REAL-TIME DRIVERS (RTD) FOR S32K3XX MCUS OVERVIEW AND INSTALLATION GUIDE

Automotive Systems & Applications Engineering Team GPIS-BL, Automotive Processing





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. © 2021 NXP B.V.





AGENDA

- 1. <u>S32K3 SW Enablement Overview</u>
- 2. <u>RTD Architecture Overview</u>
- 3. <u>RTD Installation</u>
- 4. <u>RTD Example Projects</u>
- 5. <u>Create New Projects Based on RTD</u>

S32K3 Software Enablement OVERVIEW



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. © 2021 NXP B.V.

S32K3 SOFTWARE OFFERING: STANDARD AND PREMIUM

Premium SW	Application Specific SW ISELED, etc. Security: OEM specific FW	
	Safety: SAF + SCST	
	Security: Standard FW	
Standard	Real-Time Drivers (RTD) for AUTOSAR and non-AUTOSAR	
	Including Inter-Platform Comms Framework (IPCF) and Crypto Driver	
Standard HW	S32K3	

PREMIUM OFFERING:

Application Specific SW: ISELED, etc. Price adder on top of silicon price for selected PN

Premium Security: OEM specific firmware

Price adder on top of silicon price for selected PN

Premium Safety: S32 Safety Software Framework (SAF) + Structural Core Self Test (SCST) One-time license fee for combined SAF + SCST

STANDARD OFFERING:

Included in silicon price

NXP SOFTWARE BASED ON REAL-TIME DRIVERS



SERVICES / APPLICATION SPECIFIC SOFTWARE

SECURITY & OTA

OS / DRIVERS / SAFETY

Real-Time Drivers (RTD)

Enhanced an updated AUTOSAR MCAL and non-AUTOSAR SDK

ISO 26262 compliant for all SW layers, production grade Full compliance and coverage for both HW features and HW lps, including Crypto Driver Driver examples with default configurations

MULTI CORE MANAGEMENT

S32K3 MCUs Family

Unmatched **HW scalability** across General-Purpose & Integrated Solutions MCUs combined with

Real-Time Drivers (RTD) flexibility

One SW development environment

independently by the project requirements and specifications

One configuration tool and one driver set

MEANING: less time and higher optimization of functionalities

REAL-TIME DRIVERS (RTD) NEW AND INNOVATIVE DRIVERS SET FOR AUTOSAR AND NON-AUTOSAR SOLUTIONS

Specifically focused on Real-Time Software Targeted for Arm[®] Cortex[®]-M core based MCUs Single package for each S32 MCU or Processor For AUTOSAR and NON-AUTOSAR systems

ENHANCEMENTS:

- ISO 26262 Compliance for all SW layers
- AUTOSAR functionalities (e.g. multicore, user mode) are expanded also to non-AUTOSAR environment (previously only available for AUTOSAR)
- Full IP and features coverage for both AUTOSAR and AUTOSAR
- Possible integration on platform level of middleware (FATFS for EEP, FEE for FLS *derived from MCAL*) and stacks (LIN, NFC, TCIP, ..)
- Driver examples with default configurations



PUBLIC 5

Real-Time Drivers (RTD) ARCHITECTURE OVERVIEW



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. © 2021 NXP B.V.

REAL-TIME DRIVERS (RTD) SOFTWARE PACKAGE FOR S32 MICROCONTROLLERS AND PROCESSORS

High-Level Interfaces (HLI) based (and enhanced) on former MCAL environment



- Production-qualified software abstraction of complex hardware features
- Automotive-grade and production ready: SPICE/CMMI Level 3 compliant, MISRA 2012 tested
- Developed using SPICE Level 3 and ISO 26262 standard compliant process
- Integration with NXP S32 Design Studio (S32DS) IDE
- · Supports multiple toolchains: GCC, GHS, IAR

· Full coverage of IPs

through *extensions*: extra APIs added to standard ones – e.g. *Adc_EnableCtuControlMode* to support configuration and functions related to CTU control mode of ADC unit

- AUTOSAR 4.4:

- Multicore
- LIN "follower" support
- Security: TLS, Key Manager, Security Event Memory
- Documented source code, examples, cookbook & demos for fast application start-up, using drag-drop functionality

REAL-TIME DRIVERS (RTD) SOFTWARE PACKAGE FOR S32 MICROCONTROLLERS AND PROCESSORS

Low-Level Interfaces (LLI) based (and enhanced) on former SDK environment



 Automotive-grade and production ready: SPICE/CMMI Level 3 compliant, MISRA 2012 tested

- · Complete drivers offering:
 - Low-level drivers for all MCU peripherals: FlexIO, UART, CAN FD, ISELED, etc.
 - Optional middleware: LIN, TCP/IP, NFC
 - Drivers for complementary NXP ICs: e.g. SBC
- · FreeRTOS operating system
- Integration with NXP S32 Design Studio (S32DS) IDE and 3rd party IDEs: KEIL, GHS Multi, IAR
- · Supports multiple toolchains: GCC, GHS, IAR
- Documented source code, examples, cookbook & demos for fast application start-up, using drag-drop functionality

REAL-TIME DRIVERS (RTD) SOFTWARE PACKAGE FOR S32 MICROCONTROLLERS AND PROCESSORS

Additional specific SW packages and Configuration Tools

One configuration tool can be selected for the development: EB tresos or S32 Config Tool (S32CT) \rightarrow aiming to develop S32CT with AUTOSAR functionalities



Stacks and Libraries available in both AUTOSAR and non-AUTOSAR contexts. Can be plugged into:

- High-Level Interface (AUTOSAR compliant)

- Low-Level Interface

Demo application code available for:

- Provided libraries & stacks
- High-Level Interface (AUTOSAR compliant) layer

- Low-Level Interfaces layer

ERROR MANAGEMENT BETWEEN MCAL/SDK AND RTD

The error detection and reporting mechanism for RTD is tailored for the target application type:

· HL API

- For the high-level layer, which is mainly intended for usage in AUTOSAR applications, error management follows the standard specifications for DET & DEM. RTD provides a "stub" implementation of these AUTOSAR modules, which can be used or overwritten by the customer application.
- Most of the APIs in consisting the AUTOSAR compliant HL API return Std_ReturnType (E_OK/E_NOT_OK). The specific error can then be retrieved by calling the dedicated APIs in DEM/DET.

