



ETHERNET IP SET UP GUIDE

R3

ROCKWELL AUTOMATION – LOGIC5000

RICCARDO PICCININI

SOMMARIO

Overview and process data	0
Overview.....	0
CIP Objects	0
COM object – 0x64	1
PAR object – 0x65.....	1
MISC object – 0x66.....	2
CIA402 object – 0x67.....	2
Process data.....	4
Layout Default	4
Layout A.....	5
Layout B.....	6
Layout C.....	7
Layout D	8
Application example	10
TOOLS	10
Overview.....	10
DBS basic application.....	10
Configure Controller Address.....	10
INSTALL EDS.....	12
ADD DBS SLAVE to a project.....	14
AOI IMPORT	17
AOI Explanation	18
Writing parameters with async messagging.....	22
ASYNC MESSAGING	22



OVERVIEW AND PROCESS DATA

OVERVIEW

This document outlines the conformance of the Ethernet/IP and CIP stack implemented in the servodrive; the following documents apply except otherwise stated:

- CIP Network library – Volume 1 (edition 3.3, november 2007)
- CIP Network library – Volume 2; Ethernet/IP adaptation of CIP (edition 1.4, november 2007)
- CiA DS402 v3.0.1.15

Furthermore, the CiA DS402 adaptation to CIP are described in the relevant section.

CIP OBJECTS

The servodrive implements the following objects of the CIP object model:

- CIP common objects
 - Identity object – 01h
 - Assembly object – 04h
 - TCP object – F5h
 - Ethernet object – F6h;
- Manufacturer specific objects
 - COM object – 0x64
 - PAR object – 0x65
 - MISC object – 0x66
 - CiA402 objects – 0x67

Furthermore, the following ASSEMBLIES contains process data:

- O>>T: Assembly 0x64
- T>>O: Assembly 0x65
- CFG: Assembly 0x66

COM OBJECT – 0X64

Instance	Attribute ID	Name	Data Type	Unit meas.	Access
0x0	0x01	Revision	UINT		RO
0x1	0x01	Software edition	UINT		RO
	0x02	Software CRC	UDINT		RO
	0x10	Save parameters NOTE: to save parameter, value 0x65766173 should be written	UDINT		RO

PAR OBJECT – 0X65

Instance	Attribute ID	Name	Data Type	Unit meas.	Access
0x0	0x01	Revision	UINT		RO
0x1	0x00	P000	UINT	(*1)	RW
	0x01	P001	UINT	(*1)	RW
	...				
	0xFF	P255	UINT	(*1)	RW
0x2	0x00	P256	UINT	(*1)	RW
	0x01	P257	UINT	(*1)	RW
	...				
	0xFF	P511	UINT	(*1)	RW

(*1): please refer to the parameter list for actual meaning of the parameters.

Special parameters

- Appl>Fieldbus>CIP>P170-IpAddr

This parameter is the ip address for the CIP module; since IP address is 32bit wide and normal parameters are 16bit wide, two parameters are actually used:

- Parameter n.170: 16 most-significant word of IP address
- Parameter n.171: 16 least significant word of IP address

- Appl>Fieldbus>CIP>P172-IpMask

IP net mask for CIP module, analogous to IP address:

- Parameter n.172: 16 most-significant word of IP net-mask
- Parameter n.173: 16 least significant word of IP net-mask

- Appl>Fieldbus>CIP>P174-IpGateway

Ip gateway address; used only whenever IP routing is used (not used if master/slave are in the same subnet)

- Parameter n.174: 16 most-significant word of IP gateway
- Parameter n.175: 16 least significant word of IP gateway

MISC OBJECT – 0X66

Instance	Attribute ID	Name	Data Type	Unit meas.	Access
0x0	0x01	Revision	UINT		RO
0x30 (48)	0x00	Analog input 0	UINT		RO
	0x01	Digital inputs	UINT		RO
	0x02	Digital outputs	UINT		RW
	0x03	Heatsink temperature	UINT	0.1 °C	RO
	0x05	Board temperature	UINT	0.1 °C	RO
	0x08	Actual Alarm	UINT	N/A	RO
	0x10	Dclink voltage	UINT	0.1 V	RO
	0x11	Current actual value	UINT	mA	RO
	0x12	Current limit for PP/PV/PT	UINT	mA	RO

CIA402 OBJECT – 0X67

Instance	Attr. ID	Name	Data Type	Unit meas.	Access
0x0	0x01	Revision	UINT	-	RO
0x60 (96)	0x40	Control word	UINT	-	RW
	0x41	Status word	UINT	-	RO
	0x5A	Quick-stop option code	INT	-	RW
	0x5C	Disable operation option code	INT	-	RW
	0x5E	Fault reaction code	INT	-	RW
	0x60	Mode of operation	SINT	-	RW
	0x64	Position actual value	DINT	pulses	RO
	0x65	Following error window	DINT	pulses	RW
	0x66	Following error time	INT	ms	RW
	0x67	Position window	DINT	pulses	RW
	0x68	Position window time	INT	ms	RW
	0x6C	Velocity actual value	DINT	rpm	RO
	0x6D	Velocity window	DINT	rpm	RW
	0x6E	Velocity window time	INT	ms	RW
	0x6F	Velocity threshold	DINT	rpm	RW
	0x70	Velocity threshold time	INT	ms	RW
	0x71	Target torque	INT	thousand- th of rated	RW

0x72	Max torque	INT	thousand-th of rated	RW
0x73	Max current	INT	thousand-th of rated	RW
0x75	Motor rated current	DINT	milliA	RO
0x76	Motor rated torque	DINT	milliNm	RO
0x77	Torque actual value	INT	thousand-th of rated	RO
0x78	Current actual value	INT	thousand-th of rated	RO
0x7A	Target position	DINT	pulses	RW
0x7B	Position range index1: min limit index2: max limit	ARRAY[1..2] OF DINT	pulses	RW
0x7C	Home offset	DINT	pulses	RW
0x7D	Position limits index1: min limit index2: max limit	ARRAY[1..2] OF DINT	pulses	RW
0x7E	Polarity	USINT	-	RW
0x80	Max motor speed	DINT	rpm	RO
0x81	Profile velocity	DINT	rpm	RW
0x83	Profile acceleration	DINT	rpm/s	RW
0x84	Profile deceleration	DINT	rpm/s	RW
0x85	Quick stop deceleration	DINT	rpm/s	RW
0x87	Torque slope	INT	thousandth of rated / s	RW
0x98	Homing method	SINT		RW
0x99	Homing speed index1: switch speed index2: index speed	ARRAY[1..2] OF DINT	rpm	RW
0x9A	Homing acceleration	DINT	rpm/s	RW
0xF2	Position option Code	INT	DS402	RW
0xFF	Target velocity	DINT	rpm	RW

PROCESS DATA

*** DBS55 firmware referenced by this document: ≥v4.044 ***

There are four possible configuration of the Ethernet IP process data.

- **Default:** the standard process data implementing the profile velocity/torque/position/homing
- **A:** Standard process data with the possibility to read Digital and Analog inputs and command via fieldbus the Digital Out. The latter needs the parameter P015 on 10-Fieldbus to receive commands.
- **B:** A process data with the Cia 402 Touch probe functionality
- **C:** A process data with our manufacturer custom Touch Probe implementation
- **D:** A process data with acceleration three axis rms values

You must select the corresponding device description and set the parameter **P176** to the desired value.

