JUPITER PRIME
Consulting • Training • Advisory
جوبيتر برايم للاستشارات والتدريب

COURSE BROCHURE

Test and Measurement Data Acquisition Programming Training Course

28 December 2026 – 1 January 2027
Geneva

Test and Measurement Data Acquisition Programming Training Course

Ref: 3183_120989 **Date:** 28 December 2026 – 1 January 2027 **Location:** Geneva **Fees:** 7500 GBP

Test and Measurement Data Acquisition Programming is a 1-week training course for test, instrumentation and electrical engineers who build measurement applications, centred on a structured data acquisition application specification for a test rig. As product validation, equipment commissioning and research benches rely on automated measurement, organisations need engineers whose code captures signals faithfully and survives daily use. Jupiter Prime Training treats test and measurement data acquisition programming as one path from sensor to trusted result.

Why Test and Measurement Data Acquisition Programming Matters Now

Three forces shape measurement software today. Test benches now combine many sensor channels, instruments and actuators, so hand-recorded readings no longer keep pace. Quality and reliability teams expect raw data, scaling and timestamps to be traceable when a result is challenged months later. Engineering staff also move between projects, so a bench application written without structure becomes impossible to maintain once its author leaves. Fragile acquisition code loses samples silently and casts doubt on every report. The course rests on accepted signal measurement and software design practice, described in general terms.

Who Gains Most from Test and Measurement Data Acquisition Programming

- Test engineers automating product validation and endurance benches
- Instrumentation engineers wiring sensors and configuring measurement hardware
- Electrical engineers capturing voltage, current and power quality signals
- Commissioning staff logging equipment performance during acceptance tests
- Research and laboratory engineers building repeatable experiment rigs

Test laboratories gain most when the engineers who wire benches and those who write their software attend together and agree one application structure.

Test and Measurement Data Acquisition Programming Capabilities

Participants move from a raw sensor signal to a logged, scaled and reviewable measurement handled by maintainable code.

- Design dataflow block diagrams that acquire, process and display measurements — the engineer building bench software
- Specify sample rate, input range and triggering for each channel — the instrumentation lead protecting signal fidelity
- Apply scaling, filtering and unit conversion to raw readings — the reviewer trusting reported values
- Build operator front panels with clear controls and alarms — the technician running daily tests
- Test state machine logic and error handling under fault conditions — the test manager relying on unattended runs
- Communicate a structured application specification for a test rig — the laboratory approving new benches

Key Test and Measurement Data Acquisition Programming Terms

- Dataflow programming — A style where functions run when their input data arrives, drawn as wired blocks rather than written lines.
- Sample rate — How many readings per second a channel takes, which limits the fastest signal that can be captured.
- Aliasing — False low-frequency patterns that appear when a signal is sampled too slowly for its content.
- Hardware trigger — A signal edge or level that starts acquisition at a precise moment.
- State machine — A program structure that moves between defined states, such as idle, running and fault.
- Producer-consumer loop — Two parallel loops where one acquires data and the other logs or processes it.

Test and Measurement Data Acquisition Programming Day by Day

Day 1: Measurement Chains and Dataflow Basics

- Sensor, Conditioning and Acquisition Hardware Chain — Locating Sources of Measurement Error
- Front Panels and Block Diagrams — Separating Operator View from Program Logic
- Data Types, Wires and Clusters — Choosing Structures for Channel Data
- Loops, Shift Registers and Timing — Controlling Execution Rate
- Sub-Programs and Icon Connectors — Building Reusable Measurement Blocks

Day 2: Acquiring Signals and Instruments

- Channel Configuration and Input Ranges — Matching Hardware to Sensor Output
- Sample Rate, Buffering and Aliasing — Capturing Signals Without Distortion
- Software and Hardware Triggering — Starting Measurements at the Right Moment
- Thermocouple, Strain and Current Inputs — Handling Common Bench Sensors
- Instrument Control Over Serial and Network Links — Reading Bench Meters

Day 3: Scaling, Analysis and Logging

- Linear and Polynomial Scaling — Converting Raw Counts to Engineering Units
- Filtering, Averaging and Spectrum Checks — Cleaning Noisy Signals
- File Formats, Headers and Timestamps — Making Logged Data Traceable
- Producer-Consumer Loops — Logging at Speed Without Losing Samples
- Reports and Summary Tables — Presenting Results to Reviewers

Day 4: Application Structure and Robustness

- State Machine Design — Sequencing Idle, Running, Paused and Fault States
- Event-Driven Operator Interfaces — Responding to Buttons Without Polling
- Error Clusters and Fault Recovery — Safe Shutdown of Test Equipment
- Configuration Files and Calibration Constants — Changing Tests Without Editing Code
- Code Documentation and Version Control Habits — Keeping Bench Software Maintainable

Day 5: Practising the Test Rig Application Specification

- Specifying a Motor Endurance Bench — Applying Channel and Trigger Choices
- Specifying a Battery Cycling Logger — Applying Scaling and Logging Methods
- Specifying an Equipment Acceptance Test — Applying State Machine Structure
- Bench Validation and Known-Signal Checks — Proving the Application Measures Correctly
- Completing the Data Acquisition Application Specification — Approval Before Coding Starts

Applying Test and Measurement Data Acquisition Programming After the Course

- Check the sample rate and input range of one existing bench against its fastest signal.
- Restructure one bench program into a state machine with clear fault handling.

Test and Measurement Data Acquisition Programming Questions FAQ

What is test and measurement data acquisition programming and why does it matter now?

It is the writing of software that reads sensors and instruments, scales the readings and stores them for review. It matters now because test benches have more channels and results must stay traceable.

Which roles gain most from test and measurement data acquisition programming?

Test and instrumentation engineers who build bench software gain most. Electrical, commissioning and laboratory engineers who log equipment performance also benefit.

How does test and measurement data acquisition programming differ from a plant control systems course?

It focuses on measurement benches and laboratory rigs rather than continuous plant control. You work on sampling, logging and test sequencing, not on process control loops.

What do you bring back from test and measurement data acquisition programming?

You bring back a structured specification for a test rig application. It covers channels, triggers, scaling, logging, states, faults and validation checks.

Test and Measurement Data Acquisition Programming: The Way Forward

The data acquisition application specification first informs whether a new test rig is ready for coding or needs its channels and triggers rethought. Software engineers build from it, instrumentation staff wire to it and laboratory leads approve it. After its first use, revisit the sample rate and logging choices, where most data losses begin.