

# UM2609

User manual

# STM32CubeIDE user guide

# Introduction

STM32CubeIDE is an all-in-one multi-OS development tool, and is part of the STM32Cube software ecosystem. It contains an advanced C/C++ development platform supporting software development of STM32-based products.

This document details the STM32CubeIDE features and usage, including how to get started, create and build projects, debug with standard and advanced techniques, and many other software analysis solutions. STM32CubeIDE is based on the Eclipse C/C++ Development Tools<sup>TM</sup> (CDT<sup>TM</sup>) and GCC toolchain, which cannot be entirely described in this user manual. Additional information on Eclipse<sup>®</sup> is available from the STM32CubeIDE embedded help system. Special documents covering the details of the toolchain and GDB servers are included within the product.





# 1 Getting started

STM32CubeIDE supports STM32 products based on the Arm<sup>®</sup> Cortex<sup>®</sup> processor. Refer to STMicroelectronics documents listed in Section 11 References for details.

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

arm

# **1.1 Product information**

STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, linking, and debug features. It is based on the Eclipse<sup>®</sup>/CDT<sup>™</sup> framework and GCC toolchain for the development, and GDB for the debugging. It allows the integration of the hundreds of existing plugins that complete the features of the Eclipse<sup>®</sup> IDE.

STM32CubeIDE integrates ST MCUFinder (ST-MCU-FINDER-PC) and STM32CubeMX functionalities to offer all-in-one tool experience. It makes it easy to create new STM32 MCU or board projects and build them using the included GCC toolchain.

STM32CubeIDE includes a build analyzer and a static stack analyzer that provide the user with useful information about project status and memory requirements.

STM32CubeIDE also includes standard and advanced debugging features including views of CPU core registers, memories, and peripheral registers, as well as live variable watch, and serial wire viewer interface. A fault analyzer displays error information if an error is triggered by the STM32 processor during a debug session.

Dev	ice sup	port		Projec	t					Debuggir	ng		
Integrated ST-MCU-FINDER	Integrated STM32CubeMX	STMicroelectronics STM32 products	Import System Workbench for STM32	Import Atollic <sup>®</sup> TrueSTUDIO <sup>®</sup>	Build Analyzer	Static Stack Analyzer	SFRs view	Live Expressions view	SWV and ITM views	Multi-core and multi-board debugging	ST-LINK GDB server	OpenOCD GDB server	SEGGER J-Link GDB server
	F	Project	wizard						Deb	ug configura	ation and	launch	
		(	GNU toolcl	nain	GDB debugger								
	Eclipse <sup>®</sup> plugins Modified plugins Eclipse C/C++ Development Tools™ (CDT™)						DT™)						
	Eclipse <sup>®</sup> core platform												
				Suppo	orting W	/indows	s <sup>®</sup> , Linu	x <sup>®</sup> , and	macOS	S®			
Legend		Spec	ific STM320	CubeIDE fu	nctions		Open-b	ased up	dated by	y ST	Base tech	nnology pl	atform
		STM32	CubeIDE n	nain functio	n groups	S	Thir	d-party	solutions	S	Opera	ting syste	ms

Figure 1. STM32CubeIDE key features

# 1.1.1 System requirements

STM32CubeIDE is tested and verified on the Microsoft<sup>®</sup> Windows<sup>®</sup>, Linux<sup>®</sup>, and macOS<sup>®</sup> operating systems.

#### Important:

STM32CubeIDE supports only 64-bit OS versions. For more details about supported versions of operating systems, refer to [ST-02].

Note: macOS<sup>®</sup> is a trademark of Apple Inc. registered in the U.S. and other countries. Linux<sup>®</sup> is a registered trademark of Linus Torvalds.

# 1.1.2 Downloading the latest STM32CubeIDE version

The latest version of STM32CubeIDE is available for free download from the www.st.com/stm32softwaretools website.

# 1.1.3 Installing STM32CubeIDE

The STM32CubeIDE installation guide [ST-04] gives directions on how to install on supported versions of Windows<sup>®</sup>, Linux<sup>®</sup> and macOS<sup>®</sup>. It is possible to have several versions of STM32CubeIDE installed in parallel. Read the installation guide if STM32CubeIDE is not already installed or if a new version must be installed. Installing updates and additional Eclipse plugins in this manual also provides information on how to install updates.



### 1.1.4 License

STM32CubeIDE is delivered under the *Mix Ultimate Liberty+OSS+3rd-party V1* software license agreement (SLA0048).

For more details about the license agreement of each component, refer to [ST-02].

# 1.1.5 Support

There are several different support options provided by STMicroelectronics. For instance, the ST Community is offering places to meet people with similar mind-set all over the world at any time. Choose the support option by visiting my.st.com/content/my\_st\_com/en/support/support-home.html.

# 1.2 Using STM32CubeIDE

# 1.2.1 Basic concept and terminology

The basic concept using STM32CubeIDE and Eclipse<sup>®</sup> terminology is outlined in this section.

#### Workspaces

When starting STM32CubeIDE, a workspace is selected. The workspace contains the development environment to be used. Technically, the workspace is a directory that may hold projects. The user may access any project within the active workspace.

A project contains files, which may be organized into sub-directories. Files existing somewhere else on the computer can also be linked to the project.

A single computer may hold several workspaces at various locations in the file system. The user may switch between workspaces, but only one workspace can be active at a time. Switching workspace is a quick way of switching from one set of projects to another.

In practice, the workspace and project model facilitate a well-structured hierarchy of workspaces, containing projects, which in turn contain files.

#### Information center

The first time STM32CubeIDE is started and a workspace is selected, the *Information Center* is opened. The *Information Center* provides quick access to start a new project, get access to videos, read STM32CubeIDE documentation, or get access to ST support and community. The *Information Center* can be easily accessed at any time via the *Information Center* toolbar button or from the *Help* menu.

#### Perspectives, menu bar, toolbar

When the *Information Center* is closed, STM32CubeIDE displays a perspective, which contains a menu bar, toolbar, views and editors. Each perspective is optimized for a special type of work. For instance, the C/C++ *perspective* is meant for creating, editing and building projects. The *Debug perspective* is intended to be used when debugging code on hardware.

Each perspective can be customized according to the user's need. It is possible to reset the perspective at any time if, for instance, too many views are opened or if the views are reordered. It is also possible to create new perspectives.

#### Views and editors

A perspective normally displays many views. Each view is developed to present specific information, which for instance can be collected from the project or from an embedded system under debug.

A perspective has one editor area. The editor can be used to edit project files. Many files can be edited in different tabs in the editor.



#### STM32CubeIDE window

Name of current workspace	Menu bar To	oolbar	Editor area	Outline view
Morrapac, 0.0.0. gamit 4116 55 - STM32401, Mudes/Com/Sormane - STM32604     Bit Source Reface: Navigate Search Project face with a search Project face	oc E * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 E * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 *	t. rals */ rels */	C Bladd Redupter 21 at 2010 State Anderer      Part of the State State Anderer      Part o	Control of the second sec
	<pre>rmm-nome-eabl-objdump -h -5 51037401_Nucle rmm-nome-eabl-size 510327401_Nucleo.elf 1788 22 1556 58964 2384 5100 inished building: default.size.stdout inished building: STM32F401_Nucleo.list 0:47:31 Build Finished. 0 errors, 1 warning</pre>	eo.elf > "STH02F401_Nucleo.list" ename 32F401_Nucleo.elf gs. (cok 4s.598ms)	ļ	1
(Stopped	¢		<b>&gt;</b>	<u>6</u>
Project Explorer	Cons	sole view	Buil	d Analyzer

#### Figure 2. STM32CubeIDE window

# 1.2.2 Starting STM32CubeIDE

Start STM32CubeIDE by performing the following steps depending on the operating system used.

#### Windows®

If a desktop shortcut is created during the installation of the product, the shortcut can be used to start STM32CubeIDE. The product can also be started from the Windows<sup>®</sup> start menu under STMicroelectronics programs.

Otherwise:

- 1. Locate where STM32CubeIDE is installed, for instance in C:\ST\STM32CubeIDE\_1.0.2
- 2. Open the STM32CubeIDE folder
- 3. Start the stm32cubeide.exe program

#### Linux<sup>®</sup> or macOS<sup>®</sup>

When using Linux<sup>®</sup> or macOS<sup>®</sup>, the program can be started in a similar way by opening the STM32CubeIDE folder where the product is installed.

#### STM32CubeIDE Launcher

When the product is started, it displays the *STM32CubeIDE Launcher* dialog with workspace selection. The first time the product is started, it presents a default location and workspace name. The dialog enables the user to select the name and location of the active workspace for holding all the projects currently accessible by the user. Any newly created project is stored in this workspace. The workspace is created if it does not yet exist.



Note: If Windows<sup>®</sup> is used, avoid locating the workspace folder too many levels below the file system root to avoid exceeding the Windows<sup>®</sup> path length character limitations. Build errors occur if the file paths become longer than what Windows<sup>®</sup> can handle.

IPE STM32CubeIDE Launcher	×
Select a directory as workspace	
STM32CubeIDE uses the workspace directory to store its preferences and development artifac	cts.
Workspace: C:\Users\abc\SIM32CubeIDE\Workspace_um	Browse
Use this as the default and do not ask again	
Recent Workspaces	
Launch	Cancel

Figure 3. STM32CubeIDE Launcher – Workspace selection

Click on the [Launch] button to launch STM32CubeIDE. The first time, it opens the *Information Center*, which is described in Section 1.3 Information Center.

# 1.2.3 Help system

The *Help* menu provides several different help systems as seen in Figure 4. The *Information Center* contains links to all available STM32CubeIDE documentation. It is also recommended for new users to try different Eclipse<sup>®</sup> built-in help systems to get an understanding of Eclipse<sup>®</sup> basics.

# Figure 4. Help menu

irce	Refactor	Navigate	Search	Project	Run	Window	Help
0	Informatio	on Center					
? %	Help Cont Search	tents					
	Show Cor	ntextual He	lp				
\$	Show Act Tips and T Cheat She	ive Keybino Tricks eets	dings			Ctrl+Shif	t+L

# **1.3** Information Center

The Information Center provides quick access to:

- 1. Start a new project
- 2. Import an existing project
- 3. Get access to videos
- 4. Read STM32CubeIDE documentation
- 5. Get access to Getting Started with STM32CubeIDE (STM32CubeIDE quick start guide [ST-03])
- 6. Explore the STM32 MPU and MCU wikis

- 7. Get access to STMicroelectronics support and community on Twitter<sup>™</sup>, Facebook<sup>™</sup>, YouTube<sup>™</sup>, or ST community at community.st.com
- 8. Explore the STMicroelectronics application tools

It is not required to read all material before using the product for the first time. Rather, it is recommended to consider the *Information Center* as a collection of reference information to return to, whenever required.

# 1.3.1 Accessing the Information Center

The Information Center can easily be accessed at any time, from any perspective, using the [Information Center]

toolbar button 1. This icon is located at the right of the toolbar. It is also possible to open the *Information Center* from the [Help]>[Information Center] menu command.

# Figure 5. Help - Information Center menu

File Edit Source Refactor Navigate Search Project Run Window Help





# 1.3.2 Home page

When the *Information Center* is opened, the *Home* page is displayed. It contains links to start a new project, import projects, read documentation and access STMicroelectronics support and community.



#### Figure 6. Information Center – Home page

When using an old workspace, the *Information Center* may not display valid information, showing *"This page can't be displayed"* or opening old manuals when accessing documents. In such case, reload the page by clicking on

the [Home] button at the top right corner of the Information Center window.



#### 1.3.3 Videos

The *Information Center* also contains a video browser page, which is opened from the *Home* page when clicking on the *Access to videos* link.

#### Figure 7. Information Center – Video browser page



Scroll through the *Videos* page and click on a video thumbnail in the list to open it in a web browser. The videos are listed in groups:

- STM32CubeIDE Tutorials
- STM32CubeMX
- Discover your STM32 with STM32CubeIDE

To navigate back to the Home page, press STM32CubeIDE Home at the top left of the Information Center.

# 1.4 Perspectives, editors and views

STM32CubeIDE is a powerful product with many views, loaded with various features. Displaying all views simultaneously would overload the user with information that may not be relevant to the task at hand.

To overcome such a situation, views can be organized in perspectives, where a perspective contains a number of predefined views and an editor area visible by default. A perspective typically handles one development task, such as C/C++ Code Editing or Debugging.





# 1.4.1 Perspectives

The perspectives can be customized according to the user's need; Views can be moved, resized and new views can be opened. It is possible to reset the perspective at any time if, for instance, too many views are opened or if the views are reordered. The perspective is reset by right-clicking the perspective icon in the toolbar and selecting [**Reset**] from the list. This resets the views; Added views in the perspective are closed and the default views are moved to their original location.

# Figure 8. Reset perspective

cce	ss 🗄 🖻 🖬 🔁 🕸 🔤
	Customize
	Save As
	Reset
	Close
	Show Text

As seen in Figure 8, it is also possible to customize a perspective and save the perspective with a new name. Switching from one perspective to another is a quick way to hide some views and display others. To switch perspective, select the [**Open Perspective**] toolbar buttons at the right of the toolbar.

#### Figure 9. Toolbar buttons for switching perspective

C/C++

Another way to switch perspective is to use the menu command [Window]>[Perspective]>[Open Perspective]>[Other...] and select the perspective to use.

#### 1.4.1.1 C/C++ perspective

The C/C++ perspective is intended for creating new projects, editing files, and building the project. The left part of the perspective contains the *Project Explorer* view. The editor is located in the middle. The right part contains some views for the project (*Outline* and *Build Targets* views). At the bottom in the example illustrated in Figure 10, there are the *Problems, Tasks, Console* and *Properties* views. At the lowest right, the *Build analyzer* and *Static stack analyzer* views are displayed.



workspace_um - NUCLEO-F401RE/Core/Src/main.c - 1	STM32CubeIDE	- 🗆 ×
File Edit Source Refactor Navigate Search Project	Run Window Help	
i⊐ • ⊌ % = • <b>% •</b> ≴ i ₫ • 8 • k • 6 • i %	(▼) (4) (3) (3) (4) (5) (4) (5) (4) (5) (4) (5) (5) (5) (5) (5) (5) (5) (5) (5) (5	Quick Access 📑 🖬 🗰 🖬
& Project Explorer 11 E 💘 👻 = 1	Rimaine 2     St Outline 10	aild Targets 🛛 🖂 💘 🔹 🐃 🗸 🗆
NUCLEO-F401RE	93 /* USER CODE BEGIN 2 */	
> 🔅 Binaries	94 95 (# UKER CODE END 3 #/	IRT_HandleTypeDef
> 20 Includes	95 /* USER CODE END Z */ 96	ck_Config(void) : void
Y 🦉 Core	97 /* Infinite loop */  ⊖'MX_GPUO_	init(void) : void
> 🛶 Inc	98 /* USER CODE BEGIN WHILE */	2_UARI_Init(void) : void
🛩 🎃 Src	99 while (1) * main(void)	: int
> 🖻 main.c	140 1 (* USER CODE END MUTILE */ * SystemClo	ck_Config(void) : void
> istm32f4xx_hal_msp.c	+ <sup>s</sup> MK_USAR	2_UART_Init(void) : void
> iii stm32f4xx_it.c	103 /* USER CODE BEGIN 3 */	init(void) : void
> il syscalls.c	104 } • Error Hano	ller(void) : void
> il sysmem.c	105 /* USER CODE END 3 */	d(uint8_t*, uint32_t) : void
> a system_stm32f4xx.c	100 /	
> 😁 Startup	188 /**	
> 🔑 Drivers	189 * @brief System Clock Configuration	
> 🖆 Debug	110 * Wretval None	
NUCLEO-F401RE.ioc	112 wold SystemClock Config(vold)	
STM32F401RETX_FLASH.ld	113 (	
STM32F401RETX_RAM.Id	114 RCC_0scInitTypeDef RCC_0scInitStruct = {0};	
	<pre>115 RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};</pre>	
	110 1172 /PE Configure the main internal negulator output voltage	
	118 */	
	119HAL_RCC_PWR_CLK_ENABLE();	
	120 HAL PAR VOLTAGESCALING CONFIG(PWR REGULATOR_VOLTAGE_SCALE2);	
	121 /* Initializes the CPU, AHB and APB busses clocks	
	122 PC OscipitStruct-OscillatorIve = RCC OSCILLATORIVE HST	
	124 RCC_OscInitStruct.HSIState = RCC_HSI_ON;	
	125 RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;	
	126 RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;	
	127 RCC_OSCINITSTUCT.PLL.PLISOURCE = RCC_PLLSOURCE_HSI; 128 RCC_OSCINITSTUCT.PLL.PLISOURCE_HSI;	
	120 RCC_0straitStruct_PL_PLIN = 36;	
	130 RCC_OSCINITSTruct.PLL.PLLP = RCC_PLLP_DIV4;	
	131 RCC_OscInitStruct.PLL.PLLQ = 7;	
	132 if (HAL_RCC_OscConfig(&RCC_OscInitStruct)  = HAL_OK)	
	🗄 Problems 🖶 Lasks 💭 Console 🕫 🗔 Properties Static Static Analyzer 🕫 🏛 Static Static Analyzer	rt > > 0
	CDT Build Console [NUCLEO-F401RE] NUCLEO-F401RE/Debug - Sep 3, 2019 2:00:29 PM	
	am=none-eablage - "NULEE-fallsReaf" arbjects.its" - ncpue.cortex.m4 - TC: (Users) ohansse(STB3200401 Memory Regions Memory Details	
	Finished building target: NUCLEO-F401RE.elf Bening Set add End addr. Size Fine Hand Hand Million (%)	
	anm-none-abi-obidum -b < MU(FD-F4018F a)f > "NU(FD-F4018F 1)st"	
	text data bis dec hex filename	
	7308 20 1636 8964 2304 NUCLEO-F401RE.elf	
	Finished building: default.size.stdout	
	Einiskad huilding: NUCLED-EddtBE list	
	14:00:30 Build Finished. 0 errors, 0 warnings. (took 105.309ms)	
	¢ >	

# 1.4.1.2 Debug perspective

The *Debug* perspective is intended for debugging the code. The *Debug* perspective is normally opened automatically when a new debug session is started. Later, when the debug session is closed, the perspective is switched back to the C/C++ perspective.

Unit 10 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	(u) (u) (u) (u) (u) (u) (u)						
Bit Displance         Bit Disp	Dep (2 bit   +	(副約)論書★(書)★(本)★(新)用書料及作品(計時以)論(書)書書(書)所以()	$\mathbb{F} = \mathbb{P} \Leftrightarrow \mathbb{F} \Leftrightarrow \mathbb{F} \models \mathbb{F}$				Quick Access 📑 🖬 🕸 🏘
NGUC-40102 (Sov 3) * * Discuss 41 junc 3 (Josen 3) * * Discuss 41 junc 41 jun	Micro 4011 (as 3)       i i forti The application entry point.       iii i forti The application entry point.       iii i forti The application entry point.       iii i forti The application entry point.         I i i forti The and (Sub 100 (The and Sub 2011)       i i forti The application entry point.       iii forti The application entry point.       iii forti The application entry point.         I i i forti The and (Sub 2011)       i i forti The application entry point.       iii forti The application entry point.       iii forti The application entry point.         I i forti The and (Sub 2011)       i forti The application entry point.       iii forti The application entry point.       iii forti The application entry point.         I i forti The application entry point.       i forti The application entry point.       iii forti The application entry point.       iii forti The application entry point.         I i forti The application entry point.       i forti forti The application entry point.       iii forti The application entry point.       iii forti The application entry point.         I i forti forti The application entry point.       i forti for	bug 🕸 🎭 Project Explorer 🕹 🐁 🗭 👘 👘 👘	e main_c 🕸 🗉 Oxffffffe	Variables 🍬	eakp 🚿 Expre	s 🍽 Modules 👘	Registers & Live Ex., SFRs &
2 NULCA 400 KE discussion: <ul> <li>* extra 1 mits:</li> <li>* extra 1 mits:<td><pre>ButCle Status I are in the indication in the indin the indication in the indication in the indica</pre></td><td>NUCLEO-F401RE.elf [STM32 MCU Debugging]</td><td>64 * @brief The application entry point.</td><td>^</td><td></td><td></td><td>RD N N N K</td></li></ul>	<pre>ButCle Status I are in the indication in the indin the indication in the indication in the indica</pre>	NUCLEO-F401RE.elf [STM32 MCU Debugging]	64 * @brief The application entry point.	^			RD N N N K
* # Diesel 1 med Jacob genetatione Basebardi media data data data data data data data d	<pre>* /* Inter 1   lost 1   lost 0   l</pre>	NUCLEO-F401RE.elf (cores: 0)	65 * <u>@retval int</u>	type filter text			
	NGCLO-407HLaff_SHAIX MCU Debuggeg[314]NK (S14JNK CDB server) // /////////////////////////////////	NUCLE-FORTHER (STM2) ALCU Debugging) ⇒ @ Theory of the form of the second sec	<pre>e #print The soplicition entry point. """""""""""""""""""""""""""""""""</pre>	*         Type filter text:           *         Type filter text:           *         Tim Nat.           *         Ti	Addres 68 0x4002 69 0x4002 60 0x4002 60 0x4002 0x40	Value - 000 -	80 × × × × ×
INDECOVERAD INCOME STRUCT DECOUPERING STRUCTURE (STRUCT)			Werffying				

Figure 11. Debug perspective

# 1.4.1.3 Device Configuration Tool perspective

57/

The *Device Configuration Tool* perspective contains the STM32CubeMX device configuration tool integrated in STM32CubeIDE. This perspective is used for device configuration. When an \*.ioc file is opened in an editor and the *Device Configuration Tool* perspective is used, the device can be configured in this perspective. How the device configuration is made is described in [ST-14].





#### 1.4.1.4 Remote System Explorer perspective

The *Remote System Explorer* perspective is basically used when developing STM32 Arm<sup>®</sup> Cortex<sup>®</sup> MPU-based systems. The *Remote Systems* view is used to view files and the *Remote Shell* view is used to run commands.



Figure	12	Romoto	Systom	Evoloror	norsnoctivo
Iguie		Nemole	Jystem		

DEworkspace um1 - NI	ICI FO-E401RE/Core/Src/main.c STM3	2CubalDF		- n x
File Edit Course Defecto	- Neviente Courte Designt Rue Wie			
File Edit Source Relacto	or Navigate Search Project Run Wind			
		發 ¥ : 2월 《 ¥ : 20 : 일 ¥ 신 ¥ 한 우 ¥ 다 ¥ [컵] [1]	Quick A	ccess 🔄 🔛 🛍 🍄 🔤 🛅
📕 Remote Systems 🛛	№ Team	i main.c ⊠		🗄 Outline 🛛 👘 🗖
<ul> <li>♥ Local</li> <li>♥ Local Files</li> <li>&gt; ▷ Drives</li> <li>&gt; ▷ C:</li> <li>&gt; ▷ D:</li> <li>&gt; ▷ D:<!--</td--><td>£2)                                    </td><td><pre>1 /* USER CODE BEGIN Header */ 2@/** 3 4 * @file : main.c 5 * @orief : Main program body 6 7 * @attention 8 * 9 * <h2><center>© Copyright (c) 2019 STMicroelectronics. 10 * All rights reserved.</center></h2> 11 * This software component is licensed by ST under BSD 3-Clause license, 13 * the "License"; You may not use this file except in compliance with the 14 * License. You may obtain a copy of the License at: 15 * opensource.org/licenses/BSD-3-Clause 16 * 17 </pre></td><td>****&gt; ****&gt; *</td><td>□     1<sup>2</sup>     1<sup>3</sup>     •     <b>*</b>     ▼       ■     main.h     •     ■     ■     ■       ■     main.h     •     ●     ■     ■       ■     main.h     •     ■     ■     ■       ■     main.h     •     ■     ■     ■       ■     main.h     •     ■     ■     ■       ●     huart2: UART_Har     •     ●     ©     ©       ●     ©     CRC_NUMBER : co     ●     ©     ©       ●     Distance : uint32_t     ●     Seconds : uint32_t     ●       ●     Seconds : uint32_t     ●     Seconds : uint32_t     ●       ●     SystemClock_Conf     +     *     MX_SPIO_Init/vid       +     *     MX_SPIO_Init/vid     +     *       *     MM_NUSART2_UART     •     •     •       •     min8AM read(h : in vid)     +     *     *</td></li></ul>	£2)	<pre>1 /* USER CODE BEGIN Header */ 2@/** 3 4 * @file : main.c 5 * @orief : Main program body 6 7 * @attention 8 * 9 * <h2><center>© Copyright (c) 2019 STMicroelectronics. 10 * All rights reserved.</center></h2> 11 * This software component is licensed by ST under BSD 3-Clause license, 13 * the "License"; You may not use this file except in compliance with the 14 * License. You may obtain a copy of the License at: 15 * opensource.org/licenses/BSD-3-Clause 16 * 17 </pre>	****> ****> *	□     1 <sup>2</sup> 1 <sup>3</sup> • <b>*</b> ▼       ■     main.h     •     ■     ■     ■       ■     main.h     •     ●     ■     ■       ■     main.h     •     ■     ■     ■       ■     main.h     •     ■     ■     ■       ■     main.h     •     ■     ■     ■       ●     huart2: UART_Har     •     ●     ©     ©       ●     ©     CRC_NUMBER : co     ●     ©     ©       ●     Distance : uint32_t     ●     Seconds : uint32_t     ●       ●     Seconds : uint32_t     ●     Seconds : uint32_t     ●       ●     SystemClock_Conf     +     *     MX_SPIO_Init/vid       +     *     MX_SPIO_Init/vid     +     *       *     MM_NUSART2_UART     •     •     •       •     min8AM read(h : in vid)     +     *     *
		📕 Remote System Details 🧔 Tasks 🖳 Remote Shell 🛛		
		📷 Local 🔀		
< Properties ۲۵ ه. R	emote Scratchpad	09/19/2019         06:34 PM         424,456 stm32cubeide.exe           09/19/2019         06:43 PM         464 stm32cubeide.ini           09/19/2019         06:34 PM         129,544 stm32cubeide.exe           09/19/2019         06:33 PM         600,878 uninstall.exe           09/23/2019         10:33 AM         600,6519 bytes           10 Dir(s)         599,160,823,808 bytes free		^
	📑 🔁 🔽 🗸	<pre>[m]C:\ST\STM32CubeIDE_1.1.0.19w38_targetplatform_2019-09\STM32CubeIDE&gt;</pre>		
Property Name Number of children Type	Value C:\ST\STM32CubeIDE_1.1.0.19w38 0 prompt	<pre>C:\ST\STM32CubeIDE_1.1.0.19w38_targetplatform_2019-09\STM32CubeIDE&gt; </pre>		*
		Command		~ ¥
prompt: C:\ST\STM32Cub	peIDE 1.1.0.19w38 targetplatform 2019	-09\STM32CubelDE>		Status: offline

The *Remote Systems* view contains buttons to open a new connection via FTP, Linux<sup>®</sup>, Local, SSH, Telnet and others.

Figure 14. New connection

DE New Connection					×
Select Remote System	n Type			U	
Local file system on this	computer			=0	=
System type:					
<ul> <li>✓ General</li> <li>General</li> <li>Graph FTP Only</li> <li>Linux</li> <li>Local</li> <li>SSH Only</li> <li>Telnet Only (Exumix Unix</li> <li>Windows</li> </ul>	perimental)				
?	< Back	Next >	Finish	Cancel	

# 1.4.2 Editors

The editor area in a perspective is used by editors. Any number of editors can be opened simultaneously but only one can be active at a time. Different editors can be associated with different file extensions. Example of editors are; c-editor, linker script editor, ioc-file editor for STM32CubeMX device configuration.

To open a file in the editor, double-click on the file in the *Project Explorer* view or open the file via the [**File**] menu. When a file is modified in the editor, it is displayed with an asterisk (\*) indicating that the file has unsaved changes.

#### 1.4.3 Views

Only the most common views associated with the perspective are displayed by default. There are many more views in the product supporting different features. Some of these views only provide valid data when a debug session is ongoing, while others always display data.

Views can be opened from the [Window]>[Show View] menu by selecting one of the views in the list.

# Figure 15. [Show View] menu

workspace_um - NU	CLEO-F401RE/Cor	e/Src,	/main.c - STM3	32CubelD
File Edit Source Refac	tor Navigate Se	arch	Project Run	Window
📑 🕶 🗐 🕼 💌 🔦 🕶	🗟 : 💣 🔻 🚳 🔻 🖸	<b>*</b>	New Window	N
Project Explorer 🛛		n	Editor	>
✓ ■ NUCLEO-F401RE			Appearance	>
Build Analyzer			Show View	>
Build Targets			Perspective	>
C/C++ Projects			Navigation	>
Console	Alt+Shift+Q, C		nungution	
Include Browser			Preferences	
Navigator		8	* < > > > < >	onton 0
Outline	Alt+Shift+Q, O	9	* <n2><ce< td=""><td>anterva ahts re</td></ce<></n2>	anterva ahts re
Problem Details		1	*	siles le
Problems	Alt+Shift+Q, X	2	* This so	oftware
Project Explorer		3	* the "L:	icense"
Properties		4	* License	e. You
Search	Alt+Shift+Q, S	5	*	
SFRs		6	*	
Static Stack Analyzer		7	*******	*****
Tasks		8	*/ /* USER COI	DE END
Other	Alt+Shift+Q, Q	0		
			<pre>/* Includes</pre>	s

The above list of views in Figure 15 is still not complete. It contains only the most common views for the work task related to the perspective currently selected. To access even more views, select [**Other...**] from the list. This opens the *Show View* dialog box. Double-click on any view to open it and access its additional features.

_			
E Show \	/iew		×
type filte	r text		
	++		
	linecuons		
> 🖉 Ger			
	ice Configuration Tool		
	n		
> 🗁 Ma	e (e		
> 🗁 Rer	note Systems		
∽ 🗁 SW	V		
Q	SWV Data Trace		
~	SWV Data Trace Timeline	Graph	
~	SWV Exception Timeline @	Graph	
	SWV Exception Trace Log		
Ē	SWV ITM Data Console		
<b>=</b> :	SWV Statistical Profiling		
	SWV Trace Log		
> 🗁 Tea	m		

#### Figure 16. Show View dialog

The views can be resized and their positions can be changed: Simply drag the view to a new place in STM32CubeIDE. The view can also be dragged outside the STM32CubeIDE window on the screen. Such detached views are shown in separate windows. Detached views works like the other views but are always shown in front of the workbench. Detached views can be attached again by dragging the tab in the detached view into the STM32CubeIDE window.

To restore the perspective to original state, right-click the perspective icon in the toolbar and select [Reset] from the list. Another way to reset the perspective is to use the menu [Window]>[Perspective]>[Reset Perspective].

# 1.4.4 *Quick Access* edit field

The magnifying glass in the toolbar opens the *Quick Access* text box, where any search phrase or keyword can be entered. GUI objects like menu commands, toolbar buttons, preference settings or views can be found using the text box. As any search string is typed, the *Quick Access* shows all the GUI objects that match the criteria, in real time. Type a couple of characters or more and see how the list of results is refined correspondingly on-the-fly.

The *Quick Access* is a time saver when looking for a specific GUI object that cannot be found quickly otherwise, such as a preference setting deeply buried in the configuration dialogs. It is also convenient to retrieve a menu command or toolbar button hidden in the currently active perspective.

For example, in Figure 17, the search string "SWV" entered in the Quick Access provides immediately the list of matching views, GUI commands and preference settings. To open the view or preference setting, click on the GUI object in the search result list.

SWV	v
Views	SWV Data Trace (SWV)
	😾 SWV Data Trace Timeline Graph ( <b>SWV</b> )
	🐱 SWV Exception Timeline Graph ( <b>SWV</b> )
	SWV Exception Trace Log (SWV)
	SWV ITM Data Console (SWV)
	SWV Statistical Profiling (SWV)
	SWV Trace Log (SWV)
Commands	Config - Configure SWV
	Show In (SWV Data Trace Timeline Graph)
	Show In (SWV Data Trace)
	Show In (SWV Exception Timeline Graph)
	Show In (SWV Exception Trace Log)
	Show In (SWV ITM Data Console)
	Show In (SWV Statistical Profiling)
	Show In (SWV Trace Log)
	Show View (SWV Data Trace Timeline Graph) - Shows a particular
	Show View (SWV Data Trace) - Shows a particular view
	Show View (SWV Exception Timeline Graph) - Shows a particul
	Show View (SWV Exception Trace Log) - Shows a particular vie
	Show View (SWV ITM Data Console) - Shows a particular view
	Show View (SWV Statistical Profiling) - Shows a particular view
	Show View (SWV Trace Log) - Shows a particular view
Help	察 Search ' <b>SWV</b> ' in Help

# Figure 17. Quick access

# 1.5 Configuration - Preferences

STM32CubeIDE can be customized in many ways. The menu [Window]>[Preferences] is used to open the *Preferences* dialog. In this dialog, the left pane is used to navigate to certain preference pages. There is also a filter field, which can be used to narrow down the content displayed. The arrow controls on the upper-right side of the dialog can be used to navigate back and forth across pages. The right pane contains the setting of the displayed preferences. Make any preferred change and press [Apply] to update the setting.

[**Restore Defaults**] resets all changes. The preference settings are stored in a metadata folder in the workspace of the application. Section 1.7 Managing existing workspaces in this user manual provides information on how to backup preferences and copy preferences across workspaces.

		—		$\times$
type filter text	C/C++		<-> ▼ □	> • •
<ul> <li>General</li> <li>C/C++</li> <li>Help</li> <li>Install/Update</li> <li>Remote Development</li> <li>Remote Systems</li> <li>Run/Debug</li> <li>STM32Cube</li> <li>Team Terminal</li> </ul>	General settings for C/C++ development:          Outline view         Follow unindexed header files when producing the outline view         Note: Enabling this preference may have negative impact on performance.         Refactoring C/C++ code         Save all modified resources automatically prior to refactoring         Rename in editor without dialog			
	C/C++ dialogs Clear all 'do not show again' settings and show all hidden dialogs again Restore Defaults	5	Clear	ly
? 🖻 🗹	Apply and Close		Cance	I

#### Figure 18. Preferences

It is advised to walk through the preferences pages and get an understanding of the possible configuration options. The following sections present some of them.



# 1.5.1 Preferences - Editors

•

The editor can be configured in many ways. For instance, the menu selection [General]>[Editors]>[Text Editors] provides a *Preferences* pane containing general editor settings such as:

- Displayed tab width
- Insert spaces for tabs
- Highlight current tab
- Show line numbers
- Others

IDE Preferences		$ \Box$ $\times$
line	Z Text Editors	← ▼ ⇒ ▼
line <ul> <li>General</li> <li>Compare/Patch</li> <li>Editors</li> <li>Text Editors</li> <li>Workspace</li> <li>C/C++</li> <li>Editor</li> <li>Scalability</li> <li>Syntax Coloring</li> </ul>	Text Editors         Some editors may not honor all of these settings.         See 'Colors and Fonts' to configure the font.         Undo history size:       200         Displayed tab width:       4         Insert spaces for tabs         Highlight current line         Show print margin         Print margin column:       80	\$ • \$ • •
	<ul> <li>Show line numbers</li> <li>Show range indicator</li> <li>Show whitespace characters (configure visibility)</li> <li>Show affordance in hover on how to make it sticky</li> <li>When mouse moved into hover:</li> <li>Enrich after delay </li> <li>Enable drag and drop of text</li> <li>Warn before editing a derived file</li> <li>Smart caret positioning at line start and end</li> </ul>	
	Appearance color options: Line number foreground Color: Current line highlight Print margin Find scope Selection background color Background color Background color Hyperlink	
	More colors can be configured on the <u>'Colors and Fonts'</u> preference page. Restore Def	aults Apply
	Apply and Clo	vse Cancel

# Figure 19. Preferences - Text Editors



# 1.5.2 Preferences - Code style formatter

It is possible to configure the editor to use special formatting.

The menu selection [C/C++]>[Code Style]>[Formatter] provides a *Preferences* pane containing settings to set an active profile.

Figure 20. Preferences - Formatter

DE Preferences	— 🗆 X
type filter text	Formatter 🗘 🗸 🗸
Perspectives ^	
Project Natures	Configure Project Specific Settings
Search	Active profile:
Security	K&R [built-in] · Edit Remove
Startup and Shutdown	
UI Responsiveness Monitoring	New Import
User Storage Service	Preview:
Web Browser	/*
> Workspace	* A sample source file for the code formatter preview
✓ C/C++	*/
Appearance	<pre>#include <math.h></math.h></pre>
> Build	
Code Analysis	<pre>class Point {</pre>
✓ Code Style	public:
Code Templates	Point( <b>double</b> x, <b>double</b> y) :
Formatter	x(x), y(y) {
Name Style	}
Organize Includes	<pre>double distance(const Point&amp; other) const;</pre>
Core Build Toolchains	
> Debug	double x;
✓ Editor	double y;
Content Assist	};
Folding	< >>
Hovers	Restore Defaults Apply
Mark Occurrences Y	
	Apply and Close Cancel

At this point, if [Edit...] is pressed, a new dialog is opened, where the selected profile can be updated according to specific coding rules. This is displayed in Figure 21.

Profile 'GNU [built-in]'			— 🗆 X
Profile name: GNU [built-in]			Export
Indentation Braces White Space New Lines Co	ontrol Statements Line Wrapping	Comments Off/On Tags	
Brace positions		Preview:	Show invisible characters
Class declaration:	Next line ~	/*	^
Namespace declaration:	Next line ~	* Braces */	
Function declaration:	Next line ~	<pre>#include <math.h></math.h></pre>	
Blocks:	Next line indented $\sim$	<pre>int digits[] =</pre>	
Blocks in case statement:	Next line indented $\sim$	{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }	;
'switch' statement:	Next line indented $\sim$	class Point	
Initializer list:	Next line indented $\sim$	public:	
☐ Keep empty initializer list on one line		<pre>Point (double x, double y) :     x (x), y (y) {     double     distance (const Point&amp; other) cc     int     compareX (const Point&amp; other) cc     double x;     double y; };</pre>	inst; inst;
		<	>
?		Apply	OK Cancel

# Figure 21. Preferences - Code style edit



STM32CubeIDE uses the Internet for instance to get access to STM32 devices information. If a proxy server is used for Internet access, some configuration settings are required in STM32CubeIDE. The proxy settings are set in the *Preferences* pane obtained through [General]>[Network Connections]. To change the settings, set [Active provider] to *Manual* and update the *Proxy entries* for HTTP and HTTPS with specific *Host*, *Port*, *User* and *Password* using the [Edit...] button.

IDE Preferences										×
Network	Netw	ork Co	nnections						<b>() •</b> ()	-> • •
<ul> <li>✓ General</li> <li>✓ Network Connections</li> <li>SSH2</li> </ul>	Active	e Provide entries	er: Manual V							
		Schema	Host	Port	Provider	Auth	User	Password		Edit
		HTTP			Manual	No				Clear
		HTTPS			Manual	No				cicai
		SOCKS	- ·		Manual	No				_
		нттр	Dynamic	Dyna	Native	No				
										-
	Proxy	Proxy bypass								
		Host			Provider				Ad	d Host
		localhost	t		Manual					Edit
		127.0.0.1			Manual				D	
									R	emove
							Restore	Defaults	Арр	bly
							Apply and	d Close	Cance	el

#### Figure 22. Preferences - Network Connections

If there is a problem to save the proxy settings, the reason can be a corrupt secure\_storage file. Proceed as follows to solve the problem:

- 1. Close all running STM32CubeIDE applications
- 2. Rename file C:\Users\user\_name\.eclipse\org.eclipse.equinox.security\secure\_stora ge to a new name
- 3. Restart STM32CubeIDE
- 4. Update the proxy network settings, with user and password information, and save them to create a new se cure\_storage file

Note:

# 1.5.4 Preferences - Build variables

The STM32CubeIDE preferences feature build variables that are only visible in the IDE.

The menu selection [C/C++]>[Build]>[Build Variables] provides a *Preferences* pane with *Build Variables*, which can be used as  $\{VAR\}$  in STM32CubeIDE. Enable [Show system variables] to display all available variables.

Figure 23. Preferences – Build variables

pe filter text		Build Variables			÷ .	- O 4
General	^					
C/C++		Name	Type	Value	^	Add.
Appearance		DriverData	String	C:\Windows\System32\Drivers\DriverData		Edit
✓ Build		eclipse_home	String	<eclipse dynamic="" variable=""></eclipse>		
Build Targets		env_var	String	<eclipse dynamic="" variable=""></eclipse>		Dele
<b>Build Variables</b>		FPS_BROWSER_APP_PROFILE_STRING	String	Internet Explorer		
Console		FPS_BROWSER_USER_PROFILE_STRING	String	Default		
Environment		gnu_arm_embedded_compiler_path	String	<eclipse dynamic="" variable=""></eclipse>		
Logging		gnu_tools_for_stm32_compiler_path	String	<eclipse dynamic="" variable=""></eclipse>		
<ul> <li>Makefile Editor</li> </ul>		HOMEDRIVE	String	C.	~	
Settings Settings Code Analysis > Code Style Core Build Toolchain	~	Show system variables Build Variables are IDE only variables, which ca such as environment variable value or comman	n be used for s nd line paramet	tring substitution when defining external builder ter in form of \$(VAR), internal builder may use the	configu m direc	ration, tly.
Settings Code Analysis > Code Style Core Build Toolchair	v	Build Variables are IDE only variables, which ca such as environment variable value or comman	n be u nd line	sed for s parame	sed for string substitution when defining external builder parameter in form of \$(VAR), internal builder may use the Restore Defaults	sed for string substitution when defining external builder configu parameter in form of \$(VAR), internal builder may use them direc Restore Defaults Au

#### Table 1. Examples of toolchain build variables

Variable	Description
gnu_tools_for_stm32_compiler_path	Path to GNU Tools for STM32 toolchain.
gnu_arm_embedded_compiler_path	Path to GNU Arm Embedded toolchain.
stm32cubeide_make_path	Path to make and BusyBox.

A pre-build step example using build variables to display toolchain version is given in Figure 24.

# Figure 24. Pre-build step using build variables

Properties for NUCLEO	D-F401RE			$\times$
type filter text	Settings		$\langle - \mathbf{v} \rangle \Rightarrow$	• 8
<ul> <li>Resource Builders</li> <li>C/C++ Build Build Variables Environment Logging Settings</li> <li>C/C++ General CMSIS-SVD Settings Project References Run/Debug Settings</li> </ul>	Configuration:       Debug [Active]         Image: The second	Manage Co	v	
0	Apply and	d Close	Cancel	

# 1.6 Workspaces and projects

The basic concepts of workspaces and projects compares as follows:

- A workspace contains projects. Technically, a workspace is a directory containing project directories or references to them.
- A project contains files. Technically, a project is a directory containing files that may be organized in sub-directories.
- A single computer may hold several workspaces at various locations in the file system. Each workspace may
  contain several projects.
- The user may switch between workspaces, but only one workspace can be active at one time.
- The user may access any project within the active workspace. Projects located in another workspace cannot be accessed, unless the user switches to that workspace.
- The files included in a project do not need to be physically located in a folder in the project but can be located somewhere else and linked into the project.
- Switching workspaces is a quick way of shifting from one set of projects to another. It triggers a quick restart
  of the product.

In practice, the project and workspace model facilitates a well-structured hierarchy of workspaces, containing projects, containing files.

# 1.7 Managing existing workspaces

The workspace can be selected when starting STM32CubeIDE. It is also possible to switch to another workspace during the use of STM32CubeIDE. In this case STM32CubeIDE restarts after the new workspace is selected. To restart STM32CubeIDE with a new workspace, select menu [File]>[Switch Workspace].

The workspaces known to STM32CubeIDE can be managed by selecting [Window]>[Preferences] then, in the *Preferences* dialog, selecting [General]>[Startup and Shutdown]>[Workspaces]. In the right pane, it is possible to enable [Prompt for workspace on startup] and set [Number of recent workspaces to remember] to the desired value.

DE Preferences		
type filter text	Workspaces	↓ ↓ ↓ ▼
✓ General		
> Appearance		
Compare/Patch	Number of recent workspaces to remember: 10 🜩	
Content Types	Recent workspaces	
Editors		D
Globalization		Remove
Keys		
Link Handlers		
Network Connections		
Perspectives		
Project Natures		
Search		
Security		
<ul> <li>Startup and Shutdown</li> </ul>		
Workspaces		
UI Responsiveness Monitoring		
User Storage Service		
Web Browser		
> Workspace		
> C/C++		
> Help		
Install/Update		
Remote Development		
Remote Systems	Postoro Dofaulto	Apply
> Run/Debug ~	Restore Defaults	Арріу
	Apply and Close	Cancel

#### Figure 25. Preferences - Workspaces

It is also possible to select and remove recent workspaces from the list of recent workspaces. However, removing a workspace from that list does not remove the files. Neither does it remove the files from the file system.

# 1.7.1 Backup of preferences for a workspace

It is generally a good practice to take a copy of the existing preferences for a workspace. It can be especially useful to recreate the workspace after a crash without the time-consuming process to redo the settings manually. In the menu, select [File]>[Export]. Then, in the panel, select [General]>[Preferences]. Press the [Next] button and, in the next page, enable [Export All] along with a correct filename.

### 1.7.2 Copy preferences between workspaces

To copy workspace preferences from one workspace to another, an existing export of preferences must first be created as explained in Backup of preferences for a workspace.

Then select [File]>[Switch Workspace] and the new workspace. STM32CubeIDE restarts and opens with the new workspace.

In the menu, select [File]>[Import] and in the panel select [General]>[Preferences]. Press the [Next] button and, on the next page, enable [Import All] and enter the file name. The preferences are now the same in both workspaces.

# 1.7.3 Keeping track of Java heap space

To keep track on how much Java heap space is used, select the [Window]>[Preferences] menu. In the *Preferences* page, select the [General] node and then enable [Show heap status]. The currently used and available Java heap space is then displayed in the STM32CubeIDE status bar. The garbage collector can also be triggered manually from the status bar.

Preferences				×
type filter text		General	<	> • •
<ul> <li>General</li> <li>Appearance Compare/Patch Content Types</li> <li>Editors Globalization Keys Link Handlers</li> <li>Network Connections Perspectives Project Natures Search</li> <li>Security</li> <li>Startup and Shutdown Workspaces UI Responsiveness Monitoring</li> <li>User Storage Service Web Browser</li> <li>Workspace</li> <li>C/C++</li> <li>Help</li> <li>Install/Update</li> <li>Remote Development</li> <li>Remote Systems</li> <li>Run/Debug</li> </ul>	~	Always run in background Keep next/previous editor, view and perspectives dialog of Show heap status Workbench save interval (in minutes): 5 Open mode Open mode Open whee Select on hover Open when using arrow keys Note: This preference may not take effect on all views Restore Defaults	oen Appl	у
		Apply and Close	Cancel	

#### Figure 26. Display of Java heap space status

#### 1.7.4 Unavailable workspace

Only one instance of STM32CubeIDE can access one workspace at a time. This is to prevent conflicting changes in the workspace. If STM32CubeIDE is started with a workspace that is already used by another instance of the program, the following error message is displayed.

### Figure 27. Workspace unavailable

DE Works	space Unavailable		×
	The default workspace 'C:/Users/ /STM32CubelD! created. Please choose a different one.	E/workspace_um' is ii	n use or cannot be
		Retry	Choose

If this message is displayed, choose a different workspace, or return to the already running STM32CubeIDE.

# 1.8 STM32CubeIDE and Eclipse<sup>®</sup> basics

STM32CubeIDE contains so many features that it is easy to miss some really useful capabilities. Noteworthy features are spell checking of C/C++ comments, word- and code completion, content assist, parameter hints and code templates. The editor also includes an include-file dependency browser, code navigation using hypertext-links, bookmark and to-do lists, and powerful search mechanisms. The next sections remind some of the useful tools that can be easily missed.

# 1.8.1 Keyboard shortcuts

It is convenient to use keyboard shortcuts instead of the mouse. One important shortcut to know is the shortcut **CTRL+Shift+L**. This shortcut opens a cheat sheet with all available shortcuts.

Activate Editor	F12	^
Backward History	Alt+Left	
Build All	Ctrl+B	
Build Target Build	Shift+F9	
Close	Ctrl+F4	
Close All	Ctrl+Shift+F4	
Collapse All	Ctrl+Shift+Numpad_Divide	
Content Assist	Ctrl+Space	
Context Information	Ctrl+Shift+Space	
Сору	Ctrl+Insert	
Cut	Shift+Delete	
Debug	F11	
Delete	Delete	
Expand All	Ctrl+Shift+Numpad_Multiply	
Find Text in Workspace	Ctrl+Alt+G	
Find and Replace	Ctrl+F	
Forward History	Alt+Right	
Last Edit Location	Ctrl+Q	
Maximize Active View or Editor	Ctrl+M	~
Press 'Ctrl-	+Shift+L' to open the preference pa	age

#### Figure 28. Shortcut keys

Pressing CTRL+Shift+L in this sheet opens the Keys pane in the Preferences dialog.

Figure	29.	Shortcut	preferences

DE Preferences				_	
type filter text	Keys				<> ▼ ¬> ▼ ▼
<ul> <li>✓ General</li> <li>&gt; Appearance</li> <li>Compare/Patch</li> <li>Content Types</li> </ul>	Scheme: Default ~				
<ul> <li>&gt; Editors</li> <li>Globalization</li> <li>Keys</li> <li>Link Handlers</li> <li>&gt; Network Connections</li> <li>Perspectives</li> <li>Project Natures</li> <li>Quick Search</li> <li>&gt; Security</li> <li>&gt; Startup and Shutdown</li> <li>UI Responsiveness Monit</li> <li>&gt; User Storage Service</li> <li>Workspace</li> <li>&gt; C/C++</li> <li>&gt; Help</li> <li>Install/Update</li> <li>&gt; Remote Development</li> <li>&gt; Remote Development</li> <li>&gt; Remote Systems</li> <li>&gt; Run/Debug</li> <li>&gt; STM32Cube</li> <li>&gt; Team</li> <li>Terminal</li> </ul>	Command Activate Editor Add Block Comment Add Bookmark Add Expression Group > Local Variat Add Expression Group > Registers Add Include Add Memory Block Add RegisterGroup Add Task Add to Working Set Align const qualifiers Apply Patch Back	Binding F12 Ctrl+Shift+/ Ole Ctrl+Shift+N Ctrl+Alt+M Ctrl+Shift+A	When In Windows C/C++ Editor C/C++ Editor In Memory View C/C++ Editor	Category Window C/C++ Source Edit Run/Debug C/C++ Source Run/Debug Register Grouping co Edit Edit C/C++ Source Team Navigate	User A
	Copy Command     Unbind Command       Name:     Activate Editor       Description:     Activate the editor       Binding:     F12	Restore Command	Conflicts: Command	Filters When	Export CSV
	When: In Windows	voked		Restore Defaults Apply and Close	Apply Cancel

The *Keys* pane offers the possibility to examine the shortcuts in detail and change the scheme (Default, Emacs, or Microsoft<sup>®</sup> Visual Studio), reconfigure shortcut keys, and others. Table 2 presents some of the keys to mention with their default bindings.

# Table 2. Key shortcut examples

Command	Binding	Where
Сору	Ctrl+C	In dialogs and windows
Cut	Ctrl+X	In dialogs and windows
Paste	Ctrl+V	In dialogs and windows
Debug	F11	In windows
Open declaration	F3	In C/C++ editor
References	Ctrl+Shift+G	In C/C++ editor/views
Find and open files	Ctrl+Shift+R	In C/C++ editor/views
Toggle selection mode normal/block	Alt+Shift+A	In C/C++ editor/views
Zoom In	Ctrl++	Editing text
Zoom Out	Ctrl+-	Editing text



It is possible to increase or decrease the default font size for text editors by pressing Ctrl++ and Ctrl+-:

- Ctrl++ : zoom in text
- Ctrl+- : zoom out text

Note:

If a keyboard with a numeric keypad is used and the + or – keys are pressed on the numeric keypad, use the **Shift** key in addition to make the zoom work (**Ctrl+Shift+** or **Ctrl+Shift-**).

#### workspace\_um1 - NUCLEO-F401RE/Core/Src/main.c - STM32CubeIDE × File Edit Source Refactor Navigate Search Project Run Window Help 1월 두 🔚 🖏 두 🎧 두 📓 : 🔍 : 🕲 : 🙆 두 🚳 두 😢 두 🎯 두 🌞 두 🥐 🖉 두 🖉 🖉 🗐 👖 : 💡 두 🍈 두 🏷 두 두 하 두 🗂 🚺 Quick Access 🔡 📴 🏘 눰 Project Explorer 🛛 🔲 🖶 🤝 🗖 🗖 🗄 Outline 🛛 🛞 Build Targets 尾 main.c 🖂 > 🔤 myLib Y 💯 NUCLEO-F401RE □ 1<sup>2</sup>/<sub>2</sub> ≥ ×<sup>2</sup> ● ¥ 165 Distance : uint32\_t Seconds : uint32\_t SystemClock\_Config(void) : void <sup>\$</sup>MX\_GPI0\_Init(void) : void 166 /\* Configure the system clock \*/ > 🐝 Binaries > 🔊 Includes Y 🐸 Core 167 SystemClock\_Config(); 168 > 🗁 Inc H S MX\_USART2\_UART\_Init(void) : void > In. > Src > & main.c > & stm32f4xx\_hal\_msp.c > & stm32f4xx\_it.c > & stm32f4xx\_it.c 169 /\* USER CODE BEGIN SysInit \*/ ip : int ramx : int 170 myIP\_read() : int myRAM\_read() : int 171 /\* USER CODE END SysInit \*/ InytAM\_read(): Int Jbinay, Jda, Stat: Int Jbinay, Jda, Stat: Int write(int, char', int): Int main(void): Int SystemClock\_Config(void): void <sup>5</sup> MX\_USARI2\_LUARI\_Initivoid): void ∠ stm3c14x0,ttC ≥ @ syscalls.c ≥ @ syscalls.c ≥ @ syscalls.c ≥ @ system\_stm32f4xxc.c > @ Startup > @ Drivers > @ Drivers > @ Debug > @ Debug2 172 /\* Initialize all configured peripherals \*/ 173 174 MX GPTO Tnit(): > 👷 Problems 🧟 Tasks 🖳 Console 🕸 🔲 Properties 🛛 🖻 🛡 🗂 🐨 🖓 💭 🔛 🔛 Build Analyzer 🕸 🚢 Static Stack A No consoles to display at this time. NUCLEO-F401RE.elf - /NUCLEO-F401RE/Debug - Sep 23, 2019 2:05:13 PM Memory Regions Memory Details NUCLEO-F401RE.elf.cfg NUCLEO-F401RE.elf.launch Region BRAM Start address End address 94.27 KB 96 KB 0x20000000 0x20018000 📄 readme.txt 5 FLASH 0x08000000 0x08010000 64 KB 53.7 KB 1.99 KB STM32F401RETX\_FLASH\_IPCodeInFlash.ld FLASH\_D 0x08010000 0x08010800 2 KB 2 KB STM32F401RETX\_FLASH\_ORG.Id STM32F401RETX\_FLASH\_ORG.Id STM32F401RETX\_FLASH\_AM\_CODE.Id STM32F401RETX\_FLASH.Id STM32F401RETX\_RAM.Id STM32F401RETX\_RAM.Id 0x0801f800 0x08020000 1.99 KB < Writable Smart Insert 169 : 16 : 4028

#### Figure 30. Editor with text zoomed in

### 1.8.3 Quickly find and open a file

Pressing **Ctrl+Shift+R** to find and open a file quickly is one of the featured easily missed. Type a couple of characters part of the name of the file to open. It is possible to add the \* and ? search wildcards as appropriate. The editor then lists the matching filenames. Select the desired file in the search result list, and open the file using any of these three ways:

- [Show In]: sends the file to one of the views chosen in the drop-down list (such as the #include file dependency browser view)
- [Open With]: opens the file in the editor selected in the drop-down list
- [Open]: probably the most commonly used option, simply opens the file in the standard C/C++ editor

### 1.8.4 Branch folding

57

A block of code enclosed within #if and #endif can be folded. To activate the functionality, go to [Window]>[Preferences], then [C/C++]>[Editor]>[Folding] and check the [Enable folding of preprocessor branches (#if/#endif)] checkbox. Once the checkbox is checked, the editor must be restarted. Close the file, open it again, and the small icon in the left margin of the editor showing that the functionality is activated.

Figure 31. Editor folding

workspace_um1 - NUCLEO-F401RE/Core/Src/main.c	- STM32	CubeIDE				- 🗆 ×
File Edit Source Refactor Navigate Search Project R	un Windo	w Help				
📑 🕶 🔚 🐚 🏵 🕶 🍕 🕶 🗟 🔌 🤹 📽 🕯 🛣	° ▼ Ĉ י	G • 🕸 • 🙆	) 🔗 🕶 🗾 😼 🗐 🏾	∲ • † † • †	⇒ - 🔁 🖬	Quick Access 📑 🔂 🎋
🔁 Project Explorer 🛛 🗧 🔄 🏹 🖓 🗖	尾 mair	.c 🖾			- 8	🗄 Outline 🛛 🛞 Build Targets 👘 🗖
<ul> <li>ImyLib</li> <li>WILCLEO-F401RE</li> <li>Binaries</li> <li>Binaries</li> <li>Binaries</li> <li>Core</li> <li>Core</li> <li>Src</li> <li>Src</li> <li>Src</li> <li>Syscalis.c</li> <li>Syscalis.c</li> <li>Syscalis.c</li> <li>Syscalis.c</li> <li>Syscalis.c</li> <li>System_stm32f4xx.it.c</li> <li>System.stm32f4xx.c</li> <li>System.stm32f4xx.c</li> <li>Sobug2</li> <li>Debug2</li> <li>Debug2</li> <li>Debug2</li> <li>Release</li> <li>NUCLEO-F401RE:elf.cfg</li> <li>NUCLEO-F401RE:Isf.auch</li> <li>NUCLEO-F401RE:Isf.auch</li> <li>STM32F401RE:TX_FLASH_IPCodeInFlash.ld</li> <li>STM32F401RE:TX_FLASH_ORG.Id</li> <li>STM32F401RE:TX_FLASH_RAM_CODE.Id</li> </ul>	161 162 163 164 165 166 167 168 167 167 168 177 173 174 175⊕ 177 178 80 181 182 183 184 185 183 184 185	<pre>/* USER CODI /* USER CODI /* Configur- SystemClock_ /* USER CODI /* USER CODI /* USER CODI /* Initiali: #ifdef TEST_W mem3-malloct mem5-malloct mem5-malloct mem7-malloct #endif MX_USART2_UU /* USER CODI mem=malloct() mem=malloct() mem2-malloct()</pre>	<pre>E BEGIN Init */ E END Init */ e the system cloc _Config(); E BEGIN SysInit */ E END SysInit */ ce all configured t(); t(2); t(2); t(2); t(2); t(2); E BEGIN 2 */ t(2); t(2); t(2); t(2); E BEGIN 2 */ t(2); t</pre>	k */ / peripherals */		<pre>I I v v v v v v v v v v v v v v v v v v</pre>
🖬 STM32F401RETX_FLASH.ld 🔂 STM32F401RETX_RAM.ld	R Prob	lems 🙍 Tasks 📮	Console 🖾 🔲 Prop	oerties 🗖 🗖	🗟 Build Analyzer 🖾 🚊 Stat	ic Stack Analyzer 📑 🔽 🗖 🗖
> 🚾 STM32F401_Ac6	No cons	oles to display at th	is time.	≓ 및 • 📬 •	NUCLEO-F401RE.elf - /NU PM	CLEO-F401RE/Debug - Sep 23, 2019 2:05:13
					Memory Regions Memory [	Details
		Writable	Smart Insert	1:1:0		i Stopped

#### **1.8.5** Block selection mode

Alt+Shift+A toggles the selection mode between normal and block. When the block mode is enabled, a block of text can be selected by either the mouse or the keyboard using the SHIFT+ARROW buttons.

#### Use of the block selection mode

To start using the block selection mode, press **Alt+Shift+A**. Click somewhere in the text and drag down. A column is then marked as shown in Figure 32.



Add some text and see that this text is entered in all marked rows. As an example, the text " $M_{Y}$ " is added and displayed in Figure 33.

Figure 33.	Editor	text	block	addition
------------	--------	------	-------	----------

workspace_um1 - NUCLEO-F401RE/Core/Src/main.c	- STM32CubeIDE	- 🗆 ×
File Edit Source Refactor Navigate Search Project R	ın Window Help	
📑 🕶 🔚 🐚 🥸 🕶 🐔 🕶 📾 🔯 📽 🚳 📸 🕶 6	3 ▼ 🕄 ▼ 🞯 ▼ 🐎 ▼ 🤔 🖋 ▼ 🗾 💹 🗐 Π 🔮 ▼ 将 ▼ 🏷 Φ ▼ 🔿 ▼ 🛃 🚺	Quick Access
陷 Project Explorer 🛛 🛛 🕒 🔁	ic *main.c ⊠	🗖 🗖 📴 Outline 🖾 🛞 Build Targets 👘 🗖
<ul> <li>myLib</li> <li>MUCLEO-F401RE</li> <li>Minical Simarics</li> <li>Includes</li> <li>Simarics</li> <li>Includes</li> <li>Simarics</li> <li>Includes</li> <li>Simarics</li> <li>Includes</li> <li>Simarics</li> <li>Includes</li> <li>Simarics</li> <li>Includes</li> <li>Includes<td>161       /* USER CODE BEGIN Init */         163       /* USER CODE END Init */         164       /* USER CODE END Init */         165       /* Configure the system clock */         166       /* USER CODE BEGIN SysInit */         170       /* USER CODE END SysInit */         171       /* USER CODE END SysInit */         172       /* Initialize all configured peripherals */         173       /* Initialize all configured peripherals */         174       MX_GPIO_Init();         175       #ifdef TEST_MALLOC         176       MY_mem6=malloc(12);         177       My_mem6=malloc(12);         178       My_mem6=malloc(12);         180       mem=malloc(12);         181       #endif         182       Problems @ Tasks @ Console @ Properties         183       mem2=malloc(12);         184       /* USER CODE BEGIN 2 */         185       mem2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12)</td><td>The second second</td></li></ul>	161       /* USER CODE BEGIN Init */         163       /* USER CODE END Init */         164       /* USER CODE END Init */         165       /* Configure the system clock */         166       /* USER CODE BEGIN SysInit */         170       /* USER CODE END SysInit */         171       /* USER CODE END SysInit */         172       /* Initialize all configured peripherals */         173       /* Initialize all configured peripherals */         174       MX_GPIO_Init();         175       #ifdef TEST_MALLOC         176       MY_mem6=malloc(12);         177       My_mem6=malloc(12);         178       My_mem6=malloc(12);         180       mem=malloc(12);         181       #endif         182       Problems @ Tasks @ Console @ Properties         183       mem2=malloc(12);         184       /* USER CODE BEGIN 2 */         185       mem2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12);         187       mex2=malloc(12)	The second
	writable Smart Insert 100:0 [0]	i Stonned



#### Selection and edition of areas

Select a block. In Figure 34, the block starting with "mem3" to "mem7" is selected.

