Measuring amplifier GSV-8

EtherCat Protocol

Last edit: 21/03/2015

Table of contents

General....5
EtherCAT System Architecture....5
EtherCAT Protocol....6
EtherCAT Slave Architecture....6
EEPROM EtherCAT Slave Configuration....7
States of an EtherCAT slave....7
EtherCAT Commands....8
Connection of the EtherCAT Wires....9
Error Messages....10
GSV-8 EtherCat Implementation....10
Distributed clocks....11
Object Dictionary....11
Index 1000h: Device Type....12
Index 1001h: Error Register....12
Index 1008h: Device Name....13
Index 1009h: Hardware Version....13
Index 100Ah: Software Version....13
Index 1018h: Identity....13
Sub-Index 1: Vendor ID....14
Sub-Index 2: Product Code....14
Sub-Index 3: Revision....14
Sub-Index 4: Serial Number....14
Index 10F1h: Error Settings (EtherCAT)....14
Sub-Index 1: Local Error Reaction....14
Sub-Index 2: Sync Error Counter Limit....14
Index 1A00h: TxPDO Map 1....15
Dynamic PDO-Mapping....15
Index 1C00h: Sync Manager Type (EtherCAT)....16
Index 1C12h: RxPDO assign (Sync Manager / EtherCAT)....16
Index 1C13h: TxPDO Assign (Sync Manager / EtherCAT)....17
Index 1C32h: Sync Manager 2: Output Parameter (EtherCAT)....17
Index 1C33h: Sync Manager 3: Input Parameter (EtherCAT)....17
Sub-Index 1: Synchronization Type....18
Sub-Index 2: Cycle Time....18
Sub-Index 4: Synchronization Types supported....18
Sub-Index 5: Minimum Cycle Time....18
Sub-Index 6: Calc and Copy Time....18
Sub-Index 8: Get Cycle Time....18
Sub-Index 9: Delay Time....18
Sub-Index 10: Sync0 Cycle Time....18
Sub-Index 11: SyncManager Event missed....19
Sub-Index 12: Cycle Time Too Small....19
Sub-Index 32: Sync Error....19
Index 6112h: AI Operating mode....19
Index 6114h: AI ADC Sample Rate....19
Index 6125h: AI Autozero....20
Index 6126h: AI Scaling factor....20

Index 6127h: AI Scaling offset.....21
Index 6130h: AI Process Value.....22
Index 6148h: AI Span start.....22
Index 6149h: AI Span end.....23
Index 6150h: AI Status.....23
Index 61A0h: AI Filter type.....24
Index 61A1h: AI Filter constant.....24
Index 9100h: AI Field Value.....25
Subject to modifications.....25
Trademarks and Patents 25

General

Besides the default USB interface the GSV-8-EtherCAT offers an additional EtherCAT interface. The application layer is implemented according to the EtherCAT standard, using CoE (CANopen over EtherCAT). The EtherCAT protocol is defined in IEC61158.

The EtherCAT protocol uses standard Ethernet frames with Ether-type 0x88A4. An EtherCAT telegram may consist of several sub telegrams, the so called EtherCAT commands. Most of the device configurations are done via the EtherCAT commands.

EtherCAT Overview
An EtherCAT network is built of a master and one or more slaves. While the master is usually built of a powerful PC the slaves are optimized for the fast transfer of the process data from and to the several control units. The algorithms to process the data transfer are implemented on the PC.
The main features of EtherCAT are:

- **A broad applicability**. As an EtherCAT master every commercially available computer with a normal Ethernet controller can be used.
- **Compliance to the Ethernet standard**. EtherCAT is fully compliant to the Ethernet standard and can coexist with other Ethernet devices on the same bus.
- **Highest efficiency**. The Ethernet bandwidth is nearly fully usable.
- **Short cycle times**. The EtherCAT cycle times are typically in the range of 10..50 µs.
- **Variety of bus topologies**. EtherCAT supports a huge variety of bus topologies, e. g. line, ring, star, etc., thus supporting redundancy; hot connect of segments and device exchange in an active network.

EtherCAT System Architecture

On the perspective of the “normal” Ethernet topology the EtherCAT bus shows up as a single Ethernet participant. Within this “participant” however there is no Ethernet controller with an application processor but rather several EtherCAT slaves.
As shown in Figure the EtherCAT master uses the network configuration which is stored in the EtherCAT Network Information file (ENI). The ENI is created by the EtherCAT Configuration Tool based on the EtherCAT Slave Information (ESI), which is provided for every device by the vendor (for GSV-8: GSV8EtherCat<VerNo>rev<RevNo>.xml, available on www-me-systeme.de). The slaves are connected via standard Ethernet cables.
The EtherCAT master system just requires a standard Network Interface Controller (NIC, 100 Mbit/s Full duplex) and a real time run-time environment that drives the slaves in the network.

