AD7730L - Microcontroller No-OS Driver

Supported Devices

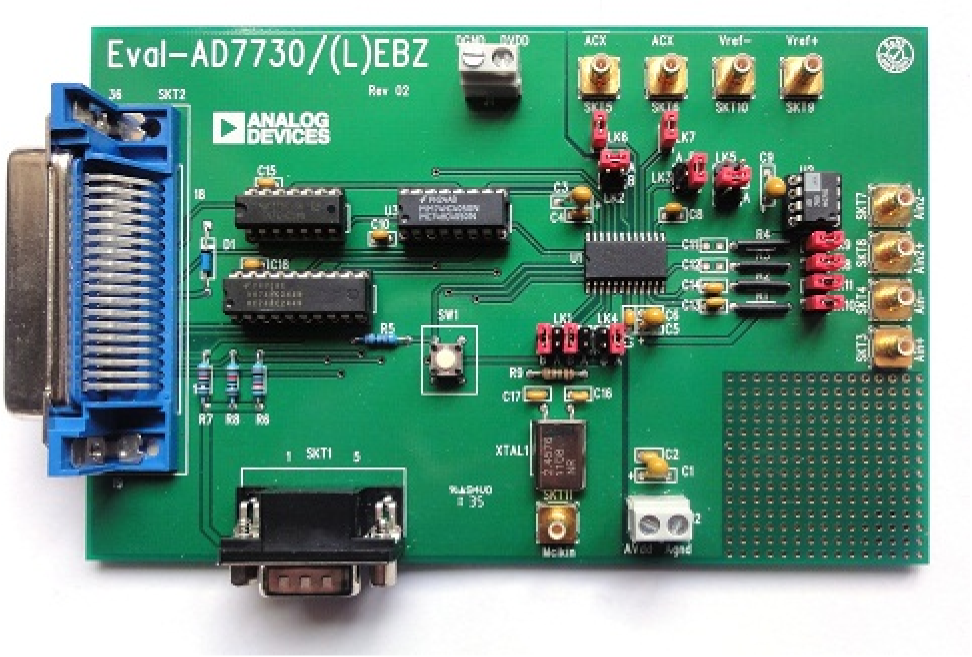
- AD7730L

Evaluation Boards

- EVAL-AD7730LEBZ

Overview

The AD7730L is a complete analog front end for weigh-scale and pressure measurement applications. The device accepts low-level signals directly from a transducer and outputs a serial digital word. The input signal is applied to a proprietary programmable gain front end based around an analog modulator. The modulator output is processed by a low pass programmable digital filter, allowing adjustment of filter cutoff, output rate and settling time.



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)
- Renesas Demo Kit for RX62N (Renesas)

Driver Description

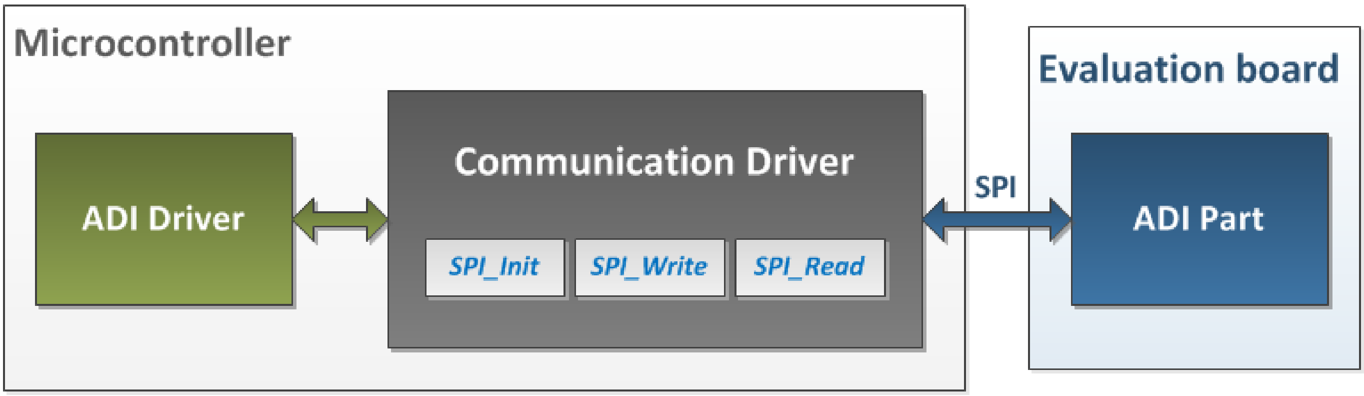
The driver contains two parts:

- The driver for the AD7730L 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 AD7730L driver can be used exactly as it is provided.

There are three functions which are called by the AD7730L driver:

- SPI_Init() – initializes the communication peripheral.
- SPI_Write() – writes data to the device.
- SPI_Read() – reads data from the device.



SPI driver architecture

The following functions are implemented in this version of AD7730L driver:

Function	Description
unsigned char AD7730L_Init(void)	Configures I/O pins from/towards the component and resets the device.
void AD7730L_SoftReset(void)	Resets the device.
void AD7730L_SetRegisterValue(unsigned char registerAddress, unsigned long registerValue, unsigned char bytesNumber)	Writes data into a register.
unsigned long AD7730L_GetRegisterValue(unsigned char registerAddress, unsigned char bytesNumber)	Reads the value of a register.
void AD7730L_WaitForReady(void)	Waits for Ready pin to go low.
void AD7730L_SetMode(unsigned short modeOperation, unsigned char inputRange, unsigned char channel)	Writes the specified mode of operation in Mode Register.
void AD7730L_SetFilterOutputRate(unsigned long outputRate, unsigned char chop, unsigned char skip)	Writes a value corresponding to an output rate in Filter Register.
void AD7730L_Calibrate(unsigned char channel, unsigned char inputRange, unsigned long outputRate, unsigned char dacValue)	Performs a calibration sequence for a channel.
unsigned long AD7730L_SingleRead(unsigned char channel, unsigned char inputRange);	Reads the result of a single conversion.
unsigned long AD7730L_ContinuousReadAvg(unsigned char channel, unsigned char inputRange, unsigned char sampleNumber);	Returns the average of several conversion results.

Downloads

- AD7730L Generic Driver
- AD7730L RL78G13 Driver
- AD7730L RX62N Driver

Renesas RL78G13 Quick Start Guide

This section contains a description of the steps required to run the AD7730L demonstration project on a Renesas RL78G13 platform.

Required Hardware

- Renesas Demo Kit for RL78G13 (Renesas)
- EVAL-AD7730LEBZ(ADI)

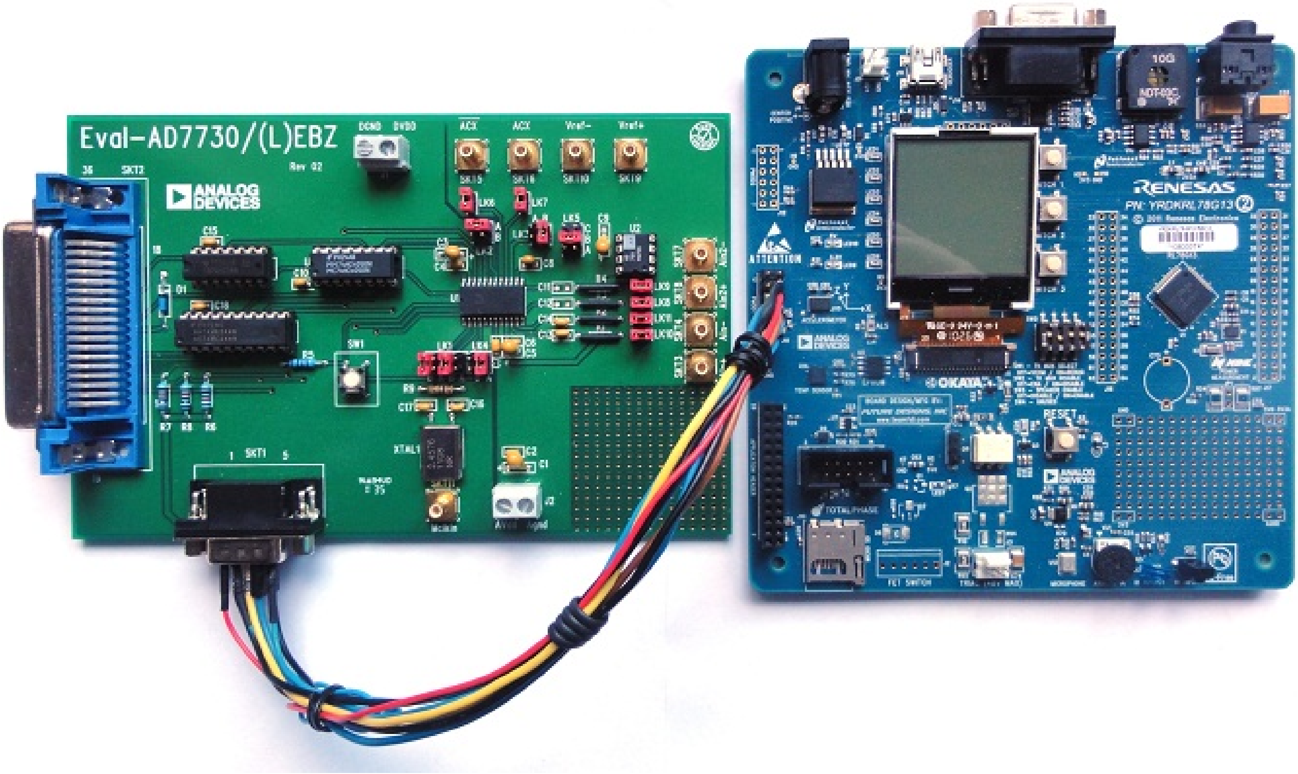
Required Software

- IAR Embedded Workbench for Renesas RL78 Kickstart
- Applilet3 for RL78G13

Hardware Setup

An EVAL-AD7730LEBZ has to be interfaced with the Renesas Demonstration Kit (RDK) for RL78G13:

```
  EVAL-AD7730LEBZ SKT1 connector Pin 3 (CS)    → YRDKRL78G13 J11 connector Pin 1
  EVAL-AD7730LEBZ SKT1 connector Pin 5 (DIN)   → YRDKRL78G13 J11 connector Pin 2
  EVAL-AD7730LEBZ SKT1 connector Pin 7 (DOUT)  → YRDKRL78G13 J11 connector Pin 3
  EVAL-AD7730LEBZ SKT1 connector Pin 1 (SCLK)  → YRDKRL78G13 J11 connector Pin 4
  EVAL-AD7730LEBZ SKT1 connector Pin 6 (DGND)  → YRDKRL78G13 J11 connector Pin 5
  EVAL-AD7730LEBZ SKT1 connector Pin 2 (RDY)   → YRDKRL78G13 J11 connector Pin 7
  EVAL-AD7730LEBZ SKT1 connector Pin 4 (RESET) → YRDKRL78G13 J11 connector Pin 9
  EVAL-AD7730LEBZ SKT1 connector Pin 9 (SYNC)  → YRDKRL78G13 J11 connector Pin 10
```



Software Setup

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

CSI10 (Clocked Serial Interface 10) – For the AD7730L part and 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)*

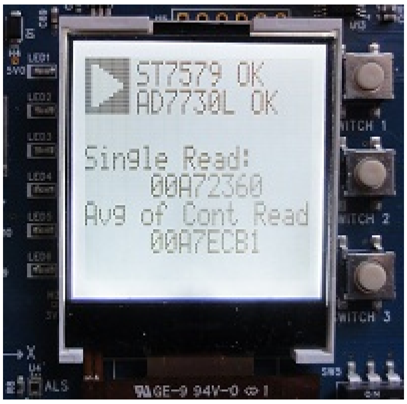
Watchdog Timer

Disable the watchdog timer:

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

Reference Project Overview

The reference project initializes the device, calibrates it and performs single and continuous reads of an analog signal. The results are displayed on the LCD.



```
ST7579 OK
AD7730L OK

Single Read:
    00A72360
Avg of Cont Read
    00A7ECB1
```

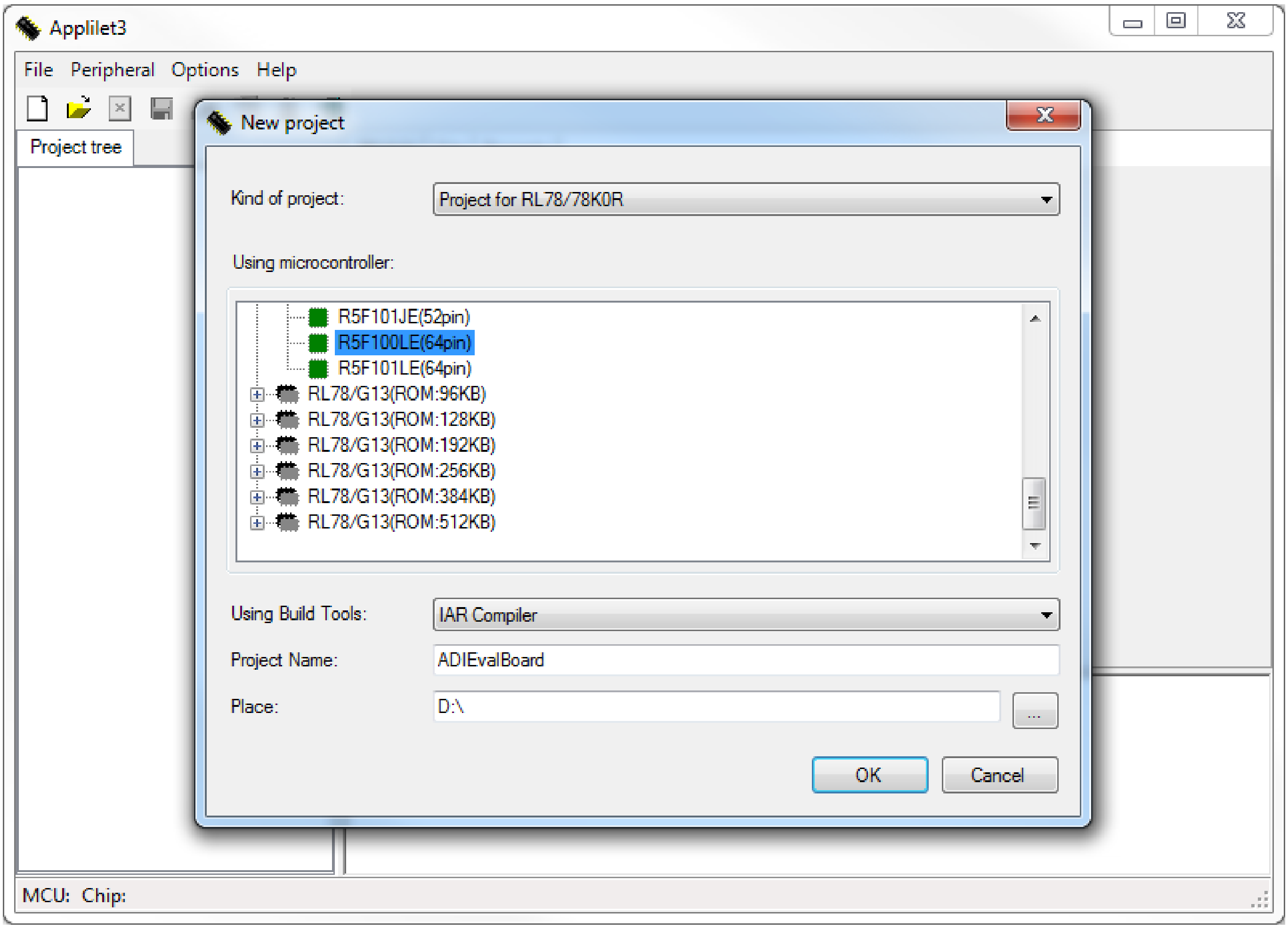
Software Project Setup

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***.



Kind of project: Project for RL78/78K0R

Using microcontroller:

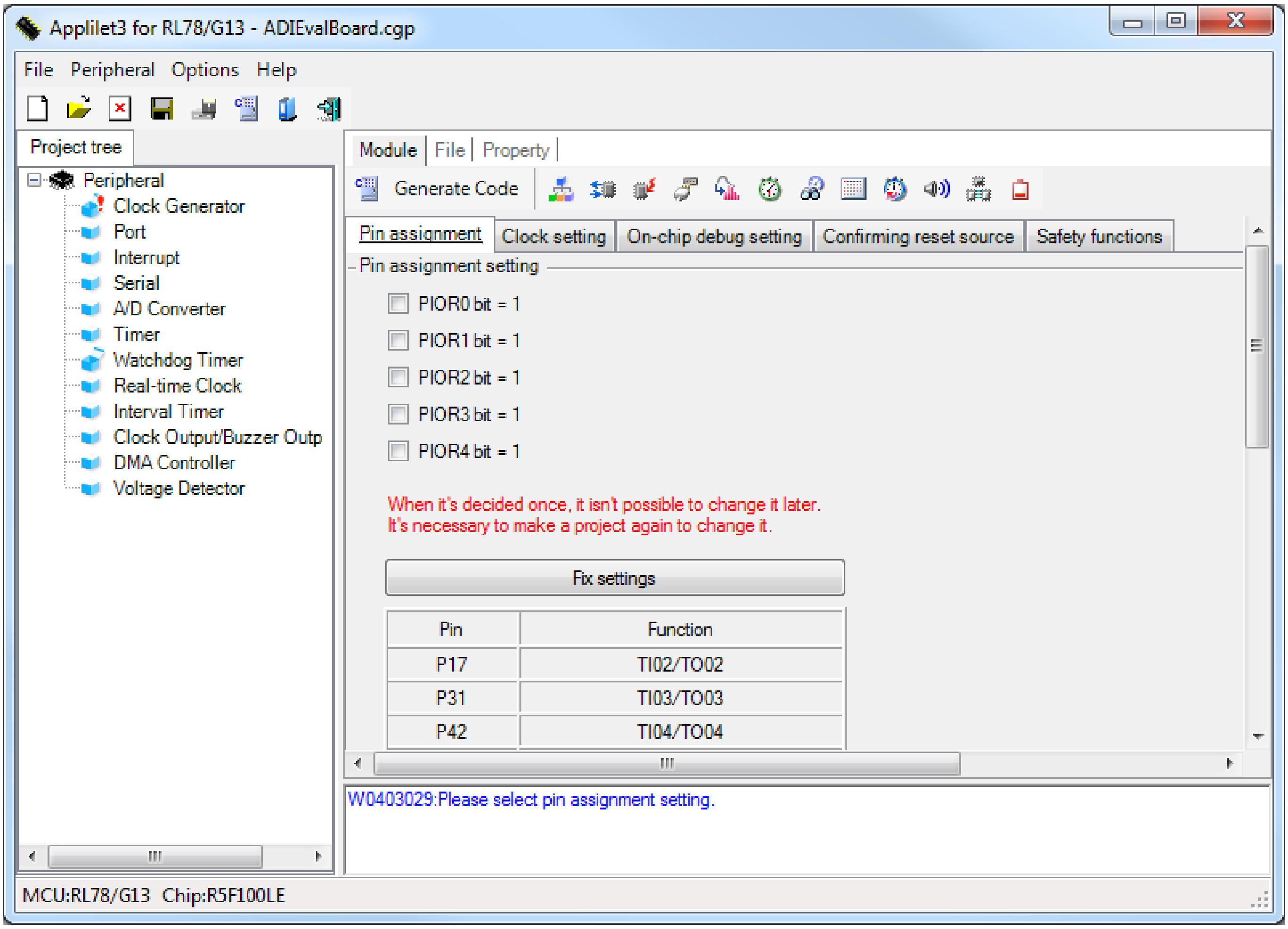
- R5F101JE(52pin)
- R5F100LE(64pin)
- R5F101LE(64pin)
- RL78/G13(ROM:96KB)
- RL78/G13(ROM:128KB)
- RL78/G13(ROM:192KB)
- RL78/G13(ROM:256KB)
- RL78/G13(ROM:384KB)
- RL78/G13(ROM:512KB)

Using Build Tools: IAR Compiler

Project Name: ADIEvalBoard

Place: D:\

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



Applilet3 for RL78/G13 - ADIEvalBoard.cgp

Pin assignment setting

□ PIOR0 bit = 1

□ PIOR1 bit = 1

□ PIOR2 bit = 1

□ PIOR3 bit = 1

□ PIOR4 bit = 1

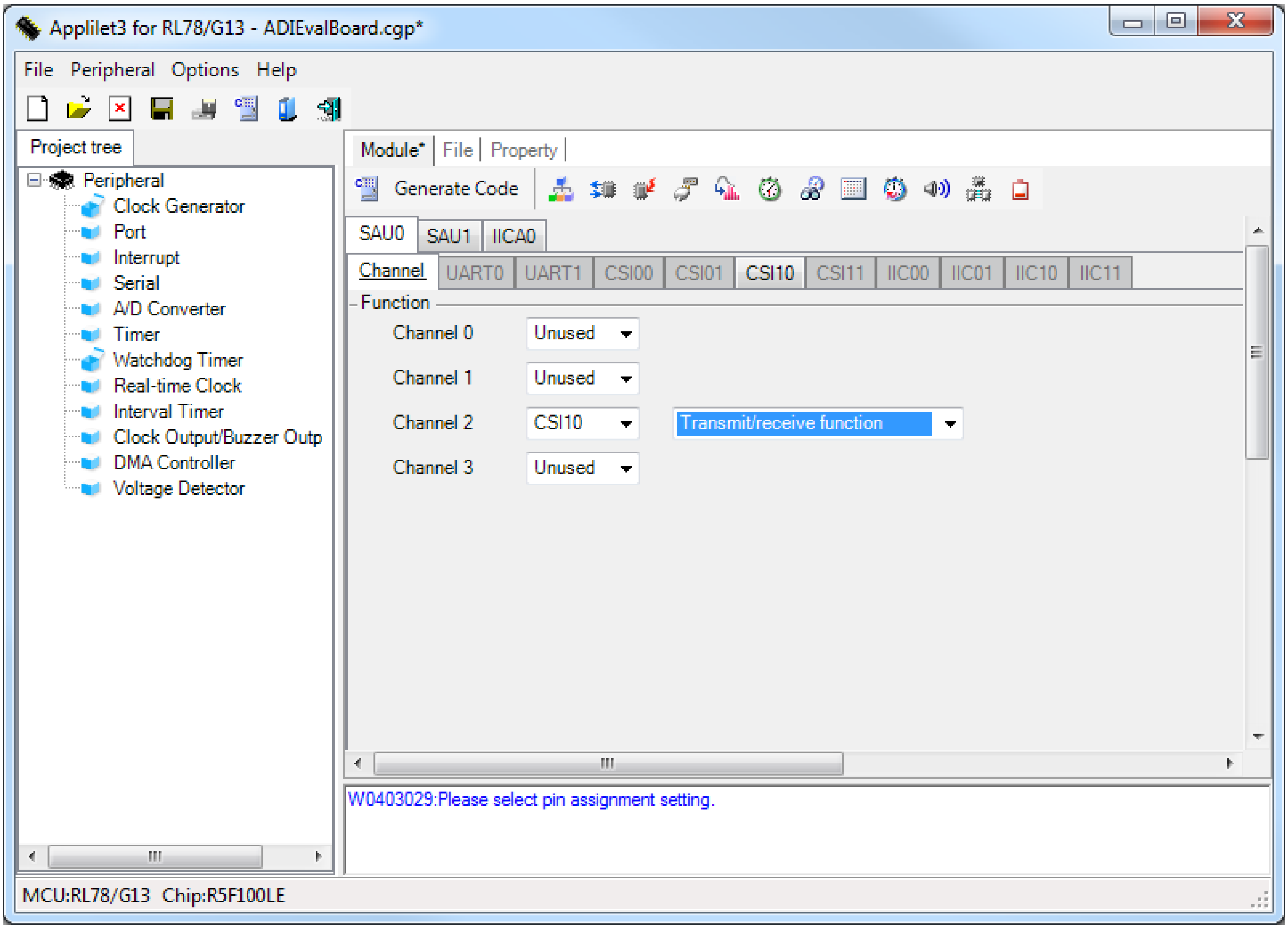
When it's decided once, it isn't possible to change it later.
It's necessary to make a project again to change it.