#### Figure 34. Editor column block selection



Copy the selected block by using **Ctrl+C**. This copied text can then be inserted elsewhere. To do so, type **Alt+Shift+A** to toggle the selection modeback to the normal mode, move the cursor to another line, and type **Ctrl+V** to paste the copied columns to the new lines.

workspace_um1 - NUCLEO-F401RE/Core/Src/main.c	- STM320	CubeIDE				– 🗆 ×
File Edit Source Refactor Navigate Search Project R	un Windo	w Help				
📑 • 🗐 🖄 • 🔦 • 🗟 🔪 😂 📩 •	ゴ ▼ 🖩 🗟 ※ ▼ 🗞 ▼ 🗟 ミ 🕼 : @ : @ ▼ 🖄 ▼ ピ ▼ 🗭 ▼ : :参 タ マ : 沙 🖓 副 🗉 ¶ : 🖗 ▼ : : シ タ マ 🖓 ▼ : : シ マ 🖓 🕄 👔 😨 🎄					
Project Explorer 🛛 🛛 🖻 🕏 🌄 🗆	🚺 *mai	n.c 🛛				🗄 Outline 🛛 🛞 Build Targets 👘 🗖
Project Explorer	™mail           179           180           181           182           183           184           185           186           186           187           198           191           192           193           194           195           196           197           198           2000           2001           202           203           204           205	n.c ☆ My_mem6=malloc My_mem7=malloc #endif MX_USART2_UAR1 /* USER CODE E mem=malloc(12 #ifdef OLD mem3=malloc(12 mem6=malloc(12 mem7=malloc(12) mem7=malloc(12 mem7=malloc(12)	<pre>:(12); :(12); :(12); :(12); ::(12); ::(12); ::(12); ::(12); :(12);::(12);::(12);::(12);:</pre>	erties	■ Build Analyzer ⊠ ■ NUCLEO-F401RE.eff	Build Targets       □         Imain       Imain         <
	NO CONSC	nes to uispiay at this t			PM	
					Memory Regions Mer	nory Details
		Writable	Overwrite	202 : 7 : 4577		i Stopped

# 1.8.6

**\\** 

### **Compare files**

To compare two files easily in STM32CubeIDE:

- 1. Select the two files in the Project Explorer view
- 2. Click on one file
- 3. Press CTRL
- 4. Click on the other file

Both files are now marked in the Project Explorer view

5. Right-click and select [Compare With]>[Each Other]

Note: It is possible to configure how the comparison of files is managed. For instance, ignoring white space can be enabled from the preferences. Open the Preferences page using [Window]>[Preferences], select [General]>[Compare/Patch], and enable [Ignore white space].



workspace_um1 - NUCLEO-F401RE/Core/Src/main.c - STM32Cub	eIDE		- 🗆 ×
File Edit Source Refactor Navigate Search Project Run Window	Help		
	° ▼ ☆ ▼   29 🔗 ▼   .⊿ 🖾 🗉 π   21 ▼ 주 ▼ ↔ ↔ ← ⊂	- 11	Quick Access
Project Explorer 🛛 🕞 🕏 🍸 🖓 🖬 🖻 🖍 main.c	x		📴 Outline 🛛 🛞 Build Targets 👘 🗖
> inc       179         > inc       181         > inc       181         > inc       181         > inc       182         > inc       182         > inc       183         > inc       182         > inc       183         > inc <td><pre>My_mem6=malloc(12); my_mem7=malloc(12); ndif (_USART2_UART_Init(); USER CODE BEGIN 2 */ =m=malloc(12); =m2=malloc(12); =m4=malloc(12); =m6=malloc(12); =m7=malloc(</pre></td> <td></td> <td><pre>Pice Pice Pice Pice Pice Pice Pice Pice</pre></td>	<pre>My_mem6=malloc(12); my_mem7=malloc(12); ndif (_USART2_UART_Init(); USER CODE BEGIN 2 */ =m=malloc(12); =m2=malloc(12); =m4=malloc(12); =m6=malloc(12); =m7=malloc(</pre>		<pre>Pice Pice Pice Pice Pice Pice Pice Pice</pre>
> 😂 startup	ns ど Tasks 🖳 Console 🖾 🔝 Properties	Build Analyzer 🖄 🚊	Static Stack Analyzer
Debug     Define     No console	s to display at this time. 📑 🖳 🔻 📩 🔻	NUCLEO-F401RE.elf - PM	/NUCLEO-F401RE/Debug - Sep 23, 2019 2:05:13
STM32F401_Ac6 Debug.cfg		Memory Regions Men	nory Details
2 items selected			i Stopped

The File Differences editor opens and compares both files.



Workspace_um1 - Two-way compare of 'NUCLEO-F401RE/Core/Src/syscalls.c' with 'NU File Edit Source Refactor Navioate Search Project Run Window Help	UCLEO-F401RE/Core/Src/sysmem.c' - STM32CubeIDE			- 🛛 ×
🗂 • 🗏 🕲 🕲 • 🐔 • 📾 🗙 😜 🙆 🖆 • છે • 🗟 • 🞯 • 🐲 • 🥭 🖉	/ • :⊠ ::: • • • • • • • • • • • • • • •			Quick Access 🛛 😰 🚾 🏇
🔁 Project Explorer 🕸 📃 🖻	Tmain.c 20 Compare ('NUCLEO-F401RE/Core/Src/syscalls.c' - 'NUCLEO-F401RE/Core/Src	/sysmem.c') \$2		🚼 Outline 😒 @ Build Targets 🛛 🗸 🗆 🗆
The Edit Score Reference Response Search Project Ban Window Help The Test Score Reference Response Search Project Ban Window Help Project Epipere Test Project Epipere Test	<pre>NetWork of Status and Status</pre>	Agamenacy 12	Image in surre and binary forms, with any of any link the fille is distributed as is, without any of any link the fille is distributed as is, without any of any link the fille is distributed as is, without any of any link the fille is distributed as is, without any of any link the fille is distributed as is, without any of any link the fille is distributed as is, without any of any link the fille is distributed as is, without any of any link the fille is distributed as is, without any of any link the fille is distributed as is, without any of any link the fille is distributed as is, without any of any link the fille is distributed as is as a fille and the fille is distributed as a fille and the fille is distributed as the fille is distributed as a fille and the fille is distributed as a d	Cutine Access Provide an outflor.
	33** ILSS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRI 4 ** ILSS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRI CONTRIBUTION OF THE SOFTWARE	33 ** 34 ** THIS SOFT	WARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTINUE OF THE INTERNATION OF THE INTERNAT	
	No consoles to display at this time.		NUCLEO-F401RE.elf - /NUCLEO-F401RE/Debug - Sep 23, 2015	2:05:13 PM
			Memory Regions Memory Details	
			Region         Start address         End address         Size           WWRAM         0x20000000         0x20018000         96 K8           WWRLASH         0x02000000         0x02018000         64 K8           WWRLASH         0x02000000         0x02018000         64 K8	Free         Used         Usege (%)         ^           94.27 K8         1.73 K8         1.80%         ^           53.7 K8         10.5 K8         Ib 10%         ^           1.39 K8         8         0.29%         ~
Left: 4 : 1, Right: 4 : 1, incoming change #1 (Left: 4 : 4, Right: 4 : 4)				Stonned 🗖
				-

Use the navigation buttons to navigate between differences, or simply navigate in the view using the scroll bar to see the file differences.

57



# 1.8.7 Local file history

57

It is recommended to maintain projects with a version control system such as Apache<sup>®</sup> Subversion<sup>®</sup> (SVN) or Git<sup>™</sup>. Still, STM32CubeIDE contains a local file with the history of edited files, which can be useful if some investigation is needed after a file has become not functional. The workspace preferences contain a *Local History* page.

DE Preferences		$ \square$ $\times$
local ×	Local History	
<ul> <li>General</li> <li>Globalization</li> </ul>	✓ Limit history size	
✓ Workspace	Days to keep files:	7
✓ C/C++	Maximum entries per file:	50
✓ Editor	Maximum file size (MB):	1
Syntax Coloring	Note: The 'Maximum ent	ies per file' and the 'Days to keep files' values
	are only applied wh	en compacting the local history on shutdown.
		Restore Defaults Apply
? 🔟 🗹		Apply and Close Cancel

#### Figure 38. Local history



To show the local history of a file:

- 1. Select the file in the *Project Explorer* view
- 2. Right-click
- 3. Select [Team]>[Show local History]

#### Figure 39. Show local history



The History view opens and displays the file history.



Figure 40. File history

In the case presented in Figure 40, there are three revisions of main.c. Double-click on a file in the *History* view to open it in the editor.

Right-click on a file in the history and select [Compare Current with Local] to compare it with the current version of the file.

💷 workspace_um1 - NUCLEO-F401RE/Core/Src/main.c - STM32CubelDE — 🗆 X			
File Edit Source Refactor Navigate Search Project Run Window Help			
📬 🕶 🔚 🐚 🥸 🕶 🐔 🕶 🔜 🖄 🧯 🍪 🖬 🖝 🕼	≌▼ 🖻 ▼ 🞯 ▼   ৡ ▼   🥭 🖋 ▼   ⊿ 🗟 🗐 👖   🖢 ▼ 🖓 ▼ ⇔ ↔ ↔	- 2 0	Quick Access 🔡 🔂 🄯
🎦 Project Explorer 🛛 🛛 🗐 🔄 🖘 🖓 🖓	i main.c ⊠		🗄 Outline 🛛 🛞 Build Targets 👘 🗖
Multiple     MUCLEO-F401RE     MUCLEO-F401RE     Multiple     Src     Src	<pre>1 /* USER CODE BEGIN Header */ 20/** 3 4 @file : main.c 5 @brief : Main program body 6 7 * @attention 8 * 9 * ch2&gt;ccenter&gt;© Copyright (c) 2019 STMicroe 10 * All rights reserved. 11 * 12 * This software component is licensed by ST unde 13 * the "License"; You may not use this file excep 14 * License. You may obtain a copy of the License 15 * 17 18 */ 19 /* USER CODE END Header */ 20 21 /* Includes 22 #include "main.h" 23 #include "main.h" 23 #include "main.h" 23 #include "main.h" 24 ************************************</pre>	electronics. er BSD 3-Clause license, pt in compliance with the at: s/BSD-3-Clause at:	Imain.h         Imain.h <t< td=""></t<>
			· summed

# Figure 41. Compare current history with local history
This opens the File Differences editor and displays the file changes.

workspace_um1 - Compare /NUCLEO-F401RE/Core	e/Src/main.c Curr	ent and Local Revision - STM32CubeIDE				- 🗆 ×
File Edit Source Refactor Navigate Search Project I	Run Window He	lp				
📑 🕶 🔚 🐚 🥸 🕶 🗞 🕶 🗟 🔍 🕵 💩 📷 💌	😚 • 🖻 • 🞯 ·	• 🔯 • 🤔 🔗 • 🔛 🗉 🖷 🐓 •	<mark>∛</mark>	=> - 📑 🚺 🗄 🛥 🎄		Quick Access 📑 🖻 💀 🚾 👪
Project Explorer 🛛 🛛 🖻 🔄 🔽 🗖	🗖 main.c	E <sup>0</sup> Compare main.c Current and Local	Revision 🖂			🔡 Outline 🖾 🛞 Build Targets 👘 🗖
ImyLib     NUCLEO-F401RE     Windows     Windows	C Compai C Transl	re lation Unit ain				There is no active editor that provides an outline.
<ul> <li>ic main.c</li> <li>ic stm32f4xx_hal_msp.c</li> <li>if stm32f4xy if c</li> </ul>	C Compa	re Viewer 🔻		M 🔁 🚭 🕸	42 🗟	
<ul> <li>≥ sunset would:</li> <li>≥ gyscalls.c</li> <li>≥</li></ul>	Local: main.c 171 /* U 172 173 /* I 174 MX_G 175 #ifdef 176 My_m 177 My_m 178 My_m 180 My_m 181 #endif 182 <	<pre>SER CODE END SysInit */ nitialize all configured peri PI0_Init(); TEST_MALLOC em3=malloc(12); em5=malloc(12); em5=malloc(12); em7=malloc(12); </pre>	Local hist 171 /* 173 /* 173 /* 174 MX 176 /* 177 M 176 /* 177 M 179 me 180 #ifd 181 me 182 me	<pre>ory: main.c Sep 20, 2019, 2:54:18 PM USER CODE END SysInit */ Initialize all configured GPIO_Init(); USART2_UART_Init(); USER CODE BEGIN 2 */ m=malloc(12); m2=malloc(12); m4=mal</pre>	~	
NUCLEO-F401RE.list NUCLEO-F401RE.map Objects.list	R Problems	🖉 Tasks 📮 Console 🛛 🔲 Properti	es 🚽	Build Analyzer 🚊 Static S     main.c	tack Ana	yzer 🔒 History 🛛 🗢 🗖 🔗 🔁 🛃 🗞 ▼ 🗐 🖨 🚰
MUCLEO-F401RE.elf.cfg     NUCLEO-F401RE.elf.cfg     NUCLEO-F401RE.elf.launch     NUCLEO-F401RE.loc     Reame.txt     STM32F401RETX_FLASH_IPCodeInFlash.ld     STM32F401RETX_FLASH_ORG.ld		- usepung of Arra Arrie.		Revision Time ■ 9/24/19, 3:51 PM ■ 9/24/19, 1:57 PM ■ 9/20/19, 2:54 PM		
Left: 185 : 1, Right: 175 : 1, incoming deletion #1 (Left:	175 : 184, Right:	before line 175)				i Status: offline

Figure 42. Compare local file differences

#### Creating and building C/C++ projects 2

As mentioned in Section 1.6 Workspaces and projects, a workspace is a directory containing projects. The first time a workspace is created, it is empty without any projects. The projects need to be created or imported in the workspace. This section contains information on how to create projects in the workspace and build projects. It also covers how to import and export projects.

#### 2.1 Introduction to projects

A project is a directory in the workspace containing files that may be organized in sub-directories. It is possible to access any project within the active workspace. The files included in a project do not need to be physically located in a folder in the project but can be located somewhere else and linked into the project. Projects located in another workspace cannot be accessed, unless the user switches to that workspace or import some of these projects into the workspace in use.

It is possible to rename and delete a project. If a workspace contains many projects, it is also possible to close some of them to make the work easier. Closed projects can be reopened again at any time.

This section focuses on the two types of STM32 projects supported by STM32CubeIDE:

- Executable programs
- Static library projects

However, the Eclipse<sup>®</sup> C/C++ Development Toolkit (CDT<sup>™</sup>), which STM32CubeIDE is based on, contains also basic project wizards, which can be used to create C managed build, C++ managed build, and makefile projects. The STM32 projects can be:

- C or C++
- Generated executable or library file
- Based on STM32Cube (using STM32 firmware library package) or empty projects

STM32 projects also support an advanced umbrella project structure, where one project contains many projects, for instance one project per core for multi-core devices.

#### 2.2 Creating a new STM32 project

#### 2.2.1 Creating a new STM32 executable project

The easiest way to create a new STM32 C/C++ project is to use the STM32 project wizard. It is selected through the menu [File]>[New STM32 Project].

Another way to create a new C/C++ project is to open the Information Center and press [Start new STM32

project]. As mentioned in Section 1.3 Information Center, the Information Center can be opened using the button on the toolbar or via the menu [Help]>[Information Center].

Both ways initialize and launch the STM32 Project Target Selection tool.

| * 🗈 🛙   | 20, C 10   
  | Features   
   
   |   | Brock Diagram   
   |  | Doos & Resources  |  
   | E Datasheet  |  |  | 🗆 Bay   |
---
---
--
--
---	---
--	--
Number Search	
  |  
   
   |   |   
   |  |   |  
   |  |  |  |   |
|   |  
  |  
   
   |   |   
   | New mu   | lticore STM32MP   | 1 Series   
   |  |  |  |   |
|   |  
  |  
   
   |   |   
   | for Indus  | trial and IoT ann   | lications  
   |  |  |  |   |
|   | 5  
  |  
   
   |   |   
   | IOI II IUUS  | indi di di di dipp  | liculions  
   |  |  |  |   |
| ge  | <u> </u>   
  |  
   
   |   |   
   |  |   |  
   |  |  |  |   |
| From 0.0 to 10.04   | ×  
  |  
   
   |   |   
   |  |   | 2  
   |  |  |  |   |
| PIDIT 0.0 10 10.54  | 10.944   
  |  
   
   |   |   
   |  | 111111  |  
   |  |  |  |   |
| om 11 to 176  |  
  |  
   
   |   |   
   |  |   |  
   |  |  |  |   |
| m From 0 to 1836  | 176<br>84 (Dytes)  
  |  
   
   |   |   
   |  |   |  
   |  |  |  |   |
|   | 16384  
  |  
   
   |   |   
   | OTIN   |   | tribution  
   |  |  |  |   |
| From 0 to 2048 (  | ((Rytas)   
  |  
   
   |   |   
   | STM  | 52MP1   | SUIDUDUIT  
   |  |  |  |   |
| From 2 to 1024 @  | 2048   
  |  
   
   |   |   
   |  |   |  
   |  |  |  |   |
|   | (Buter)  
  |  
   
   |   |   
   |  |   |  
   |  |  |  |   |
|   | (Bytes)  
  |  
   
   |   |   
   |  |   |  
   |  |  |  |   |
| From 24 to 650 (1   | (MHO)  
  |  
   
   |   |   
   |  |   |  
   |  |  |  |   |
| 1 From 24 to 650 (1   | (Bytes)<br>1024<br>(MHP)<br>830  
  |  
   
   |   |   
   |  |   | 57   
   |  |  |  |   |
| From 24 to 650 (f   | iBytes)<br>1004<br>(MHe)<br>300<br>  
  |  
   
   |   |   
   |  |   | 577  
   |  |  |  |   |
| From 24 to 650 (f<br>cod Graphic<br>solid   | 1000000)<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>100000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>10000<br>1000000   
  |  
   
   |   |   
   |  |   | 577  
   |  |  |  |   |
| From 24 to 650 (f<br>cost ciraphic<br>soble<br>lenal<br>IC 12-bit   | (Dytes)<br>(Mec)<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0  
  | PUs List 1515 licms  
   
   | + Display strifar ferrs   | Datation 2014   
   | United by SMA  |   |  
   | 6502   | 15   | [00]   |   |
| From 24 to 650 (f<br>cost Graphic<br>able<br>crail<br>C 12-bit<br>C 18-bit  | Bytes)         Model           1001         1001           1001  
  | PUS List: 1515 Items   
   
   | + Stiplay strikar forms<br>STR422000C0Tx<br>STR422000C0Tx   | Notice States   
   | 1.011010101010101000<br>0.052<br>0.052   | REU Rend CORES  | 32 (5yes)  
   | 6534<br>4 165y146  | 15<br>38   | 18 Mr2   | 0.0   |
| From 24 to 050 (1<br>cod Graphic<br>solite<br>enal<br>C 12-bit<br>C 16-bit<br>S   | Bytes)<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>100-1<br>1   
  | PUS Lat: 1515 litens<br>57402/201008<br>57442/201008<br>57442/201008   
   
   | Tuping stream forms<br>STM250000Tr<br>STM2570000Tr<br>STM2570000Tr  | Forkers) 2011<br>Active<br>Active<br>Active   
   | 0.542<br>0.657<br>0.0  | 102 004 70400<br>(J726)<br>(J726)   | 32 köyöss<br>34 köyöss<br>746 ivertaa  
   | i köytes<br>0 köytes<br>22 könna   | 15<br>23<br>39<br>37   | (10)<br>48 站地<br>48 站地   | 6.0<br>0.0<br>0.0   |
| From 24 to 050 (f<br>cod Graphic<br>solite<br>erail<br>IC 12-bit<br>IC 16-bit<br>IS<br>N<br>NUPP  | Bytes)<br>Mines  
  | PUX List 1919 litens<br>57%429403008<br>37%42940300<br>37%42940300<br>67%429403001   
   
   | Tigsty striag form     STMS2P0000Tr     STMS2P000Tr     STMS2P0000Tr     STMS2P000Tr     STMS2P000Tr     STMS2P000Tr     STMS2P000Tr     STMS2P000Tr     STMS2P000Tr     STMS2P000Tr     STMS2P000Tr     STMS2P00Tr     S   | Active<br>Active<br>Active<br>Active  | 0.642<br>0.642<br>0.657<br>1.0<br>0.386   
  | 10 000 CPP8<br>CPP8<br>CPP8<br>CPP8<br>CPP8<br>T00020   | 22 Köykes<br>32 Köykes<br>34 Köykes<br>75 Köytas<br>19 Köyöres   
   | 4 Köyles<br>0 Köyles<br>27 Köyles<br>4 Köyles  | 15<br>39<br>37<br>15   | 18 MP2<br>48 MP2<br>48 MP4<br>48 MP4<br>48 MP4   | 0.0<br>0.0<br>0.0<br>0.0  |
| From 24 to 650 (t<br>cod Graphic<br>mible<br>enal<br>C 12-bit<br>S<br>S<br>N<br>NIMP<br>REDIC   | whores         whores<   
   | PUX List: 1515 litens.<br>5 KM2/201028<br>5 TM2/201028<br>5 TM2/201074<br>5 TM2/201074<br>5 TM2/201074  
  | E Display striker Jones<br>STMA2P000C0TF<br>STMA2P000C0TF<br>STMA2P000C0TF<br>STMA2P000CFF<br>STMA2P00CFF<br>STMA2P00CFF   
  | Environa y Suri A<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active   | 0.042<br>0.642<br>0.657<br>10<br>0.086<br>0.471  
   | 201 001 0010<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00714<br>00  | 22 165/04<br>32 165/04<br>36 169/14<br>160 169/14<br>161 163/05<br>32 164/145   
  | 4 köytes<br>0 köytes<br>77 Höytes<br>4 köytes<br>4 köytes  | 15<br>38<br>39<br>77<br>15<br>25   | 48 MPc<br>48 MPc<br>48 MPc<br>48 MPc<br>48 MPc   | 0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0  |
| From 24 to 050 (t<br>cool Graphic<br>nable<br>erail<br>C 12-bit<br>S<br>S<br>N<br>NP<br>IRDIC<br>VP   | 40/ret)<br>40/ret)<br>40/ret)<br>50/ret<br>50/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60/ret<br>60  
  | PUS Lat: 1515 litens<br>57102/00029<br>57102/0029<br>57102/0029<br>57102/0044<br>57102/0048  
   
   | Tiglay share here<br>STA129050C6Tk<br>STA129000C6Tk<br>STA129000C74k<br>STA129000C74k<br>STA129000C4Tk  | Endering Start<br>Active<br>Active<br>Active<br>Active<br>Active  
   | 0.542<br>0.542<br>0.657<br>1.0<br>0.086<br>0.471<br>0.665  | A Contraction of the contraction  | 32 Hityses<br>34 Kityses<br>36 Kityses<br>36 Kityses<br>32 Kityses<br>36 Kityses<br>36 Kityses   
   | 4 X2/ytes<br>8 X2/ytes<br>3 X3/ytes<br>4 X2/ytes<br>8 X5/ytes  | 10)<br>33<br>39<br>37<br>15<br>25<br>55  | 7000<br>48 Mire<br>48 Mire<br>48 Mire<br>48 Mire<br>48 Mire<br>48 Mire   | 0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0                                   |
| From 24 to 650 (f<br>cod Graphic<br>sable<br>crail<br>C 12-bit<br>C 12-bit<br>S<br>N<br>N/P<br>RODC<br>YP<br>C 12-bit   | 40cm)<br>  
  | Pik lat 1515 kms<br>57143740026<br>57143740026<br>5714374000<br>57143740004<br>5714374004<br>5714374004<br>57143740045   
   
   | <ul> <li>Digital official form</li> <li>Digital production</li> <li>Diffusion concerns</li> </ul>   | Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active  
   | 6.62<br>6.62<br>10<br>0.636<br>0.471<br>0.665<br>1.1   | Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Correl<br>Co  | 22 18594<br>64 18595<br>76 18695<br>28 18695<br>28 18695<br>28 18695<br>28 18695<br>28 18695<br>28 18695   
   | <ul> <li>A Styles</li> <li>B Köytes</li> <li>A Köytes&lt;</li></ul>  | 15<br>38<br>39<br>37<br>15<br>23<br>55<br>55<br>51<br>31   | 46 May<br>48 May<br>48 May<br>48 May<br>48 May<br>48 May<br>48 May<br>48 May   | 0 0 0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0                   |
| From 24 to 650 (f<br>cod Graphic<br>stable<br>eral<br>C 12-bit<br>C 16-bit<br>S<br>S<br>N<br>N/<br>M/<br>M/<br>N/<br>N/<br>V/<br>P<br>C 12-bit<br>S<br>S<br>N/<br>N/<br>N/<br>N/<br>N/<br>N/<br>N/<br>N/<br>N/<br>N/<br>N/<br>N/<br>N/  | 40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)<br>40pes)  | PUS.Lot. 1515 Items<br>574.03/6105.05<br>574.03/6105.05<br>574.03/0004<br>574.03/0044<br>574.03/0044<br>574.03/0044<br>574.03/0044<br>574.03/0044  | <ul> <li>Toplay thruse heres</li> <li>Toplay concerning</li> <li>STI Last proceeding</li> <li>STI Last proceeding<td>Industry Start<br/>Active<br/>Active<br/>Active<br/>Active<br/>Active<br/>Active<br/>Active<br/>Active<br/>Active</td><td>0.642 (0.647)<br/>0.542 (0.657)<br/>1.0<br/>0.536 (0.447)<br/>0.668 (0.647)<br/>1.0<br/>0.53 (0.647)</td><td>Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         <td< td=""><td>2000<br/>32 köynes<br/>34 köynes<br/>32 köynes<br/>32 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes</td><td>4 kBytes<br/>0 kDytes<br/>77 kBytes<br/>4 kDytes<br/>8 kDytes<br/>8 kDytes<br/>0 kDytes<br/>4 kDy</td><td>15<br/>39<br/>77<br/>15<br/>55<br/>51<br/>78<br/>29<br/>20<br/>51<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20</td><td>7000<br/>48 MHz<br/>48 MHz</td><td>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0</td></td<></td></li></ul> | Industry Start<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active  | 0.642 (0.647)<br>0.542 (0.657)<br>1.0<br>0.536 (0.447)<br>0.668 (0.647)<br>1.0<br>0.53 (0.647)   | Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016 <td< td=""><td>2000<br/>32 köynes<br/>34 köynes<br/>32 köynes<br/>32 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes<br/>34 köynes</td><td>4 kBytes<br/>0 kDytes<br/>77 kBytes<br/>4 kDytes<br/>8 kDytes<br/>8 kDytes<br/>0 kDytes<br/>4 kDy</td><td>15<br/>39<br/>77<br/>15<br/>55<br/>51<br/>78<br/>29<br/>20<br/>51<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20</td><td>7000<br/>48 MHz<br/>48 MHz</td><td>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0</td></td<>  | 2000<br>32 köynes<br>34 köynes<br>32 köynes<br>32 köynes<br>34 köynes<br>34 köynes<br>34 köynes<br>34 köynes<br>34 köynes<br>34 köynes<br>34 köynes<br>34 köynes   | 4 kBytes<br>0 kDytes<br>77 kBytes<br>4 kDytes<br>8 kDytes<br>8 kDytes<br>0 kDytes<br>4 kDy   | 15<br>39<br>77<br>15<br>55<br>51<br>78<br>29<br>20<br>51<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20   | 7000<br>48 MHz<br>48 MHz   | 0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0              |
| From 24 to 650 (f<br>cod Graphic<br>uble<br>eral<br>C 12-bit<br>C 12-bit<br>S N<br>N<br>NDIP<br>RPDC<br>C 12-bit<br>M<br>RPDC<br>C 12-bit<br>RPDC<br>C 12-bit<br>RPDC<br>C 12-bit<br>RPDC<br>RPDC<br>RPDC<br>RPDC<br>RPDC<br>RPDC<br>RPDC<br>RPDC   | 40ces)<br>   
  | Pair Lot 1515 Anne           97103/2010.64           97103/2010.64           97103/2010.64           97103/2010.64           97103/2010.64           97103/2010.64           97103/2010.64           97103/2010.64           97103/2010.64           97103/2010.64           97103/2010.64           97103/2016.6           97103/2016.6   
   
   | E Stiglay timer form<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatproperty<br>Thatprop  | Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active  
   | 6.62<br>6.62<br>6.657<br>10<br>6.636<br>6.471<br>6.66<br>11<br>11<br>11<br>11<br>0.76  | Image: Section 2016   | 22 Alfyses<br>23 Alfyses<br>24 Kiljens<br>25 Ginghan<br>13 Kiljens<br>24 Kiljens<br>24 Kiljens<br>24 Kiljens<br>24 Kiljens<br>24 Kiljens<br>24 Kiljens<br>24 Kiljens<br>24 Kiljens<br>25 Ginghan<br>13 Kiljens<br>26 Kiljens<br>26 Kiljens<br>27 Kiljens<br>27 Kiljens<br>28 Kil   | 4 80%<br>9 80%<br>20 18%<br>20 18%   | 10<br>39<br>37<br>15<br>25<br>55<br>51<br>28<br>39<br>20   | 4 MPC<br>44 MPE<br>44 MPE<br>44 MPE<br>44 MPE<br>44 MPE<br>44 MPE<br>44 MPE<br>44 MPE<br>44 MPE   
  | 0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0              |
| From 24 to 6550 (f<br>add (Sraphic<br>addre<br>era)<br>C 12-bit<br>S<br>S<br>N<br>NPP<br>C 12-bit<br>NP<br>IRDIC<br>NPP<br>C 12-bit<br>NI<br>NI<br>NI<br>NI<br>NI<br>NI<br>NI<br>NI<br>NI<br>NI<br>NI<br>NI<br>NI   | Where:         Where:<   | PUS.Lot. 1515 fitms<br>574.02/00056<br>574.02/00056<br>574.02/00064<br>574.02/0044<br>574.02/0046<br>574.02/0046<br>574.02/0046<br>574.02/0046<br>574.02/0046<br>574.02/0166<br>574.02/014   | <ul> <li>Toplay strute forms</li> <li>Toplay strute forms</li> <li>STI Largeocochy</li> <li>STI Largeocochy</li></ul>   | Industry Start<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active  | 040 Point for Starl<br>0.542<br>0.63<br>0.536<br>0.471<br>0.685<br>1.1<br>0.685<br>0.03<br>0.776<br>0.776<br>0.771   | Image: Control of the contro   | 200<br>21 42/bytes<br>24 42/bytes   | 0.00<br>d 32/plats<br>D 42/plats<br>20 42/plats<br>d 42/plats<br>d 42/plats<br>20 52/plats<br>20 52/plats<br>d 42/plats<br>d 42/plat   | 15<br>23<br>29<br>27<br>25<br>55<br>51<br>25<br>51<br>28<br>29<br>20<br>15   | 4.4 May<br>4.4 May<br>4.5 May<br>4.5 May  | 6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0                            |
| Prom 24 to 650 (f<br>cod Graphic<br>able<br>crail<br>c 12-bit<br>c 12-bit<br>S<br>S<br>N<br>N<br>NPP<br>C 12-bit<br>S<br>S<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N<br>N   | Image: 1         Image: 2  
  | Pia Lat 1515 Atms<br>97102/981026<br>97102/981026<br>97102/98008<br>97102/9804<br>97102/9804<br>97102/9804<br>97102/9804<br>97102/9804<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/9814<br>97102/97102<br>97102/97102<br>97102/97102<br>971   
   | These operations have<br>strategy operations and<br>strategy operating<br>strategy operating  
   | Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active  | 6.50 11110122810<br>0.657<br>0.637<br>0.636<br>0.636<br>0.636<br>0.11<br>0.636<br>0.716<br>0.716<br>0.716<br>0.716   | Image         Image         Image         Image           Image         Image         Image         Image         Image           Image </td <td>22 249yas<br/>84 459yas<br/>705 1995<br/>84 559yas<br/>84 559yas<br/>84 559yas<br/>84 559yas<br/>82 2459yas<br/>82 2459yas<br/>83 259yas<br/>93 1959yas<br/>94 1959yas<br/>94 1959yas<br/>94 1959yas<br/>94 1959yas</td>
<td>6.00<br/>6.00/pts<br/>32.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts<br/>4.00/pts</td> <td>10<br/>23<br/>25<br/>25<br/>55<br/>51<br/>25<br/>25<br/>51<br/>28<br/>29<br/>29<br/>29<br/>20<br/>15<br/>15</td> <td>48 Mile<br/>44 Mile<br/>44 Mile<br/>48 Mile</td> <td>6.0<br/>6.0<br/>6.0<br/>6.0<br/>6.0<br/>6.0<br/>6.0<br/>6.0<br/>6.0<br/>6.0</td>   | 22 249yas<br>84 459yas<br>705 1995<br>84 559yas<br>84 559yas<br>84 559yas<br>84 559yas<br>82 2459yas<br>82 2459yas<br>83 259yas<br>93 1959yas<br>94 1959yas<br>94 1959yas<br>94 1959yas<br>94 1959yas  | 6.00<br>6.00/pts<br>32.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts<br>4.00/pts   
  | 10<br>23<br>25<br>25<br>55<br>51<br>25<br>25<br>51<br>28<br>29<br>29<br>29<br>20<br>15<br>15   | 48 Mile<br>44 Mile<br>44 Mile<br>48 Mile  | 6.0<br>6.0<br>6.0<br>6.0<br>6.0<br>6.0<br>6.0<br>6.0<br>6.0<br>6.0              | | | | | | | | | |
| Prem 24 to 650 (f<br>acc Graphic<br>nitite<br>enal<br>to 12-bit<br>5<br>NM<br>NP<br>NP<br>NP<br>NP<br>NP<br>C 12-bit<br>NP<br>NP<br>DUG<br>SDM<br>HK05T<br>serent   | Work:         Work: <th< td=""><td>Fill Lizt         1510 States           FTA12/States         Silva2/States           Silva2/States         Silva2/States</td><td>Display in the Meeter     STA1299000179     STA1299000179     STA1299000179     STA12990000749     STA12990000479     STA1299000479     STA1299000479     STA1299001479     STA129900     STA129900147     STA129900147     STA129900147      STA</td><td>Tolony) teri<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv<br/>Artiv</td><td>Define at a lot           6.42         6.62           6.64         6.64           6.64         6.65           6.65         1.1           6.65         6.65           6.76         6.76           6.76         6.76           6.73         -</td><td></td><td>22 Hojne<br/>23 Hojne<br/>24 Hojne<br/>25 Hojne<br/>25 Hojne<br/>25 Hojne<br/>25 Hojne<br/>26 Hojne<br/>26 Hojne<br/>26 Hojne<br/>27 Hojne<br/>28 Hojne<br/>29 Hojne<br/>20 Hojne<br/>21 Hojne<br/>21 Hojne<br/>22 Hojne<br/>23 Hojne<br/>24 Hojne<br/>25 Hojne<br/>26 Hojne<br/>26 Hojne<br/>27 Hojne<br/>27 Hojne<br/>28 Hojne<br/>29 Hojne<br/>20 Hojne<br/>21 Hojne<br/>21</td><td>500<br/>6 kSylas<br/>8 kSylas<br/>6 kSylas<br/>8 kSylas<br/>8 kSylas<br/>9 kSylas<br/>9 kSylas<br/>6 kSylas<br/>7 kSyla</td><td>15<br/>39<br/>15<br/>55<br/>51<br/>20<br/>20<br/>20<br/>20<br/>15<br/>15<br/>21<br/>21<br/>21<br/>21</td><td>43 Mile<br/>44 Mile<br/>44 Mile<br/>45 Mile<br/>44 Mile</td><td>6 0<br/>6 0<br/>6 0<br/>6 0<br/>6 0<br/>6 0<br/>6 0<br/>6 0<br/>6 0<br/>6 0</td></th<> | Fill Lizt         1510 States           FTA12/States         Silva2/States           Silva2/States         Silva2/States   | Display in the Meeter     STA1299000179     STA1299000179     STA1299000179     STA12990000749     STA12990000479     STA1299000479     STA1299000479     STA1299001479     STA129900     STA129900147     STA129900147     STA129900147      STA   | Tolony) teri<br>Artiv<br>Artiv<br>Artiv<br>Artiv<br>Artiv<br>Artiv<br>Artiv<br>Artiv<br>Artiv<br>Artiv<br>Artiv<br>Artiv<br>Artiv   | Define at a lot           6.42         6.62           6.64         6.64           6.64         6.65           6.65         1.1           6.65         6.65           6.76         6.76           6.76         6.76           6.73         -  |   | 22 Hojne<br>23 Hojne<br>24 Hojne<br>25 Hojne<br>25 Hojne<br>25 Hojne<br>25 Hojne<br>26 Hojne<br>26 Hojne<br>26 Hojne<br>27 Hojne<br>28 Hojne<br>29 Hojne<br>20 Hojne<br>21 Hojne<br>21 Hojne<br>22 Hojne<br>23 Hojne<br>24 Hojne<br>25 Hojne<br>26 Hojne<br>26 Hojne<br>27 Hojne<br>27 Hojne<br>28 Hojne<br>29 Hojne<br>20 Hojne<br>21 | 500<br>6 kSylas<br>8 kSylas<br>6 kSylas<br>8 kSylas<br>8 kSylas<br>9 kSylas<br>9 kSylas<br>6 kSylas<br>7 kSyla | 15<br>39<br>15<br>55<br>51<br>20<br>20<br>20<br>20<br>15<br>15<br>21<br>21<br>21<br>21   | 43 Mile<br>44 Mile<br>44 Mile<br>45 Mile<br>44 Mile   | 6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0              |
| Prem 24 to 050 (t<br>cod Graphic<br>wildle<br>Cr 12-bit<br>Cr 12-bit<br>S<br>N<br>NHP<br>HIGHC<br>VP<br>Cr 12-bit<br>NHP<br>HIGHC<br>Cr 12-bit<br>NHP<br>HIGHC<br>DLG<br>SDM<br>HIGHT<br>HIGHT<br>CAN   | Uben)   | PLA Lat 1515 terms<br>51102/00029<br>51102/00029<br>51102/00024<br>51102/00044<br>51102/00044<br>51102/00045<br>51102/00045<br>51102/00045<br>51102/0016<br>51102/016  | TRADY IN FACTOR DESIGN<br>TRADY DESCRIPTION<br>TRADY D   | Action Students (and<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Ac  | 6.92 (b) (c) (c) (c) (c)<br>(c) (c)<br>(c)<br>(c)<br>(c)<br>(c)<br>(c)<br>(c)<br>(c)<br>(c)<br>(c)   | Image: Control Control         Image: Control           Image: Control         Control         Control           Image: Contro  | 22 Allyses<br>24 Allyses<br>26 Kilyses<br>26 Kilyses<br>26 Kilyses<br>28 Kilyses<br>28 Kilyses<br>28 Kilyses<br>28 Kilyses<br>29 Kilyses<br>29 Kilyses<br>20 Kil   | 4 800066 500<br>0 420065<br>20 420065<br>20 420066<br>20 42006<br>20 4006<br>20 40000<br>20 40000   | 10<br>33<br>39<br>37<br>15<br>25<br>55<br>51<br>30<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>30<br>30<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20   | 48 MPc<br>48 | 6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0<br>6 0              |
| Prem 24 to 550 (f<br>coo Graphic<br>antike<br>erail<br>0 12-bit<br>0 12-  | Oben         X           0         0  | Tell and 1011 Auror           5714/2000/2 <td< td=""><td>THASPOROSIN<br/>STRASPOROSIN<br/>THASPOROSIN<br/>THASPOROSIN<br/>THASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPOROSIN<br/>STRASPORO</td><td>Tobardy Set C<br/>Alfan<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afate<br/>Afat</td><td>641000 (2.93)<br/>6.42<br/>6.62<br/>6.62<br/>6.44<br/>1.6<br/>6.63<br/>6.44<br/>1.6<br/>6.7<br/>1.6<br/>6.7<br/>1.7<br/>1.7<br/>1.7<br/>1.7<br/>1.7<br/>1.7<br/>1.7<br/>1</td><td>Image: State State</td><td>22 45/min<br/>24 45/min<br/>24 45/min<br/>24 45/min<br/>25 45/min<br/>25 45/min<br/>25 45/min<br/>25 45/min<br/>25 45/min<br/>25 45/min<br/>25 45/min<br/>26 45/</td><td>200     2</td><td>10<br/>39<br/>39<br/>39<br/>30<br/>35<br/>55<br/>55<br/>55<br/>55<br/>55<br/>55<br/>55<br/>55<br/>55<br/>55<br/>55<br/>55</td><td>4.4 Mag<br/>4.4 Ma</td><td>8.6<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0.0<br/>0</td></td<> | THASPOROSIN<br>STRASPOROSIN<br>THASPOROSIN<br>THASPOROSIN<br>THASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPOROSIN<br>STRASPORO   | Tobardy Set C<br>Alfan<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afate<br>Afat | 641000 (2.93)<br>6.42<br>6.62<br>6.62<br>6.44<br>1.6<br>6.63<br>6.44<br>1.6<br>6.7<br>1.6<br>6.7<br>1.7<br>1.7<br>1.7<br>1.7<br>1.7<br>1.7<br>1.7<br>1   | Image: State  | 22 45/min<br>24 45/min<br>24 45/min<br>24 45/min<br>25 45/min<br>25 45/min<br>25 45/min<br>25 45/min<br>25 45/min<br>25 45/min<br>25 45/min<br>26 45/   | 200     2  | 10<br>39<br>39<br>39<br>30<br>35<br>55<br>55<br>55<br>55<br>55<br>55<br>55<br>55<br>55<br>55<br>55<br>55   | 4.4 Mag<br>4.4 Ma  | 8.6<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0.0<br>0         |
| Prem 24 to 050 (t<br>coo Graphic<br>white<br>coal coal<br>c 14 bit<br>c 14 bit<br>S 0 12 bit<br>M MP<br>C 14 bit<br>M MP<br>MOIC<br>PVP<br>C 12 bit<br>M MOIC<br>PVP<br>C 12 bit<br>M<br>M<br>M<br>M<br>M<br>M<br>M<br>M<br>M<br>M<br>M<br>M<br>M<br>M<br>C<br>C<br>C   |   | Total USE Imm           Total VISE Imm  | Tablety strend here:<br>Stream Stream S  | Exhang to of<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Action<br>Aco   | 0419-047-047<br>6.627<br>7.6<br>7.6<br>6.664<br>1.1<br>1.6<br>6.664<br>1.1<br>1.6<br>7.7<br>1.6<br>6.7<br>7.7<br>1.6<br>6.7<br>7.7<br>1.6<br>6.7<br>7.7<br>1.6<br>6.7<br>7.7<br>1.6<br>6.7<br>7.7<br>1.6<br>6.7<br>7.7<br>7.7<br>7.7<br>7.7<br>7.7<br>7.7<br>7.7<br>7.7<br>7   | Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016         Image: Section 2016           Image: Section 2016         Image: Section 2016 <td< td=""><td>21 249yee<br/>22 249yee<br/>23 249yee<br/>25 249yee<br/>26 249</td><td>4 800066 1000<br/>8 550055<br/>20 55005<br/>20 550</td><td>78<br/>28<br/>29<br/>77<br/>25<br/>55<br/>55<br/>54<br/>51<br/>55<br/>59<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20<br/>20</td><td>Free           48 MHz           48 MHz</td><td>6,0<br/>6,0<br/>6,0<br/>6,0<br/>6,0<br/>6,0<br/>6,0<br/>6,0<br/>6,0<br/>6,0</td></td<> | 21 249yee<br>22 249yee<br>23 249yee<br>25 249yee<br>26 249   | 4 800066 1000<br>8 550055<br>20 55005<br>20 550   | 78<br>28<br>29<br>77<br>25<br>55<br>55<br>54<br>51<br>55<br>59<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20<br>20   | Free           48 MHz  | 6,0<br>6,0<br>6,0<br>6,0<br>6,0<br>6,0<br>6,0<br>6,0<br>6,0<br>6,0              |
| Prim 24 to 550 (f<br>cool Graphic<br>staff<br>erail<br>to 12-bit<br>to   |   | Pa (ar 191) Imm<br>5740 Yan Yangar<br>5740 Yangar<br>1440 Yangar<br>5440 Yanga   | Balaka vintue famili<br>sinuara toxocini<br>mataratoxocini<br>sinuara toxocini<br>sinuara toxocini<br>sinu  | Active Control of Active<br>Active<br>Article<br>Article<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Active<br>Act   | 6.942<br>6.942<br>6.947<br>1.0<br>6.645<br>1.0<br>7.0<br>7.0<br>7.0<br>7.0<br>7.0<br>7.0<br>7.0<br>7.0<br>7.0<br>7   | No.         Cont         Annu           Image: Cont of the state of  | 22 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2   | 0.00<br>0.00 percent<br>0.00 perce   | 15<br>33<br>37<br>37<br>35<br>35<br>31<br>35<br>31<br>33<br>30<br>30<br>30<br>30<br>31<br>31<br>31<br>32<br>32<br>32<br>32<br>32<br>32<br>32<br>33<br>33<br>33<br>34<br>34<br>35<br>35<br>35<br>35<br>35<br>35<br>35<br>35<br>35<br>35<br>35<br>35<br>35 | 16         Max           4.8         Max           4.8         Max           4.4         Max           4.4         Max           4.4         Max           4.5         Max           4.6         Max   | 0 0<br>0 0<br>0 0<br>0 0<br>0 0<br>0 0<br>0 0<br>0 0<br>0 0<br>0 0              |
| Priori 24 to 650 (5<br>con Graphic<br>make<br>con 1<br>con 2<br>con 2 |   | *** Lat 100 km<br>5************************************  | Provide control for<br>and and records in<br>the and records in<br>the<br>and and in the<br>and and records in<br>the<br>and and records in<br>the<br>and and in the<br>and and in the<br>and and in the<br>and and in the<br>and and and and<br>and<br>and<br>and<br>and<br>a   | Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Artine<br>Ar  | 0410-042440         6.62         6.63         6.75 | Image: Section of the sectio   | 2 Constant<br>2 Cons   | 100     1  | 0<br>33<br>39<br>37<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>39<br>39  | 44 Mine<br>44 Min  | 60<br>60<br>60<br>60<br>60<br>60<br>60<br>60<br>60<br>60<br>60<br>60<br>60<br>6 |
| Prior 24 to 650 (*<br>secol Graphic<br>milde<br>single<br>26 12 abs<br>25 5<br>30 12 12 12 12<br>30 12 12 12 12<br>30 12 12 12 12<br>30 12 12 12<br>30 12 12 12<br>30 12<br>30<br>30 12<br>30  |   | Pill (2011) Brance         Image: Second   | Palety share here<br>shared sectors and<br>management of the sectors<br>management of the sectors<br>management of the<br>management of the<br>management of the<br>management of the<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>management<br>mana   | A Call Robust of a constraint   | 6.00         Not 100 00 00 00           0.00         0  | Image: Control of Con   | 22 2 2 99 m<br>24 2 99 m<br>25 2 99 m<br>25 2 99 m<br>26 2 99 m<br>27 2 99 m   | 4 400 doi:<br>10 400   | 10<br>30<br>30<br>30<br>30<br>30<br>30<br>30<br>30<br>30<br>30<br>30<br>30<br>30   | 48 Mine<br>48 Mine<br>49 Mine<br>44  | 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   |

#### Figure 43. STM32 target selection

The *MCU/MPU selector* and *Board Selector* tabs can be selected at the top of the window. Use the first tab to create project for a specific device and the second if a project for a specific board is needed.

This section presents the creation of a project for the NUCLEO-F401RE board using the Board Selector.

Among the different filters available for use on the left of the window, type "401" in the *Part Number Search* field to filter the boards with names containing this string. In Figure 44, two boards are listed, a Nucleo board board and a Discovery board. The NUCLEO-F401RE board is selected.

### Figure 44. STM32 board selection

	elector							
തം ലം ാ	Fta	lures	Large P	icture	Docs & Resources		Datasheet	🖪 Bay
warch ~								
~	V NOLLEO-PAPIRE	-		STM croslectronics NUCLEO #401RE Bo	rd Support and Examples			
				larger land			THE COLORS	
				Patrick In Independent			PHONE (2020)202	
8 ho 13 0 13 n 13 n 15 mm 0 ho 8 (MHz) 5er 0 1 1 m Foster 0 0 n 0 0 Dut 0 1		No offer a first second		n na seran na na hana na	SUDITI AT LASS TRACINGUMANI ANNA			
8 2 0 0								
n Factor 0 0 0 108	Boards List: 2 kems							
m Factor 0 0 0 108	Boards List: 2 kerns	Doctored States Water	NUCLEO-F401RE	Hall Ite -	Naclas64	Monday) Artha	ido Uni treo Al 13.8	141 Microbiol The STMTER REPORT of
B 2     Comparison of the sector of the	Boards List: 2 Rems		NUCLEO-PROTRE STM32F4010-DISCO	Parts -	nee Nacionifi Cocovery M	i Koong	138 2.0	B State of S
B 2 0 m Factor 0 0 0 108 (Oraphics) 0 0 (Oraphics) 0 0 (Oraphics) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Baards List: 2 Herror c	Denter Militär Militär	NUCLEO-FAOIRE STMS2FAOIC-DISCO	Tel Is."	Non Ristadi Discovey ki	ausara y	ula i dan kara di 138 8.0	et en anteres anteres En anteres anteres En anteres
8         2           9         0           9         100           9         100           9         100           9         100           9         0           9         0           9         0           9         0           9         0           9         0           9         0           9         0	Daarde Lint: 2 Ameri o	Anter Marine Man	NUCLEO-FIOTRE STI432F401C-DISCO	Del Ref	Nacio34 Nacio34 Cascorry II	Antoni ay	500 (1000) 138 88	10 Benefit in State of State State of State of State State of State
n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n           n         n         n         n	Danish Lin: 2 Imm r r		NUCLEO-FROME STM32F401C DISCO	Tento -	ver Nackadri Discovej ki	assing: Actha	000 000 000 000 000 000 000 000 000 00	et e source a source
0         2           m f actor         3         0           0         108         108           0         0         0           0         0         0           0         0         0           0         0         0           0         0         0           0         0         0           0         0         0           0         0         0           0         0         0           0         0         1           0         0         1           0         0         1           0         0         1           0         0         1           0         0         1           0         0         1           0         0         1           0         0         1           0         1         1           0         0         1           0         1         1	Banki Liti: 2 Jame r v		MULTEOLFOIRE STM32F4VIC DISCO	Det S -	Noticely U	Atha	aan bortanad 138 2.6	Al Executive Sector of Constraints Sector of Constraints (Incurse Vector)
a fatter a b c c c c c c c c c c c c c c c c c c	Darok Lie: 2 here r		NUCLEOFICIARE STINISTROIC DISCO	NC.	Instandel Discovery kil	Active	ana bo tha and 518 8.0	nte en

Five tabs, *Features*, *Large Pictures*, *Docs & Resources*, *Datasheet*, and *Buy*, offer the possibility to display detailed information about the selected board or device. For instance, documentation available for the board is displayed and can be opened when *Docs & Resources* is selected. When *Datasheet* is selected, the board datasheet is downloaded from STMicroelectronics web site.

Pressing [Next] when the NUCLEO-F401RE board is selected opens the Project setup page.

Enter a project name and select the desired setting for the project in the dialog boxes. The project named *"NUCLEO-F401RE"* is filled in as an example in Figure 45.

DE STM32 Proje	ct		
Project Setup			IDE
Setup STM32 p	project		
Project Name:	NUCLEO-F401RE		
🗹 Use default	location		
Location:	C:/Users/johansse/STM32CubeIDE/workspace_um	I	Browse
Options Targeted Lan C C C+ Targeted Bin Executal Targeted Pro STM32C	nguage + nary Type ole O Static Library oject Type Cube O Empty		
?	< Back Next > Finish		Cancel

Figure 45. Project setup

According to the settings in Figure 45, the project is meant to be stored in the default location with the following options set:

- C project
- Executable binary type
- STM32Cube targeted project type

Press [Next] to open the Firmware Library Package Setup page.

Figure 46.	Firmware	library	package	setup
------------	----------	---------	---------	-------

IDE STM32 Project — — — X
Firmware Library Package Setup
Setup STM32 target's firmware
Target and Firmware Package
Target Reference: NUCLEO-F401RE
Firmware Package Name and Version: STM32Cube FW_F4 $\bigcirc$ V1.24.1 $\checkmark$
Firmware package Repository Location: C:\Users\johansse\STM32Cube\Repository
See <u>'Firmware Updater'</u> for settings related to firmware package installation
Code Generator Options O Add necessary library files as reference in the toolchain project configuration file O Copy all used libraries into the project folder O Copy only the necessary library files
Cancel       < Back

In this page, it is possible to select the STM32Cube firmware package to use when creating the project. In this case, the default settings are used. Press [**Finish**] to create the project. As a result, the following dialog is displayed.



DE Board	Project Options:		×
?	Initialize all peripherals with their default Mode ?		
		Yes	No

Press **[Yes]** since it is a good practice to get the software needed to initialize the peripherals. This opens the new dialog shown in Figure 48.

### Figure 48. STM32CubeMX perspective opening

DE Open	Associated Perspective?		×
?	This kind of project is associated with the STM32CubeM perspective now?	x perspective. Do you	u want to open this
Rem	ember my decision	Yes	No

Opening the *STM32CubeMX* perspective is a good decision if there are any needs to configure the device. Enable [**Remember my decision**] if the question msut not be asked the next time a new project is created. Press [**Yes**] to continue.

At this point, the project creation starts. The time it takes depends on the amount of files that need to be downloaded to create the project.

irmware Lik	orary Package Setup				Л
Setup STM32	target's firmware				
Target and Fir	mware Package				
Target Refere	nce:	NUCLEO-F401RE			
Firmware Pac	kage Name and Version	: STM32Cube FW_F4 V1.24.1 ~			
Firmware pac	kage Repository				
Location:					
C:\Users\joha	ansse\STM32Cube\Repo	sitory			
C:\Users\joha	ansse\STM32Cube\Repo <u>re Updater'</u> for settings	sitory related to firmware package install	lation		
C:\Users\joha See <u>'Firmwa</u>	ansse\STM32Cube\Repc <u>re Updater'</u> for settings	sitory related to firmware package install	lation		
C:\Users\joha See <u>'Firmwa</u> Code Generat	ansse\STM32Cube\Repc <u>re Updater'</u> for settings <b>cor Options</b>	sitory related to firmware package install	lation		
C:\Users\joha See <u>'Firmwar</u> Code Generat	ansse\STM32Cube\Repo re Updater' for settings tor Options ssary library files as refer	related to firmware package install rence in the toolchain project conf	lation	n file	
C:\Users\joha See <u>'Firmwa</u> Code Generat O Add neces O Copy all u	ansse\STM32Cube\Repo re Updater' for settings cor Options ssary library files as refer sed libraries into the pro	related to firmware package install rence in the toolchain project conf oject folder	lation	n file	
C:\Users\joha See <u>'Firmwa</u> Code Generat O Add neces O Copy all u O Copy only	ansse\STM32Cube\Repo re Updater' for settings tor Options ssary library files as refer sed libraries into the pro the necessary library fil	related to firmware package install rence in the toolchain project conf oject folder es	lation	n file	
C:\Users\joha See <u>'Firmwa</u> Code Generat O Add neces O Copy all u O Copy only	ansse\STM32Cube\Repo re Updater' for settings for Options ssary library files as refer sed libraries into the pro- the necessary library fil	related to firmware package install rence in the toolchain project conf oject folder es	lation	n file	
C:\Users\joha See <u>'Firmwa</u> Code Generat O Add neces O Copy all u O Copy only erform Project	ansse\STM32Cube\Repo re Updater' for settings for Options ssary library files as refer sed libraries into the pro the necessary library fil c Creation. Please Wait Fo	related to firmware package install rence in the toolchain project conf oject folder es or Completion	lation	n file	
C:\Users\joha See <u>'Firmwa</u> Ode Generat O Add neces O Copy all u O Copy only erform Project	ansse\STM32Cube\Repo re Updater' for settings for Options ssary library files as refer sed libraries into the pro the necessary library fil Creation. Please Wait Fo	related to firmware package install rence in the toolchain project conf oject folder es or Completion	lation	n file	
C:\Users\joha See <u>'Firmwa</u> Odd Generat O Add neces O Copy all u O Copy only erform Project	ansse\STM32Cube\Repo re Updater' for settings cor Options ssary library files as refer sed libraries into the pro the necessary library fil Creation. Please Wait Fo	related to firmware package install rence in the toolchain project conf oject folder es or Completion	iguratio	n file	

## Figure 49. Project creation started

When the project is created, the *STM32CubeMX* perspective is opened with a window for configuring the peripherals, clock, middleware, and power consumption.

## Figure 50. STM32CubeMX



The new project is listed in the *Project Explorer* view with some of the folders and files it contains.

The NUCLEO-F401RE.ioc file contains the configuration settings and is opened in the STM32CubeMX editor. This editor contains tabs for *Pinout & configuration, Clock configuration, Project manager* and *Tools*. When changes are made in the STM32CubeMX editor, the .ioc file in the tab is marked as changed. If the file is saved, a dialog opens asking *"Do you want to generate Code?"*, making it easy to generate new code in the project that supports the new device configuration. For more information on how to use the STM32CubeMX editor, refer to [ST-14].

It is possible to create an STM32 project with less files and folders by selecting the targeted project type [Empty] instead of [STM32Cube] (refer to Figure 45. Project setup). When [Empty] is selected, the generated project only contains some folders, a device startup file with Reset\_Handler code and vector table, the main.c file, and some other c files and linker script files. STM32 header files, system files and CMSIS files must be added manually. These files can for instance be copied from some other STM32Cube targeted project or from an STM32 example project.

Note: For empty projects, make sure to configure the floating-point unit setting to use software FPU or hardware FPU according to application requirements. When using hardware FPU, initialize the FPU. For non-empty projects, the initialization of the FPU is normally done in the SystemInit function in file system\_stm32fxxx.c. To notify that the FPU configuration may be needed, the main.c file created in an empty project contains a compiler warning stating #warning "FPU is not initialized, but the project is compiling for an FPU. Please initialize the FPU before use."

## 2.2.2 Creating a new STM32 static library project

The method described in Section 2.2.1 Creating a new STM32 executable project can be used also to create a static library project. However, static libraries are often reused in multiple application projects, possibly targeting different STM32 products. The STM32 project wizard does not support changing the MCU device. The standard  $Eclipse^{\$}/CDT^{TM}$  projects solution with some STM32 related extensions does support changing the MCU device per build configuration.

The recommended way to create static library projects is therefore to rely on Eclipse<sup>®</sup>/CDT<sup>™</sup> projects as described below.



To create an Eclipse<sup>®</sup>/CDT<sup>™</sup>, go to [File]>[New]>[C/C++ Project]. This opens the window displayed in Figure 51.

New C/C++ Project	_		$\times$
Templates for New C/C++ Proj	ect		
All Make	<b>C Managed Build</b> A C Project build using the CDT's managed build system.		
G	<b>C++ Managed Build</b> A C++ Project build using the CDT's managed build system.		
G	Makefile Project (Experimental) Create a new project that builds with the make' build tool using CDT's new Core Build System	n.	
0	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel	

Figure 51. New C/C++ project

Select either *C Managed Build* or *C++ Managed Build* depending on what the project requires and click on [Next]. This brings up the project type selector. The *Empty Project* type is the only type supporting the *MCU ARM GCC* toolchain. Make sure to select *Empty Project* under the *Executable* folder and then select the *MCU ARM GCC* toolchain as seen in Figure 52.



# Once the project naming and type selection are done, click on [Next].

Figure 5	2. Project	type
----------	------------	------

IDE			$\times$
C Project			T)
Create C project of selected type			
Project name: myLib			
✓ Use default location			
Location: C:\Users\girdlanm\STM32CubelDE\workspace_1.7.0\myLib		Brows	e
Choose file system: default \vee			
Project type:	Toolchains:		
<ul> <li>Executable         <ul> <li>Empty Project</li> <li>Hello World ANSI C Project</li> <li>Shared Library</li> <li>Static Library</li> <li>Makefile project</li> </ul> </li> <li>Makefile project</li> </ul>	Cross GCC Cygwin GCC MCU ARM GCC		
0	< Back Next > Einish	Cance	i

This launches a standard Eclipse<sup>®</sup> project configuration window as shown in Figure 53 Click on [Next]..

## Figure 53. Project configuration selection

IDE		— 🗆 X
Select Configurations Select platforms and configurations you wish to deploy on		Ď
Project type: Executable Toolchains: MCU ARM GCC Configurations:		
☑ ☜ Debug ☑ ☜ Release		Select all Deselect all
Use "Advanced settings" button to edit project's properties.		Advanced settings
Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on property pages.		
0	< <u>B</u> ack <u>N</u> ext > Einish	Cancel

In the target selector screen shown in Figure 54, make sure to select the appropriate target by clicking on the [Select...] button and filtering the correct target for the project. The target selector helps to set the -mcpu=cortex-mX toolchain flag correctly in the already defined build configurations seen in the previous step (see Figure 53).

# Figure 54. Project default target selector

IDE					$\times$
Select default target for the project					->
The selected target can be changed, per build configuration, later					
					Culuat
					Select
					~
Core:					~
0	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Canc	el



When clicking on [Select...], the filter dialog shown in Figure 55 shows up, allowing users to filter and select the correct device.

• •			
MCU/MPU Selector			$\times$
Please select your STM32 target device			
STM32F407			
MCU/MPU			
STM32F407IEHx			
STM32F407IETx			
STM32F407IGHx			
STM32F407IGTx			
STM32F407VETx			
STM32F407VGTx		 	
STM32F407ZETx			
STM32F407ZGTx			
?	ОК	Cance	į

# Figure 55. Project MCU/MPU selector



After the target selection (*MCU*), the *CPU* and *Core* fields are automatically populated in the simple single core case as shown in Figure 56.

IDE			$\times$
Selec	t default target for the project		->
The s	elected target can be changed, per build configuration, later	1	
MCU			alast
WCU:	511452F407VG1X		select
CPU:	Cortex-M4 (0)		~
Core:	0		~
(?)	< Back Next > Finish	Cancel	
	Terrer Turner	carreet	

For more advanced devices such as multi-core STM32H7 microcontrollers, the user must select the *CPU* and *Core* that the project targets to make sure that the code is built correctly. These settings are also used later to properly setup debug configurations. Make sure that the settings are as needed for the project and click on **[Finish]**.

#### Figure 57. Project target selection (advanced)

IDE						) ×
Selec	t default target for the project					Ď
The s	elected target can be changed, per build configuration, la	ter				
MCU:	STM32H745ZITx					Select
CPU:						~
Core:	Cortex-M7 (0) Cortex-M4 (1)					
(?)		< Back	Next >	Finish	Ca	ncel
		. Tares	TTOM >	And Court 1		

#### Figure 56. Project target selection

After the project creation, it is possible to create different build configurations for different targets as described in Section 2.3.1.2 . After a new build configuration is created, right-click the project in the *Project Explorer*, go to [**Properties**]>[**C/C++ Build**]>[**Settings**]>[**Tool Settings**]>[**MCU Settings**], and click on [**Select...**] to select a new target for the specific build configuration.

Properties for myl	ib			$ \Box$ $\times$
type filter text	Settings			⇔ ▼ ⇔ ▼ 🖇
Image: Properties for myLib         type filter text         > Resource Builders         * C/C++ Build Build Variables Environment Logging Settings Tool Chain Edit         > C/C++ General Project Natures Project Reference: Run/Debug Settir         * MCU Toolchain MCU Settings         > MCU OCC Assembler @ General Debugging @ Preprocessor @ Include paths @ Miscellaneous         * MCU GCC Compiler @ General @ Debugging @ Preprocessor @ Include paths @ Miscellaneous         * MCU GCC Linker @ General @ Debugging         @ MCU GCC Linker @ General         @ Miscellaneous         * MCU GCC Linker         @ General         @ Miscellaneous         * MCU GCC Linker         @ General         @ Miscellaneous		Build Artifact Build Artifact MCU CPU Core Floating-point unit Floating-point ABI Instruction set Runtime library Use float with pr Use float with sc	Arry Parsers  Frror Parsers  STM32F407VGTx  Cortex-M4 (0)  None  Software implementation (-mfloat-a Thumb2  Reduced C (specs=nano.specs) intf from newlib-nano (-u_printf_floa anf from newlib-nano (-u_scanf float)	
		Use float with sc	ant from newlib-nano (-u _scant_float	)
< >			Restore D	efaults <u>Apply</u>
?			Apply and C	lose Cancel

#### Figure 58. Project target change

Tip:

47/

CDT<sup>™</sup> projects can also be used to produce target independent applications or executable projects.

# 2.3 Configure the project build setting

When an STM32 project is created, it contains default C/C++ build settings for the project. There are however a lot of different options that can be used by GCC, each embedded system having its own requirements. It is therefore possible to configure the project build settings further than the default build settings.

It is also common to have different requirements on build settings during different phases of the project development; for instance during the debugging and release phases. To handle this, different build configurations for each project are supported by STM32CubeIDE. This section presents the build configurations first, and then the project build settings.

## 2.3.1 Project build configuration

Each build configuration allows different variants of a project and contains a specific build setting. When an STM32 project is created in STM32CubeIDE, two build configurations, *Debug* and *Release*, are created by default. The *Debug* configuration makes the project built with debug information and without any optimization. The *Release* configuration makes the project optimized for smaller code size and with no debug information. By default, the *Debug* configuration is set as the active build configuration when the project is created.

It is possible to create new build configurations for a project at any time. Such new build configuration can be based on an earlier available build configuration.

When building the project, the active build configuration is used and during build the files generated are written into a folder with the same name as the active build configuration.

Note: The build configuration only handles the build settings. How to configure debug settings is described later in this manual.

#### 2.3.1.1 Change the active build configuration

To change the active build configuration:

- 1. Select the project name in the *Project Explorer*
- 2. Use the toolbar in the C/C++ perspective and click on the arrow to the right of the [Build] toolbar button
- 3. The build configurations are listed

Select the build configuration to use from the list.

## Figure 59. Set the active build configuration using the toolbar

workspace\_um - myLib/Src/myLib.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help



Another way to change the active build configuration is to right-click on the project name in the *Project Explorer* view, select [**Build Configurations**]>[**Set Active**], and select the preferred build configuration.

# Figure 60. Set active build configuration using right-click

workspace\_um - myLib/Src/myLib.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

	🛞 🔻 🐔 👻 🔜 (	७∶৫ • 😂 • € •	6 • 🔅 • 🍅 🔗 •
Project Exp	lorer 🖾		
<ul> <li>✓ IDE myLib</li> <li>&gt; Includ</li> <li>&gt; Includ</li> <li>&gt; Includ</li> <li>&gt; Includ</li> <li>&gt; Inc</li> <li>✓ Inc</li> <li></li></ul>	ves des yLib.c scalls.c smem.c		
> 🎇 Bina	New	>	
> 🚮 Inclu	Go Into		
V 🚰 Core	Open in New Window		
> 🗁 s	Copy	Ctrl+C	
> 🤁 S		CtrI+V Delete	
🗸 🔁 Driv			
> 🔁 C	Move		
> 🤁 S	Rename	F2	
	└── Import └── Export		
<u>चि</u> STM	Build Project Clean Project Refresh Close Project Close Unrelated Projects	F5	
	Build Configurations	✓ 1 Debug	Set Active >
	Build Targets	2 Release	Manage
	Index	>	Build All
	Show in Remote System:	s view	Clean All
	Kun As		Build Selected
	Profile As	>	
	Team	>	
	Compare With	>	
	Restore from Local Histo	ry	



It is also possible to select the active build configurations using the menu [**Project**]>[**Build Configurations**]>[**Set Active**] and select the chosen build configuration.

