



Specialists in Information Displays

**Electronic Displays, Inc.  
EDV111 Series LED Signs  
Allen Bradley  
AOI (Add on Instruction)  
Software Manual**

---



**Version Control**

Version	Date	Author	Change Description
1.0	12/01/2012	c.elston	Initial release
1.1	8/30/2012	Szukewich	Modification

---

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b> .....	<b>3</b>
1.1	Supported PLC Controllers .....	3
1.2	Add-On Instructions .....	3
1.3	Supported LED Signs.....	4
<b>2</b>	<b>SYSTEM BLOCK DIAGRAM</b> .....	<b>5</b>
2.1	Typical Connection Diagram .....	5
2.2	Internal EDV111 Connection Diagram.....	6
2.3	Single Sign Connection.....	7
2.4	Multiple Sign Connection (More than 50 Feet apart) .....	7
<b>3</b>	<b>CUSTOMIZE THE IP ADDRESS EIP GATEWAY</b> .....	<b>9</b>
3.1	Install Electronic Displays Network Manager (note: rta to provide EDI with the network manager to be available on EDI's website).....	<b>Error! Bookmark not defined.</b>
<b>4</b>	<b>QUICK START USING TEMPLATE PROGRAM</b> .....	<b>10</b>
4.1	Quick Start with Template Program .....	10
4.2	Update Sign IP Address in Sample Program.....	11
4.3	Update Messages and Sign Format Tags .....	13
4.4	Create and Customize Ladder Logic .....	14
<b>5</b>	<b>IMPORTING WITH NEW PROGRAM OR EXISITING PROGRAM</b> .....	<b>15</b>
5.1	Start a new project with RS Logix 5000 .....	15
5.2	Create a New Ethernet Connection .....	16
5.3	Importing Data-Types.....	20
5.4	Importing Add-on Instructions .....	22
<b>6</b>	<b>USING THE AOI INSTRUCTIONS IN THE PROJECT</b> .....	<b>26</b>
6.1	Add AOIs to Ladder Programming via Drag and Drop .....	26
6.2	Creating Unique Tags for AOI.....	27
6.3	Creating Unique Tags for Sign Parameters.....	28
6.4	Setting up the Sign Tag .....	30

6.5	Mapping Sign Tag to AOI Function Block in Ladder Logic .....	32
<b>7</b>	<b>ELECTRONIC DISPLAY AOI INSTRUCTIONS.....</b>	<b>33</b>
7.1	Send Sign Message AOI.....	33
7.2	Reset Sign AOI .....	34
7.3	Setup Sign Variables AOI .....	34
7.4	Update Sign Variables AOI.....	35
7.5	Expert Sign AOI .....	35

# 1 INTRODUCTION

This manual is provided as a guide for using EVD111 series LED Signs with RS Logix 5000 software by Allen Bradley. This manual provides detailed configuration instructions to configure Ethernet/IP and importing AOIs (add-on-instructions) to RS Logix 5000 software projects.

## 1.1 Supported PLC Controllers

At this time only Allen Bradley CompactLogix and ControlLogix PLC CPUs that use RS Logix 5000 software are supported. Sample projects can be downloaded from the Electronic Displays, Inc. website.

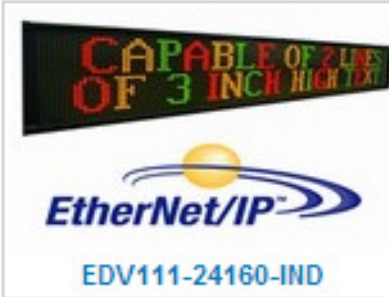
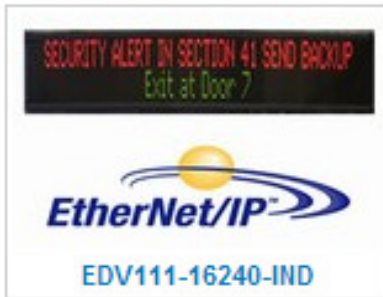
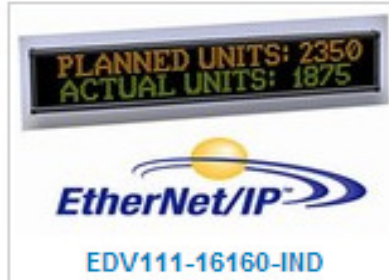
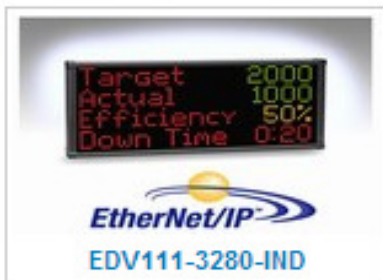
Allen Bradley Micrologix, SLC500 or PLC5 PLC CPUs are **NOT supported** using RS Logix 500 software. Please refer to the ASCII protocol manual for examples. Typically, you will need to connect the serial port from the PLC directly to the LED sign using the DF1 channel 0 port with these types of PLCs.

## 1.2 Add-On Instructions

Add-on instructions provided in this manual are used to make ladder logic based programming very easy. These set of AOIs can be imported into your project and reused in ladder flow.

### 1.3 Supported LED Signs

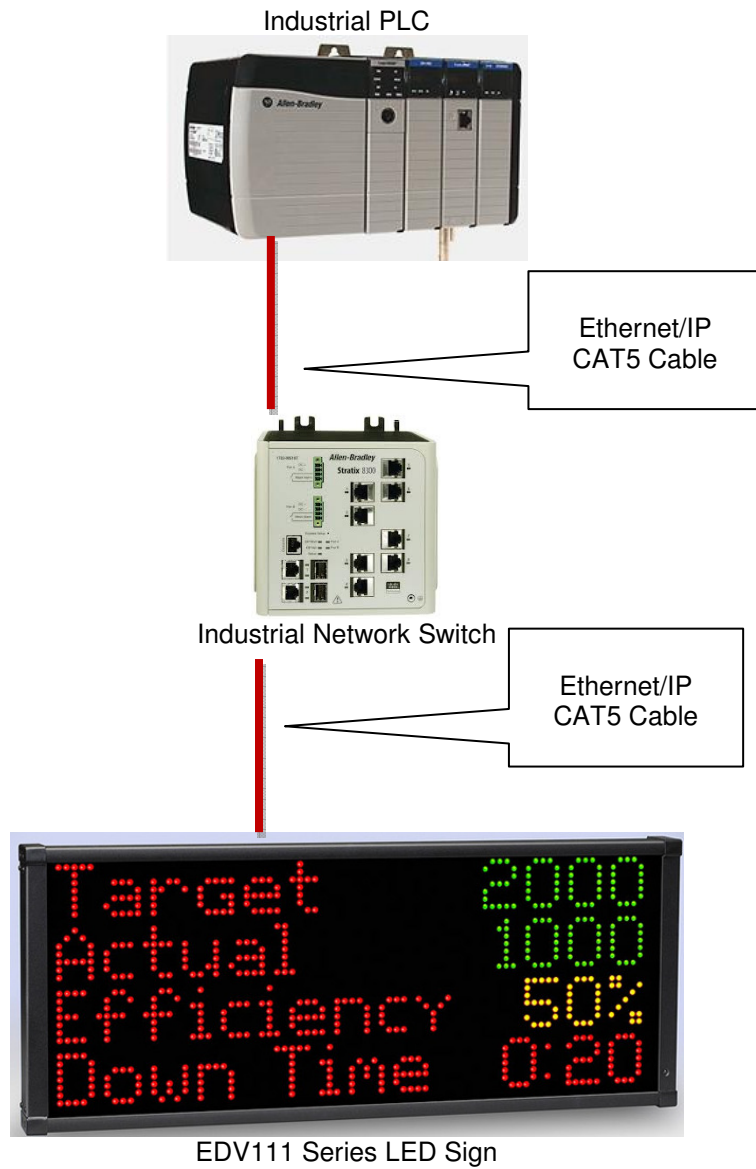
- EDV111-3280-IND, EDV111-16160-IND, EDV111-16128-IND
- EDV111-1680-IND, EDV11132340-IND, EDV111-16240-IND
- EDV111-24160-IND



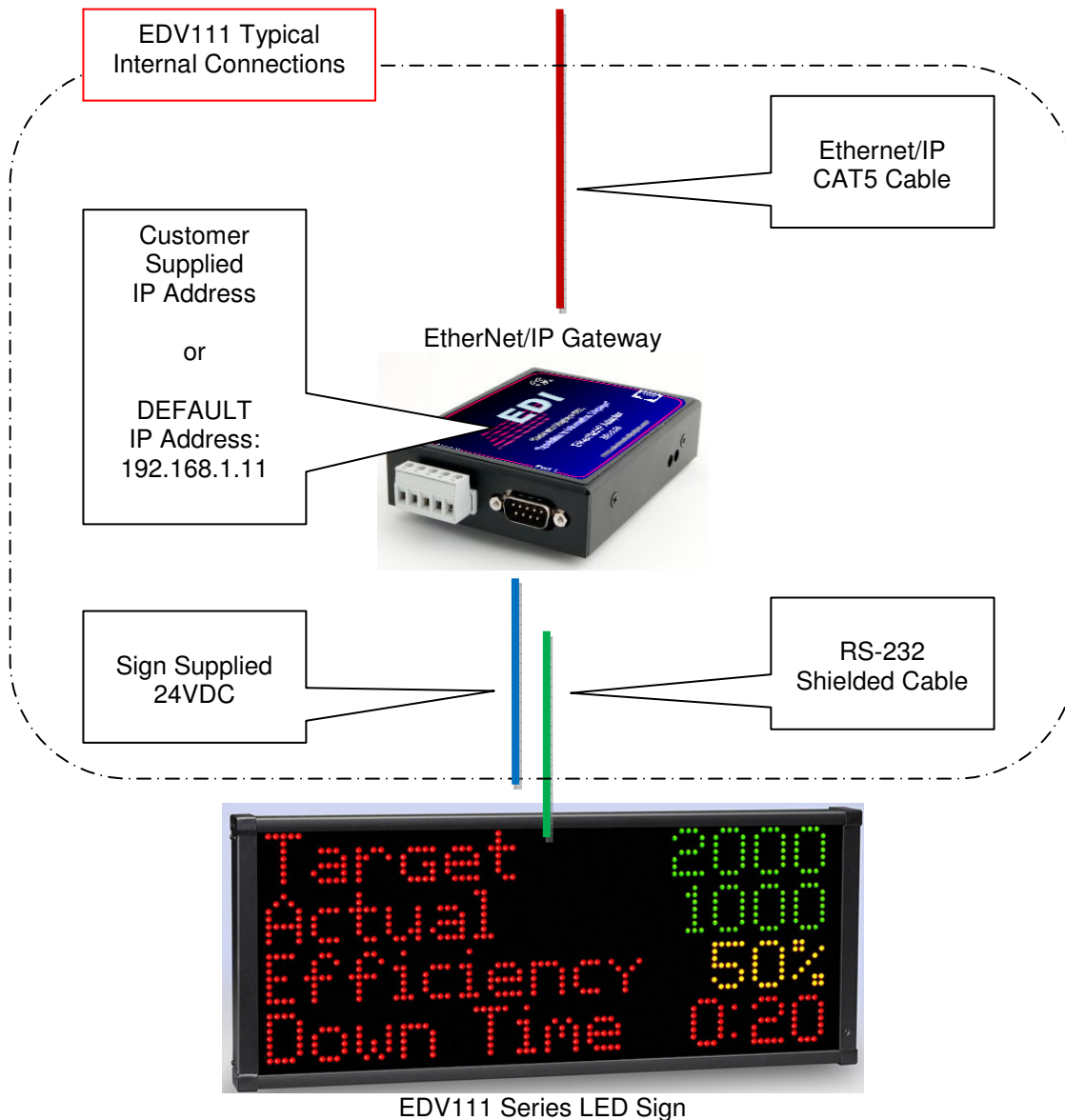
## 2 SYSTEM BLOCK DIAGRAM

EDV111 LED signs come equipped with an Ethernet/IP (EIP) gateway device that exchanges the Ethernet/IP protocol into ASCII serial strings compatible with the LED signs. This allows for the LED signs to be connected via an Ethernet CAT5 cable and not limited to a short distance RS-232 cable typically connected to traditional LED signs displays.

