

# LabVIEW™ Core 3

## 교육과정 매뉴얼

교육과정 소프트웨어 버전 2013  
2013년 10월판  
325510C-0129

### 저작권

© 2004–2013 National Instruments. 판권 소유.

저작권법에 따라, 이 간행물은 National Instruments Corporation의 사전 서면동의없이 간행물의 전부 또는 일부라도 사진 복사, 기록, 정보검색시스템으로 저장, 번역을 할 수 없음은 물론이거니와 전자 또는 기계방식의 여하한 형태로도 복제 또는 송신될 수 없습니다.

National Instruments는 타인의 지적재산권을 존중하며 사용자에게도 그렇게 할 것을 요청하고 있습니다. NI 소프트웨어는 저작권 및 기타 지적재산권법에 의해 보호받고 있습니다. NI 소프트웨어를 타인 소유의 소프트웨어 또는 기타 자료들을 복제하는데 사용할 수 있는 경우, 여러분은 NI 소프트웨어를 라이선스 또는 기타 법적 제한조건에 따라 복제해도 되는 자료들을 복제하는데에만 사용해야 합니다.

사용자 라이선스 협약 (**EULA: End-User License Agreements**) 및 타사 법적 공지

사용자 라이선스 협약 (EULA) 및 타사 법적 공지는 다음의 위치에서 찾을 수 있습니다:

- 법적 공지는 <National Instruments>\\_Legal Information 및 <National Instruments>에 있습니다.
- 사용자 라이선스 협약은 <National Instruments>\Shared\MDF\Legal\license 디렉토리에 있습니다.
- NI 제품으로 빌드된 설치 프로그램에 법적 정보를 포함시키는 방법은 <National Instruments>\\_Legal Information.txt를 참조하십시오.

## 상표

National Instruments 상표에 대한 더 자세한 정보는 [ni.com/trademarks](http://ni.com/trademarks) 에서 *NI Trademarks and Logo Guidelines* 를 참조하십시오 .

ARM, Keil, and  $\mu$ Vision are trademarks or registered of ARM Ltd or its subsidiaries.

LEGO, the LEGO logo, WEDO, and MINDSTORMS are trademarks of the LEGO Group. ©2013 The LEGO Group.

TETRIX by Pitsco is a trademark of Pitsco, Inc.©2013

FIELD BUS FOUNDATION™ and FOUNDATION™ are trademarks of the Fieldbus Foundation.

EtherCAT® is a registered trademark of and licensed by Beckhoff Automation GmbH.

CANopen® is a registered Community Trademark of CAN in Automation e.V.

DeviceNet™ and EtherNet/IP™ are trademarks of ODVA.

Go!, SensorDAQ, and Vernier are registered trademarks of Vernier Software & Technology. Vernier Software & Technology and [vernier.com](http://vernier.com) are trademarks or trade dress.

Xilinx is the registered trademark of Xilinx, Inc.

Tapfite and Trilobular are registered trademarks of Research Engineering & Manufacturing Inc.

FireWire® is the registered trademark of Apple Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Handle Graphics®, MATLAB®, Real-Time Workshop®, Simulink®, Stateflow®, and xPC TargetBox® are registered trademarks, and TargetBox™ and Target Language Compiler™ are trademarks of The MathWorks, Inc.

Tektronix®, Tek, and Tektronix, Enabling Technology are registered trademarks of Tektronix, Inc.

The Bluetooth® word mark is a registered trademark owned by the Bluetooth SIG, Inc.

The ExpressCard™ word mark and logos are owned by PCMCIA and any use of such marks by National Instruments is under license.

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

이 문서에서 언급된 다른 제품과 회사의 이름들은 각각 해당 회사들의 상표이거나 상호들입니다 .

National Instruments Alliance Partner Program 의 멤버들은 National Instruments 와는 다른 독자적인 사업 기구들이며 National Instruments 와 어떠한 대리관계나 파트너십 또는 joint-venture 관계를 가지고 있지 않습니다 .

