1-800-367-6056

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

TABLE OF CONTENTS

1 INTRODUCTION 4

1.1 Supported PLC Controllers 4

1.2 Add-On Instructions 4

1.3 Supported LED Signs 5

2 SYSTEM BLOCK DIAGRAM 6

2.1 Typical Connection Diagram 6

2.2 Internal EDV111 Connection Diagram 7

2.3 Single Sign Connection 8

2.4 Multiple Sign Connection (Less than 50 Feet apart) 8

2.5 Multiple Sign Connection (More than 50 Feet apart) 9

3 CUSTOMIZE THE IP ADDRESS HMS ANYBUS COMMUNICATOR 10

3.1 Install Anybus Configuration Manager Software 10

3.2 Connect to Anybus RS-232 Programming Port 11

3.3 Anybus Configuration Manager 12

3.4 Downloading Configuration to Anybus Communicator 15

4 QUICK START USING TEMPLATE PROGRAM 18

4.1 Quick Start with Template Program 18

4.2 Update Sign IP Address in Sample Program 19

4.3 Update Messages and Sign Format Tags 20

4.4 Create and Customize Ladder Logic 21

5 IMPORTING WITH NEW PROGRAM OR EXISITING PROGRAM 22

5.1 Start a new project with RS Logix 5000 22

5.2 Create a New Ethernet Connection 23

5.3 Importing Data-Types 27

5.4 Importing Add-on Instructions 29

6 USING THE AOI INSTRUCTIONS IN THE PROJECT 33

6.1 Add AOIs to Ladder Programming via Drag and Drop 33

6.2 Creating Unique Tags for AOI.. 34

6.3 Creating Unique Tags for Sign Parameters... 35

6.4 Setting up the Sign Tag 37

6.5 Mapping Sign Tag to AOI Function Block in Ladder Logic 39

7 ELECTRONIC DISPLAY AOI INSTRUCTIONS.. 40

7.1 Send Sign Message AOI.. 40

7.2 Reset Sign AOI 41

7.3 Setup Sign Variables AOI 41

7.4 Update Sign Variables AOI 42

7.5 Expert Sign AOI 42

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

EDV111-3280-IND

EDV111-16160-IND

EDV111-16128-IND

EDV111-16240-IND

EDV111-24160-IND

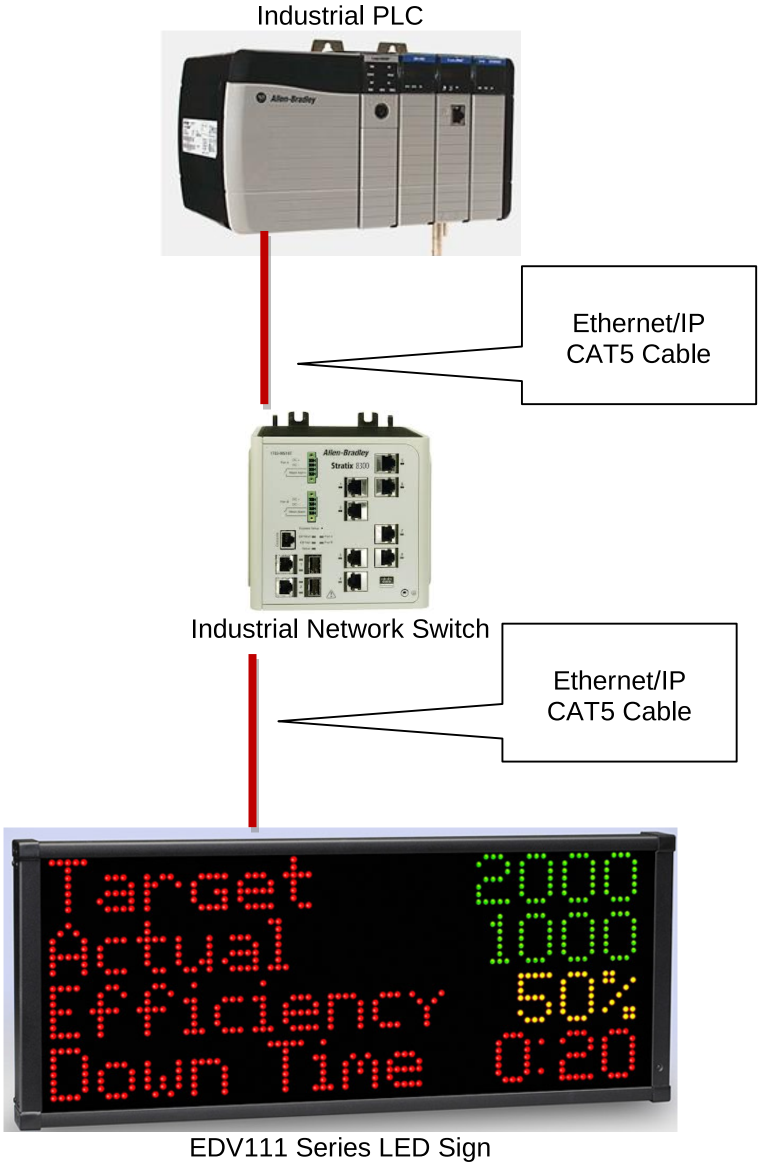
EDV111-32240-IND

EDV111-1680-IND

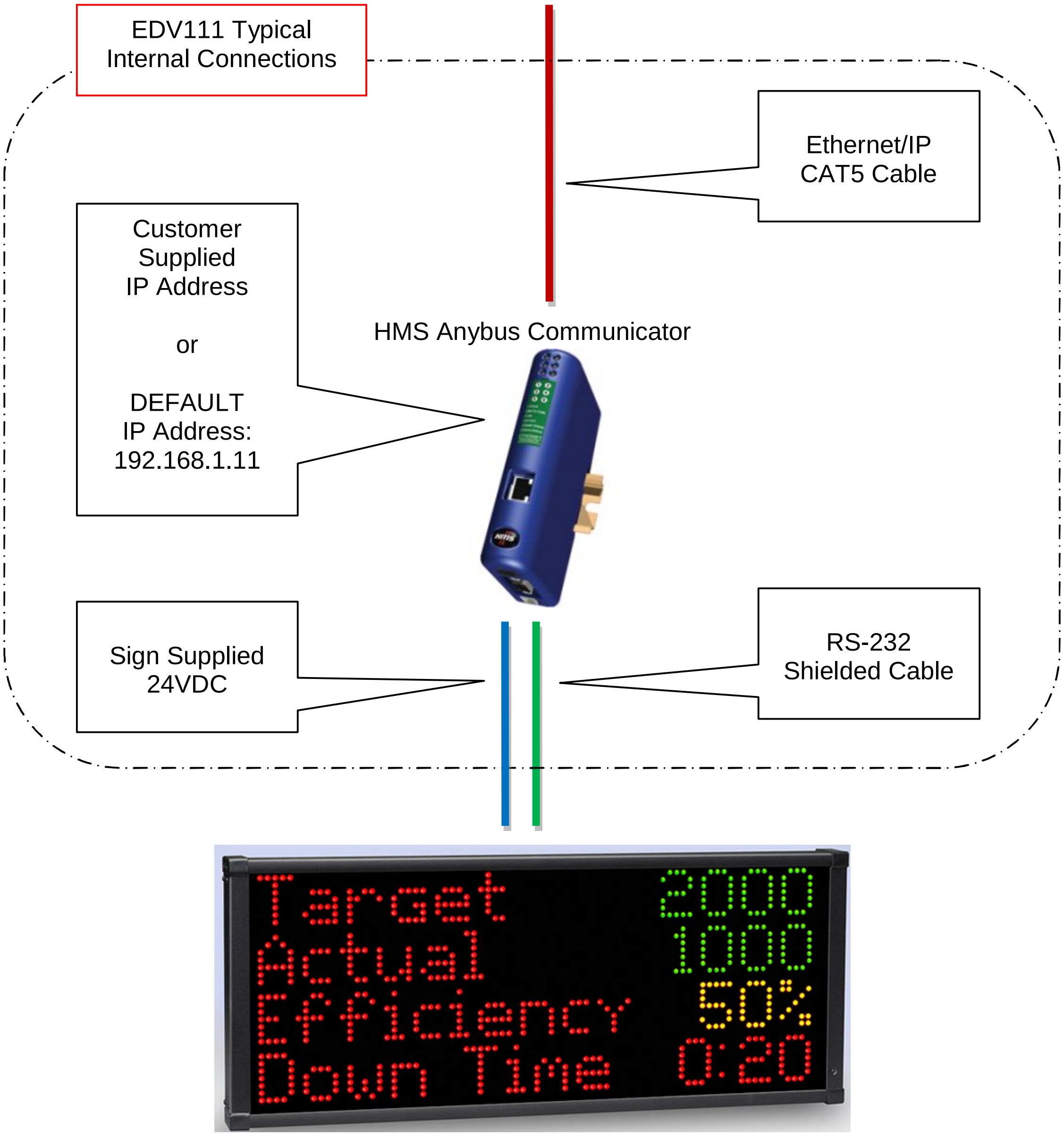
2 SYSTEM BLOCK DIAGRAM

EDV111 LED signs come equipped with an Anybus Communicator 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



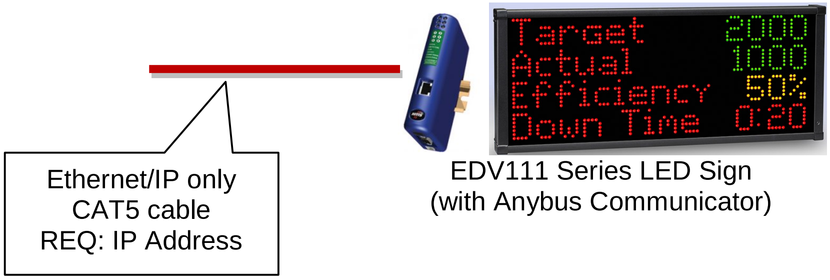
2.2 Internal EDV111 Connection Diagram



EDV111 Series LED Sign

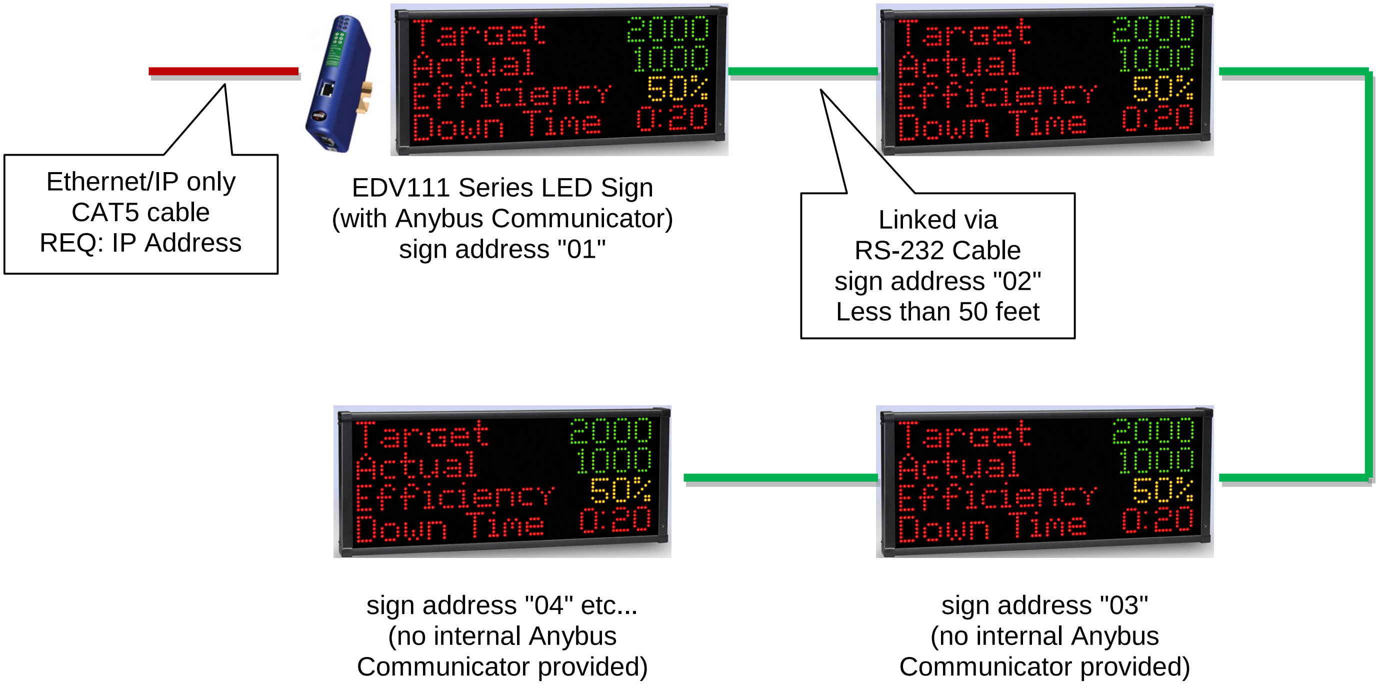
2.3 Single Sign Connection

Single sign connection must have an installed "Master" HMS Anybus Communicator device.



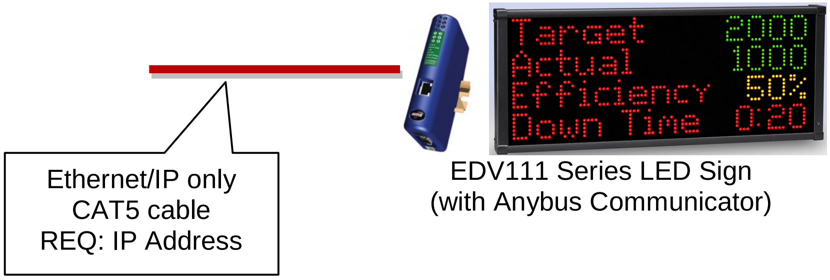
2.4 Multiple Sign Connection (Less than 50 Feet apart)

At least one EDV111 sign must have an installed "Master" HMS Anybus Communicator device. This master can reproduce RS-232 signals in a repeater fashion to slave signs by linking multiple EDV111 series signs together using a standard RS-232 cable. Signs must not be mounted more than 50 feet apart. Downstream "slave" signs will not require internal HMS Anybus Communicators.

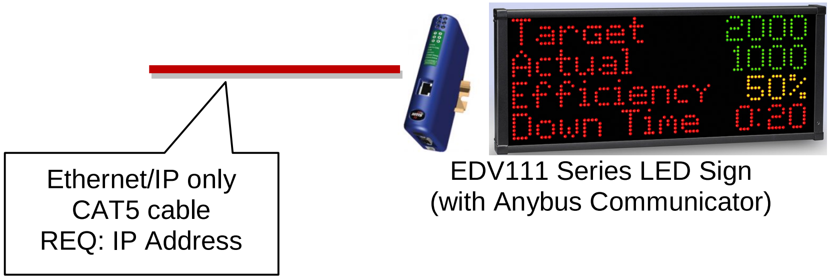


2.5 Multiple Sign Connection (More than 50 Feet apart)

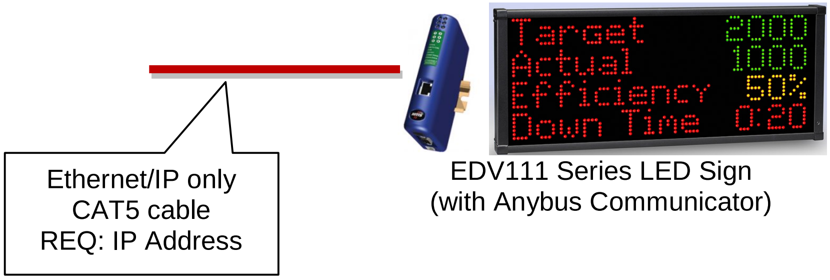
Each sign will require an EDV111 to be internally installed as a "Master" HMS Anybus Communicator device.



EDV111 Series LED Sign
(with Anybus Communicator)



EDV111 Series LED Sign
(with Anybus Communicator)



EDV111 Series LED Sign
(with Anybus Communicator)

3 CUSTOMIZE THE IP ADDRESS HMS ANYBUS COMMUNICATOR

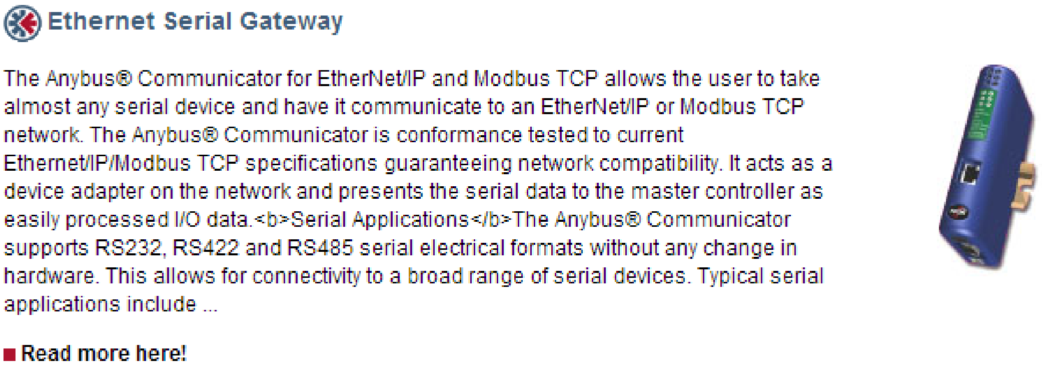
Download the sample configuration file from the Electric Displays website.

Electronic_Displays_datamode-v1.1.cfg
(or latest version)

3.1 Install Anybus Configuration Manager Software

Download and install the Anybus Configuration Manager software from HMS website.