### 2.1 Typical Connection Diagram

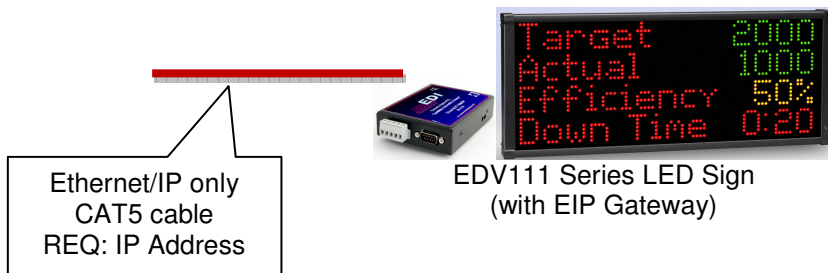


## 2.2 Internal EDV111 Connection Diagram



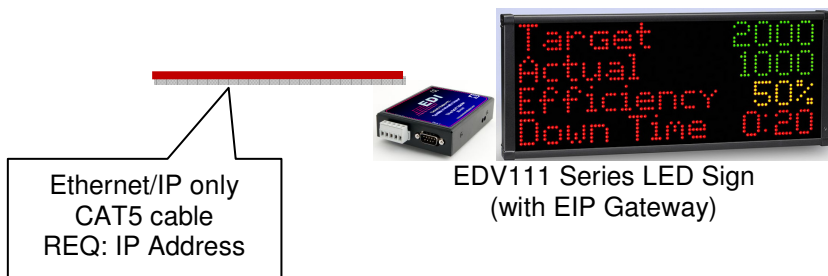
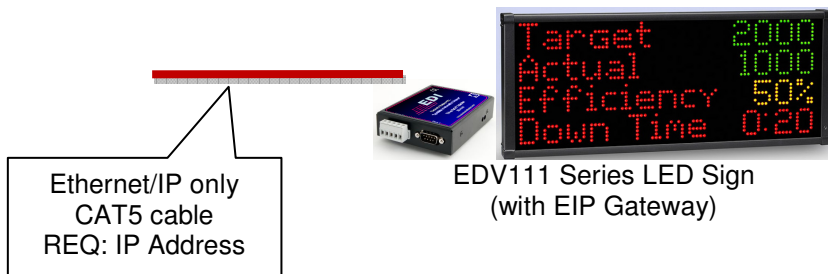
### 2.3 Single Sign Connection

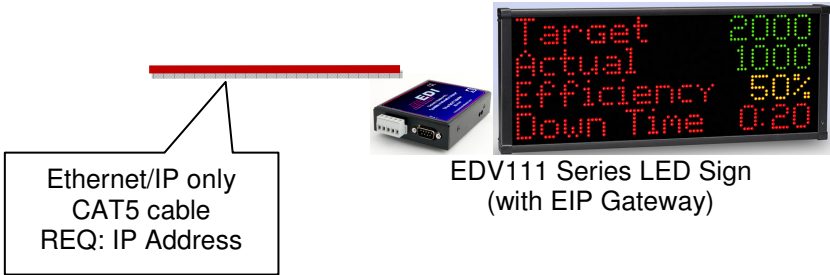
Single sign connection must have an installed "Master" EIP gateway device.



### 2.4 Multiple Sign Connection (More than 50 Feet apart)

Each sign will require an EDV111 to be internally installed as a "Master" EIP gateway device.







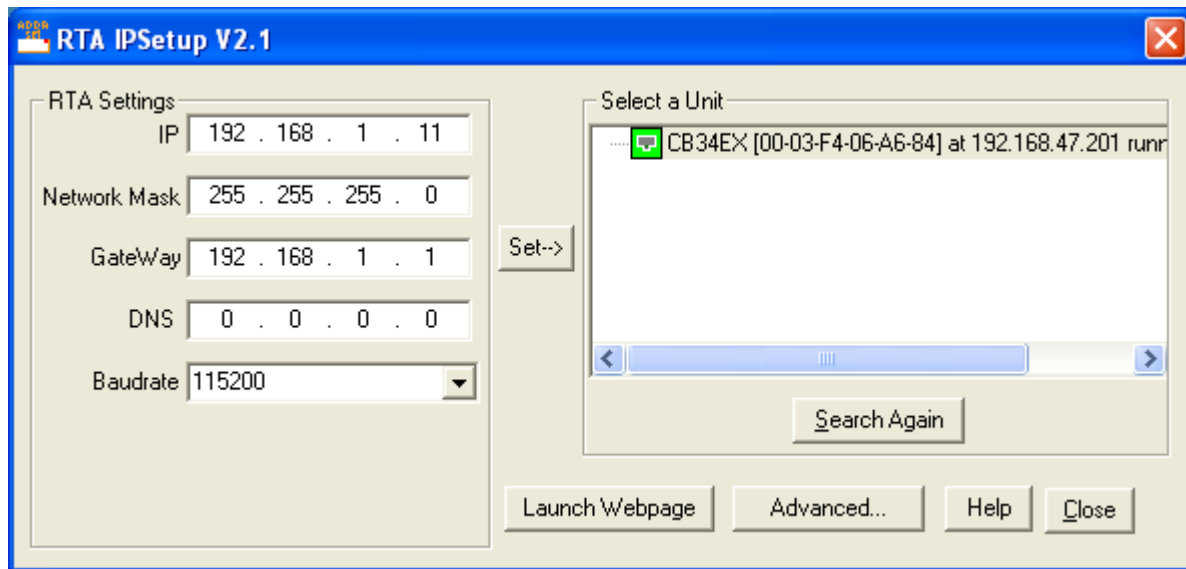
### 3 CUSTOMIZE THE IP ADDRESS EIP Gateway

#### 3.1 Install Electronic Displays Network Manager

Download and install the EDI Network Manager software from EDI website Before the device can be configured, the gateway's network settings must be set.

Location: <http://www.electronicdisplays.com/> → [Support](#) → [Downloads](#) → [Allen Bradley](#) → **ED3600 Network Manager**

- 1) If the PC is currently setup with DHCP turned on, turn off DHCP and set a static IP and corresponding Subnet Mask for your PC.
- 2) Connect the 7-30 VDC power source to the device.
- 3) Using the supplied crossover cable, connect the device to the PC.



- 4) RTA Settings: IP Address is set to 192.168.1.11 and Subnet Mask is set to 255.255.255.0 by default.
- 7) Configure the IP Address and the Subnet Mask so that it matches your PC's network settings.
- 8) Click **Set->**. This will restart the gateway.
- 9) Under Select a Unit, the gateway will come back online. When visible again, highlight and click **Launch Webpage**.

- 10) If gateway does not reappear under Select a Unit, click **Search Again** and repeat step 10. If problems continue, jump to the Troubleshooting section. Otherwise, you may continue with your normal gateway configuration.

## QUICK START USING TEMPLATE PROGRAM

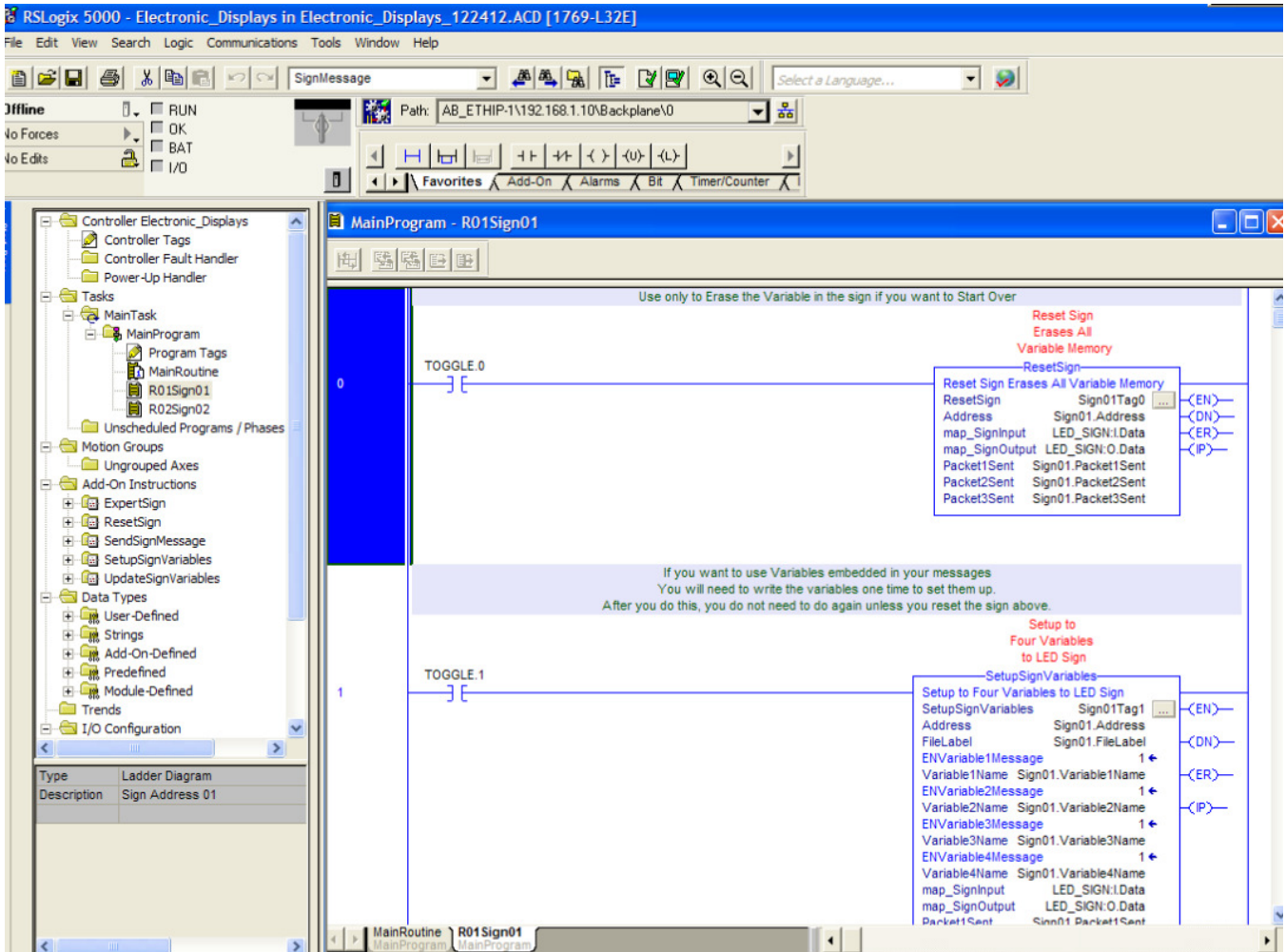
Download the sample PLC program from the Electric Displays website.

