INC, LLC.
CANopen
Operator Interface
Software Specifications
(Released)

Revision	Description	Date
0.0	Initial Specification	12/12/2002
0.1	Additional Features Added	04/01/2003
1.1B	Specs. tentatively agreed to	06/13/2003

Table of Contents

Overview:

The INC CANopen Operator Interface allows an operator to send and receive information on the CANopen network. The operator interface contains a photocell and thermal sensor for measuring the current light intensity and temperature of the surrounding area. These values can be used to automatically control the contrast and backlighting of the display in environments where the conditions may frequently change such as in a vehicle. The display contains objects that can be used as building blocks for the construction of user interface screens. These include static text, multilingual text, operator input blocks, operator output blocks, etc. This allows the display to be configured with pre-programmed user screens and the network can simply select the screen to display. This pre-configuration saves the cumbersome task of sending text messages across a network. The display also has 254 variables of each supported data type (UNSIGNED8, UNSIGNED16, UNSIGNED32, SIGNED8, SIGNED16, SIGNED32, REAL32) available for operator input and for operator output. The variables can be used by the operator and or the network. Other features also include a piezo element for alerting the user to warnings and errors. The display includes a Windows configuration tool that allows all screens to be constructed and tested in a Windows environment. The configuration can then be downloaded to the display using a USB connection, IRDA, or one of several CANopen interface cards. The operator interface can also be configured using the included EDS file or through SDO commands. The operator interface can support up to 8 TxPDOs and 8 RxPDOS. The first RxPDO pair defaults to being used to select the screen and the first TxPDO to getting its status. The Second RxPDO pair defaults to being used to send infrequently used variables and the second TxPdo to receiving them. The other PDOs default to the mapping of DataIn and DataOut variable. Of course, all of the PDOs are dynamically mapped so the user can configure them as they wish.

Physical

Network Switch Settings:

SW1 and SW2

Node ID	Meaning
0x00	Re-Configure Module (Does Not got Online)
0x01-0x7F	Use this Node Address
0x80-0xFF	Use value stored in NV

SW3

Baud Rate	Meaning
0x0	Auto-Baud
0x1	10k bps
0x2	20k bps
0x3	50k bps
0x4	125k bps
0x5	250k bps
0x6	500k bps
0x7	800k bps
0x8	1M bps
0x9-0xF	Use value stored in NV

Bit Timing Settings

Bit rate Bus length (1)	Nominal bit time t_b	Number of time quanta per bit	Length of time quantum t_q	Location of sample point
1 Mbit/s 25 m	1 us	8	125 ns	6 t_q(750 ns)
800 kbit/s 50 m	1,25 us	10	125 ns	8 t_q(1 us)
500 kbit/s 100 m	2 us	16	125 ns	14 t_q(1,75 us)
250 kbit/s 250 m(2)	4 us	16	250 ns	14 t_q(3,5 us)
125 kbit/s 500 m(2)	8 us	16	500 ns	14 t_q(7 us)
50 kbit/s 1000 m(3)	20 us	16	1,25 us	14 t_q(17,5 us)
20 kbit/s 2500 m(3)	50 us	16	3,125 us	14 t_q(43,75 us)
10 kbit/s 5000 m(3)	100 us	16	6,25 us	14 t_q(87,5 us)

LED Indicators:

The Operator interface has two bi-color LEDs, CAN and MOD, which indicate the status of the module.

Supported Codes
Supported SDO abort codes

Abort code	Description
0503 0000h	Toggle bit not alternated.
0504 0000h	SDO protocol timed out.
0504 0001h	Client/server command specifier not valid or unknown.
0504 0002h	Invalid block size (block mode only).
0504 0003h	Invalid sequence number (block mode only).
0504 0004h	CRC error (block mode only).
0504 0005h	Out of memory.
0601 0000h	Unsupported access to an object.
0601 0001h	Attempt to read a write only object.
0601 0002h	Attempt to write a read only object.
0602 0000h	Object does not exist in the object dictionary.
0604 0041h	Object cannot be mapped to the PDO.
0604 0042h	The number and length of the objects to be mapped would exceed PDO length.
0604 0043h	General parameter incompatibility reason.
0604 0047h	General internal incompatibility in the device.
0606 0000h	Access failed due to an hardware error.
0607 0010h	Data type does not match, length of service parameter does not match
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0011h	Sub-index does not exist.

The abort codes not listed here are reserved.

Supported Emergency Error Codes

Error Code (hex)	Meaning
00xx	Error Reset or No Error
10xx	Generic Error
81xx	Communication
8110	CAN Overrun (Objects lost)
8120	CAN in Error Passive Mode
8130	Life Guard Error or Heartbeat Error
8140	Recovered from bus off
8150	Transmit COB-ID
82xx	Protocol Error
8210	PDO not processed due to length error
8220	PDO length exceeded
FFxx	Device specific

Emergency Object Data

The Emergency Telegram consists of 8 bytes with the data as shown in

Emergency Object

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code (see Table)		Error register (Object 1001H)	TBD	TBD	TBD		

Pre-defined connection set

Bit Number: COB-Identifier

10	9	8	7	6	5	4	3	2	1	0
Function Code				Node-ID						

Identifier allocation scheme for the pre-defined connection set

Broadcast Objects of the Pre-defined Connection Set

Object	function code(binary)	resulting COB-ID	Communication Parameters at Index
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h,1006h,1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

Peer-to-Peer Objects of the Pre-defined Connection Set

object	function code (binary)	Resulting COB-IDs	Communication Parameters at Index
EMERGENCY	0001	129 (81h) - 255 (FFh)	1014h, 1015h
PDO1 (tx)	0011	385 (181h) - 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) - 639 (27Fh)	1400h
PDO2 (tx)	0101	641 (281h) - 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) - 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) - 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) - 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) - 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) - 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) - 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) - 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) - 1919 (77Fh)	1016h, 1017h

Seen from the device's point of view.
The pre-defined connection set always applies to the standard CAN frame with 11-bit Identifier, even if extended CAN frames are present in the network.

Assigning COB-IDS

When Assigning COB-Ids to SDO and PDO objects, the user should use care. If the user selects a COB-ID for a PDO that is already assigned to the display's SDO by the pre-defined connection set, the display will accept the COB-ID, but the module may not function as the user desires. The COB-ID acceptance is done in order to provide greater flexibility for the advanced user.

Object Dictionary Data Types

Index	Object	Name
0001	DEFTYPE	BOOLEAN
0002	DEFTYPE	INTEGER8
0003	DEFTYPE	INTEGER16
0004	DEFTYPE	INTEGER32
0005	DEFTYPE	UNSIGNED8
0006	DEFTYPE	UNSIGNED16
0007	DEFTYPE	UNSIGNED32
0008	DEFTYPE	REAL32
0009	DEFTYPE	VISIBLE_STRING
000A	DEFTYPE	OCTET_STRING
000B	DEFTYPE	UNICODE_STRING
000C	DEFTYPE	TIME_OF_DAY
000D	DEFTYPE	TIME_DIFFERENCE
000E	reserved	-
000F	DEFTYPE	DOMAIN
0010	DEFTYPE	INTEGER24
0011	DEFTYPE	REAL64
0012	DEFTYPE	INTEGER40
0013	DEFTYPE	INTEGER48
0014	DEFTYPE	INTEGER56
0015	DEFTYPE	INTEGER64
0016	DEFTYPE	UNSIGNED24
0017	reserved	-
0018	DEFTYPE	UNSIGNED40
0019	DEFTYPE	UNSIGNED48
001A	DEFTYPE	UNSIGNED56
001B	DEFTYPE	UNSIGNED64
001C-001F	reserved	-
0020	DEFSTRUCT	PDO_COMMUNICATION_PARAMETER
0021	DEFSTRUCT	PDO_MAPPING
0022	DEFSTRUCT	SDO_PARAMETER
0023	DEFSTRUCT	IDENTITY
0024-003F	reserved	-
0040-005F	DEFSTRUCT	Manufacturer Specific Complex Data Types
0060-007F	DEFTYPE	Device Profile (0) Specific Standard Data Types
0080-009F	DEFSTRUCT	Device Profile (0) Specific Complex Data Types
00A0-00BF	DEFTYPE	Device Profile 1 Specific Standard Data Types
00C0-00DF	DEFSTRUCT	Device Profile 1 Specific Complex Data Types
00E0-00FF	DEFTYPE	Device Profile 2 Specific Standard Data Types
0100-011F	DEFSTRUCT	Device Profile 2 Specific Complex Data Types
0120-013F	DEFTYPE	Device Profile 3 Specific Standard Data Types
0140-015F	DEFSTRUCT	Device Profile 3 Specific Complex Data Types
0160-017F	DEFTYPE	Device Profile 4 Specific Standard Data Types
0180-019F	DEFSTRUCT	Device Profile 4 Specific Complex Data Types
01A0-01BF	DEFTYPE	Device Profile 5 Specific Standard Data Types
01C0-01DF	DEFSTRUCT	Device Profile 5 Specific Complex Data Types
01E0-01FF	DEFTYPE	Device Profile 6 Specific Standard Data Types
0200-021F	DEFSTRUCT	Device Profile 6 Specific Complex Data Types
0220-023F	DEFTYPE	Device Profile 7 Specific Standard Data Types
0240-025F	DEFSTRUCT	Device Profile 7 Specific Complex Data Types

Object Dictionary Structure

Index (hex)	Object
0000	not used
0001-001F	Static Data Types
0020-003F	Complex Data Types
0040-005F	Manufacturer Specific Complex Data Types
0060-007F	Device Profile Specific Static Data Types
0080-009F	Device Profile Specific Complex Data Types
00A0-0FFF	Reserved for further use
1000-1FFF	Communication Profile Area
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardized Device Profile Area
A000-FFFF	Reserved for further use

Object Dictionary Overview

Object Dictionary Entries for Communication

Standard Objects

Index(hex)	Object (Symbolic Name)	Name	Type	Acc. 1	M/O
1000	VAR	device type	UNSIGNED32	ro	M
1001	VAR	error register	UNSIGNED8	ro	M
1002	VAR	Manufacturer status register	UNSIGNED32	ro	O
1003	VAR	Pre-defined error field	UNSIGNED32	Ro	O
1005	VAR	COB-ID SYNC	UNSIGNED32	rw	O
1008	VAR	Manufacturer device name	Vis-String	const	O
1009	VAR	Manufacturer hardware version	Vis-String	const	O
100A	VAR	Manufacturer software version	Vis-String	const	O
100C	VAR	guard time	UNSIGNED16	rw	O
100D	VAR	life time factor	UNSIGNED8	rw	O
1010	ARRAY	store parameters	UNSIGNED32	rw	O
1011	ARRAY	restore default parameters	UNSIGNED32	rw	O
1012	VAR	COB-ID TIME	UNSIGNED32	rw	O
1014	VAR	COB-ID EMCY	UNSIGNED32	rw	O
1015	VAR	Inhibit Time EMCY	UNSIGNED16	rw	O
1016	ARRAY	Consumer heartbeat time	UNSIGNED32	RW	O
1017	VAR	Producer heartbeat time	UNSIGNED16	rw	O
1018	RECORD	Identity Object	Identity (23h)	ro	M
1020	ARRAY	Verify Configuration	UNSIGNED32	rw	O
1029	ARRAY	Error Behavior	UNSIGNED8	rw	O

Server SDO Parameter

1200	RECORD	1st Server SDO parameter	SDO Parameter (22h)	ro	O
1201	RECORD	2st Server SDO parameter	SDO Parameter (22h)	rw	O

Client SDO Parameter

Not Supported

Receive PDO Communication Parameter

1400	RECORD	1st receive PDO Parameter	PDO CommPar (20h)	rw	M/O*
1401	RECORD	2nd receive PDO Parameter	PDO CommPar (20h)	rw	M/O*
:::::	:::::	:::::	:::::	:::::	:::::
141F	RECORD	32nd receive PDO Parameter	PDO CommPar (20h)	rw	M/O*

Receive PDO Mapping Parameter

1600	RECORD	1st receive PDO mapping	PDO Mapping (21h)	rw	M/O*
1601	RECORD	2nd receive PDO mapping	PDO Mapping (21h)	rw	M/O*
:::::	:::::	:::::	:::::	:::::	:::::
161F	RECORD	32nd receive PDO mapping	PDO Mapping (21h)	rw	M/O*

Transmit PDO Communication Parameter

1800	RECORD	1st transmit PDO Parameter	PDO CommPar (20h)	rw	M/O*
1801	RECORD	2nd transmit PDO Parameter	PDO CommPar (20h)	rw	M/O*
:::::	:::::	:::::	:::::	:::::	::::
181F	RECORD	32nd transmit PDO Parameter	PDO CommPar (20h)	rw	M/O*

Transmit PDO Mapping Parameter

1A00	RECORD	1st transmit PDO mapping	PDO Mapping (21h)	rw	M/O*
1A01	RECORD	2nd transmit PDO mapping	PDO Mapping (21h)	rw	M/O*
:::::	:::::	:::::	:::::	:::::	:::::
1A1F	RECORD	32nd transmit PDO mapping	PDO Mapping (21h)	rw	M/O*

Ranges 1600-161Fh and 1A00-1A1Fh can also used to map multiplexed PDOs. See specification below.

