ORCA-EIP™ Reference Manual

UG260417

Introduction

ORCA-EIP is a cabinet-ready industrial Ethernet interface designed specifically for ORCA™ motors. It enables synchronized multi-axis control through EtherNet/IP while optimizing data throughput and deterministic timing for ORCA motor communication.

Engineered for PLC integration, industrial reliability, and simplified deployment, ORCA-EIP allows up to four ORCA motors to operate as coordinated nodes within modern automation systems.

The ORCA-EIP acts as an EtherNet/IP adapter which can send and receive a continuous stream of motor information via implicit (I/O) messaging from up to four motors. Motor settings that are not included in the implicit messaging stream can be pre-configured via IrisControls or via explicit asynchronous messaging supported by the ORCA-EIP.

ORCA Motor Firmware Version

The ORCA-EIP will only operate with ORCA motors with firmware versions:

- >= **6.4.0** (ORCA-6, ORCA-15)
- >= **7.2.0** (ORCA-3)

The firmware version can be checked on the home page of IrisControls. If your motor requires an update, and one was not provided with the ORCA-EIP, please contact Iris Dynamics.

Physical Hardware

Required Components

To operate the ORCA-EIP, the following hardware is required.

1. ORCA-EIP
2. 12-60V Power Supply
3. ORCA Motor (up to 4 supported)

ORCA-EIP Connections

- EtherNet/IP A/B
  - Ethernet interface for EtherNet/IP
  - Both inputs route to eth0
  - Defaults to DHCP unless configured otherwise
- ORCA Motor 0-4
  - Connect to up to four ORCA motors
  - Directly connect ORCA motor data cable into the RJ45 ports

    - OR -

    Connect the ORCA motor to an ORCA Starter Kit splitter, and connect a standard ethernet patch cable between output 2 of the splitter and any of the axes inputs. This configuration retains the ability to use IrisControls through output 1 of the splitter, if desired.
- VIN -/+
  - 12–60 VDC
- FFO +/-
  - Fast Force Off (not yet implemented)
- USB Serial
  - May output debug information from the firmware
  - Not a required connection for normal use

LEDs

ORCA 0 - 3

- Blue flashing: ORCA motor not connected to the ORCA -EIP
- Blue: ORCA motor connected to the ORCA -EIP
- Green: ORCA motor connected to the ORCA -EIP and network
- Red: ORCA motor error

EtherNet/IP

- Off: No IP configured
- Blue: IP configured
- Green: CIP connection present

FFO

- Not in use currently

Software Configuration

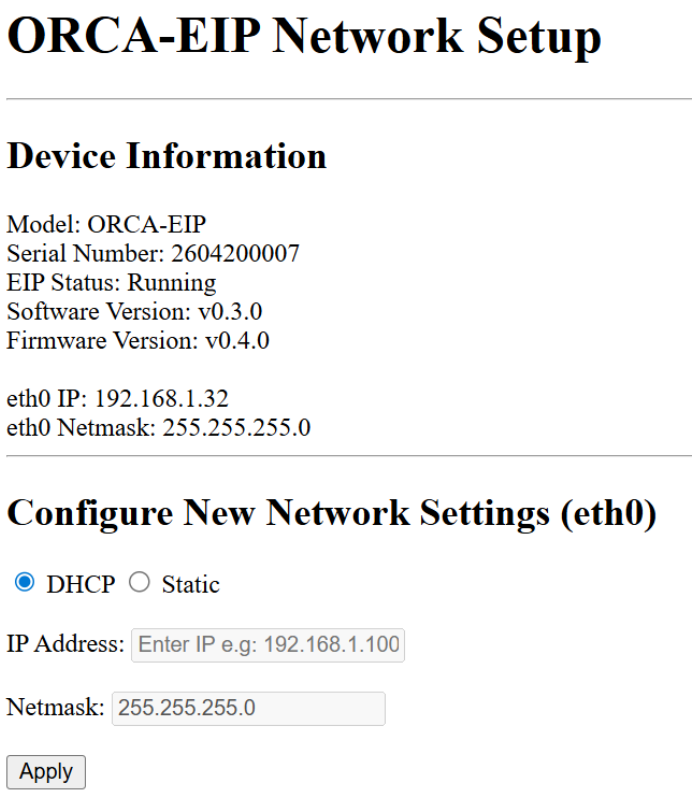
Changing ORCA-EIP IP Address for EtherNet/IP

The ORCA-EIP will arrive with eth0 in a DHCP configuration. If a static IP is preferred, you will first need to dynamically assign the device an IP then connect to the device's webserver to reconfigure the IP to static.