Pin	Function
P17	TI02/TO02
P31	TI03/TO03
P42	TI04/TO04

W0403029:Please select pin assignment setting.

MCU:RL78/G13 Chip:R5F100LE

- 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.



Channel 0 Unused

Channel 1 Unused

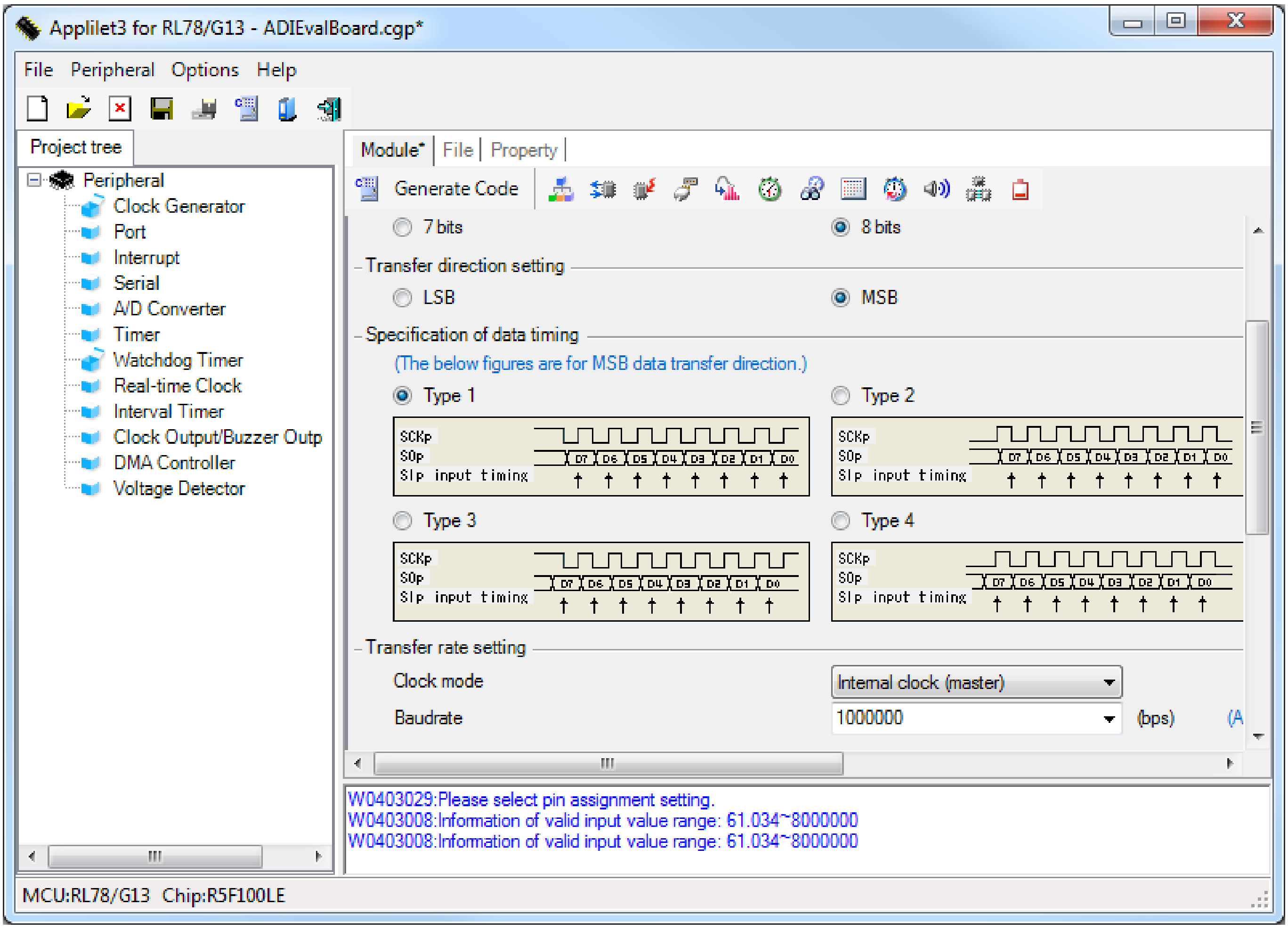
Channel 2 CSI10 Transmit/receive function

Channel 3 Unused

W0403029:Please select pin assignment setting.

MCU:RL78/G13 Chip:R5F100LE

- To configure the ***CSI10*** 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.



○ 7 bits ● 8 bits

Transfer direction setting

○ LSB ● MSB

Specification of data timing

(The below figures are for MSB data transfer direction.)

● Type 1 ○ Type 2

○ Type 3 ○ Type 4

Transfer rate setting

Clock mode: Internal clock (master)

Baudrate: 1000000 (bps)

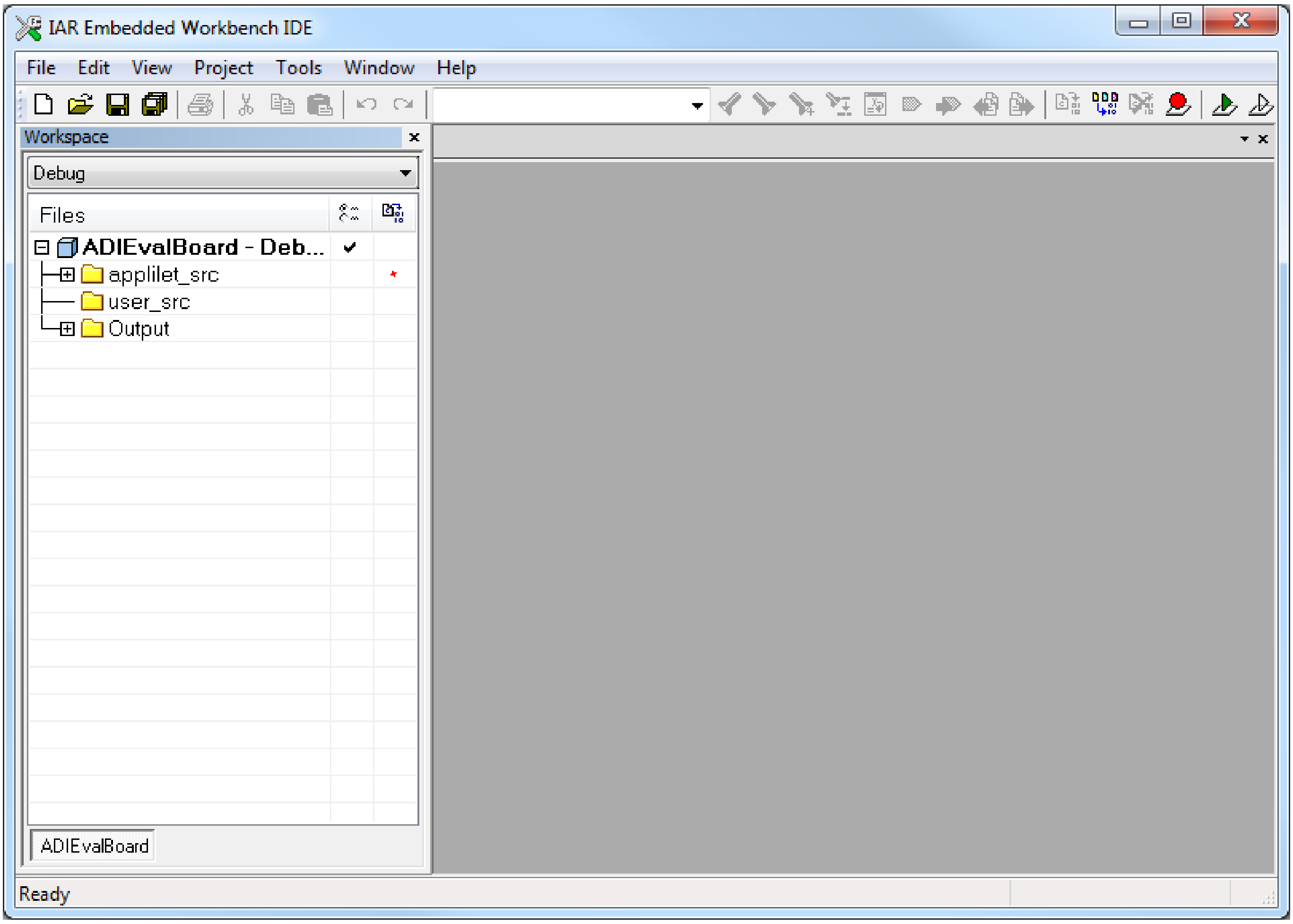
```
W0403029:Please select pin assignment setting.
W0403008:Information of valid input value range: 61.034~8000000
W0403008:Information of valid input value range: 61.034~8000000
```