## 특허권

National Instruments 제품 / 기술에 대한 특허권에 관하여는 귀하의 소프트웨어에 있는 도움말>>특허, 귀하의 미디어에 있는 [patents.txt](http://patents.txt) 파일 또는 [ni.com/patents](http://ni.com/patents) 의 *National Instruments Patent Notice* 를 참조하십시오 .

전세계 기술 지원 및 제품 정보

[ni.com/korea](http://ni.com/korea)

전 세계 현지 사무소

[ni.com/niglobal](http://ni.com/niglobal) 을 방문하여 최신 연락 정보, 지원 전화번호, E-메일주소 및 이벤트 정보를 제공하는 각 사무소의 웹 사이트에 접속할 수 있습니다.

**National Instruments** 본사

11500 North Mopac Expressway Austin, Texas 78759-3504 USA 전화: 512 683 0100

추가적인 지원 정보는, *추가적인 정보와 리소스* 부록을 참조하십시오. National Instruments 문서에 대한 문의사항은 National Instruments 웹 사이트의 [ni.com/info](http://ni.com/info) 에서 정보 코드 feedback 을 입력하십시오.

# 목차

---

## 수강생 안내서

A. NI 자격증 .....	vii
B. 교육과정 설명 .....	vii
C. 시작하기 전의 준비사항 .....	viii
D. 교육과정 소프트웨어 설치하기 .....	viii
E. 교육과정 목표 .....	viii

## 1 과

### 성공적인 어플리케이션 개발하기

A. 확장성이 있고, 이해 및 유지보수가 쉬운 VI .....	1-2
B. 소프트웨어 개발 과정 개요 .....	1-4
C. 요구사항 .....	1-9
D. 작업 분석 .....	1-10

## 2 과

### 프로젝트 구성하기

A. 프로젝트 라이브러리 .....	2-2
B. 프로젝트 탐색기 도구와 구성 .....	2-6
C. 프로젝트 충돌 .....	2-9

## 3 과

### 어플리케이션 구조 생성하기

A. 구조 테스트 .....	3-2
B. LabVIEW 스타일 지침 .....	3-4
C. 사용자 이벤트 .....	3-8
D. 큐 메시지 핸들러 .....	3-10
E. 어플리케이션 데이터 타입 .....	3-13
F. 알림자 (Notifiers) .....	3-15

## 4 과

### 사용자 인터페이스 사용자 정의하기

A. 사용자 인터페이스 스타일 지침 .....	4-2
B. 사용자 인터페이스 원형 .....	4-9
C. 사용자 인터페이스 사용자 정의하기 .....	4-10
D. 사용자 인터페이스 확장하기 .....	4-15
E. 윈도우 모양 .....	4-17
F. 사용자 문서 .....	4-18
G. 사용자 인터페이스 초기화 .....	4-21
H. 사용자 인터페이스 테스트 .....	4-24

## 5 과

### 에러 관리하고 로그하기

A. 에러 테스트하기 .....	5-2
B. 로컬 에러 핸들링 .....	5-2
C. 글로벌 에러 핸들링 .....	5-4
D. 에러 로그하기 .....	5-8

## 6 과

### 모듈러 코드 생성하기

A. 모듈러 어플리케이션 설계하기.....	6-2
B. 코드 모듈 테스트하기.....	6-3
C. 통합 테스트하기.....	6-4

## 부록 A

### 보일러 컨트롤러 요구사항

A. 보일러 기본사항.....	A-1
B. 일반적인 어플리케이션 요구사항.....	A-2
C. 어플리케이션 요구사항.....	A-2

## 부록 B

### 보일러 컨트롤러 사용자 스토리

## 부록 C

### IEEE 요구사항 문서

A. IEEE (Institute of Electrical and Electronic Engineers) 표준.....	C-2
--	-----

## 부록 D

### 추가적인 정보와 리소스

# 수강생 안내서

LabVIEW Core 3 교육과정 키트를 구입해 주셔서 감사드립니다. 교육 과정 매뉴얼 및 첨부된 소프트웨어는 3 일간의 실습 교육과정인 LabVIEW Core 3 에서 사용됩니다.