Detailed Specification of Communication Profile Specific Objects

Object 1000h: Device Type

Contains information about the device type. The object at index 1000h describes the type of device and its functionality. It is composed of a 16-bit field which describes the device profile that is used and a second 16-bit field which gives additional information about optional functionality of the device. The Additional Information parameter is device profile specific.

OBJECT DESCRIPTION

INDEX	1000h
Name	device type
Object Code	VAR
Data Type	UNSIGNED32
Category	Mandatory

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Byte:

MSB	LSB
Additional Information	Device Profile Number
0x0000	0x0000

Object 1001h: Error Register

This object is an error register for the device. The device can map internal errors in this byte. This entry is mandatory for all devices. It is a part of an Emergency object.

OBJECT DESCRIPTION

INDEX	1001h
Name	error register
Object Code	VAR
Data Type	UNSIGNED8
Category	Mandatory

ENTRY DESCRIPTION

Access	ro
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

Structure of the Error Register

Bit	M/O	Supported	Meaning
0	M	Yes	generic error
1	O	No	Current
2	O	Yes	Voltage
3	O	No	Temperature
4	O	Yes	communication error (overrun, error state)
5	O	No	device profile specific
6	O	No	Reserved (always 0)
7	O	Yes	manufacturer specific

If a bit is set to 1 the specified error has occurred. The generic error is signaled at any error situation.

Object 1002h: Manufacturer Status Register

This object is a common status register for manufacturer specific purposes. In this document only the
size and the location of this object is defined.

OBJECT DESCRIPTION

INDEX	1002h
Name	manufacturer status register
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Access	ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

MSB LSB

Object 1003h: Pre-defined Error Field

The object at index 1003h holds the errors that have occurred on the device and have been signaled via the Emergency Object. In doing so it provides an error history.
1. The entry at sub-index 0 contains the number of actual errors that are recorded in the array starting at sub-index 1.
2. Every new error is stored at sub-index 1, the older ones move down the list.
3. Writing a “0” to sub-index 0 deletes the entire error history (empties the array). Values higher than 0 are not allowed to write. This have to lead to an abort message (error code: 0609 0030h).
4. The error numbers are of type UNSIGNED32 (see Table 7-18) and are composed of a 16 bit error code and a 16 bit additional error information field which is manufacturer specific. The error code is contained in the lower 2 bytes (LSB) and the additional information is included in the upper 2 bytes (MSB). The additional information consists of the information contained in the Error register Object (1001h) The length entry on sub-index 0h and at least one error entry at sub-index 1H.

Byte: MSB LSB

Additional Information	Error code

Structure of the pre-defined error field

OBJECT DESCRIPTION

INDEX	1003h
Name	pre-defined error field
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	number of errors
Entry Category	Mandatory
Access	rw
PDO Mapping	No
Value Range	0 - 10
Default Value	0

Sub-Index	1h
Description	standard error field
Entry Category	Optional
Access	ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Sub-Index	2h – 10
Description	standard error field
Entry Category	Optional
Access	ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Object 1005h: COB-ID SYNC message

Index 1005h defines the COB-ID of the Synchronization Object (SYNC). Further, it defines whether the device generates the SYNC.

UNSIGNED32
MSB LSB

bits	*31*	*30*	*29*	*28-11*	*10-0*
11-bit-ID	X	0/1	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	X	0/1	1	29 -bit Identifier	

Structure of SYNC COB-ID entry

Description of SYNC COB-ID entry

bit number	value	meaning
31 (MSB)	X	do not care
30	0 1	Device does not generate SYNC message Device generates SYNC message
29	0 1	11-bit ID (CAN 2.0A) 29-bit ID (CAN 2.0B)
28 – 11	0 X	if bit 29=0 if bit 29=1: bits 28-11 of 29-bit-SYNC-COB-ID
10-0 (LSB)	X	bits 10-0 of SYNC-COB-ID

Bits 29, 30 may be static (not changeable). If a device is not able to generate SYNC messages, an attempt to set bit 30 is responded with an abort message (abort code: 0609 0030h). Devices supporting the standard CAN frame type only either ignore attempts to change bit 29 or respond with an abort message (abort code: 0609 0030h). The first transmission of SYNC object starts within 1 sync cycle after setting Bit 30 to 1. It is not allowed to change Bit 0-29, while the objects exist (Bit 30=1).

OBJECT DESCRIPTION

INDEX	1005h
Name	COB-ID SYNC
Object Code	VAR
Data Type	UNSIGNED32
Category	Conditional;Mandatory, if PDO communication on a synchronous base is supported

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	80h or 8000 0080h

Object 1008h: Manufacturer Device Name

Contains the manufacturer device name.

OBJECT DESCRIPTION

INDEX	1008h
Name	manufacturer device name
Object Code	VAR
Data Type	Visible String
Category	Optional

ENTRY DESCRIPTION

Access	const
PDO Mapping	No
Value Range	No
Default Value	CANopen Display

Object 1009h: Manufacturer Hardware Version

Contains the manufacturer hardware version description.

OBJECT DESCRIPTION

INDEX	1009h
Name	manufacturer hardware version
Object Code	VAR
Data Type	Visible String
Category	Optional

ENTRY DESCRIPTION

Access	const
PDO Mapping	No
Value Range	No
Default Value	01

Object 100Ah: Manufacturer Software Version

Contains the manufacturer software version description.

OBJECT DESCRIPTION

INDEX	100Ah
Name	Manufacturer software version
Object Code	VAR
Data Type	Visible String
Category	Optional

ENTRY DESCRIPTION

Access	Const
PDO Mapping	No
Value Range	No
Default Value	1.01

Object 100Ch: Guard Time

The objects at index 100Ch and 100Dh include the guard time in milliseconds and the life time factor.
The life time factor multiplied with the guard time gives the life time for the Life Guarding Protocol. It is 0 if not used.

OBJECT DESCRIPTION

INDEX	100Ch
Name	guard time
Object Code	VAR
Data Type	UNSIGNED16
Category	Conditional; Mandatory, if heartbeat is not supported

ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

Object 100Dh: Life Time Factor

The life time factor multiplied with the guard time gives the life time for the node guarding protocol. It is 0 if not used.

OBJECT DESCRIPTION

INDEX	100Dh
Name	life time factor
Object Code	VAR
Data Type	UNSIGNED8
Category	Conditional; Mandatory, if heartbeat is not supported

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Object 1010h: Store parameters

This object supports the saving of parameters in non volatile memory. By read access the device provides information about its saving capabilities. Several parameter groups are distinguished:
Sub-Index 0 contains the largest Sub-Index that is supported.
Sub-Index 1 refers to all parameters that can be stored on the device.
Sub-Index 2 refers to communication related parameters (Index 1000h - 1FFFh manufacturer specific communication parameters).
Sub-Index 3 refers to application related parameters (Index 6000h - 9FFFh manufacturer specific application parameters).
At Sub-Index 4 - 127 manufacturers may store their choice of parameters individually.
Sub-Index 128 - 254 are reserved for future use.
In order to avoid storage of parameters by mistake, storage is only executed when a specific signatureis written to the appropriate Sub-Index. The signature is “save”.

Signature MSB LSB
ISO 8859
(“ASCII”)
hex

E	V	a	s
65h	76h	61h	73h

Storage write access signature

On reception of the correct signature in the appropriate sub-index the device stores the parameter and then confirms the SDO transmission (initiate download response). If the storing failed, the device responds with an Abort SDO Transfer (abort code: 0606 0000h).
If a wrong signature is written, the device refuses to store and responds with Abort SDO Transfer (abort code: 0800 002xh).
On read access to the appropriate Sub-Index the device provides information about its storage functionality with the following format:

UNSIGNED32

MSB		LSB
31-2	*1*	*0*
reserved (=0)	0/1	0/1

Storage read access structure

Structure of read access

bit number	Value	meaning
31-2	0	*reserved (=0)*
1	0	Device does not save parameters autonomously
	1	Device saves parameters autonomously
0	0	Device does not save parameters on command
	1	Device saves parameters on command

Autonomous saving means that a device stores the storable parameters in a non-volatile manner without user request.

OBJECT DESCRIPTION

INDEX	1010h
Name	store parameters
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	largest subindex supported
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1h - 7Fh
Default Value	4

Sub-Index	1h
Description	Save all parameters
Entry Category	Mandatory
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 55 for write access; Figure 56 for read access)
Default Value	01h

Sub-Index	2h
Description	save communication parameters
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 55 for write access; Figure 56 for read access)
Default Value	01h

Sub-Index	3h
Description	Save application parameters
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 55 for write access; Figure 56 for read access)
Default Value	02h

Sub-Index	4h
Description	Save I/O Configuration and Node Address
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 55 for write access; Figure 56 for read access)
Default Value	01h

Note Application parameters are stored autonomously. Communication parameters must be manually stored.

Object 1011h: Restore default parameters

With this object the default values of parameters according to the communication or device profile are restored. By read access the device provides information about its capabilities to restore these values.
Several parameter groups are distinguished:
Sub-Index 0 contains the largest Sub-Index that is supported.
Sub-Index 1 refers to all parameters that can be restored.
Sub-Index 2 refers to communication related parameters (Index 1000h - 1FFFh manufacturer specific communication parameters).
Sub-Index 3 refers to application related parameters (Index 6000h - 9FFFh manufacturer specific application parameters).
At Sub-Index 4 - 127 manufacturers may restore their individual choice of parameters.
Sub-Index 128 - 254 are reserved for future use.
In order to avoid the restoring of default parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-index. The signature is “load”.

Signature MSB LSB
ASCII
Hex

d	a	o	l
64h	61h	6Fh	6Ch

Figure 57: Restoring write access signature

On reception of the correct signature in the appropriate sub-index the device restores the default parameters and then confirms the SDO transmission (initiate download response). If the restoring failed, the device responds with an Abort SDO Transfer (abort code: 0606 0000h). If a wrong signature is written, the device refuses to restore the defaults and responds with an Abort SDO Transfer (abort code: 0800 002xh).
The default values are set valid after the device is reset (reset node for sub-index 1h - 7Fh, reset communication for sub-index 2h) or power cycled.
On read access to the appropriate sub-index the device provides information about its default parameter restoring capability with the following format:
UNSIGNED32

MSB	LSB
31-1	*0*
Reserved (=0)	0/1

Figure 59: Restoring default values read access structure

Table 50: Structure of restore read access

bit number	value	meaning
31-1	0	*reserved (=0)*
0	0	Device does not restore default parameters
	1	Device restores parameters

OBJECT DESCRIPTION

Index	1011h
Name	restore default parameters
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	largest subindex supported
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1h- 7Fh
Default Value	4

Sub-Index	1h
Description r	Restore all default parameters
Entry Category	Mandatory
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 57)
Default Value	01h

Sub-Index	2h
Description	Restore communication default parameters
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 57)
Default Value	01h

Sub-Index	3h
Description	Restore application default parameters
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 57)
Default Value	01h

Sub-Index	4h
Description	Restore Manufacturer Specific parameters
Entry Category	Optional
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 57)
Default Value	01h

The number and type of physically attached modules control most of the manufacturer specific parameters, therefore Sub-Index 4 is reserved for future use.

Object 1012h: COB-ID Time Stamp Object

Index 1012h defines the COB-ID of the Time-Stamp Object (TIME). Further, it defines whether the device consumes the TIME or whether the device generates the TIME. The structure of this object is shown in Figure 60 and Table 51.

UNSIGNED32
MSB LSB

bits	*31*	*30*	*29*	*28-11*	*10-0*
11-bit-ID	0/1	0/1	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	0/1	0/1	1	29 -bit Identifier	

Figure 60: Structure of TIME COB-ID entry

Table 51: Description of TIME COB-ID entry

bit number	value	meaning
31 (MSB)	0	Device does not consume TIME message
	1	Device consumes TIME message
30	0	Device does not produce TIME message
	1	Device produces TIME message
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28 – 11	0	if bit 29=0
	X	if bit 29=1: bits 28-11 of 29-bit-TIME-COB-ID
10-0 (LSB)	X	bits 10-0 of TIME-COB-ID

Bits 29, 30 may be static (not changeable). If a device is not able to generate TIME messages, an attempt to set bit 30 is responded with an abort message (abort code: 0609 0030h). Devices supporting the standard CAN frame type only, an attempt to set bit 29 is responded with an abort message (abort code: 0609 0030h). It is not allowed to change Bits 0-29, while the object exists (Bit 30=1).

OBJECT DESCRIPTION

INDEX	1012h
Name	COB-ID time stamp message
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	100h

Object 1014h: COB-ID Emergency Object

Index 1014h defines the COB-ID of the Emergency Object (EMCY). The structure of this object is shown in Figure 61.

