
Lapis_BLE Documentation

Release 0

Silica

Mar 16, 2017

Contents

1	INTRODUCTION	3
1.1	Liberty developement tools	4
1.2	GNURL78 registration email	4
1.3	Documents references	5
1.3.1	Quick start guide	6
1.3.2	Installing E2studio Developement Suite	19
1.3.3	Firmware specification	34
1.3.4	Project tips	39
1.3.5	Linker file contents	52

Version 1.0.0A

Copyright (C)2016 Avnet Silica company

Date 30 may 2014



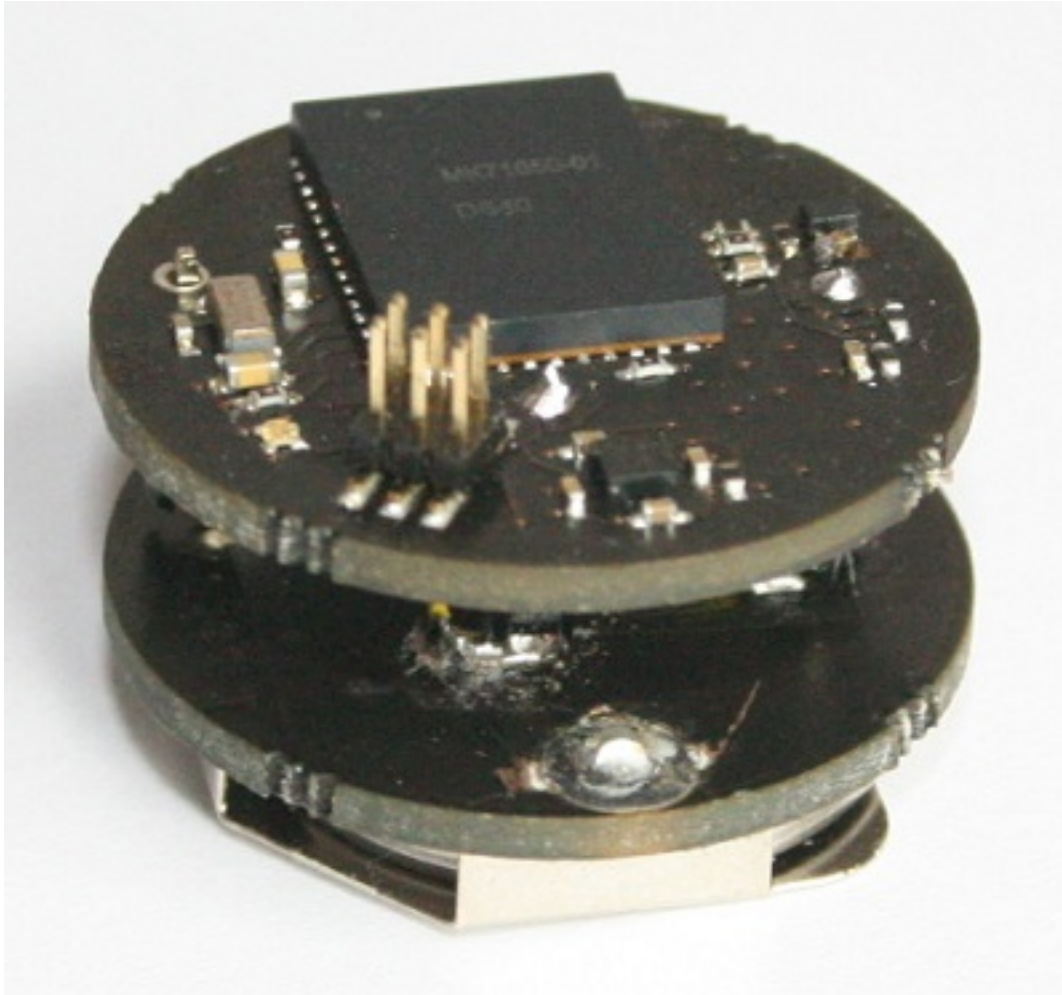
CHAPTER 1

INTRODUCTION

Avnet Silica Liberty board is designed to evaluate BlueTooth Low Energy performance. The main processor is a Renesas R5F100EGANA. The board is also equipped with one ML7105 Lapis BLE module, one KXTJ2-1009 Kionix Accelerometer, one BD1020HFV Rohm Temperature sensor and one SMLP34RGB1W Rohm RGB led.

Using a TTL-USB converter connected with UART_TX_OUT signal of 6 pins header and a Terminal SW on the PC (such as HyperTerminal) it's possible to see a “dump” of all data ‘send to’ and ‘received from’ ML7105 module.

BLE_lapis FW project is based on **VSSPP_ML610QXXX_Ver0_094_beta** firmware revision developed for Lapis PEXL7105_WSN kit and adapted for Avnet Silica Liberty Board. It is adapted for VSP application; no test has been made for VSSPP application.



Liberty developement tools

Firmware was developed using:

- e2studio Version 2.2.0.13 based on Eclipse SDK 3.8.2 and CDT 8.2.1. ([click here to download e2studio 2.2.0.13](#))
- KPIT GNURL78 Version 13.02-ELF-MP1 compiler ([click here to download GNURL78 13.02](#))

Note: Registration and login to Renesas and KPIT site are needed before using link above to download

Important: e2studio 2.2.0.13 or KPIT GNURL78 13.02-ELF-MP1 are also available on request. In this case, you must register at Renesas and KPIT official sites for license activation

GNURL78 registration email

Important: After registration on KPIT site, you receive email such as follow:

Welcome to KPIT GNU Tools

Dear YourName,

Thank you.

Your registration with KPITCummins has been successful with the following registration details:

Login Details:

1. User Name: YourEmail
2. Password: YourPassword

Personal Details:

3. Name: YourName
4. Email Address: YourEmail
5. Activation Code: **YourCode**
6. Address:
7. Telephone Number:

Company Details:

8. Name:
9. Address:
10. Telephone Number:
- 11.URL:
- 12.Position:

Please, note that YourCode is requested during first time GNUKPIT installation!!

You can also find a video guide [here](#)

Firmware project is included into Liberty.zip archive file

Documents references

Reference guide and datasheet are included into **Liberty_doc.zip** archive file and are:

Reference Document	Description
PEDK71050-02-02.pdf	ML7105 Module DataSheet (used in this project)
FEXL7105_BACI_Manual-01.pdf	ML7105_Baci_interface_specifications
FEXL7105_AppDevelopersGuide-01.pdf	Using BACI interface with ML7105
FEUL7105-01.pdf	ML7105 User's Manual
FEDL7105-002-01.pdf	ML7105 DataSheet

Inside **Liberty_doc.zip** there are 3 other folders:

- **ML7105_kit** folder that contains:

Reference Document	Description
FEXL7105_WSNapp-01.pdf	ML7105 WSN Lapis kit user's manual
PEXL7105_VSP_IF-01.pdf	ML7105 WSN Lapis VSP software interface
PEXL7105_VSSPP-UG-01.pdf	ML7105 WSN Lapis VSSPP guide
PEXL7105_WSN_HardManual-03.pdf	ML7105 Lapis kit hardware manual









- **e2studio** folder that contains:

Reference Document	Description
r20ut2771ej0100_e2_start_s.pdf	e2studio quick start

- **DataSheet** folder that contains:

Reference Document	Description
bd1020hfv-e.pdf	Temperature sensor IC
KXTJ2-1009 Specification Rev 4.pdf	Accelerometer
RL78_G13_Hardware.pdf	Renesas RL78 series hardware manual
SMLP34RGBxW_last.pdf	RGB Led

Liberty_doc.zip contents - see image below

Name	Date modified	Type	Size
 PEDK71050-02-02.pdf	11-10-2013 Fri 11:10	Documento Adobe Acrobat	297 KB
 FEXL7105_BACI_Manual-01.pdf	18-06-2013 Tue 15...	Documento Adobe Acrobat	505 KB
 FEXL7105_AppDevelopersGuide-01.pdf	28-06-2013 Fri 18:15	Documento Adobe Acrobat	616 KB
 FEUL7105-01.pdf	11-03-2014 Tue 11...	Documento Adobe Acrobat	536 KB
 FEDL7105-002-01.pdf	04-09-2013 Wed 0...	Documento Adobe Acrobat	469 KB
 ML7105_kit	22-05-2014 Thu 13...	File folder	
 e2studio	20-05-2014 Tue 14...	File folder	
 DataSheet	20-05-2014 Tue 14...	File folder	

You can download the Liberty.zip and Liberty_doc.zip accessing to Avnet Silica Architech website (*registration is needed for download*)

Installing E2studio and KPIT Development Suite chapter will guide you through the basic steps of the installation procedure of Development Suite

We suggest you to read first the Quick Start Guide to perform a correct setup.

Quick start guide

This guide explains how to use this application and provides an overview of the structure of the project firmware

Contents:

Quick start guide

Hardware requirements

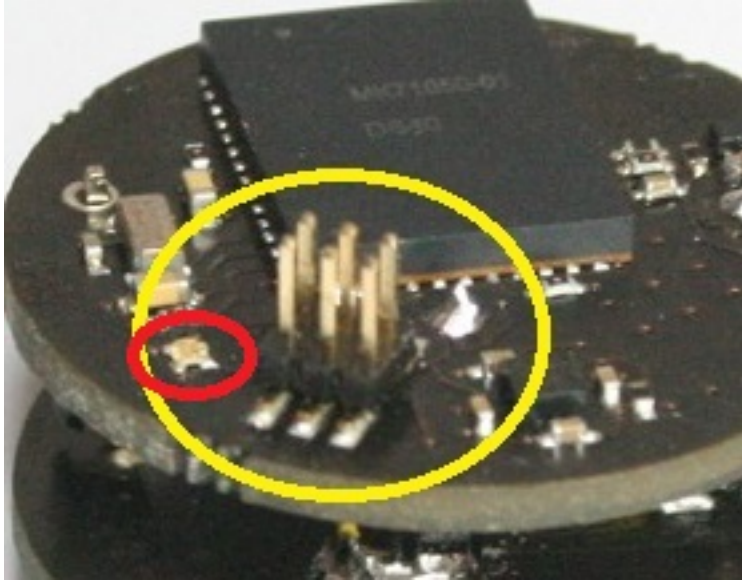
- Silica Liberty Board kit
- One TTL-USB adapter (for communication dump)
- Renesas E1 on-chip debugger
- PC with terminal software (such as HyperTerminal)

Board features

Liberty board has a 6 pin header connector (*yellow circled in figure below*) that can be used for:

- debug (with E1 pod - see hardware documentation for pinout)
- connect TTL-USB converter to see a “dump” of all communication between Host MCU and BLE ML7105 module

Liberty has a RGB led that monitor main BLE state (*red circled in figure below*)



Note: see hardware documentation for Liberty header pinout and how to connect for debug and serial dump

Software requirements

- e2studio Version 2.2.0.13 and KPIT GNURL78 Version 13.02-ELF-MP1 compiler
- Liberty_BLE_v1 firmware
- PC terminal emulator (such as HyperTerminal)
- Liberty Software Application for PC or SmartPhone (to connect with Liberty board)

Note:

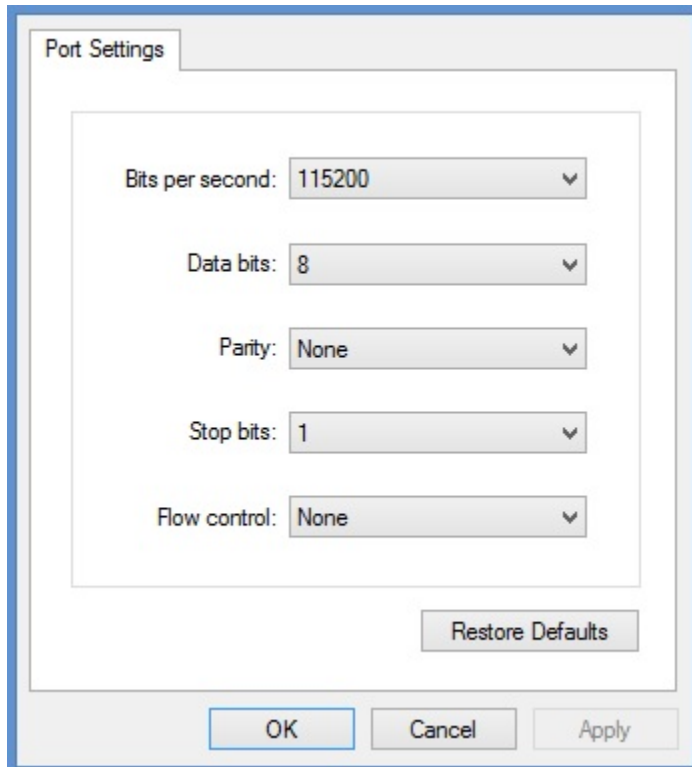
If you have not yet installed e2studio developement suite, before proceeding, go to:

[Installing E2studio Development Suite](#)

Communication dump

If you like to use serial dump to monitor communication between CPU and ML7105, connect a TLL-USB converter as described in *[Board features](#)* note. You can use a terminal software on your PC to see this dump. Here settings required for HyperTerminal COMx parameter:

speed = 115200 baud
data with = 8
parity = none
stop bit = 1
flow control = none



Hardware setup

Plug the Battery into socket adapter (see figure below)



Red led will ON to indicate initial settings (if connected, you can see the communication dump on PC HyperTerminal)

After initialization, the green led will flash every 2 second to indicate that Liberty is in active state waiting connection.

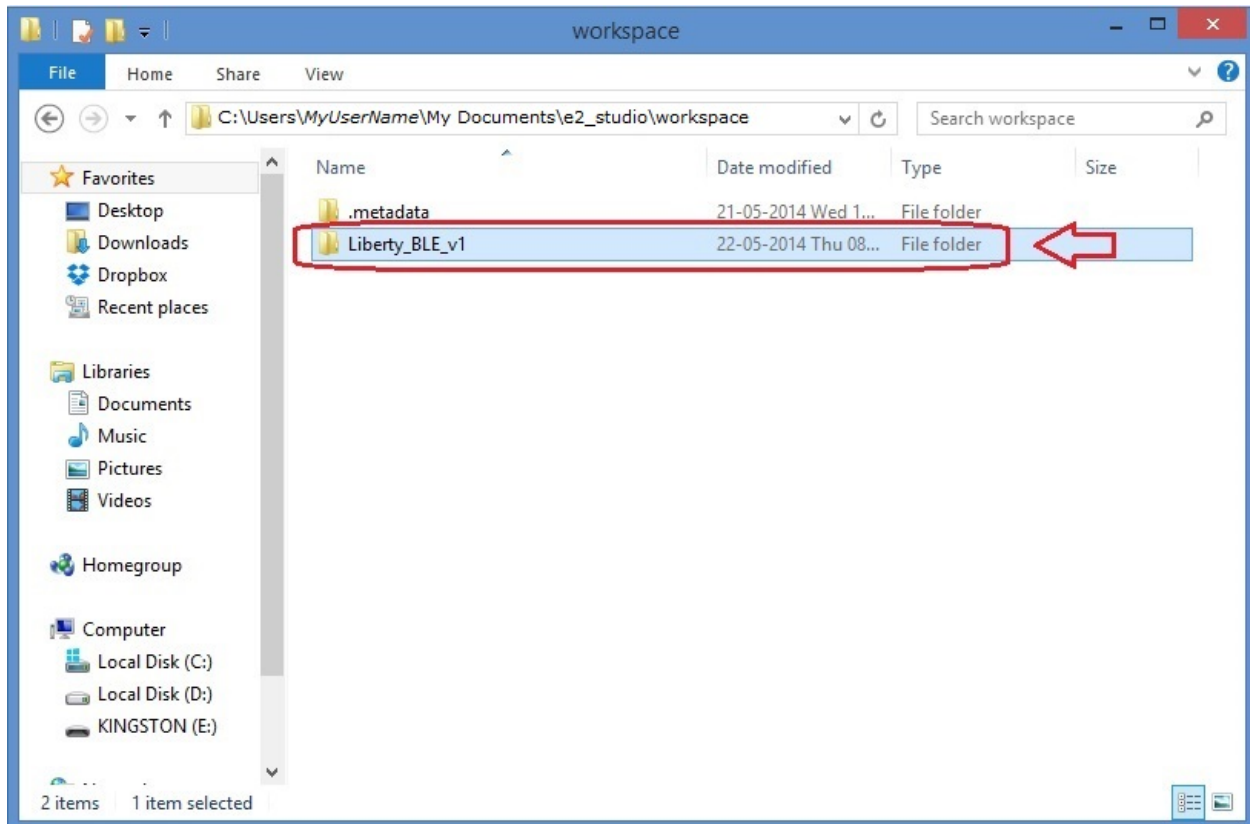
Try to connect an pair with Liberty Software Application, the green led will lamp (half second period), and after the green ON means “device paired” Use Liberty Application (PC or Smartphone) Software to perform basic functions with Liberty board.

For debugging purpose, you can use Renesas E1 on-chip debugger. You must have a cable adapter to connect emulator pod with Liberty board (see hardware documentation)

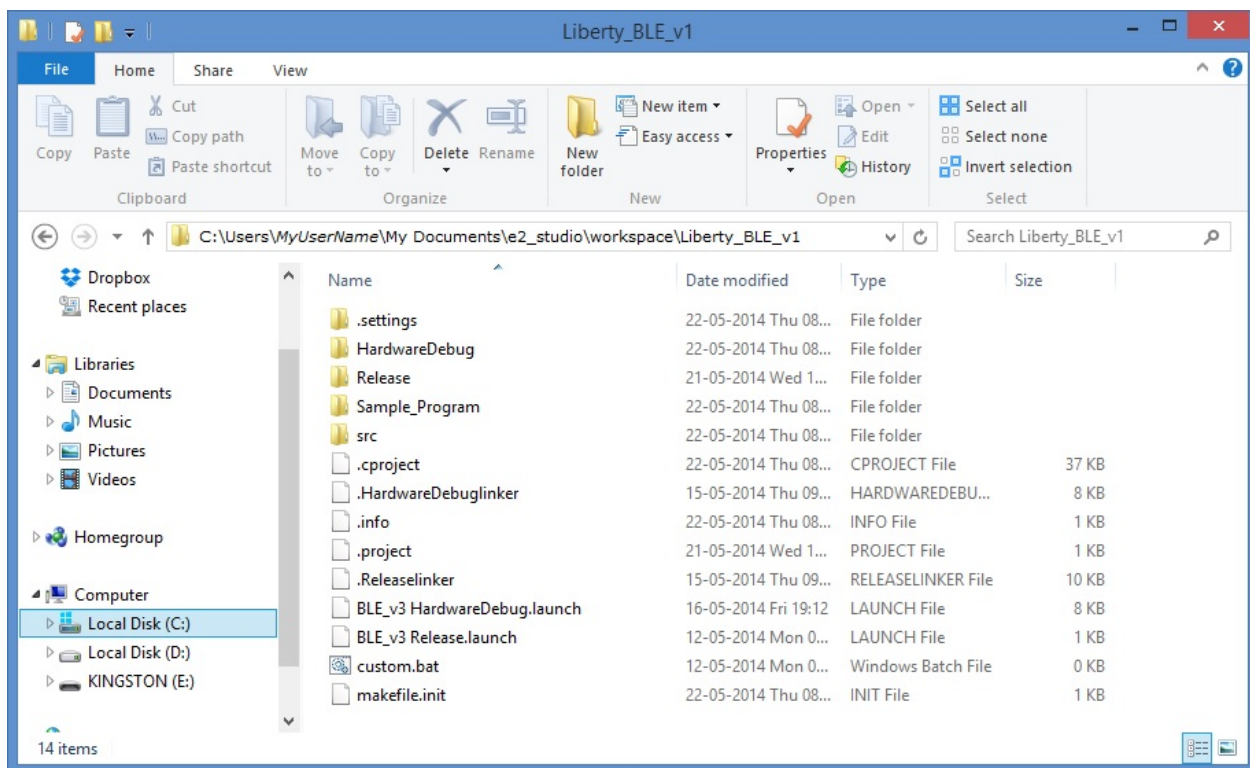
Liberty Board FW installation & setup

Tip: Install first E2studio development suite, then proceed with firmware setup

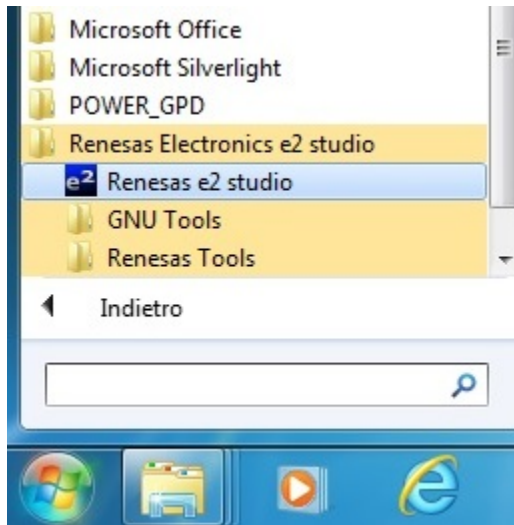
- Inside E2studio workspace folder (C:\Users\ *MyUserName* \My Documents\e2_studio\workspace) create new folder named **Liberty_BLE_v1**



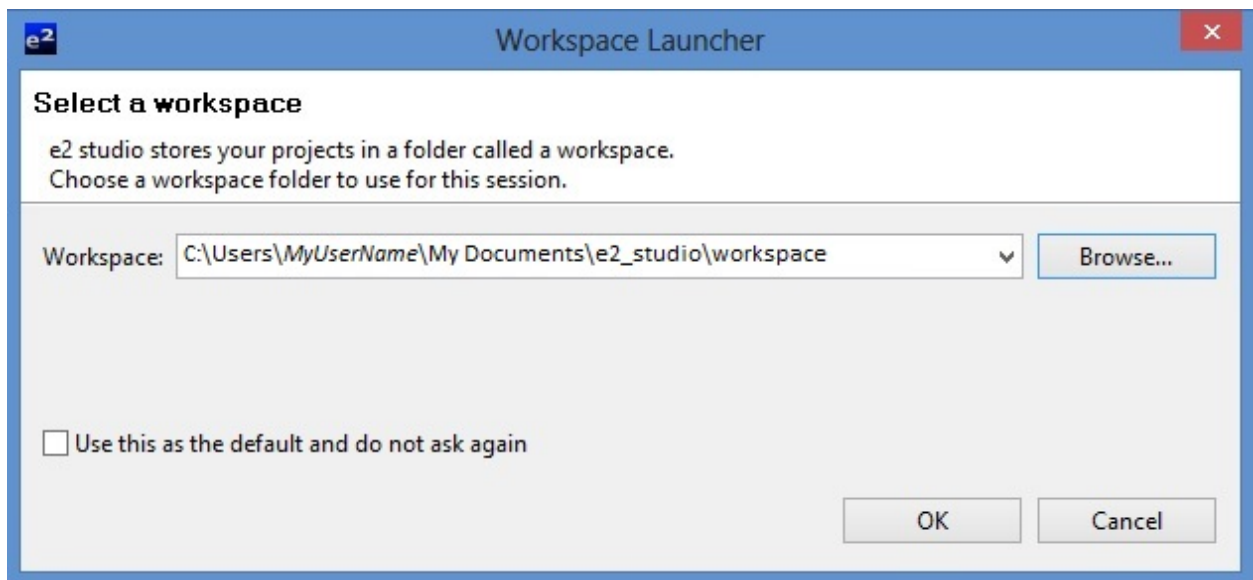
- Unzip all files from Liberty.zip into the folder `C:\Users\ MyUserName \My Documents\e2_studio\workspace\Liberty_BLE_v1` just created