TIPS: Development errors are always reported using **DET**;

runtime may be reported using **DEM** or **DET**, depending on the impact they have on the application integrity.

· IP API

The errors reported by the IP layer are still split in two categories:

- Development errors: usually parameters checking but not only, these errors are checked using DevAssert function; in case an error is detected, this will halt the program execution in the default implementation. The default behavior of DevAssert function can also be overwritten by the application. This mechanism is almost identical to the DEV_ASSERT functionality in older SDK, the only improvement being that these statements are now enabled/disabled for each driver separately, as opposed to the SDK approach where this was a global configuration (check the picture below).
- Runtime errors: as opposed to the SDK, where all runtime errors reported by drivers were grouped in the generic enumeration called *status_t*, the RTD define a set of runtime errors per driver. The naming convention for these errors is <*IP_Name>_Ip_StatusType*, as shown is the example below:
 - Each driver defines the set of errors that can be reported by the controlled IP; these errors can either be used by the non-AUTOSAR application implemented on top of the IP layer for retrieving the status of the driver, or further fed into the high-level state machine of the layers on top.

CONFIGURATION FILES DIFF BETWEEN MCAL/SDK AND RTD

The configuration data files are now split following a more granular approach to ensure the possibility of using the IP drivers stand-alone.

From a functional point of view, all the data that is needed in an AUTOSAR application will be exported through the **HLD** files, so nothing changes in the application flow.

MCAL \$32K1/\$32K2	RTD S32K3	Comments
<mdl>_Cfg.h</mdl>	<mdl>_Cfg.h <mdl>_lpw_Cfg.h <lp>_Cfg.h</lp></mdl></mdl>	Contains precompile parameters used in the driver, usually defines and constants, extern declarations and data types
<mdl>_Cfg.c</mdl>	<mdl>_Cfg.c <mdl>_lpw_Cfg.c <lp>_Cfg.c</lp></mdl></mdl>	Static configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <i><mdl>_Init</mdl></i> function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one inoudle even for Precomplies variant.

<mdl>_PBcfg_<variant>.c</variant></mdl>	<mdl>_PBcfg _<variant>.c <mdl>_lpw_PBcfg _<variant>.c <lp>_PBcfg_<variant>.c</variant></lp></variant></mdl></variant></mdl>	There is one file for each variant. The name of the file contains the name of the variant, as defined in the EcuC. This file contains the configuration structure used by the driver that have variant aware members. Each file contains the configuration parameters for its corresponding variant. All parameters and/or structures that are not variant aware and were generated once in the <mdl>_Cfg.c file are referenced in the structures from <mdl>_PBcfg_<variant>.c files if needed. The configuration structures are used in all variants.</variant></mdl></mdl>
<mdl>_PBcfg_<variant>.h</variant></mdl>	<mdl>_PBcfg _<variant>.h <mdl>_Ipw_PBcfg _<variant>.h <ip>_PBcfg_<variant>.h</variant></ip></variant></mdl></variant></mdl>	It was created to export the extern declaration of each configuration structure, to be used when calling <i><mdl>_Init</mdl></i> in the application. There is one file for each variant. The name of the file contains the name of the variant, as defined in the EcuC.

NP

MAPPING BETWEEN RTD DRIVERS AND \$32K3XX PERIPHERALS

- Add UART support as complex driver for MCAL 4.4
- Add Flexio_SENT to support SENT communication
- Oslf can support FreeRTOS and AUTOSAR OS as well as bare-metal timer, but no semaphore, mutex and queue support
- REG_PROT is included in BASE
- Functional safety related driver, such as MPU and XRDC are included in Resource Manager as complex driver
- FLS also includes QuadSPI external Flash memory, EEP is a standard MCAL for D-Flash emulated EEPROM implementation.
- All RTD drivers have timeout and multicore support per AUTOSAR 4.4 standard and S32K3xx multi-core architecture required.

Real Time Driver	S32K3XX IP	Comments
DEM	-	Diagnostics Event Manager
		Reference code provided by NXP to be used in Non Autosar
		applications.
		To be replaced to Autosar Standard Implementation for Autosar applications
DET	-	Default Error Tracer
		Reference code provided by NXP to be used in Non Autosar
		applications.
		To be replaced to Autosar Standard Implementation for Autosar
		applications
ECUC		Ecu Configuration – add support for multicore
		Reference code provided by NXP to be used in Non Autosar
		To be replaced to Autosar Standard Implementation for Autosar
		applications
ECUM	-	Ecu Manager
		Reference code provided by NXP to be used in Non Autosar applications.
		To be replaced to Autosar Standard Implementation for Autosar
DTC		applications
RIE	-	Run Time Environment – implements exclusive areas
		applications
		To be replaced to Autosar Standard Implementation for Autosar
Oslf	1	OS Abstraction laver
		Integrates support for FreeRTOS and Autosar OS
Resource	-	Resource driver – collection for all derivatives features
BASE		Base driver Collection of header files
	REG_PROT	A subset of driver's files provided by NXP can be used in Non Autosar
		applications
		A subset or driver's files provided by NXP can be replaced to Autosar Standard Implementation for Autosar applications
мси	MC_CGM	Microcontroler Unit Driver
	FIRC	Provides services for basic microcontroller initialization, mode
	SIRC	management and clock management
	PLLDIG	
	FXOSC	
	MC_RGM	
	MC_ME	
	SXOSC	
	RAM Controller	
	MC_PCU	
	PMC	
PLATFORM	MSCM	Platform – Complex Device Driver
	NVIC	Integrates functionalities specific to platform and interrupt management
	INTM	management
	MCM	
RM	MPU	Resource Manager – Complex Device Driver
	XRDC	

