



How to build BeeKit exported CW10.6 projects using Kinetis Design Studio 2.0.0

ABSTRACT:

This document provides the steps required to successfully build a CodeWarrior 10.6 BeeKit generated project for Kinetis KW2x wireless platform, using the Kinetis Design Studio (KDS)

AUTHOR(S)

SIGN-OFF SIGNATURE #1

SIGN-OFF SIGNATURE #2

Mihai-Andrei Dragnea



Contents

1.1	Purpose	3
1.2	Audience	3
1.3	Revision Information.....	3
1.4	Build Steps.....	4
1.5	Notes	13



1.1 Purpose

This document describes the steps required to successfully build a CodeWarrior 10.6 project that is exported from the Freescale BeeKit application and is intended to be build using Kinetis Design Studio (KDS), version 2.0.0.

1.2 Audience

This document is intended for software and system engineers that are willing to develop Freescale connectivity solutions based on IEEE 802.15.4 standard and are willing to use the Freescale BeeKit connectivity application for Kinetis KW2x wireless platform.

1.3 Revision Information

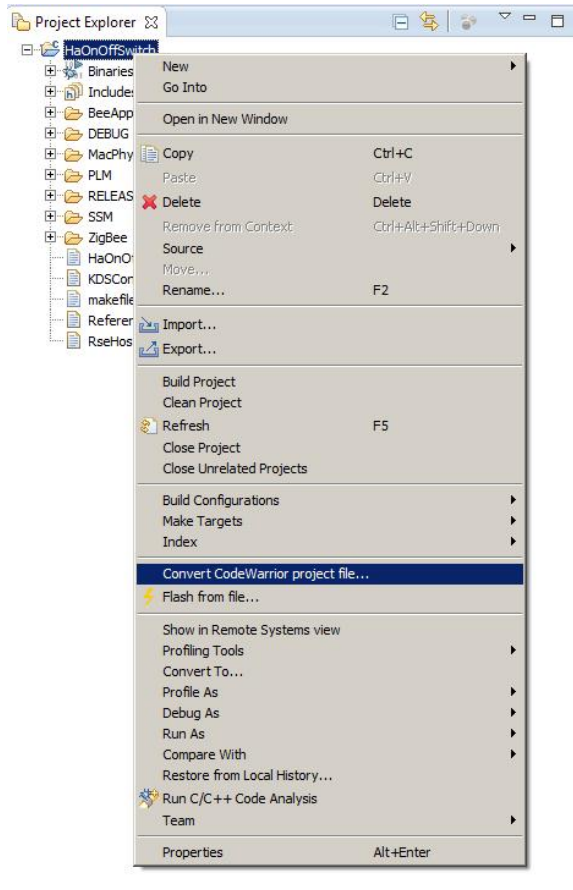
Date	Revision	Comments
March, 2015	Draft 1.0	Initial draft

