



AD5669R - Microcontroller No-OS Driver

Supported Devices

- [AD5669R](#)

Evaluation Boards

- [EVAL-AD5669RSDZ](#)

Overview

The [AD5669R](#) device is a low power, octal, 16-bit, buffered voltage-output DACs. It operates from a single 2.7 V to 5.5 V supply and is guaranteed monotonic by design. The AD5669R has an on-chip reference with an internal gain of 2. The user can select the AD5669R with a 1.25 V 5 ppm/°C reference (AD5669R-1), giving a full-scale output range of 2.5 V, or a 2.5 V 5 ppm/°C reference (AD5669R-2 and AD5669R-3), giving a full-scale output range of 5 V. The on-chip reference is off at power-up, allowing the use of an external reference. The internal reference is enabled via a software write. The part incorporates a power-on reset circuit that ensures that the DAC output powers up to 0 V and remains powered up at this level until a valid write takes place. The part contains a power-down feature that reduces the current consumption of the device to 400 nA at 5 V and provides software-selectable output loads while in power-down mode for any or all DAC channels.



The goal of this project (Microcontroller No-OS) is to be able to provide reference projects for lower end processors, which can't run Linux, or aren't running a specific operating system, to help those customers using microcontrollers with ADI parts. Here you can find a generic driver which can be used as a base for any microcontroller platform and also specific drivers for Renesas platforms.

HW Platform(s):

- [Renesas Demo Kit for RL78G13 \(Renesas\)](#)

Driver Description

The driver contains two parts:

- The driver for the AD5669R part, which may be used, without modifications, with any microcontroller.
- The Communication Driver, where the specific communication functions for the desired type of processor and communication protocol have to be implemented. This driver implements the communication with the device and hides the actual details of the communication protocol to the ADI driver.

The Communication Driver has a standard interface, so the AD5669R driver can be used exactly as it is provided.

There are three functions which are called by the AD5669R driver:

- I2C_Init() - initializes the communication peripheral.
- I2C_Write() - writes data to the device.

- I2C_Read() - reads data from the device.



I2C driver architecture

The following functions are implemented in this version of AD5669R driver:

Function	Description
unsigned char AD5669R_Init(void)	Resets the device and performs several initializations.
void AD5669R_PowerMode(unsigned char pwrMode)	Sets the device in a specific power mode.
void AD5669R_Reset(void)	Resets the device.
void AD5669R_SetInputRegister(unsigned long registerValue)	Writes a 24-bit data-word to the Input Register of the device.
unsigned short AD5669R_GetRegisterValue(void)	Reads the last register written by user.

Downloads

- [AD5669R Generic Driver](#)
- [AD5669R RL78G13 Driver](#)
- [AD5669R RX62N Driver](#)

Renesas RL78G13 Quick Start Guide

This section contains a description of the steps required to run the AD5669R demonstration project on a Renesas RL78G13 platform.

Required Hardware

- [Renesas Demo Kit for RL78G13 \(Renesas\)](#)
- [EVAL-AD5669RSDZ\(ADI\)](#)

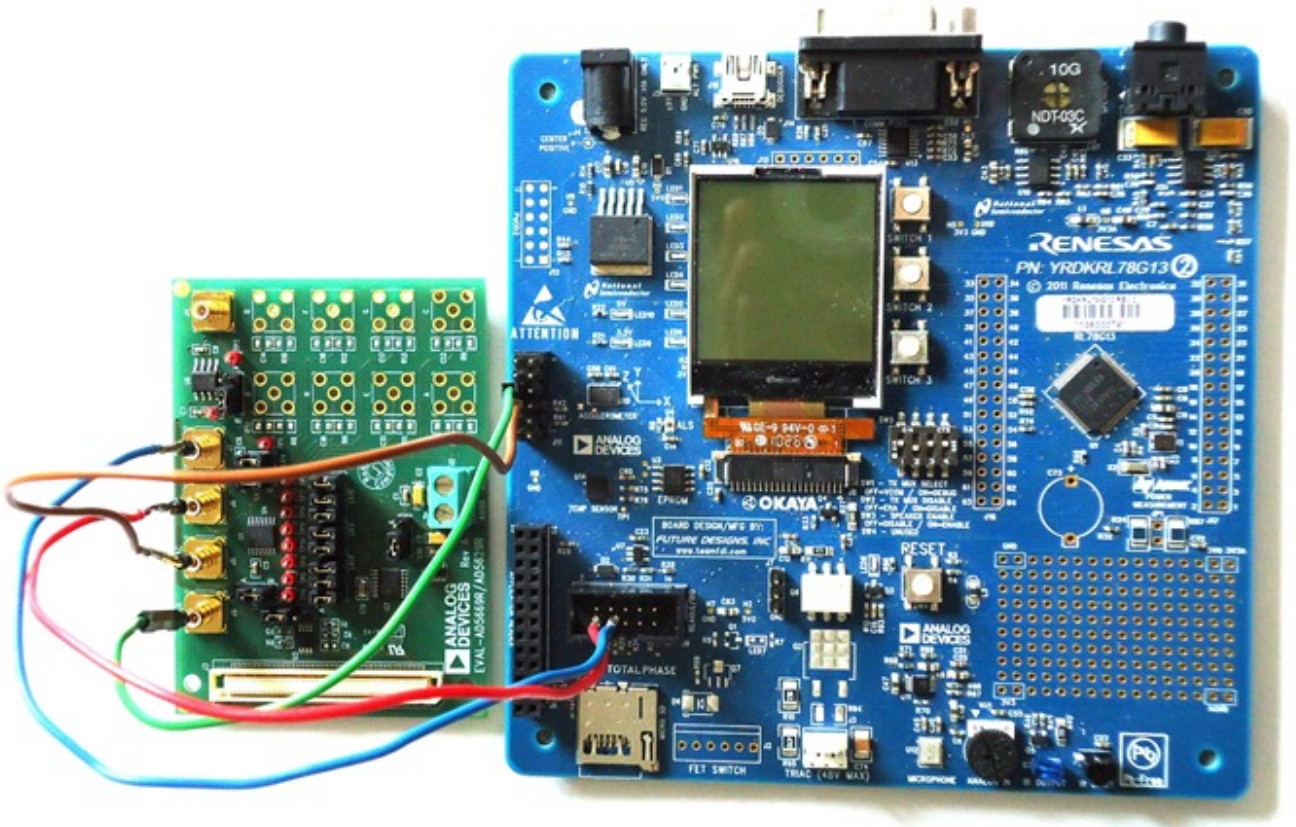
Required Software

- [IAR Embedded Workbench for Renesas RL78 Kickstart](#)
- [Applilet3 for RL78G13](#)

Hardware Setup

An EVAL-AD5669RSDZ has to be interfaced with the Renesas Demonstration Kit (RDK) for RL78G13:

EVAL-AD5669RSDZ J6 connector Pin 1	SCL (SCL)	→ YRDKRL78G13 J9
EVAL-AD5669RSDZ J7 connector Pin 3	SDA (SDA)	→ YRDKRL78G13 J9
EVAL-AD5669RSDZ J8 connector Pin 9	\LDAC (LDAC)	→ YRDKRL78G13 J11
EVAL-AD5669RSDZ J9 connector Pin 10	\CLR (CLR)	→ YRDKRL78G13 J11



Software Setup

With the **Applilet3 for RL78G13** tool the following peripherals have to be configured:

CSI10 (Clocked Serial Interface 10) - For the ST7579 LCD

Choose to generate the Transmit/receive function for the CSI10 and configure the interface with the following settings:

- Transfer mode setting: **Single transfer mode**
- Data length setting : **8 bits**
- Transfer direction setting: **MSB**
- Specification of data timing: **Type 1**
- Transfer rate setting - Clock mode: **Internal clock (master)**
- Transfer rate setting - Baudrate: **1000000** (bps)

- Interrupt setting - Transfer interrupt priority (INTCSI10): **Low**
- Uncheck the callback functions.

TM00 (Timer 00) - For the DelayMs() function

Configure TM00 as an interval timer:

- Interval timer setting - Interval value(16 bits): **1** ms
- Interval timer setting - Uncheck *Generates INTM00 when counting is started*
- Interrupt setting - Uncheck *End of timer channel 0 count, generate an interrupt (INTM00)*

Watchdog Timer

Disable the watchdog timer:

- Choose for the Watchdog timer operation setting: **Unused** option.

Reference Project Overview

The reference project powers-up all DAC channels, displays the content of the power register and starts to generate a saw-tooth wave.



