AURIX Development Studio

+ HighTec GCC toolchain

Version 1.0, February, 2022

Table of Contents

Terms & Abbreviations

GCC

Gnu Compiler Collection; free open source compiler produced by GNU Project

AURIX

Automotive Realtime Integrated neXt generation architercture; 32-bit Infineon's microcontroller family targeting automotive industry

IDE

Integrated Development Environment; software for developing applications that combines developer tools into single graphical user interface (GUI)

1. Document purpose

This document describes how to download and install Infineon's AURIX Development Studio and configure it for use with the HighTec GCC compiler.

2. AURIX Development Studio

2.1. Download

The AURIX Development Studio can be downloaded from Infineon's pages using the link below.

https://www.infineon.com/cms/en/product/promopages/aurix-development-studio

It is available for the Windows platform as an executable installation file.

Fig. 1. Download of AURIX Development Studio

2.2. Installation and run

The process of installation is straightforward. When it finishes, the main screen appears.

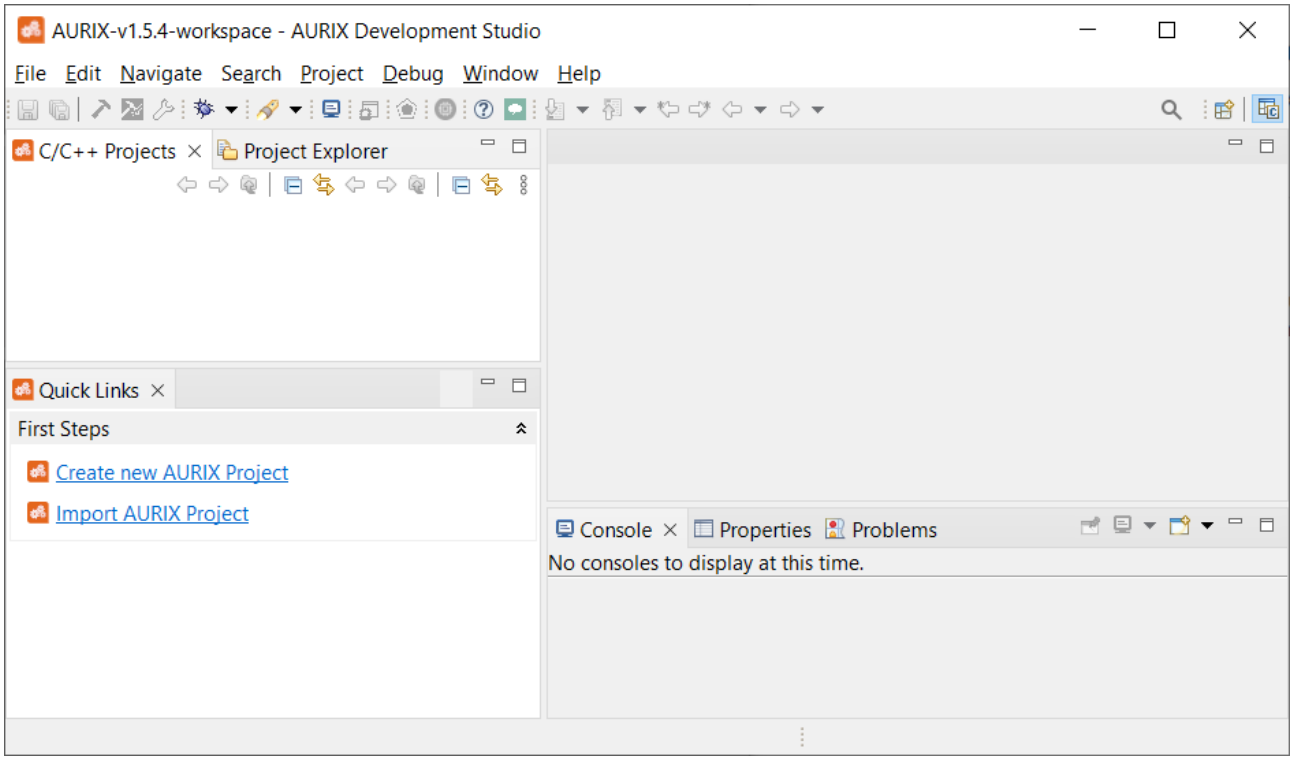

Fig. 2. Initial screen of the AURIX Development Studio

3. External GCC toolchain

3.1. Documentation

The configuration of an external GCC toolchain is described in the documentation of the AURIX Development Studio.

