

---

**Certified LabVIEW Developer Examination**

---

Examinee \_\_\_\_\_ Date: \_\_\_\_\_

Administrator \_\_\_\_\_ Date: \_\_\_\_\_

**Instructions:**

If you did not receive this exam in a sealed envelope stamped “NI Certification,” **DO NOT ACCEPT** this exam. Return it to the proctor immediately. You will be provided with a replacement exam.

- **Please do not detach the binding staple of any section. If any part of the exam paper is missing or detached when returned to National Instruments, you will be deemed to have failed the exam.**
- This examination may not be taken from the examination area or reproduced in any way. You may not keep any portion of this exam after you have completed it.
- Please do not ask the proctor for help. If you believe the intent of any part of the exam is not clear, you may make appropriate assumptions. Please document your assumptions either on the question paper or on the LabVIEW block diagram.
- The exam requires you to develop a LabVIEW application based on a set of specifications.
- A computer with a standard installation of LabVIEW is the only reference allowed for the examination. Externally developed code and third party tools are not allowed in the exam.
- You may use LabVIEW design patterns, templates, and examples available in the development environment as a guide/resource for the application development.
- The application must be specifically developed for the exam submission.
- Submit your finished application on the disk provided.
- Total time allocated for the exam: 4 hours
- Exam passing grade: 75%

**Grading:**

The application development exam consists of a total of 40 points which are allocated as follows:

- Programming style (**15 points**)
- Functionality (**15 points**)
- Documentation (**10 points**)

**IMPORTANT:**

- **When you have completed the exam, place the exam document, the disk with the saved application, and any deliverables in the envelope provided.**
- **Please SEAL the envelope.**
- **Give the sealed envelope to your proctor.**

**Application Development**  
**Section I: General Requirements**

The Certified LabVIEW Developer exam tests your ability to develop a LabVIEW application.

The application should:

- Function as specified in Section II of this document.
- Conform to LabVIEW coding style and documentation standards (found in LabVIEW documentation – LabVIEW Development Guidelines).
- Be created expressly for this exam using VIs and functions available in LabVIEW. Templates, examples, or code developed outside the bounds of this exam are not acceptable for use in the application.
- Be hierarchical in nature. All major functions should be performed in subVIs.
- Use a state machine that either uses a type defined enumerated control, queue, or Event structure for state management.
- Be easily scalable to add more states / features without having to manually update the hierarchy.
- Minimize the use of excessive structures, variables (locals / globals) and Property Nodes.
- Respond to front panel controls (within 100 ms) and not utilize 100% of CPU time.
- Close all opened references and handles where used.
- Be well documented and include the following:
  - Labels on appropriate wires within the main VI and subVIs.
  - Descriptions for each algorithm.
  - Documentation in VI Properties » Documentation for both main VI and subVIs.
  - Tip strip and Description for front panel controls and indicators.
  - Labels for constants

