User Guide

Bus - Option: CANopen (F8)

From CANopen - Version V 1.01
from COMPAX software version >V3.0

June 2000

HAUSER
We automate motion

CERTIFIED DIN EN ISO 9001 QUALITY SYSTEM
Reg. Nr. 36 38

Main Office
Parker Hannifin GmbH
EMD HAUSER
P. O. Box: 77607-1720
Robert-Bosch-Str. 22
D-77656 Offenburg, Germany
Phone: +49 (0)781 509-0
Fax: +49 (0)781 509-176
http://www.Parker-EMD.com

Great Britain:
Parker Hannifin plc
EMD Digiplan
21 Balena Close
Poole, Dorset
BH17 7DX UK
Phone: +44 (0)1202 69 9000
Fax: +44 (0)1202 69 5750
http://www.Parker-EMD.com

1 Content

2 Device assignment

This documentation applies for these devices:

- COMPAX 10XXSL with the Option F8
- COMPAX 25XXS with the Option F8
- COMPAX 45XXS with the Option F8
- COMPAX 85XXS with the Option F8

- COMPAX P1XXM with F8 option without N1 option
- COMPAX 02XXM with the Option F8
- COMPAX 05XXM with the Option F8
- COMPAX 15XXM with the Option F8
- COMPAX 35XXM with the Option F8

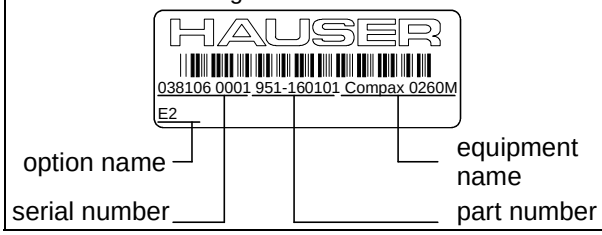
XX: any characters
F8: Option CANopen
N1: Single-phase supply (not possible simultaneous with F8)

Key to unit designation

e.g.: COMPAX 0260M:
COMPAX: name
02: performance class
60: Variant e.g. "00": Standard model
M: Model M: multi-axis unit
S: single-axis unit

HAUSER rating plate

The type plate is found on the upper side of the unit and contains the following:



3 General

CANopen provides for the use of two types of communication with different characteristics:

- Real-time data are transmitted with the process data objects (PDO).
- Parameters- and programs are transmitted via the service data objects (SDO).

The communication channels of CANopen are configured using specific communication objects.
A device profile with manufacturer-specific objects (commands) was defined for access to COMPAX. These objects can:

- be addressed as service data objects
- and partly mapped to process data objects for cyclical access.

CANopen for COMPAX

The CANopen interface enhances the flexibility of the Compact Servo-Controllers COMPAX-M and COMPAX-S. You have access to all normal COMPAX functions such as:

- Changing parameters.
- Presetting commands.
- Reading and writing control inputs and outputs.
- Writing to record storage.
- Reading status.

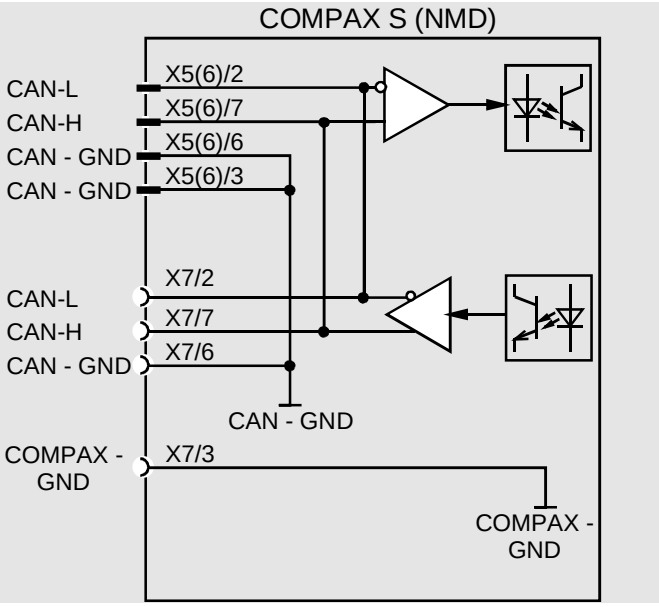
CANopen Hardware

The module (option CAN-Bus) contains the PCA82C200T BasicCAN stand-alone controller by Philips. It supports the CAN protocol in specification 1.2. The physical data cables are accessed by the Si9200 CAN transceiver by TEMIC in accordance with the CIA standard according to ISO/DIS 11898.

