S32G-VNP-GLDBOX3

SOFTWARE ENABLEMENT GUIDE

MARCH 2023

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

CONTENTS SOFTWARE ENABLEMENT GUIDE

- Get Software and Tools for S32G3
- Install Software and Tools for S32G3
- Light Up RGB LED using S32 Real Time Drivers
- Enable Linux BSP On Cortex-A53 Core

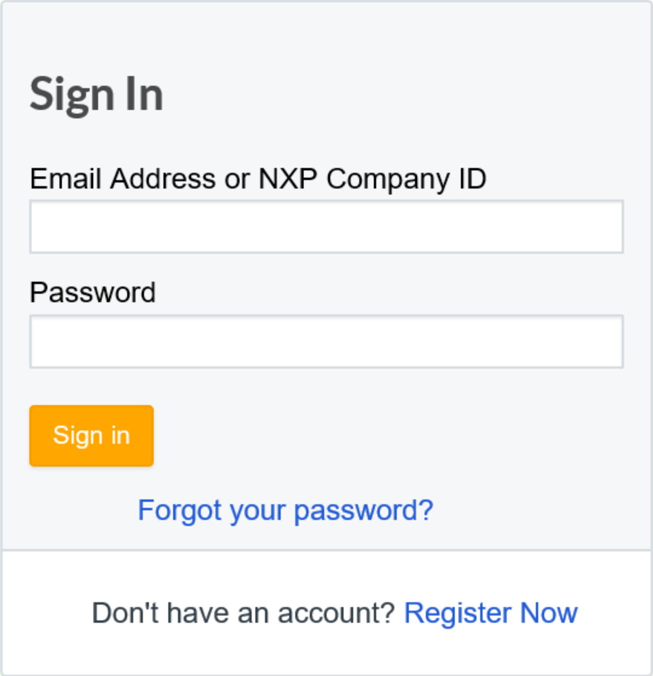
Get Software and Tools for S32G3

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

GET SOFTWARE AND TOOLS FOR S32G3

- Please go to: S32G Processors for Vehicle Networking and use your NXP account to sign in.



Sign In

Email Address or NXP Company ID

Password

Sign in

Forgot your password?

Don't have an account? Register Now

- Please firstly download the below software for enablement

Production	Install Packages
S32 Design Studio for S32 Platform	SW32G_S32DS_3.5.1_D2210.zip
	S32DS.3.5_b220726_win32.x86_64.exe
S32G3 - Real Time Drivers	SW32_RTD_4.4_4.0.0_DS_updatesite_D2210.zip

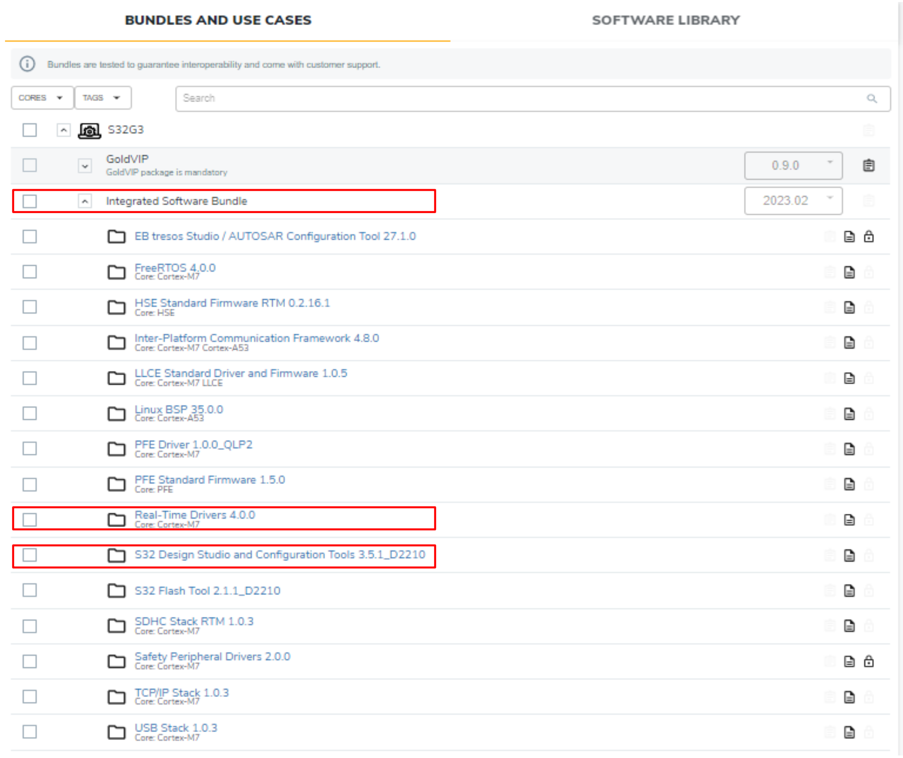
GET SOFTWARE AND TOOLS FOR S32G3

- Download S32G3 - Real Time Drivers and S32DS update package:

  a. Please go to: https://www.nxp.com/app-autopackagemgr/software-package-manager:AUTO-SW-PACKAGE-MANAGER and choose S32G3 integrated Software Bundle(2023.02).

  b. Choose Real-Time-Drivers 4.0.0 item and S32 Design Studio and Configuration Tools 3.5.1_D2210, download NXP_Multi_Install_XXX.exe.

  c. Double click on NXP_Multi_Install_XXX.exe on your local PC to download the packages . You will find SW32_RTD_4.4_4.0.0_DS_updatesite_D2210.zip and SW32G_S32DS_3.5.1_D2210.zip in NXP_Multi_Installer_XXX folder.



Bundles are tested to guarantee interoperability and come with customer support.

□ S32G3

□ GoldVIP 0.9.0
GoldVIP package is mandatory

□ Integrated Software Bundle 2023.02

□ EB tresos Studio / AUTOSAR Configuration Tool 27.1.0

□ FreeRTOS 4.0.0
Core: Cortex-M7

□ HSE Standard Firmware RTM 0.2.16.1
Core: HSE

□ Inter-Platform Communication Framework 4.8.0
Core: Cortex-M7 Cortex-A53

□ LLCE Standard Driver and Firmware 1.0.5
Core: Cortex-M7 LLCE

□ Linux BSP 35.0.0
Core: Cortex-A53

□ PFE Driver 1.0.0_QLP2
Core: Cortex-M7

□ PFE Standard Firmware 1.5.0
Core: PFE

□ Real-Time Drivers 4.0.0
Core: Cortex-M7

□ S32 Design Studio and Configuration Tools 3.5.1_D2210

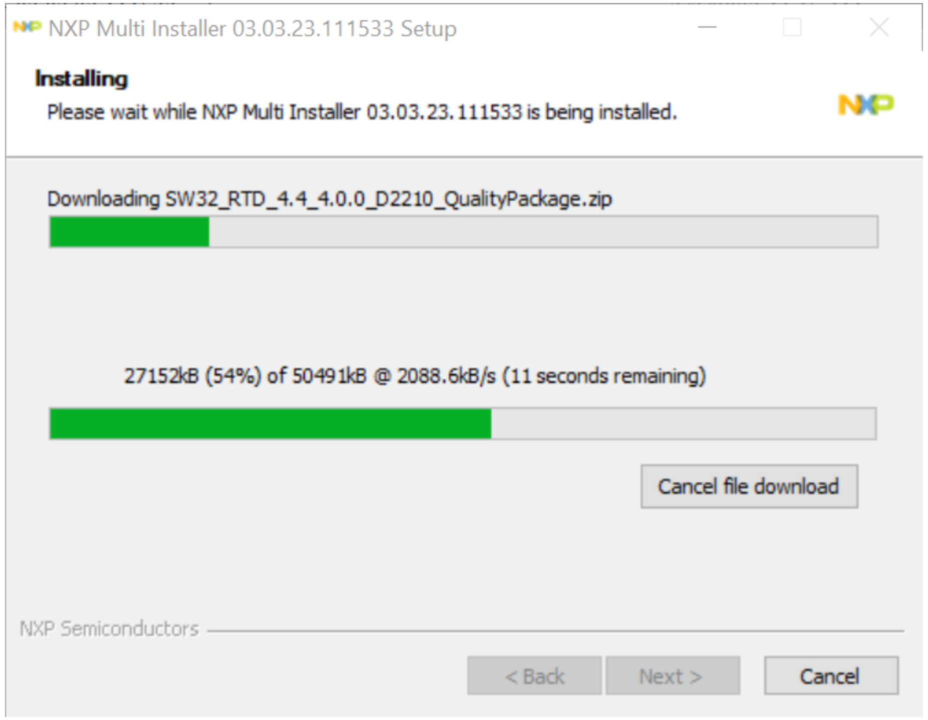
□ S32 Flash Tool 2.1.1_D2210

□ SDHC Stack RTM 1.0.3
Core: Cortex-M7

□ Safety Peripheral Drivers 2.0.0
Core: Cortex-M7

□ TCP/IP Stack 1.0.3
Core: Cortex-M7

□ USB Stack 1.0.3
Core: Cortex-M7


NXP Multi Installer 03.03.23.111533 Setup

Installing

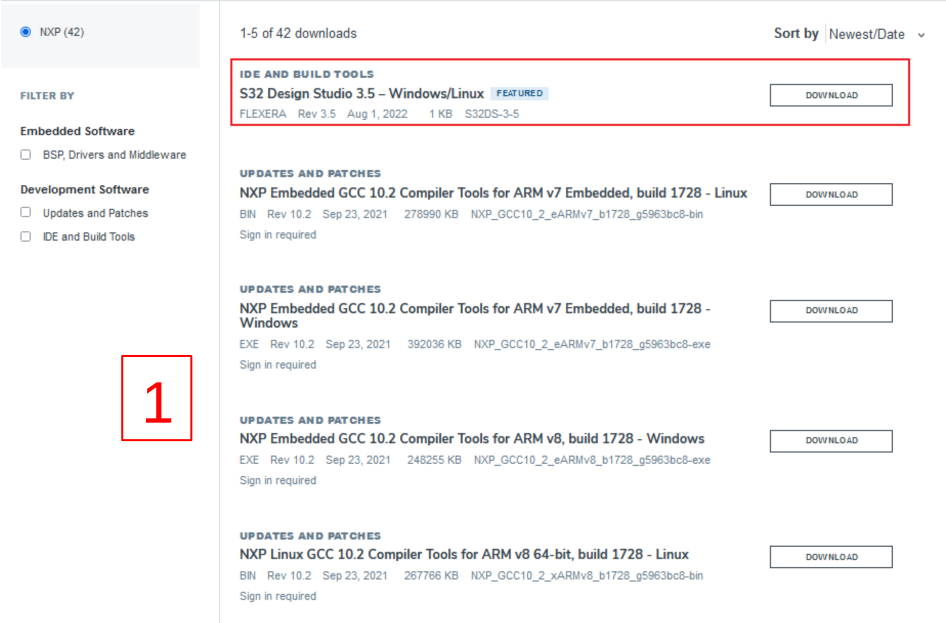
Please wait while NXP Multi Installer 03.03.23.111533 is being installed.

Downloading SW32_RTD_4.4_4.0.0_D2210_QualityPackage.zip

27152kB (54%) of 50491kB @ 2088.6kB/s (11 seconds remaining)

GET SOFTWARE AND TOOLS FOR S32G3

- Download S32DS.3.5_b220726_win32.x86_64.exe from https://www.nxp.com/design/software/development-software/s32-design-studio-ide/s32-design-studio-for-s32-platform:S32DS-S32PLATFORM



● NXP (42)

FILTER BY

Embedded Software

□ BSP, Drivers and Middleware

Development Software

□ Updates and Patches

□ IDE and Build Tools

1-5 of 42 downloads   Sort by Newest/Date

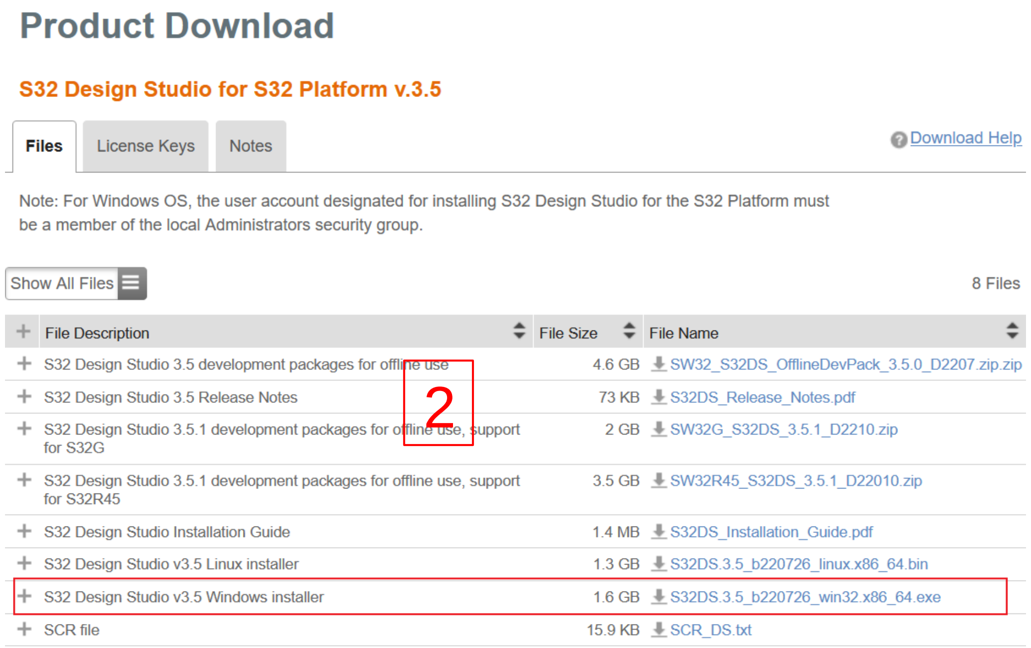
IDE AND BUILD TOOLS
S32 Design Studio 3.5 – Windows/Linux FEATURED
FLEXERA Rev 3.5 Aug 1, 2022 1 KB S32DS-3-5
DOWNLOAD

UPDATES AND PATCHES
NXP Embedded GCC 10.2 Compiler Tools for ARM v7 Embedded, build 1728 - Linux
BIN Rev 10.2 Sep 23, 2021 278990 KB NXP_GCC10_2_eARMv7_b1728_g5963bc8-bin
Sign in required
DOWNLOAD

UPDATES AND PATCHES
NXP Embedded GCC 10.2 Compiler Tools for ARM v7 Embedded, build 1728 - Windows
EXE Rev 10.2 Sep 23, 2021 392036 KB NXP_GCC10_2_eARMv7_b1728_g5963bc8-exe
Sign in required
DOWNLOAD

UPDATES AND PATCHES
NXP Embedded GCC 10.2 Compiler Tools for ARM v8, build 1728 - Windows
EXE Rev 10.2 Sep 23, 2021 248255 KB NXP_GCC10_2_eARMv8_b1728_g5963bc8-exe
Sign in required
DOWNLOAD

UPDATES AND PATCHES
NXP Linux GCC 10.2 Compiler Tools for ARM v8 64-bit, build 1728 - Linux
BIN Rev 10.2 Sep 23, 2021 267766 KB NXP_GCC10_2_xARMv8_b1728_g5963bc8-bin
Sign in required
DOWNLOAD



Product Download

S32 Design Studio for S32 Platform v.3.5

Files | License Keys | Notes   Download Help

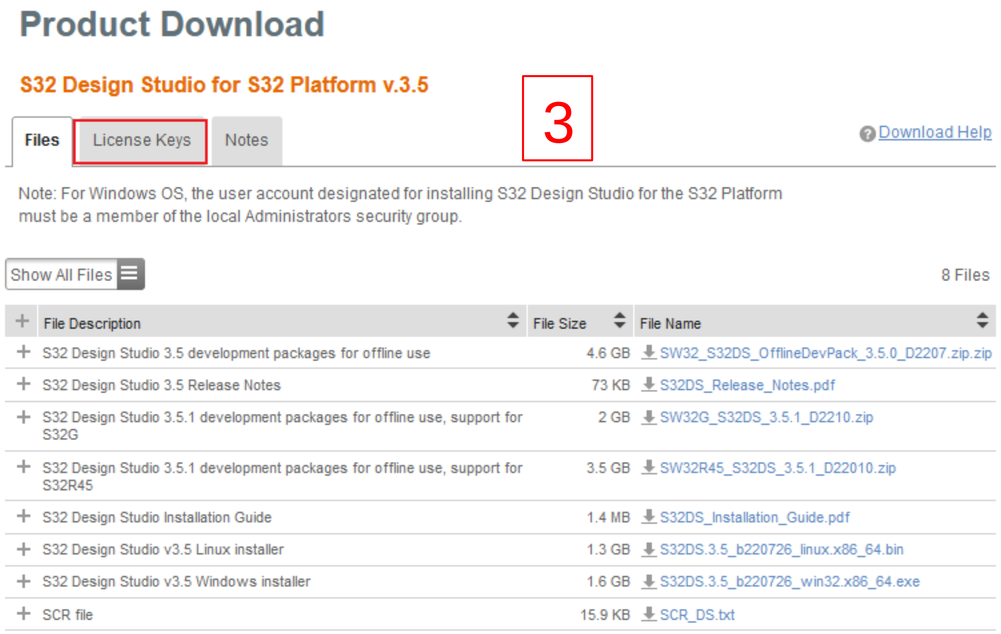
Note: For Windows OS, the user account designated for installing S32 Design Studio for the S32 Platform must be a member of the local Administrators security group.

Show All Files   8 Files

File Description	File Size	File Name
S32 Design Studio 3.5 development packages for offline use	4.6 GB	SW32_S32DS_OfflineDevPack_3.5.0_D2207.zip.zip
S32 Design Studio 3.5 Release Notes	73 KB	S32DS_Release_Notes.pdf
S32 Design Studio 3.5.1 development packages for offline use, support for S32G	2 GB	SW32G_S32DS_3.5.1_D2210.zip
S32 Design Studio 3.5.1 development packages for offline use, support for S32R45	3.5 GB	SW32R45_S32DS_3.5.1_D22010.zip
S32 Design Studio Installation Guide	1.4 MB	S32DS_Installation_Guide.pdf
S32 Design Studio v3.5 Linux installer	1.3 GB	S32DS.3.5_b220726_linux.x86_64.bin
S32 Design Studio v3.5 Windows installer	1.6 GB	S32DS.3.5_b220726_win32.x86_64.exe
SCR file	15.9 KB	SCR_DS.txt

GET SOFTWARE AND TOOLS FOR S32G3

- Get S32DS activation Code:



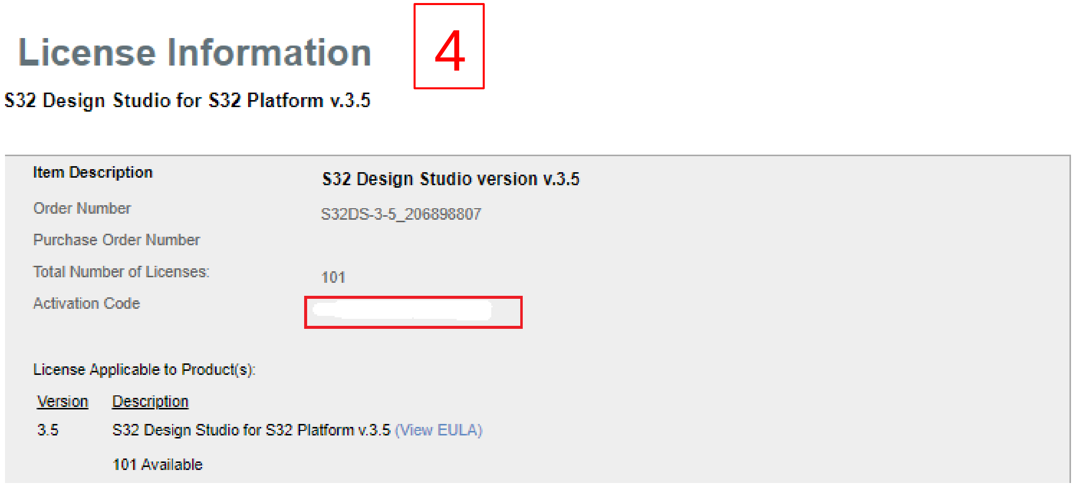
Product Download

S32 Design Studio for S32 Platform v.3.5

Note: For Windows OS, the user account designated for installing S32 Design Studio for the S32 Platform must be a member of the local Administrators security group.

8 Files

File Description	File Size	File Name
S32 Design Studio 3.5 development packages for offline use	4.6 GB	SW32_S32DS_OfflineDevPack_3.5.0_D2207.zip.zip
S32 Design Studio 3.5 Release Notes	73 KB	S32DS_Release_Notes.pdf
S32 Design Studio 3.5.1 development packages for offline use, support for S32G	2 GB	SW32G_S32DS_3.5.1_D2210.zip
S32 Design Studio 3.5.1 development packages for offline use, support for S32R45	3.5 GB	SW32R45_S32DS_3.5.1_D22010.zip
S32 Design Studio Installation Guide	1.4 MB	S32DS_Installation_Guide.pdf
S32 Design Studio v3.5 Linux installer	1.3 GB	S32DS.3.5_b220726_linux.x86_64.bin
S32 Design Studio v3.5 Windows installer	1.6 GB	S32DS.3.5_b220726_win32.x86_64.exe
SCR file	15.9 KB	SCR_DS.txt



License Information

S32 Design Studio for S32 Platform v.3.5

Item Description	**S32 Design Studio version v.3.5**
Order Number	S32DS-3-5_206898807
Purchase Order Number	
Total Number of Licenses:	101
Activation Code	

License Applicable to Product(s):

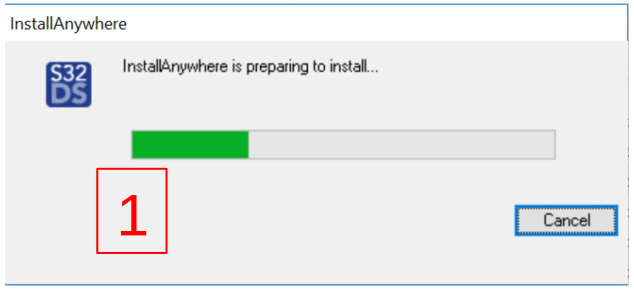
Version	Description
3.5	S32 Design Studio for S32 Platform v.3.5 (View EULA) 101 Available

Install Software and Tools for S32G3

SECURE CONNECTIONS
FOR A SMARTER WORLD

STEP 1: INSTALL S32 DESIGN STUDIO 3.5

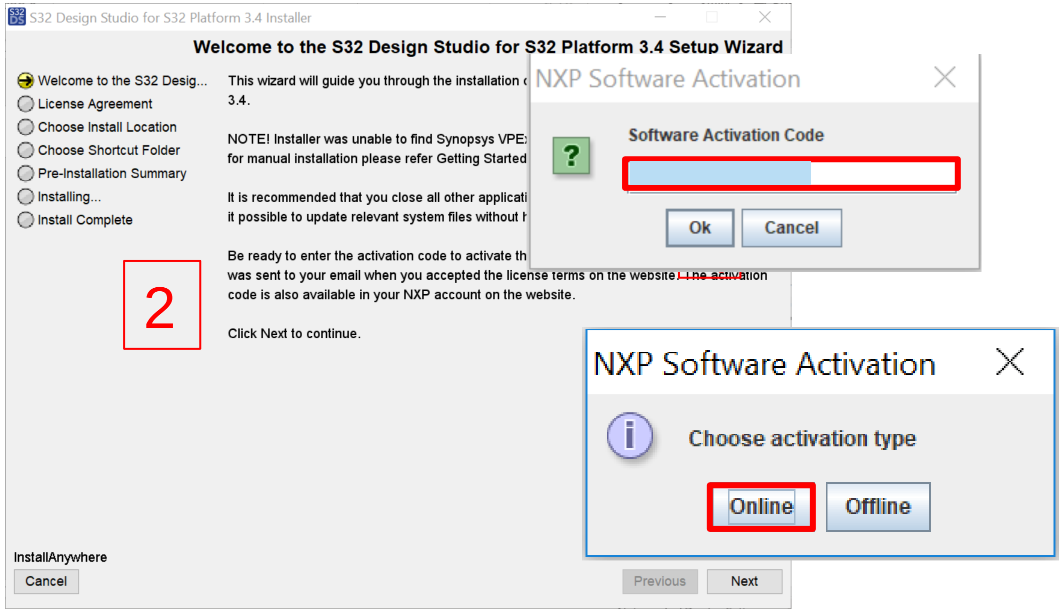
- Double click on the downloaded S32DS.3.5_b220726_win32.x86_64.exe file to start installation.



InstallAnywhere

InstallAnywhere is preparing to install...

- Click "Next" to install the software step by step. Input the Activation code got from page 5 and choose "Online".



S32 Design Studio for S32 Platform 3.4 Installer

Welcome to the S32 Design Studio for S32 Platform 3.4 Setup Wizard

- Welcome to the S32 Desig...
- License Agreement
- Choose Install Location
- Choose Shortcut Folder
- Pre-Installation Summary
- Installing...
- Install Complete

This wizard will guide you through the installation c
3.4.

NOTE! Installer was unable to find Synopsys VPE:
for manual installation please refer Getting Started

It is recommended that you close all other applicati
it possible to update relevant system files without r

Be ready to enter the activation code to activate th
was sent to your email when you accepted the license terms on the website. The activation
code is also available in your NXP account on the website.

Click Next to continue.

NXP Software Activation

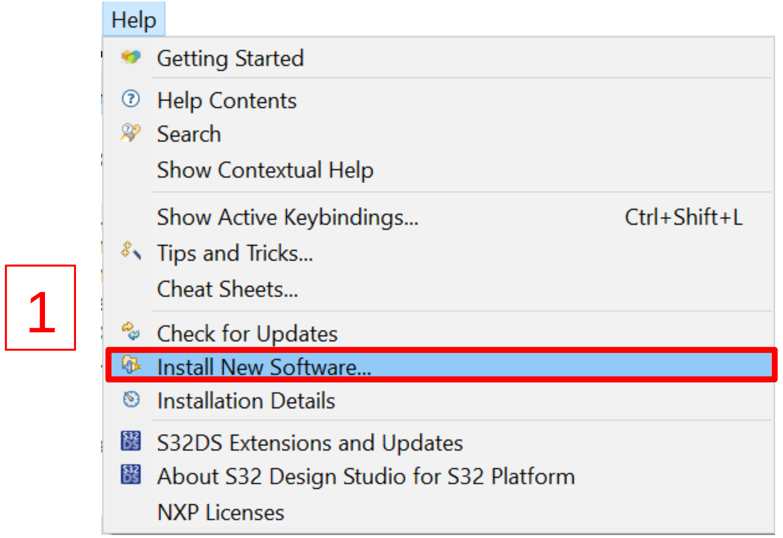
Software Activation Code

NXP Software Activation

Choose activation type

STEP 2: INSTALL DEVELOPMENT PACKAGES FOR S32G

- Open S32DS 3.5 and select a directory as workspace and select “Install New Software …” option on Help menu.



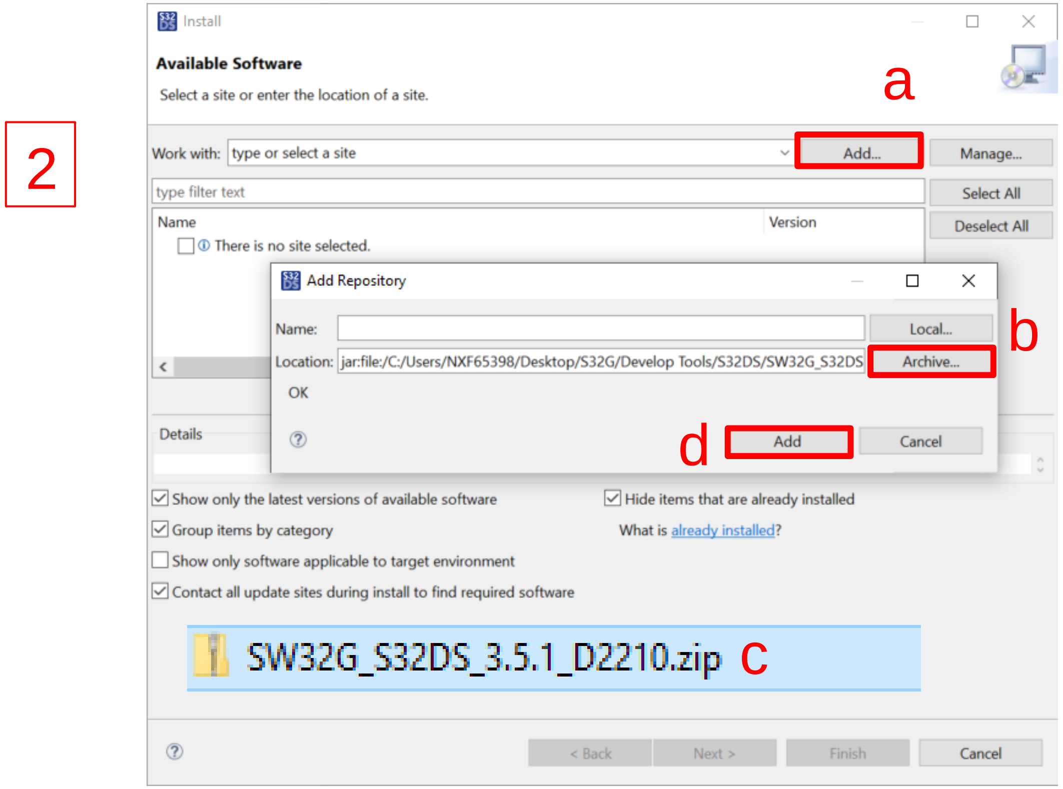
- Add update package of S32DS.

a. Click on “Add” button ”

b. Click on “Archive” button in Add Repository dialog

c. Select the file SW32G_S32DS_3.5.1_D2210.zip and click on “open”

d. Click on “Add” button in Add Repository dialog



Available Software

Select a site or enter the location of a site.