MCU:RL78/G13 Chip:R5F100LE

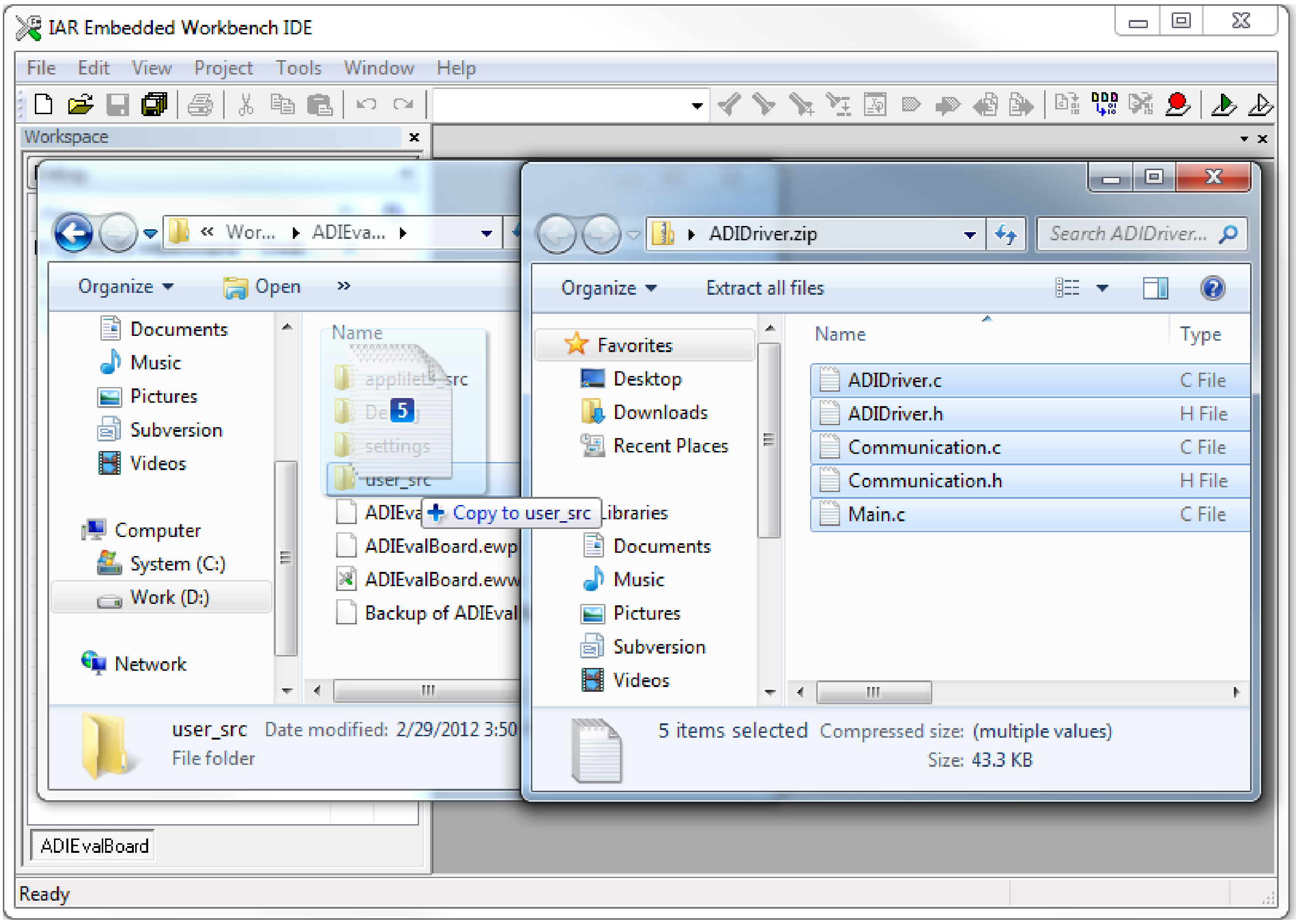
- After all the desired peripherals are configured click on the ***Generate Code*** button and a new workspace and a new project for the ***IAR Embedded Workbench*** will be generated. After the code was generated close the ***Applilet3 for RL78G13*** tool.

Step 2 - IAR Embedded Workbench for Renesas RL78

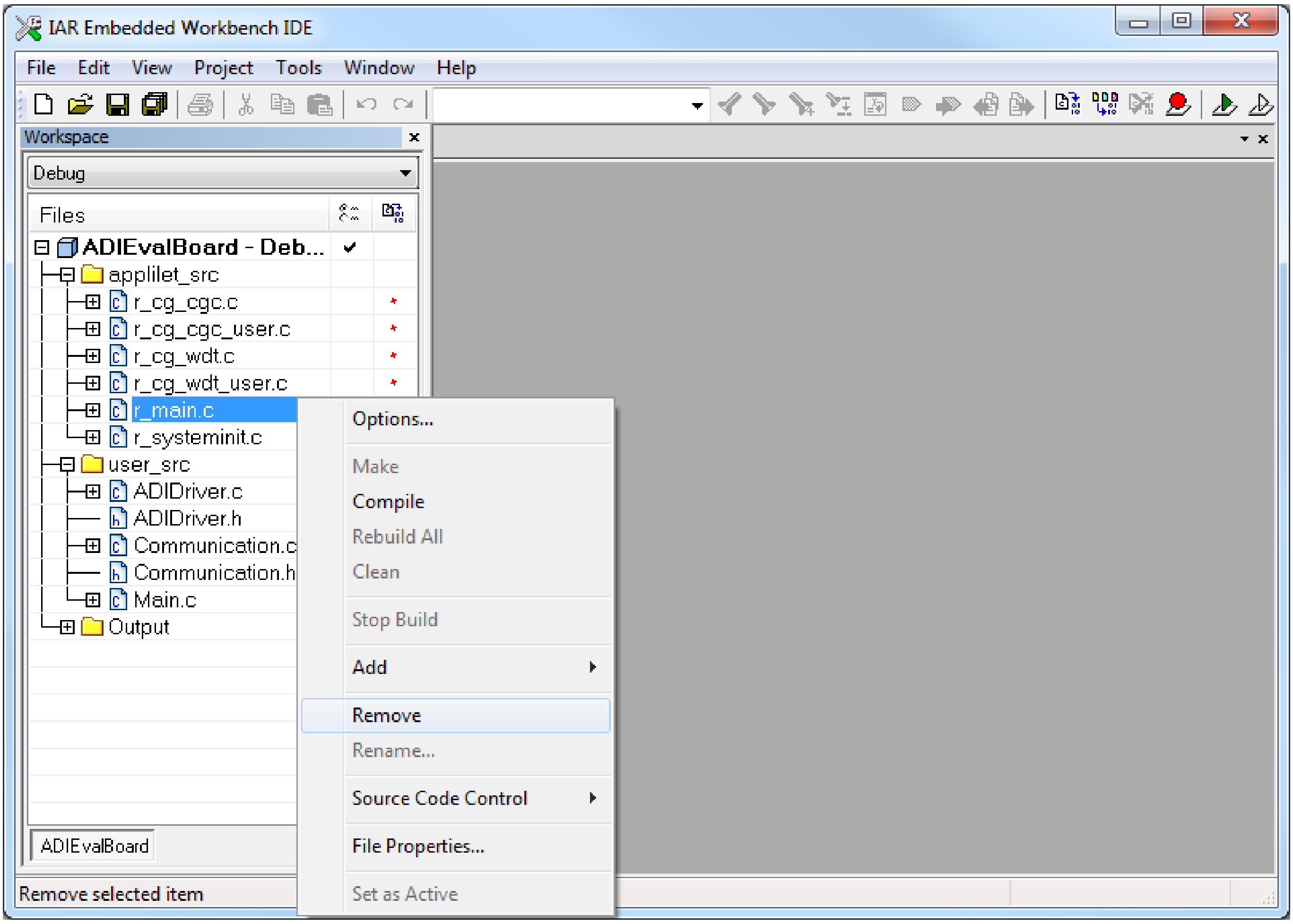
- Run the ***IAR Embedded Workbench*** and open the workspace created with the ***Applilet3*** tool.



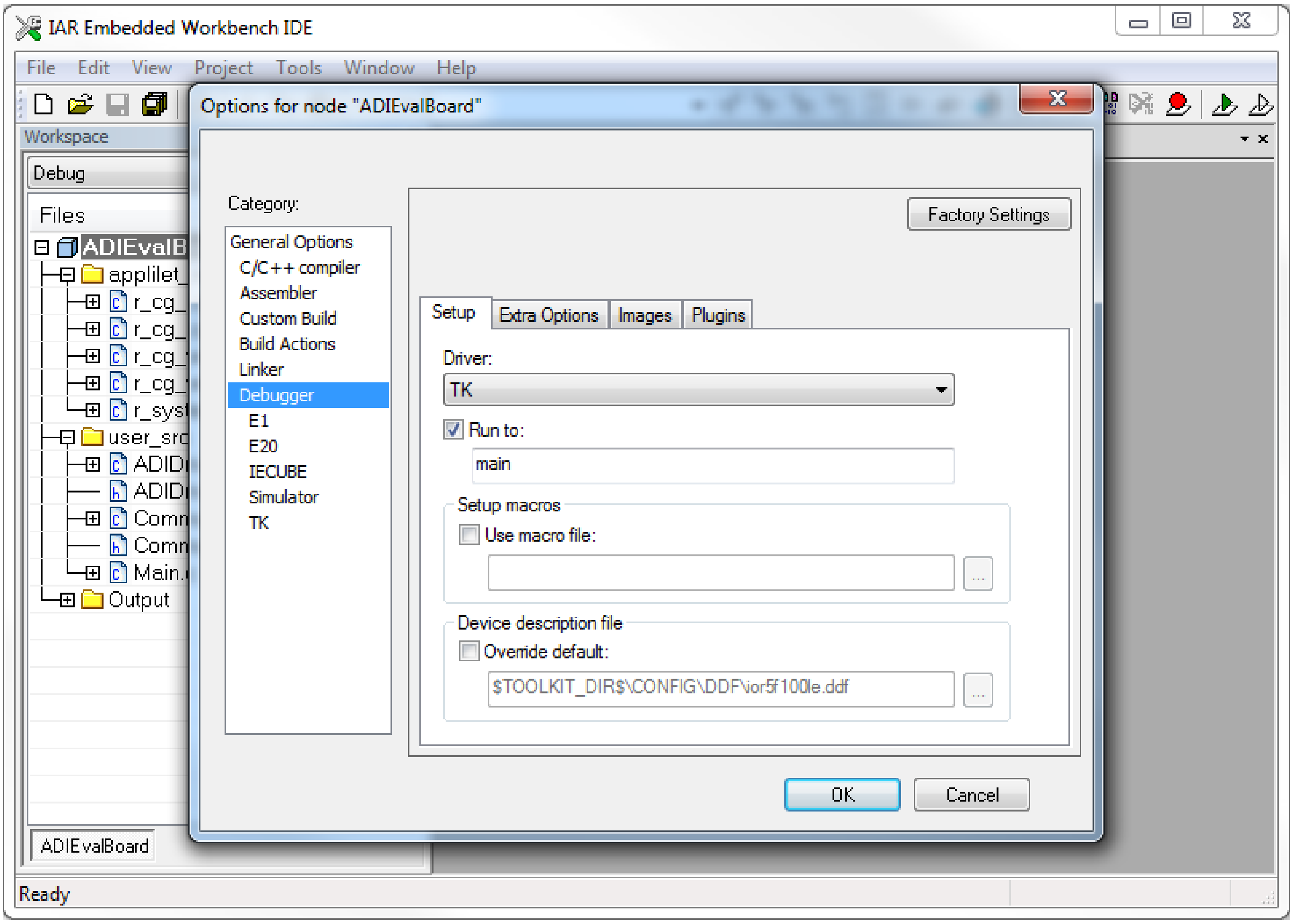
- Copy the files extracted from the zip file into the ***user_src*** folder, located in the project’s folder.



- 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 ***applilet_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.



