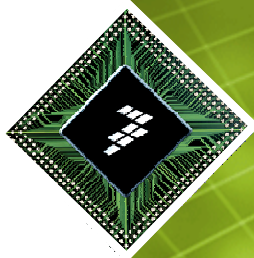


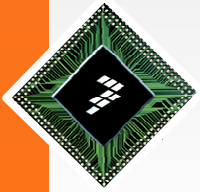


FRDM-KL25Z

Programming and Debugging in CodeWarrior

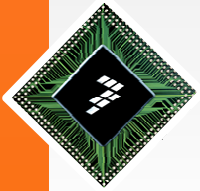


Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, the Energy Efficient Solutions logo, mobileGT, PowerQUICC, QorIQ, StarCore and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, ColdFire+, CoreNet, Flexis, Kinetic, MXC, Platform in a Package, Processor Expert, QorIQ Qonverge, Qoriva, QUICC Engine, SMARTMOS, TurboLink, VortiQa and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2011 Freescale Semiconductor, Inc.



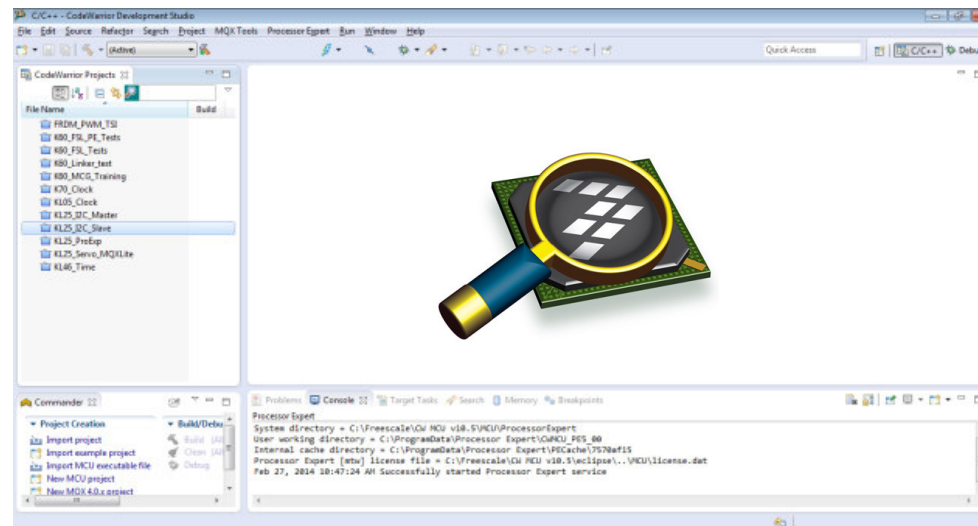
Agenda

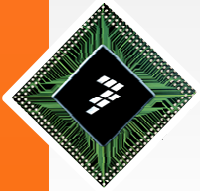
1. What is CodeWarrior?
2. Creating a new Project
3. C/C++ Perspective
4. Debugging a Project
5. Debug Perspective
6. Settings
7. Flashing a Binary File
8. Running Example



What is CodeWarrior?

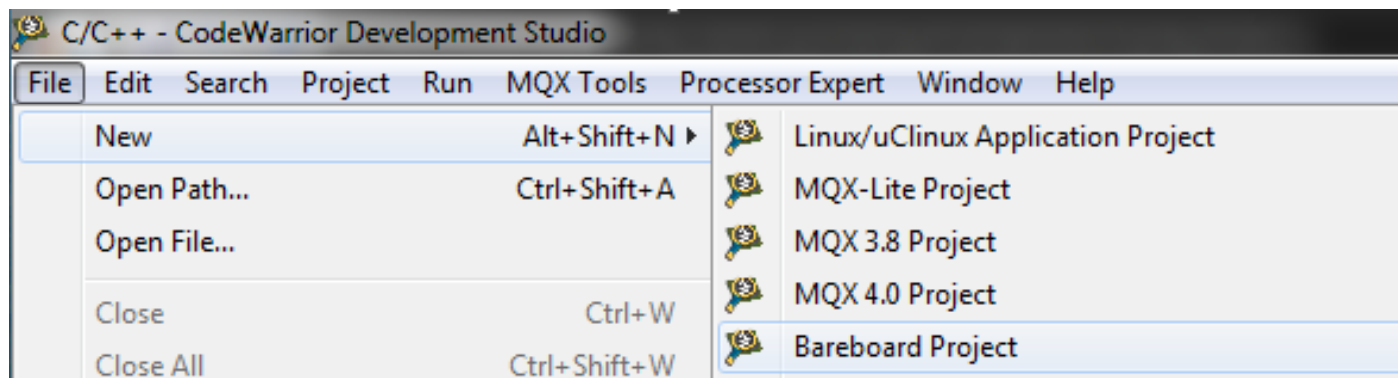
- Complete Integrated Development Environment (IDE).
- Based on Eclipse platform
- Visual Interface and automated framework
- Compiler, assembler and debugging support.
- Tracing support
- Processor expert rapid development tool

