









# System Workbench for MCU

- System Workbench for STM32 is an embedded systems IDE developed by Ac6 for programming STM32 micro-controllers
- It is a set of Eclipse plug-ins
  - It can run under Windows or Linux
  - It can be installed either
    - Using a stand-alone installer
    - In an existing Eclipse platform (Juno or Kepler)
- It is designed to integrate all project activities in ONE environment
- It can also easily cooperate with other Eclipse plugins
  - For example, with System Workbench for Linux, it can help developing distributed applications running on micro-controllers and Linux processors
  - It will support programming multi-core asymetric SoCs For example Cortex-A + Cortex-M



# System Workbench for MCU

- System Workbench for STM32 fully supports the STM32 microcontroller family
- When creating a project you just define
  - The MCU you will use
  - The board on which the program will run, either
    - A supported evaluation board
    - A user-defined board
- Then system Workbench for STM32 will automatically:
  - Retrieve the board- and mcu-specific firmware and definitions
    - Either at the StdPeriph format or the new HAL format
  - Configure the firmware
  - Provide startup code for your C or C++ programs
  - Optionally provide an example main program
- You are then able to immediately insert, compile and debug your code



#### System Workbench for MCU Architecture





#### Main elements

#### • Main project

- Main project of the embedded system
- It contains the main source code for the system
- It references external libraries
  - Standard libraries (StdPeriph or HAL)
  - Extensions (RTOS, USB stack, ...)
  - User-provided libraries
- It is linked to an executable program
- HAL source code
  - It contains the official Cube HAL source code
- HAL library
  - It references (part of) the Cube HAL library, depending on the selected board
  - It compiles this source code creating a library, usable by all projects running on the same board
- StdPeriph source code and library
  - These projects are used in case the firmware for the board was provided using the legacy (Std\_Periph) format
- RTOS, stacks and user libraries
  - Depending on the needs, Ac6 will provide support for various RTOSes (FreeRTOS...), protocol stacks (USB, TCP/IP...), middlewares...



Users can also create their own libraries



# Creating a program



# Creating a program (1/4)

🔵 C/	/C++ - Ec	lipse	-	Concession of the local division of the loca	and succession in which the	-		
File	Edit S	ource	Refactor	Navigate	Search	Project	Run	Window Help
	New				Alt	+Shift+N	<u>کی</u>	Makefile Project with Existing Code
	Open File	e					C.	C++ Project
	Close					Ctrl+W	C°	C Project
	Close All				Ctrl	+ Shift+ W	2	Project
m	Save					Ctrl+S	C++	Convert to a C/C++ Autotools Pro
	Save As					Curto	C++	Convert to a C/C++ Project (Adds
	Save All				Ctr	l+Shift+S	63	Source Folder
46	Suve All				Cu	it shift+5	<u> </u>	Folder

- To create a program with System Workbench for STM32 you must first create a project
  - Choose a C or a C++ Project
  - Give it a name
  - Create an Ac6 MCU Project
    - Executable or Static Library
    - May also be an simple Empty project
  - Select the Ac6 MCU GCC toolchain

C Project	
C Project Create C project of selected type	
Project name: my_project	
✓ Use default location	
Location: D:\Documents and Settings\Dautrevaux\w	vorkspace3\my_project Browse
Choose file system: default 💌	
Project type:	Toolchains:
👂 🗁 GNU Autotools	Ac6 STM32 MCU GCC
a 🗁 Executable	
Empty Project	
Ac6 STM32 MCIL Project	
Shared Library	
Static Library	
Makefile project	
Show project types and toolchains only if they are	supported on the platform
?	Next > Finish Cancel



## Creating a program (2/4)

C Project		C Project	x
Select Configurations Select platforms and configurations you wish to deploy on		MCU Configuration Select the board and configurations	4
Project type: Executable Toolchains: Ac6 MCU GCC Configurations:		<ul> <li>Show ST Discovery boards</li> <li>Show ST EVAL boards</li> <li>Show ST NUCLEO boards</li> <li>Show custom boards</li> </ul>	
<ul> <li>Debug</li> <li>Release</li> </ul>	Select all Deselect all	Series :       STM32F4         Board :       NUCLEO-F411RE         Create a new custom board       Remove this custom board	•
	Advanced settings	MCU     STM32F411RETx       Core     ARM Cortex-M4       Package     LQFP64       Memory     RAMEXEC - size : 0x20000	4 III +
Use "Advanced settings" button to edit project's properties. Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on prope	rty pages.		
Reck Next Sector Finish	Cancel	? < Back Next > Finish Cancel	



You should then select your evaluation board (select first the family to filter out unneeded references); you can also manage custom boards here.