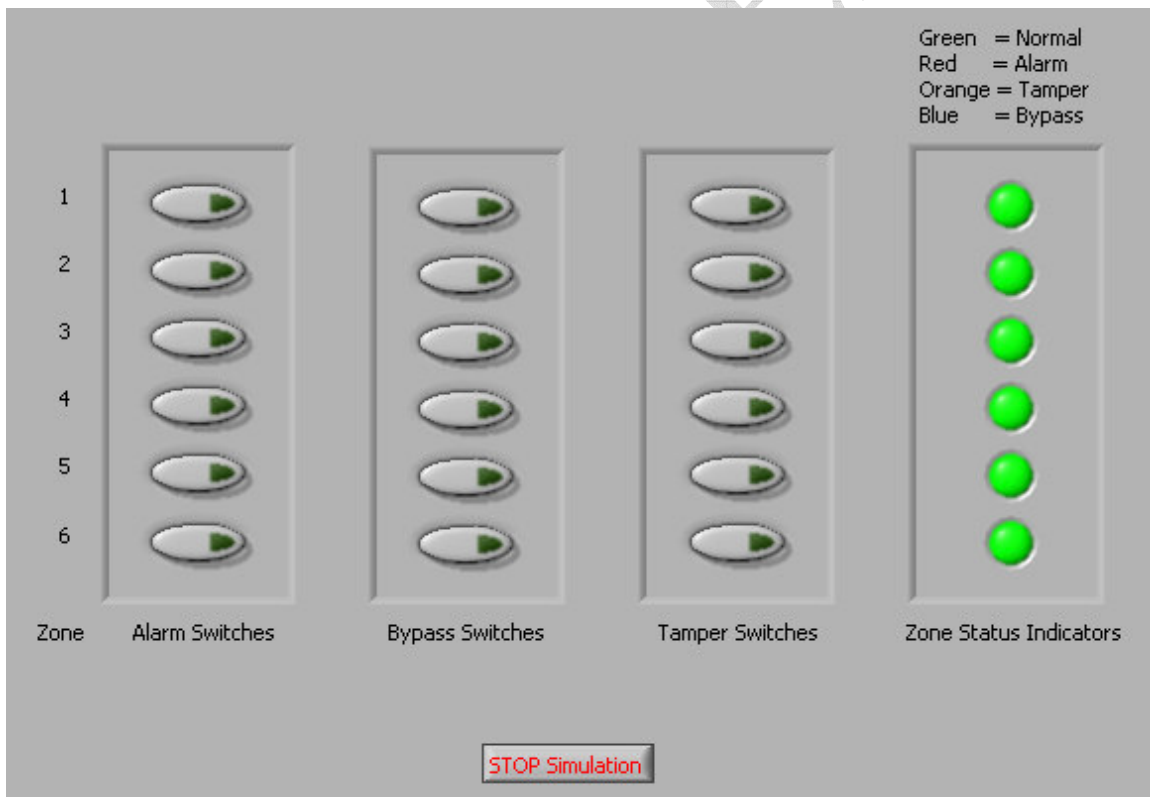
**Application Development**  
**Section II: Application Requirements**  
**Security System**

**Task:**

Your company has received a contract to produce a security system for a local manufacturer. The purpose of the security system is to monitor the perimeter of the manufacturing complex to prevent unauthorized personnel from entering the grounds. Your job is to design the security system using LabVIEW that satisfies the provided specifications.

**Instructions:**

Using a front panel similar to the graphic provided, create a LabVIEW application that implements the actions of the security system. You will create the software for six perimeter zones, with the actual hardware being simulated by Boolean switches on the Front Panel. The application should be easily scalable to add more zones without having to manually update the hierarchy.



**Definitions:**

- Zone: A perimeter area with its own security status indication  
Alarm: A zone condition indicating an intrusion into a zone  
Bypass: A state in which a zone will not indicate an alarm condition. Placing a zone in bypass prevents nuisance alarms when maintenance work is

performed in the area near the perimeter fence, or when severe weather may cause false alarms.

**Tamper:** A condition where the wiring of a zone has been altered in some way. All zone wiring is supervised. That is, if an individual attempts to circumvent the system by altering the wiring, that zone will indicate a tamper condition.

### **Description of Controls/Indicators**

The security system software accepts Boolean inputs, via front panel switches (one set for each zone) for the following functions:

- Alarm input
- Tamper input
- Bypass input

The security system provides for one indicator light for each zone. The color of the light provides the status information for that zone. The colors are:

- Green: Normal
- Red: Alarm
- Blue: Bypass
- Orange: Tamper

### **Operation of the Security System**

#### **System**

- A Bypass input should always override an Alarm input but should not turn off an existing Alarm condition. An Alarm condition should not be indicated while a zone is in a Bypass condition.
- A Tamper input should always override both an Alarm input and/or a Bypass input but should not turn off existing Alarm and/or Bypass conditions. Alarms and Bypass conditions should not be indicated while a zone is in a Tamper condition.
- An existing condition should be indicated when an overriding input is removed.

#### **File Logging**

The security system should produce an ASCII disk file in a spreadsheet-readable format. The data, when imported into a spreadsheet, should be in the following format: (where XXXX represents logged data)

Date	Time	Zone	Status
XXXXXXX	XXXXXXX	XXXXXXX	XXXXXXX
XXXXXXX	XXXXXXX	XXXXXXX	XXXXXXX

The Status field should contain a string indicating the condition of the zone: Normal, Alarm, Bypass or Tamper.

The log file should be stored in a location relative to the location of the security application and should be updated only when the status of a zone changes and closed after every transaction.

SAMPLE EXAM