## S32G-VNP-RDB3

## **REAL TIME DRIVER**

## **EXAMPLE ENABLEMENT GUIDE**

## PUBLIC

MARCH 2023



PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2023 NXP B.V.





## CONTENTS

- Hands on UART Real Time Driver example
- Hands on ETH Real Time Driver example
- Hands on CAN Real Time Driver example

## HARDWARE REQUIREMENT AND SOFTWARE INSTALLATION

#### Hardware Requirement

- S32G-VNP-RDB3
- S32 Debug Probe
- AD/DC power supply
- Serial port cable for UART example

#### **Software Installation**

- Install S32DS 3.5 according to <u>S32G-VNP-RDB3 Software Enablement Guide</u>
- Install SW32G\_RTD\_4.4\_4.0.0\_D2210(RTD) according to S32G-VNP-RDB3 Software Enablement Guide

# Hands on UART Example



#### PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2023 NXP B.V.

## HANDS ON UART: OBJECTIVE

- How to import the UART example into S32DS
- How to configure the clock of UART via S32DS
- How to configure the UART module via S32DS
- How to debug the UART example with S32 debug probe

## HANDS ON UART: IMPORT UART EXAMPLE PROJECT

Open S32 Design Studio, go to "File -> New -> S32DS Project From Example". Select "Linflexd\_Uart\_lp\_Example\_S32G399A\_M7" example, click on "Finish". The project should now be copied into the current workspace.



The "Linflexd\_Uart\_Ip\_Example\_S32G399A\_M7" example is a simple application which shows the usage of UART driver.



## HANDS ON UART: PIN CONFIGURATION

Open the **Pins configuration tool**. According to schematic of RDB3, configure pins routing. By default, this example already has corresponding pin routing configuration.

<ul> <li>Einflexd_Uart_lp_Exan</li> </ul>	New New	>	
> Project Settings	Go Into		
<ul> <li>Separation of the second second</li></ul>	Go Into Open in New Window Show In Show in Local Terminal Copy Copy Copy Copy Copy Copy Copy Copy	Alt+Shift+W > > Ctrl+C Delete Ctrl+V F2 >	
	<ul> <li>Export</li> <li>Build Project</li> <li>Clean Project</li> <li>Refresh</li> <li>Close Project</li> <li>Close Unrelated Projects</li> </ul>	F5	
A Dashboard ≅	Build Targets Index	>	= 0
* Project Creation	Build Configurations	>	
<ul> <li>S32DS Application Prc</li> <li>S32DS Library Project</li> </ul>	Build path Build Configurations Explorer	>	
<ul> <li>Build/Debug</li> <li>Build (All)</li> <li>Clean (All)</li> <li>Debug</li> </ul>	Attach/Detach SDKs Manage SDKs Migrate		
- Settings	Show in Remote Systems view		
<ul> <li>Project settings</li> <li>Build settings</li> <li>Debug settings</li> </ul>	<ul> <li>Run As</li> <li>Debug As</li> <li>Restore from Local History</li> </ul>	>	
* Miscellaneous	S32 Configuration Tools	>	Open Pins
<ul> <li>Getting Started</li> <li>Quick access</li> </ul>	Team Compare With Configure Source Validate	) ) ) )	Open Clocke     Open Pins     Open DCD     Open IVT     Open QuadSPI     Open DDR     Open DDR
	Properties	Alt+Enter	<ul> <li>Open eFUSE</li> <li>Import Configuration (*.m.</li> </ul>

≣ Pins 🖾	🗠 Peri	Vow.			Package [Pins Bot	ttom] 🛛	Expansio	n Header			⊎ € (* ■	
880	₩ Ш 🗕	<b>⊕</b>	¥ (2)	۶			23 22 21 20 19	18 17 16 15 14 13 12 11	10 9 8 7 6	5 4 3 2 1		
Pin	Pin name	Lab	bel	^								_
✓K12	VSS359					A				S32		
⊻L12	VDD451					e	821 822 821 820 919		010 00 00 07 OK		1 114.01	_
✓M12	VSS360					D	D23 D22 D21 D20 D10	C8 EP C8 C8 CN C0 C0 C1 D18 D17 D16 D15 D14 D15 D12 D11	DTO 09 DE 07 DE	All signals on	pin [012]:	
✓N12	VDD452					F	627 622 621 628 69 623 622 621 620 619	RN 20 20 20 20 20 20 20 20 20	F10 F9 F6 F7 F6	BOOT:rcon	,4	
✓P12	VSS361					а н	623 622 621 620 619 163 162 161 168 118	G18 G17 G15 G14 G13 G11 HB HD HB HB HD HD HD	G10 G9 G8 G7 G8	FTM_0:ext_	_clk,0	
⊻R12	VDD453					к	123 122 121 120 119	318 317 316 315 314 315 312 311	110 19 18 17 16	LINFlex_1:t	xd	
⊻T12	VDD_IO_A	<b>\</b>				L	120 122 121 120 119	LTIR LT7 LT6 LT5 LT6 LT3 LT2 LT1	L10 L9 L9 L7	PFE_MAC1	:rx_err	
<b>⊻</b> U12	PA_13					N	907 907 907 909 979 N22 N22 N21 N20 N19	NER NET NER NES NES NES NES NES	NTO NO NO NT NO	SIUL2_0:gp	pio,13	
⊻V12	VSS362					R	P23 P32 P31 P38 P38	PR FD PR PR PR PR PD PD PD	FE FE FE FE FE FE	SPI_0:sclk,0	)	
<b>⊻</b> W12	PB_00					т	00 00 01 08 78 U22 U22 U21 U20 U19		115 15 15 15 15 15			
Enable AA12	PC_10 <b>e pin ar</b> PA_14	id sel	lect	co	rrect signal	v w v	123 122 121 120 119 123 122 121 120 119 123 122 121 120 119	VID         VID <td>VID VS VS V7 VS</td> <td></td> <td>Done</td> <td></td>	VID VS VS V7 VS		Done	
✓AB12	VSS363					AA	ARES ARES ARES ARES ARES		. 2011 202 202 202 202 202	A		
AC12	VSS364			$\sim$		AC	ACD ACT ACT ACT		-		AC	
		<	>									
Routin	g Details											⊒ =
Pins Si	gnals 🔎 t	ype filte	er text	:								
Routing	) Details fo	r	2	Ð	8 ^ •							
# F	Peripheral	Signal	Ar	row	Routed pin/signal	Label	Identifier	Power group	Direction	Output Buffer	Open Drain	Input Buffer
W12 L	INFlex_1	rxd	<-		[W12] PB_00		n/a	VDD_IO_A (0V)	Input	Disabled	Disabled	Enabled
U12 L	INFlex_1	txd	->		[U12] PA_13		n/a	VDD_IO_A (0V)	Output	Enabled	Disabled	Disabled
Confi	aure pi	n's a	ttrib	ute	es				PU	BLIC	6	NP

## HANDS ON UART: CLOCK CONFIGURATION 1

Open the Clocks Diagram:

- Right-click the Project
- Select S32 Configuration Tools
- Select Open Clocks

in the second		New		
Proi		Go Into		
Src ⇒ ⊡ r Sinclu		Open in New Window Show In Show in Local Terminal	Alt+Shift+W>	
🖉 des(		Copy Delete Paste Rename Source Move	Ctrl+C Delete Ctrl+V F2 >	
	èa ∠i	Import Export		
	Ð	Build Project Clean Project Refresh Close Project Close Unrelated Project	F5	
		Build Targets Index Build Configurations	>	
		Build path Build Configurations Explorer SDKs Migrate	>	
	0	Show in Remote Systems view Run As Debug As Restore from Local History	>	
	*	S32 Configuration Tools	>	Open Pins
		Team Compare With Configure Source Validate	>	Open Clocks  Open Perip Open Clocks  Open DCD Open IVT Open QuadSPI
		Properties	Alt+Enter	Open DDR Import Configuration (* mex

PUBLIC

NO

## HANDS ON UART: CLOCK CONFIGURATION 2

Open the **Peripheral Clock View**, and double-click the Lin module. The **Clocks Diagram** will show the clock tree of the LinFlexD. The default clock configuration of UART is 48 MHZ which comes from FIRC directly.



## HANDS ON UART: UART CONFIGURATION 1

Open the Peripherals Diagram:

- Right-click the Project
- Select S32 Configuration Tools
- Select Open Peripherals



## HANDS ON UART: UART CONFIGURATION 2

The **Components** shows all drivers which used by this example, the **Linflexd\_Uart** Component includes the configuration of UART driver

Components 🕺 🦞 Peripherals		읍 Linflexd_U	art 🛙		° c
Components of Peripheruis		Linflexc	Uart Configuration [Drivers]		🗈 🕋 <
type filter text	O 11	Name Linfle	exd_Uart		Custom name [
		Mode LINF	LEXD UART Mode		
MCAL	0				Preset Custom
		Name Cont	igTimeSupport GeneralConfiguration Uart	tGlobalConfig	
Drivers	0	Name	LartGlobalConfig		Preset Custom V
0		Name	ourcooldecomig		
IntCtri_Ip	Siul2_Port	✓ UartC	hannel + × ^ ×		
		UartCha	nnel_0 Name	UartChannel_0	
			UartHwChannel	LinflexD_1	~
LIART module confi	iguration :		UartClockFunctionalGroupRef	BOARD_BootClockRUN	~
	guration.		<ul> <li>DetailModuleConfiguration</li> </ul>	n	
Configura Hort Ho	rdwara Channal		Name	DetailModuleConfiguration	
			Desire Baudrate	LINFLEXD_UART_BAUDRATE_9600	~
Ocationa Developt			Uart Asynchronous Method	LINFLEXD_UART_IP_USING_INTERRUPTS	~
- Configure Baudrat	le		✓ Tx DMA channel		
	vite a Electric sections		+		
- Configure Uart Pai	rity Enable option -		🗙 Rx DMA channel		
- Configure Llart Sto	on Rit Numbor.		+		
			Uart Parity Enable		
- Configure Hart M/c	ord Longth		Uart Parity Type	LINFLEXD_UART_IP_PARITY_EVEN	×
			Uart Stop Bit Number	LINFLEXD_UART_IP_TWO_STOP_BIT	~
		<	> Uart Word Length	LINFLEXD_UART_IP_8_BITS	×

