SuperLogic Pro
User Guide

Document Revision 1.5

(Updated September 19, 2019)

Buford, GA USA

For more information please visit the product web page:

www.vsys.co/smart3g

Contents

License Agreement 3
I. Introduction 4
II. SuperLogic Pro User Interface 5
Ladder Editor 6
Project Explorer Window 7
Toolbox Window 7
Properties Window 8
Debug Window 8
Output Window 8
Device Status Window 9
Tag Database Window 10
Cross Reference 11
III. System Files 12
Integer Register 12
Float Registers 13
Binary Registers 13
Timer Registers 13
Counter Registers 14
Inputs 15
Outputs 15
Control Words 15
Control Bits 17
Barcode 18
IV. Ladder Commands 19
Normally Open Command 19
Normally Closed Command 19
Compare Command 19
Text (Barcode) Lookup Command 20
Output Command 21
Timer/Counter Command 22
Move Command 24
Send Message Command 24
Math Command 24
Reset Command 26

Ladder Subroutine Command 26
V. FAQs 27
Where do I Start? 27
Creating your First Ladder Program 27
Importing a (.prg) Ladder Program 29
Runtime Operation 31
Barcode Lookups 32
Barcode Comparison 36
What do enabled/disabled rungs and contacts do? 38
What are Ladder Subroutines and how do I use them? 39

License Agreement

Before using this software and accompanying software tools, please take a moment to go thru this License agreement. Any use of this software and associated hardware indicate your acceptance to this agreement.

SuperLogic Pro is protected by copyright laws and international treaties. Unauthorized reproduction or distribution of this software may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

It is the nature of all mechanical and electrical systems to be hazardous. In order to be permitted to use SuperLogic Pro with any machine you must agree to the following terms:

I agree that no-one other than the user of this software, will, under any circumstances be responsible for its operation, safety, and use. I agree there is no situation under which I would consider Vital Systems, or any of its distributors to be responsible for any losses, damages, or other misfortunes suffered through the use of SuperLogic Pro. I understand that SuperLogic Pro and associated hardware is very complex, and though the engineers make every effort to achieve a bug free environment, that I will hold no-one other than myself responsible for mistakes, errors, material loss, personal damages, secondary damages, faults or errors of any kind, caused by any circumstance, any bugs, or any undesired response by the software while running my machine or device.

I fully accept all responsibility for the operation of this software and associated hardware and for its operation by others who may use the software. It is my responsibility to warn any others who may operate any device under the control of the software of the limitations so imposed.

I fully accept the above statements, and I will comply at all times with standard operating procedures and safety requirements pertinent to my area or country, and will endeavor to ensure the safety of all operators, as well as anyone near or in the area of operation.

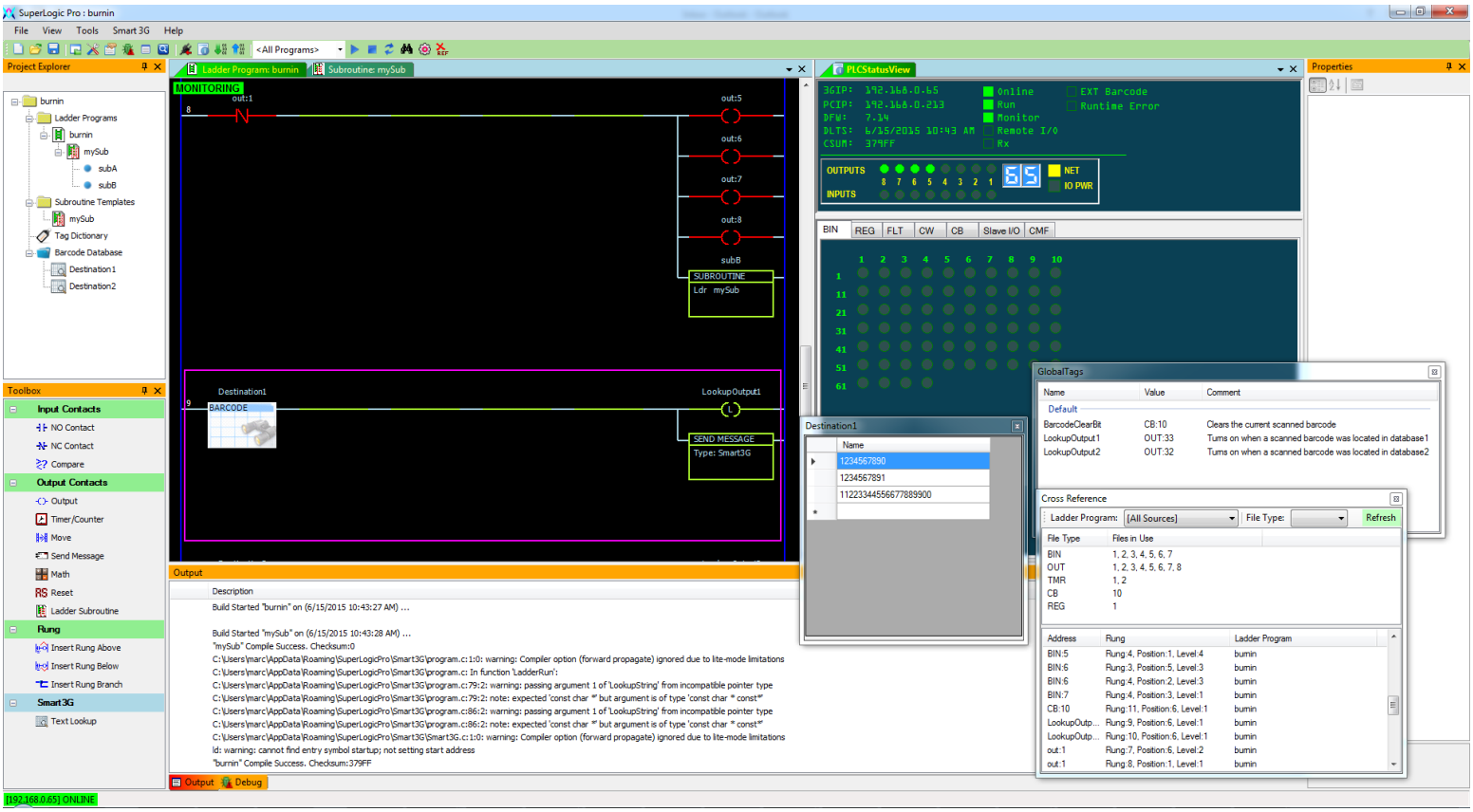
I. Introduction

SuperLogic Pro is a graphical and feature-rich ladder-logic programming environment for the Smart3G cards that is developed to meet industrial programming standards for many specialized applications. Operation-wise, the SuperLogicPro application also serves as the direct interface to the connected Smart3G card.

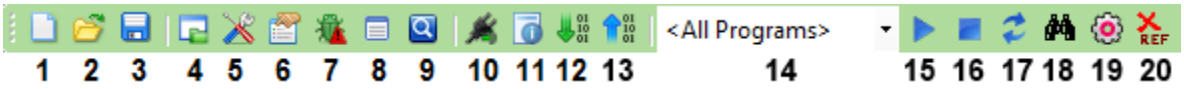
Features:

- Create robust, powerful, and reusable ladder-logic programs and download them to cards.
- Upload and Modify existing ladder-logic programs straight from a Smart3G card.
- Monitor and debug program execution during runtime.
- Temporarily change Ladder Files contents while monitoring without altering rungs.
- Drag-and-drop Graphical User Interface for simple, yet extensive ladder-logic programming.

II. SuperLogic Pro User Interface

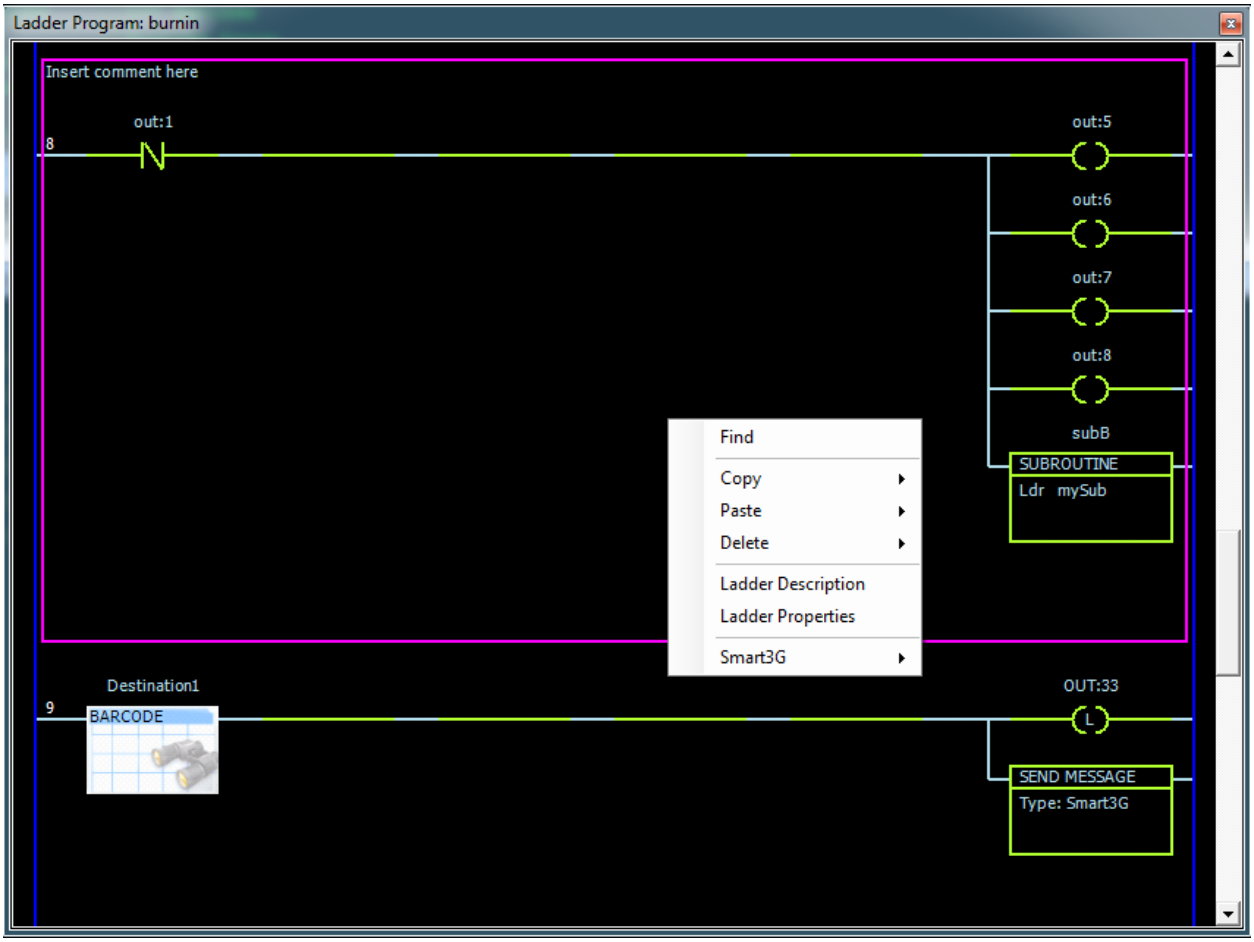


Menu Bar



1. Create New Project
2. Open Existing Project
3. Save Project
4. View Project Explorer
5. View Toolbox
6. View Property Window
7. View Debug Window
8. View Output Window
9. Find
10. Connect to Device
11. View Device Status
12. Install Current Project on Device
13. Upload Project from Device
14. Ladder Program Selector (Affects 14 – 19)
15. Run Selected Ladder Program on Device
16. Stop Selected Ladder Program on Device
17. Reload Selected Ladder Program on Device
18. Monitor Selected Ladder from Device
19. Configure Selected Device
20. View Cross Reference

Ladder Editor



The Ladder Editor is where ladder programs are edited and monitored. This window can be accessed by Double-Clicking a ladder program node under “Ladder Programs” in the Project Explorer, or by creating a new Ladder Program.