LAYOUT DEFAULT

The configuration for process data is the following:

- Assembly 0x64 (Originator to Target, **DBS/FC inputs - PLC outputs**)
Assembly length : 50bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see Dsp402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	x 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	x 0.1% Trq
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	ProfilePosition see Dsp402
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota see Dsp402

- Assembly 0x65 (Target to Originator, **DBS/FC outputs - PLC inputs**)
Assembly length : 20bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	see Dsp402
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units

0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	x 0.1% Trq
0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	x 0.1V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	see Dsp402
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual

LAYOUT A

The configuration for process data is the following:

- Assembly 0x64 (Originator to Target, **DBS/FC inputs - PLC outputs**)
Assembly length : 50bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see Dsp402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	x 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	x 0.1% Trq
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	ProfilePosition see Dsp402
0x0F	U8	Padding Byte					
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota see Dsp402
0x29	U8	Padding Byte					
0x2A	U16	Digital output	0x67	0x30	0x02	0	bit0: DOUT1 bit1..15: not used <i>Note: P015=10 required for fieldbus operation</i>

- Assembly 0x65 (Target to Originator, **DBS/FC outputs - PLC inputs**)
Assembly length : 30bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	see Dsp402
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units
0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	x 0.1% Trq

0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	x 0.1V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	see Dsp402
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual
0x12	S16	Analog Input	0x67	0x30	0x00	+0	-32768 ... 32767
0x14	U16	Digital Input	0x67	0x30	0x01	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used

LAYOUT B

The configuration for process data is the following:

- Assembly 0x64 (Originator to Target, **DBS/FC inputs - PLC outputs**)
Assembly length : 50bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see Dsp402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	x 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	x 0.1% Trq
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	ProfilePosition see Dsp402
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota see Dsp402
0x29	U8	Padding Byte					
0x2A	U16	Digital output	0x67	0x30	0x02	0	bit0: DOUT1 bit1..15: not used <i>Note: P015=10 required for fieldbus operation</i>
0x2C	U16	TouchProbe function	0x67	0x60	0xB8	0	see Cia402

- Assembly 0x65 (Target to Originator, **DBS/FC outputs - PLC inputs**)
Assembly length : 40bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	see Dsp402
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units
0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	x 0.1% Trq
0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	x 0.1V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	see Dsp402
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual
0x12	S16	Analog Input	0x67	0x30	0x00	+0	-32768 ... 32767
0x14	U16	Digital Input	0x67	0x30	0x01	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used
0x16	U16	TouchProbe status	0x67	0x60	0xB8	0	see Cia402
0x18	S32	TouchProbe rising edge position	0x67	0x60	0xBA	+0	position units
0x1C	S32	TouchProbe falling edge position	0x67	0x60	0xBB	+0	position units

LAYOUT C

The configuration for process data is the following:

- Assembly 0x64 (Originator to Target, **DBS/FC inputs - PLC outputs**)
Assembly length : 60bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see Dsp402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	x 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	x 0.1% Trq
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	ProfilePosition see Dsp402
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units

0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota see Dsp402
0x29	U8	Padding Byte					
0x2A	U16	Digital output	0x67	0x30	0x02	0	bit0: DOUT1 bit1..15: not used <i>Note: P015=10 required for fieldbus operation</i>
0x2C	S32	TouchProbe positioning offset	0x67	0x30	0x20	+0	Position units
0x30	U32	Touchprobe position. modulo	0x67	0x30	0x21	4096	Position units

- Assembly 0x65 (Target to Originator, **DBS/FC outputs - PLC inputs**)
Assembly length : 30bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	see Dsp402
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units
0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	x 0.1% Trq
0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	x 0.1V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	see Dsp402
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual
0x12	S16	Analog Input	0x67	0x30	0x00	+0	-32768 ... 32767
0x14	U16	Digital Input	0x67	0x30	0x01	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used

LAYOUT D

Assembly 0x64 (Originator to Target, **DBS/FC inputs - PLC outputs**)

1. Assembly length : 50bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see Dsp402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	x 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	x 0.1% Trq

0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	ProfilePosition see Dsp402
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota see Dsp402
0x29	U8	Padding Byte					
0x2A	U16	Digital output	0x67	0x30	0x02	0	bit0: DOUT1 bit1..15: not used <i>Note: P015=10 required for fieldbus operation</i>

2. Assembly 0x65 (Target to Originator, DBS/FC outputs - PLC inputs)

Assembly length : 30bytes

Offset	Type	Attribute	Class	Instance	Attribute	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	see Dsp402
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units
0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	x 0.1% Trq
0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	x 0.1V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	see Dsp402
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual
0x12	S16	Analog Input	0x67	0x30	0x00	+0	-32768 ... 32767
0x14	U16	Digital Input	0x67	0x30	0x01	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used
0x16	U16	Acceleration rms x (milli-G)	0x67	0x30	0x31	-	
0x18	U16	Acceleration rms y (milli-G)	0x67	0x30	0x32	-	
0x1A	U16	Acceleration rms z (milli-G)	0x67	0x30	0x33	-	

APPLICATION EXAMPLE

TOOLS

>>[Program Example \(no AOI\)](#)<<

>>[DBS/FC BSI Interface Software](#)<<

>>[AOI Download](#)<<

OVERVIEW

This document shows the creation of a DBS55 smart motor application using Ethernet/IP communication interface; a Rockwell CompactLogix controller is configured and an application is written into the Studio 5000 development environment.

References

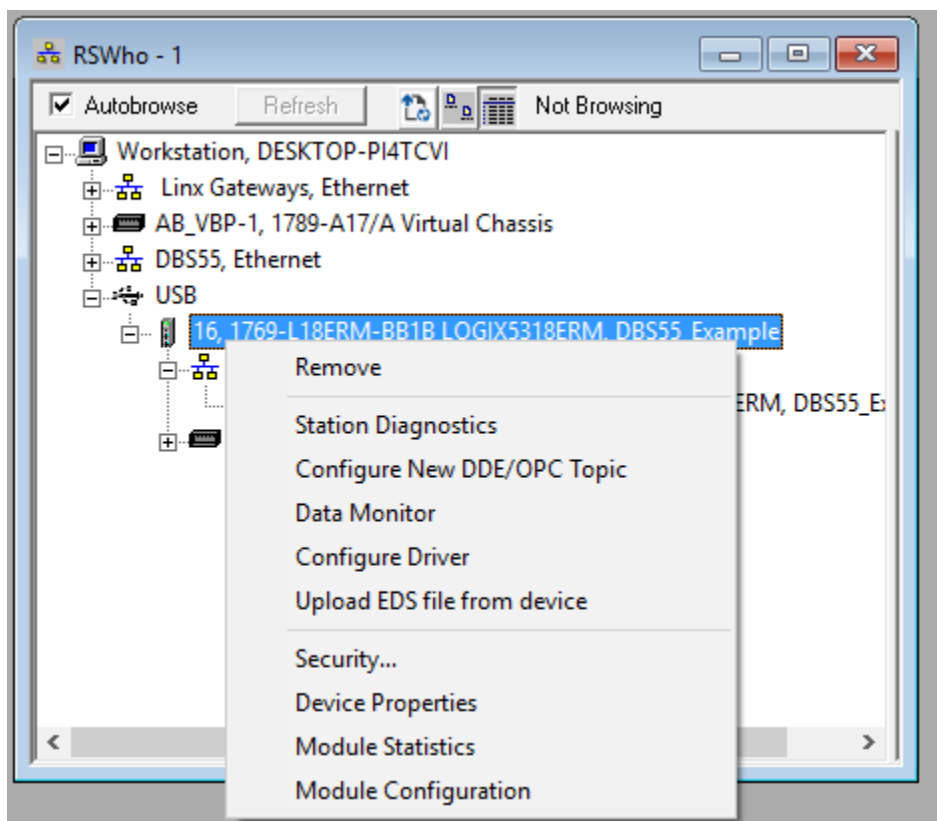
- CompactLogix L18ERM
- DBS/FC Firmware version ≥ 4.044
- Studio 5000, version 31.01.00

