

UM2052 User manual

Getting started with STM32 MCU Discovery Kits software development tools

Introduction

The STM32 Discovery boards are low-cost and easy-to-use development platforms used to quickly evaluate and start a development with an STM32 device.

This document provides guidelines for beginners on STM32 MCU Discovery Kits on how to build and run a sample application and allows them to build and debug their application. It has the following structure:

- The first chapter presents software and hardware requirements (some toolchains supporting the STM32 families, ST-LINK/V2 installation and firmware package presentation).
- The second chapter provides a step by step guideline on how to build and debug an application using some toolchains:
 - IAR Embedded Workbench[®] for Arm[®] (EWARM) by IAR systems[®]
 - Microcontroller Development Kit for Arm[®] (MDK-ARM) by Keil[®]
 - TrueSTUDIO[®] by Atollic[®]
 - System Workbench for STM32 (SW4STM32) by AC6

This manual does not cover all the topics relevant to software development environments, but it describes the first basic steps necessary to get started with the compilers and debuggers. It also offers links to the documents needed to fully understand every single step.

Туре	Part numbers
STM32 MCU Discovery Kits	STM32F7308-DK, 32F0308DISCOVERY, 32F072BDISCOVERY, 32F3348DISCOVERY, 32F411EDISCOVERY, 32F412GDISCOVERY, 32F413HDISCOVERY, 32F429IDISCOVERY, 32F469IDISCOVERY, 32F723EDISCOVERY, 32F746GDISCOVERY, 32F769IDISCOVERY, 32L0538DISCOVERY, 32L100CDISCOVERY, 32L152CDISCOVERY, 32L476GDISCOVERY, 32L496GDISCOVERY, 32L4R9IDISCOVERY, B- L072Z-LRWAN1, B-L475E-IOT01A, P-L496G-CELL01, P-L496G- CELL02, STM32F0DISCOVERY, STM32F3DISCOVERY, STM32F4DISCOVERY, STM32F7508-DK, STM32VLDISCOVERY, STM32G071B-DISCO

Table	1.	Apr	olicable	products
10010	•••	" Y PP	11000010	producto

Contents

1	Gen	eral information
2	Syst	em requirements6
	2.1	IDEs supporting STM32 families7
	2.2	ST-LINK/V2 installation
	2.3	Firmware package
3	Exec	cuting and debugging firmware using the software toolchains 10
	3.1	EWARM toolchain
	3.2	MDK-ARM toolchain
	3.3	TrueSTUDIO toolchain
	3.4	SW4STM32 toolchain 17
4	Revi	sion history



List of tables

Table 1.	Applicable products 1	l
Table 2.	Most used integrated development environments	7
Table 3.	Document revision history)



List of figures

Hardware environment	3
Firmware package content	9
IAR embedded workbench IDE)
EWARM project successfully compiled	C
Download and Debug button	1
IAR embedded workbench debugger screen	1
Go button	1
uVision5 IDE	2
MDK-ARM project successfully compiled	2
Start/Stop debug session button	2
MDK-ARM debugger screen	3
Run button	3
TrueSTUDIO workspace launcher dialog box13	3
Atollic TrueSTUDIO import projects dialog box 14	4
TrueSTUDIO project successfully compiled 1	5
SW4STM32 workspace launcher dialog box1	7
SW4STM32 import source select dialog box1	7
SW4STM32 debug window	9
	Hardware environment 6 Firmware package content 7 IAR embedded workbench IDE 10 EWARM project successfully compiled 10 Download and Debug button 11 IAR embedded workbench debugger screen 11 Go button 11 UVision5 IDE 12 MDK-ARM project successfully compiled 12 Start/Stop debug session button 12 MDK-ARM debugger screen 13 Run button 14 TrueSTUDIO workspace launcher dialog box 14 Atollic TrueSTUDIO import source select dialog box 14 TrueSTUDIO project successfully compiled 14 TrueSTUDIO project successfully compiled 14 SW4STM32 import source select dialog box 14 SW4STM32 import projects dialog box 15 SW4STM32 project successfully compiled 15 SW4STM32 project successfully compiled 16 SW4STM32 debug window 16 SW4STM32 debug window 16



