

# **Freescale MQX™ RTOS 4.0.2**

## **Release Notes**

<b>PRODUCT:</b>	Freescale MQX™ RTOS
<b>PRODUCT VERSION:</b>	4.0.2
<b>DESCRIPTION:</b>	Freescale MQX™ RTOS, version 4.0.2
<b>RELEASE DATE:</b>	August, 2013



**How to Reach Us:**

**Home Page:**

[www.freescale.com](http://www.freescale.com)

**Web Support:**

<http://www.freescale.com/support>

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [freescale.com/SalesTermsandConditions](http://freescale.com/SalesTermsandConditions).

Freescale, the Freescale logo, Kinetis, CodeWarrior, ColdFire, and Processor Expert are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Vybrid is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM, ARM Cortex-M4, and ARM Cortex-A5 are the registered trademarks of ARM Limited. Keil Software and Design®, the Keil Software Logo, µVision®, RealView®, C51™, C166™, MDK™, RL-ARM™, ULINK®, Device Database®, and ARTX™ are trademarks or registered trademarks of Keil, Tools by ARM, and ARM Ltd. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2008-2013 Freescale Semiconductor, Inc.



## Table of Contents

<b>Freescale MQX™ RTOS 4.0.2 Release Notes</b> .....	<b>i</b>
<b>1 Read Me First</b> .....	<b>3</b>
1.1 Requirements.....	3
1.1.1 Development Tools .....	3
1.1.2 System Requirements.....	4
1.1.3 Target Requirements .....	4
1.2 Special instructions .....	5
1.2.1 Setup Installation instructions .....	5
<b>2 What is New?</b> .....	<b>6</b>
<b>3 Release Content</b> .....	<b>7</b>
<b>4 MQX Release Overview</b> .....	<b>10</b>
4.1 MQX RTOS PSP .....	11
4.2 MQX RTOS BSPs .....	11
4.2.1 I/O Drivers Supported .....	11
4.2.2 Default IO channel .....	14
4.2.3 MQX PSP and BSP Directory Structure .....	15
4.3 MQX MFS .....	16
4.4 MQX RTCS .....	16
4.5 MQX USB Host .....	16
4.6 MQX USB Device.....	17
4.7 MQX Shell.....	17
4.8 Changing the MQX Source Files .....	17
4.9 Building the MQX Libraries.....	17
4.10 Example Applications .....	18

<b>5</b>	<b>Known Issues and Limitations.....</b>	<b>21</b>
5.1	Performance of Code Running in MRAM.....	21
5.2	Default Kernel Configuration of Small-RAM Devices .....	21
5.3	USB Host HUB Examples .....	21
5.4	OSBDM / OSJTAG Firmware Compatibility .....	21
5.5	Supporting “Hot Device Uninstall” in MQX I/O Subsystem .....	22
5.6	TWR-MEM Compact Flash interface issues .....	22
5.7	Idle task is required on Kinetis platforms .....	22
5.8	USB EHCI stack cannot handle cached memory.....	22
5.9	Flash Cache Disabled on TWR-K60N512 BSP – erratum e2647.....	23
5.10	TWR-K60N512 BSP 256KB Flash boundary issue.....	23
5.11	ARP entries issue.....	23
5.12	FlexCan driver issues.....	23
5.13	ARM MDK Keil $\mu$ Vision Support – Issue linking the TWR-K40X256 RTCS library .....	23
5.14	Low power modes and PE generated code related issues .....	23
5.15	Keil UV4 Mixing the C99 and C++ code in one project .....	24
5.16	Compilation warnings during CW10.3 GCC compilation .....	24
5.17	LFN to SFN translation when creating a file not handled properly .....	24
5.18	The Timeline View does not show the graph for PPC platforms .....	24
5.19	Android USB MSD cannot be interfaced.....	24
5.20	HMI and User Mode functionality in CW10 GCC .....	24
5.21	MFS does not check validity of directory rename.....	24
5.22	CodeWarrior 10.3 & 10.4 GCC compiler requires latest version of New Project Wizard .....	24
5.23	Special Considerations when using the Cache with Low Power Modes on K60F and K70F platforms .....	26
6	Change Log.....	26

# 1 Read Me First

This is the release notes for Freescale MQX™ RTOS version 4.0.2 product released for Freescale Kinetis, Vybrid, ColdFire, and Power Architecture® processor families.

## 1.1 Requirements

### 1.1.1 Development Tools

Freescale MQX RTOS version 4.0.2 release was compiled and tested with the following development tools:

- CodeWarrior Development Studio for Microcontrollers version 10.4
  - Support available for Kinetis, ColdFire and PX series of Power Architecture devices
  - See build projects in the `cw10` and `cw10gcc` subdirectories
  - Makefiles build option (Kinetis GCC only): `TOOL=cw10gcc`
- IAR Embedded Workbench® for ARM® Version 6.50.5
  - Support available for Kinetis and Vybrid devices
  - See build projects in `iar` subdirectories
- ARM Development Studio 5 (DS-5™) version 5.15.0
- DS-5 Starter Kit for Vybrid Controllers version 5.15.0
- DS-5 Vybrid Controller Edition 5.15.0
  - Support available for Vybrid devices only. It is very important to use DS-5 version 5.15.0 or later for Vybrid.
  - See build projects in the `ds5` subdirectories
- ARM-MDK™ - Keil μVision® version 4.71
  - Support available for Kinetis devices
  - See build projects in the `uv4` subdirectories
- Sourcery CodeBench Lite Edition version 2012.03-56
  - Support available for Vybrid devices
  - Makefile build option: `TOOL= gcc_cs`
- Makefile support (mingw32-make version 3.8.2)
  - Library makefiles are located in the `<MQX_root_dir>/build/make/<board>`
  - Application makefiles are located in `<example_dir>/make/<board>`
- Support of the following tools is discontinued since MQX 4.0.0
  - CodeWarrior Development Studio for mobileGT version 9.2
  - CodeWarrior Development Studio for Microcontrollers version 6.3.1
  - CodeWarrior Development Studio for ColdFire Architectures version 7.2.2

### 1.1.2 System Requirements

System requirements are based on the requirements for the development tools. There are no special host system requirements for hosting the Freescale MQX RTOS distribution itself.

Minimum PC configuration:

As required by Development and Build Tools

Recommended PC configuration:

2 GHz processor – 2 GB RAM - 2 GB free disk space

Software requirements:

OS: Windows XP or later

### 1.1.3 Target Requirements

Freescale MQX RTOS 4.0.2 supports the evaluation boards mentioned below. There are no special requirements for the target hardware other than what each board requires for its operation (power supply, cabling, jumper settings etc). For more details about the board-specific setup for MQX applications, see *Getting Started with Freescale MQX™ RTOS*.

#### Evaluation boards supported:

- Vybrid
  - TWR-VF65GS10 Development Kit
  - AutoEVB Vybrid Evaluation Board
- Kinetis
  - TWR-K20D50M Development Kit
  - TWR-K20D72M Development Kit
  - TWR-K21D50M Development Kit
  - TWR-K40X256 Development Kit
  - TWR-K40D100M Development Kit
  - TWR-K53N512 Development Kit
  - TWR-K60N512 Development Kit
  - TWR-K60D100M Development Kit
  - TWR-K60F120M Development Kit
  - TWR-K70F120M Development Kit
  - KwikStik - K40X256 based Evaluation Board
- ColdFire V1
  - TWR-MCF51JF Development Kit

- ColdFire V2
  - M52259EVB Evaluation Board
  - M52259DEMOKIT Development Kit
  - TWR-MCF52259 Development Kit
- ColdFire V4
  - TWR-MCF54418 Development Kit
- PowerPC:
  - TWR-PXD10 Development Kit
  - TWR-PXN20 Development Kit
  - TWR-PXS20 Development Kit
  - TWR-PXS30 Development Kit

## 1.2 Special instructions

### 1.2.1 Setup Installation instructions

Run the self-extracting MQX installer application and proceed according to instructions on the screen.

If you already have MQX version 4.0.0 installed, all existing files are overwritten and updated in the original installation path.

If you are not updating from MQX version 4.0.0, the installation program offers the following folder as a default installation location: `C:\Freescale\Freescale_MQX_4_0`. Install MQX to a path without spaces to avoid build problems with certain build tools.

The installation package contains full MQX source code, IDE build projects, and makefiles for all supported boards.

**Note:** Since version 4.0, the pre-built libraries are not distributed in the MQX release package which makes it necessary to compile all MQX libraries for a particular board before the first use. For detailed build instructions, see *Building the MQX Libraries* section in the *Getting Started with Freescale MQX™ RTOS*.

### 1.3 Technical Support

For a description of available support options and restrictions, please visit: [freescale.com/mqx/support](http://freescale.com/mqx/support).

## 2 What is New?

This section describes the major changes and new features implemented in this release.

- AutoEVB Vybrid A5 and M4 Board support packages
  - Support for ARM® Cortex®-A5 and ARM® Cortex®-M4 cores of the dual core Vybrid processor
  - Supporting standard set of I/O drivers
- Vybrid BSP extensions
  - RTC and the NAND Flash driver are ported to the Vybrid platform
  - MMU support extended to handle 4KB memory blocks
  - Introduced new QuadSPI driver
  - Cortex-M4 boot option enabled.
  - Direct code execution from QuadSPI flash (XIP) provided. This feature is demonstrated in Vybrid QuadSPI boot loader.
- Introduced the first version of eDMA driver – the driver is experimental and will be further extended in the next MQX version. Documentation is not provided.
- Kinetis SPI driver was updated to use the eDMA driver.
- Hardware timer driver support extended to SysTick and GPT HW modules. Kinetis and Vybrid BSPs updated to leverage the Hardware timer to provide MQX tick time.
- RTCS Hardware checksum acceleration enabled in ENET driver (for TCP and UDP); the benefits are the increased throughput and a reduced processor loading. This option is enabled by default for the K60N512 platform.
- HTTP server redesigned to provide a faster and more robust solution. The server API is simplified and changed to correspond to the RTCS standard.
- New SMTP client functionality provided as a part of the RTCS network suite. The client offers a simple API for e-mail handling. The new RTCS Shell command “email” demonstrates its functionality.
- The security\_email, security\_webserver, and security\_telnet demo applications are removed from this release. Networking functionality is demonstrated in RTCS shell and httpsrv example applications.
- Multicore Communication (MCC) library updated to version 1.1. Fixed the incorrect usage of cache macros in the `mcc_send()` function.



### 3 Release Content

This release contains the following:

Deliverable	Location	Status
<b>Configuration Files and Mass-Build Projects</b>	<install_dir>/config/...	
Configuration and mass-build project for all supported boards	.../config/<board>	Updated
<b>MQX PSP, BSP Source Code, and Examples</b>	<install_dir>/mqx/...	
MQX PSP source code for Kinetis/Vybrid Cortex-M core	.../mqx/source/psp/cortex_m	Updated
MQX PSP source code for Vybrid Cortex-A core	.../mqx/source/psp/cortex_a	new
MQX PSP source code for ColdFire	.../mqx/source/psp/coldfire	Updated
MQX PSP source code for PowerPC	.../mqx/source/psp/powerpc	Updated
MQX PSP build projects	.../mqx/build/<compiler>/psp_<board>	Updated
MQX BSP Source Code	.../mqx/source/bsp/<board>	Updated
MQX BSP build projects	.../mqx/build/<compiler>/bsp_<board>	Updated
MQX example applications	.../mqx/examples/...	Updated
<b>RTCS Source Code and Examples</b>	<install_dir>/rtcs/...	
RTCS source code	.../rtcs/source	Updated
RTCS build projects	.../rtcs/build/<compiler>/rtcs_*	Updated
RTCS example applications	.../rtcs/examples	Updated
<b>MFS Source Code and Examples</b>	<install_dir>/mfs/...	
MFS source code	.../mfs/source	Updated
MFS build projects	.../mfs/build/<compiler>/mfs_*	Updated
MFS example applications	.../mfs/examples	Updated
<b>USB Host Drivers Source Code and Examples</b>	<install_dir>/usb/host/...	
USB Host source code and class drivers	.../usb/host/source	Updated
HUB Class Driver	.../usb/host/source/classes/hub	Updated
Human Interface Device (HID) Class Driver	.../usb/host/source/classes/hid	Updated
Mass Storage (MSD) Class Driver	.../usb/host/source/classes/msd	Updated
Printer Class Driver	.../usb/host/source/classes/printer	Updated
CDC Class Driver	.../usb/host/source/classes/cdc	Updated
USB Host build projects	.../usb/host/build/<compiler>/usb_hdk_*	Updated
USB Host example applications (HID, MSD, HUB)	.../usb/host/examples	Updated
<b>USB Device Drivers Source Code and Examples</b>	<install_dir>/usb/device/...	
USB Device source code	.../usb/device/source	Updated
USB Device build projects	.../usb/device/build/<compiler>/usb_ddk_<board>	Updated
USB Device example applications (HID, MSD, CDC, PHDC)	.../usb/device/examples	Updated
<b>Shell Library Source Code</b>	<install_dir>/shell/...	
Shell source code	.../shell/source	Updated
Shell build projects	.../shell/build/<compiler>/shell_<board>	Updated
<b>Build tools plug-ins</b>	<CodeWarrior_dir>/...	
CW10.2 TAD and New Project wizard plugins for manual installation (CW10.3 and later plugins are installed directly from the IDE)	.../tools/codewarrior_extensions/...	Updated
Keil Task Aware Debugging plugin (TAD)	.../tools/keil_extensions/	Updated
IAR Task Aware Debugging plugin (TAD)	.../tools/iar_extensions/	Updated
<b>PC Host Tools</b>	<install_dir>/tools	
BSP Cloning wizard	.../tools/BSPCloningWizard/BSPCloningWizard.exe	New
TFS Make Utility	.../tools/mktrfs.exe	from 3.0

Check for Latest Version tool	.../tools/webchk.exe	from 3.0
AWK interpreter (GNU General Public License)	.../tools/gawk.exe	from 3.1
SNMP code generation scripts	.../tools/snmp/*.awk	from 3.1
Timing HTML report tool (for mqx/examples/benhmrk/timing)	.../tools/timing.exe	Updated
Code size HTML report tool (for mqx/examples/benhmrk/codesize)	.../tools/codesize.exe	Updated
TAD string and configuration files	.../tools/tad	Updated
Demo Applications	<install_dir>/demo	
Various demo applications demonstrating complex MQX functionalities.	.../demo/...	Updated
Documentation	<install_dir>/doc	
User Guides and Reference Manuals for MQX RTOS, RTCS, MFS, I/O Drivers, USB etc.	.../doc	Updated

The following image shows the Freescale MQX RTOS directories installed to the user host computer (subdirectories reduced for clarity):

