Document Number: MQXRTOS
Rev. 02, 04/2013

How-to Change Default Clock Settings in Kinetis BSPs

PRODUCT:	Freescale MQX™ RTOS
PRODUCT VERSION:	4.0.1
DESCRIPTION:	Document describing clock settings changes in the Kinetis BSPs using the ProcessorExpert tool
RELEASE DATE:	April 19th, 2013

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

Document Number: MQXRTOS
Rev. 02, 04/2013

Contents

1 Introduction

The MQX 3.8 introduces the low power management features on Kinetis platforms. Key building blocks of this solution are the Low Power Manager and Clock Manager modules referred to as LPM and CM in this document. The Clock Manager allows runtime switching between clock configurations statically defined at the BSP level.

In the BSP Clock Manager *bsp_cm.h* header file, each clock configuration is assigned an index and symbolic name specified. See the BSP_CLOCK_CONFIGURATION enum type:

```
typedef enum {
    BSP_CLOCK_CONFIGURATION_0 = 0,  /* 96 MHZ PEE    */
    BSP_CLOCK_CONFIGURATION_1,      /* 12 MHz PEE    */
    BSP_CLOCK_CONFIGURATION_2,      /*  2 MHz BLPI   */
    BSP_CLOCK_CONFIGURATIONS        /* Number of available clock
                                       configurations */
} BSP_CLOCK_CONFIGURATION;
```

The values of the enum type are used as an input parameter to generic CM functions as well as to the BSP low level CM functions. See the MQX API Reference Manual for the complete description of Clock Manager functions.

```
_cm_set_clock_configuration();
_cm_get_clock_configuration();
_cm_get_clock();
```

There is the low level implementation of the CM functions in the BSP *bsp_cm.c* file. This implementation contains the following functions:

```
_bsp_initialize_hardware();
_bsp_set_clock_configuration();
_bsp_get_clock_configuration();
_bsp_get_clock();
```

There is also an optional function used to trim the internal oscillator:

```
_bsp_osc_autotrim();
```

The _bsp_initialize_hardware() function is responsible for basic MCU setting and setting of the Multipurpose Clock Generator (MCG) module. It switches the MCG module from the reset state through the MCG state machine to the default clock configuration indexed as BSP_CLOCK_CONFIGURATION_0. See more information about the MCG Mode State Diagram in the reference manual for the selected Kinetis platform.

The _bsp_set_clock_configuration() function is responsible for runtime switching between defined clock configurations.

The _bsp_get_clock_configuration() function returns the index of active clock configuration.

The _bsp_get_clock() function returns the clock in Hz of the selected clock source (CM_CLOCK_SOURCE) for a given clock configuration (BSP_CLOCK_CONFIGURATION). In MQX BSPs this function is implemented as a static look up table for a fast access.

When the BSP and the target application use internal oscillator, the _bsp_osc_autotrim() function has to be implemented to adjust its accuracy. See the MCG Auto TRIM feature described in the “Clock Modules” section in the Kinetis reference manual.

2 Default Clock Configuration in MQX BSPs

By default the MQX supports the following target boards with crystals and oscillators:

Kwikstik	8MHz crystal
TWR-K40X256	4MHz crystal
TWR-K53N512	50MHz oscillator
TWR-K60N512	50MHz oscillator

Three predefined clock configurations were selected as common for all target boards mentioned above:

- 96MHz normal run mode (MGG PEE mode)
- 12MHz normal run mode (MCG PEE mode – used also for auto-trimming the internal oscillator)
- 2MHz low power run mode (MCG BLPI mode)

The clock setting configurations were defined in the ProcessorExpert tool. The PE project files are located in the *"build\cw10\bsp_<board>\ProcessorExpert.pe"* file. The resulting generated code was put into the CM BSP *"source\bsp\<board>\bsp_cm.*"* source files.

3 Using ProcessorExpert to Create Custom Clock Configuration

When creating or cloning the BSP for a custom board, it is often necessary to modify the code inside the BSP low level CM functions to accommodate all clock settings requirements.

