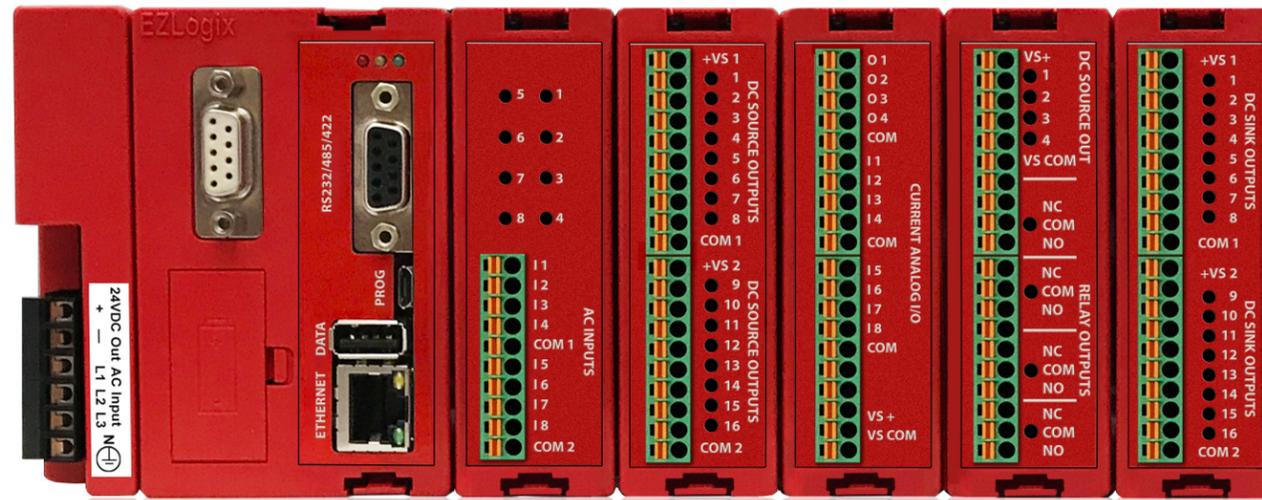


# EZRackPLC™ No Frustration, EZ to Program

# Free PLC Programming Software!!

**Timers, Counters, Scaling, Drum Sequencer, Math Equations, Statistical Functions (avg, min/max, etc.)**



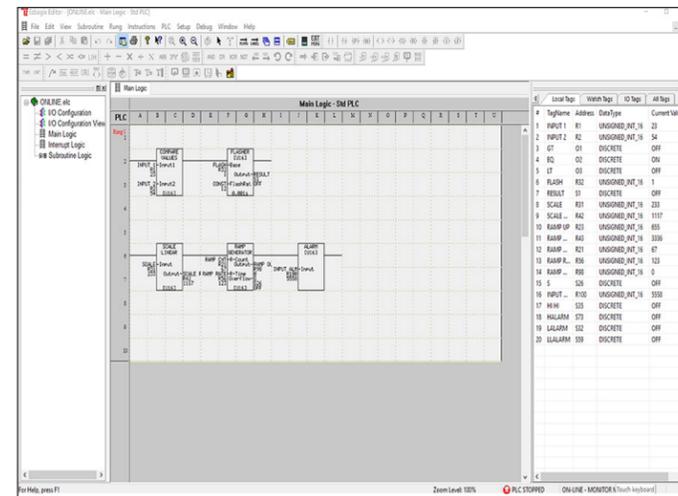
**500,000 Instruction Words User Memory, 16384 Registers**

**16384 Registers**  
**1MB Available for Ladder & Memory**

**2.4 ms Scantime**

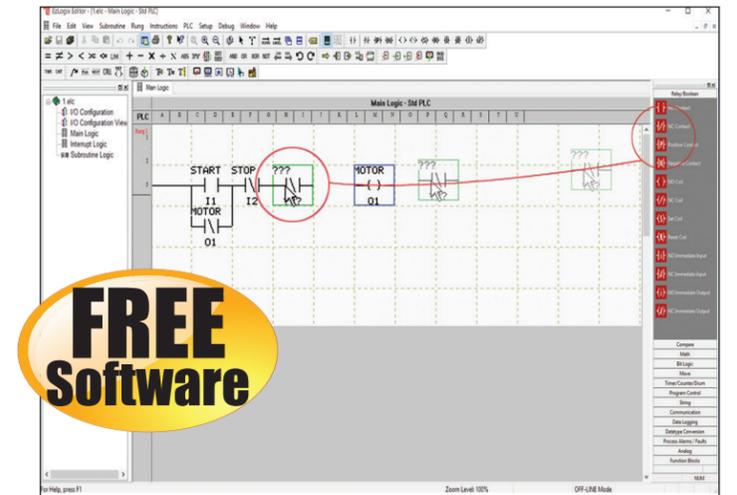


**Advanced Instruction Set & Function Blocks**



**EZLogix has powerful instructions & Advanced function blocks found only in much more expensive PLCs**

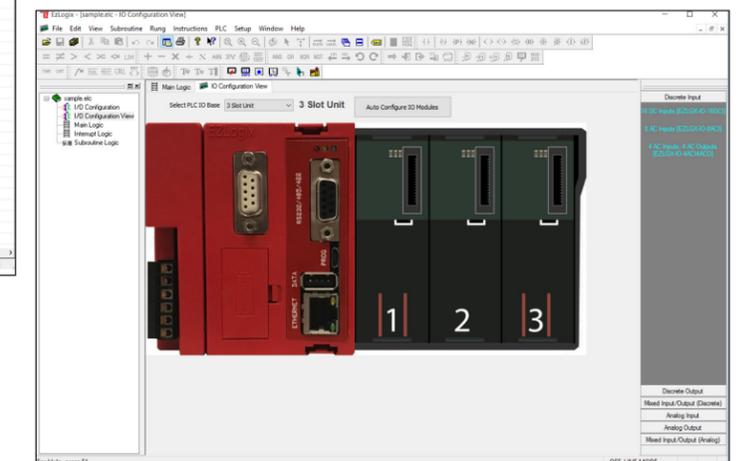
**Simple Drag-n-Drop Ladder Logic Software**



