JKI State Machine Online Course
Course Topics and Student Learning

Course Overview

JKI State Machine Basics introduces you to structured practices to create high-quality LabVIEW applications. The processes and techniques covered in this course will help you learn how easy state machines make creating useful real-world applications, use state machines to build a signal plotter and add new features to your applications, use the JKI State Machine to build a sequencer, and discover JKI State Machine best practices to avoid common pitfalls of application development. By incorporating these design practices early in your development, you will create LabVIEW applications faster than ever.

Course Goals and Student Outcomes

After completing this course, students should be able to **create real-world test, measurement, and control applications** using the JKI State Machine.

Specifically, after completing this course, students should be able to:

1. Determine when to use a state machine
2. Differentiate between a JKI State Machine and a "traditional" state machine
3. Explain the value of the JKI State Machine
4. Describe how a JKI State Machine works
5. Apply JKI State Machines best practices
6. Compose manageable and scalable code using the JKI SM
7. Create a real-world application in LabVIEW

Prerequisites

Basic LabVIEW familiarity is all that's needed.

Students should have a basic familiarity with LabVIEW (understanding of While Loops, Clusters, Arrays, Strings, and User Interface Events) -- the course does not require much LabVIEW experience.

And, it's great for advanced LabVIEW developers, too!

Also, this course is extremely useful for advanced LabVIEW developers, too. We've had students take our course after using LabVIEW for 10 years and get a LOT out of the course (it really impressed upon them, the value of using the JKI State Machine in their applications).

JKI State Machine Basics Course Outline

Module	Overview	Lessons and Exercises
1. Building a Signal Plotter State Machine	This module describes how state machines are used by engineers to create useful real-world applications. You learn how to build a state machine in LabVIEW, when to use one, and how to refactor existing code into a state machine.	• Why State Machines Are Useful • Creating a Simple Signal Plotter VI • Clearing the Chart at Startup • Generating More Types of Signals • (Exercise) Clearing the Chart When User Changes the Signal Type • State Machine Concepts and How they Map Into LabVIEW • Converting the Signal Plotter VI into a State Machine
2. Adding Features and Capabilities to our Signal Plotter State Machine	This module describes how to add new features to an applications that uses a state machines. You'll learn why programmers use state machines, how to add new features to the Signal Plotter VI, and how to add new structural capabilities to a state machine in LabVIEW.	• Lesson 1: Why Programmers Use State Machines • (Exercise) Add the exit condition to the state machine • Adding an Event Structure to Capture User Input • (Exercise) Add Support for the User to Control the Signal Amplitude
3. Introducing the JKI State Machine	This module describes how the JKI State Machine enables fast and easy development of LabVIEW applications. You'll learn the fundamental parts of the JKI State Machine, how it improves a traditional State Machine, and the benefits of using the JKI State Machine in your applications.	• Introducing the JKI State Machine • Application Challenges You Can Solve with the JKI State Machine • Key Parts of the JKI State Machine
4. Fundamentals of using the JKI State Machine	This module describes how all the parts of the JKI State Machine work together in a running VI, and how to use the LabVIEW IDE integration features like the State Machine Explorer and other helper windows. You'll learn how to add a JKI State Machine to your code and run it, how to navigate the state and data flow of a JKI State Machine, and how the JKI State Machine handles errors.	• Basics of Using the JKI State Machine • Using the JKI State Machine Explorer (Navigating Your Code) • Using the JKI State Machine Explorer (Editing Your Code) • Workflow for Adding New Features to Your VI (Introduction) • Workflow for Adding New Features to Your VI (More Examples) • Quick Reference (PDF) for the JKI State Machine Course
5. Building a Sequencer using the JKI State Machine	This module allows you to apply the workflow you learned in Module 4 (how to add new features to your VI) into practice, as you build a simple sequencer using the JKI State Machine. You'll learn, by doing the exercise, how to apply a workflow process, to properly organize application code and data inside the JKI State Machine.	• Introducing the Simple Sequencer Hands-on Exercise • (Exercise) Adding the "Load Sequence" functionality • (Exercise) Adding functionality to be able to run a *single* step in the sequence • (Exercise) Adding functionality to be able to run the entire sequence
6. Best Practices	This module describes best practices for using the JKI State Machine, in order to avoid common pitfalls of application development. You'll learn over a dozen best practices that will help you write better code by avoiding common mistakes. You'll learn from the experience of others who have been successful using the JKI State Machine.	• Add States to Front of Queue • Don't hide your state strings in subVIs • Don't add code and logic inside the Event Structure • Keep the Original Size (i.e. don't grow the structures) • Use macros instead of "chaining" sequential states • Left-justify State Strings • Limit the Number of Total States • Know when to use SubVIs (and when not to...) • Use State Categories and Headings • Use "UI: Cursor Set >> [Busy | Idle]" • Understand How "Timeout" Works (and use it properly) • Call the Event Structure Explicitly (in long or looping sequences) • Don't Block the Loop (with long-executing states)
7. Advanced Communication & Event Handling	This module describes dynamic user events and how to configure a JKI State Machine to send and receive events. It demonstrates how to adding a remote user interface VI that controls a low-level sequencer. You'll learn how to send and receive messages between multiple JKI State Machine in order to achieve interprocess communication (IPC), which will take your JKI State Machine to the next level!	• Introduction to Dynamic User Events • Subscribing to Dynamic Events in a JKI State Machine (Inbound Messaging) • Responding to Inbound Messages to Establish Bidirectional Communication (Outbound Messaging)