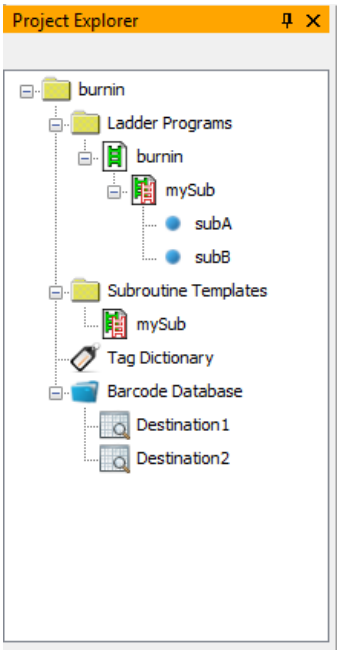
To create a new Ladder Program:

- Right-Click on the “Ladder Programs” node in the Project Explorer.
- Select “Add New”.
- Input a valid Ladder Program Name (this can be changed later on).
- Confirm.

The Ladder Editor for a ladder program is also automatically opened when monitoring is enabled from the Project Explorer, or the Icons Menu Bar.

NOTES: *When modifying a contact, it can almost always be done from the Property Window. It is also possible to copy and paste rungs and contacts between multiple Ladder Editors.*

Project Explorer Window

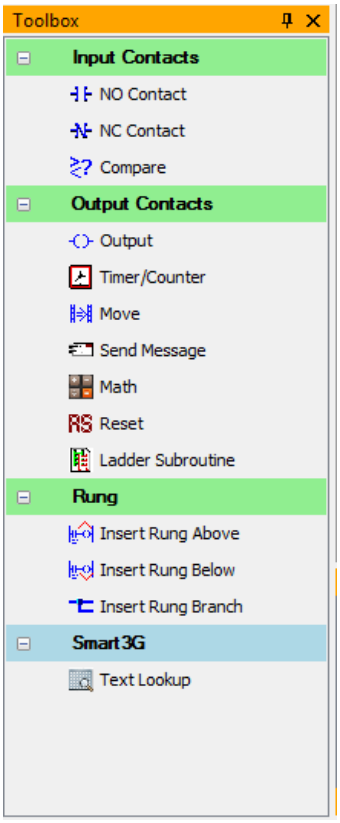


The Project Explorer Window is the basic navigation and management window for the application. It can be accessed from the View Menu or by clicking the Project Explorer Icon under the Menu Bar.

The Project Explorer diplays:

- The current active project.
- The Ladder Program on the active project.
- The Ladder Subroutines on the active project.
- The Tag Database for creating any form of text definitions.
- The Text Database where Smart3G serial barcodes can be defined.

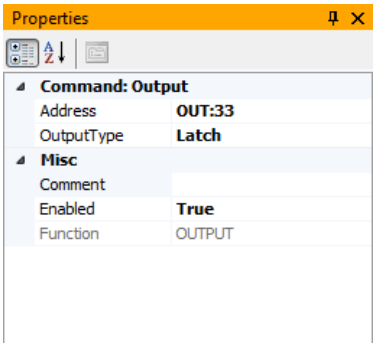
Toolbox Window



The Tool Box contains the needed tools and actions for creating a ladder program. Adding contacts, rungs, or branches, etc. can be done from here.

The Tool Box can be accessed from the View Menu, or by clicking the Tool Box Icon under the Menu Bar.

For a list of available commands, see Ladder Commands.



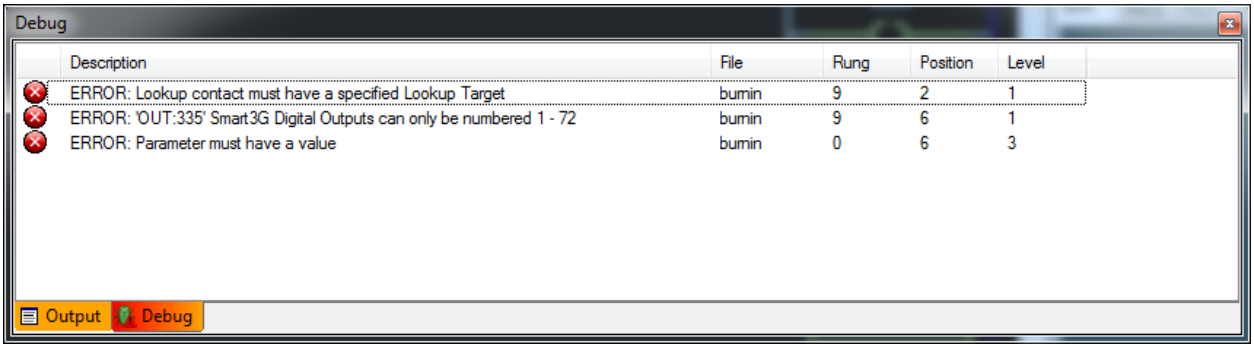
Properties Window

The Properties Window displays the properties for each command in the Ladder Program Editor.

It can be accessed by Right-Clicking inside a Ladder View and Selecting "Properties", from the View Menu, or by clicking the Property Window Icon under the Menu Bar.

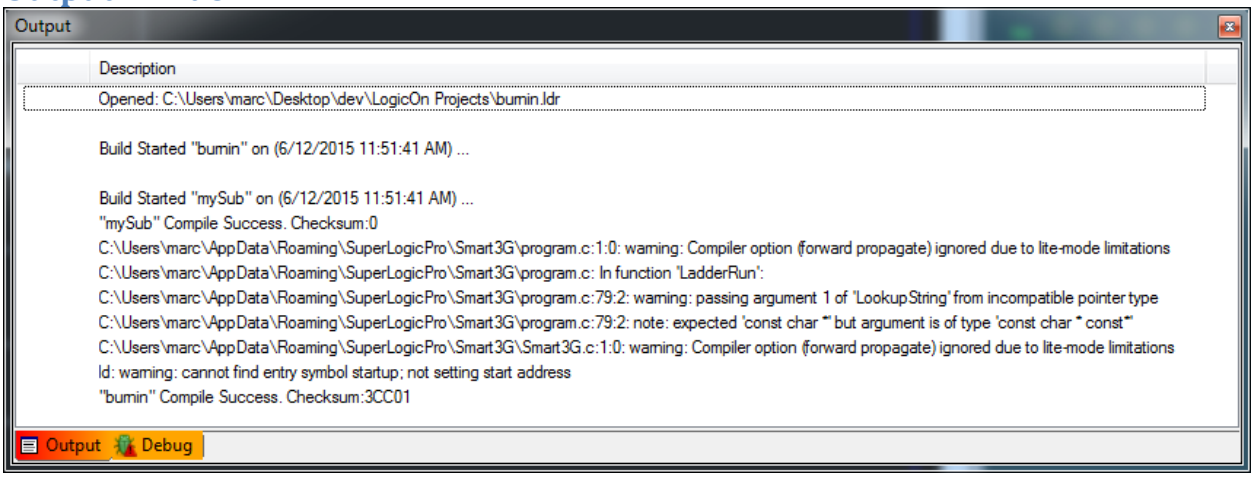
NOTE: *Modification is disabled when the ladder program is being monitored.*

Debug Window



The Debug Window displays compilation errors for the ladder program (if there are any) after the compilation process. Error entries can be clicked to jump to the source of the error. The Debug Window can be accessed by clicking the Debug Window Icon under the menu bar, otherwise it pops-up when errors are detected after compiling a Ladder Program.

Output Window

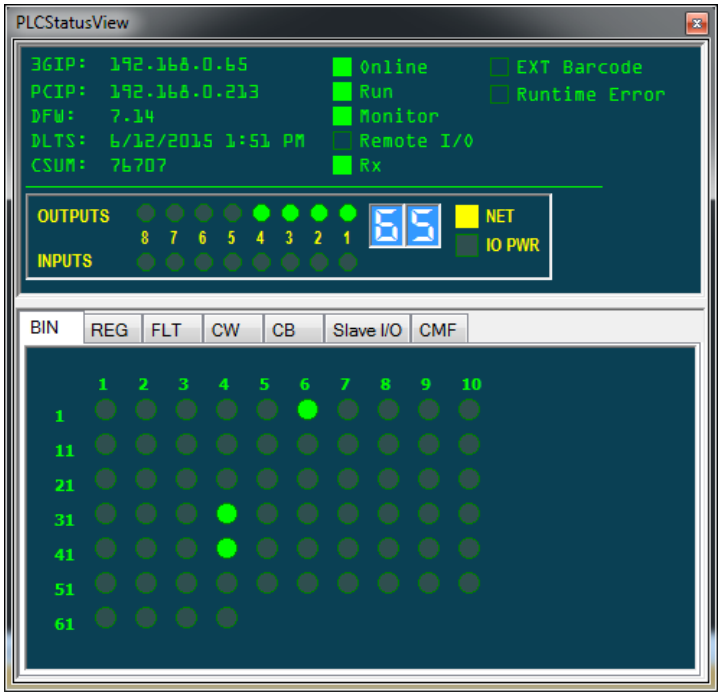


The Output Window provides detailed and technical feedback such as application log messages, compile messages, and errors. The Output Window can be accessed by clicking the Output Window Icon under the menu bar, otherwise it pops-up automatically when necessary.

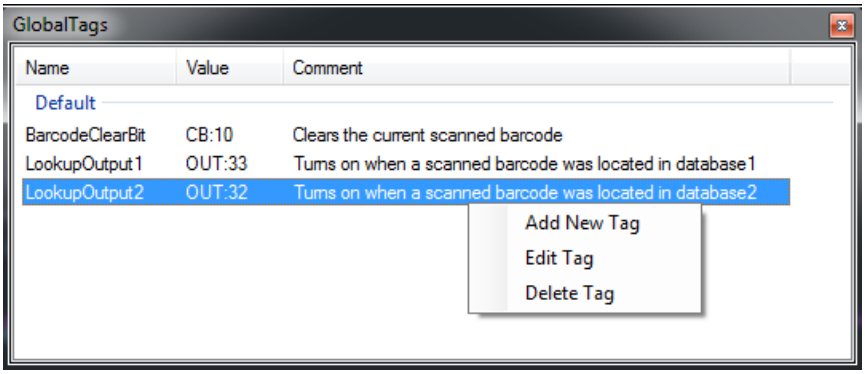
Device Status Window

The Ladder Status window displays general information about the target device, as well as its files. The current values may be viewed in this window if there is a connection is present. The user may also change the file values from this window, which provides a degree of manual control on the execution of the ladder program.