# Creating a program (3/4)

C Project	
Select the project structure and firmware	
🔿 No firmware	
Standard Peripheral Library (StdPeriph)	
Hardware Abstraction Layer (Cube HAL)	
③ Firmware "STM32Cube_FW_F4_V1.4.0" has been found.	
Download target firmware	
Extract all firmware in separate folder 🕦	
Add low level drivers in the project	
As sources (1)	
As static external libraries     (1)	
Additional drivers :	
STemWin	
STM32_Audio	
STM32_USB_Device_Library	
STM32_USB_Host_Library	
Additional utilities and third-party utilities:	
FatFs	
FreeRTOS	
LibJPEG	
LwIP	
PolarSSL	
You may have to make manual ajustement for third party utilities	
? Sack Next > Finish	Cancel

- You will have to select the firmware components you need in your project
  - First you can choose which kind of firmware to use (if any)
  - Only available firmwares will be proposed.
- If you choose to import firmware for your board
  - You may have to download the firmware
    - You will have to accept the firmware license agreement
    - Some firmware may have a more complex installer...
  - You have the choice to import the firmware either:
    - As separate library projects (recommended)
    - Directly in the application project
  - CMSIS, peripherals and utilities are automatically imported
  - You may choose to import more libraries
    - However you may then have to make manual configuration and adjustments



# Creating a program (3/4)

C/C++ - mx411/src/main.c - Eclips	ie In the second		
File Edit Source Refactor Nav	igate Search Project Run Window Help		
<b>☆ - 1</b> (; <b>2</b>   ≫ <b>- (</b> , - (	ਛੇ   🔌   🗇 ▾ 🛱 ▾ 🕑 ▾   🐎 ▾ 💽 ▾ 🚱 ▾ 🥵 ▾ 😂 😕 💋 ▾   🥒 🌮 🗉 🖬   ½ ▾ 🖓 ▾ 🤝 <	> ▼ ⇔ ▼	Quick Access
Project Explorer       Build            Project Explorer          Project Explorer            Project Explorer	<pre>Mi i i i i i i i i i i i i i i i i i i</pre>		Quick Access       Image: C/C++         Image: C/C++       Image: C/C++         Image: C/C+++       Image: C/C++         Image: C/C+++       Image: C/C+++      <
<ul> <li>Cproject</li> <li>.cproject</li> <li>.project</li> <li>LinkerScript.Id</li> <li>STM32Cube_FW_F4_V1.3.0</li> </ul>	94 95 /* Insert delay 100 ms */ 96 HAL_Delay(100); 97 } ♥ Problems ② Tasks ♀ Console ※  Properties  Search CDT Build Console [mx411] text data bss dec hex filename 3724 1100 1588 6412 190c mx411.elf	- - - - - - - - - - - - - - - - - - -	
<	18:07:04 Build Finished (took 8s.150ms)		•



Then clicking "Build" will build your project; then you are ready to debug it.