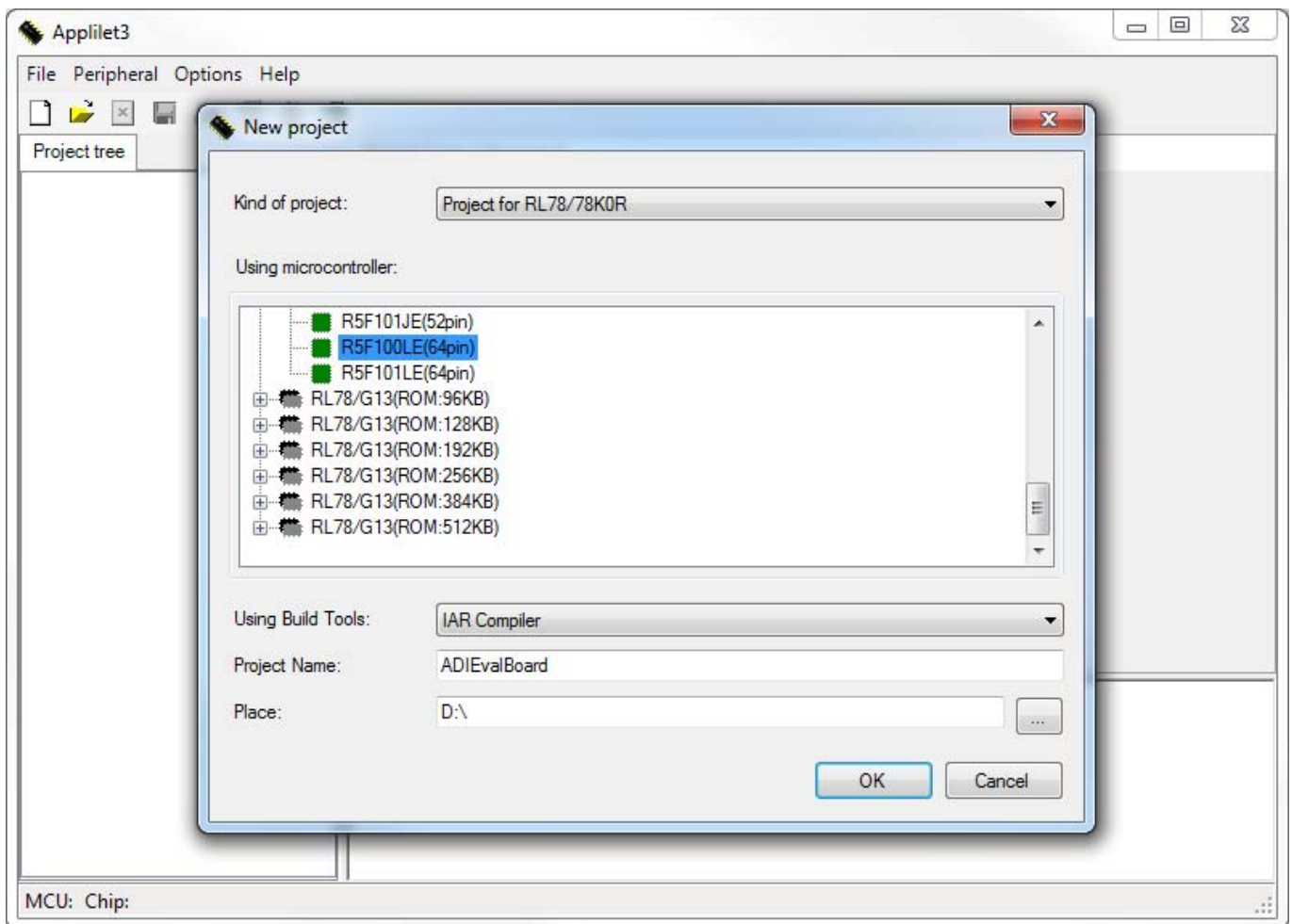
Software Project Setup

This section presents the steps for developing a software application that will run on the **Renesas Demo Kit for RL78G13** for controlling and monitoring the operation of the **ADI** part.

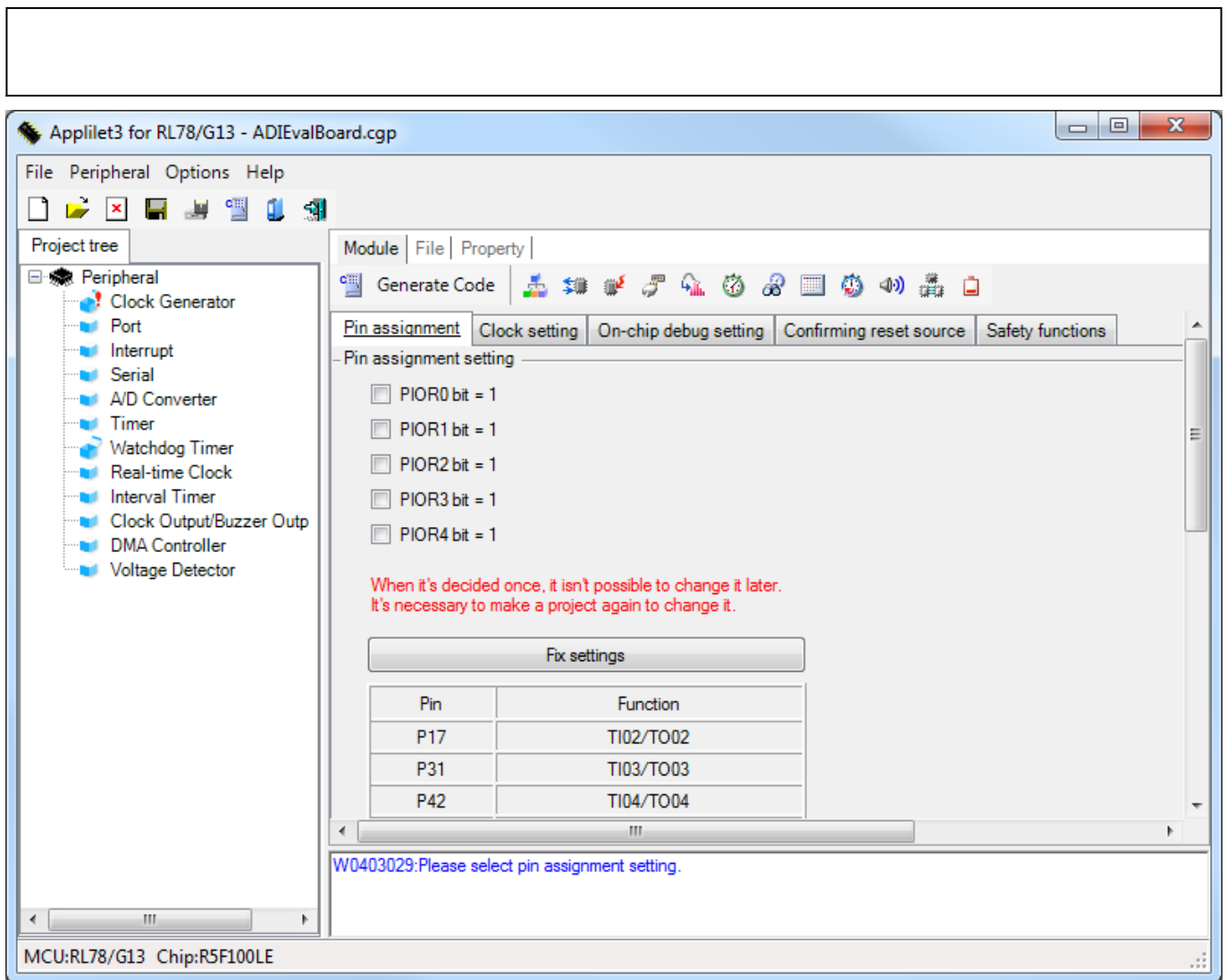
Two software applications have to be used: **Applilet3 for RL78G13** (a tool that automatically generates device drivers for MCU peripheral functions) and **IAR Embedded Workbench for Renesas RL78** (the integrated development environment).

Step 1 - Applilet3 for RL78G13

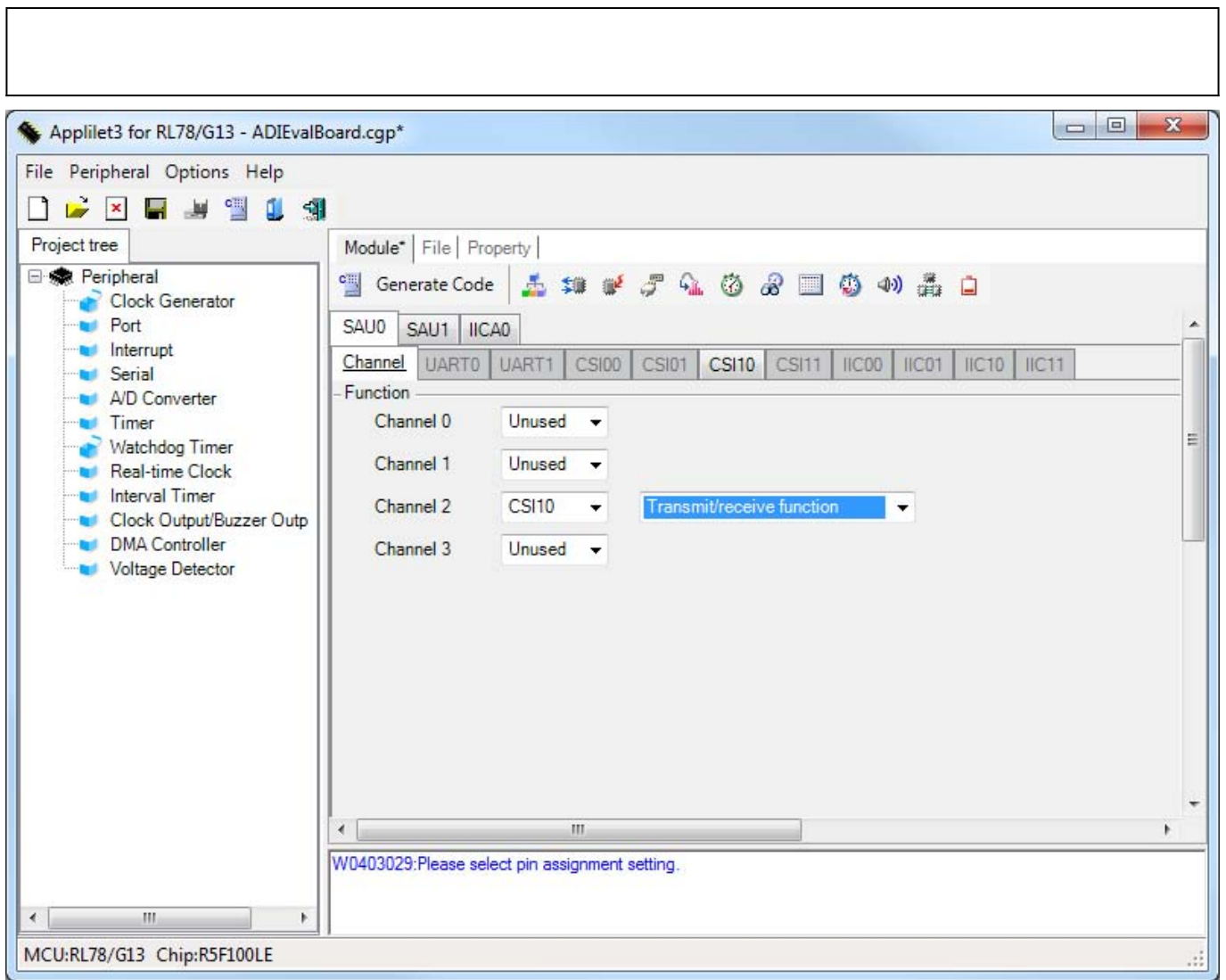
- Run the **Applilet3 for RL78G13** tool and create a new project for **R5F100LE** processor. Select **IAR Compiler** build tool, a project name, a location for the new project and press **OK**.