Options for node "ADIEvalBoard"

Category:

Driver: TK

☑ Run to: main

Setup macros

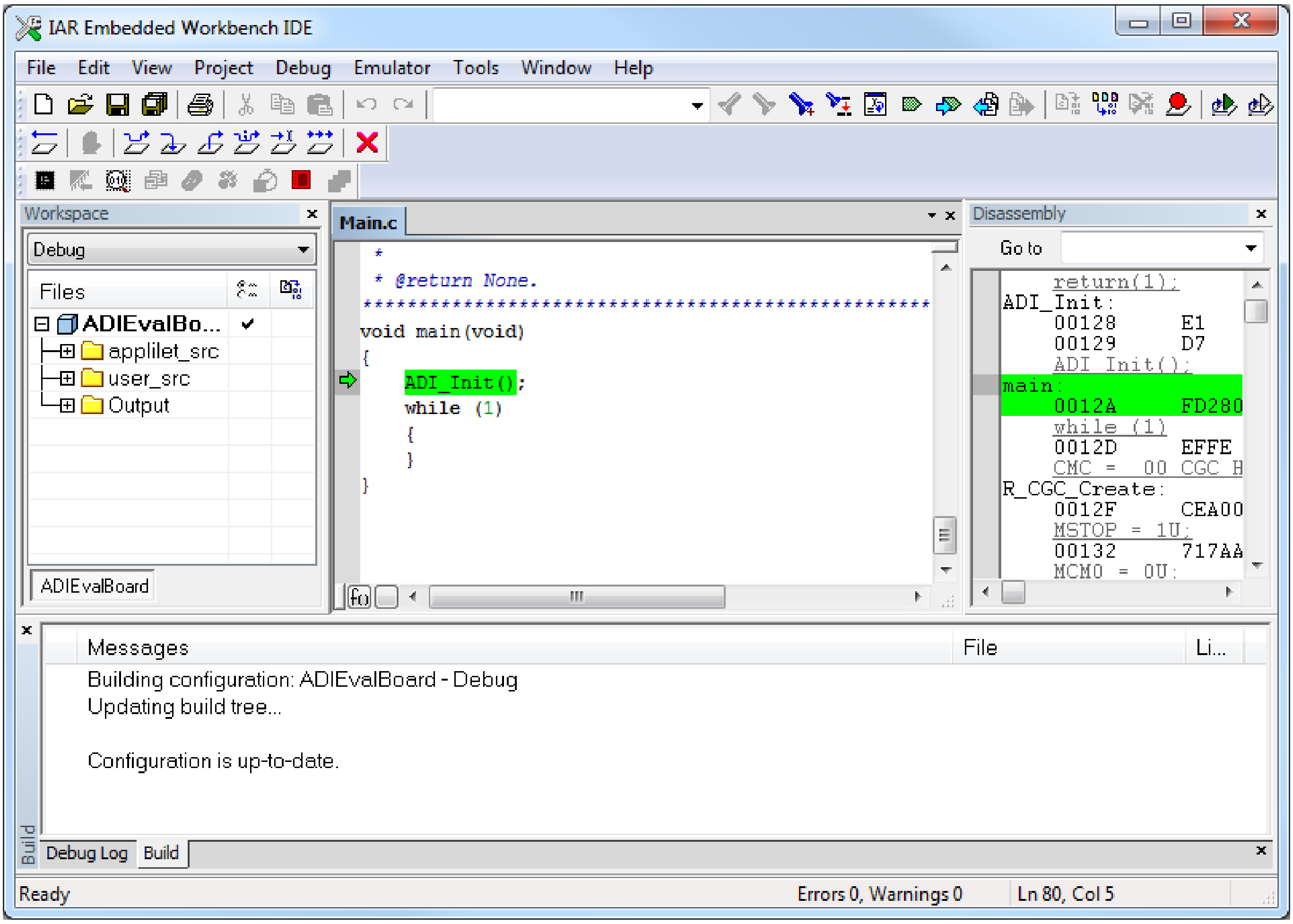
□ Use macro file:

Device description file

□ Override default:

$TOOLKIT_DIR$\CONFIG\DDF\ior5f100le.ddf

- 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.



```
 *
 * @return None.
*******************************************************
void main(void)
{
    ADI_Init();
    while (1)
    {
    }
}
```

```
    return(1);
ADI_Init:
    00128       E1
    00129       D7
    ADI_Init();
main:
    0012A       FD280
    while (1)
    0012D       EFFE
    CMC =  00 CGC_H
R_CGC_Create:
    0012F       CEA00
    MSTOP = 1U;
    00132       717AA
    MCM0 = 0U;
```

```
Building configuration: ADIEvalBoard - Debug
Updating build tree...

Configuration is up-to-date.
```

```
Ready                    Errors 0, Warnings 0    Ln 80, Col 5
```

29 Feb 2012 16:01 · Dragos Bogdan

Renesas RX62N Quick Start Guide

This section contains a description of the steps required to run the AD7730 demonstration project on a Renesas RX62N platform.

Required Hardware

- Renesas Demo Kit for RX62N (Renesas)
- EVAL-AD7730LEBZ(ADI)

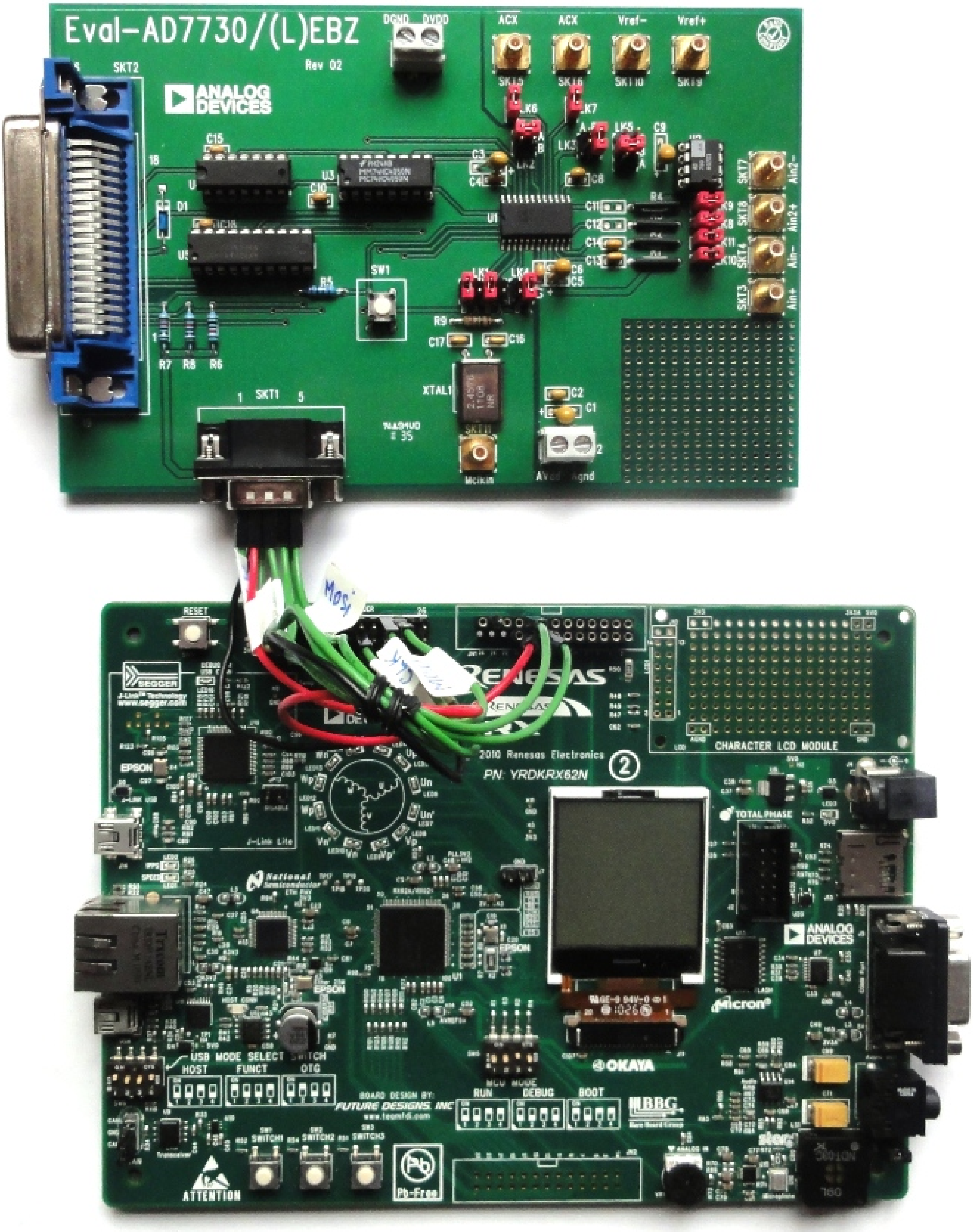
Required Software

- High-performance Embedded Workshop for RX62N family

Hardware Setup

An EVAL-AD7730LEBZ board has to be interfaced with the Renesas Demonstration Kit (RDK) for RX62N:

```
  EVAL-AD7730LEBZ SKT1 connector Pin 3 (CS)    → YRDKRX62N J8 connector Pin 15
  EVAL-AD7730LEBZ SKT1 connector Pin 5 (DIN)   → YRDKRX62N J8 connector Pin 19
  EVAL-AD7730LEBZ SKT1 connector Pin 7 (DOUT)  → YRDKRX62N J8 connector Pin 22
  EVAL-AD7730LEBZ SKT1 connector Pin 1 (SCLK)  → YRDKRX62N J8 connector Pin 20
  EVAL-AD7730LEBZ SKT1 connector Pin 6 (DGND)  → YRDKRX62N J8 connector Pin 4
  EVAL-AD7730LEBZ SKT1 connector Pin 2 (RDY)   → YRDKRX62N JN1 connector Pin 18
  EVAL-AD7730LEBZ SKT1 connector Pin 4 (RESET) → YRDKRX62N JN1 connector Pin 16
  EVAL-AD7730LEBZ SKT1 connector Pin 9 (SYNC)  → YRDKRX62N JN1 connector Pin 17
```



Reference Project Overview

The reference project initializes the device, calibrates it and performs single and continuous reads of an analog signal. The results are displayed on the LCD.

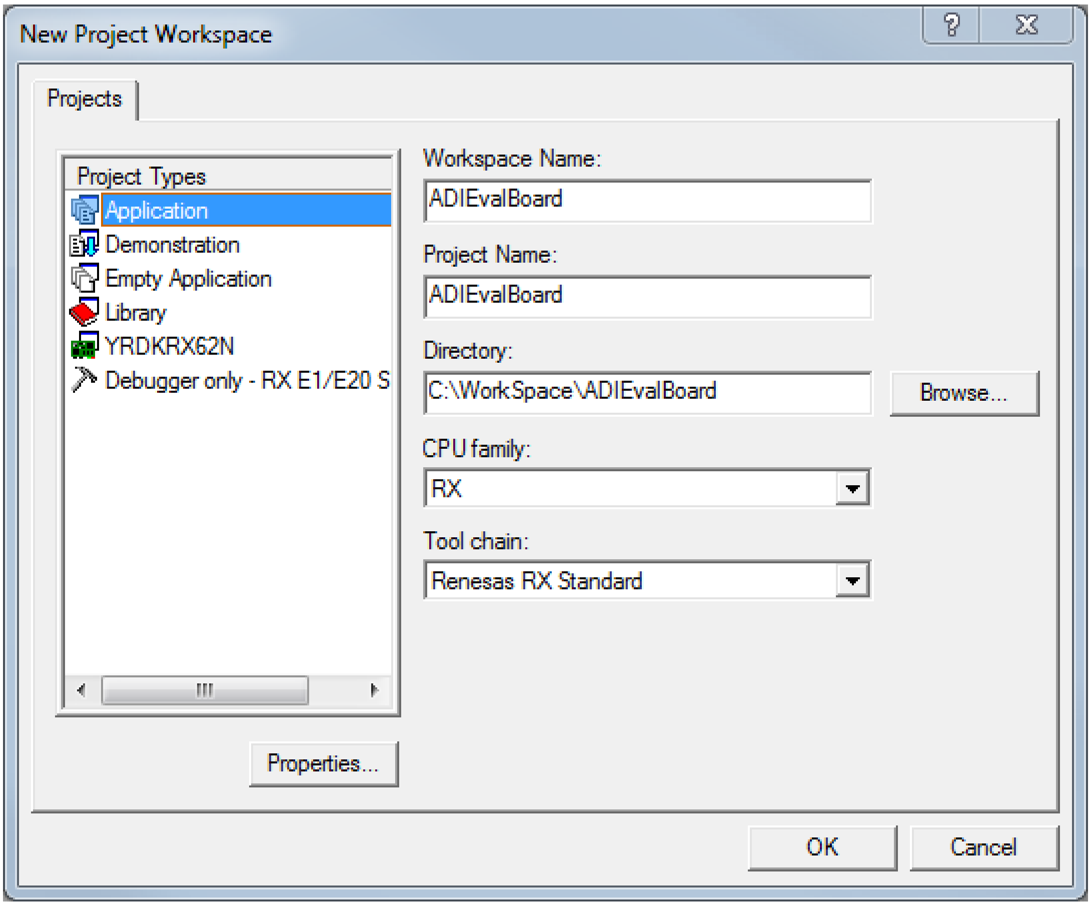
```
ST7579 OK
AD7730L OK

Single Read
    00B31B69
Avg of Cont Read
    00B30B89
```

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*.



