



# Quick Start Guide

## 1. Allen-Bradley Controller Example Project

This quick start guide describes an example application how to use **CG Drives & Automation VSD (VFX/FDU)** device with Allen-Bradley controller and with HMS one port module.

The version's we have used in the example is: Studio 5000, Logix Designer Version: 21.00 Professional Edition

Revision Level: (CPR 9 SR 5.1)

Allen-Bradley CompactLogix 5370 Controller, 1769-L16ER-BB1B, Firmware Revision 21.11 ,

HMS ABCC- EIP 2-Port, Firmware V.1.12 Build 2, HMS ABCC-EIP 1-Port, Firmware V.2.11 Build 3

Zip [file](#): 506-0174-EDS ABCC EIP 2-port.zip, 005A0000002E0100.eds (2-Port module)

Zip [file](#): 368-8182-EDS ABCC-EIP.zip, 005A000000630200.eds (1-Port module)

### Open the example project

Retrieve the example project *CLOGIX5000\_GC\_EthernetIP\_2\_Port.ACD*.

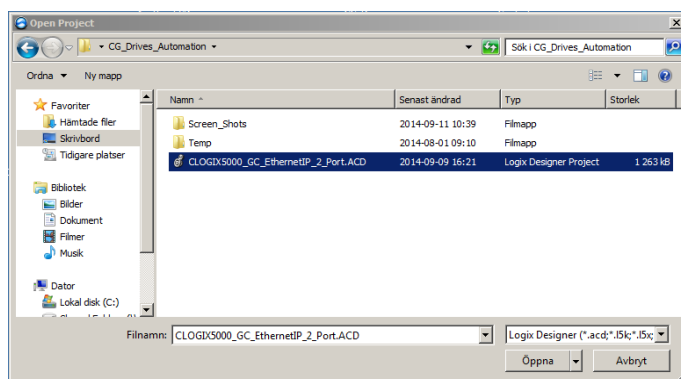
Startup the Studio 5000.

Select: *Open -> Existing Project*



### Select the project file

Select: the project file *CLOGIX5000\_GC\_EthernetIP\_2\_Port.ACD*.





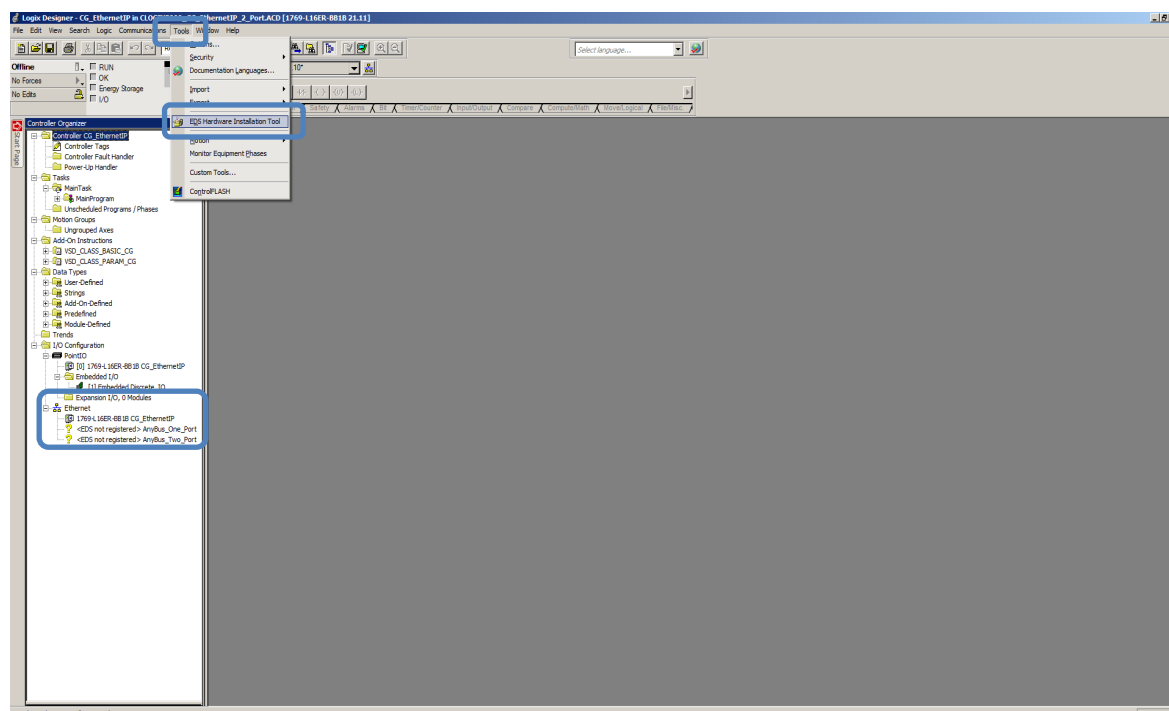
# Quick Start Guide

## Install the EDS file

Install the EDS file “005A0000002E0100” for two device.

Open EDS hardware Installation Tool

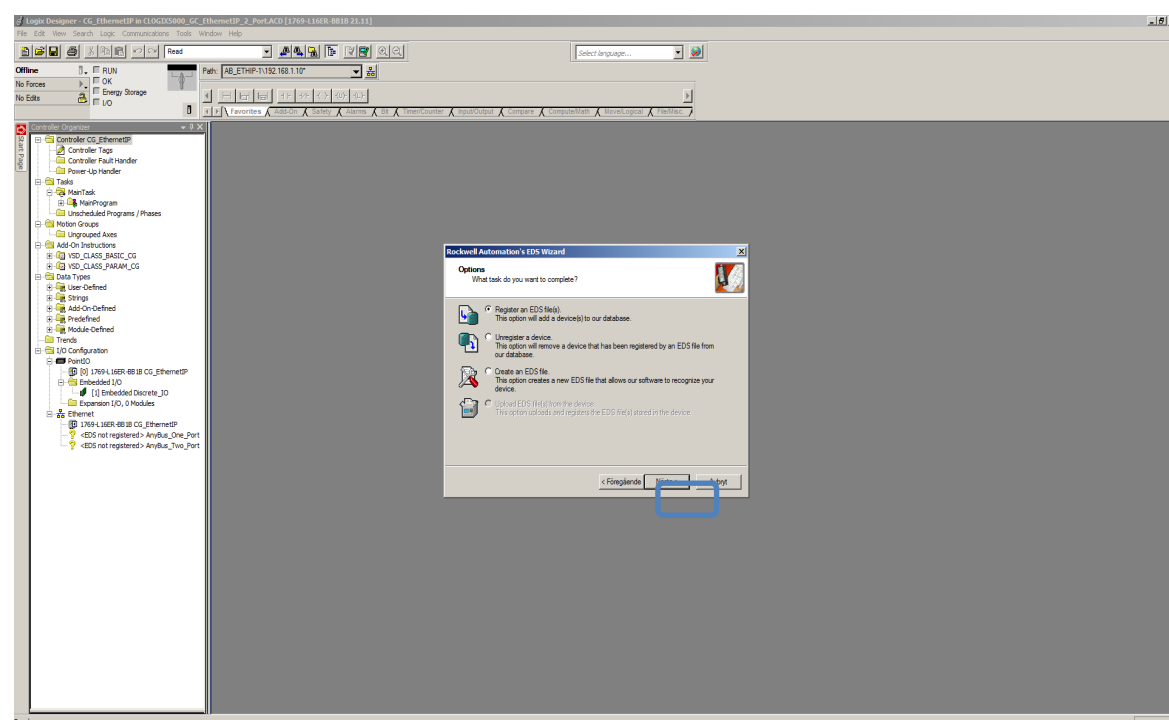
Select *Tools* from the Logix Designer menu and select the *EDS Hardware Installation Tool* from the dropdown list to access the tool. Observe the question mark in the *Controller Organizer* tree view <EDS not registered> AnyBus on the Ethernet node.



## To install the EDS file

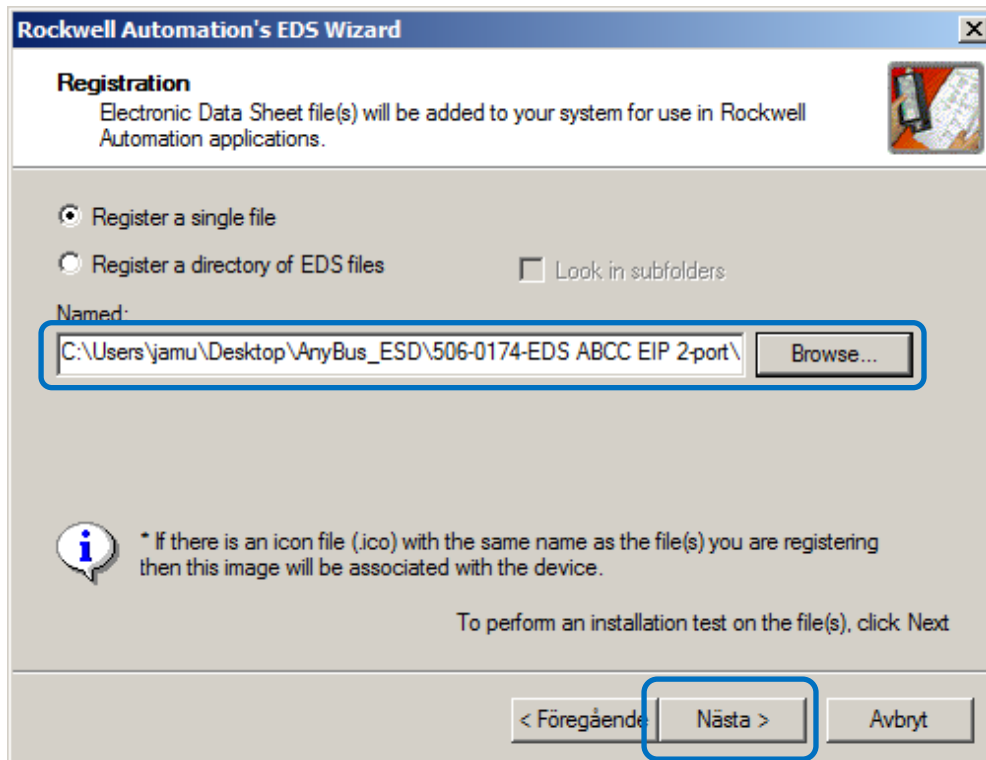
From the EDS hardware installation tool wizard screen, *click Next* button to get the options to handle the EDS files.