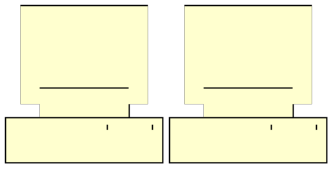

Figure : EtherCAT Network Architecture

The slaves process the data “on the fly” while receiving them and putting appropriate

process data into the stream in the same instance. I. e. the data stream is not copied first than processed and finally an answer is sent back. In fact the whole processing takes place in one step while the stream is running through a slave device, thus the whole frame is delayed by just a few bits. Figure shows this principle.

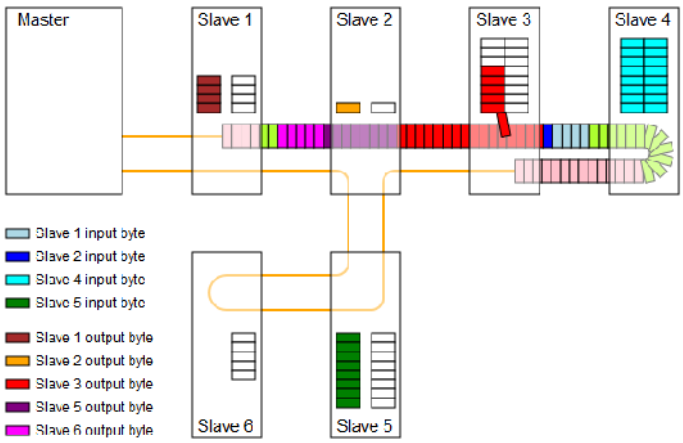

Figure : EtherCAT communication principle

Each slave device holds an addressable 64kByte RAM area that can be read or written to or even read and written in the same cycle. There may be multiple EtherCAT commands within one Ethernet frame to address or read/write individual slave devices.

EtherCAT Protocol

The EtherCat commands were transferred in the datagram area of an Ethernet frame as shown in Figure . Since the EtherType is set to 0x88A4 to identify an EtherCAT frame they will not pass any router. The GSV-8 only supports the Direct Mode Addressing, i.e. the IP address and MAC of an Ethernet frame is ignored.
Each EtherCAT datagram consists of the datagram header, the data and the so called “working counter” (WKC). The working counter is incremented by every device that is addressed by an EtherCAT command.

Figure : EtherCAT protocol frame

Field	Value / Description
Length	Length of the EtherCAT datagrams (excl. FCS)
Res	Reserved, 0
Type	Protocol Type. Only EtherCAT commands (Type = 0x01) are supported by ESCs
Cmd	EtherCAT Command Type
Idx	The index is a numeric identifier used by the master for identification of duplicates / lost datagrams that shall not be changed by the slaves.

Table : EtherCAT field descriptions

EtherCAT Slave Architecture

The main components of the EtherCAT slaves are:

- Physical Layer: Network interface

- Data Link Layer: EtherCAT Slave Controller (ESC, communication module) and EEPROM
- Application Layer: Application controller or microcontroller

Figure : EtherCAT Slave Architecture

The ESC is a hardware module for EtherCAT communication. The ESC handles the EtherCAT protocol in real-time by processing the EtherCAT frames on the fly and providing the interface for data exchange between the EtherCAT master and the slave's local application controller via registers and a DPRAM.
The ESC processes EtherCAT frames on the fly and exchanges data with the local controller of the GSV-8, which processes the measuring data.

EEPROM EtherCAT Slave Configuration

Since the DPRAM in the ESC is a volatile RAM, it also has an EEPROM (NVRAM, also called Slave Information Interface, SII). The EEPROM stores slave identity information and information about the slave's functionality corresponding to the ESI file, see Figure . The content of the EEPROM is configured by the manufacturer with necessary (default) settings. EEPROM information can be derived from the ESI file.

Figure : EEPROM Registers

States of an EtherCAT slave

The slave runs a state machine to indicate which functionalities are actually available. This EtherCAT State Machine (ESM) is shown in Figure .
ESM requests are written by the master to the slave's AL Control register in the ESC. If the configuration for the requested state is valid, the slave acknowledges the state by setting the AL Status register. If not, the slave sets the error flag in the AL Status register and writes an error code to the AL Status Code register.

Figure : EtherCAT Slave State Machine

State	**Available Functions**
INIT	Init state. No communication on the application layer is available. The master has access only to the DL-information registers.
PREOP	Pre-Operational state. Mailbox communication on the application layer available, but no process data communication available.
SAFEOP	Safe-Operational state. Mailbox communication on the application layer, process (input) data communication available. In SafeOp only inputs like measuring values are evaluated; outputs are kept in 'safe' state.
OP	Operational state. Process data inputs and outputs are valid.

BOOT	Bootstrap state. In this state some devices use the FoE protocol for firmware download. Not supported by GSV-8, which uses the USB port for firmware download.

Table : EtherCAT slave states

The initialization information of every EtherCAT state transition is derived from the EtherCAT Slave Information file (ESI, available from ME-Messsysteme) by a network configurator and stored in the network information file (ENI). Each slave gets its required initialization commands for each state transition. The EtherCAT master initializes the slave(s) using this ENI, e.g. logical slave I/O mapping is done according to the network topology.

