

# How to setup a development environment for XMC™ microcontrollers

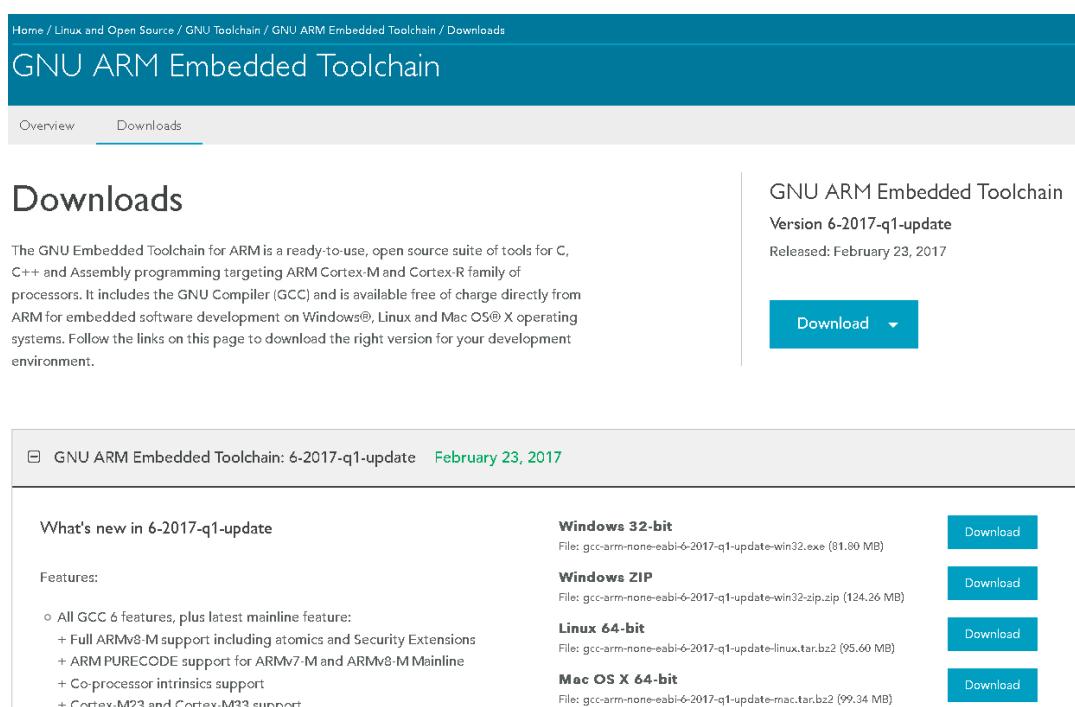
This document describes the steps to setup a development environment for XMC™ microcontrollers running in Windows, Linux and Mac OS X using free available tools.



## 1. GCC ARM Embedded Tool Chain

Download and install GNU ARM GCC toolchain

<https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads>



The screenshot shows the official GNU ARM Embedded Toolchain download page. At the top, there's a breadcrumb navigation: Home / Linux and Open Source / GNU Toolchain / GNU ARM Embedded Toolchain / Downloads. Below the header, the page title is "GNU ARM Embedded Toolchain". There are two tabs: "Overview" and "Downloads", with "Downloads" being the active tab. On the left, under "Downloads", there's a section titled "Downloads" with a brief description of what the toolchain is. On the right, there's a summary for the "Version 6-2017-q1-update" release, dated February 23, 2017. It includes a "Download" button. Below this summary, there's a table showing download links for different platforms:

Platform	File	Download
Windows 32-bit	gcc-arm-none-eabi-6-2017-q1-update-win32.exe (81.00 MB)	<a href="#">Download</a>
Windows ZIP	gcc-arm-none-eabi-6-2017-q1-update-win32.zip.zip (124.26 MB)	<a href="#">Download</a>
Linux 64-bit	gcc-arm-none-eabi-6-2017-q1-update-linux.tar.bz2 (95.60 MB)	<a href="#">Download</a>
Mac OS X 64-bit	gcc-arm-none-eabi-6-2017-q1-update-mac.tar.bz2 (99.34 MB)	<a href="#">Download</a>

