clock (MHz): 0.0		Driver settings			
	Limit SWO clock		Driver: ThreadX	~		
	Maximum SWO clock (kHz): auto detect Port: cortex_m0			~		
	Port number: 3344		Port number 60000			
	External loaders					
	Loader	Enabled	Initialize	Add		
	STM32H7S78-DK_ExtMemLoader.stldr	💞 true	😢 false	Edit		
				Remove		
	Misc					
	Enable live expressions					
	ST LINK Client Setue					
	SI-LINK Client Setup					
				Revert Apply		
er matched 8 of 9 items				Revert Apply		

Figure 11. OpenOCD debug configuration (2 of 2)

The rest of the configuration is the same as for debugging with ST-LINK GDB server (refer to Figure 7).



### 3.2.1 Launching the configurations (OpenOCD)

The same approach is used as with ST-LINK GDB server (refer to Section 3.1.1).

### Figure 12. OpenOCD debug configuration launch

脊	• 🛈 • 💁 • 🎒 🛷 • 🏄	P	ā
IDE	1 STM32H7S78-DK_Appli Debug		
	Debug As		>
	Debug Configurations		
	Organize Favorites		

### **Revision history**

### Table 2. Document revision history

Date	Revision	Changes
19-Feb-2025	1	Initial release.



### Contents

1	Gene	ral info	rmation	2
	1.1	Prerequ	uisites	2
	1.2	The use	e cases in this document	2
2	Creat	e and i	mport projects	3
	2.1	Create	a new STM32 project	3
	2.2	Create	an empty project based on the template in the STM32CubeH7RS MCU Package	6
	2.3	Import a	a project from the STM32CubeH7RS MCU Package	7
3	Debu	gging .		8
	3.1	Setting	up with ST-LINK GDB server	9
		3.1.1	Launching the configurations (ST-LINK GDB server)	. 11
	3.2	Setting	up with OpenOCD	. 12
		3.2.1	Launching the configurations (OpenOCD)	. 13
Revi	ision h	istory		.14
List	of tabl	les		.16
List	of figu	ires		.17



### List of tables

Table 1.	Applicable products	L
Table 2.	Document revision history	ŧ



# List of figures

Figure 1.	New STM32 project.	3
Figure 2.	Target selection	4
Figure 3.	Projet setup (STM32CubeMX-managed project creation)	5
Figure 4.	Projet setup (empty project creation)	6
Figure 5.	Firmware project selection	7
Figure 6.	ST-LINK GDB server debug configuration (1 of 4) and OpenOCD debug configuration (1 of 2)	8
Figure 7.	ST-LINK GDB server debug configuration (2 of 4)	9
Figure 8.	ST-LINK GDB server debug configuration (3 of 4)	0
Figure 9.	ST-LINK GDB server debug configuration (4 of 4)	1
Figure 10.	ST-LINK GDB server debug configuration launch 1	1
Figure 11.	OpenOCD debug configuration (2 of 2)	2
Figure 12.	OpenOCD debug configuration launch	3

#### 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.

© 2025 STMicroelectronics – All rights reserved