Generally there are two ways to achieve this:

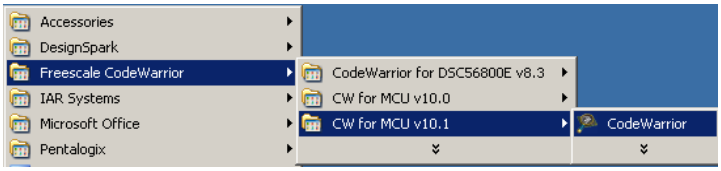
1. Manually: create or modify the low level functions in the *bsp_cm.c* and *.h* files.
2. Use ProcessorExpert tool to generate the clock setting code.

Manual modification of the code may be a difficult task as the current MQX CM code is already generated by the ProcessorExpert tool (this is the case for MQX 3.8, it may change in future versions).

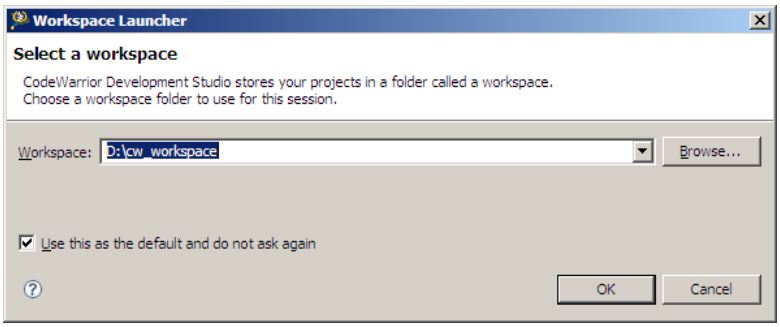
Note: The current implementation of ProcessorExpert limits the number of available clock configurations to 3.

Steps to use the ProcessorExpert tool to generate clock settings code:

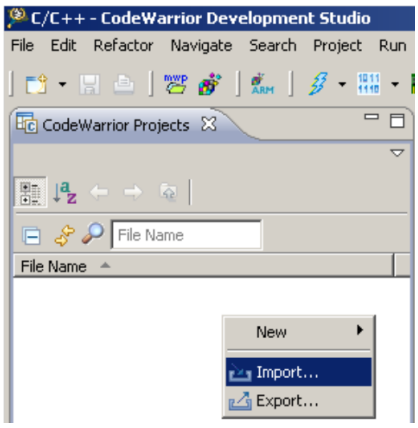
- Open CodeWarrior 10.x



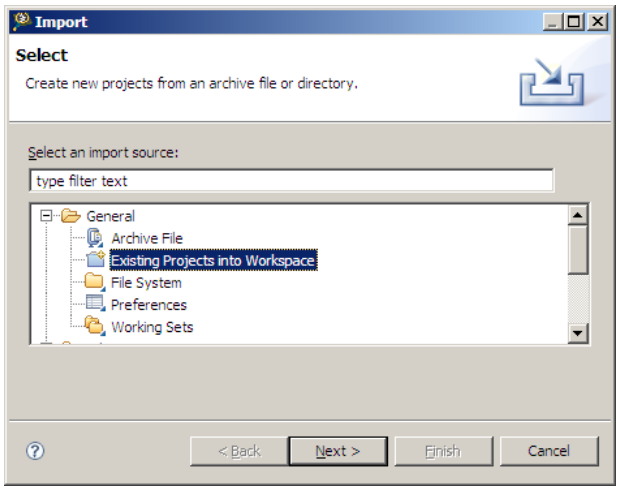
- Select your Workspace and press OK.