**Select option:** Register an EDS file and click on the *Next* button.

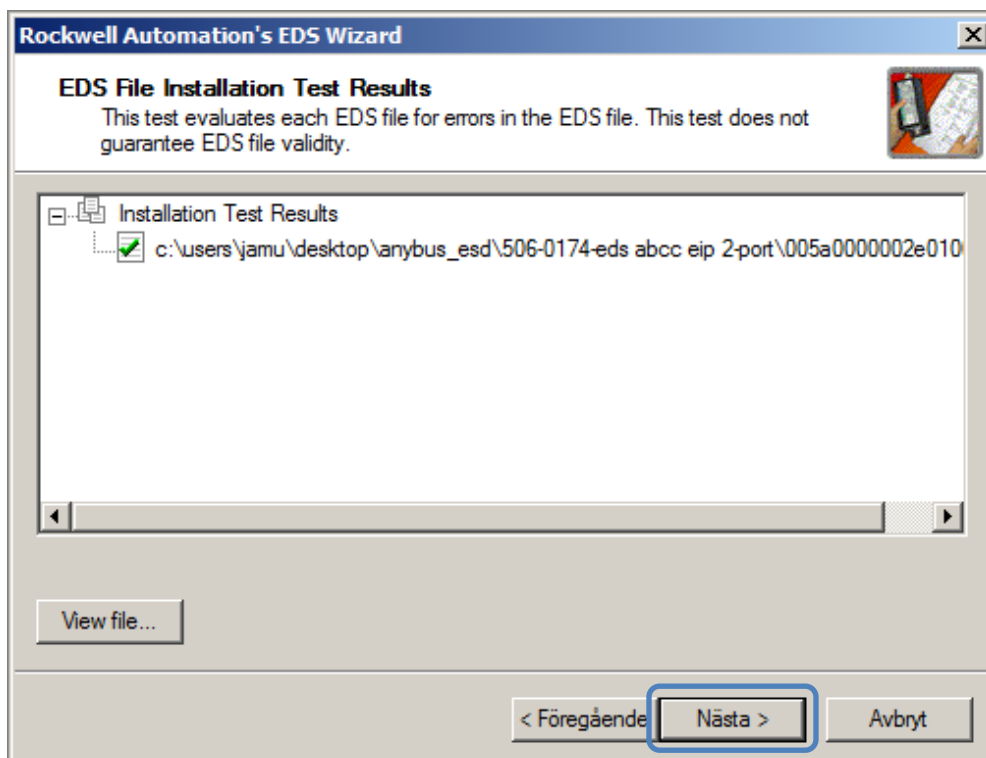


## Quick Start Guide

Select the file to install, *click* on the *Browse* button to get the EDS file “005A0000002E0100.eds” from the file dialog. After selecting the file, push the *Next* button.

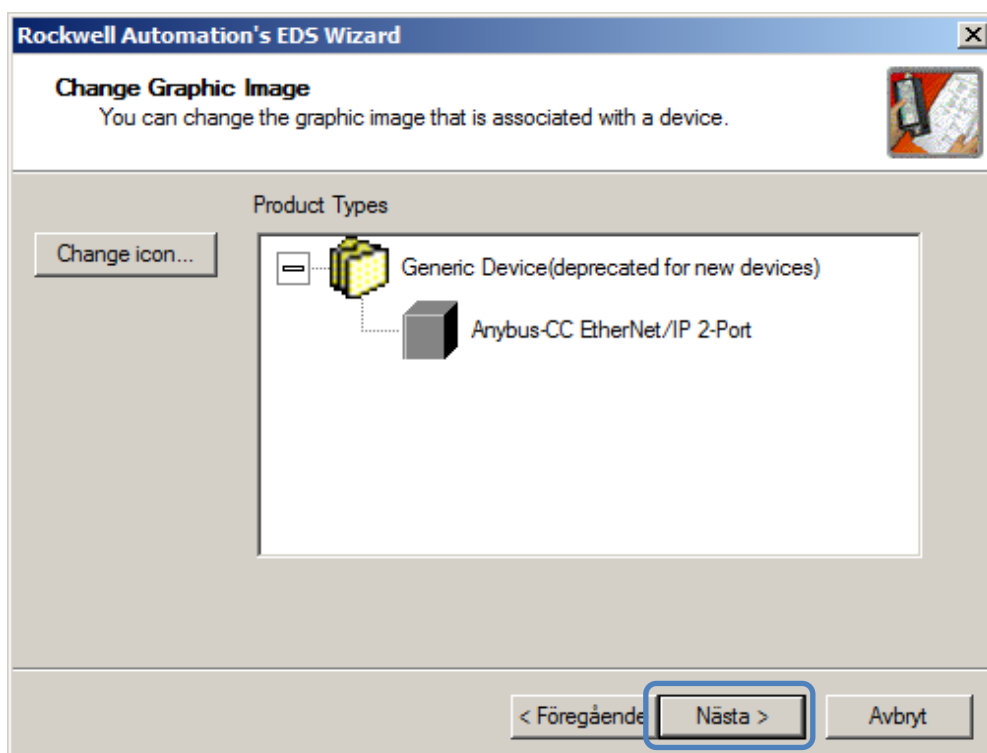


The tool evaluates the EDS file. Click on the *Next* Button.

