S32K116 EVB

QUICK START GUIDE

APPLIES FOR: S32K116 EVB (SCH_30003 REV B)





SECURE CONNECTIONS FOR A SMARTER WORLD

EXTERNAL USE

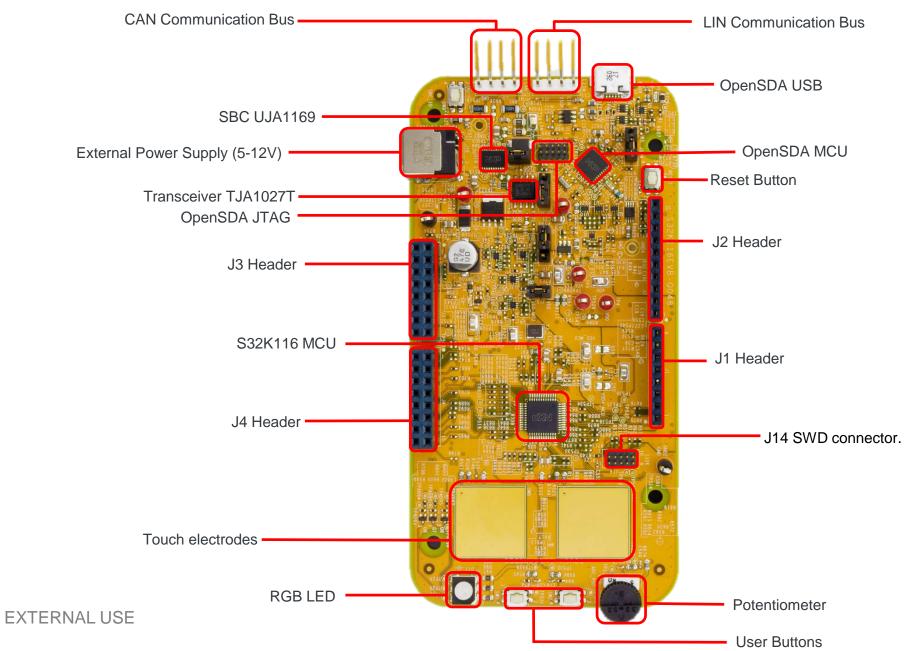
Contents:

- Get to Know S32K116 EVB
- Out of the Box Setup
- Introduction to OpenSDA
- Creating a new S32DS project for S32K116:
 - Download
 - Create a project
 - Create a project from SDK example
- S32DS Debug basics
- Create a P&E debug configuration



Get to know S32K116-EVB

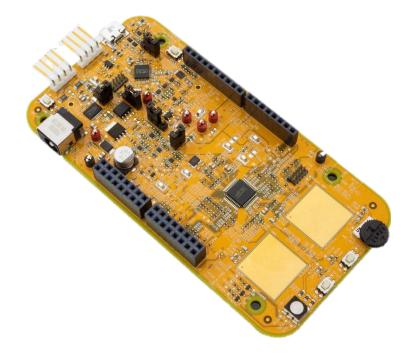
2





S32K116 EVB Features:

- Supports **48LQFP and 32QFN** packages
- Small form factor size 4.5" x 2.3"
- Arduino[™] UNO footprint-compatible with expansion "shield" support
- Integrated open-standard serial and debug adapter (OpenSDA) with support for several industry-standard debug interfaces
- Easy access to the MCU I/O header pins for prototyping
- On-chip connectivity for CAN, LIN, UART/SCI.
- SBC UJA1169 and LIN phy TJA1027
- Potentiometer for precise voltage and analog measurement
- RGB LED
- Two push-button switches (SW2 and SW3) and two touch electrodes
- Flexible power supply options
 - microUSB
 - external 12V power supply

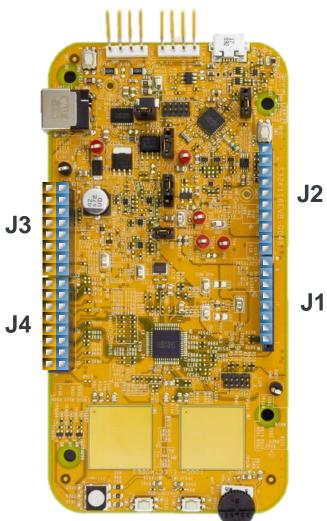




Header/Pinout Mapping for S32K116

PIN	PORT	FUNCTION	J3	PIN	PORT	FUNCTION
J3-02	PTB0	GPIO		J3-01	-	VBAT
J3-04	PTB1	GPIO		J3-03	-	VDD_PERH
J3-06	PTB6	GPIO		J3-05	PTA5	RESET
J3-08	PTB7	GPIO		J3-07	-	3.3V
J3-10	PTE4	GPIO		J3-09	-	5V
J3-12	PTE5	GPIO		J3-11	-	GND
J3-14	PTA11	GPIO		J3-13	-	GND
J3-16	PTD3	GPIO		J3-15	-	VBAT

PIN	PORT	FUNCTION	J4	PIN	PORT	FUNCTION
J4-02	PTC6	GPIO		J4-01	PTA7	ADC0_SE3
J4-04	PTC7	GPIO		J4-03	PTC8	GPIO
J4-06	PTC8	GPIO		J4-05	PTC1	ADC0_SE9
J4-08	PTC9	GPIO		J4-07	PTC14	ADC0_SE12
J4-10	PTD5	GPIO		J4-09	PTC15	ADC0_SE13
J4-12	PTD15	GPIO		J4-11	PTC16	ADC0_SE14
J4-14	PTD16	GPIO		J4-13	PTC9	GPIO
J4-16	PTE8	GPIO		J4-15	PTB13	GPIO



