Getting Started with AURIX™ Development Studio

Installation and first steps

AURIX™ Development Studio Training
V1.0.15

Scope of work

This tutorial provides a guide for the user to:

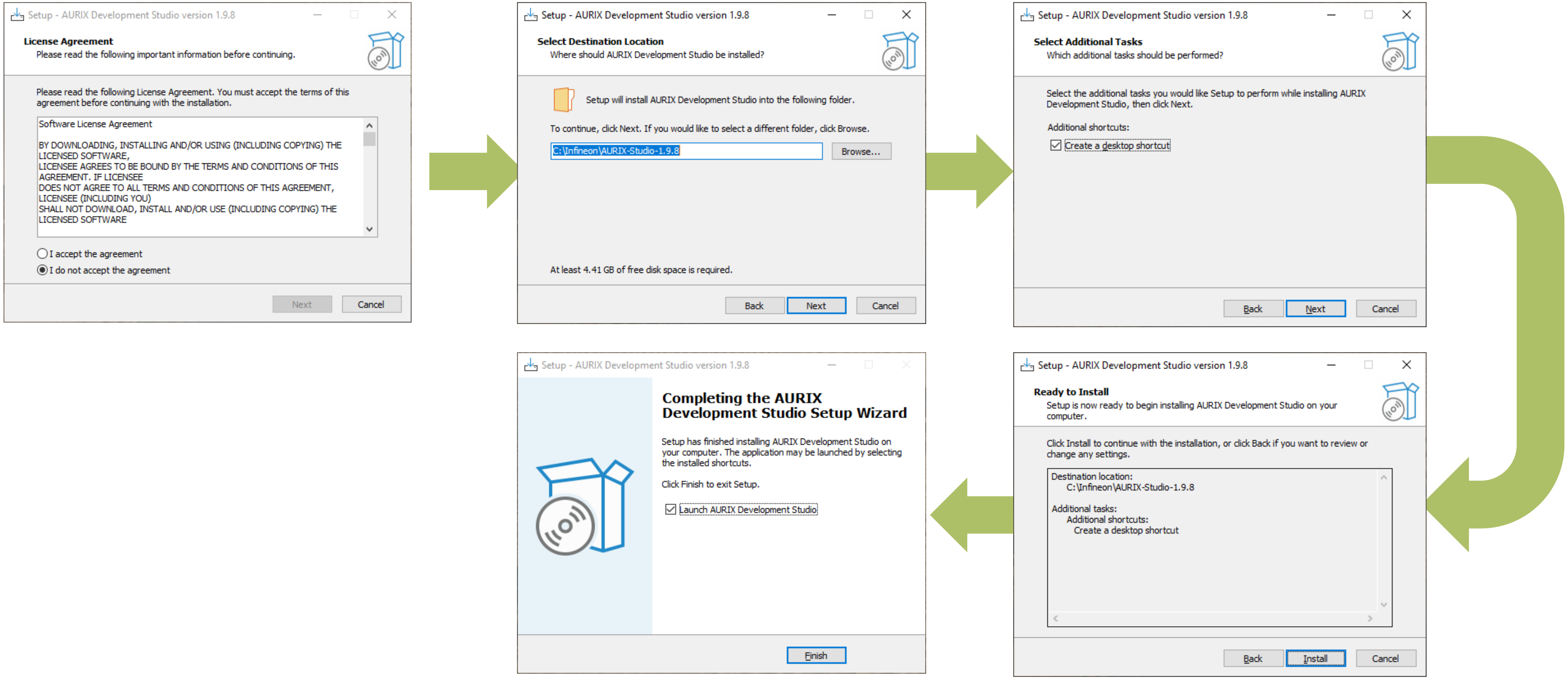
- Install AURIX™ Development Studio V1.9.8
- Create new project
- Import project (Infineon Code Examples Repository)
- Build project
- Debug project
- Additional material

Download

- The installation package of AURIX™ Development Studio can be found here: https://www.infineon.com/aurixdevelopmentstudio

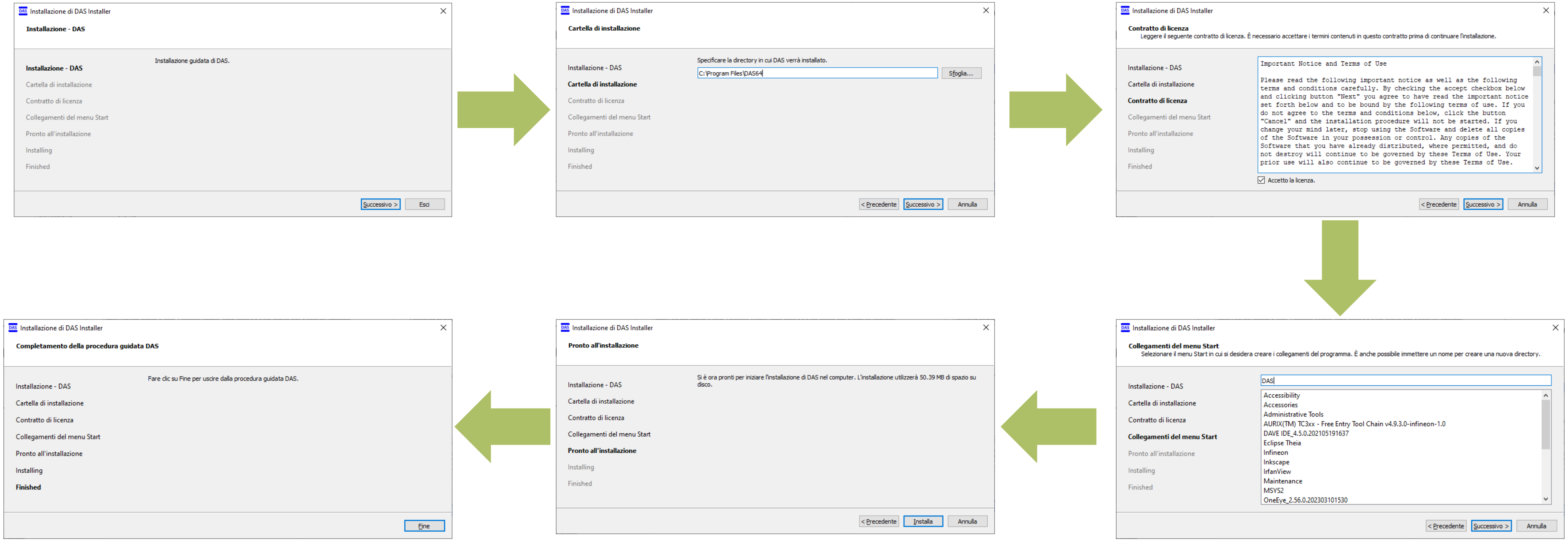
Install AURIX™ Development Studio - 1

› To install AURIX™ Development Studio, launch the installation package and follow the steps:



