

T.O.M.A.S Team







In specific cases, especially in real-time applications (i.e. motor control), this is tough (or even risky) to use the regular debug session.

There are few complementary debug tools and methods which can be used instead:

- STMStudio application
- Use of printf() redirected to
 - One of the serial interfaces (we loose serial port & IO lines)
 - Single Wire Output (SWO) pin (no extra resources used except PB3 pin)





- Practice a bit with STMStudio monitoring variables and creating expressions
- Practice a bit with printf implementation using USART and any terminal application
- Practice a bit with printf implementation using SWO channel and STLink Utility application





Complementary debug tools Using STMStudio



Checklist 6

Using STMStudio

- □ Preparation of the hardware and output files to be analyzed
- Importing global variables
- Different viewers usage
- Creating expressions from the selected variables
- □Using build-in functions on the selected variables
- Log text files generation





Let's run STMStudio and follow the next steps using L4_DAC_ADC project



STMStudio overview

Application allows non-intrusive sampling and real time visualization of user's variables while the application is running (it doesn't affect application timing).



STMStudio - procedure

- In order to start monitoring, the following procedure should be performed:
 - Embedded application should be programmed into the MCU
 - MCU should be connected to the computer (where STMStudio is run) via STLink or Rlink programmers over SWD/JTAG for STM32 and SWIM for STM8 devices
 - The same executable file (.elf, .out, .axf) should be opened by STMStudio and desired global variable should be imported.
 - Acquisition should be started by *Run→Start* or clicking "**Play**" button
 - Acquisition can be stopped anytime by *Run→Stop* or clicking "**Stop**" button



For detailed information refer to dedicated user manual UM1025

🚬 🗋 堤 📮 ST-Link SWD 🔹 🖻 🕨							
Show workspace <<			VarViewer	·1			
isplay Variables Write Variables	250						
Display Variables settings	240	240					
Name Address Type	Color 230						
J J J J J	220						
	210						
	200 -						
	190						
Tuno	180						
Name Expression Type	Color 170						
	160						
Name AcqVar Function Scope	Color						
	150						
	150						
	150						
Name Function Type Descript	150 140 0 <th></th> <th></th> <th></th> <th></th> <th></th>						
Name Function Type Descript	Color 150 140 140 Color 130 120 140						
Name Function Type Descript	Color 150						
Name Function Type Descript	Color 150 Color 140 Color 120 110 100						
Name Function Type Descript iewers settings General Display Point Viewer VarViewer1 as Curve	Color 150 Color 120 110 100 90						
Name Function Type Descript iewers settings General Display VarViewer1 as Curve VarViewer1	Color 150 Color 120 110 100 90 80						
Name Function Type Descript iewers settings	Color 150 140 140 120 120 110 100 90 80 70						
Name Function Type Descript iewers settings Display Point Viewer VarViewer1 as Curve /arViewer1 List of Variables No Variable	Color 150 140 140 120 120 110 100 90 80 70 60						
Name Function Type Descript iewers settings Image: Setting	Color 150 Color 140 Color 120 100 100 90 50 50						
Name Function Type Descript iewers settings Display VarViewer1 as Curve Point Viewer1 VarViewer1 as Curve VarViewer1 List of Variables No Variable No Variable	Color 150 140 140 120 120 110 100 90 80 70 60 50 40						
Name Function Type Descript iewers settings	Color 150 Color 140 Color 120 110 100 90 80 70 80 50 40 30						
Name Function Type Descript iewers settings Display Point Viewer VarViewer1 as Curve VarViewer1 List of Variables No Variable	Color 140 Color 140 140 120 120 110 100 90 80 70 60 50 40 30 20						
Name Function Type Descript Tewers settings General Display Point Viewer VarViewer1 as Curve VarViewer1 List of Variables No Variable Delete Delete All	Color 150 140 120 120 120 110 100 80 70 80 50 40 30 20 10						





Before the next step we need to perform some slight modification of our L4_DAC_ADC code.