## 2. Eclipse IDE

Download and install Eclipse for C/C++ from

<http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/neon2>

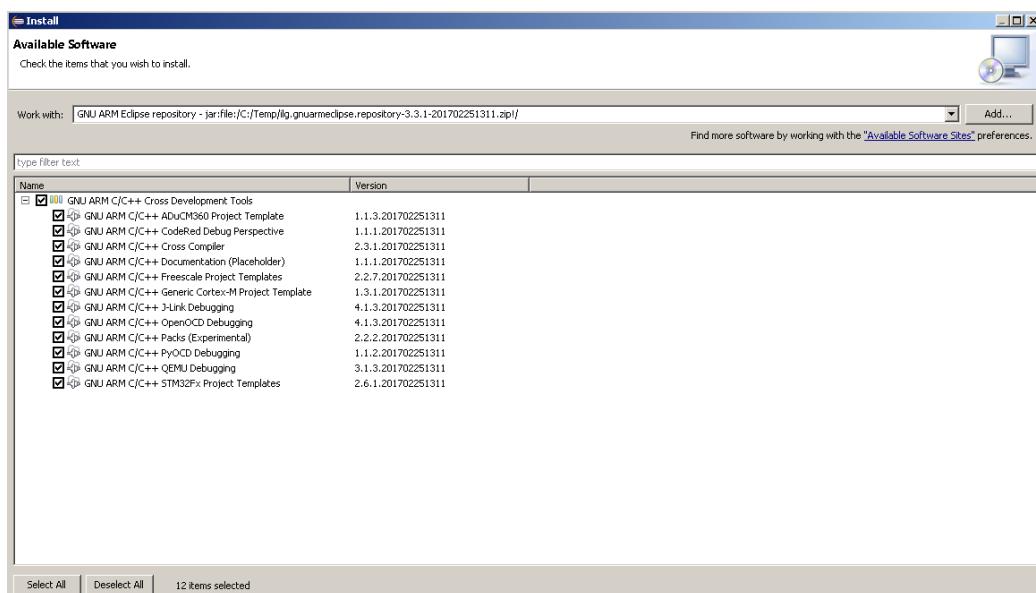
Open eclipse and create a new workspace.

## 3. GNU ARM Eclipse Plugin

Download the latest version of the plugin from

<https://github.com/gnuaarmeclipse/plug-ins/releases>

In eclipse, go to Help>Install New Software... and click on Add..., select Archive and browse to the previous downloaded file. Check all items and click on Next



Under windows you will need also tools for building the projects using make

<https://github.com/gnuaarmeclipse/windows-build-tools/releases>

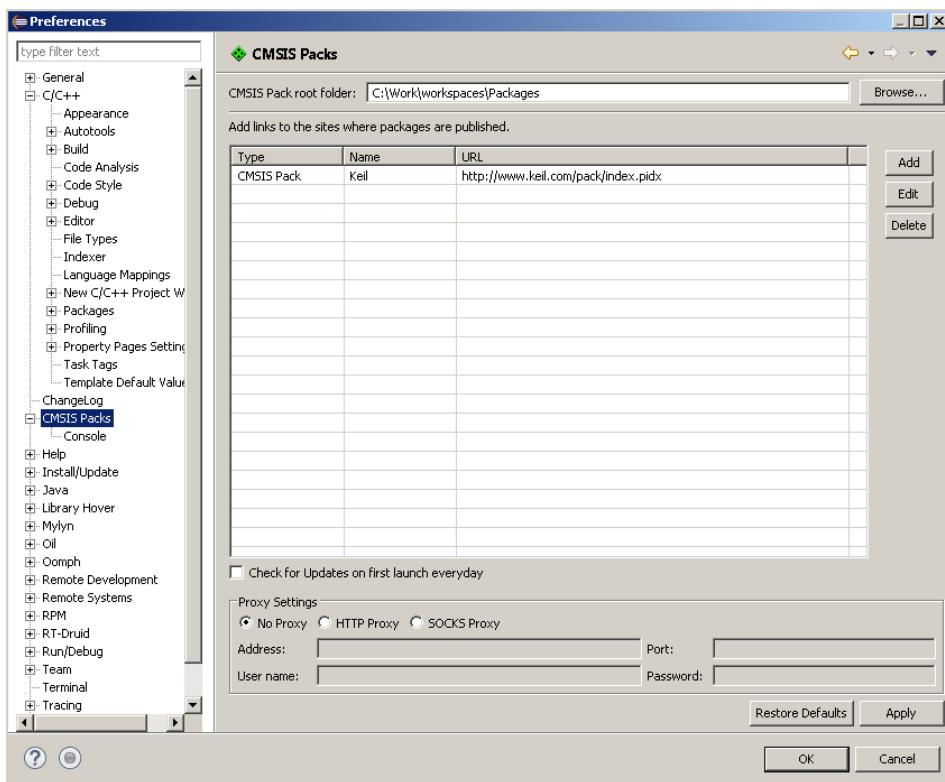
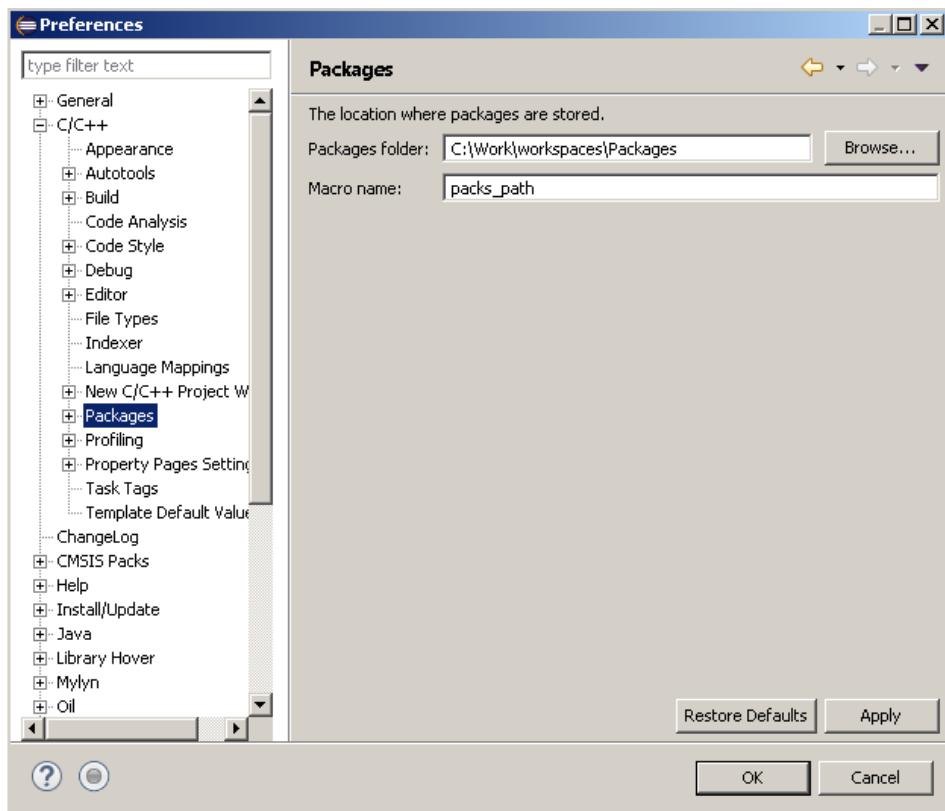
## 4. CMSIS Pack Plugin

