

T.O.M.A.S Team





- Now, to complete our task, we have to
 - Switch to SW4STM32 for some software modification
 - Compile the code with added new features
 - Run the code on NUCLEO-L476RG board using SW4STM32







Our goals for this session

✓ How to migrate the project between two different microcontrollers:

- ✓ Importing STM32CubeMX project
- ✓ Running automatic migration to new platform (STM32L476RGT6 MCU)
- ✓ Tuning project in STM32CubeMX:
 - ✓ Updating clocks
 - ✓ Re-configuring peripherals
 - ✓ Generating project for new MCU

□Tuning project in SW4STM32:

Importing modified project to the same workspace
 Apply necessary modifications to generated sources
 Compile and run project on the new platform





Concept of the system migration to NUCLEO-L476RG



Importing L4_DAC_ADC example

Import newly generated L4 DAC ADC p into a SW4STM32 workspace (a new one of previously used).

C/C++ - L4_DAC_ADC/Src/main.c - Eclipse

Project Explorer 🛛

🐸 L4 Blinkv

🔺 😂 L4_DAC_ADC

Includes

Drivers

🖻 😂 Inc

life.auamented

File Edit Source Refactor Navigate Search Project 📑 🕶 🔚 🐚 | 🥸 🕶 🐔 🕶 🔜 l 🔯 🗃 🖝 😂 🕶 🕞 🕶 🎯

🖻 main.c 🖾

1 /**

ĺ	Workspace Launcher				
ADC project ew one or any	Select a workspace Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session. Workspace: C:_Work_Tools_lab ▼ Browse				
n <u>P</u> roject <u>R</u> un <u>W</u> indow	<u>H</u> elp				
$\overrightarrow{\mathbf{C}} \bullet \overrightarrow{\mathbf{C}} \bullet \mathbf{$					
×	OK Cancel				
*					
*****	· * * * * * * * * * * * * * * * * * * *				
<pre>* File Name * Description ************************************</pre>	: main.c : Main program body ************************************				

* COPYRIGHT(c) 2016 STMicroelectronics

workspace selecting active project is done by click on it (right button on mouse)

Import the project into the workspace 1/3 SW4STM32

Follow the below steps to import L4_DAC_ADC project into empty workspace:



life.auamented



6

It is possible to import multiple projects into a single workspace

Import the project into the workspace 2/3 SW4STM32

Cancel

Select L4_DAC_ADC project (previously generated by STM32CubeMX)

Import Projects Select a directory to search for existing Eclipse projects.	
Select project location (as configured in STM32CubeMX)	Browse Browse Folder r Folder folde
Options Search for nested projects Copy projects into workspace Hide projects that already exist in the workspace	> _Design Consulting > _Local support, phone, mail > _Seminar > .metadata > L4_Blinky L4_DAC_ADC
Working sets Working sets:	▼ Select

Import					
Import Projects Import Projects Select a directory to search for existing Eclipse projects. Import Projects					
 Select root directory: Select archive file: Projects: 	C:_Work_Seminar\L4_DAC_ADC	Browse Browse			
Id_DAC_ADC (C:_ Options Search for nested pro Copy projects into wo Hide projects that alr	Work_Seminar\L4_DAC_ADC) jects prkspace eady exist in the workspace	Select All Deselect All Refresh			
Working sets Add project to work Working sets:	ing sets	Select			
?	< Back Next > Finish	Cancel			

SW4STM32





Let's add some code to our generated template to make it working





Make the L4_DAC_ADC project running adding some code





Modifying the code adding own variables - task





Modifying the code adding own variables - solution

Tasks:

- 1. Add data buffer to be used by DAC to generate signal (sine wave example on the right available within **sine.h** file)
- 2. Declare data buffer for ADC to store measured data





life.auamentec

Modifying the code

13

start the peripherals configured within STM32CubeMX - task

Tasks (in main.c file):