L4_DAC_ADC project tuning

- As within STMStudio we can monitor each particular variable, not complete tables it would be more useful to change our **adcbuf[]** table to single variable (table with size 1).
- Please change in *main.c* file ADCBUFSIZE from 32 to 1.
- **Recompile** the code and **program** the board (by starting the debug session).
- Please **STOP** the debug session afterwards to release STLink for STMStudio application.

Now we are ready to monitor our application behavior (adcbuf[0] values change in particular) using STMStudio application



Importing global variables

	illi S	TM Studio New co	onfig	
1	File	Run Views Optio	ons l	Help
	8	Open	L	k SWD 🔻 📰 🕨
		New Project	Ν	pace <<
	暍	Save Project As	Α	
	E	Save	S	ables
(2)		Import variables	Ι	
		Recent Config	•	
			In	port variables from executable

- Application can import global variables from executable files (.elf, .out, .axf) generated by any tool for STM32 and STM8 MCUs.
- To select proper file and variables, use File →Import variables shortcut or dedicated button
- To select particular components within the table, check "Expand table elements" option.

🗔 Import varia	bles from execu	utable			×
File selection Executable file C:_Work_Seminar\L4_DAC_ADC\Debug\L4_DAC_ADC.elf Image: Store executable path relatively to the user settings file Image: Expand table elements (this may take several seconds more)					
Variables					Selection
Add variables to	the display variab	les table		-	Select all
Show symbols o	ontaining		Match case		Unselect all
File	Name	Address	Туре		Import
/Drivers/CMS	AHBPrescTabl	0x8002e48	unsigned 8-bit		(4)
/Drivers/CMS	APBPrescTabl	0x8002e58	unsigned 8-bit		
/Drivers/CMS	MSIRangeTabl	0x8002e18	unsigned 32		Import scaled variable in expression
/Drivers/CMS	SystemCoreCl	0x200000	unsigned 32	Ξ	
/Drivers/STM	uwTick	0x200000	unsigned 32		Linear expression A*variable + B:
/Src/main.c	adcbuf[0]	0x200001	unsigned 16		Import with A and B as constants
/Src/main.c	dacbuf[0]	0x8002e60	unsigned 16		
/Src/main.c	hadc1.Instance	0x200000	unsigned 32		Import with A and B as expressions
/Src/main.c	hadc1.Init.Cloc	0x200000	unsigned 32		

Task: Import adcbuf[0] and dacbuf[0] variables from L4_DAC_ADC application

Adding variables to viewers 13

After selection of variables from executable file there are few options possible:

- Use of drag-and-drop to add them directly to the Viewers settings window
- Creation of an expression based on the variable(s) described on next slides
- Use one of predefined functions (i.e. max, min, ...) on the selected variable – described on next slides

Display Variabl	Write V	ariables						
Display Variables settings								
Name		Address		Ту	Туре		lor	
D adcbuf[0]		0x2000	0134	uns	igned 16-bit			
D dacbuf[0]		0x8002	e60	uns	igned 16-bit)	
				_			_	
Name		Expres	sion	Ту	ре	Co	lor	
Name	Ac	qVar Function		n	n Scope		Color	
Name	Name Fur		nction Type		Descripti		lor	
Viewers setting	js							
General	٢D	isplay						
Point Viewer	Vā	arViewer	1 as	Curve •				
VarViewer1	Hexadecimal							
	List of Variables							
	a	adcbuf[0]						
	da	acbuf[0]						
							Ξ	





STMStudio viewers

- There are 3 possible viewers available in STMStudio: **bar graph**, **curve** and **table**
- Switching among the viewers can be done anytime, even during acquisition without necessity to stop it
- All selected data are automatically stored in text log file that can be analyzed afterwards