UNSIGNED32

MSB LSB

Bits	*31*	*30*	*29*	*28-11*	*10-0*
11-bit-ID	0/1	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	0/1	0	1	29 -bit Identifier	

Figure 61: Structure of the EMCY Identifier entry

Table 52: Description of EMCY COB-ID entry

bit number	value	Meaning
31 (MSB)	0	EMCY exists / is valid
	1	EMCY does not exist / is not valid
30	0	reserved (always 0)
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28 - 11	0	if bit 29=0
	X	if bit 29=1: bits 28-11 of 29-bit-COB-ID
10-0 (LSB)	X	bits 10-0 of COB-ID

Devices supporting the standard CAN frame type only, an attempt to set bit 29 is responded with an abort message (abort code: 0609 0030h). It is not allowed to change Bits 0-29, while the object exists (Bit 31=0).

OBJECT DESCRIPTION

INDEX	1014h
Name	COB-ID Emergency message
Object Code	VAR
Data Type	UNSIGNED32
Category	Conditional;Mandatory, if Emergency is supported

ENTRY DESCRIPTION

Access	ro;optional rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	80h + Node-ID

Object 1015h: Inhibit Time EMCY

The inhibit time for the EMCY message can be adjusted via this entry. If this entry exists it must be writeable in the object dictionary. The time has to be a multiple of 100ms.

OBJECT DESCRIPTION

INDEX	1015h
Name	Inhibit Time EMCY
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

Object 1016h: Consumer Heartbeat Time

The consumer heartbeat time defines the expected heartbeat cycle time and thus has to be higher than the corresponding producer heartbeat time configured on the device producing this heartbeat.
Monitoring starts after the reception of the first heartbeat. If the consumer heartbeat time is 0 the corresponding entry is not used. The time has to be a multiple of 1ms.

UNSIGNED32
MSB LSB

Bits	**31-24**	**23-16**	**15-0**
Value	reserved (value: 00h)	Node-ID	heartbeat time
Encoded as	-	UNSIGNED8	UNSIGNED16

Table 62: Structure of Consumer Heartbeat Time entry

At an attempt to configure several consumer heartbeat times unequal 0 for the same Node-ID the device aborts the SDO download with abort code 0604 0043h

OBJECT DESCRIPTION

INDEX	1016h
Name	Consumer Heartbeat Time
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	number entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 - 127
Default Value	4

Sub-Index	1h
Description	Consumer Heartbeat Time
Entry Category	Mandatory
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 62)
Default Value	0

Sub-Index	2h - 4h
Description	Consumer Heartbeat Time
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 62)
Default Value	No

Object 1017h: Producer Heartbeat Time

The producer heartbeat time defines the cycle time of the heartbeat. The producer heartbeat time is 0 if it not used. The time has to be a multiple of 1ms.

OBJECT DESCRIPTION

INDEX	1017h
Name	Producer Heartbeat Time
Object Code	VAR
Data Type	UNSIGNED16
Category	Conditional; Mandatory if guarding not supported

ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

Object 1018h: Identity Object

The object at index 1018h contains general information about the device.
The Vendor ID (sub-index 1h) contains a unique value allocated to each manufacturer.
The manufacturer-specific Product code (sub-index 2h) identifies a specific device version.
The manufacturer-specific Revision number (sub-index 3h) consists of a major revision number and a
minor revision number. The major revision number identifies a specific CANopen behaviour. If the CANopen functionality is expanded, the major revision has to be incremented. The minor revision number identifies different versions with the same CANopen behavior.

31 16	15 0
major revision number	minor revision number
MSB	LSB

Structure of Revision number

The manufacturer-specific Serial number (sub-index 4h) identifies a specific device.

OBJECT DESCRIPTION

INDEX	1018h
Name	Identity Object
Object Code	RECORD
Data Type	Identity
Category	Mandatory

ENTRY DESCRIPTION

Sub-Index	0h
Description	number of entries
Entry Category	Mandatory
Access	Ro
PDO Mapping	No
Value Range	1 .. 4
Default Value	4

Sub-Index	1h
Description	Vendor ID
Entry Category	Mandatory
Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0x00000000

Sub-Index	2h
Description	Product code
Entry Category	Optional
Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0x00000001

Sub-Index	3h
Description	Revision number
Entry Category	Optional
Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	0x00010001

Sub-Index	4h
Description	Serial number
Entry Category	Optional
Access	Ro
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	No

Object 1020h: Verify Configuration

If a device supports the saving of parameters in non-volatile memory, a network configuration tool or a CANopen manager can use this object to verify the configuration after a devices reset and to check if a reconfiguration is necessary. The configuration tool shall store the date and time in that object and shall store the same values in the DCF. Now the configuration tool lets the device save its configuration by writing to index 1010h Sub-Index 1h the signature "save". After a reset the device shall restore the last configuration and the signature automatically or by request. If any other command changes boot-up configuration values, the device shall reset the object Verify Configuration to 0.
The Configuration Manager compares signature and configuration with the value from the DCF and decides if a reconfiguration is necessary or not.

Index	Object	Name	Type	Attr.	M/O
1020h	ARRAY	Verify Configuration	Unsigned32	rw	O

The sub-objects for the Verify Configuration Object are:

Index	Sub-Index	Field in Configuration Verify	Data Type
1020h	0h	Number of supported entries	Unsigned8
	1h	Configuration date	Unsigned32
	2h	Configuration time	Unsigned32

Configuration date shall contain the number of days since January 1,1984. Configuration time shall be
the number of ms after midnight.

Application hint: The usage of this object allows a significant speed-up of the boot-up process. If it is used, the system integrator has to consider that a user may change a configuration value and afterwards activate the command store configuration 1010h without changing the value of 1020h. So the system integrator has to ensure a 100% consequent usage of this feature.

Object 1029: Error behavior object

If a serious device failure is detected in Operational State, the module shall enter by default autonomously the pre-operational state. If object 1028h (Error Behavior) is implemented, the device can be configured to enter alternatively the stopped state or remain in the current state in case of a device failure. Device failures shall include the following communication errors:

- Bus-off conditions of the CAN interface
- Life guarding event with the state 'occurred'
- Heartbeat event with state 'occurred'

Serious device errors also can be caused by device internal failures.
The value of the Error Classes is as follows:
0 = pre-operational (only if current state is operational)
1 = no state change
2 = stopped
3 .. 127 = reserved

OBJECT DESCRIPTION

INDEX	1029h
Name	Error Behavior
Object Code	ARRAY
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Sub-Index	0h
Description	No. of Error Classes
Entry Category	Mandatory
Access	Ro
PDO Mapping	No
Value Range	1h
Default Value	1

Sub-Index	1h
Description	Communication Error
Entry Category	Mandatory
Access	rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Object 1200h-1201h: Server SDO Parameter

In order to describe the SDOs used on a device the data type SDO Parameter is introduced. The data type has the index 22h in the Object Dictionary. The structure is described in 9.5.4.
The number of supported entries in the SDO object record is specified at sub-index 0h. The values at 1h and 2h specify the COB-ID for this SDO. Sub-index 3 gives the server of the SDO in case the record describes an SDO for which the device is client and gives the client of the SDO if the record describes an SDO for which the device is server.

UNSIGNED32

MSB LSB

bits	*31*	*30*	*29*	*28-11*	*10-0*
11-bit-ID	0/1	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	0/1	0	1	29-bit Identifier	

Figure 64: Structure of SDO COB-ID entry

Table 53: Description of SDO COB-ID entry

bit number	value	Meaning
31 (MSB)	0	SDO exists / is valid
	1	SDO does not exist / is not valid
30	0	reserved (always 0)
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28 – 11	0	if bit 29=0
	X	if bit 29=1: bits 28-11 of 29-bit-COB-ID
10-0 (LSB)	X	bits 10-0 of COB-ID

An SDO is only valid if both SDO-valid-bits are 0. Devices supporting the standard CAN frame type only, an attempt to set bit 29 is responded with an abort message (abort code: 0609E0030h).
These objects contain the parameters for the SDOs for which the device is the server. If a device handles more than one server SDO the default SDO must be located at index 1200h as the first server SDO. This entry is read only 2 . All additional server SDOs are invalid by default (invalid bit - see Table 53), there description is located at subsequent indicies. It is not allowed to change the COB-ID while the SDO exists.
The description of the Client of the SDO (sub-index 3h) is optional. It is not available for the default SDO (no Sub-index 3h at Index 1200h), as this entry is read only.

OBJECT DESCRIPTION

INDEX	1200-1201h
Name	Server SDO parameter
Object Code	RECORD
Data Type	SDO Parameter
Category	Conditional Index 1200h: Optional Index 1201h - 127Fh: Mandatory for each additionally supported server SDO

ENTRY DESCRIPTION

Sub-Index	0h
Description	number of entries
Entry Category	Mandatory
Access	Ro
PDO Mapping	No
Value Range	Index 1200h: 2 Index 1201h - 127F: 2 – 3
Default Value	No

Sub-Index	1h
Description	COB-ID Client->Server (rx)
Entry Category	Mandatory
Access	Index 1200h: ro, Index 1201h-127Fh: rw
PDO Mapping	No
Value Range	UNSIGNED32 (Table 53)
Default Value	Index 1200h: 600h+Node-ID, Index 1201h-127Fh: No

Sub-Index	2h
Description	COB-ID Server -> Client (tx)
Entry Category	Mandatory
Access	Index 1200h: ro Index 1201-127Fh: rw
PDO Mapping	No
Value Range	UNSIGNED32 (Table 53)
Default Value	Index 1200h: 580h+Node-ID, Index 1201h-127Fh: No

Object 1400h – 141Eh: Receive PDO Communication Parameter

Contains the communication parameters for the PDOs the device is able to receive. The type of the PDO communication parameter (20h) is described in 9.5.4. The sub-index 0h contains the number of valid entries within the communication record. Its value is at least 2. If inhibit time supported the value is 3. At sub-index 1h resides the COB-ID of the PDO. This entry has been defined as UNSIGNED32 in order to cater for 11-bit CAN Identifiers (CAN 2.0A) as well as for 29-bit CAN identifiers (CAN 2.0B).
The entry has to be interpreted as defined in Figure 65 and Table 54.

UNSIGNED32

MSB LSB

bits	*31*	*30*	*29*	*28-11*	*10-0*
11-bit-ID	0/1	0/1	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-bit Identifier
29-bit-ID	0/1	0/1	1	29-bit Identifier	

Figure 65: Structure of PDO COB-ID entry

Table 54: Description of PDO COB-ID entry

bit number	value	Meaning
31 (MSB)	0	PDO exists / is valid
	1	PDO does not exist / is not valid
30	0	RTR allowed on this PDO
	1	no RTR allowed on this PDO
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28 – 11	0	if bit 29=0
	X	if bit 29=1: bits 28-11 of 29-bit-COB-ID
10-0 (LSB)	X	bits 10-0 of COB-ID

The PDO valid/not valid allows to select which PDOs are used in the operational state. There may be PDOs fully configured (e.g. by default) but not used, and therefore set to "not valid" (deleted). The feature is necessary for devices supporting more than 4ÊRPDOs or 4 TPDOs, because each device has only default identifiers for the first four RPDOs/TPDOs. Devices supporting the standard CAN frame type only or do not support Remote Frames, an attempt to set bit 29 to 1 or bit 30 to 0 is responded with an abort message (abort code: 0609Ê0030h).
It is not allowed to change bit 0-29 while the PDO exists (Bit 31=0).
The transmission type (sub-index 2) defines the transmission/reception character of the PDO (see 9.2.1.1). Table 55 describes the usage of this entry. On an attempt to change the value of the transmission type to a value that is not supported by the device an abort message (abort code: 0609E0030h) is generated.

transmission type PDO transmission

-	Cyclic	acyclic	synchronous	asynchronous	RTR only
0	-	X	X	-	-
1-240	X	-	X	-	-
241-251	-reserved -				
252	-	-	X	-	X
253	-	-	-	X	X
254	-	-	-	X	-
255	-	-	-	X	-

Synchronous (transmission types 0-240 and 252) means that the transmission of the PDO shall be related to the SYNC object as described in 9.3. Preferably the devices use the SYNC as a trigger to output or actuate based on the previous synchronous Receive PDO respectively to

update the data transmitted at the following synchronous Transmit PDO. Details of this mechanism depend on the device type and are defined in the device profile if applicable.
Asynchronous means that the transmission of the PDO is not related to the SYNC object.
A transmission type of zero means that the message shall be transmitted synchronously with the SYNC object but not periodically.
A value between 1 and 240 means that the PDO is transferred synchronously and cyclically. The transmission type indicating the number of SYNC which are necessary to trigger PDO transmissions.
Receive PDOs are always triggered by the following SYNC upon receiption of data independent of the transmission types 0 - 240.
The transmission types 252 and 253 mean that the PDO is only transmitted on remote transmission request. At transmission type 252, the data is updated (but not sent) immediately after reception of the SYNC object. At transmission type 253 the data is updated at the reception of the remote transmission request (hardware and software restrictions may apply). These value are only possible for TPDOs.
For TPDOs transmission type 254 means, the application event is manufacturer specific (manufacturer specific part of the Object Dictionary), transmission type 255 means, the application event is defined in the device profile. RPDOs with that type trigger the update of the mapped data with the reception.
Sub-index 3h contains the inhibit time. This time is a minimum interval for PDO transmission. The value is defined as multiple of 100ms. It is not allowed to change the value while the PDO exists (Bit 31 of sub-index 1 is 0).
Sub-index 4h is reserved. It does not have to be implemented, in this case read or write access leads to Abort SDO Transfer (abort code: 0609 0011h).
In mode 254/255 additionally an event time can be used for TPDO. If an event timer exists for a TPDO (value not equal to 0) the elapsed timer is considered to be an event. The event timer elapses as multiple of 1 ms of the entry in sub-index 5h of the TPDO. This event will cause the transmission of this TPDO in addition to otherwise defined events. The occurence of the events set the timer. Independent of the transmission type the RPDO event timer is used recognize the expiration of the RPDO.

