S32G-VNP-RDB2

SOFTWARE ENABLEMENT GUIDE

SECURE CONNECTIONS
FOR A SMARTER WORLD

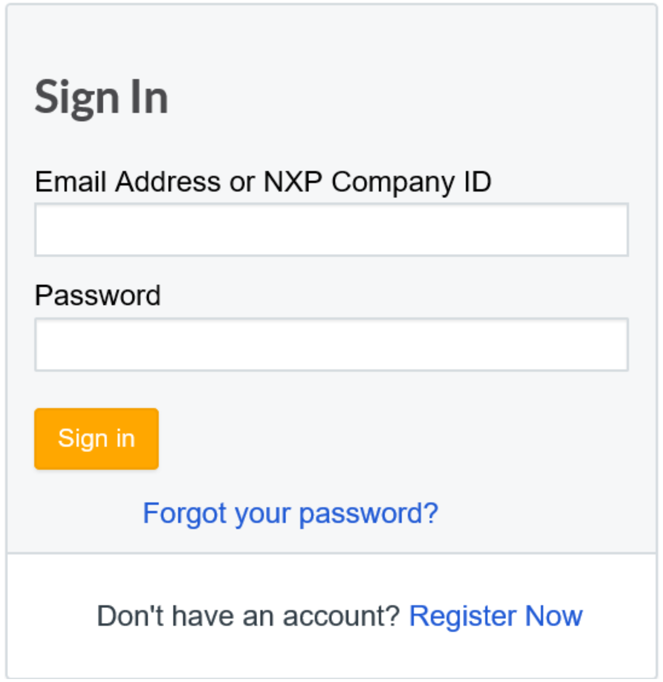
CONTENTS SOFTWARE ENABLEMENT GUIDE

PRE: Get S32G Software

SECURE CONNECTIONS
FOR A SMARTER WORLD

PREREQUISITES: GET S32G SOFTWARE

- 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

Don't have an account? Register Now

Please firstly download the below software for enablement

Production	Install Packages
S32 Design Studio for S32 Platform	SW32G2_S32DS_3.4.0_D2012.zip
	S32DS.3.4_b201217_win32.x86_64.exe
S32G2 - Real Time Drivers	S32_RTD_4.4_1.0.0_HF01_D2102_DS_Updatesite.zip
S32G2 - Linux BSP*	binaries_auto_linux_bsp28.0_s32g274_pfe.tgz
	S32G274_LinuxBSP28.0.0_User_Manual.pdf

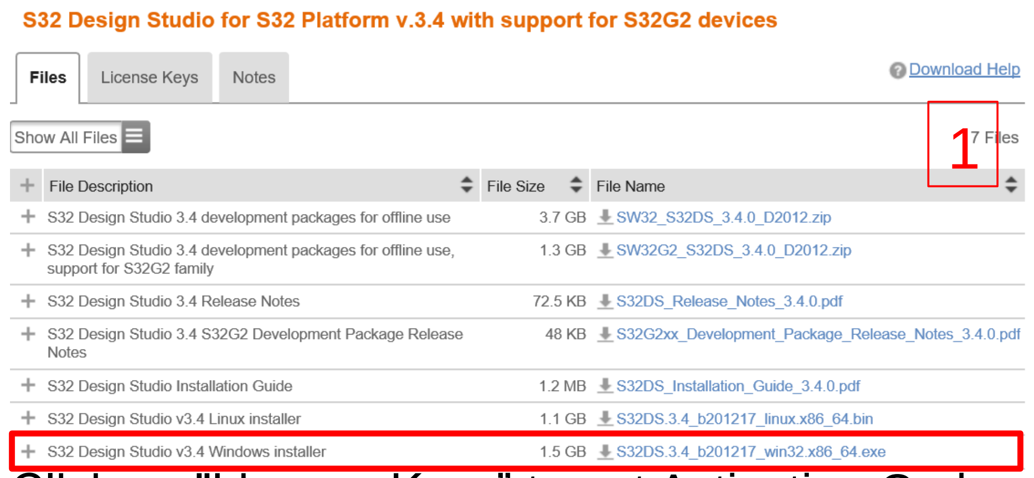
*: User can download the BSP28 or the newer Linux BSP version

Install Software Development Tool

SECURE CONNECTIONS
FOR A SMARTER WORLD

STEP 1: INSTALL S32 DESIGN STUDIO 3.4

- Download installation package for your machine



S32 Design Studio for S32 Platform v.3.4 with support for S32G2 devices

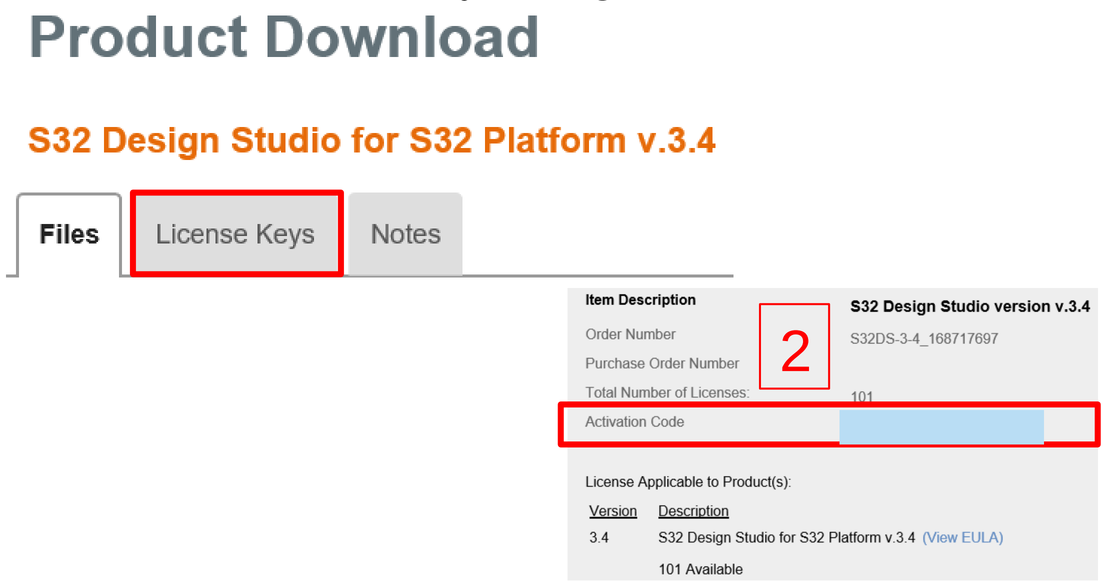
Files | License Keys | Notes — Download Help

Show All Files — 7 Files

1

File Description	File Size	File Name
S32 Design Studio 3.4 development packages for offline use	3.7 GB	SW32_S32DS_3.4.0_D2012.zip
S32 Design Studio 3.4 development packages for offline use, support for S32G2 family	1.3 GB	SW32G2_S32DS_3.4.0_D2012.zip
S32 Design Studio 3.4 Release Notes	72.5 KB	S32DS_Release_Notes_3.4.0.pdf
S32 Design Studio 3.4 S32G2 Development Package Release Notes	48 KB	S32G2xx_Development_Package_Release_Notes_3.4.0.pdf
S32 Design Studio Installation Guide	1.2 MB	S32DS_Installation_Guide_3.4.0.pdf
S32 Design Studio v3.4 Linux installer	1.1 GB	S32DS.3.4_b201217_linux.x86_64.bin
S32 Design Studio v3.4 Windows installer	1.5 GB	S32DS.3.4_b201217_win32.x86_64.exe

- Click on "License Keys” to get Activation Code



Product Download

S32 Design Studio for S32 Platform v.3.4

Files | License Keys | Notes

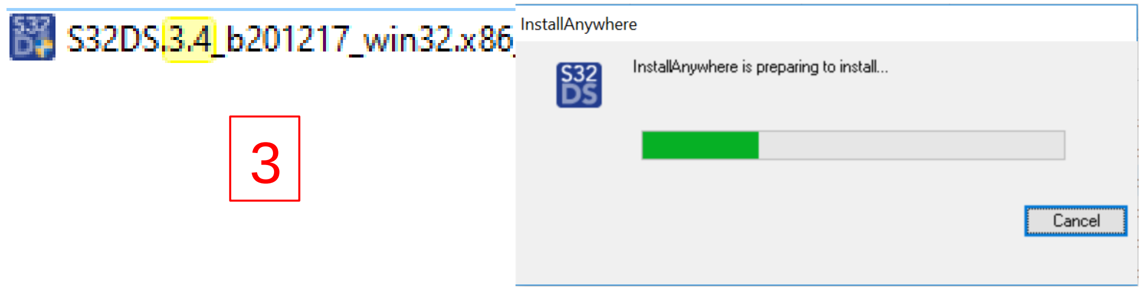
2

Item Description	S32 Design Studio version v.3.4
Order Number	S32DS-3-4_168717697
Purchase Order Number	
Total Number of Licenses:	101
Activation Code	

License Applicable to Product(s):

Version	Description
3.4	S32 Design Studio for S32 Platform v.3.4 (View EULA)
	101 Available

- Click .exe file to start installation



S32DS.3.4_b201217_win32.x86

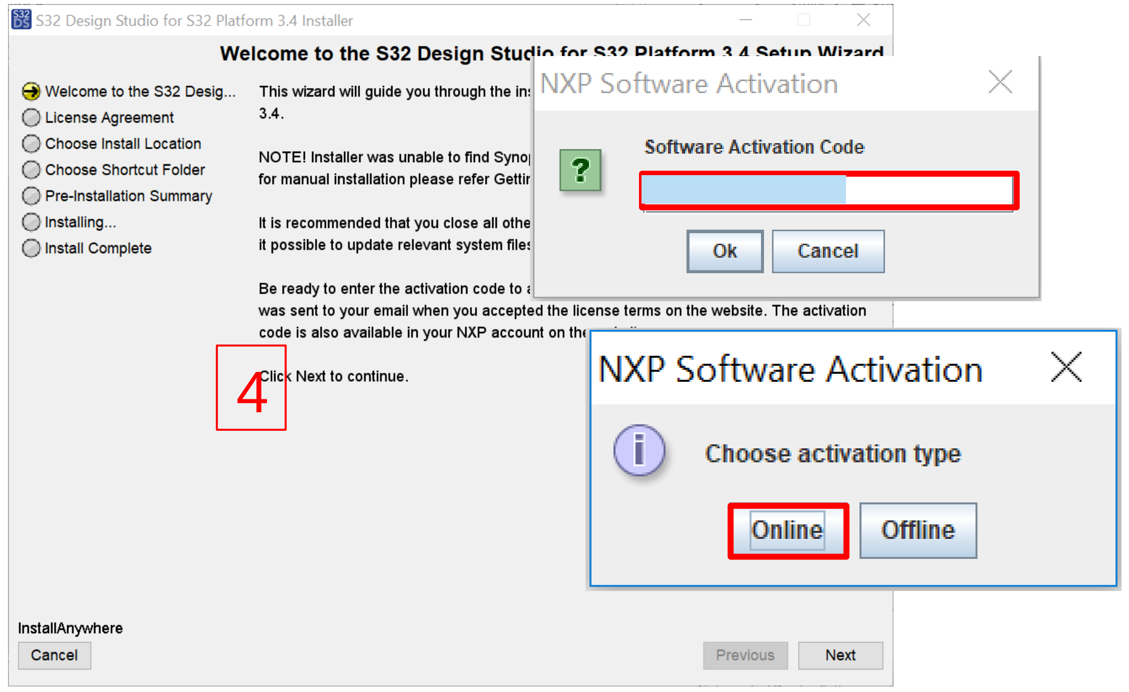
3

InstallAnywhere

InstallAnywhere is preparing to install...

Cancel

- Click “Next” to step by step install. Input the Activation code got in step3 when necessary and click on “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 ins 3.4.

NOTE! Installer was unable to find Synop for manual installation please refer Gettir

It is recommended that you close all othe it possible to update relevant system files

Be ready to enter the activation code to a 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

Click Next to continue.

4

InstallAnywhere

Cancel | Previous | Next

NXP Software Activation

Software Activation Code

Ok | Cancel

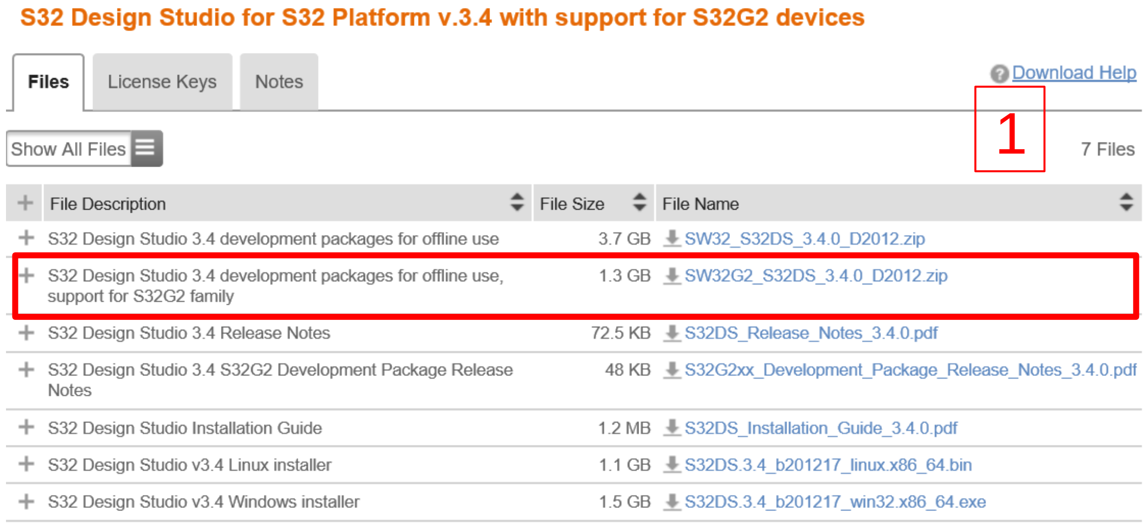
NXP Software Activation

Choose activation type

Online | Offline

STEP 2: INSTALL DEVELOPMENT PACKAGES FOR S32G2

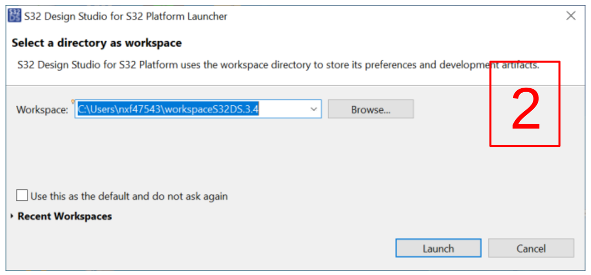
- Download S32G2 development packages



S32 Design Studio for S32 Platform v.3.4 with support for S32G2 devices

File Description	File Size	File Name
S32 Design Studio 3.4 development packages for offline use	3.7 GB	SW32_S32DS_3.4.0_D2012.zip
S32 Design Studio 3.4 development packages for offline use, support for S32G2 family	1.3 GB	SW32G2_S32DS_3.4.0_D2012.zip
S32 Design Studio 3.4 Release Notes	72.5 KB	S32DS_Release_Notes_3.4.0.pdf
S32 Design Studio 3.4 S32G2 Development Package Release Notes	48 KB	S32G2xx_Development_Package_Release_Notes_3.4.0.pdf
S32 Design Studio Installation Guide	1.2 MB	S32DS_Installation_Guide_3.4.0.pdf
S32 Design Studio v3.4 Linux installer	1.1 GB	S32DS.3.4_b201217_linux.x86_64.bin
S32 Design Studio v3.4 Windows installer	1.5 GB	S32DS.3.4_b201217_win32.x86_64.exe

- Open S32DS 3.4 and select a directory as workspace and click "Install New Software …" option on help menu



S32 Design Studio for S32 Platform Launcher

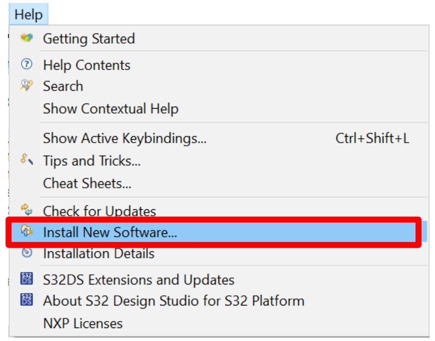
Select a directory as workspace

S32 Design Studio for S32 Platform uses the workspace directory to store its preferences and development artifacts.

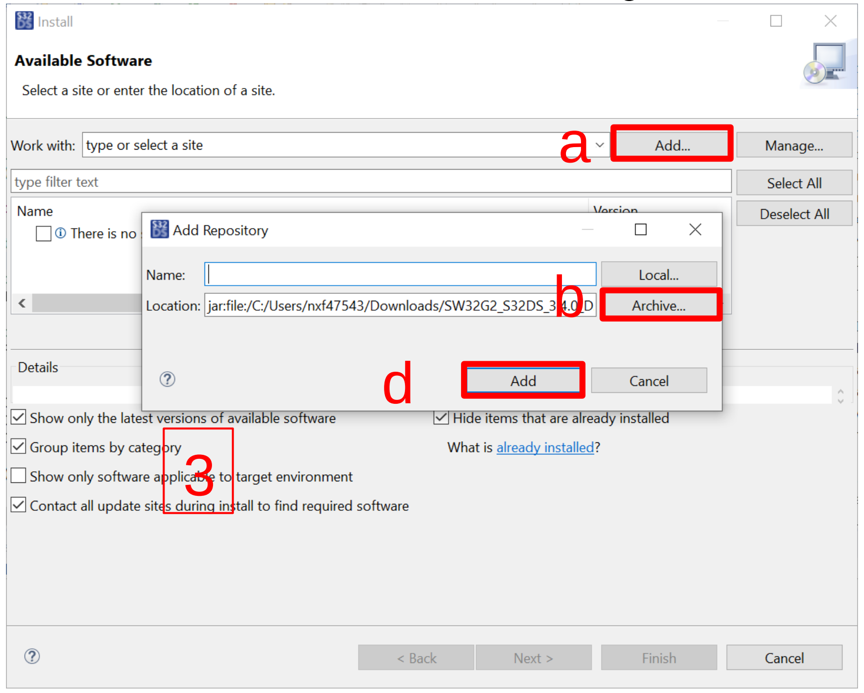
Workspace: C:\Users\nxf47543\workspaceS32DS.3.4

☐ Use this as the default and do not ask again

Recent Workspaces



- Add update package of S32DS
  a. Click on "Add" button "
  b. Click on "Archive" button in Add Site dialog
  c. Select SW32G2_S32DS_3.4.0_D2012.zip file and click on "open"
  d. Click on "Add" button in Add Site dialog



Install

Available Software

Select a site or enter the location of a site.

Work with: type or select a site

Add Repository

Name:

Location: jar:file:/C:/Users/nxf47543/Downloads/SW32G2_S32DS_3.4.0_D

☑ Show only the latest versions of available software

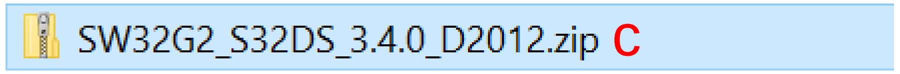
☑ Group items by category

☐ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

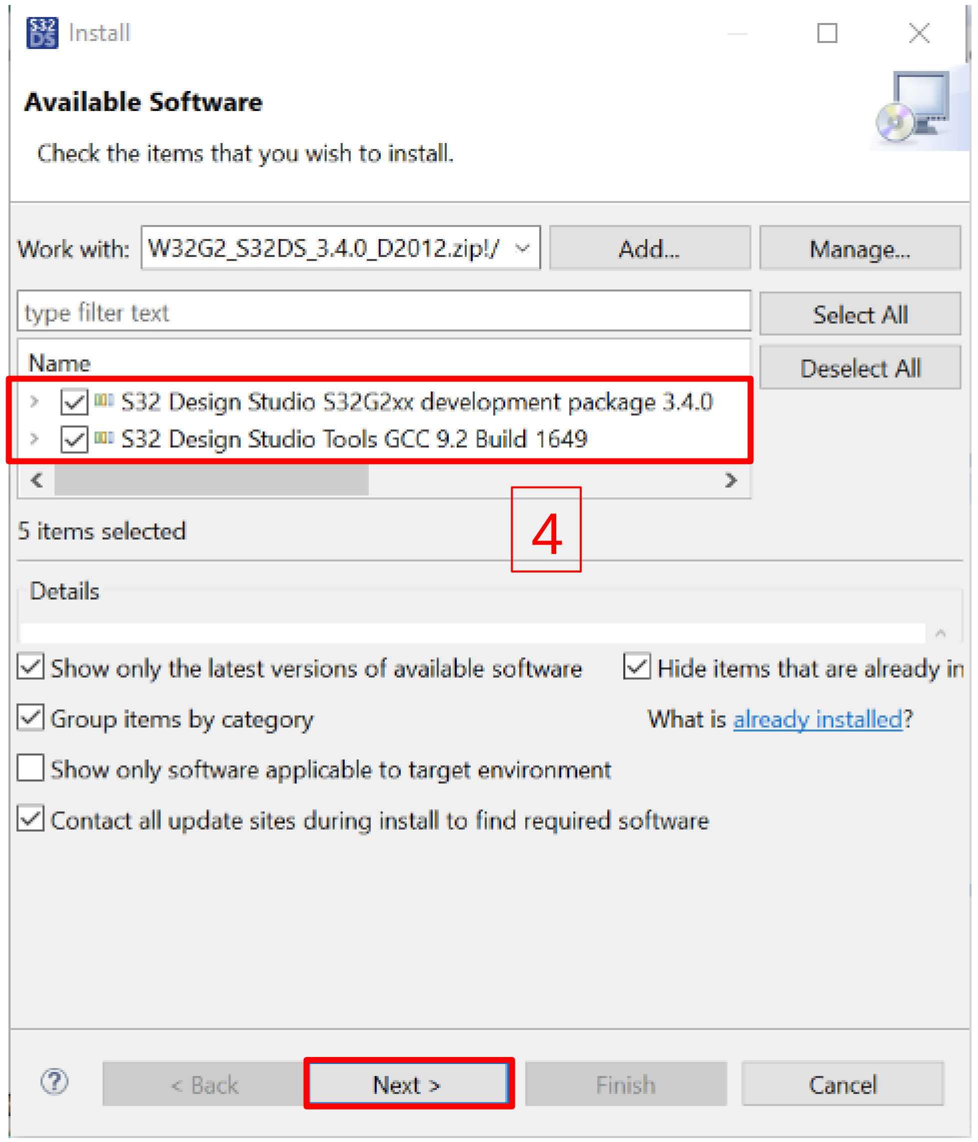
☑ Hide items that are already installed

What is already installed?