J2	PIN	PORT	FUNCTION
	J2-01	PTC2	FTM0_CH2
	J2-02	PTC3	FTM0_CH3
	J2-03	PTB5	LPSPI0_PCS
	J2-04	PTB4	LPSPI0_SOUT
	J2-05	PTB3	LPSPI0_SIN
	J2-06	PTB2	LPSPI0_SCK
	J2-07	-	GND
	J2-08	-	AREF
	J2-09	PTA1	LPI2C0_SDA
	J2-10	PTA0	LPI2C0_SCL

J1

J1	PIN	PORT	FUNCTION
	J1-01	PTA2	LPUART0_RX
	J1-02	PTA3	LPUART0_TX
	J1-03	PTA13	FTM1_CH7
	J1-04	PTA12	FTM1_CH6
	J1-05	PTD0	FTM0_CH2
	J1-06	PTD1	FTM0_CH3
	J1-07	PTD2	FXIO_D4
	J1-08	PTE9	FTM0_CH7

EXTERNAL USE 4

Arduino compatible pins NXP pins

*0ohm resistor is not connected

Jumper Settings

Jumper	Configuration	Description
J104	1-2	Reset signal to OpenSDA, use to enter into OpenSDA Bootloader mode
	2-3 (Default)	Reset signal direct to the MCU, use to reset S32K116.
J107	1-2 (Default)	S32K116 powered by 12V power source.
	2-3	S32K116 powered by USB micro connector.
J10	2-3 (Default)	MCU voltage 5v
	1-2	MCU voltage 3.3v
J108	1-2 (Default)	Select LIN master option
J15	1-2 (Default)	Used for current measurement



HMI mapping

Component	S32K116
Red LED	PTD16 (FTM0 CH1)
Blue LED	PTE8(FTM0 CH6)
Green LED	PTD15(FTM0 CH0)
Potentiometer	PTA7 (ADC0_SE3)
SW2	PTD3
SW3	PTD5
OpenSDA UART TX	PTB1(LPUART0_TX)
OpenSDA UART RX	PTB0(LPUART0_RX)
CAN TX	PTE5(CAN0_TX)
CAN RX	PTE4 (CAN0_RX)
LIN TX	PTC7(LPUART1_TX)
LIN RX	PTC6 (LPUART1_RX)
SBC_SCK	PTB2 (LPSPI0_SCK)
SBC_MISO	PTB3(LPSPI0_SIN)
SBC_MOSI	PTB4(LPSPI0_SOUT)
SBC_CS	PTB5(LPSPI0_PCS1)



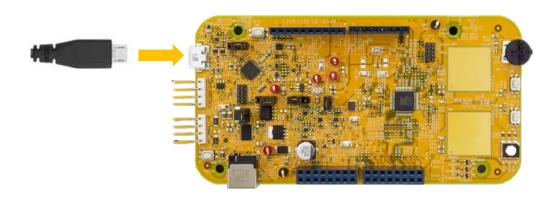
6 EXTERNAL USE

S32K116 EVB OUT OF THE BOX



Step 1: Power up the Board – EVB Power Supplies

- The S32K116-EVB evaluation board powers from a USB or external 12V power supply. By default 12V power is enabled with J107 (check slide 5)
- Connect the USB cable to a PC using supplied USB cable .
- Connect other end of USB cable (microUSB) to mini-B port on S32K116-EVB at J7
- Allow the PC to automatically configure the USB drivers if needed
- Debug is done using OpenSDA through J7





Step 2: Power up the Board – Is it powered on correctly?

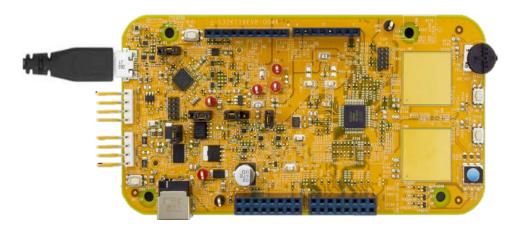
- When powered through USB, LEDs D2 and D3 should light green
- Once the board is recognized, it should appear as a mass storage device in your PC with the name S32K116EVB.





Step 3: Power up the Board – Is it powered on correctly?

 Board is preloaded with a software, in which the red, blue and green leds will toggle at different rates.





S32K116 JUMPSTART EXPERIENCE **BASED ON THE** FREEMASTER TOOL



Install the FreeMASTER tool

- Download and install the FreeMASTER PC application <u>www.nxp.com/FreeMASTER</u>.
- Open the FreeMASTER application on your PC. You should see Welcome page:





Power up the EVB board

- Powers the S32K116EVB evaluation board from a USB. By default, the USB power is enabled by J07 jumper.
- Connect the USB cable to a PC and connect micro USB connector of the USB cable to micro-B port J7 on the S32K116EVB.
- Allow the PC to automatically configure the USB drivers if needed.
- When EVB is powered from USB, LEDs D2 and D3 should light green.
- The EVB board is preloaded with a software toggling the RGB LED colors periodically between RED-GREEN-BLUE.