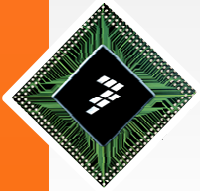




Creating a New Project

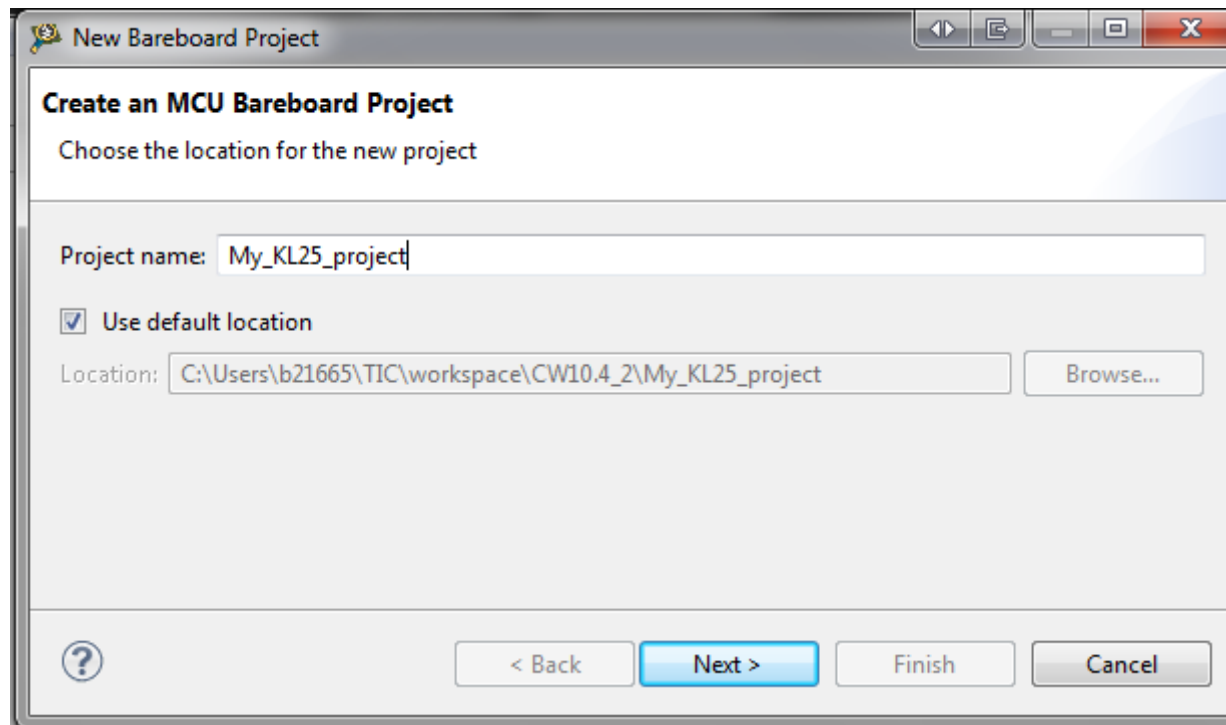
- Open CodeWarrior 10.3 or above
- Select menu File > New > Bareboard Project