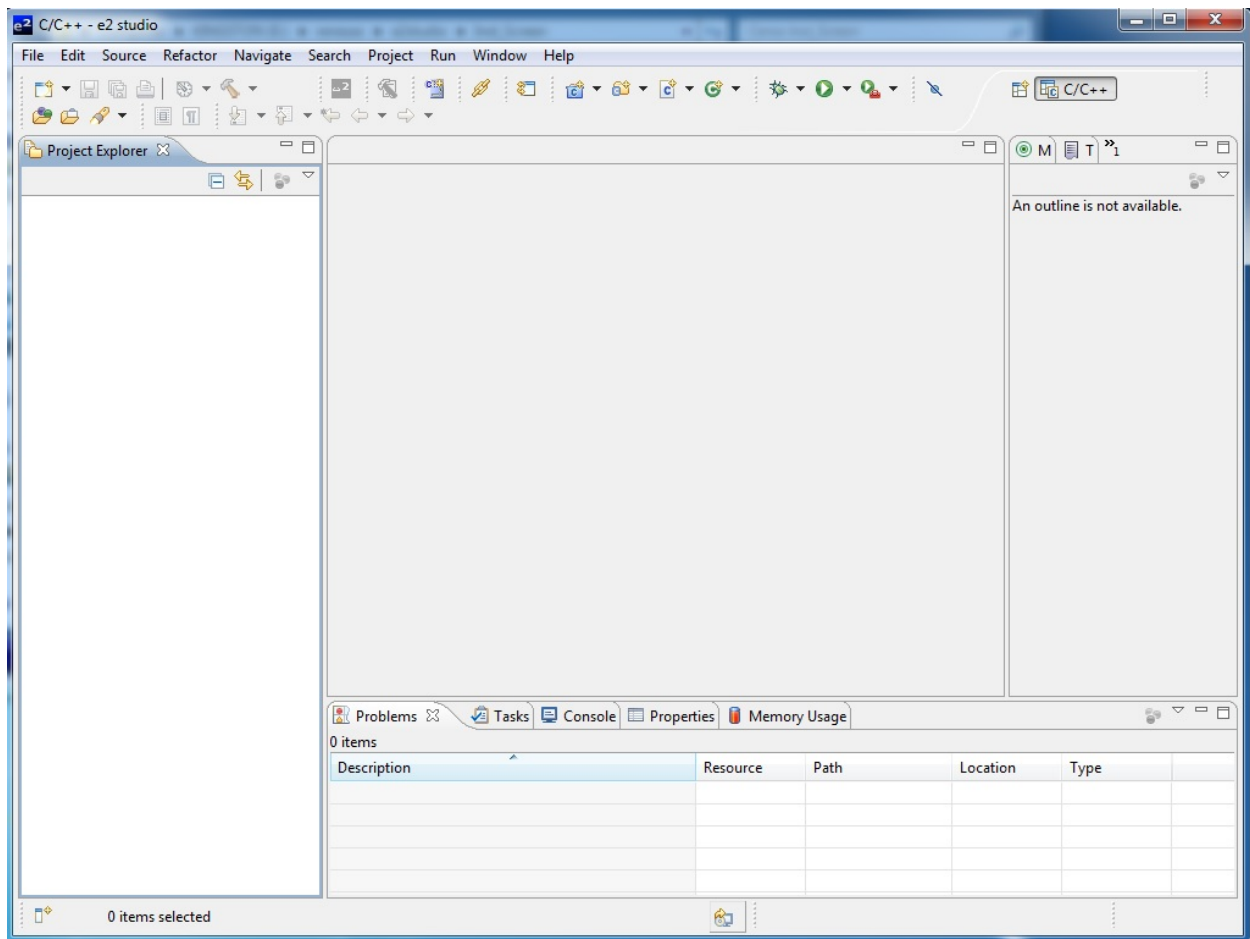
- Go to Start → Renesas Electronics e2 studio → Renesas e2 studio → Eclipse and click on to start E2studio suite



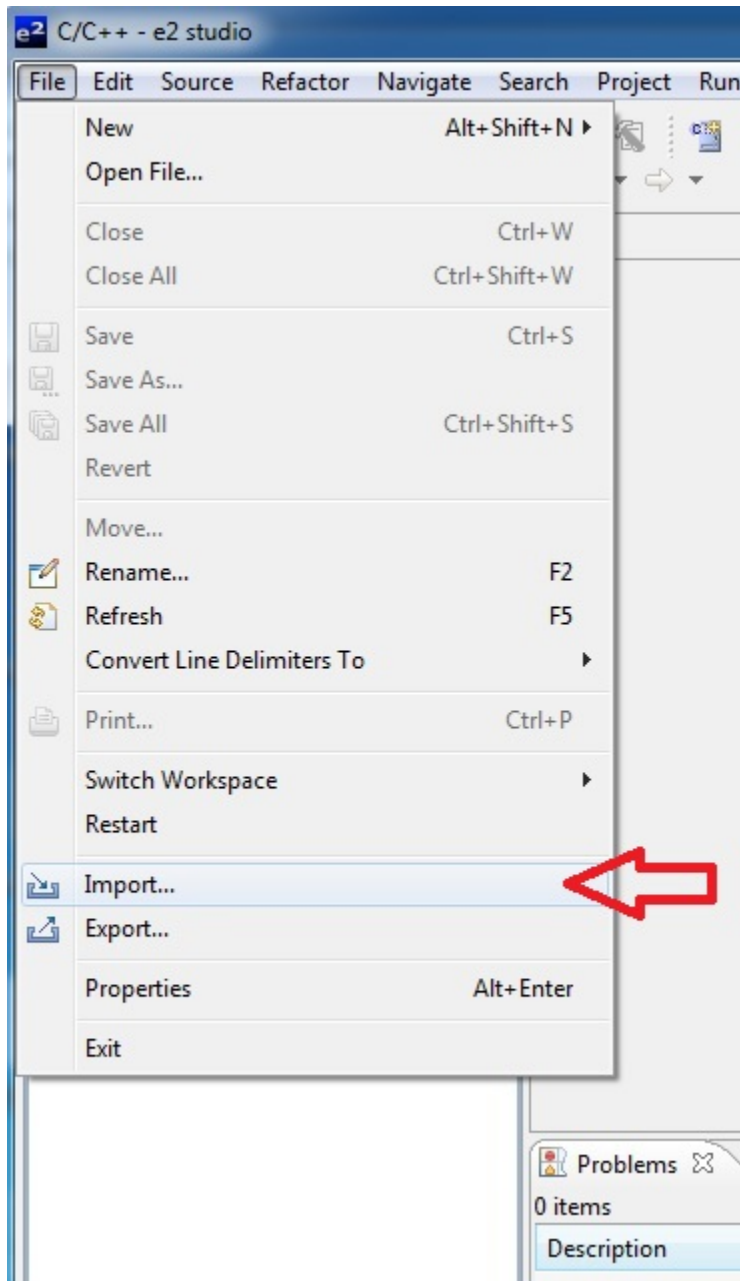
- Check if workspace setting is like figure below. Then click OK to proceed.



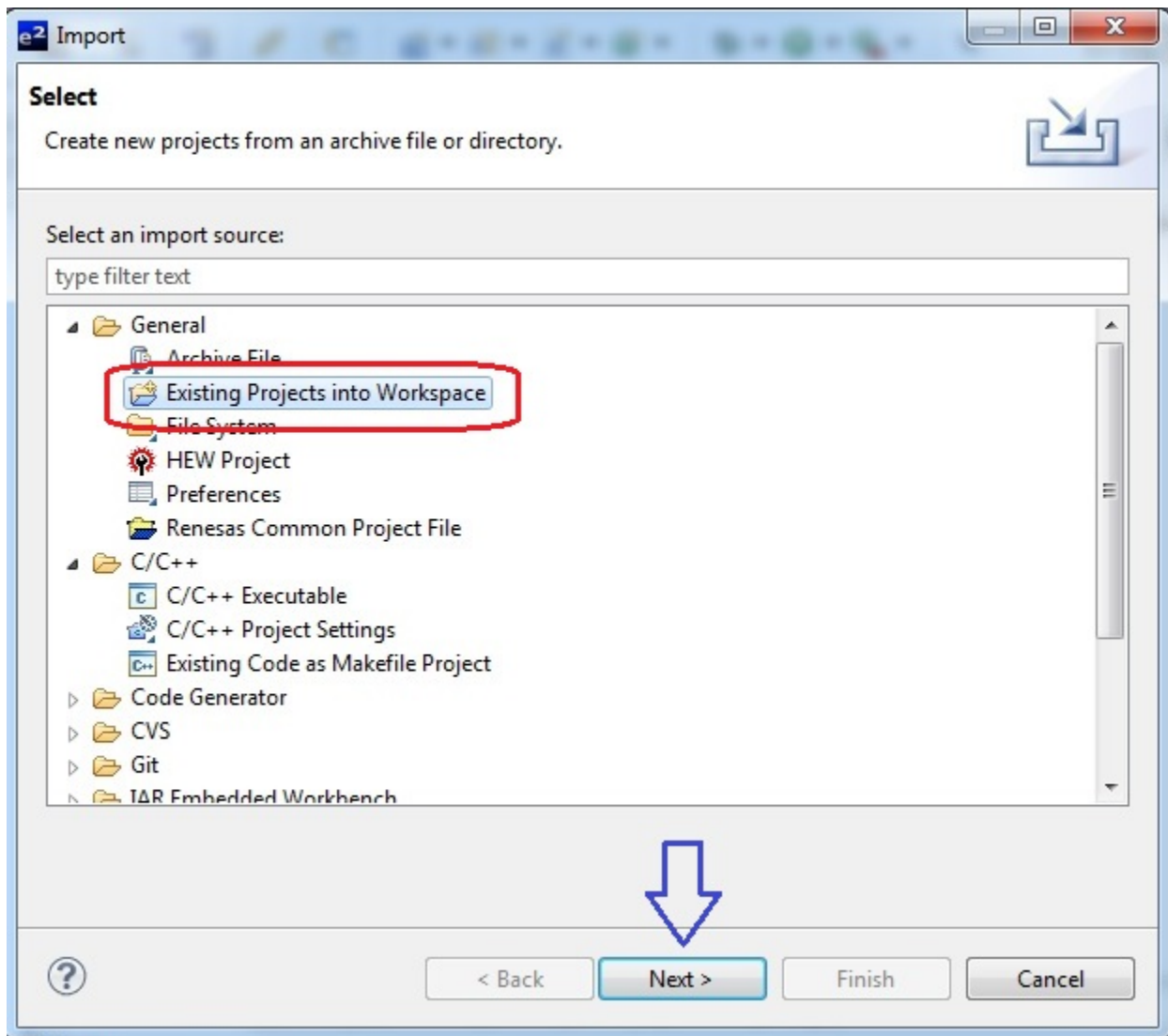
- Now you can see the Eclipse Main Window



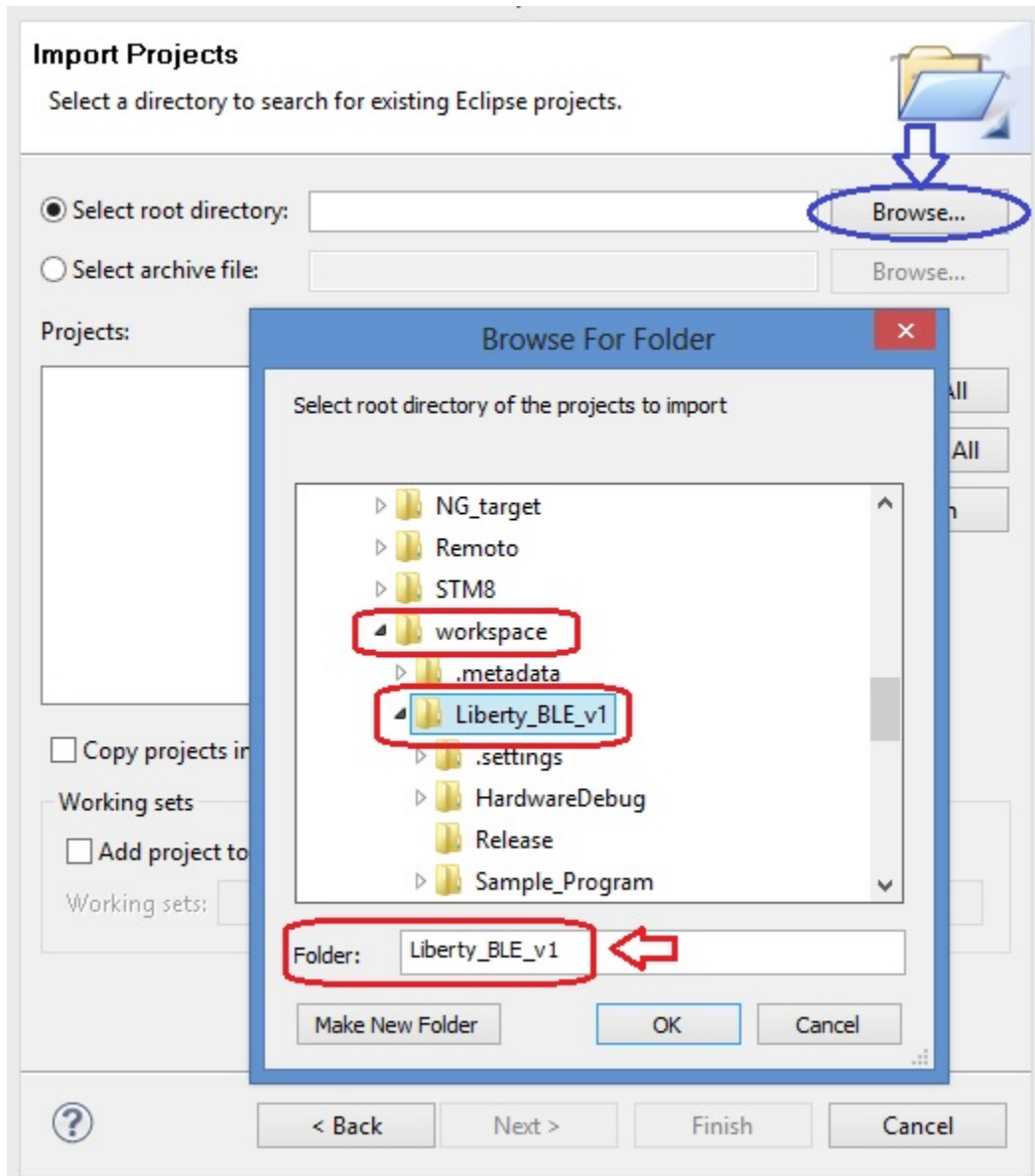
- Select menu File -> Import



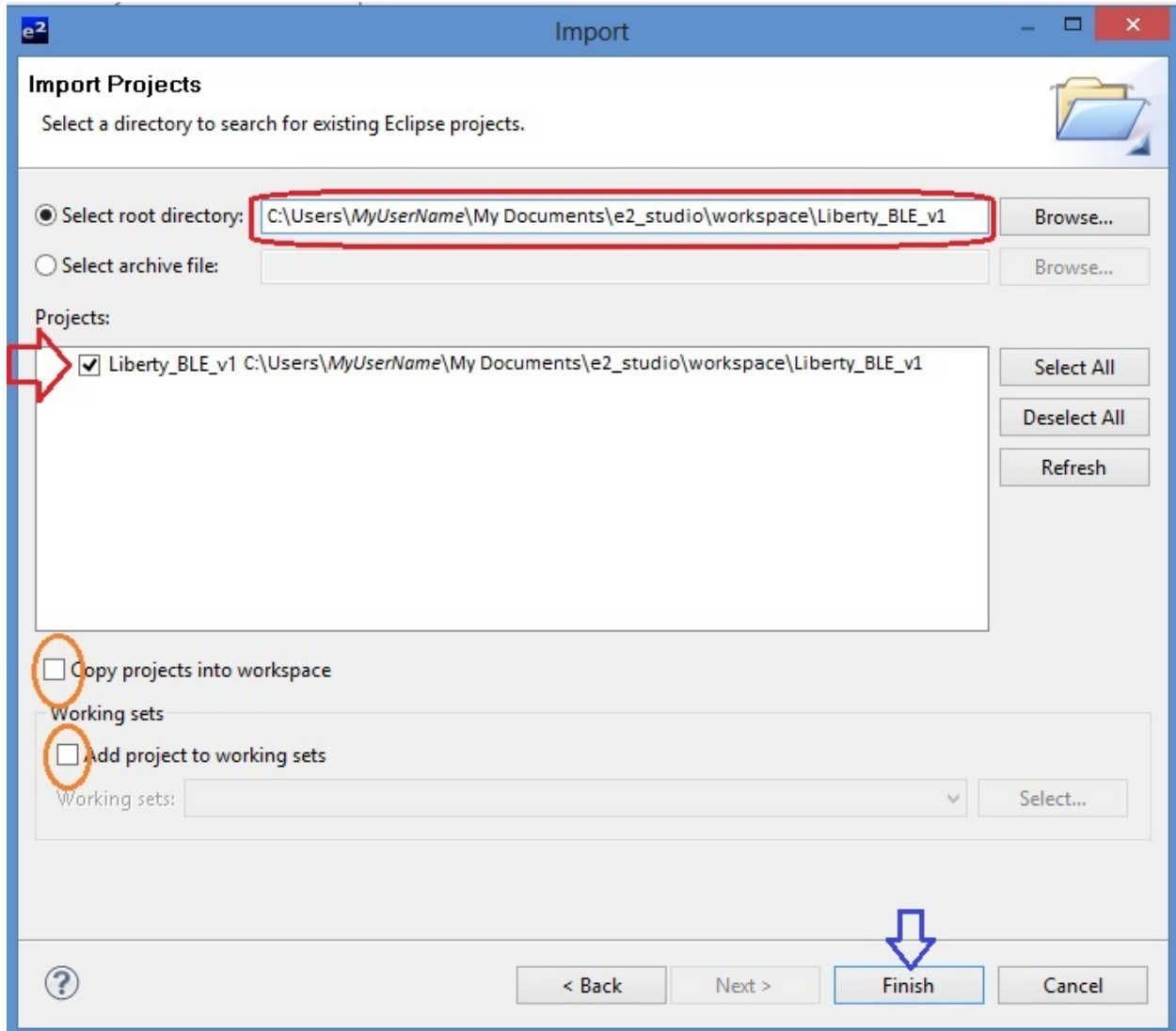
- In the dialog box that will open, click on **General**, select **Existing Projects Into Workspace** and after click "Next" button: new dialog will open.



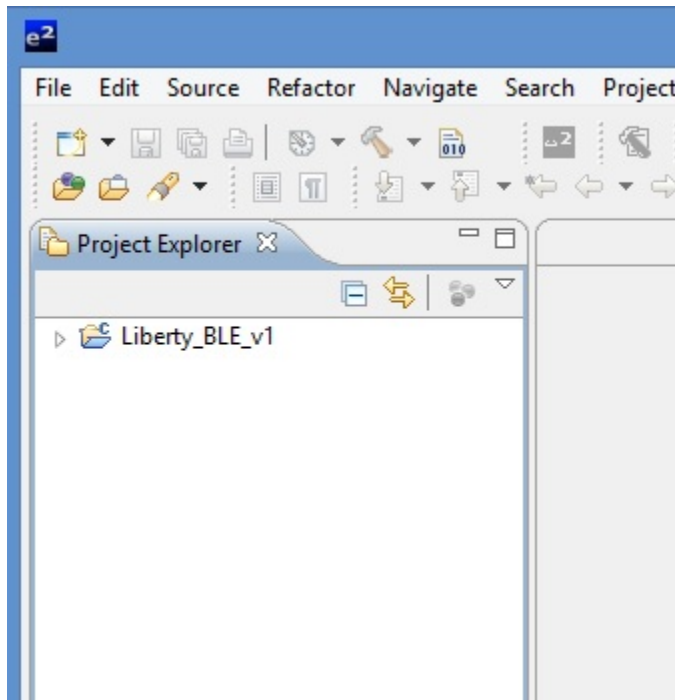
- Click on “Browse...” button and navigate to `C:\Users\ MyUserName \My Documents\e2_studio\workspace\Liberty_BLE_v1` folder. Click on “OK” button



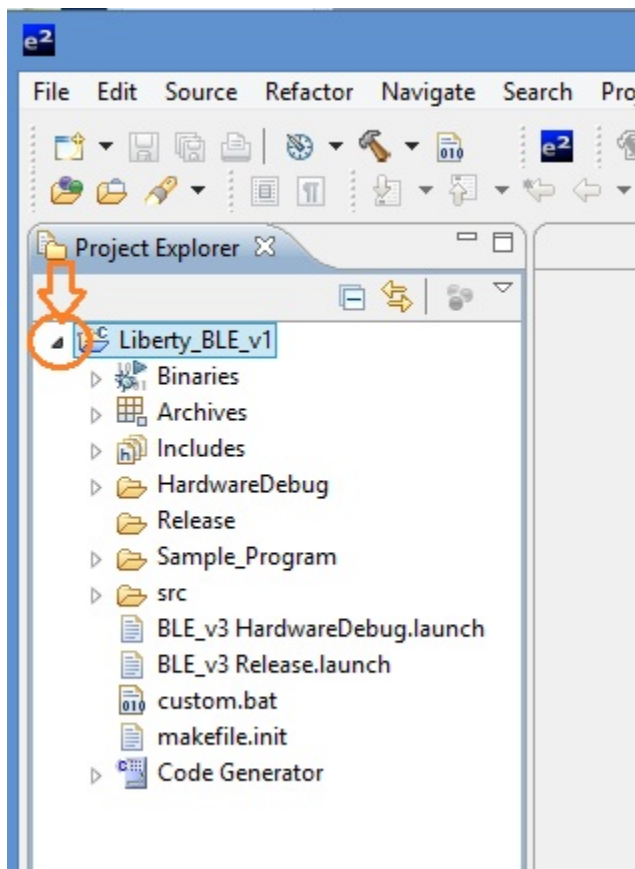
- Check options and setting as the image below, then click “Finish” button to import project.



