TD1208 AN0002 ECLIPSE GCC TOOLCHAIN

Disclaimer: The information in this document is provided in connection with Telecom Design products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Telecom Design products.

TELECOM DESIGN ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL TELECOM DESIGN BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF TELECOM DESIGN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Telecom Design makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Telecom Design does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Telecom Design products are not suitable for, and shall not be used in, automotive applications. Telecom Design products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2013 Telecom Design S.A. All rights reserved. Telecom Design®, logo and combinations thereof, are registered trademarks of Telecom Design S.A. SIGFOX™ is a trademark of SigFox S.A.. Other terms and product names may be trademarks of others.

1 Overview

This document provides instructions for installing and using an Eclipse/GCC toolchain for the Telecom Design TD1208 module.

1.1 Scope

As the TD1208 module provides access to its internal ARM® Cortex-M3 processor for developing applications, it is possible for a user or a third-party to integrate custom software and algorithms with the libraries required for accessing the SIGFOX™ radio network, thus allowing the best overall low-power performance and lower cost.

This guide focuses on the description of the installation of an Eclipse/GCC toolchain, providing step-by-step instructions.

1.2 Organization

Each section in this document covers a separate topic, organized as follow:

- Section 1 is this overview
- Section 2 contains the installation instructions

1.3 Relevant Documents

Additional information on the TD1208 module and on its dedicated evaluation board can be found in:

- *TD1208 Datasheet*
- *TD1202 EVB User's Guide*
- *TD1208 Reference Manual*

TABLE OF CONTENTS

Section **Page**

1 Overview 3
1.1 Scope 3
1.2 Organization 3
1.3 Relevant Documents 3
2 Installation Instructions 5
2.1 Eclipse Setup 5
2.2 Mentor Sourcery CodeBench Lite Edition 6
2.3 Optional J-Link SWD Debugger 8
2.4 GNU ARM Eclipse Plugin 9
2.5 Additional Eclipse Plugins 12

2 Installation Instructions

This section provides detailed instructions on how to install the Eclipse/GCC toolchain for the TD1202/8 modules.

Telecom Design decided to bring the integrated EFM32G210F128 ARM® Cortex-M3 processor built into this module available for external developments. This CPUs is an Energy Micro EFM32 Gecko families members which features fast and energy-friendly processing core, modern and efficient peripherals and large RAM/Flash memory spaces.

In order to alleviate all prerequisites, Telecom Design chose to use only free tools for software development.

For the compiler toolchain itself, the size-limited IAR KickStart compiler was chosen over the free GCC compiler chain for the EFM32TG210F32, because of its established better architecture support and efficiency for generating ARM® Cortex-M3 machine code, saving up to 40% of program memory space. As the 32 KB size limitation matches exactly the total amount of memory available in the TD1202 EFM32TG210F32 CPU, this means that practically no size restriction applies. However, this choice implies that developments must be performed on a Windows platform, as this compiler toolchain is not available natively on OS X or Linux.

The separate “Eclipse IDE for C/C++” was preferred over the integrated “IAR Embedded Workbench®” because of its large industry acceptance and its flexible plugin architecture allowing an easy customization.

The installation of an additional Eclipse/IAR plugin is required to integrate the compiler toolchain into the IDE.

However, for the larger Flash memory capacity EFM32G210F128 CPU, the size-limited IAR KickStart compiler can no longer be used for free, and we need to use GCC as the only free alternative for generating code up to 128 KB, despite its less optimized Flash memory code generation. This should not be a problem though, because in this case the available Flash memory amount is much larger.

2.1 Eclipse Setup

The Java Runtime Environment (version 6 or later) is required in order to run the Eclipse software. It can be downloaded from:

http://java.com/

The latest release of the Eclipse IDE can be downloaded directly from the Eclipse Web site:

http://www.eclipse.org/downloads/

Unless you are interested into developments using other software languages than C/C++ or have an already existing Eclipse setup, the default bundle to choose is the “Eclipse IDE for C/C++ Developers”. At the time of this writing, the last version is nicknamed “Juno SR2”. Please select the package appropriate for your platform (Windows 32 or 64-bit).

The 130 MB downloaded file is a simple “ZIP” archive without any automatic installer, so once obtained, you must unzip the archive into a location of your choice. But as file names containing white space characters may cause problems with some plugins, we recommend using the standard location “C:\”, i.e. directly at the root of your main hard disk partition.