<ul style="list-style-type: none"> <li>[-] [Freescale_MQX_4_0] <ul style="list-style-type: none"> <li>[-] [config] <ul style="list-style-type: none"> <li>[common]</li> <li>[-] [twrk60n512] <ul style="list-style-type: none"> <li>[cw10]</li> <li>[iar]</li> <li>[uv4]</li> </ul> </li> </ul> </li> <li>[+] [demo]</li> <li>[+] [doc]</li> <li>[-] [lib] <ul style="list-style-type: none"> <li>[-] [twrk60n512.cw10] <ul style="list-style-type: none"> <li>[+] [debug]</li> <li>[+] [release]</li> </ul> </li> <li>[+] [twrk60n512.iar]</li> <li>[+] [twrk60n512.uv4]</li> </ul> </li> <li>[-] [mfs] <ul style="list-style-type: none"> <li>[+] [build]</li> <li>[+] [examples]</li> <li>[+] [source]</li> </ul> </li> <li>[-] [mqx] <ul style="list-style-type: none"> <li>[+] [build]</li> <li>[+] [examples]</li> <li>[+] [source]</li> </ul> </li> <li>[-] [rtcs] <ul style="list-style-type: none"> <li>[+] [build]</li> <li>[+] [examples]</li> <li>[+] [source]</li> </ul> </li> <li>[-] [shell] <ul style="list-style-type: none"> <li>[+] [build]</li> <li>[+] [source]</li> </ul> </li> <li>[+] [tools]</li> <li>[-] [usb] <ul style="list-style-type: none"> <li>[+] [common]</li> <li>[-] [device] <ul style="list-style-type: none"> <li>[+] [build]</li> <li>[+] [examples]</li> <li>[+] [source]</li> </ul> </li> <li>[-] [host] <ul style="list-style-type: none"> <li>[+] [build]</li> <li>[+] [examples]</li> <li>[+] [source]</li> </ul> </li> </ul> </li> </ul> </li> </ul>	<p>MQX configuration header files for different boards (may reuse common files in directory "common")</p> <p>Documentation</p> <p>MQX libraries compiled in CodeWarrior for particular board (this is where MQX applications get the MQX from)</p> <p>MFS File System (sources, build projects and examples)</p> <p>MQX Operating System (sources, build projects and examples)</p> <p>RTCS TCP/IP Stack (sources, build projects and examples)</p> <p>Shell Support Library</p> <p>PC Host Tools</p> <p>USB Device Drivers (sources, build projects and examples)</p> <p>USB Host Drivers (sources, build projects and examples)</p>
---	---

## 4 MQX Release Overview

	MQX PSP+BSP Libraries	MFS Library (FAT File System)	RTCS Library (TCP/IP Stack)	Shell Library	USB Host Library	USB Device Library	UART (polled and interrupt driven)	I2C (polled and interrupt driven)	SPI	LWGPIO	HW Timer (PIT, SysTick, GPT)	ADC, LWADC	Audio driver I2S or SAI	QuadSPI	FLASH	NAND flash driver	ESDHC	Compact Flash Card driver	SD Card driver (SPI or SDHC based)	RTC, IRTC (Real Time Clock)	TSS - Touch Sensing	DIU	FlexCAN / mCAN	Ethernet driver	USB Host/Device driver
<b>Vybrid</b>																									
TWRVF65GS10_A5	•	•	•	•	•	•	•	•	•	•	•	2)		•		•	•		•	•				•	•
TWRVF65GS10_M4	•	•	•	•	•	•	•	•	•	•	•	2)		•		•	•		•	•				•	•
AUTOEVB_VYBRID_A5	•	•	•	•	•	•	•	•	•	•	•	2)				•	•		•	•				•	•
AUTOEVB_VYBRID_M4	•	•	•	•	•	•	•	•	•	•	•	2)				•	•		•	•				•	•
<b>Kinetis</b>																									
TWRK20D50M	•	•	1)	•	•	•	•	•	•	•	•				•		•		•	•			•	•	
TWRK20D72M	•	•	1)	•	•	•	•	•	•	•	•				•		•		•	•			•	•	
TWRK40X256	•	•	1)	•	•	•	•	•	•	•	•	•				•		•		•	•		•	•	
TWRK40D100M	•	•	1)	•	•	•	•	•	•	•	•						•		•	•			•	•	
TWRK53N512	•	•	•	•	•	•	•	•	•	•	•						•		•	•			•	•	
TWRK60D100M	•	•	•	•	•	•	•	•	•	•	•						•		•	•			•	•	
TWRK60F120M	•	•	•	•	•	•	•	•	•	•	•						•		•	•			•	•	
TWRK60N512	•	•	•	•	•	•	•	•	•	•	•						•		•	•			•	•	
KwikStik (K40)	•	•	1)	•	•	•	•	•	•	•	•						•		•	•			•	•	
TWRK70F120M	•	•	•	•	•	•	•	•	•	•	•						•		•	•			•	•	
<b>ColdFire V1</b>																									
TWRMCF51JF	•	•	1)	•	•	•	•	•	•	•	•						•		•	•				•	
<b>ColdFire V2-V4</b>																									
M52259DEMO	•	•	•	•	•	•	•	•	•	•	•								•	•			•	•	
M52259EVB	•	•	•	•	•	•	•	•	•	•	•								•	•			•	•	
TWRMCF52259	•	•	•	•	•	•	•	•	•	•	•								•	•			•	•	
TWRMCF54418	•	•	•	•	•	•	•	•	•	•	•					•	•		•	•			•	•	
<b>PowerPC</b>																									
TWR-PXD10	•	•	1)	•			•	•	•	•	•								•	•			•		
TWR-PXS20	•	•	1)	•			•		•	•	•								•	•			•		
TWR-PXS30	•	•	•	•			•	•	•	•	•								•	•			•	•	
TWR-PXN20	•	•	•	•			•	•	•	•	•								•	•			•	•	

- New in this release

- 1) Onchip Ethernet not available, RTCS can be used with PPP or custom ENET driver (such as Wi-Fi over SPI)
- 2) Provided LWADC driver only

MQX 4.0.2 release is intended for Freescale ColdFire microcontrollers, Kinetis ARM Cortex microcontrollers, and Freescale PowerPC.

The Freescale MQX RTOS release includes MQX real time kernel, core system components, integrated TCP/IP network stack - RTCS, file system – MFS, and USB Host and Device stacks.

## 4.1 MQX RTOS PSP

Freescale MQX RTOS 4.0.2 release contains ARM Cortex-M, ARM Cortex-A, ColdFire, and PowerPC Platform Support Packages. Contact the Embedded Access Inc. ([www.embedded-access.com](http://www.embedded-access.com)) support for other Freescale platforms.

The platform-specific code from `/mqx/source/psp/<platform>` is built together with the generic MQX core files. These two parts form a static library generally referred to as “PSP” which enables the target application to access RTOS features.

## 4.2 MQX RTOS BSPs

Freescale MQX RTOS 4.0.2 release includes Board Support Packages for boards mentioned in the first section.

The board-specific code from `/mqx/source/bsp/<board>` is built together with I/O driver files from `/mqx/source/bsp/io`. These two parts form a static library generally referred as “BSP”. The functions included in this library enable the board and operating system to boot up and use the I/O driver functions.

The following section describes drivers supported by the MQX BSPs.

### 4.2.1 I/O Drivers Supported

The following list describes I/O drivers available in the latest MQX release. The drivers are an optional part of the MQX RTOS and their installation can be enabled or disabled in the BSP startup code. To provide the optimal code and RAM application size, most of the drivers are disabled in the `/config/<board>/user_config.h` file by default. The drivers required by demonstration applications (in the `/demo` folder) are enabled by default.

**Note:** When `BSPCFG_` driver-enabling macros are missing in the `/config/<board>/user_config.h` file, the default setting is taken from the BSP-specific header file located in `/mqx/source/bsp/<board>/<board>.h`.

The user decides whether to enable the automatic installation of the driver in the BSP startup code (by enabling appropriate `BSPCFG_ENABLE_XXX` macro in `user_config.h`), or manually in the application code.

### TFS – Trivial Filesystem

Trivial Filesystem is used as a simple read-only file repository instead of the fully featured MFS. TFS is not installed in the BSP startup code. Application must initialize the TFS and pass a pointer to the filesystem data image. The `mkfts` tool is available (both as executable and Perl script) to generate the image from the existing directory structure. The RTCS HTTP example demonstrates the use of TFS.

### I2C I/O Driver

This driver supports a polled I2C interface in a master mode. If enabled in user configuration, the I2C device drivers are installed during the BSP startup code as “`i2c0:`” and “`i2c1:`”. An example application is provided in the MQX source tree.

### I2S &SAI I/O Driver

This driver supports I2S interface in a master mode. If enabled in user configuration, the I2S device driver is installed during the BSP startup code as “`i2s0:`”. An example application is provided in the MQX source tree.

## SPI I/O Driver

This driver supports the operation master mode. If enabled in user configuration, the SPI device drivers are installed during the BSP startup code as “**spi0:**” (or “**spiX:**” where X is index of the SPI module used). The SPI driver is significantly rewritten in the MQX 4.0, so that there is no distinct interrupt or polled driver type. See *I/O User Guide* for details. On Kinetis and Vybrid platforms, the driver uses DMA to function.

## QuadSPI I/O Driver

This driver provides a C language API to QuadSPI peripheral module. If enabled in user configuration, the QuadSPI device drivers are installed during BSP startup code as “**qspi0:**” (or “**qspiX:**” where X is index of QSPI module used). The QSPI driver is a new feature in MQX 4.0.2. See the *MQX I/O User Guide* for details.

## FlexCAN Driver

This driver provides a C language API to the FlexCAN peripheral module. An example application is provided in the MQX source tree.

## msCAN Driver

This driver provides a C language API to the msCAN peripheral module. An example application is provided in the MQX source tree.

## RTC Driver

This driver provides a C language API to the Real Time Clock peripheral module and functions, and synchronizes the clock time between RTC and MQX systems. If enabled in user configuration, the RTC module is initialized and MQX time is renewed automatically during BSP startup.

## Serial I/O Driver

The standard SCI (UART) driver supports both polled and interrupt-driven modes. If enabled in user configuration, the serial devices are installed as “**ttya:**”, “**ttyb:**” and “**ttyc:**” (polled mode) and “**ittya:**”, “**ittyb:**” and “**ittyc:**” (interrupt mode) automatically during BSP startup.

## GPIO I/O Driver (obsolete)

Support of this driver has been discontinued in Freescale MQX.

This I/O driver provides a uniform interface to all GPIO pins available on a particular device. If enabled in user configuration, the GPIO driver is installed as “**gpio:**” automatically by the BSP startup. This driver is replaced by the LWGPIO driver for all supported platforms.

## LWGPIO I/O Driver

This driver provides a C language API to all GPIO ports available on a particular device. It is significantly faster and has a smaller footprint than the GPIO driver.

## ADC Driver

This I/O driver provides a uniform interface to ADC channels. ADC uses internal PIT, or a different timer, for periodic sampling. If enabled in user configuration, the ADC driver is installed as “**adc:**” (or “**adc1:**”, “**adc2:**”) device automatically during the BSP startup.

## LWADC I/O Driver

This driver provides a C language API to ensure a uniform access to ADC peripheral basic features.

## DAC Driver (obsolete)

The full version of this driver is provided as part of the Processor Expert driver suite.

This driver provides C language API to DAC peripheral module. It is adopted from the Freescale Processor Expert toolbox. The DAC driver is installed and used directly from the application code.

## Flash I/O Driver

This I/O driver provides a standard interface to either internal or external Flash memory. If enabled in user configuration, the Flash driver (called FlashX) is installed as “**flashx:**” device automatically by the BSP startup code. Note that “**flash0**”, “**flash1**” etc. device names are used for FlashX device installed for external Flash memory.

For devices with internal Flash memory, the FlashX driver depends on several parameters passed in a form of global symbols from application or from Linker Command File. For more information, see driver installation code in the BSP and an example application provided in the MQX source tree.

## ENET Driver

The low-level Ethernet driver is used by the RTCS TCP/IP software stack. The driver is initialized directly by the application before RTCS is used for the first time. The RTCS Shell and HTTP examples demonstrate the use of this driver.

## PCCard I/O Driver

This I/O driver provides a low-level access to the PCCard functionality by using Flexbus and CPLD circuit. The CPLD code can be found in `<install_dir>/mqx/source/io/pccard/<card_name>`. If enabled in the user configuration, the PCCard device driver is installed as “**pccarda:**” automatically during the BSP startup.

## PCFlash I/O Driver

The Compact Flash Card I/O driver is installed on top of the PCCard low-level driver and enables standard disk drive operations. The MFS file system can be installed on top of this device. If enabled in user configuration, the PCFlash device driver is installed as “**pcflasha:**” automatically during the BSP startup.

## SD Card I/O Driver

This I/O driver implements a subset of the SD protocol v2.0 (SDHC). The driver can use either the MQX SPI driver or the MQX (e)SDHC driver to communicate with the SD Card device. Install the driver at the application level, and pass a lower-layer driver handle to it. The MFS file system can be installed on top of this device.

## (E)SDHC I/O Driver

This I/O driver covers the (e)SDHC peripheral module and provides low-level communication interface for various types of cards including SD, SDHC, SDIO, SDCOMBO, SDHCCOMBO, MMC, and CE-ATA.

## Resistive Touch-screen Driver

This I/O driver accesses the ADC and GPIO modules to detect touch events and acquire touch coordinates on a resistive touch-screen unit.

## DIU Display driver

This driver provides a generic C language API to frame buffer-based display units. It is initialized and used from a user-application. Since the support for MPC5125 BSP was removed in the MQX 4.0, this driver is not currently used by any BSP.

## **I/ODebug driver**

This driver redirects I/O functions, such as printf, to a debug probe-based communication channel. The CodeWarrior 10, IAR EWARM 6, or Keil  $\mu$ Vision4<sup>®</sup> debugger consoles are supported. See *Getting Started with Freescale MQX™ RTOS* for details about the setup and use of this feature.

## **HWTimer driver (new, experimental)**

This driver provides a C language API for uniform access to the features of various HW timer modules such as PIT, SysTick, etc.

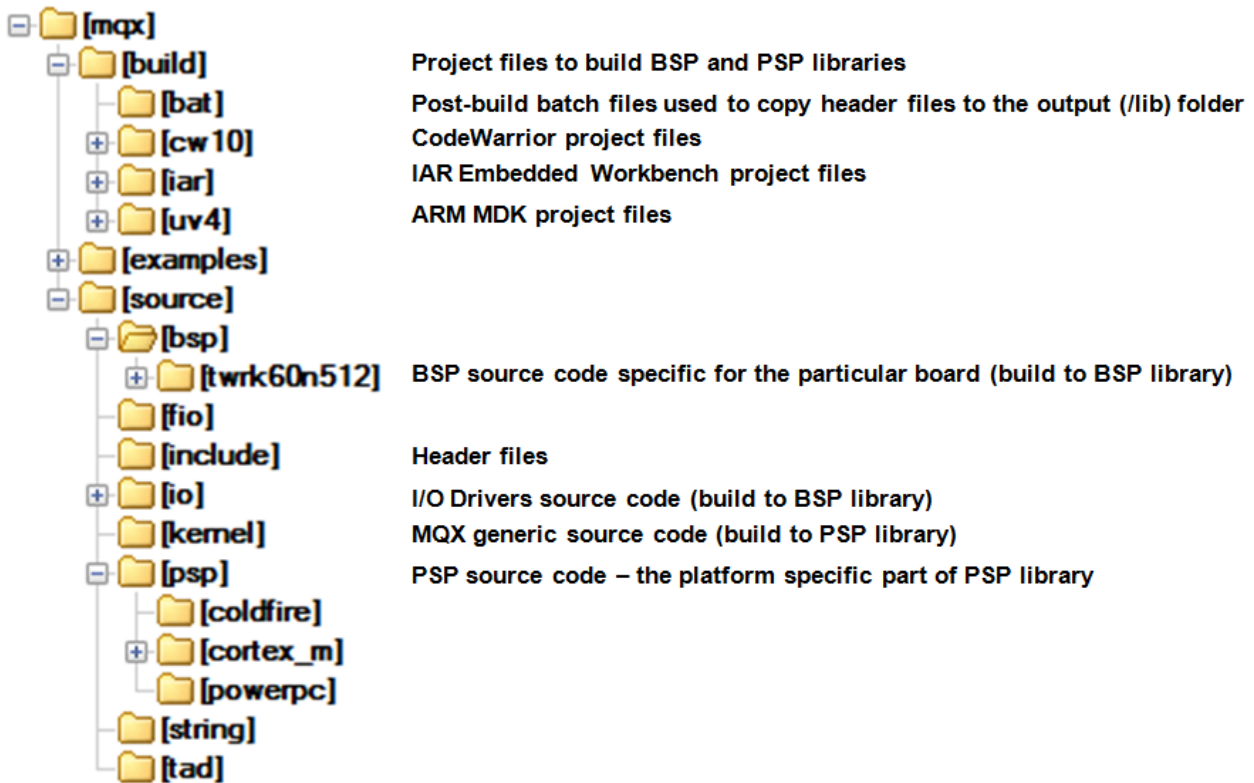
### **4.2.2 Default I/O channel**

An I/O communication device installed by MQX BSP can be used as the standard I/O channel. See *Getting Started with Freescale MQX™ RTOS* for the default console setting for each supported development board.