Setup serial connection in the FreeMASTER tool

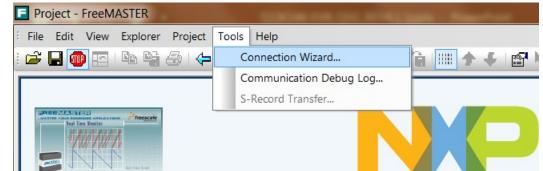
Setup communication port to "OpenSDA" and speed to 115200 b/s:

 Setup communication manualy: Go to: "Project > Options > Comm"

ommunicatio		ck Dir HTML Pages Demo Mode Views & Bars
RS232:	Port	opensda OpenSDA - CDC Serial Port (http://www.per
	Speed:	: 115200 Timeouts
C Plug-in N	lodule:	_
	- Autor au	drv=4;ptype=3;pnum=1;devid=PE5011560;devl - Configure
Connect	sung:	

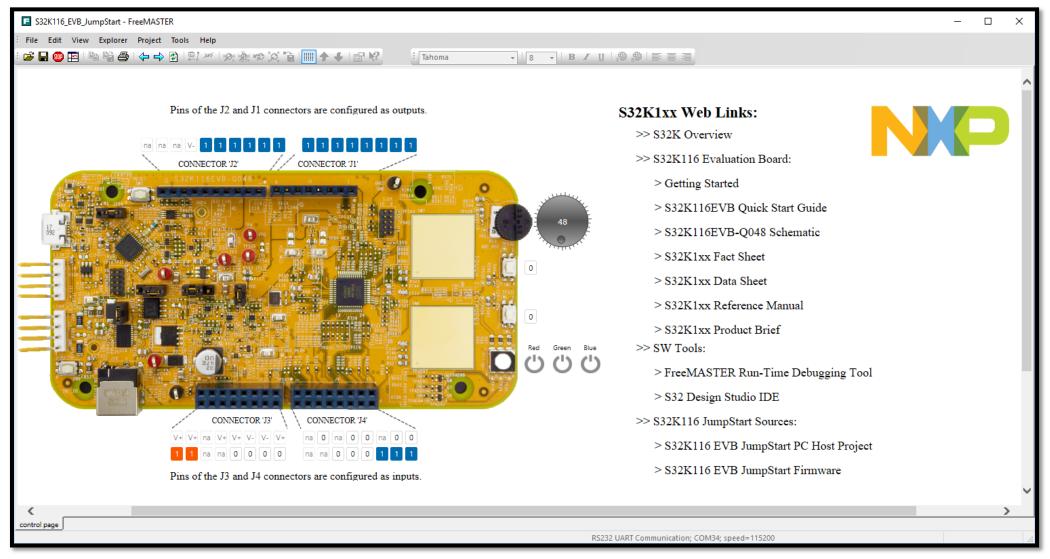
OR

 Setup communication automatically: Go to "Tools > Connection Wizard"

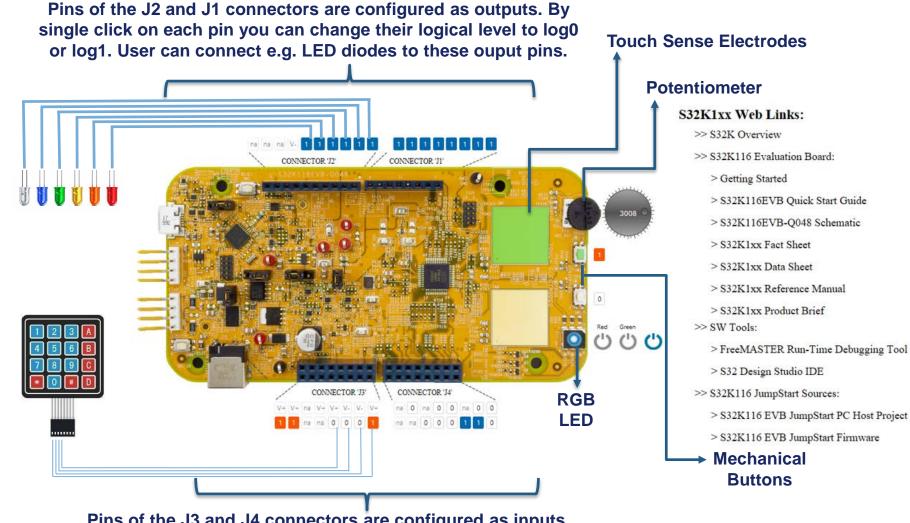




The FreeMASTER JumpStart project is loaded



The FreeMASTER JumpStart project description



Pins of the J3 and J4 connectors are configured as inputs. Logical level (log0/log1) is visualised for all connector pins. User can connect e.g. push-button keyboard to these input pins. **EXTERNAL USE**

16

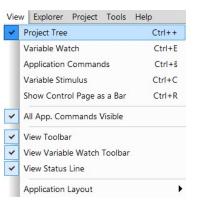
> FreeMASTER Run-Time Debugging Tool

Links to S32K1xx docs:

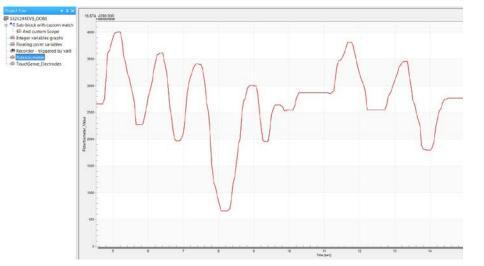
- Fact Sheet
- Data Sheet
- Reference Manual
- Product Brief
- S32K116EVB schematic
- S32K116EVB Quick Start Guide
- Tools:
 - FreeMASTER •
 - S32 Design Studio IDE
- S32K116EVB JumpStart source files

The FreeMASTER JumpStart oscilloscope feature examples

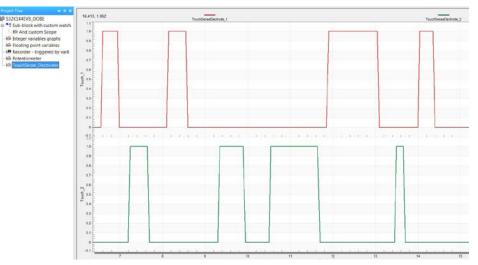
Display main project panel "View > Project Tree".



Display real-time oscilloscope graph examples such as "Potentiometer" or "Touch Sense Electrodes".