# Figure 61. Set active build configuration using menu

workspace\_um - myLib/Src/myLib.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

<b>⊡ - 8 6</b>		Open Project Close Project		• 🥎 • 🕸 • 🕲 • 🕤 • 🛍
Project Exp	010	Build All	Ctrl+B	
✓ I myLib		Build Config 🗸 1 Debu	g	Set Active >
> 🎞 Arch		Build Project 2 Relea	se	Manage
> 🗗 Inclu > 🚰 Inc • 🚰 Src > 🖻 m	~	Build Working Set Clean Build Automatically	>	Build by Working Set>Set Active by Working Set>Manage Working Sets
> 💼 sy		Build Targets	>	
> 🖻 sy > 🗁 Debu > 🚾 NUCLEC > 🎇 Binar	MX	C/C++ Index Generate Report Generate Code Properties	>	

## 2.3.1.2 Create a new build configuration

To create a new build configuration:

- 1. Right-click on the project name in the *Project Explorer* view
- 2. Either:
  - Select [Build Configurations]>[Manage...]
  - Use the menu [Project]>[Build Configurations]>[Manage...]

Both methodes open the Manage Configurations dialog.

## Figure 62. Manage Configurations dialog

NUCLEO-F401RE: Manage Configurations					
Configuration Debug	Description	Status Active			
Release					
Set Active	New Dele	te Rename			
E	OK	Cancel			



As shown in Figure 62, some buttons in the dialog are used to manage the configurations:

- [Set Active] is used to change and select another configuration to be active
- [New...] is used to create a new build configuration
- [Delete] is used to delete an existing build configuration
- [Rename...] is used to rename the build configuration

To create a new build configuration, press the [**New...**] button. This opens the *Create New Configuration* dialog. In this dialog, a name and description is entered. The name must be a valid directory name since it is used as the directory name when building the project with the new configuration.

#### Figure 63. Create a new build configuration

DE Create New Configuration X					
Note: The co system. Plea	onfiguration n use ensure that	ame w at it is v	<i>i</i> ill be used as a dire valid for your platfo	ectory name in the file rm.	
Name:	Debug2				
Description:	My own deb	oug co	nfiguration		
Copy settings	s from				
• Existing co	onfiguration	Debu	g	$\sim$	
O Default co	onfiguration	Debu Poloa	g		
O Import fro	om projects	not	t selected	~	
O Import pr	edefined	not	t selected	~	
		[	ОК	Cancel	

As seen in Figure 63, the new build configuration is based on an existing build configuration. In the case illustrated, the new configuration is based on the existing *Debug* configuration. Press [**OK**] when finished with the settings.

The Manage Configurations dialog opens and the new debug configuration is displayed.

IDE NUCLEO-F401RE: Manage Configurations X					
Configuration	Description		Status		
Debug			Active		
Debug2	My own debu	ıg con			
Release					
Set Active	New	Del	ete	Rer	name
		ОК		Canc	el

#### Figure 64. Updated Manage Configurations dialog

Change the active configuration to another configuration if needed and press [**OK**] to save and close the configurations dialog when finished managing configurations.

#### 2.3.1.3 Delete a build configuration

To delete a build configuration:

- 1. Open the Manage Configurations dialog
- 2. Select the configuration to be deleted
- 3. Press the [Delete] button

For instance, if the *Debug2* configuration is selected and [**Delete**] button is pressed, the following confirmation dialog opens.

## Figure 65. Configuration deletion dialog

DE Confi	m Delete		×
?	Are you sure you want to delete the "Debug2" configura	ation?	
		Yes	No

In this case, select [No] to keep the Debug2 configuration.

### 2.3.1.4 Rename a build configuration

To rename a build configuration:

- 1. Open the Manage Configurations dialog
- 2. Select the configuration to be renamed
- 3. Press the [Rename...] button

For instance, if the *Debug2* configuration is selected and [**Rename...**] button is pressed, the following confirmation dialog opens.

#### Figure 66. Configuration renaming dialog

Rename Configuration X					
Note: The configuration name will be used as a directory name in the file system. Please ensure that it is valid for your platform.					
Name:	Name: Debug2				
Description:	Description: My own debug2 configuration				
	ОК	Cancel			

Update the name, description, or both and press [**OK**] to rename the *Debug2* configuration. In this case, press [**Cancel**] and keep the name.

## 2.3.2 Project C/C++ build settings

Each build configuration contains one project C/C++ build setting. The project C/C++ build setting is updated in project properties. To update the build setting, right-click on the project name in the *Project Explorer* view and select [**Properties**]or use the menu [**Project**]>[**Properties**]. Both these ways open the *Properties* window for the project.

Select [C/C++ Build]>[Settings] in the Properties left pane. The right part is then filled with tabs Tool Settings, Build Steps, Build Artifact, Binary Parsers, and Error Parsers. The first two tabs are the most useful ones.



Properties for NU	ICLEO-F401RE	— 🗆 X
type filter text	Settings	
> Resource		^
✓ C/C++ Build	Configuration: Debug [ Active ]	✓ Manage Configurations
Build Variat		
Environmer Logging	😻 Tool Settings 🎤 Build Steps 😤 Build Artifact 📓 Binary Parsers 🧕 Error Pars	sers
Settings	MCU Toolchain     Mcu     STM32F401RETx	
C/C++ Genera CMSIS-SVD Se	MCLI Past huild outputs Board NUCLEO-F401RE	
Project Referei 🗡	Source Post build outputs     Floating-point unit     Floating-point unit     FPv4-SP-D16	~
< >	So I Elosting-point ARI Hardware implementation /-mf	loat-abi-bard) v
?	Apply	and Close Cancel

#### Note:

Resize the dialog window or use the top-right arrow buttons if all tabs are not visible.

The *Settings* pane contains a **[Configuration]** selection to decide if new selections are used in the active configuration only, in another configuration, in all configurations or in multiple configurations. Press **[Manage Configurations]** to open the *Manage Configurations* dialog.

#### Figure 68. Properties configurations

<b>IDE</b> Properties for N	IUC	CLEO-F401RE					$\times$	
type filter text		Settings				¢ ▼ ¢	*	000
> Resource	•						,	~
✓ C/C++ Build Build Variat		Configuration:	Debug [Active] Debug [Active]	~	Manage Conf	igurations		
Environmer		Tool Setting	Debug2 Selease		Error Parsers			
Settings > C/C++ Genera		MCU Too MCU Set	All configurations ] Multiple configurations] nas					

The Tool Settings tab is further split into MCU Toolchain, MCU Settings, MCU Post build outputs, MCU GCC Assembler, MCU GCC Compiler and MCU GCC Linker.

*MCU Toolchain* is used to change toolchains. STM32CubeIDE includes one version of the *GNU Tools for STM32* toolchain. The *Toolchain Manager* is used to download other *GNU ARM Embedded* toolchains and to configure to use local *GNU ARM Embedded* toolchains.

Information about patches made in *GNU Tools for STM32* can be read in [EXT-12]. The document can be opened from the *Technical Documentation* page in the *Information Center*.

Properties for NU	CLEO-F401RE	— — X
type filter text	Settings	
<ul> <li>Resource</li> <li>Builders</li> <li>C/C++ Build</li> <li>Build Variables</li> <li>Environment</li> </ul>	Configuration: Debug [Active]	✓ Manage Configurations
Settings C/C++ General CMSIS-SVD Settin Project Reference Run/Debug Settin	MCU Toolchain MCU Settings MCU Post build outputs MCU GCC Assembler MCU GCC Compiler MCU GCC Linker	Select what toolchain to use Use workspace setting (GNU Tools for STM32 (9-2020-q2-update)) Fixed GNU Tools for STM32 (9-2020-q2-update) Toolchain Manager
< >>		Configure the workspace toolchain, and manage installed toolchains. Open Toolchain Manager
?		Apply and Close Cancel

#### Figure 69. Properties toolchain version

## Select [Fixed] to enable the toolchain selection.

Properties for NUC	CLEO-F401RE – 🗆 🗙
type filter text	Settings $\diamond \star \diamond \star$
<ul> <li>Resource Builders</li> <li>C/C++ Build Build Variables</li> </ul>	Configuration: Debug [Active]
Environment Logging Settings > C/C++ General CMSIS-SVD Settin Project Reference Run/Debug Settin	<ul> <li>Tool Settings          Build Steps         Build Artifact         Binary Parsers         Error Parsers         Error Parsers         Select what toolchain to use         Use workspace setting (GNU Tools for STM32 (9-2020-q2-update))         Select what toolchain to use         Use workspace setting (GNU Tools for STM32 (9-2020-q2-update))         Fixed         GNU Tools for STM32 (9-2020-q2-update)         Select what toolchain and manage         Solution         Select what toolchain to use         Use workspace setting (GNU Tools for STM32 (9-2020-q2-update))         Select what toolchain Manager         Solution         Select what toolchain to use         Outrools for STM32 (9-2020-q2-update)         Solution         Select what toolchain Manager         Solution         Select what toolchain Manager         Select what toolchain Manager</li></ul>
< >	· · · · · · · · · · · · · · · · · · ·
(?)	Apply and Close Cancel

#### Figure 70. Properties toolchain selection

As shown in Figure 70, only the default toolchain *GNU Tools for STM32* is available by default. To install additional toolchains, click on the [**Open Toolchain Manager.**.] button to open the *Toolchain Manager*. Section 2.11 Toolchain Manager contains detailed information on how to install, uninstall toolchains and select the default workspace toolchain.

*MCU Settings* displays the selected MCU and board for the project and proposes to select how to handle floating point, instruction set and runtime library.

Properties for NUC	CLEO-F401RE						$\times$
type filter text	Settings					⇔ ▼ ⇔	• 8
type filter text > Resource Builders ✓ C/C++ Build Build Variables Environment Logging Settings > C/C++ General CMSIS-SVD Settin Project Reference Run/Debug Settin	Settings Configuration: Debug [Active] Tool Settings Puild Steps MCU Toolchain MCU Settings MCU Post build outputs MCU GCC Assembler Bebugging Preprocessor Binclude paths Miscellaneous Seneral Bebugging Debugging Seneral Biscellaneous Seneral Biscellaneous	Build Artifact Mcu Board Floating-point unit Floating-point ABI Instruction set Runtime library Use float with p Use float with so	Binary Parsers STM32F401RETx NUCLEO-F401RE FPv4-SP-D16 Hardware implem Thumb2 Reduced C (spec rintf from newlib-na canf from newlib-na	<ul> <li>Error Parsers</li> <li>entation (-mfloat-a</li> <li>:s=nano.specs)</li> <li>ano (-u _printf_floatano (-u _scanf_floatano (-u _scanf_floatano</li></ul>	Manage Co abi=hard)	• • •	· · · · · · · · · · · · · · · · · · ·
< >	<ul> <li>Include paths</li> <li>Optimization</li> <li>Warnings</li> <li>Miscellaneous</li> <li>MCU GCC Linker</li> <li>General</li> <li>Libraries</li> <li>Miscellaneous</li> </ul>						~
?				Apply	/ and Close	Cancel	

## Figure 71. Properties tool MCU settings

*MCU Post build outputs* proposes to convert the elf file to another file format, show build size information, and generate list file. The output file can be converted to:

- Binary file
- Intel Hex file
- Motorola S-record file
- Motorola S-record symbols file
- Verilog file

Properties for NUC	CLEO-F401RE					×
type filter text	Settings				$\langle \neg \bullet \circ \rangle$	• 8
<ul> <li>Resource Builders</li> <li>C/C++ Build Build Variables Environment Logging Settings</li> <li>C/C++ General CMSIS-SVD Settin</li> </ul>	Configuration: Debug [Active]  Tool Settings MCU Toolchain MCU Settings MCU Post build outputs	<ul> <li>Build Artifact Binary Parsers</li> <li>Convert to binary file (-O binary)</li> <li>Convert to Intel Hex file (-O ihex)</li> <li>Convert to Motorola Screeged file (-</li> </ul>	Error Parsers	Manage Config	gurations.	
Project Reference Run/Debug Settin	<ul> <li>MCU GCC Assembler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>Miscellaneous</li> <li>MCU GCC Compiler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>Optimization</li> <li>Warnings</li> <li>Miscellaneous</li> <li>MCU GCC Linker</li> <li>General</li> <li>Libraries</li> <li>Miscellaneous</li> </ul>	<ul> <li>Convert to Verilog file (-O verilog)</li> <li>Convert to Motorola S-record (sym</li> <li>Show size information about built a</li> <li>✓ Generate list file</li> </ul>	-O srec) ibols) file (-O symbolsre artifact	ec)		•
?			Apply and	l Close	Cancel	

# Figure 72. Properties tool MCU post-build settings

The *MCU GCC Assembler* settings contains selections for the assembler. The main node presents all the assembler command-line options that are currently enabled in the sub-node settings. The sub-nodes are used to view the current settings or change any settings for the assembler.

Properties for NUC	CLEO-F401RE			—	
type filter text > Resource Builders • C/C++ Build Build Variables Environment Logging Settings > C/C++ General CMSIS-SVD Settin Project Reference Run/Debug Settin	Settings Configuration: Debug [Active] Tool Settings Puild Steps	<ul> <li>✓ Manage Con</li> <li>s 9 Build Artifact     <li>Binary Parsers</li> <li>O Error Parsers</li> <li>Command:</li> <li>Gcc</li> <li>All options:</li> <li>-mcnu=cortex-m4 -co3 -c -x assembler-with-con</li> </li></ul>			
	<ul> <li>MCU Settings</li> <li>MCU Post build outputs</li> <li>MCU GCC Assembler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>Miscellaneous</li> <li>MCU GCC Compiler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>Optimization</li> <li>Warnings</li> <li>Miscellaneous</li> <li>MCU GCC Linker</li> <li>General</li> </ul>	All options: Expert settin Command line pattern:	-mcpu=cortex-m4 -g3 -c -x assembler-with specs=nano.specs -mfpu=fpv4-sp-d16 -mfl gs: \${COMMAND} {FLAGS} {OUTPUT_FLAG}	cpp oat-abi=hard -mth \${OUTPUT_PREFIX}	umb
< > ?	Miscellaneous		Apply	and Close	<b>∨</b> Cancel

### Figure 73. Properties tool GCC assembler settings

The *MCU GCC Compiler* settings contains selections for the compiler. The main node presents all the compiler command-line options that are currently enabled in the sub-node settings. The sub-nodes are used to view the current settings or change any settings for the compiler.

Properties for NUC	CLEO-F401RE					×
<ul> <li>Properties for NUC</li> <li>type filter text</li> <li>Resource Builders</li> <li>C/C++ Build Build Variables Environment Logging Settings</li> <li>C/C++ General CMSIS-SVD Settin Project Reference Run/Debug Settin</li> </ul>	CLEO-F401RE Settings Configuration: Debug [Active] Tool Settings MCU Toolchain MCU Settings MCU Post build outputs MCU GCC Assembler Debugging Preprocessor	Build Ar Command: All options: Expert settin	✓ Ma tifact ■ Binary Parsers ● Error Parsers gcc -mcpu=cortex-m4 -std=gnu11 -g3 -DDEBUG -DUS -DSTM32F401xE -c -l/Core/Inc - I/Drivers/STM32F4xx_HAL_Driver/Inc - gs:	anage Conf SE_HAL_DR	□	× *
	<ul> <li>Preprocessor</li> <li>Include paths</li> <li>Miscellaneous</li> <li>MCU GCC Compiler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>Optimization</li> <li>Warnings</li> <li>Miscellaneous</li> <li>MCU GCC Linker</li> <li>General</li> <li>Libraries</li> <li>Miscellaneous</li> </ul>	Expert settin Command line pattern:	gs: \${COMMAND} \${INPUTS} \${FLAGS} \${OUTPUT_FLA	<u>G}</u> \${OUTP	UT_PREFI)	~
?			Apply and Clo	ose	Cancel	

### Figure 74. Properties tool GCC compiler settings

57

The *MCU GCC Linker* settings contains selections for the linker. The main node presents all the linker commandline options that are currently enabled in the sub-node settings. The sub-nodes are used to view the current settings or change any settings for the linker.

Properties for NUC	CLEO-F401RE			_		$\times$
type filter text	Settings				<> ▼ ⇒	▼ 8
<ul> <li>C/C++ Build</li> <li>Build Variables</li> <li>Environment</li> </ul>	Configuration: Debug [Active]	😤 Build Ar	tifact 🗟 Binary Parsers 🥝 Error Parsers	Manage Confi	igurations	
Logging Settings > C/C++ General CMSIS-SVD Settin Project Reference Run/Debug Settin	<ul> <li>MCU Toolchain</li> <li>MCU Settings</li> <li>MCU Post build outputs</li> <li>MCU GCC Assembler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>MCU GCC Compiler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>Optimization</li> <li>Warnings</li> <li>Miscellaneous</li> <li>MCU GCC Linker</li> <li>General</li> <li>Libraries</li> <li>Miscellaneous</li> </ul>	Command: All options: Expert settin Command line pattern:	gcc -mcpu=cortex-m4 -T"C:\Users\johansse\STM32 \workspace_um5\NUCLEO-F401RE\STM32F401F specs=nosys.specs -WI,-Map="\${BuildArtifactFil gs: \${COMMAND} \${OUTPUT_FLAG} \${OUTPUT_PRE	!CubeIDE RETX_FLASH.lc leBaseName}.r EFIX}\${OUTPU	I" nap" V	~
?			Apply and	Close	Cancel	

## Figure 75. Properties tool GCC linker settings

The *Build Steps* settings contains fields used to provide pre-build and post-build steps, which run before and after building the project. Edit the fields to run any pre-build or post-build step.

Properties for NUCLEO-F401R	e – 🗆 X
type filter text	Settings $(\neg \bullet \ominus) \bullet \bullet \bullet$
<ul> <li>Resource</li> <li>C/C++ Build Build Variables Discovery Options Environment Logging</li> </ul>	Configuration: Debug [Active]   Manage Configurations  Tool Settings  Build Steps  Build Artifact  Binany Parsers Frror Parsers
Settings > C/C++ General CMSIS-SVD Settings Project Natures Project References Run/Debug Settings	Pre-build steps Command: Pest-build steps Command: Post-build steps Command:
	Command: Description:
?	Apply and Close Cancel

## Figure 76. Properties build steps settings

Note: It is possible to add more advanced post-build operations using makefile targets as described in Section 2.4.7.

# 2.4 Building the project

To start a build, select the corresponding project in the *Project Explorer* view and click on the [**Build**] <sup>6</sup> Toolbar button.

#### Figure 77. Project build toolbar

The build can also be started from menu [**Project**]>[**Build Project**]. The [**Project**] menu contains also some other usable build commands such as [**Build All**], [**Build Project**] or [**Clean**].

Another way to start a build is to right-click on the project in the *Project Explorer* view. This opens the context menu with the [**Build**] command and some other build options.

During the build, the *Console* view lists the build process. At the end, when the elf file is created normally, it lists size information.

#### Figure 78. Project build console

🖹 Problems 🖉 Tasks 📮 Console 🛛 🔲 Properties 🕹 🗘 🖓 🖓 🖓 🖓 👘 🖆 🖤 😁 🕶	
CDT Build Console [NUCLEO-F401RE] arm-none-eabl-gcc "/Core/Src/stm32f4xx_hal_msp.c" -mcpu=cortex-m4 -std=gnul1 -g3 -DUSE_HAL_DRIVER -DSTM32F401xE -DDEBUG -c arm-none-eabl-gcc "/Core/Src/stm32f4xx_hal_msp.c" -mcpu=cortex-m4 -std=gnul1 -g3 -DUSE_HAL_DRIVER -DSTM32F401xE +DD arm-none-eabl-gcc "/Core/Src/stm32f4xx_it.c" -mcpu=cortex-m4 -std=gnul1 -g3 -DUSE_HAL_DRIVER -DSTM32F401xE +DD arm-none-eabl-gcc "/Core/Src/syscalls.c" -mcpu=cortex-m4 -std=gnul1 -g3 -DUSE_HAL_DRIVER -DSTM32F401xE +DDEBUG arm-none-eabl-gcc "/Core/Src/syseme.c" -mcpu=cortex-m4 -std=gnul1 -g3 -DUSE_HAL_DRIVER -DSTM32F401xE +DDEBUG arm-none-eabl-gcc "/Core/Src/system_stm32f4xx.c" -mcpu=cortex-m4 -std=gnul1 -g3 -DUSE_HAL_DRIVER -DSTM32F401xE +DDEBUG arm-none-eabl-gcc "/Core/Src/system_stm32f4xx.c" -mcpu=cortex-m4 -std=gnul1 -g3 -DUSE_HAL_DRIVER -DSTM32F401xE +DDEBUG arm-none-eabl-gcc -o "NUCLEO-F401RE.elf" @"objects.list" -mcpu=cortex-m4 -T"C:\Users\johansse\STM32CubeIDE\wor Finished building target: NUCLEO-F401RE.elf	-1. E -1 EBU -c c - DI ksp
arm-none-eabi-size NUCLEO-F401RE.elf arm-none-eabi-objdump -h -S NUCLEO-F401RE.elf > "NUCLEO-F401RE.list"	
text data bss dec hex filename	
Finished building: default.size.stdout	
Finished building: NUCLEO-F401RE.list	
12:42:04 Build Finished. 0 errors, 0 warnings. (took 5s.932ms)	
	~
<	>

# 2.4.1 Building all projects

The toolbar contains the [**Build all**] button, which is used to build the active build configuration for all open projects in workspace.

It is also possible to use the menu [Project]>[Build All] to start a build of all projects.

#### Figure 79. Project build all

workspace\_um - NUCLEO-F401RE/Core/Src/main.c - STM32CubeIDE



# 2.4.2 Build all build configurations

To build all build configurations for a project, right-click the project and select [**Build Configurations**]>[**Build All**] in the context menu.

Project Explorer	· 🛛			.c
<ul> <li>Image: myLib</li> <li>Archives</li> <li>Image: myLib</li> <li>Image: my</li></ul>				
NUCLEO     Binari	New	>		
> 🗊 Includ	Go Into			
✓ <sup>2</sup> Core	Open in New Window			
> 🔁 Inc 📑	Сору	Ctrl+C		
src 📑	Paste	Ctrl+V Delete		
> 🔤 🐣	Delete	Delete		
> 尾	Move	ŕ		
> .c	Rename	F2		
> 🚺	Import			
	Export			
> 🖂 Driver	Build Proiect			
> 🗁 Debu	Clean Project			
> 👝 Debu 🗞	Refresh	F5		
> 🔁 Releas	Close Project			
MX NUCL	Close Unrelated Projects			
STM3	Build Configurations	>	Set Active	>
TI STM3	Build Targets	>	Manage	
	Index	>	Build All	
	Show in Remote Systems view		Clean All	
0	Run As	>	Build Selected	

Figure 80. Project build-all configurations

#### 2.4.3 Headless build

Headless build is intended to be used to build projects that must be integrated into script-controlled builds, such as nightly builds on build servers for continuous integration process methods or others. The STM32CubeIDE GUI is never displayed in this case, and the user is not requested any manual interaction with STM32CubeIDE. STM32CubeIDE includes a headless-build command file to run headless builds. For instance, when using Windows<sup>®</sup>, it is located in the C:\ST\STM32CubeIDE\_1.7.0\STM32CubeIDE STM32CubeIDE installation folder. The headless-build.bat file is intended to be run from a command prompt.

Before running any headless build, make sure that the workspace is not opened by STM32CubeIDE. If there is an STM32CubeIDE running already using the workspace, it is not possible for the headless-build process to open and build the project.

To run headless build in Windows<sup>®</sup>, use the following procedure:

- 1. Open a command prompt.
- 2. Navigate to the STM32CubeIDE installation directory. Open the folder in which the IDE is stored. For example: cd C:\ST\STM32CubeIDE 1.7.0\STM32CubeIDE
- 3. Enter the following command to build the NUCLEO-F401RE project in the workspace

C:\Users\Name\STM32CubeIDE\workspace\_1.7.0:

\$ headless-build.bat -data C:\Users\Name\STM32CubeIDE\workspace\_1.7.0
-cleanBuild NUCLEO-F401RE

Note:

To get help on headless build parameters, use headless build with option -help. Figure 81 shows the result of command \$ headless-build.bat -help.

#### Figure 81. Headless build

Command Prompt	-		$\times$
C:\ST>cd STM32CubeIDE_1.7.0.21alpha1			^
C:\ST\STM32CubeIDE_1.7.0.21alpha1>headless-build.bat -help			
Usage: PROGRAM -data <workspace> -application org.eclipse.cdt.managedbuilder.core.headlessbuild [ OPTIONS ]</workspace>			
<pre>-data {/path/to/workspace} -import {[uri:/]/path/to/project} -importAll {[uri:/]/path/to/projectTreeURI} Import all projects under URI -build {project_name_reg_ex{/config_reg_ex}   all} -cleanBuild {project_name_reg_ex{/config_reg_ex}   all} -cleanBuild {project_name_reg_ex{/config_reg_ex}   all} -markerType Marker types to fail build on {all   cdt   marker_id} -mo-indexer Disable indexer -printErrorMarkers Print all error markers -I {include_file} additional include_file to pass to tools -include {include_file} addition preprocessor defines to pass to the tools -D {prepoc_define} addition preprocessor defines to pass to the tools -E {var=value} replace/add value to environment variable when running all tools -Ep {var=value} prepend value to environment variable when running all tools -Er {var=value} prepend value to environment variable when running all tools -Er {var=value} prepend value to anol option value in each configuration built -Ta {toolid} {optionid=value} prepend to a tool option value in each configuration built -Tp {toolid} {optionid=value} prepend to a tool option value in each configuration built -Tp {toolid} {optionid=value} prepend to a tool option value in each configuration built -Tp {toolid} {optionid=value} prepend to a tool option value in each configuration built -Tp {toolid} {optionid=value} prepend to a tool option value in each configuration built -Tp {toolid} {optionid=value} prepend to a tool option value in each configuration built -Tp {toolid} {optionid=value} prepend to a tool option value in each configuration built</pre>			
Tool option values are parsed as a string, comma separated list of strings or a boolean based on the o	ptions	type	
C:\ST\STM32CubeIDE_1.7.0.21alpha1>			~

### 2.4.4 Temporary assembly file and preprocessed C code

Save the temporary assembly file by adding the *-save-temps* flag to the compiler:

- 1. In the menu, select [Project]>[Properties]
- 2. Select [C/C++ build]>[Settings]
- 3. Open the *Tool Settings* tab
- 4. Add -save-temps in the [C Compiler]>[Miscellaneous] settings
- 5. Rebuild the project

The assembler file is located in the build output directory with name filename.s.

The file filenamz.i containing the preprocessed C code is generated also. It shows the code after the preprocessor but before the compilation. It is advise to examine the content of this file in case of problems with defines.

# 2.4.5 Build logging

To enable or disable project build logging, right-click on the project in the *Project Explorer* view and select [**Properties**]. Then, select [**C/C++ Build**]>[**Logging**]. The log file location and name are also specified. To enable a global build log for all projects in a workspace, select [**Window**], [**Preferences**], and open [**C/C++**, **Build**, **Logging**]>[**Enable global build logging**].

## 2.4.6 Parallel build and build behaviour

Parallel build occurs when more than one thread is used at the same time to compile and build the code. Most often, it reduces build time significantly. The optimal number of threads to use is usually equal to the number of CPU cores of the computer. Parallel build can be enabled and disabled.

To configure parallel build:

- 1. Right-click on the project in the Project Explorer view
- 2. Select menu [Project]>[Properties]
- 3. Select [C/C++ Build] in the Properties panel
- 4. Open the *Behavior* tab and configure [Enable parallel build]

The *Behavior* tab also contains build settings on how to behave on errors, build on resource save, incremental build, and clean.

Properties for NUCLEO-F401RE	:		– 🗆 X
type filter text	C/C++ Build		<> ▼ <> ▼ ▼
<ul> <li>Resource</li> <li>C/C++ Build</li> <li>C/C++ General CMSIS-SVD Settings Project Natures Project References Run/Debug Settings</li> </ul>	Configuration: Debug [Active]	<ul> <li>Mar</li> <li>Refresh Policy</li> </ul>	age Configurations
	Build settings	<ul> <li>Enable parallel build</li> <li>Use optimal jobs (4)</li> <li>Use parallel jobs:</li> <li>Use unlimited jobs</li> </ul>	]
	Workbench Build Behavior		
	Workbench build type:	Make build target:	
	Build on resource save (Auto build)	all	Variables
	Note: See Workbench automatic build	preference	
	Build (Incremental build)	all	Variables
	Clean	clean	Variables
		Restore Defaults	Apply
?		Apply and Close	Cancel

# Figure 82. Parallel build

# 2.4.7 Post-build with makefile targets

It is possible to add advanced post-build scripts by using makefile targets. To do this:

- 1. Create a new file
- 2. Name it makefile.targets
- 3. Place it in the root directory of the project

The content of the file must be similar to the example presented below. The example just copies the elf generated file to a new file and uses macros BUILD\_ARTIFACT, BUILD\_ARTIFACT\_PREFIX, BUILD\_ARTIFACT\_NAME, and BUILD\_ARTIFACT\_EXTENSION, which are generated into the makefile by STM32CubeIDE from v1.5.0.

```
secure_target := \
    $ (BUILD_ARTIFACT_PREFIX)$(BUILD_ARTIFACT_NAME) - secure.$(BUILD_ARTIFACT_EXTENSION)
main-build: $(secure_target)
$(secure_target): $(BUILD_ARTIFACT)
    # Do what you want here... simple copy file for demo
    cp "$<" "$0"</pre>
```

*Note: make requires that tabs are used instead of spaces.* 

# 2.5 Linking the project

This section contains basic information about the linker and linker script files. Detailed information about the linker can be found in the *GNU Linker* manual ([EXT-05]), which is accessed from the *Information Center*. Click on the

[Information Center] toolbar button 1 and open the *Information Center* view. Open the linker documentation using the [C/C++ Linker The GNU Linker PDF] link.

Figure 83. Linker documentation	Figure
---------------------------------	--------

IDE			×
1 Information	Center 🛛	🟠 🗘 🖒 A* 🖉	
			^
	TOOL CHAIN MANUALS (CNULTOOLS FOR STM227 2019 O2 LIPDATE)		
	100E01414 MAROALS (GR0-100E3-FOR-STM32.7-2018-02-0FDATE)		
	Description	File format	
	Assembler	PDF	
	The GNU Assembler	PDF	
	Binary Utilities	PDF	
	The GNU Binary Utilities	<u></u>	
			-
	C Math Library	PDF	
	The Red Hat newlib C Math Library	_	
	C Preprocessor	PDF	
	The GNU C Preprocessor		
	C Runtime Library	PDF	
	I në Red nat newlib C Library		
	C++ Library Manual	HTML	
	C/C++ Compiler		
	GNU Compiler Collection	PDF	
	C/C++ Linker		
	The GNU Linker	PDF	
			-
	Debugger	225	
	Debugging with GDB		
	GDB Quick Reference Card	PDF	
	The GNU GDB Reference Card	<u></u>	
	Newlib-nano readme	тхт	
	Newlib-nano C runtime library readme	_	
	Patch List	PDF	
	Patch list for GNU Tools for STM32		
<		>	, ×
			- 4

# 2.5.1 Run time library

The toolchains included in STM32CubeIDE contain two prebuilt run time C libraries based on newlib. One is the standard C newlib library and the other is the reduced C newlib-nano. Use newlib-nano to achieve smaller code size. For information about the differences between newlib-nano and the standard newlib, refer to the newlib-nano readme file ([ST-09]), accessible from the *Information Center*.

To select the desired run time library for use in the project.

- 1. Right-click on the project in the Project Explorer view
- 2. Select menu [Project]>[Properties]
- 3. Select [C/C++ Build]>[Settings] in the *Properties* panel
- 4. Open the Tool Settings tab, select [MCU Settings] and configure the [Runtime library] setting



#### Figure 84. Linker run time library

When newlib-nano is used while floating-point numbers must be handled by scanf/printf, additional options are required. The reason is that newlib-nano and newlib handle floating-point numbers differently. In newlib-nano, formatted floating-point number inputs and outputs are implemented as weak symbols. Therefore, the symbols must be pulled by specifying explicitly if %f is used with scanf/printf using the -u option:

- -u \_scanf\_float
- -u \_printf\_float

For example, to enable output float with printf, the command line is as follows:

\$ arm-none-eabi-gcc --specs=nano.specs -u \_printf\_float \$(OTHER\_LINK\_OPTIONS)
The options can be enabled using the [Use float ...] checkboxes in [MCU Settings] in the Tool Settings tab.

IDE Properties for NUC	CLEO-F401RE			—		×
type filter text	Settings				<> ▼ <	~ •
<ul> <li>Resource</li> <li>C/C++ Build Build Variables Discovery Opti Environment Logging Settings</li> <li>C/C++ General CMSIS-SVD Settin Project Reference Refactoring Histor Run/Debug Settin</li> </ul>	Configuration: Debug [Active Toolchain Version Tool Set MCU Settings MCU Post build outputs MCU GCC Assembler General Debugging Preprocessor Include paths Miscellaneous MCU GCC Compiler General Debugging Preprocessor Include paths Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous	] Mcu Board Floating-point unit Floating-point ABI Instruction set Runtime library ☑ Use float with p ☑ Use float with so	Build Artifact Binary STM32F401RETx NUCLEO-F401RE FPv4-SP-D16 Hardware implementati Thumb2 Reduced C (specs=nar rintf from newlib-nano (-total) canf from newlib-nano (-total)	Manage Cont Parsers @ Err on (-mfloat-ab no.specs) u _printf_float) u _scanf_float)	figuration: for Parsers	· · · · · · · · · · · · · · · · · · ·
?			Apply and	d Close	Cancel	

Figure 85. Linker newlib-nano library and floating-point numbers

# 2.5.2 Discard unused sections

Linker optimization is the process where the linker removes unused code and data sections, dead code, from the output binary. Run time and middleware libraries typically include many functions that are not used by all applications, thus wasting valuable memory unless removed from the output binary.

When using the project wizard to create new projects, the default configuration is that the linker discards unused sections. To check or change the setting about unused sections, open at any time the build settings for the project:

- 1. Right-click the project in the *Project Explorer* view and select [Properties]
- 2. In the dialog, select [C/C++ Build]>[Settings]
- 3. Select the *Tool Settings* tab in the panel
- 4. Select [MCU GCC Linker]>[General]
- 5. Configure [Discard unused sections (-WI, --gc-sections)] according to the project requirements
- 6. Rebuild the project

57/



#### Figure 86. Linker discard unused sections

## 2.5.3 Page size allocation for malloc

When the GNU Tools for STM32 toolchain is used with the standard C newlib library, the page size setting for malloc can be changed. The newlib default page size is 4096 bytes. If a sysconf() function is implemented in the user project, this user function is called by  $_malloc_r()$ .

The following example shows how to implement a sysconf() function with a 128-byte page size. Add a similar function if there is a need for the application to use a smaller page size than the default 4096 bytes.

```
/**
* * * *
** File
        : sysconf.c
***
                                    **/
/* Includes */
#include <errno.h>
#include <unistd.h>
long sysconf(int name)
 if (name== SC PAGESIZE)
   return 128;
 }
 else
   errno=EINVAL;
   return -1;
 }
```

Note: If the "GNU ARM Embedded" toolchain is used, it does not call any <code>sysconf()</code> function implemented in the application but always uses the default <code>sysconf()</code> function in <code>newlib</code>. Also, no call to <code>sysconf()</code> is made if the "GNU Tools for STM32" toolchain is used with the reduced C <code>newlib-nano</code> library.

# 2.5.4 Include additional object files

STM32CubeIDE makes it easy to include additional object files that must be linked to a project. They can be files from other projects, precompiled libraries where no source code is available, or object files created with other compilers.

- 1. Right-click the project in the Project Explorer view and select [Properties]
- 2. In the dialog, select [C/C++ Build]>[Settings]
- 3. Select the *Tool Settings* tab in the panel
- 4. Select [MCU GCC Linker]>[Miscellaneous]
- 5. Use the [Add...] icon to add additional object files in several possible ways:
  - Enter the filenames in the Add file path dialog
    - Use the [Workspace...] or [File system...] buttons to locate the files

#### Figure 87. Linker include additional object files

Properties for NUCLEO-F401RE			$ \square$ $\times$
type filter text	Settings		<> ▼ <> ▼ <
<ul> <li>Resource</li> <li>C/C++ Build</li> <li>Build Variables</li> <li>Discovery Options</li> <li>Environment</li> </ul>	Configuration: Debug [Act	ve ]	✓ Manage Configurations
Logging	🛞 Tool Settings 🎤 Build	Steps   ∰ Build Artifact 🗟 Binary Parser	rs 😣 Error Parsers 🔹 🕨
Settings > C/C++ General CMSIS-SVD Settings Project Natures Project References Run/Debug Settings	<ul> <li>MCU Settings</li> <li>MCU Post build outp</li> <li>MCU GCC Assembler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Miscellaneous</li> <li>MCU GCC Compiler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>MICU GCC Compiler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Miscellaneous</li> <li>MS MCU GCC Linker</li> <li>General</li> <li>Libraries</li> <li>Miscellaneous</li> </ul>	Other flags	<ul> <li>●</li> <li>●</li> <li>●</li> <li>●</li> <li>●</li> <li>●</li> <li>○</li> <li>○</li></ul>
?		[	Apply and Close Cancel

## 2.5.5 Treat linker warnings and errors

The GNU linker is normally silent for warnings. One example of such silent warning is seen if the startup code containing the normal <code>Reset\_Handler</code> function is missing in the project. The GNU linker in normal silent mode creates an elf file and only report a warning output in the *Console* window about the missing <code>Reset\_Handler</code>.

arm-none-eabi-gcc -o "NUCLEO-F401RE.elf" @"objects.list" -mcpu=cortex-m4 -T"C:\Users\username\STM32CubeIDE\workspace\_um\NUCLEO-F401RE\STM32F401RETX\_FLASH.ld" --specs=nosys.specs -Wl,-Map="NUCLEO-F401RE.map" -Wl,--gc-sections -static mfpu=fpv4-sp-d16 -mfloat-abi=hard -mthumb -Wl,--start-group -lc -lm -Wl,--end-group c:\st\stm32cubeide\_1.1.0.19w37\stm32cubeide\plugins\com.st.stm32cube.ide.mcu.extern altools.gnu-tools-for-stm32.7-2018-q2-update.win32\_1.0.0.201904181610\tools\armnone-eabi\bin\ld.exe: warning: cannot find entry symbol Reset\_Handler; defaulting to 000000008000000

Finished building target: NUCLEO-F401RE.elf

In this case, a new elf file is created but, if the warning is not detected, it will not work to debug the project because the program does not contain the Reset\_Handler function. It is possible to configure the linker to treat warnings as errors by adding the --fatal-warnings option.

When the --fatal-warnings option is used, the linker does not generate the elf file but displays an error in the console log:

```
c:\st\stm32cubeide_1.1.0.19w37\stm32cubeide\plugins\com.st.stm32cube.ide.mcu.extern
altools.gnu-tools-for-stm32.7-2018-q2-update.win32_1.0.0.201904181610\tools\arm-
none-eabi\bin\ld.exe: warning: cannot find entry symbol Reset_Handler; defaulting
to 000000000000
collect2.exe: error: ld returned 1 exit status
make: *** [makefile:40: NUCLEO-F401RE.elf] Error 1
"make -j4 all" terminated with exit code 2. Build might be incomplete.
11:26:30 Build Failed. 1 errors, 6 warnings. (took 7s.193ms)
```

To use the -Wl, --fatal-warnings option:

- 1. Right-click the project in the Project Explorer view and select [Properties]
- 2. In the dialog, select [C/C++ Build]>[Settings]
- 3. Select the Tool Settings tab in the panel
- 4. Select [MCU GCC Linker]>[Miscellaneous]
- 5. Add -Wl, --fatal-warnings to the [Other flags] field.


## Figure 88. Linker fatal warnings

# 2.5.6 Linker script

The linker script file (.1d) defines the files to include and where things end up in memory. Some important parts of the linker script file are described in the next sections. For detailed information about the linker, read the C/C++ linker *GNU Linker* manual ([ST-05]). This manual is available in the documentation section of the *Information Center*. Consider sections 3.6 and 3.7 especially.

The linker script specifies the memory regions and the location of the stack, heap, bss, data, rodata, text, and program entry. The size of stack and heap are configurable by editing the \_Min\_Stack\_Size and \_Min\_Heap\_Size values in the linker script file. However, these values are only used by the linker to validate that stack and heap fit in memory. When running the program, the stack or heap may require more memory, which may lead to unexpected results if data is overwritten.

Table 3 presents as an example the typical program and memory layout of an STM32F4 device with 512-Kbyte Flash memory and 96-Kbyte SRAM. The device is based on the Cortex<sup>®</sup>-M core with 32-bit address space (0x0000 0000 to 0xFFFF FFFF).

Example: STM32F4 96-Kbyte SRAM 512-Kbyte Flash memory	Usage	Files Linker script.ld, or.h and.c files	Comment
0xFFFF FFFF 0xE000 0000	Cortex-M4 internal peripherals.	.h and .c files.	SysTick, NVIC, ITM, debug and others.
0xDFFF FFFF 0x6000 0000	External memory FMC (Flexible memory controller).	Must be added in linker script, and . h and . c files. <sup>(1)</sup>	NOR Flash memory,NAND Flash memory, SPI Flash memory, PSRAM, SDRAM and others.
0x5FFF FFFF 0x4000 0000	STM32 Peripherals.	.h and .c files.	GPIO, ADC, timers, USB, USART and others.
0x2001 8000	96-Kbyte SRAM Stack Heap	Linker script _estack _Min_Stack_Size _Min_Heap_Size user_heap_stack	The stack contains local data <sup>(2)</sup> <sup>(3)</sup> Heap used by malloc <sup>(4)</sup> Data
0x2000 0000	Data	.bss .data	<pre>Static global data (.bss and .data) .bss == Uninitialized data Cleared to zero by the startup codedata == Initialized data Copied from Flash memory to SRAM by the startup code.</pre>
0x0808 0000	512-Kbyte Flash memory Data	Linker script .data .rodata	Initialized data to copy to SRAM. Read-only data placed in Flash memory.
0x0800 0000	Program Interrupt vector table	ENTRY Reset_Handler <sup>(5)</sup> .text .isr_vector <sup>(6)</sup>	<pre>.text == Program, such as main() in main.c, SystemInit() in system_stm32f4xx.c, Reset_Handler in startup_stm32*.s, g_pfnVectors in startup_stm32*.s, Vector table in startup_stm32*.s.</pre>

## Table 3. Memory map layout

Color legend

Cortex<sup>®</sup>-M internal peripherals and STM32 peripherals.

External memory. Normally the linker script, header files, and C files must be updated to use external memories.

Flash memory and SRAM where program, data, heap, and stack are located. Usually, when creating a project with STM32CubeIDE, these Flash memory and RAM regions are accessible and usable without any updates of the linker script or other files. The linker script file defines how to place code, data, heap, and stack in memory.

- 1. If external memory is used, the memory must be added into the linker script file. See in chapter Section 2.5.7.1 how to add a new memory region.
- 2. The stack grows downwards and may go into the heap.
- 3. When running the program, the stack or heap may require more memory, which might lead to unexpected results if data is overwritten.
- 4. The heap grows upwards and may go into the stack.
- 5. The linker script file contains the entry point definition of the program. Normally, ENTRY (Reset\_Handler).
- 6. The interrupt vector table contains the reset value of the stack pointer, the start addresses of the program (Reset\_Handler), exception handlers and interrupt handlers. Normally the Reset\_Handler code and vector table (g pfnVectors) are available in file <startup stm32xxx.s>.

See below the default linker script generated by STM32CubeIDE for an STM32F4 device with 512-Kbyte Flash memory and 96-Kbyte SRAM.

The beginning of the code excerpt shows the linker script header, entry, stack, heap and memory definitions.

```
/**
           ********
* @file LinkerScript.ld
* @author Auto-generated by STM32CubeIDE
* Abstract : Linker script for NUCLEO-F401RE Board embedding STM32F401RETx Device from
stm32f4 series
                      512Kbytes FLASH
 *
                      96Kbytes RAM
 *
 *
            Set heap size, stack size and stack location according
 *
            to application requirements.
            Set memory bank area and size if external memory is used
 *
                                                                  ******
 ***
 * @attention
 * <h2><center>&copy; Copyright (c) 2020 STMicroelectronics.
 * All rights reserved.</center></h2>
 * This software component is licensed by ST under BSD 3-Clause license,
 * the "License"; You may not use this file except in compliance with the
 * License. You may obtain a copy of the License at:
                       opensource.org/licenses/BSD-3-Clause
 */
/* Entry Point */
ENTRY(Reset Handler)
/* Highest address of the user mode stack */
estack = ORIGIN(RAM) + LENGTH(RAM); /* end of "RAM" Ram type memory */
Min Heap Size = 0x200; /* required amount of heap */
_Min_Stack_Size = 0x400; /* required amount of stack */
/* Memories definition */
MEMORY
               : ORIGIN = 0x20000000, LENGTH = 96K
  RAM
       (xrw)
  FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 512K
```

The continuation of the code excerpt shows the definition of sections.

```
/* Sections */
SECTIONS
{
  /* The startup code into "FLASH" Rom type memory */
  .isr vector :
  {
    . = ALIGN(4);
   KEEP(*(.isr vector)) /* Startup code */
    = ALIGN(4);
  } >FLASH
  /* The program code and other data into "FLASH" Rom type memory */
  .text :
  {
    = ALIGN(4);
                      /* .text sections (code) */
   *(.text)
                      /* .text* sections (code) */
/* glue arm to thumb code */
   *(.text*)
   *(.glue_7)
   *(.glue 7t)
                      /* glue thumb to arm code */
   *(.eh_frame)
   KEEP (*(.init))
   KEEP (*(.fini))
   = ALIGN(4);
    _etext = .;
                      /* define a global symbols at end of code */
  } >FLASH
  /* Constant data into "FLASH" Rom type memory */
  .rodata :
  {
    = ALIGN(4);
   *(.rodata)
                      /* .rodata sections (constants, strings, etc.) */
   *(.rodata*)
                      /* .rodata* sections (constants, strings, etc.) */
    = ALIGN(4);
  } >FLASH
  .ARM.extab
               : {
    = ALIGN(4);
    *(.ARM.extab* .gnu.linkonce.armextab.*)
    = ALIGN(4);
  } >FLASH
  .ARM : {
   = ALIGN(4);
      exidx start = .;
   *(.ARM.exidx*)
   __exidx_end = .;
    = ALIGN(4);
  } >FLASH
  .preinit_array
                    :
  {
    . = ALIGN(4);
    PROVIDE_HIDDEN (__preinit_array_start = .);
   KEEP (*(.preinit array*))
   PROVIDE HIDDEN ( preinit array end = .);
    = ALIGN(4);
  } >FLASH
  .init_array :
  {
    = ALIGN(4);
   PROVIDE HIDDEN ( init array start = .);
   KEEP (*(SORT(.init_array.*)))
    KEEP (*(.init_array*))
   PROVIDE_HIDDEN (__init_array_end = .);
    . = ALIGN(4);
  } >FLASH
```

```
57
```

```
.fini_array :
  {
    . = ALIGN(4);
   PROVIDE HIDDEN (
                      fini array start = .);
   KEEP (*(SORT(.fini array.*)))
   KEEP (*(.fini array*))
   PROVIDE_HIDDEN (__fini_array_end = .);
    = ALIGN(4);
  } >FLASH
  /* Used by the startup to initialize data */
  sidata = LOADADDR(.data);
  /* Initialized data sections into "RAM" Ram type memory */
  .data :
  {
   . = ALIGN(4);
   _sdata = .;
*(.data)
                        /* create a global symbol at data start */
                       /* .data sections */
                       /* .data* sections */
   *(.data*)
                       /* .RamFunc sections */
/* .RamFunc* sections */
   *(.RamFunc)
   *(.RamFunc*)
   . = ALIGN(4);
   _edata = .;
                       /* define a global symbol at data end */
  } >RAM AT> FLASH
  /* Uninitialized data section into "RAM" Ram type memory */
  = ALIGN(4);
  .bss :
  {
   /* This is used by the startup in order to initialize the .bss section */
   _sbss = .; /* define a global symbol at bss start */
_bss_start__ = _sbss;
   *(.bss)
   *(.bss*)
   * (COMMON)
    = ALIGN(4);
   _ebss = .; /:
_bss_end_ = _ebss;
                     /* define a global symbol at bss end */
  } >RAM
 /* User heap stack section, used to check that there is enough "RAM" Ram type memory left
*/
  ._user_heap_stack :
  {
    . = ALIGN(8);
   PROVIDE ( end = . );
   PROVIDE ( end = . );
   . = . + _Min_Heap_Size;
. = . + _Min_Stack_Size;
    = ALIGN(8);
  } >RAM
  /* Remove information from the compiler libraries */
  /DISCARD/ :
  {
   libc.a ( * )
   libm.a ( * )
   libgcc.a ( * )
  }
  .ARM.attributes 0 : { *(.ARM.attributes) }
}
```

#### 2.5.6.1 The ENTRY command defines the start of the program

The first instruction to execute in a program is defined with the ENTRY command.

#### Example:

/\* Entry Point \*/
ENTRY(Reset\_Handler)

The ENTRY information is used by GDB so that the program counter (PC) is set to the value of the ENTRY address when a program is loaded. In the example, the program starts to execute from Reset\_Handler when a step or continue command is given to GDB after a load.

*Note:* The start of the program can be overridden if the GDB script contains a monitor reset command after the load command. Then the code starts to run from reset.

# 2.5.6.2 Stack location

The stack location is normally used by the startup file using the \_estack symbol. The startup code normally initializes the stack pointer with the address given in the linker script. For Cortex<sup>®</sup>-M based devices, the stack address is also set at the first address in the interrupt vector table.

#### Example:

```
/* Highest address of the user mode stack */
_estack = ORIGIN(RAM) + LENGTH(RAM); /* end of "RAM" Ram type memory */
```

#### 2.5.6.3 Define heap and stack minimum sizes

It is common to define in the linker script the heap and stack minimum sizes to be used by the system.

#### Example:

```
_Min_Heap_Size = 0x200; /* required amount of heap */
Min_Stack_Size = 0x400; /* required amount of stack */
```

The values defined here are normally used later in the linker script to make it possible for the linker to test if the heap and stack fit in the memory. The linker can then issue an error if there is not enough memory available.

## 2.5.6.4 Specify memory regions

The memory regions are specified with names ORIGIN and LENGTH. It is common also to have an attribute list specifying the usage of a particular memory region, such as (rx) with "r" standing for read-only section and "x" for executable section. It is not required to specify any attribute.

Example:

```
/* Memories definition */
MEMORY
{
    RAM (xrw) : ORIGIN = 0x20000000, LENGTH = 96K
    FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 512K
}
```

## 2.5.6.5 Specify output sections (.text and .rodata)

The output sections define where the sections such as '.text', '.data' or others are located in the memmory. The example below tells the linker to put all sections such as .text, .rodata and others in the Flash memory region. The glue sections mentioned in the example are used by GCC if there are some mixed code in the program. For instance, the glue code is used if some Arm<sup>®</sup> code makes a call to thumb code or vice versa.

#### Example:

```
/* Sections */
SECTIONS
  /* The startup code into "FLASH" Rom type memory */
  .isr vector :
  {
    = ALIGN(4);
   KEEP(*(.isr vector)) /* Startup code */
    = \text{ALIGN}(4);
  } >FLASH
  /* The program code and other data into "FLASH" Rom type memory */
  .text :
  {
   . = ALIGN(4);
   *(.text)
                       /* .text sections (code) */
                       /* .text* sections (code) */
   *(.text*)
                      /* glue arm to thumb code */
/* glue thumb to arm code */
   *(.glue 7)
    *(.glue 7t)
   *(.eh frame)
   KEEP (*(.init))
   KEEP (*(.fini))
    . = ALIGN(4);
    etext = .;
                       /* define a global symbols at end of code */
  } >FLASH
```

#### 2.5.6.6 Specify initialized data (.data)

Initialized data values require extra handling as the initialization values must be placed in the Flash memory and the startup code must be able to initialize the RAM variables with correct values. The example below creates symbols \_sidata, \_sdata and \_edata. The startup code can then use these symbols to copy the values from Flash memory to RAM during program start.

#### Example:

```
/* Used by the startup to initialize data */
  _sidata = LOADADDR(.data);
  /* Initialized data sections into "RAM" Ram type memory */
  .data :
    . = ALIGN(4);
    _sdata = .;
*(.data)
                        /* create a global symbol at data start */
                       /* .data sections */
/* .data* sections */
    *(.data*)
                       /* .RamFunc sections */
    *(.RamFunc)
    *(.RamFunc*)
                        /* .RamFunc* sections */
    = ALIGN(4);
                        /* define a global symbol at data end */
    _edata = .;
  } >RAM AT> FLASH
```

# 2.5.6.7 Specify uninitialized data (.bss)

Uninitialized data values must be reset to 0 by the startup code: the linker script file must identify the locations of these variables. The example below creates symbols \_sbss and \_ebss. The startup code can then use these symbols to set the values of the uninitialized variables to 0.

### Example:

# 2.5.6.8 Check if user heap and stack fit in the RAM

One section of the code is normally dedicated to linker checks about the fact that the needed heap and stack fit into the RAM together with all other data.

#### Example:

```
/* User_heap_stack section, used to check that there is enough "RAM" Ram type memory left
*/
   ._user_heap_stack :
   {
        . = ALIGN(8);
        PROVIDE ( end = . );
        PROVIDE ( _end = . );
        . = . + _Min_Heap_Size;
        . = . + _Min_Stack_Size;
        . = ALIGN(8);
    } >RAM
```

Note: The stack is placed on top of RAM and heap after data with a gap in between. See Table 3. Memory map layout.

#### 2.5.6.9 Linker map and list files

When building a project generated with STM32CubeIDE, a map and a list file are created in the debug or release build output folders. These files contain detailed information on the final locations of code and data in the program.

The Build Analyzer view can be used to analyse the size and location of a program in detail. Read more about this in Section 8 Build Analyzer.

# 2.5.7 Modify the linker script

This section presents common use cases requiring to edit the linker script. Editing and managing the script allows for more exact placements of the code and data.

### 2.5.7.1 Place code in a new memory region

Many devices have more than one memory region. It is possible to use the linker script to specifically place code in different areas. The example below shows how to update a linker script to support code to be placed in a new memory region named IP CODE.

#### Example:

```
Original MEMORY AREA
/* Memories definition */
MEMORY
{
    RAM (xrw) : ORIGIN = 0x2000000, LENGTH = 96K
    FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 512K
}
Add IP_CODE into MEMORY AREA
/* Memories definition */
MEMORY
{
    RAM (xrw) : ORIGIN = 0x2000000, LENGTH = 96K
    FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 256K
    IP_CODE (rx) : ORIGIN = 0x8040000, LENGTH = 256K
}
```

Place the following code a bit further down in the script, between the .data  $\{ \dots \}$  and the .bss  $\{ \dots \}$  section in the linker script file:

Example:



This tells the linker to place all sections named .  $IP_Code*$  into the  $IP_CODE$  memory region, which is specified to start at target memory address 0x804 0000.

In the C code, tell the compiler which functions must go to this section by adding \_\_attribute\_\_((section(".IP\_Code"))) before the function declaration.

Example:

```
__attribute__((section(".IP_Code"))) int myIP_read()
{
    // Add code here...
    return 1;
}
```

The myIP read() function is now placed in the IP CODE memory region by the linker.

### 2.5.7.2 Place code in RAM

To place code in the RAM, some modifications of the linker script and startup code are needed. The example below describes the changes to be applied when the internal RAM is split into a few sections and the code is placed and executed in one of the internal RAM sections.

Define a new memory region in the MEMORY {} region in the linker script:

```
Original MEMORY AREA
/* Memories definition */
MEMORY
{
    RAM (xrw) : ORIGIN = 0x2000000, LENGTH = 96K
    FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 512K
}
Split RAM into memory areas RAM1, RAM_CODE, RAM
/* Memories definition */
MEMORY
{
    RAM1 (xrw) : ORIGIN = 0x2000000, LENGTH = 16K
    RAM_CODE (xrw) : ORIGIN = 0x20004000, LENGTH = 16K
    RAM (xrw) : ORIGIN = 0x20008000, LENGTH = 64K
    FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 512K
}
```

Define an output section for the code in the linker script. This must be placed with a Load Memory Address (LMA) belonging to the Flash memory, and a Virtual Memory Address (VMA) in RAM:

The RAM code area must be initialized and code copied from the Flash memory to the RAM code area. The startup code can access the location information symbols \_siram\_code, \_sram\_code and \_eram\_code. Add load address symbols for RAM CODE into the startup file:

```
/* Load address for RAM_CODE */
.word _siram_code;
.word _sram_code;
.word eram code;
```

Add a piece of code into the startup code to copy the RAM code from the Flash memory (LMA) to the RAM (VMA):

```
Reset Handler:
  ldr sp, =_estack
                                   /* set stack pointer */
/* Copy the ram code from flash to RAM */
 movs r1, #0
  b LoopRamCodeInit
RamCodeInit:
  ldr r3, =_siram_code
ldr r3, [r3, r1]
       r3, [r3, r1]
  str r3, [r0, r1]
  adds r1, r1, #4
LoopRamCodeInit:
  ldr r0, =_sram_code
ldr r3, =_eram_code
adds r2, r0, r1
  cmp r2, r3
 bcc RamCodeInit
/* Copy the data segment initializers from flash to SRAM */
  movs r1, #0
  b LoopCopyDataInit
CopyDataInit:
```

In the C code, instruct the compiler about which functions must go to this section by adding \_\_attribute\_\_((section(".RAM\_Code"))) before the functions declarations:

```
__attribute__((section(".RAM_Code"))) int myRAM_read()
{
    // Add code here...
    return 2;
}
```

Refer to [ST-12] for information on how to execute application code from CCM SRAM using STM32CubeIDE. It contains examples on how to setup the linker script and startup code to execute a function or an interrupt handler from RAM. The example in the chapter 4 of [ST-12] can be used as an inspiration on how to add other RAM regions and setup code sections to be located in RAM.

### 2.5.7.3 Place variables at specific addresses

It is possible to place variables at specific addresses in the memory. To achieve this, the linker script must be modified. The example presented in this section places constant variables handling a product <code>VERSION\_NUMBER</code>, CRC NUMBER, and <code>BUILD ID</code> in memory.

The first step is to create a new memory region in the linker script:

```
Original MEMORY AREA
/* Memories definition */
MEMORY
{
    RAM (xrw) : ORIGIN = 0x20000000, LENGTH = 96K
    FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 512K
}
Add a new 2K FLASH_V memory region at end of flash
/* Memories definition */
MEMORY
{
    RAM (xrw) : ORIGIN = 0x20000000, LENGTH = 96K
    FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 512K-2K
    FLASH_V (rx) : ORIGIN = 0x807F800, LENGTH = 2K
}
```

At this point, the memory section must be added:

This instructs the linker to place all sections named  $flash_v^*$  into the  $flash_v$  output section in the  $FLASH_v$  memory region, which is specified to start at target memory address  $0 \times 807$  F800.

A section can be called almost anything except some predefined names such as "data".

Now, the variables that must be located into the FLASH V memory must be defined with attributes in the C files:

\_\_attribute\_\_((section(".flash\_v.VERSION"))) const uint32\_t VERSION\_NUMBER=0x00010003; \_\_attribute\_\_((section(".flash\_v.CRC"))) const uint32\_t CRC\_NUMBER=0x55667788; \_\_attribute\_\_((section(".flash\_v.BUILD\_ID"))) const uint16\_t BUILD\_ID=0x1234;



When debugging this example and examining the memory, it can be observed that:

- Address 0x807 f800 contains VERSION NUMBER
- Address 0x807 f804 contains CRC NUMBER
- Address 0x807 f808 contains BUILD ID

#### Figure 89. Linker memory output

IDE					_		×
🚺 Memory 🖾			1010	1010 🗋 🛃	->1: 1:	•	
Monitors 🕂	**	0x807f800	: 0x807F800	<hex> 🔀</hex>	🕂 New Re	nderings	
0x807f800		Address	0 - 3	4 - 7	8 - B	C - F	^
		0807F800	03000100	88776655	3412FFFF	FFFFFFF	
		0807E810	FFFFFFF	FFFFFFF	FFFFFFFF	FFFFFFF	×

If the inserted data order in the Flash memory is important, map the order of the variables in the linker script. This makes it possible to define the variables in any file. The linker outputs the variables in the defined order independently on how the files are linked. As a result, if the CRC\_NUMBER is calculated in some way after the linker has built the file, the CRC\_NUMBER can be inserted into the Flash memory file by another tool:

```
Decide the order in the linker script by adding the specially named sections in order
BUILD_ID, VERSION_NUMBER, CRC_NUMBER, and others(*).
.flash_v :
{
    *(.flash_v.BUILD_ID*);
    *(.flash_v.VERSION*);
    *(.flash_v.CRC*);
    *(.flash_v*);
} > FLASH_V
```

When debugging this example and examining the memory, it can be observed that:

- Address 0x807 f800 contains BUILD\_ID
- Address 0x807 f804 contains VERSION NUMBER
- Address 0x807 f808 contains CRC NUMBER

### Figure 90. Linker memory output specified order



#### 2.5.7.4 Linking in a block of binary data

It is possible to link in a block of binary data into the linked file. The example below describes how to include a . . /readme.txt file.

#### Example:

```
File: readme.txt
Revision: Version 2
Product news: This release ...
```

One way to include this in the project is to make a reference in a C file to include it using the incbin directive and the allocatable ("a") option on the section:

```
asm(".section .binary_data,\"a\";"
".incbin \"../readme.txt\";"
);
```

The new section  $binary_data$  is then added into the linker script with instructions that the section must be put in the Flash memory. The KEEP() keyword can be used to surround an input section so that the linker garbage collector does not eliminate the section even if not called:

```
.binary_data :
{
    _binary_data_start = .;
    KEEP(*(.binary_data));
    _binary_data_end = .;
} > FLASH
```

This block can then be accessed from the C code:

```
extern int _binary_data_start;
int main(void)
{
   /* USER CODE BEGIN 1 */
   int *bin_area = &_binary_data_start;
```

The binary data, in this case the readme file, can be observed in the Memory view when the project is debugged.

IDE												×
🚺 Memory 🛛									1012 1010 恮 🛃	⇒ti 🔢 🕏	<b>•</b> 0 • • •	
Monitors 🛛 🕂 💥 💥	0x800261c	: 0x8002610	C <hex> 🖂</hex>	🕂 New Re	enderings		0x800261c <	Floating	0x800261c:0x8002	6 🔀 🕂	New Rende	erings
0x800261c	Address	0 - 3	4 - 7	8 - B	C - F	^	0x0800261C	656C6946	6572203A	File	: re	^
	08002610	9D220008	B9010008	95010008	46696C65		0x08002624	656D6461	7478742E	adme	.txt	
	08002620	3A207265	61646D65	2E747874	200D0A52		0x0800262C	520A0D20	73697665	R	evis	
	08002630	65766973	696F6E3A	20566572	73696F6E		0x08002634	3A6E6F69	72655620	ion:	Ver	
	08002640	2032200D	0A50726F	64756374	206E6577		0x0800263C	6E6F6973	0D203220	sion	2.	
	08002650	733A2054	68697320	72656C65	61736520		0×08002644	6F72500A	74637564	.Pro	duct	
	08002660	2E2E2E0D	0A030000	00000000	000024F4		0x0800264C	77656E20	54203A73	new	s: T	
	08002670	00100000	00010000	00180000	20000000		0x08002654	20736968	656C6572	his	rele	
	08002680	00040300	206C0300	20D40300	20000000		0x0800265C	20657361	0D2E2E2E	ase		
	08002690	00000000	00000000	00000000	00000000	~	0×08002664	0000030A	00000000			~

Figure 91. Linker memory displaying file readme

# 2.5.7.5 Locate uninitialized data in memory (NOLOAD)

There is sometimes a need to have variables located into the Flash, or some other non-volatile memory, which must not be initialized at startup. In such cases, it is possible to create a specific MEMORY AREA in the linker script (FLASH\_D) and use the NOLOAD directive in the section using the area.

#### Example:

```
The MEMORY AREA can be defined like this

/* Memories definition */

MEMORY

{

RAM (xrw) : ORIGIN = 0x20000000, LENGTH = 96K

FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 512K-4K

FLASH_D (rx) : ORIGIN = 0x807F000, LENGTH = 2K

FLASH_V (rx) : ORIGIN = 0x807F800, LENGTH = 2K

}
```

Add a section for FLASH\_D using the NOLOAD directive. This can be done using the following code a bit further down in the linker script:

Finally, data can be used somewhere in the program by adding a section attribute when declaring the variables that must be located in the FLASH\_D memory.

```
__attribute__((section(".flash_d"))) uint32_t Distance;
attribute_((section(".flash_d"))) uint32_t Seconds;
```

# 2.5.8 Include libraries

To include a library into a project:

- 1. Right-click the project where the library must be included in the *Project Explorer* view and select [**Properties**]
- 2. In the dialog, select [C/C++ Build]>[Settings]
- 3. Select the Tool Settings tab in the panel
- 4. Select [C Linker]>[Libraries]
- 5. Add the library name to the [Libraries] field.