1.4 Build Steps

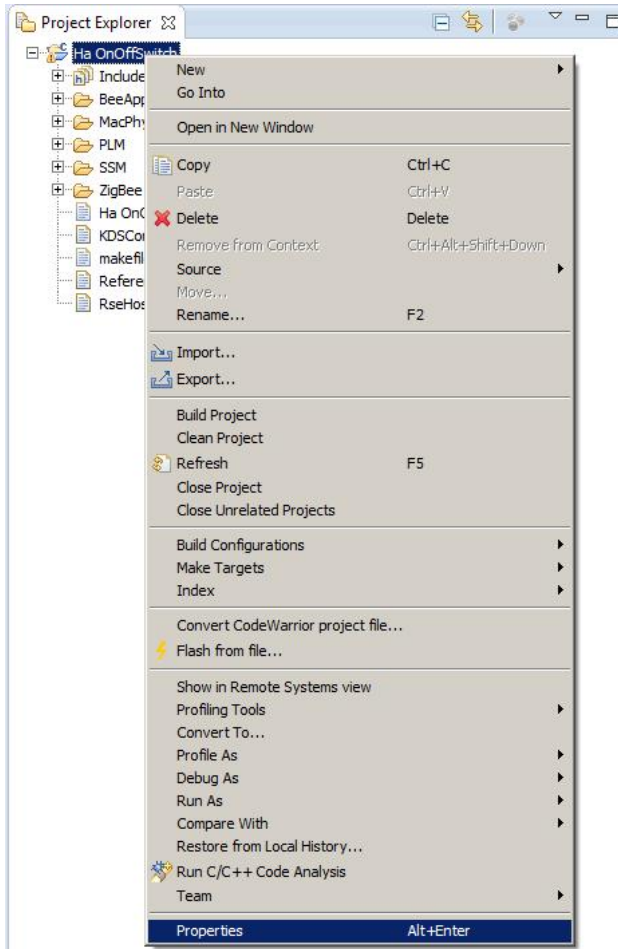
- Import CW 10.6 project under KDS

NOTE: The project name and project path shall not contain any spaces! Beekit default solution name and some of the projects name have spaces so you will need to make some rename operations.

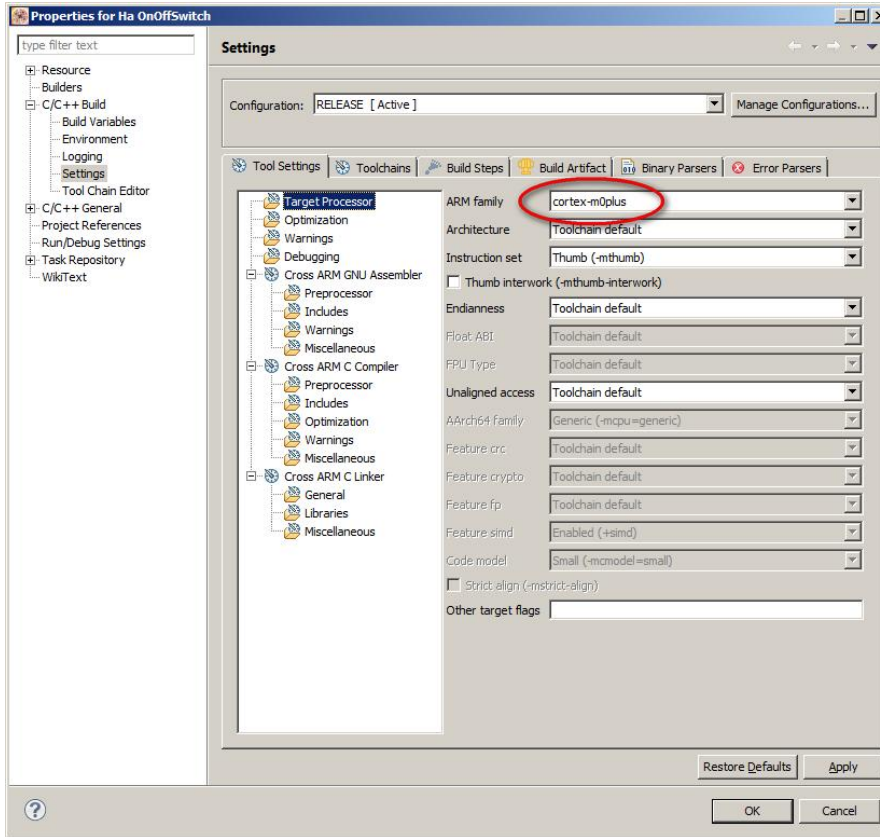
- *File > Import > General > Existing project into Workspace* or right click on the Project Explorer view and then *Import > General > Existing project into Workspace*
- Browse to CW10.6 project path and add the project into the KDS workspace. Select the option to copy project into workspace if you need it.
- Select the project, right click on it and click on “*Convert CodeWarrior project file ...*”
Acknowledge the conversion complete message box.