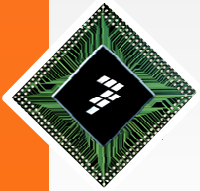




Creating a New Project

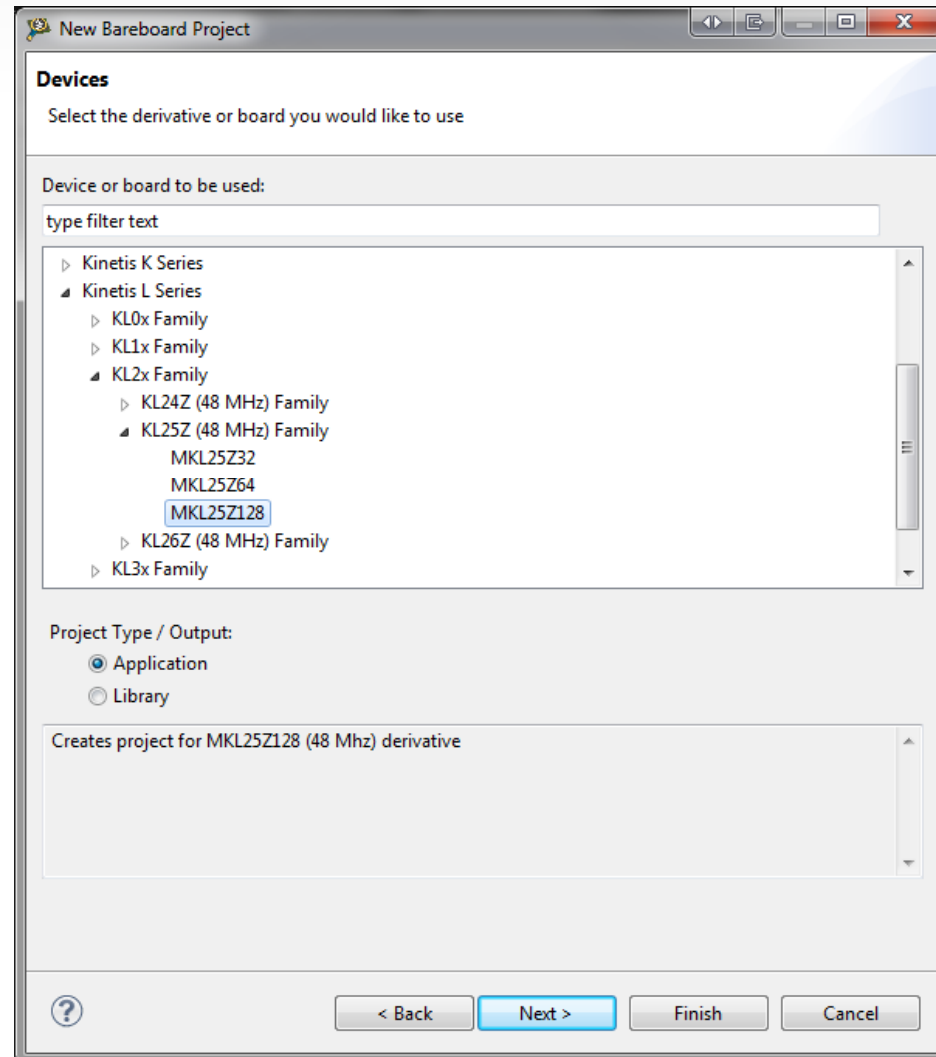
- In the next window write a name for your project and click 'Next >' button