Make sure the libray name is added and not the path. According to the GCC convention, the library name is its filename without the "lib" prefix and ".a" extension.

Example: for a library file named libmyLib.a, add the library name myLib.

If by any chance the library name do not comply with the GCC convention, the full library name can be entered, preceded by a colon ":".

Example: for a library file named STemWin524b\_CM4\_GCC.a, add the library name :STemWin524b\_CM4\_GCC.a.

- 6. In the [Library Paths] list, set the library location path. Do not include the name of the library in the path. Example: \${workspace\_loc:/myLib/Debug} is the path to the archive file of the library project myLib residing in the same workspace as the application project.
- 7. Enable [Place libraries in a linker group (-WI,--start-group \$(LIBS) -WI,--end-group)] if libraries need to be linked several times to resolve circular dependencies.

# Figure 92. Include a library



The source folders for the header files must also be added to the [Include paths] field:

- 1. Select [MCU GCC Compiler]>[Include paths]
- 2. press the [Add...] button and add the paths to the source folders for the header files in the library



#### Figure 93. Add library header files to the include paths

Note:

Libraries added by include paths are considered as static libraries because they are provided by external parties. The header files are not rescanned as the content must not have changed for external header files. If external libraries must be treated as normal source folders, the folders must also be added as source folders to the project.

Refer to Section 2.5.9 Referring to projects for more information if a project is referring to another project, a library or a normal project.

# 2.5.9 Referring to projects

Whenever a project is using code from another project, both projects must be referring to each other. For a project to refer to a specific build of another project:

- 1. Select instead [Project]>[Properties]
- 2. Select [C/C++ General]>[Paths and Symbols]
- 3. Open the *References* tab
- 4. select the [Configuration] that the current project is referring to

Note:

### Figure 94. Set project references

Properties for NUCLEO-F401R	E	—		×
type filter text	Paths and Symbols	Ç	• <> •	•
<ul> <li>&gt; Resource</li> <li>&gt; C/C++ Build</li> <li>&gt; C/C++ General</li> <li>&gt; Code Analysis         <ul> <li>&gt; Documentation</li> <li>File Types</li> <li>Formatter</li> <li>Indexer</li> <li>Language Mappings</li> <li>Paths and Symbols</li> <li>Preprocessor Include Patl</li> <li>CMSIS-SVD Settings</li> <li>Project Natures</li> <li>Project References</li> <li>Refactoring History</li> <li>Run/Debug Settings</li> </ul> </li> </ul>	Configuration:       Debug [Active]       ✓       N	Manage Configu References Expa Colla Move	ind All pse All ve Up e Down	^
< >	Restore Defau	ults Ap	ply	~
?	Apply and C	Close C	ancel	

When multiple projects are used as references, use the [Move Up] and [Move Down] buttons to setup the priorities.

There are many advantages to set project references correctly:

- The projects involved are not rebuilt more than necessary.
- The indexer is able to find functions from the library and open them. To use this possibility, press the **Ctrl** key and, in the editor, click the library function where it is used to open the library source file in the editor.
- It is possible to create the call hierarchy for the functions in the library. To find the call hierarchy, mark the function name and press **Ctrl+Alt+H** to display the call hierarchy in the *Call Hierarchy*view.

If a library project is added as a reference, all the correct settings in the *Paths and Symbols* property page for the library is set. The tool settings that depend on this property page are adjusted also.

This is the recommended method of adding libraries developed locally. For more information about adding libraries, refer to Section 2.5.8 Include libraries.

Another way to have projects referring to each other is as follows:

- 1. Select [**Project**]>[**Properties**]
- 2. Select [Project References]

3. Select and mark the project for reference

With this method, however, it is not possible to refer to different build configurations and libraries are not set up automatically.

# 2.6 I/O redirection

The C run time library contains many functions, including some to handle I/Os. The I/O-related run time functions include printf(), fopen(), fclose(), and many others. It is common practice to redirect the I/O from these functions to the actual embedded platform. For instance, the printf() output can be redirected to an LCD display or serial cable while file operations like fopen() and fclose() can be redirected to a Flash memory file system middleware.

# 2.6.1 printf() redirection

There are several ways to perform printf() redirection, such as using UART or SWV/ITM. Another solution is the Real-Time Transfer technology (RTT) provided by SEGGER. The three techniques compare as follows:



- The UART output is maybe the most commonly used method, where the output from the embedded system is sent for instance to a terminal using RS-232. It requires some CPU overhead and medium bandwidth.
- The Instrumentation Trace Macrocell (ITM) output is efficient but requires that the Arm<sup>®</sup> CoreSight<sup>™</sup> debugger technology with Serial Wire Viewer (SWV) is supported by the device. This is normally the case for Cortex<sup>®</sup>-M3, Cortex<sup>®</sup>-M4, Cortex<sup>®</sup>-M7, and Cortex<sup>®</sup>-M3 based devices. However, the SWV signals must be available and connected to the board also. It requires low CPU overhead but limited bandwidth. ITM output is explained in Section 4 Debug with Serial Wire Viewer tracing (SWV).
- The RTT solution is described by SEGGER on their website. RTT is a fast solution but requires SEGGER J-LINK debug probe.

To enable I/O redirection with UART or ITM output, the file <code>syscalls.c</code> must be included and built into the project. When <code>printf()</code> is used, it calls the <code>\_write()</code> function, which is implemented in <code>syscalls.c</code>. The <code>syscalls.c</code> file is normally created and included in the project when creating a new STM32CubeIDE project. The <code>\_write()</code> function in this file must be modified to enable <code>printf()</code> redirection by modifying the call to <code>\_\_io\_putchar()</code>. The way to modify <code>\_write()</code> depends on the hardware and library implementation.

The example below shows how to update <code>syscalls.c</code> so that <code>printf</code> ouput is redirected to ITM with an STM32F4 Series device. This is done by adding some header files to access <code>ITM\_SendChar()</code> and make a call to <code>ITM\_SendChar()</code>.

```
Original write() function
  _attribute__((weak)) int _write(int file, char *ptr, int len)
    int DataIdx;
    for (DataIdx = 0; DataIdx < len; DataIdx++)</pre>
    {
          io putchar(*ptr++);
    1
    return len:
}
Modified with added header files calling ITM SendChar(*ptr++);
#include "stm32f4xx.h"
#include "core cm4.h"
  attribute ((weak)) int write(int file, char *ptr, int len)
    int DataIdx;
    for (DataIdx = 0; DataIdx < len; DataIdx++)</pre>
    {
        // io putchar(*ptr++);
        ITM SendChar(*ptr++);
    return len;
```

It can be noticed that the \_write function in syscalls.c contains a weak attribute. This means that the \_write function can be implemented in any C file used by the project.



For instance, the new \_write() function can be added directly into main.c. Omit the weak attribute in that case, as shown in the example below.

```
int _write(int file, char *ptr, int len)
{
    int DataIdx;
    for (DataIdx = 0; DataIdx < len; DataIdx++)
    {
        //__io_putchar(*ptr++);
        ITM_SendChar(*ptr++);
    }
    return len;
}</pre>
```

# 2.7 Thread-safe wizard for empty projects and CDT<sup>™</sup> projects

STM32CubeIDE includes a thread-safe wizard to generate files to support the use of resources that can be updated by application code and interrupts or when using a real-time operating system.

The thread-safe wizard may only be used for STM32CubeIDE empty projects. For projects managed by STM32CubeMX, the thread-safe implementation configuration must be made using STM32CubeMX dialogs. The thread-safe wizard creates three files and adds the STM32\_THREAD\_SAFE\_STRATEGY define to the project. The files are:

- newlib\_lock\_glue.c
- stm32\_lock\_user.h
- stm32\_lock.h

First, in the example below, a myThreadSafe folder is created in the empty project. This folder is selected in the *Thread-Safe Solution* wizard so that files are generated in this folder.

Note:



Open the menu [File]>[New]>[Other...] to obtain the wizard selection window shown in Figure 95.

Figure 95. Select a wizard

DE Select a wizard		×
Select a wizard		Ď
Generate a set of files that provide the ST thread-safe solution.		
Wizards:		
type filter text		
> 🗁 General		
> 😕 C/C++		
> 🗁 Launch Targets		
> 🧀 Remote System Explorer		
SIM32 Project		
Thread-Safe Solution		
Openia     Next      Finish	Cara	al
< Back Next > Finish	Canc	ei



In the [ST] node select [Thread-Safe Solution] and press [Next >] to open the Thread-Safe Solution wizard.

Figure 96. Thread-Safe Solution wizard	Figure	96.	Thread-Safe	Solution	wizard
--	--------	-----	-------------	----------	--------

Generate T	Generate Thread-Safe Solution						
Thread-Safe		IDE					
<sup>88</sup> Folder 'STM3	32F401RE_empty' is not in a s	source folder.					
Source folder:	STM32F401RE_empty				Browse		
Strategy:					~		
?	< Back	Next >	Finish	Can	cel		

Press [Browse] to open the Generate Into Source Folder dialog.

Figure 97. Thread-safe source folder location

Generate Into Source Folder		$\times$
Select the folder to generate into.		
> IDE NUCLEO-F401RE		
✓ Image: STM32F401RE_empty		
🗁 .settings		
🗁 Inc		
🗁 myThreadSafe		
🗁 Src		
🗁 Startup		
OK	Cancel	

Select the source folder to generate the files into and press [OK].

The wizard proposes to select among five different thread-safe strategies:

- 1. User-defined thread-safe implementation.
- 2. Allow lock usage from interrupts.
- 3. Deny lock usage from interrupts.
- 4. Allow lock usage from interrupts. Implemented using FreeRTOS<sup>™</sup> locks.
- 5. Deny lock usage from interrupts. Implemented using FreeRTOS<sup>™</sup> locks.



The different strategies are explained in file stm32 lock.h.

```
* 1. User defined thread-safe implementation.
    User defined solution for handling thread-safety.
    NOTE: The stubs in stm32_lock_user.h needs to be implemented to gain
    thread-safety.
* 2. Allow lock usage from interrupts.
    This implementation will ensure thread-safety by disabling all interrupts
    during e.g. calls to malloc.
    NOTE: Disabling all interrupts creates interrupt latency which
    might not be desired for this application!
* 3. Deny lock usage from interrupts.
    This implementation assumes single thread of execution.
    Thread-safety dependent functions will enter an infinity loop
    if used in interrupt context.
* 4. Allow lock usage from interrupts. Implemented using FreeRTOS locks.
    This implementation will ensure thread-safety by entering RTOS ISR capable
    critical sections during e.g. calls to malloc.
    By default this implementation supports 2 levels of recursive locking.
    Adding additional levels requires 4 bytes per lock per level of RAM.
    NOTE: Interrupts with high priority are not disabled. This implies
    that the lock is not thread-safe from high priority interrupts!
* 5. Deny lock usage from interrupts. Implemented using FreeRTOS locks.
    This implementation will ensure thread-safety by suspending all tasks
    during e.g. calls to malloc.
    NOTE: Thread-safety dependent functions will enter an infinity loop
    if used in interrupt context.
```



Select a strategy as shown in Figure 98.

	Figure 98. Thread-safe strategy selection		
DE Generate T	hread-Safe Solution		×
Thread-Safe	Solution		IDE
Generate a se	t of files that provide the ST thread-safe solution.		
Source folder:	STM32F401RE_empty/myThreadSafe	Bro	wse
Strategy:	Generic Strategy #3 - Deny lock usage from interrupts		~
	Generic Strategy #1 - Custom implementation		
	Generic Strategy #2 - Allow lock usage from interrupts		
	Generic Strategy #3 - Deny lock usage from interrupts		
	FreeRTOS Strategy #4 - Allow lock usage from interrupts		
	FreeRTOS Strategy #5 - Deny lock usage from interrupts		
?	< Back Next > Finish	Cancel	

Select a strategy and press [Finish] to generate the files into the selected source folder.

The files generated are the same and contain the same information independently of the selected strategy.

A new define, STM32\_THREAD\_SAFE\_STRATEGY=3, is added by the wizard to the project for use by the preprocessor when building the project. The define value is set according to the strategy selected in the wizard. The define can be observed by opening the project properties and looking into the *Tool Settings* tab.

Properties for STN	132F401RE_empty			_		×
type filter text	Settings				⇔ ◄ ⇔	▼ 8
<ul> <li>Resource Builders</li> <li>C/C++ Build Build Variables</li> </ul>	Configuration: Debug [Active]		~	Manage Conf	igurations	<b>^</b>
Environment	🕲 Tool Settings 🎤 Build Steps 🍕	Build Artifact 🗟 Binary Parsers 🥺 Error Parsers				
Settings C/C++ General C/SIS-SVD Settir Project Reference: Run/Debug Settir	<ul> <li>MCU Toolchain</li> <li>MCU Settings</li> <li>MCU Post build outputs</li> <li>MCU GCC Assembler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>Miscellaneous</li> <li>MCU GCC Compiler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> </ul>	Define symbols (-D) DEBUG STM32 STM32F401RETx STM32F4 STM32_THREAD_SAFE_STRATEGY=3		ହି ର୍ଶ 	2 9 Şi (	
< >	Include paths A continuization	Undefine symbols (-U)		<b>e</b> 1	2 월 신 문	~
?			Apply an	d Close	Cancel	

# Figure 99. Thread-safe properties

The generated files are shown in the Project Explorer.

Note: The generated files require that some CMSIS header files are included in the project. The files must be copied and added manually into the project if they are missing.

# Figure 100. Thread-safe files

File	Edit	Source	Refac	tor I	Navi	gate	Searc	ch	Pro
<b>1</b>	. 8 (	è   🗞 🔻	≪ ▼	D i 1	<u> </u>	62 🤻	C ·	• @	<u>۽</u> ا
🔁 Pro	oject	Explorer	×			E \$	8		
> IDE	NUC	LEO-F40	1RE						
Y IDE	STM	32F401R	E_emp	oty					
>	றி Ir	ncludes							
>	😕 Ir	nc							
>	😕 S	rc							
>	😕 S	tartup							
~	😕 n	nyThread	Safe						
	> .	newlib	_lock_g	Jue.c					
	> .	🖻 stm32_	lock_u	ser.h					
	> .	🖻 stm32_	lock.h						
	🗟 S	TM32F40	D1RET)	(_FLA	SH.I	d			
	🗟 S	TM32F40	D1RET)	(_RAI	M.ld				

File Edit Source Refactor Navigate Search P

If the wizard is started while the project is managed by STM32CubeMX, an error is displayed stating that STM32CubeMX must be used to manage the thread-safe strategy.

	Figure 10	J1. Inread-sate erro	r dialog	
Generate Threa	d-Safe Solution		_	
Thread-Safe So	lution			IDE
Project is manag	ed by STM32CubeMX. U	se STM32CubeMX to	o manage the strategy.	
Source folder: NU	ICLEO-F401RE			Browse
Strategy:				~
?	< Back	Next >	Finish	Cancel

#### 2.8 Position-independent code

This section is of interest to users working on applications where the final address location is not defined in the system. This occurs for instance when using a bootloader: the system designer must be able to define the final location of the application. In such case, position-independent code (PIC) can be used. The -fple compiler option enables the compiler/linker to generate position-independent executable.

Compiling with option -fPIE generates position-independent executable so that if the application is linked for address 0x800 0000 but placed at 0x800 1000, it still runs.

However, the information in this section is not complete. The solution it describes works when using global data initialized to zero (.bss) but it does not work when using initialized data and has several other limitations. One such limitation is that run time libraries included in the STM32 toolchain cannot be used as these libraries are built without the -fPIE option for optimization. Instead of using position-independent code in a system, it is worth considering other solutions.

#### Alternate solution example:

If a system is designed to have a bootloader and multiple versions of an application in Flash memory at different slots, it may be easier to setup multiple build configurations for the application. Each build configuration is based on its own linker script file. In this case, there is no need to use position-independent code since run time libraries can be used. Each build configuration links the application to a unique slot in Flash memory, generating one single elf file per slot. When downloading a new version of the application into a slot, the correct elf file must be used. The bootloader can be designed to validate the addresses in the elf file and generate an error if it contains addresses that are out of the slot. The application can copy the interrupt vector table to RAM and update vector copies depending on the slot where the application is stored.

# 2.8.1 Adding the -fpie option

57

To add the -fple option into the tool settings:

- 1. Right-click the project in the *Project Explorer* view and select [Properties]
- 2. In the dialog, select [C/C++ Build]>[Settings]
- 3. Select the *Tool Settings* tab in the panel
- 4. Select [MCU GCC Compiler]>[Miscellaneous]
- 5. Add -fPIE to the [Other flags] field.

Properties for NUCLEO-F401RE			-	
type filter text	Settings			<> ▼ <> ▼ ▼
<ul> <li>Resource</li> <li>C/C++ Build</li> <li>Build Variables</li> <li>Discovery Options</li> <li>Environment</li> </ul>	Configuration: Debug [Active]		~ Manag	ge Configurations
Logging	🛞 Tool Settings 🎤 Build Steps	🚇 Build Artifact   🗟 Binary Parser	rs 🔞 Error Parsers	4
<ul> <li>&gt; C/C++ General</li> <li>CMSIS-SVD Settings</li> <li>Project Natures</li> <li>Project References</li> <li>Refactoring History</li> <li>Run/Debug Settings</li> </ul>	MCU Settings MCU Post build outputs MCU GCC Assembler Debugging Preprocessor Include paths Miscellaneous MCU GCC Compiler Beneral Debugging Preprocessor Include paths Optimization Warnings Miscellaneous MCU GCC Linker Beneral Libraries Miscellaneous	Other flags -fPIE ↓Verbose (-v) Position Independent Code (-fPIC) Enable stack usage analysis (-fstace)	k-usage) Restore Defaults	▲ 全 行 公
?			Apply and Close	Cancel

# Figure 102. Position independent code, -fpie

# 2.8.2 Run time library

The C run time library is compiled without using the -fPIE option. So any call to the library must be avoided when generating position-independent executable. The startup code normally has a call to libc init array. This call must be removed as in the example below:

```
/* Call static constructors */
/* bl __libc_init_array */
```

# 2.8.3 Stack pointer configuration

Make sure that the stack pointer is set up correctly. The stack pointer must be set in the Reset\_Handler in the startup file as shown in the example below. It must not be assumed that the stack pointer is set by a reset reading it from the vector table.

```
Reset_Handler:
ldr sp, =_estack /* set stack pointer */
```

# 2.8.4 Interrupt vector table

The vectors in the vector table must be updated if the program is loaded to an offset address. If a program needs to add the offset to each vector in the table, it can copy the interrupt vector table to the RAM and add the offset to this vector table.

The vector base register must also be changed so that it points to the new located vector table as shown in the example below:

```
/* Set Vector Base Address */
SCB->VTOR=RAM VectorTable;
```

# 2.8.5 Global offset table

The global offset table (GOT) is a table of addresses normally stored in the data section when building and using the -fPIE option. It is used by the executed program to find, during run time, addresses of global variables, unknown at compile time. If no global variable location change is needed, the variables can be located at same place as located when linking the program. Then the GOT table can be placed in the .text section in the Flash memory area instead.

The example below shows how to update the linker script with the .got\* section. In this case the  $GOT\_START$  and  $GOT\_END$  symbols are added also so that the tools are able to know the GOT location and size.

```
/* The program code and other data into "ROM" Rom type memory */
  .text :
  {
    = ALIGN(4);
   *(.text)
                      /* .text sections (code) */
    *(.text*)
                      /* .text* sections (code) */
   GOT START = .;
   *(.got*)
   GOT END = .;
    *(.glue 7)
                      /* glue arm to thumb code */
   *(.glue 7t)
                      /* glue thumb to arm code */
    *(.eh frame)
   KEEP (*(.init))
   KEEP (*(.fini))
    = ALIGN(4);
    etext = .;
                      /* define a global symbols at end of code */
```

# 2.8.6 Interrupt vector table and symbols

When debugging the code with an offset, both the load offset and the new symbol address must be specified. The symbol address to provide is the .text section address. The linker script can be updated by defining .isr vector to be located into .text. This avoids the issue of finding the location of .text.

```
Remove the following
  .isr vector :
  {
    . = ALIGN(4);
   KEEP(*(.isr_vector)) /* Startup code */
    = \text{ALIGN}(4);
  } >FLASH
Add KEEP(*(.isr_vector)) instead to first location of .text
  /* The program code and other data into "FLASH" Rom type memory */
  .text :
  {
    = ALIGN(4);
   KEEP(*(.isr_vector)) /* Startup code */
    *(.text) /* .text sections (code) */
*(.text*) /* .text* sections (code) */
   *(.text*)
   GOT START = .;
    *(.got*)
   GOT_END = .;
    *(.glue_7)
                        /* glue arm to thumb code */
    *(.glue_7t)
                        /* glue thumb to arm code */
    *(.eh frame)
   KEEP (*(.init))
   KEEP (*(.fini))
    = ALIGN(4);
     etext = .;
                      /* define a global symbols at end of code */
  } >FLASH
```

# 2.8.7 Debugging position-independent code

When debugging position-independent code located at an offset, the download offset and new symbol address must be set.

2 🕅 🍋 🗮 🗮 🚍 🖈 🕶	Name: NUCLEO-F401RE.elf				
Y I (I) (I) (I) (I) (I) (I) (I) (I) (I) (	Name:       NUCLEO-F401RE.elf         Main ** Debugger       Startup ** Source       Common         Initialization Commands       Initialization Commands         Ecoad Image and Symbols       File         > Debug/NUCLEO-F401RE.elf [NUCLEO-F401RE]       0x1000 symbols © 0x8001000         Runtime Options       Set program counter at (hex):         Set breakpoint at:       main         Exception on divide by zero       Exception on unaligned access         Halt on exception       Resume         Run Commands       set \$pc=Reset_Handler	Build See Main tab	Download true ti item erties of this item to in the "Main" tab NUCLEO NUCLEO th: Debug/I th: Debug/I load offset (hex) [	Load symbols true	Add Edit Remove Move up Move dow
			OK	Cancel	Arris

# Figure 103. Debugging position independent code

Figure 103 illustrates an example where the download offset is 0x1000 and the symbol address is 0x800 1000. It is possible to set the symbol address to 0x800 1000 in this case because the .isr\_vector is added into the .text section as described in Section 2.8.6 Interrupt vector table and symbols.

If instead the <code>.isr\_vector</code> is located in another section outside <code>.text</code>, the start address of the <code>.text</code> section must be used with the offset added. For instance, if the map file states that <code>.text</code> starts at <code>0x0000 0000 0800 0194</code>, the symbol address in this case must be set to <code>0x800 1194</code>.

Figure 103 shows that the breakpoint is set at main and that the program counter (\$pc) is set to the Reset\_Handler symbol into [Run Commands]. This symbol contains the correct address to the Reset\_Handler because gdb uses the base symbol address 0x800 1000. If \$pc is not setup during this debug configuration, the [Resume] checkbox must be disabled to make the program stop after load. In this case, the program counter must be set manually in the *Registers* view before starting the program.

57/

# 2.9 Exporting projects

A project can be exported in many different ways. This section shows how to export a project as a compressed zip file.

Right-click the project in the *Project Explorer* view and select [Export...].

# Figure 104. Export project

▼ INOCLEO-F4	New	>
	Go Into	
✓ <sup>™</sup> Core	Open in New Window	
> 🗁 Inc 📲	Сору	Ctrl+C
> 🔁 Src 🥤	Paste	Ctrl+V
🗸 🗁 Startı 🔰	Collete	Delete
> 🔝 st	Source	>
📄 st	Move	
📄 st	Rename	F2
> 😕 Drivers 💦	Import	
> 🔁 Debug 🍃	Export	
> 🔁 Debug2 📛		
> 🔁 Release		
	_ Refresh	F5
	Close Project	
	Close Unrelated Projects	
	Build Configurations	>
	Build Targets	>
	Index	>
STM32F	Show in Remote Systems view	
> DE STM32F40	Run As	>
> 🗖 STM32MP 🔻	🌣 Debug As	>
> 🗀 SW4STM32	Profile As	>
	Team	>
	Compare With	>
	Restore from Local History	
4	Generate Code	
c	靠 Convert to C++	
*	🦻 Run C/C++ Code Analysis	
	Configure	>
	Properties	Alt+Enter

# The Export dialog opens. Select [General]>[Archive File] and press [Next >].

Export		
elect		R
Export resources to an archive	e file on the local file system.	Ľ
Select an export wizard:		
type filter text		
<ul> <li>✓ General</li> <li>↓ Archive File</li> <li>↓ File System</li> <li>↓ Preferences</li> <li>&gt; ∠ C/C++</li> <li>&gt; ▷ Install</li> <li>&gt; ▷ Remote Systems</li> <li>&gt; ▷ Run/Debug</li> <li>&gt; ▷ Team</li> </ul>		

# Figure 105. Export dialog

The *Export* dialog is updated. Select the project to be exported. It is possible to exclude some project files from the export. In the example in Figure 106, all project and library files are included. A file name must be entered into the [**To archive file**] field, possibly browsing to a folder location for the file with the [**Browse..**] button. In the example, the default options values are kept unchanged. Press [**Finish**] to export the project and create the zip file.

IDE Export				- 🗆	×	
Archive file Export resources to an archive file on the l	ocal file system.			-	Ţ	
					_	
		cproject			^	
RemoteSystemsTempFiles		.project				
> 📴 STM32F401_empty		NUCLEO-F401	IRE.elf.cfg			
> C STM32F4xx-Nucleo		NUCLEO-F401	IRE.elf.launch			
>06 STM32MP151C_MP0_CM4		NUCLEO-F401	IRE.ioc			
		STM32F401R	TX_FLASH.id TX_FLASH_IPCodeInFla	ash.ld		
		✓ 1 STM32F401RE	ETX_FLASH_ORG.Id	ishina -	~	
Filter Types Select All	Deselect All					
To archive file: ive - STMicroelectronics\d	lev\CubeIDE\Manua	ls\STM32CubeIDE_UG	\nucleo-f401.zip V	Browse	·	
Options						
• Save in zip format		Create directory	structure for files			
○ Save in tar format		O Create only selected directories				
$\bigcirc$ Compress the contents of the file						
Resolve and export linked resources						
?	< Back	Next >	Finish	Cance	əl	

### Figure 106. Export archive

# 2.10 Importing existing projects

This section describes different ways to import existing projects into an STM32CubeIDE workspace. The standard Eclipse<sup>®</sup> importer is capable of importing Eclipse<sup>®</sup> projects. This is used to import projects created with STM32CubeIDE. The project importer is also extended to support the import of ac6 System Workbench for STM32 projects and Atollic<sup>®</sup> TrueSTUDIO<sup>®</sup> projects. Such projects are converted during the import phase to STM32CubeIDE projects.

It is possible to import and debug an existing elf file developed by another IDE or toolchain. More information on how this is done is available in Section 3.8.

# 2.10.1 Importing an STM32CubeIDE project

A project can be imported in many different ways. This section shows how to import a project that was exported as a compressed zip file.

- One way to open the Import dialog is to use the menu [File]>[Import...]
- Another way is to right-click the Project Explorer view and select [Import...]

# Figure 107. Import project



Figure 108. Import dialog



# Figure 109. Import projects

IDE Import		_		×
<b>Import Projects</b> Select a directory to search for existing Eclipse projects.				7
<ul> <li>○ Select root directory:</li> <li>● Select archive file:</li> <li>DE\Manuals\STM32CubeIDE_UG\Projects\nucleo-f401.zip</li> </ul>	~	B	rowse	
✓ myLib (myLib/) ✓ NUCLEO-F401RE (NUCLEO-F401RE/)		Se De: R	elect All select Al Refresh	
Options          Search for nested projects         Copy projects into workspace         Close newly imported projects upon completion         Hide projects that already exist in the workspace				
Working sets Add project to working sets Working sets:		Ne	ex	
Contract		C	Cancel	

# 2.10.2 Importing System Workbench and TrueSTUDIO<sup>®</sup> projects

To import an ac6 System Workbench for STM32 project or an Atollic<sup>®</sup> TrueSTUDIO<sup>®</sup> project into STM32CubeIDE, it is advised to work on a project copy:

- 1. Create a copy of the project, either as a copy of the project folder or an export of the project in a zip file
- 2. Use the copied project for the import into STM32CubeIDE

The way to import the copied project is to open the *Import* dialog by means of the menu [File]>[Import...] or by right-clicking the *Project Explorer* view.
Select [Import ac6 System Workbench for STM32 project] or [Import Atollic TrueSTUDIO project] depending on the original tool used to create the project and press [Next >].

Figure 110. Import System workbench projects (1 of 5)					
IDE Import			×		
Select Import System Workbench for STM32 projects		Ľ	5		
Select an import wizard:					
type filter text					
<ul> <li>Archive File</li> <li>Existing Projects into Workspace</li> <li>File System</li> <li>Import ac6 System Workbench for STM32 Project</li> <li>Import an Existing STM32CubeMX Configuration File (.ioc)</li> <li>Import Atollic TrueSTUDIO Project</li> <li>Import STM32Cube Example</li> <li>Preferences</li> <li>Projects from Folder or Archive</li> <li>C/C++</li> <li>Install</li> <li>Remote Systems</li> <li>Run/Debug</li> <li>Team</li> </ul>			<		
Sack Next > Finish		Cance	I		

In this example, the ac6 project is copied into the STM32CubeIDE workspace, hence the [Directory...] button is used and project STM32F401\_Ac6 is selected. The import wizard detects that this is a System Workbench project.

Import Projects from File System or Archive					
Import Projects from File System or Archive					
This wizard analyzes the content of your folder or archive file to find projects and import t	hem in the IDE.				
Import source: C:\Users\johansse\STM32CubeIDE\workspace_um1\STM32F401_Ac6			~	Directory Are	:hive
type filter text				Select A	II
Folder	Import as			Deselect .	All
STM32F401_Ac6	Convert 'System Work!	pen			
				1 of 1 selected	
				Hide already op	en projects
Lise installed project configurators to:					
Search for nested projects					
Detect and configure project natures					
Working sets					
Add project to working sets				Nev	v
Working sets:				<ul> <li>✓ Sele</li> </ul>	ct
			-	Show other specialized imp	oort wizards
(?)		< Back	Next >	Finish	ancel
		- buck			

## Figure 111. Import System Workbench projects (2 of 3)

#### Press [Finish] to open the Project converter dialog.

## Figure 112. Import System Workbench projects (3 of 3)



Press [OK] to convert the project to an STM32CubeIDE project.

There are two migration guides explaining how to migrate from ac6 System Workbench for STM32 ([ST-06]) and Atollic<sup>®</sup> TrueSTUDIO<sup>®</sup> to STM32CubeIDE ([ST-05]). These guides can be opened from the *Technical Documentation* page in the *Information Center*.

## 2.10.3 Importing using project files association

When STM32CubeIDE is started, a pop-up window asks if .cproject and .project files must be associated to the program.

#### Figure 113. Import using project files association

DE Overr	ide		×
?	The .project file extension is currently associated to anot (C:\ST\STM32CubeIDE_1.1.0.19w38_targetplatform_2019- Would you like to associate the .project files to the STM3	her STM32CubelDE ir 09\STM32CubelDE\st 32CubelDE instance ir	nstallation tm32cubeide.exe). n use?
Rem	ember my decision		
		Yes	No

If the association is selected, double-clicking on a .project file in the personal computer file browser triggers the project import by STM32CubeIDE into the current workspace. The project converter investigates the project, which is imported directly if made for STM32CubeIDE. If the project comes from another tool, the project converter tries to identify if it is a known project format and, in such case, converts the project to an STM32CubeIDE project as described in Section 2.10.2 Importing System Workbench and TrueSTUDIO projects.

#### 2.10.4 Prevent "GCC not found in path" error

When importing old projects, an error in the *Problems* view can state "*Program* "gcc" not found in PATH". The error is caused by the project use of a deprecated discovery method setting. The error can be removed by updating the *Window Preferences* and *Project Properties* settings.

- 1. Open [Window]>[Preferences]. In the *Preferences* dialog, select [C/C++]>[Property Pages Settings] and enable checkbox [Display "Discovery Options" page].
- Open [Project Properties]>[C/C++ Build]>[Discovery Options] and disable checkbox [Automate discovery of paths and symbols].

## 2.11 Toolchain Manager

The *Toolchain Manager* is used to install toolchains, uninstall toolchains and select the default workspace toolchain when building a project.

To open the Toolchain Manager from the Tool Settings tab in project properties:

1. Select the [MCU Toolchain] node

Properties for NUCLEO-	F401RE					×
type filter text		1	<> ▼ <	> ▼ 8		
<ul> <li>Resource</li> <li>Builders</li> <li>C/C++ Build</li> <li>Build Variables</li> </ul>	Configuration: Debug [Active]		~	Manage Co	onfigurations	·
Logging	😻 Tool Settings 🎤 Build Steps	😤 Build Artifact  🗟 Binary Parse	rs 🤨 Ei	rror Parsers		
Settings > C/C++ General CMSIS-SVD Settings Project References Run/Debug Settings	Settings          • MCU Toolchain        Select what toolchain to use          C/C++ General         CMSIS-SVD Settings           • MCU Toolchain        Select what toolchain to use          WCU Settings          • MCU Toolchain           • MCU Toolchain           • Use workspace setting (GNU Tools for          Project References           • MCU GCC Assembler           • Fixed GNU Tools for STM32 (9-2020          Run/Debug Settings          • Preprocessor           • Configure the workspace toolchain, and          Open Toolchain Manager          • Open Toolchain Manager			TM32 (9-202 2-update) anage install	0-q2-update	
	<ul> <li>MCU GCC Compiler</li> </ul>					~
?		4	Apply and	d Close	Cancel	

## Figure 114. Open Toolchain Manager

2. Click on [Open Toolchain Manager...]

The *Toolchain Manager* can also be opened from the [Window]>[Preferences] menu:

1. Select [STM32Cube]>[Toolchain Manager]

## Figure 115. Toolchain Manager

IDE Preferences						$\Box$ $\times$
type filter text		Toolchair	n Manager	1		
> Help > Install/Update	^	The Toolch	nain Manager allows users to configure the wor	kspace toolchain, and manage	e installed toolchains.	1 111
Remote Development		Default	Name	State		Add Local
Remote Systems			GNU ARM Embedded (7-2018-q2-update)	Not installed		Edit Local
> Run/Debug			GNU ARM Embedded (9-2020-q2-update)	Not installed		Edit Local
✓ STM32Cube			GNU Tools for STM32 (7-2018-q2-update)	Not installed		Uninstall
Build		⊳	GNU Tools for STM32 (9-2020-q2-update)	Installed with STM32CubeID	DE 1.6	Delevel
Device Configuration Tool						Reload
End User Agreements						Set Default
File Association						
Firmware Updater						
MPU Serial						
Serial Wire Viewer						
larget Status		By default	let STM32CubeIDE select the toolchain to use.			
		Note: inst	alling and uninstalling toolchains requires a res	tart of STM32CubeIDE.		
Torminal			5 5 1			
Termina	$\checkmark$				1	lestore Defaults
					Apply and Close	Cancel

## The columns in the Toolchain Manager are described in Table 4.

## Table 4. Toolchain Manager column details

Name	Description					
Default	<ul> <li>A green/grey arrow symbol indicates the default workspace toolchain.</li> <li>The arrow color is:</li> <li>green when the toolchain is manually set as default by the user</li> <li>grey when the toolchain is selected as default by STM32CubeIDE logic</li> </ul>					
Name	The name of the toolchain.					
State	The state of the toolchain. Toolchains available for download from STMicroelectronics online repository are listed as <i>"installed"</i> or <i>"not installed"</i> . Local toolchains added by the user are listed as <i>"local"</i> .					

The buttons in the *Toolchain Manager* are described in Table 5.

## Table 5. Toolchain Manager button information

Name	Description
Add Local	Add reference to local toolchain.
Edit Local	Edit reference to local toolchain.
Install Uninstall Remove	<ul> <li>The button text depends on the type of the selected toolchain. It is used to:</li> <li>Install / Uninstall the selected toolchain provided by the repository</li> <li>Remove the selected local toolchain</li> </ul>
Reload	Reload the toolchain list from the repository.
Set Default	Set selected toolchain to be used by default.
Restore Defaults	Restore and use the default toolchain.
Apply and Close	Apply selection and close dialog.
Cancel	Cancel dialog.

## 2.11.1 Install new toolchain

Open the Toolchain Manager to install a new toolchain.

## Figure 116. Install toolchain

DE Preferences				$ \square$ $\times$
type filter text	Toolcha	in Manager		↓ ↓ ↓ 8
> Help	The Tool	chain Manager allows users to configure the w	installed toolchains.	
<ul> <li>Remote Development</li> </ul>	Default	Name	State	Add Local
<ul> <li>Remote Systems</li> <li>Run/Debug</li> </ul>		GNU ARM Embedded (7-2018-q2-update) GNU ARM Embedded (9-2020-q2-update)	Not installed Not installed	Edit Local
✓ STM32Cube		GNU Tools for STM32 (7-2018-q2-update)	Not installed	Install
Build Device Configuration Tool	⊳	GNU Tools for STM32 (9-2020-q2-update)	Installed with ST	Reload
End User Agreements File Association Firmware Updater MPU Serial Serial Wire Viewer				Set Default
Target Status Toolchain Manager > Team Terminal	By defau Note: ins	It let STM32CubeIDE select the toolchain to use stalling and uninstalling toolchains requires a re	2. estart of STM32CubeIDE.	Restore Defaults
? <u>} /</u>			Apply and	Close Cancel

Select the toolchain to install and click on [Install...]. The Install dialog opens and displays the items to be installed.

#### Figure 117. Check items to install

IDE Install		— 🗆 X
Install		3
Check the items that you wish to install.		
Name	Version	ld
STM32CubeIDE MCU External Tools GNU Tools for STM32 - Binaries (7-2018-q2-update)	1.5.0.202011040924	com.st.stm32cube.ide.feature.mcu.externaltools
Select All Deselect All		
Details		
STMicroelectronics STM32CubeIDE MCU External Tools GNU Tools for STM32 - Binaries (7-20	18-q2-update)	0
2	< Pack	Next > Einich Cancel
₩ 		

Check the items to install and click on [Next].

## Figure 118. Review items to install

Install					-		×
Install Details							
Review the items to be installed.							
Name	Version	ld			1 1 11		
STM32CubeIDE MCU External Tools GNU	Too 1.5.0.202011040924	com.st.stm32cub	e.ide.feature.mcu.	externaltools.gnu_to	ools_for_stm32.7_201	8_q2_upda	
STM32CubeIDE MCU External Tools GN	U <sup>-</sup> 1.5.0.202011040924	com.st.stm32cub	e.ide.feature.mcu.	externaltools.gnu_to	pols_for_stm32.7_201	8_q2_upda	
Details							
STMicroelectronics STM32CubeIDE MCU Extern	al Tools GNU Tools for STM	132 - Binaries (7-20	18-q2-update)				~ ~
0			< Back	Next >	Finish	Cance	el

#### Review the items and click on [Next].

## Figure 119. Review and accept licenses

Install	- 0	$\times$
Review Licenses Licenses must be reviewed and accepted before the software can b	e installed.	2
Licenses:	License text:	
> STMicroelectronics Software License Agreement	STMicroelectronics Software License Agreement SLA0048 Rev4/March 2018	^
	BY INSTALLING COPVING, DOWNLOADING, ACCESSING OR OTHERWISE USING THIS SOFTWARE PACKAGE OR ANY PART THEREOF (AND THE RELATED DOCUMENTATION) FROM STMICROELECTRONICS INTERNATIONAL N.V., SWISS BRANCH AND/OR ITS AFFILIATED COMPANIES (STMICROELECTRONICS), THE RECIPIENT, ON BEHALF OF HIMSELF OR HERSELF, OR ON BEHALF OF ANY ENTITY BY WHICH SUCH RECIPIENT IS EMPLOYED AND/OR ENGAGED AGREES TO BE BOUND BY THIS SOFTWARE PACKAGE LICENSE AGREEMENT. Under STMicroelectronics' intellectual property rights and subject to applicable licensing terms for any third-party software incorporated in this software package and applicable Open Source Terms (as defined here below), the redistribution, reproduction and use in source and binary forms of the software package or any part thereof, with or without modification, are permitted provided that the	5
	<ul> <li>I accept the terms of the license agreement</li> <li>I do not accept the terms of the license agreement</li> </ul>	
Ø	< Back Next > Finish Cancel	

## Review the licenses, select [I accept the terms of the license agreements] and click on [Finish].

At this point, the software installation starts. The progress bar displayed at the bottom of the STM32CubeIDE window shows the installation completion rate. Wait until the installation is completed.

## The following warning may appear before the installation is finished.

Figure	120.	Security	warning
--------	------	----------	---------



In this case, to finalize the installation, click on [Install anyway]. After some time, the following dialog is displayed.

Figure 121. Restart to apply software update

IDE Softw	are Updates	$\times$
?	Restart STM32CubeIDE to apply the software update?	
	Restart Now No	

Click on [Restart Now] to be able to use the installed toolchain in STM32CubeIDE. STM32CubeIDE is restarted and the new toolchain can be used.

Open the Toolchain Manager to verify the installation.

## Figure 122. Toolchain installed

IDE Preferences						×
type filter text	Toolchai	n Manager			⇔ ▼ ⇔ ▼	8
> General > C/C++	The Toolch	hain Manager allows users to configure the w	orkspace toolchain, and manage insta	alled toolcl	hains.	
> Help	Default	Name	State		Add Loca	al
Install/Update		GNU ARM Embedded (7-2018-q2-update)	Not installed		EditLoc	
Remote Development		GNU ARM Embedded (9-2020-q2-update)	Not installed		Eult Loca	d1
Remote Systems		GNU Tools for STM32 (7-2018-q2-update)	Installed		Uninstal	II
> Run/Debug	$\triangleright$	GNU Tools for STM32 (9-2020-q2-update)	Installed with STM32CubeIDE 1.6			
✓ STM32Cube					Reload	1
Build					Set Defa	ult
Device Configuration Tool						
End User Agreements						
File Association						
Firmware Updater						
MPU Serial						
Serial Wire Viewer						
Target Status	By defaul	t let STM32CubeIDE select the toolchain to use	2.			
Toolchain Manager	Note: inst	alling and uninstalling toolchains requires a re	estart of STM32CubeIDE.			
> Team		5 5 7		D		1
Terminal				Res	store Defa	ults
? è Z			Apply and Close		Cancel	

In this case, Figure 122 shows that two versions of GNU Tools for STM32 are installed.

## 2.11.2 Manage default toolchain

The Toolchain Manager highlights the default workspace toolchain with an arrow in the Default column.

Preferences			-			
type filter text	Toolchai	n Manager		← ← ⇒ < §		
> General > C/C++	The Toolc	hain Manager allows users to configure the w	orkspace toolchain, and manage installe	d toolchains.		
> Help	Default	Name	State	Add Local		
<ul> <li>Install/Update</li> <li>Remote Development</li> </ul>		GNU ARM Embedded (7-2018-q2-update) GNU ARM Embedded (9-2020-q2-update)	Not installed Not installed	Edit Local		
Remote Systems		GNU Tools for STM32 (7-2018-q2-update)	Installed	Uninstall		
<ul> <li>Run/Debug</li> <li>STM32Cube</li> </ul>		GNU Tools for STM32 (9-2020-q2-update)	Installed with STM32CubeIDE 1.6	Reload		
Build				Set Default		
Device Configuration Tool End User Agreements File Association Firmware Updater MPU Serial Serial Wire Viewer						
Target Status Toolchain Manager > Team Terminal	By default let STM32CubeIDE select the toolchain to use. Note: installing and uninstalling toolchains requires a restart of STM32CubeIDE. Restore Defaults					
? > 4			Apply and Close	Cancel		

#### Figure 123. Default toolchain

Figure 123 shows that *GNU Tools for STM32 version 9-2020-q2-update* is the default workspace toolchain. The *GNU Tools for STM32 version 7-2018-q2-update* line is marked in blue, which indicates that this toolchain selected. Any line in the table can be selected with the mouse.

Click on [Set default]: the selected toolchain to be used as the default workspace toolchain is highlighted with an arrow symbol in the *Default* colum of the *Toolchain Manager*.

#### Figure 124. Default toolchain updated

IDE Preferences				
type filter text	Toolchai	n Manager		
> General	The Toolc	hain Manager allows users to configure the w	orkspace toolchain, and manage installe	d toolchains.
> C/C++			g	
> Help	Default	Name	State	Add Local
Install/Update		GNU ARM Embedded (7-2018-q2-update)	Not installed	Edit Local
Remote Development		GNU ARM Embedded (9-2020-q2-update)	Not installed	Luit Local
Remote Systems	•	GNU Tools for STM32 (7-2018-q2-update)	Installed	Uninstall
> Run/Debug		GNU Tools for STM32 (9-2020-g2-update)	Installed with STM32CubeIDE 1.6	
✓ STM32Cube		, , , , , , , ,		Reload
Build				Set Default
Device Configuration Tool				See Deradit
End User Agreements				
File Association				
Firmware Updater				
MPU Serial				
Serial Wire Viewer				
Target Status	By defaul	t let STM32CubeIDE select the toolchain to us	0	
Toolchain Manager	by defuul			
> Team	Note: inst	calling and uninstalling toolchains requires a r	estart of STM32CubeIDE.	
Terminal				Restore Defaults
? à ú			Apply and Close	Cancel

Click on [Apply and Close] to apply the setting and update which toolchain is set to be the default workspace toolchain.

57

## 2.11.3 Uninstall toolchain

It is not possible to uninstall the *GNU Tools for STM32* toolchain, which is installed by default with STM32CubeIDE. Any other installed toolchain can be uninstalled.

## Figure 125. Uninstall toolchain

Preferences						۲
type filter text		Toolc	nain Manager			ŝ
<ul> <li>&gt; Help</li> <li>&gt; Install/Update</li> <li>&gt; Remote Development</li> </ul>	^	The To manag	olchain Manager allows users to confi e installed toolchains.	gure the workspace t	oolchain, and	
<ul> <li>Remote Systems</li> </ul>		Def	Name	State	Add Loca	al
Run/Debug			GNU ARM Embedded (7-2018-q2	Not installed	E-Pellener	1
✓ STM32Cube			GNU ARM Embedded (9-2020-q2	Not installed	Edit Loca	
Build			GNU Tools for STM32 (7-2018-q2-u	Installed	Uninstal	I
Device Configuration Tool			GNU Tools for STM32 (9-2020-q2-u	Installed with ST	Dalaad	
End User Agreements					Reload	1
File Association					Set Defa	ult
Firmware Updater						
MPU Serial						
Serial Wire Viewer		By def	ault let STM32CubeIDE select the tool	chain to use.		
Target Status		Note:	installing and uninstalling toolchains r	equires a restart of ST	M32CubeIDE.	
Terminal					Destave Defe	
lerrinia	$\mathbf{\vee}$				Restore Defai	ult
			[	Apply and Close	Cancel	

# To uninstall a toolchain, select it in the *Toolchain Manager* and click on [**Uninstall...**]. This opens the *Uninstall* dialog.

#### Figure 126. Uninstall details

Uninstall Uninstall Details Review and confirm the items to be uninstalled.					×
Name  STM32CubeIDE MCU External Tools GNU Tools for STM32 - Binaries (7-2018-q2-	update)	Version 1.5.0.202011040924	ld com.st.stm32cube.id	e.feature	
Details					
0	< Ba	ck Next >	Finish	Car	ncel

## Click on [Finish] to start the software uninstallation. The Software Updates dialog is displayed.

## Figure 127. Software updates

DE Softw	vare Updates	×
?	Restart STM32CubeIDE to apply the software update?	
	Restart Now	No

Click on [**Restart Now**] to apply the software update. The product is restarted. Open the *Toolchain Manager* to verify the installation.

#### Figure 128. Toolchain uninstalled

IDE Preferences							×
type filter text		Toolchai	n Manager			⇔ ◄ ⇔	<b>▼</b> 8
<ul> <li>C/C++</li> <li>Help</li> <li>Install/Update</li> </ul>	^	The Toolch toolchains	nain Manager allows users to configure the w 	vorkspace toolchain, and	manage	installed	
Remote Development		Default	Name	State		Add Lo	ocal
Remote Systems			GNU ARM Embedded (7-2018-q2-update)	Not installed		Edit Lo	ocal
> Run/Debug			GNU ARM Embedded (9-2020-q2-update)	Not installed			
✓ STM32Cube			GNU Tools for STM32 (7-2018-q2-update)	Not installed		Unins	tall
Build			GNU Tools for STM32 (9-2020-q2-update)	Installed with ST		Dala	a d
Device Configuration Tool						Reio	ad
End User Agreements						Set De	efault
File Association							
Firmware Updater							
MPU Serial							
Serial Wire Viewer							
Target Status		By default	t let STM32CubeIDE select the toolchain to us	ρ			
Toolchain Manager		Note: inst	alling and uninstalling toolchains requires a	e.			
> Team		Note. unst	alling and uninstalling toolchains requires a r	estant of STMSZCuberDE			
Terminal	$\checkmark$				R	estore De	faults
? è 4				Apply and Close	2	Cancel	

In this case, Figure 128 shows that only one version of GNU Tools for STM32 is installed.

## 2.11.4 Using local toolchain

It is possible to add and use an already installed local GNU ARM toolchain. To add a local toolchain, follow the steps below:



1. Open *Toolchain Manager* and press the [Add Local...] button.

Figure	129.	Add	local	toolchain

IDE		_		$\times$
Add loca Give th	<b>il toolchain</b> e toolchain a name			
Name:				
Prefix:	arm-none-eabi-			
Location:				Browse
?	1	Finish	Can	cel

## 2. Add a name and specify location.

Figure 130. Specify local toolchain locatio	Figure 130	. Specify	local too	Ichain	locatio
---	------------	-----------	-----------	--------	---------

IDE	—		$\times$
Add loca	al toolchain		
The fol none-ear	lowing toolchain applications are missing: arm-none-eabi-addr2line.exe, arm- abi-gcc.exe, arm-none-eabi-nm.exe, arm-none-eabi-objcopy.exe, arm-none-		
Name:	TrueSTUDIO		
Prefix:	arm-none-eabi-		
Location:	C:\Program Files (x86)\Atollic\TrueSTUDIO for STM32 9.3.0\ARMTools\bin		Browse
?	Finish	Canc	el

As seen in Figure 130, some naming problems can occur. In this case, the problem results from a wrong prefix that prevents the toolchain application validation.

Update the toolchain prefix. The prefix must end with a dash (-).

## Figure 131. Specify local toolchain prefix

IDE				$\times$
Add loca	al toolchain			
Name:	TrueSTUDIO			
Prefix:	arm-atollic-eabi-			
Location:	C:\Program Files (x86)\Atollic\TrueSTUDIO for STM32 9.3.0\ARMT	ools\bin		Browse
?	Finis	sh	Canc	el

## 3. Press [Finish].

57

DE Preferences				-	_		$\times$
type filter text		Toolc	hain Manager			⇔ ▼ ↔	<b>▼</b> 00
<ul> <li>Install/Update</li> <li>Remote Development</li> <li>Remote Systems</li> </ul>	^	The Toolchain Manager allows users to configure the workspace toolchain, and manage installed toolchains.					
<ul> <li>Remote systems</li> <li>Run/Debug</li> <li>STM32Cube</li> </ul>	i.	De	Name GNU ARM Embedded (7-2018-q2	State Not installed		Add Lo	ocal
Build Device Configuration Tool End User Agreements		⊳	GNU AKM Embedded (9-2020-q2 GNU Tools for STM32 (7-2018-q2 GNU Tools for STM32 (9-2020-q2	Installed Installed with ST		Remo	ve
File Association Firmware Updater			TrueSTUDIO	Local		Reloa	ad
MPU Serial Serial Wire Viewer						Set De	lauit
larget Status Toolchain Manager		By def	ault let STM32CubeIDE select the toolo	chain to use.	TM32C	uheIDE	
Terminal	~	, vore.	associating and analistation protocolaris r	equites a restart of 3	Re	store De	faults
? ù <b>Ľ</b>				Apply and Close		Cancel	

Figure 132. Local toolchain added

4. Use the [Edit Local...] button to edit local toolchain. The *Edit local toolchain* dialog opens, and it is possible to update *Prefix* and *Location*.

IDE	_		×
Edit loca	Il toolchain		
Name:	TrueSTUDIO		
Prefix:	arm-atollic-eabi-		
Location:	C:\Program Files (x86)\Atollic\TrueSTUDIO for STM32 9.3.0\ARMTools\bin		Browse
?	Finish	Canc	el

Figure 133. Edit local toolchain

5. Update *Prefix* or *Location* and press [Finish] to update local toolchain settings.

57

## 2.11.5 Network error

In case of problem to access the update site, the *Network error...* dialog is displayed.

## Figure 134. Toolchain network error

IDE Netwo	ork error	×
8	Failed to fetch the toochain list, please check your network settings.	
		ОК

Check the network settings. Information on how to configure network proxy settings are described in Section 1.5.3 Preferences - Network proxy settings.

# 3 Debugging

## 3.1 Introduction to debugging

STM32CubeIDE includes a powerful graphical debugger based on the GDB command-line debugger. It also bundles GDB servers for the ST-LINK and SEGGER J-Link JTAG probes.

The GDB server is a program that connects GDB on the PC to a target system. The STM32CubeIDE debug session can autostart a local GDB server or connect to a remote GDB server.

The remote GDB server can be running on the same PC, or on a PC that is accessible via the network and specified with *Host name* or *IP address* and a *Port number*. When connecting to a remote GDB server, this GDB server must be started first before a debug session is started in STM32CubeIDE.

When autostart local debugging is selected, STM32CubeIDE automatically starts and stops the GDB server as required during debugging, thus integrating the GDB server seamlessly.

Note: It is recommended to use compiler optimization level -00 when building a project that must be debugged. Debugging with optimization level -0g may work but higher optimization level is hard to debug because of compiler code optimization.

It is also possible to use the GDB server only to download the application into the target system and run it without starting a debug session. This is performed by creating run configurations, which is described later in this chapter (refer to Section 3.7 Run configurations).

STM32CubeIDE can be used to debug an existing elf file developed with another IDE or toolchain by importing the elf file using the import of STM32 Cortex<sup>®</sup>-M executable. This is described in Section 3.8 Import STM32 Cortex-M executable.



57/

Debug configurations are used to debug an STM32 program. Run configurations are used to flash a new program into the STM32 and start it. The flowchart in Figure 135 presents the order of starting the GDB server, reset the device, load the program, set run time options, exceptions, program counter, breakpoints, Standby/Stop/ Sleep, watchdog, and external loader initialization when starting a debug session. It also displays the differences between debug and run sessions.





## 3.2 Debug configurations

A debug configuration for the project is needed before a debug session can be started. To create the first debug configuration for the project, right click on the project name in the *Project Explorer* view and select [**Debug As**]>[**STM32 Cortex-M C/C++ Application**].

#### Figure 136. Debug as STM32 MCU

workspace\_um4 - NUCLEO-F401RE/Core/Src/main.c - STM32CubelDE

File Edit Source Refactor Navigate Search Project Run Window Help

📑 – 🖪 🛛	🖹   📎 🔻 🗞 🔻 🔜 🔂 👻 💕 🤹	• 🖸 • 🕲 • 🔅	• O • 💁 • 🗋	×   @	🙋 🔗 🔻 🍠 🖻 🔲
🍐 Project Ex	plorer 🛛		\$3 ▽ - □	.c mai	in.c 🖾
> 📭 myLib				153	
	EO-F401RE			154	
	New	>		156	/* MCU Configurati
	Go Into			157	
	Open in New Window			158	/* Reset of all pe
	Show In	Alt. Chift. W/ N		160	// HAL Tnit():
	Show in	Ait+Shirt+W 7		161	// ////////////////////////////////////
	📄 Сору	Ctrl+C		162	/* USER CODE BEGIN
1	Paste	Ctrl+V		163	/* USED CODE END T
	Delete	Delete		165	/* USER CODE END L
•	Course .	Delete		166	/* Configure the s
	Source	,		167	SystemClock_Config
	Move			168	(* USER CODE RESTU
	Rename	F2		170	/* USER CODE BEGIN
	Import			171	/* USER CODE END S
<u>I</u>	import			172	
E	Export			173	/* Initialize all
	Build Project			174	MX_GPIO_Init(); MX_USART2_UART_Ini
	Clean Project			176	/* USER CODE BEGIN
	D D-frach			177	
e	Kerresn	F)		178	<pre>mem=malloc(12);</pre>
	Close Project			179	<pre>mem2=malloc(12); #ifdef OLD</pre>
	Close Unrelated Projects			181	<pre>mem3=malloc(12);</pre>
	Build Configurations	```		182	<pre>mem4=malloc(12);</pre>
				183	<pre>mem5=malloc(12);</pre>
	Build Targets	>		184	<pre>memb=malloc(12); mem7=malloc(12);</pre>
	Index	>		186	#endif
	Show in Remote Systems view			187	/* USER CODE END 2
	Pup Ac	、 、		188	
			1.071402.0	109	
2	pr Debug As	>	TSTM32 Cort	ex-IVI C/0	C++ Application
	Profile As	>	Debug Config	urations	
	Team	>	_	194	/* LISER CODE END
	Compare With	>		194	c = mem[i%12]:
	Restore from Local History			196	mem[i%12] = i;

Another way to create a new debug configuration is to select the project name in the *Project Explorer* view and use the menu [**Run**]>[**Debug As**]>[**STM32 Cortex-M C/C++ Application**].

## Figure 137. Debug as STM32 MCU menu

IDE workspace_um4 - NUCLEO-F401RE/Core/Src/main.c - STM32CubelDE								
File Edit Source	ce Refactor Navigate Search	n Project Run W	/indow Help					
i 📬 🗕 🖬 🔍	Run	ş	• 🔿 • 💁 • 🔪 💩	😕 🛷 🕶 🗾 📴 [				
Project E:	Debug	F11	🖏 🗸 🗖 🗖 🚺 ma	in.c 🛛				
> 😥 myLit	Run History	>	153					
V 💦 NUCL 🜔	Run As	>	154					
> 🎉 Bir	Run Configurations		155	/* MCU Configur				
> 🔑 Co	Debug History	>	157	/* Reset of all				
> 🔑 Dr 🔯	Debug As	> IDE	1 STM32 Cortex-M C/C-	++ Application				
> 👝 De	Debug Configurations		161	// HAC_THITC();				

A third way to create a new debug configuration is to select project name in the *Project Explorer* view and press **[F11**].

All three different ways open the Debug Configuration dialog.

## 3.2.1 Debug configuration

The Debug Configuration dialog contains the following tabs:

- Main
- Debugger
- Startup
- Source
- Common

The *Debugger* and *Startup* tabs must be updated when creating a new debug configuration while the others do not require update.

## 3.2.2 Main tab

The *Main* tab contains the configuration of the C/C++ application to debug. Usually, when creating a debug configuration using the sequence described earlier in this chapter, there is no need to make any change in the *Main* tab. Make sure the correct elf file and project are selected.

## Figure 138. Debug configuration main tab

Edit Configuration		
Edit launch configuration properties		TO.
Name: NUCLEO-F401RE		
🖹 Main 🎋 Debugger 🕨 Startup 🦻 Source 🗉 Common		
Project:		
NUCLEO-F401RE		Browse
C/C++ Application:		
Debug/NUCLEO-F401RE.elf	Search Project	Browse
Build (if required) before launching		
Build Configuration: Select Automatically		~
○ Enable auto build ○ Disable auto	o build	
Use workspace settings <u>Configure Wor</u>	rkspace Settings	
	Revert	Apply
0	ОК	Cancel

Note: It is possible in the Main tab to define if a build must be made before the debug session is started.

## 3.2.3 Debugger tab

The *Debugger* tab configures how to start the GDB server and connect to it. It also defines which GDB server must be used if [**Autostart local GDB server**] is selected.

57		
		IDE [
		Edit

Figure 13	39. Debug	configuration	debugger tab
-----------	-----------	---------------	--------------

Edit Configuration	$ \Box$ $\times$
Edit launch configuration properties	Tor.
	C.
Name: NUCLEO-F401RE	
🖹 Main 🎋 Debugger 🕨 Startup 🦻 Source 🔲 Common	
GDB Connection Settings	
Autostart local GDB server     Host name or IP address	ss localhost
$\bigcirc$ Connect to remote GDB server Port number	61234
Debug probe ST-LINK (ST-LINK GDB server) 🗡	
GDB Server Command Line Options	
Show Command Line	
Interface	
● SWD ○ JTAG	
ST-LINK S/N	✓ Scan
Frequency (kHz): Auto	~
Access port: 0 - Cortex-M4	
Reset behaviour	
type. connect under reset	
Device settings	
Debug in low power modes: Enable	~
Suspend watchdog counters while halted: No configurat	ion 🗸
Serial Wire Viewer (SWV)	RTOS Kernel Awareness
Core Clock (MHz): 16.0	Driver settings
	Port: cortex_m0 ~
Port number: 61235	Port number: 60000
Misc	
Vorify flach download	
venty flash uowilloau	
Enable live expressions	
Enable live expressions     Log to file:     C:\Users\johansse\STM32Cubel	DE\workspace_um8\NUCLEO-F401RE\D Browse
Verify fish download     Enable live expressions     Log to file: C:\Users\johansse\STM32Cubel     External Loader:	DE\workspace_um8\NUCLEO-F401RE\D Browse
Verify flash download     Enable live expressions     Log to file: C:\Users\johansse\STM32Cubel     External Loader:     Shared ST-LINK	DE\workspace_um8\NUCLEO-F401RE\D Browse  Scan Initialize
Verify fiash download      Enable live expressions      Log to file: C:\Users\johansse\STM32Cubel      External Loader:      Shared ST-LINK      Max halt timeout(s): 2	DE\workspace_um8\NUCLEO-F401RE\D Browse Scan Initialize
Verify fiash download         Enable live expressions         Log to file:       C:\Users\johansse\STM32Cubel         External Loader:         Shared ST-LINK         Max halt timeout(s):       2	DE\workspace_um8\NUCLEO-F401RE\D Browse Scan Initialize
Image: Second control of the second	DE\workspace_um8\NUCLEO-F401RE\D Browse  Scan Initialize
Verify hash download     Enable live expressions     Log to file: C:\Users\johansse\STM32Cubel     External Loader:     Shared ST-LINK     Max halt timeout(s): 2	DE\workspace_um8\NUCLEO-F401RE\D Browse
Verify hash download Enable live expressions Log to file: C:\Users\johansse\STM32Cubel External Loader: Shared ST-LINK Max halt timeout(s): 2	DE\workspace_um8\NUCLEO-F401RE\D Browse Scan Initialize Revert Apply
Verify hash download Enable live expressions Log to file: C:\Users\johansse\STM32Cubel External Loader: Shared ST-LINK Max halt timeout(s): 2	DE\workspace_um8\NUCLEO-F401RE\D Browse Scan Initialize Revert Apply

The [**Port number**] edit field contains the default value used by the GDB server selected in field [**Debug probe**]. Field [**Host name or IP address**] must be set when [**Connect to remote GDB server**] is selected. Field [**Debug probe**] selects the probe and GDB server to be used for debugging. When using an ST-LINK debug probe, *ST-LINK GDB server* or *OpenOCD* can be used. When using a SEGGER J-LINK probe, use the *SEGGER J-LINK GDB server*.

Pressing the [Show Command Line] button opens the *GDB Server command line* dialog. The dialog displays how the GDB server is started according to the current [GDB Server Command Line options] settings.

IDE GDB Server command line			$\times$
C:\ST\STM32CubeIDE_1.7.0.21w21\STM32CubeIDE\plugins \com.st.stm32cube.ide.mcu.externaltools.stlink-gdb- server.win32_2.0.0.202105051205\tools\bin\ST-LINK_gdbserver.exe -d -s -cp C:\ST\STM32CubeIDE_1.7.0.21w21\STM32CubeIDE\plugin \com.st.stm32cube.ide.mcu.externaltools.cubeprogrammer.win32_2 1353\tools\bin -m 0 -k	e -p 612 is 2.0.0.20	234 -l 1 210506	^
			~
Copy & Close		Close	

## Figure 140. GDB server command line dialog

Use the [**Copy & Close**] button to copy the current command line settings to the clipboard, for instance to start the GDB server manually in a command line window by pasting the command.

The [**GDB Server Command Line options**] selections are updated as a function of the [**Debug probe**] selected. Detailed information about these settings are available in Section 3.4 Debug using different GDB servers and sub-sections.

## 3.2.4 Startup tab

The *Startup* tab configures how to start a debug session.