## HANDS ON UART: UPDATE CODE

#### Generate code method:

1. Open the view of any configuration tool, like Pins, then click **Update Code** (ensure desired project is selected)

2. The Update Files window pops up. It shows the detailed update information. Click **OK** button.

3. The configuration .c and .h files will be generated in "generate" folder.



NC

## HANDS ON UART: APPLICATION CODE 1

#### Dissecting the main.c



## HANDS ON UART: APPLICATION CODE 2

Dissecting the main.c



## HANDS ON UART: BUILD AND DEBUG 1

#### Build the target :

- Right-click the Project
- Select Build Project
- Print Build information on Console window
- Linflexd\_Uart\_lp\_Example\_S32G399A\_M7.elf is generated







## HANDS ON UART: BUILD AND DEBUG 2

Go to debug configuration:

- Right-click the Project
- Select the Debug As
- Click on Debug Configurations

v 🕼	Linfle	xd	• •		
>	🐝 Bir	nari	C a		>
>	🔊 Inc	lud	GC	) Into	
>	😕 Pro	ojec	Op	pen in New Window	
>	🖉 RT	D	Sh	iow In	Alt+Shift+W>
2	bo	ard	Sh	ow in Local Terminal	>
2	ge ge	ner	Co	ру	Ctrl+C
5	e ge	ner	X De	elete	Delete
5	e ge		🗈 Pa:	ste	Ctrl+V
5	🗁 De	bu	So	urce	FZ >
>	🧭 inc	lud	Mo	ove	
	📕 de	scri	> Im	port	
	🖊 Lin	fle:	🗠 Ex	port	
			Bu	uild Project	
			Cle	ean Project	
			🐑 Re	fresh	F5
			Clo	ose Project	
			Clo	ose Unrelated Project	
			Bu	uild Targets	>
			Inc	dex	>
			Bu	uild Configurations	>
			Bu	uild path	>
			Bu	ild Configurations Explore	r
			SD	)Ks	
			Mi	igrate	
			Sh	ow in Remote Systems vie	w
			O Ru	in As	>
		- L	🎋 De	ebug As	>
			Re	store from Local History	
			S3	2 Configuration Tools	>

#### Debug configuration setting:

- Connect the S32 Debug probe with PC and RDB3
- Click on target project
- Select the target device and core as S32G399A\_M7\_0
- Select target S32 Debug Probe

3 🖻 🕫 🔚 🗶 🖻 🍸 👻	Name: Linflexd Uart Ip Example S32G399A M7 Debug RAM S32Debug		
type filter text  C (/C++ Application C (/C++ Remote Application E C/C++ Remote Application E Cclipse Application	Main * Debugger Startup * Source Common *** SVD Support * OS Awareness * Trace and Profile Hardware Device: S32G399A Core: M7_0	Select device	and cor
GDB Hardware Debugging     GDB PEMicro Interface Debugging     GDB SEGGER J-Link Debugging	Initialization script: \$(532DS_INITIALIZATION_SCRIPTS_DIR)/s32g3xx/s32g3xx_generic_bareboard.py Initial core	Browse V	ariables
C GDB SEGGER J-Link Debugging C Launch Group Launch Group (Deprecated) Launch Group for S32 Debugger C S32 Debugger C Linflexd Uart Jp_Example_S32G399A_M7_Debug_RAM_S32Debug S32 Debugger Frash Programmer VLAB Simulator Debugging	Debug Probe Connection         Interface:       S32 Debug Probe - US8         USB device:       00:04:9f:06:6d:1e - S32 Debug Probe         Interface:       S32 Debug Probe         Interface:       Target Communication Speed	Test connection	
	JTAG Speed (KHz): 16000 Timeout 30 s Delay after reset: 0 ms GDB Server Claunch server Server port number® 45000 Enable log		
	GDB Client Executable: \$(S32DS_GDB_ARM32_PY) Commands:	elect Browse V	ariables

NC

## HANDS ON UART : DEBUG AND RUN

Power on the RDB3, click on "Apply", then click on "Debug". The view will switch to the Debug Perspective, and you can use the controls to control the program flow.

	File Edit Source Refactor Navigate Search Project Configloois Run FreeRIOS Window Help
Debug Probe Connection	
Interface: S32 Debug Probe - USB	🏶 Debug 🕴 🎦 Project Explorer 👘 🗖 📴 main.c 🛱 🗟 Siul2_Port_lp.c
USB device: 00:04:9f:06:6d:1e - S32 Debug Probe	<pre>     Bin Windows Control Content Control Control Control Control Control Control Control</pre>
Revert Apply Debug Close	<pre>74 */ 75 Clock_Ip_Init(&amp;Mcu_aClockConfigPB[0]); 76 77 /* Interrupts controller initialization */ 78 IntCtrl_Ip_Init(&amp;IntCtrlConfig_0); 79 IntCtrl_Ip_ConfigIrqRouting(&amp;intRouteConfig); 80 81 /* Initialize pins */ 82 Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0); 83</pre>

NAME OF A DATA OF CALL D

## HANDS ON UART: TEST RESULT

Check the test result:

- Connect UART1 with PC and RDB3
- Open Serial terminal like Tera Term and configure the serial port
- Click on the Resume option in Debug view
- The Serial terminal will print messages
- Then input "Hello" in step 3
- UART1 will output "Hello World!" back





# Hands on ETH Example



#### PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2023 NXP B.V.

## HANDS ON ETH – OBJECTIVE

How to import the ETH example into S32DS
How to configure the clock of ETH via S32DS
How to configure the port of ETH via S32DS
How to use the ETH module to transmit/receive ETH frame
How to debug the ETH example using S32 debug probe

## HANDS ON ETH: IMPORT ETH EXAMPLE PROJECT

Open S32 Design Studio, go to "File -> New -> S32DS Project From Example". Select "Gmac\_lp\_InternalLoopback\_S32G399A\_M7 " example, then click on "Finish". The project is copied into the current workspace.



This "Gmac\_lp\_InternalLoopback\_S32G399A\_M7" example demonstrates the GMAC transmission and reception in internal loopback mode. The ETH frame is transmitted back directly through GMAC, and the frame will not be transmitted to PHY.



PUBLIC

## HANDS ON ETH : PORT CONFIGURATION

## Pins configuration setting:

- Right-click the Project
- Select S32 Configuration Tools
- Select Open Pins
- Configure pins to provide the external clock to Tx, Rx signals



Gmac\_lp\_InternalLoop

> 🥴 Project\_Settings

description.txt

Gmac\_Example\_DS 📑 Copy

> 🥶 src

> 6 include

Go Into

Show In

X Delete

Paste

Rename...

Export... Build Project

> Clean Project Refresh

**Close Project** 

Source Move...

Open in New Window

Show in Local Terminal

Alt+Shift+W>

Ctrl+C

Delete

Ctrl+V

F2

F5

NC

## HANDS ON ETH : CLOCK CONFIGURATION

Open the **Peripheral Clock View**, and double-click the GMAC0 module. The **Clocks Diagram** shows the clock tree of GMAC module

😭 Overview 🕝 Pe	eripheral Clock View	🛛 🖸 Code Preview	Register	s 🧮 Detail
Clock Name	Enable	Source	Divider	Frequency
FLEXCAN3_CL	< 🗹	FIRC_CLK	/1	48 MHz
FRAY0_CLK	$\checkmark$	FIRC_CLK	/2	24 MHz
FTIMER0_CLK	$\checkmark$	FIRC_CLK	/1	48 MHz
FTIMER1_CLK	$\checkmark$	FIRC_CLK	/1	48 MHz
GMAC0_RX_CL	K 🗹	GMAC_EXT_RX_REF	/1	25 MHz
GMAC0_TS_CL	к 🗹	FIRC_CLK	/1	48 MHz
GMAC0_TX_CL	K 🗹	PERIPH PHI5	/ 5	25 MHz
IIC0_CLK	$\checkmark$	XBAR_DIV3_CLK	/1	8 MHz
IIC1_CLK	$\checkmark$	XBAR_DIV3_CLK	/1	8 MHz
IIC2_CLK	$\checkmark$	XBAR_DIV3_CLK	/1	8 MHz
IIC3_CLK		XBAR_DIV3_CLK	/1	8 MHz
IIC4_CLK	$\checkmark$	XBAR_DIV3_CLK	/1	8 MHz
LBIST0_CLK	$\checkmark$	FIRC_CLK	/1	48 MHz
LBIST1_CLK	$\checkmark$	FIRC_CLK	/1	48 MHz
LBIST2_CLK	$\checkmark$	FIRC_CLK	/1	48 MHz
LBIST3_CLK		FIRC_CLK	/1	48 MHz
LBIST4_CLK	$\checkmark$	FIRC_CLK	/1	48 MHz
LBIST5_CLK	$\checkmark$	FIRC_CLK	/1	48 MHz
LBIST6_CLK	$\checkmark$	FIRC_CLK	/1	48 MHz
LBIST7_CLK		FIRC_CLK	/1	48 MHz



## HANDS ON ETH: ETH CONFIGURATION

Open the peripheral configuration:

- Right-click the Project
- Select S32 Configuration Tools
- Select Open Peripherals

Project Explorer 🛛					E 🕏 🏹 🕴 🗖 🖬	
<ul> <li>SGmac_lp_InternalLoop</li> <li>Includes</li> <li>Project_Settings</li> <li>Src</li> <li>Sinclude</li> </ul>		New Go Into Open in New Window Show In	> Alt+Shift+W >			
☑ description.txt ☑ Gmac_Example_DS	<ul> <li></li> <li><td>Show in Local Terminal Copy Delete Paste Rename Source Move</td><td>Ctrl+C Delete Ctrl+V F2</td><td></td><td></td><td></td></li></ul>	Show in Local Terminal Copy Delete Paste Rename Source Move	Ctrl+C Delete Ctrl+V F2			
	24 24	Import Export				
	Ł	Build Project Clean Project Refresh Close Project Close Unrelated Project	F5			
		Build Targets Index Build Configurations	>			
		Build path Build Configurations Explorer SDKs Migrate	>			
	0 *	Show in Remote Systems view Run As Debug As Restore from Local History	>			
		S32 Configuration Tools	>	۰	Open Pins	
		Team Compare With Configure Source Validate	>	🔹 🖗 III 🖗 🍝	Open Clocks Open Peripherals Open DCD Open IVT Open QuadSPI Open DDR	
	_	Properties	Alt+Enter		Import Configuration (*.mex) Manage SDK Components	