### 4.2.3 MQX PSP and BSP Directory Structure

RTOS files are located in the mqx subdirectory of the Freescale MQX RTOS installation. The directory structure is shown in the image below.



### 4.3 MQX MFS

MFS files from `/mfs/source` directory are built into a static library. When linked to the user application, the MFS library enables the application to access FAT12, FAT16, or FAT32-formatted drives.

### 4.4 MQX RTCS

RTCS files from `/rtcs/source` directory are built into a static library. When linked to the user application, the RTCS library enables the application to provide and consume network services of the TCP/IP protocol family.

MQX 4.0 RTCS stack is IPv6 ready with respect to legacy Silver Logo certification and has passed all required tests. IPv6 support is available as a separate update package available from Freescale.

### 4.5 MQX USB Host

Freescale MQX RTOS 4.0.2 release includes the USB Host drivers and USB class drivers.

The USB HDK (Host Development Kit) files from `/usb/host/source` directory are built into a static library. When linked to the user application, the USB HDK library enables the application to communicate with various USB devices attached on the USB bus.

The HDK contains the following USB class drivers:

- USB Hub class used to attach multiple devices to a single host port. If enabled at the application level, the HUB support is fully transparent. Only the user application needs to be modified to handle multiple USB devices simultaneously. A Keyboard/Mouse example application is provided.
- Human-interface Class (HID) used to access mouse, keyboard, and similar devices
- Mass storage device (MSD) Class used to access USB drives
- Communication Device Class (CDC) used as a serial communication device implementing virtual “tty” ports
- Audio Class
- PHDC for medical applications
- Basic Printer Class

## 4.6 MQX USB Device

Freescale MQX RTOS 4.0.2 release includes the USB Device drivers and example applications implementing various USB devices.

The USB DDK (Device Development Kit) files from `/usb/device/source` directory are built into the static library. When linked to the user application, the USB DDK library enables the application to act as a USB device supporting one or more of the following classes:

- HID (mouse functionality demonstrated)
- MSD (internal RAM area accessed as mass storage device)
- CDC COM (virtual serial line implementation)
- CDC NIC (virtual network interface card implementation)
- PHDC (medical applications)
- Audio

## 4.7 MQX Shell

The shell and command-line handling code is implemented as a separate library called Shell.

## 4.8 Changing the MQX Source Files

The Freescale MQX RTOS is distributed in the source code form. Do not modify any of the source files other than the compile-time configuration files. This recommendation applies to all files under “source” and “build” sub-directories in all MQX, RTCS, MFS, USB, and other core components folders.

If you are creating custom board support packages or adding additional I/O drivers, add the new files and subdirectories to the following directories:

```
<install_dir>/mqx/source/bsp
```

```
<install_dir>/mqx/source/io
```

## 4.9 Building the MQX Libraries

For more details about building MQX libraries and applications with CodeWarrior tools, see *Getting Started with Freescale MQX™ RTOS*.

When using MQX for the first time and making changes to the compile-time user configuration file or MQX kernel source files, rebuild MQX libraries to ensure that the changes are propagated to the user application.

## 4.10 Example Applications

Demo applications are in the following directory:

`<install_dir>/demo`

The examples are written to demonstrate the most frequently used features of the Freescale MQX RTOS.

In addition to these demo applications, there are simpler example applications available in MQX, RTCS, MFS, and USB directories.

The following tables summarize all demo and example applications provided in this release.

### MQX Example Applications

`mqx/examples/...`

Name	Description
<code>access_usr</code>	Demonstration of memory protection between the user and privilege tasks.
<code>adc</code>	Shows usage of the ADC driver sampling analog values from two ADC channels.
<code>can/flexcan</code>	Shows usage of FlexCAN API functions to transmit and receive CAN frames.
<code>can/mscan</code>	Shows usage of msCAN API functions to transmit and receive CAN frames.
<code>can/mscan_loopback</code>	The msCAN API driver demonstration in loopback mode.
<code>cplus</code>	Shows simple C++ application
<code>demo</code>	Shows MQX multitasking and inter-process communication using standard objects like semaphores, events, or messages. See <code>lwdemo</code> for the same example using the lightweight objects.
<code>event</code>	Simple demonstration of MQX events.
<code>flashx</code>	Demonstration of FlashX driver functionality.
<code>flexnvm</code>	Demonstrates use of the Flex NVM functionality on Kinetis platforms where this feature is supported.
<code>freq_change</code>	Show dynamic frequency change for Kinetis and ColdFire+ platforms only.
<code>gpio</code>	Shows usage of LWGPIO driver to control on-board LEDs and switches.
<code>hello</code>	A trivial Hello World application using a single task.
<code>hello2</code>	A trivial Hello World application spread across two tasks.
<code>hmi</code>	Demonstrates integration of the MQX application and the TouchSensing library.
<code>hwtimer</code>	Shows usage of HW timer driver abstraction. Demonstrates how to initialize HW timer for various modules, set frequency, callback, start, and stop the timer.
<code>i2c</code>	Shows how to read/write data from/to external EEPROM. Additional HW setup is needed.
<code>i2c_slave</code>	Shows usage of the I2C driver in slave mode – emulates the external EEPROM behavior.
<code>i2s_demo</code>	Demonstrates use of audio I2S driver. TWR-AUDIO card is needed to run this example.
<code>io</code>	Demonstrates use of an alternate UART port as a console output.
<code>ipc</code>	UART-based inter-processor communication demonstration.
<code>isr</code>	Shows how to install interrupt service routine and how to chain it with the previous handler.
<code>klog</code>	Shows kernel events being logged and later the log entries dumped on console.
<code>log</code>	Shows application-specific logging feature.
<code>lwdemo</code>	Same as the "demo" application, but implemented using lightweight components only.
<code>lwdemo_usr</code>	Shows MQX multitasking and inter-process communication using user mode tasks and lightweight objects such as semaphores, events, or messages.
<code>lwevent</code>	Simple demonstration of MQX lightweight events.
<code>lwlog</code>	Simple demonstration of MQX lightweight log feature.
<code>lwmsgq</code>	Simple demonstration of MQX lightweight inter-process messaging.
<code>lwsem</code>	Simple demonstration of MQX task synchronization using the lightweight semaphore object.
<code>lwsem_usr</code>	Simple demonstration of MQX user/privilege task synchronization using the lightweight semaphore object.

lowpower	Demonstration of low power functionality on Kinetis and CF+ platforms.
msg	Simple demonstration of MQX inter-process message passing.
mutex	Simple demonstration of MQX task synchronization using the mutex object.
nil1	Even simpler than Hello World. A void application which may be used for copy/paste to start custom application.
rtc	Shows the Real Time Clock module API. Demonstrates how to synchronize RTC and MQX time and how to use RTC alarm interrupts.
sem	Simple demonstration of MQX task synchronization using the semaphore object.
spi	Shows how to read/write data from/to external SPI EEPROM. Additional HW setup is needed.
taskat	Shows how task can be created within statically allocated memory buffer (avoid heap allocation for task stack and context).
taskq	Shows custom task queue and how the queue can be suspended and resumed.
test	Shows the self-testing feature of each MQX component.
tfs	Shows the usage of ROM-based Trivial File System in an MQX application.
timer	Simple demonstration of MQX timer component.
usermode	Memory management and dynamic task creation from user-mode tasks.
watchdog	Simple demonstration of MQX task timeout detection using the kernel (not to be confused with watchdog) component.

## RTCS Example Applications

rtcs/examples/...

Name	Description
httpsrv	Simple web server with CGI-like scripts and web pages stored in internal flash.
shell	Shell command line providing commands for network management.
snmp	SNMP protocol example providing microprocessor state information.
telnet_to_serial	Simple character passing between the UART console and the telnet session. Shows custom "lightweight" telnet.

## MFS Example Applications

mfs/examples/...

Name	Description
mfs_ftp	RTCS FTP demo accessing the MFS filesystem mounted on the USB mass storage. For FTP example without USB functionality, refer to RTCS Shell demo.
mfs_usb	Console shell-based example showing how to access MFS filesystem mounted on the USB mass storage.
cfcard	Console shell-based example showing the MFS filesystem used with and CFCard storage.
ramdisk	Shows use of MFS accessing the external RAM (or MRAM).
sdcard	Shows use of MFS accessing the SPI-connect SD Card.

## USB Host Example Applications

usb/host/examples/...

Name	Description
audio/microphone	Able to connect USB microphone and record the sound to SD Card (wav format)
audio/speaker	Able to connect USB speaker and play the sound from SD Card (wav format)
cdc/cdc_serial	This example demonstrates the virtual serial port capability with abstract control model. Redirects the communication from CDC device, which is connected to the board, to the serial port.
hid/keyboard	This application echoes keys pressed on the USB keyboard onto serial console.
hid/mouse	Displays USB mouse events on serial console.
hid/keyboard+mouse	Keyboard and mouse demos combined in a single application.
msd/msd_commands	Executes the standard "mass storage device" commands to the USB disk and shows the response on the serial console (see MFS examples for USB filesystem access).

## USB Device Example Applications

usb/device/examples/...

Name	Description
audio/generator	Acts as USB microphone, playing out the short audio loop.
audio/speaker	Receive audio stream data from the host (PC) and play it out through I2S driver.
msd/disk	Implements small storage device in internal RAM memory.
hid/mouse	Creates virtual mouse which keeps moving in square loop, 100 pixels in size.
cdc/virtual_com	Implements a virtual serial line loopback.
cdc/virtual_nic	Implements a virtual network interface cards.
phdc/bridge	Ethernet to PHDC bridge. Receives Medical Data Exchange packets or APDU's from the Continua Agent Device on the Ethernet link as passes it to the USB Continua Manager. See <a href="http://www.continuaalliance.org">http://www.continuaalliance.org</a> .

## Demo Applications

demo/...

Name	Description
hvac	Simple implementation of console-based HVAC with optional USB logging and FTP access.
hvac_error	Intentional error injected to the HVAC demo code to demonstrate power of TAD plug-in.
web_hvac	The HVAC demo with HTTP server implementing the GUI. Ajax-based pages demonstrate an advanced use of HTTP server.
pe_demo	This example demonstrates how to use Processor Expert to configure MQX BSP.

## 5 Known Issues and Limitations

### 5.1 Performance of Code Running in MRAM

Runtime performance of the MRAM-based targets is approximately 8x degraded when compared to the Flash-based execution. MRAM is an external memory device connected to the ColdFire core by 8-bit data bus with one wait-state generated for each access. In order to fetch one 32-bit value (instruction) from the MRAM memory, four accesses need to occur. Each access inserts one wait-state clock.

This behavior is normal by design and applies to other processors using the external memory to store executable code.

### 5.2 Default Kernel Configuration of Small-RAM Devices

The default kernel configuration of small-RAM devices is optimized to run demonstration applications located in the `/demo` folder. To meet tight RAM constraints, some MQX or RTCS features are disabled by default. Additionally, some I/O drivers, which are not used by the main demo applications, are disabled.

Typically, a compile-time error message occurs if trying to run an example application while the required kernel feature or I/O driver is missing in the library code. To execute some example applications, enable the required features in the `/config/<board>/user_config.h` file and recompile all MQX libraries.

### 5.3 USB Host HUB Examples

HUB class support is enabled in HID example applications. The applications run correctly with the USB device attached either directly or through the hub. However, the example code only handles a single device. A combined Mouse+Keyboard demo handles one mouse and one keyboard simultaneously. The same kind of multiple devices, which are attached through the hub, cannot be used in the example applications.

### 5.4 OSBDM / OSJTAG Firmware Compatibility

The latest versions of the CodeWarrior, IAR, and Keil tools add a new version of the OSBDM / OSJTAG Debugger interface with improved performance and stability. The new interface requires a firmware update. See the instructions provided in the Release Notes which apply to the development tools you are using.

The latest OSJTAG firmware, drivers and tools are available on the following web page:  
[pemicro.com/osbdm/index.cfm](http://pemicro.com/osbdm/index.cfm)

## 5.5 Supporting “Hot Device Uninstall” in MQX I/O Subsystem

In the current implementation of the MQX I/O subsystem, the application is responsible for dealing with application tasks which have opened file handles while uninstalling a device driver.

A typical demonstration of the problem is USB mass storage handling:

When a USB attach event is detected, an application installs the MFS partition manager and MFS file system ‘device’ on top of the USB driver.

The application runs tasks, such as shell, which open and access files provided by the MFS filesystem device.

When the user unplugs the USB mass storage device, the application has a limited way to detect an opened file before uninstalling the MFS filesystem device.

The file I/O functions start reporting errors when accessing the device after it is physically detached. Design the application code so that the tasks close all files affected by the detach event before the MFS filesystem driver can be uninstalled.

If there is an attempt to close the MFS handle prior to closing all related files, a sharing violation error is returned. An example application "mfs\_usb" demonstrates how to close files by retrying the closing operation of the MFS handle. If a task keeps one or more files open for an extensive time period, use a suitable method to notify it about the ongoing filesystem un-installation.

This implementation may add an additional application overhead. MQX I/O subsystem is under enhancement process to ensure that file operations safely return error states even after the underlying device driver is uninstalled. This enhancement will simplify the application code error recovery.

## 5.6 TWR-MEM Compact Flash interface issues

Some Compact Flash cards do not work correctly with TWR-MEM and MQX CF card driver. The reasons could be the following:

An issue in the TWR-MEM CPLD code, Rev. A, causes incorrect communication with some types of cards, such as Kingston. A fixed CPLD firmware is available in

`<install_dir>/mqx/source/io/pccard/twr_mem_pccard_cpld/` folder. The firmware can be loaded to the TWR-MEM CPLD by using the Altera Quartus II design tool and a BLASTER connection cable.

In some cases, the MQX driver incorrectly detects the card in the slot. First, check whether all CF related jumpers are set correctly as described in the *Getting Started with Freescale MQX™ RTOS*. If the incorrect behavior persists, connect the two pull-up resistors between the card detect pins (CF\_CD1, CF\_CD2) and the 3.3V VCC.