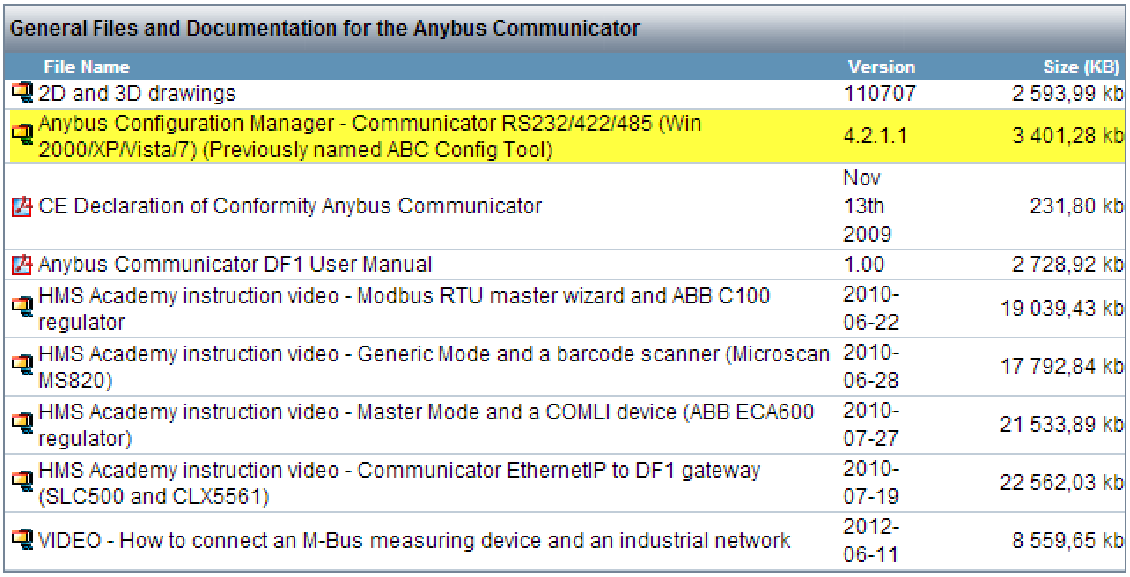
http://www.hms.se/support/support.asp?PID=110&ProductType=Anybus%20Communicator



Ethernet Serial Gateway

The Anybus® Communicator for EtherNet/IP and Modbus TCP allows the user to take almost any serial device and have it communicate to an EtherNet/IP or Modbus TCP network. The Anybus® Communicator is conformance tested to current Ethernet/IP/Modbus TCP specifications guaranteeing network compatibility. It acts as a device adapter on the network and presents the serial data to the master controller as easily processed I/O data.<b>Serial Applications</b>The Anybus® Communicator supports RS232, RS422 and RS485 serial electrical formats without any change in hardware. This allows for connectivity to a broad range of serial devices. Typical serial applications include ...

■ Read more here!



General Files and Documentation for the Anybus Communicator

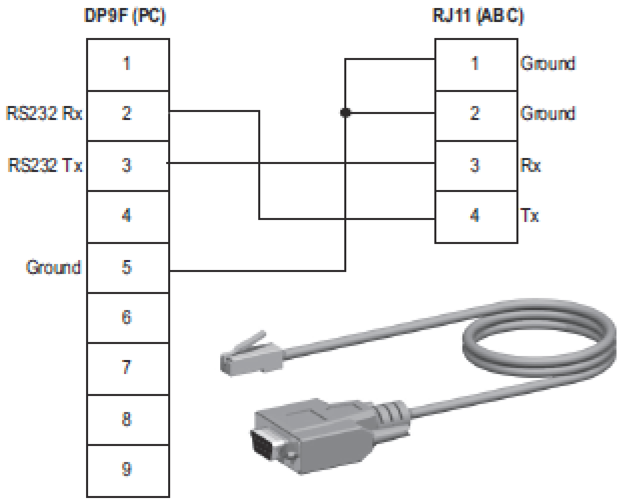
File Name	Version	Size (KB)
2D and 3D drawings	110707	2 593,99 kb
Anybus Configuration Manager - Communicator RS232/422/485 (Win 2000/XP/Vista/7) (Previously named ABC Config Tool)	4.2.1.1	3 401,28 kb
CE Declaration of Conformity Anybus Communicator	Nov 13th 2009	231,80 kb
Anybus Communicator DF1 User Manual	1.00	2 728,92 kb
HMS Academy instruction video - Modbus RTU master wizard and ABB C100 regulator	2010-06-22	19 039,43 kb
HMS Academy instruction video - Generic Mode and a barcode scanner (Microscan MS820)	2010-06-28	17 792,84 kb
HMS Academy instruction video - Master Mode and a COMLI device (ABB ECA600 regulator)	2010-07-27	21 533,89 kb
HMS Academy instruction video - Communicator EthernetIP to DF1 gateway (SLC500 and CLX5561)	2010-07-19	22 562,03 kb
VIDEO - How to connect an M-Bus measuring device and an industrial network	2012-06-11	8 559,65 kb

3.2 Connect to Anybus RS-232 Programming Port

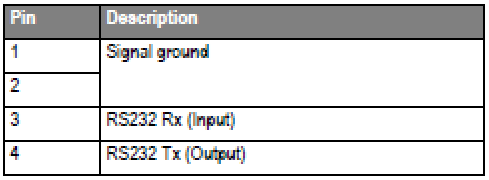
Connect to the Anybus RS-232 PC Port using supplied programming cable.

Connect DB-9 to serial port on computer with installed Anybus Configuration Manager software.

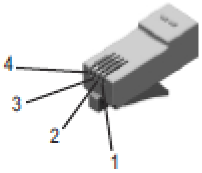
Configuration Cable Wiring



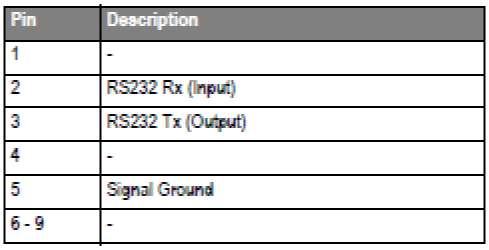
RJ11 (4P4C modular)[1] : ABC



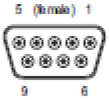
Pin	Description
1	Signal ground
2	
3	RS232 Rx (Input)
4	RS232 Tx (Output)



DB9F : PC



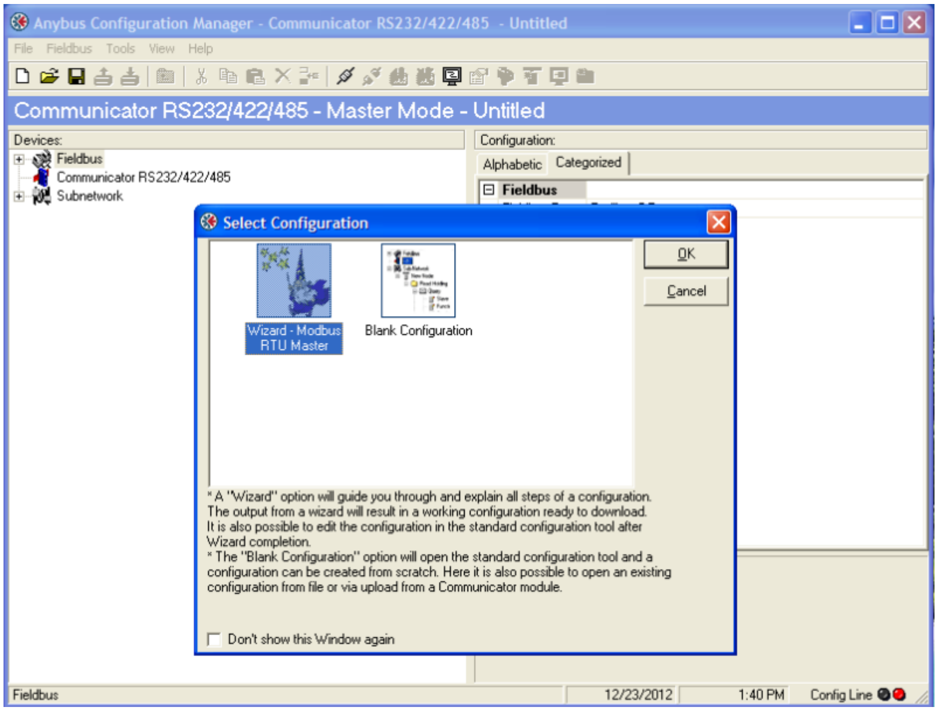
Pin	Description
1	-
2	RS232 Rx (Input)
3	RS232 Tx (Output)
4	-
5	Signal Ground
6 - 9	-



3.3 Anybus Configuration Manager

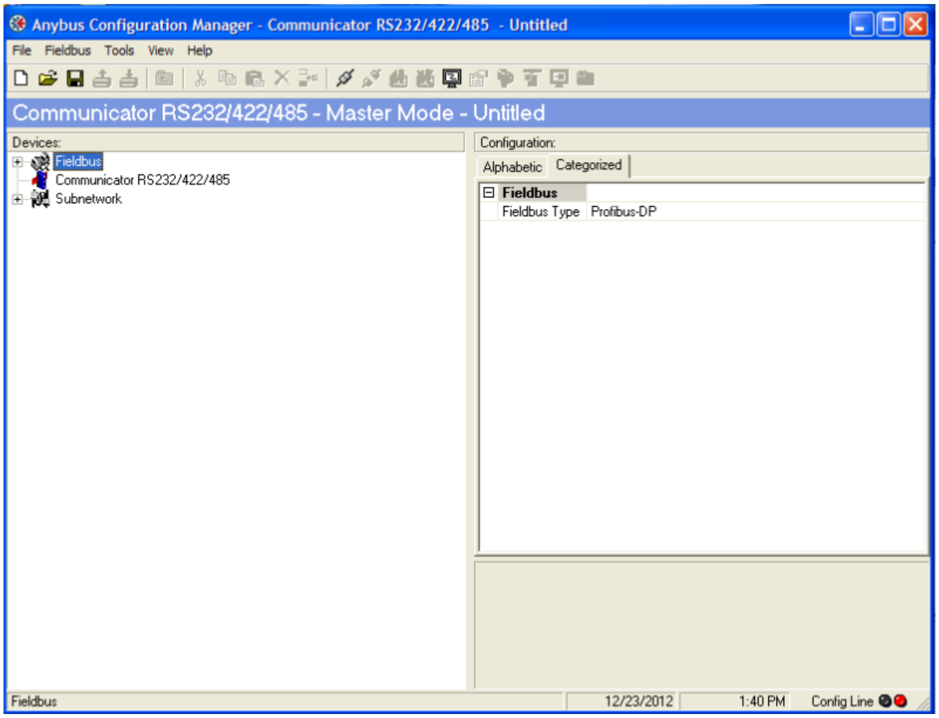
Launch Anybus Configuration Manager

Choose "Blank Configuration"


Anybus Configuration Manager - Communicator RS232/422/485 - Untitled
File Fieldbus Tools View Help
Communicator RS232/422/485 - Master Mode - Untitled
Devices:
Fieldbus
Communicator RS232/422/485
Subnetwork
Configuration:
Alphabetic Categorized
Fieldbus
Select Configuration
Wizard - Modbus RTU Master
Blank Configuration
OK
Cancel
Fieldbus
12/23/2012
1:40 PM
Config Line

* A "Wizard" option will guide you through and explain all steps of a configuration. The output from a wizard will result in a working configuration ready to download. It is also possible to edit the configuration in the standard configuration tool after Wizard completion.
* The "Blank Configuration" option will open the standard configuration tool and a configuration can be created from scratch. Here it is also possible to open an existing configuration from file or via upload from a Communicator module.

□ Don't show this Window again

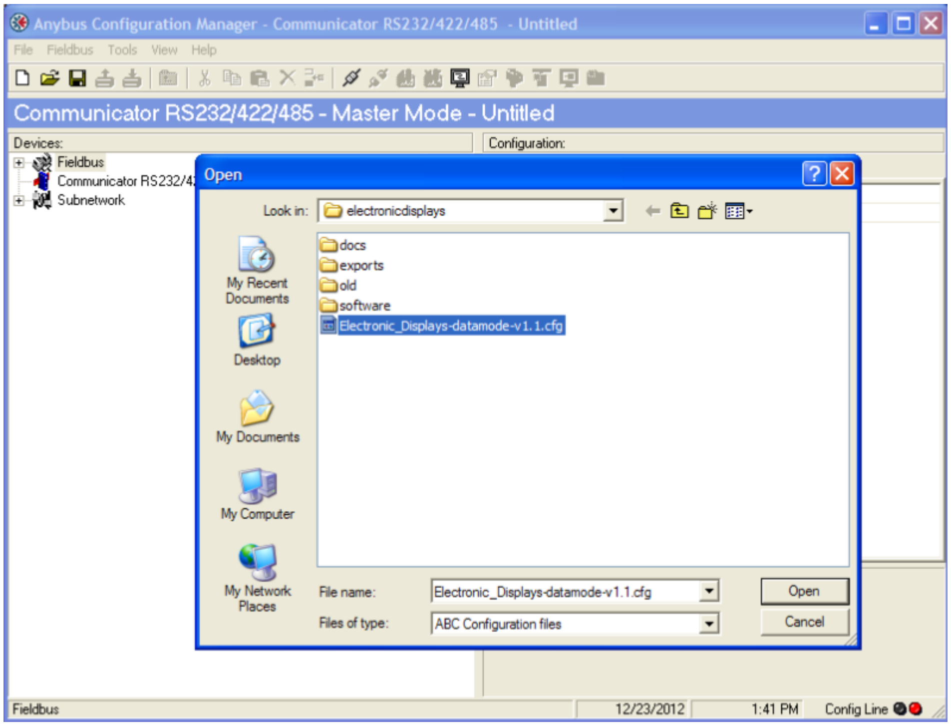

Anybus Configuration Manager - Communicator RS232/422/485 - Untitled
File Fieldbus Tools View Help
Communicator RS232/422/485 - Master Mode - Untitled
Devices:
Fieldbus
Communicator RS232/422/485
Subnetwork
Configuration:
Alphabetic Categorized
Fieldbus
Fieldbus
12/23/2012
1:40 PM
Config Line

Fieldbus Type: Profibus-DP

Click File, Open.

Choose the .CFG file downloaded from Electronic Displays website.

Click Open


Anybus Configuration Manager - Communicator RS232/422/485 - Untitled

Communicator RS232/422/485 - Master Mode - Untitled

Open

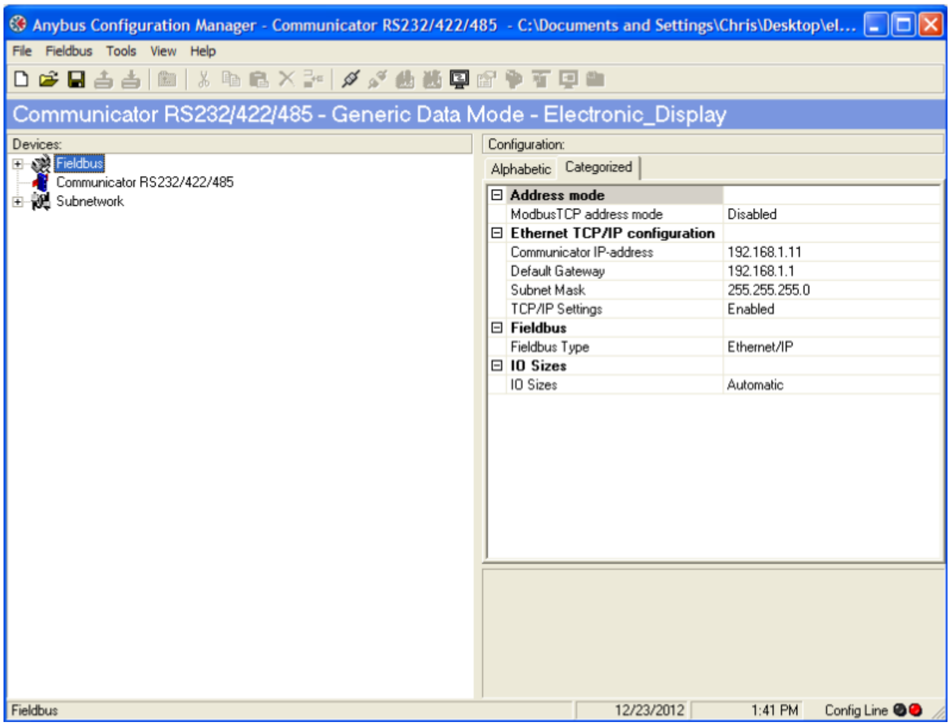
Look in: electronicdisplays

docs
exports
old
software
Electronic_Displays-datamode-v1.1.cfg

File name: Electronic_Displays-datamode-v1.1.cfg

Files of type: ABC Configuration files

Fieldbus | 12/23/2012 | 1:41 PM | Config Line


Anybus Configuration Manager - Communicator RS232/422/485 - C:\Documents and Settings\Chris\Desktop\el...