Analog values from potentiometer.



Responses from touch sense electrodes.



INTRODUCTION TO OPENSDA



Introduction to OpenSDA: 1 of 2

OpenSDA is an open-standard serial and debug adapter. It bridges serial and debug communications between a USB host and an embedded target processor. OpenSDA software includes a flash-resident USB mass-storage device (MSD) bootloader and a collection of OpenSDA Applications. S32K116 EVB comes with the MSD Flash Programmer OpenSDA Application preinstalled. Follow these instructions to run the OpenSDA Bootloader and update or change the installed OpenSDA Application.

Enter OpenSDA Bootloader Mode

- 1. Unplug the USB cable if attached
- 2. Set J104 on position 1-2.
- 3. Press and hold the Reset button (SW5)
- 4. Plug in a USB cable (not included) between a USB host and the OpenSDA USB connector (labeled "SDA")
- 5. Release the Reset button

A removable drive should now be visible in the host file system with a volume label of BOOTLOADER. You are now in OpenSDA Bootloader mode.

IMPORTANT NOTE: Follow the "Load an OpenSDA Application" instructions to update the MSD Flash Programmer on your S32K116 EVB to the latest version.

Load an OpenSDA Application

- 1. While in OpenSDA Bootloader mode, double-click **SDA_INFO.HTML** in the **BOOTLOADER** drive. A web browser will open the OpenSDA homepage containing the name and version of the installed Application. This information can also be read as text directly from **SDA_INFO.HTML**
- 2. Locate the **OpenSDA Applications**
- 3. Copy & paste or drag & drop the MSD Flash Programmer Application *to the BOOTLOADER drive*
- Unplug the USB cable and plug it in again. The new OpenSDA Application should now be running and a S32K116 EVB drive should be visible in the host file system

You are now running the latest version of the MSD Flash Programmer. Use this same procedure to load other OpenSDA Applications.



Introduction to OpenSDA: 2 of 2

The MSD Flash Programmer is a composite USB application that provides a virtual serial port and an easy and convenient way to program applications into the S32K116 MCU. It emulates a FAT file system, appearing as a removable drive in the host file system with a volume label of S32K116EVB. Raw binary and Motorola S-record files that are copied to the drive are programmed directly into the flash of the S32K116 and executed automatically. The virtual serial port enumerates as a standard serial port device that can be opened with standard serial terminal applications.

Using the MSD Flash Programmer

- 1. Locate the .srec file of your project , file is under the Debug folder of the S32DS project.
- 2. Copy & paste or drag & drop one of the .srec files to the S32K116EVB drive

The new application should now be running on the S32K116 EVB. Starting with v1.03 of the MSD Flash Programmer, you can program repeatedly without the need to unplug and reattach the USB cable before reprogramming.

Drag one of the .srec code for the S32K116 EVB board over USB to reprogram the preloaded code example to another example.

NOTE: Flash programming with the MSD Flash Programmer is currently only supported on Windows operating systems. However, the virtual serial port has been successfully tested on Windows, Linux and Mac operating systems.

Using the Virtual Serial Port

- Determine the symbolic name assigned to the S32K116EVB virtual serial port. In Windows open Device Manager and look for the COM port named "PEMicro/Freescale – CDC Serial Port".
- 2. Open the serial terminal emulation program of your choice. Examples for Windows include <u>Tera Term</u>, <u>PuTTY</u>, and <u>HyperTerminal</u>
- 3. Press and release the Reset button (SW5) at anytime to restart the example application. Resetting the embedded application will not affect the connection of the virtual serial port to the terminal program.
- 4. It is possible to debug and communicate with the serial port at the same time, no need to stop the debug.

NOTE: Refer to the OpenSDA User's Guide for a description of a known Windows issue when disconnecting a virtual serial port while the COM port is in use.



INSTALLING S32DS





Download S32DS from:





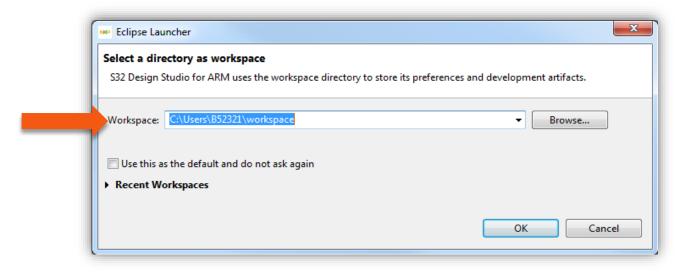
22 EXTERNAL USE

CREATE A NEW PROJECT IN S32 DESIGN STUDIO



Create New Project: First Time – Select a Workspace

- Start program: Click on "S32 Design Studio for ARM v2.0" icon
- Select workspace:
 - Choose default (see below example) or specify new one
 - Suggestion: Uncheck the box "Use this as the default and do not ask again"
 - Click OK





