AN12938

Using RSDK1.4.0 Demo Application with RDK-S32R274 or DCC Kit

Rev. 0 — July, 2020 Application Note

by: NXP Semiconductors

Contents

1 Introduction

This application note shows how to install, build and run the Radar SDK (RSDK) on the NXP RDK-S32R274 Development Kit or DCC kit hardware which is based on the S32R274 radar microcontroller and TEF8102 integrated radar transceiver. You will be able to install S32 Design Studio and RSDK 1.4.0 on a host PC, build the RSDK demo application example project, deploy it to the RDK sensor hardware and then execute the demo software viewing the output on host PC.

2 Software requirements

Table 1. Software required

	Requirement	Version	Description	Download Link
IDE	S32 Design Studio for Power Architecture	Version: 2.1	S32 Design Studio for Power Architecture for Automotive and Ultra-Reliable MCUs.	https://www.nxp.com/design/software/development-software/s32-design-studio-ide/s32-design-studio-for-power-architecture:S32DS-PA?&tab=Design_Tools_Tab
	S32 Design Studio for Power Architecture v2.1 Update 8	Update 8	RSDK 1.4.0 RTM for S32R274 and S32R372.	https://www.nxp.com/design/software/development-software/s32-design-studio-ide/s32-design-studio-for-power-architecture:S32DS-PA?&tab=Design_Tools_Tab
Server	OpenTFTPServer	Version:1.64	MultiThreaded TFTP Server Open Source Freeware .	https://sourceforge.net/projects/tftp-server/
	Alternative: UART_server	N/A	UART server for RSDK.	In rsdk package path: rsdk\Tools\UART_server
Debug	PEmicro debugger	Multilink/ Multilink FX	Advanced High Performance Debug Probe .	http://www.pemicro.com/products/product_viewDetails.cfm?product_id=15320143
	Alternative: Lauterbach Trace32	Support for Power Architecture	Lauterbach Power Debug.	See Lauterbach website

Table continues on the next page...

Table 1. Software required (continued)

	Requirement	Version	Description	Download Link
Analysis	Matlab (2015b-2017b)	Version: 2017b	MATLAB is a commercial mathematics software produced by MathWorks.	https://www.mathworks.com/
	Model-Based Design Toolbox	Version:1.3.0	The SPT Toolbox for MATLAB.	NXP software account or contact FAE
Tool	7-Zip	Latest	To unzip/untar compressed folders.	https://www.7-zip.org/

2.1 S32 Design studio download

Download the S32DS IDE and upgrade package from:

S32 Design Studio for Power Architecture®v2.1 - Windows/Linux

S32 Design Studio for Power Architecture v2.1 Update 8

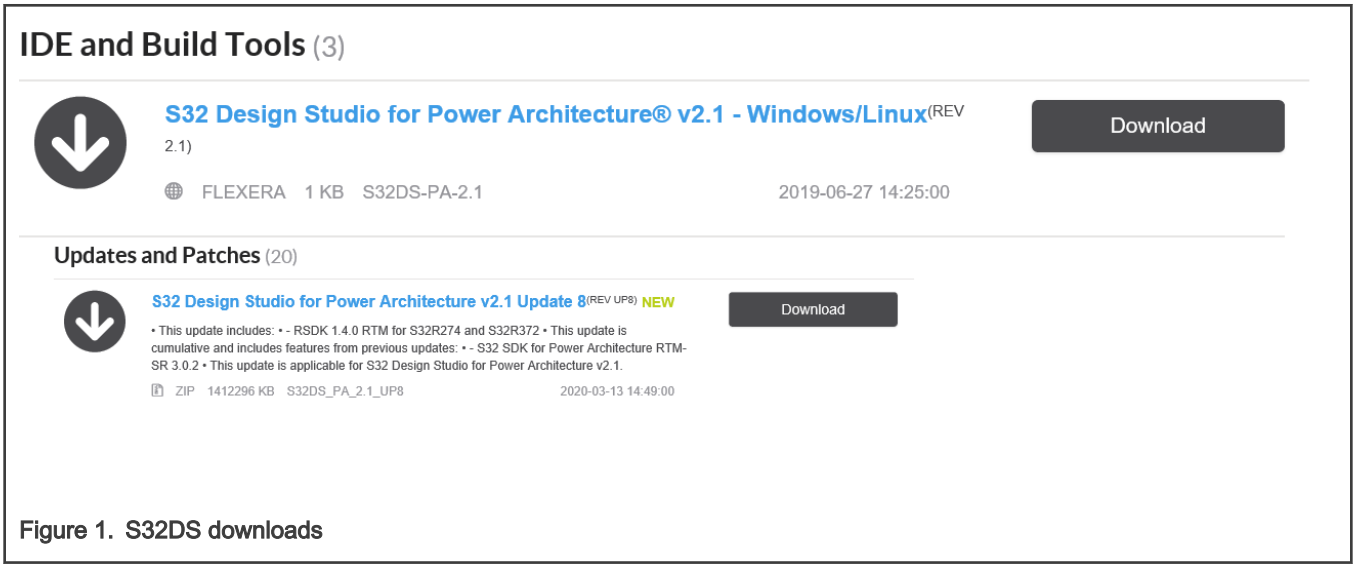


Figure 1. S32DS downloads

2.2 S32 Design Studio Installation

S32 Design Studio (S32DS) for Power Architecture V2.1 is installed by running the downloaded installer executable:

Figure 2. S32DS Installation

Update 8 for S32DS is installed using the following method:

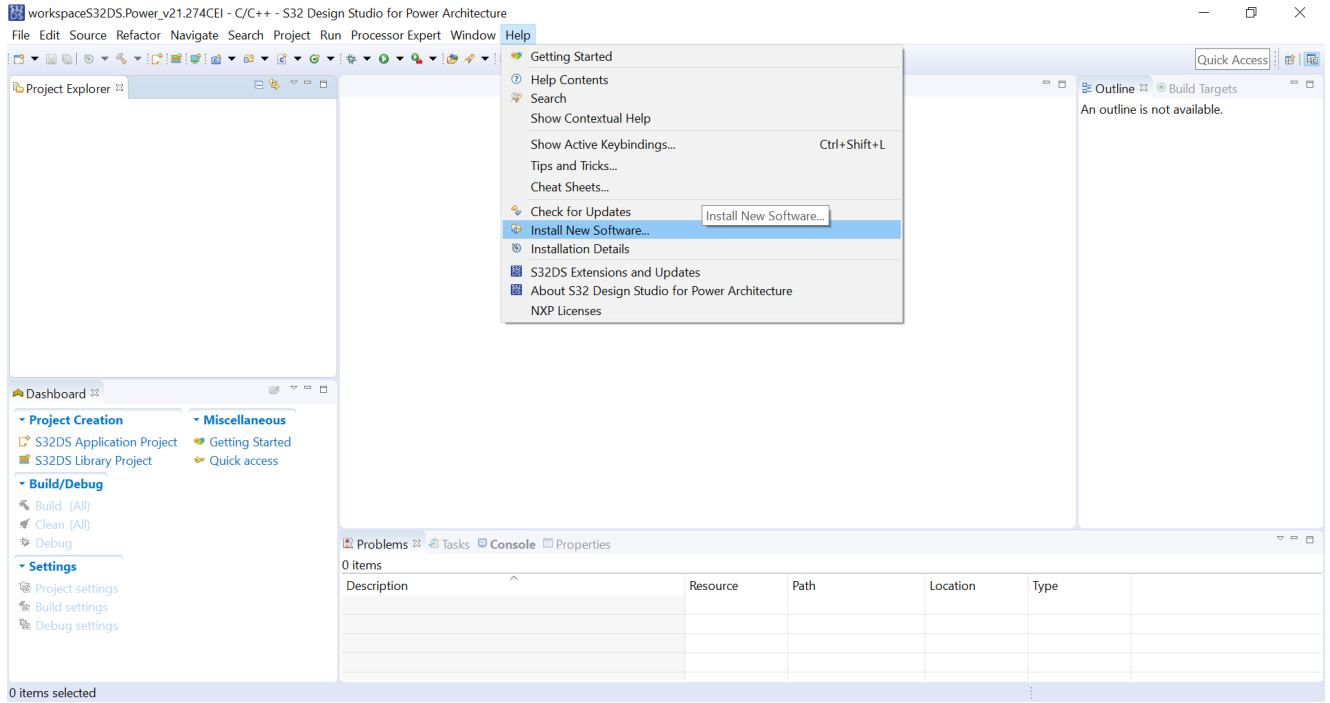


Figure 3. Install new software

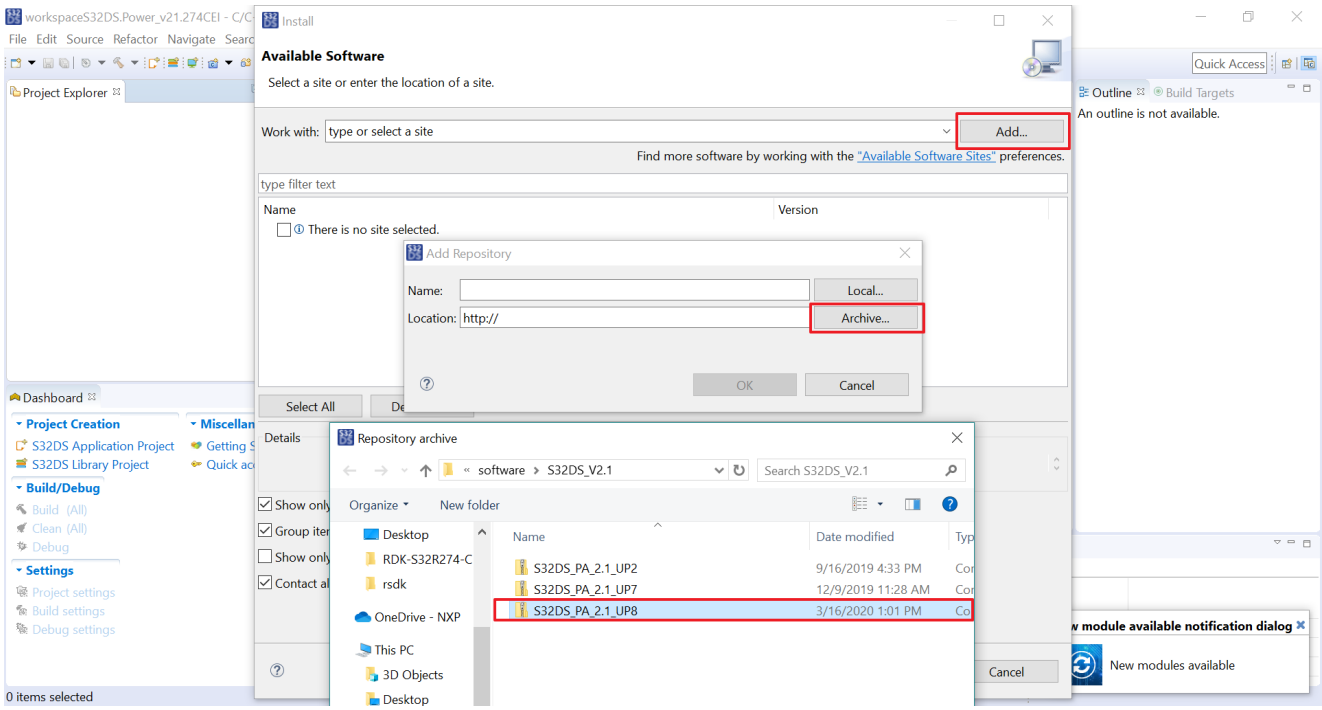


Figure 4. Select the installation package

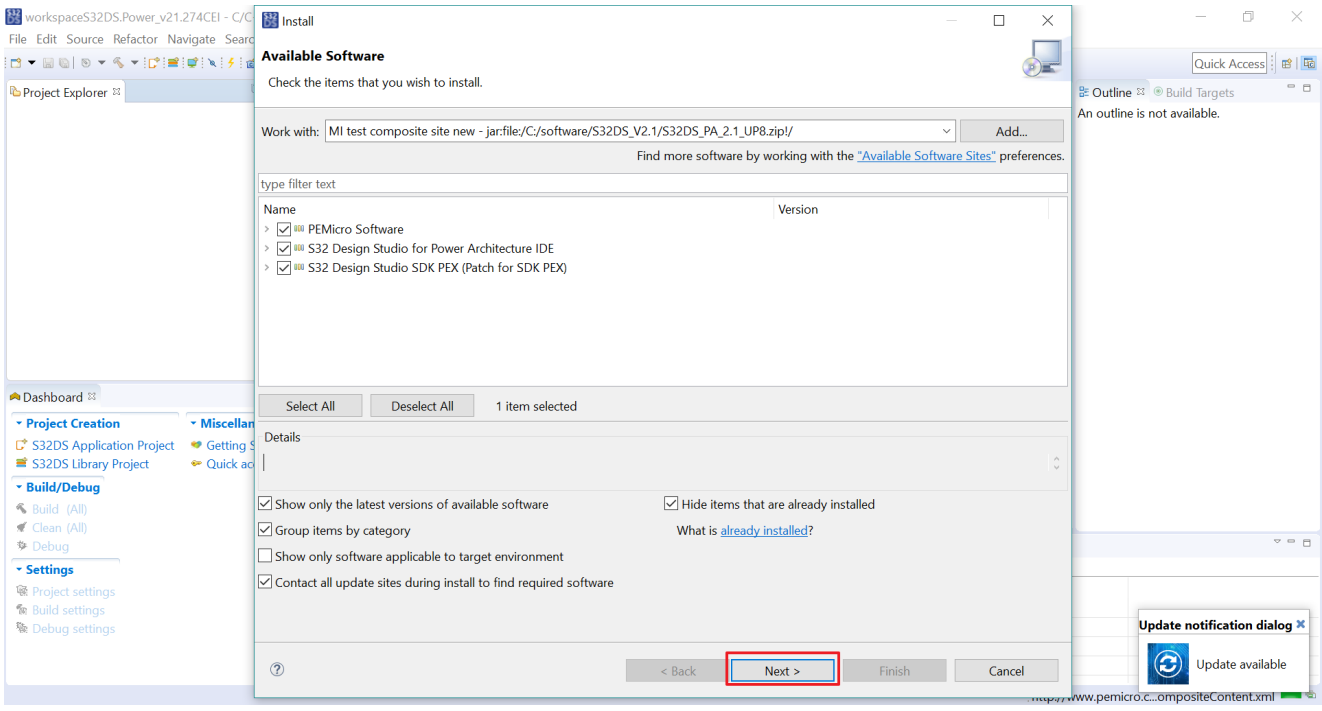


Figure 5. Select installation options

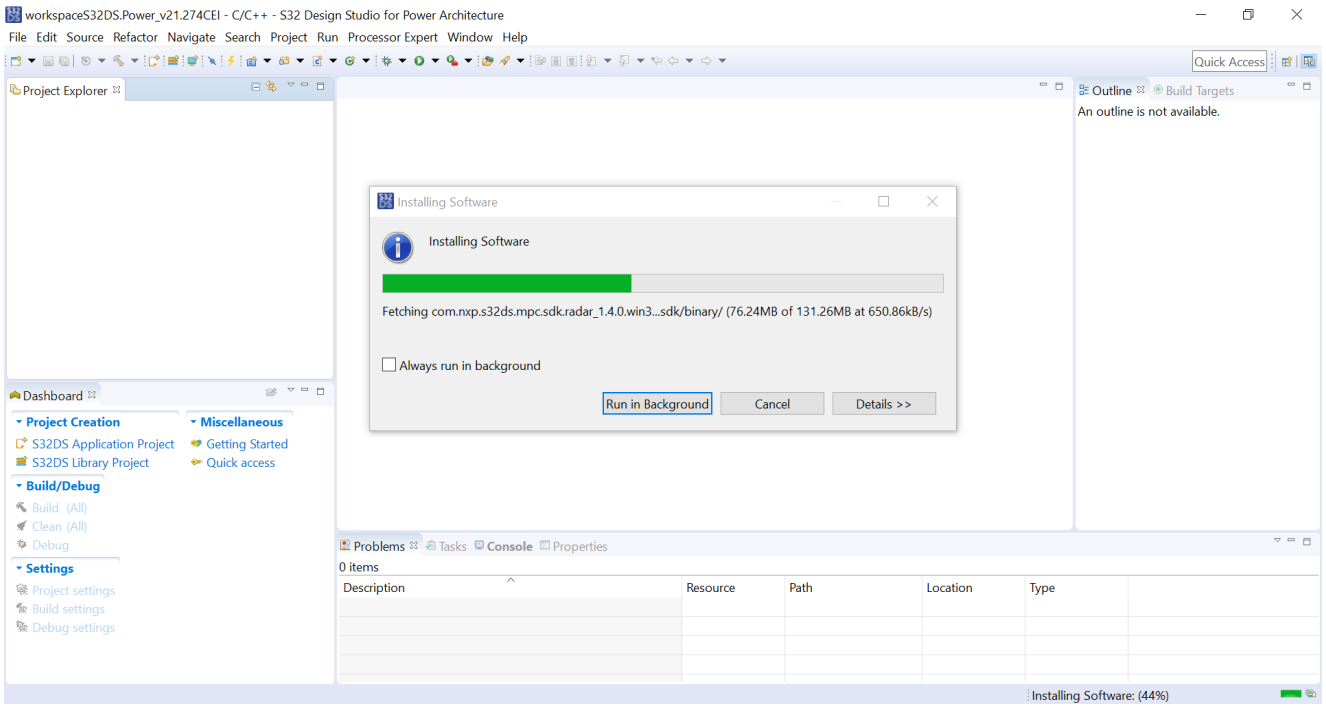


Figure 6. Complete installation

NOTE

Refer to Table 1 for other software installation.

2.3 Radar SDK installation

Radar SDK (RSDK) 1.4.0 is included in S32DS update 8 package. After successful installation of the update, the embedded RSDK files can be found in the S32DS installation directory as shown in the following figure.

Open the RSDK_Release_Notes PDF to find a link to the RSDK User Manual presented in HTML.

The layout of the RSDK folder structure is shown in the following figure, you can can use this to locate the pre-built libraries (.a files) and library API headers.

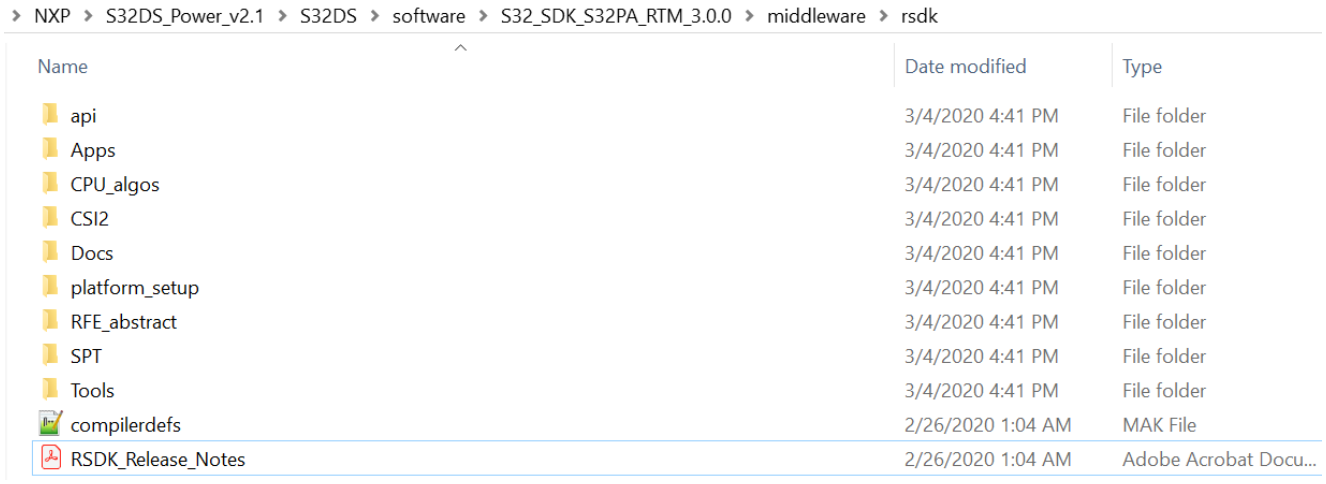


Figure 7. RSDK components

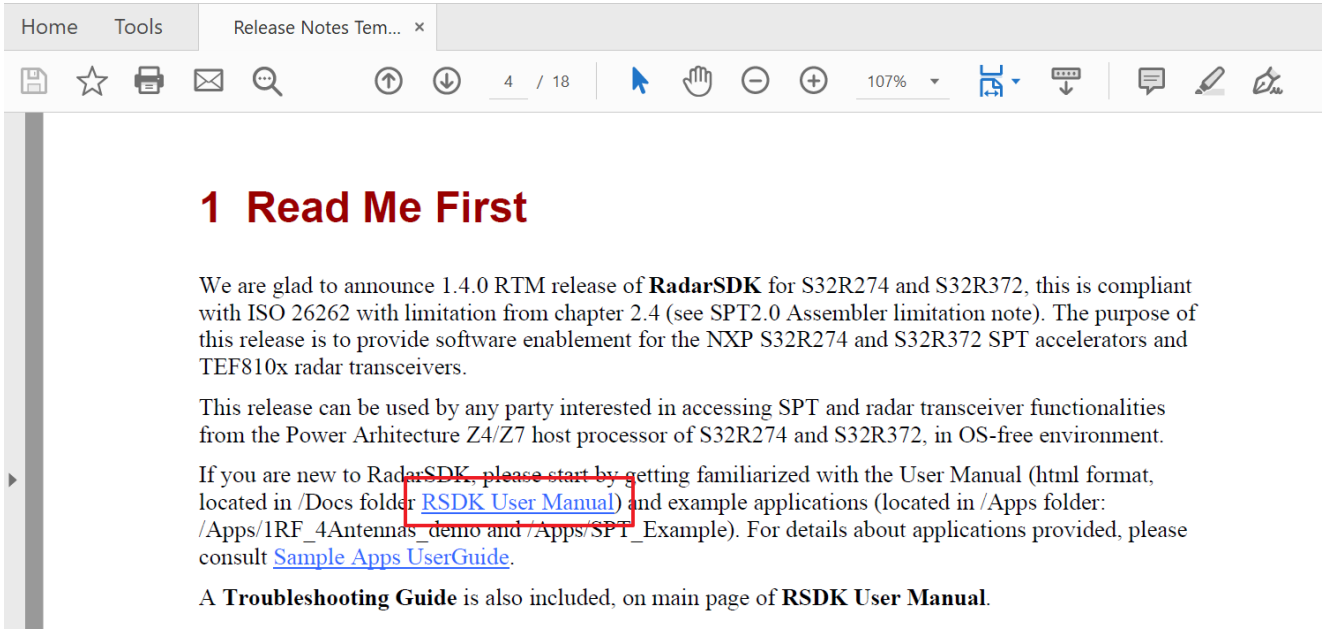


Figure 8. RSDK release note

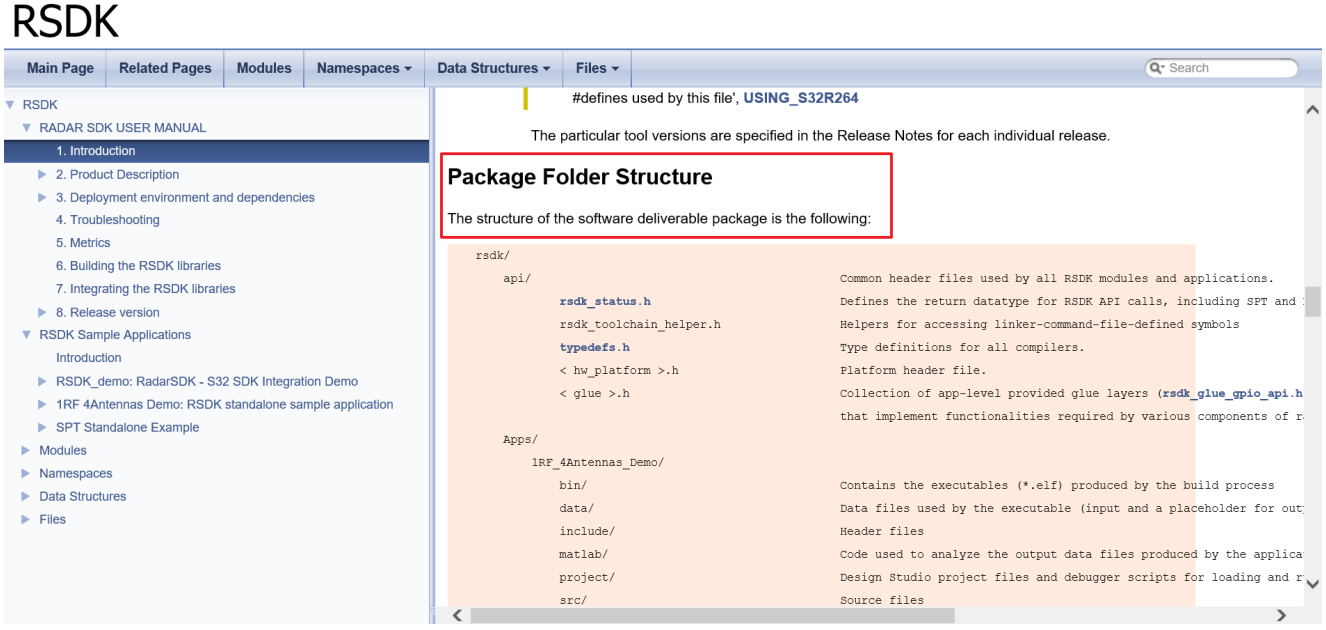


Figure 9. RSDK user manual

2.4 RSDK standalone install (optional)

It is possible to install the RSDK independently of S32DS in case the reader wants to use it with other IDEs/toolchains. This method results in an RSDK installation that can also be used with S32DS however it does not enable RSDK 1.4.0 as an S32DS 'integrated SDK', see Differences in RSDK standalone vs S32DS integrated install.