**Electronic\_Displays\_122412.acd**  
(or latest version)

### 4.1 Quick Start with Template Program

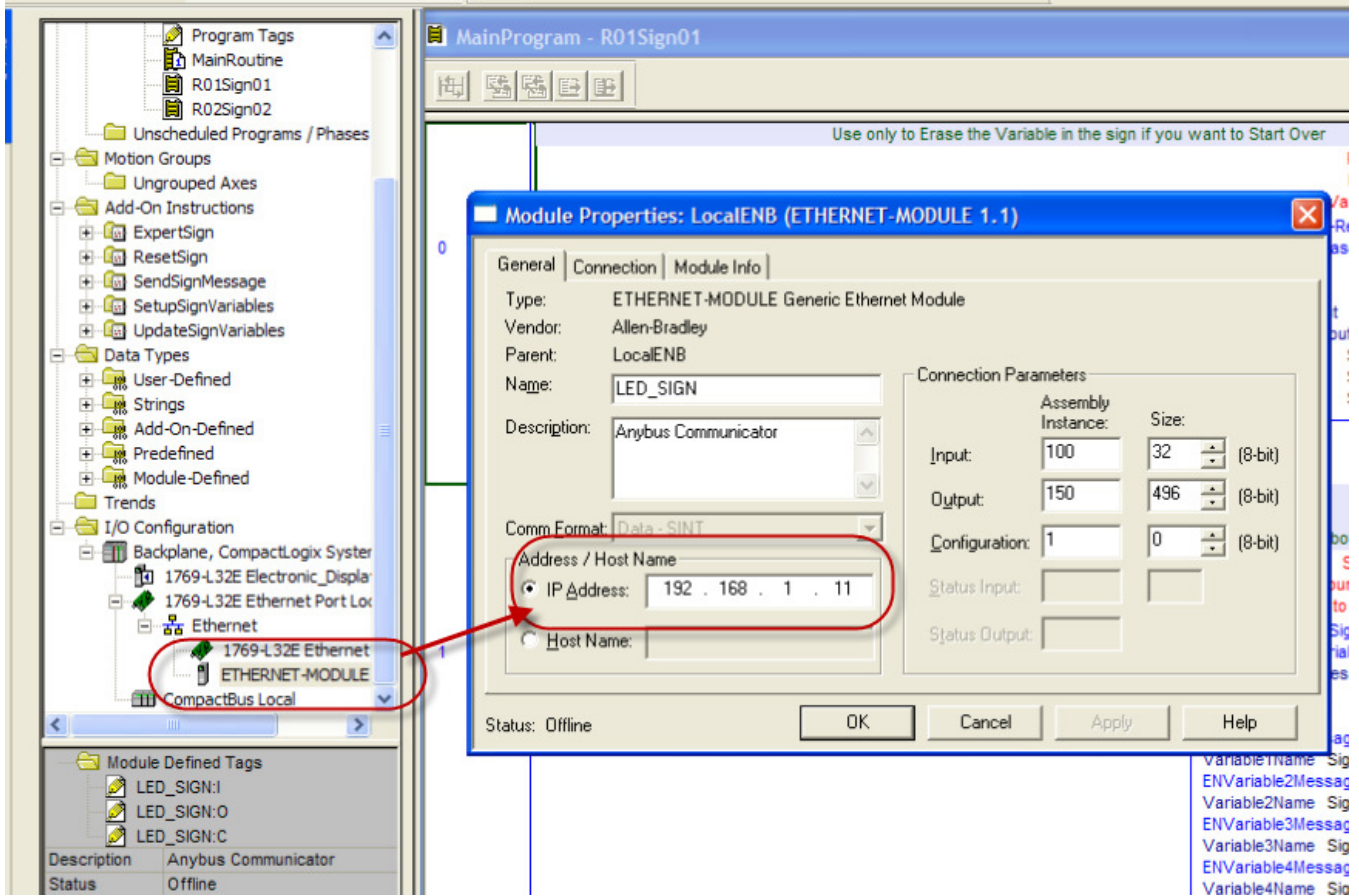
The purpose of the template sample program is provide a bases of settings and tags that are setup in the PLC along with sample ladder logic that can be written to send messages to the sign.

This template file is design to communicate with two signs. Sign #1 is a master sign which is connected via an RS-232 cable daisy chained to Sign #2. **Is this really an RS232 Daisy Chain?**



## 4.2 Update Sign IP Address in Sample Program

Open the Ethernet Module setting and update the IP Address of the sample program. The sample program is setup with the default sign IP address of 192.168.1.11. If you customized your IP, you will need to update this target address below.



### 4.3 Update Messages and Sign Format Tags

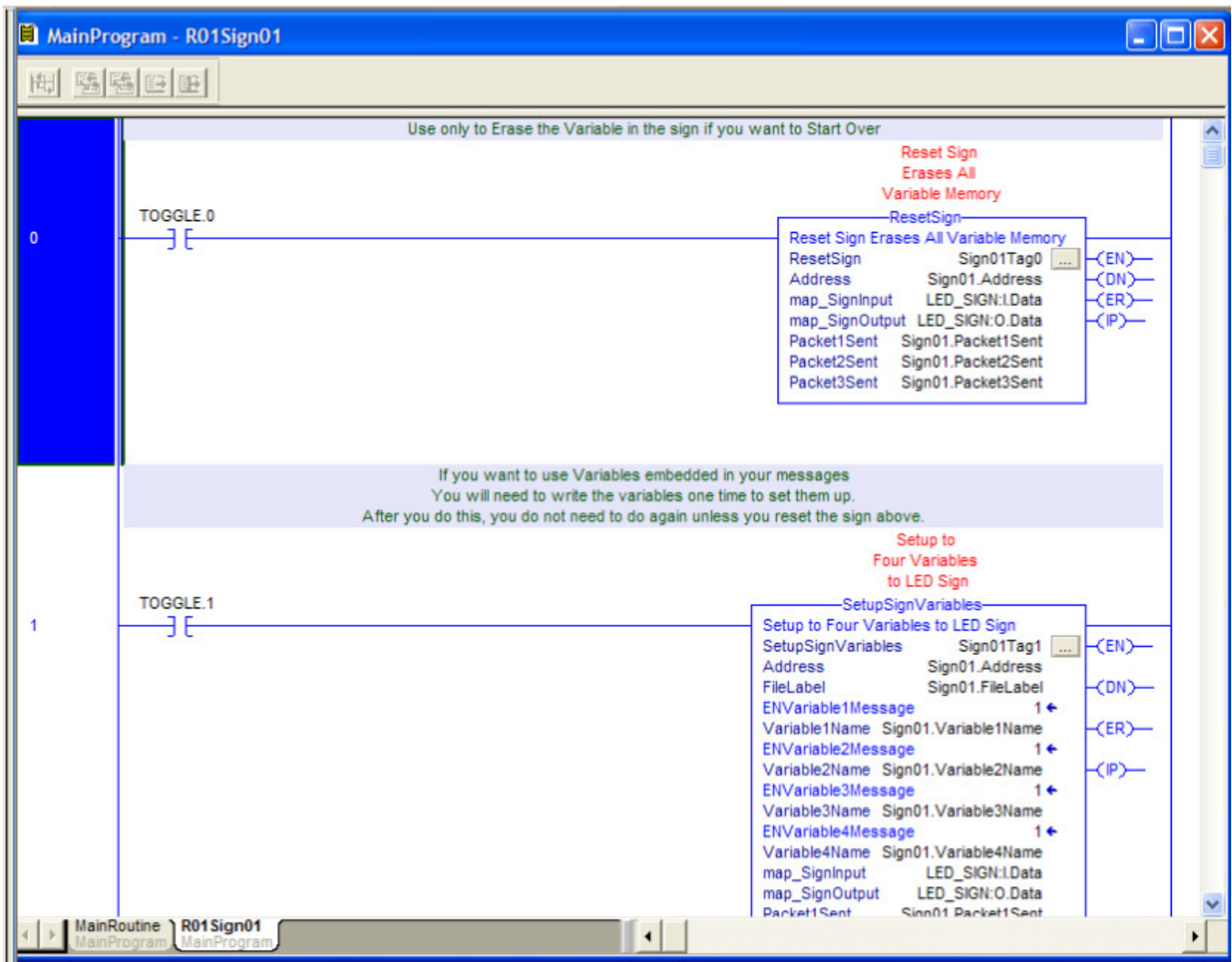
Open the Controller Tags and update the Sign01 and Sign02 tags with the message and formatting you desire. See the description or AOI help file to determine which options are available.

The screenshot shows the AOI software interface. On the left, a tree view shows the project structure, with 'Controller Tags' selected under 'Controller Electronic\_Displays'. The main window, titled 'Controller Tags - Electronic\_Displays(controller)', displays a table of tags for 'Sign01'. A red box highlights the 'Sign01' tag and its sub-items. A red arrow points from the 'Controller Tags' folder in the tree view to the 'Sign01' tag in the table.

Name	Value	Style	Data Type	Description
DEEScore	99.82241	Float	REAL	
Sign01	{...}		LEDSIGN	
+ Sign01.Address	'01'		STRING	"00"=all signs "01"=sign address 1 "02"=sign address 2 etc... per protocol (must be t...
+ Sign01.CommandCode	'A'		STRING	'A'=write restart 'B'=write variable 'C'=special 'D'=write no restart
+ Sign01.FileLabel	'A'		STRING	Default is = A
+ Sign01.Text1Message	'Total '		STRING	Actual Text Line 1
+ Sign01.Text2Message	'Rejects '		STRING	Actual Text Line 2
+ Sign01.Text3Message	'Cycle Time '		STRING	Actual Text Line 3
+ Sign01.Text4Message	'OEE Score '		STRING	Actual Text Line 4
+ Sign01.TextFont	'0'		STRING	'0'=SS7 '1'=SF7 '2'=SF10 '3'=SS16 '4'=SF16
+ Sign01.TextColor	'1'		STRING	'0'=Red '1'=Green '2'=Yellow '3'=Rainbow
+ Sign01.TextAlign	'L'		STRING	'M'=middle line 'T'=top line 'B'=bottom line 'F'=fill (best for variables) 'L'=left 'R'=r...
+ Sign01.TextAttribute	'0'		STRING	'0'=flashing off '1'=flashing on '2'=wide off '3'=wide on '4'=bold off '5'=bold on
+ Sign01.TextEffect	'H'		STRING	'S'=scrolls 'H'=hold 'F'=flash 'A'=slide up 'B'=slide down 'C'=slide left 'D'=slide r...
+ Sign01.TextSpeed	'3'		STRING	Options = 1 - 8, 3:Default 1=Fast 8=Slow
+ Sign01.TextPause	'02'		STRING	Options = 00 - 99, 02:Default defined in seconds (must be two digits) if not will not work!
+ Sign01.Variable1Name	'U'		STRING	Name of variable Default is = U
+ Sign01.Variable2Name	'V'		STRING	Name of variable Default is = V
+ Sign01.Variable3Name	'W'		STRING	Name of variable Default is = W
+ Sign01.Variable4Name	'X'		STRING	Name of variable Default is = X
+ Sign01.Variable1Value	'10136'		STRING	Actual Variable1 to be Sent (text format)
+ Sign01.Variable2Value	'18'		STRING	Actual Variable2 to be Sent (text format)
+ Sign01.Variable3Value	'0.763'		STRING	Actual Variable3 to be Sent (text format)
+ Sign01.Variable4Value	'99.82'		STRING	Actual Variable4 to be Sent (text format)
+ Sign01.VariableFont	'0'		STRING	'0'=SS7 '1'=SF7 '2'=SF10 '3'=SS16 '4'=SF16
+ Sign01.VariableColor	'0'		STRING	'0'=Red '1'=Green '2'=Yellow '3'=Rainbow
+ Sign01.VariableAttribute	'4'		STRING	'0'=flashing off '1'=flashing on '2'=wide off '3'=wide on '4'=bold off '5'=bold on
+ Sign01.Packet1Sent	'*B01*ABU10136*C*B...'		STRING	The actual packet sent to the sign. (Packet1)
+ Sign01.Packet2Sent	'*'		STRING	The actual packet sent to the sign. (Packet2)

### 4.4 Create and Customize Ladder Logic

Create ladder logic to enable the rungs in sequence to send message to the sign. If this is the first time you have connect the PLC to the sign, enable the Reset Sign function block, which erases all variable date allocated in the sign, then enable the SetupSignVariables which will allocated memory space in the sign to accept dynamic variables that can be written to the sign using the UpdateSignVariable AOI. Send message to the sign by enabling the SendSignMessage AOI.



