

## MOST Logic Workbench

8459-LC-MT

- ◆ Project management tool
- ◆ Network configuration & administration
- ◆ Tagname-based configuration - simplifying system configuration
- ◆ Auto-tagname generation
- ◆ Control strategy development tools
- ◆ Control strategy simulation
- ◆ I/O configuration
- ◆ Peer to peer communications
- ◆ Remote data table for mapping Modbus® devices
- ◆ Import/export utilities and reports
- ◆ Project back-up and restore

### Overview

The MOST Logic Workbench is an integrated project development environment that centralizes and coordinates project engineering including instrument index, I/O database configuration and control strategy development. It is a client-server implementation allowing multiple engineers to work on the same project.

In addition to the extensive control software, the Workbench also provides significant tools for your control system, including I/O Configurator, Program Compiler, Downloader, Remote Data Table Builder for Modbus, Tag Builder, Network Configurator and a Peer to Peer table builder. With this extensive range of tools, the Workbench is much more than a development system, it also provides overall system management.

Designed using a Microsoft® Windows® Explorer paradigm, the Workbench provides a powerful and flexible framework that significantly reduces project implementation and maintenance costs. In addition to working with an easy-to-use hierarchy of objects and folders, capabilities such as drag-and-drop, right mouse click support and context sensitive help inherent within the Workbench make developing, testing and maintaining automation projects easy and straightforward. The Workbench also provides system-wide security allowing for multiple levels of access to the control system.

### MOST Workbench comprehensive tools

The MOST Logic Workbench provides IEC61131-3 programming for discrete control. It configures the control logic and generates an export data base to create the operator (HMI) database.

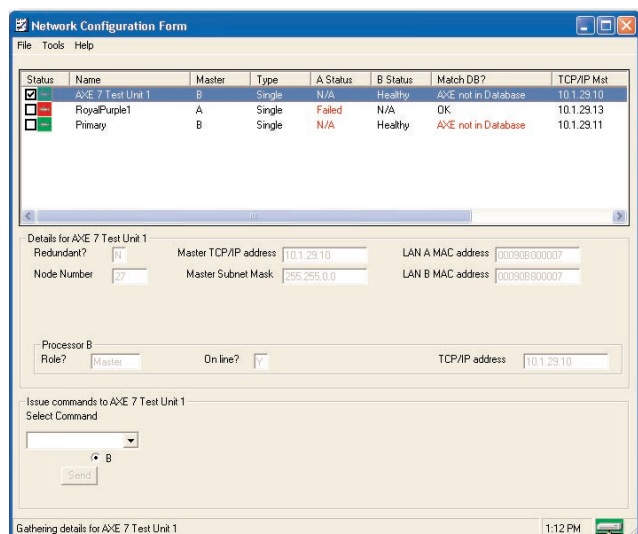
The system eliminates expensive, laborious and sometimes error-prone manual engineering work typically required during control projects. Manual generation of bill of materials, control strategy diagrams, and operator interface requirements are abolished. Repetitive programming of control strategies for advanced control is no longer necessary. Similarly, eliminating the repetitive input of system information significantly reduces errors and configuration time.

All Workbench activity is centered on a project(s) that is created to address a control need. Controllers are selected and assigned to the project after it has been created.



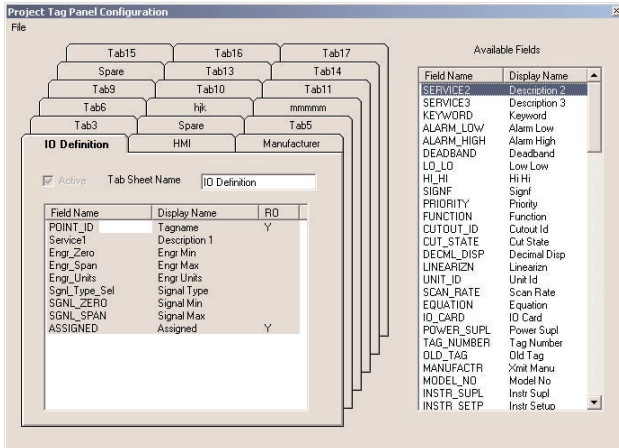
### Network configuration tool

The Workbench's Network Configuration Tool can be used to configure your entire plant network, setting up and maintaining TCP/IP addresses for all devices on the network. After the network is setup, the Workbench is used to monitor and maintain the devices on the network. Each controller, including redundant controllers, needs IP addresses, MAC addresses and node numbers. If the MAC addresses are unknown, they can be found through the Network Configuration tool. Then the Network Configurator can be used to manage the assignment of IP addresses to the controllers. It also provides additional information about each controller that has been configured, such as whether it is a single or redundant controller and the status of both Ethernet LANs.