	SEMA42					
	PRAMC					
	Pflash					
	VIRT_Wrapper					
	XBIC - MCR reg					
	XBAR					
MCL	eDMA	Microcontroller Library – Complex Device Driver				
	LCU	Offers DMA and Cache functionalities				
	DMAMUX					
	TRGMUX					
	Cache_M7					
PORT	SIUL2	PORT Driver				
		Provides the services for initializing the whole structure of PORT driver				
DIO		Digital Input Output Driver				
		Provides services for accessing the microcontroller's hardware pins				
EEP		EEPROM Driver				
FLS	QuadSPI	Flosh Driver				
FL3	C40					
	Pflash					
ICU	eMIOS	Input Capture Unit Driver				
	SIUL2					
	CMP					
	WKPU					
ocu	eMIOS	Output Capture Unit Driver				
GPT	eMIOS	General Purpose Timer Driver				
	PIT					
	RTC					
	STM					
PWM	eMIOS	Pulse Width Modulation Driver				
	FlexIO_PWM					
QD	eMIOS	Quadrature Complex Device Driver				
12C	LPI2C	I2C Complex Device Driver				
	FlexIO_I2C					
ETH	ENET	Ethernet Driver				
UART	LPUART	UART Complex Device Driver				
LIN	LPUART	Lin Driver				
SPI	LPSPI	Serial Parallel Interface Driver				
	FlexIO_SPI					
CAN	FlexCan	Can Driver				
ADC	ADC	Analog Digital Comparator Driver				
		There of the comparator of the				
	BCTU					
Crypto	HSE_M	Crypto Driver				
	MU					
WDG	SWT	Watchdog Driver				
SENT	FlexIO_SENT	SENT driver				
CRC	CRCU	CRC Complex Device Driver				

Table 1 Mapping between Drivers and S32K3XX Drivers

PUBLIC

12



CUSTOMER FACING: ONE PACKAGE FOR AUTOSAR & NON-AUTOSAR ENVIRONMENTS

RTD offers both abstracted/standardized interfaces and HW specific interfaces (exported by the IPL Interface) These two interface types are exclusive, cannot be used at the same time



IPL (IP Layer):

- Peripheral specific layer implementing support for all IP features
- Constant for the same IP across platforms.
- Dedicated to export all hardware functionalities
- IP layer to come with standalone ISO 26262 compliance

HL (High Layer):

- Implements the standard APIs described into the AUTOSAR specifications
- Implements the APIs extensions based on:
 - Specific customer requirements
 - IP features exposed from layer below



REAL-TIME DRIVERS (RTD) – ARCHITECTURE

Depending on the context wanted to be used by the upper layer, a specific interface must be used:

AUTOSAR context:

- High Level Interface usage

non-AUTOSAR context:

- High Level Interface usage

or

- IP Layer (ex-SDK)





REAL-TIME DRIVERS (RTD) – CONFIGURATION SCHEME



PUBLIC 15

REAL-TIME DRIVERS (RTD) – LAYERED ARCHITECTURE



S32DS and RTD for S32K3 INSTALL GUIDE



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. © 2021 NXP B.V.

1) LOG IN WITH YOUR CREDENTIALS



Software Licensing and Support

2) DOWNLOAD: S32K3 STANDARD SOFTWARE PACKAGE

Login your account on NXP website, and download the **S32K3 Standard Software** from: https://www.nxp.com/webapp/swlicensing/sso/downloadSoftware.sp?catid=SW32K3-STDSW-D

Product Information

Automotive SW - S32K3 Standard Software

Your choice contains a suite of products. Please select one of the product lines below: To register a New Product please click on the button below

Register

Automotive SW - S32K3 - HSE Firmware Automotive SW - S32K3 - Inter-Platform Communication Framework Automotive SW - S32K3 - Model-Based Design Toolbox Automotive SW - S32K3 - Platform Software Integration Automotive SW - S32K3 - Real-Time Drivers for Cortex-M Automotive SW - S32K3 - S32 Design Studio Automotive SW - S32K3 - S32 FreeMASTER Automotive SW - S32K3 - Stacks Automotive SW - Elektrobit Tresos Studio / AUTOSAR Configuration Tool Make sure to login your account first.

Then click link to access the available **Automotive SW – S32K3 Standard Software**

