



Getting started with EasyRF and Eclipse IDE

Date: 11-11-2014
Version: 1



Table of Content

1. Introduction	3
1.1. Hardware used in this Getting Started document	3
2. Installing all needed programs	4
Step 1: Download project sources	4
Step 2: Extract sources to a workspace directory	4
Step 3: Download the Atmel ASF	4
Step 4: Download and install the GNU compiler	5
Step 5: Download and install GnuWin32 for Windows	5
Step 6: Download and install Eclipse IDE for C/C++ Developers	6
3. Creating Eclipse Makefile project	7
Step 1: Start Makefile project	7
Step 2: Add a make target	8
Step 2: MAKE!	9
4. Programming the hardware	10

1. Introduction

EasyRF is designing and manufacturing wireless communication solutions with an easy-to-use approach. With the different product groups EasyRF targets different applications and markets.

The products are standard and off-the-shelf products. Customization of the products is possible.

With the knowable design-center EasyRF is able to make quick design changes depending on the requests and complexity.

The firmware is build up with the latest developments on the open source platforms. The project is a so called Makefile project, so all you need to make your own solutions is the freely delivered reference project, the right compiler and the program Make.

This document describes how to get started on a Windows operating system using the Eclipse IDE. The Eclipse IDE is a program that makes browsing the code and building your solutions more easily, but is not necessary. You can select which ever IDE and/or operating system you are familiar with. The complete EasyRF code is buildable when the right compiler and make is installed on the system.

1.1. Hardware used in this Getting Started document

The ERF1000 module and its Evaluation Kit feature the latest ARM Cortex-M0+ CPU with integrated 2.4 GHz transceiver.

The module and kit are especially designed with IoT projects in mind.

For more information or to buy a kit or module, see the EasyRF website:

<http://shop.top-electronics.eu/evaluation-kit-for-erf1000-module-p-17199.html>

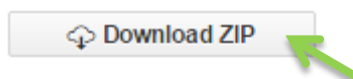
2. Installing all needed programs

First of all we needed the EasyRF sources. The code is written using Contiki. Contiki is a tiny operating system suited for embedded applications. It contains lots of mature drivers and tools designed with Internet-Of-Things in mind. For more info about Contiki visit: <http://contiki-os.org>

Step 1: Download project sources

The firmware is published on GitHub. GitHub is a revision control and source code management tool. You can make firmware request, contribute to the project by adding your own sources and sync with the latest developments.

If you are not familiar with GitHub you can download a complete sourcecode package of the latest release on the website: <https://github.com/EasyRF/contiki> by clicking:

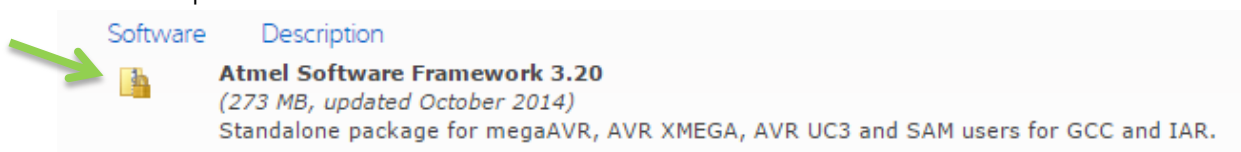


Step 2: Extract sources to a workspace directory

Extract the ZIP file to a map eg. "C:\workspace" and rename the map "contiki-master" to "easyrf".

Step 3: Download the Atmel ASF

The module contains a Atmel processor. Atmel has a well-documented firmware package which makes using the processor peripherals easy. Download it from <http://www.atmel.com/tools/avrsoftwareframework.aspx> (registration is needed). In this example we download version 3.20.



Extract the package to the map C:\workspace\easyrf\thirdparty\atmel. The ASF map has the name xdk-asf-3.20.1. Do not change this, because the variable ASF_ROOT in the Makefile C:\workspace\easyrf\cpu\atsamr21\Makefile.atsamr21 is revering to it.