### Tagname based system

All variables are referred to by tagname, providing a descriptive reference to all variables throughout the system, replacing the need to reference register addresses. The Logic Workbench models a process system input/output with the assignment of module and



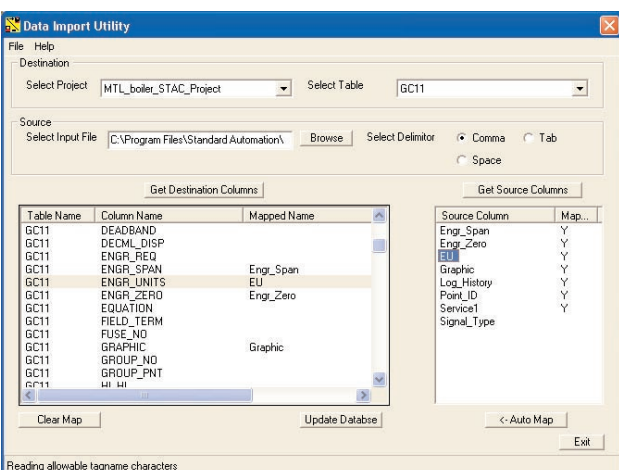
point destinations. Using a predetermined template, this task is accomplished as a simple "fill in the blanks" procedure and builds the project I/O configuration database.

### Tag import utility

The Tag Import Utility can be used to import a tag list that is already defined in a spreadsheet format. The utility imports a .csv, tab or space separated file directly into the Workbench, saving the need to manually recreate this information if it already exists.

### Peer to peer capability

The peer to peer capability allows data from a controller to be used on the network. This takes advantage of the high-speed peer to peer communications of the Ethernet network, resulting in more efficient



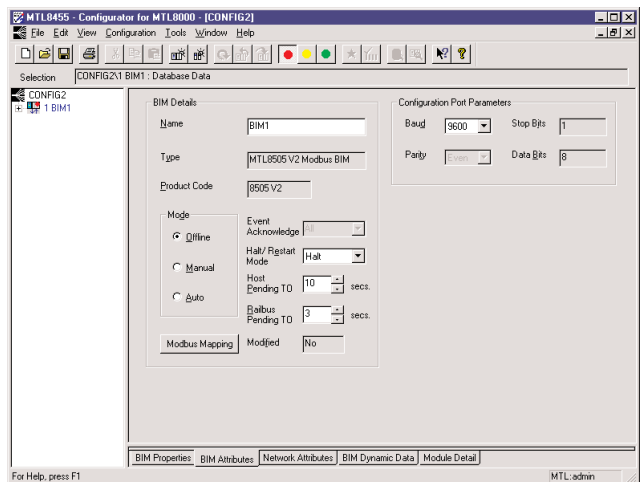
control processing. Peer to peer data is setup with tag name assignment, with the tag names being available to all controllers on the network. Assigning peer to peer data is easy, all you need to do is simply drag and drop Tagnames from another project or controller to the current controller.

### Standardized reports

All databases are MSDE (Microsoft Data Engine) databases, which are fully SQL compliant and can be readily shared throughout the system. This design enables many standardized reports to be readily generated, including a list of Assigned I/O, Available I/O, Unassigned Points, Factory Acceptance Test status, Point Overview report that lists transducer information for all the tags in the system, Tagname for HMI (which returns the information located on the HMI tab of the Project Tags window) and Tagname Index (which returns the information located on the I/O Definition tab of the Project Tags). These reports simplify the overall management of the project.

### I/O Configurator

I/O modules connected to a controller must be configured to permit operation. The controller needs to be told the identity of each I/O module fitted and values of relevant parameters. Modules are added by a simple drag and drop process. Each module has its own set of parameters and these appear with default values when the module is first added to the system. These can then be edited individually to specify the required values and then tagnames are assigned to each I/O point. Failsafe values can be defined to assure that the modules operate safely if there is a system interruption.



### Dynamic data

Real-time data is available on-screen. The viewable data includes primary variables as well as status parameters. This can be refreshed continually or when the operator requires a screen refresh.

### Configuration upload

Configuration data already contained in a controller can be uploaded to the Workbench. This can be saved as a PC file and copied to other Controllers. PC files that were created for earlier versions of the controller firmware can also be opened and edited for use with the latest versions of the controller.

### Simple tabbed pages

The right hand side of the screen contains information on any item that is highlighted in the navigation tree to the left. For example, with the network icon selected, three tabbed pages of information are available. These pages define and describe the network protocol, the controllers that are attached to the network and the details of the project contained in the overall configuration file.