### Figure 141. Debug configuration startup tab

DE Edit Configuration				— 🗆 X
Edit launch configuration properties				TO.
Name: NUCLEO-F401RE				
📄 Main 🕸 Debugger ⊳ Startup 🤤 Source 🔲 Comm	on			
Initialization Commands				<u>`</u>
Load Image and Symbols				
File  Fulle	Build See Main tab	Download 🖋 true	Load symbols 🖋 true	Add
				Remove
				Move up
Runtime Ontions				Move down
Start Address				
Default start address				
Set program counter (hex):     Specify vector table (hex):				
Set breakpoint at: main  Exception on divide by zero  Exception on unaligned access				
Halt on exception				
Run Commands				
				< ~
			Revert	Apply
?			ОК	Cancel

The [Initialization Commands] edit field can be updated with any kind of GDB or GDB server monitor commands if there is any special need to send some commands to the GDB server before load commands are sent. For instance, when using ST-LINK GDB server a monitor flash mass\_erase command can be entered here if a Flash memory erase is needed before load.

The [Load Image and Symbols] list box must contain the file(s) to debug. This list is associated with the following command buttons:

- [Add...]: Add new lines for files for download and/or load symbols
- [Edit...]: Edit the selected line
- [Remove]: Remove the selected line from the list
- [Move up]: Move the selected line upwards
- [Move down]: Move selected line downwards

The [**Runtime Options**] section contains checkboxes to set the start address and breakpoint, and enable exception handling and resume.

The start address can be selected as:

problem

- [Default start address]: \$pc is set to the start address found in the last loaded elf file
- [Set program counter (hex)]: \$pc is set to the hex value specified in the edit field
- [Specify vector table (hex)]: \$pc is updated with the value found in memory using specified address + offset of 4. This is similar to how \$pc is set by a reset using vector table in a Cortex<sup>®</sup>-M device

The [Set breakpoint at:] checkbox is enabled by default and the edit field displays main. It means that, by default, a breakpoint is set at main when the program is debugged.

Three exception checkboxes, [Exception on divide by zero], [Exception on unaligned access] and [Halt on exception], are used to make it easier to find problems when debugging an application.

- [Exception on divide by zero] is enabled by default to make it easier to trap a divide-by-zero error when debugging
- [Exception on unaligned access] can be enabled to get exceptions if there are any unaligned accesses
   [Halt on exception] is enabled by default so that program execution halts when an exception error occurs during debugging. If an exception occurs, the *Fault Analyzer* view can be used to find the location of the

Note:

The application software needs to enable **Exception on divide by zero** and **Exception on unaligned access** if they must be issued also when running the application and not only during debugging. The CMSIS Cortex<sup>®</sup>-M header files contain defines to update the SCB Configuration Control Register. For instance, core\_cm4.h contains the SCB->CCR register, and SCB CCR DIV 0 TRP and SCB CCR UNALIGN TRP defines.

When the **[Resume]** selection is enabled, a continue command is issued to GDB after load to start the program. Usually, in this case, the program breaks at main if a breakpoint at main is set. Otherwise, when the **[Resume]** selection is disabled, the program stays at the ENTRY location specified in the linker script, normally the Reset\_Handler function. A step may be needed in this case to display the Reset\_Handler function in the editor.

When a line in the listbox is selected and [Edit...] is pressed, the following dialog appears for selecting if the file must be downloaded and if symbols must be loaded.

DE Add/Edit item		_		×
Some properties of be changed in the "	this item c Main" tab.	an only		
Project:	NUCLEO-	F401RE		
Program path:	Debug\N	UCLEO-	F401RE.	elt
🗹 Download				
Use download offs	et (hex)			
✓ Load symbols				
Use symbol addres	ss (hex)			
	OK		Cancel	

#### Figure 142. Add/Edit item

## 3.3 Manage debug configurations

57/

Each project can have several debug configurations. It is easy to create a copy of an existing debug configuration and update it with some changes. For instance, one configuration may contain Flash memory loading of new programs while another does not load any program.

When opening debug configurations from the menu [**Run**]>[**Debug Configurations...**], the *Debug Configurations* dialog opens. This dialog contains a navigation window on the left side with a toolbar, and the debug configuration on the right side with the tabs and fields described in Section 3.2 Debug configurations.

Figure	143.	Manage	debug	config	gurations
--------	------	--------	-------	--------	-----------

Debug Configurations					$\times$
Create, manage, and run configuration	ons			3	Ś.
© C/C++ Application © C/C++ Attach to Application © C/C++ Postmortem Debugger © C/C++ Remote Application © GDB Hardware Debugging ♣ Launch Group ♥ STM32 Cortex-M C/C++ Application ■ NUCLEO-F401RE	Name:       NUCLEO-F401RE            Main	cally Configure V	Search Project auto build Workspace Settings	Browse	
Filter matched 8 of 8 items			Revert	Apply	
0			Debug	Close	

The [Name] field on top of the right pane can be edited using a name for the debug configuration which reflects the configuration. This name then appears in the navigation window under the [STM32 Cortex-M C/C++ Application] node to the left when pressing [Apply].

The toolbar left of the navigation window contains icons to manage configurations, for instance to duplicate or delete a selected configuration.

Figure 144. Manage debug configurations toolbar





These icons are used for the following purpose, from left to right:

- Create new launch configuration
- New launch of configuration prototype
- Export launch configuration
- Duplicate currently selected launch configuration
- Delete selected launch configuration(s)
- Collapse all expanded launch configurations
- Filter launch configurations

## 3.4 Debug using different GDB servers

STM32CubeIDE includes the following GDB servers:

- ST-LINK GDB server
- OpenOCD GDB server
- SEGGER J-Link GDB server

All three GDB servers support normal debug, live expressions and SWV.

All GDB servers also support RTOS Kernel Awareness debugging for Microsoft<sup>®</sup> Azure<sup>®</sup> RTOS ThreadX and FreeRTOS<sup>™</sup> operating systems using an RTOS proxy. The RTOS proxy is included in STM32CubeIDE.

Different command-line options are used when starting these GDB servers. Therefore the *Debugger* tab in the *Debug Configurations* dialog displays different settings depending on the GDB server selected. This section describes the individual settings for each server.

## 3.4.1 Debug using the ST-LINK GDB server

Usually, when the ST-LINK GDB server is used for debugging, there is no need to update the [**GDB Server Command Line Options**] in the Debugger tab. In some cases, the default configuration must be updated, for instance if SWV is intended to be used or if several STM32 boards are connected to the PC.

#### Figure 145. ST-LINK GDB server debugger tab

Debug Configurations		- 🗆 X			
Create, manage, and run configuratio	ns				
type filter text	Main & Debugger  Startup	Common			
C/C++ Application	GDB Connection Settings				
C/C++ Attach to Application	Autostart local GDB server     Host name	ne or IP address localhost			
<ul> <li>C/C++ Remote Application</li> </ul>	O Connect to remote GDB server Port num	ber 61234			
GDB Hardware Debugging	Debug probe ST-LINK (ST-LINK GDB server)	~			
Launch Group     STM32 Cortex-M C/C++ Application	GDB Server Command Line Options				
• NUCLEO-F401RE	Show Command Line				
NUCLEO-F401RE (OpenOCD)	Interface				
NUCLEO-F401RE (SEGGER)	• SWD JTAG				
	ST-LINK S/N	Scan			
	Frequency (kHz): Auto	~			
	Access port: 0 - Cortex-M4	~			
	Reset behaviour				
	Type: Connect under reset V				
	Device settings				
	Debug in low power modes:	Enable ~			
	Suspend watchdog counters while halted:	No configuration $\checkmark$			
	Serial Wire Viewer (SWV)	RTOS Kernel Awareness Enable RTOS Proxy			
	Core Clock (MHz): 84.0	Driver settings			
	Limit SWO clock	Driver: ThreadX			
	Maximum SWO clock (kHz): auto detect	Port: cortex_m0 ~			
	Port number: 61235	Port number: 60000			
	Misc Verify flash download Verify flash download Interview expressions Log to file: C:\Users\johansse	a\STM32CubeIDE\workspace_um8\NUCLEC Browse			
	Scan Initialize				
	Max halt timeout(s): 2				
Filter matched 10 of 10 items		Revert Apply			
?		Debug Close			

Select [SWD] or [JTAG] in [Interface] to define how the ST-LINK probe must connect with the microcontroller. The SWD interface is usually the preferred choice. It must be selected if SWV is to be used.

When [**ST-LINK S/N**] is enabled, the serial number of the ST-LINK probe to be used must be entered in the edit/list field. The [**Scan**] button can be used to scan and list all detected ST-LINK devices connected to the PC. After a scan, the S/N of these ST-LINK devices are listed in the list box from which the desired ST-LINK can be selected. When [**Use specific ST-LINK S/N**] is enabled, the ST-LINK GDB server is started and connects only to the ST-LINK with the selected S/N.

Note:

The [**Frequency (kHz)**] selection defines the communication speed between the ST-LINK and STM32 device. When [**Auto**] is selected, the maximum speed provided by ST-LINK is used. Reduce the frequency in case of hardware limitations.

The [Access port] selection is used only when debugging a multi-core STM32 device. In such case, the ST-LINK is connected to the device and the ST-LINK GDB server must be informed of the core to debug.

The [Reset behaviour] contains selections for [Type] and [Halt all cores]. The [Halt all cores] selection is only visible for multi-core devices.

The [Type] can be set as follows:

- [Connect under reset] (default): ST-LINK reset line is activated and ST-LINK connects in the SWD or JTAG mode while reset is active. Then the reset line is deactivated.
- [Software system reset]: System reset is activated by software writing in a register. This resets the core and peripherals, and can reset the whole system as the reset pin of the target is asserted by itself.
- [Hardware reset]: ST-LINK reset line is activated and deactivated (pulse on reset line), then ST-LINK connects in the SWD or JTAG mode.
- [Core reset]: Core reset is activated by software writing in a register (not possible on Cortex<sup>®</sup>-M0, Cortex<sup>®</sup>-M0+ and Cortex<sup>®</sup>-M33 cores). This only resets the core, not the peripherals nor the reset pin.
- [None]: For attachment to a running target where the program is downloaded into the device already. There must not be any file program command in the *Startup* tab.

The selected reset behaviour is overridden if the debug configuration includes Flash programming, in which case the ST-LINK GDB server uses the STM32CubeProgrammer (STM32CubeProg) command-line program *STM32\_Programmer\_CLI* to program the Flash memory. This program is always started by the ST-LINK GDB server with mode=UR reset=hwRst so that a device reset is done when loading a new program, disregarding the selection of the [None] option. This ensures that device programming is made correctly.

[Halt all cores] can be used only when debugging multi-core devices. The [Halt all cores] selection is not visible for single core devices.

[Device settings] contains selections for [Debug in low power modes] and [Suspend watchdog counters while halted]. These can be defined as:

- [No configuration]
- [Enable]
- [Disable]

The [Serial Wire Viewer (SWV)] selections can be used only when the [SWD] interface is selected. When [SWV] is enabled, it is required to configure the [Clock Settings]. The [Core Clock] must be set to the device speed. More information about SWV configuration is available in Section 4.2.1 SWV debug configuration.

The [**RTOS Kernel Awareness**] selections are used to to enable RTOS-kernel-aware debugging with the ThreadX and FreeRTOS<sup>TM</sup> operating systems. When RTOS-kernel-aware debugging is enabled and a debug session is started, all threads are listed in the *Debug* view. By selecting a thread in the *Debug* view the current line executed by the thread is displayed in the editor. More information about RTOS-kernel-aware debugging is available in Section 6.3.

The [Misc] selections contains:

- [Verify flash download]
- [Enable live expressions] (To be able to use the *Live Expressions* view during debugging, the live expression mechanism must be enabled during startup. It is enabled by default.)
- [Log to file] (Enable in case of debugging problems. It starts the ST-LINK GDB server with a higher log level and saves the log into a file.)
- [External Loader] (Enable if loading must be made to non-internal STM32 Flash memory). A [Scan] button is available to access STM32CubeProgrammer external Flash loader files.
  - When [External Loader] is enabled, there is also an [Initialize] selection. When it is enabled, the Init() function in STM32CubeProgrammer is called after reset. It can be used to configure the device for external memory access. Usually, initialization must be done by the debugged application.
- [Shared ST-LINK] (Shared ST-LINK must be enabled if other programs must be able to connect to the same ST-LINK during a debug session.). Refer to Section 3.6.2 Shared ST-LINK for details.
   A detailed description of the ST-LINK GDB server is available in the ST-LINK GDB server manual ([ST-07]), which is available from the *Information Center*.

57

*Note: STM32\_Programmer\_CLI* is used by the ST-LINK GDB server to program the STM32 or external Flash memory. In this case, such external Flash memory programming is automatically done using the external loader.

## 3.4.2 Debug using OpenOCD and ST-LINK

When OpenOCD is used, the [**GDB Server Command Line Options**] in the *Debugger* tab contains a generator options toggle field, which alternates between [**Show generator options**...] and [**Hide generator options**...]. When the field is set to [**Hide generator options**...], the dialog displays additional [**GDB Server Command Line Options**] as shown in Figure 146.

## Figure 146. OpenOCD debugger tab

Debug Configurations		— D ×			
Create, manage, and run configuratio	ns				
] 🖻 ଭ 🗎 🗶 🖻 🍸 🔻					
type filter text	Name: NUCLEO-F401RE (OpenOCD)				
C/C++ Application	Main Sebugger Startup Source Commo	n			
© C/C++ Attach to Application	GDB Connection Settings				
C/C++ Postmortem Debugger	Autostart local GDB server Host name or IP a	ddress localhost			
C/C++ Remote Application	• Connect to remote GDB server Port number	3333			
GDB Hardware Debugging	Debug probe ST-LINK (OpenOCD)				
Launch Group	GDB Server Command Line Options				
SIM32 Cortex-M C/C++ Application INFICE CO. E401PE	Chan Canada Lina				
NUCLEO F401RE (OpenOCD)					
NUCLEO-F401RE (SEGGER)	OpenOCD Setup				
	OpenOCD Command:				
	"\${stm32cubeide_openocd_path}\openocd.exe"	Browse			
	OpenOCD Options :	\$			
	Configuration Script				
	Configuration Script				
	$\odot$ Automated Generation $\bigcirc$ User Defined	Hide generator options			
	Script File: \${ProjDirPath}\NUCLEO-F401RE (OpenOCD).cfg Browse Reload				
	Generator Options				
	Mode Setup				
	Connection Setup Reset Mode: Connect under reset				
	Interface: Swd ~	C Enable debug in law newer modes			
	Frequency: 8 MHz V				
		Stop watchdog counters when halt			
	Serial Wire Viewer (SWV)	RTOS Kernel Awareness			
	Enable	Enable RTOS Proxy			
	Core Clock (MHz): 8	Driver settings			
	Limit SWO clock	Driver: ThreadX			
	Maximum SWO dock (kHz): auto detect	Port: cortex m0			
	Port number: 3344	Cortex_mo			
		Port number: 60000			
	Misc				
	□ Enable live expressions				
	L				
ilter matched 10 of 10 items		Revert Apply			
	1				
(?)		Debug Close			

The [OpenOCD Command] edit field contains the <code>openocd.exe</code> file to be used when debugging. The [Browse] button can be used to select another version of OpenOCD.

The [**OpenOCD Options**] edit field can be used to add additional command-line parameters to be used when starting OpenOCD.

The [Configuration Script] selections can be [Automated Generation] or [User Defined]. When [Automated Generation] is selected, an <code>openocd.cfg</code> file is created automatically based on the selections made in the *Debugger* tab. When [User Defined] is selected, the file must be specified in the [Script File] edit field.

The [Interface]selection [Swd] or [Jtag] selects how the ST-LINK probe must connect with the microcontroller. [Swd] is usually the preferred choice.

The **Frequency** selection configures the communication speed between the ST-LINK and STM32 device. The [**Reset Mode**] selection contains:

- [Connect under reset] (default): ST-LINK reset line is activated and ST-LINK connects in the SWD or JTAG mode while reset is active. Then the reset line is deactivated.
- [Hardware reset]: ST-LINK reset line is activated and deactivated (pulse on reset line), then ST-LINK connects in the SWD or JTAG mode.
- [Software system reset]: System reset is activated by software writing in a register. This is resetting the core and peripherals, and can reset the whole system as the reset pin of the target is asserted by itself.
- [Core reset]: Core reset is activated by software writing in a register (not possible on Cortex<sup>®</sup>-M0, Cortex<sup>®</sup>-M0+ and Cortex<sup>®</sup>-M33 cores). This is only resetting the core, not the peripherals nor the reset pin.
- **[None]**: For attachment to a running target where the program is downloaded into the device already. There must not be any file program command in the *Startup* tab.

[Enable debug in low power modes] enables debug also with the STM32 device in low-power mode. [Stop watchdog counters when halt] stops the watchdog when the debug session halts the STM32 device. Otherwise, a watchdog interrupt may be triggered.

The [Serial Wire Viewer (SWV)] selections can be used only when the [SWD] interface is selected. When [SWV] is enabled, it is required to configure the [Clock Settings]. The [Core Clock] must be set to the device speed. More information about SWV configuration is available in Section 4.2.1 SWV debug configuration.

The [**RTOS Kernel Awareness**] selections are used to to enable RTOS-kernel-aware debugging with the ThreadX and FreeRTOS<sup>TM</sup> operating systems. When RTOS-kernel-aware debugging is enabled and a debug session is started, all threads are listed in the *Debug* view. By selecting a thread in the *Debug* view the current line executed by the thread is displayed in the editor. More information about RTOS-kernel-aware debugging is available in Section 6.3.

[Enable live expressions] must be enabled if the *Live Expressions* view is meant to be used during debugging. [Shared ST-LINK] must be enabled if other programs have to connect to the same ST-LINK during a debug session. Refer to Section 3.6.2 Shared ST-LINK for details.

## 3.4.3 Debug using SEGGER J-Link

When [**SEGGER J-LINK**] is selected in the *Debugger* tab, the [**GDB Server Command Line Options**] corresponds to SEGGER J-Link GDB server.

#### Figure 147. SEGGER debugger tab

Debug Configurations						×
Create, manage, and run configuration	15					Ú.
		CED				
type filter text	Main & Debugger Start		mman			
<ul> <li>type filter text</li> <li>C/C++ Application</li> <li>C/C++ Attach to Application</li> <li>C/C++ Postmortem Debugger</li> <li>C/C++ Remote Application</li> <li>GDB Hardware Debugging</li> <li>Launch Group</li> <li>STM32 Cortex-M C/C++ Application</li> <li>NUCLEO-F401RE</li> <li>NUCLEO-F401RE (OpenOCD)</li> <li>NUCLEO-F401RE (SEGGER)</li> </ul>	<ul> <li>Main * Debugger ► Startt</li> <li>GDB Connection Settings</li> <li>Autostart local GDB serve</li> <li>Connect to remote GDB s</li> <li>Debug probe SEGGER J-LINH</li> <li>GDB Server Command Line</li> <li>Interface</li> <li>SWD ) JTAG Initial Sp</li> <li>Device STM32F401RE</li> <li>Reset strategy Type 0: Nor</li> <li>JTAG Scan Chain</li> <li>Auto Manual Positic</li> <li>Serial Wire Viewer (SWV)</li> <li>Enable</li> <li>Core Clock (MHz):</li> <li>Limit SWO dock</li> <li>Maximum SWO dock (kHz)</li> <li>Port number:</li> <li>Misc</li> <li>Use J-Link script file</li> <li>✓ Enable live expressions</li> <li>✓ Verify flash download</li> </ul>	up Source Cor r Host name of rever Port number C V V Dptions mal V 8.0 auto detect 2332	nmon r IP address localhost 2331 z Use specific J-Link S/N 0 RTOS Kernel Awareness Driver settings Driver Settings Driver ThreadX Port: cortex_m0 Port number: 60000	s syy O J-Link Search Project	Brows	÷
Filter matched 10 of 10 items				Revert	Apply	
۲				Debug	Close	

The [Interface] selection [SWD] or [JTAG] selects how the SEGGER J-Link probe must connect with the microcontroller. The [SWD] interface is usually the preferred choice; it is required if SWV is used.

The [Initial Speed] selection configures the communication speed used between SEGGER J-Link and the STM32 device.

When [Use specific J-Link S/N] is enabled, enter the S/N of the J-Link to be used when debugging in the edit/list field. When [Use specific J-Link S/N] is enabled, the SEGGER J-Link GDB server is started and connects only to the J-Link with the selected S/N.

The **Device** edit field is used if it contains an entry. This field can be used if there is a problem to start the SEGGER J-Link GDB server with the default device name used in STM32CubeIDE. In such case, enter the device name used by the SEGGER GDB server in the edit field.

The [Reset strategy] selection contains:

- [Type 0: Normal] Default.
- [None] Intended to be used for attaching to the running target. In this case, the program must already be downloaded into the device. There must not be any file program command in the *Startup* tab.

The [JTAG Scan Chain] selections can be used only when the [JTAG] interface is selected.

The [Serial Wire Viewer (SWV)] selections can be used only when the [SWD] interface is selected. When [SWV] is enabled, it is required to configure the [Clock Settings]. The [Core Clock] must be set to the device speed. More information about SWV configuration is available in Section 4.2.1 SWV debug configuration.

The [**RTOS Kernel Awareness**] selections are used to to enable RTOS-kernel-aware debugging with the ThreadX and FreeRTOS<sup>TM</sup> operating systems. When RTOS-kernel-aware debugging is enabled and a debug session is started, all threads are listed in the *Debug* view. By selecting a thread in the *Debug* view the current line executed by the thread is displayed in the editor. More information about RTOS-kernel-aware debugging is available in Section 6.3.

The [Misc] selections contains:

- [Use J-Link script file]
- [Enable live expressions]

To be able to use the *Live Expressions* view during debug, the live expression mechanism must be enabled during startup.

- [Verify flash download]
- [Select RTOS variant] list box can be used if [Thread-aware RTOS support] is used with [FreeRTOS] and [embOS].

When [Thread-aware RTOS support] is used, update the *Startup* tab: disable [Resume] and [in Run Commands], add thread 2 and continue. This forces a thread context switch before the continue command is sent.

Note: A detailed description of SEGGER J-Link GDB server is available in the SEGGER J-Link manual, which can be accessed from the "Information Center".

## 3.5 Start and stop debugging

When a debug configuration is created for the project with the preferred JTAG probe, it is ready for debugging. In the following sections, the ST-LINK GDB server is used. However, the way to debug the STM32 project is quite independent of the choice among ST-LINK GDB server, OpenOCD or SEGGER J-Link.

Perform the following steps to prepare for debug:

- 1. Determine whether the board supports the JTAG debug, SWD debug, or both. SWD-mode debug is usually the preferred choice.
- Connect the JTAG cable between the JTAG probe and the target board.
   When using STMicroelectronics STM32 Nucleo and Discovery boards, the ST-LINK is usually integrated on the board. Also, most STMicroelectronics STM32 Evaluation boards contain an embedded ST-LINK.
- 3. Connect the USB cable between the PC and the JTAG probe.
- 4. Make sure that the target board has a proper power supply attached.

Once the steps above are performed, a debug session can be started.

## 3.5.1 Start debugging

Open the *Debug Configurations* dialog with a right click on the project name in the *Project Explorer* view and select [**Debug As**]>[**Debug Configurations...**].

It is also possible to open the dialog using the menu [Run]>[Debug Configurations...].

This opens the Debug Configurations dialog.

Note: It is possible to select the project in the "Project Explorer" view and press [**F11**] to restart a debug session after it has been closed.

## Figure 148. Debug configurations

Debug Configurations	- D X
Create, manage, and run configurations	
[] @ 🖗 🗎 🗶 🖃 🐳 -	Name: NUCLEO-F401RE
type filter text	Main 🕸 Debugger 🕨 Startup 🧤 Source 🔲 Common
type filter text © C/C++ Application © C/C++ Attach to Application © C/C++ Postmortem Debugger © C/C++ Remote Application © GDB Hardware Debugging (■ Launch Group ► Launch Group (Deprecated) ♥ IS STM32 Cortex-M C/C++ Application INI NUCLEO-F401RE INI NUCLEO-F401RE (SEGGER) INI NUCLEO-F401RE (SEGGER)	Main Stebugger       Startup is Source       Common         GDB Connection Settings
	☐ Max halt timeout(s): 2
Filter matched 11 of 11 items	Revert Apply
?	Debug Close

Select in the left pane the debug configuration to use. Press the [**Debug**] button to start a debug session if all debug configurations have been made. The project is built if file updates are made, but the building depends on the debug configuration.

STM32CubeIDE launches the debugger and the following dialog is opened.

#### Figure 149. Confirm perspective switch



It is recommended to enable [**Remember my decision**] and press [**Switch**]. It opens the *Debug* perspective, which provides a number of views and windows suitable for debugging.

## 3.5.2 Debug perspective and views

The Debug perspective contains menus, toolbars and views frequently used during debugging.

#### Figure 150. Debug perspective

workspace_um1 - NUCLEO-F401RE/Core/Src/main.c - STM32CubeIDE					- 🗆 ×
File Edit Source Refactor Navigate Search Project Run Window Help					
(1) * 回転 B13(本本) = N3(5)(2)(4)(4)(2)(3)(4) * (3)(4)(4)(3)(4)(4)(4)(4)(4)(4)(4)(4)(4)(4)(4)(4)(4)					Quick Access
体 Debug 💈 😓 Project Explorer 💦 🕹 👾 🐃 🗠 C	🕲 mainc 🔅 🗄 startup stm32/401retos 📧 Reset, Handler() at startup stm32/401retos:106.0x8000908	👓 Variables 🛸 Breakpo	oints & Express	sions 38 Registers 4	Live Expressions = SFRs 🖘 👘 🔍
Working - Schneide Laplace TMM (STML2 Contex M (CC+ Application) EMALICE (CF+ MR) (MR) (STML2 Contex M (CC+ Application) - # United H) (man) 1 (cont (B) Science-oil Emangement) - Emand and annual Science (B) Science-oil Emangement - Emand and annual Science (B) (Science-oil Emandement) - Emand and annual Science (B) (SCIENCE-OIL Emandement) - Crist(FML2COLOGE, 1) (SCIENCE-OIL Emandement) - SCIENCE (SF-UNK COE) server)	Brainice 3 Braining and/90 heres. Effect Headleng at sump simil/00 heres.100.000/000     1       116 - ** effects     116 - ** effects       116 - ** effects     117 - */       117 - */     */       118 - ** effects     118 - **       118 - ** effects     119 - **       118 - ** effects     119 - **       118 - ** effects     119 - **       119 - **     **       119 - **     **       111 - **     **       112 - **     **       118 - **     **       119 - **     **       119 - **     **       119 - **     **       119 - **     **       119 - **     **       119 - **     **       110 - **     **       111 - **     **       112 - **     **       113 - **     **       114 - **     **       115 - **     **       111 - **     **       112 - **     **       113 - **     **       114 - **     **       115 - **     **       115 - **     **       111 - **     **       112 - **     **       113 - **     **       114 - **     **       115 - *	Image: second	Address V	alue	
		<ul> <li>MOTG_FS_DEVICE</li> <li>MOTG_FS_HOST</li> <li>MOTG_FS_HOST</li> <li>MOTG_FS_PWRCE</li> <li>MPWR</li> <li>MPCC</li> <li>MRTC</li> <li>MSDIO</li> </ul>	AL CLK		
	135 _/* Configure the system clock */	> MISYSCEG			
	<pre>11 by yearnusker_consp() 11 by yearnusker_consp() 12 / USER CORE END Systint */ 13 // USER CORE END Systint */ 14 // USER CORE END Systint */ 15 // USER CORE END Systint */ 16 // USER CORE END Systint */ 17 // USER CORE END Systint */ 18 // USER CORE END Systint */ 19 // US</pre>	• 511/264 • 511/120 • 611/261 • 611/263 • 611/264 • 701/264 • 701/264			
	133 while (3) 134 ( / user coor tho Will t */ v W Concor 13 Redense O Tannatates Wentying Download verified successfully				、 、 、 、 、 、 、 、 、 、 、 、 、 、
c	T «				>
					V Province and Pro

The most important views opened by default in the Debug perspective are:

- The *Debug* view, which displays the program under debug. It also lists threads and offers the possibility to navigate in the thread by selecting a line in threads.
- The *Editor* view, which displays the program file. It is possible to set break points and follow program execution in the file. It is also possible to hoover the cursor over a variable to display its current value. The features available during file edition are available also during debug, such as opening the declaration of a function and others.



- The Variables view, which displays local variables automatically with their current value when the program is not running.
- The *Breakpoints* view, which displays current breakpoints. It is possible to disable and enable breakpoints in the list. The *Breakpoints* view also contains a toolbar, which, for instance, enables to remove breakpoints, and skip breakpoints with one click on the [Skip All Breakpoints] icon.
- The *Expressions* view, which is used to add and view expressions. An expression may be a single global variable, structure, or an expression calculating some variables. The values are only updated when the program is stopped. It is possible to select a global variable in the *Editor* and drag it over to the *Expressions* view instead of entering the variable name.
- The *Registers* view, which displays the debugged device current values. The values are only updated when the program is stopped.
- The *Live Expressions* view, which displays expression values sampled and updated regularly during program execution. The view allows the creation of mathematical expressions that are evaluated automatically, such as (Index\*4+Offset). The *Live Expressions* view update requires that live expressions are enabled in the debug configuration. Refer to Section 3.6.1 Live Expressions view for details.
- The SFRs view, which displays the Special Function Registers in the debugged device. Refer to Section 5 Special Function Registers (SFRs) for details.
- The Console view, which displays different console outputs. By default, the console output from the GDB server log is displayed. It is possible to change the console log by pressing the [Display Selected Console] icon to the right of the Console view.

Other views are also useful during debug, among which:

• The *Debugger Console* view, which can be used if there is a need to manually enter GDB commands. The easiest way to open the *Debugger Console* view is to use the [Quick Access] field and enter Debugger in this field. It lists choices containing the *Debugger Console* view. Select it to open the view. GDB can be entered in the *Debugger Console* view.

For instance, to display 16 words of memory from address  $0 \times 800 \ 0000$ , enter the GDB command  $\times /16 \ 0 \times 8000000$ .

x /16 0x80000	000			
0x8000000:	0x20018000	0x080008b1	0x080007e9	0x080007f7
0x8000010:	0x080007fd	0x08000803	0x08000809	0x00000000
0x8000020:	0x00000000	0x00000000	0x00000000	0x0800080f
0x8000030:	0x0800081d	0x00000000	0x0800082b	0x08000839

- The Memory and Memory Browser views, which can be used to display and update memory data.
- The Disassembly view, which is used to view and step in the assembly code.
- The SWV views. Refer to Section 4 Debug with Serial Wire Viewer tracing (SWV) for details.
- The Fault Analyzer view. Refer to Section 7 Fault Analyzer for details.

## 3.5.3 Main controls for debugging

The [Run] menu in the Debug perspective contains a number of execution control functions.
#### Figure 151. [Run] menu

IDE WO	orkspace_um1 - NUCLEO-F401RE	/Core/Src/main.c
File Ec	dit Source Refactor Navigate Sea	rch Project Run
-	퉉 Terminate And Relaunch	
	i⇒ Instruction Stepping Mode	
₩ Dei	-ඖ Move to Line (C/C++)	
✓ IDE	Resume at Line (C/C++)	
~	Resume	F8
	III Suspend	
	Terminate	Ctrl+F2
	N Disconnect	
	Resume Without Signal	5
	🕹 Reset	
	-3- Step Into	F5
	Step Over	Fo
	- Step Return	F7
	Run to Line	Ctri+R
	Step Inte Selection	
		Ctri+F5
	No. Debug	F11
	Debug History	>
	🎋 Debug As	>
	Debug Configurations	
	Breakpoint Types	>
	<ul> <li>Toggle Breakpoint</li> </ul>	Ctrl+Shift+B
	<ul> <li>Toggle Line Breakpoint</li> </ul>	
	🞾 Toggle Watchpoint	
	<ul> <li>Toggle Method Breakpoint</li> </ul>	
	🔌 Skip All Breakpoints	Ctrl+Alt+B
	💥 Remove All Breakpoints	5

Alternatively, the *Debug* perspective toolbar has the following main debug control icons.

#### Figure 152. Debug toolbar



These icons are used for the following purpose, from left to right:

- Reset the device and restart the debug session
- Skip all breakpoints (Ctrl+Alt+B)
- Terminate and relaunch
- Resume (F8)
- Suspend

57/

- Terminate (Ctrl+F2)
- Disconnect
- Step into (F5)
- Step over (**F6**)
- Step return (**F7**)
- Instruction stepping mode (assembler stepping)

Press [**Terminate and relaunch**] to terminate the current debug session, build a new program if the source code is modified, and relaunch the debug session.

When pressing **[Instruction stepping mode]**, the *Disassembly* view is opened and further stepping uses assembler instruction stepping level. Press **[Instruction stepping mode]** again to toggle back to C/C++ level stepping.

#### 3.5.4 Run, start and stop a program

Use the toolbar icons as follows to run, step, or stop the program:

- Run the program with the [Resume] toolbar icon ([F8])
- Step into a function with the [Step into] toolbar icon ([F5])
- Step over a function with the [Step over] toolbar icon ([F6])
- Step until return from a function with the [Step return] toolbar icon ([F7])
- Abort running program with the [Suspend] toolbar icon

#### 3.5.5 Set breakpoints

It is common during a debug session to set breakpoints and let the code execute until it reaches a breakpoint.

**UM2609** Start and stop debugging

#### 3.5.5.1 Standard breakpoint

**[\** 

A standard code breakpoint at a source code line can easily be inserted by double-clicking in the left editor margin, or by right-clicking in the left margin of the C/C++ source code editor. A context menu is proposed in the latter case.

15		
	Toggle Breakpoint	Ctrl+Shift+B
	Add Breakpoint	Ctrl+Double Click
	Add Dynamic Printf	
	Disable Breakpoint	Shift+Double Click
	Breakpoint Properties	Ctrl+Double Click
	Breakpoint Types	>
	Go to Annotation	Ctrl+1
	Add Bookmark	
2	Add Task	
$\checkmark$	Show Quick Diff	Ctrl+Shift+Q
$\checkmark$	Show Line Numbers	
	Folding	>
	Preferences	

#### Figure 153. Debug breakpoint

Select the [Toggle Breakpoint] menu command to set or remove a breakpoint at the corresponding source code line.

#### 3.5.5.2 Conditional breakpoint

When setting a standard breakpoint at a source code line, the program breaks each time it reaches this line. If that is not the desired behaviour, a condition can be set on the breakpoint that regulates if the program should actually break or not on that breakpoint.

Update breakpoint properties with a right-click on the breakpoint icon visible left of the editor on a line with breakpoint set. The [**Breakpoint Properties**] can also be opened from the *Breakpoints* view.

<u>~1</u>		
2	Toggle Breakpoint	Ctrl+Shift+B
	Add Breakpoint	Ctrl+Double Click
	Add Dynamic Printf	
	Disable Breakpoint	Shift+Double Click
	Breakpoint Properties	Ctrl+Double Click
	Breakpoint Types	>
	Go to Annotation	Ctrl+1
	Add Bookmark	
	Add Task	
$\checkmark$	Show Quick Diff	Ctrl+Shift+Q
~	Show Line Numbers	
	Folding	>
	Preferences	

#### Figure 154. Breakpoint properties

Select [Breakpoint Properties...]. The following window opens. In the example illustrated below, *i*>20 is entered as a condition.

	-,			_	
	Common			<> ▼ □	\$ <b>•</b> •
Common Actions Filter	Class: Type: File: Line number: Enabled Condition: Ignore count:	C/C++ Line Breakpoint Regular C:\Users\johansse\STM32CubeIDE\workspa 156 i>20 0	ace_um1\NUCLEO-F401RE\Cc	ore\Src\ma	iin.c
2			Apply and Close	Cancel	

# With the condition above set, the program breaks each time the line is executed, then GDB tests the condition and restarts execution if the variable $\pm$ is not greater than 20. It takes some time for GDB to evaluate the

condition.

The conditions are written in C-style. It is therefore possible to write expressions such as i%2==0" to set more complex conditions.

#### 3.5.6 Attach to running target

It is possible to connect STM32CubeIDE and a debugger via JTAG/SWD to the embedded target without performing a reset. This approach is useful when trying to resolve problems that occur at rare occasions. Finding the root cause of the problem in case of a CPU crash is further simplified by learning how to use the *Fault Analyzer* view (refer to Section 7 Fault Analyzer).

Before trying this approach, consider whether halting the application in the wrong state could potentially harm the hardware (for instance in the case of a motor controller application). This is because when GDB connects to the target, the CPU is halted. This behaviour cannot be modified.

The following three or four steps are needed to update the debug configuration and to attach to running target:

- 1. Modify the debug configuration to attach to the running target
- 2. Connect the debug probe to the embedded target
- 3. Start a debug session using the modified debug configuration
- 4. Optionally, analyze the CPU fault condition with the Fault Analyzer tool (refer to Fault Analyzer)

#### Step 1: Modify the debug configuration

The default generated debug configurations in STM32CubeIDE contains settings to reset the device and download new program, and sets a breakpoint at main. This is not of any use when connecting to a running system which may, or may not, have crashed.

In order to create a modified debug configuration, perform these steps:

- 1. Open the Debug Configurations dialog.
- 2. In the left frame of the *Debug Configurations* dialog, select the debug configuration associated to the project to debug and make a copy of this by right-clicking it and selecting [**Duplicate**].
- 3. Give the duplicate debug configuration a name.
- 4. Update the Debugger tab in Debug Configurations:
  - When using ST-LINK GDB server and OpenOCD, select [None] as [Reset behaviour].
  - When using SEGGER J-Link GDB server, select [None] as [Reset strategy].

# 5. Change needed/recommended in the *Startup* tab of *Debug Configurations* for both ST-LINK GDB server and SEGGER J-Link GDB server:

- Disable file [Download] in [Load Image and Symbols].
- Disable [Set program counter at (hex)].
- Disable [Set breakpoint at].
- [Exception on divide by zero] and [Exception on unaligned access] can be disabled or enabled.
- Disable [Resume].
   If the [Resume] is enabled, the debugger stops the target during connection and, after a short period of time, sends a continue command.

Debug Configurations		– – ×
,		
	Name: NUCLEO-F401RE Debug (ST-LINK attach)	
type filter text	■ Main 参 Debugger ► Startup  Source □ Common	
$\mathbb{C}$ C/C++ Application	Initialization Commands	
$\bigcirc$ C/C++ Postmortem Debugger		^
C/C++ Remote Application		$\sim$
GDB Hardware Debugging	Load Image and Symbols	
<ul> <li>Launch Group (Deprecated)</li> </ul>	File Build Download Load sym	Add
✓ ■ STM32 Cortex-M C/C++ Application	Debug/NUCLEO-F401RE.elf [NUCLEO-F See Main 🔒 false 🛛 📌 true	Edit
NUCLEO-F401RE Debug (OpenOCD)		Demonstra
••• NUCLEO-F40TRE Debug (SEGGER)		Remove
NUCLEO-F401RE Debug (ST-LINK)		Move up
		Move down
	Runtime Options	
	Set program counter at (hex):	
	Set breakpoint at: main	
	Exception on divide by zero	
	Exception on unaligned access	
	Halt on exception	
	Run Commands	
		×
Filter matched 12 of 12 items	Revert	Apply
The matched 12 of 12 items		, , , ,
(?)	Debug	Close

#### Figure 156. Startup tab attach

#### Step 2: Connect ST-LINK or SEGGER J-Link to the embedded target

Connect first ST-LINK or the SEGGER J-link to the computer. Then connect it to the embedded target. No reset is issued.

57



#### Step 3: Start a debug session using the modified debug configuration

#### Important:

Do not launch the debug session using the wrong debug configuration, which may reprogram and reset the target. Use [Run]>[Debug Configurations...], select the modified debug configuration in the left frame, and click [Debug]. This is the safest way to launch a debug session with full control of the debug configuration applied and prevents from a potential reset.

The debugger is now connected to the embedded target, which is automatically halted. At this point, different status registers and variables can be investigated in the application. If the CPU has crashed, the *Fault Analyzer* can be used to get a better understanding of the root causes.

#### 3.5.7 Restart or terminate debugging

This section presents various ways to restart and stop a debug session.

#### 3.5.7.1 Restart

During debugging, it is sometimes needed to restart the program to examine more carefully problems observed during debug. In such case, restart the program using the [**Reset the chip and restart debug session**] toolbar button or [**Run**]>[**Restart**] menu command. This resets the device, and starts the program if [**Resume**] is enabled in the debug configuration.

Note: To make restart work, the interrupt vector must be configured and used with the hardware reset. This is usually the case for STM32 programs located in the Flash memory. However, if the program is located elsewhere such as in RAM, some manual handling may be needed to make the program start from the expected Reset\_Handler.

#### 3.5.7.2 Restart configurations

It is possible to create restart configurations defining how the reset and restart of a debug session must be performed. Click on the arrow to the right of the [**Reset the chip and restart debug session**] toolbar icon.

#### Figure 157. Reset the chip toolbar

లి	•	ÒQ	🖷				М	₽	R	.R
oject	Re	set t	he cł	nip ar	nd re	start	deb	ug si	essio	n

#### This expands the menu with the [Restart Configurations...] selection.

#### Figure 158. Restart configurations selection



When [**Restart Configurations...**] is selected, the restart configurations dialog opens.

IDE Restart configurations			×
Reset and restart configurations			
Name	Reset		
Reset Type:	Reset		$\sim$
Additi	onal commands:		
		Apply	~
?	ОК	Cance	:1

Figure 159. Restart configurations dialog

The dialog contains a left and right pane:

- The left pane is used to select and create new restart configuration, duplicate an existing restart configuration, and delete the selected restart configuration. The default restart configurations cannot be deleted.
- The right pane is used to set [Name] and select the [Type] of reset to be used for the selected configuration. It is also possible to add additional commands to be used with the reset.

Press [Apply] to save a setting.

Figure 160 shows a setting where a new restart configuration is created, which contains an additional command to set pc to 0x8000ca0.

		_				
Restart configurations				—		×
Reset and restart configura	ations					
	Name	Reset_and_set_	pc			
Reset_and_set_pc	Туре:	Reset				~
Reset	Additio	onal commands:				
	set Sp	oc=0x8000ca0			Apply	~
?			Oł	(	Cance	ł

Figure 160. Restart configurations dialog with additional command

When several reset configurations are defined, they appear in the toolbar dropdown menu in order of use. Select the desired one to perform a reset.

Figure 161. Select restart configuration

٠ 🕹	• 🔍   🖡 🕩 💷 🔳 🙌 🌫 🤆						
	Reset						
	Reset_and_set_pc						
	Restart Configurations						
тиана и соле, од совренаса , отсакроih							

#### 3.5.7.3 Terminate

The most common way to stop a debug session is by clicking the [**Terminate**] toolbar button. It is also possible to stop the debug session with the [**Run**]>[**Terminate**] menu. When the debug session is stopped, STM32CubeIDE switches automatically to the C/C++ perspective.

#### 3.5.7.4 Terminate and relaunch

Use the [**Terminate And Relaunch**] toolbar button if changes in the source code have been made during the debug session. Menu command [**Run**]>[**Terminate And Relaunch**] can also be used for this purpose. This stops the debug session, rebuild the program, and relaunches a debug session with the new program loaded.

#### **3.6** Debug features

#### 3.6.1 *Live Expressions* view

The *Live Expressions* view in STM32CubeIDE works very much like the *Expression* view with the exceptions that all the expressions are sampled live during debug execution. The sampling speed is determined by the number of expressions being sampled. An increased number of expressions being sampled results in a slower sample rate. The view displays many different types of global variables. The view also allows users to create mathematical expressions that are evaluated automatically, such as (i \* 4 + offset).

(x)= Variables 💁 Breakpoints 🙀 Expressi	ons 🚻 Registers 🚱 Live Exp	pressions 🔀 📾 SFRs 🛛 🕷	K 🐝 🕴 🗖 🗖
Expression	Туре	Value	View Menu
(x)= i	uint32_t	2	
(×)= offset	uint32_t	8	
(x)= i*4+ offset	unsigned int	16	
🐈 Add new expression			

#### Figure 162. Live Expressions

The view can parse complicated data types and display complex data types like C-language structures. Only one format of numbers is used at the same time. To change this format, use the dropdown menu.

#### Figure 163. Live expressions number format

(x)= Variables 💁 Breakpoints 🚭 Expression	ns 1010 Registers	ଜୁନ	Live	Expressions	83	SFRs		*	8	
Expression	Туре			Default			Numbe	r For	mat	$\rightarrow$
(×)= i	uint32_t			Hex						
(×)= offset	uint32_t		•	Decimal						
(×)= i*4+ offset	unsigned int			Octal						
🐈 Add new expression				Rinand						
			_	binary						

The variable values can be changed on the fly in the *Live Expressions* view while the program is running. Select the variable and change its value. It requires that only single variable name is used in the expression, and that no calculation involved.

*Note:* To be able to use the Live Expressions view during debug, the live expression mechanism must be enabled during startup.

#### 3.6.2 Shared ST-LINK

In the *Debugger* tab in *Debug Configurations* for ST-LINK GDB server and OpenOCD, a selection enables shared ST-LINK. When shared ST-LINK is enabled, the communication to ST-LINK goes via the ST-LINK server. The ST-LINK server makes it possible for several programs to access the same ST-LINK when shared ST-LINK is enabled.

STM32CubeProgrammer (STM32CubeProg) also contains a configuration for shared ST-LINK. This means that when shared ST-LINK is enabled in the debug configuration in STM32CubeIDE, it is possible to debug a program and let STM32CubeProgrammer access and read the device Flash memory and RAM at the same time.

Enabling the shared mode causes STM32CubeIDE to launch the ST-LINK server, if it is not already running, with the default port 7184 for listening to the TCP connection. This default port is not editable from STM32CubeIDE.

#### 3.6.3 Debug multiple boards

Debugging with multiple boards is possible using two ST-LINK or SEGGER J-Link probes at the same time. Connected to two different microcontrollers, both probes are connected to one PC on different USB ports. In this section, let us suppose that two different boards/microcontrollers are used: HW\_A and HW\_B.

It is possible to run one instance of STM32CubeIDE containing one project for HW\_A and one project for HW\_B. The default port to be used is:

- 61234 for ST-LINK GDB server
- 3333 for OpenOCD
- 2331 SEGGER J-Link

This is presented in the *Debugger* tab in the *Debug Configurations* dialog. The port number must be changed for one of the projects to use another port, such as port 61244.

The debug configuration can use GDB connection selection [Autostart local GDB server]. Note that when debugging multiple boards, two or more debug probes are connected to the PC; the correct serial number must be selected for each debug configuration.

When the debug configurations has been configured for both projects so that each board is associated to a specific probe, it is time to test and debug each board individually first. When it is confirmed that this is working, the debug of both targets at the same time can be started as follow:

- 1. Start to debug HW\_A.
- 2. The perspective switches automatically to the *Debug* perspective in STM32CubeIDE when a debug session for HW\_A is started.
- 3. Switch to the C/C++ perspective.
- 4. Select the project for HW\_B and start debugging it. The *Debug* perspective opens again.
- 5. There are two application stacks/nodes in the *Debug* view, one for each project. When changing the selected node in the *Debug* view, the related editor, variable view and others are updated to present information associated to the selected project.

It is also possible to start the GDB servers manually: select [**Connect to remote GDB server**] in the debug configuration. In such case, make sure that the GDB servers are started with parameters defining the individual ports and serial numbers to be used, and that the corresponding port numbers are used in the *Debug Configurations* dialog for each project.

Below is an example using SEGGER J-Link GDB server connecting to SEGGER J-Link, with port=2341 and S/N=123456789:

>JLinkGDBServerCL.exe -port 2341 -if SWD -select usb=123456789

Information on command-line parameters to be used when starting the GDB servers manually are provided in the GDB server manuals available from the *Information Center*.

#### 3.6.4 STM32H7 multicore debugging

Information about how to use STM32H7 multicore devices in STM32CubeIDE is available in [ST-09].

#### 3.6.5 STM32MP1 debugging

Information about how to use STM32MP1 devices in STM32CubeIDE is available in [ST-08]. Users are advised to keep updated with the documentation evolution of the STM32MP1 Series at www.st.com/en/ microcontrollers-microprocessors/stm32mp1-series.

#### 3.6.6 STM32L5 debugging

Information about how to use STM32L5 devices with TrustZone® in STM32CubeIDE is available in [ST-10].

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

#### 3.7 Run configurations

It is possible to create run configurations to download applications and reset the target without launching a full debug session. The *Run Configurations* dialog is similar to the *Debug Configurations* dialog, however disabled widgets in the lower part of the *Startup* tab are not performed. When running a run configuration, the specified program is flashed but, after program counter is set, the program execution is started in target and the "run" session in STM32CubeIDE is closed.

To create a run configuration for the project, right-click on the project name in the *Project Explorer* view and select **[Run As]**>**[STM32 Cortex-M C/C++ Application]**.

Another way to create a run configuration is to select the project name in the *Project Explorer* view and use the menu [Run]>[Run As]>[STM32 Cortex-M C/C++ Application].

	rigare for run comigure				
Run Configurations					– 🗆 X
Create, manage, and run configurations					
C C/C++ Application C C/C++ Application Launch Group Launch Group (Deprecated) STM32 Cortex-A Linux Deployment STM32 Cortex-M C/C++ Application NUCLEO-F401RE (OpenOCD) NUCLEO-F401RE (SEGGER)	Name:       NUCLEO-F401RE            Main          Main          Surce        Comm          Initialization Commands        Comm          Initialization Commands        Load Image and Symbols         File       Debug\NUCLEO-F401RE.elf [NUCLEO-F401RE]         Batt Address       Start Address         © Default start address       © Set program counter (hex):         © Specify vector table (hex):       © Set breakpoint at:         main       Exception on divide by zero         Exception on unaligned access       Halt on exception         Run Commands       Resume	ion Build See Main tab	Download ✓ true	Load symbols	Add Edit Remove Move up Move down
Filter matched 9 of 9 items				Revert	Apply
?				Run	Close

#### Figure 164. Run configurations startup tab



Use menu [File]>[Import...] to open the Import dialog.

#### Figure 165. Cortex<sup>®</sup>-M executable import dialog

Import – 🗌	×
Select	Ľ
Imports an externally built STM32 Cortex-M executable into a new project and configures the debugger.	
Select an import wizard:	
type filter text	
> 🗁 General	
✓ ➢ C/C++	
C/C++ Executable	
C/C++ Project Settings     C/C++ Project	
Existing Code as Makefile Project	
STM32 Cortex-M Executable	
> 🖻 Install	
> 🗁 Remote Systems	
> 🗁 Run/Debug	
> 🗁 Team	
? < Back Next > Finish Cance	el



#### Select [STM32 Cortex-M Executable] and press [Next >].

STM32 Cortex-M Executable				_	
Select executable to debug and c	onfigure a matchin	g STM32 device.			
Executable: C:\dev\STM32F401_	Nucleo_ExtBuilt.elf				Browse
STM32 device					
MCU:					Select
CPU:					~
Core:					~
?	< Back	Next >	Finish		Cancel

#### Figure 166. STM32 Cortex<sup>®</sup>-M executable dialog

Use the [**Browse...**] button and select the elf file to import. When the elf file is selected, the STM32 device must be selected manually so that STM32CubeIDE can be used for debugging. Press [**Select...**] to open the *MCU/MPU Selector* dialog.

#### Figure 167. STM32 Cortex<sup>®</sup>-M executable MCU/MPU selection

DE MCU/MPU Selector		_		$\times$
Please select your STM32 target device				
401			 	
MCU/MPU				^
STM32F401RDTx				
STM32F401RETx				
STM32F401VBHx				- 10
STM32F401VBTx				
STM32F401VCHx				
STM32F401VCTx				$\sim$
?	C	)K	Cancel	

Select the microcontroller or microprocessor to be used. The search field can be used to find the device. Press [OK] once the device is selected.

As a result, the CPU and core are presented in the dialog.

#### Figure 168. STM32 Cortex<sup>®</sup>-M CPU and core

IDE STM32 Cortex-M Executable	
Select executable to debug and configure a matching STM32 device.	
Executable: C:\dev\STM32F401_Nucleo_ExtBuilt.elf	Browse
STM32 device	
MCU: STM32F401RETx	Select
CPU: Cortex-M4 (0)	$\sim$
Core: 0	$\sim$
<pre>? &lt; Back Next &gt; Finish</pre>	Cancel

Press [Finish] and the debug configuration dialog automatically opens.

	lobug comgutation it			
Edit Configuration				×
Edit configuration and launch.				TO.
				2
Name: STM32F401_Nucleo_ExtBuilt.elf				
■ Main 参 Debugger ► Startup ⁵ Source □ 0	Common			
Project:				
STM32F401_Nucleo_ExtBuilt.elf			Browse	e
C/C++ Application:				
STM32F401_Nucleo_ExtBuilt.elf		Search Project	Browse	e
Build (if required) before launching				
Build Configuration: Use Active				$\sim$
○ Enable auto build	<ul> <li>Disable au</li> </ul>	to build		
OUse workspace settings	Configure Wo	orkspace Settings		
		Revert	Apply	y
0		Debug	Close	)

#### Figure 169. Cortex<sup>®</sup>-M debug configuration for imported project

The debug configuration can then be set up in similar way as with any other STM32CubeIDE project. Once the configuration is completed, press [**Debug**] to start a debug session.

The imported project is displayed in the *Project Explorer* view.

Figure 170. Project explorer view with imported project
workspace_um7 - Device Configuration Tool - S
File Edit Source Refactor Navigate Search Prc
📑 🕶 🔚 🐚   🥸 🕶 🗞 🕶 📾 📷 🖝 😂 🕶 😅 🕶
Project Explorer ⊠ 🖻 🕏 🍸 🕴 🗖 🗖
✓ I™ NUCLEO-F401RE
> 🔊 Includes
> 🙆 Core
> 😕 Drivers
NUCLEO-F401RE.ioc
STM32F401RETX_FLASH.ld
STM32F401RETX_RAM.Id
<ul> <li>STM32F401_Nucleo_ExtBuilt.elf</li> </ul>
> 🔊 Includes
STM32F401_Nucleo_ExtBuilt.elf



## 4 Debug with Serial Wire Viewer tracing (SWV)

#### 4.1 Introduction to SWV and ITM

This section provides information on how to use Serial Wire Viewer tracing (SWV) in STM32CubeIDE.

System analysis and real-time tracing in STM32 requires a number of interaction technologies: Serial Wire Viewer (SWV), Serial Wire Debug (SWD), Instrumentation Trace Macrocell (ITM) and Serial Wire Output (SWO). These technologies are part of the Arm<sup>®</sup> CoreSight<sup>™</sup> debugger technology. They are explained below.

Serial Wire Debug (SWD) is a debug port similar to JTAG. It provides the same debug capabilities (run, stop on breakpoints, single-step) but with fewer pins. It replaces the JTAG connector with a 2-pin interface (one clock pin and one bi-directional data pin). The SWD port alone does not allow real-time tracing.

The Serial Wire Output (SWO) pin can be used in combination with SWD. It is used by the processor to emit real-time trace data, thus extending the two SWD pins with a third pin. The combination of the two SWD pins and SWO pin enables Serial Wire Viewer (SWV) real-time tracing in compatible Arm<sup>®</sup> processors.

Beware that, SWO being just one pin, it is easy to set a configuration that produces more data than the SWO is able to send.

The Serial Wire Viewer (SWV) is a real-time trace technology that uses the Serial Wire Debug (SWD) port and the Serial Wire Output (SWO) pin. The Serial Wire Viewer provides advanced system analysis and real-time tracing without the need to halt the processor to extract the debug information.

Serial Wire Viewer (SWD) provides the following types of target information:

- Event notification on data reading and writing
- Event notification on exception entry and exit
- Event counters
- Timestamp and CPU cycle information, which can be used for program statistical profiling

The Instrumentation Trace Macrocell (ITM) enables applications to write arbitrary data to the SWO pin, which can be interpreted and visualized in the debugger. For example, ITM can be used to redirect printf() output to a SWV console view in the debugger. The standard is to use port 0 for this purpose.

The ITM port has 32 channels. Writing different types of data to different ITM channels allows the debugger to interpret or visualize the data on various channels differently.

Writing a byte to the ITM port takes only one write cycle, thus taking almost no execution time from the application logic.

Based on SWV, and ITM trace data, STM32CubeIDE can provide advanced debugger capabilities with special *SWV* views.

Note: Arm<sup>®</sup> does not include SWV/ITM in Cortex<sup>®</sup>-M0 or Cortex<sup>®</sup>-M0+ cores. Therefore, STM32 devices based on these cores, such as STM32L053 microcontrollers, do not support SWV/ITM.

#### 4.2 SWV debugging

To debug and use the Serial Wire Viewer (SWV) in STM32CubeIDE, the JTAG probe and the GDB server must support SWV. The board must also support SWD, and the SWO pin needs to be available and connected to the JTAG probe.

The following sections describe the process to create a debug configuration, SWV settings configuration, and how to use SWV tracing in a debug session.

#### 4.2.1 SWV debug configuration

#### Step 1: Open the Debug Configurations dialog

Use for instance menu [Run]>[Debug Configurations...] and select the STM32 Cortex<sup>®</sup>-M debug configuration to update.

#### Step 2: Select the SWD interface

Select the [SWD] interface in the Debug Configurations dialog.

#### Step 3: Enable SWV

Enable [SWV] in the Debug Configurations dialog.

#### Step 4: Enter the core clock frequency

Enter the [**Core Clock**] frequency in the *Debug Configurations* dialog. This must correspond to the value set by the application program to be executed.

Usually, the core clock setting is stored in the SystemCoreClock variable when using projects imported from STM32 firmware examples or created with STM32CubeMX. One method to inspect the core clock value is to start a debug session and add the SystemCoreClock variable to the *Expressions* view. Make sure that the system core clock is configured by the application before reading the value.

If the SystemCoreClock is not updated, change the program and add a call to the function SystemCoreClockUpdate(). Rebuild the program, restart debugging and inspect the SystemCoreClock value again.

IDE					×
🕵 Expressions 🛛		約 🍕 🕞 🖕 💥 🎇	📬	🖻 8	
Expression (x)= SystemCoreClock	Type uint32_t	Value 84000000			

#### Figure 171. SWV core clock

#### Step 5: Enter the SWO clock frequency

The [Serial Wire Viewer (SWV)] selections in the *Debug Configurations* dialog can be used only when the [SWD] interface is selected. When [SWV] is enabled, it is required to configure the [Clock Settings]. The [Core Clock] must be set to the device speed. The SWO clock is automatically set to the highest possible speed depending on debug probe used and core clock. However, if the debugged hardware does not allow too-high SWO clock speed, it is possible to enable [Limit SWO clock] and enter the maximum SWO clock speed in kHz. The SWV [Port number] must be set to the port to be used for SWV data communication. The SWV port cannot be set equal to the GDB connection [Port number].

#### Figure 172. SWV debug configuration

Serial Wire Viewer (SWV) Serial Wire Viewer (SWV)		
Core Clock (MHz):	84.0	
Limit SWO clock		
Maximum SWO clock (kHz):	auto detect	
Port number:	61235	

#### Step 6: Save the configuration

Press [Apply]in the Debug Configurations dialog to save the configuration.

#### Step 7: Start a debug session

Press [Debug] to start a debug session. Make sure that the probe and board are connected.

#### Step 8: Possibly suspend the target

[Suspend] the target if it has not stopped at a breakpoint.

#### Step 9: Open a SWV view

Open one of the *SWV* views. For first-time users, it is recommended to open the *SWV Trace log* view because it gives a good overview of incoming SWV packets and how well the tracing is working. Select the [**Window**]>[**Show View**]>[**SWV**]>[**SWV Trace log** ] menu command to open the *SWV Trace log* view.

Project Run Window	Help		
New Window	- 🍠 🚱 🔻 🖓 💌 🏷 🗢 🔿	- 🖬 🚺	
Editor	> nain c 🐰 🕄 startup stm32t	401 rety s Res	et Handler() at startun stm32f401rety s
Appearance			
Show View	> 🗉 SWV	>	🔲 SWV Trace log
Perspective	> • Breakpoints	Alt+Shift+Q, B	SWV Exception Trace log
Navigation	> 🗟 Build Analyzer		SWV Exception Timeline Graph
Preferences	Console	Alt+Shift+Q, C	SWV Data Trace
pended : Signal : 0	T. The Debug		SWV Data Trace Timeline Graph
f6	1 Debugger Console		SWV Statistical Profiling
n32f401retx.s:113 (	1 S Frror Log	Alt+Shift+O_I	SWV Statistical Froming
CubeIDE/plugins/cc	1 Q Executables		
	1 e Expressions		
	1 🝯 Fault Analyzer		peripherals */
	1 👾 Live Expressions		
	14 🗴 Memory		
	1 🔋 Memory Browser		
	1′ ➡ Modules		
	1 🗄 Outline	Alt+Shift+Q, O	
	1' 🔝 Problems	Alt+Shift+Q, X	
	<sup>1</sup> S Progress		
	Project Explorer		
	1 Registers		
	1 Signals		
	1 Static Stack Analyzer		
1	1 Strace Control		
	(x)= Variables	Alt+Shift+Q, V	
	1 Other	Alt+Shift+Q, Q	

#### Figure 173. SWV show view

Step 10: View the trace log

The SWV Trace log view is now visible.



🗳 Consol	le 🔝 Problem	ns 🛛 Executable	es 🔳 SWV Trac	e Log 🛛		×●× ₩ = =
Index	Туре	Data	Cycles	Time(s)	Extra info	

Overflow packets: 0

#### 4.2.2 SWV settings configuration

#### Step 1: Open the Serial Wire Viewer settings

Click on the [Configure Trace] toolbar button in the SWV Trace Log view to open the Serial Wire Viewer settings dialog.

#### Figure 175. SWV [Configure Trace] toolbar button



#### Note: The [Configure Trace] toolbar button is available in all SWV views.

#### Step 2: Configure the trace data

Configure the data to be traced in the *Serial Wire Viewer settings* dialog. For this example [**PC Sampling**] and [**Timestamps**] are enabled.

#### Figure 176. SWV settings dialog

DE Serial Wire Viewer settings	s for NUCLEO-F401RE Debug (ST-LINK)		×
Clock Settings Core Clock: 84 MHz Clock Prescaler: 42 SWO Clock: 2000.0 kHz	Trace Events         z       CPI: Cycles per instruction         SLEEP: Sleep cycles       LSL         FOLD: Folded instructions       EXE	C: Exception overhead J: Load store unit cycles ETRC: Trace Exceptions	PC Sampling ✓ Enable Resolution: 16384 ∨ Cycles/sample Timestamps ✓ Enable Prescaler: 1 ∨
Data Trace Comparator 0 Data Trace Var/Addr: 0x0 Access: Read/Write Size: Word Generate: Data Value ITM Stimulus Ports	Comparator 1 Enable Var/Addr: 0x0 Access: Read/Write Size: Word Generate: Data Value	Comparator 2 Enable Var/Addr: 0x0 Access: Read/Write Size: Word Generate: Data Value	Comparator 3 Enable Var/Addr: 0x0 Access: Read/Write Size: Word Generate: Data Value
Privileged only ports: Por	t 3124 Port 2316 Port 158	Port 70	OK Cancel

The SWV settings dialog has the following configurations:

- [Clock Settings]: These fields are disabled and only present the values used and configured in the *Debug Configurations* for the debug session. If these values need to be changed, close the debug session and open the *Debug Configurations* to modify them.
- [Trace Events]: The following events can be traced.
  - [CPI]: Cycles per instruction. For each cycle beyond the first one that an instruction uses, an internal counter is increased with one. The counter (DWT CPI count) can count up to 256 and is then set to 0. Each time that happens, one of these packets are sent. This is one aspect of the processors performance and used to calculate instructions per seconds. The lower the value, the better the performance.
  - [SLEEP]: Sleep cycles. The number of cycles the CPU is in sleep mode. Counted in DWT Sleep count register. Each time the CPU has been in sleep mode for 256 cycles, one of these packets is sent. This is used when debugging for power consumption or waiting for external devices.
  - [FOLD]: Folded instructions. A counter for how many instructions are folded (removed). Every 256 instruction folded (taken zero cycles) will receive one of these events. Counted in DWT Fold count register.

Branch folding is a technique where, on the prediction of most branches, the branch instruction is completely removed from the instruction stream presented to the execution pipeline. Branch folding can significantly improve the performance of branches, taking the CPI for branches below 1.

- **[EXC]**: Exception overhead. The DWT Exception count register keeps track of the number of CPU cycles spent in exception overhead. This includes stack operations and returns but not the time spent processing the exception code. When the timer overflows, one of these events is sent. Used to calculate the actuel exception handling cost to the program.
- [LSU]: Load Store Unit Cycles. The DWT LSU count register counts the total number of cycles the
  processor is processing an LSU operation beyond the first cycle. When the timer overflows, one of
  these events is sent.
  - With this measurement, it is possible to track the amount of time spent in memory operations.
- [EXETRC]: Trace Exceptions. Whenever an exception occurs, exception entry, exception exit and
  exception return events are sent. These events can be monitored in the SWV Exception Trace Log
  view. From this view, it is possible to jump to the exception handler code for that exception.
- [PC Sampling]: Enabling this starts sampling the Program Counter at some cycle interval. Since the SWO pin has a limited bandwidth, it is not advised to sample to fast. Experiment with the [Resolution] (cycles/ sample setting) to be able to sample often enough. The results from the sampling are used, among other things, for the SWV Statistical Profiling view.
- [Timestamps]: Must be enabled to know when an event occurred. The [Prescaler] should only be changed as a last effort to reduce overflow packets.
- [Data Trace]: It is possible to trace up to four different C variable symbols, or fixed numeric areas of the memory. To do that, enable one comparator and enter the name of the variable or the memory-address to trace. The value of the traced variables can be displayed both in the Data Trace and Data Trace Timeline Graph views.
- [ITM Stimulus Ports]: There are 32 ITM ports available, which can be used by the application. For instance, the CMSIS function ITM\_SendChar can be used to send characters to port 0 refer to Section 4.3.5 SWV ITM Data Console and printf redirection). The packets from the ITM ports are displayed in the SWV ITM Data Console view.