OBJECT DESCRIPTION

INDEX	1400h – 141Fh
Name	receive PDO parameter
Object Code	RECORD
Data Type	PDO CommPar
Category	Conditional; Mandatory for each supported PDO

ENTRY DESCRIPTION

Sub-Index	0h
Description	largest sub-index supported
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2 – 5

Sub-Index	1h
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	ro; rw if variable COB-ID is supported
PDO Mapping	No
Value Range	UNSIGNED32 (Table 54)
Default Value	Index 1400h: 200h + Node-ID, Index 1401h: 300h + Node-ID, Index 1402h: 400h + Node-ID, Index 1403h: 500h + Node-ID, Index 1404h Đ 15FFh: disabled

Sub-Index	2h
Description	transmission type
Entry Category	Mandatory
Access	ro;rw if variable transmission type is supported
PDO Mapping	No
Value Range	UNSIGNED8 (Table 55)
Default Value	(Device Profile dependent)

Sub-Index	3h
Description	inhibit time (not used for RPDO)
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	No

Sub-Index	4h
Description	compatibility entry
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	No

Sub-Index	5h
Description	event timer

Entry Category	Optional (not used for RPDO)
Access	rw
PDO Mapping	No
Value Range	0 = not used UNSIGNED16
Default Value	No

Object 1600h – 161Eh: Receive PDO Mapping Parameter

Contains the mapping for the PDOs the device is able to receive. The type of the PDO mapping parameter (21h) is described in 9.5.4. The sub-index 0h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted/received with the corresponding PDO. The sub-indices from 1h to number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length (Figure 66). All three values are hexadecimal coded. The length entry contains the length of the object in bit (1..40h). This parameter can be used to verify the overall mapping length. It is mandatory.

The structure of the entries from sub-index 1h - 40h is as follows:

Byte: MSB LSB

index (16 bit)	sub-index (8 bit)	object length (8 bit)

Figure 66: Structure of PDO Mapping Entry

If the change of the PDO mapping cannot be executed (e.g. the PDO length is exceeded or the SDO client attempts to map an object that cannot be mapped) the device responds with an Abort SDO Transfer Service.
Subindex 0 determines the valid number of objects that have been mapped. For changing the PDO mapping first the PDO has to be deleted, the sub-index 0 must be set to 0 (mapping is deactivated). Then the objects can be remapped. When a new object is mapped by writing a subindex between 1 and 64, the device may check whether the object specified by index / sub-index exists. If the object does not exist or the object cannot be mapped, the SDO transfer must be aborted with the Abort SDO Transfer Service with one of the abort codes 0602 0000h or 0604 0041h. After all objects are mapped subindex 0 is set to the valid number of mapped objects. Finally the PDO will be created by writing to its communication parameter COB-ID. When subindex 0 is set to a value >0 the device may validate the new PDO mapping before transmitting the response of the SDO service. If an error is detected the device has to transmit the Abort SDO Transfer Service with one of the abort codes 0602 0000h, 0604 0041h or 0604 0042h.
When subindex 0 is read the actual number of valid mapped objects is returned.
If data types (Index 1h-7h) are mapped they serve as “dummy entries”. The corresponding data in the PDO is not evaluated by the device. This optional feature is useful e.g. to transmit data to several devices using one PDO, each device only utilizing a part of the PDO. It is not possible to create a dummy mapping for a TPDO.

OBJECT DESCRIPTION

INDEX	1600h – 161Fh
Name	receive PDO mapping
Object Code	RECORD
Data Type	PDO Mapping
Category	Conditional; Mandatory for each supported PDO

ENTRY DESCRIPTION

Sub-Index	0h
Description	number of mapped application objects in PDO
Entry Category	Mandatory
Access	ro; rw if dynamic mapping is supported
PDO Mapping	No
Value Range	0: deactivated 1 - 64: activated
Default Value	(device profile dependent)

Sub-Index	1h - 40h
Description	PDO mapping for the nth application object to be mapped
Entry Category	Conditional depends on number and size of object be mapped
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	(device profile dependent)

Object 1800h – 181Fh: Transmit PDO Communication Parameter

Contains the communication parameters for the PDOs the device is able to transmit. The type of the PDO communication parameter (20h) is described in 9.5.4. A detailed description of the entries is done in the section for the Receive PDO Communication Parameter (1400h – 141Fh).

OBJECT DESCRIPTION

INDEX	1800h – 181Fh
Name	transmit PDO parameter
Object Code	RECORD
Data Type	PDO CommPar
Category	Conditional; Mandatory for each supported PDO

ENTRY DESCRIPTION

Sub-Index	0h
Description	largest sub-index supported
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	2 – 5

Sub-Index	1h
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	ro; rw if COB-ID can be configured
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 65)
Default Value	Index 1800h: 180h + Node-ID, Index 1801h: 280h + Node-ID, Index 1802h: 380h + Node-ID, Index 1803h: 480h + Node-ID, Index 1804h - 18FFh: disabled

Sub-Index	2h
Description	transmission type
Entry Category	Mandatory
Access	ro; rw if transmission type can be changed
PDO Mapping	No
Value Range	UNSIGNED8 (Table 54)
Default Value	(device profile dependent)

Sub-Index	3h
Description	inhibit time
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	No

Sub-Index	4h

Description	reserved
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	No

Sub-Index	5h
Description	event timer
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	0 = not used UNSIGNED16
Default Value	No

Object 1A00h – 1A1Fh: Transmit PDO Mapping Parameter

Contains the mapping for the PDOs the device is able to transmit. The type of the PDO mapping parameter (21h) is described in 9.5.4. A detailed description of the entries is done in the section for the Receive PDO Mapping Parameter (1600h – 161Fh).

OBJECT DESCRIPTION

INDEX	1A00h – 1A1Fh
Name	transmit PDO mapping
Object Code	RECORD
Data Type	PDO Mapping
Category	Conditional; Mandatory for each supported PDO

ENTRY DESCRIPTION

Sub-Index	0h
Description	number of mapped application objects in PDO
Entry Category	Mandatory
Access	ro; rw if dynamic mapping is supported
PDO Mapping	No
Value Range	0: deactivated 1 - 64: activated
Default Value	(device profile dependent)

Sub-Index	1h - 40h
Description	PDO mapping for the n-th application object to be mapped
Entry Category	Conditional; depends on number and size of objects to be mapped
Access	Rw
PDO Mapping	No
Value Range	UNSIGNED32
Default Value	(device profile dependent)

Multiplexed PDOs

MPDO Protocol

This protocol is used to implement MPDO services. The MPDO producer sends data and the multiplexor indicating the source or destination address.

MPDO ----------→ PDO
Producer Consumer

Byte	0		1-3	4-8
Meaning	f	Addr	m	D

'd' shall contain the data to be transferred. The value always shall be filled up to 32 bit. 3
'm' shall contain the multiplexor (Index and Sub-Index) of the variable in the object dictionary.
The MSB 'f' of the first byte shall be a format flag, and 'addr' shall be an address field, which may be used in the following combinations:

f	addr	Usage
0	0	Reserved
0	1-127	Source addressing. addr is a single producer's Node ID. Multiplexor is index and sub-index of the object dictionary of the producer.
1	0	Destination addressing. The consumer is a group.
1	1-127	Destination addressing. addr is a single consumer's Node ID. Multiplexor is index and sub-index of the object dictionary of the consumer.

Destination Address Mode (DAM)

The *addr* and the *m* field of the MPDO refers to the consumer. This allows access to the consumer's Object Dictionary in an SDO-like manner. With *addr = 0* , it allows multicasting and broadcasting, to write into the Object Dictionaries of more than one node simultaneously, without having a PDO for each single object.
Initiating a DAM-MPDO is application-dependent, like it is for SDOs.

Source Address Mode (SAM)

The *addr* and the *m* field of the MPDO refers to the producer. Only one producer MPDO of this type is allowed for each node.
Transmission type has to be 254 or 255.
The producer uses an Object Scanner List in order to know, which objects are to send. The consumer uses an Object Dispatcher List as a 'cross reference'. 3 The restriction about using 32-bit transfers only will not present problems in practice since all of the participating devices know the data types (and sizes) of their related objects.

Object dictionary entries

PDO Mapping Record

The meaning of Sub-Index 0 (number of mapped objects is extended. The valid range for non-multiplexed PDOs is 0 to 64. A value of 255 indicates a DAM-MPDO, a value of 254 indicates an SAM-MPDO.

For SAM, the further entries in the MR are don't care.

For DAM the first object describes the local object (there can be mapped only one object into an MPDO).

Index	Object	Name	Type
16XXh-1AXXh	0h	Number of mapped objects in the PDO: 0 .. 64: Valid range for number of mapped objects 254: formatted as SAM- MPDO 255: formatted as DAM- MPDO	Unsigned 8

This leaves open the possibility for further alternative PDO formats.

Additional error code meanings

Error Code Meaning
TBD

Predefinitions

Introduction

All TPDOs with transmission type 255 shall be transmitted when entering the OPERATIONAL state.

PDO Mapping

1st RPDO mapping (Screen Control)

This RPDO receives asynchronously the objects that control the screen to display. The default transmission type shall be 255. The default values of the mapped outputs are described in the Default State objects. *Note:* After power-on and application reset these default objects are valid.

Index	Sub-Index	Comment	Default Value
1600h	0h	number of mapped objects	7
1600h	1h	1st object to be mapped	2108 00 08h
1600h	2h	2nd object to be mapped	2106 00 08h
1600h	3h	3th object to be mapped	2003 00 08h
1600h	4h	4th object to be mapped	2004 00 08h
1600h	5h	5th object to be mapped	2800 01 08h
1600h	6h	6th object to be mapped	2800 02 08h
1600h	7h	7th object to be mapped	2801 01 10h

6.2.4 1st TPDO mapping (Screen Status)

This TPDO transmits event-driven the values of maximum 64 digital inputs. The default transmission type shall be 255; the default values for inhibit and event timer are 0. If one of the values changes its value, this PDO shall be transmitted immediately.

Index	Sub-Index	Comment	Default Value
1A00h	0h	number of mapped objects	6
1A00h	1h	1st object to be mapped	2109 00 08h
1A00h	2h	2nd object to be mapped	2108 00 08h
1A00h	3h	3th object to be mapped	2600 01 08h
1A00h	4h	4th object to be mapped	2600 02 08h
1A00h	5h	5th object to be mapped	2601 01 10h
1A00h	6h	6th object to be mapped	2601 02 10h

6.2.5 2nd RPDO mapping (Multiplexed Data Request)

This RPDO receives asynchronously the Multiplexed Data Requests. The default transmission type shall be 255. The default values of the mapped outputs are described in the Default State objects. *Note:* After power-on and application reset these default objects are valid.

Index	Sub-Index	Comment	Default Value
1601h	0h	number of mapped objects	4
1601h	1h	1st object to be mapped	2A00 00 08h
1601h	2h	2nd object to be mapped	2A01 00 10h
1601h	3h	3rd object to be mapped	2A02 00 08h
1601h	4h	4th object to be mapped	2A03 00 20h

6.2.6 2nd TPDO mapping (Multiplexed Data Response)

This TPDO transmits event-driven the values of the Multiplexed Data Response. The default transmission type shall be 255; the default values for inhibit and event timer are 0. If any of the variables changes its value, the PDO shall be transmitted immediately.

Index	Sub-Index	Comment	Default Value
1A01h	0h	number of mapped objects	4
1A01h	1h	1st object to be mapped	2A10 00 08h
1A01h	2h	2nd object to be mapped	2A11 00 10h
1A01h	3h	3rd object to be mapped	2A12 00 08h
1A01h	4h	4th object to be mapped	2A13 00 20h

6.2.7 3rd RPDO mapping (DataOut UNSIGNED8)

This RPDO receives asynchronously the DataOut values for subindex 3-0Ah.
The default transmission type shall be 255.