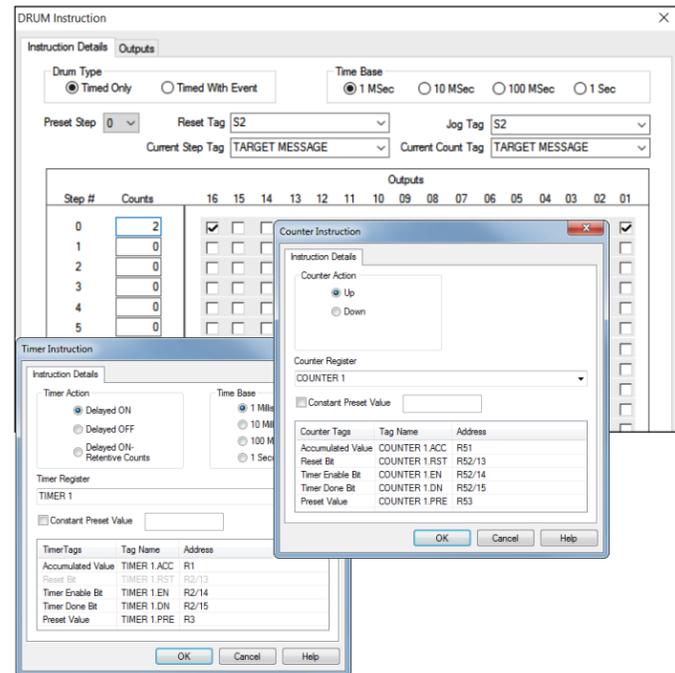
**FREE Software**

- Flexible, powerful and easy to use instructions designed to simplify programming
- Free Flow Ladder Logic
- Each Rung Commented
- User friendly dialog boxes

**Automatic I/O Configuration**



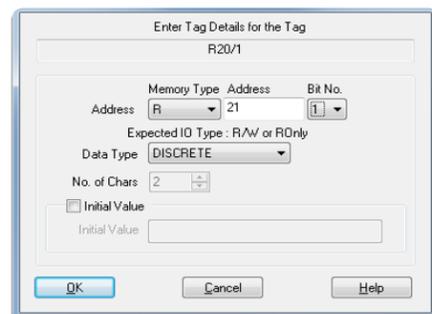
**Connect to the EZLogix & automatically detect your I/O modules and addresses in the local base**



**Built-in Simulator**

**Creates a virtual PLC so you can test your logic without any hardware.**

**Bit Level Addressing**  
**Access each bit within a register**



**Break-point Ladder Logic Execution**

**Debug Ladder Logic by executing rung-by-rung**

# EZRackPLC™ Built-in Simulator

The built-in simulator creates a virtual PLC so you can test your ladder logic without any PLC hardware present.

- Visually see on the virtual PLC, LED indicators light up discrete inputs & outputs based on your ladder logic simulation.
- See register values within simulator to test out proper ladder logic instructions.
- View LED Indicators and Ladder Logic rungs simultaneously in Simulation mode.
- Simulates discrete and analog I/O with access to timers, counters, control bits, etc.
- Force discrete I/O "On" or "Off" to troubleshoot ladder logic
- Break-point debugging while in simulation mode to test certain portions of the ladder program
- Our Windows application uses the same code as the EZLogix CPU firmware for the most accurate simulation.

**Virtually see how your EZLogix ladder program performs for FREE before buying any hardware**

The screenshot displays the EzLogix Designer Pro interface. At the top, the title bar reads "EzLogix Designer Pro - Test1.elc - Main Logic - Std PLC". The main window is divided into several sections:

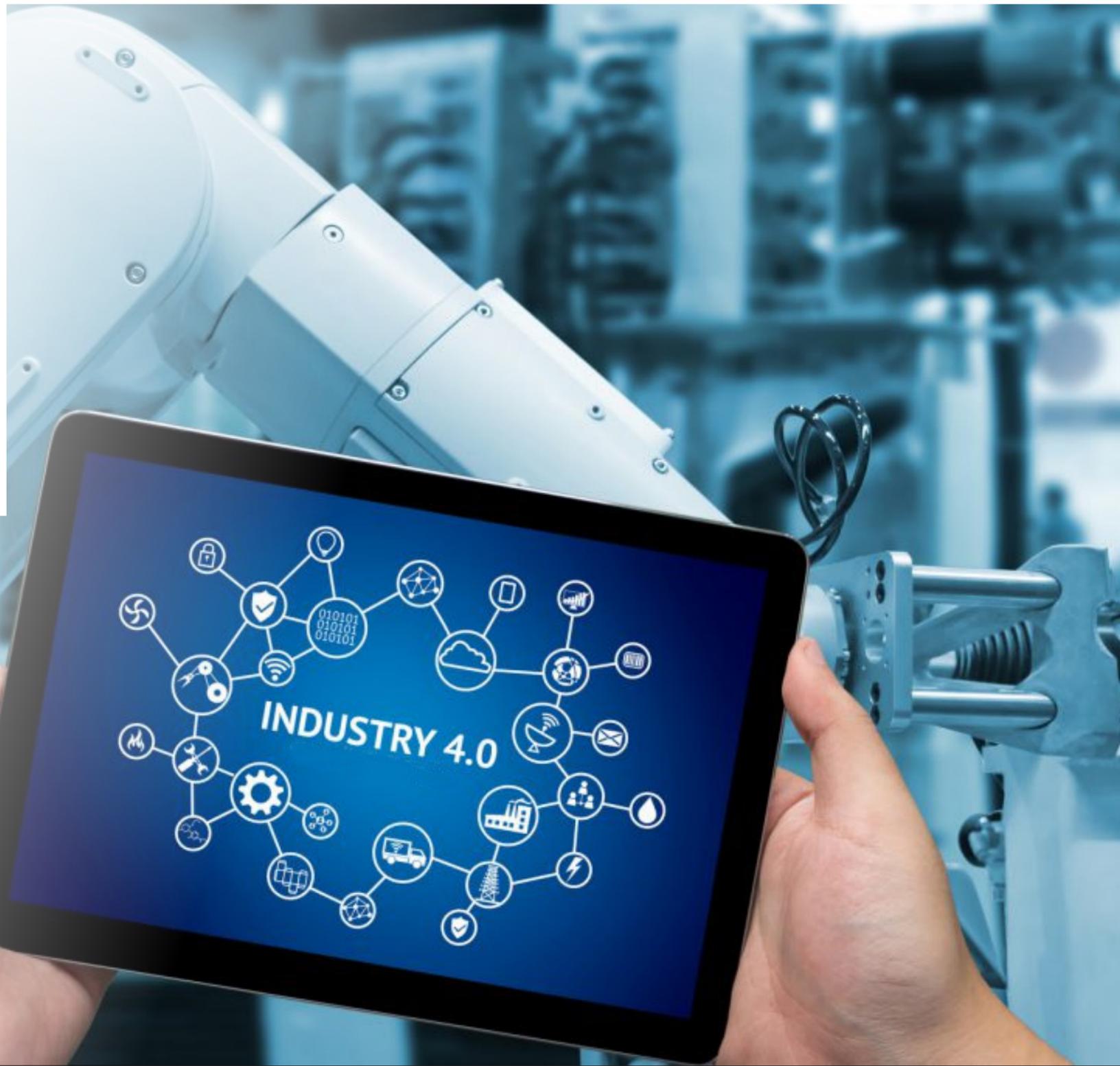
- Left Panel:** A tree view showing the project structure for "Test1.elc", including Main Logic, Interrupt Logic, Subroutine Logic, Hardware Setup, I/O Table Layout, I/O Graphical Layout, Communication Setup, COM Configuration, Ethernet Setup, Database, Tag Database, Tag Cross Reference, Message Database, PID Tuning, PID Setup, PID Monitor, CPU Control, Start PLC, Stop PLC, Transfer to EZLogix, Create USB Loader File, Monitor, Go Offline, Simulate, Switch to Monitor Mod, Debug, Stop Debug, Run Debug, Single Step, and Disable Outputs.
- Center Panel:** "Test1.elc - IO Graphical Layout" showing a virtual PLC rack with three modules: "EZLogix IO-16DCI", "EZLogix IO-16DCOP", and "EZLogix IO-8ANALOGV". Each module has its own terminal block with labels for DC inputs, DC source outputs, and voltage analog I/O.
- Bottom Panel:** "Test1.elc - Main Logic - Std PLC" showing a ladder logic rung. Rung 1 contains a "COMPARE VALUES" instruction (R1) comparing "Input1" (I1) and "Input2" (I2) against a constant value of 49152. This is followed by a "TIMER" instruction (TMR) with a pre-set value of 0.001s and a "FLASHER" instruction (FLASH) with a base of 0 and a flash rate of 0.001s.
- Right Panel:** "Debug/Monitor" window showing a table of tag values. The table has columns for Address, DataType, and Current Value.

Address	DataType	Current Value
I1	DISCRETE	OFF
I2	DISCRETE	OFF
I3	DISCRETE	OFF
I4	DISCRETE	OFF
I5	DISCRETE	FORCED OFF
I6	DISCRETE	OFF
I7	DISCRETE	OFF
I8	DISCRETE	FORCED OFF
I9	DISCRETE	ON
I10	DISCRETE	OFF
I11	DISCRETE	OFF
I12	DISCRETE	OFF
I13	DISCRETE	ON
I14	DISCRETE	OFF
I15	DISCRETE	OFF
I16	DISCRETE	OFF
OR1	UNSIGNED_INT_16	2048
OR2	UNSIGNED_INT_16	950
OR3	UNSIGNED_INT_16	0
OR4	UNSIGNED_INT_16	10
IR1	UNSIGNED_INT_16	2048
IR2	UNSIGNED_INT_16	950
IR3	UNSIGNED_INT_16	0
IR4	UNSIGNED_INT_16	10
IR5	UNSIGNED_INT_16	0
IR6	UNSIGNED_INT_16	0
IR7	UNSIGNED_INT_16	0
IR8	UNSIGNED_INT_16	0
O1	DISCRETE	OFF
O2	DISCRETE	OFF
O3	DISCRETE	ON
O4	DISCRETE	OFF

**Built-in Simulator**  
Creates a virtual PLC so you can test your logic without any hardware.



# EZ RackPLC™ IIoT/ Industry 4.0 Ready





# EZRackPLC™ Data Logging Instruction on USB

- Store up to 64GB of valuable PLC data on EZLogix built-in USB port
- Stay connected from anywhere in real-time with plant production and maintenance data using EZLogix secure IIoT / MQTT protocol for data transfer
- Store real-time PLC data based on tag event and time intervals in .csv format
- Remotely access PLC data using Free EZIIoT utility with integrated message queuing telemetry transport (MQTT) protocol



Log Data to File Instruction

**Instruction Details**

File Name and Size  
 Max 32 char including ext and any appended fields.  
 Tag Based  
 Fixed   
 Append to File Name  Hour (Uses 4 char)  Day (Uses 2 char)  Month (Uses 3 char)  
 File Size Tag   
 Tag shows file size in bytes. Data Saved in CSV Format.