# Select Components to find out GMAC Driver and double-click

MCAL       Drivers       © Gmac       © osif_1       © Siul2_Port       © mix H       © mix H <th>- compo</th> <th>nents 🛛</th> <th>Peripherals</th> <th></th>	- compo	nents 🛛	Peripherals	
MCAL				• ti
Orivers     Image: Constraint     Image: Constraint <th></th> <th></th> <th>MCAL</th> <th>0</th>			MCAL	0
Image: Signed       Image: Signed Signe			Drivers	0
	<b>©</b> Gr	nac	<sup>O</sup> osif_1	Siul2_Port
Name         EthConfigSet           • EthConfigSet         • EthConfigSet           • EthConfigSet         • EthCurloadpenties           • EthConfigSet         • EthCurloadpenties           • EthCurloadpenties         • EthCurloadpenties	Name Gmac			Custom name
Name         EthConfigSet           • EthCtrlConfig         • • • • • • • • • • • • • • • • • • •	Name Gmac Mode GMAC Mode			Custom name Preset [Custom_
• EthCtriConfig         • EthCtriConfig_0           0         Name         EthCtriConfig_0           EthCtriFabileMiti	Name Gmac Mode GMAC Mode Name ConfigTimeSuppor	rt EthGeneral EthConfig	set	Custom name Preset Custom Preset Custom
0         Name         EthCtrlConfig_0           EthCtrlEnableMid	Name Gmac GMAC Mode Name ConfigTimeSuppor Name EthConfigSet	rt EthGeneral EthConfig	set	Custom name Preset [Custom Preset [Custom \
EthCtrlFnableMii	Name Gmac Mode GMAC Mode Name ConfigTimeSuppor Name EthConfigSet V EthCtrlConfig	rt EthGeneral EthConfigs	iet	Custom name Preset Custom Preset Custom N
Eth.ChrilhableRhinterupt Eth.ChrilhableRhinterupt Eth.ChrilhableRhinterupt Eth.ChrilhableRhinterupt Eth.ChrilhacLayerSype Eth.ChrilhacLayerSype REDUCED	Name Gmac Mode GMAC Mode Name ConfigTimeSuppor Name EthConfigSet V EthCtrlConfig 0 N	rt EthGeneral EthConfig5	EthCtriConfig_0	Custom name Preset Custom
EthCtrlEnableTxInterrupt  EthCtrlAndLayerType ETH_MAC_LAYER_TYPE_XGMII  EthCtrlMacLayerSubType Eth_MAC_LAYER_TYPE_XGMII  EthCtrlMacLayerSubType  Eth_MAC_LAYER_TYPE_XGMII  EthCtrlMacLayerSubType  Eth	Name Gmac Mode GMAC Mode Name ConfigTimeSuppor Name EthConfigSet V EthCtrlConfig 0 N	rt EthGeneral EthConfig5	EthCtriConfig_0	Custom name Preset [Custom Preset [Custom v
EthCtrlMacLayerType ETH_MAC_LAYER_TYPE_XGMII    EthCtrlMacLayerSubType  EEDUCED  EED	Name Gmac Mode GMAC Mode Name ConfigTimeSuppor Name EthConfigSet • EthCtriConfig 0 N Eth Eth Eth Eth Eth Eth Eth Eth	rt EthGeneral EthConfigS	Ett:CtriConfig_0	Custom name Preset [Custom Preset [Custom v
EthCtrlMacLayerSpee EthCtrlMacLayerSpeed ETH_MAC_LAYER_SPEED_100M	Name Gmac Mode GMAC Mode Name ConfigTimeSuppor Name EthConfigSet	rt EthGeneral EthConfig3	EthCtriConfig_0	Custom name Preset Custom v Preset Custom v
Eth.tmmacLayersuB1ype      REDUCED     V      Eth.CtrlMacLayerSpeed      ETH_MAC_LAVER_SPEED_100M     V      Eth.CtrlPhyAddress	Name Gmac Mode GMAC Mode Name ConfigTimeSuppor Name EthConfigSet	rt EthGeneral EthConfigS	EthCtriConfig_0	Custom name Preset Custom v Preset Custom v
	Name Gmac Mode GMAC Mode Name ConfigTimeSuppor Name EthConfigSet	rt EthGeneral EthConfigS	EthCtriConfig_0	Custom name Preset Custom v Preset Custom v
EthCtrlMacLayerSpeed     EthChrlMacLayerSpeed     Eth_MAC_LAYER_SPEED_100M     V	Name Gmac Mode GMAC Mode Name ConfigTimeSuppoo Name EthConfigSet EthCtrlConfig N R Eth EthCtrlConfig Eth	tt EthGeneral EthConfigt + EthGeneral EthConfigt ame thCtriEnableMii thCtriEnableMind thCtriEnableMind thCtriEnableMinterrupt thCtriEnableRinterrupt thCtriMacLayerType r EthCtriMacLayerSubTyp resource	EthCtriConfig_0	Custom name Preset Custom • Preset Custom •
EthLtriMacLayerspeed     Eth_MacLayerspeed     V      EthLtriMacLayerspeed     V      V	Name Gmac Mode GMAC Mode Name ConfigTimeSuppo Name EthConfigSet • EthCtriConfig 0 N EthCtriConfig 0 R Eth Eth Eth Eth Eth Eth Eth Eth	rt EthGeneral EthConfigs + EthGeneral EthConfigs ame thCtrEnableMii thCtrEnableRinterrupt thCtrEnableRinterrupt thCtrIIdax thCtrIId	EthCtriConfig_0  EthCtriConfig_0  0  Eth_MAC_LAYER_TYPE_XGMII e	Custom name
Eth_MAC_LAYER.SPEED_100M     V	Name Gmac Mode GMAC Mode Name ConfigTimeSuppo Name EthConfigSet • EthCtriConfig 0 N Eth Eth Eth Eth Eth Eth Eth Eth	rt EthGeneral EthConfigs + EthGeneral EthConfigs ame thCtrIEnableMin thCtrIEnableMind thCtrIEnableRinterrupt thCtrIEnableEXinterrupt thCtrIIEnableEXinterrupt thCtrIIAa thCtrIMacLayerSype r EthCtrIMacLayerSype IEDUCED	EthCtriConfig_0  EthCtriConfig_0  Eth[Mac_Laver_Type_XGMII  e	Custom name Preset Custom Preset Custom V
✓ EthCtrlPhyAddress	Name Gmac Mode GMAC Mode Name ConfigTimeSuppo PathConfigSet EthCCnfigSet EthCCnfig 0 N EthCCnfig 0 N Eth EthConfig 0 N Eth EthConfig 0 N Eth EthConfig 0 N Eth Eth Eth Eth Eth Eth Eth Eth	rt EthGeneral EthConfigs + EthGeneral EthConfigs ame thCtrIEnableMii thCtrIEnableMid thCtrIEnableMid thCtrIEnableTxInterrupt thCtrIIIdx thCtrIIIdx thCtrIIIdx thCtrIIAcLayerSpeed > thCtrIMacLayerSpeed	EthCtriConfig_0  EthCtriConfig_0  EthLMAC_LAYER_TYPE_XGMII e	Custom name Preset Custom Preset Custom V V V
EthCtrIPhyAddress	Name Gmac Mode GMAC Mode Name ConfigTimeSuppo VethConfigSet ConfigTimeSuppo Name EthConfigSet ConfigTimeSuppo Name Reference EthConfig Reference Et	rt EthGeneral EthConfigs	EthCtriConfig_0  EthCtr	Custom name Preset Custom Preset Custom V V V
	Name Gmac Mode GMAC Mode Name ConfigTimeSuppo Vame EthConfigSet ConfigTimeSuppo Config	tt EthGeneral EthConfigt t t t t t t t t t t t t t t t t t t	EthCtriConfig_0	Custom name Preset Custom Preset Custom V

## HANDS ON ETH: UPDATE CODE

#### Generate code method:

1. Open the view of any configuration tool, like Pins, then click **Update Code** (ensure desired project is selected)

2. The Update Files window pops up. It shows the detailed update information. Click **OK** button.

3.The configuration .c and .h files will be generated in "generate" folder.



## HANDS ON ETH: APPLICATION CODE 1

#### Dissecting the main.c

#### int main(void)

Gmac\_Ip\_TxOptionsType txOptions = {TRUE, GMAC\_CRC\_AND\_PAD\_INSERTION, GMAC\_CHECKSUM\_INSERTION\_DISABLE}; Gmac\_Ip\_BufferType txBuffer = {0}; Gmac\_Ip\_TxInfoType txInfo; Gmac\_Ip\_RxInfoType rxInfo; Gmac\_Ip\_StatusType status; uint8 macAddr[6U] = {0U}; uint8 i; uint8 j = 0U; boolean result = TRUE;

OsIf\_Init(NULL\_PTR);

```
Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0);
```

Clock\_Ip\_Init(&Mcu\_aClockConfigPB[0]);

Gmac\_Ip\_Init(INST\_GMAC\_0, &Gmac\_0\_ConfigPB\_BOARD\_INITPERIPHERALS);

Initialize pins to provide the external clock for GMAC

Enable GMAC controller, initialize Tx and Rx buffer via the function Gmac\_lp\_Init

/\* Setup the frame with the Mac address and size \*/
Gmac\_Ip\_GetMacAddr(INST\_GMAC\_0, macAddr);

```
/* Request a buffer of at least 64 bytes */
txBuffer.length = 64U;
if ((GMAC_STATUS_SUCCESS != Gmac_Ip_GetTxBuff(INST_GMAC_0, 0U, &txBuffer, NULL_PTR)) || (txBuffer.length < 64U))
ł
    result = FALSE;
}
for (i = 0U; i < 12U; i++)</pre>
    *((uint8 *)(&txBuffer.data[0U] + i)) = macAddr[0 + j];
    if (j < 5U)
    {
        j++;
    else
    ſ
        j = 0U;
    }
}
```

Apply for Txbuffer via the function Gmac\_lp\_GetTxBuff and initialize transmission buffer