1 General information

The STM32 Discovery boards embed STM32 32-bit microcontrollers based on the ${\rm Arm}^{\otimes(a)}$ Cortex $^{\otimes}$ -M processor.

arm

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



Note:

2 System requirements

Before running the application, the user must:

- 1. Install the preferred Integrated Development Environment (IDE)
- 2. The ST-LINK/V2 driver is installed automatically. In case of problem, the user can install it manually from the toolchains install directory (further details are available in *Section 2.2: ST-LINK/V2 installation*).

This step is not needed for STM32VLDISCOVERY since it embeds an ST-LINK (not an ST-LINK/V2) which does not require a driver installation.

- 3. Download the STM32 Discovery firmware from the STMicroelectronics website at *www.st.com*.
- 4. Establish the connection with the STM32 Discovery board as shown hereafter. In *Figure 1*, the STM32F072 Discovery board is chosen as an example.

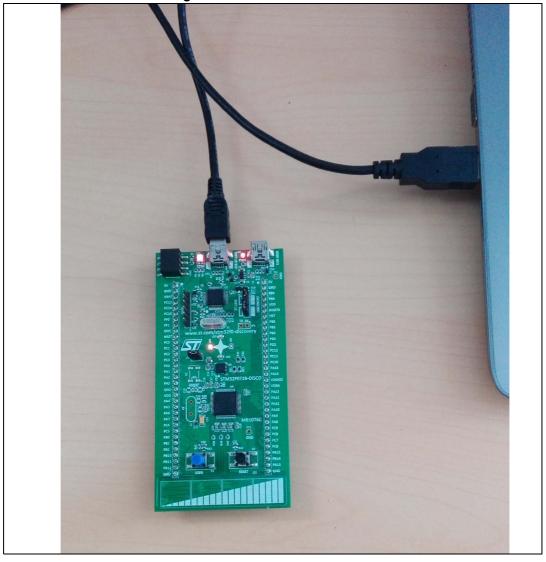


Figure 1. Hardware environment



The steps numbered above are detailed in the coming sections.

The minimum requirements to run and develop any firmware application on the STM32 Discovery board are:

- Windows[®] OS (XP, 7, 8) or Linux 64-bit or Mac OS[®] X
- USB type A to Mini-B cable, used to power the STM32 Discovery board (through USB connector CN1) from the host PC and connect to the embedded ST-LINK/V2 for debugging and programming.

2.1 IDEs supporting STM32 families

The STM32 microcontrollers of 32-bit Arm Cortex-M core-based microcontrollers are supported by a complete range of software tools.

It encompasses traditional integrated-development environments such as IDEs with C/C++ compilers and debuggers from major third-party companies, free versions of up to 64 KB of code (depending on partner) and completed with innovative tools from STMicroelectronics.

The following table includes general information about some integrated development environments as well as the versions supporting the STM32 microcontrollers.

-		-		
Toolchain	Company	Compiler	Version	Information ⁽¹⁾
EWARM ⁽²⁾	IAR Systems	IAR C/C++	7.60.1 and later	www.iar.com – 30-day evaluation edition KickStart edition – 16 Ko limitation for Cortex M0
MDK-ARM ⁽²⁾	Keil	ARMCC	5.18a and later	www.keil.com – MDK-Lite – 32 Ko code size limitation
TrueSTUDIO ⁽²⁾	Atollic	GNUC	5.5.1 and later	www.atollic.com – 30 days professional version trial – 32 Ko limitation – 8 Ko on Cortex-M0 and Cortex-M1
SW4STM32 ⁽³⁾	AC6	GNUC	1.8 and later	www.openstm32.org No limitation

Table 2. Most used integrated development environments

1. Register prior to downloading the toolchain.

2. On Windows only.

3. The SW4STM32 does not support ST-LINK. Consequently, STM32VLDISCOVERY can not be used with SW4STM32 software toolchain.

Information on the toolchain version supporting the STM32 devices is available on the toolchain release note at the third-party website.