## 5 IMPORTING WITH NEW PROGRAM OR EXISITING PROGRAM

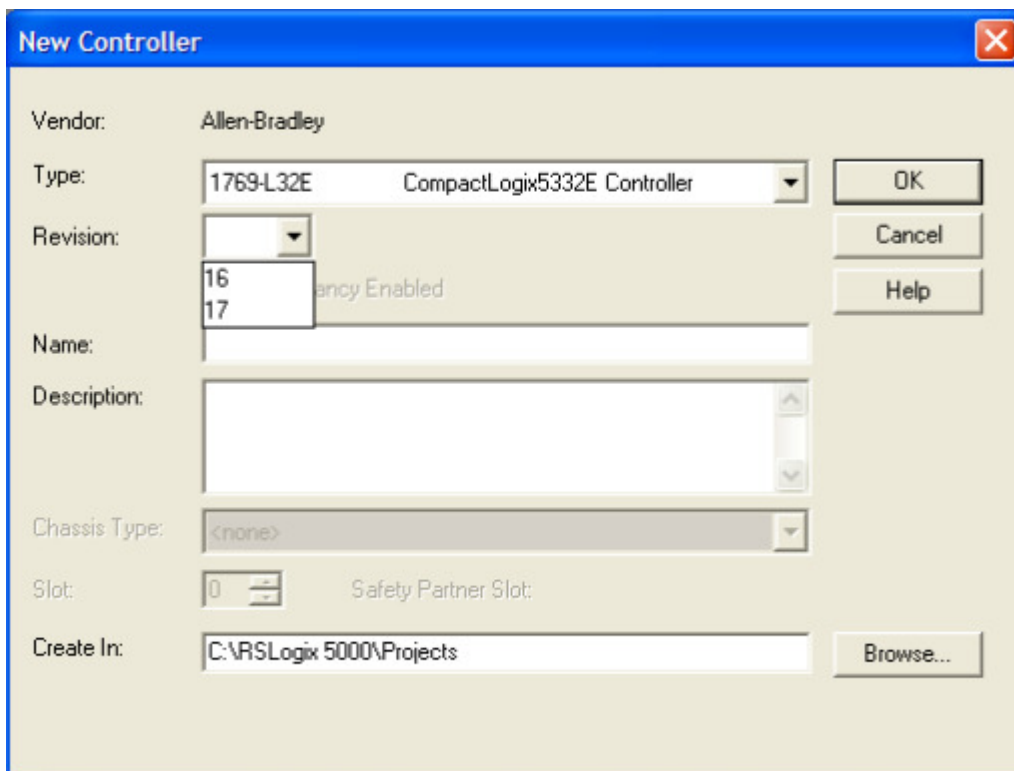
### 5.1 Start a new project with RS Logix 5000

Click File, New Project to start a new PLC project.

Choose PLC Type.

Choose PLC firmware revision.

Name your PLC Project.



The screenshot shows the 'New Controller' dialog box with the following fields and options:

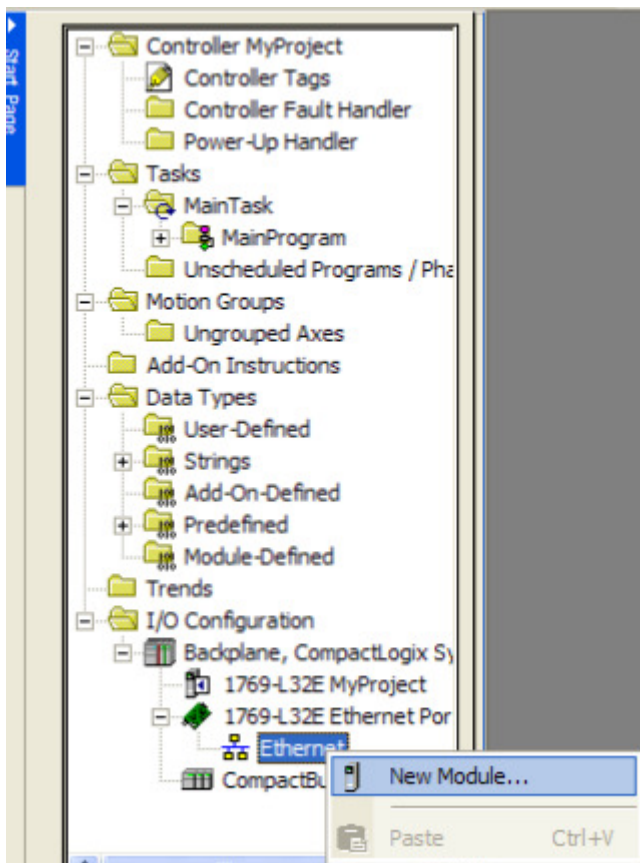
- Vendor:** Allen-Bradley
- Type:** 1769-L32E CompactLogix5332E Controller
- Revision:** 16 (with a dropdown menu showing 16 and 17, and a note 'Safety Enabled')
- Name:** (empty text field)
- Description:** (empty text area)
- Chassis Type:** <none>
- Slot:** 0 (with a dropdown menu) and Safety Partner Slot: (empty)
- Create In:** C:\RSLogix 5000\Projects (with a 'Browse...' button)
- Buttons:** OK, Cancel, Help

## 5.2 Create a New Ethernet Connection

In the controller tree view.

Right click the Ethernet ICON

Choose New Module

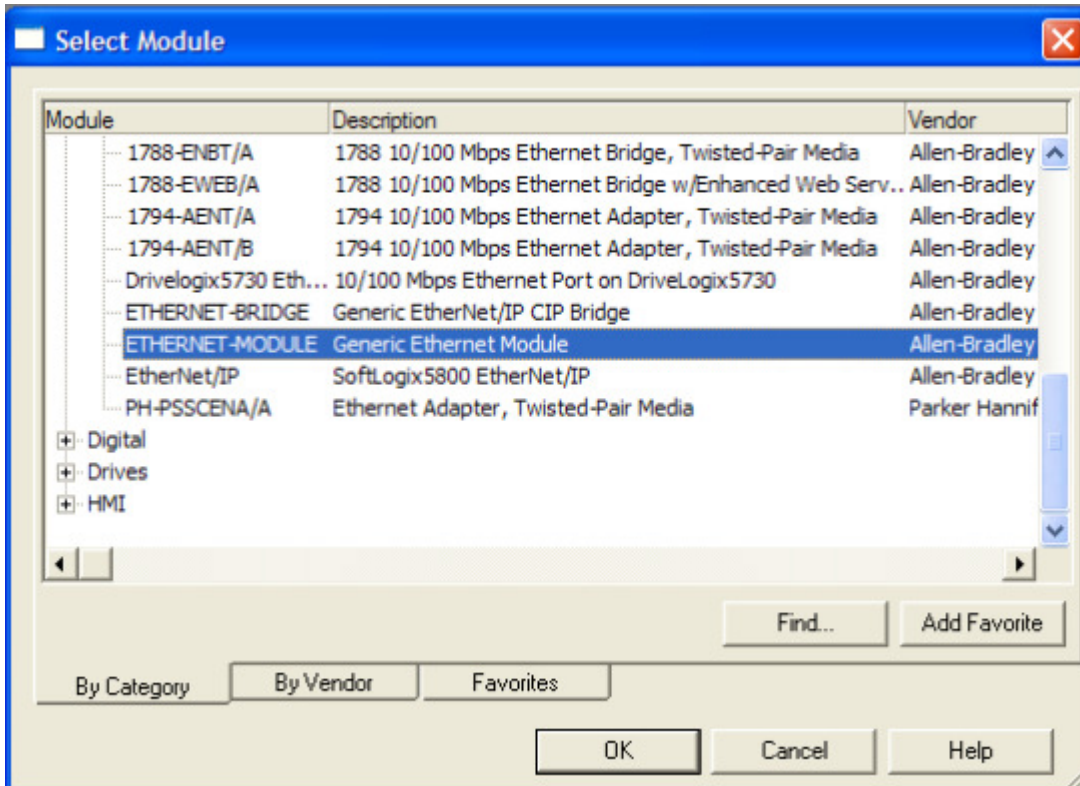




Choose Communication.

Choose ETHERNET-MODULE (Generic Ethernet Module)

Click OK



Name the Ethernet Connection:

**LED\_SIGN**

(This will be the prefix name of the tags in the controller.)

Enter the EIP gateway Default IP address or your custom IP address:

**192.168.1.11**

or

**custom IP address**

**(xxx.xxx.xxx.xxx)**

Choose Comm Format

**Data-SINT**

(Important)