Communicator RS232/422/485 - Generic Data Mode - Electronic_Display

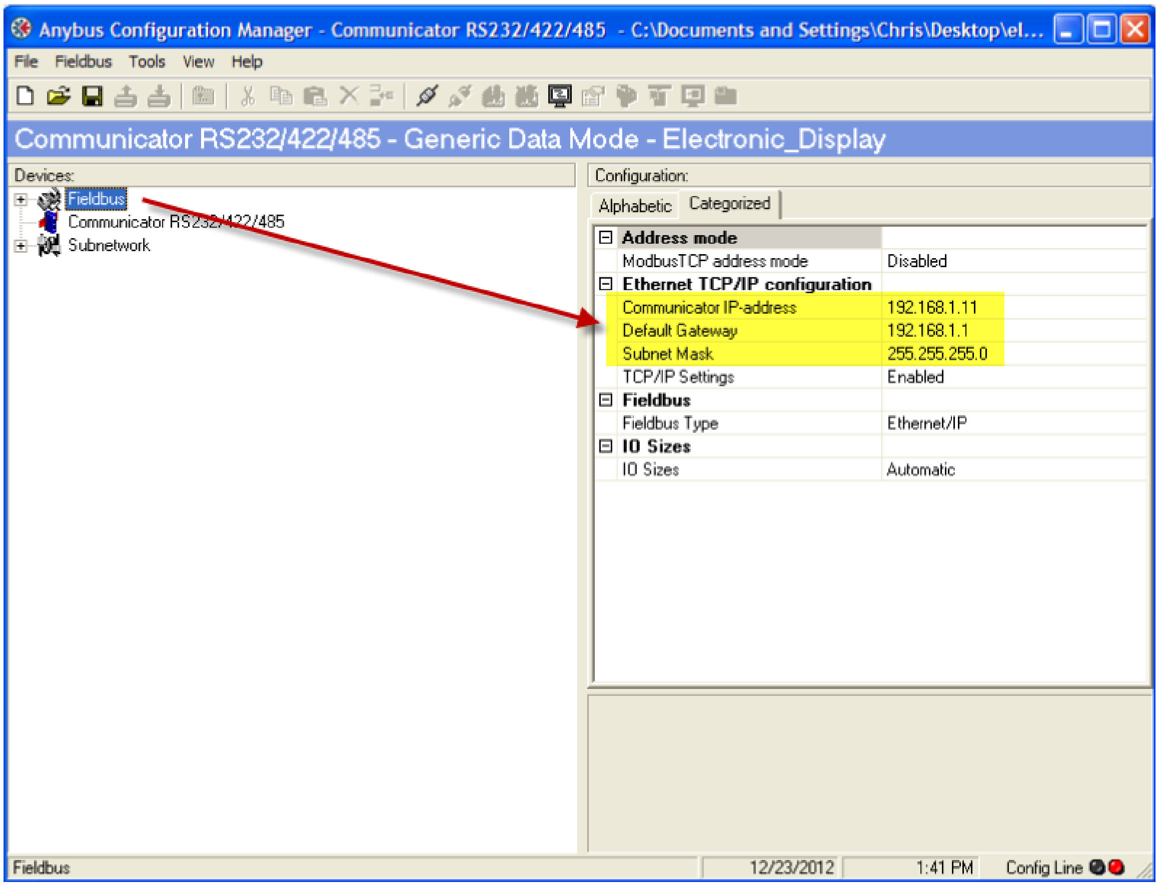
Address mode	
ModbusTCP address mode	Disabled
Ethernet TCP/IP configuration	
Communicator IP-address	192.168.1.11
Default Gateway	192.168.1.1
Subnet Mask	255.255.255.0
TCP/IP Settings	Enabled
Fieldbus	
Fieldbus Type	Ethernet/IP
IO Sizes	
IO Sizes	Automatic

Fieldbus | 12/23/2012 | 1:41 PM | Config Line

Customize IP Address

Customize Default Gateway

Customize Subnet Mask


Anybus Configuration Manager - Communicator RS232/422/485 - C:\Documents and Settings\Chris\Desktop\el...

Communicator RS232/422/485 - Generic Data Mode - Electronic_Display

Devices:

- Fieldbus
- Communicator RS232/422/485
- Subnetwork

Configuration:

⊟ **Address mode**	
ModbusTCP address mode	Disabled
⊟ **Ethernet TCP/IP configuration**	
Communicator IP-address	192.168.1.11
Default Gateway	192.168.1.1
Subnet Mask	255.255.255.0
TCP/IP Settings	Enabled
⊟ **Fieldbus**	
Fieldbus Type	Ethernet/IP
⊟ **IO Sizes**	
IO Sizes	Automatic

Fieldbus 12/23/2012 1:41 PM Config Line

Do not change ANY other settings in this configuration file or the Anybus Communicator module will not communicate to the PLC or the LED Sign.

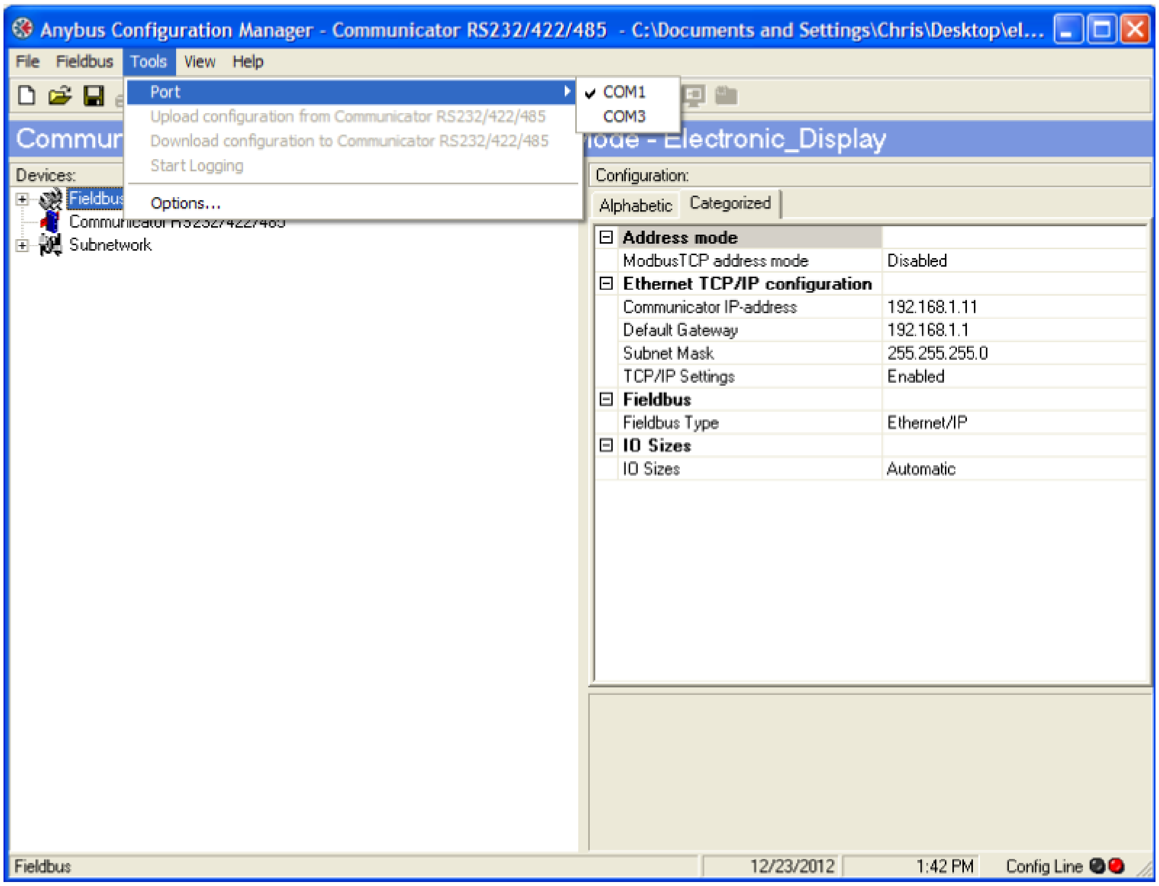
Save your file as your own customized file configuration file with your customized IP address for your Anybus Communicator device.

3.4 Downloading Configuration to Anybus Communicator

Under Tools, choose your COM port on your PC.

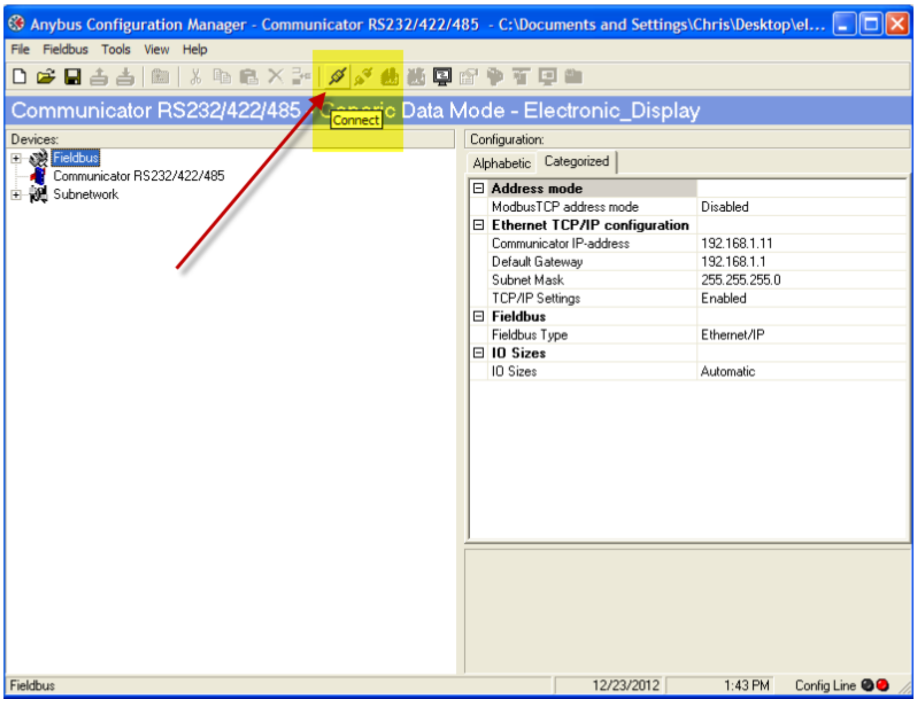
Connect PC Programming Cable to PC

Connect RJ-11 Cable to Programming Port on Anybus Communicator


Anybus Configuration Manager - Communicator RS232/422/485 - C:\Documents and Settings\Chris\Desktop\el...
File
Fieldbus
Tools
View
Help
Port
✓ COM1
COM3
Upload configuration from Communicator RS232/422/485
Download configuration to Communicator RS232/422/485
Start Logging
Options...
Commun
node - Electronic_Display
Devices:
Fieldbus
Communicator RS232/422/485
Subnetwork
Configuration:
Alphabetic
Categorized
Fieldbus
12/23/2012
1:42 PM
Config Line

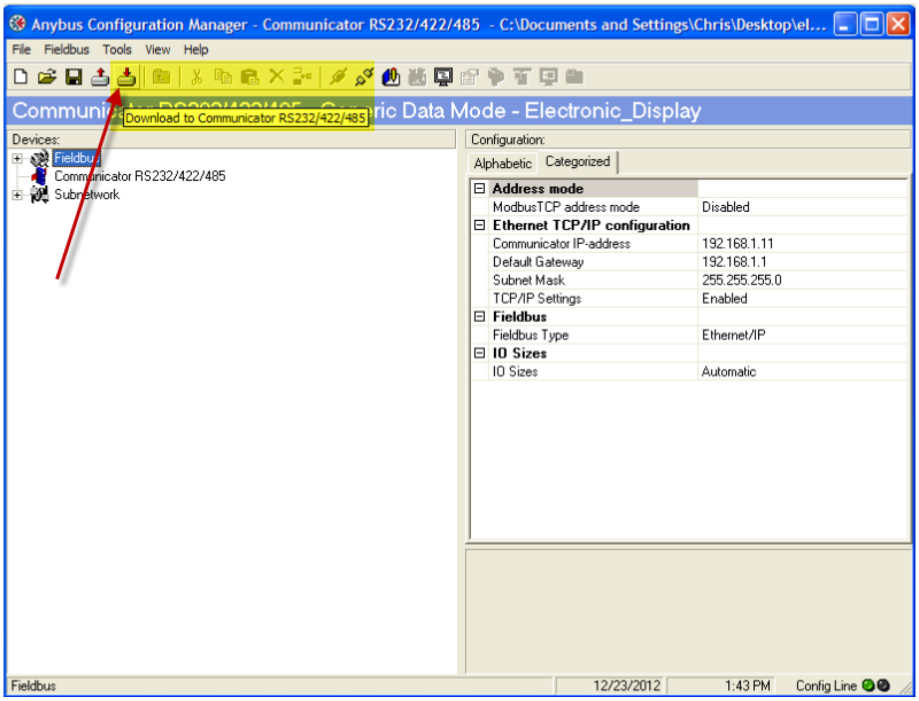
Address mode	
ModbusTCP address mode	Disabled
Ethernet TCP/IP configuration	
Communicator IP-address	192.168.1.11
Default Gateway	192.168.1.1
Subnet Mask	255.255.255.0
TCP/IP Settings	Enabled
Fieldbus	
Fieldbus Type	Ethernet/IP
IO Sizes	
IO Sizes	Automatic

Click on Connect ICON


Anybus Configuration Manager - Communicator RS232/422/485 - C:\Documents and Settings\Chris\Desktop\el...
File
Fieldbus
Tools
View
Help
Connect
Communicator RS232/422/485 - Generic Data Mode - Electronic_Display
Devices:
Fieldbus
Communicator RS232/422/485
Subnetwork
Configuration:
Alphabetic
Categorized
Fieldbus
12/23/2012
1:43 PM
Config Line

Address mode	
ModbusTCP address mode	Disabled
Ethernet TCP/IP configuration	
Communicator IP-address	192.168.1.11
Default Gateway	192.168.1.1
Subnet Mask	255.255.255.0
TCP/IP Settings	Enabled
Fieldbus	
Fieldbus Type	Ethernet/IP
IO Sizes	
IO Sizes	Automatic

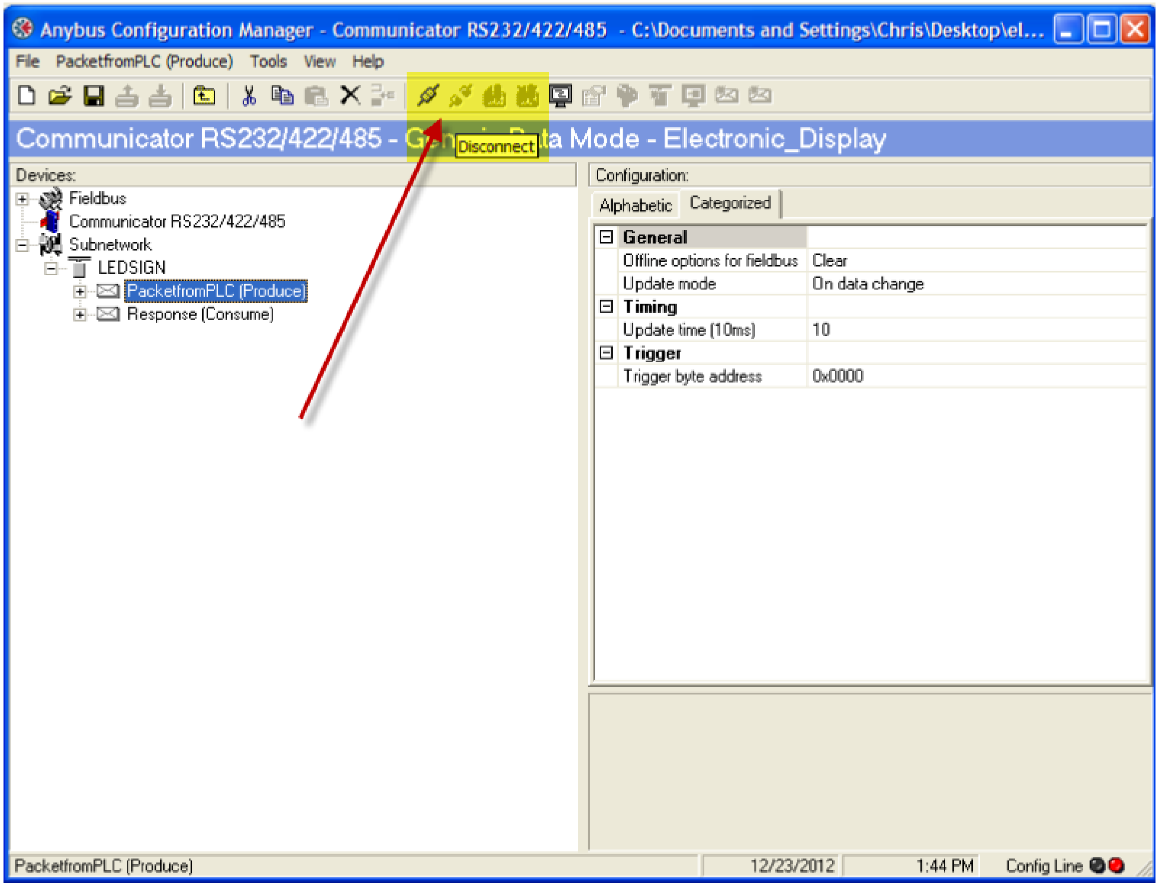
Click on the Download ICON


Anybus Configuration Manager - Communicator RS232/422/485 - C:\Documents and Settings\Chris\Desktop\el...
File
Fieldbus
Tools
View
Help
Download to Communicator RS232/422/485
Communicator RS232/422/485 - Generic Data Mode - Electronic_Display
Devices:
Fieldbus
Communicator RS232/422/485
Subnetwork
Configuration:
Alphabetic
Categorized
Fieldbus
12/23/2012
1:43 PM
Config Line

Address mode	
ModbusTCP address mode	Disabled
Ethernet TCP/IP configuration	
Communicator IP-address	192.168.1.11
Default Gateway	192.168.1.1
Subnet Mask	255.255.255.0
TCP/IP Settings	Enabled
Fieldbus	
Fieldbus Type	Ethernet/IP
IO Sizes	
IO Sizes	Automatic

Click on the Disconnect ICON

Reboot the Anybus Communicator / LED Sign



General	
Offline options for fieldbus	Clear
Update mode	On data change
Timing	
Update time (10ms)	10
Trigger	
Trigger byte address	0x0000

4 QUICK START USING TEMPLATE PROGRAM

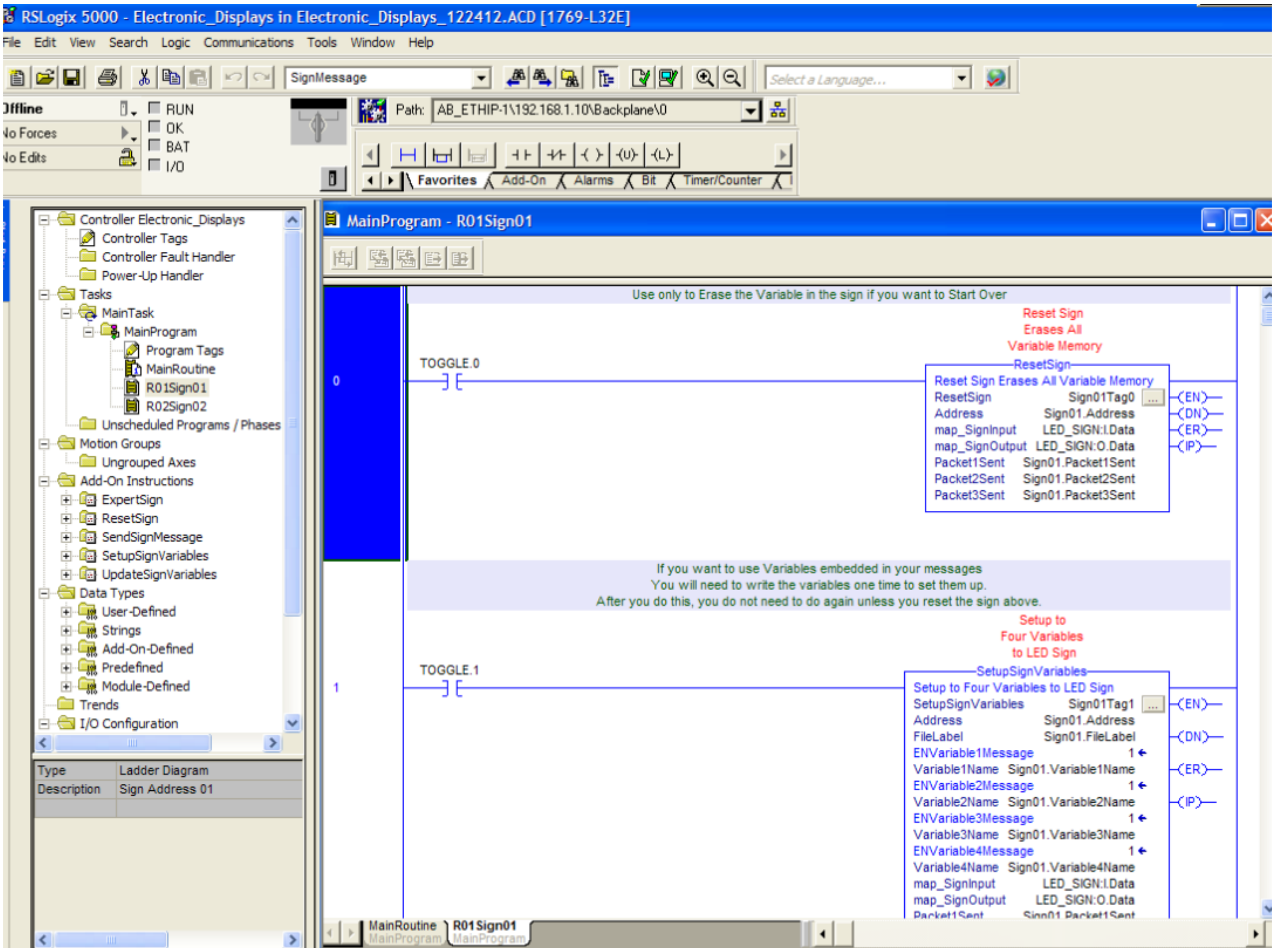
Download the sample PLC program from the Electric Displays website.