### **Debugging - Fast track**





### Debugging - Custom track (1/3)





### Debugging - Custom track (2/3)

Debug Configurations		×
Create, manage, and run cont	figurations	Ť
New launch configuration Ac6 STM32 Debugging C C/C++ Application C C/C++ Attach to Applic C C/C++ Postmortem Del C C/C++ Remote Applicat C GDB Hardware Debuggi Launch Group Filter matched 7 of 7 items	<ul> <li>Configure launch settings from this dialog:</li> <li>  • Press the 'New' button to create a configuration of the selected type.  </li> <li>  • Press the 'Duplicate' button to copy the selected configuration.  </li> <li>  • Press the 'Delete' button to remove the selected configuration.  </li> <li>  • Press the 'Filter' button to configure filtering options.  </li> <li>  • Edit or view an existing configuration by selecting it.  </li> <li>  Configure launch perspective settings from the 'Perspectives' preference page.</li></ul>	
?	Debug	Close

life.augmented

### Debugging - Custom track (3/3)

Debug Configurations		Debug Configurations	×
Create, manage, and run co	onfigurations	Create, manage, and run configurations	\$
type filter text ▲ C STM32 Debuggin ▲ test_STM32F0_Dir C C/C++ Application C C/C++ Attach to Appl C C/C++ Remote Appl C GDB Hardware Debu ► Launch Group	Name: test_STM32F0_Discovery Debug	Image and Symbols         Image and Symbols <t< td=""><td></td></t<>	
Filter matched 8 of 8 items	Apply Revert	Filter matched 8 of 8 items     Apply     Rev	/ert
?	Debug Close		lose



Debug - test_STM32F0_Discovery/src/main.c - Eclipse					• ×
File Edit Source Refactor Navigate Search Project Run Window Help					
: C* ▼ 🗒 📾   📾   🕸 ▼ 💽 ▼ 💁 ▼ ! X 🕪 !! 🔳 🙌 32 👁: i> =5, IX   😂 🗁 🔗 ▼   .	୬ ≫ [ h = P = * ↔ ↔ + ⇒	- ▼   - ⊂ Qu	iick Access	📄 🖹   🗟 C/C++ (	参 Debug
🐞 Debug 🛛 🧏 🎲 🌄 🗖 🗖	🗱 🕬= Variables 🔀 💁 Breakpoints	1111 Registers 🗰 I/C	Registers 🛛 🛋 M	odules	
🔺 😜 test_STM32F0_Discovery Debug [Ac6 STM32 Debugging]			R.	🍂 🖂 🖉 🗙 🍇 📑	ो 🛃 🗸
⊿ 🙀 test_STM32F0_Discovery.elf	Name	Type		Value	
Thread [1] (Suspended : User Request)	(×)= ii	uint32_t		4294967295	
main() at main.c:bb 0x80003/e	GPIO_InitStructure	GPIO_InitTypeDef		{}	
adb					
					*
	4				
🗈 main.c 🔀 💼 main.c 🛛 🔂 main.c			🗄 Outline 🖾		
**		*		😭 🕞 🖧 😿 💊	# ~
*/			💾 stm32	2f0xx.h	A
			stm32	2f0_discovery.h	
unt32 t ii = 0; coro Little PG			stm32	2f0xx_rcc.h	
delo_initistructure;			# LED E	2f0xx_gpio.h	
/* TODO - Add your application code here */			# LED1	o.c.	=
Systick_contig(4800); /* 0.1 ms = 100us it clock trequency 12 MHz */			# LED2		
SystemCoreClockUpdate();			# KEY_F	PORT	
<pre>ii = SystemCoreClock; /* This is a way to read the System core clock */ ii = 0:</pre>			# KEY		
		-	<ul> <li>timer</li> <li>timer</li> </ul>	Flag: uint32_t	
4		4		nag . unito_t	÷
😑 Console 🗱 🔎 Tasks 🖹 Problems 🕥 Executables 📋 Memory		■ × %			
test STM32F0 Discovery Debug (Ac6 STM32 Debugging) adb					
					*
Tomponany hoodknoint 1 main () at /coc/main c:66					
66 uint32 t ii = 0;					=
-					-
					Þ
			1		