From Webserver

To access the webserver:

1. Plug your network cable into either eth0 port.
2. Open a browser and enter the device's IP into the address bar (e.g. 192.168.1.32).
3. You should see this page if you are connected correctly.



4. If you wish to configure the device as static, select static, enter your desired IP address and netmask and press apply.

WARNING: Once you change the IP, you will lose connection to the webserver. Change the address to the new IP to restore connection.

Through Serial Port / IrisControls

EDS File

An EDS file is available for download irisdynamics.com/downloads. This can be loaded into Studio 5000 or other PLC software. More information about PLC Integration.

Communication

I/O (IMPLICIT) Communication

The primary communication method between the PLC and the ORCA-EIP is EtherNet/IP implicit messaging. When the ORCA-EIP is added as an I/O device in the PLC configuration, cyclic data is exchanged automatically between the PLC and the ORCA-EIP according to the configured **Requested Packet Interval (RPI)**.

This data exchange occurs independently of the PLC logic scan and provides deterministic communication with the connected ORCA motors.

I/O communication can be configured in one of three Modes: Force, Position , Kinematic, corresponding with the different running modes of the ORCA motors. This is done by selecting which assembly instances to use when adding the I/O device. In all cases, use input assembly 100. The output assembly chosen will depend on the desire d mode. Mode cannot be changed during runtime and must be changed in configurations. See Appendix A for assemblies.

Explicit Communication

ORCA motor registers can be explicitly read and written using CIP explicit messaging. This uses CIP class #44 (Modbus), instance #1. In effect, we are making Modbus read and write

to the motor registers via the ORCA-EIP. Note: each ORCA register is 2 bytes, little-endian. See *ORCA Motors Reference Manual* for register table and information.

Register Address

The register address for each ORCA motor are mapped as follows:

Axis	Address
Axis 0	1000 + Local Register Address
Axis 1	2000 + Local Register Address
Axis 2	3000 + Local Register Address
Axis 3	4000 + Local Register Address

Service Codes & Message Format

Two service codes are supported by CIP class #44 (Modbus), instance #1. One for reading, one for writing.

0x4E: READ

Message Data[4bytes]: [2bytes start address], [2bytes number of registers to read]

Response Data: [2bytes 1st register value],[2bytes 2nd register value] ...etc.

0x50: WRITE

Message Data[4+2*number of registers to write]:
[2bytes start address], [2bytes number of registers to write], [2*number of registers to write]

Enable Sequence

1. Set ORCA-EIP Enable bit.
2. Set Axis Enable bit.
3. Write target position / force.

APPENDIX A: I/O and UDT Data Structures

All assemblies use data type SINT (1 byte), little-Endian (least significant byte first).

Output Assemblies

Size: 24 bytes

OUTPUT INSTANCE 170 : KINEMATIC					
Out Addr. (START)	Size (BYTES)	Axis	Type (UDT)	UDT Field	Units/Notes - Instance Dependant
0	4	Common	DINT	ORCA_EIP_Control_Words_Out	Command Bits .0 - ORCA-EIP Enable
4	4		DINT	Axis_Control_Words_Out	.0 -Axis 1 Enable .1 -Axis 2 Enable .2 -Axis 3 Enable .3 -Axis 4 Enable .4 -Axis 1 Autozero .5 -Axis 2 Autozero .6 -Axis 3 Autozero .7 -Axis 4 Autozero
8	4	Axis 1	DINT	Target_Axis_1	32 bit rising edge triggered bitmap for kinematic motion. .0 - Motion ID 0 .1 - Motion ID 1 .2 - Motion ID 2 . . . .31 - Motion ID 31
12	4	Axis 2	DINT	Target_Axis_2	32 bit rising edge triggered bitmap for kinematic motion
16	4	Axis 3	DINT	Target_Axis_3	32 bit rising edge triggered bitmap for kinematic motion
20	4	Axis 4	DINT	Target_Axis_4	32 bit rising edge triggered bitmap for kinematic motion