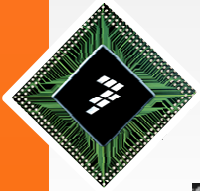




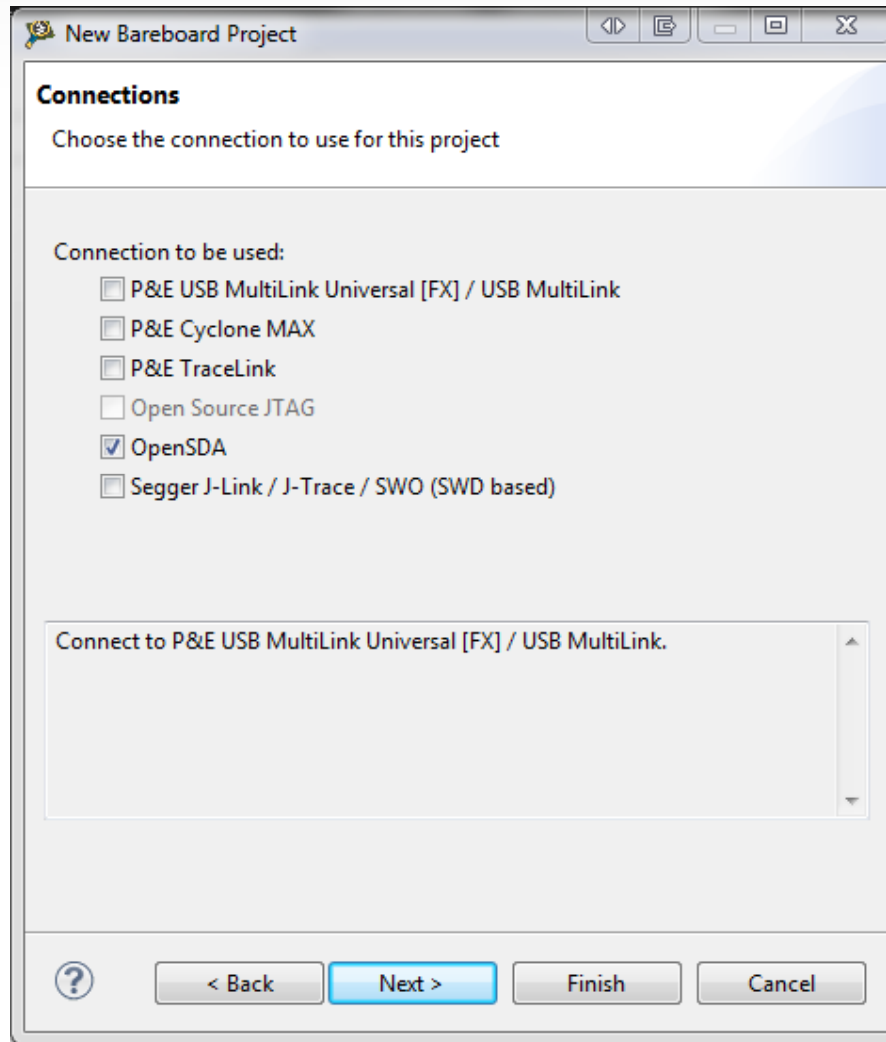
Creating a New Project

- Then select the MCU you will use. In this case MKL25Z128 is used.

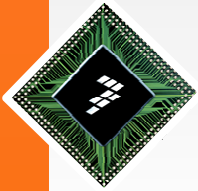




Creating a New Project

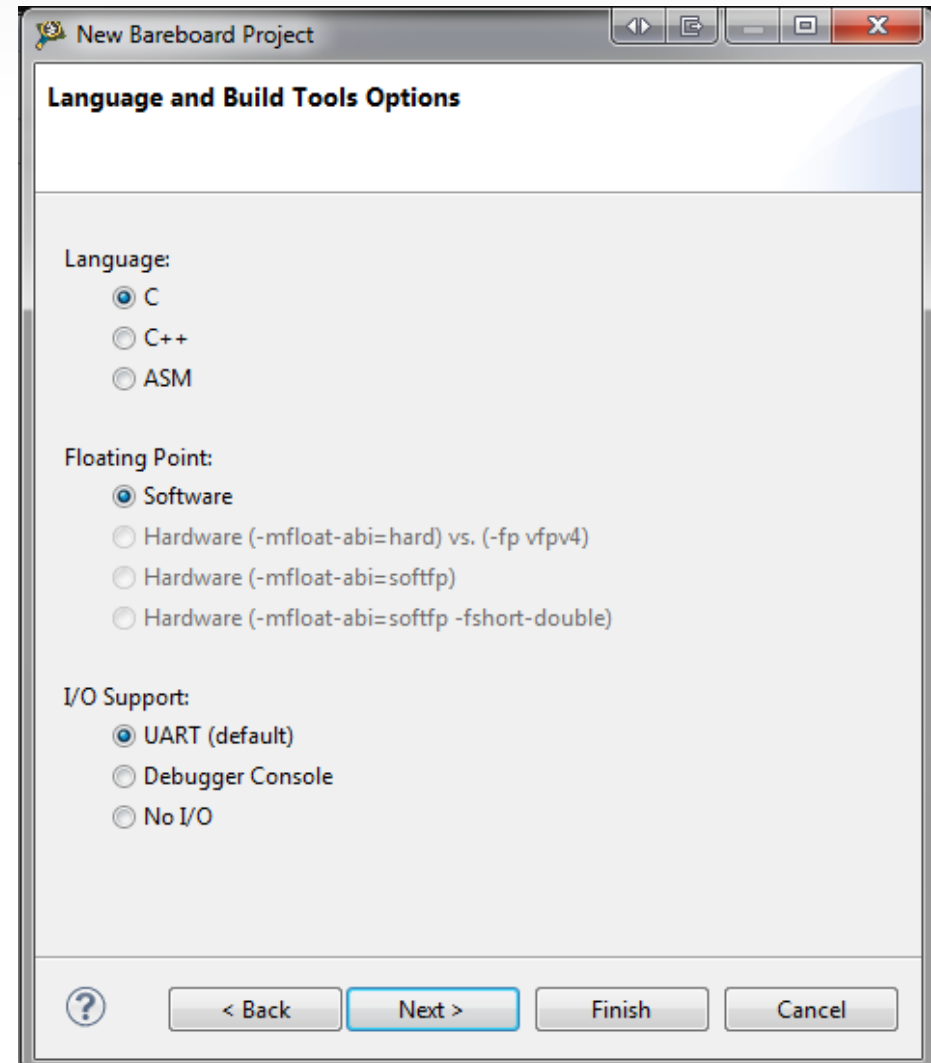


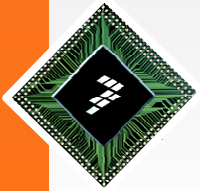
- Select the connection type that you are using, you can select as many connections as you want. In this lab only 'OpenSDA' is used.