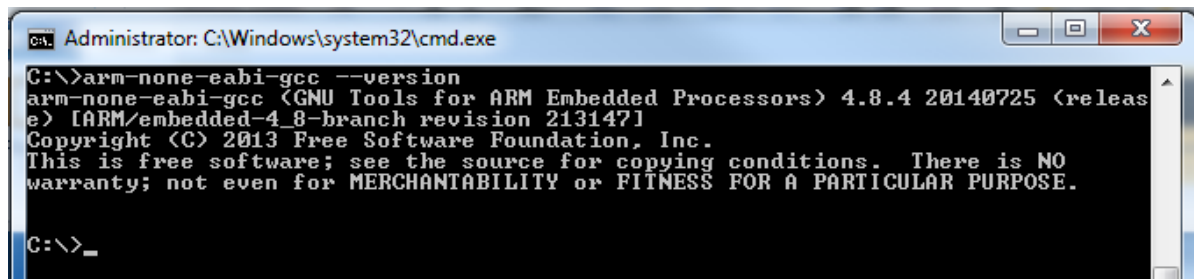
Step 4: Download and install the GNU compiler

The compiler allows us to make the binary suitable for our microcontroller. A mature open source compiler is found here:

<https://launchpad.net/gcc-arm-embedded/+download>

 [gcc-arm-none-eabi-4_8-2014q3-20140805-win32.exe \(md5\)](#)

Download and install the compiler to C:\gntools\4_8. The installer will add a link to the bin map to your system PATH variable. Check if the compiler is findable by opening a cmd and type "arm-none-eabi-gcc -version".




```
Administrator: C:\Windows\system32\cmd.exe
C:\>arm-none-eabi-gcc --version
arm-none-eabi-gcc (GNU Tools for ARM Embedded Processors) 4.8.4 20140725 (release) [ARM/embedded-4_8-branch revision 213147]
Copyright (C) 2013 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

C:\>_
```

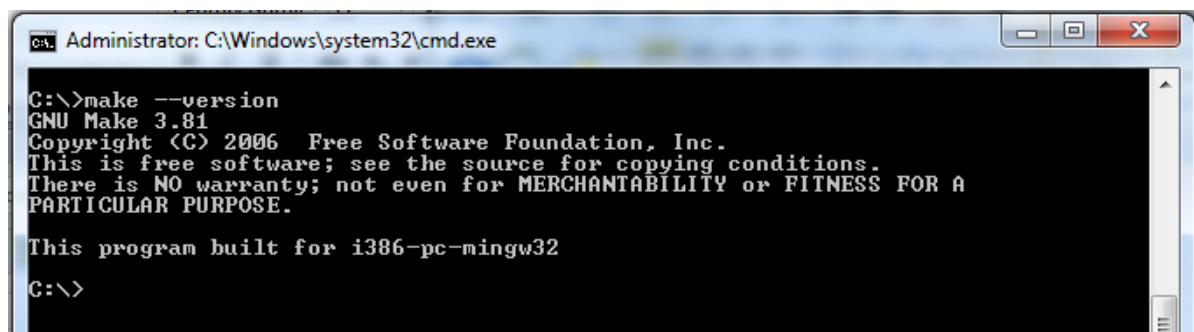
Step 5: Download and install GnuWin32 for Windows

Some handy tools like "Make" are ported to the windows platform within the GnuWin32. Download and install the package from:

<http://sourceforge.net/projects/getgnuwin32/files/getgnuwin32/0.6.30/>

 [GetGnuWin32-0.6.3.exe](#) 2009-11-16 3.4 MB

Check if the PATH variables are updated properly by typing "make -version" in a cmd.

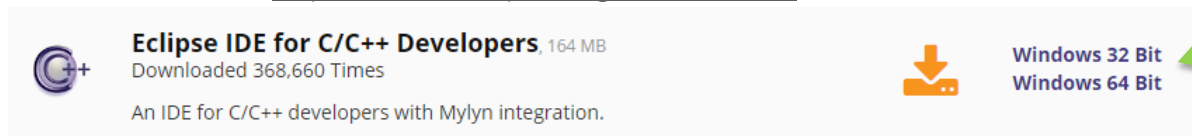


```
Administrator: C:\Windows\system32\cmd.exe
C:\>make --version
GNU Make 3.81
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for i386-pc-mingw32
C:\>
```

