



Freescale Enablement Solutions

Freescale MQX™ Software Solutions

Enabling embedded systems—
Accelerating success



Freescale streamlines embedded design with a complimentary real-time operating system and software stacks

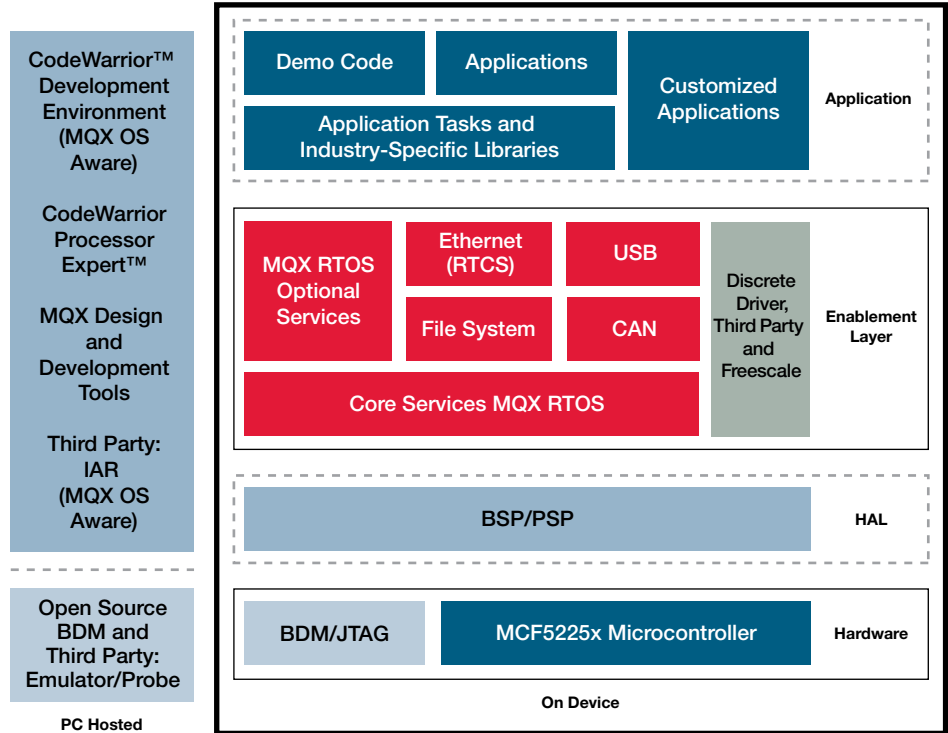
The increasing complexity of industrial applications and expanding functionality of semiconductors are driving embedded developers toward solutions that combine proven hardware and software platforms. To help accelerate time to market and improve application development success, Freescale Semiconductor is offering the Freescale MQX real-time operating system (RTOS) with TCP/IP and USB software stacks to ColdFire® microcontroller (MCU) customers at no additional charge. The combination of Freescale MQX software solutions and silicon portfolio creates a comprehensive source for hardware, software, tools and services.

Freescale plans to expand the availability of this complimentary, integrated, enablement software to include many embedded processors in its broad portfolio. The first line of Freescale embedded controllers to feature Freescale MQX software solutions is the MCF5225x family of ColdFire MCUs. Now shipping, this family of MCUs comes complete with RTOS, TCP/IP stacks and USB stacks—all features of the Freescale MQX software solution.