DBS BASIC APPLICATION

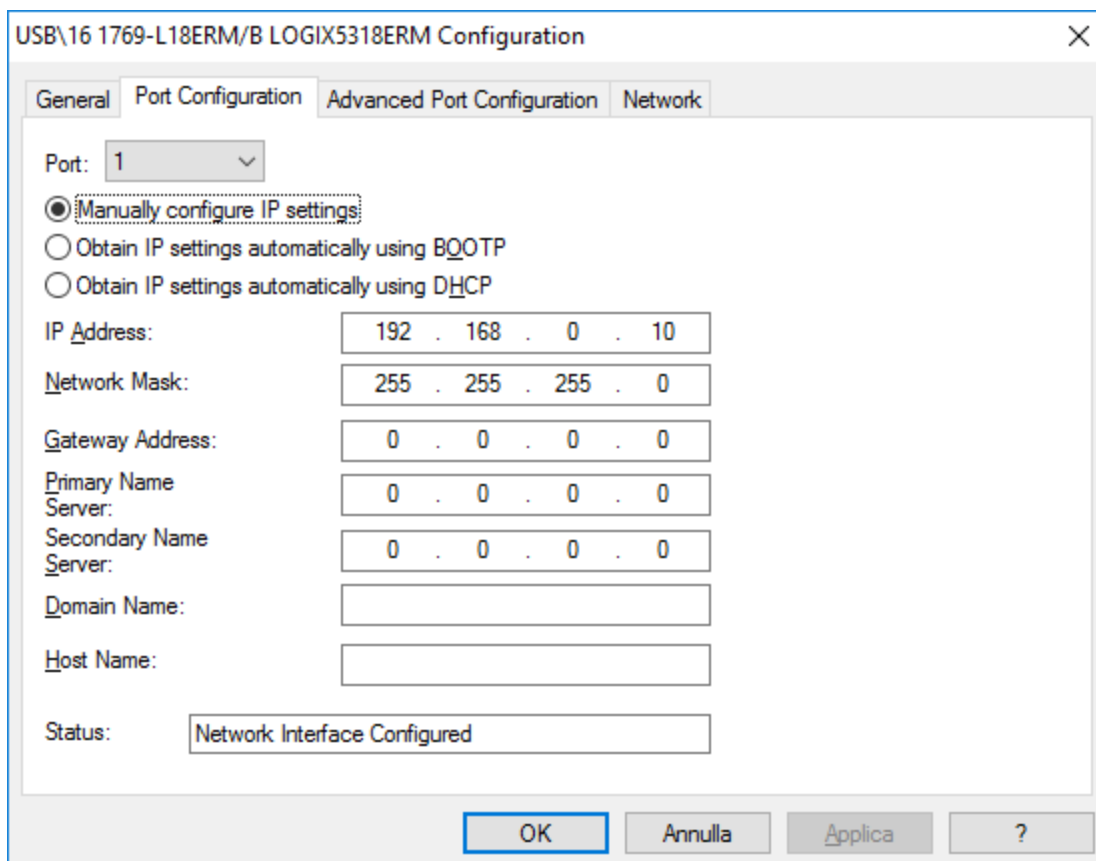


CONFIGURE CONTROLLER ADDRESS

- To set controller address, start RSLinx
- Select Communication >> RSWho and open controller node (in the example the controller is connected via USB); then right click and select Module configuration:



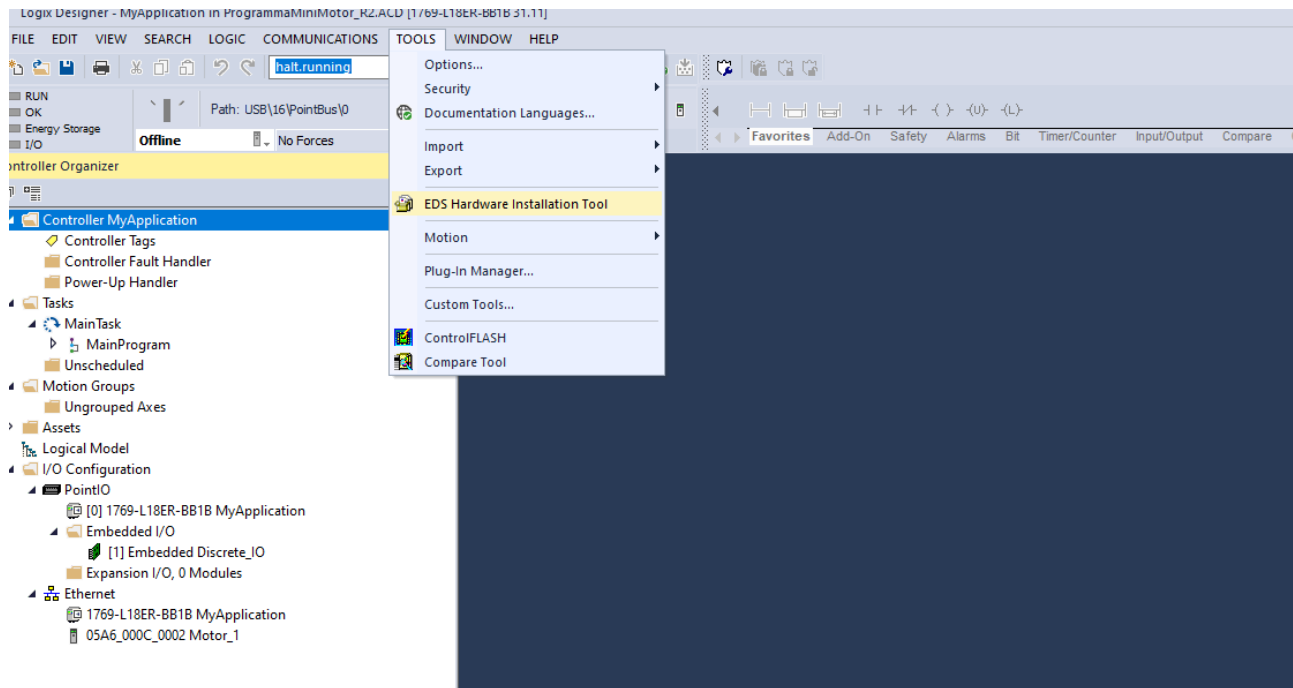
- In the configuration dialog, select “Port configuration” (1), then “Manually configure IP settings” (2) and insert address and mask (3); in the example 192.168.0.10 is set



