Logix Sim®

User Guide

Logic Design Inc.

INDEX

1. LogixSim Lab Simulation Software

LogixSim is a suite of simulation software products, and is designed to provide "real world" interactive laboratory simulation of electromechanical devices and circuits, robotics, and programmable logic controllers. The four simulators included with LogixSim are 3DLab, CircuitLogix, RoboLogix and PLCLogix. 3DLab have dozens of realistic lab environments, ranging from a test bench with electromechanical devices and measuring instruments to traffic lights, factories, car washes and elevators. The lab projects that feature 3D environments are accessed through either 3DLab or PLCLogix.

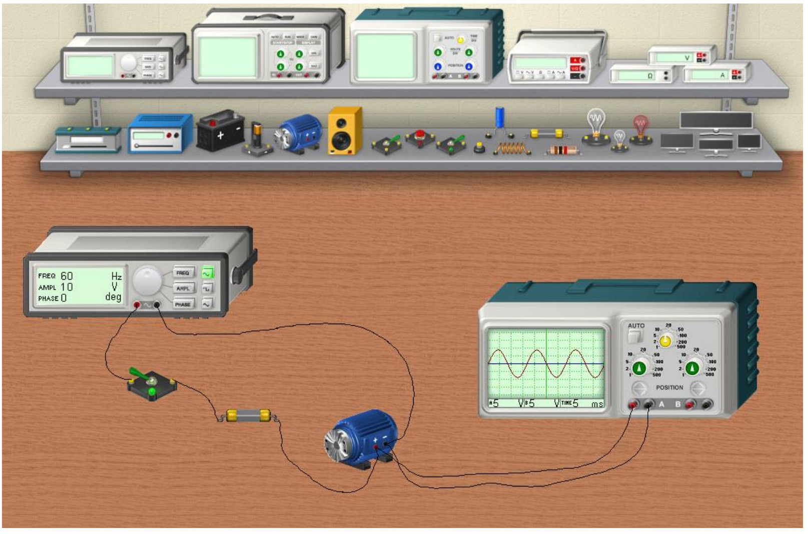
CircuitLogix is a software simulation program that converts your computer into a fully-functioning electronics laboratory with thousands of components and devices that are easily interconnected. It supports analog, digital and mixed-signal circuits, and its proven SPICE simulation gives accurate real-world results you can trust.

RoboLogix is designed to emulate real-world industrial robotics applications. With RoboLogix, you teach, test, run, and debug programs that you have written yourself using a five-axis robot in a 3D simulated manufacturing environment. It is ideal for students as well as industry professionals and provides engineering-level robotics simulation at an affordable price.

PLCLogix is a Programmable Logic Controller (PLC) simulator that enables you to gain "hands on" experience in the operation of the Logix 5000 PLC. PLCLogix provides much-needed programming practice by creating and running your own ladder logic programs using tag-based memory in realistic 3D PLC control applications.

1-1 3DLab

3DLab is a simple-to-use, yet very powerful simulation software product that closely replicates a fully functioning electrical-electronics laboratory in a realistic 3D environment. This 3D simulation software is a "virtual reality"-type of lab that combines an interactive 3-dimensional learning environment and "real world" electronic devices and tools, to greatly enhance the user's comprehension of basic electricity and electronics. The image below show the basic configuration of the 3DLab simulation environment



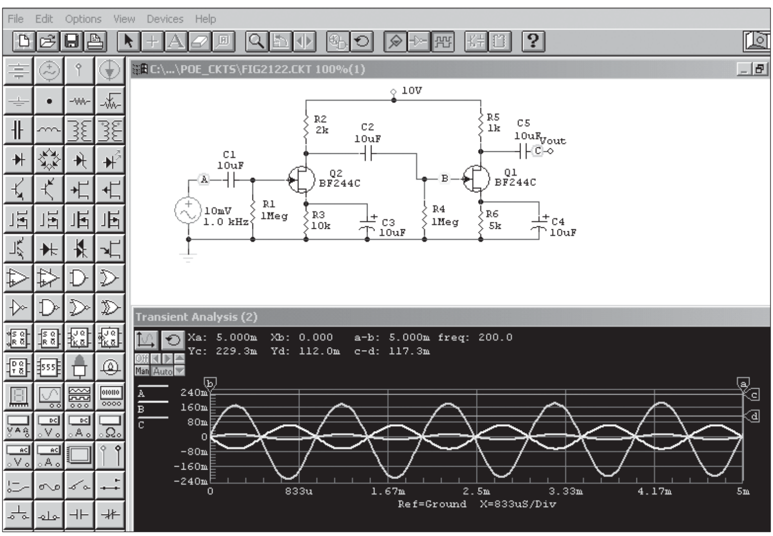
3DLab is intended for the study of basic electricity and electronics, including DC fundamentals, AC circuits and troubleshooting techniques. The user-friendly interface provides a highly intuitive method of observing circuit behavior through use of laboratory experiments and procedures. 3DLab provides a virtual electronics lab where you can design, build, and test electrical and electronic circuits. Placing devices onto the worktable is as simple as moving your mouse. Once you have selected a device off the shelf, left-click again to place the device on the table. To return the device to the shelf, right click on the device with your mouse and move it off of the table.

There are approximately 30 different devices and instruments available in 3DLab. These components include batteries, switches, meters, lamps, resistors, fuses, inductors, capacitors and instruments including oscilloscopes, logic analyzers and frequency counters.

In a real lab, you would place components on a work surface and prepare them for interconnection. Wiring would be done either on a breadboard, or by using jumper wires. Parts are selected by picking them off the shelf with your mouse and moving them onto the workspace. Wiring of the circuit is accomplished by using your mouse cursor to draw a wire between two points. To start a wire run in 3DLab, move the cursor toward one of the component's connection terminals until the cursor changes into a small circle. Left-click with your mouse and begin to draw the wire by moving the mouse cursor towards the next connection point. You can draw along any curve and 3DLab will create a smooth path. Switches and other devices in 3DLab can be turned on or off, opened or closed, by left-clicking on the device with your mouse.

1-2 CircuitLogix

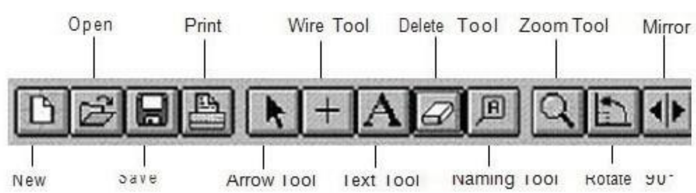
CircuiLogix has advanced schematic capabilities which allow you to design electronic circuits and output netlists for PCB design tools and autorouters. You can also perform fast, accurate simulations of digital, analog, and mixed analog/digital circuits. With a minimum of electronics theory, you can successfully use CircuitLogix to design and simulate circuits. For beginners, CircuitLogix is perfect for learning and experimenting with electronics and circuit design. For advanced users, CircuitLogix powerful analyses provide a sophisticated environment for testing and trying all the "what if" scenarios for your design. Best of all, you can accomplish more in less time than traditional prototyping methods.

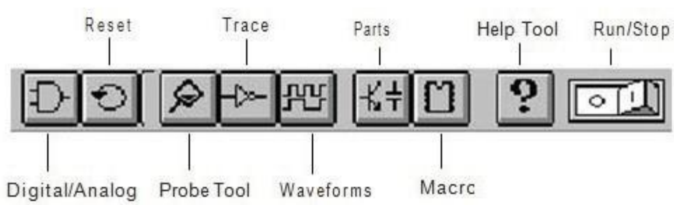


When you start CircuitLogix, the blank workspace appears. This is where you place devices that represent real-life components such as resistors, transistors, power supplies, etc. The CircuitLogix workspace also includes the Toolbar, Menu Bar, and special windows for circuit simulation and testing purposes. After placing devices exactly where you want, you simply wire them together. The wiring lines you draw form intelligent links between the devices, which then allow the circuit to be simulated, tested, and analyzed.

