



## Technical Education Services Course Specification

Course Number: RH3800

Course Title: SIMulation Workbench™

Course Duration: 3 Days

### **Purpose:**

SIMulation Workbench provides a complete framework to develop and execute real-time hardware-in-the-loop and man-in-the-loop simulations from C/C++/Fortran/MATLAB/Simulink models. This course guides the student through the process to configure, start, stop, record and play back simulation runs; build and execute photo-realistic HMIs. Concurrent iHawk multiprocessing systems running SIMulation Workbench are based on COTS components offering the latest, leading-edge processor, chipset, memory and I/O bus technology. Students will learn how to take advantage of these features to target individual I/O processes and multi-rate models to specific CPU cores and I/O buses for parallel execution, thus allowing the simulation loop to run at faster frame rates.

### **Intended Audience:**

This course is intended for application engineers running simulation models in realtime on Concurrent systems using the RedHawk™ Linux® real-time operating system.

### **Course Objectives:**

Upon successful completion of this course students are able to:

- Describe the SIMulation Workbench system architecture to include the real-time host, logging facilities, and clients in a monolithic or distributed environment.
- Explain the cycles of the SIMulation Workbench scheduler frame, their related components and function of each, and the various mechanisms available to configure the SIMulation Workbench scheduler.
- Describe the three main components of the Real-Time DataBase, the purpose and functionality of the current and alternative values table.
- Describe the synchronous and asynchronous I/O tasks including the functionality and capabilities of each class.
- Using the SIMulation Workbench MLToolkit and Control Center GUI, configure, build and run Simulink created models in realtime.
- Describe SIMulation Workbench I/O device configuration, as well as the creation of analog, digital, and string points.
- Describe, configure, and use the Data Log Viewer to visualize SIMulation Workbench RTDB values.

- Using the SIMulation Workbench HMI Display Builder and HMI Run-time, create and run basic GUIs.
- Write, build, and run scripts for running during the simulation test cycle.
- Describe how simulation tests combine scripts, models, an RTDB, initial conditions and into a full simulation test session.
- Describe, configure, and use the Real-Time Viewer to monitor running tests.
- Explain the mechanism to use Real-Time Forms to export and import sets of variables with running tests.

**Prerequisites:**

- C/C++ Programming Language – understand C/C++ language source code and syntactical constructs.
- Linux System Capability – prior experience with basic Linux system commands and utilities used to create programs.

**Course Topic Outline:**

- I. Real-time Overview
  - A. Hard, Firm, and Soft attributes
  - B. RedHawk Linux Real-time Support
  - C. Hardware-In-the-Loop
    1. Multi-core
    2. Parallel and I/O Independence
- II. SIMulation Workbench Installation
  - A. Prerequisites
    1. Hardware
    2. Network
    3. Software
  - B. Server and Client Installation
  - C. Simulink MLToolkit
- III. SIMulation Workbench Architecture
  - A. Scheduler Frame Cycles
  - B. Models, RTDB, I/O Tasks and Scripts
  - C. Synchronous vs Asynchronous I/O
  - D. RTDB CVT and Alternate Values
  - E. Scheduling Tests
    1. FBS/RCIM mechanisms
    2. OS Timers
  - F. Data Logging
- IV. SIMulation Workbench Control Center

- A. User Controls & Authentication
- B. RTDB & I/O Configuration
  - 1. Analog, Digital, and String Points
  - 2. Engineering Units Conversion
  - 3. I/O Devices, Tasks, and Mappings
- C. Programming
  - 1. Simulink Models and ML Toolkit
  - 2. Scripts and User Programs
  - 3. Client, Real-time, and Script APIs
  - 4. Environment Variables
- D. Tests, Sessions, Suites, and Initial Conditions
- E. Data Log Viewer
- F. RT Viewer and Forms
- G. HMI Display Builder and Run-time

**Laboratory Exercises:**

Students are provided with the opportunity to perform hands-on exercises for topics presented to include the following:

- Review exercises are fill-in type questions that require the student to review the material presented and respond. These questions reinforce the important points presented in each topic.
- Hands-on exercises provide the student with experience in using the Control Center, commands, utilities, APIs, and techniques from the material presented. This allows students to better understand what they have learned.
- Students will be provided example source code in order to run, modify, and explore various concepts to reinforce topics presented.