Index	Sub-Index	Comment	Default Value
1602h	0h	number of mapped objects	8
1602h	1h	1st object to be mapped	2800 03 08h
1602h	2h	2nd object to be mapped	2800 04 08h
1602h	3h	3rd object to be mapped	2800 05 08h
1602h	4h	4th object to be mapped	2800 06 08h
1602h	5h	5th object to be mapped	2800 07 08h
1602h	6h	6th object to be mapped	2800 08 08h
1602h	7h	7th object to be mapped	2800 09 08h
1602h	8h	8th object to be mapped	2800 0A 08h

The number of mapped objects into the PDO depends on the hardware.

6.2.8 3rd TPDO mapping (DataIn UNSIGNED8)

This TPDO transmits event-driven the 8-bit for DataIn UNSIGNED8 Objects subindex 3-0Ah. The default transmission type shall be 255. If any of the values changes its value, the PDO shall be transmitted immediately.

Index	Sub-Index	Comment	Default Value
1A02h	0h	number of mapped objects	8
1A02h	1h	1st object to be mapped	2600 03 08h
1A02h	2h	2nd object to be mapped	2600 04 08h
1A02h	3h	3rd object to be mapped	2600 05 08h
1A02h	4h	4th object to be mapped	2600 06 08h
1A02h	5h	5th object to be mapped	2600 07 08h
1A02h	6h	6th object to be mapped	2600 08 08h
1A02h	7h	7th object to be mapped	2600 09 08h
1A02h	8h	8th object to be mapped	2600 0A 08h

6.2.9 4th RPDO mapping (DataOut Large)

This RPDO receives asynchronously the following DataOut objects.
The default transmission type shall be 255.

Index	Sub-Index	Comment	Default Value
1603h	0h	number of mapped objects	3
1603h	1h	1st object to be mapped	2801 02 10h
1603h	2h	2nd object to be mapped	2801 03 10h
1603h	3h	3rd object to be mapped	2802 01 20h

6.2.10 4th TPDO mapping (DataIn Large)

This TPDO transmits event-driven the DataIn Objects Mapped. The default transmission type shall be 255. If any of the mapped DataIn objects input changes its value, the PDO shall be transmitted immediately.

Index	Sub-Index	Comment	Default Value
1A03h	0h	number of mapped objects	3
1A03h	1h	1st object to be mapped	2601 03 10h
1A03h	2h	2nd object to be mapped	2601 04 10h
1A03h	3h	3rd object to be mapped	2602 01 20h

Manufacturer Specific Objects

Overview:

Operator Interface Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2000	VAR	Program Mode	BOOLEAN	Rw(p)	N	V
2001	VAR	Language Select	UNSIGNED8	Rw	N	NV
2002	VAR	Units Select	UNSIGNED8	Rw	N	NV
2003	VAR	Piezo State	UNSIGNED8	Rw	Y	V
2004	VAR	Piezo Volume	UNSIGNED8	Rw	Y	V
2005	VAR	Temperature	SIGNED8	Ro	Y	V
2006	VAR	Brightness	UNSIGNED8	Ro	Y	V

Display Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2100	VAR	Number Columns	UNSIGNED8	Const	N	C
2101	VAR	Number Rows	UNSIGNED8	Const	N	C
2102	VAR	Contrast Mode	UNSIGNED8	Rw	N	NV
2103	ARRAY	Contrast Level	UNSIGNED8	Rw	N	NV
2104	VAR	Backlight Mode	UNSIGNED8	Rw	N	NV
2105	ARRAY	Backlight Level	UNSIGNED8	Rw	N	NV
2106	VAR	Current Screen	UNSIGNED8	Rw	Y	V
2107	VAR	Screen Status	UNSIGNED8	Ro	Y	V
2108	VAR	Display Command	UNSIGNED8	Rw	Y	V
2109	VAR	Display Response	UNSIGNED8	Ro	Y	V

Screen Configuration Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2200	ARRAY	Screen 0 Items	SCREEN_ITEM	Rw*	N	NV
…	…	…	…	…	…	…
22FF	ARRAY	Screen 255 Items	SCREEN_ITEM	Rw*	N	NV

Static Text Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2300	ARRAY	Static Text	VISIBLE_STRING	Rw*	N	NV

Multilingual Text Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2400	ARRAY	Multilingual Text Language 0	VISIBLE_STRING	Rw*	N	NV
2401	ARRAY	Multilingual Text Language 1	VISIBLE_STRING	Rw*	N	NV
2402	ARRAY	Multilingual Text Language 2	VISIBLE_STRING	Rw*	N	NV

Text Lookup Table Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2500	VAR	Lookup Table Search Value	USIGNED16	Rw	N	V
2501	VAR	Lookup Table Text	VISIBLE_STRING	Ro	N	NV
2502	VAR	Lookup Table Number Entries	USIGNED16	Rw*	N	NV
2503	VAR	Lookup Table Entry Select	USIGNED16	Rw*	N	V
2504	VAR	Lookup Table Entry Value	USIGNED16	Rw*	N	NV
2505	VAR	Lookup Text	VISIBLE_STRING	Rw*	N	NV
2506	VAR	Entry Not Found Text Language 0	VISIBLE_STRING	Rw*	N	NV

Multilingual Text Lookup Table Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2510	VAR	Lookup Table Search Value	USIGNED16	Rw	N	NV
2511	VAR	Lookup Table Text	VISIBLE_STRING	Ro	N	NV
2512	VAR	Lookup Table Number Entries	USIGNED16	Rw*	N	NV
2513	VAR	Lookup Table Value Entry	USIGNED16	Rw*	N	V
2514	VAR	Lookup Text Language 0	VISIBLE_STRING	Rw*	N	NV
2515	VAR	Lookup Text Language 1	VISIBLE_STRING	Rw*	N	NV
2516	VAR	Lookup Text Language 2	VISIBLE_STRING	Rw*	N	NV
2517	VAR	Entry Not Found Text Language 0	VISIBLE_STRING	Rw*	N	NV
2518	VAR	Entry Not Found Text Language 1	VISIBLE_STRING	Rw*	N	NV
2519	VAR	Entry Not Found Text Language 2	VISIBLE_STRING	Rw*	N	NV

DataIn Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2600	ARRAY	DataIn_UNSIGNED8	UNSIGNED8	Ro	Y	V
2601	ARRAY	DataIn_UNSIGNED16	UNSIGNED16	Ro	Y	V
2602	ARRAY	DataIn_UNSIGNED32	UNSIGNED32	Ro	Y	V
2603	ARRAY	DataIn_SIGNED8	SIGNED8	Ro	Y	V
2604	ARRAY	DataIn_SIGNED16	SIGNED16	Ro	Y	V
2605	ARRAY	DataIn_SIGNED32	SIGNED32	Ro	Y	V
2606	ARRAY	DataIn_REAL32	REAL32	Ro	Y	V

DataIn Status Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2700	ARRAY	DataIn_UNSIGNED8 Status	UNSIGNED8	Rwr	Y	V
2701	ARRAY	DataIn_UNSIGNED16 Status	UNSIGNED8	Rwr	Y	V
2702	ARRAY	DataIn_UNSIGNED32 Status	UNSIGNED8	Rwr	Y	V
2703	ARRAY	DataIn_SIGNED8 Status	UNSIGNED8	Rwr	Y	V
2704	ARRAY	DataIn_SIGNED16 Status	UNSIGNED8	Rwr	Y	V
2705	ARRAY	DataIn_SIGNED32 Status	UNSIGNED8	Rwr	Y	V
2706	ARRAY	DataIn_REAL32 Status	UNSIGNED8	Rwr	Y	V

DataOut Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2800	ARRAY	DataOut_UNSIGNED8	UNSIGNED8	Rww	Y	V
2801	ARRAY	DataOut_UNSIGNED16	UNSIGNED16	Rww	Y	V
2802	ARRAY	DataOut_UNSIGNED32	UNSIGNED32	Rww	Y	V
2803	ARRAY	DataOut_SIGNED8	SIGNED8	Rww	Y	V
2804	ARRAY	DataOut_SIGNED16	SIGNED16	Rww	Y	V
2805	ARRAY	DataOut_SIGNED32	SIGNED32	Rww	Y	V
2806	ARRAY	DataOut_REAL32	REAL32	Rww	Y	V

DataOut Status Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2900	ARRAY	DataOut_UNSIGNED8 Status	UNSIGNED8	Rwr	Y	V
2901	ARRAY	DataOut_UNSIGNED16 Status	UNSIGNED8	Rwr	Y	V
2902	ARRAY	DataOut_UNSIGNED32 Status	UNSIGNED8	Rwr	Y	V
2903	ARRAY	DataOut_SIGNED8 Status	UNSIGNED8	Rwr	Y	V
2904	ARRAY	DataOut_SIGNED16 Status	UNSIGNED8	Rwr	Y	V
2905	ARRAY	DataOut_SIGNED32 Status	UNSIGNED8	Rwr	Y	V
2906	ARRAY	DataOut_REAL32 Status	UNSIGNED8	Rwr	Y	V

Multiplexed Data Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2A00	VAR	MD Request Control Byte	UNSIGNED8	Rww	Y	V
2A01	VAR	MD Request Index	UNSIGNED16	Rww	Y	V
2A02	VAR	MD Request SubIndex	UNSIGNED8	Rww	Y	V
2A03	VAR	MD Request Data	UNSIGNED32	Rww	Y	V
2A10	VAR	MD Response Status Byte	UNSIGNED8	Ro	Y	V
2A11	VAR	MD Response Data Index	UNSIGNED16	Ro	Y	V
2A12	VAR	MD Response SubIndex	UNSIGNED8	Ro	Y	V
2A13	VAR	MD Response Data	UNSIGNED32	Ro	Y	V

Operator Input Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2B00	ARRAY	Operator Input 0	OPERATOR_INPUT_OBJECT	Rw	N	NV
…	…	…	….	…	…	…
2BFF	ARRAY	Operator Input 255	OPERATOR_INPUT_OBJECT	Rw	N	NV

Operator Output Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2C00	ARRAY	Operator Output 0	OPERATOR_OUTPUT_OBJECT	Rw	N	NV
…	…	…	….	…	…	…
2CFF	ARRAY	Operator Output 255	OPERATOR_OUTPUT_OBJECT	Rw	N	NV

Status Code Text List Objects

Index (hex)	Object (Symbolic Name)	Name	Type	Access	I/O MAP	NV
2D00	ARRAY	Status Code Entry	VISIBLE_STRING	Rw*	N	NV

Manufacturer Specific Object Details:
Operator Interface Objects:

Object 2000h: Program Mode

The when set, Program Mode object allows the network to change the configuration of various objects that are read only when not in the program mode.

OBJECT DESCRIPTION

INDEX	2000h
Name	Program Mode
Object Code	VAR
Data Type	BOOLEAN
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	BOOLEAN
Default Value	0

Object 2001h: Language Select

The Language Select Object allows the user to set which sets of multi-lingual lookups are used.

OBJECT DESCRIPTION

INDEX	2001h
Name	Language Select
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	0-2
Default Value	0

This object affects which of the multilingual text objects are used for the Operator Output Objects.

Object 2002h: Units Select

The Units Select Object allows the user to select the set of conversion factors are to be used.

OBJECT DESCRIPTION

INDEX	2002h
Name	Language Select
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	0-2
Default Value	0

The Units Select object affects which set of conversion factor is used for the Operator Input and Operator Output Objects.

Object 2003h: Piezo State

The Piezo State Object allows the network to change the state of the piezo alarm.

OBJECT DESCRIPTION

INDEX	2003h
Name	Piezo Volume
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	Yes
Value Range	0-3
Default Value	0

The Piezo State object supports the following values:
0=off
1=on
2=slow off on (approximately 1 second)
3=fast off on (approximately 500 msec)

Object 2004h: Piezo Volume

The Piezo Volume Object allows the network to change the volume of the piezo alarm.

OBJECT DESCRIPTION

INDEX	2002h
Name	Piezo Volume
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	Yes
Value Range	UNSIGNED 8
Default Value	100

Object 2005h: Temperature

The Temperature Object allows the network to read the temperature of the Operator Interface.

OBJECT DESCRIPTION

INDEX	2005h
Name	Temperature
Object Code	VAR
Data Type	SIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	Yes
Value Range	SIGNED8
Default Value	Current Temperature

The Temperature object indicates the current temperature in degrees Celcius.

Object 2006h: Brightness

The Brightness Object allows the network to read the brightness of the visible light hitting the operator interface.

OBJECT DESCRIPTION

INDEX	2006h
Name	Brightness
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	Yes
Value Range	UNSIGNED8
Default Value	Current Brightness

The brightness object indicates the current brightness in a scale of 0-255.