An important feature of CircuitLogix is the way electrical connections between the elements in your design are recognized. The concept of connectivity is the key to using CircuitLogix to draw and simulate electronic circuits. The program stores connection information for simulation, and it is also used for creating and exporting netlists into PCB layout programs to create a working printed circuit board (PCB).

The image shown below is the quick reference toolbar displayed on the top of the CircuitLogix screen.

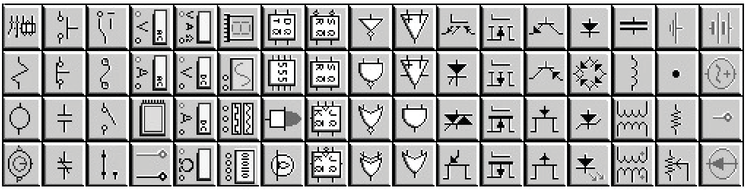




Toolbar Descriptions

Arrow Tool	Select, move and edit devices, wires and text
Wire Tool	Connect the circuit [+Shift for bus wires]
Text Tool	Add text to the circuit
Delete Tool	Delete devices, wires and text
Naming Tool	Name scopes, I/O's, macros, pins and bus wires
Zoom Tool	Magnify or reduce the circuit [+Shift to reduce]
Rotate 90°	Rotate one or more selected devices
Mirror	Mirror one or more selected devices
Digital/Analog	Simulation mode [Digital, Analog]
Reset	Initialize analog and digital simulations
Step	Single-step digital simulations [Setup in Digital Options]
Run/Stop	Run and stop simulations
Probe Tool	Indicates digital states and plot analog data
Trace	Digital simulation method to show the state of all nodes
Waveforms	Displays digital waveforms and analog plot windows
Parts	Displays and selects devices

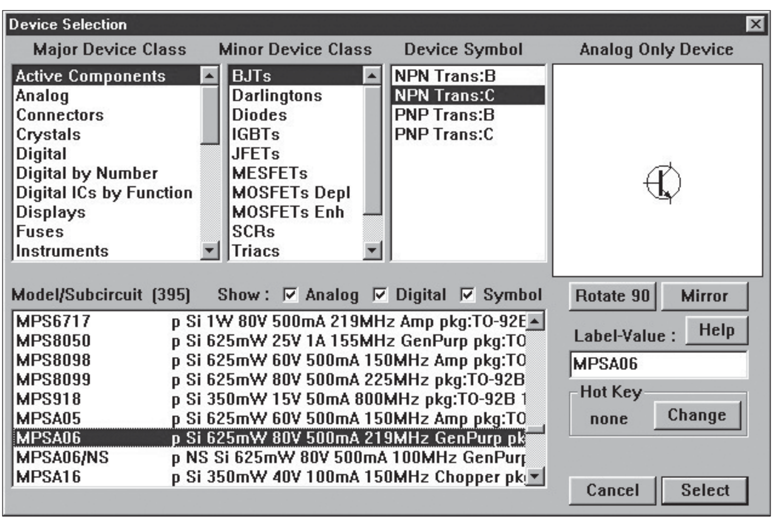
The main device toolbar shown below is located on the left side of the screen and provides access to 68 of the most commonly used devices and instruments. Components are selected from this toolbar by left-clicking on the desired component with your



mouse, and then moving the mouse cursor onto the work area and left- clicking again to place the component.

A much more extensive listing of the available devices is contained in the device library, as shown below. Parts can be selected from the library using the Device Selection dialog box. Commonly-used devices can be added to the Hotkeys menus, allowing you to select the device by pressing one of the user-definable hotkeys on the keyboard.

Locate the device you wish to use by first selecting a Major Device Class, a Minor Device Class, a Device Symbol, and if required, a SPICE model or subcircuit. In this manual, device location is indicated as [major class/minor class]. For example, a 2N3904 could be found at [Active Components/BJTs]. Click on the Select button to select the device for placement in your circuit (or just double-click on the item to be selected in the Device Symbol list or the Model/ Subcircuit list).



Wiring the Circuit

Your circuit must be correctly wired together in order to function. *The circuit will not function if you simply place devices so that device pins are touching the pins of other devices where you want connections to occur!* Instead, you must connect a wire between the pins you want connected together. All wires must be vertical, horizontal, or combinations of these two. It is not possible to wire at odd angles such as 45° or 60°.

There are two basic methods of wiring: Auto Routing and Manual Routing. These two methods are fully integrated; you don't have to switch between wiring modes.

Auto routing has the advantage of being quick and easy to use. Simply place the Wire Tool (or the Arrow Tool if the "Arrow/Wire" option is enabled) on a valid connection point (a valid connection point is any device pin or wire), then *click and hold* the left mouse button. Drag the mouse to another valid connection point and release the mouse button. The wire is automatically routed between the two points.

Auto routing requires two valid connection points. You can- not draw a wire with auto routing that does not connect to something on both ends. Also, you cannot draw bus wires with the auto routing method. The auto routing can be either Simple or Intelligent, depending on what you have selected in the Preferences dialog box. Simple routing draws only one or two wire segments, horizontal and/or vertical, making the shortest path without regard for devices that might be in the way. Intelligent routing tries to find a path which does not cross directly over any devices. If no reasonable path can be found, the simple method is used.

Manual routing has the advantage of routing wires exactly where you want them without need for readjustment after the wire is drawn. It also allows you to place wires in your circuit which are not connected on both ends. Manual routing must be used to draw bus wires.

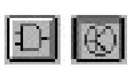
Move the Wire Tool (or the Arrow Tool if the "Arrow/Wire" option is enabled) to the position where you wish to start the wire, then *click and release* the left mouse button. The Wire Tool cursor will disappear until you begin to drag the mouse. It will then be replaced by an extended wiring cursor. The extended cursor simplifies the task of precisely aligning wires with other objects. Click once with the left mouse button to turn 90° or double-click to end the wire. If the "Single Click Connect" option is enabled in the Preferences dialog box, a single click of the mouse will terminate the wire when it is at a valid connection point. A wire can be cancelled at any time while it is being drawn by pressing any key.

Tutorial Example

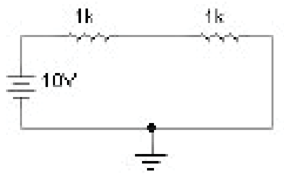
CircuitLogix is capable of simulating analog circuits, digital circuits, and mixed-mode circuits. The following tutorial example is for an analog circuit:


The New Button

1. Click on the New button in the Toolbar. An "Untitled" circuit window will be opened.

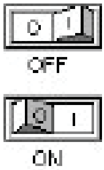

The Digital/Analog Button

2. Select Analog simulation mode. The transistor icon should be visible in the Toolbar, not the AND gate icon. If the AND gate icon is displayed on the button, click on the button.



3. Make sure the "Auto Designation" option in the Options menu is enabled (so there is a check mark by it).
4. Draw the circuit as shown, using the following devices. Click on the Parts button and select 1 Battery [Analog/ Power] (b), 1 Ground [Analog/Power] (0 (zero)) and 2 Resistors [Passive Components/Resistors] (r). *Note: every analog circuit must have a Ground and every node in the circuit must have a DC path to ground.* Use the Wire Tool to wire the circuit together.

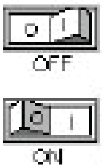
5. Select "Analog Options..." from the Options menu. Under the section "Analysis data saved in RAW file", select the third option, "Node V, Supply I, Device I, Device P". This option will allow you to make current and power measurements with the Probe Tool. Click on the OK button to exit.



6. Click on the ON/OFF switch located in the top right corner of the screen to switch on power to the circuit and start the simulation. An interactive SPICE simulation window will be displayed during the SPICE data collection process showing the progress of the simulation. When the SPICE data collection process has completed, the Value Window will be displayed.

