

Green Hills Software Product Training Courses

Get your team started with training

Green Hills Software's real-time operating systems and embedded software development tools offer powerful capabilities to produce totally reliable, absolutely secure, and maximum performance devices—in the least time and at the lowest cost. With expert training from Green Hills Software consultants, you can become productive faster as well as learn how to take full advantage of the power these tools provide.

Customize your training by selecting from several core components and multiple advanced courses.



Combining Core Training with a selection of Advanced Courses provides a thorough immersion in Green Hills Software's products with the flexibility to focus on those concepts of particular interest to students.

Core training classes

MULTI IDE (MUL-100)

This two-day course provides complete exposure to the MULTI® Integrated Development Environment (IDE), teaching students how to fully tap in to all of its powerful capabilities while developing their applications.

The first day covers the basics of using the MULTI IDE. Students will learn how to create, build, and customize MULTI projects as well as how to connect to targets and download and debug applications. Through both lectures and hands-on exercises, students will learn the full extent of the MULTI debugger's capabilities.

The second day covers how to use advanced tools—such as the Profiler and EventAnalyzer—to analyze and tune applications for performance and memory usage. Students will also learn advanced techniques for using Green Hills Software's kernel-aware debugger to quickly find difficult bugs.

Product training courses offered

Core Training

- ▲ MULTI integrated development environment
- ▲ Real-time programming with INTEGRITY
- ▲ INTEGRITY real-time operating system
- ▲ *velOSity* real-time operating system
- ▲ *μ-velOSity* microkernel

Advanced Training

- ▲ MULTI: Custom data visualization and scripting
- ▲ MULTI: Trace and TimeMachine
- ▲ INTEGRITY: Intro to device drivers
- ▲ INTEGRITY: File systems and networking
- ▲ INTEGRITY: Tasks and communication
- ▲ INTEGRITY: Memory management
- ▲ INTEGRITY: Using POSIX
- ▲ INTEGRITY: BSPs and device drivers
- ▲ *μ-velOSity*: File systems and networking

Topics covered

- ▲ Building and debugging applications
- ▲ Advanced Debug Tools (Profiler, EventAnalyzer and Run-time error checking)
- ▲ Customizing the MULTI IDE

Course length: 2 days

Prerequisites: None

Real-Time Programming with INTEGRITY (INT-100)

In this three-day course, students are taught the concepts of a real-time operating system (RTOS) and how the INTEGRITY® RTOS applies them. Developed from the ground-up to be secure and reliable, the INTEGRITY RTOS takes full advantage of a processor's MMU or MPU to provide a fully memory-protected system. Students learn to leverage this advanced technology in their applications.

The INTEGRITY RTOS provides a great degree of flexibility. Consequently, there are often several alternative mechanisms developers can use to accomplish a specific objective, such as intertask communication. This course discusses each mechanism in detail, so that students can understand which method would work best in their application.

We start with sample product requirements, and work through the design and development process to produce a final application. Along the way, students cover all the fundamental concepts behind developing an application for a protected RTOS.

Topics covered

- ▲ Protected RTOS concepts
- ▲ Inter-task communications
- ▲ Tasks and scheduling
- ▲ Memory Management
- ▲ Clocks and alarms

Course length: 3 days

Prerequisites: MUL-100

INTEGRITY RTOS (INT-101)

In this three-day course, students are taught how to apply known RTOS concepts, such as task control and scheduling to the INTEGRITY RTOS. Additionally, students are taught the underlying INTEGRITY data structures and communications mechanisms, including how to incorporate these features into their applications.

Developed from the ground-up to be secure and reliable, the INTEGRITY RTOS takes full advantage of a processor's MMU or MPU to provide a fully memory-protected system. Students learn to leverage this advanced technology in their applications. Advanced concepts, such as sharing memory between applications, are discussed in detail.

The INTEGRITY RTOS provides a great degree of flexibility. Consequently, there are often several alternative mechanisms developers can use to accomplish a specific objective, such as intertask communication. This course discusses each mechanism in detail, so that students can understand which method would work best in their applications.



Topics covered

- ▲ Tasks and scheduling
- ▲ Inter-task communication
- ▲ Memory Management
- ▲ Clocks and alarms
- ▲ Synchronization
- ▲ Sharing objects

Course length: 3 days

Prerequisites: MUL-100 and familiarity with embedded RTOS concepts

velOSity RTOS (VEL-100)

In this two-day course, students are taught how to apply RTOS concepts, such as task control and scheduling to the velOSity™ RTOS. Additionally, students are taught the underlying velOSity data structures and communications mechanisms, including how to incorporate these features in their applications.

The velOSity RTOS provides a great degree of flexibility. Consequently, there are often several alternative mechanisms developers can use to accomplish a specific objective, such as intertask communication. This course discusses each mechanism in detail, so that students can understand which method would work best in their application.

Topics covered

- ▲ Tasks and scheduling
- ▲ Inter-task communications
- ▲ Memory Management
- ▲ Clocks and alarms

Course length: 2 days

Prerequisites: MUL-100

μ-velOSity microkernel (UVL-100)

In this two-day course, students are taught how to apply RTOS concepts, such as task control and scheduling to the μ-velOSity™ microkernel. The μ-velOSity microkernel is designed to be a minimal-footprint microkernel, and special consideration is paid to evaluating size and speed parameters of the various RTOS constructs.

The μ-velOSity microkernel provides a great degree of flexibility. Consequently, there are often several alternative mechanisms developers can use to accomplish a specific objective, such as intertask communication. This course discusses each mechanism in detail, so that students can understand which method would work best in their application.