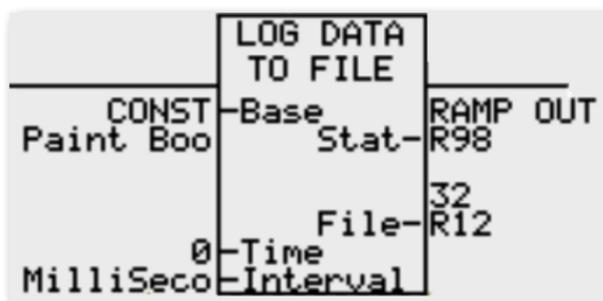
**Log**  
 Log Type:   
 Event/Enable Tag:   
 Log Time-interval:    
 Status Tag:   
 Status value definitions:  
 00:Normal operation (No Errors)  
 02:File open error (USB Drive may not be plugged)  
 04:File write error (USB Drive may be full)

**Select Tags**  
 Tag Names are used as column headers in CSV file. With each record  Log PLC Date  Log PLC Time Decimal Places for Floating Point Tags

Available Tags:			Selected Tags: (4/10)		
Name	Address	Type	Name	Address	Type
INPUT 1	R1	UNSIGNED_INT_16	R1	R1	UNSIGNED_INT_16
R1.ACC	R1	UNSIGNED_INT_16	INPUT 2	R2	UNSIGNED_INT_16
R1.PRE	R3	UNSIGNED_INT_16	FLASH	R32	UNSIGNED_INT_16
RAMP CNT	R21	UNSIGNED_INT_16	RAMP MIN	R43	UNSIGNED_INT_16
RAMP UP	R23	UNSIGNED_INT_16			
SCALE	R31	UNSIGNED_INT_16			
FLASH.RES1	R33	UNSIGNED_INT_16			
FLASH.RES2	R34	UNSIGNED_INT_16			
SCALE RESULT	R42	UNSIGNED_INT_16			
RAMP RATE	R56	UNSIGNED_INT_16			

Buttons: Delete Tag(s), Move Tag Up, Move Tag Down

OK Cancel Help

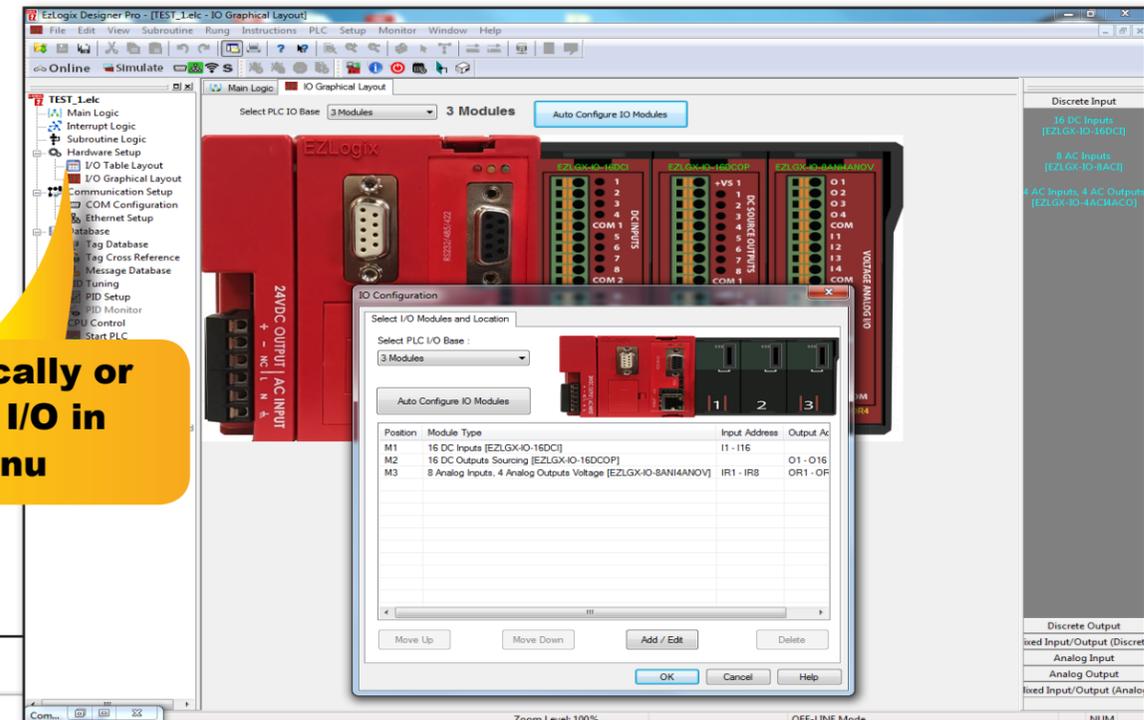


# EZRackPLC™ Automatic I/O Configuration

Configure your EZLogix I/O modules automatically when you are connected over USB, Ethernet, EZ WiFi or Serial programming ports.