## HANDS ON ETH: APPLICATION CODE 2

#### Dissecting the main.c

```
/* Payload = Frame - (DstAddr + SrcAddr + EtherType/Length + FCS) */
*((uint32 *)(txBuffer.data + 13U)) = 64U - (6U + 6U + 2U + 4U);
/* Send the ETH frame */
                             /* Don't count FCS, because it is automatically inserted by the controller in this example */
txBuffer.length = 64U - 4U;
if (GMAC_STATUS_SUCCESS != Gmac_Ip_SendFrame(INST_GMAC_0, 0U, &txBuffer, &txOptions))
                                                                                                                             Transmit frame via Gmac_lp_SendFrame
    result = FALSE;
/* Wait for the frame to be transmitted */
do {
    status = Gmac Ip GetTransmitStatus(INST GMAC 0, 0U, &txBuffer, &txInfo);
} while (status == GMAC_STATUS_BUSY);
/* Check the frame status */
if ((GMAC_STATUS_SUCCESS != status) || (0U != txInfo.errMask))
    result = FALSE;
3
/* Wait for the frame to be received */
                                                                             Verify frame is transmitted or received
do {
    status = Gmac_Ip_ReadFrame(INST_GMAC_0, 0U, &rxBuffer, &rxInfo);
} while (status == GMAC_STATUS_RX_QUEUE_EMPTY);
/* Check the frame status */
if ((GMAC_STATUS_SUCCESS != status) || (0U != rxInfo.errMask))
    result = FALSE;
```

Gmac\_Ip\_ProvideRxBuff(INST\_GMAC\_0, 0U, &rxBuffer);

Gmac\_Ip\_DisableController(INST\_GMAC\_0);

## HANDS ON ETH: BUILD AND DEBUG 1

#### Build target Project:

- Right-click the Project
- Build Project

v

- The console print build information
- Gmac\_lp\_InternalLoopback\_S32G399A\_M7.elf is created

100	Gmac In Int			
>	Includes		New Go Into	>
>	Project_Se			
>	🐸 RTD		Open in New Window	
>	🐸 board		Show In	Alt+Shift+W >
	🐸 generate		Show in Local Terminal	>
>	😕 generate/		Сору	Ctrl+C
>	🐸 generate/	×	Delete	Delete
v	🐸 src	Ē	Paste	Ctrl+V
	> 🖻 main.c	2	Rename	F2
>	💋 include		Source	>
	🖉 descriptio		Move	
	M Gmac_Exa	è	Import	
		2	Export	
			Build Project	
			Clean Project	
		8	Refresh	F5
			Close Project	
			Close Unrelated Project	
			Build Targets	>



- Gmac\_lp\_InternalLoopback\_S32G399A\_M7: Debug\_RAM
  - v 🖑 Rinaries
    - Gmac\_lp\_InternalLoopback\_S32G399A\_M7.elf [arm/le]
  - > 🔊 Includes
  - > B Project\_Settings
  - > 🤒 RTD
  - > 😂 board
  - > 🤒 generate
  - > 🐸 generate/include
  - > 🐸 generate/src
  - 🗸 🐸 src
    - > 🖻 main.c
  - > 🗁 Debug\_RAM
  - > 🗭 include
  - description.txt
  - Gmac\_Example\_DS\_003.mex

🖺 Problems 🦪 Tasks 📮 Console 🛛 🗔 Properties 🛷 Search 🤫 Progress 🌞 Debug 🍃 Call Hierarchy

CDT Build Console [Gmac\_lp\_InternalLoopback\_S32G399A\_M7]

Building target: Gmac\_Ip\_InternalLoopback\_S32G399A\_M7.elf Invoking: Standard S32DS C Linker arm-none-eabi-gcc -o "Gmac\_Ip\_InternalLoopback\_S32G399A\_M7.elf" "@Gmac\_Ip\_Internal

c:/nxp/s32ds.3.4/s32ds/build\_tools/gcc\_v9.2/gcc-9.2-arm32-eabi/bin/../lib/gcc/arm-Finished building target: Gmac\_Ip\_InternalLoopback\_S32G399A\_M7.elf

Invoking: Standard S32DS Print Size

arm-none-eabi-size --format=berkeley Gmac\_Ip\_InternalLoopback\_S32G399A\_M7.elf
text data bss dec hex filename
291884 0 12289 304173 4a42d Gmac\_Ip\_InternalLoopback\_S32G399A\_M7.elf
Finished building: Gmac\_Ip\_InternalLoopback\_S32G399A\_M7.siz

10:41:15 Build Finished. 0 errors, 0 warnings. (took 17s.527ms)

## HANDS ON ETH: BUILD AND DEBUG 2

## Go to debug configuration:

- Right-click the Project
- Select the Debug As
- Click on Debug Configurations

#### Debug configuration setting:

- Connect the S32 Debug probe with PC and RDB3
- Click on target project

🗋 🙆 😭

type filter text

C/C++ /

C/C++

Eclipse

🖸 GDB Ha

🔛 GDB PE

GDB SE

🖡 Launch

Launch

🛼 Launch

👐 S32 Del

🚥 Gma

- S32 Del

··· VLAB Si

- Select the target device and core as S32G399A\_M7\_0
- Select target S32 Debug Probe



	Name: Gmac_lp_InternalLoc	pback_S32G399A_M7_Debug_RAM_S32Debug		
	📄 Main 🗱 Debugger 🕨	Startup 💱 Source 🔲 Common 🛲 SVD Support 🥮 OS Awareness 💺 Trac	ce and Profile	
Application Remote Application Application rdware Debugging GGR J-Link Debugging GGR J-Link Debugging Group (Deprecated) Group for S32 Debugger jugger c. Ip_InternalLoopback_S32G399A_M7_Debug_RAM_S32Debug jugger Flash Programmer mulator Debugging	Mail       ✓ Debugger         Hardware       Device: S32G399A         Initialization script:       §[S32]         ✓ Initial core       Debug Probe Connection         Debug Probe Connection       Interface:         USB device:       00:04:9ft         Hostname or IP:       Target Communication Sp         TAG Speed (KHz):       1600         Delay after reset:       0         GDB Server       ✓         Launch server       Server port number# 4500	Core: M7_0 DS_INITIALIZATION_SCRIPTS_DIR//s32g3xx/s32g3xx_generic_bareboard.py  ug Probe - USB  eeed Timeout: 30 s ms	Select de Browse	vice and core



## HANDS ON ETH: DEBUG AND RUN

Power on the RDB3, click on "Apply", then click on "Debug". The view will switch to the Debug Perspective, and you can use the controls to control the program flow.

				· · · · · · · · · · · · · · · · · · ·
				🕸 Debug 🕴 🎦 Project Explorer 💦 🗖 🔹 main.c 😫
				□ 張   题 1 + 8 52
☑ Initial core				Gmac_lp_InternalLoopback_S32G399A_M7_Debug_RAM_S32/Debugger]     LOCAL_FUNCTIONS
Debug Probe C	Connection			✓ P Thread #1 1 [core: 0] (Suspended : Breakpoint 55 56
Interface:	S32 Debug Probe - USB 🗸			■ main() at main.c66 0x340001e0     57⊕ /*====================================
USB device:	00:04:9f:06:6d:1e - S32 Debug Probe	Refresh	Test connection	S32 Debugger
Hostname or IP	×			wa seminosung 0007 ← 1 61 * @brief Transmit & receive in internal <u>loopback</u> mode 62 * @details
Target Commun JTAG Speed (Kł Delay after n GDB Server Launch serve Server port nun Enable log GDB Client	nication Speed Hz): 16000 Timeout: 30 s reset: 0 ms er nber: 45000	Revert	Apply	<pre>63 */ 64*int main(void) 65 { 66 [ Gmac.Ip_IxOptionsType TxOptions = {TRUE, GMAC_CRC_AND_PAD_INSERTION, GMAC_CHECKSUM_INSERTION_DISABLE}; 67 Gmac.Ip_BufferType TxBuffer = {0}; 68 Gmac_Ip_SufferType TxBuffer = {0}; 69 Gmac_Ip_RkInfotype TxInfo; 70 Gmac_Ip_StatusType Status; 71 uint8 MacAddr[60] = {00}; 72 uint8 j = 00; 73 boolean Result = TRUE; 74 Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0); 75 Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0); 76 Clock_Ip_Init(8Mcu_aClockConfigP8[0]); 77 Gmac_Ip_Init(INST_GMAC_0, &amp;Gmac_0_ConfigP8_BOARD_INITPERIPHERALS); 78 /* Setup the frame with the Mac address and size */ 79 Gmac_Ip_GetMacAddr(INST_GMAC_0, MacAddr); 70 Gmac_Ip_GetMacAddr(INST_GMAC_0, MacAddr); 71 Gmac_Ip_SetMacAddr(INST_GMAC_0, MacAddr); 72 Gmac_Ip_SetMacAddr(INST_GMAC_0, MacAddr); 73 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 74 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 75 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 76 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 77 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 78 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 79 Siul2_Port_INITPERIPHERALS); 70 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 71 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 72 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 73 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 74 Gmac_Ip_SetMacAddr(INST_SMAC_0, MacAddr); 75 Gmac_Ip_SMAC_0, MacAddr); 75 Gmac_</pre>

## HANDS ON ETH: TEST RESULT 1

In this project, the Ethernet module works in internal loopback mode. Add a breakpoint to the last line of the main function, then click on the Resume option. The received ETH frame can be watched from rxBuffer.