EtherCAT Commands

All supported EtherCAT Command types are listed in table below. For Read-write operations, the Read operation is performed before the Write operation.

CMD	Abbr.	Name	Description
0	NOP	No operation	Slave ignores command
1	APRD	Auto Increment Read	Slave increments address. Slave puts read data into the EtherCAT datagram if received address is zero.
2	APWR	Auto Increment Write	Slave increments address. Slave writes data into memory location if received address is zero.
3	APRW	Auto Increment R/W	Slave increments address. Slave puts read data into the EtherCAT datagram and writes the data into the same memory location if received address is zero.
4	FPRD	Configured Address Read	Slave puts read data into the EtherCAT datagram if address matches with one of its configured addresses.
5	FPWR	Configured Address Write	Slave writes data into memory location if address matches with one of its configured addresses
6	FPRW	Configured Address R/W	Slave puts read data into the EtherCAT datagram and writes data into the same memory location if address matches with one of its configured addresses.
7	BRD	Broadcast Read	All slaves put logical OR of data of the memory area and data of the EtherCAT datagram into the EtherCAT datagram. All slaves increment position field.
8	BRW	Broadcast Write	All slaves write data into memory location. All slaves increment position field.

CMD	Abbr.	Name	Description
9	BRW	Broadcast R/W	All slaves put logical OR of data of the memory area and data of the EtherCAT datagram into the EtherCAT datagram, and write data into memory location. BRW is typically not used. All slaves increment position field.
10	LRD	Logical Memory Read	Slave puts read data into the EtherCAT datagram if received address matches with one of the configured FMMU areas for reading.
11	LWR	Logical Memory Write	Slaves writes data to into memory location if received address matches with one of the configured FMMU areas for writing.
12	LRW	Logical Memory R/W	Slave puts read data into the EtherCAT datagram if received address matches with one of the configured FMMU areas for reading. Slaves writes data to into memory location if received address matches with one of the configured FMMU areas for writing.
13	ARMW	Auto Increment Read Multiple Write	Slave increments address. Slave puts read data into the EtherCAT datagram if received address is zero, otherwise slave writes the data into memory location.

Table : EtherCAT Commands

Connection of the EtherCAT Wires

EtherCAT two network connectors. The connection Port0 is used for the connection towards the EtherCAT-Master, Port1 to connect more Slaves. The GSV-8 uses M12 connectors (IP code 67) according to IEC 61918, annex H.

Signal	Function	Pin-No. M12 connector	Pin-No. RJ45
TD+	Transmit data +	**1**	1
TD-	Transmit data -	**3**	2
RD+	Receive data +	**2**	3
RD-	Receive data -	**4**	6

Error Messages

These error codes are returned by the Mailbox/SDO when accessing the Object Dictionary.

Error Code (hex)	Meaning
0503 0000	Toggle bit not changed
0504 0000	SDO protocol timeout (*reserved*)
0504 0001	Command Byte invalid or unknown
0601 0000	Unsupported access to an object
0601 0001	Attempt to read a write only object
0601 0002	Attempt to write a read only object
0601 0003	Entry can not be written because Subindex0 is not 0
0602 0000	Object does not exist in the object dictionary
0604 0041	Object can not be mapped to PDO
0604 0042	Mapped Object exceeds PDO (*reserved*)
0604 0043	General parameter incompatibility reason (*reserved*)
0604 0047	Device incompatibility (*reserved*)
0606 0000	Access failed due to an hardware error (*reserved*)
0607 0010	Parameter length error
0607 0012	Length of service parameter too high (*reserved*)
0607 0013	Length of service parameter too low (*reserved*)
0609 0011	Sub-index does not exist
0609 0030	Invalid value for parameter (download only)
0609 0031	Value of parameter too high (download only)
0609 0032	Value of parameter too low (download only)
0800 0000	General error (*reserved*)
0800 0020	Data cannot be transferred or stored to the application
0800 0022	Data cannot be transferred or stored to the application in the present device state.
0800 0023	Object is not in the object dictionary

Table : Mailbox/SDO Error Codes

GSV-8 EtherCat Implementation

The GSV-8 measuring amplifier has eight analogue inputs, which are digitized with a 24 Bit Sigma-Delta analog-to-digital converter, who converts all channels simultaneously. In conjunction with the buffered-mode Sync-Manager of the EtherCat interface, the data of the eight channels representing the excitation of the plugged sensors belong exactly to the same point of time. The rate, with which the measuring controller updates the values, is

configurable from 1 to 48000 values/second.
The GSV-8 implements the CoE CanOpen 404 device profile, which is used for measuring devices.
Most configuration parameters can be accessed through objects communicated through the Sync-Manager's mailbox mode. The objects are identified by indices, which in itself are sub-divided into sub-indices. Sub-indices 1 to 8 often refer to the corresponding analogue input channel configuration.

The EtherCat Slave interface is realized using the Slave controller ASIC ET1100 by Beckhoff Automation GmbH, Germany.

