Quantum® LeaPs
Modern Embedded Software

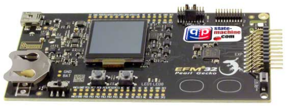
Quantum Leaps, LLC
103 Cobble Ridge Drive
Chapel Hill, NC 27516
state-machine.com

Modern Embedded Systems Programming with QP™ Frameworks and QM™ Modeling Tool

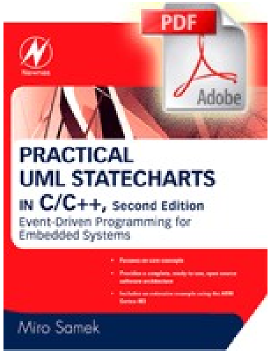
This 3-day training uses hands-on exercises combined with instruction to illustrate the concepts underlying the QP™ active object frameworks and how they are applied in embedded systems programming requiring hard real-time performance. Upon completion of the training, participants have good understanding of event-driven systems, skills in designing and implementing UML state machines manually in C or C++ and graphically with the QM™ modeling tool, and the ability to build their own QP™ applications. Additionally, participants gain understanding of the real-time kernel options built-into the QP frameworks, learn how to reuse behavior with sub-machines, find out how to apply software tracing to debug and optimize their event-driven code and how to apply unit testing to active objects and state machines.

Each attendee gets:

✓ **EFM32-SLSTK3401A** ARM Cortex-M4F development kit



✓ A searchable PDF of the class textbook **Practical UML Statecharts in C/C++**



PDF
Adobe
PRACTICAL UML STATECHARTS IN C/C++, Second Edition
Event-Driven Programming for Embedded Systems
Miro Samek

✓ Training materials download with class notes, PSiCC2 book in PDF, and the latest versions of QP™, QM™ and QTools™

Day 1

Session 1A — Active objects: This session is designed to help engineers through the paradigm shift from the traditional sequential programming to the modern event-driven programming with active objects. The session covers encapsulation for concurrency, inversion of control, blocking versus non-blocking code, run-to-completion (RTC) execution, asynchronous event passing and the role of state machines.

Session 1B — UML state machines: This session explains state machine concepts, the UML notation and the semantics, including hierarchical state nesting, entry/exit actions, internal transitions, etc. The session culminates with the design of a hierarchical state machine in QM followed by automatic code generation, building, and executing the code.

Day 2

Session 2A — QP internals: This session examines the inner workings of the QP framework, which due to the inversion of control can offer benefits that go beyond the traditional RTOS. The session covers direct event passing, publish-subscribe, time events, dynamic event management, and automatic garbage collection for events.

Session 2B — QP application example: This session uses the fun "Fly 'n' Shoot" game to explain all the steps necessary to design and code a real-time QP application.

Session 2C — State design patterns: This session presents four state machine design patterns. Every pattern is illustrated with executable code, which allows attendees to fully understand the pattern and gives them a good starting point for their own implementations.

Day 3 (customizable)

Session 3A — Real-Time Kernels in QP™: This session explains and compares the real-time kernels built into the QP/C and QP/C++ kernels. The kernels covered include: the cooperative QV kernel, the preemptive run-to-completion QK kernel, and the preemptive dual-mode (RTC/blocking) QXK kernel.

Session 3B — Sub-machines: This hands-on session introduces the concepts of sub-machines and sub-machine states. Attendees apply the concepts to the Calculator model introduced earlier, generate code, and build the working application.

Session 3C — Software Tracing: In this session attendees learn how to apply QP/Spy software tracing to debug, monitor, and optimize their event-driven QP applications without stopping or significantly degrading the real-time performance.

Session 3D — Unit Testing: This session presents the **QUTest™** unit testing harness. Attendees learn how to write test fixtures, test doubles, and test scripts to unit test active objects and state machines on the desktop and on embedded targets.

Prerequisites

In order to get the best results participants should have a working knowledge of C, as well as basic knowledge of real-time embedded programming. Previous QP™ or UML experience is not required.

Each attendee needs to bring a laptop running Windows (Windows 10/8/7) with a free USB port.

Instructor

Dan Smith is an embedded software consultant specializing in bringing the modern event-driven paradigm to embedded software teams, who want to adopt hierarchical finite state machines (UML statecharts), real-time frameworks, modeling, and automatic code generation. He has been one of the earliest adopters of the QP active object frameworks and over the years has accumulated many real-life QP projects under his belt. Firmware he developed lies at the heart of many successful products including consumer electronics, telecom/datacom, industrial controls, and medical devices. Dan enjoys teaching and is a frequent speaker at the Embedded Systems Conferences, the Embedded World Conference, and various other venues. He earned his BSEE at Princeton.

Price, Availability, and Contact Information

The training is delivered **on-site** at your location. The price of the 3-day QP™ training in the U.S. is **$15,000.-**. This price includes all training materials for up to 10 attendees (EFM32-SLSTK3401A development kit and the training materials package including the course textbook in PDF) as well as the instructor's time and travel expenses. Each additional attendee beyond the first 10 can be enrolled for **$500.-**. Training **outside the U.S.** can be arranged for additional fee depending on the destination to cover the extra time and travel expenses. Please contact Quantum Leaps to reserve the dates or to request more information about the QP™ Training.