STEP 2: INSTALL UPDATE WITH SUPPORT FOR S32G2

- Select the two package and click on ”Next>” button



Install

Available Software

Check the items that you wish to install.

Work with: W32G2_S32DS_3.4.0_D2012.zip!/

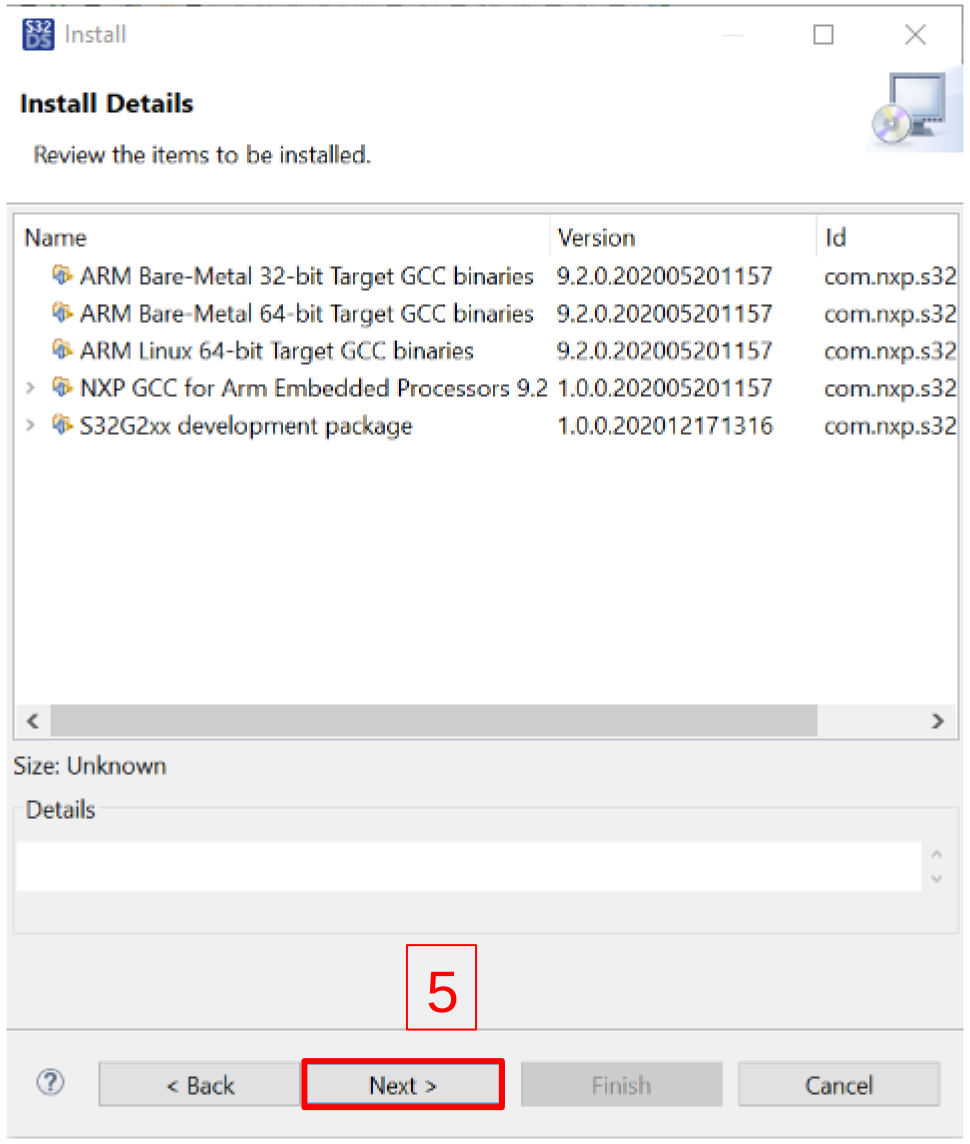
- [✓] S32 Design Studio S32G2xx development package 3.4.0
- [✓] S32 Design Studio Tools GCC 9.2 Build 1649

5 items selected

Details

- [✓] Show only the latest versions of available software
- [✓] Hide items that are already in
- [✓] 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.

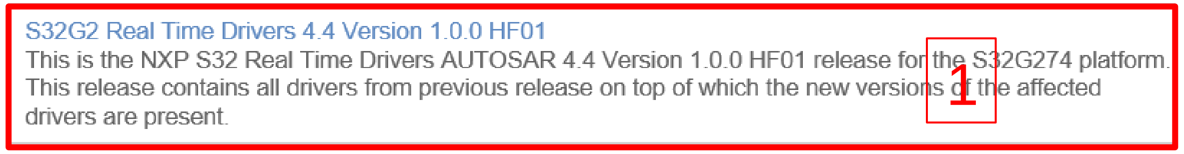
Name	Version	Id
ARM Bare-Metal 32-bit Target GCC binaries	9.2.0.202005201157	com.nxp.s32
ARM Bare-Metal 64-bit Target GCC binaries	9.2.0.202005201157	com.nxp.s32
ARM Linux 64-bit Target GCC binaries	9.2.0.202005201157	com.nxp.s32
NXP GCC for Arm Embedded Processors 9.2	1.0.0.202005201157	com.nxp.s32
S32G2xx development package	1.0.0.202012171316	com.nxp.s32

Size: Unknown

Details

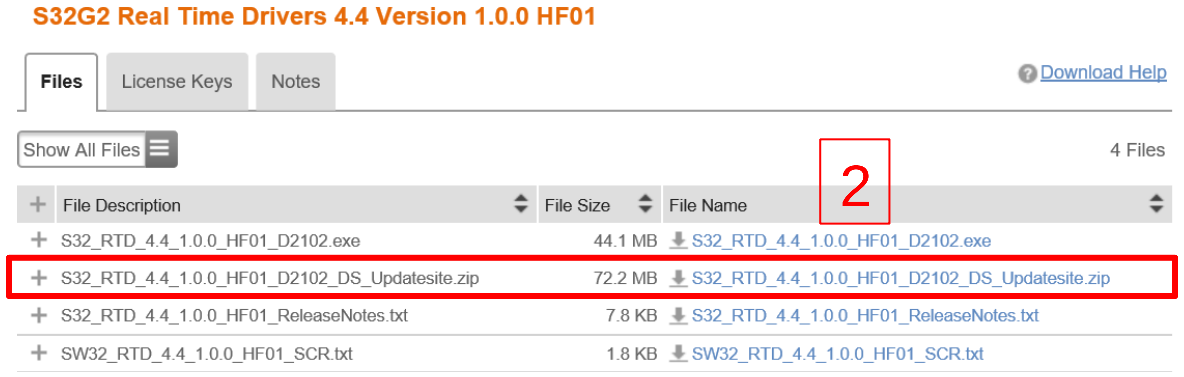
STEP 3: INSTALL S32G2 REAL-TIME DRIVERS

- Click on “S32G2 Real Time Drivers 4.4 Version 1.0.0 HF01”



S32G2 Real Time Drivers 4.4 Version 1.0.0 HF01
This is the NXP S32 Real Time Drivers AUTOSAR 4.4 Version 1.0.0 HF01 release for the S32G274 platform. This release contains all drivers from previous release on top of which the new versions of the affected drivers are present.

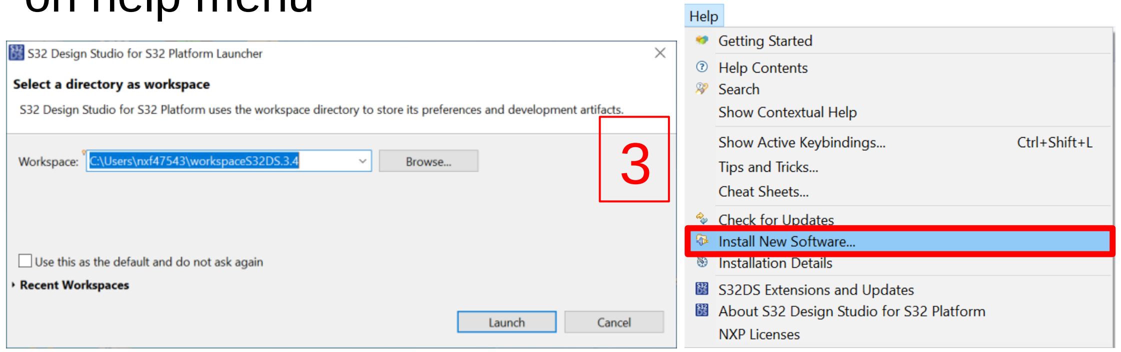
- Download S32_RTD_4.4_1.0.0_HF01_D2102_DS_Updatesite.zip package



S32G2 Real Time Drivers 4.4 Version 1.0.0 HF01

File Description	File Size	File Name
S32_RTD_4.4_1.0.0_HF01_D2102.exe	44.1 MB	S32_RTD_4.4_1.0.0_HF01_D2102.exe
S32_RTD_4.4_1.0.0_HF01_D2102_DS_Updatesite.zip	72.2 MB	S32_RTD_4.4_1.0.0_HF01_D2102_DS_Updatesite.zip
S32_RTD_4.4_1.0.0_HF01_ReleaseNotes.txt	7.8 KB	S32_RTD_4.4_1.0.0_HF01_ReleaseNotes.txt
SW32_RTD_4.4_1.0.0_HF01_SCR.txt	1.8 KB	SW32_RTD_4.4_1.0.0_HF01_SCR.txt

- Open S32DS 3.4 and select a directory as workspace and click “Install New Software …” option on help menu



S32 Design Studio for S32 Platform Launcher

Select a directory as workspace

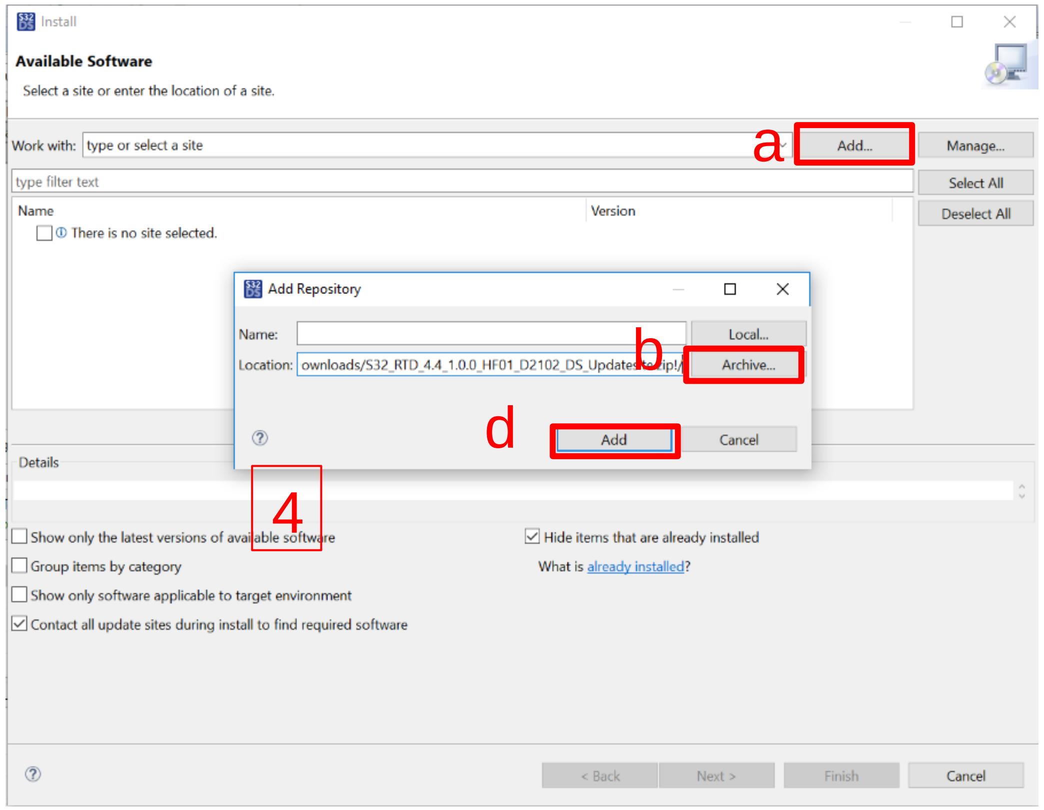
S32 Design Studio for S32 Platform uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\Users\nxf47543\workspaceS32DS.3.4

☐ Use this as the default and do not ask again

Recent Workspaces

- Add update package of S32DS
  - a. Click on “Add…”
  - b. Click on “Archive” button in Add Repository dialog
  - c. Select S32_RTD_4.4_1.0.0_HF01_D2102_DS_Updatesite.zip file and click on “open”
  - d. Click on ”ok” to go back “Available Software” dialog



Install

Available Software

Select a site or enter the location of a site.

Work with: type or select a site

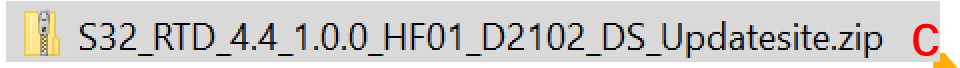
There is no site selected.

Add Repository

Name:

Location: ownloads/S32_RTD_4.4_1.0.0_HF01_D2102_DS_Updatesite.zip!/

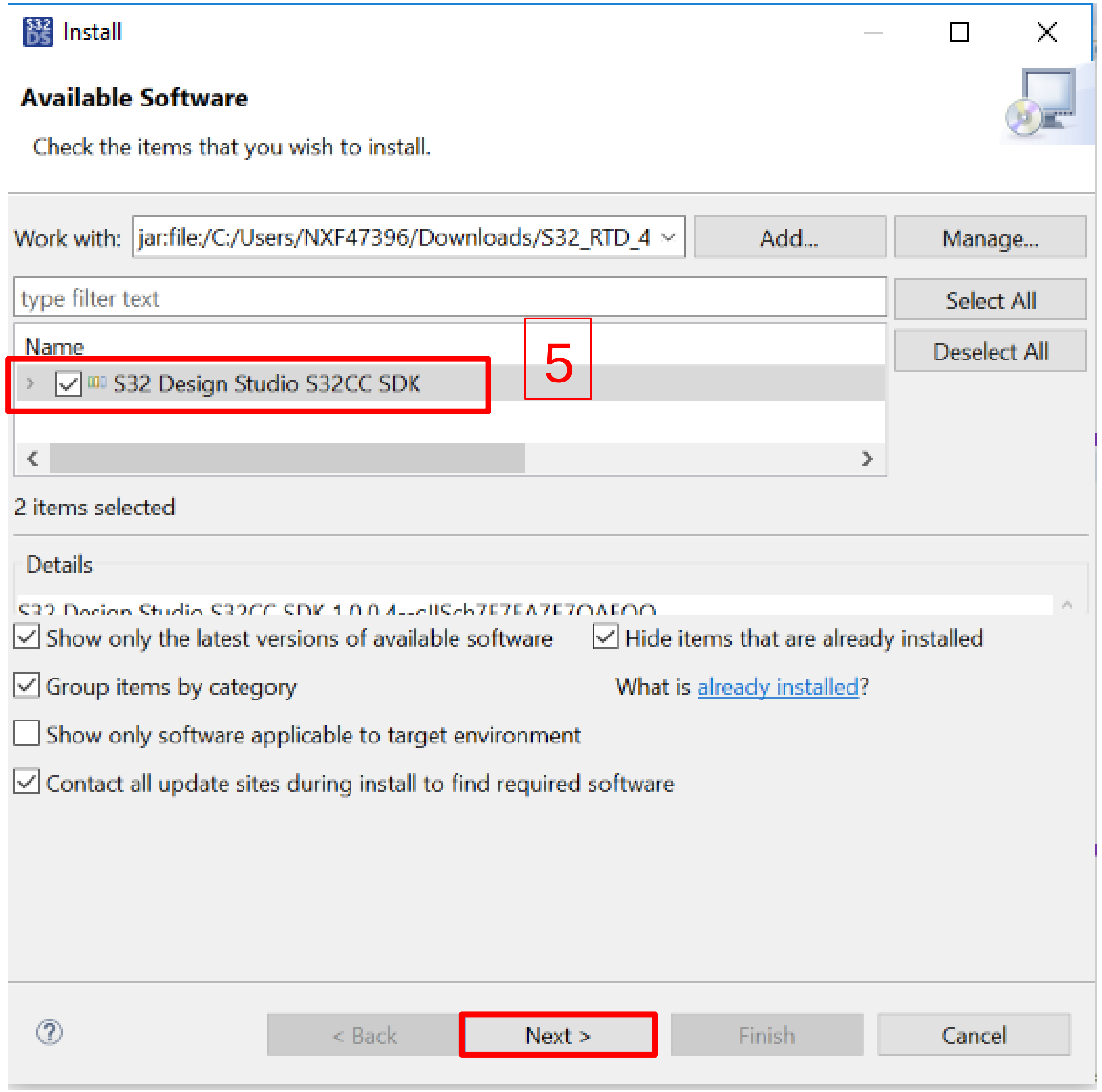
☐ Show only the latest versions of available software
☑ Hide items that are already installed
☐ Group items by category
What is already installed?
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software



S32_RTD_4.4_1.0.0_HF01_D2102_DS_Updatesite.zip

STEP 3: INSTALL S32G2 REAL-TIME DRIVERS

- Check the "RTD S32CC" box 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/NXF47396/Downloads/S32_RTD_4

type filter text

Name

☑ S32 Design Studio S32CC SDK

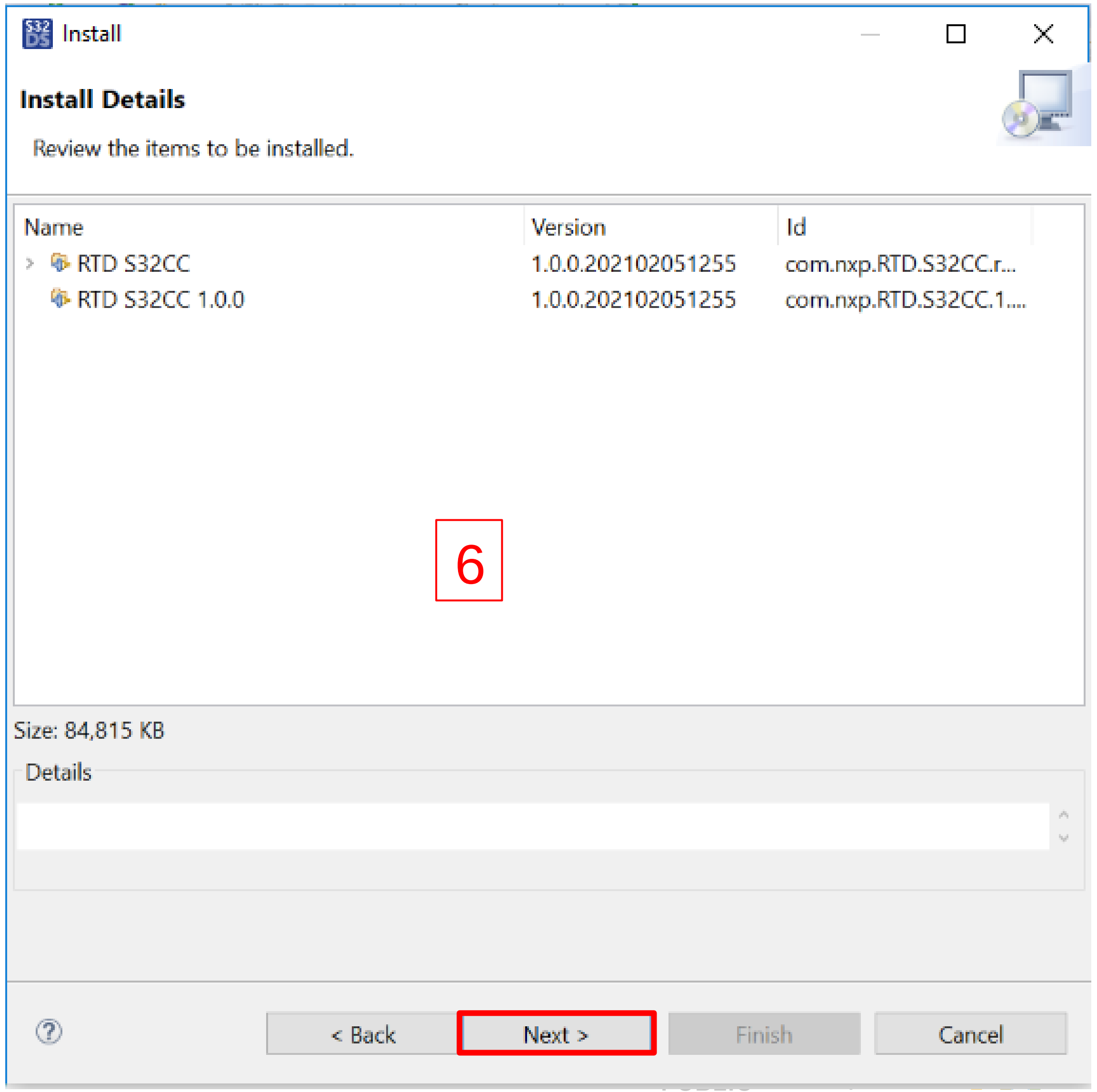
5

2 items 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 S32CC	1.0.0.202102051255	com.nxp.RTD.S32CC.r...
RTD S32CC 1.0.0	1.0.0.202102051255	com.nxp.RTD.S32CC.1....