The Status Window can be accessed by clicking on the Status Window icon under the menu bar.



Tag Database Window



SuperLogic Pro supports user-defined tags that can be substituted in place of the standard file addressing of the Ladder-Logic syntax. These definitions are created and modified in the Tag Database Window. The Tag Database Window can be accessed by double-clicking the "Tag Database" node in the Project Explorer.

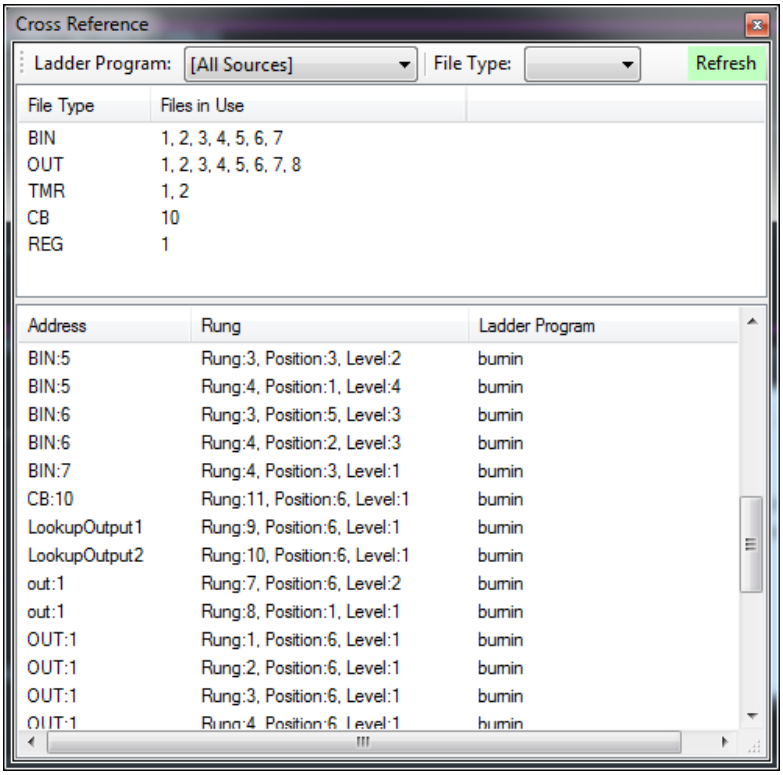
Creating a tag has the following parameters:

- Name – The definition mask over the real value of the tag.
- Value – The real file address in correct Ladder-Logic syntax.
- Comment

Advantages of using tags:

- More descriptive file addressing.
- If you are using a tag multiple times in the ladder program, you can simply change the value in the Tag Database Editor once and it will affect all commands using the tag.
- Message Formats can be defined using tags.

Cross Reference



The Cross Reference View provides a list of which Device Files and Defined Tags are used in the current project. It also lists which rung and Ladder Program is referencing the file, device, or tag.

III. System Files

Smart3G PLC devices support multiple addressable file types of which are:

- Signed 32-bit Integers
- Signed 32-bit Floats (floating point value)
- Bits (1 or 0)
- Timers
- Counters
- Barcode text strings
- Etc.

Below is a more detailed explanation for each of these files.

NOTE: *The following information was written based on version 7.14 of the Smart3G firmware.*

Legend:

Description	General Information and notes on the file type.
Keyword	The keyword syntax used to reference the file.
Index	The available range of indexing values.
Attributes	The attributes that are present for the given file. The file type of the attribute is enclosed in parentheses. Bit Indexing returns a bit using a specified bit position (Known as BIN in SuperLogic).
Format	The Format address syntax used to reference a specific file in the Ladder Program. Italicized text enclosed in <> are variable values.
Usage	How the file can be used (Read, Write, or both).

Integer Register

Description	A 32-bit signed integer. Referred to plainly as "Register" in SuperLogic Pro. **NOTE**: *Although numerical values with decimal point precision can be assigned to these files, the digits following the decimal point are dropped. For numerical values with a decimal point, consider using the float registers (FLT).*
Keyword	REG

Index	• **1 to 50**
Attributes	Bit Indexing [0 – 31]
Format	REG:*<file index>.<bit index>*
Usage	Read, Write

Float Registers

Description	A 32-bit signed floating-point value that can use fractional values following a decimal point.
Keyword	FLT
Index	• **1 to 25**
Attributes	
Format	FLT:*<file index>*
Usage	Read, Write

Binary Registers

Description	A bit value that can be 0 or 1.
Keyword	BIN
Index	• **1 to 64**
Attributes	
Format	BIN:*<file index>*
Usage	Read, Write

Timer Registers

Description	Timers that can keep time in a resolution of milliseconds. The timer keeps time until the user-defined preset value is reached. **NOTE**: *Timer Registers are directly usable only with the Reset and Timer/Counter command. Otherwise, only the timer attributes are used.*
Keyword	TMR
Index	• **1 to 50**
Attributes	• **AC** – Accumulated Time in milliseconds (REG value). • **PR** – Preset Value in milliseconds (REG value). • **EN** – Timer Enabled (BIN value). • **DN** – Done Timing (BIN value). • **TM** – Currently Timing (BIN value).
Format	TMR:*<file index>.<attribute>*
Usage	Read

Counter Registers

Description	Counters that increment their accumulated value by 1 every time the rung state transitions to true. Directly usable only with the Reset and Counter Command. **NOTE**: Counter Registers are *directly usable only with the Reset and Timer/Counter command. Otherwise, only the timer attributes are used.*
Keyword	CNT
Index	• **1 to 10**
Attributes	• **AC** – Accumulated Counts (REG value). • **PR** – Preset Value (REG value). • **EN** – Timer Enabled (BIN value). • **DN** – Done Timing (BIN value).
Format	CNT:*<file index>.<attribute>*
Usage	Read

Inputs

Description	Input states of the Smart3G Device (similar to BIN values in usage). **NOTE**: *Although a Smart3G device has only 8 Digital Inputs, it can utilize a maximum of 72 Inputs through a master/slave configuration.*
Keyword	IN
Index	1 to 72
Attributes	
Format	IN:*<fileIndex>*
Usage	Read

Outputs

Description	Output states of the Smart3G Device (similar to BIN values in usage). **NOTE**: *Although a Smart3G device has only 8 Digital Outputs, it can utilize a maximum of 72 Outputs through a master/slave configuration.*
Keyword	OUT
Index	1 to 72
Attributes	
Format	OUT:*<file index>*
Usage	Read, Write

Control Words

Description	Special purpose 16-bit integers for device control.
Keyword	CW
Index	1 to 85

Attributes	Bit Indexing [0 – 15]
Format	CW:*<file index>.<attribute>*
Usage	

Index	Use	Description
1 – 2	RW	Data is transmitted to master in Ethernet/IP poll. Can be used to transmit status info to master.
3 – 4		Reserved
5	RW	Serial Port Baud Rate. **Baud rate = (*CW5value* x 100)**. *Ex: Write 96, 144, 384 etc. for 9600, 14400, 38400 Baud*
6	R	Merge Device ID (Read Only)
7	R	Divert 1 Device ID (Read Only)
8	R	Divert 2 Device ID (Read Only)
9 – 10	RW	Ethernet/IP or Modbus/TCP – Data Write from Master
11	RW	Local Device ID
12	RW	Downstream Device ID
13	RW	Upstream Device ID
14	R	Master/Slave Protocol Scan List Status. **Bits 0 – 7** indicate online status for each device. **Bits 8 – 15** indicate Output Power Status.
15	R	User Defined. This value is read from the SuperLogic Zone Count configuration parameter. (Read Only).
16	RW	Ladder Logic Transmit Message Destination Device ID
17 – 22	RW	Ladder Logic Transmit Message Data. Each control word can have a value of 0 thru 255.
23 – 24	RW	Special Definition for Modbus/TCP Holding Register Read CW:23 = Read 8 Inputs (Bits 0 – 7) and new serial port data bit (Bit 15) Read CW:24 = Read 8 Outputs (Bit 0 – 7) and Output Enable Bit (Bit 15) Write CW:23 = Write to CW:23 Write CW:24 = Write to CW:24, or Write directly to output pins if Remote I/O Checkbox is checked.
25 – 30	RW	Ladder Logic Receive Message Data. Each control word can have value of 0 to 255.
31 – 40		Reserved
41 – 48	RW	Device IDs for remote (slave) Smart-3G controllers. Used to enable slave device scanning for the S3G Master/Slave protocol. Value of 0 disables scanning.
49 – 56	R	Input/Output data for slave devices for S3G protocol. **Bits 0 – 7** are inputs and **bits 8 – 15** are outputs. This data is also accessible using the IN/OUT file, *(e.g. IN:55, OUT:71 etc.)*

	57	R	Scanned barcode length
	58 – 60		Reserved
	61 – 81	R	Received barcode text

Control Bits

Description	Special purpose binary values for device control.
Keyword	CB
Index	1 to 25
Attributes	
Format	CB:*<file index>*

Usage

Index	Use	Description
1	R	Package Arriving from Merge Branch. (Input, Read Only)
2	RW	Merge Ready Status to Branch (Output, Read/Write)
3	RW	Package Available for Divert Branch 1. (Output, Read/Write)
4	R	Divert Branch 1 is Ready (Input, Read Only)
5	RW	Package Available for Divert Branch 2. (Output, Read/Write)
6	R	Divert Branch 2 is Ready (Input, Read Only)
7	R	FIFO is Empty (Read Only)
8	R	FIFO is FULL (Read Only)
9	R	Output Power On (Read Only)
10	RW	New Barcode Received (Read/Write)
11	RW	Package Available Status to Downstream main line (Output, Read/Write)
12	RW	Ready to Take Packages from Upstream main line (Output, Read/Write)
13		Reserved
14	R	General Purpose Configuration Bit. The value is set in the SuperLogic Configuration screen by the user and saved in Smart3G Flash memory. (Read Only for Ladder program). This bit can be used for any configuration selection the user may decide.
15		Barcode Overrun. This bit is activated when a new barcode is received when CB:10 is on. This bit must be cleared manually by the ladder program.
16		Reserved

	17		Reserved
	18	R	Package Available from Upstream main line. (Input, Read Only)
	19	R	Downstream main line is ready to take packages. (Input, Read Only)

NOTE: *When using serial scanners (e.g. barcode scanning), CB:10 must be unlatched in order to receive new serial data.* ***Failure to unlatch CB:10 will result in new serial data being ignored by the 3G Device.***

Refer to this section for more information on correct Barcode Lookups.