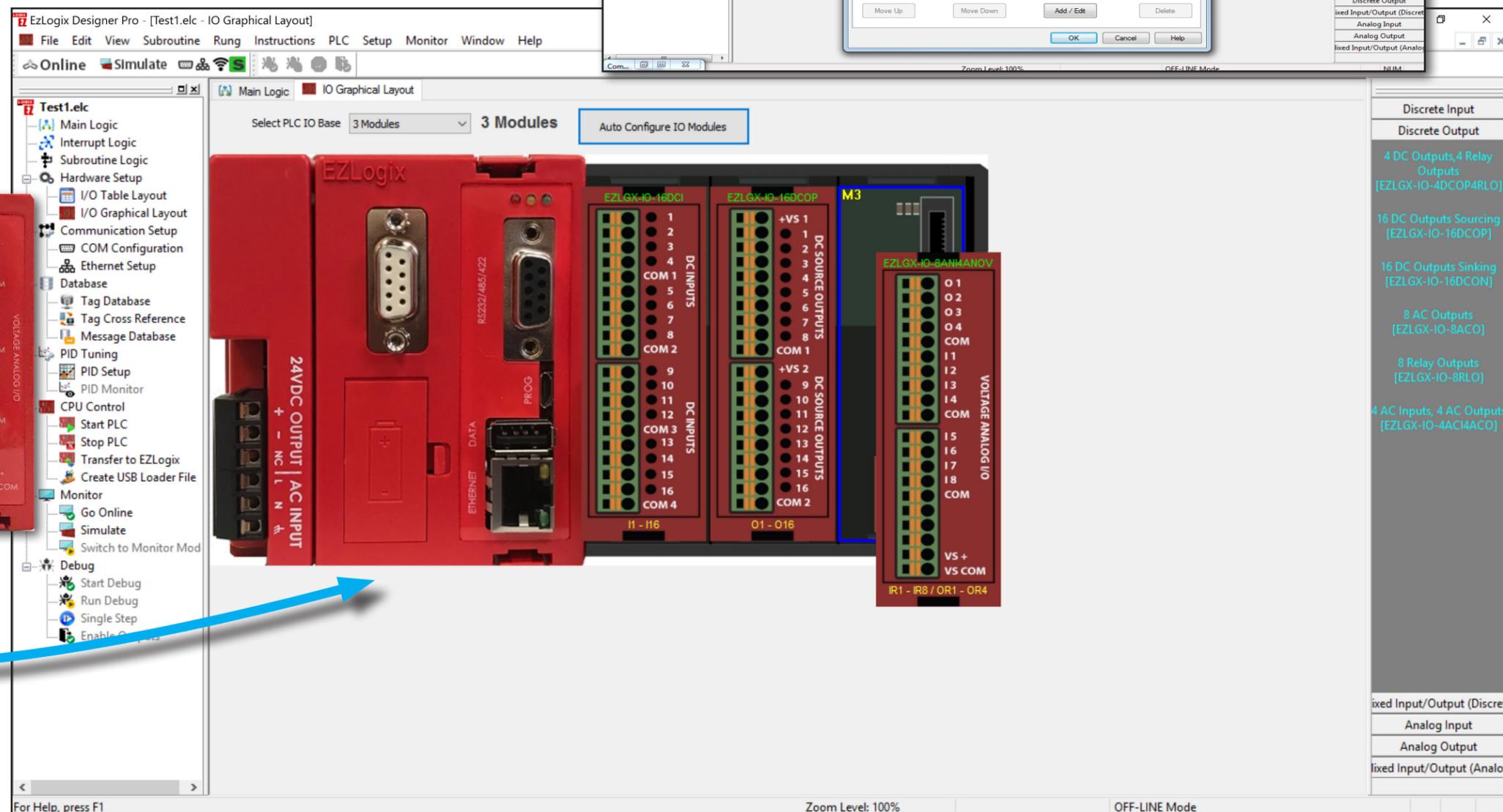
- Auto detect discrete, analog, and speciality modules within EZLogix Designer Pro.
- Automatically assigns respective tag address range.

**Option to automatically or manually configure I/O in hardware setup menu**



## Automatic I/O Configuration

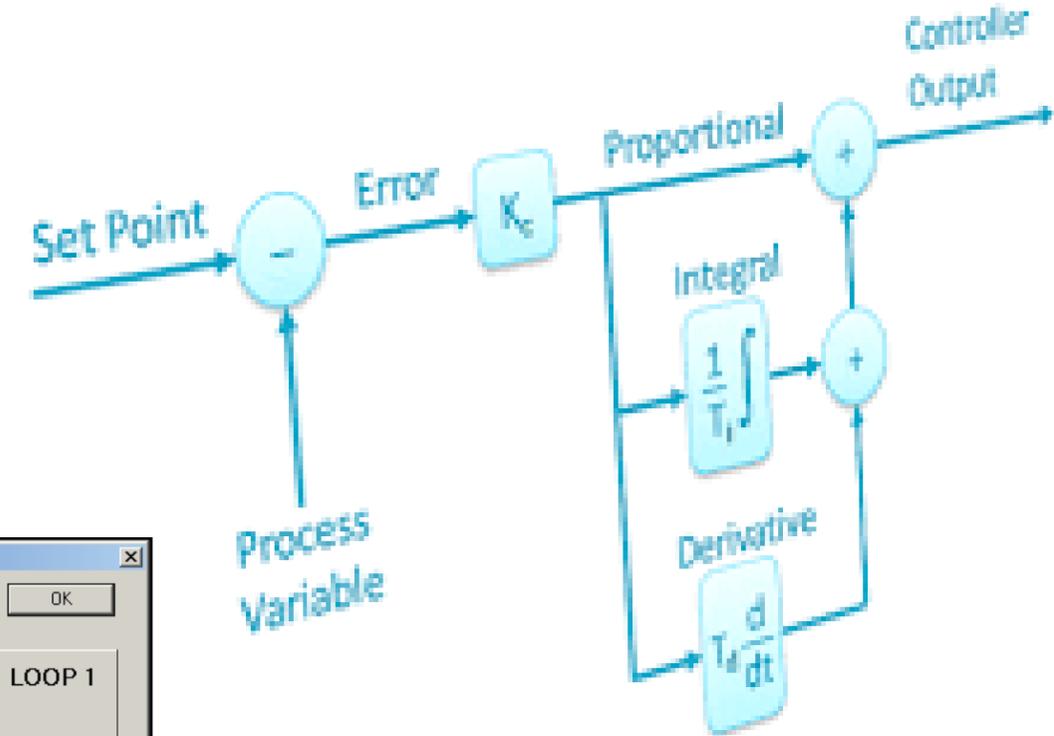
Connect to the EZLogix & automatically detect your I/O modules and addresses in the local base



# EZRackPLC™ Auto-tuned PID Loops

## What is PID?

PID is one of the most popular control algorithms used in the industry to control the variables involved in an industrial manufacturing process for the proper operation of the process. PID stands for Proportional, Integral and Derivative control algorithm. With a proper choice of P, I, and D settings, a user can maintain a process value very close to the setpoint. In addition, if the setpoint changes, the PID algorithm can quickly bring the process back under control. EZLogix supports up to 8 auto-tuned PID loops. For each loop you have to define several parameters, as shown below in the PID Setup window. You may change most of these parameters during run time, using EZLogix Designer Pro in online mode.



## PID Loop Auto Tuning

To achieve a stable and responsive process control, it is very important to select the proper PID parameters. Experienced users can estimate good starting values for these parameters and later tweak them to optimize the PID loop performance. This is called as the manual tuning of the process. Whereas, those who want help in estimating the starting values of the parameters like P, I, and D coefficients, EZLogix provides an Autotune feature.