Note: It is recommended to limit the amount of data traced. Most STM32 microcontrollers read and write data faster than the maximum SWO pin throughput. Too many trace data result in data overflow, lost packets and possibly corrupt data. For optimum performance, trace only data necessary to the task at hand.

Overflow while running SWV is an indication that SWV is configured to trace more data than the SWO pin is able to process. In such a case, decrease the amount of data traced.

Enable [Timestamps] to use any of the timeline views in STM32CubeIDE. The default [Prescaler] is 1. Keep this value, unless problems occur related to SWV packet overflow.

Three examples are provided below for illustrating SWV trace configuration:

- **Example 1:** To trace the value of a global variable, enable [**Comparator**] and enter the name of the variable or the memory address to be traced.
  - The value of the traced variable is displayed both in the Data Trace and Data Trace Timeline Graph views.
- **Example 2:** To profile program execution, enable [**PC sampling**]. In the beginning, a high value for the [**Cycles/sample**] is recommended.
- The result from the PC sampling is then displayed in the SWV Statistical Profilingview.
- Example 3: To trace the exceptions occurring during program execution, enable [Trace Event EXETRC: Trace Exceptions].

Information about the exceptions is then displayed in the SWV Exception Trace Log view.

#### Step 3: Save the SWV configuration

Click on the [**OK**] button to save the SWV configuration. The configuration is saved together with other debug configurations and remains effective until changed.

#### 4.2.3 SWV tracing

#### Step 1: Start SWV trace recoding

Press the [Start/Stop Trace] toolbar button in one of the SWV views to send the SWV settings to the target board and start the SWV trace recoding. This toolbar button is available in all SWV views. The board does not send any SWV packet until it is properly configured. The SWV configuration must be resent if the configuration registers on the target board are reset. Actual tracing does not start until the target starts to execute.

#### Figure 177. SWV [Start/Stop Trace] toolbar button

Note:



The tracing cannot be configured while the target is running. Pause the debugging before attempting to send a new configuration to the board. Each new or updated configuration must be sent to the board to take effect. The configuration is sent to the board when the [**Start/Stop Trace**] button is pressed.

#### Step 2: Start the target

Press the [Resume] toolbar button on top of the Debug perspective to start the target.

#### Step 3: SWV Trace Log view

SWV packets are displayed in the SWV Trace Log view.

#### Figure 178. SWV Trace Log PC sampling

IDE					$-\Box$	×
SWV Tr	ace Log 🛛				ו• × 🖬	- 8
Index	Туре	Data	Cycles	Time(s)	Extra info	^
10362	PC Sample	0x8000508	169777034	2.021155 s		
10363	PC Sample	0x8000516	169793417	2.021350 s		
10364	PC Sample	0x8000528	169809800	2.021545 s		
10365	PC Sample	0x8000500	169826183	2.021740 s		
10366	PC Sample	0x8000510	169842566	2.021935 s		
10367	PC Sample	0x80004f2	169858949	2.022130 s		
10368	PC Sample	0x8000504	169875332	2.022325 s		
10369	PC Sample	0x8000516	169891715	2.022520 s		
10370	PC Sample	0x8000528	169908098	2.022715 s		
Overflow	v packets: 0					~

#### Step 4: Clear collected SWV data

When the target is not running, the collected SWV data can be cleared by pressing the [**Remove all collected SWV data**] toolbar button. This toolbar button is available in all SWV views.

#### Figure 179. [Remove all collected SWV data] toolbar button



#### 4.3 SWV views

The SWV views that display SWV traces data are:

- SWV Trace Log: Lists all incoming SWV packets in a spreadsheet. Useful as a first diagnostic for the trace quality.
- SWV Exception Trace Log: The view has two tabs, one is similar to the SWV Trace Log view and the other tab displays statistical information about exception events.
- SWV Data Trace: Tracks up to four different symbols or areas in the memory.
- SWV Data Trace Timeline Graph: A graphical display that shows the distribution of variable values over time.
- SWV ITM Data Console: Prints readable text output from the target application. Typically this is done via printf() with output redirected to ITM channel 0.
- *SWV Statistical Profiling*: Displays statistics based on the Program Counter (PC) sampling. Shows the amount of execution time spent within various functions.

#### Figure 180. SWV views selectable from the menu

SWV Trace log	
---------------	--

- SWV Exception Trace log
- SWV Data Trace
- 🗠 SWV Data Trace Timeline Graph
- SWV ITM Data Console
- SWV Statistical Profiling
- *Note:* More than one SWV view may be open at the same time for the simultaneous tracking of various events. The SWV views toolbars contain these usual control icons.

#### Figure 181. SVW views common toolbar



These icons are used for the following purpose, from left to right:

- Configure trace
- Start/Stop trace
- Remove all collected SWV data
- Scroll lock
- Minimize
- Maximize



The SWV graph views toolbars contain these extra control icons.

#### Figure 182. SVW graph views extra toolbar



These icons are used for the following purpose, from left to right:

- Save graph as image
- Switch between seconds and cycle scale
- Adjust the Y-axis to best fit
- Zoom in
- Zoom out

#### 4.3.1 SWV Trace Log

The *SWV Trace Log* view lists all incoming SWV packets in a spreadsheet. The data in this view can be copied to other applications in CSV format by selecting the rows to copy and type **Ctrl+C**. The copied data can be pasted into another application with the **Ctrl+V** command.

#### Figure 183. SWV Trace Log PC sampling and exceptions

IDE						_		×
SWV Tr	ace Log 🛛						* 🔵 🗙	<u>a</u> :
Index	Туре	Data	Cycles	Time(s)	Extra info			^
25012	PC Sample	0x80004f6	258481871	3.077165 s				
25013	PC Sample	0x8000508	258498254	3.077360 s				
25014	PC Sample	0x8000518	258514637	3.077555 s				
25015	Exception entry	SYSTICK (EXC 15)	258522309	3.077647 s				
25016	Exception exit	SYSTICK (EXC 15)	258522367	3.077647 s				
25017	Exception return	N/A (EXC 0)	258522374	3.077647 s				
25018	PC Sample	0x80004fc	258531017	3.077750 s				
25019	PC Sample	0x800050e	258547400	3.077945 s				
25020	PC Sample	0x800051e	258563783	3.078140 s				
25021	DC Sampla	022000/16/	258580166	2 078225 c				~
Overflow	packets: 0							

The column information in the SWV Trace Log view is described in Table 6.

#### Table 6. SWV Trace Log columns details

Name	Description
Index	The packet ID. Shared with the other SWV packets.
Туре	The type of packet (example PC sample, data PC value (comp 1), exceptions, overflow).
Data	The packet data information.
Cycles	The timestamp of the packet in cycles.
Time(s)	The timestamp of the packet in seconds.
Extra info	Optional extra packet information.

#### 4.3.2 SWV Exception Trace Log

The SWV Exception Trace Log view is composed of two tabs.

#### Data tab

The first tab is similar to the *SWV Trace Log* view, but is restricted to exception events. It also provides additional information about the type of event. The data can be copied and pasted into other applications. Each row is linked to the code for the corresponding exception handler. Double-click on the event to open the corresponding interrupt hander source code in the *Editor* view.

Note:

Enable [**Trace Event EXETRC: Trace Exceptions**] in the Serial Wire Viewer settings dialog to trace exceptions during program execution. Enable [**Timestamps**] to log cycle and time for each interrupt packet.

IDE							— 🗆	$\times$
SWV E	xception Trace Log						× • ×	<b>a</b>
Data S	Statistics							
Index	Туре	Name	Peripheral	Function	Cycles	Time(s)	Extra info	^
17629	Exception exit	SYSTICK (EXC 15)		SysTick_Handler()	58204401	692.909536 ms		
17630	Exception return	N/A (EXC 0)			58205926	692.927690 ms	Timestamp delayed. Packet delayed.	
17636	Exception entry	SYSTICK (EXC 15)		SysTick_Handler()	58288335	693.908750 ms		
17637	Exception exit	SYSTICK (EXC 15)		SysTick_Handler()	58288393	693.909440 ms		
17638	Exception return	N/A (EXC 0)			58288400	693.909524 ms		
17644	Exception entry	SYSTICK (EXC 15)		SysTick_Handler()	58372327	694.908655 ms		
17645	Exception exit	SYSTICK (EXC 15)		SysTick_Handler()	58372385	694.909345 ms		
17646	Exception return	N/A (EXC 0)			58372392	694.909429 ms		
								¥

#### Figure 184. SWV Exception Trace Log – Data tab

The column information in the SWV Exception Trace Log – Data tab is described in Table 7.

#### Table 7. SWV Exception Trace Log – Data columns details

Name	Description
Index	The exception packet ID. Shared with the other SWV packets.
Туре	Each exception generates three packets: Exception entry, Exception exit and then an Exception return packet.
Name	The name of the exception. Also the exception or interrupt number.
Peripheral	The peripheral for the exception.
Function	The name of the interrupt handler function for this interrupt. Updated when debug is paused. Is cached during the whole debug session. By double clicking the function, the editor will open that function in the source code.
Cycles	The timestamp for the exception in cycles.
Time(s)	The timestamp for the exception in seconds.
Extra info	Optional extra information about that packet.

#### Statistics tab

The second tab displays statistical information about exception events. This information may be of great value when optimizing the code. Hypertext links to exception handler source code in the editor is included.

IDE													- 0	×
SWV Exception T	■ SWV Exception Trace Log ≅								% 🖲 X	<b>8</b> - D				
Data Statistics														
Exception	Handler	% of	Number of	% of excepti	% of debug time	Total runtime	Avg runtime	Fastest	Slowest	First	First (s)	Latest	Latest (s)	
SYSTICK (EXC 15)	SysTick_Handler()	100.0000%	2172	100.0000%	0.0690%	40309	57	57	58	71567	851.988095 µs	58372327	694.908655 ms	
Total for all			2172		0.0690%	40309	18							
Overflow packets:	Overflow packets: 0													

#### Figure 185. SWV Exception Trace Log – Statistics tab

The column information in the SWV Exception Trace Log – Statistics tab is described in Table 8.

Name	Description
Exception	The name of the exception provided by the manufacturer. Also the exception or interrupt number.
Handler	The name of the interrupt handler for this interrupt. Updated when debug is paused. Is cached during the whole debug session.
	By double clicking the handler, the editor will open that function in the source code.
% of	This exception type's share, in percentage, of all exceptions.
Number of	The total number of entry packets received by SWV of this exception type.
% of exception time	How big part of the execution time for all exceptions that this exception type have.
% of debug time	How big part of the total execution time for this debug session that this exception type have. All the timers are restarted when the Empty SWV-Data button is pressed.
Total runtime	The total execution time in cycles for this exception type.
Avg runtime	The average execution time in cycles for this exception type.
Fastest	The execution time in cycles for the fastest exception of this exception type.
Slowest	The execution time in cycles for the slowest exception of this exception type.
First	The first encounter of an entry event for this exception type in cycles.
First(s)	The first encounter of an entry event for this exception type in seconds.
Latest	The latest encounter of an entry event for this exception type in cycles.
Latest(s)	The latest encounter of an entry event for this exception type in seconds.

#### Table 8. SWV Exception Trace Log – Statistics columns details

#### 4.3.3 SWV Data Trace

The *SWV Data Trace* view tracks up to four different symbols or areas in the memory. For example, global variables can be referenced by name. The data can be traced on Read, Write and Read/Write. Enable [**Data Trace**] in *Serial Wire Viewer settings*. In Figure 186, two global variables pos1 and pos2 in the program are traced on [**Write**] access.

#### Figure 186. SWV Data Trace configuration

DE Serial Wire Viewer settings for	or NUCLEO-F401RE Debug (ST-LINK)					×
Clock Settings Core Clock: 84 MHz Clock Prescaler: 42 SWO Clock: 2000.0 kHz	Trace Events CPI: Cycles per instruction EXC SLEEP: Sleep cycles LSL FOLD: Folded instructions EXE	PC Sampling ✓ Enable Resolution: 16384 ∨ Cycles/samp Timestamps ✓ Enable Prescaler: 1				
Data Trace						
Comparator 0 🗹 Enable	Comparator 1	Comparator 2		Comparato	r 3	
Var/Addr: pos1	Var/Addr: pos2	Var/Addr: 0x0		Var/Addr:	0x0	
Access: Write $\vee$	Access: Read	Access: Read/Write	$\sim$	Access:	Read/Write	$\sim$
Size: Word	Size: Word	Size: Word	$\sim$	Size:	Word	$\sim$
Generate: Data Value + PC $$	Generate: Data Value + PC ${\scriptstyle\checkmark}$	Generate: Data Value	$\sim$	Generate:	Data Value	$\sim$
ITM Stimulus Ports Enable port: 31	]24 2316 J124 Port 2316 Port 158	5 15	]8 7[[		0	
				ОК	Car	ncel

When running the program in debugger with SWV trace enabled the SWV Data Trace view displays this information when [Comparator 0] with pos1 data is selected in the [Watch] list.

#### Figure 187. SWV Data Trace

IDE							$\times$
SWV Data	Trace 🛛				* 🔵	K 🗟 🚮 🔨	7
Watch							
Comp	Name	Value					^
0	pos1	10					
1	pos2	0					$\checkmark$
History (pos	1)						
Access	Value	PC	Cycles	Time			^
WRITE	8	0x8000578	642414276	7.647789 s			
WRITE	1	0x8000578	645655051	7.686370 s			
WRITE	2	0x8000578	649164268	7.728146 s			
WRITE	3	0x8000578	652673485	7.769922 s			
WRITE	4	0x8000578	656182631	7.811698 s			
WRITE	5	0x8000578	659691850	7.853474 s			
WRITE	6	0x8000578	663004479	7.892910 s			
WRITE	7	0x8000578	666513696	7.934687 s			
WRITE	9	0x8000578	673532061	8.018239 s			
WRITE	10	0x8000578	677041280	8.060015 s			$\checkmark$

The column information in the SWV Data Trace described in Table 9.

#### Table 9. SWV Data Trace columns details

Name	Description
Access	Read or Write access type.
Value	The value of data read or written.
PC	The PC location where read or write access occurs.
Cycles	The timestamp for the packet in cycles.
Time(s)	The timestamp for the packet in seconds.

#### 4.3.4 SWV Data Trace Timeline Graph

The SWV Data Trace Timeline Graph view contains a graphical display that shows the distribution of variable values over time. It applies to the variables or memory areas in the SWV Data Trace. The following is displayed when using the timeline graph displaying global variables pos1 and pos2 counting up and down.

#### Figure 188. SWV Data Trace Timeline Graph



The SWV Data Trace Timeline Graph has the following features:

- The graph can be saved as a JPEG image file by clicking on the camera toolbar button.
- The graph shows the time in seconds by default but can be changed to cycles by clicking on the clock toolbar button.
- Y-axis can be adjusted to best fit by clicking on the y-axis toolbar button.
- Zoom in and out by clicking on the [+] and [-] toolbar buttons.
- The zoom range is limited while debug is running. Zoom details are available when debug is paused.

#### 4.3.5 SWV ITM Data Console and printf redirection

The SWV ITM Data Console prints readable text output from the target application. Typically, this is done via printf() with output redirected to ITM channel 0. Other ITM channels can get their own console views.

To use the SWV ITM Data Console view, first enable one or more of the 32 ITM ports in the Serial Wire Viewer settings dialog.

#### Figure 189. SWV settings

Serial Wire Viewer settings for	or NUCLEO-F401RE Debug (ST-LINK)			×
Clock Settings Core Clock: 84 MHz Clock Prescaler: 42 SWO Clock: 2000.0 kHz	Trace Events CPI: Cycles per instruction EXC SLEEP: Sleep cycles FOLD: Folded instructions EXE	C: Exception overhead J: Load store unit cycles ETRC: Trace Exceptions	npling ble Resolution: 16384 V Cycles/sample amps ble Prescaler: 1 V	
Data Trace				
Comparator 0	Comparator 1	Comparator 2		Comparator 3
Enable	Enable	Enable		Enable
Var/Addr: pos1	Var/Addr: pos2	Var/Addr: 0x0		Var/Addr: 0x0
Access: Write	Access: Write	Access: Read/Write		Access: Read/Write
Size: Word 🗸 🗸	Size: Word	Size: Word	$\sim$	Size: Word
Generate: Data Value + PC 🗸	Generate: Data Value + PC \vee	Generate: Data Value	$\sim$	Generate: Data Value
ITM Stimulus Ports Enable port: 31	124 23	5 15	]8 7[[	OK Cancel

The packets from the ITM ports are displayed in the *SWV ITM Data Console* view. The CMSIS function <code>ITM\_SendChar()</code> can be used by the application to send characters to the port 0, and the <code>printf()</code> function can be redirected to use the <code>ITM\_SendChar()</code> function.

The following describes how to setup printf redirection over ITM:

1. Configure first file syscalls.c. Usually, the syscalls.c file is located in the same source folder as main .c.

If no syscalls.c file is available in the project, it can be copied from another STM32CubeIDE project. One way to get the file is to create a new STM32 empty project for the device. In the src folder, this project contains a syscall.c file. Copy this file to a source folder in the project where it is needed.

2. Inside the syscalls.c file, replace the \_write() function with code calling ITM\_SendChar() instead of
\_\_io\_putchar()

```
int _write(int file, char *ptr, int len)
{
    int DataIdx;
    for (DataIdx = 0; DataIdx < len; DataIdx++)
    {
        //__io_putchar(*ptr++);
        ITM_SendChar(*ptr++);
    }
    return len;
}</pre>
```

3. Locate the core\_cmX.h file, which contains the function ITM\_SendChar(). The core\_cmX.h file is included by the Device Peripheral Access Layer header file (for instance stm32f4xx.h, which in turn must be included in the syscalls.c file).

#include "stm32f4xx.h"

Use the *Include Browser* view to find the Device Peripheral Access Layer header file. Drop the core file in the *Include Browser* view, and check which files are including the core\_cmX.h file.

4. Test by adding include stdio.h and call to printf() into the application. Make sure that printf() is not called too often.

#include <stdio.h>

printf("Hello World %d\n", pos1);

- 5. Start a debug session and enable [ITM port 0] in the SWV ITM Data Console view.
- 6. Open the *SWV ITM Data Console* view and start tracing using the red [**Start/Stop Trace**] button on the toolbar in this view.
- 7. Start the program. Print commands are logged to the *Port 0* tab in the view.

#### Figure 190. SWV ITM Data Console

IDE			×
SWV ITM Data Console 🛛	*	e 🗙 🕞 🚮	<b>⊹</b> □ □
Port 0 🛛			
Hello World 1			$\sim$
Hello World 2 Hello World 3			
Hello World 4			
Hello World 5			
Hello World 6			
Hello World 7 Hello World 8			
Hello World 9			
Hello World 10			
			~

8. It is possible to open new port x tabs (x from 1 to 31) by pressing the green [+] button on the toolbar. This opens the *Add Port* dialog. In the dialog select the **[ITM Port number]** to be opened to display it as a tab in the *SWV ITM Data Console*view.

#### Figure 191. SWV ITM port configuration

IDE Add Port		×
ITM Port number {0}	: 1 ~	
ОК	Cancel	

*Note:* Study the ITM\_SendChar() function to learn how to write a function that transmits characters to another ITM port channel.

#### 4.3.6 SWV Statistical Profiling

The *SWV Statistical Profiling* view displays statistics based on Program Counter (PC) sampling. It shows the amount of execution time spent within various functions. This is useful when optimizing code. The data can be copied and pasted into other applications. The view is updated when debugging is suspended.

1. Configure SWV to send Program Counter samples, as shown in Figure 192. Enable [**PC Sampling**] and [**Timestamps**].

With the given [Core Clock] cycle intervals, SWV reports the Program Counter values to STM32CubeIDE. Set the [PC Sampling] to a high [Cycle/sample] value to avoid interface overflow.

Clock Settings			Trace Events		PC Sam	npling					
Core Clock:	84	MHz	CPI: Cycles per instruction	(C: Exception overhead	🗹 Ena	$\overline{2}$ Enable Resolution: 16384 $\checkmark$ Cycles/samp					
Clock Prescaler:	42		SLEEP: Sleep cycles	SLEEP: Sleep cycles LSU: Load store unit cycles		Timestamps					
SWO Clock:	2000.0	kHz	FOLD: Folded instructions EXETRC: Trace Exceptions			Enable Prescaler: 1					
Data Trace											
Comparator 0			Comparator 1	Comparator 2		Comparator 3					
Enable			Enable	Enable		Enable					
Var/Addr: pos	s1		Var/Addr: pos2	Var/Addr: 0x0		Var/Addr: 0x0					
Access: Wri	Write $\checkmark$ Access: Write $\checkmark$		Access: Read/Write	$\sim$	Access: Read/Write						
Size: Wo	Size: Word 🛛 Size: Word 💛 Size: Word		Size: Word	$\sim$	Size: Word						
Generate: Dat	a Value	+ PC	Generate: Data Value + PC 👋	Generate: Data Value	$\sim$	Generate: Data Value					
TM Stimulus Po	rts										
Enable port: 3					ר א ד						
Privileged only	ports: [	Port	3124 Port 2316 Port 158	_ Port 70							
						OK Cancel					

Figure 192. SWV PC sampling enable

- 2. Open the SWV Statistical Profiling view by selecting [Window]>[Show View]>[SWV Statistical Profiling]. The view is empty since no data is collected yet.
- 3. Press the red [Start/Stop Trace] button to send the configuration to the board.
- 4. Resume program debugging. STM32CubeIDE starts collecting statistics about function usage via SWV when the code is executing in the target system.
- 5. Suspend (Pause) the debugging. The view displays the collected data. The longer the debugging session, the more statistics are collected.

57/

Figure 1	93.	SWV	Statistical	Profiling
----------	-----	-----	-------------	-----------

IDE							×
SWV Statistical Pr	rofiling 🛛					* 🔵	×
Function	% in use	Samples	Start addr	Size			
readSpeed()	24.82%	20413	0x80003bd 0x80004d5	0x100 0x46			
readTemp()	16.04%	13191 56	0x800051b	0x2e			
SysTick_Handler()	0.07%	8	0x800001a	0x54 0xc			
writeSpeed()	0.00%	1	0x8000549	0x46			
Overflow packets: (	0 PC Samp	oles: 82244					

Note: A double-click on a function line in the SWV Statistical Profiling view opens the file containing the function in the editor.

The column information in the SWV Statistical Profiling is described in Table 10.

#### Table 10. SWV Statistical Profiling columns details

Name	Description
Function	The name of the function which is calculated by comparing address information in SWV packets with the program $\tt elf$ file symbol information.
% in use	The calculated percentage of time the function is used.
Samples	The number of samples received from the function.
Start address	The start address for the function.
Size	The size of the function.

### 4.4 Change the SWV trace buffer size

The incoming SWV packets are saved in the Serial Wire Viewer trace buffer, which has a default maximum size of 2 000 000 packets. To trace more packets, this figure must be increased.

Select the [Windows]>[Preferences] menu. In the *Preferences* dialog, select [STM32Cube]>[Serial Wire Viewer]. Update [Trace buffer size] if needed.

IDE Preferences				×
type filter text	Serial Wire Viewer			• 🖒 • •
<ul> <li>General</li> <li>C/C++</li> <li>Help</li> <li>Install/Update</li> <li>Remote Development</li> <li>Remote Systems</li> <li>Run/Debug</li> <li>STM32Cube     Build     Device Configuration Tool     File Association     Firmware Updater     Serial Wire Viewer     Target Status</li> <li>Team     Terminal</li> </ul>	Serial Wire Viewer Trace buffer size: 200000	) Restore Defaults	Ар	ply
		Apply and Close	Canc	el

#### Figure 194. SWV Preferences

The buffer is stored in the heap. The allocated heap is displayed by first selecting the [Windows]>[Preferences] menu. In the *Preferences* dialog, select [General]. Enable [Show heap status] to display the current heap and allocated memory in the bottom right corner of STM32CubeIDE. There is an upper limit to the amount of memory STM32CubeIDE can allocate. This limit can be increased to store more information during a debug session.

To update the memory limit, proceed as follows:

- 1. Navigate to the STM32CubeIDE installation directory. Open the folder in which the IDE is stored.
- 2. Edit the stm32cubeide.ini file and change the -Xmx1024m parameter to the desired size in megabytes.
- 3. Save the file and restart STM32CubeIDE.

#### 4.5 Common SWV problems

The following issues can occur when attempting to debut with SWV tracing:

- SWV is not enabled in the debug configuration currently used.
- The SWV Trace is not started, the red Start/Stop Trace button on the toolbar in some SWV view needs to be pressed to enable SWV and send SWV configuration to the target board. Then start the program to receive SWV data. For some SWV views the program then needs to be stopped again to visualize received SWV information.
- The SWO receives an excess of data. Reduce the amount of data enabled for tracing.
- The JTAG probe, the GDB server, the target board, or possibly some other part, does not support SWV.



The target [Core Clock] is incorrectly set. It is very important to select the right [Core Clock].
 If the frequency of the target [Core Clock] is unknown, it can sometimes be found by setting a breakpoint in a program loop and open the *Expressions* view, when the breakpoint is hit.
 Click on [Add new expression], type SystemCoreClock and press [Enter]. This is a global variable that, according to the CMSIS standard, must be set by the software to the correct speed of the [Core Clock].
 In CMSIS standard libraries, a function called SystemCoreClockUpdate() can be included in main() to set the SystemCoreClock variable. Use the Variable view to track it.

# *Note:* If the software dynamically changes the CPU clock speed during runtime, this might cause SWV to stop as the clocking suddenly becomes wrong during execution.

To make sure that all data is received, apply the following steps:

- 1. Open the SWV configuration. Disable all tracing except [**PC Sampling**] and [**Timestamps**]. Set the [**Resolution**] to the highest possible value.
- 2. Save, and open the SWV Trace Log view.
- 3. Start tracing.
- 4. Make sure that incoming packets can all be seen in the SWV Trace Log view.

# 5 Special Function Registers (SFRs)

### 5.1 Introduction to SFRs

Special Function Registers (SFRs) can be viewed, accessed and edited via the *SFRs* view. The view displays the information for the current project. Its content changes if another project is selected. To open the view from the menu, select the [**Window**]>[**Show View**]>[**SFRs**] menu command or use the [**Quick Access**] field, search for *"SFR"*, and select it from the views.

SFR		•
Views	📟 SFRs (Debug)	
Commands	Show In (SFRs)	
	Show View (SFRs) - Shows a particular view	
Help	😵 Search ' <b>SFR</b> ' in Help	

#### Figure 195. Open the SFRs view using the [Quick Access] field

#### 5.2 Using the SFRs view

The *SFRs* view contains information about peripherals, registers and bit fields for the STM32 device used in the project. When debugging the project, the registers and bit fields are populated with the values read from the target. The view contains two main nodes, the Cortex<sup>®</sup>-M node and the STM32 node. The Cortex<sup>®</sup>-M node includes common Cortex<sup>®</sup>-M core information and the STM32 node includes the STM32 device specific peripherals.

IDE				_			>	<
📟 SFRs 🔀	RD	X <sub>16</sub> X <sub>1</sub>	x2	×	ď	000		
type filter text								
Register	Address	Value						^
∽ 🧇 Cortex_M4								
> 👬 Control								
> 👬 FPE								
> 👬 ID								
> 👬 MPU								
> m NVIC								
> 👬 SysTick								
✓ ♦ STM32F401								
> ## ADC_Common								
> ## ADC1								
> ## CRC								
> ## DBG								
> ## EXTI								
> m FLASH								
✓ iiii IWDG								
> 1111 KR	0x40003000							
> 6161 PR	0x40003004	0x0						
V bibi RLR	0x40003008	0xttt						
INIT RL	[0:12]	Uxttt						
	0x4000300c	UXU						
S M OTG_FS_GLOBAL								
								~
MSB 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0	1 1	1 1	1 1	1 1	1	1 1
Parister PLP								
Address: 0x40003008								
Value: 0xfff								
Size: 32								
Reset value: 0xfff								
Access permission: RW								
Read action:								
Description:								
Keload register								

#### Figure 196. SFRs view

The top of the *SFRs* view contains a search field to filter visible nodes, such as peripherals, registers, bit fields. Upon text entry in the search field, only the nodes containing this text are displayed.

The information at the bottom of the *SFRs* view displays detailed information about the selected line. For registers and bit fields, this includes [Access permission] and [Read action] information.

The [Access permission] contains the following details:

- [RO](read-only)
- [WO](write-only)
- [RW](read-write)
- [W1](writeOnce)
- [**RW1**](read-writeOnce)

The Read action contains information only if there is a read action when reading the register or bit field:

- [clear]
- [set]
- [modify]
- [modifyExternal]


#### The toolbar buttons are located at the top-right corner of the SFRs view.

#### Figure 197. SFRs view toolbar buttons



The **[RD]** button in the toolbar is used to force a read of the selected register. It causes a read of the register even if the register, or some of the bit fields in the register, contains a ReadAction attribute set in the SVD file. When the register is read by pressing the **[RD]** button, all the other registers visible in the view are read again also to reflect all register updates.

The program must be stopped to read registers.

The base format buttons ([X16], [X10], [X2]) are used to change the registers display base.

The [**Configure SVD settings**] button opens the *CMSIS-SVD Settings Properties Panel* for the current project. The [**Pin**] button ("don't follow" selection) can be used to keep focus on the current displayed SVD file even if the *Project Explorer* view is switched to another project.

## 5.3 Updating CMSIS-SVD settings

The SFRs view for a project can display two CMSIS-SVD (System View Description) files for this project:

- The default file selected by STM32CubeIDE is the SVD file for the selected device in the project
- The other file can be a custom SVD file made to visualize specific user hardware configuration

To update the settings, use the [**Configure SVD settings**] toolbar button in the *SFRs* view to open the *CMS/S-SVD Settings* properties.

IDE Properties for NUCLEO-F401R	E —		$\times$
	CMSIS-SVD Settings	¢ •	⇔ ▼ ▼
> Resource > C/C++ Build	CMSIS SVD (System View Description) Data Files		
> C/C++ General	Device file platform:/plugin/com.st.stm32cube.ide.mcu.productdb.debug/resources/cmsis/STMicroelectronics_CMSIS_SVD/STM32F401.svd	Br	rowse
Project References	Custom file	Br	rowse
Run/Debug Settings			
	Restore Defaults	Арр	bly
0	Apply and Close	Cance	2

#### Figure 198. SFRs CMSIS-SVD Settings

All SVD files must comply with the syntax outlined in the CMSIS-SVD specification available on Arm<sup>®</sup> website. If these requirements are not met, the *SFRs* view is likely not to show any register information.

The [**Device file**] field is used for the System View Description (SVD) file. This file must describe the whole device. Other views may fetch information from the SVD file pointed out by this field, therefore it is recommended to use this field only for SVD files containing full STM32 device description. Updated SVD files can be obtained from STMicroelectronics (refer to the *HW Model, CAD Libraries and SVD* columns in the device description section on the STMicroelectronics website at *www.st.com*.

The [**Custom file**] field can be used to define special function registers related to custom hardware, in order to simplify the viewing of different register states. Another possible use case is to create an SFR favourites' file, containing a subset of the content in the [**Device file**]. This subset may be for instance composed of frequently checked registers. If a [**Custom file**] is pointed out, a new top-node in the *SFRs* view is created, which contains the [**Custom file**] related register information.

Both fields may be changed by the user and both fields may be used at the same time.

Note:

- It is possible to write new values in the value columns of registers and bit fields when these have write access permission.
- It is possible to use the SFRs view while the target is running when using the ST-LINK GDB server. However the [Live expression] option in the debug configuration must be enabled in this case.
- It is not possible to use SFRs view while the target is running when using OpenOCD or SEGGER J-Link.
- The SFRs view can also be useful in the C/C++ Editing perspective, however then only the names and addresses of the registers are displayed.

## 6 RTOS-aware debugging

Real-time operating systems (RTOS) add different kinds of objects to the design such as threads, semaphores, and timers. STM32CubeIDE includes dedicated set of views to handle Microsoft<sup>®</sup> Azure<sup>®</sup> RTOS ThreadX and FreeRTOS<sup>™</sup> kernel objects.

These views visualize the status of the RTOS objects when stepping through the code or when the program hits a breakpoint during a debug session.

Note:FreeRTOS is a trademark of Amazon in the United States and/or other countries.All other trademarks are the property of their respective owners.

## 6.1 Azure<sup>®</sup> RTOS ThreadX

The following views are available for ThreadX:

- ThreadX Thread List
- ThreadX Semaphores
- ThreadX Mutexes
- ThreadX Message Queues
- ThreadX Event Flags
- ThreadX Timers
- ThreadX Memory Block Pools
- ThreadX Memory Byte Pools

## 6.1.1 Finding the views

In the *Debugger* perspective, the ThreadX-related views are opened from the menu. Select the menu command [Window]>[Show View]>[ThreadX]>[...] or use [Quick Access] and search for "ThreadX" and select it from the views.

#### Figure 199. ThreadX views selectable from the menu

	Wind	dow	Help						
		New	Window		0	• 💁 • 🍅 🛷 • 🗉	▲ 월 ▼ 福 ▼ 🎨		> ▼ <> ▼   🛃   🚺
		Edito	or	>					Q i 🖻   🖬 🔯 🚥
		Арр	earance	>	k_tin	ne_get.c 🛛 🔭	🗆 🔲 🙁 🖂	•	● B 🙀 E 🚻 R 🎇 🗖 🗖
1		Shov	v View	>	60	ThreadX	>	60	ThreadX Thread List
		Pers	pective	>	60	FreeRTOS	>	60	ThreadX Semaphores
		Navi	gation	>	▦	SWV	>	60	ThreadX Mutexes
			-		•	Breakpoints	Alt+Shift+Q, B	60	ThreadX Message Queues
		Prefe	erences	_	010	Build Analyzer		69	ThreadX Event Flags
					₽	Console	Alt+Shift+Q, C	60	ThreadX Timers
Γ					蓉	Debug		60	ThreadX Memory Block Pools
_1	time	r_sy	stem_clock			Debugger Console		60	ThreadX Memory Byte Pools

## 6.1.2 ThreadX Thread List view

The *ThreadX Thread List* view displays detailed information regarding all available threads in the target system. The thread list is updated automatically each time the target execution is suspended.

There is one column for each type of thread parameter, and one row for each thread. If the value of any parameter for a thread has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

106											×
<b>19</b> T	(Thread X Thread List 🔅										
	Name	Priority	State	Run Count	Stack Start	Stack End	Stack Size	Stack Ptr	Stack	Usage	
	Main Thread	5	SUSPENDED (MySemaphore_1)	1	0x24001634	0x24001833	512	0x240016cc	Disab	led	
	System Timer Thread	0	SUSPENDED	0	0x24000a18	0x24000e17	1024	0x24000dcc	Disab	led	
	Thread One	10	SUSPENDED (MyMutex_1)	1	0x2400183c	0x24001a3b	512	0x240018d4	Disab	led	
$\rightarrow$	Thread Two	10	RUNNING	1	0x24001a44	0x24001c43	512	0x24001bfc	Disab	led	
	Idle										

#### Figure 200. ThreadX Thread List view (default)

Due to performance reasons, the *Stack Usage* column is disabled by default. To enable the stack analysis, use the [**Toggle Stack Checking**] toolbar button (circled in pink in Figure 201) in the *ThreadX Thread List* view toolbar.



IDE											
<b>ј (</b> Т	Thread K Thread List 82								- (	-	
	Name	Priority	State	Run Count	Stack Start	Stack End	Stack Size	Stack Ptr	Stack Usage		1
	Main Thread	5	SUSPENDED (Event Flag)	2	0x240015a4	0x240017a3	512	0x24001614	512		
	System Timer Thread	0	SUSPENDED	505	0x24000984	0x24000d83	1024	0x24000c4c	1024		
	Thread One	10	READY	381	0x240017ac	0x240019ab	512	0x24001824	512		
⇒	Thread Two	8	RUNNING	126	0x240019b4	0x24001bb3	512	0x24001b0c	512		
	ldle										

The column information in the *ThreadX Thread List* view is described in Table 11.

#### Table 11. ThreadX Thread List details

Name	Description
N/A	A green arrow symbol indicates the currently running thread.
Name	The name assigned to the thread.
Priority	The thread priority.
State	The current state of the thread.
Run Count	The threads run counter.
Stack Start	The start address of the stack area.
Stack End	The end address of the stack area.
Stack Size	The size of the stack area (bytes).
Stack Ptr	The address of the stack pointer.
	The maximum thread stack (bytes).
Stack Usage	By default, ThreadX fills every byte of thread stacks with a $0 \times EF$ data pattern during thread creation. See the note below for more information.

Note:

If the Stack Usage column contains the same values as the Stack Size column for all threads, the reason could be that the thread stack has not been filled with the 0xEF data pattern during task creation. This happens if the ThreadX kernel is built with stack data pattern filling disabled. Normally, a <tx\_user.h> file is used, which contains a TX\_DISABLE\_STACK\_FILLING define. Comment this define as shown in the example below and rebuild the project to solve the problem. It is good to know that the <tx\_user.h> file also contains a TX\_ENABLE\_STACK\_CHECKING define, which can be enabled to get run-time stack checking if stack corruption is detected. Additional information can be found in the ThreadX user guide.

Example of tx user.h file from ThreadX header file with commented TX ENABLE STACK CHECKING define:

- /\* Determine is stack filling is enabled. By default, ThreadX stack filling is enabled, which places an 0xEF pattern in each byte of each thread's stack. This is used by debuggers with ThreadX-aw:areness and by the ThreadX run-time stack checking feature. \*/
- /\* #define TX DISABLE STACK FILLING \*/
- /\* Determine whether or not stack checking is enabled. By default, ThreadX stack checking is disabled. When the following is defined, ThreadX thread stack checking is enabled. If stack

```
checking is enabled (TX_ENABLE_STACK_CHECKING is defined), the TX_DISABLE_STACK_FILLING
define is negated, thereby forcing the stack fill which is necessary for the stack
checking
logic. */
```

/\*#define TX ENABLE STACK CHECKING\*/

## 6.1.3 ThreadX Semaphores view

The *ThreadX Semaphores* view displays detailed information regarding all available resource semaphores in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of semaphore parameter, and one row for each semaphore. If the value of any parameter for a particular semaphore has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

IDE				×
🙀 ThreadX Semaphores 🖇	3			
Name	Count	Suspended		
MySemaphore_1	0	Main Thread		
MySemaphore_2	5			
MySemaphore_3	8			

#### Figure 202. ThreadX Semaphores view

#### Table 12. ThreadX Semaphores details

Name	Description
Name	The name assigned to the semaphore.
Count	The current semaphore count.
Suspended	The threads currently suspended because of the semaphore state.

## 6.1.4 ThreadX Mutexes view

The *ThreadX Mutexes* view displays detailed information regarding all available mutexes in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of mutex parameter, and one row for each mutex. If the value of any parameter for a particular mutex has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

Figure	203.	ThreadX	<b>Mutexes</b>	view
--------	------	---------	----------------	------

IDE				_	×
😭 ThreadX Mutexes 🛛					
Name	Owner	Owner Count	Suspended		
MyMutex_1	Main Thread	1	Thread One, Thread Two		
MyMutex_2		0			
MyMutex_3		0			

#### Table 13. ThreadX Mutexes details

Name	Description
Name	The name assigned to the mutex.
Owner	The thread that currently owns the mutex.
Owner Count	The mutex owner count (number of get operations performed by the owner thread).
Suspended	The threads currently suspended because of the mutex state.

## 6.1.5 ThreadX Message Queues view

The *ThreadX Message Queues* view displays detailed information regarding all available message queues in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of message queue parameter, and one row for each message queue. If the value of any parameter for a particular message queue has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

#### Figure 204. ThreadX Message Queues view

IDE								×
🎲 ThreadX Message Queues 🙁								
Name	Address	Capacity	Used	Free	Message size	Suspended		
Message Queue One	0x240002b8	10	0	10	1			
Message Queue Two	0x2400003c	10	0	10	1			

#### Table 14. ThreadX Message Queues details

Name	Description
Name	The name assigned to the message queue.
Address	The address of the message queue.
Capacity	The maximum number of entries allowed in the queue.
Used	The current number of used entries in the queue.
Free	The current number of free entries in the queue.
Message size	The size (in 32-bit words) of each message entry.
Suspended	The threads currently suspended because of the message queue state.

#### 6.1.6 ThreadX Event Flags view

The *ThreadX Event Flags* view displays detailed information regarding all available event flag groups in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of parameter, and one row for each event flag group. If the value of any parameter for a particular event flag group has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

IDE			_	×
🎲 ThreadX Event Flags 🛛				
Name	Flags	Suspended		
Event Flag1	0	Main Thread		
Event Flag2	0			

#### Figure 205. ThreadX Event Flags view

## Table 15. ThreadX Event Flags details

Name	Description
Name	The name assigned to the event flag group.
Flags	The current value of the event flag group.
Suspended	The threads currently suspended because of the event flag group.

## 6.1.7 *ThreadX Timers* view

The *ThreadX Timers* view displays detailed information regarding all available software timers in the target system. The timers view is updated automatically each time the target execution is suspended.

There is one column for each type of timer parameter, and one row for each timer. If the value of any parameter for a particular timer has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

IDE			– 🗆 X
🞲 ThreadX Timers 🛛			
Name	Remaining	Re-init	Function
MyTimer_1	68	100	0x80005d1 < MyTimerFunction1>
MyTimer_2	72	200	0x80005f3 < MyTimerFunction2>
MyTimer_3	276	500	0x8000615 < MyTimerFunction3>

## Figure 206. ThreadX Timers view

## Table 16. ThreadX Timers details

Name	Description
Name	The name assigned to the timer.
Remaining	The remaining number of ticks before the timer expires.
Re-init	The timer re-initialization value (ticks) after expiration. It contains value 0 for one-shot timers.
Function	The address and name of the function that is called when the timer expires.

## 6.1.8 ThreadX Memory Block Pools view

The *ThreadX Memory Block Pools* view displays detailed information regarding all available memory block pools in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of parameter, and one row for each memory block pool. If the value of any parameter for a particular memory block pool has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

#### Figure 207. ThreadX Memory Block Pools view

IDE							_	×
😭 ThreadX Memory Block	Pools 🛛							
Name	Address	Used	Free	Total	Block size	Pool size	Suspended	
MyBlockPool_1	0x240005ec < P	0	3	3	28	100		
MyBlockPool_2	0x240004ac < P	0	4	4	40	200		
MyBlockPool_3	0x2400034c <p< th=""><th>0</th><th>5</th><th>5</th><th>52</th><th>300</th><th></th><th></th></p<>	0	5	5	52	300		

#### Table 17. ThreadX Memory Block Pools details

Name	Description
Name	The name assigned to the memory block pool.
Address	The starting address of the memory block pool.
Used	The current number of allocated blocks.
Free	The current number of free blocks.
Total	The total number of memory block pools available.
Block size	The size (bytes) of each block.
Pool size	The total pool size (bytes).
Suspended	The threads currently suspended because of the memory block pool state.

#### 6.1.9 ThreadX Memory Byte Pools view

The *ThreadX Memory Byte Pools* view displays detailed information regarding all available memory byte pools in the target system. The view is updated automatically each time the target execution is suspended.

There is one column for each type of parameter, and one row for each memory byte pool. If the value of any parameter for a particular memory byte pool has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

## Figure 208. ThreadX Memory Byte Pools view

IDE						_	×
🔗 ThreadX Memory Byte P	Pools 🛛						
Name	Address	Used	Free	Size	Fragments	Suspended	
Byte Pool	0x24000f84 "\214\021"	1664	6528	8192	7		

## Table 18. ThreadX Memory Byte Pools details

Name	Description
Name	The name assigned to the memory byte pool.
Address	The starting address of the memory byte pool.
Used	The current number of allocated bytes.
Free	The current number of free bytes.
Size	The number of fragments.
Fragments	The size (bytes) of each block.
Suspended	The threads currently suspended because of the memory byte pool state.

57

## 6.1.10 Azure<sup>®</sup> RTOS TraceX tool

Important: The Microsoft<sup>®</sup> Azure<sup>®</sup> RTOS TraceX tool (TraceX) only exists for Windows<sup>®</sup>.

To open TraceX automatically upon data export, select the [Windows]>[Preferences] menu to associate the file type .trx with TraceX through the *Preferences* window as shown in Figure 209.

IDE Preferences				$\times$	
type filter text	File Associations		<b>⇔</b> ▼ <	5 <b>v</b>	8
✓ General	See 'Content Types' for content-type based file associations.				
> Appearance					
Compare/Patch	Open unassociated files with: System Editor; if none: lext Editor				
Content Types	File types:				
✓ Editors	i atta atta atta atta atta atta atta at		Add.		
Autosave	🗟 *.dtsi		Remov		
File Associations	●*.htm	_	Bernov	re:	-
> lext Editors					
Kous	e *.ioc				
Link Handlers	bl.* 🛍				
> Network Connections	@*.shtml				
Perspectives	®*.trx				
Project Natures					
Quick Search					
Search					
> Security					
> Startup and Shutdown					
UI Freeze Monitoring					
> User Storage Service					
Web Browser					
> Workspace					
> C/C++					
> Help					
<ul> <li>Instali/Opdate</li> <li>Remote Development</li> </ul>					
> Run/Debug					
> STM32Cube					
SWIChart Extensions					
Terminal					
> Version Control (Team)					
		-			
	Associated editors:				
	TraceX (default)		Add.		П
			Remo	10	i
			пешол	ne -	
			De <u>f</u> au	lt	
		_			
	Apply and Close		Cancel		

#### Figure 209. File associations

The Azure<sup>®</sup> RTOS ThreadX kernel can generate various system events into the MCU RAM buffer. These events can later be analyzed *"off target"* by the TraceX application. This requires an export of the RAM buffer to a suitable file format. Trace data can be exported using the [**Export trace**] button available from the menu of any Azure<sup>®</sup> RTOS ThreadX view as shown in Figure 210 and Figure 211.

۲	Console 🗈 Problen	ns 🖸 I	Executables 🗟 Debugger	Console 0 N	Memory 🔐 T	hreadX Threa	ad List ⊠			≡ 8 □ □
	Name	Pri	State	Run Count	Stack Start	Stack End	Stack Size	Stack Ptr	Stack Usa	View Menu
	sine wave	8	SLEEP (1)	124	0x24002	0x24002	1024	0x24002	Disabled	
	System Timer T	0	SUSPENDED	123	0x24013	0x24013	1024	0x24013	Disabled	
	thread 0	1	SLEEP (10)	13	0x24000	0x24000	1024	0x24000	Disabled	
	thread 1	16	READY	1874	0x24000	0x24000	1024	0x24000	Disabled	
-	thread 2	16	RUNNING	1876	0x24000	0x24000	1024	0x24000	Disabled	
	thread 3	8	SUSPENDED (semaph	62	0x24000	0x24001	1024	0x24000f	Disabled	
	thread 4	8	SLEEP (2)	62	0x24001	0x24001	1024	0x24001	Disabled	
	thread 5	4	SUSPENDED (event fl	13	0x24001	0x24001	1024	0x24001	Disabled	
	thread 6	8	SUSPENDED (mutex 0)	62	0x24001	0x24001	1024	0x24001	Disabled	
	thread 7	8	SLEEP (2)	62	0x24001	0x24002	1024	0x24001f	Disabled	
	Idle									

## Figure 210. RAM buffer export (1 of 2)

## Figure 211. RAM buffer export (2 of 2)

	Console 🗈 Probler	ns 이	Executables 🗟 Debugger	Console 🛛 I	Memory 🔐 T	hreadX Thre	ad List 🛙				≡ 8	- 0
	Name	Pri	State	Run Count	Stack Start	Stack 🤗 T	hreadX Sem	aphores			Show view	$\rightarrow$
	sine wave	8	SLEEP (1)	124	0x24002	0x24(	hreadX Mut	exes			Export trace	
	System Timer T	0	SUSPENDED	123	0x24013	0x24(	hreadX Mes	sage Queues		d		
	thread 0	1	SLEEP (10)	13	0x24000	0x24(	hreadX Even	it Flags		d		
	thread 1	16	READY	1874	0x24000	0x24( 🐖 T	hreadX Men	norv Block Poc	ale	d		
-	thread 2	16	RUNNING	1876	0x24000	0x24( 🍻 T	hreadX Men	nory Byte Pool	s	d		
	thread 3	8	SUSPENDED (semaph	62	0x24000	0x24001	1024	0x24000f	Disable	ed		
	thread 4	8	SLEEP (2)	62	0x24001	0x24001	1024	0x24001	Disable	ed		
	thread 5	4	SUSPENDED (event fl	13	0x24001	0x24001	1024	0x24001	Disable	ed		
	thread 6	8	SUSPENDED (mutex 0)	62	0x24001	0x24001	1024	0x24001	Disable	ed		
	thread 7	8	SLEEP (2)	62	0x24001	0x24002	1024	0x24001f	Disable	ed		

There are four prerequisites to export traces:

- The Azure<sup>®</sup> RTOS ThreadX kernel must be built with trace enabled
  - The embedded STM32CubeMX editor provides GUI support to enable the trace events
- The function tx\_trace\_enable() must be called before any data can be exported
- The trace export operation must not be performed inside the kernel API to avoid data file corruption
- The RAM buffer can only be read when the target is halted

#### On run

When exporting, the export destination is prepopulated from the active debug context available at: <code>\$PATH\_TO \_PROJECT/TraceX/\$LAUNCH\_CONFIGNAME.trx</code>. If previous trace data already exists, the user is prompted whether to overwrite them as shown in Figure 212.

#### Figure 212. Existing trace overwrite

/* There is a message waiting in t		×
<pre>/* Setup source and destination po source = queue_ptr -&gt; tx_que destination = TX_VOID_TO_ULONG_PO size = queue_ptr -&gt; tx_que /* Copy message. Note that the sou incremented by the macro. */ TX_QUEUE_MESSAGE_COPY(source, dest /* Determine if we are at the end. if (source == queue_ptr -&gt; tx_queu</pre> Export ThreadX trace buffer to file Export ThreadX trace buffer to file Image: TX_VOID_TO_ULONG_PO Image: TX_VOID_TO_ULON	t. 1.7.0.mx_azurertos_stm32h	rowse
{ III ThreadX Trace Export X		
File already exists and will be overwritten by export. Continue?	OK Can	cel
OK Cancel	ThreadX Thread List 22	
b3 Debug [STM32 Cortex-M C/C++ Application]	Name sine wave	Priority 8

## On export

When exporting the data to TraceX, STM32CubeIDE reads the RAM buffer from the target. The corresponding data is then used to create a \*.trx file, which can later be opened with the TraceX tool. By default, a TraceX directory is created in the project, containing the .trx file.



#### Figure 213. TraceX analysis

Remember:

- The export function only works once TraceX is initialized (tx\_trace\_enable()). The tool exports the last N trace events.
- There is a risk of exported trace data corruption if the export is performed inside the kernel API. To avoid such a
  corruption, make sure that the export is performed when the target is at a suitable location. For instance, set a breakpoint
  outside the kernel API or configure the trace full callback.

## 6.2 FreeRTOS<sup>™</sup>

The following views are available for FreeRTOS<sup>™</sup>:

- FreeRTOS Task List
- FreeRTOS Timers
- FreeRTOS Semaphores
- FreeRTOS Queues

## 6.2.1 Requirements

To be able to populate the FreeRTOS<sup>TM</sup>-related views with detailed information about the RTOS status, some files in the FreeRTOS<sup>TM</sup> kernel must be configured. The following sections describes some required configurations. Consult the *FreeRTOS reference manual* for detailed information.

## 6.2.1.1 Enable trace information

The define configUSE\_TRACE\_FACILITY in freeRTOSConfig.h must be enabled (set to 1). It results in additional structure members and functions to be included in the build and enables for instance stack checking in the *FreeRTOS Task List* view and lists the semaphore types in the *FreeRTOS Semaphores* view.

#### Example:

freeRTOSConfig.h
#define configUSE TRACE FACILITY

1

## 6.2.1.2 Add to registry

The application software must call the vQueueAddToRegistry() function to make the *FreeRTOS Queues* and *FreeRTOS Semaphores* views able to display objects. The function adds an object to the FreeRTOS<sup>TM</sup> *Queue registry* and takes two parameters, the first is the handle of the queue, and the second is a description of the queue, which is presented in FreeRTOS<sup>TM</sup>-related views.

#### Example:

```
vQueueAddToRegistry(mailId, "osMailQueue");
vQueueAddToRegistry(osQueueHandle, "osQueue");
vQueueAddToRegistry(osSemaphoreHandle, "osSemaphore");
```

#### 6.2.1.3 RTOS profiling information

To get valid RTOS run time statistics, the application must set up a run time statistics time base. The time-base clock is recommended to run at least 10 times faster than the frequency of the clock used to handle the RTOS tick interrupt. To enable the FreeRTOS<sup>™</sup> collection of run time statistics, file freeRTOSConfig.h must include:

- 1. Define configGENERATE RUN TIME STATS 1
- 2. Define portCONFIGURE\_TIMER\_FOR\_RUN\_TIME\_STATS () to call the function that configures a timer to be used for profiling
- 3. Define portGET\_RUN\_TIME\_COUNTER\_VALUE() to call the function that reads the current value from the profiling timer

#### Example:

```
freeRTOSConfig.h
#define configGENERATE_RUN_TIME_STATS 1
#define portCONFIGURE_TIMER_FOR_RUN_TIME_STATS() configureRunTime()
#define portGET_RUN_TIME_COUNTER_VALUE() getRunTimeCounter()
```

#### Or, if a run time variable is available in the system:

```
freeRTOSConfig.h
#define configGENERATE_RUN_TIME_STATS 1
#define portCONFIGURE_TIMER_FOR_RUN_TIME_STATS() (RunTime=0UL)
#define portGET RUN_TIME_COUNTER_VALUE() RunTime
```

If the *Run Time* column in the *FreeRTOS Task List* view displays N/A after making these three settings, the problem can a arise if project is not built with optimization level -00. The reason is quite likely found in the declaration in tasks.c of ulTutoralRunTime.

Example:

```
#if ( configGENERATE_RUN_TIME_STATS == 1 )
    PRIVILEGED_DATA static uint32_t ulTaskSwitchedInTime = OUL;
    /*< Holds the value of a timer/counter the last time a task was switched in. */
    PRIVILEGED_DATA static uint32_t ulTotalRunTime = OUL;
    /*< Holds the total amount of execution time as defined by the run time counter clock. */
#endif</pre>
```

Solutions:

Either declare the variable as volatile:

```
PRIVILEGED_DATA volatile static uint32_t ulTotalRunTime = OUL;
/*< Holds the total amount of execution time as defined by the run time counter clock.
*/</pre>
```

- Or simply change the optimization level only for tasks.c by
  - 1. Right-clicking it in Project Explorer view and open Properties
  - 2. Select [Properties]>[C/C++ Build]>[Settings]>[Tool Settings]>[Optimization]
  - 3. Set [Optimization Level] to None (-00)

## 6.2.2 Finding the views

In the *Debugger* perspective, the FreeRTOS<sup>™</sup>-related views are opened from the menu. Select the menu command **[Window]**>**[Show View]**>**[FreeRTOS]**>[...] or use **[Quick Access**], search for *"FreeRTOS"* and select from the views.



#### Figure 214. FreeRTOS<sup>™</sup>-related views selectable from the menu

## 6.2.3 FreeRTOS Task List view

The *FreeRTOS Task List* view displays detailed information regarding all available tasks in the target system. The task list is updated automatically each time the target execution is suspended.

There is one column for each type of task parameter, and one row for each task. If the value of any parameter for a task has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow, as shown in the example in Figure 215.

#### Figure 215. FreeRTOS Task List (default)

										×
🧽 Fr	eeRTOS Task List 22								=	• •
	Name	Priority (Base/	Start of Stack	Top of Stack	State	Event Object	Min Free Stack	Run Time (%)		
-	IDLE	0/0	0x200003c0	0x20000558 <ucheap+1308></ucheap+1308>	RUNNING		Disabled	99%		
	LEDThread	3/3	0x20000150	0x20000308 <ucheap+716></ucheap+716>	SUSPENDED		Disabled	0%		
	Tmr Svc	2/2	0x20000630	0x200009b0 <ucheap+2420></ucheap+2420>	BLOCKED	TmrQ	Disabled	1%		
										_

57

Due to performance reasons, stack analysis (the *Min Free Stack* column) is disabled by default. To enable stack analysis (refer to Figure 217), use the *Toggle Stack Checking* toolbar button in the *FreeRTOS Task List* view toolbar as shown in Figure 216.

## Figure 216. FreeRTOS<sup>™</sup> Toggle Stack Checking



#### Figure 217. FreeRTOS Task List (Min Free Stack enabled)

01											×
ill Fre	FreeRTOS Task List 30										
	Name	Priority (Base/	Start of Stack	Top of Stack	State	Event Object	Min Free Stack	Run Time (%)			
$\rightarrow$	IDLE	0/0	0x200003c0	0x20000560 <ucheap+1316></ucheap+1316>	RUNNING		>256	99%			
	LEDThread	3/3	0x20000150	0x200002b8 <ucheap=636></ucheap=636>	DELAYED		>256	0%			
	Time Svc	2/2	0x20000630	0x200009b0 <ucheap=2420></ucheap=2420>	BLOCKED	TmrQ	>256	1%			
_											

The *FreeRTOS Task List* view in Figure 217 contains a *Min Free Stack* column. The column information is changed to *Stack Usage* if the project is built with the following define set:

#define configRECORD STACK HIGH ADDRESS

In this case, the full stack usage is presented according to the format Used/Total(%Used) as shown in Figure 218.

1

## Figure 218. FreeRTOS Task List with ConfigRECORD\_STACK\_HIGH\_ADDRESS enabled

ID	E									
<b>:</b> @	FreeRTOS Task List 🛛									
	Name	Priority (B	Start of S	Top of St	State	Event Object	Stack Usage	Run Time		
$\rightarrow$	IDLE	0/0	0x20000	0x20000	RUNNING		96B / 2052B (4.7%)	N/A		
	THREAD1	24/24	0x20001	0x20001	DELAYED		144B / 512B (28.1%)	N/A		
	THREAD2	24/24	0x20001	0x20001	DELAYED		144B / 512B (28.1%)	N/A		
	Tmr Svc	2/2	0x20000	0x20000	BLOCKED	TmrQ	168B / 1028B (16.3%)	N/A		

The column information in the FreeRTOS Task List view is described in Table 19.

#### Table 19. FreeRTOS Task List details

Name	Description
N/A	A green arrow symbol indicates the task currently running.
Name	The name assigned to the task.
Priority (Base/Actual)	The task base priority and actual priority. The base priority is the priority assigned to the task. The actual priority is a temporary priority assigned to the task due to the priority inheritance mechanism.
Start of Stack	The address of the stack region assigned to the task.
Top of Stack	The address of the saved task stack pointer.
State	The current state of the task.
Event Object	The name of the resource that has caused the task to be blocked.
Min Free Stack <sup>(1)</sup>	The stack "high watermark". Displays the minimum number of bytes left on the stack for a task. A value of 0 (most likely) indicates that a stack overflow has occurred.
	Note: This feature must be enabled in the "View" toolbar.
Run Time (%)	The run time statistics provide information on the percentage of time the task has been used. This can be used for profiling the system during development.

1. When the application is built with configRECORD\_STACK\_HIGH\_ADDRESS = 1, the column name is changed to "Stack Usage". It displays the stack usage in detailed format as "Used/Total(%Used)".

## 6.2.4 FreeRTOS Timers view

The *FreeRTOS Timers* view displays detailed information regarding all available software timers in the target system. The view is updated automatically each time the target execution is suspended. There is one column for each type of timer parameter, and one row for each timer. If the value of any parameter for a timer has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

## Figure 219. FreeRTOS Timers

IDE					-	$\times$
🔐 FreeRTOS Timers 🛛						
Name myTimerTEST	Active True	Period 200	Type Auto-Reload	ld 0x0	Callback 0x8000429 <ostimercallback></ostimercallback>	

## The column information in the FreeRTOS Timers view is described in Table 20.

#### Table 20. FreeRTOS Timers details

Name	Description
Name	The name assigned to the timer.
Active	The active status information.
Period	The time (in ticks) between timer start and the execution of the callback function.
Туре	The type of timer. Auto-reload timers are automatically reactivated after expiration. One-shot timers expire only once.
ld	The timer identifier.
Callback	The address and name of the callback function executed when the timer expires.

Note:

1. If no name appears in the Name field, check that the timer is created with a name. The first parameter when calling *xTimerCreate()* must contain the timer name string.

2. When using software timers, a *Tmr Svc* task and a *TmrQ* queue are created automatically. These objects are displayed in the FreeRTOS Task List view and FreeRTOS Queues view.

## 6.2.5 FreeRTOS Semaphores view

The *FreeRTOS Semaphores* view displays detailed information regarding all available synchronization objects in the target system, including:

- Mutexes
- Counting semaphores
- Binary semaphores
- Recursive semaphores

The view is updated automatically each time the target execution is suspended. There is one column for each type of semaphore parameter, and one row for each semaphore. If the value of any parameter for a semaphore has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

#### Figure 220. FreeRTOS Semaphores

IDE							×
🔐 FreeRTOS Semaphores 🔅	3						- 0
Name osSemaphore	Address 0x20000058	Type BINARY_SEMAPHORE	Size 1	Free 0	# Blocked tasks 0		

Note:

If the Type information displays N/A, make sure that the define configUSE\_TRACE\_FACILITY is enabled in file FreeRTOSconfig.h.

The column information in the FreeRTOS Semaphores view is described in Table 21.

#### Table 21. FreeRTOS Semaphores details

Name	Description
Name	The name assigned to the semaphore.
Address	The address of the object.
Туре	The type of the object.
Size	The maximum number of owning tasks.
Free	The number of free slots currently available.
#Blocked tasks	The number of tasks currently blocked waiting for the object.

## 6.2.6 FreeRTOS Queues view

The *FreeRTOS Queues* view displays detailed information regarding all available queues in the target system. The view is updated automatically each time the target execution is suspended. There is one column for each type of queue parameter, and one row for each queue. If the value of any parameter for a queue has changed since the last time the debugger was suspended, the corresponding row is highlighted in yellow.

#### Figure 221. FreeRTOS Queues

IDE									×	
🔐 FreeRTOS Queues 🕴	😭 FreeRTOS Queues 🙁									
Name osQueue	Address 0x20000068	Max Length 1	Item Size 2	Current Length 0	# Waiting Tx 0	# Waiting Rx 1				

The column information in the FreeRTOS Queues view is described in Table 22.

#### Table 22. FreeRTOS Queues details

Name	Description
Name	The name assigned to the queue in the queue registry.
Address	The address of the queue.
Max Length	The maximum number of items that the queue can hold.
Item Size	The size in bytes of each queue item.
Current Length	The number of items currently in the queue.
#Waiting Tx	The number of tasks currently blocked waiting to be sent to the queue.
#Waiting Rx	The number of tasks currently blocked waiting to be received from the queue.

## 6.3 RTOS-kernel-aware debug

The RTOS-kernel-aware debug in STM32CubeIDE supports the Microsoft<sup>®</sup> Azure<sup>®</sup> RTOS ThreadX and FreeRTOS<sup>™</sup> operating systems using an RTOS proxy. The RTOS proxy is included in STM32CubeIDE and can be used with ST-LINK GDB server, OpenOCD, and SEGGER J-Link GDB server.

When RTOS-kernel-aware debugging is enabled and a debug session is started, all threads are listed in the *Debug* view. By selecting a thread in the *Debug* view, the thread current context is visualized in views. For instance, the *Variables*, *Registers*, *Editor* views reflect the active stack frame.

Figure 222 shows a debug session. The *ThreadX Thread List* view displays that the Message Queue Receiver Thread is RUNNING. This can also be seen in the *Debug* view. In the *Debug* view the MsgSenderThreadTwo\_Entry function is selected, and the editor area displays that the thread is waiting in a sleep for 500 ms state.