OUTPUT INSTANCE 160: POSITION					
Out Addr. (START)	Size (BYTES)	Axis	Type (UDT)	UDT Field	Units/Notes - Instance Dependant
0	4	Common	DINT	ORCA_EIP_Control_Words_Out	Command Bits .0 - ORCA-EIPEnable

4	4		DINT	Axis_Control_Words_Out	.0 -Axis 0 Enable .1 -Axis 1 Enable .2 -Axis 2 Enable .3 -Axis 3 Enable .4 -Axis 0 Autozero .5 -Axis 1 Autozero .6 -Axis 2 Autozero .7 -Axis 3 Autozero
8	4	Axis 1	DINT	Target_Axis_0	Position target in um
12	4	Axis 2	DINT	Target_Axis_1	Position target in um
16	4	Axis 3	DINT	Target_Axis_2	Position target in um
20	4	Axis 4	DINT	Target_Axis_3	Position target in um
OUTPUT INSTANCE 150: FORCE					
Out Addr. (START)	Size (BYTES)	Axis	Type (UDT)	UDT Field	Units/Notes - Instance Dependant
0	4	Common	DINT	ORCA_EIP_Control_Words_Out	Command Bits .0 - ORCA-EIP Enable
4	4		DINT	Axis_Control_Words_Out	.0 -Axis 0 Enable .1 -Axis 1 Enable .2 -Axis 2 Enable .3 -Axis 3 Enable .4 -Axis 0 Autozero .5 -Axis 1 Autozero .6 -Axis 2 Autozero .7 -Axis 3 Autozero
8	4	Axis 1	DINT	Target_Axis_0	
12	4	Axis 2	DINT	Target_Axis_1	
16	4	Axis 3	DINT	Target_Axis_2	
20	4	Axis 4	DINT	Target_Axis_3	

Input Assembly

Size: 216 bytes

INPUT INSTANCE 100					
Instance Data Address (START)	Size (BYTES)	Axis	Type (UDT)	Field (UDT)	Units/Notes
0	1	Common	BYTE	ORCA-EIP Status Word In	ORCA-EIP Status Bits .0 - ORCA-EIP Enabled : Always set.
1	1		BYTE	Axis Enable In	.0 - Axis 1 Enabled .1 - Axis 2 Enabled .2 - Axis 3 Enabled .3 - Axis 4 Enabled
2	1		BYTE	Axis Status In	.0 - Axis 1 Connected .1 - Axis 2 Connected .2 - Axis 3 Connected .3 - Axis 4 Connected
3	1		BYTE	GPIO_InputBitmap In	8 digital inputs, Not implemented
4	1		BYTE	GPIO_OutputBitmap In	8 digital outputs, Not implemented
5	3		--	--	Alignment / future
8	4	Axis 1	DINT	Force In 1	mN (millinewtons)
12	4		DINT	Position In 1	µm (micrometers)
16	4		DINT	Speed In 1	mm/s (0.001 mm/s increments)
20	4		DINT	Acceleration In 1	mm/s²
24	4		DINT	Board Temperature In 1	1°C increments
28	4		DINT	Coil Temperature In 1	1°C increments

32	4		DINT	Driver Voltage In 1	mV
36	4		DINT	Power In 1	W
40	4		DINT	Mode In 1	ORCA Axis 1 Mode 1 = Sleep 2 = Force 3 = Position 4 = Haptic 5 = Kinematic 11 = Pulse Width 55 = Auto Zeroing
44	4		DINT	Kinematic Motion Bitmap In 1	Bit reflecting current motion in progress
48	4		DINT	Kinematic Complete Count In 1	Not implemented
52	4		DINT	Errors In 1	Error Bits: .0 - Configuration .5 - Force Clipping .6 - Temp Exceeded .7 - Force Exceeded .8 - Power Exceeded .9 - Shaft Image Failed .10 - Voltage Invalid .11 - Comms Timeout .13 - Auto Zero Failed
56	4		DINT	Status Bits In 1	Axis 1 .0 - Auto zero complete .1 - Autozero in progress
60	4	Axis 2	DINT	Force In 2	mN (millinewtons)
64	4		DINT	Position In 2	µm (micrometers)
68	4		DINT	Speed In 2	mm/s (0.001 mm/s increments)
72	4		DINT	Acceleration In 2	mm/s²
76	4		DINT	Board Temperature In 2	1°C increments