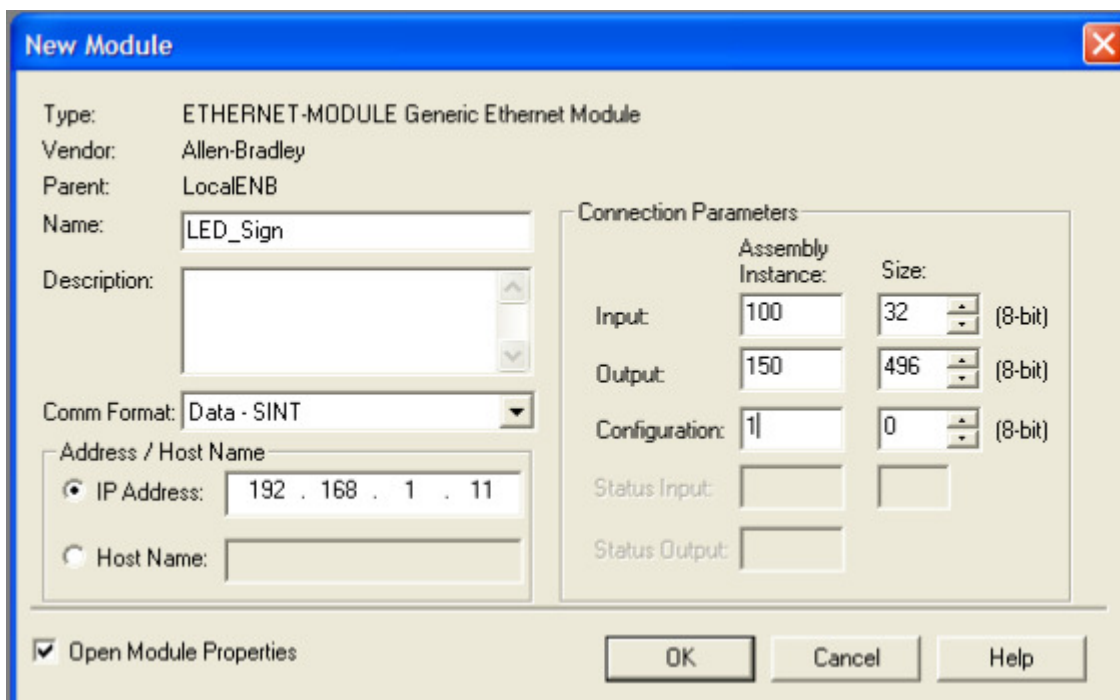
Enter Required Assembly Instance

**Input: 100 and 32 bytes**

**Output: 150 and 496 bytes**

**Configuration: 1 and 0 bytes**

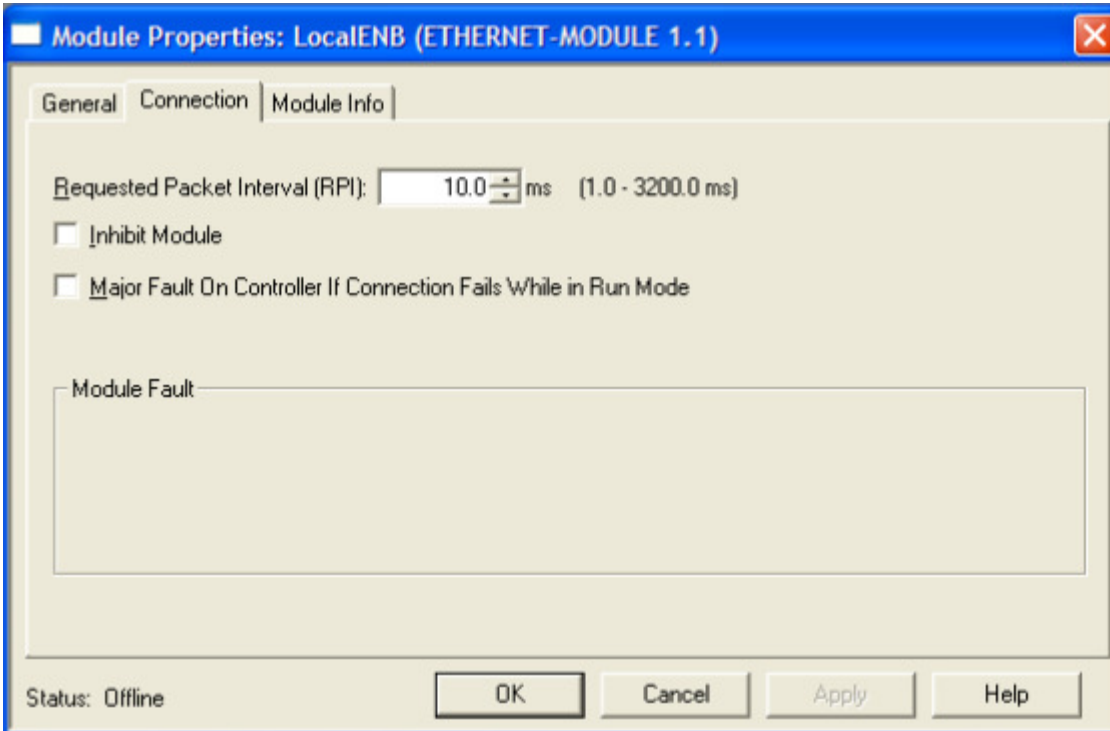
Click OK



Choose RPI interval:

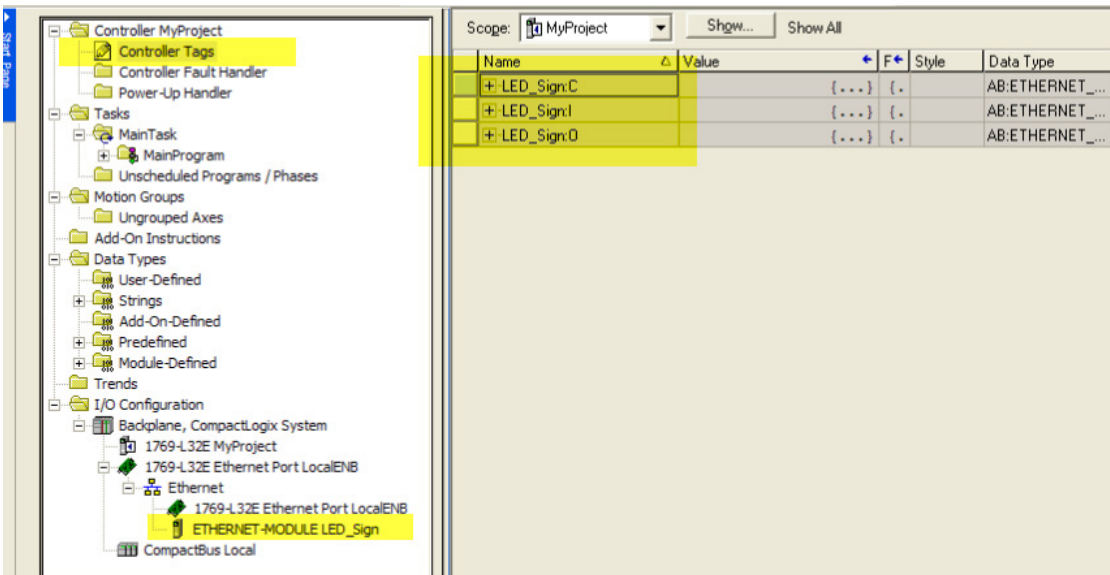
Default 10.0 ms is ok

Click OK



Confirm Controller Tags

Confirm Ethernet Module is configured

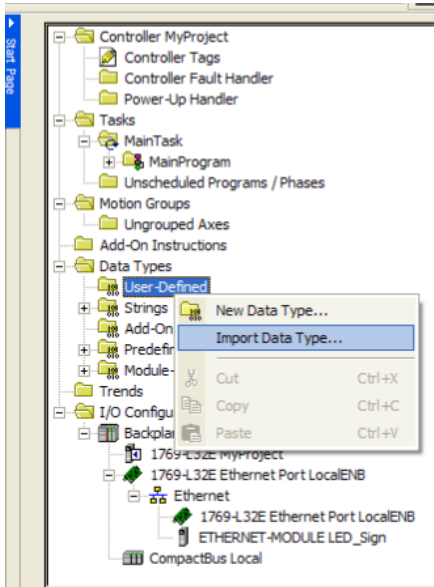


### 5.3 Importing Data-Types

In the controller tree view

Right click User-Defined under "Data Types"

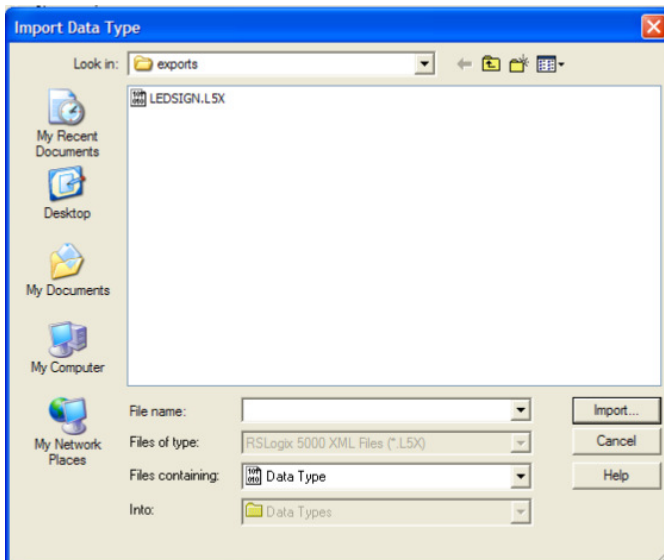
Choose Import Data Type



Browse to the folder containing Data Type

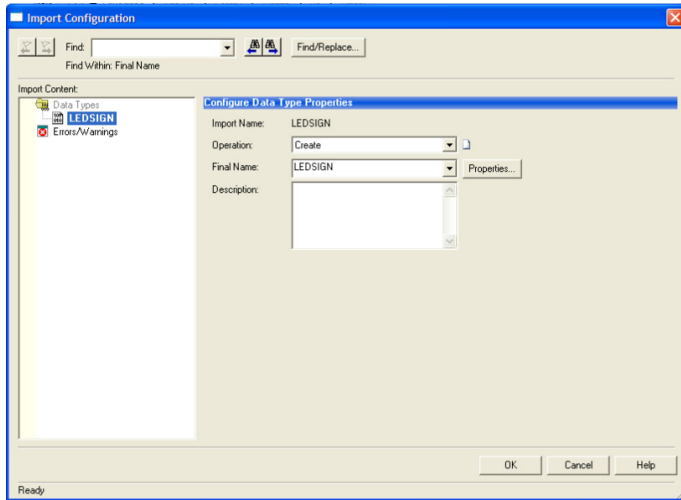
Import LEDSIGN.L5X file

Click OK



Confirm no version conflicts

Click OK



Confirm Data Type "LEDSIGN"

Name: LEDSIGN

Description:

Members: Data Type Size: 3080 byte(s)

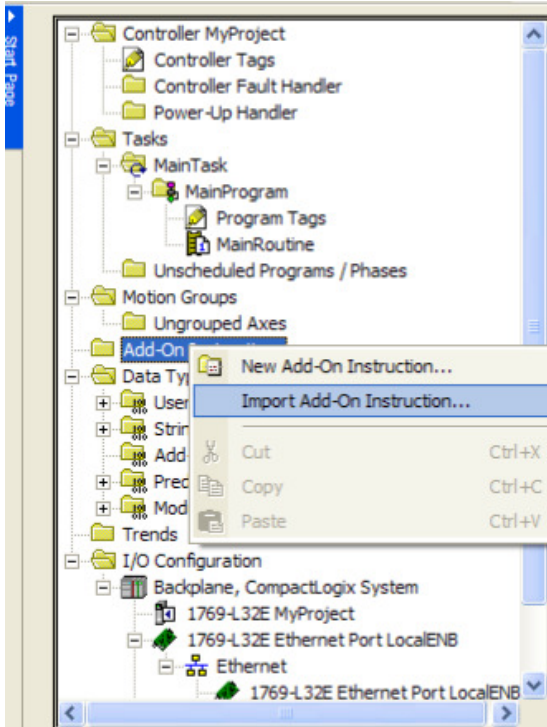
Name	Data Type	Style	Description
Address	STRING		Options = 00,01,02 "00"=all signs "01"=sign
CommandCode	STRING		Options = A,B,C,D per protocol "A"=write te Default is = A
FileLabel	STRING		
Text1Message	STRING		Actual Text Line 1
Text2Message	STRING		Actual Text Line 2
Text3Message	STRING		Actual Text Line 3
Text4Message	STRING		Actual Text Line 4
TextFont	STRING		Options = 0,1,2,3,4 per protocol "0"=SS7 "
TextColor	STRING		Options = 0,1,2,3 per protocol 0=Red 1=Gr
TextAlign	STRING		Options = 0,1,2,3,4,5,6,7,8 per protocol "0"
TextEffect	STRING		Options = S,H,F per protocol "S"=scrolls "H
TextSpeed	STRING		Options = 1 - 8, 3:Default per protocol 1=Fa
TextPause	STRING		Options = 00 - 99, 02:Default per protocol c
Variable1Name	STRING		Name of variable Default is = U
Variable2Name	STRING		Name of variable Default is = V
Variable3Name	STRING		Name of variable Default is = W
Variable4Name	STRING		Name of variable Default is = X
Variable1Value	STRING		Actual Variable1 to be Sent (text format)
Variable2Value	STRING		Actual Variable2 to be Sent (text format)
Variable3Value	STRING		Actual Variable3 to be Sent (text format)