Choose **Help→Help Contents**, and navigate to **AURIX Development Studio User Guide→Tasks→External Toolchains→Configuring and external GCC Toolchain**.

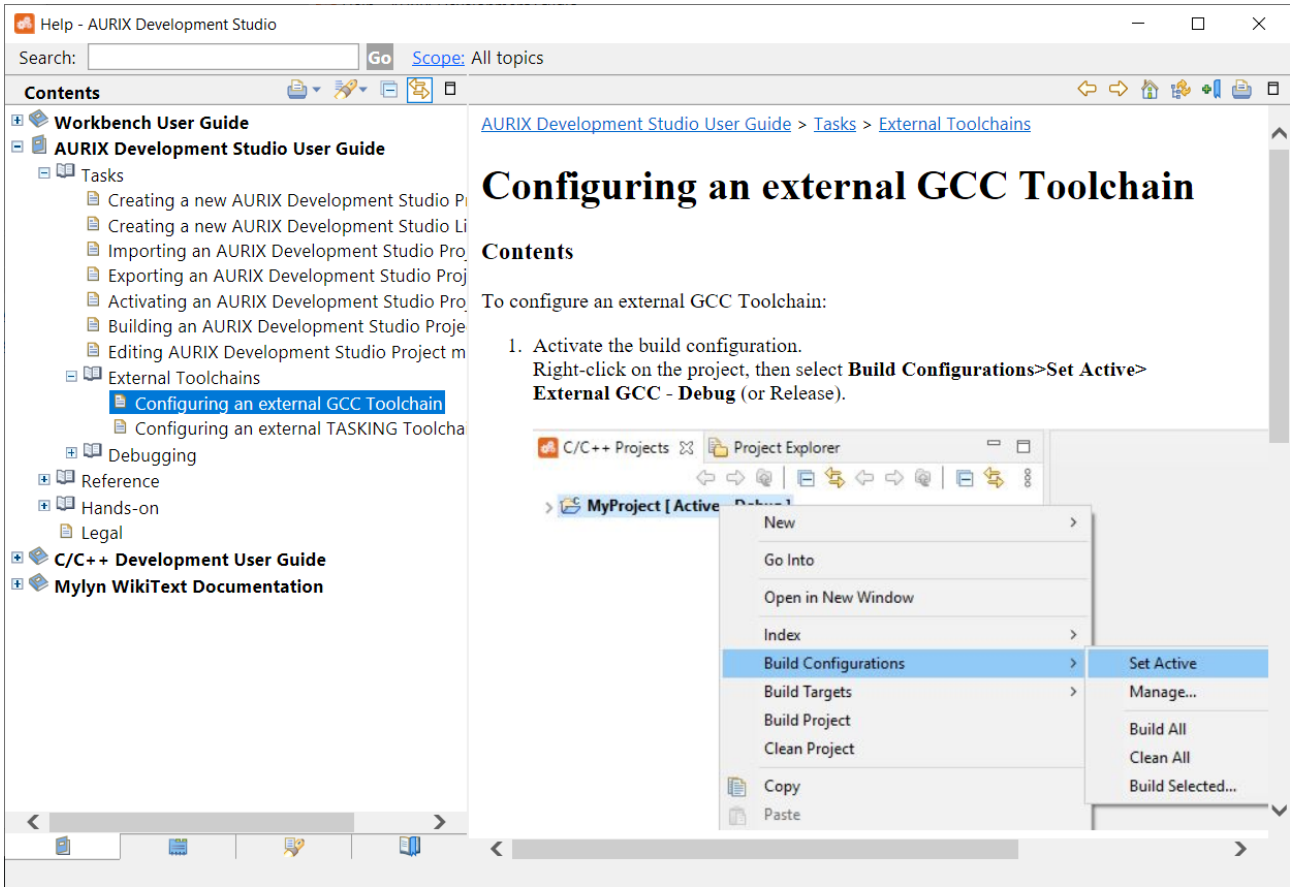

Fig. 3. Navigation inside Help Content window

3.2. New project

The toolchain configuration is bound to a project, so a new project must be created.