Distributed clocks

The synchronization mode used is the telegram synchronous mode, synchronous with Sync Manager 3, which is used for the process data inputs, containing the mapped measurement values (see Obj.Descr. 1A00h and 1C33h).

The hardware synchronization mode available is the Latch 1 method.

After a measurement data frame acquisition is completed, 1.45 µs later the slave controller captures the system time of the positive edge of its Latch 1 input (ET1100 registers 0x09C0..0x09C7).

When the calculation of the physical PDO values is completed, the system time of the negative edge of the Latch 1 input is captured by the ET1100 (registers 0x09C8..0x09CF).

Object Dictionary

This chapter specifies the objects of the GSV-8-EtherCAT implementation.

Index (Hex)	Name	Type
1000	Device Type	Communication
1001	Error Register	Communication
1008	Device Name	Communication
1009	Hardware Version	Communication
100A	Software Version	Communication
1018	Identity Object	Communication
10F1	Error Settings	Communication
1A00	Tx PDO 1 Mapping Parameter	Communication
1C00	Sync Manager Type	Communication
1C12	RxPDO Assign	Communication
1C13	TxPDO Assign	Communication
1C32	SyncManager Output Parameter	Communication

Index (Hex)	Name	Type
1C33	SyncManager Input Parameter	Communication
6112	AI Operating Mode	Application, **A**nalog **I**nput
6114	AI ADC sample rate	Application, **A**nalog **I**nput
6125	AI Autozero	Application, **A**nalog **I**nput
6126	AI Scaling Factor	Application, **A**nalog **I**nput
6127	AI Scaling Offset	Application, **A**nalog **I**nput
6130	AI Process Value Float	Application, **A**nalog **I**nput
6148	AI Span Start Float	Application, **A**nalog **I**nput
6149	AI Span End Float	Application, **A**nalog **I**nput
6150	AI Status	Application, **A**nalog **I**nput
61A0	AI Filter Type	Application, **A**nalog **I**nput
61A1	AI Filter constant	Application, **A**nalog **I**nput
9100	AI Field Value	Application, **A**nalog **I**nput

Table : Object Dictionary

Index 1000h: Device Type

This object describes the device type and which profile the device is conforming to.

Sub-Index	Data Type	Access	Description	Default value
0	Unsigned32	ro	Device Type/Profile	0x00220194

Table : Index 1000h

This object is read-only and only implements the sub-index 0. Any other access will result in an error.
The value 0x00220194 contains the following two fields:

- Device Profile Number 0194h = 404d
- Additional Information 0022h = 00100010b

The device is claiming conformity to the CiA404 Measuring Device Profile and that the following function blocks are present:

- Bit 16: Digital input (optionally)
- Bit 17: Analogue input block (always present in standard GSV-8)
- Bit 18: Digital output (optionally)
- Bit 21: Alarm block (optionally)
- Bit 31: Device-specific PDO-Mapping (always set)

Index 1001h: Error Register

This object describes the device Error state.

Sub-Index	Data Type	Access	Description	Default value
0	Unsigned8	ro	Error flags	0x00

Table : Index 1001h

This object is read-only and only implements the sub-index 0. Any other access will result in an error.
Of these flags the following are implemented:

- Bit 0: Generic Error. This flag will be 1 on any error and 0 if there is none.

Index 1008h: Device Name

This object contains the device name.

Sub-Index	Data Type	Access	Description	Default Value
0	String(5)	ro	Device Name	"GSV-8"

Table : Index 1008h

This object is read-only and only implements the sub-index 0. Any other access will result in an error.

Index 1009h: Hardware Version

This object contains the hardware version string.

Sub-Index	Data Type	Access	Description	Default Value
0	String(2)	ro	Hardware version	"01"

Table : Index 1009h

This object is read-only and only implements the sub-index 0. Any other access will result in an error.

Index 100Ah: Software Version

This object contains the software version string.

Sub-Index	Data Type	Access	Description	Default Value
0	String(5)	ro	Software version	"01.02"

Table : Index 100Ah

This object is read-only and only implements the sub-index 0. Any other access will result in an error.

Index 1018h: Identity

This object contains the device identity.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x04
1	Unsigned32	ro	Vendor ID	0x00000270
2	Unsigned32	ro	Product code	0x00080000
3	Unsigned32	ro	Revision	0x00010002

Sub-Index	Data Type	Access	Description	Default Value
4	Unsigned32	ro	Serial number	-

Table : Index 1018h

This object is read-only and only implements the sub-index 0 to 4. Any other access will result in an error.

Sub-Index 1: Vendor ID

The vendor ID is a unique manufacturer identification number assigned by the EtherCAT association. ME Meßsysteme GmbH has the vendor ID = **0270h**

Sub-Index 2: Product Code

The product code is a unique identification number of the product assigned by the vendor. **00080000h** is the code for GSV-8

Sub-Index 3: Revision

The revision is currently the binary equivalent to the software version (**100Ah**).