Workspace Name: ADIEvalBoard

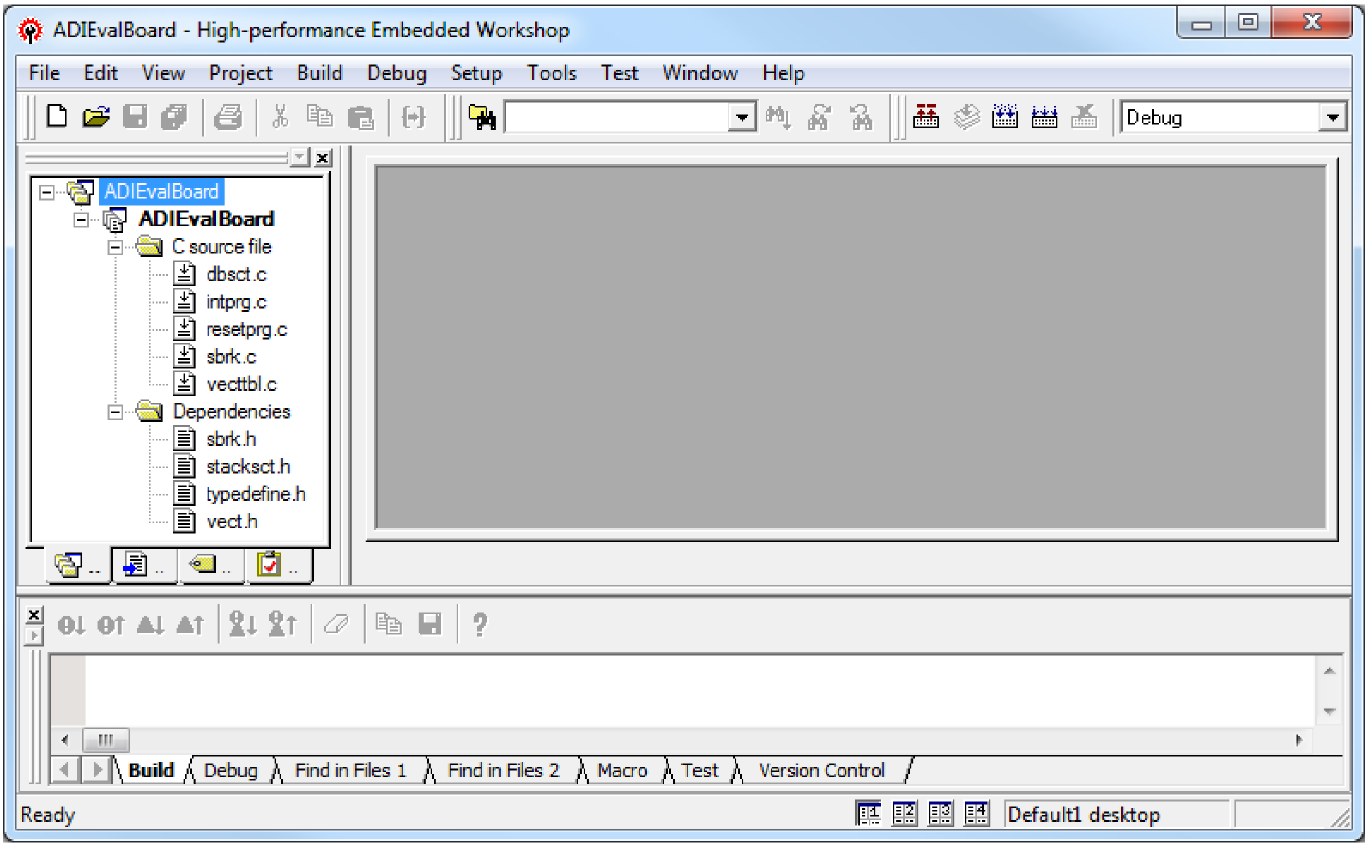
Project Name: ADIEvalBoard

Directory: C:\WorkSpace\ADIEvalBoard

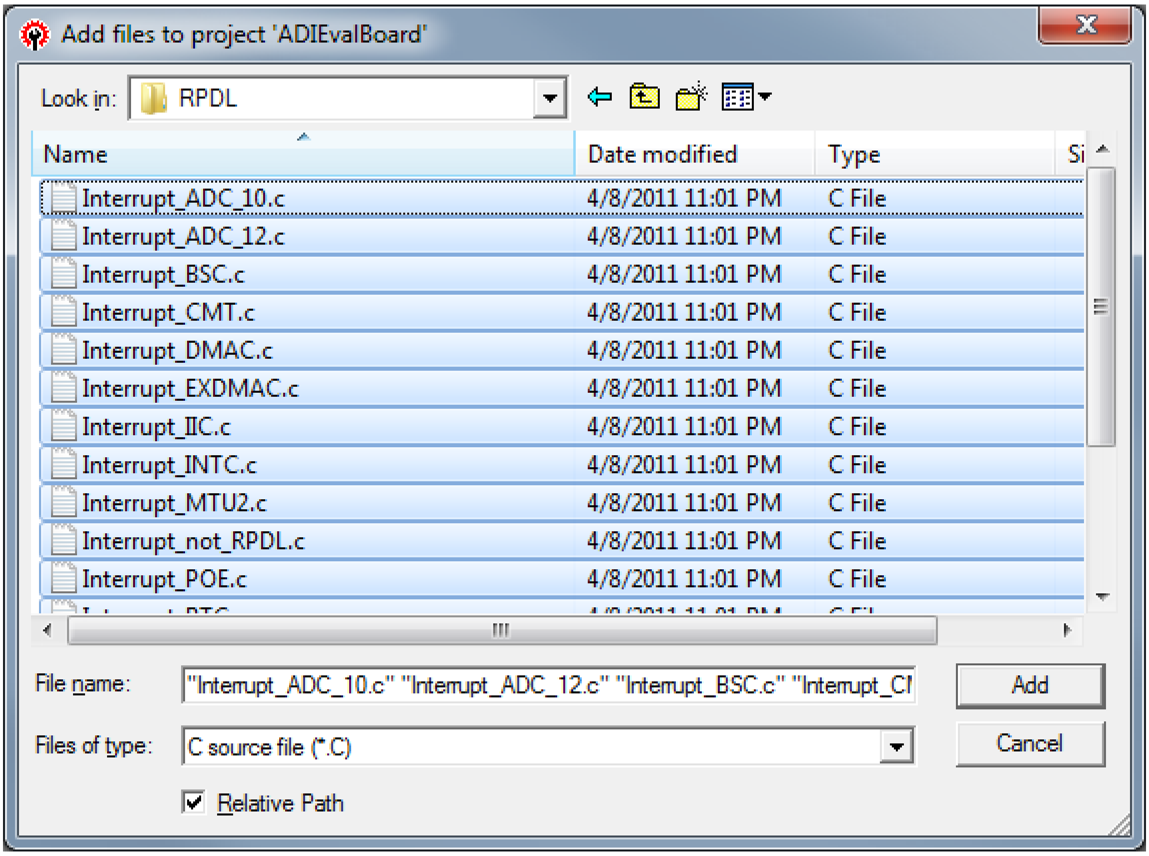
CPU family: RX

Tool chain: Renesas RX Standard

- 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.


ADIEvalBoard - High-performance Embedded Workshop
File Edit View Project Build Debug Setup Tools Test Window Help
Debug
ADIEvalBoard
ADIEvalBoard
C source file
dbsct.c
intprg.c
resetprg.c
sbrk.c
vecttbl.c
Dependencies
sbrk.h
stacksct.h
typedefine.h
vect.h
Build Debug Find in Files 1 Find in Files 2 Macro Test Version Control
Ready
Default1 desktop

- The RPDL (Renesas Peripheral Driver Library) has to integrated in the project. Unzip the RPDL files (double-click on the file “*RPDL_RX62N.exe*”). Navigate to where the RPDL files were unpacked and double-click on the “*Copy_RPDL_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 RPDL folder. From the “*Files of type*” drop-down list, select “*C source file (*.C)*”. Select all of the files and press *Add*.



Add files to project 'ADIEvalBoard'

Look in: RPDL

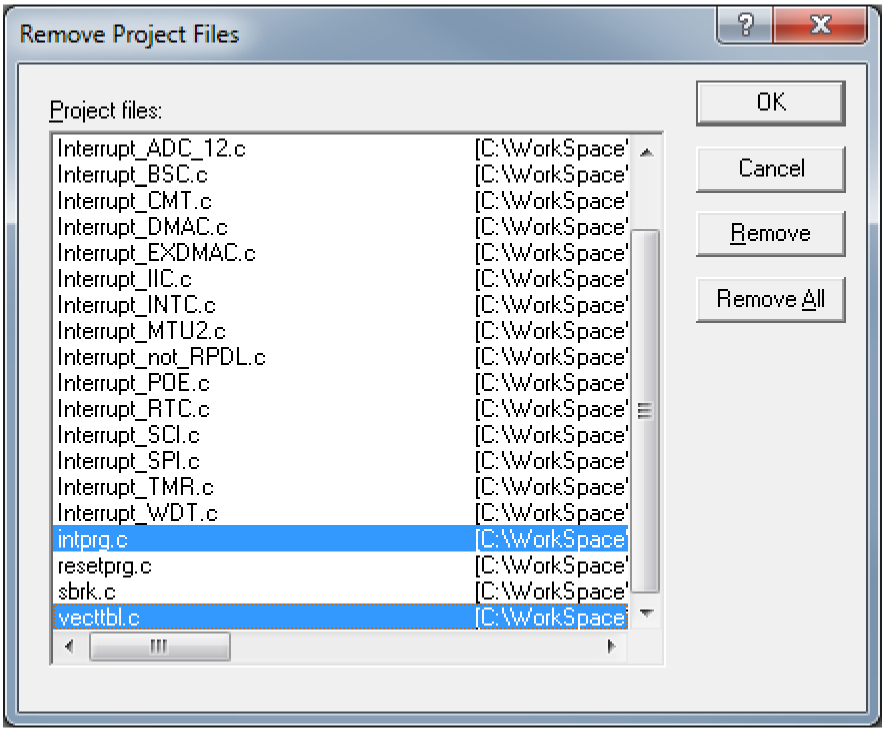
Name	Date modified	Type
Interrupt_ADC_10.c	4/8/2011 11:01 PM	C File
Interrupt_ADC_12.c	4/8/2011 11:01 PM	C File
Interrupt_BSC.c	4/8/2011 11:01 PM	C File
Interrupt_CMT.c	4/8/2011 11:01 PM	C File
Interrupt_DMAC.c	4/8/2011 11:01 PM	C File
Interrupt_EXDMAC.c	4/8/2011 11:01 PM	C File
Interrupt_IIC.c	4/8/2011 11:01 PM	C File
Interrupt_INTC.c	4/8/2011 11:01 PM	C File
Interrupt_MTU2.c	4/8/2011 11:01 PM	C File
Interrupt_not_RPDL.c	4/8/2011 11:01 PM	C File
Interrupt_POE.c	4/8/2011 11:01 PM	C File

File name: "Interrupt_ADC_10.c" "Interrupt_ADC_12.c" "Interrupt_BSC.c" "Interrupt_CI

Files of type: C source file (*.C)

☑ Relative Path

- 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*.