NOTE

If you follow the steps properly menioned in S32 Design studio download and S32 Design Studio Installation, you do not need to install the RSDK package again.

The latest version of the RSDK (1.4.0 at the time of writing) can be downloaded using your NXP software account as shown in the figure below. If you cannot get it, please contact your local NXP FAE or sales representative.

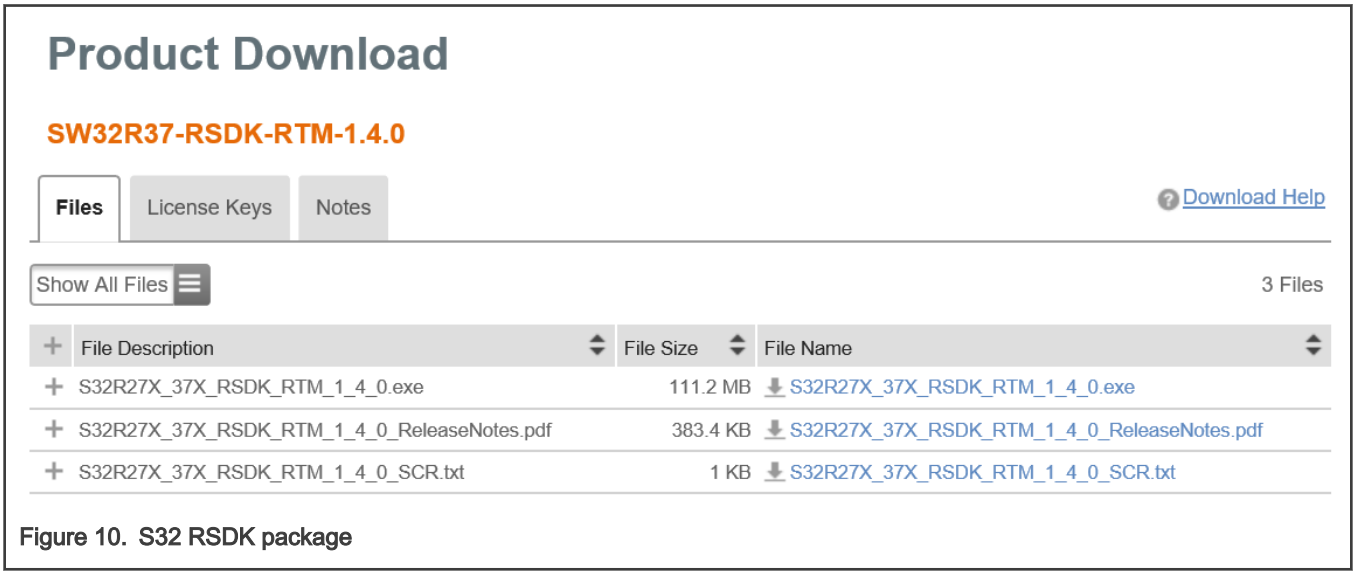


Figure 10. S32 RSDK package

The RSDK is installed by running the installer executable and following the steps:

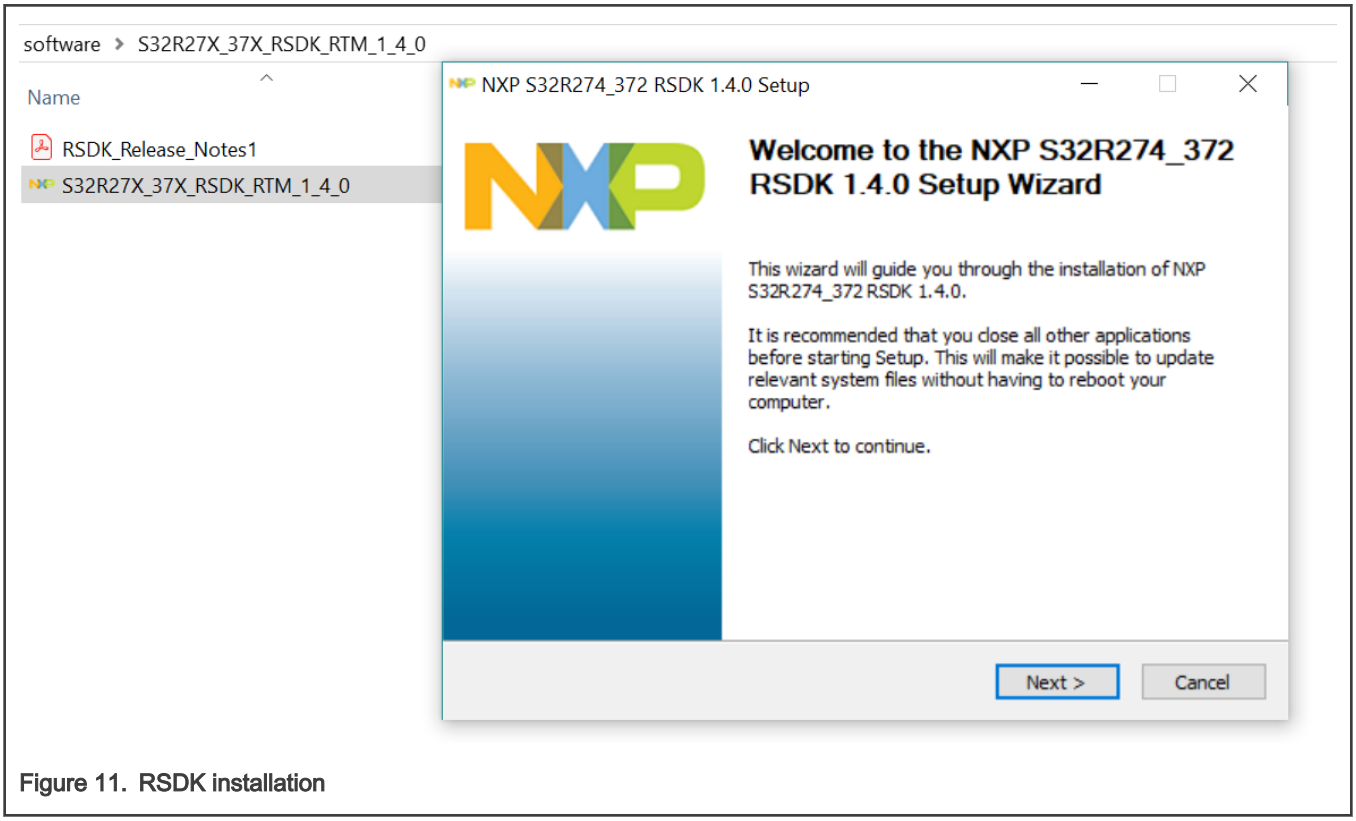


Figure 11. RSDK installation

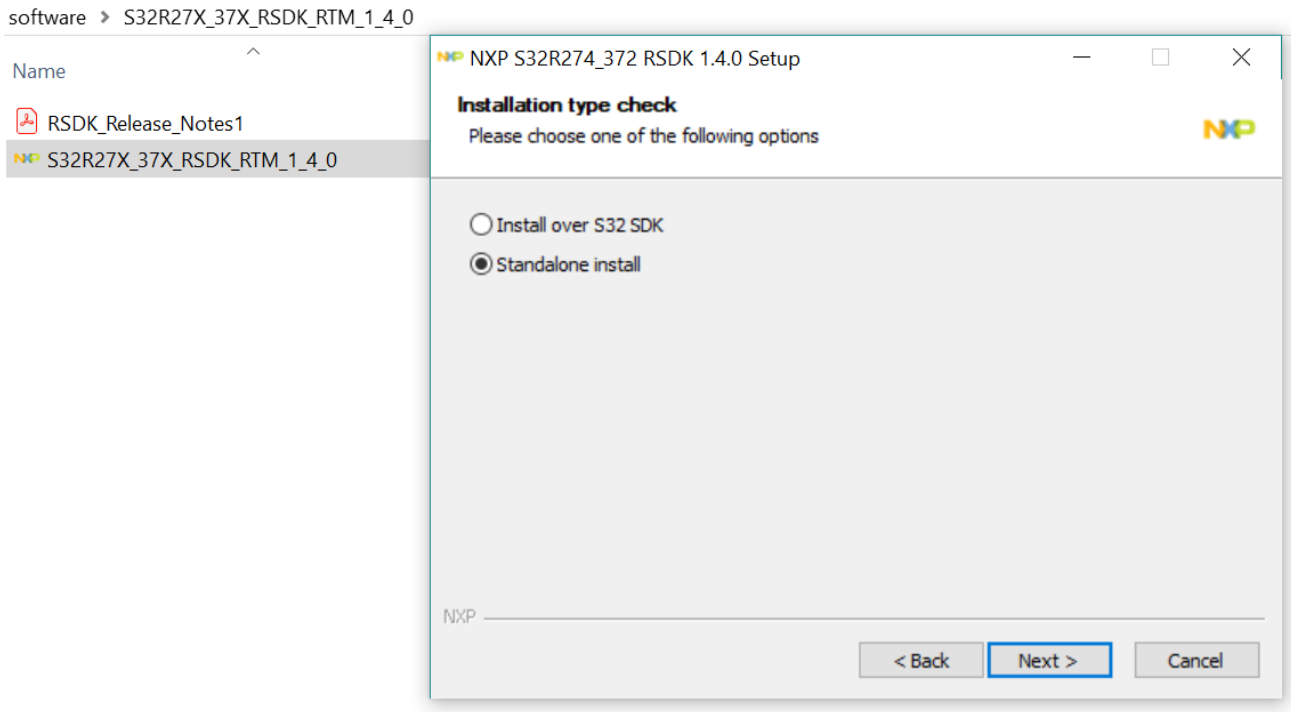


Figure 12. Installation mode selection

NOTE

Installation package can be installed under the Platform SDK module of S32DS or independently in the specified path.

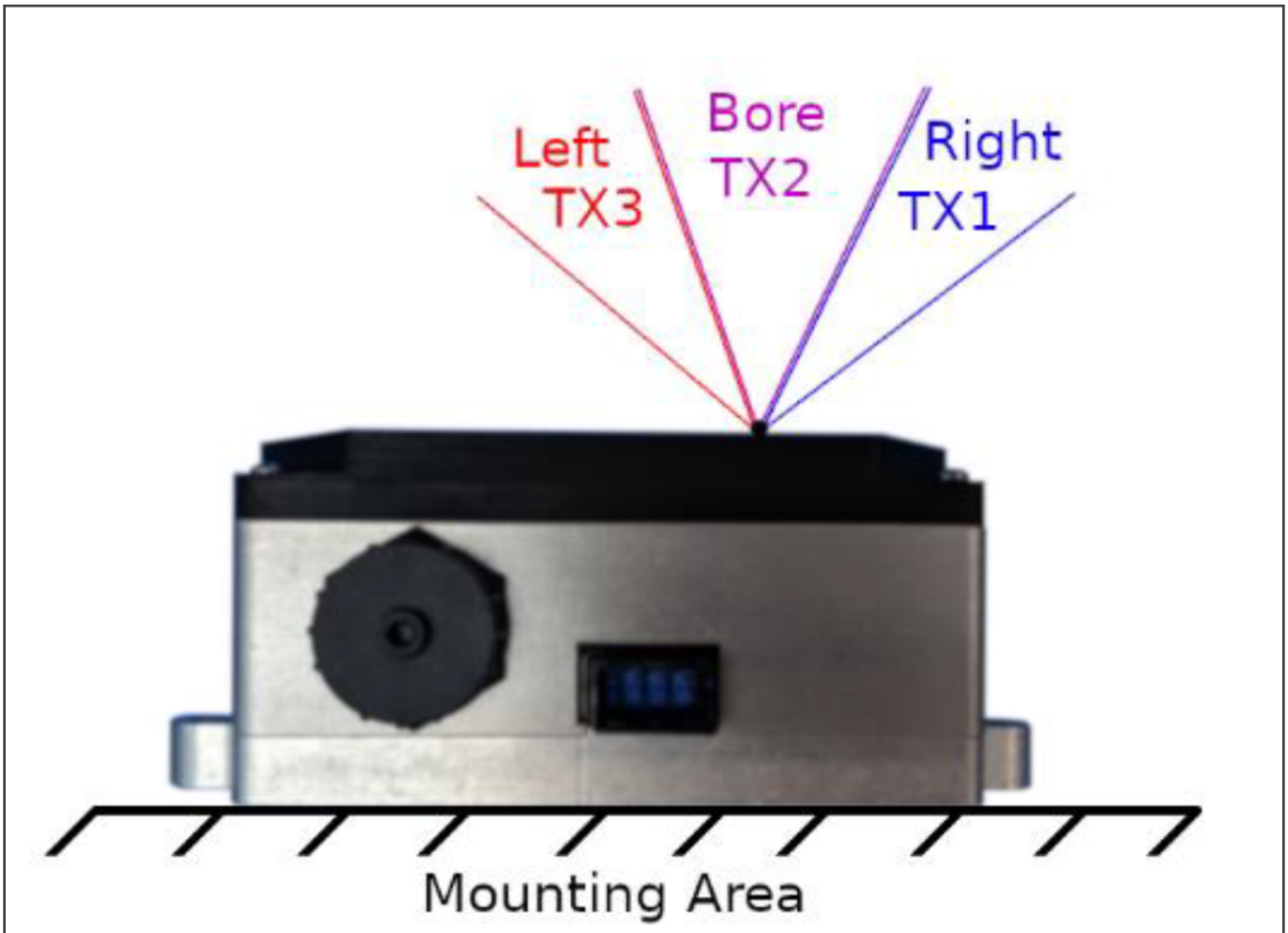
2.5 Differences in RSDK standalone vs S32DS integrated install

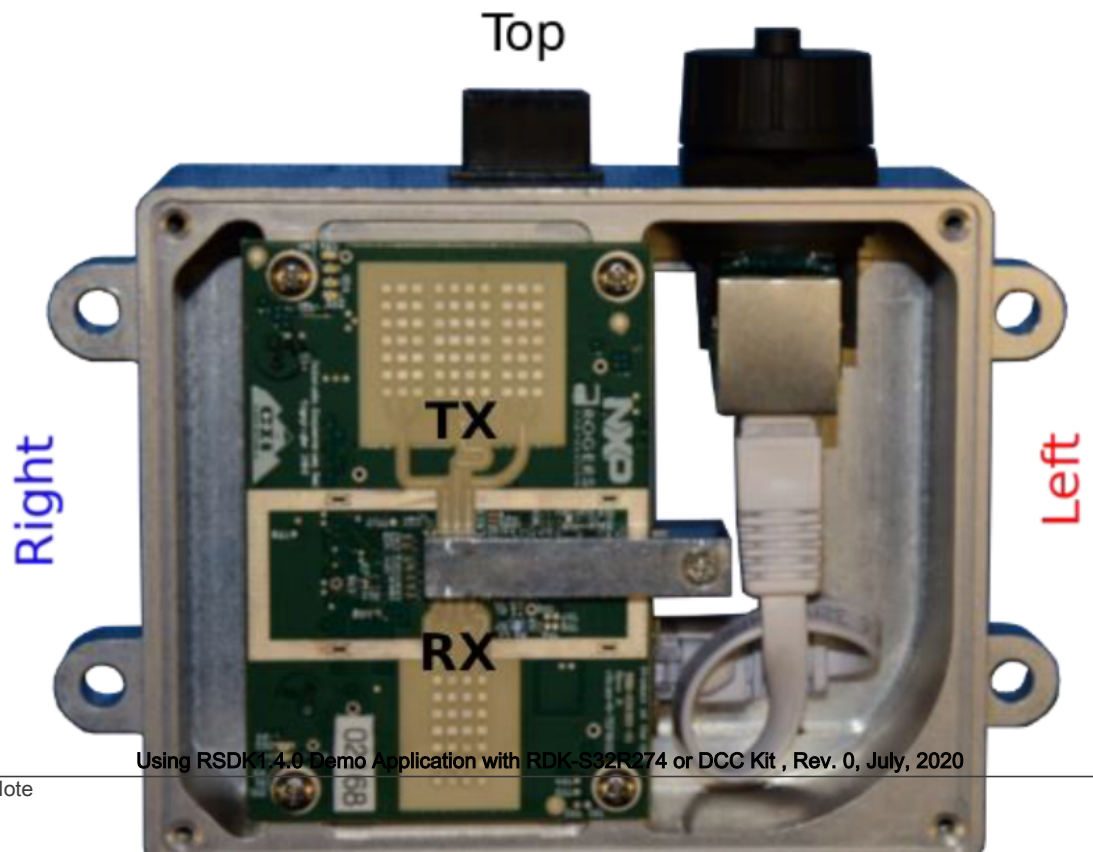
The contents in RSDK 1.4.0 as part of S32 Design Studio for power Architecture V2.1 update 8 and S32R274_372 RSDK 1.4.0 release package are mostly the same. The difference is that the RSDK 1.4.0 included in the former is installed inside the S32DS installation as an integrated SDK, allowing RSDK to be used as a module of the platform SDK making use of the peripheral drivers such as SPI, interrupt controller etc. The latter can be installed and used completely independent of S32DS and the platform SDK.

3 Hardware requirements

The RDK-S32R274 platform is a Radar Electronic Control Unit (ECU) comprising of the S32R274 radar processing microcontroller, TEF8102 radar chip and 77 GHz antenna.

OThe following figure outlines the labels for the directions of the three steered antennas - left, bore, and right.

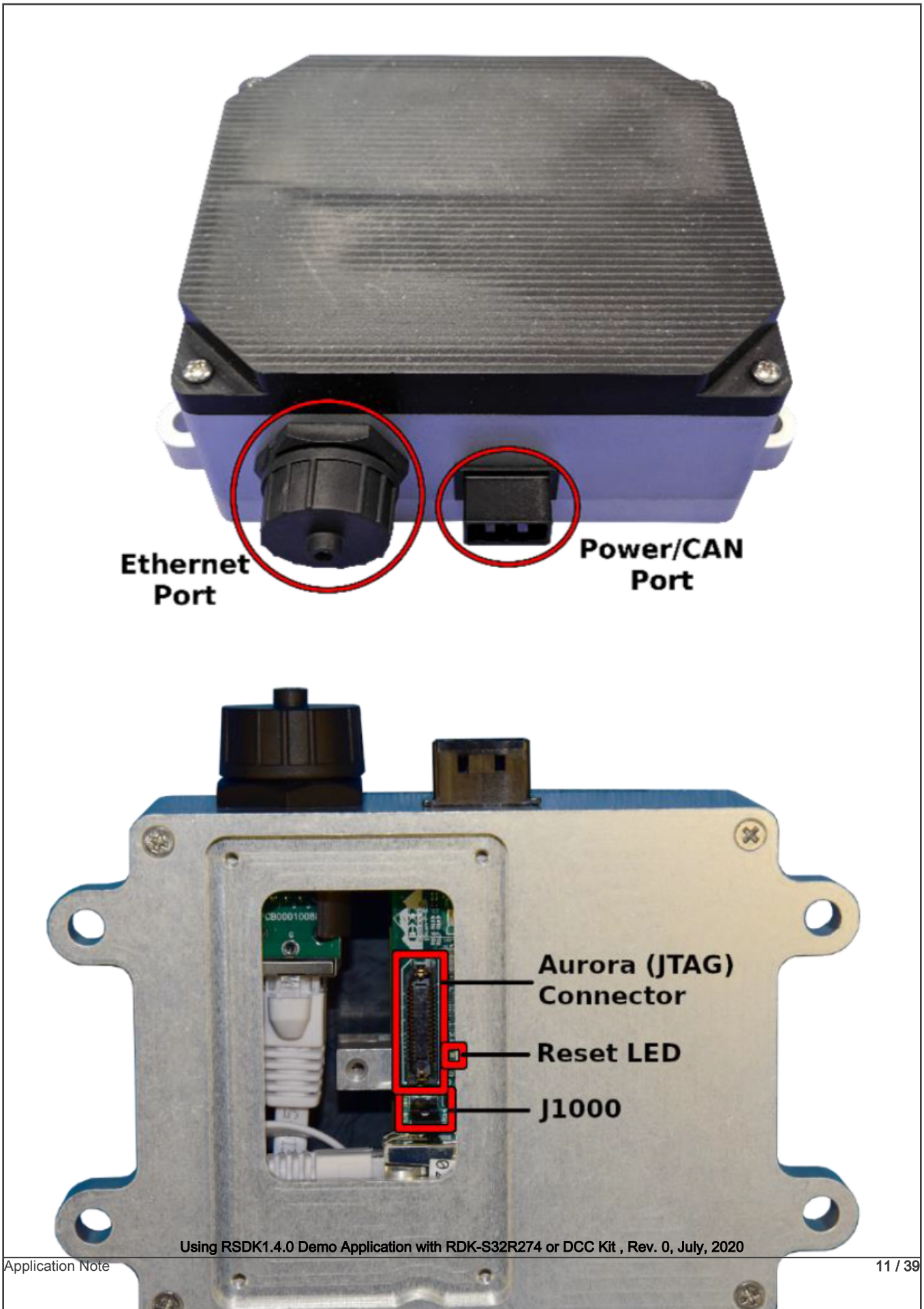

Left
TX3
Bore
TX2
Right
TX1
Mounting Area


Top
Right
Left
TX
RX

3.1 HW accessories and connectors

The following hardware connectors and accessories are required:

1. RDK-S32R274
2. Power Adapter Cable
3. AC Adapter
4. Wall outlet supplying 100-240 V at 50-60 Hz
5. PEmicro: Multilink / Multilink FX


Ethernet Port
Power/CAN Port
Aurora (JTAG) Connector
Reset LED
J1000

3.2 Power dteps

The below steps needs to be followed to supply power to the RDK in a lab setting.