- 1. Start ADC calibration
- 2. Start ADC to work with DMA on declared ADC buffer
- 3. Start DAC to work with DMA on declared DAC buffer
- Start Channel4 of Timer2 to work in "Output Compare without output" mode (trigger of ADC on Capture Compare event on channel 2 and for DAC on update)

To simplify coding process you can use code assistant by pressing **Ctrl+SPACE**





Modifying the code

start the peripherals configured within STM32CubeMX - solution

Tasks:

- 1. Start ADC calibration (necessary delay is implemented in dedicated HAL function)
- 2. Start ADC to work with DMA on declared ADC buffer
- 3. Start DAC to work with DMA on declared DAC buffer
- 4. Start Channel4 of Timer2 to work in "Output Compare without output" mode (trigger of ADC on Capture Compare event on channel 2 and for DAC on update)

```
/* USER CODE BEGIN 2 */
HAL_ADCEx_Calibration_Start(&hadc1,ADC_SINGLE_ENDED);
HAL_ADC_Start_DMA(&hadc1,(uint32_t *)adcbuf, ADCBUFSIZE);
HAL_DAC_Start_DMA(&hdac1,DAC_CHANNEL_1,(uint32_t *)dacbuf,DACBUFSIZE,DAC_ALIGN_12B_R);
HAL_TIM_OC_Start(&htim2, TIM_CHANNEL_2);
/* USER CODE END 2 */
```





Let's check whether the application is working correctly

- Configure the toolchain if necessary (parallel build, C99 dialect)
- Compile the code
- Start debug session





Useful project settings in SW4STM32 C dialect and parallel build





life.auamented

Build the project in SW4STM32

- To build the project either press Ctrl+B or click Make All icon
- In case of multiple compilation errors, re-run Indexing of the project
- After proper build there are information about code/data space usage in Console window displayed



16:02:57 Build Finished (took 1s.750ms)





Configure the debug session in SW4STM32 for single project in the workspace

🖨 De

C

?

- Before running debug session this is necessary to configure it for current project
- In case, there is a single project in the workspace, this is enough to click the "bug" icon and:
 - 1. Select "Ac6 STM32 C/C++ Application" line and click OK
 - 2. In case the project was generated on existing/defined board (like NUCLEO-L476RG in our example) debug will run automatically
 - 3. Otherwise it is necessary to configure debug device (STLinkV2-1 in our case) and debug interface (SWD in our case) and click OK
- Next step would be to run the debug session (see the next slide)

Debug As	
elect a way to debug 'L4_BLinky':	
Ac6 STM32 C/C++ Application	🗳 No board found : Define new board or select an e 💻 💴
Local C/C++ Application	No board has been found for this project. Before launch debug, a board has to be defined. Please create a new board or use an existing one. Define new board Use existing board
Description Debug with the Ac6 STM32 Debugging	Enter new board name : L4_DAC_ADC Select board chip
	Select series : STM32L4
	Select mcu : STM32L476RGTx
	Select board :
Cancel	Debug device : ST-LinkV2-1
	Debug interface : SWD -
	Ok Cancel





Configure the debug session in SW4STM32 for multiple projects in the workspace 1/2





Run the debug session in SW4STM32 for multiple projects in the workspace 2/2

- Connect Nucleo board with miniUSB cable (ST-Link)
- In case of the projects generated for ST board, there should be selected board configuration script which specifies debug device and its interface (you can check it in Debugger tab)
- Debug perspective will be run (please select Yes in the information window)
- This is enough just to click a "bug" icon to enter debug session next time.