Download the latest version of the plugin from

<https://github.com/ARM-software/cmsis-pack-eclipse/releases>

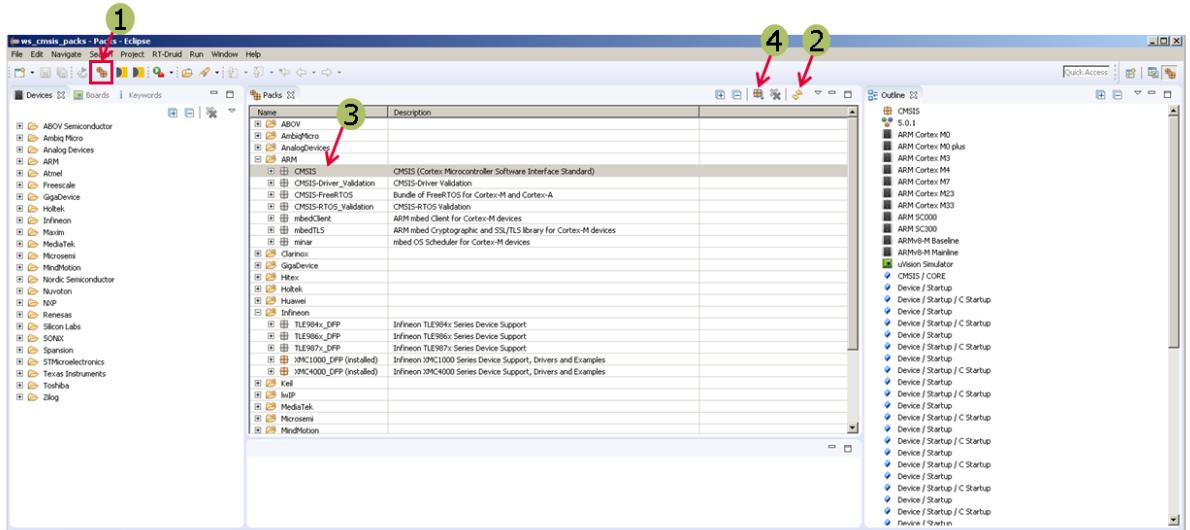
In eclipse, go to Help>Install New Software... and click on Add..., select Archive and browse to the previous downloaded file. Check all items and click on Next

Setup the Packages folder path in the Preferences eclipse menu as shown in the next pictures.