Work with: type or select a site

type filter text

Name Version

□ There is no site selected.

Details

Add Repository

Name:

Location: jar:file:/C:/Users/NXF65398/Desktop/S32G/Develop Tools/S32DS/SW32G_S32DS

OK

☑ Show only the latest versions of available software ☑ Hide items that are already installed

☑ Group items by category What is already installed?

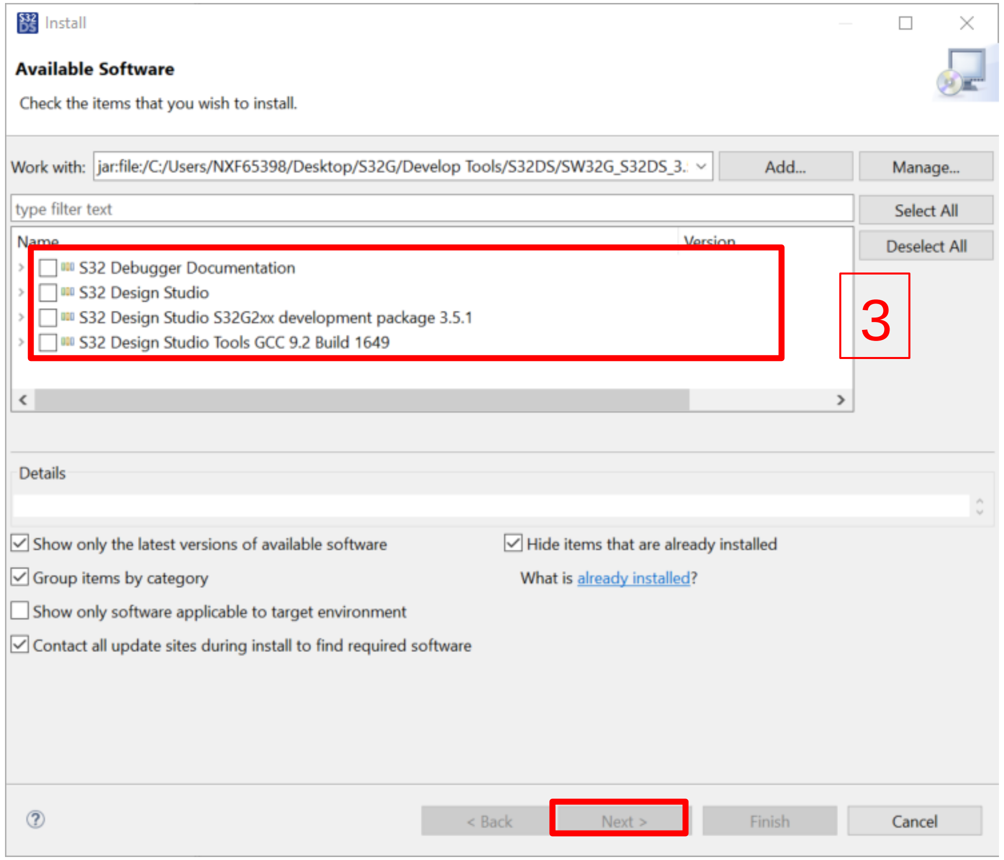
□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

SW32G_S32DS_3.5.1_D2210.zip

STEP 2: INSTALL DEVELOPMENT PACKAGES FOR S32G

- Select the four items as below and click on ”Next>” button.



Install

Available Software

Check the items that you wish to install.

Work with: jar:file:/C:/Users/NXF65398/Desktop/S32G/Develop Tools/S32DS/SW32G_S32DS_3.

type filter text

Name	Version
□ S32 Debugger Documentation	
□ S32 Design Studio	
□ S32 Design Studio S32G2xx development package 3.5.1	
□ S32 Design Studio Tools GCC 9.2 Build 1649	

3

Details

☑ Show only the latest versions of available software

☑ Hide items that are already installed

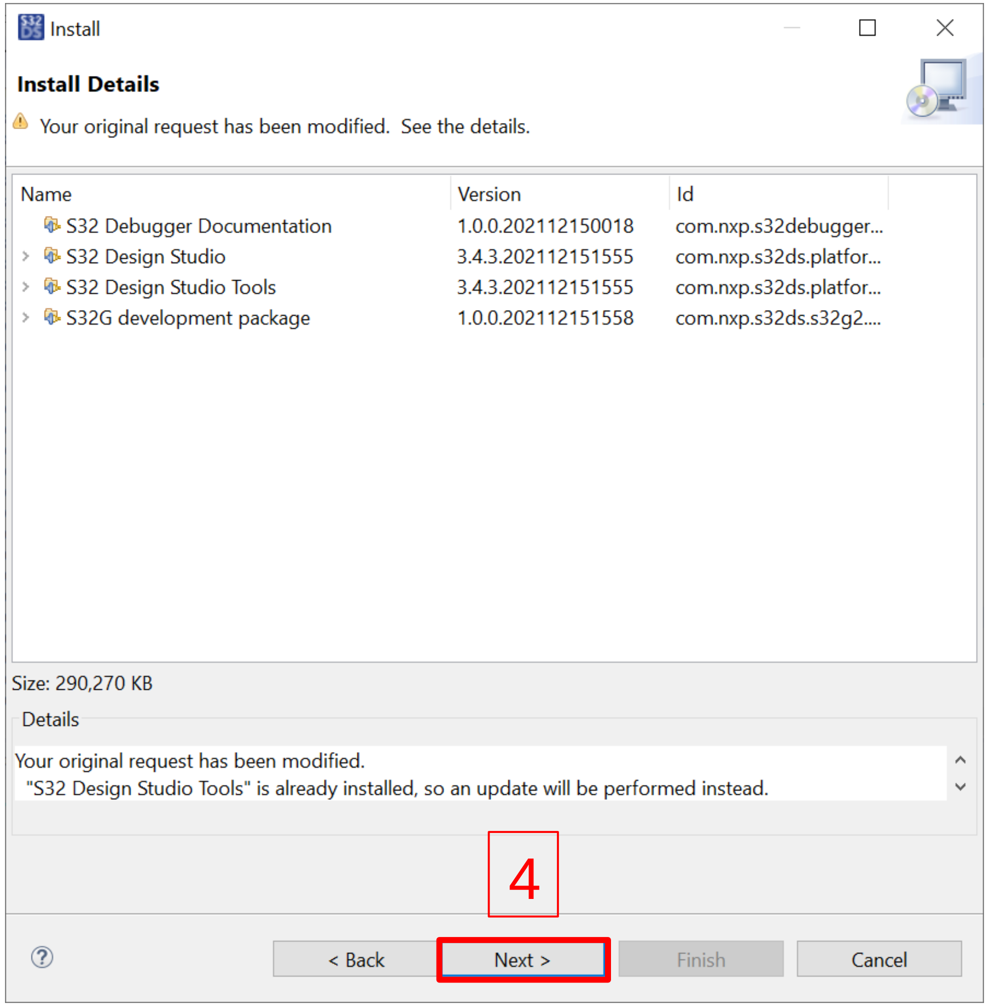
☑ Group items by category

What is already installed?

□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

- Click on ”Next>” button



Install

Install Details

Your original request has been modified. See the details.

Name	Version	Id
S32 Debugger Documentation	1.0.0.202112150018	com.nxp.s32debugger...
S32 Design Studio	3.4.3.202112151555	com.nxp.s32ds.platfor...
S32 Design Studio Tools	3.4.3.202112151555	com.nxp.s32ds.platfor...
S32G development package	1.0.0.202112151558	com.nxp.s32ds.s32g2....

Size: 290,270 KB

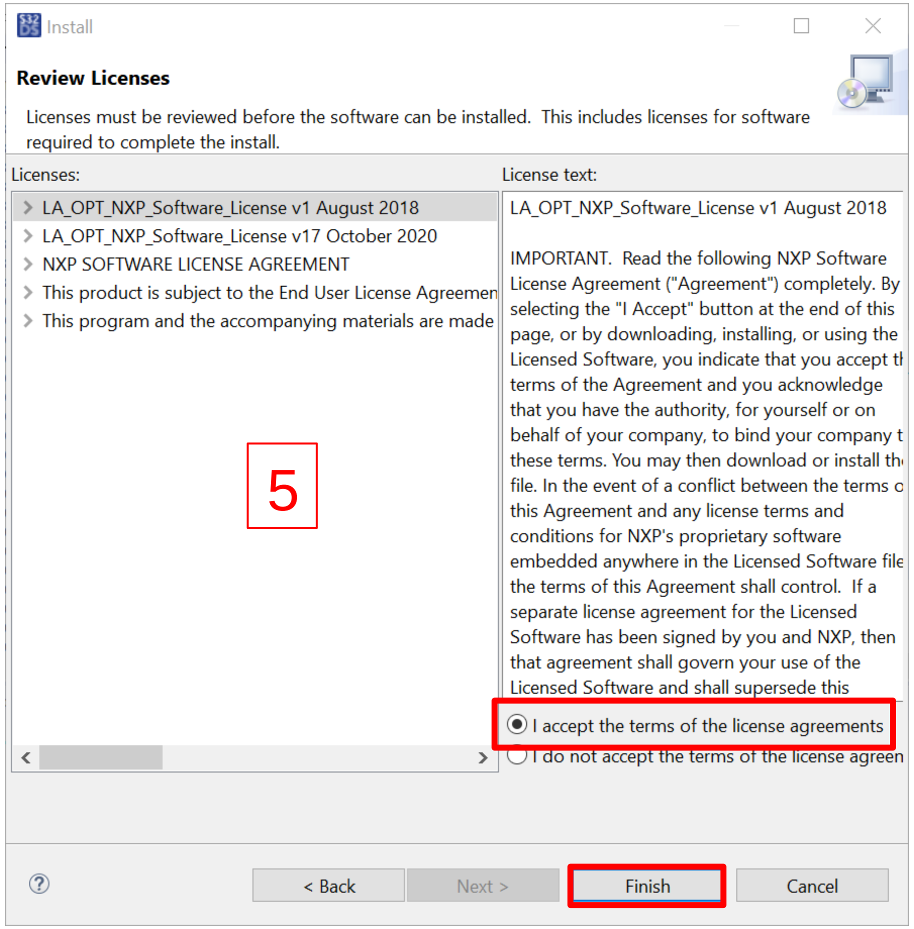
Details

Your original request has been modified.
 "S32 Design Studio Tools" is already installed, so an update will be performed instead.

4

STEP 2: INSTALL DEVELOPMENT PACKAGES FOR S32G

- Select the ”I accept the terms of the license agreements” button and click on ”Finish” button.



Licenses must be reviewed before the software can be installed. This includes licenses for software required to complete the install.

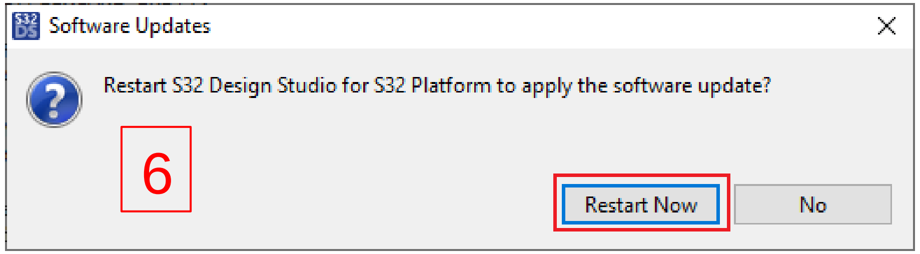
- LA_OPT_NXP_Software_License v1 August 2018
- LA_OPT_NXP_Software_License v17 October 2020
- NXP SOFTWARE LICENSE AGREEMENT
- This product is subject to the End User License Agreemen
- This program and the accompanying materials are made

LA_OPT_NXP_Software_License v1 August 2018

IMPORTANT. Read the following NXP Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, or by downloading, installing, or using the Licensed Software, you indicate that you accept tl terms of the Agreement and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company t these terms. You may then download or install th file. In the event of a conflict between the terms c this Agreement and any license terms and conditions for NXP's proprietary software embedded anywhere in the Licensed Software file the terms of this Agreement shall control. If a separate license agreement for the Licensed Software has been signed by you and NXP, then that agreement shall govern your use of the Licensed Software and shall supersede this

● I accept the terms of the license agreements
○ I do not accept the terms of the license agreen

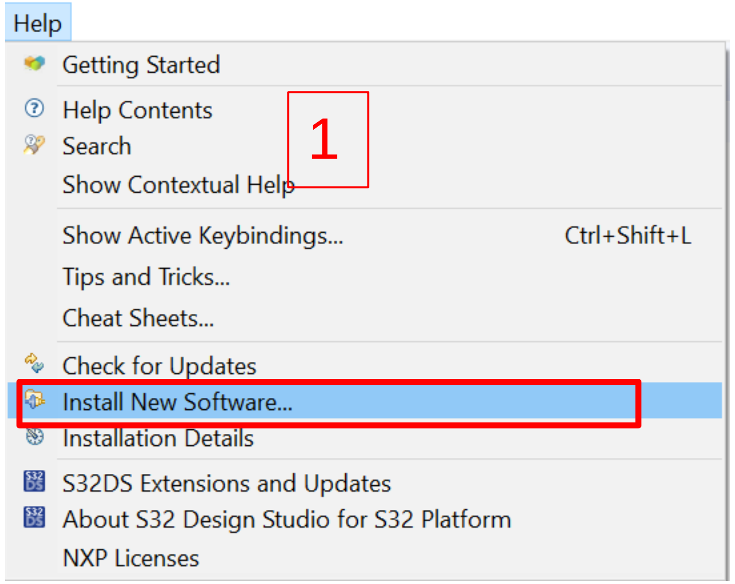
- Waiting for finishing the installation process and click on ” Restart Now ” button.



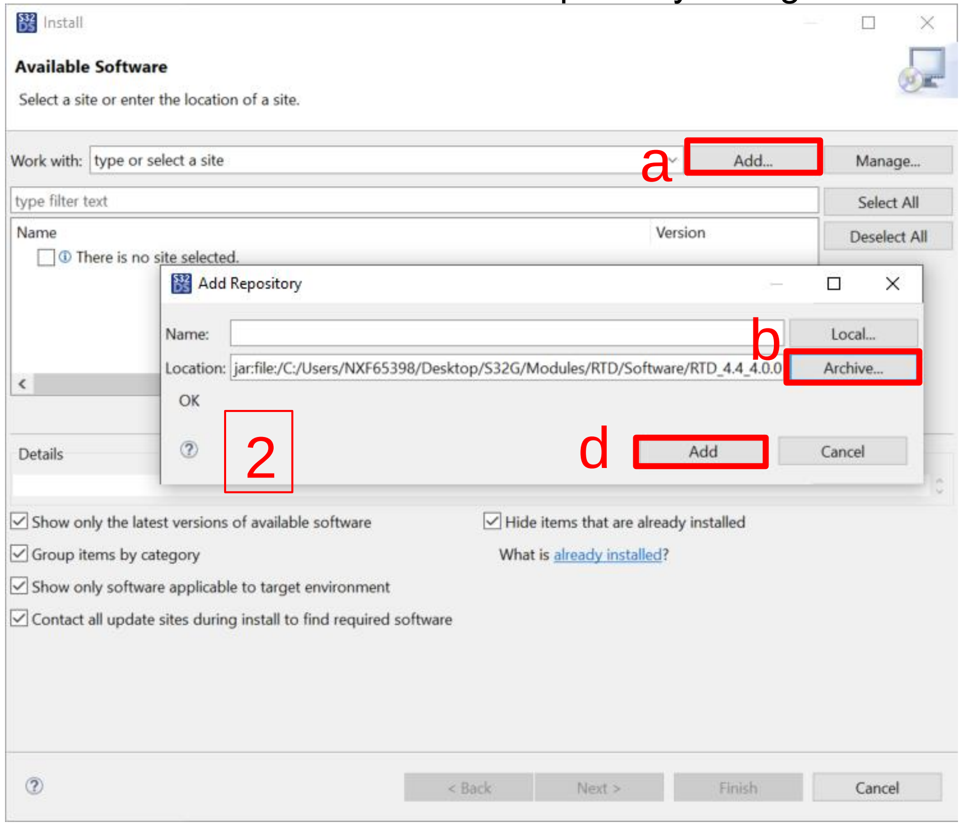
Restart S32 Design Studio for S32 Platform to apply the software update?

STEP 3: INSTALL S32G REAL-TIME DRIVERS

- Open S32DS 3.5 and select a directory as workspace and click “Install New Software …” option on help menu



- Add development package of S32DS
  a. Click on “Add…”
  b. Click on “Archive” button in Add Repository dialog
  c. Select SW32_RTD_4.4_4.0.0_DS_updatesite_D2210.zip file and click on “open”
  d. Click on “Add” button in Add Repository dialog.



Select a site or enter the location of a site.

Work with: type or select a site

type filter text

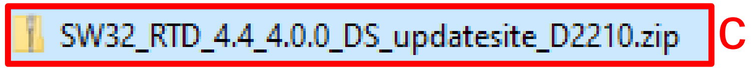
□ ⓘ There is no site selected.

Name:

Location: jar:file:/C:/Users/NXF65398/Desktop/S32G/Modules/RTD/Software/RTD_4.4_4.0.0

OK

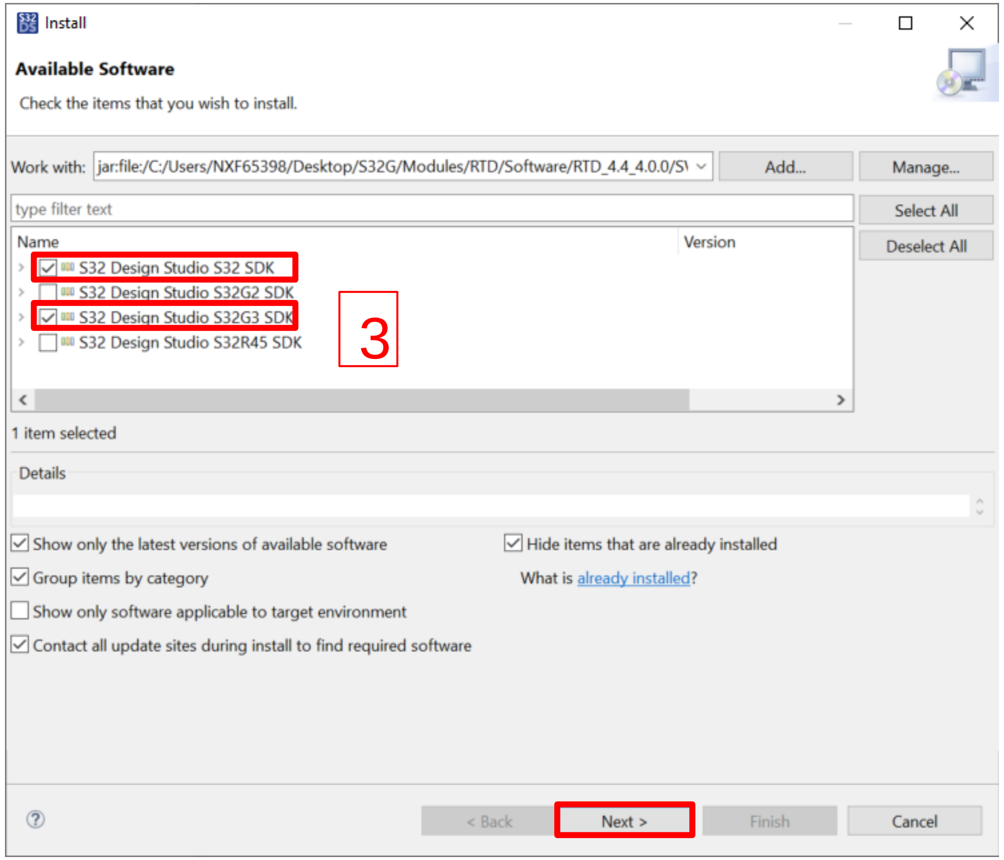
☑ Show only the latest versions of available software
☑ Hide items that are already installed
☑ Group items by category
What is already installed?
☑ Show only software applicable to target environment
☑ Contact all update sites during install to find required software



SW32_RTD_4.4_4.0.0_DS_updatesite_D2210.zip

STEP 3: INSTALL S32G REAL-TIME DRIVERS

- Check the two items as below and click on "Next" to install step by step



Install

Available Software

Check the items that you wish to install.

Work with: jar:file:/C:/Users/NXF65398/Desktop/S32G/Modules/RTD/Software/RTD_4.4_4.0.0/S\

type filter text

Name	Version
☑ S32 Design Studio S32 SDK	
□ S32 Design Studio S32G2 SDK	
☑ S32 Design Studio S32G3 SDK	
□ S32 Design Studio S32R45 SDK	

1 item selected

Details

☑ Show only the latest versions of available software

☑ Hide items that are already installed

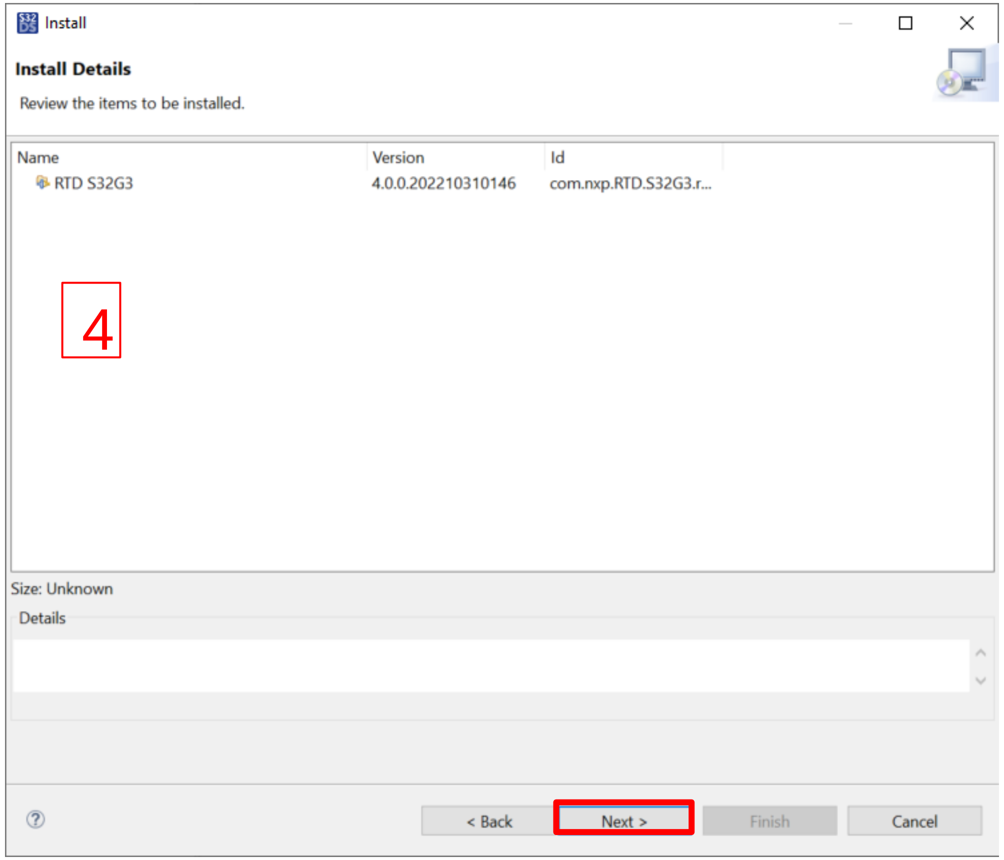
☑ Group items by category

What is already installed?

□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

- Click on "Next>" button



Install

Install Details

Review the items to be installed.

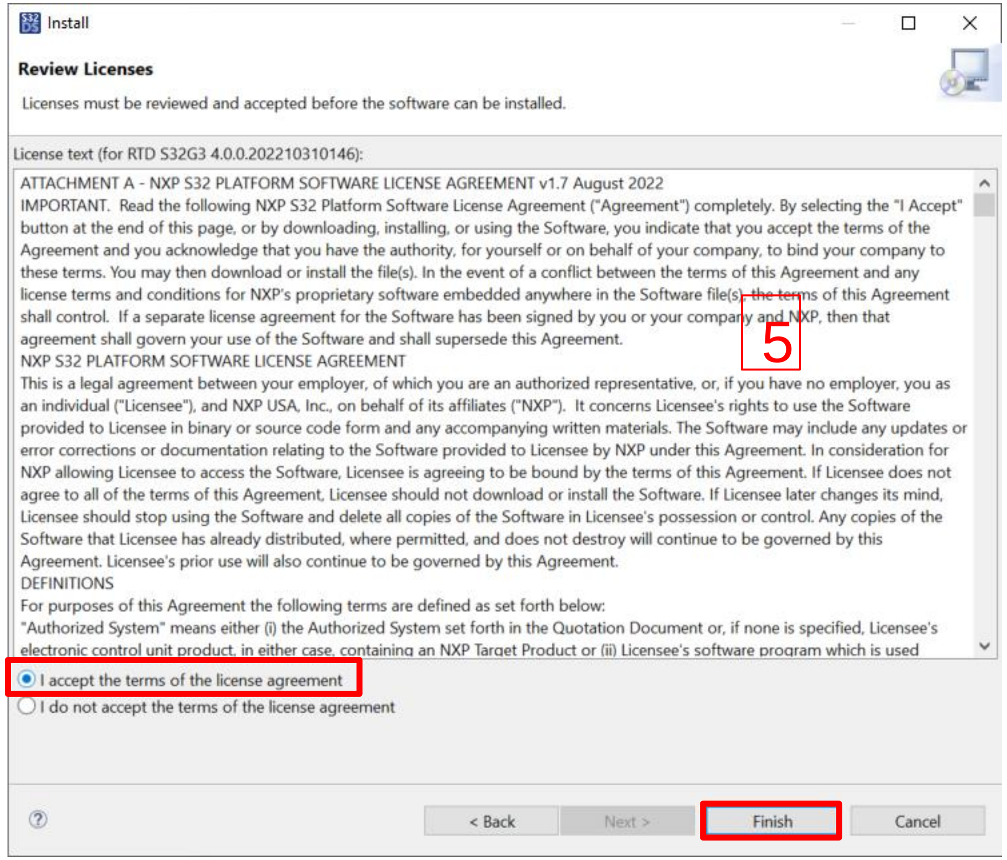
Name	Version	Id
RTD S32G3	4.0.0.202210310146	com.nxp.RTD.S32G3.r...

Size: Unknown

Details

STEP 3: INSTALL S32G REAL-TIME DRIVERS

- Choose ”I accept the terms of the license agreement.” option and click on ”Finish” button.
- Select ”NXP USA\,INC;null;NXP USA\,INC.” option and click on “Accept selected” button.



Install

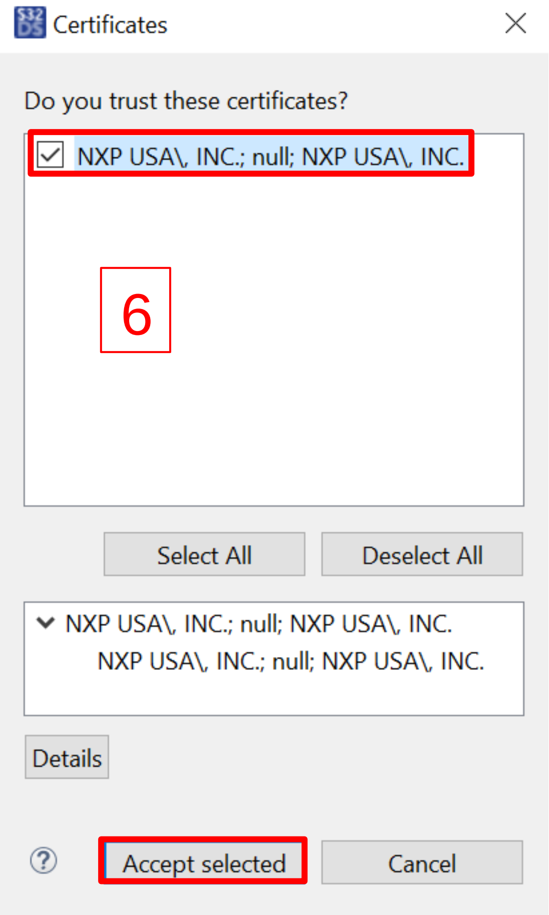
Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

License text (for RTD S32G3 4.0.0.202210310146):

ATTACHMENT A - NXP S32 PLATFORM SOFTWARE LICENSE AGREEMENT v1.7 August 2022
IMPORTANT. Read the following NXP S32 Platform Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, or by downloading, installing, or using the Software, you indicate that you accept the terms of the Agreement and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company to these terms. You may then download or install the file(s). In the event of a conflict between the terms of this Agreement and any license terms and conditions for NXP's proprietary software embedded anywhere in the Software file(s), the terms of this Agreement shall control. If a separate license agreement for the Software has been signed by you or your company and NXP, then that agreement shall govern your use of the Software and shall supersede this Agreement.
NXP S32 PLATFORM SOFTWARE LICENSE AGREEMENT
This is a legal agreement between your employer, of which you are an authorized representative, or, if you have no employer, you as an individual ("Licensee"), and NXP USA, Inc., on behalf of its affiliates ("NXP"). It concerns Licensee's rights to use the Software provided to Licensee in binary or source code form and any accompanying written materials. The Software may include any updates or error corrections or documentation relating to the Software provided to Licensee by NXP under this Agreement. In consideration for NXP allowing Licensee to access the Software, Licensee is agreeing to be bound by the terms of this Agreement. If Licensee does not agree to all of the terms of this Agreement, Licensee should not download or install the Software. If Licensee later changes its mind, Licensee should stop using the Software and delete all copies of the Software in Licensee's possession or control. Any copies of the Software that Licensee has already distributed, where permitted, and does not destroy will continue to be governed by this Agreement. Licensee's prior use will also continue to be governed by this Agreement.
DEFINITIONS
For purposes of this Agreement the following terms are defined as set forth below:
"Authorized System" means either (i) the Authorized System set forth in the Quotation Document or, if none is specified, Licensee's electronic control unit product, in either case, containing an NXP Target Product or (ii) Licensee's software program which is used

● I accept the terms of the license agreement
○ I do not accept the terms of the license agreement



Certificates

Do you trust these certificates?

