Certified LabVIEW Embedded Systems Developer (CLED) Certification and Exam Preparation Guide

CLED Overview

The National Instruments Certified LabVIEW Embedded Systems Developer (CLED) is an expert level exam designed to distinguish LabVIEW certified professionals experienced in embedded control and monitoring applications.

A CLED demonstrates proficiency and experience in analyzing requirements, designing, developing, debugging and deploying mission critical, medium-to-large scale monitoring and control applications by efficiently using the NI CompactRIO/Single-Board RIO, LabVIEW Real-Time and FPGA platforms in accordance with LabVIEW Development Guidelines and recommended LabVIEW Real-Time and FPGA best practices.

The Certified Embedded LabVIEW Developer demonstrates proficiency and experience in these key areas:

- Utilize software engineering principles to design modular, scalable, maintainable and well documented software
- Apply best practice guidelines, in all phases of the project, as communicated in related National Instruments training and documentation
- Analyze, interpret and translate customer specifications to project requirements
- Select appropriate hardware and configure a project-specific system hardware architecture
- Utilize software architectures, reference designs, and communication mechanisms to engineer reliable mission critical systems
- Apply software tools and technologies to prototype, develop, optimize, debug and test software applications
- Develop strategies and implement processes to deploy and maintain software on deployed execution systems

CLED Exam Eligibility Criteria

Candidates must pass these eligibility criteria before registering for the CLED exam:

- Must have a valid CLD or CLA certification.

CLED Exam Preparation Resources

The following courses are suggested::

- LabVIEW Real-Time 1
- LabVIEW FPGA
- LabVIEW Real-Time 2
- NI RIO Integrator's training
- Embedded Control and Monitoring using LabVIEW

CLED Exam Preparation: CLED Preparation E-Kit (includes links to preparation guide and sample exam).

Candidates must be familiar and must be proficient at applying the best practices and guidelines in the following documentation:

- LabVIEW Development guidelines
- cRIO Developer's guide
- Large application development best practices

CLED Overview

The CLED certification consists of two proctored exams:

- Part 1: One hour, 30 multiple choice questions, computer based.
- Part 2: Five hours, practical (application development) exam.

  * Passing grade for both exams is 70%.

Note:

- Candidates must pass Part 1 to be eligible to attempt Part 2.
- No certificate will be given on passing Part 1.
- Certificate, logos and shirt will be given on passing Part 2.

CLED Logistics

Part 1: Multiple Choice Exam

- Part 1 of the exam can be taken on any Windows or Mac computer.
- A National Instrument's proctor must be present at the time of the exam.
- The computer must have a reliable and unrestricted internet connection to connect to the exam server via an internet browser. Compatible browsers include Internet Explorer, Google Chrome and Mozilla Firefox.
- Candidates will be provided with login instructions prior to the exam.
- The online exam system will provide exam results on completion of the exam.

Part 2: Practical Exam

- Part 2 of the exam will require a hardware exam kit provided by National Instruments.
- A paper copy of the exam requirements and a USB stick with a VI containing the front panel of the solution will be included in a sealed exam envelope with exam kit. Candidates must open the sealed envelope in the presence of the proctor at exam time.
- The exam computer will either be provided by National Instruments or a Certified Training Center.
- Candidates may request the proctor to allow a few minutes, before the exam, to customize the LabVIEW environment. The proctor will only hand over the exam when the candidate is ready to begin working on the exam.
- Candidates are permitted to use resources available in LabVIEW, such as the *LabVIEW Help*, examples, templates, and sample projects. *Externally developed VIs or resources are prohibited.*

- A detailed application specification will be provided. The specifications consist of general and technical requirements for the application. <u>Candidates are not permitted to detach the binding staple, copy, or reproduce any section of the exam document. Failure to comply will result in failure.</u>
- Candidates are responsible for transferring the solution to the provided USB memory stick. Candidates must validate the copied solution on the USB stick before sealing back in the envelope and returning it to the proctor.
- If the exam kit was shipped directly to the candidate, the candidate is responsible for shipping the kit back to Training and Certification at National Instruments, on completion of the exam.