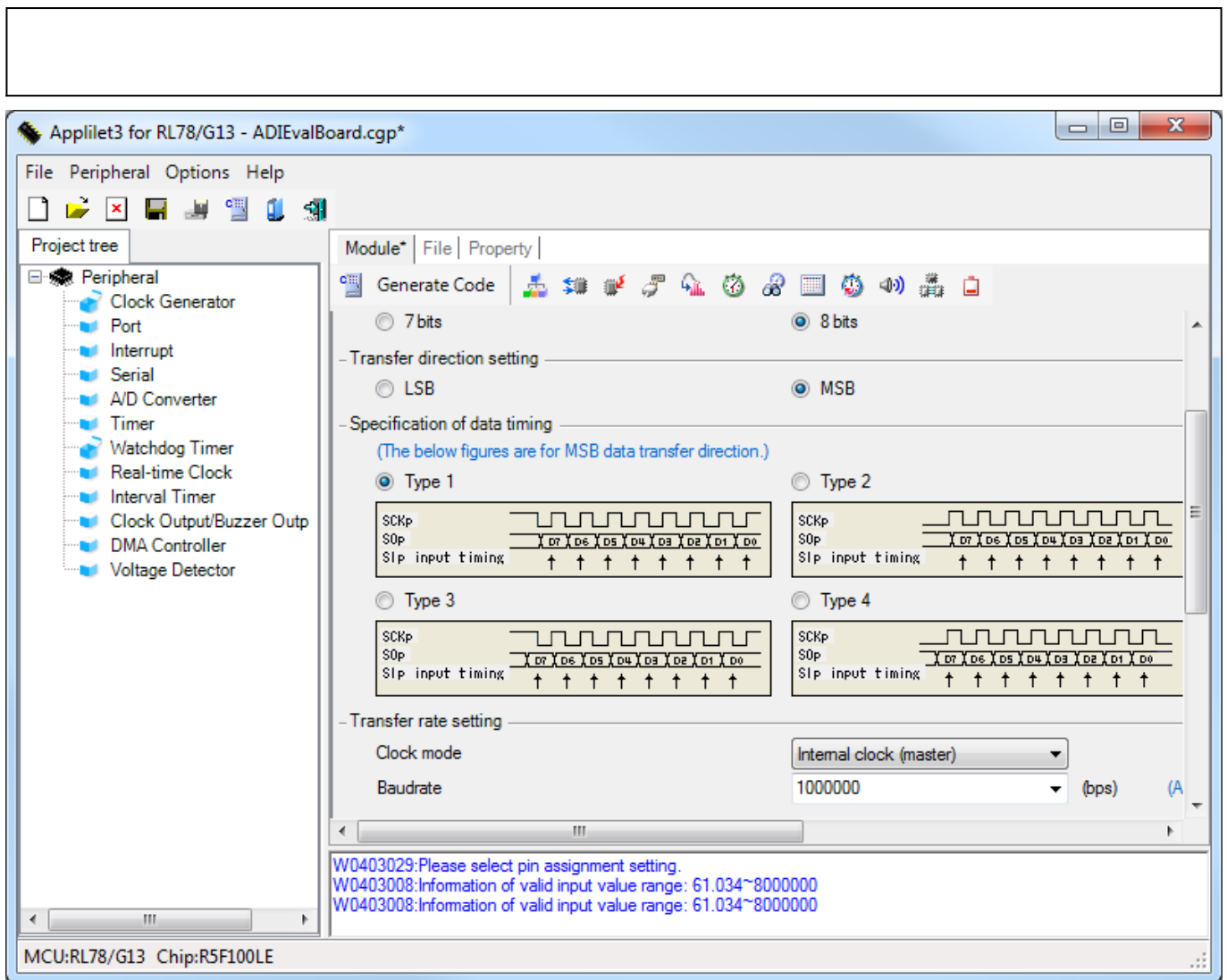
- Keep the default **Pin assignment** setting and click **Fix settings**.



- Now the desired peripherals can be configured and the code can be generated. For example, if the clocked serial interface 10 (**CSI10**) has to be configured, select the **Serial** peripheral, choose for the Channel 2 of Serial Array Unit 0 (**SAU0**) the **CSI10** interface, **Transmit/receive function** option and then go to **CSI10** tab.



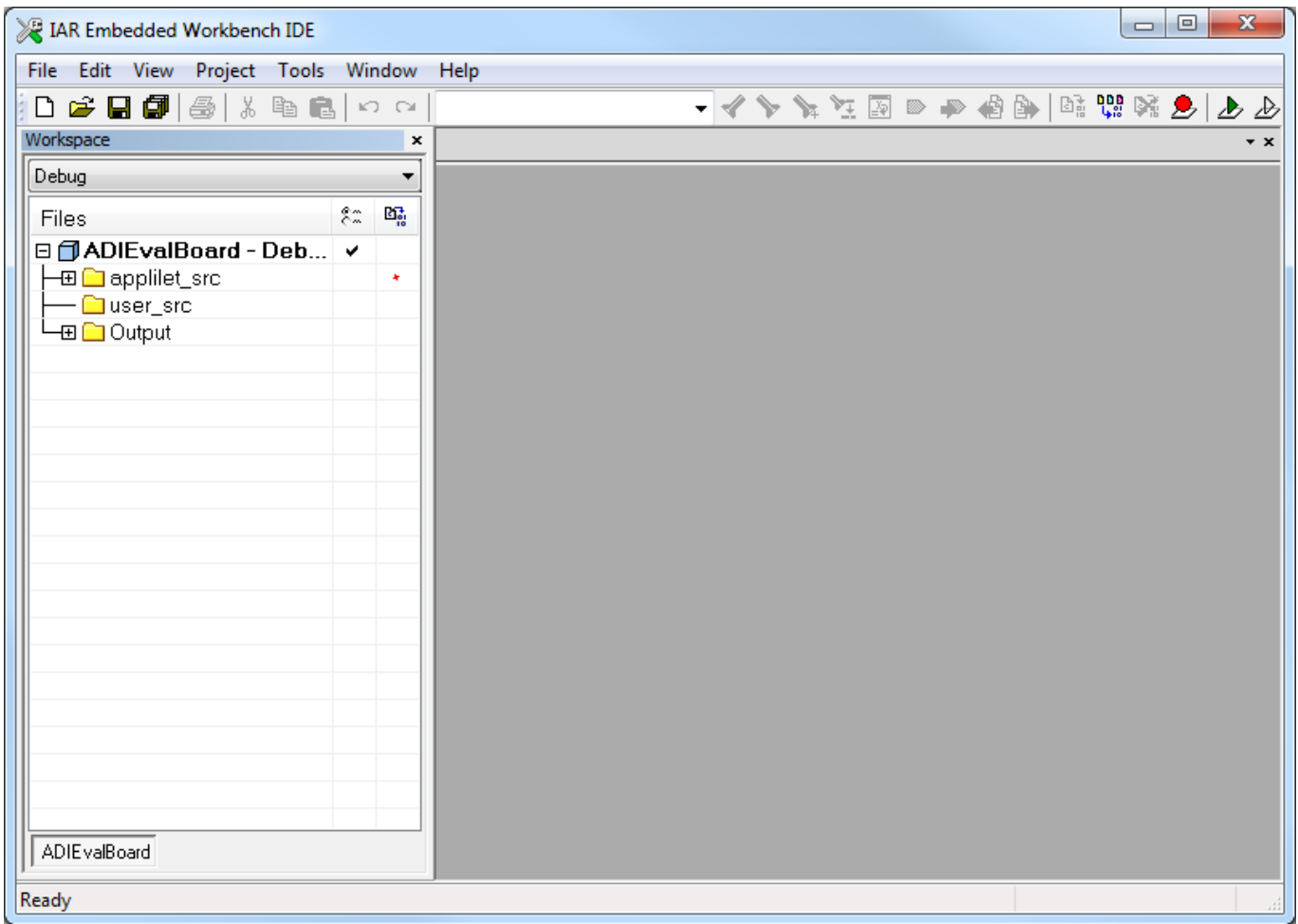
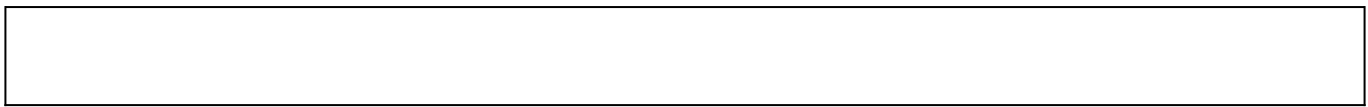
- To configure the **CS10** interface for serial transmissions of 8 bits, with MSB first, with the data captured on clock's rising edge, with a frequency of the clock of 1 MHz and the idle state high, the settings from the following image have to be made.



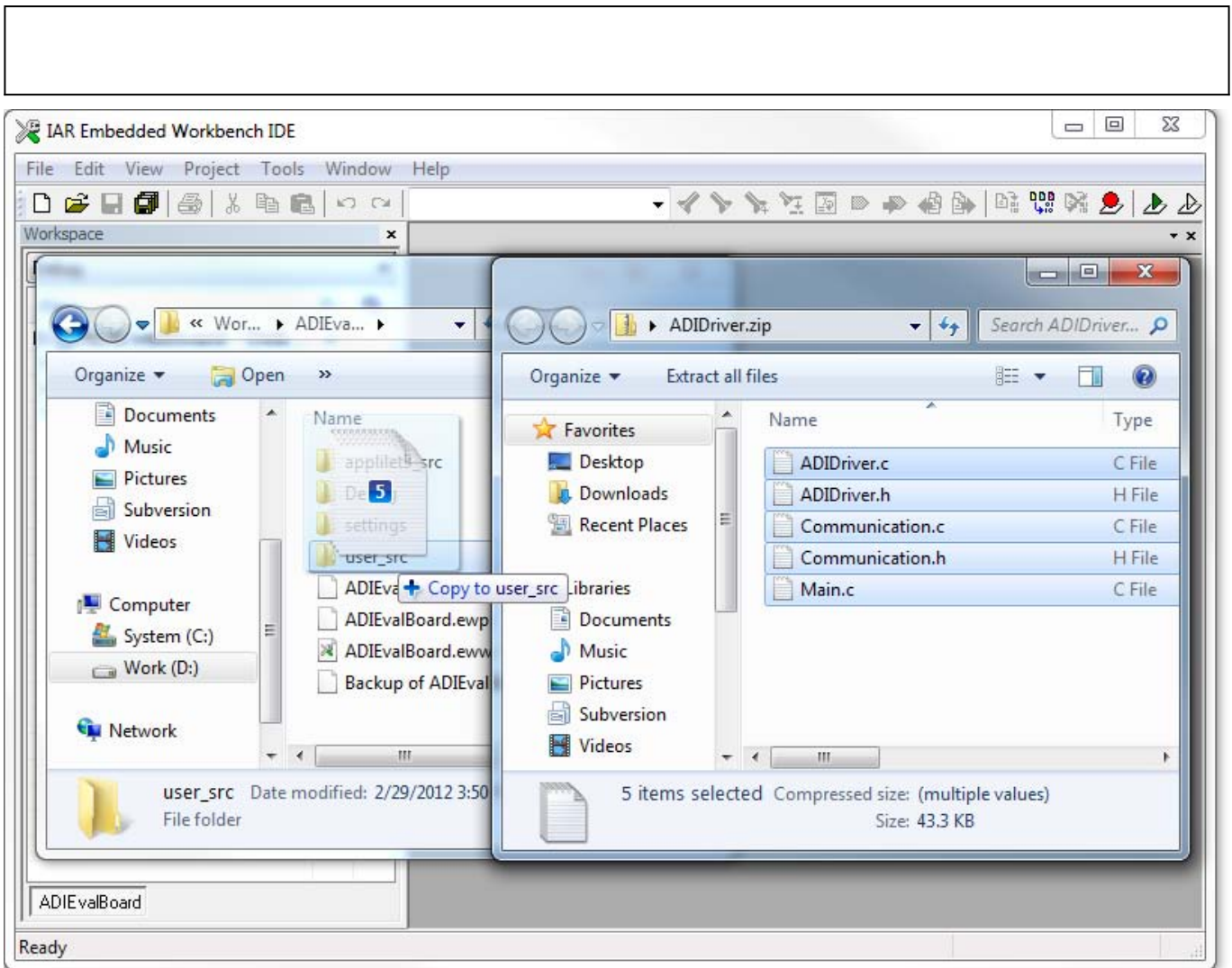
- After all the desired peripherals are configured click on the **Generate Code** button and a new workspace and a new project for the **IAR Embedded Workbench** will be generated. After the code was generated close the **Applilet3 for RL78G13** tool.