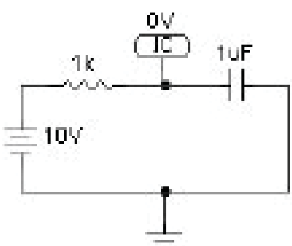
7. Click on the wire connected to the + terminal of the battery with the tip of the Probe Tool. The DC voltage at that node (+10V) will be displayed in the Value Window. Click on the wire connected between the two resistors. The DC voltage at that node (+5V) will be displayed in the Value Window. SPICE data is not collected for the Ground node in the circuit; it is always at zero volts. Click on the + pin of the battery or on one of the resistor pins. The current through that device (5mA) will be displayed in the Value Window. Click directly

on one of the resistors. The power dissipated by that resistor (25mW) will be displayed in the Value Window.



8. Click on the ON/OFF toggle switch again to stop the simulation and return to editing mode.

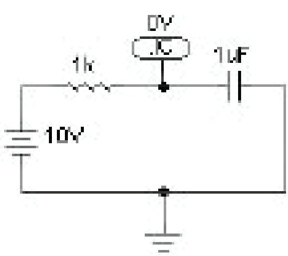
   Now, let's replace one of the resistors with a capacitor to create a simple RC circuit where we can see the charging of the capacitor. Transient Analysis begins its simulation in a stable DC condition where the capacitors are already charged. Since we want to see the capacitor charging from time zero, we must set the initial condition of the capacitor to 0V:



9. Delete the second resistor (the one connected to ground) and the wire leading to it and replace it with a Capacitor [Passive Components/Capacitors] (c).

10. Select .IC device [Analog/ SPICE Controls] (I) and connect it between the resistor and capacitor. This will set an initial condition of 0V on the capacitor for the analysis.

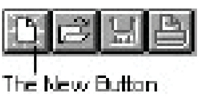
11. Run the simulation again. This time the Transient Analysis window (displays data like an oscilloscope) will be displayed. Click in the Transient Analysis window to select it, then click with the tip of the Probe Tool between the resistor and capacitor. You will see a diagonal line across the scope. This is actually the beginning of the charge curve for the capacitor. Our view of the curve is limited by start and stop times of the Transient Analysis that were selected by default. We now have the option of changing the Transient Analysis settings to increase the size of the time segment that we can view with the scope, or we can reduce the component values so the capacitor will charge quicker. For this example, we will change the component values.



12. Stop the simulation. Double-click on the Resistor to bring up the Edit Device Data dialog box. Change the Label Value from 1k to 100. Click on the OK button to exit.
13. Double-click on the Capacitor. Change the Label-Value from 1uF to .001uF. Click on the OK button to exit.

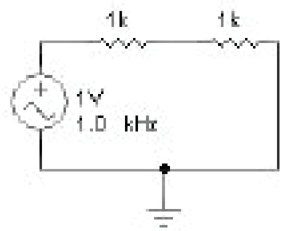
14. Run the simulation again. This time you will see the charge curve of the capacitor.

    Now let's create a simple AC circuit using a Signal Generator and 2 Resistors:



15. Click on the New button in the Toolbar. An "Untitled" circuit window will be opened.

16. Draw the circuit as shown, using the following devices. Select 1 Signal Gen [Analog/Instruments] (g), 1 Ground [Analog/Power] (0 (zero)) and 2 Resistors [Passive Components/Resistors] (r). Use the Wire Tool to wire the circuit together.

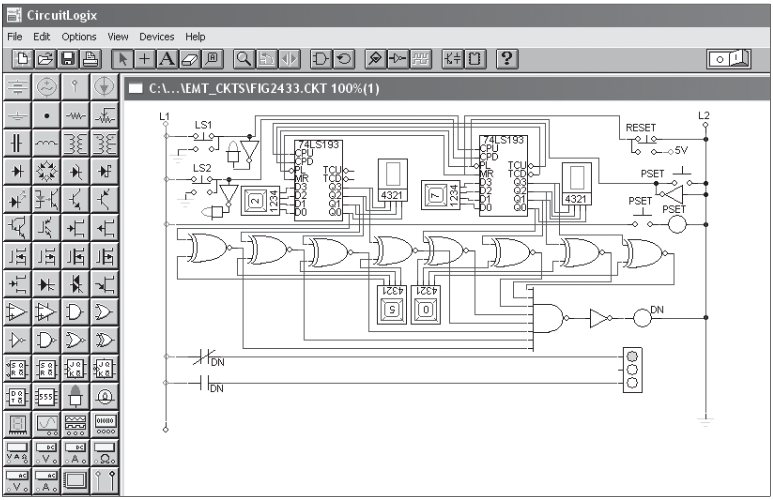


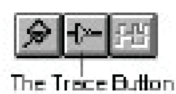
17. Run the simulation. Click in the Transient Analysis window to select it, then click on the wire connected to the output of the Signal Generator. The sine wave will be displayed on the scope. Hold down the SHIFT key and click on the wire connected between the two resistors. A second waveform will be displayed on the scope.

18. Stop the simulation.

1-3 CircuitLogix Digital Simulation

Industrial automation is almost exclusively a digital world. While there are some analog input and output control devices used in automation, these devices quickly have their signals converted to digital in order to be processed, transmitted and stored.

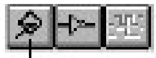




CircuitLogix digital simulation is completely interactive, meaning that the circuit responds immediately to changes from input stimulus, and the operation of the circuit is shown as it happens right on the screen. Operation of the circuit can be observed in several ways:

INDUSTRIAL AUTOMATION **20**

- With the CircuitLogix "Trace" feature enabled, the state of every node in the circuit is shown simultaneously as the simulation runs. In this mode, wires at a logic one are shown as red, wires at a logic zero as blue, and wires at an unknown or tri-state as green (these colors may be changed with the "Select Colors…" command in the Options menu).



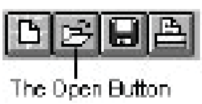
- Any number of SCOPE probes can be connected to any nodes in the circuit, so that the timing diagrams for those nodes are shown in a separate digital Waveforms window. The timing information is updated continuously to show changes as they happen.

- Circuit operation can be observed by connecting any of a variety of displays and noting the conditions shown on them.

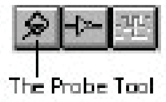
- You can use the Probe Tool to probe any wire in the circuit either during simulation or after it has been stopped. The logic states seen by the Probe Tool can also be charted in the Waveforms window.

Digital Simulation Tutorial

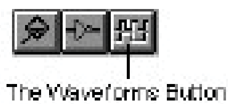
The best way to see how the digital simulation works is to load an example circuit and try some of the available commands. First, launch CircuitLogix by double-clicking on its icon. After a few seconds CircuitLogix will load into memory and will then be ready to use. To experiment with a pre-built circuit, follow these steps:



1. Click on the Open button in the Toolbar.
2. Select the file "SIM.CKT" from the list of available circuits. This circuit contains several mini-circuits and is useful for demonstrating CircuitLogix digital simulation features.
3. Click on the ON/OFF button in the Toolbar to start the simulation. The fact that the simulation is running will be evident because you will see a Hex Display showing a count sequence.



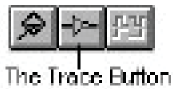
4. Select the Probe Tool from the Toolbar and touch its tip to the wire just to the left of the label "Probe Wire To the Left". If the Probe Tool shows a small triangle pointing down, the wire is at a zero state, if it shows a triangle pointing up, the wire is at a one state, and if the probe indicator is empty, this would indicate an unknown or tri-state.



5. Move the tip of the Probe Tool to the Logic Switch labeled "Toggle Switch" and click near its center. The Logic Display connected to the output of this mini-circuit should then start to toggle on and off rapidly.