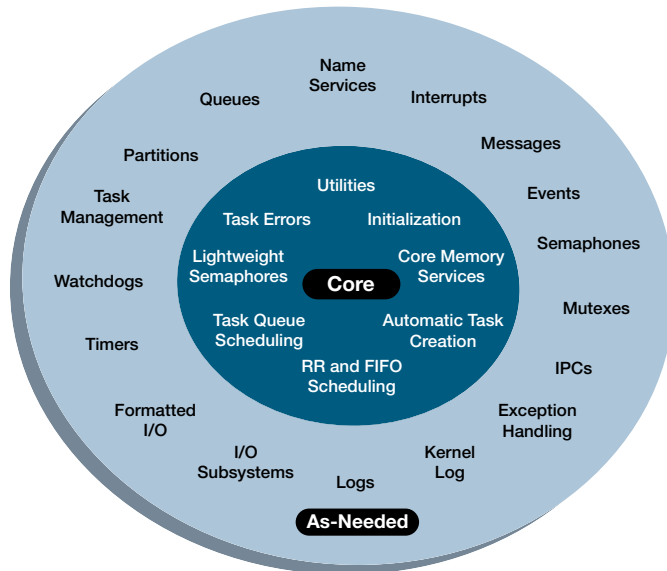
Reducing cost, accelerating success

By providing complimentary Freescale MQX software solutions with its silicon products, Freescale is helping alleviate much of the initial software investment hurdle faced by embedded developers. Comparable full-featured software offerings on the market today may cost developers as much as \$95,000 (USD) in licensing fees.

Freescale Comprehensive Solution



MQX RTOS—Customizable Component Set



According to recent research, development teams spend approximately 60 percent of their resources on software. Embedded projects based on 32-bit devices have a greater need for software reuse to manage development costs. The Freescale MQX RTOS and software stacks address these developer needs by providing a scalable, reusable platform that works across a wide range of Freescale processor architectures, development tools and third-party software environments.

Freescale silicon ships with full production Freescale MQX source code, including communications software stacks at no additional cost. Freescale is able to provide customers a commercial-friendly software licensing model, enabling developers to keep their source modifications while being able to distribute the required binary code.

Full featured, proven and scalable

The MQX RTOS has been the backbone of embedded products based on Freescale silicon for more than 15 years. MQX software deployment spans a broad range of market segments and leading manufacturers worldwide.

The Freescale MQX RTOS offers leading-edge software technology for embedded designs based on Freescale processors and MCUs. The Freescale MQX RTOS offers a straightforward application programming interface (API) with a modular, component-based architecture that makes it simple to fine-tune custom applications. It also allows developers to add Web servers, e-mail, network management, security and routing to their designs. Components are linked in only if needed, preventing unused functions from bloating the memory footprint. By leveraging Freescale's strong network of partners, Freescale MQX software solutions easily scale across third-party software and tools such as security, industrial protocols and graphical plug-ins.

The Freescale MQX RTOS offers powerful, preemptive real-time performance with optimized context switch and interrupt times, enabling fast, highly predictable response times. Its small, configurable size conserves memory space for embedded applications, and it can be configured to take as little as 6 KB of ROM, including kernel, interrupts, semaphores, queues and memory manager.