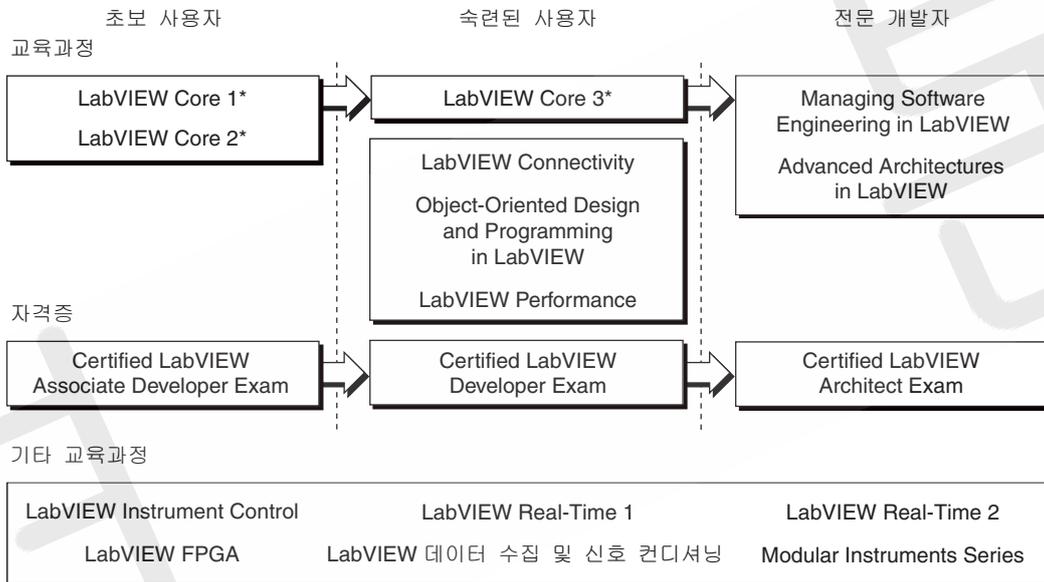
교육과정 키트 구입 후 90 일 이내에 해당 교육과정의 수업을 신청하는 경우, 수강 등록비에서 키트 구매액을 할인해 드립니다. 교육과정 일정, 강의 개요, 교육 센터, 클래스 등록에 관한 정보는 [niedu.co.kr](http://niedu.co.kr) 을 방문하십시오.



**노트** 교육과정과 연습문제 매뉴얼의 업데이트 및 수정 사항을 확인하려면 [nie.com/info](http://nie.com/info) 에서 정보 코드 core3 을 입력하십시오.

## A. NI 자격증

LabVIEW Core 3 과정은 사용자가 LabVIEW 에 능숙해지도록 하며, 또한 NI LabVIEW 자격증 시험 준비에 도움이 되도록 구성된 교육과정 시리즈의 일부입니다. 다음은 LabVIEW 교육 시리즈의 교육과정이 어떻게 구성되는지 보여줍니다. NI 자격증에 대한 더 자세한 정보는 [niedu.co.kr](http://niedu.co.kr) 을 참조하십시오.



\*Core 교육과정은 LabVIEW를 사용할 때 효율성을 극대화하도록 도와주는 필수적인 과정입니다.

## B. 교육과정 설명

LabVIEW Core 3 교육과정을 통해 수강생은 LabVIEW 어플리케이션의 설계, 구현, 문서화 및 테스트를 구조적으로 실습할 수 있습니다. 본 교육과정은 확장성과 가독성 및 유지보수성이 있고 계층구조로 구성된 어플리케이션을 개발하는 방법에 중점을 둡니다. 개발자는 본 교육과정에서 다루어진 과정 및 기법을 사용하여 개발 시간을 단축하고 어플리케이션의 안정성을 향상시킬 수 있습니다. 개발 초기에 이러한 설계 방식을 적용하면 어플리케이션을 불필요하게 다시 디자인해야 하는 상황을 피하고, 기존 VI 를 다시 사용하는 빈도를 늘리며 어플리케이션 유지보수에 드는 비용을 최소화할 수 있습니다.

이 교육과정에서는 수강생이 LabVIEW Core 1 및 LabVIEW Core 2 교육과정을 이수했거나, 또는 그에 준하는 경험을 가졌다고 간주합니다.