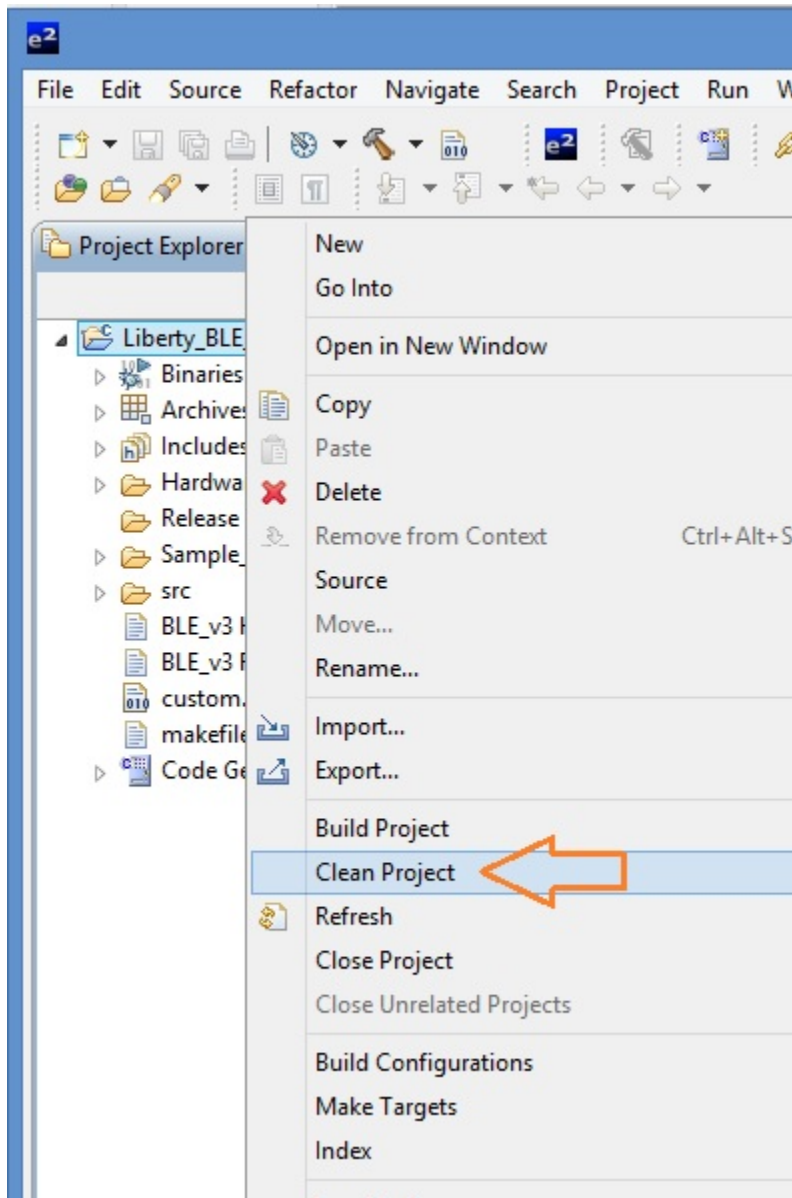
- Wait for project import, then you can see Liberty_BLE_v1 project in the Project Explorer windows of E2studio Platform.



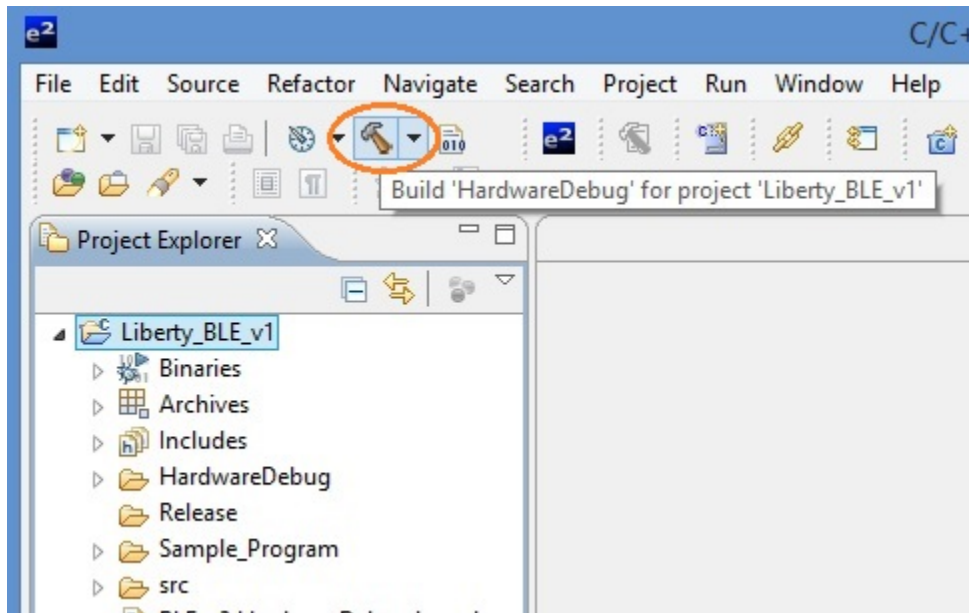
- **First of all**, open project tree by clicking on the arrow located at left of project name (*orange circled*).



- Now, right click over “Liberty_BLE_v1” in the Project Explorer window to point out it, then select “Clean Project”.



- When clean has been made, point mouse over hammer icon on the toolbar and click it (see image below). Hardware debug build will start.



Note: Don't care if you have this warning in the Problems tab (see image above)

Problems					
0 errors, 11 warnings, 0 others					
Description	Resource	Path	Location	Type	
Warnings (11 items)					
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 489	C/C++ Probl...	
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 572	C/C++ Probl...	
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 585	C/C++ Probl...	
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 653	C/C++ Probl...	
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 749	C/C++ Probl...	
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 1397	C/C++ Probl...	
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 1923	C/C++ Probl...	
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 2008	C/C++ Probl...	
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 2466	C/C++ Probl...	
assignment from incompatible pointer type [appl_att_serv...	/Liberty_BLE_v1/Sa...	line 2502	C/C++ Probl...	
assignment from incompatible pointer type [gatt_db_api.c	/Liberty_BLE_v1/Sa...	line 913	C/C++ Probl...	

Now you are ready to work with `Liberty_BLE_v1` project

Installing E2studio Development Suite

This small guide is intended to explain the main steps to properly install the E2studio development suite.



As possibly not shown here, refer to the E2studio quick start guide [r20ut2771ej0100_e2_start_s.pdf](#).

All steps here described are performed with WINDOWS 7 OS

First step

Download E2studio version 2.2.0.13 and KPIT GNURL78 Version 13.02-ELF-MP1. You can find how to make here [Liberty development tools](#) for tools setup

You must have these two files:

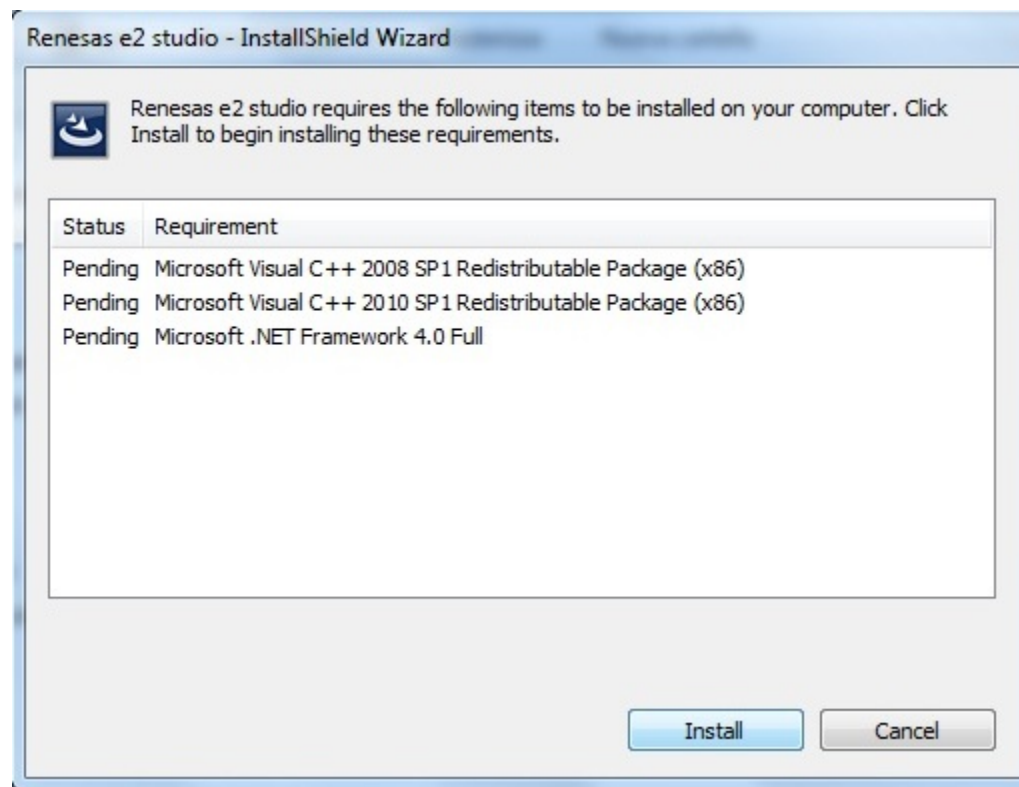
Name	Date modified	Type	Size
 GNURL78v1302-ELF-MP1.exe	20-12-2013 Fri 15:33	Application	51.229 KB
 Renesas_e2_studio_2.2.0.13.exe	02-01-2014 Thu 12...	Application	796.062 KB

Then, you are ready for E2studio Development Suite setup.

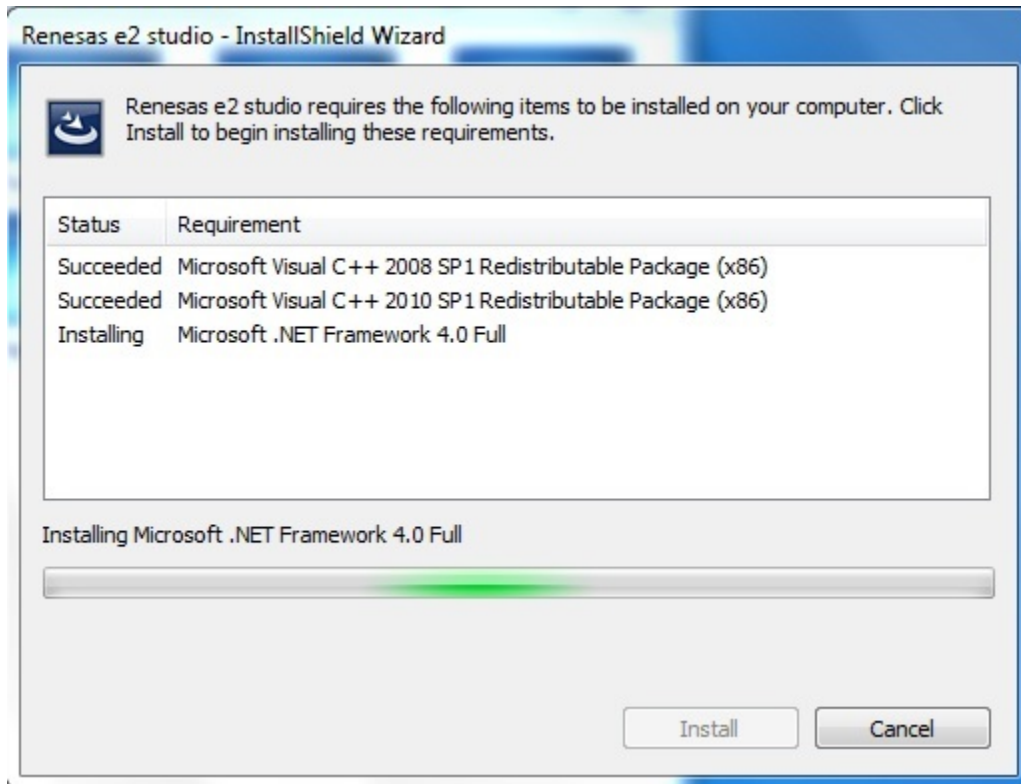
Starting Install E2studio environment

Double click on **Renesas_e2_studio_2.2.0.13.exe**.

First of all, it will ask you to upgrade these OS system settings. If your OS is already updated, this popup will not open

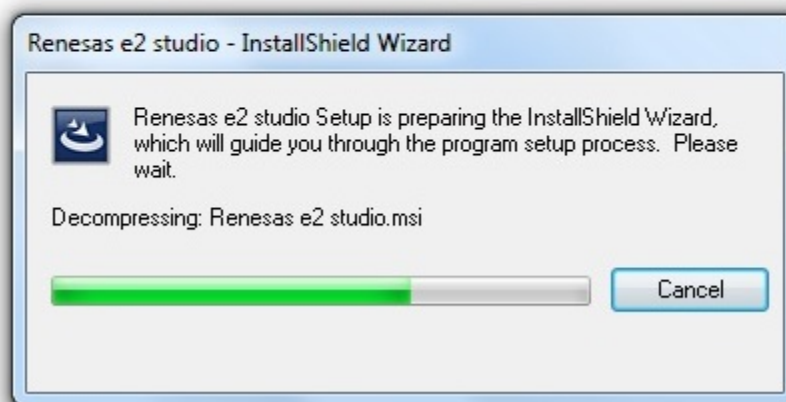


Wait for OS system update. This can take much time

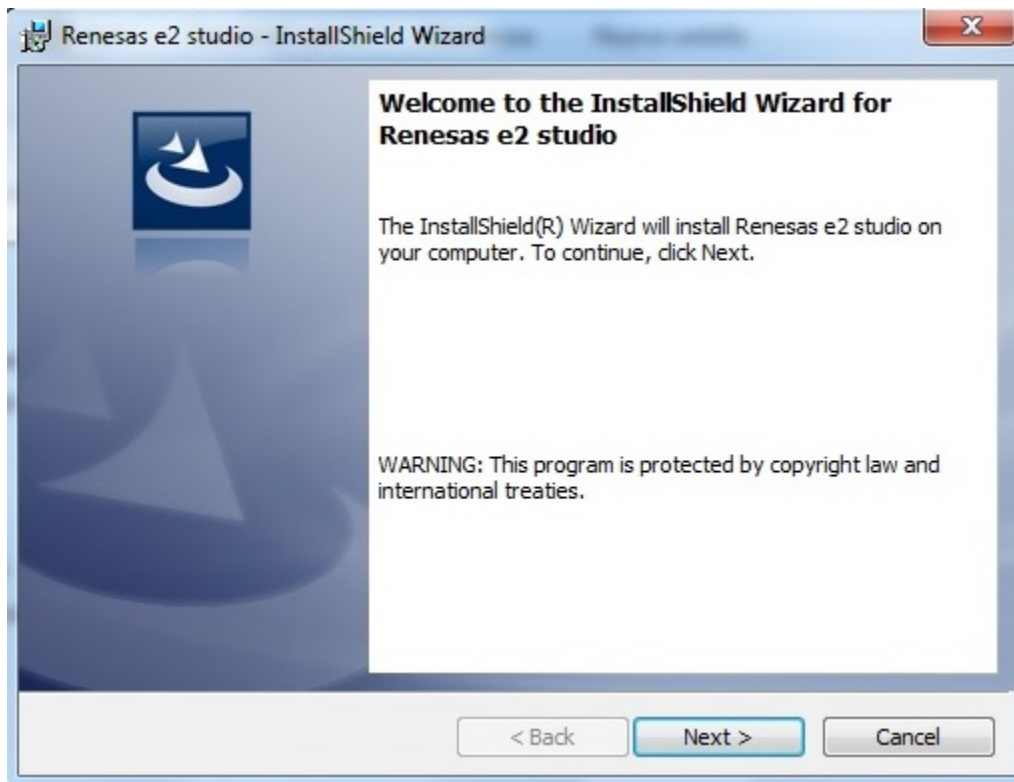


When update ends E2studio starts decompressing

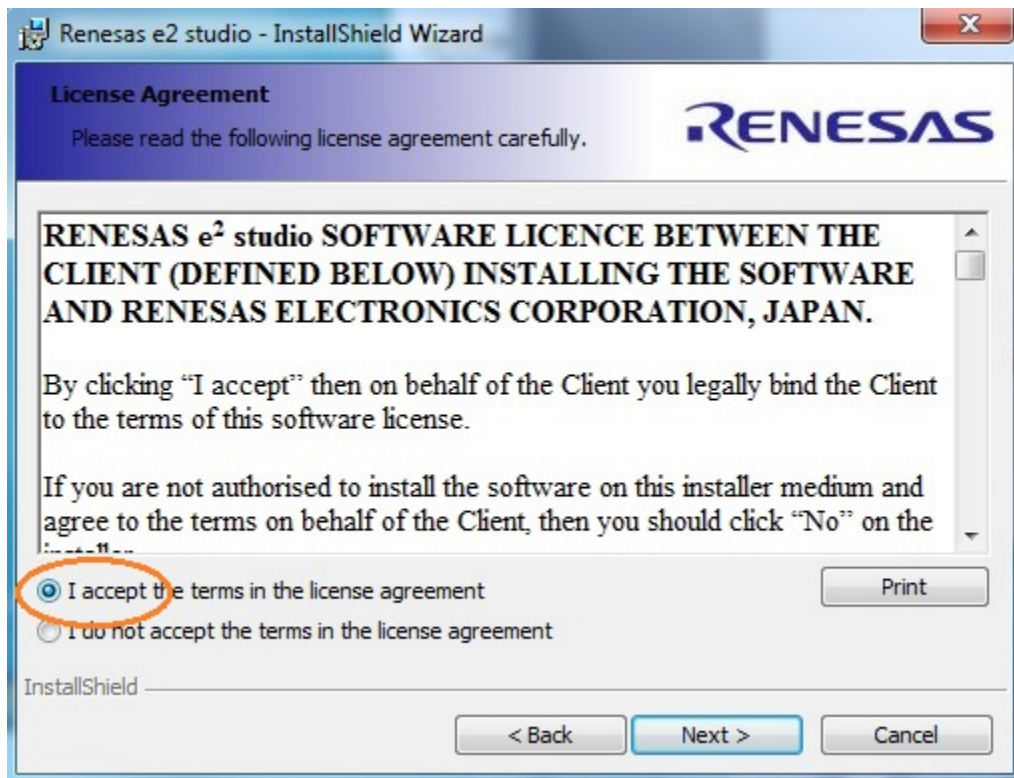
	GNURL78v1302-ELF-MP1	20/12/2013 16:33	Applicazione	51.229 KB
	Renesas_e2_studio_2.2.0.13	02/01/2014 13:42	Applicazione	796.062 KB



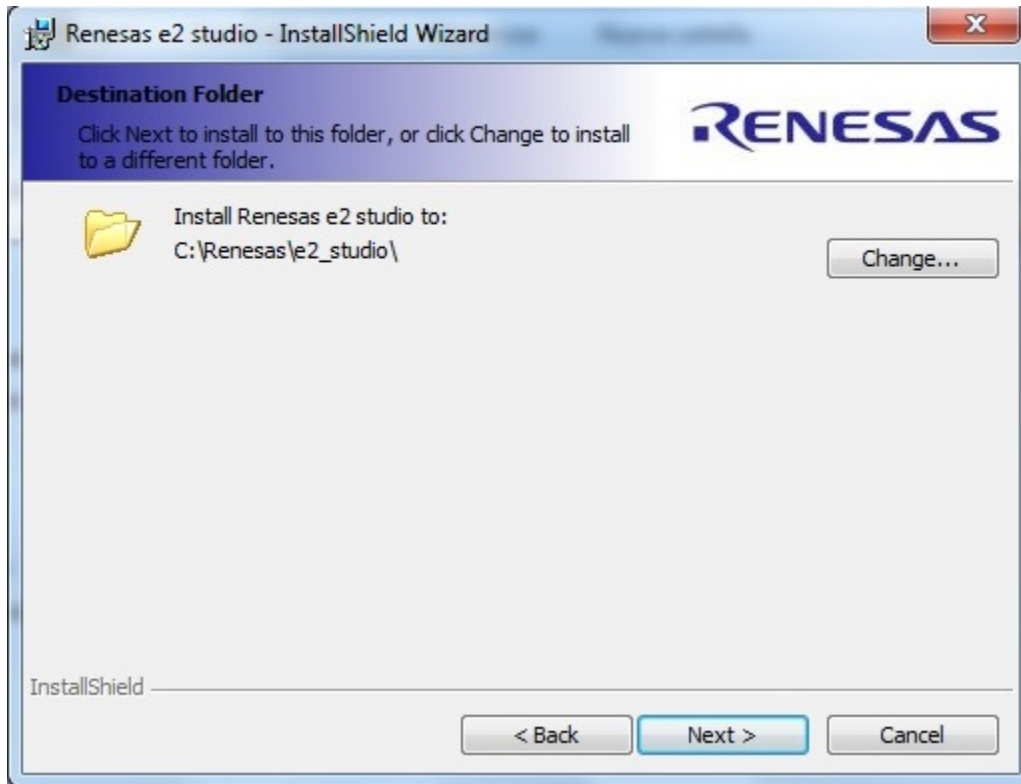
When the following screen will appear, you are ready to install E2studio suite:



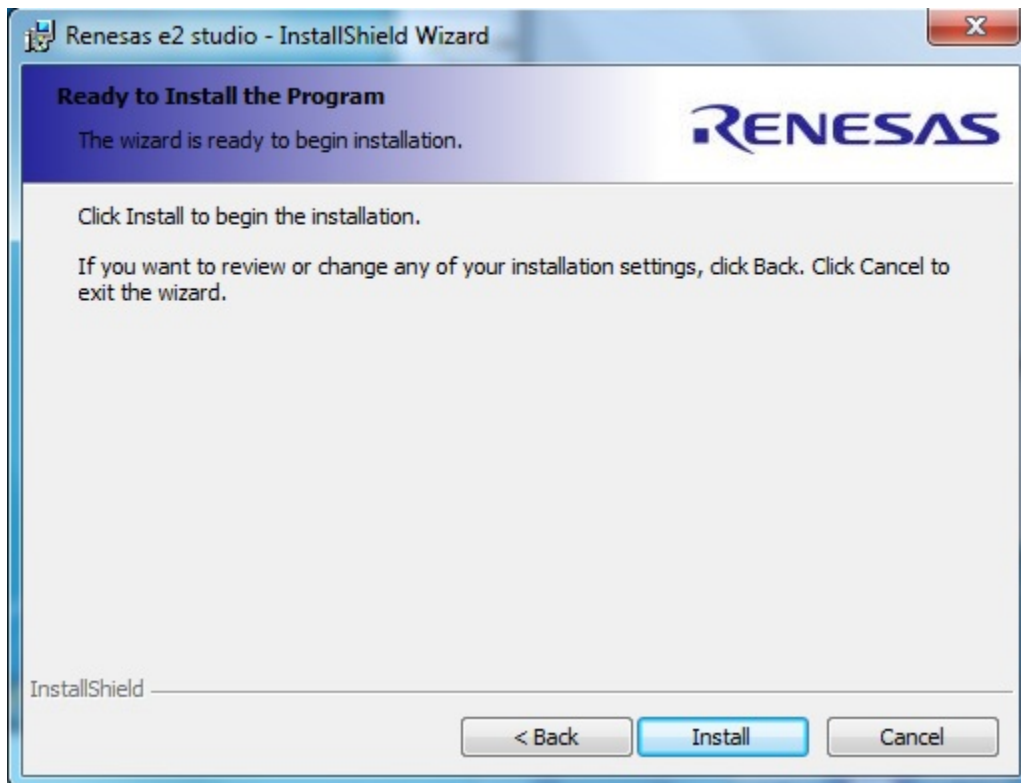
click “next” button, and setup will ask you to agree license



check ‘I accept ...’ then click “next”