6

Size: 84,815 KB

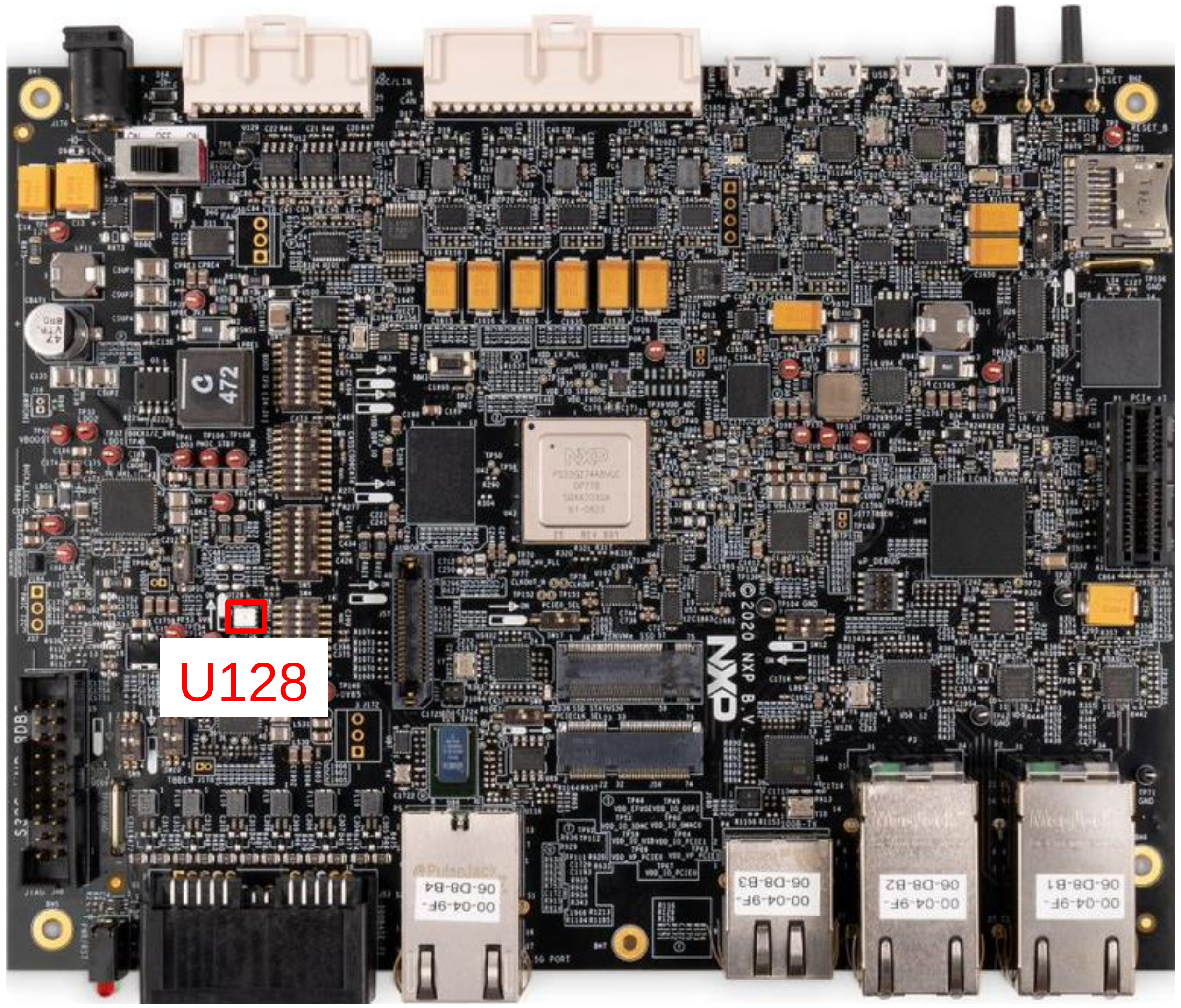
Details

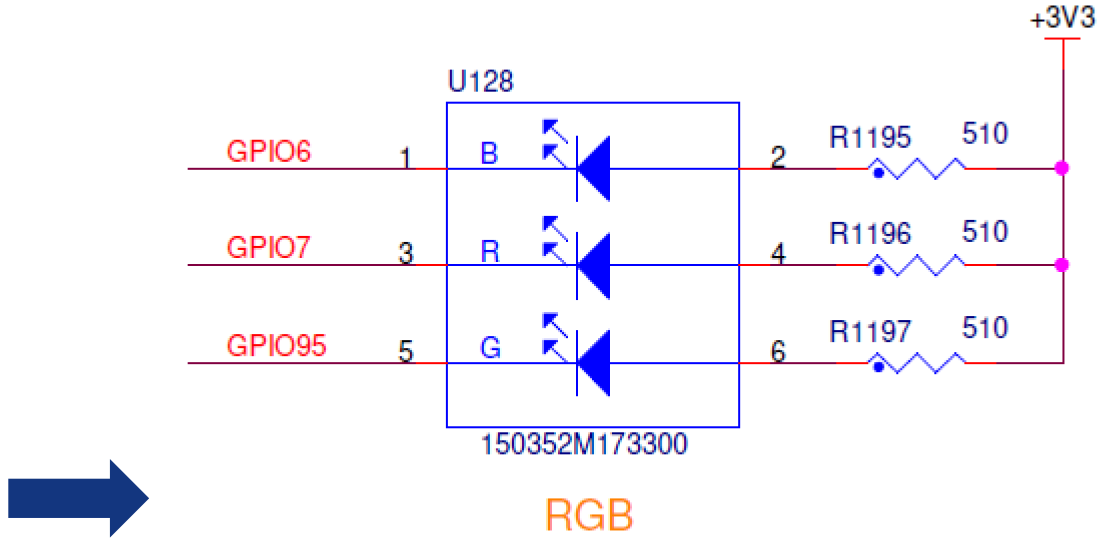
Light Up RGB LED Based On Real Time Drivers

SECURE CONNECTIONS
FOR A SMARTER WORLD

LIGHT UP RGB LED: HARDWARE RESOURCES

- Resources to be used: on-board RGB LED

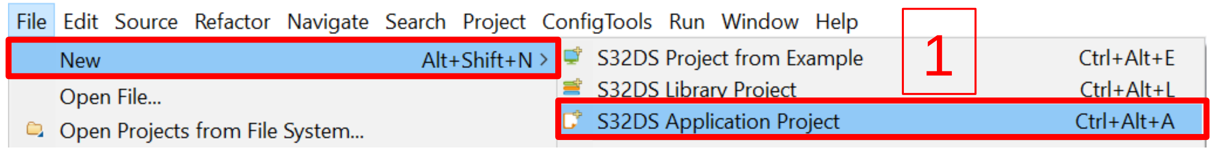




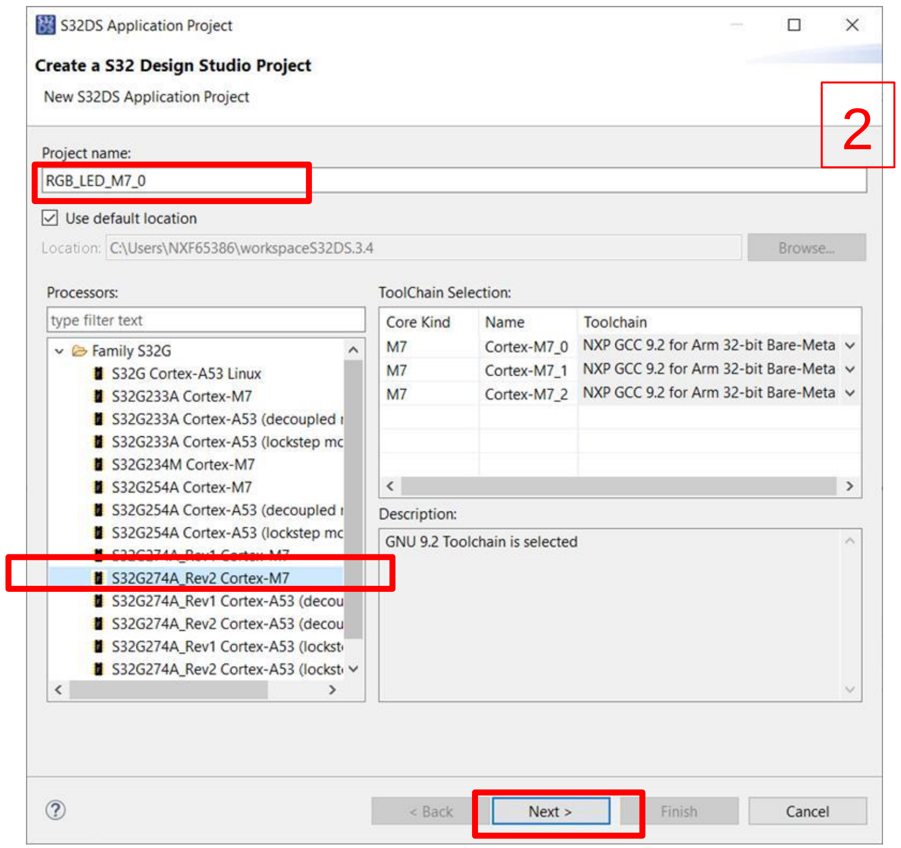
- An RGB LED is a combination of three LED in one package: 1x Blue LED, 1x Red LED and 1x Green LED.
- Because the LEDs are very close to each other, our eyes see the result of the combination of colors, rather than the three colors individually.

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 and select S32G274A_REV2_Cortex-M7 as Processors, then click on "Next"


S32DS Application Project
Create a S32 Design Studio Project
New S32DS Application Project
2
Next >
Back
Finish
Cancel
Browse...

Project name: RGB_LED_M7_0

☑ Use default location

Location: C:\Users\NXF65386\workspaceS32DS.3.4

Processors:

- Family S32G
  - S32G Cortex-A53 Linux
  - S32G233A Cortex-M7
  - S32G233A Cortex-A53 (decoupled
  - S32G233A Cortex-A53 (lockstep mc
  - S32G234M Cortex-M7
  - S32G254A Cortex-M7
  - S32G254A Cortex-A53 (decoupled
  - S32G254A Cortex-A53 (lockstep mc
  - S32G274A_Rev1 Cortex-M7
  - S32G274A_Rev2 Cortex-M7
  - S32G274A_Rev1 Cortex-A53 (decou
  - S32G274A_Rev2 Cortex-A53 (decou
  - S32G274A_Rev1 Cortex-A53 (lockst
  - S32G274A_Rev2 Cortex-A53 (lockst

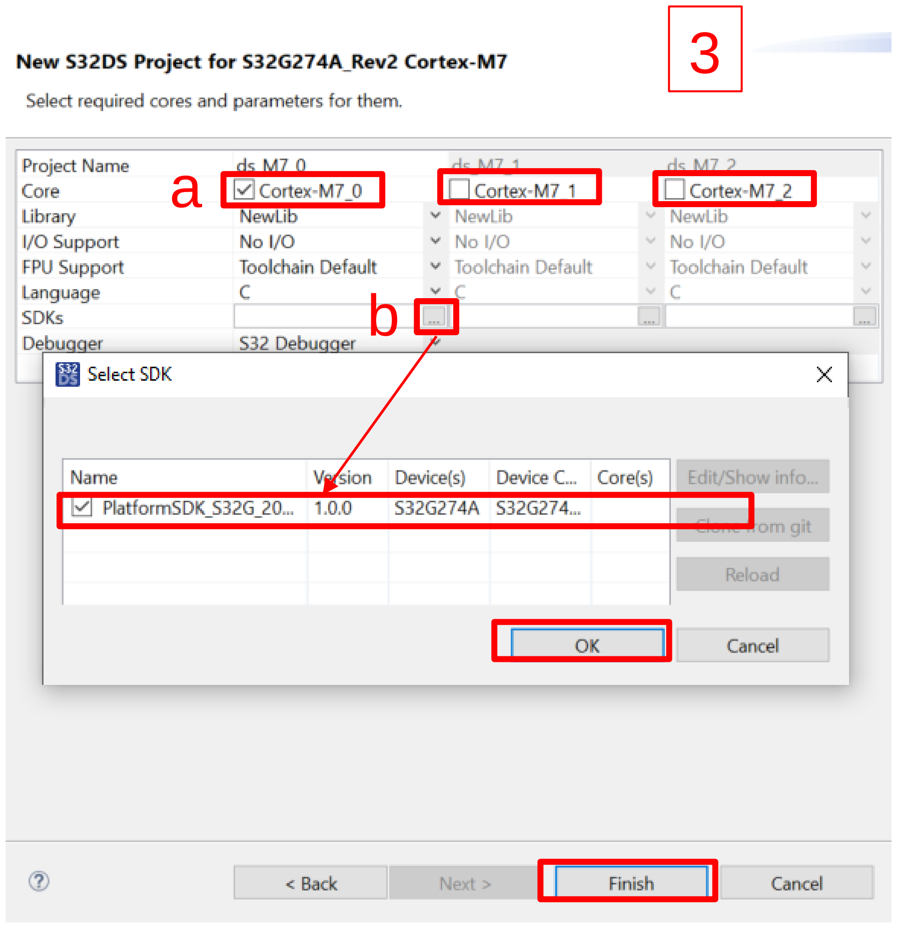
ToolChain Selection:

Core Kind	Name	Toolchain
M7	Cortex-M7_0	NXP GCC 9.2 for Arm 32-bit Bare-Meta
M7	Cortex-M7_1	NXP GCC 9.2 for Arm 32-bit Bare-Meta
M7	Cortex-M7_2	NXP GCC 9.2 for Arm 32-bit Bare-Meta

Description:

GNU 9.2 Toolchain is selected

- Select required core and SDKs
  a. Check only Cortex-M7_0 core
  b. Check 'RTD_CD01…' as SDKs and click on "ok"
  c. Click on "Finish" to complete configuration



New S32DS Project for S32G274A_Rev2 Cortex-M7

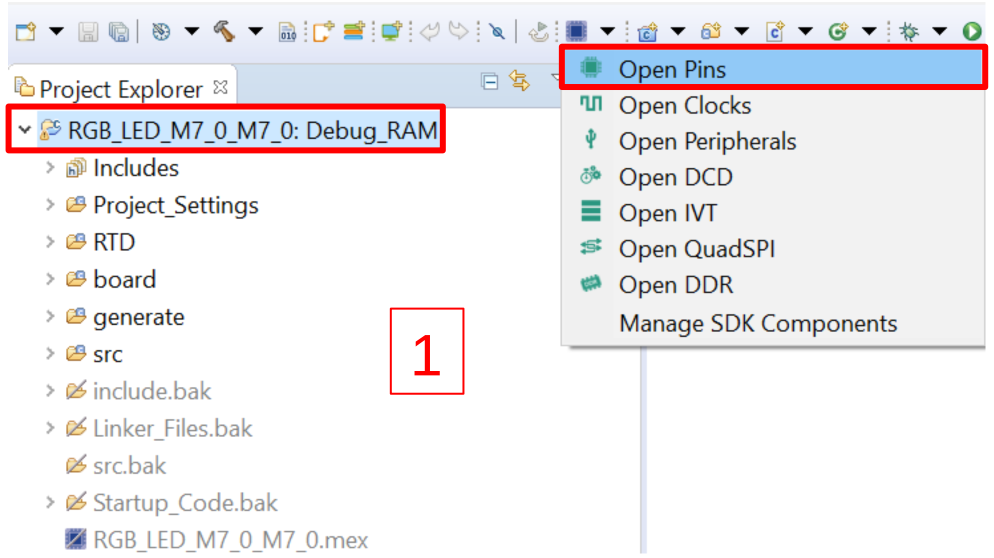
Select required cores and parameters for them.

Project Name	ds_M7_0	ds_M7_1	ds_M7_2
Core	☑ Cortex-M7_0	☐ Cortex-M7_1	☐ Cortex-M7_2
Library	NewLib	NewLib	NewLib
I/O Support	No I/O	No I/O	No I/O
FPU Support	Toolchain Default	Toolchain Default	Toolchain Default
Language	C	C	C
SDKs			
Debugger	S32 Debugger		

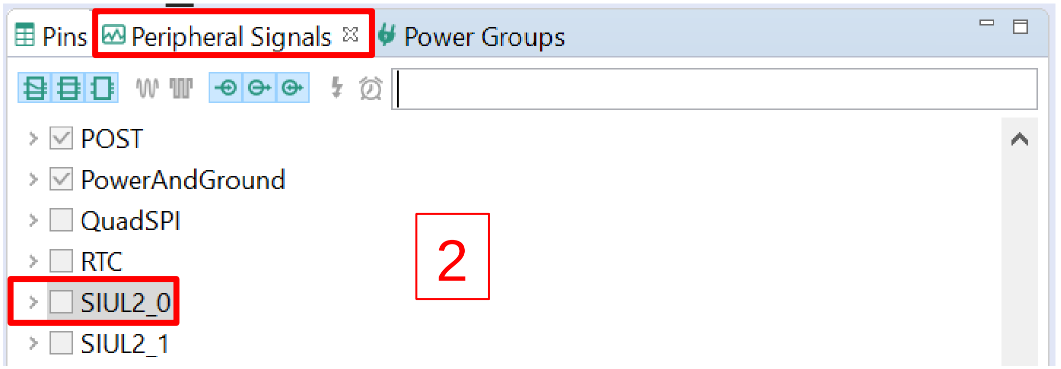
Name	Version	Device(s)	Device C...	Core(s)
☑ PlatformSDK_S32G_20...	1.0.0	S32G274A	S32G274...	