# MOST Logic Workbench

## MOST Logic Workbench

8459-LC-MT

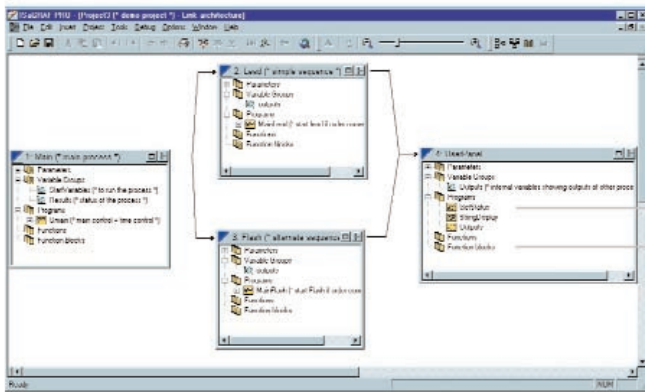
### IEC 61131-3 languages

MOST Logic Workbench is 32-bit software for distributed control applications. It supports all five IEC 61131-3 languages: Ladder Diagram (LD), Structured Text (ST), Instruction List (IL), Sequential Function Chart (SFC) and Function Block Diagram (FBD), plus the Flow Chart (FC) language.

Its workbench is used initially to create and edit the control code and may also be used later for simulation, debugging and on-line monitoring. The code created in the workbench is then downloaded to the MOST Logic Controller.

### Language options

**The Structured Text (ST)**, is a high level structured language with a syntax similar to Pascal but more intuitive to the automation engineer. This language is mainly used to implement complex procedures that cannot be easily expressed with graphical languages.



The ST text editor guides the user to the correct syntax and punctuation and provides the best validation and programmer assistance facilities.

**The Sequential Function Chart (SFC)** language divides the

process cycle into a number of well-defined steps, separated by transitions. The Workbench fully supports graphical Function Chart programming, which is familiar to many engineers.

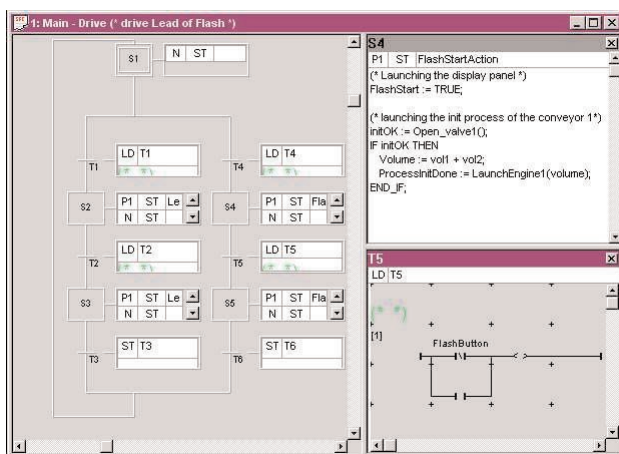
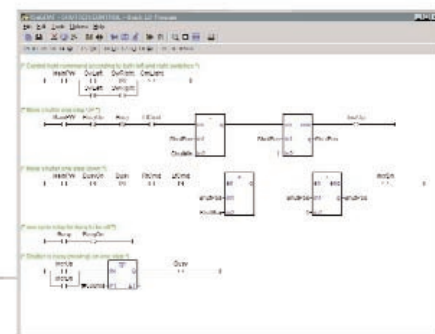
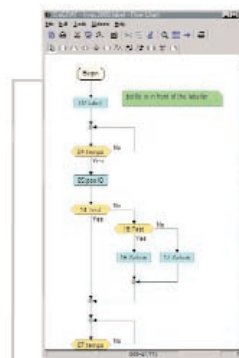
**The Ladder Diagram (LD)** is one of the most familiar methods of representing logic equations and simple actions. Contacts represent input arguments and coils represent output results. Each block in the selection list has a description text.

**The Instruction List (IL)** is a low-level language, similar to the simple textual PLC languages. IL is a register-level programming language. MOST Logic Control has a set of more than 60 IEC functions and function blocks. Users can enlarge this set by writing

functions and function blocks in the LD, FBD, ST, and IL languages.

**The Flow Chart** is an easy to read decision diagram where actions are organized in a graphic flow. The Flow Chart Editor has full support for connectors and sub-programs.

The development process begins with the division of a project into several PLC Loops, also called Resources, identifying their host hardware



platforms, and defining the links between them. The physical division of a project is represented by configurations and communications networks. A Configuration represents a hardware platform which contains a single resource.