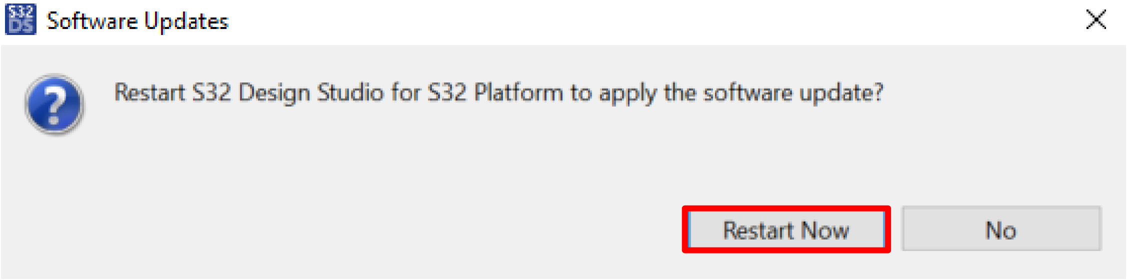
☑ NXP USA\, INC.; null; NXP USA\, INC.

NXP USA\, INC.; null; NXP USA\, INC.
NXP USA\, INC.; null; NXP USA\, INC.

STEP 3: INSTALL S32G REAL-TIME DRIVERS

- Waiting for finishing the installation process and click on ” Restart Now ” button.

7



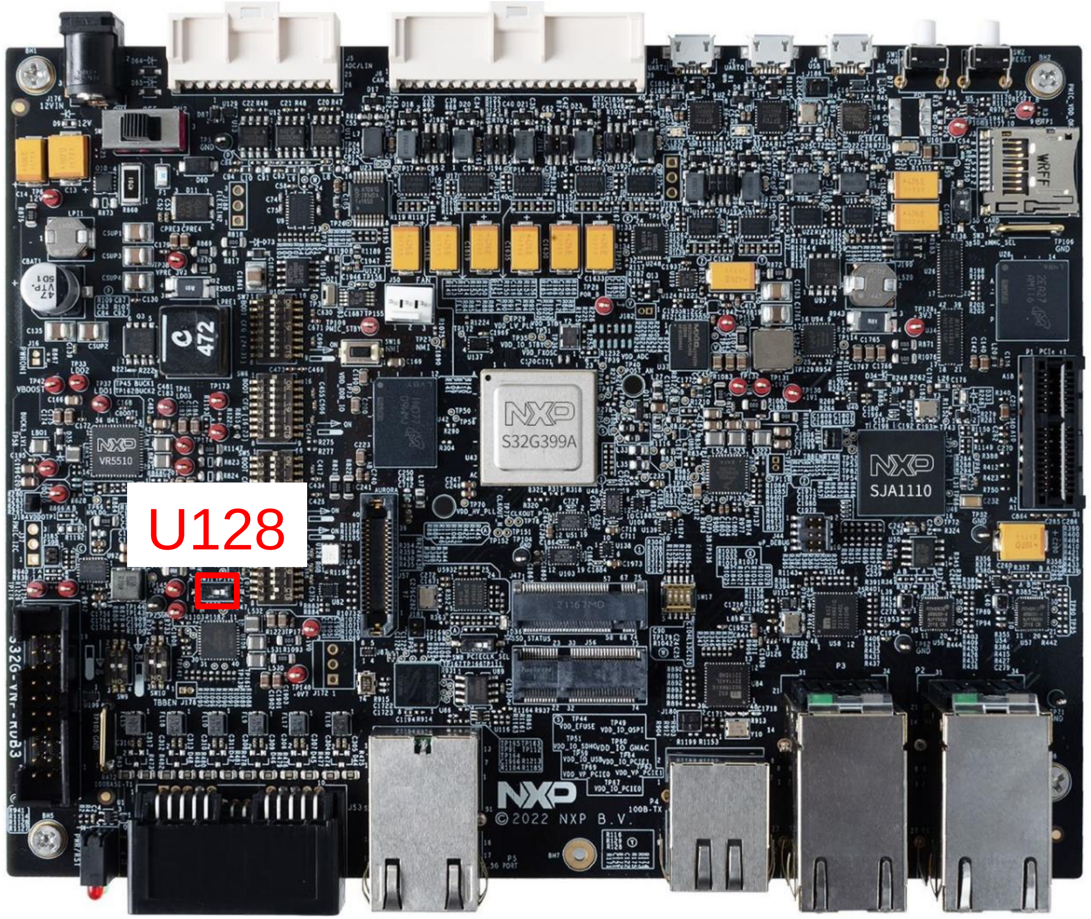
Restart S32 Design Studio for S32 Platform to apply the software update?

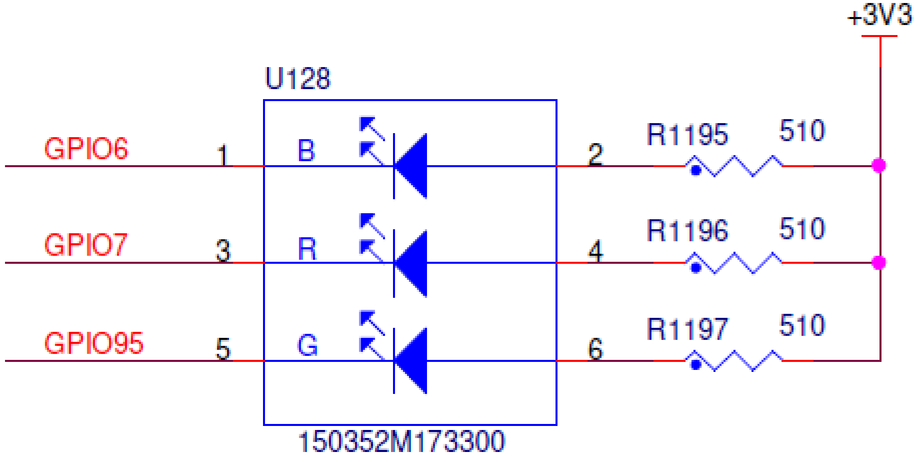
Light Up RGB LED using S32 Real Time Drivers

SECURE CONNECTIONS
FOR A SMARTER WORLD

LIGHT UP RGB LED: HARDWARE RESOURCES

- Resources to be used: on-board RGB LED



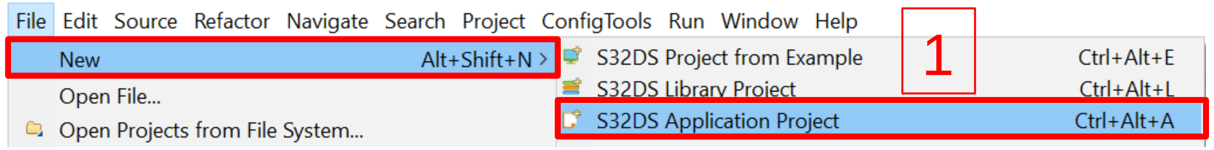


RGB

- An RGB LED is a combination of three LED in one package: 1x Blue LED, 1x Red LED and 1x Green LED.
- On this guide, only the Blue LED(GPIO6) is used.

LIGHT UP RGB LED: CREATE PROJECT

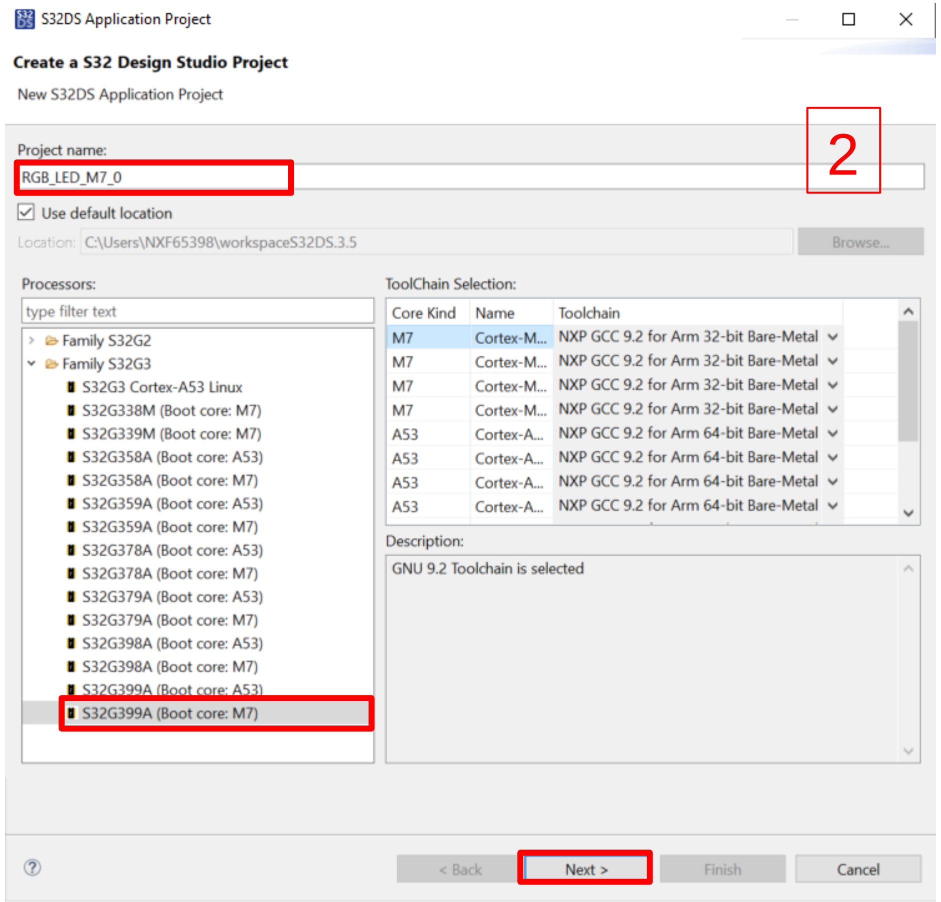
- Create a new S32DS Application Project



File Edit Source Refactor Navigate Search Project ConfigTools Run Window Help

New Alt+Shift+N >
Open File...
Open Projects from File System...
S32DS Project from Example Ctrl+Alt+E
S32DS Library Project Ctrl+Alt+L
S32DS Application Project Ctrl+Alt+A
1

- Input project name, select S32G399A(Boot core:M7) as processor, then click on “Next” button.



S32DS Application Project

Create a S32 Design Studio Project

New S32DS Application Project

Project name: RGB_LED_M7_0

☑ Use default location

Location: C:\Users\NXF65398\workspaceS32DS.3.5

Processors:

type filter text

- > Family S32G2
- ⌄ Family S32G3
  - S32G3 Cortex-A53 Linux
  - S32G338M (Boot core: M7)
  - S32G339M (Boot core: M7)
  - S32G358A (Boot core: A53)
  - S32G358A (Boot core: M7)
  - S32G359A (Boot core: A53)
  - S32G359A (Boot core: M7)
  - S32G378A (Boot core: A53)
  - S32G378A (Boot core: M7)
  - S32G379A (Boot core: A53)
  - S32G379A (Boot core: M7)
  - S32G398A (Boot core: A53)
  - S32G398A (Boot core: M7)
  - S32G399A (Boot core: A53)
  - S32G399A (Boot core: M7)

ToolChain Selection:

Core Kind	Name	Toolchain
M7	Cortex-M...	NXP GCC 9.2 for Arm 32-bit Bare-Metal
M7	Cortex-M...	NXP GCC 9.2 for Arm 32-bit Bare-Metal
M7	Cortex-M...	NXP GCC 9.2 for Arm 32-bit Bare-Metal
M7	Cortex-M...	NXP GCC 9.2 for Arm 32-bit Bare-Metal
A53	Cortex-A...	NXP GCC 9.2 for Arm 64-bit Bare-Metal
A53	Cortex-A...	NXP GCC 9.2 for Arm 64-bit Bare-Metal
A53	Cortex-A...	NXP GCC 9.2 for Arm 64-bit Bare-Metal
A53	Cortex-A...	NXP GCC 9.2 for Arm 64-bit Bare-Metal

Description:

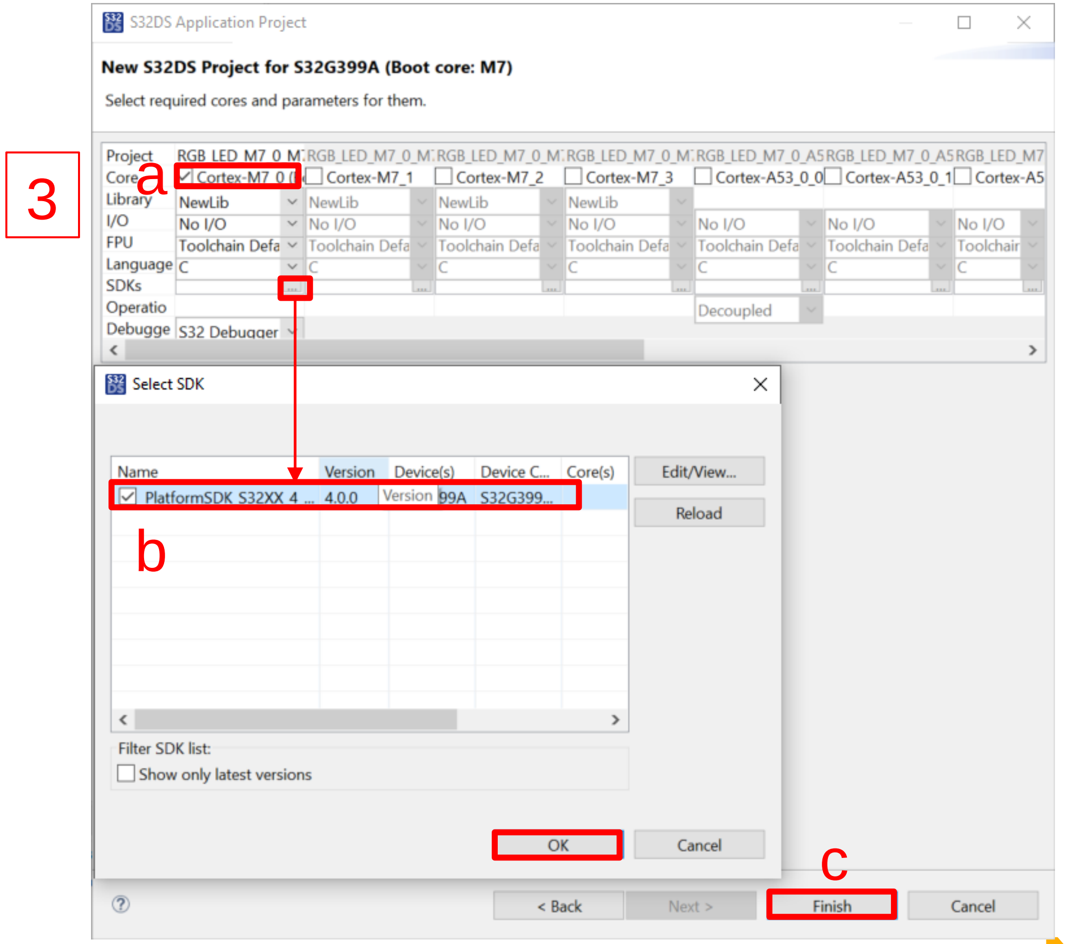
GNU 9.2 Toolchain is selected

- Select required core and SDKs

 a. Check only Cortex-M7_0 core option and uncheck other cores options.

 b. Select ‘PlatfprmSDK_S32XX**(4.0.0) as SDKs and click on “ok” button.

 c. Click on “Finish” button to complete configuration.



S32DS Application Project

New S32DS Project for S32G399A (Boot core: M7)

Select required cores and parameters for them.

Project	RGB_LED_M7_0_M	RGB_LED_M7_0_M	RGB_LED_M7_0_M	RGB_LED_M7_0_M	RGB_LED_M7_0_A5	RGB_LED_M7_0_A5	RGB_LED_M7
Core	☑ Cortex-M7_0 (	□ Cortex-M7_1	□ Cortex-M7_2	□ Cortex-M7_3	□ Cortex-A53_0_0	□ Cortex-A53_0_1	□ Cortex-A5
Library	NewLib	NewLib	NewLib	NewLib			
I/O	No I/O	No I/O	No I/O	No I/O	No I/O	No I/O	No I/O
FPU	Toolchain Defa	Toolchain Defa	Toolchain Defa	Toolchain Defa	Toolchain Defa	Toolchain Defa	Toolchair
Language	C	C	C	C	C	C	C
SDKs	...	...	...	...	...	...	...
Operatio					Decoupled		
Debugge	S32 Debugger						

Select SDK

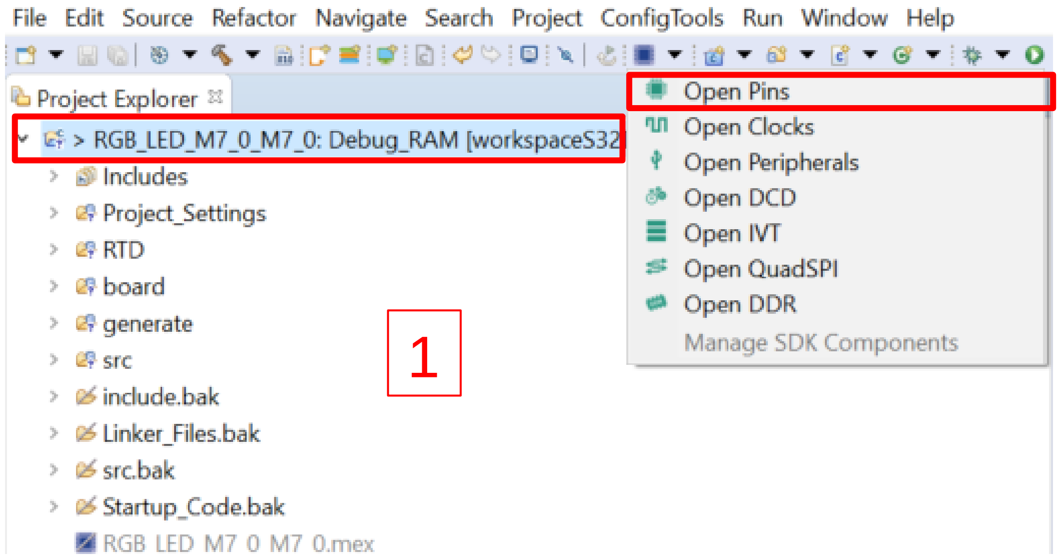
Name	Version	Device(s)	Device C...	Core(s)
☑ PlatformSDK_S32XX_4 ...	4.0.0	Version 99A	S32G399...	

Filter SDK list:

□ Show only latest versions

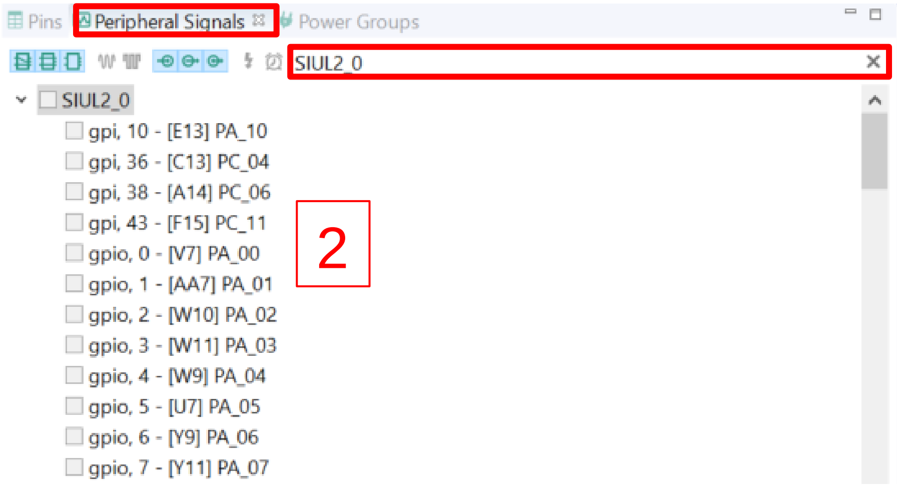
LIGHT UP RGB LED: PINS CONFIGURATION

- Select the created project and open Pins tool



RGB_LED_M7_0_M7_0: Debug_RAM [workspaceS32
- Includes
- Project_Settings
- RTD
- board
- generate
- src
- include.bak
- Linker_Files.bak
- src.bak
- Startup_Code.bak
- RGB_LED_M7_0_M7_0.mex

- Select peripheral Signals, input ”SIUL2_0” and find out SIUL2_0 item



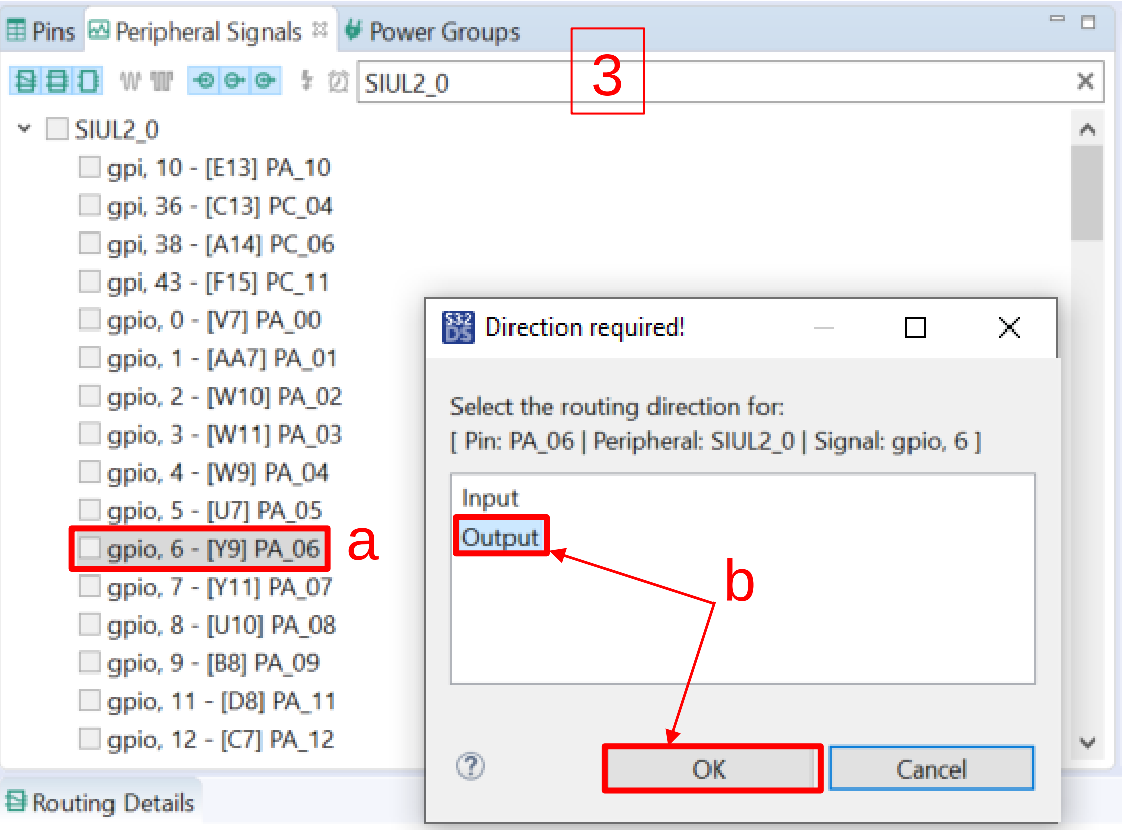
SIUL2_0

- □ SIUL2_0
  - □ gpi, 10 - [E13] PA_10
  - □ gpi, 36 - [C13] PC_04
  - □ gpi, 38 - [A14] PC_06
  - □ gpi, 43 - [F15] PC_11
  - □ gpio, 0 - [V7] PA_00
  - □ gpio, 1 - [AA7] PA_01
  - □ gpio, 2 - [W10] PA_02
  - □ gpio, 3 - [W11] PA_03
  - □ gpio, 4 - [W9] PA_04
  - □ gpio, 5 - [U7] PA_05
  - □ gpio, 6 - [Y9] PA_06
  - □ gpio, 7 - [Y11] PA_07

- Configure the corresponding gpio pins.

  a. Check gpio, 6 option.

  b. Click on “Output” button and ”OK” button to complete the pin configuration.



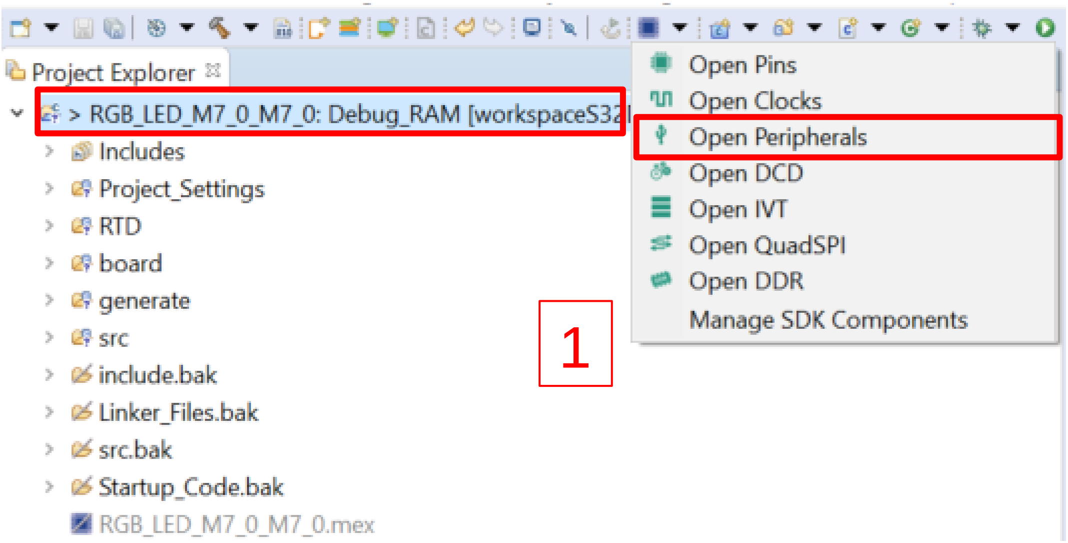
SIUL2_0

- □ SIUL2_0
  - □ gpi, 10 - [E13] PA_10
  - □ gpi, 36 - [C13] PC_04
  - □ gpi, 38 - [A14] PC_06
  - □ gpi, 43 - [F15] PC_11
  - □ gpio, 0 - [V7] PA_00
  - □ gpio, 1 - [AA7] PA_01
  - □ gpio, 2 - [W10] PA_02
  - □ gpio, 3 - [W11] PA_03
  - □ gpio, 4 - [W9] PA_04
  - □ gpio, 5 - [U7] PA_05
  - □ gpio, 6 - [Y9] PA_06
  - □ gpio, 7 - [Y11] PA_07
  - □ gpio, 8 - [U10] PA_08
  - □ gpio, 9 - [B8] PA_09
  - □ gpio, 11 - [D8] PA_11
  - □ gpio, 12 - [C7] PA_12

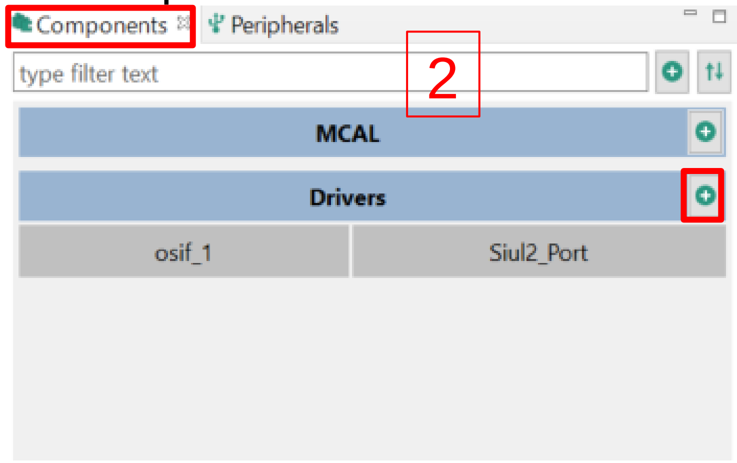
Select the routing direction for:
[Pin: PA_06 | Peripheral: SIUL2_0 | Signal: gpio, 6]

LIGHT UP RGB LED: ADD GPIO DRIVERS

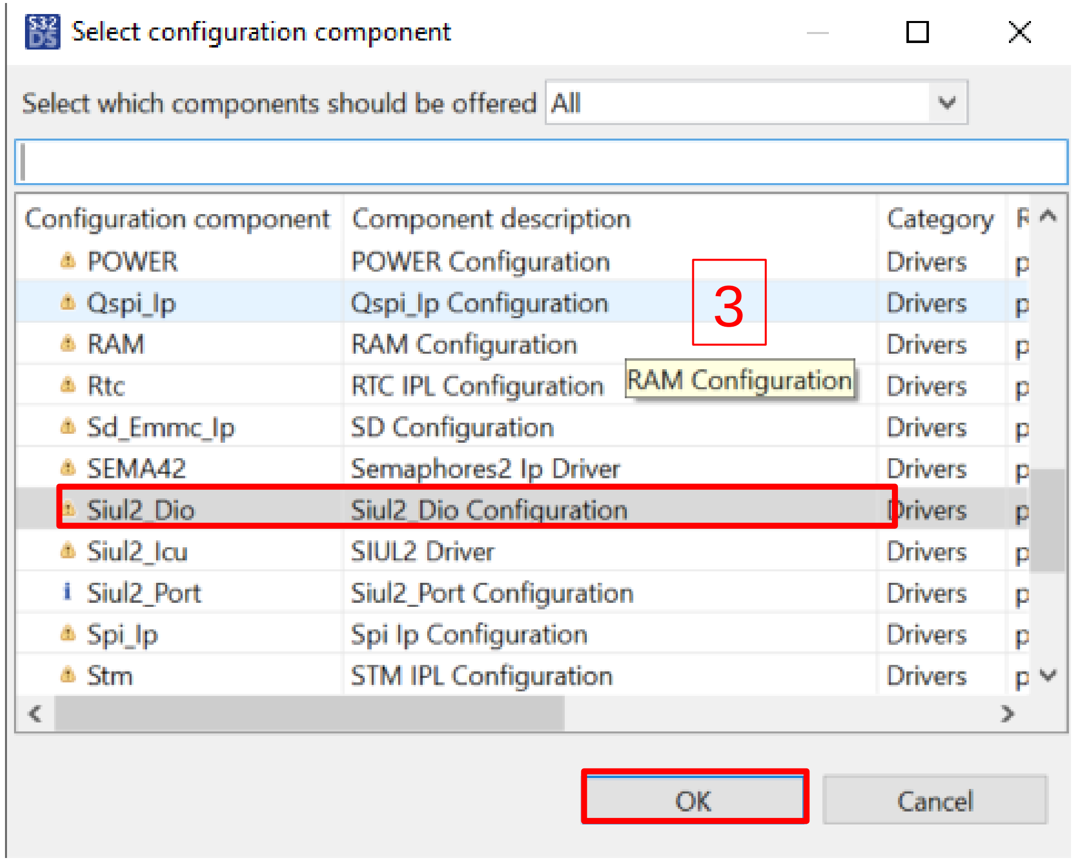
- Select the created project and open Peripherals tool



- Select Components to find out Drivers option and Click on ”+” option.