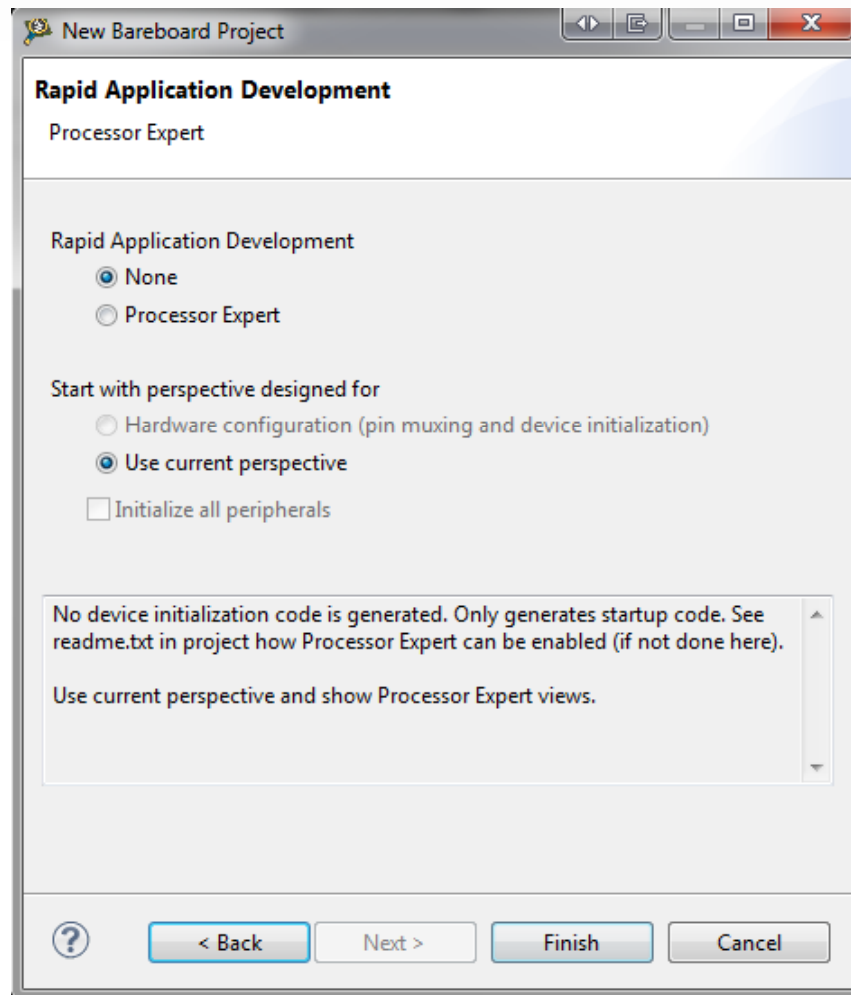
Creating a New Project

- In the next window you can set:
- Language
 - Select C, C++ or ASM according to your requirements. For this lab select C language.
- Floating Point support
 - KL25 does not feature Hardware Floating Point unit. Software is selected automatically.
- I/O Support
 - Codewarrior provides 3 different library options for I/O Support. Select UART for this lab.
 - You can change your library set later in menu Project > Properties > C/C++ Build > Settings > Librarian

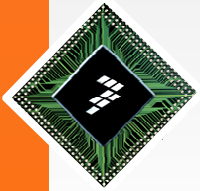




Creating a New Project



- Rapid Application Development features are not covered in this lab, select 'None' and click 'Finish' Button.



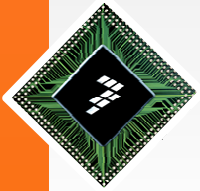
C/C++ Perspective

The screenshot displays the CodeWarrior Development Studio interface in C/C++ perspective. The main editor shows a C program named `main.c` with the following code:

```
1 /*  
2  * main implementation: use this 'C' sample to create your own application  
3  *  
4  */  
5  
6 #include "derivative.h" /* include peripheral declarations */  
7  
8 int main(void)  
9 {  
10     int counter = 0;  
11  
12     for(;;)  
13     {  
14         counter++;  
15     }  
16  
17     return 0;  
18 }  
19
```

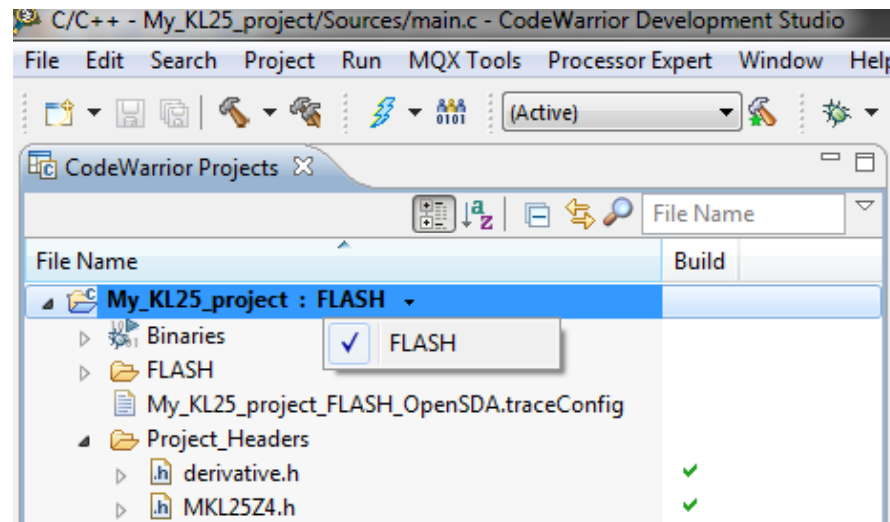
Annotations in the image include:


- A red box around the top toolbar, with an arrow pointing to the **C/C++** icon.
- A red box around the **Project Name** field in the Project Explorer, with an arrow pointing to **My_KL25_project : FLASH**.
- A red box around the **Views** area at the bottom, with an arrow pointing to the **Target Tasks** tab, which shows the output: `<terminated> ARM Processors, My_KL25_project.elf`.
- The word **Perspectives** is written in the upper right of the editor area, with an arrow pointing to the **Debug** icon in the top toolbar.
- The word **Views** is written in the lower center of the editor area, with an arrow pointing to the **Target Tasks** tab.

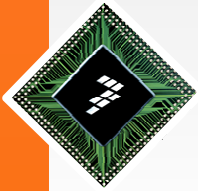


Debugging a Project

- Once your project is created you must choose a 'Build Configuration'. KL25 project only provide FLASH configuration, but other Freescale devices provide FLASH and RAM targets.

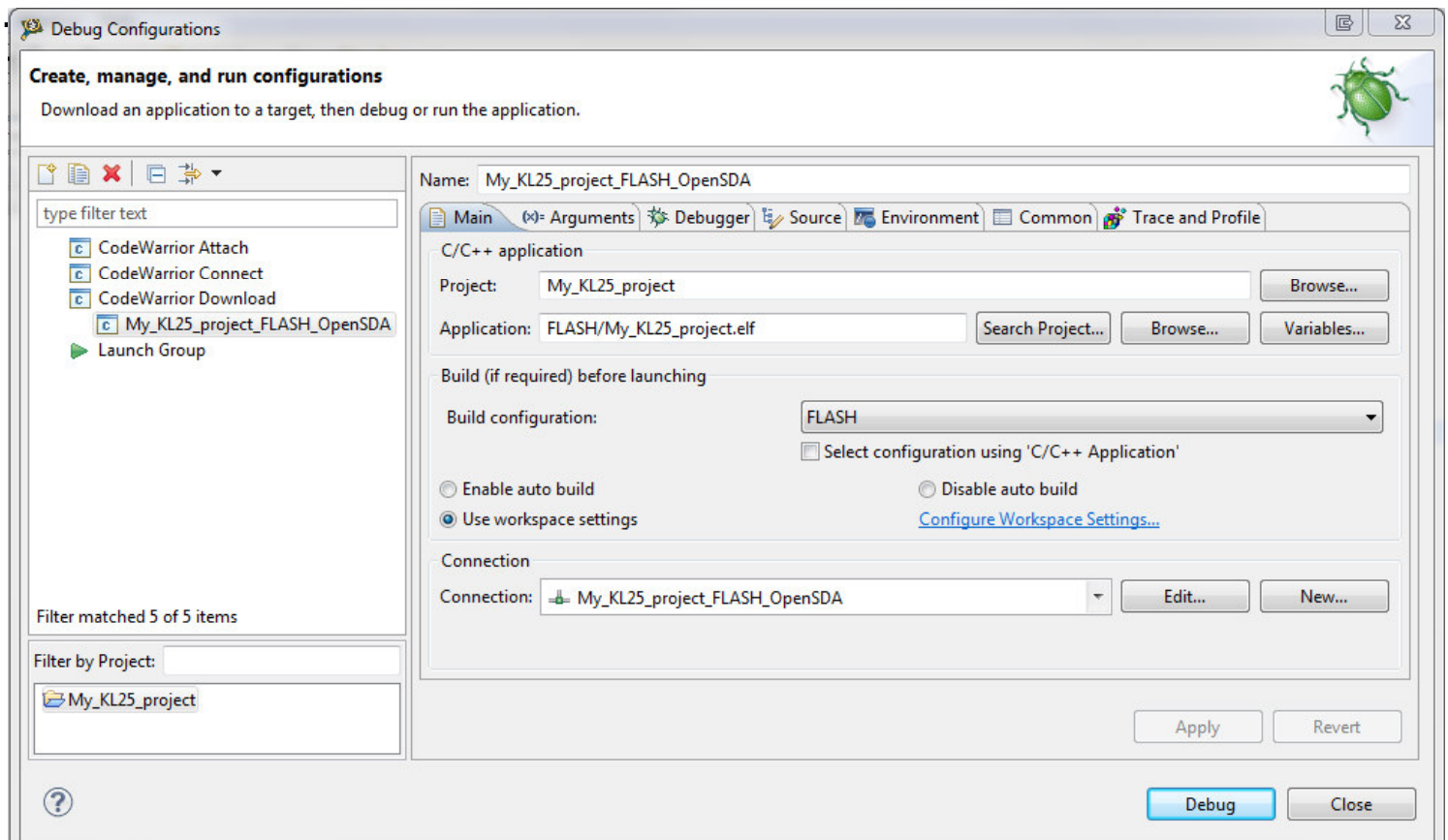


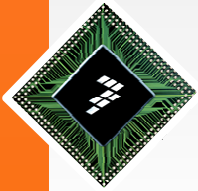
- To build the project, go to menu Project > Build Project, or alternately click the build button 



Debugging a Project

- To debug your project, go to menu Run > Debug Configurations... and select the Debug Configuration that matches with your Build Configuration and connection.
- Then click 'Debug'



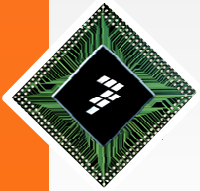


Debug Perspective