## 5.4 Importing Add-on Instructions

In the controller tree view

Right click Add-On Instruction

Choose Import Add-On Instruction

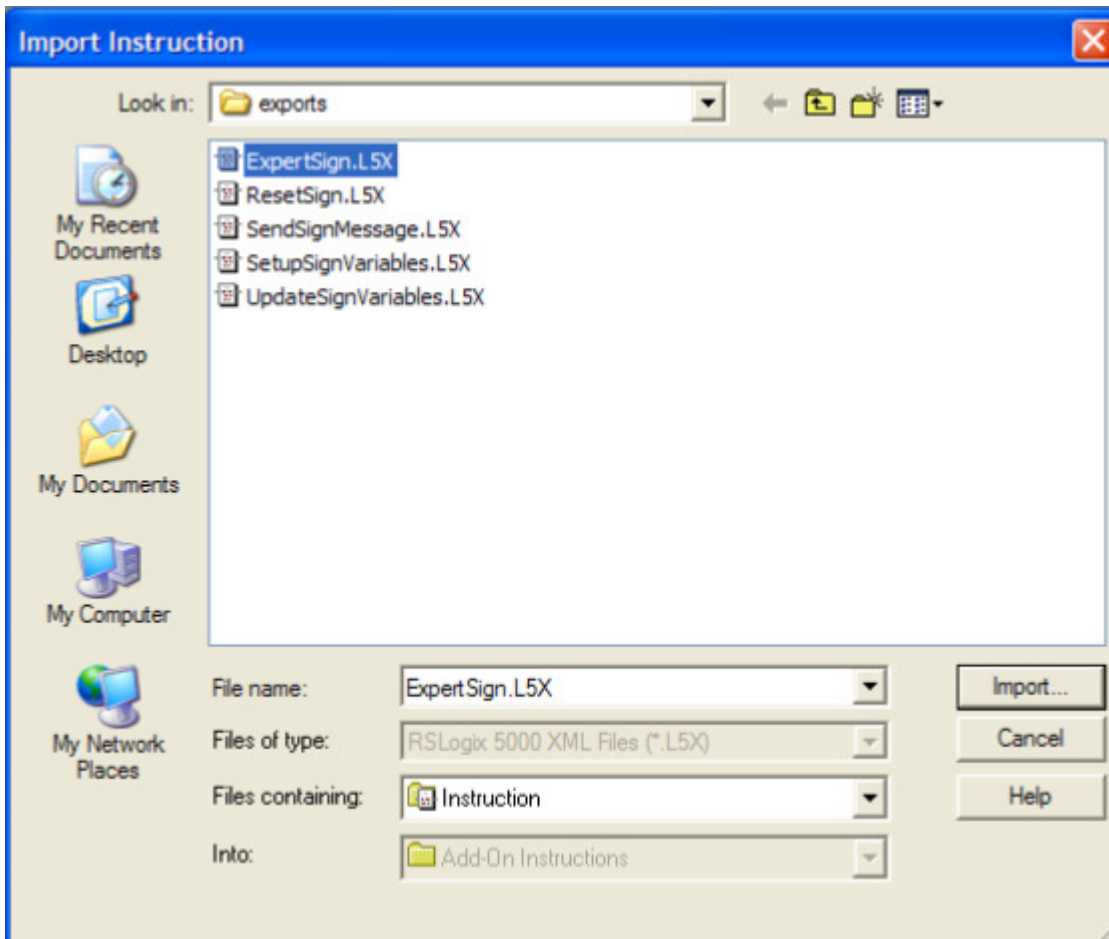


Browse to the folder containing Add-On Instructions

Import all files with .L5X extension.

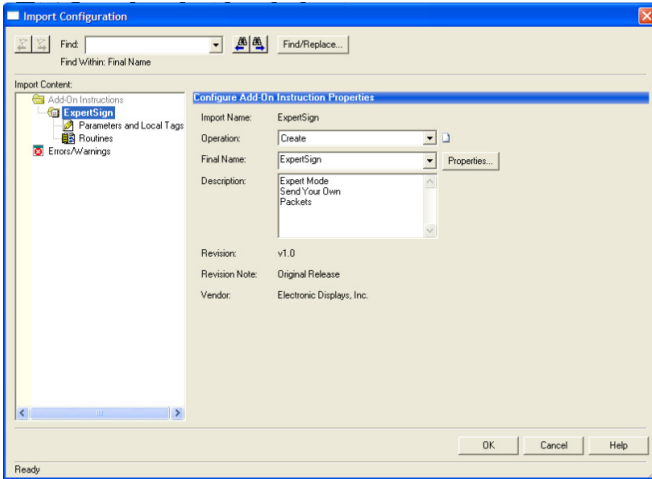
Do this process until all add-on instructions are imported.

Click OK

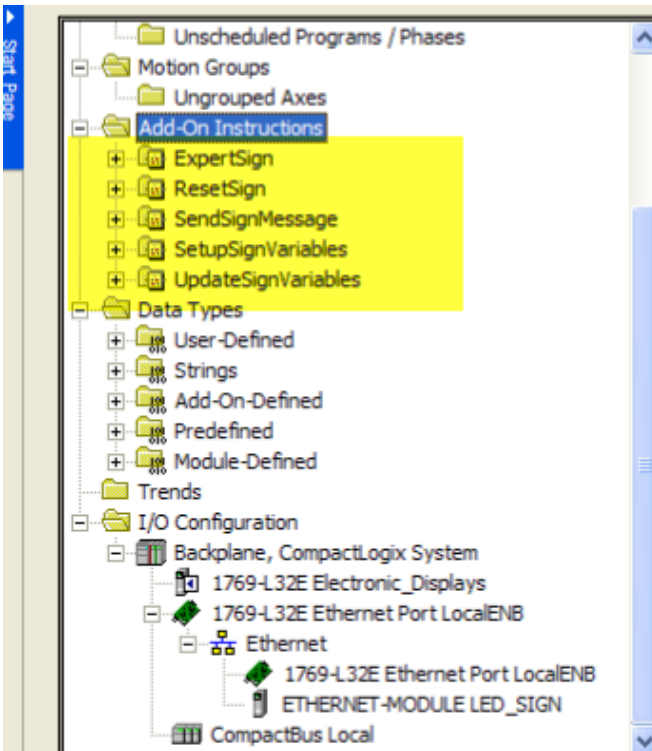


Confirm no version conflicts

Click OK

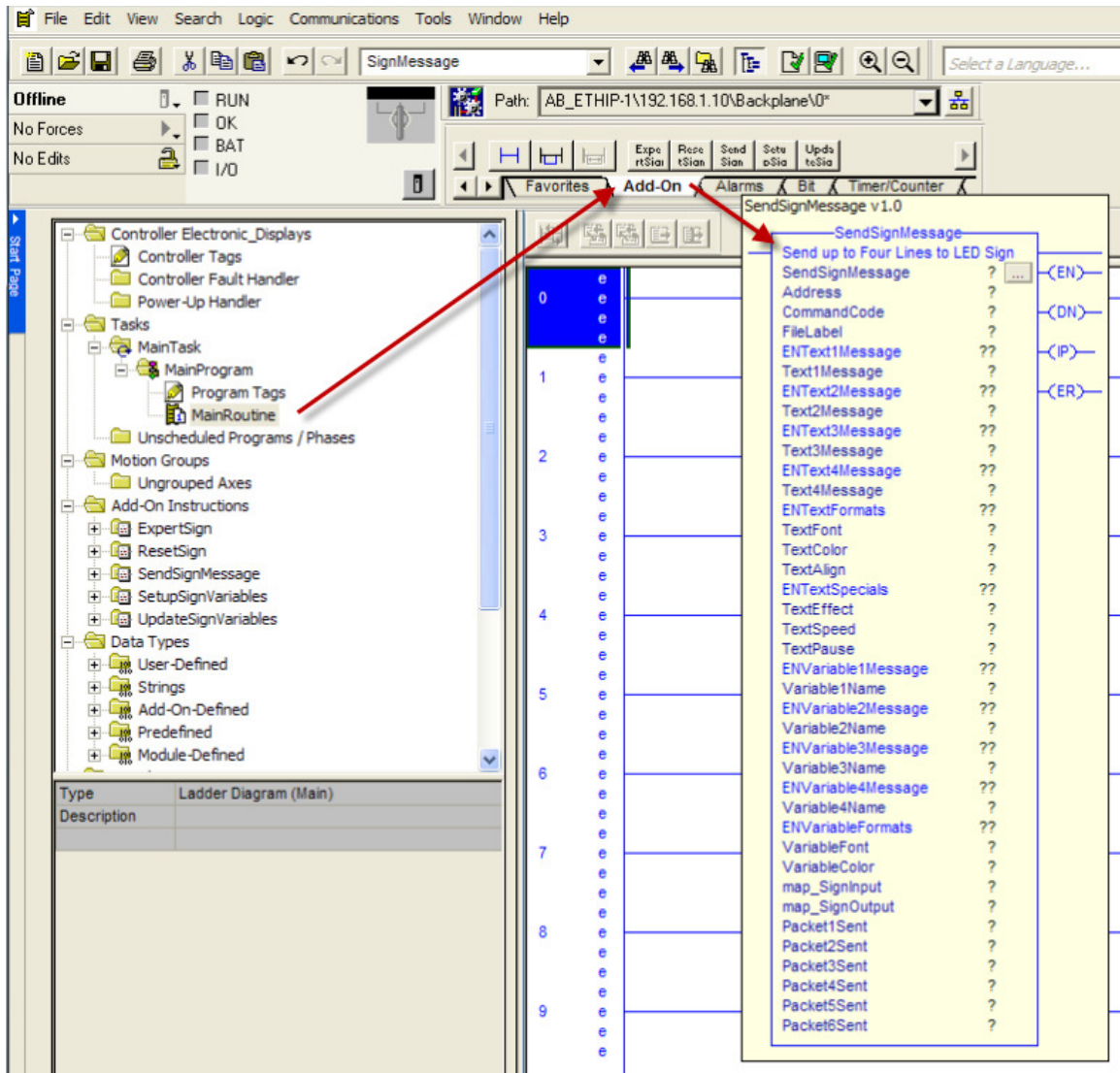


Confirm all Add-on Instructions are imported





Confirm AOIs are added to Toolbar in RS Logix 5000



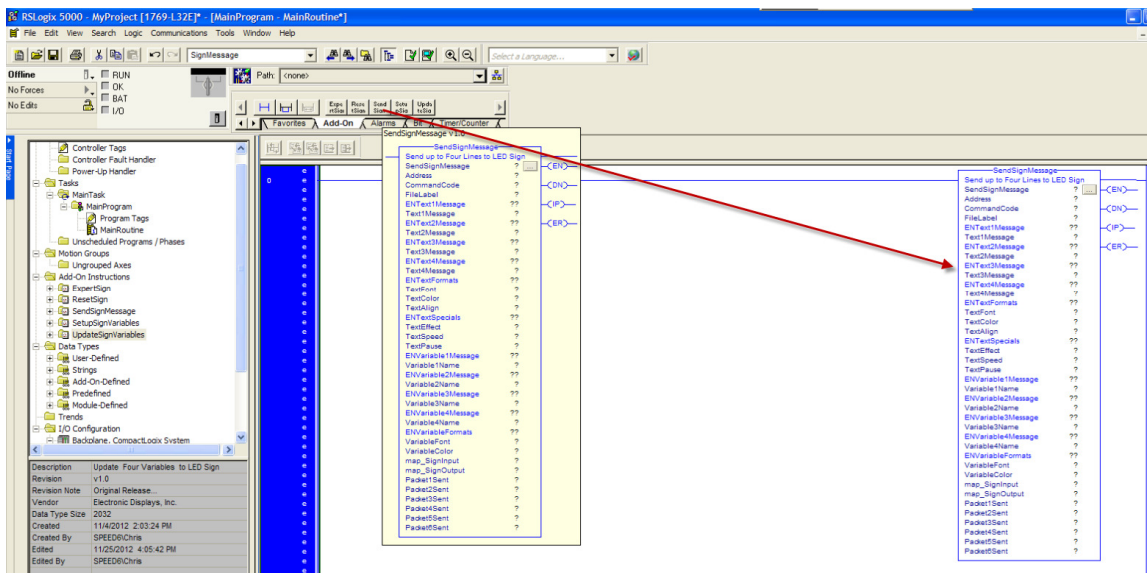
## 6 USING THE AOI INSTRUCTIONS IN THE PROJECT