- Select ”Siul2_Dio” option and click on ”OK” button to add Siul2_Dio driver.

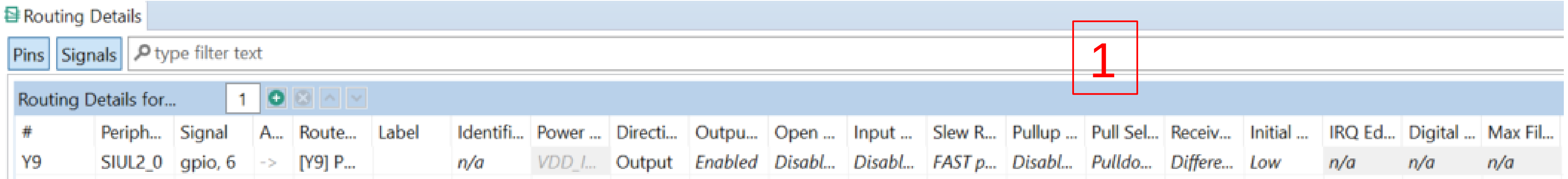

Select configuration component

Select which components should be offered: All

Configuration component	Component description	Category	F
POWER	POWER Configuration	Drivers	p
Qspi_Ip	Qspi_Ip Configuration	Drivers	p
RAM	RAM Configuration	Drivers	p
Rtc	RTC IPL Configuration	Drivers	p
Sd_Emmc_Ip	SD Configuration	Drivers	p
SEMA42	Semaphores2 Ip Driver	Drivers	p
Siul2_Dio	Siul2_Dio Configuration	Drivers	p
Siul2_Icu	SIUL2 Driver	Drivers	p
Siul2_Port	Siul2_Port Configuration	Drivers	p
Spi_Ip	Spi Ip Configuration	Drivers	p
Stm	STM IPL Configuration	Drivers	p

LIGHT UP RGB LED: CHECK CONFIGURATION AND UPDATE CODE

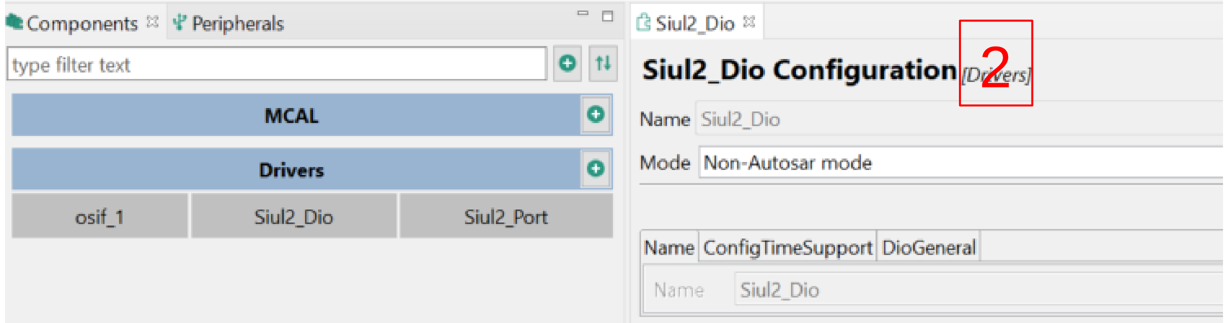
- Open Pins tool to check configuration



Routing Details for... 1

#	Periph...	Signal	A...	Route...	Label	Identifi...	Power ...	Directi...	Outpu...	Open ...	Input ...	Slew R...	Pullup ...	Pull Sel...	Receiv...	Initial ...	IRQ Ed...	Digital ...	Max Fil...
Y9	SIUL2_0	gpio, 6	->	[Y9] P...		n/a	VDD_I...	Output	Enabled	Disabl...	Disabl...	FAST p...	Disabl...	Pulldo...	Differe...	Low	n/a	n/a	n/a

- Open Peripherals tool to check configuration



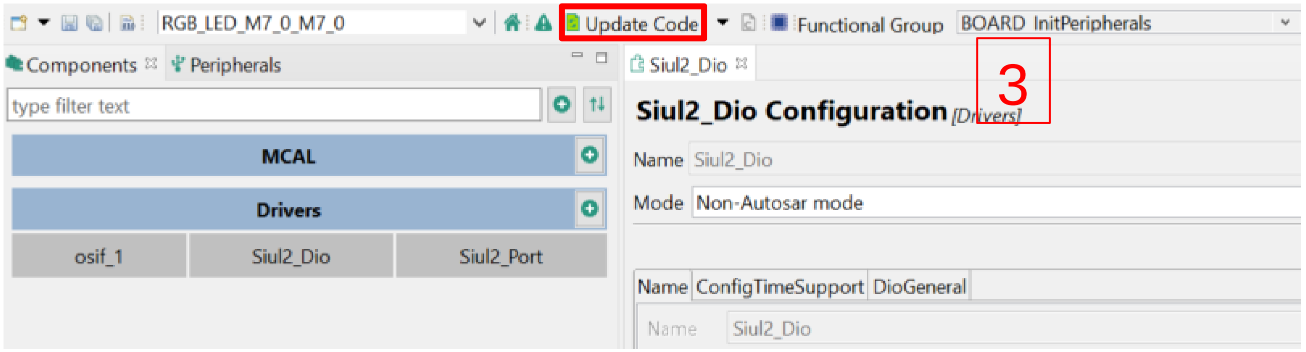
Siul2_Dio Configuration [Drivers]

Name Siul2_Dio

Mode Non-Autosar mode

Name	ConfigTimeSupport	DioGeneral
Name Siul2_Dio		

- Click on “Update Code” button.



Siul2_Dio Configuration [Drivers]

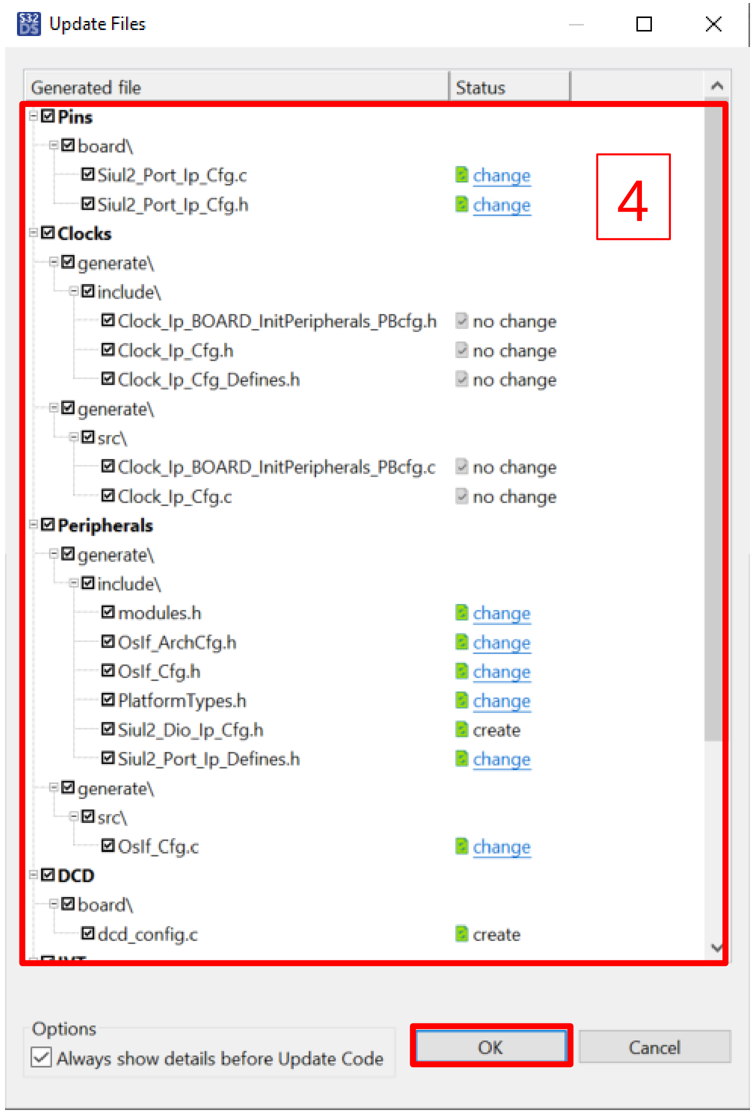
Name Siul2_Dio

Mode Non-Autosar mode

Name	ConfigTimeSupport	DioGeneral
Name Siul2_Dio		

LIGHT UP RGB LED: CHECK CONFIGURATION AND UPDATE CODE

- Click on ”OK” button to start to update codes.



Update Files

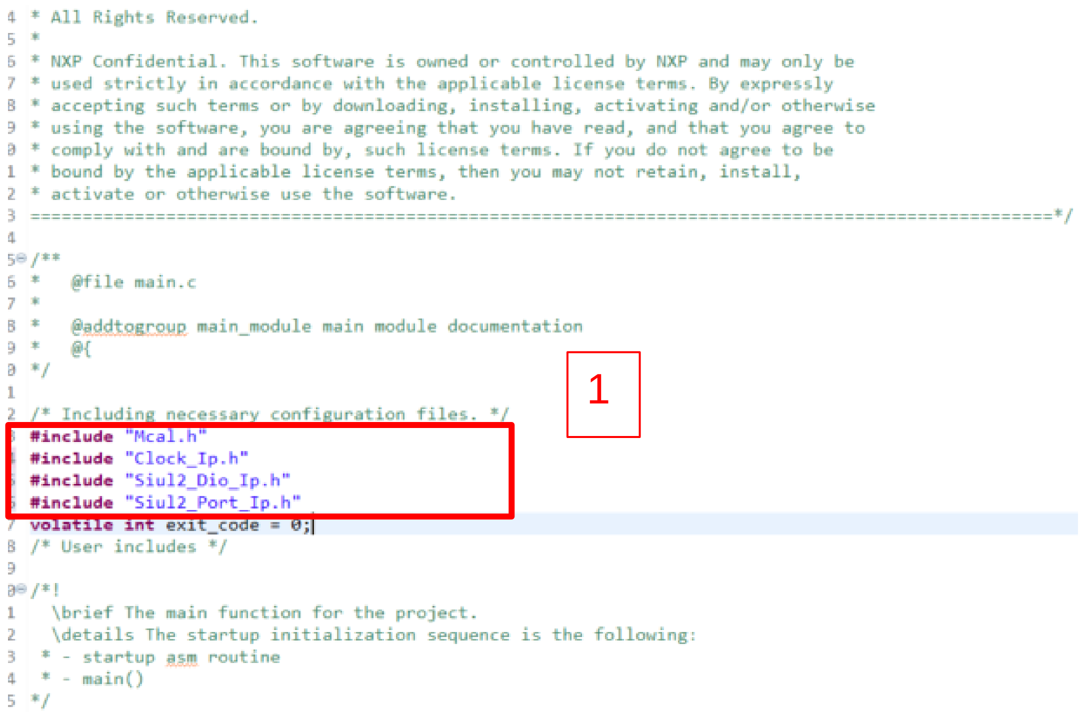
Generated file	Status
☑ **Pins**	
☑ board\	
☑ Siul2_Port_Ip_Cfg.c	change
☑ Siul2_Port_Ip_Cfg.h	change
☑ **Clocks**	
☑ generate\	
☑ include\	
☑ Clock_Ip_BOARD_InitPeripherals_PBcfg.h	no change
☑ Clock_Ip_Cfg.h	no change
☑ Clock_Ip_Cfg_Defines.h	no change
☑ generate\	
☑ src\	
☑ Clock_Ip_BOARD_InitPeripherals_PBcfg.c	no change
☑ Clock_Ip_Cfg.c	no change
☑ **Peripherals**	
☑ generate\	
☑ include\	
☑ modules.h	change
☑ OsIf_ArchCfg.h	change
☑ OsIf_Cfg.h	change
☑ PlatformTypes.h	change
☑ Siul2_Dio_Ip_Cfg.h	create
☑ Siul2_Port_Ip_Defines.h	change
☑ generate\	
☑ src\	
☑ OsIf_Cfg.c	change
☑ **DCD**	
☑ board\	
☑ dcd_config.c	create

Options

☑ Always show details before Update Code

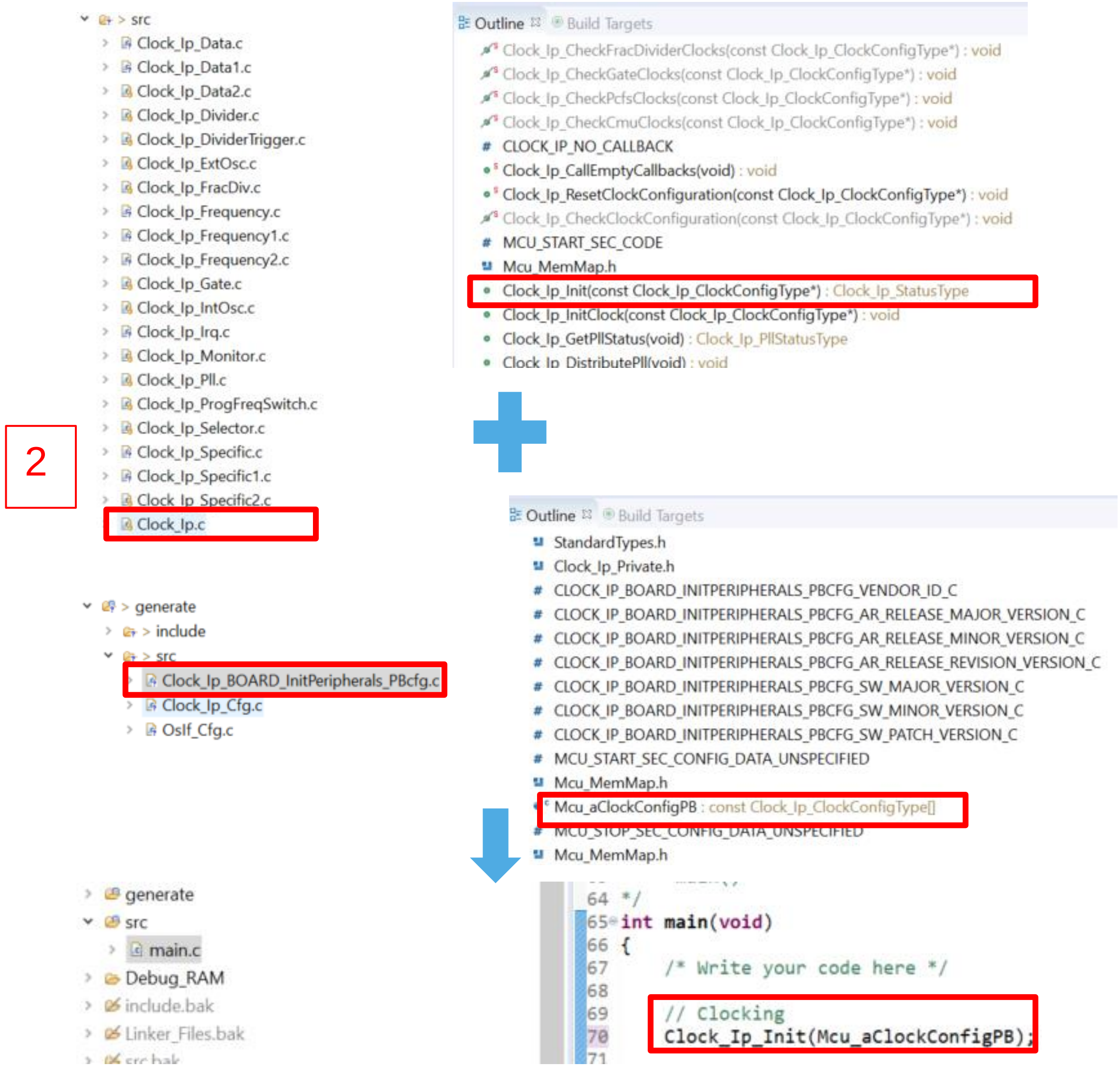
LIGHT UP RGB LED: APPLICATION CODE

- Add header files of project configuration and module drivers in main.c file



```
 * All Rights Reserved.
 *
 * NXP Confidential. This software is owned or controlled by NXP and may only be
 * used strictly in accordance with the applicable license terms. By expressly
 * accepting such terms or by downloading, installing, activating and/or otherwise
 * using the software, you are agreeing that you have read, and that you agree to
 * comply with and are bound by, such license terms. If you do not agree to be
 * bound by the applicable license terms, then you may not retain, install,
 * activate or otherwise use the software.
==================================================================================================*/

/**
 *   @file main.c
 *
 *   @addtogroup main_module main module documentation
 *   @{
 */

/* Including necessary configuration files. */
#include "Mcal.h"
#include "Clock_Ip.h"
#include "Siul2_Dio_Ip.h"
#include "Siul2_Port_Ip.h"
volatile int exit_code = 0;
/* User includes */

/*!
  \brief The main function for the project.
  \details The startup initialization sequence is the following:
 * - startup asm routine
 * - main()
 */
```

- Initialize clocks



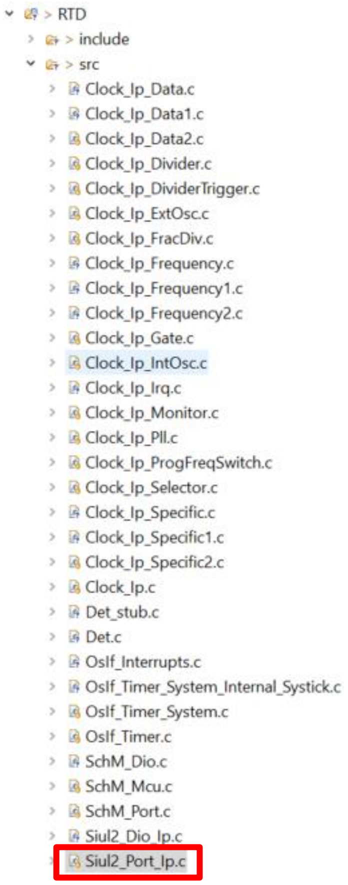
```
Clock_Ip_CheckFracDividerClocks(const Clock_Ip_ClockConfigType*) : void
Clock_Ip_CheckGateClocks(const Clock_Ip_ClockConfigType*) : void
Clock_Ip_CheckPcfsClocks(const Clock_Ip_ClockConfigType*) : void
Clock_Ip_CheckCmuClocks(const Clock_Ip_ClockConfigType*) : void
CLOCK_IP_NO_CALLBACK
Clock_Ip_CallEmptyCallbacks(void) : void
Clock_Ip_ResetClockConfiguration(const Clock_Ip_ClockConfigType*) : void
Clock_Ip_CheckClockConfiguration(const Clock_Ip_ClockConfigType*) : void
MCU_START_SEC_CODE
Mcu_MemMap.h
Clock_Ip_Init(const Clock_Ip_ClockConfigType*) : Clock_Ip_StatusType
Clock_Ip_InitClock(const Clock_Ip_ClockConfigType*) : void
Clock_Ip_GetPllStatus(void) : Clock_Ip_PllStatusType
Clock_Ip_DistributePll(void) : void
```

```
StandardTypes.h
Clock_Ip_Private.h
CLOCK_IP_BOARD_INITPERIPHERALS_PBCFG_VENDOR_ID_C
CLOCK_IP_BOARD_INITPERIPHERALS_PBCFG_AR_RELEASE_MAJOR_VERSION_C
CLOCK_IP_BOARD_INITPERIPHERALS_PBCFG_AR_RELEASE_MINOR_VERSION_C
CLOCK_IP_BOARD_INITPERIPHERALS_PBCFG_AR_RELEASE_REVISION_VERSION_C
CLOCK_IP_BOARD_INITPERIPHERALS_PBCFG_SW_MAJOR_VERSION_C
CLOCK_IP_BOARD_INITPERIPHERALS_PBCFG_SW_MINOR_VERSION_C
CLOCK_IP_BOARD_INITPERIPHERALS_PBCFG_SW_PATCH_VERSION_C
MCU_START_SEC_CONFIG_DATA_UNSPECIFIED
Mcu_MemMap.h
Mcu_aClockConfigPB : const Clock_Ip_ClockConfigType[]
MCU_STOP_SEC_CONFIG_DATA_UNSPECIFIED
Mcu_MemMap.h
```

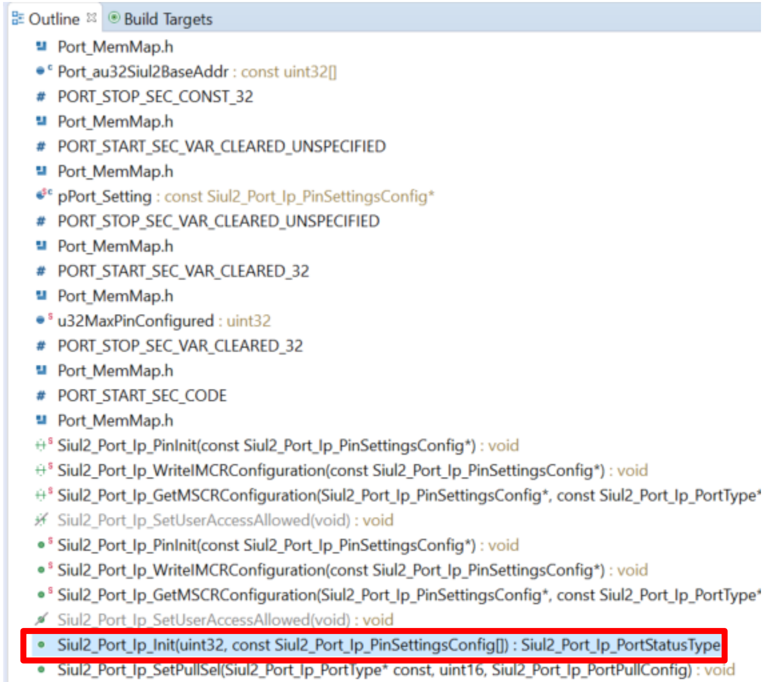
```
 */
int main(void)
{
    /* Write your code here */

    // Clocking
    Clock_Ip_Init(Mcu_aClockConfigPB);
```

LIGHT UP RGB LED: APPLICATION CODE

- Initialize SIUL2



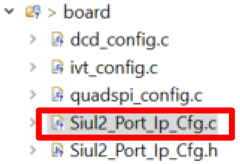
- RTD
  - include
  - src
    - Clock_Ip_Data.c
    - Clock_Ip_Data1.c
    - Clock_Ip_Data2.c
    - Clock_Ip_Divider.c
    - Clock_Ip_DividerTrigger.c
    - Clock_Ip_ExtOsc.c
    - Clock_Ip_FracDiv.c
    - Clock_Ip_Frequency.c
    - Clock_Ip_Frequency1.c
    - Clock_Ip_Frequency2.c
    - Clock_Ip_Gate.c
    - Clock_Ip_IntOsc.c
    - Clock_Ip_Irq.c
    - Clock_Ip_Monitor.c
    - Clock_Ip_Pll.c
    - Clock_Ip_ProgFreqSwitch.c
    - Clock_Ip_Selector.c
    - Clock_Ip_Specific.c
    - Clock_Ip_Specific1.c
    - Clock_Ip_Specific2.c
    - Clock_Ip.c
    - Det_stub.c
    - Det.c
    - OsIf_Interrupts.c
    - OsIf_Timer_System_Internal_Systick.c
    - OsIf_Timer_System.c
    - OsIf_Timer.c
    - SchM_Dio.c
    - SchM_Mcu.c
    - SchM_Port.c
    - Siul2_Dio_Ip.c
    - Siul2_Port_Ip.c



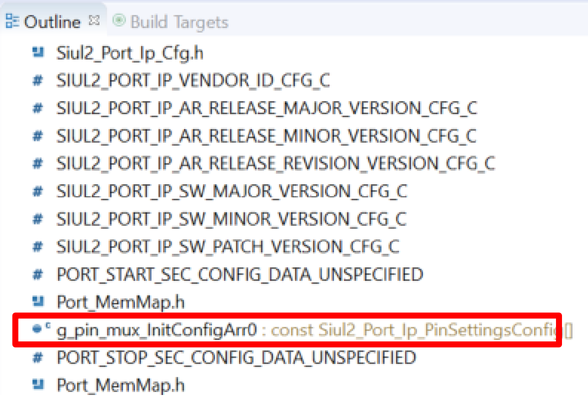
Outline | Build Targets

- Port_MemMap.h
- Port_au32Siul2BaseAddr : const uint32[]
- PORT_STOP_SEC_CONST_32
- Port_MemMap.h
- PORT_START_SEC_VAR_CLEARED_UNSPECIFIED
- Port_MemMap.h
- pPort_Setting : const Siul2_Port_Ip_PinSettingsConfig*
- PORT_STOP_SEC_VAR_CLEARED_UNSPECIFIED
- Port_MemMap.h
- PORT_START_SEC_VAR_CLEARED_32
- Port_MemMap.h
- u32MaxPinConfigured : uint32
- PORT_STOP_SEC_VAR_CLEARED_32
- Port_MemMap.h
- PORT_START_SEC_CODE
- Port_MemMap.h
- Siul2_Port_Ip_PinInit(const Siul2_Port_Ip_PinSettingsConfig*) : void
- Siul2_Port_Ip_WriteIMCRConfiguration(const Siul2_Port_Ip_PinSettingsConfig*) : void
- Siul2_Port_Ip_GetMSCRConfiguration(Siul2_Port_Ip_PinSettingsConfig*, const Siul2_Port_Ip_PortType*
- Siul2_Port_Ip_SetUserAccessAllowed(void) : void
- Siul2_Port_Ip_PinInit(const Siul2_Port_Ip_PinSettingsConfig*) : void
- Siul2_Port_Ip_WriteIMCRConfiguration(const Siul2_Port_Ip_PinSettingsConfig*) : void
- Siul2_Port_Ip_GetMSCRConfiguration(Siul2_Port_Ip_PinSettingsConfig*, const Siul2_Port_Ip_PortType*
- Siul2_Port_Ip_SetUserAccessAllowed(void) : void
- Siul2_Port_Ip_Init(uint32, const Siul2_Port_Ip_PinSettingsConfig[]) : Siul2_Port_Ip_PortStatusType
- Siul2_Port_Ip_SetPullSel(Siul2_Port_Ip_PortType* const, uint16, Siul2_Port_Ip_PortPullConfig) : void

3

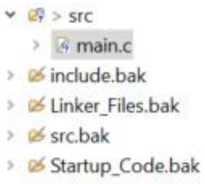


- board
  - dcd_config.c
  - ivt_config.c
  - quadspi_config.c
  - Siul2_Port_Ip_Cfg.c
  - Siul2_Port_Ip_Cfg.h

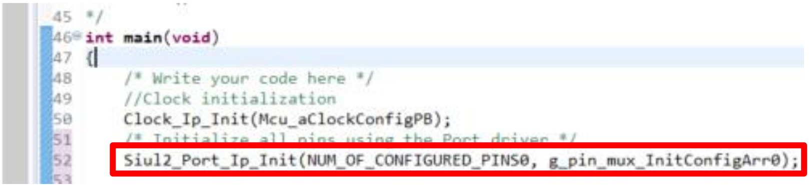


Outline | Build Targets

- Siul2_Port_Ip_Cfg.h
- SIUL2_PORT_IP_VENDOR_ID_CFG_C
- SIUL2_PORT_IP_AR_RELEASE_MAJOR_VERSION_CFG_C
- SIUL2_PORT_IP_AR_RELEASE_MINOR_VERSION_CFG_C
- SIUL2_PORT_IP_AR_RELEASE_REVISION_VERSION_CFG_C
- SIUL2_PORT_IP_SW_MAJOR_VERSION_CFG_C
- SIUL2_PORT_IP_SW_MINOR_VERSION_CFG_C
- SIUL2_PORT_IP_SW_PATCH_VERSION_CFG_C
- PORT_START_SEC_CONFIG_DATA_UNSPECIFIED
- Port_MemMap.h
- g_pin_mux_InitConfigArr0 : const Siul2_Port_Ip_PinSettingsConfi[]
- PORT_STOP_SEC_CONFIG_DATA_UNSPECIFIED
- Port_MemMap.h