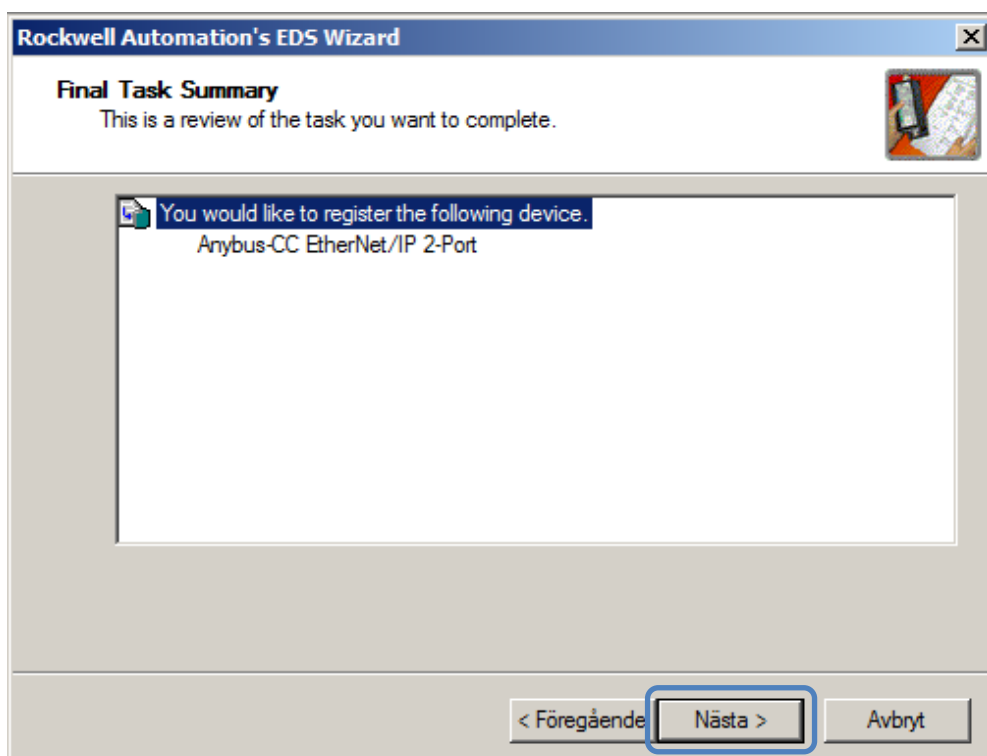


## Quick Start Guide

When the *Change Graphic Image* Dialog shows up, click on the *Next* button

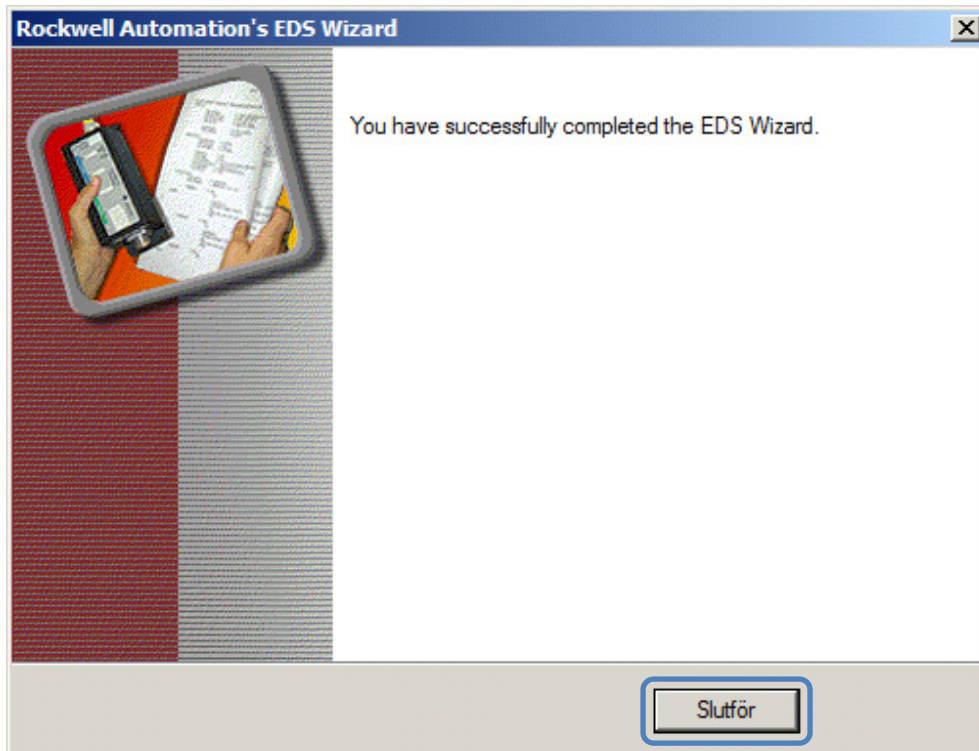


The *Final Task Summary* dialog shows up, click on the *Next* button

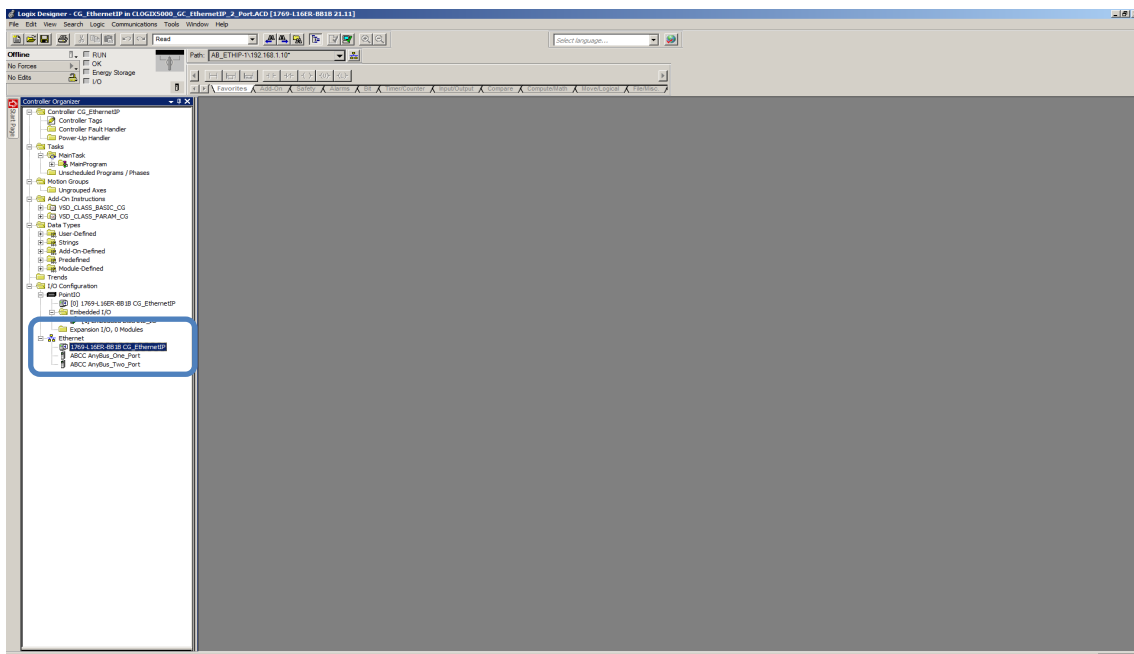


## Quick Start Guide

On the next final dialog click on the *End* button



Now on the question mark *<EDS not registered>* AnyBus should disappear, if not *click* on another item in the tree view to update the tree view node list.





# Quick Start Guide

## 2. Hardware assignment

Configure the hardware setup to apply to your physical hardware.

Assign the IP address and device name, follow the description in the document *Ethernet\_IP\_IO\_Setup\_Device*.

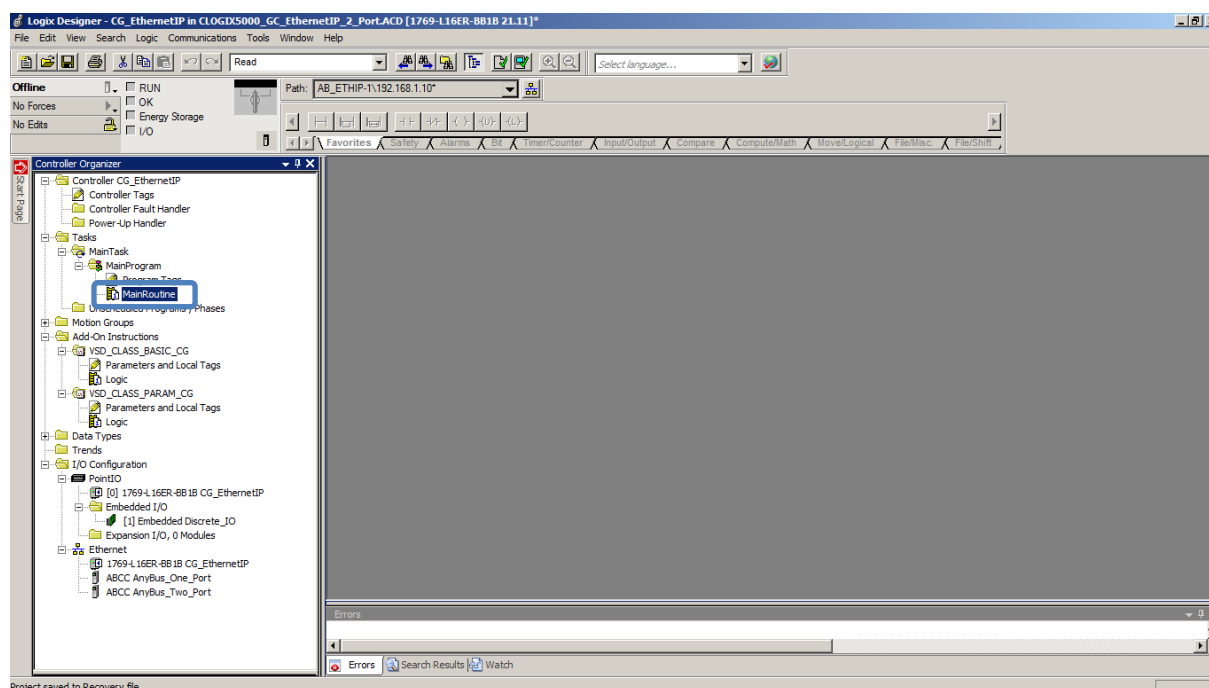
Setup the input and output size for the module for your device.

Per default VSD (VFX/FDU) supports 4 bytes inputs and 4 bytes output basic cyclic process data.

(See the description in the document *Ethernet\_IP\_IO\_Setup\_Device*).

## 3. EtherNet/IP example project

From Logix Designer open the *MainRoutine* and scroll down to the Add-On Instruction *VSD\_CLASS\_BASIC\_CG*.





## Quick Start Guide

### Add-On Instruction VSD\_CLASS\_BASIC.CG

The Add-On Instruction *VSD\_CLASS\_BASIC.CG* is used for basic cyclic process data (Implicit message) Class 1 connection.

To control the state and behaviour of the Add-On object instance there is a derived tag *VSD\_Basic\_ProcessData* which is derived from User defined data type *VSD\_BASIC\_UDT.CG*.

The attributes VSDIN and VSDOUT is referenced to AnyBus\_Two\_Port:I and AnyBus\_Two\_Port:O it is important that the data type (signatures) is correct and corresponds to the ABCC module in the Add-On Attributes.

The screenshot displays the Logix Designer software interface. The left pane shows the Project Explorer with a tree structure including 'Controller CG\_EthernetIP', 'MainProgram', and 'Add-On Instructions'. The main workspace shows the 'ADD ON INSTRUCTION FOR PROCESS DATA EXCHANGE (2-Port Device)' configuration. The configuration table is as follows:

Attribute	Value
Instance Of VSD	Basic Process Data
Add On Instruction	(2-Port Device)
VSDIN	AnyBus_Two_Port:I
VSDOUT	AnyBus_Two_Port:O
Mode	2
AnOutOne	50
AnOutTwo	50
Stat	0

Below the configuration table, there is a section for 'MainRoutine: DEMO EXPLICIT DATA EXCHANGE' with the following details:

- VERSION: 1.0
- AUTHOR: CG Drives & Automation AS
- DATE 2014-07-29

The bottom status bar indicates 'Rung 0 of 17' and 'APP | VER'.



