



Embedded Software Boot Camp

Netrino's Embedded Software Boot Camp is a one-week skills strengthening program consisting of a series of lecture and hands-on courses. The set of courses is designed to quickly and dramatically raise embedded software programming skills. The ratio of lecture to hands-on time can be adjusted. This and other options allow for a significant amount of customization.

"Netrino training courses should be required before writing any embedded code."

— Senior Software Engineer
GENERAL DYNAMICS



Hands-on Programming

Netrino, LLC
6030 Marshalee Drive, #355
Elkridge, Maryland 21075
866.78.EMBED

www.netrino.com
embed@netrino.com

Mon	<h3>Hardware Interfacing in C (Part 1)</h3> <p>The development of device drivers and embedded software is full of challenges. But it is possible to write easy-to-read portable C code to control peripherals ranging from simple timers and UARTs to complex custom FPGAs.</p>
Tue	<h3>Hardware Interfacing in C (Part 2)</h3> <p>The hands-on course continues, moving on from peripheral control via memory-mapped I/O, struct overlays, bitmasks and bitfields, to more advanced topics. The capstone exercise involves an interrupt-based device driver.</p>
Wed	<h3>Multithreaded Programming (Part 1)</h3> <p>There is a lot of misinformation about multithreaded programming and the use of real-time operating system (RTOS) features. Misinformation and programmer inexperience often lead to poor task partitioning and can cause system failures such as task starvation, deadlock, and priority inversion.</p>
Thu	<h3>Multithreaded Programming (Part 2)</h3> <p>The second half of this course demonstrates the proper use of RTOS primitives such as semaphores, mailboxes/message queues, and event flags to develop multithreaded software. A series of hands-on exercises culminates in the design of a multithreaded real-time system with deadlines.</p>
Fri	<h3>RTOS Alternatives (optional)</h3> <p>Though there are good reasons to use an RTOS in certain applications, this decision also creates a number of problems for embedded software developers. This course examines the tradeoffs as well as a set of alternative approaches to multitasking that do not require a preemptive scheduler.</p>