Sub-Index 4: Serial Number

The number in its decimal representation, prepended with zeros if the representation has less than 8 digits, can be found on the device's specification plate.

Index 10F1h: Error Settings (EtherCAT)

This object contains the EtherCAT error setting.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x02
1	Unsigned32	rw	Local Error Reaction	0x00000000
2	Unsigned32	rw	Sync Error Counter Limit	0x00000000

Table : Index 10F1h

This object is read-only on sub-index 0 and read/write on sub-index 1 to 8. Any other access will result in an error.

Sub-Index 1: Local Error Reaction

The Local Error Reaction defines how the slave shall behave if a local error occurs.
(1) PDO state
(2) Disable SyncManager
(3) Device specific state
The default setting is to report the error via PDO state.

Sub-Index 2: Sync Error Counter Limit

The Sync Error Counter is incremented with every missing Sync Management Event and decremented if an event is received. If the Sync Error Counter exceeds this limit the system changes into the SAFEOP state with the 'Synchronization Lost' error. The Sync

Error Counter is reset when the error was acknowledged.

Index 1A00h: TxPDO Map 1

This object contains the transmit PDO mapping; the description which object value has to be transmitted on request from the EtherCAT master.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	rw	Highest sub-index	0x10
1	Unsigned32	rw	Mapping Object 1	0x61300120[1]
2	Unsigned32	rw	Mapping Object 2	0x61300220 [1]
3	Unsigned32	rw	Mapping Object 3	0x61300320 [1]
4	Unsigned32	rw	Mapping Object 4	0x61300420 [1]
5	Unsigned32	rw	Mapping Object 5	0x61300520 [1]
6	Unsigned32	rw	Mapping Object 6	0x61300620 [1]
7	Unsigned32	rw	Mapping Object 7	0x61300720 [1]
8	Unsigned32	rw	Mapping Object 8	0x61300820 [1]
9	Unsigned32	rw	Mapping Object 9	0x61500120
10	Unsigned32	rw	Mapping Object 10	0x61500220
11	Unsigned32	rw	Mapping Object 11	0x61500320
12	Unsigned32	rw	Mapping Object 12	0x61500420
13	Unsigned32	rw	Mapping Object 13	0x61500520
14	Unsigned32	rw	Mapping Object 14	0x61500620
15	Unsigned32	rw	Mapping Object 15	0x61500720
16	Unsigned32	rw	Mapping Object 16	0x61500820

Table : Index 1A00h

This object implements the sub-index 0 to 16 for read/write-access. A write to sub-index 0 with a value greater than 16 will result in an error. Any access on another sub-index will also result in an error.

Each entry has the following form:

Bits 31:16	Bits 15:8	Bits 7:0
Index	Sub-Index	Size in Bits

Table : Sub-Index Bit Mapping for Index 1A00h

Dynamic PDO-Mapping

To change the TxPDO mapping one has to follow the following Steps:

1. The device must be placed into the PREOP-state.
2. The PDO mapping has to be invalidated by writing a Null into the sub-index 0

1 Def. value applies to device software version **1.26 and higher**. Older versions map Obj. 0x9100 (AI Process value Int32) by default at 1A00.1 =91000120 to 1A00.8 =91000820.

3. Change the PDO mappings by writing the desired object index, sub-index and Bit-size value into the sub-index 1 to 16. Do not leave empty entries in between.
4. Make the PDO mapping valid by writing the number of the highest sub-index into sub-index 0.
5. Switch the device back into SafeOP- or OP-state.

Caution: The EtherCAT implementation only allows entries of the objects **6130h** (AI Process value), **6150h** (AI Status) and **9100h** (AI Field value) in the TxPDO mapping object entries.

Index 1C00h: Sync Manager Type (EtherCAT)

This object contains the Sync Manager types implemented in the device firmware.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x04
1	Unsigned8	ro	Sync Manager Type channel 0	0x01
2	Unsigned8	ro	Sync Manager Type channel 1	0x02
3	Unsigned8	ro	Sync Manager Type channel 2	0x03
4	Unsigned8	ro	Sync Manager Type channel 3	0x04

Table : Index 1C00h

This object is read-only and only implements the sub-index 0 to 4. Any other access will result in an error.

This object contains the Sync Manager types implemented in the device firmware.

Sync Manager Channel	Value	Description
0	1	Mailbox Write (SDO Request)
1	2	Mailbox Read (SDO Response)
2	3	Process Output Data (RxPDO, unused)
3	4	Process Input Data (TxPDO)

Table : Sync Manager Channels of Index 1C00h

Index 1C12h: RxPDO assign (Sync Manager / EtherCAT)

This object contains the Sync Manager RxPDO assignment.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x00
1	Unsigned16	ro	Active RxPDO Map 1	-

Table : Index 1C12h

This object is read-only and only implements the sub-index 0 to 1. Any other access will result in an error.
If Digital output and Alarm function block (see object 1000h) is not present, this object has

no active RxPDO (sub-index 0 is 0x00), therefore the value of sub-index 1 is left unspecified.