Barcode

Description	Received barcode text located on CW:61 – 81 **NOTE**: This feature requires the "Extended Lookup" activation.
Keyword	BAR
Index	
Attributes	• **Subtext. *BAR:<start index>*** ➢ Returns all the barcode characters at the specified *<start index>.* • **Subtext. *BAR<start index>,<length>*** ➢ Returns a number of barcode characters (specified by *<length>*) at the specified *<start index>*. *Example*: Current barcode is "1234567890". ***BAR:0*** will return "1234567890" ***BAR:5*** will return "*67890*" ***BAR:0,5*** will return "*12345*" ***BAR:1,5*** will return *"23456"* **NOTE**: The ***<start index>*** *can be any value from 0 – 39, but the* ***<length>*** *must not be greater than (40 - <start index>).* *Currently, this can only be used with the compare command.*
Format	*BAR:<start index>* **or** *BAR<start index>,<length>*
Usage	Read

IV. Ladder Commands

A Ladder Program is comprised of multiple rungs that are executed from top to bottom. Each rung is comprised of two types of commands, namely the inputs and outputs.

Input commands are used to evaluate if the state of the rung (active or not) which affects the execution of the output commands.

Output commands are commands that carry out actions in the ladder program such as writing to file values. Some output commands are not executed if the rung is inactive, while others will simply behave in a different manner.

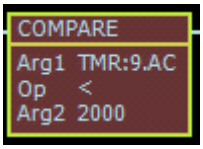
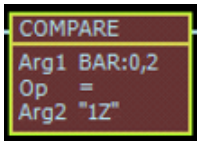
Normally Open Command

Description	An input command whose condition is true when the addressed bit value is active.
Type	Input
Parameters	**Address** – the referenced bit to read from. (*Ex. IN:1, OUT:2, TMR:2.TM*).
Usage	Read
File Types	All Binary Types. (BIN, IN, OUT, CB, Register Bits, TMR/CNT Attributes, etc.)

Normally Closed Command

Description	An input command whose condition is true when the addressed bit value is inactive.
Type	Input
Parameters	**Address** – the referenced bit to read from (*Ex. IN:1, OUT:2, TMR:2.TM*).
File Types	All Binary Types. (BIN, IN, OUT, CB, Register Bits, TMR/CNT Attributes, etc.)

Compare Command

Description	A command whose condition depends on the logical comparison of the values of two referenced files.
Type	Input
Parameters	• **Arg1** – The first argument (referenced file or constant) for comparison. (*Ex. 10, REG:5, "text", BAR:8,10*). • **Arg2** – The second argument (referenced file or constant) for comparison. (*Ex. 10, REG:5, "text", BAR:8,10*). • **Operation** – The logical comparison to make between the 2 arguments. ➢ ***Equal (=)*** – true if the 2 arguments are equal. ➢ ***Not Equal(!=)*** – true if the 2 arguments are not equal. ➢ ***Greater Than (>)*** – true if Arg1 is greater than Arg2. ➢ ***Less Than (<)*** – true if Arg1 is less than Arg2. ➢ ***Greater Than or Equal to (>=)*** – true if Arg1 is greater than or equal to Arg2. ➢ ***Less Than or Equal to (<=)*** – true if Arg1 is less than or equal to Arg2.
File Types	Numerical values (REG, FLT, CW). Text values (BAR, BAR:5, BAR:8,10). Timer .PR .AC and Counter .PR, .AC Accumulators.
Examples	The Compare Command can also be used for received Barcode Comparisons, and on Timer/Counter Accumulators like in the example below.  

Text (Barcode) Lookup Command

Description	Checks if the current scanned barcode is defined in the selected "Text/Barcode Database". **NOTE**: Whenever a new barcode is scanned **CB:10** is activated. *Additional barcodes are not accepted until **CB:10** is cleared (usually with an unlatch output contact).*
Type	Input

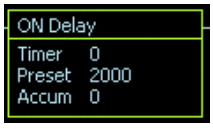
Parameters	• **Lookup Target** – The "Text/Barcode Database" where the lookup will be performed. For more information on creating the barcode database, see Barcode Lookups. Once the database has been created, it will show in the drop down list in the contact properties.
File Types	Text/Barcode Database

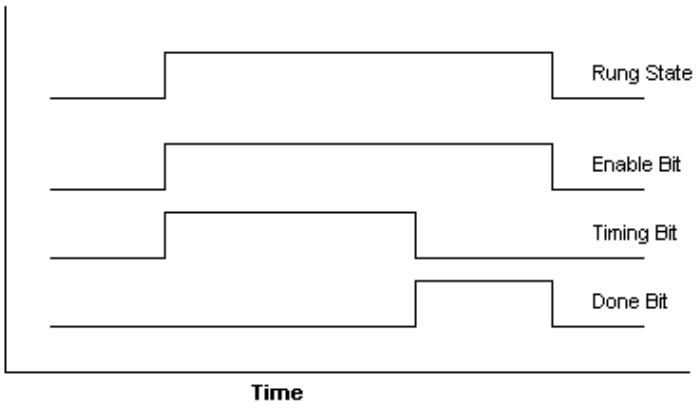
Output Command

Description	This command sets the addressed bit to true or false. When the rung condition is true, the addressed bit or output is set to true (1 or high) and with rung condition false, the bit or output is set to false (0 or low).
Type	Output
Parameters	• **Address** – the referenced bit value to write to (Ex. Out:1, REG:5.3, etc). • **Output Type** – Controls the behavior of the command. ➢ Normal ( ) – If the rung is active, it writes 1, otherwise it writes 0.  ➢ Latch (**L**) – If the rung is active, it writes 1, otherwise it does nothing.  ➢ Unlatch (**U**) – If the rung is active, it writes 0, otherwise it does nothing.  ➢ Latch Transition to True(**^**) – Writes 1 only when the current rung state becomes active and the previous rung state was inactive.  ➢ Latch Transition to False(**v**) – Writes 1 only when the current rung state becomes inactive and the previous rung state was inactive.

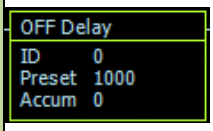
	
File Types	Binary files (BIN, OUT, CB, Register Bits)

Timer/Counter Command

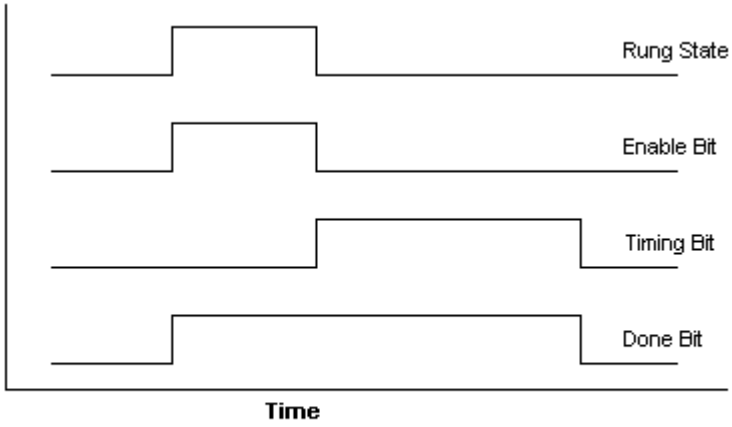
Description	The Timer/Counter command makes use of either a Timer or Counter register for its functionality • **Timer** – The timer register keeps timing until the preset value (in milliseconds) is reached. • **Counter** – The counter register keeps counting until the reset value is reached.
Type	Output
Parameters	• **ID** – The timer or counter register that is bound to this command. • **Preset** – A 32-bit integer value that specifies when the command stops timing/counting. *For timers, this value is in milliseconds.* • **Type** – Controls the behavior of the command. **NOTE**: *Timers increment the "Accumulator" value every millisecond, while Counters increment the "Accumulator" value every time the rung state transitions from false to true.* ***Timer Types:***  **ON DELAY** – This timer starts timing when the rung condition becomes true. As long as the rung condition is true, the accumulator keeps on timing until it reaches the preset value. When the Accumulator is equal to Preset, the 'Done' bit is set and the Timing bit is reset. The Timer Enable bit follows the rung condition. The Done bit, Timing Bit and Enable Bit are reset as soon as the rung condition becomes false. The Accumulator is reset to 0 until the rung condition becomes true again.



ON Delay Timing Diagram

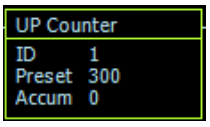


OFF DELAY – This timer starts timing when the rung condition transitions to false. When the rung state transitions from true to false, the Accumulator keeps incrementing until the preset value is reached or the rung condition becomes true again. The enable bit follows the rung state, while the Done bit is a combination (logical 'OR') of the Enable and Timing bits. The Timing and Done bits are reset when the Accumulator reaches the Preset value.



OFF Delay Timing Diagram

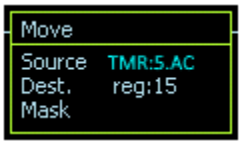
Counter Types:



UP COUNT – The "UP Counter" counts every rung state transition from false to true until the Preset value is reached. Each transition of rung condition from false to true is registered by incrementing the Accumulator by 1. When the Accumulator value becomes equal to the Preset value, the Done bit is set to true. The Enable Bit follows the rung condition. The "UP Counter" can only be reset with "Reset contact". At reset, the Accumulator value and the Done Bit are set to zero. Please refer to Timer/Counter Reset element.

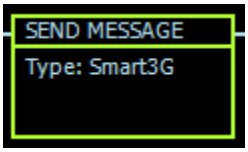
File Types	TMR, CNT

Move Command



Description	Copies a specified source file's value (or a constant numerical value) into a specified destination file while the rung is active.
Type	Output
Parameters	• **Source** – The referenced file or constant value to be moved (Ex. 5000, 0.056, REG:9, CW:10, etc). • **Destination** – The file where the source file's value is written (Ex. FLT:2, REG:9, CW:10, etc).
File Types	Numerical Values. (REG, FLT, CW)

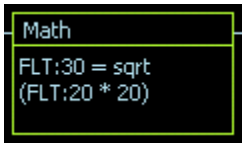
Send Message Command



Description	Sends a message over Ethernet to other card or PC Host. **NOTE**: *This command only triggers when the rung state transitions to true.*
Type	Output
Parameters	➢ ***Smart3G*** – This protocol sends an explicit message to another ***Smart3G Device,*** or to the host. When the rung condition becomes true, the 6 bytes of data in the send-buffer will be sent to the receive buffer of the destination device. The send and receive buffers can only hold values from 0 to 255 in each location.

