



---

**INDUSTRIAL CONTROL COMMUNICATIONS, INC.**

---

# **EtherCAT Slave Driver Manual**

---



## TABLE OF CONTENTS

<b>1 EtherCAT Slave .....</b>	<b>2</b>
1.1 Overview .....	2
1.2 Process Data Exchange .....	2
1.3 Supported Services .....	2
1.4 CANopen Over EtherCAT (CoE) Object Library .....	3
1.5 Process Data Settings .....	6

# 1 EtherCAT Slave

## 1.1 Overview

The gateway supports an EtherCAT slave driver on the EtherCAT ports. Some notes of interest are:

- CANopen over EtherCAT (CoE) is supported with the last 128 bytes of the internal database directly accessible via byte, word, or double word arrays.
- The driver is compliant with the DS301 communication profile.
- Up to 254 bytes of process data can be mapped in each direction.

## 1.2 Process Data Exchange

The driver supports up to 254 bytes each of transmit and receive process data. The transmit and receive data sizes and database mappings are configured using the Configuration Studio (refer to section 1.5.) The designated starting address, combined with the corresponding size, defines a block of data for transmission or reception.

## 1.3 Supported Services

The driver supports the following EtherCAT services:

- Auto increment physical read (APRD)
- Auto increment physical write (APWR)
- Auto increment Read Write (APRW)
- Configured address read (FPRD)
- Configured address write (FPWR)
- Configured address Read Write (FPRW)
- Broadcast Read (BRD)
- Broadcast Write (BWR)
- Logical Read (LRD)
- Logical Write (LWR)
- Logical Read Write (LRW)
- Auto increment physical read multiple write (ARMW)
- Configured read multiple write (FRMW)

The driver also supports the following CANopen over EtherCAT (CoE) services:

- SDO Download Expedited (writes up to four octets to the slave)
- SDO Download Normal (writes up to a negotiated number of octets to the slave)
- Download SDO Segment (writes additional data if the object size exceeds the negotiated number of octets)

- SDO Upload Expedited (reads up to four octets from the slave)
- SDO Upload Normal (reads up to a negotiated number of octets from the slave)
- Upload SDO Segment (reads additional data if the object size exceeds the negotiated number of octets)
- Abort SDO Transfer (server abort of service in case of an erroneous condition)
- Get OD List (reads a list of available indices)
- Get Object Description (reads details of an index)
- Get Entry Description (reads details of a sub-index)
- Emergency (reports unexpected conditions)

#### **1.4 CANopen Over EtherCAT (CoE) Object Library**

The driver supports acyclic data exchange using the CoE object library. The standard object dictionary is implemented according to the DS301 communication profile. The last 128 bytes of the internal database are also accessible via the CoE object library as manufacturer-specific objects. There are three such objects located at indices 2001, 2002 and 2003, providing byte, word and double-word access to the database, respectively. Refer to Table 1.

**Table 1: CoE Object Library**

Index	Object Name	Sub-Index	Description	Type	Access	Notes
1000	Device type	00h	Device Type	U32	RO	0000 0000h (No profile)
1001	Error register	00h	Error Register	U8	RO	
1003	Pre-defined error field	00h	# of errors	U8	RW	
		01h...05h	Error field	U32	RO	
1008	Device name	00h	Manufacturer device name	Visible string	RO	ECAT-1000
1009	Hardware version	00h	Manufacturer hardware version	Visible string	RO	REV:A
1011	Restore parameters	00h	Largest sub index supported	U8	RO	01h
		01h	Restore all default parameters	U32	RW	
1018	Identity object	00h	# of entries	U8	RO	04h
		01h	Vendor ID	U32	RO	0000 0558h
		02h	Product Code	U32	RO	0000 0001h
		03h	Revision Number	U32	RO	0002 0000h
		04h	Serial Number	U32	RO	
1600	Receive PDO mapping	00h	# of mapped application objects in PDO	U8	RO	# of mapped objects (0...254)
		01h	Mapped obj #1	U32	RO	
		02h	Mapped obj #2	U32	RO	
		:	:	:	:	
		NNh	Mapped obj #NN	U32	RO	
1A00	Transmit PDO mapping	00h	# of mapped application objects in PDO	U8	RO	# of mapped objects (0...254)
		01h	Mapped obj #1	U32	RO	
		02h	Mapped obj #2	U32	RO	
		:	:	:	:	
		NNh	Mapped obj #NN	U32	RO	
1C00	Sync Mgr Comm. Type	00h	# of entries	U8	RO	4
		01h	Mailbox write	U8	RO	1
		02h	Mailbox read	U8	RO	2
		03h	Process Data out	U8	RO	3
		04h	Process Data in	U8	RO	4
1C12	Sync Mgr RxPDO assignment	00h	# of assigned PDOs	U8	RO	1
		01h	Assigned PDO	U16	RO	1600h

Index	Object Name	Sub-Index	Description	Type	Access	Notes
1C13	Sync Mgr TxPDO assignment	00h	# of assigned PDOs	U8	RO	1
		01h	Assigned PDO	U16	RO	1A00h
1C32	SM output parameter	00h	# of entries	U8	RO	1
		01h	Sync mode	U16	RO	0 (FREE_RUN)
1C33	SM input parameter	00h	# of entries	U8	RO	1
		01h	Sync mode	U8	RO	0 (FREE_RUN)
2001	Byte Array 0	00h	# of bytes	U8	RO	80h total
		01h	Byte 1	U8	RW	
		02h	Byte 2	U8	RW	
		:	:	:	:	
		80h	Byte 128	U8	RW	
2002	Word Array 0	00h	# of words	U8	RO	40h total
		01h	Word 1	U16	RW	
		02h	Word 2	U16	RW	
		:	:	:	:	
		40h	Word 64	U16	RW	
2003	Double Word Array 0	00h	# of double words	U8	RO	20h total
		01h	DWord 1	U32	RW	
		02h	DWord 2	U32	RW	
		:	:	:	:	
		20h	DWord 32	U32	RW	
2004	Read PD	00h	# of mapped application objects in PDO	U8	RO	# of mapped objects (0...254)
		01h	Mapped obj #1	U32	RO	
		02h	Mapped obj #2	U32	RO	
		:	:	:	:	
		NNh	Mapped obj #NN	U32	RO	
2005	Write PD	00h	# of mapped application objects in PDO	U8	RO	# of mapped objects (0...254)
		01h	Mapped obj #1	U32	RO	
		02h	Mapped obj #2	U32	RO	
		:	:	:	:	
		NNh	Mapped obj #NN	U32	RO	

## 1.5 Process Data Settings

Process data support is automatically added to the EtherCAT driver, and cannot be removed.

### **Transmit Data Start Address**

Specifies the starting address in the internal database for the transmit process data. The configuration studio will not allow entry of a starting database address that will cause the transmit data to run past the end of the database. The highest valid database address, therefore, will depend on the *Transmit Data Size* setting.

### **Transmit Data Size**

Specifies the size in bytes of the transmit process data (0...254).

### **Receive Data Start Address**

Specifies the starting address in the internal database for the receive process data. The configuration studio will not allow entry of a starting database address that will cause the receive data to run past the end of the database. The highest valid database address, therefore, will depend on the *Receive Data Size* setting.

### **Receive Data Size**

Specifies the size in bytes of the receive process data (0...254).



---

**INDUSTRIAL CONTROL COMMUNICATIONS, INC.**

1600 Aspen Commons, Suite 210  
Middleton, WI USA 53562-4720  
Tel: [608] 831-1255 Fax: [608] 831-2045

<http://www.iccdesigns.com>

Printed in U.S.A