LIGHT UP RGB LED: PINS CONFIGURATION

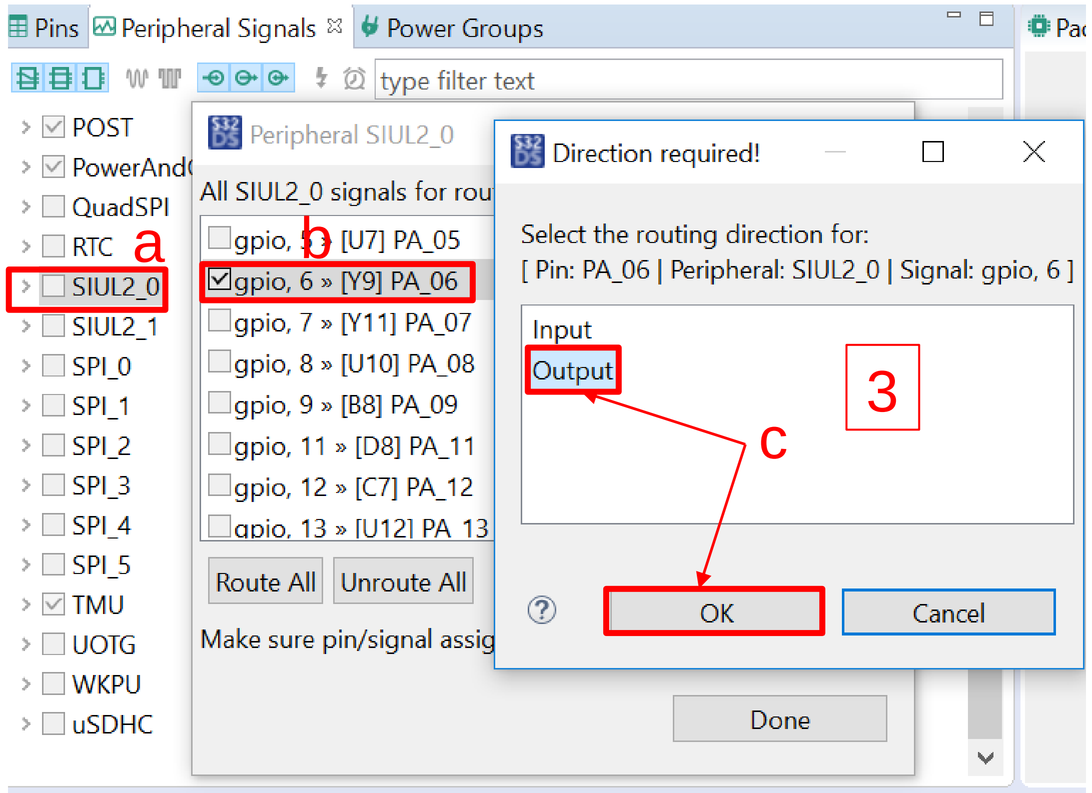
- Select the created project and open pins tool



- Select peripheral Signals page and find out SIUL2_0 box



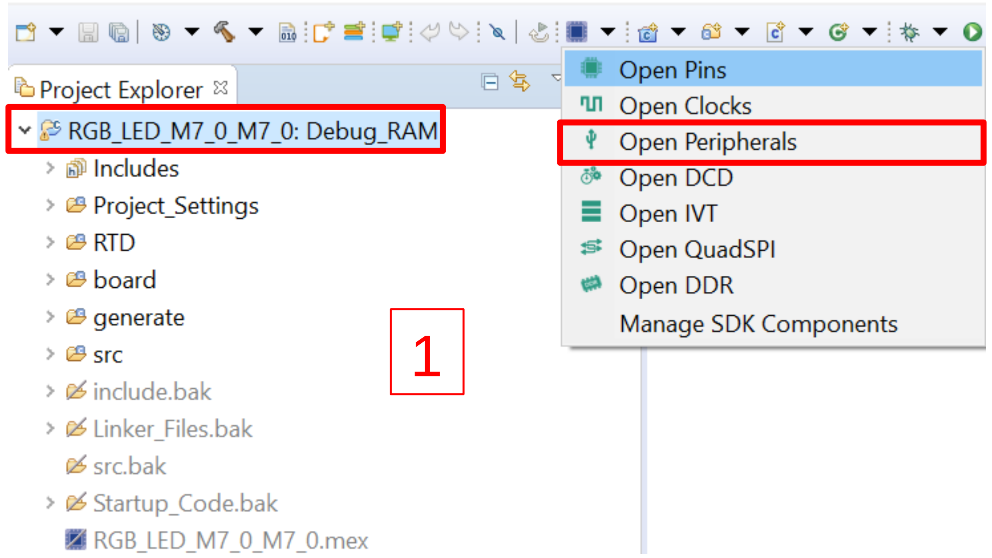
- Configure the corresponding gpio pins according to page 12.
  a. Click on “SIU2_0”
  b. Check gpio, 6 box
  c. Click on “Output” and ok to complete one pin configuration
  d. Follow b and c to configure gpio 7, 95, then click on “Done”



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



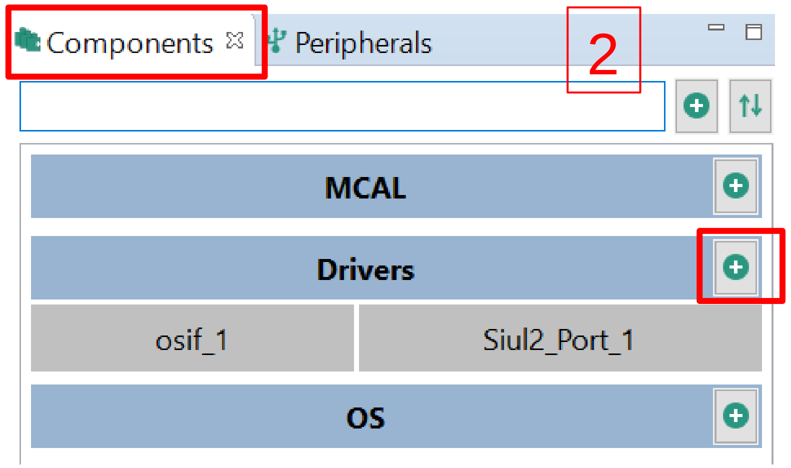
Project Explorer

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

Open Pins
Open Clocks
Open Peripherals
Open DCD
Open IVT
Open QuadSPI
Open DDR
Manage SDK Components

1

- Select Components to find out Drivers option

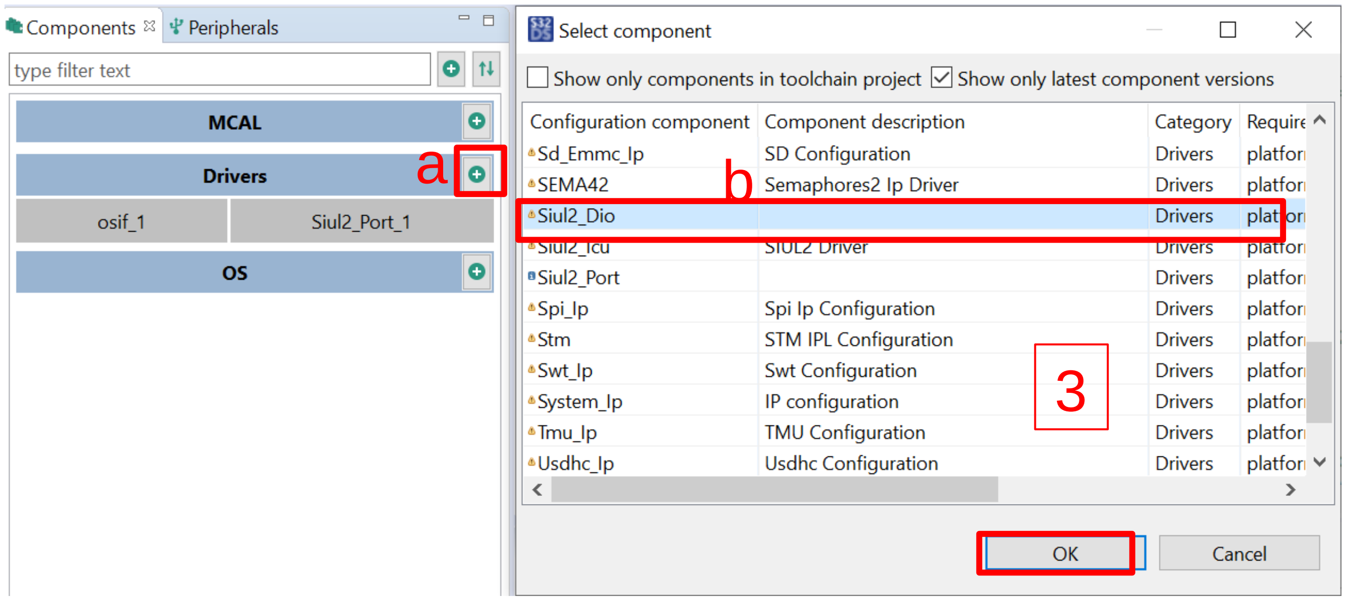


Components | Peripherals

MCAL
Drivers
osif_1 | Siul2_Port_1
OS

2

- Add gpio dio driver
  a. Click on "+" option
  b. Select "Siul2_Dio" and click on "ok"



Components | Peripherals

type filter text

MCAL
Drivers
osif_1 | Siul2_Port_1
OS

a

Select component

☐ Show only components in toolchain project ☑ Show only latest component versions

Configuration component	Component description	Category	Require
Sd_Emmc_Ip	SD Configuration	Drivers	platfor
SEMA42	Semaphores2 Ip Driver	Drivers	platfor
Siul2_Dio		Drivers	platfor
Siul2_Icu	SIUL2 Driver	Drivers	platfor
Siul2_Port		Drivers	platfor
Spi_Ip	Spi Ip Configuration	Drivers	platfor
Stm	STM IPL Configuration	Drivers	platfor
Swt_Ip	Swt Configuration	Drivers	platfor
System_Ip	IP configuration	Drivers	platfor
Tmu_Ip	TMU Configuration	Drivers	platfor
Usdhc_Ip	Usdhc Configuration	Drivers	platfor

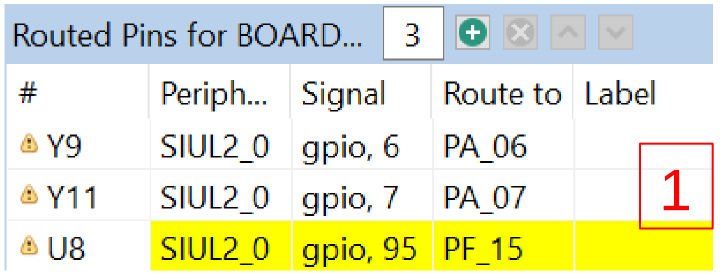
b

3

OK | Cancel

LIGHT UP RGB LED: CHECK CONFIGURATION AND UPDATE CODE

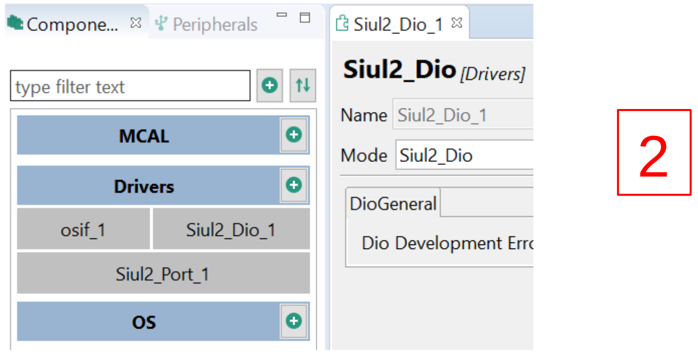
- open pins tool to check configuration



Routed Pins for BOARD... 3

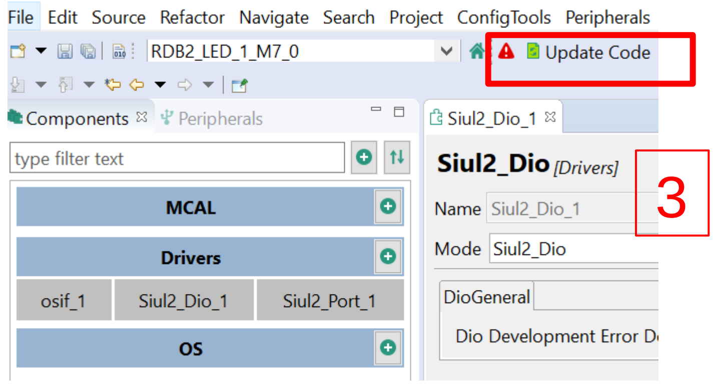
#	Periph...	Signal	Route to	Label
Y9	SIUL2_0	gpio, 6	PA_06	
Y11	SIUL2_0	gpio, 7	PA_07	
U8	SIUL2_0	gpio, 95	PF_15	

- open Peripherals tool to check configuration



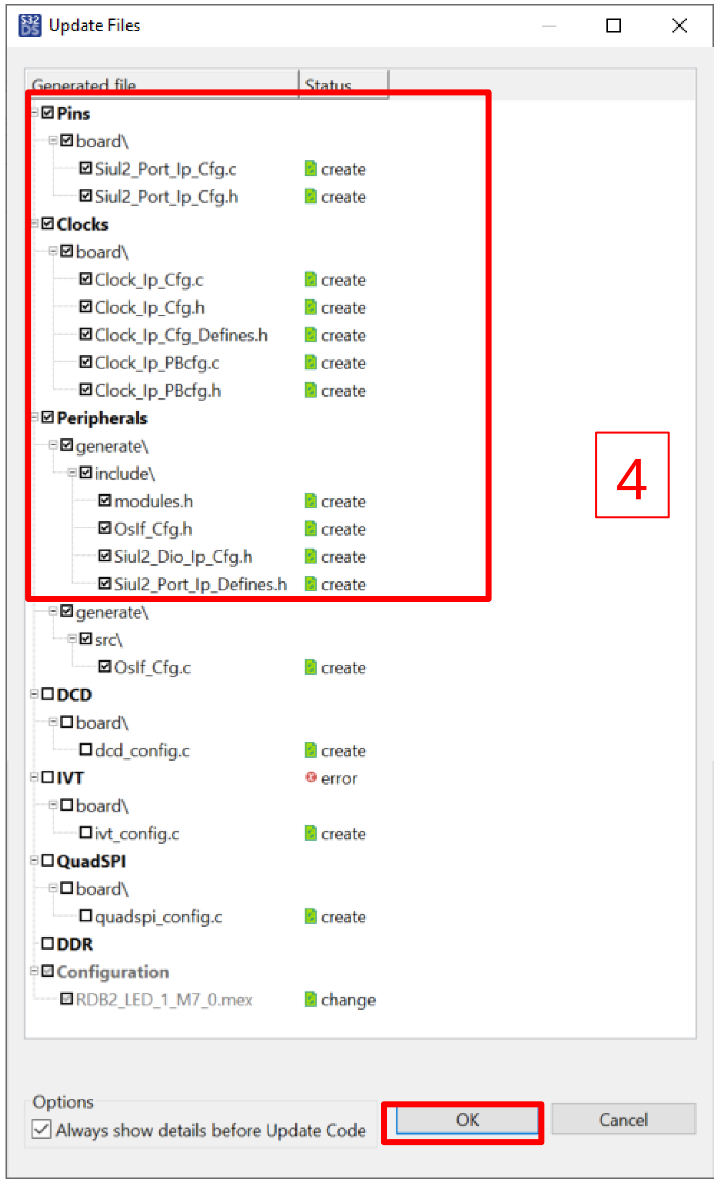
Siul2_Dio [Drivers]
Name Siul2_Dio_1
Mode Siul2_Dio
DioGeneral
Dio Development Err

- Click on “Update Code”



Siul2_Dio [Drivers]
Name Siul2_Dio_1
Mode Siul2_Dio
DioGeneral
Dio Development Error D

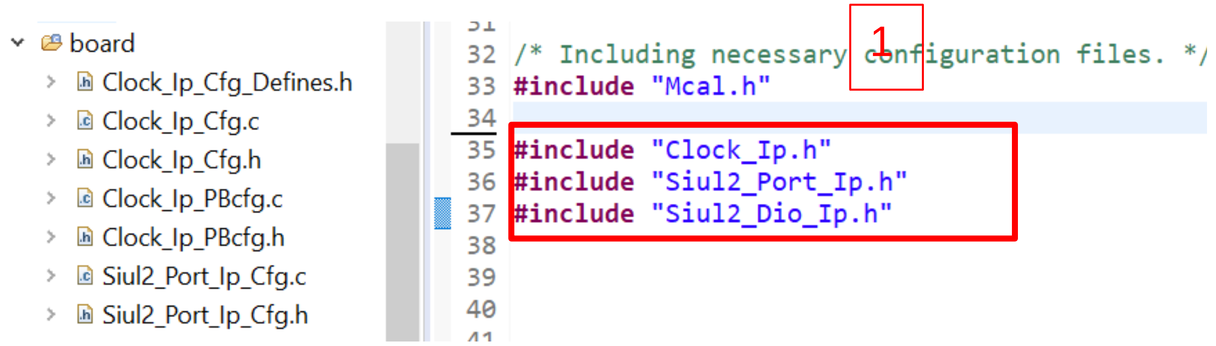
- Check the “Pins” and “Peripherals” box and click on “ok” to start update code, uncheck the other boxes



```
Generated file                         Status
☑ Pins
  ☑ board\
    ☑ Siul2_Port_Ip_Cfg.c              create
    ☑ Siul2_Port_Ip_Cfg.h              create
☑ Clocks
  ☑ board\
    ☑ Clock_Ip_Cfg.c                   create
    ☑ Clock_Ip_Cfg.h                   create
    ☑ Clock_Ip_Cfg_Defines.h           create
    ☑ Clock_Ip_PBcfg.c                 create
    ☑ Clock_Ip_PBcfg.h                 create
☑ Peripherals
  ☑ generate\
    ☑ include\
      ☑ modules.h                      create
      ☑ OsIf_Cfg.h                     create
      ☑ Siul2_Dio_Ip_Cfg.h             create
      ☑ Siul2_Port_Ip_Defines.h        create
  ☑ generate\
    ☑ src\
      ☑ OsIf_Cfg.c                     create
☐ DCD
  ☐ board\
    ☐ dcd_config.c                     create
☐ IVT                                  error
  ☐ board\
    ☐ ivt_config.c                     create
☐ QuadSPI
  ☐ board\
    ☐ quadspi_config.c                 create
☐ DDR
☑ Configuration
  ☑ RDB2_LED_1_M7_0.mex                change

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

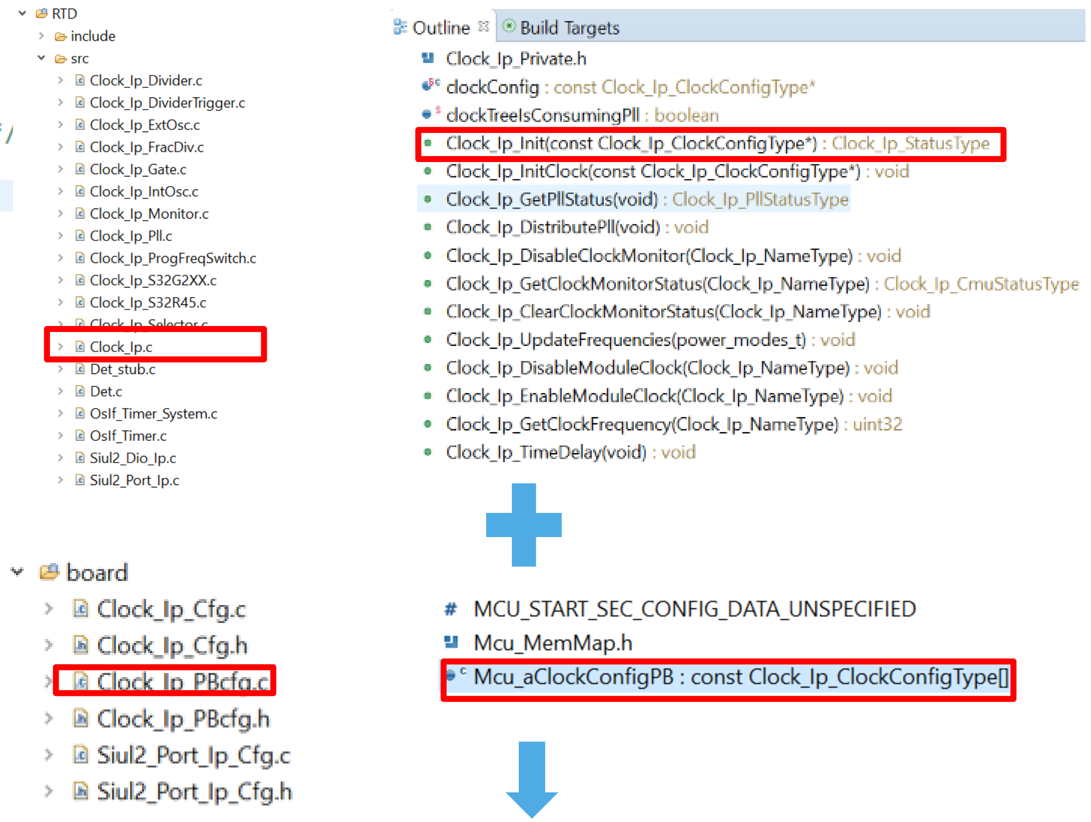


```
board
  Clock_Ip_Cfg_Defines.h
  Clock_Ip_Cfg.c
  Clock_Ip_Cfg.h
  Clock_Ip_PBcfg.c
  Clock_Ip_PBcfg.h
  Siul2_Port_Ip_Cfg.c
  Siul2_Port_Ip_Cfg.h
```

1

```c
/* Including necessary configuration files. */
#include "Mcal.h"

#include "Clock_Ip.h"
#include "Siul2_Port_Ip.h"
#include "Siul2_Dio_Ip.h"
```

- Initialize clocks

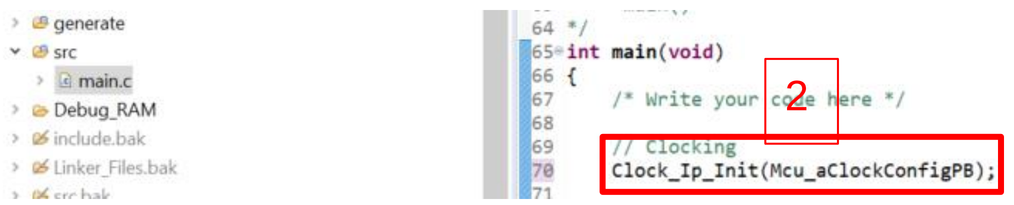