### 6.1 Add AOIs to Ladder Programming via Drag and Drop

Click on the Add-On Toolbar

Drag and Drop the desired control AOI block to a new rung

**TIP: You can also drag and drop from the Add-On Menu Tree on the left as well**

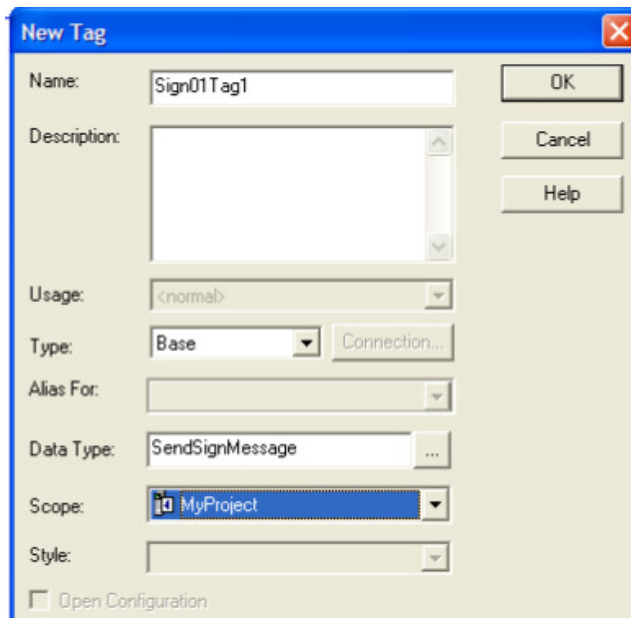
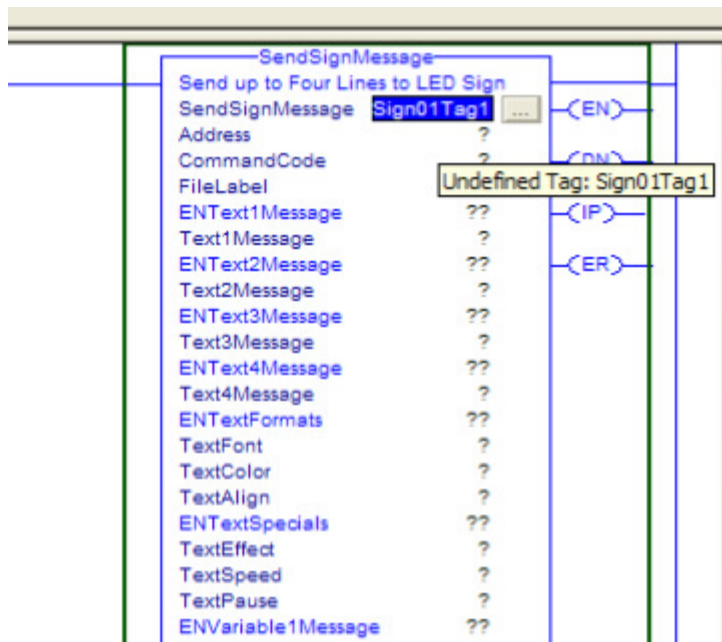


## 6.2 Creating Unique Tags for AOI

Under the AOI "Tag" parameter, begin to type a desired tag name.

Best Practice here might be to name the tag with the sign address. In this case, the default sign address is "01". Sign01Tag1 might be an example.

Be sure you scope your tag properly.

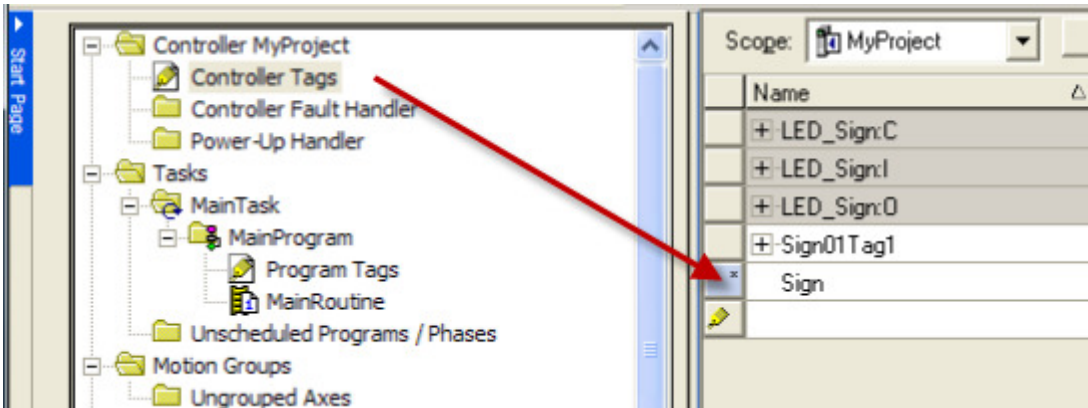


### 6.3 Creating Unique Tags for Sign Parameters

Each sign, needs a "parameter" file which it retrieves all information and settings from the PLC to the sign.

Best practice would be to create a tag with the name of your sign.

Choose LEDSIGN as the



Name	Alias For	Base Tag	Data Type	Style
+ LED_Sign:C			AB:ETHERNET_...	
+ LED_Sign:I			AB:ETHERNET_...	
+ LED_Sign:O			AB:ETHERNET_...	
+ Sign01Tag1			SendSignMessage	
* Sign			DINT	Decimal



**Map the DATA TYPE of your "Sign" tag to the LEDSIGN data type.**  
 (This data type was imported earlier)

Scope: MyProject Shgw... Show All

Name	Alias For	Base Tag	Data Type	Style
+ LED_Sign:C			AB:ETHERNET_...	
+ LED_Sign:I			AB:ETHERNET_...	
+ LED_Sign:O				
+ Sign01Tag1				
* Sign				

**Select Data Type**

Data Types:

LEDSIGN

FUNCTION\_GENERATOR

HL\_LIMIT

IMC

INT

INTEGRATOR

LEAD\_LAG

LEAD\_LAG\_SEC\_ORDER

**LEDSIGN**

LIGHT\_CURTAIN

Array Dimensions:

Dim 2: 0 Dim 1: 0 Dim 0: 0

Show Data Types by Groups

OK Cancel Help

Scope: MyProject Shgw... Show All

Name	Alias For	Base Tag	Data Type
+ LED_Sign:C			AB:ETHERNET_...
+ LED_Sign:I			AB:ETHERNET_...
+ LED_Sign:O			AB:ETHERNET_...
+ Sign01Tag1			SendSignMessage
+ Sign			LEDSIGN

## 6.4 Setting up the Sign Tag

Sign tag will now need parameter information filled out.

Click the **■■■** ICON to begin filling in initial information.

Use the description column for "help".

Several of the parameters have "defaults" that can be used.

Hover mouse over description column to see details.

Name	Value	Force Mask	Style	Data Type	Description
Sign	{...}	{...}		LEDSIGN	Sign Parameters
+ Sign.Address	■■■	**	{...}	STRING	Sign Parameters Options = 00,01,02 'C
+ Sign.CommandCode		**	{...}	STRING	Sign Parameters Options = A,B,C,D pe
+ Sign.FileLabel		**	{...}	STRING	Sign Parameters Default is = A
+ Sign.Text1Message		**	{...}	STRING	Sign Parameters Actual Text Line 1
+ Sign.Text2Message		**	{...}	STRING	Sign Parameters Actual Text Line 2
+ Sign.Text3Message		**	{...}	STRING	Sign Parameters Actual Text Line 3
+ Sign.Text4Message		**	{...}	STRING	Sign Parameters Actual Text Line 4
+ Sign.TextFont		**	{...}	STRING	Sign Parameters Options = 0,1,2,3,4 pe
+ Sign.TextColor		**	{...}	STRING	Sign Parameters Options = 0,1,2,3 per
+ Sign.TextAlign		**	{...}	STRING	Sign Parameters Options = 0,1,2,3,4,5,
+ Sign.TextEffect		**	{...}	STRING	Sign Parameters Options = S,H,F per p
+ Sign.TextSpeed		**	{...}	STRING	Sign Parameters Options = 1 - 8, 3:Def
+ Sign.TextPause		**	{...}	STRING	Sign Parameters Options = 00 - 99, 02
+ Sign.Variable1Name		**	{...}	STRING	Sign Parameters Name of variable Def.
+ Sign.Variable2Name		**	{...}	STRING	Sign Parameters Name of variable Def.
+ Sign.Variable3Name		**	{...}	STRING	Sign Parameters Name of variable Def.
+ Sign.Variable4Name		**	{...}	STRING	Sign Parameters Name of variable Def.
+ Sign.Variable1Value		**	{...}	STRING	Sign Parameters Actual Variable1 to be
+ Sign.Variable2Value		**	{...}	STRING	Sign Parameters Actual Variable2 to be
+ Sign.Variable3Value		**	{...}	STRING	Sign Parameters Actual Variable3 to be

Name	Value	Force Mask	Style	Data Type	Description
Sign	{...}	{...}		LEDSIGN	Sign Parameters
+ Sign.Address	'01'	**	{...}	STRING	Sign Parameters Options = 00,01,02 '00'=all signs
+ Sign.CommandCode	'A'	**	{...}	STRING	Sign Parameters Options = A,B,C,D per protocol 'A'
+ Sign.FileLabel	'A'	**	{...}	STRING	Sign Parameters Default is = A
+ Sign.Text1Message	'Message1'	**	{...}	STRING	Sign Parameters Source: (Type) <LEDSIGN.CommandCode>
+ Sign.Text2Message	'Message2'	**	{...}	STRING	Sign Parameters Options = A,B,C,D per protocol 'A'
+ Sign.Text3Message	'Message3'	**	{...}	STRING	Sign Parameters Options = A,B,C,D per protocol 'A'
+ Sign.Text4Message	'Message4'	**	{...}	STRING	Sign Parameters Options = A,B,C,D per protocol 'A'
+ Sign.TextFont	■■■	**	{...}	STRING	Sign Parameters Options = 0,1,2,3,4,5,6,7,8 per p
+ Sign.TextColor		**	{...}	STRING	Sign Parameters Options = 0,1,2,3 per
+ Sign.TextAlign		**	{...}	STRING	Sign Parameters Options = 0,1,2,3,4,5,6,7,8 per p

Example shown below of Sign Tag with all parameters filled in.

Sign "01" will receive all these parameters when AOI instruction is executed.

Scope: <input type="text" value="Electronic_Displk"/>		Show...		Show All	
Name	Value	Style	Data Type		
- Sign	{...}	{.	LEDSIGN		
+ Sign.Address	'01'	{.	STRING		
+ Sign.CommandCode	'A'	{.	STRING		
+ Sign.FileLabel	'A'	{.	STRING		
+ Sign.Text1Message	'Total : '	{.	STRING		
+ Sign.Text2Message	'Rejects : '	{.	STRING		
+ Sign.Text3Message	'Cycle : '	{.	STRING		
+ Sign.Text4Message	'OEE : '	{.	STRING		
+ Sign.TextFont	'1'	{.	STRING		
+ Sign.TextColor	'1'	{.	STRING		
+ Sign.TextAlign	'0'	{.	STRING		
+ Sign.TextEffect	'd'	{.	STRING		
+ Sign.TextSpeed	'3'	{.	STRING		
+ Sign.TextPause	'02'	{.	STRING		
+ Sign.Variable1Name	'U'	{.	STRING		
+ Sign.Variable2Name	'V'	{.	STRING		
+ Sign.Variable3Name	'W'	{.	STRING		
+ Sign.Variable4Name	'X'	{.	STRING		
+ Sign.Variable1Value	'30990'	{.	STRING		
+ Sign.Variable2Value	'30990'	{.	STRING		
+ Sign.Variable3Value	'30990'	{.	STRING		
+ Sign.Variable4Value	'30990'	{.	STRING		
+ Sign.VariableFont	'1'	{.	STRING		
+ Sign.VariableColor	'0'	{.	STRING		
+ Sign.Packet1Sent	'^B01^AAA^F1^01^U0...	{.	STRING		
+ Sign.Packet2Sent	'^00^NW^F1^01^MOEE...	{.	STRING		
+ Sign.Packet3Sent	'''	{.	STRING		
+ Sign.Packet4Sent	'''	{.	STRING		
+ Sign.Packet5Sent	'''	{.	STRING		
+ Sign.Packet6Sent	'''	{.	STRING		
+ Sign.ExternPacket1	'^B01^AAA^GTH^02^F...	{.	STRING		

Monitor Tags / Edit Tags

## 6.5 Mapping Sign Tag to AOI Function Block in Ladder Logic

Begin mapping all the sign tags to the fields in the AOI function blocks.

Sign tag parameters are word for word matched.

Map all parameters.