Debug - test_STM32F0_Discovery/src/main.c - Eclipse	A DECKE AND DECK			
File Edit Source Refactor Navigate Search Project	Run Window Help			
📫 🕈 🗒 💼   🎰   🏇 🕶 💽 🕶 💁 🖬	<b>- N</b> 3. 9. 11 ið 5. 21 😕	🕒 🛷 ㅋ 📝 🌣 🖗 ㅋ 🖓 ㅋ 🏷 🤇	Ģ ▼ ⇒ ▼   🖻 🛛 Qui	ck Access 👔 🗄 🖬 C/C++ 🕸 Debug
🏇 Debug 🖾	🧏 🕅 İ⇒	▽ □ 🔲 🗱 Variables 💁 Breakpoi	ints 🐰 Registers 🖬 I/O Reg	isters 🖾 🛋 Modules 🛛 🖓 🗖 🗖
a 🕥 test STM32F0 Discovery Debug (Ac6 STM32 Debuggin	al			
▲ if test_STM32F0_Discovery.elf		(i) Double-click on regist	ter to fetch value	Su -
Thread [1] (Suspended : Step)		Register	Hex value	Binary value
main() at main.c:70 0x8000382		⊿		
openocd		⊿ តិថា USART1		
j gdb			0x000000	00000_0_00000_00000_0_0_0_0_0_0_0_0_0_0_
			[bit 27] 0x0	0
			bits 25-211 0x00	00000 -
		<		4
🔂 main.c 🔀 🔂 main.c 🛛 🔂 main.c 🔂 main.c				🗄 Outline 🖾 👘 🗖
GPIO_InitTypeDef GPIO_InitStructur	e;		-	🝃 🕞 🎼 😿 😼 🖷 🗰 🏹
Expression	Type	Value		stm32f0xx.h
SvsTick Config(4800):	ture GPIO InitTypeDef	{}		stm32f0_discovery.h
(x)= GPIO Pin	uint32 t	134219575		stm32f0xx_rcc.h
SystemCoreClockUpdate() (X)= GPIO_Mod	e GPIOMode_TypeDe	f 255 🕏		stm32f0xx_gpio.h
ii = 0:		i i i i i i i i i i i i i i i i i i i		# LED_PORT
Name : GPIO_Pin				# LED1
/* GPIOA-C Periph clock Details:1342	19575			# KEV DORT
RCC_AHBPeriphClockCmd(R Default:1342	19575			# KEY
RCC_AHBPeriphClockCmd(R Hex:0x800073	1			timer: uint32 t
Pinamu 10000	000000000011100110111		P a P	timerFlag : uint8_t
🔄 📃 Console 🧔 Tasks   Problems 🜔 Executables 📋 M	emory 🖾			🕆 📼 🔍 🔻 👪 🔛 🗰 🗸 🖓 🖬
Monitors 🕂 💥 🔆	&GPIO_InitStructure : 0x20001FEC <he< td=""><td>ex&gt; 🛛 🕂 New Renderings</td><td></td><td></td></he<>	ex> 🛛 🕂 New Renderings		
♦ &GPIO_InitStructure	Address 0 - 3 4 - 7	8 - B C - F		*
	20001FE0 54080008 0000	0000 ED000008 37070008		
	20001FF0 FFFFFFF 0000	0000 FFFFFFF FB010008		
	20002000 0000000 0000	00000000 0000000 0000		
	20002010 0000000 0000	<u>aaaa aaaaaaaa aaaaaaaa</u>		<b>T</b>
		Writable	Smart Insert 67 : 37	1



😄 De	Debug - test_STM32F0_Discovery/src/main.c - Eclipse								
File	Edit Source Refactor Navigate Search Project Run Window	Help							
1	$ \exists \bullet   \exists \bullet   \exists \bullet   \exists \bullet   \exists \bullet   \forall \bullet \bullet   \bullet   \bullet   \bullet   \bullet   \bullet   \bullet   \bullet  $								
	🕪= Variables 💩 Breakpoints 👭 Registers 🐹 🗰 I/O Registers 🛋	Modules	🏝 🏘 🖂 📑 🖻 🎽 🖓 🖶 🗗						
*	Name	Value	Description	₽					
	General Registers		General Purpose and FPU Register Group						
8	01 888	536872000							
E	888 r1	0							
-	888 r2	12							
<u>~</u>	888 r3	0							
- 22	3139 r4	-1							
$\mathbf{O}$	888 r5	-1							
0	818 r6	-1							
	3131 r7	536879080							
8	8181 r8	-1							
	888 r9	-1							
٣	3889 r10	-1							
	1000 r11	-1							
	1989 r12	-1							
	1000 sp	0x20001fe8							
	1919 Ir	134218235							
	888 pc	0x8000382 < main+10>							
	lili xPSR	1627389952							
	lili msp	0x20001fe8	3						
	1000 psp	0xffffffc							
	IIII primask	0							
	888 basepri	0							
	1111 faultmask	0							
	iiii control	0							
	<pre>Name : xPSR Hex:0x61000000 Decimal:1627389952 Octal:014100000000 Binary:11000010000000000000000000000 Default:1627389952</pre>								