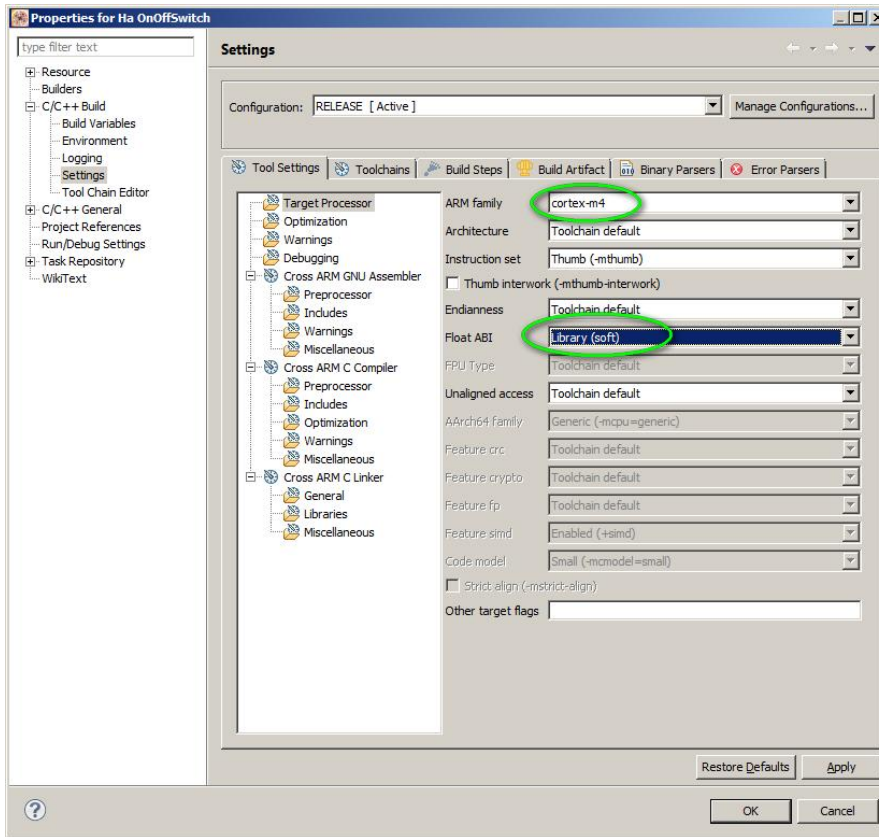


- Select the project again and right-click on it. Select “Properties”. The next steps needs to be performed for each configuration in project, if that configuration is to be used. Default configurations are “RELEASE” and “DEBUG”. Also, the “All configurations” selection is available, if all configurations are to be used.

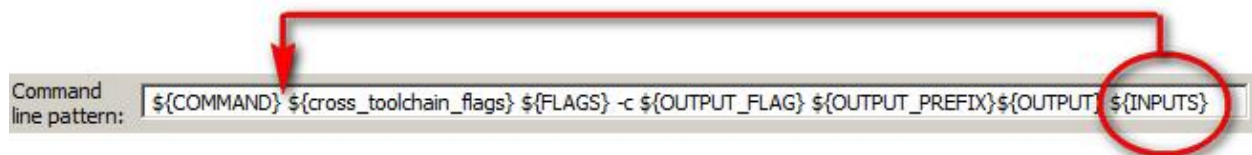


- The Properties view will open. Expand “C/C++ Build” and select “Settings”. On the right side of the view, under “Tool Settings” tab, select “Target Processor”. Modify the ARM family from Cortex M0+ to Cortex M4.





- Select "Cross ARM C Compiler". Modify the "Command line pattern" by moving the **`\${INPUTS}** from the end of the line just after the **`\${COMMAND}**:




- Next, select the “Includes” section under the “Cross ARM C Compiler”. Here, all the include paths needs to be modified by replacing the backslash ‘\’ with slash ‘/’. The quickest way is to directly edit (search/replace) the xml beyond the project, i.e. the “.cproject” file.

```

Include paths (-I)
${MCUToolsBaseDir}/ARM_GCC_Support/ewl/EWL_C/include*
${MCUToolsBaseDir}/ARM_GCC_Support/ewl/EWL_Runtime/include*
${ProjDirPath}/ZigBee/Interface/
${ProjDirPath}/BeeApps/
${ProjDirPath}/BeeApps/ZDO/Interface/
${ProjDirPath}/BeeApps/Configure/
${ProjDirPath}/BeeApps/StackGlobals/
${ProjDirPath}/BeeApps/ZCL/
${ProjDirPath}/BeeApps/HA/
${ProjDirPath}/BeeApps/HC/
${ProjDirPath}/BeeApps/SE/
${ProjDirPath}/BeeApps/ASL/
${ProjDirPath}/MacPhy\Interface\
${ProjDirPath}/MacPhy/Phy/Interface/
${ProjDirPath}/SSM/TS/Interface/
${ProjDirPath}/SSM/TS/Source/
${ProjDirPath}/SSM/ZTC/Interface/
${ProjDirPath}/SSM/ZTC/Source/
${ProjDirPath}/SSM/FunctionLib/Interface/
${ProjDirPath}/SSM/MsgSystem/Interface/