Topics covered

- ▲ Tasks and scheduling
- ▲ Inter-task communications
- ▲ Resource Management
- ▲ Timers

Course length: 2 days

Prerequisites: MUL-100

Advanced training classes

MULTI: Custom Data Visualization and Scripting (MUL-210)

Students will learn how to use MULTI's custom data visualization tools to examine and analyze their data. Students will also learn how to add custom file processing into the MULTI Builder.

Course length: 1/2 day
Prerequisites: MUL-100

MULTI: Trace and TimeMachine (MUL-300)

The MULTI TimeMachine™ debugging suite offers a wide variety of trace analysis tools that enable embedded software developers to find and fix bugs faster, optimize with ease, and test with confidence.

Students will get full exposure to the TimeMachine suite and explore its ability to find and analyze complex problems.

Course length: 1 day
Prerequisites: MUL-100

INTEGRITY: Intro to Device Drivers (INT-210)

Because the INTEGRITY RTOS is a secure and reliable RTOS, it supports a unique protected device driver model in addition to traditional device drivers. Students learn how to implement both types of INTEGRITY drivers as well as how to integrate custom device drivers into a Board Support Package (BSP).

Course length: 1 day
Prerequisites: MUL-100, INT-100 or INT-101

INTEGRITY: FileSystems and Networking (INT-220):

In this course, students will learn how to add and configure the INTEGRITY File System services and GHNet2 network stack. Topics covered include tuning the modules for speed or size requirements, and using special features of the modules, such as the host_io filesystem, for decreasing development time.

Course length: 1/2 day
Prerequisites: MUL-100, INT-100 or INT-101

INTEGRITY: Tasks and Communication (INT-221)

Expanding on the basics covered in the Core modules, this course shows students how to take advantage of several inter-task communication mechanisms built into INTEGRITY. Topics such as TaskKeys, EventFlags, and advanced use of the ResourceManager are covered.

Course length: 1/2 day
Prerequisites: MUL-100, INT-100 or INT-101

INTEGRITY: Memory Management (INT-222)

Expanding on the basics covered in the Core modules, this course show students how to take advantage of several advanced memory management tools provided with INTEGRITY. Topics such as MallocPools, Shared Libraries, and using the Free Page List are covered.

Course length: 1/2 day
Prerequisites: MUL-100, INT-100 or INT-101

INTEGRITY: Using POSIX (INT-223)

The INTEGRITY RTOS provides complete conformance with the IEEE 1003.1TM-2003 (POSIX®.1) System Interfaces. This course will review the POSIX interfaces and how POSIX threads interoperate with INTEGRITY tasks. The POSIX section covers thread control and scheduling, mutexes and semaphores, and shared memory. At the end of this two-day section, students will have been exposed to all core INTEGRITY concepts. They will be able to apply these concepts to their applications to produce a maximum performance, secure and reliable system that takes full advantage of the flexibility of the INTEGRITY RTOS.

Course length: 1/2 day
Prerequisites: MUL-100, INT-100 or INT-101

INTEGRITY: BSPs and Device Drivers (INT-300)

This is a four-day course, with exercises for each section of the course. Students will be exposed to the complete process from bringing up a new/untested board to booting from an INTEGRITY Application burned into flash.

We start with an overview of INTEGRITY's startup sequence, and then explore each step of the process in detail. At the beginning of the course, students are given a template of a BSP. Working with provided hardware, the students will construct the pieces needed for the BSP. By the end of the course, students will have created a complete working BSP.

Green Hills Software Product Training Courses

While BSPs can be different for different processors and peripherals, the course aims to educate students on the framework of constructing a BSP. Students will be able to take the framework shown in class, and extend this knowledge for their custom boards. Students will understand what sections of the BSP are provided by Green Hills, and where they need to make customizations for specific boards.

Topics covered

- ▲ How to reserve memory for device drivers and when memory is available for kernel and application code
- ▲ How INTEGRITY handles interrupts with minimum latency, and how to write device drivers to take advantage of this feature
- ▲ How to connect and configure Green Hills' hardware probes for a new board
- ▲ How to add timers and clocks to an INTEGRITY system

Course length: 4 days

Prerequisites: MUL-100, INT-100 or INT-101

μ -ve/OSity: FileSystems and Networking (UVL-220)

In this course, students will learn how to add and configure the μ -File System services and GHNet2 network stack. Includes tuning the modules for speed or size requirements.

Course length: 1/2 day

Prerequisites: MUL-100 and UVL-100

Suggested curriculums

Green Hills Training can be customized to meet your training needs. Below are a few suggested curriculums:

- ▲ For students using INTEGRITY who are new to embedded programming
 - MULTI IDE (MUL-100) 2 days
 - Real-Time Programming with INTEGRITY (INT-101) 3 days
- ▲ For students using INTEGRITY who are familiar with embedded programming
 - MULTI IDE (MUL-100) 2 days
 - INTEGRITY RTOS (INT-100) 2 days
 - Advanced courses, MUL-2xx or INT-2xx 1 day
- ▲ For students using INTEGRITY or ve/OSity who are developing BSPs:
 - INTEGRITY: BSPs and DeviceDrivers (INT-300) 4 days

Note: also requires one of the above curriculums



30 West Sola Street ▲ Santa Barbara, CA 93101 ▲ ph. 805.965.6044 ▲ fax 805.965.6343 ▲ www.ghs.com ▲ sales@ghs.com