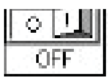
6. Click on the Waveforms button in the Toolbar to open the digital Waveforms window. Each node in the circuit that has a SCOPE attached is charted in this window.

 INDUSTRIAL AUTOMATION 23



Select "Scope Probe" from the Options menu. A new waveform titled "Probe" will be displayed in the Waveforms window. Watch what happens to this waveform as you move the Probe Tool around the circuit.

8. Click on the Trace button in the Toolbar to see the state of every wire in the circuit as it changes.



9. Click on the ON/OFF button again in the Toolbar to stop the simulation.

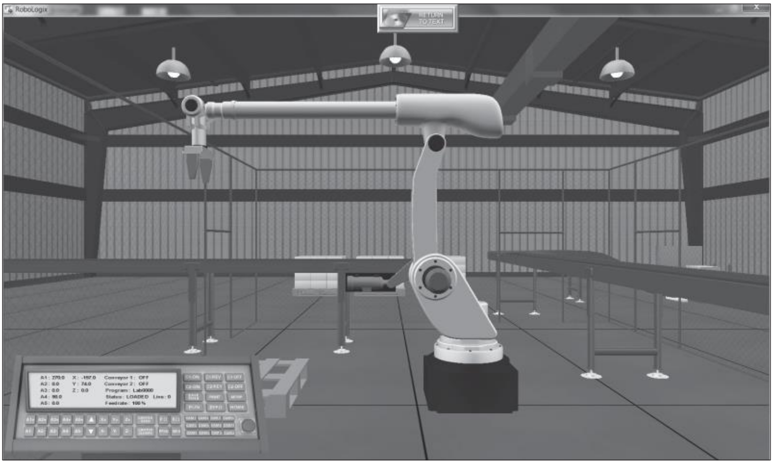
INDUSTRIAL AUTOMATION 24

1-4 RoboLogix

RoboLogix is a robotics simulation software package that is designed to emulate real-world robotics applications. With RoboLogix, you teach, test, run, and de-bug programs that you have written yourself using robot arms and end effectors in a wide range of practical applications. These applications include pick and place, palletizing, welding, and painting and allow for customized environments so that you can design your own robotics application. With RoboLogix, the user can run the simulator to test and visually examine the execution of robot programs and control algorithms, while plotting instantaneous joint accelerations, velocities and positions.

RoboLogix introduces the concepts of multi-axes robots and demonstrates how they can be used in a plant or manufacturing system utilizing 3D simulation . The primary focus of the software is on automated manufacturing processes and allows for the programming, testing, and debugging of robot programs. Users gain practical, "hands on" programming of an industrial robot through a combination of teach-pendant programming, and 3D simulation environments. It is a powerful learning tool that provides students with access to robotic equipment worth tens of thousands of dollars.

The default work envelope for RoboLogix is shown in the image below. This scene consists of two conveyors, workpieces (boxes), pallets, and the control panel. The robot arm used in this environment will perform pick-and-place, point-to-point, palletizing and de-palletizing and a range of instructions to demonstrate programming and de-bugging operations.



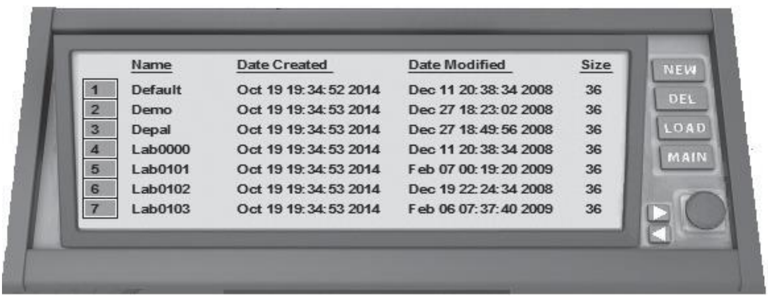
The control panel consists of both robot control instructions as well as environment control functions such as conveyor controls, on-off hardwired control, etc. Various program command keys are also included on the control panel. These keys are used to write, edit, store, recall, and execute robotic programs.

In addition to the default Main Screen displayed on the Control Panel, there are four additional screens that are displayed when programming with RoboLogix. One of these screens is for displaying position registers, variable registers, text-based messages, and the actual program code. The Control Panel has four buttons to access the position register screen (PRS), the variable register screen (VRS), the programming screen (PRG), and the text information screen (INFO).

The Programming Screen (PRG) is the display screen where RoboLogix programs are entered, edited, saved, and retrieved. Each line of program code can be edited, by clicking on the line to launch a dialog box with various edit functions. A new line of

 INDUSTRIAL AUTOMATION 26

code can be started by using the *New* instruction, which will add a new line immediately below the last line. The Programming Screen displays up to 7 lines of code at a time. When a program is longer than 7 lines, subsequent lines of code would be displayed on the next program screen which is accessed by the right arrow key located beside the emergency stop button.



RoboLogix provides 12 viewpoints, or camera angles, for a given robot work envelope. These viewpoints are accessed by the nine CAM keys and allow for the viewing from a variety of angles and perspectives. By using these camera viewpoints, the user can move around in a 3D world in much the same way they would in the real-world. When programming the robot, the camera views allow you to make fine adjustments to the arm and gripper position, or to view the entire work envelope and surrounding area.

RoboLogix uses a scripting-type programming language that uses a command, or instruction set, that is common among major robot manufacturers. Like most robot programming languages, RoboLogix programs consist of data objects and program flow. The data objects reside in registers and the program flow represents the list of instructions, or instruction set, that is used

INDUSTRIAL AUTOMATION 27

to program the robot. Programming languages are generally designed for building data structures and algorithms from scratch, while scripting languages are intended more for connecting, or *gluing*, components and instructions together. Consequently, the RoboLogix instruction set is a streamlined list of program commands that are used to simplify the programming process and provide rapid application development.

The RoboLogix instruction set contains 16 commands, which are usually written as a program on a line-by-line basis. These commands are used to instruct the robot to perform tasks such as moving to a specific location, picking up an object, executing a subroutine, waiting, etc.

INDUSTRIAL AUTOMATION 28

1-5 RoboLogix Tutorial

i) Load and run an existing program

1. Press [ ] key, which displays all programs stored in memory.

2. Select a program on the display screen by clicking on the number button to the left of the program name. A red highlight bar will display the selected item.

3. Press the LOAD key to load the program.

4. Return to main screen MAIN, press [ ] key to run program.

ii) Edit an existing program

1. Press [ ] key, which displays all programs stored in memory.

2. Select a program on the display screen by clicking on the number button to the left of the program name. A red highlight bar will display the selected item.

3. Press the LOAD key to load the program.

4. Select a line of the program to be edited by clicking on the line number box located to the left of the program line.

5. Click on the load LOAD button to load the program line for editing

INDUSTRIAL AUTOMATION 30

6. An edit display box should now appear.

7. Change the numeric values in the display box by +20%, and click OK in the display box.

8. Return to main MAIN screen press RUN key to run program.

iii) Write a program.

Objective: Write program that will pick up a box from Conveyor 1 and move it to Conveyor 2.

1. Press key, and key to create new program. Type the word "test" in the dialog box to name the new program. Click "OK".

2. Enter the first line of the program by pressing the NEW key and selecting the HOME instruction.

   This first line of the program will instruct the robot to begin from its Home position.

3. Add a program line to start Conveyor 1: CONV 1

   Note that the default setting for Conveyor 1 is in the forward (fwd) position. If it is not in this position, use more mouse to select it.

4. Add a program line to instruct the robot to wait 18 seconds before moving towards box NEW WAIT

   Note. This Wait time is required to allow the box to travel from the chute to the stop point on the conveyor.