## 5. Install CMSIS Software Packs

1. Make the C/C++ Packs perspective visible.
2. Update the packages definitions from all repositories.
3. Select the CMSIS Software packs, i.e. select CMSIS, XMC4000\_DFP and XMC1000\_DFP entries holding CTRL key down.
4. Install a local copy of the selected packages.



## 6. Install J-Link debugger software

Download and install the latest version from  
<https://www.segger.com/downloads/jlink>

**J-Link Software and Documentation Pack**

- All-in-one debugging solution
- Can be downloaded and used free of charge by any owner of a SEGGER J-Link, J-Trace, Flasher ARM or Flasher RX model. Not all features of it may be available on all J-Link / J-Trace / Flasher models
- Updated frequently
- [Release Notes](#)
- [More information](#)

[Click for downloads](#)

Download	File Name	Size	Date
<a href="#">Download</a>	J-Link Software and Documentation pack for Windows	V6.14b 24,015 KB	<a href="#">Older versions</a> [2017-03-09]
<a href="#">Download</a>	J-Link Software and Documentation pack for MacOSX	V6.14b 41,966 KB	<a href="#">Older versions</a> [2017-03-09]
<a href="#">Download</a>	J-Link Software and Documentation pack for Linux, DEB Installer, 32-bit	V6.14b 10,523 KB	<a href="#">Older versions</a> [2017-03-09]
<a href="#">Download</a>	J-Link Software and Documentation pack for Linux, DEB Installer, 64-bit	V6.14b 16,878 KB	<a href="#">Older versions</a> [2017-03-09]
<a href="#">Download</a>	J-Link Software and Documentation pack for Linux, RPM Installer, 32-bit	V6.14b 10,530 KB	<a href="#">Older versions</a> [2017-03-09]
<a href="#">Download</a>	J-Link Software and Documentation pack for Linux, RPM Installer, 64-bit	V6.14b 14,081 KB	<a href="#">Older versions</a> [2017-03-09]
<a href="#">Download</a>	J-Link Software and Documentation pack for Linux, TGZ archive, 32-bit	V6.14b 10,601 KB	<a href="#">Older versions</a> [2017-03-09]
<a href="#">Download</a>	J-Link Software and Documentation pack for Linux, TGZ archive, 64-bit	V6.14b 16,957 KB	<a href="#">Older versions</a> [2017-03-09]
<a href="#">Download</a>	J-Link Software and Documentation pack for Linux ARM systems	V6.14b 10,438 KB	<a href="#">Older versions</a> [2017-03-09]

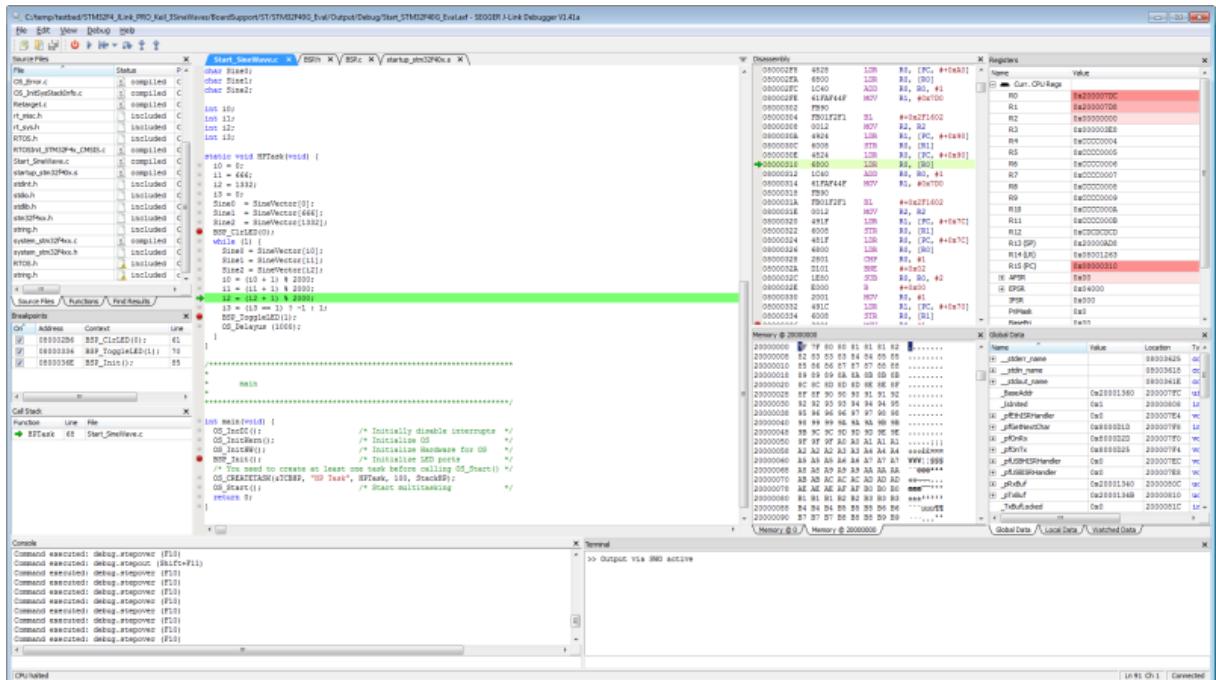
**Note:** This package comes without any support.