Create New Project: Top Menu Selection

• File – New – Project

File Edit Source Refactor Navigate Search Project Run Processor Expert Window Help New Alt+Shift+N * C* S32DS Application Project V V Quick Access C Quick Access C C Quick Access C	workspace - C/C++ - S32 Design Studio fe	for ARM			
Open File S3205 Library Project Open File S3205 Library Project Open PolyCist from File System S3205 Supplication Project Close All Ctrl-Shift-W See Ctrl-S C/C+Project See Ctrl-S C/C+Project See All Ctrl-Shift-S See All Ctrl-Shift-S Save All Ctrl-Shift-S See All Ctrl-Shift-S See All Ctrl-Shift-S See All Ctrl-Shift-S See All Ctrl-Shift-S Save All Ctrl-Shift-S			essor Expert Window Help		
Open File S3205 Library Project S3205 Application Project Open Projects from File System S3205 Project from Example Image: Console Control S3205 Application Project Swe Control Swe Control Concet Control Concet Control Concet Control An outline is not available. Swe Control Concet Control Concet Control Concet Control See Control Concet Control See Control Concet Control Con	New	Alt+Shift+N ▶ [😤 S32DS Application Project		Quick Access 🗄 😭 🕼
Close Cut+W Close Cut+W Close Cut+W Close Cut+W Close Cut+W Correct C+Project Save A Cut+Suit+S <	Open File	=	S32DS Library Project S32DS Application	on Project	
Close Close Cutrl-With RM Matchile Project with Existing Code Sove Close All Cutrl-Shift-With Cutrl-Shift-With Cutrl-Shift-With Sove All Cutrl-Shift-Sith-Sith-Sith-Sith-Sith-Sith-Sith-Sit	Open Projects from File System	4	S32DS Project from Example		
Close All Ctrl+Shift+W C C + Project Swe Choice C + Project C C/C+Project Swe As Super As <th>Close</th> <th>Ctrl+W</th> <th>Makefile Project with Existing Code</th> <th></th> <th>An outline is not available</th>	Close	Ctrl+W	Makefile Project with Existing Code		An outline is not available
Bord Curles Save As Rename Foreforts Save As Save As Pint Curl P Source File Suite Horkspace Restart Save As Properties Alt-Enter Imain.c. [S32K144_Rev2_EVB_CAN_FD_LCD] 2icd_func.h [S32K144_Rev2_EVB_CAN_FD] Exit Properties Alt-Enter Imain.c. [S32K144_Rev2_EVB_CAN_FD] Exit Console S3 Properties Savers Debugger Console	Close All	Ctrl+Shift+W	C++ Project		
Save As Save As Save All Ctrl+Shift+S Save All Ctrl+Shift+S Revert Source Folder Rename F2 Refrash F5 Convert Line Delimiters To Source Folder Print Ctrl+Project Print Ctrl+Project Switch Workspace File from Template Switch Workspace File from Template Export Class Import Chrl+P Projectis Alt+Enter 1 mainc. [S32K144,Rev2,EVB_CAN_FD_LCD] 2 Icd func.h [S32K144,Rev2,EVB_CAN_FD]	Save	Ctrl+S	C Project		
Save All Ctrl+Shift+S Revet S2DS Application Project Wove Project Wove Project Remane F2 Refesh Convert to a C/C++ Project (Adds C/C++ Nature) Refesh Convert to a C/C++ Project (Adds C/C++ Nature) Project Convert to a C/C++ Project (Adds C/C++ Nature) Switch Workspace F1 Retart F1 Switch Workspace F1 Retart F1 Projectis Alt+Enter I main.c [S32K144_Rev2_EVB_CAN_FD_LCD] 2 Icd func.h [S32K144_Rev2_EVB_CAN_FD_LCD] <					
Revert Project Move Convert Une Delimiters To Refresh Source Folder Convert Line Delimiters To Source Folder Print Cht+PP Header File Switch Workspace Surce File Restart Source File Switch Workspace File from Template Restart Class Import Stample Properties Alt+Enter 1 main.c [\$32K144_Rev2_EVB_CAN_FD] Exit Console 33 Properties @ Debugger Console FreeRTOS Task Aware Debugger Console Project settings Mode settings		Ctrl+Shift+S			
Move Rename P2 Refresh Source Folder Convert Line Delimiters To Pint Print Convert Tile Delimiters To Switch Workspace Pile from Template Restart Pile from Template Gotta Class Import Properties Att+Enter Poletr Import Properties Lid_funch [S32KL44_Rev2_EVB_CAN_FD] Lid_funch [S32KL44_Rev2_EVB_CAN_FD] Lid_funch [S32KL44_Rev2_EVB_CAN_FD] Project is ettings Project is ettings Project is ettings					
Rename Refresh Convert Line Delimiters To Print Curvet Line Delimiters To Print Print Curvet Line Delimiters To Properties Alt+Enter I main.c. [S32K144 Rev2_EVB_CAN_FD] Exit Properties Properties Properties Properties Properties Proper	Move		Y Project		
Refresh Source Folder Convert Line Delimiters To Folder Print Ctrl+P Print Ctrl+P Switch Workspace File Restart Class Import Export Export Class Properties Alt+Enter Imain.c [S32K144,Rev2_EVB_CAN_FD_LCD] Let v Console ½ Properties @ Debugger Console FreeRTOS Task Aware Debugger Console		F2 E	Convert to a C/C++ Project (Adds C/C++ Nature)		
Convert Line Delimiters To Folder Print Ctrl-P Print Ctrl-P Switch Workspace Header File Switch Workspace File from Template Restart Class Import Export Properties Alt+Enter Import Console X3 Properties 🗟 Debugger Console File FreeRTOS Task Aware Debugger Console			Source Folder		
Print Ctrl+P Print Ctrl+P Switch Workspace Pile from Template Restart Class Import Example Export Example Properties Alt+Enter 1 mainc. [S32K144_Rev2_EVB_CAN_FD_LCD] 2 Lcd_func.h [S32K144_Rev2_EVB_CAN_FD] Exit Properties Q Debugger Console Properties Properties Q Debugger Console Properties Properties Q Debugger Console Properties		• -			
Switch Workspace Restart Import Import Export Export Properties Alt+Enter 1 main.c [S32K144_Rev2_EVB_CAN_FD_LCD] 2 lcd_func.h [S32K144_Rev2_EVB_CAN_FD] Exit Console 🔅 Properties R Debugger Console FreeRTOS Task Aware Debugger Console	Drint				
Restart Class Import Export Export Export Properties Alt+Enter 1 main.c. [\$32K144_Rev2_EVB_CAN_FD_LCD] 2 lcd_func.h. [\$32K144_Rev2_EVB_CAN_FD] 2 lcd_func.h. [\$32K144_Rev2_EVB_CAN_FD] Exit		LE CONTRACTOR DE LE CON			
Import Export Export Export Properties Alt+Enter 1 main.e. [S32K144_Rev2_EVB_CAN_FD_LCD] 2 lcd_func.h. [S32K144_Rev2_EVB_CAN_FD] Exit Console Si Properties Reporties Debugger Console FreeRTOS Task Aware Debugger Console Project settings Build settings					
Export Export Properties Alt+ Enter 1 main.c [\$32K144_Rev2_EVB_CAN_FD_LCD] 2 lcd_func.h [\$32K144_Rev2_EVB_CAN_FD] Exit Console © Properties © Debugger Console Settings Project settings Build settings	Restart				
Properties Alt+ Enter 1 main.c. [S32K144_Rev2_EVB_CAN_FD_LCD] 2 lcd_func.h. [S32K144_Rev2_EVB_CAN_FD] Exit	Import		😚 Example		
1 main.c [S32K144_Rev2_EVB_CAN_FD_LCD] 2 lcd_func.h [S32K144_Rev2_EVB_CAN_FD] Exit Console III Properties III Debugger Console Settings Build settings	Export		😚 Other	Ctrl+N	
2 lcd_func.h [S32K144_Rev2_EVB_CAN_FD] Exit Console © Properties 및 Debugger Console Settings Project settings Build settings	Properties	Alt+Enter			
2 lcd_func.h [S32K144_Rev2_EVB_CAN_FD] Exit Console © Properties 및 Debugger Console Settings Project settings Build settings	1 main.c [S32K144_Rev2_EVB_CAN_FD_	_LCD]			
Settings FreeRTOS Task Aware Debugger Console Project settings Build settings	2 lcd_func.h [S32K144_Rev2_EVB_CAN_	L_FD]			
Settings FreeRTOS Task Aware Debugger Console Project settings Build settings	Exit	3	Console 🔀 🔲 Properties 💀 Debugger Console		
🙀 Project settings					
🙀 Build settings					A
🗽 Debug settings					
	4	•			