## 5.7 Idle task is required on Kinetis platforms

The Kinetis kernel, by design, cannot operate without an idle task. The MQX\_USE\_IDLE\_TASK configuration options must be set to 1.

## 5.8 USB EHCI stack buffer restrictions

Align the buffers used by KHCI at 4B.

Align the buffers used by EHCI at cache line and the size to cache line boundary in the cached area. If the goal is to optimize performance, allocate the buffer used by EHCI in the un-cached memory space.



## 5.9 Flash Cache Disabled on TWR-K60N512 BSP – erratum e2647

A workaround is implemented for erratum e2647 on the MK60N512 based BSP. The workaround disables the use of the Flash cache for processors revision 1.0 (mask 0M33Z). It fixes sudden application crashes, but results in lower CPU performance (~30% performance hit). Newer processor mask sets with fixed e2647 erratum are not affected.

## 5.10 TWR-K60N512 BSP 256KB Flash boundary issue

During TWR-K60N512 BSP testing, an unexpected application crash occurs when the application code spans over the 256 KB boundary. The issue is related to the silicon revision 1.0 (mask 0M33Z) only. The TWR-K60N512 linker files are prepared so that the code is placed to the lower addresses and constant data is placed to higher addresses in the Flash memory.

## 5.11 ARP entries issue

When the board is put into a busy Ethernet environment with many ARP requests, the ARP entries cause memory fragmentation, which leads to `RTCSERR_TCPIP_NO_BUFFS` when `connect()` is called.

## 5.12 FlexCan driver issues

Several issues are identified during the development of the FlexCAN driver:

On TWR-K70F120M board, the TX/RX signals are not routed to the elevator by default and the FlexCAN example does not work. To enable the FlexCAN operation, solder the zero-ohm resistors, R22 and R23, on TWR-K70F120M board.

The 10-kbit baudrate doesn't generally work. FlexCAN detects bit0 errors in its own transmitted messages.

## 5.13 ARM MDK Keil µVision Support – Issue linking the TWR-K40X256 RTCS library

The Keil µVision linker fails to link the TWR-K40X256 RTCS-based application projects. The linker failure occurs because the Keil linker tries to place the functions which are not used in the final application. This issue was confirmed by ARM and can be solved by using `__weak` modifier before the definition of the failing function.

## 5.14 Low power modes and PE generated code related issues

TWRK20D50M and TWRK40X256 – the low power timer either cannot wake up the chip from LLS power mode or the wake up leads to chip reset.

TWRMCF51JF - the switch to VLPR power mode in 2 MHz clock configuration does not work. The chip doesn't acknowledge the power mode change in the PMSTAT register.

CW10.2 Processor Expert generated low power BSP code on K60F and K70F chips contains bugs that do not allow the usage of 2 MHz clock configuration and the VLPR power mode. These bugs are corrected in the MQX source code. Correction in PE will be released with the next PE version. The issue is fixed in the CW10.3.

## 5.15 Keil UV4 Mixing the C99 and C++ code in one project

The Keil  $\mu$ Vision does not support using both C99 and C++ options at the same time in a project setting. All MQX projects, except the C Plus application, are set up to use C99 compiler option. When using both C99 and C++ in one project, specify these options for each file.

## 5.16 Compilation warnings during CW10.3 GCC compilation

The warnings produced during CW10.3 GCC compilations will be resolved in next MQX releases.

## 5.17 LFN to SFN translation when creating a file not handled properly

When two or more files are created with LFN (long file name) containing special characters which map to the same indexed SFN (short file name), the SFN entry contains garbage characters.

## 5.18 The Timeline View does not show the graph for PPC platforms

Timeline View, part of the Performance Tool, does not show the graph for PPC platforms in online mode. To fix this issue, save the data into a file (MQX->Save Performance Data menu), then switch to the offline mode, and open the file which contains the data.

## 5.19 Android USB MSD cannot be interfaced

If certain types of Android phones are connected to the system, the attach event is not generated. The issue is currently investigated and will be fixed in the next MQX version.

## 5.20 HMI and User Mode functionality in CW10 GCC

HMI and User Mode functionality is not supported in the GCC compiler.

## 5.21 MFS does not check validity of directory rename

MFS\_Rename\_file() function does not check the necessary precondition when renaming a directory. If the directory is renamed to its own subdirectory, the directory becomes inaccessible and lost cluster chains are created.

## 5.22 CodeWarrior 10.3 & 10.4 GCC compiler requires latest version of New Project Wizard

Linker problems may arise during the compilation of the projects created by the New Project Wizard in CodeWarrior 10.4 & 10.3 GCC. There are two kind of possible issues: the explicit linker errors or the incorrectly linked binary (starts but does not reach the main function).

To fix this issue, either update the New Project Wizard to the newest version (at least 1.1.1) by using the "Help/Install New Software..." menu, or change the command line pattern in the linker section of the project preferences to the following:

```
"${ARMSourceryDir}/${COMMAND}" -Xlinker --start-group ${INPUTS} -Xlinker --end-group  
${FLAGS} ${OUTPUT_FLAG}${OUTPUT_PREFIX}${OUTPUT}
```

## 5.23 Special Considerations when using the Cache with Low Power Modes on K60F and K70F platforms

If the cache is enabled, entry into a low power mode may interrupt a cache operation in progress, leading to corruption of the cache content. To fix this issue, download a patch to the MQX code base and a more detailed description about the cache low-power limitation and workaround implementation by navigating to the following web page: [freescale.com/mqx](http://freescale.com/mqx). Note that the workaround code is not included in this MQX release. Instead, a build error is issued when building the K60F or K70F PSP with both cache and low-power services enabled.

## 6 Change Log

### Version 3.0.0 (December 5<sup>th</sup> 2008)

- This is initial release supporting the MCF52259 processor, M52259EVB, and M52259DEMO boards.

### Version 3.0.1 (January 22<sup>nd</sup> 2009)

- Small enhancements through the whole code base of MQX kernel and other components
  - Memory block “type” information added to all system memory allocations. TAD is now able to give detailed information about each memory block allocated by kernel or system component.
  - Dedicated memory allocation routines in RTCS, MFS, and USB simplify the memory pool usage.
- IPCFG Ethernet link monitoring features and automatic IP address binding functionality was added to RTCS. The “ipconfig” shell command replaces the old “bind”, “ifbind” and “dhcp” shell commands.
- The HTTP server in RTCS was re-written to enable multiple sessions to be served by a single task. This feature brings more reliable HTTP server while maintaining low memory requirements.
- USB Host HUB class was added.
- MQX User Guide and several other documents are included in the setup package.

### Version 3.1.0 (April 3<sup>rd</sup> 2009)

- USB Device low-level driver has been implemented for MCF522xx family and example applications were created for M52259EVB and M52223EVB.
- The “usb” subdirectory of the /lib output folder was split to “host” and “device” parts.
- Added PSP, BSPs, and other support files for MCF52223 and MCF52235 evaluation boards:
  - USB Host and Device libraries were ported to M52223EVB
  - RTCS library was ported to M52235EVB
  - MFS and Shell libraries ported to both new boards
- USB Host HUB issue resolved with MCF52259 (excessive number of errors observed when USB devices were accessed through USB HUB). The issue was solved by implementation of SOF frame scheduler in the USB Host low-level driver. An issue with a similar impact remains on MCF5222x implementation (silicon problem, no workaround known).
- Error codes naming and numbering convention has been made more consistent across MQX, RTCS, and other libraries. Old error code names remain implemented for backward compatibility (numeric values changed).
- Kernel Data and other internal structures were optimized for size. Parts of the structures were made conditionally compiled based on the user configuration.

TAD (CodeWarrior debugger plugin) was updated to handle renaming in internal structures.

- Memory type information was also added to lightweight memory structures. TAD is now able to display the type of information for memory blocks allocated by using lightweight memory allocator.
- Lightweight memory allocator has been made the default option for all BSPs.
- The `/config/<board>/user_config.cw` files were eliminated. Such files have contained a subset of configuration options for assembler-coded files in the MQX kernel. All kernel assembler-coded files are now preprocessed using C preprocessor and make use of standard C header macros. The C-language-specific content of the header files is conditionally excluded during assembler compilation by using `__ASM__` macro.
- The assembler `vectors.cw` file in each BSP was re-coded to C syntax. A new user configuration macro (`MQX_ROM_VECTORS`) may be set non-zero to avoid vector table being copied to RAM.
- The BSP, PSP, RTCS and other library build projects were changed to be easier to understand (virtual folders inside each CodeWarrior project was updated).
- SNMPv2 code was changed to enable ROM-based MIB structures. Example application is provided demonstrating custom MIB nodes and user traps.
- Several source code files were renamed to better reflect the content, especially in the I/O driver directories.
- Added “root directory” concept to the FTP server, “rename” command added.
- SPI I/O driver added (Master mode only). Example application is available.
- Initial ADC I/O driver was added. This driver (including its API) is subject to improvement in future releases. ADC device driver usage is demonstrated in ADC example application.
- RTC API updated, example application added.
- The new IPCFG API updated and documented
- The new HTTP server API documented
- MQX I/O Driver User Guide added

### Version 3.2.0 (May 15<sup>th</sup> 2009)

- Added PSP, BSPs and other support files for TWR-MCF51CN-KIT Tower Kit with ColdFire V1 processor MCF51CN128.
  - RTCS library was also ported to MCF51CN128 device.
- Ethernet driver significantly re-written and generalized to cover both ColdFire V1 and V2 processors. Application is now able to change some of the driver parameters dynamically, without need to recompile the driver code.
- Most of the features missing in the previous release of the ADC driver were implemented. The driver documentation is still on the to-do list.
- ADC driver partially re-written and low-level layer ported to ADC module of MCF51CN device. There are implementation differences between MCF51CNxx and MCF522xx drivers which are still to be addressed by future releases.
  - MCF51CN driver uses internal lightweight timer component
  - MCF522xx driver still uses PIT timer for internal timing. Implementation with lightweight timer (lower performance) is planned for future versions

- New GPIO callback-on-interrupt feature was added to GPIO driver of MCF51CN. This feature will be ported to other supported devices in future releases.
- Three new “security” demo applications were created for MCF51CN device demonstrating the ADC and GPIO drivers, network protocols and low-power mode of operation. The applications will be ported to other devices in future.

### Version 3.2.1 (May 28<sup>th</sup> 2009)

- Sparse interrupt table implemented and made a default option for small-RAM devices. This feature saves up to 1k of RAM. The TAD CodeWarrior plug-in was updated to support this feature.
- The legacy MQX Flash driver (called FlashX) was ported to support internal Flash memory of all supported devices (MCF51CNxxx, MCF522xx). The Flash driver requires Flash parameters to be passed from the application or linker command file. The linker command files were updated in all BSP. Sample application is provided.
- The GPIO driver callback-on-interrupt feature was ported to MCF52xx devices. This feature is still a subject to change in future versions. The MCF51xx driver was optimized for code size.
- Minor bug fixes in Serial driver (fflush system call implemented).
- MCF51CNxx SPI example enhanced to support both SPI EEPROM and SPI Flash devices. The example applications may be used with Memory Storyboard.
- Fixed known issue with MQX Stationery as it was described in the last release notes. The MQX\_PATH named source tree is no longer used for referencing the debugger initialization file (this was causing Flash Programmer crash). Copies of debugger initialization and memory configuration files were made part of all stationery projects and are referenced using relative path only in the debugger settings.
- I/O Driver User Guide updated. The GPIO, ADC, SPI and FlashX driver description added.

### Version 3.3.0 (Aug 7<sup>th</sup> 2009)

- Added BSP for MCF52259 Tower Kit.
  - CodeWarrior projects, demos and example applications were ported from M52259EVB to this BSP.
  - CodeWarrior projects and the BSP also support debugging targets and running code from external MRAM memory. This is prepared for future support of Memory Storyboard for the Tower Kit.
- Added PSP, BSPs and other support files for M54455EVB ColdFire V4 system. This is the first V4 device supported by Freescale MQX operating system. Few new features were added to PSP:
  - Cache control support added. Non-cached memory pool allocation API added.
  - Flash-to-RAM code copying enabled in startup.
  - Startup code is now part of PSP. Two files ([CF\\_startup.c](#) and [ROMCopy.c](#)) were reused from CodeWarrior runtime library. These files were modified to support ROM-to-RAM code copying.

- Flash Programmer config files available in `tools/flash_programmer_scripts/config` directory.
- SPI driver for M54455EVB created (the DSPI module), example supports onboard SPI Flash
- Ethernet driver and RTCS were significantly re-written.
  - Device-independent part of the driver was re-written to support multiple Ethernet MAC devices of the same or even different kinds. A support for memory-optimized handling of small frames was also added.
  - Device-dependent part of the Ethernet driver was re-written to support ColdFire V1-V4 Fast Ethernet Controller module.
  - Several RTCS features were re-tested and fixed to support multiple physical controllers.
- RTCS updated
  - The `rtcs_shell` example now uses the MFS lib (with RAM-disk) and supports all FTP commands (conditional compilation)
  - Minor changes in IPCFG API to support multiple devices, potentially of different kind than ENET.
- TAD updated
  - Strings displayed in TAD moved to separate text files available inside MQX installation.
  - New Ethernet driver screen added.
  - Symbolic reader enhanced some objects (`LWsem`, `LWevent`, ...) are displayed not only with address information, but also with symbolic names.
- Several key TAD-like screen dumps implemented also on embedded application side in PSP. These functions may be used to print out the TAD-like debugging data by the application.
- Shell library updated
  - RTCS support commands changed (`ipconfig`, `netstat`).
  - MFS format command was made available (also see `mfs/examples/ramdisk` application)
  - Shell interface to TAD-like screens dumps added.
- Workaround for MCF5223x Ethernet PHY auto-negotiation issue implemented according to chip errata.
- FlashX driver updated:
  - Support of external Flash devices made available. Tested with external flash memory devices of M54455EVB.
  - FlashX example application modified to support external flash1 device on M54455EVB.
  - Write protect ioctl command implemented for external Flash devices.
- USB Host updated:
  - Code refactored and generally updated (MASS storage class renamed to MSD).

- CDC class added. Example application shows how to forward characters between standard UART TTY and a virtual USB TTY device.
- I2C driver refactored and updated, eeprom example changed.
- CodeWarrior “Stationery” templates for creating new MQX projects were changed:
  - All kinds of stationery projects now have full set of libraries added to project (BSP, PSP, MFS, RTCS, USB Host, USB Device, Shell). The standard CodeWarrior linker will optimize out the unreferenced code. Feel free to remove the unused libraries from the project.
  - MQX-Only stationery is a simple “hello-world” like example.
  - MQX+MFS stationery is simple shell example working with RAM disk.
  - MQX+RTCS stationery is simple shell example with basic networking commands and telnet server.
  - MQX+RTCS+MFS stationery is a union of the two examples above.
  - USB Host is a simple application able to detect devices attached to the host port or devices attached through USB hub. You still need to refer to the USB host example applications to see how to interface attached devices.
  - The USB Device stationery is not available in this release. Please refer to the USB Device example applications.
- Device-specific sections added to this release notes document. Please read the required jumper settings for each evaluation board supported.