```
RTD
  include
  src
    Clock_Ip_Divider.c
    Clock_Ip_DividerTrigger.c
    Clock_Ip_ExtOsc.c
    Clock_Ip_FracDiv.c
    Clock_Ip_Gate.c
    Clock_Ip_IntOsc.c
    Clock_Ip_Monitor.c
    Clock_Ip_Pll.c
    Clock_Ip_ProgFreqSwitch.c
    Clock_Ip_S32G2XX.c
    Clock_Ip_S32R45.c
    Clock_Ip_Selector.c
    Clock_Ip.c
    Det_stub.c
    Det.c
    OsIf_Timer_System.c
    OsIf_Timer.c
    Siul2_Dio_Ip.c
    Siul2_Port_Ip.c
```

```
Outline   Build Targets
Clock_Ip_Private.h
clockConfig : const Clock_Ip_ClockConfigType*
clockTreeIsConsumingPll : boolean
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
Clock_Ip_DisableClockMonitor(Clock_Ip_NameType) : void
Clock_Ip_GetClockMonitorStatus(Clock_Ip_NameType) : Clock_Ip_CmuStatusType
Clock_Ip_ClearClockMonitorStatus(Clock_Ip_NameType) : void
Clock_Ip_UpdateFrequencies(power_modes_t) : void
Clock_Ip_DisableModuleClock(Clock_Ip_NameType) : void
Clock_Ip_EnableModuleClock(Clock_Ip_NameType) : void
Clock_Ip_GetClockFrequency(Clock_Ip_NameType) : uint32
Clock_Ip_TimeDelay(void) : void
```

```
board
  Clock_Ip_Cfg.c
  Clock_Ip_Cfg.h
  Clock_Ip_PBcfg.c
  Clock_Ip_PBcfg.h
  Siul2_Port_Ip_Cfg.c
  Siul2_Port_Ip_Cfg.h
```

```
MCU_START_SEC_CONFIG_DATA_UNSPECIFIED
Mcu_MemMap.h
Mcu_aClockConfigPB : const Clock_Ip_ClockConfigType[]
```



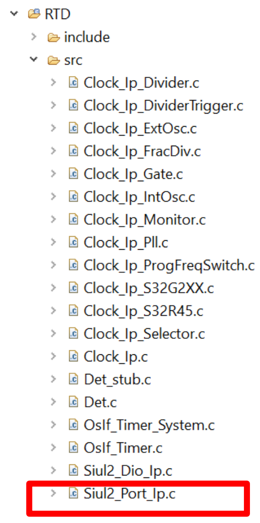
```
generate
src
  main.c
Debug_RAM
include.bak
Linker_Files.bak
src.bak
```

2

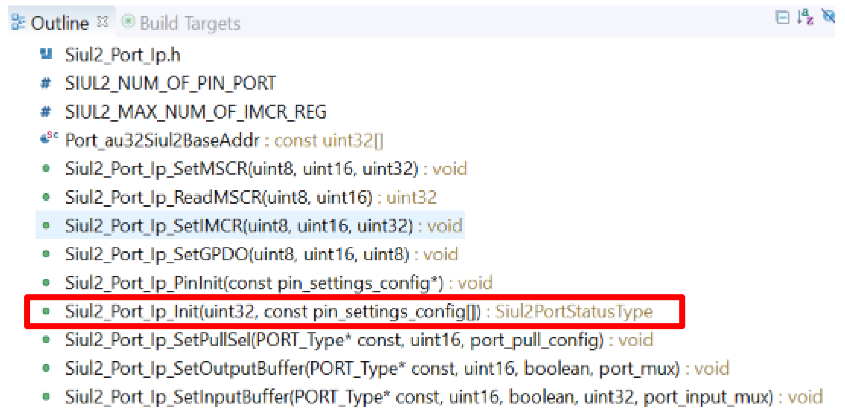
```c
*/
int main(void)
{
    /* Write your code here */

    // Clocking
    Clock_Ip_Init(Mcu_aClockConfigPB);
```

LIGHT UP RGB LED: APPLICATION CODE

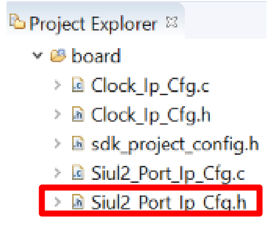
- Initialize pins



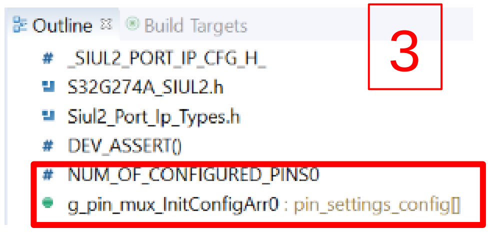
```
RTD
  include
  src
    Clock_Ip_Divider.c
    Clock_Ip_DividerTrigger.c
    Clock_Ip_ExtOsc.c
    Clock_Ip_FracDiv.c
    Clock_Ip_Gate.c
    Clock_Ip_IntOsc.c
    Clock_Ip_Monitor.c
    Clock_Ip_Pll.c
    Clock_Ip_ProgFreqSwitch.c
    Clock_Ip_S32G2XX.c
    Clock_Ip_S32R45.c
    Clock_Ip_Selector.c
    Clock_Ip.c
    Det_stub.c
    Det.c
    OsIf_Timer_System.c
    OsIf_Timer.c
    Siul2_Dio_Ip.c
    Siul2_Port_Ip.c
```



```
Outline   Build Targets
  Siul2_Port_Ip.h
  SIUL2_NUM_OF_PIN_PORT
  SIUL2_MAX_NUM_OF_IMCR_REG
  Port_au32Siul2BaseAddr : const uint32[]
  Siul2_Port_Ip_SetMSCR(uint8, uint16, uint32) : void
  Siul2_Port_Ip_ReadMSCR(uint8, uint16) : uint32
  Siul2_Port_Ip_SetIMCR(uint8, uint16, uint32) : void
  Siul2_Port_Ip_SetGPDO(uint8, uint16, uint8) : void
  Siul2_Port_Ip_PinInit(const pin_settings_config*) : void
  Siul2_Port_Ip_Init(uint32, const pin_settings_config[]) : Siul2PortStatusType
  Siul2_Port_Ip_SetPullSel(PORT_Type* const, uint16, port_pull_config) : void
  Siul2_Port_Ip_SetOutputBuffer(PORT_Type* const, uint16, boolean, port_mux) : void
  Siul2_Port_Ip_SetInputBuffer(PORT_Type* const, uint16, boolean, uint32, port_input_mux) : void
```

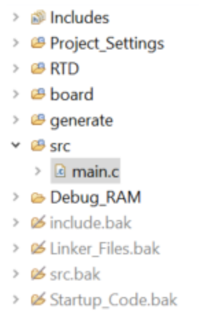


```
Project Explorer
  board
    Clock_Ip_Cfg.c
    Clock_Ip_Cfg.h
    sdk_project_config.h
    Siul2_Port_Ip_Cfg.c
    Siul2_Port_Ip_Cfg.h
```

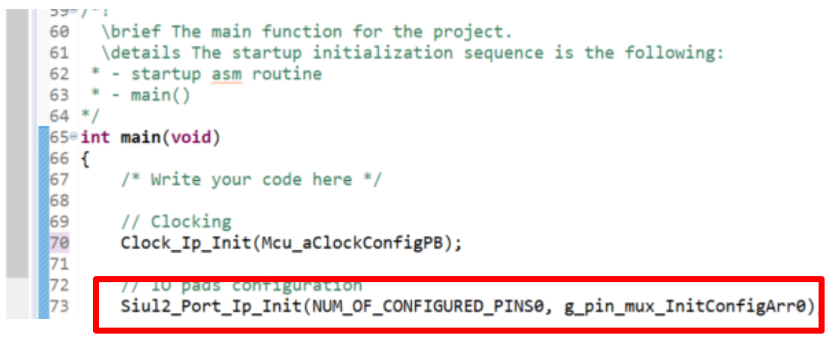


```
Outline   Build Targets
  _SIUL2_PORT_IP_CFG_H_
  S32G274A_SIUL2.h
  Siul2_Port_Ip_Types.h
  DEV_ASSERT()
  NUM_OF_CONFIGURED_PINS0
  g_pin_mux_InitConfigArr0 : pin_settings_config[]
```

3

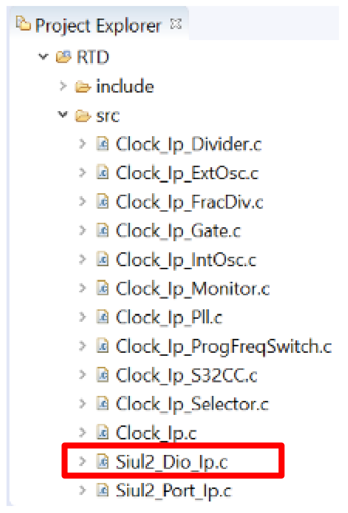


```
Includes
Project_Settings
RTD
board
generate
src
  main.c
Debug_RAM
include.bak
Linker_Files.bak
src.bak
Startup_Code.bak
```

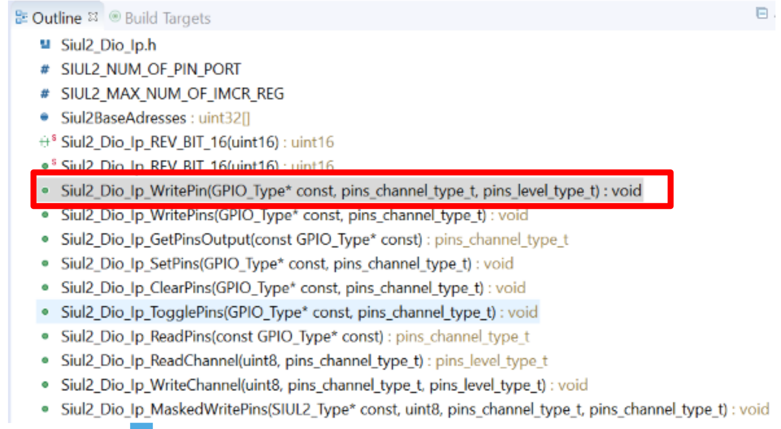


```
    \brief The main function for the project.
    \details The startup initialization sequence is the following:
 * - startup asm routine
 * - main()
 */
int main(void)
{
    /* Write your code here */

    // Clocking
    Clock_Ip_Init(Mcu_aClockConfigPB);

    // IO pads configuration
    Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0)
```

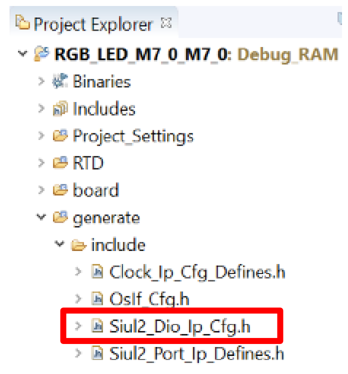
- Add the implementation of lighting up LED



```
Project Explorer
  RTD
    include
    src
      Clock_Ip_Divider.c
      Clock_Ip_ExtOsc.c
      Clock_Ip_FracDiv.c
      Clock_Ip_Gate.c
      Clock_Ip_IntOsc.c
      Clock_Ip_Monitor.c
      Clock_Ip_Pll.c
      Clock_Ip_ProgFreqSwitch.c
      Clock_Ip_S32CC.c
      Clock_Ip_Selector.c
      Clock_Ip.c
      Siul2_Dio_Ip.c
      Siul2_Port_Ip.c
```

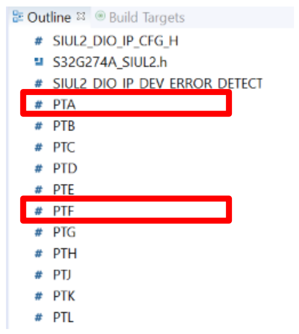


```
Outline   Build Targets
  Siul2_Dio_Ip.h
  SIUL2_NUM_OF_PIN_PORT
  SIUL2_MAX_NUM_OF_IMCR_REG
  Siul2BaseAdresses : uint32[]
  Siul2_Dio_Ip_REV_BIT_16(uint16) : uint16
  Siul2_Dio_Ip_REV_BIT_16(uint16) : uint16
  Siul2_Dio_Ip_WritePin(GPIO_Type* const, pins_channel_type_t, pins_level_type_t) : void
  Siul2_Dio_Ip_WritePins(GPIO_Type* const, pins_channel_type_t) : void
  Siul2_Dio_Ip_GetPinsOutput(const GPIO_Type* const) : pins_channel_type_t
  Siul2_Dio_Ip_SetPins(GPIO_Type* const, pins_channel_type_t) : void
  Siul2_Dio_Ip_ClearPins(GPIO_Type* const, pins_channel_type_t) : void
  Siul2_Dio_Ip_TogglePins(GPIO_Type* const, pins_channel_type_t) : void
  Siul2_Dio_Ip_ReadPins(const GPIO_Type* const) : pins_channel_type_t
  Siul2_Dio_Ip_ReadChannel(uint8, pins_channel_type_t) : pins_level_type_t
  Siul2_Dio_Ip_WriteChannel(uint8, pins_channel_type_t, pins_level_type_t) : void
  Siul2_Dio_Ip_MaskedWritePins(SIUL2_Type* const, uint8, pins_channel_type_t, pins_channel_type_t) : void
```

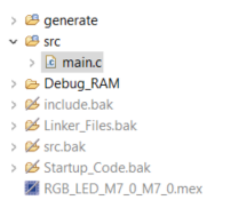


```
Project Explorer
  RGB_LED_M7_0_M7_0: Debug_RAM
    Binaries
    Includes
    Project_Settings
    RTD
    board
    generate
      include
        Clock_Ip_Cfg_Defines.h
        OsIf_Cfg.h
        Siul2_Dio_Ip_Cfg.h
        Siul2_Port_Ip_Defines.h
```

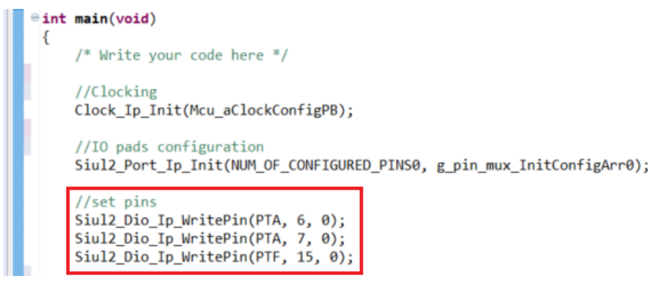
4



```
Outline   Build Targets
  SIUL2_DIO_IP_CFG_H
  S32G274A_SIUL2.h
  SIUL2_DIO_IP_DEV_ERROR_DETECT
  PTA
  PTB
  PTC
  PTD
  PTE
  PTF
  PTG
  PTH
  PTJ
  PTK
  PTL
```



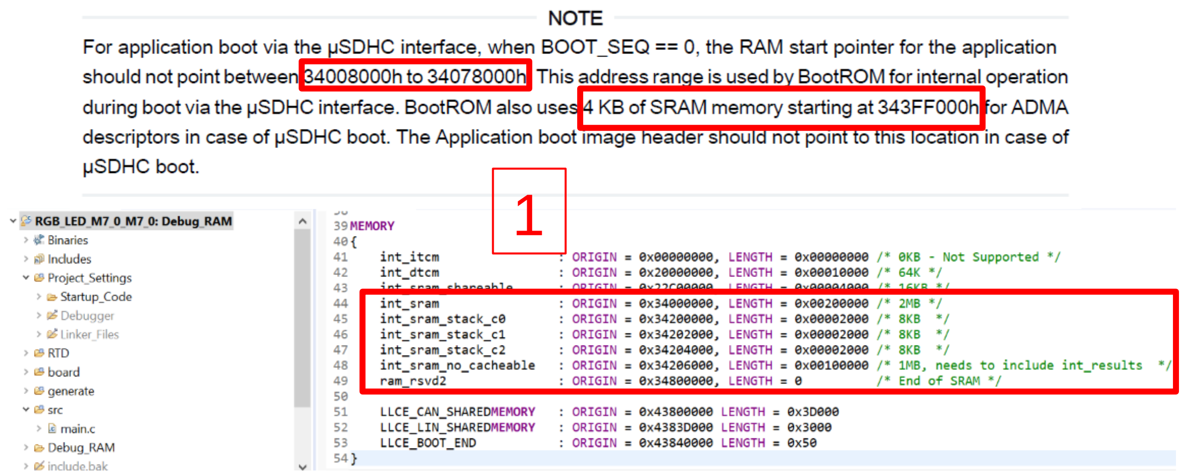
```
generate
src
  main.c
Debug_RAM
include.bak
Linker_Files.bak
src.bak
Startup_Code.bak
RGB_LED_M7_0_M7_0.mex
```



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

    //Clocking
    Clock_Ip_Init(Mcu_aClockConfigPB);

    //IO pads configuration
    Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0);

    //set pins
    Siul2_Dio_Ip_WritePin(PTA, 6, 0);
    Siul2_Dio_Ip_WritePin(PTA, 7, 0);
    Siul2_Dio_Ip_WritePin(PTF, 15, 0);
```

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

- Open and modify the link file according to the noted information which is from reference manual of S32G

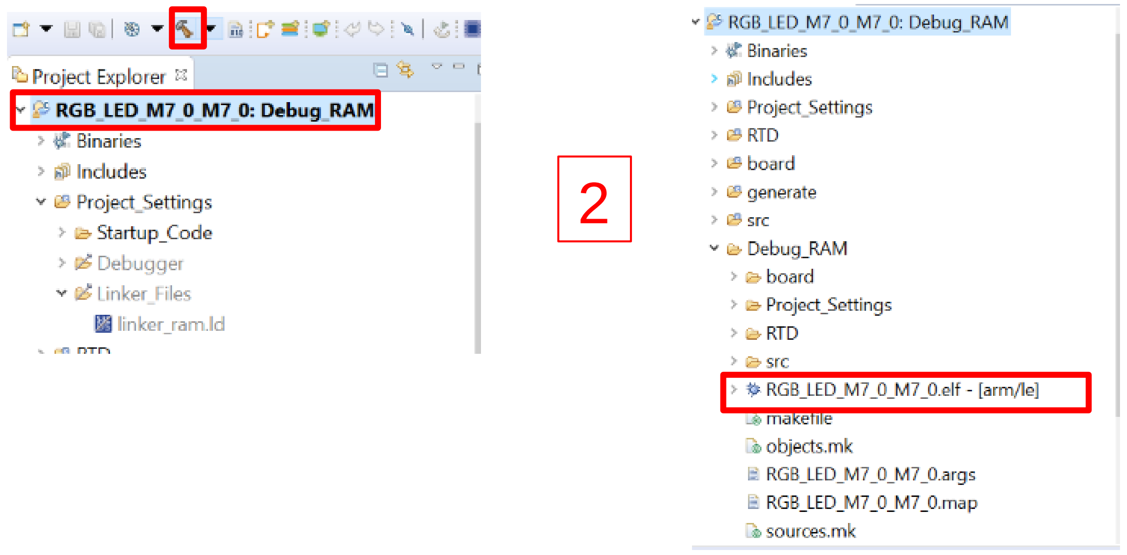


NOTE

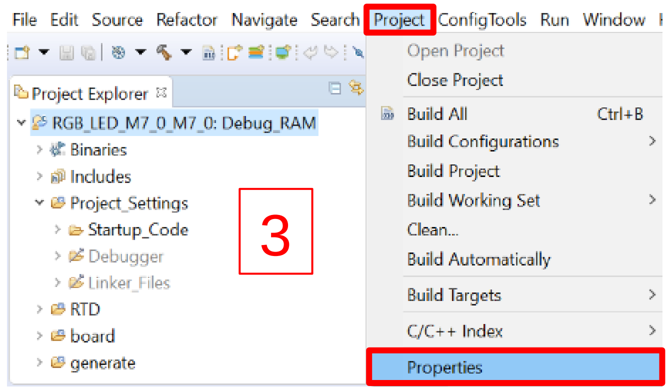
For application boot via the µSDHC interface, when BOOT_SEQ == 0, the RAM start pointer for the application should not point between 34008000h to 34078000h. This address range is used by BootROM for internal operation during boot via the µSDHC interface. BootROM also uses 4 KB of SRAM memory starting at 343FF000h for ADMA descriptors in case of µSDHC boot. The Application boot image header should not point to this location in case of µSDHC boot.

```
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 = 0x34000000, LENGTH = 0x00200000 /* 2MB */
    int_sram_stack_c0       : ORIGIN = 0x34200000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_stack_c1       : ORIGIN = 0x34202000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_stack_c2       : ORIGIN = 0x34204000, LENGTH = 0x00002000 /* 8KB  */
    int_sram_no_cacheable   : ORIGIN = 0x34206000, LENGTH = 0x00100000 /* 1MB, needs to include int_results  */
    ram_rsvd2               : ORIGIN = 0x34800000, LENGTH = 0          /* End of SRAM */

    LLCE_CAN_SHAREDMEMORY   : ORIGIN = 0x43800000 LENGTH = 0x3D000
    LLCE_LIN_SHAREDMEMORY   : ORIGIN = 0x4383D000 LENGTH = 0x3000
    LLCE_BOOT_END           : ORIGIN = 0x43840000 LENGTH = 0x50
}
```