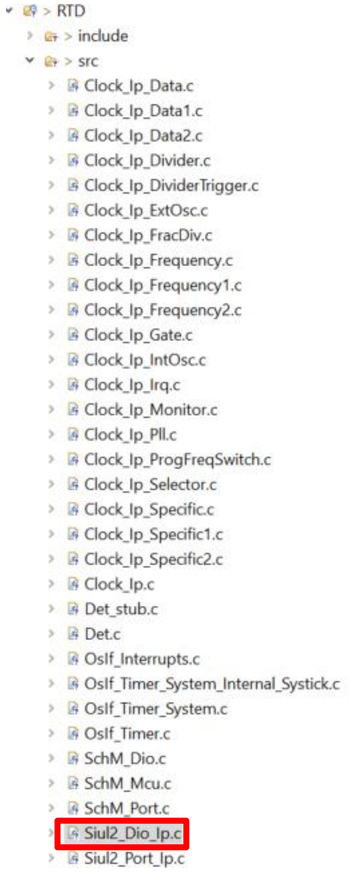
- src
  - main.c
- include.bak
- Linker_Files.bak
- src.bak
- Startup_Code.bak



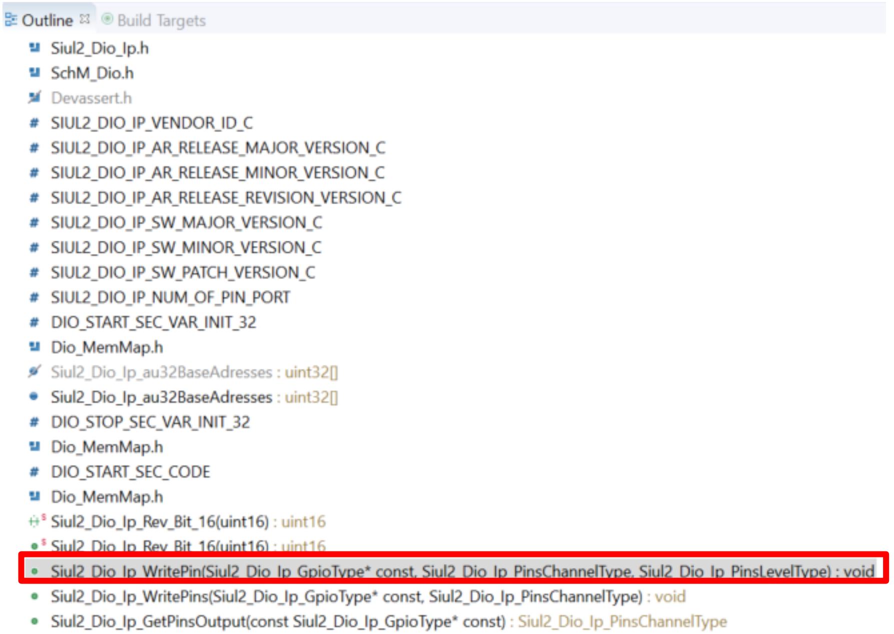
```
*/
int main(void)
{
    /* Write your code here */
    //Clock initialization
    Clock_Ip_Init(Mcu_aClockConfigPB);
    /* Initialize all pins using the Port driver */
    Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0);
```

LIGHT UP RGB LED: APPLICATION CODE

- Add the implementation of lighting up LED

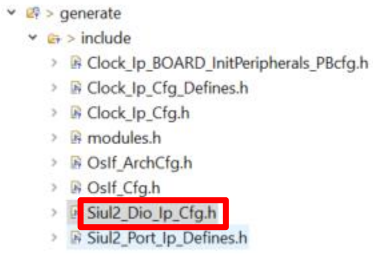


```
RTD
  include
  src
    Clock_Ip_Data.c
    Clock_Ip_Data1.c
    Clock_Ip_Data2.c
    Clock_Ip_Divider.c
    Clock_Ip_DividerTrigger.c
    Clock_Ip_ExtOsc.c
    Clock_Ip_FracDiv.c
    Clock_Ip_Frequency.c
    Clock_Ip_Frequency1.c
    Clock_Ip_Frequency2.c
    Clock_Ip_Gate.c
    Clock_Ip_IntOsc.c
    Clock_Ip_Irq.c
    Clock_Ip_Monitor.c
    Clock_Ip_Pll.c
    Clock_Ip_ProgFreqSwitch.c
    Clock_Ip_Selector.c
    Clock_Ip_Specific.c
    Clock_Ip_Specific1.c
    Clock_Ip_Specific2.c
    Clock_Ip.c
    Det_stub.c
    Det.c
    OsIf_Interrupts.c
    OsIf_Timer_System_Internal_Systick.c
    OsIf_Timer_System.c
    OsIf_Timer.c
    SchM_Dio.c
    SchM_Mcu.c
    SchM_Port.c
    Siul2_Dio_Ip.c
    Siul2_Port_Ip.c
```

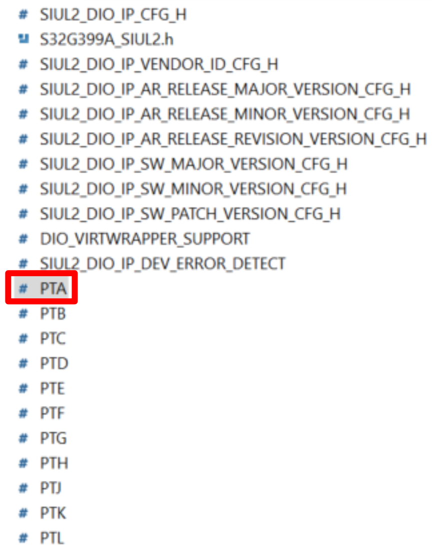


```
Outline    Build Targets
Siul2_Dio_Ip.h
SchM_Dio.h
Devassert.h
SIUL2_DIO_IP_VENDOR_ID_C
SIUL2_DIO_IP_AR_RELEASE_MAJOR_VERSION_C
SIUL2_DIO_IP_AR_RELEASE_MINOR_VERSION_C
SIUL2_DIO_IP_AR_RELEASE_REVISION_VERSION_C
SIUL2_DIO_IP_SW_MAJOR_VERSION_C
SIUL2_DIO_IP_SW_MINOR_VERSION_C
SIUL2_DIO_IP_SW_PATCH_VERSION_C
SIUL2_DIO_IP_NUM_OF_PIN_PORT
DIO_START_SEC_VAR_INIT_32
Dio_MemMap.h
Siul2_Dio_Ip_au32BaseAdresses : uint32[]
Siul2_Dio_Ip_au32BaseAdresses : uint32[]
DIO_STOP_SEC_VAR_INIT_32
Dio_MemMap.h
DIO_START_SEC_CODE
Dio_MemMap.h
Siul2_Dio_Ip_Rev_Bit_16(uint16) : uint16
Siul2_Dio_Ip_Rev_Bit_16(uint16) : uint16
Siul2_Dio_Ip_WritePin(Siul2_Dio_Ip_GpioType* const, Siul2_Dio_Ip_PinsChannelType, Siul2_Dio_Ip_PinsLevelType) : void
Siul2_Dio_Ip_WritePins(Siul2_Dio_Ip_GpioType* const, Siul2_Dio_Ip_PinsChannelType) : void
Siul2_Dio_Ip_GetPinsOutput(const Siul2_Dio_Ip_GpioType* const) : Siul2_Dio_Ip_PinsChannelType
```

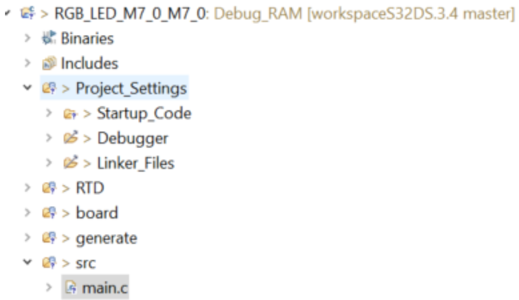
4



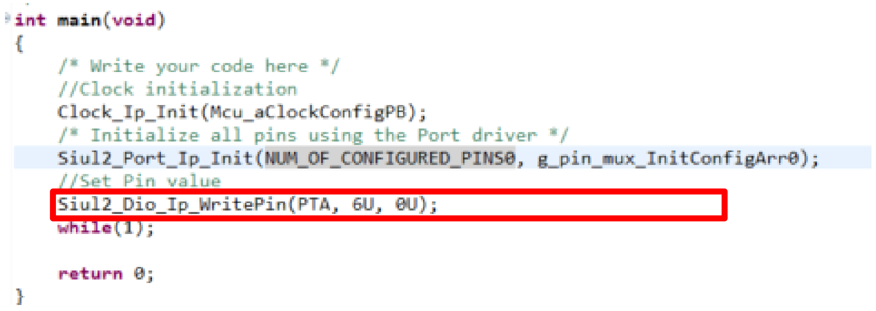
```
generate
  include
    Clock_Ip_BOARD_InitPeripherals_PBcfg.h
    Clock_Ip_Cfg_Defines.h
    Clock_Ip_Cfg.h
    modules.h
    OsIf_ArchCfg.h
    OsIf_Cfg.h
    Siul2_Dio_Ip_Cfg.h
    Siul2_Port_Ip_Defines.h
```



```
SIUL2_DIO_IP_CFG_H
S32G399A_SIUL2.h
SIUL2_DIO_IP_VENDOR_ID_CFG_H
SIUL2_DIO_IP_AR_RELEASE_MAJOR_VERSION_CFG_H
SIUL2_DIO_IP_AR_RELEASE_MINOR_VERSION_CFG_H
SIUL2_DIO_IP_AR_RELEASE_REVISION_VERSION_CFG_H
SIUL2_DIO_IP_SW_MAJOR_VERSION_CFG_H
SIUL2_DIO_IP_SW_MINOR_VERSION_CFG_H
SIUL2_DIO_IP_SW_PATCH_VERSION_CFG_H
DIO_VIRTWRAPPER_SUPPORT
SIUL2_DIO_IP_DEV_ERROR_DETECT
PTA
PTB
PTC
PTD
PTE
PTF
PTG
PTH
PTJ
PTK
PTL
```



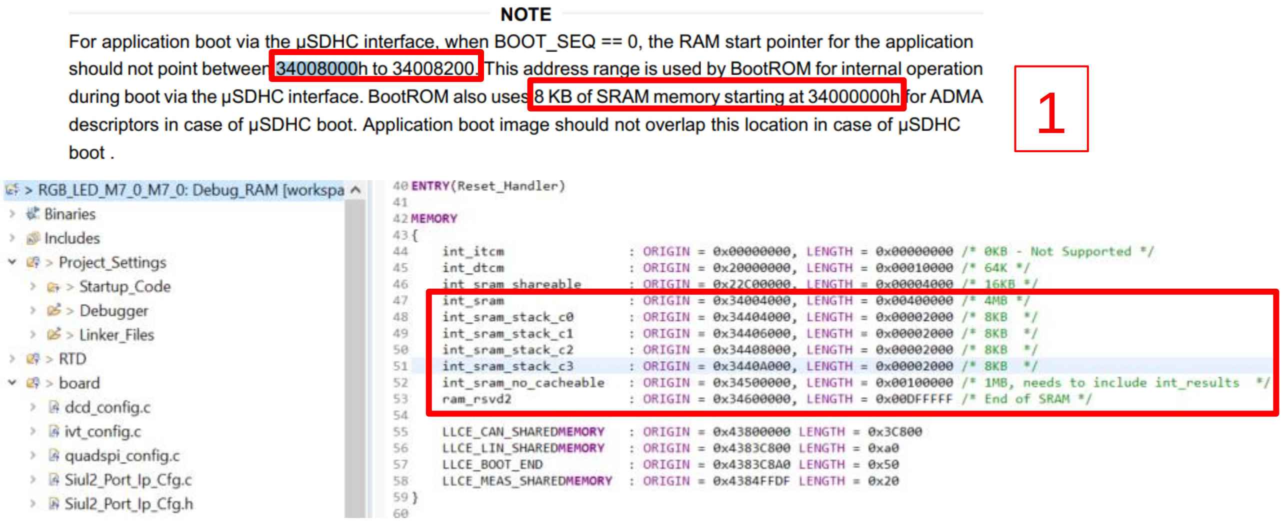
```
RGB_LED_M7_0_M7_0: Debug_RAM [workspaceS32DS.3.4 master]
  Binaries
  Includes
  Project_Settings
    Startup_Code
    Debugger
    Linker_Files
  RTD
  board
  generate
  src
    main.c
```



```
int main(void)
{
    /* Write your code here */
    //Clock initialization
    Clock_Ip_Init(Mcu_aClockConfigPB);
    /* Initialize all pins using the Port driver */
    Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0);
    //Set Pin value
    Siul2_Dio_Ip_WritePin(PTA, 6U, 0U);
    while(1);

    return 0;
}
```

LIGHT UP RGB LED: BUILD PROJECT AND GENERATE .BIN FILE

- Open and modify the link file according to the noted information from reference manual of S32G3

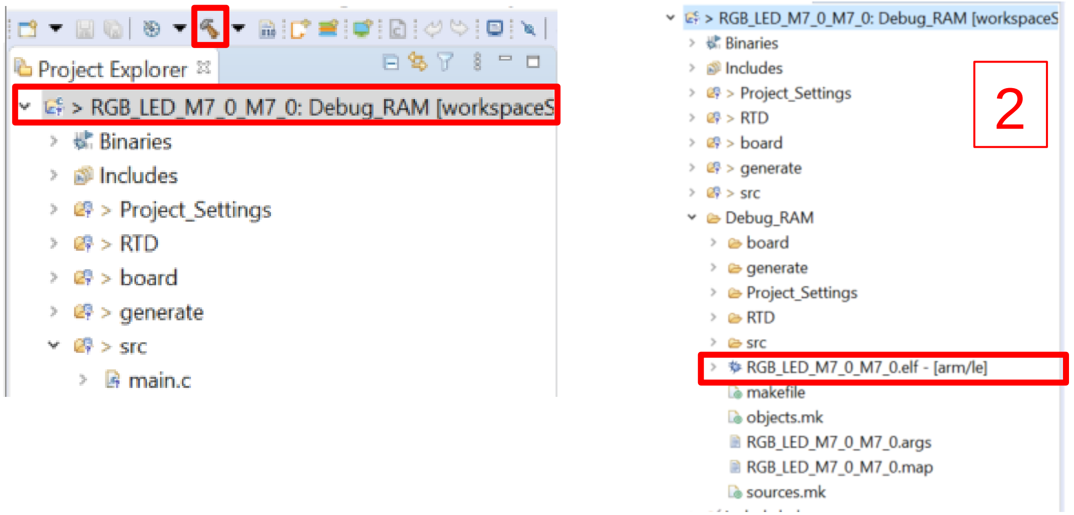


NOTE

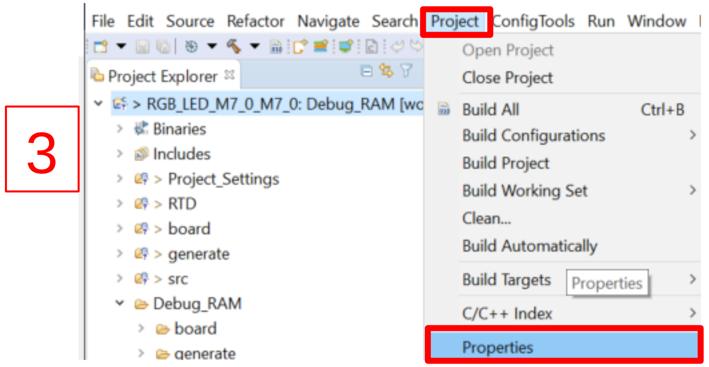
For application boot via the µSDHC interface, when BOOT_SEQ == 0, the RAM start pointer for the application should not point between 34008000h to 34008200. This address range is used by BootROM for internal operation during boot via the µSDHC interface. BootROM also uses 8 KB of SRAM memory starting at 34000000h for ADMA descriptors in case of µSDHC boot. Application boot image should not overlap this location in case of µSDHC boot .

```
ENTRY(Reset_Handler)

MEMORY
{
    int_itcm                : ORIGIN = 0x00000000, LENGTH = 0x00000000 /* 0KB - Not Supported */
    int_dtcm                : ORIGIN = 0x20000000, LENGTH = 0x00010000 /* 64K */
    int_sram_shareable      : ORIGIN = 0x22C00000, LENGTH = 0x00004000 /* 16KB */
    int_sram                : ORIGIN = 0x34004000, LENGTH = 0x00400000 /* 4MB */
    int_sram_stack_c0       : ORIGIN = 0x34404000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_stack_c1       : ORIGIN = 0x34406000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_stack_c2       : ORIGIN = 0x34408000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_stack_c3       : ORIGIN = 0x3440A000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_no_cacheable   : ORIGIN = 0x34500000, LENGTH = 0x00100000 /* 1MB, needs to include int_results  */
    ram_rsvd2               : ORIGIN = 0x34600000, LENGTH = 0x00DFFFFF /* End of SRAM */

    LLCE_CAN_SHAREDMEMORY   : ORIGIN = 0x43800000 LENGTH = 0x3C800
    LLCE_LIN_SHAREDMEMORY   : ORIGIN = 0x4383C800 LENGTH = 0xa0
    LLCE_BOOT_END           : ORIGIN = 0x4383C8A0 LENGTH = 0x50
    LLCE_MEAS_SHAREDMEMORY  : ORIGIN = 0x4384FFDF LENGTH = 0x20
}
```

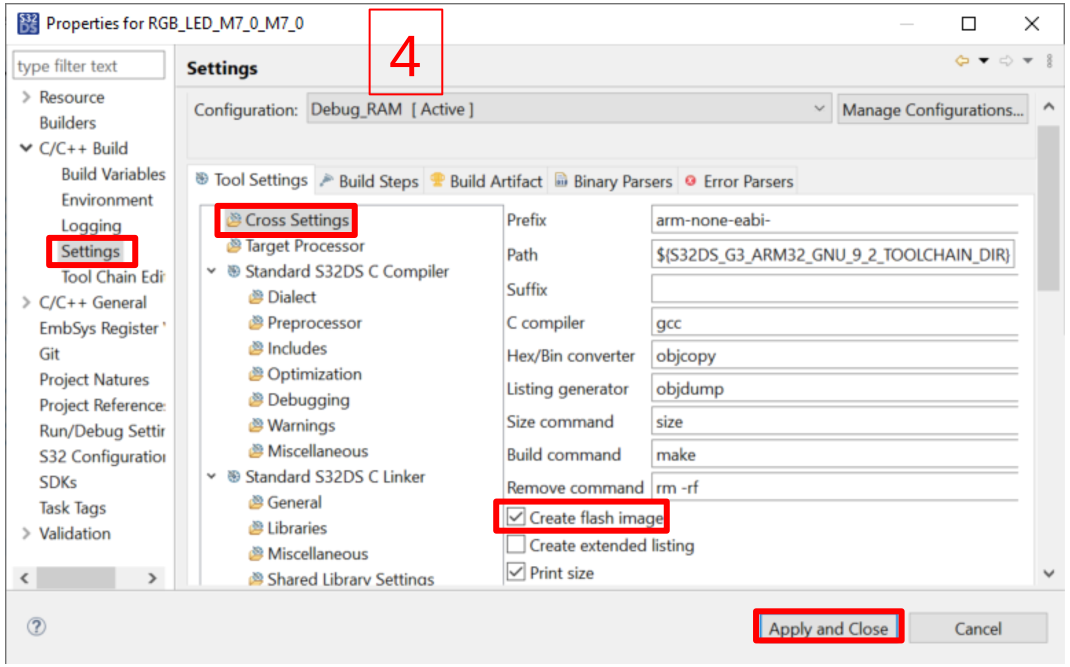
- Select and build project, the .elf file will be generated



- Open the properties of project



- Check the “Create flash image” box and click on “Apply and Close”



Properties for RGB_LED_M7_0_M7_0

Settings

Configuration: Debug_RAM [Active]

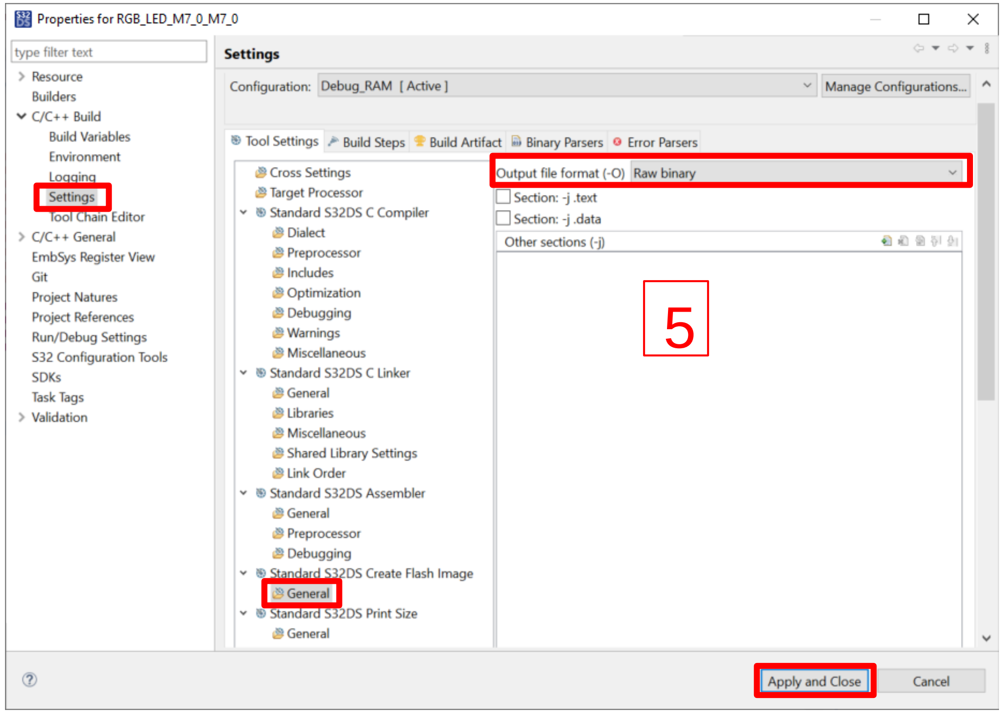
Tool Settings > Cross Settings

Prefix	arm-none-eabi-
Path	${S32DS_G3_ARM32_GNU_9_2_TOOLCHAIN_DIR}
Suffix	
C compiler	gcc
Hex/Bin converter	objcopy
Listing generator	objdump
Size command	size
Build command	make
Remove command	rm -rf

☑ Create flash image
☐ Create extended listing
☑ Print size

LIGHT UP RGB LED: BUILD PROJECT AND GENERATE .BIN FILE

- Re-open the properties of project, select “Raw binary” as output file format and click on “Apply and Close”



Properties for RGB_LED_M7_0_M7_0

type filter text

- Resource
- Builders
- C/C++ Build
  - Build Variables
  - Environment
  - Logging
  - Settings
  - Tool Chain Editor
- C/C++ General
- EmbSys Register View
- Git
- Project Natures
- Project References
- Run/Debug Settings
- S32 Configuration Tools
- SDKs
- Task Tags
- Validation

Settings

Configuration: Debug_RAM [Active] Manage Configurations...

Tool Settings | Build Steps | Build Artifact | Binary Parsers | Error Parsers

- Cross Settings
- Target Processor
- Standard S32DS C Compiler
  - Dialect
  - Preprocessor
  - Includes
  - Optimization
  - Debugging
  - Warnings
  - Miscellaneous
- Standard S32DS C Linker
  - General
  - Libraries
  - Miscellaneous
  - Shared Library Settings
  - Link Order
- Standard S32DS Assembler
  - General
  - Preprocessor
  - Debugging
- Standard S32DS Create Flash Image
  - General
- Standard S32DS Print Size
  - General

Output file format (-O) Raw binary

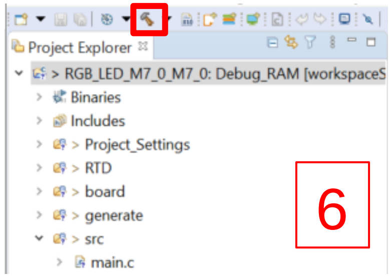
□ Section: -j .text

□ Section: -j .data

Other sections (-j)

Apply and Close | Cancel

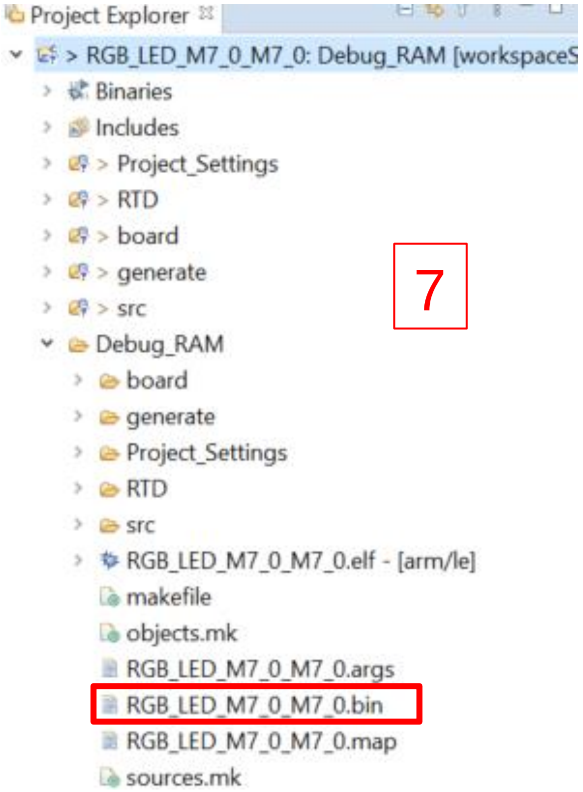
- Re-build the project



Project Explorer

- > RGB_LED_M7_0_M7_0: Debug_RAM [workspaceS
  - Binaries
  - Includes
  - > Project_Settings
  - > RTD
  - > board
  - > generate
  - > src
    - main.c

- The .bin file will be generated.