Create New Project: S32DS Project

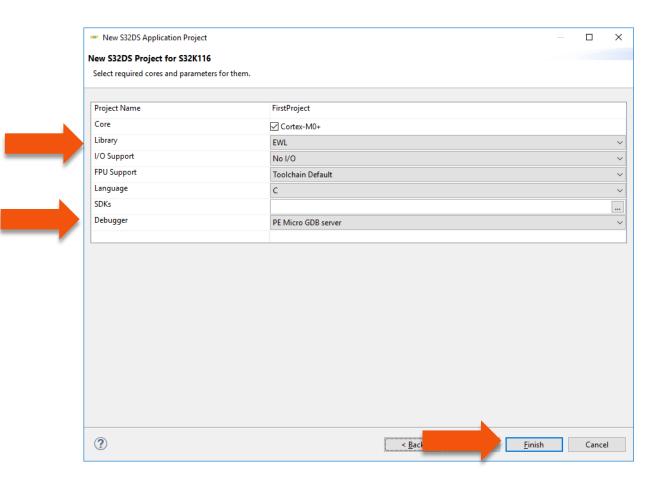
- Project Name:
 - Example: FirstProject
- Project Type:
 - Select from inside executable or library folder
- Next

Project name: FirstProject				
Use default location				
Location: C:\Users\nxa12689\workspaceS32DS.A	ARM\FirstProject			Bi
Processors :	ToolChain Selec	ction:		
type filter text	Core Kind	Name	Toolchain	
 Family KEA Family S32K1xx S32K144 S32K142 S32K148 S32K148 S32K146 S32K146 S32K116 	Description :	Cortex-M0+		
 > Earnily MAC57D5xx > Earnily S32V 	GCC toolchair	is selected		



Create New Project: S32DS Project

- Select Debugger Support and Library Support
- Click Finish





27 EXTERNAL USE

OpenSDA Configuration

28

EXTERNAL USE

- To Debug your project with OpenSDA, it is necessary to select the OpenSDA in the Debug Configuration.
- Select your project, and click on debug configuration

e Edit Source Refactor Navigati	Search Project Ri	Processor Expert Window Help		
	C = x &			Quick Access 🗄 📑 🔤
Project Explorer 💥		là main.c 52 (no launch history)	- 8	🖳 🖓 🛞 Build Tar
Project Explorer 🔀		<pre>mainc S3</pre>		E Outline I I I I I I I I I I I I I I I I I I I
Project Creation New S32DS Project from Example S32DS Application Project S32DS Library Project	Settings Settings Settings Build settings Debug settings	Problems @ Tasks	×	▶
Dashboard 33 Project Creation New S32DS Project from Example S32DS Library Project Build/Debug Build (All) Clean (All) Debug	 ▼ Settings Image: Project settings Image: Build settings 	Processor Expert	× >	



OpenSDA Configuration

- Select the Debug configuration under GDB PEMicro Interface Debugging
- Click on Debugger tab

Create, manage, and run configura	cions
Image: Second Secon	Name: FirstProject_Debug Main Debugger Source Common I OS Awareness Project: FirstProject Bro Specify the number of additional ELF Files you wish to program: 0 Generate ELF Fields
	C/C++ Application: Debug/FirstProject.elf Build (if required) before launching Build Configuration: Debug © Enable auto build © Disable auto build
Filter matched 9 of 11 items	O Use workspace settings Configure Workspace Settings Revert



OpenSDA Configuration

- Select OpenSDA as the interface, if your board is plugged should appear in the Port field.
- Click Apply and debug to finish.