E	dit Source Refactor Navigate Search	Broject Rup	Window Help				
-	an source Relactor Navigate Search	Project Run				1	
	님 🖻 🗅   🖬 ! 🍄 🕶 🖸 🕶 💁 🕶 🦄	<ul><li>N</li></ul>	3. O . : i> ≂, 3. (2 ⊖ A ▼ A \$ )	2 <b>▼</b> {2 <b>▼</b> *	$\Rightarrow \Leftrightarrow \bullet \bullet \bullet$	1	Quick Access 腔目睑 C/C++   黎 I
(	×)= Variables 🔍 Breakpoints 👬 Registers	I/O Registers	🔀 🛋 Modules				
	(i) Double-click on register to fetch value						\$
	Register	Hex value	Binary value	Reset value	Access	Address	Description
	a 🗁 USART						Universal synchronous asynchronous receiver tran
	⊿ mm USART1					0x40013800	Universal synchronous asynchronous receiver tran
	▲ 1919 CR1	0x00000000	00000_0_00000_00000_0_0_0_0_0_0_0_0_0_0_	0x00000000	READ-WRITE	0x40013800	Control register 1
	EOBIE [bit 27]	0x0	0				End of Block interrupt enable
	at RTOIE [bit 26]	0x0	0				Receiver timeout interrupt enable
	DEAT [bits 25-21]	0x00	00000				Driver Enable assertion time
	DEDT [bits 20-16]	0x00	00000				Driver Enable deassertion time
	OVER8 [bit 15]	0x0	0				Oversampling mode
	an CMIE [bit 14]	0x0	0				Character match interrupt enable
	MME [bit 13]	0x0	0				Mute mode enable
	• M [bit 12]	0x0	0				Word length
	WAKE [bit 11]	0x0	0				Receiver wakeup method
	a PCE [bit 10]	0x0	0				Parity control enable
	PS [bit 9]	0x0	0				Parity selection
	a PEIE [bit 8]	0x0	0				PE interrupt enable
	at TXEIE [bit 7]	0x0	0				interrupt enable
	TCIE [bit 6]	0x0	0				Transmission complete interrupt enable
	RXNEIE [bit 5]	0x0	0				RXNE interrupt enable
	IDLEIE [bit 4]	0x0	0				IDLE interrupt enable
	at TE [bit 3]	0x0	0				Transmitter enable
	IN RE [bit 2]	0x0	0				Receiver enable
	UESM [bit 1]	0x0	0				USART enable in Stop mode
	UE [bit 0]	0x0	0				USART enable
	▶ 1818 CR2	0x00000000	0000 0000 0 00 0 0 0 0 0 0 0 0 0 0 0 0 0	0x00000000	READ-WRITE	0x40013804	Control register 2
	▶ 1919 CR3	0x00000000	000000000 00 000 00 0 0 0 0 0 0 0 0 0 0	0x00000000	READ-WRITE	0x40013808	Control register 3
	▷ 1919 BRR	0x00000000	000000000000000000000000000000000000000	0x00000000	READ-WRITE	0x4001380C	Baud rate register
	⊳ lili GTPR	0x00000000	000000000000000000000000000000000000000	0x00000000	READ-WRITE	0x40013810	Guard time and prescaler register
	▶ 1919 RTOR	0x00000000	00000000 000000000000000000000000000000	0x00000000	READ-WRITE	0x40013814	Receiver timeout register
	> 1919 ROR	0x00000000	000000000000000000000000000000000000000	0x00000000	READ-WRITE	0x40013818	Request register
	▶ 1919 ISR	0x000000C0		0x000000C0	READ-ONLY	0x4001381C	Interrupt & status register
	<						4