Step 2 - IAR Embedded Workbench for Renesas RL78

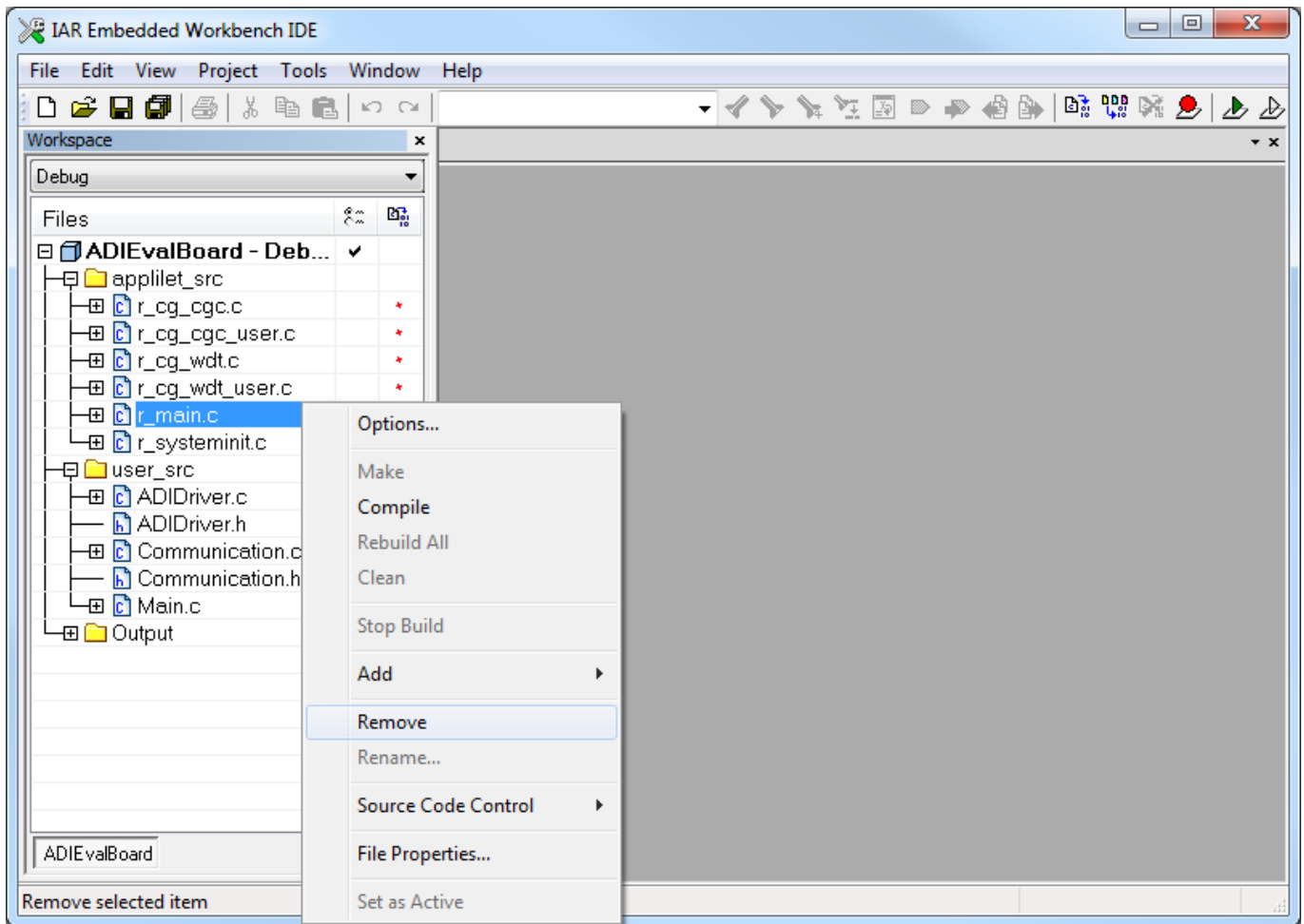
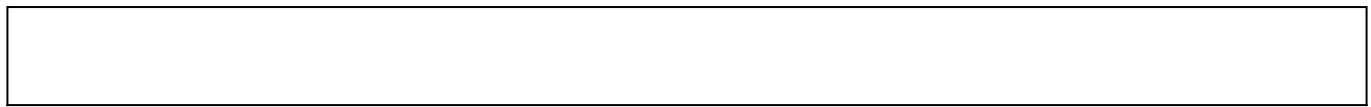
- Run the **IAR Embedded Workbench** and open the workspace created with the **Applilet3** tool.



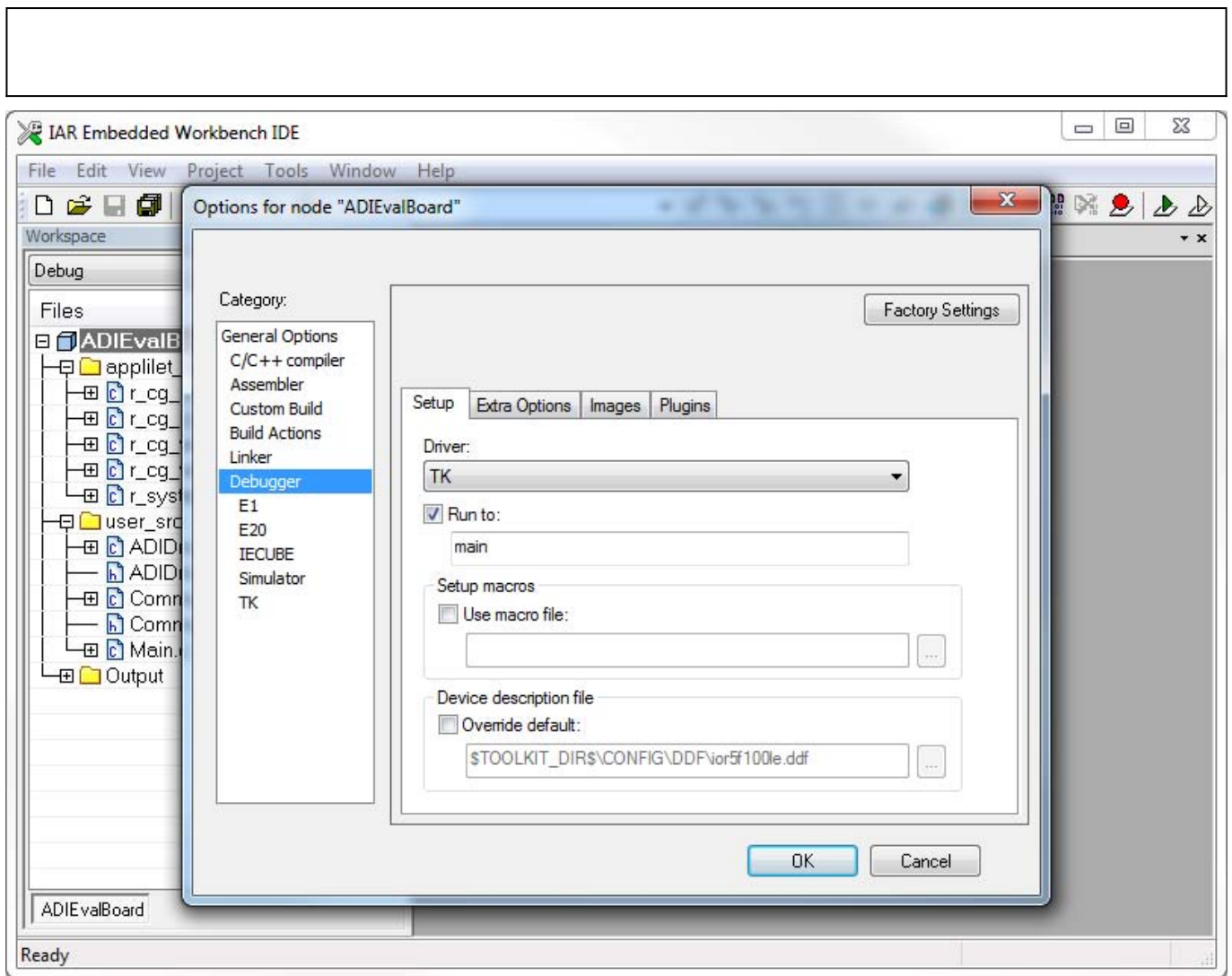
- Copy the files extracted from the zip file into the **user_src** folder, located in the project's folder.