Show workspace SS	VarViewer1	STM Studio New config*					
	Val Viewei 1	File Run Views Options Help					
ay Variables 8,250		🖿 🔜 🖳 📰 ST-Link SWD 👻 🔳					
lay Variables settings 7,750		Show workenson	A VarVieword	STM Studio New config*			
ame Address Type Color 7,500			Valviewell	File Run Views Ontions Help			
dcbuf[0] 0x2000010c unsigned 16-bit 7,250 acbuf[0] 0x8002152 unsigned 16-bit 7,000		Display Variables Write Variables	8,250				
6,750		Display Variables settings	7,750				
6,500		Name Address Type Color	7.500	Show workspace <<		VarViewer1	
6,250		adcbuf[0] 0x2000010c unsigned 16-bit	7,250	Display Variables Write Variables	Variable Name	Address/Evonesion Pea	ad Value
ama Exprancian Turna Calar 5,750			7,000	Display Variables settings	adcbuf[0]	0x2000010c 802	B
5,500			8,500	Name Address Type Color	dacbuf[0]	0x8002152 0	-
5,250 -			6.250	adcbuf[0] 0x2000010c unsigned 16-bit			
ame AcoVar Function Scope Color 4,750			6,000				
4,500 - 4,500 -		Name Expression Type Color	5,750				
- 4,250			5,250				
ame Function Type Descript, Color	8,028		5,000				
3,750		Name AcqVar Function Scope Color	4,750	Name Expression Type Color			
3,250			4,500				
3,000 ·			× 4.000				
eneral Display 2,750		Name Function Type Descript Color	3.750	Name AcqVar Function Scope Color			
t Viewer VarViewer1 as Bar Graph ▼ 2,500			3,500				
Viewer1 Hexadecimal 2,000			3,250				
1,750 -		Viewers settings	2,750	Name Function Type Descript Color			
adchuff0] 1,500		General Display	2,500				
dacbuf[0] 1,000		VarViewer1	2.250				
= 750		Hexadecimal	2,000	Viewers settings			
500 -		List of Variables	1,500	General Display			
		adcbuf[0]	1.250	Point Viewer VarViewer1 as Table			
Delete Delete All	101		1,000	VarViewer1 Hexadecimal			
Value Range	30-		760	List of Variables			
lower Value 0.0	Variables		250	adcbuf[0]			
		Delete Delete All		dacbuf[0]			
		Value Pange	10,100 40,000 40,000 40,000 40,000 40,000 40,000 40,000 40,000 40,000 40,000 40,000 40,000 40,000 40,000 40,000				
		lower Value 0.0	adcbuf[0] dacbuf[0]				
				Delete All			



Adding expressions

It is possible to define own expressions based on the monitored variables

Task: Add a new expression adcbuf[0]-dacbuf[0] 3			Expression editor		
3	ask: dd a new expression	on adcbuf[0]-dacbuf[0]	newExpr_0 adcbuf[0]-dacbuf[0]	•	Functions Category: Logical <
Image: Structure Structur	Image: Constraint of the sector of the se	STM Studio New config* File Run Views Options Help Structure Show workspace Show workspace Display Variables Write Variables Display Variables settings Name Address Type Color Address Type Color	3 <pre>expression </pre> <pre>Acquisition variables and Statistical v </pre> <pre>adcbuf[0] </pre> <pre>dacbuf[0] </pre>	СК	> = & <-> (1) (-) (+) <= >= <-< >> IF(,,) Operators Usage: (-arg1) Description: Returns -arg1 (negate arg1) Can

life.auamentec

Adding predefined function

It is possible to use some predefined functions on the monitored variables

Task: Select predefined Min() function for adcbuf[0] variable



	Display Variables	Write \	/ariables					
	Display Variables settings							
	Name	Addres	s	Туре	Color			
	D adcbuf[0]	0x2000	010c	unsigned 16-bit	<u> </u>			
		0,0002	152	unsigned 10-bi				
3	Name	Expres	sion	Туре	Color			
Select	D newExpr_0	adcbuf[0]-da	double				
Min								
	North	eaVar	Functio	n Scope	Color			
	Dadcbuf[0 ad	dcbuf[0]	Min	all				
			Min					
	Name F	unction	Max	Descript	Color			
		anoton	Std Dev	/				
				1				

Add expressions/function results to viewers

It is possible to add all defined expressions and functions to the viewers for monitoring (like single variables) by drag & drop mechanism

Task:

Add new values (expression and min() function) to the Viewers window start collecting data.