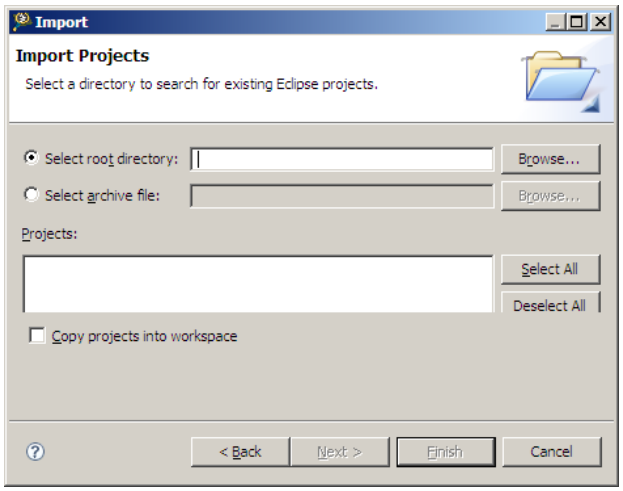


- Import the BSP project into workspace by Right-Clicking the Project Explorer view and selecting *Import*.

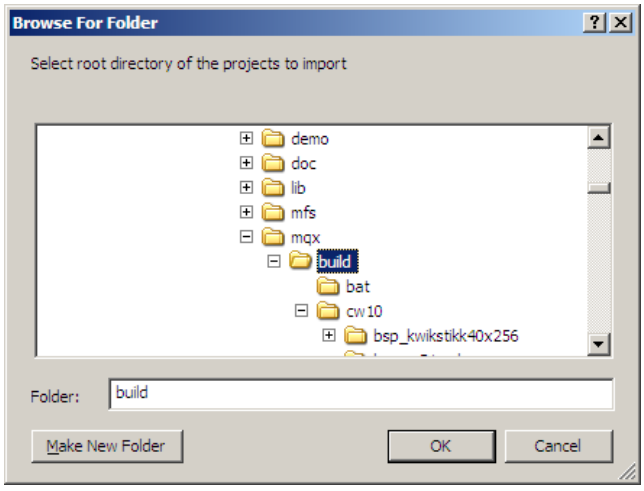


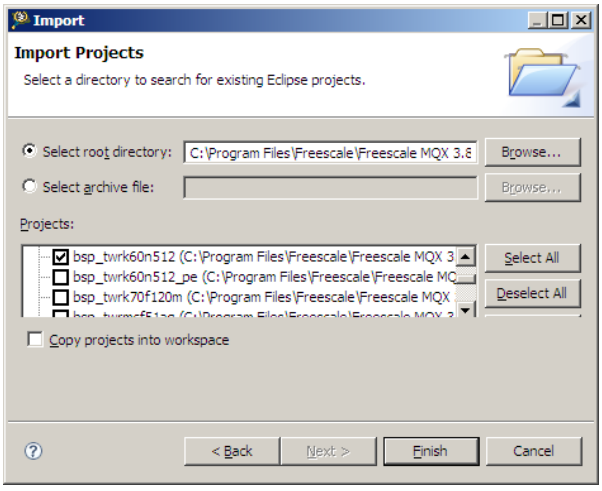
- Import the Existing BSP Project into the Workspace.





- Navigate to the *<mqx_installation_folder>\mqx\build* folder.

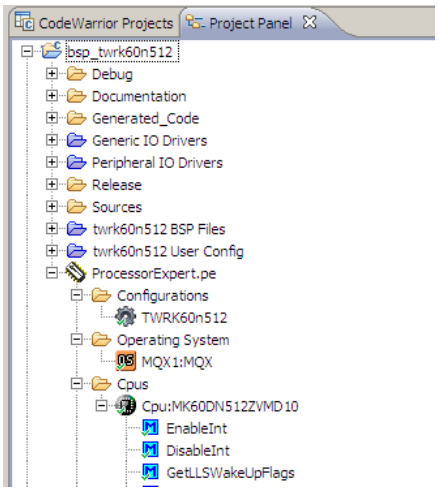




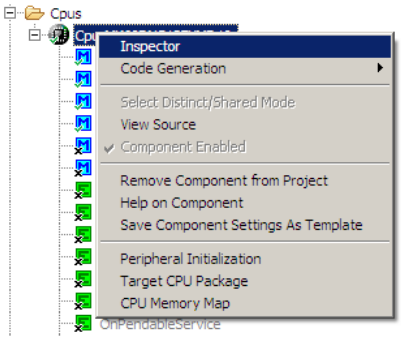
- Select the BSP project to import and click Finish.