1. Plug in the adapter cable into the RDK. There should be an audible “click” when it is properly plugged in.
2. Plug the provided AC adapter into the wall outlet with the specifications.
3. Plug the DC barrel of the AC adapter to the DC jack on the adapter cable on the RDK.
4. The system should power up in less than 5 seconds.

Since there is no user feedback from outside the enclosure, the best way to determine if the system is on and operating is by checking the lights on Ethernet port of the computer the RDK is plugged into. If they are illuminated, then the system is powered up and booted.

3.3 Debug mode

The RDK system utilizes the NXP FS84 series safety System Basis Chip (SBC) to provide the required voltage regulation and safety monitoring for both the S32R274 and the TEF8102 devices. The FS84 has a suite of functional safety features, including a watchdog which can be used as part of the safety case to ensure that the MCU and the application is still active. This watchdog requires routine servicing via the SPI interface from the MCU (and therefore must be part of the software running on the MCU), which if not provided will cause the SBC to provide an external reset to the MCU in an attempt to recover the application. If there is still no response from the MCU after multiple resets then the SBC will cut power to the MCU as a safety feature.

For debug purposes this watchdog functionality can be bypassed by putting the SBC into debug mode, which allows constant power to the MCU without servicing via SPI required. This is the default configuration for the RDK board, debug mode is enabled by a tabbed jumper J1000 on the board. If during development the normal watchdog enabled mode of the SBC is required then J1000 can be removed.

NOTE

Only install/remove J1000’s jumper when the RDK is powered off.

For more information concerning the FS84, documentation can be found on NXP’s website: NXP FS8400 Website

NOTE

Before the programing, connect the PEmicro debuger and ethernet port to the hardware platform and computer respectively, then power up hardware platform.

4 OneRF_4Antennas_demo

OneRF_4Antennas_demo is an bare-metal code project, which can be used independently of S32DS.

4.1 Creating a project

Create a new project from the example path. The steps are shown in the following figures.

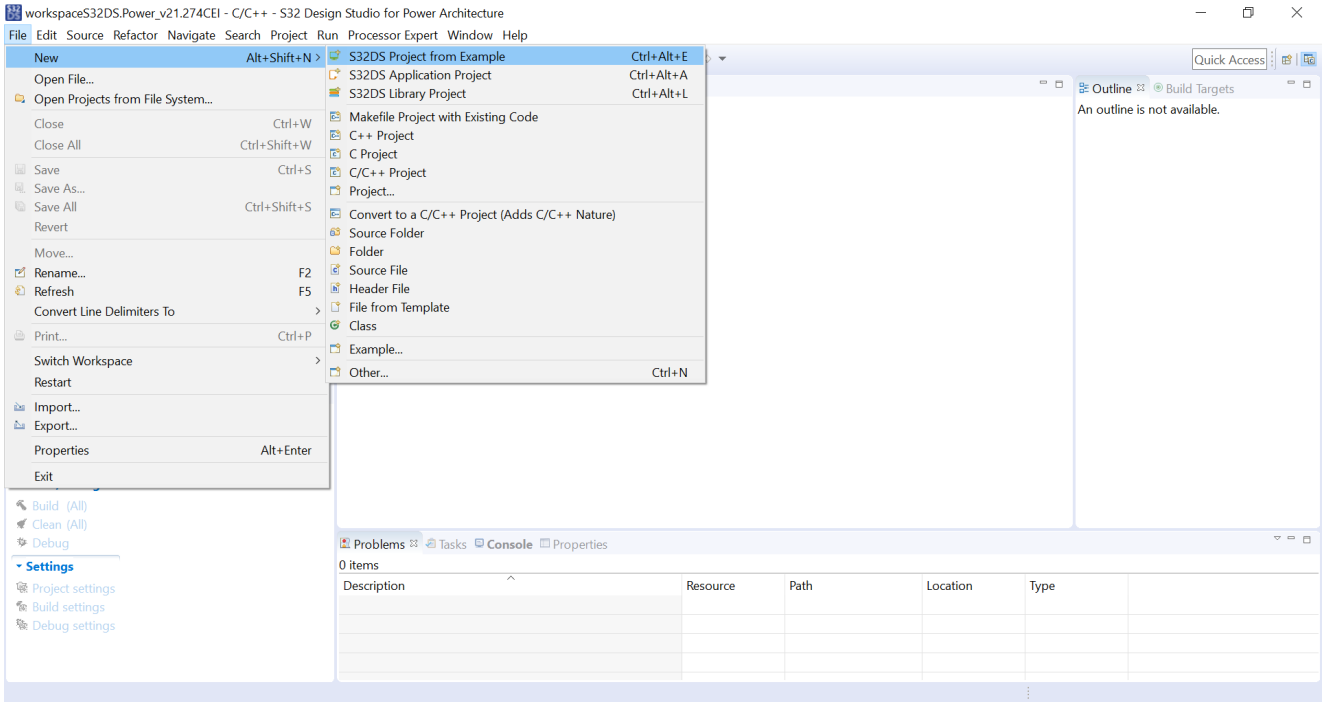


Figure 15. S32DS project from example

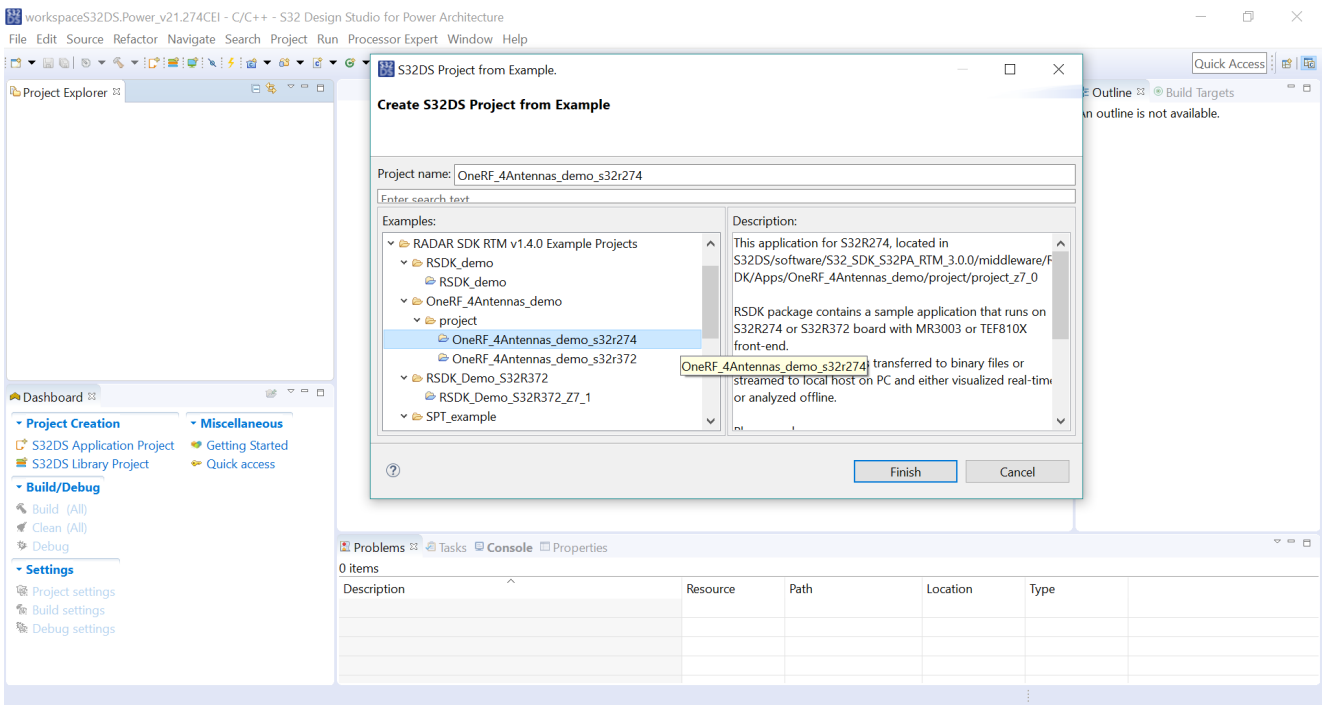


Figure 16. Select OneRF_4Antennas_demo

NOTE

- OneRF_4Antennas_demo is based on bare-metal code as shown in figure 4-3. The project is created as a virtual project by creating S32DS project from example. There is no entities file under the workspaceS32DS.Power path. The entities file is located in rsdk \ Apps \ OneRF_4Antennas_demo.
-

You can use the method of import projects to create this projects as shown in the following figure, which is convenient for creating this projects in S32DS or other IDEs.

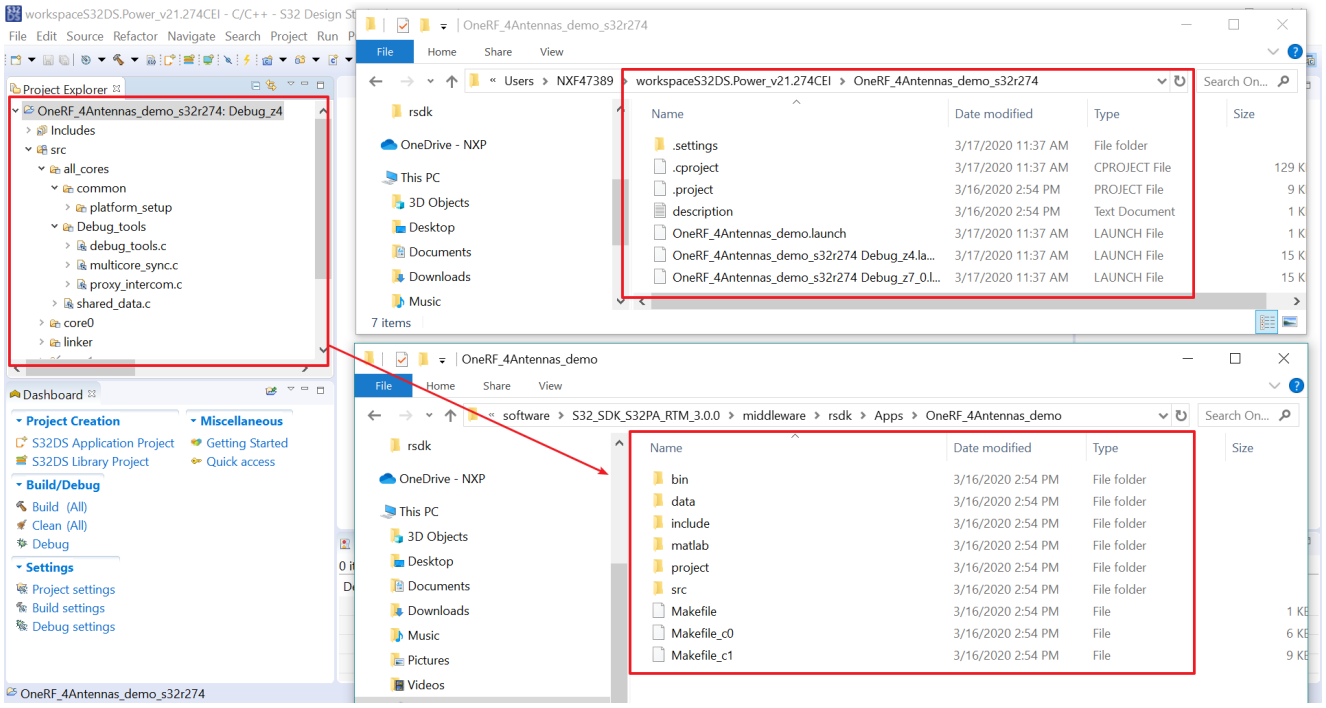


Figure 17. OneRF_4Antennas_demo entities file

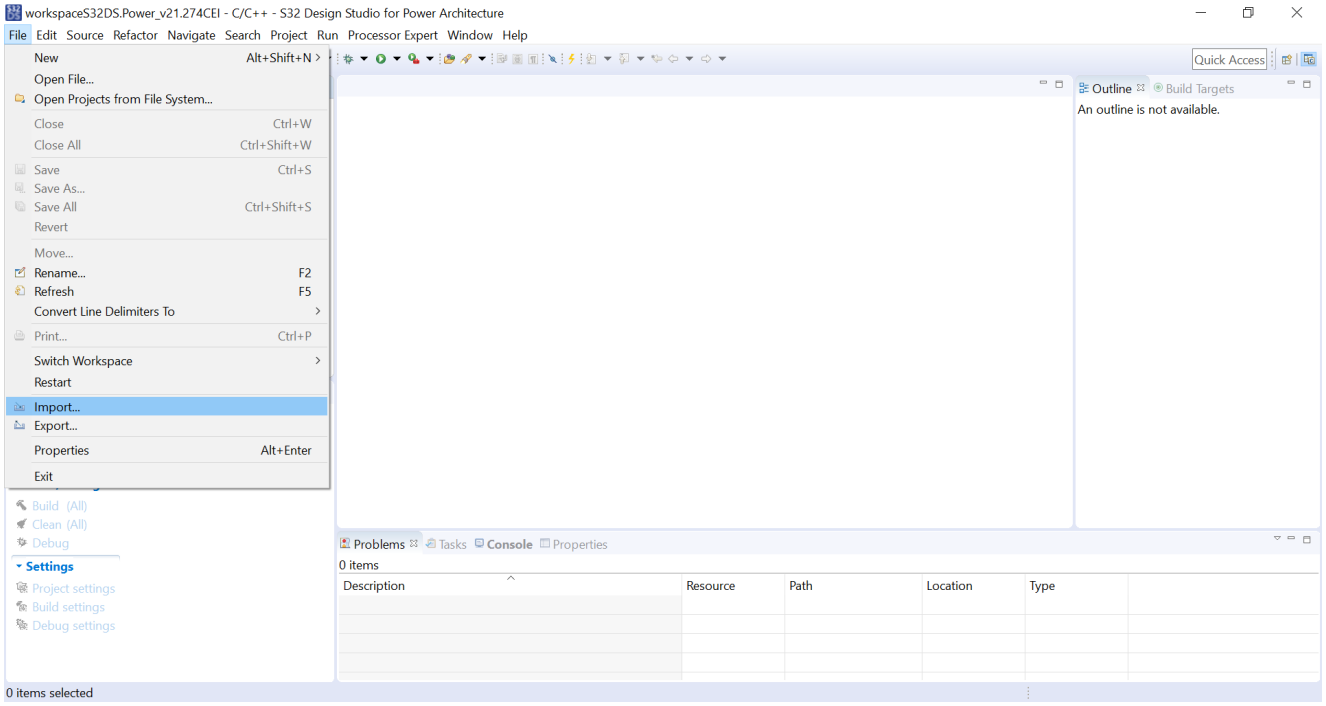


Figure 18. Import demo project

The project import steps are shown in the following figures. The key point is to find the file path where rsdk is located.

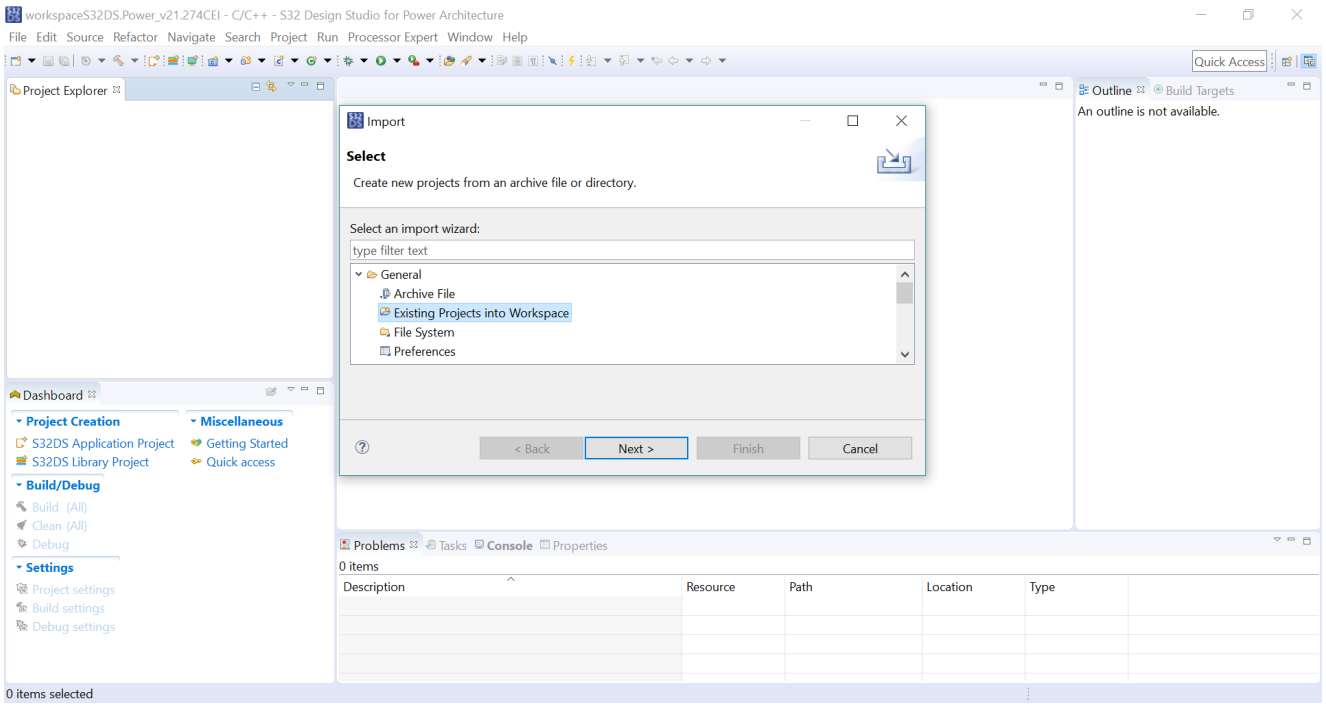


Figure 19. Select demo file path

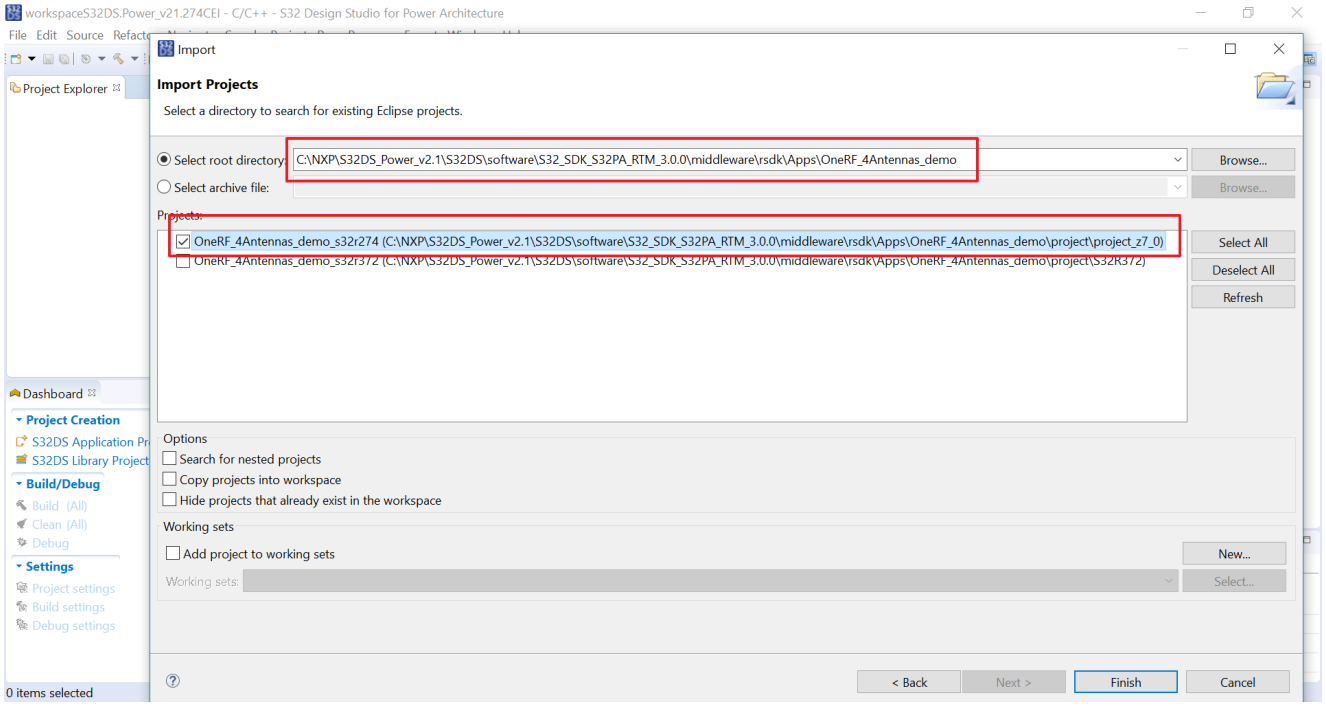


Figure 20. Import OneRF_4Antennas_demo

NOTE

Import the demo project of S32R274 (project_z7_0).

4.2 Building a project

Build the Z4 and Z7 core projects as shown in the following figures.