**Function Block Diagram (FBD)** is a graphical language which allows the user to build complex procedures by taking existing function blocks from the Workbench library, and wiring them together on screen. The FBD editor allows manual input of variables. The diagrams can be zoomed to view the whole diagram or specific areas in more detail. User can mix LD and FBD programming in the same chart.

### Program development process

A resource contains several programming units called Program Organization Units (POUs). POUs can be programs, functions or function blocks. Programs can be described in any of the five IEC 61131-3 languages plus Flow Chart.

#### Linking variables to I/O channels

Before defining the project code, the I/O is defined in the Workbench in the form of a tag database, which identifies the I/O channels and their respective field devices. Once a device is selected, a simple mouse-click 'connects' a variable to a channel.

#### The Build

To validate the project, the project code must be built. This step is also very useful for syntax checking; all detected errors can be easily located with a simple mouse-click. The generated code is fully publicly documented and supported. The code generator produces the code for each resource.

#### Control Strategy Simulation

Simulation enables the project code to be tested without the need for any hardware to be attached. Each resource can be executed cycle-by-cycle, and various system variables, such as the cycle time, can be monitored or forced.

For debugging with real platforms, or to perform maintenance operations on 'live' systems, changes can be downloaded on-line without stopping the running resources. During the testing phase I/O devices can be set as virtual, if the MOST Logic Controller is not completely ready or is unavailable to the programmer. Function block instances can be directly debugged from editors.

The configuration manager and virtual machine are launched and a *Simulate I/Os* panel is displayed. The Simulation Mode can be started from the Project Manager or from a programming language editor. The workbench is intuitive and user-friendly, but to further assist the user, an HTML-based cross-referenced on-line help system is provided, which includes a complete language reference.

The Workbench also offers a document generator. Project items are shown as a tree, the table of contents of the project documentation can be customized by a simple click on each item.

To allow re-use of code, libraries of IEC functions and function blocks can now be developed. Functions and function blocks are designed and tested as in any other projects, but other projects can be linked to these "library projects" to allow the use of their functions and function blocks. Once a "library project" has been selected, its blocks can be selected as standard blocks. As libraries, import/export functionality allows the sharing of POUs between projects. It is also a comfortable way to integrate the work of several programmers to constitute the final project.

### The Control System

Each downloaded application is a complete distributed application running on the target controller. Multiple applications, running on separate controllers, can be linked together using the peer to peer tool, which identifies tagnames for use across the network. This data exchange mechanism provides a very efficient way of sharing information between multiple controllers.

### Project backup and restore

You can create a backup file of your project at any time. This useful feature enables you to save your application in a known state, and if necessary, restore the project to that state. You can import and export resources from one project to another, enabling you to re-use prior work. If the results of importing or exporting resources are unsatisfactory, you can choose to restore the project.

### Downloader

The Workbench Downloader is used to download the firmware and control strategy to the controller. With Redundant Controllers, strategy changes can be downloaded to a backup controller without disrupting control in the primary controller. When the download is complete, the backup controller will take over control and update the primary controller with the changes. Control can revert back to the primary controller after these changes are incorporated. Downloading to a simplex controller can be done on-line and without initializing data, the new strategy starting on the next execution cycle.

### Remote data table builder for Modbus import

The remote data table is used for connecting to remote Modbus master or Modbus slave devices. Tagnames are assigned to these devices, which are then included in the control strategy. An example of a remote Modbus device is a Modbus TCP compliant H1 Linking device that could be used to bring Foundation Fieldbus H1 data into the control system.

### MOST

The mission of MTL Open System Technologies (MOST) is to design, manufacture and distribute modular, open hardware and software technology-based components for use in the most demanding real-time system applications. These field-proven components can be "snapped" together to create sophisticated control systems.

## System requirements

#### Processor

Minimum .....Intel Pentium III - 700Mhz

Recommended .....Intel Pentium 4 - 1.7Ghz

**Operating System** .....Microsoft® Windows® 2000 (SP2) or XP

**RAM** (to open all aspects of the Workbench)

Minimum .....128 MB

Recommended .....512 MB

**Hard Disk space** (for complete installation)

(Including MSDE components - See notes below)

Minimum .....500 MB

Recommended .....1 GB

#### CD-ROM drive

#### Notes

1) Installation software checks for presence of Microsoft Data Engine (MSDE) and installs the necessary components if they are missing.

2) Additional space will be required to store any configurations the user creates.

On-line help files are available for the individual components of the Workbench.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation.  
Modbus® is a registered trademark of Schneider Automation Inc.