

Using Eclipse to program STM32 CPUs

Jan Hieber
Nico Zellinger
Tim Kurz

Inhaltsverzeichnis

1	Further information if you encounter problems	1
2	Setup Eclipse	2
3	Setup Toolchain	2
4	Explanation of components	3
5	Workaround if you have no permissions in Windows	4
6	Create new project	5
7	Program the target	5
8	Hardware Debugging	6
9	Additional notes	7

1 Further information if you encounter problems

- Eclipse and Plug-ins
 - <http://gnuarmclipse.livius.net/blog/plugins-install/>
- Toolchain
 - GCC ARM
 - * <http://gnuarmclipse.livius.net/blog/toolchain-install/>
 - Build Tools (make, rm ...)
 - * <http://gnuarmclipse.livius.net/blog/build-tools-windows/>
 - OpenOCD
 - * <http://gnuarmclipse.livius.net/blog/openocd-install/>
 - ST-LINK/V2
 - * <http://gnuarmclipse.livius.net/blog/openocd-install/#ST-LINKV2>
- Using Linux
 - Most software is available in your distributors software repository
 - Ubuntu: eclipse, eclipse-cdt, gcc-arm-none-eabi, binutils-arm-none-eabi, libnewlib-arm-none-eabi, gdb-arm-none-eabi, openocd
 - Arch Linux: eclipse, eclipse-cdt, arm-none-eabi-gcc, arm-none-eabi-binutils, arm-none-eabi-newlib, arm-none-eabi-gdb, openocd, stlink

2 Setup Eclipse

- Download and install *Eclipse IDE for C/C++ Developers* from <https://eclipse.org>
 - Start Eclipse, create/select workspace
- Click *Help > Install New Software*
 - Work with: Luna (or later version e. g. Mars)
- Select everything from *Mobile and Device Development* and install
- Click *Help > Install New Software > Add*
 - name: GNU ARM Eclipse Plug-ins
URL: <http://gnuarmecclipse.sourceforge.net/updates>
 - Select: Cross Compiler; Generic Cortex-M Project Templates; OpenOCD Debugging; Packs; STM32Fx Project Templates