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.



Path: AB_ETHIP-1\192.168.1.10\Backplane\0

Type	Ladder Diagram
Description	Sign Address 01

Use only to Erase the Variable in the sign if you want to Start Over

Rung 0: TOGGLE.0

Reset Sign
Erases All
Variable Memory

```
ResetSign
Reset Sign Erases All Variable Memory
ResetSign            Sign01Tag0     (EN)
Address              Sign01.Address (DN)
map_SignInput      LED_SIGN:I.Data  (ER)
map_SignOutput     LED_SIGN:O.Data  (IP)
Packet1Sent    Sign01.Packet1Sent
Packet2Sent    Sign01.Packet2Sent
Packet3Sent    Sign01.Packet3Sent
```

If you want to use Variables embedded in your messages
You will need to write the variables one time to set them up.
After you do this, you do not need to do again unless you reset the sign above.

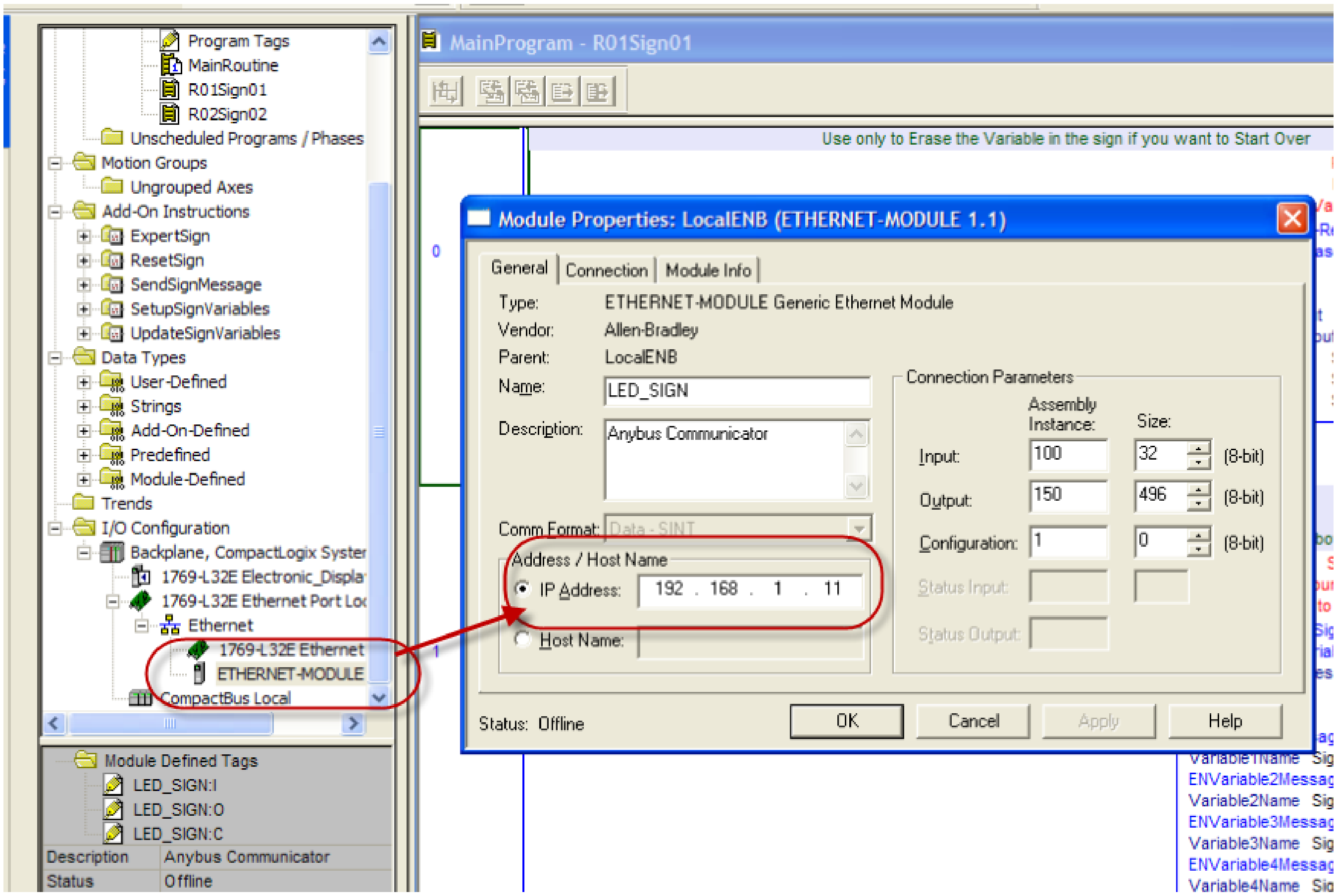
Rung 1: TOGGLE.1

Setup to
Four Variables
to LED Sign

```
SetupSignVariables
Setup to Four Variables to LED Sign
SetupSignVariables         Sign01Tag1   (EN)
Address                Sign01.Address
FileLabel            Sign01.FileLabel   (DN)
ENVariable1Message                  1
Variable1Name  Sign01.Variable1Name     (ER)
ENVariable2Message                  1
Variable2Name  Sign01.Variable2Name     (IP)
ENVariable3Message                  1
Variable3Name  Sign01.Variable3Name
ENVariable4Message                  1
Variable4Name  Sign01.Variable4Name
map_SignInput         LED_SIGN:I.Data
map_SignOutput        LED_SIGN:O.Data
Packet1Sent        Sign01.Packet1Sent
```

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.



Description: Anybus Communicator
Status: Offline

Module Properties: LocalENB (ETHERNET-MODULE 1.1)

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: LocalENB
Name: LED_SIGN
Description: Anybus Communicator
Comm Format: Data - SINT

Address / Host Name
● IP Address: 192 . 168 . 1 . 11
○ Host Name:

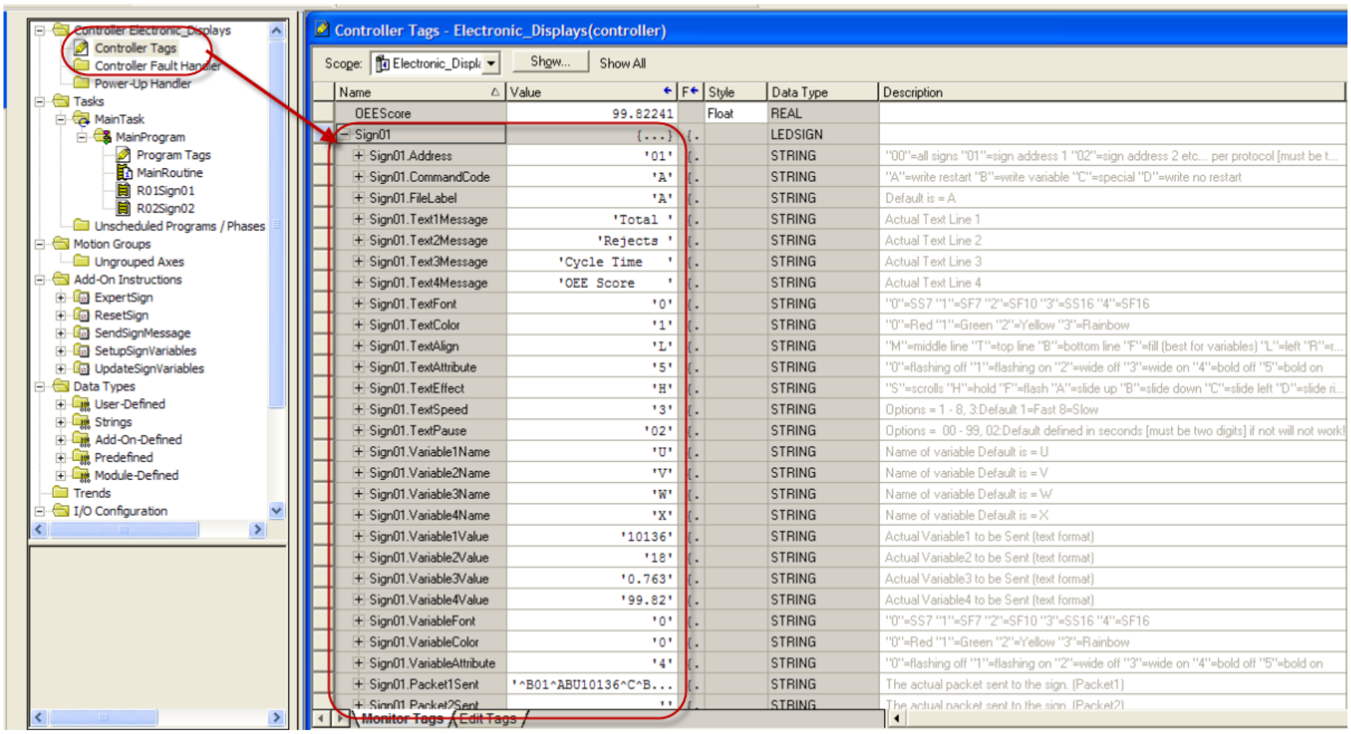
Connection Parameters

	Assembly Instance:	Size:	
Input:	100	32	(8-bit)
Output:	150	496	(8-bit)
Configuration:	1	0	(8-bit)
Status Input:			
Status Output:			

Status: Offline

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.

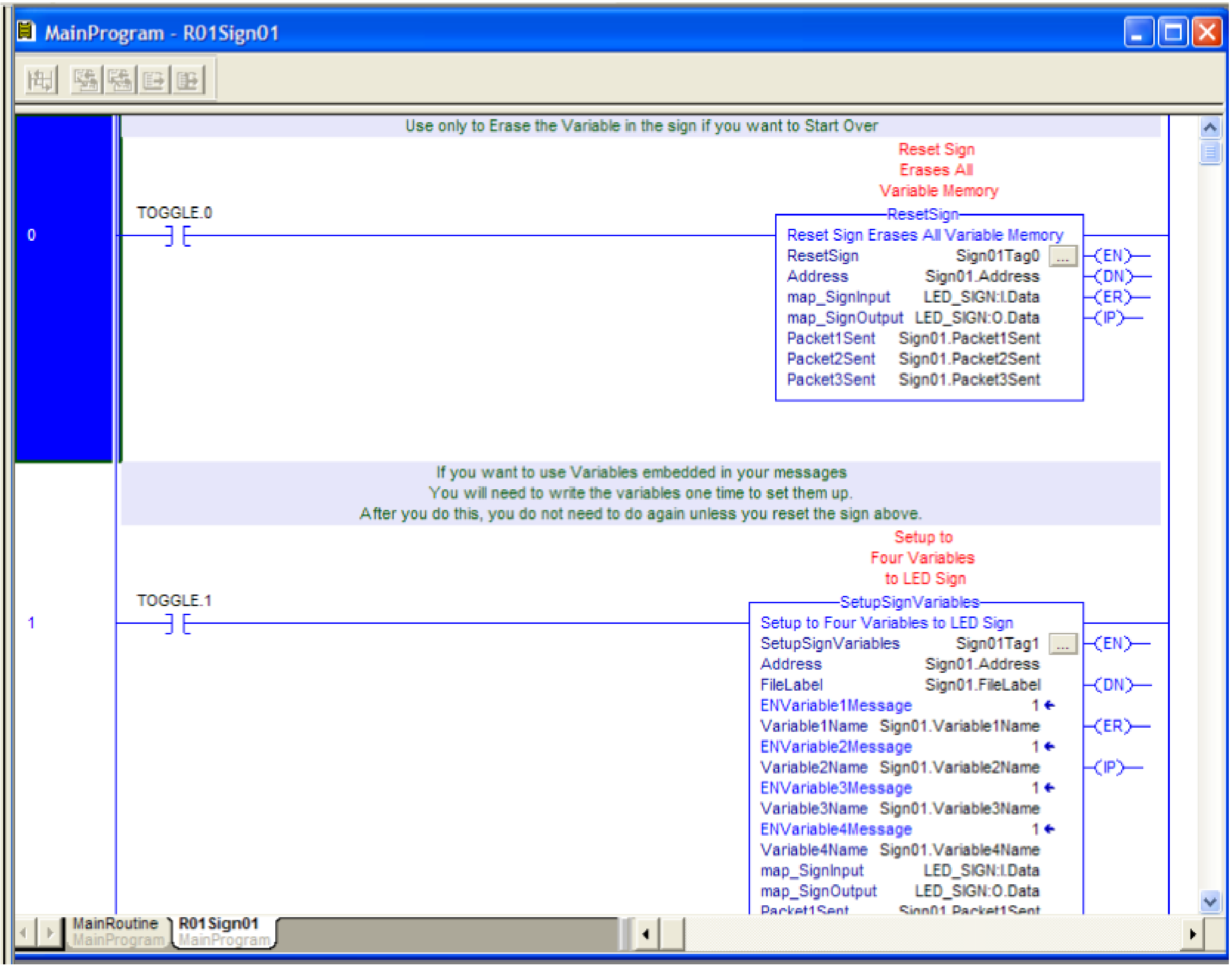


Controller Tags - Electronic_Displays(controller)

Name	Value	F	Style	Data Type	Description
OEEScore	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	'5'	{.		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 ri...
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 SetupSignVaribles which will allocated memory space in the sign to accept dynamic variables that can be written to the sign using the UpdateSignVarible AOI. Send message to the sign by enabling the SendSignMessage AOI.



Use only to Erase the Variable in the sign if you want to Start Over

Reset Sign
Erases All
Variable Memory

ResetSign

Reset Sign Erases All Variable Memory

ResetSign	Sign01Tag0
Address	Sign01.Address
map_SignInput	LED_SIGN:I.Data
map_SignOutput	LED_SIGN:O.Data
Packet1Sent	Sign01.Packet1Sent
Packet2Sent	Sign01.Packet2Sent
Packet3Sent	Sign01.Packet3Sent

If you want to use Variables embedded in your messages
You will need to write the variables one time to set them up.
After you do this, you do not need to do again unless you reset the sign above.