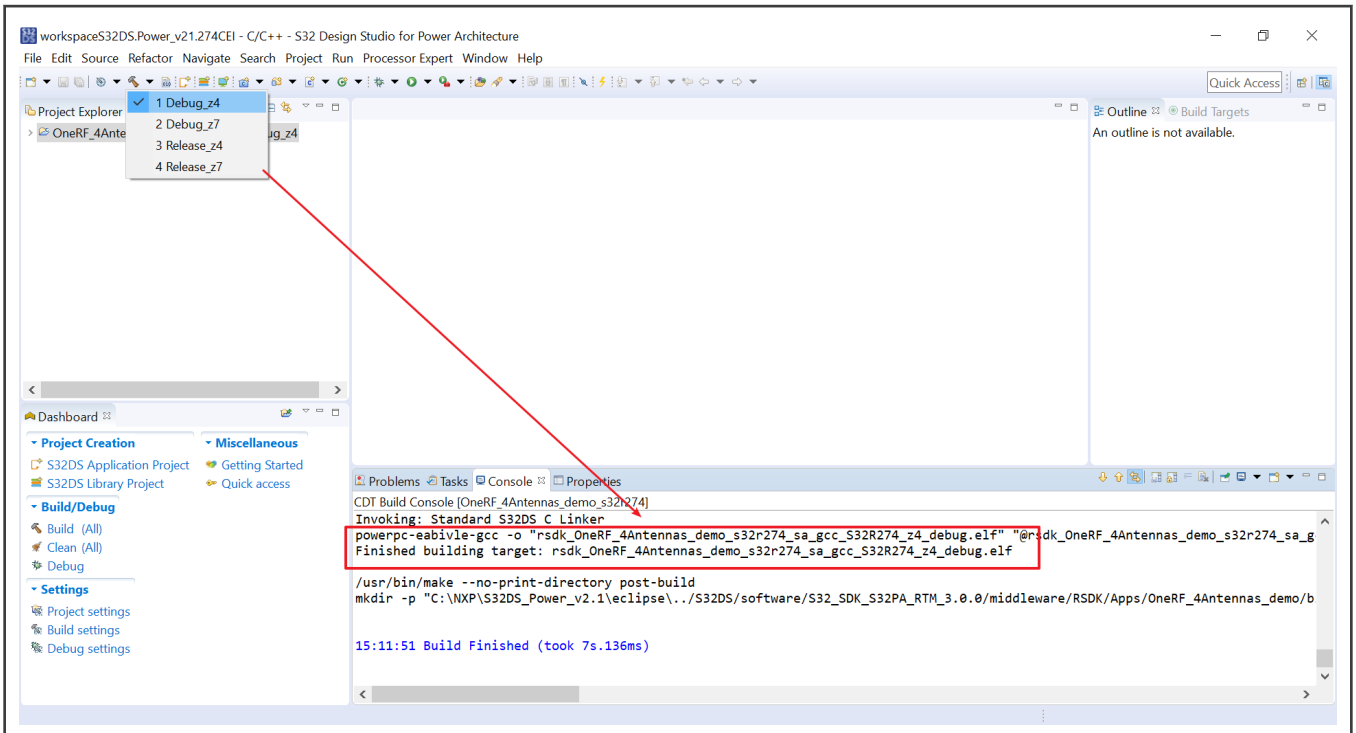


Figure 21. Build z4 core project

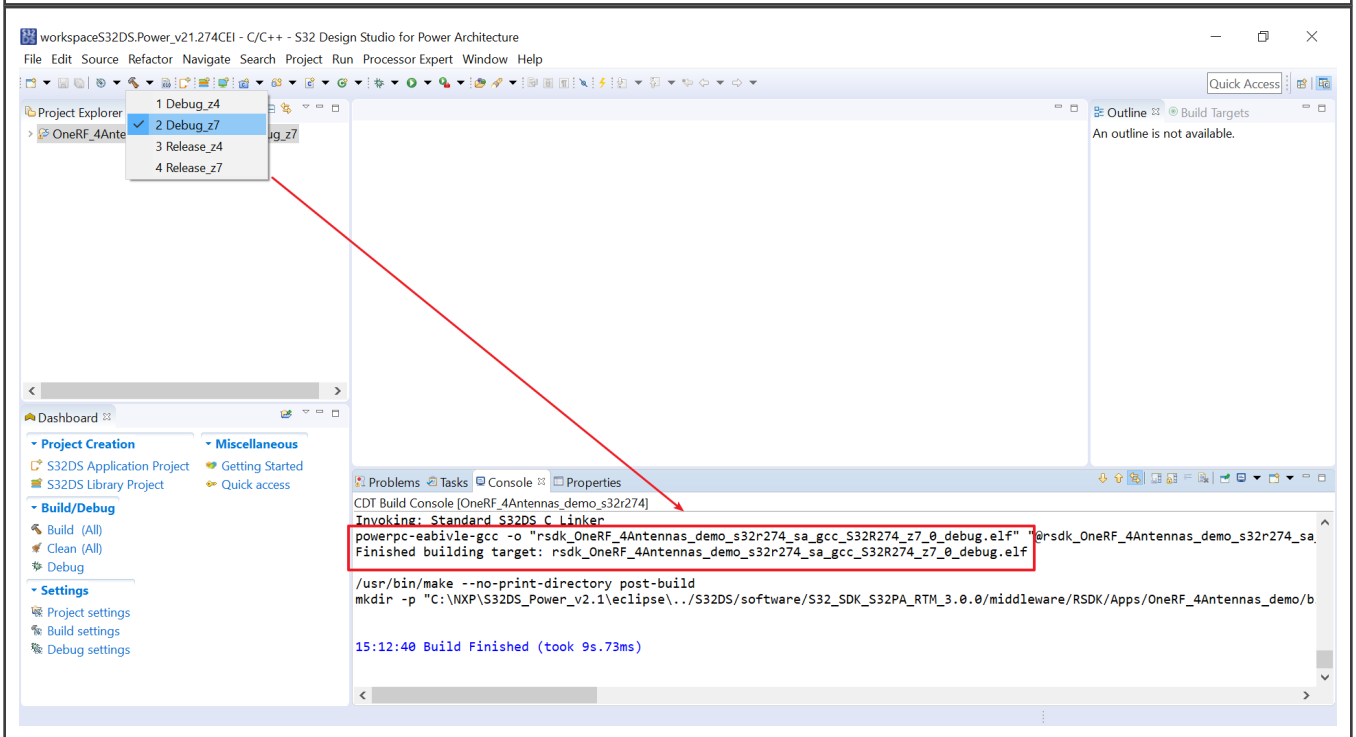


Figure 22. Build z7 core project

The corresponding debug.elf file will be generated after the project is compiled.

4.3 Data output configuration

This demo reads the local configuration file through OpenTFTPServer to achieve mode, chirp and output data configuration.

For the specific means of the parameters in the s32r274_tef810x_config_0 file, please see the rsdk/Docs/RSDK_User_Manual/rsdk_sampleapps.html#rsdk_sa_1rf4ant : Application config file

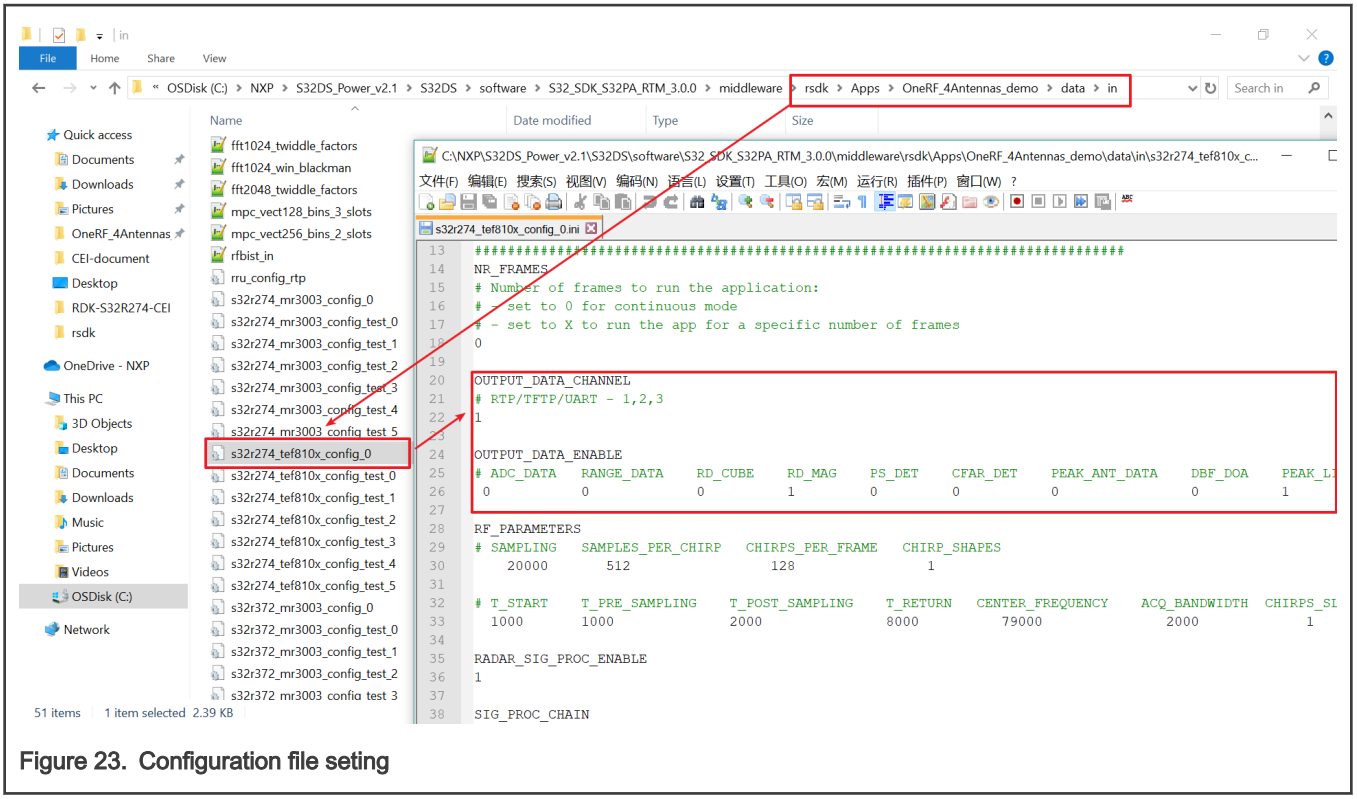


Figure 23. Configuration file seting

OpenTFTPServer is a server for RDK hardware platform and computer data transmission. You need to configure IP address, rsdk file path and read write permissions, as shown in the following figures.

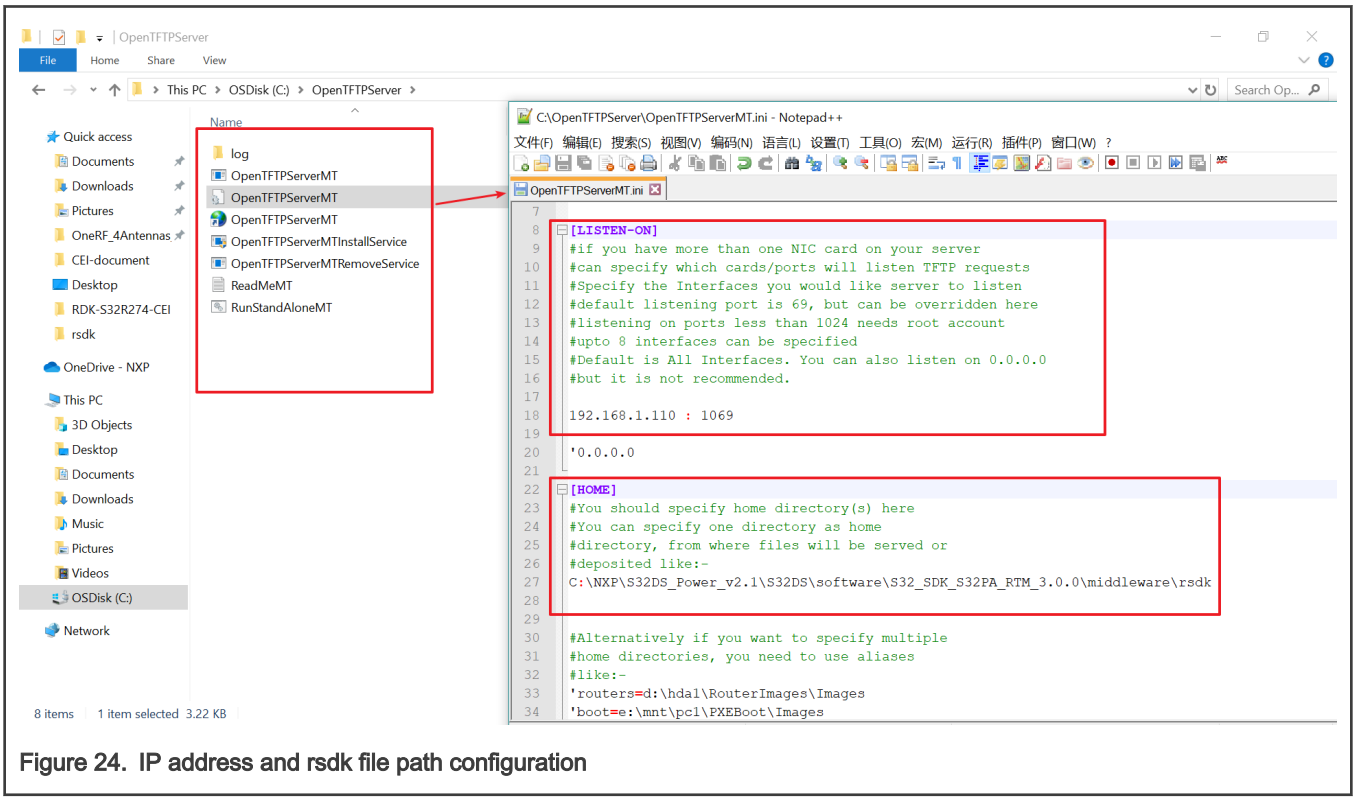


Figure 24. IP address and rsdk file path configuration

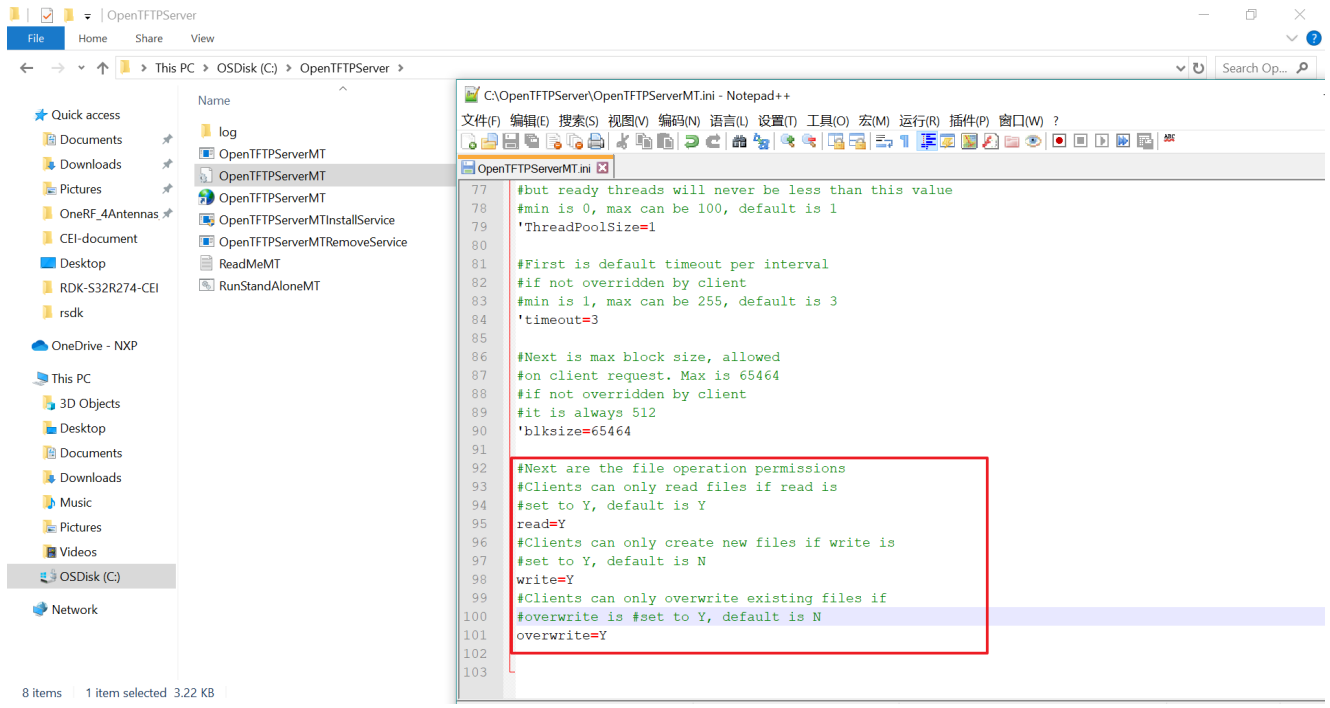


Figure 25. File read and write configuration

After the RDK platform is powered on and connected to the computer, follow the instructions shown in the below figure to configure the host computer to communicate with the HW.

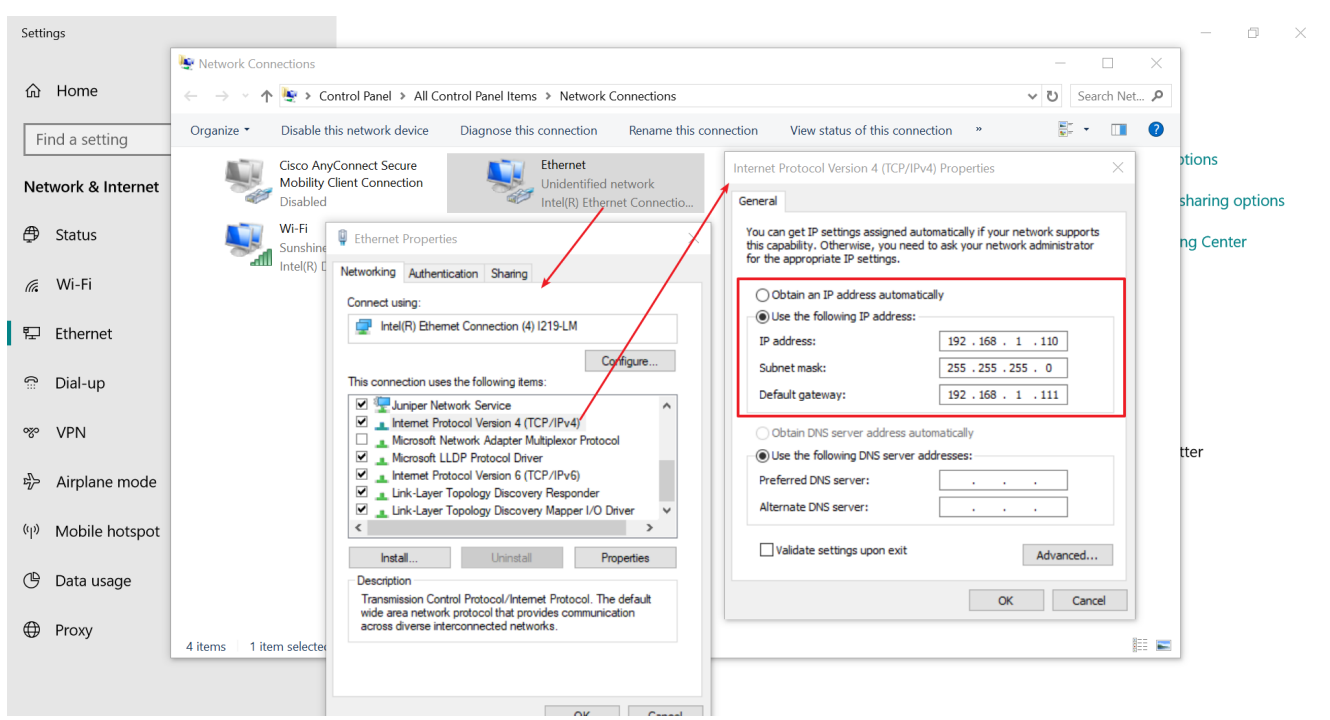


Figure 26. Host computer ethernet IP configuration

4.4 Debugging and data visualization

Before debugging, please open CMD console and cd to server app folder (default C: / OpenTFTPServer /).

Start running server (OpenTFTPServerMT.exe -v).

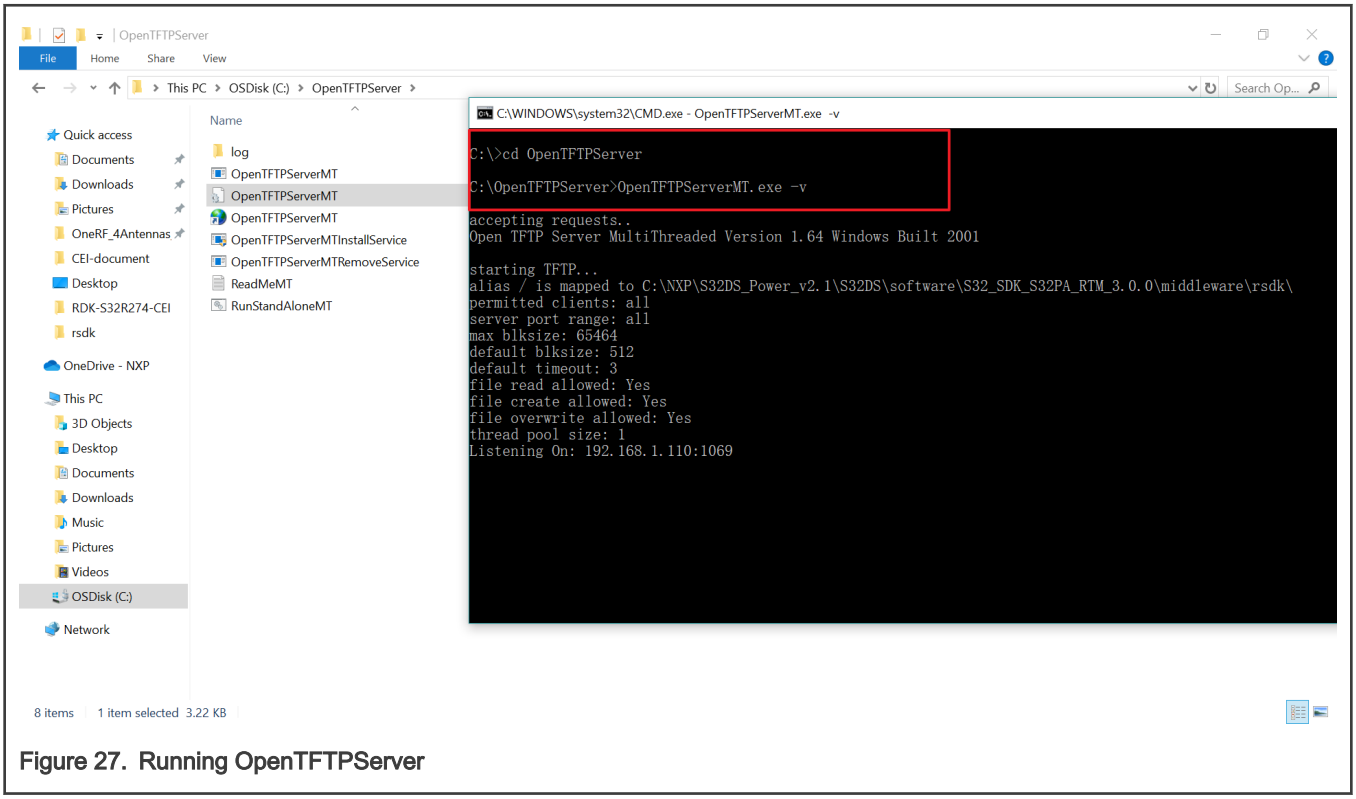


Figure 27. Running OpenTFTPServer

Download the debug.elf file through S32DS + PEmicro, as shown in the following figures.