Display Objects:

Object 2100h: Number Columns

The Number Columns object allows the network to read the number of columns on the display.

OBJECT DESCRIPTION

INDEX	2100h
Name	Number Columns
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Const
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	20 (depends on display type)

Object 2101h: Number Rows

The Number Rows object allows the network to read the number of rows on the display.

OBJECT DESCRIPTION

INDEX	2101h
Name	Number Rows
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Const
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	4 (depends on display type)

Object 2102h: Contrast Mode

The Contrast Mode object allows the network to set the mode of the contrast adjust for the LCD display.

OBJECT DESCRIPTION

INDEX	2102h
Name	Contrast Mode
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Supported Values:
0=Auto (contrast adjusted automatically by temperature)
1=Manual (contrast adjusted by operator)

Object 2103h: Contrast Level

This object provides a method to set the contrast of the display.

Object Description

INDEX	2103h
Name	Contrast Level
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Contrast settings
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	4

Sub-Index	1h
Description	Contrast Manual
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

Sub-Index	2h
Description	Contrast Cold
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	UNSIGNED8
Default value	No

Sub-Index	3h
Description	Contrast Warm
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	UNSIGNED8
Default value	No

Sub-Index	4h
Description	Contrast Hot
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	UNSIGNED8
Default value	No

Object 2104h: Backlighting Mode

The Backlighting Mode object allows the network to set the mode of the backlight adjustment for the LCD display.

OBJECT DESCRIPTION

INDEX	2104h
Name	Backlighting Mode
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Supported Values:
0=Auto (backlight adjusted automatically by surrounding light)
1=Manual (backlight adjusted by operator)

Object 2105h: Backlight Level

This object provides a method to set the backlight level of the display.

Object Description

INDEX	2105h
Name	Backlight Level
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Backlight settings
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	4

Sub-Index	1h
Description	Backlight Manual
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	No

Sub-Index	2h
Description	Backlight Dark
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	UNSIGNED8
Default value	No

Sub-Index	3h
Description	Backlight Medium
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	UNSIGNED8
Default value	No

Sub-Index	4h
Description	Backlight Bright
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	UNSIGNED8
Default value	No

Object 2106h: Current Screen

The Current Screen object allows the network to set the screen that is currently displayed on ther the LCD display.

OBJECT DESCRIPTION

INDEX	2106h
Name	Current Screen
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

Object 2107h: Screen Status

The Screen Status object allows the network to determine the state of the current screen.

OBJECT DESCRIPTION

INDEX	2107h
Name	Screen Status
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

This value is the maximum value of all operator input status bytes. If this value is 0 all operations are complete. If there are no operator input objects on the screen then this value is immediately set to 0 once the text has actually been written to the LCD.

Object 2108h: Display Command

The Display Command object allows the network to send a command to the display. This is useful when the network is using process data to control the operator interface.

OBJECT DESCRIPTION

INDEX	2108h
Name	Display Command
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

The Display Command is a series of bits that can control the display.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value								Request Toggle

Request Toggle: This bit must be toggled if the network wants to re-display the same screen as the last screen displayed.

Object 2109h: Display Response

The Display Response object allows the network to see a response to a command that it sent to the display. This is useful when the network is using process data to control the operator interface.

OBJECT DESCRIPTION

INDEX	2109h
Name	Display Response
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

The Display Command is a series of bits that can control the display.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Value								Command Ack

Command Ack: This bit will be toggled to match the Command Request bit in the Display Command object when the command is complete.

Screen Configuration Objects

Object 2200-22FFh: Screen Configuration Objects

These objects provide a means for the user to configure each of the 256 screens that the Operator interface supports.

Object Description

INDEX	2200-22FFh
Name	Screen X Configuration
Object Code	Array
Data Type	SCREEN_ITEM
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Screen Items Screen X
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Sub-Index	1h
Description	Screen Item 1 Screen #X
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	SCREEN_ITEM
Default Value	No

To

Sub-Index	FEh
Description	Screen Item Feh Screen #X
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	SCREEN_ITEM
Default value	No

The Screen Configuration Objects store the configuration of each screen in the display. When using the Windows configuration tool, this object is invisible to the user, but it may be helpful to know how the information is used by the Operator interface to display the screen. This information is also necessary to build screens using an eds file or SDO commands.

The format for a SCREEN_ITEM is as follows

```
typedef struct{
        UNSIGNED8 Left;         //X starting location on display
        USNIGNED8 Top;          //Y location on display
        UNSIGNED8 Width;
        UNSIGNED8 Height;
        UNSIGNED8 Mode;         //Describes display mode of item
        UNSIGNED32 Mapped Screen Object;        //The object to be displayed
```

} SCREEN_ITEM;

The user should configure screen 0 (object 2200h) to display the startup message, as this will be the default screen displayed on power up.

Example:

If the user wants to display the following message in screen 0 perform the following steps.

Column

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0							I	N	C	,	L	L	C							
1								C	A	N	o	p	e	n						
2		O	p	e	r	a	t	o	r		I	n	t	e	r	f	a	c	e	
3																				

The user would program the following:
In this example we will use the first three static text objects:
Index 2300h SubIndex 1 set to “INC,LLC.”
Index 2300h SubIndex 2 set to “CANopen.”
Index 2300h SubIndex 3 set to “Operator Interface”

Now configure screen 0:
Index 2200h SubIndex 0 set to 3 (this sets screen 0 to 3 items)
Index 2200h SubIndex 1 set to {6,0,0,0,0x23000100}
Index 2200h SubIndex 2 set to {7,1,0,0, 0x23000200}
Index 2200h SubIndex 3 set to {1,2,0,0, 0x23000300}

Please see CANopen Operator Interface Examples document for further examples

Static Text Objects

Object 2300: Static Text Objects

This object provides a way for the user to enter static text messages.

Object Description

INDEX	2300h
Name	Static Text Entry
Object Code	Array
Data Type	VISIBLE_STRING
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Static Text Entries
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	32

Sub-Index	1h
Description	Static Text 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	No

…

Sub-Index	20h
Description	Static Text 32
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	VISIBLE_STRING
Default value	No

Multilingual Text Objects

Object 2400h: Multilingual Text Language 1

This object provides a way for the user to enter multilingual text messages.

Object Description

INDEX	2400h
Name	Multilingual Text Language 1
Object Code	Array
Data Type	VISIBLE_STRING
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Multilingual Text Entries Language 1
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	Multilingual Text Language 1 Text 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	VISIBLE_STRING
Default Value	No

To

Sub-Index	Feh
Description	Multilingal Text Language 1 Text Feh
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	VISIBLE_STRING
Default value	No

The multilingual text objects allow the user to program text objects with the same meaning in up to three languages. If the operator interface is displaying a screen that has a multilingual text object mapped in it, the display will use the Language Select object to determine which text string to display.

Object 2401h: Multilingual Text Language 2

This object provides a way for the user to enter multilingual text messages.

Object Description

INDEX	2401h
Name	Multilingual Text Language 2
Object Code	Array
Data Type	VISIBLE_STRING
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Multilingual Text Entries Language 2
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	Multilingual Text Language 2 Text 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	VISIBLE_STRING
Default Value	No

To

Sub-Index	Feh
Description	Multilingal Text Language 2 Text Feh
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	VISIBLE_STRING
Default value	No

Object 2402h: Multilingual Text Language 3

This object provides a way for the user to enter multilingual text messages.

Object Description

INDEX	2402h
Name	Multilingual Text Language 3
Object Code	Array
Data Type	VISIBLE_STRING
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Multilingual Text Entries Language 3
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	Multilingual Text Language 3 Text 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	VISIBLE_STRING
Default Value	No

To

Sub-Index	Feh
Description	Multilingal Text Language 3 Text Feh
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	VISIBLE_STRING
Default value	No

Lookup Table Objects:

Object 2500h: Lookup Table Search Value

The Lookup Table Search value allows the network to send a value to use to look for a match in the Lookup table. If a match is found the Text object displays the programmed value. If the value is not found the Text object displays what is programmed into the Entry Not Found Text object.

OBJECT DESCRIPTION

INDEX	2500h
Name	Lookup Table Search Value
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

Object 2501h: Lookup Table Text Result

The Lookup Table Text Object returns either the text value found in the lookup table or the text found in the Entry Not Found Text object if the value is not found

OBJECT DESCRIPTION

INDEX	2501h
Name	Lookup Table Text Result
Object Code	VAR
Data Type	VISIBLE_STRING
Category	Optional

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	No
Value Range	VISIBLE_STRING
Default Value	0

Object 2502h: Lookup Table Number Entries

The Lookup Table Number of Entries object allows the network to read and write the number of entries in the lookup table.

OBJECT DESCRIPTION

INDEX	2502h
Name	Lookup Table Number of Entries
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

Object 2503h: Lookup Table Entry Select

The Lookup Table Entry Select object allows the network to read and write the index of the selected entry in the lookup table.

OBJECT DESCRIPTION

INDEX	2503h
Name	Lookup Table Entry Select
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	0-Number of Entries
Default Value	0

Object 2504h: Lookup Table Entry Value

The Lookup Table Entry Value object allows the network to read and write the value of the selected entry in the lookup table.

OBJECT DESCRIPTION

INDEX	2504h
Name	Lookup Table Entry Value
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

Object 2505h: Lookup Table Entry Text

The Lookup Table Entry Text Object allows the user to read or write the text entry pointed to by the selected entry object.

OBJECT DESCRIPTION

INDEX	2505h
Name	Lookup Table Entry Text
Object Code	VAR
Data Type	VISIBLE_STRING
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	VISIBLE_STRING
Default Value	0

Object 2506h: Lookup Table Entry Not Found Text

The Lookup Table Entry Not Found Text Object allows the user to read or write the text that will be returned if an item is not found in the lookup table.

OBJECT DESCRIPTION

INDEX	2506h
Name	Lookup Table Entry Not Found Text
Object Code	VAR
Data Type	VISIBLE_STRING
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	No
Value Range	VISIBLE_STRING
Default Value	0

DataIn Objects

Object 2600h: DataIn_UNSIGNED8

This object provides a storage area for UNSIGNED8 variables.

Object Description

INDEX	2600h
Name	DataIn_UNSIGNED8
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_UNSIGNED8s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_UNSIGNED8 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataIn_UNSIGNED8 FEh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataIn_UNSIGNED8 objects allow the network to retrieve information requested from the operator in the form of an UNSIGNED8. To do this the configuration software maps this object into the corresponding USER_INPUT data object field. If this object is used in this manner, the corresponding status byte will be set to the state of the operator's progress in entering the requested data (see User Input objects).

Object 2601h: DataIn_UNSIGNED16

This object provides a storage area for UNSIGNED16 variables.

Object Description

INDEX	2601h
Name	DataIn_UNSIGNED16
Object Code	Array
Data Type	UNSIGNED16
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_UNSIGNED16s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_UNSIGNED16 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED16
Default Value	0

To

Sub-Index	Feh
Description	DataIn_UNSIGNED16 FEh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED16
Default value	0

The DataIn_UNSIGNED16 objects allow the network to retrieve information requested from the operator in the form of an UNSIGNED16. To do this the configuration software maps this object into the corresponding USER_INPUT data object field. If this object is used in this manner, the corresponding status byte will be set to the state of the operator's progress in entering the requested data (see User Input objects).

Object 2602h: DataIn_UNSIGNED32

This object provides a storage area for UNSIGNED32 variables.

Object Description

INDEX	2602h
Name	DataIn_UNSIGNED32
Object Code	Array
Data Type	UNSIGNED32
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_UNSIGNED32s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_UNSIGNED32 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED32
Default Value	0

To

Sub-Index	Feh
Description	DataIn_UNSIGNED32 FEh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED32
Default value	0

The DataIn_UNSIGNED32 objects allow the network to retrieve information requested from the operator in the form of an UNSIGNED32. To do this the configuration software maps this object into the corresponding USER_INPUT data object field. If this object is used in this manner, the corresponding status byte will be set to the state of the operator's progress in entering the requested data (see User Input objects).

Object 2603h: DataIn_SIGNED8

This object provides a storage area for SIGNED8 variables.

Object Description

INDEX	2603h
Name	DataIn_SIGNED8
Object Code	Array
Data Type	SIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_SIGNED8s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_SIGNED8 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	SIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataIn_SIGNED8 FEh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	SIGNED8
Default value	0

The DataIn_SIGNED8 objects allow the network to retrieve information requested from the operator in the form of an SIGNED8. To do this the configuration software maps this object into the corresponding USER_INPUT data object field. If this object is used in this manner, the corresponding status byte will be set to the state of the operator's progress in entering the requested data (see User Input objects).

Object 2604h: DataIn_SIGNED16

This object provides a storage area for SIGNED16 variables.