Setup to
Four Variables
to LED Sign

SetupSignVariables

Setup to Four Variables to LED Sign

SetupSignVariables	Sign01Tag1
Address	Sign01.Address
FileLabel	Sign01.FileLabel
ENVariable1Message	1 ←
Variable1Name	Sign01.Variable1Name
ENVariable2Message	1 ←
Variable2Name	Sign01.Variable2Name
ENVariable3Message	1 ←
Variable3Name	Sign01.Variable3Name
ENVariable4Message	1 ←
Variable4Name	Sign01.Variable4Name
map_SignInput	LED_SIGN:I.Data
map_SignOutput	LED_SIGN:O.Data
Packet1Sent	Sign01.Packet1Sent

5 IMPORTING WITH NEW PROGRAM OR EXISITING PROGRAM

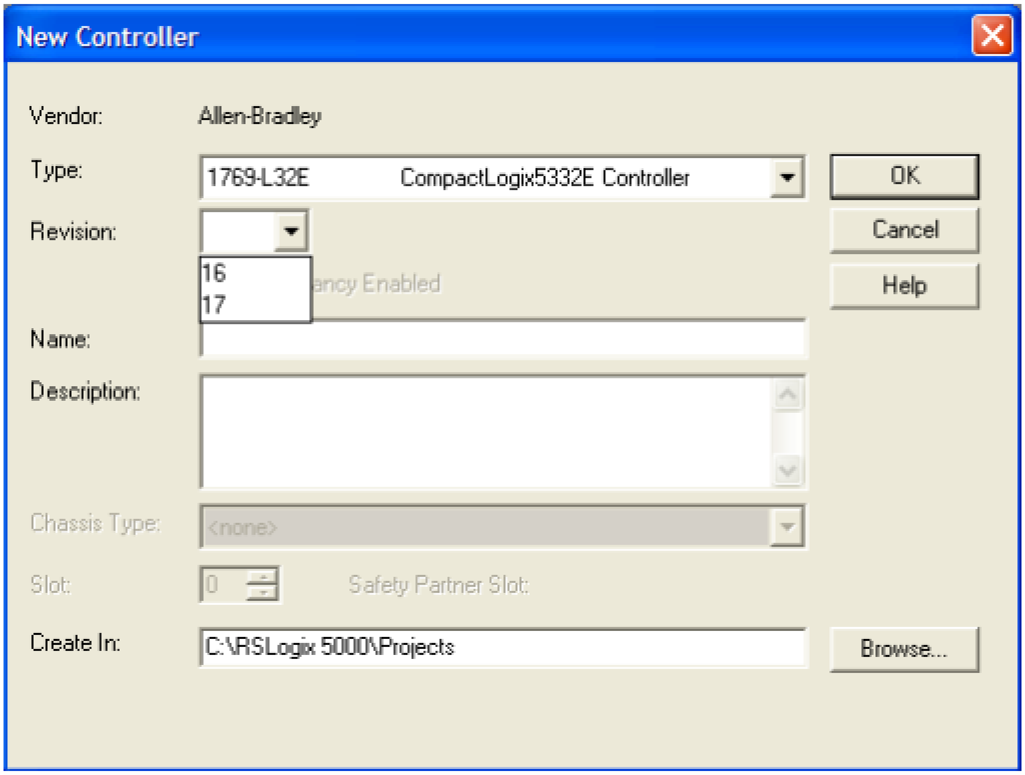
5.1 Start a new project with RS Logix 5000

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

Choose PLC Type.

Choose PLC firmware revision.

Name your PLC Project.



Vendor: Allen-Bradley

Type: 1769-L32E CompactLogix5332E Controller

Revision:
16
17

Name:

Description:

Chassis Type: <none>

Slot: 0 Safety Partner Slot:

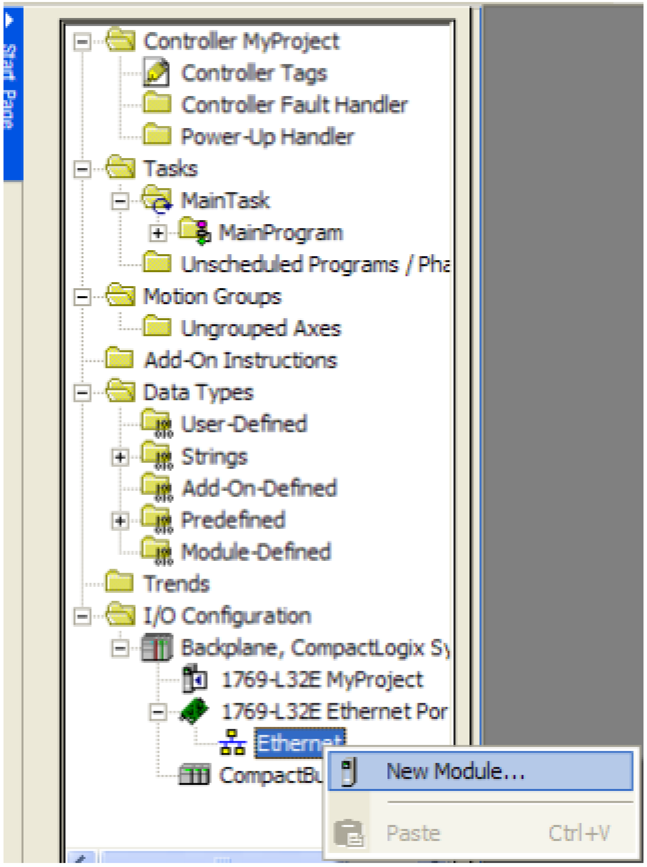
Create In: C:\RSLogix 5000\Projects

5.2 Create a New Ethernet Connection

In the controller tree view.

Right click the Ethernet ICON

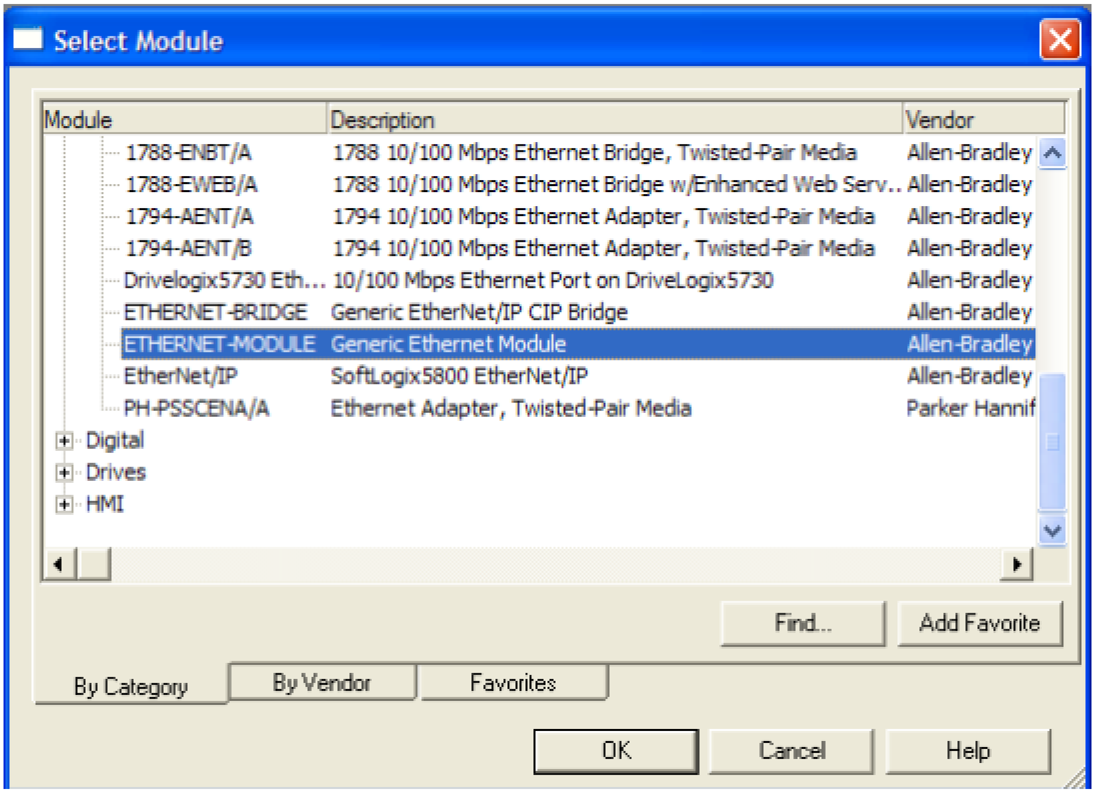
Choose New Module



Choose Communication.

Choose ETHERNET-MODULE (Generic Ethernet Module)

Click OK



Module	Description	Vendor
1788-ENBT/A	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Serv...	Allen-Bradley
1794-AENT/A	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
1794-AENT/B	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
Drivelogix5730 Eth...	10/100 Mbps Ethernet Port on DriveLogix5730	Allen-Bradley
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradley
ETHERNET-MODULE	Generic Ethernet Module	Allen-Bradley
EtherNet/IP	SoftLogix5800 EtherNet/IP	Allen-Bradley
PH-PSSCENA/A	Ethernet Adapter, Twisted-Pair Media	Parker Hannif
⊞ Digital		
⊞ Drives		
⊞ HMI		

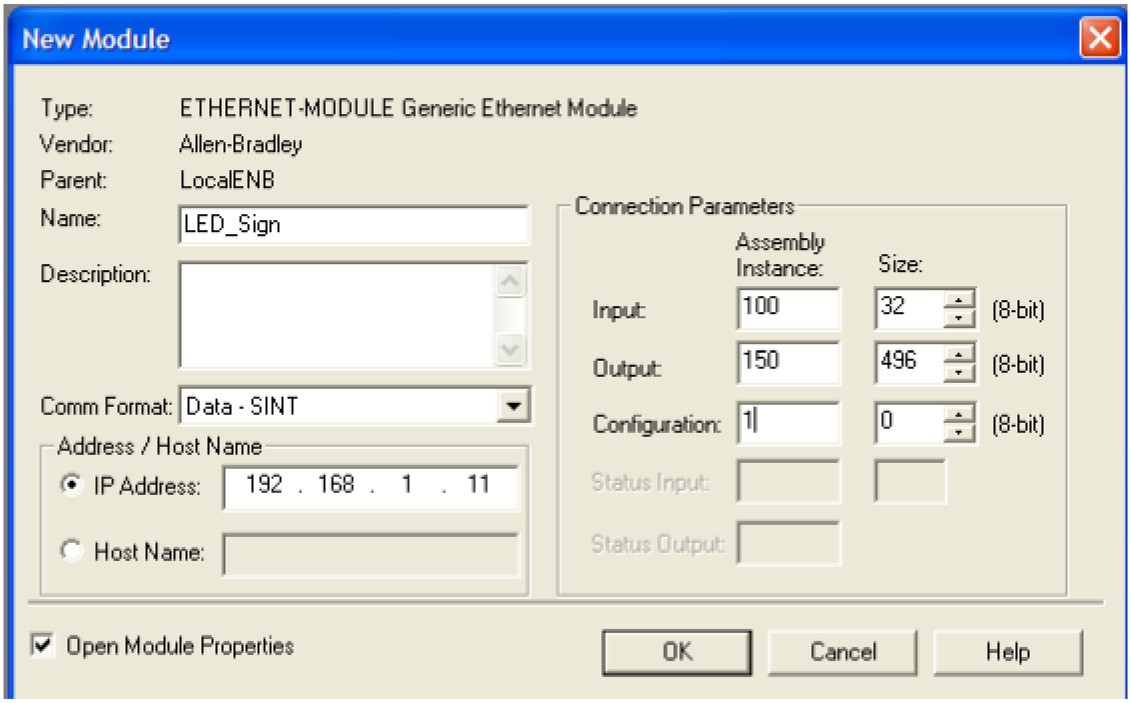
Name the Ethernet Connection:
LED_SIGN
(This will be the prefix name of the tags in the controller.)

Enter the Anybus Communicator 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



New Module

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: LocalENB
Name: LED_Sign
Description:
Comm Format: Data - SINT

Address / Host Name
● IP Address: 192 . 168 . 1 . 11
○ Host Name:

Connection Parameters

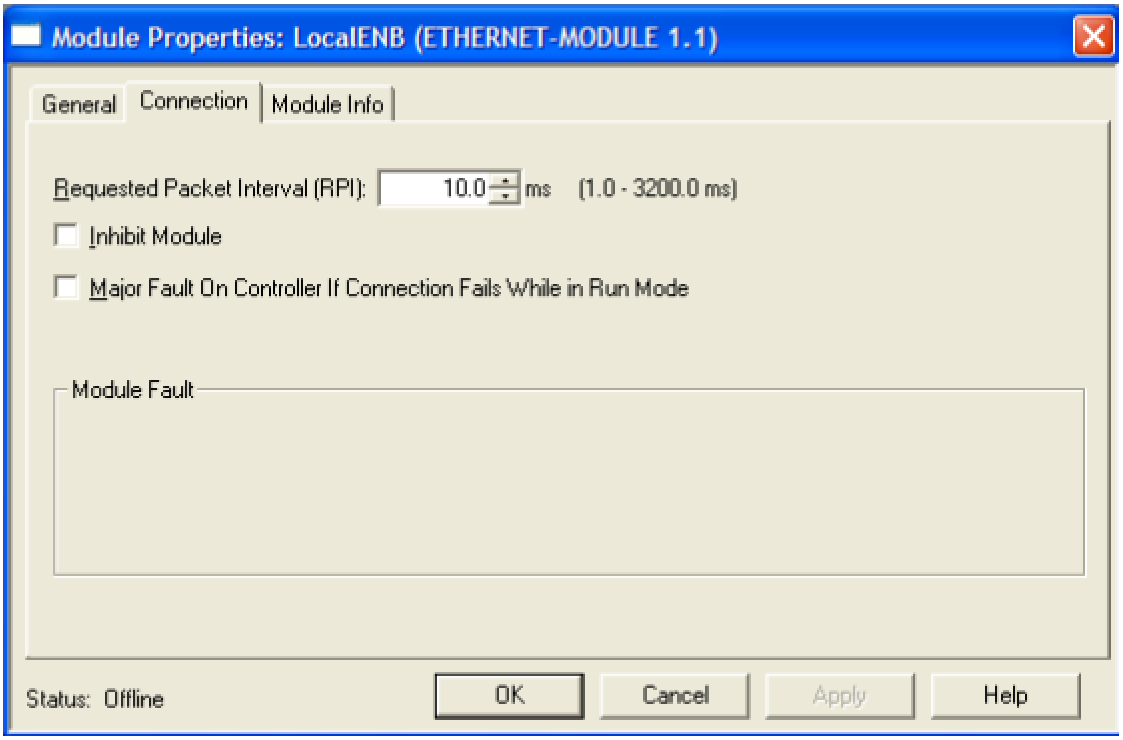
	Assembly Instance:	Size:	
Input:	100	32	(8-bit)
Output:	150	496	(8-bit)
Configuration:	1	0	(8-bit)
Status Input:			
Status Output:			

☑ Open Module Properties

Choose RPI interval:

Default 10.0 ms is ok

Click OK



Requested Packet Interval (RPI): 10.0 ms (1.0 - 3200.0 ms)

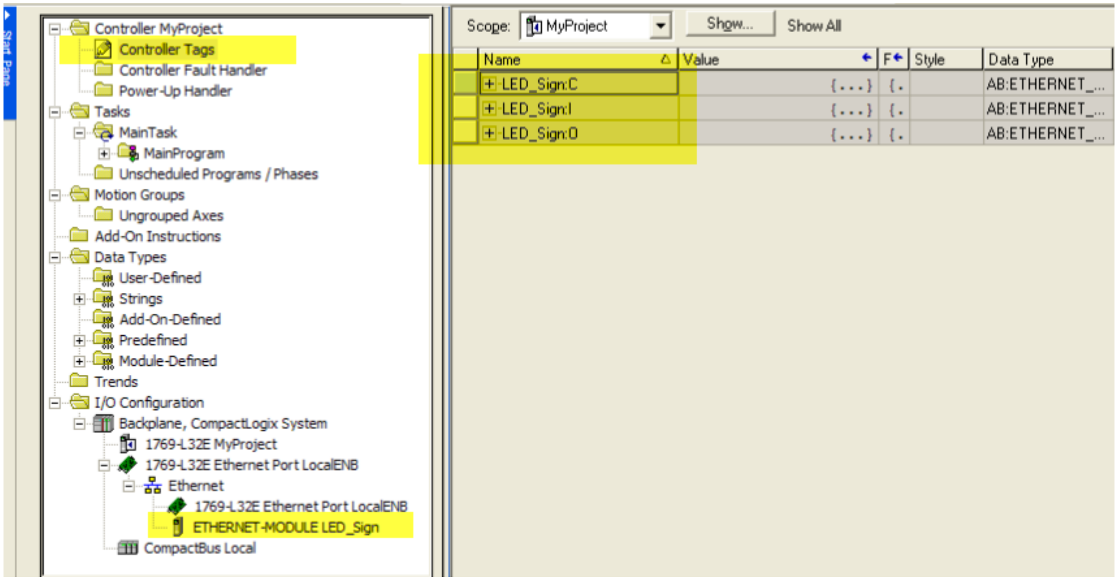
□ Inhibit Module

□ Major Fault On Controller If Connection Fails While in Run Mode

Status: Offline

Confirm Controller Tags

Confirm Ethernet Module is configured



Scope: MyProject

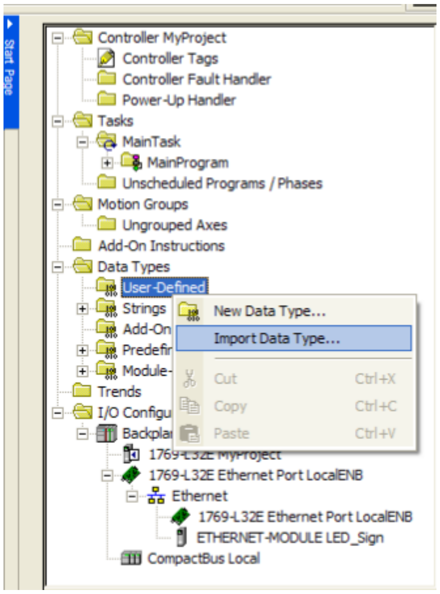
Name	Value	F	Style	Data Type
+ LED_Sign:C	{...}	{.		AB:ETHERNET_...
+ LED_Sign:I	{...}	{.		AB:ETHERNET_...
+ LED_Sign:O	{...}	{.		AB:ETHERNET_...