# Quick Start Guide

## Description of the User-defined Data Type VSD\_BASIC\_UDT.CG

Name (member in structure)	Data Type	Description
Start	BOOL	Start of The Drive / Frequency converter = com
Left	BOOL	Direction "Left" (One direction "Left or Right" must be Set =TRUE)
Right	BOOL	Direction "Right"
Reset	BOOL	Reset Error Status Of The Device
ModeSet	SINT	Drive Mode 0=Speed, 1= Torque, 2=V/Hz, 3=Control Panel
ParSet	SINT	Parameter set 0=A, 1=B, 2=C, 3=D
Ref	INT	Speed/Frequency Reference Value (0...32767) Eng.Unit 0-100%
Oper	BOOL	VSD Is In Operation (Run=1/Stop=0)
RunDirR	BOOL	Rotation Right Activated
RunDirL	BOOL	Rotation Left Activated
RelOne	BOOL	State Relay 1 (Default Function = Trip)
RelTwo	BOOL	State Relay 2 (Default Function = Run)
DigOutOne	BOOL	State Dig Out 1 (Default Function= Ready)
DigOutTwo	BOOL	State Dig Out 2 (Default Function =Brake)
RunCtrl	BOOL	VSD Control Via Bus is Activated 0=run/stp Via Remote/Keyboard/Option 1=run/stp Via Com
Mode	SINT	Drive Mode (00=Speed, 01=Torque, 10=V/Hz, 11=Control Panel)
Trip	BOOL	VSD Trip Bit (see Fieldbus Option Manual)
Link	BOOL	VSD Link 1=OK, I/O Device Link established
AnOutOne	SINT	VSD Default=Speed Is Selected For Speed Mode, 0-10V=0-255d,4-20mA=50-255d
AnOutTwo	SINT	VSD Default=Torque Is Selected For Torque Mode, 0-10V=0-255d,4-20mA=50-255d
Stat	SINT	Status Error And Warnings (see Fieldbus Option Manual)

## Description of the members for VSD\_CLASS\_BASIC.CG

Name	Usage	Data Type	Description
EnableIn	Input	BOOL	Enable Input - System Defined Parameter
EnableOut	Output	BOOL	Enable Output - System Defined Parameter
TAG	InOut	VSD_BASIC_UDT.CG	Control Of The Drive / Frequency Converter = com
VSDIN	InOut	_005A:ABCC_85D36 DB4:I:0	VSD LOCAL INPUT 4 BYTES (ABCC-EIP Module)
VSDOUT	InOut	_005A:ABCC_AAB94 180:O:0	VSD LOCAL OUTPUT 4 BYTES (ABCC-EIP Module)
Oper	Output	BOOL	In Operation
RunDirR	Output	BOOL	Rotation Right Activated
RunDirL	Output	BOOL	Rotation Left Activated
RelOne	Output	BOOL	State Relay 1 (Default Function = Trip)
RelTwo	Output	BOOL	State Relay 2 (Default Function = Run)
DigOutOne	Output	BOOL	State Dig Out 1 (Default Function= Ready)
DigOutTwo	Output	BOOL	State Dig Out 2 (Default Function =Brake)
RunCtrl	Output	BOOL	VSD Control Via Bus is Activated 0=run/stp
Mode	Output	SINT	Drive Mode (00=Speed, 01=Torque, 10=V/Hz, 11=Control Panel)
Trip	Output	BOOL	VSD Trip Bit (see fieldbus manual)
Link	Output		VSD Link 1=OK, I/O Device Link established
AnOutOne	Output	SINT	VSD Default=Speed Is Selected For Speed Mode, 0-10V=0-255d,4-20mA=50-255d
AnOutTwo	Output	SINT	VSD Default=Torque Is Selected For Torque Mode, 0-10V=0-255d,4-20mA=50-255d
Stat	Output	SINT	Error And Warnings (see fieldbus manual)





# Quick Start Guide

The data type (signatures) for AnyBus ABCC-EIP in the member declaration for the Add-on is a complex type and need to match the system generated data types (signatures) in the *Controller Tags* list.

The screenshot shows the Logix Designer interface with the Controller Tags list for a CG EthernetIP controller. The list is organized into columns: Name, Base Tag, Data Type, and Description. The tags are grouped by function, including AnyBus modules, VSD modules, and I/O modules. The AnyBus modules are highlighted with a blue box.

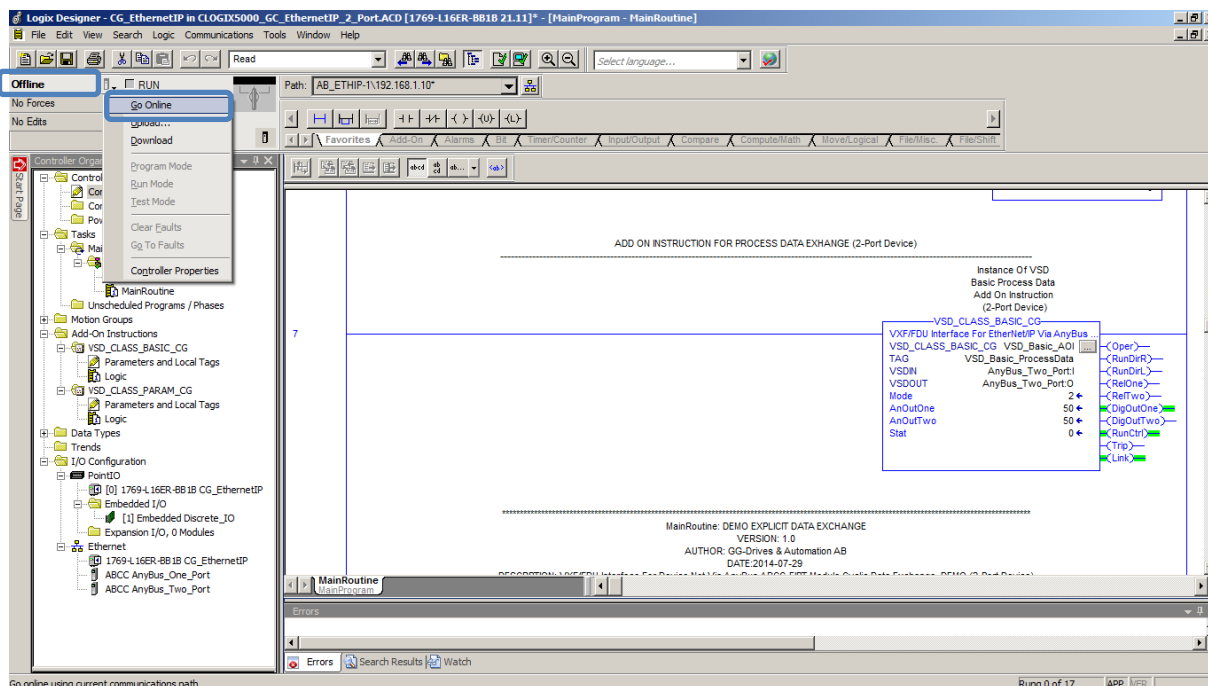
Name	Base Tag	Data Type	Description
AnyBus_One_Port		_005A-ABCC_85D36DB4:1:0	AnyBus ABCC-EIPT Module Input (4 BYTE) Data (1-Port Device)
AnyBus_One_Port_O		_005A-ABCC_AAB94180:0:0	AnyBus ABCC-EIPT Module Output (4 BYTE) Data (1-Port Device)
AnyBus_Two_Port		_005A-ABCC_85D36DB4:1:0	AnyBus ABCC-EIPT Module Input (4 BYTE) Data (2-Port Device)
AnyBus_Two_Port_O		_005A-ABCC_AAB94180:0:0	AnyBus ABCC-EIPT Module Output (4 BYTE) Data (2-Port Device)
VSD			
bLeft		BOOL	Demo VSD Test Left Bit (2-Port Device)
bOnePortGetData		BOOL	Demo VSD Get Data From Device (1-Port Device)
bOnePortLeft		BOOL	Demo VSD Test Left Bit (1-Port Device)
bOnePortReset		BOOL	Demo VSD Reset Bit (1-Port Device)
bOnePortRight		BOOL	Demo VSD Test Right Bit (1-Port Device)
bOnePortSetData		BOOL	Demo VSD Set Data To Device (1-Port Device)
bOnePortSetLen		BOOL	Demo VSD Set The Write Data Byte Length FALSE (0) = 2 bytes, TRUE (1) = 4 Bytes
bOnePortSetPath		BOOL	Demo VSD Set Communication Path For Message (1-Port Device)
bOnePortStart		BOOL	Demo VSD Test Start Bit (1-Port Device)
bReset		BOOL	Demo VSD Reset Bit (2-Port Device)
bRight		BOOL	Demo VSD Test Right Bit (2-Port Device)
bSetData		BOOL	Demo VSD Set Data To Device (2-Port Device)
bSetLen		BOOL	Demo VSD Set The Write Data Byte Length FALSE (0) = 2 bytes, TRUE (1) = 4 Bytes
bSetPath		BOOL	Demo VSD Set Communication Path For Message (2-Port Device)
bStart		BOOL	Demo VSD Test Start Bit (2-Port Device)
iDataWrite		INT	Demo VSD Data Write To Application Data Object (2-Port Device)
iOnePortDataWrite		INT	Demo VSD Data Write To Application Data Object (1-Port Device)
iOnePortReadInstanceID		DINT	Demo VSD Set Instance ID For Read Data To Object (1-Port Device)



