

## PACSystems Programming

### Course Description

The **PACSystems Programming** course familiarizes students with GE Intelligent Platforms PACSystems Controller programming concepts and Proficy Machine Edition software. Starting with the PACSystems software architecture, students are taught how to effectively develop control applications using building block concepts. This course builds upon Object-Oriented concepts with User-Defined Function Blocks (UDFBs), as well as the development of application components using Structured Text (ST) and Function Block Diagram (FBD) programming languages.



### Who Should Attend?

This course is intended for those who are or will be involved in the development, modification, and troubleshooting of control systems using Proficy Logic Developer and PACSystems controllers.

### Are There Any Prerequisites?

Participants should be comfortable operating Microsoft Windows, and have some experience of:

- Control fundamentals
- Proficy Logic Developer PLC
- Ladder Diagram (LD) programming

### What Tasks Will Be Taught in This Class?

Upon completion of this course, the student will be able to:

- Describe the PACSystems Software Architecture
- Describe the RX3i PACSystems Controller Operation & Components
- Utilize the basic functions of Proficy Machine Edition – Logic Developer PLC
- Create New Machine Edition Projects
- Configure and work with RX3i PACSystems Controllers
- Utilize programming guidelines for developing robust control applications
- Understand and effectively use the different types of PACSystems Variables
- Use UDFBs to build structured applications
- Effectively use the Machine Edition Toolchest as a repository for Application Building Blocks
- Create, Monitor, and Modify running PACSystems applications
- Develop application components using Structured Text & Function Block Diagram programming
- Create application using Application Building Blocks previously developed in the course

### Course Length

4 days

### Suggested Class Size

10 students

### Class Hours

8:00 am - 5:00 pm, daily



## Course Agenda

*(Schedule and Content may vary.)*

### Day 1

#### Morning

##### **PACSystems Basics**

PACSystems High Level Architecture;  
What is a "PAC"?  
PACSystems Blocks used to build applications;  
Programming Languages Overview;  
PACSystems Variable Concepts

##### **RX3i Controller Operation & Components**

RX3i Controller and Components Overview;  
Finding PACSystems technical information;

##### **Proficy Logic Developer PLC**

Proficy Machine Edition programming environment:  
Navigator, Inspector, Companion, Editors, Feedback  
Zone, Data Watch, InfoViewer, and Toolchest;  
Basic Machine Edition concepts: Projects & Targets;  
Machine Edition "Best Kept Secrets";  
Project Management and documentation.

#### Afternoon

##### **Working with PACSystems Controllers**

Serial and Ethernet Communications;  
Project Validation and Download;  
Upload from Controller;  
Verify Controller Equality;  
Side-by-side Ladder Diagram Compares;  
Clear Controller Contents;  
Work with Fault Tables;  
View Controller Status Information;  
Configure PACSystems RX3i CPU, Ethernet, and I/O  
modules;  
Hardware Reference View;  
Identify and Correct Configuration Errors;  
Export and Import hardware configuration;  
Print Hardware Configuration Report.

### Day 2

#### Morning

##### **Monitor & Modify Applications**

Program changes using Online Change Methods:  
Word-for-Word, Run Mode Store, and Test Edit;  
Monitoring Tools: Data Watch, Reference View  
Table, Logic Editors, and Data Monitor  
Ladder Diagram Editor Options

##### **Proficy Machine Edition Toolchest**

Navigating through the Toolchest;  
Create Toolchest Drawers;  
Share Toolchest Drawers;  
Save logic to a Toolchest drawer;  
Use Toolchest logic in application;  
Save logic modifications to Toolchest;  
Toolchest use of Universal Variables "\$".

#### Afternoon

##### **Structured Text Programming**

Create ST Blocks;  
Create ST Logic;  
Create ST Comments;  
ST language Statements, Operators, and Syntax;  
Types of ST statements: Assignment, CASE, Function  
Call, RETURN, EXIT, IF, FOR, WHILE, Repeat, and  
Empty;  
Edit ST logic;  
Use/Create variables in ST logic,  
Monitor ST logic;  
Call UDFBs from ST logic;  
ST Editor configuration options;  
Considerations when programming in ST;  
Build ST Blocks for Re-use.



## Course Agenda

*(Schedule and Content may vary.)*

### Day 3

#### Morning

##### Function Block Diagram (FBD) Programming

Introduction to Function Block Diagram;  
Create FBD Blocks;  
Create FBD Logic;  
FBD logic solve order;  
FBD Comments;  
FBD Instructions;  
Edit and wire FBD logic, negative Boolean flow;  
Assign FBD logic variables, change online values;  
Monitor FBD logic;  
Call UDFBs from FBD logic;  
FBD Editor configuration options;  
Print FBD logic;  
Considerations when programming in FBD;  
Build ST Blocks for Re-use.

#### Afternoon

##### User Defined Function Blocks

Basic UDFB facts;  
UDFB Input, Output, and Member Variables;  
How UDFBs operate;  
How to create and monitor UDFBs;  
UDFB Toolchest considerations;  
Update an existing UDFB with a new UDFB;  
Build UDFB Blocks for Re-use.

### Day 4

#### Morning

##### Working with PACSystems Variables

Variable Types: Scalar, Array, & Structure;  
Understand Variable Data Types: INT, REAL, etc.  
Variable Scopes: Universal, Global, and Local;  
Access System Variables;  
Use Symbolic, IO, Mapped, and Alias Variables;  
Access Variable information – bit in word, array element, indirect, and structure element;  
Variable attributes;  
Understand Constants.

#### Afternoon

##### Create Sample Application with Application Building Blocks

Create, monitor, and debug a sample application using the Application Building Blocks previously developed during this training course: UDFBs, ST, and FBD Blocks;  
Use Toolchest to re-use application components;  
Manage application with multiple components.

##### Programming Guidelines

Guidelines such as program organization and naming conventions for variables, blocks, etc.