이 교육과정 키트는 순서대로 학습해 나가도록 구성되어 있습니다. 본 교육과정 및 연습문제 매뉴얼은 여러 개의 과로 나뉘어져 있습니다.

교육과정 매뉴얼의 각 과는 다음과 같이 구성됩니다 :

- 해당 과의 학습 목표 및 내용을 설명하는 도입부
- 학습 토픽에 대한 설명
- 해당 과에서 학습한 주요 개념 및 기법을 확인하고 복습하는 요약 퀴즈

연습문제 매뉴얼의 각 과는 다음과 같이 구성됩니다 :

- 각 과에서 설명한 토픽에 대한 이해를 돕는 일련의 연습문제
- 시간 여유가 있는 경우 학습할 수 있는 고급 연습문제 또는 추가 연습문제도 일부 포함



**노트** 이 교육과정의 연습문제는 단계적으로 구성되어 있어, 최종적으로 교육과정을 마칠 때에는 하나의 어플리케이션이 완성되도록 꾸며져 있습니다. 한 연습문제를 건너뛰고 그 다음 연습문제를 풀어야 하는 경우에는, <Solutions>\LabVIEW Core 3 디렉토리에서 건너뛴 연습문제에 대한 해답 VI 를 사용하십시오.

## C. 시작하기 전의 준비사항

이 교육과정 매뉴얼을 사용하기 전에 다음 아이템이 준비되어 있는지 확인하십시오.

- Windows XP 또는 이후 버전이 설치된 컴퓨터
- LabVIEW Professional Development System 2012 또는 이후 버전
- 다음의 폴더가 포함된 LabVIEW Core 3 교육과정 CD:

폴더 이름	설명
Exercises	이 교육과정에 사용되는 VI 및 기타 파일이 들어있는 폴더
Solutions	완성된 연습문제가 들어 있는 폴더

## D. 교육과정 소프트웨어 설치하기

다음 단계에 따라 교육과정 소프트웨어를 설치하십시오.

1. 교육과정 CD 를 컴퓨터에 넣습니다.
2. 지시에 따라 교육과정 교재를 설치합니다.

설치 프로그램은 Exercises 와 Solutions 폴더를 루트 디렉토리의 최상위 레벨에 놓습니다. 연습문제 파일은 <Exercises>\LabVIEW Core 3 디렉토리에 위치합니다.



**팁** <Exercises> 처럼 꺾쇠 괄호 사이에 있는 폴더 이름은 사용자 컴퓨터의 루트 디렉토리에 있는 폴더를 나타냅니다.

## E. 교육과정 목표

LabVIEW 개발 프로젝트를 위한 요구사항 문서가 제공되면, 소프트웨어 개발 과정을 따라 가독성, 확장성, 유지보수성 요구사항을 충족시키는 방식으로 핵심 어플리케이션 기능을 설계, 구현, 문서화, 테스트하게 됩니다.

---

## 성공적인 어플리케이션 개발하기

이번 과에서는 여러분이 개발자로서 확장성이 있고, 이해 및 유지보수가 쉬운 VI를 생성할 수 있도록 해주는 다양한 개발 기술을 설명합니다. 여러분은 소프트웨어 개발 과정을 따르는 것의 중요성과 요구사항 문서를 평가하는 방법을 배우게 됩니다.



**노트** 이 교육과정은 LabVIEW 개발자의 관점에서 이러한 토픽에 접근합니다. Managing Software Engineering in LabVIEW 교육과정은 LabVIEW Architect의 관점에서 여러 같은 토픽에 접근합니다.

### 토픽

---

- A. 확장성이 있고, 이해 및 유지보수가 쉬운 VI
- B. 소프트웨어 개발 과정 개요
- C. 요구사항
- D. 작업 분석

본문

컨텐츠

## A. 확장성이 있고 , 이해 및 유지보수가 쉬운 VI

LabVIEW 어플리케이션은 하나의 VI 로 구성된 간단한 어플리케이션에서부터 , 여러 개의 VI 가 복잡한 계층구조로 구성된 확장된 기능의 어플리케이션에 이르기까지 다양합니다 . LabVIEW 의 용도가 확장되고 그로 인해 보다 복잡한 어플리케이션을 생성하게 될수록 , 어플리케이션을 구성하는 코드 도 점점 복잡해지게 됩니다 .

