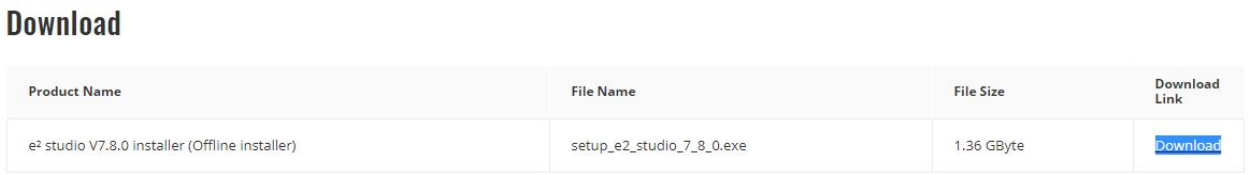
1. Install Renesas e2studio and Renesas FSP v1.1.0

1.1 Download the Renesas e2studio (v7.8.0) installer:
https://www.renesas.com/in/en/software/D4001318.html



Download

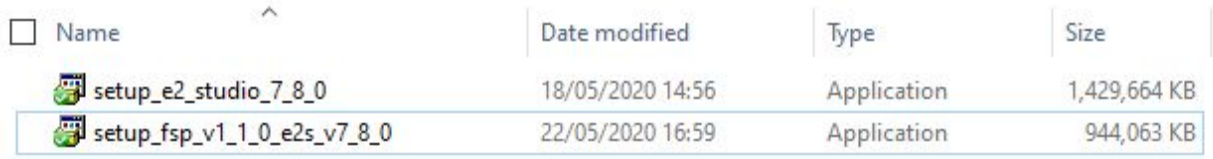
Product Name	File Name	File Size	Download Link
e² studio V7.8.0 installer (Offline installer)	setup_e2_studio_7_8_0.exe	1.36 GByte	Download

1.2 Download the Flexible Software Package (FSP) - v1.1.0 - installer:
https://www.renesas.com/eu/en/products/software-tools/software-os-middleware-driver/software-package/ra-fsp.html

1.2.1 Complete the form and proceed with the download:

1.3 Make sure you have the following installer files:

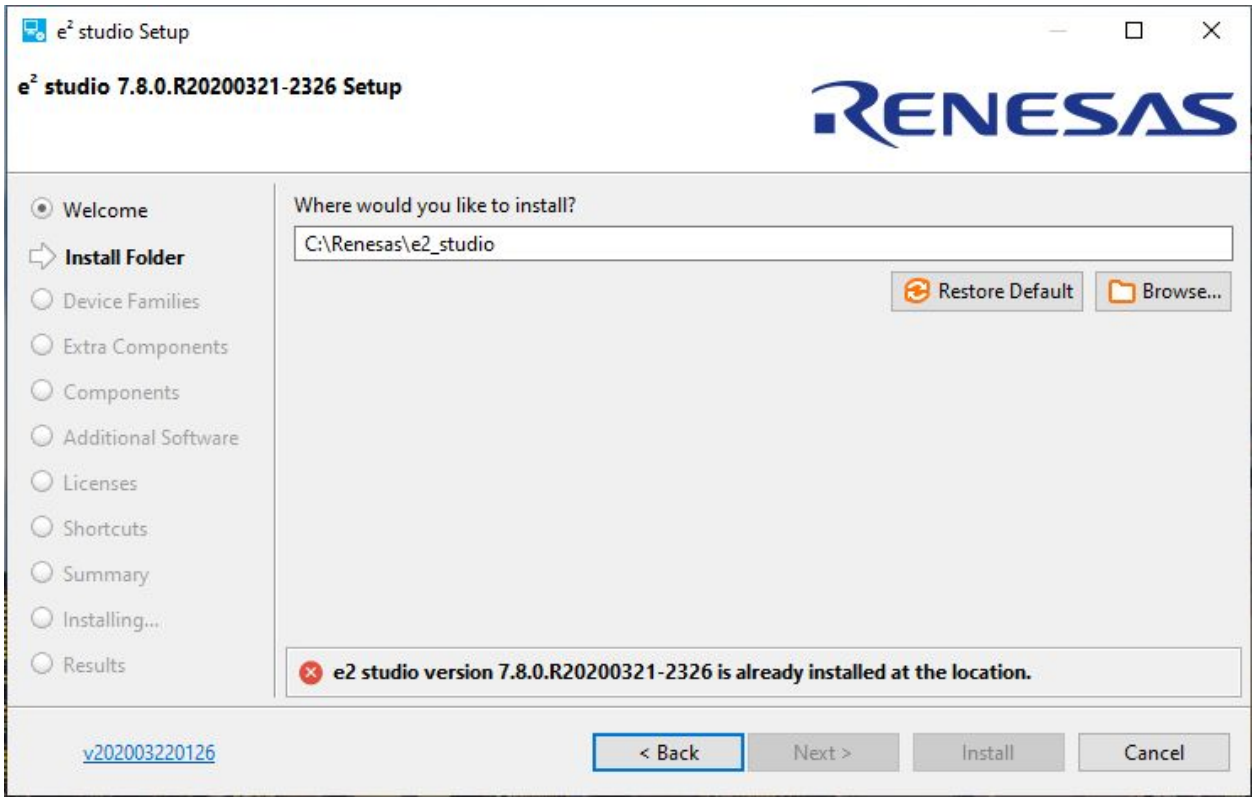
1.3.1 - setup_e2_studio_7_8_0
1.3.2 - setup_fsp_v1_1_0_e2s_v7_8_0