A directory “eclipse” is created into the chosen directory that contains the full Eclipse distribution.

2.2 Mentor Sourcery CodeBench Lite Edition

The latest release of the Mentor Sourcery CodeBench Lite Edition containing the compiler toolchain can be downloaded directly from the Mentor Web site:

http://www.mentor.com/embedded-software/sourcery-tools/sourcery-codebench/editions/lite-edition/arm-eabi

At the time of this writing, the last version is 2012.09-63. In order to proceed to download, you will need to register with Mentor if this is not already the case.

The 100+ MB file contains an automatic installer, that will first extract installation files into a temporary location, then display the following dialog window:

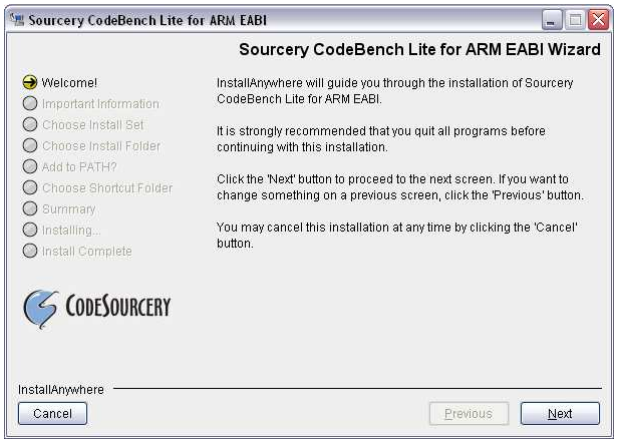

Figure 1- Sourcery CodeBench Welcome Screen

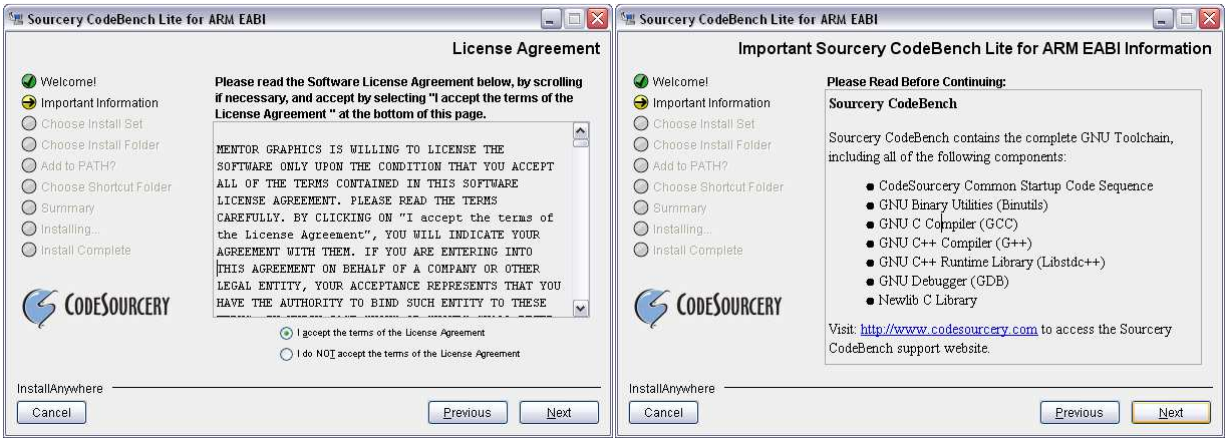

Figure 2- CodeBench License Agreement & Information

Click "**Next**", then check the "**I accept the terms of the license agreement**" radio button and click "**Next**" twice.