## Figure 222. RTOS-kernel-aware debug

The Life source National Natio	workspace_um8 - Tx_Thread_MsgQueue/Application/User/app_threadx.c - STM32CubeIDE									-	0 × 0
<ul> <li>Compare length and the length of the length o</li></ul>	File Edit Source Refactor Navigate Search Project Run Window Help										
<ul> <li>         (b) (b) (c) (b) (c) (c) (c) (c) (c) (c) (c) (c) (c) (c</li></ul>	11 ▼ [1] [2] 2 ▼ [3] ▼ [3] ▼ [3] ▼ [3] ▼ [3] [3] [3] [3] [3] [3] [3] [3] [3] [3]	tod \$ ▼ \$ ▼   <b>1</b>   <b>8</b>									०. । 🖻 । 💀 🔯
<ul> <li>• Buthesd Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Applaces)</li> <li>• A Under Vegeene (BM2 cons + UC+ Appl</li></ul>	🏶 Debug 💠 📴 Project Explorer 👘 🐘 🕯 🖓 🗖	■ app threadx.c ≅ ■ tx thread sy ■	tx threa	ad s 🖻 tx queue	re 🖻 tx th	read s 🖻 txe gi	ueue r " 🗆	🚥 Var 🖾 🍫 Br	🚿 Ex 🛋 Mo.		■SF □ □
<ul> <li>* The start function of the start is a start of the start is a st</li></ul>	Tx_Thread_MsqQueue [STM32 Cortex-M C/C++ Application]	195 * @param thread_input:	Not u	ised			^			<u>انگ</u>	488
<pre>* dimension of the start (Manage Gause Start Bind Ded) (2007) 844 Segmended: Contact Bind Bind Field Ded) 844 Segmended Start (Manage Start Bind Ded) Bind Bind Bind Bind Bind Bind Bind Bind</pre>	✓ ₽ Tx_Thread_MsgQueue.elf	196 * @retval None						Name	Тур	e	Value
<pre>styles_value(style="type:</pre>	Thread #2 [Message Queue Sender Thread One] 603979944 (Suspended : Container)	197 */	+====(1)	I ONG thread inn	·+· \			↔ thread_inp	out ULO	NG	0
<pre></pre>	get_ipsr_value() at tx_port.h:368 0x8003e86	199 {	ici y(o	cond chireau_inpo	,()			👐 Msg	ULO	NG	1
<pre></pre>	_tv_thread_system_return_inline() at tx_port.h:447 0x8003e86	200 ULONG Msg = TOGGLE_LED;									
<pre>     #_sthead_skep_dtr_thead_skep_let_thead_skep_dtr_thead_sk</pre>	_tx_thread_system_suspend() at tx_thread_system_suspend.c:549 0x8003e86	<pre>201 (void) thread_input;</pre>									
<pre>e logicodIntestOre: [inty]: tage [integl::expl::the double: [integl::expl::the double::the double::the double::the double::t</pre>	_tx_thread_sleep() at tx_thread_sleep.c:189 0x8003a70	202 /* Infinite loop */ 203 while(1)									
<pre>     (, thread_lable_mrty) is the made_lable_mrty(14 0000393     () for a for blacksgo threads for the made (359802) for blacksgo to 2000 (36)     () for a for blacksgo threads for blacksgo threads (359802) for blacksgo to 2000 (36)     () for a for blacksgo thread for blacksgo thread (359802) for blacksgo to 2000 (36)     () for a for blacksgo thread for bla</pre>	MsgSenderThreadOne_Entry() at app_threadx.c:189 0x80007cc	204 {									
<pre> • Test state is made to be a front from the state is state i</pre>	_tx_thread_shell_entry() at tx_thread_shell_entry.c:114 0x8003938	205 /* Send message to Ms	gQueue	Two. */	TT FOREVER)		、 、				
<pre></pre>	= uxmmme	200 II (CX_queue_senu(ans	squeue	1wo, ansg, 1x_w	ATT_FOREVER)	1= 1X_30000233,	,				
<pre>* development must grant plant plant</pre>	<ul> <li>Printead #3 (Message Queue Receiver Intread) 603980120 (KUNINING) (Suspended : Signal : Signal</li> <li>tra gueue received at the gueue receive =103.0v80042de</li> </ul>	208 Error_Handler();									
<pre>* 1.thread table arrow(1) thread.table graph(2):110.0000383 * Universe * 1.thread table graph(2):110.0000383 * Universe * 1.thread table graph(2):110.0000386 * Universe * 1.thread</pre>	Interpretation of the second state of the second	209 }									
<pre>     outfittiin     outfittiin</pre>	Its thread chall ontrol at the thread chall ontrol c114 0x9002020	210 /* Sleep for 500s */									
<pre></pre>	=	212 }									
<pre> •</pre>	A Thread #4 (Massage Queue Sender Thread Two) 603980296 (Suspended : Container)	213 }									
<pre>s_theod ystem supervised it h:rest_ystems in the rest_ystem supervised it h:rest_ystems ystem ystem it h:rest_ystems ystem ystem it h:rest_ystems in the rest_if h:rest_if h:rest_if</pre>	meda #4 (message cacae senaer meda moj oossoosso (suspended : container)	214									
i. Lithord system suspends at the system suspend system suspend system suspend system suspend at the system suspend system suspend system suspend system system suspend system	tx thread system return inline() at tx port.h:447 0x8003e86	2150/** 216 * Obnief Eurotion impl	monti	ng the MagReceiv	enThread th	head					
<pre>     "sthead shep it tyhead shep its tyhead the its tyhead shep its tyhea</pre>	tx thread system suspend() at tx thread system suspend.c:549 0x8003e86	217 * @param thread input:	Not u	ised	ver milleau ch	neau.					
Maginder Marsdalve Linding Lind	tx thread sleep() at tx thread sleep.c:189 0x8003a70	218 * @retval None									
<pre> •</pre>	MsgSenderThreadTwo_Entry() at app_threadx.c:211 0x8000802	219 */		our shared down							
<ul> <li> <ul> <li></li></ul></li></ul>	_tx_thread_shell_entry() at tx_thread_shell_entry.c:114 0x8003938	220 Void Msgkeceiverinread_En	try(UL	ONG thread_input	=)						
<ul> <li></li></ul>	■ Oxffffffe	222 ULONG RMsg = 0;									
<pre>eet_ips: value() at v_Dorth/380 8000386 i=_trimed, at v_Dorth/380 8000386 i=_trime thread entry at thisme, trimed, at v_Dorth/380 8000386 i=_trime thread entry at thisme, trimed, at v_Dorth/380 8000386 i=_trime thread entry at thisme, trimed, at v_Dorth/380 8000386 i=_trime thread entry at thisme, trimed, at v_Dorth/380 8000386 i=_trime thread entry at thisme, trimed, at v_Dorth/380 8000386 i=_trime thread entry at thisme, trimed, at v_Dorth/380 8000386 i=_trime thread entry at thisme, trimed, at v_Dorth/380 8000386 i=_trime thread entry at thisme, trimed, at v_Dorth/380 8000386 i=_trime thread entry at thisme, trimed, at v_Dorth/380 8000386 i=_trime thread entry at v_Dorth/380 8000380 i=_trime thread entry at v_Dorth/380 8</pre>	<ul> <li>Phread #5 [System Timer Thread] 603980976 (Suspended : Container)</li> </ul>	<pre>223 UINT status = 0;</pre>									
<pre>s_thread system_submed_3 ts_thread_system_submed_c549 0x800x866</pre>	get_ipsr_value() at tx_port.h:368 0x8003e86	224 (void) thread_input;									
Suchard system suspend() at University 0x00030366     Suchard system suspend() at University 0x00030366     Suchard system suspend() at University 0x0003036     Suchard system suspend()     Suchard system     Suchard system     Suchard system system system system     Suchard system     Suchard system	_tx_thread_system_return_inline() at tx_port.h:447 0x8003e86	226 while (1)						<			>
status = tytime_thread_enty_04 tytime_thread_enty_c42 0.08004adt = tytime_thread_shell_enty_c114 0.08003938 = 0/fiffifie al arm-one sebigdb (03.12.0191211) al ST-UNK (ST-UNK GD8 server) al RTOS Proy          225       /* totes = TX_SUCCESS) 230       if (Rtsg l= rOSGLE_LED) 231       if (Rtsg l= rOSGLE_LED) 233       if (Rtsg l= rOSGLE_LED) 234       if (Rtsg l= rOSGLE_LED) 235       if (Rtsg l= rOSGLE_LED) 236       if (Rtsg l= rOSGLE_LED) 237       if (Rtsg l= rOSGLE_LED) 236       if (Rtsg l= rOSGLE_LED) 236       if (Rtsg l= rOSGLE_LED) 237       if (Rtsg l= rOSGLE_LED) 236       if (Rtsg l= rOSGLE_LED) 2	_tx_thread_system_suspend() at tx_thread_system_suspend.c:549 0x8003e86	227 {									
<ul> <li>Cythread_shell_entry_clt14 0x8003938</li> <li>Gyttmiffie</li> <li>af (status = TX_guce_mectave(kekgQueueOne, &amp;RMsg, TX_MO_kAIT);</li> <li>bf (status = TX_guce_mectave(kekgQueueOne, &amp;RMsg, TX_MO_kAIT);</li> <li>af (status = TX_guce_metrave(kekgQueueOne, &amp;RMsg, TX_MO_kAIT);</li> <li>bf (status = TX_guce_mectave(kekgQueueOne, &amp;RMsg, TX_MO_kAIT);</li> <li>bf (status = TX_guce_mectave(kekgQueueOne, &amp;RMsg, TX_MO_kAIT);</li> <li>bf (status = TX_guce_metrave(kekgQueueOne, Trave(KekgQueueOne, Trave(KekgQueueOne, Trave(KekgQueueOne, Trave(KekgQueueOne, Trave(KekgQu</li></ul>	_tx_timer_thread_entry() at tx_timer_thread_entry.c:462 0x8004adc	228 /* Determine whether	a mess	age MsgQueueOne	or MsgQueue	Two is availabl	le */				
■ Oxfiffife al arm-none-cabisdig (8.3.1.20191211) al ST-LINK (ST-LINK GOB server) al RTOS Proxy	_tx_thread_shell_entry() at tx_thread_shell_entry.c:114 0x8003938	229 status = tx_queue_rec 230 if (status == TX_SUCC	eive(ä =ss)	MsgQueueone, &R	isg, IX_NO_W	(AIT);					
	■ 0xffffffe	231 {									
All SH-LINK CD-LINK CDB server)       233       if (RPsgl != TOGGLE_LED)         JI RTOS Proxy       234       {       235       if (RPsgl != TOGGLE_LED)         235       if (RPsgl != TOGGLE_LED)       235       if (RPsgl != TOGGLE_LED)       if (RPsgl != TOGGLE_LED)         236       {       235       if (RPsgl != TOGGLE_LED)       if (RPsgl != TOGGLE_LED)       if (RPsgl != TOGGLE_LED)         238       {       235       if (RPsgl != TOGGLE_LED)       if (RPsgl != TOGGLE_LED)       if (RPsgl != TOGGLE_LED)         238       {       128       if (RPsgl != TOGGLE_LED)       if (RPsgl != TOGGLE_LED)       if (RPsgl != TOGGLE_LED)         238       {       128       Name       Result       if (RPsgl != TOGGLE_LED)       if (RPsgl != TOGGLE_LED)         238       {       128       Result       Result       Result       if (RPsgl != TOGGLE_LED)       if (RPsgl != TOGGLE_LED)         238       {       128       Result       Result       Result       Result       Stack Start       Stack Red       Stack Start       Stack Red       Stack Start       Stack Red       Stack Start       Stack Red	arm-none-eabi-gdb (8.3.1.20191211)	232 /* Check Message va	lue */								
AB KIOS Provy       235 \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	ST-LINK (ST-LINK GDB server)	233 if (RMsg != TOGGLE_	ED)								
<sup>236</sup> / <sub>237</sub> e <sup>13e</sup> / <sub>238</sub> <sup>237</sup> / <sub>238</sub> <sup>85</sup> / <sub>216</sub> <sup>238</sup> / <sub>238</sub> <sup>85</sup> / <sub>216</sub> <sup>116</sup> / <sub>216</sub> <sup>116</sup> / <sub>2166</sub>	all RIOS Proxy	234 [ 235 Error Handler():									
		236 }									
		237 else									
		238 { 230 RED   ED Togg]o(/ E									
<td< td=""><td></td><td>235 BSP_LED_TOggle(LE</td><td>_GNEE</td><td>···),</td><td></td><td></td><td>~</td><td></td><td></td><td></td><td><math>\sim</math></td></td<>		235 BSP_LED_TOggle(LE	_GNEE	···),			~				$\sim$
c       Console 12 Poolems © Executables @Debugger Console 10 Memory @ThreadX Thread List #       Stack End       Stack End       Stack End       Stack End       Stack List #       Stack End		<					>	<			>
Name       Pic       State       Run Court       Stack Ind       Stack		Console 🗈 Problems 🧿 Executables 🖣	Debug	ger Console 🏮 Memo	ory 📽 ThreadX 1	Thread List 🛙					=
<ul> <li>Message Queue Sender Thread</li> <li>Message Queue Sender Thread</li> <li>S</li> <li>SLEP (200)</li> <li>Resource Queue Sender Thread</li> <li>SUSPENDED</li> <li>SUSPENDED</li> <li>SUSPENDED</li> <li>SUSPENDED</li> <li>SUSPENDED</li> <li>SUSPENDED</li></ul>		Name	Pri	State	Run Count	t Stack Start	Stack End	Stack Size	Stack Ptr	Stack Usage	
Message Queues Bender Thread Two             5                74               0x240018               512               0x240013               0x240013               0x240013               0x240013               0x240013               0x240013               0x240013               0x240003		⇒ Message Queue Receiver Thread	10	RUNNING	875	0x2400139c	0x2400159b	512	0x240014fc	Disabled	
Message Queue Sender Thread Two         5         SLEEP (404)         30         0x24001393         512         0x24001264         Disabled           System Timer Thread         0         SUSPENDED         874         0x24000570         0x2400096F         1024         0x2400083c         Disabled           Idle         -		Message Queue Sender Thread One	5	SLEEP (200)	74	0x24000f8c	0x2400118b	512	0x2400105c	Disabled	
System Timer Thread         0         SUSPENDED         874         Ox240009570         Ox2400096f         1024         Ox2400083c         Disabled           Idle         Idle <td></td> <td>Message Queue Sender Thread Two</td> <td>5</td> <td>SLEEP (404)</td> <td>30</td> <td>0x24001194</td> <td>0x24001393</td> <td>512</td> <td>0x24001264</td> <td>Disabled</td> <td></td>		Message Queue Sender Thread Two	5	SLEEP (404)	30	0x24001194	0x24001393	512	0x24001264	Disabled	
Ide		System Timer Thread	0	SUSPENDED	874	0x24000570	0x2400096f	1024	0x2400083c	Disabled	
c		Idle									
c											
c											
	<	4									>
											5

To enable RTOS-kernel-aware debugging the *Debugger* tab in the *Debug Configurations* dialog contains settings to enable RTOS proxy, driver (RTOS ThreadX or FreeRTOS<sup>™</sup>), port (Cortex<sup>®</sup> core) and configuration of port number to use with the proxy.

The *RTOS* tab also contains a *Driver settings* selection to select the *Driver* (*"ThreadX"* or *"FreeRTOS"*) and the port used. The *"Auto-detect"* driver setting is still experimental.

Figure	223.	<b>RTOS-kerne</b>	l-awareness	debug	config	uration
Iguic	220.	INTOO-KOINC	-uwurchess	ucoug	conné	julution

Debug Configurations		— 🗆 X			
Create, manage, and run configura	ations				
" ┏ 🕫 🖿 🗙 🖻 🍸 🗕					
type filter text	Maine: Mocleo 140 Me				
C/C++ Application	GDB Connection Settings				
C/C++ Attach to Application	Autostart local GDB server Host name or IP ac	ddress localhost			
C/C++ Postmortem Debugger	○ Connect to remote GDB server Port number	61234			
GDB Hardware Debugging	Debug probe CT LINK (CT LINK CDD convert)				
🕏 Launch Group	GDP Server Command Line Options				
STM32 Cortex-M C/C++ Applicati STM32 Cortex-M C/C++ Applicati	Charles Command Line Options				
MOCLEO-F40 IKE	Snow Command Line				
		✓ Scan			
	Erequency (kHz): Auto				
	Access port: 0 - Cortex-M4				
	Access port. Or Contex INIA				
	Reset behaviour				
	Type: Connect under reset 🗸				
	Device settings				
	Debug in low power modes:				
	Suspend watchdog counters while halted: No config	guration ~			
	Serial Wire Viewer (SWV)	RTOS Kernel Awareness			
	✓ Enable	✓ Enable RTOS Proxy			
	Core Clock (MHz): 84.0	Driver settings			
	Limit SWO clock	Driver: ThreadX ~			
	Maximum SWO clock (kHz): auto detect	Port: cortex_m0 ~			
	Port number: 61235	Port number: 60000			
	Misc				
	✓ Verify flash download				
	Enable live expressions				
	C:\Users\jonansse\STM32C	uberDetworkspace_1.7.0.2 tw2 t(NOCLEO-F401KE Browse			
	External Loader:	Scan Initialize			
	Shared ST-LINK				
	wax ndit timeout(s): 2				
< >					
ilter matched 8 of 8 items		Revert Apply			
0		Debug Close			

## Figure 224. ThreadX-kernel-awareness debug configuration

FI	RTOS Ke	rnel Awareness		
ŀ	Enable RTOS Proxy			
	Driver	settings		
	Driver:	ThreadX	$\sim$	
	Port:	cortex_m0	$\sim$	
	Port nu	mber: 60000		

The port selection lists the supported cores. The items listed depend on the selected RTOS driver as displayed in Figure 225 and Figure 226.

RTOS	ernel Awareness ble RTOS Proxy
Driv	r settings
Driv	: ThreadX ×
Por	cortex_m0 ~
	cortex_m0
Port	u cortex_m3
	cortex_m4
	cortex_m7
	cortex_m33

## Figure 225. ThreadX port configuration

## Figure 226. FreeRTOS<sup>™</sup> port configuration

RTOS Ke	ernel Awareness le RTOS Proxy	
Driver	settings	
Driver:	FreeRTOS	~
Port:	ARM_CM0	~
	ARM_CM0	
Port nu	ARM_CM3	
	ARM_CM3_MPU	
	ARM_CM4F	
	ARM_CM4_MPU	
	ARM_CM7	
	ARM_CM33	
oace_1.7.0	ARM_CM33_NTZ	_

#### **Known limitations**

- · Live expressions must be disabled when used with the ST-LINK GDB server
- The *Registers* view content for swapped out threads is intermixed with active CPU context for some registers (all registers are not saved by the context switcher)
- The Registers view floating point registers are not updated correctly

51

# 7 Fault Analyzer

## 7.1 Introduction to the Fault Analyzer

The STM32CubeIDE Fault Analyzer feature interprets information extracted from the Cortex<sup>®</sup>-M nested vector interrupt controller (NVIC) in order to identify the reasons that caused a fault. This information is visualized in the *Fault Analyzer* view. It helps to identify and resolve hard-to-find system faults that occur when the CPU is driven into a fault condition by the application software.

Among such conditions are:

- Accessing invalid memory locations
- Accessing memory locations on misaligned boundaries
- Executing undefined instruction
- Division by zero

Upon fault occurrence, the code line where the fault occurred is displayed in the debugger. The view displays the reasons for the error condition. Faults are coarsely categorized into hard, bus, usage and memory faults.

- Hard and bus faults occur when an invalid access attempt is made across the bus, either of a peripheral register or a memory location
- Usage faults are the result of illegal instructions or other program errors
- Memory faults include attempts of access to an illegal location or violations of rules maintained by the memory protection unit (MPU)

To further assist fault analysis, an exception stack frame visualization option provides a snapshot of the MCU register values at the time of the crash. Isolating the fault to an individual instruction allows to reconstruct the MCU condition at the time the faulty instruction was executed.

In the *Debugger* perspective, the *Fault Analyzer* view is opened from the menu. Select the menu command [Window]>[Show View]>[Fault Analyzer] or use the [Quick Access] field, search for "Fault Analyzer" and select it from the views.

#### Figure 227. Open the Fault Analyzer view

IDE workspace\_um1 - NUCLEO-F401RE/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

	5	,						
📬 🗕 🖬 🖨	🗟 : 🕹   🔌 🖡 🕪 💷 🔳 💦 🌫 👁 📭 ji	New Window		• •	🍠 🖗	- 🖓 🕶 🏷	1	
🎋 Debug	🛛 🔁 Project Explorer	Editor	>	~	- 8	☑ main.c 🛛	startup_stm3	c
🗸 🔤 NUC	LEO-F401RE Debug (ST-LINK) [	Show View		Â	 S\\/\/	218	1=0:	>
🗸 🕃 N	UCLEO-F401RE.elf [cores: 0]	Show view	~	•••	Brook	points		
× 🖬	• Thread #1 [main] 1 [core: 0] (	Perspective	>		Duild	Application	AIL+SIIIIL+Q, I	
	HardFault_Handler() at str	Navigation	>	010	Game	Analyzer		_
	≡ <signal called="" handler="">() a</signal>	Preferences		<u> </u>	Debu	ole T	AIL+Shill+Q, C	-
	≡ writeSpeed() at main.c:147	0x800055c		₩ ₩	Debu	y ggar Cancola		
	= main() at main.c:232 0x800	0684		96	Disco	gger Console		
	■ Reset Handler() at startup	stm32f401retx.s:11	3 0x	0	Disas	lag		
J C	/ST/STM32CubeIDE 1.1.0/STM3	32CubeIDE/plugins	/con	•	Error	LOG	AIL+Shiil+Q,	
ST	-LINK (ST-LINK GDB server)			e c	Execu			
				o <sub>x</sub> :	Expre	ssions		_
					Fault	Analyzer		
				ovg.	Live E	xpressions		
				0	Mem	ory		5

## 7.2 Using the Fault Analyzer view

The *Fault Analyzer* view has five main sections, which can be expanded and collapsed. The sections contain different kinds of information for better understanding the reason that caused a particular fault to occur. The sections are:

- Hard Fault Details
- Bus Fault Details
- Usage Fault Details
- Memory Management Fault Details
- Register Content During Fault Exception

When a fault has occurred, it is possible to [**Open editor on fault location**] and [**Open disassembly on fault location**] by pressing the buttons in the view.

Figure 228 shows an example of the *Fault Analyzer* view when an error is detected. In this example, the error is caused by a project making a divide by zero with the debugger stopped in the <code>HardFault\_Handler()</code>.

Opening the *Fault Analyzer* view when this happens displays the reason of the error. In the example, it displays [Usage Fault Detected] and [Attempt to perform a division by zero (DIVBYZERO)]. The *Register Content During Fault Exception* presents register values when the problem occurred.

## Figure 228. Fault Analyzer view

Fault Analyzer 13       Image: Statu Detected       Image: Statu Details       Image: Statu Details          Busis, memory management or usage fault (FORCED) <ul> <li>Failed vector facts (VECTB)</li> <li>Debug event (OEBUEV)</li> <li>Busis Analyzer (STRER)</li> <li>Instruction access violation (RECESER)</li> <li>Unstacking error (UNSTRER)</li> <li>Stacking error (CRTSRR)</li> <li>Floating point lazy state preservation error (LSPER)</li> <li>Interpret deta access violation (NACCENTR)</li> <li>Attempt to execute an undefined instruction (NOCP)</li> <li>Attempt to execute an exponse with Notation (ACCVICI)</li> <li>Unstacking error (MISTRER)</li> <li>Stacking error (MISTRER)</li> <li>Stacking error (MISTRER)</li> <li>Stacki</li></ul>	IDE		-	- 🗆	2	X
* Hard Fault Detected Image Record   * Bus, memory management ousge fault (FORCED) > Pailed vector fricth (VECTBL)   > Pailed vector fricth (VECTBL) > Debug event (DEBUGEVT)   Use Job content on the State Process veloation (REVSERR) > Instruction access violation (REVSERR) > Process data access violation (REVSERR) > Process data access violation (REVSERR) > Process data access violation (REVSERR) > Instruction access violation (REVSERR) > Process data access violation (REVSERR) > Process data access violation (REVSERR) > Instruction access violation (REVSERR) > Process data access violation (REVSERR) > Attempt to execute a undefined instruction (UNDEFINSTR) > Attempt to execute a undefined access (UNALINED) > Attempt to execute a undefined access (UNALINED) > Attempt to perform an undifined access (UNALINED) > Attempt to perform an undifined access (UNALINED) > Data access violation (REVCVIO) > Data access violation (REVCVIO) > Data access violation (REVSERR) Mem manage address register (MIXFR) Poating point lay state preservation enter (MLSPERR) Mem Value Mark > Address register (MIXFR) > Roading point lay state preservation enter (MLSPERR) Mem value > Address register (MIXFR) > Address register (MIXFR	🛙 Fault Ana	alyzer 🛛				
Hard Fault Details    Bus, memory management or usage fault (FORCED)    Bus fault Details  Instruction access violation (BBUSERR)  Instruction access violation (MPRECISERR)  Attempt to execute a undefined instruction (UNDERINSTR)  Attempt to execute a undefined access (UNLINER)  Attempt to execute a corpocision instruction (UNDERINSTR)  Attempt to execute a corpocision instruction (UNDERINSTR)  Attempt to execute a corpocision instruction (NOCP)  Attempt to perform an unaligned access (UNLIANDED)  Attempt to perform an unaligned access	1 Hard Fa	ault Detected		🖻 🔤 PC	$\sim$	l
<ul> <li>Bis, semony management or usage fault (FORCED)</li> <li>Field vector field (VECTBL)</li> <li>Debug event (DEBUGET)</li> </ul> Anternation access violation (REUSERR) Instruction access violation (RECISERR) Each lard access violation (RUSERR) Attempt to execute a concorecess instruction (NUDEFINSTR) Attempt to execute a concorecess instruction (RUSE) Attempt to execute a concorecess instruction (RUSE) Attempt to execute a concorecess instruction (RUSE) Attempt to perform an unaligned access (VALIGNED) Total access violation (RACCVIOL) Data access violation (RACCVIOL) Attempt to perform a livision by zero (RUSERER) Each access register (MMAR): Buse Bisdue access (RUSER (RUSER) Bisdue access register (RUMAR): Buse Ruma manage address register (RUMAR): Buse Ruma violation by zero (RUSERER) Bisdue access register (RUMAR): Buse Ruma violation by zero (RUSERER) Bisdue access register (RUMAR): Buse Ruma violation by zero (RUSERER) Bisdue access register (RUMAR): Buse Ruma vi	Hard Fault	Details				
Pialed vector (tech (VECTEL))   Debug verent (DEBUGEVT)   Bus Fault Details   Instruction access violation (IMPECISERR)   Unstacking error (UNSTKERR)   Instruction access violation error (ISPERR)   Unstacking error (UNSTKERR)   Stacking error (UNSTKERR)   Attempt to execute an undefined instruction (UNDEFINSTR)   Attempt to execute an undefined anstruction (VNCP)   Attempt to execute an undefined access (UNAURINEDE)   Ottomery Management Fault Details   Instruction access violation (ACCVIOL)   Unstracking error (MUNSTKERR)   Stacking error (MUNSTKERR)   Stacking error (MUNSTKERR)   Stacking error (MUNSTKERR)   Stacking error (MUNSTKERR)   Mare manage address register (MMFAR): Dece   Instruction access violation (ACCVIOL)   Unstruction access violation (ACCVIOL)   Stacking error (MUNSTKERR)   Stacking error (MUNSTKERR)   Dece	🔕 Bus, m	nemory management or usage fa	ult (FORCED)			
Debug event (DEBUGEVT)  Sus Fail Details  Instruction access violation (IMPECISERR) Instruction access violation (IMPECISERR) Impecise data access violation (IMPECISERR) Stacking error (INSTRIR) Stacking error (INSTRIR) Stacking error (INSTRIR) Attempt to execute an undefined instruction (UNDEFINSTR) Attempt to execute an undefined instruction (UNDEFINSTR) Attempt to execute an undefined instruction (INOPEC) Attempt to execute an undefined access (UNALIGNED) Instruction access violation (ACCVIOL) Data access violation (ACCVIOL) Data access violation (ACCVIOL) Data access violation (ACCVIOL) Attempt to person ad violation (EXCVIOL) Data access violation (ACCVIOL) Data access violation (ACCVIOL) Attempt age address register (MMFAR) Stacking error (MISTERRR) Stacking error (MISTERRR) Attempt to be accute accepton with the solute on access violation (ACCVIOL) Data access violation (A	🛽 Failed	vector fetch (VECTBL)				
Bus Fault Details  Instruction access violation (RUSERR)  Forecise data access violation (RUSERR)  Frecise data access violation (RUPRECISERR)  Floating point lazy state preservation error (LSPERR)  Floating point lazy state preservation error (LSPERR)  Attempt to execute an undefined instruction (UNDEFINSTR)  Attempt to perform a division by zero (DIVBYZERO)  Memory Management Fault Details  Instruction access violation (ACCVIOL)  Data access violation (ACCVIOL)  Ustacking error (MSTRERR)  Stacking error (MSTRERR)  Floating point lazy state preservation error (MLSPERR)  Mem manage address register (MMFAR): Dixed  Register Content During Fault Exception  Name Value  Register Content During Fault Exception  Register Content During Fault Exception  Register Content During Fault Excep	🛽 Debug	g event (DEBUGEVT)				
<ul> <li>Instruction access violation (IRPECISERR)</li> <li>Imprecise data access violation (IMPECISERR)</li> <li>Unstacking error (UNSTKERR)</li> <li>Stacking error (TIKERR)</li> <li>Floating point lazy state preservation error (LSPERR)</li> <li>Attempt to exception with bod value in EKC. RETURN number (INVPC)</li> <li>Attempt to exception with bod value in EKC. RETURN number (INVPC)</li> <li>Attempt to exception with bod value in EKC. RETURN number (INVPC)</li> <li>Attempt to exception with bod value in EKC. RETURN number (INVPC)</li> <li>Attempt to exception with bod value in EKC. RETURN number (INVPC)</li> <li>Attempt to exception with bod value in EKC. RETURN number (INVPC)</li> <li>Attempt to perform an unaligned access (UNALIGNED)</li> <li>Attempt to perform an unaligned access (UNALIGNED)</li> <li>Attempt to perform an unaligned access (UNALIGNED)</li> <li>Data access violation (IACCVIOL)</li> <li>Data access violation error (IMLSPERR)</li> <li>Braching point lazy state preservation error (IMLSPERR)</li> <li>Mem manage address register (MMFAR): [Dxc0]</li> <li>Register Content Uming Fall Exception</li> <li>Matempt to excelo accession with the fault occurred. Please verify that this value points to a valid stack memory region.</li> <li>MSP = Main Stack Pointer</li> </ul>	Bus Fault [	Details				
Precise data access violation (PRECISERR) Imprecise data access violation (IMPRECISERR) Imprecise data access violation (IMPRECISERR) Imprecise data access violation (IMPRECISERR) Isus fault acdress register (BFAR): Toxe0 Imprecise data access violation (IMPRECISERR) Attempt to execute an undefined instruction (UNDEFINSTR) Attempt to perform an unaligned access (UNALIGNED) Data access violation (IACCVIOL) Data access violation (IACCVIOL	🛛 Instru	ction access violation (IBUSERR)				
Inprecise data access violation (MPRECISERR) Unstacking error (TSTERR) Floating point lay state preservation error (LSPERR) Bus fault address register (GFAR): oxee Use fault betaits Attempt to execute an undefined instruction (UNDEFINSTR) Attempt to execute an undefined instruction (UNDEFINSTR) Attempt to do exception with bad value in EXEC_RETURN number (INVPC) Attempt to evecute an undefined access (UNALGNED) Attempt to perform a undivision by zero (DIVBYZERO) Unstacking error (MNSTRRR) Instruction access violation (ACCVOL) Data access violation (ACCVOL) Instruction access violation (ACCVOL) Instruction access violation (ACCVOL) Instruction access violation (ACCVOL) Instruction grant Exercited PRR) Instruction access violation (MLSPERR) Instruction access violation (ACCVOL) Instruction access violation (MLSPERR) Instruction access violation (ACCVOL) Instruction access violation (MLSPERR) Instruction access violation (MLSPERR) Instruction access violation (MLSPERR) Instruction access violation (MLSPERR) Instruction access violation (ACCVOL) Instruction access violation (ACCVOL) Instruction access violation (ACCVOL) Instruction access violation (ACCVOL) Instruction access violation (MLSPERR) Instruction access violation (ACCVOL) Instructin access violation (ACCVOL) Instruction access violation (ACC	🛽 Precis	e data access violation (PRECISE	(R)			
	🙆 Impre	ecise data access violation (IMPRI	CISERR)			
Stacking error (STKERR) Filoating point lazy state preservation error (LSPERR) Bus fault address register (BFAR): 0xe0 Usage Fault Details • Attempt to execute an undefined instruction (UNDEFINSTR) • Attempt to oxecute an undefined instruction (UNDEFINSTR) • Attempt to do exception with bad value in EXEC, RETURN number (INVPC) • Attempt to do exception with bad value in EXEC, RETURN number (INVPC) • Attempt to do exception with bad value in EXEC, RETURN number (INVPC) • Attempt to perform an unaligned access (UNALIGNED) • Attempt to perform a division by zero (DIVBYZERO) Wemory Management Fault Details • Instruction access violation (IACCVIOL) • Data access violation (IACCVIOL) • Unstacking error (MUNSTKERR) • Floating point lazy state preservation error (MLSPERR) Mem manage address register (MMFAR): 0xe0 Register Content During Fault Exception Time value Time value • Value • Value • Oxe000000 • Interving Fault Exception • Costo0000 • Costo0	🙆 Unsta	cking error (UNSTKERR)				
P Floating point lazy state preservation error (LSPERR) Bus fault address register (BFAR): [bxe0] Utage Fault Details  Attempt to execute a undefined instruction (UNDEFINSTR) Attempt to avoitch to invalid state (INVSTATE) Attempt to oxeception with bad value in EXEC_RETURN number (INVPC) Attempt to do exception with bad value in EXEC_RETURN number (INVPC) Attempt to perform an unaligned acces (UNALIGNED) Data access violation (DACCVIOL) Data access violation violated Exception Marm manage address register (MMFAR): Dxe0 Data access violation (DACCVIOL) Data access violation violate Exception Data access violation violate the fault occurred. Please verify that this value points to a valid stack memory region. DASP = Main Stack Pointer Data Back Point	🙆 Stacki	ing error (STKERR)				
Bus fault address register (BFAR): 0xe0 Utger Fault Details Attempt to execute an undefined instruction (UNDEFINSTR) Attempt to switch to invalid state (INVSTATE) Attempt to execute a coprocessor instruction (NOCP) Attempt to execute a coprocessor instruction (NOCP) Attempt to evecute a coprocessor instruction (NOCP) Attempt to perform an division by zero (DIVBYZERO) Memory Management Fault Details Instruction access violation (IACCVIOL) Data access violation (IACCVIOL) Data access violation (IACCVIOL) Data access violation (IACCVIOL) Data access violation (IACCVIOL) Stacking error (MISTKERR) Stacking error (MISTKERR) Stacking error (MISTKERR) Stack perior During Fault Exception Mame Value Tags (ML, 0xc001760 Tags (ML, 0xc001760 Tags (ML) Tags (M	🛽 Floati	ng point lazy state preservation e	rror (LSPERR)			
Usage Fault Details         Attempt to execute an undefined instruction (UNDEFINSTR)         Attempt to switch to invalid state (INVSTATE)         Attempt to do exception with bad value in EXEC_RETURN number (INVPC)         Attempt to perform a unaligned access (UNALIGNED)         Attempt to perform a division by zero (DIVBYZERO)         Wemory Management Fault Details         Instruction access violation (IACCVIOL)         Data access violation (IACCVIOL)         Unstacking error (MUNSTKERR)         E floating point fazy state preservation error (MLSPERR)         Mem manage address register (IMMFAR): Dxe0         Register Content During Fault Exception         Marg         Marg         Marg         Value         Marg         Value         Image Munosco000         Image Munosco001         Marg         Value         Image Munosco001         Value         Image Munosco001         Image Munosco001         Value	Bus fault	address register (BFAR): 0xe0				
Attempt to execute an undefined instruction (UNDEFINSTR)     Attempt to switch to invalid state (INVSTATE)     Attempt to de execute a coprocessor instruction (NOCP)     Attempt to perform an unaligned access (UNALIGNED)     Attempt to perform a division by zero (DIVBYZERO)      Attempt to perform a division by zero (DIVBYZERO)      Attempt to perform a division (ACCVIOL)     Data access violation (MISTKERR)     Floating point lazy state preservation error (MLSPERR)     Horm manage address register (MMFAR): [0xe0  Register Content During Fault Exception     Marm Value     To 0x0     Data access of 000     Data access of 000     Data access of 000     Data access of 000     Data access     Attempt to 0 = 000     Data access     To 0 = 000     Data access     D	Usage Fau	It Details				
Attempt to switch to invalid state (INVSTATE) Attempt to exception with bad value in EXEC_RETURN number (INVPC) Attempt to exception with bad value in EXEC_RETURN number (INVPC) Attempt to perform an unigned access (INALIGNED) Attempt to perform and unigned access (INALIGNED) Attempt to perform and division by zero (DIVBYZERO) Memory Management Fault Details Instruction access violation (IACCVIOL) Data access violation (IACCVIOL) Unstacking error (MUNSTKERR) Stacking error (MUNSTKERR) Floating point lazy state preservation error (MLSPERR). Mem manage address register (MMFAR): [0xe0 Register Content During Fault Exception Name Value Window 0x00000 Window 0x00000 Window 0x000000 Window 0x0000000 Window 0x0000000 Window 0x0000000 Window 0x0000000 Window 0x00000000 Window 0x000000000 Window 0x000000000 Window 0x000000	Attem	pt to execute an undefined instru	iction (UNDEFINSTR)			
Attempt to do exception with bad value in EXEC_RETURN number (INVPC) Attempt to execute a coprocessor instruction (NOCP) Attempt to perform an unaligned access (UNALIGNED) Attempt to perform a division by zero (DIVBYZERO) Wemory Management Fault Details Instruction access violation (ACCVIOL) Data access violation (ACCVIOL) Unstacking error (MUNSTKERR) Stacking error (MUNSTKERR) Boting point lazy state preservation error (MLSPERR) Mem manage address register (MMFAR): 0xe0 Register Content During Fault Exception Name Value	Attem	pt to switch to invalid state (INV)	ТАТЕ)			
Attempt to execute a coprocessor instruction (NOCP) Attempt to execute a coprocessor instruction (NOCP) Attempt to perform an unaligned access (UNALIGNED) Attempt to perform a division by zero (DIVBYZERO)  Memory Management Fault Details Instruction access violation (ACCVIOL) Data access violation (ACCVIOL) Data access violation (ACCVIOL) Stacking error (MUNSTKERR) Stacking error (MUNSTKERR) Floating point lazy state preservation error (MLSPERR) Mem manage address register (MMFAR): OxeO  Register Content During Fault Exception Mame Value Manage Mulo OxeO0 Management Fault During Fault Exception Mame Value Manage address register (MMFAR): OxeO  Register Content During Fault Exception Mame Value Manage Mulo OxeO0 Management Fault During Fault Exception Mame Value Manage Mulo OxeO0 Management Fault Exception Mame Value Manage Mulo OxeO0 Management Fault Exception Marge OxeO0 Management Fault Exception Marge OxeO0 Management Fault Exception Marge Value Manage Mulo OxeO0 Management Fault Exception Marge Mulo OxeO0 Management Fault Exception Marge OxeO0 Management Fault Exception Marge Management Fault Exception Marge OxeO0 Management Fault Exception Marge OxeO0 Management Fault Exception Manage	Attem	nt to do exception with bad valu	e in EXEC RETURN number (INVPC)			
Attempt to perform an unaligned access (UNALIGNED)         Attempt to perform a division by zero (DIVBYZERO)         Memory Management Fault Details         Instruction access violation (IACCVIOL)         Data access violation (DACCVIOL)         Unatacking error (MUNSTKERR)         Stacking error (MUNSTKERR)         Stacking error (MUNSTKERR)         Stacking error (MUNSTKERR)         Register Content During Fault Exception         Name       Value         ##70       0x0         ##71       0x0         ##72       0x0         ##73       0x8000000         ##74       0x8000000         ##75       0x8000000         ##74       0x8000000         ##75       0x8000000         ##75       0x8000000         ##75       0x8000000         ##76       0x8000000         ##77       0x8000000         ##78       0x8000000         ##79       0x21000000         The value of the stack pointer when the fault occurred. Please verify that this value points to a valid stack memory region.         MSP = Main Stack Pointer       PPP = Proceess Stack Pointer	Attem	nt to execute a conrocessor inst	uction (NOCP)			
A Attempt to perform a division by zero (DIVBYZERO)  Memory Management Fault Details  Instruction access violation (IACCVIOL) Data access violation (DACCVIOL) Data access violation (DACCVIOL) Stacking error (MUNSTKERR) Stacking error (MUNSTKERR) Floating point lazy state preservation error (MLSPERR) Mem manage address register (MMFAR): 0xe0  Register Content During Fault Exception Name Value Stacking error (MUSTRER) Name	Attem	pt to perform an upaligned acce				
Memory Management Fault Details         Instruction access violation (IACCVIOL)         Data access violation (DACCVIOL)         Unstacking error (MUNSTKERR)         Istaking error (MUNSTKERR)         Floating point lazy state preservation error (MLSPERR)         Mem manage address register (MMFAR): 0xe0         Register Content During Fault Exception         Name       Value         Imro       0x0         Imr1       0x0         Imr2       0x0         Imr2       0x800000         Imr12       0x800000         Imr2       0x800000         Imr3       0x800000         Imr4       0x8000000         Imr5       0x8200000         Imr4       0x8000000         Imr5       0x8200000         Imr4       0x8000000         Imr5       0x8200000		pt to perform a division by zero				
Stacking error (MSTKERR) Floating point lazy state preservation error (MLSPERR) Mem manage address register (MMFAR): Dxe0 Register Content During Fault Exception Name Value Imr0 0x0 Imr1 0x0 Imr2 0x0 Imr2 0x0 Imr3 0x800000 Imr12 0x0 Imr2 0x8000685 Imr2 Imr2 0x8000685 Imr2 Imr2 0x20000 Imr2 0x20000 Imr2 Imr2 0x8000685 Imr2 Imr2 0x20000 Imr2 Imr2 0x20000 Imr2 Imr2 0x8000685 Imr2 Imr2 0x20000 Imr2 Imr2 0x20000 Imr2 Imr2 0x8000685 Imr2 Imr2 0x20000 Imr2 Imr2 0x8000685 Imr2 Imr2 0x20000 Imr2 Imr2 0x8000685 Imr2 Imr2 0x8000685 Imr2 Imr2 0x8000685 Imr2 Imr2 0x8000685 Imr2 Imr2 0x800000 Imr2 0x8000685 Imr2 Imr2 0x800000 Imr2 0x8000685 Imr2 Imr2 0x800000 Imr2	<ul> <li>Instruction</li> <li>Data a</li> <li>Unstact</li> </ul>	ction access violation (IACCVIOL) Iccess violation (DACCVIOL) cking error (MUNSTKERR)				
Floating point lazy state preservation error (MLSPERR)   Mem manage address register (MMFAR): 0xe0     Register Content During Fault Exception   Name Value   Way 0x20017fc0   Wir1 0x0   Wir2 0x0   Wir3 0x800000   Wir12 0x0   Wir12 0x0   Wir12 0x8000005c   Wir20 0x800005c   Wir21 0x800005c   Wir20 0x800005c   Wir21 0x21000000   The value of the stack pointer when the fault occurred. Please verify that this value points to a valid stack memory region.	🔕 Stackii	ng error (MSTKERR)				
Mem manage address register (MMFAR): 0xe0         Register Content During Fault Exception         Name       Value         Imsp (M 0x20017fc0         Imr1       0x0         Imr2       0x0         Imr3       0x800000         Imr12       0x0         Imr20       0x0         Imr3       0x800000         Imr4       0x800000         Imr5       0x8000000         Imr4       0x8000000         Imr5       0x8000000         Imr5       0x8000000         Imr5       0x8000000         Imr5       0x8000000         Imr5       0x8000000         Imr5       0x8000000         Imr6       0x8000000	Icontine Continue	ng point lazy state preservation e	rror (MLSPERR)			
Register Content During Fault Exception Name Value  Sep (M 0x20017fc0  Sep (M 0x20017fc0  Sep (M 0x20017fc0  Sep (M 0x0 Sep (M	Mem ma	nage address register (MMFAR):	0xe0			
Name Value   Wirsp (M 0x20017fc0   Wir 0 0x0   Wir 1 0x0   Wir 2 0x0   Wir 3 0x800000   Wir 12 0x0   Wir 12 0x0   Wir 10 0x8000685   Wir 11 0x800055c   Wir 20 0x21000000	Register C	ontent During Fault Exception				
misp (M 0x2017fc0   mir 0 0x0   mir 1 0x0   mir 2 0x0   mir 3 0x800000   mir 12 0x0   mir 0 0x8000685   mir 0 0x800055c   mir 0 0x2100000	Name	Value				
Impo 0x0   Impo 0x0   Impo 0x0   Impo 0x0   Impo 0x800000   Impo 0x8000685   Impo 0x800055c   Impo 0x800055c   Impo 0x21000000	188 (M	0x20017fc0				
Implement 0x0   Implement 0x0   Implement 0x800000   Implement 0x0   Implement 0x0   Implement 0x8000685   Implement 0x800055c   Implement 0x21000000   The value of the stack pointer when the fault occurred. Please verify that this value points to a valid stack memory region.    MSP = Main Stack Pointer PSP = Process Stack Pointer	##r0	0x0				
Imr2 0x0   Imr3 0x800000   Imr12 0x0   Imr17 0x8000685   Imrpc 0x800055c   Imrxpsr 0x2100000   The value of the stack pointer when the fault occurred. Please verify that this value points to a valid stack memory region.    MSP = Main Stack Pointer PSP = Process Stack Pointer	lilir1	0x0				
##r3       0x800000         ##r12       0x0         ##lr       0x8000685         ##pc       0x80005c         ##xpsr       0x2100000	###r2	0x0				
Imple       0x0         Imple       0x8000685         Imple       0x800055c         Imple       0x21000000    The value of the stack pointer when the fault occurred. Please verify that this value points to a valid stack memory region.          MSP = Main Stack Pointer         PSP = Process Stack Pointer	iiiir3	0x8000000				
Impc       0x8000685         Impc       0x800055c         Impxpsr       0x21000000         The value of the stack pointer when the fault occurred. Please verify that this value points to a valid stack memory region.         MSP = Main Stack Pointer         PSP = Process Stack Pointer	🔤r12	0x0				
Impc       0x800055c         Impc       0x2100000         The value of the stack pointer when the fault occurred. Please verify that this value points to a valid stack memory region.         MSP = Main Stack Pointer         PSP = Process Stack Pointer	iiii r	0x8000685				
Image: Construction of the stack pointer when the fault occurred. Please verify that this value points to a valid stack memory region.         MSP = Main Stack Pointer         PSP = Process Stack Pointer	lill pc	0x800055c				
The value of the stack pointer when the fault occurred. Please verify that this value points to a valid stack memory region. MSP = Main Stack Pointer PSP = Process Stack Pointer	xpsr	0x21000000				
MSP = Main Stack Pointer PSP = Process Stack Pointer	The value	of the stack pointer when the fa	ult occurred. Please verify that this value points to a valid stack memory region.		$\wedge$	
MSP = Main Stack Pointer PSP = Process Stack Pointer						
PSP = Process Stack Pointer	MSP = M	ain Stack Pointer				
	PSP = Prc	ocess Stack Pointer				

The Fault Analyzer view contains these toolbar buttons:

## Figure 229. Fault Analyzer toolbar



- The first toolbar button (left) opens the *Editor* on the fault location return address by using the information in the PC and LR registers in the stack and the symbol information in the debugged elf file.
- The second toolbar button (middle) opens the *Disassembly* view on the fault location return address by using the information in PC and LR registers in the stack and the symbol information in the debugged elf file.
- The third toolbar button (right) selects if the PC or LR register is used when opening the *Editor* or *Disassembly* view on error location.

Figure 230 and Figure 231 show the *Editor* and *Disassembly* views opened using the toolbar buttons to find the fault location in the example.

🗹 mai	n.c 🛛	🗟 startup_stm3	system_stm3	🗟 STM32F401RET
142				
143	int	writeSpeed(int	pos)	
144	{			
145				
146		<pre>// update spee</pre>	d	
147		<pre>speed= pos/tse</pre>	c;	
148		<pre>return speed;</pre>		
149				
150	}			

#### Figure 230. Fault analyzer open editor on fault



(×)= Variables	• Breakpoints	🛋 Modules	📟 Disassem	bly 🛛	1889 Registers	s 📟 SFRs 🕸 Live Expressions
					Fn	iter location hei 🗸 🔊 🟠 🔽 📑 t
0800055c	: sdiv	r2, r1,	r2			
08000560	: ldr	r1, [pc,	,#28] ;	(Øx	8000580	<writespeed+56>)</writespeed+56>
08000562	: ldr	r1, [r3,	, r1]			
08000564	: str	r2, [r1,	, #0]			
148	retur	n speed;				

Note: The Fault Analyzer can be used on all STM32 projects. It requires no special code and no special build configuration. All data are collected for the Cortex<sup>®</sup>-M registers. The symbol information is read from the debugged elf file.

## 8 Build Analyzer

## 8.1 Introduction to the Build Analyzer

The STM32CubeIDE Build Analyzer feature interprets program information from the elf file in detail and presents the information in a view. If a map file, with similar name, is found in the same folder as the elf file the information from the map file is also used and even more information can be presented.

The *Build Analyzer* view is useful to optimize or simplify a program. The view contains two tabs, the *Memory Regions* and *Memory Details* tabs:

- The Memory Regions tab is populated with data if the elf file contains a corresponding map file. When the map file is available, this tab can be seen as a brief summary of the memory regions with information about the region name, start address and size. The size information also comprises the total size, free and used part of the region, and usage percentage.
- The *Memory Details* tab contains detailed program information based on the elf file. The different section names are presented with address and size information. Each section can be expanded and collapsed. When a section is expanded, functions/data in this section is listed. Each presented function/data contains address and size information.

## 8.2 Using the Build Analyzer

The *Build Analyzer* view is by default open in the *C/C*++perspective. If the view is closed it can be opened from the menu. Select the menu command [**Window**]>[**Show View**]>[**Build Analyzer**] or use the [**Quick Access**] field, search for *"Build Analyzer"* and select it from the views.

When the *Build Analyzer* view is open, select an elf file in the *Project Explorer* view. The *Build Analyzer* view is then updated with the information from this file. When an elf file is selected and a map file, with similar name, is found in the same folder, additional information from the map file is also used by the view.

The *Build Analyzer* view is also updated if a project node in the *Project Explorer* view is selected. In this case the *Build Analyzer* uses the elf file that corresponds to the current active build configuration of the project.

IDE						
🗟 Build Analyzer 🛛						
STM32F4xx-Nucleo.elf - /STM32F4xx-Nu	cleo/Debug - Oct 21, 2019 2:44:29 PM					
Memory Regions	Memory Details					
Region	Start address	End address	Size	Free	Used	Usage (%)
<b>FLASH</b>	0x08000000	0x08080000	524288	518840	5448	1.04%
•••RAM	0x20000000	0x20018000	98304	96604	1700	1.73%

#### Figure 232. Build analyzer

#### 8.2.1 *Memory Regions* tab

The *Memory Regions* tab in the *Build Analyzer* view displays information based on the corresponding map file. If no information is displayed, it means that there is no corresponding map file found. When a map file is found, the region names, start address, end address, total size of region, free size, used size and usage information are presented.

These regions are usually defined in the linker script file (.1d) used when building the program. Update the linker script file if a memory region location or size must be changed.

Note: The Memory Regions tab is empty if the elf file has no corresponding map file.

## Figure 233. Memory Regions tab

IDE					_		×
🗟 Build Analyzer 🛛						2	7
NUCLEO-F401RE.elf - /NUCLEO-F401RE	/Debug - Oct 25, 2019 10:24:05 AM						
Memory Regions	Memory Details						
Region	Start address	End address	Size	Free	Used	Usage (%)	
	0x20000000	0x20018000	96 KB	16.23 KB	79.77 KB	83.09%	
<b>FLASH</b>	0x08000000	0x08040000	256 KB	236.17 KB	19.83 KB	7.75%	
<b>FLASH_ICONS</b>	0x08040000	0x08050000	64 KB	44.47 KB	19.53 KB	<mark>30</mark> .52%	
FLASH_IMAGES	0x08050000	0x08070000	128 KB	10.81 KB	117.19 KB	91.55%	
FLASH_SOUND	0x08070000	0x0807f000	60 KB	7.75 KB	52.25 KB	87.08%	
FLASH_D	0x0807f000	0x0807f800	2 KB	1.99 KB	8 B	0.39%	
■FLASH_V	0x0807f800	0x08080000	2 KB	1.99 KB	12 B	0.59%	

The column information is described in the Table 23.

#### Table 23. Memory Regions tab information

Name	Description
Region	Name of memory region (if a corresponding map file is found).
Start address	The start address of the region, defined in the linker script.
End address	End address of the region.
Size	The total size of memory region.
Free	The free size in the memory region.
Used	The used size in the memory region.
Usage %	The percentage of used size relative to the total memory region size. See Table 24 for the bar icon color information.

The Usage (%) column contains a bar icon corresponding to the percentage value. The bar has different colors depending on the percentage of used memory.

#### Table 24. Memory Regions usage color

Usage color	Description
Green	Less than 75% of memory used.
Yellow	75% to 90% of memory used.
Red	More than 90% of memory used.

## 8.2.2 *Memory Details* tab

The *Memory Details* tab of the *Build Analyzer* view contains information for the elf file. Each section in the *Memory Details* tab can be expanded so that individual functions and data can be seen. The tab presents columns with name, run address, load address, and size information.

## Figure 234. Memory Details tab

IDE					×
Build Analyzer ≅				2	▼ □ □
NUCLEO-F401RE.elf - /NUCLEO-F401RE/Debug - Oct 25, 2019	11:15:52 AM				
Memory Regions Memory [	Details				
Search					
Name	Run address (VMA)	Load address (LMA)	Size		
✓ ■ RAM	0x20000000		96 KB		
> 🗯 .data	0x20000000	0x08004f49	12 B		
> 🛱 .bss	0x2000000c		78.25 KB		
🗉user_heap_stack	0x2001390c		1.5 KB		
> 🎟 FLASH	0x08000000		256 KB		
FLASH_ICONS	0x08040000		64 KB		
✓ ■ FLASH_IMAGES	0x08050000		128 KB		
👻 🗉 .flash_images	0x08050000	0x08050000	117.19 KB		
■ image3	0x08064c08	0x08064c08	34.18 KB		
■ image2	0x080561a8	0x080561a8	58.59 KB		
■ image1	0x08050000	0x08050000	24.41 KB		
✓ ■ FLASH_SOUND	0x08070000		60 KB		
> 📴 .flash_sound	0x08070000	0x08070000	52.25 KB		
✓ ■ FLASH_D	0x0807f000		2 KB		
> ≒.flash_d	0x0807f000		8 B		
✓ ■ FLASH_V	0x0807f800		2 KB		
> 🖼 .flash_v	0x0807f800	0x0807f800	12 B		

The column information is described in Table 25.

## Table 25. Memory Details tab information

Name	Description
Name	Name of memory region, section, function, and data. A green icon is used to mark functions while the blue icon is used for data variables.
Run Address (VMA)	The Virtual Memory Address contains the address used when the program is running.
Load Address (LMA)	The Load Memory Address is the address used for load, for instance for the initialization values of global variables.
Size	Used size (total size for Memory Regions).

Note:

The memory region name is only displayed if a corresponding map file is found.

#### 8.2.2.1 Size information

The size information in the Memory Details tab is calculated from the symbol size in the elf file. If a corresponding map file is investigated, it may contain a different size value. The size is usually correct for C files but the value presented for assembler files depends on how the size information is written in the assembler files. The constants used by the function must be defined within the section definition. At the end of the section, the size directive is used by the linker to calculate the size of the function.



#### Example: Reset\_Handler in startup.s file

This example shows how to write the <code>Reset\_Handler</code> in an assembler startup file to include the constants <code>\_sidata,\_sdata,\_edata,\_sbss</code>, and <code>\_ebss</code> in the <code>Reset\_Handler</code> size information in the elf file. If these constants are defined out of the <code>Reset\_Handler</code> section definition, their sizes are not included in the calculated size of the <code>Reset\_Handler</code>. To include them in the size of the <code>Reset\_Handler</code>, these definitions must be placed inside the <code>Reset\_Handler</code> section as presented in the code example below.

```
.section .text.Reset Handler
.weak Reset Handler
.type Reset_Handler, %function
Reset Handler:
ldr sp, = estack /* set stack pointer */
/* Copy the data segment initializers from flash to SRAM */
movs r1, #0
b LoopCopyDataInit
CopyDataInit:
ldr r3, =_sidata
/* initialization code data, bss, ... */
. . .
/* Call the application's entry point */
bl main
bx lr
/* start address for the initialization values defined in linker script */
.word _sidata
.word _sdata
.word _edata
.word _sbss
.word _ebss
.size Reset_Handler, .-Reset_Handler
```

#### 8.2.2.2 Sorting

The sort order of a *Memory Details* tab column can be changed by clicking on the column name.

IDE				_		$\times$
🗟 Build Analyzer 🛛					Ľ	<b>v</b> - a
NUCLEO-F401RE.elf - /NUCLEO-F401RE/Debug - Oct 25, 2019	11:15:52 AM					
Memory Regions Memory I	Details					
Search						
Name	Run address (VMA)	Load address (LMA)	Size 🗸			
> 🎟 FLASH	0x08000000		256 KB			
✓ ■ FLASH_IMAGES	0x08050000		128 KB			
👻 🗄 .flash_images	0x08050000	0x08050000	117.19 KB			
image2	0x080561a8	0x080561a8	58.59 KB			
image3	0x08064c08	0x08064c08	34.18 KB			
image1	0x08050000	0x08050000	24.41 KB			
✓ ■ RAM	0x20000000		96 KB			
> 🖼.bss	0x2000000c		78.25 KB			
🗄user_heap_stack	0x2001390c		1.5 KB			
> 🖼.data	0x20000000	0x08004f49	12 B			
> Image: FLASH_ICONS	0x08040000		64 KB			
> I FLASH_SOUND	0x08070000		60 KB			
✓ ■ FLASH_D	0x0807f000		2 KB			
> 🗯 .flash_d	0x0807f000		8 B			
✓ ■ FLASH_V	0x0807f800		2 KB			
> 🗯 .flash_v	0x0807f800	0x0807f800	12 B			

## Figure 235. Memory Details sorted by size

## 8.2.2.3 Search and filter

The information in the *Memory Details*tab can be filtered by entering a string in the search field. Figure 236 shows a search example for names including the string *"sound"*.

## Figure 236. Memory Details search and filter

IDE					×
Build Analyzer ≅				2	▽ □ □
NUCLEO-F401RE.elf - /NUCLEO-F401RE/Debug - Oct 25, 2019 1	1:15:52 AM				
Memory Regions Memory D	Details				
, include , for the second sec					
sound					
Name	Run address (VMA)	Load address (LMA)	Size 🗸		
✓ ■ FLASH_SOUND	0x08070000		60 KB		
✓  ⊑ .flash_sound	0x08070000	0x08070000	52.25 KB		
sound1	0x08070000	0x08070000	19.53 KB		
sound2	0x08074e20	0x08074e20	19.53 KB		
sound4	0x0807afc8	0x0807afc8	8.3 KB		
sound3	0x08079c40	0x08079c40	4.88 KB		

57

## 8.2.2.4 Calculate the sum of sizes

The sum of the sizes of several lines in the *Memory Details* tab can be calculated by selecting these lines in the view. The sum of the selection is presented above the *Name* column in the view.

IDE						$\times$
🗟 Build Analyzer 🛛					2	▼ □ □
NUCLEO-F401RE.elf - /NUCLEO-F401RE	E/Debug - Oct 25, 2019 11:15:52 AM					
Memory Regions	Memory Details					
Selection: 92.77 KB	3					
Search						
Name	Run	address (VMA)	Load address (LMA)	Size ~		
> 📟 FLASH	0x08	8000000		256 KB		
✓ ■ FLASH_IMAG	ES 0x08	8050000		128 KB		
👻 😉 .flash_ima	ges 0x08	8050000	0x08050000	117.19 KB		
image2	2 0x08	80561a8	0x080561a8	58.59 KB		
image3	3 0x08	8064c08	0x08064c08	34.18 KB		
image1	0x08	8050000	0x08050000	24.41 KB		
> 📟 RAM	0x20	000000		96 KB		
> FLASH_ICON	S 0x08	8040000		64 KB		
✓ ■ FLASH_SOUN	0x08	8070000		60 KB		
> 🗯 .flash_sour	nd 0x08	8070000	0x08070000	52.25 KB		
> 📟 FLASH_D	0x08	807f000		2 KB		
> 📟 FLASH_V	0x08	807f800		2 KB		

#### Figure 237. Sum of sizes

## 8.2.2.5 Display the size information in byte format

The *Build Analyzer* view can display size information in different format according to the [**Show Byte**], [**Show Hex**] or [**Show Human**] selection. The icon in the *Build Analyzer* toolbar is used to switch between these formats. Prefer [**Show Byte**] or [**Show Hex**] when copying and pasting of data into an Excel<sup>®</sup> document for later calculations.

Figure	238.	Show	byte	count
--------	------	------	------	-------

IDE				- 0	×
🗟 Build Analyzer 🛛					
NUCLEO-F401RE.elf - /NUCLEO-F401RE	/Debug - Oct 25, 2019 11:15:52 AM				Show Byte
Memory Regions	Memory Details				Show Hex
Selection: 85000	, , , , , , , , , , , , , , , , , , ,				Show Huma
Selection: 05000					
Jearch					
Name	Run address (V	MA) Load address (LMA)	Size		
> 🔤 RAM	0x20000000		98304		
> Image: FLASH_V	0x0807f800		2048		
✓ ■FLASH_SOUN	ID 0x08070000		61440		
> 🗯 .flash_sour	nd 0x08070000	0x08070000	53500		
✓ ■ FLASH_IMAG	ES 0x08050000		131072		
👻 🗉 .flash_imag	ges 0x08050000	0x08050000	120000		
image3	0x08064c08	0x08064c08	35000		
image2	0x080561a8	0x080561a8	60000		
image1	0x08050000	0x08050000	25000		
> == FLASH_ICON	5 0x08040000		65536		
> I FLASH_D	0x0807f000		2048		
> I FLASH	0x08000000		262144		

#### Figure 239. Show hex count

DE					×
Build Analyzer 🛛				2	
UCLEO-F401RE.elf - /NUCLEO-F401RE/Debug - Oct 25, 2	2019 11:15:52 AM				Show Byte
Memory Regions Memor	v Details				Show Hex
Selection: 0v14c09	,				Show Human
Selection. 0x 14c06					
Search V					
Name	Run address (VMA)	Load address (LMA)	Size		
> •••• RAM	0x2000000		0x18000		
> Image: FLASH_V	0x0807f800		0x800		
FLASH_SOUND	0x08070000		0xf000		
> 🗉 .flash_sound	0x08070000	0x08070000	0xd0fc		
FLASH_IMAGES	0x08050000		0x20000		
✓ ≒ .flash_images	0x08050000	0x08050000	0x1d4c0		
image3	0x08064c08	0x08064c08	0x88b8		
image2	0x080561a8	0x080561a8	0xea60		
image1	0x08050000	0x08050000	0x61a8		
> == FLASH_ICONS	0x08040000		0x10000		
> == FLASH_D	0x0807f000		0x800		
	0x0800000		0x40000		

## 8.2.2.6 Copy and paste

The data in the *Memory Details* tab can be copied to other applications in CSV format by selecting the rows to copy and typing **Ctrl+C**. The copied data can be pasted into another application with the **Ctrl+V** command.

## Figure 240. Copy and paste

IDE					×
🗟 Build Analyzer 🛛				2	▼ □ □
NUCLEO-F401RE.elf - /NUCLEO-F401RE/Debug - Oct 25, 2019 11:	15:52 AM				
Memory Regions Memory De	etails				
Selection: 193500					
Search					
Nama	Pup address (V/MA)	Load address (LMA)	Sizo		
		LOAU AUUTESS (LIVIA)	00204		
	0x2000000		98304		
	0x08071800		2048		
✓ ■FLASH_SOUND	0x08070000	0.09070000	61440		
	0x08070000	0x08070000	22200		
= sound4	0x0607aico	0x00070100	6500		
= sound3	0x08079c40	0x08079C40	2000		
- sound2	0x06074e20	0x00074020	20000		
	0x08070000	0x08070000	20000		
FLASH_IMAGES	0x08050000	0.00050000	131072		
<ul> <li>Itash_Images</li> </ul>	0x08050000	0x08050000	120000		
Image3	0x08064c08	0x08064c08	35000		
Image2	0x08056188	0x08056188	60000		
	0x08050000	0x08050000	25000		
FLASH_ICONS	0x08040000	0.00040000	00000		
✓ Istash_icons	0x08040000	0x08040000	20000		
	0x08040000	0x08040000	20000		
> === FLASH_D	UXU8U/1000		2048		
> mm FLASH	0x0800000		262144		

The Ctrl+C copy of the lines selected in Figure 240 provides the Ctrl+V results below:

"sound4";"0x0807afc8";"0x0807afc8";"8500"
"sound3";"0x08079c40";"0x08079c40";"5000"
"sound2";"0x08074e20";"0x08074e20";"20000"
"sound1";"0x08070000";"0x08070000";"20000"
"image3";"0x08064c08";"0x08064c08";"35000"
"image1";"0x080561a8";"0x080561a8";"20000";"25000"
"icons";"0x08040000";"0x08040000";"20000"

## 9 Static Stack Analyzer

## 9.1 Introduction to the *Static Stack Analyzer*

The STM32CubeIDE Static Stack Analyzer calculates the stack usage based on the built program. It analyzes the .su files, generated by gcc, and the elf file in detail, and presents the resulting information in the view.