```

- Next step: move to “Cross ARM C Linker” section. Do the same operation as described at steps 5. Then, select “Libraries” and perform the following actions:

- Add the ‘librt’ library to “Libraries” by pressing the  button and by simply typing *rt*

```

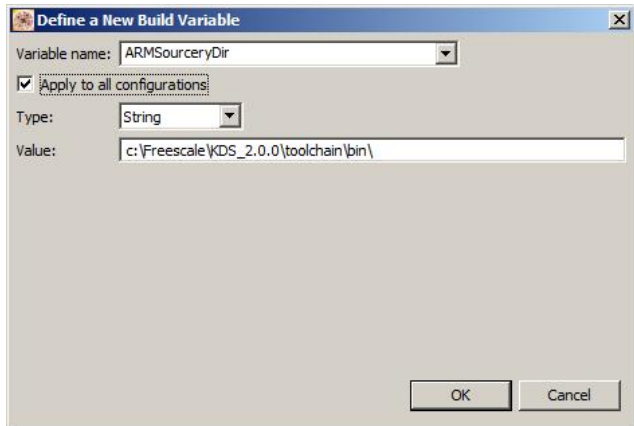
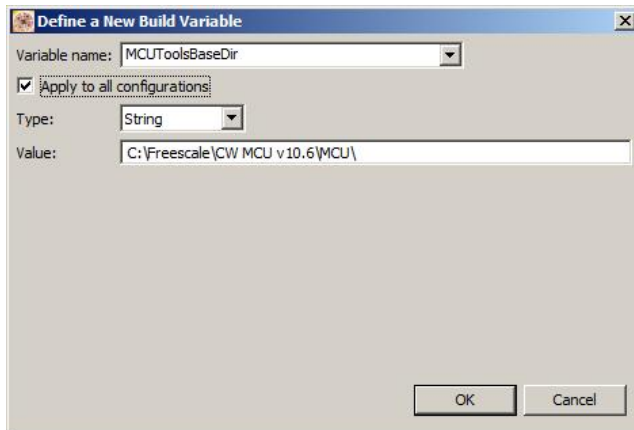
Libraries (-l)
rt
:lib_mmcau.a
:lib_crypto.a
:libecc.a
:libecc_sect283k1.a
:Freescale_802.15.4_MAC_MC1324x_GNU.a
:BeeStackZedRsRNPSp2_GNU.a

```

- Correct if necessary the libraries search path by replacing the ‘\’ with ‘/’, similar to step 6.

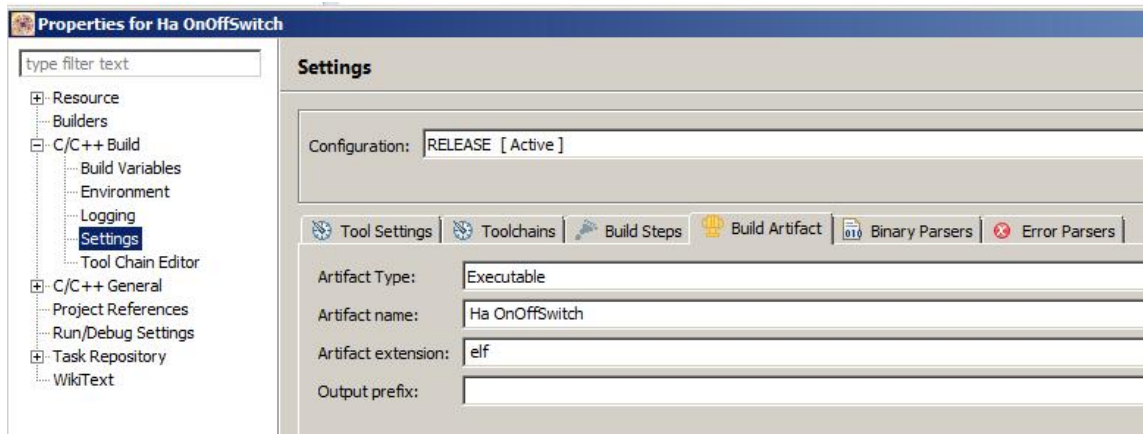
```
Library search path (-L)
"${MCUToolsBaseDir}/ARM_GCC_Support/ewl/lib/armv7e-m"
"${ProjDirPath}/MacPhy/Libs"
"${ProjDirPath}/PLM/Lib/MMCAU"
"${ProjDirPath}/PLM/Lib/CRYPTO"
"${ProjDirPath}/BeeApps/SE"
"${ProjDirPath}/ZigBee/Libs"
```