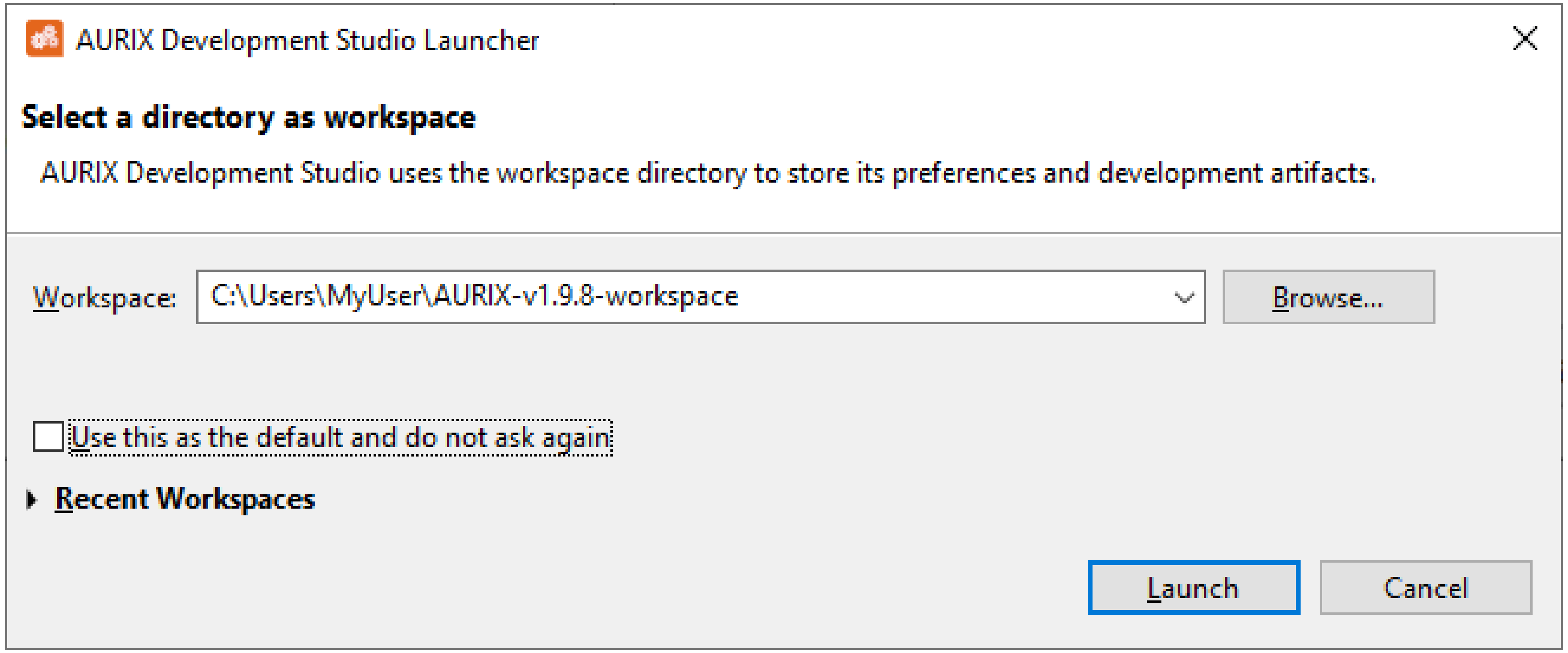
Install AURIX™ Development Studio - 2

› If DAS64 is not installed or outdated, it will be installed automatically during the AURIX™ Development Studio installation:



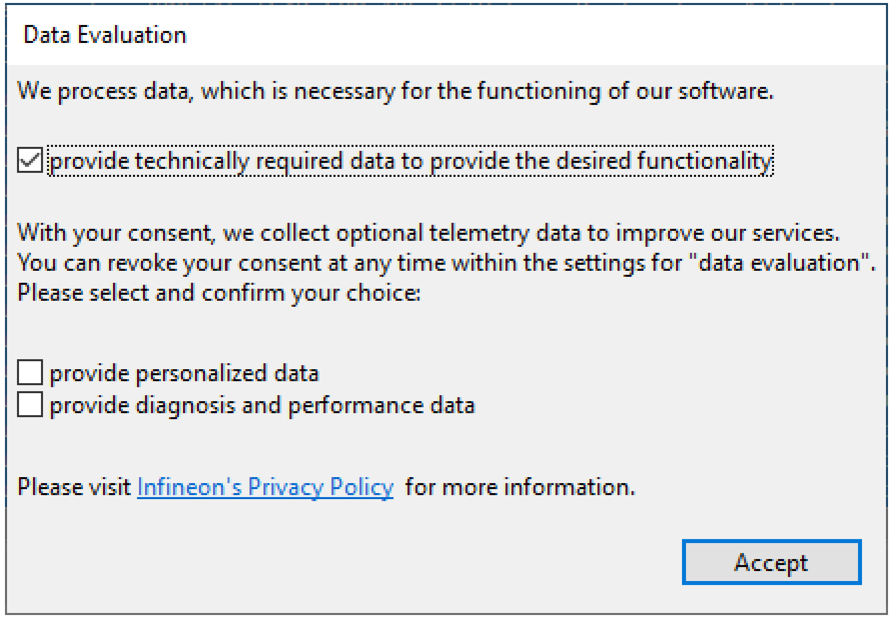
Workspace definition

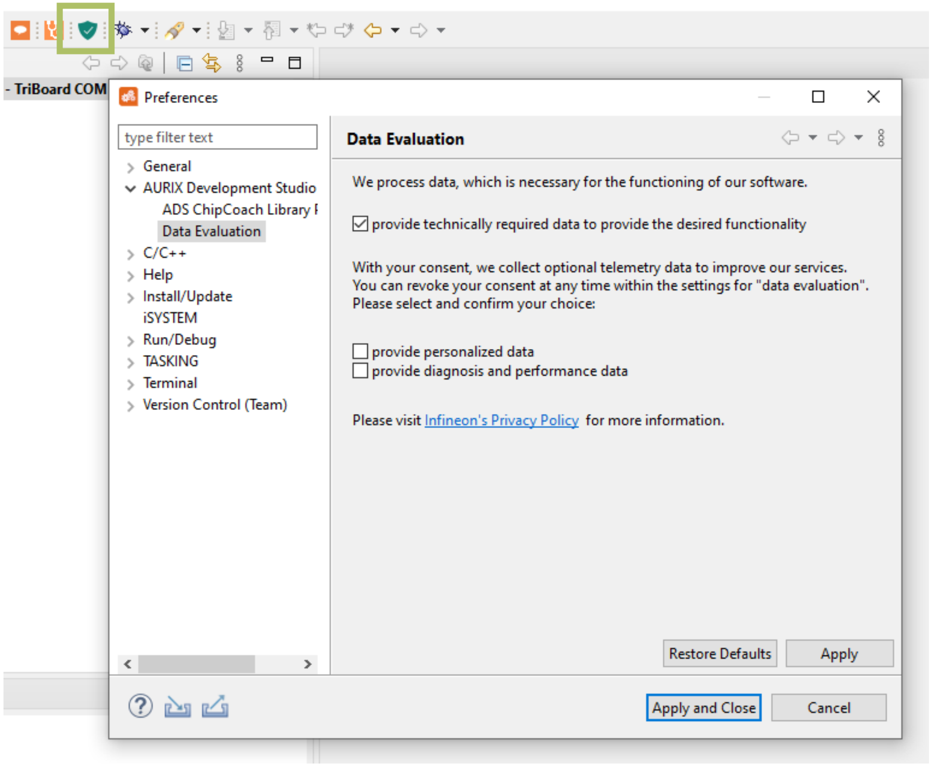
› After launching the AURIX™ Development Studio, it is necessary to select a workspace



Data evaluation consent

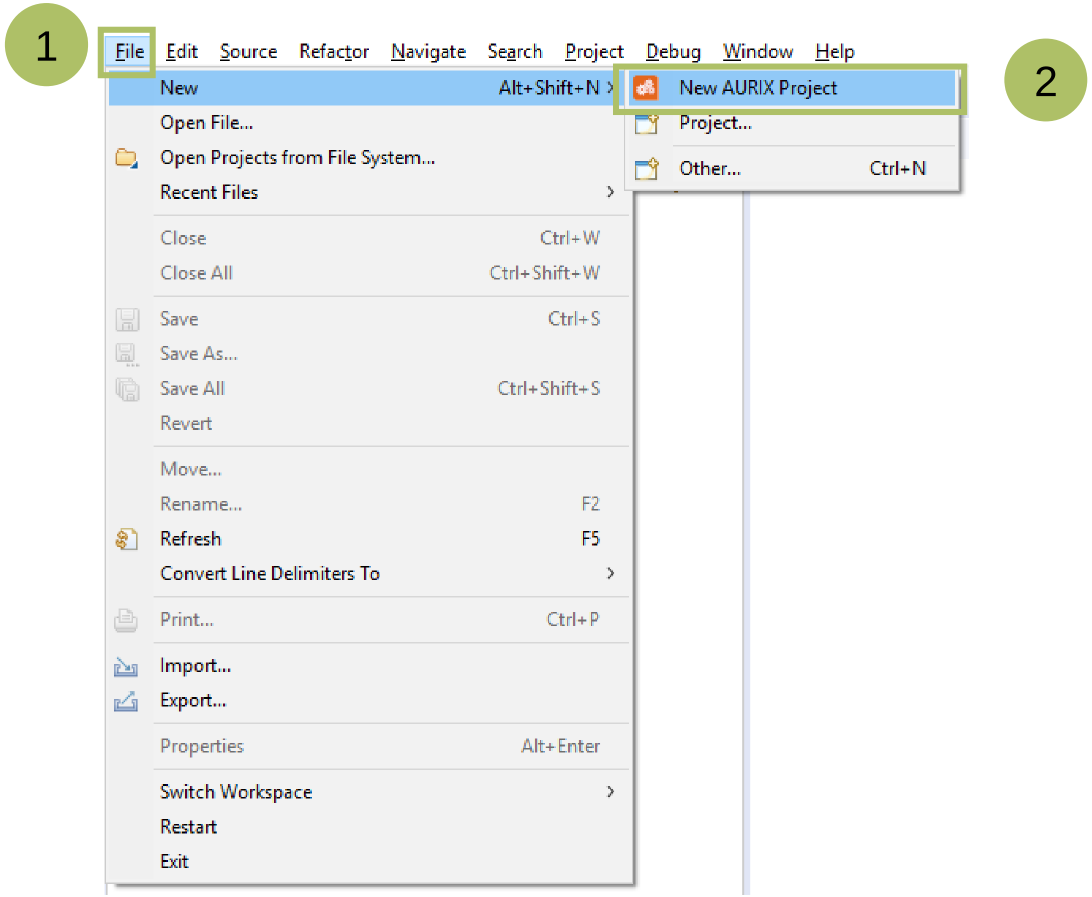
- On the first start you are asked for the consent to send data to Infineon for evaluation purposes.
- You can choose which data to send.
- You can later decide to modify your consent by clicking on “Open Data Evaluation Settings” on the toolbar