80	4		DINT	Coil Temperature In 2	1°C increments
84	4		DINT	DriverVoltage In 2	mV
88	4		DINT	Power In 2	W
92	4		DINT	Mode In 2	ORCA Axis 2 Mode 1 = Sleep 2 = Force 3 = Position 4 = Haptic 5 = Kinematic 11 = Pulse Width 55 = Auto Zeroing
96	4		DINT	Kinematic Motion Bitmap In 2	Bit reflecting current motion in progress
100	4		DINT	Kinematic Complete Count In 2	Not implemented
104	4		DINT	Errors In 2	Error Bits: .0 - Configuration .5 - Force Clipping .6 - Temp Exceeded .7 - Force Exceeded .8 - Power Exceeded .9 - Shaft Image Failed .10 - Voltage Invalid .11 - Comms Timeout .13 - Auto Zero Failed
108	4		DINT	Status Bits In 2	Axis 2 .0 - Auto zero complete .1 - Autozero in progress
112	4	Axis 3	DINT	Force In 3	mN (millinewtons)
116	4		DINT	Position In 3	µm (micrometers)
120	4		DINT	Speed In 3	mm/s (0.001 mm/s increments)
124	4		DINT	Acceleration In 3	mm/s²

128	4		DINT	Board Temperature In 3	1°C increments
132	4		DINT	Coil Temperature In 3	1°C increments
136	4		DINT	Driver Voltage In 3	mV
140	4		DINT	Power In 3	W
144	4		DINT	Mode In 3	ORCA Axis 3 Mode 1 = Sleep 2 = Force 3 = Position 4 = Haptic 5 = Kinematic 11 = Pulse Width 55 = Auto Zeroing
148	4		DINT	Kinematic Motion Bitmap In 3	Bit reflecting current motion in progress
152	4		DINT	Kinematic Complete Count In 3	Not implemented
156	4		DINT	Errors In 3	Error Bits: .0 - Configuration .5 - Force Clipping .6 - Temp Exceeded .7 - Force Exceeded .8 - Power Exceeded .9 - Shaft Image Failed .10 - Voltage Invalid .11 - Comms Timeout .13 - Auto Zero Failed
160	4		DINT	Status Bits In 3	Axis 3 .0 - Auto zero complete .1 - Autozero in progress
164	4	Axis 4	DINT	Force In 4	mN (millinewtons)
168	4		DINT	Position In 4	µm (micrometers)

172	4		DINT	Speed In 4	mm/s (0.001 mm/s increments)
176	4		DINT	Acceleration In 4	mm/s²
180	4		DINT	Board Temperature In 4	1°C increments
184	4		DINT	Coil Temperature In 4	1°C increments
188	4		DINT	DriverVoltage In 4	mV
192	4		DINT	Power In 4	W
196	4		DINT	Mode In 4	ORCA Axis 4 Mode 1 = Sleep 2 = Force 3 = Position 4 = Haptic 5 = Kinematic 11 = Pulse Width 55 = Auto Zeroing
200	4		DINT	Kinematic Motion Bitmap In 4	Bit reflecting current motion in progress
204	4		DINT	Kinematic Complete Count In 4	Not implemented
208	4		DINT	Errors In 4	Error Bits: .0 - Configuration .5 - Force Clipping .6 - Temp Exceeded .7 - Force Exceeded .8 - Power Exceeded .9 - Shaft Image Failed .10 - Voltage Invalid .11 - Comms Timeout .13 - Auto Zero Failed
212	4		DINT	Status Bits In 4	Axis 4 .0 - Auto zero complete .1 - Autozero in progress