From the main menu, select **File→New→New AURIX Project** and specify the project's name and location. Then select the required **Device** and **Board** and click **Finish**.

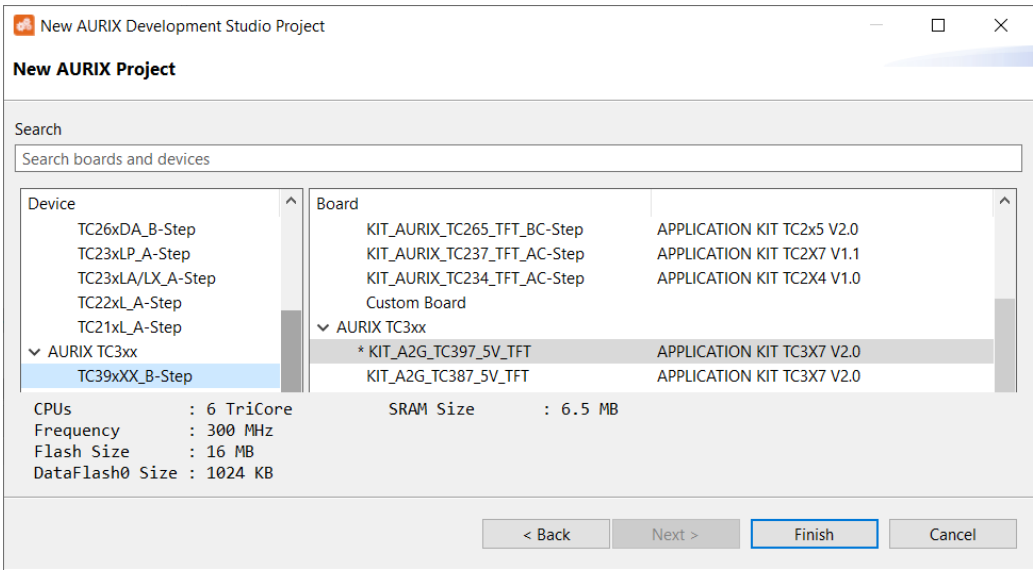

Fig. 4. Setting the device and board for the new project

3.3. Active build configuration

To make the configuration with the external GCC active, right-click on the project and select **Build Configuration→Set Active→External GCC - Debug**.