Send Buffer	CW:17 - CW:22
Receive Buffer	CW:25 - CW:30
Destination	CW:16

Math Command



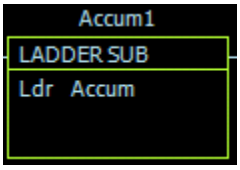
Description	Performs binary and unary mathematical operations (depending on how many arguments were specified) and writes the result to a specified destination file while the rung is active. **NOTE**: *This contact works most effectively when working with float (FLT) values. If the destination file is of type REG, all numbers following the decimal point are dropped.*
Type	Output
Parameters	• **Arg1** – The first argument. (Ex. FLT:4, 5000, 4.556, REG:9, CW:10, etc) • **Arg2** – The second argument. (Ex. FLT:4, 5000, 4.556, REG:9, CW:10, etc) • **Destination** – Address of File to store the result. (Ex. FLT:4, REG:9, CW:10, etc) • **Binary Operation** – operation to perform between the two arguments. All Bit manipulation values must be done with REG (integer) values. **NOTE**: *When using Float type for bitwise operations, the digits after decimal point are dropped, eg. FLT:4 BitAND 123.* ➢ *None* – the result is the value of Arg1. Arg2 is ignored. ➢ *Addition* ➢ *Subtraction* ➢ *Multiplication* ➢ *Division* ➢ *Power/Exponentiation* ➢ *BitAND* – Bitwise AND ➢ *BitOR* – Bitwise OR ➢ *BitXOR* – Bitwise Exclusive OR ➢ *BitShiftLeft* – Shifts Arg1's bit value by a specified number of digits (Arg2's value) to the left. ➢ *BitShiftRight* – Shifts Arg1's bit value by a specified number of digits (Arg2's value) to the right. • **Unary Operation** – operation to perform on the result of the Binary Operation. ➢ *None* ➢ *Negative* – Negates the value. ➢ *Bitwise Inversion* – Invert the bit value. ➢ *Absolute* – Absolute value. ➢ *Square Root*

	➢ *Sine* ➢ *Cosine* ➢ *Tangent* ➢ *Cosecant* ➢ *Secant* ➢ *Cotangent* ➢ *Natural Logarithm* ➢ *Common Logarithm*
File Types	Numerical Types. (REG, FLT, CW)

Reset Command

Description	Resets a timer or counter when the rung becomes active and the previous state was inactive.
Type	Output
Parameters	• **Address** – The Timer or Counter to reset. (Ex. TMR:20, CNT:10 etc)
File Types	TMR, CNT

Ladder Subroutine Command



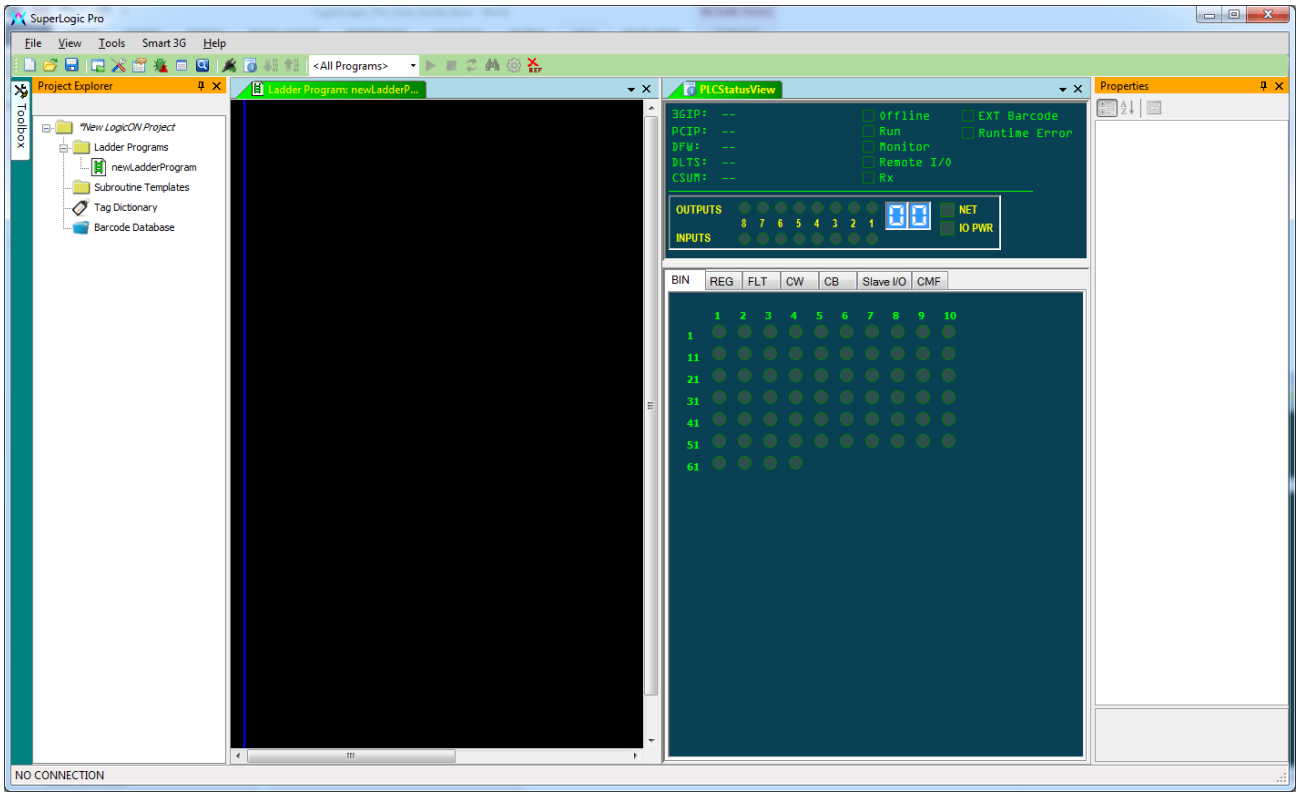
Description	Creates a subroutine call using a Ladder Subroutine from the project. For more information on Subroutines, refer to What are Ladder Subroutines and how do I use them?
Type	Output
Parameters	• **Call Name** – The unique name for this instance • **Ladder Subroutine** – The Ladder Subroutine to be used.
File Types	Ladder Subroutine

V. FAQs

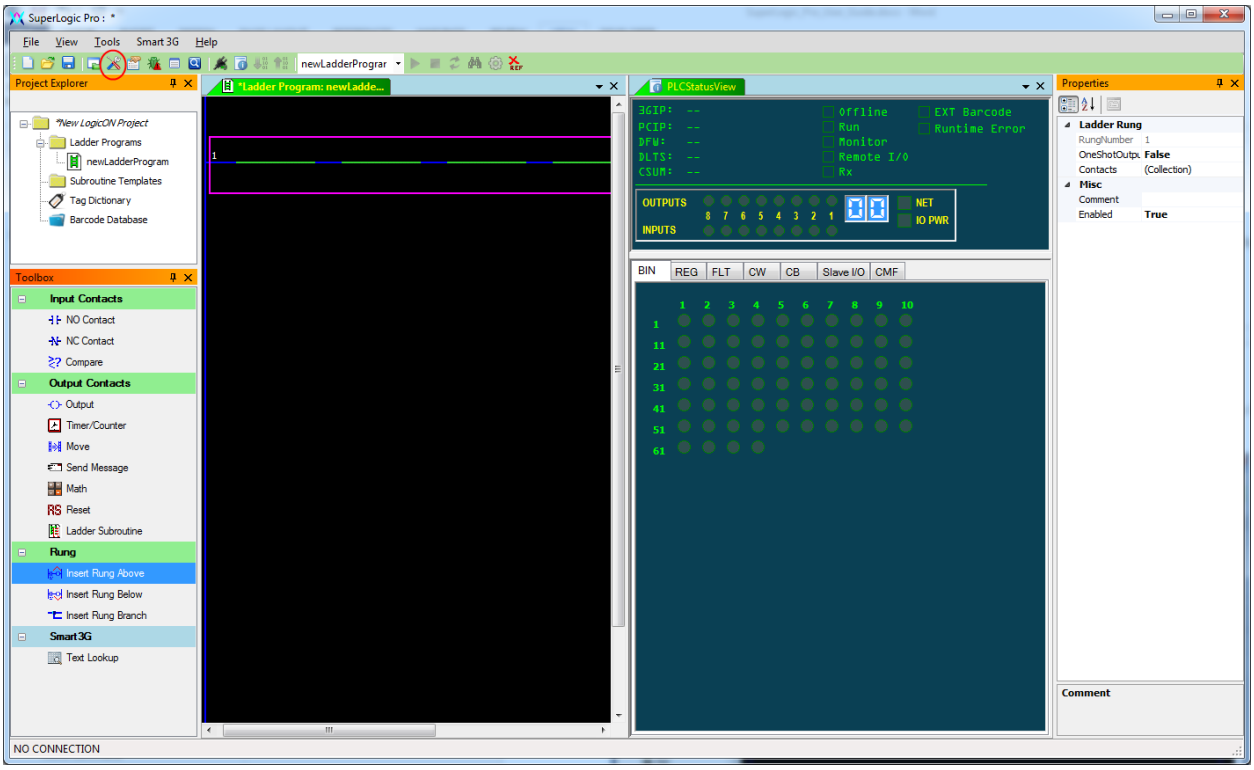
Where do I Start?

Creating your First Ladder Program

1. Click ***File***, then ***New Project***. You should then see the screen below.



2. You may now start editing your new Ladder Program. You can create the first rung, as well as add contacts from the Toolbox Window (Accessible from the View Toolbox Menu circled in red). Click on the Thumb-pin of the Toolbox window to make it visible all the time.

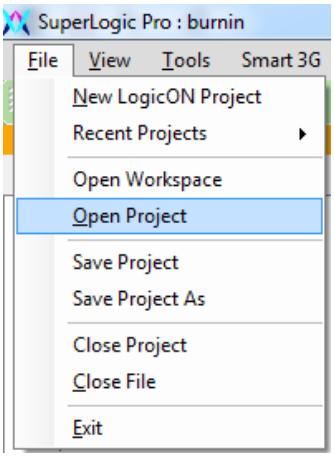


3. You may give the project a name when you save it. From the Menu Bar, click ***File*** -> ***SaveAs.***

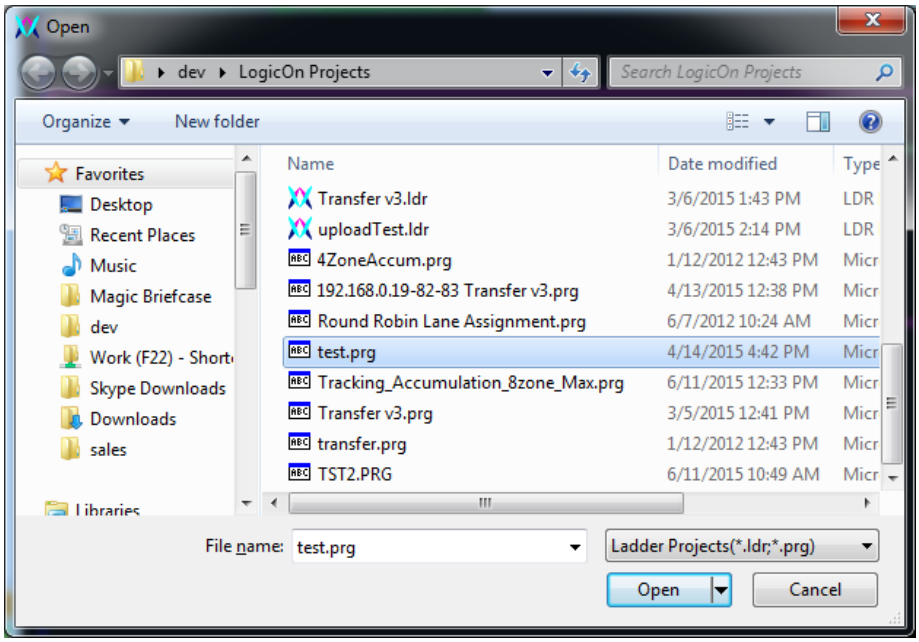
Importing a (.prg) Ladder Program

Instead of creating a ladder program from scratch, SuperLogic Pro can also import older Ladder Logic programs (.prg files). This can be done the same way as opening any ladder project.