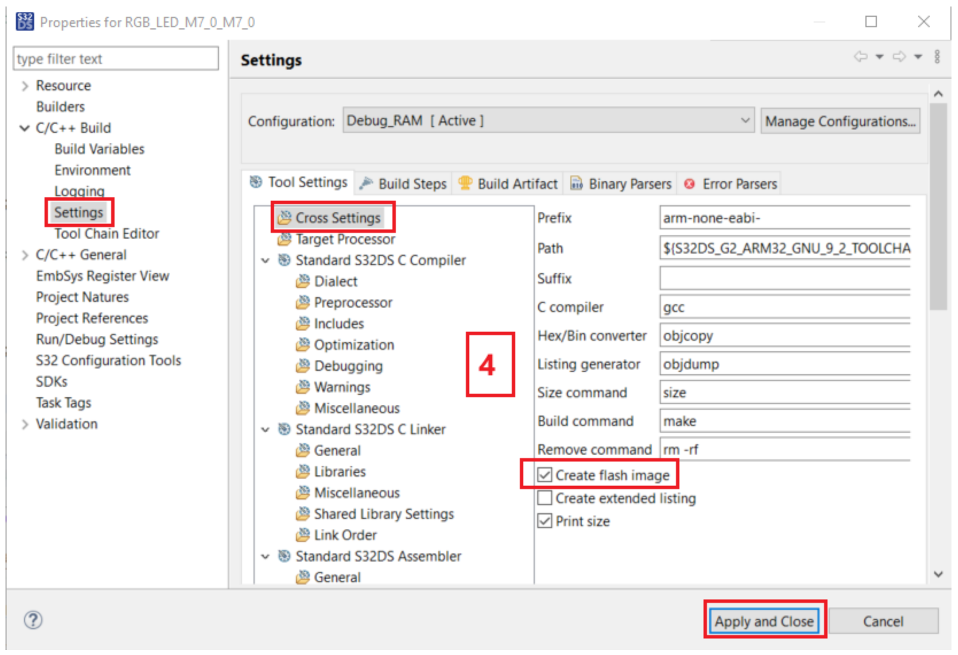
- Select and build project, .elf file will be generated



- Open the properties of project



- Check the "Create flash image" box and click on "OK"



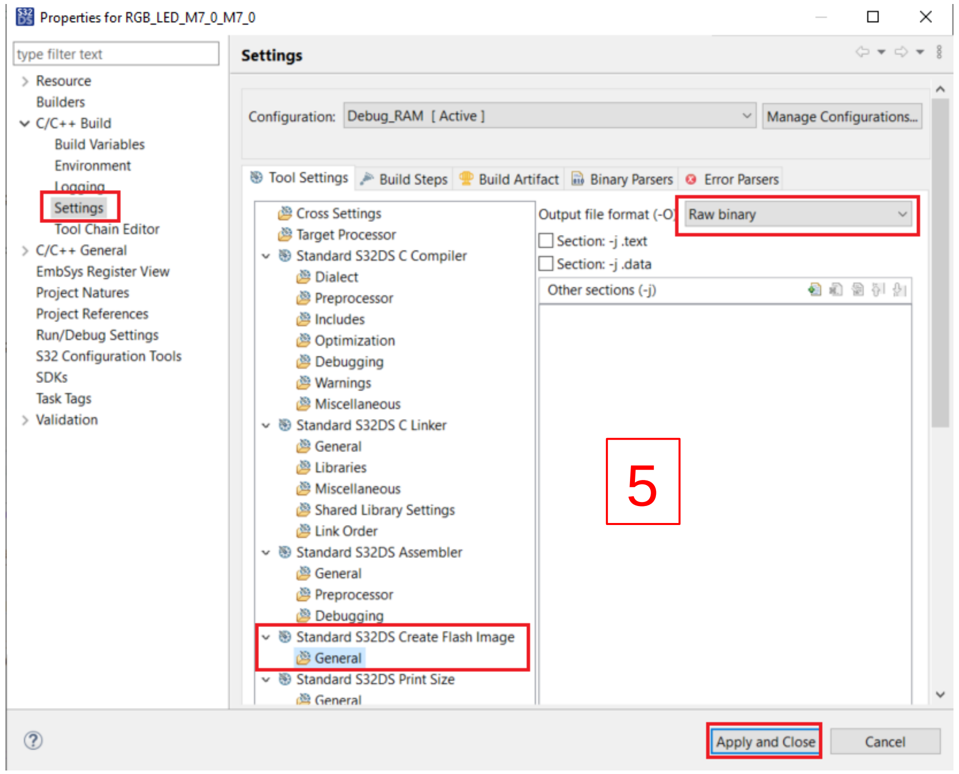
Configuration: Debug_RAM [Active]

Prefix	arm-none-eabi-
Path	$(S32DS_G2_ARM32_GNU_9_2_TOOLCHA
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 "OK"



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
- 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

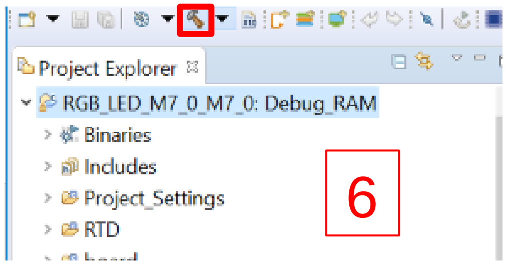
☐ Section: -j .text

☐ Section: -j .data

Other sections (-j)

Apply and Close   Cancel

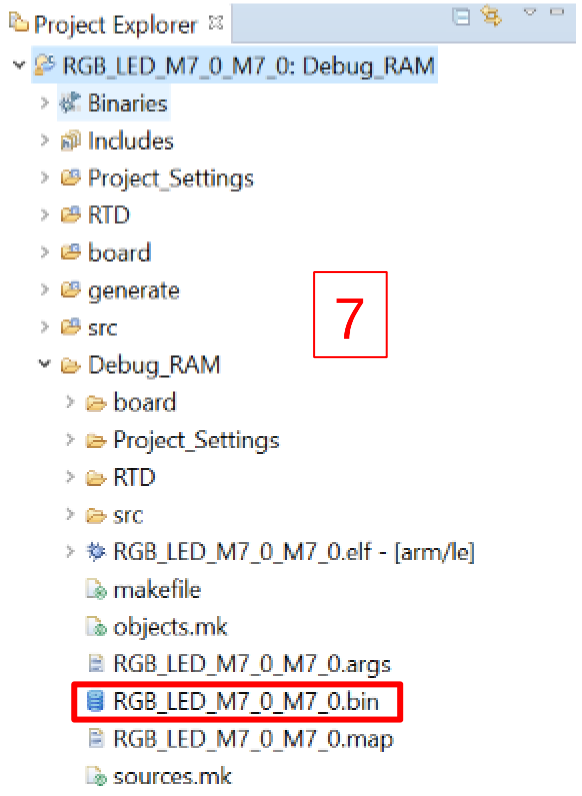
- Re-build the project



Project Explorer

- RGB_LED_M7_0_M7_0: Debug_RAM
  - Binaries
  - Includes
  - Project_Settings
  - RTD

- The .bin file has been generated

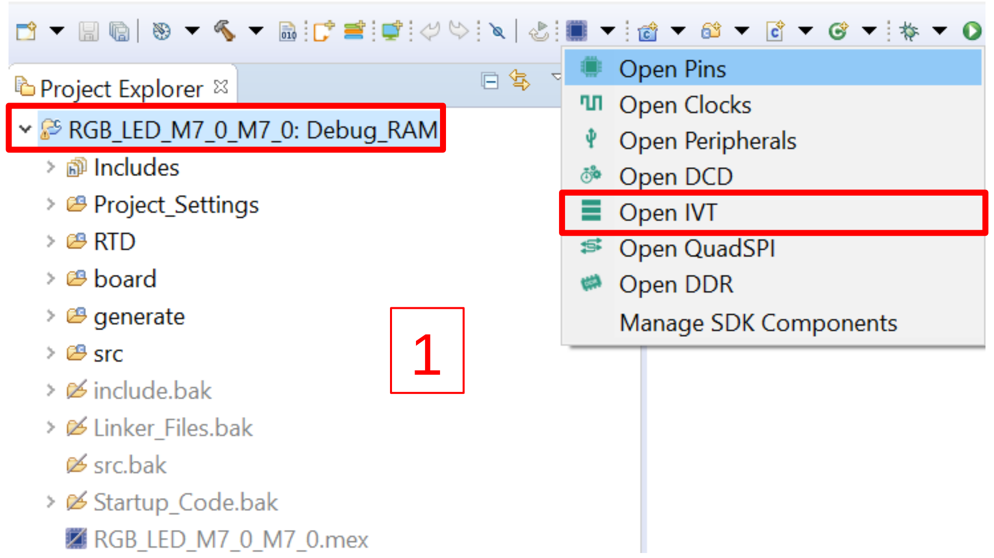


Project Explorer

- RGB_LED_M7_0_M7_0: Debug_RAM
  - Binaries
  - Includes
  - Project_Settings
  - RTD
  - board
  - generate
  - src
  - Debug_RAM
    - board
    - 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



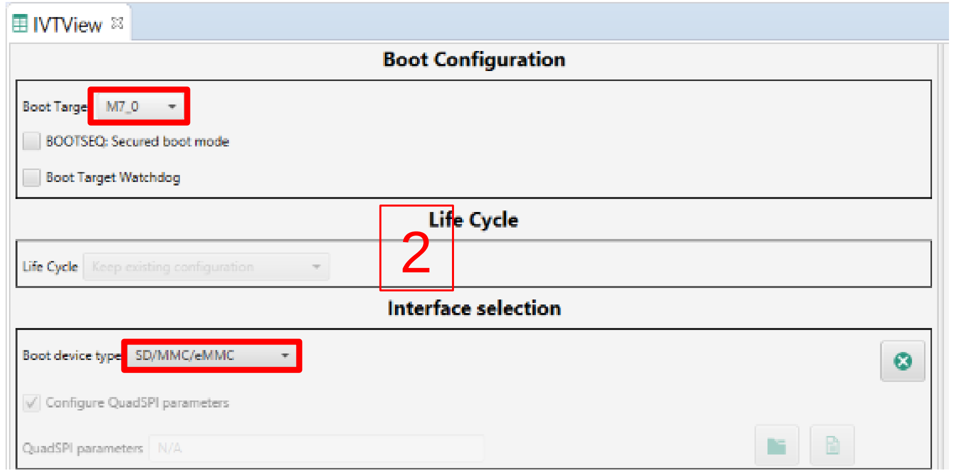
1

Project Explorer

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

Open Pins
Open Clocks
Open Peripherals
Open DCD
Open IVT
Open QuadSPI
Open DDR
Manage SDK Components

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



2

IVTView

Boot Configuration

Boot Targe: M7_0

BOOTSEQ: Secured boot mode

Boot Target Watchdog

Life Cycle

Life Cycle: Keep existing configuration

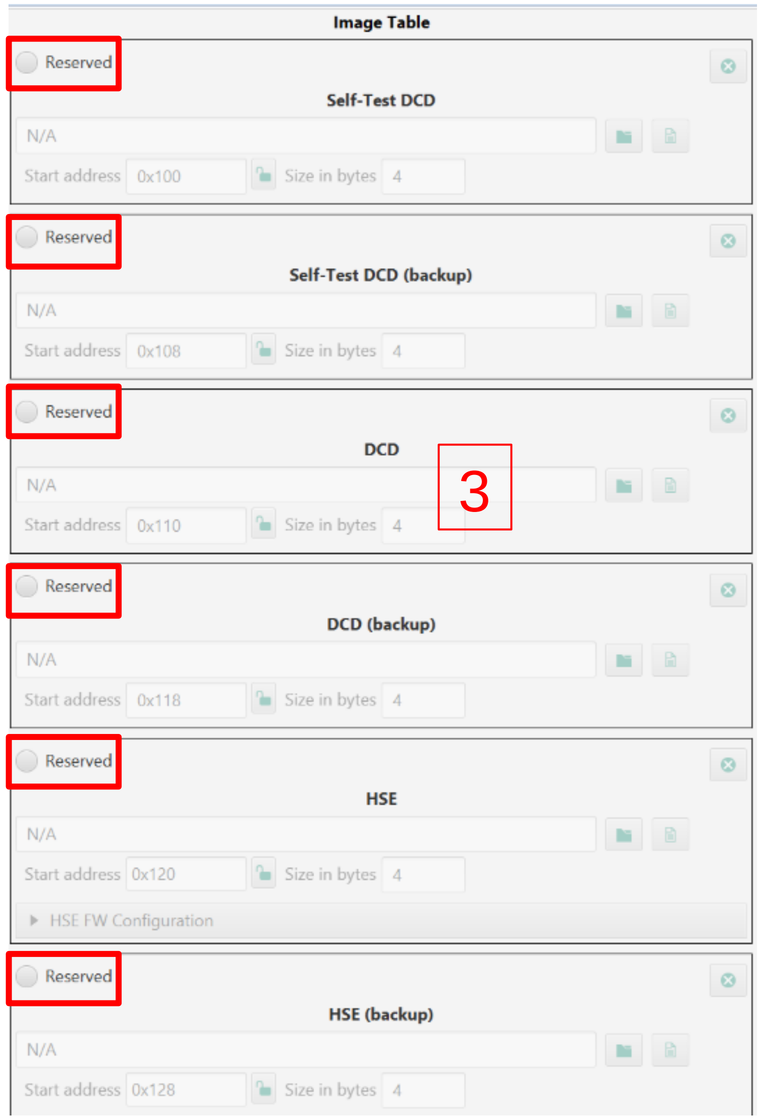
Interface selection

Boot device type: SD/MMC/eMMC

✓ Configure QuadSPI parameters

QuadSPI parameters: N/A

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



3

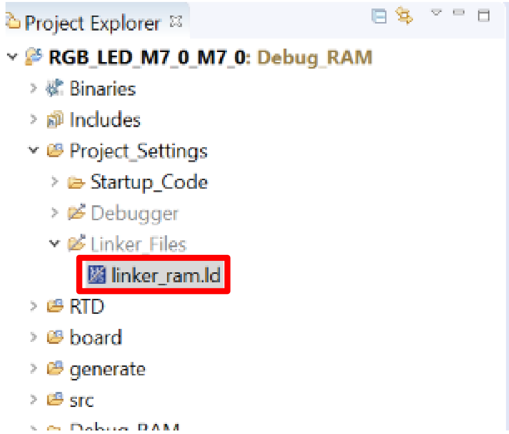
Image Table

Reserved	Entry	Path	Start address	Size in bytes
Reserved	Self-Test DCD	N/A	0x100	4
Reserved	Self-Test DCD (backup)	N/A	0x108	4
Reserved	DCD	N/A	0x110	4
Reserved	DCD (backup)	N/A	0x118	4
Reserved	HSE	N/A	0x120	4
Reserved	HSE (backup)	N/A	0x128	4

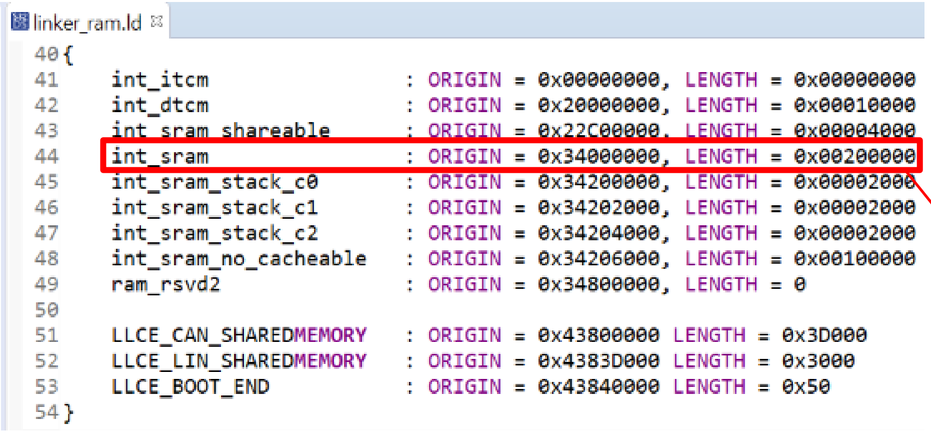
HSE FW Configuration

LIGHT UP RGB LED: MAKE IMAGE BY IVT TOOL

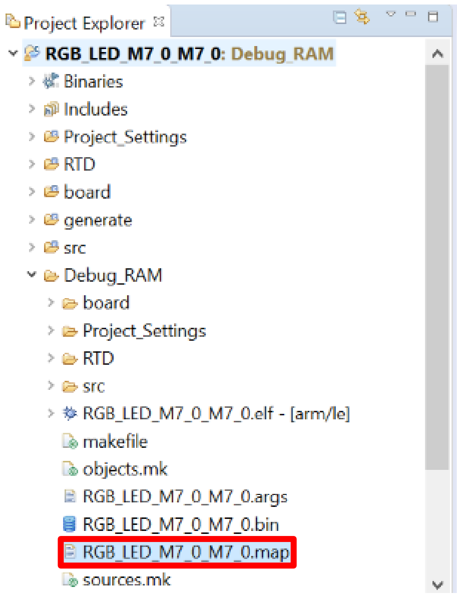
- Configure Application Boot Image according to .ld file and .map file


Project Explorer

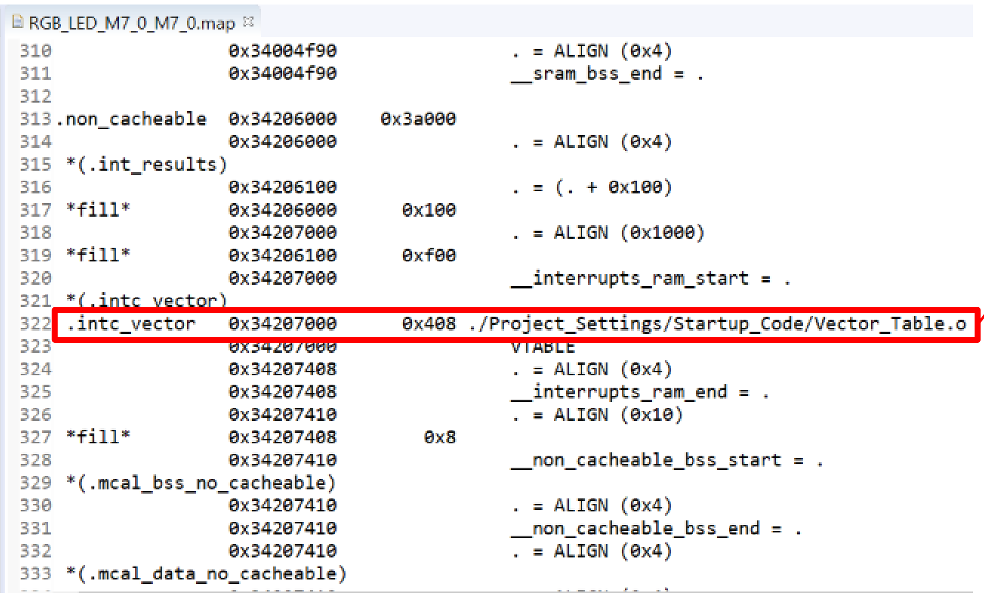
- RGB_LED_M7_0_M7_0: Debug_RAM
  - Binaries
  - Includes
  - Project_Settings
    - Startup_Code
    - Debugger
    - Linker_Files
      - linker_ram.ld
  - RTD
  - board
  - generate
  - src
  - Debug_RAM


linker_ram.ld

```
{
    int_itcm                : ORIGIN = 0x00000000, LENGTH = 0x00000000
    int_dtcm                : ORIGIN = 0x20000000, LENGTH = 0x00010000
    int_sram_shareable      : ORIGIN = 0x22C00000, LENGTH = 0x00004000
    int_sram                : ORIGIN = 0x34000000, LENGTH = 0x00200000
    int_sram_stack_c0       : ORIGIN = 0x34200000, LENGTH = 0x00002000
    int_sram_stack_c1       : ORIGIN = 0x34202000, LENGTH = 0x00002000
    int_sram_stack_c2       : ORIGIN = 0x34204000, LENGTH = 0x00002000
    int_sram_no_cacheable   : ORIGIN = 0x34206000, LENGTH = 0x00100000
    ram_rsvd2               : ORIGIN = 0x34800000, LENGTH = 0

    LLCE_CAN_SHAREDMEMORY   : ORIGIN = 0x43800000 LENGTH = 0x3D000
    LLCE_LIN_SHAREDMEMORY   : ORIGIN = 0x4383D000 LENGTH = 0x3000
    LLCE_BOOT_END           : ORIGIN = 0x43840000 LENGTH = 0x50
}
```