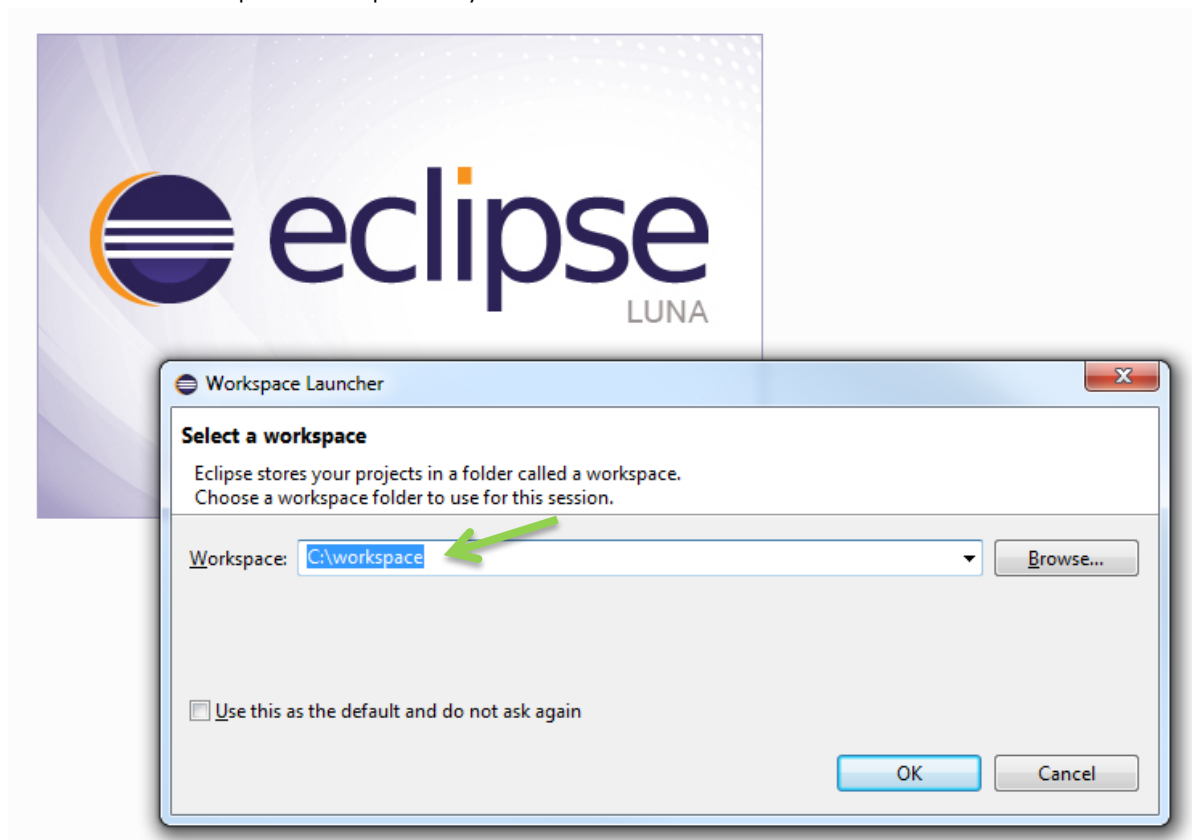
Step 6: Download and install Eclipse IDE for C/C++ Developers

Go to the website <https://www.eclipse.org/downloads/> and download:

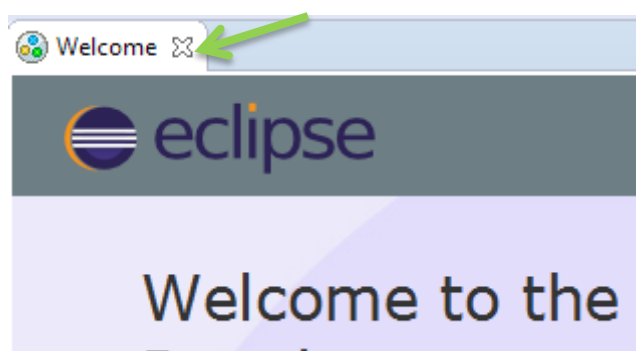


Eclipse is Java based and does not need an installer. Just extract the ZIP file to C:\eclipse and you can run eclipse by opening C:\eclipse\eclipse.exe.

Select the workspace map that you created and hit OK.



Close the Welcome tab.