Index 1C13h: TxPDO Assign (Sync Manager / EtherCAT)

This object contains the Sync Manager TxPDO assignment.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x01
1	Unsigned16	ro	Active TxPDO Map 1	0x1A00

Table : Index 1C13h

This object is read-only and only implements the sub-index 0 to 1. Any other access will result in an error.
The active TxPDO map is used by the Sync Manger to point to the active TxPDO. The TxPDO Map 1 (**1A00h**) is always used.

Index 1C32h: Sync Manager 2: Output Parameter (EtherCAT)

This object contains the Sync Manager parameter for ‘outputs’ (RxPDO) assigned in the RxPDO assignment (**1C12h**) object.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x00

Table : Index 1C32h

This object is read-only and only implements the sub-index 0 only. Any other access will result in an error.
All time entries are specified in nanoseconds.

Index 1C33h: Sync Manager 3: Input Parameter (EtherCAT)

This object contains the Sync Manager parameter for ‘inputs’ (TxPDO) assigned in the TxPDO assignment (**1C13h**) object.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x20
1	Unsigned16	rw	Synchronization type	0x0001
2	Unsigned32	ro	Cycle time	0x00000000
3	-	ro	Unused	-
4	Unsigned16	ro	Synchronization types supported	0x4003
5	Unsigned32	ro	Minimum cycle time	0x00005161
6	Unsigned32	ro	Calc and copy time	0x00001644
7	-	ro	Unused	-
8	Unsigned16	rw	Get cycle time	0x0000
9	Unsigned32	ro	Delay time	0x00000000
10	Unsigned32	rw	Sync0 cycle time	0x00000000

Sub-Index	Data Type	Access	Description	Default Value
11	Unsigned16	ro	Sync Manager event missed	0x00000000
12	Unsigned16	ro	Cycle time too small	0x00000000
13..31	-	ro	Unused	-
32	Bool	ro	Sync error	0

Table : Index 1C33h

This object is read-only and only implements the sub-index 0 to 32. Any other access will result in an error.
All time entries are specified in nanoseconds.

Sub-Index 1: Synchronization Type

The set synchronization type is set to 0x0001 ‘Synchronous with SyncManager 3’.
By setting the value to 0x0000, free-run mode is enabled.

Sub-Index 2: Cycle Time

This defines the minimum time between two SyncManager events.

Sub-Index 4: Synchronization Types supported

The bits indicate which synchronization type the implementation supports. It is for information only.

- Bit 0: Free Run supported
- Bit 1: Synchronous Mode supported
- Bit 14: Dynamic cycle times supported

Sub-Index 5: Minimum Cycle Time

This defines the minimum time between two SyncManager events which the application supports.

Sub-Index 6: Calc and Copy Time

Time needed by the application to perform the necessary calculations and copy the process data from the local memory to the SyncManager.

Sub-Index 8: Get Cycle Time

This entry controls the measurement of the local cycle time.

- Bit 0 controls if the measurement is active (1) or inactive (0).
- Bit 1 controls a reset of all measurement times.

Sub-Index 9: Delay Time

Definition of the delay time of the hardware before the latched value becomes active.
In the present mode of operation this value is unused.

Sub-Index 10: Sync0 Cycle Time

Communication cycle time in nanoseconds, between two Sync0 events.

In the present mode of operation this value is unused.

Sub-Index 11: SyncManager Event missed

The error counter used for missing Sync-Events (see **10F1h**).

Sub-Index 12: Cycle Time Too Small

This error counter is incremented, if the time between two Sync-Events is too small, so that a local cycle cannot be completed and input data cannot be provided before the next Sync-Event.

Sub-Index 32: Sync Error

Flag to indicate a SyncManger-Event missed has occurred.

Index 6112h: AI Operating mode

This object represents the device's mode of operation.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Unsigned8	ro	AI Operating mode channel 1	0x01
2	Unsigned8	ro	AI Operating mode channel 2	0x01
3	Unsigned8	ro	AI Operating mode channel 3	0x01
4	Unsigned8	ro	AI Operating mode channel 4	0x01
5	Unsigned8	ro	AI Operating mode channel 5	0x01
6	Unsigned8	ro	AI Operating mode channel 6	0x01
7	Unsigned8	ro	AI Operating mode channel 7	0x01
8	Unsigned8	ro	AI Operating mode channel 8	0x01

Table : Index 6112h

This object is read-only on sub-index 0 and read/write on sub-index 1 to 8. Any other access will result in an error.
The operating mode has two states: Normal operation of channel specified by Sub-index (=1) or channel off (=0).