3) DOWNLOAD: S32DS (S32DS 3.4) AND K3 RTD

Download latest version of **RTD**

	DUICTS APPLICATIONS DESIGN SUPPORT COMPANY						
	SOULS AFFEICATIONS DESIGN SOFFORT COMPANY						
NXP > Software & Support > F	Product List	Current	Previous				
Software & Support Product List		Version	Version - Description				
Product Search	Welcome to the premier delivery, update, and software management service; designed to help yo effectively manage your rapidly changing software from a secure, personalized website.	1.0.0	 S32K3 Real Time Drivers Version 1.0.0 This is the NXP S32K3 Real Time Drive used standalone, or the update site car software included in this release has R) ers Version 1.0.0 re n be installed on to TM quality status ir	elease for the S32K3 family of processors. It can be p of S32 Design Studio IDE v3.4 Service Pack 2. A n terms of testing and quality documentation with	Download Log	
Recent Product Releases	Order History Recent Product Releases To access an item, select a product below.		some driver's exceptions which are qualified as BETA (CRYPTO driver) and EAR (I3C driver). RTM qualified drivers can be used in production.				
Recent Updates	Automotive SW - AUTOSAR MCAL / ISO 26262						
Licensing License Lists Offline Activation	Automotive SW - AUTOSAR MCAL / QM Automotive SW - AUTOSAR OS Automotive SW - Core Self Test with Instruction Coverage Metric Automotive SW - Fault Graded Core Self Test		Download S	32 Desig	n Studio Installer		
Automotive SW - LIN Stack Software		Current Devices					
Download Help	Automotive SW - Middleware for General Purpose and Integrated Solutions						
	Automotive SW - S32K2TV Standard Software		Version — Description				
Table of Contents	Automotive SW - S32K3 Standard Software	\rightarrow	3.4 - S32 Design Studio for S	S32 Platform v.3,4	with support for S32K3 devices		
FAQs	Automotive SW - Security Software Automotive SW - Software Development Kit for Cortex-M Automotive SW - Software SoftTast	This is the S32 Designstudio for S32 Platform version 3.4 release with support for S32K3 devices . All tools included in this release have RTM quality status in terms of testing and quality documentation .					
	Automotive SW - Structural Core Self Test Automotive SW - Vision Software					⊒	
	NXP Software	+ File Desc	iption	File Size 🌲	File Name	\$	
	Automotive SW - Platform Software Integration	+ S32 Desig for offline	n Studio 3.4 development packages use	3.7 GB	₽ SW32 S32DS 3.4.0 D2012.zip		
Autom	otive SW - S32K3 - HSE Firmware	+ S32 Desig for offline	n Studio 3.4 development packages use, support for S32K3 family	1.3 GB	₽ SW32K3 S32DS 3.4.0 D2012.zip		
Autom	otive SW - S32K3 - Platform Software Integration otive SW - S32K3 - Real-Time Drivers for Cortex-M	+ S32 Desig	n Studio 3.4 Release Notes	72.5 KB	₽ S32DS Release Notes 3.4.0.pdf		
Autom	otive SW - S32K3 - S32 Design Studio otive SW - S32K3 - Stacks	+ S32 Desig Package	n Studio 3.4 S32K3 Development Release Notes	42.2 KB	♣ S32K3xx Development Package Release	Notes 3.4.0.pdf	
Autom	otive SW - Elektrobit Tresos Studio / AUTOSAR Configuration Tool	+ S32 Desig	n Studio Installation Guide	1.2 MB	₽ S32DS Installation Guide 3.4.0.pdf		
		+ S32 Desig	n Studio v3.4 Linux installer	1.1 GB	₽ \$32D\$.3.4 b201217 linux.x86 64.bin		
		+ S32 Desig	n Studio v3.4 Windows installer	1.5 GB	₽ \$32D\$.3.4 b201217 win32.x86 64.exe		



NP

- "Automotive SW - S32K3 - S32 Design Studio"

- Step 1: Run "S32 Design Studio v3.4 Windows installer"
 - "S32DS.3.4_b201217_win32.x86_64.exe"
 - License Keys => Activation Code.
- Step 2: : Install "S32 Design Studio 3.4 development packages for offline use, support for S32K3 family" in S32DS3.4:

S32DS.3.4 b201217 win32.x86 64.exe

After the installation was completed, open the S32DS3.4 and go to Help => Install New Software







- Step 3: Look for the available software
 - Drop down on the menu
 - Select "S32DesignStudio http://www.nxp.com/lgfiles/updates/Eclipse/S32DS_3.4"

🗱 Install		— D >	×
Available Software			
Select a site or enter the location of a site.			
Work with: ⁰ type or select a site	51	d Manage	
type or select a site type filter All Available Sites com.nxp.RTD.S32K3XX.repository - https://www.nxp.com/lgfiles/updates/Eclipse/APSW/S32 GNU ARM Eclipse Plug-ins - http://gnuarmeclipse.sourceforge.net/updates GNU ARM Eclipse Plug-ins - http://gnuarmeclipse.sourceforge.net/updates http://www.pemicro.com/eclipse/updates/com.pemicro.debug.gdbjtag.pne.updatesite http://www.pemicro.com/eclipse/updates/com.pemicro.debug.gdbjtag.pne.updatesite/4.7. http://www.pemicro.com/eclipse/updates/com.pemicro.debug.gdbjtag.pne.updatesite/4.9. LauterbachGmbHUpdateSite - http://www.lauterbach.com/eclipse PEMicro - http://www.pemicro.com/eclipse/updates RTD - http://www.nxp.com/lgfiles/updates/Eclipse/S32DS_3.4 SW32_S32DS_3.4.1_D211015 - http://www.nxp.com/lgfiles/updates/Eclipse/S32DS_3.4	2DS_3.4	Select All Deselect All	1
iSystemDebugger - http://www.isystem.si/eclipseUpdate/debuggerJuno42			
			÷
Show only the latest versions of available software	Hide items that are already installed		
Show only software applicable to target environment	what is <u>aneady instanco</u> :		
Contact all update sites during install to find required software			

PUBLIC 22

- Step 4: Choose the S32DS S32K3xx development package
- Select the S32DS S32K3 development package 3.4.1 and click Next



PUBLIC

23

- Step 5: Install the S32DS S32K3xx development package
 - Select the option "Update my installation to be compatible with the items being installed" and click Next



- Step 5: Install the S32DS S32K3xx development package
 - Now you can review the installation details. Click Next





- Step 5: Install the S32DS S32K3xx development package
 - Accept the license and click Finish

🗱 Install	_		×
Review Licenses			
Licenses must be reviewed before the software can be installed. This includes licenses for software required to com	plete the install.	C	Harr
Licenses:	License text:		
 Software License LA_OPT_NXP_Software_License v1 August 2018 LA_OPT_NXP_Software_License v11 February 2020 LA_OPT_NXP_Software_License v17 October 2020 NXP SOFTWARE LICENSE AGREEMENT This product is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installing the product. It is subject to the End User License Agreement accepted when installed the product accepted accep	 *Software License* This Software and accompanying documentation are protected by United States Copyright law and also by International Treaty provisions. Copyright notices have been included in the Software. P&E Microcomputer Systems, Inc ("PEMICRO") grants to the individual or entity ("LICENSEE") who received this Software from PEMICRO, either directly or indirectly through one of our authorized partners, a non-exclusive and non-transferable License to use this Software with PEMICRO's Cyclone, Multilink, Tracelink, OpenSDA, and OSJTAG products solely to develop, evaluate, or demonstrate source code. Under no circumstance may this product be used for production line programming or be used as part of a manufacturing process without PEMICRO's prior written consent. All rights, title and interest in the Software are and shall remain with PEMICRO. Under no circumstances may you copy this Software or documentation for the purpose of distribution to others without prior written authorization from PEMICRO. Under no conditions may you remove the copyright notices from this Software or documentation. LICENSEE may not reverse engineer, decompile, or disassemble the Software, nor attempt in any other manner to obtain the source code. The Software is licensed as a single product. Its component parts covered by this License may not be separated for use on more than one computer, nor otherwise used separately from the other parts. LICENSEE may not rent or lease the Software to someone else. LICENSEE may not I accept the terms of the license agreements I do not accept the terms of the license agreements 		~
0	< Back Next > 2 Finish	Cancel	



- Step 1: Install "S32K3 Real Time Drivers (RTD)" in S32DS3.4:
 - Open the S32DS3.4 and go to Help => Install New Software
 - Click on the "Add" button.