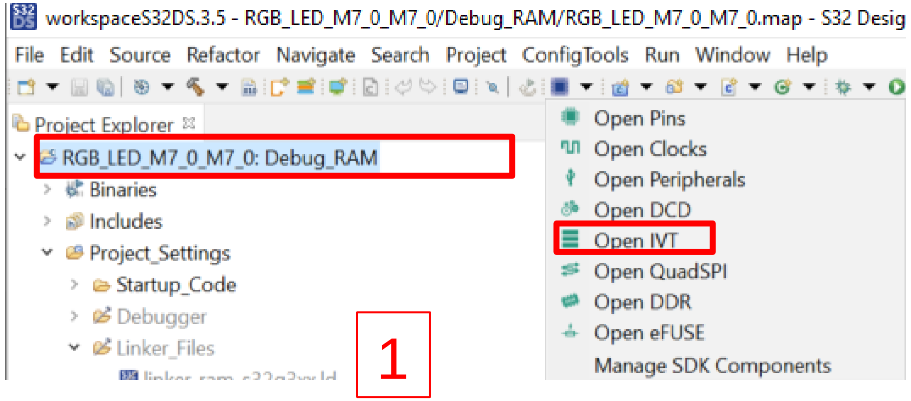


Project Explorer

- > RGB_LED_M7_0_M7_0: Debug_RAM [workspaceS
  - Binaries
  - Includes
  - > Project_Settings
  - > RTD
  - > board
  - > generate
  - > src
  - Debug_RAM
    - board
    - generate
    - Project_Settings
    - RTD
    - src
    - RGB_LED_M7_0_M7_0.elf - [arm/le]
    - makefile
    - objects.mk
    - RGB_LED_M7_0_M7_0.args
    - RGB_LED_M7_0_M7_0.bin
    - RGB_LED_M7_0_M7_0.map
    - sources.mk

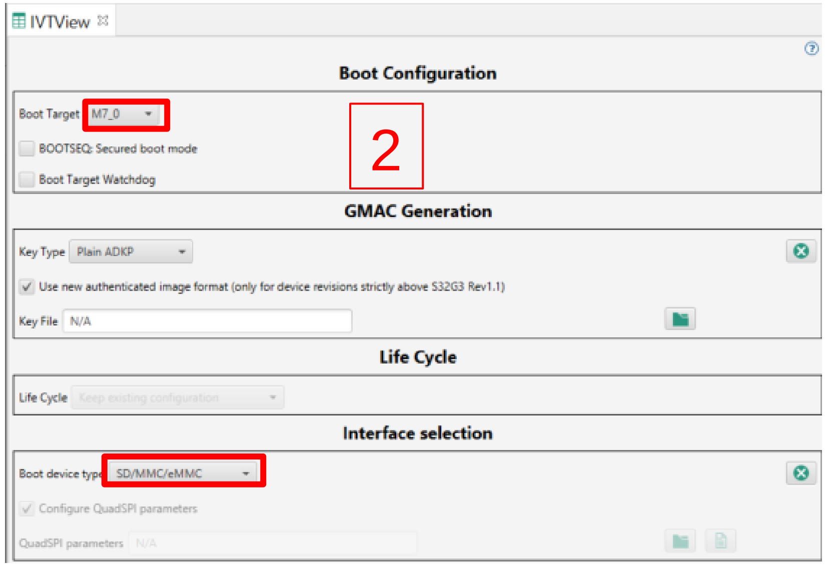
LIGHT UP RGB LED: MAKE IMAGE BY IVT TOOL

- Select the created project and open IVT tool



workspaceS32DS.3.5 - RGB_LED_M7_0_M7_0/Debug_RAM/RGB_LED_M7_0_M7_0.map - S32 Desig

- Select M7_0 as Boot Target core and select SD/MMC/eMMC as Boot device type



IVTView

Boot Configuration

Boot Target: M7_0

□ BOOTSEQ: Secured boot mode

□ Boot Target Watchdog

GMAC Generation

Key Type: Plain ADKP

☑ Use new authenticated image format (only for device revisions strictly above S32G3 Rev1.1)

Key File: N/A

Life Cycle

Life Cycle: Keep existing configuration

Interface selection

Boot device type: SD/MMC/eMMC

☑ Configure QuadSPI parameters

QuadSPI parameters: N/A

- Uncheck Self-Test DCD, DCD and HSE to be reserved

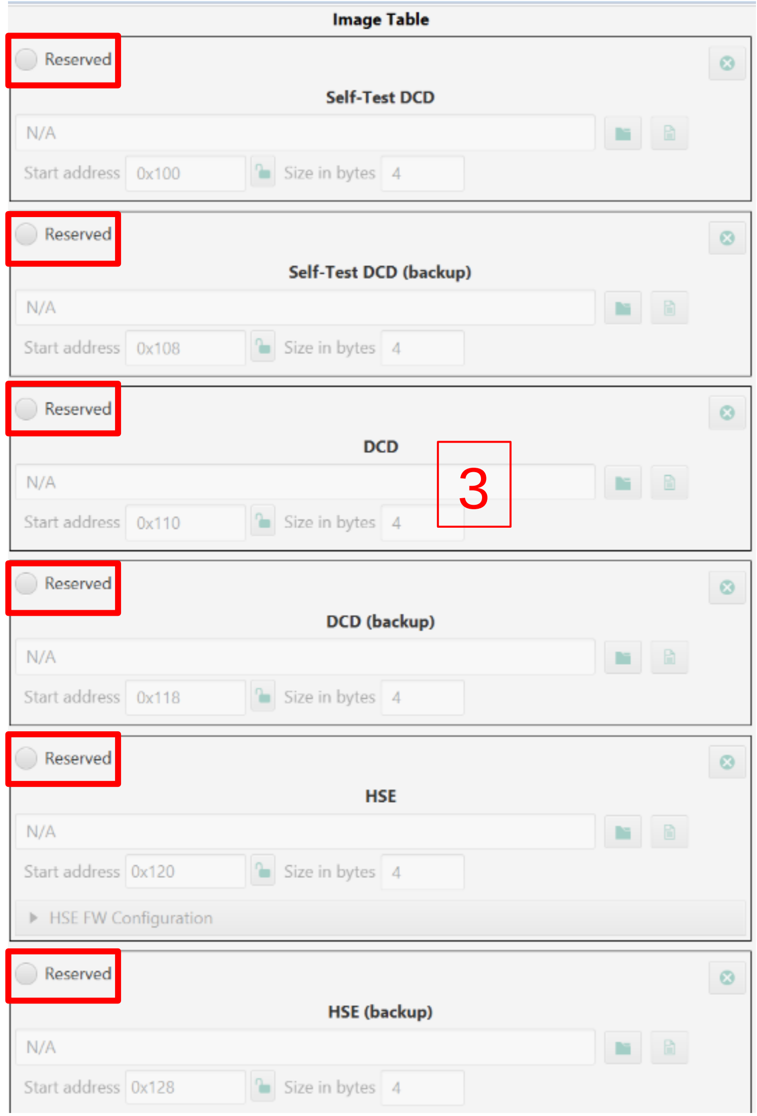


Image Table

○ Reserved

Self-Test DCD

N/A

Start address: 0x100 Size in bytes: 4

○ Reserved

Self-Test DCD (backup)

N/A

Start address: 0x108 Size in bytes: 4

○ Reserved

DCD

N/A

Start address: 0x110 Size in bytes: 4

○ Reserved

DCD (backup)

N/A

Start address: 0x118 Size in bytes: 4

○ Reserved

HSE

N/A

Start address: 0x120 Size in bytes: 4

HSE FW Configuration

○ Reserved

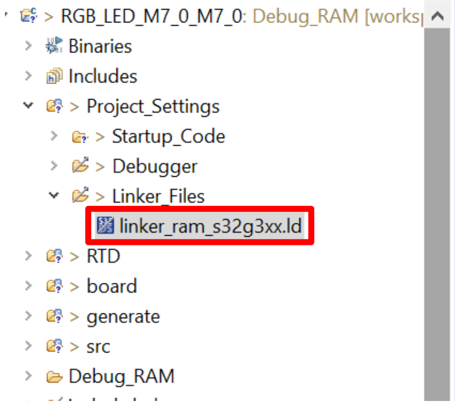
HSE (backup)

N/A

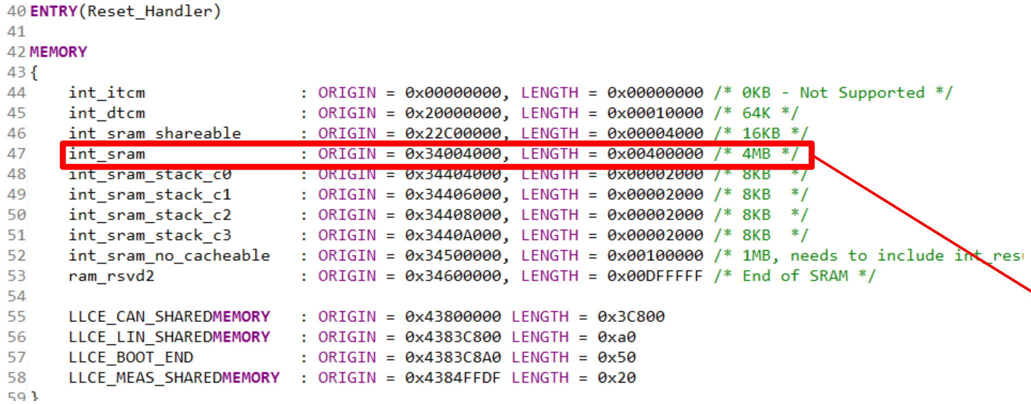
Start address: 0x128 Size in bytes: 4

LIGHT UP RGB LED: MAKE IMAGE BY IVT TOOL

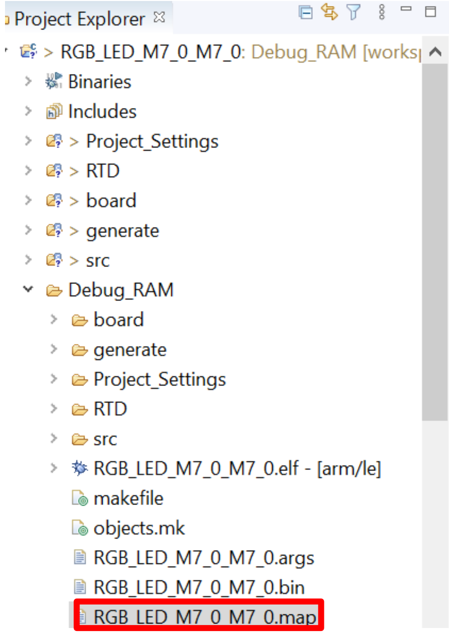
- Configure Application Boot Image according to .ld file and .map file



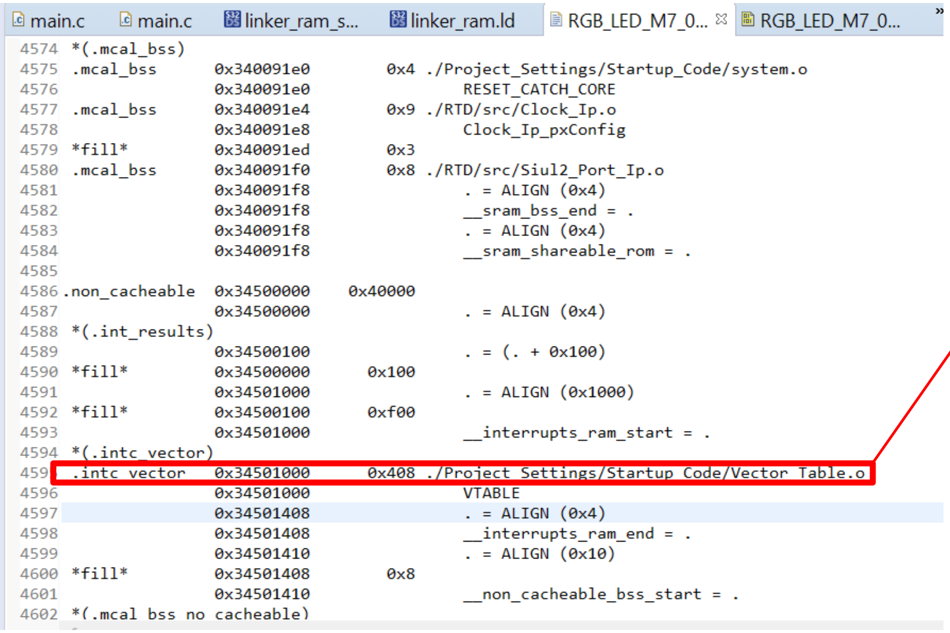
```
RGB_LED_M7_0_M7_0: Debug_RAM [works
  Binaries
  Includes
  Project_Settings
    Startup_Code
    Debugger
    Linker_Files
      linker_ram_s32g3xx.ld
  RTD
  board
  generate
  src
  Debug_RAM
```



```
ENTRY(Reset_Handler)

MEMORY
{
    int_itcm                : ORIGIN = 0x00000000, LENGTH = 0x00000000 /* 0KB - Not Supported */
    int_dtcm                : ORIGIN = 0x20000000, LENGTH = 0x00010000 /* 64K */
    int_sram_shareable      : ORIGIN = 0x22C00000, LENGTH = 0x00004000 /* 16KB */
    int_sram                : ORIGIN = 0x34004000, LENGTH = 0x00400000 /* 4MB */
    int_sram_stack_c0       : ORIGIN = 0x34404000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_stack_c1       : ORIGIN = 0x34406000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_stack_c2       : ORIGIN = 0x34408000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_stack_c3       : ORIGIN = 0x3440A000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_no_cacheable   : ORIGIN = 0x34500000, LENGTH = 0x00100000 /* 1MB, needs to include int_resu
    ram_rsvd2               : ORIGIN = 0x34600000, LENGTH = 0x00DFFFFF /* End of SRAM */

    LLCE_CAN_SHAREDMEMORY   : ORIGIN = 0x43800000 LENGTH = 0x3C800
    LLCE_LIN_SHAREDMEMORY   : ORIGIN = 0x4383C800 LENGTH = 0xa0
    LLCE_BOOT_END           : ORIGIN = 0x4383C8A0 LENGTH = 0x50
    LLCE_MEAS_SHAREDMEMORY  : ORIGIN = 0x4384FFDF LENGTH = 0x20
}
```

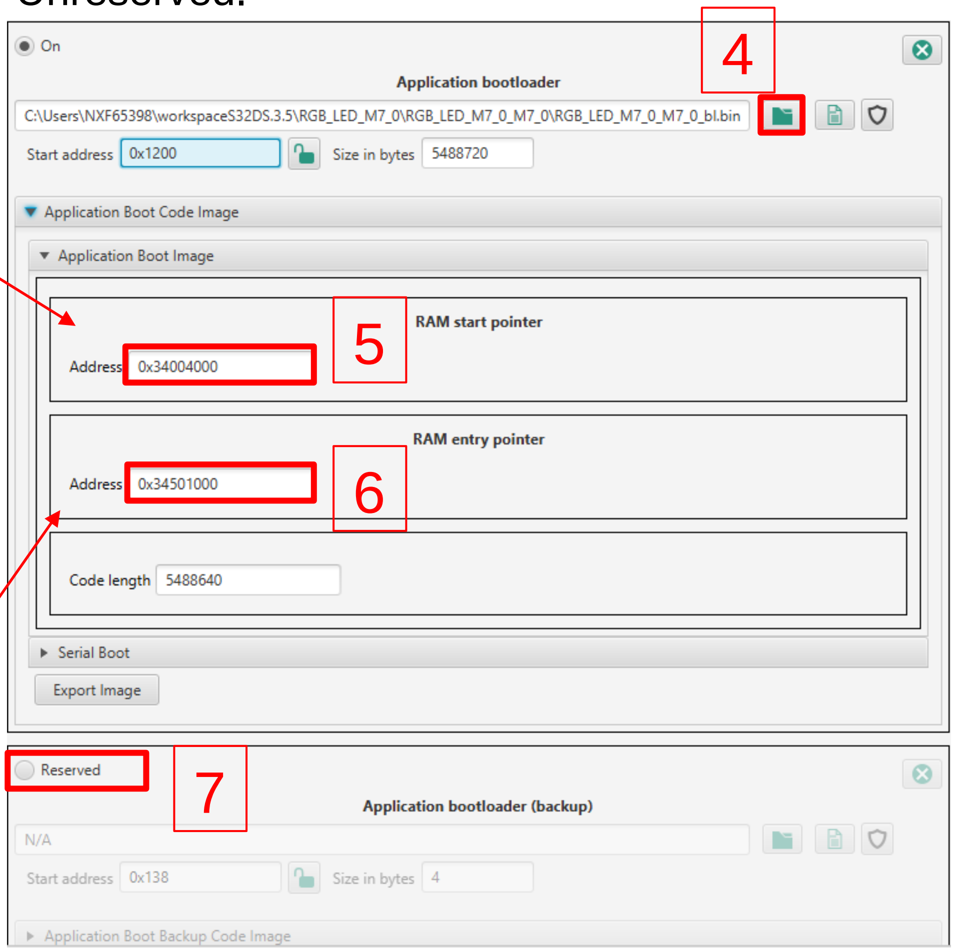


```
Project Explorer
RGB_LED_M7_0_M7_0: Debug_RAM [works
  Binaries
  Includes
  Project_Settings
  RTD
  board
  generate
  src
  Debug_RAM
    board
    generate
    Project_Settings
    RTD
    src
    RGB_LED_M7_0_M7_0.elf - [arm/le]
    makefile
    objects.mk
    RGB_LED_M7_0_M7_0.args
    RGB_LED_M7_0_M7_0.bin
    RGB_LED_M7_0_M7_0.map
```



```
*(.mcal_bss)
.mcal_bss       0x340091e0       0x4 ./Project_Settings/Startup_Code/system.o
                0x340091e0               RESET_CATCH_CORE
.mcal_bss       0x340091e4       0x9 ./RTD/src/Clock_Ip.o
                0x340091e8               Clock_Ip_pxConfig
*fill*          0x340091ed       0x3
.mcal_bss       0x340091f0       0x8 ./RTD/src/Siul2_Port_Ip.o
                0x340091f8               . = ALIGN (0x4)
                0x340091f8               __sram_bss_end = .
                0x340091f8               . = ALIGN (0x4)
                0x340091f8               __sram_shareable_rom = .

.non_cacheable  0x34500000     0x40000
                0x34500000               . = ALIGN (0x4)
*(.int_results)
                0x34500100               . = (. + 0x100)
*fill*          0x34500000     0x100
                0x34501000               . = ALIGN (0x1000)
*fill*          0x34500100     0xf00
                0x34501000               __interrupts_ram_start = .
*(.intc_vector)
.intc_vector    0x34501000     0x408 ./Project_Settings/Startup_Code/Vector_Table.o
                0x34501000               VTABLE
                0x34501408               . = ALIGN (0x4)
                0x34501408               __interrupts_ram_end = .
                0x34501410               . = ALIGN (0x10)
*fill*          0x34501408       0x8
                0x34501410               __non_cacheable_bss_start = .
*(.mcal_bss_no_cacheable)
```

Click browse to select .bin files generated from Page 26 and set Application bootloader (backup) Unreserved.



● On

Application bootloader

C:\Users\NXF65398\workspaceS32DS.3.5\RGB_LED_M7_0\RGB_LED_M7_0_M7_0\RGB_LED_M7_0_M7_0_bl.bin

Start address: 0x1200 — Size in bytes: 5488720

▼ Application Boot Code Image

▼ Application Boot Image

RAM start pointer — Address: 0x34004000

RAM entry pointer — Address: 0x34501000

Code length: 5488640

○ Reserved

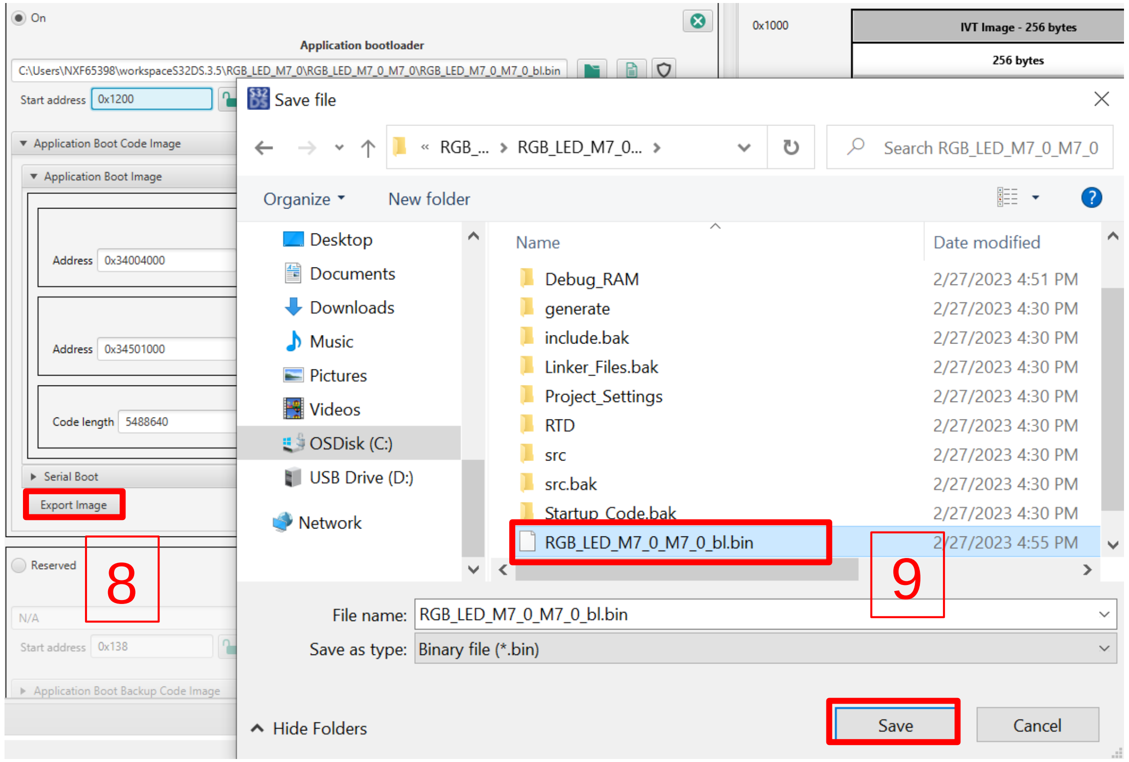
Application bootloader (backup)

N/A

Start address: 0x138 — Size in bytes: 4

LIGHT UP RGB LED: MAKE IMAGE BY IVT TOOL

- Export and save image



Application bootloader

C:\Users\NXF65398\workspaceS32DS.3.5\RGB_LED_M7_0\RGB_LED_M7_0_M7_0\RGB_LED_M7_0_M7_0_bl.bin

Start address 0x1200

Address 0x34004000

Address 0x34501000

Code length 5488640

Start address 0x138

0x1000

IVT Image - 256 bytes

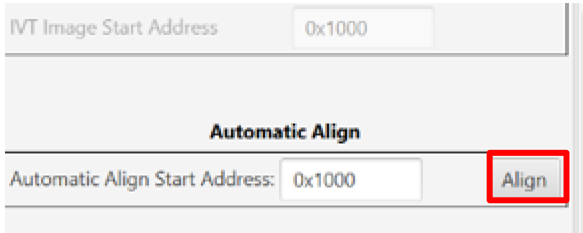
256 bytes

File name: RGB_LED_M7_0_M7_0_bl.bin

Save as type: Binary file (*.bin)

Name	Date modified
Debug_RAM	2/27/2023 4:51 PM
generate	2/27/2023 4:30 PM
include.bak	2/27/2023 4:30 PM
Linker_Files.bak	2/27/2023 4:30 PM
Project_Settings	2/27/2023 4:30 PM
RTD	2/27/2023 4:30 PM
src	2/27/2023 4:30 PM
src.bak	2/27/2023 4:30 PM
Startup_Code.bak	2/27/2023 4:30 PM
RGB_LED_M7_0_M7_0_bl.bin	2/27/2023 4:55 PM

- Click on “Align” button to align address.

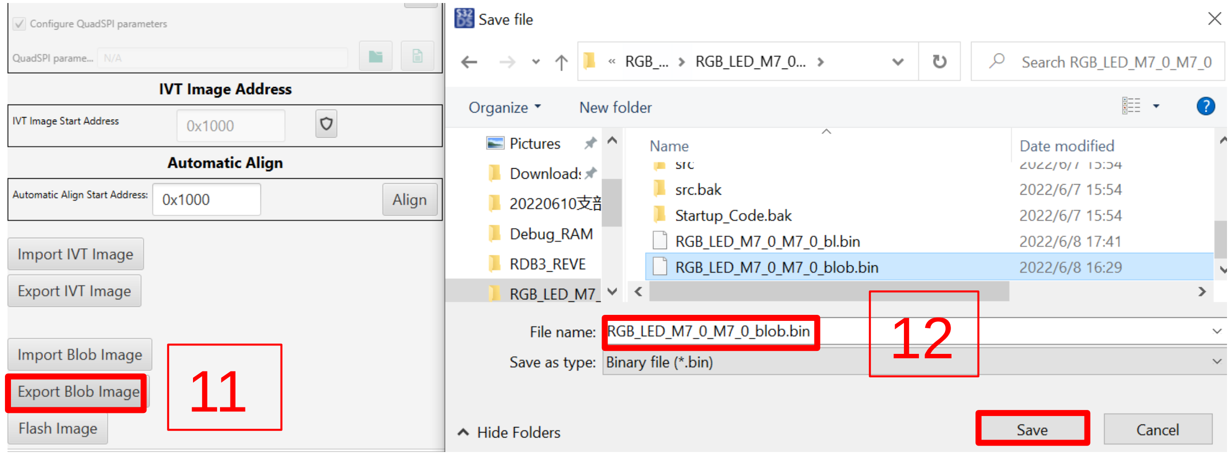


IVT Image Start Address 0x1000

Automatic Align

Automatic Align Start Address: 0x1000

- Click on “Export Blob Image” to generate blob image and save final blob image.



Configure QuadSPI parameters

QuadSPI parame... N/A

IVT Image Address

IVT Image Start Address 0x1000

Automatic Align

Automatic Align Start Address: 0x1000

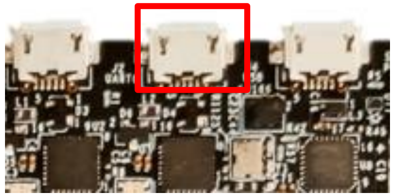
File name: RGB_LED_M7_0_M7_0_blob.bin

Save as type: Binary file (*.bin)

Name	Date modified
src	2022/6/7 15:54
src.bak	2022/6/7 15:54
Startup_Code.bak	2022/6/7 15:54
RGB_LED_M7_0_M7_0_bl.bin	2022/6/8 17:41
RGB_LED_M7_0_M7_0_blob.bin	2022/6/8 16:29

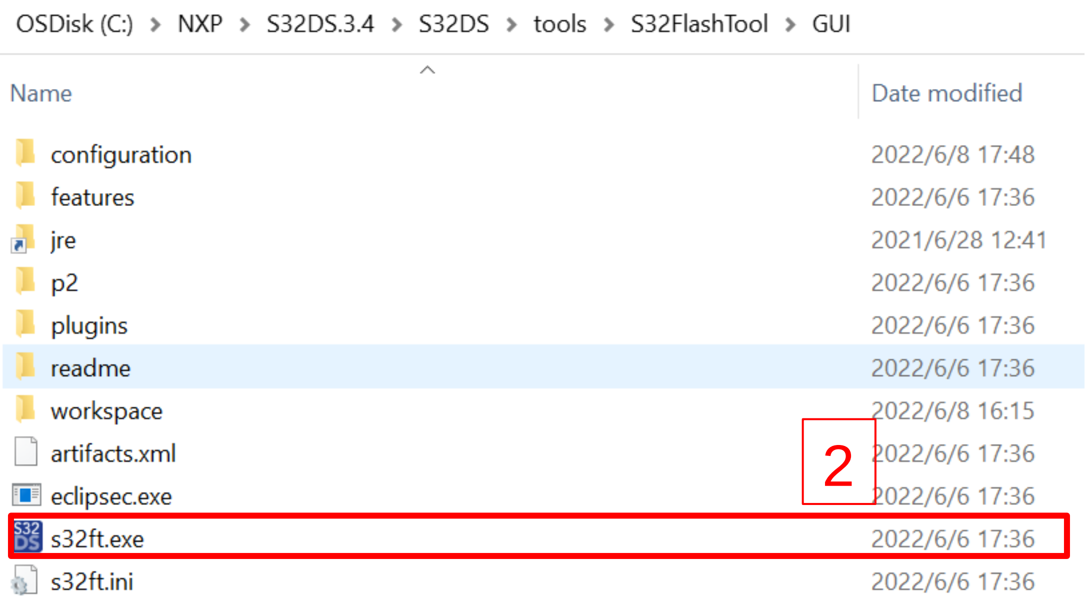
LIGHT UP RGB LED: DOWNLOAD IMAGE INTO SD CARD

- Connect the UART0 port of S32G-VNP-GLDBOX3 with PC.



1

- Open S32FlashTool

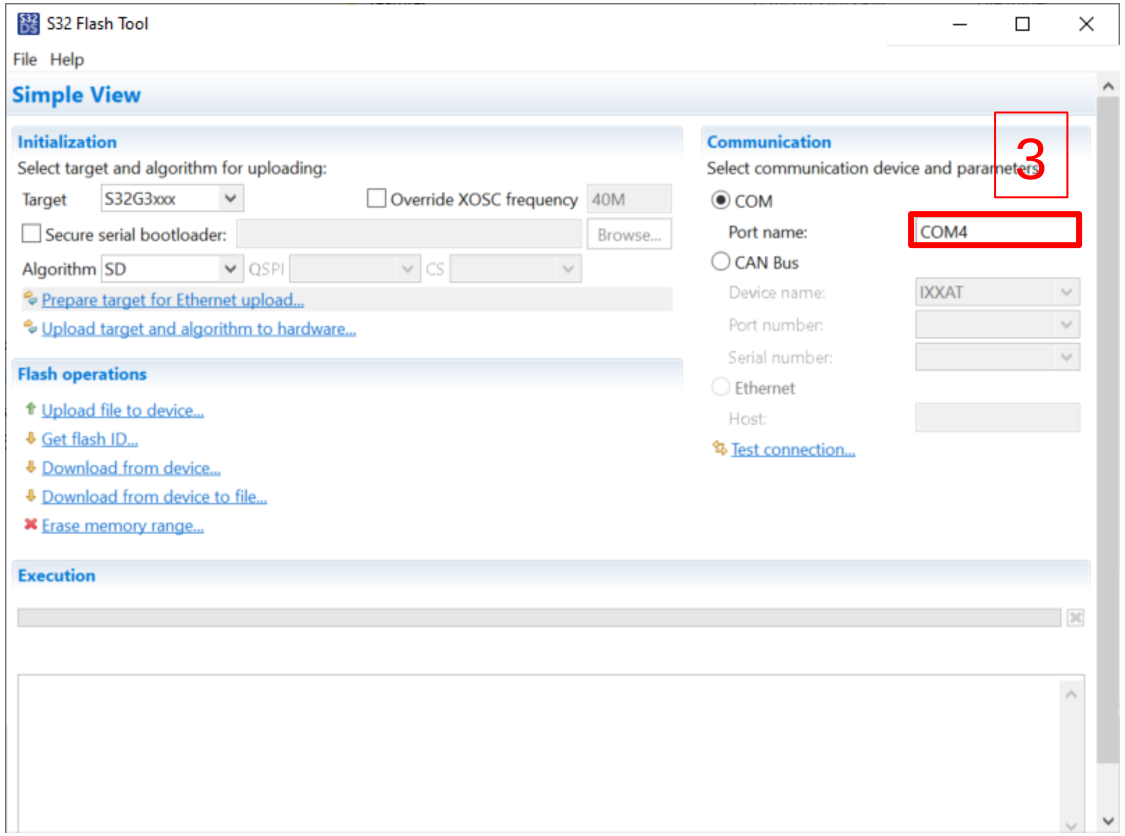


OSDisk (C:) › NXP › S32DS.3.4 › S32DS › tools › S32FlashTool › GUI

Name	Date modified
configuration	2022/6/8 17:48
features	2022/6/6 17:36
jre	2021/6/28 12:41
p2	2022/6/6 17:36
plugins	2022/6/6 17:36
readme	2022/6/6 17:36
workspace	2022/6/8 16:15
artifacts.xml	2022/6/6 17:36
eclipsec.exe	2022/6/6 17:36
s32ft.exe	2022/6/6 17:36
s32ft.ini	2022/6/6 17:36

2

- Set the port name of COM interface according to the actual condition of local PC.



S32 Flash Tool

Simple View

Initialization

Select target and algorithm for uploading:

Target: S32G3xxx

□ Override XOSC frequency: 40M

□ Secure serial bootloader:

Algorithm: SD

QSPI

CS

Prepare target for Ethernet upload...

Upload target and algorithm to hardware...

Flash operations

Upload file to device...

Get flash ID...

Download from device...

Download from device to file...

Erase memory range...