2.2 ST-LINK/V2 installation

All the STM32 Discovery boards include the ST-LINK/V2 embedded debug tool interface. This interface requires to install the ST-LINK/V2 dedicated USB driver. The STM32 Discovery boards include an ST-LINK/V2 embedded debug tool interface that is supported by the following software toolchains:

• IAR Embedded Workbench for Arm (EWARM)

The toolchain is installed by default in the C:\Program Files\IAR Systems\Embedded Workbench x.x directory on the PC local hard disk.

After installing the EWARM, the user should install the ST-LINK/V2 driver by running the ST-Link_V2_USB.exe from *IAR_INSTALL_DIRECTORYJ\Embedded Workbench x.x\arm\drivers\ST-Link \ST-Link_V2_USBdriver.exe.*

• RealView Microcontroller Development Kit for Arm (MDK-ARM) by Keil toolchain

The toolchain is installed by default in the C:\Keil directory on the PC local hard disk; the installer creates a start menu μ Vision5 shortcut.

When connecting the ST-LINK/V2 tool, the PC detects the new hardware and asks to install the ST-LINK_V2_USB driver. The "Found New Hardware wizard" appears and guides the user through the steps needed to install the driver from the recommended location.

• Atollic TrueSTUDIO STM32

The toolchain is installed by default in the C:\Program Files\Atollic directory on the PC local hard disk.

• AC6 SW4STM32 STM32

The toolchain is installed by default in the C:\Program Files\AC6 directory on the PC local hard disk.

The ST-Link_V2_USB.exe is installed automatically when installing the software toolchain.

Note: The embedded ST-LINK/V2 only supports the SWD interface for STM32 devices.



UM2052

2.3 Firmware package

The STM32 Discovery firmware applications, demonstration and IPs examples are provided in one single package and supplied in one single .zip file. The extraction of the .zip file generates the folder STM32-Discovery_FW_VX.Y.Z, which contains the subfolders shown in *Figure 2* where the STM32 F072B-Discovery_FW_VX.Y.Z is chosen as an example.



Figure 2. Firmware package content

Template project is a pre-configured project with empty main function to be customized by the user. This is helpful to start creating an application based on the peripherals drivers.

Example project includes the toolchain projects for each peripheral example ready to be run.

Applications includes set of applications ready to be run.

Demonstration includes the demonstration firmware ready to be run.



3 Executing and debugging firmware using the software toolchains

The user can follow the procedure described below to compile/link and execute an existing EWARM project. The following steps can be applied to an already existing example, demonstration or template project available at STM32 Discovery firmware package available on STMicroelectronics website. In this manual the STM32 F072B-Discovery_FW_VX.Y.Z is used as an example.

The user should go through the firmware/readme.txt file which contains the firmware description and hardware/software requirements.

3.1 EWARM toolchain

1. Open the IAR Embedded Workbench for Arm (EWARM).

Figure 3 shows the basic names of the windows referred to in this document.

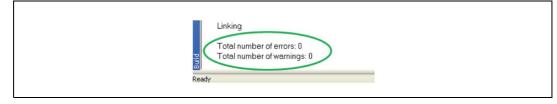
	With Embedded Warkbench IDI File Edt Was Project Tools Window Help			

			497.]	10 - ×
Workspace window — File window — Build window —		>		
	Messages		File	Line
			Fire	Cone .
	5			
	Ready	Errors D, Warnings D		NUM T

Figure 3. IAR embedded workbench IDE

- 2. In the **File** menu, select **Open** and click **Workspace** to display the **Open Workspace** dialog box. Browse to select either an example or demonstration or template workspace file and click **Open** to launch it in the Project window.
- 3. In the **Project menu**, select **Rebuild All** to compile the project
- 4. If the project is successfully compiled, the window shown in *Figure 4* is displayed.