성공적인 LabVIEW 어플리케이션은 확장성이 있고 , 이해 및 유지보수가 쉬운 VI 를 사용합니다 .

- 확장성 -VI 를 완전히 새로 설계하지 않고도 더 많은 업무를 처리할 수 있도록 쉽게 설계를 확장할 수 있음 .
- 가독성 -VI 의 설계를 시각적으로 확인하여 VI 의 목적과 기능을 쉽게 이해할 수 있음 .
- 유지보수성 - 원래 기능에 영향을 미치지 않으면서 쉽게 VI 에 새로운 기능을 추가할 수 있음 .

LabVIEW 로 프로그래밍을 하는 경우에도 , 텍스트 기반 언어로 프로그래밍 작업을 할 때 겪게 되는 것과 같은 설계 문제를 많이 경험할 수 있습니다 . 그러나 LabVIEW 에서는 개발자가 코드 오류 문제 또는 메모리 문제보다는 프로젝트를 위한 솔루션 개발 자체에 더 집중할 수 있도록 해주는 프로그래밍 기법을 제공합니다 .

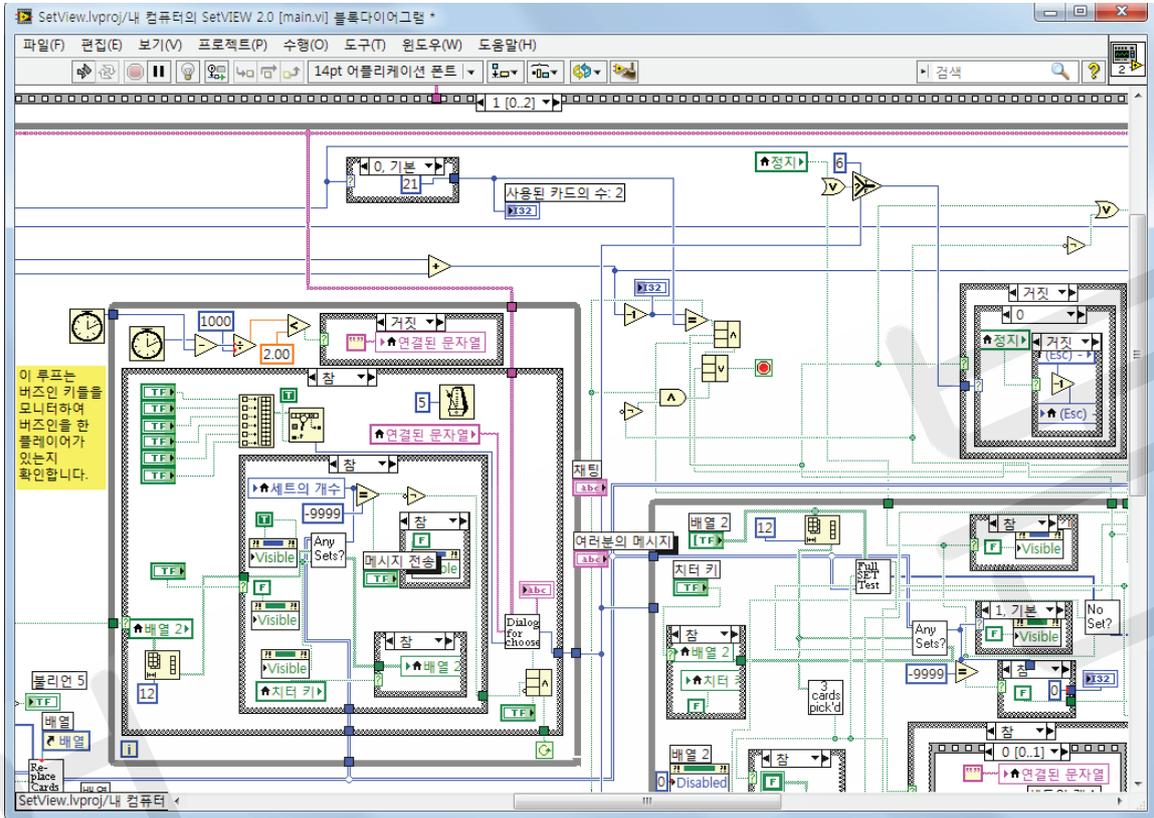
### 확장성이 있는 VI

확장성이 있는 VI 를 만들려면 , 설계 과정 초기에서부터 어플리케이션을 어떻게 만들 것인가에 대해 생각해 보아야 합니다 . 잘 설계된 , 확장성 있는 VI 를 사용하면 기존 설계를 쉽게 확장하여 더 많은 업무를 처리할 수 있습니다 . 