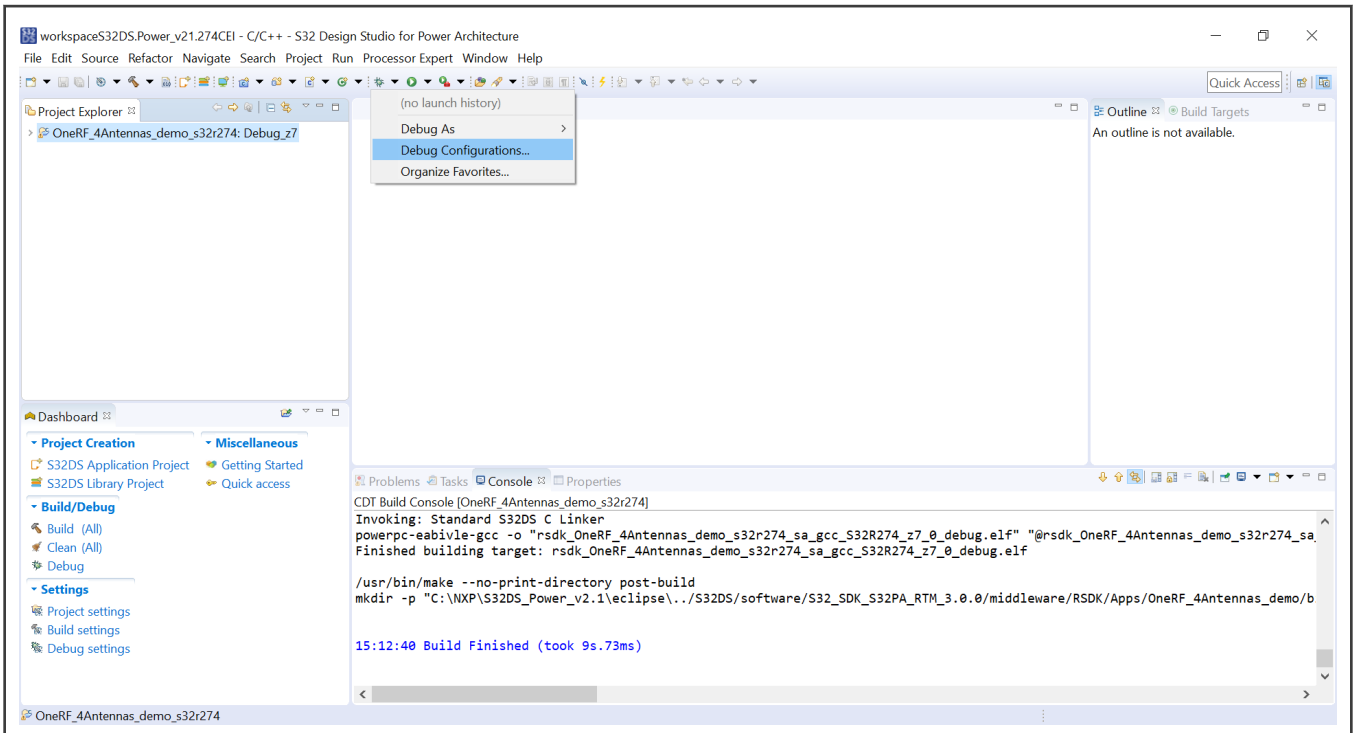
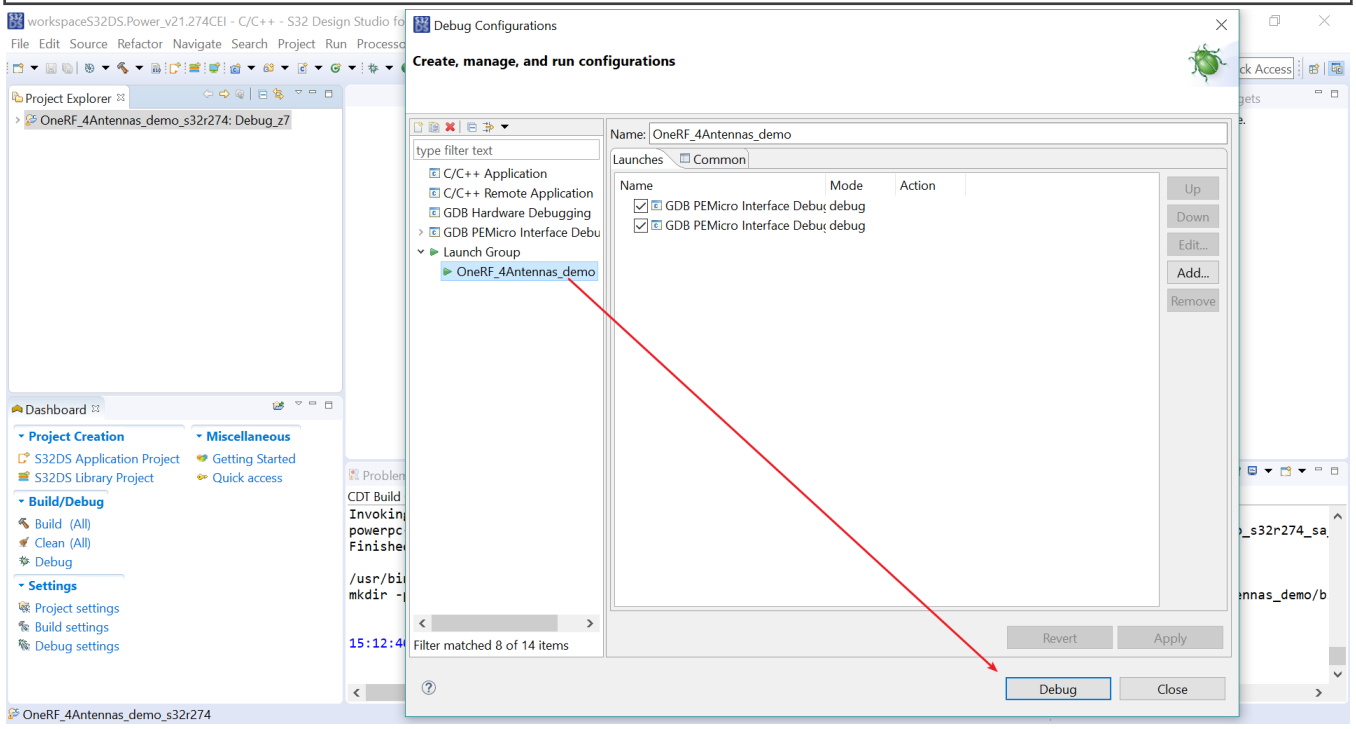


Figure 28. Select debug configuration

Figure 29. Download debug.elf file



Keep OpenTFTPServer running, and run the demo at full speed, as shown in the following figures.

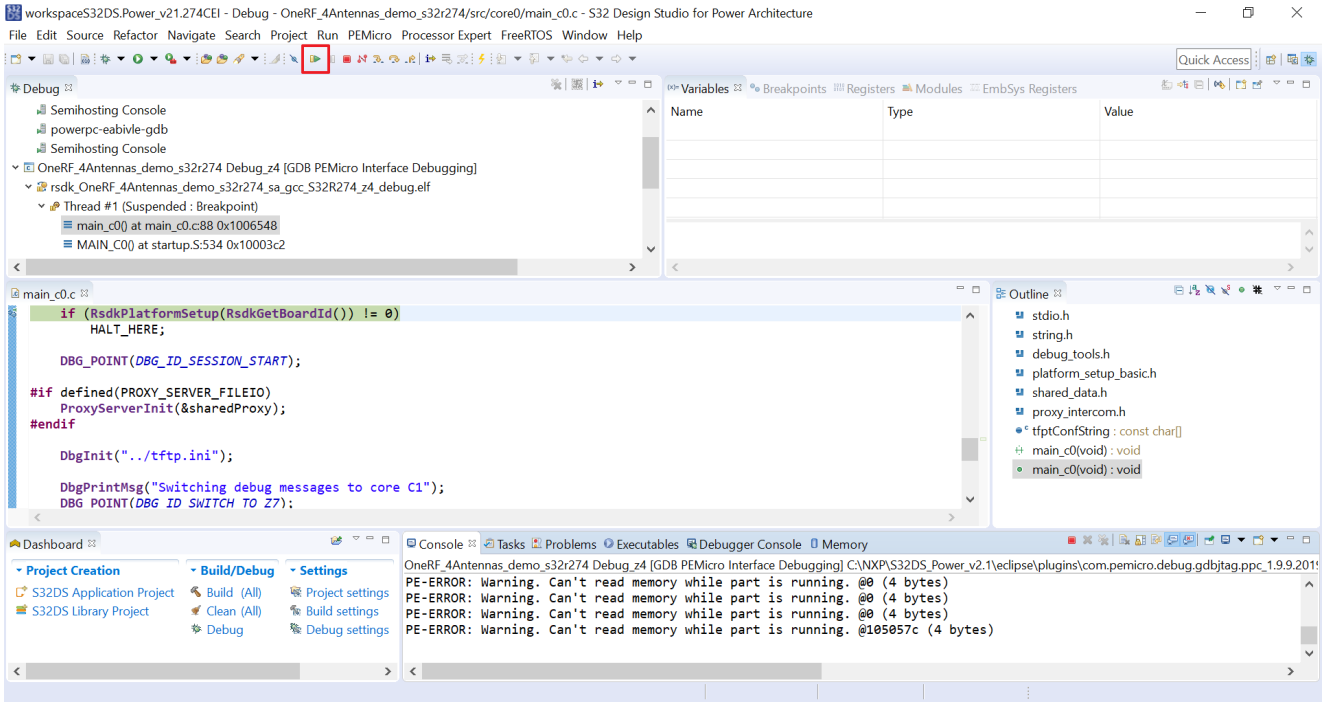


Figure 30. Running demo at full speed

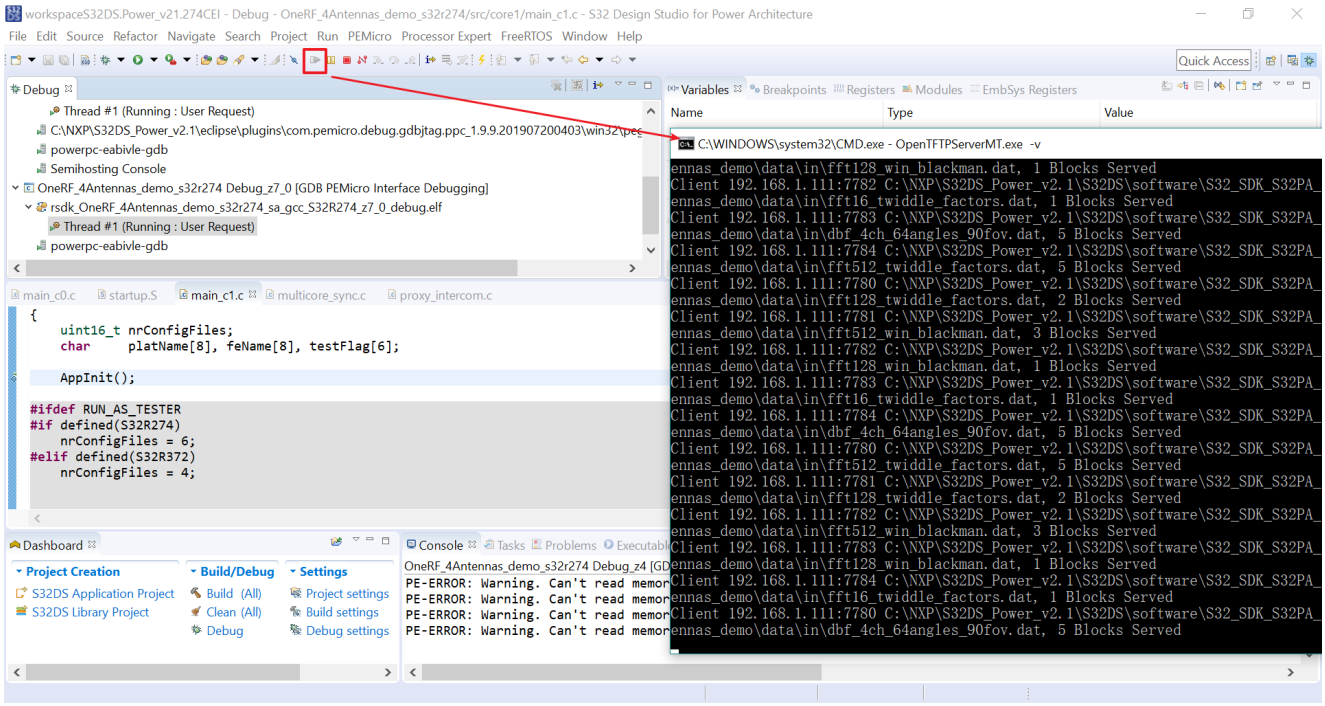


Figure 31. Data transfer information

NOTE

OpenTFTPServer reads the parameter file from the computer to the RDK platform, and uploads the data results from the RDK platform to the host computer through the RTP protocol, as shown in Figures 4-17.

Find the RunRadarApp.m script under rsdk \ Apps \ OneRF_4Antennas_demo \ matlab.

Start the matlab software and install ***NXP_RADAR_Toolbox_S32R_1.3.0.***

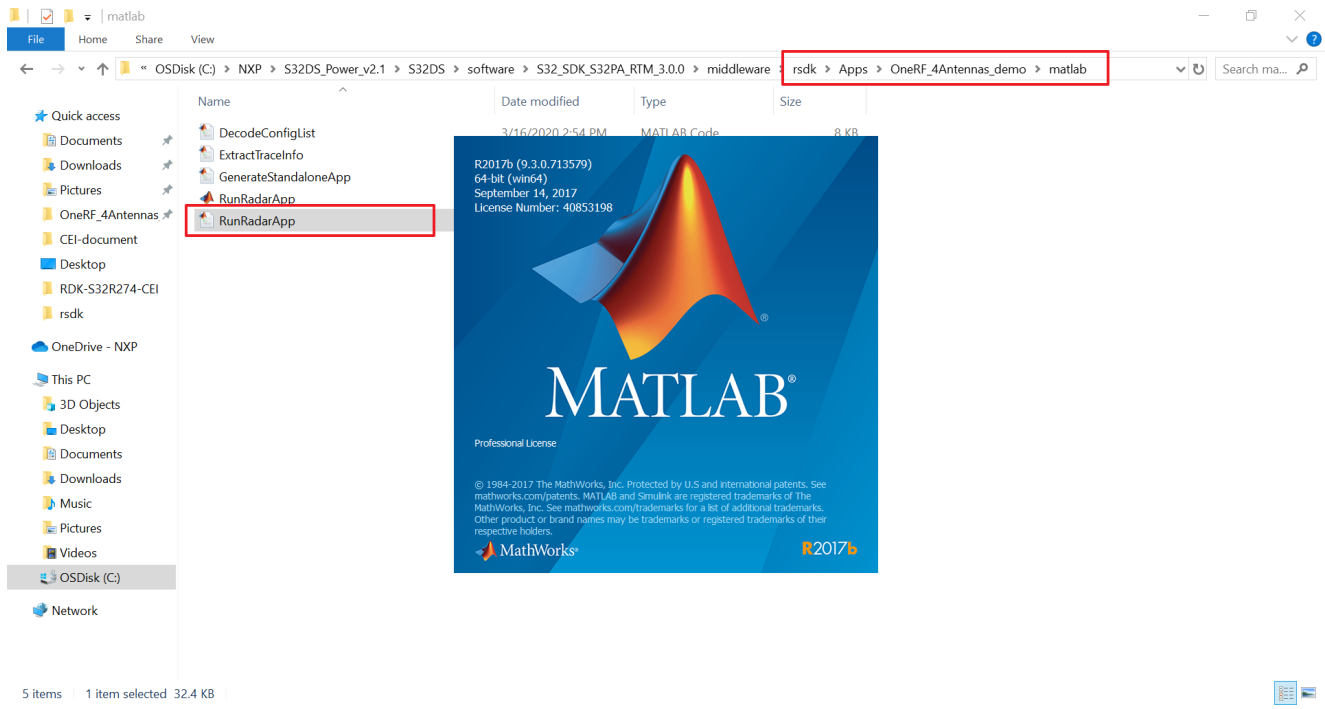


Figure 32. Start matlab software

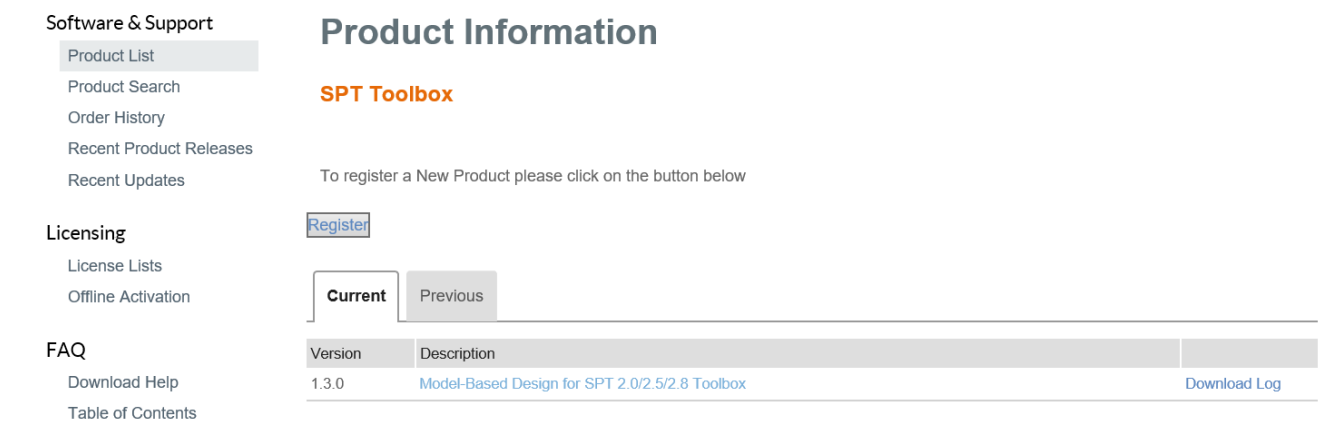


Figure 33. NXP_RADAR_Toolbox_S32R_1.3.0 installation

NOTE

NXP_RADAR_Toolbox_S32R_1.3.0 can be downloaded from your NXP software account (as shown in the figure below). If you cannot get it, please contact FAE.

Run the RunRadarApp.m script at full speed. The data is visualized as shown in the following figure.

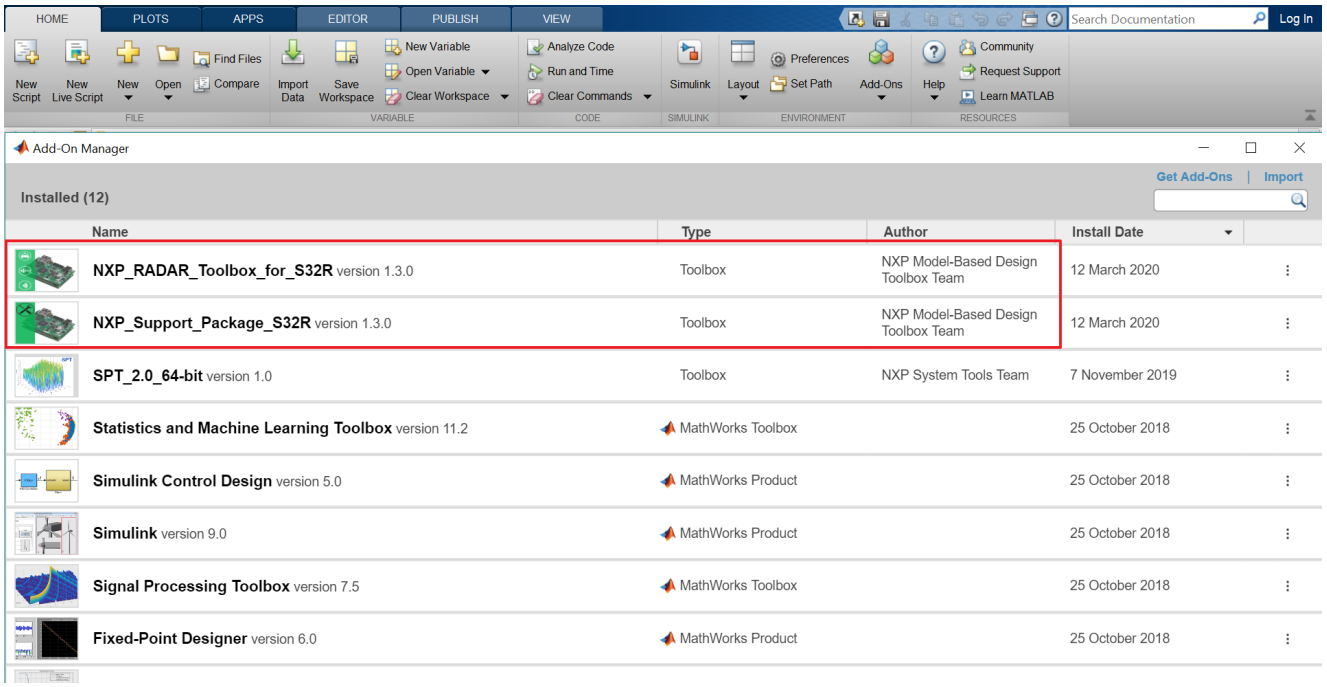


Figure 34. Run the RunRadarApp.m script at full speed

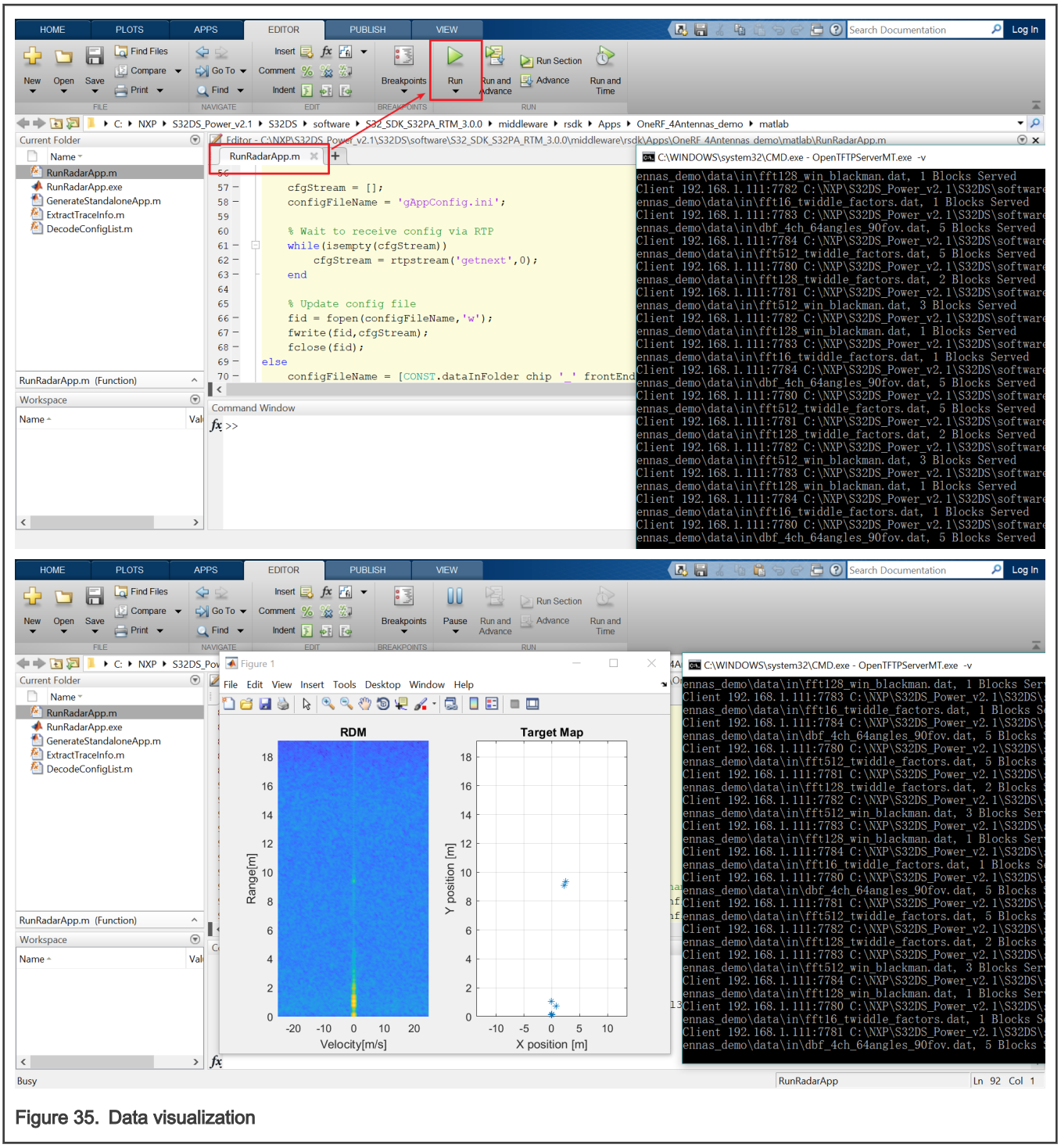


Figure 35. Data visualization

The visualized data in the above figure includes the Range-Doppler magnitude matrix and Peak list, which are set by the configuration file shown in .