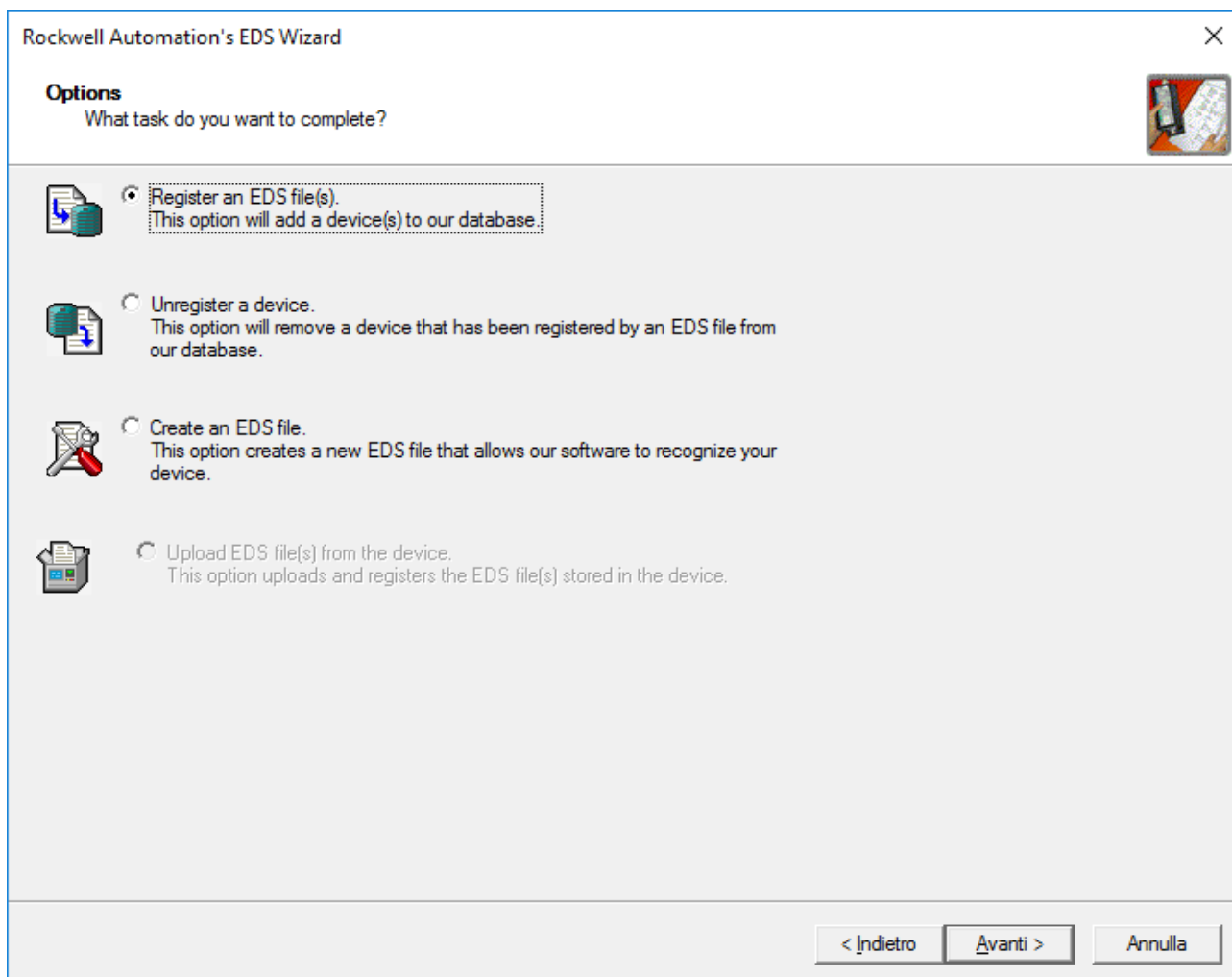
- Confirm with OK.

INSTALL EDS

- In Studio 5000 click on Tools >> EDS Hardware Installation Tool



- Register and EDS file



- Browse
- Open the BSI software revision\Device Desc\EthernetIP folder and select the EIP Layout you desire. For this example we will select "DBS55_CIP_dfl.eds". **Install only the one you want to use.**

Condividi Visualizza

MM-DBS_v4.041 > DeviceDesc > EthernetIP

Nome	Ultima modifica	Tipo	Dimensione
DBS55_CIP.eds	21/11/2022 16:31	File EDS	6 KB
DBS55_CIP_dfl.eds	21/11/2022 16:31	File EDS	21 KB
DBS55_CIP_ext-a.eds	21/11/2022 16:31	File EDS	23 KB
DBS55_CIP_ext-b.eds	21/11/2022 16:31	File EDS	26 KB
DBS55_CIP_ext-c.eds	21/11/2022 16:31	File EDS	27 KB
DBS55_CIP_ext-d.eds	21/11/2022 16:31	File EDS	30 KB
Readme.txt	21/11/2022 16:31	Documento di testo	1 KB

Rockwell Automation's EDS Wizard

**Registration**

Electronic Data Sheet file(s) will be added to your system for use in Rockwell Automation applications.



☒ Register a single file

☐ Register a directory of EDS files

☐ Look in subfolders

Named:

Desktop\MM-DBS_v4.037\DeviceDesc\EthernetIP\DBS55_CIP_dfl.eds

Browse...



* If there is an icon file (.ico) with the same name as the file(s) you are registering then this image will be associated with the device.