Config Assembly

Size: 50 bytes

CONFIG ASSEMBLY 140				
Instance Data Address (START)	Size (BYTES)	Type	Field	Units/Notes
0	2	UINT	PositionCtrl_PGain	Position controller proportional gain Default - 1000
2	2	UINT	PositionCtrl_IGain	Position controller integral gain
4	2	UINT	PositionCtrl_DVGain	Position controller derivative velocity gain
6	2	UINT	PositionCtrl_DEGain	Position controller derivative error gain
8	4	DINT	PositionCtrl_F_Saturation	Position controller force saturation
12	2	UINT	PositionCtrl_Softstart_ms	Position controller softstart
14	4	DINT	User_Max_Force_mN	User defined maximum force. value of 0 = no user limit
18	2	UINT	User_Max_Power_W	User defined maximum force. value of 0 = no user limit
20	2	UINT	User_Max_Board_Temp_C	User defined maximum temperature of electronics value of 0 = no user limit
22	2	UINT	User_Max_Coil_Temp_C	User defined maximum temperature of coils value of 0 = no user limit
24	2	UINT	Temp_Error_Hysteresis	Temperature error hysteresis. Reduction in temperature below limit required before error can be cleared
26	2	UINT	Safety_Damping	Communication disconnect/Fast force off safety dampin
28	2	UINT	Position_Inversion	Direction of positive position.
30	2	UINT	Pos_Max_Speed	Position control maximum velocity
32	2	UINT	Pos_Max_Acceleration	Position control maximum acceleration

34	2	UINT	Pos_Max_Deceleration	Position control maximum deceleration
36	2	UINT	Sensed_Force_Filter	Filter on the force sensed by the motor
38	2	UINT	Sensed_Position_Filter	Filter on the position sensed by the motor
40	2	UINT	Command_Force_Filter	Filter on the force commanded to the motor
42	2	UINT	Command_Position_Filter	Filter on the position commanded to the motor
44	2	UINT	Zero_Configuration	Type of zeroing 0 - Zeroes on retraction 1 - Manual zeroing (negative positions 2 - Autozeroing enabled 3 - Autozero on boot
46	2	UINT	Auto_Zero_Force_N	Maximum for to use during autozeroing
48	2	UINT	Auto_Zero_Speed_mmps	Maximum speed to use for autozeroing

Appendix B: Expanded UDT Field Descriptions

INPUT Fields

These fields are read only and reflect the current state of the ORCA-EIP and the connected motors.

ORCA-EIP Status In:

Bit 0: ORCA-EIP Enabled. This bit reflects the state of the ORCA -EIP Control Word bit 0. This bit is always enabled as long as the ORCA -EIP is running.

Axis Enable In:

Bit 0-3: These bits reflect the state of **ORCA-EIP Axis Control Word** (Output) bits. Check these bits to determine if the ORCA -EIP successfully received the **ORCA-EIP Axis Control Word** bits to enable the motors.

Axis Status In:

Bit 0-3: These bits reflect which ports of the ORCA -EIP have an ORCA motor connected and communicating with. Check to see which ports have powered motors connected to them.

Force In, Position, Speed In, Acceleration In, Board Temperature In, Coil Temperature In, Driver Voltage In, Power In:

Motor data. See appendix A or *ORCA Motors Reference Manual* for more information

Mode In:

A number indicated what ORCA mode the motor is running in. Depending on the assembly instances used, the ORCA -EIP will put the ORCA motor in the corresponding mode when both the ORCA -EIP enable bit and motor enable bits are set. Eg mode = 3 when assembly instance 160 is used. If auto zeroing is commanded, the motor mode will change to 55 while it completes that routine.

Kinematic Motion Bitmap In:

Motors can be configured with up to 32 kinematic motions. Each bit 0 -31 of this field represents the corresponding kinematic motion and is set while that motion is in progress. Only used if using kinematic output assembly instance 170. See *ORCA Motors Reference Manual* for more information on kinematic motions.

Error In:

Current errors on ORCA motors. See *ORCA Motors Reference Manual* for more information.

Status Bits:

Bit 0 : Set if the ORCA motor has completed an auto zero cycle since its last power cycle.

Bit 1: Set while the ORCA motors is performing an auto zero routine.

OUTPUT Fields

These fields read and write and are the commanded data sent to the ORCA-EIP and connected motors.

ORCA-EIP Control Words Out:

Bit 0 : A global enable for or disable for the ORCA-EIP module. When this bit is cleared, all connected motors modes will be set to Sleep. While in Sleep mode, updates to motor position or kinematic motions will be ignored. Auto zero is still possible while this bit = 0.

Axis Control Words Out:

Bit 0-3: Enable bits for axis 0-3.

Bit 3-7: Autozero bits for axis 0-3. See 'Auto-Zero Motors'

Target_Axis_X:

Different functions depending on the assembly instance used. See Position or Kinematic mode sections for detailed descriptions.