- On the same project “Properties” page, under “C/C++ Build”, select “Build Variables”. Add the following two variables, as depicted below. The Value is the default one.
 - ARMSourceryDir
 - MCUToolsBaseDir



Name	Type	Value
ARMSourceryDir	String	c:\Freescale\KDS_2.0.0\toolchain\bin\
MCUToolsBaseDir	String	C:\Freescale\CW MCU v10.6\MCU\

- Return to “C/C++ Build” and under “Settings” select the “Build Artifact” tab. Remove the quotes in the artifact name:



- Click “Apply” to have your changes saved and then press “OK”.
- Open the ‘.cproject’ file in a text editor, preferably with XML syntax highlight. Search for `__pformatter_` keyword and remove the entire option that contains it, like below:

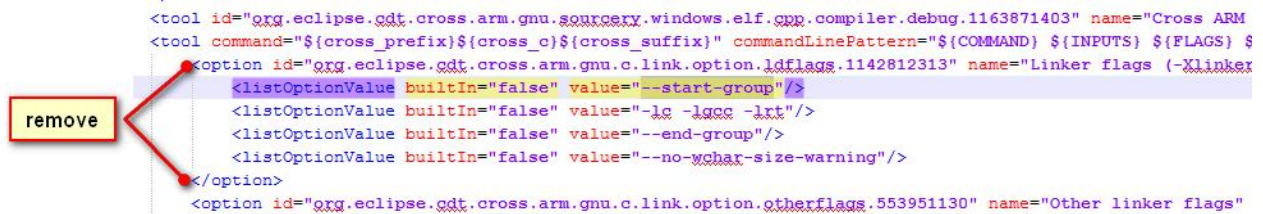
Remove

```

<tool id="org.eclipse.cdt.cross.arm.gnu.sourcery.windows.elf.c.linker.debug.84341242" name="Cross ARM C Linker" su
<option id="org.eclipse.cdt.cross.arm.gnu.c.link.option.ldflags.1142812313" name="Linker flags (-Wlinker [opti
<listOptionValue builtIn="false" value="--undefined=__pformatter__"/>
<listOptionValue builtIn="false" value="--defsym=__pformatter=__pformatter"/>
<listOptionValue builtIn="false" value="--start-group"/>
<listOptionValue builtIn="false" value="-lg -lgcc -lxt"/>
<listOptionValue builtIn="false" value="--end-group"/>
<listOptionValue builtIn="false" value="--no-wchar-size-warning"/>
</option>
<option id="org.eclipse.cdt.cross.arm.gnu.c.link.option.otherflags.553951130" name="Other linker flags" super
<option id="org.eclipse.cdt.cross.arm.gnu.c.link.option.nodeflibs.1342050394" name="Do not use default librari
<option id="org.eclipse.cdt.cross.arm.gnu.c.link.option.nostdlibs.1700108776" name="No startup or default libs
<option id="org.eclipse.cdt.cross.arm.gnu.c.link.option.gcsections.1555859167" name="Remove unused sections (-
<option id="org.eclipse.cdt.cross.arm.gnu.c.link.option.paths.2114571134" name="Library search path (-L) supe
<listOptionValue builtIn="false" value=""${MCUToolsBaseDir}/ARM_GCC_Support/ewl/lib/armv7e-m6""/>
<listOptionValue builtIn="false" value=""${ProjDirPath}\MacPhy\Libs""/>
<listOptionValue builtIn="false" value=""${ProjDirPath}\PLM\Lib\MMCAU""/>
<listOptionValue builtIn="false" value=""${ProjDirPath}\PLM\Lib\CRYPTO6""/>
<listOptionValue builtIn="false" value=""${ProjDirPath}\BeeApps\SE""/>
<listOptionValue builtIn="false" value=""${ProjDirPath}\ZigBee\Libs""/>