# The OpenSTM32.org community site



#### OpenSTM32.org

Open STM32 Community The STM32 Systems Resource							
Home 🔻 Wiki 👻 Blogs 👻	FAQs - Forums - Documentation - Service - Log in - Q	in: Entire Site 🔻 Advanced: 🔲					
Menu Home System Workbench for	HomePage	Last-Modified Blogs 1. System Workbench for STM32					
STM32 Contact Us My user details My login details	Welcome to the STM32 Community Welcome to the STM32 Community site. The goal of this site is to create a knowledge hub for everyone who is interested in building STM32-based embedded systems to participate and collaborate together.	Newest FAQs 1. System Workbench for STM32 - Project creation 2. System Workbench for STM32					
Wiki  Blogs Forums File Galleries FAQs	This is also the place to find "System Workbench for STM32 - Bare Metal Edition" the free Integrated Development Environment for STM32 microprocessors. <b>Registration</b> Accessing OpenSTM32 org is free, but you need to be logged in to have access to some parts of the site, like the	<ol> <li>System Workbench for STM32 - Debugging</li> <li>OpenSTM32</li> <li>System Workbench for STM32 - Installation</li> </ol>					
	System Workbench for STM32 documentation and download instructions. You should either Log in or register on OpenSTM32.org. To be able to download and test System Workbench for STM32, you should request access to System Workbench as beta tester by sending a mail to the webmaster explaining why you would like to became a beta tester.	Newest Forum Posts 1. Re: Target Firmware by tarek.azhari, 17:07 2. Re: Target Firmware by PPR38, 14:23 3. System Workbench and					





- The OpenSTM32.org website (<u>http://www.openstm32.org</u>) is the focal point for STM32 embedded system development with System Workbench for STM32
- This is the place where to download System Workbench for STM32
  - It is a fully free functional development tool
    - No size limitation
    - No time limitation
  - The only limit is your imagination!
- You will also find there:
  - Documentation for System Workbench for STM32
  - Forums to discuss STM32 development
  - FAQs and Blogs with
    - insight in STM32 development,
    - tips and tricks on System Workbench for STM32 use





- You are welcome to participate in the OpenSTM32 adventure
  - Registration is free
  - Once registered you have access to all the content of the community:
    - Browse Wiki pages, Blogs and FAQs
    - Read on-line documentation
    - Post questions on the Forums or answer other members questions
    - And, of course, download System Workbench for STM32
  - Gaining the status of "Author" to be able to submit content in the Blogs and Wiki sections of the site is simple:
    - You just have to send a mail to a site admin explaining what kind of content you would like to submit on the web site
- The OpenSTM32.org Website is live now, with the Windows-based version of System Workbench for STM32
  - The Linux version is currently in the last beta phase (Release Candidate)
  - The MacOS/X version will be ready Q2 2015







#### System Workbench for STM32 Preliminary Roadmap

Release 1.0 (available)

- This release is the first official release
  - It supports all chips and evaluation boards available at this date
  - It supports firmware at the StdPeriph and HAL format wherever appropriate
  - It runs exclusively on Windows

#### Release 1.1 (expected end of Q1 2015)

- This release will be a minor release
  - It will support new devices from the STM32 product line
  - It will support Linux as development environment
    - We test and will support it on Ubuntu 12.04 and 14.04 LTS releases



#### System Workbench for STM32 *Preliminary Roadmap* Release 1.2 (expected Q2 2015)

- This release will be a minor release
  - It will support new devices from the STM32 product line
  - It will support MacOS/X as development environment
    - We test and will support it on OS/X 10.10 Yosemite

Release 2.0 (expected Q3 2015)

• This release is expected to be a major release



## **Regular Updates**

- Independently from scheduled updates, System Workbench for STM32 will be regularly updated
  - To correct bugs
  - To provide updated versions of some external tools (like GCC)
  - To enhance user experience
  - To support newly released STM32 chips
- These releases will be available through the standard Eclipse update mechanism