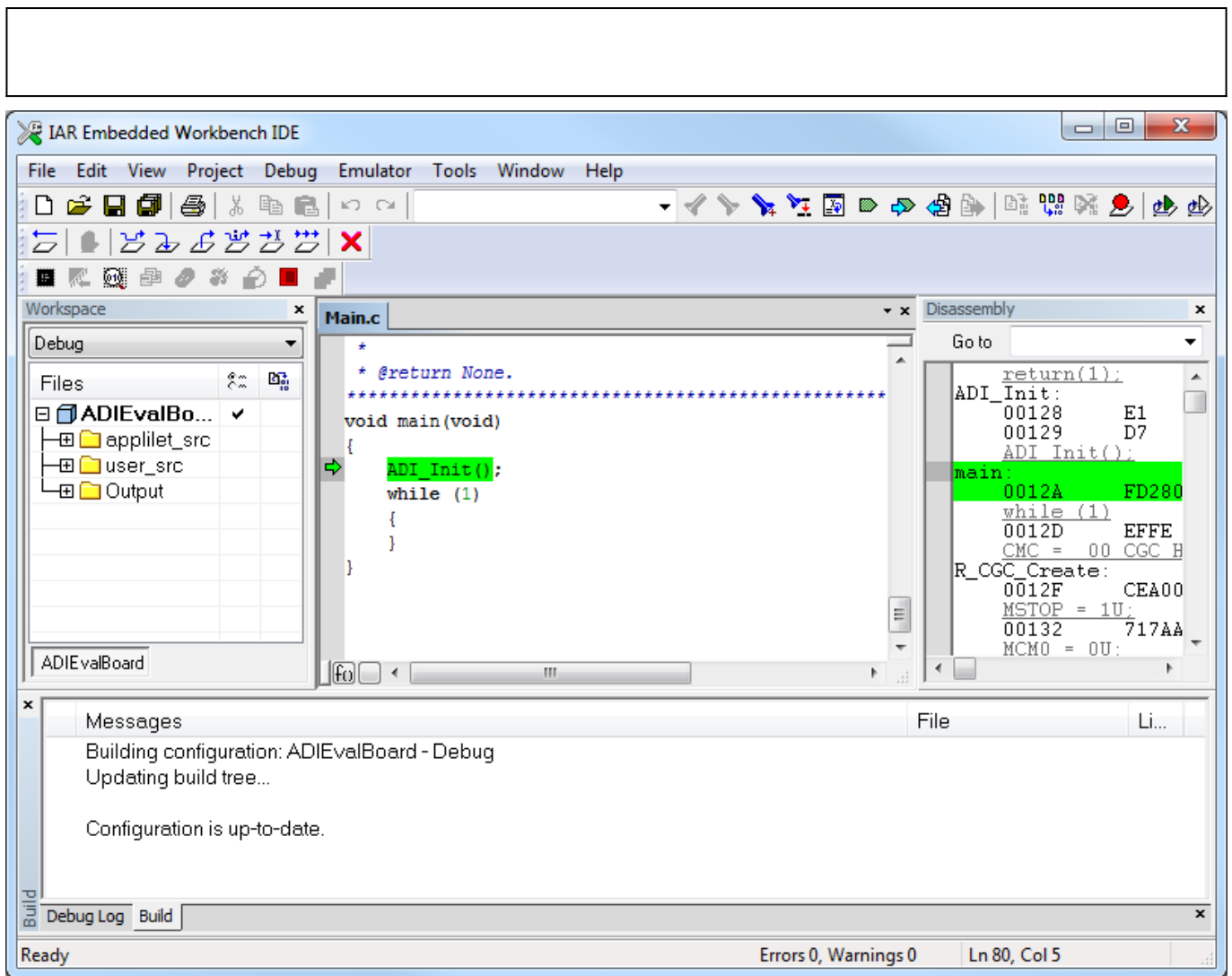
- The new source files have to be included into the project. Add in the **user_src** group the files from the corresponding folder (Right click on the group and select **Add - Add Files...**). Because a new **Main** file was included the **r_main.c** file from the **appliance_src** group has to be deleted (Right click on the file and select **Remove**).



- Now the debugger driver has to be selected from the project's options. Right click on the project name and select **Options**. From the **Debugger** category choose the **TK Debugger Driver**.



- Now, the project is ready to be compiled and downloaded on the board. Press the **F7** key to compile it. Press **CTRL + D** to download and debug the project.



29 Feb 2012 16:01 · [Dragos Bogdan](#)

Renesas RX62N Quick Start Guide

This section contains a description of the steps required to run the AD5669R demonstration project on a Renesas RX62N platform.

Required Hardware

- [Renesas Demo Kit for RX62N \(Renesas\)](#)
- [EVAL-AD5669REBZ \(ADI\)](#)

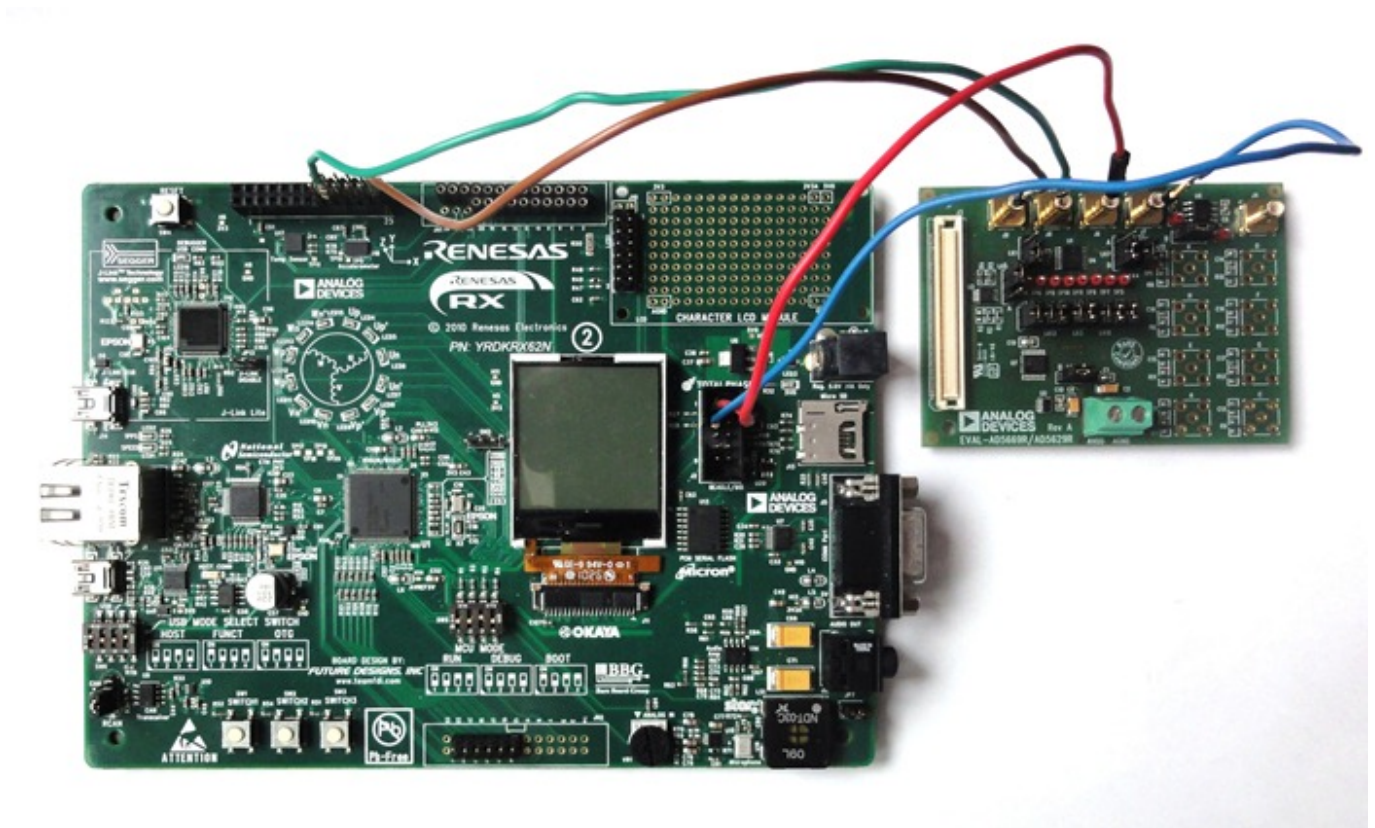
Required Software

- [High-performance Embedded Workshop for RX62N family](#)

Hardware Setup

An EVAL-AD5669REBZ board has to be interfaced with the Renesas Demonstration Kit (RDKit) for RX62N:

EVAL-AD5669RSDZ J6 connector Pin 1	SCL (SCL)	→ YRDKRX62N J2 connector Pin 1
EVAL-AD5669RSDZ J7 connector Pin 3	SDA (SDA)	→ YRDKRX62N J2 connector Pin 3
EVAL-AD5669RSDZ J8 connector Pin 17	\LDAC (LDAC)	→ YRDKRX62N J8 connector Pin 17
EVAL-AD5669RSDZ J9 connector Pin 25	\CLR (CLR)	→ YRDKRX62N J8 connector Pin 25



Reference Project Overview

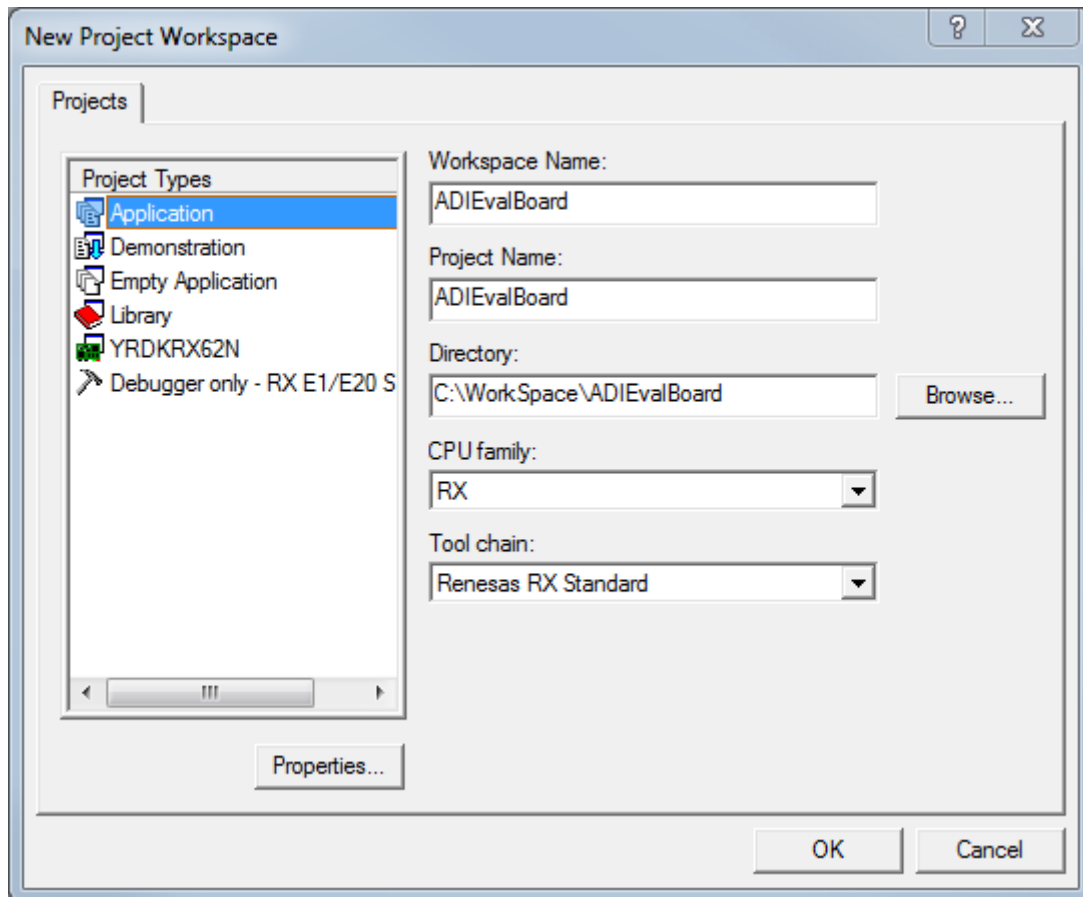
The reference project: The reference project powers-up all DAC channels, displays the content of the power register and starts to generate a saw-tooth wave.