To perform an installation test on the file(s), click Next

< Indietro

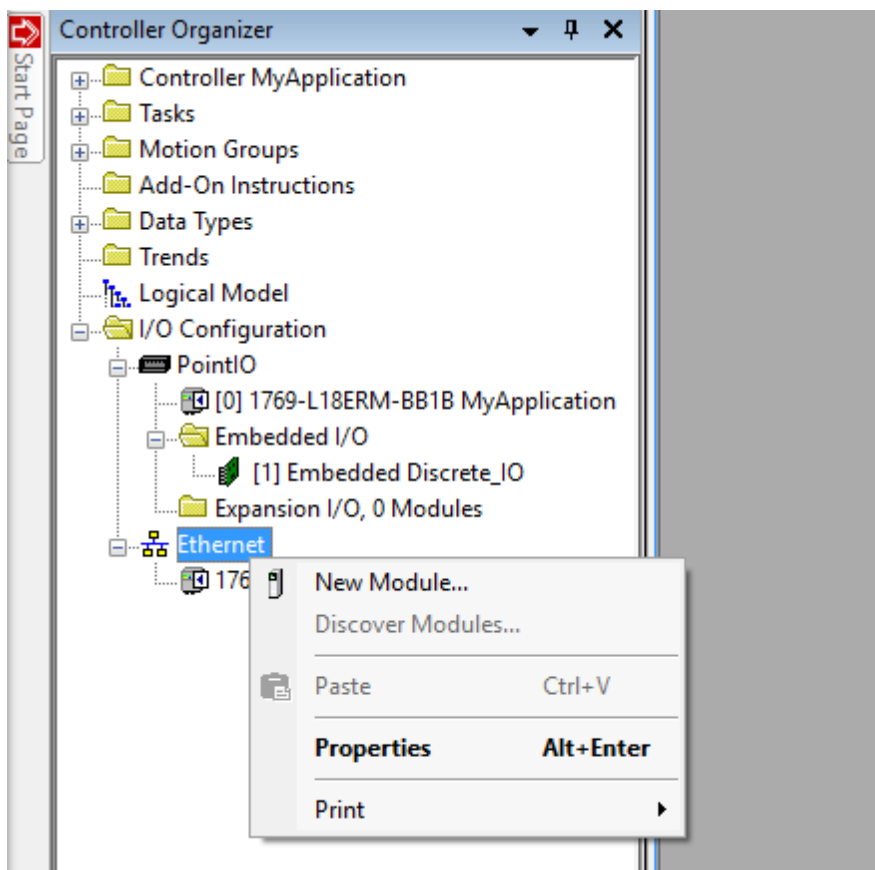
Avanti >

Annulla

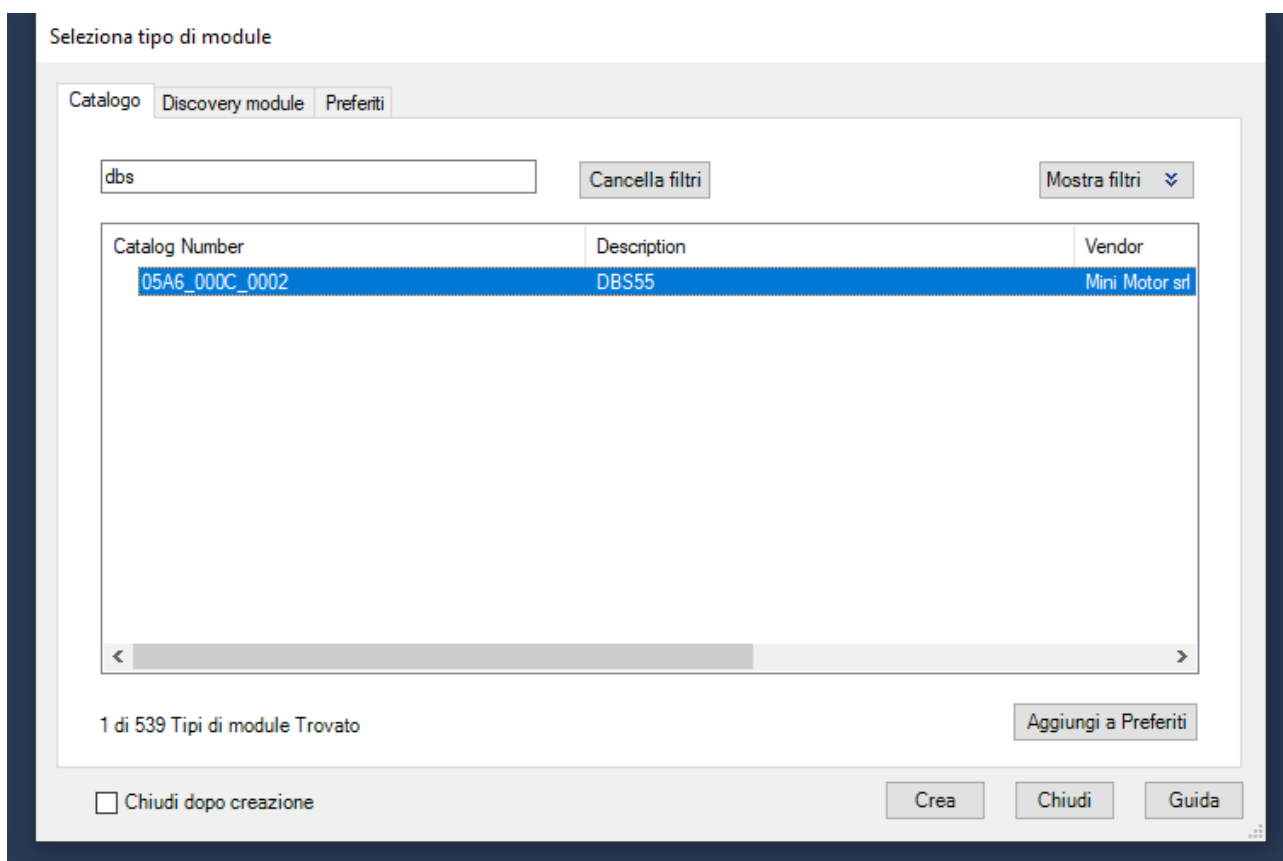
- Next ->, Next _>, Next- >; click on Finish to complete insertion.

ADD DBS SLAVE TO A PROJECT

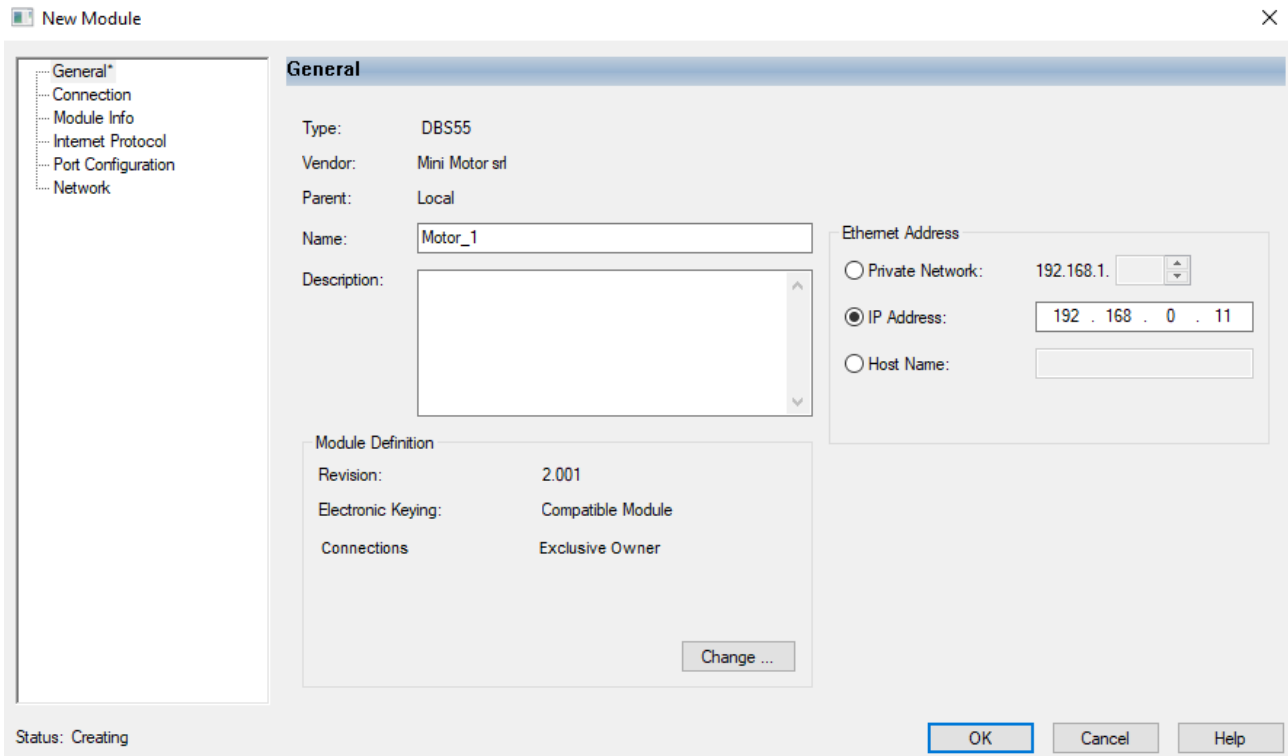
- Right click on the "I/O Configuration > Ethernet" and select "New Module"



- Type dbs in the filter and Select “05A6_000C_0002 – DBS55”, then click on Create



- In the “new module” window enter the device name the IP address if the device and finally click on Change.