Index 6114h: AI ADC Sample Rate

This object represents the current sample period of one ADC conversion.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08

Sub-Index	Data Type	Access	Description	Default Value
1	Unsigned32	ro	AI ADC sample rate channel 1	0x00000015
2	Unsigned32	ro	AI ADC sample rate channel 2	0x00000015
3	Unsigned32	ro	AI ADC sample rate channel 3	0x00000015
4	Unsigned32	ro	AI ADC sample rate channel 4	0x00000015
5	Unsigned32	ro	AI ADC sample rate channel 5	0x00000015
6	Unsigned32	ro	AI ADC sample rate channel 6	0x00000015
7	Unsigned32	ro	AI ADC sample rate channel 7	0x00000015
8	Unsigned32	ro	AI ADC sample rate channel 8	0x00000015

Table : Index 6114h

This object is read-only on sub-index 0 to 8. Any other access will result in an error.
Caution: The sample rate of the ADC is set to a constant value of 48000Hz, which corresponds to a time period of 20.83 µs between two samples. The nearest integer value (21d = 15h) is used in this object.

Index 6125h: AI Autozero

Writing the signature value 0x6f72657a (“zero”) to this object will modify the input offset in such a way that the AI Input Process Value (**6130h**) and the AI Input Field Value (**9130h**) both become Zero.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Unsigned32	wo	AI Autozero channel 1	-
2	Unsigned32	wo	AI Autozero channel 2	-
3	Unsigned32	wo	AI Autozero channel 3	-
4	Unsigned32	wo	AI Autozero channel 4	-
5	Unsigned32	wo	AI Autozero channel 5	-
6	Unsigned32	wo	AI Autozero channel 6	-
7	Unsigned32	wo	AI Autozero channel 7	-
8	Unsigned32	wo	AI Autozero channel 8	-

Table : Index 6125h

This object is read-only on sub-index 0 and write-only on sub-index 1 to 8. Any other access will result in an error.
Caution: The AI Scaling Offset (**6127h**) is not accounted for during the Autozero operation.

Index 6126h: AI Scaling factor

This object represents the factor by which the Field Value is scaled in the calculation to create the Process Value (see **6130h**).

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Float	rw	AI Scaling factor channel 1	2.0
2	Float	rw	AI Scaling factor channel 2	2.0
3	Float	rw	AI Scaling factor channel 3	2.0
4	Float	rw	AI Scaling factor channel 4	2.0
5	Float	rw	AI Scaling factor channel 5	2.0
6	Float	rw	AI Scaling factor channel 6	2.0
7	Float	rw	AI Scaling factor channel 7	2.0
8	Float	rw	AI Scaling factor channel 8	2.0

Table : Index 6126h

This object is read-only on sub-index 0 and read/write on sub-index 1 to 8. Any other access will result in an error.

Index 6127h: AI Scaling offset

This object represents the offset that is added to the scaled field value to create the process value (see **6130h**).

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Float	rw	AI Scaling offset channel 1	0.0
2	Float	rw	AI Scaling offset channel 2	0.0
3	Float	rw	AI Scaling offset channel 3	0.0
4	Float	rw	AI Scaling offset channel 4	0.0
5	Float	rw	AI Scaling offset channel 5	0.0
6	Float	rw	AI Scaling offset channel 6	0.0
7	Float	rw	AI Scaling offset channel 7	0.0

Sub-Index	Data Type	Access	Description	Default Value
8	Float	rw	AI Scaling offset channel 8	0.0

Table : Index 6127h

This object is read-only on sub-index 0 and read/write on sub-index 1 to 8. Any other access will result in an error.

Index 6130h: AI Process Value

This object represents the measuring values for each analogue input channel, which are processed AI Field Values. It is the result of the following Equation:

AIProcessValue = (((AIFieldValue * 1.05) / 8388608) * AIScalingFactor) + AIScalingOffset

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Float	ro	AI Process Value channel 1	-
2	Float	ro	AI Process Value channel 2	-
3	Float	ro	AI Process Value channel 3	-
4	Float	ro	AI Process Value channel 4	-
5	Float	ro	AI Process Value channel 5	-
6	Float	ro	AI Process Value channel 6	-
7	Float	ro	AI Process Value channel 7	-
8	Float	ro	AI Process Value channel 8	-

Table : Index 6130h

This object is read-only on sub-index 0 to 8. Any other access will result in an error.
This object on sub-index 1 to 8 can be mapped in the TxPDO (**1A00h**).

Index 6148h: AI Span start

This object specifies the lower limit of the process value (**6130h**). If a process value is equal or lower than this limit, the negative overload and invalid flag is set (**6150h**).

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Float	rw	AI Span start channel 1	-2.1
2	Float	rw	-2.1	-2.1
3	Float	rw	AI Span start channel 3	-2.1
4	Float	rw	AI Span start channel 4	-2.1

Sub-Index	Data Type	Access	Description	Default Value
5	Float	rw	AI Span start channel 5	-2.1
6	Float	rw	AI Span start channel 6	-2.1
7	Float	rw	AI Span start channel 7	-2.1
8	Float	rw	AI Span start channel 8	-2.1

Table : Index 6148h

This object is read-only on sub-index 0 and read/write on sub-index 1 to 8. Any other access will result in an error.
The default is such that no error will occur with the other default settings.