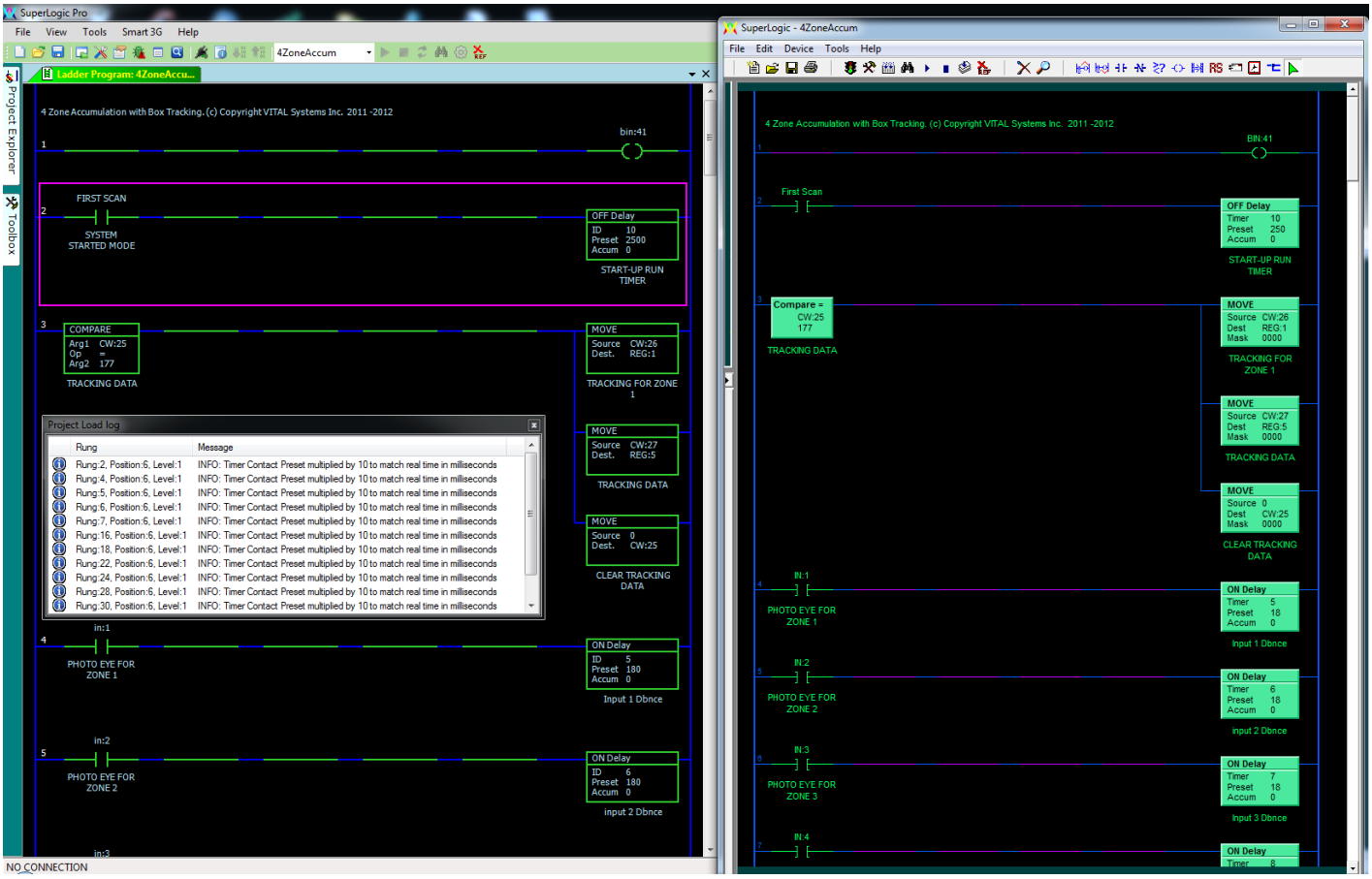
1. On the main menu, click "File → Open Project".



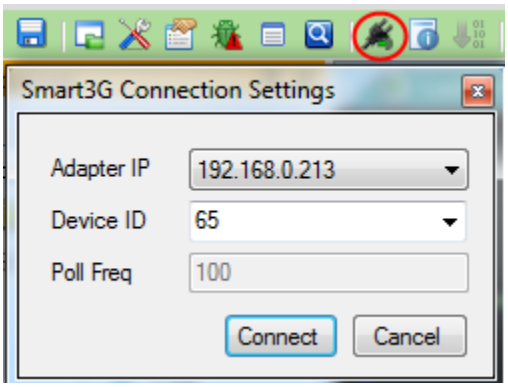
2. Select a (.prg) ladder program to open.



3. Imported (.prg) programs will most probably require a few conversions, some of which are performed automatically.


SuperLogic Pro
File View Tools Smart 3G Help
4ZoneAccum
Ladder Program: 4ZoneAccu...
4 Zone Accumulation with Box Tracking. (c) Copyright VITAL Systems Inc. 2011 -2012
bin:41
FIRST SCAN
SYSTEM STARTED MODE
OFF Delay
ID 10
Preset 2500
Accum 0
START-UP RUN TIMER
COMPARE
Arg1 CW:25
Op =
Arg2 177
TRACKING DATA
MOVE
Source CW:26
Dest. REG:1
TRACKING FOR ZONE 1
MOVE
Source CW:27
Dest. REG:5
TRACKING DATA
MOVE
Source 0
Dest. CW:25
CLEAR TRACKING DATA
Project Load log
Rung
Message
Rung:2, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:4, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:5, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:6, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:7, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:16, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:18, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:22, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:24, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:28, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
Rung:30, Position:6, Level:1 INFO: Timer Contact Preset multiplied by 10 to match real time in milliseconds
in:1
PHOTO EYE FOR ZONE 1
ON Delay
ID 5
Preset 180
Accum 0
Input 1 Dbnce
in:2
PHOTO EYE FOR ZONE 2
ON Delay
ID 6
Preset 180
Accum 0
input 2 Dbnce
in:3
NO CONNECTION
SuperLogic - 4ZoneAccum
File Edit Device Tools Help
4 Zone Accumulation with Box Tracking. (c) Copyright VITAL Systems Inc. 2011 -2012
BIN:41
First Scan
OFF Delay
Timer 10
Preset 250
Accum 0
START-UP RUN TIMER
Compare =
CW:25
177
TRACKING DATA
MOVE
Source CW:26
Dest REG:1
Mask 0000
TRACKING FOR ZONE 1
MOVE
Source CW:27
Dest REG:5
Mask 0000
TRACKING DATA
MOVE
Source 0
Dest CW:25
Mask 0000
CLEAR TRACKING DATA
IN:1
PHOTO EYE FOR ZONE 1
ON Delay
Timer 5
Preset 18
Accum 0
Input 1 Dbnce
IN:2
PHOTO EYE FOR ZONE 2
ON Delay
Timer 6
Preset 18
Accum 0
input 2 Dbnce
IN:3
PHOTO EYE FOR ZONE 3
ON Delay
Timer 7
Preset 18
Accum 0
input 3 Dbnce
IN:4
ON Delay

Runtime Operation



Although SuperLogic Pro can compile Ladder-Logic programs, running them requires a connection to a Smart3G Device.

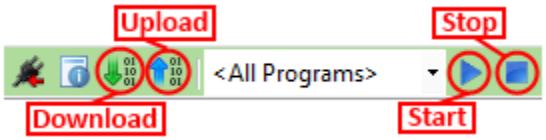
Click the Connect Button (circled in red) to setup a connection. A window with Connection Parameters will appear. Set the necessary values for the specified fields then click on "Connect".

The ***Adapter IP*** entry defines on which network the Smart3G device is located. This setting can be changed for PCs with more than one network adapter.

The ***Device ID*** entry determines which Smart3G device SuperLogic Pro will connect to.

Poll Frequency determines the interval (in milliseconds) to request monitoring data from the Smart3G device.

> **NOTE**: *Before attempting a connection, make sure that the Smart3G device is connected to the network and is not in an error state (such as conflicting IP Address).*



If the connection attempt is successful, you are given the option to then "***Download"*** your current project on the Smart3G Device, or "***Upload"*** the running project from the Smart3G device.

- **Download** – Installs the current project on the Smart3G device. After downloading the project, the ladder program can then be started by clicking on the "Run Ladder Program" button. Additionally, the Smart3G device will now always run the ladder program on power-up.
- **Upload** – Retrieve the project from the Smart3G device.

Barcode Lookups

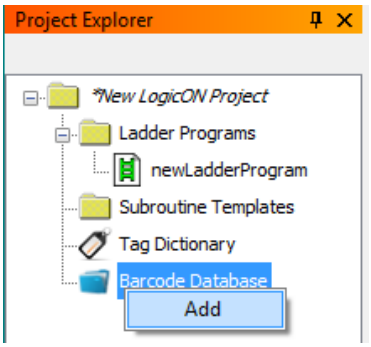
NOTE: *Make sure the Barcode scanner is set to the correct **Baudrate (default in 3G card is 38400)**, **DataBits=8**, **Parity=None**, **StopBits=1**, and **CR** or **LF** is sent at the end of the barcode string.*

***Host ID** in the 3G config should be set to **None** in order to use the on-board lookup feature.*

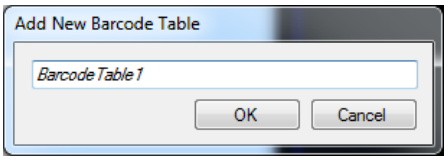
***CB:10** must also be cleared in order to receive new barcodes. **If CB:10 is activated**, receiving a new barcode will activate **CB:15 (Buffer Overrun)**, and the new barcode will be ignored.*

*The **Baudrate** on the 3G card can be modified by writing the Baud value divided by 100 to CW:5, e.g., for 9600 baud, write 96 to CW:5. Other port settings are fixed at 8 bits, No Parity, 1 Stop Bit.*

1. Barcode Lookups first require a barcode table where the lookup can be performed. To do this, go to the Project Explorer and right-click the "***Barcode Database***" and select ***Add***.