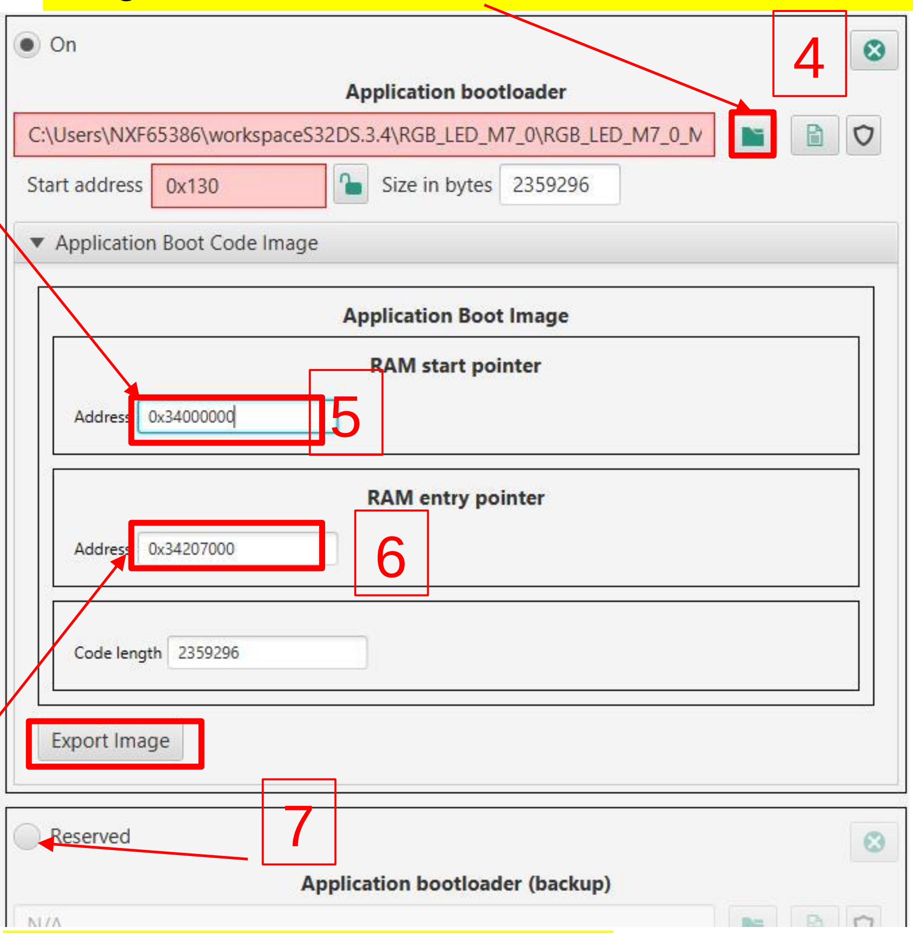

Project Explorer

- RGB_LED_M7_0_M7_0: Debug_RAM
  - Binaries
  - Includes
  - Project_Settings
  - RTD
  - board
  - generate
  - src
  - Debug_RAM
    - board
    - 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


RGB_LED_M7_0_M7_0.map

```
                 0x34004f90                . = ALIGN (0x4)
                 0x34004f90                __sram_bss_end = .

.non_cacheable   0x34206000    0x3a000
                 0x34206000                . = ALIGN (0x4)
 *(.int_results)
                 0x34206100                . = (. + 0x100)
 *fill*          0x34206000      0x100
                 0x34207000                . = ALIGN (0x1000)
 *fill*          0x34206100      0xf00
                 0x34207000                __interrupts_ram_start = .
 *(.intc_vector)
 .intc_vector    0x34207000      0x408 ./Project_Settings/Startup_Code/Vector_Table.o
                 0x34207000                VTABLE
                 0x34207408                . = ALIGN (0x4)
                 0x34207408                __interrupts_ram_end = .
                 0x34207410                . = ALIGN (0x10)
 *fill*          0x34207408        0x8
                 0x34207410                __non_cacheable_bss_start = .
 *(.mcal_bss_no_cacheable)
                 0x34207410                . = ALIGN (0x4)
                 0x34207410                __non_cacheable_bss_end = .
                 0x34207410                . = ALIGN (0x4)
 *(.mcal_data_no_cacheable)
```

Click browse to select .bin files generated from Page 20



● On

Application bootloader

C:\Users\NXF65386\workspaceS32DS.3.4\RGB_LED_M7_0\RGB_LED_M7_0_M

Start address 0x130 Size in bytes 2359296

▼ Application Boot Code Image

Application Boot Image

RAM start pointer

Address 0x34000000

RAM entry pointer

Address 0x34207000

Code length 2359296

Export Image

○ Reserved

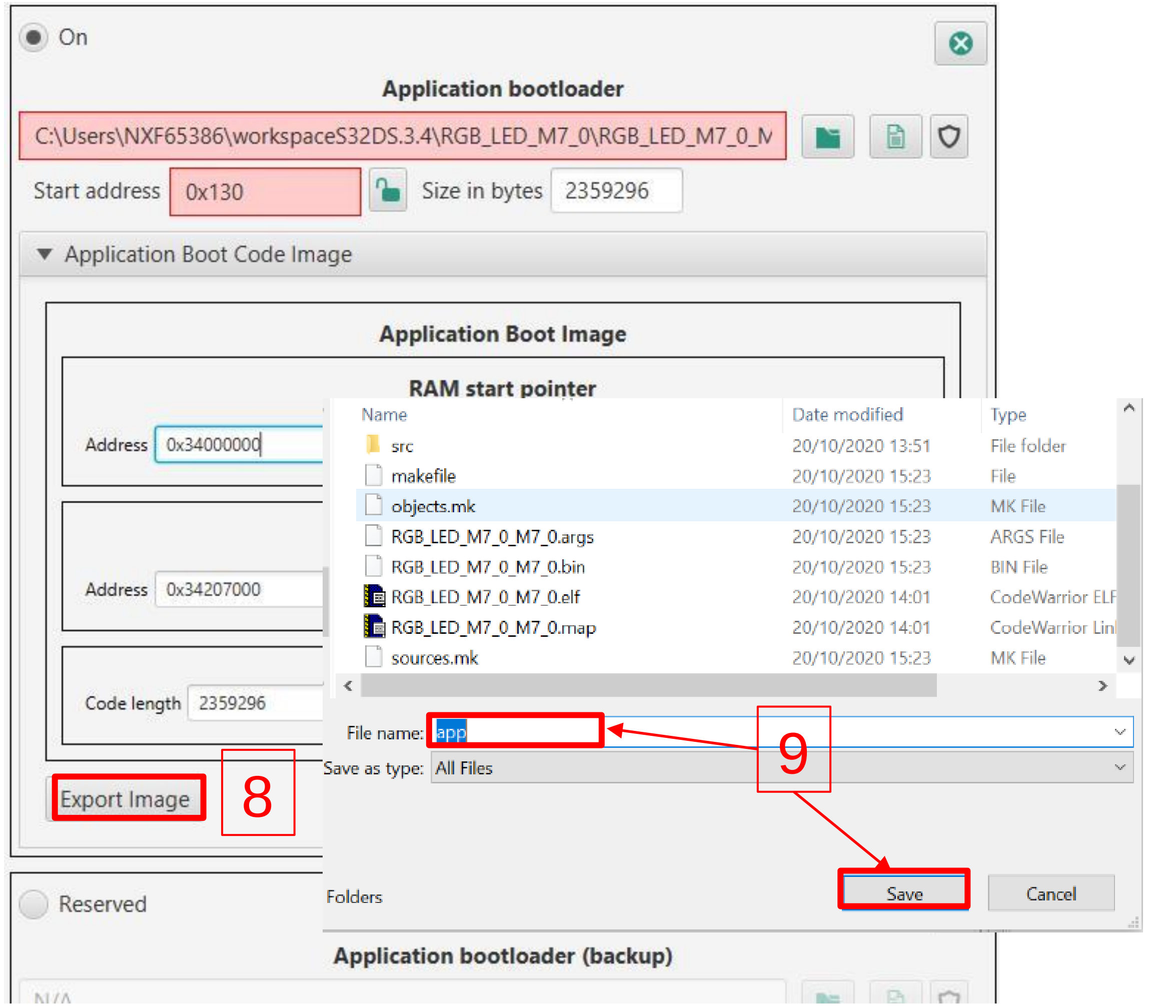
Application bootloader (backup)

N/A

Set the backup to be reserved

LIGHT UP RGB LED: MAKE IMAGE BY IVT TOOL

- Export and save image as any name



On

Application bootloader

C:\Users\NXF65386\workspaceS32DS.3.4\RGB_LED_M7_0\RGB_LED_M7_0_M

Start address: 0x130 — Size in bytes: 2359296

Application Boot Code Image

Application Boot Image

RAM start pointer

Address: 0x34000000

Address: 0x34207000

Code length: 2359296

Name	Date modified	Type
src	20/10/2020 13:51	File folder
makefile	20/10/2020 15:23	File
objects.mk	20/10/2020 15:23	MK File
RGB_LED_M7_0_M7_0.args	20/10/2020 15:23	ARGS File
RGB_LED_M7_0_M7_0.bin	20/10/2020 15:23	BIN File
RGB_LED_M7_0_M7_0.elf	20/10/2020 14:01	CodeWarrior ELF
RGB_LED_M7_0_M7_0.map	20/10/2020 14:01	CodeWarrior Lin
sources.mk	20/10/2020 15:23	MK File

File name: app

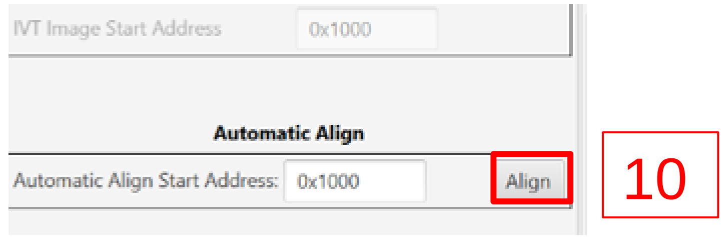
Save as type: All Files

Reserved

Application bootloader (backup)

N/A

- Click on “Align” to resolve error

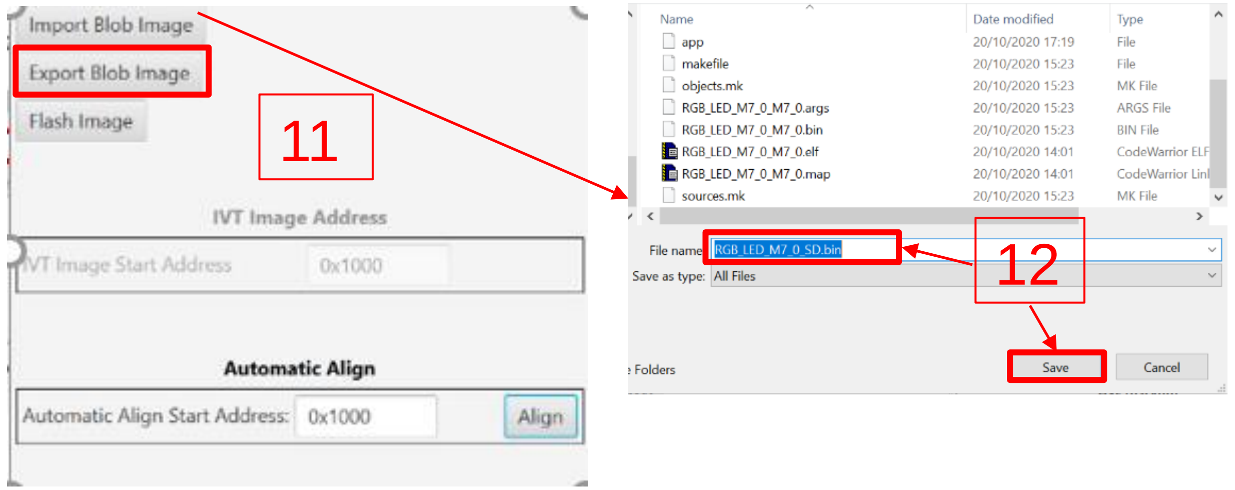


IVT Image Start Address: 0x1000

Automatic Align

Automatic Align Start Address: 0x1000

- Click on “Export Blob Image” to generate and save final image as any name



IVT Image Address

IVT Image Start Address: 0x1000

Automatic Align

Automatic Align Start Address: 0x1000

Name	Date modified	Type
app	20/10/2020 17:19	File
makefile	20/10/2020 15:23	File
objects.mk	20/10/2020 15:23	MK File
RGB_LED_M7_0_M7_0.args	20/10/2020 15:23	ARGS File
RGB_LED_M7_0_M7_0.bin	20/10/2020 15:23	BIN File
RGB_LED_M7_0_M7_0.elf	20/10/2020 14:01	CodeWarrior ELF
RGB_LED_M7_0_M7_0.map	20/10/2020 14:01	CodeWarrior Lin
sources.mk	20/10/2020 15:23	MK File

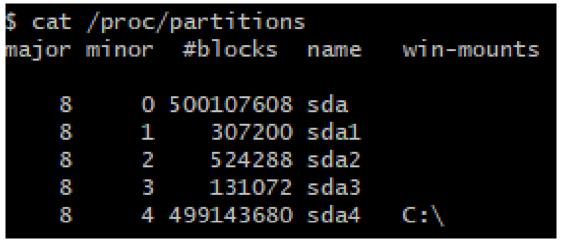
File name: RGB_LED_M7_0_SD.bin

Save as type: All Files

LIGHT UP RGB LED: DOWNLOAD IMAGE INTO SD CARD

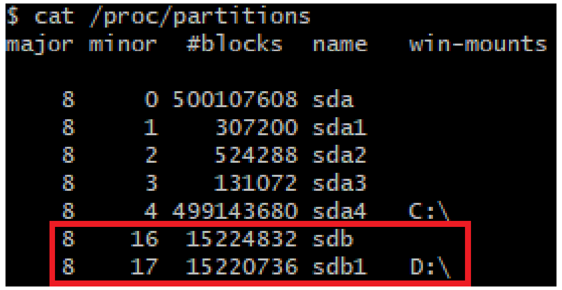
1. Install and Run Cygwin as administrator.

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:\
```

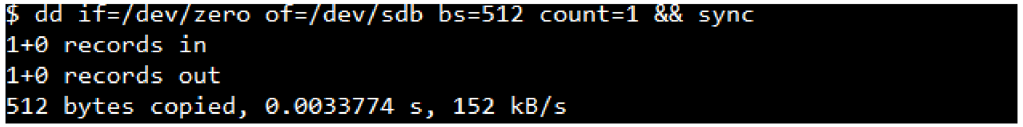
3. After inserting SD card into the slot, run “cat /proc/partitions” cmd again and find out the SD card descriptor



```
$ 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

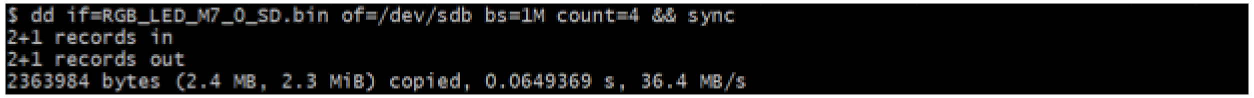
dd if=/dev/zero of=/dev/sdb bs=512 count=1 && sync



```
$ 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
```

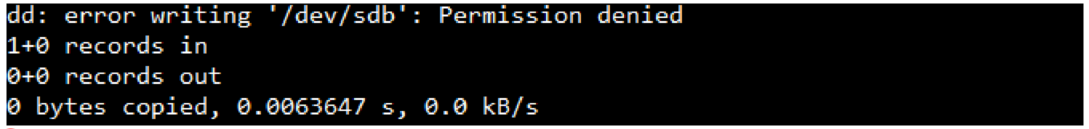
5. Run “cd path of bin file” cmd to find out image generated in page 26, download image into the SD card

dd if=RGB_LED_M7_0_SD.bin of=/dev/sdb bs=1M count=4 && sync



```
$ dd if=RGB_LED_M7_0_SD.bin of=/dev/sdb bs=1M count=4 && sync
2+1 records in
2+1 records out
2363984 bytes (2.4 MB, 2.3 MiB) copied, 0.0649369 s, 36.4 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
```

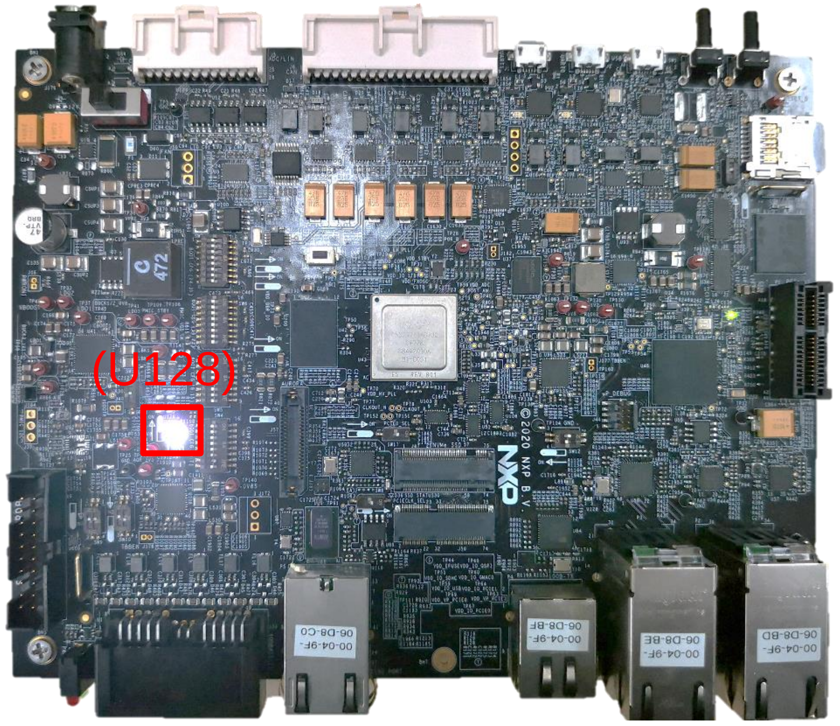
① Take out the SD card and insert it again
② Execute dd if=/dev/zero of=/dev/sdb bs=512 count=1 && sync
③ Take out the SD card and insert it again
④ Burn the image to the SD card

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 S32G pin(Refer to Quick Start Guide to find out the corresponding switch)

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 RDB2, the RGB LED (U128) will be lighted in white color