## PID Monitor

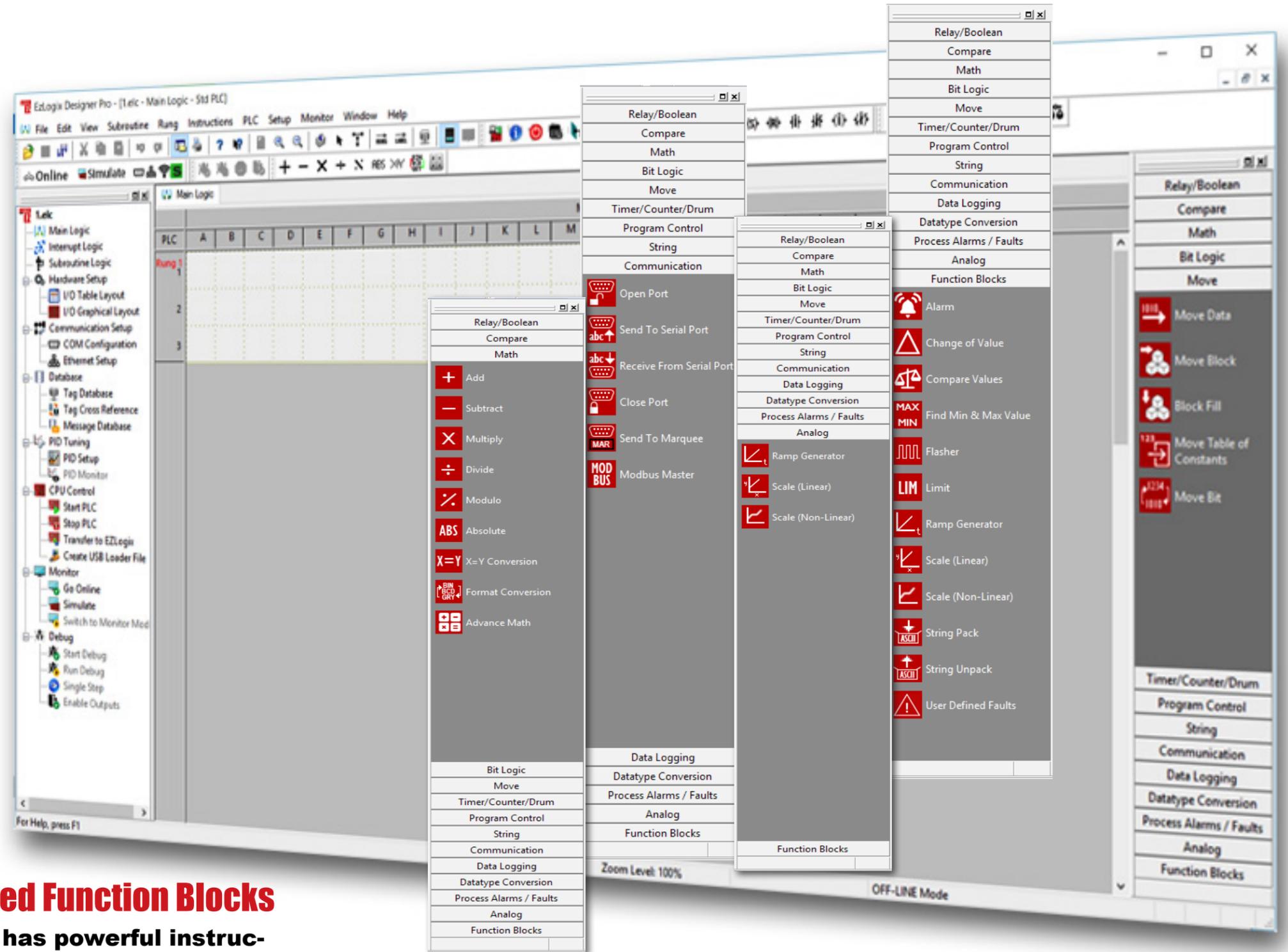
You can use the PID Monitor function to monitor and make real-time changes to your PID Loop. In order to use it, you must be connected to the PLC and select Main Menu > EZLogix > PID Monitor. A PID Monitor window will show up. Here you can change the current values of the parameters by entering a value in the New Value field. Once all of the parameters are defined, press the Apply button and then the Start Monitoring button at the bottom, to begin monitoring your PID Loop. A graph will begin to appear as shown in the image below.

# EZ RackPLC™ Advanced Function Blocks

EZLogix Designer Pro comes with a library of pre-defined function blocks such as scaling, compare, hi/low alarm, averaging, min/max, ramp generator, advanced math and many more, found typically in much more expensive PLCs.

The Free EZLogix Designer Pro software has been designed to provide our customers a more flexible and easy to use PLC programming experience. The EZLogix function blocks will continue to grow with customer requests at no additional costs to upgrade.

- Alarm
- Advanced Math
- Change of Value
- Compare Values
- Find Min and Max
- Flasher
- Limit
- Ramp generator
- Scale (Linear)
- Scale (Non-linear)
- String Pack
- String Unpack
- User defined faults
- IIoT (MQTT Publish)