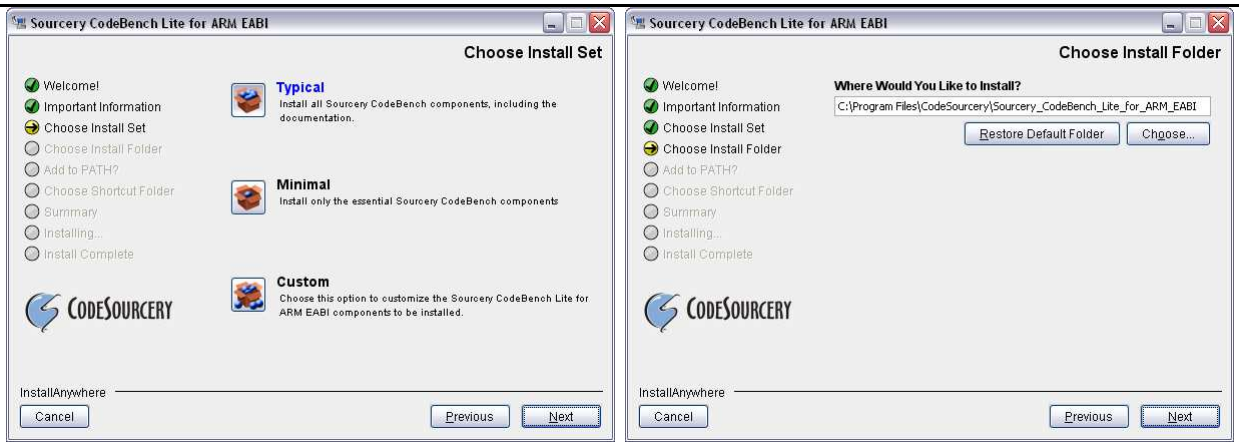

Figure 3 – CodeBench Install Set & Destination Folder Selection

Select the "**Typical**" install set and click on "**Next**", then choose the destination location or leave it as proposed and click "**Next**" again.

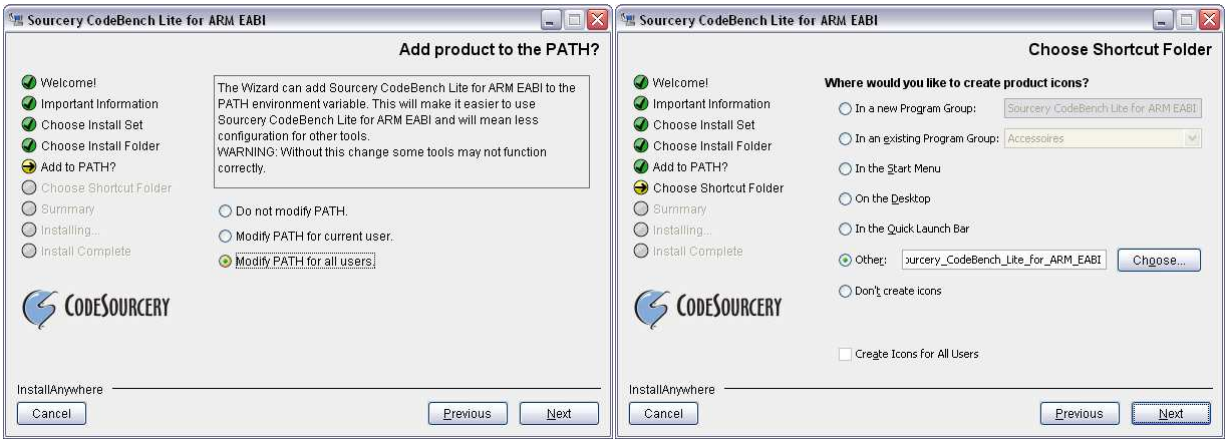

Figure 4- CodeBench PATH & Shortcut Folder Dialog Boxes

Check either "**Modify PATH for current user**" or "**Modify PATH for all users**" and click on "**Next**", then select where do you want to create the default shortcut folder and click on "**Next**" again.

Figure 5- CodeBench Summary & Install Dialog Boxes

Click on "**Install**" to start the actual installation process.