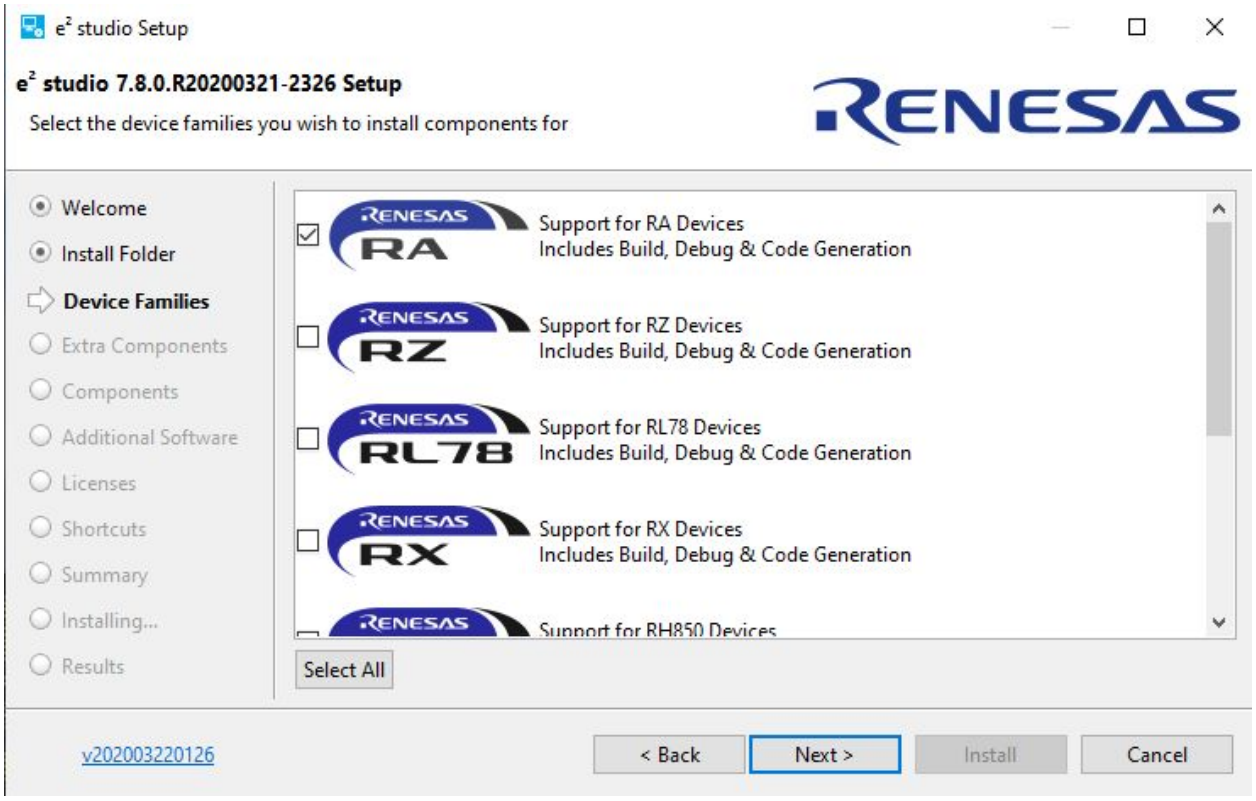
1.4 Install the Renesas e2studio (v7.8.0)

1.4.1 - Run the setup_e2_studio_7_8_0 exe.

1.4.2 - Select the install folder:



1.4.3 - Select the support for the RA Devices:



1.4.4 - Leave the additional Optional Components as selected to be installed.

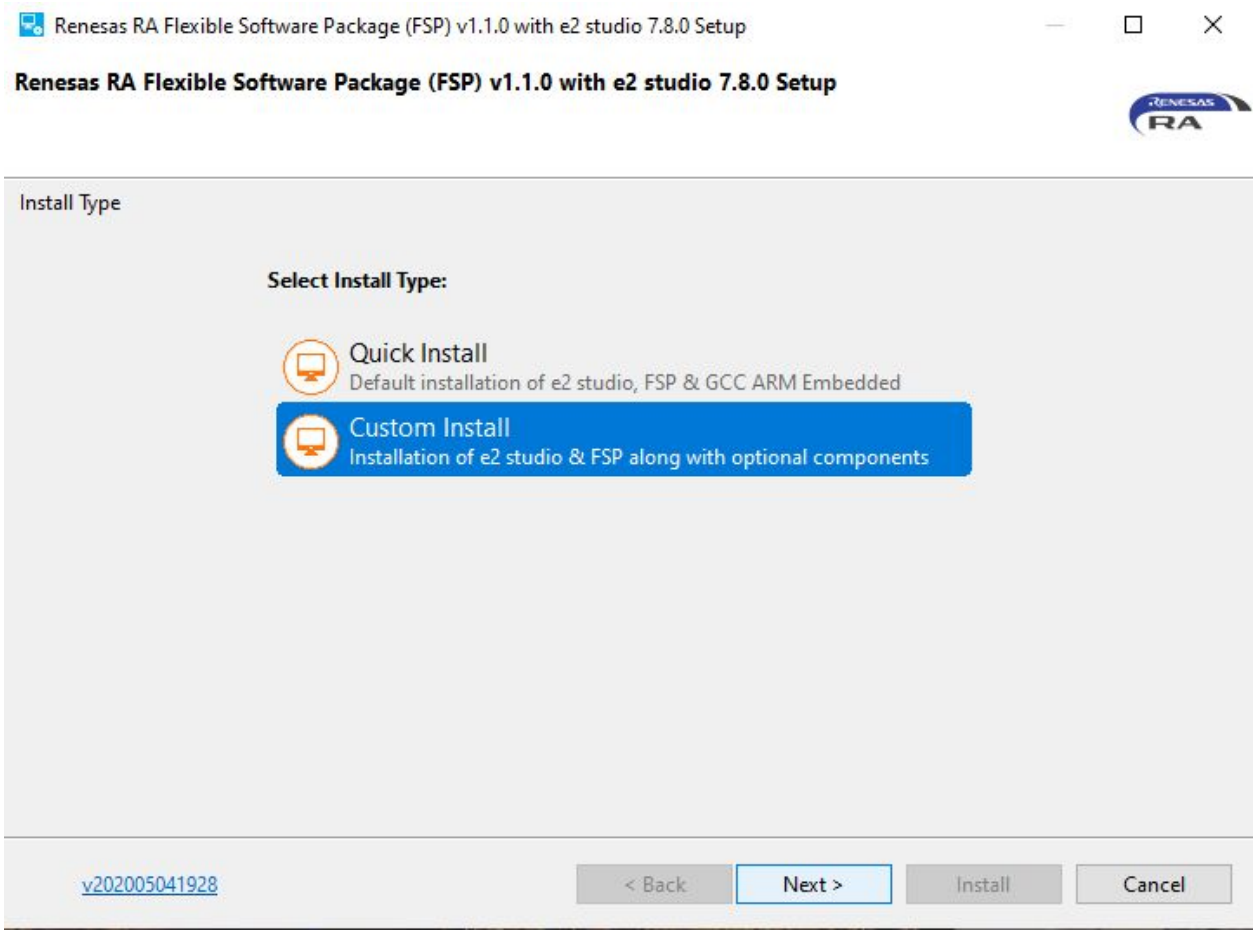
1.4.5 - Make sure to install the GCC Arm Embedded Toolchains.