			🔀 Install				_	
			Available Software					
			Select a site or enter the loca	ation of a site.				O.
Hel	Ip							
**	Getting Started	5	Work with: type or select a sit	te			1 Add	Manage
•	Help Contents		type filter text					Select All
*	Show Contextual Help		Name		Version			Deselect All
8.	Show Active Keybindings Tips and Tricks Cheat Sheets	Ctrl+Shift+L		.tea.				
-	Check for Updates							
69 10	Install New Software 2 Installation Details							
國國	S32DS Extensions and Updates About S32 Design Studio for S32 Platform							
_	NXP Licenses		Details					
_								\$
			Show only the latest version	is of available software	✓ Hide items that	are already installed		
			Group items by category		What is <u>already</u> i	nstalled?		
			Show only software application	ble to target environment				
			Contact all update sites duri	ing install to find required software				

NC

- Step 2: Look for the available software
 - Click on the "Archive" button and look for the SW32K3_RTD_4.4_1.0.0_DS_updatesite_D2110 file inside your files. Remember that we downloaded the S32K3 RTD 1.0 package from the NXP official website (Slide 20)



- Step 2: Look for the available software
 - Select a name to identify this new package. For example, "S32K3 RTD 1.0"
 - Remember to select the SW32K3_RTD_4.4_1.0.0_DS_updatesite_D2110 file and click "Add"

🔡 Install						
Available Software						
Select a site or enter the location of a site.						
Work with: type or select a site		~	Add	Manage		
type filter text				Select All		
Name	Version			Deselect All		
	🞇 Add Repository	– 🗆 X				
	1 S32K3 RTD 1.0	Local				
	jar:file:/C:/SW32K3_RTD_4.4_1.0.0_DS_updatesite_D2110.zip!/	Archive				
	⑦ 3 Add	Cancel				
Details				0		
Show only the latest versions of available software	Hide items that are alre	eady installed				
Group items by category	What is <u>already installed</u>	1?				
Show only software applicable to target environment						
✓ Contact all update sites during install to find required	software					
		🔒 SW32K3_F	RTD_4.4	4_1.0.0_	DS_updatesite_D2110	.zip
(?)		< Back Next >	Finish	Cancel		

- Step 3: Choose the S32DS S32K3xx RTD package
 - Select the S32 Design Studio S32K3 RTD package and click Next



PUBLIC 30



- Step 4: Install the S32DS S32K3xx RTD package
 - Now you can review the installation details. Click Next

🔀 Install				—	
Install Details Review the items to be installed.					
Name > % RTD S32K3XX % RTD S32K3XX 1.0.0	Version 1.0.0.202110071857 1.0.0.202110071857	ld com.nxp.RTD.S32K3XX.root com.nxp.RTD.S32K3XX.1.0.0			
Size: Unknown Details					^ ~
(?)			< Ba 1 Next >	Finish	Cancel

NP

- Step 4: Install the S32DS S32K3xx RTD package
 - Accept the license and click Finish

🔀 Install	— — X
Review Licenses	
Licenses must be reviewed and accepted before the software can be installed.	() Harris
Licenses:	License text:
NXP Automotive Software License Agreement v1.8	NXP Automotive Software License Agreement v1.8 MPORTANT. Read the following NXP Automotive Software License Agreement
0	< Back Next 2 Finish Cancel

PUBLIC 32

6) CONFIRM: RTD FILES

After installation, the **RTD for S32K3** files can be found in the following path: *C:\NXP\S32DS.3.4\S32DS\software\PlatformSDK_S32K3_2021_10*

S32DS.3.4 > S32DS > software > SW32	K344_RTD_4_4_081_D2011	
Name	Date modified	Туре
Adc_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Base_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Can_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
CanIf_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Crc_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Crylf_TS_T40D34M811R0	1/15/2021 4:00 PM	File folder
Crypto_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Csm_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Dem_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Det_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Dio_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
EcuC_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
EcuM_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Eth_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Ethlf_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
EthSwt_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
EthTrcv_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
Fee_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
FIs_TS_T40D34M811R0	1/15/2021 4:00 PM	File folder
Gpt_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
I2c_TS_T40D34M8I1R0	1/15/2021 4:00 PM	File folder
	1/15/2021 4:00 DM	E1 6 1 1



REAL-TIME DRIVERS (RTD) for S32K3 EXAMPLE PROJECTS



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. © 2021 NXP B.V.

34

IMPORT EXISTING EXAMPLE PROJECTS

• File \rightarrow Import \rightarrow General \rightarrow Existing Projects into Workspace \rightarrow Next



PUBLIC 35

IMPORT EXISTING EXAMPLE PROJECTS

Example projects path:

 $C: WXP \ S32DS. 3.4 \ S32DS \ software \ Platform SDK_S32K3_2021_10 \ SW32K3_RTD_4_4_1_0_0_D2110 \ Can_TS_T40D34M1010R0 \ examples \ S32DS \$

"examples\EBT" is for **EB tresos** example project and "examples\S32DS" is for **S32DS** example

Recommend to select "Copy projects into workspace" to save original example project for reference

Available SDK examples:

- adc_ip_example
- Can
- Dio_example_DS
- Eth_Example_DS_001
- FLS_IP_C40_Example_001
- FLS_IP_QSPI_Example_001
- Gpt_example_DS
- *l*2c_CodeDrop_example_DS
- Icu_example_DS
- dma_ip_transfer
- Power_lp_Example_CT
- Clock_lp_Example_CT
- Port_example_DS
- Ip_Lpspi_example_DS
- swt_ip_interrupt

🔡 Import	_		×
Import Projects			7
Select a directory to search for existing Eclipse projects.			-
Select root directory: C:\NXP\S32DS.3.4\S32DS\software\PlatformSDK S32K3 2021 03\SW32K3 RTD 4 4 0 9 0 D2103\Can TS T40D34M9I0R0\examples\S32DS	1	Browse	
O Select archive file:	1	Browse	
Projects:			
AN_example_CT (C:\NXP\S32DS.3.4\S32DS\software\PlatformSDK_S32K3_2021_03\SW32K3_RTD_4_4_0_9_0_D2103\Can_TS_T40D34M9I0R0\examples\S32DS\CAN_example_CT]	Select All	
FlexCAN_example_CT (C:\NXP\S32DS.3.4\S32DS\software\PlatformSDK_S32K3_2021_03\SW32K3_RTD_4_4_0_9_0_D2103\Can_TS_T40D34M9I0R0\examples\S32DS\FlexCAN_example_CT	D	eselect A	u
		Refresh	
Options			
Search for nested projects			
□ Close newly imported projects upon completion			
Hide projects that already exist in the workspace			
Working sets			
Add project to working sets		New	
Working sets:	S	elect	
Sack Next > Finish		Cancel	

CREATE AN S32DS PROJECT FROM EXAMPLE

File \rightarrow New \rightarrow S32DS Project from Example

532 DS	workspaceS32DS.3.4 - S32K344_RTD0	090_TestPrj/src/ma	in.c	c - S32 Design Studio for S32 Platform
File	Edit Source Refactor Navigate	Search Project (Con	figTools Run Window Help
	New	Alt+Shift+N >	¢	S32DS Project from Example Ctrl+Alt+E
•	Open File Open Projects from File System Recent Files	>	2 2 2	S32DS Library Project Ctrl+Alt+L S32DS Application Project Ctrl+Alt+A Makefile Project with Existing Code Ctrl+Alt+A
	Close Close All	Ctrl+W Ctrl+Shift+W		C/C++ Project Project
1 1 1 1	Save Save As Save All Revert	Ctrl+S Ctrl+Shift+S	61 61 61 61 61 61	Source Folder Folder Source File Header File
1 1 1	Move Rename Refresh	F2 F5	© ©	File from Template Class
	Convert Line Delimiters To	>		Other Ctrl+N
2	Print	Ctrl+P	Г	used strictly in accordance wit
2	Import Export Migrate			 accepting such terms or by down using the software, you are agr comply with and are bound by, s bound by the applicable license activate or otherwise use the s
	Properties	Alt+Enter		
	Switch Workspace Restart Exit	>		⊕/** * @file main.c *

Select the RTD version and example project



GENERATE CODE FOR EXAMPLE PROJECT

Double click the "mex" file to open SDK configuration tool.

	WorkspaceS32DS.3.4 - FlexCAN_example_CT/src/	nain.c - S32 Design S	tudio for S32 Platform			– 🗆 X
✓ № FlexCAN_example_CT: Debug_FLASH	File Edit Source Refactor Navigate Search Proje	t ConfigTools Perir	oherals Run Window Help			
> 🗊 Includes	🔁 🔻 🗟 🕼 🕴 FlexCAN_example_CT	🗸 🐐 🔒 Update	Code Functional Group BOARD InitPeriph	erals 🔹 💌 🖻 🗟 🖉 🗣 🍬 🖉 🕶 🎉 🔍 🖓 👻	• \$ \$ •	○▼ ⊡ Q.ie: ⊑ # will > ≣ # #
> 🔑 Project Settings	Components # # Peripherals	Start Cosif_1	FlexCAN_43_1 🛱		4 0v	verview 🖾 Code Preview 👛 🗖
> CA generate	type filter text O 14	IP FlexCAN	Driver [Drivers]		D^ ~	Configuration - General Info
	MCAL	Name FlexCAN	11	Custom name		Configuration - HW Info
> 🦰 generate/src		Mode General No	- ode		v F	Processor: 532K344 Part number: 532K344 257BGA
V 📇 src	Drivers			*		Core: Cortex-M7
> 🚺 main.c	FlexCAN_43_1 IntCtrl_lp_1 osif_1	Electronic and and	Free Fler CAN Convert Treese & Conference	Preset Custom	<u> </u>	DK Version: PlatformSDK_S32K3_2021_03
description.txt	Siul2_Port_1	FlexCAN configur	ations HexCAN General Timeout Configuration		~	Project
ElexCAN example CT.mex		+ × ~ •			^	Peripherals
A seeders tot		0	FlexCAN Hardware Channel	FL XCAN_0		Configures the initialization of the SDK peripheral drivers.
readme.txt			Name	FlexCAN_Config0		Ψ.
			Number Of MB	16		Committed and
			FlexCAN Rx FIFO filters number	8 x FIFO Filters 🗸		Update code enabled
			Enable Legacy RxFiFO		6	generate\include\FlexCAN_lp_Cfg.h
			Enable FD Can		E	generate\include\FlexCAN_IRD_InitPeripherals_PBcfg.h
			HexCAN operation modes	Lopp-back mode	6	generate\include\IntCtrl_Ip_Cfg.h
Step1:			Ry EIEO Transfer Type	liking interments	6	generate\include\IntCtrl_Ip_CfgDefines.h
			DMA Channel Lised	Using interrupts	6	generate\include\modules.h
check or update configurations	S (Enable EnhancedCBT Can		6	generate\include\OsIf_Cfg.h
for PIN Clock and Parinharal			Enable BitRate Switch		6	generate\include\Siul2_Port_Ip_Defines.h
TOI FIN, CIUCK, and Fengherals	5.		Can FD ISO		6	generate\src\FlexCAN_Ip_SD_InitPeripherals_PBcfg.c
	11		Auto Bus Off	Contrard time have in CAN bit deals	8	generate\src\IntCtrl_Ip_Cfg.c
			TimeStamp Timer Source	Message huffer time stamp base is TIMER	6	generate\src\Oslf_Cfg.c
			TimeStamp HR Capture Config	The high resolution time stamp capture is disabled. Enable Time V	stamp capture is disabled. Enable Time Y	
			FlexCAN Protocol Clock	4800000		BOARD_InitPeripherals
			 FlexCAN bitrate 		^	Other tools
Ston2.			Synchronization segment 1		A Pro	oblems 🛛 🛛 🖓 🖪 Y
Stepz.			Propagation segment 7		type	e filter text
Click "Update Code" to Genera	ate 🗲		Phase segment 1 6		Leve	el Resource Issue
			Phase segment 2 2 Prescaler division factor 6			
code for the configuration.			Resync jump width 2			
6			Bitrate (Kbps) 500			
			Sampling point (%) 87.5			
		100	FlexCAN CBT bitrate		~ <	>
	1003M of 1220M FlexCAN_example_CT					