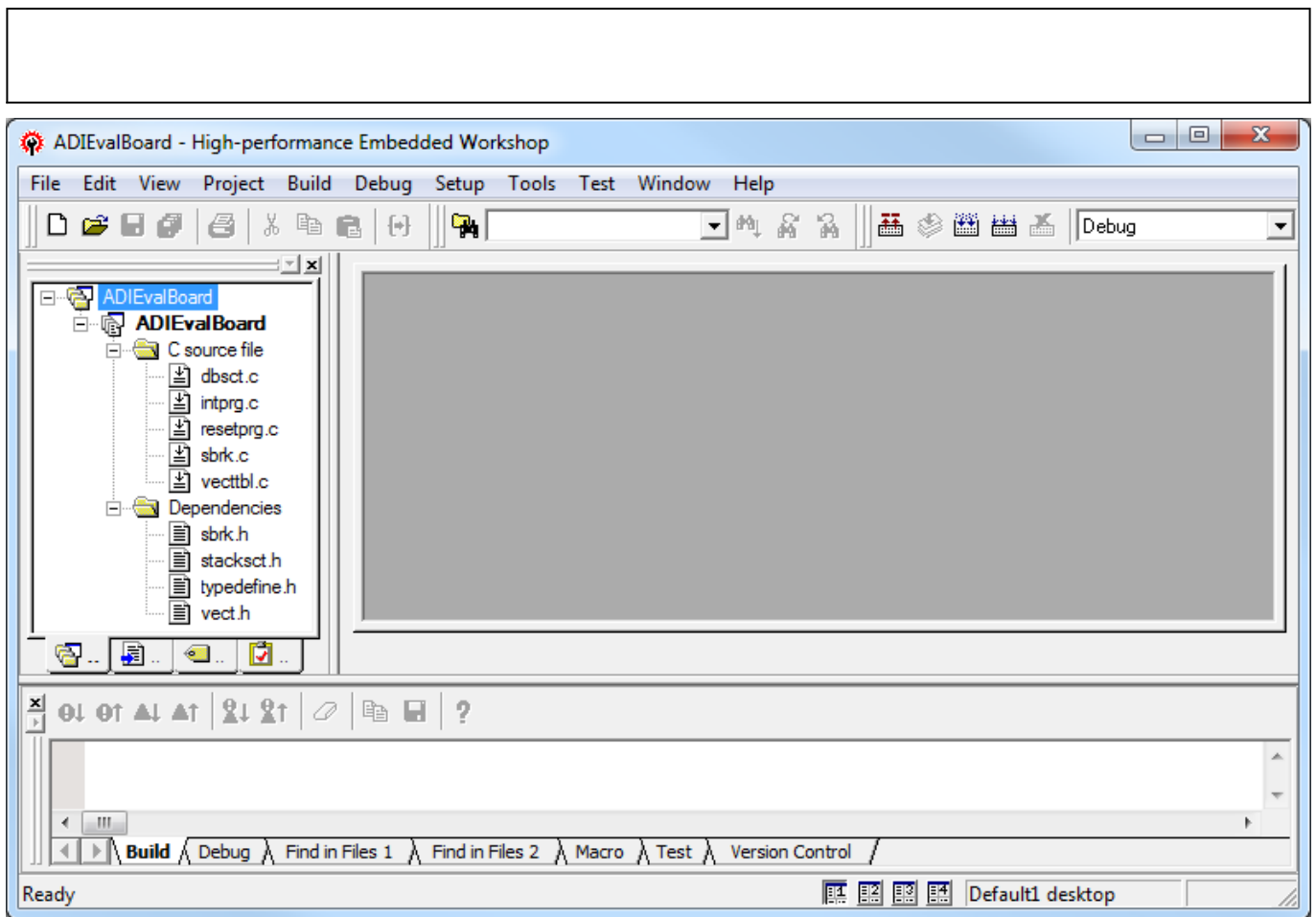
Software Project Setup

This section presents the steps for developing a software application that will run on the **Renesas Demo Kit for RX62N** for controlling and monitoring the operation of the ADI part.

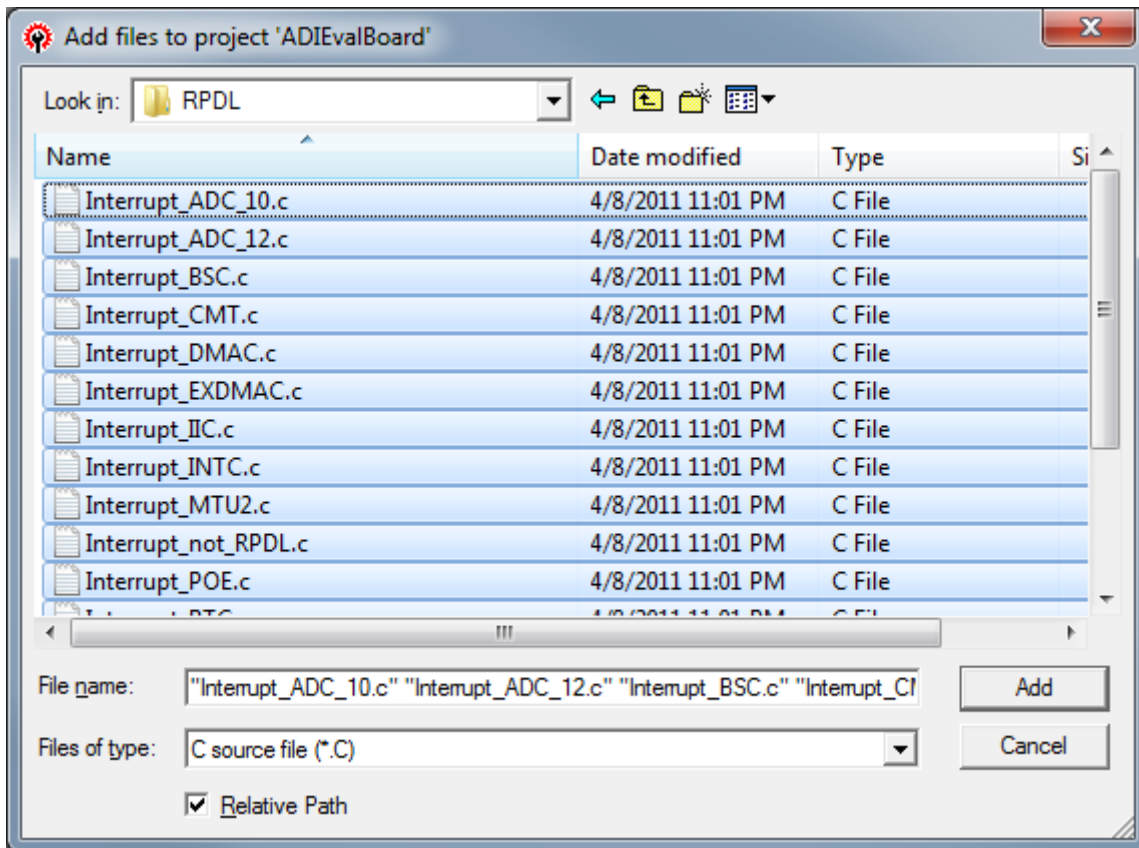
- Run the **High-performance Embedded Workshop** integrated development environment.
- A window will appear asking to create or open project workspace. Choose “Create a new project workspace” option and press *OK*.
- From “Project Types” option select “Application”, name the Workspace and the Project “ADIEvalBoard”, select the “RX” CPU family and “Renesas RX Standard” tool chain. Press *OK*.



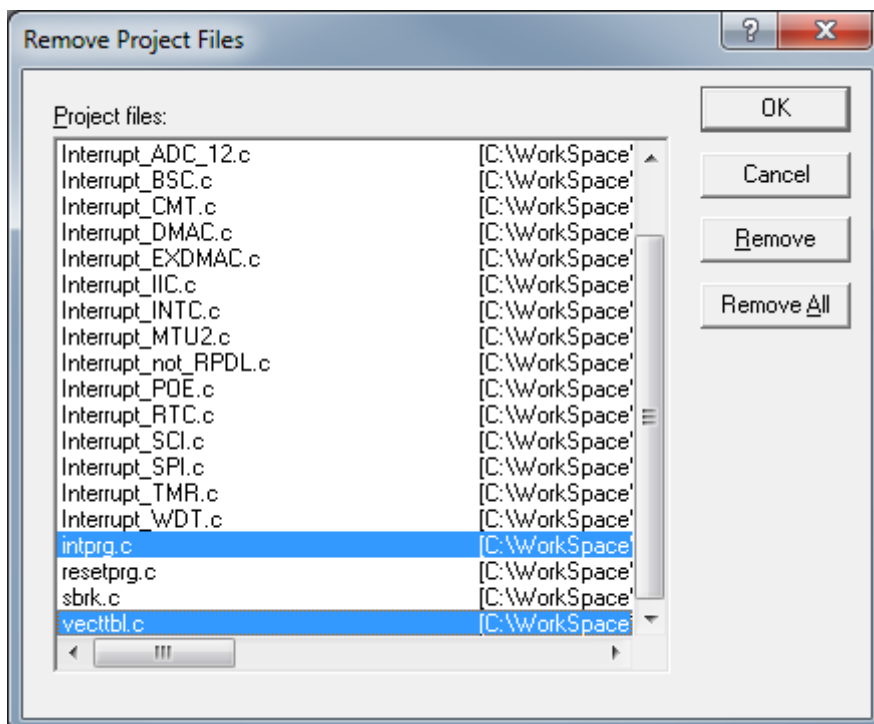
- A few windows will appear asking to configure the project:
 - In the "Select Target CPU" window, select "RX600" CPU series, "RX62N" CPU Type and press Next.
 - In the "Option Setting" windows keep default settings and press Next.
 - In the "Setting the Content of Files to be generated" window select "None" for the "Generate main() Function" option and press Next.
 - In the "Setting the Standard Library" window press "Disable all" and then Next.
 - In the "Setting the Stack Area" window check the "Use User Stack" option and press Next.
 - In the "Setting the Vector" window keep default settings and press Next.
 - In the "Setting the Target System for Debugging" window choose "RX600 Segger J-Link" target and press Next.
 - In the "Setting the Debugger Options" and "Changing the Files Name to be created" windows keep default settings, press Next and Finish.
- The workspace is created.