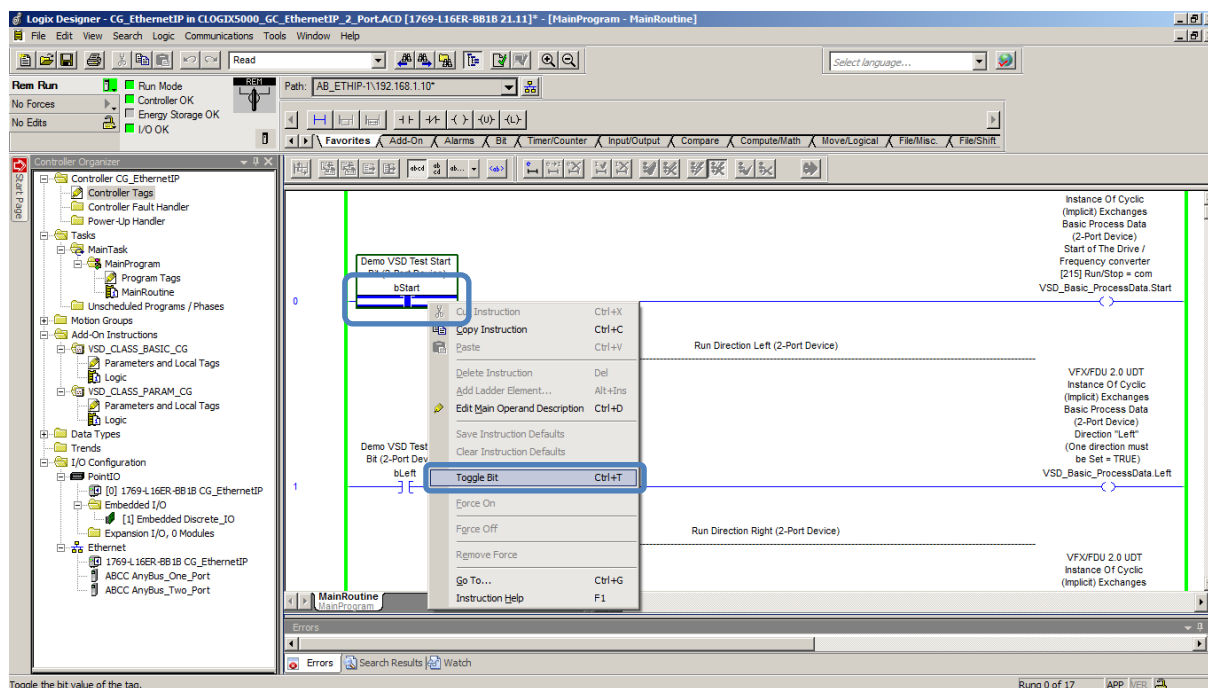
# Quick Start Guide

## Monitor VSD\_Basic\_AOI (VSD\_CLASS\_BASIC\_CG)

To monitor and simulate the *VSD\_Basic\_AOI* (VSD\_CLASS\_BASIC\_CG) Add-on Instruction, go to on line mode. Select the controller tab (icon) and select *Go Online*.

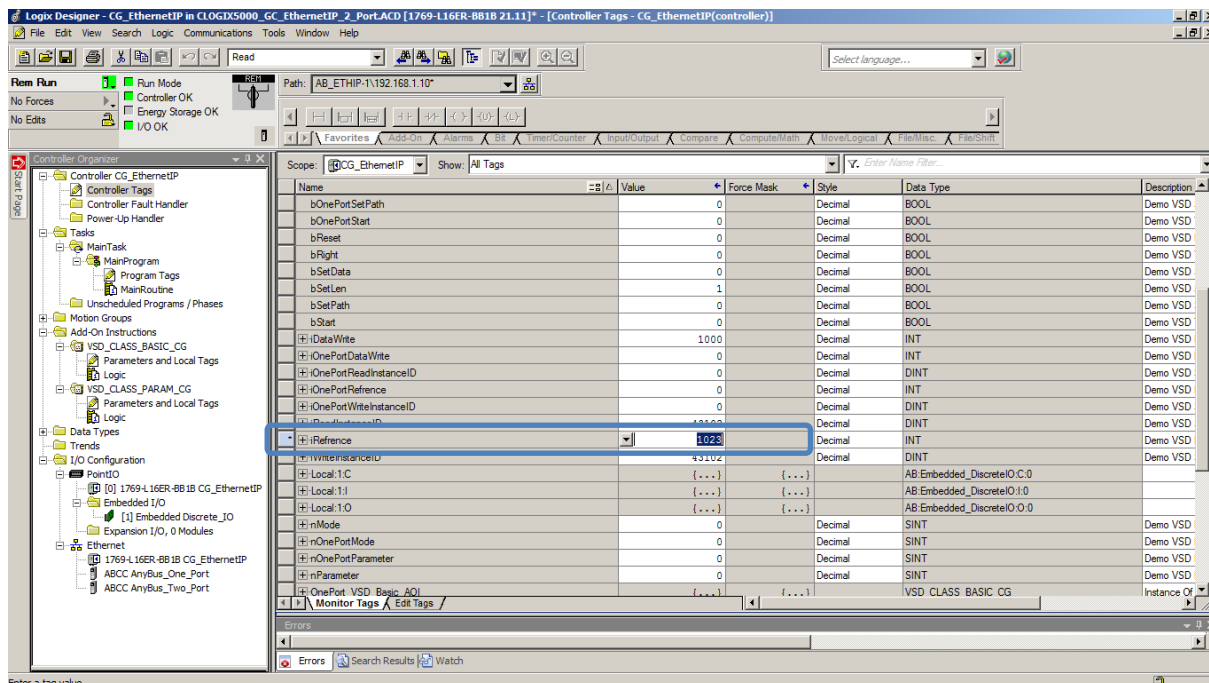


Scroll up to modify input property to the Add-On object *right click* on the input variable e.g. *bStart* Vsd test bit for this example and toggle the bit to TRUE. Modify also the variable value *bRight* bit to TRUE.



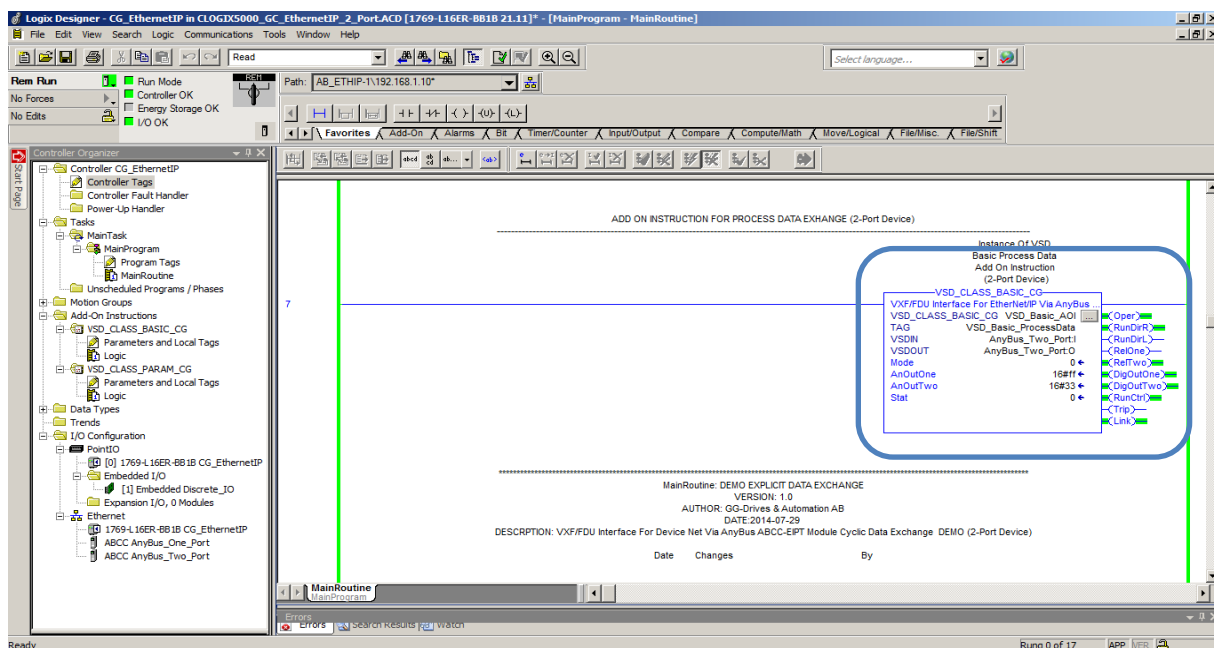
# Quick Start Guide

Scroll down and modify the *iReference* test variable by *right click* on the variable and select *Monitor iReference*. The test variable *iReference* is referenced to *VSD\_Basic\_ModProcessData.Ref* which is bind to the actual VSD reference value. Type in the speed reference value e.g. 1023 decimal from *Monitor Tags* dialog. The reference of 1023d represents 100% engineering unit of the speed. Is set in this example see picture below. Close the *Monitor Tags* dialog to return to *Main Routine* monitor.



When write of values is performed to the VSD block, the outputs should indicate following:

- Operations status on and run right direction active.
- Relay 2 (default run setting in VSD menu [552]).
- Digital output 1 indicates ready (Default setting in VSD menu [541]).
- Digital output 2 releases the brake (default setting in VSD menu [542]).
- Analog out 1 (where speed is default setting in VSD menu [530]) FFh 255d, assuming full speed reached.