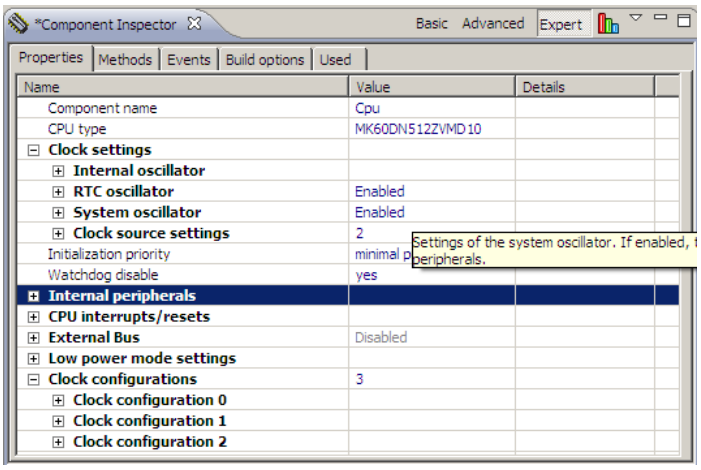
- Open the Processor Expert Project Panel if not already opened. Select *Processor Expert* / *Show Views* menu.

- Expand the CPU component item under the ProcessorExpert.pe project file item as follows:

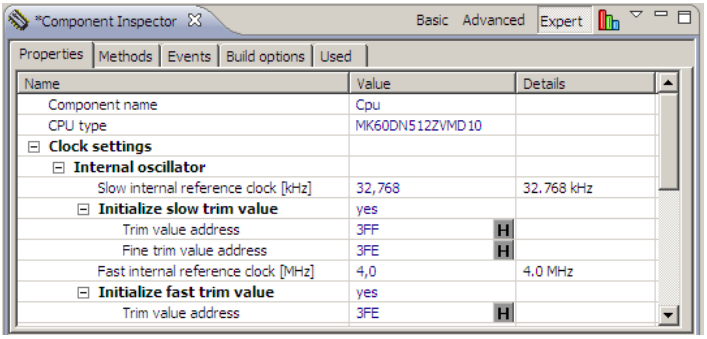


- Right click the CPU and select “Inspector”. The CPU Component Inspector opens.

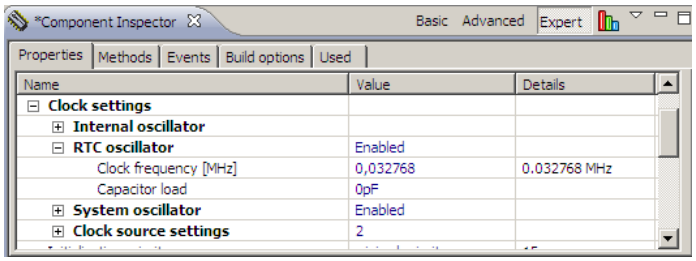




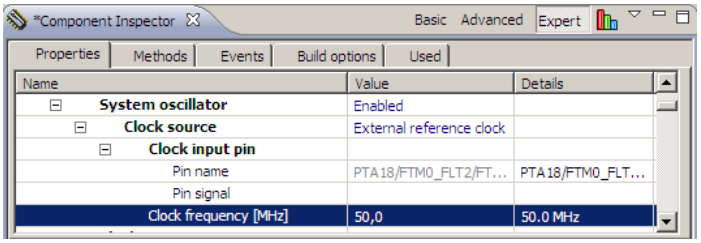
- Unfold the “Internal oscillator” item and set the required parameters.