The screenshot displays the CodeWarrior Development Studio interface in the Debug Perspective. The top toolbar contains various debug tools, with a red box highlighting the 'Debug Tools' section. Below the toolbar, the 'Debug Views' panel is visible, containing tabs for Variables, Registers, Breakpoints, Memory, and Modules. A table in the Variables tab shows the following data:

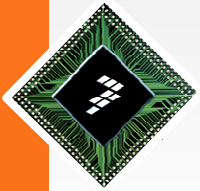
Name	Value	Location
(*) counter	0	0x20002ff4

The 'C Source View' shows the source code for main.c, with a red box highlighting the file name. The 'Disassembly View' shows the assembly code for the same function, with a red box highlighting the 'Disassembly' tab. The bottom of the interface features a 'Commander' panel with project management options and a 'Problems' panel with a table of 0 items.




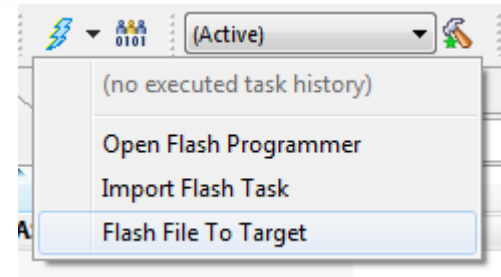
Settings

- To edit your project settings go to menu Project > Properties > C/C++ Build > Settings
- To open a new perspective go to menu Window > Open Perspective > Debug
- You can find all the views and open a new one in menu Window > Show View
- To edit CodeWarrior settings and preferences go to menu Window > Preferences

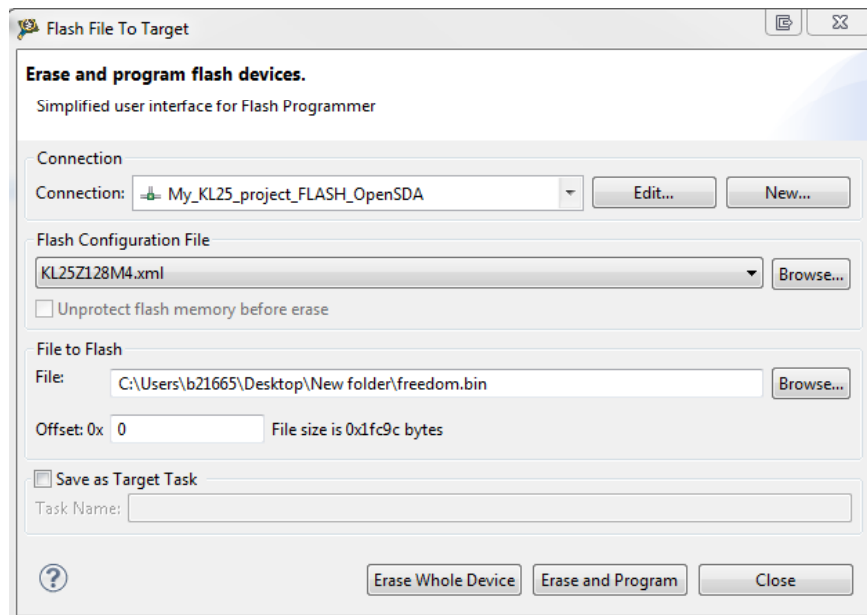


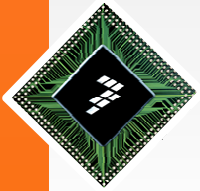
Flashing a Binary File

- Click the drop down arrow beside the Flash programmer button  to display the context menu and select 'Flash File to Target'.