Error handling of MTALB Script:

- Encountered Error : error readIniFile (line 44) nline(nline == '') = [];

  Please replace this code with nline(cellfun(@isempty,nline))=[];

- Encountered Error : error DecodeConfigList (line 23) switch paramName

Please replace this code with paramName = char(configList{i});

4.5 OneRF_4Antennas_demo analysis

The red part of the following figure is the code that the bare-metal code application project must be contained. This code is located in rsdk \ platform_setup \ src \ PPC as shown in the following figures.

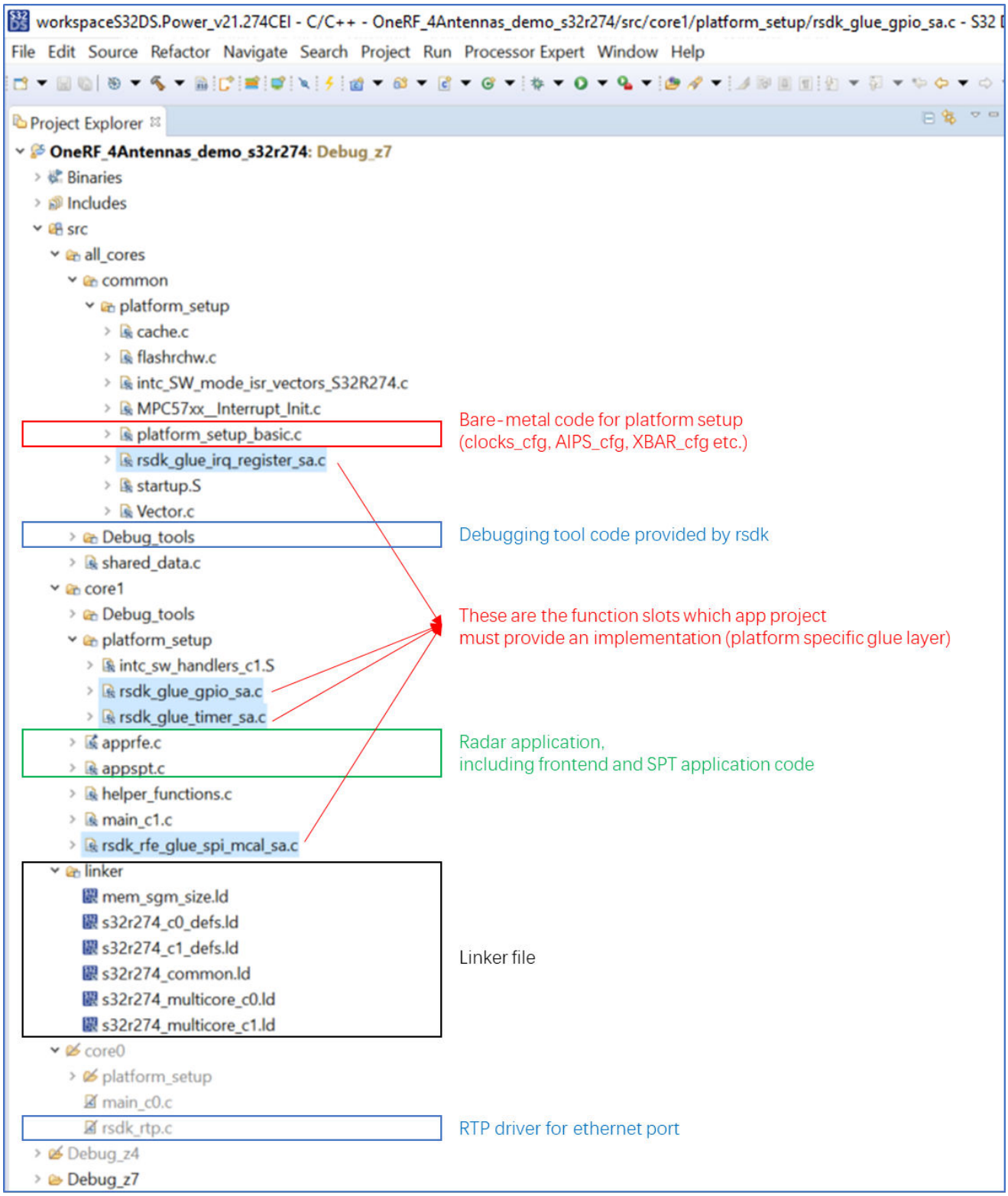


Figure 36. Demo analysis

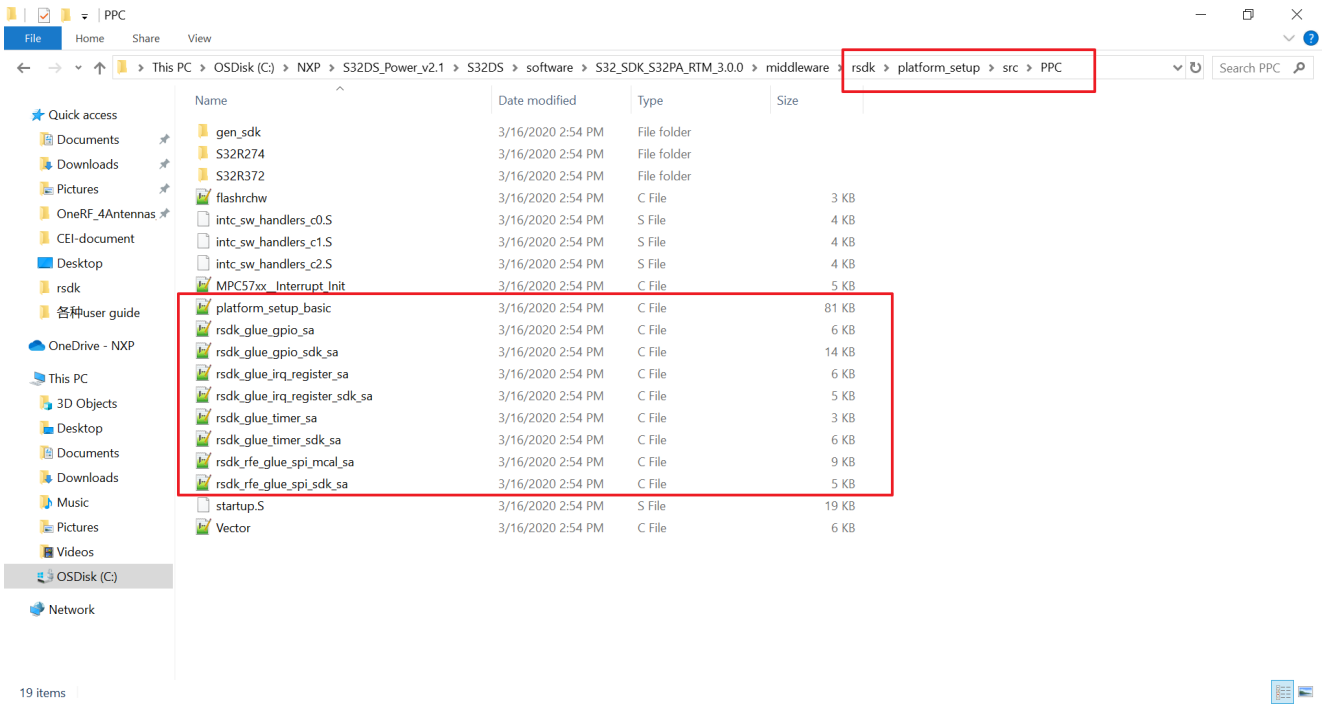


Figure 37. Platform specific glue layer

rsdk_glue_XXX_sa is the code that must be added for a bare-metal code application.

rsdk_glue_XXX_sdk_sa is the code that must be added S32DS when using platform SDK.

The two codes are written in different style. ***rsdk_glue_XXX_sdk_sa*** will be explained in the RSDK_demo chapter.

Use OneRF_4Antennas_demo as an example to illustrate how library API headers and pre-built libraries (.a files) are added to the this demo project.

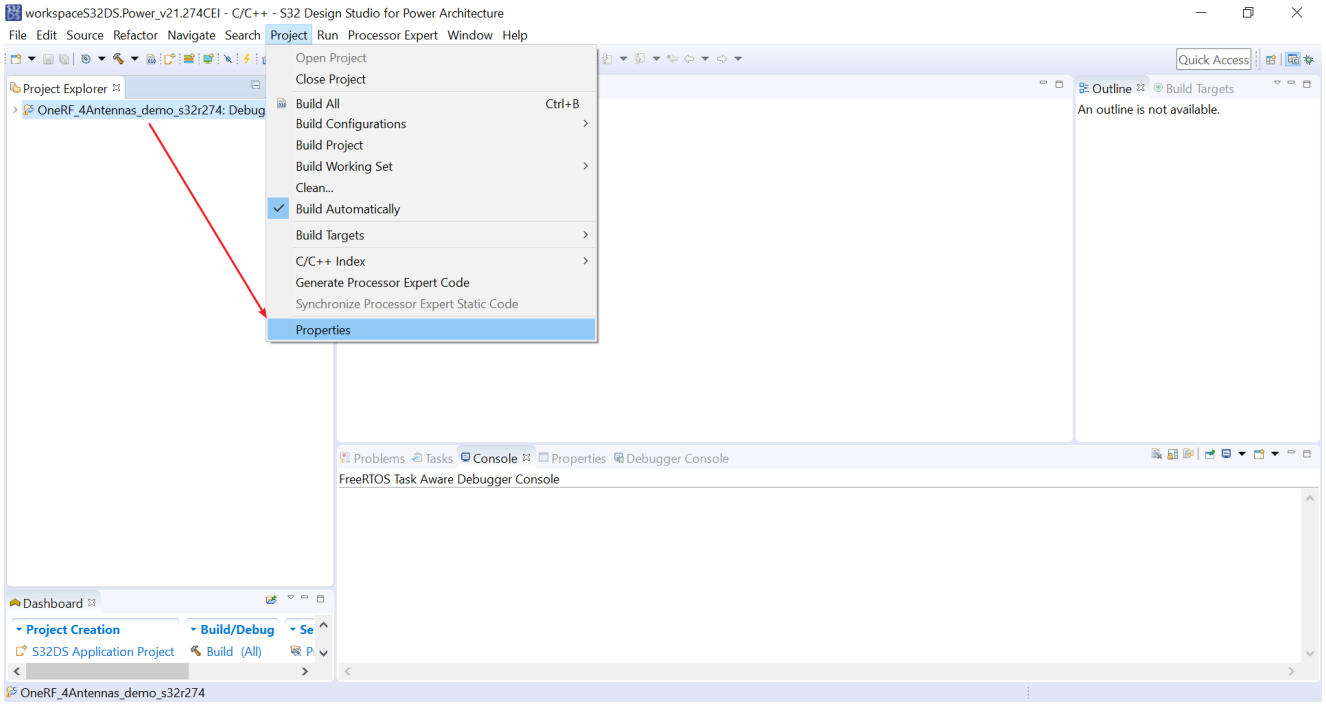


Figure 38. Properties for demo

The configuration of the Z4 project is shown in the following figures.

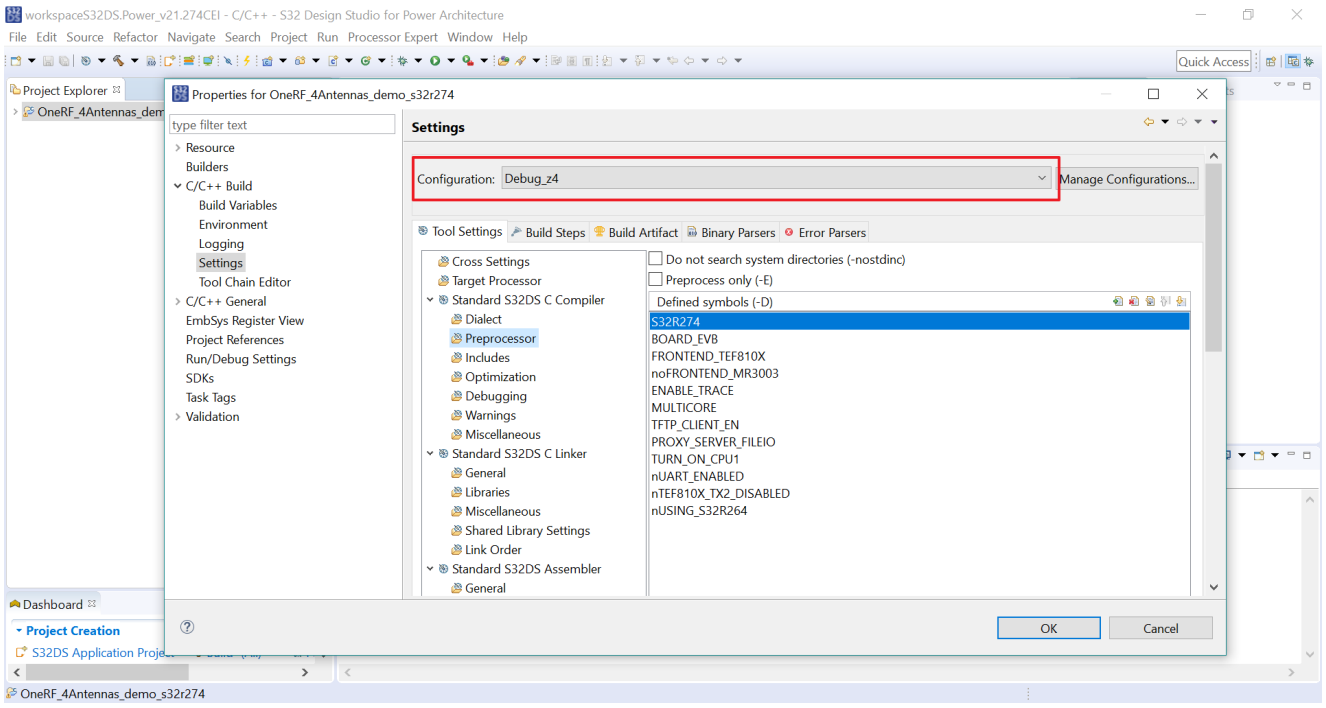


Figure 39. Defined symbols

The library API headers are added as shown in the figure below, which is mainly located in the api folder in rsdk path.

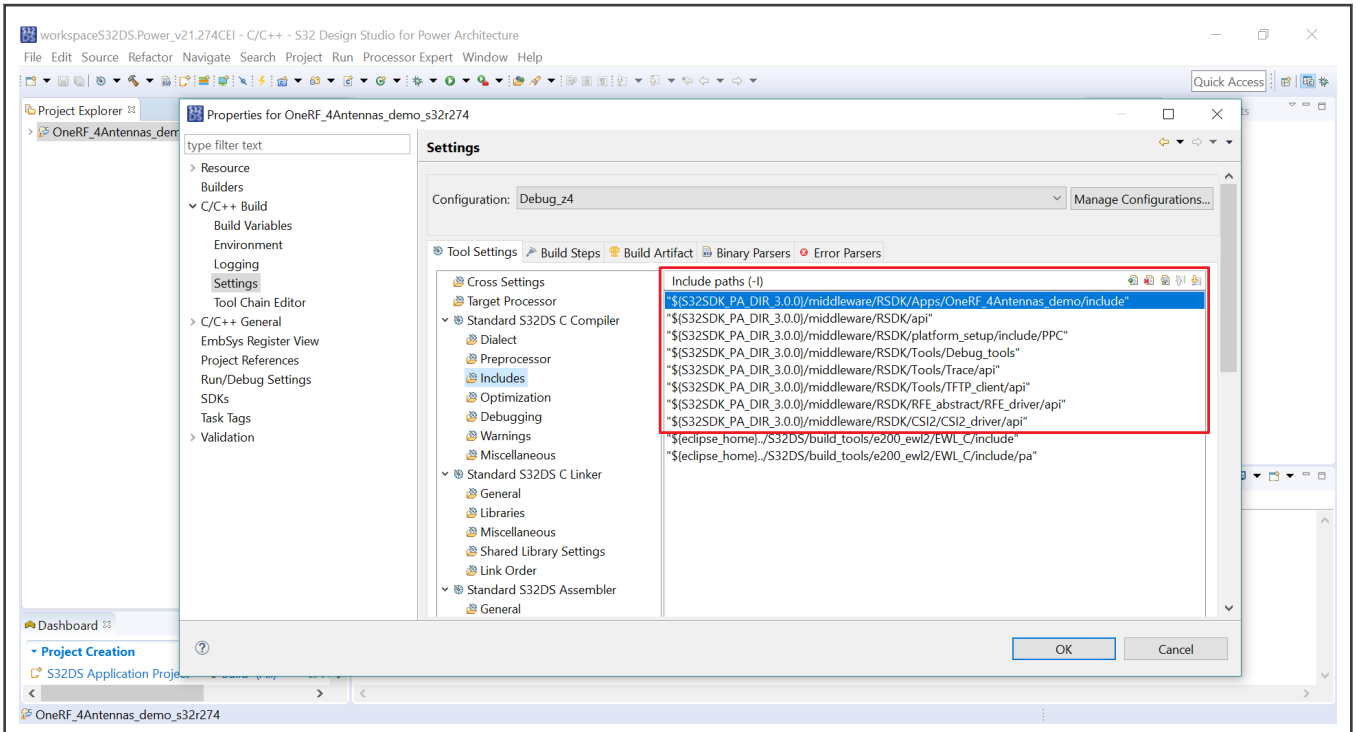


Figure 40. Library API headers

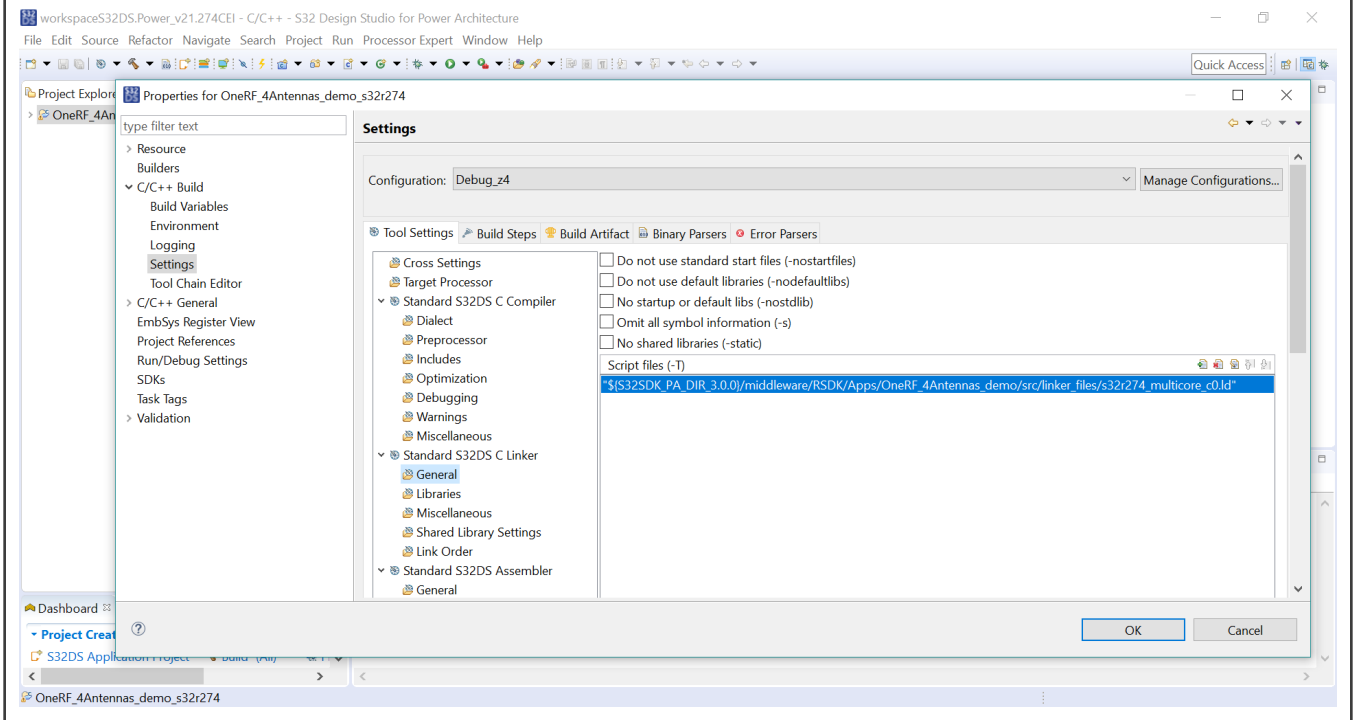


Figure 41. Linker file

The pre-built libraries are added as shown in the following figure, which is mainly located in the bin folder in rsdk path.

