Embedded Control and Monitoring Using LabVIEW Course

The Embedded Control and Monitoring Using LabVIEW Course provides hands-on training for designing, prototyping, and deploying a reliable embedded control and monitoring application. At the end of the course, you will be able to translate your embedded system requirements into a scalable software architecture, choose appropriate methods for inter-process and network-based communication, design your real-time application for reliability, and efficiently deploy and replicate your embedded system.

Course Details:

DURATION

- Instructor-led Classroom: Five (5) Days

AUDIENCE

- Users preparing to develop embedded control and monitoring applications using the LabVIEW Real-Time Module and LabVIEW FPGA Module with CompactRIO, PXI, or multifunction I/O hardware
- Users who need the performance and reliability of real-time (RT) and FPGA hardware targets

PREREQUISITES

- LabVIEW Core 1 Course and LabVIEW Core 2 Course or equivalent experience

NI PRODUCTS USED

- LabVIEW
- LabVIEW FPGA Module
- LabVIEW Real-Time Module
- CompactRIO Controller
- C Series analog input, analog output, temperature input, and digital modules

After attending this course, you will be able to:

- Design, prototype, and deploy an embedded control and monitoring application
- Acquire and generate analog and digital signals, control timing, and implement signal processing on RT and FPGA
- Implement functionality on the FPGA for maximum performance and reliability using the LabVIEW FPGA Module

- Implement additional functionality on the RT target for logging, network communication, system health monitoring, and reliability using the LabVIEW Real-Time Module
- Implement a human machine interface (HMI) on the PC
- Communicate data between the FPGA, RT target, and PC
- Debug, benchmark, and test your application
- Deploy your application to multiple systems

Embedded Control and Monitoring Using LabVIEW Course Outline

Lesson	Overview	Topics
Introduction to Embedded Control and Monitoring	This lesson introduces embedded control and monitoring systems using LabVIEW. You learn about the FPGA, real-time, and HMI components of the system.	▪ Embedded control and monitoring system overview ▪ FPGA ▪ Real-time processor ▪ HMI ▪ Example applications
Configuring Your Hardware	This lesson describes how to setup real-time hardware and software. You receive hands-on experience in configuring a CompactRIO RT system.	▪ Setup hardware and PC ▪ Configure RT system settings and software ▪ Configure network settings ▪ Configure RT target from web browser
Identifying Application Requirements	This lesson walks through a variety of considerations when designing and developing an embedded control and monitoring application.	▪ Identify I/O and I/O rate requirements ▪ Identify processes ▪ Identify process timing ▪ Identify data transfer types ▪ Identify performance and reliability requirements

Documenting Your Design	This lesson describes how to use and create different types of diagrams to document your system design	▪ Overview of diagrams ▪ Creating a communication diagram ▪ Typical embedded control & monitoring diagrams ▪ Additional documentation
Accessing Your I/O in LabVIEW	This lesson describes how to access the inputs and outputs of your RT target in your application	▪ Configuring RT targets through the LabVIEW Project ▪ Accessing I/O using driver APIs or Scan Engine ▪ Accessing I/O from FPGA
Programming Using LabVIEW FPGA	In this lesson, you learn how to program the FPGA using the LabVIEW FPGA Module. You gain a high-level understanding of how logic is implemented on the FPGA and how LabVIEW code is translated and compiled into FPGA hardware. After you develop an FPGA VI, you test, debug, compile and then execute on an FPGA target. You examine different reports generated during compilation and learn techniques to optimize your code for size.	▪ Developing the FPGA VI ▪ Simulating the FPGA VI ▪ Compiling the FPGA VI ▪ Basic optimizations
Using FPGA I/O and Timing	In this lesson, you learn how to add FPGA I/O to your LabVIEW project and access it on the block diagram using FPGA I/O Nodes. You also learn how to set FPGA loop rates, add delays between	▪ Using FPGA I/O ▪ Handling FPGA I/O errors ▪ Implementing loop execution rates ▪ Synchronizing C Series Multifunction I/O Modules

	events, and benchmark your FPGA code.	▪ Creating delays between events ▪ Measuring time between events ▪ Benchmarking loop periods
Signal Processing	In this lesson, you learn a variety of ways to process your signals in an FPGA VI.	▪ Using fixed-point data types ▪ Using single-precision floating-point ▪ Performing FPGA math and analysis ▪ Integrating third-party intellectual property (IP)
Inter-Process Communication in FPGA	In this lesson, you learn how to transfer data between multiple loops on your FPGA VI. You examine data sharing methods including variables, memory items, register items, and FPGA FIFOs. You learn the benefits of each technique and when each should be used.	▪ Transferring latest data (tag) ▪ Transferring buffered data (stream, message) ▪ Comparing data sharing methods
Communicating Between the FPGA and RT	In this lesson, you learn how to transfer data between your FPGA VI and RT VI.	▪ Programmatically communicating with the FPGA from RT ▪ Deploying an FPGA VI ▪ Transferring latest data (tag) ▪ Transferring buffered data (stream, message) ▪ Synchronizing the host VI and FPGA VI ▪ Implementing an FPGA watchdog

Optimizing FPGA Code	In this lesson, you learn how to use techniques to optimize for speed or FPGA size if necessary	▪ When should you optimize? ▪ Optimization techniques for FPGA size ▪ Optimization techniques for FPGA speed/throughput ▪ Executing code in Single-Cycle Timed Loops (SCTL) ▪ Pipelining ▪ Four-wire handshaking ▪ LabVIEW FPGA next steps
Programming Using LabVIEW Real-Time	This lesson describes how to design the real-time VI. You learn about the setting priorities and improve the determinism of an application.	▪ Sharing data between deterministic and non-deterministic processes ▪ Sharing data between non-deterministic processes.
Inter-Process Communication in RT	This lesson describes how to transfer data between multiple loops on your RT VI.	▪ Sharing data between deterministic and non-deterministic processes ▪ Sharing data between non-deterministic processes.
Communicating Between RT Target and PC	This lesson describes how to communicate between the RT target and PC. You also learn about various communication techniques such as network-published shared variables, network streams, TCP, UDP, and reference libraries.	▪ Implementing network communication ▪ Transferring latest values (tag) ▪ Transferring buffered values (stream, message)

Managing Memory and Monitoring System Health	This lesson discusses techniques to manage memory properly and monitor the health of the embedded system.	▪ Impacts of memory usage ▪ Memory management ▪ System monitoring
Reliability	This lesson discusses techniques to improve the reliability of the real-time application.	▪ Safe shutdown ▪ Specific and central error handling ▪ Implementing a watchdog ▪ Redundancy
Debugging, Benchmarking, and Testing	This lesson discusses methods of debugging, benchmarking, and testing your real-time application.	▪ Debugging tools ▪ Benchmarking performance and code duration ▪ Testing a real-time application
Deployment and Replication	This lesson discusses how to deploy a stand-alone RT application and replicate it on multiple RT targets.	▪ Introduction to RT deployment ▪ Communicating with deployed applications ▪ Replicating RT systems

Suggested Next Courses

- LabVIEW Core 3 Course
- Advanced Architectures in LabVIEW Course