Alternatively you can also use OpenOCD

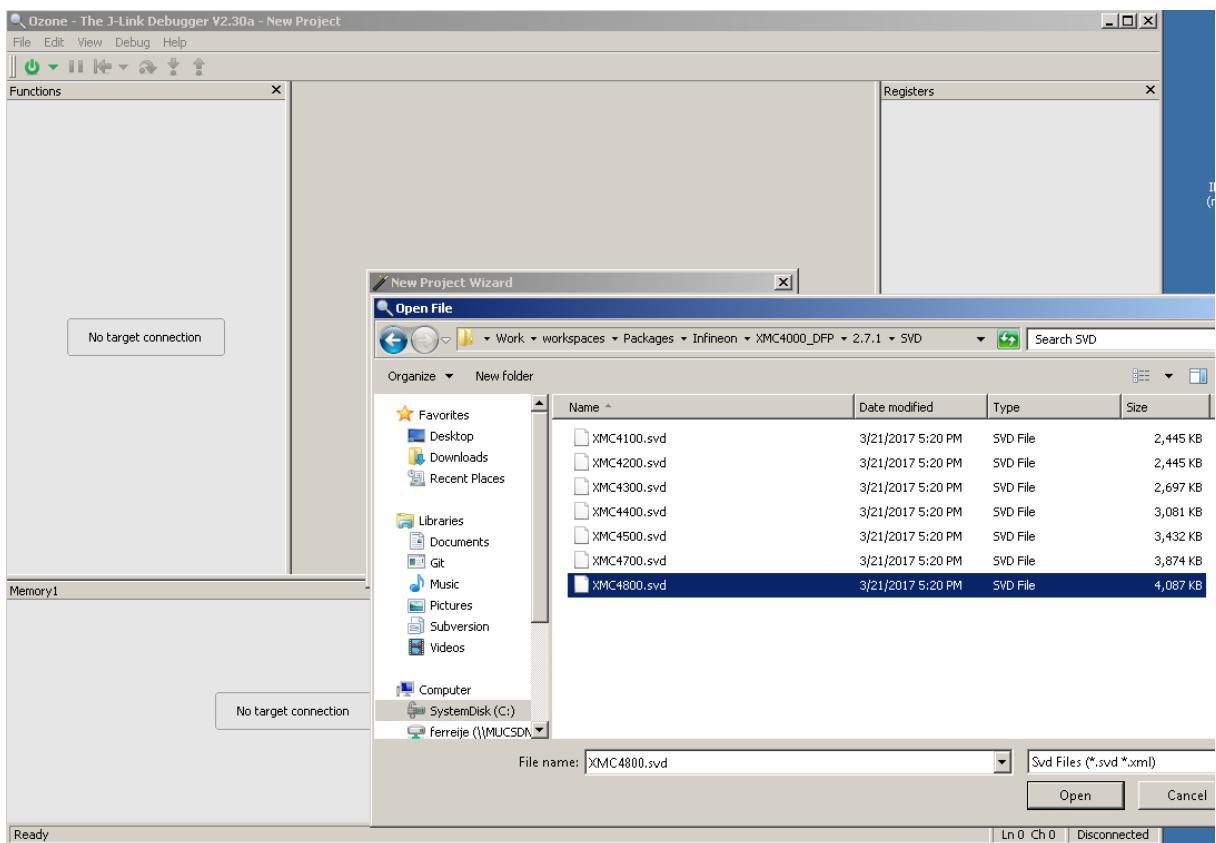
<https://github.com/gnuarmeclipse/openocd/releases>

## 7. Install Ozone debugger

Download and install the latest version from  
<https://www.segger.com/downloads/jlink>