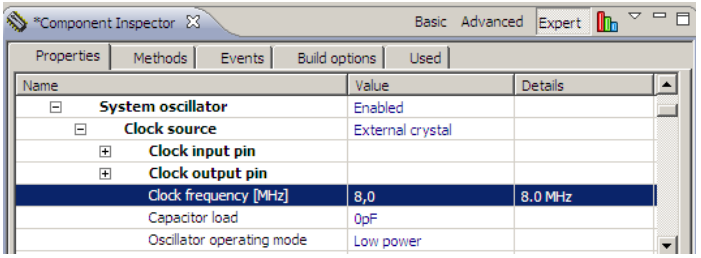
- Unfold “RTC oscillator” and set
  - Clock frequency [MHz]
  - Capacitor load [pF]



- Unfold “System oscillator” and set
  - For TWR-K60N512/K53N512
    - Clock source [External reference clock]
    - Clock frequency [MHz] = 50MHz



  - For TWR-K40X256 and KwikStik
    - Clock source[External crystal]
    - Clock frequency [MHz]
      - TWR-K40X256 = 4MHz
      - KwikStik = 8MHz
    - Capacitor load [pF]
    - Oscillator operating mode [High gain / Low power]



- Define clock source settings as follows:

  Kinetis BSPs use 2 clock source settings

  - PLL Engaged External (PEE)

    In PEE mode, the MCG output clock is derived from the PLL output clock, which is controlled by the external reference clock. The FLL is disabled in a low-power state. This setting is used in normal run mode as a clock source for BSP_CLOCK_CONFIGURATION_0 and BSP_CLOCK_CONFIGURATION_1.

  - Bypassed Low Power Internal (BLPI)

    In BLPI mode, MCG output clock is derived either from the slow internal reference clock or fast internal reference clock. Both the FLL and PLL are disabled. This setting is used in low power run mode as a clock source for BSP_CLOCK_CONFIGURATION_2.

Clock source settings 0

Name	Value	Details
Clock source setting 0		
Internal reference clock		
MCGIRCLK clock	Disabled	
MCGIRCLK in stop	Disabled	
MCGIRCLK source	Fast	
MCGIRCLK clock [MHz]	0,0	0.0 MHz; IRCLK is.
External reference clock		
OSCERCLK clock	Disabled	
OSCERCLK in stop	Disabled	
OSCERCLK clock [MHz]	0,0	0.0 MHz; System .
ERCLK32K clock source	RTC oscillator	
ERCLK32K. clock [kHz]	0,032768	0.032768 MHz
MCG settings		
MCG mode	PEE	
MCG output clock	PLL clock	
MCG output [MHz]	96,0	96.0 MHz
MCG external ref. clock source	System oscillator	
MCG external ref. clock [MHz]	50,0	50.0 MHz
Clock monitor	Enabled	
FLL settings		
PLL settings		
PLL module	Enabled	
PLL module in Stop	Enabled	
PLL output [MHz]	96,0	96.0 MHz
Reference clock divider	Auto select	25
PLL reference clock [MHz]	2,0	2.0 MHz
Multiplication factor	Auto select	48
Loss of lock interrupt	Disabled	

Clock source settings 1

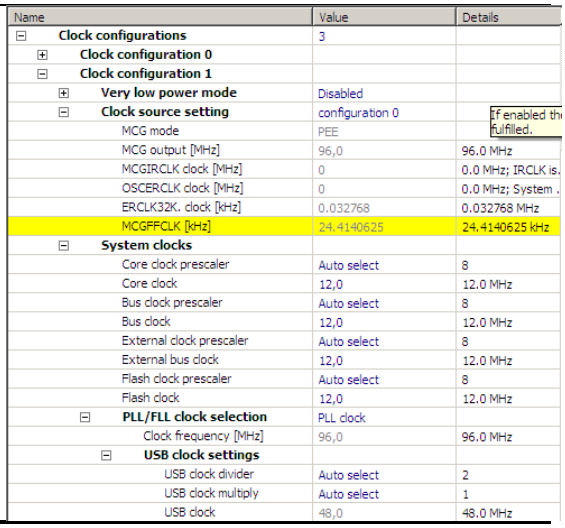
Name	Value	Details
Clock source setting 1		
Internal reference clock		
MCGIRCLK clock	Enabled	
MCGIRCLK in stop	Enabled	
MCGIRCLK source	Fast	
MCGIRCLK clock [MHz]	2,0	2.0 MHz
External reference clock		
OSCERCLK clock	Disabled	
OSCERCLK in stop	Disabled	
OSCERCLK clock [MHz]	0,0	0.0 MHz; System .
ERCLK32K clock source	RTC oscillator	
ERCLK32K. clock [kHz]	0,032768	0.032768 MHz
MCG settings		
MCG mode	BLPI	
MCG output clock	Internal clock	
MCG output [MHz]	2,0	2.0 MHz
MCG external ref. clock source	System oscillator	
MCG external ref. clock [MHz]	50,0	50.0 MHz
Clock monitor	Disabled	
FLL settings		
PLL settings		
PLL module	Disabled	
PLL module in Stop	Disabled	
PLL output [MHz]	0,0	0.0 MHz; PLL is di.
Reference clock divider	Auto select	
PLL reference clock [MHz]	2,0	2.0 MHz
Multiplication factor	Auto select	
Loss of lock interrupt	Disabled	