Communication

Select communication device and parameters

● COM

Port name: COM4

○ CAN Bus

Device name: IXXAT

Port number:

Serial number:

○ Ethernet

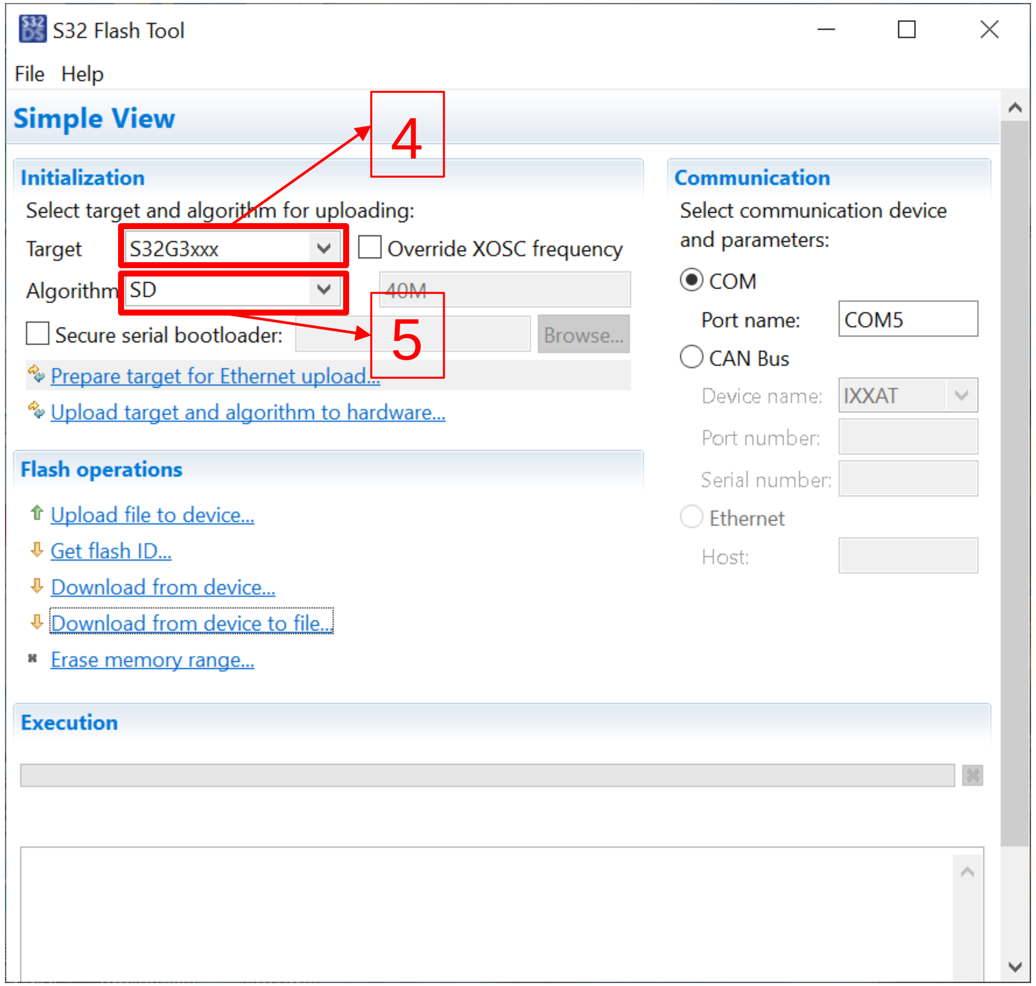
Host:

Test connection...

Execution

LIGHT UP RGB LED: DOWNLOAD IMAGE INTO SD CARD

- Select S32G3xxx as the target image and set SD as Algorithm



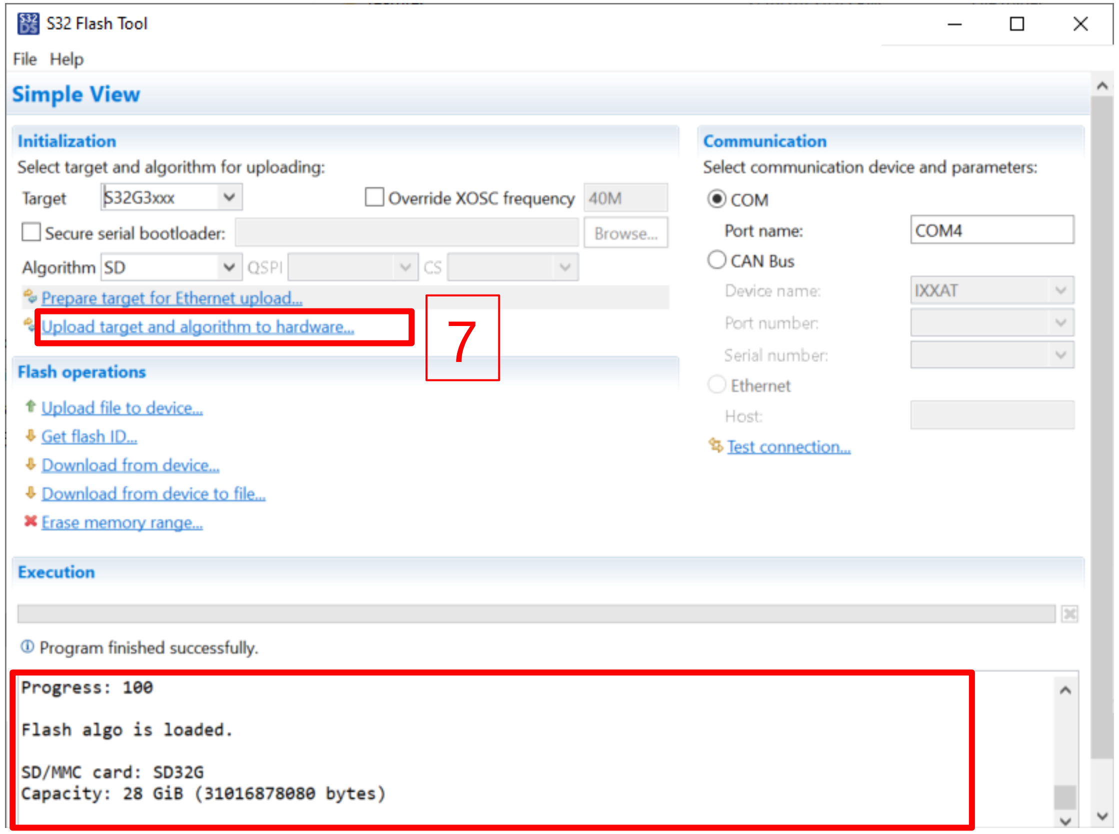
6

- Insert SD Card into the slot of S32G-VNP-GLDBOX3, set SW3 on and set SW9,10 to select serial boot mode (Refer to APPENDIX A)

Part Reference Number	Setting
SW9	1-OFF, 2-OFF
SW10	1-OFF, 2-OFF
SW3	ON

LIGHT UP RGB LED: DOWNLOAD IMAGE INTO SD CARD

- Click on ”Upload target and algorithm to hardware…” button and the log should be shown as below.



Select target and algorithm for uploading:

Target: S32G3xxx

□ Override XOSC frequency: 40M

□ Secure serial bootloader:

Algorithm: SD QSPI CS

Select communication device and parameters:

● COM

Port name: COM4

○ CAN Bus

Device name: IXXAT

Port number:

Serial number:

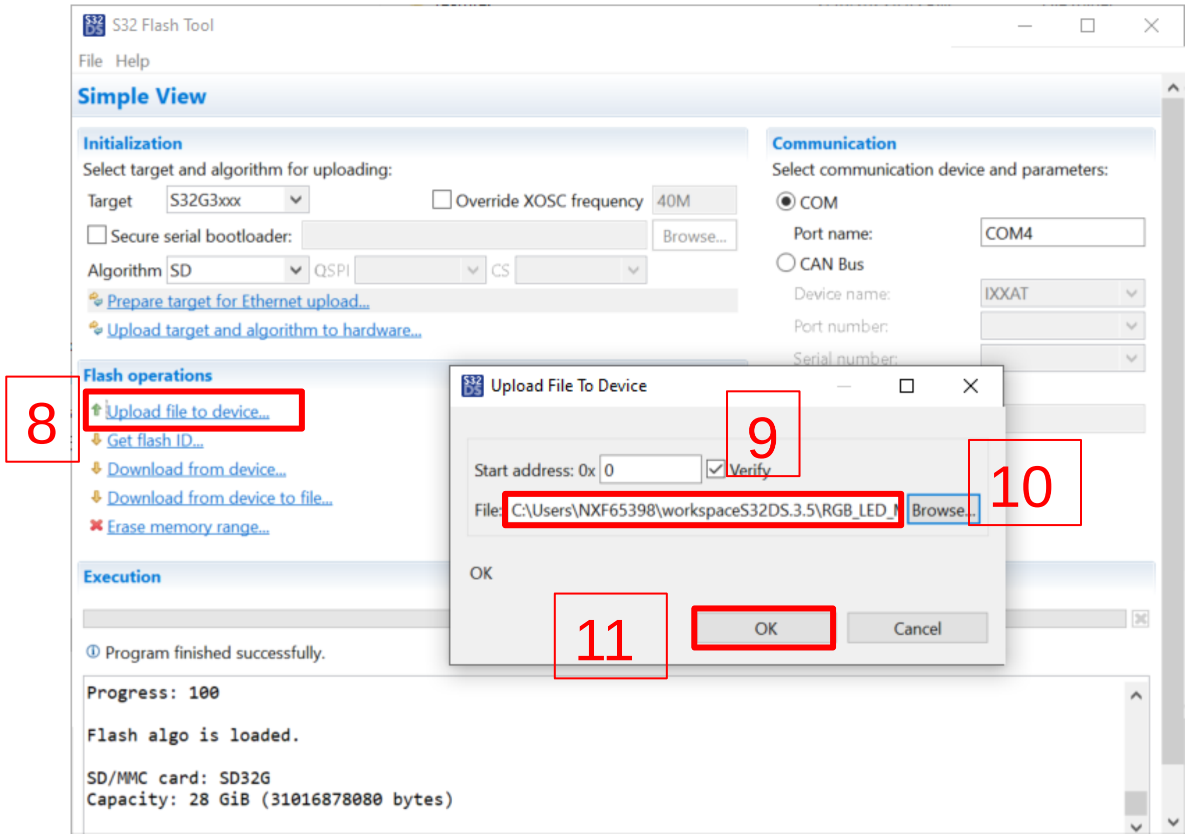
○ Ethernet

Host:

ⓘ Program finished successfully.

```
Progress: 100

Flash algo is loaded.

SD/MMC card: SD32G
Capacity: 28 GiB (31016878080 bytes)
```

- Click ”Upload file to device…” button, set start address as 0x00 and select the generated blob image from page 30, click on ”OK” button.



Select target and algorithm for uploading:

Target: S32G3xxx

□ Override XOSC frequency: 40M

□ Secure serial bootloader:

Algorithm: SD QSPI CS

Select communication device and parameters:

● COM

Port name: COM4

○ CAN Bus

Device name: IXXAT

Port number:

Serial number:

Start address: 0x 0 ☑ Verify

File: C:\Users\NXF65398\workspaceS32DS.3.5\RGB_LED_I

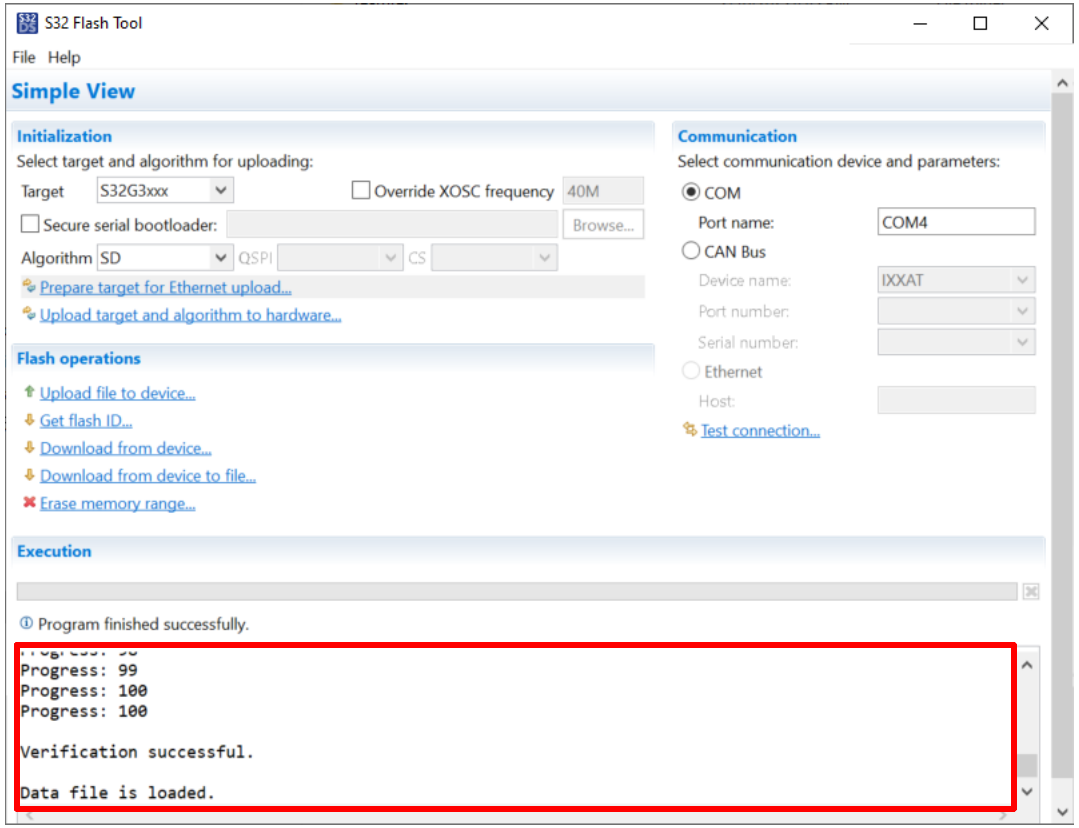
OK

ⓘ Program finished successfully.

```
Progress: 100

Flash algo is loaded.

SD/MMC card: SD32G
Capacity: 28 GiB (31016878080 bytes)
```

LIGHT UP RGB LED: DOWNLOAD IMAGE INTO SD CARD

- If the image downloading process is successful, the result log should be the similar as below.



Target S32G3xxx

□ Override XOSC frequency 40M

□ Secure serial bootloader:

Algorithm SD QSPI CS

● COM

Port name: COM4

○ CAN Bus

Device name: IXXAT

Port number:

Serial number:

○ Ethernet

Host:

Program finished successfully.

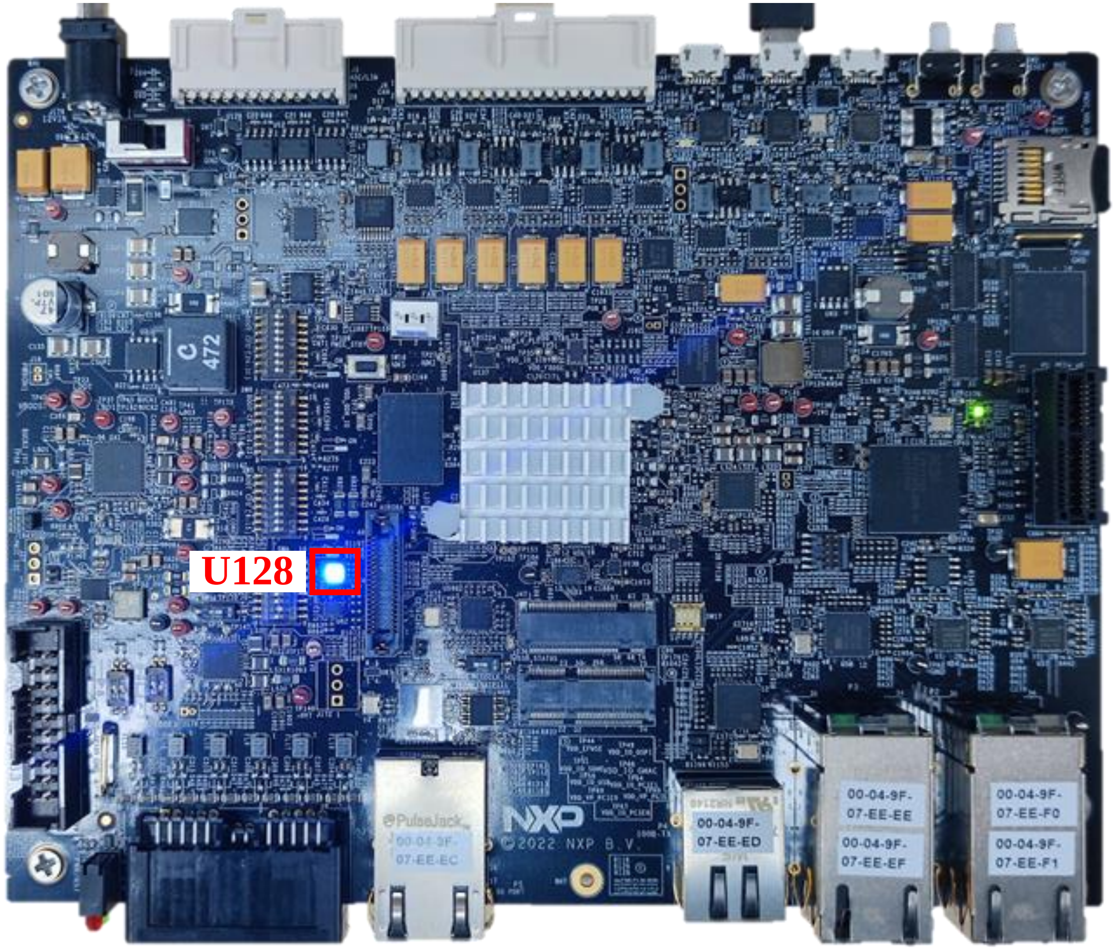
```
Progress: 99
Progress: 100
Progress: 100

Verification successful.

Data file is loaded.
```

LIGHT UP RGB LED: SET RDB2 AND RUN APPLICATION

- Set SW3, 4, 9,10 to select SD card boot mode and set SW11 into “ON” to connect RGB LED with S32G3 pin(Refer to APPENDIX B)

Part Reference Number	Setting
SW3	ON
SW4	7-ON, Other-OFF
SW9	1-OFF, 2-OFF
SW10	1-ON, 2-OFF
SW11	ON

- Power on GLDBOX3, the RGB LED (U128) will be lighted in blue color.



Enable Linux BSP On Cortex-A53 Cores

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

STEP 1: INSTALL DEPENDENCES FOR DEFAULT YOCOTO

- The steps below have been validated on Ubuntu-18.04 LTS.

➢ Update the package manager

```
sudo apt-get update
```

➢ Install python 2x – 2.6 or newer

```
sudo apt-get install python
```

➢ Install git 1.8.3 or newer

```
sudo apt-get install git
```

➢ Install curl

```
sudo apt-get install curl
```

➢ Install repo

```
mkdir ~/bin
curl http://commondatastorage.googleapis.com
/git-repo-downloads/repo > ~/bin/repo
chmod a+x ~/bin/repo
PATH=${PATH}:~/bin
```

➢ Configure your git environment (you may skip this option if you have git already configured):

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

STEP 2: DOWNLOAD YOCTO PROJECT ENVIRONMENT

- ➢ Create a dedicated directory for BSP

```
mkdir fsl-auto-yocto-bsp

cd fsl-auto-yocto-bsp
```

- ➢ Choose the BSP branch and initialize the Yocto project for BSP

```
repo init -u https://github.com/nxp-auto-linux/auto_yocto_bsp -b release/bsp35.0

repo sync
```

STEP 3: BUILD LINUX BSP IMAGE

- Prepare and confirm the building environment for the BSP

  (*Enter the directory fsl-auto-yocto-bsp*)

  ```
  ./sources/meta-alb/scripts/host-prepare.sh
  ```

- Create build directory for target platform board

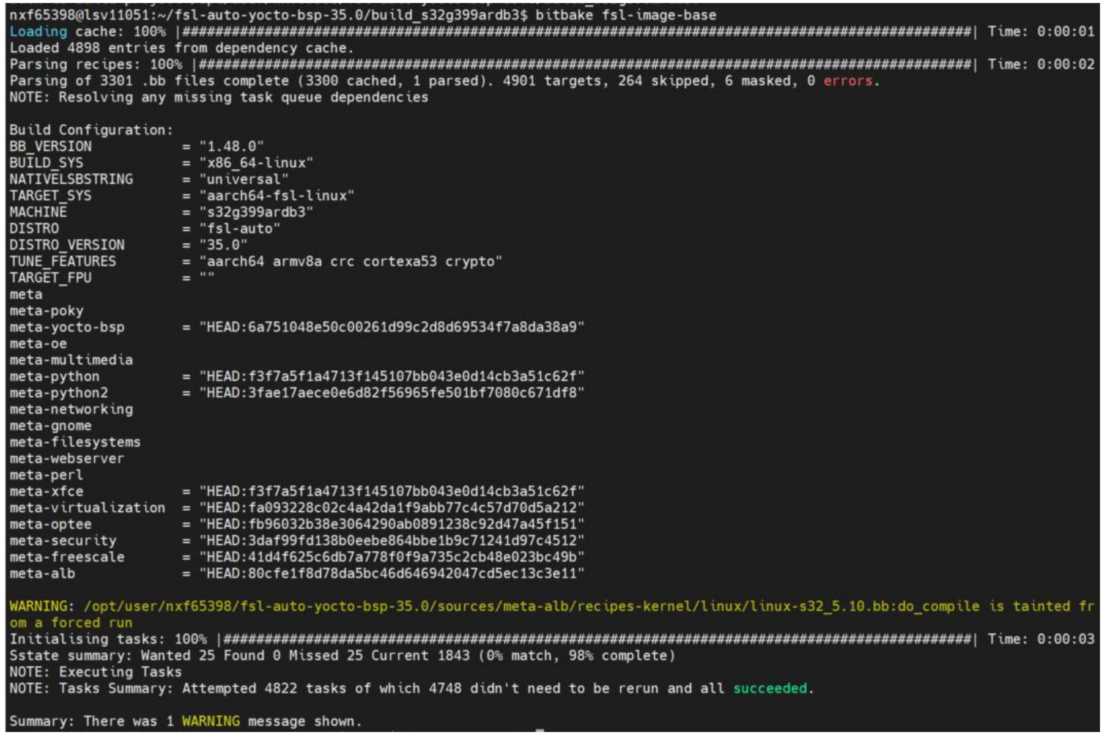
  ```
  source nxp-setup-alb.sh -m s32g399ardb3
  ```

- Build BSP base Image

  ```
  bitbake fsl-image-base
  ```

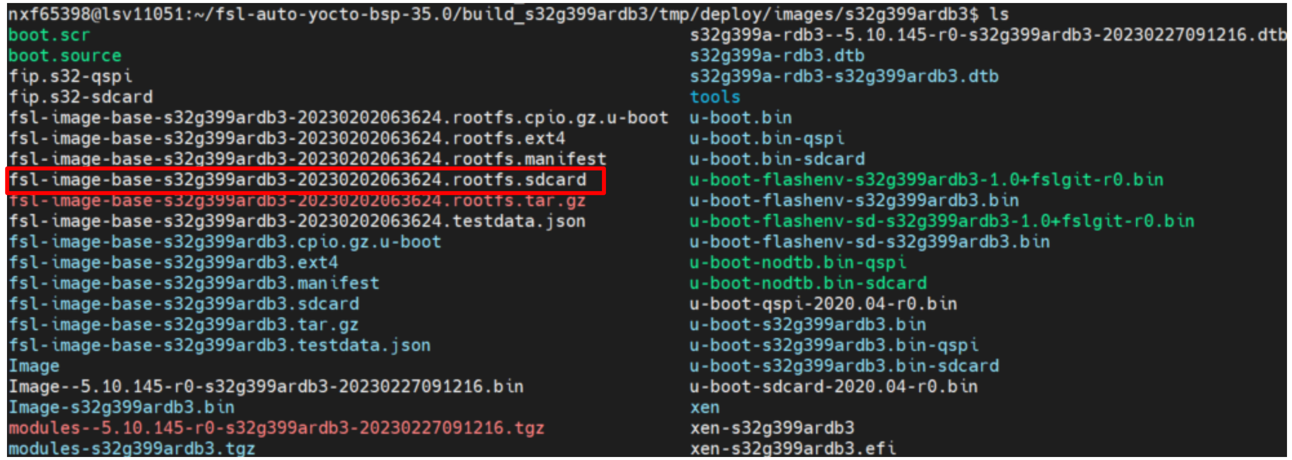
STEP 3: BUILD LINUX BSP IMAGE

➢ Waiting for the building process finish, the BSP image will be generated like below.



```
nxf65398@lsv11051:~/fsl-auto-yocto-bsp-35.0/build_s32g399ardb3$ bitbake fsl-image-base
Loading cache: 100% |###############################################################################################| Time: 0:00:01
Loaded 4898 entries from dependency cache.
Parsing recipes: 100% |#############################################################################################| Time: 0:00:02
Parsing of 3301 .bb files complete (3300 cached, 1 parsed). 4901 targets, 264 skipped, 6 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.48.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "aarch64-fsl-linux"
MACHINE              = "s32g399ardb3"
DISTRO               = "fsl-auto"
DISTRO_VERSION       = "35.0"
TUNE_FEATURES        = "aarch64 armv8a crc cortexa53 crypto"
TARGET_FPU           = ""
meta
meta-poky
meta-yocto-bsp       = "HEAD:6a751048e50c00261d99c2d8d69534f7a8da38a9"
meta-oe
meta-multimedia
meta-python          = "HEAD:f3f7a5f1a4713f145107bb043e0d14cb3a51c62f"
meta-python2         = "HEAD:3fae17aece0e6d82f56965fe501bf7080c671df8"
meta-networking
meta-gnome
meta-filesystems
meta-webserver
meta-perl
meta-xfce            = "HEAD:f3f7a5f1a4713f145107bb043e0d14cb3a51c62f"
meta-virtualization  = "HEAD:fa093228c02c4a42da1f9abb77c4c57d70d5a212"
meta-optee           = "HEAD:fb96032b38e3064290ab0891238c92d47a45f151"
meta-security        = "HEAD:3daf99fd138b0eebe864bbe1b9c71241d97c4512"
meta-freescale       = "HEAD:41d4f625c6db7a778f0f9a735c2cb48e023bc49b"
meta-alb             = "HEAD:80cfe1f8d78da5bc46d646942047cd5ec13c3e11"

WARNING: /opt/user/nxf65398/fsl-auto-yocto-bsp-35.0/sources/meta-alb/recipes-kernel/linux/linux-s32_5.10.bb:do_compile is tainted fr
om a forced run
Initialising tasks: 100% |##########################################################################################| Time: 0:00:03
Sstate summary: Wanted 25 Found 0 Missed 25 Current 1843 (0% match, 98% complete)
NOTE: Executing Tasks
NOTE: Tasks Summary: Attempted 4822 tasks of which 4748 didn't need to be rerun and all succeeded.

Summary: There was 1 WARNING message shown.
```

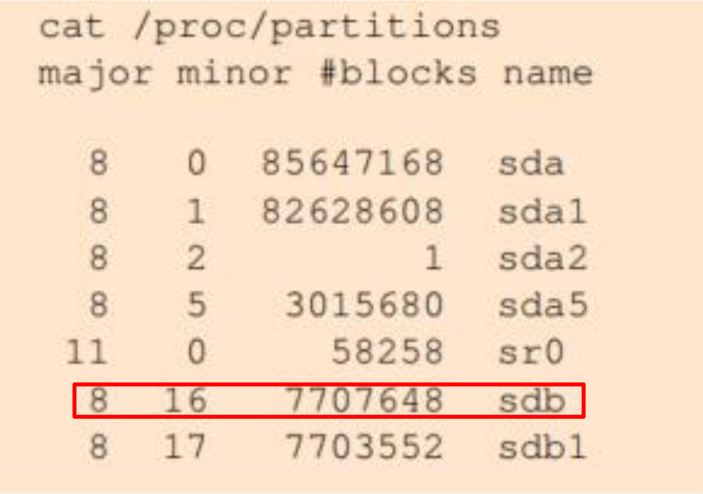
➢ The generated BSP image(fsl-image-s32g399ardb3-xxx.rootfs.sdcard) is located at ” *<builddirectory>/tmp/deploy/images/s32g399ardb3/* ”.



```
nxf65398@lsv11051:~/fsl-auto-yocto-bsp-35.0/build_s32g399ardb3/tmp/deploy/images/s32g399ardb3$ ls
boot.scr                                                        s32g399a-rdb3--5.10.145-r0-s32g399ardb3-20230227091216.dtb
boot.source                                                     s32g399a-rdb3.dtb
fip.s32-qspi                                                    s32g399a-rdb3-s32g399ardb3.dtb
fip.s32-sdcard                                                  tools
fsl-image-base-s32g399ardb3-20230202063624.rootfs.cpio.gz.u-boot  u-boot.bin
fsl-image-base-s32g399ardb3-20230202063624.rootfs.ext4          u-boot.bin-qspi
fsl-image-base-s32g399ardb3-20230202063624.rootfs.manifest      u-boot.bin-sdcard
fsl-image-base-s32g399ardb3-20230202063624.rootfs.sdcard        u-boot-flashenv-s32g399ardb3-1.0+fslgit-r0.bin
fsl-image-base-s32g399ardb3-20230202063624.rootfs.tar.gz        u-boot-flashenv-s32g399ardb3.bin
fsl-image-base-s32g399ardb3-20230202063624.testdata.json        u-boot-flashenv-sd-s32g399ardb3-1.0+fslgit-r0.bin
fsl-image-base-s32g399ardb3.cpio.gz.u-boot                      u-boot-flashenv-sd-s32g399ardb3.bin
fsl-image-base-s32g399ardb3.ext4                                u-boot-nodtb.bin-qspi
fsl-image-base-s32g399ardb3.manifest                            u-boot-nodtb.bin-sdcard
fsl-image-base-s32g399ardb3.sdcard                              u-boot-qspi-2020.04-r0.bin
fsl-image-base-s32g399ardb3.tar.gz                              u-boot-s32g399ardb3.bin
fsl-image-base-s32g399ardb3.testdata.json                       u-boot-s32g399ardb3.bin-qspi
Image                                                           u-boot-s32g399ardb3.bin-sdcard
Image--5.10.145-r0-s32g399ardb3-20230227091216.bin              u-boot-sdcard-2020.04-r0.bin
Image-s32g399ardb3.bin                                          xen
modules--5.10.145-r0-s32g399ardb3-20230227091216.tgz            xen-s32g399ardb3
modules-s32g399ardb3.tgz                                        xen-s32g399ardb3.efi
```

STEP 4: DOWNLOAD BSP IMAGE INTO SD CARD: IN LINUX

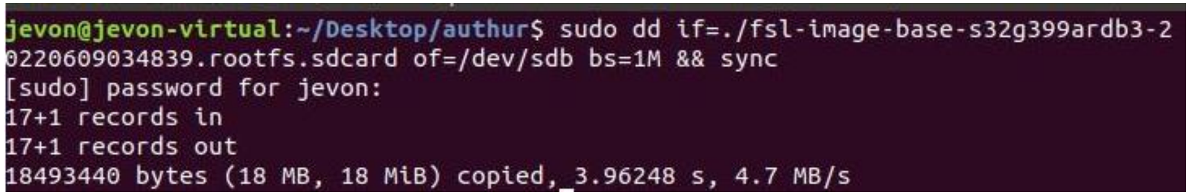
- Insert SD/MMC card reader into PC and identify the device node assigned to the SD/MMC card. It is assumed that the device assigned is /dev/sdb.



```
cat /proc/partitions
major minor #blocks name

   8     0 85647168   sda
   8     1 82628608   sda1
   8     2        1   sda2
   8     5  3015680   sda5
  11     0    58258   sr0
   8    16  7707648   sdb
   8    17  7703552   sdb1
```

- Copy the generated image from page 39 to SD card device using dd command like below:

sudo dd if=./ fsl-image-base-s32g399ardb3-xxx.rootfs.sdcard of=/dev/sdb bs=1M && sync



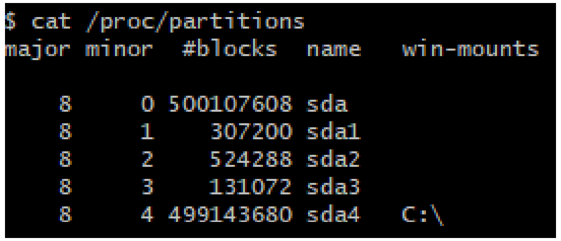
```
jevon@jevon-virtual:~/Desktop/authur$ sudo dd if=./fsl-image-base-s32g399ardb3-2
0220609034839.rootfs.sdcard of=/dev/sdb bs=1M && sync
[sudo] password for jevon:
17+1 records in
17+1 records out
18493440 bytes (18 MB, 18 MiB) copied, 3.96248 s, 4.7 MB/s
```

Note: The steps in this page are only supported in Linux environment, if the user want to download the BSP image into SD card on Windows, please follow the steps on the next page.