L° 📄 🗙 🕞 券 ▼	Name: FirstProject_Debug Main (参 Debugger) Startup) Startup Source Common # OS Awareness
 C/C++ Application C/C++ Remote Application GDB Hardware Debugging GDB PEMicro Interface Debugging FirstProject_Debug_RAM FirstProject_Release GDB SEGGER J-Link Debugging Launch Group 	Within the bebugger Software Registration Please register your software to remove this message. Register now PEMicro Interface Settings Interface: OpenSDA Embedded Debug - USB Port Port: USB1 - OpenSDA (F7287E07) Select Device Vendor: NXP an K1xx Target: S32K116F128M4 Core: M0 Specify IP Specify Network Caro Additional Options Emergency Kinetis Device Recovery by Full Chip Erase Use SWD protocol Advanced Options Hardware Interface Power Control (Voltage> Power-Out Jack) Provide power to target Regulator Output Voltage Power off target upon software exit, 20
ilter matched 9 of 21 items	Revert Apply

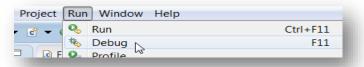


DEBUG BASICS



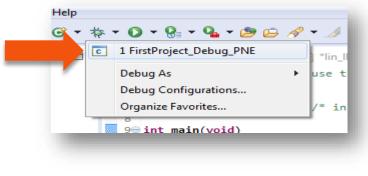
Debug Basics: Starting the Debugger

- Debug configuration is only required once. Subsequent starting of debugger does not require those steps.
- Three options to start debugger:
 - If the "Debug Configuration" has not been closed, click on "Debug" button on bottom right
 - Select Run Debug (or hit F11)



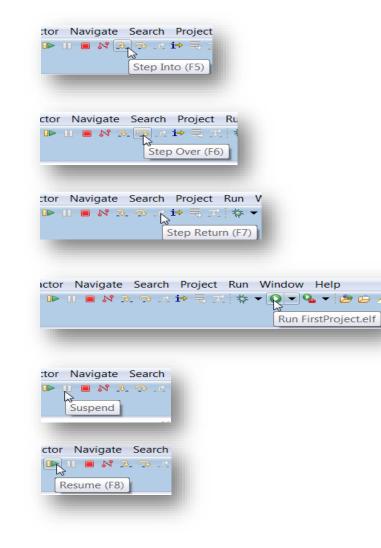
Note: This method currently selects the desktop target (*project.elf*) and gives an error. Do not use until this is changed.

 <u>Recommended Method</u>: Click on pull down arrow for bug icon and select ..._debug.elf target



Debug Basics: Step, Run, Suspend, Resume

- Step Into (F5)
- Step Over (F6)
- Step Return (F7)
- Run
- Suspend
- Resume (F8)





Debug Basics: View & Alter Variables

- View variables in "Variables" tab.
- Click on a value to allow typing in a different value.

lame	Туре	Value
⇔= counter	int	8
		2



Debug Basics: View & Alter Registers

- View CPU registers in the "Registers" tab
- Click on a value to allow typing in a different value
- View peripheral registers in the EmbSys Registers tab

	Name		Value
-	🕯 🛗 General Registe	rs	
	1111 rO		3
	3939 r1		5
	1919 r2		536866944
	1919 r3		8
	1919 -4		0

Arch: cortex	-m0 Vendor: Freescale Cl	hip: SKEAZ1284	Board: none				
egister		Hex	Bin	Reset	Access	Address	Description
a 🗁 IRQ							Interrupt
🔺 💦 S	SC	0x00	0000000	0x00	RW	0x40031000	Interrupt Pin Request Status and Co
	IRQMOD (bit 0)	0x0	0				0: IRQ event is detected only on f
	 IRQIE (bit 1) 	0x0	0				🕲 0: Interrupt request when IRQF se
	IRQACK (bit 2)	0x0	0				IRQ Acknowledge
	 IRQF (bit 3) 	0x0	0				😳 0: No IRQ request
	IRQPE (bit 4)	0x0	0				O: IRQ pin function is disabled.
	IRQEDG (bit 5)	0x0	0				🙆 0: IRQ is falling-edge or falling-e
	IRQPDD (bit 6)	0x0	0				O: IRQ pull device enabled if IRQI
	RESERVED (bit 7)	0x0	0				no description available
- CBC							Courtie Dealers along an an Obra alo



Debug Basics: View & Alter Memory

Add Memory Monitor

Console	Tasks	🖹 Problems	Executables	🚺 Memory 🛿
Monitors		Add Memor	y Monitor	

• Select Base Address to Start at : 0x2000000

NP Mo	nitor Memo	ory			×
Enter a	ddress or e	pression	n to mo	nitor:	
0x200	00000				\sim
?	OK		С	ancel	

• View Memory

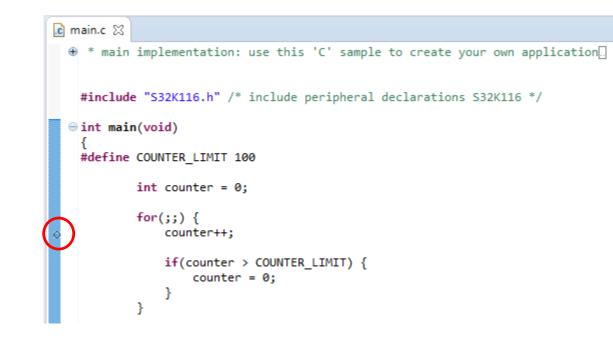
📃 Console 🛛 🙍 Tasks 🖹 Problems	🜔 Executables 🛛 🖳 Debugger Co	nsole 📋 Mem	nory 🛛 🔚 Ou	utline	
Monitors 🕂 💥 💥 0x20000000 : 0x20000000 < Hex> 🖄 🕂 New Renderings					
Ox20000000	Address	0 - 3	4 - 7	8 - B	C - F
	2000000	00000000	00000000	00000000	0000000
	20000010	00000000	00000000	00000000	0000000
	2000020	00000000	00000000	00000000	0000000