INDUSTRIAL AUTOMATION 32

5. Move gripper towards box in angular direction until it is above the box (1st pos. point) NEW | A P[] deg/sec

   Enter the number "0" *Hint: Use A1+ key to swing the arm in an angular direction towards Conveyor 1.*

6. Move gripper down to box: L P[] cm/sec

   Enter the number "1" (2nd pos. point)

7. Close the gripper to grasp box: NEW | GRIPPER
   Select "Close"

8. Move gripper to point directly above box L P[] cm/sec

   Enter the number "0" Note. This "retract" point for the box is the same as the "entry" point.

9. Move gripper towards Conveyor 2 until it is at a point above the conveyor (3rd position point): NEW | A P[] deg/sec

10. Move gripper down towards Conveyor 2 NEW | L P[] cm/sec
    (4th pos. point)

11. Open gripper to release box: NEW | GRIPPER
    Select "Open"

12. Move gripper up from Conveyor 2:

13. Return to Home position: HOME Now that the program code has been written. The next steps involve setting the position points by moving the robot using the "jog" keys.Now that the program code has been written. The next steps involve setting the position points by moving the robot using the "jog" keys.

Click on the Main to return to the Main Screen. Switch on Conveyor 1 to send a box into the work envelope. Verify that the robot is in the Home position by clicking on the Home button. HOME

Wait until the box comes to a stop on Conveyor 1. Press and hold the A1+ key to rotate the robot arm towards Conveyor 1. Use the various camera angles to verify that the end effector is directly above the box on Conveyor 1. Once you have determined that the gripper is above the box, press the Save Position key SAVE POSN , to store the 1st position point.

Move the gripper directly down towards the box using the Y-axis key. Y+ Use the various camera angles to verify that the gripper fingers are properly positioned to pick up the box. Hint: The exact location where the box can be picked up has an X coordinate of -40 and a Z coordinate of 155. Click on the Save Position key SAVE POSN to store the 2nd position point.

Close the gripper to pick up the box using the instruction. GRIPPER CLOSED

 INDUSTRIAL AUTOMATION 34

Press the Y+ key to lift the box up from the conveyor. Move the

box to the 3rd position point by pressing and holding the A1+ key. Once the box is positioned directly over the conveyor, press the SAVE POSN key to save the 3rd point.

INDUSTRIAL AUTOMATION 35

Lower the gripper towards Conveyor 2 by pressing the Y+ key.

Press the GRIPPER OPEN key to drop the box onto Conv. 2.

Press C2-ON key to move the box out of the work envelope.

Press HOME key to return to the home position.

Verify that the four position points have been entered by clicking on the P[] key to display the position registers. Position

Registers P0 - P3 should all have data.

Click on the MAIN key to return to the Main Screen. Verify that the program named "Test" is shown in the Program Title, and press the Run key RUN to start the program.

1-6 PLCLogix

PLCLogix is a Programmable Logic Controller (PLC) simulator that emulates the operation of a ControlLogix controller and RSLogix 5000 software. PLCLogix is an ideal tool for learning the fundamentals of ladder logic programming. It will allow you to practice and develop your programming skills using the industry-standard RSLogix 5000 PLC programming software. It provides users with the ability to write, edit and debug programs written using a tag-based format. RSLogix 5000 uses tags, which is a powerful method of programming PLCs but also more complex. PLCLogix provides an interactive approach to learning and understanding the operation of a sophisticated tag-based PLC in a realistic simulated manufacturing environment.

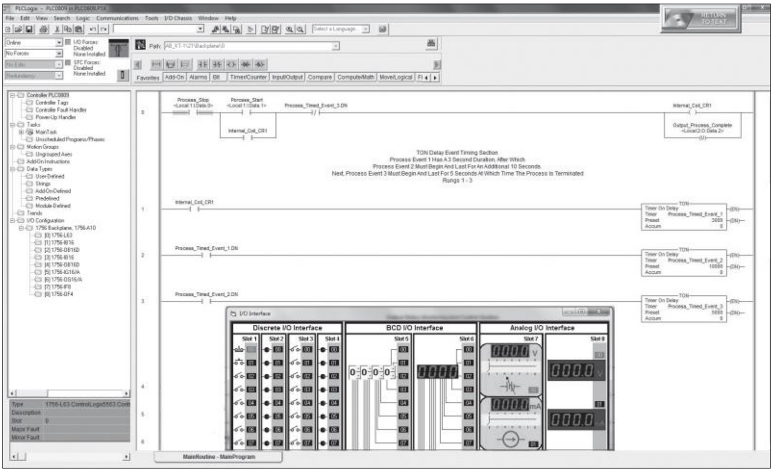
One of the main advantages of using PLCLogix is that it enables you to gain “hands on” experience in the operation of the Logix 5000 PLC. By using PLCLogix, you are able to gain much-needed programming practice by creating and running your own ladder logic programs using tag-based memory. PLCLogix functionality includes a graphical controller organizer and a point-and-click method of configuring various I/O. The application organization is based on using tasks, programs, and routine structures. In addition, it features sophisticated data handling and incorporates both arrays and structures to provide maximum flexibility and emulation of real world control applications. PLCLogix also includes a free-form ladder editor that allows you to modify multiple rungs of logic at the same time. The point-and-click graphical interface provides a simple, intuitive method of entering and editing ladder logic programs.

The PLCLogix Graphic User Interface (GUI) displays the ladder logic, controller organizer, I/O chassis, and a range of control panels. The ladder logic display is the same format as Logix 5000. The controller organizer also follows the same convention as Logix 5000 to provide a seamless transition from PLCLogix simulation to the real-world Logix 5000 control.

Graphic User Interface (GUI)

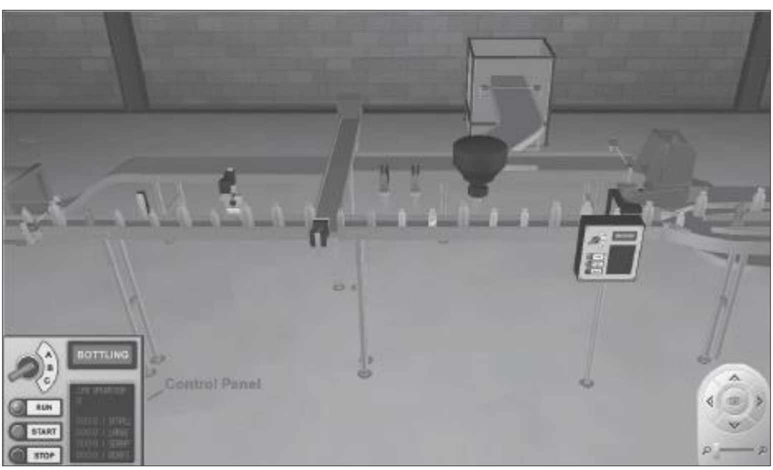
The Graphic User Interface for PLCLogix is designed to emulate RSLogix 5000, with the main difference being the addition of a virtual I/O chassis and a range of 3D simulation worlds. The purpose of the GUI is to provide a range of information displayed on a single screen. This information ranges from bit display to program code to status indicators. The Bit Status is represented by green horizontal bars on either side of the I/O device. The bar on the left is referred to as the rung-condition-in and the bar on the right is the rung-condition-out. When the green bars are illuminated, it indicates a high bit (1) is present in the I/O memory location. If the I/O point is not green, it means that a low bit (0) is present at that address.

The image below illustrates the main components for the PLCLogix GUI. The Menu Bar provides access to a variety of Windows-based functions including Help, Search, and I/O Worlds. The Windows Commands Toolbar contains common Windows-based instructions such as New, Open, Save, Print, Cut, Copy, etc. The Ladder Instruction Toolbar provides category tabs which contain a wide range of instructions in various subsets, or categories. Instructions are inserted into ladder programs by clicking on instructions in the various categories.