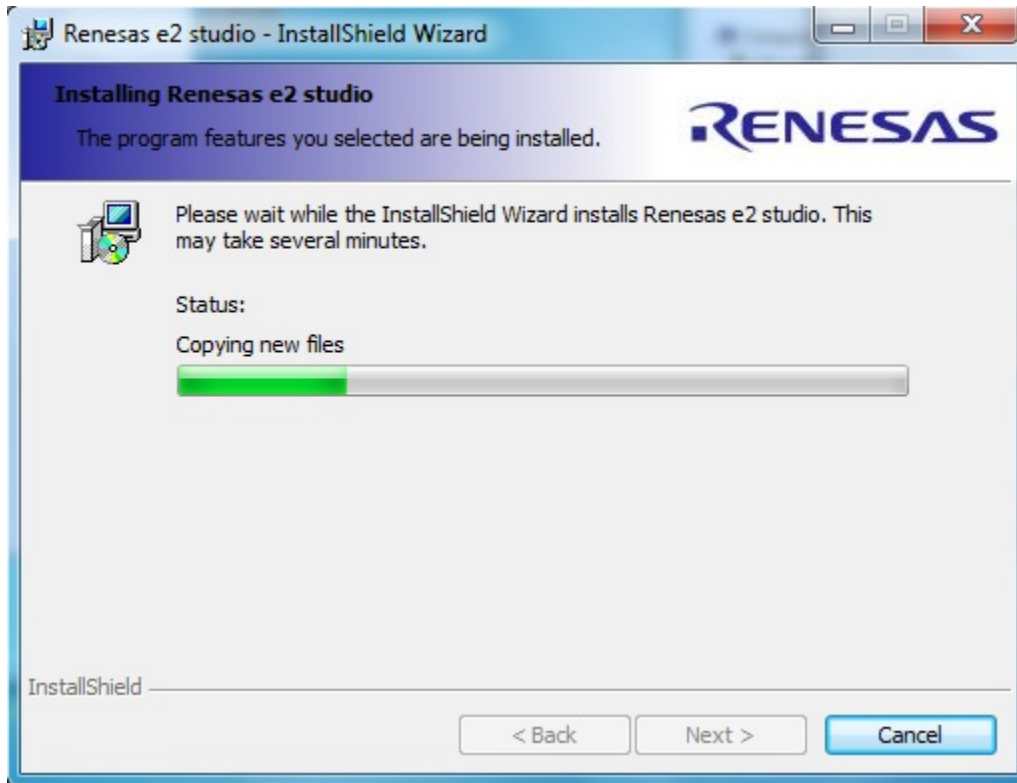
leave unchanged the destination folder and click “next” button



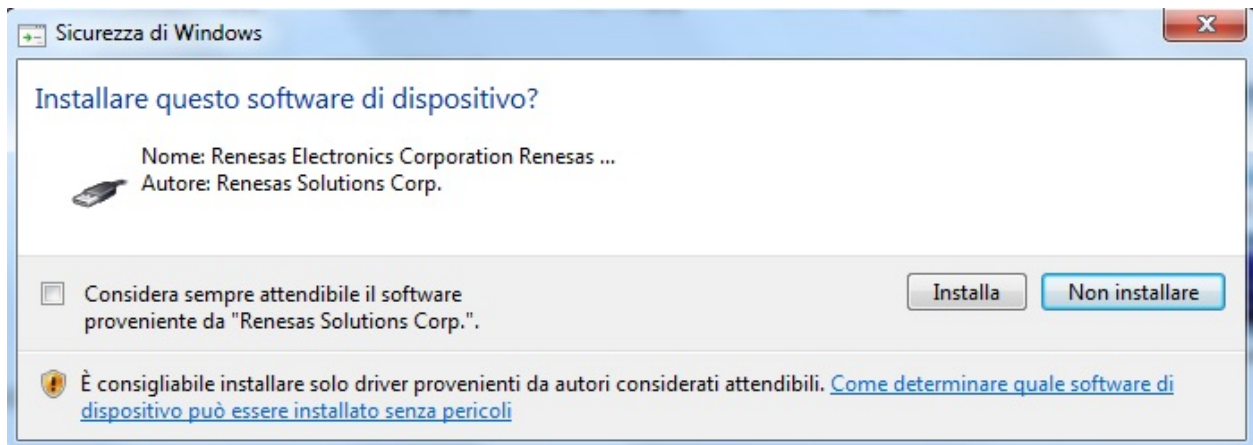
click “install” button to start intallation

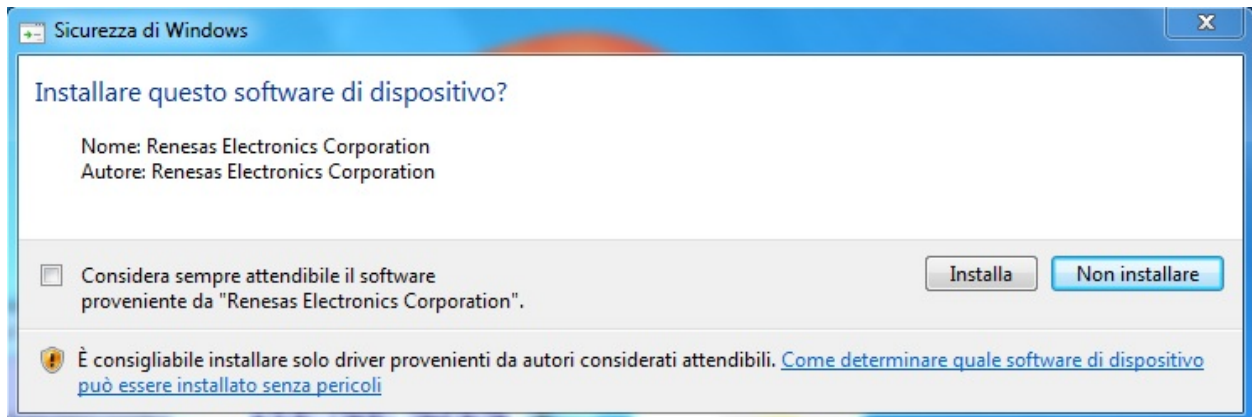
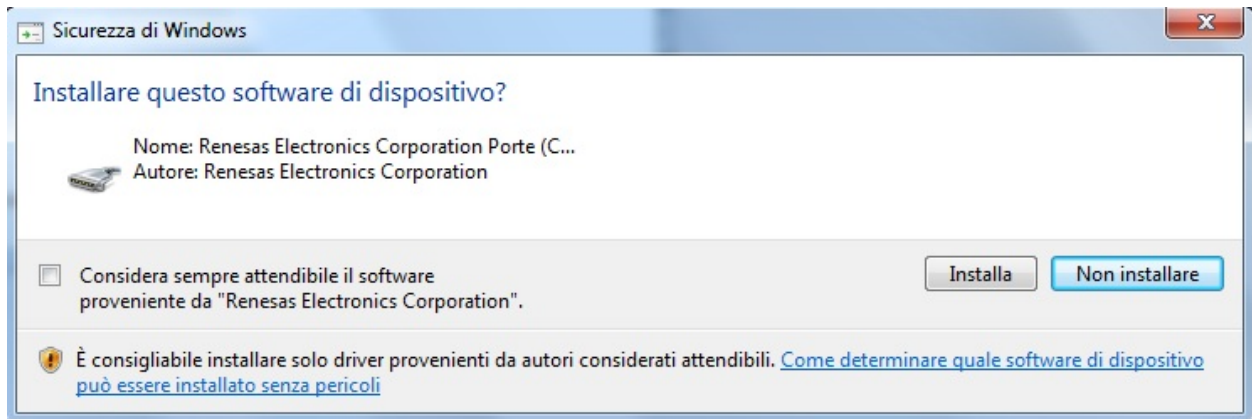
E2studio installation steps

Wait for Wizard operations end.

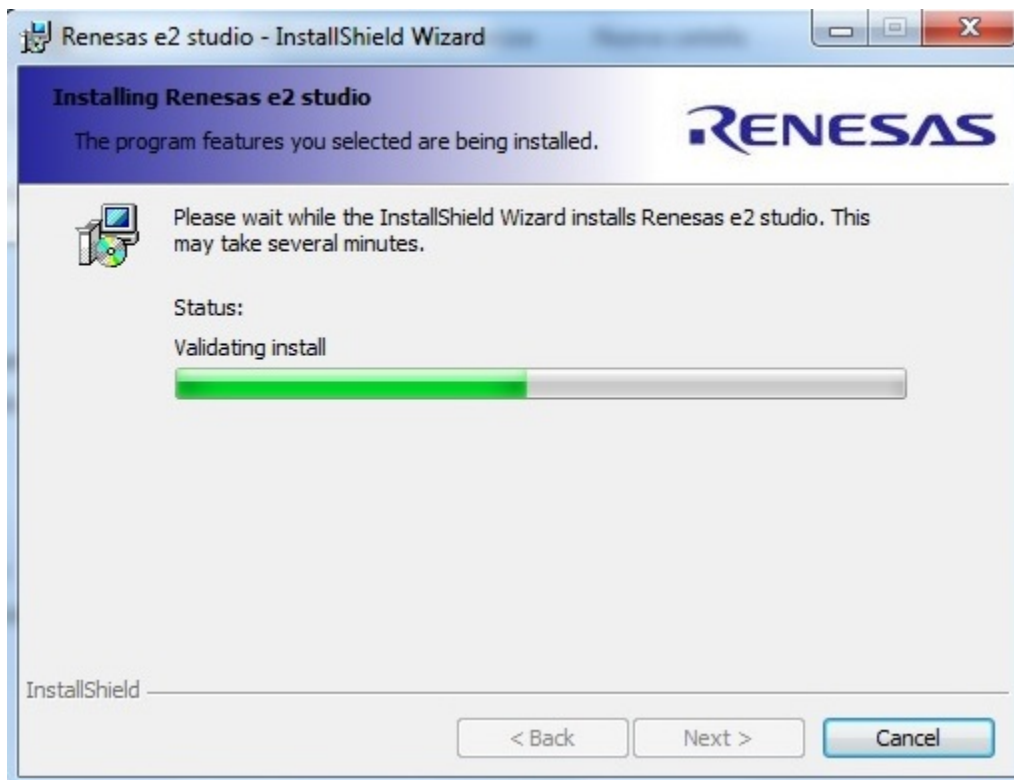


Now Wizard ask for hardware device driver installation: for every popup, click “Install” button

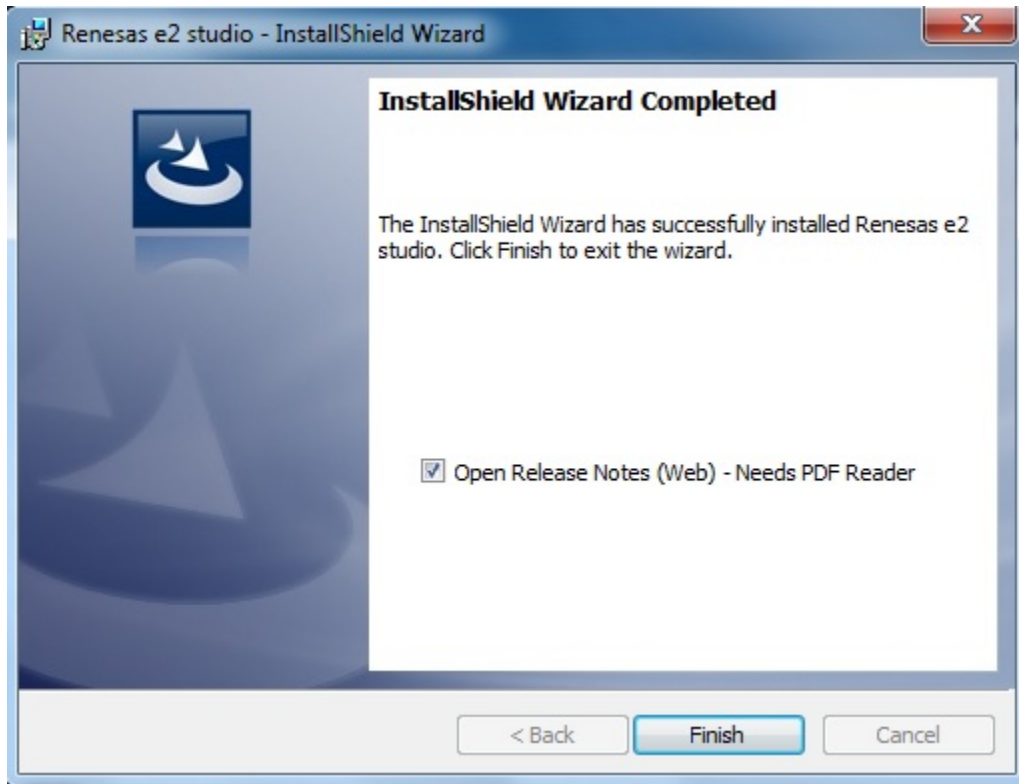




Wait for E2studio validation



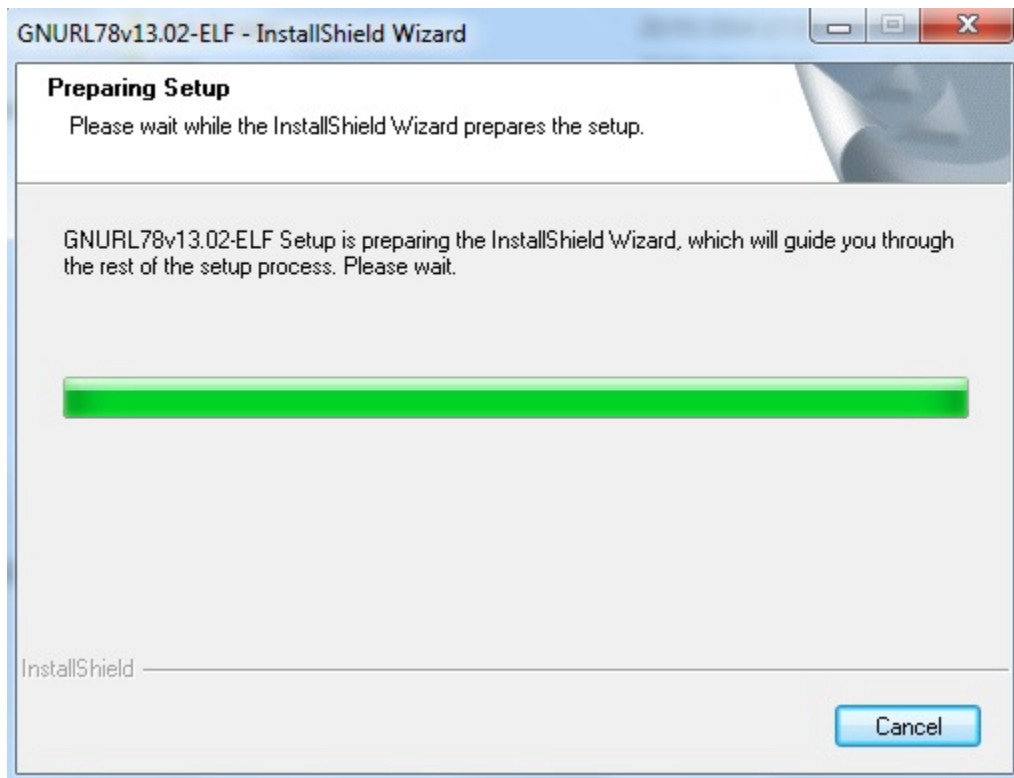
E2 studio succesfull installed on your system



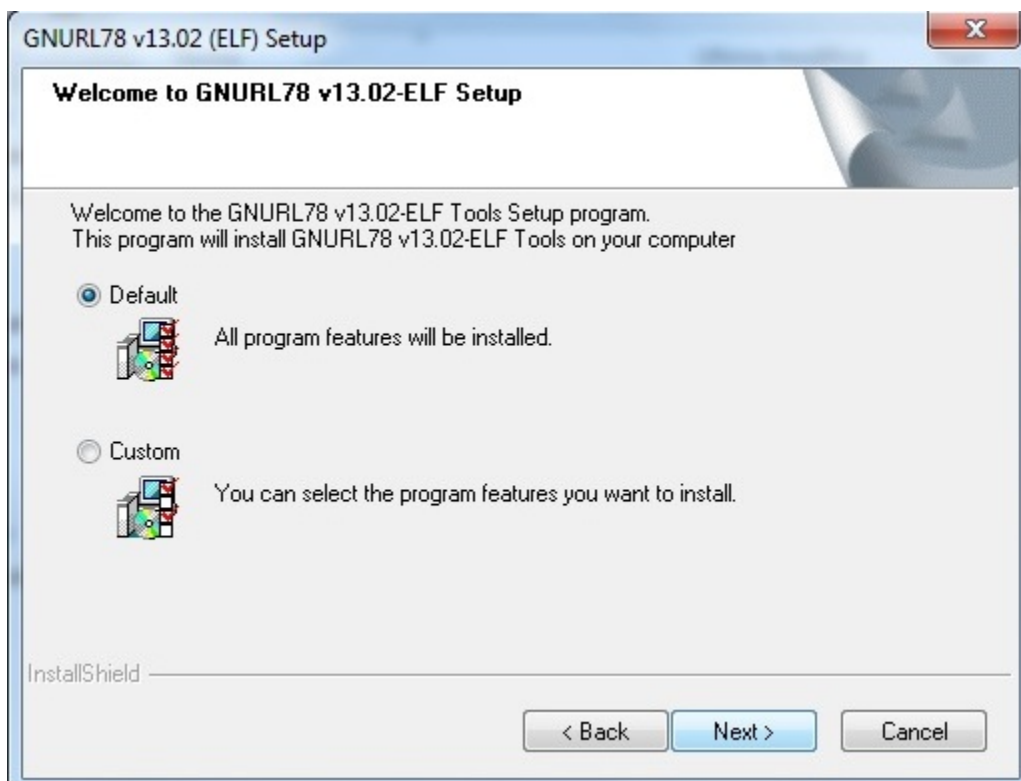
click "Finish" to end wizard.

Installing GNURL78 KPIT compiler

Double click on **GNURL78v1302-ELF-MP1.exe**.

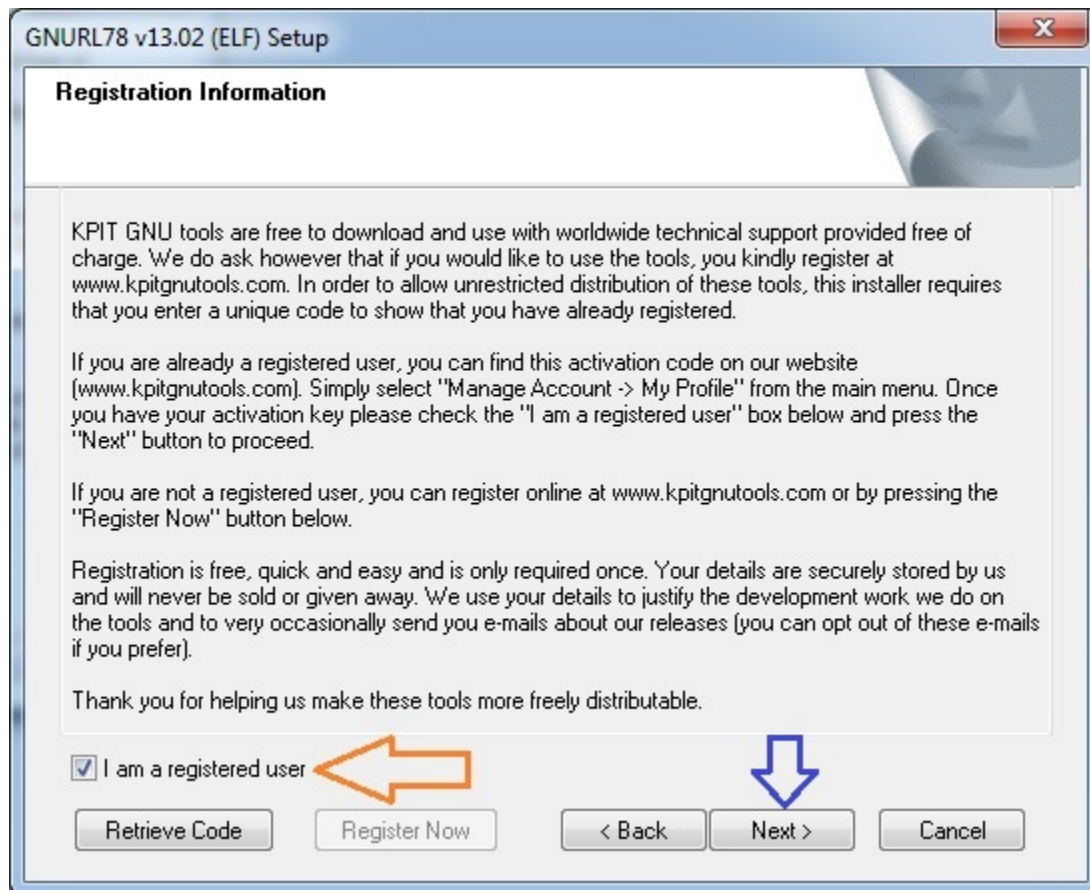


When wizard starts, a first popup will appear

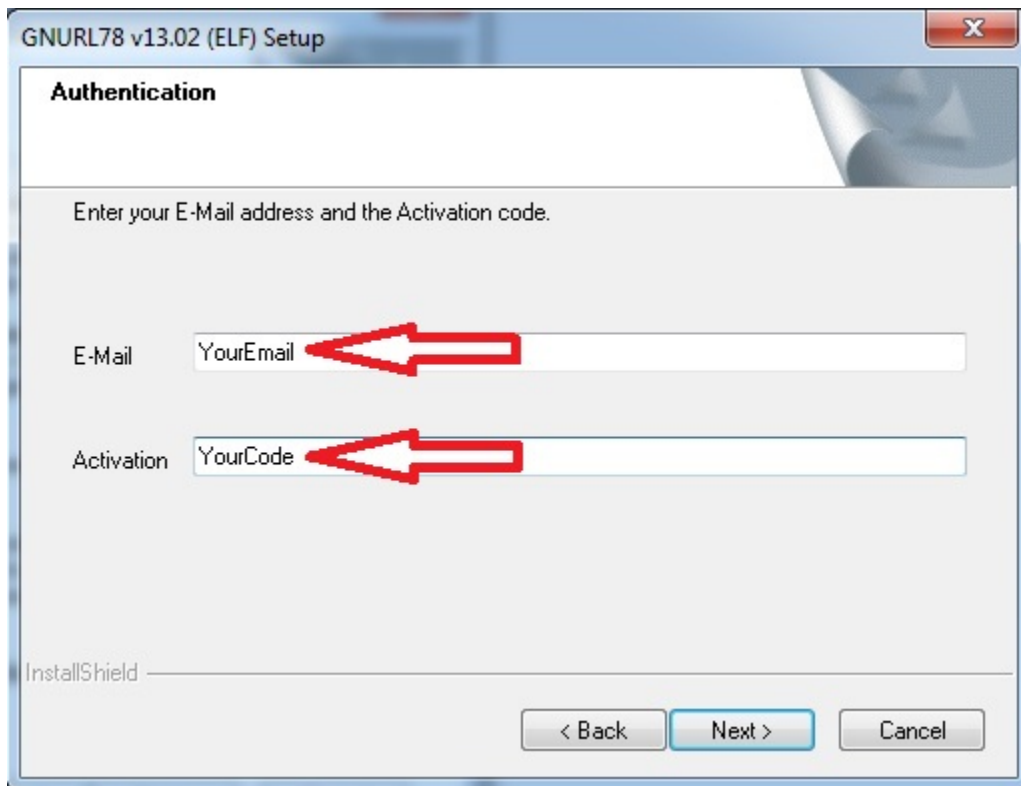


Leave unchanged "Default" radio button and click "OK"