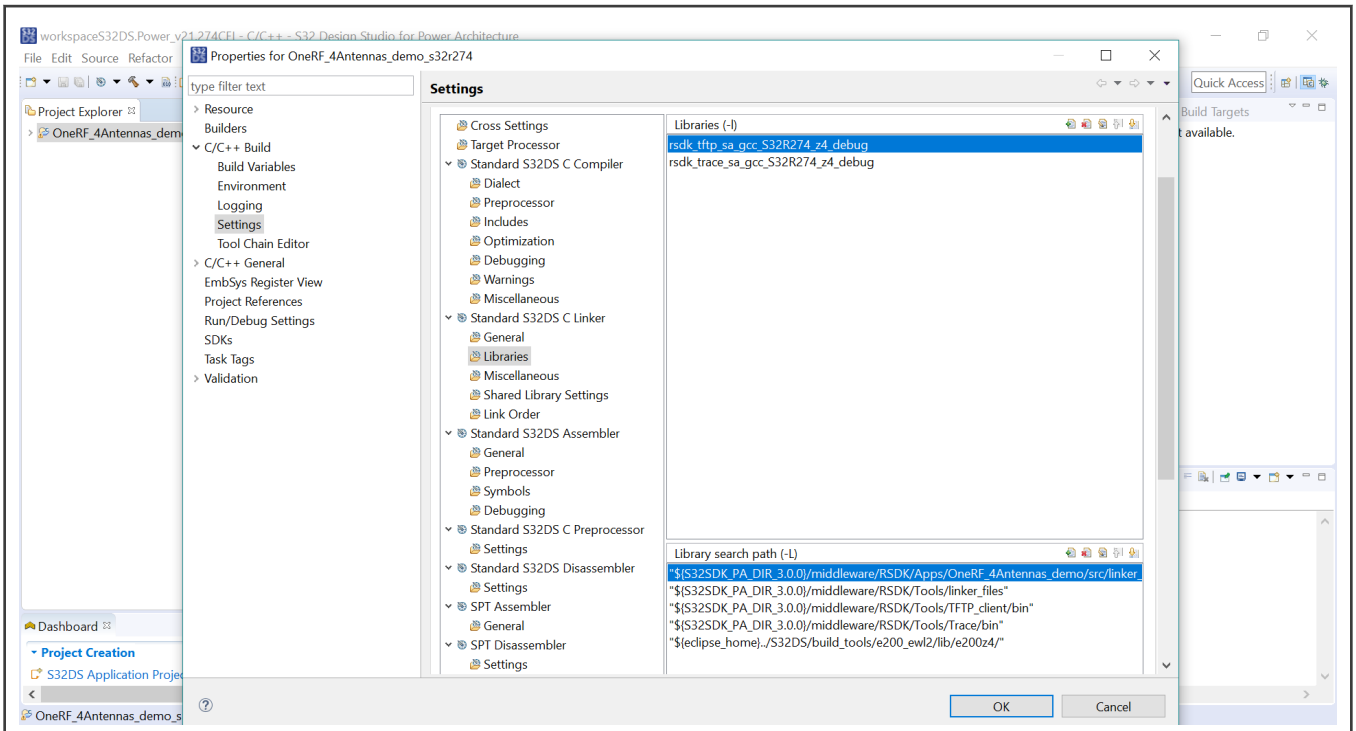


Figure 42. Pre-built libraries

The configuration of the Z7 project is shown in the following figures.

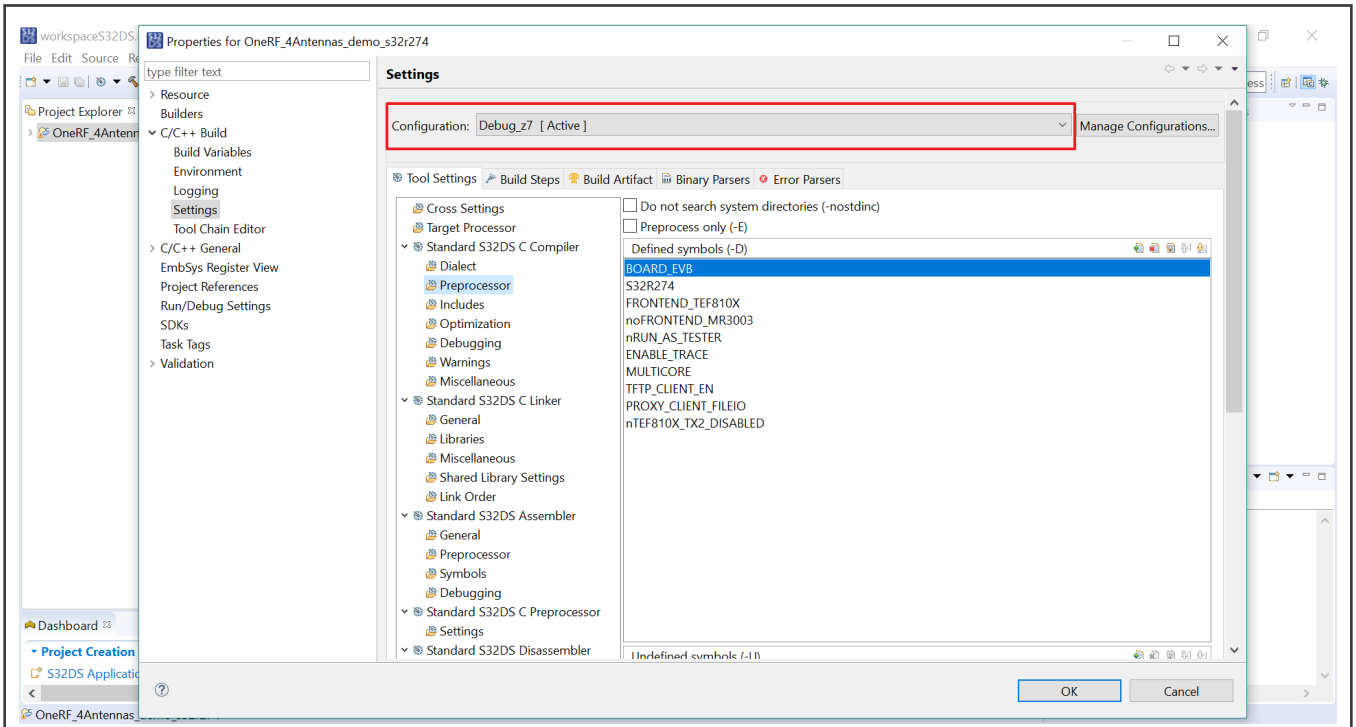


Figure 43. Defined symbols

The library API headers are added as shown in the figure 4-30, which is mainly located in the api folder in rsdk path.

The radar processing flow is located in the Z7 core, so the settings is different with the Z4 core project.

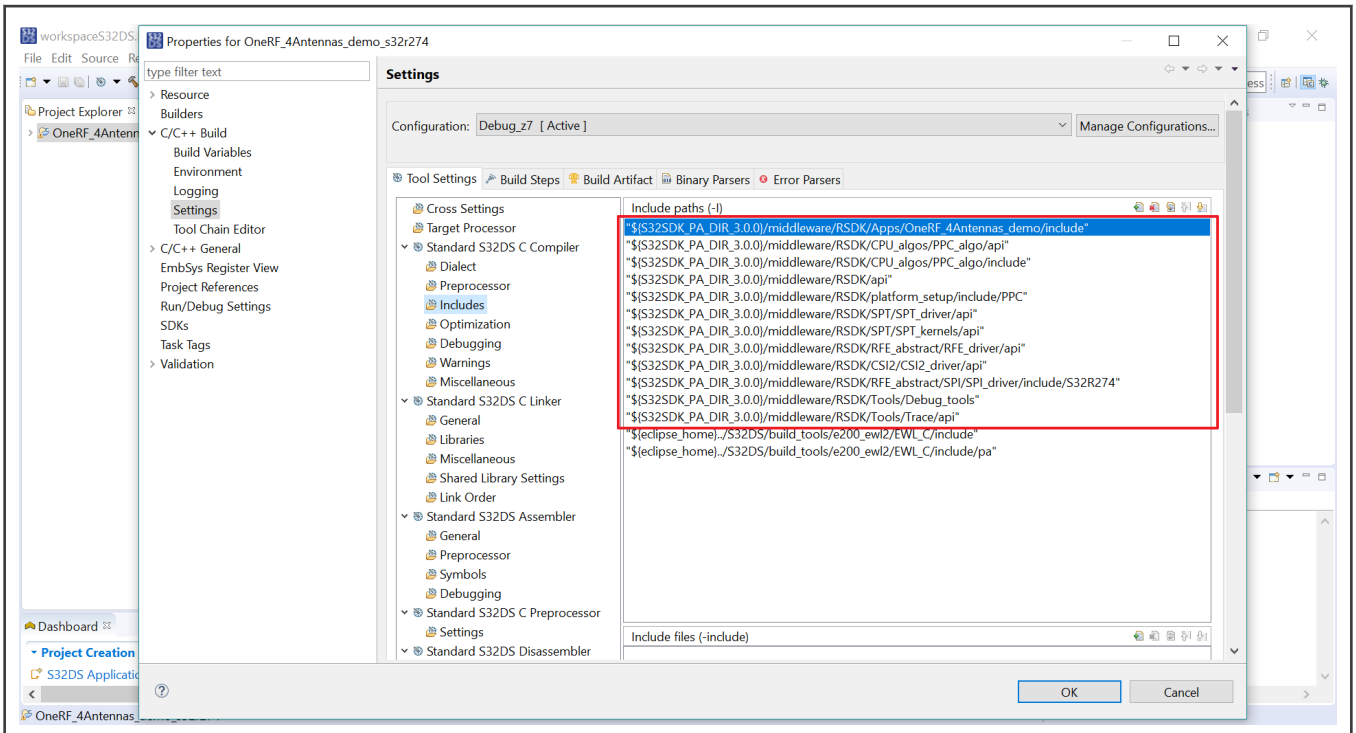


Figure 44. Library API headers

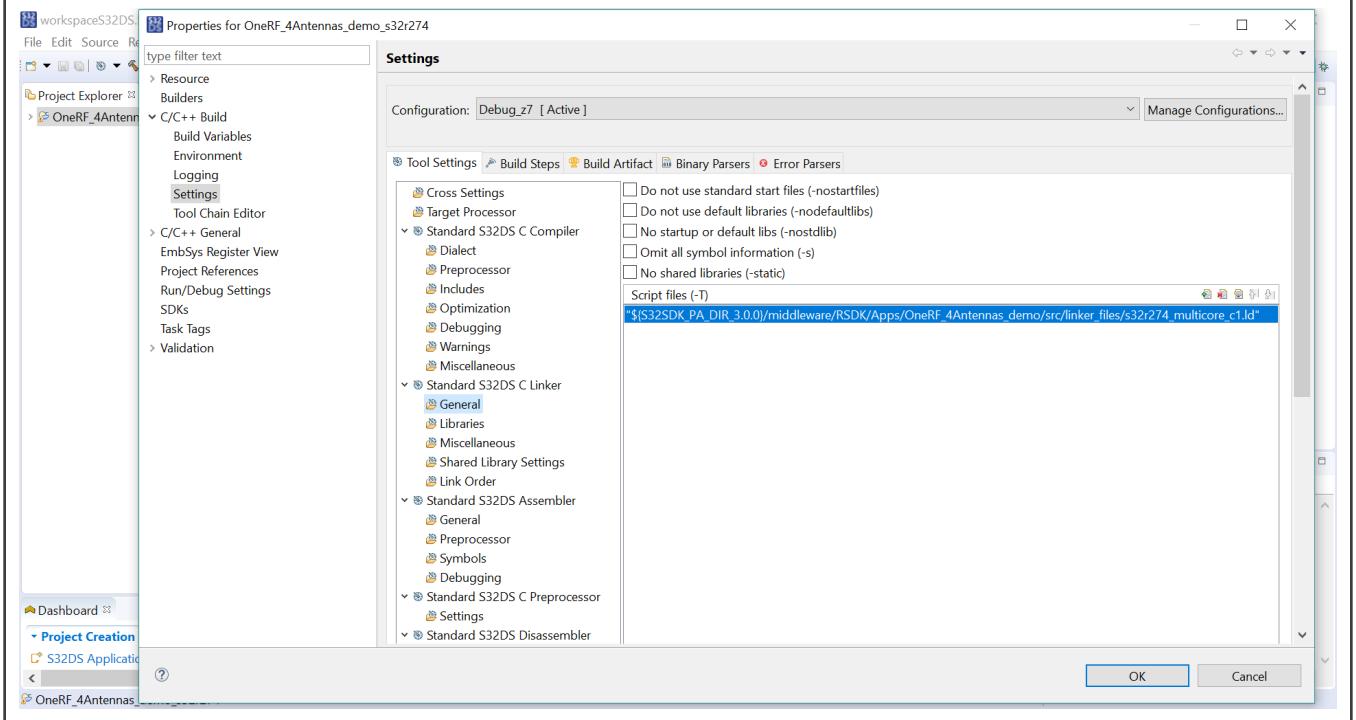


Figure 45. Linker file

The radar processing flow is located in the Z7 core, so the settings is different with the Z4 core project.

Key pre-built libraries for radar processing are added, as shown in the following figure.

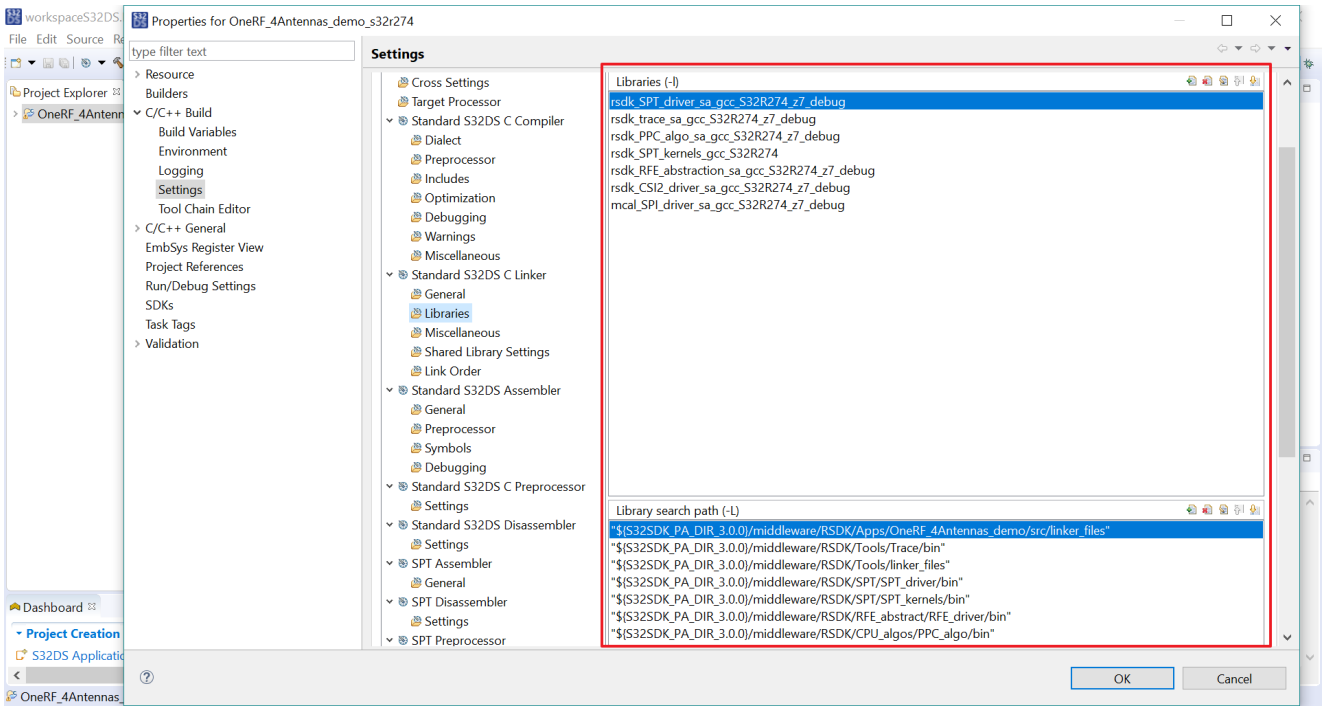


Figure 46. Pre-built libraries

4.6 Create bare code project and add RSDK

If you follow the steps in S32 Design studio download and S32 Design Studio Installation to install RSDK 1.4.0, then you can add library API headers and pre-built libraries (.a files) to your project very easy. We can add RSDK when creating a project or add RSDK to an existing project.

The steps for creating a new project and add RSDK are shown in the following figures.