Object Description

INDEX	2604h
Name	DataIn_SIGNED16
Object Code	Array
Data Type	SIGNED16
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_SIGNED16s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_SIGNED16 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	SIGNED16
Default Value	0

To

Sub-Index	Feh
Description	DataIn_SIGNED16 FEh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	SIGNED16
Default value	0

The DataIn_SIGNED16 objects allow the network to retrieve information requested from the operator in the form of an SIGNED16. To do this the configuration software maps this object into the corresponding USER_INPUT data object field. If this object is used in this manner, the corresponding status byte will be set to the state of the operator's progress in entering the requested data (see User Input objects).

Object 2605h: DataIn_SIGNED32

This object provides a storage area for SIGNED32 variables.

Object Description

INDEX	2605h
Name	DataIn_SIGNED32
Object Code	Array
Data Type	SIGNED32
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_SIGNED32s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_SIGNED32 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	SIGNED32
Default Value	0

To

Sub-Index	Feh
Description	DataIn_SIGNED32 FEh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	SIGNED32
Default value	0

The DataIn_SIGNED32 objects allow the network to retrieve information requested from the operator in the form of an SIGNED32. To do this the configuration software maps this object into the corresponding USER_INPUT data object field. If this object is used in this manner, the corresponding status byte will be set to the state of the operator's progress in entering the requested data (see User Input objects).

Object 2606h: DataIn_REAL32

This object provides a storage area for REAL32 variables.

Object Description

INDEX	2606h
Name	DataIn_REAL32
Object Code	Array
Data Type	REAL32
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_REAL32s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_REAL32 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	REAL32
Default Value	0

To

Sub-Index	Feh
Description	DataIn_REAL32 FEh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	REAL32
Default value	0

The DataIn_REAL32 objects allow the network to retrieve information requested from the operator in the form of an REAL32. To do this the configuration software maps this object into the corresponding USER_INPUT data object field. If this object is used in this manner, the corresponding status byte will be set to the state of the operator's progress in entering the requested data (see User Input objects).

DataIn Status Objects

Object 2700h: DataIn_UNSIGNED8 Status

This object provides a status indication for UNSIGNED8 input variables.

Object Description

INDEX	2700h
Name	DataIn_UNSIGNED8 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_UNSIGNED8 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_UNSIGNED8 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataIn_UNSIGNED8 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataIn_UNSIGNED8 Status objects allow the network to monitor the status of the corresponding DataIn_UNSIGNED8 input variable. The typical application would be where the input variable is mapped to a UER_INPUT object. The network can then monitor the status byte to see whether the user has started keying in the information or the input is complete (see User Input objects).

Object 2701h: DataIn_UNSIGNED16 Status

This object provides a status indication for UNSIGNED16 input variables.

Object Description

INDEX	2701h
Name	DataIn_UNSIGNED16 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_UNSIGNED16 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_UNSIGNED16 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Optional
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataIn_UNSIGNED16 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataIn_UNSIGNED16 Status objects allow the network to monitor the status of the corresponding DataIn_UNSIGNED16 input variable. The typical application would be where the input variable is mapped to a UER_INPUT object. The network can then monitor the status byte to see whether the user has started keying in the information or the input is complete (see User Input objects).

Object 2702h: DataIn_UNSIGNED32 Status

This object provides a status indication for UNSIGNED32 input variables.

Object Description

INDEX	2702h
Name	DataIn_UNSIGNED32 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_UNSIGNED32 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_UNSIGNED32 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataIn_UNSIGNED32 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataIn_UNSIGNED32 Status objects allow the network to monitor the status of the corresponding DataIn_UNSIGNED32 input variable. The typical application would be where the input variable is mapped to a UER_INPUT object. The network can then monitor the status byte to see whether the user has started keying in the information or the input is complete (see User Input objects).

Object 2703h: DataIn_SIGNED8 Status

This object provides a status indication for SIGNED8 input variables.

Object Description

INDEX	2703h
Name	DataIn_SIGNED8 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_SIGNED8 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_SIGNED8 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataIn_SIGNED8 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataIn_SIGNED8 Status objects allow the network to monitor the status of the corresponding DataIn_SIGNED8 input variable. The typical application would be where the input variable is mapped to a UER_INPUT object. The network can then monitor the status byte to see whether the user has started keying in the information or the input is complete (see User Input objects).

Object 2704h: DataIn_SIGNED16 Status

This object provides a status indication for SIGNED16 input variables.

Object Description

INDEX	2704h
Name	DataIn_SIGNED16 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_SIGNED16 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_SIGNED16 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataIn_SIGNED16 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataIn_SIGNED16 Status objects allow the network to monitor the status of the corresponding DataIn_SIGNED16 input variable. The typical application would be where the input variable is mapped to a UER_INPUT object. The network can then monitor the status byte to see whether the user has started keying in the information or the input is complete (see User Input objects).

Object 2705h: DataIn_SIGNED32 Status

This object provides a status indication for SIGNED32 input variables.

Object Description

INDEX	2702h
Name	DataIn_SIGNED32 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_SIGNED32 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_SIGNED32 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataIn_SIGNED32 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataIn_SIGNED32 Status objects allow the network to monitor the status of the corresponding DataIn_SIGNED32 input variable. The typical application would be where the input variable is mapped to a UER_INPUT object. The network can then monitor the status byte to see whether the user has started keying in the information or the input is complete (see User Input objects).

Object 2706h: DataIn_REAL32 Status

This object provides a status indication for REAL32 input variables.

Object Description

INDEX	2706h
Name	DataIn_REAL32 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataIn_REAL32 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataIn_REAL32 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataIn_REAL32 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataIn_REAL32 Status objects allow the network to monitor the status of the corresponding DataIn_REAL32 input variable. The typical application would be where the input variable is mapped to a UER_INPUT object. The network can then monitor the status byte to see whether the user has started keying in the information or the input is complete (see User Input objects).

DataOut Objects

Object 2800h: DataOut_UNSIGNED8

This object provides a storage area for UNSIGNED8 output variables.

Object Description

INDEX	2800h
Name	DataOut_UNSIGNED8
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_UNSIGNED8s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_UNSIGNED8 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataOut_UNSIGNED8 Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataOut_UNSIGNED8 objects allow the network to pass values to the operator interface in the form of an UNSIGNED8. The typical application would be where the output variable is mapped to a USER_OUTPUT object. The network can then update the variable and if the current screen contains this USER_OUTPUT object then the display is automatically updated. The object also has a corresponding status byte that allows the network to signal to the operator interface that the data in this object is valid or invalid (see User Output objects).

Object 2801h: DataOut_UNSIGNED16

This object provides a storage area for UNSIGNED16 output variables.

Object Description

INDEX	2801h
Name	DataOut_UNSIGNED16
Object Code	Array
Data Type	UNSIGNED16
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_UNSIGNED16s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_UNSIGNED16 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED16
Default Value	0

To

Sub-Index	Feh
Description	DataOut_UNSIGNED16 Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED16
Default value	0

The DataOut_UNSIGNED16 objects allow the network to pass values to the operator interface in the form of an UNSIGNED16. The typical application would be where the output variable is mapped to a USER_OUTPUT object. The network can then update the variable and if the current screen contains this USER_OUTPUT object then the display is automatically updated. The object also has a corresponding status byte that allows the network to signal to the operator interface that the data in this object is valid or invalid (see User Output objects).

Object 2802h: DataOut_UNSIGNED32

This object provides a storage area for UNSIGNED32 output variables.

Object Description

INDEX	2802h
Name	DataOut_UNSIGNED32
Object Code	Array
Data Type	UNSIGNED32
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_UNSIGNED32s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_UNSIGNED32 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED32
Default Value	0

To

Sub-Index	Feh
Description	DataOut_UNSIGNED32 Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED32
Default value	0

The DataOut_UNSIGNED32 objects allow the network to pass values to the operator interface in the form of an UNSIGNED32. The typical application would be where the output variable is mapped to a USER_OUTPUT object. The network can then update the variable and if the current screen contains this USER_OUTPUT object then the display is automatically updated. The object also has a corresponding status byte that allows the network to signal to the operator interface that the data in this object is valid or invalid (see User Output objects).

Object 2803h: DataOut_SIGNED8

This object provides a storage area for SIGNED8 output variables.

Object Description

INDEX	2803h
Name	DataOut_SIGNED8
Object Code	Array
Data Type	SIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_SIGNED8s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_SIGNED8 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	SIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataOut_SIGNED8 Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	SIGNED8
Default value	0

The DataOut_SIGNED8 objects allow the network to pass values to the operator interface in the form of an SIGNED8. The typical application would be where the output variable is mapped to a USER_OUTPUT object. The network can then update the variable and if the current screen contains this USER_OUTPUT object then the display is automatically updated. The object also has a corresponding status byte that allows the network to signal to the operator interface that the data in this object is valid or invalid (see User Output objects).

Object 2804h: DataOut_SIGNED16

This object provides a storage area for SIGNED16 output variables.

Object Description

INDEX	2804h
Name	DataOut_SIGNED16
Object Code	Array
Data Type	SIGNED16
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_SIGNED16s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_SIGNED16 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	SIGNED16
Default Value	0

To

Sub-Index	Feh
Description	DataOut_SIGNED16 Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	SIGNED16
Default value	0

The DataOut_SIGNED16 objects allow the network to pass values to the operator interface in the form of an SIGNED16. The typical application would be where the output variable is mapped to a USER_OUTPUT object. The network can then update the variable and if the current screen contains this USER_OUTPUT object then the display is automatically updated. The object also has a corresponding status byte that allows the network to signal to the operator interface that the data in this object is valid or invalid (see User Output objects).

Object 2805h: DataOut_SIGNED32

This object provides a storage area for SIGNED32 output variables.

Object Description

INDEX	2805h
Name	DataOut_SIGNED32
Object Code	Array
Data Type	SIGNED32
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_SIGNED32s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_SIGNED32 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	SIGNED32
Default Value	0

To

Sub-Index	Feh
Description	DataOut_SIGNED32 Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	SIGNED32
Default value	0

The DataOut_SIGNED32 objects allow the network to pass values to the operator interface in the form of an SIGNED32. The typical application would be where the output variable is mapped to a USER_OUTPUT object. The network can then update the variable and if the current screen contains this USER_OUTPUT object then the display is automatically updated. The object also has a corresponding status byte that allows the network to signal to the operator interface that the data in this object is valid or invalid (see User Output objects).

Object 2806h: DataOut_REAL32

This object provides a storage area for REAL32 output variables.

Object Description

INDEX	2806h
Name	DataOut_REAL32
Object Code	Array
Data Type	REAL32
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_REAL32s
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_REAL32 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	REAL32
Default Value	0

To

Sub-Index	Feh
Description	DataOut_REAL32 Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	REAL32
Default value	0

The DataOut_REAL32 objects allow the network to pass values to the operator interface in the form of an REAL32. The typical application would be where the output variable is mapped to a USER_OUTPUT object. The network can then update the variable and if the current screen contains this USER_OUTPUT object then the display is automatically updated. The object also has a corresponding status byte that allows the network to signal to the operator interface that the data in this object is valid or invalid (see User Output objects).

DataOut Status Objects

Object 2900h: DataOut_UNSIGNED8 Status

This object provides a status indication for UNSIGNED8 output variables.

Object Description

INDEX	2900h
Name	DataOut_UNSIGNED8 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_UNSIGNED8 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_UNSIGNED8 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataOut_UNSIGNED8 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataOut_UNSIGNED8 Status objects allow the network to indicate to the Operator Interface whether the value in the corresponding DataOut_UNSIGNED8 output variable is valid or not. The typical application would be where the output variable is mapped to an OPERATOR_OUTPUT object. The display would output an error string defined during the configuration process instead of the actual value if the status is non-zero (see Operator Output objects).

Object 2901h: DataOut_UNSIGNED16 Status

This object provides a status indication for UNSIGNED16 output variables.

Object Description

INDEX	2901h
Name	DataOut_UNSIGNED16 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_UNSIGNED16 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_UNSIGNED16 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataOut_UNSIGNED16 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataOut_UNSIGNED16 Status objects allow the network to indicate to the Operator Interface whether the value in the corresponding DataOut_UNSIGNED16 output variable is valid or not. The typical application would be where the output variable is mapped to an OPERATOR_OUTPUT object. The display would output an error string defined during the configuration process instead of the actual value if the status is non-zero (see Operator Output objects).

Object 2902h: DataOut_UNSIGNED32 Status

This object provides a status indication for UNSIGNED32 output variables.

Object Description

INDEX	2902h
Name	DataOut_UNSIGNED32 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_UNSIGNED32 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_UNSIGNED32 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataOut_UNSIGNED32 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataOut_UNSIGNED32 Status objects allow the network to indicate to the Operator Interface whether the value in the corresponding DataOut_UNSIGNED8 output variable is valid or not. The typical application would be where the output variable is mapped to an OPERATOR_OUTPUT object. The display would output an error string defined during the configuration process instead of the actual value if the status is non-zero (see Operator Output objects).