Index 6149h: AI Span end

This object specifies the upper limit to the process value (**6130h**). If a process value is higher or equal than this limit, the positive overload and invalid flag is set (**6150h**).

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Float	rw	AI Span end channel 1	2.1
2	Float	rw	AI Span end channel 2	2.1
3	Float	rw	AI Span end channel 3	2.1
4	Float	rw	AI Span end channel 4	2.1
5	Float	rw	AI Span end channel 5	2.1
6	Float	rw	AI Span end channel 6	2.1
7	Float	rw	AI Span end channel 7	2.1
8	Float	rw	AI Span end channel 8	2.1

Table : Index 6149h

This object is read-only on sub-index 0 and read/write on sub-index 1 to 8. Any other access will result in an error.
The default is such that no error will occur with the other default settings.

Index 6150h: AI Status

This object reflects the status of each Input Channel.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Unsigned8	ro	AI Status channel 1	0x00
2	Unsigned8	ro	AI Status channel 2	0x00
3	Unsigned8	ro	AI Status channel 3	0x00
4	Unsigned8	ro	AI Status channel 4	0x00
5	Unsigned8	ro	AI Status channel 5	0x00
6	Unsigned8	ro	AI Status channel 6	0x00
7	Unsigned8	ro	AI Status channel 7	0x00
8	Unsigned8	ro	AI Status channel 8	0x00

Table : Index 6150h

This object is read-only on sub-index 0 to 8. Any other Access will result in an Error.

Each entry has the following bits defined:

Bits 7:3	Bit 2	Bit 1	Bit 0
reserved	negative overload	positive overload	not valid

Table : Sub-Index Bit Mapping for Index 6150h

Bit 1 and 2 are mutual exclusive.
This object on sub-index 1 to 8 can be mapped in the TxPDO (**1A00h**).

Index 61A0h: AI Filter type

This object defines the filter used for calculating the field value (**9100h**). The only implemented filter is a 'Repeating Average' filter (Filter Type 2), that is always active.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Unsigned8	ro	AI Filter type channel 1	0x02
2	Unsigned8	ro	AI Filter type channel 2	0x02
3	Unsigned8	ro	AI Filter type channel 3	0x02
4	Unsigned8	ro	AI Filter type channel 4	0x02
5	Unsigned8	ro	AI Filter type channel 5	0x02
6	Unsigned8	ro	AI Filter type channel 6	0x02
7	Unsigned8	ro	AI Filter type channel 7	0x02
8	Unsigned8	ro	AI Filter type channel 8	0x02

Table : Index 61A0h

This object is read-only on sub-index 0 to 8. Any other access will result in an error.

Index 61A1h: AI Filter constant

This object defines the number of samples used by the average filter (see **9100h**).

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Unsigned16	rw	AI Filter constant channel 1	0x12C0
2	Unsigned16	rw	AI Filter constant channel 2	0x12C0
3	Unsigned16	rw	AI Filter constant channel 3	0x12C0
4	Unsigned16	rw	AI Filter constant channel 4	0x12C0
5	Unsigned16	rw	AI Filter constant channel 5	0x12C0
6	Unsigned16	rw	AI Filter constant channel 6	0x12C0

Sub-Index	Data Type	Access	Description	Default Value
7	Unsigned16	rw	AI Filter constant channel 7	0x12C0
8	Unsigned16	rw	AI Filter constant channel 8	0x12C0

Table : Index 61A1h

This object is read-only on sub-index 0 and read/write on sub-index 1 to 8. Any other access will result in an error.
Only integers from 1 to 48000 are allowed as filter constants. Trying to set another value will result in an error and no change will be made to this setting.
Caution: Any write will set all entries to the exact same value.
The default value is 4800d = 12C0h and thus a resulting default data rate of 10Hz.

Index 9100h: AI Field Value

This object represents the filtered ADC values in a twos complement 32Bit integer value for each input channel.

The field value of a channel is calculated by averaging N values from the corresponding ADC-channel. The constant N in the following equation is the value of the AI filter constant (**61A1h**).

The input buffer is reset after the field value was calculated.

Sub-Index	Data Type	Access	Description	Default Value
0	Unsigned8	ro	Highest sub-index	0x08
1	Integer32	ro	AI Field Value channel 1	-
2	Integer32	ro	AI Field Value channel 2	-
3	Integer32	ro	AI Field Value channel 3	-
4	Integer32	ro	AI Field Value channel 4	-
5	Integer32	ro	AI Field Value channel 5	-
6	Integer32	ro	AI Field Value channel 6	-
7	Integer32	ro	AI Field Value channel 7	-
8	Integer32	ro	AI Field Value channel 8	-

Table : Index 9100h

This object is read-only on sub-index 0 to 8. Any other access will result in an error.
This object on sub-index 1 to 8 can be mapped in the TxPDO (**1A00h**).

Subject to modifications

All information herein describes our products in a general manner. They mean no assurance of properties in terms of §459 Abs. 2 BGB and will not cause any liabilities.

Trademarks and Patents

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff

Automation GmbH, Germany. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Made in Germany