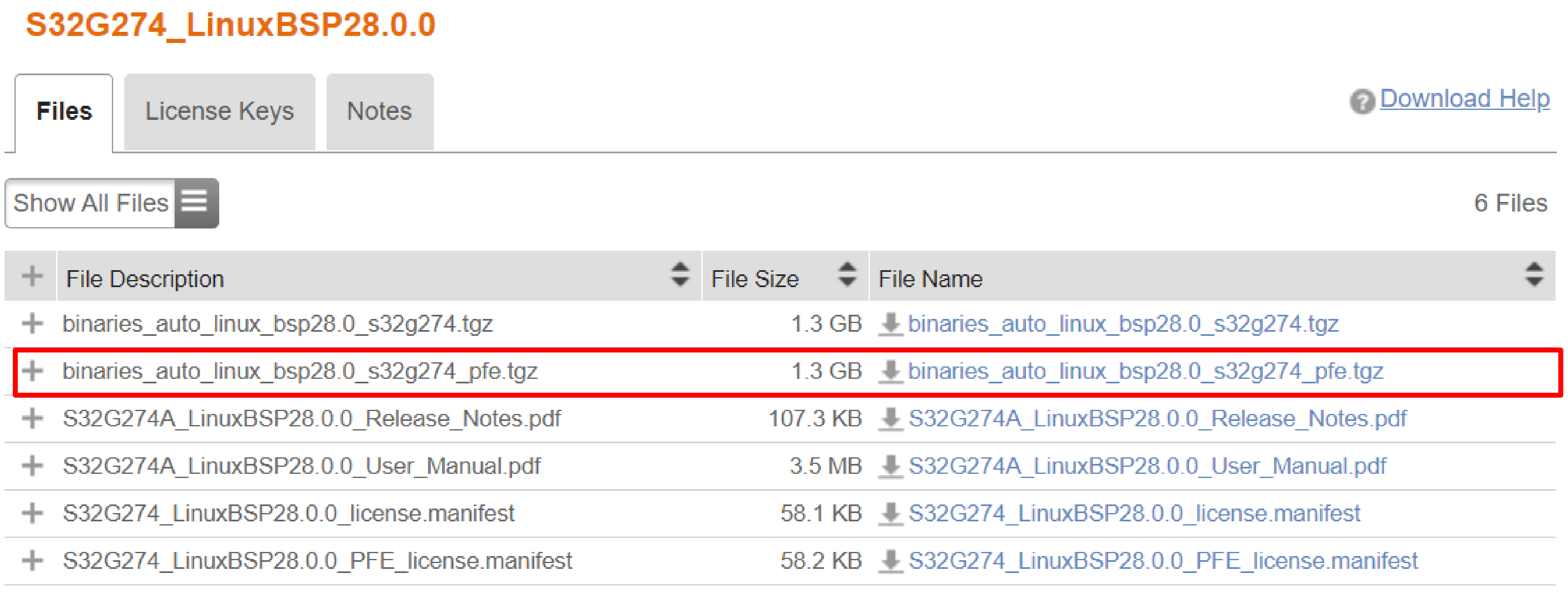
RUN Linux BSP On Cortex-A53 Cores

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

STEP 1: DOWNLOAD LINUX BSP FROM SOFTWARE CENTER

- Download the file which included PFE



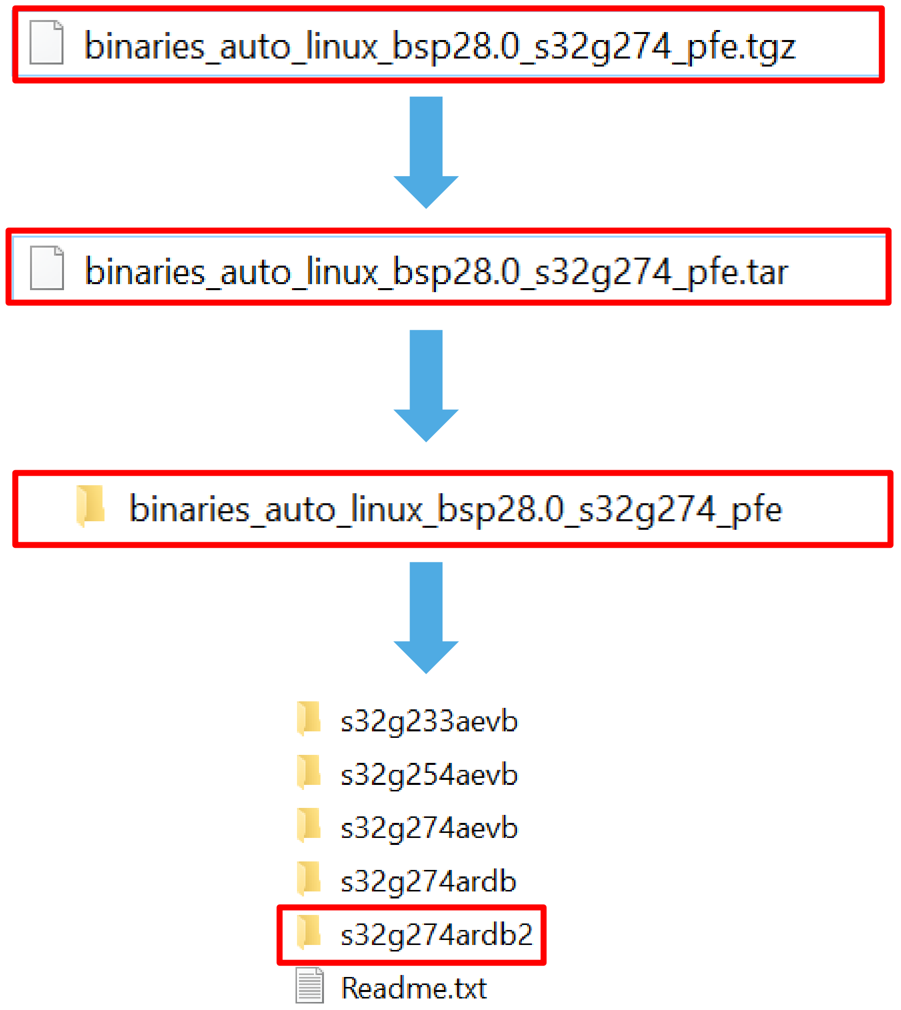
S32G274_LinuxBSP28.0.0

File Description	File Size	File Name
binaries_auto_linux_bsp28.0_s32g274.tgz	1.3 GB	binaries_auto_linux_bsp28.0_s32g274.tgz
binaries_auto_linux_bsp28.0_s32g274_pfe.tgz	1.3 GB	binaries_auto_linux_bsp28.0_s32g274_pfe.tgz
S32G274A_LinuxBSP28.0.0_Release_Notes.pdf	107.3 KB	S32G274A_LinuxBSP28.0.0_Release_Notes.pdf
S32G274A_LinuxBSP28.0.0_User_Manual.pdf	3.5 MB	S32G274A_LinuxBSP28.0.0_User_Manual.pdf
S32G274_LinuxBSP28.0.0_license.manifest	58.1 KB	S32G274_LinuxBSP28.0.0_license.manifest
S32G274_LinuxBSP28.0.0_PFE_license.manifest	58.2 KB	S32G274_LinuxBSP28.0.0_PFE_license.manifest

Note: If need more information for building BSP, refer to S32G274_Linux_BSP_28.0.0_User_Manual.pdf

STEP 1: DOWNLOAD LINUX BSP FROM SOFTWARE CENTRE

- Unzip and untar the download file by 7-zip

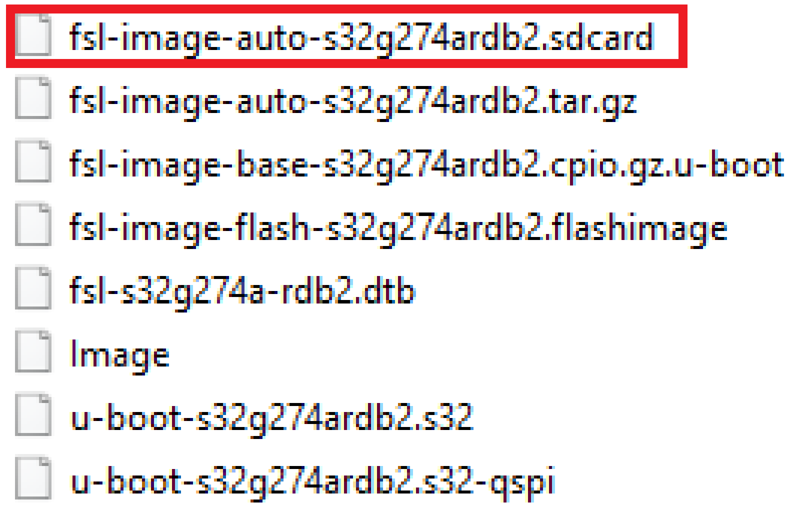

binaries_auto_linux_bsp28.0_s32g274_pfe.tgz

binaries_auto_linux_bsp28.0_s32g274_pfe.tar

binaries_auto_linux_bsp28.0_s32g274_pfe

s32g233aevb
s32g254aevb
s32g274aevb
s32g274ardb
s32g274ardb2
Readme.txt

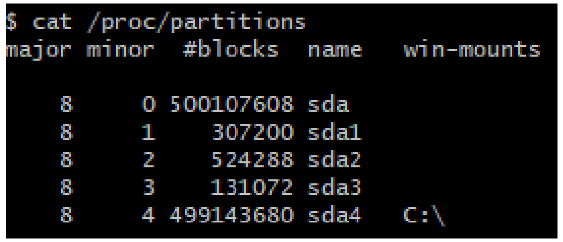
- Find the .sdcard file in the s32g274ardb2 folder


fsl-image-auto-s32g274ardb2.sdcard
fsl-image-auto-s32g274ardb2.tar.gz
fsl-image-base-s32g274ardb2.cpio.gz.u-boot
fsl-image-flash-s32g274ardb2.flashimage
fsl-s32g274a-rdb2.dtb
Image
u-boot-s32g274ardb2.s32
u-boot-s32g274ardb2.s32-qspi

Note: The .sdcard file can be loaded into SD or eMMC. Refer to next step to know loading image

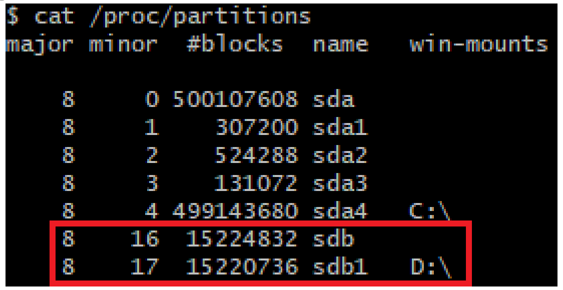
STEP 2: DOWNLOAD BSP IMAGE INTO SD CARD

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.



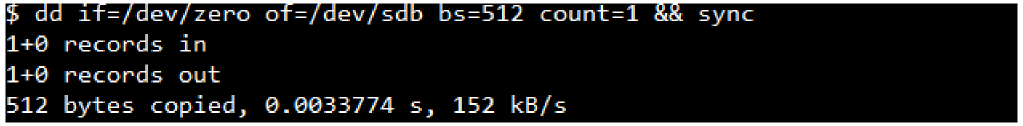
```
$ cat /proc/partitions
major minor  #blocks  name   win-mounts

   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



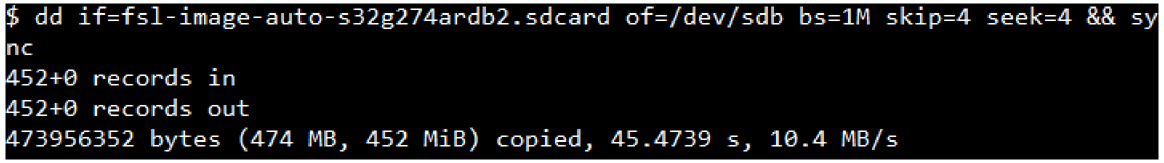
```
$ cat /proc/partitions
major minor  #blocks  name   win-mounts

   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



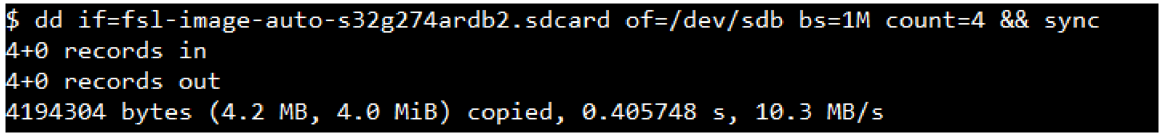
```
$ dd if=/dev/zero of=/dev/sdb bs=512 count=1 && sync
1+0 records in
1+0 records out
512 bytes copied, 0.0033774 s, 152 kB/s
```

5. Burn all contents of the BSP image but the first four mega bytes into the SD card
   dd if=fsl-image-auto-s32g274ardb2.sdcard of=/dev/sdb bs=1M skip=4 seek=4 && sync



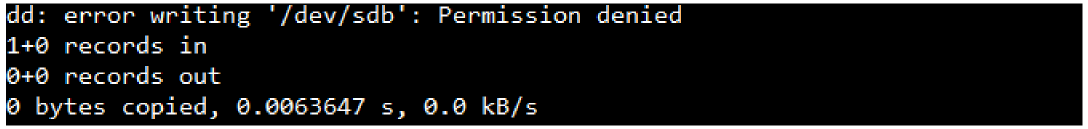
```
$ dd if=fsl-image-auto-s32g274ardb2.sdcard of=/dev/sdb bs=1M skip=4 seek=4 && sy
nc
452+0 records in
452+0 records out
473956352 bytes (474 MB, 452 MiB) copied, 45.4739 s, 10.4 MB/s
```

6. Burn the first four mega bytes of the BSP image into the SD card
   dd if=fsl-image-auto-s32g274ardb2.sdcard of=/dev/sdb bs=1M count=4 && sync



```
$ dd if=fsl-image-auto-s32g274ardb2.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, 0.405748 s, 10.3 MB/s
```

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

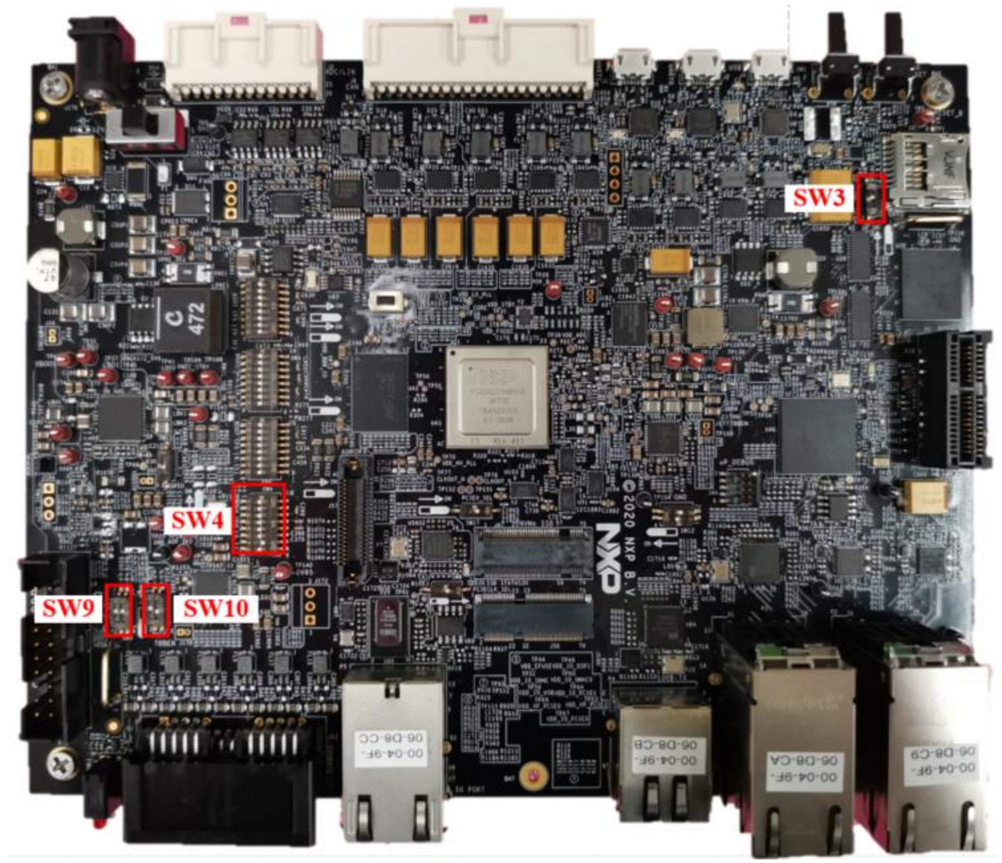


```
dd: error writing '/dev/sdb': Permission denied
1+0 records in
0+0 records out
0 bytes copied, 0.0063647 s, 0.0 kB/s
```

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

STEP 3: SELECT BOOT MODE OF RDB2

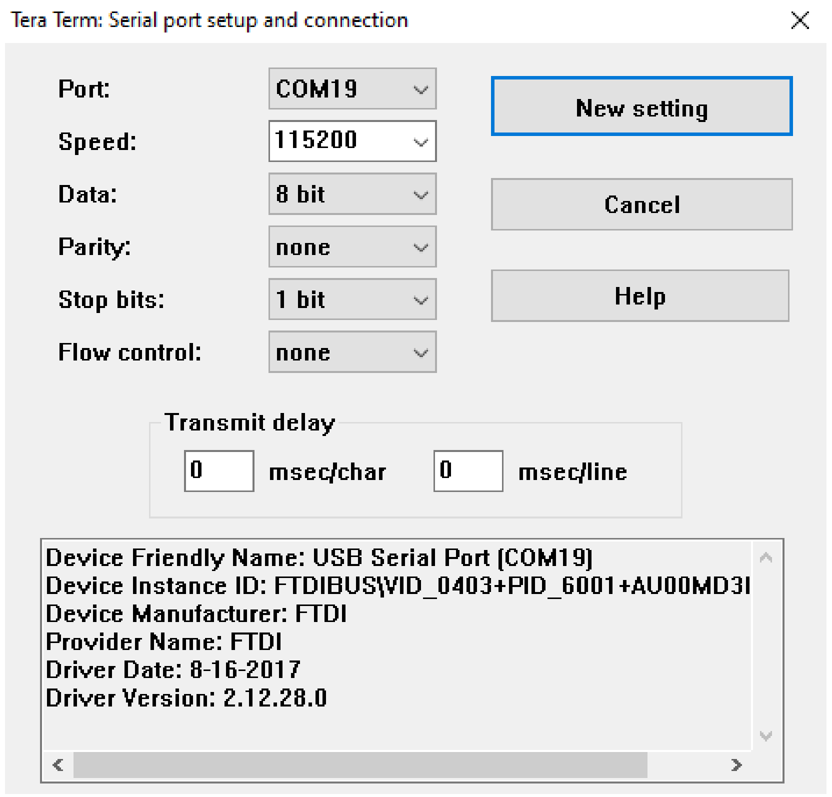
- Set RDB2 to SD card boot mode



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

STEP 4: RUN LINUX BSP

- Connect UART cable to UART0. Then open serial terminal and configure COM port



Tera Term: Serial port setup and connection

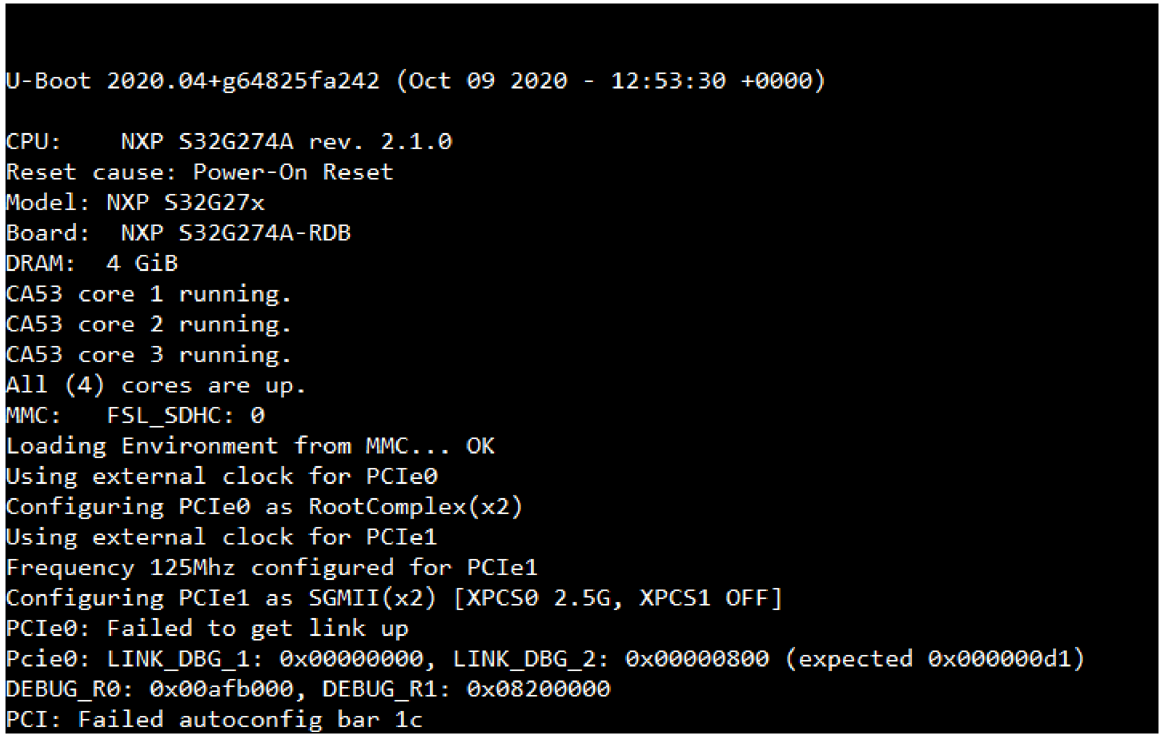
Port:	COM19
Speed:	115200
Data:	8 bit
Parity:	none
Stop bits:	1 bit
Flow control:	none

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-RDB2 and view print message in serial terminal



```
U-Boot 2020.04+g64825fa242 (Oct 09 2020 - 12:53:30 +0000)

CPU:    NXP S32G274A rev. 2.1.0
Reset cause: Power-On Reset
Model: NXP S32G27x
Board:  NXP S32G274A-RDB
DRAM:  4 GiB
CA53 core 1 running.
CA53 core 2 running.
CA53 core 3 running.
All (4) cores are up.
MMC:   FSL_SDHC: 0
Loading Environment from MMC... OK
Using external clock for PCIe0
Configuring PCIe0 as RootComplex(x2)
Using external clock for PCIe1
Frequency 125Mhz configured for PCIe1
Configuring PCIe1 as SGMII(x2) [XPCS0 2.5G, XPCS1 OFF]
PCIe0: Failed to get link up
Pcie0: LINK_DBG_1: 0x00000000, LINK_DBG_2: 0x00000800 (expected 0x000000d1)
DEBUG_R0: 0x00afb000, DEBUG_R1: 0x08200000
PCI: Failed autoconfig bar 1c
```

SECURE CONNECTIONS
FOR A SMARTER WORLD