Object 2903h: DataOut_SIGNED8 Status

This object provides a status indication for SIGNED8 outputvariables.

Object Description

INDEX	2903h
Name	DataOut_SIGNED8 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_SIGNED8 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_SIGNED8 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataOut_SIGNED8 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataOut_SIGNED8 Status objects allow the network to indicate to the Operator Interface whether the value in the corresponding DataOut_SIGNED8 output variable is valid or not. The typical application would be where the output variable is mapped to an OPERATOR_OUTPUT object. The display would output an error string defined during the configuration process instead of the actual value if the status is non-zero (see Operator Output objects).

Object 2904h: DataOut_SIGNED16 Status

This object provides a status indication for SIGNED16 output variables.

Object Description

INDEX	2904h
Name	DataOut_SIGNED16 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_SIGNED16 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_SIGNED16 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataOut_SIGNED16 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataOut_SIGNED16 Status objects allow the network to indicate to the Operator Interface whether the value in the corresponding DataOut_UNSIGNED16 output variable is valid or not. The typical application would be where the output variable is mapped to an OPERATOR_OUTPUT object. The display would output an error string defined during the configuration process instead of the actual value if the status is non-zero (see Operator Output objects).

Object 2905h: DataOut_SIGNED32 Status

This object provides a status indication for SIGNED32 output variables.

Object Description

INDEX	2902h
Name	DataOut_SIGNED32 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_SIGNED32 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_SIGNED32 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataOut_SIGNED32 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataOut_SIGNED32 Status objects allow the network to indicate to the Operator Interface whether the value in the corresponding DataOut_SIGNED32 output variable is valid or not. The typical application would be where the output variable is mapped to an OPERATOR_OUTPUT object. The display would output an error string defined during the configuration process instead of the actual value if the status is non-zero (see Operator Output objects).

Object 2906h: DataOut_REAL32 Status

This object provides a status indication for REAL32 output variables.

Object Description

INDEX	2906h
Name	DataOut_REAL32 Status
Object Code	Array
Data Type	UNSIGNED8
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of DataOut_REAL32 Status Bytes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	DataOut_REAL32 Status 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	Default
Value Range	UNSIGNED8
Default Value	0

To

Sub-Index	Feh
Description	DataOut_REAL32 Status Feh
Access	Rw
Entry Category	Optional
PDO Mapping	Optional
Value Range	UNSIGNED8
Default value	0

The DataOut_REAL32 Status objects allow the network to indicate to the Operator Interface whether the value in the corresponding DataOut_REAL32 output variable is valid or not. The typical application would be where the output variable is mapped to an OPERATOR_OUTPUT object. The display would output an error string defined during the configuration process instead of the actual value if the status is non-zero (see Operator Output objects).

Multiplexed Data Objects:

Object 2A00h: MD Request Control Byte

The MD Request Control Byte Object allows the network to send requests to the operator interface using process data.

OBJECT DESCRIPTION

INDEX	2A00h
Name	MD Request Control Byte
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	Yes
Value Range	UNSIGNED8
Default Value	0

Control Byte Format:

Bit 7	Bit 6	Bit 5	Bit 4	Bits 3-0
Reserved	Reserved	Reserved	Toggle	Command

Bit Definitions:

Commands:

0 Nop – No operation is performed

1 Read Object – Object Specified is read and data can be found in MUX Request Data

2 Write Object – Object Specified is written with data supplied in MUX Request Data

Toggle:

The user must toggle this bit if the device is to perform the requested command again. Optionally the user can switch commands such as write then read or write then NOP then write again.

Object 2A01h: MD Request Index

The MD Request Index allows the network to send requests to the operator interface using process data. The index object specifies the object to read or write.

OBJECT DESCRIPTION

INDEX	2A01h
Name	MD Request Index
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	Yes
Value Range	UNSIGNED16
Default Value	0

Object 2A02h: MD Request SubIndex

The MD Request SubIndex allows the network to send requests to the operator interface using process data. The SubIndex object specifies the subindex of the object to read or write.

OBJECT DESCRIPTION

INDEX	2A02h
Name	MD Request SubIndex
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	Yes
Value Range	UNSIGNED8
Default Value	0

Object 2A03h: MD Request Data

The MD Request Data allows the network to send requests to the operator interface using process data. The Request Data object specifies the data to be written to the requested index and subindex. The amount of data transferred depends on the size of the object written to.

OBJECT DESCRIPTION

INDEX	2A03h
Name	MD Request Data
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Access	Rw
PDO Mapping	Yes
Value Range	UNSIGNED32
Default Value	0

Object 2A10h: MD Response Status Byte

The MD Response Status Byte Object allows the network to get the response to the request made using the MD Request Objects using process data.

OBJECT DESCRIPTION

INDEX	2A10h
Name	MD Response Status Byte
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	Yes
Value Range	UNSIGNED8
Default Value	0

Status Byte Format:

Bit 7	Bit 6	Bit 5	Bit 4	Bits 3-0
Error	Reserved	Reserved	Toggle	Command Echo

Bit Definitions:

Commands:

0 Nop – No operation is performed

1 Read Object – Object Specified is read and data can be found in MUX Request Data

2 Write Object – Object Specified is written with data supplied in MUX Request Data

Toggle Echo:

This is the echo of the toggle in the request control byte.

Error:

This bit is set if there was an error performing the requested command. i.e. object doesn't exist.

Object 2A11h: MD Response Index

The MD Response Index Object allows the network to send requests to the operator interface using process data. The index object specifies the object that was written or read from using the MD Request.

OBJECT DESCRIPTION

INDEX	2A11h
Name	MD Response Index
Object Code	VAR
Data Type	UNSIGNED16
Category	Optional

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	Yes
Value Range	UNSIGNED16
Default Value	0

Object 2A12h: MD Response SubIndex

The MD Response SubIndex allows the network to get the response of requests to the operator interface using process data. The SubIndex object specifies the subindex of the object that was written to or read from.

OBJECT DESCRIPTION

INDEX	2A12h
Name	MD Response SubIndex
Object Code	VAR
Data Type	UNSIGNED8
Category	Optional

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	Yes
Value Range	UNSIGNED8
Default Value	0

Object 2A13h: MD Response Data

The MD Response Data allows the network to get data returned from requests to the operator interface using process data. The Response Data object will return the data of the object that was read to the requested index and subindex. The amount of data transferred depends on the size of the object read from.

OBJECT DESCRIPTION

INDEX	2A13h
Name	MD Response Data
Object Code	VAR
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

Access	Ro
PDO Mapping	Yes
Value Range	UNSIGNED32
Default Value	0

Operator Input Objects

Object 2B00h: Operator Input Objects

These objects provide a means for the user to configure each of the 254 operator input objects that the Operator interface supports.

Object Description

INDEX	2B00h
Name	Operator Input Object
Object Code	Array
Data Type	OPERATOR_INPUT
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Operator Inputs
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Sub-Index	1h
Description	Operator Input 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	OPERATOR_INPUT
Default Value	No

To

Sub-Index	Feh
Description	Operator Input Feh
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	OPERATOR_INPUT
Default value	No

```
typedef struct{
        UNSIGNED32 Data_Object;
        REAL32 Conversion_Factor_Units_0;
        REAL32 Conversion_Factor_Units_1;
        REAL32 Conversion_Factor_Units_2;
        UNSIGNED8 Format;
        UNSIGNED8 Digits;
        UNSIGNED8 Decimal_Places;
        UNSIGNED8 Reserved;
        }OPERATOR_INPUT;
```

Operator Output Objects

Object 2C00h: Operator Output Objects

These objects provide a means for the user to configure each of the 254 operator output objects that the Operator interface supports.

Object Description

INDEX	2C00h
Name	Operator Output Object
Object Code	Array
Data Type	OPERATOR_OUTPUT
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Operator Outputs
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	0

Sub-Index	1h
Description	Operator Output 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	OPERATOR_OUTPUT
Default Value	No

To

Sub-Index	Feh
Description	Operator Output Feh
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	OPERATOR_OUTPUT
Default value	No

```
typedef struct{
        UNSIGNED32 Data_Object;
        REAL32 Conversion_Factor_Units_0;
        REAL32 Conversion_Factor_Units_1;
        REAL32 Conversion_Factor_Units_2;
        UNSIGNED8 Format;
        UNSIGNED8 Digits;
        UNSIGNED8 Decimal_Places;
        UNSIGNED8 Reserved;
        UNSIGNED32 Status_Code_List;
        }OPERATOR_OUTPUTS;
```

Status Code Objects

Object 2D00h: Status Code Objects

This object provides a way for the user to enter status code messages that are displayed in the Operator_Output area on the screen if the mapped objects status byte is non-zero.

Object Description

INDEX	2D00h
Name	Status Code
Object Code	Array
Data Type	VISIBLE_STRING
Category Conditional	Optional

Entry Description

Sub-Index	0h
Description	Number of Status Codes
Access	Ro
Entry Category	Mandatory
PDO Mapping	No
Value Range	UNSIGNED8
Default Value	254

Sub-Index	1h
Description	Status Code 1
Access	Rw
Entry Category	Mandatory
PDO Mapping	No
Value Range	VISIBLE_STRING
Default Value	No

To

Sub-Index	Feh
Description	Status Code Feh
Access	Rw
Entry Category	Optional
PDO Mapping	No
Value Range	VISIBLE_STRING
Default value	No

Example:
Status Code 01 =”Sensor Failure”
Status Code 02 =”Over Limit”
Status Code 03 =”Under Limit”
Assume Screen is configured as
Temp=XXXXXX
where XXXXXX= Operator Output 1
Operator Output # 1 Data Object is 28000108

DataOut UNSIGNE8 Status=0
DataOut UNSIGNE8 = 220
The screen would show:
Temp= 220

DataOut UNSIGNE8 Status=1
DataOut UNSIGNE8 = 220
The screen would show:
Temp= Sensor Failure

DataOut UNSIGNE8 Status=2
DataOut UNSIGNE8 = 220
The screen would show:
Temp= Over Limit

Reading and Writing Objects using PDOs.

There are two methods for writing Objects using PDOs. One is the use of a Destination Address Mode DAM-MPDO (CANopen standard) and the other is to use the MUX Data Objects.

DAM-MPDO:
Request (RxPDO)

Byte 0	Byte 1-2	Byte 3	Byte 4-7
Node-ID	Index	SubIndex	Request Data

Example:
To write Index 0x2800 SubIndex 1 with a value of 0x01 On Node 0x0A
Send the following:

Byte 0	Byte 1-2		Byte 3	Byte 4-7			
0x0A	0x00	0x28	01	0x01	0x00	0x00	0x00

MUX Data Objects:
To read and write objects using PDOs, the MUX Data objects were created. The mechanism works with a request response method. RxPDO 2 is mapped (by default) with the Mux Request values and TxPDO 2 is mapped with the Mux Data Response values.

The default mapped PDO format is as follows:

Request (RxPDO) Format:

Byte 0	Byte 1-2	Byte 3	Byte 4-7
Control	Index	SubIndex	Request Data

Control Byte Format:

Bit 7	Bit 6	Bit 5	Bit 4	Bits 3-0
Reserved	Reserved	Reserved	Toggle	Command

Bit Definitions:
Commands:
0 Nop – No operation is performed
1 Read Object – Object Specified is read and data can be found in MUX Request Data
2 Write Object – Object Specified is written with data supplied in MUX Request Data

Toggle:
The user must toggle this bit if the device is to perform the requested command again. Optionally the user can switch commands such as write then read or write then NOP then write again.

Response (TxPDO) Format:

Byte 0	Byte 1-2	Byte 3	Byte 4-7
Status	Index	SubIndex	Response Data

Status Byte Format:

Bit 7	Bit 6	Bit 5	Bit 4	Bits 3-0
Error	Reserved	Reserved	Toggle	Command Echo

Bit Definitions:
Commands:
0 Nop – No operation is performed
1 Read Object – Object Specified is read and data can be found in MUX Request Data
2 Write Object – Object Specified is written with data supplied in MUX Request Data

Toggle Echo:
This is the echo of the toggle in the request control byte.

Error:
This bit is set if there was an error performing the requested command. i.e. object doesn't exist.

Example
To write Index 0x2800 SubIndex 1 with a value of 0x01
Send the following:

Byte 0	Byte 1-2		Byte 3	Byte 4-7			
0x02	0x00	0x28	01	0x01	0x00	0x00	0x00

The TxPDO will be updated with

Byte 0	Byte 1-2		Byte 3	Byte 4-7			
0x02	0x00	0x28	01	0x01	0x00	0x00	0x00

To Read Index 0x2601 SubIndex 0x01
Send the following:

Byte 0	Byte 1-2		Byte 3	Byte 4-7			
0x01	0x01	0x26	01	0x00	0x00	0x00	0x00

The TxPDO will be updated with

Byte 0	Byte 1-2		Byte 3	Byte 4-7			
0x01	0x01	0x26	01	Value LSB	Value MSB	0x00	0x00