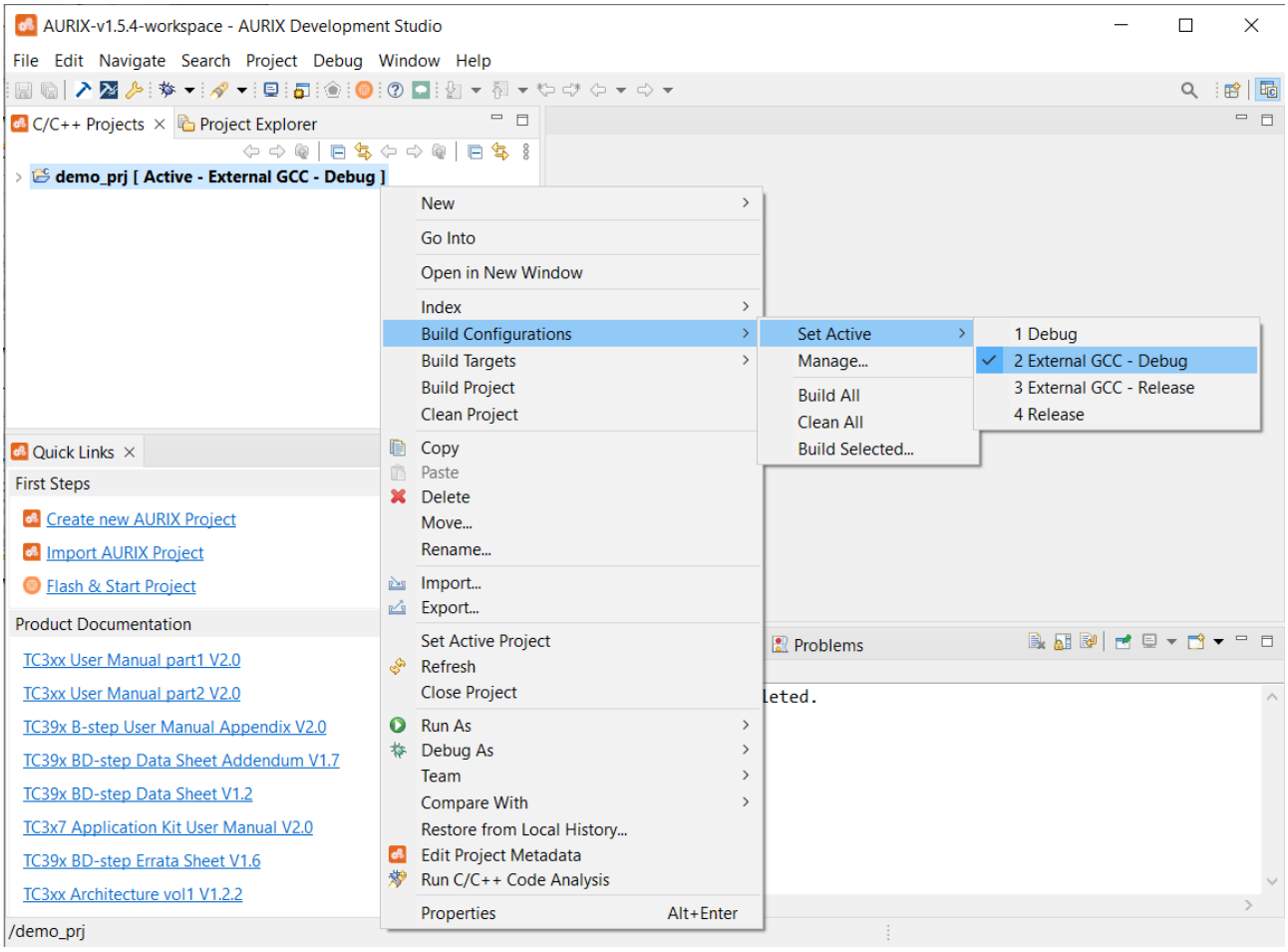

Fig. 5. Setting the active build configuration

3.4. Path to the compiler

The path and name prefix of the Hightec compiler must be updated for the active build configuration.

Right-click on the project and select **Properties→C/C++ Build→Settings**. Change the **Prefix** and **Path** fields under the **Tool Settings** tab. The prefix is always "tricore-" for AURIX devices; the path might be different.