- Define Clock configurations as follows:

Clock configuration 0 derived from “Clock Source setting 0”	
BSP_CLOCK_CONFIGURATION_0	
Core clock [MHz]	96
Bus clock [MHz]	48
External bus clock [MHz]	48
Flash clock [MHz]	24
PLL/FLL clock selection	PLL clock
Clock frequency [MHz]	96
USB clock	48

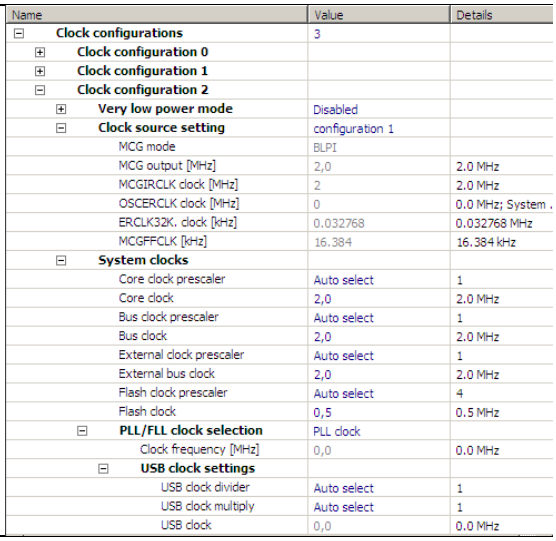
Name	Value	Details
Clock configurations	3	
Clock configuration 0		
Very low power mode	Disabled	
VLP mode entry	User	
VLP exit on interrupt	Disabled	
Clock source setting	configuration 0	
MCG mode	PEE	
MCG output [MHz]	96,0	96.0 MHz
MCGIRCLK clock [MHz]	0	0.0 MHz; IRCLK is.
OSCERCLK clock [MHz]	0	0.0 MHz; System .
ERCLK32K. clock [kHz]	0.032768	0.032768 MHz
MCGFFCLK [kHz]	24.4140625	24.4140625 kHz
System clocks		
Core clock prescaler	Auto select	1
Core clock	96,0	96.0 MHz
Bus clock prescaler	Auto select	2
Bus clock	48,0	48.0 MHz
External clock prescaler	Auto select	2
External bus clock	48,0	48.0 MHz
Flash clock prescaler	Auto select	4
Flash clock	24,0	24.0 MHz
PLL/FLL clock selection	PLL clock	
Clock frequency [MHz]	96,0	96.0 MHz
USB clock settings		
USB clock divider	Auto select	2
USB clock multiply	Auto select	1
USB clock	48,0	48.0 MHz

Clock configuration 1 derived from “Clock source setting 0”	
BSP_CLOCK_CONFIGURATION_1	
Core clock [MHz]	12
Bus clock [MHz]	12
External bus clock [MHz]	12
Flash clock [MHz]	12
PLL/FLL clock selection	PLL clock
Clock frequency [MHz]	96
USB clock	48