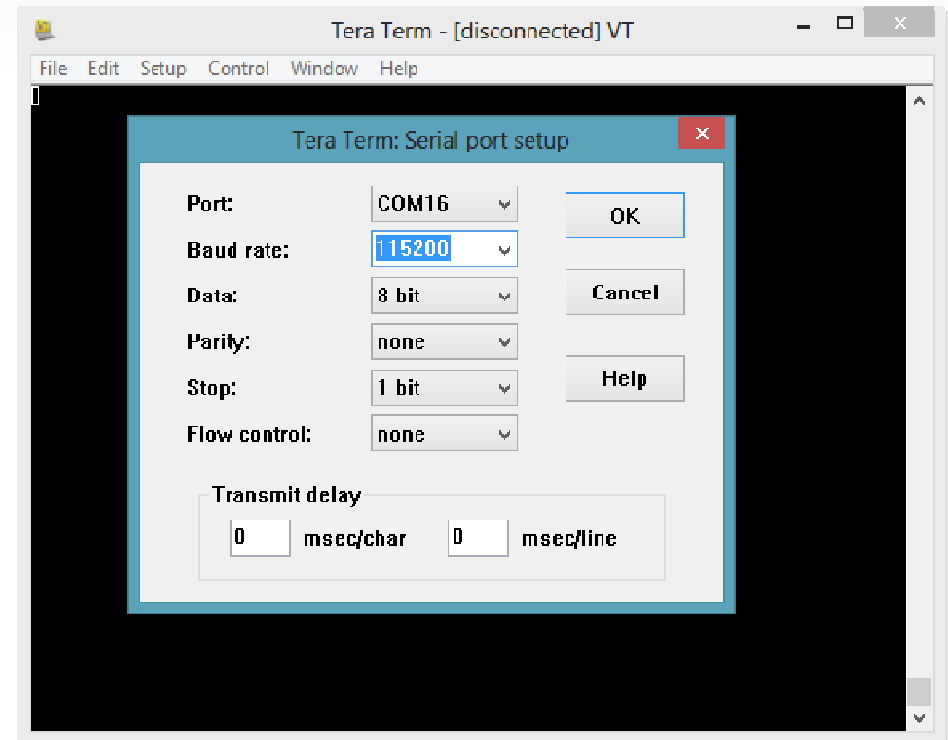
- Select your connection.
- In file to Flash search for freedom.bin
- Click 'Erase and Program'.

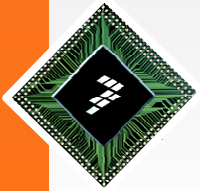




Running Example

- Open a terminal program. A good choice in my view is [TeraTerm](#)
- Connect to the board with 115200 baud



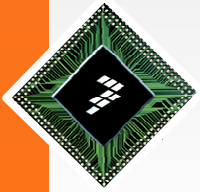


Run Example

- Push the onboard reset button. You will see next screen in terminal.

```
COM16:115200baud - Tera T...
File Edit Setup Control Window Help
      eeeee eeeee eeee 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
8e 8 88 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
88 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
88 e 8 8 88 8 88 88 8 88 8 e 88 88 88 e
88 88 8 88 88 8 88 88 88 88 88 88 88 88
Welcome to CoreBOOT on the Freedom Board.
Copyright (c) 2004-2012 Bayliber Associates Limited.

5,144 bytes free. Core at 40 MHz. Firmware 1.2.2.
>
```



Running Example

- Below you can find some instructions

```
print "Hello World!"
```

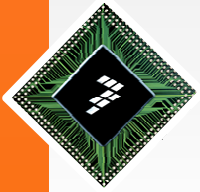
```
core.led = %red
```

```
core.led = %blue
```

```
core.led = %green
```

```
core.led = %black
```

Other colors are accepted too (see below or the [CoreBASIC manual](#)).

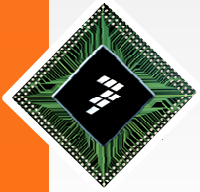


Running Example

- A LED blink loop. You can copy and paste the following code:

```
10 for i = 1 to 5
20 print i
30 core.led = %magenta
40 pause 0.5
50 core.led = %black
60 pause 0.5
70 next i
```

- The command to run a program is: of course *run*



Running Example

- Please visit the link below to see the whole example post.
- <http://mcuoneclipse.com/2013/01/18/back-to-basics-with-the-freedom-board/>