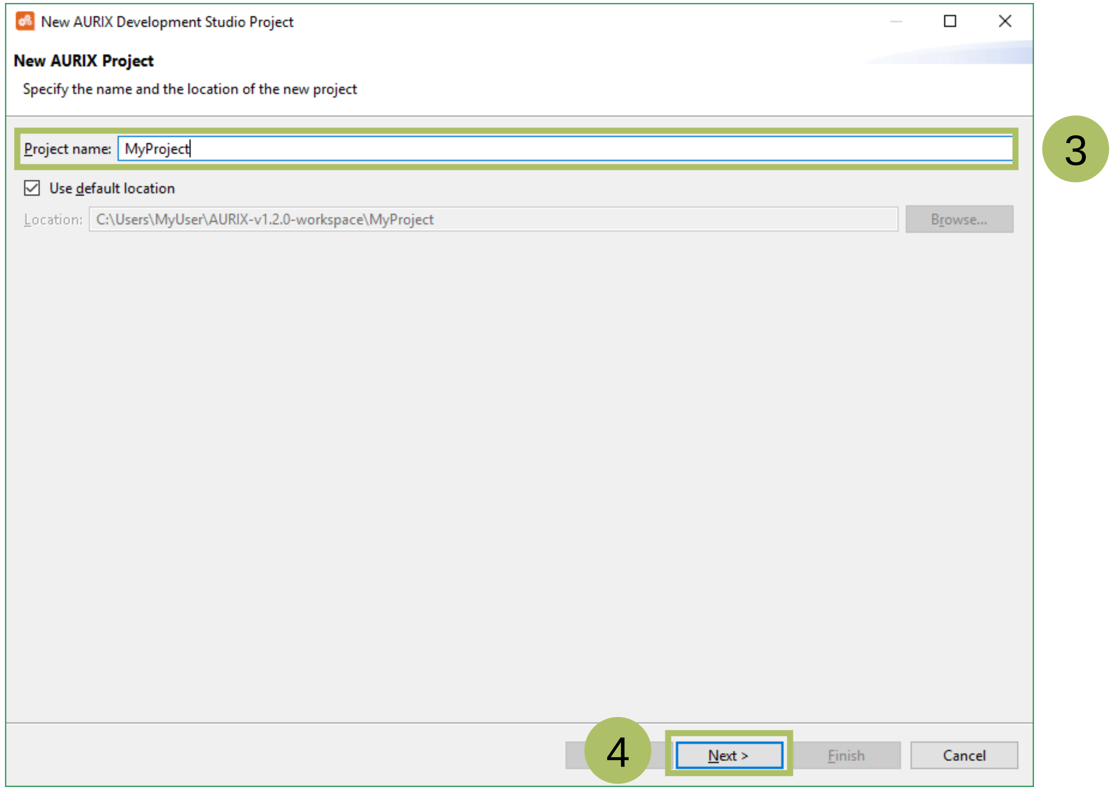
Create new project - 1

- Once the program is started, a new project can be created by selecting File >> New >> “New AURIX Project”



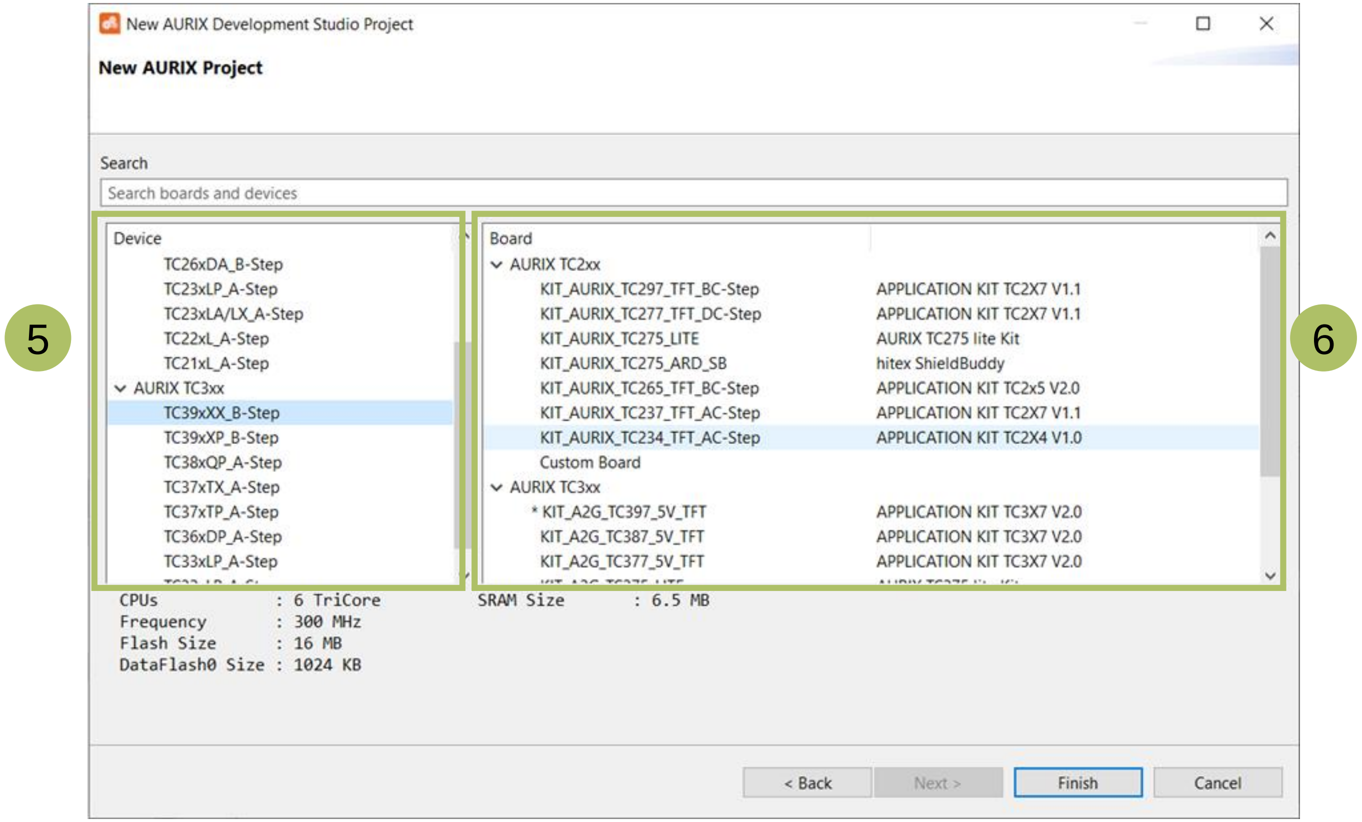
Create new project - 2

- From the “New AURIX Development Studio Project” window, choose a name for the new project (3)
- The “Use default location” checkbox should be set in order to create the project inside the current selected workspace