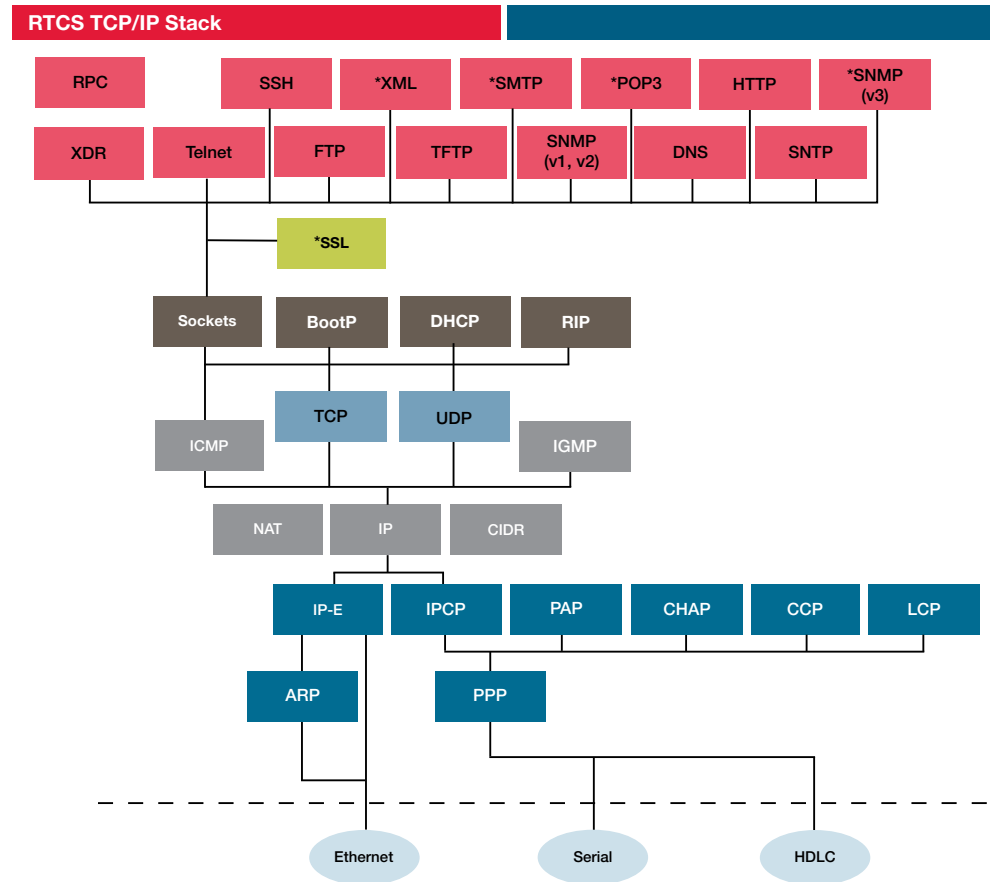
Freescale MQX Real-Time TCP/IP Communication Suite (RTCS)

The Freescale MQX RTCS is a fast, efficient and low-footprint embedded Internet stack that supports a rich set of standard TCP/IP protocols — perfect for Internet-enabled, ColdFire-based devices. It comes complete with a number of application layer protocols such as Telnet, FTP, TFTP, SNMP v1 and SNMP v2. A number of optional, pre-integrated protocols and products are also available from third parties.

The Freescale MQX RTCS is scalable, allowing you to easily define the feature set you want to accommodate your ROM and RAM memory budgets.

Freescale MQX design and development tools

The Freescale MQX RTOS task-aware debugging features plug into either the CodeWarrior™ or IAR debugger, creating a powerful embedded debugging environment, helping to speed development, reduce cost and improve finished product quality. Ready-to-run project files and project stationery are available enabling developers to quickly create their own Freescale MQX platform-based applications. Stack utilization, CPU utilization, resource status (memory, messages and semaphores) and TCP/IP socket information are just a few features at your fingertips.



*Denotes optional products from Embedded Access (www.embedded-access.com)

Application Presentation Session Transport Network Data Link Physical

Supported by Embedded Access, Inc.

Embedded Access, Inc. (EAI) is the world leader in MQX and has over 30 years of hands-on experience with MQX development and support. Freescale has partnered with EAI to provide premium support services. EAI provides embedded software platforms for Freescale ColdFire MCUs, Power Architecture® and i.MX processors and is a provider of add-on components to the Freescale MQX platform such as security products, industrial protocols, flash file systems, e-mail clients, SNMPv3 and others.

In addition to offering engineering services to speed device development based on the Freescale MQX platform, EAI provides training, technical support and software product licensing.

Visit www.embedded-access.com for additional information.