Figure 4. EWARM project successfully compiled



UM2052 Rev 2



To change the project settings (Include and preprocessor defines), the user should go through project options:

- For 'Include directories', select Project>Options...>C/C++ compiler>
- For pre-processor, define Project>Options...C/C++ compiler>pre-processor>
- 5. In the IAR Embedded Workbench IDE, from the **Project** menu, select **Download and Debug** or alternatively, click on the **Download and Debug** button the in tool bar, to program the Flash memory and start debugging.

Figure 5. Download and Debug button

File Edit View Project Tools Win	ndow Help	
🗅 🛩 🖬 🕼 🖧 🕺 🖻 🖻	L い い	Þ
Workspace ×		ownload and Debug

The debugger in the IAR Embedded Workbench can be used to debug the source code at C and assembly levels, set breakpoints, monitor individual variables and watch events during the code execution.

Project - IAR Embedded Workbe	nch IDE - ARM 7.60.1					x
ile Edit View Project Debug	g Disassembly ST-Link Tools Window Help					
🗅 🚅 🖬 🕼 🎒 🍏 X 🖻 🛍	1 (이 이 · · · · · · · · · · · · · · · · ·					
5-882888	₩ X					
/orkspace ×	main.c	f0 - ×	Disassen	nbly		×
STM32F072B-Discovery -	/* Private function prototypes*/	-	Goto	Memory	•	3
Files 🔅 😋	<pre>void SystemClock Config(void); static uint16_t Buffercmp(uint8_t *pBuffer1, uint8_t *pBuffer2, uint16_t BufferLen</pre>	gth);		sassembly at main(void)		^
	<pre>static void Error_Handler(void); /* Private functions</pre>					
- 🕀 🗀 Example	/* FIIVALE INNELIONS			0x8001498: 0xb5f0	PUSH	
🛏 🧰 Output	<pre>/** * @brief Main program</pre>	=		0x800149a: 0xb083 HAL_Init();	SUB	
	* Gparam None * Gretval None */			0x800149c: 0xf7ff 0xfdcs SystemClock_Config(): 0x80014a0: 0xf000 0xf883		
	int main (void)			BSP LED Init(LED5):	. DL	
	/* STM32FOxx HAL library initialization: - Configure the Flash prefetch			0x80014a4: 0x2002 0x80014a6: 0xf000 0xfd2	MOVS BL	
	 Systick timer is configured by default as source of time base, but user can eventually implement his proper time base source (a general purpose timer for example or other time source), keeping in mind that Time base 			BSP_LED_Init(LED4); 0x80014aa: 0x2001 0x80014ac: 0xf000 0xfd26 BSP_LED_Init(LED3);	MOVS BL	
	duration should be kept Ims since PPP_TIMEOUT_VALUEs are defined and handled in milliseconds basis. - Low Level Initialization			0x80014b0: 0x2000 0x80014b2: 0xf000 0xfd23		
	- */ HAL_Init();			I2cHandle.Instance 0x80014b6: 0x4c38 0x80014b8: 0x4838	= I2C: LDR.N LDR.N	<u>8</u> .;
	/* Configure the system clock to 48 MHz */			0x80014ba: 0x6020	STR	
Project	SystemClock_Config();			I2cHandle.Init.Timing	= EEP	RC
rigedi	•	- + -	•	m		٢
Messages				File	Line	П
stm32f0xx_it.c system_stm32f0xx.c Linking Project.out Converting						
- Total number of errors: 0 Total number of warnings:	.0					
Debug Log Build						
adv	Errors 0	141 1		95. Col 1 System	NUM	

Figure 6. IAR embedded workbench debugger screen

To run the application, from the **Debug** menu, select **Go**. Alternatively, click on the **Go** button in the toolbar to run the application.

	Figu	re 7. (Go bı	utton	
🔀 s	iysTick - IA	R Embe	dded V	/orkbench l	DE
File	Edit View	Project	Debug	Disassembly	ST-
				6 10 CI	
<u>t</u>	1 2	26	岁2	27 ×	
EIM	SWO			Go	
Work	kspace			× mair	n.c



UM2052

3.2 MDK-ARM toolchain

1. Open Keil MDK-ARM Microcontroller Kit,

Figure 8 shows the basic names of the "Keil uVision5" windows referred to in this document.