The screenshot shows a ladder logic diagram with a function block titled "Send up to Four Lines to LED Sign". The function block is labeled "SendSignMessage" and has a dropdown menu set to "Sign01Tag1" and a text field set to "Sign.Address". Below the diagram is a table of parameters:

Name	Data Type	Description
-Sign	LEDSIGN	Sign Parameters
+ Sign.Address	STRING	Sign Parameters Options = 00,01,02 "00"...
+ Sign.CommandCode	STRING	Sign Parameters Options = A,B,C,D per pr...
+ Sign.FileLabel	STRING	Sign Parameters Default is = A
+ Sign.Text1Message	STRING	Sign Parameters Actual Text Line 1

Below the table are sections for "Controller" and "Program", and a "Show:" list containing: LEDSIGN, STRING, ExpertSign, ResetSign, SendSignMessage, SetupSignVariables, U;

This screenshot shows a detailed view of the "SendSignMessage" function block. The parameters are mapped as follows:

- Send up to Four Lines to LED Sign: SendSignMessage
- SendSignMessage: Sign01Tag1
- Address: Sign.Address
- CommandCode: Sign.CommandCode
- FileLabel: Sign.FileLabel
- ENText1Message: 0
- Text1Message: Sign.Text1Message
- ENText2Message: 0
- Text2Message: Sign.Text2Message
- ENText3Message: 0
- Text3Message: Sign.Text3Message
- ENText4Message: 0
- Text4Message: Sign.Text4Message
- ENTextFormats: 0
- TextFont: ?
- TextColor: ?
- TextAlign: ?
- ENTextSpecials: 0

On the right side, there are four output terminals labeled (EN), (DN), (IP), and (ER).



## 7 ELECTRONIC DISPLAY AOI INSTRUCTIONS

### 7.1 Send Sign Message AOI

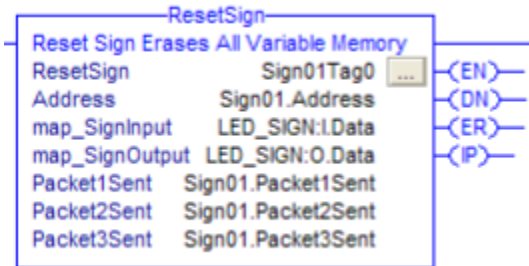
Instruction used to send up to four messages to a sign. (see video tutorials)



Operand	Type	Description
SendSignMessage	Tag	Unique Tag
Address	String	Two character sign address "00"
CommandCode	String	Protocol Command "A,B,C,D"
Filelabel	String	Protocol Command "A" typically
ENText1Message	Bool	Flag to send message 1=send 0=no
Text1Message	String	Actual message in string to send
ENText2Message	Bool	Flag to send message 1=send 0=no
Text2Message	String	Actual message in string to send
ENText3Message	Bool	Flag to send message 1=send 0=no
Text3Message	String	Actual message in string to send
ENText4Message	Bool	Flag to send message 1=send 0=no
Text3Message	String	Actual message in string to send
ENTextFormats	Bool	Flag to format message 1=yes 0=no
TextFont	String	Protocol Command Font Size
TextColor	String	Protocol Command Color
TextAlign	String	Protocol Command Text Align
TextAttribute	String	Protocol Command Text Styling
ENTextSpecials	Bool	Flag to format effects 1=yes 0=no
TextEffect	String	Protocol Command Text Special
TextSpeed	String	Protocol Command Text Speed
TextPause	String	Protocol Command Text Pause
ENVariable1Message	Bool	Flag to send variable 1=send 0=no
Variable1Name	String	Protocol Command Variable Name
ENVariable2Message	Bool	Flag to send variable 1=send 0=no
Variable2Name	String	Protocol Command Variable Name
ENVariable3Message	Bool	Flag to send variable 1=send 0=no
Variable3Name	String	Protocol Command Variable Name
ENVariable4Message	Bool	Flag to send variable 1=send 0=no
Variable4Name	String	Protocol Command Variable Name
ENVariableFormats	Bool	Flag to format variable 1=yes 0=no
VariableFont	String	Protocol Command Font Size
VariableColor	String	Protocol Command Color
VariableAttribute	String	Protocol Command Text Styling
map_SignInput	I:Data	Ethernet/IP Input Data Mapping
map_SignOutput	O:Data	Ethernet/IP Output Data Mapping
Packet1Sent	String	82 Length String Debug of Packet
Packet2Sent	String	82 Length String Debug of Packet
Packet3Sent	String	82 Length String Debug of Packet
Packet4Sent	String	82 Length String Debug of Packet
Packet5Sent	String	82 Length String Debug of Packet
Packet6Sent	String	82 Length String Debug of Packet
EN	Bool	Instruction is enabled
DN	Bool	Instruction is done sending message
IP	Bool	Instruction is in progress sending
ER	Bool	Instruction failed to send message

## 7.2 Reset Sign AOI

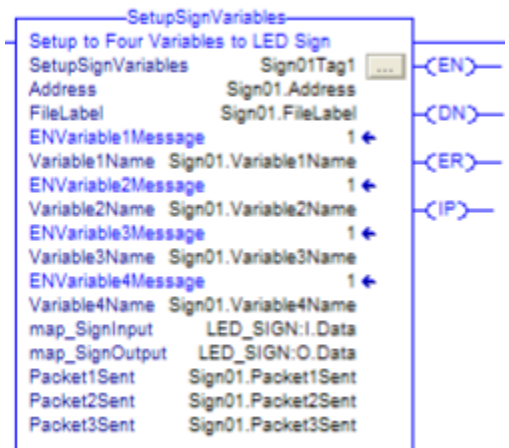
Instruction used to clean all variable data previously stored in sign. (see video tutorials)



Operand	Type	Description
ResetSign	Tag	Unique Tag
Address	String	Two character sign address "00"
map_SignInput	I:Data	Ethernet/IP Input Data Mapping
map_SignOutput	O:Data	Ethernet/IP Output Data Mapping
Packet1Sent	String	82 Length String Debug of Packet
Packet2Sent	String	82 Length String Debug of Packet
Packet3Sent	String	82 Length String Debug of Packet
EN	Bool	Instruction is enabled
DN	Bool	Instruction is done sending message
IP	Bool	Instruction is in progress sending
ER	Bool	Instruction failed to send message

## 7.3 Setup Sign Variables AOI

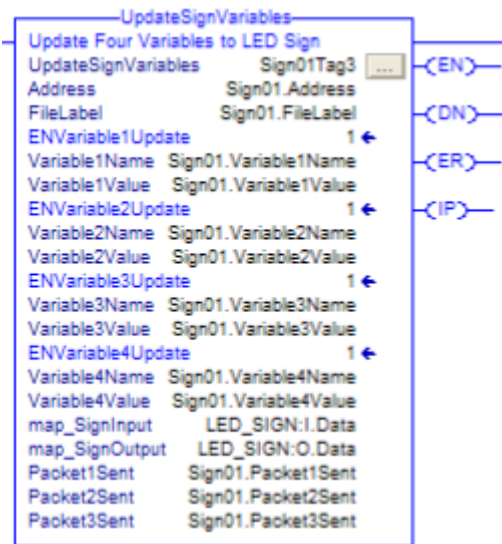
Instruction to write the memory area in the sign for variables up to 4 memory spots. (see video tutorials)



Operand	Type	Description
SendSignMessage	Tag	Unique Tag
Address	String	Two character sign address "00"
Filelabel	String	Protocol Command "A" typically
ENVariable1Message	Bool	Flag to send variable 1=send 0=no
Variable1Name	String	Protocol Command Variable Name
ENVariable2Message	Bool	Flag to send variable 1=send 0=no
Variable2Name	String	Protocol Command Variable Name
ENVariable3Message	Bool	Flag to send variable 1=send 0=no
Variable3Name	String	Protocol Command Variable Name
ENVariable4Message	Bool	Flag to send variable 1=send 0=no
Variable4Name	String	Protocol Command Variable Name
map_SignInput	I:Data	Ethernet/IP Input Data Mapping
map_SignOutput	O:Data	Ethernet/IP Output Data Mapping
Packet1Sent	String	82 Length String Debug of Packet
Packet2Sent	String	82 Length String Debug of Packet
Packet3Sent	String	82 Length String Debug of Packet
EN	Bool	Instruction is enabled
DN	Bool	Instruction is done sending message
IP	Bool	Instruction is in progress sending
ER	Bool	Instruction failed to send message

## 7.4 Update Sign Variables AOI

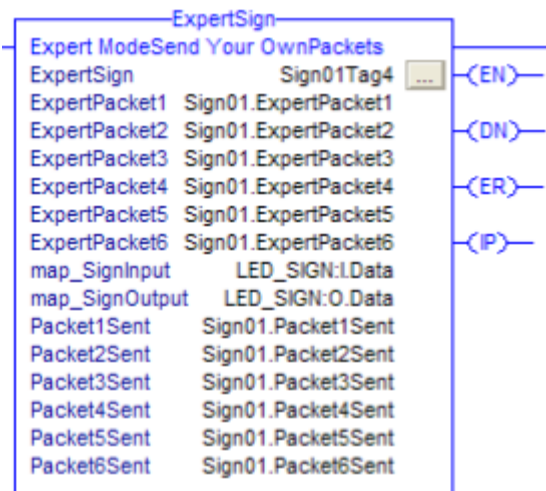
Instruction to update variable memory space in the sign (see video tutorials)



Operand	Type	Description
UpdateSignVariables	Tag	Unique Tag
Address	String	Two character sign address "00"
Filelabel	String	Protocol Command "A" typically
ENVariable1Update	Bool	Flag to update variable 1=send 0 =no
Variable1Name	String	Protocol Command Variable Name
Variable1Value	String	Actual variable data to send
ENVariable1Update	Bool	Flag to update variable 1=send 0 =no
Variable2Name	String	Protocol Command Variable Name
Variable2Value	String	Actual variable data to send
ENVariable1Update	Bool	Flag to update variable 1=send 0 =no
Variable3Name	String	Protocol Command Variable Name
Variable3Value	String	Actual variable data to send
ENVariable1Update	Bool	Flag to update variable 1=send 0 =no
Variable4Name	String	Protocol Command Variable Name
Variable4Value	String	Actual variable data to send
map_SignInput	I:Data	Ethernet/IP Input Data Mapping
map_SignOutput	O:Data	Ethernet/IP Output Data Mapping
Packet1Sent	String	82 Length String Debug of Packet
Packet2Sent	String	82 Length String Debug of Packet
Packet3Sent	String	82 Length String Debug of Packet
EN	Bool	Instruction is enabled
DN	Bool	Instruction is done sending message
IP	Bool	Instruction is in progress sending
ER	Bool	Instruction failed to send message

## 7.5 Expert Sign AOI

Instruction to make your own packets and send to the sign. (see video tutorials)



Operand	Type	Description
ExpertSign	Tag	Unique Tag
ExpertPacket1	String	82 Length String of Your Commands
ExpertPacket2	String	82 Length String of Your Commands
ExpertPacket3	String	82 Length String of Your Commands
ExpertPacket4	String	82 Length String of Your Commands
ExpertPacket5	String	82 Length String of Your Commands
ExpertPacket6	String	82 Length String of Your Commands
map_SignInput	I:Data	Ethernet/IP Input Data Mapping
map_SignOutput	O:Data	Ethernet/IP Output Data Mapping
Packet1Sent	String	82 Length String Debug of Packet
Packet2Sent	String	82 Length String Debug of Packet
Packet3Sent	String	82 Length String Debug of Packet
Packet4Sent	String	82 Length String Debug of Packet
Packet5Sent	String	82 Length String Debug of Packet
Packet6Sent	String	82 Length String Debug of Packet
EN	Bool	Instruction is enabled
DN	Bool	Instruction is done sending message
IP	Bool	Instruction is in progress sending
ER	Bool	Instruction failed to send message