New Module

General*

Type: DBS55
 Vendor: Mini Motor srl
 Parent: Local
 Name: Motor_1
 Description:

Ethernet Address

☐ Private Network: 192.168.1.
☒ IP Address: 192 . 168 . 0 . 11
☐ Host Name:

Module Definition

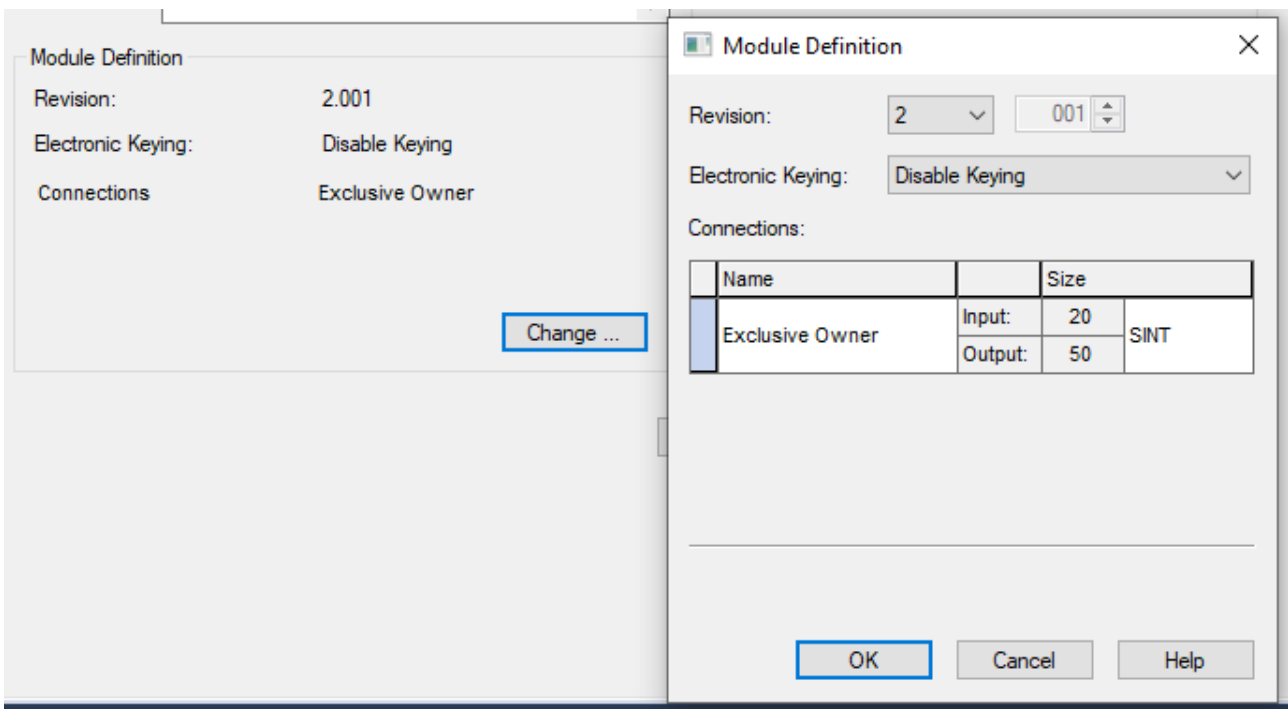
Revision: 2.001
 Electronic Keying: Compatible Module
 Connections: Exclusive Owner

Change ...

Status: Creating

OK Cancel Help

Please note that the IP address of device should be in the same subnet of the master IP address; the IP address of the slave is configured by the BSI configuration software (see DBS/FC manual).



Module Definition

Revision: 2 001
 Electronic Keying: Disable Keying
 Connections: Exclusive Owner

Change ...

Connections:

Name	Size
Exclusive Owner	Input: 20 Output: 50

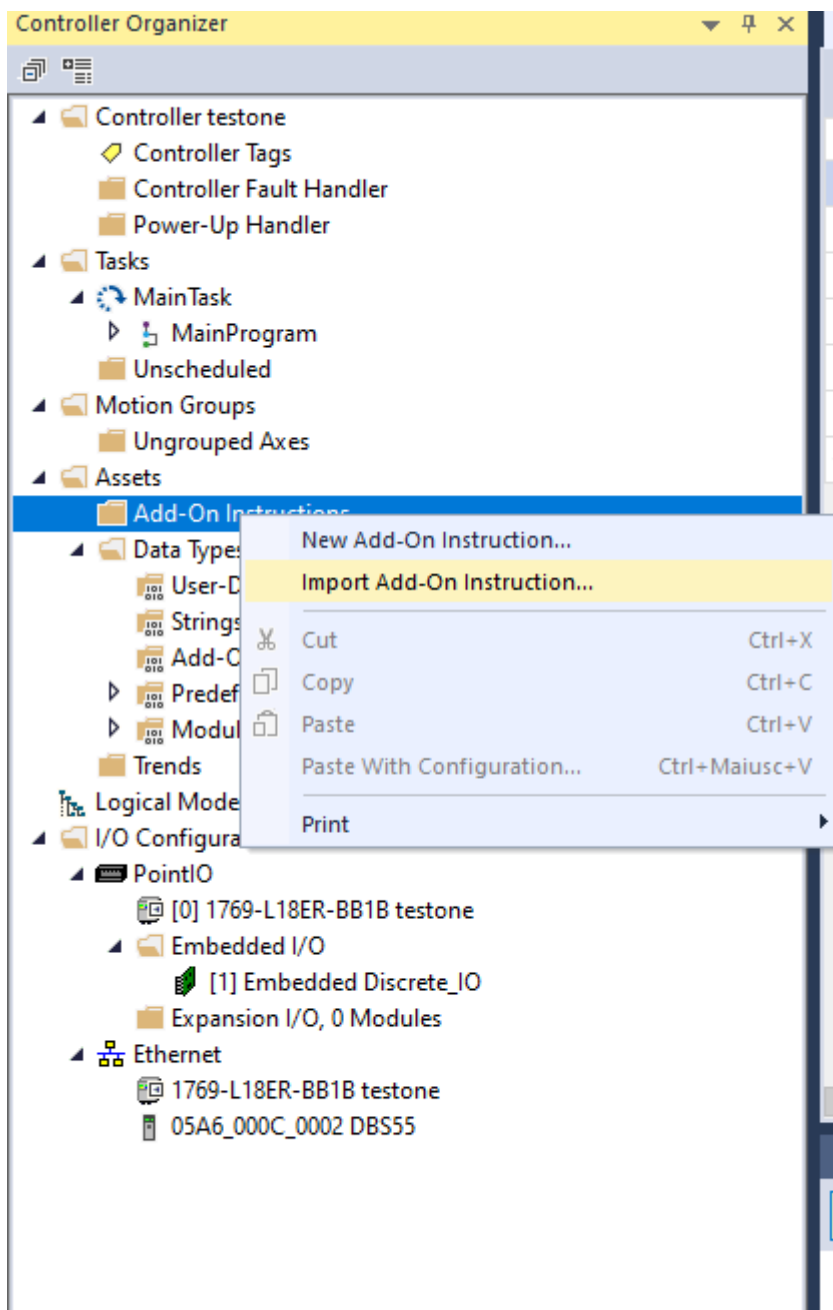
SINT

OK Cancel Help

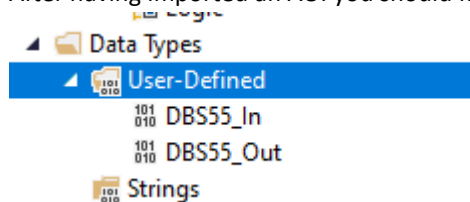
- In the module definition window, in the Electronic keying select “**Disable keying**”
- Confirm module definition dialog and finally click on “Create” and then close dialog

AOI IMPORT

- Import the DBS AOI you need for your project. Go to Assets -> Add On Instructions -> Right Click -> select which AOI to import.



- After having imported an AOI you should find two User Defined Data Types.



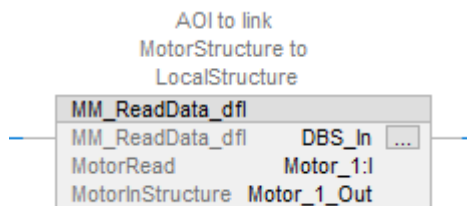
- Create for each motor a variable of both types

Controller Tags - testone(controller) x						
Scope: testone		Show: All Tags				
Name	Alias For	Base Tag	Data Type	Description		
Local:1:O			AB:Embedded_DiscretelO:O:0			R
Local:1:I			AB:Embedded_DiscretelO:I:0			R
Local:1:C			AB:Embedded_DiscretelO:C:0			R
DBS1_In			DBS55_In	PdoRX		R
DBS1_out			DBS55_Out	PdoTX		R

AOI EXPLANATION

There are 7 AOI that can be used for DBS and FC systems. We'll Look at them here.

MM_READDATA_DFL



This instruction should be the first and its job is to take the motor assembly data and assign it to a DBS55_Out Structure. Each Minimotor Integrated Drive system creates a **DB13776:I:0** Data Type that must be assigned to a specific MM_ReadData_Dfl AOI. Then each data is passed to a specific DBS55_Out variable. This structure will be used in all operative blocks

LocalStructure

Motor_ReadData

Motor_ReadData DBS_In

MotorRead Motor_1:I

MotorInStructure

Enter Name Filter...

Show: All Tags

Name	Data Type	Usage	Description
Motor_1:I	_05A6:000C_0002_DB13776:I:0	<controller>	

MotorInStructure

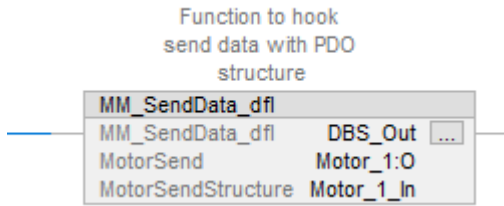
Motor_1_Out

Enter Name Filter...

Show: All Tags

Name	Data Type	Usage	Description
Motor_1_Out	DBS55_Out	<controller>	PdoTX

MM_SENDDATA_DFL

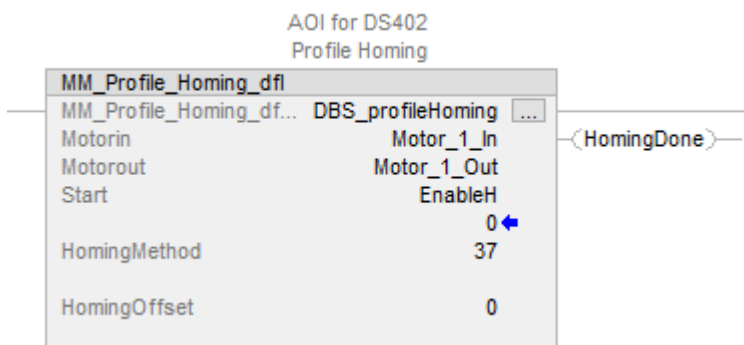


This instruction should be the last and its job is to move data from a DBS55_In data type to the motor data inlet. Each Minimotor Integrated Drive system creates a **E7C21ABF:0:0** Data Type that must be assigned to a specific MM_SendData_Dfl AOI. In this way the data manipulated by the rest of the blocks will be passed from the DBS55_In to the motor.

Name	Data Type	Usage	Description
Motor_1:0	_05A6:000C_0002_E7C21ABF:0:0	<controller>	

Name	Data Type	Usage	Description
Motor_1_In	DBS55_In	<controller>	PdoRX

MM_PROFILE_HOMING_DFL



With the mm_profile_homing_Dfl you can home the Integrated Drive family following the DS402 homing procedure as explained in the main DBS/FC manual.

The MotorIn and Motorout are hooked to the DBS55_In and DBS55_Out data types of the specific motor while the Homing Method is the homing procedure selector and Homing Offset is the offset to add once the procedure is done.

A single bit signals the completion of the homing procedure. This procedure can be started with a Start BOOLEAN value.

MM_PROFILE_TORQUE

AOI to move DBS in
Profile Torque DS402

MM_Profile_Torque_dfl	
MM_Profile_Torque_df...	DBS_proftqr
motorin	Motor_1_In
Start	EnableT
	0
Target_Torque	1000
ProfileVelocity	1500
Acceleration	2000
Deceleration	2000

The motor can be used in DS402 Profile Torquemodes, useful when you want to have better torque control in application where speed is secondary.

The AOI must be hooked to a DBS55_In data types of the motor to send commands. A BOOL Start makes the motor turn.

You can set the maximum torque the motor can use with Target_Torque and constrain the trajectory with the acceleration and deceleration.

MM_PROFILE_VELOCITY_DFL

AOI to move DBS
motor in Profile
Velocity DS402

MM_Profile_Velocity_dfl	
MM_Profile_Velocity_...	DBS_profvel
motorin	Motor_1_In
Start	EnableV
	0
Target_Velocity	2000
Acceleration	1500
MaxTorque	1000
Deceleration	1500
PosLimitPos	2147483647
PoslimitNeg	-2147483648

The motor can be used in DS402 Profile Velocity mode, enabling the motor to reach and maintain a specific target speed.

The AOI must be hooked to a DBS55_In data type of the motor to send commands. A BOOL Start makes the motor turn.

You can set the target speed the motor can use with and constrain the trajectory with the other parameters. If the position limits are at their maximum value (2147483647 and -2147483648 respectively) then they are disabled. If they

are both at 0 the motor will not move.