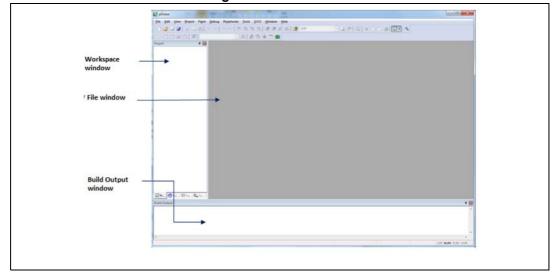
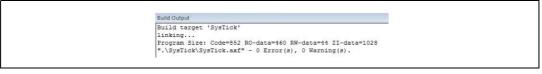


Figure 8. uVision5 IDE

- 2. In the **Project** menu, select **Open Project**. Browse to select either an example or demonstration or template project file and click **Open** to launch it in the Project window.
- 3. In the **Project** menu, select **Rebuild All** target files to compile the project
- 4. If the project is successfully compiled, the window shown in *Figure 9* is displayed.

Figure 9. MDK-ARM project successfully compiled



If the user needs to change the project settings (Include and preprocessor defines), the user should go through the project options:

- For 'Include directories', select **Project>Options for Target > C/C++ > Include Paths**
- For pre-processor definition Project>Options for Target > C/C++ > Preprocessor symbols > Define
- 5. In the MDK-ARM IDE, from the **Debug** menu, select **Start/Stop Debug Session** or, alternatively, click the **Start/Stop Debug Session** button in the tool bar to program the Flash memory and begin debugging.

Figure 10. Start/Stop debug session button

D 🐸 🖬 🎒 💰 🔩 🕯	1 日本 1 日本 1 日本 1 日本	👷 譚 譚 //É //授 🧭 GetBase 💽 🗽 🖉 🍭 🔍 🔗 🔗 🏩 💽 🔦
🧼 🖽 🎬 🧼 📇 🞇	Systick 💽 🔊 🛔 🖷	Start/Stop Debug Session (Ctrl+F5)
Project 🛛 📮 📴		Enter or leave a debug session

6. The debugger in the MDK-ARM can be used to debug source code at C and assembly levels, set breakpoints, monitor individual variables and watch events during the code execution.





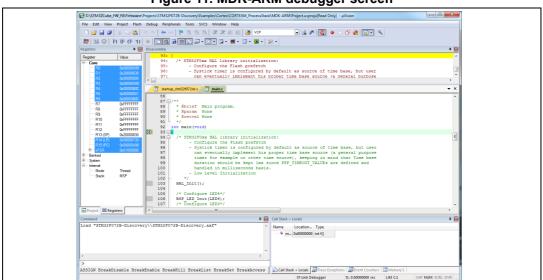


Figure 11. MDK-ARM debugger screen

To run the application, from the **Debug** menu, select **Run**. Alternatively, click the **Run** button in the toolbar to run the application.

Figure 12. Run button

新 EL ⑧ 砂 ひ O
Register B Run (F5)

3.3 TrueSTUDIO toolchain

1. Open Atollic TrueSTUDIO for Arm microcontrollers. The program launches and asks for the Workspace location.

Workspace Launcher	CO.	x
Select a workspace Atollic TrueSTUDIO® for ARM® Pro Choose a workspace folder to use fo	stores your projects in a folder called a workspace. r this session.	
Workspace: -Discovery_FW\Project Copy Settings	\Peripheral_Examples\SysTick_Example\TrueSTUDIO -	Browse
?	ОК	Cancel

Figure 13. TrueSTUDIO workspace launcher dialog box

- 2. Browse to select a TrueSTUDIO workspace of either an example or demonstration or template workspace file and click **OK** to load it.
- 3. To load an existing project in the selected workspace, select **Import** from the **File** menu to display the **Import** dialog box.
- 4. In the Import window, open **General**, select **Existing Projects** into Workspace and click **Next**.