1.4.6 - Go through the installation process till it finishes.

1.5 Install the Renesas e2studio (v7.8.0)

1.5.1 - Run the setup_fsp_v1_1_0_e2s_v7_8_0 exe.

1.5.2 - Select a custom install:



1.5.3 - Select the install folder:
C:\Renesas\RA\e2studio_v7.8.0_fsp_v1.1.0

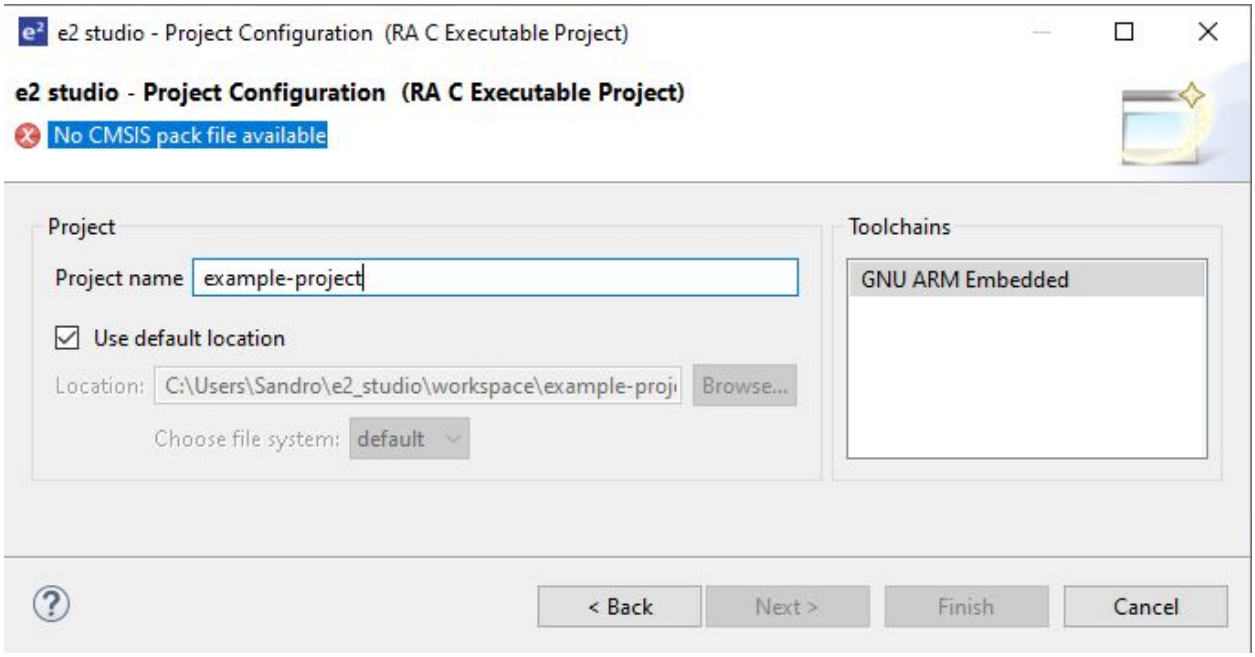
1.5.4 - Leave the additional Optional Components as selected to be installed.

1.5.5 - Make sure to install the GCC Arm Embedded Toolchains.

1.5.6 - Go through the installation process till it finishes.

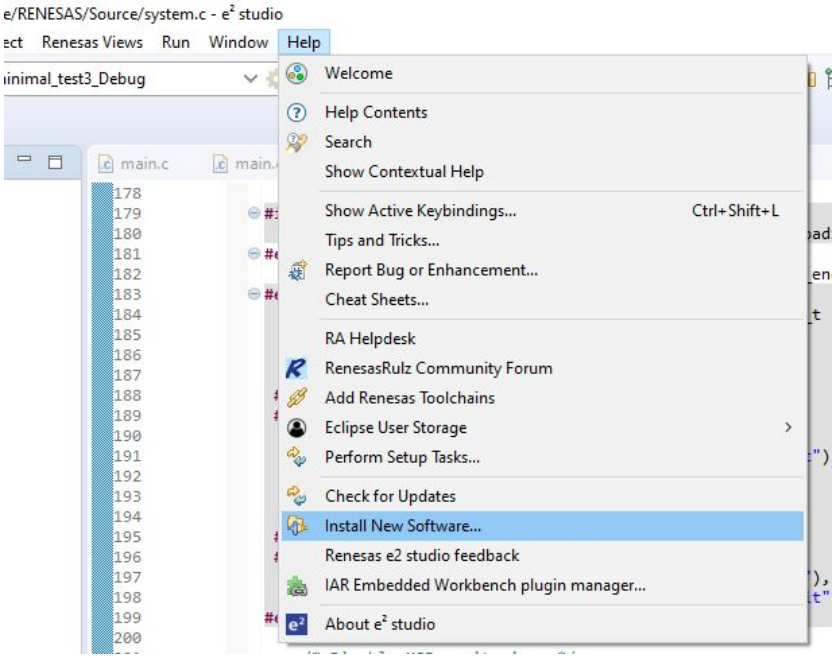
NOTE: By default, the shortcut of the e2studio App created in the Desktop is the one without the extension of the FSP 1.1.0. Make sure you run the e2studio instance available at the C:\Renesas\RA\e2studio_v7.8.0_fsp_v1.1.0\eclipse folder!