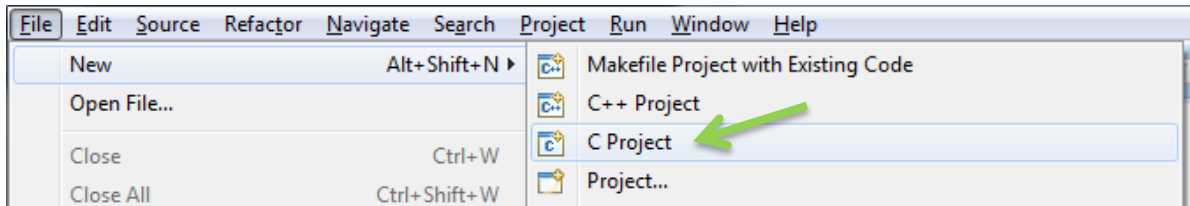


3. Creating Eclipse Makefile project

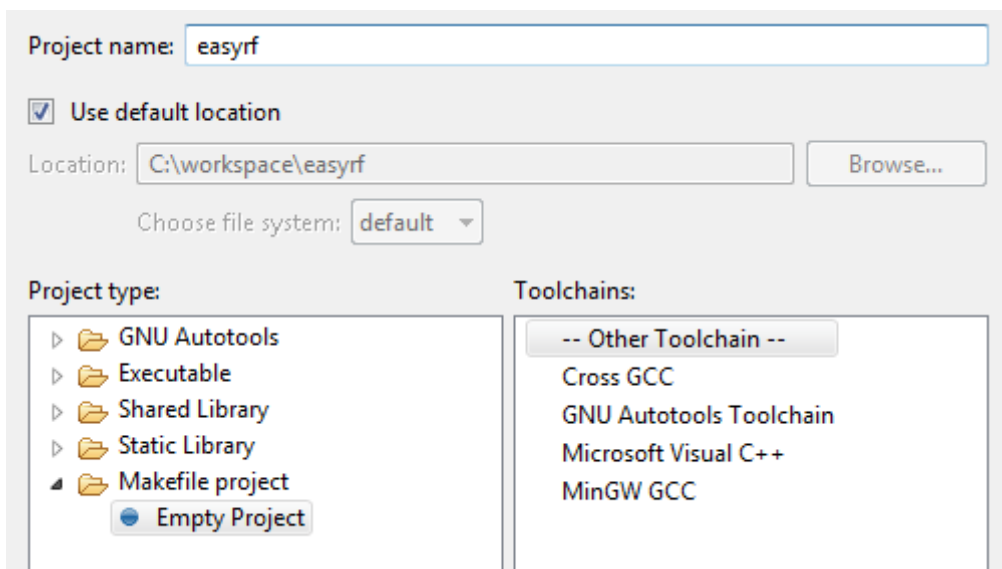
Now that all necessary programs are installed and in place, we are ready to start building an application.

Step 1: Start Makefile project

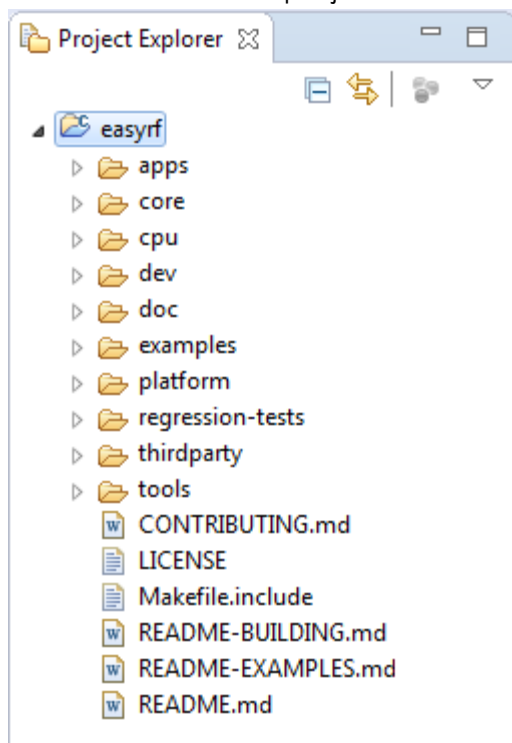
Start a new C project.



Name the project easyrf and select a Makefile project -> Empty Project -> -- Other Toolchain --.