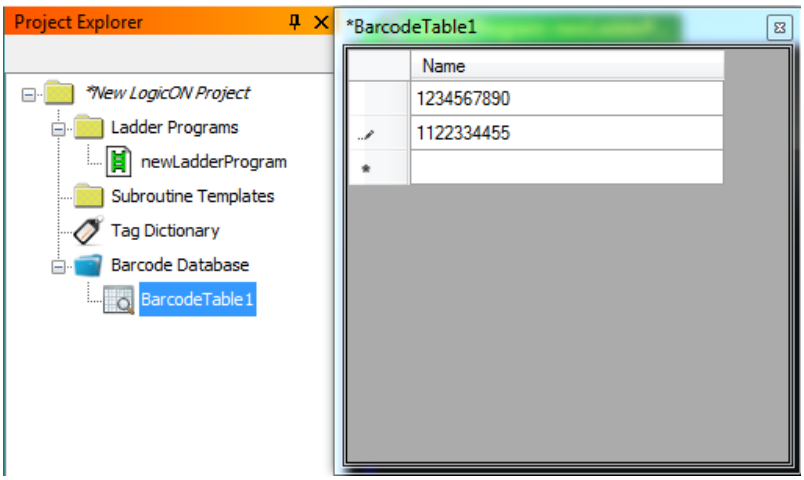


2. Enter a name for the new barcode table then click OK to create the new barcode table.

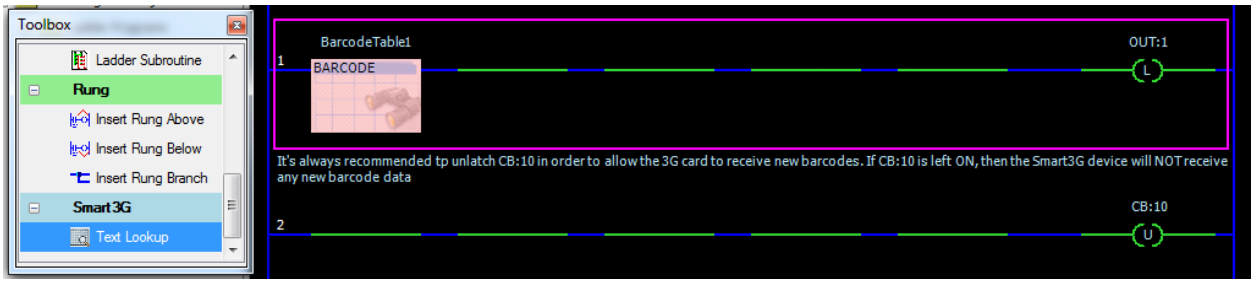


3. A new entry should appear under the ***Barcode Database*** for the new barcode table. Double-click it to open the new barcode table. Barcodes can be added and removed from here. Once the barcode table has all the necessary entries, it can now be used within the ladder program.

NOTE: *Even though SuperLogic allows creating multiple barcode tables, only one Barcode Table can be searched by default. The "**Extended Barcode Table Lookup**" feature is required for using multiple barcode tables in the Smart3G device. Please Contact www.vsys.co to purchase the Extended Barcode Feature Activation.*



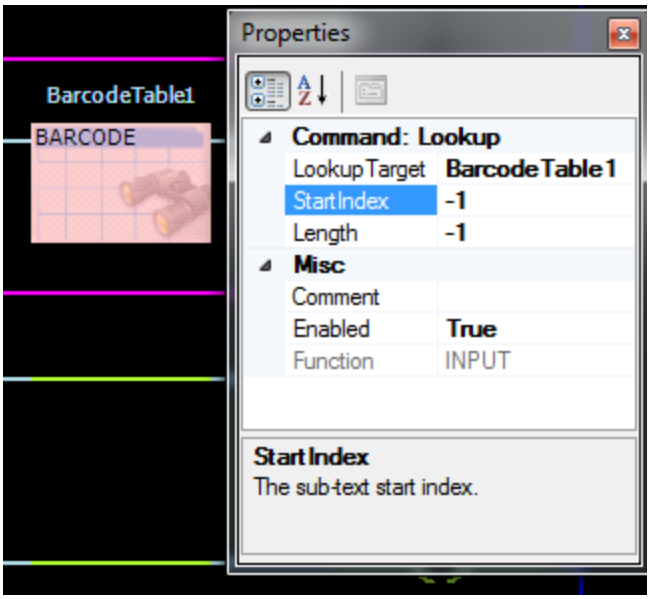
4. To use the new barcode table, it will need to be referenced by a "***Lookup Contact***". Add a Lookup Contact in the ladder program (from the toolbox).



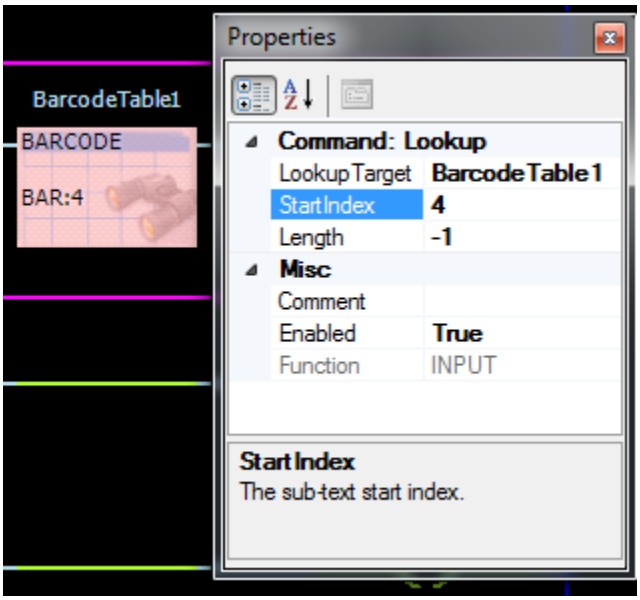
> **NOTE**: *It's always recommended to unlatch **CB:10** in order to allow the 3G card to receive new barcodes. If **CB:10** is left ON, then the Smart3G device will NOT receive any new barcode data.*

5. Click on the new Lookup Contact and check its properties from the Properties Window. The Lookup Contact requires a ***LookupTarget***, which is any barcode table from the current project, where the lookup will be performed. A Start Index and Length may also be specified if only a part of the barcode string is required to perform the lookup.

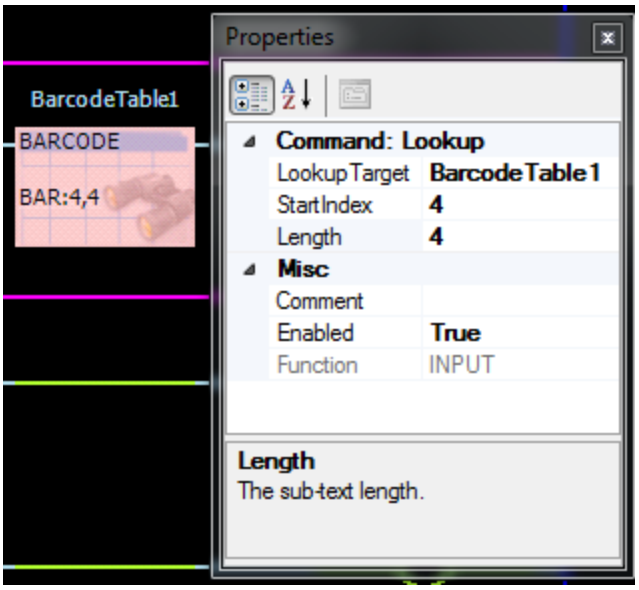
   This example will check if the current barcode is defined as an entry in "**BarcodeTable1**".



This example will check if all the remaining characters, after the 4th character of the barcode string is a text entry defined in "**BarcodeTable1**".

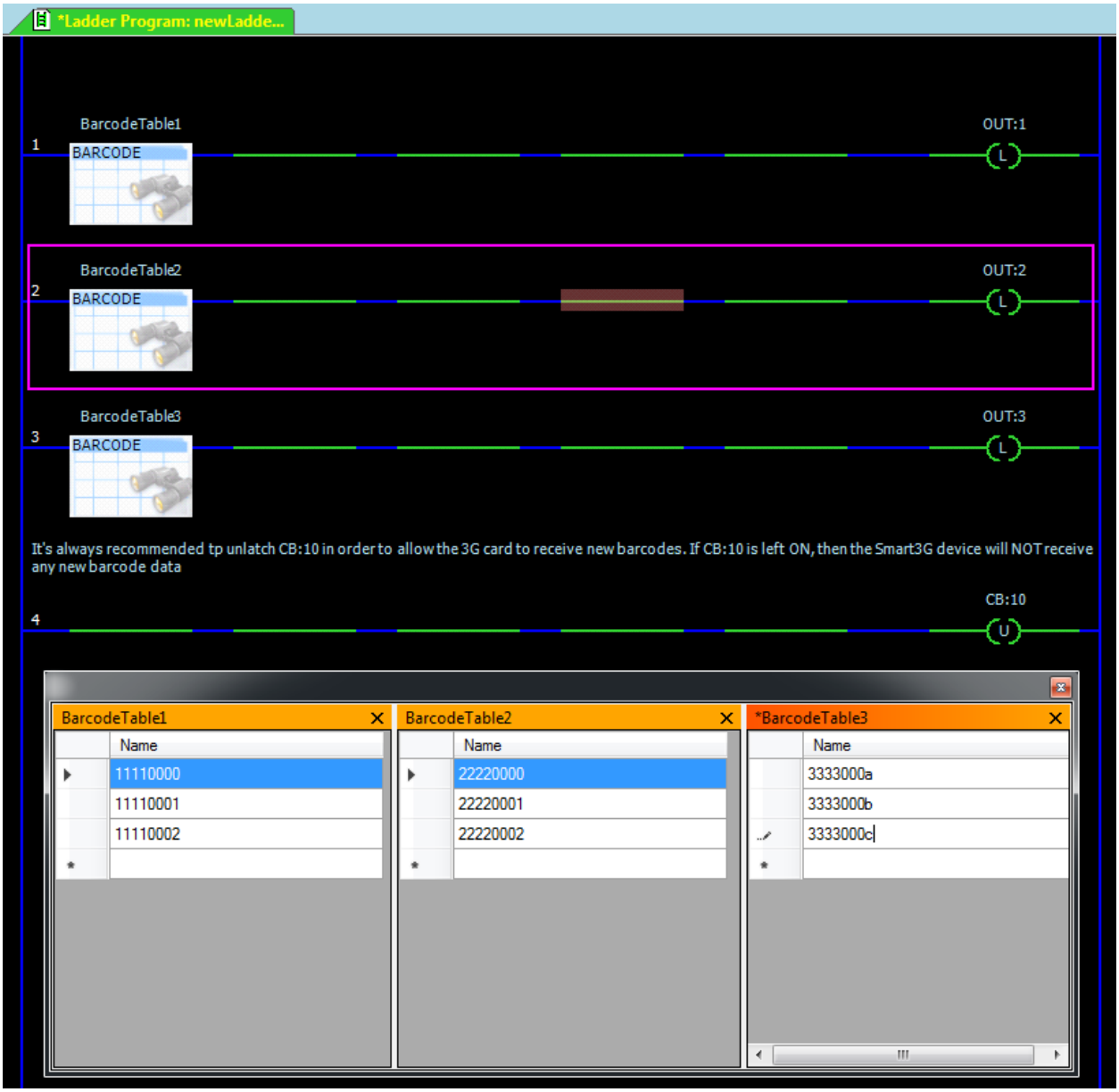


This example will check if the next 4 characters, after the 4th character of the barcode string, is a text entry defined in "**BarcodeTable1**".