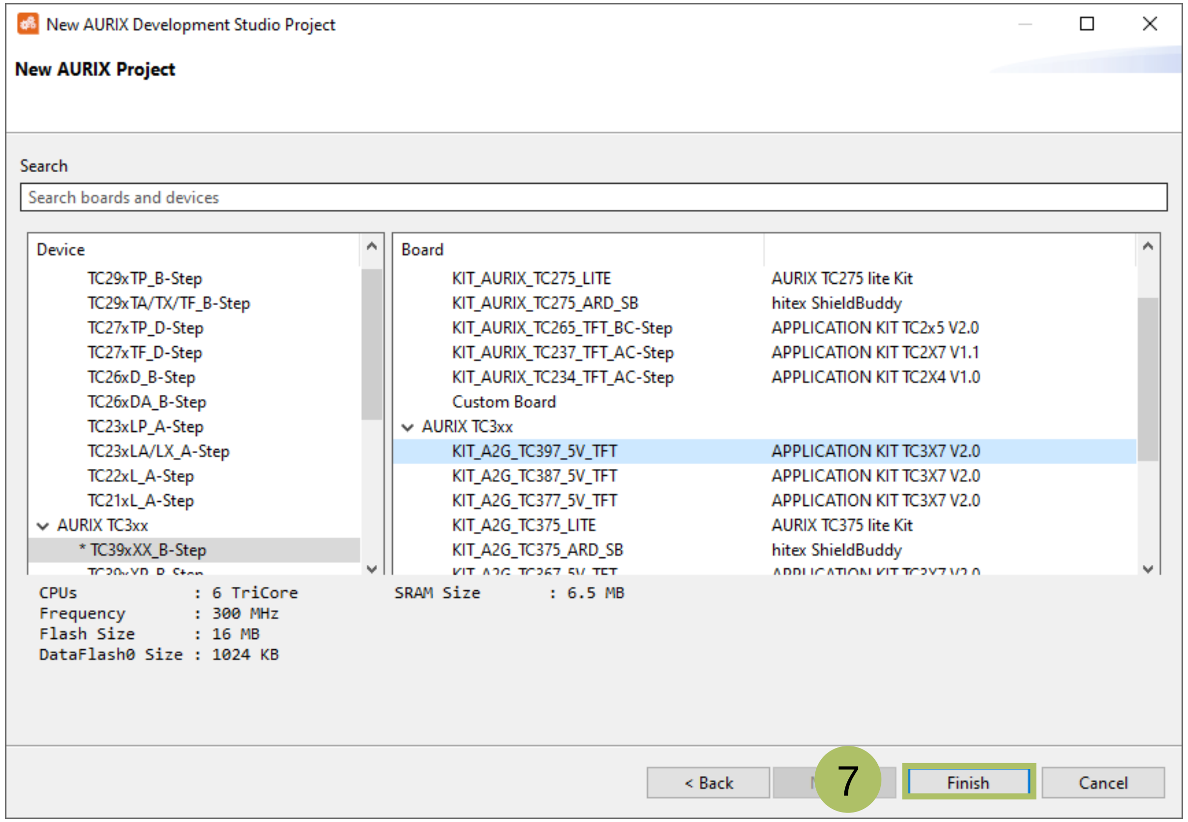
Create new project - 3

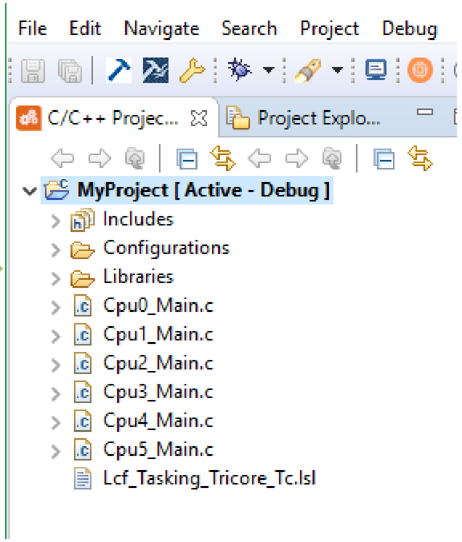
› From the “New AURIX Development Studio Project” window, choose the device or the board. A specific device (5) or board (6) can be chosen from the left or right list. Furthermore, while selecting a board, the tool highlights the supported devices for that board and vice versa



Create new project - 4

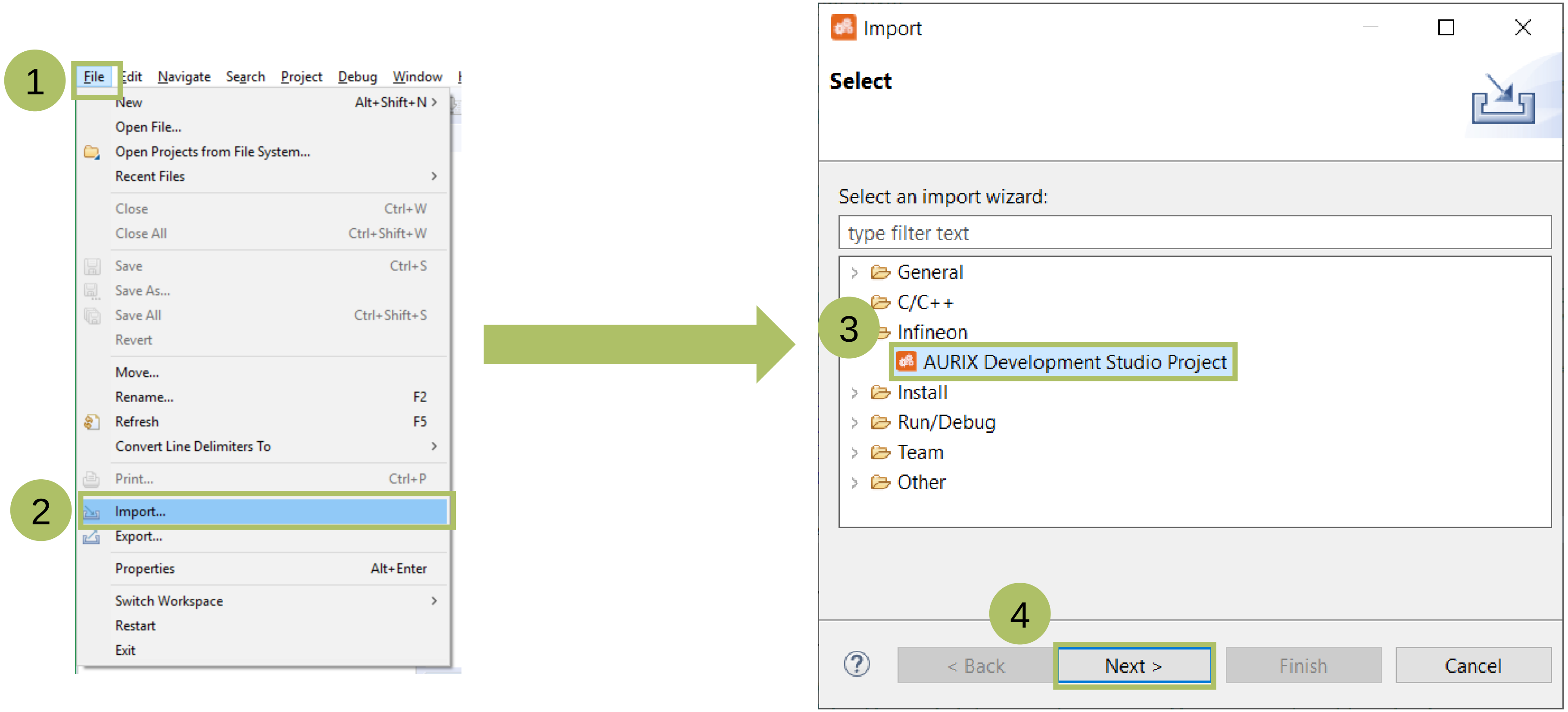
› By pressing “Finish” (7), a new project is created





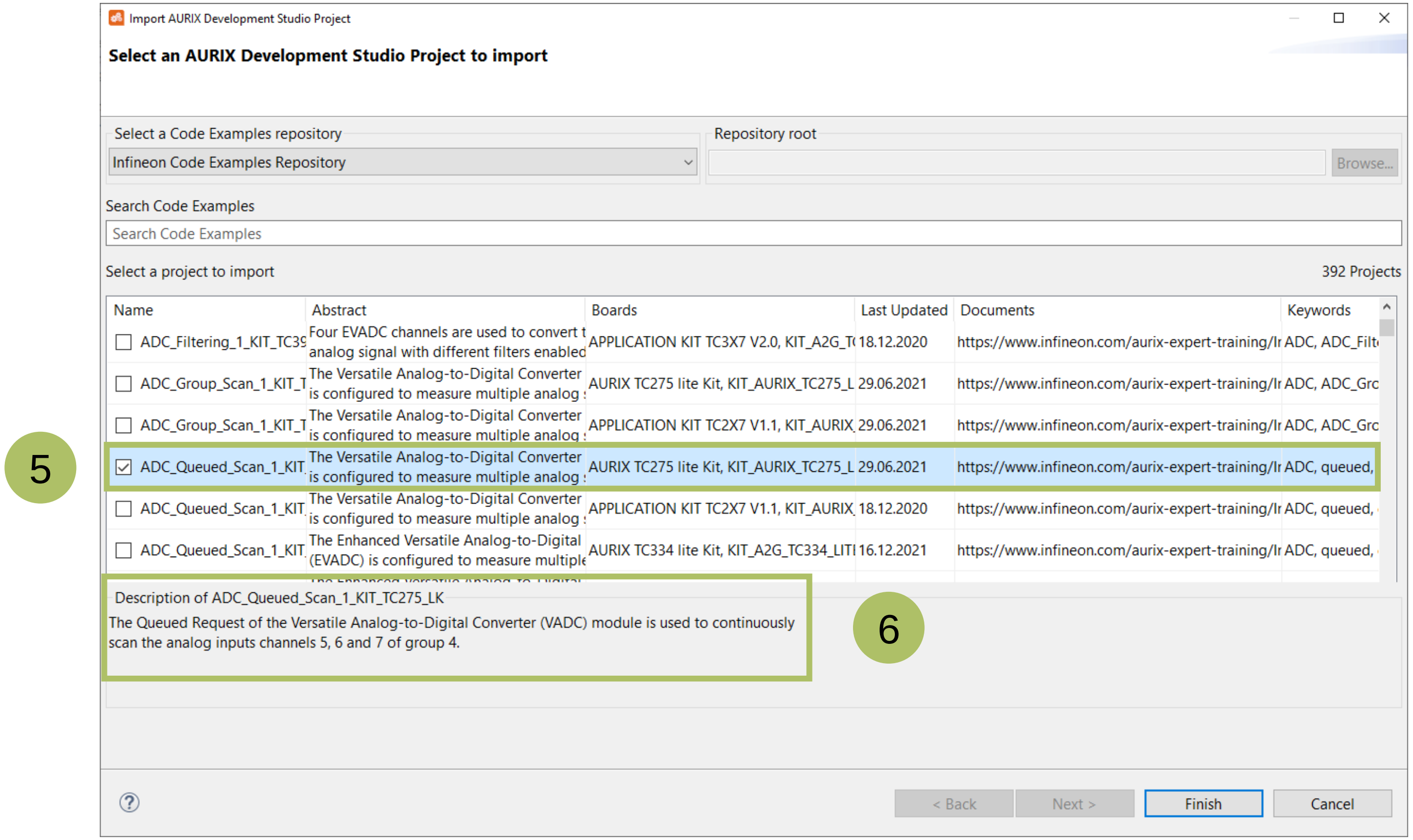
Import project (Infineon Code Examples Repository) - 1