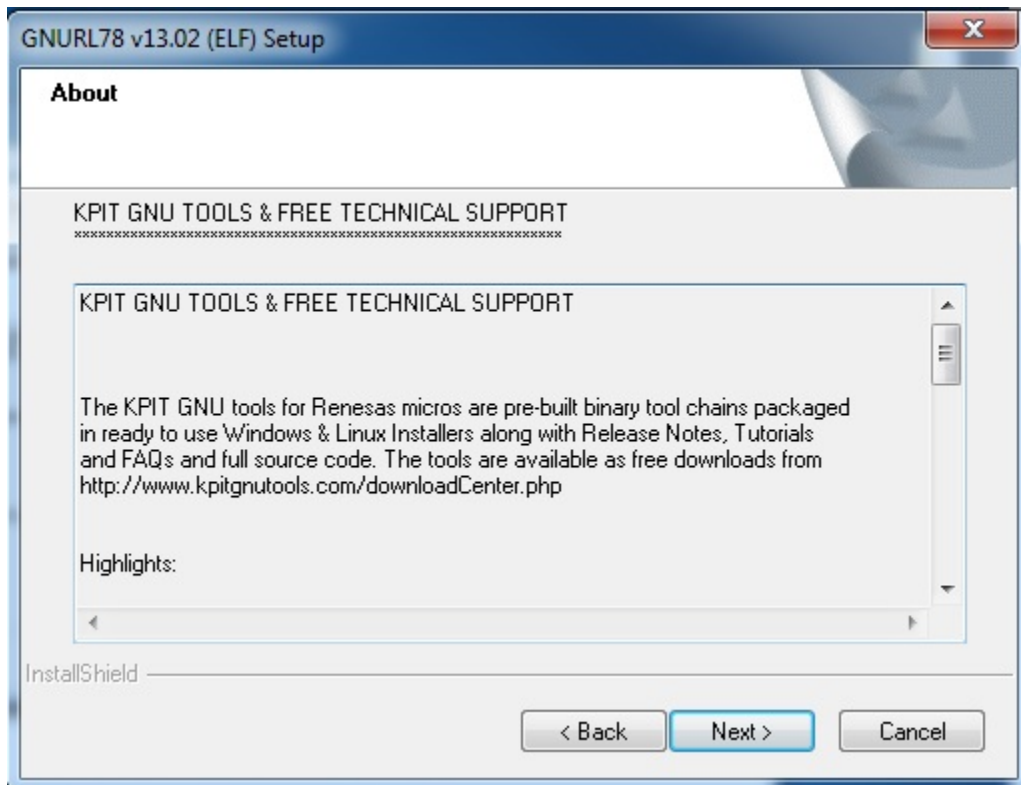
Important: The steps following are needed only first time you install GNURL78 compiler. The data needed for activate license are in the email received after registration :*GNURL78 registration email*



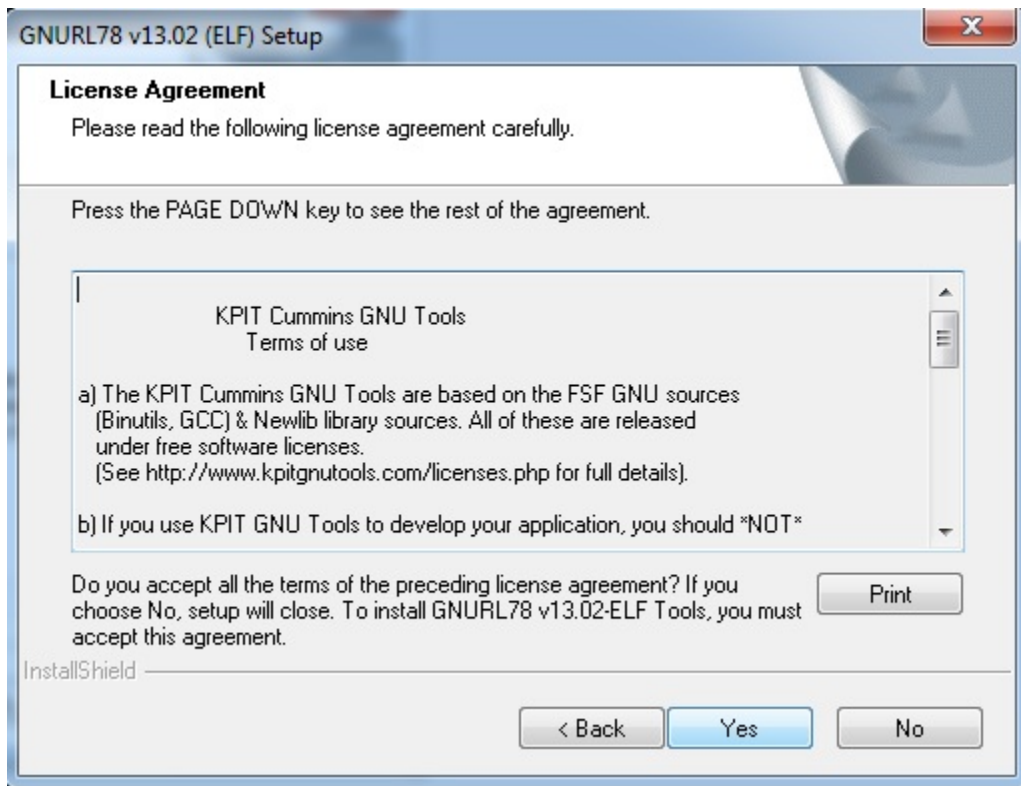
GNURL78 wizard now ask for activation code. Check "I am a registered user", then click "Next" button and take care at registration email.



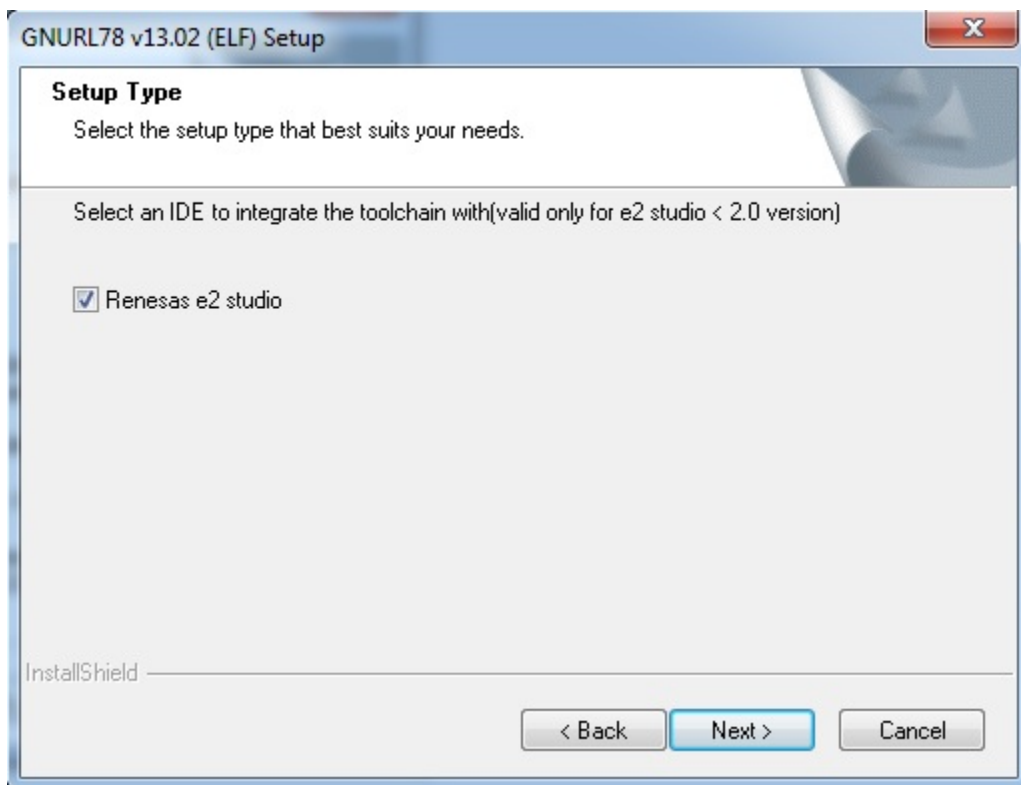
Insert YourEmail and YourCode as received in GNURL78 registration email and then click “Next” button



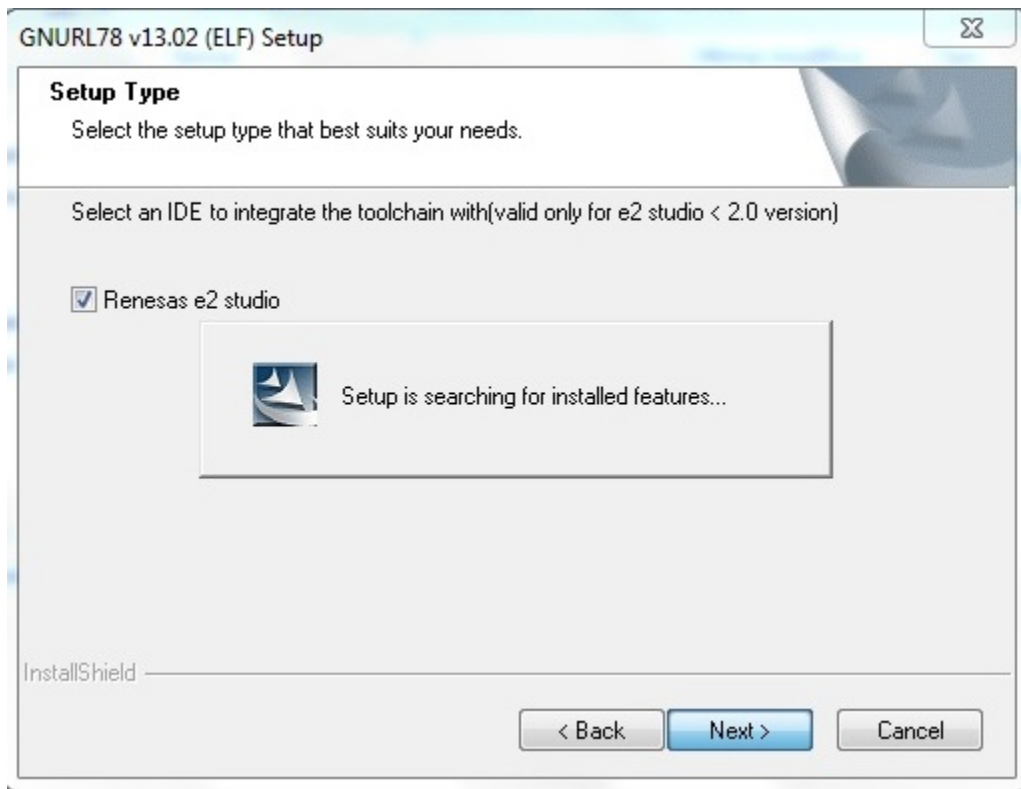
Click “Next” button



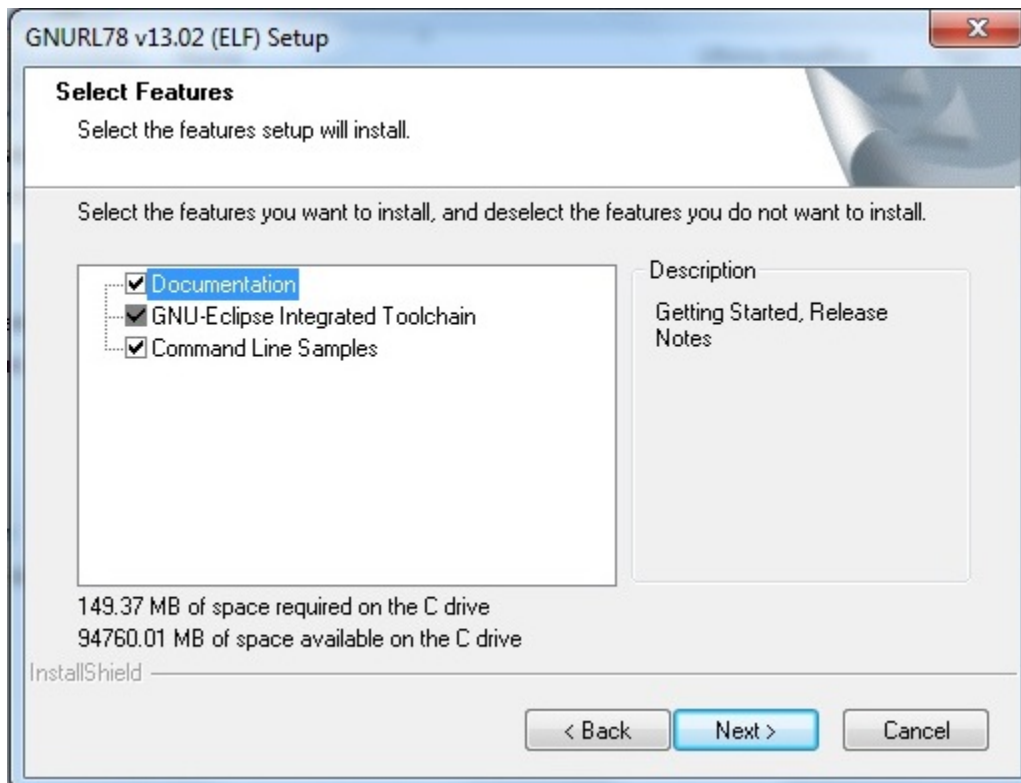
Click "Yes" button to agree License



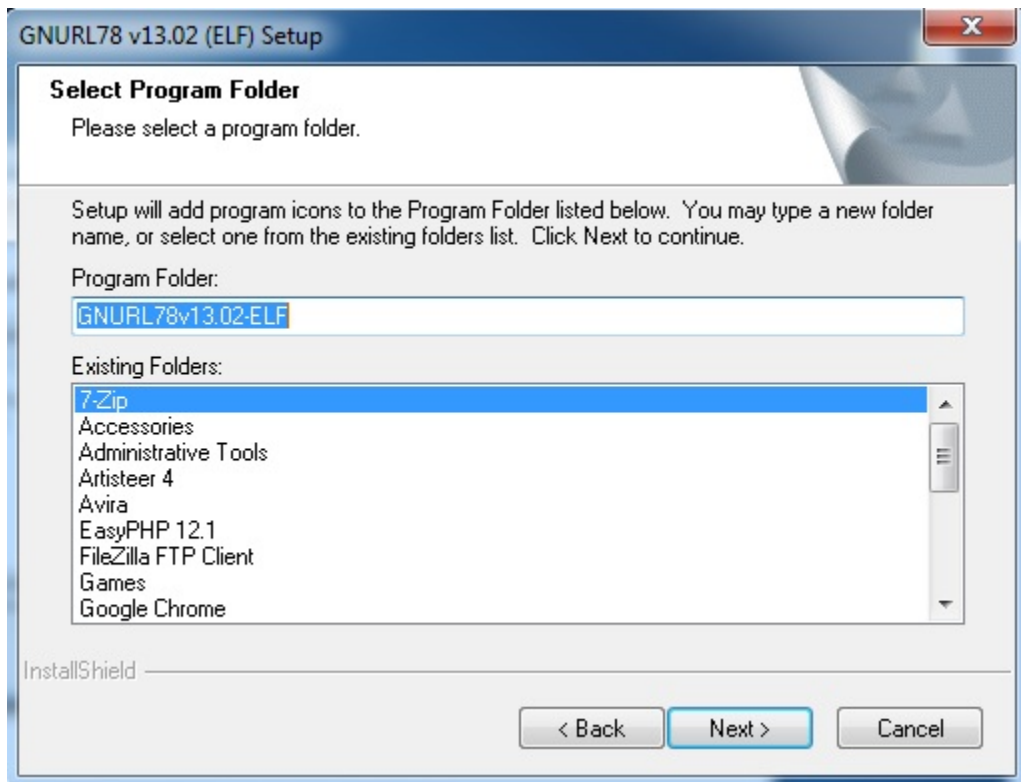
GNURL78 wizard ask for confirm E2studio toolchain integration. Check and click "Next" button



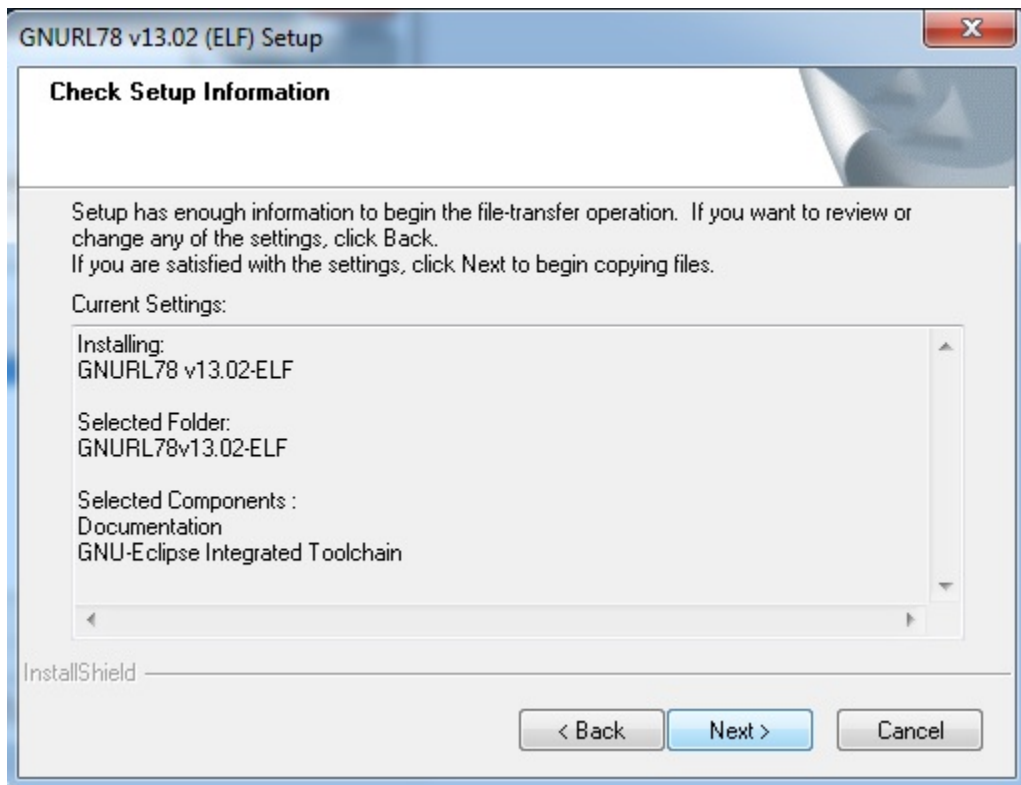
Wait for wizard ...



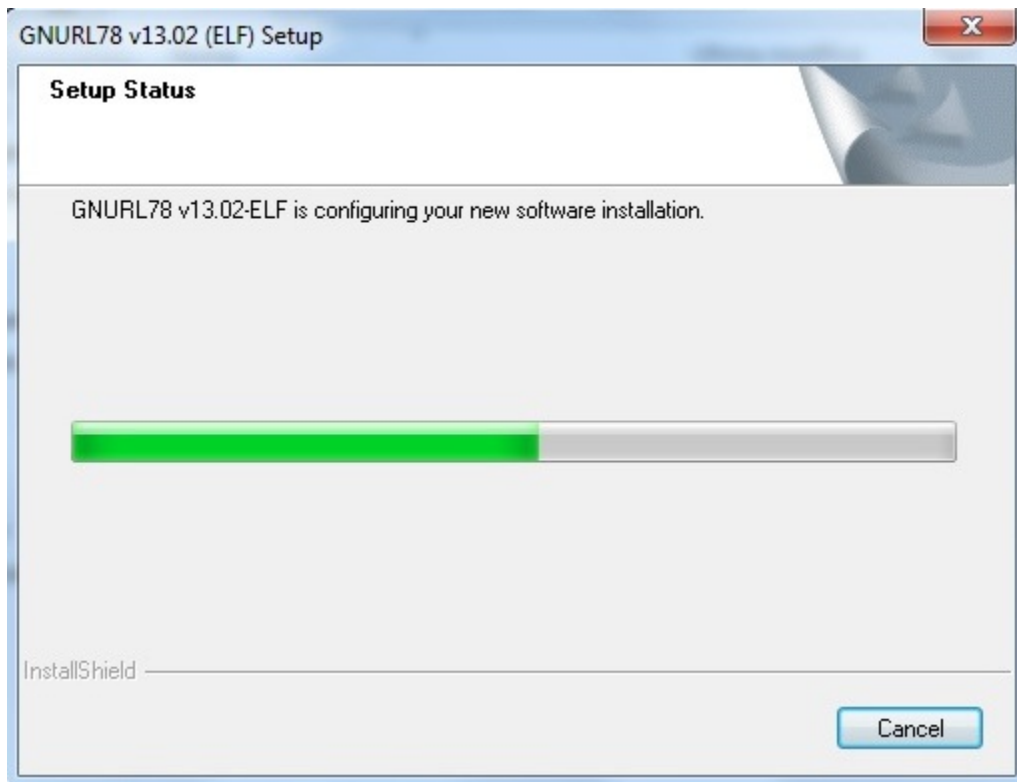
GNURL78 features settings. Set as image above then click “Next” button



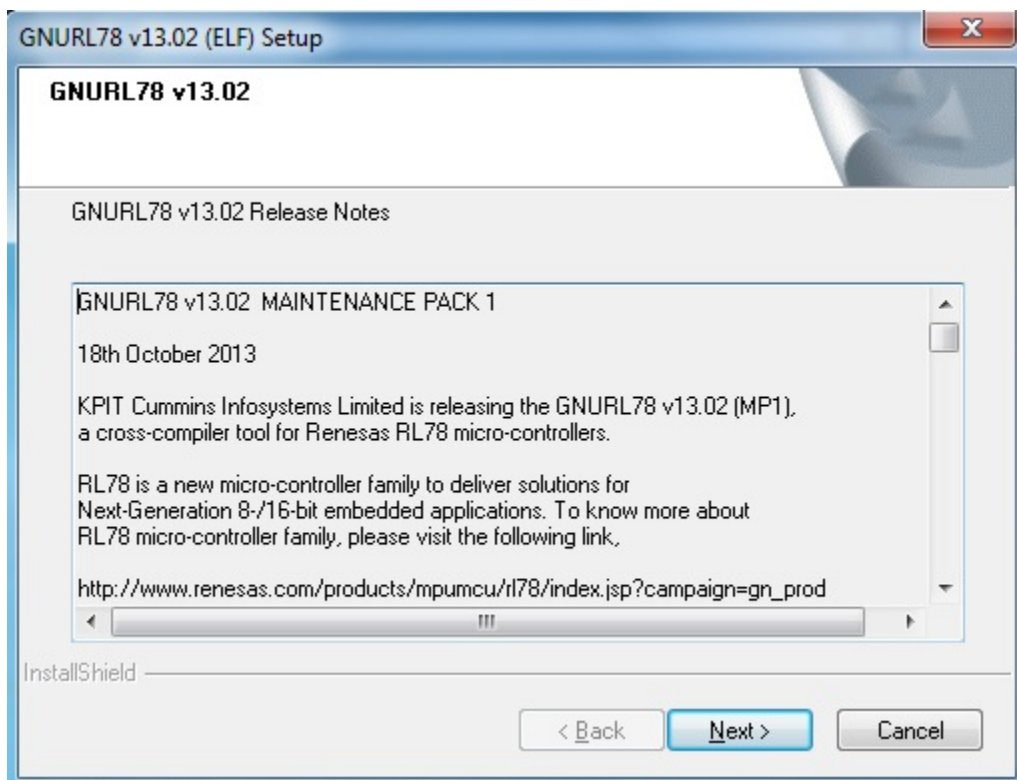
GNURL78 install folder. Leave unchanged then click “Next” button