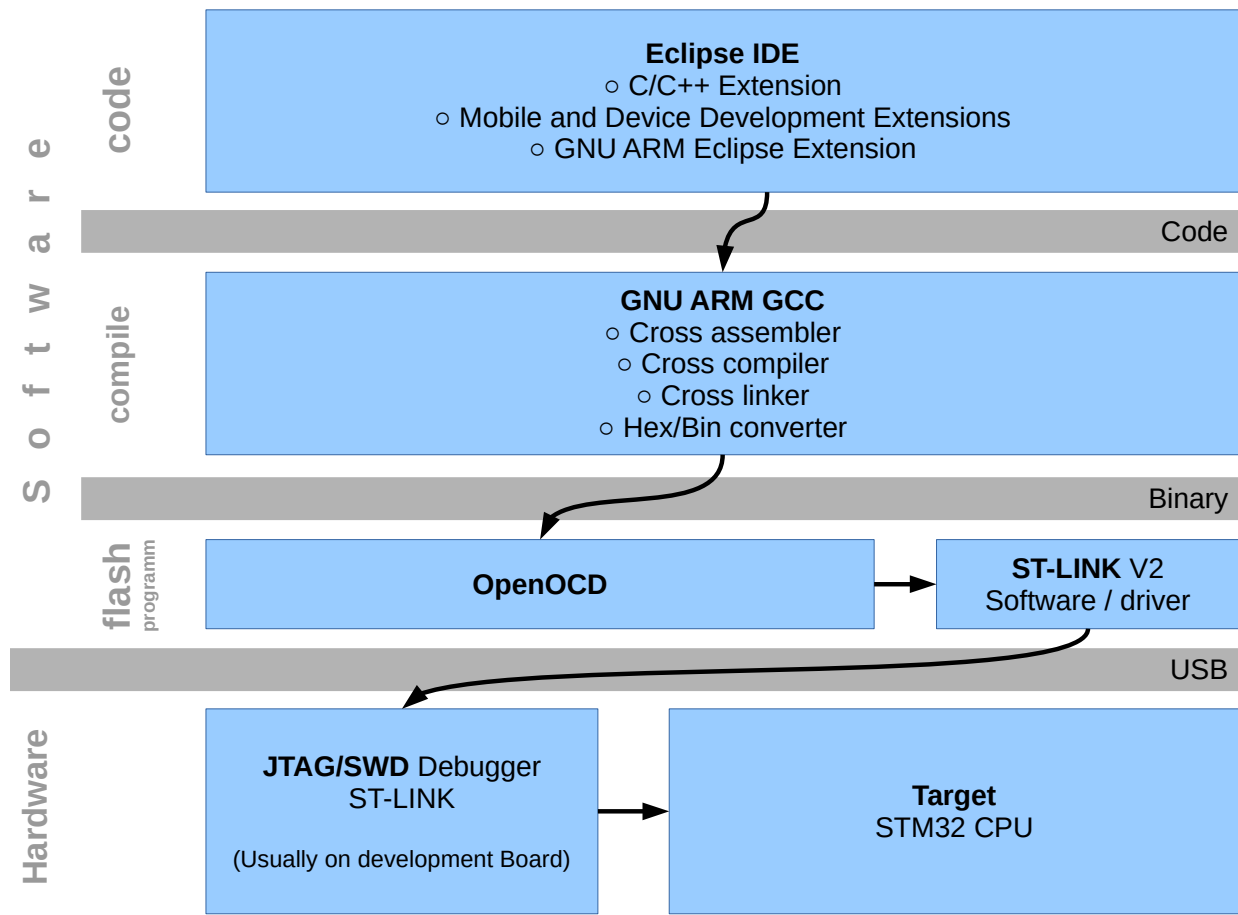
3 Setup Toolchain

- GCC ARM
 - <https://launchpad.net/gcc-arm-embedded>
 - Download ZIP package, not the installer
- OpenOCD
 - <http://sourceforge.net/projects/gnuarmecclipse/files/OpenOCD/>
 - No permissions to install?
 - * Unpack bin folder with 7-ZIP, rename to *openocd*
 - * Unpack \$_OUTDIR into *openocd*
- ST-LINK/V2
 - Windows 7/XP: www.st.com/web/catalog/tools/FM147/SC1887/PF258167
 - Windows 8: www.st.com/web/catalog/tools/FM147/SC1887/PF259459
 - No permissions to install?
 - * Driver for Windows: developer.mbed.org/media/uploads/dan/stlinknucleodriverson.zip
 - * ST-LINK Utility: -

- Build Tools (make, rm ...)
 - sourceforge.net/projects/gnuarmecclipse/files/Build%20Tools/
 - No permissions to install? Extract bin folder with 7-ZIP and rename to *BuildTools*

4 Explanation of components

- Eclipse
 - Is a programming environment for Java developers
- Eclipse CDT
 - Is the extended version for C/C++ programmers
- GNU ARM Eclipse Plug-ins
 - Adds the support for ARM cross compile projects
- GCC ARM
 - Is the compiler (and debugger etc) for targeting the ARM CPUs
- OpenOCD
 - Is the programmer for flashing and debugging the target
- ST-LINK
 - Is the in-circuit debugger and programmer for STM8 and STM32 CPUs



5 Workaround if you have no permissions in Windows

- The problem is that Eclipse expects build tools like make and rm in the Windows PATH variable, which you can not alter because you have no permissions.
- Solution, use the Batch file provided below to start Eclipse. Basically this file changes the PATH variable and then starts Eclipse with this variable

```

rem set path, use current path where this script is
set P=%~dp0
set ECLIPSE=%P%EclipseCDT
set OPENOCD=%P%OpenOCD
set STLINK=%P%stlink
set TOOLS=%P%BuildTools
set PATH=%OPENOCD%;%STLINK%;%TOOLS%;%PATH%
start "" %ECLIPSE%\eclipse.exe
  
```

6 Create new project

- After installing everything, restart Eclipse and close the Welcome page
- File → New → Other → C Project
 - Project Type: *STM32F4xx C/C++ Project*
 - Toolchain: *GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc)*
 - Toolchain path: where you installed GCC, the folder should contain: *arm-none-eabi, bin, lib, share*
- Window → Open Perspective → Packs
 - Update the the repos
 - Select STMicroelectronics → STM32F4 Series
Select Keil → STM32F4xx_DFP → install latest version
 - WARNING: This is really big. The packs are stored in your active workspace
Consider creating/moving your workspace to a USB drive.
- Check your external oscillator frequency (HSE_VALUE) in file in
<projectname>\system\src\cmsis\system_stm32f4xx.c

7 Program the target

- Before compiling (first time after creating project)
 - Select *Raw Binary* in Project properties → *C/C++ Build* → *Settings* → *Tool Settings* → *Cross* → *ARM GNU Create Flash Image* → *General* → *Output file format*
 - Select the target CPU in Project properties → *C/C++ Build* → *Settings* → *Devices*
- Compile the project
 - Rightclick on project → *Build Project*
 - Watch the Console window for errors
- In *Run* → *External Tools* → *External Tools Configurations...*
 - Select *Programm* and click *new*
 - In Location enter path to OpenOCD executable or only openocd.exe if you have OpenOCD in your systems PATH variable

- In *Working Directory* enter: `${workspace_loc:${project_name}/Debug}`
- In Arguments enter:
 - `–s E:\HS\OpenOCD`
 - `–f stm32f429discovery`
 - `–c "init"`
 - `–c "reset halt"`
 - `–c "sleep 100"`
 - `–c "wait_halt 2"`
 - `–c "flash write_image erase ${project_name}.bin 0x08000000"`
 - `–c "sleep 100"`
 - `–c "reset run"`
 - `–c shutdown`
- In tab *Common*, check *Display in favourites menu*

8 Hardware Debugging

- *Run > Debug Configurations...*
 - Select *GDB OpenOCD Debugging* and click *new*
 - In Tab *Debugger* enter the following in *Config options*
 - `–s E:\HS\OpenOCD`
 - `–f stm32f429discovery`

9 Additional notes

- Optional things, you only need this if you encounter problems or if you are very bored
 - Updating hardware abstraction library and other files (stm32f4-hal)
 - * Download latest STM32CubeF4 www.st.com/web/en/catalog/tools/PF259243
 - * Copy Drivers\STM32F4xx_HAL_Driver\Src* to
 <projectname>\system\src\stm32f4-hal
 - * Copy Drivers\STM32F4xx_HAL_Driver\Inc* to
 <projectname>\system\include\stm32f4-hal
 - * Copy Drivers\CMSIS\Device\ST\STM32F4xx\Include* to
 <projectname>\system\include\cmsis
 - * Copy Drivers\CMSIS\Device\ST\STM32F4xx\Source\Templates
 \system_stm32f4xx.c to <projectname> \system\src \cmsis
 - * Copy Drivers\CMSIS\Include* to <projectname> \system\include\cmsis