- The RPD (Renesas Peripheral Driver Library) has to be integrated in the project. Unzip the RPD files (double-click on the file "RPD_RX62N.exe"). Navigate to where the RPD files were unpacked and double-click on the "Copy_RPD_RX62N.bat" to start the copy process. Choose the LQFP package, type the full path where the project was created and after the files were copied, press any key to close the window.
- The new source files have to be included in the project. Use the key sequence *Alt, P, A* to open the "Add files to project 'ADIEvalBoard'" window. Double-click on the RPD folder. From the "Files of type" drop-down list, select "C source file (*.C)". Select all of the files and press *Add*.

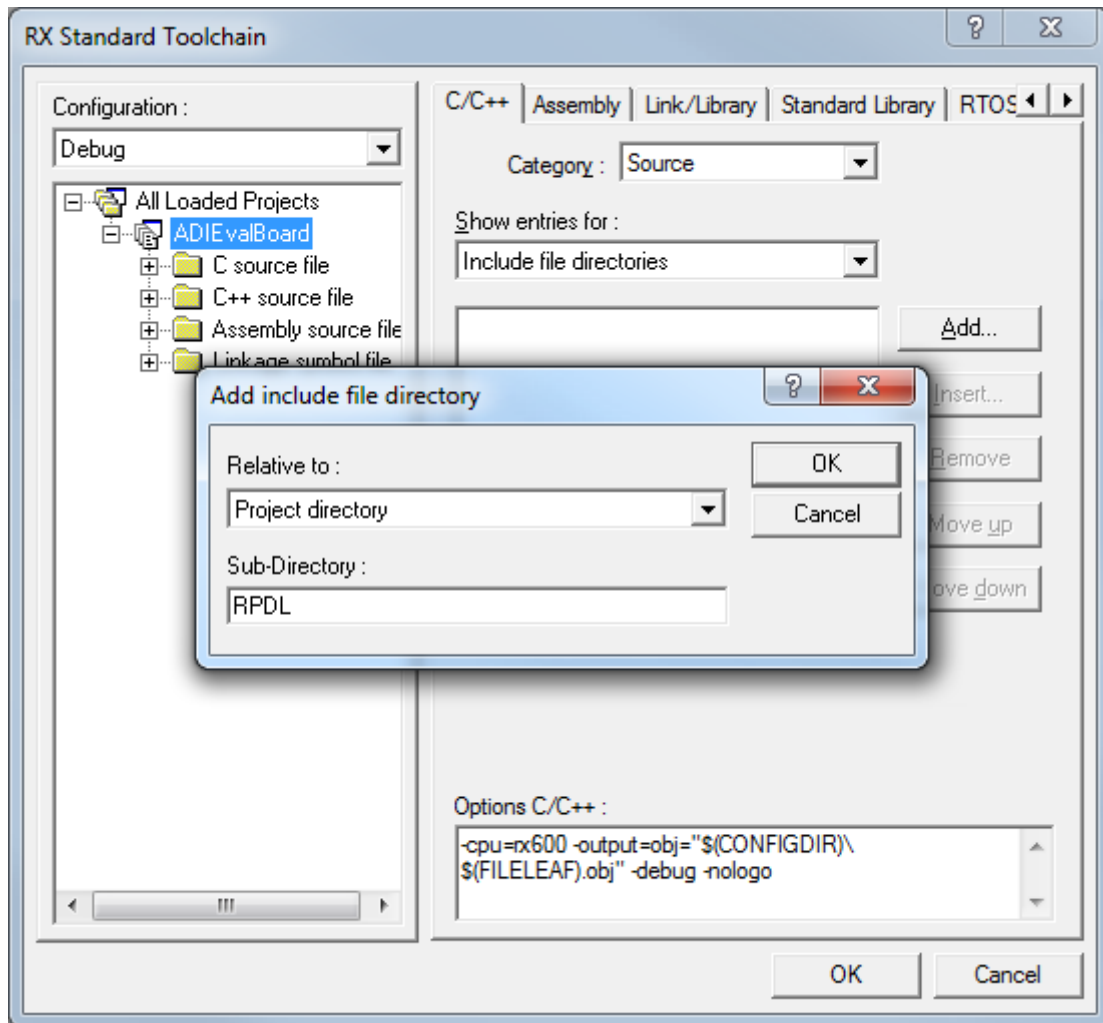


- To avoid conflicts with standard project files remove the files *intprg.c* and *vecttbl.c* which are included in the project. Use the key sequence *Alt, P, R* to open the *Remove Project Files* window. Select the files, click on Remove and press *OK*.

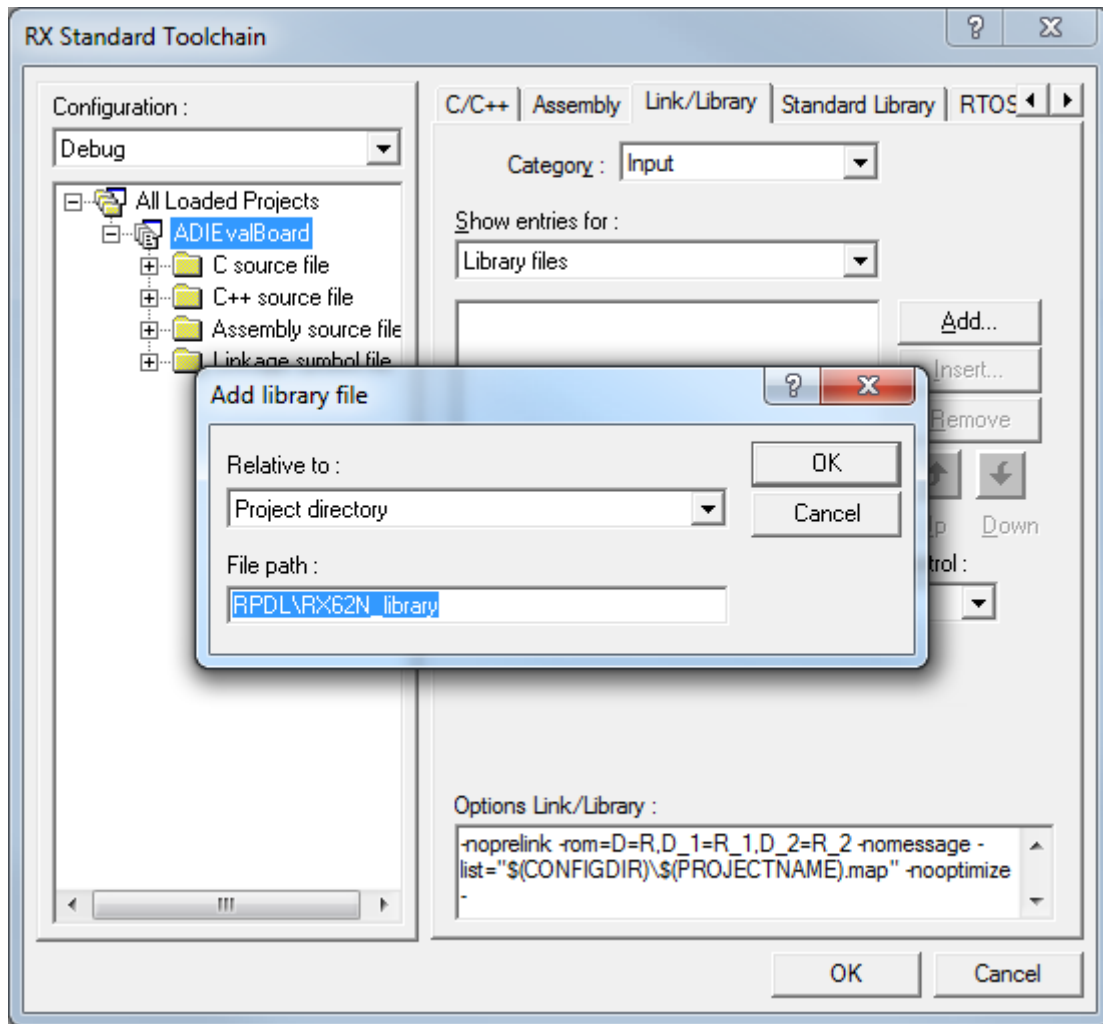


- Next the new directory has to be included in the project. Use the key sequence *Alt, B, R* to open the *RX Standard Toolchain* window. Select the C/C++ tab, select *Show entries for: Include file directories* and press *Add*. Select *Relative to: Project directory*, type *RPDL* as sub-directory and