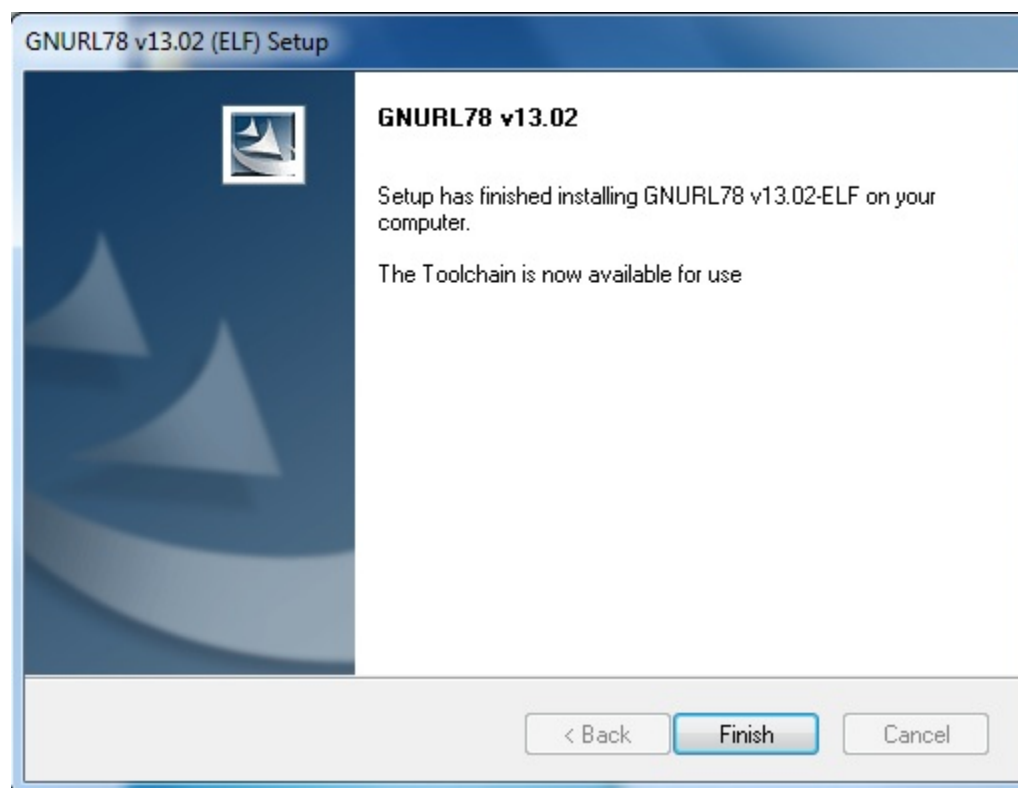
GNURL78 settings summary. Click “Next” button



Wait for GNURL78 integration inside E2studio suite



Activation of GNURL78 Maintenance Pack 1. Click “Next” button



GNURL78 compiler successfull installed and configured for E2studio suite!






Firmware specification

Firmware overview

The Liberty_BLE_v1 firmware has been developed starting from **VSSPP_ML610QXXX_Ver0_094_beta** firmware revision developed from Lapis to use with your PEXL7105_WSN kit.

This “beta” revision include basic BLE stack, an example of GATT database and some function that may be usefull for end-user application development.

More detail regarding ML7105 application and “beta” firmware can be found in “BLE_doc\ML7105_kit” folder included in Liberty_doc.zip archive file

Name	Date modified	Type	Size
 FEXL7105_WSNapp-01.pdf	25-10-2013 Fri 12:46	Documento Adob...	187 KB
 PEXL7105_VSP_IF-01.pdf	31-10-2013 Thu 08...	Documento Adob...	489 KB
 PEXL7105_VSSPP-UG-01.pdf	28-11-2013 Thu 17...	Documento Adob...	244 KB
 PEXL7105_WSN_HardManual-03.pdf	26-03-2014 Wed 1...	Documento Adob...	1.761 KB
 readme.txt	27-11-2013 Wed 1...	Text Document	2 KB

Other documents:

FEXL7105_BACI_Manual-01.pdf — ML7105 Baci reference guide

FEXL7105_AppDevelopersGuide-01.pdf — Summary of basic BLE stack functions

Firmware restrictions

The Libert_BLE_v1 firmware assumes the following restrictions:

- It is designed and configured only for VSP application.
- No testing has been made for VSSPP application.
- No very Low Power mode available. Liberty low-power mode is only for evaluation
- Support few examples of ATT and GATT database (only basic tables)
- There will be no security implemented.

Main firmware features

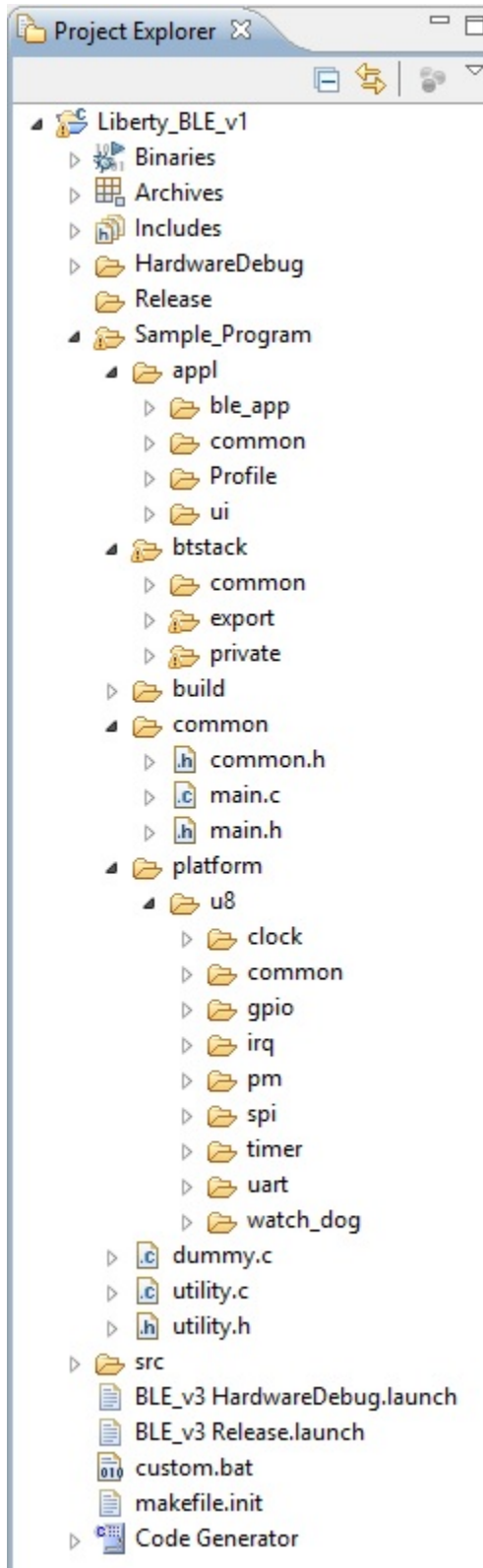
Liberty_BLE_v1 firmware is designed for demo application. It's also possible, starting from it, to develop user's application. In this case, debug and testing are charged to the end user.

By using PC or SmartPhone Liberty Software, you can perform:

- Reading or writing characteristic handle
- Discover and pair device
- Read sensor's data stored in GATT database
- Read sensor's data immediately (real time value)
- ON/OFF blue led

Liberty firmware also “dump” continuously the communication between Renesas CPU and ML7105 module. Please refer to *Hardware setup* in “Quick Start Guide”

Project structure



The application's file structure includes the following folders:

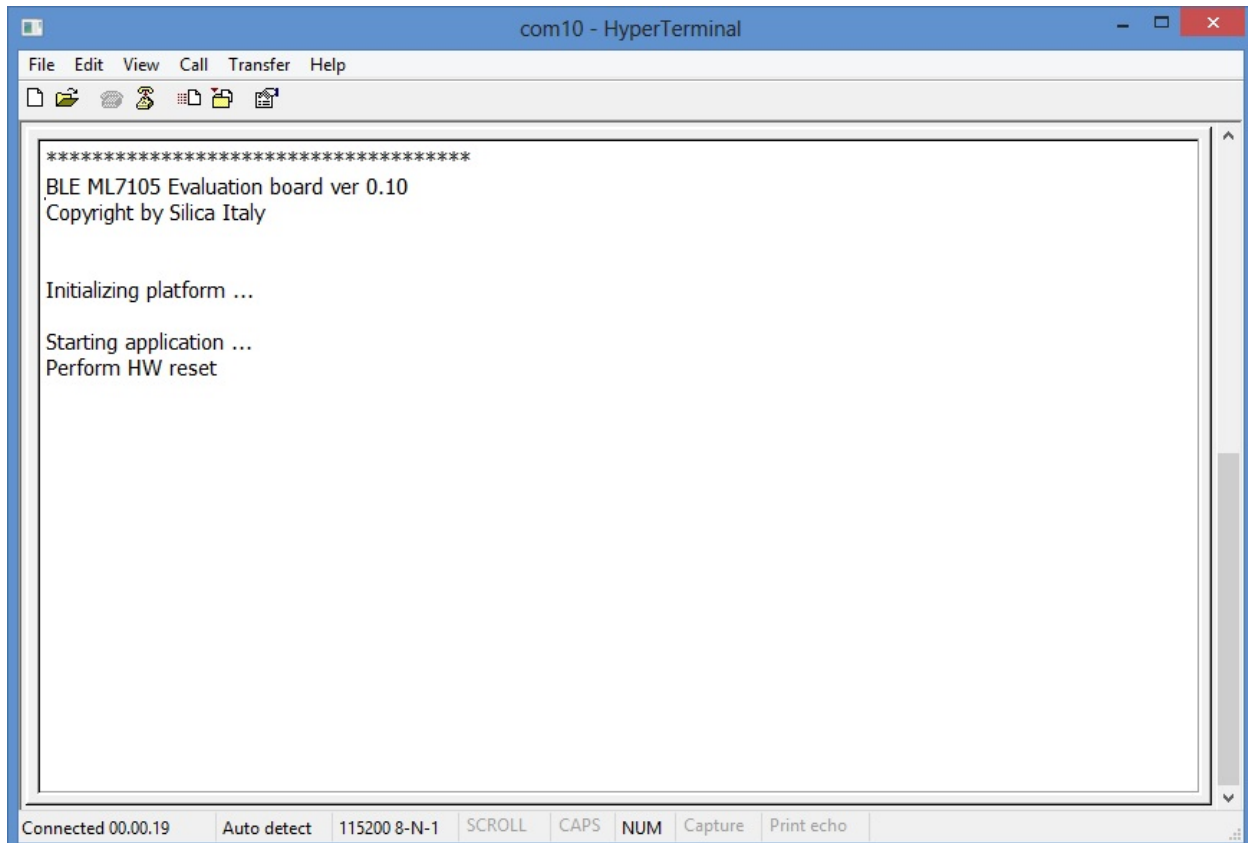
- **Sample_Program** – contains source files imported from VSSPP_ML610QXXX_Ver0_094_beta
 - **appl** specific BLE application file
 - **btstack** specific BLE stack management application file
 - **common** BLE application main file and common definition
 - **platform** specific Platform application file: U8 suffix original file are adapted fo Renesas CCPU
- **src** – Renesas RL78 Hardware setup and driver (generated by Code Generator integrated in E2studio)

Brief of firmware performance

The firmware will include a minimal example to initialize the ML7105 and functions to store periodically (every 30 seconds) the sensors data and to perform pairing with PC or SmartPhone. When paired, using a PC or SmartPhone Liberty Software application, it's also possible to read/write both Characteristic Value Handle and Characteristic UUID, set/clear periodically notify (only for Accelerometer data), read Temperature value and Accelerometer data (periodically stored by 30 seconds internal timer) or make a flash read of data from sensors (Temperature and Accelerometer). It's also possible (writing into reserved Characteristic Value Handle) turn on and turn off the blue LED on the board.

Using “dump” you can see and store all BACI packet between RL78 cpu and ML7105 BLE module, and you can analyze much more detail about Liberty Board state.

When board startup, in the terminal windows you can see



```

*****
BLE ML7105 Evaluation board ver 0.10
Copyright by Silica Italy

Initializing platform ...

Starting application ...
Perform HW reset
  
```

Connected 00.00.19 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo

The RED led will on, and in a few seconds start the initial settings communication. Here above the last BACI packets that end initial settings

```

t[01] r[ff] t[0c] r[ff] t[00] r[ff] t[ff] r[ff]
t[ff] r[04] t[ff] r[03] t[ff] r[0b] t[ff] r[0c] t[ff] r[00] t[ff] r[08] t[ff] r[06] t[ff] r[0b] t[ff] r[00] t[ff] r[06] t[ff] r[6a]
t[ff] r[00] t[ff] r[02] t[ff] r[14]

t[01] r[ff] t[0b] r[ff] t[01] r[ff] t[00] r[ff] t[ff] r[04] t[ff] r[03] t[ff] r[0f] t[ff] r[0b] t[ff] r[00] t[ff] r[0c] t[ff] r[d1] t[ff]
r[d2] t[ff] r[d3] t[ff] r[d3] t[ff] r[d2] t[ff] r[d1] t[ff] r[18] t[ff] r[a2] t[ff] r[a4] t[ff] r[d9] t[ff] r[9a] t[ff] r[c7]

t[01] r[ff] t[21] r[ff] t[06] r[ff] t[18] r[ff] t[a2] r[ff] t[a4] r[ff] t[d9] r[ff] t[9a] r[ff] t[c7] r[ff] t[ff] r[ff]

t[ff] r[04] t[ff] r[03] t[ff] r[03] t[ff] r[21] t[ff] r[00] t[ff] r[00]

t[01] r[ff] t[01] r[ff] t[00] r[ff] t[ff] r[ff]

t[ff] r[04] t[ff] r[01] t[ff] r[01] t[ff] r[80]

t[01] r[ff] t[23] r[ff] t[16] r[ff] t[00] r[ff] t[14] r[ff] t[02] r[ff] t[01] r[ff] t[06] r[ff] t[10] r[ff] t[08] r[ff] t[4c] r[ff]
t[41] r[ff] t[50] r[ff] t[49] r[ff] t[53] r[ff] t[20] r[ff] t[56] r[ff] t[53] r[ff] t[50] r[ff] t[32] r[ff] t[20] r[ff] t[57] r[ff]
t[53] r[ff] t[4e] r[ff] t[00] r[ff] t[ff] r[ff] t[01] r[ff] t[0f] r[ff] t[10] r[ff] t[01] r[ff] t[20] r[ff] t[00] r[ff] t[20] r[ff] t[00]
r[ff] t[00] r[ff] t[01] r[ff] t[00] r[ff] t[00] r[ff] t[00] r[ff] t[00] r[ff] t[00] r[ff] t[00] r[ff] t[00] r[ff] t[00] r[ff] t[07] r[ff] t[00] r[ff]
t[ff] r[ff]

```

t[hex value] means data from CPU to MODULE

r[hex value] means data from MODULE to CPU

The green led flash (50msec every 2 seconds) to indicate that Liberty is in standby state, waiting for connection or pairing. When in this state, every 180 seconds the firmware will reset thne ML7105 module and clear pairing information stored in the eeprom

If connection is established, the green led lamps regullary (0,5 seconds ON, 0,5 seconds OFF) to indicate that Liberty is in active state, but not yet paired. When in active state, after 180 seconds the firmware reset the module as when in standby state

The function called to perform this is:

**BLEutil_StartTimer(BLE_APP_TIM_PAIR_INFO_CLEAR, BLE_TIM_180S,
BLE_App_Pairing_Info_Clear_Timer_Handler)**

(see inside file *ble_app.c* at function *BLE_App_Fsm*)

When green led lighth on, Liberty is active and paired. The 180 seconds timer is disabled.

Note: Yellow led will flash when Liberty are in “unpaired state” (not yet cleared pairing informations stored in the module). In this state, connection or pairing requests are refused. You must wait for expire of pairing

timeout (about 30 seconds). The green led will return as in standby state and new connections are enabled.

Low-power evaluation

Liberty Firmware has a low-power mode capability (only for demo purpose) When compiling, the Macro define **LOWP_EN** is used to enable/disable auto low-power mode. If enabled, when a 60 seconds timer expires, CPU will enter in stop mode. The CPU will awake from ML7105 IRQ request.

Important: Low Power evaluation with $I_{\text{average}} < 80\mu\text{A}$ is available **ONLY WHEN ML7105 IS PAIRED**. The Low Power evaluation mode, if entered when ML7105 is unpaired, take a current I_{average} of about 2mA. This is because ML7105 (if not yet paired) enable power to radio reception every 5msec.

The maximum time before battery cutoff is about 2000 hours (whith device paired, receive and transmit never used)

Note: default setting for macro **LOWP_EN** is **TRUE** (Low-power enabled)

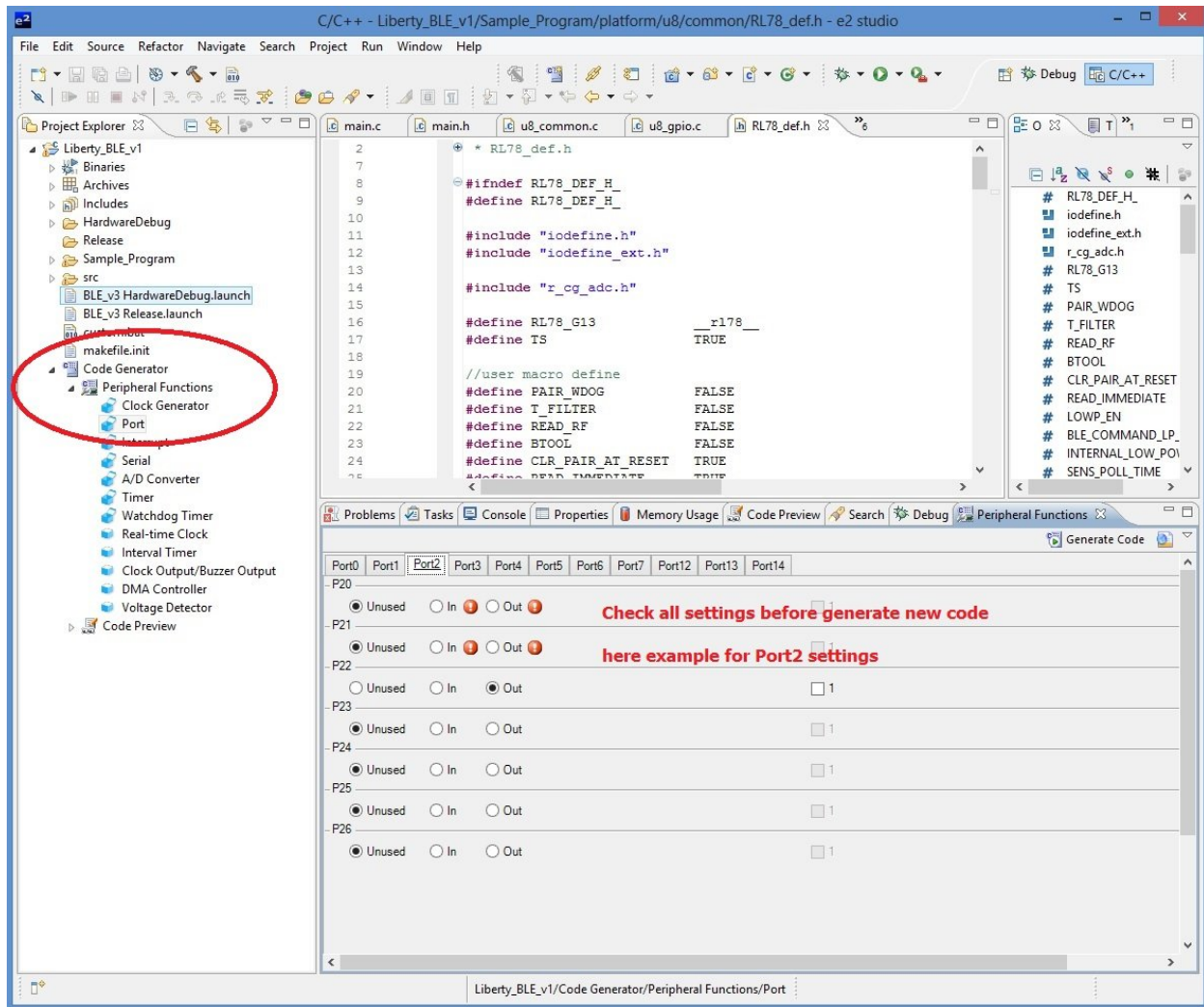
The 60 seconds low-power timer will recharge every time that ML7105 assert IRQ signal

Firmware & documents download

Firmware project file Liberty.zip and Liberty_doc.zip can be found at Silica ArchiTech page. Registration is needed to access at download section. Click [here](#) to go to ArchiTech main page.