3DWorlds

The I/O chassis is a useful simulation environment, due to the very high number and variety of inputs and outputs available. However, one of the main strengths of PLCLogix is its ability to simulate “real world” manufacturing environments. In addition to the I/O chassis, each of the 10 virtual worlds have control panels, such as



INDUSTRIAL AUTOMATION 39

the one shown below, which contain switches, pushbuttons, dials, levers, data input, thumbwheels, etc. The configuration of these control panels varies based on the operating requirements of the application.

PLCLogix simulates real-world control using its “3Dworld” engine and display to create interactive animations. This feature is accessed by the “I/O Worlds” tab on the main menu and interfaces the ladder logic programming with 10 different manufacturing environments including an I/O chassis. The 3Dworld interactive animations graphically simulate process control operations in the following applications: conveyors, compressors, silos, traffic lights, car washes, elevators, bottling lines, batch processes, and robotics.

The I/O worlds featured in PLCLogix create an immersive 3D experience for the user, where they interact with industrial control equipment worth millions of dollars in a safe, controlled environment. The integration of the ladder programs with the 3D worlds provides a unique opportunity for programming in real-time and observing the operation of complex control devices and systems.

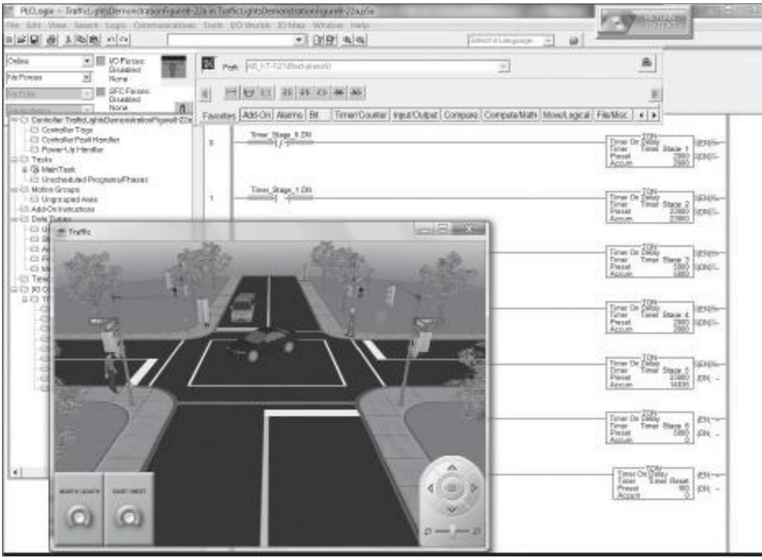
Each of the 10 worlds in PLCLogix have interactive control panels which contain switches, pushbuttons, dials, levers, data input, thumbwheels, etc. The configuration of these control panels varies based on the operating requirements of the application. PLCLogix also includes a camera keypad with five camera views as well as a zoom function for all views. This allows the user to “move” around in the environment in order to gain a better understanding of the PLCs operation in its virtual 3D environment.

 INDUSTRIAL AUTOMATION 40

The realistic 3D interactive animations in PLCLogix allow programmers, electricians, technicians, and engineers to test and troubleshoot ladder logic programs in a "real world" 3D environment, without worrying about incorrect I/O connections and programming errors that could result in damage to the equipment in the 3D worlds. Using simulation tools like PLCLogix, PLC programmers have the freedom to try all the "what-if" scenarios changing ladder logic instructions and programs, then re-running the simulation to see how changes affect the PLC's operation and performance. This type of testing is often not feasible using hardwired operating PLCs that control processes often worth hundreds of thousands – or millions of dollars.

The 3Dworld applications range from batch mixing to production lines and feature bipeds and other avatars that operate in the various worlds. In the traffic light "world" shown in Figure 3-1(b), a ladder logic program controls a traffic light system with "push to cross" pushbuttons, shown in the bottom left corner. The people and cars that are shown would be moving to simulate a realistic world where the traffic control is taking place. The ladder logic program can be modified to change the operation of the "world" and creating an interactive animation where you can perfect your PLC programming and troubleshooting skills.

The image below shows an example of a 3Dworld for a traffic light control system. The box that this world appears in can be moved around on the screen by clicking and dragging with your mouse or by using your finger if it is a touch screen. The 3Dworld can be displayed in full screen by clicking on the top right button beside the "X". You can use your mouse wheel or slider on the camera keypad shown in the bottom right corner of the 3DWorld to zoom and pan the virtual world.



The camera keypad shown below has five camera views as well as a zoom function for all views. This allows the user to “move” around in the environment in order to gain a better understanding of the PLC’s operation. There are five cameras positioned within each manufacturing world: front, back, sides, and top. This allows the user to “walk” inside the world and obtain different perspectives and viewpoints of the manufacturing process controlled by the PLC. Some “eye level” camera views are also provided in the various worlds to provide a perspective from someone standing and interacting in the work environment.

 INDUSTRIAL AUTOMATION 42

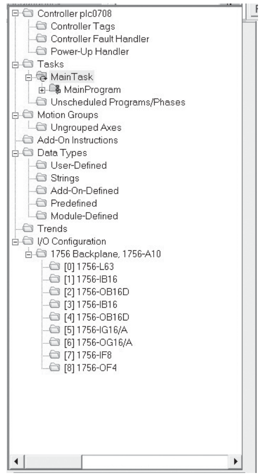
Each I/O world also contains an I/O map which allows you to cross reference the inputs and outputs that are being controlled in the 3Dworld with the I/O points used in the ladder logic control program. The I/O map is selected from the toolbar on the top of PLCLogix and can be accessed for reference at any time. An example of the I/O map used for the traffic light world is shown below.

INPUTS		OUTPUTS	
Local:1:I.Data.0	North/South Crosswalk NO Pushbutton	Local:2:O.Data.0	Northbound Red Light
Local:1:I.Data.1	Northbound Pressure Sensor	Local:2:O.Data.1	Northbound Yellow Light
Local:1:I.Data.2	Southbound Pressure Sensor	Local:2:O.Data.2	Northbound Green Light
		Local:2:O.Data.3	Southbound Red Light
Local:3:I.Data.0	East/West Crosswalk NO Pushbutton	Local:2:O.Data.4	Southbound Yellow Light
Local:3:I.Data.1	Eastbound Pressure Sensor	Local:2:O.Data.5	Southbound Green Light
Local:3:I.Data.2	Westbound Pressure Sensor	Local:2:O.Data.6	North/South Crosswalk Light
		Local:4:O.Data.0	Eastbound Red Light
		Local:4:O.Data.1	Eastbound Yellow Light
		Local:4:O.Data.2	Eastbound Green Light
		Local:4:O.Data.3	Westbound Red Light
		Local:4:O.Data.4	Westbound Yellow Light
		Local:4:O.Data.5	Westbound Green Light
		Local:4:O.Data.6	East/West Crosswalk Light

Controller Organizer

The PLCLogix controller organizer is based on the same format as the RSLogix 5000 controller organizer, and displays tasks, programs, and structures in a tree-like format, similar to Windows Explorer. The controller organizer is a graphical representation of the contents of a project, and makes it easy to see all related information about programs, data and I/O configurations, and to navigate through programs, routines, and tasks. An example of a PLCLogix Controller Organizer is shown below. The main folders in the Controller Organizer tree are: Controller Project Name, Tasks, Motion Groups, Add-On Instructions, Data Types, Trends, and I/O Configuration.

Earlier versions of PLC control software, such as the RSLogix 500, would have a main program and subprogram(s) that would control an application. RSLogix 5000 and PLCLogix have controller organizational models that allow for multiple applications, and each application is known as a *project*. A project holds all of the elements that are contained in an application, including tasks, programs, and routines.



Tasks are an important part of a controller organizer, and provide scheduling and priority information for one or more programs that are executed based on specific criteria. A task is basically a scheduling mechanism used for executing programs. A PLCLogix project can have multiple tasks, and each task can be triggered (executed) continuously, periodically, or when an event occurs.