NON-DISCLOSURE AGREEMENT (NDA) AND TERMS OF USE FOR NATIONAL INSTRUMENTS EXAMS

- The exam is confidential and is protected by trade secret law. It is made available to the examinee, solely for the purpose of becoming certified in the technical area referenced in the title of exam.
- Candidates are expressly prohibited from disclosing, publishing, reproducing, or transmitting the exam, in whole or in part, in any form or by any means, verbal or written, electronic or mechanical, for any purpose, without the prior express written permission of National Instruments - Training & Certification.
- By beginning work on the exam, the candidate accepts the NDA statement and agrees not to disclose the content of exam.

Topics for Part 1: Multiple Choice Exam

1. LabVIEW Real-Time
2. NI Scan Engine
3. LabVIEW FPGA
4. Data Communication
5. Hardware Synchronization
6. Reliability
7. Test, benchmark and debug applications
8. Deployment
9. Integration with other LabVIEW Modules

Topics for Part 1: Multiple Choice Exam	
Topic	**Subtopic/Detail**
1. LabVIEW Real-Time	a. Thread priorities b. Priority inversion, shared resources, and starvation c. Execution systems and their relation to threads and priority d. VI priority versus timed loop priority e. OS thread priority f. Analyze application requirements and their relation to priorities g. Error handling and logging h. Multi-core programming
2. NI Scan Engine	a. Apply and select between NI Scan Engine, Hybrid Mode, or LV FPGA Mode b. Understand scan engine timing consideration c. Handle scan engine faults
3. LabVIEW FPGA	a. Emulation mode b. Arbitration c. Buffering techniques for DMA FIFOs d. Fixed-point data type for FPGA operations e. Enable chain f. FPGA optimization for space/size g. FPGA optimization for performance (throughput & SCTL) h. Compile report
4. Data Communications	a. Commands, tags, and streaming b. Best practices for communications with the following: i. Tags ii. Network streams iii. Command/message iv. FPGA interprocess c. TCP and UDP d. UDP multicast and broadcast e. Client-server
5. Hardware Synchronization	a. FPGA – via shared backplane bus b. Clocks synchronization for distributed systems c. Synchronization bottle necks d. 1588, NI Time sync and SMTP protocols

6. Reliability	a. Failure modes, failure states b. Redundancy c. Error logging d. Alarming e. LabVIEW Real-Time Watchdog f. LabVIEW FPGA watchdog (Fail Safe Control Architecture) g. Acknowledgement based reliable communication h. System health monitoring and maintenance i. Types of memory allocation j. Identify components that allocate memory k. Identify which non-application components (DMA, drivers, TCP) affect memory l. Memory fragmentation and its impact on RT targets m. Buffer allocation and it affect on memory n. Behavior of LabVIEW Real-Time when the system runs out of memory o. Coding practices/strategies for working with fixed-size data
7. Test, benchmark and debug applications	a. Test system functional requirements b. Benchmark uptime, throughput, and data rates c. Debug and /or benchmark thread and VI execution, memory allocation, and resource contention using LabVIEW Real-Time Execution Trace Toolkit d. Benchmark memory usage, CPU usage, execution time, throughput, latency, jitter, and FPGA usage e. Interpret a compile report to estimate if a FPGA program will fit on FPGA f. Prepare system for benchmarking by removing unused software components from OS, disabling debugging, and building executable g. Debug / extract benchmarking info from a headless system by using console, syslog and other tools
8. Deployment	a. Create a system image for replication b. Utilize System Config tools for deployment c. Build an EXE and set as startup d. Deploy NI Scan Engine and shared variables settings e. Deploy software and runtime updates f. Deploy updates that are deployed on reboot g. Deploy and replicate touch panels
9. Integration with other LabVIEW Modules	a. Log and display alarm, event and historical trend data with the LabVIEW DSC Module

Overview for Part 2: Practical Exam

For the practical exam, candidates will be provided with an exam hardware kit. The hardware kit will consist of two NI sbRIO-9623's, a power supply and an Ethernet cable. The NI sbRIO will be designated as follows:

- **Simulator:** This NI sbRIO has a startup executable of the plant simulation. A web service running on the Simulator allows selection of the simulator behavior.
- **Controller:** Candidates will connect to the Controller NI sbRIO to download code to monitor and control the simulator FPGA. The Controller's static IP address will be provided on the exam document.