Import
Select Create new projects from an archive file or directory.
Select an import source:
type filter text
(?) < Back Next > Finish Cancel

Figure 14. Atollic TrueSTUDIO import source select dialog box

5. Click **Select root directory**, browse to the TrueSTUDIO workspace folder.

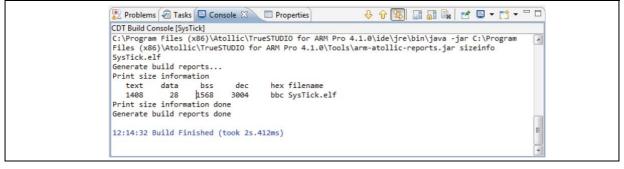
Figure 15. Atollic TrueSTUDIO import projects dialog box

Import		
Import Pro Select a dir	jects ectory to search for existing Eclipse projects.	
 Select ro Select an Projects: 	ot directory: C:\STM32F072B-Discovery_FW_VX.Y.Z\Projects chive file:	Browse
<	roject to working sets	Select
?	< Back Next > Finish	Cancel



- 6. In the **Projects** panel, select the project and click **Finish**.
- 7. In the **Project Explorer**, select the project, open the **Project** menu, and click **Build Project**.
- 8. If the project is successfully compiled, the following messages will be displayed on the Console window.

Figure 16. TrueSTUDIO project successfully compiled



If the user needs to change the project settings (Include directories and preprocessor defines), go through Project>Properties, select C/C++ Build>Settings from the left panel:

- For 'Include directories' C Compiler>Directories>Include path
- For pre-processor defines C Compiler>Symbols> Defined symbols
- 9. To debug and run the application, the user should select the project In the Project Explorer and press F11 to start a debug session (see *Figure 17*).



5 • •											
Debug - STM32F072B-Discovery/Example/User/main.c - Atollic TrueSTUDIO for ARM	-										
<u>File Edit View Run Window H</u> elp		- 0-									
🗄 🖻 🖄 🍪 🕹 🕪 🗉 🔳 🖡 🥄 👁 🖈 📴 🔗 ▼ 🖙 ⇔	- I 🛛 I 🖬			(.							
		Quick Acc	ess	🖹 🗄 🔂 C/C++	🎋 Debug						
🕸 Debug 🛛 🧏 🎽 🗖 🗖	(×)= Variables 🔀	● Breakpoints 🐰 Re	gisters 📟 SFRs	🛋 Modules	- 8						
STM32F072B-Discovery.elf [Embedded C/C++ Application]			A 🐴 🛛	a # * %	📫 🖻 🔍						
▲	Name	Туре		Value							
Thread [1] < main> (Suspended : Breakpoint) main() at main.c:96 0x8000334											
pdb											
📓 ST-LINK											
					*						
	٠										
e main.c X			E Outline								
93 * @ <u>retval</u> None 94 */		^		📄 🖓 🙀 😼	● # ~						
950 int main(void) main.h 950 int main(void) main.h 970 /* STM32F0xx HAL library initialization: # EEPROM_ADDRESS 98 - Configure the Flash prefetch 99 - Systick timer is configured by default as source of time base, but user 100 can eventually implement his proper time base source (a general purpose 101 timer for example or other time source), keeping in mind that Time base 102 duration should be kept 1ms since PPP_TIMEOUT_VALUEs are defined and 103 handled in milliseconds basis. 104 - Low Level Initialization 04 - Low Level Initialization											
						105 */		-	•	Memory Address :	uint16 t
								P			,
						📮 Console 🕱 🖉 Tasks 🐰 Problems 🜔 Executables 🧻 Memory 👘 🗖	SWV Trace Lo	g 🔀 📃 SWV Console		🔀 🖷 🗙 🛛	
						■ × ¾ 🛼 🔠 📰 🖅 🖽 🖃 ▼ 🗂 ▼	Index	Type Dat	a	Cycles	Time(s)
						STM32F072B-Discovery.elf [Embedded C/C++ Application] gdb					
Temporary breakpoint 1, main () at\\Src\main.c:96	•				•						
96 {											
4 F	Overflow pack	ELS: U									
Read-Only	Smart Insert	100 : 28									
	1	1									

Figure 17. TrueSTUDIO debug window

The debugger in the Atollic TrueSTUDIO can be used to debug source code at C and assembly levels, set breakpoints, monitor individual variables and watch events during the code execution.