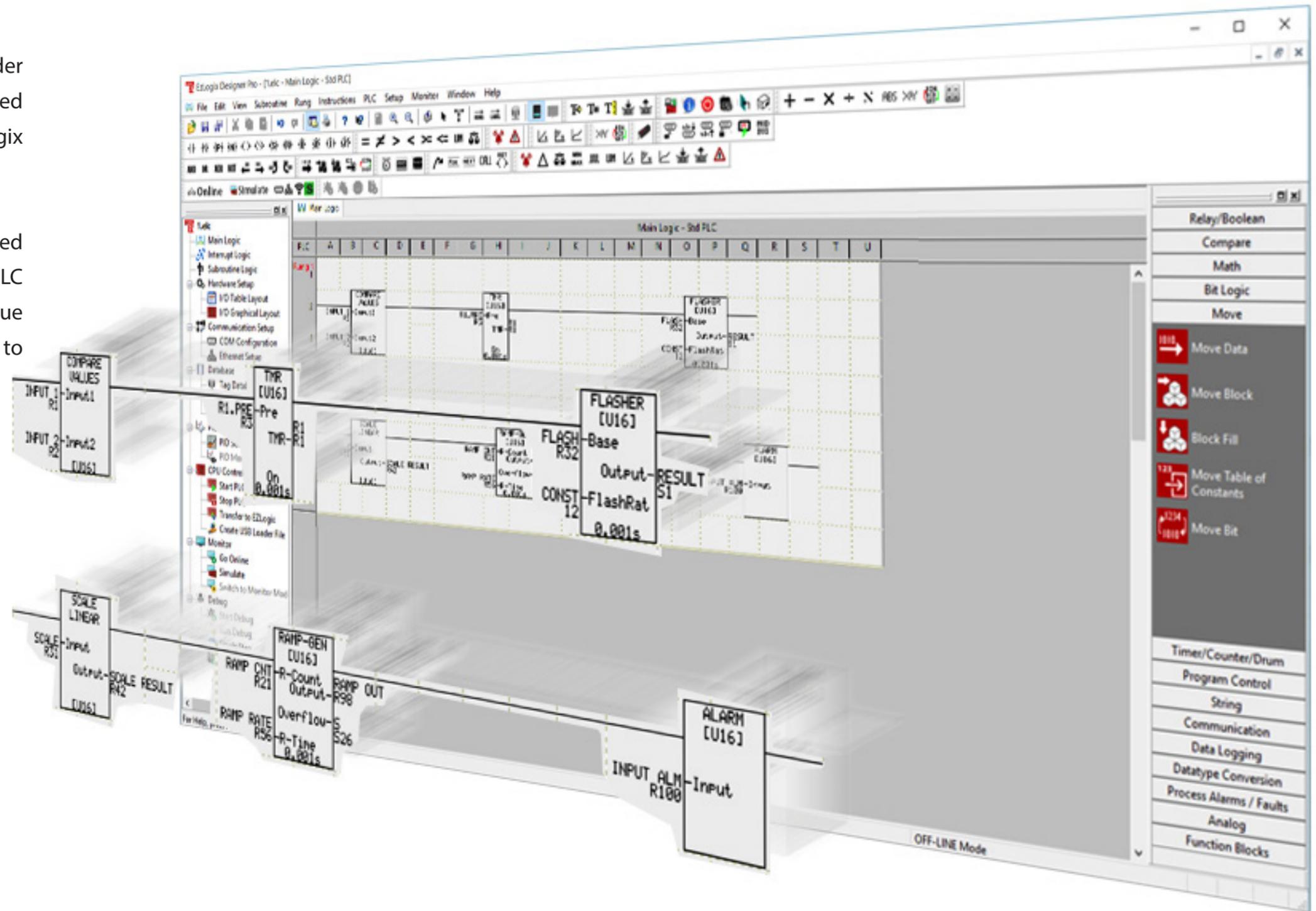
**Advanced Function Blocks**  
**EZLogix has powerful instructions & Advanced function blocks found only in much more expensive PLCs**

# EZ RackPLC™ Rich Instruction Set

With optimized instruction sets whether it be simple ladder relay, boolean, move, bit logic instructions etc... or advanced math instructions for complex algorithm, the EZLogix Designer Pro has it all.

The Free EZLogix Designer Pro software has been designed to provide our customers a more flexible and easy to use PLC programming software. The EZLogix functions will continue to grow upon customer requests at no additional costs to upgrade.

- Relay/Boolean
- Compare
- Advanced Math
- Bit Logic
- Move
- Time, Counter, Drum
- Program control
- String
- Communication
- Data Logging
- Datatype conversion
- Process alarms/ Faults
- Analog
- IIoT



# EZRackPLC™ Force Inputs/Outputs (Great Troubleshooting Tool)

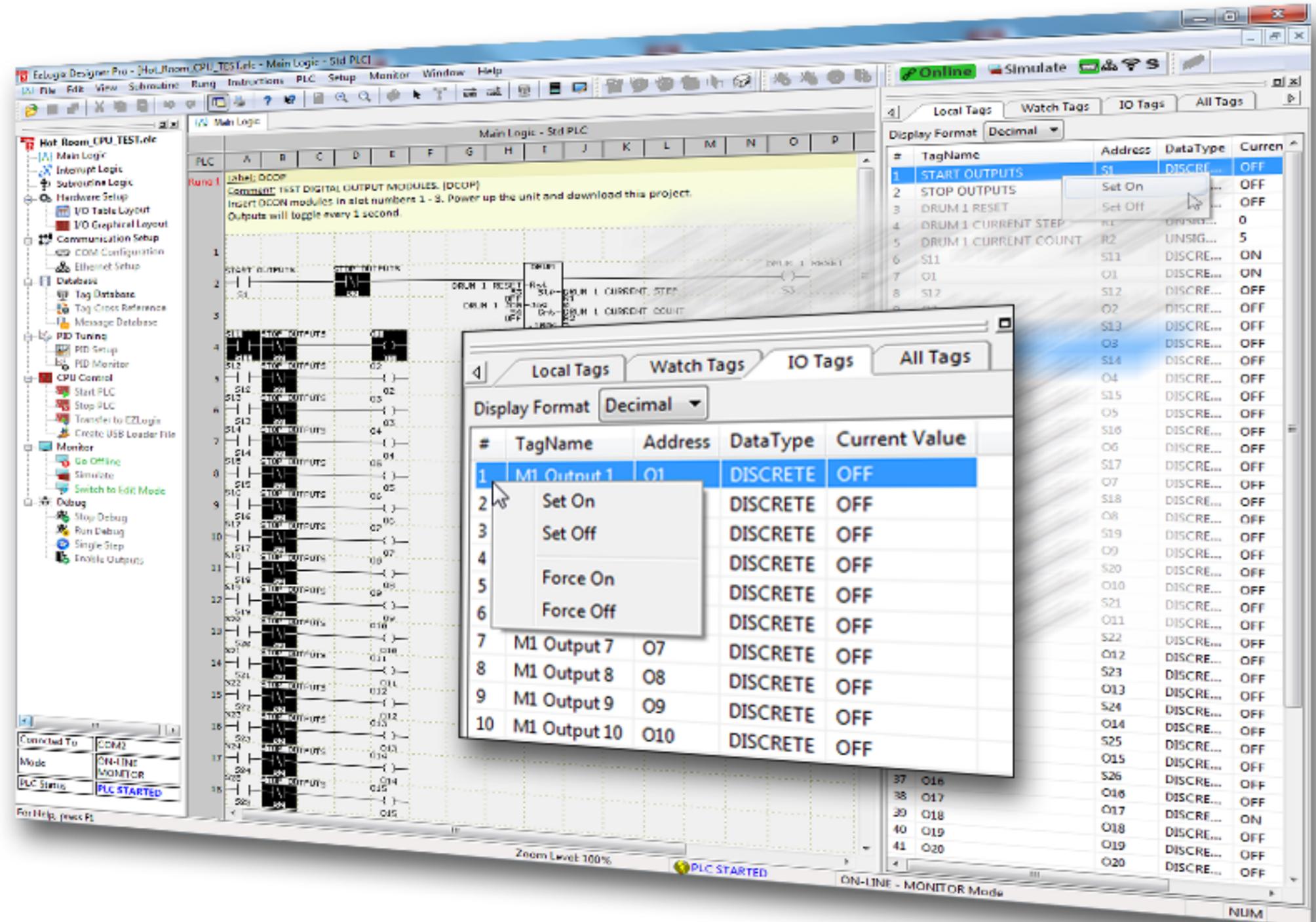
EZLogix CPU supports true Forcing of I/O and internal memory elements. Discrete I/O can be forced to either an ON or OFF state. Analog I/O points can be forced to constant values.