Project tips

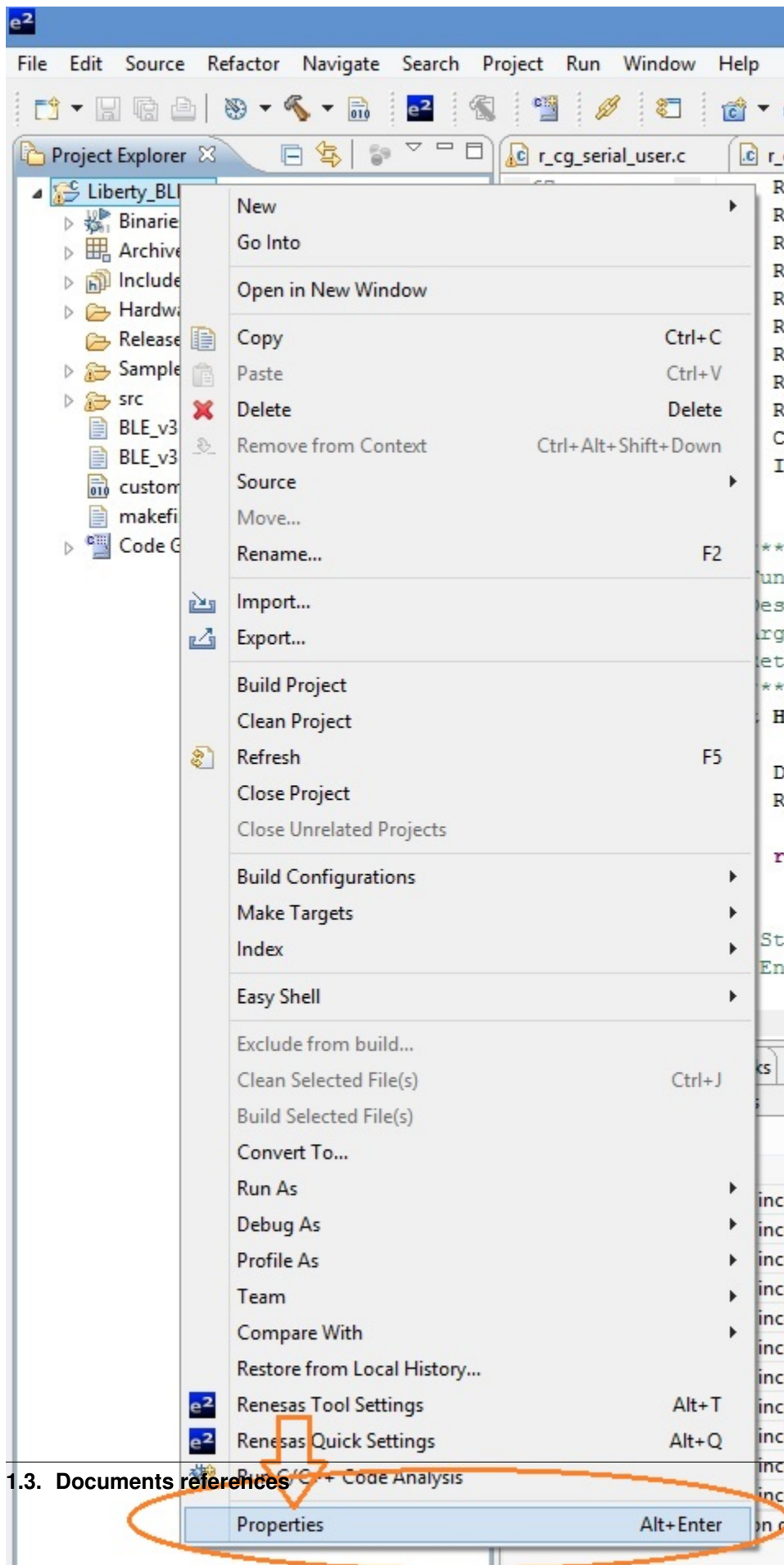
Important: Liberty_BLE_v1 firmware use “Code Generator” as integrated in E2studio suite to make source file for vector table, hardware driver and reset. For more detail about, read [Code Generator user’s manual](#) in Renesas official site



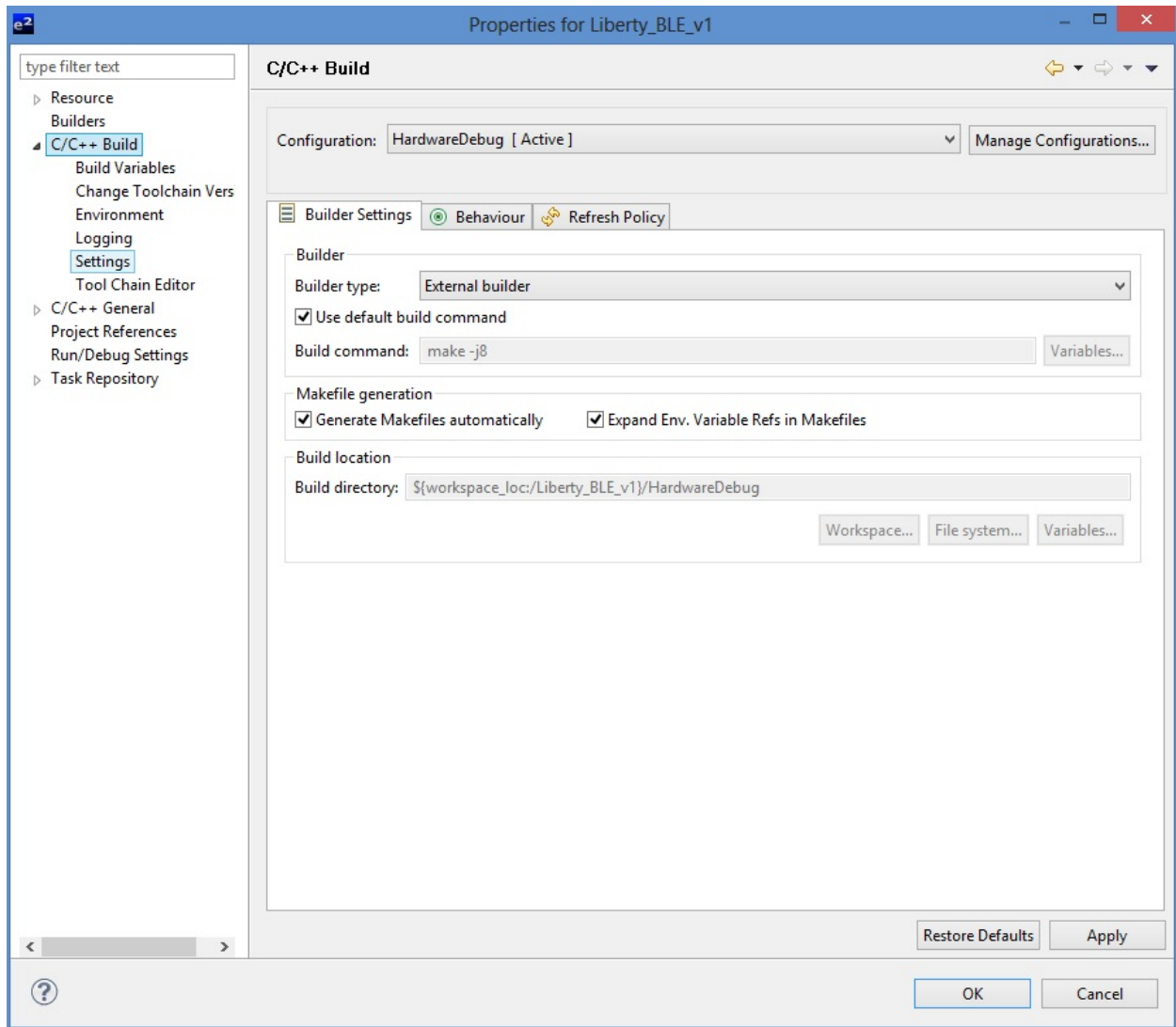
Important: Take care at changes on hardware driver using Code Generator → Peripheral Functions. (red circled on image above) It's important to check if all peripheral settings are correct for the actual project, before generate new code.

Project properties

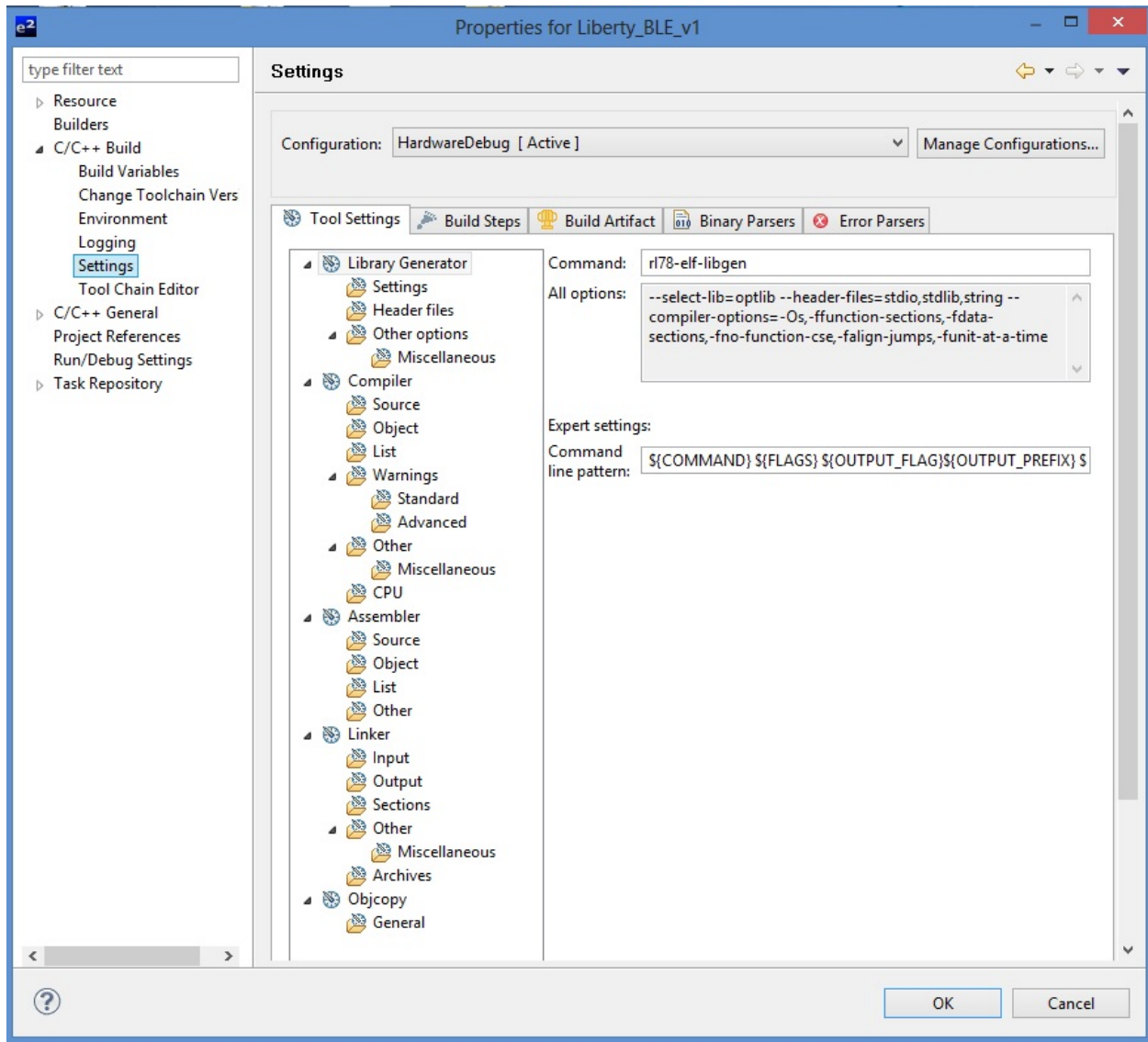
To access at project properties, right click on **Liberty_BLE_v1** on the Project Explorer window of E2studio suite



In the popup menu that appear, select and click on Property (orange circled in figure above)



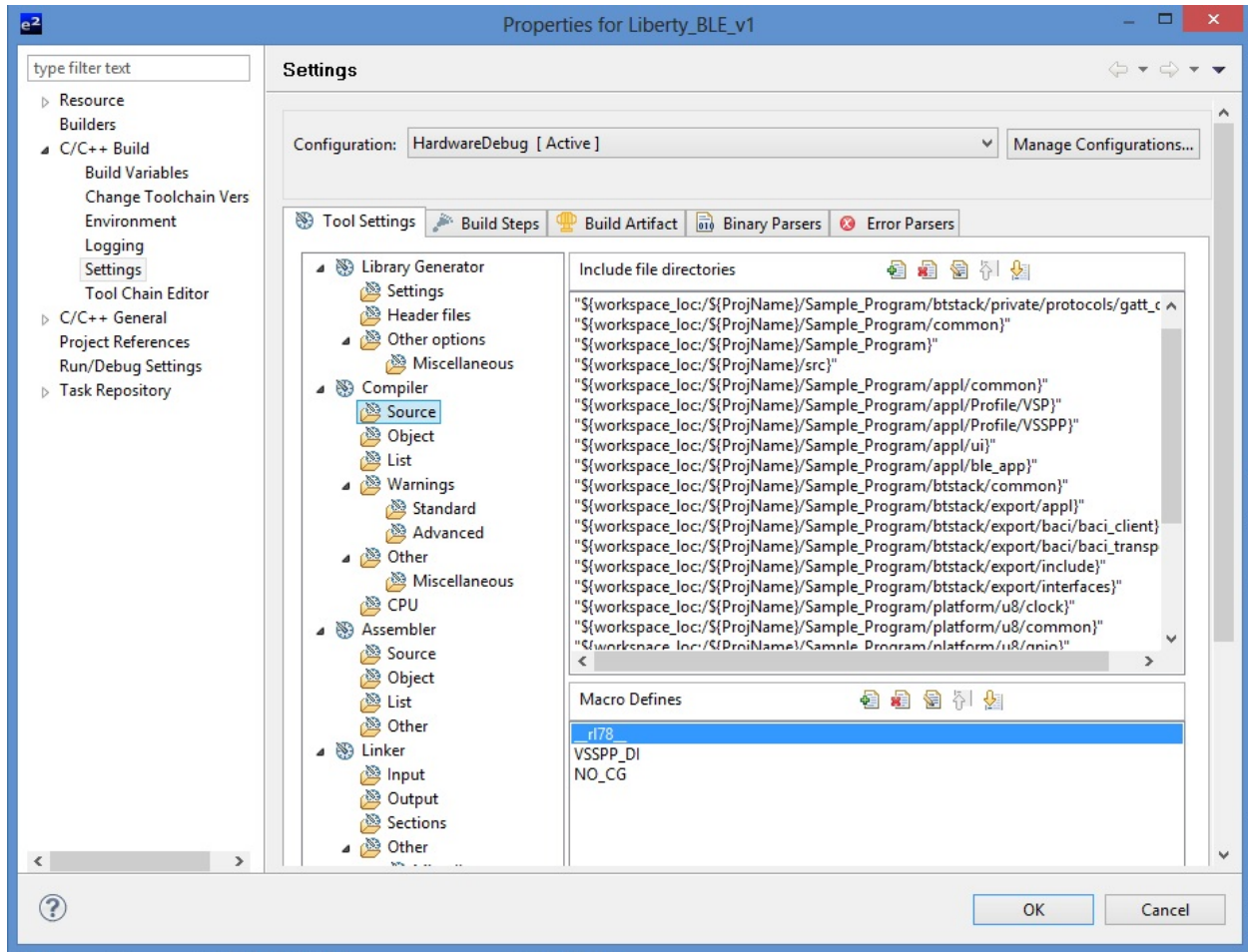
Expand **C/C++ Build** and click on **Settings**



Project settings window will appear. Now you are ready to check and change main project configurations

Macro defines

There are some macros defined in the project. The main macro defines are in the **Compiler**→**Source** settings



Details of macro

- **__r178__** macro defines: used to adapt hardware driver for Renesas Cpu. The source file inside **Platform** folder (all named U8_xxx) use this macro for hardware adapting. This macro is also used in some application files for the same scope.
- **VSSPP_DI** macro define: this macro disable all VSSPP functionality (not performed in this firmware revision)
- **NO_CG** macro define: Used only in source files automatically generated by “Code Generator”.

Note:

If you change setting of “Code Generator” the macro **NO_CG** can be erased. Please, before change, note your use inside files:

r_cg_adc.c

r_cg_serial_user.c

r_cg_serial.c

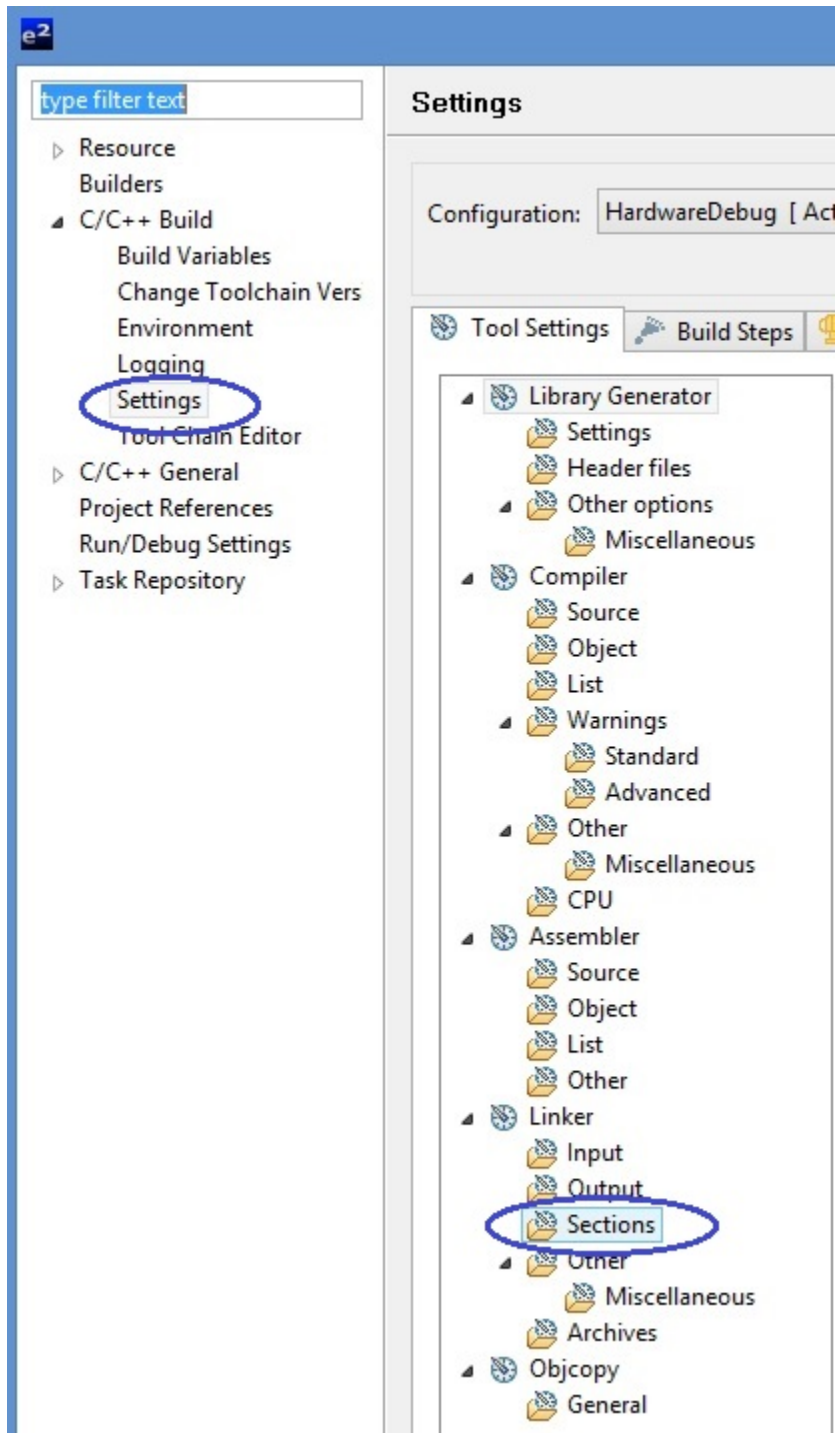
Other macros

Inside file **RL78_def.h** there are these user macro defines:

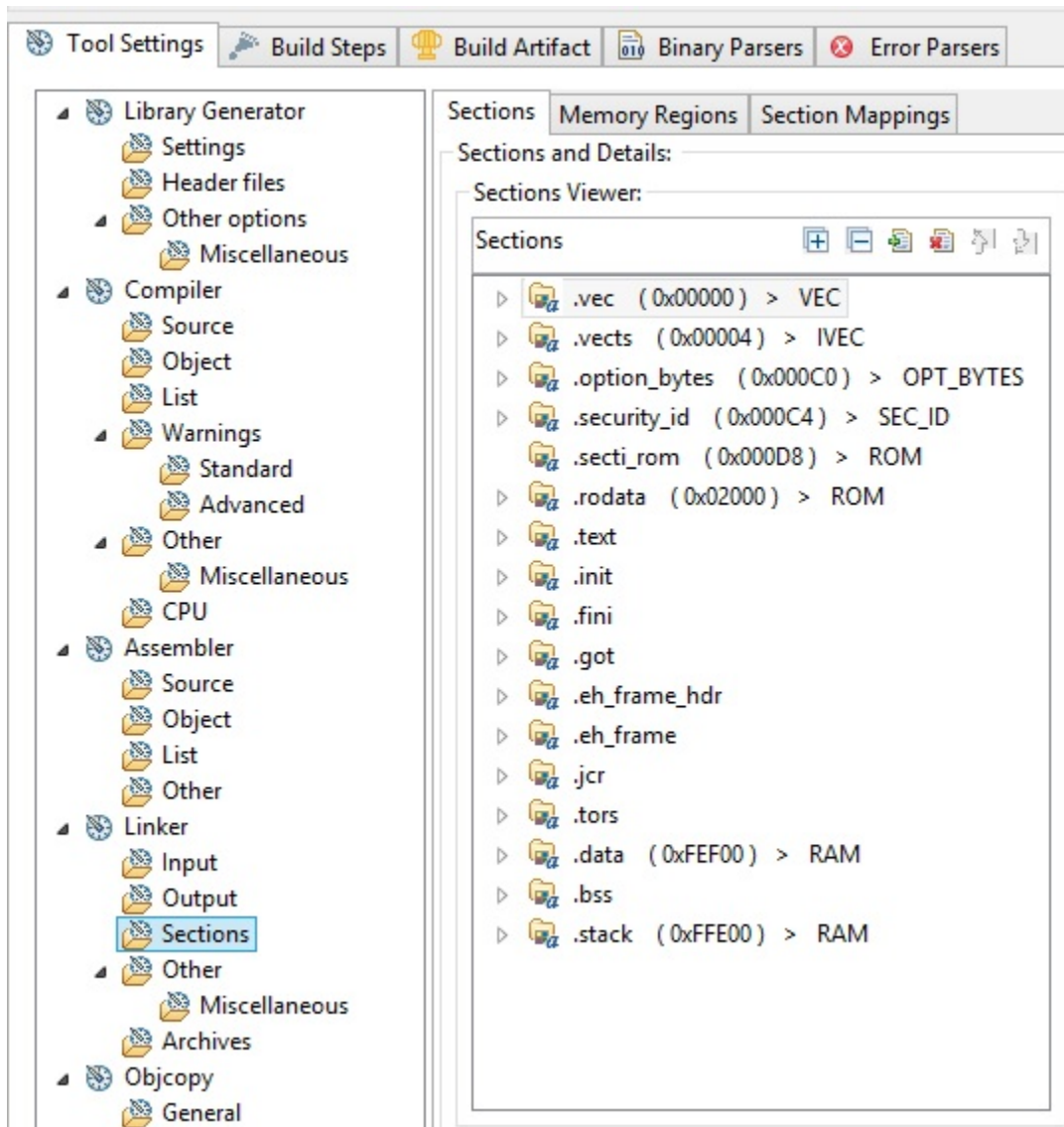
- #define PAIR_WDOG FALSE Used in BLE_App_Fsm function. If true, after 180 seconds timer expire, the system will wait for WatchDog reset. If false, the system performs a software initialization.
- #define T_FILTER FALSE enable/disable Temperature sensor average readings
- #define READ_RF FALSE enable/disable during initial configuration of ML7105 module of the external low power crystal oscillator.
- #define BTOOL FALSE only for testing purpose. Do not change
- #define CLR_PAIR_AT_RESET TRUE enable/disable automatically clear of pairing information stored in the eeprom every time that system is power_up or reseted by WatchDog.
- #define READ_IMMEDIATE TRUE If true, when external application ask for reading a properly UUID characteristic, the system perform a “flash reading of sensors value and send it as answer. If false, the system ask with last stored read.
- #define LOWP_EN TRUE enable/disable a demo feature of low power mode. This mode is available only when Liberty is in paired state

Sections settings

Go to the project property window (as described in [Project properties](#)) Then select Setting → Sections as blue circled in figure below