To run the application, from the **Run** menu, select **Resume**, or alternatively click the **Resume** button in the toolbar.



3.4 SW4STM32 toolchain

1. Open the AC6 SW4STM32 for Arm microcontrollers. The program launches and prompts for the Workspace location.

Figure 18. SW4STM32 workspace launcher dialog box

	kspace s your projects in a folder called a workspace. rkspace folder to use for this session.		
Workspace:	cts\STM32F072B-Discovery\Examples\ADC\ADC_DMA_	Transfer\SW4STM32 🔹	Browse
?		ок	Cancel

- 2. Browse to select a SW4STM32 workspace of either an example or demonstration or template workspace file and click **OK** to load it.
- 3. To load an existing project in the selected workspace, select **Import** from the **File** menu to display the **Import** dialog box.
- 4. In the **Import** window, open **General**, select **Existing Projects** into Workspace and click **Next**.

lect	
reate new projects from an archive file or directory.	Ľ
elect an import source:	
type filter text	
General G	E

Figure 19. SW4STM32 import source select dialog box

5. Click **Select root directory**, browse to the SW4STM32 workspace folder.

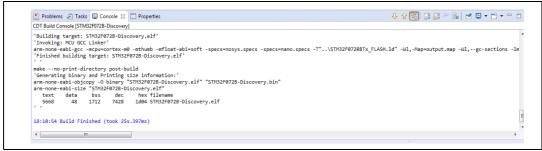


Import Projects Some projects cannot be imported because they already exist in the workspace Select root directory: C.\STMB2\STM32Cube_FW_F0_V1_2.0\Projects\STM32F0728-Discovery\Examplex\ADC\ADC_DMA_Transfer\SW45TM32 Select archive file: Projects: RemoteSystemTempFiles (CLSTM32\STM32Cube_FW_F0_V1_2.0\Projects\STM32F0728-Discovery\Examplex\ADC\ADC_DMA_Transfer\SW45TM32 PenoteSystemTem STM32F0728-Discovery (CLSTM32\STM32Cube_FW_F0_V1_2.0\Projects\STM32F0728-Discovery\Examplex\ADC\ADC_DMA_Transfer\SW45TM32\STM32F0728-Discovery\Examplex\ADC\ADC_DMA_Tran	Browse
Select archive file Projects: RemoteSystemsTempFiles (CASTM32/STM32Cube_FW_F0_V12.20.Projects/STM32F0728-Discovery/Examples/ADC/ADC_DMA_Transfer/SW4STM32/RemoteSystemsTe STM32F0728-Discovery (CASTM32/STM32Cube_FW_F0_V12.20.Projects/STM32F0728-Discovery/Examples/ADC/ADC_DMA_Transfer/SW4STM32/STM32F0728-Discovery (CASTM32/STM32/STM32Cube_FW_F0_V12.20.Projects/STM32F0728-Discovery/Examples/ADC/ADC_DMA_Transfer/SW4STM32/STM32F0728-Discovery (CASTM32/STM32/STM32Cube_FW_F0_V12.20.Projects/STM32F0728-Discovery/Examples/ADC/ADC_DMA_Transfer/SW4STM32/STM32F0728-Discovery (CASTM32/STM32/STM32/STM32/STM32F0728-Discovery/Examples/ADC/ADC_DMA_Transfer/SW4STM32/STM32F0728-Discovery (CASTM32/STM32/STM32/STM32/STM32F0728-Discovery)	• Browse
Projects: Projects: RemoteSystemsTempFiles [C\STM32.STM32Cube_FW_F0_V1.2.0\Projects\STM32F0728-Discovery\Samples\ADC\ADC_DMA_Transfer\SW4STM32\RemoteSystemsTe STM32F0728-Discovery(C\STM32\STM32Cube_FW_F0_V1.2.0\Projects\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ STM32F0728-Discovery(C\STM32\STM32Cube_FW_F0_V1.2.0\Projects\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ STM32F0728-Discovery(C\STM32\STM32Cube_FW_F0_V1.2.0\Projects\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ Tage = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1	
RemoteSystemaTempFiles IC\STM32\STM32Cube_PW_F0_VI_20\Projects\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\RemoteSystemaTe STM32F0728-Discovery (C\STM32\STM32\STM32Cube_PW_F0_VI_20\Projects\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(C\STM32\STM32\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(C\STM32\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(C\STM32\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Disc	Brows
RemoteSystemaTempFiles IC\STM32\STM32Cube_PW_F0_VI_20\Projects\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\RemoteSystemaTe STM32F0728-Discovery (C\STM32\STM32\STM32Cube_PW_F0_VI_20\Projects\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(C\STM32\STM32\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(C\STM32\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(C\STM32\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery(E\STM32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Discovery\ stm32F0728-Discovery\Examples\ADC\ADC_DMA_Transfer\SW4STM32\STM32F0728-Disc	
	Þ
Options Search for nested projects Copy projects into workspace Hide projects that already exist in the workspace	
Working sets	
Add project to working sets	
Working sets:	Select

