Course Title: C Programming Techniques for Embedded Systems

Course Length: 3 Days

Time Online: N/A

Time in Class: 8 hours (Lab and Class) per day

Time in Lab: Varies by Day

Class Size: Minimum 10 / Maximum 15

Price Per Student: $290.00*

Location: Company Site

Course Description:

C Programming Techniques for Embedded Systems is designed to give attendees the real-world techniques and processes that they need to quickly get up to speed and be productive in a development environment using C. The course goes beyond the syntax by leveraging real-world examples and experiences in addition to providing design patterns that can be easily inserted into any application back at the office.

Lab Projects Description:
Each session has a lab associated with it that exercises the concepts learned during that session. The labs explore real world implementation techniques for using the C programming language in an embedded system.

Course Learning Objectives:
Attendees can expect to walk away from this course with not only the theoretical knowledge on how to use C to develop a product, but also hands-on experience through a careful pairing of lecture and labs designed to give developers real-world knowledge.
Engineers & Developers will walk away understanding:

- The C programming language and where it fits amongst other languages
- Challenges facing embedded developers and how to manage them during development
- The software development life cycle (SDLC) required to successfully design, implement, test and deploy an embedded product
- The process to develop a scalable and reusable embedded software architecture
- How to structure an embedded application
- Debugging strategies and techniques on modern microcontrollers
- Embedded software toolchains and configuration
- How to manage linker files, the stack, heap, interrupts, exceptions and other concepts

* Price based on minimum enrollment, subject to change

- Programming techniques and design patterns such as communication protocol design, parsing commands and messages and implementing state machines
- A review of advanced topics such as function pointers, scheduling techniques and policies, watchdog strategies, assertions and real-time operating systems