예를 들어 , 3 개의 열전쌍에서 데이터를 수집하는 데이터 수집 VI 를 생각해 봅시다 . 어플리케이션의 요구사항이 변경되어 수백 개의 열전쌍에서 데이터를 수집해야 할 경우 , 새로운 VI 를 설계하는 것보다 기존 VI 를 확장하는 것이 더 쉽습니다 . 이렇듯 , 어플리케이션을 설계할 때에는 어플리케이션의 목적 및 사용 범위가 원래 스펙을 벗어나게 될 경우 그러한 변경사항을 어떻게 다루어야 할 지도 고려해야 합니다 .

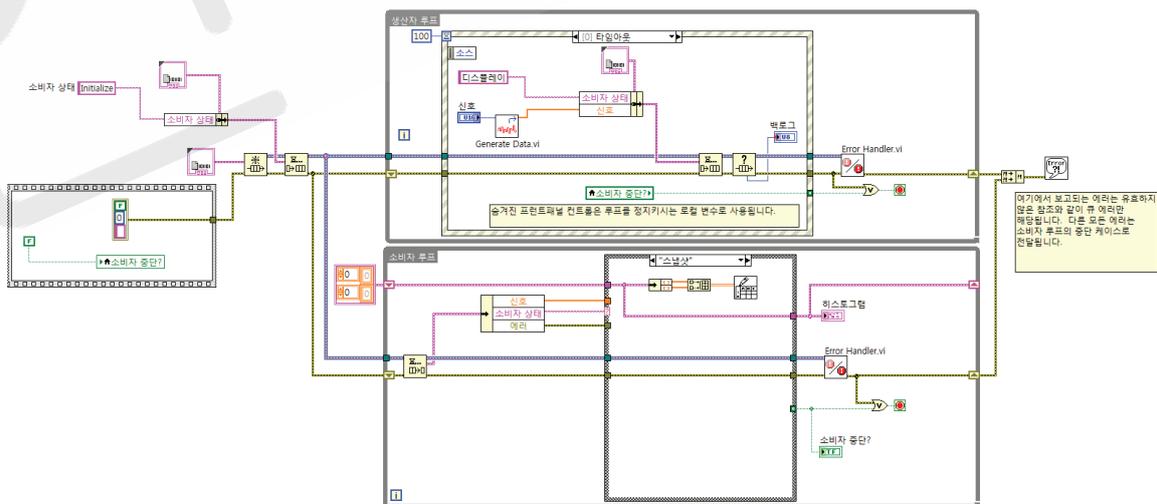
## 가독성이 있는 VI

LabVIEW 를 사용하면서, 올바르게 설계되어 있지 않아서 가독성이 낮고 이해하기 힘든 블록 다이어그램을 본 적이 있을 것입니다. 코드가 혼란스럽고 판독이 어렵다면 블록 다이어그램에서 의도한 기능을 파악하는 것도 어렵습니다. 그림 1-1 는 잘못 설계된 블록 다이어그램 및 올바르게 설계된 블록 다이어그램을 보여줍니다.

그림 1-1. 잘못 설계된 블록 다이어그램 및 올바르게 설계된 블록 다이어그램의 예



1



2

1 잘못 설계된 블록 다이어그램

2 올바르게 설계된 블록 다이어그램



팁 판독 및 이해가 어려운 코드는 향후 유지보수하기도 어렵습니다.