The view contains two tabs, the *List* and *Call Graph* tabs.

The *List* tab is populated with the stack usage for each function included in the program. The tab lists one line per function, each line consisting of the *Function, Local cost, Type, Location* and *Info* columns.

IDE					×
🚢 Static Stack Analyzer 🛛					2 -
STM32F4xx-Nucleo.elf - /STM32F4xx-Nucleo/Debug - Oct 25, 2019 1	2:03:09 PM				
List Call graph					
Hide dead code					
Function	Local cost	Туре	Location	Info	^
• main	88	STATIC	main.c:79		
TIM_TI1_SetConfig	16	STATIC	stm32f4xx_hal_tim.c:4540		
TIM_SlaveTimer_SetConfig	12	STATIC	stm32f4xx_hal_tim.c:4983		
TIM_CCxChannelCmd	8	STATIC	stm32f4xx_hal_tim.c:4739		
TIM_Base_SetConfig	0	STATIC	stm32f4xx_hal_tim.c:4481		
SystemInit	0	STATIC	system_stm32f4xx.c:148		
HAL_TIM_TriggerCallback	0	STATIC	stm32f4xx_hal_tim.c:4364		
HAL_TIM_SlaveConfigSync	16	STATIC	stm32f4xx_hal_tim.c:4143		
HAL_TIM_ReadCapturedVal	0	STATIC	stm32f4xx_hal_tim.c:4217		
HAL_TIM_PeriodElapsedCal	0	STATIC	stm32f4xx_hal_tim.c:4304		
HAL_TIM_PWM_PulseFinish	0	STATIC	stm32f4xx_hal_tim.c:4349		
HAL_TIM_OC_DelayElapsed	0	STATIC	stm32f4xx_hal_tim.c:4319		
HAL_TIM_IRQHandler	8	STATIC	stm32f4xx_hal_tim.c:2809		
HAL_TIM_IC_Start_IT	8	STATIC	stm32f4xx_hal_tim.c:1672		$\checkmark$
<					>

Figure 241. Static Stack Analyzer List tab


The *Call Graph* tab contains an expandable list with functions included in the program. Lines representing functions calling other functions can be expanded to see the call hierarchy.

IDE								_		×
🚢 Static Stack	Analyzer ≈									<b>2</b>
STM32F4bor-Nucleo.elf - /STM	M32F4xx-Nucleo/Debug - Oct 25, 2019 12:	03:09 PM								
List Ca	ll graph									
main								Search 🗹	Case se	ensitive
Function		Depth	Max čost	Local cost	Туре	Location	Info			
🗘 Usagel	Fault_Handler	?	?	0	STATIC	stm32f4xx_it.c:116	Max cost uncertain. Recursive			
ADC_I	RQHandler	?	?	0			Max cost uncertain. Recursive. No stack usage int	ormation availa	able for	this
🗘 BusFau	ult_Handler	?	?	0	STATIC	stm32f4xx_it.c:103	Max cost uncertain. Recursive			
🗘 HardFa	ault_Handler	?	?	0	STATIC	stm32f4xx_it.c:77	Max cost uncertain. Recursive			
🗘 Mem M	/lanage_Handler	?	?	0	STATIC	stm32f4xx_it.c:90	Max cost uncertain. Recursive			
> 🗟 Reset_	Handler	7	184	0			Max cost uncertain. No stack usage information a	vailable for thi	s functio	on
> • TIM4_I	IRQHandler	3	8	0	STATIC	stm32f4xx_it.c:173	Max cost uncertain			
NMI_H	landler	0	0	0	STATIC	stm32f4xx_it.c:68				
PendS <sup>1</sup>	V_Handler	0	0	0	STATIC	stm32f4xx_it.c:147				
🗟 frame_	_dummy	0	0	0			Max cost uncertain. No stack usage information a	vailable for thi	s functio	on
>  SysTick	k_Handler	1	0	0	STATIC	stm32f4xx_it.c:156				
SVC_H	andler	0	0	0	STATIC	stm32f4xx_it.c:129				
Debug	Mon_Handler	0	0	0	STATIC	stm32f4xx_it.c:138				
🗟do_g	lobal_dtors_aux	0	0	0			Max cost uncertain. No stack usage information a	vailable for thi	s functio	on
🗟 _fini		0	0	0			Max cost uncertain. No stack usage information a	vailable for thi	s functio	on

#### Figure 242. Static Stack Analyzer Call Graph tab

# 9.2 Using the Static Stack Analyzer

The Static Stack Analyzerview is by default open in the C/C++perspective. If the view is closed, it can be opened from the menu. Select the menu command [Window]>[Show View]>[Static Stack Analyzer]. Another way to open the Static Stack Analyzerview is to type "Static Stack Analyzer" in the [Quick Access search bar] and select it from the views.

#### Figure 243. Open the Static Stack Analyzer view

IDE workspace\_um1 - NUCLEO-F401RE/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

📬 🕶 🔚 🗟 👻 🔦 👻 🗟 i 🔌 i 🤹 🖄 i 🕍 💌 🤅	New Window		〃 ▼ : ⊿ № 0 π : ⊵ ▼ 주 ▼ ♥ ↓ ↓ ▼ 🛃 🕄				
🔁 Project Explorer 🛛	Editor	(	🕼 main.c 🛛 🖻 stm32f4xx_it.c 🛛 🗟 STM32F4	10			
V INUCLEO-F401RE	Appearance						
> & Binaries	Show View >	👬 Build Analyzer					
> Mincludes	Perspective	>	Build Targets				
<ul> <li>✓ <sup>™</sup> Core</li> </ul>	Navigation	>	C/C++ Projects				
> 🗁 Inc	Preferences		Console Alt+Snift+Q, C				
✓			<ul> <li>Navigator (Deprecated)</li> </ul>				
> 😼 main.c			E Outline Alt+Shift+Q, O				
> 🗟 stm32f4xx_hal_msp.c			Problem Details				
> is stm32f4xx_it.c			Problems Alt+Shift+Q, X				
> is syscalls.c			🛍 Project Explorer				
> isysconf.c			Properties				
> 🖻 sysmem.c							
> is system_stm32f4xx.c			📟 SFRs				
> 🗁 Startup			🛎 Static Stack Analyzer				
> 😕 Drivers			Zasks				
✓ ➢ Debug			Other Alt+Shift+Q, Q				

The *Static Stack Analyzer* view is populated when a built project is selected in the *Project Explorer*. The project must be built with option [Generate per function stack usage information] enabled, otherwise the view cannot present any stack information.

How to setup the compiler to generate stack usage information is explained in the next section.

## 9.2.1 Enable stack usage information

If the top of the view displays the message No stack usage information found, please enable in the compiler settings, the build configuration must be updated for the compiler to generate stack information:

- 1. Open the project properties, for instance with a right-click on the project in the *Project Explorer* view
- 2. Select Properties and, in the dialog, select [C/C++ Build]>[Settings]
- 3. Select the Tool Settings tab
- 4. Select [MCU GCC Compiler]>[Miscellanous]
- 5. Select [Enable stack usage information (-fstack-usage)] as shown in Figure 244
- 6. Save the setting and rebuild the program

#### Figure 244. Enable generate per function stack usage information

IDE Properties for NUCLEO-F401	RE					X
type filter text	Settings				⇐ ▼ ⇔	<b>•</b> •
<ul> <li>Resource</li> <li>C/C++ Build</li> <li>Build Variables</li> <li>Discovery Options</li> <li>Environment</li> </ul>	Configuration: Debug [Active]	Manage	Manage Configuratio			
Logging Settings > C/C++ General CMSIS-SVD Settings Project References Refactoring History Run/Debug Settings	<ul> <li>Tool Settings</li> <li>MCU Settings</li> <li>MCU Post build outputs</li> <li>MCU GCC Assembler</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>General</li> <li>Debugging</li> <li>Preprocessor</li> <li>Include paths</li> <li>Optimization</li> <li>Warnings</li> <li>Miscellaneous</li> <li>MCU GCC Linker</li> <li>General</li> <li>Libraries</li> <li>Miscellaneous</li> </ul>	Build Artifact     Other flags     Other flags     Verbose (-v)     Position Indepe	Binary Parsers	Error Parsers		·
		Enable stack us	age analysis (-fstacl	(-usage) Restore Defaults	Apply	
0			A	oply and Close	Cancel	

### 9.2.2 *List* tab

The *List* tab contains a list of all functions included in the selected program with options to [Hide dead code] functions and [Filter] visible functions.

Use the [Hide dead code] selection to enable or disable the listing of dead code functions. If used, the [Filter] field restricts the display to functions matching the characters it contains.

#### Figure 245. Static Stack Analyzer List tab

IDE						×
🚢 Static Stack Analyzer 🛙						2 -
NUCLEO-F401RE.elf - /NUC	LEO-F401F	RE/Debu	g - Oct 25, 2019 1:58:05 PM	N		
List Call graph						
Hide dead code						
Function	Local cost	Туре	Location	Info		^
SystemClock_Config	88	STATIC	main.c:410			
HAL_RCC_MCOConfig	56	STATIC	stm32f4xx_hal_rcc.c:749			
MX_GPIO_Init	48	STATIC	main.c:487			
HAL_UART_MspInit	48	STATIC	stm32f4xx_hal_msp.c:88			
HAL_DMA_PollForTransfer	48	STATIC	stm32f4xx_hal_dma.c:612			
• main	40	STATIC	main.c:183			
• NVIC_EncodePriority	40	STATIC	core_cm4.h:1863			
• NVIC_DecodePriority	40	STATIC	core_cm4.h:1890			
HAL_FLASH_Program	40	STATIC	stm32f4xx_hal_flash.c:156			
HAL_GPIO_Init	40	STATIC	stm32f4xx_hal_gpio.c:171			
HAL_RCC_GetSysClockFreq	40	STATIC	stm32f4xx_hal_rcc.c:859			
HAL_UART_Transmit	40	STATIC	stm32f4xx_hal_uart.c:1019			
HAL_UART_Receive	40	STATIC	stm32f4xx_hal_uart.c:1101			
HAL_UART_IRQHandler	40	STATIC	stm32f4xx_hal_uart.c:1998			
e write	20	CTATIC	main c·106			*

The column information in the *List* tab is described in Table 26.

#### Table 26. Static Stack Analyzer List tab details

Name	Description
Function	Function name.
Local cost	The number displays how many bytes of stack the function uses.
Туре	Tells if the function uses a STATIC or DYNAMIC stack allocation. When DYNAMIC allocation is used the actual stack size is run-time dependent and the the <i>Local cost</i> value is uncertain due to the dynamic size of stack.
Location	Indicates where the function is declared. It is possible to double-click on a line and open the file with the defined function in the editor.
Info	Additional information about the calculation.

The List tab sort order can be changed by clicking on a column name.

Note:

By double-clicking on a line that displays the file location and line number in the List tab, the function is opened in the Editor view.

**[**]



The *Call Graph* tab contains detailed program information in a tree view. Each function included in the program but not called by any other function is presented at the top level. It is possible to expand the tree to see called functions. Only functions available in the elf file can be visible in the tab.

When used, the [**Search...**] button triggers the display of the functions matching the characters in the search field. The search can be made case sensitive or not depending on the selection in checkbox [**Case sensitive**].

IDE						- 0	×
🛎 Static Stack Analyzer 🛙							<b>d</b> - D
NUCLEO-F401RE.elf - /NUCLEO-F4	401RE/Debug	J - Oct 25, 20	019 1:58:05	РМ			
List Call graph							
						Search 🗹 Case sen	sitive
Function	Depth	Max čost	Local cost	Туре	Location	Info	^
ADC_IRQHandler	?	?	0			Max cost uncertain. Recursive. No stack usage information available for this	
> BReset_Handler	16	248	0			Max cost uncertain. No stack usage information available for this function	
> 🔜swrite	3	48	0			Max cost uncertain. No stack usage information available for this function	
> 🗟sread	2	32	0			Max cost uncertain. No stack usage information available for this function	
> msseek	2	24	0			Max cost uncertain. No stack usage information available for this function	
> 🗟sclose	2	16	0			Max cost uncertain. No stack usage information available for this function	
> SysTick_Handler	1	12	8	STATIC	stm32f4xx_it.c:182		
• NMI_Handler	0	4	4	STATIC	stm32f4xx_it.c:70		
DebugMon_Handler	0	4	4	STATIC	stm32f4xx_it.c:156		
MemManage_Handler	0	4	4	STATIC	stm32f4xx_it.c:98		
UsageFault_Handler	0	4	4	STATIC	stm32f4xx_it.c:128		
PendSV_Handler	0	4	4	STATIC	stm32f4xx_it.c:169		
BusFault_Handler	0	4	4	STATIC	stm32f4xx_it.c:113		
HardFault_Handler	0	4	4	STATIC	stm32f4xx_it.c:83		
SVC_Handler	0	4	4	STATIC	stm32f4xx_it.c:143		
🗟 frame_dummy	0	0	0			Max cost uncertain. No stack usage information available for this function	$\checkmark$

The column information in the Call Graph tab is described in Table 27.

#### Table 27. Static Stack Analyzer Call Graph tab details

Name	Description
Function	Function name.
Depth	Specifies the call stack depth this function uses:         •       0: the function does not call any other functions         •       Number ≥ 1: the function calls other functions         •       ?: the function makes recursive calls or the depth cannot be calculated
Max cost	Specifies how many bytes of stack the function uses including stack needed for called functions.
Local cost	Specifies how many bytes of stack the function uses. This column does not take into account any stack that may be needed by the functions it may call.
Туре	<ul> <li>Specifies if the function uses a STATIC or DYNAMIC stack allocation.</li> <li>STATIC: the function uses a fixed stack</li> <li>DYNAMIC: the function uses a run-time dependent stack</li> <li>Empty field: no stack usage information available for the function</li> </ul>
Location	Indicates where the function is declared. It is possible to double-click on a line and open the file with the defined function in the editor.
	Contains specific information about the stack usage calculation. For instance, it can hold a combination of the following messages:
	• Max cost uncertain: the reason can be that the function makes a call to some sub-function where the stack information is not known, the function makes recursive calls, or others
Info	Recursive: the function makes recursive calls
	• No stack usage information available for this function: no stack usage information available for this function
	• Local cost uncertain due to dynamic size, verify at run-time: the function allocates stack dynamically, for instance depending on a parameter

The Call Graph tab sort order can be changed by clicking on a column name.

By double-clicking on a line that displays the file location and line number in the tab, the function is opened in the *Editor* view.

Note:

The main function is usually called by the Reset\_Handler and can in those cases be seen when expanding the Reset\_Handler node.

If unused functions are listed in the tab, check if linker option [**dead code removal**] is enabled to remove unused code from the program. Read more on this in Section 2.5.2 Discard unused sections.

The small icon left of the function name in column Function column indicates the following:

- Green dot: the function uses STATIC stack allocation (fixed stack).
- Blue square: the function uses DYNAMIC stack allocation (run-time dependent).
- 010 icon: used if the stack information is not known. This can be the case for library functions or assembler functions.
- Three arrows in a circle: used in the Call Graph tab when the function makes recursive calls.

# Figure 247. Function symbols in *Static Stack Analyzer*

IDE	
≝ Static Stack Analyzer 🛛	
NUCLEO-F401RE.elf - /NUCLEO-F401R	E/Debug
List Call graph	
Function	Depth
🗘 ADC_IRQHandler	?
✓ mathematical Reset_Handler	16
🗸 🗟 LoopCopyDataInit	15
🗸 🗟 LoopFillZerobss	14
🖌 🔍 main	13
SystemClock_Config	5
MX_USART2_UART_Init	4
MX_GPIO_Init	1
SystemCoreClockUpdate	0
> 🗟 iprintf	12
readTemp	0
readSpeed	0
writeSpeed	0
writeTemp	0
SystemInit	0
🗟 FillZerobss	0

### 9.2.4 Using the filter and search field

The *List* and *Call Graph* tabs contain a filter/search field, which can be used to search a specific function or functions matching the characters entered in the field.

Figure 248 displays the *List* tab where the [**Filter**] field is used to seek functions containing the *"read"* string in their name.

IDE						×
🚢 Static Stack Analyzer 🛛						2 - 0
NUCLEO-F401RE.elf - /NUC	CLEO-F401F	RE/Debu	g - Oct 25, 2019 1:	58:05 PM		
List Call graph						
🗹 Hide dead code						
read						
Function	Local cost	Туре	Location	Info		
● _read	32	STATIC	syscalls.c:97			
readSpeed	16	STATIC	main.c:133			
● readTemp	16	STATIC	main.c:146			

## Figure 248. Static Stack Analyzer List tab using search

Figure 249 shows a use example of the [**Search...**] field in the *Call Graph* tab for filtering functions with name matching the *"read"* string.

IDE						— 🗆 X
🛎 Static Stack Analyzer 🔤						2 · · ·
NUCLEO-F401RE.elf - /NUCLEO-F401RE/Debu	g - Oct 28,	2019 1:09:2	9 PM			
List Call graph						
read						Search 🗹 Case sensitive
Function	Depth	Max cost	Local cost	Туре	Location	Info
• _read	0	32	32	STATIC	syscalls.c:97	
> 🔜sclose	2	16	0			Max cost uncertain. No stack usage information available for this function
do_global_dtors_aux	0	0	0			Max cost uncertain. No stack usage information available for this function
• UsageFault_Handler	0	4	4	STATIC	stm32f4xx_it.c:128	
> • SysTick_Handler	1	12	8	STATIC	stm32f4xx_it.c:182	
<ul> <li>SVC_Handler</li> </ul>	0	4	4	STATIC	stm32f4xx_it.c:143	
<ul> <li>Reset_Handler</li> </ul>	16	232	0			Max cost uncertain. No stack usage information available for this function
<ul> <li>LoopCopyDataInit</li> </ul>	15	232	0			Max cost uncertain. No stack usage information available for this function
<ul> <li>LoopFillZerobss</li> </ul>	14	232	0			Max cost uncertain. No stack usage information available for this function
🗸 🔍 main	13	232	32	STATIC	main.c:183	Max cost uncertain
<ul> <li>writeTemp</li> </ul>	0	16	16	STATIC	main.c:164	
writeSpeed	0	16	16	STATIC	main.c:156	
<ul> <li>readTemp</li> </ul>	0	16	16	STATIC	main.c:146	
readSpeed	0	16	16	STATIC	main.c:133	
> 🗟 iprintf	12	24	0			Max cost uncertain. No stack usage information available for this function
SystemCoreClockUpdate	0	32	32	STATIC	system_stm32f4xx.c:239	
<ul> <li>SystemClock_Config</li> </ul>	5	200	88	STATIC	main.c:410	Max cost uncertain
		110	0	CTATIC	1 454	
						/

#### Figure 249. Static Stack Analyzer Call Graph using search

## 9.2.5 Copy and paste

The data in the *List* tab can be copied to other applications in CSV format by selecting the rows to copy and typing **Ctrl+C**. The copied data can be pasted into another application with the **Ctrl+V** command.

rigure 200. Oopy and paste	Figure	250.	Сору	and	paste	
----------------------------	--------	------	------	-----	-------	--

IDE						×
🚢 Static Stack Analyzer 🛙						2 - 0
NUCLEO-F401RE.elf - /NUC	LEO-F401R	E/Debug	g - Oct 25, 2019 1:58:05 PM			
List Call graph						
✓ Hide dead code						
Function	Local cost	Туре	Location	Info		^
SystemClock_Config	88	STATIC	main.c:410			
MX_GPIO_Init	48	STATIC	main.c:487			
HAL_UART_MspInit	48	STATIC	stm32f4xx_hal_msp.c:88			
• main	40	STATIC	main.c:183			
• NVIC_EncodePriority	40	STATIC	core_cm4.h:1863			
HAL_GPIO_Init	40	STATIC	stm32f4xx_hal_gpio.c:171			
HAL_RCC_GetSysClockFreq	40	STATIC	stm32f4xx_hal_rcc.c:859			
• _write	32	STATIC	main.c:106			
• _read	32	STATIC	syscalls.c:97			
• _write	32	STATIC	syscalls.c:109			
SystemCoreClockUpdate	32	STATIC	system_stm32f4xx.c:239			~
						•

The Ctrl+C copy of the lines selected in Figure 250 provides the Ctrl+V results below:

"SystemClock\_Config";"88";"STATIC";"main.c:410";""

```
"main";"40";"STATIC";"main.c:183";""
```

"HAL\_GPIO\_Init";"40";"STATIC";"stm32f4xx\_hal\_gpio.c:171";""



# **10** Installing updates and additional Eclipse<sup>®</sup> plugins

# **10.1** Check for updates

STM32CubeIDE checks for available updates regularly and opens the *Available Updates* dialog when a new update is detected. It is also possible to check for updates manually. Use menu **[Help]>[Check for Updates]** to check if new software is available.

When updates are found, select the update to install and press [Next].

<b>Figure</b>	251.	STM32CubeIDE	available	updates
---------------	------	--------------	-----------	---------

📴 Available Updates		_		×
<b>Available Updates</b> Check the updates that you wish to install.				
Name	Version 1.0.2	ld com.st.stm32cube.ide.mcu.rcp.product		
Select All Deselect All Details				\$
?	< Back N	ext > Finish	Cancel	

### Update details is displayed. Review and confirm the update. Press Next.

Image: Available Updates:       Image: Constant Stand St						
Update Details         Review and confirm the updates.         Name       Version       Id         STM32CubelDE       1.0.2       com.st.stm32cube.ide.feature.ncu         STM32CubelDE Build Analyzer       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu         STM32CubelDE Build Analyzer       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         STM32CubelDE Build Analyzer       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         STM32CubelDE C/C++ Debugging Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         STM32CubelDE C/C++ Embedded Development Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         STM32CubelDE C/C++ Embedded Development Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         STM32CubelDE C/C++ Embedded Development Tools for MCU       1.0.2.20190701052       com.st.stm32cube.ide.feature.com         STM32CubelDE Common N Services       1.0.2.20190701052       com.st.stm32cube.ide.feature.com       v         Strust2cubelDE Common Uli       1.0.2.20190701052       com.st.stm32cube.ide.feature.com       v         Strust2cubelDE Common Uli       1.0.2.20190701052       com.st.stm32cube.ide.feature.com       v         Strust2cubelDE Common Uli       1.0.2.20190701052       com.st.stm32cub	IDE Available Updates			—		$\times$
Review and confirm the updates.       Version       Id         Name       Version       Id         STM32CubeIDE       1.0.2       com.st.stm32cube.ide.mcu.rcp.prod         STM32CubeIDE Build       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu.t         STM32CubeIDE Build       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         STM32CubeIDE Build Analyzer       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         STM32CubeIDE C/C++ Debugging Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         STM32CubeIDE C/C++ Embedded Development Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         STM32CubeIDE Common Mx Services       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu	Update Details					
Name       Version       Id       Image: Comparison of the standard sta	Review and confirm the updates.				0	
<ul> <li></li></ul>	Name	Version		Id		^
Image: STM32CubeIDE ARM Toolchain Definition       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu.t         Image: STM32CubeIDE Build       1.0.2.201907091052       com.st.stm32cube.ide.feature.mcu         Image: STM32CubeIDE Build Analyzer       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         Image: STM32CubeIDE Build UI       1.0.2.201907092029       com.st.stm32cube.ide.feature.mcu         Image: STM32CubeIDE C/C++ Debugging Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         Image: STM32CubeIDE C/C++ Embedded Development Tools for MCU       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu         Image: STM32CubeIDE Common Mx Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com       com.st.stm32cube.ide.feature.com         Image: STM32CubeIDE Common Mx Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com       com.st.stm32cube.ide.feature.com         Image: STM32CubeIDE Common VI       1.0.2.201907091052       com.st.stm32cube.ide.feature.com       Image: STM32CubeIDE Common VI	✓	1.0.2		com.st.stm32cube.ide.mcu.r	cp.prod	
Image: STM32CubelDE Build       1.0.2.201907091052       com.st.stm32cube.ide.feature.mcu         Image: STM32CubelDE Build Analyzer       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         Image: STM32CubelDE Build UI       1.0.2.201907092029       com.st.stm32cube.ide.feature.mcu         Image: STM32CubelDE C/C++ Debugging Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         Image: STM32CubelDE C/C++ Embedded Development Tools for MCU       1.0.2.201907121423       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common Mx Services       1.0.2.20190701052       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common Services       1.0.2.201907120816       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common UI       1.0.2.201907091052       com.st.stm32cube.ide.feature.com	🖗 STM32CubeIDE ARM Toolchain Definition	1.0.2.201907121423		com.st.stm32cube.ide.featu	re.mcu.t	
Image: STM32CubelDE Build Analyzer       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         Image: STM32CubelDE Build UI       1.0.2.201907092029       com.st.stm32cube.ide.feature.mcu         Image: STM32CubelDE C/C++ Debugging Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         Image: STM32CubelDE C/C++ Embedded Development Tools for MCU       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu         Image: STM32CubelDE Common Mx Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common UI       1.0.2.201907091052       com.st.stm32cube.ide.feature       Image: STM32cube.ide.feature         Image: STM32CubelDE Common UI       1.0.2.201907091052       com.st.stm32cube.ide.feature.	🖗 STM32CubelDE Build	1.0.2.201907091052		com.st.stm32cube.ide.featu	re.mcu	
Image: Stm32cubelDE Build UI       1.0.2.201907092029       com.st.stm32cube.ide.feature.mcu         Image: Stm32cubelDE C/C++ Debugging Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         Image: Stm32cubelDE C/C++ Embedded Development Tools for MCU       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu.i         Image: Stm32cubelDE C/C++ Embedded Development Tools for MCU       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu.i         Image: Stm32cubelDE Common Mx Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Image: Stm32cubelDE Common VI       1.0.2.201907091052       com.st.stm32cube.ide.feature       Image: Stm32cube.ide.feature         Image: Stm32cubelDE Common VI       1.0.2.201907091052       com.st.stm32cube.ide.feature       Image: Stm32cube.ide.feature       Image: Stm32cube.ide.feature       Image: Stm32c	🖗 STM32CubelDE Build Analyzer	1.0.2.201907120816		com.st.stm32cube.ide.featu	re.mcu	
Image: STM32CubelDE C/C++ Debugging Tools for MCU       1.0.2.201907120816       com.st.stm32cube.ide.feature.mcu         Image: STM32CubelDE C/C++ Embedded Development Tools for MCU       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu.i         Image: STM32CubelDE Common Mx Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common UI       1.0.2.201907091052       com.st.stm32cube.common.feature         Image: STM32CubelDE Common UI       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common UI       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common UI       Image: STM32CubelDE Common UI       Image: STM32CubelDE Common UI         Imag	> 🖗 STM32CubeIDE Build UI	1.0.2.201907092029		com.st.stm32cube.ide.featu	re.mcu	
Image: STM32CubelDE C/C++ Embedded Development Tools for MCU       1.0.2.201907121423       com.st.stm32cube.ide.feature.mcu.i         Image: STM32CubelDE Common Mx Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Image: STM32CubelDE Common UI       1.0.2.201907091052       com.st.stm32cube.common.feature         Image: STM32CubelDE Common UI       1.0.2.201907091052       com.st.stm32cube.common.feature         Image: STM32CubelDE Common UI       1.0.2.201907091052       com.st.stm32cube.common.feature         Image: STM32CubelDE Common UI       Image: STM32CubelDE Common UI       Image: STM32CubelDE Common UI         Image: STM32CubelDE	🖗 STM32CubeIDE C/C++ Debugging Tools for MCU	1.0.2.201907120816		com.st.stm32cube.ide.featu	re.mcu	
Image: STM32CubelDE Common Mx Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         STM32CubelDE Common Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         STM32CubelDE Common UI       1.0.2.201907120816       com.st.stm32cube.ide.feature.com         STM32CubelDE Common Utilities       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Size: Unknown       1.0.2.201907091052       com.st.stm32cube.ide.feature.com       v	🖗 STM32CubeIDE C/C++ Embedded Development Tools for MCU	1.0.2.201907121423		com.st.stm32cube.ide.featu	re.mcu.i	
Image: STM32CubelDE Common Services       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         STM32CubelDE Common UI       1.0.2.201907120816       com.st.stm32cube.ide.feature.com         STM32CubelDE Common Utilities       1.0.2.201907091052       com.st.stm32cube.ide.feature.com         Size: Unknown       Size: Unknown       com.st.stm32cube.ide.feature.com       v	🖗 STM32CubeIDE Common Mx Services	1.0.2.201907091052		com.st.stm32cube.ide.featu	re.com	
Image: STM32CubelDE Common UI       1.0.2.201907120816       com.st.stm32cube.ide.feature.com         STM32CubelDE Common Utilities       1.0.2.201907091052       com.st.stm32cube.common.feature         Size: Unknown       Details       Image: Stm32CubelDE Common Utilities       Image: Stm32CubelDE Common.feature	🖗 STM32CubeIDE Common Services	1.0.2.201907091052		com.st.stm32cube.ide.featu	re.com	
Image: STM32CubelDE Common Utilities       1.0.2.201907091052       com.st.stm32cube.common.feature       Image: Structure         Size: Unknown       Details       Image: Structure       Image: Structur	🖗 STM32CubeIDE Common UI	1.0.2.201907120816		com.st.stm32cube.ide.featu	re.com	
Size: Unknown Details	STM32CubelDE Common Utilities	1.0.2.201907091052		com.st.stm32cube.common	.feature	~
Details	Size: Unknown					
	Details					
						^
						Ý
< Back Next > Finish Cancel		< Back	Next >	Finish	Cancel	

## Figure 252. STM32CubeIDE update details

*Review Licenses* details are displayed. Review the licenses, select [I accept the terms of the license agreements] and press [Finish] to install the update.

## Figure 253. STM32CubeIDE update review licenses

eview Licenses Licenses must be reviewed before the software can be installed. This includes lic	licenses for software required to complete the install.
censes:	License text:
<ul> <li>Eclipse Foundation Software User Agreement Marketplace Client 1.7.7.v20190521-1752</li> <li>STMicroelectronics Software License Agreement</li> <li>STM32CubeIDE 1.0.2</li> <li>STM32CubeIDE ARM Toolchain Definition 1.0.2.201907121423</li> <li>STM32CubeIDE Build 1.0.2.201907091052</li> <li>STM32CubeIDE Build Analyzer 1.0.2.201907120816</li> <li>STM32CubeIDE Build UI 1.0.2.201907092029</li> <li>STM32CubeIDE C/C++ Debugging Tools for MCU 1.0.2.201907120816</li> <li>STM32CubeIDE C/C++ Embedded Development Tools for MCU 1.0.2.201907</li> <li>STM32CubeIDE Common Mx Services 1.0.2.201907091052</li> <li>STM32CubeIDE Common UI 1.0.2.201907091052</li> <li>STM32CubeIDE Common UI 1.0.2.201907091052</li> <li>STM32CubeIDE Core Toolchain Helper 1.0.2.201907091052</li> <li>STM32CubeIDE Core Toolchain Helper 1.0.2.201907091052</li> </ul>	<ul> <li>STMicroelectronics Software License Agreement</li> <li>SLA0048 Rev4/March 2018</li> <li>BY INSTALLING COPYING, DOWNLOADING, ACCESSING OR OTHERWISE USING THIS SOFTWARE PACKAGE OR ANY PART THEREOF (AND THE RELATED DOCUMENTATION) FROM STMICROELECTRONICS INTERNATIONAL N.V, SWISS BRANCH AND/OR ITS AFFILIATED COMPANIES (STMICROELECTRONICS), THE RECIPIENT, ON BEHALF OF HIMSELF OR HERSELF, OR ON BEHALF OF ANY ENTITY BY WHICH SUCH RECIPIENT IS EMPLOYED AND/OR ENGAGED AGREES TO BE BOUND BY THIS SOFTWARE PACKAGE LICENSE AGREEMENT.</li> <li>I accept the terms of the license agreements</li> <li>I do not accept the terms of the license agreements</li> </ul>

The progress bar displayed at the bottom of the STM32CubeIDE window shows the installation completion rate. Restart STM32CubeIDE when the update is finished.

#### 10.2 Install from the Eclipse<sup>®</sup> market place

It is possible to install additional third-party Eclipse<sup>®</sup> plugins in STM32CubeIDE using the Eclipse Marketplace. To install from Eclipse Marketplace, select menu [**Help**]>[Eclipse Marketplace...].

		Figure 254. Eclipse	Marketplace menu	
IDE works	spa	ce_um1 - NUCLEO-F401RE/	/Core/Src/main.c - STM32Cub	elDE
File Edit S	Sou	rce Refactor Navigate Sea	rch Project Run Window Help	D
📑 🗕 🖪 (	6	Information Center		: 🍠 🔂 🛛
🔓 Projec	?	Help Contents Search		ore_cm4
		Show Contextual Help		hfigur
v III NI I	\$	Show Active Keybindings Tips and Tricks	Ctrl+Shift+L	nClock nCore(
> 🛞		Cheat Sheets		ER COD
> 🔊	۹	Eclipse User Storage	Open the Eclipse Marketplace	e wizard
v 😕 (	~~ ₽	Check for Updates Install New Software		ER COL
~	4	Eclipse Marketplace		itiali
Ť		Data Refresh		[O_Ini
		Check for Updates		
		Manage embedded softwa ST-LINK Upgrade	are packages	ER COD
	IDE	About STM32CubeIDE		

## 254 Eclipso Markotol

The *Eclipse Marketplace* dialog opens. Search for the plugin or use the tabs (*Recent, Popular, Favorites*) to find the software wanted and install it.

IDE Eclipse Marketplace	×
Eclipse Marketplace	
Select solutions to install. Press Install Now to proceed with installation. Press the "more info" link to learn more about a solution.	
Search Recent Popular Favorites Installed 🖓 Giving IoT an Edge	
Darkest Dark Theme with DevStyle CI 2019.9.16	^
Darkest Dark is now DevStyle - a free plugin providing an enhanced set of experiences for Eclipse. Included: Darkest Dark theme - #1 in the Marketplace: True <u>more info</u> by <u>Genuitec, LLC</u> , Commercial - Free dark theme Darkett Dark Genuitec	
± 3048 → Installs: <b>987K</b> (41 486 last month)	
Spring Tools 4 - for Spring Boot (aka Spring Tool Suite 4) 4.4.1.RELEASE Spring Tools 4 is the next generation of Spring Boot tooling for your favorite coding	>
Marketplaces	
? < Back Install Now > Finish Cancel	

Figure 255. Eclipse marketplace

Wait until the installation is finished and restart STM32CubeIDE.

# 10.3 Install using [Install new software...]

Another way to install new software is to use menu [Help]>[Install New Software...].

*Note:* When installing a new toolchain, it is recommended to use the Toolchain Manager described in Section 2.11 Toolchain Manager.

# Figure 256. Install new software menu

IDE workspace\_um1 - NUCLEO-F401RE/Core/Src/main.c - STM32Cube

File Edit Source Refactor Navigate Search Project Run Window Help

	- 84	0	Information Center	
>	Projec ह my	?	Help Contents Search Show Contextual Help	
*	> % > %	\$	Show Active Keybindings Tips and Tricks Cheat Sheets	Ctrl+Shift+L
	>	2 2	Eclipse User Storage Check for Updates	>
		<b>6</b> 3.	Install New Software	
		4	Eclipse Marketplace	
			Data Refresh	
			Check for Updates	
			Manage embedded software packages	
			ST-LINK Upgrade	
		IDE	About STM32CubeIDE	

The Install dialog opens. Enter the plugin update site URL. If the URL is not known, use [--All Available Sites--].

Figure 257. Install new software

IDE Install			
Available Software			
Select a site or enter the location of a site.			
Work with: type or select a site	×	Add	Manage
type or select a site type filter tAll Available Sites			Select All
Name http://download.eclipse.org/mpc/releases/1.7.7			Deselect All
The http://download.eclipse.org/usssdk/updates/release/latest			
https://download.eclipse.org/releases/2019-09			
http://www.genuitec.com/updates/devstyle/ci/			
Details			
			0
$\boxdot$ Show only the latest versions of available software	Hide items that are already installed		
Group items by category	What is <u>already installed</u> ?		
Show only software applicable to target environment			
Contact all update sites during install to find required software			
٢	( Deally Navet	Finish	Canaal
	< Back Next >	Finish	Cancel

If no direct Internet connection is available, the plugin can be downloaded into an archive on a computer with an Internet connection, and then manually transferred to the computer with an STM32CubeIDE installation. Add the archived file by clicking on the [Add...] button and then select [Archive and select the downloaded file].

IDE Add Re	epository		×
Name:			Local
Location:	http://		Archive
?		Add	Cancel

#### Figure 258. Install new software from computer

Select the appropriate plugins and install the software. Restart STM32CubeIDE when installation is finished.

Remember: Not all Eclipse<sup>®</sup> plugins are compatible with STM32CubeIDE.

# 10.4 Uninstalling installed additional Eclipse<sup>®</sup> plugins

To uninstall a plugin that is no longer needed, select menu [Help]>[About STM32CubeIDE].

### Figure 259. About STM32CubeIDE

DE About STM32CubelDE				×
STM32 CubeIDE	STM32CubelDE Version: 1.4.0.20rc1 Build: 7141_20200610_1836 (UTC) (C) 2019 STMicroelectronics ALL RIGHTS RESERVED			
C: 😑 💷 💕	💉 💽 💺			
Installation Details		C	Close	

#### Press the [Installation Details] button to open the STM32CubeIDE Installation Details dialog.

STM32CubeIDE Installation Details			— C	ı ×
Installed Software Installation History Features Plug-ins Configuration				
type filter text				
Name	Version	Id	Provider	
	1.11.0.201909171704	com.genuitec.eclipse.theming.feature.feature.gr.	Genuitec, L	LC
> list STM32CubeIDE	1.1.0	com.st.stm32cube.ide.mcu.rcp.product		
<				>
DevStyle provides themes & more to enhance Eclipse's style, including the #1 Darkest Dark theme.				^
				· ·
0	Update	Uninstall Properties	CI	ose

#### Figure 260. Installation details

Select the plugin to uninstall in the *Installed Software* tab and press [**Uninstall...**]. Restart STM32CubeIDE when the uninstallation is finished.

# **10.5** Update to new CDT<sup>™</sup>

When a new version of STM32CubeIDE is installed based on a new version of Eclipse<sup>®</sup>, CDT<sup>™</sup> or both, it is recommended to create a new workspace instead of using a former workspace. The following warning is displayed when trying to use an old workspace with a new STM32CubeIDE.

#### Figure 261. Older workspace version warning

DE Older	Workspace Version	$\times$
i	Workspace 'C:/Users/ /STM32CubeIDE/workspace_um' was written with an older version the product and will be updated. Updating the workspace can make it incompatible with older versions of the product.	of
	Continue with this workspace?	
Do r	not warn again about workspace versions Continue Change Workspace Exit	

# **11** References

57

Reference	Document short name	Description	Document source
[ST-01]	DB3871	STM32CubeIDE data brief	
[ST-02]	RN0114	STM32CubeIDE release note	
[ST-03]	UM2553	STM32CubeIDE quick start guide	
[ST-04]	UM2563	STM32CubeIDE installation guide	
[ST-05]	UM2578	Migration guide from TrueSTUDIO <sup>®</sup> to STM32CubeIDE	www.st.com
[ST-06]	UM2579	Migration guide from System Workbench to STM32CubeIDE	-
[ST-07]	UM2576	STM32CubeIDE ST-LINK GDB server	
[ST-08]	Getting started with p	Refer to STM32CubeIDE in the <i>"Tools"</i> section of wiki.st.com/stm32mpu	
[ST-09]	AN5361	Getting started with projects based on dual-core STM32H7 microcontrollers in STM32CubeIDE	
[ST-10]	AN5394	Getting started with projects based on the STM32L5 Series in STM32CubeIDE	-
[ST-11]	AN5564	Getting started with projects based on dual-core STM32WL microcontrollers in STM32CubeIDE	
[ST-12]	AN4296	Use STM32F3/STM32G4 CCM SRAM with IAR Embedded Workbench <sup>®</sup> , Keil <sup>®</sup> MDK- ARM, STMicroelectronics STM32CubeIDE and other GNU-based toolchains	- www.st.com
[ST-13]	SLA0048	License agreement applicable to STM32CubeIDE	_
[ST-14]	UM1718	STM32CubeMX for STM32 configuration and initialization C code generation	

# Table 28. STMicroelectronics reference documents

1. Legacy application note AN5360 remains available on www.st.com.

## Table 29. External reference documents

Reference	Description	Document source
[EXT-01]	GNU Assembler	
[EXT-02]	GNU Compiler Collection	
[EXT-03]	GNU C Library	
[EXT-04]	GNU C Preprocessor	
[EXT-05]	GNU Linker	
[EXT-06]	GNU Binary Utilities	GNU tool suite <sup>(1)</sup>
[EXT-07]	Red Hat Newlib C Library	
[EXT-08]	Red Hat Newlib C Math Library	
[EXT-09]	Newlib nano readme	
[EXT-10]	Debugging with GDB	
[EXT-11]	GDB Quick Reference Card	
[EXT-12]	GNU Tools for STM32 Patch list	Information Center

1. For GNU documentation principles, refer to www.gnu.org.

# **Revision history**

# Table 30. Document revision history

Date	Revision	Changes
24-Jul-2020	1	Initial release.
2-Nov-2020	2	<ul> <li>Document updated for STM32CubeIDE v1.5.0:</li> <li>Only one toolchain installed by default</li> <li>The SFRs view displays Arm<sup>®</sup> Cortex<sup>®</sup> core registers node</li> <li>Debug with OpenOCD supports SWV and live expressions</li> <li>Added Preferences - Build variables</li> <li>Added Toolchain Manager</li> <li>Added RTOS-aware debugging with FreeRTOS™ information</li> <li>Added General debug and run launch flow</li> <li>Added Post-build with makefile targets</li> </ul>
18-Feb-2021	3	<ul> <li>Document updated for STM32CubeIDE v1.6.0:</li> <li>Added the Azure RTOS ThreadX section into chapter RTOS-aware debugging. Reorganized the FreeRTOS section</li> <li>Updated the Toolchain Manager section for the support of local toolchains</li> <li>Updated the Project C/C++ build settings section, MCU toolchain selection moved</li> <li>Updated Information Center</li> <li>Updated the entire document for the "SWV packet" terminology</li> <li>Updated References</li> <li>Removed Section 4.3.3 SWV Exception Timeline Graph</li> </ul>
5-Jul-2021	4	<ul> <li>Document updated for STM32CubeIDE v1.7.0:</li> <li>Added Section 2.7 Thread-safe wizard for empty projects and CDT projects</li> <li>Added Section 3.8 Import STM32 Cortex-M executable</li> <li>Added Section 6.3 RTOS-kernel-aware debug</li> <li>Updated Information Center – Home page</li> <li>Updated Headless build description</li> <li>Updated Section 2.5.6 Linker script with new memory map layout figure and additional description</li> <li>Updated Position-independent code description</li> <li>Updated debug configuration descriptions for ST-LINK GDB server, OpenOCD and SEGGER in Debug using different GDB servers</li> <li>Updated FreeRTOS Task List view</li> </ul>
17-Nov-2021	5	<ul> <li>Document updated for STM32CubeIDE v1.8.0:</li> <li>Added Section 1.3.3 Videos</li> <li>Added Section 6.1.10 Azure RTOS TraceX tool</li> <li>Updated Section 2.2.2 Creating a new STM32 static library project</li> <li>Removed Section 1.3.3 Technical documentation and Section 1.3.4 Closing the Information Center</li> </ul>

# Contents

1	Gett	ing stai	rted	2
	1.1	Produc	ct information	2
		1.1.1	System requirements	3
		1.1.2	Downloading the latest STM32CubeIDE version	3
		1.1.3	Installing STM32CubeIDE	3
		1.1.4	License	4
		1.1.5	Support	4
	1.2	Using	STM32CubeIDE	4
		1.2.1	Basic concept and terminology	4
		1.2.2	Starting STM32CubeIDE	5
		1.2.3	Help system	6
	1.3	Inform	ation Center	6
		1.3.1	Accessing the Information Center	7
		1.3.2	Home page	8
		1.3.3	Videos	9
	1.4	Perspe	ectives, editors and views	9
		1.4.1	Perspectives	10
		1.4.2	Editors	14
		1.4.3	Views	14
		1.4.4	Quick Access edit field	16
	1.5	Config	juration - Preferences	17
		1.5.1	Preferences - Editors	18
		1.5.2	Preferences - Code style formatter	19
		1.5.3	Preferences - Network proxy settings	21
		1.5.4	Preferences - Build variables	22
	1.6	Works	paces and projects	23
	1.7	Manag	ging existing workspaces	23
		1.7.1	Backup of preferences for a workspace	24
		1.7.2	Copy preferences between workspaces	24
		1.7.3	Keeping track of Java heap space	24
		1.7.4	Unavailable workspace	24



	1.8	STM32	CubeIDE and Eclipse <sup>®</sup> basics	25
		1.8.1	Keyboard shortcuts	25
		1.8.2	Editor zoom in and zoom out	27
		1.8.3	Quickly find and open a file	27
		1.8.4	Branch folding	28
		1.8.5	Block selection mode	28
		1.8.6	Compare files	31
		1.8.7	Local file history	33
2	Creat	ting and	I building C/C++ projects	38
	2.1	Introduc	ction to projects	38
	2.2	Creating	g a new STM32 project	38
		2.2.1	Creating a new STM32 executable project	38
		2.2.2	Creating a new STM32 static library project	43
	2.3	Configu	re the project build setting	49
		2.3.1	Project build configuration	50
		2.3.2	Project C/C++ build settings	55
	2.4	Building	the project	62
		2.4.1	Building all projects	63
		2.4.2	Build all build configurations	63
		2.4.3	Headless build	64
		2.4.4	Temporary assembly file and preprocessed C code	65
		2.4.5	Build logging	65
		2.4.6	Parallel build and build behaviour	65
		2.4.7	Post-build with makefile targets	66
	2.5	Linking	the project	66
		2.5.1	Run time library	67
		2.5.2	Discard unused sections	69
		2.5.3	Page size allocation for malloc	70
		2.5.4	Include additional object files	71
		2.5.5	Treat linker warnings and errors	71
		2.5.6	Linker script	73
		2.5.7	Modify the linker script	80



	2.5.8	Include libraries	. 87
	2.5.9	Referring to projects	. 89
2.6	I/O red	lirection	. 90
	2.6.1	printf() redirection	. 90
2.7	Thread	l-safe wizard for empty projects and CDT <sup>™</sup> projects	. 92
2.8	Positio	n-independent code	. 99
	2.8.1	Adding the -fpie option	100
	2.8.2	Run time library	101
	2.8.3	Stack pointer configuration	101
	2.8.4	Interrupt vector table	101
	2.8.5	Global offset table	101
	2.8.6	Interrupt vector table and symbols	102
	2.8.7	Debugging position-independent code	102
2.9	Exporti	ing projects	104
2.10	Importi	ing existing projects	106
	2.10.1	Importing an STM32CubeIDE project	106
	2.10.2	Importing System Workbench and TrueSTUDIO <sup>®</sup> projects	108
	2.10.3	Importing using project files association	111
	2.10.4	Prevent "GCC not found in path" error	111
2.11	Toolcha	ain Manager	111
	2.11.1	Install new toolchain	114
	2.11.2	Manage default toolchain	117
	2.11.3	Uninstall toolchain	118
	2.11.4	Using local toolchain	119
	2.11.5	Network error	123
Debu	ugging	1	24
3.1	Introdu	iction to debugging	124
	3.1.1	General debug and run launch flow	125
3.2	Debug	configurations	126
	3.2.1	Debug configuration	127
	3.2.2	Main tab	127
	3.2.3	Debugger tab	128

	3.2.4	<i>Startup</i> tab
3.3	Manag	e debug configurations
3.4	Debug	using different GDB servers
	3.4.1	Debug using the ST-LINK GDB server
	3.4.2	Debug using OpenOCD and ST-LINK
	3.4.3	Debug using SEGGER J-Link
3.5	Start a	nd stop debugging141
	3.5.1	Start debugging
	3.5.2	Debug perspective and views
	3.5.3	Main controls for debugging
	3.5.4	Run, start and stop a program
	3.5.5	Set breakpoints
	3.5.6	Attach to running target
	3.5.7	Restart or terminate debugging
3.6	Debug	features
	3.6.1	Live Expressions view
	3.6.2	Shared ST-LINK
	3.6.3	Debug multiple boards
	3.6.4	STM32H7 multicore debugging154
	3.6.5	STM32MP1 debugging
	3.6.6	STM32L5 debugging154
3.7	Run co	nfigurations
3.8	Import	STM32 Cortex <sup>®</sup> -M executable156
Debu	ıg with	Serial Wire Viewer tracing (SWV)
4.1	Introdu	ction to SWV and ITM161
4.2	SWV d	ebugging
	4.2.1	SWV debug configuration
	4.2.2	SWV settings configuration
	4.2.3	SWV tracing
4.3	SWV v	iews
	4.3.1	SWV Trace Log
	4.3.2	SWV Exception Trace Log



		4.3.3	SWV Data Trace
		4.3.4	SWV Data Trace Timeline Graph
		4.3.5	SWV ITM Data Console and printf redirection
		4.3.6	SWV Statistical Profiling
	4.4	Change	e the SWV trace buffer size
	4.5	Commo	on SWV problems
5	Spec	ial Fund	ction Registers (SFRs) 179
	5.1	Introduc	ction to SFRs
	5.2	Using th	ne SFRs view
	5.3	Updatin	g CMSIS-SVD settings
6	RTOS	S-aware	debugging
	6.1	Azure®	RTOS ThreadX
		6.1.1	Finding the views
		6.1.2	ThreadX Thread List view
		6.1.3	ThreadX Semaphores view
		6.1.4	ThreadX Mutexes view
		6.1.5	ThreadX Message Queues view
		6.1.6	ThreadX Event Flags view
		6.1.7	ThreadX Timers view
		6.1.8	ThreadX Memory Block Pools view
		6.1.9	ThreadX Memory Byte Pools view
		6.1.10	Azure <sup>®</sup> RTOS TraceX tool
	6.2	FreeRT	OS <sup>™</sup> 194
		6.2.1	Requirements
		6.2.2	Finding the views
		6.2.3	FreeRTOS Task List view
		6.2.4	FreeRTOS Timers view
		6.2.5	FreeRTOS Semaphores view
		6.2.6	FreeRTOS Queues view
	6.3	RTOS-k	kernel-aware debug
7	Fault	Analyz	er



Con	tents			4
Rev	ision h	istory	23	3
11	Refer	ences	23	0 1
	10.4	Undete	to now CDT <sup>TM</sup>	9
	10.4	Linineta	lling installed additional Eclinse <sup>®</sup> nlugins	á
	10.3	Install u	sing [Install new software] 22	7
	10.2	rom the Eclipse <sup>®</sup> market place 22	6	
	10.1	0.1 Check for updates		
10	Installing updates and additional Eclipse <sup>®</sup> plugins			4
		9.2.5	Copy and paste	3
		9.2.4	Using the filter and search field	1
		9.2.2	Call Graph tab	9
		9.2.1	Enable stack usage information    21	8
	9.2	Using th	ne Static Stack Analyzer	7
	9.1	Introduc	ction to the <i>Static Stack Analyzer</i> 21	6
9	Statio	Stack	Analyzer	6
		8.2.2	Memory Details tab	9
		8.2.1	Memory Regions tab	8
	8.2	Using th	ne Build Analyzer	8
8.1 Introdu			tion to the <i>Build Analyzer</i>	8
8	Build	Analyz	20 zer	8
	7.2	Using the Fault Analyzer view		
	7.1	Introduc	ction to the Fault Analyzer	4

# List of tables

Table 1.	Examples of toolchain build variables	22
Table 2.	Key shortcut examples	26
Table 3.	Memory map layout	74
Table 4.	Toolchain Manager column details	113
Table 5.	Toolchain Manager button information	113
Table 6.	SWV Trace Log columns details.	168
Table 7.	SWV Exception Trace Log – Data columns details	169
Table 8.	SWV Exception Trace Log – Statistics columns details	170
Table 9.	SWV Data Trace columns details	172
Table 10.	SWV Statistical Profiling columns details.	176
Table 11.	ThreadX Thread List details	185
Table 12.	ThreadX Semaphores details	186
Table 13.	ThreadX Mutexes details	186
Table 14.	ThreadX Message Queues details	187
Table 15.	ThreadX Event Flags details	188
Table 16.	ThreadX Timers details.	188
Table 17.	ThreadX Memory Block Pools details	189
Table 18.	ThreadX Memory Byte Pools details	190
Table 19.	FreeRTOS Task List details	198
Table 20.	FreeRTOS Timers details	199
Table 21.	FreeRTOS Semaphores details	200
Table 22.	FreeRTOS Queues details	200
Table 23.	Memory Regions tab information	209
Table 24.	Memory Regions usage color	209
Table 25.	Memory Details tab information	210
Table 26.	Static Stack Analyzer List tab details	219
Table 27.	Static Stack Analyzer Call Graph tab details	220
Table 28.	STMicroelectronics reference documents	231
Table 29.	External reference documents	232
Table 30.	Document revision history	233

# List of figures

Figure 1.	STM32CubeIDE key features	3
Figure 2.	STM32CubeIDE window	5
Figure 3.	STM32CubeIDE Launcher – Workspace selection	6
Figure 4.	Help menu	6
Figure 5.	Help - Information Center menu	7
Figure 6.	Information Center – Home page	8
Figure 7.	Information Center – Video browser page	9
Figure 8.	Reset perspective	0
Figure 9.	Toolbar buttons for switching perspective	0
Figure 10.	C/C++ perspective	11
Figure 11.	Debug perspective	11
Figure 12.	Device Configuration Tool perspective	2
Figure 13.	Remote System Explorer perspective	13
Figure 14.	New connection	13
Figure 15.	[Show View] menu	4
Figure 16.	Show View dialog	15
Figure 17.	Quick access	16
Figure 18.	Preferences	17
Figure 19.	Preferences - Text Editors	18
Figure 20.	Preferences - Formatter	19
Figure 21.	Preferences - Code style edit	20
Figure 22.	Preferences - Network Connections	21
Figure 23	Preferences – Build variables	 >2
Figure 24.	Pre-build step using build variables	 >2
Figure 25.	Preferences - Workspaces	23
Figure 26.	Display of Java heap space status	.0 24
Figure 27	Workspace unavailable	25
Figure 28	Shortcut keys	.0 25
Figure 20.	Shortcut preferences	.0 26
Figure 30	Editor with text zoomed in	.0 70
Figure 31	Editor folding	., 28
Figure 32		.0 20
Figure 32.		 20
Figure 34		.9 20
Figure 34.		20 21
Figure 36		21 22
Figure 30.		22
Figure 38		י <i>ב</i> י גע
Figure 30.	Show local history	ло 2Л
Figure 35.		25
Figure 40.		20
Figure 41.		27
Figure 42.		20
Figure 43.	STM32 larget selection	20
Figure 44.		10
Figure 45.		11
Figure 40.		21 14
Figure 47.		81 10
Figure 40.		⊮∠ 10
Figure 49.		}∠ 10
Figure 50.		13 1⊿
Figure 51.	New C/G++ project	14 17
Figure 52.		ιD

57

Figure 53.	Project configuration selection
Figure 54.	Project default target selector
Figure 55.	Project MCU/MPU selector
Figure 56.	Project target selection
Figure 57.	Project target selection (advanced)
Figure 58.	Project target change
Figure 59.	Set the active build configuration using the toolbar 50
Figure 60.	Set active build configuration using right-click
Figure 61.	Set active build configuration using menu
Figure 62.	Manage Configurations dialog
Figure 63.	Create a new build configuration
Figure 64.	Updated Manage Configurations dialog
Figure 65.	Configuration deletion dialog
Figure 66.	Configuration renaming dialog
Figure 67.	Properties tabs
Figure 68.	Properties configurations
Figure 69.	Properties toolchain version
Figure 70.	Properties toolchain selection
Figure 71.	Properties tool MCU settings
Figure 72.	Properties tool MCU post-build settings
Figure 73.	Properties tool GCC assembler settings 59
Figure 74.	Properties tool GCC compiler settings 60
Figure 75	Properties tool GCC linker settings 61
Figure 76	Properties build steps settings 62
Figure 77	Project build toolbar
Figure 78	
Figure 70.	Project build all
Figure 80	Project build all configurations
Figure 80.	
Figure 01.	Decellel build
Figure 02.	
Figure 03.	
Figure 84.	Linker run time library and flasting point numbers
Figure 85.	Linker newlip-nano library and floating-point numbers
Figure 86.	
Figure 87.	Linker include additional object files
Figure 88.	Linker fatal warnings
Figure 89.	Linker memory output
Figure 90.	Linker memory output specified order
Figure 91.	Linker memory displaying file readme
Figure 92.	Include a library
Figure 93.	Add library header files to the include paths
Figure 94.	Set project references
Figure 95.	Select a wizard
Figure 96.	Thread-Safe Solution wizard
Figure 97.	Thread-safe source folder location
Figure 98.	Thread-safe strategy selection
Figure 99.	Thread-safe properties
Figure 100.	Thread-safe files
Figure 101.	Thread-safe error dialog
Figure 102.	Position independent code, -fPIE 100
Figure 103.	Debugging position independent code
Figure 104.	Export project
Figure 105.	Export dialog
Figure 106.	Export archive
-	

Figure 107	Import project	107
Figure 108		107
Figure 109		108
Figure 110	Import System Workbench projects (1 of 3)	100
Figure 111	Import System Workbench projects (2 of 3)	110
Figure 112	Import System Workbanch projects (2 of 3)	110
Figure 112.		111
Figure 113.		111
Figure 114.		112
Figure 115.		112
Figure 116.		114
Figure 117.		114
Figure 118.		115
Figure 119.		115
Figure 120.		116
Figure 121.		116
Figure 122.		116
Figure 123.		117
Figure 124.	Default toolchain updated	117
Figure 125.	Uninstall toolchain.	118
Figure 126.	Uninstall details	118
Figure 127.	Software updates	119
Figure 128.	Toolchain uninstalled	119
Figure 129.	Add local toolchain	120
Figure 130.	Specify local toolchain location	121
Figure 131.	Specify local toolchain prefix	121
Figure 132.	Local toolchain added	122
Figure 133.	Edit local toolchain	122
Figure 134.	Toolchain network error	123
Figure 135.	General debug and run launch flowchart	125
Figure 136.	Debug as STM32 MCU	126
Figure 137.	Debug as STM32 MCU menu.	127
Figure 138.	Debug configuration main tab.	128
Figure 139.	Debug configuration debugger tab	129
Figure 140.	GDB server command line dialog	130
Figure 141.	Debug configuration startup tab	131
Figure 142.	Add/Edit item	132
Figure 143.	Manage debug configurations.	133
Figure 144.	Manage debug configurations toolbar	133
Figure 145.	ST-I INK GDB server debugger tab	135
Figure 146.	OpenOCD debugger tab	138
Figure 147.	SEGGER debugger tab	140
Figure 148		142
Figure 140.		143
Figure 145.		1/3
Figure 150.		145
Figure 151.		145
Figure 152.		140
Figure 153.	Breakpoint properties	1/17
Figure 154.	Conditional brookpoint	1/10
Figure 155.		140
Figure 156.	Startup tab attachar	149
Figure 157.		100
Figure 158.		150
Figure 159.		151
⊢ıgure 160.	Restart configurations dialog with additional command	152

Figure 161.	Select restart configuration.	152
Figure 162.	Live Expressions	153
Figure 163.	Live expressions number format	153
Figure 164.	Run configurations startup tab	155
Figure 165.	Cortex <sup>®</sup> -M executable import dialog	156
Figure 166.	STM32 Cortex <sup>®</sup> -M executable dialog	157
Figure 167	STM32 Cortex <sup>®</sup> -M executable MCU/MPU selection	157
Figuro 168	STM32 Cortex®-M CPUL and core	158
Figure 100.	Cartov <sup>®</sup> M debug configuration for imported project	150
Figure 169.		109
Figure 170.		160
Figure 171.		162
Figure 172.		162
Figure 175.		164
Figure 174.		164
Figure 175.		164
Figure 170.		104
Figure 177.		100
Figure 170.	[Perceve all collected SW0/ data] toolbar button	167
Figure 179.		107
Figure 180.		107
Figure 101.		107
Figure 102.	SWW graph views exita toolbar	100
Figure 184	SWV Trace Log PC sampling and exceptions	100
Figure 185	SWV Exception Trace Log – Data tab	160
Figure 185.		109
Figure 180.		171
Figure 187.		172
Figure 180.		172
Figure 190	SWV Settings	174
Figure 190.	SWV ITM Data Console	174
Figure 191.		175
Figure 192.	SW/V Statistical Profiling	176
Figure 194	SWV Preferences	177
Figure 194.	Onen the SERs view using the [Ouick Access] field	179
Figure 196	SFRs view	180
Figure 197	SFRs view toolbar buttons	181
Figure 198	SFRs CMS/S-SV/D Settings	181
Figure 199.	ThreadX views selectable from the menu	183
Figure 200.	ThreadX Thread List view (default)	184
Figure 201.	ThreadX Thread List view (Stack Usage enabled)	184
Figure 202.	ThreadX Semaphores view	186
Figure 203.	ThreadX Mutexes view	186
Figure 204.	ThreadX Message Queues view	187
Figure 205.	ThreadX Event Flags view	188
Figure 206.	ThreadX Timers view.	188
Figure 207.	ThreadX Memory Block Pools view	189
Figure 208.	ThreadX Memory Byte Pools view.	190
Figure 209.	File associations	191
Figure 210.	RAM buffer export (1 of 2)	192
Figure 211.	RAM buffer export (2 of 2)	192
Figure 212.	Existing trace overwrite	193
Figure 213.	TraceX analysis	193
Figure 214.	FreeRTOS <sup>™</sup> -related views selectable from the menu	196
-		

Figure 215.	FreeRTOS Task List (default)	. 196
Figure 216.	FreeRTOS <sup>™</sup> Toggle Stack Checking	. 197
Figure 217.	FreeRTOS Task List (Min Free Stack enabled)	. 197
Figure 218.	FreeRTOS Task List with ConfigRECORD_STACK_HIGH_ADDRESS enabled.	. 197
Figure 219.	FreeRTOS Timers.	. 198
<b>Figure 220</b> .	FreeRTOS Semaphores	. 199
Figure 221.	FreeRTOS Queues	. 200
<b>Figure 222.</b>	RTOS-kernel-aware debug.	. 201
Figure 223.	RTOS-kernel-awareness debug configuration	. 202
Figure 224.	ThreadX-kernel-awareness debug configuration	. 202
Figure 225.	ThreadX port configuration	. 203
Figure 226.	FreeRTOS <sup>™</sup> port configuration	. 203
Figure 227.	Open the Fault Analyzer view	. 204
Figure 228.	Fault Analyzer view	. 206
Figure 229.	Fault Analyzer toolbar	. 207
Figure 230.	Fault analyzer open editor on fault	. 207
Figure 231.	Fault Analyzer open disassembly on fault.	. 207
Figure 232.	Build analyzer.	. 208
Figure 233.	Memory Regions tab	. 209
Figure 234.	Memory Details tab	. 210
Figure 235.	Memory Details sorted by size	. 212
Figure 236.	Memory Details search and filter	. 212
Figure 237.	Sum of sizes	. 213
Figure 238.	Show byte count	. 213
Figure 239.	Show hex count	. 214
Figure 240.	Copy and paste	. 214
Figure 241.	Static Stack Analyzer List tab	. 216
Figure 242.	Static Stack Analyzer Call Graph tab	. 217
Figure 243.	Open the Static Stack Analyzer view	. 217
Figure 244.	Enable generate per function stack usage information	. 218
Figure 245.	Static Stack Analyzer List tab	. 219
Figure 246.	Static Stack Analyzer Call Graph tab	. 220
Figure 247.	Function symbols in <i>Static Stack Analyzer</i>	. 221
Figure 248.	Static Stack Analyzer List tab using search	. 222
Figure 249.	Static Stack Analyzer Call Graph using search	. 222
Figure 250.	Copy and paste	. 223
Figure 251.	STM32CubeIDE available updates	. 224
Figure 252.	STM32CubeIDE update details	. 225
Figure 253.	STM32CubeIDE update review licenses	. 225
Figure 254.	Eclipse Marketplace menu	. 226
Figure 255.	Eclipse marketplace	. 227
Figure 256.		. 228
Figure 257.	Install new software.	. 228
⊢igure 258.	Install new software from computer	. 229
Figure 259.	About S1M32CubeIDE	. 229
Figure 260.		. 230
Figure 261.	Older workspace version warning	. 230

#### IMPORTANT NOTICE - PLEASE 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 acknowledgement.

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, please 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.

© 2021 STMicroelectronics - All rights reserved