MM_PROFILE_POSITION_DFL

AOI for
ProfilePosition
DS402

MM_Profile_Position_dfl	
MM_Profile_Position_...	DBS_prfpos ...
motorin	Motor_1_In
TorqueEnable	EnableP
	0 
Startpositioning	Doposition
	0 
TargetPosition	9000
ProfileVelocity	1500
MaxTorque	1000
Acceleration	2000
Deceleration	2000
PosLimitPos	2147483647
PosLimitNeg	-2147483648
ChangeImmediately	changeimmediatly
	0 

The motor can be commanded to reach specific positions using its integrated absolute multiturn encoder.

The AOI must be hooked to a DBS55_In data type of the motor to send commands

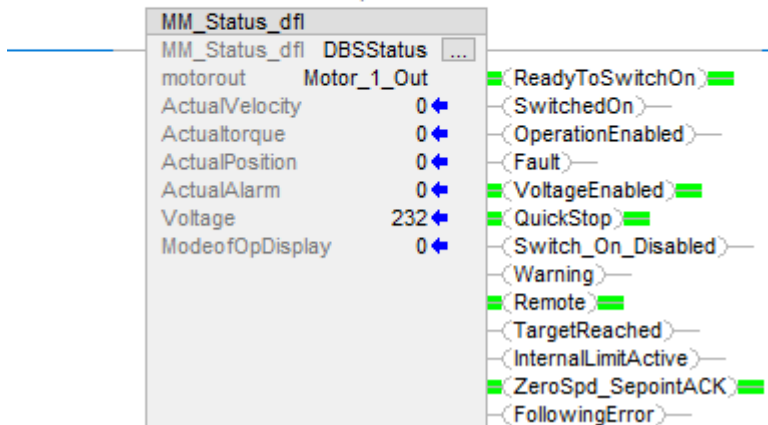
The AOI has two separate commands to start the motor. TorqueEnable makes the motor enabled in standstill, keeping the actual position. Startpositioning is a rising edge command that makes the motor start its positioning.

The trajectory can be constrained via the parameters shown (Velocity, Deceleration, Acceleration, Position limits and max torque).

The ChangeImmediately command is used to make the new position command immediately execute, discarding the current motion.

MM_STATUS_DFL

AOI block to read motor data and have a quick look at the status of the motor, via StatusWord bits and main operative parameters.

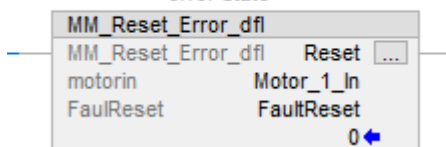


This block can be used to have a quick access to the motor status hooking a DBS55_Out structure to it.

It shows the motor speed, actual torque, actual position, actual alarm and voltage. It also shows the bits of the StatusWord that can be used to power internal logics or diagnostics. It gives the feedback for the Target Reached and other behaviour for the various operating modes.

MM_RESET_ERROR_DFL

AOI to reset motor error state



This AOI is used to reset the error status on the motor. It's hooked to a DBS55_In instance to send the command. The Fault status can be seen on the MM_Status_Dfl block.

WRITING PARAMETERS WITH ASYNC MESSAGING

More complex things are possible with the DBS/FC system.

ASYNC MESSAGING

The application layer for the DBS55 motor is very close to the CiA402 servodrive definition; the plc can interact with the motor CiA402 state machine using ControlWord and StatusWord.

Nevertheless, an important part of the CiA402 servodrive model is managed via asynchronous read/write of the CanOPEN object dictionary.

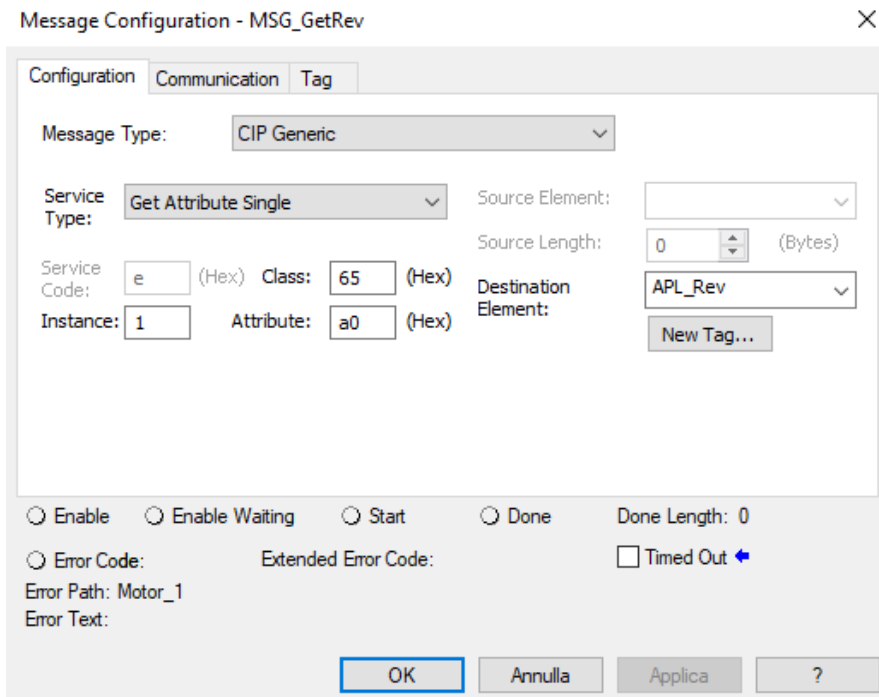
In the Ethernet/IP implementation, the asynchronous messaging is implemented as Class3 GetAttribute/SetAttribute services, where portions of the CanOpen object dictionary are exposed to the Ethernet/IP world via attributes.

There are 4 main Ethernet/IP manufacturer objects that expose the attribute which act as gateway to the CanOPEN world:

- COM object – object 0x64
This object exposes some basic information (e.g. Software release) and allows to save parameter to non volatile storage
- PAR object – object 0x65
This object allows changing via PLC the motor parameter; this allows for example changing dynamically the Kp of the speed loop if the load of the motor changes during a machine phase.
- MISC object – object 0x66
This object allows reading and writing hardware related informations (e.g. analog input value, heatsink temperature etc)
- CIA402 objects – object 0x67
This object allows reading and writing CiA402 objects: for example reading object 6081h.00 is translated to a GetAttribute on object 0x67, instance 0x60, attribute 0x81.

As an example we will read the BSi parameter P160, PosFactorNum, which is used to scale the motor position value.

Using the MSG block, the configuration will be



As seen in the [first chapter](#) of this guide, P160 is part of the Class 0x65, it is in the first instance and its value in hex is A0.

Remember to create a Destination element where to write down the value of interest.

Conversely, to write the same Parameter, the MSG must be set in Set Attribute Single

Message Configuration - MSG_GetRev



Configuration*
Communication
Tag

Message Type:
CIP Generic

Service Type:
Set Attribute Single
Source Element:
Posscaling

Service Code:
10
(
Hex
)
Class:
65
(
Hex
)
Source Length:
2
(
Bytes
)
Destination Element:

Instance:
1
Attribute:
A0
(
Hex
)
New Tag...

☐ Enable
☐ Enable Waiting
☐ Start
☐ Done
Done Length: 0

☐ Error Code:
Extended Error Code:
☐ Timed Out

Error Path: Motor_1
Error Text:

OK
Annulla
Applica
?