The **Sections** main tab will open

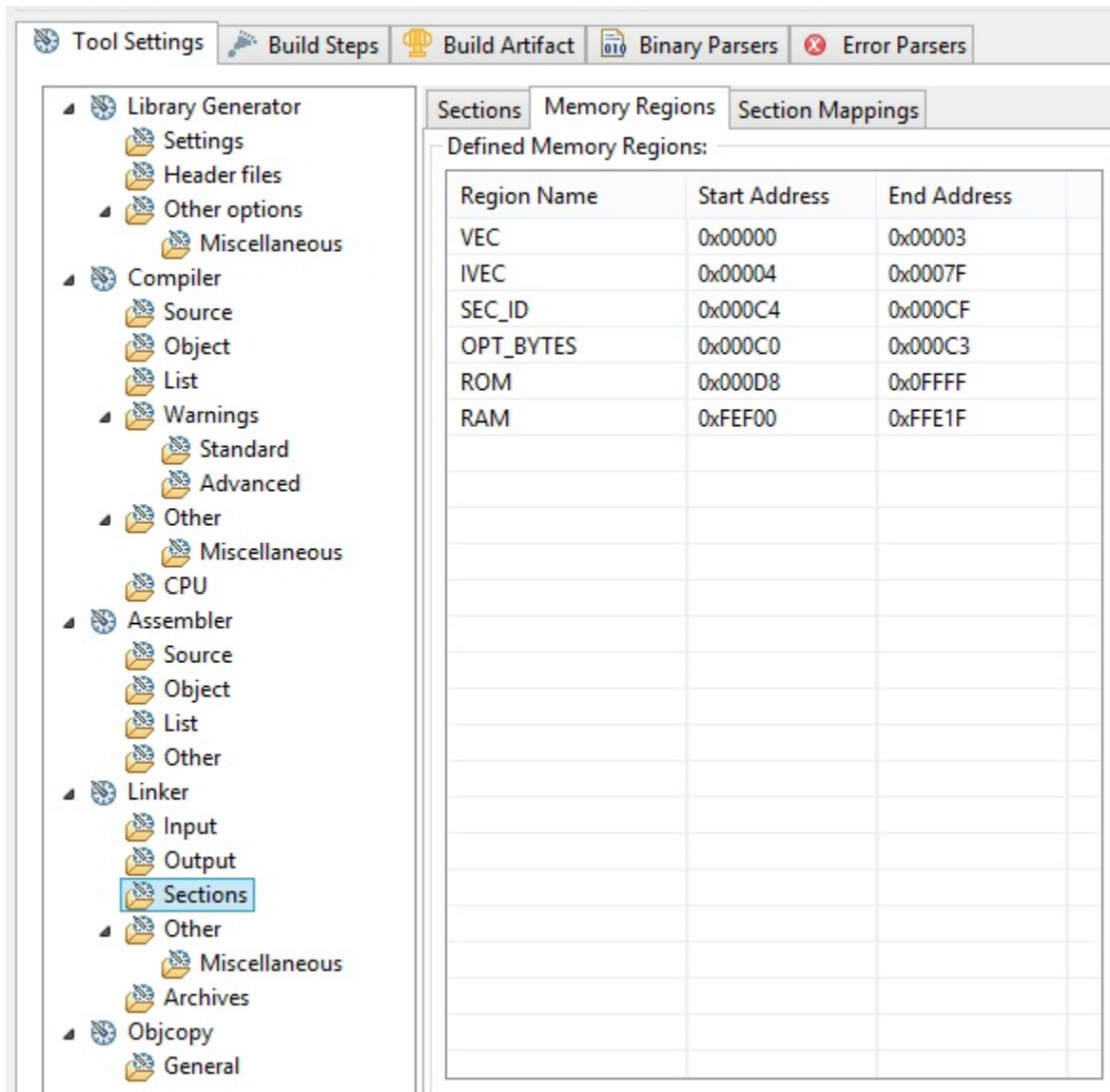


You can check if settings are the same as in figure above

Important: If you make changes in Code Generator and generate new code, the section mapping in this graphical interface is lost. However, the linker file is not changed and maintains correct settings. If needed, you can see the linker file contents [Linker file contents](#)

Memory region settings

As section settings, go to project property window, then select Settings → Sections and click on Memory Region tab



Note: The same effect will be done as in Important notice described in paragraph Section settings

Optimization

Some files in the project are optimized by code and size, and no debug information will be issued. Debuggin inside these files is not allowed. If you want it, you must change compiler setting. The files optimizer are listed in box below

```
Optimization level: Speed and code size
Debug level: none
```

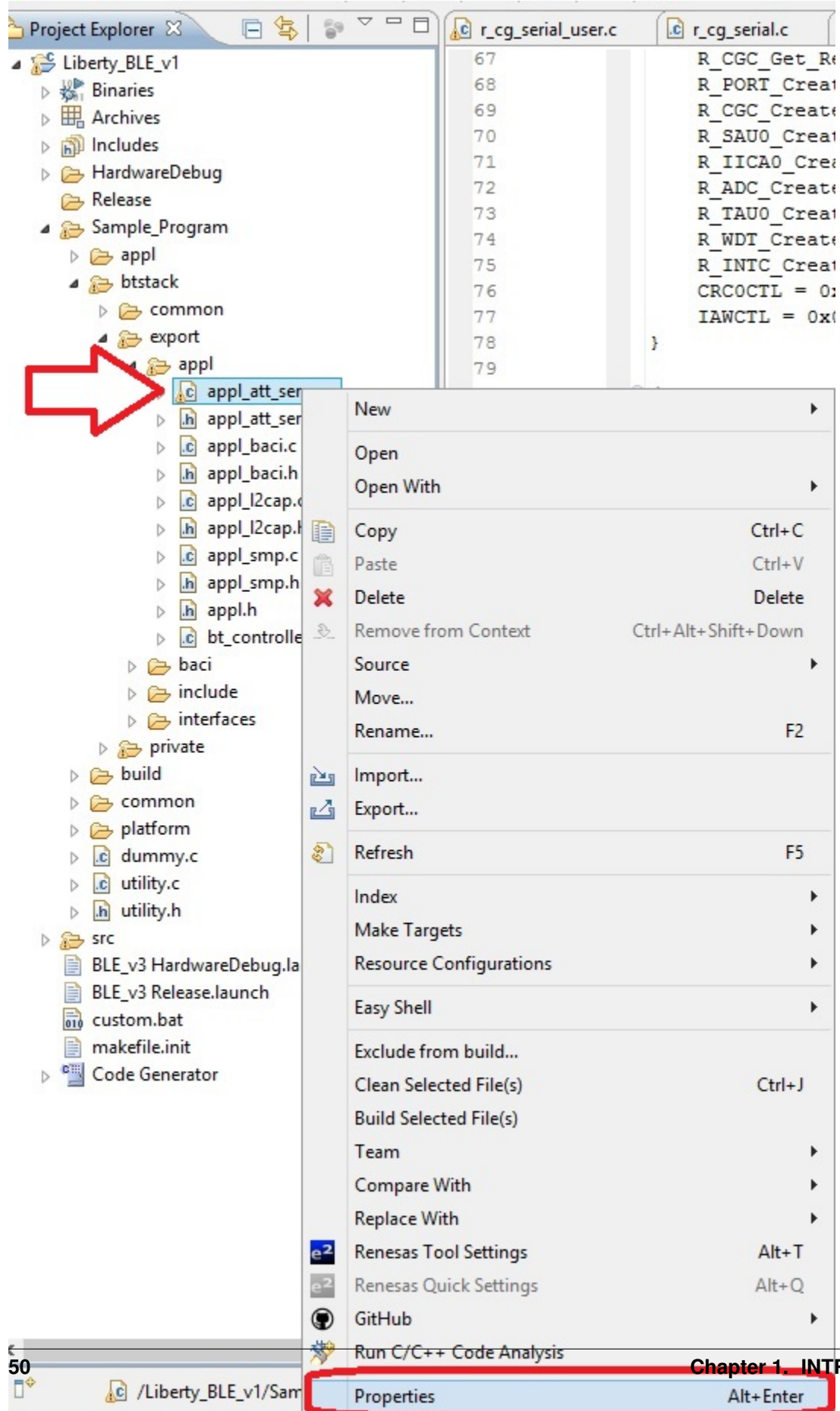
```

\Sample_Program\btstack\export\appl\appl_att_server.c
\Sample_Program\btstack\export\appl\appl_baci.c
\Sample_Program\btstack\export\appl\appl_smp.c
\Sample_Program\btstack\export\appl\appl_l2cap.c
\Sample_Program\btstack\export\baci\baci_transport\baci_transport.c

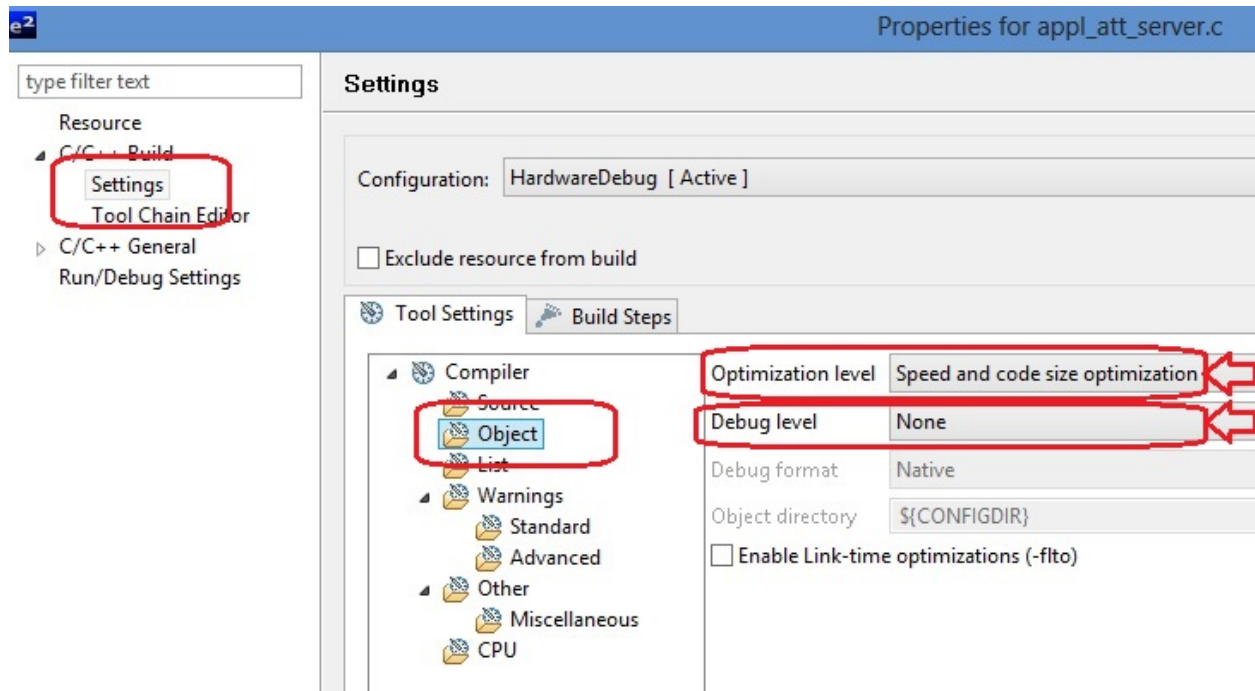
```

```
\Sample_Program\btstack\export\baci\baci_client\baci_client.c
```

To change optimization level, you must go to the file property

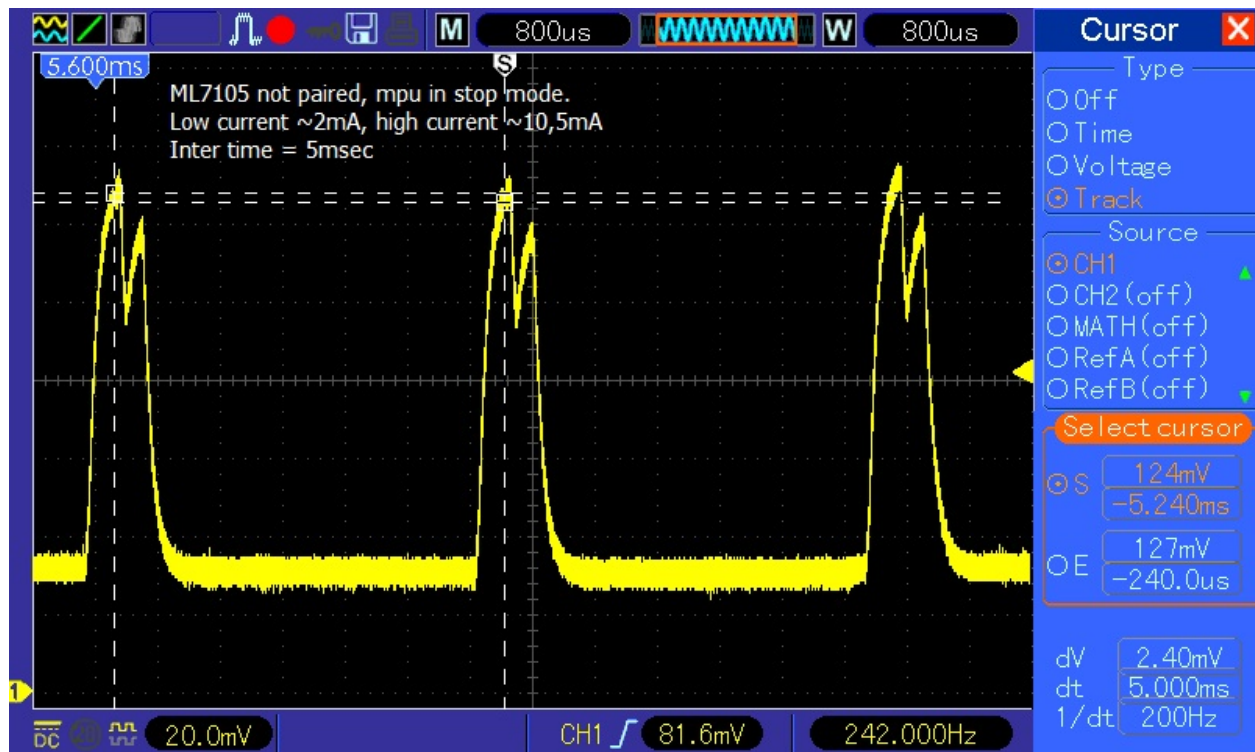


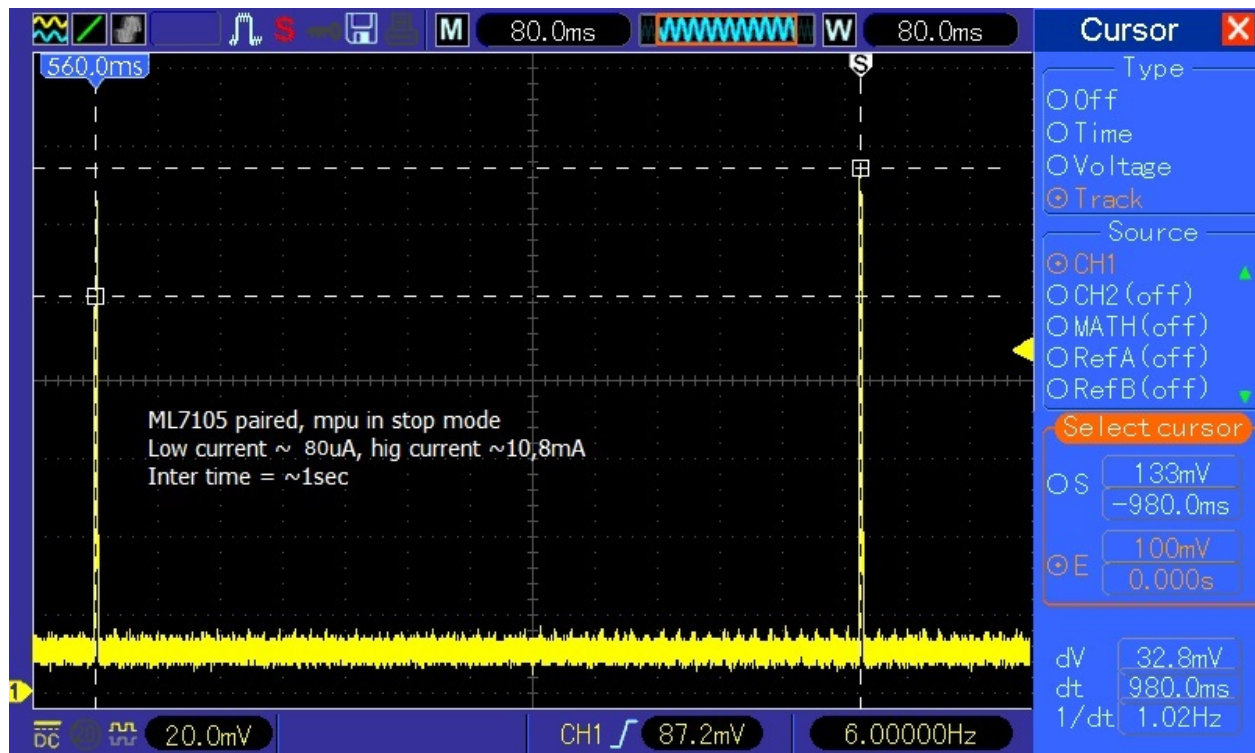
Right click on selected file, then click on Properties.



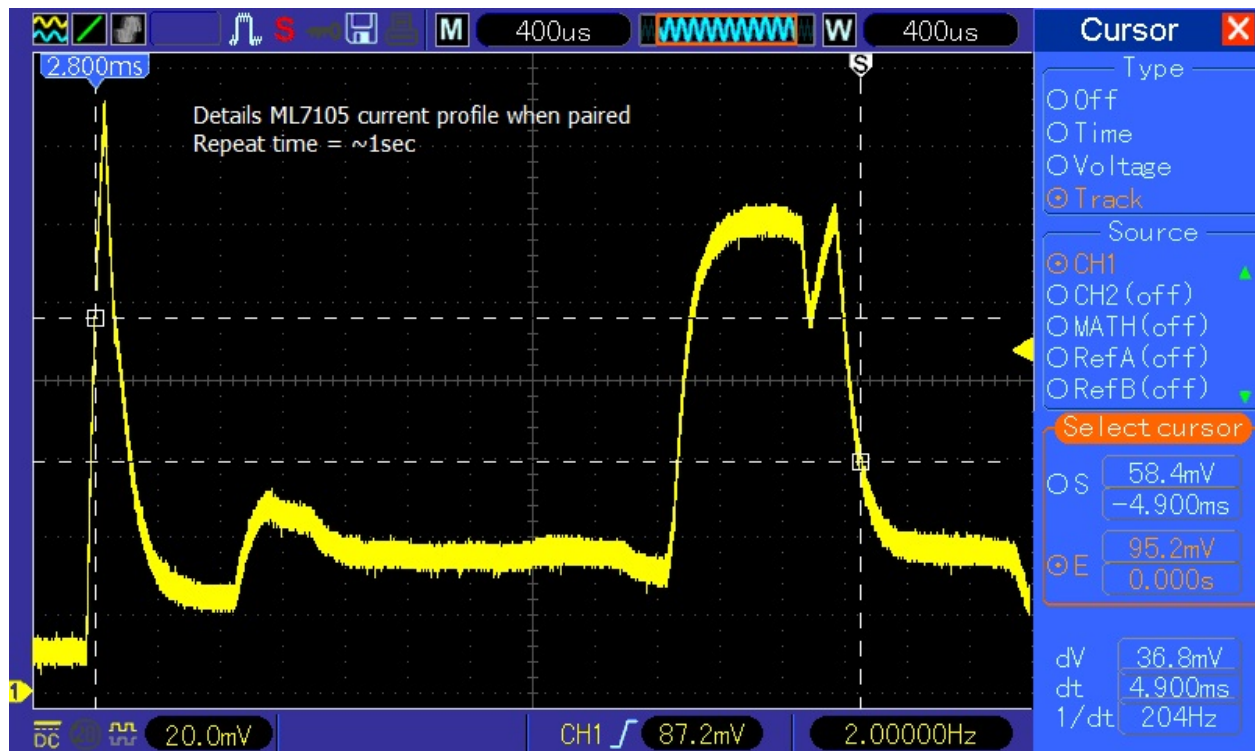
You can see optimization levels and change it (only for the selected file)

Low-Power current profile





here below a detail of the current profile during rx peak



Linker file contents

This is the contents of linker file named `Liberty_BLE_v1_HardwareDebug_auto.gsi`


```

MEMORY
{
    VEC : ORIGIN = 0x0, LENGTH = 4
    IVEC : ORIGIN = 0x4, LENGTH = 124
    SEC_ID : ORIGIN = 0xC4, LENGTH = 12
    OPT_BYTES : ORIGIN = 0xC0, LENGTH = 4
    ROM : ORIGIN = 0xD8, LENGTH = 65320
    RAM : ORIGIN = 0xFEF00, LENGTH = 3872
}

SECTIONS
{
    .vec 0x0 : AT (0x0)
    {
        KEEP(*( .vec))
    } > VEC
    .vects 0x4 : AT (0x4)
    {
        KEEP(*( .vects))
    } > IVEC
    .option_bytes 0xC0 : AT (0xC0)
    {
        *(.option_bytes)
    } > OPT_BYTES
    .security_id 0xC4 : AT (0xC4)
    {
        *(.security_id)
    } > SEC_ID
    .secti_rom 0xD8 : AT (0xD8)
    {
    } > ROM
    .rodata 0x2000 : AT (0x2000)
    {
        . = ALIGN(2);
        *(.rodata)
        *(.rodata.*)
        _erodata = .;
    } > ROM
    .text :
    {
        *(.text)
        *(.text.*)
        etext = .;
        . = ALIGN(2);
    } > ROM
    .init :
    {
        *(.init)
    } > ROM
    .fini :
    {
        *(.fini)
    } > ROM
    .got :
    {
        *(.got)
        *(.got.plt)
    } > ROM

```

```

.eh_frame_hdr :
{
    *(.eh_frame_hdr)
} > ROM
.eh_frame :
{
    *(.eh_frame)
} > ROM
.jcr :
{
    *(.jcr)
} > ROM
.tors :
{
    __CTOR_LIST__ = .;
    . = ALIGN(2);
    __ctors = .;
    *(.ctors)
    __ctors_end = .;
    __CTOR_END__ = .;
    __DTOR_LIST__ = .;
    __dtors = .;
    *(.dtors)
    __dtors_end = .;
    __DTOR_END__ = .;
    . = ALIGN(2);
    _mdata = .;
} > ROM
.data 0xFFEF00 : AT (_mdata)
{
    . = ALIGN(2);
    _data = .;
    *(.data)
    *(.data.*)
    . = ALIGN(2);
    _edata = .;
} > RAM
.bss :
{
    . = ALIGN(2);
    _bss = .;
    *(.bss)
    *(.bss.***)
    . = ALIGN(2);
    *(COMMON)
    . = ALIGN(2);
    _ebss = .;
    _end = .;
} > RAM
.stack 0xFFEE00 (NOLOAD) : AT (0xFFEE00)
{
    _stack = .;
} > RAM
}

```

E

E2Inst, [19](#)

F

ftips, [39](#)

I

index, [1](#)

L

linker, [52](#)

P

PrjFiles, [34](#)