```

Similar, look for `--start-group` keyword and remove the entire option that contain this field:



```

<tool id="org.eclipse.cdt.cross.arm.gnu.sourcery.windows.elf.cpp.compiler.debug.1163871403" name="Cross ARM
<tool command="{cross_prefix}{cross_c}{cross_suffix}" commandLinePattern="{COMMAND} {INPUTS} {FLAGS} $
<option id="org.eclipse.cdt.cross.arm.gnu.c.link.option.ldflags.1142812313" name="Linker flags (-Xlinker
  <listOptionValue builtIn="false" value="--start-group"/>
  <listOptionValue builtIn="false" value="-lc -lgcc -lrt"/>
  <listOptionValue builtIn="false" value="--end-group"/>
  <listOptionValue builtIn="false" value="--no-wchar-size-warning"/>
</option>
<option id="org.eclipse.cdt.cross.arm.gnu.c.link.option.otherflags.553951130" name="Other linker flags"

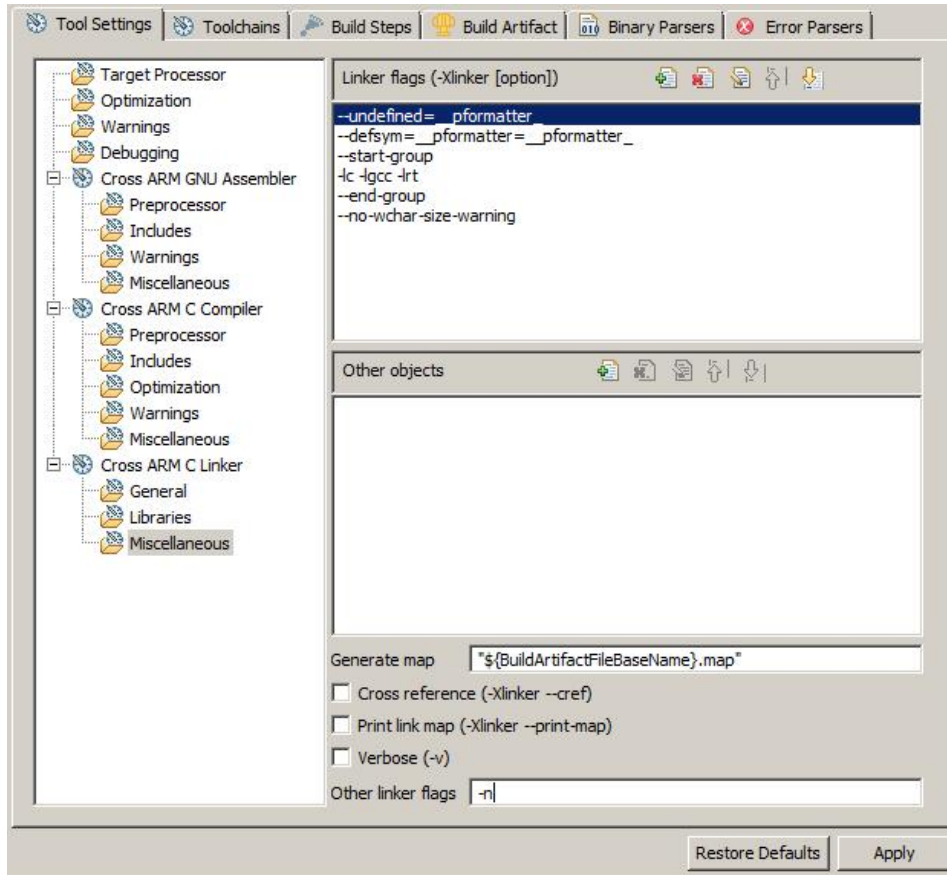
```

- Save the file and return to KDS workspace. Press 'F5' to refresh the project.
- Open again the project properties page, select the "Miscellaneous" options under "Cross ARM C Linker". Add the following linker flags:

```

--undefined=__pformatter_
--defsym=__pformatter=__pformatter_
--start-group
-lc -lgcc -lrt
--end-group
--no-wchar-size-warning

```



- Click “Apply” and “OK”.
- Build the project.

1.5 Notes

- CodeWarrior 10.6 for MCU is required to be installed, even without any license but default.
- When configuring the project properties it is recommended to click the “Apply” button from time to time to avoid losing of the already made changes, by mistake.
- All the information contained in this document is applicable only for BeeKit generated projects for Kinetis KW2x wireless platform.