6. Whenever a scanned barcode is present in the Smart3G device, the lookup contact will check the barcode table and see if the scanned barcode is contained within. If the scanned barcode is defined inside the barcode table, then the contact's state becomes true, and false if not. Multiple barcode tables can be utilized in order to perform certain functions depending on the scanned barcode.

NOTE: *When using serial scanners (e.g. barcode scanning), CB:10 must be unlatched in order to receive new serial data.* ***Failure to unlatch CB:10 will result in new serial data being ignored by the 3G Device.***

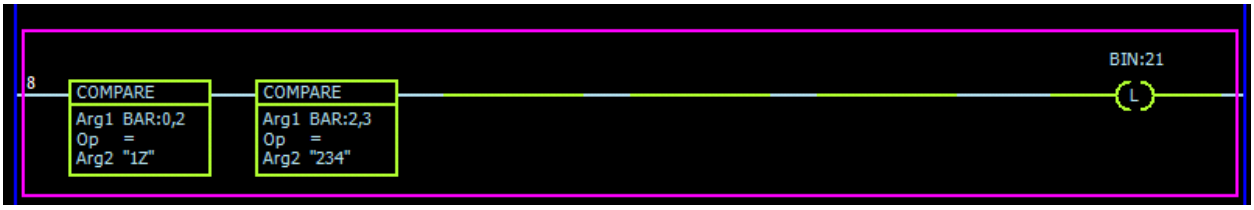


Barcode Comparison

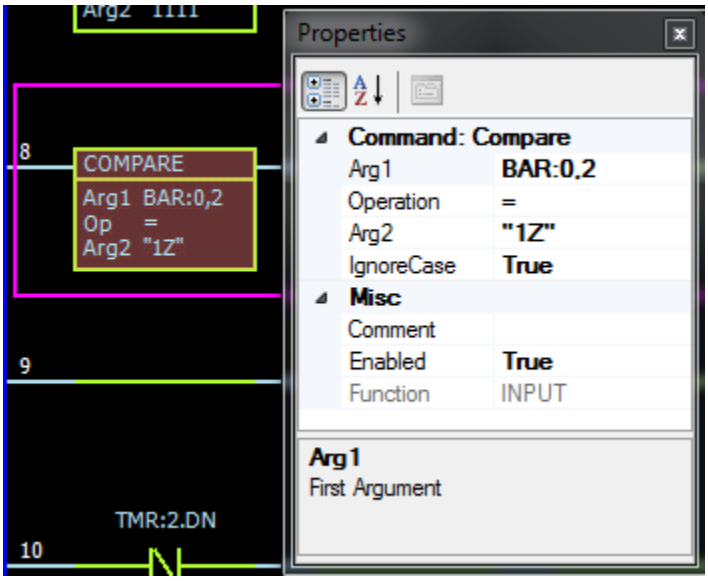
Specific regions of text in the barcode string can be checked via the compare command. See Barcode file type.

NOTE: *This feature requires the "Extended Lookup" activation.*

NOTE: *This method of comparison will default to **FALSE if CB:10 is NOT ACTIVATED**.*



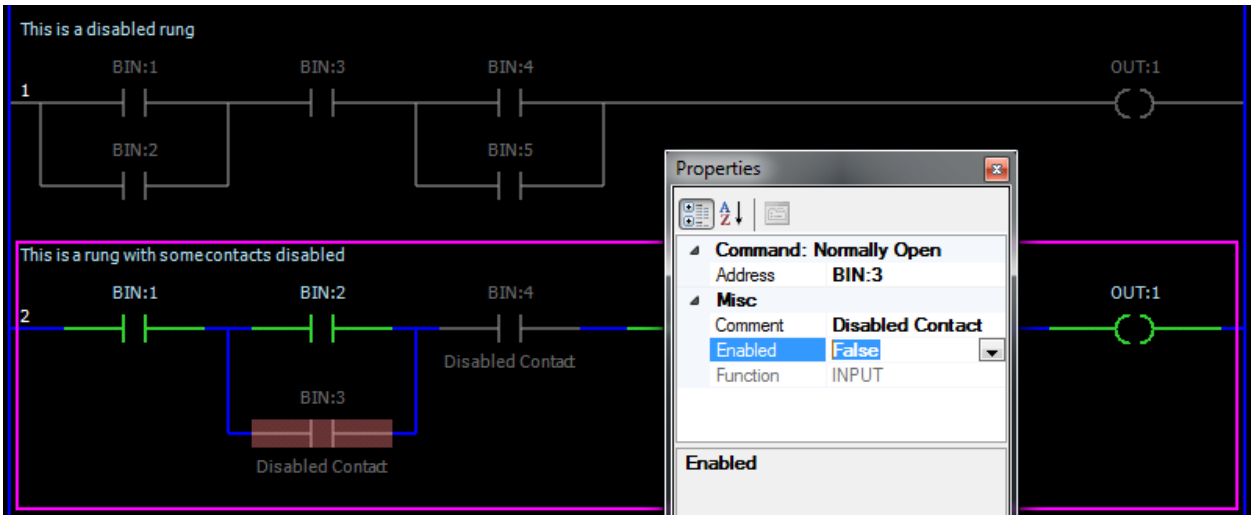
Example:



BAR:0,2 indicates that the first 2 characters of the barcode string are what will be used for the compare operation. Therefore, if a barcode which begins with “1Z”, or “1z” (because *IgnoreCase* has been set to true) is received, then the compare contact will evaluate to an active state.

What do enabled/disabled rungs and contacts do?

By default, all rungs and contacts are enabled for the ladder program. This means that they are included in the ladder code compilation.



The “Enabled” setting for a contact or rung can be toggled from the Property Window. Setting “Enabled” to false disables the contact.

Setting a contact as disabled tells the compiler to ignore this contact and exclude it from the compiled ladder program. Disabled input contacts are treated as short contacts while disabled outputs are ignored. Disabled rungs, on the other hand, tell the compiler to ignore every contact regardless if the contacts themselves are enabled.

This feature comes in handy if you wish to temporarily exclude (without deleting the rung or contact itself) certain elements of the ladder program for debugging purposes.

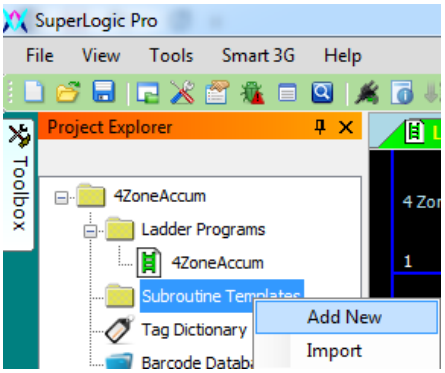
What are Ladder Subroutines and how do I use them?

A Ladder Subroutine is a type of Ladder Program that only needs to be created once and can be used multiple times in a Ladder Program as well as in multiple Ladder Programs. The use of subroutines can greatly minimize programming time due to the use of generic code.

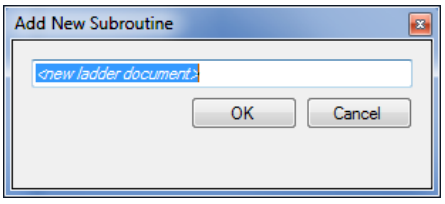
Unlike normal ladder programs, subroutines cannot be directly executed, but rather, they are executed from within a true ladder program. Subroutines have access to all registers and files as the main program.

To use a Ladder Subroutine in an existing Ladder Program, you must first create the Ladder Subroutine Template.

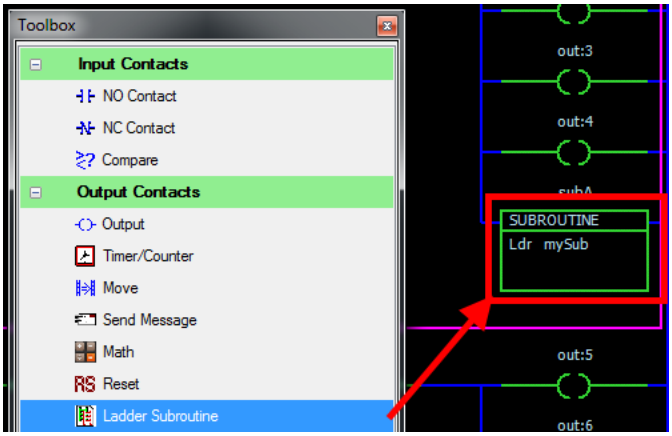
1. Right Click on the "***Subroutine Templates***" folder in the Project Explorer and select "***Add New***"



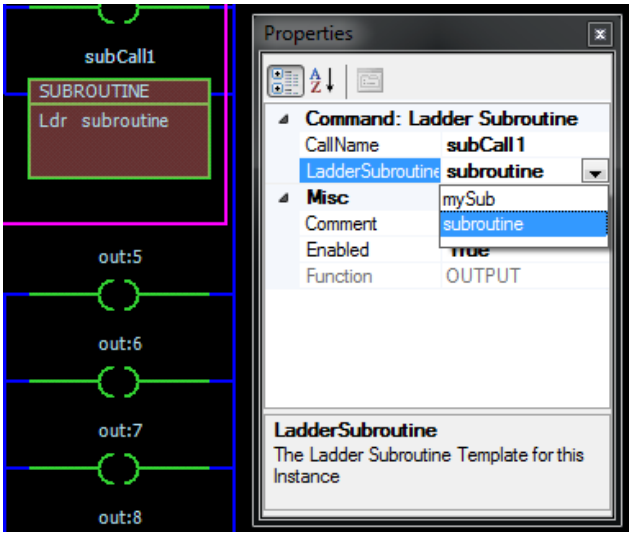
2. Enter the name for your new Subroutine and click OK.



3. After creating your Subroutine Template, you may now create a subroutine instance from within an existing ladder program. This can be done by adding a ladder subroutine command (from the Toolbox).



4. Select the new contact and view its properties. Give your new subroutine instance a new **Call Name** (this value must be unique across the entire project) in the **CallName** property. In the **LadderSubroutine** property, you should see a drop down list of all your Ladder Subroutine Templates. Select the one you wish to use.



5. You can make any number of subroutine instances in your Main Ladder Program now. Any changes you make to the Ladder Subroutine Template will reflect on all its instances.