- Alternatively, it is possible to import an example project using File >> “Import…” utility (1-2) and selecting Infineon >> “AURIX Development Studio Project” type (3)
- At the end, press “Next” (4)



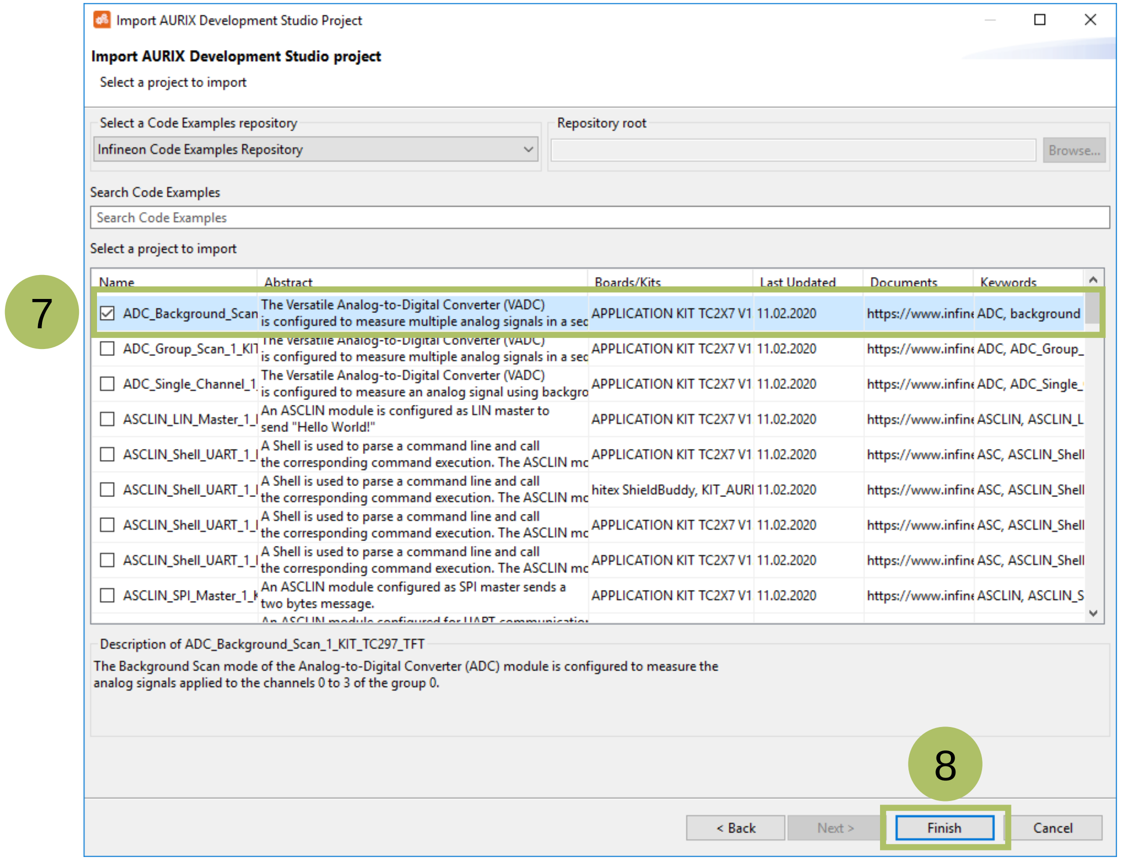
Import project (Infineon Code Examples Repository) - 2

› Hint: Clicking on an example project (5) in the list shows the example description (6)



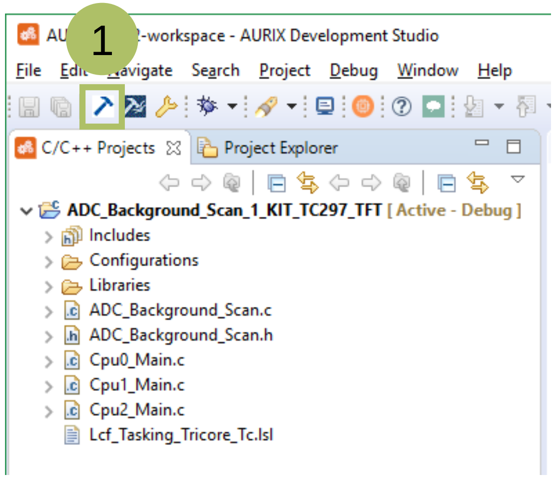
Import project (Infineon Code Examples Repository) - 3

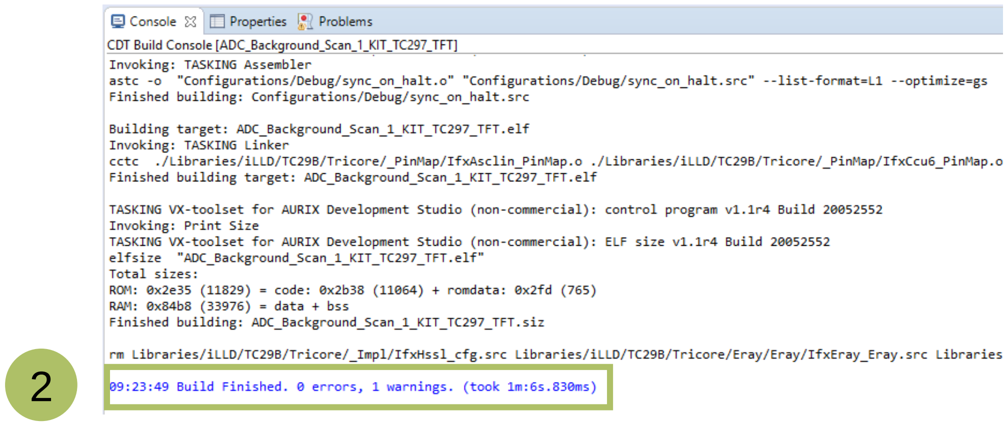
› Select (double-click) an example project (7) from the list and press “Finish” (8). This creates a local copy of the example in your workspace directory and opens the project



Build project

› Before debugging, it is necessary to build the project. Press the “Build Active Project” icon (1) and when the build is finished, check that there are no compiling errors (2)