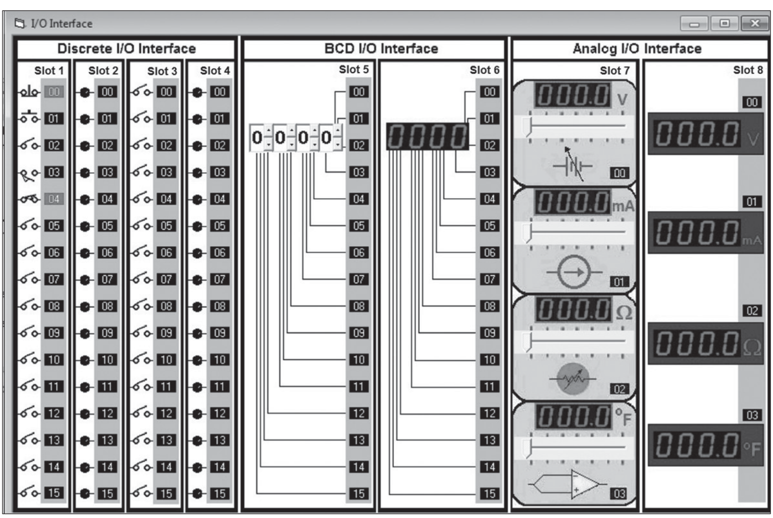
A Continuous Task is self-triggered and automatically repeats. It runs constantly in the background, and when it completes a full scan it immediately restarts. For most applications the continuous task will hold the PLC user-created program. Only one task can be executed at a time so continuous tasks will be executed whenever other tasks are not triggered. When a new project is initiated, a continuous task is created by default. A program does not require a continuous task, and there can be only one continuous task, regardless of number of tasks.

Periodic tasks operate at specific pre-determined intervals and contain program commands that need to be executed on a timed

 INDUSTRIAL AUTOMATION 44

basis. A periodic task performs a function at a specific rate. The time period can be adjusted from 1 ms to 2000 s. Periodic tasks can be assigned a priority level with high priority tasks interrupting lower-priority tasks. Event-driven tasks will execute when a specified event takes place. Tasks in the Logix5000 controller are executed by priority. Continuous tasks have the lowest priority, which is fixed. Periodic and Event-driven tasks have adjustable priority levels. Event-driven tasks are generally used for Axis- and motion-control applications.

Tasks are divided into one or more programs, and each task can operate up to 100 of these programs. Once a task is executed, every program assigned to the task will be triggered in the order they are stored in the controller's memory. A program is basically a set of related tags and routines. Each program consists of tags, a main executable routine and other routines, such as a fault routine. A routine is a set of logic instructions written in a PLC language, such as ladder logic.



 INDUSTRIAL AUTOMATION 45

Each of the simulated discrete input devices are selectable among five different types of devices by right-clicking on the device with your mouse. The five settings available are normally closed (NC) pushbutton, normally open (NO) pushbutton, NO switch, NO limit switch, and NC limit switch.

The Analog I/O interface in the PLCLogix chassis provides a four-channel analog input and a four channel analog output. The analog input signals simulated are voltage, current, resistance, and temperature. Each of the four channels has a default setting of one of these signals. The channels can be further customized by changing the values in the parameter settings window. The analog input and output displays also include a decimal point and linear adjustment for setting precise analog input values ranging from 000.0 to 999.9.

Tags

Logix 5000 and PLCLogix controllers define memory by using variable names, also known as *tags*. Tag-based memory structures are the newest type of memory addressing used by PLCs. A tag is simply another name for a memory location with an assigned data type. For example, Start_PB1 could be a tag name assigned to a start button, instead of something less user-friendly, such as I:1/05.

When a tag is created, it must be given a data type. A data type is a definition of the size and layout of memory allocated for the created tag. Data types define how many bits, bytes, or words of data that will be used by a tag. There are two main data types: basic and structured. The table below shows the data types that are considered to be basic, or atomic. These include Boolean (BOOL), Short Integer (SINT), Integer (INT), Double Integer (DINT) and real

INDUSTRIAL AUTOMATION 46

numbers (REAL). RSLogix 5000 data structures include timers, counters, arrays, messages, and PID. PLCLogix has 32-bit memory locations, which means that a tag will always reserve 32 bits of memory, regardless of whether the data is Boolean or integer. The memory bits shown below indicate 32-bits for each memory type, even though some use less (e.g. Bool only uses one bit).

Data Types	Memory Bits			
	0	1 - 7	8 - 15	16 - 31
Bool	0 or 1	Not Used	Not Used	Not Used
Sint	-128 to 127		Not Used	Not Used
Int	-32,768 to 32,767			Not Used
Dint	-2,147,483,648 to 2,147,483,647			
Real	-3.40282347E38 to -1.17549435E-38 (neg. values) 1.17549435E-38 to 3.40282347E38 (pos. values)			

A tag can also be defined as a compound set of the data types such as a structure or an *array***.** Unlike other PLCs, Logix 5000 and PLCLogix processors do not use indexed or direct addressing. Instead, they use arrays. An array is a type of tag that contains a block of data, and is similar to a data table. Arrays are numerically sequenced tags of the same data type that occupy a contiguous memory location.

An array is basically a table of tags, and is capable of holding the values of multiple tags. It is, essentially, a type of tag that consists of a block of multiple pieces of data. Each individual piece of data in the array is called an *element*. Each element in an array must be of the same type. An array tag holds each element in its assigned order in a contiguous block of the controller's memory. Arrays are useful for indexing applications, when the elements are required to be stepped through (indexed). Arrays can be created in 1, 2, or 3 dimensions. The image below shows an example of a 1-dimensional array which holds 5 different values of pressure ranging from 100 to 140.

Alias Tags

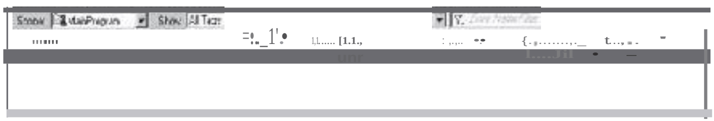
Another type of tag in Logix5000 controllers is an alias tag, which is an identifier for all or part of another tag in the application. Alias tags mirror the base tag to which they refer. In other words, an alias tag is a tag that represents another tag. When the base tag value changes, so does the alias tag. Alias tags allow a user to write a program with tag names assigned to physical I/O points. So, once the system design is complete all that is needed is the mapping of the appropriate tags. Unlike a base tag, an alias tag does not have a defined data-type because it simply assumes the data-type of the tag that it refers to. Aliases are commonly used to assign a descriptive name to an I/O device, or to simplify a name of a complex tag. During the download process, PLCLogix converts program references for alias tags to the physical memory used by the data that the aliases point to. The main benefit of the alias tag is that it allows you to create a new name for a piece of data.

A summary of the tags used in a given program is stored in a tag list. PLCLogix can access the tag list, online in the controller or offline in the software file, and make the tag list available to other software packages (e.g. Excel). With RSLogix 5000, once the software has been configured for the chosen controller, input and output variables are defined in the tag list to establish the link between hardware and software. In PLCLogix, the software is preconfigured and the controller link is already established. The data editors in both systems allow for the creation of a tag by assigning a tag name and defining the data type. The minimum memory allocation for a tag is 4 bytes, or 32 bits. An additional 40 bytes are required for each tag name. Since PLCLogix and ControlLogix are 32-bit controllers, a tag always reserves 32 bits of memory even if it is a Boolean or integer data type.

Each tag is stored individually in the processor, which allows for new tags to be created while the controller is on-line and in the Run mode. Because tags are independent of PLC I/O points in ControlLogix and PLCLogix, it is possible to write a program before any PLC I/O points have even been decided or configured. This allows the PLC programmer to create data within the controller that is structured to suit the needs of the application. Unlike older-model PLCs, where the user has access to a fixed set of either registers or bits, the ControlLogix and PLCLogix processors treat memory much more dynamically. Its versatility allows the user to choose their own names, data types, complex structures, and even arrays - much like how computer programmers might expect to treat memory.