Figure 20. SW4STM32 import projects dialog box

- 6. In the **Projects** panel, select the project and click **Finish**.
- 7. In the Project Explorer, select the project, open the Project menu, and click Build Project.
- 8. If the project is successfully compiled, the following messages display on the Console window.





If the user needs to change the project settings (Include directories and preprocessor defines), simply go through Project>Properties, select C/C++ Build>Settings from the left panel:

- For Include directories C Compiler>Directories>Include path
- For pre-processor defines C Compiler>Symbols> Defined symbols
- 9. To debug and run the application, select the project In the **Project Explorer** and press F11 to start a debug session. See *Figure 22*.



🎋 Debug 🛛 🦌 🖓 🖬 🔽 🗖 🗖	(x)= Varia 💁 Brea	💱 Expr 🔀	1919 Regis 🗰 I/O 🛋 Mo	odu 🗖 E
 STM32F072B-Discovery.elf [Ac6 STM32 Debugging] STM32F072B-Discovery.elf 	Expression		縼 🍕 🖻 🖶 💥 🔆 Type	Val
 Interad #1 (Suspended: Breakpoint) □ main() at main.c88 0x8000254 □ openocd □ C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.externaltools.arm-none.win32_1 	Add new expres	sion		
	•	m		
۲	4			Þ
<pre>@ mainc % [gdb[0].proc[42000].threadGroup[1].gdb[0].proc[42000].OSthread[1].t 86 - Low Level Initialization 87 */ 85 88 HAL_Init(); 89 /* Initialize LED5 */ 91 BSP_LED_Init(LED5); 92 </pre>	tnread (1). trame (U)			_HandleType
© Console ⊗ ⊘ Tasks ♥ Problems ● Executables ● Memory STM32F072B-Discovery.eff [Ac6 STM32 Debugging] (2/Ac6/SystemWorkbench/plugins/fr.a Find the GOB manual and other documentation resources online at: http://www.gnu.org/software/gdb/documentation/>. For help, type "help". Type "apropos word" to search for commands related to "word". Warning: the current language does not match this frame.	ac6.mcu.externaltools.arm		Ⅱ 📰 💭 (♥) 📑 🔍 ▼ 0.201507241045/tools/compiler/	bin/arm-none

Figure 22. SW4STM32 debug window

The debugger in the AC6 SW4STM32 can be used to debug source code at the C and assembly levels, to set breakpoints, to monitor individual variables and to watch events during the code execution.

To run the application, from the **Run** menu, select **Resume**, or alternatively click the **Resume** button in the toolbar.



4 Revision history

Date	Revision	Changes
20-May-2016	1	Initial release.
25-Feb-2019	2	Added Table 1: Applicable products. Added Section 1: General information.

Table 3. Document revision history



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. 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.

© 2019 STMicroelectronics – All rights reserved



UM2052 Rev 2