Candidates must not tamper with the exam kit in any way. Failure to comply may result in the exam solution not being graded.

A matrix showing the connection details between the Simulator and Controller will be provided with the exam.

The exam computer must have the following software installed before the exam:

- LabVIEW 2016 Professional Development System or later
- LabVIEW 2016 Real-Time Module or later
- LabVIEW 2016 FPGA Module or later
- NI-CompactRIO 16.0 driver or later

The exam computer must have two connections enabled:

1. To directly connect to the NI sbRIO Controller via a static IP address
2. To connect to the Compile Cloud Server (via the internet) for compiling the FPGA code.
    - Candidates may use their existing NI Cloud Compile service to compile the exam code.

      OR
    - Candidates may sign up for a 30-day evaluation of the NI Cloud service allowing for adequate time in evaluation period to use the service during the exam. Please refer to the Getting Started with the LabVIEW FPGA Compile Cloud Service to create an evaluation account.

      Note: Candidates should not expect to be given time before the exam sign-up for the evaluation, so they must have their login information available and verified before the exam.

The CLED practical exam consists of a total of 100 points, allocated as follows:

1. Functionality:	50 points
2. Design:	30 points
3. Programming style:	15 points
4. Documentation:	5 points

Passing Score: (70%)

Topics for Part 2: Practical Exam

1. LabVIEW software design and development
2. Human Machine Interface design and development
3. LabVIEW Real-Time application design and development
4. LabVIEW FPGA application design and development
5. Network communication
6. Error Handling
7. Configuration data and file logging
8. Failure Modes
9. Determinism
10. Performance
11. Deployment

Topics for Part 2: Practical Exam	
Topic	**Subtopic/Detail**
1. LabVIEW software design and development	a. Develop system design b. Design modular, scalable, maintainable and well documented software c. Apply best practice guidelines communicated in related National Instruments training and documentation in all phases of the project d. Utilize standard debugging tools throughout development e. Install minimal software set to target
2. Human Machine Interface design and development	a. Handle user interface events with minimal latency b. Manage configuration data (recipes, parameters, etc) c. Group data in meaningful date structures d. Display I/O data in a waveform graph e. Display system health data in a waveform graph
3. LabVIEW Real-Time application design and development	a. Design a recipe engine b. Leverage standard design patterns c. Select the most appropriate scalable method for interprocess communication d. Monitor available memory and memory fragmentation e. Develop and implement mechanisms to log alarms and events
4. LabVIEW FPGA application design and development	a. Define and design and control algorithm (IO variables, process variables, set points) b. Implement custom control (PID, hysteresis or interlock type control) c. Use no more than 70% of the FPGA fabric (interpret compile report, optimize if needed) d. Develop data communication between FPGA and RT
5. Network communication	a. Specify and select communications protocols based on application requirements b. Send buffered commands from user interface to RT target with minimum latency c. Incorporate acknowledgements into the command sender framework d. Handle network communication in processes separate from any deterministic or event-based processes e. Send current value data of I/O channels to user interface with minimum overhead

	f. Send current value of RIO system health variables to user interface with minimum overhead g. Manage multiple clients i. Local host plus a supervisory host ii. One active client, and one passive client
6. Error Handling	a. Real-Time application reports, handles and logs at least two classes of errors b. Custom error codes
7. Configuration data and file logging	a. Manage recipe configuration data b. Develop strategy to transfer data from target to host c. Develop file spanning method to minimize data loss d. Manage file logging on target and host
8. Failure Modes	a. Design system with multiple safe states b. FPGA outputs go into a safe state upon watchdog and I/O node errors c. Design ramp down conditions, versus setting outputs to zero d. Reboot system when memory gets low e. Utilize bi-directional RT and FPGA watchdog
9. Determinism	a. Avoid dynamic memory allocations within deterministic processes b. Shared resources within any deterministic processes set to "skip if busy" c. Ensure deterministic loops finish on time
10. Performance	a. Ensure CPU usage stays below 80% b. Monitor and ensure contiguous memory stays relatively constant
11. Deployment	a. Build application into an exe and set as startup