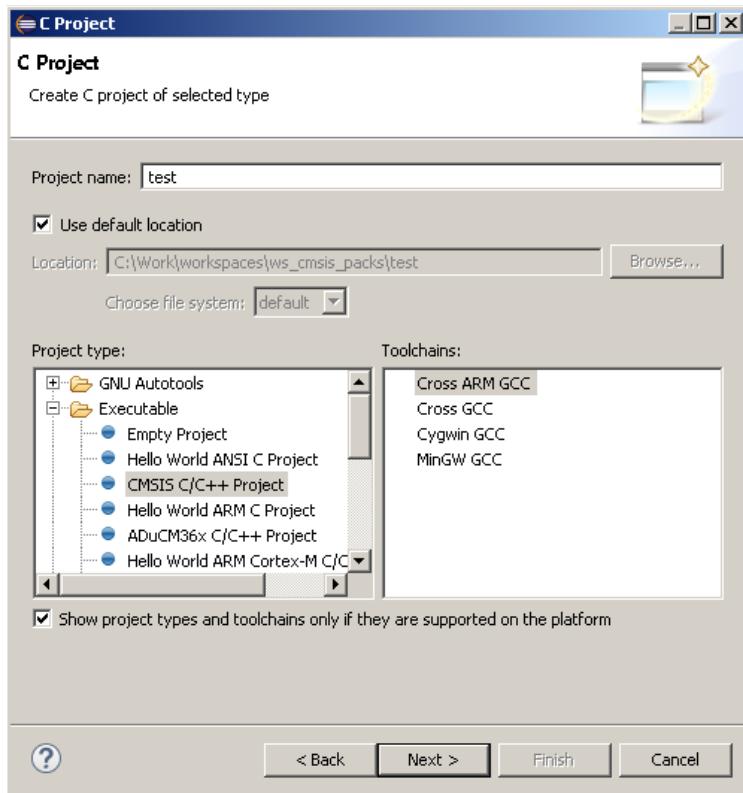


When creating a new project in Ozone, a peripheral description file (SVD) is optionally required. The files for each of the XMC™ microcontroller family members can be found at in the SVD folder in the XMC4000\_DFP or XMC1000\_DFP installation as shown below.



## 8. Creating a project

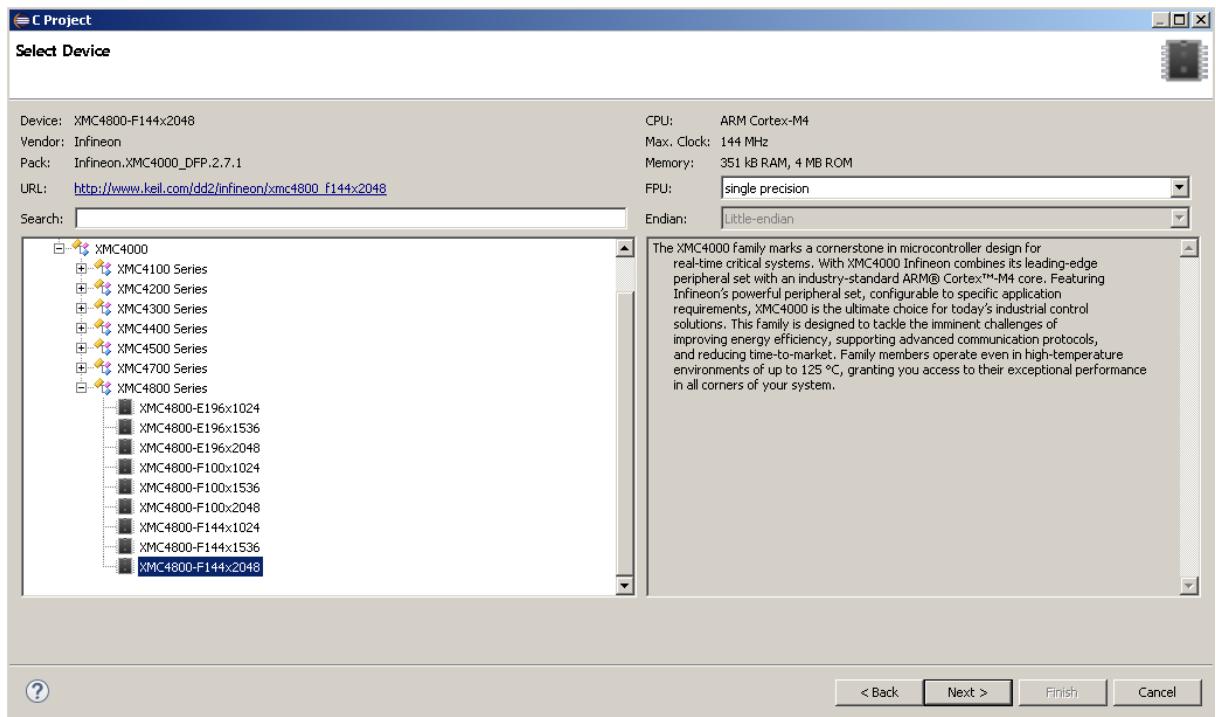
1. In eclipse, click File>New>Project, select C Project.
2. Give a name to the project
3. Select CMSIS C/C++ Project as Project type.
4. Click Next