### Version 3.4.0 (Sep 25<sup>th</sup> 2009)

- Support for Register ABI (register parameter passing) was implemented
  - RegABI build targets (for both Release and Debug configurations) added in all library build projects. The binary libraries compiled with RegABI configuration get the “\_regabi” postfix.
  - Former build targets which use the Standard ABI are still maintained in the library build projects for backward compatibility. The targets were renamed to “StdABI”.
  - All example and stationery applications were reconfigured to use Register ABI and RegABI MQX libraries.
- M52277EVB BSP added
- M52233DEMO BSP added
- USB EHCI Host Support implemented
  - USB Host functionality enabled on MCF52277 and MCF54455.
- USB Device Stack reworked and enhanced
  - The USB Device Stack code has been partially rewritten to be consistent with the similar bare-metal stack available for Freescale HCS08 platform (released separately).
  - CDC class implementation examples added (virtual serial line and virtual network interface card).
  - PHDC medical class implementation examples added.



- SPI driver reworked to support all kinds of ColdFire SPI modules (SPI, QSPI and DSPI)
  - Former QSPI driver was removed from the release.
- IPC inter-processor communication files made available in the release.
  - IPC Example applications provided.
  - SCI-based IPC tested only.
- SPI-based SD Card Driver added
  - The driver was tested with Memory Storyboard for Tower Kit only (MCF51CN and MCF52259 devices). Not tested with M52277EVB board.
- Bug fixed: Wrong SCI baud rate divisors calculation fixed.
- Bug fixed: Shell “dir” command file attribute filter is now applied correctly.
- Bug fixed: MFS read and write calls correctly return negative value when physical device returns access error. Please see also another known issue described in section .

### Version 3.5.0 (Jan 26<sup>th</sup> 2010)

- CodeWarrior build projects location was changed. The `build/codewarrior` subdirectory was replaced by several directories depending on type and version of the CodeWarrior tool.
  - `build/cwmcu62` contains build projects for CodeWarrior Development Studio for Microcontrollers version 6.2 (ColdFire V1 projects from MQX 3.4 and earlier)
  - `build/cwmcu63` contains build projects for CodeWarrior Development Studio for Microcontrollers version 6.3 (New versions of ColdFire V1 projects)
  - `build/cwcf71` contains build projects for CodeWarrior Development Studio for ColdFire Architectures version 7.1 (ColdFire V2-V4 projects from MQX 3.4 and earlier)
  - `build/cwcf72` now contains build projects for CodeWarrior Development Studio for ColdFire Architectures version 7.2 (New versions of ColdFire V2-V4 projects)
  - The post-linker batch files were separated to `build/bat` out of the build folders to be able to reuse them between different build project versions.
  - Mass-build projects of all libraries were also moved from `config/<board>` folder to an appropriate subdirectory.
  - Build projects for the later CodeWarrior versions (`cwcf72` and `cwmcu63`) no longer support standard (on-stack) parameter passing. StdABI targets are not available in build projects.
- The code and folder structure was updated to make it ready for IAR toolset support
  - CodeWarrior-specific C and Assembler syntax was changed to be compatible with IAR compilers. Only few parts of code made conditionally compiled depending on the compiler.
  - CodeWarrior-specific portions of BSP (start-up code, memory initialization code, debugger configuration files) were moved to `cw` subfolder. The `iar` subfolder will be added by the IAR support patch in future.
- Changes in PSP
  - CPU numbering scheme was changed. The CPU is still defined in the `user_config.h` file, but uses constants pre-defined in `psp_cpudef.h` file.

- Several new code-size compile-time configuration options were added. See updated MQX User's Guide documentation.
- Changes in MFS
  - Dynamic read-only status checking of the physical device was added to MFS write functions as optional feature (MFSCFG\_READ\_ONLY\_CHECK). This option is useful for example with SD Card storage with write-protect switch.
- RTCS stack updated
  - Set of RTCS iwcfg\_xxx() functions was implemented to enable application-level control of any future WiFi functionality.
  - ARP resend and expiration timeout values were made configurable through the user\_config.h file. Especially the "expire incomplete" time needs attention as it affects the total time a send call takes when target IP address does not exist on the local subnet.
  - Default setting for DHCP DISCOVER message was changed. Broadcast flag is set to FALSE to request unicast DHCP responses by default.
  - Internal structure names were renamed to keep consistency with C typedef names. The structure tag names are now always the lowercase variant of the C typedef name. This change enables proper ELF file format parsing by Task Aware Debugger plugin.
  - Bug fixed – RTCS\_selectset function now behaves as in RTCS versions prior to 3.0. The socket array is not altered by this function.
  - RTCS\_select call enhanced to support more than one socket in the array.
- Shell library was extended by new "iwconfig" command to enable the iwcfg\_xxx() control from the Shell environment
- BSP and support for MCF51EM256-based DEMOEM board was added
  - SCI, SPI, I2C and other standard drivers were ported to this platform
  - ADC, RTC, FlashX were enhanced to support additional features the MCF51EM offers
- ENET Driver was updated
  - Generic Wi-Fi support was added into driver code for future WiFi support.
  - ENET MAC interface structure was extended to support both generic and device-specific control commands (so-called media control commands).
  - Set of Wi-Fi-related media control commands was defined (setting the ESSID, encryption, ...)
  - ENET driver initialization structure was extended by new device-dependent initialization parameter.
  - ENET driver concept of device context locking was enhanced (from global interrupt disable to lwsem-based locking).
  - HVAC, WEB\_HVAC, HTTPSrv and RTCS Shell example applications were made ready for WiFi functionality.
- SPI driver was updated
  - Support for SPI16 module of MCF51EM was added.

- Driver code for 8-bit ColdFire V1 SPI module was internally renamed to SPI8.
- The SPI driver now provides uniform I/O API to SPI8, SPI16, QSPI and DSPI modules.
- Bug fixed – QSPI chip select was incorrectly asserted during driver initialization making a glitch on this signal.
- GPIO driver was extended
  - Open flags were added to control polarity of edge which triggers interrupt callback routine to be called (rising/falling edge).
  - I/O control commands were added to enable/disable interrupt callback function to be called.
- ADC driver now supports the PDB trigger module and sixteen-bit ADC module available at MCF51EM.
  - ADC driver install function was changed to accept additional device-specific initialization structure.
- SD Card driver now supports the SD Card interface available on M52277EVB. SD Card Example application was added for this board.
- Example applications
  - SD Card example application now decodes GPIO inputs for card presence and read-only switch.
  - Code-size benchmark application was added for MCF52259EVB. The application makes artificial reference to MQX kernel API functions so the kernel code-size can be analyzed from map files. A tool automating the build, map file parsing, and HTML report generation is also available.
  - Timing benchmark application was added for MCF51CN, MCF52259 and MCF54455 devices. This application measures key timing parameters of the MQX kernel. HTML report generator tool is also available.
  - RTC example application enhanced to support new features available in MCF51EM.
  - SPI example application extended from EEPROM testing to general SPI-based memory device (EEPROM, Flash or serial MRAM).

### **Version 3.5.1 (Feb 19th 2010)**

- Fixed bugs:
  - MCF5225x ADC driver - pin initialization was not performed properly; problem appears with “Release” targets.
  - FlashX driver was not working correctly with internal Flash memory of MCF52xx processors.
- Example applications updated
  - MFS SD Card example application demonstrates use of Partition Manager now.
  - Code-size benchmark applications and results are available for M52259EVB, M54455EVB, TWR-MCF51CN.

### **Version 3.6.0 (Jun 21st 2010)**

- The support for older CodeWarrior for ColdFire 7.1 and CodeWarrior for Microcontrollers 6.2 was removed. Build projects are available only for versions 7.2 and 6.3.
- Eclipse-based CodeWarrior 10 support is added. All library and application build projects are also available in the native CW10 format.
- MCF51AC-based DEMOAC Board Support Package added with basic set of peripheral drivers: ADC, FLASHX, GPIO, I2C, INT\_CTRL, SERIAL, SPI.
- MCF51JM-based EVB51JM128 Board Support Package added with basic set of peripheral drivers: ADC, FLASHX, GPIO, I2C, INT\_CTRL, SERIAL, SPI. A support for USB Host/Device stack is also available.
- MPC5125 BSP added. This is the first PowerPC port supported by Freescale MQX. This version includes the code released recently as MPC5125 patch to MQX 3.5.1.
  - Basic set of I/O drivers supporting the MPC5125 peripherals is included: PSC UART, Fast Ethernet controller, DIU, msCAN and RTC.
  - Example applications available: Standard set of MQX and RTCS examples, the “Digital Sign” demo application demonstrating the graphical DIU module of MPC5125 and the HDMI interface.
- The MQX\_SUPPRESS\_FILE\_DEF and MQX\_SUPPRESS\_STDIO\_MACROS may now be defined as 1 in user\_config.h to avoid declaration of generic FILE, FILE\_PTR types and <stdio.h> functions like read, write, printf etc.
  - The MQX code was refactored to use the new MQX\_FILE and MQX\_FILE\_PTR types and make aliases to backward compatible FILE and FILE\_PTR types only if not suppressed.
  - The key changes were implemented in the file fio.h in mqx/source/include.
  - Default values of both configuration options are defined such that backward compatibility is not affected.
- The user\_config.h files and shared configuration files in /config/common directory were re-formatted and updated:
  - All BSPCFG options related to given board are now available in individual user\_config.h files.
  - MQX\_TASK\_DESTRUCTION and MQX\_EXIT\_ENABLED options are now set to 0 in small\_ram\_config.h file.
  - MQX\_COMPONENT\_DESTRUCTION option was removed from the verif\_enabled\_config.h file. The default value of 1 defined in mqx\_cnfg.h is used. The ColdFire V1 processors override this option to 0 in user\_config.h to save code size.
- A new \_lwevent\_get\_signalled() API call was added to enable detecting which lwevent bits caused the active task to return from \_lwevent\_wait() call.
- Inline assembler code was eliminated where possible to simplify migration to alternative C compilers.
  - MCF52xx and MCF51xx FlashX flashing routines were rewritten in C.
  - The “nop” instruction primitive \_ASM\_NOP() added to compiler-specific header file and used instead of asm(nop) inline assembler statement.
  - The “stop” instruction primitive \_ASM\_STOP(x) added to compiler-specific header file and used instead of asm(stop #(x)) inline assembler statement.

- Support for Freescale eGUI software and TWR-LCD display board has been added. See eGUI MQX demos within the Freescale eGUI package. The eGUI comes with set of low-level drivers for non-OS bare-metal operation. The drivers in MQX 3.6 enable the eGUI package to be configured for hardware-independent “MQX” mode where all board-dependent code resides in the MQX BSP.
  - Resistive Touch Screen driver added to TWR-MCF52259 and TWR-MCF51CN BSPs.
  - LCD memory regions added to TWR-MCF52259 and TWR-MCF51CN linker command files in BSPs.
  - GPIO pins, ADC channels and SPI channel specific for LCD and touch screen control are defined in TWR-MCF52259 and TWR-MCF51CN BSP header files.
- Inter-processor communication (IPC) over the UDP protocol was added together with an example application in RTCS.
- RTCS ipconfig shell command enhanced to be able to print immediate values of low-level PHY registers.
- Serial driver supports new open mode-flag (IO\_SERIAL\_NON\_BLOCKING) to enable accessing the serial driver in non-blocking mode. Supported by both polled and interrupt serial drivers.
- New I2C ioctl command (IO\_IOCTL\_I2C\_GET\_BUS\_AVAILABILITY) implemented to detect idle/busy condition on the bus.
- Register I/O map structures were updated to cover all supported parts and changes in I/O drivers. In addition to changes in processor specific header files, the following shared I/O map files have been changed:
  - mcf5xxx\_spi16.h ... SPI16 no-FIFO module added.
  - mcf5xxx\_usbotg.h ... ULPI control registers added.
  - mcf522x\_usbotg.h ... registers added
  - mcf51xx\_i2c.h ... I2C with/without SMB support defined
  - mcf51xx\_ftsr.h ... registers added
- Register I/O map structures of MCF54455 were expanded to cover peripherals not supported by MQX drivers (DMA timers, EPORT, PCI, RNG, SSI).
- The entire code was “untabified” (TAB characters replaced by spaces).
- Readme files created for selected example applications (GPIO, I2C, etc.).
- IPC demo application code generalized for various TTY ports availability.
- Codesize MAP-file analysis script now prints code-size statistics for RTCS, MFS and other system libraries.
- MQX I/O User Guide updated:
  - SPI, I2C, ADC, RTC and SD Card driver sections updated.
  - ESDHC, FlexCAN, DAC Driver and NAND Flash Driver sections added.
- Board-specific information related to MQX release has been moved from Release Notes into a separate document (see *Getting Started with Freescale MQX™ RTOS*).

## Bug fixes in 3.6.0

- MQX Kernel:
  - The `_io_fclose()` no longer locks the kernel's `IO_LWSEM` semaphore before calling the device driver's close function. This could cause a race condition with layered drivers.
  - The EWL library linking is now disabled in all MQX library projects (PSP, BSP, MFS, RTCS, ...). Having this option enabled in "Librarian" projects settings panel was causing all EWL code to be included in each MQX library. This was effectively disabling EWL function overrides from MQX code. For example the `malloc()` code was taken from EWL instead of from MQX and was causing linking issues (`malloc` is also part of C++ `new` operator implementation).
  - Handling of priority messages was fixed. In some cases a priority message could have been received from a wrong message queue.
  - Format specifier parameter is declared as `const` in the `scanf` prototype. This avoids compiler warning messages when passing constant string as the scan format.
  - The MQX watchdog component (do not confuse this with the on-chip hardware watchdog) was fixed in `_watchdog_isr()` to expire at proper time on ColdFire V1 platforms. Potentially this also solves problems in case the watchdog uses other timer than the main BSP tick timer.
- BSPs and I/O drivers:
  - EPURT interrupt level macros in MCF522xx board-specific headers in BSP are now set correctly according to the levels hardwired in the controller device. Be aware that `EPURT_EPF7` level is hardwired to non-maskable interrupt so no MQX calls may occur in the interrupt routine.
  - GPIO button pin identifiers (`BSP_SW1` and `BSP_SW2`) were fixed in M52233DEMO BSP header.
  - ADC driver functions now return more detailed error codes instead of generic `IO_ERROR` value.
  - MCF522xx interrupt initialization and unmasking functions fixed. Problem was that the code was also inadvertently unmasking other peripheral interrupt sources in the interrupt controller.
  - TWR-MCF51CN tick timer interrupt handler was fixed in BSP. Due to wrong timer interrupt acknowledge code, the MQX system tick time was one half of the configured value.
  - MCF51EM-specific code was removed from generic MCF5xxx RTC driver code.
  - I2C clock source changed from `BSP_BUS_CLOCK` to `BSP_SYSTEM_CLOCK` on all MCF52259-based boards. This fixes the problem of I2C bitrate which was 2x higher than configured.
  - Uncached memory region definitions were fixed in M54455EVB linker command files in BSP.
  - Default PLL system clocks setup was changed in M52277EVB BSP startup code. The MCF52277 processor now runs at system clock 160MHz (80MHz bus clock, 60MHz USB clock).
  - USB I/O pin initialization routine was fixed in M52277EVB BSP. The problem with pin settings made USB Host functionality unusable on M52277EVB.
  - SPI chip select control optimized for performance.