This problem is manifested when you try to create a “RA C/C++ Project” - the message “No CMSIS pack file available is displayed”:

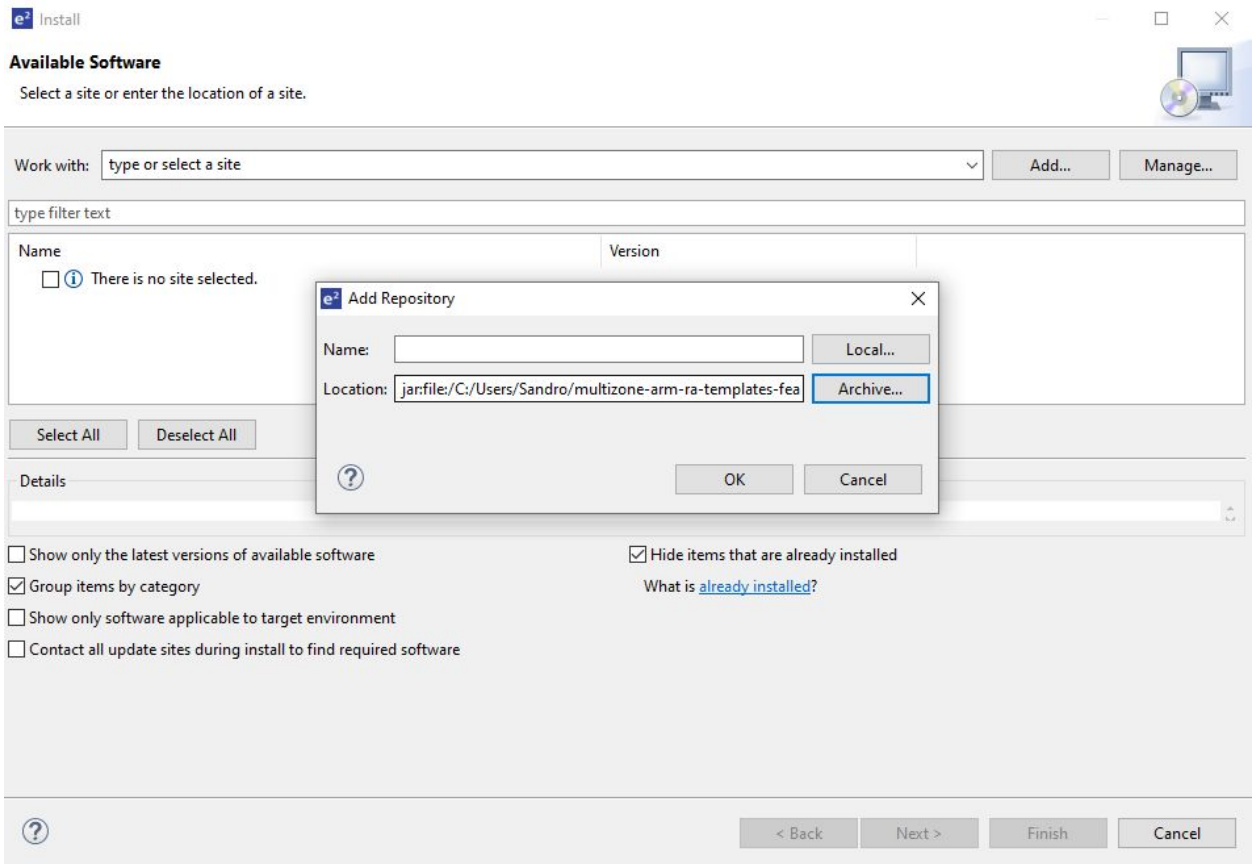

e2 studio - Project Configuration (RA C Executable Project)
e2 studio - Project Configuration (RA C Executable Project)
No CMSIS pack file available
Project
Project name
example-project
Use default location
Location:
C:\Users\Sandro\e2_studio\workspace\example-proj
Browse...
Choose file system:
default
Toolchains
GNU ARM Embedded
< Back
Next >
Finish
Cancel

