**Installazione e configurazione della toolchain mbed offline**

2017

Indice

[Introduzione 2](#_Toc494730622)

[Installazioni necessarie 2](#_Toc494730623)

[Come importare un progetto 3](#_Toc494730624)

[Configurazioni per la compilazione del codice in Eclipse 3](#_Toc494730625)

[Configurazione per la compilazione di Default 3](#_Toc494730626)

[Configurazione per la compilazione di Debug 4](#_Toc494730627)

[Effettuare debug del codice 5](#_Toc494730628)

# Introduzione

La seguente guida fornisce le istruzioni per configurare un ambiente di sviluppo e debug per piattaforme mbed con l’IDE Eclipse.

I passaggi di seguito elencati sono stati testati nel mese di Settembre 2017 con Eclipse Oxygen Release 4.7.1.

Non sarà possibile, seguendo quanto di seguito riportato, utilizzare in Eclipse la libreria *mbed* presente in ARMmbed, sarà invece necessario sfruttare la libreria *mbed-os* reperibile tramite al seguente indirizzo <https://github.com/ARMmbed/mbed-os>

# PASSO1: Installazioni necessarie

Di seguito si riporta l'elenco dei software necessari al fine di poter sviluppare, compilare e fare debug del codice su piattaforma *mbed* tramite Eclipse nell'ordine d'installazione che, in base all'esperienza maturata, l'autore ritiene più opportuno:

1. **STM32 ST-LINK Utility v.4.1.0** (installare anche i driver proposti al termine dell’installazione);
2. **JDK 8 Update 144** (installare tutte le features disponibili senza modificare il percorso d’installazione di default e alla fine dell’installazione selezionare *Close* e non *Next steps*);
3. **Eclipse Oxygen Release v.4.7.1** (una volta avviato è consigliabile verificare la presenza di aggiornamenti andando su *Help -> Check for new Updates*);
4. Verificare che il Plugin **C/C++ Development Tools** risulti installato nella versione di Eclipse in uso andando su *Help -> About Eclipse -> Installation Details* e controllando la presenza di *C/C++ Development Tools* tra gli *Installed sofware* (nel caso non fosse installato scaricarlo andando su *Help -> Install new software*);
5. **GCC ARM none eabi v.6 2017q2** (alla fine dell'installazione aggiungere l'opzione *Add to Path* lasciando le altre spunte invariate);
6. **Git v.2.13.2** (con sistema operativo Windows è necessario spuntare l'opzione *"Use Git from the Windows Command Prompt"*);
7. **OpenOCD v.0.9.0** (installare il binario senza modificare le spunte di default);
8. **GNU MCU v.4.1.1** che non va estratto dal suo archivio, ma deve essere installato tramite Eclipse nel modo seguente:

* cliccare *Help -> Install new software -> Add -> Archive*;
* nella finestra che appare va selezionato l'archivio *GNU MCU* e poi *Apri*;
* ora in *Local* ci sarà il percorso dell'archivio, mentre in *Name* si può scrivere un nome qualunque, ad esempio *GNU\_MCU*;
* ora premere *Select all -> Next -> Next*;
* accettare i termini della licenza;
* premere *Finish*;
* premere *Install anyway* nel caso esca un avviso prima del completamento dell'installazione;
* riavviare;

1. **Python v.2.7.13** (è necessario aggiungere l'opzione *"Add python.exe to Path"* selezionando “*Entire feature will be installed on local hard* drive” durante l'installazione);
2. **mbed-cli v.1.2.0** che non va estratto dal suo archivio, ma deve essere installato tramite il file *mbed-cli\_1.2.bat* nel modo seguente:

* aprire il file *mbed-cli\_1.2.bat*;
* attendere il messaggio *Successfully installed mbed-cli-1.2.0* (nel caso in cui il comando *pip* non fosse trovato, inserire *C:\Python27\Scripts* tra le Variabili di sistema);
* chiudere il *Prompt dei comandi*.

# PASSO2: Importare un progetto scaricato da mbed.com

Per importare un progetto esistente in Eclipse è sufficiente seguire i seguenti passaggi:

* Cliccare su *File -> New -> Makefile Project with Existing Code*;
* Nella finestra che si apre premere su *Browse*, cliccare sulla cartella del progetto da importare e premere *Ok*;
* Infine premere *Finish*.

# PASSO3: Configurare Eclipse per la compilazione del codice

## Configurazione per la compilazione di Default

I seguenti passaggi sono fondamentali al fine di effettuare la compilazione di Default ed ottenere, tra gli altri, il file con estensione .bin.

Tale guida è stata realizzata per la piattaforma Nucleo-F401RE, in caso di piattaforma differente basta cambiare il nome della scheda nella stringa da digitare in *Build Command*.

Se tutto si è svolto correttamente, in *Console* apparirà la scritta blu *Build Finished* ed una tabella con le colonne Module, .text, .data, .bss.

Si sottolinea che quanto di seguito elencato va ripetuto per ogni progetto creato.