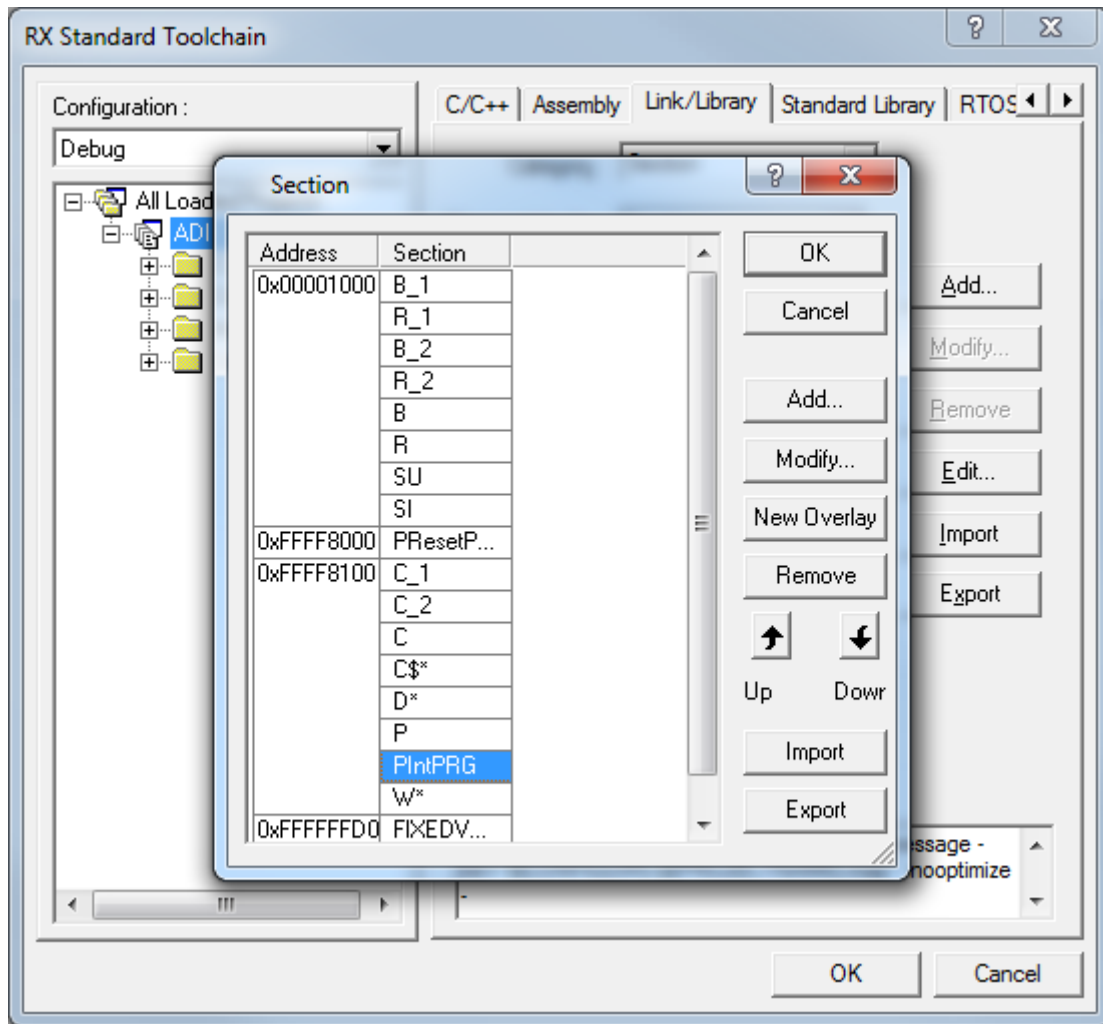
press OK.



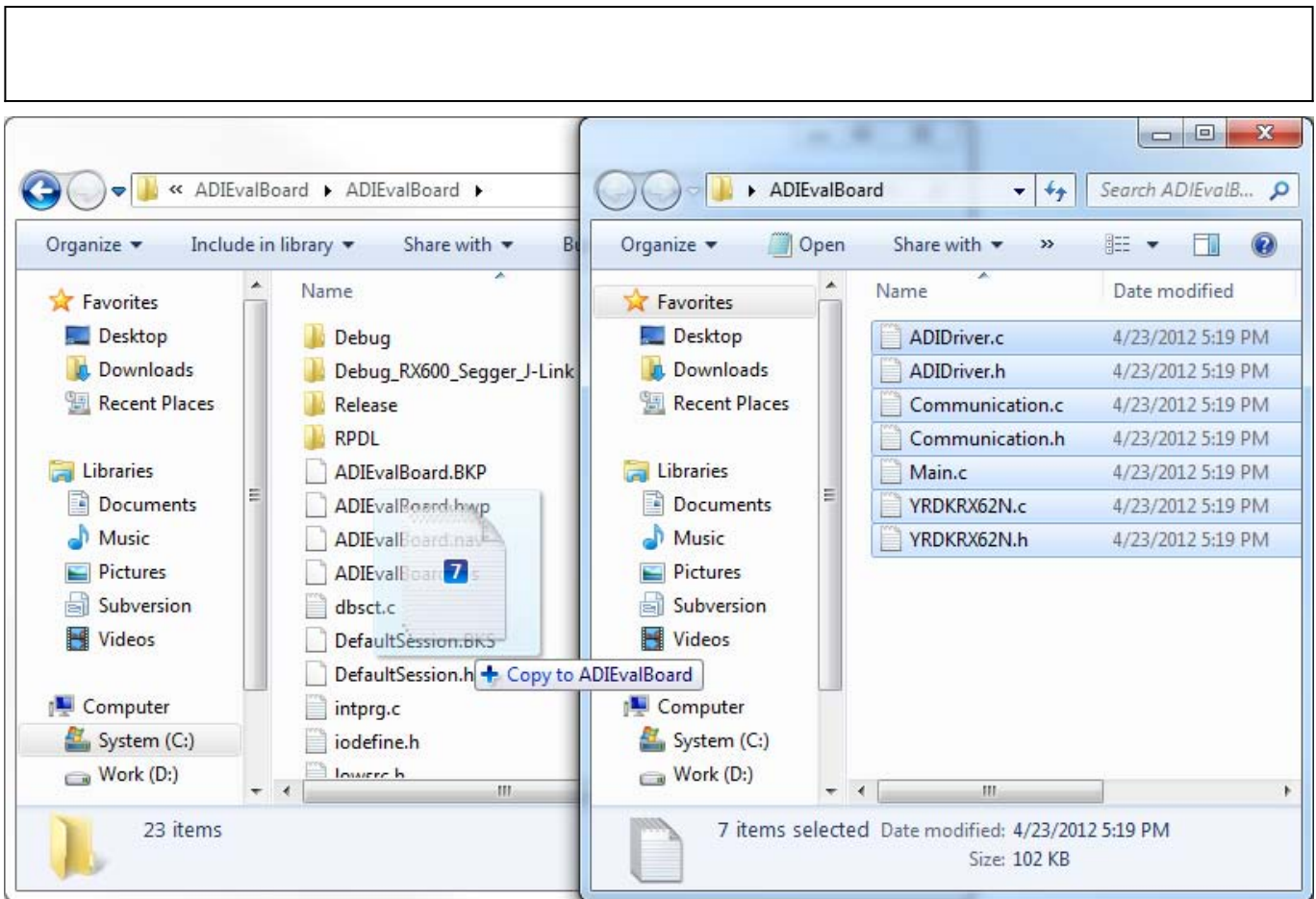
- The library file path has to be added in the project. Select the Link/Library tab, select "Show entries for: Library files" and press Add. Select "Relative to: Project directory", type "RPDL\RX62N_library" as file path and press OK.



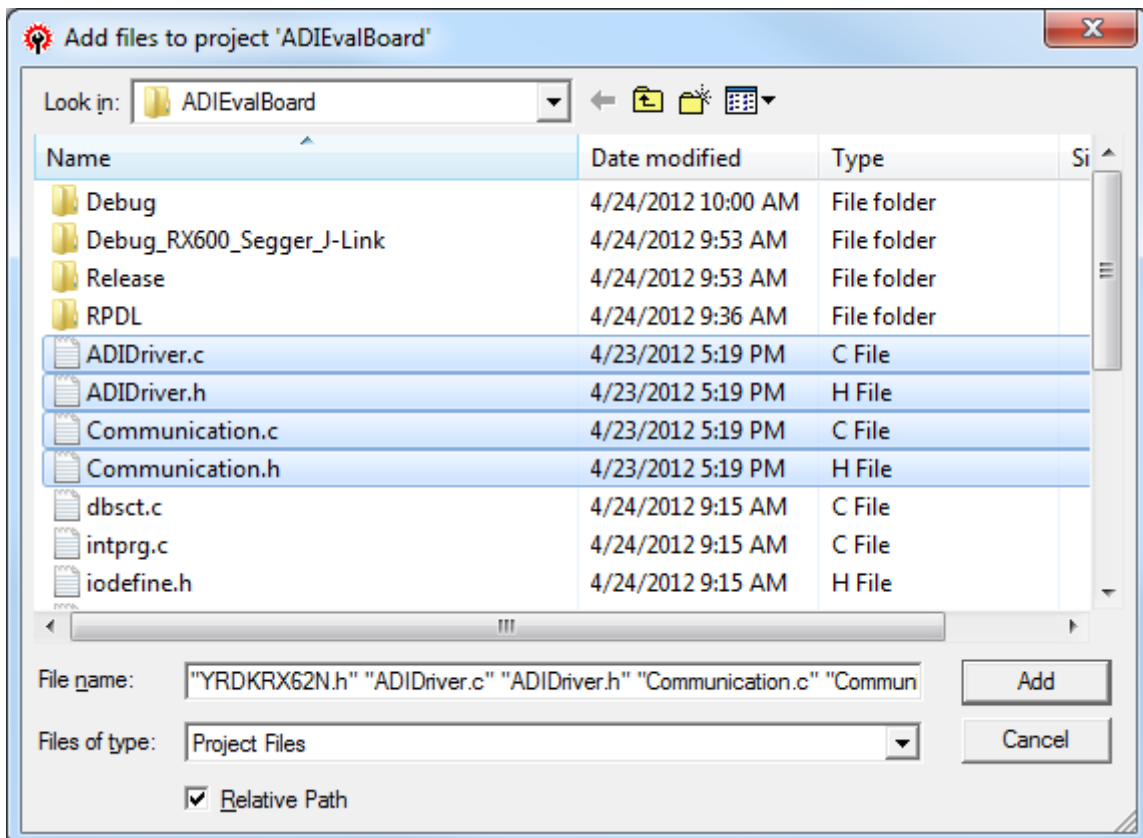
- Because the *“intprg.c”* file was removed the *“PIntPrg”* specified in option *“start”* has to be removed. Change *“Category”* to *“Section”*. Press *“Edit”*, select *“PIntPRG”* and press *“Remove”*. From this window the address of each section can be also modified. After all the changes are made press *OK* two times.



- At this point the files extracted from the zip file located in the "Software Tools" section have to be added into the project. Copy all the files from the archive into the project folder.



- Now, the files have to be included in the project. Use the key sequence *Alt, P, A* to open the “Add files to project ‘ADIEvalBoard’” window. Navigate into ADI folder. From the “Files of type” drop-down list, select “Project Files”. Select all the copied files and press *Add*.



- Now, the project is ready to be built. Press *F7*. The message after the Build Process is finished has to



be “0 Errors, 0 Warnings”. To run the program on the board, you have to download the firmware into the microprocessor’s memory.

03 Feb 2012 14:32 · [Dragos Bogdan](#)

More information

- [ask questions about the Microcontroller no-OS Drivers](#)
- Example questions:
 - [how to configure AD9361 for to obtain the carrier frequency in the transmission channel?](#) by vv_rustamov
 - [9361 ADC Capture Function Data Format](#) by kruse
 - [AD9361 Artix 7 API drivers](#) by Raveendra
 - [AD9361 Artix-7 drivers](#) by Raveendra
 - [AD9361 no-OS manual gain control](#) by pharbour

01 Jun 2012 11:21

© Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.



www.analog.com