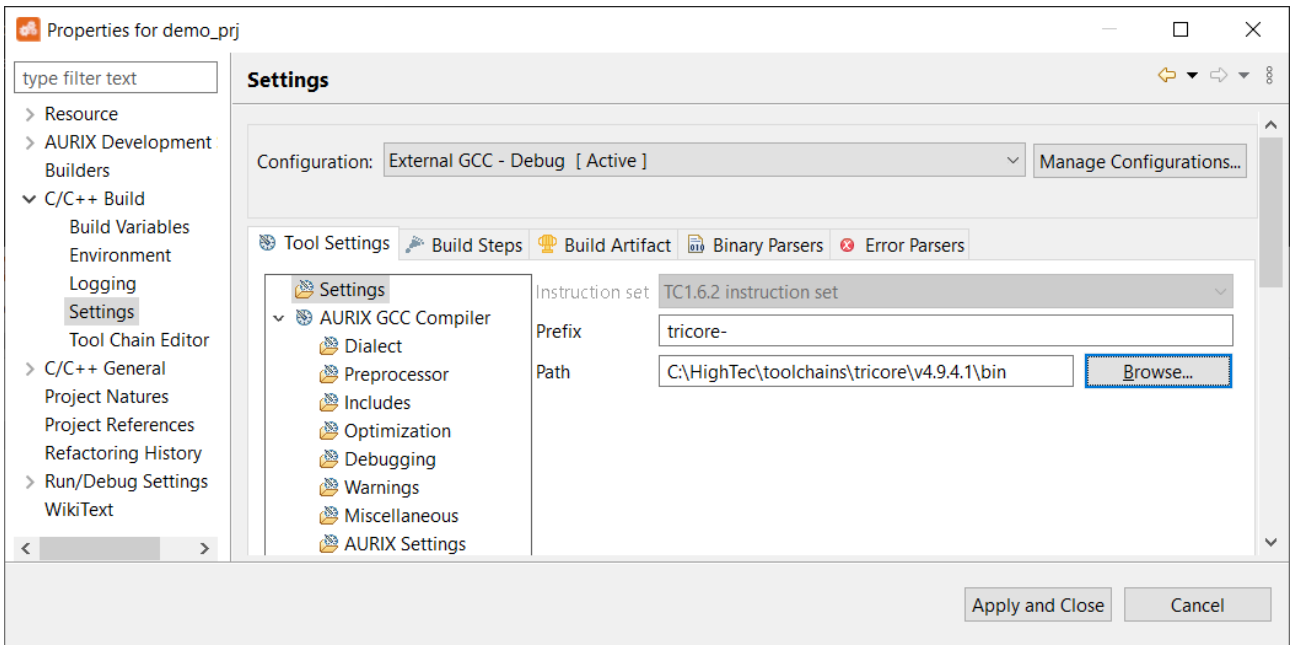

Fig. 6. Setting the prefix and path to the HighTec GCC compiler

Now, the project will use the HighTec GCC compiler to build the application.

3.5. Import existing Infineon example

To test if the HighTec compiler works correctly, let's import one of the existing Infineon projects.

Select **File→Import→Infineon AURIX Development Studio Project** and click **Next**.

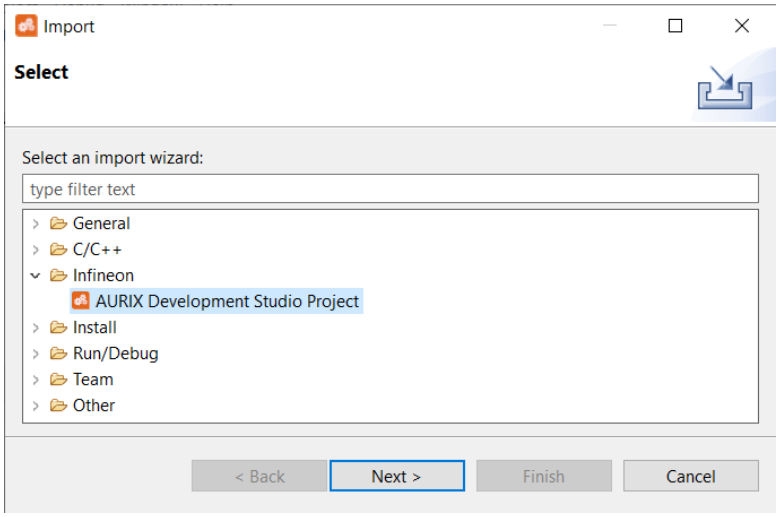

Fig. 7. Import Infineon example

Then select an example of your choice (we will test the "blinky" example for Application Kit TC397).