WorkspaceS32DS.3.4 - Gmac_lp_InternalLoc	workspaceS32DS.3.4 - Gmac_Ip_InternalLoopback_S32G399A_M7/src/main.c - S32 Design Studio for S32 Platform — 🗇 🗙									
File Edit Source Refactor Navigate	<b>xc</b> h Project ConfigTools Run Window Help									
! <u> </u>	<b>▲ [ ▶ ]</b> 11 ■ & 3. ③   <b>i</b> → ≂, ∞   <b>¿</b> ]      +   ☆ + O + <b>4</b> +   <b>2</b>  2  2  4  2  +    →    →   → +   11		۹	📴 🖥 🖷 🗤 🕴 👼 🚍 🕬	*					
🎋 Debug 🛛 🎦 Project Explorer 🛛 🗖 🗖	🖻 main.c 🖻 main.c 🗈 Siul2_Port_Ip.c 🖻 Siul2_Port_I 🔡 linker_ram 🕓 startup_cm7.s 🖬 main.c 🖾 鞜 👘 👘 👘	(x)= Var 🛛 💁 Bre 🖗	🕸 Ex 🛋 Mo 🔤 Per	. 🕮 Ar 🔤 Em 🔀 Per 🧮 🕻	3					
<ul> <li>E % I 1 → 8</li> <li>✓ III Gmac_lp_InternalLoopback_S32G399A</li> <li>✓ B Gmac_lp_InternalLoopback_S32G399</li> </ul>	<pre>120 Status = Gmac_Ip_GetIransmitstatus(INSI_GMAC_0, 00, &amp;IxButter, &amp;IxInto); 121 } while (Status == GMAC_STATUS_BUSY); 122 123 /* Check the frame status */</pre>	Name > 🕖 TxOptions	Type Gmac Ip TxOptionsType	&	000					
✓ P Thread #1 1 [core: 0] (Suspender	124 if ((GMAC STATUS SUCCESS != Status)    (0U != TxInfo.ErrMask))	> 🥭 TxBuffer	Gmac_lp_BufferType	{}						
≡ main() at main.c:146 0x34000		🗸 🥟 RxBuffer	Gmac_lp_BufferType	{}						
C:/NXP/S32DS.3.4/S32DS/tools/gdl	126 Result = FALSE;	> 🔹 Data	uint8 *	0x34501d00 <gmac_0_rxring_0_d< td=""><td>Dat</td></gmac_0_rxring_0_d<>	Dat					
斗 S32 Debugger		⇔= Length	uint16	64						
📕 Semihosting	120 129 /* Wait for the frame to be received */	> 🥭 TxInfo	Gmac_lp_TxInfoType	{}						
	130 do {	> 🥭 RxInfo	Gmac_lp_RxInfoType	{}						
	<pre>131 Status = Gmac_Ip_ReadFrame(INST_GMAC_0, 0U, &amp;RxBuffer, &amp;RxInfo);</pre>	⇔= Status	Gmac_lp_StatusType	<optimized out=""></optimized>						
	<pre>132 } while (Status == GMAC_STATUS_RX_QUEUE_EMPTY);</pre>	> 🥭 MacAddr	uint8 [6]	0x34401f9c						
		(×)= j	uint8	<optimized out=""></optimized>						
	134 /* Check the frame status */	(×)= j	uint8	<optimized out=""></optimized>						
	135 1 ((GMAC_STATUS_SUCCESS != STATUS)    (OU != KXINTO.EPPMask))	⇔= Result	boolean	<optimized out=""></optimized>						
	137 Result = FALSE:									
	138 }	<			>					
	139									
	<pre>140 Gmac_Ip_ProvideRxBuff(INST_GMAC_0, 0U, &amp;RxBuffer);</pre>	🖳 Outline 🕅 🚥 Disa	sembly	□ <u>14, № %</u> ● # 8 □ [						
	141 142 - Gmar In DisableController(INST GMAC 0):		SSETTIDTY							
	143	Gmac_ip.n								
	144 Exit_Example(Result);	Siui2_Port_ip.n								
	145	Glask la b								
	2146 return 0;	Clock_ip.n	a h							
< >	140 History and Angeling	<ul> <li>cneck_example</li> <li>main(void) vin</li> </ul>	+							
◆ Dachbeard ♡ 😥 🖇 🔍 🗖	140 #irdetcplusplus	• mam(void). m	L							
	150 #endif									
	151 🗸									
📑 S32DS Application Project 🛛 🎯 Projec	<									



## HANDS ON ETH: TEST RESULT 2

The received frame can be watched in rxBuffer.

							1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	C1 F1	- 8
Name	Туре	Value							
> 🥏 TxOptions	Gmac_Ip_TxOptionsType	{}							
> 🥭 TxBuffer	Gmac_Ip_BufferType	{}							
🗸 🥭 RxBuffer	Gmac_lp_BufferType	{}							
> 🔹 Data	uint8 *	0x34501d00 <gm< td=""><td>AC_0_RxRing_</td><td>0_DataBuffe</td><td>er&gt; "fUD3\"\0</td><td>21fUD3\"\021"</td><td></td><td></td><td></td></gm<>	AC_0_RxRing_	0_DataBuffe	er> "fUD3\"\0	21fUD3\"\021"			
🕬 Length	uint16	64							
> 🥭 TxInfo	Gmac_lp_TxInfoType	{}							
Console 🕮 Registers 🖹 Prob	olems 🗿 Executables 🗊 Debug Shell 🔋 Mem	ory Spaces III Watch registe	s 🕢 Debungari	i I Me	emory 🖾				
Console IIII Registers 🖹 Prob	blems 📀 Executables 🗵 Debug Shell 🟮 Mem	ory Spaces IIII Watch registe Ox34501d00 : 0x34501D00	s 🔀 Debugger ( <hex> 🛙 🍦 I</hex>	Concelle 0 Me New Rendering	emory ⊠ Is				
Oconsole IIII Registers III Prob onitors • 0x34501d00	olems 🔾 Executables 🕘 Debug Shell 🟮 Mem	ory Spaces         WW Watch register           0x34501d00 : 0x34501D00           Address         0 - 3	s <u>R Debugger (</u> <hex> ⊠ ♣ 4 - 7</hex>	New Rendering 8 - B	emory ⊠ s C - F				
Console IIII Registers I Prob ponitors Ox34501d00	olems 🔾 Executables 🗓 Debug Shell 🔋 Mem 🛓 🗱 🍕	Ory Spaces         IIII Watch register           0x34501d00 : 0x34501D00         Address         0 - 3           34501D00         66554433	s ₽ Debugger ( <hex> 22 ₽ 4 - 7 22116655</hex>	New Rendering 8 - B 44332211	mory ⊠ s C - F 002E0000				
Console IIII Registers I Protonitors	blems 💽 Executables 🗓 Debug Shell 🟮 Mem	Ory Spaces         W Watch register           0x34501d00 : 0x34501D00         Address         0 - 3           34501D00         66554433         34501D10         00000000	s	Concole   Mew Rendering  8 - B  44332211  0000000	mory ≅ s C - F 002E0000 0000000				
Oconsole IIII Registers III Prob onitors Ox34501d00	olems 💽 Executables 🗓 Debug Shell 🔋 Mem 🗧 🗱 🕅	Ory Spaces         WW Watch register           0x34501d00 : 0x34501D00         Address         0 - 3           34501D00         66554433         34501D10         00000000           34501D10         00000000         34501D20         00000000	S Debugger <hex> ⊠ ↓ 4 - 7 22116655 00000000 00000000 00000000</hex>	Image: Constraint of the second sec	mory ☆ s C - F 002E0000 00000000 00000000				
Console IIII Registers I Protonitors	olems 💽 Executables 🗊 Debug Shell 🕕 Mem	Ory Spaces         IIII Watch register           0x34501d00 : 0x34501D00         Address         0 - 3           34501D00         66554433         34501D10         00000000           34501D10         00000000         34501D20         00000000           34501D30         000000000         34501D30         00000000	s	Cenecli I Me New Rendering 8 - B 44332211 00000000 00000000 00000000 00000000	mory ≅ s C - F 002E0000 00000000 00000000 00000000 0000000				
Console IIII Registers I Protonitors • 0x34501d00	blems 💽 Executables 🕢 Debug Shell 🚺 Mem	Ory Spaces         W Watch register           0x34501d00:0x34501D00         Address         0 - 3           34501D00         66554433         34501D10         00000000           34501D20         00000000         34501D20         00000000           34501D30         00000000         34501D40         00000000           34501D40         00000000         34501D40         00000000	S Debugger <hex> ⊠ ↓ 4 - 7 22116655 0000000 0000000 0000000 0000000 000000</hex>	Consoli         Image: Mail           New Rendering         8           8         -           44332211         0000000           00000000         0000000           00000000         0000000           00000000         0000000           00000000         0000000	mory ≅ s C - F 002E0000 0000000 00000000 65EA1543 00000000 00000000				
Console IIII Registers Protonitors	olems 🕢 Executables 🔃 Debug Shell 🚺 Mem	Ory Spaces         W Watch register           0x34501d00:         0x34501D00           Address         0         -         3           34501D00         66554433         34501D10         00000000           34501D20         00000000         34501D20         00000000           34501D30         00000000         34501D30         00000000           34501D40         00000000         34501D50         00000000           34501D50         00000000         34501D60         00000000	S Debugger <hex> ⊠ ↓ 4 - 7 22116655 0000000 0000000 0000000 0000000 000000</hex>	B         B           44332211         0000000           00000000         0000000           00000000         0000000           00000000         0000000           00000000         0000000           00000000         0000000           00000000         00000000	mory № s] C - F 002E0000 0000000 0000000 65EA1543 00000000 00000000 00000000				
Console IIII Registers Protonitors	olems 💽 Executables 🗊 Debug Shell 🔋 Mem	Ory Spaces         IIII Watch register           0x34501d00:0x34501D00         Address         0 - 3           34501D00         66554433           34501D10         00000000           34501D20         00000000           34501D30         00000000           34501D40         00000000           34501D50         00000000           34501D50         00000000           34501D50         00000000           34501D70         00000000	s	Concell Me New Rendering 8 - B 44332211 00000000 00000000 00000000 00000000	mory ≅ s C - F 002E0000 00000000 00000000 00000000 0000000				

# Hands on CAN Example



#### PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2023 NXP B.V.

## HANDS ON CAN – OBJECTIVE

How to import the CAN example into S32DS
How to configure the clock of CAN via S32DS
How to configure the port of CAN via S32DS
How to modify the CAN loopback
How to debug the CAN example with S32 debug probe

## HANDS ON CAN : IMPORT CAN EXAMPLE PROJECT

Open S32 Design Studio, go to "File -> New -> S32DS Project From Example". Select "FlexCAN\_Ip\_Example\_S32G399A\_M7" example, then click on "Finish". The project is copied into current workspace.