5.3 Importing Data-Types

In the controller tree view

Right click User-Defined under "Data Types"

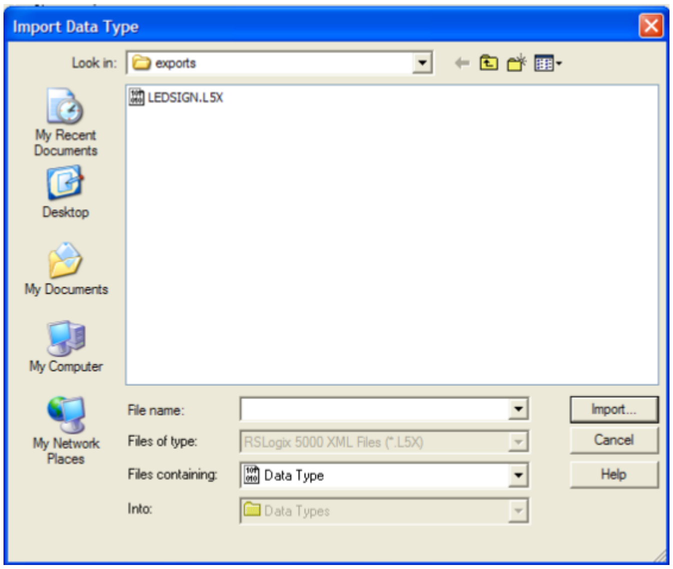
Choose Import Data Type



Browse to the folder containing Data Type

Import LEDSIGN.L5X file

Click OK



Look in: exports

File name:

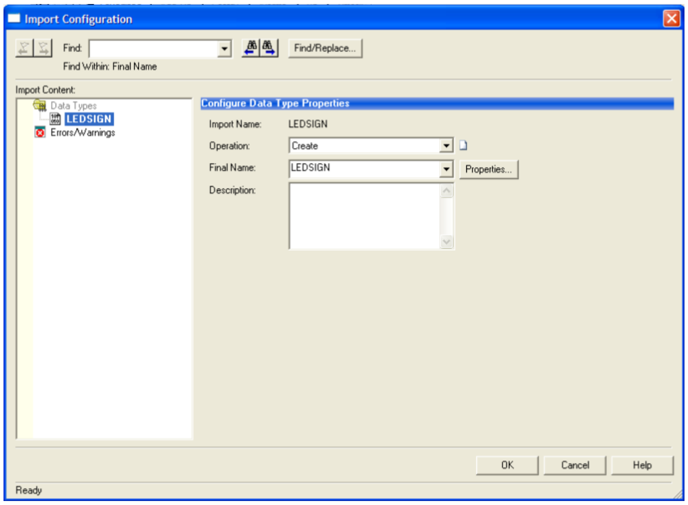
Files of type: RSLogix 5000 XML Files (*.L5X)

Files containing: Data Type

Into: Data Types

Confirm no version conflicts

Click OK



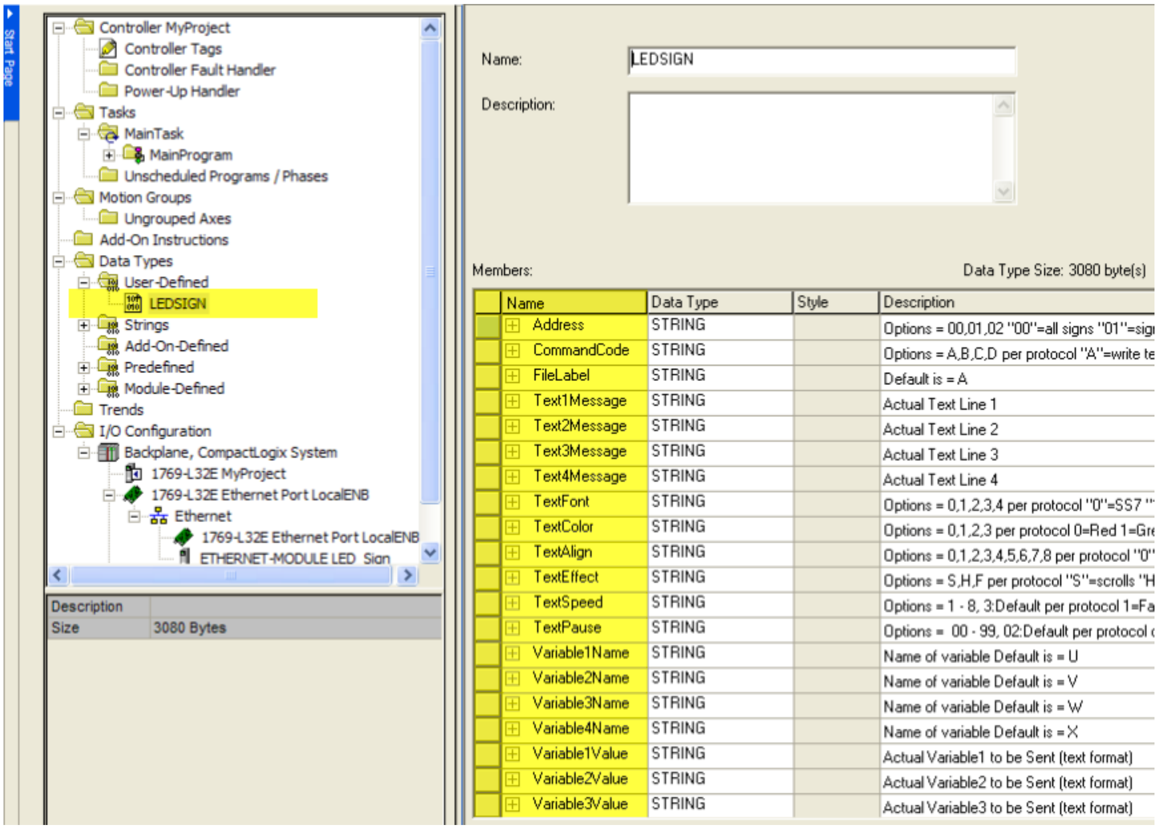
Import Name: LEDSIGN

Operation: Create

Final Name: LEDSIGN

Description:

Confirm Data Type "LEDSIGN"



Description	
Size	3080 Bytes

Name: LEDSIGN

Description:

Members: Data Type Size: 3080 byte(s)

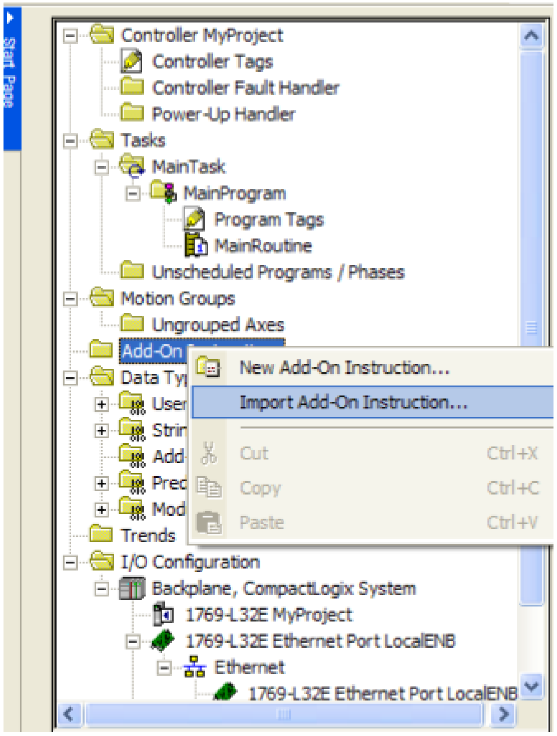
Name	Data Type	Style	Description
Address	STRING		Options = 00,01,02 "00"=all signs "01"=sigı
CommandCode	STRING		Options = A,B,C,D per protocol "A"=write te
FileLabel	STRING		Default is = A
Text1Message	STRING		Actual Text Line 1
Text2Message	STRING		Actual Text Line 2
Text3Message	STRING		Actual Text 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=Gre
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 (
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

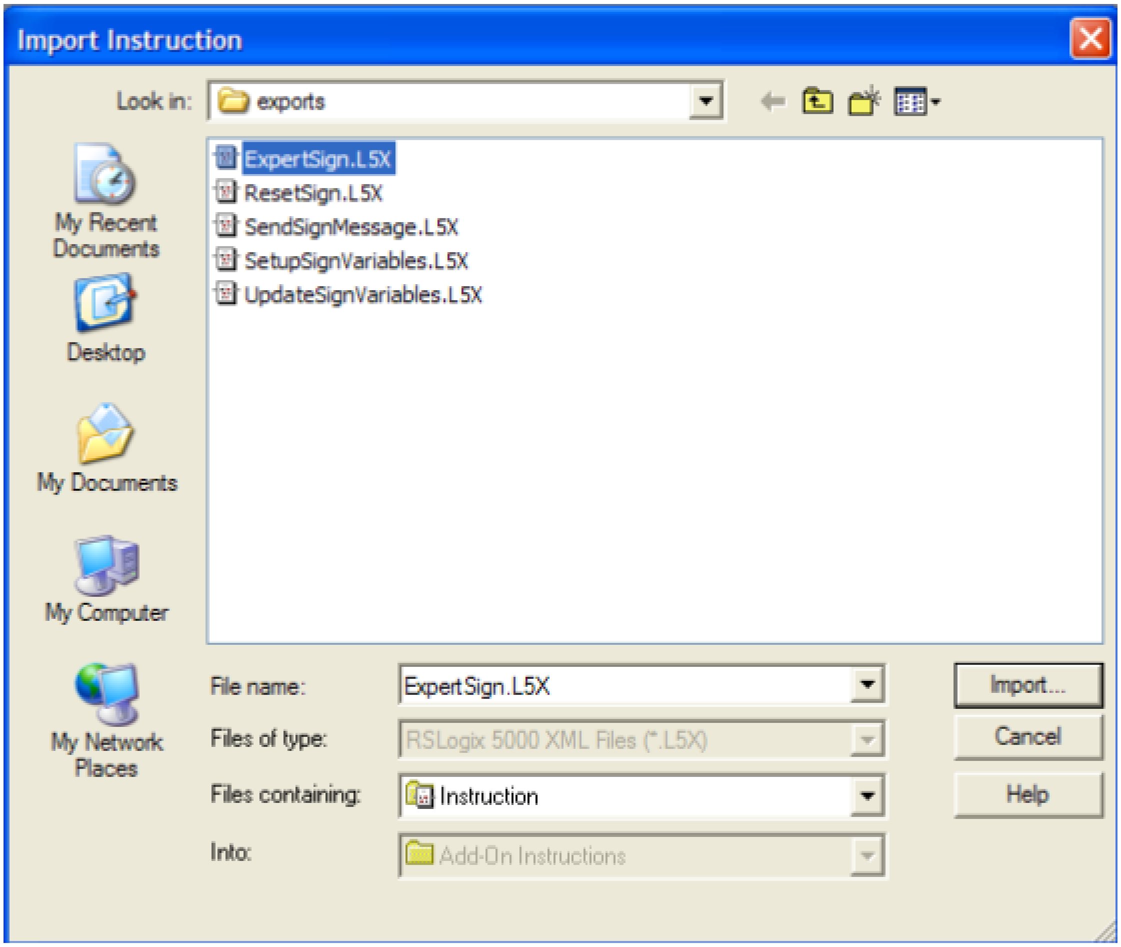


Browse to the folder containing Add-On Instructions

Import all files with .L5X extension.

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

Click OK



Import Instruction

Look in: exports

ExpertSign.L5X
ResetSign.L5X
SendSignMessage.L5X
SetupSignVariables.L5X
UpdateSignVariables.L5X

File name: ExpertSign.L5X

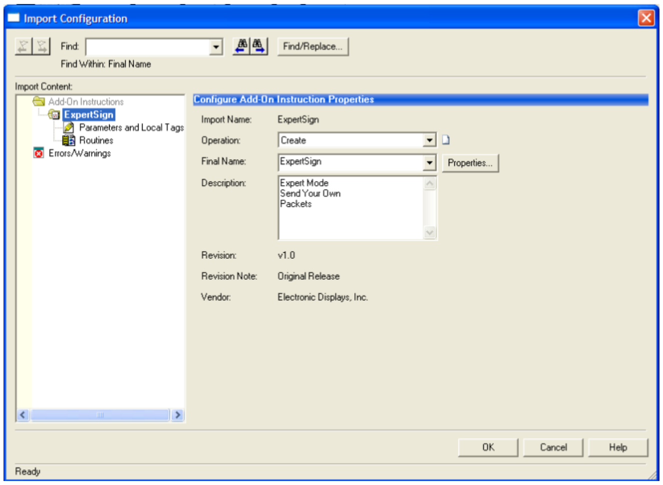
Files of type: RSLogix 5000 XML Files (*.L5X)

Files containing: Instruction

Into: Add-On Instructions

Confirm no version conflicts

Click OK



Import Name: ExpertSign

Operation: Create

Final Name: ExpertSign

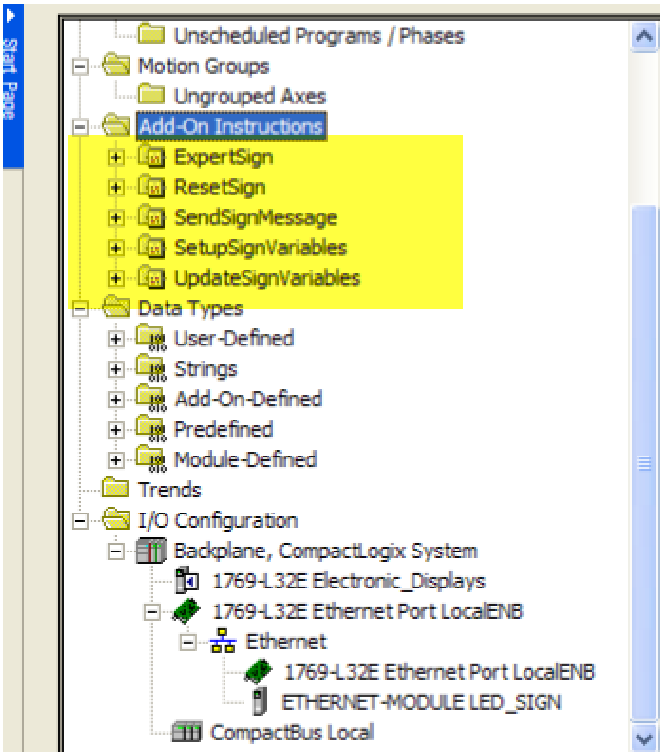
Description: Expert Mode Send Your Own Packets

Revision: v1.0

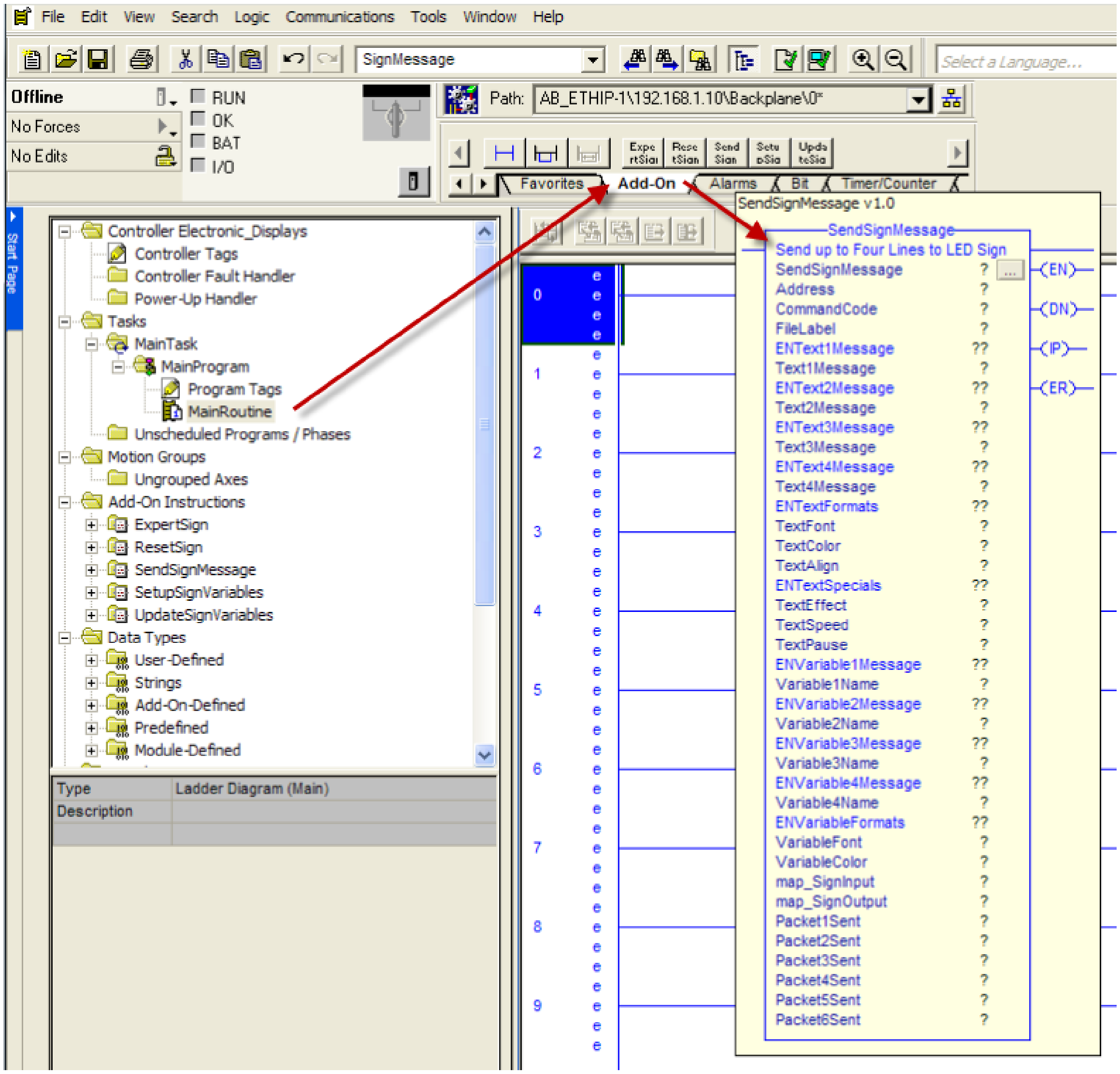
Revision Note: Original Release

Vendor: Electronic Displays, Inc.