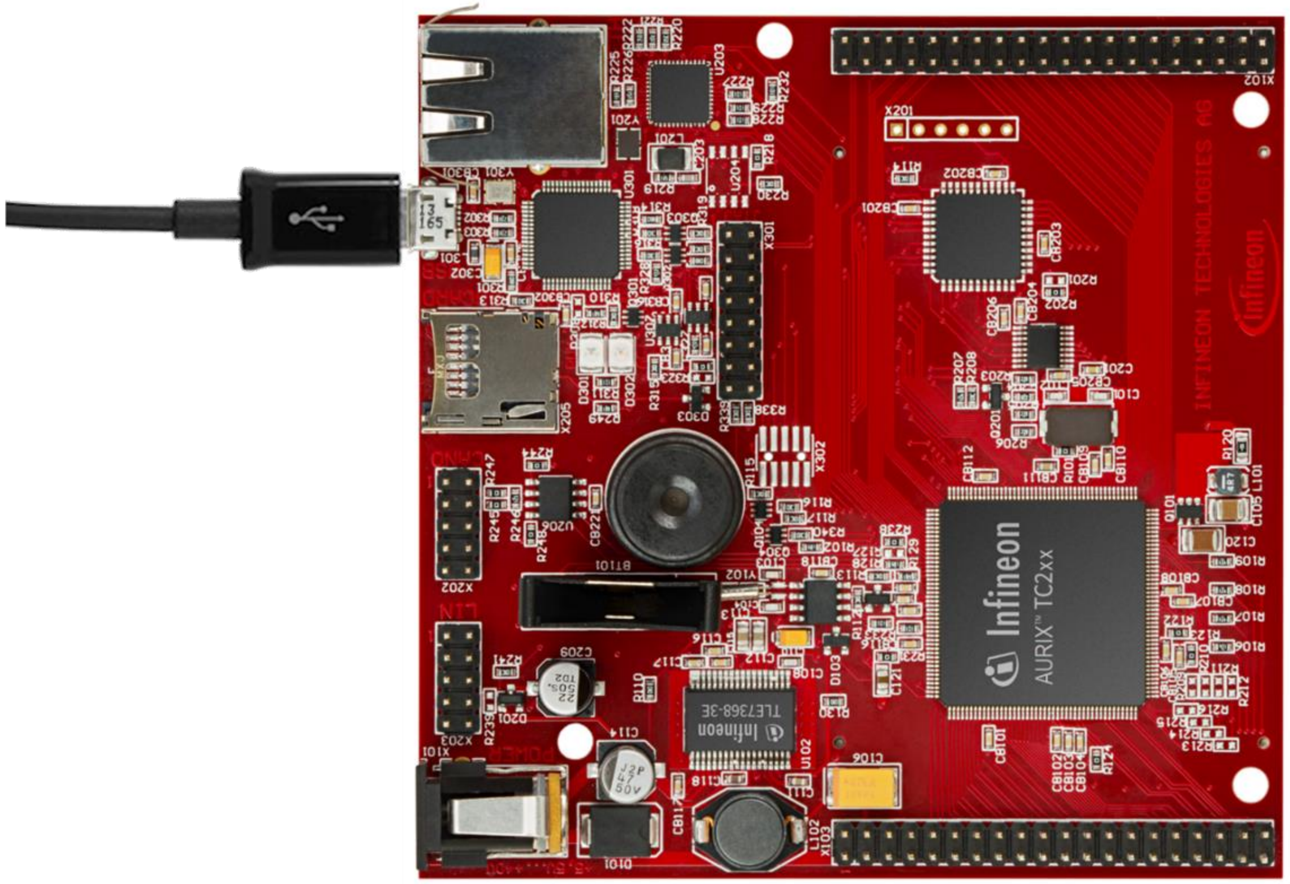




Check

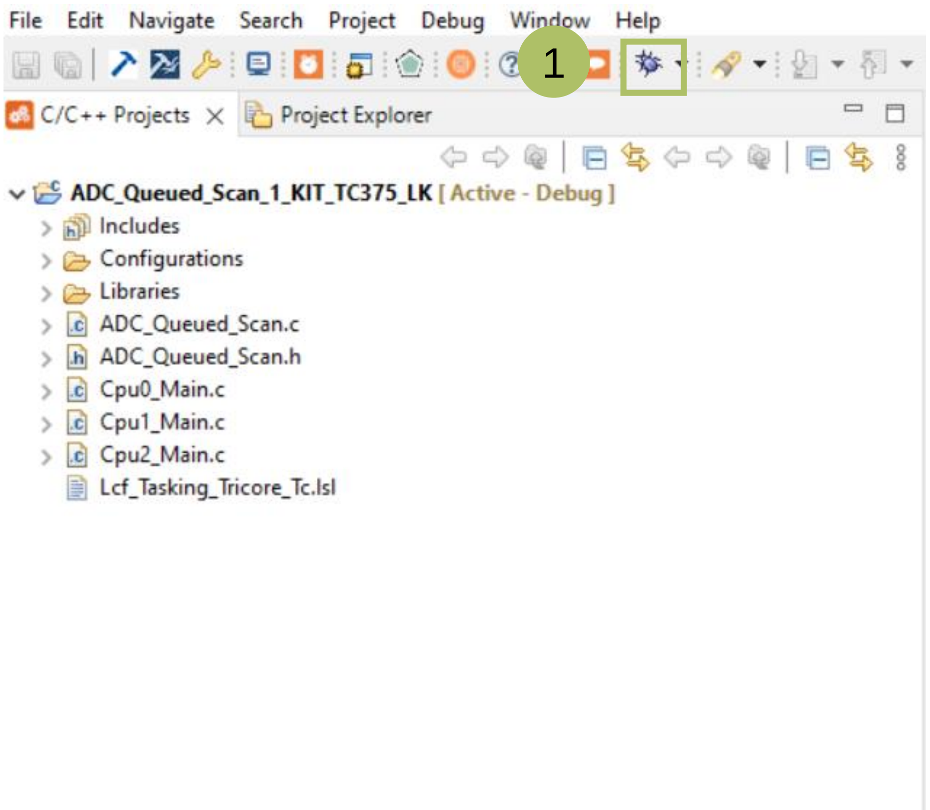
Debug project - 1

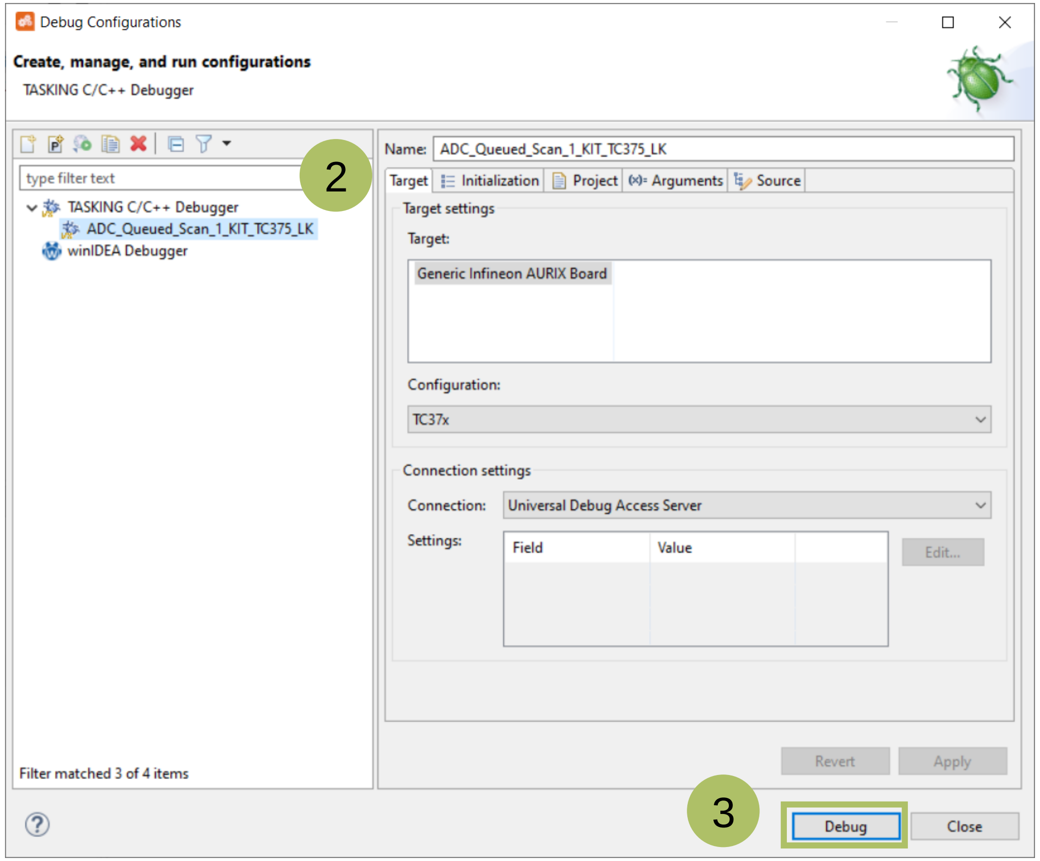
› Connect your device via an USB cable to the PC



Debug project - 2

› In order to flash and debug the code, press the “Debug Active Project” icon (1), chose a debugger (2) and then press the “Debug” button on the “Debug Configurations” window (3)