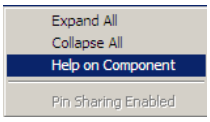
Name	Value	Details
Clock configurations	3	
Clock configuration 0		
Clock configuration 1		
Very low power mode	Disabled	
Clock source setting	configuration 0	
MCG mode	PEE	
MCG output [MHz]	96,0	96.0 MHz
MCGIRCLK clock [MHz]	0	0.0 MHz; IRCLK is.
OSCERCLK clock [MHz]	0	0.0 MHz; System .
ERCLK32K. clock [kHz]	0.032768	0.032768 MHz
MCGFFCLK [kHz]	24.4140625	24.4140625 kHz
System clocks		
Core clock prescaler	Auto select	8
Core clock	12,0	12.0 MHz
Bus clock prescaler	Auto select	8
Bus clock	12,0	12.0 MHz
External clock prescaler	Auto select	8
External bus clock	12,0	12.0 MHz
Flash clock prescaler	Auto select	8
Flash clock	12,0	12.0 MHz
PLL/FLL clock selection	PLL clock	
Clock frequency [MHz]	96,0	96.0 MHz
USB clock settings		
USB clock divider	Auto select	2
USB clock multiply	Auto select	1
USB clock	48,0	48.0 MHz

Clock configuration 2 derived from “Clock source setting 1”	
BSP_CLOCK_CONFIGURATION_2	
Core clock [MHz]	2
Bus clock [MHz]	2
External bus clock [MHz]	2
Flash clock [MHz]	0.5
PLL/FLL clock selection	PLL clock
Clock frequency [MHz]	0
USB clock	0



Name	Value	Details
Clock configurations	3	
Clock configuration 0		
Clock configuration 1		
Clock configuration 2		
Very low power mode	Disabled	
Clock source setting	configuration 1	
MCG mode	BLPI	
MCG output [MHz]	2,0	2.0 MHz
MCGIRCLK clock [MHz]	2	2.0 MHz
OSCERCLK clock [MHz]	0	0.0 MHz; System .
ERCLK32K. clock [kHz]	0.032768	0.032768 MHz
MCGFFCLK [kHz]	16.384	16.384 kHz
System clocks		
Core clock prescaler	Auto select	1
Core clock	2,0	2.0 MHz
Bus clock prescaler	Auto select	1
Bus clock	2,0	2.0 MHz
External clock prescaler	Auto select	1
External bus clock	2,0	2.0 MHz
Flash clock prescaler	Auto select	4
Flash clock	0,5	0.5 MHz
PLL/FLL clock selection	PLL clock	
Clock frequency [MHz]	0,0	0.0 MHz
USB clock settings		
USB clock divider	Auto select	1
USB clock multiply	Auto select	1
USB clock	0,0	0.0 MHz

For a detailed description of all CPU component parameters, see the ProcessorExpert Help. Right click the component and select “Help on Component.”



- When all clock settings is finished and no error is displayed in the Component Inspector, select menu *“Project->Generate Processor Expert Code“* to generate code.
- The generated files appear in Generated_Code directory. The code overloads the default setting in *bsp_cm.c.* See how the PE_LDD_VERSION macro is used for conditional compilation of either the default code or the newly generated code.
    - The *Cpu.c* file contains the SetClockConfiguration, GetClockConfiguration, MCGAutotrim methods and set of dependent static functions.
    - The *Cpu.h* contains defined symbols of CPU frequencies.
    - The *PE_LDD.c* file contains newly generated array of clock frequencies.

- Change the BSP_CLOCK_CONFIGURATION enumeration in the *bsp_cm.h* file to match your new settings.
- Rebuild the BSP with new clock settings.
- Optionally, you may want to move the generated code back to the *bsp_cm.c* and *.h* files and make it a new default clock setting.

Note: When not using the CodeWarrior 10.x as the MQX development environment, you can download standalone ”Processor Expert Driver Suite” and generate source code as described above. The only difference is that files generated in the standalone tool need to be moved to the BSP folder manually or the code has to be merged into the *bsp_cm.c* and *.h* files directly.