Debug Basics: Breakpoints

Add Breakpoint: Point and Click

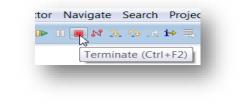
• light blue dot represents debugger breakpoint





Debug Basics: Reset & Terminate Debug Session

- Reset program counter
- Terminate Ctl+F2()



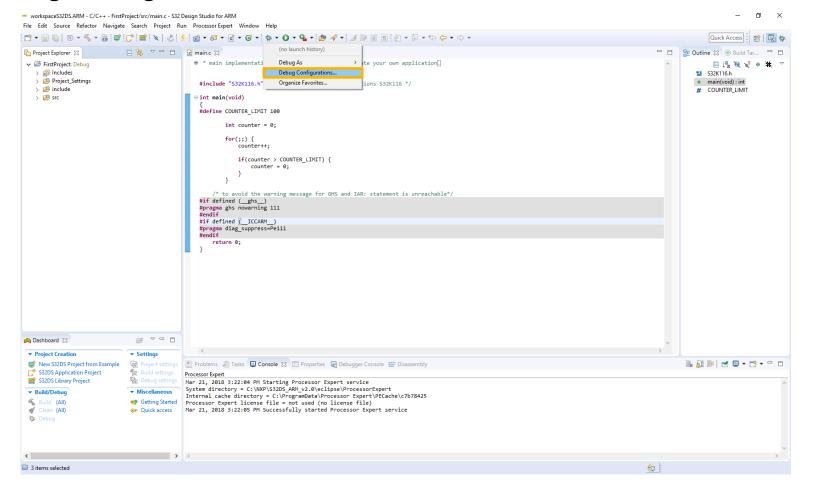


CREATE A P&E DEBUG CONFIGURATION (OPTIONAL)



New P&E debug configuration

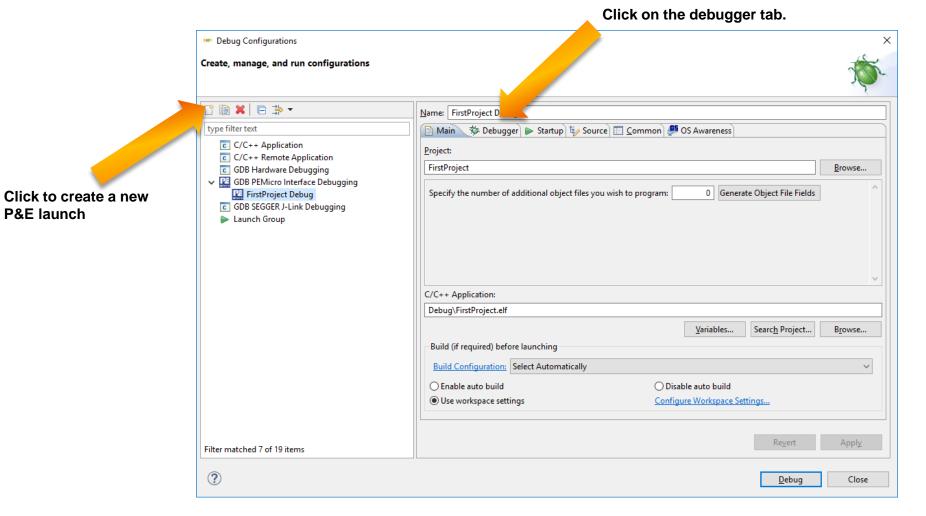
Click in debug configurations





New P&E debug configuration

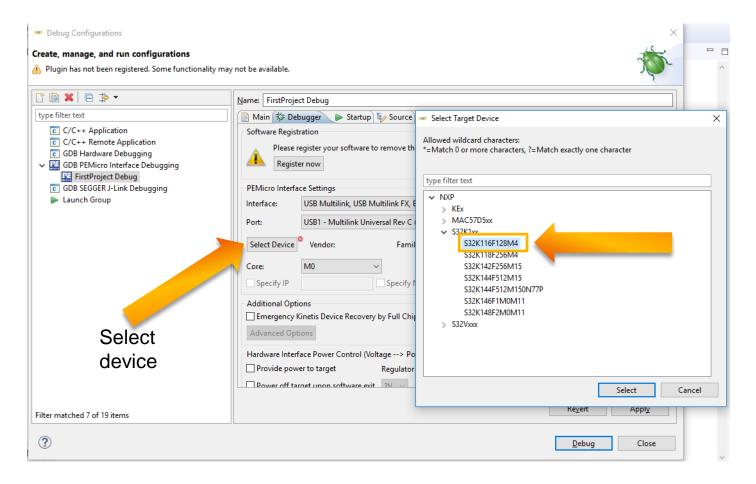
Create a new P&E launch configuration





New P&E debug configuration

• Select the device



• Click Apply and debug your application



42 EXTERNAL USE

USEFUL LINKS



Useful Links

- <u>Cookbook application note</u>. This application note contains a bunch of simple examples of how to use different peripherals.
- <u>S32K1xx community</u>. Visit this site for request support on the S32K1xx products, you can also look for threads that may contain the answer that you are looking for.



Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

NXP: <u>S32K116EVB-Q048</u>