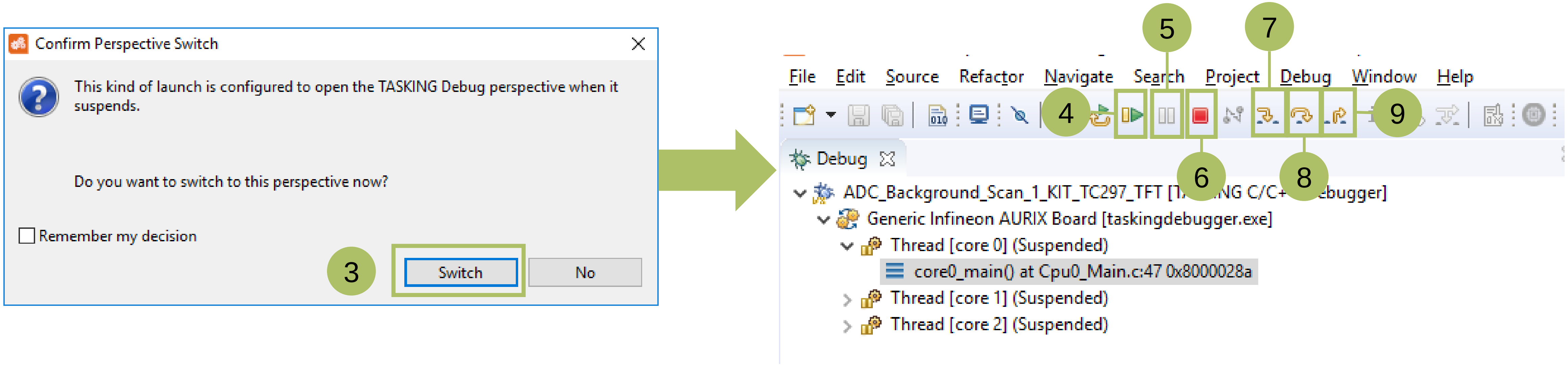




Debug project - 3

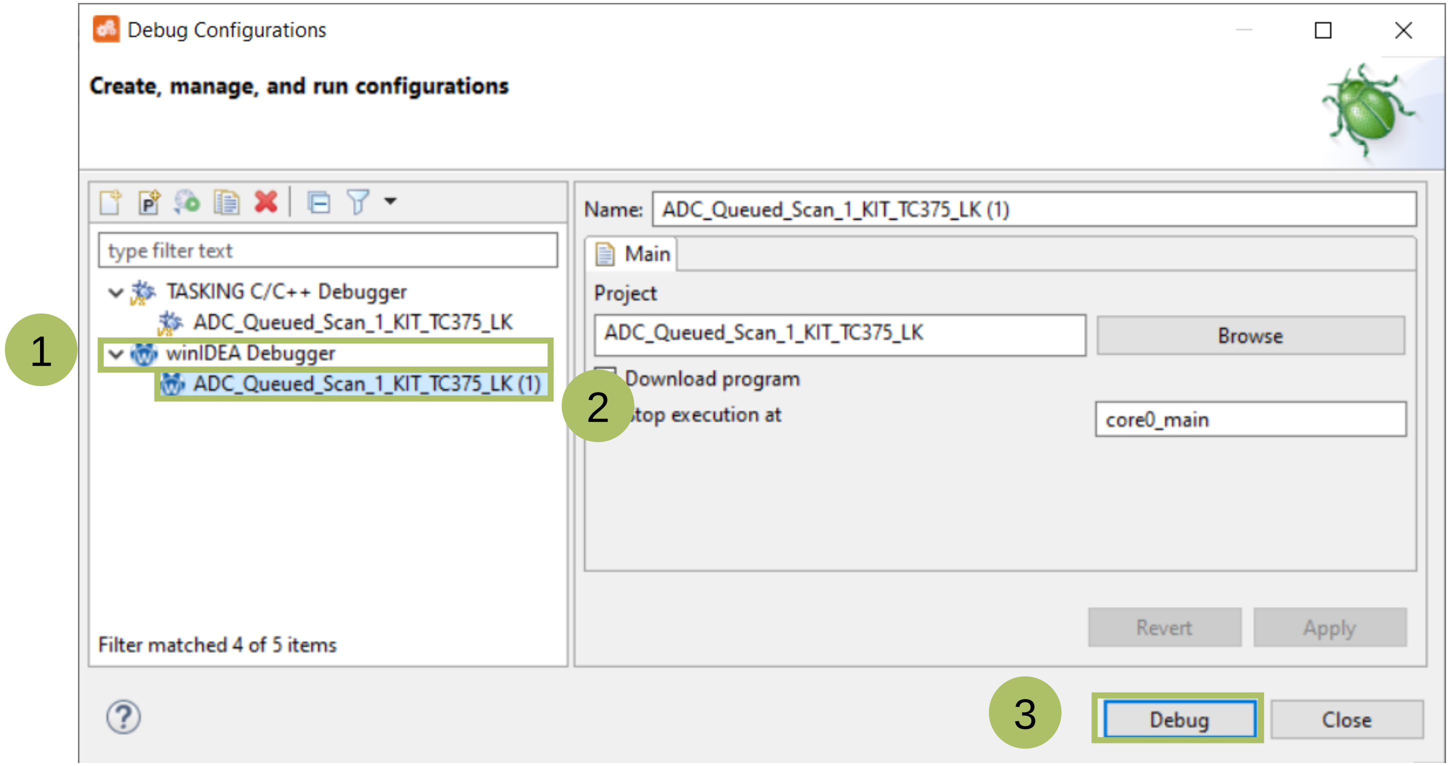
When using TASKING C/C++ Debugger:

- Switch the perspective when asked (3) and press “Resume” (4) to run the code
- While running, the code can be stopped with the “Suspend” button (5)
- To terminate the debug session, press the “Terminate” button (6)
- Additionally, in the Debug perspective, it is also possible to run the code in single or multiple steps with the buttons “Step Into” (7), “Step Over” (8) and “Step Return” (9)



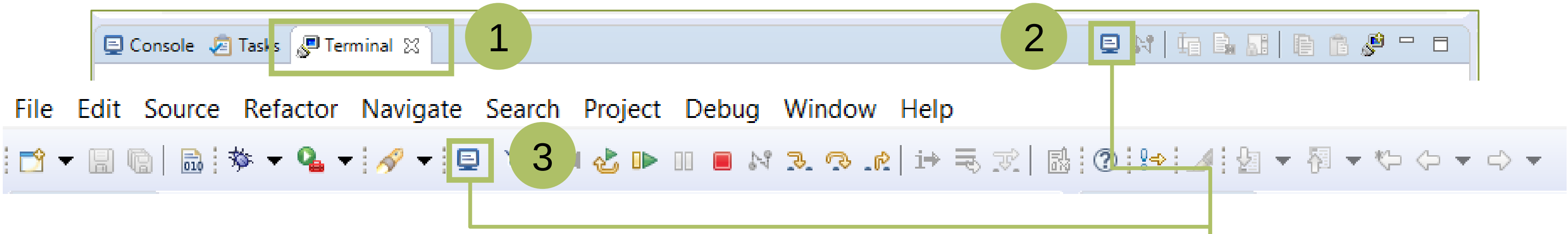
Debug project - 4

› To select winIDEA as debugger:
  – Double click on “winIDEA Debugger” to create a configuration (1)
  – Select the configuration (2)
  – Press the Debug button (3)

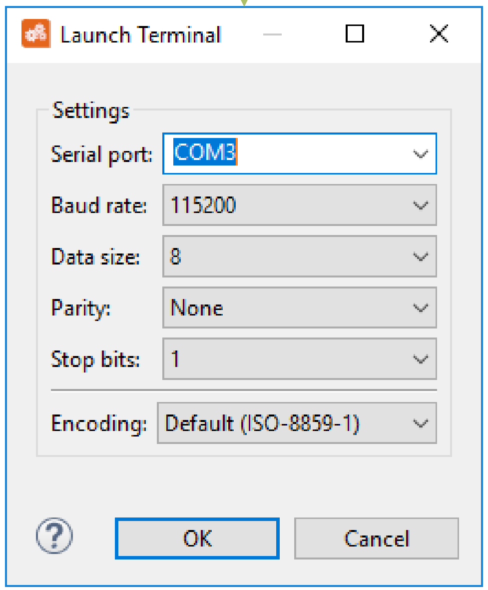


Serial Monitor View

› A serial monitor is open by default (1) in the Debug Perspective inside the AURIX™ Development Studio, or it can be open manually from the terminal icon (3)