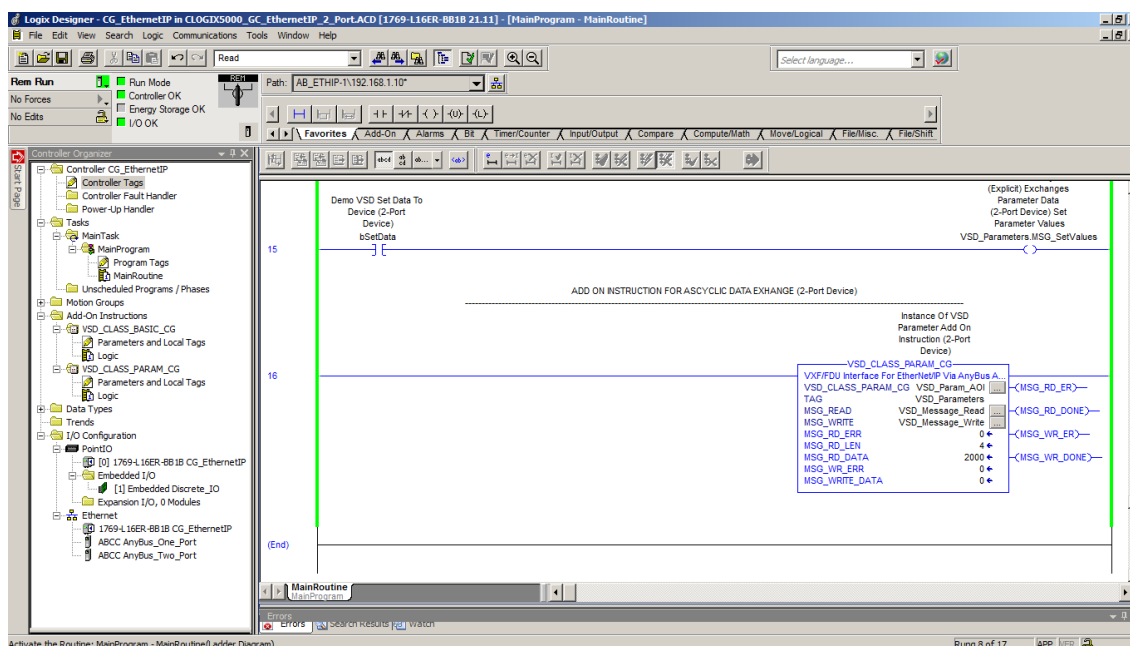
# Quick Start Guide

## Add-On Instruction VSD\_CLASS\_PARAM\_CG

The Add-On instruction VSD\_CLASS\_PARAM\_CG is used to read and write acyclic data (Explicit) Class 3 connection.

Scroll down in the *Main routine* to view and monitor the instance of the Add-On instruction *VSD\_Param\_AOI*. To control the state of the Add-on object instance, there is a derived tag *VSD\_Parameters* referenced to the object. The tag is derived from User defined composite data type *VSD\_MSG\_UDT\_CG*.

For the explicit acyclic CIP message communication there is two instances of the MSG instruction in the Add-on object instance *VSD\_Param\_AOI*. These instructions controls the read and write message to the CIP device ABCC-EIP 2-port.



## Configured Message Instruction data type instance

In this example the MESSAGE structure data type is declared in the *Controller Tags* dialog as *VSD\_Message\_Read* and *VSD\_Message\_Write* and configured as message type *CIP Generic* and service type as *Get Single Attribute* and *Set Single Attribute*.

Name	Base Tag	Data Type	Description
ReadInstanceID		DINT	Demo VSD Set Instance ID For Read Data To Object (2-Port Device)
Reference		INT	Demo VSD Reference Set Integer (2-Port Device)
WriteInstanceID		DINT	Demo VSD Set Instance ID For Write Data To Object (2-Port Device)
Local:1C		AB Embedded_DiscreteIO:0	
Local:1I		AB Embedded_DiscreteIO:1:0	
Local:1Q		AB Embedded_DiscreteIO:0:0	
Mode		SINT	Demo VSD Mode Set Byte (2-Port Device)
OnePortMode		SINT	Demo VSD Mode Set Byte (1-Port Device)
OnePortParameter		SINT	Demo VSD Parameter Set Byte (1-Port Device)
Parameter		SINT	Demo VSD Parameter Set Byte (2-Port Device)
OnePort_VSD_Basic_AOI		VSD_CLASS_BASIC_CG	Instance Of VSD Basic Process Data Add On Instruction (1-Port Device)
OnePort_VSD_Basic_ProcessData		VSD_BASIC_UDT_CG	VFX/FDU 2.0 UDT Instance Of Cyclic (Implicit) Exchanges Basic Process Data (1-Port Device)
OnePort_VSD_Message_Read		MESSAGE	VFX/FDU 2.0 DRIVE PARAMETER READ (Anybus) (1-Port Device)
OnePort_VSD_Message_Write		MESSAGE	VFX/FDU 2.0 DRIVE PARAMETER WRITE (Anybus) (1-Port Device)
OnePort_VSD_Param_AOI		VSD_CLASS_PARAM_CG	Instance Of VSD Parameter Add On Instruction (1-Port Device)
VSD_Parameters		VSD_MSG_UDT_CG	VFX/FDU 2.0 UDT Instance Of Acyclic (Explicit) Exchanges Parameter Data (1-Port Device)
IPAddress		STRING	Demo VSD Acyclic (Explicit) Message IP Address Set 192.168.1.12 (2-Port Device)
OnePortIPAddress		STRING	Demo VSD Acyclic (Explicit) Message IP Address Set 192.168.1.11 (1-Port Device)
VSD_Basic_AOI		VSD_CLASS_BASIC_CG	Instance Of VSD Basic Process Data Add On Instruction (2-Port Device)
VSD_Basic_ProcessData		VSD_BASIC_UDT_CG	VFX/FDU 2.0 UDT Instance Of Cyclic (Implicit) Exchanges Basic Process Data (2-Port Device)
VSD_Message_Read		MESSAGE	VFX/FDU 2.0 DRIVE PARAMETER READ (Anybus) (2-Port Device)
VSD_Message_Write		MESSAGE	VFX/FDU 2.0 DRIVE PARAMETER WRITE (Anybus) (2-Port Device)
VSD_Param_AOI		VSD_CLASS_PARAM_CG	Instance Of VSD Parameter Add On Instruction (2-Port Device)
VSD_Parameters		VSD_MSG_UDT_CG	VFX/FDU 2.0 UDT Instance Of Acyclic (Explicit) Exchanges Parameter Data (2-Port Device)



## Quick Start Guide

Message data type instance configuration in this example

### Description of the User-defined Data Type VSD\_MSG\_UDT\_CG

Name	Data Type	Description
MSG_PathIPAddress	STRING	Ethernet IP Address To The Device
MSG_CommsPath	STRING	Communication Path For Message Instance (Constant)
STRING_Empty	STRING	Empty Char (For internal Use)
MSG_GetValues	BOOL	Get Parameter Values (Enable)
MSG_SetValues	BOOL	Set Parameter Values (Enable)
MSG_SetPath	BOOL	Set Communication Path (Enable)
MSG_Write_Data	DINT	Parameter Value To Object
MSG_Read_Data	DINT	Parameter Value From Object
MSG_Read_AttributeID	INT	Message Object Attribute ID Read Set
MSG_Read_ClassID	INT	Message Object Class ID Read Set
MSG_Read_InstanceID	DINT	Message Object Instance ID Read Set
MSG_ReadData_Len	INT	Message Length Of the Read Data
MSG_ReadError	BOOL	Message Read Error
MSG_ReadErrorCode	INT	Message Read Error Code *
MSG_ReadDone	BOOL	Message Read Done
MSG_Write_AttributeID	INT	Message Object Attribute ID Write Set
MSG_Write_ClassID	INT	Message Object Class ID Write Set
MSG_Write_InstanceID	DINT	Message Object Instance ID Write Set
MSG_SetWrite_Len	BOOL	Message Write Data Byte Length FALSE (0) =2 bytes, TRUE (1) = 4 Bytes
MSG_WriteError	BOOL	Message Write Error
MSG_WriteErrorCode	INT	Message Write Error Code *
MSG_WriteDone	BOOL	Message Write Done

\* For more information about the CIP error codes, Use the Logix Designer Online Help.