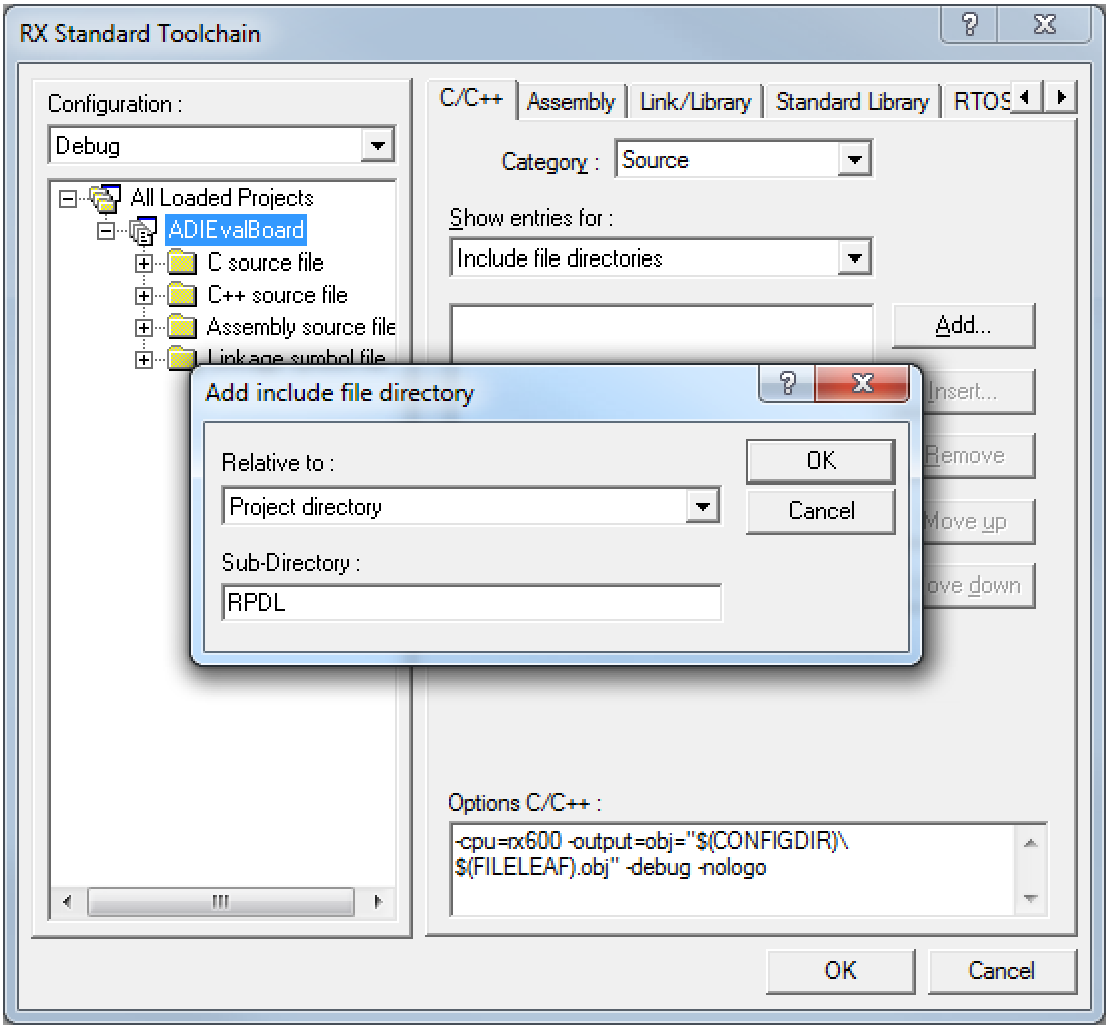


Remove Project Files

Project files:

```
Interrupt_ADC_12.c      [C:\WorkSpace'
Interrupt_BSC.c         [C:\WorkSpace'
Interrupt_CMT.c         [C:\WorkSpace'
Interrupt_DMAC.c        [C:\WorkSpace'
Interrupt_EXDMAC.c      [C:\WorkSpace'
Interrupt_IIC.c         [C:\WorkSpace'
Interrupt_INTC.c        [C:\WorkSpace'
Interrupt_MTU2.c        [C:\WorkSpace'
Interrupt_not_RPDL.c    [C:\WorkSpace'
Interrupt_POE.c         [C:\WorkSpace'
Interrupt_RTC.c         [C:\WorkSpace'
Interrupt_SCI.c         [C:\WorkSpace'
Interrupt_SPI.c         [C:\WorkSpace'
Interrupt_TMR.c         [C:\WorkSpace'
Interrupt_WDT.c         [C:\WorkSpace'
intprg.c                [C:\WorkSpace'
resetprg.c              [C:\WorkSpace'
sbrk.c                  [C:\WorkSpace'
vecttbl.c               [C:\WorkSpace'
```

- 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*.



RX Standard Toolchain

Configuration : Debug

All Loaded Projects
ADIEvalBoard
C source file
C++ source file
Assembly source file
Linkage symbol file

C/C++ | Assembly | Link/Library | Standard Library | RTOS

Category : Source

Show entries for : Include file directories

Add include file directory

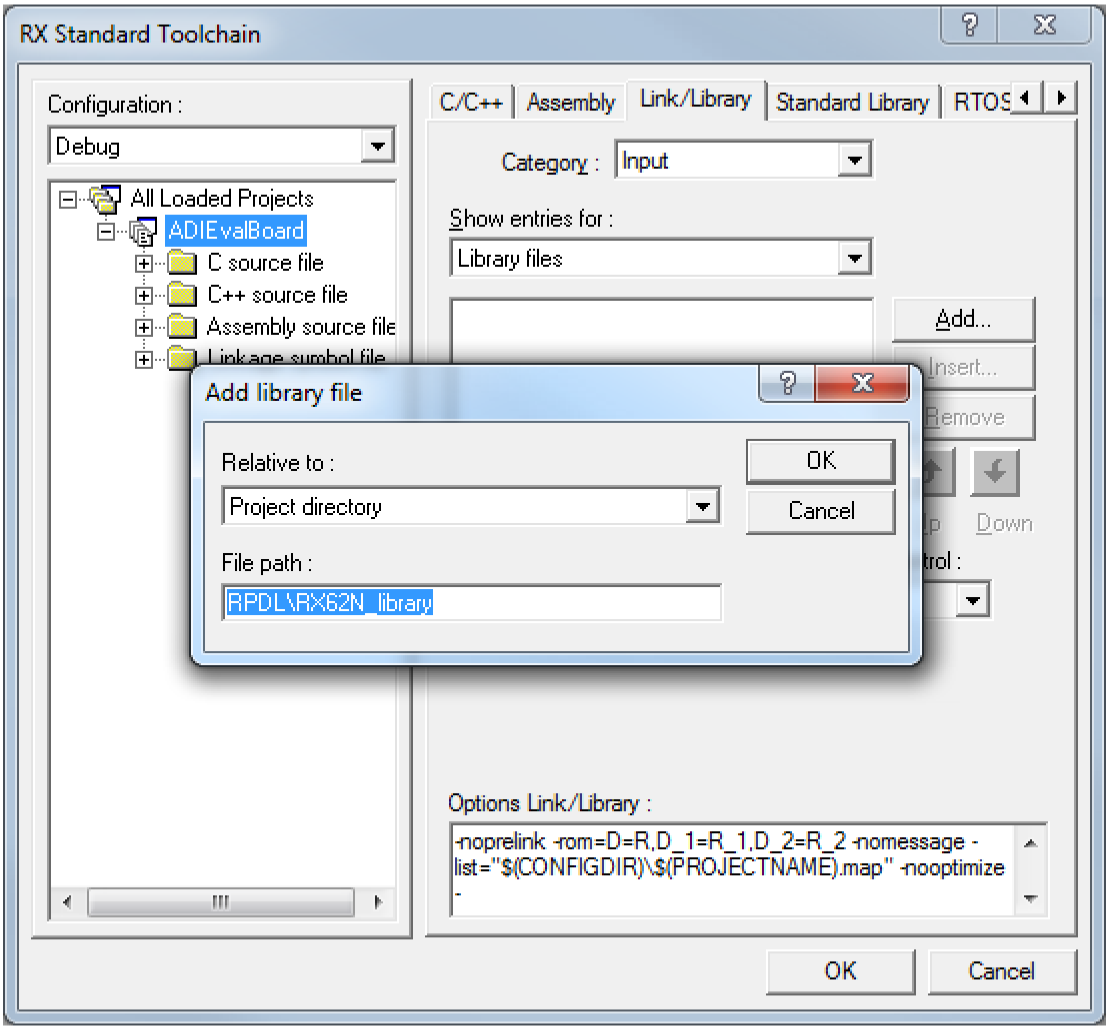
Relative to : Project directory

Sub-Directory : RPDL

Options C/C++ :

```
-cpu=rx600 -output=obj="$(CONFIGDIR)\
$(FILELEAF).obj" -debug -nologo
```

- 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*.