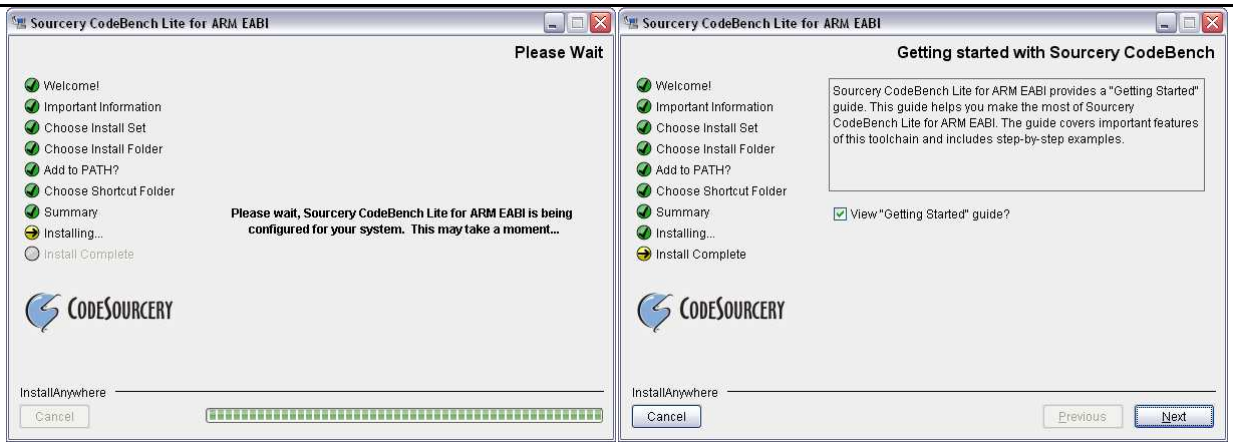

Figure 6 – CodeBench Configuration & Getting Started Dialog boxes

Click on "**Next**", then on "**Done**".

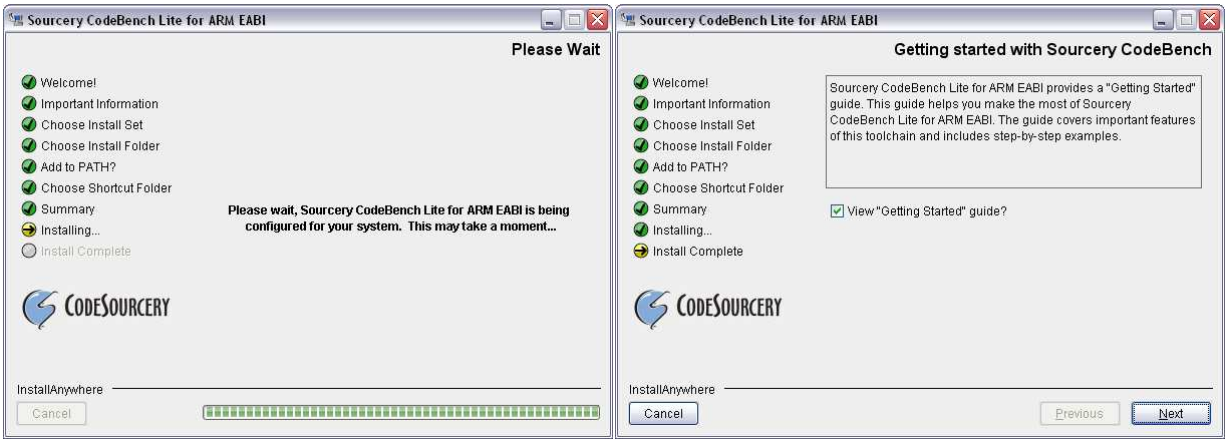

Figure 6 – CodeBench Configuration & Getting Started Dialog boxes

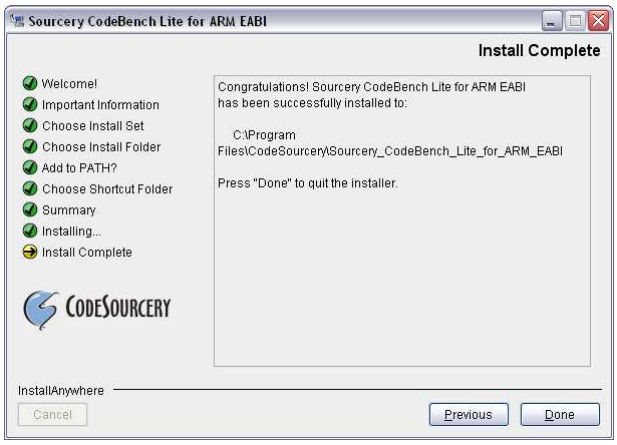

Figure 7 – CodeBench Final Screen

2.3 Optional J-Link SWD Debugger

At your option, you may want to use either a separate Segger J-Link JTAG/SWD Emulator with USB interface (http://www.segger.com/jlink.html) or an EFM32 Tiny Gecko Starter Kit from Energy Micro (http://www.energymicro.com/tools/efm32-tiny-gecko-starter-kit), featuring an integrated Segger J-Link programmer/debugger as well as a built-in Advanced Energy Monitoring capability.

Please install the required drivers provided with the above debugger.