Features	Benefits
Freescale MQX RTOS	
Small code density	<ul style="list-style-type: none"> Context switch and low-level interrupt routines hand-optimized in assembly Can be configured to a memory footprint of 12 KB ROM and 2.5K RAM on CFV2, including kernel, task applications, LW Semaphore, interrupt stack, queues and memory manager
Component-based architecture	<ul style="list-style-type: none"> 25 components—eight core, 17 optional Components are linked in only if needed, preventing unused functions from bloating the memory footprint
Full and lite services	<ul style="list-style-type: none"> Further control of size, RAM/ROM utilization and performance options
Real-time, priority-based preemptive, multithreading	<ul style="list-style-type: none"> Threads execute in order of their priority Allows high-priority threads to meet their deadlines consistently, no matter how many other threads are competing for CPU time
Optimized for Freescale architecture	<ul style="list-style-type: none"> Optimized assembly code to accelerate key real-time portions of the RTOS like context switching
Faster development time	<ul style="list-style-type: none"> Provides the developer faster development time by relieving engineers from creating an efficient scheduling system and interrupt handling Use of multiple communication protocols like USB or TCP/IP
Code reuse	<ul style="list-style-type: none"> Provides a framework with a simple API to build and organize the features across Freescale's broad portfolio of embedded processors
Intuitive API	<ul style="list-style-type: none"> Writing code for MQX is straight forward with a complete API and available reference documentation
Fast boot sequence	<ul style="list-style-type: none"> Ensures the application is running fast after the hardware has been reset
Simple message passing between processors	<ul style="list-style-type: none"> Messages can be either from a system/private pool and sent with either an urgent status, or a user-defined priority, and can be broadcast or task specific For maximum flexibility, a receiving task can be operating on either the same CPU as the sending task or on a different CPU within the same system
Freescale MQX Real-Time TCP/IP Communication Suite (RTCS)	
Designed for embedded applications	<ul style="list-style-type: none"> Specifically designed for adding TCP/IP connectivity to embedded systems Provides fully compliant feature set of networking stacks and configurable enough to fit into the small memory confines of an embedded devices Tightly integrated with Freescale MQX RTOS device drivers for Ethernet and other access layers Tested on Freescale embedded architectures
Small configurable memory footprint	<ul style="list-style-type: none"> Implemented as a C library Allows only the features and protocols used by the application to be included in the image Can be configured to take as little as 30 KB of ROM
RTCS protocol support	<ul style="list-style-type: none"> Provided with a large number of standard protocols One product allows real TCP/IP applications without the need to acquire other application-level protocols
Advanced networking protocols for RTCS	<ul style="list-style-type: none"> RTCS can be extended to support additional industry-standard protocols including: security, advanced routing/network access, embedded Web server/e-mail support and network management protocols
Very scalable	<ul style="list-style-type: none"> Customizable suite can meet a wide range of application RAM and ROM requirement by selectively choosing only the necessary protocols for your design
Full featured	<ul style="list-style-type: none"> Great flexibility in the way you provide connectivity to your device, ranging from simple application such as Ethernet-Serial to complex gateway systems
Support for standard protocols and sockets	<ul style="list-style-type: none"> RTCS not only provides application layer protocols but is a complete OSI model solution that spans data link to application layer standard protocols
Freescale MQX File System (MFS)	
Designed for embedded applications	<ul style="list-style-type: none"> Provides full MS-DOS compatible file system that is configurable to fit into small memory footprint Brings support for desktop PC features such as long file names, multiple disk volumes and directory handling to embedded systems
Portability and Modularity	<ul style="list-style-type: none"> The MFS FAT file system provides a portable, compatible implementation of the MS-DOS file system and library of file system functions File system functions are separated from the device driver functions, allowing for increased modularity Supports different types of storage media Trivial File System (TFS) is a simple read only file system used to avoid the need of MFS in HTTP* Virtual File System (VFS) is used to mount other file systems to one large file system*
Freescale MQX USB Host/Device Stack*	
Designed for embedded applications	<ul style="list-style-type: none"> Specifically designed for adding USB functionality to embedded systems Provides fully compliant USB 1.1 and 2.0 feature set of stacks and drivers
Small configurable memory footprint	<ul style="list-style-type: none"> Designed to fit in a small (<10 KB RAM) and with code size of <32 KB
Supports a variety of class functionality	<ul style="list-style-type: none"> Hub: supports connection of one or more external USB hubs to the system and enables the use of hub-connected devices transparently Mass storage allows flash drives to plug into the host system and to be read by the host's file system and is compatible with the MFS FAT File System HID examples: mouse, keyboard and joystick CDC: virtual serial connection to standard PC driver

*USB Device (Mass Storage and CDC), Trivial File System and Virtual File System coming soon

Learn More: For more information about the Freescale MQX platform, please visit www.freescale.com/mqx.