BUILD AND DEBUG THE EXAMPLE PROJECT

Build the project



- > 🤒 board
- > 🐸 generate
- generate/include
- > 🤒 generate/src
- > 🐸 include
- Y 🤒 src
 - > I main.c
- > Debug_FLASH
 - ☑ description.txt
 - FlexCAN_example_CT.mex

Configure Debugging



Download and Debug the example project



REAL-TIME DRIVERS (RTD) for S32K3 CREATING NEW PROJECTS



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. © 2021 NXP B.V.

CREATE NEW PROJECT

Select MCU S32K344

Set project name.

\$32 D5	workspaceS32	DS.3.4 - S	32 Design S	itudio fo	r S32 Platf	orm		
File	Edit Source	Refactor	Navigate	Search	Project (Conf	igTools Run Window Help	
	New			Alt+	Shift+N >	¢	S32DS Project from Example	Ctrl+Alt+E
	Open File					=	S32DS Library Project	Ctrl+Alt+L
	Open Projects	from File	System			C	S32DS Application Project	Ctrl+Alt+A
	Recent Files				>	2	Makefile Project with Existing Code	
	Close				Ctrl+W	2	C/C++ Project	
	Close All			Ctrl+	Shift+W		Project	
	Save				Ctrl+S		Convert to a C/C++ Project (Adds C/C++ Nature)	
	Save As				Carro	63	Source Folder	
	Save All			Ctrl+	+Shift+S	R	Source File	
	Revert					h	Header File	
	Move					Ľ	File from Template	
M	Rename				F2	ଙ	Class	
8	Refresh				F5		Example	
	Convert Line [Delimiters	То		>		Other	Ctrl+N

832DS Application Project						×
Create a S32 Design Studio Pr	oject					
New S32DS Application Project						
Project name:						
S32K344_EVB_RTD_TestPrj						
✓ Use default location						
Location: C:\Users\nxa07657\work	spaceS32DS.3.4				Brows	e
Processors:	ToolChain S	election:				
type filter text	Core Kind	Name	Toolchain			
> 🗁 Family S32K1xx	M7	Cortex-M7	NXP GCC 9.2 for	Arm 32-bit Bare-Meta	- V	
 Family S32K3xx S32K314 						
S32K324						
S32K344	Description:					
	GNU 9.2 To	olchain is sel	ected			~
		Help text				
						\sim



SELECT SDK FOR S32K3XX NEW PROJECT

Select the desired RTD version ("RTD_Dxxxx_xxx") for the new project.

oject Name	S32K	344 EVB R1	D TestPri						
ore	VC	ortex-M7	_ ,						
brary	NewLib								
) Support	No I/O								
U Support	Tool	chain Defau	lt						
nguage	C								
)Ks	Platfo	ormSDK S3	2K3 2021 03 S3	2K344 M7	v 1.0.0				
ebugger	GDB	PEMicro De	bugging Interfa	e					
Select SDK									
Vame	Version	Device(s)	Device Core(s)	Туре	Description	Edit/Show in			
Iame PlatformSDK_S32K3_2021_03_S32K344_	Version M7 1.0.0	Device(s) S32K344	Device Core(s) S32K344_M7	Type External	Description PlatformSDK_S32K3_2021_03	Edit/Show in			
Vame PlatformSDK_S32K3_2021_03_S32K344_ 	Version M7 1.0.0 0.8.1	Device(s) S32K344 S32K344	Device Core(s) S32K344_M7 S32K344_M7	Type External External	Description PlatformSDK_S32K3_2021_03 S32K3XX AUTOSAR 4.4 RTD 0.8.1 D2011	Edit/Show in Clone from			
Vame ✓ PlatformSDK_S32K3_2021_03_S32K344_ _ RTD_D2011_S32K344_M7	Version M7 1.0.0 0.8.1	Device(s) S32K344 S32K344	Device Core(s) S32K344_M7 S32K344_M7	Type External External	Description PlatformSDK_S32K3_2021_03 S32K3XX AUTOSAR 4.4 RTD 0.8.1 D2011	Edit/Show in Clone from Reload			
Vame PlatformSDK_S32K3_2021_03_S32K344_ RTD_D2011_S32K344_M7	Version M7 1.0.0 0.8.1	Device(s) S32K344 S32K344	Device Core(s) S32K344_M7 S32K344_M7	Type External External	Description PlatformSDK_S32K3_2021_03 S32K3XX AUTOSAR 4.4 RTD 0.8.1 D2011	Edit/Show in Clone from Reload			



OPEN SDK CONFIG TOOL

Double click the ".mex" file to open SDK config tool

WorkspaceS32DS.3.4 - S32K344_EVB_RTD_TestPrj/src/main.c - S32 Design Studio for S32 Platform

V 🖷 🏔 🛢 Update Code 💌 🔳 Functional Group BOARD InitPins

File Edit Source Refactor Navigate Search Project ConfigTools Pins Run Window Help