In order to use the built-in Advanced Energy Monitoring capability from the Energy Micro EFM32 Tiny Gecko Starter Kit, you may need to perform the following hardware modifications on the board:

- Place a jumper between the 2 bottom row breakout pins “VMCU” and “3V3”
- Place a wire and jumper between the slider switch bottom-right pin and the 2x10 0.1” pitch shrouded header top-right pin

With these proposed modifications, the Advanced Energy Monitoring system will be able to monitor the external TD1202/TD1208 device instead of the built-in EFM32 CPU.

2.4 GNU ARM Eclipse Plugin

The installation of the required Eclipse GNU ARM Eclipse plugin is better performed from within Eclipse itself.

Navigate to the Eclipse installation directory and launch the “**eclipse.exe**” application. This will bring up a dialog window asking for a “workspace” location:

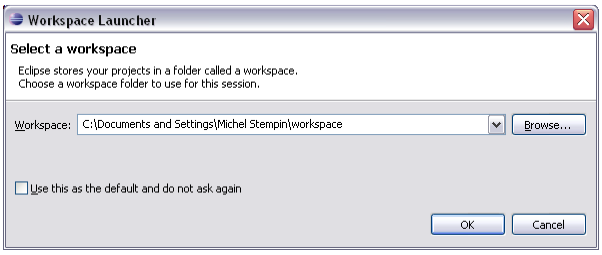

Figure 5- Eclipse Workspace Launcher

The Eclipse “workspace” is the location where Eclipse will store all the files related to your development projects. If you intend to have a single workspace for all your work, you may check the appropriate box to use it as a default and not be asked for it at every launch. Click “**OK**”, the main Eclipse window will appear:

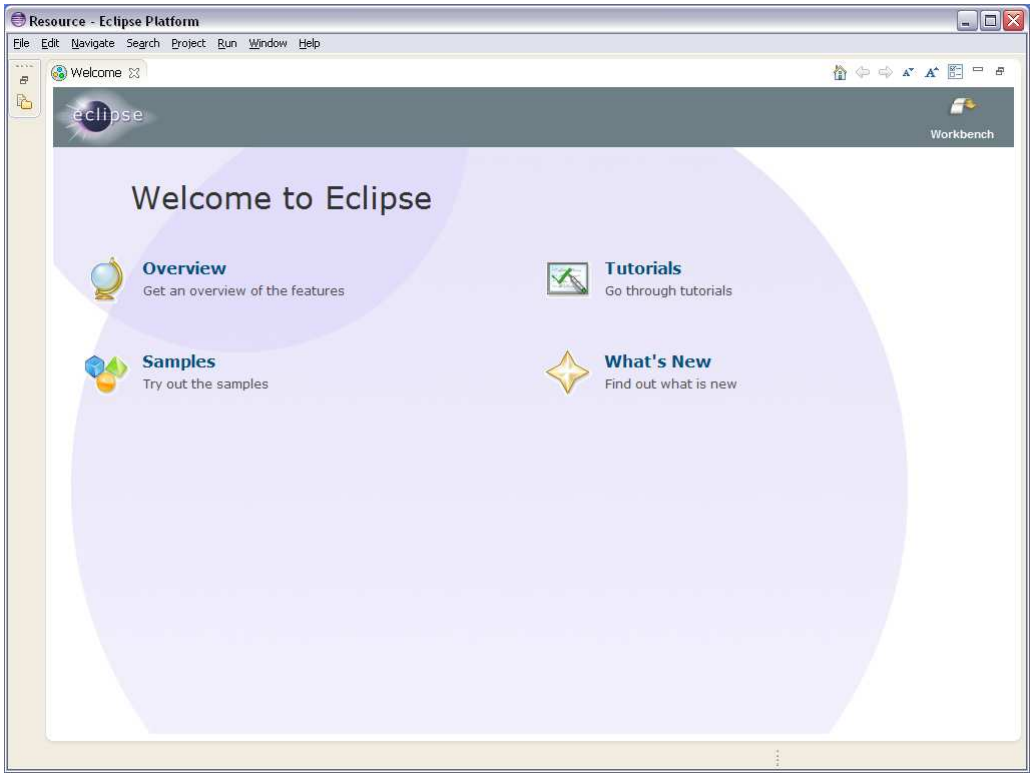

Figure 6- Eclipse Welcome Screen

To install IAR Embedded Workbench for Eclipse, choose "**Help>Install New Software**" from the menus:

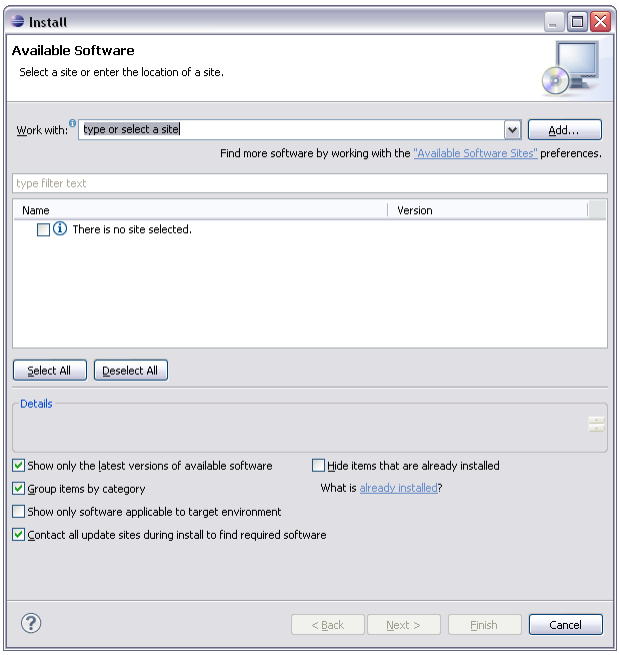

Figure 7- Eclipse Install New Software Dialog

Click on the "**Add**" button and use the following update site:

http://gnuarmeclipse.sourceforge.net/updates

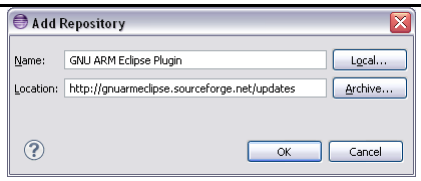

Figure 8- Add an Eclipse Software Repository

Fill in the fields as appropriate and click "**OK**", Eclipse will fetch the available software package from the provided repository URL and display a list:

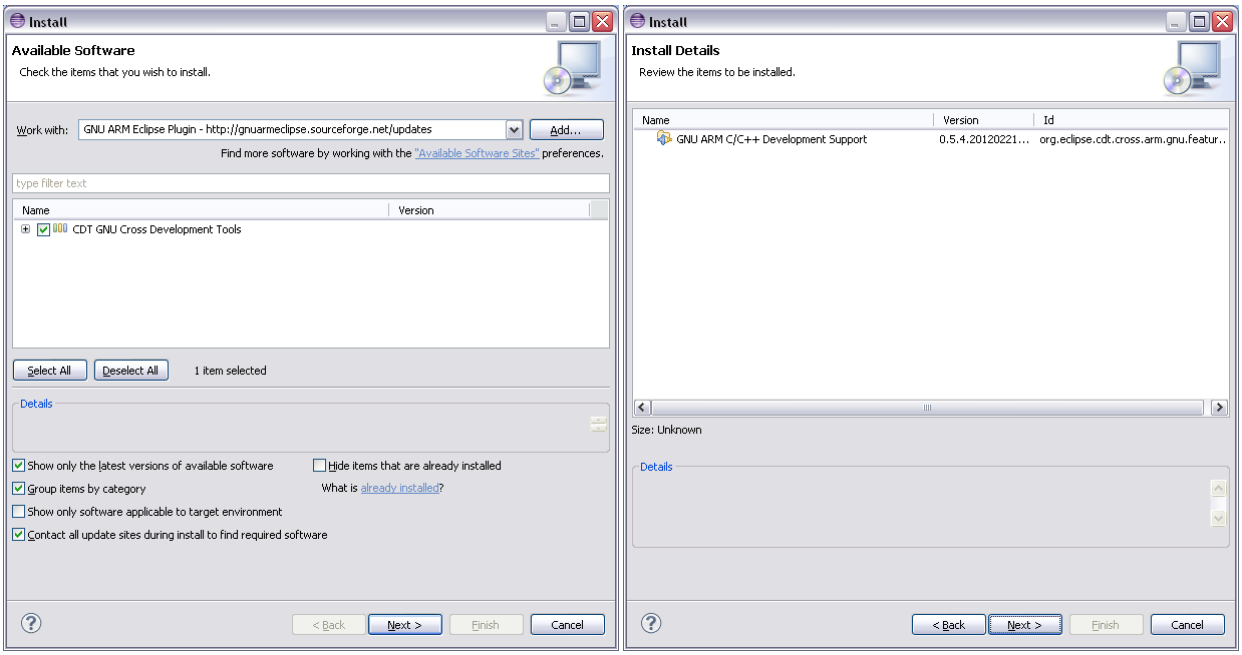

Figure 9- Eclipse Software Install Details

Check the box in front of the "**CDT GNU Cross Development Tools**" item in the list, then click "**Next**" twice.