- MCF5xxx FEC driver initialization of 2<sup>nd</sup> Ethernet interface (ipconfig 1 init) is no longer dependent on having 1<sup>st</sup> interface initialized.
- Compiling MCF5xxx FEC code generates error message if un-cached memory support is disabled in MQX configuration.
- MFS Filesystem:
  - The CLUSTER\_SIZE\_BYTES member of MFS\_DRIVE\_STRUCT is now extended to 32 bit to avoid size overflow in case of 65k cluster size (solves some USB memory stick problems).
  - Hidden attribute checking was fixed in MFS\_Attribute\_match() function.
  - MFS write/append performance problem was solved by avoiding unnecessary sector reading.
- SHELL Library:
  - Memory violation error fixed in the shell “help” command implementation.
- USB Host Stack:
  - USB Host HUB class enhanced to support asynchronous status inquires.
  - USB CDC Host driver now waits for control transactions to ACM interface to complete. This fixes a problem of invalid memory access with some CDC devices.
  - KHCI USB bus control is deactivated after device is detached. This avoids problems where noise on detached bus was evaluated as a valid communication.
  - KHCI USB data bus weak pull-down resistors enabled to improve detection of reset/detach event.
  - EHCI fixed to toggle data tokens correctly when a transfer has odd number of transactions.
- Example applications:
  - Digital sign demo running on TWR-MPC5125 now runs the display task with higher priority than the network task. This fixes a problem where images were displayed after 3 minutes ARP timeout when board was not connected to network.
  - Bar-graph CGI code in HTTP Server demo now reads ADC channels properly.
  - msCAN example application fixed on TWR-MPC5125.
- Task Aware Debugger plug-in for CodeWarrior
  - The value of SR register is now displayed correctly in context of any tasks being examined.
- Other:
  - New Version Checker tool was fixed so it always runs in a single instance. Each of the TAD DLLs (current version and all backward-compatibility versions) starts an instance of the web checker tool upon CodeWarrior start. Running the tool in multiple instances was causing problems with loading the *webchk.wcp* configuration file or resulted in multiple “New Version Available” notifications being displayed.

### Version 3.6.1 (Aug 31st 2010)

- MCF51MM-based TWR-MCF51MM Board Support Package was added

- Standard set of peripheral drivers is enabled: FLASHX, GPIO, I2C, INT\_CTRL, UART, SPI, RTC, CF card, ADC, DAC, SD card
- New driver was added: DAC (API driver).
- USB Host and Device stacks are available. The host (KHCI) driver can be configured to include software workaround for early silicon issue of the MCF51MM device. This workaround is disabled by default.
- MCF53015-based M53015EVB Board Support Package was added
  - Standard set of peripheral drivers is enabled FLASHX, GPIO, I2C, INT\_CTRL, SERIAL, SPI, SDHC, CF card, SD card.
  - USB Host and Device stacks are available.
  - Two Ethernet interfaces are enabled in RTCS TCP/IP.
- New SDHC I/O driver covering the eSDHC peripheral module was created. This driver allows low-level access to various cards such as SD, SDHC, SDIO, SDCOMBO, SDHCCOMBO, MMC and CE-ATA. MQX currently implements upper layer for SD and SDHC card drivers only.
- HTTP server listen address can now be configured in application and may be optionally set to other value than INADDR\_ANY. This may help to run the web-server application on just one of several network interfaces.
- New TAD “Memory Blocks Summary” screen implemented for lightweight memory blocks. The screen shows memory allocation grouped by categories.
- The MQX documentation was updated:
  - MFS User Guide chapter 3.1.9 was updated (search attributes).
  - The lwevent description was updated in MQX Reference Manual.
  - The description of \_time\_diff\_xxx() function parameters corrected in MQX Reference Manual.
  - The \_RTCS\_mem\_pool setting documented in chapter 2.6 of RTCS User Guide.
  - IO Guide SPI driver description was extended by new IOCTL command.
- Release notes and Getting started documents were reorganized and extended by more detailed description of the MQX build process.
- New communication mode was implemented in SPI driver for the QSPI module. In this mode, the communication is split to smaller chunks (16 bytes) to fully leverage hardware-driven CS signal activation. Also in this mode, the CSIV bit is not controlled by the driver flush() function which is normally causing glitch on the other QSPI CS signals.

## Bug fixes

- The MFS functionality IO\_IOCTL\_FIND\_FIRST\_FILE was not working. In case the search attribute MFS\_SEAR v2CH\_NORMAL was used no files were found. Attribute matching algorithm was corrected and user manual was updated.
- MFS file write bug which appeared in MQX 3.6.0 append-file performance optimization was fixed. In case two or more files were opened simultaneously, wrong content could be written as the sector cache was not updated correctly.



- The interrupt enabling code in the GPIO driver was fixed for MCF5223x and MCF5225x devices. GPIO interrupt callbacks now work properly on these platforms.
- When the `_task_restart()` function was restarting the calling task itself, destroying task stack frame could crash the system. The behavior was fixed.
- Touch screen driver did not work properly for M5225X platforms and internal flash targets built in “Release” configuration. The driver code was fixed.
- Setting I2C baudrate did not selected the best closest achievable rate. Algorithm for baud rate calculation was changed to find closest match instead of closest-lower rate as it was before.
- FlashX driver fixed. The bug was causing writing incorrect data in case `fseek()` was used to navigate to the last byte of the Flash file and user wanted to overwrite it.
- Precompiled TWR-MPC5125 libraries were not included in MQX 3.6.0 release. The libraries are now part of the MQX 3.6.1 full installation package.
- New MQX projects created for MPC5125 platform were not using the `MQX_PATH` variable in the access paths. This might cause mismatch in library and include search paths in the newly created projects. This option is now set properly in project stationery.
- USB DATA0/1 bit toggling fixed in the EHCI USB Host driver code. This issue was affecting the PHDC and CDC data transfers.
- USB Device example applications for high-speed data access (EHCI) were updated to use internal data buffers in fast on-chip SRAM memory. This is a temporary solution which will be replaced by generic support of high-speed data transfer in future.
- The internal bind functionality was fixed in RTCS in order to enable the same TCP port being opened on two different network interfaces.
- Task suspended/running indication bug fixed in TAD. Call history is now displayed correctly for all tasks in both classic CodeWarrior and CodeWarrior version 10.
- Default PLL system clocks setup was changed back to MQX 3.5.0 values in M52277EVB BSP startup code. The SD RAM chip setting (160MHz) was unstable and caused sporadic errors during debugging.
- 
- The `lwtimer_create_periodic_queue()` functionality was fixed. Timer callback was not invoked in some cases.

### Version 3.6.2 (Nov 8th 2010)

- EAR v2 version of the Freescale Kinetis ARM Cortex-M4 Platform Support Package was prepared. See known issues related to this version in sections below.
- The Kinetis port of the Freescale MQX is supported by the IAR Embedded Workbench for ARM. The support includes:
  - The EWARM workspace files ready to be used to batch-build all MQX libraries.
  - Projects files for standard set of libraries: PSP, BSP, RTCS, MFS, USB Host and Device.
  - Project files for all example applications and demo projects. Projects are tested with on-board P&E Micro OSJTAG interface and external Segger/IAR J-Link interface.

- By default all projects are prepared for EWARM 5.50.6 and are configured for IAR J-Link debugging interface. The projects are ready to be converted to EWARM 6.10 format just by opening them in the new environment. In EWARM 6.10, the debugging connection may be changed to P&E Micro OSJTAG.
- MQX Task Aware Debugging plug-in for both EWARM 5.50 and EWARM 6.10 versions.
- MK40X256-based TWR-K40X256 Board Support Package was added
  - Standard set of peripheral drivers is enabled: GPIO, I2C, INT\_CTRL, UART, SPI, RTC, ADC, SD card and FlashX.
  - USB Host and Device stacks are available.
- MK60N512-based TWR-K60N512 Board Support Package was added
  - Standard set of peripheral drivers is enabled GPIO, I2C, INT\_CTRL, SERIAL, SPI, SDHC, SD card and FlashX.
  - USB Host and Device stacks are available.
  - Ethernet interface is supported by ENET driver and RTCS TCP/IP stack.
- Changes in the generic PSP code
  - Prototypes of functions which may be in conflict between the MQX and compiler-specific <string.h> header files were made conditionally compiled by the MQX\_SUPPRESS\_STRINGH\_MACROS configuration option. This suppression is enabled by default for IAR EWARM compiler version 6.
  - New memory allocation API function defined. The `_mem_alloc_at` and `_lwmem_alloc_at` functions enable to allocate dynamic memory block at specific address within the memory pool. This is used to implement memory “barriers” between physical RAM blocks which cannot be accessed as a continuous area. (this is the case for Kinetis platform).
  - Support for MQX exception handling implemented for sparse interrupt tables.
  - Kernel log entries added for new memory functions and low-level ENET driver functions.
- Changes in the I/O drivers code
  - New MAC-NET Ethernet module driver with hardware time-stamping support was added to support Kinetis platform.
  - RTC driver API enhanced to be able to work with MQX DATE\_STRUCT data type directly.
  - Minor changes done in ADC and GPIO drivers towards generalizing the code across multiple platforms. No functional change.
- Changes in the USB code
  - KHCI Host code significantly rewritten to support new Register I/O maps unified across ColdFire and Kinetis families. No functional change.
- The MQX documentation was updated:
  - Reference manual was extended by description of `_lwevent_get_signalled()` and `_task_create_at()` API functions.

- Description of interrupt-level task priorities and NMI handling were extended in the User Manual.

### Bug Fixes:

- In the PSP code
  - The `_task_get_template_ptr()` and `_task_get_td()` now safely returns NULL for invalid input parameter values.
- In the RTCS code
  - Compressed domain names haven't been handled correctly in the DNS client parser. Some DNS server responses were not handled properly.
  - Memory leak in the DHCP functionality fixed. Memory was not de-allocated properly during address renewal process.
  - DHCP client requests were not sent complete. With some servers the client was not obtaining the subnet information.
  - UDP service fixed in order to receive frames to a proper socket in case the same UDP port is open by multiple network interfaces.
- In the I/O drivers code
  - ADC examples were modified to work properly also on MCF51MM, MCF51AC and MCF51CN platforms.
  - MCF52259 GPIO\_PTCPAR\_PTCPA macro values were fixed.
  - KBI Interrupt functionality was fixed in the MCF51MM GPIO driver.
  - IO\_IOCTL\_I2C\_GET\_BUS\_AVAILABILITY command implemented in I2C driver for ColdFire V3 family.
  - IO\_IOCTL\_SERIAL\_SET\_DATA\_BITS command behavior fixed in serial for ColdFire V1 family.
  - Slave functionality was fixed in DSPI driver for ColdFire families. The slave was sending additional byte after normal data block.
- Other
  - Shell "help" command was fixed to display detailed description properly.

### Version 3.7.0 (Apr 8th 2011)

- New Board support packages:
  - MCF51AG-based TWR-MCF51AG BSP
  - MCF51JE-based TWR-MCF51JE BSP
  - MCF5208-based M5208EVB BSP
  - MCF5329-based M5329EVB BSP
  - MCF54418-based TWR-MCF54418 BSP
- Kinetis BSPs TWRK60X512 and TWRK40N256 modified:
  - Clock speed changed from 48MHz to 96MHz
  - Programmable Gain Amplifier – PGA enabled in ADC driver
  - Flash driver extended by FlexNVM functionality (EEPROM emulation)

- TWR-MPC5125 BSP was extended by:
  - USB Host and Device
  - NAND, SDHC, SD Card, SPI and I2C drivers
  - NAND boot targets and NAND Flashing targets are available. See details in MQX Getting Started document chapter TWR-MPC5125-KIT BSP.
- CodeWarrior Development Studio for Microcontrollers Version 10.1 support
  - TAD and New project wizard for ColdFire and Kinetis Platforms
  - MQX libraries, example and demo projects
- Kinetis BSP projects are CodeWarrior 10.1 Processor Expert Ready
  - Processor Expert drivers are enabled in MQX RTOS environment
  - Two BSPs with Processor Expert drivers enabled `<mqx>/build/twrk40n256_pe` and `<mqx>/build/twrk60x513_pe`
  - Example application demonstrating PE functionality `\demo\pe_demo`
  - Processor Expert drivers are supported for Kinetis platform only
- Kinetis scheduler enhanced
  - The `_int_install_kernel_isr()` call to install a non-MQX interrupt service routine into interrupt table is supported
  - The RAM-based vector table is supported.
- New smaller and faster version of GPIO driver prepared – LWGPIO driver. The driver is currently available for MCF52259 and MK40 and MK60 based BSPs only. See <doc/MQXIOUG.pdf> documentation and `\mqx\examples\lwgpio` example application for details. The driver will be ported to other platforms in future MQX releases.
- Serial driver was extended by new options allowing RS485 half duplex communication. New `ioctl` command for handling HW flow control (RTS/CTS signal) and SCI parity were added.
- The output directory for MQX PSP and BSP libraries has been changed – instead of `lib/<board_name>/mqx` are newly used `lib/<board_name>/psp` and `lib/<board_name>/bsp` folders. See [doc/FSL\\_MQX\\_Porting\\_Guide.pdf](doc/FSL_MQX_Porting_Guide.pdf) for instructions if you are porting the project based on older MQX releases.
- `Mem_copy` and `mem_zero` functions were set to speed optimized version by default. The code is longer but up to seven times faster than original implementation.
- Byte-swapping macros optimized and unified in PSP library (`psp.h`)
- Handling of multicast packets has been simplified in RTCS, no functional change.
- Task priorities changed from 5 to 8 in several example applications. Note that task priorities below 7 have special functionality with regards to interrupt masking.
- HTTP server reworked. The server is able to operate in one of three modes: polled mode, static task pool or dynamic task created per session. See RTCS documentation for details.
- MFS USB example application modified to support multiple USB sticks simultaneously
- USB Audio Class support added into USB Host stack for KHCI based platforms. Example application prepared for ColdFire V2 and Kinetis based platforms (`usb\host\examples\audio\audio_stream`)

- TCPIP stack (RTCS) is newly available for platforms without Ethernet controller. Ready for WiFi and PPP support.
- CodeSize script and reports updated. The script is able to generate detailed reports for PSP, BSP and newly also RTCS, MFS and USB libraries
- User documentation was updated

## Bug fixes

- Management of free memory for Kinetis devices using `_mem_alloc_at` or `_lwmem_alloc_at` allocation has been fixed. Using `_mem_alloc_at` could lead to losing part of free memory.
- USB Virtual\_com example code was fixed. Communication buffers are now allocated in un-cached memory.
- Serial driver: The `IO_IOCTL_SERIAL_TRANSMIT_DONE` command was corrected. The `IO_IOCTL_SERIAL_SET_HW_SIGNAL` and `IO_IOCTL_SERIAL_SET_PARITY` commands were implemented for all platforms
- SDHC sporadic read errors problem solved for Kinetis and MCF54418 devices. Default baud-rate has been reduced. Adaptive baudrate setting will be implemented in future MQX released.
- The MCF54418 GPIO driver interrupt related functionality has been fixed.
- The NAND ECC and back block byte handling in the MCF54418 NAND Flash driver has been fixed. Wrong ECC size was defined in BSP header file.
- Several USB Host stack EHCI bug-fixes improving stack stability and compatibility with some mass storage devices (Error handling during enumeration improved, improved packet receiving algorithm)
- MFS free cluster counting has been fixed.
- ROM section setting corrected in the linker files for MCF51CN, MCF51AC and MCF51JM BSPs.
- `MFSCFG_FAT_CACHE_SIZE` set to value 2 for all configurations to improve USB Mass Storage write speed for devices with EHCI peripheral module.
- `IO_IOCTL_SERIAL_TRANSMIT_DONE` result was inverted bug was fixed
- Register definition file MCF54455 platform was corrected (`mcf5445_edma_struct`)
- QSPI baud rate calculation source code was unified across all platforms. The QWR setting was corrected.
- USB EHCI Host does not handle data properly, the packets are sometime rejected – fixed in asynchronous scheduler; the code accessing overlay area was removed
- M53015EVB board does not start without debugger – fixed in BSP initialization code for this platform
- `psp_priv.s` file kernel logging related compilation check was corrected
- HVAC and Security Webserver demo applications was updated to run on Kinetis platforms and IAR compiler
- SNMP compilation macros have been fixed. The `RTCSCFG_ENABLE_SNMP` macro was added to all SNMP related files.