Confirm all Add-on Instructions are imported



Confirm AOIs are added to Toolbar in RS Logix 5000



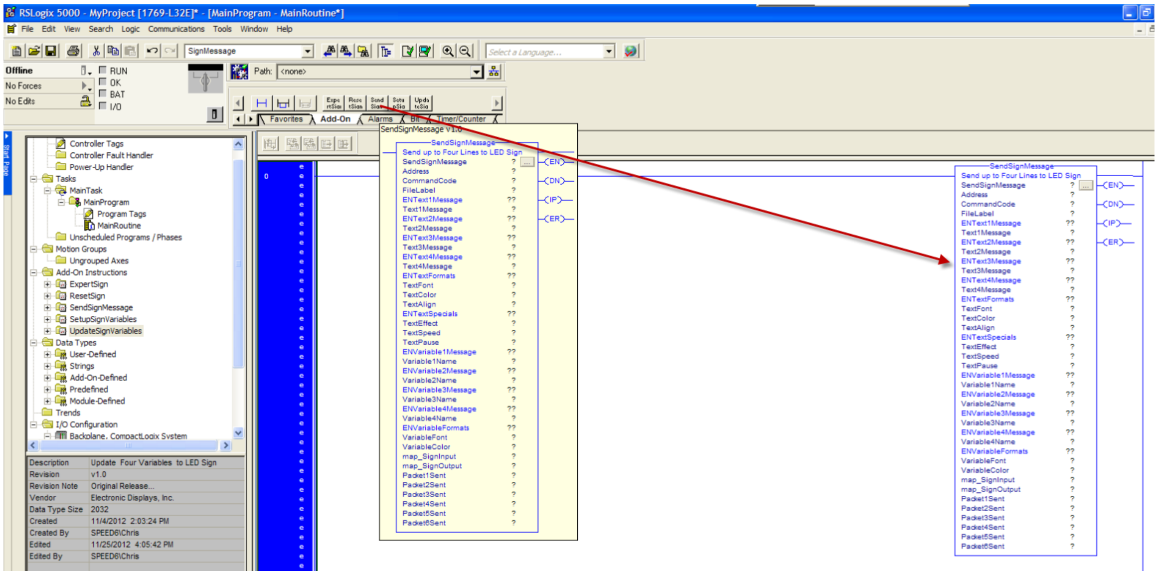
6 USING THE AOI INSTRUCTIONS IN THE PROJECT

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

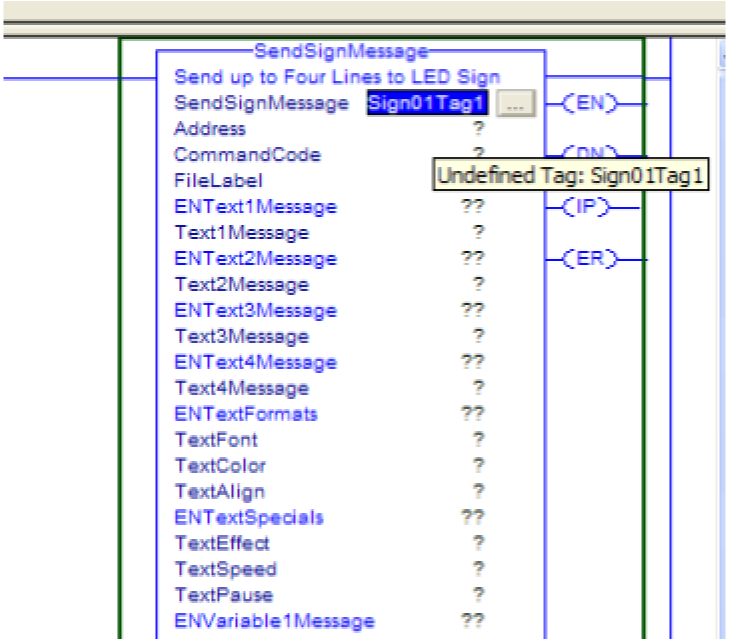


6.2 Creating Unique Tags for AOI

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.

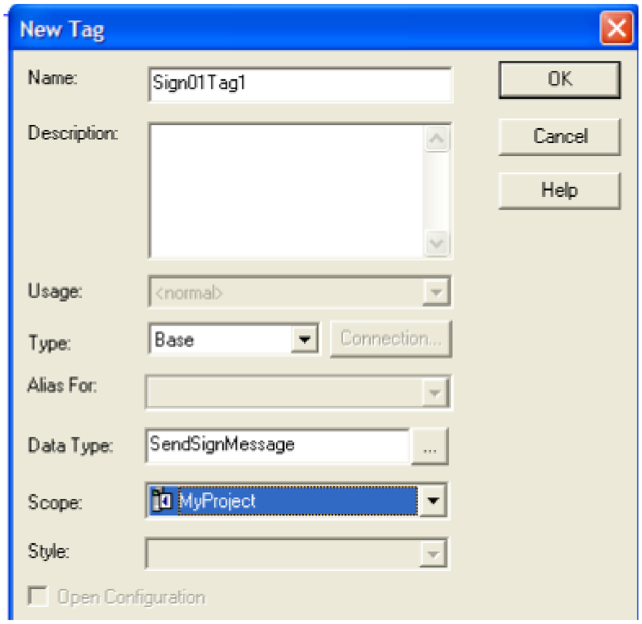
Be sure you scope your tag properly.



SendSignMessage
Send up to Four Lines to LED Sign

SendSignMessage	Sign01Tag1
Address	?
CommandCode	?
FileLabel	
ENText1Message	??
Text1Message	?
ENText2Message	??
Text2Message	?
ENText3Message	??
Text3Message	?
ENText4Message	??
Text4Message	?
ENTextFormats	??
TextFont	?
TextColor	?
TextAlign	?
ENTextSpecials	??
TextEffect	?
TextSpeed	?
TextPause	?
ENVariable1Message	??

Undefined Tag: Sign01Tag1



New Tag

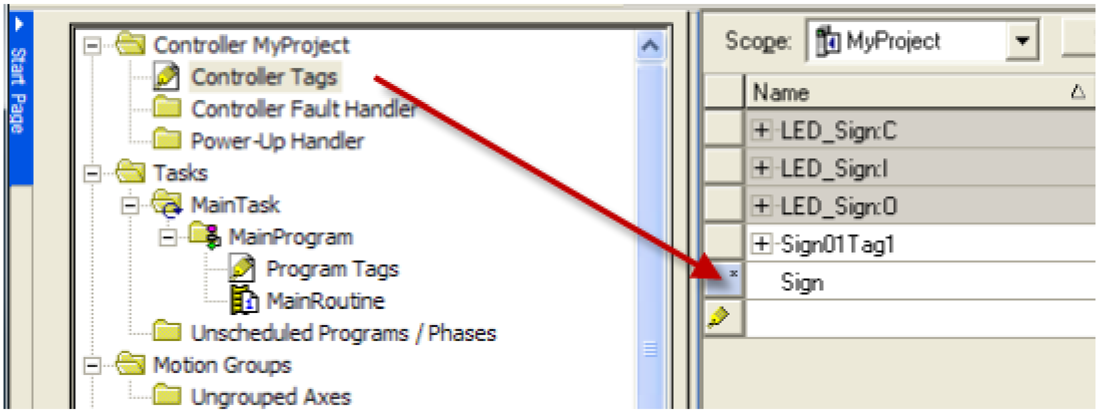
Name: Sign01Tag1
Description:
Usage: <normal>
Type: Base
Alias For:
Data Type: SendSignMessage
Scope: MyProject
Style:
□ Open Configuration

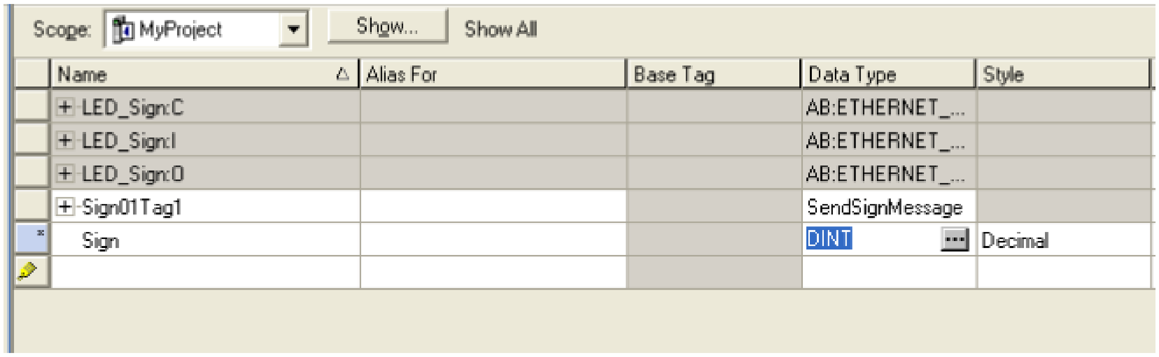
6.3 Creating Unique Tags for Sign Parameters

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

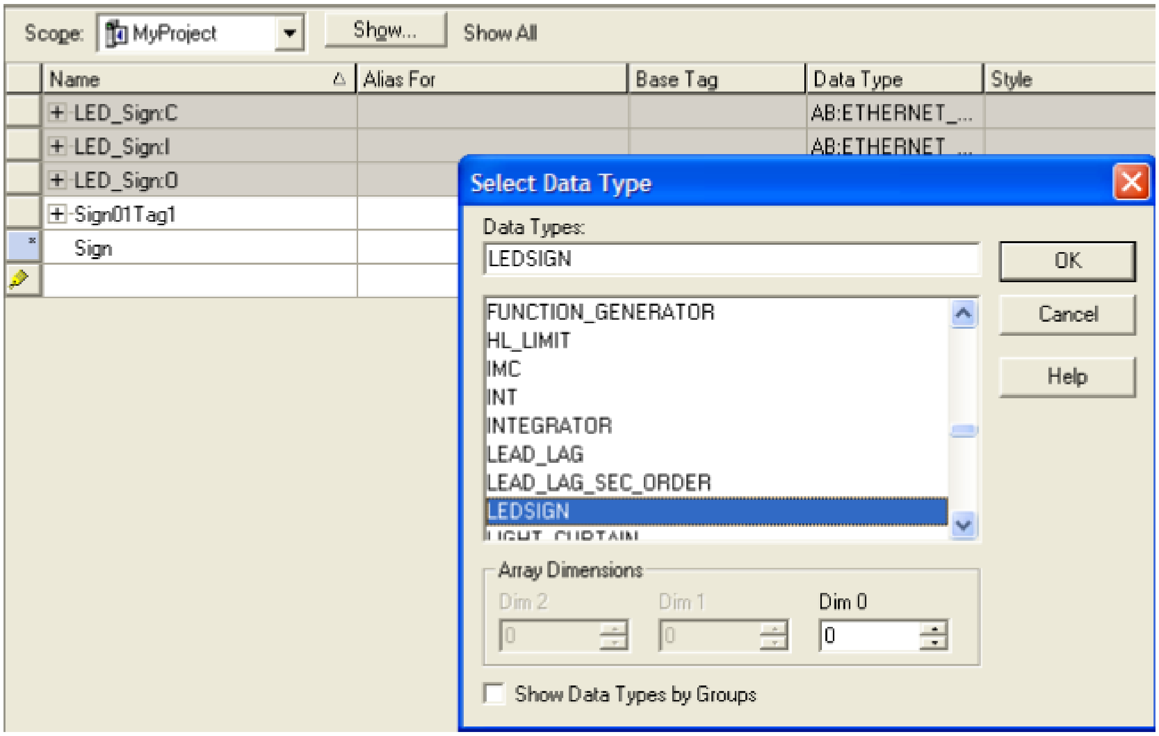




Name	Alias For	Base Tag	Data Type	Style
+ LED_Sign:C			AB:ETHERNET_...	
+ LED_Sign:I			AB:ETHERNET_...	
+ LED_Sign:O			AB:ETHERNET_...	
+ Sign01Tag1			SendSignMessage	
Sign			DINT	Decimal

Map the DATA TYPE of your "Sign" tag to the LEDSIGN data type.

(This data type was imported earlier)



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

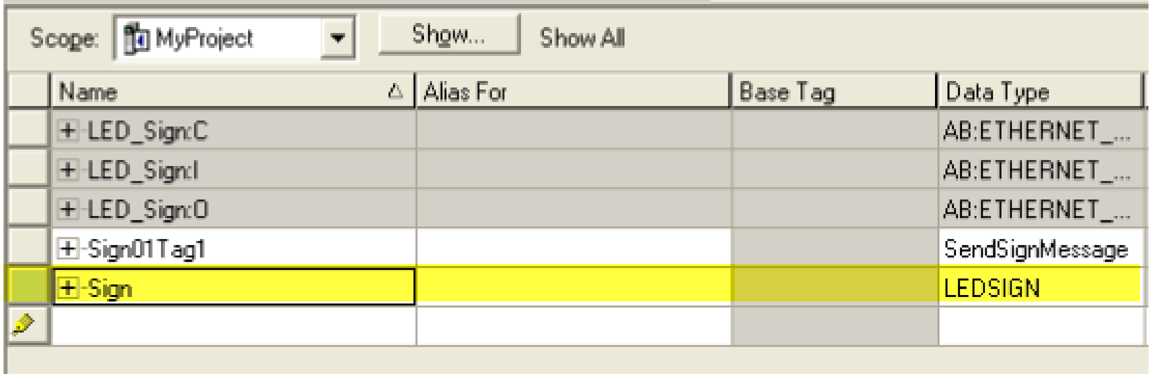
Data Types: LEDSIGN

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



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

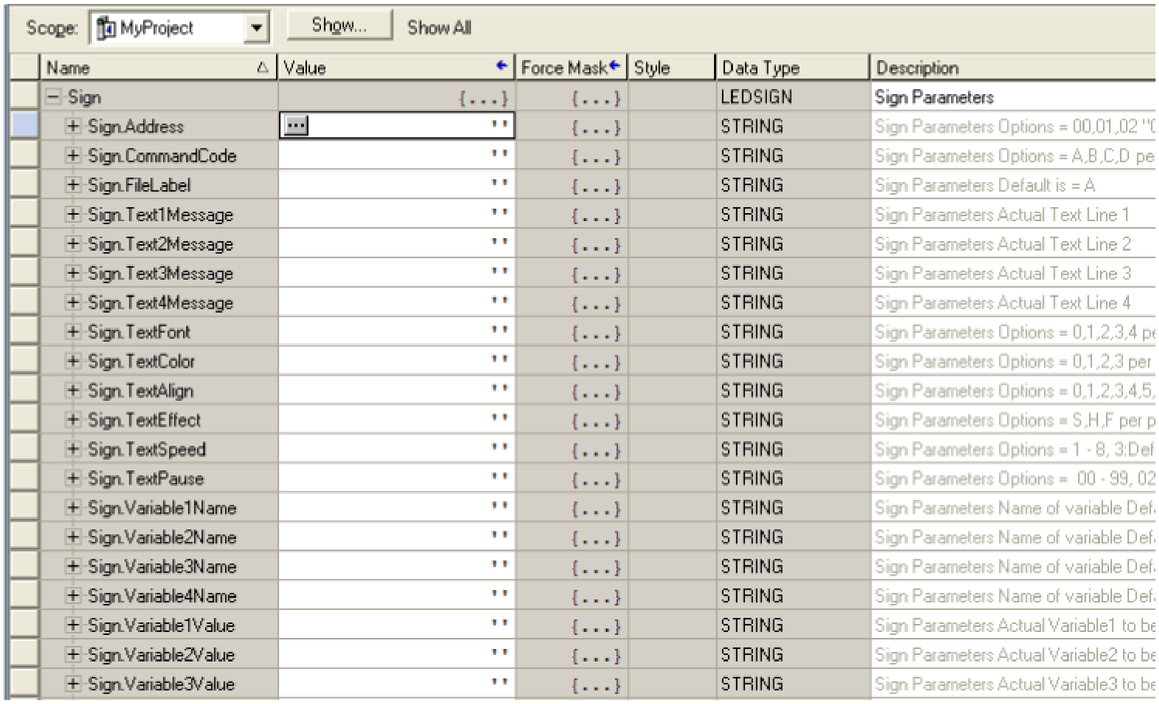
Sign tag will now need parameter information filled out.

Click the ▪▪▪ ICON to begin filling in initial information.

Use the description column for "help".

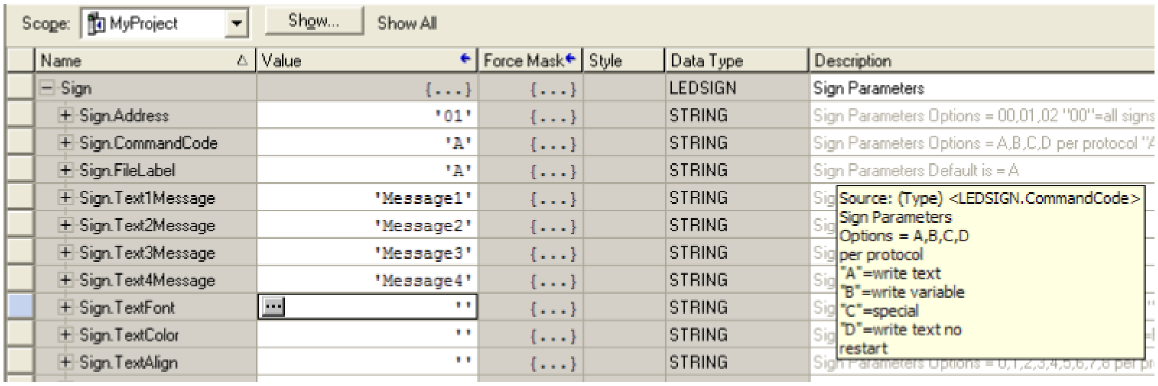
Several of the parameters have "defaults" that can be used.

Hover mouse over description column to see details.