2. Install MultiZone Arm (RA) Project Templates plugins

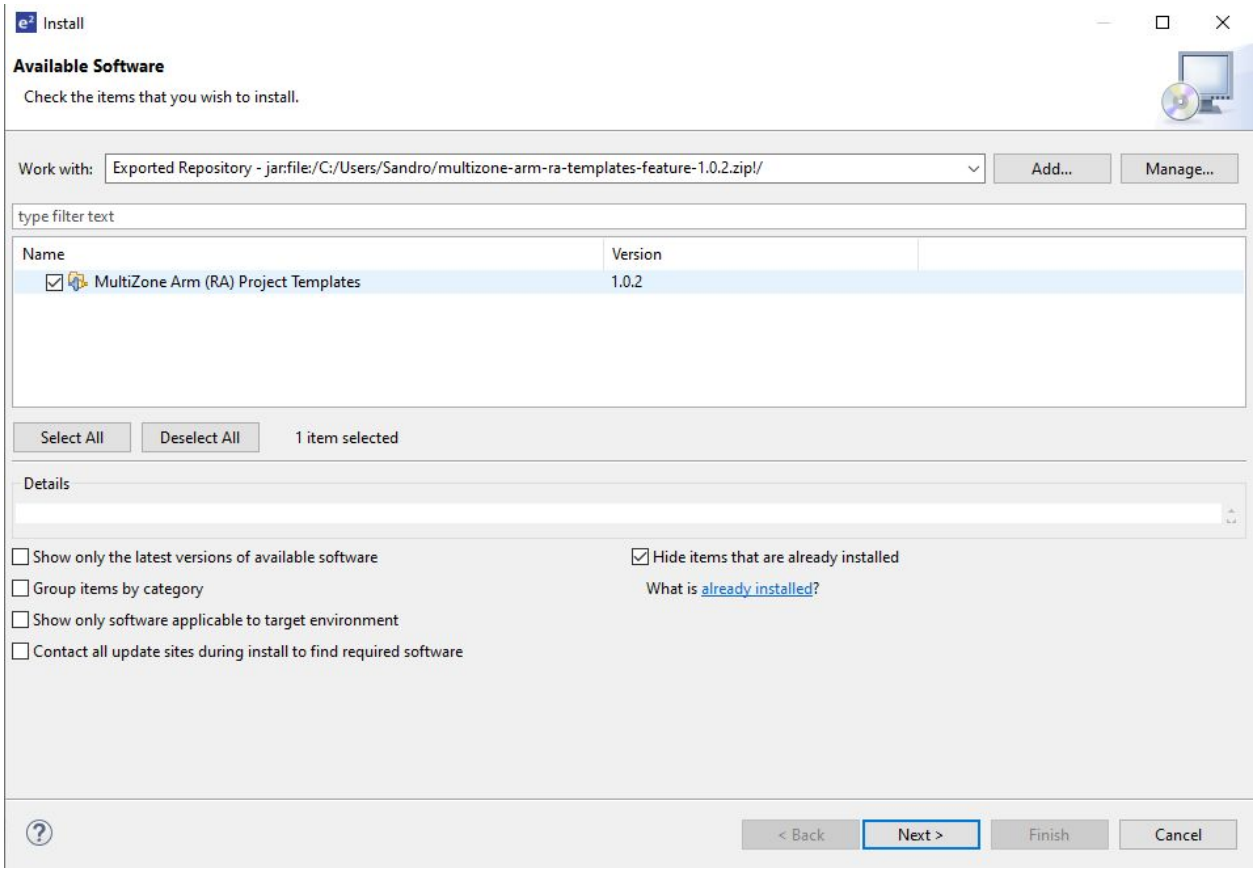
2.1 In the “Help” tab, select “Install New Software”.



2.2 In the Install window, select “Add”, and click the “Archive” option. Select the “multizone-arm-ra-templates-feature-1.0.2” file and click OK.



2.3 After clicking “OK”, please make sure to unset the “Group items by category” option so that the software gets available.