RX Standard Toolchain

Configuration : Debug

All Loaded Projects

ADIEvalBoard

Category : Input

Show entries for : Library files

Add library file

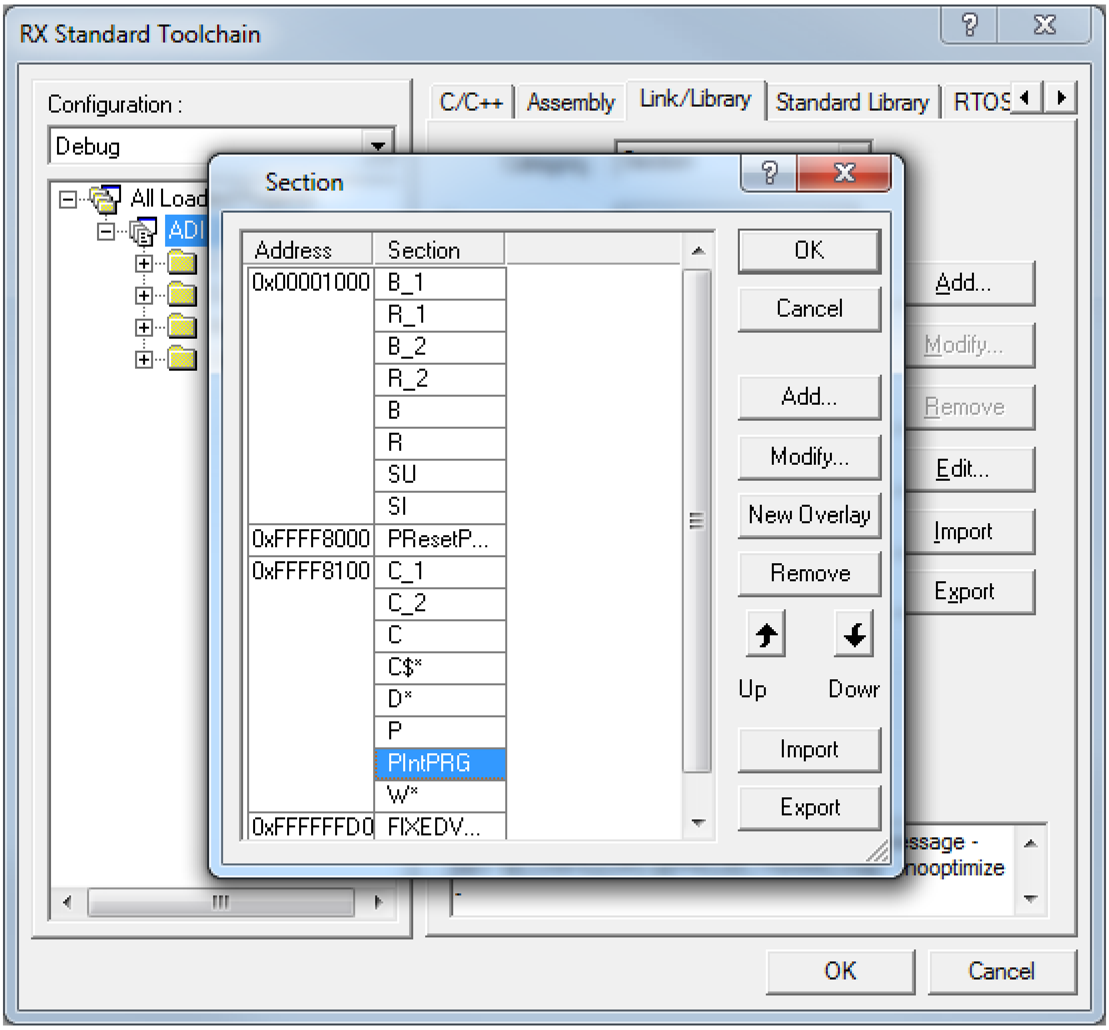
Relative to : Project directory

File path : RPDL\RX62N_library

Options Link/Library :

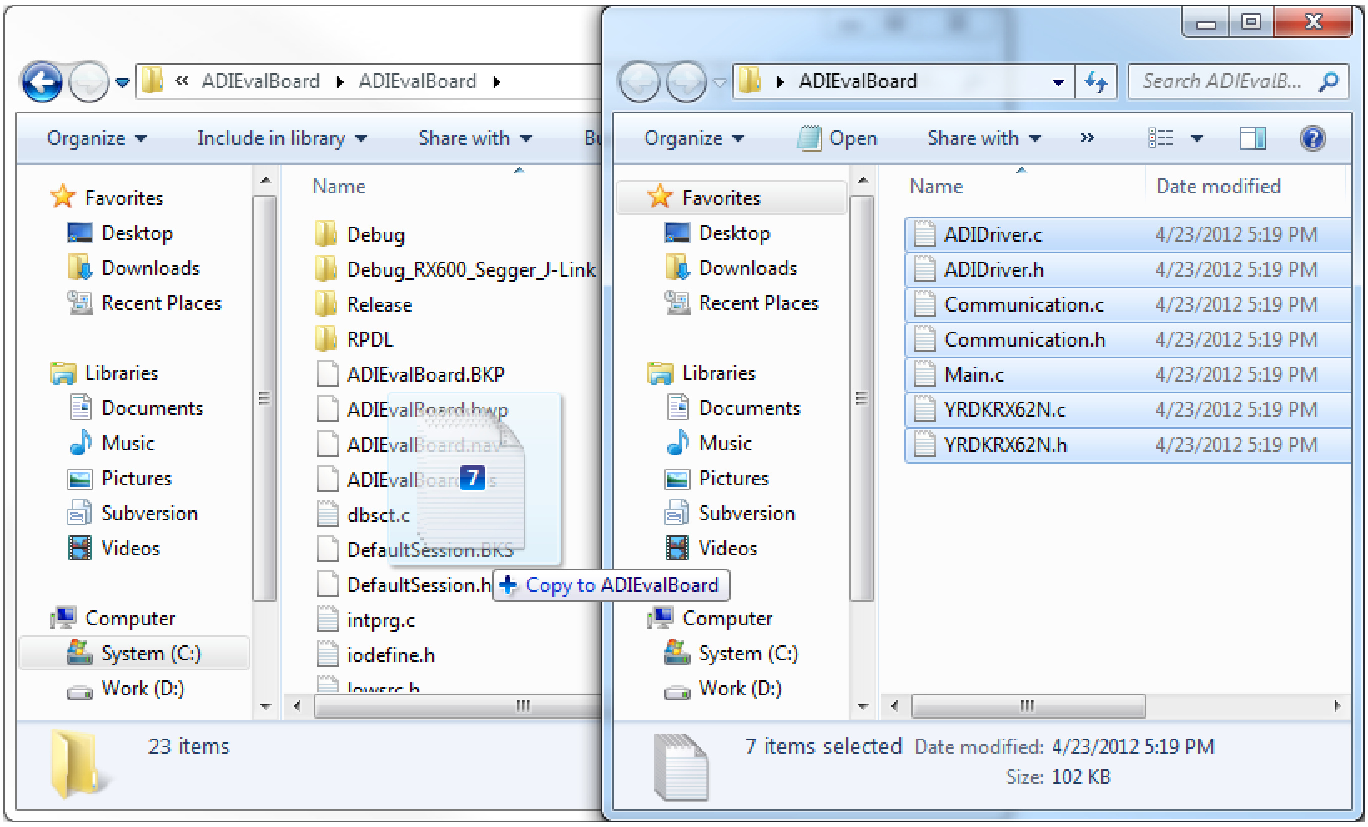
```
-noprelink -rom=D=R,D_1=R_1,D_2=R_2 -nomessage -
list="$(CONFIGDIR)\$(PROJECTNAME).map" -nooptimize
-
```

- 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.



Address	Section
0x00001000	B_1
	R_1
	B_2
	R_2
	B
	R
	SU
	SI
0xFFFF8000	PResetP...
0xFFFF8100	C_1
	C_2
	C
	C$*
	D*
	P
	PIntPRG
	W*
0xFFFFFFD0	FIXEDV...

- 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.



Left window:

Debug
Debug_RX600_Segger_J-Link
Release
RPDL
ADIEvalBoard.BKP
ADIEvalBoard.hwp
ADIEvalBoard.nav
ADIEvalBoard.s
dbsct.c
DefaultSession.BKS
DefaultSession.h
intprg.c
iodefine.h
lowsrc.h

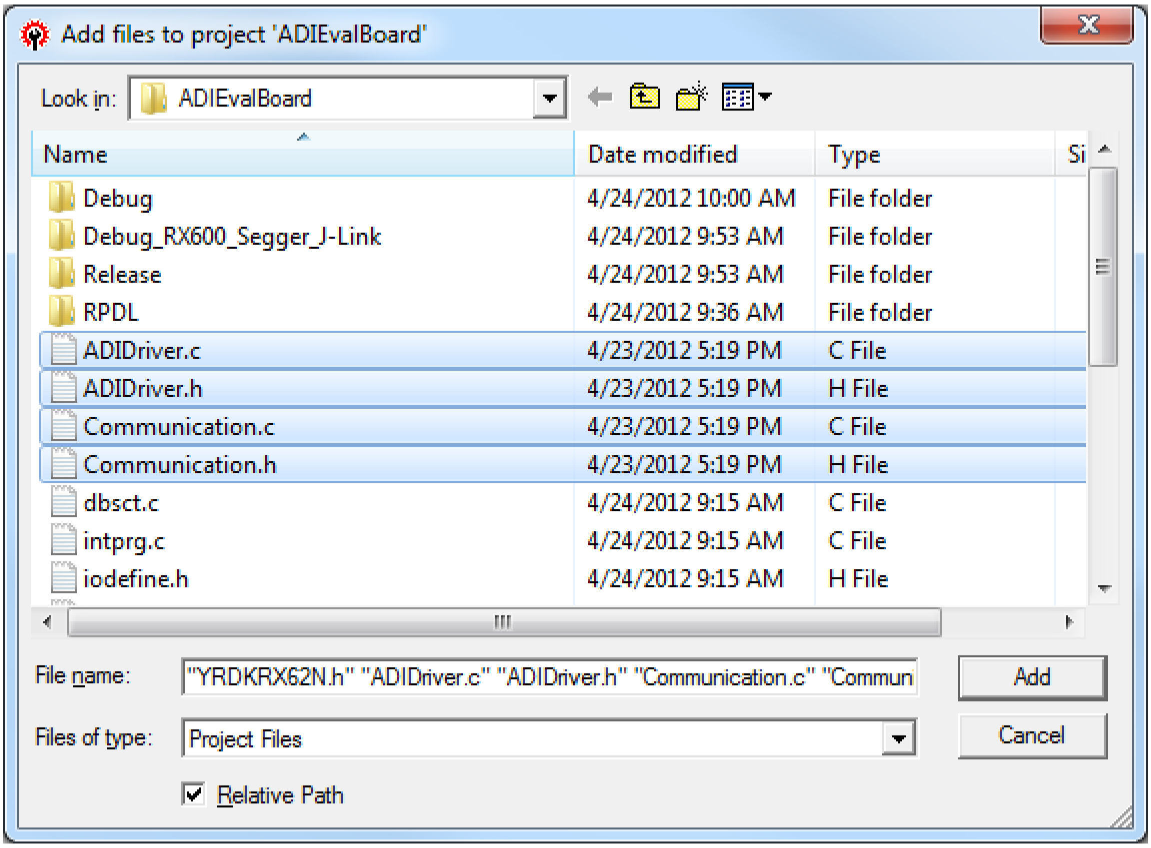
23 items

Right window:

Name	Date modified
ADIDriver.c	4/23/2012 5:19 PM
ADIDriver.h	4/23/2012 5:19 PM
Communication.c	4/23/2012 5:19 PM
Communication.h	4/23/2012 5:19 PM
Main.c	4/23/2012 5:19 PM
YRDKRX62N.c	4/23/2012 5:19 PM
YRDKRX62N.h	4/23/2012 5:19 PM

7 items selected Date modified: 4/23/2012 5:19 PM
Size: 102 KB

- 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*.



Look in: ADIEvalBoard

Name	Date modified	Type	Si
Debug	4/24/2012 10:00 AM	File folder	
Debug_RX600_Segger_J-Link	4/24/2012 9:53 AM	File folder	
Release	4/24/2012 9:53 AM	File folder	
RPDL	4/24/2012 9:36 AM	File folder	
ADIDriver.c	4/23/2012 5:19 PM	C File	
ADIDriver.h	4/23/2012 5:19 PM	H File	
Communication.c	4/23/2012 5:19 PM	C File	
Communication.h	4/23/2012 5:19 PM	H File	
dbsct.c	4/24/2012 9:15 AM	C File	
intprg.c	4/24/2012 9:15 AM	C File	
iodefine.h	4/24/2012 9:15 AM	H File	

File name: "YRDKRX62N.h" "ADIDriver.c" "ADIDriver.h" "Communication.c" "Commun

Files of type: Project Files

☑ Relative Path

- 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:
  - Disable AD9361 HB1, HB2, HB3 and FIR filters by 85083074@qq.com
  - Xilinx AC701 issues with FMCOMMS4 by Raptor99
  - AD5780 & No-OS Driver by mleung
  - fmcomms2 No-OS software (SocKit + RFC AD9361), Unsupported PRODUCT_ID error by dpt_vkt
  - Run fmcomms2 No-OS software (Sockit+RFC ad9361) error by dpt_vkt

01 Jun 2012 11:21