# Quick Start Guide

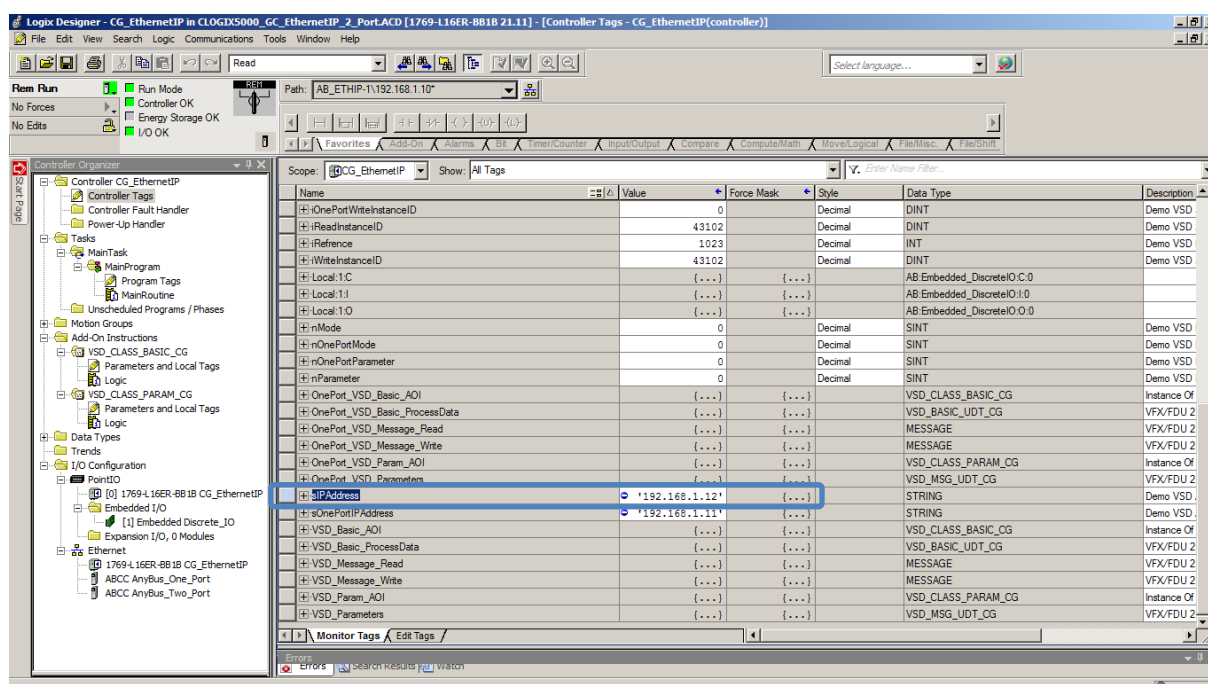
## Description of the properties for VSD\_CLASS\_PARAM.CG

Name	Usage	Data Type	Description
EnableIn	Input	BOOL	Enable Input - System Defined Parameter
EnableOut	Output	BOOL	Enable Output - System Defined Parameter
TAG	InOut	VSD_MSG_UDT.CG	CG Drives & Automation VSD Acyclic Message Data Type Configuration
MSG_READ	InOut	MESSAGE	Message Read VSD
MSG_WRITE	InOut	MESSAGE	Message Write VSD
MSG_RD_ER	Output	BOOL	Message Read Error
MSG_RD_ERR	Output	INT	Message Read Error Code
MSG_RD_LEN	Output	INT	Message Length Of Read Data
MSG_RD_DONE	Output	BOOL	Message Read Done
MSG_RD_DATA	Output	DINT	Message Read Data From Object
MSG_WR_ER	Output	BOOL	Message Write Error
MSG_WR_ERR	Output	INT	Message Write Error Code *
MSG_WR_DONE	Output	BOOL	Message Write Done
MSG_WR_DATA	Output	DINT	Message Write Data To Object

\* For more information about the CIP error codes, Use the Logix Designer Online Help.

## IP Address assignment to VSD\_Param\_AOI

IP address need to be set to the Add-On Object instance before accessing the ABCC-EIP device from the object. In this example there is a test variable declared in *Controller Tags* list is assigned as a constant (*sIPAddress*). Open the *Controller Tags* list and set the IP address for the actual ABCC-EIP device in the *Monitor Tags* dialog.





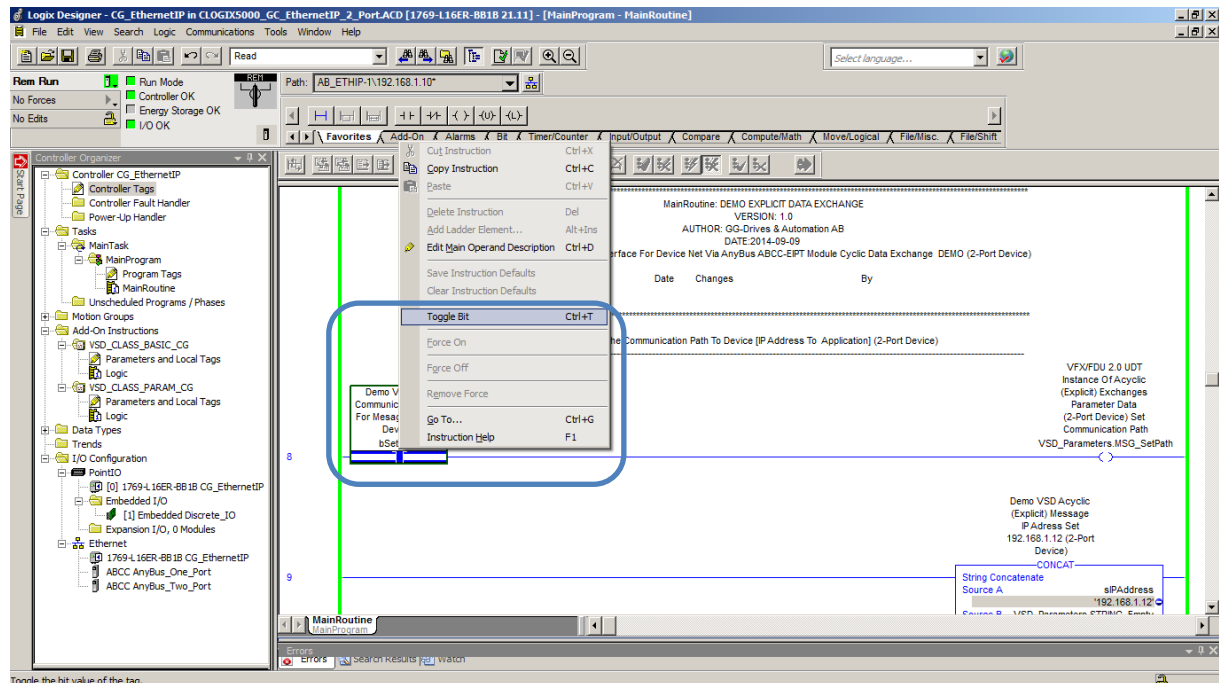


# Quick Start Guide

## Enable the communication path to VSD\_Param\_AOI

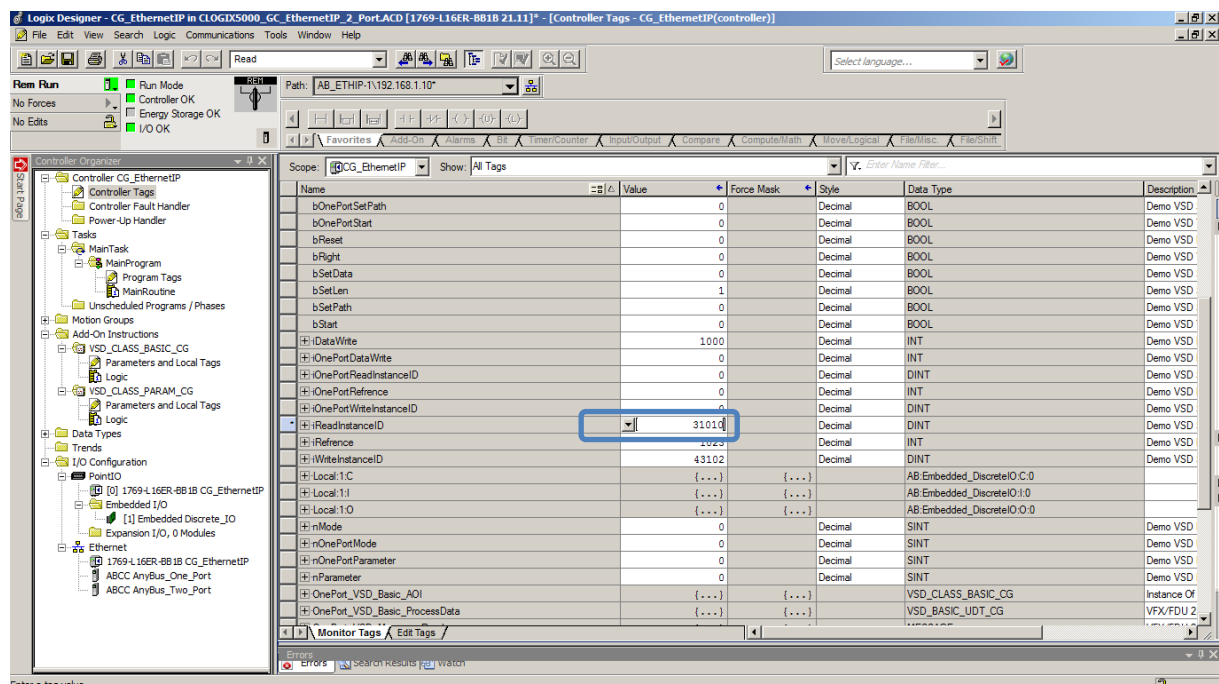
After manually setting the IP address, its need to be enabled to the communication path to the property on the Add-On object. The communication path property is the reference to the ABCC device that is accessed from the Add-On object instance (Message instruction).

Scroll down and toggle the *bSetPath* from the *MainRoutine* monitor to set the communication Path.