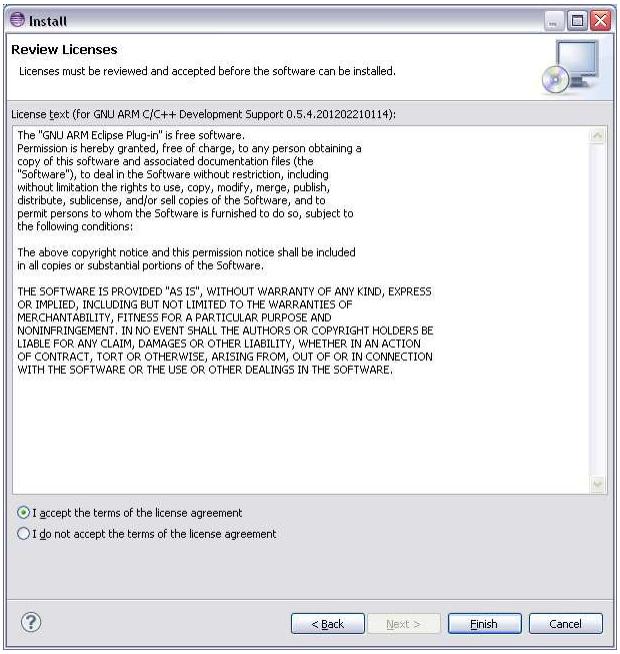

Figure 10- Eclipse Software License Review

Check the "**I accept the terms of the license agreements**" radio button and click "**Finish**", the installation will begin. Acknowledge any security warning that may occur by pressing on the "**OK**" button if required.

Eventually, Eclipse will need to be restarted for the changes to take effect:

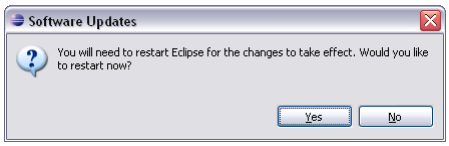

Figure 11- Restart Eclipse

Please accept by pressing "**Yes**".

2.5 Additional Eclipse Plugins

Although not mandatory, a few plugins are recommended for getting an efficient team development environment and documentation tools under the Eclipse IDE:

- The "EGit – Git Team Provider" plugin is available from "**Help>Eclipse Marketplace…**" in the menus. This plugin provides an integration of the Git software version system into Eclipse, which is the one used by Telecom Design to provide its SDK distribution. The installation is very similar to the plugin above
- The "Eclox" plugin is available from http://home.gna.org/eclox/#download and can be installed like the Eclipse IAR plugin. This plugin is a simple doxygen (http://www.doxygen.org/) frontend plugin for Eclipse, providing an integrated multi-language code documentation tool into the Eclipse IDE

DOCUMENT CHANGE LIST

Revision 1.0

- First Release

Revision 1.1

- Changed contact information

NOTES:

NOTES:

CONTACT INFORMATION

Telecom Design S.A.

Europarc — 22 Avenue Léonard de Vinci
33600 PESSAC, France
Tel: +33 5 57 35 63 70
Fax: +33 5 57 35 63 71

Please visit the Telecom Design web page:
http://www.telecomdesign.fr/

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Telecom Design assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Telecom Design assumes no responsibility for the functioning of undescribed features or parameters. Telecom Design reserves the right to make changes without further notice. Telecom Design makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Telecom Design assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Telecom Design products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Telecom Design product could create a situation where personal injury or death may occur. Should Buyer purchase or use Telecom Design products for any such unintended or unauthorized application, Buyer shall indemnify and hold Telecom Design harmless against all claims and damages.

Telecom Design is a trademark of Telecom Design S.A.
SIGFOX™ is a trademark of SigFox S.A.
Other products or brand names mentioned herein are trademarks or registered trademarks of their respective holders.