The forcing of numeric and bit memory elements simple means the CPU sets the element to the forced value and does not permit any additional update to the memory element as long as it is forced.

### What is Forcing I/O?

The ability to Force I/O allows you to troubleshoot particular sections of your ladder program by “forcing” a state, in the case of a discrete I/O, or value, in an analog I/O, to make sure you are getting the expected result.

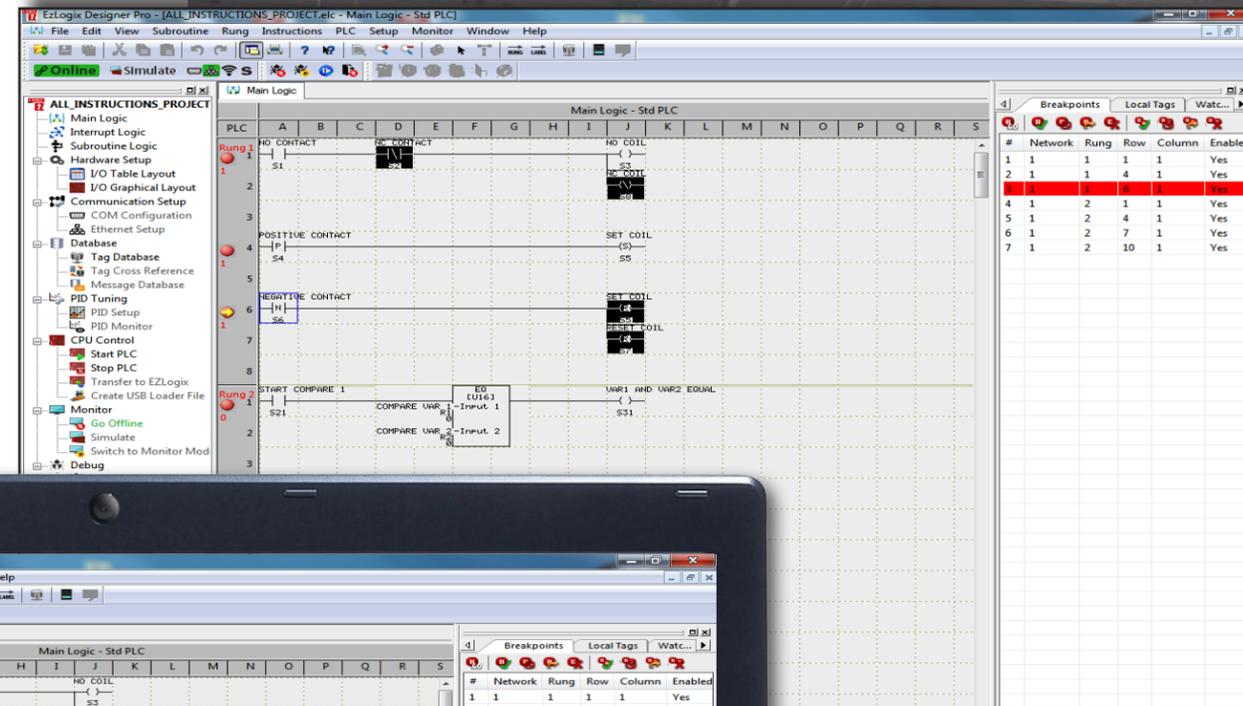
In order to comprehend the benefit of Forcing I/O, one must first distinguish the difference between the physical I/O of a PLC, called the “field side”, and the internal status of that physical I/O within the ladder program, called the “logic side”. In regular operation of a PLC, the status of the physical inputs is copied to the logic side at the top of the PLC scan, and the logic side of the outputs is copied to the fields side at the bottom of each scan. Forcing I/O interrupts the normal processing of the inputs and outputs. Instead, when an I/O is forced, the “logic side” is set to the forced value, and any change in the physical I/O is ignored, and any attempt to change the value or state in the logic is also ignored until the force is released.



# EZ RackPLC™

# Break-Point / Single Step Debugging

EZLogix Designer Pro's Break Point feature is an excellent debugging function to troubleshoot programming errors. Break points can be set at certain positions in the program in order to force an execution stop. At each stop, respective variable values can be examined. Only the tasks that reached the break point are stopped while all other tasks continue to run. The break point feature is ideal for troubleshooting large ladder logic programs that do not have syntax errors but are not performing in the manner intended. It provides the programmer a step-by-step execution of variables in order to see where the potential bugs are in the ladder logic code.



**Single Step Debugging**  
Takes you step-by-step to debug the rung from the start of your break-point