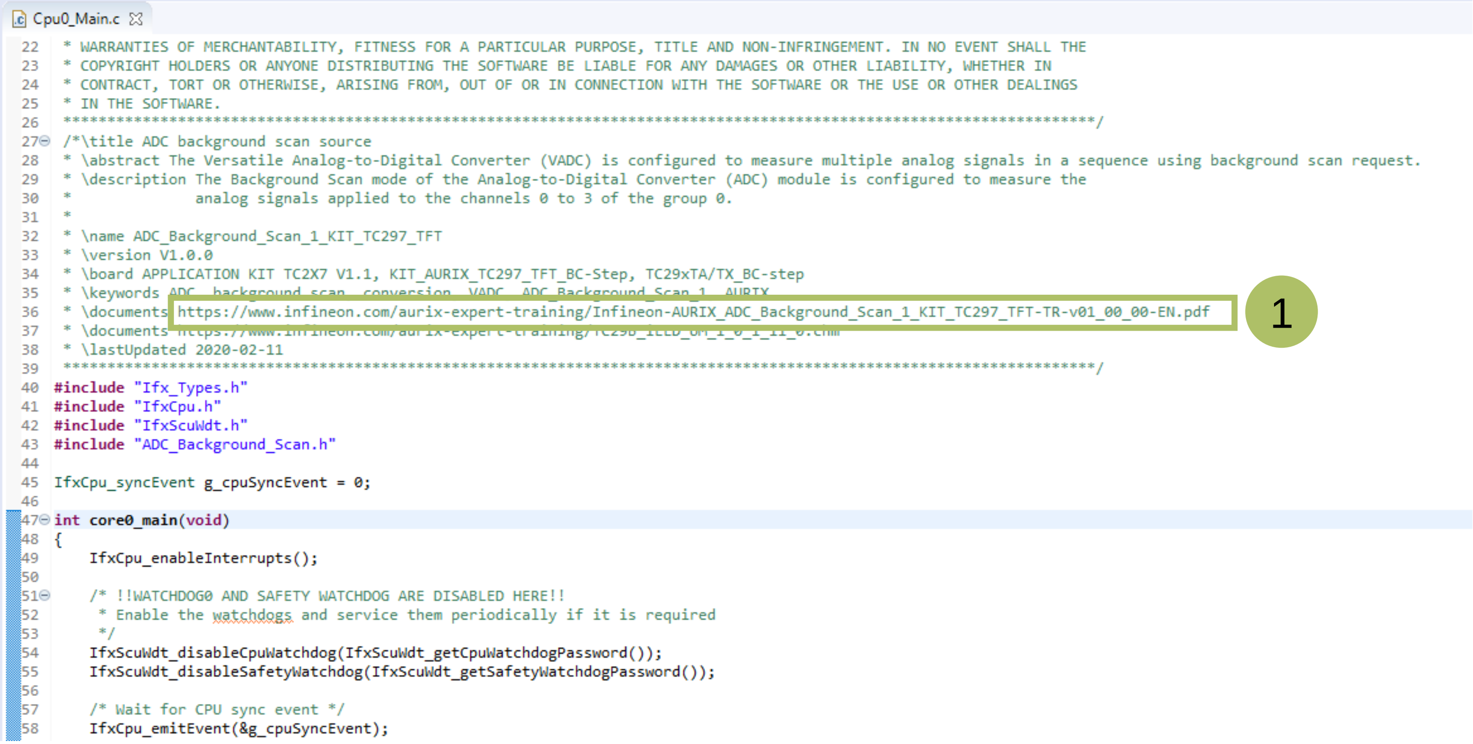


› The serial monitor must be configured (2-3) with the following parameters to enable the communication between the board and the PC:
  - Serial port number
  - Speed (baud rate)
  - Data size
  - Parity
  - Stop bits



Additional material - 1

› All the imported examples from Infineon come with a tutorial explaining the needed HW/SW setup, the code and how to run and test the example
› The tutorial is accessible from the AURIX™ Development Studio by Ctrl + click on the link (1) in the Cpu0_Main.c file



Additional material - 2

› From the same Cpu0_Main.c file, it is possible to download the Infineon Low Level Drivers documentation (2) for the specific device used in the example

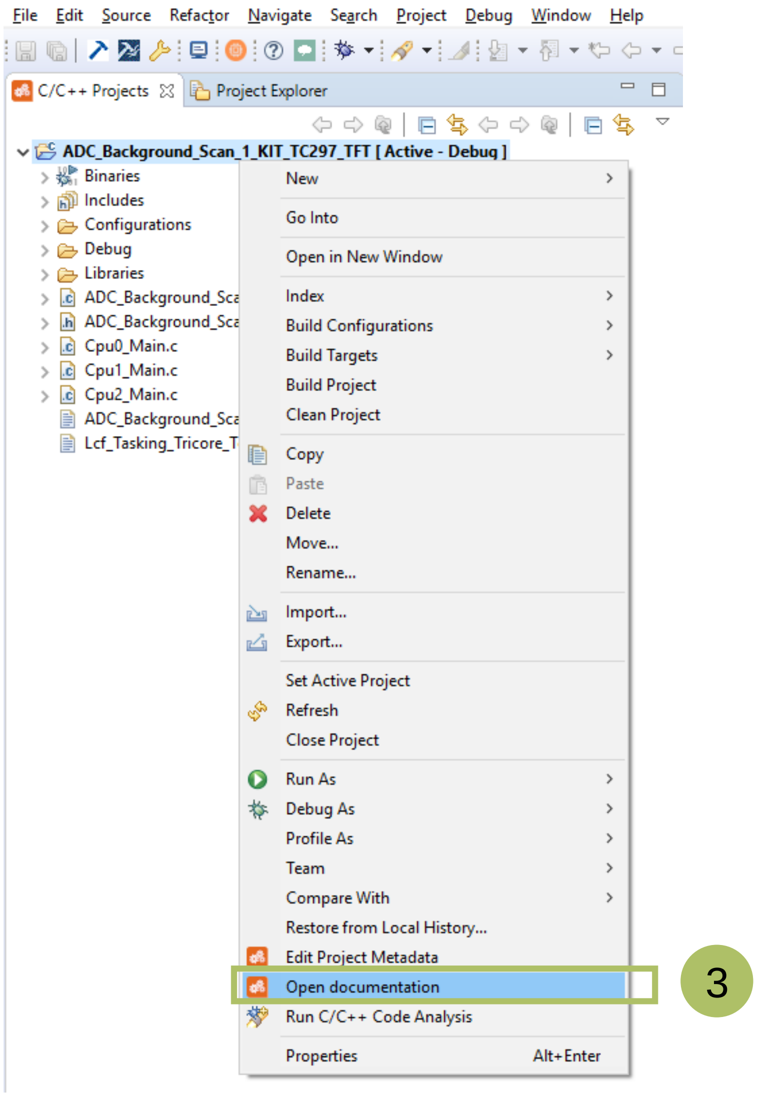
Cpu0_Main.c

```c
 * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE
 * COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN
 * CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
 * IN THE SOFTWARE.
 *********************************************************************************************************************/
/*\title ADC background scan source
 * \abstract The Versatile Analog-to-Digital Converter (VADC) is configured to measure multiple analog signals in a sequence using background scan request.
 * \description The Background Scan mode of the Analog-to-Digital Converter (ADC) module is configured to measure the
 *              analog signals applied to the channels 0 to 3 of the group 0.
 *
 * \name ADC_Background_Scan_1_KIT_TC297_TFT
 * \version V1.0.0
 * \board APPLICATION KIT TC2X7 V1.1, KIT_AURIX_TC297_TFT_BC-Step, TC29xTA/TX_BC-step
 * \keywords ADC, background scan, conversion, VADC, ADC_Background_Scan_1, AURIX
 * \documents https://www.infineon.com/aurix-expert-training/Infineon-AURIX_ADC_Background_Sc   T_TC297_TFT-TR-v01_00_00-EN.pdf
 * \documents https://www.infineon.com/aurix-expert-training/TC29B_iLLD_UM_1_0_1_11_0.chm
 * \lastUpdat
 *********************************************************************************************************************/
#include "Ifx_Types.h"
#include "IfxCpu.h"
#include "IfxScuWdt.h"
#include "ADC_Background_Scan.h"

IfxCpu_syncEvent g_cpuSyncEvent = 0;

int core0_main(void)
{
    IfxCpu_enableInterrupts();

    /* !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdogs and service them periodically if it is required
     */
    IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
    IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());

    /* Wait for CPU sync event */
    IfxCpu_emitEvent(&g_cpuSyncEvent);
```

2

Additional material - 3

› Hint: both the example's tutorial and the iLLD documentation can be opened by Right clicking on the project name and pressing the "Open documentation" utility (3)



Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-07
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
Getting Started with
AURIX™ Development Studio
V1.0.15

Part of your life. Part of tomorrow.