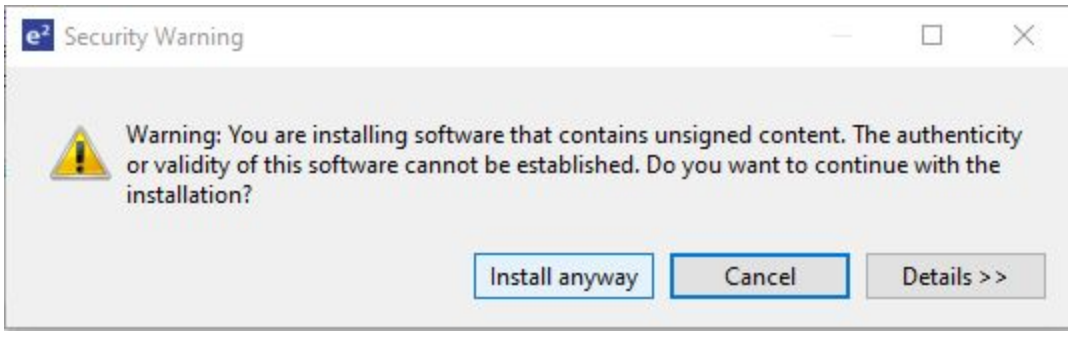


2.4 Select the “MultiZone Arm (RA) Project Templates” feature and click “Next”.

2.5 Review the item to be installed and click “Next”.

2.6 Accept the terms of the license agreement and click “Finish”.

2.7 A security warning will show up - click “Install anyway”.

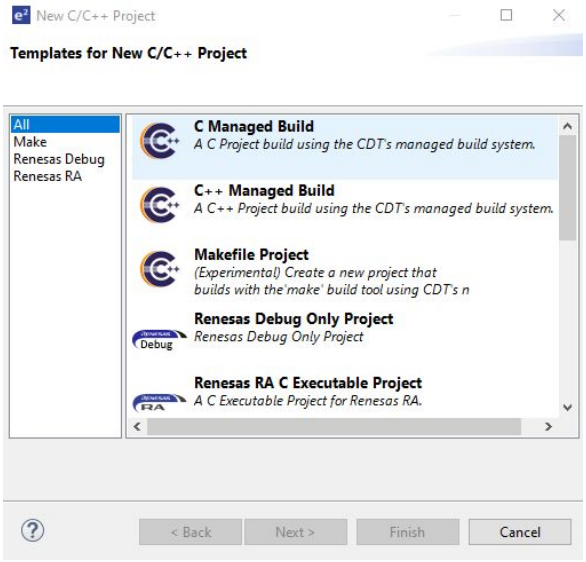


2.8 After the installation make sure to “Restart Now”.

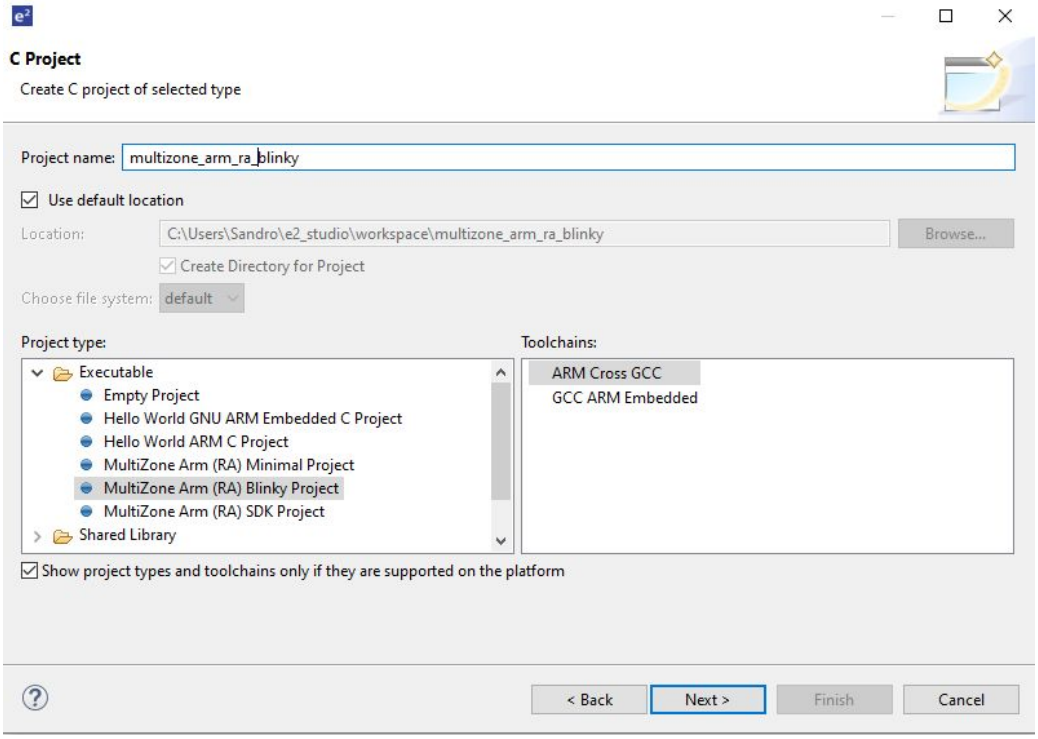
3. Create a MultiZone Arm (RA) Minimal/Blinky/SDK Project

3.1 Select “File”, “New”, “C/C++ Project”.

3.2 Choose the “C Managed Build” option and click “Next”:



3.3 In the “Executable” option, select, for example, “MultiZone Arm (RA) Blinky Project”. Specify the project name and click “Next”:

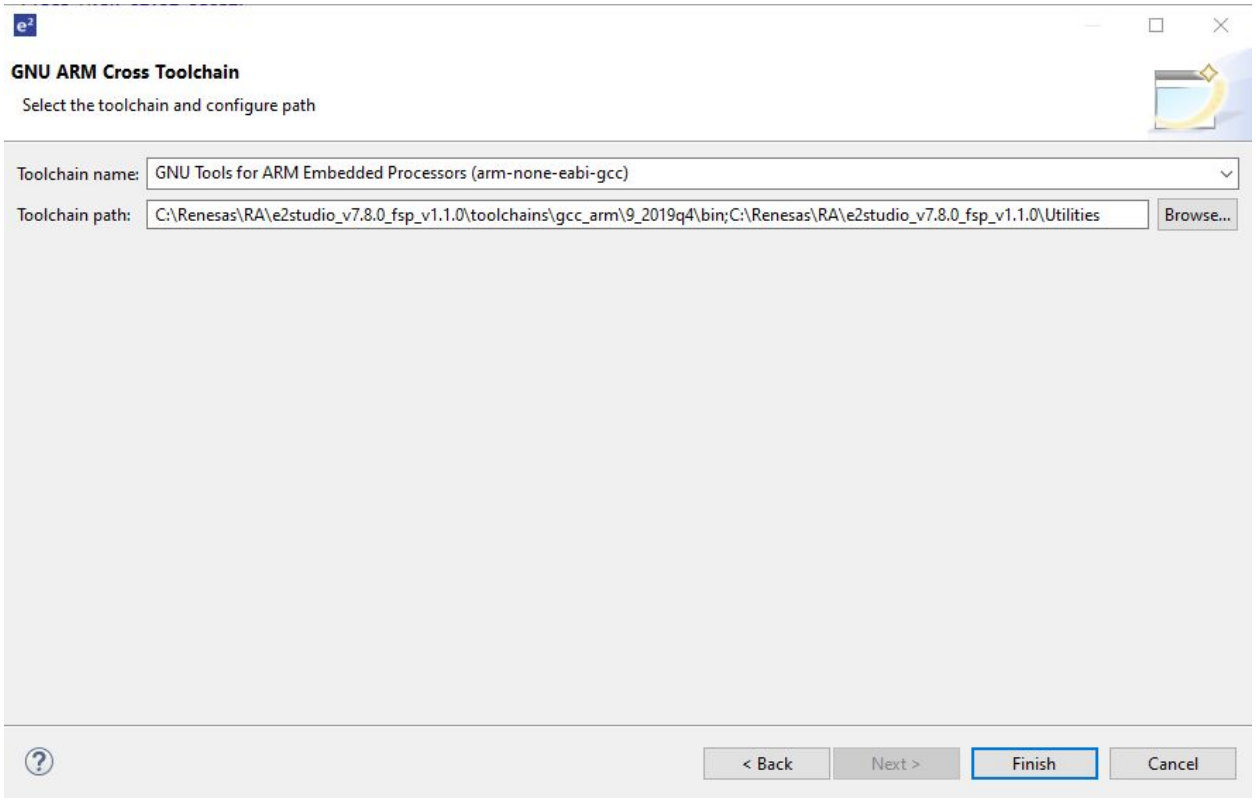


3.4 In the “Select Configurations” just click “Next”.

3.5 In the “GNU ARM Cross Toolchain” make sure to point to the folder of the Toolchain and Utilities where you have installed the Renesas FSP V1.1.0.