Debug Configurations

Create,	manage,	and	run	configurations
---------	---------	-----	-----	----------------

l î i X □ i · · ·	Name: L4_Blinky Debug			
type filter text	🗈 Mair 🏇 Debugger 🕞 Startup 🔲 Common 🧤 Source			
 Ac6 STM32 Debugging L4_Blinky Debug C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application C/C++ Unit GDB Hardware Debugging Launch Group 	GDB Setup GDB Command: \${openstm32_compiler_path}\arm-none-eabi-gdb Browse OpenOCD Setup OpenOCD Command: "\${openstm32_openocd_path}\openocd.exe" Browse OpenOCD Options : Port number: 3333			
	💿 Use default script 🔘 Use local script 🔘 Manual spec			
	Configuration Script: nucleo_I476rg.cfg			
Filter matched 9 of 16 items	Revert			
?	3 Debug			



Run the debug session in SW4STM32

for multiple projects in the workspace, but no board specification 21

- Connect Nucleo board with miniUSB cable (ST-Link)
- Under **Debugger** tab select debug device (ST LinkV2-1 for Nucleo ones) and debug interface (SWD)
- Click Apply and then Debug

- Debug perspective will be run (please select Yes in the information window)
- This is enough just to click a "bug" icon to enter debug session next time.

L4_Blinky Debug	
🕒 🐜 👘 Debugger 🔪 🕨 Startup 🔲 Common 🦤 Source	
GDB Setup	
GDB Command:	
{openstm32_compiler_path}\arm-none-eabi-gdb Browse Variables	
OpenOCD Setup	
OpenOCD Command:	
"\${openstm32_openocd_path}\openocd.exe" Browse Variables	Confirm Perspective Switch
OpenOCD Options :	This kind of launch is configured to open the Debug perspective when it suspends.
Port number: 3333	This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.
© Use default script ◎ Use local script ◎ Manual spec	Do you want to open this perspective now?
Configuratic 3 pt: Browse	Remember my decision
Debug device: ST-LinkV2-1	4 Yes No
Debug interface: SWD -	



Debug session perspective

watching the variables

- This is possible to monitor CPU registers, peripherals registers and variables during debug session when we pause the code execution (no live view is possible for the time being).
- To add variable to be monitored select it, press right mouse button and select "Add Watch Expression". It will appear in Expressions tab.
- Values which have changed since previous project pause will be presented on yellow background





Debug session perspective

watching the registers content

- This is possible to monitor CPU registers, peripherals registers and variables during debug session when we pause the code execution (no live view is possible for the time being).
- To add peripheral register to watch select it, press right mouse button and select "Activate".
 Peripheral icon and its registers names will be highlighted in green and will contain "caught" values on next debug pause.
- Values which have changed since previous project pause will be highlighted in red.





Handling the debug session SW4STM32



- 1. Skip all breakpoints
- 2. Run/resume
- 3. Suspend
- 4. Terminate debug session
- 5. Disconnect from the target
- 6. Step into
- 7. Step Over

life.auamentec

8. Step Return



24

windows configuration in debug perspective

Let's run the debug session and see some effects

- Add adcbuf[] as a new item to be monitored
- After a while (at least 6.4 seconds to fill whole **adcbuf**[] buffer) pause the debug session and analyze the **adcbuf**[] content

Results in debug session monitoring adcbuf[] buffer

To add new variable to be monitored in debug session it is needed to:

- 1. Right click on the variable name
- 2. Select "Add Watch Expression"
- New position in "Expressions" tab will appear.
- 4. This is not live debug, therefore to check the values it is necessary to pause the session
- 5. Updated positions are highlighted in yellow