STEP 4: DOWNLOAD BSP IMAGE INTO SD CARD: IN WINDOWS

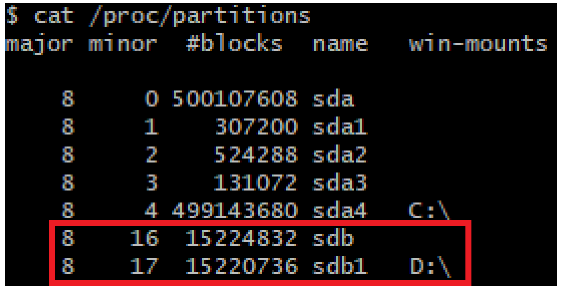
1. Install and Run Cygwin as administrator

2. Before inserting SD card into the slot, run “cat /proc/partitions” cmd and note the current devices.



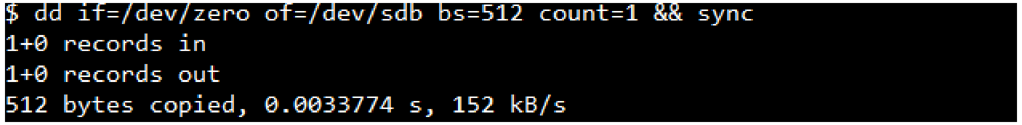
```
$ cat /proc/partitions
major minor  #blocks  name   win-mounts

   8     0 500107608 sda
   8     1    307200 sda1
   8     2    524288 sda2
   8     3    131072 sda3
   8     4 499143680 sda4   C:\
```

3. After inserting SD card into the slot, run cat /proc/partitions again and find out the SD card descriptor



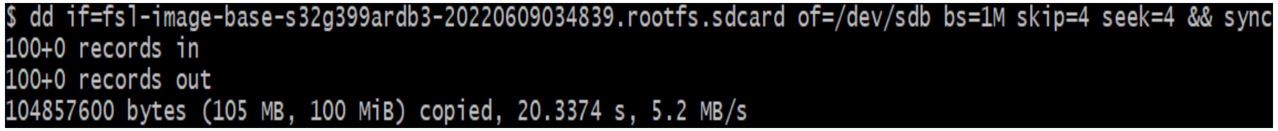
```
$ cat /proc/partitions
major minor  #blocks  name   win-mounts

   8     0 500107608 sda
   8     1    307200 sda1
   8     2    524288 sda2
   8     3    131072 sda3
   8     4 499143680 sda4   C:\
   8    16  15224832 sdb
   8    17  15220736 sdb1   D:\
```

4. Erase the sub-partition info on the SD card
   dd if=/dev/zero of=/dev/sdb bs=512 count=1 && sync



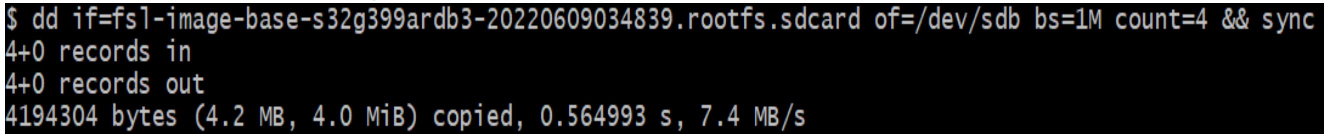
```
$ dd if=/dev/zero of=/dev/sdb bs=512 count=1 && sync
1+0 records in
1+0 records out
512 bytes copied, 0.0033774 s, 152 kB/s
```

5. Copy the generated BSP image to the local folder on Windows and switch to the directory of folder by Cygwin. Burn all contents of the BSP image but the first four mega bytes into the SD card.
   dd if= fsl-image-base-s32g399ardb3-xxx.rootfs.sdcard of=/dev/sdb bs=1M skip=4 seek=4 && sync



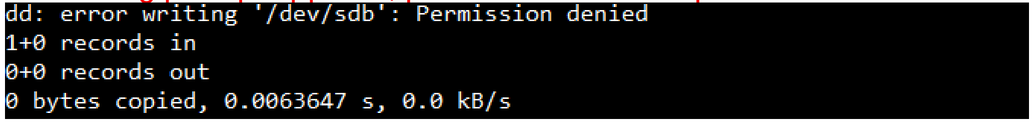
```
$ dd if=fsl-image-base-s32g399ardb3-20220609034839.rootfs.sdcard of=/dev/sdb bs=1M skip=4 seek=4 && sync
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 20.3374 s, 5.2 MB/s
```

6. Burn the first four mega bytes of the BSP image into the SD card
   dd if= fsl-image-base-s32g399ardb3-xxx.rootfs.sdcard of=/dev/sdb bs=1M count=4 && sync



```
$ dd if=fsl-image-base-s32g399ardb3-20220609034839.rootfs.sdcard of=/dev/sdb bs=1M count=4 && sync
4+0 records in
4+0 records out
4194304 bytes (4.2 MB, 4.0 MiB) copied, 0.564993 s, 7.4 MB/s
```

Note:
If the following prompt appears, please follow the steps below

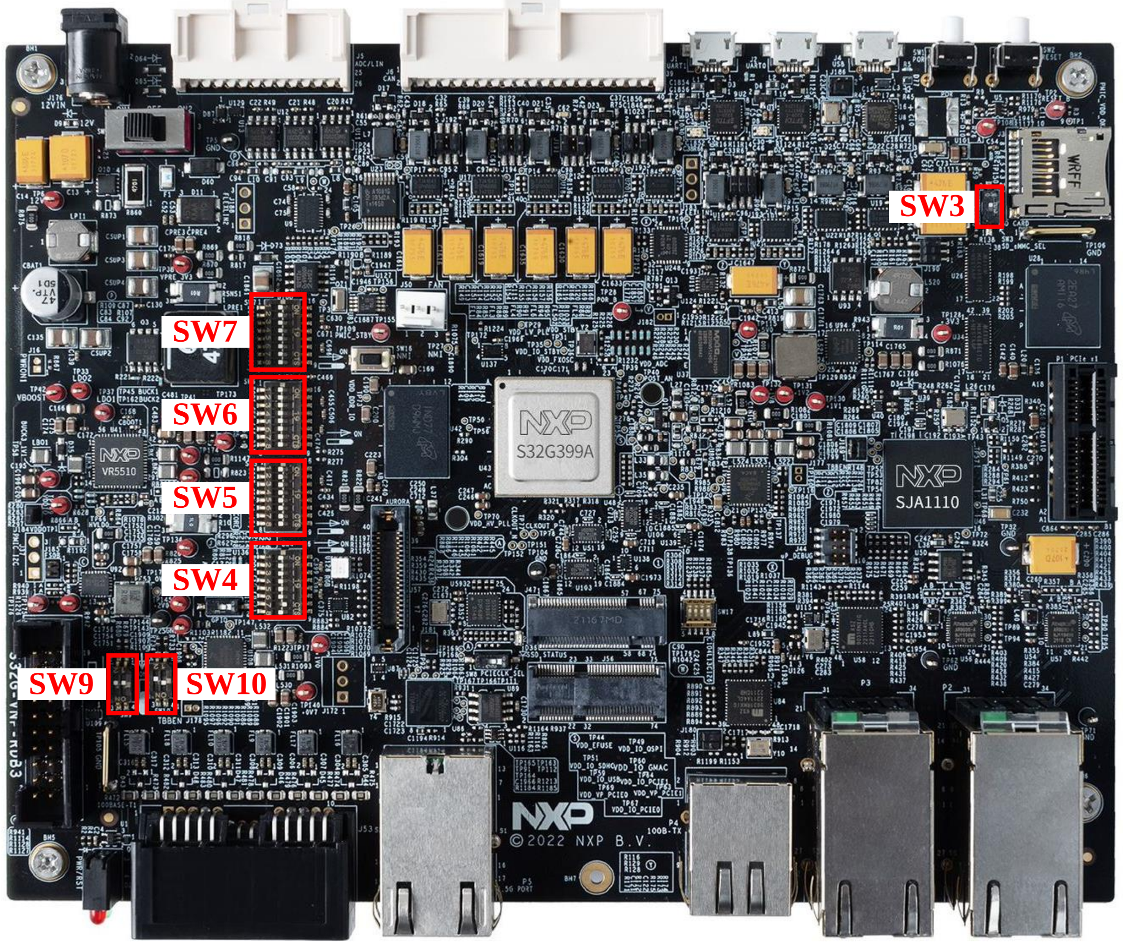


```
dd: error writing '/dev/sdb': Permission denied
1+0 records in
0+0 records out
0 bytes copied, 0.0063647 s, 0.0 kB/s
```

① Take out the SD card and insert it again
② Execute dd if=/dev/zero of=/dev/sdb bs=512 count=1 && sync
③ Take out the SD card and insert it again
④ Burn the image to the SD card

STEP 5: SELECT BOOT MODE OF RDB2

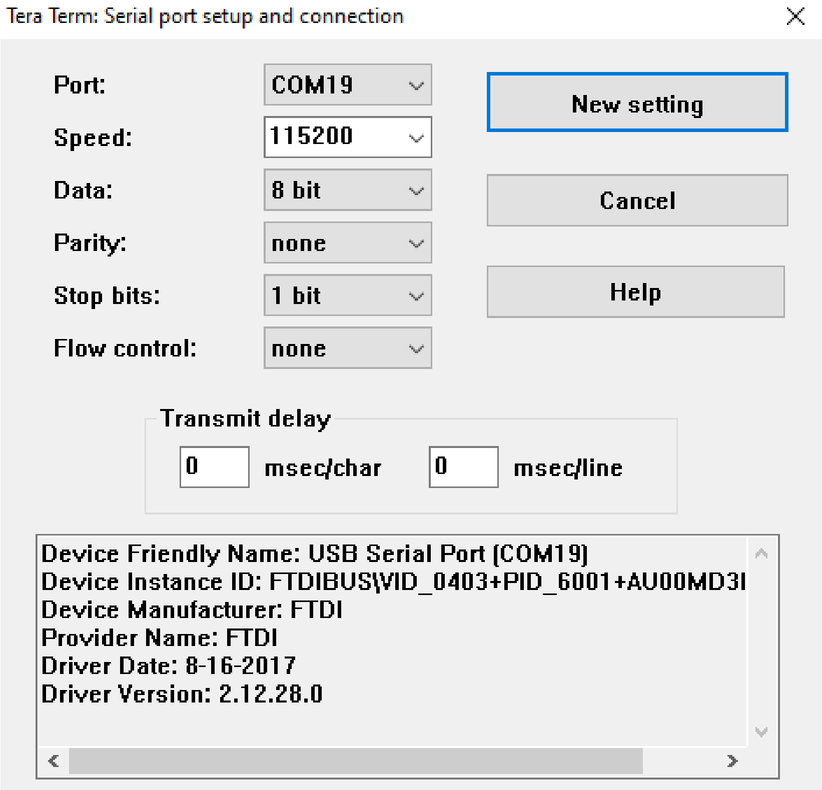
- Set GLDBOX3 to SD card boot mode



Part Reference Number	Setting
SW3	ON
SW4	7-ON, Other-OFF
SW9	1-OFF, 2-OFF
SW10	1-ON, 2-OFF

STEP 6: RUN LINUX BSP

- Connect UART cable to UART0. Then open serial terminal and configure COM port, the baud rate is 115200.



Tera Term: Serial port setup and connection

Port: COM19

Speed: 115200

Data: 8 bit

Parity: none

Stop bits: 1 bit

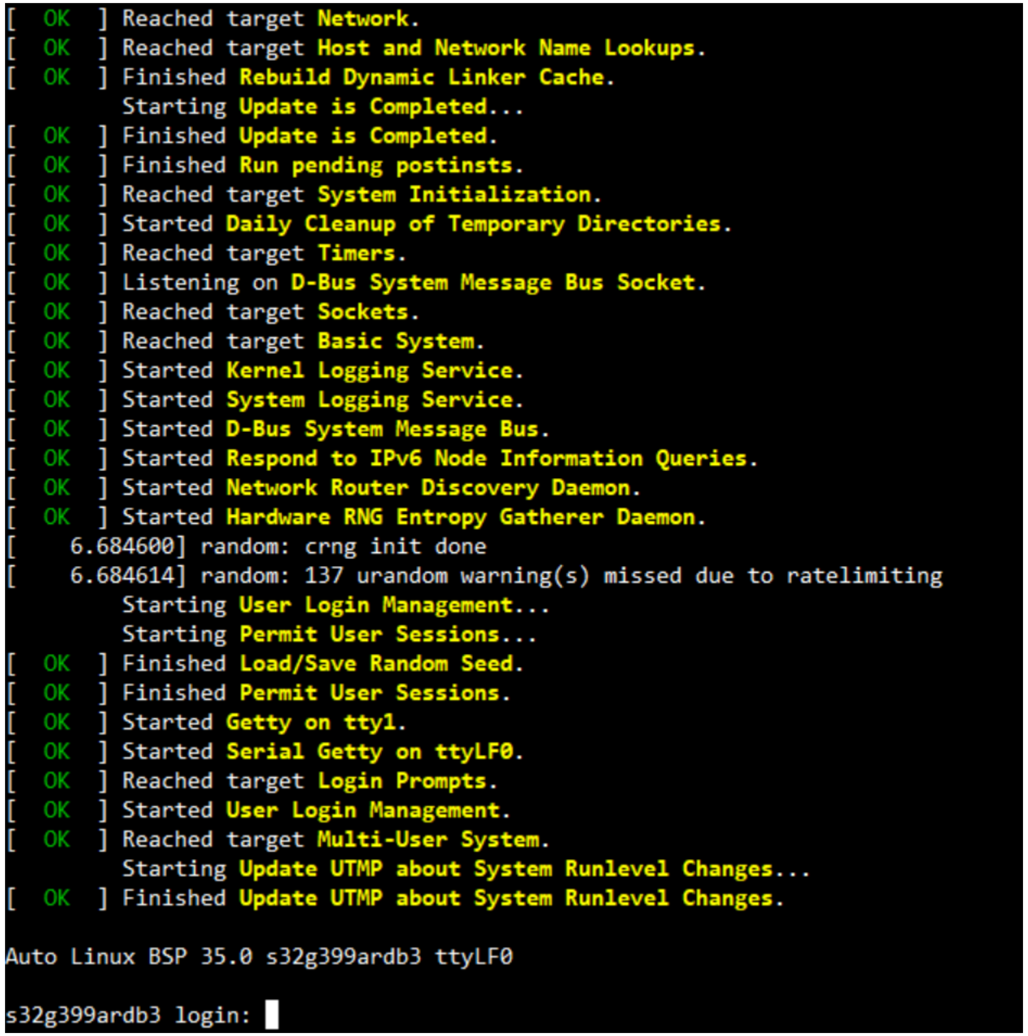
Flow control: none

Transmit delay

0 msec/char 0 msec/line

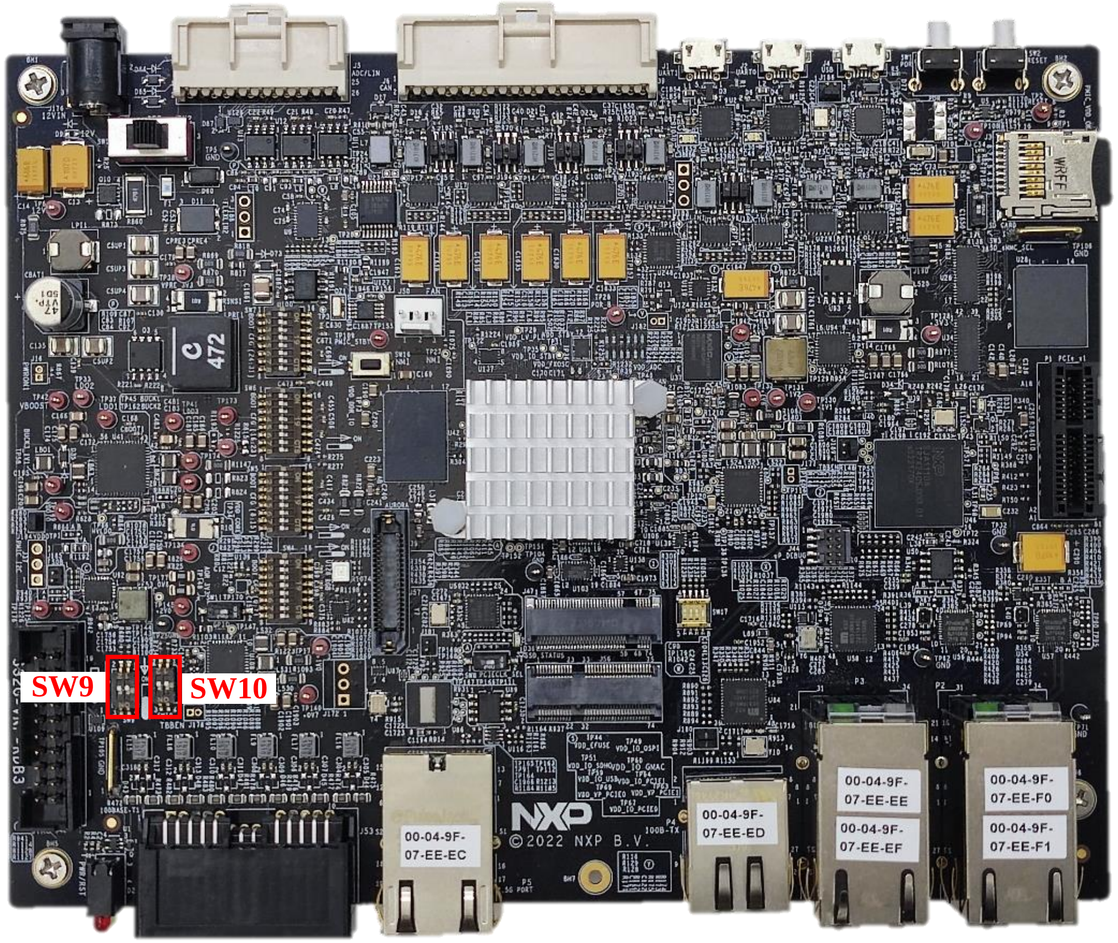
Device Friendly Name: USB Serial Port (COM19)
Device Instance ID: FTDIBUS\VID_0403+PID_6001+AU00MD3I
Device Manufacturer: FTDI
Provider Name: FTDI
Driver Date: 8-16-2017
Driver Version: 2.12.28.0

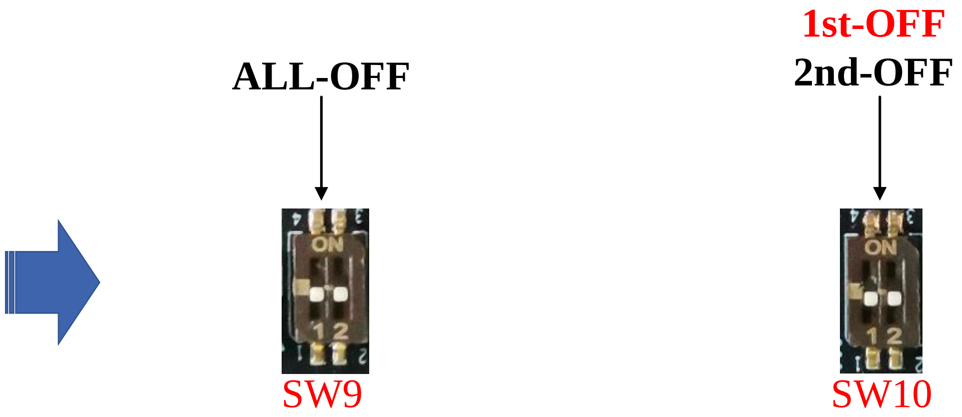
- Power up the S32G-VNP-GLDBOX3 and view print message in serial terminal



```
[  OK  ] Reached target Network.
[  OK  ] Reached target Host and Network Name Lookups.
[  OK  ] Finished Rebuild Dynamic Linker Cache.
         Starting Update is Completed...
[  OK  ] Finished Update is Completed.
[  OK  ] Finished Run pending postinsts.
[  OK  ] Reached target System Initialization.
[  OK  ] Started Daily Cleanup of Temporary Directories.
[  OK  ] Reached target Timers.
[  OK  ] Listening on D-Bus System Message Bus Socket.
[  OK  ] Reached target Sockets.
[  OK  ] Reached target Basic System.
[  OK  ] Started Kernel Logging Service.
[  OK  ] Started System Logging Service.
[  OK  ] Started D-Bus System Message Bus.
[  OK  ] Started Respond to IPv6 Node Information Queries.
[  OK  ] Started Network Router Discovery Daemon.
[  OK  ] Started Hardware RNG Entropy Gatherer Daemon.
[    6.684600] random: crng init done
[    6.684614] random: 137 urandom warning(s) missed due to ratelimiting
         Starting User Login Management...
         Starting Permit User Sessions...
[  OK  ] Finished Load/Save Random Seed.
[  OK  ] Finished Permit User Sessions.
[  OK  ] Started Getty on tty1.
[  OK  ] Started Serial Getty on ttyLF0.
[  OK  ] Reached target Login Prompts.
[  OK  ] Started User Login Management.
[  OK  ] Reached target Multi-User System.
         Starting Update UTMP about System Runlevel Changes...
[  OK  ] Finished Update UTMP about System Runlevel Changes.

Auto Linux BSP 35.0 s32g399ardb3 ttyLF0

s32g399ardb3 login:
```

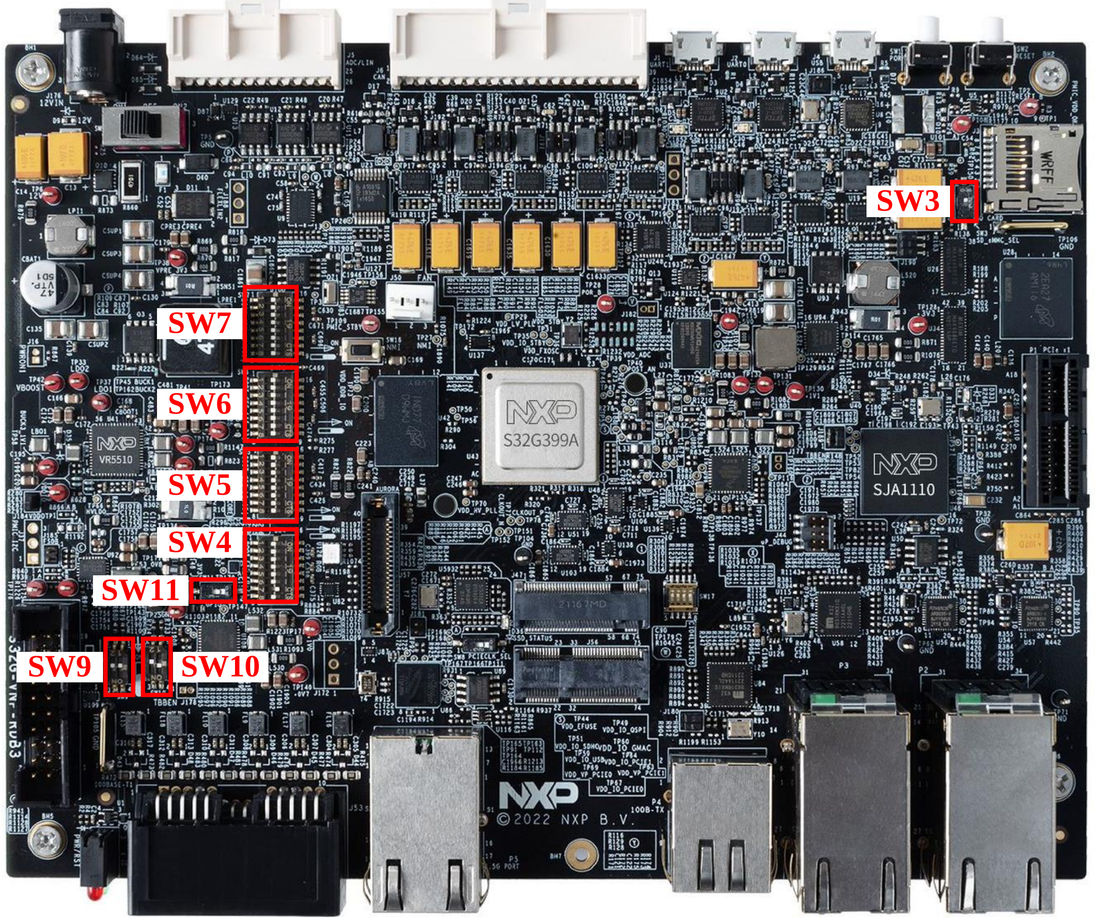
APPENDIX A: S32G-VNP-GLDBOX3 SERIAL BOOT SWITCH SETTINGS

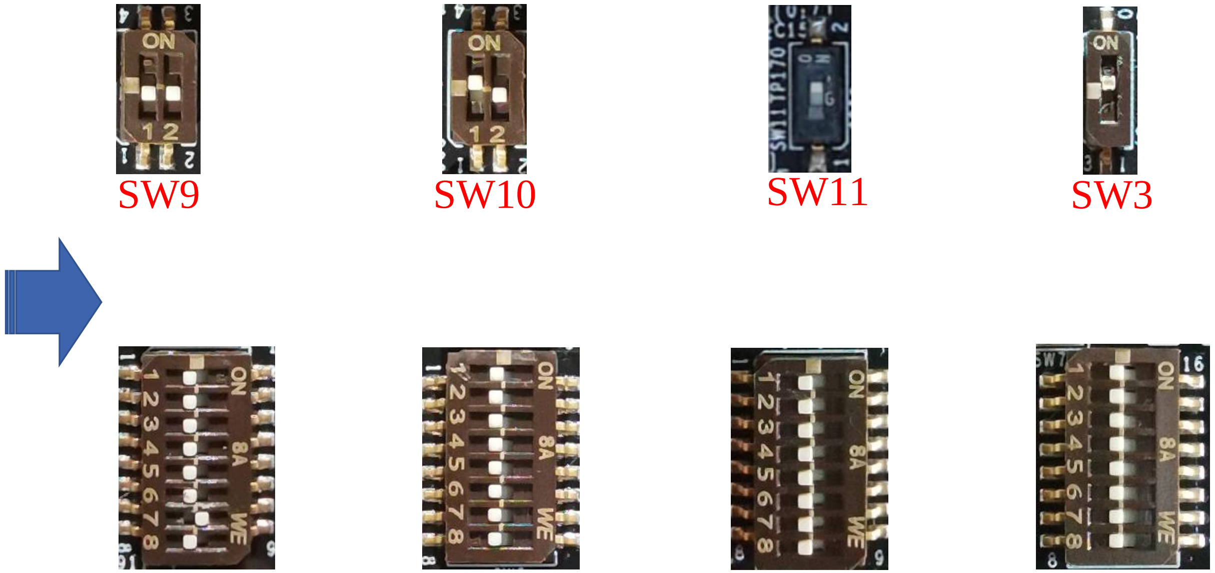




Compare with default setting, the 1st of SW10 need to be changed

APPENDIX B: S32G-VNP-GLDBOX3 SD-CARD BOOT AND LED LIGHT UP SWITCH SETTINGS





SW9: ALL-OFF

SW10: 1st-ON
2nd-OFF

SW11: ON

SW3: ON

SW4: 6th-OFF
7th-ON
OTHERS-OFF

SW5: ALL-OFF

SW6: ALL-OFF

SW7: ALL-OFF

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(s). 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.

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.

SECURE CONNECTIONS
FOR A SMARTER WORLD