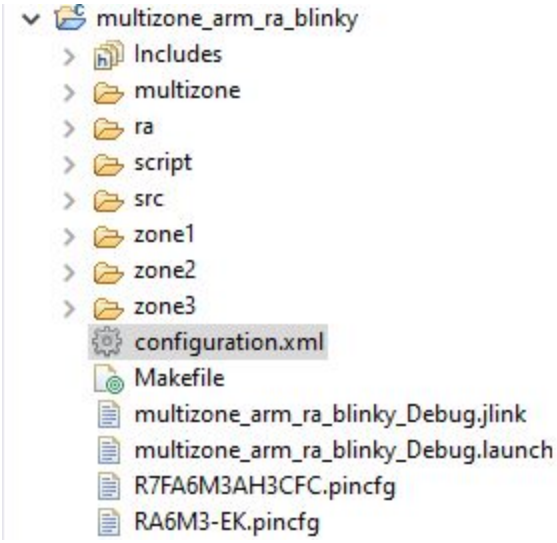
Assuming you have installed the e2studio and the FSP in the C:\

C:\Renesas\RA\e2studio_v7.8.0_fsp_v1.1.0\toolchains\gcc_arm\9_2019q4\bin;C:\Renesas\RA\e2studio_v7.8.0_fsp_v1.1.0\Utilities

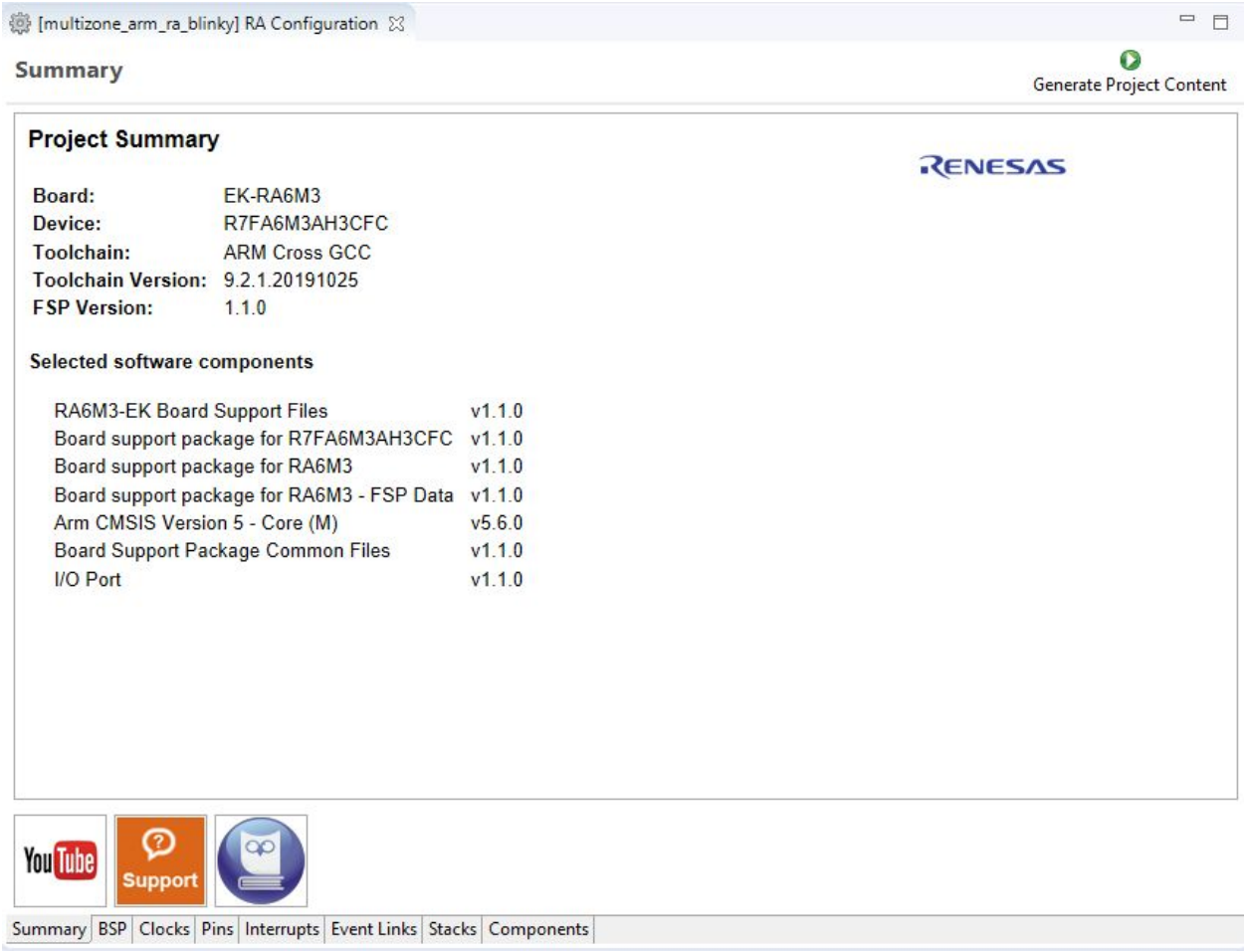


Click “Finish”.