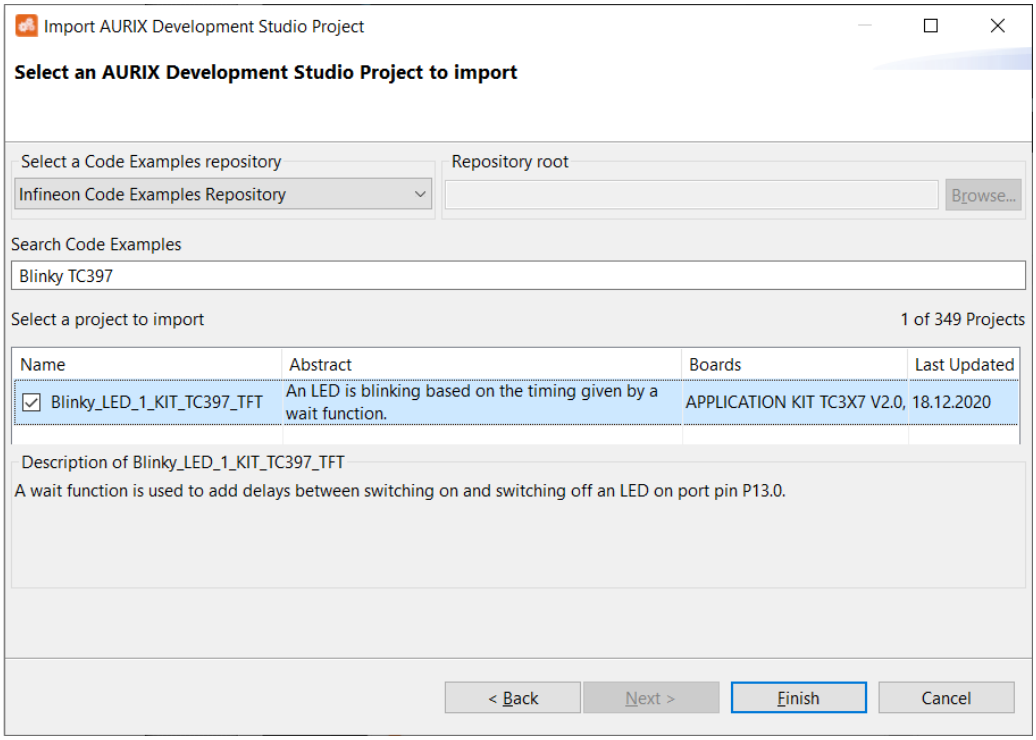

Fig. 8. Select an example from the list (search is used as a filter)

3.6. Add new configuration

To build previously imported Infineon example, a new configuration must be added. Right-click on the project and select **Build Configuration→Manage**.

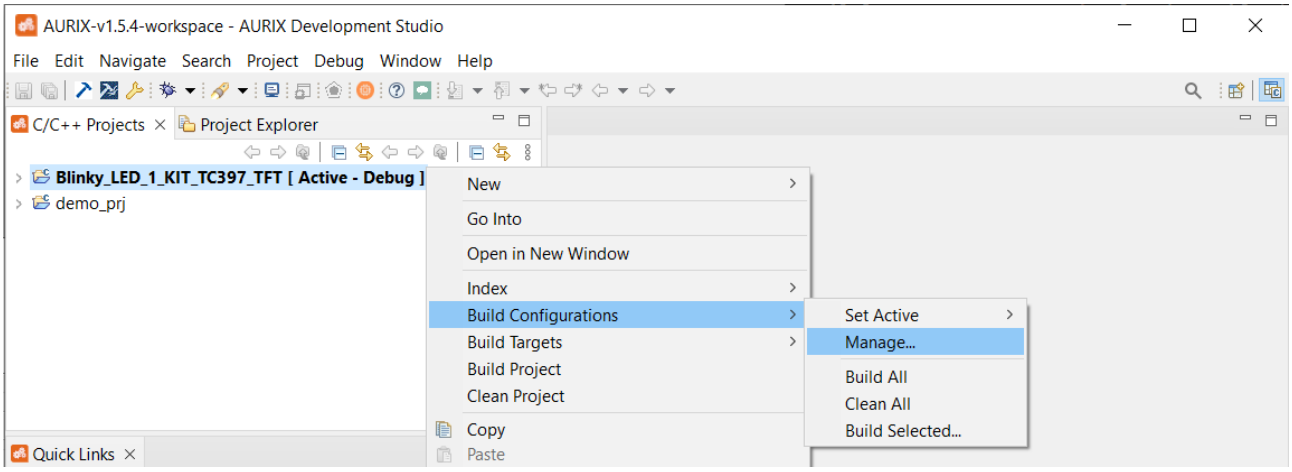

Fig. 9. Manage build configurations of the project

Add new configuration by clicking on the **New...** button. Fill in the configuration name and select the **Import from projects**. From the drop-down list, select the previously created configuration as shown on the figure below. (In our case, it is the **demo_prj > External GCC - Debug**).