New	Alt+Shift+N >	S32DS Project from Example	Ctrl+Alt+E	Project na	me: FlexCAN_Ip_Example_S32G399A_M7		FlexCAN ID Example S32G399A M7: Debug R
Open File Open Projects from File System		<ul> <li>S32DS Library Project</li> <li>S32DS Application Project</li> </ul>	Ctrl+Alt+L Ctrl+Alt+A	can		Description:	>   Includes
Recent Files Close	Ctrl+W	Makefile Project with Existing Code C/C++ Project Project		<ul> <li>✓ ▷ S3</li> <li>✓ ▷</li> </ul>	2 RTD AUTOSAR 4.4 3.0.2 HF01 D2204 Exam A Can Examples	Example Description     This is an FlexCAN IP driver component LoopBac     example.	Project_Settings
Save Save As Save All Revert	Ctrl+S Ctrl+Shift+S	Convert to a C/C++ Project (Adds C/C++ Nature)     Source Folder     Folder     Source File     Meader File			Can_example_532G399A_M7 Can_example_532R45_M7 FlexCAN_Ip_Example_532G274A_M7 FlexCAN_Ip_Example_532G399A_M7 FlexCAN_Ip_Example_532R45_M7 FlexCAN_Ip_Example_532R45_M7	1.1 The application software functionality Using Clock Jp_Init it initializes the clock sourc the clock tree and to configures the clock gating of th peripherals. The clock configuration that is used will enable and use the PLL as source clock.	> 🗭 include 🖉 description.txt
Move Rename Refresh Convert Line Delimiters To	F2 F5 >	File from Template     Class     Example     Other	Ctrl+N	< 20	2 RTD AUTOSAR 4.4 D2105 Example Project Can Examples Can_example_5326274A_M7 Can_example_532R45_M7	with inttru-jp_init it initializes the interrupt controller and it applies the interrupt configuration according to the Platform configuration. The interrup handlers are mapped using calls of IntCrt_Jp_InstalHandler. The application will register CAN0_ORED_0_7_MB_IRQHandler ISR handler and wi v	HexCAN_example_C1.mex

By default, "FlexCAN\_Ip\_Example\_S32G399A\_M7" project only shows the LoopBack function of FlexCAN.



PUBLIC 34

## HANDS ON CAN: MODIFY THE EXAMPLE

The "FlexCAN\_Ip\_Example\_S32G399A\_M7" project only supports loopback mode. This guide will demonstrate how to modify the default configuration to transmit CAN frame from FlexCAN\_0 to FlexCAN\_1



Automotive Bus Port





## HANDS ON CAN: PORT CONFIGURATION 1

### Go to Pin configuration view:

- Right-click the Project
- Select S32 Configuration Tools
- Select Open Pins
- Enable Pins

<ul> <li>✓ HescANJp_Exa &gt; @ Project_Setti &gt; @ src &gt; @ findude</li></ul>						
Image: Second Secon	<ul> <li>FlexCAN_Ip_Exa</li> <li>FlexCAN_Ip_Exa<td></td><td>New Go Into</td><td>&gt;</td><td></td><td></td></li></ul>		New Go Into	>		
Image: FlexCAN_exation       Copy       Ctrl+C         Delete       Delete         Paste       Ctrl+V         Image: Rename       F2         Source       >         Move       Import         Import       Export         Build Project       Clean Project         Close Project       Close Project         Close Unrelated Project       Build Configurations         Build Configurations Explorer       SDKs         Migrate       Show in Remote Systems view         Restore from Local History       Open Pins         Team       10         Configure       0         Source       0         Validate       0         Properties       Alt+Enter	> 2 src > 2 include 2 description.t		Open in New Window Show In Show in Local Terminal	Alt+Shift+W > >		
Import         Export         Build Project         Clean Project         Close Droject         Close Droject         Close Unrelated Project         Build Configurations         Build Configurations Explorer         Show in Remote Systems view         Restore from Local History         Team         Compare With         Compare With         Configure         Source         Validate         Properties	FlexCAN_exa	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Copy Delete Paste Rename Source Move	Ctrl+C Delete Ctrl+V F2 >		
Build Project         Clean Project         Close Vroject         Close Vorgett         Close Unrelated Project         Build Targets         Build Configurations         Build Configurations Explorer         SDKs         Migrate         Show in Remote Systems view         Restore from Local History         East         Validate         Validate         Validate         Properties		24 24	Import Export			
Build Targets       >         Index       >         Build Configurations       >         Build Configurations Explorer       >         SDKs       Migrate         Show in Remote Systems view          Run As       >         Restore from Local History          Source       >         Source       >         Validate       Open NVT         Properties       Alt+Enter		Ð	Build Project Clean Project Refresh Close Project Close Unrelated Project	F5		
Build path       >         Build Configurations Explorer       >         SDKs       Migrate         Show in Remote Systems view       Run As         Run As       >         Team       >         Configuration Tools       Image: Configuration Tools         Team       Image: Configuration Tools         Configure       Image: Configuration Tools         Source       Image: Configuration Tools         Validate       Image: Configuration Configuration (*, Manage SDK Componer)			Build Targets Index Build Configurations	>		
Show in Remote Systems view       Run As       Debug As       Restore from Local History       S32 Configuration Tools       Team       Compare With       Configure       Source       Validate       Properties       Alt+Enter			Build path Build Configurations Explorer SDKs Migrate	>		
Image: S32 Configuration Tools     Image: Open Pins       Team     10     Open Clocks       Compare With     Image: Open Peripherals       Configure     Image: Open IVT       Source     Image: Open QuadSPI       Validate     Image: Open QuadSPI       Properties     Alt+Enter       Image: DDC     Image: Open QuadSPI       Manage SDK Component     Image: Open Open Open Open Open Open Open Open		0 *	Show in Remote Systems view Run As Debug As Restore from Local History	>		
Team     >     ¶I     Open Clocks       Compare With     >     ?     Open Peripherals       Configure     >     0     Open IOCD       Source     >     Image: Open QuadSPI       ✓     Validate      Ø Open QuadSPI       Properties     Alt+Enter     Import Configuration (*, Manage SDK Componer			S32 Configuration Tools	>	۲	Open Pins
Import Configuration (*. Manage SDK Componen			Team Compare With Configure Source Validate Properties	> > > Alt+Enter	© ♥ @     % ♥ ⊡	Open Clocks Open Peripherals Open DCD Open IVT Open QuadSPI Open DDR
		-	roperace	And Ellier		Import Configuration (*.m Manage SDK Components

<ul> <li>Configuration - Gen</li> </ul>	eral Info	V Configuratio	on - HW Info	V Pro	oject	
Name: FlexCAN_exan Path: C:\Users\NXF65	nple_CT.mex 386\woExample_S32G399	Processor: S A_M7 Part number: S Core: C SDK Version: F	532G399A 532G399A_525bga Cortex-M7 (Core #0) PlatformSDK_S32XX_2022_0	Proje T P 3 Part	ect name: FlexCAN_Ip_Exam oolchain: NXP GCC 9.2 for A rocessor: S32G399A number: S32G399A_525bga Core: Cortex-M7 (Core #	nple_S32G399A_M rm 32-bit Bare-Me n 0)
Pins     Configure     Tray including     Generation     Gene	<ul> <li>Cocks</li> <li>Configure</li> <li>Configure</li> <li>Configure</li> <li>Configure</li> <li>Configure</li> <li>Configure</li> <li>Configure</li> <li>System, bus, and periphe rai</li> <li>Cocks</li> <li>Cocks</li> <li>Cocks</li> <li>System, bus, and periphe rai</li> <li>Cocks</li> <li>System, bus, and periphe rai</li> <li>Cocks</li> <li>Second</li> <li>General-Bcfg.</li> <li>General-Cfg.c</li> <li>General-Cfg.c</li></ul>	<ul> <li>Peripherals</li> <li>Configue res the store of the SDK peripherals</li> <li>Generated code</li> <li>Update code enable</li> <li>generaCfgh</li> <li>generaCfgh</li> <li>generatfines.h</li> <li>genera</li></ul>	► DCD Device Configur ation Data is informat informat ion containe d in the DCD image, that the DCD image, that the Configur e various prophe rais on the device.	▼ IVT Configures (V), which is the first label (V), which is the first label (V), which is the first label attack e he label the Boot device. An IVT contains the boot device. An IVT contains the require d data compone ents like orgon point, a Dofter point, a Dofter point, a Dofter point, a Dofter point, a Dofter point, a Dofter an dofter point, a Dofter point, a Dofter point, a Dofter point, a Dofter point, a Dofter point, a Dofter an Configures an Boote d data (Configures) Configures an Dofter point, a Dofter point,	▼ QuadSPI tool allows the configure ation of the parameters ation of the ation of ation of the ation of ation of the ation of ation of	
					Close and Update Co	de <sup>0</sup> Close

NP

## HANDS ON CAN: PORT CONFIGURATION 2

#### Add the Pins as the schematic of FlexCAN0 and FlexCAN1





Routed Pins	for BOARD.	4 🖸	8~~							
#	Peripheral	Signal	Route to	Label	Identifier	Power group	Direction	Output Buffer	Open Drain	Input Buffer
D7	CAN_0	rxd	PB_02		n/a	VDD_IO_B (0V)	Input	Disabled	Disabled	Enabled
E7	CAN_0	txd	PB_01		n/a	VDD_IO_B (0V)	Output	Enabled	Disabled	Disabled
E8	CAN_1	rxd	PB_04		n/a	VDD_IO_B (0V)	Input	Disabled	Disabled	Enabled
C6	CAN_1	txd	PB_03		n/a	VDD_IO_B (0V)	Output	Enabled	Disabled	Disabled

## HANDS ON CAN: PORT CONFIGURATION 3

FlexCAN_Ip_     FlexCAN_Ip_     Sill Includes     Sill Project Set		New Go Into	>		
✓ <sup>™</sup> src > <sup>™</sup> main.c✓ include		Open in New Window Show In Show in Local Terminal	Alt+Shift+W > >		
Ø description	₽ × 10 11	Copy Delete Paste Rename Source Move	Ctrl+C Delete Ctrl+V F2 >		
	22 23	Import Export			
	£	Build Project Clean Project Refresh Close Project Close Unrelated Project	F5		
		Build Targets	>	L .	
		Index	>		
		Build Configurations	>		
		Build path Build Configurations Explorer SDKs Migrate	>		
		Show in Remote Systems view		L .	
	0	Run As	>	1	
	枠	Debug As	>		
	100	S32 Configuration Tools	>		Open Pins
		Team	>	'n	Open Clocks
		Compare With	>	Ŷ	Open Peripherals
		Configure	>	ð	Open DCD
		Source	>		Open IVT
		Validate		3	Open QuadSPI
		Properties	Alt+Enter		Import Configuration (*.mex)
	_				Manage SDK Components