Creating Tags

There are several methods available for creating tags – they can be created one at a time as you write the program, or they can be created in the tag editor shown below. The tag editor contains a spreadsheet-like view of the tags where the tags can be created and edited. When an instruction is first used a “?” will indicated the need for a tag. There are three simple ways to create a tag using the “?” symbol. One method is to either right click or double click on the 7 and select an existing tag from the drop down box. Another method is to double click on the •J• and type in a tag name. If the name does not yet exist, right click on the tag name and select 'Create New Tag Name

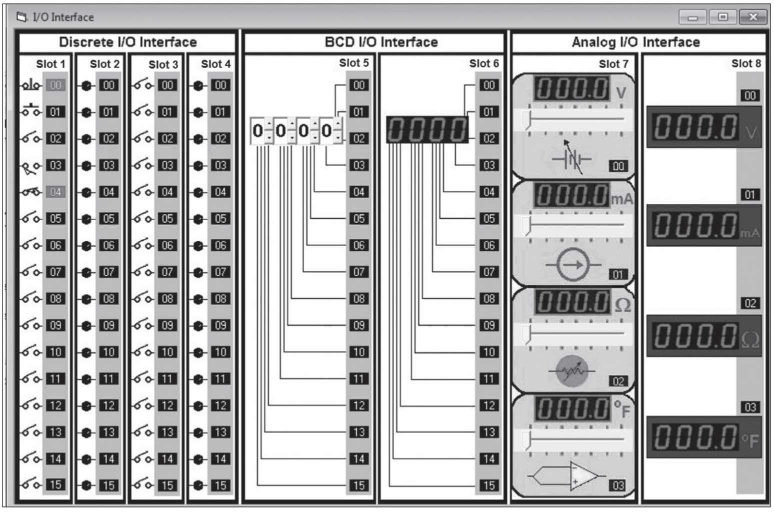


INDUSTRIAL AUTOMATION **50**

1-7 PLCLogix Tutorial

These tutorial exercises are designed to familiarize you with the operation of PLCLogix PLC simulation software and to step you through the process of creating, editing and testing simple PLC programs utilizing the Ladder Logic Instructions supported by PLCLogix.

From the Main Toolbar at the top of the screen, Click on the I/O Chassis tab and select I/O Rack.The simulator screen shown below should now be in view. For this exercise we will be using the Discrete I/O Interface section, which consists of 32 switches and 32 lights. Two groups of 16 toggle switches are shown connected to 2 Input cards of our simulated PLC. Likewise two groups of 16 Lights are connected to two output cards of our PLC. The two input cards are addressed as "Slot 1" and "Slot 3" while the output cards are addressed "Slot 2" and "Slot 4".



Use your mouse and right click on the various switches and note that the type of switch being used can be changed with each click. There are five possible switches that can be used as a discrete input device: normally open pushbutton, single pole switch, normally open limit switch, normally closed pusbutton, and normally closed limit switch. The color of the light connected to the discrete output device module can be changed by right clicking with your mouse on one of the lights and selecting from one of three colors" red, green, or yellow. Close the I/O rack by clicking on the red X in the upper right corner of the I/O rack window.

Tags and Aliases Tutorial

Create a new file by selecting File, New from the Main Toolbar. A window should now appear titled "New Controller". Enter a name for the new controller (i.e. Tutorial 2) and click OK. the Controller Organizer should now appear on the left-side of the screen.

To enter tags and aliases, follow the steps below:

1. Right click on the Controller Tags folder in the Controller Organizer and select *Edit Tags*. The Tag Editor appears, as shown below. The Editor displays the I/O modules (Data Type) and the descriptions for the tag. These descriptions are currently blank since no tag information has been assigned.

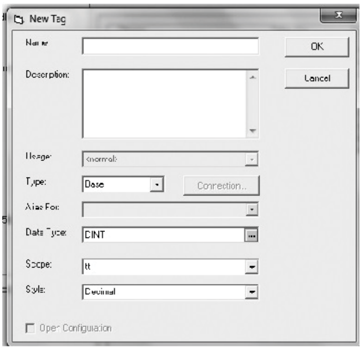
	Name	Alias For	Base Tag	Data Type	Style	Description
	⊞ Local:1:I			AB:1756_DI:I:0		
	⊞ Local:2:O			AB:1756_DO:O:0		
	⊞ Local:3:I			AB:1756_DI:I:0		
	⊞ Local:4:O			AB:1756_DO:O:0		
	⊞ Local:5:I			AB:1756_DI:I:0		
	⊞ Local:6:O			AB:1756_DO:O:0		
	⊞ Local:7:I			AB:1756_IF8_Float:I:0		
	⊞ Local:8:O			AB:1756_OF4_Float:O:0		

 INDUSTRIAL AUTOMATION 52

2. To create a Base Tag, enter the tag data type.

3. To create an Alias Tag, enter the tag which the new tag refers to.

Another method to create a new tag is to use the New Tag dialog function, as shown below. The New Tag dialog is accessed from the File tab on the main toolbar and by selecting New Component - Tag. Using this method, the Data Type selected will automatically select a default Style. You can also manually choose the Style in which you want to display the value of the tag (Hex, Binary, or Octal).



Ladder Logic Tutorial

Follow these steps to enter the ladder logic you will use to define your programs and routines.

1. Create a new File (File, New) assign a name and click OK.

2. Left click on the "+" in the box beside Main Program in the Controller Organizer.

3. Left double-click on the "Main Routine" displayed in the Controller Organizer. The ladder logic window should now be displayed as shown below.

4. From the Ladder Instruction toolbar, click on the tab corresponding to the instruction group from which you want to add an instruction.

5. Select the desired instruction and the instruction is added to the rung or branch on which you chose to put it.

6. Modify the instruction as necessary. Use the Tag Browser to choose a tag.

7. Use the Ladder Instruction toolbar to add additional rungs, branches, branch levels, or instructions as required by your routine.

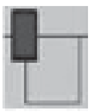
8. Make the necessary modifications to your rung.

9. Right mouse click and choose “Accept Pending Rung Edit”.

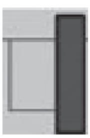
INDUSTRIAL AUTOMATION 54

Branching Tutorial

Add a branch. This icon on the instruction toolbar is used to insert a branch in a ladder logic program. If the cursor is on an instruction, the branch is placed immediately to the right of the instruction. If the cursor is on the rung number, the branch is placed first on the rung.



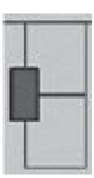
Move a branch level. In order to move a branch level to another location, click on the upper left corner of the branch.



Expand a branch. Click the right leg of the branch, and then drag the leg to the right or left.

Nest a branch. To place another branch structure within the original branch structure, place the cursor at the upper left corner of a branch leg and click on the Add Branch button.

Parallel branch. Place the cursor at the bottom left corner of a branch leg and click on the Add Branch Level button.



Copy branch level. Select the branch level you want to copy by clicking on the left edge and then click Copy Branch Level from the right mouse click menu. Then, click on a rung or instruction in the ladder logic program and click Paste from the right mouse click menu.

INDUSTRIAL AUTOMATION 55

Copy entire branch structure. Select the right leg of the branch structure, and then Click Copy in the right mouse menu. Next, click on a rung or instruction in the ladder logic program and select Paste from the right mouse menu to insert branch structure.

Delete a branch. Place the cursor on the right branch line, click the right mouse button, and then select Delete. Cutting or deleting a branch will delete all instructions on the branch.

Delete a branch level. Place the cursor on the left branch line, click the right mouse button, and then click Delete. Cutting or deleting a branch will delete all instructions on the branch.