Scope: MyProject ▾ Show... Show All

Name	Value	Force Mask	Style	Data Type	Description
⊟ Sign	{...}	{...}		LEDSIGN	Sign Parameters
⊞ Sign.Address	... ''	{...}		STRING	Sign Parameters Options = 00,01,02 ''(
⊞ 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



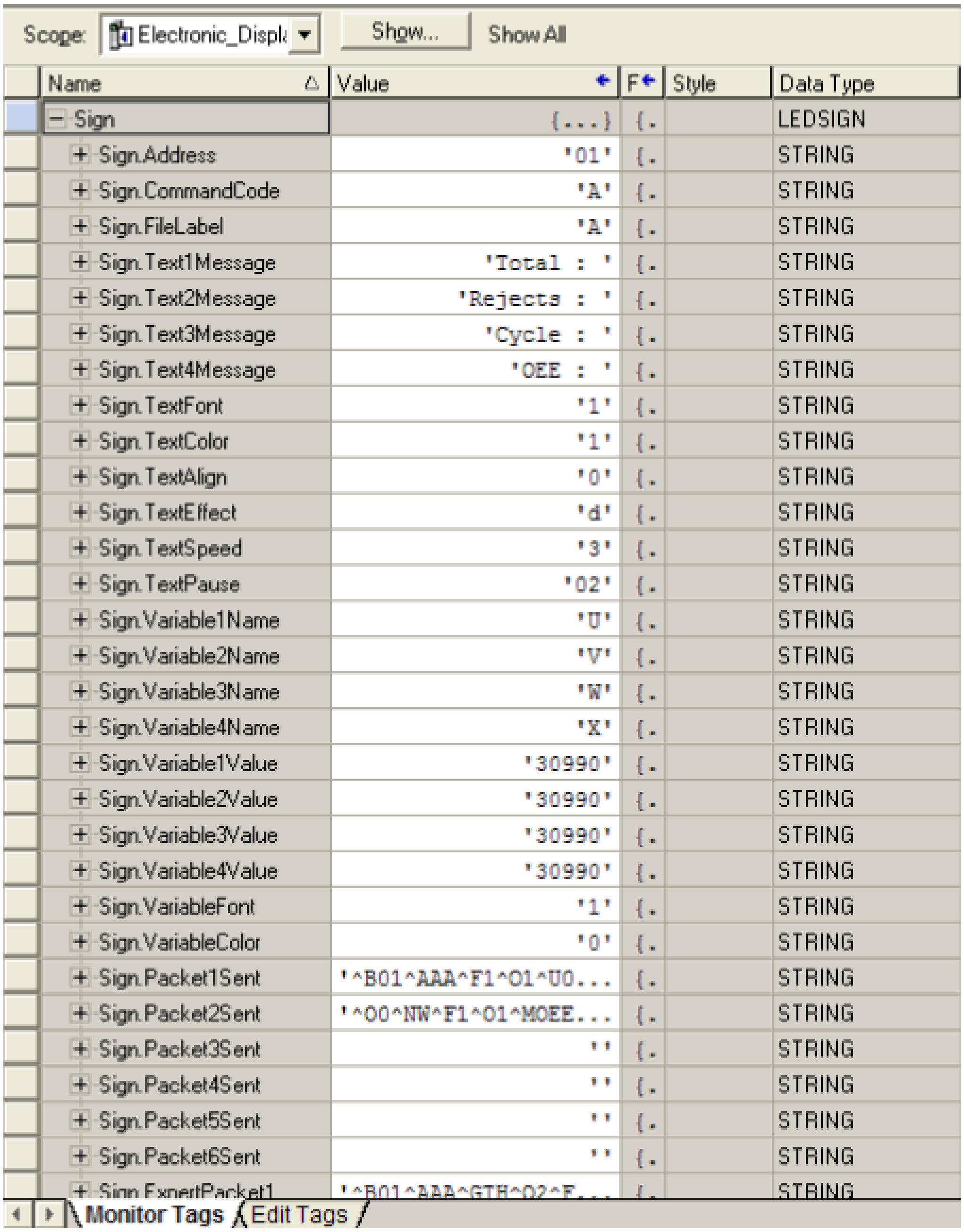
Scope: MyProject ▾ Show... Show All

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	Sig
⊞ Sign.Text2Message	'Message2'	{...}		STRING	Sig
⊞ Sign.Text3Message	'Message3'	{...}		STRING	Sig
⊞ Sign.Text4Message	'Message4'	{...}		STRING	Sig
⊞ Sign.TextFont	... ''	{...}		STRING	Sig
⊞ Sign.TextColor	''	{...}		STRING	Sig
⊞ Sign.TextAlign	''	{...}		STRING	Sign Parameters Options = 0,1,2,3,4,5,6,7,8 per pr

Source: (Type) <LEDSIGN.CommandCode>
Sign Parameters
Options = A,B,C,D
per protocol
"A"=write text
"B"=write variable
"C"=special
"D"=write text no
restart

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

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



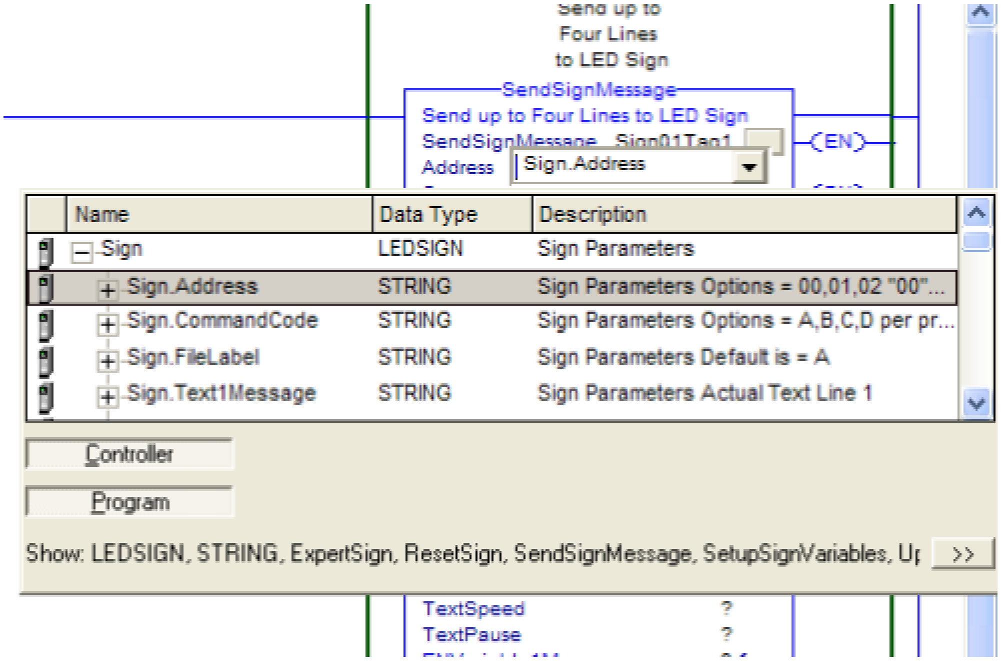
Name	Value	F	Style	Data Type
− Sign	{...}	{.		LEDSIGN
+ Sign.Address	'01'	{.		STRING
+ Sign.CommandCode	'A'	{.		STRING
+ Sign.FileLabel	'A'	{.		STRING
+ Sign.Text1Message	'Total : '	{.		STRING
+ Sign.Text2Message	'Rejects : '	{.		STRING
+ Sign.Text3Message	'Cycle : '	{.		STRING
+ Sign.Text4Message	'OEE : '	{.		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^O1^U0...	{.		STRING
+ Sign.Packet2Sent	'^O0^NW^F1^O1^MOEE...	{.		STRING
+ Sign.Packet3Sent	''	{.		STRING
+ Sign.Packet4Sent	''	{.		STRING
+ Sign.Packet5Sent	''	{.		STRING
+ Sign.Packet6Sent	''	{.		STRING
+ Sign.ExpertPacket1	'^B01^AAA^GTH^O2^F...	{.		STRING

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.



Send up to
Four Lines
to LED Sign

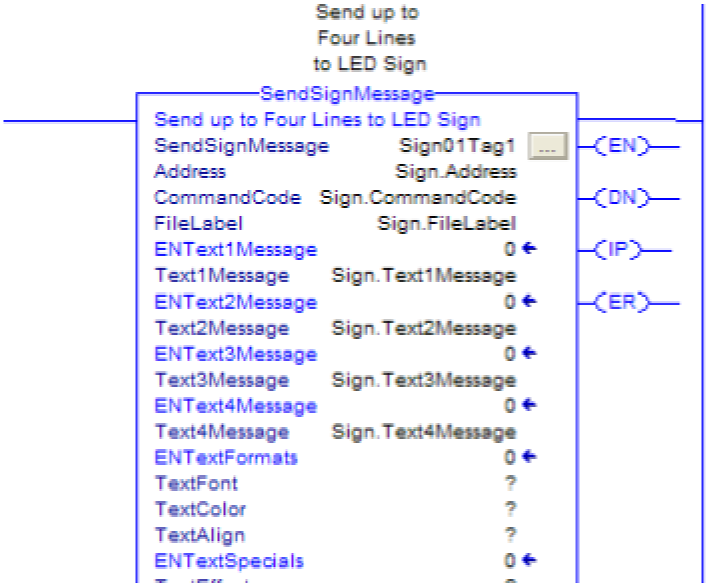
SendSignMessage

Send up to Four Lines to LED Sign
SendSignMessage Sign01Tag1
Address Sign.Address

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

Show: LEDSIGN, STRING, ExpertSign, ResetSign, SendSignMessage, SetupSignVariables, U

TextSpeed ?
TextPause ?



Send up to
Four Lines
to LED Sign

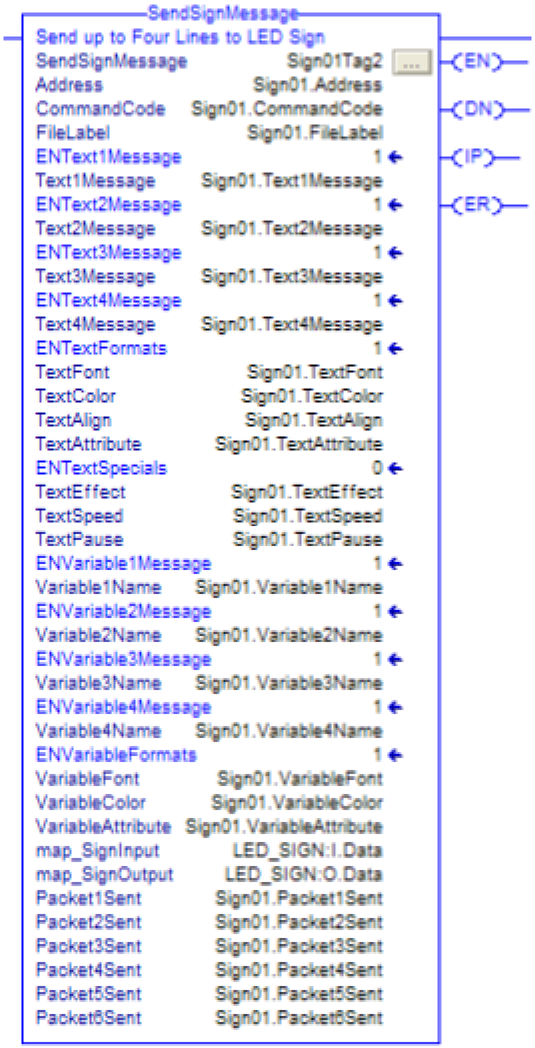
SendSignMessage

Send up to Four Lines to LED Sign
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

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)

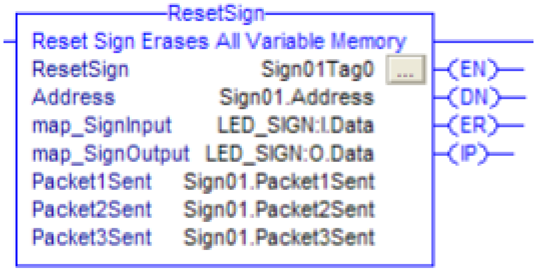


```
                    SendSignMessage
Send up to Four Lines to LED Sign
SendSignMessage                Sign01Tag2  ...   -(EN)-
Address                    Sign01.Address
CommandCode            Sign01.CommandCode        -(DN)-
FileLabel                Sign01.FileLabel
ENText1Message                        1 ←        -(IP)-
Text1Message          Sign01.Text1Message
ENText2Message                        1 ←        -(ER)-
Text2Message          Sign01.Text2Message
ENText3Message                        1 ←
Text3Message          Sign01.Text3Message
ENText4Message                        1 ←
Text4Message          Sign01.Text4Message
ENTextFormats                         1 ←
TextFont                  Sign01.TextFont
TextColor                Sign01.TextColor
TextAlign                Sign01.TextAlign
TextAttribute        Sign01.TextAttribute
ENTextSpecials                        0 ←
TextEffect              Sign01.TextEffect
TextSpeed                Sign01.TextSpeed
TextPause                Sign01.TextPause
ENVariable1Message                    1 ←
Variable1Name        Sign01.Variable1Name
ENVariable2Message                    1 ←
Variable2Name        Sign01.Variable2Name
ENVariable3Message                    1 ←
Variable3Name        Sign01.Variable3Name
ENVariable4Message                    1 ←
Variable4Name        Sign01.Variable4Name
ENVariableFormats                     1 ←
VariableFont          Sign01.VariableFont
VariableColor        Sign01.VariableColor
VariableAttribute Sign01.VariableAttribute
map_SignInput            LED_SIGN:I.Data
map_SignOutput           LED_SIGN:O.Data
Packet1Sent            Sign01.Packet1Sent
Packet2Sent            Sign01.Packet2Sent
Packet3Sent            Sign01.Packet3Sent
Packet4Sent            Sign01.Packet4Sent
Packet5Sent            Sign01.Packet5Sent
Packet6Sent            Sign01.Packet6Sent
```

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
Text Align	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
VariableAtrribute	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)



ResetSign

Reset Sign Erases All Variable Memory

ResetSign Sign01Tag0

Address Sign01.Address

map_SignInput LED_SIGN:I.Data

map_SignOutput LED_SIGN:O.Data

Packet1Sent Sign01.Packet1Sent

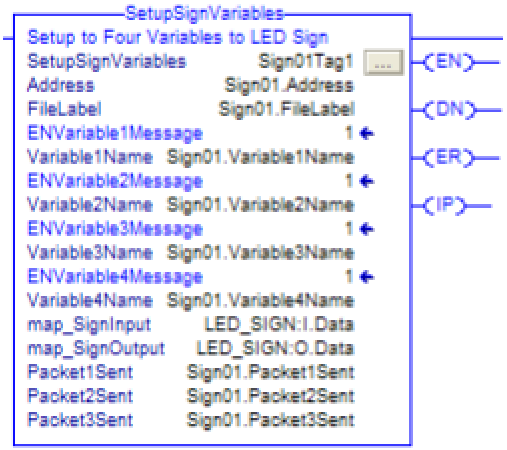
Packet2Sent Sign01.Packet2Sent

Packet3Sent Sign01.Packet3Sent

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)



SetupSignVariables

Setup to Four Variables to LED Sign

SetupSignVariables Sign01Tag1

Address Sign01.Address

FileLabel Sign01.FileLabel

ENVariable1Message 1

Variable1Name Sign01.Variable1Name

ENVariable2Message 1

Variable2Name Sign01.Variable2Name

ENVariable3Message 1

Variable3Name Sign01.Variable3Name

ENVariable4Message 1

Variable4Name Sign01.Variable4Name

map_SignInput LED_SIGN:I.Data

map_SignOutput LED_SIGN:O.Data

Packet1Sent Sign01.Packet1Sent

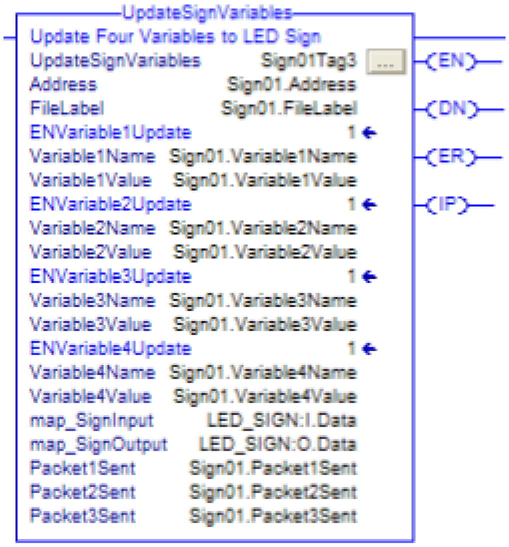
Packet2Sent Sign01.Packet2Sent

Packet3Sent Sign01.Packet3Sent

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)

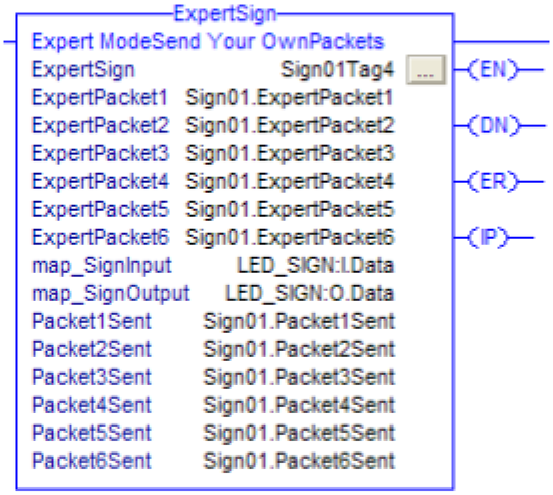


UpdateSignVariables
Update Four Variables to LED Sign
UpdateSignVariables Sign01Tag3
Address Sign01.Address
FileLabel Sign01.FileLabel
ENVariable1Update 1
Variable1Name Sign01.Variable1Name
Variable1Value Sign01.Variable1Value
ENVariable2Update 1
Variable2Name Sign01.Variable2Name
Variable2Value Sign01.Variable2Value
ENVariable3Update 1
Variable3Name Sign01.Variable3Name
Variable3Value Sign01.Variable3Value
ENVariable4Update 1
Variable4Name Sign01.Variable4Name
Variable4Value Sign01.Variable4Value
map_SignInput LED_SIGN:I.Data
map_SignOutput LED_SIGN:O.Data
Packet1Sent Sign01.Packet1Sent
Packet2Sent Sign01.Packet2Sent
Packet3Sent Sign01.Packet3Sent

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)



ExpertSign
Expert ModeSend Your OwnPackets
ExpertSign Sign01Tag4
ExpertPacket1 Sign01.ExpertPacket1
ExpertPacket2 Sign01.ExpertPacket2
ExpertPacket3 Sign01.ExpertPacket3
ExpertPacket4 Sign01.ExpertPacket4
ExpertPacket5 Sign01.ExpertPacket5
ExpertPacket6 Sign01.ExpertPacket6
map_SignInput LED_SIGN:I.Data
map_SignOutput LED_SIGN:O.Data
Packet1Sent Sign01.Packet1Sent
Packet2Sent Sign01.Packet2Sent
Packet3Sent Sign01.Packet3Sent
Packet4Sent Sign01.Packet4Sent
Packet5Sent Sign01.Packet5Sent
Packet6Sent Sign01.Packet6Sent

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