1	🖨 Debug -	L4_DAC_AD	C/ ain.c - Eclipse	at Deschart 21	000 9 0 0	
4	Undo	Ctrl+Z	act. Navigate Search	Project Run Window He	p	
	Revert File	Chilly C	▶ II 🔳 💦 곳 곳t i⇒	🗟 📰 🔯 🕶 💽 🖛 💁 🕶 🖄	🕒 🖉 💌 🍠 💱 🖢 👻 🗟 👻 😓 🗸	→ ▼
	Save	Ctri+S			2	Quick Access
	Open Type Hierarchy	Open Declaration F3 Open Type Hierarchy F4				
	Open Call Hierarchy	Ctrl+Alt+H	i ≫ № i→ ▽□□	(x)= Variables 💁 Breakpoints	🛠 Expressions 🛛 🛲 I/O Registers	🖾 🕫 🕞 🕂 🗶 💥 📑 🗹 🔻 🗖
	Quick Outline	Ctrl+O	Ac6 STM32 Debugging]	Expression	Туре	Value 🔺
	Quick Type Hierarchy Explore Macro Expansion	Ctrl+I Ctrl+#	nlf	🔺 🥬 adcbuf	uint16 t [32]	0x20000134 <adcbuf></adcbuf>
	Toggle Source/Header	Ctrl+Tab	Suspended : Signal : SIGI	(X)= adcbuf[0]	uint16 t	54
	Open With	•	main.c:106 0x8002c60	(×)= adcbuf[1]	uint16 t	2041
	Show In	Alt+Shift+W		(×)= adcbuf[2]	uint16 t	2821 5
	Cut Copy	Ctrl+X Ctrl+C	Workbench/plugins/fr.ac	(×)= adcbuf[3]	uint16 t	3484
	Paste	Ctrl+V		(×)= adcbuf[4]	uint16 t	3937
	Quick Fix	Ctrl+1	-	6.5		
	Source	Alt+Shift+S ►		•	III	4
	Refactor	•	p. •	•		4
	Declarations References	•	4xx hal rcc c	[1] proc[42000] threadGroup[i1]	adb[1] proc[42000] OStbread[Securitine Securitine
	Search Text	•	;+(),	[1],proc[12000].tinetidoroup[11	,,gab(1),proc(12000),000tinead(
	Make Targets	•	άι(), «			
	Resource Configurations	tions •	DE BEGIN 2 */		ſ	bade1 + ADC HandleTypeDef =
Step Into Selection Ctrl+F5 Calibration Start(&hadc1,ADC SI			&hadc1,ADC_SINGLE_ENDED);	hdma_ada1 : DMA_HandlaTura	
2	Move to Line	Ctrl+R	art_DMA(&hadc1,(uir	nt32_t *) <mark>adcbuf</mark> , ADCBUF	SIZE);	 hdma_ddC1 - DMA_mandleTyp hdac1 - DAC_HandleTypoDof
2	Resume at Line		<pre>art_DMA(&hdac1,DAC_CHANNEL_1,(uint32_t *)dacbuf ,DACBUFSIZE,DAC_ALIGN_1</pre>			 Indaci : DAC_HandleTypeDer hdma das sh1 : DMA Handlei
×+y	Add Watch Expression	•				 httm2::TIM_HandleTypeDef
	Debug As	•	DE END Z */			
	Profile As	•		111	4	< •
	Profiling Tools Team	*	🕄 Problems 🔘 Execut	ables	🔳 🗙 🔆 🛙	🗼 🔝 🖻 📑 💽 🛃 🚽 🗖 👻 🗖 🗖
	Compare With	•	M32 Debugging1 C:/Ac6/	/SystemWorkbench/plugins/fr.a	c6.mcu.externaltools.arm-none.win32 1.7.	0.201602121829/tools/compiler/bin/arm-none-
Replace With				-,,,,,,,,,,,,,		, , , , , , , , , , , ,
	Preferences				1	
5,	Remove from Context Ctrl+Al	It+Shift+Down				

What have we learnt?

✓ How to migrate the project between two different microcontrollers:

- ✓ Importing STM32CubeMX project
- ✓ Running automatic migration to new platform (STM32L476RGT6 MCU)
- ✓ Tuning project in STM32CubeMX:
 - ✓ Updating clocks
 - ✓ Re-configuring peripherals
 - ✓ Generating project for new MCU

✓ Tuning project in SW4STM32:

- ✓ Importing modified project to the same workspace
- ✓ Apply necessary modifications to generated sources
- ✓ Compile and run project on the new platform

www.st.com/mcu

