2017

Install and configure the mbed toolchain offline

Contents

Preface

The following guide provides instructions on how to set up a development and debugging environment for mbed platforms with the Eclipse IDE.

The steps listed below have been tested in September 2017 with Eclipse Oxygen Release 4.7.1.

You cannot, as follows, using in Eclipse the library mbed available in ARM mbed, you must capitalize the library mbed-os that can be found through the following address https://github.com/ARMmbed/mbed-os

STEP1: Toolchain Installations

Below the list of software that you have to install in the suggested order:

1. STM32 ST-LINK Utility v.4.1.0 (install the driver suggested at the end of the installation);

2. JDK 8 Update 144 (install all the available features. Do not change the default installation and select *Close,* do not select *Next steps*);

3. Eclipse Oxygen Release v.4.7.1 (Once installed, check for new updates *Help -> Check for new Updates*);

4. Double check the Plugin C/C++ Development Tools is installed. Go to *Help -> About Eclipse -> Installation Details* and verify that *C/C++ Development Tools* is listed in *Installed sofware* (if not installed download and install it *Help -> Install new software*);

5. GCC ARM none eabi v.6 2017q2 (at the end of installation, select the option *Add to Path* leaving the other options to their default values);

6. Git v.2.13.2 (In Windows OS, you have to select the option *"Use Git from the Windows Command Prompt"*);

7. OpenOCD v.0.9.0 (Install without changing the default options);

8. GNU MCU v.4.1.1 It is an archive file. DO NOT EXTRACT IT. Install it via Eclipse as follows:
   - Select *Help -> Install new software -> Add -> Archive*;
   - Choose the archive *GNU MCU* and then *Open*;
   - In *Local* field there will be the archive file path, *Name* field you could write *GNU_MCU*;
   - Choose *Select all -> Next -> Next*;
   - accept the licence agreements;
   - Select *Finish*;

- Select *Install anyway* in case a popup arise before the end of the installation;
- restart;

9. Python v.2.7.13 (Select the option *"Add python.exe to Path"* and select "*Entire feature will be installed on local hard* drive" during the installation process);

10. mbed-cli v.1.2.0 It is an archive file. DO NOT EXTRACT THE ARCHIVE. It has to be installed running the file *mbed-cli_1.2.bat* as follows:
    - run the file *mbed-cli_1.2.bat*;
    - wait for the message *Successfully installed mbed-cli-1.2.0* (In case the command *pip* has not been found, add to the PATH the following *C:\Python27\Scripts*);
    - Close the *Command prompt*.

STEP2: Importing a project downloaded from mbed.org

In order to import an existing project in Eclipse please follow the following:

- Click *File -> New -> Makefile Project with Existing Code*;
- In the opened window choose *Browse*, select the project folder to be imported and select *Ok*;
- Click *Finish*.

STEP3: Configuring Eclipse for compilation purpose

Default configuration

The following procedure will drive you to build the binary file to be copied to the board.

The installation guide is targeted to Nucleo-F401RE, for other platform you have to change the name in the *Build Command* field.

At the end of a successful compilation in the Eclipse *Console* there will be blue printed *Build Finished* and a table including several information.

The following steps have to be taken for each new project.

1. In Eclipse project explorer, Right click on the project folder and choose *Properties*;
2. Click *C/C++ Build*;
3. Remove the flag *Use default build command*;
4. In *Build Command* type:

mbed compile -t GCC_ARM -m NUCLEO_F401RE

You could add the option *-c* in order to have a *clean build* creating a new *Build folder,* otherwise the folder is updated. In the last case, the building may take more time;

5. In the *Behavior* select *Build (Incremental build)* and delete the content of the field;
6. Deselect *Clean*;
7. Click *Apply*;
8. In order to star the compilation click the hammer icon in the toolbar;
9. At the end of a successful compilation, you could program the STM32 microcontroller copying the binary file in the Nucleo drive as seen by windows file explorer.
10. In case of unsuccessful compilation because of a missing *make.py* or other packets or system variables, a possible solution is the following:
    - Deinstall *Python v.2.7.13*;
    - Delete the folder *C:\Python27* (default path, you could have changed the path at installation time);
    - Re-install *Python v.2.7.13* e *mbed-cli v.1.2.0*.

Debug configuration

The following procedure will drive you to create a Debug configuration useful for debug purposes.

The installation guide is targeted to Nucleo-F401RE, for other platform you have to change the name in the *Build Command* field.

At the end of a successful compilation in the Eclipse *Console* there will be blue printed *Build Finished* and a table including several information.

The following steps have to be taken for each new project.

1. Click on *Project* in the Eclipse menu and select *Build Configuration -> Manage*;
2. Click *New*;
3. In the box *Name* type *Debug*;
4. Select copy from *Default*;
5. Click *Ok* and again *Ok*;
6. In Eclipse project explorer, Right click on the project folder and choose *Properties*;
7. Click *C/C++ Build*;
8. In the box *Configuration* select *Debug*;
9. In the box *Build Command* type:

   mbed compile -t GCC_ARM -m NUCLEO_F401RE -c --profile mbed-os/tools/profiles/debug.json

Verify the name of the board is the right one;

10. Click *Apply* and then *Ok*;
11. In order to start the compilation select *Debug* in the menu of Toolbar.

STEP4: Debugging the code

Take the following steps in order to debug the code.

The installation guide is targeted to Nucleo-F401RE, for other platform you have to change the name in the *Config* field.

In order to start the debug, the Nucleo has to be plugged in.

1. Click *Run* in the Eclipse menu and select *Debug Configurations*;
2. Click *GDB OpenOCD Debugging* and select *New*;
3. Check that the name in *Project* is the right one;
4. In the box *C/C++ Application* select .elf file path. It should be in the folder *BUILD\NUCLEO_F401RE\GCC_ARM\….elf*;
5. Select *Debug* in the box *Build Configuration*;
6. Select *Disable Auto Build*;
7. In the tab *Debugger* select *Start OpenOCD locally*;
8. In *Executable* click *Browse* and select the path to the file *openocd.exe* (the default is *Programs\GNU ARM Eclipse\OpenOCD\<version>\bin*);
9. Type 3333 as *GDB port* and 4444 as *Telnet port*;
10. In the *Config options* box, type:

    -f interface\stlink-v2.cfg -f board\st_nucleo_f4.cfg

    Verifying that the configuration file name is the right one for the board you are using;

11. In *GDB Client Setup,* in the box *Executable* click *Browse* and insert the path to the file *arm_non_eabi_gdb.exe* (the default path is *Programs (x86)\GNU Tools ARM Embedded\<version>\bin*);
12. Type

    set mem inaccessible-by-default off

    in the box *Commands*;

13. Select the tab *Common*;
14. Select *Shared files.* Do not modify the box content;

15. In *Display in favorites menù* select *Debug*;
16. Click *Apply* and then *Debug*.