IIII STM Studio New config*								
File Run Views Options Help								
🖿 🖥 🖏 🖬 🦉 ST-Link SWD 🔻 🔳 📕								
Show workspace <<	VarViewer1							
Display Variables Write Variables	Variable Name	Address/Expression	Read Value					
Display Variables settings	adcbuf[0]	0x2000010c	8028					
Name Address Type Color	dacbuf[0]	0x8002152	0					
Dadcbuf[0] 0x2000010c unsigned 16-bit	adchuf[0] Min	Min	8028.0					
D dacbuf[0] 0x8002152 unsigned 16-bit	newSnr_0	Snr	0					
Name Expression Type Color D PewExpr_0 adcbuf[0]-da double Name AcqVar Function Scope Color D adcouf[0 adcbuf[0] Min all Name Function Type Descript Color D adcouf[0 adcbuf[0] Min all Name Function Type Descript Color D newSin G Snr unsigned Signal/N Viewers settings General Visplay Viewers Vat/viewer1 as Table Var/viewer1 Hextdecimal adcbuf[0] newSnr_0 Delete Delete Delete All								
Value Range								
lower Value 0.0								
upper Value 8430.0								
	1-							



STMStudio configuration

- Configuration of STMStudio is accessible by selecting *Options->Acquisition Settings* (or using end button)
- Main parameters which can be configured there are:
 - 1. Graphical refresh rate
 - 2. Acquisition rate
 - 3. Variable selections
 - 4. Location of the text log file
 - 5. Log file settings

E Acquisition Settings
Graphical Settings Image: Constraint of the set of the se
O Under-sample data for graphical display O Display all data (may delay the acquisition)
Variable selection • Acquire all variables • Acquire only variables used by viewers • Acquire only variables used by visible viewers
Acquisition Parameters Triggers
4 © Replay from file © Get data from target
Log file: C:_Work_Seminar\L4_DAC_ADC\log.txt
Log file settings Log file settings Log at graphical rate Log at graphical rate Append mode Log only variations Log only variations
Target application shared data
For snapshot variables: Address of snapshot trace header: 0x0 Record each 1 calls to DumpTrace
For direct variables: Synchronize with target application 0x0
Target communication
Init comm on each start Automatically stop the acquisition after 10 consecutive communication errors
Detect STM8 low power modes Automatically stop the acquisition after 40 consecutive detections in low power mode
OK Cancel





Using log file for data analysis

All selected data are stored automatically "runtime" in log text file which can be analyzed afterwards

📃 log.txt -	Notepad								
File Edit	Format V	iew Help							
Syntax y	version=	4							
Mon Jun	Mon Jun 06 10:12:30 2016								
Startin	g proces	s (2533)	261 cl	ocks per se	c)				
LogDowD	ata falc	~							
LOGRAWD	ata=1 a 15 0n_1	e							
D.	el.=T	adchuf	[0]	dachuf	01	InowEv	nr 01 Ladchuf[0] Minl		
D.	time(ms)	$(0x^{2})$	$2000010c^{-2}$	(0x8002	152 2)	ladchuf[0]-dachuf[0]		
D:	0.17	8028	0	8028.0	8028.0	0.0	: adebui [0] daebui [0]:		
D:	25.51	8028	ŏ	8028.0	8028.0	0.0			
D:	33.46	8028	Ō	8028.0	8028.0	0.0			
D:	41.50	8028	0	8028.0	8028.0	0.0			
D:	49.50	8028	0	8028.0	8028.0	0.0			
D:	73.50	8028	0	8028.0	8028.0	0.0			
D:	81.50	8028	0	8028.0	8028.0	0.0			
D:	89.50	8028	0	8028.0	8028.0	0.0			
D:	97.41	8028	0	8028.0	8028.0	0.0			
D:	121.36	8028	0	8028.0	8028.0	0.0			
D:	129.35	8028	0	8028.0	8028.0	0.0			
D:	15/.54	8028	0	8028.0	8028.0	0.0			
D:	160 25	8028 8028	0	8028.0	8028.0	0.0			
D.	177 25	0020 8028	0	8028.0	8028.0	0.0			
0.	T(1.33	0020	0	0020.0	0020.0	0.0			





- Practice a bit with STMStudio monitoring variables and creating expressions
- Practice a bit with printf implementation using USART and any terminal application
- Practice a bit with printf implementation using SWO channel and STLink Utility application







www.st.com/mcu