Add the Port configuration:

- Right-click the Project,
- Select S32 Configuration Tools
- Select Open Peripherals
- Click on the plus button
- Click on the Siul2\_Port component
- Click on OK
- The Siul2\_Port driver will be added



SS Se	Select configuration component — 🗆 >									
Select	which components sh	hould be offered Late	est component	~						
siul2					×					
Config &	guration component Siul2_Dio	Component descrip Siul2_Dio Configura	tion tion	Category Drivers	Rec plat					
â :	Siul2_lcu	SIUL2 Driver		Drivers	plat					
۵	Siul2_Port	Siul2_Port Configura	tion	Drivers	plat					
<		1 <sup>1</sup> Decisioner la	ОК	Cancel	,					
	Components 🖾	Peripherals								
t	type filter text			•	t¥.					
		MCAL			D					
		Drivers			Ð					
	<sup>O</sup> FlexCAN	<sup>O</sup> IntCtrl_Ip	<sup>O</sup> osif_1	Siul2_Port						

Switch to clocks configuration view:

- Right-click the Project,
- Select S32 Configuration Tools
- Select Open Clocks

		_			
~ 4	FlexCAN_Ip_Example_		New	>	
	Includes     Project Settings		Go Into		
	RTD		Open in New Window		
	src 🖉		Show In	Alt+Shift+W >	
	> 🖻 main.c		Show in Local Terminal	>	
	🧭 include	阍	Сору	Ctrl+C	
	description.txt	×	Delete	Delete	
	FlexCAN_example_C	1	Paste	Ctrl+V	
			Rename	F2	
			Source	>	
			Move		
		<u>in</u>	Import		
		14	Export		
			Build Project		
			Clean Project		
		8	Refresh	F5	
			Close Project		
			Close Unrelated Project		
			Build Targets	>	
			Index	>	
			Build Configurations	>	
			Duild eath		
			Build Car Caratian Sector	1	
			Build Configurations Explorer		
			SUKS		
			migrate		
			Show in Remote Systems view		
		0	Run As	>	
		夺	Debug As	>	
		-	Restore from Local History		
			S32 Configuration Tools		Open Pins
			Compare With	( <b>L</b>	Open Peripherals
			Configure		Open DCD Open Clocks
			Source		Open IVT
			Validate	5	Copen QuadSPI
		12.1	Fundate		Open DDR
		_	Properties	Alt+Enter	Import Configuration (*.mex)
					Manage SDK Components

Open the Peripheral Clock View, double-click the FLEXCAN0\_CLK. The Clocks Diagram will show the clock tree and the key node can be re-set. The default clock configuration of FlexCAN is 48 MHZ. Switch the clock which source from FXOSC(40 MHZ).



NO

Mode setting for FlexCAN0:

- Right-click the Project,
- Select S32 Configuration Tools
- Select Open Peripherals
- Double-click FlexCAN component
- Set mode for FlexCAN0





Configure the BaudRate as 500Kbps for FlexCAN0

- TimeQuantum (seconds) = Prescaler / CanClockFrequency
- No. of CanTimeQuantas = (1 / CancontrollerBaudRate) / TimeQuantum
- No. of CanTimeQuantas = 1 + CanControllerPropSeg + CanControllerSeg1 + CanControllerSeg2

FlexCAN Protocol Clock	40000000
<ul> <li>FlexCAN bitrate</li> </ul>	
Name	FlexCAN_lp_TimeSeg0
Synchronization segment	1
Propagation segment	7
Phase segment 1	7
Phase segment 2	5
Prescaler division factor	4
Resync jump width	1
Bitrate (Kbps)	500
Sampling point (%)	75

#### Add FlexCAN1:

- Right-click the Project,
- Select S32 Configuration Tools
- Select Open Peripherals

<ul> <li>FlexCAN_Ip_</li> <li>Includes</li> <li>Project Se</li> </ul>		New Go Into	· · ·		
✓ <sup>™</sup> src > <sup>™</sup> main.c > <sup>™</sup> include		Open in New Window Show In Show in Local Terminal	Alt+Shift+W > >		
Ø description ₩ FlexCAN_e	₿ ¥ 10 11	Copy Delete Paste Rename Source Move	Ctrl+C Delete Ctrl+V F2 >		
	22 23	Import Export			
	Ł	Build Project Clean Project Refresh Close Project Close Unrelated Project	F5		
		Build Targets Index	>		
		Build Configurations Build path Build Configurations Explorer SDKs Migrate	>		
	0 *	Show in Remote Systems view Run As Debug As Restore from Local History	>		
		S32 Configuration Tools	>	۲	Open Pins
		Team	>	'n	Open Clocks
		Compare With	>	Y	Open Peripherals
		Configure	>		Open IVT
		Source	>	\$	Open QuadSPI
		Properties	Alt+Enter	123	Open DDR
	_	riopendes	Ait+Enter		Import Configuration (*.mex)
					Manage SDK Components



#### Add FlexCAN1:

- Double-click FlexCAN component
- Click on plus button to add FlexCAN\_1
- Set Clock as 40MHz for FlexCAN\_1
- Set Bitrate as 500kbps

📚 Components 🕴 🦞 Peripherals 🗧 🗖				
type filter text				
	MCAI	L	0	
(1)	Driver	0		
<sup>O</sup> FlexCAN	<sup>O</sup> IntCtrl_Ip	osif_1	Siul2_Port	

IP FlexCA	AN Driver [Drivers]	
Name FlexCA	N	Custom nam
Mode Genera	al Mode	
		Preset Custom
Name Config	Time Support FlexCAN configurations FlexC	AN General Timeout Configuration
+ × 4	•	Add FlexCAN_1
0	FlexCAN Hardware Channel	FLEXCAN_1
1	Name	FlexCAN_Config1
	Number Of MB	16
	FlexCAN Rx FIFO filters number	8 Rx FIFO Filters
	Enable Legacy RxFiFO	
	Enable FD Can	
	FlexCAN operation modes	Normal mode or user mode
-	Payload Size (bytes)	8
	Rx FIFO Transfer Type	Using interrupts
	DMA Channel Used	0
	Enable EnhancedCBT Can	
-	Enable BitRate Switch	
	Auto Bus Off	
	Remote Request Store	
-	TimeStamp Free Runing Timer Source	e Captured time base is CAN bit clock.
-	TimeStamp Timer Source	Message buffer time stamp base is TIMER.
	TimeStamp HR Capture Config	The high resolution time stamp capture is disabled. Enable Timer Source
	FlexCAN Protocol Clock	4000000
	✓ FlexCAN bitrate	
	Name FlexC	AN_Ip_TimeSeg1
	Synchronization segment 1	
	Propagation segment 7	
	Phase segment 1 7	Cat Ditrota
Phase segment 2 5		Set Billate
	Prescaler division factor 4	
	Resync jump width 1	
	Bitrate (Kbps) 500	
	Sampling point (%) 75	

## HANDS ON CAN: UPDATE CODE

#### Generate code method:

1. Open the view of any configuration tool, like Pins, then click **Update Code** (ensure desired project is selected)

2. The Update Files window pops up. It shows the detailed update information. Click **OK** button.

3.The configuration .c and .h files will be generated at "generate" folder.



## HANDS ON CAN: APPLICATION CODE 1

#### Modify the main.c:



PUBLIC 46

## HANDS ON CAN: APPLICATION CODE 2

Modify the main.c:

Flexcan\_Ip\_DataInfoType rx\_info = { .msg\_id\_type = FLEXCAN\_MSG\_ID\_STD, .data\_length = 8u, .is polling = TRUE, .is\_remote = FALSE }; Flexcan Ip MsgBuffType rxData; FlexCAN\_Ip\_Init(FLEXCAN0\_INST, &FlexCAN\_State0, &FlexCAN\_Config0); FlexCAN Ip SetStartMode(FLEXCAN0 INST); FlexCAN\_Ip\_Init(FLEXCAN1\_INST, &FlexCAN\_State1, &FlexCAN\_Config1); Initialize FlexCAN 1 Controller FlexCAN Ip SetStartMode(FLEXCAN1 INST); FlexCAN Ip ConfigRxMb(FLEXCAN1\_INST, RX\_MB\_IDX, &rx\_info, MSG\_ID); Configure Rx MailBox for FlexCAN 1 rx\_info.is\_polling = FALSE; FlexCAN0 send the message FlexCAN\_Ip\_Send(FLEXCAN0\_INST, TX\_MB\_IDX, &rx\_info, MSG\_ID, (uint8 \*)&dummyData); FlexCAN\_Ip\_Receive(FLEXCAN1\_INST, RX\_MB\_IDX, &rxData, TRUE); while(FlexCAN\_Ip\_GetTransferStatus(FLEXCAN1\_INST, RX\_MB\_IDX) != FLEXCAN\_STATUS\_SUCCESS) FlexCAN1 receive the message { FlexCAN\_Ip\_MainFunctionRead(FLEXCAN1\_INST, RX\_MB\_IDX); } FlexCAN Ip SetStopMode(FLEXCAN0 INST);

FlexCAN\_Ip\_SetStopMode(FLEXCAN0\_INST);
FlexCAN\_Ip\_Deinit(FLEXCAN0\_INST);
FlexCAN\_Ip\_Deinit(FLEXCAN0\_INST);
FlexCAN\_Ip\_Deinit(FLEXCAN1\_INST);
Exit\_Example(TRUE);

## HANDS ON CAN: BUILD AND DEBUG 1

#### Build target Project:

- Right-click the Project
- Build Project
- The console will print build information
- FlexCAN\_lp\_Example\_S32G399A\_M7.elf is created





## HANDS ON CAN: BUILD AND DEBUG 2

Go to debug configuration:

- Right-click the Project
- Select the Debug As
- Click on Debug Configurations

<ul> <li>FlexCAN_lp_Exa</li> <li>Rinaries</li> </ul>		New Go Into	>		
> 🔐 Includes > 🤐 Project_Settin > 🥔 RTD > 🥝 board		Open in New Window Show In Show in Local Terminal	Alt+Shift+W>		
<ul> <li>Solid</li> <li>generate</li> <li>generate/incl</li> <li>generate/src</li> <li>grc</li> <li>grc</li> <li>main.c</li> <li>Debug_RAM</li> </ul>		Copy Delete Paste Rename Source Move	Ctrl+C Delete Ctrl+V F2		
<ul> <li>include</li> <li>description.tx</li> </ul>	R R	Import Export			
M FlexCAN_exat	£	Build Project Clean Project Refresh Close Project Close Unrelated Project	F5		
		Build Targets Index Build Configurations	> > >		
		Build path Build Configurations Explorer SDKs Migrate	>		
	0	Show in Remote Systems view Run As	,		
	*	Debug As Restore from Local History	>	C	1 Local C/C++ Application 2 S32DS C/C++ Application
	-	S32 Configuration Tools Team	>		Debug Configurations
		Configure Source	>		
		Validate Properties	Alt+Enter		

#### Debug configuration setting:

- Connect the S32 Debug probe with PC and RDB3
- Click on target project

2 2 0

~ C/C C F C/C

- Select the target device and core as S32G399A\_M7\_0
- Select target S32 Debug Probe

2 📭 🗟 🗶 🖻 🏹 🔸	Name: FlexCAN_Ip_Example_S32G399A_M7_Debug_RAM_S32Debug				
pe filter text	Main 🕸 Debugger 🕨 Startup 💱 Source 🔲 Common 🕮 SVD Support 🕮 OS Awareness 👼 Trace and Profile				
C C/C++ Application FlexCAN_Ip_Example_S32G399A_M7.elf	Hardware				
C/C++ Remote Application	Device: S32G399A Core: M7_0	Selec	Select device and con		
Eclipse Application     GDB Hardware Debugging     GDB Hirdware Debugging	Initialization script: \$(532DS_INITIALIZATION_SCRIPTS_DIR)/s32g3xx/s32g3xx_generic_bareboard.py	Bro	wse Variables		
GDB PEMicro Interface Debugging	Debug Probe Connection				
Launch Group	Interface: S32 Debug Probe - USB	~			
<ul> <li>Launch Group (Deprecated)</li> <li>Launch Group for S32 Debugger</li> <li>S32 Debugger</li> <li>FlexCAN_Ip_Example_S32G399A_M7_Debug_RAM_S32Debug</li> <li>S32 Debugger Flash Programmer</li> <li>VLAB Simulator Debugging</li> </ul>	USB device: 00:04:9f:06:6d:1e - S32 Debug Probe	~ Refresh	Test connection		
	Hostname or IP: Target Communication Speed JTAG Speed (KHz): 16000 Timeout: 30 s Delay after reset: 0 ms GDB Server Launch server Server port number® 45000 Enable log				



## HANDS ON CAN: DEBUG AND RUN

Power on the RDB3, click on "Apply", then click on "Debug". The view will switch to the Debug Perspective, and you can use the controls to control the program flow.

Name: FlexCAN_Jp_Example_S32G399A_M7_Debug_RAM_S32Debug	File Edit Source Refactor Navigate Search Project ConfigTools Run FreeRTOS Window Help
Main (* Debugger > Startup > Source Common IIII SVD Support . SO Awareness > Trace and Profile	[] • [] 입] 월[입] 성 역 [] ] · [ ] · [] · [] · [] · [] · [] ·
Hardware	A Debug ﷺ B Project Explorer     B T □ @ mainc ﷺ B Siul2_Port_Ip_Cfg.h   0x34020404   S32G399A_COMMON.h
Device: S32G399A Core: M7_0 Select device Initialization script: \$(S32DS INITIALIZATION SCRIPTS DIR)/s32p3xx/s32p3xx generic bareboard.pv Browse, V/	and core
✓ Initial core	✓ PrecAN_Ip_Example_S32G399A_M7.elf [1] [con 32⊕ /*1
Debug Probe Connection	■ main() at main.c:44 0x340001e0 33 \brief The main function for the project.
Interface: S32 Debug Probe - USB v	₩ C:/NXP/S32DS.3.4/S32DS/tools/gdb-arm/arm3 35 * - startup asm routine
USB device: 00:04:9f:06:6d:1e - S32 Debug Probe V Refresh Test co	nnection S32 Debugger 36 * - main() Semihosting 37 */ 38 extern void CANO ORED 0 7 MB_IROHandler(void):
Target Communication Speed	39 extern void CAN1_ORED_0_7_MB_IRQHandler(void);
JTAG Speed (KHz): 16000 Timeout: 30 s	41⊕int main(void)
Delay after reset: 0 ms	42 { 43 /* Write your code here */
GDB Server	<pre>&gt;44 Clock_Ip_Init(&amp;Mcu_aClockConfigPB[0]);</pre>
C Launch server Server port number: 45000 Enable log	45 46 Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0); 47 48 IntCtrl Ip_EnableIrg(CAN0_ORED_0_7_MB_IROp):
GDB Client	49 IntCtrl_Ip_InstallHandler( <i>CAN0_ORED_0_7_MB_IRQn</i> , CAN0_ORED_0_7_MB_IRQHandler, NULL_PTR); 50
Executable: \$\\$32DS_GDB_ARM32_PY\ Select Browse Vi	51 IntCtrl_Ip_EnableIrq(CAN1_ORED_0_7_MB_IRQn);
Commands:	<pre>&gt;&gt; &gt;&gt; &gt;&gt;</pre>
Revert	Apply S6 .data_length = 8u, 57 .is_polling = TRUE, 58 .is_remote = FALSE
Debug	Close 59 };

NXC

## HANDS ON CAN: TEST RESULT

Add a breakpoint to the FlexCAN\_Ip\_SetStopMode function, then click on the Resume option. The received CAN frame can be watched from rxData.

WorkspaceS32DS.3.4 - FlexCAN_p_Fampl_S32G39	IA_M7/src/main.c - S32 Design Studio for S32 Platform				- 0 ×
File Edit Source Refactor Navigate search Proj	ect ConfigTools Run FreeRTOS Window Help				
	○ (1) (1) (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2			c	ዲ 📴 🗟 🖷 ጣ 🕴 💩 🚍 🕬 🔯
to Debug 🕮 🏠 Project Explorer	🖻 main.c 🕄 🖻 main.c 🖹 Clock_lp_BOA 🖹 Siul2_Port_L 🗟 Siul2_Port_lp.c 🗟 Siul2_Port_L 📓 linker_ram 🖹 startup_cm7.s 🦜 🧮	🗆 🕬 Variables 🕮 💁 Breakpoi	ớć Expressio 🛋 Modules 🕮 🛙	Periphera 🕮 Arm Syst	EmbSys So Peripherals
<ul> <li>FlexCAN_Jp_Example_S32G399A_M7_Debug_RAM</li> <li>FlexCAN_Jp_Example_S32G399A_M7_elf [1] [col</li> <li>Thread #11 [core: 0] (Suspended : Breakpoi</li> <li>main() at mainc.76 0x3400027e</li> <li>C/NXP/S32DS.34/S32DS/tools/gdb-arm/arm3</li> <li>S32 Debugger</li> <li>Semihosting</li> </ul>	<pre>45 5 Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0); 47 47 47 47 47 47 47 47 47 47</pre>	Name > @ rx_info	Type Flexcan_Ip_DataInfoType Flexcan_Ip_MsgBuffType uint32 uint32 uint8 [64] uint8 uin8 uin8 uin8 uin8 uin8 uin8 uin8 uin	Value           {}           34078855           20           0x34401fb0           1 \001'           2 \002'           3 \003'           4 \004'           5 \006'           7 \a'           0 \0'	
Constant ≥      Constant =      Constant =      Constant =      Constant =      Constant	<pre>70 FlexCAN_Ip_Send(FLEXCAN0_INST, TX_MB_IDX, ℞_info, MSG_ID, (uint8 *)&amp;dummyData); 71 72 FlexCAN_Ip_Receive(FLEXCAN1_INST, RX_MB_IDX, &amp;rxData, TRUE); 73 while(FlexCAN_Ip_GetTransferStatus(FLEXCAN1_INST, RX_MB_IDX) != <i>FLEXCAN_STATUS_SUCCESS</i>) 74 { flexCAN_Ip_MainFunctionRead(FLEXCAN1_INST, RX_MB_IDX); } 75 FlexCAN_Ip_SetStopMode(FLEXCAN0_INST); 76 FlexCAN_Ip_Deinit(FLEXCAN0_INST); 77 FlexCAN_Ip_Deinit(FLEXCAN0_INST); 78 FlexCAN_Ip_Deinit(FLEXCAN0_INST); 79 FlexCAN_Ip_Deinit(FLEXCAN0_INST); 79 FlexCAN_Ip_Deinit(FLEXCAN0_INST); 80 Exti_Example(TRUE); 81 } 81 } 82 83 /* END main */ 84 *! 85 ** @) 86 */</pre>	V Galariti Name : data[0] B: Outline IS IS: Disassembly IntCtrl.jp.h Check_example.h Siul2_Port.jp.h FLEXCAN0_INST FLEXCAN0_INST MSG_JD RX_MB_JDX V exit_code: volatile int dummyData: uintB[] CAN0_ORED_0_7_MB_IR CAN0_ORED_0_7_MB_IR CAN0_ORED_0_7_MB_IR CAN0_ORED_0_7_MB_IR CAN0_ORED_0_7_MB_IR	QHandler(void) : void		□ 授 残 ¥ ● 推 8 □ □

Note: Make sure FlexCAN\_0 connect to FlexCAN\_1 via physical wiring



# SECURE CONNECTIONS FOR A SMARTER WORLD



#### **LEGAL INFORMATION**

#### Definitions

**Draft** — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

#### Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

**Right to make changes** — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

#### **LEGAL INFORMATION**

**Applications** — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer's. NXP does not accept any liability in this respect.

**Terms and conditions of commercial sale** — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application of any product in the design of the provided by the product of the provided by the pro

#### **LEGAL INFORMATION**

**Export control** — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

**Translations** — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

**Security** — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

#### Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners. NXP — wordmark and logo are trademarks of NXP B.V.