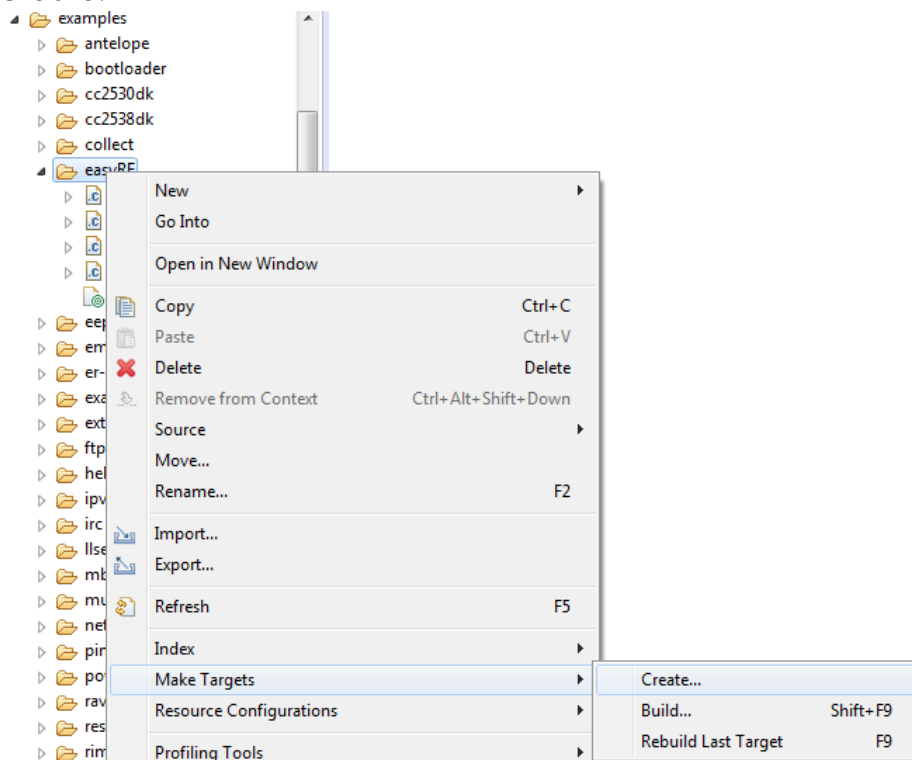


Click finish and the project is visible in the project explorer:

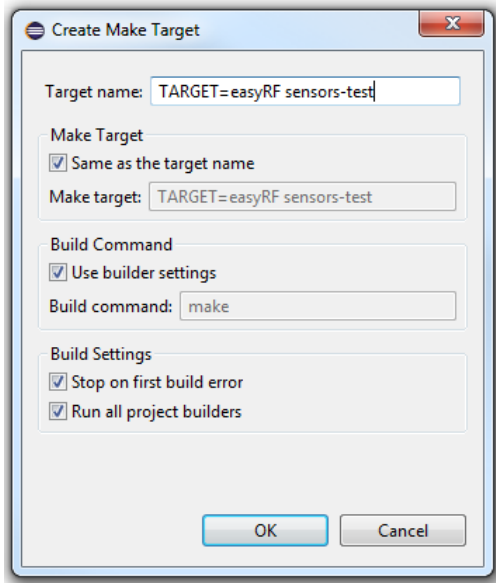


Step 2: Add a make target

Browse to examples/EasyRF. Right click the EasyRF folder and select Make target -> Create.

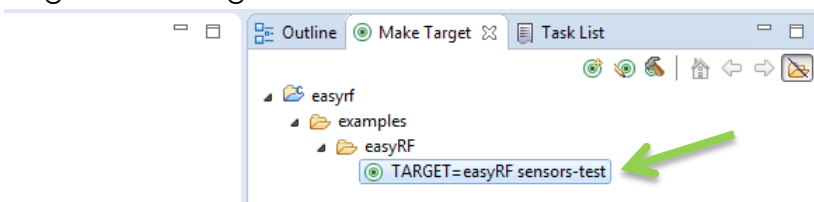


Fill in the target name "TARGET=EasyRF sensors-test".

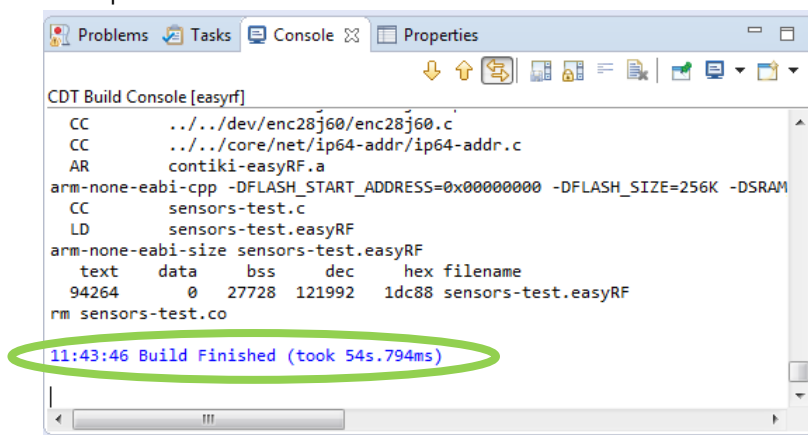


Step 2: MAKE!

Run the target by pushing F9 and selecting the created target, or double click the target in the target browser.



You must be able to see the building progress in the console toolbar and the compile result will be shown.



The bin file "sensors-test.easyRF" is created and placed in the example map "C:\workspace\easyrf\examples\easyRF"

4. Programming the hardware