- MPC5125 RTC example build error has been resolved
- LWDNS behavior was corrected for TWRK60X512 platform. Endian swapping macro was incorrectly used.
- GPIO driver – fclose() on Kinetis based platforms was corrected. The function incorrectly uninstalled the interrupts for other open pins.
- USB Host stack enumeration error handling has been improved. The stack is trying to re-assert transaction if error occurs.
- USB Host stack reviewed to correctly handle Big to Little endian conversion. USB MSD device class endianness problem fixed on Kinetis platform.
- ADC driver for ColdFire V1 platforms was corrected – wrong use of lwevent in conversion complete lead to incorrect signaling of the end of measurement
- Compilation errors if RTCS\_ENABLE\_IP\_REASSEMBLY was enabled were corrected
- The file lwe\_gets.c (lw event get signaled) was not included in PSP libraries for all platforms
- Function FLEXCAN\_Rx\_message() returned FLEXCAN\_NO\_MESSAGE when FlexCAN message box overrun. The function has been changed to return FLEXCAN\_MESSAGE\_OVERWRITTEN.

#### Version 3.8.0 (Dec 15<sup>th</sup> 2011)

- New Board Support Packages added
  - TWR-K70F120M
  - TWR-K53N512
  - KwikStik (K40X256)
  - TWR-MCF51JF
  - TWR-MCF51QM
  - MPC8308-RDB
  - TWR-PXS20
- MQX, IO, RTCS, MFS and USB documentation updated. New USB User Guide documents created.
- New documents prepared to describe tool-specific settings for TAD and other MQX debugger plug-ins for CodeWarrior 10.x, IAR EWARM, ARM MDK - Keil  $\mu$ Vision environments.
- The Kinetis port of the Freescale MQX is newly supported by the ARM MDK – Keil  $\mu$ Vision IDE. The support includes:
  - Projects files for standard set of libraries: PSP, BSP, RTCS, MFS, USB Host and Device.
  - The uvmpw workspace files ready to be used to batch-build all MQX libraries.
  - Project files for all example applications and demo projects. Projects are tested with on-board P&E Micro OSJTAG interface and external ULINK® interface.

- All projects are prepared for  $\mu$ Vision 4.22a and are set up for OSBDM debugging interface. The debugging connection may be changed to ULINK debugging interface – see more information in the Getting Started document.
- MQX Task Aware Debugging plug-in.
- The Cortex-M PSP (Kinetis kernel) has been modified to be able to create tasks running in user (restricted) mode. The lightweight memory management, dynamic task creation, lightweight semaphore, event and message API functions were implemented to operate from user mode. This is an experimental prototype of the feature available for K60 and IAR build tools only in this version.
- HW floating point support added to Kinetis PSP. This feature is supported on the new MK70F120M platform.
- New LWGPIO Driver - smaller and faster GPIO driver was implemented for all supported platforms. MQX example applications were updated to use this driver. The support of legacy GPIO driver is discontinued and it will not be ported to future platforms.
- New I/ODebug I/O driver was created. This driver allows redirecting I/O functions, such as printf, to a debug probe communication channel. The CodeWarrior 10, IAR EWARM 6 or Keil  $\mu$ Vision4 debugger consoles are supported. See the Getting Started document for details on how to setup and use this feature.
- The FlashX driver and example application were rewritten to achieve effective operation and support of non-continuous Flash memory space used on Kinetis processors.
- I2C Slave driver functionality was re-implemented and new example applications have been created for all supported processors.
- TCHRES (Resistive touch screen) driver and example application demonstration for TWR-LCD board.
- The Ethernet boot loader example application has been created – this example application provides functionality similar to the well-known U-Boot application. Only high-end platforms with external RAM are supported. Example contains implementation for both NAND and NOR flash memory devices.
- New C++ example application was created and C++ functionality was enabled on all platforms.
- Example project wizard introduced – users are able to create MQX example application projects directly from the CW10.x IDE.
- Added preliminary support for CodeWarrior 10.2 Beta. This requires manual installation of the CodeWarrior 10.2 plug-ins; see the Getting Started document for more details.
- Created Working Set Importer plug-in in the CW 10.2. It allows developers to export and import Working Sets to/from XML file. The aim of this plug-in is to let the developer to import whole set of MQX libraries required for a chosen board at once. This feature is not supported for CW10.1.
- USB Host and Device code have been modified to support both little- and big-endian USB peripheral module.
- IAR EWARM 6: Assembler-coded routines have been annotated by special CFI directives. This enables the call-stack history to be displayed properly in the IAR IDE. Use the latest EWARM TAD plugin from `<MQX install dir>\tools\iar_extensions\Embedded Workbench 6.X`.

- Kinetis register description header files have been updated to reflect the latest Reference Manual.

### Bug fixes:

- HTTP Server web pages which contains CGI does not work correctly in Chrome and Android browsers - Root cause for this issue was in inappropriate formatting of HTTP responses for CGI scripts and wrong Content-Length header field when the content is created dynamically.
- The `_mqx_exit()` call was eliminated from all examples (replaced by `_task_block()`) to allow stdout output channel to be flushed properly and also to enable easier debugging of example applications in case of errors.
- TFTP client cannot download file form TFTP server – incorrect use of `read()` fixed in the `tftp` shell command.
- MQX crash under ping-flood stress test of RTCS applications – Issue was caused by incorrect Ethernet TX and RX priority setting. The interrupt priority setting has been corrected (Rx and Tx interrupts need to be set at the same level) in all BSPs.
- RTC applications build fail for MPC8308.
- Task destroy does not clean lwmem-allocated blocks memory - the lwmem blocks are now freed during the owner's task destroy.
- Documentation updated for `ENET_initialize()`.
- MTIM timer does not run on MCF51JF and MCF51QM - new MTIM16 driver implemented for these platforms.
- Fixed GPIO IRQ functionality with on DEMOEM board.
- MQX uses old Kinetis header files - all Kinetis header files has been updated to new file revisions.
- MPC8308RDB - missing `ipsum.cw` in installer package added.
- Log application build fail for `Kwikstikk40x256`.
- I2C interrupt priority levels on Kinetis corrected - set to 5, sublevels removed.
- When selecting second alternative interface on the USB device from MQX USB host, the host stack freezes on searching pipe handles for that interface.
- SPI timing issue in DSPI driver - DSPI driver now calculates appropriate delays corresponding to the selected baud rate.
- The UART2 channel setting on the TWR-MPC5125 was fixed.
- Serial driver `IO_IOCTL_SERIAL_GET_STOP_BITS` missing - functionality added to serial driver for Kinetis K40, K53, K60 and ColdFire MCF51QM, MCF51JF
- MQX does not check all registered DNS servers - fixed.
- `snprintf` function was fixed.
- Multiple initialization of DNS - `ipcfg_add_dns_ip()` has been removed.
- Race condition in `_task_get_id_from_name` and `_task_get_template_index` has been fixed.
- MPC8308 output libraries file names are incorrect – projects were re-generated with the correct name.



- Telnet performance increased by IOSOCK buffering.
- The RTCS telnet echo parameter option has been implemented.
- The MPC5125 MQX decremter initialization bug fixed.
- Incorrect pin handling in the legacy MCF5225x GPIO driver was fixed.
- Incorrect Target Settings of the MPC8308 projects fixed.
- Internal implementation of taskat function fixed.
- MPC5125 MQX FIFO corrected
- MPC5125 CodeWarrior v 9.2 New Project Stationeries were fixed
- Printf and scanf support for 64-bit numbers and %L modifier.
- Fixed printf, sscanf and scanf functions not working when using floats (%f) on Kinetis.
- Floating point libraries setting fixed for MCF52277 platform.
- MQX does not support %g %G printf() format specifiers – the missing functionality was implemented.
- ADC driver for Kinetis does not handle some input channels correctly.
- Bug in MCF51AC ADC driver provides invalid results.
- Kinetis linker map file did not make use of 32 Kbytes RAM.
- Problem in reading free space in USB stick.
- K60 pin muxing in “init\_gpio.c” for UART4 (ttye).
- TWRK60x512 cannot attach USB Keyboard.
- MPC5125 SD card driver cannot read 16GB cards.
- ENET\_shutdown() does not clean up all resources.
- Bug in RTCS select() if timeout is less than 1 millisecond.
- RTCS tick timer event scheduling fixed.
- MCF54455 RMII setting fixed.
- Interrupts pending from a boot loader before MQX starts may cause unhandled exception.
- WebHVAC application crashes on TWR-K60N512 board – the issue partially resolved by disabling Flash Cache - erratum e2647 for early silicon revisions (mask 0M33Z, see known issues for details).
- WebHVAC sporadic erratic behavior observed also with Flash Cache disabled on the early K60N512 silicon revisions (mask 0M33Z) when using IAR EWARM tools. The linker file was changed to put code to smaller addresses and constant data to higher addresses in the Flash. Application crashes not observed anymore after the change.
- Incorrect initialization of I2C2 and I2C3 peripheral drivers fixed on TWR-MCF51JF and TWR-MCF51QM.
- Issues with writing large file to USB memory stick on Cayman - fixed.
- PHDC weighscale application bug - missing reset state handler in phd\_state\_mc\_func functions array - fixed.
- \_time\_ticks\_to\_xdate() produces incorrect values - fixed.

- The ip.c file does not compile when RTCSCFG\_ENABLE\_NAT is enabled - fixed.
- Documentation of lightweight object destruction corrected.
- Nested comment in 5329 code has been removed.
- Documentation corrected – removed features which are not supported by current release (IO\_IOCTL\_SERIAL\_START\_BREAK and IO\_IOCTL\_SERIAL\_STOP\_BREAK).
- MPC5125, setting invalid MQX time when RTC is enabled - Add check of \_time\_from\_date function return value.
- TAD in IAR EWARM 6.21.4 freezes - LWMEM pools TAD screen has been fixed.
- TWR-MCF5418 demo web\_hvac cannot browse webpages from USB - Problem with memory cache. Changed MFS\_mem\_alloc\_system\_zero function - replaced \_mem\_alloc\_system\_zero with \_mem\_alloc\_system\_zero\_uncached.
- \_task\_get\_td() returning NULL when using SYSTEM\_TASK\_ID as parameter - fixed.
- Legacy GPIO driver interrupts handling corrected on MCF51 platforms.
- Boot loader example application and documentation has been updated. Line buffers were not handled correctly (function bootstrap\_ddr\_to\_nand(void))
- Serial port driver IO\_SERIAL\_HW\_FLOW\_CONTROL I/O control command fixed for interrupt mode on MCF5225x and Kinetis platforms
- USB CDC host fread() functionality fixed. The bug appeared in case application tried to read less data than received. This bug caused freezing of the CDC IN pipe.
- MCF54418 RTC driver fixed – Improved register unlock procedure is used in bsp\_rtc\_io\_init() to ensure that RTC is correctly set.
- The TWRMPC5125 timing was inaccurate (33,333 MHz oscillator vs. 32,768 MHz). Fixed the frequency values in the lcf files.
- DSPI baudrate calculation code modified - Baud rate calculation was extended by DBR setting to provide wider set of frequencies. Change is relevant for Kinetis and ColdFire processor families.
- All Kwikstikk40x256 host examples removed from release package - Kwikstik supports USB Device only.
- The MPC8308RDB I2C fixed - onboard I2C EEPROM uses 2 byte-wide memory addressing. I2C example was updated.
- TFTP client compatibility improved - TFTP read operation should ACK packets with already passed IDs and drop those with future IDs for better compatibility with some TFTP servers (www.dhcpserver.de). Timeout prolonged to 5 sec.
- Fixed PPP over UART client authentication code.
- Function ungetc() did not behave as expected. The ungetc behavior has been re-implemented. Other functions like ftell, fread, fgetc, fflush and fseek were made ungetc aware.
- Kinetis Int RAM Targets with vector table in RAM do not work correctly – problem fixed in IAR linker command file.
- Printf formatting correction - The hex formatting (%Nx) does not print leading zeros for integers - fixed.

- Incorrect usage of MQX\_USE\_SMALL\_MEM\_ZERO macro in `_mem_zero()` function for PowerPC could lead to the non optimal code generation.
- SPI baudrate was incorrect on MPC5125 due to incorrect endianness of baudrate divisor value in CCR register - fixed.

#### Version 3.8.1 (Jun 29<sup>th</sup> 2012)

- New Board Support Packages added
  - TWR-K20D50M
  - TWR-K20D72M
  - TWR-K40D100M
  - TWR-K60D100M
  - TWR-K60F120M
  - TWR-PXS30
  - TWR-PXN20
- Introduced new LWADC driver implemented for PX processor family. Documentation updated, new example application `mqx/examples/lwadc` has been prepared
- TWR-K70F120M BSP update
  - System cache has been enabled, resulting multiply time higher speed in IntFlash DDR data targets
  - USB High Speed (EHCI) operation has been enabled
  - Processor Expert support has been added (allowing use of PE drivers and Low Power features)
- TWR-PXS20 BSP update
  - Added LW ADC driver
- USB Host library has been updated to be able to handle both Full Speed (EHCI) and High Speed (KHCI) modules simultaneously.
- User mode support extended to Keil and CW10 toolsets for K60N512 BSP
- The Low Power features and Processor Expert support have been extended to TWR-MCF51JF and TWR-MCF51QM BSPs
- Sleep on Idle – MQX RTOS implements possibility to put processor core to the sleep mode and save energy during execution of the idle task. This feature is currently available for ColdFire and Kinetis platforms only. See details in `MQXIOUG.pdf` chapter Low Power Manager.
- Low Power Timer driver has been implemented and the functionality has been demonstrated in the `mqx/example/lowpower` application.
- SDCARD and DSPI and SDHC driver improvements
  - The clock speed of setting of communication channel (SPI, SDHC) has been corrected
  - DSPI polled driver optimized for performance, increased throughput at higher baud rates

- CRC calculation in `sdcard_spi` driver implemented using a table, significant performance increase
- SDCARD drivers prepared to make use of multi-sector transfers in the future (next release)
- ESDHC driver does not use intermediate buffer for data freeing up 0.5 KB of RAM

### Bug fixes:

- MPC5125 serial with interrupt is not working – serial driver `putc` routine has been fixed
- I2C driver on `mcf51jm` gets stuck – prepared workaround using short wait loops before reading of IIC status register
- Compiler warnings for CW10.2 - `#pragma opt_branchtailmerge off` – removed obsolete compiler macro
- User mode tests fail in the INTRAM targets – linker command file sections has been corrected
- NFC Driver - incorrect ECC status word reading - ECC status word reading fixed
- SPI example application failed for TWRPXS20 BSP chip select of connected accelerometer was disabled by pin pull-up resistor and rx buffer overflow was prevented in interrupt driver ISR
- MQX Telnet server update – incorrect ECHO handling
- Support negotiation of binary mode.
- Introduced new modes CR LF, CR NULL and character block binary mode.
- Introduced new IOCTL commands to switch between modes.
- Update demos to handle character mapping between application and telnet NVT.
- RTC for `mcf5xxx` - date gets out of valid range due to wrong `RTC_BASE_YEAR`
- SPI clock line is not set up during `spi_init` TWR-MPC5125 BSP– BSP GPIO setting fixed, added missing IOCTL commands
- Printf for floating types compatibility issue - Differences between gnu libs and MQX representation was fixed.
- `ipcfg_bind_staticip` do not fail if the same address is present on network - `IPIF_bind()` corrected
- The C library `exit()` function call was removed from MQX all example applications (replaced by `_task_block()`) to allow stdout output channel to be flushed properly and also to enable easier debugging of example applications in case of errors.
- In the full memory allocator, the highwater mark was updated incorrectly when an allocation was made in the pool extension area (created by `_mem_extend()`) - Now the highwater mark is updated only if block is allocated from the main pool area - so it shows valid information. The highwater mark is not available for pool extension areas.
- The TAD was fixed and updated to display all allocated memory blocks, even the ones from the memory extension areas created by `_mem_extend()` call. In the previous implementation, the TAD might have failed to display the memory blocks correctly in case the extension area was created.