- S32K344 EV8 RTD TestPrj



900	W = 00	- 0- 5 Q	2 P type filter text											Configuration - General Info
Pin	Pin name L	abel Id	lentifier SIUL2	eMIOS	FlexIO	LPSPI	LPUART	ADC	FLEXCAN	WKPU	L	cu	^	Configuration - HW Info
At	PTG10		SIUL2,gpi											Processor: S32K344
B1	PTA18		SIUL2,gpi_	eMIOS_1,c		LPSPI_1,lp.	LPUART_1.	_ ADC_2.p_i						Part number: S32K344_257BGA
C1	PTA19		SIUL2,gpi_	eMIOS_1,c		LPSPI_1,lp.	LPUART_1	ADC_2.p_j						Core: Cortex-M7
D1	PTA21		SIUL2,gpi_	eMIO5_1,c	FlexIO,flexi	. LPSPI_2,lp.	*	ADC_2.s_i						SDK Version: PlatformSDK_S32K3_2021_03
E1	PTE11		SIUL2.gpi_	eMIOS_0,c	FlexIO,flexi	. LPSPI_2.lp.	LPUART_4	_ ADC_0,p_i		WKPU,	vk			· Project
F1	PTE13		SIUL2.gpi_	eMIOS_1,c	FlexIO, flexi	. LPSPI_2,lp.	**	ADC_1,s_i_						a Dire
G1	PTA24		SIUL2,gpi,	eMIOS_1,c	FlexIO, flexi									- • • • • • • • • • • • • • • • • • • •
H1	PTA25		SIUL2,gpi,	eMIOS_1,c	FlexIO, flexi.	4				WKPU,v	vk		0	and run-time pin configuration.
M1	PTE12		SIUL2.gpi_	eMIOS_1,c	FlexIO,flexi		LPUART_2		FlexCAN_5.					
N1	PTE3		SIUL2.gpl_	eMIOS_0,c	FlexIO,flexi		LPUART_2	-	FlexCAN_4.				1	_ * _ •
P1	PTA31		SIUL2.gpi_	eMIOS_1,c_	FlexIO,flexi	. LPSPI_0,lp.	-							- * Generated code
R1	PTB18		SIUL2,gpi_	eMIOS_1,c	FlexIO, flexi	. LPSPI_1,lp.	LPUART_1	-						_ * Update code enabled
T1	PTB20		SIUL2,gpi_	eMIOS_1,c	FlexIO, flexi		LPUART_1							- / B board/Siul2_Port_Ip_Cfg.c
U1	PT821		SIUL2.gpi_	eMIOS_1,c	FlexIO,flexi		LPUART_1			WKPU,	vk			- 1 B handfield fair to Clark
A2	PTAB		SIUL2.gpi_	eMIOS_1,c	FlexIO,flexi	. LPSPI_2,lp.	LPUART_2	ADC_0,p_i		WKPU,v	vk		1	= boardSol2_ForCip_Cigit
C2	PTE16		SIUL2,gpi	eMIOS_0,c	FlexIO,flexi	. LPSPI_2,lp.	LPUART_3,	ADC_0,p_i		WKPU,v	vk			 Functional groups
D2	PTE15		SIUL2,gpi	eMIOS_0,c	FlexIO,flexi	. LPSPI_2,lp.	LPUART_3	ADC_0,p_i					M	BOARD_InitPins
E2	PTE10		SIUL2,gpi	eMIOS_0,c	FlexIO, flexi.	. LPSPI_3.lp.	LPUART_4	ADC_0,p_i			_		н	- 4
F2	PTE5		SIUL2.gpi	eMIOS_1,c	FlexIO,flexi		LPUART_1	ADC_2.s_i		WKPU,	vk			_ , Other tools
G2	PTG9		SIUL2.gpi_						FlexCAN_4.					
H2	PTG1		SIUL2,gpi_	eMIOS_1,c			LPUART_1	2					Ŧ	_ · ('UI) (Y)
12	PTG0		SIUL2.gpi_	eMIOS_1,c			LPUART_1						U.	
K2	PTG3		SIUL2,gpi_	eMIOS_1,c_			LPUART_1				_			
12	PTA26		SIUL2.gpi_	eMIOS_1,c	FlexIO,flexi.	. LPSPI_1,lp.				WKPU,v	vk			
M2	PTA27		SIUL2.gpi	eMIOS_1,c	FlexIO,flexi	-	LPUART_0	_	FlexCAN_0.					
N2	PTA28		SIUL2,gpi	eMIOS_1,c		LPSPI_1.lp.	LPUART_0	-	FlexCAN_0.					
P2	PTD16		SIUL2,gpi_	eMIO5_0,c		LPSPI_0,lp.	LPUART_2	-		-				
0.5	010140		C8.8.3 am	-14/00 1 -	rianto end		IDECADT 4			LANCTON C.	240	>		
Routed	d Pins													· • •
p type	filter text													
Routed	Pins for BO	ARD. 0	0089											
	Periph_	Signal	Route to Label	Identifi Dir	ecti Slew R	Outpu	Safe M In	out Pull Se	Pullup 1	nput I P	ad ke	Drive	Initial	
														A Problems of Table 1
														type filter text
														Level Resource Issue

Package [Pins Bottom] #

Q Q C 📷 🖬 🧔 🗁 🗇 👘 Overview 🗉 🗟 Code Preview 🗊 Registers

- 0 ×

요 (명) 다 = 10 ? 사 프 # 4

S32DS CONFIGURATION TOOL

Click the button on top right to switch between different configuration tools and source code editor.



PUBLIC 44

REAL-TIME DRIVERS (RTD)

Learn more at <u>nxp.com/RTD</u>

Join us at <u>NXP Connects</u>! Registration now open.

NXP Connects EMEA: Nov 9-10, 2021

NXP Connects AMEC: Nov 10-11, 2021

NXP Connects APAC: Nov 16-17, 2021

SECURE CONNECTIONS FOR A SMARTER WORLD

SHOWROOM.NXP.COM

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. © 2021 NXP B.V.