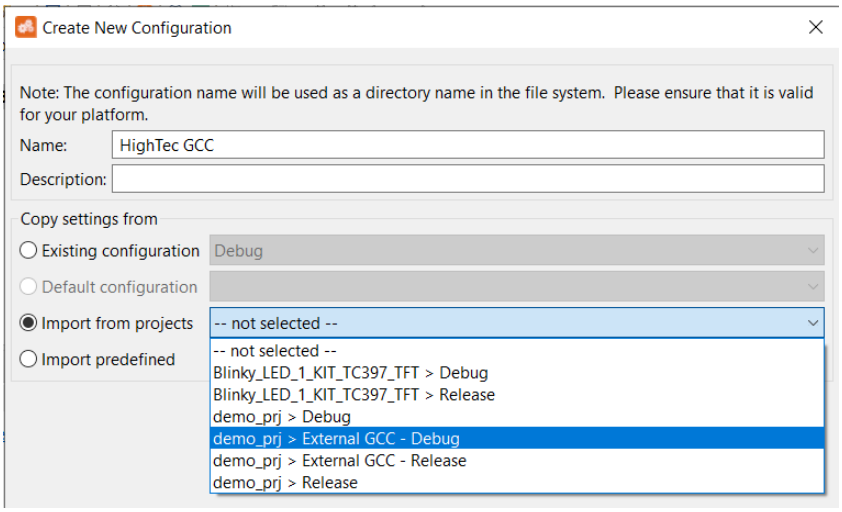

Fig. 10. Create new configuration as a copy of the existing one

Finally, select this configuration as active by clicking on **Set Active**.

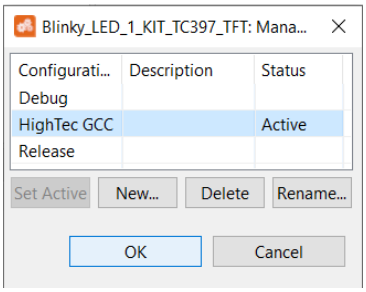

Fig. 11. Set new active build configuration

3.7. Linker script

Before the imported Infineon example can be built with HighTec compiler, the `Lcf_Gnu_Tricore_Tc.lsl` **linker script file must be copied** from the "New Project" `demo_prj` into the newly imported Infineon project.

The final project content after a successful build is shown on the below figure (contains already the copied linker file and generated HEX file).

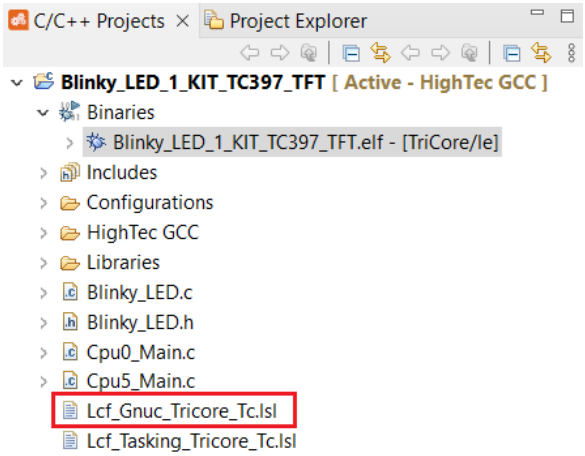

Fig. 12. Content of the imported Infineon project after successful build

Document References

[1] GNU Compiler Collection on Wikipedia

https://en.wikipedia.org/wiki/GNU_Compiler_Collection

[2] AURIX Development Studio" on Infineon pages

https://www.infineon.com/cms/en/product/promopages/aurix-development-studio/

Document history

Version	Date	Changes to the previous version
1.0	February 2022	Initial version

Disclaimer

Please Read Carefully:

This document contains descriptions for copyrighted products that are not explicitly indicated as such. The absence of the TM symbol does not infer that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this document.

The information in this document has been carefully checked and is believed to be entirely reliable. However, HighTec EDV-Systeme GmbH assumes no responsibility for any inaccuracies. HighTec EDV-Systeme GmbH neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this document or its associated product. HighTec EDV-Systeme GmbH reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages that might result.

HighTec EDV-Systeme hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

HighTec EDV-Systeme GmbH
Europaallee 19, D-66113 Saarbrücken
info@hightec-rt.com
+49-681-92613-16
www.hightec-rt.com