## Monitor VSD\_Param\_AOI (VSD\_CLASS\_PARAM\_CG)

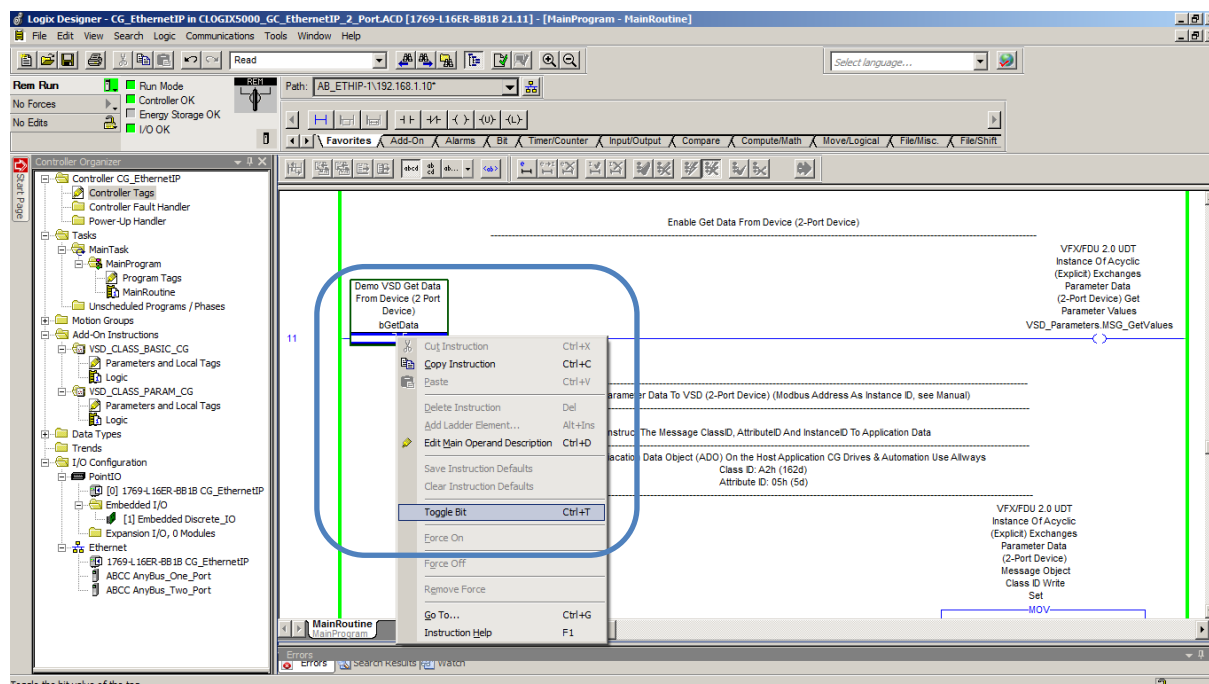
To read acyclic data (Explicit message) scroll up modify the input property to the Add-On object modify the read instance ID, *right click* on *iReadInstanceID* test variable and select *Monitor iReadInstanceID*. Set the variable to the modbus number *31010* which corresponds to the read the parameter [719] DC-voltage of the drive. Close the Monitor Tags dialog to continue to *Main Routine* monitor.



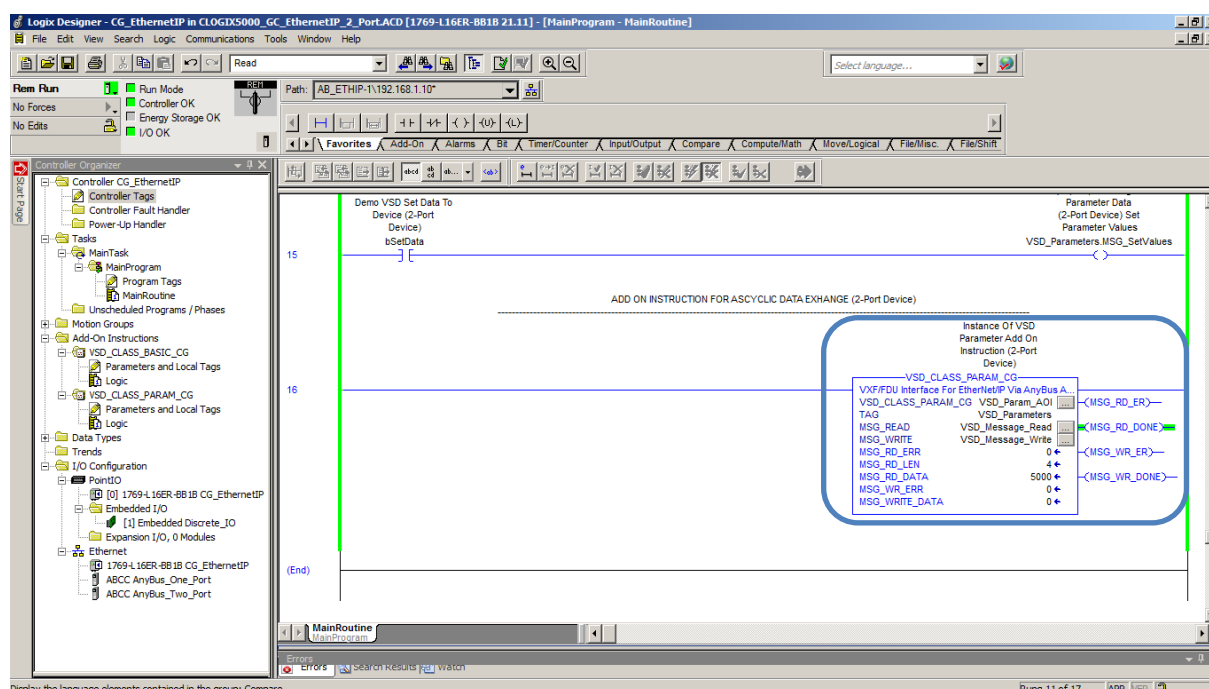


# Quick Start Guide

Select the test variable e.g. *bGetData* (message get values enable bit) and *right click* to toggle the bit to TRUE.



Scroll down to the Add-On Object Instance to monitor the value of the DC-voltage it is displayed in property *MSG\_RD\_DATA* in this case the value is 5000 (500.0 VDC). Observe that the *MSG\_READ\_DONE* is true and the *MSG\_RD\_LEN* indicates that 4 Bytes is read in message.



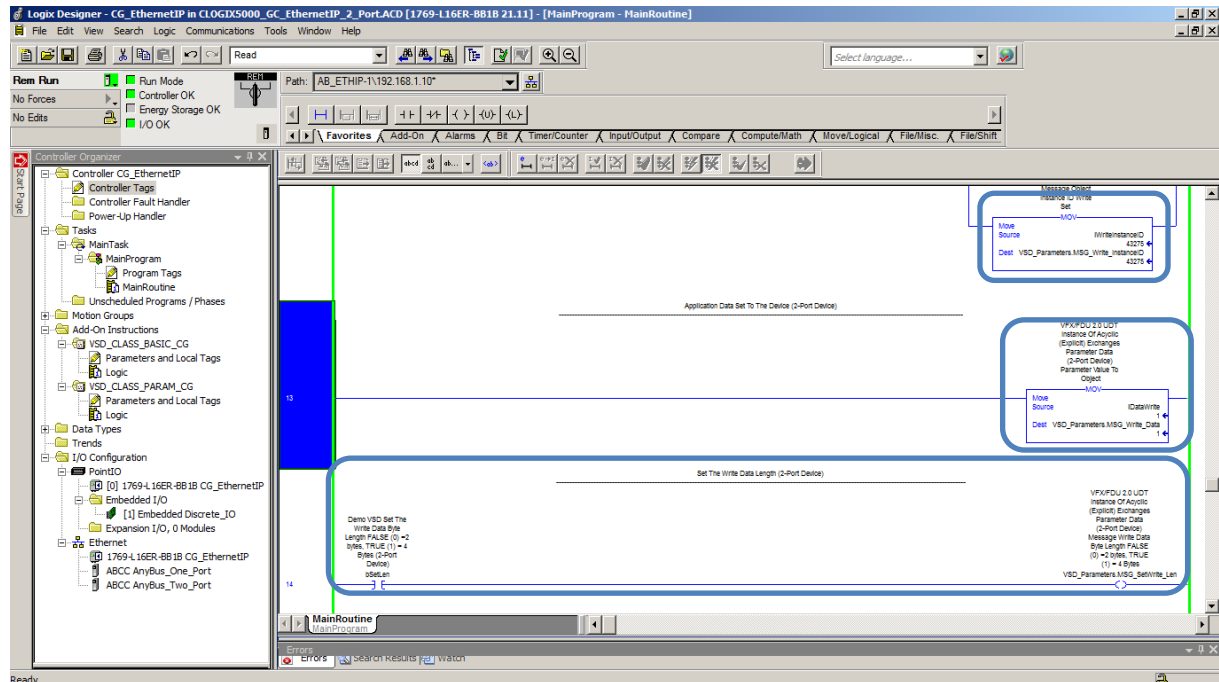




## Quick Start Guide

Write acyclic data (Explicit) scroll to modify the input property write Instance ID via the variable *iWriteInstanceID*. In this example the property is set to the modbus number 43275, VSD menu [553] function output relay 3. The Input property “.MSG\_SetWrite\_Len” in this example we use the variable *bSetLen* as test variable, this property need to be set to FALSE when writing 2 byte data and TRUE when writing to 4 byte data.

Scroll up and set the *iDataWrite* test variable value to 1, this variable is associated with the input property “.MSG\_Write\_Data”.



To trigger write message of the acyclic data, change the input property “.MSG\_SetValues” by set the variable *bSetData* to TRUE. The property *MSG\_WR\_DONE* will be activated. The relay function [553] output relay 3 in the VSD should indicate *On* status.