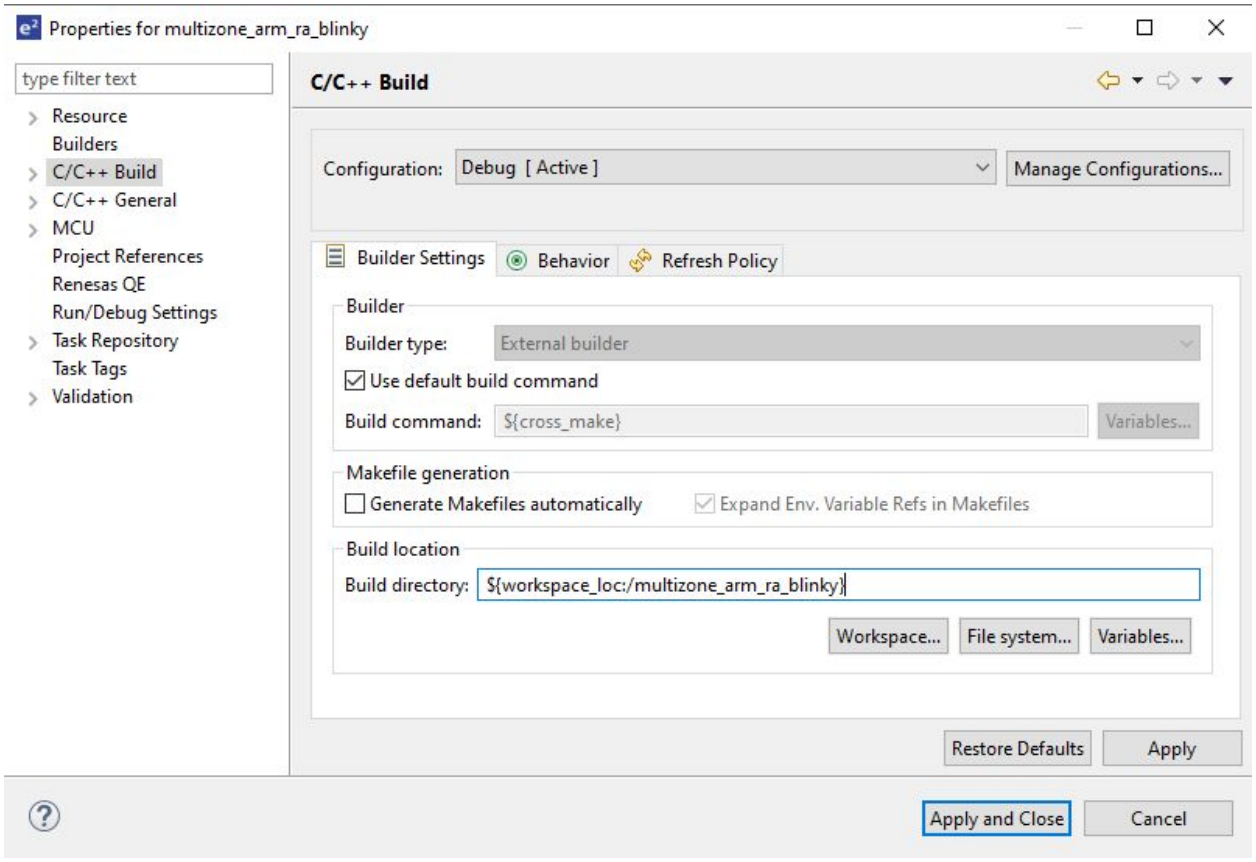
3.6 Open the RA Configuration Tool to generate the Renesas FSP source code for the configuration of the project. Double click the “configuration.xml” file.



3.7 The RA Configuration Tool provides an easy-to-use interface to configure the components and settings of the project. At this stage, just click “Generate Project Content”.



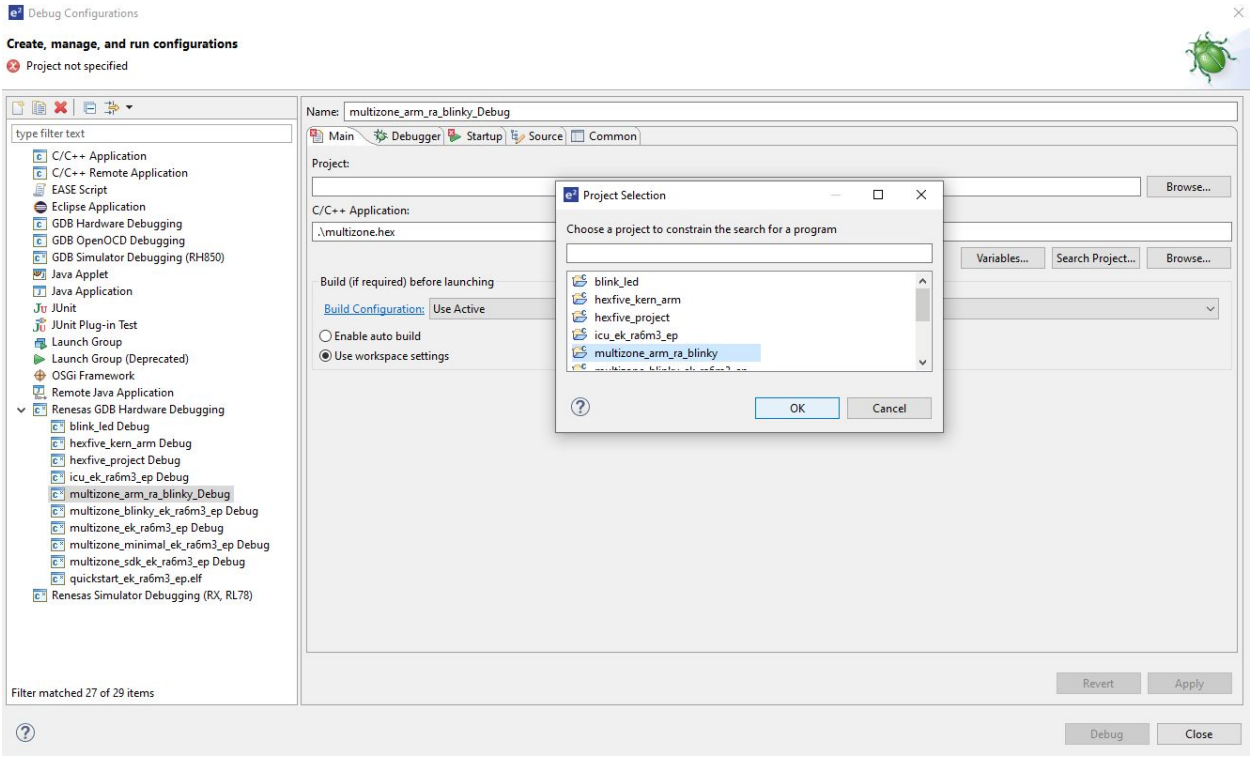
3.8 Before compiling, make sure to modify the C/C++ Build settings. Right click over the project and go to “Properties” and select the ”C/C++ Build” tab. Uncheck the “Generate Makefile automatically” and point the “Build directory” to the top directory of the project (remove the “/Debug”). Select “Apply and Close”.



3.9 Compile the project. Right click over the project and select “Build Project”. At the end of the compilation stage, a new folder called “Debug” will host all object files for the different zones (and OEM boot) of the system.

3.10 To Debug your project on the target EKRA6M3 platform, right click over the project and select “Debug As”->”Debug configurations...”. Select the correct target debug configuration (in this case “multizone_arm_ra_blinky_Debug”). In the “Main” tab, specify

the correct project (“multizone_arm_ra_blinky”) and select OK. Select “Apply” and “Debug”.



3.11 The debug session will start at address 0x500 (entry point of the MultiZone kernel). Happy testing!