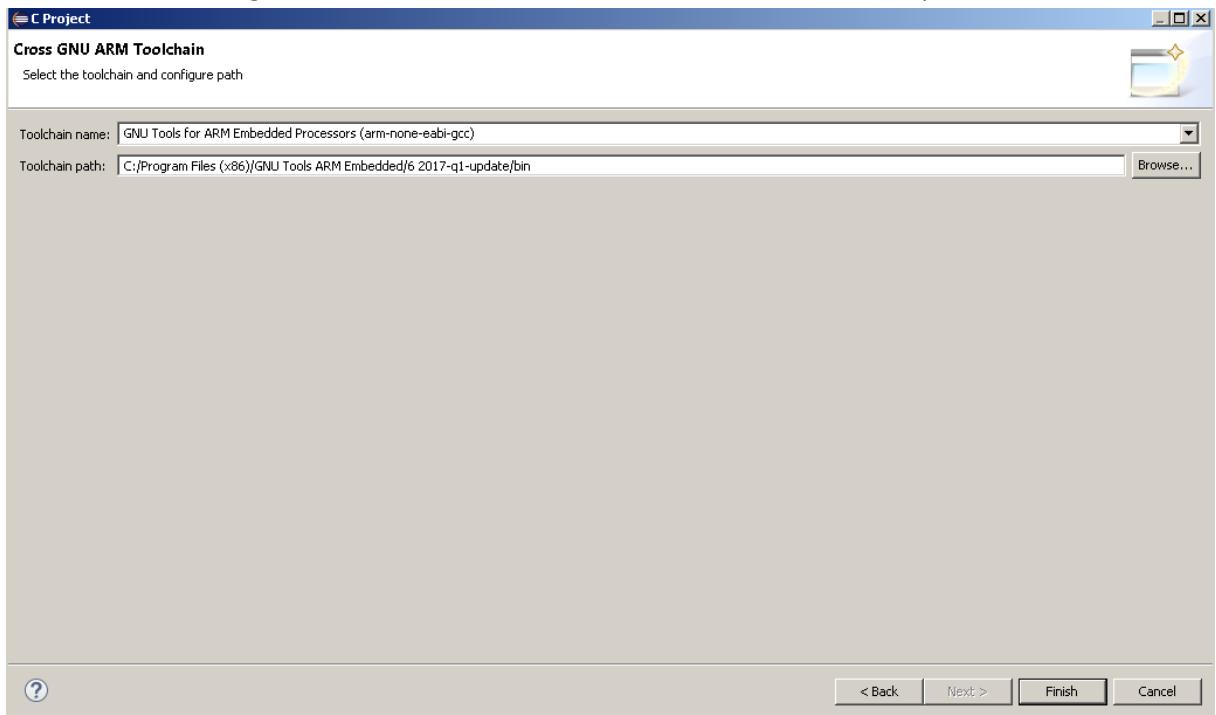
5. In the next dialog enable the creation of default main.c. Click Next.