1. Cliccare col tasto destro del mouse sulla cartella del progetto da compilare e selezionare *Proprietà*;
2. Cliccare su *C/C++ Build*;
3. Rimuovere il flag da *Use default build command*;
4. In *Build Command* scrivere:

*mbed compile -t GCC\_ARM -m NUCLEO\_F401RE*

volendo è possibile anche aggiungere *-c* alla fine per ottenere una *clean build* e quindi rigenerare la cartella *Build* al posto di aggiornarla semplicemente, tale procedura è comunque sconsigliata poiché in genere non necessaria e la creazione di una nuova cartella *BUILD* impiega molto tempo;

1. Nel tab *Behavior* selezionare *Build (Incremental build)* e cancellare il contenuto del relativo box;
2. Deselezionare *Clean*;
3. Premere *Apply*;
4. Per iniziare la compilazione cliccare sul simbolo del martello presente nella Toolbar di Eclipse;
5. Terminata con successo la compilazione è possibile effettuare il flash del codice sulla board semplicemente copiando il file con estensione .bin presente nella cartella *BUILD* appena generata nell'unità esterna *NUCLEO* vista dal computer.
6. Nel caso in cui la compilazione non andasse a buon fine segnalando la mancanza di *make.py* o di altri pacchetti o di variabili di sistema, una possibile soluzione è rappresentata da:

* Disinstallare *Python v.2.7.13*;
* Eliminare la cartella *C:\Python27* (percorso valido se non è stato modificato il path di default);
* Reinstallare *Python v.2.7.13* e *mbed-cli v.1.2.0*.

## Configurazione per la compilazione di Debug

I seguenti passaggi sono fondamentali al fine di effettuare la compilazione di Debug che è propedeutica per poter poi fare debug del codice.

Tale guida è stata realizzata per la piattaforma Nucleo-F401RE, in caso di piattaforma differente basta cambiare il nome della scheda nella stringa da digitare in *Build Command*.

Se tutto si è svolto correttamente, in *Console* apparirà la scritta blu *Build Finished* ed una tabella con le colonne Module, .text, .data, .bss.

Si sottolinea che quanto di seguito elencato va ripetuto per ogni progetto creato.

1. Cliccare su *Project* nel Menu di Eclipse e scegliere *Build Configuration -> Menage*;
2. Cliccare su *New*;
3. Nel box *Name* digitare *Debug*;
4. Selezionare la copia da *Default*;
5. Premere *Ok* e poi di nuovo *Ok*;
6. Cliccare col tasto destro sulla cartella del progetto e selezionare *Proprietà*;
7. Cliccare su *C/C++ Build*;
8. Nel box *Configuration* selezionare *Debug*;
9. Nel box *Build Command* scrivere:

*mbed compile -t GCC\_ARM -m NUCLEO\_F401RE -c --profile mbed-os/tools/profiles/debug.json*

verificando che il nome della board sia corretto;

1. Cliccare su *Apply* e poi su *Ok*;
2. Per iniziare la compilazione selezionare *Debug* nel menù relativo al simbolo del martello presente nella Toolbar di Eclipse.

# PASSO4: Effettuare il debug del codice

I seguenti passaggi sono necessari per effettuare debug del codice.

Tale guida è stata realizzata per la piattaforma Nucleo-F401RE, in caso di piattaforma differente basta cambiare il nome del file di configurazione nella stringa da digitare nel box *Config*.

Si ricorda che per poter avviare il Debug la piattaforma deve essere già collegata al PC.

1. Cliccare su *Run* nel Menu di Eclipse e scegliere *Debug Configurations*;
2. Cliccare su *GDB OpenOCD Debugging* e selezionare *New*;
3. Verificare che in *Project* ci sia il nome corretto del progetto;
4. Nel box *C/C++ Application* mettere il percorso al file con estensione .elf, ottenuto dalla compilazione in configurazione Debug, cliccando su *Browse*, entrando nella cartella del progetto e poi in *BUILD\NUCLEO\_F401RE\GCC\_ARM\….elf*;
5. Selezionare *Debug* nel box *Build Configuration*;
6. Selezionare *Disable Auto Build*;
7. Nel tab *Debugger* selezionare *Start OpenOCD locally*;
8. In *Executable* cliccare su *Browse* ed indicare il percorso al file *openocd.exe* (se non è stato cambiato il percorso di installazione di default, dovrebbe essere in *Programmi\GNU ARM Eclipse\OpenOCD\<version>\bin\*);
9. Digitare 3333 come *GDB port* e 4444 come *Telnet port*;
10. Nel box *Config options* digitare:

*-f interface\stlink-v2.cfg -f board\st\_nucleo\_f4.cfg*

verificando che il nome del file di configurazione sia corretto per la board in uso;

1. In *GDB Client Setup* nel box *Executable* cliccare su *Browse* ed inserire il percorso al file *arm\_non\_eabi\_gdb.exe* (se non è stato cambiato il percorso di installazione di default, dovrebbe essere in *Programmi (x86)\GNU Tools ARM Embedded\<version>\bin\*);
2. Digitare

*set mem inaccessible-by-default off*

nel box *Commands*;

1. Selezionare il tab *Common*;
2. Selezionare *Shared files* senza modificare il relativo box;
3. In *Display in favorites menù* selezionare *Debug*;
4. Cliccare su *Apply* e successivamente su *Debug*.