- Linker Command File TWRMPC5125 modified - kernel\_data start symbol is placed automatically after statically allocated data now.
- PAP authentication fixed for PPP connection - problem has been fixed in both PAP&CHAP state machine and example application. See example readme, pap.c and chap.c files for details.
- Functionality for length modifiers (h hh and L ll) was implemented for printf(), fprintf(), sprintf(), snprintf() functions.
- No source available while debugging the TWR-PXS20 platform - All example project for have been updated with correct setting ("Generate DWARF information")
- Solved problem with reattaching the low speed devices to MQX USB Host. The attach failed every second time because of incorrect speed handshake algorithm
- Data and instruction cache has been enabled for TWRK70F120M board in default setting
- TWRK70F120M – linker command files for intflash\_ddrdata and intflash\_sramdata have been fixed – flash (ROM) space is fully used now
- IPC UDP example was not working properly. IPC used default initialization structure because no custom one was passed. '\_task\_create' call was modified and it now correctly passes this structure to IPC initialization task.
- Bug in case Kinetis ADC fopen/fclose was called multiple times fixed in PDB initialization.
- Flashx driver on twrmcf51mm and twrmcf51je supports now 2 banks of internal flash memory.
- Kinetis serial driver now allows to set 1 or 2 stop bits using IOCTL command IO\_IOCTL\_SERIAL\_SET\_STOP\_BITS.  
Since the HW doesn't support variable number of stop bits, the second stop bit is emulated using 9bit communication with last bit set to 1.  
The 2 stop bits configuration is therefore available only in 8 bit data mode with parity disabled.
- MQX-memcpy() failed, when KEIL-optimization is enabled – mem\_copy functionality corrected and by default the MQX uses use runtime (tool provided) memcpy instead of mqx\_mem\_copy to achieve best performance
- MFS\_Get\_disk\_free\_space() - fixed free space calculation in MFS
- When ping request expires due to timeout, the associated ARP request queue is being cleared. The clearing code was fixed so it doesn't block RTCS task in case entries in the queue don't match the ping request.
- IODEBUG documentation has been added to MQX IO guide

#### Version 4.0.0 (Dec 7<sup>th</sup> 2012)

- New board support packages added:
  - TWR-K21D50M
  - TWR-PXD10 - MQX kernel was ported to e200 VLE assembly instructions
- MQX ported to new compilers and build tools:
  - Makefiles for command-line GCC build provided for selected Kinetis BSPs (TWR-K60N512, TWR-K60D100M, TWR-K70F120M). See the makefile files in the [/build/<board>/make](#) folder.

- Ready for new Freescale Vybrid PSP and Cortex-A5 processor core support.
  - The MQX 4.0 has been tested with the Cortex-A PSP and Freescale Vybrid BSP libraries which are available as a standalone installation package.
- Ready for new Freescale NAND Flash File System library.
  - The MQX 4.0 has been tested with the new FFS library which implements the wear leveling algorithms and support for on-chip NAND Flash controller.
  - This library enables the MFS filesystem to be installed and used on top of the NAND Flash memory devices.
  - The FFS library is available as a standalone installation package.
- RTCS stack was extended by IPv6 protocol support, functionality provided was fully tested by legacy IPv6 Silver logo test set. The IPv6 support is available in the separate installation package. Currently supported protocols and services are:
  - IPv6 IP layer, ICMPv6/ping, HTTP server
- The following board support packages were removed from the release and will be supported only as a part of MQX 3.8.1 version maintenance.
  - TWR-MCF51AG, TWR-MCF51CN, TWR-MCF51JE, TWR-MCF51MM, TWR-MCF51QM, DEMOEM, DEMOAC, EVB51JM128, M5208EVB, M52223EVB, M52233DEMO, M52235EVB, M52277EVB, M53015EVB, M5329EVB, M54455EVB and MPC8308RDB
- Consolidated source files with generic kernel implementation. Number of PSP files was reduced which results in faster library build time.
- Support of CodeWarrior Classic development environment was dropped and all library and example projects were updated to CodeWarrior 10.2.
- Pre-compiled binary libraries are no longer included in the installer. For detailed build instructions please refer to “*Building the MQX Libraries*” section in the “MQX Getting Started” document.
- The MFS read and write speed was improved leveraging the multi-sector transfer functionality and reworked SPI and SDHC drivers. Measured maximum speed is up to 10 times higher in case 32K blocks are used (~10MB read/2,5 MB write for class 10 SD Card and TWRK40D100@96Mhz board)
- RTCS TCP/IP code has been updated and throughput has been improved.
  - Measured by Freescale FNET fBench tool: TCP TX~11Mbps; TCP RX~5Mbps; UDP TX ~25Mbps; UDP RX~24Mbps tested with TWRK60D100@96Mhz board at default configuration.
- New I2S & SAI audio drivers added for Kinetis processor family. Example application is provided.
- USB Audio examples updated – microphone and speaker examples are available for selected platforms.

#### RTCS Bug fixes

- The dual\_webserver example application was removed from release and current webserver application has been extended to handle two and more Ethernet interfaces instead
- FTP long file names could cause stack overwrite –file name handling corrected.

- Several PPP bugs resolved during the PPP code update (PPP PAP authentication, shutdown, reconnect etc.). New RTCS shell commands for PPP handling were introduced.
- A return value of the `pseudo_phy_discover_addr()` in the `pseudo_phy.c` source file was added.
- The HTTP server in static task setting had the hard coded priority - the static task priority value is derived from `HTTPDCFG_DEF_SERVER_PRIO` macro now.
- The HTTP server was extended by "keep-alive" functionality.
- The `RTCS_selectall()` documentation was modified to match intended functionality – the `RTCS_selectall()` waits on all sockets in the system
- DNS problem was caused by disabled loopback interface in the default RTCS configuration. Fixed by proper configuration in the `rtcscfg.h` file.

#### Build Tools related bug fixes

- Several changes were made in the CodeWarrior projects to be compatible with the upcoming 10.3 version. For example, use of slash in the configuration name has been removed and more compatibility changes done.
- Missing nandflash example was added to the New Project Wizard
- Performance Tool documentation link was fixed and documentation was updated.
- Invalid project paths referencing the `PE_LDD` macros were removed from the PE-enabled projects.
- Fixed C++ operators new/delete to be working correctly in Keil IDE. Consolidated overriding of `malloc`, `calloc` and `free` functions to use memory allocation system of MQX for all compilers.

#### MFS / IO subsystem bug fixes

- Attempt to delete open file is correctly denied with `MFS_SHARING_VIOLATION`.
- The `_io_mfs_dir_read()` API time and date handling corrected. Added check for month value, if the value is not between valid range (from 1 to 12), the "???" string is printed out instead of month abbreviation.
- The `fseek/fwrite` functionality was fixed to work according to standard in case the `fseek()` moves position beyond EOF (the file is extended by zeros to the `fseek` position upon the following `fwrite`).
- Functionality of `_sem_wait_ticks` fixed on ColdFire processor. The functions did not waited for expected number of ticks.
- The `printf()` function fixed to correctly print out the zero character and also other formatting now matches the GCC standard style.
- Corrected printing of different sized variables on big endian platforms (PPC, ColdFire)
- Shell `dir` command has been modified to correctly display directory content for file systems which are not current.

#### USB

- Both USB Host and Device stacks were significantly re-factored and reviewed. Compatibility issues of USB Mass Storage devices which were reported for older versions were reviewed and fixed.
- USB audio examples were reworked and updated to use I2S audio driver and SD card wav file read/write.

- Fixed problem of USB Host Pipe structure list being corrupted in some circumstances when USB device was re-attached.
- The EHCI Pipe private data were not cleared properly when EHCI pipe got free.
- The USB Host stack HighSpeed HUB functionality was fixed

#### Example application modifications and bug fixes

- The mutex example was changed to better demonstrate the mutex features
- Incorrect usage of feof() in hvac demos was corrected. Added condition checking for both EOF and error code to the example application.
- The mfs\_usb example was not de-allocating all resources from the heap when USB stick was unplugged. The memory leak fixed in the example application.
- The ADC potentiometer channel macro fixed in BSPs and is now used consistently in example applications. The macro was fixed to use single-ended mode on the following platforms TWR-K60F120M, TWR-K70F120M and TWR-K53N51.

#### Other bug fixes

- Kinetis ADC driver was updated: ADC\_MAX\_SW\_CHANNELS was modified, adc\_max\_frq\_table[] limit values were corrected to 12MHz at 16bit resolution and to 18MHz for lower resolution modes.
- TWR-PXN20 BSP FlashX driver configuration (block size and with) was corrected to match the processor flash specification.
- The FlashX and FlexNVM functionality has been implemented for TWR-MCF51JF and TWR-K40D100M.
- Kinetis FlashX driver updated. Fixed incorrect functionality in case number of sectors was not divisible by 8. Fixed chip erase IOCTL command. Corrected problem in io\_flashx\_find\_correct\_sectors() in case a flashx\_file spreads over more than one flashx\_block.
- Several ADC driver fixes made on Kinetis platform : Accumulate functionality was modified, the ADC\_VALUE\_TYPE changed to 32 bits, the GPIO init function for TWR-K60F100M and TWR-K70F100M corrected.
- The LWGPIO driver was fixed to correctly handle PORTTH pins on MCF52259 processor.
- The functionality of \_mem\_alloc\_at() function was fixed. The issue appeared when this API was used more than once in the application code.
- The \_task\_disable\_fp(), \_task\_enable\_fp() API functions were implemented for Kinetis architecture.
- Fixed race condition in the \_lwmem\_alloc\_internal() which could lead to the incorrect memory allocation in some circumstances.
- Changed size of FLASH memory from 512kB to 1MB in linker command files for TWR-K60F120M board. Changed initialization struct for FlashX driver from 2 banks to 4 banks.
- RAM vectors table feature was fixed on all Kinetis platforms for ARM Keil µVision IDE.
- The BSS section definition was fixed in the cw10 linker command file for twrk60n512. The .bss and .sbss (zero initialized data) was located incorrectly in the main\_application\_data section resulting in larger flash footprint.



- The function `nfc_write_page()` did not return `NANDFLASHERR_TIMEOUT` unless the time out occurs for all pages being written. The behavior was corrected to be able to detect the error on any page being written.
- Size of ROM vector table has been corrected from 0x410 to 0x400 in the linker command file for CodeWarrior 10.2 and TWR-K60F120M BSP.

### Version 4.0.1 (April 19th 2013)

- New architecture ARM Cortex-A5 and new processor family Vybrid support added.
  - MQX kernel written in ARM ASM, C-sources compiled in thumb mode
  - NEON and FPU supported
  - The `mem_copy` function optimized for Cortex-A5 platform using NEON instruction set
  - L1 and L2 cache enabled
- New board support packages added:
  - TWR-VF65GS10\_A5 – support for Cortex-A5 core for dual core Vybrid processor
  - TWR-VF65GS10\_M4 – support for Cortex-M4 core for dual core Vybrid processor
- MQX ported to new compilers and build tools:
  - Code Warrior 10.3 GCC build tools fully supported – IDE projects and makefiles provided
  - ARM Development Studio 5 (DS-5) – IDE projects provided (Vybrid only)
  - Sourcery CodeBench Lite Edition – makefiles provided (Vybrid only)
- Introduced new HW Timer driver – generic interface to various timer modules. It is currently in experimental phase and supports Kinetis and Vybrid PIT HW module only.
- Multicore Communication (MCC) API provided allowing inter core (OS) communication
  - Tested with MQX to MQX, MQX to Linux
- BSP Cloning wizard was updated
  - CW10.3 GCC support
  - New option to export whole MQX source tree for given board
- Extended IPv6 support - 85% of the “IPv6 Ready Logo” Phase-2 tests (Sections 1, 2, 3, and 5) passed.

### RTCS Bug fixes

- Added "`ipconfig <device> release`" shell command.
- Added additional command line parameters (count, hop limit, interval, pattern, and size) for the ping shell function.
- Implemented missing interface release functions: `ipcfg_release_interface()`, `ipcfg_release_device()`.
- Fixed `ENET_shutdown()` not releasing all allocated resources.
- Calculation of time interval was replaced by `RTCS_timer_get_interval()` to avoid the time-rollover issue.

- Replaced IPs from Microsoft private pool (169.254.x.x) which were used in examples by more generic IPs (192.168.1.x).
- Fixed problems with TFTP PUT function.
- Fixed incorrect echo handling in telnet.
- Other minor fixes.

#### USB Bug fixes

Fixed the IDT leak issue - the uncached memory allocation for each qTD/ ITD/ SITD was added to the code.

- Fixed USB Host ISO transfer buffer to be aligned to 4K boundary.

Added missing inf files for both virtual nic and virtual com example applications.

#### MFS Bug fixes

- Fixed - MFS rename functionality. The rename of file to another directory required filename to be the same.

#### MQX Bug fixes

- DSPI driver - last bit always 1 in SPI MODE1/3 - ASC field of CTAR register set to correspond with half period of SCK.
- Fixed FlashX driver access to the sector at the top of addressable space (i.e. at 0xffff\_ffff).
- Fix in shell\_write function to ensure proper reporting of errors when a file larger than the disk is written.
- Fixed fread(), zero length parameter read; behavior changed to match other implementations (Linux).
- The lwmsgq\_send function now checks if it is running in ISR. Input blocking flags then cause return error MQX\_CANNOT\_CALL\_FUNCTION\_FROM\_ISR.
- Fixed multiple PDB setup for ADC channels with different base frequency.
- Added extern "C" to nvic.h file.
- Fixed FlashX driver incorrectly handling file open/close which was leading to memory leak.
- FlashX driver - added NULL pointer checks to fttl\_init\_ram\_function() and fttl\_flash\_init().
- Fixed cache flush/invalidate functionality when multiple lines for unaligned blocks of data are used.

#### Version 4.0.2

See Chapter 2 ["What is New"](#).