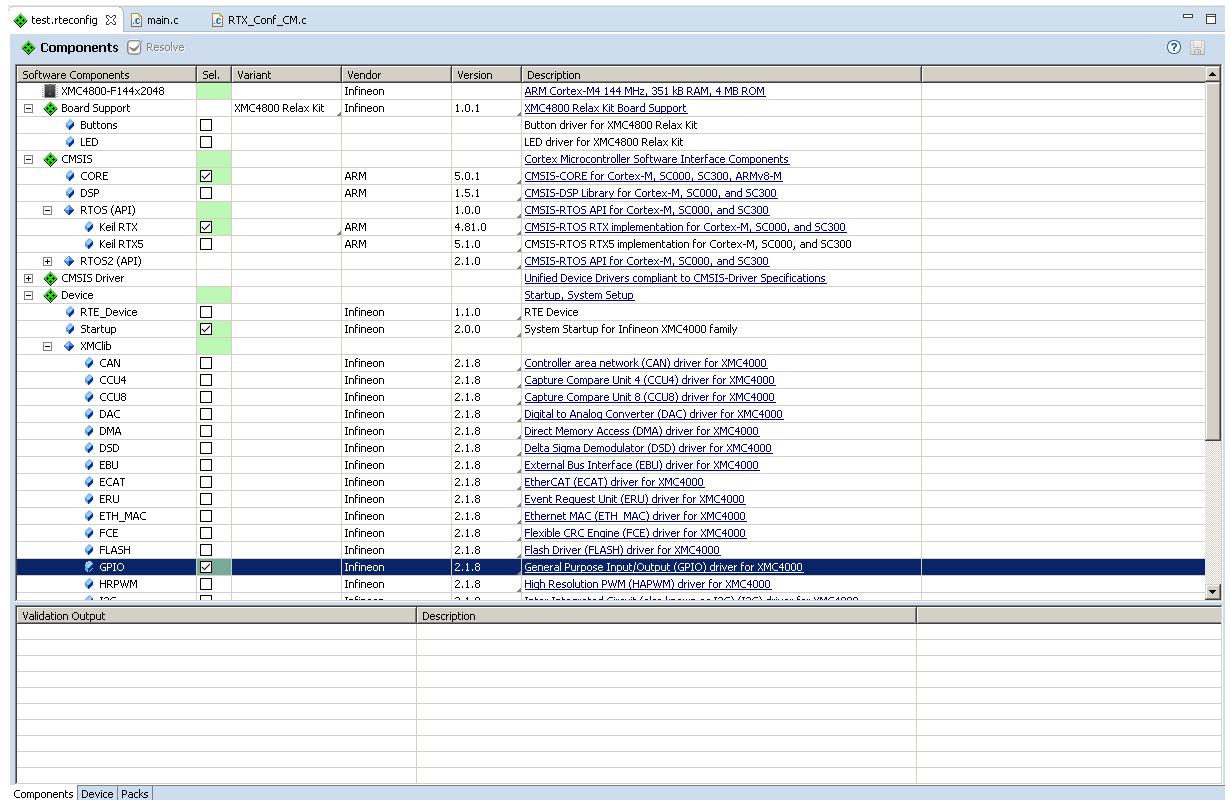
6. In the next dialog, select the device and click next



7. In the next dialog crosscheck that the GNU GCC ARM is detected correctly. Click Finish.



8. At this point you should see the RTE configuration page. Make the selections as shown below, click on resolve button to satisfy dependencies and save the changes.



9. The CMSIS RTOS RTX Pack need to be configured. Open the file RTX\_Conf\_CM.c and change the OS\_CLOCK to 144000000. This the default frequency of the XMC4800 after initialization.

10. Open main.c file, and make the modifications as below.

```
#ifdef _RTE_
    #include "RTE_Components.h"                                // Component selection
#endif
#ifndef RTE_CMSIS_RTOS
    #include "cmsis_os.h"                                       // when RTE component CMSIS RTOS is used
#endif
#include "xmc_gpio.h"

#define LED1 P5_9
#define LED2 P5_8

void Thread_1(void const *arg);                                // function prototype for
Thread_1
osThreadDef(Thread_1, osPriorityNormal, 1, 0);                // define Thread_1

void Thread_1(void const *arg)
{
    while (true)
    {
        XMC_GPIO_ToggleOutput(LED1);
        XMC_GPIO_ToggleOutput(LED2);

        osDelay(1000);
    }
}
```

```

}

/* main function */
int main(void)
{
#ifdef RTE_CMSIS_RTOS                         // when using CMSIS RTOS
    osKernelInitialize ();                      // initialize CMSIS-RTOS
#endif

/* Initialize device HAL here */
XMC_GPIO_CONFIG_t gpio_config;
gpio_config.mode = XMC_GPIO_MODE_OUTPUT_PUSH_PULL;
gpio_config.output_level = XMC_GPIO_OUTPUT_LEVEL_HIGH;

XMC_GPIO_Init(LED1, &gpio_config);

gpio_config.output_level = XMC_GPIO_OUTPUT_LEVEL_LOW;
XMC_GPIO_Init(LED2, &gpio_config);

osThreadCreate(osThread(Thread_1), NULL);

#ifdef RTE_CMSIS_RTOS                         // when using CMSIS RTOS
// create 'thread' functions that start executing,
// example: tid_name = osThreadCreate (osThread(name), NULL);
osKernelStart ();                            // start thread execution
#endif

/* Infinite loop */
while (1)
{
    /* Add application code here */
}
}

```

11. Compile and download the code and debug your code. You can use the GDB client included in eclipse following the instructions found at <http://gnuarmeclipse.github.io/debug/>. Alternatively you can use the previously installed Ozone debugger. Please refer to the instruction in the user manual.

## Other tools

[XMC™ Flasher](#)

## References

<http://gnuarmeclipse.github.io/>

<https://developer.arm.com/open-source/gnu-toolchain/gnu-rm>

