This project demonstrates how to make an MCU expresso IDE's SDK project standalone.

To demonstrate this we use the frdmk22f development board.

**First create the SDK example project**

We started by creating a new-> sdk example -> select frdmk22f

A screenshot of a computer

Description automatically generated

Now select next and then expand the demo\_apps menu and select the "led\_blinky" project and click next

In the next window you will be prompted to select some project configurations such as floating point settings, compiler selection

And memory configuration ( this is where you would change the memory configuration if you want a non default memory configuration for your project.

A screenshot of a computer

Description automatically generated

**Now create the standalone project ( i.e not using the SDK )**

New -> Create a new C/C++ project

A screenshot of a computer

Description automatically generated

From this menu select Preinstalled MCUs and select the Generic -M4 and click next

A screenshot of a project

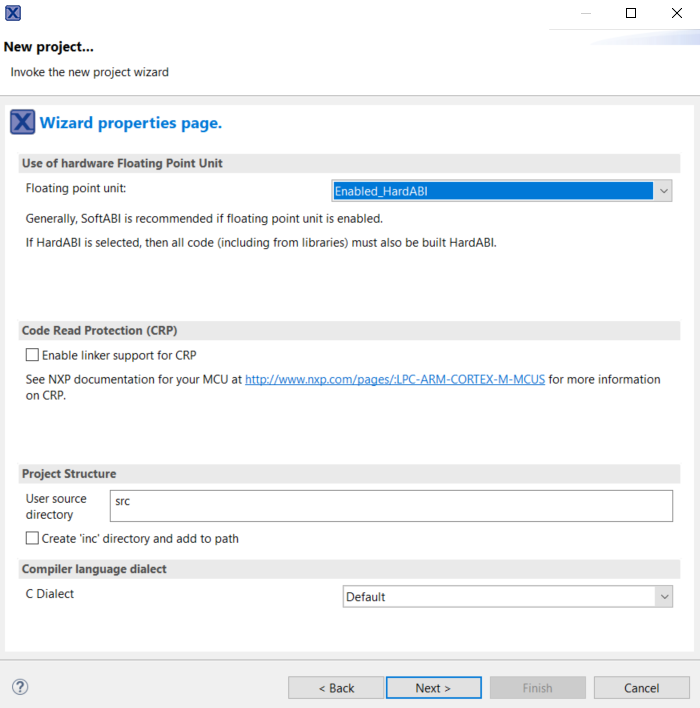
Description automatically generated

From this menus select "C Project" and click next

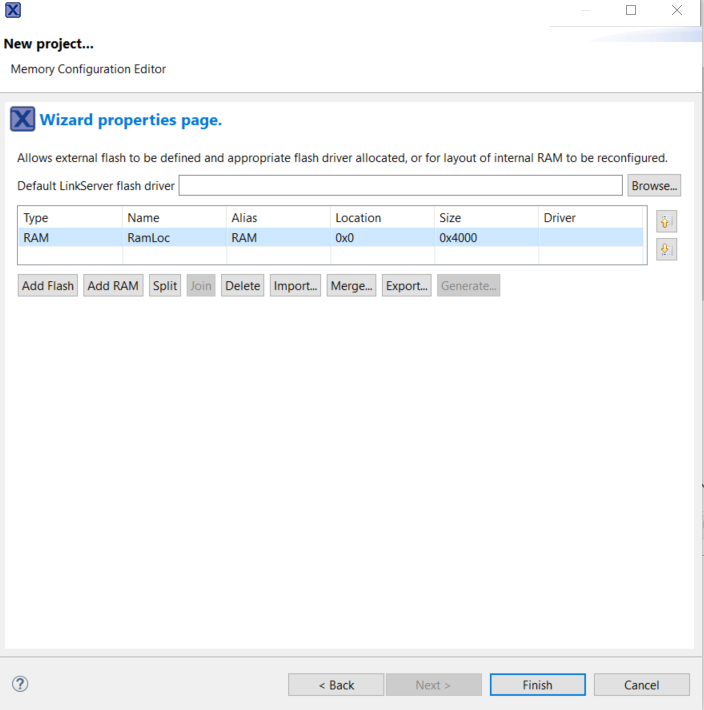
A screenshot of a computer program

Description automatically generated

In the project name menu type the name you want to give your standalone project and click next.



In this menu you can select the floating point support and the name of the directory where source is located. You may chose as appropriate for your project but make sure floating point support is similar to the SDK project settings you are going to import to this project.



In this screen you can select the FLASH and RAM allocations for your project to match the flash and RAM locations of the FRDM K22. (We will leave it unchanged for now and change it later in the process).

**Importing files and changing project settings**

The SDK based blinky project has the following files

A screenshot of a computer

Description automatically generated

The newly created standalone project has the following file structure

A screen shot of a computer

Description automatically generated

**Copy folders into standalone project**

Copy the CMSIS, board, component, device , drivers, startup, utilities, Debug folders into the standalone project location.

Copy the contents of **"SDK projects"/ source** folder into **"stand alone project" /src** location.

**Delete files inside standalone project**

Inside  **"stand alone project" /src** delete the following files inside cr\_startup\_cm4.c , flashconfig.c, K22Standalone.c

**Create a new folder called "stand alone project" /linker**

Create a new folder inside **"stand alone project" /**  called linker.

Now copy the following files inside **"SDK projects"/ Debug to "stand alone project" /linker**

Copy frdmk22f\_led\_blinky\_Debug.ld, frdmk22f\_led\_blinky\_Debug\_library.ld, frdmk22f\_led\_blinky\_Debug\_memory.ld into **"stand alone project" /linker**

**Changing standalone project settings**

* **Add #defines**

Open Properties-> C/C++ General -> Paths and Symbols now click the #Symbols

A screenshot of a computer

Description automatically generated

Now copy all the #defines and corresponding values from **"SDK projects" to "stand alone project"** , after copying the list should look like above.

* **Add the new source paths**

Open Properties-> C/C++ General -> Paths and Symbols now click the Source Location

A screenshot of a computer

Description automatically generated

Add the newly added folders that contain source code, it should look like above

* **Add the linker script**

Open Properties-> C/C++ Build -> Settings click Tool Settings tab, now click Managed Linker Script menu under MCU Linker

A screenshot of a computer

Description automatically generated

De select **Manage liner script** check box.

Add the folder where the linker is located in the **Script path** box which should be ../linker/

The **Linker script** box should contain the name of the main linker file , "frdmk22f\_led\_blinky\_Debug.ld"

**You should be able to build the project successfully now**

**The Debug configuration settings should look like this**