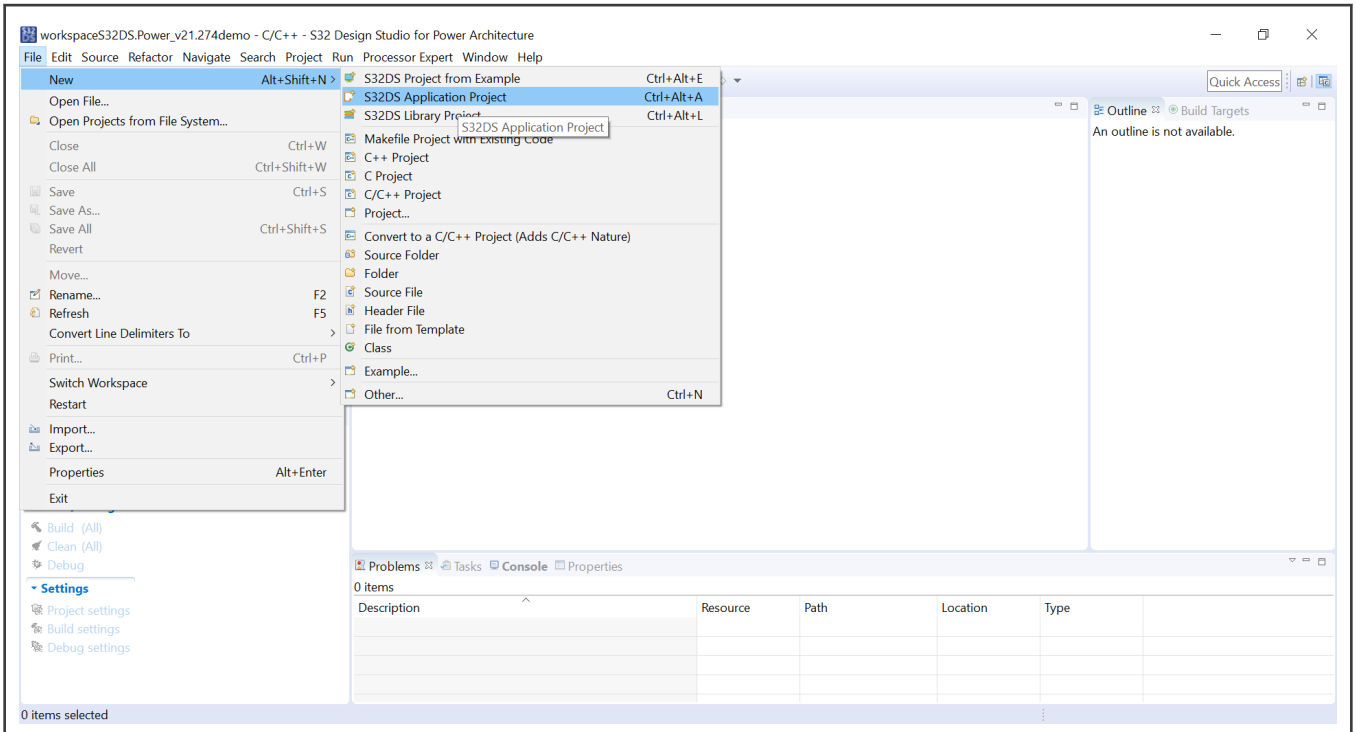


Figure 47. Create new application project

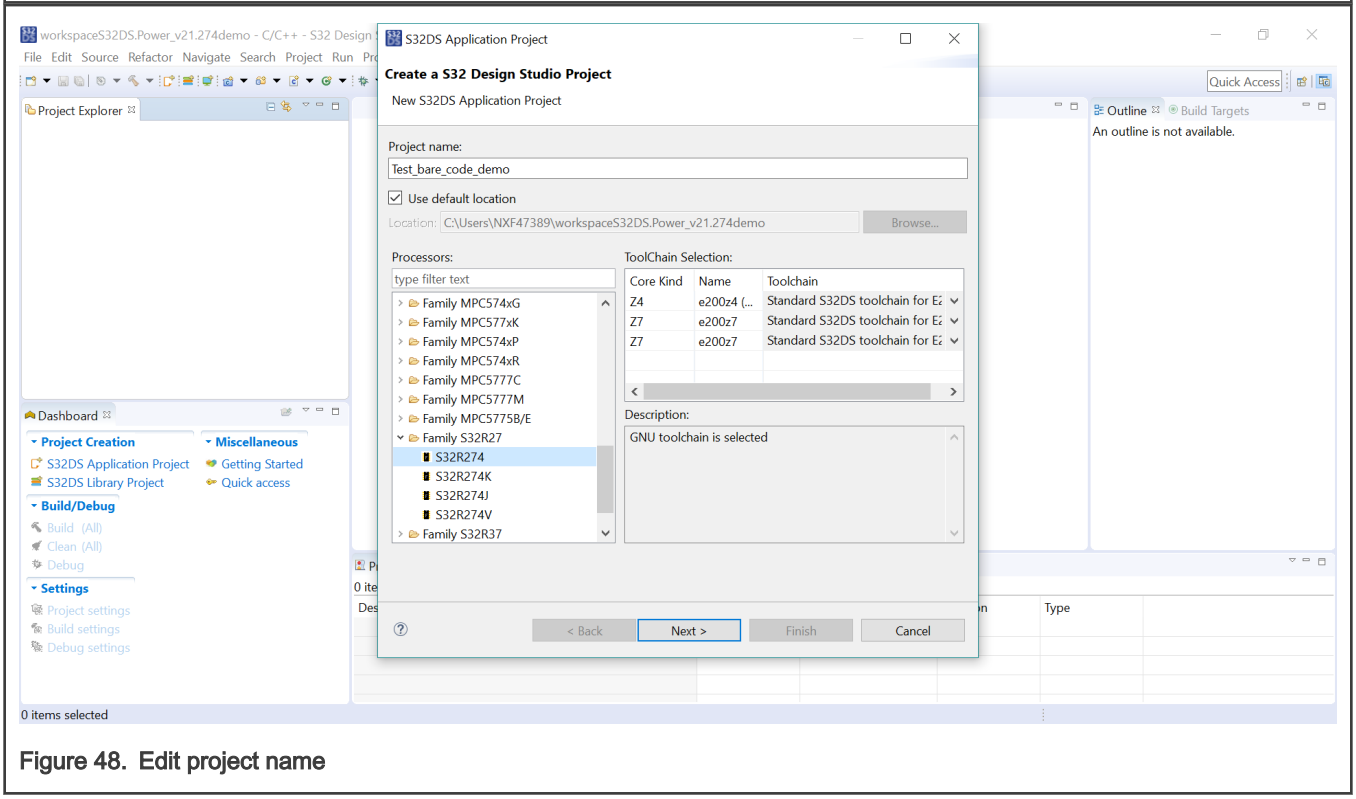


Figure 48. Edit project name

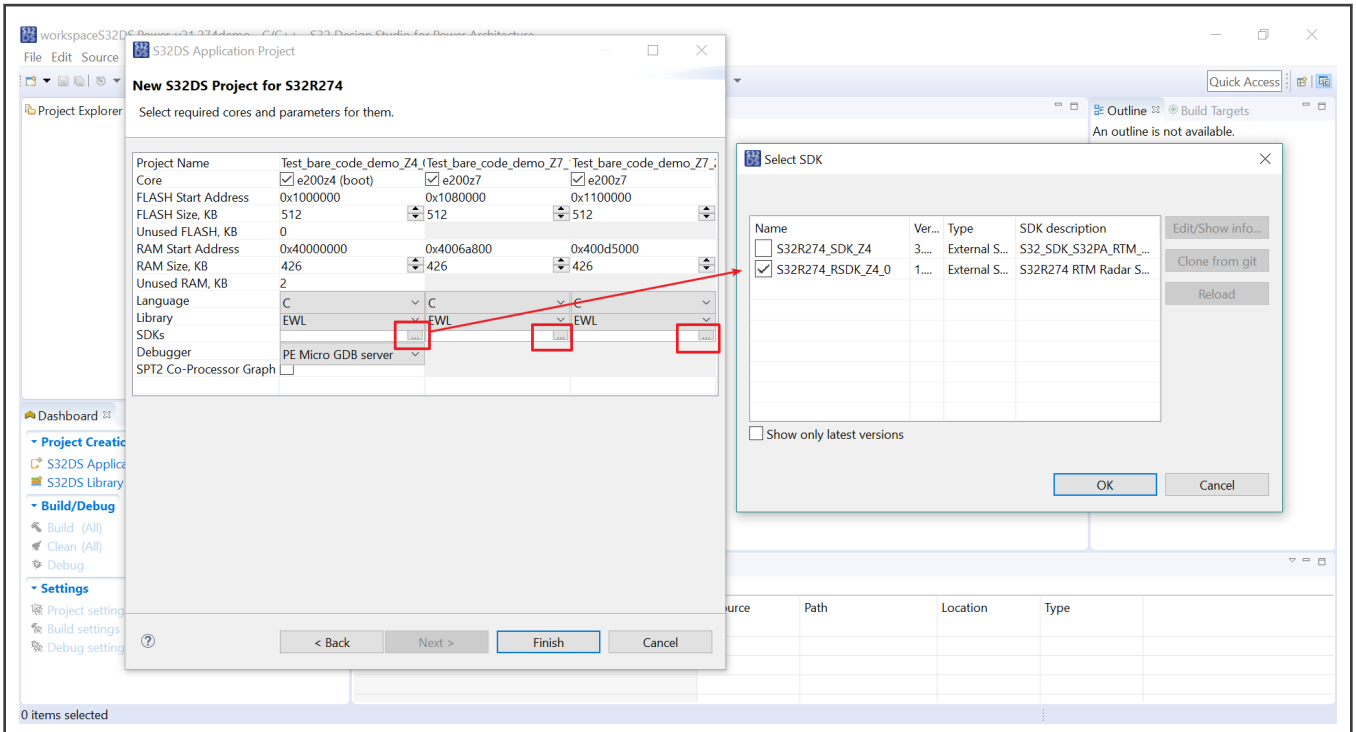


Figure 49. Add RSDK modules to each project

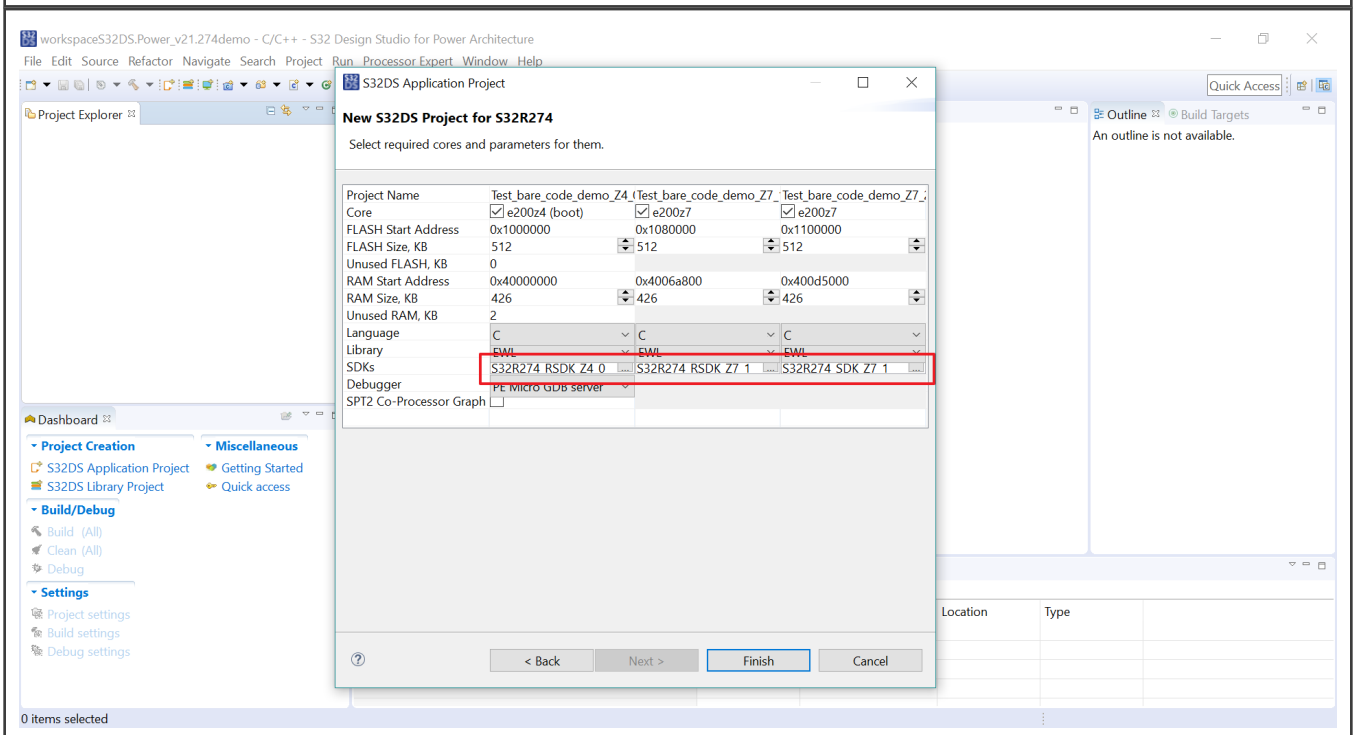


Figure 50. Complete project creation

To add RSDK to an existing project are shown in the following figures:

1. Left click on your project folder
2. Clicking “SDKs” on the menu

3. Selecting RSDK with latest version (1.4.0)
4. Adding RSDK to the configurations you want

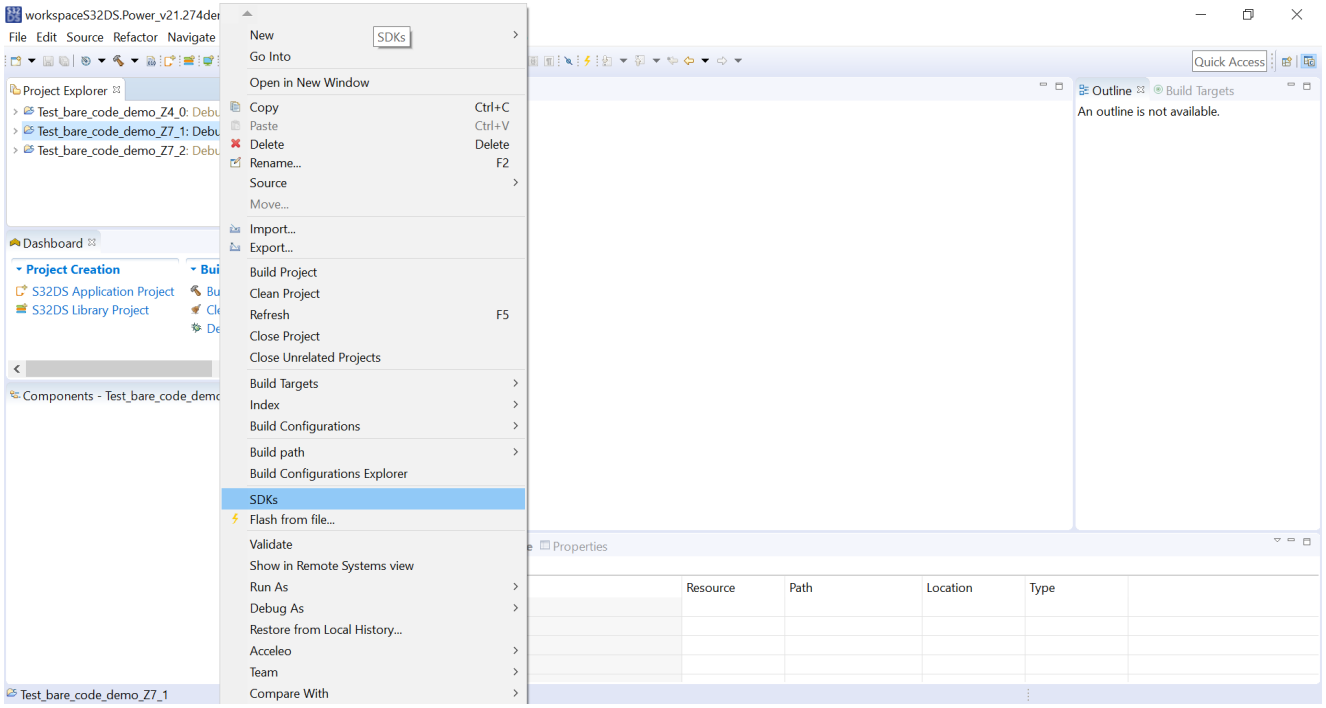


Figure 51. Clicking “SDKs” on the menu

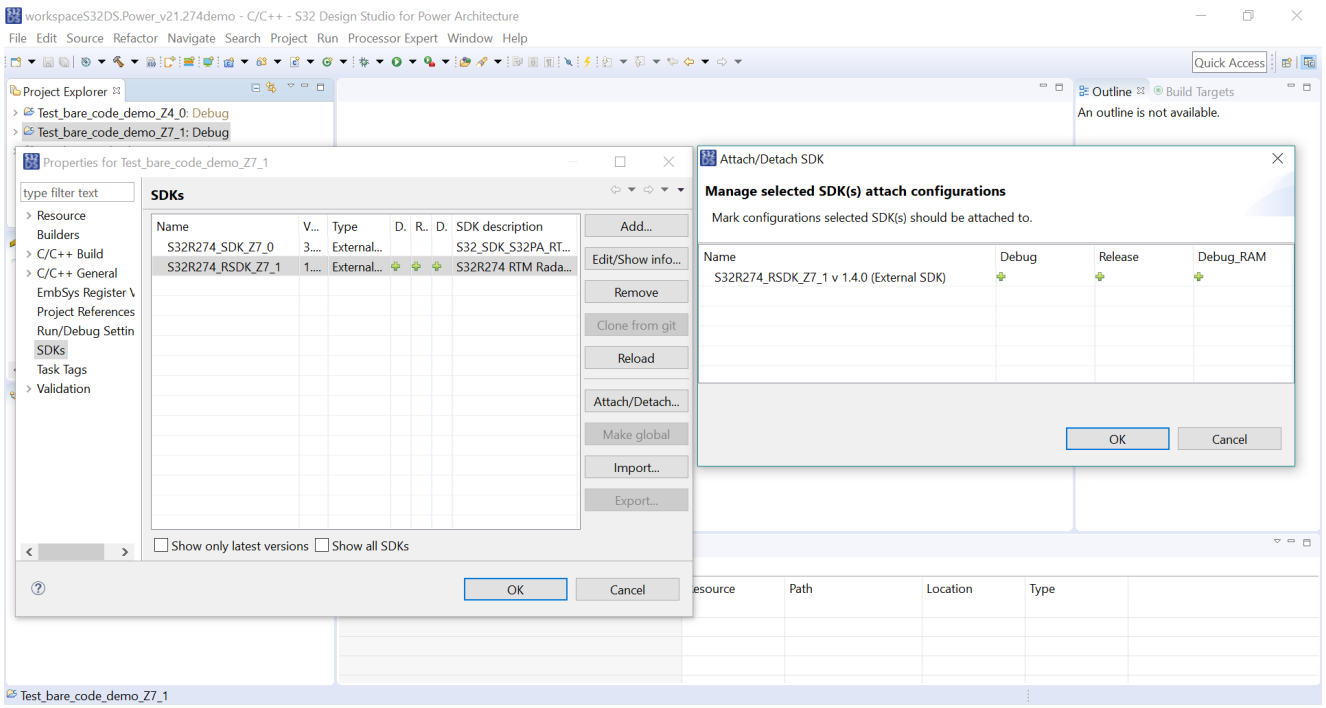


Figure 52. Adding RSDK to the configurations

The above two methods can add library API headers and pre-built libraries (.a files) to the project, and the results are shown in the following figures.

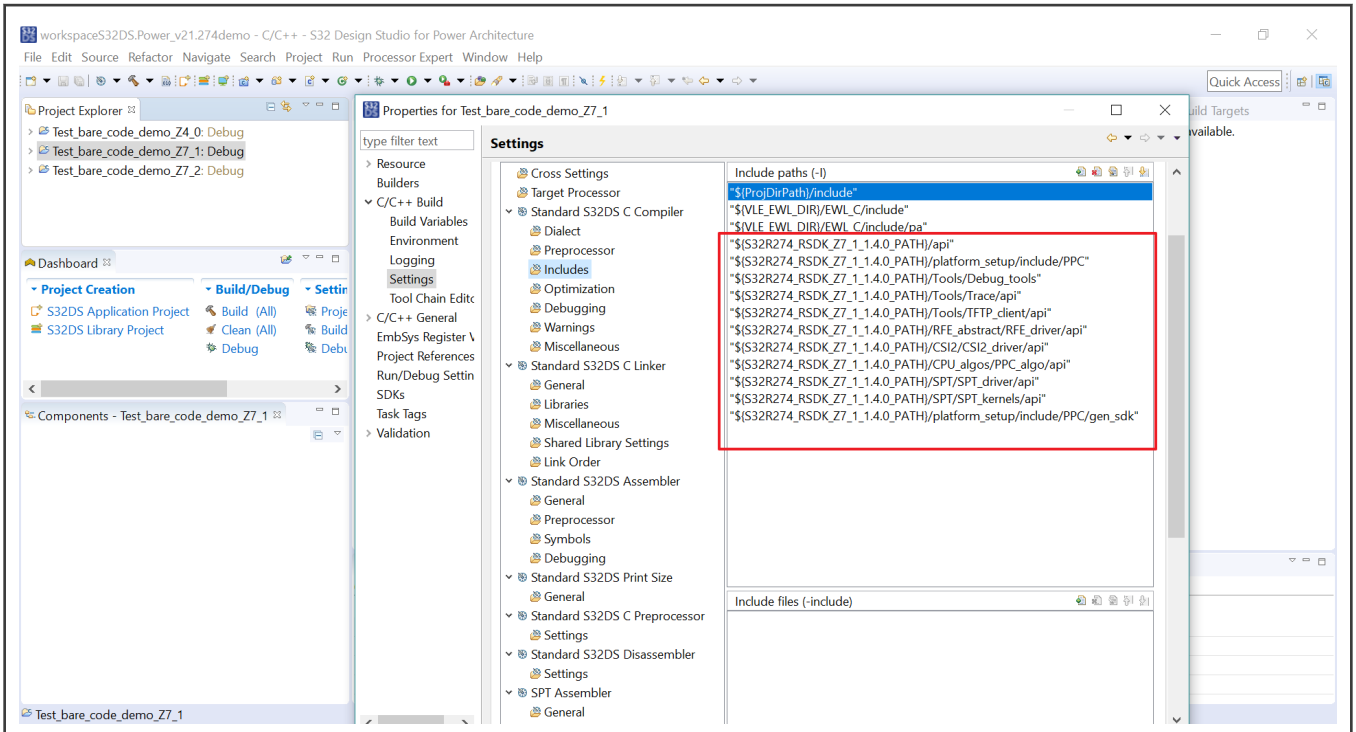


Figure 53. Library API headers

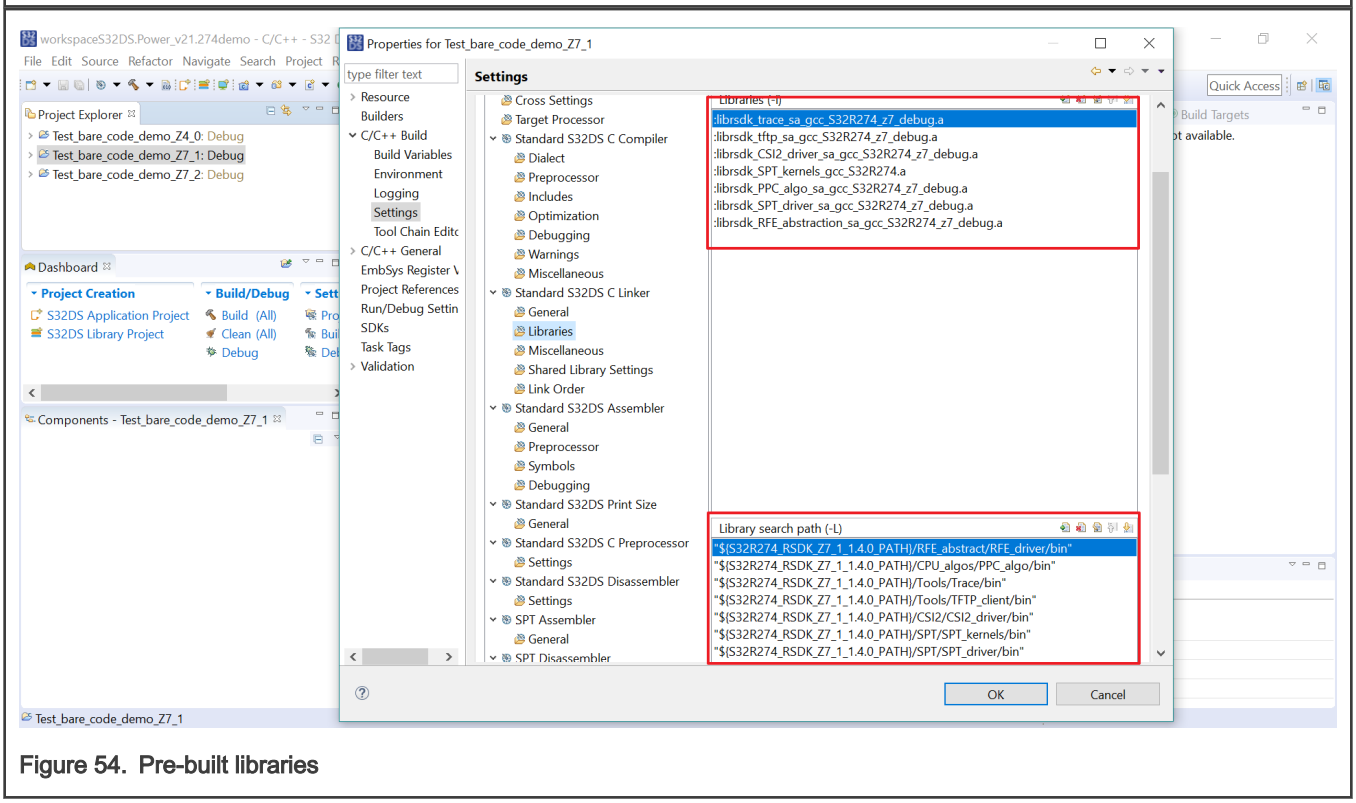


Figure 54. Pre-built libraries

Adding RSDK in non-S32DS IDE

Refer to the library API headers and pre-built libraries (.a files) shown in the following figures.

1. Copy header files and library files from RSDK installation folder to IDE project.

2. Then invoking RSDK API functions directly.

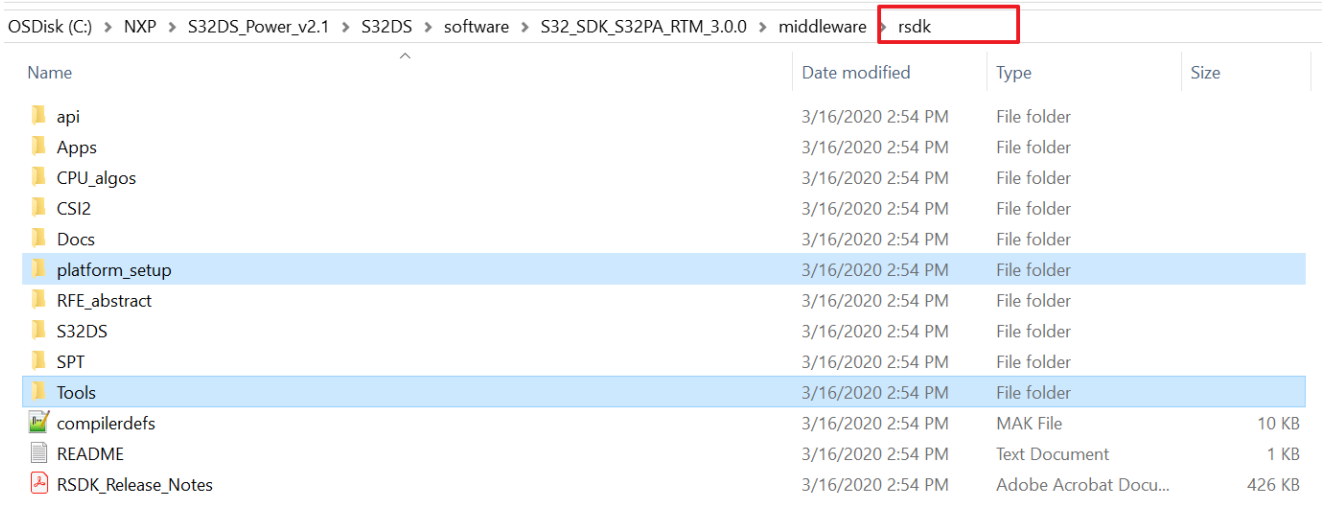


Figure 55. RSDK components

NOTE

Platform-setup (GPIO driver, Interrupt service, Timer functions, SPI driver, MCU platform configuration) and Tools code, as shown in the figure above refer to OneRF_4Antennas_demo analysis to add to your own project according to your needs.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: July, 2020
Document identifier: AN12938