4 Bus Wiring

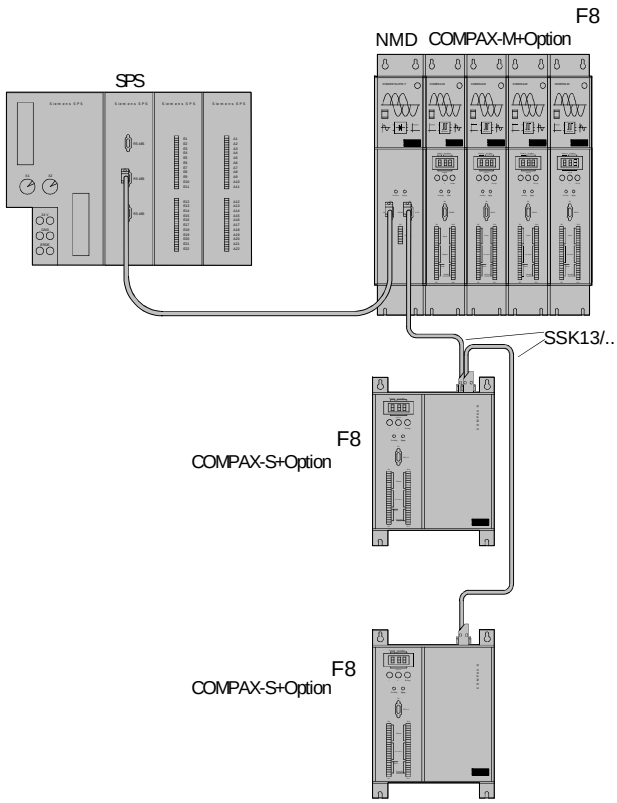
CAN-Bus connector assignment

COMPAX-S: X5 (pins) / X7 (socket)
NMD: X6 (pins) / X7 (sockets)



CAN - GND and COMPAX - GND are electrically separated.

4.1 Design



Within a system assembly consisting of COMPAX-M and network module, the Profibus signals are transmitted over the existing ribbon cable connection.

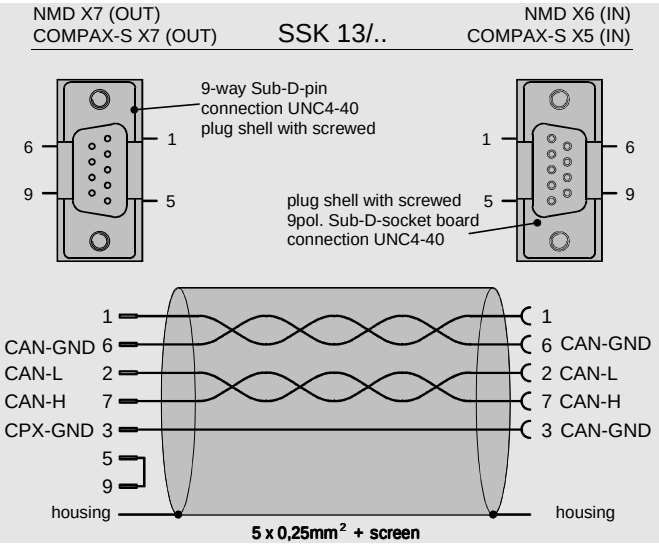
4.1 CAN-Bus components

- A computer or PLC with a CAN-Bus interface.
- Cable from the IPC or PLC to the network module or directly to **COMPAX-S**.
- Network module / COMPAX-S to Network module / COMPAX-S: SSK13/.. .
- COMPAX - units with the F8 option.

Cable plan SSK13/ :

Possible connections with SSK13/..:

- Network module X7 (OUT) → Network module X6 (IN)
- Network module X7 (OUT) → COMPAX-S/COMPAX 35XXM X5 (IN)
- COMPAX-S X7 (OUT) → Network module X6 (IN)
- COMPAX-S X7 (OUT) → COMPAX-S/COMPAX 35XXM X5 (IN)



The connection of pin 1 and pin 3 is not required for the CAN bus; neither is the bridge on the side of X7: X7/5 - X7/9.

There is electrical signal separation between the individual COMPAX.

Terminator The terminator with 120W resistance (between CAN-H and CAN-L) at the start and end of the serial cable (e.g. control and last COMPAX) is recommended.

4.2 Device Settings

CAN-Bus settings are made using COMPAX parameters; these may be set from the front panel (see next page).

4.2.1 Node ID (Device address)

Settable on every COMPAX using parameter P194; value range: 1...127.
99: Standard setting;

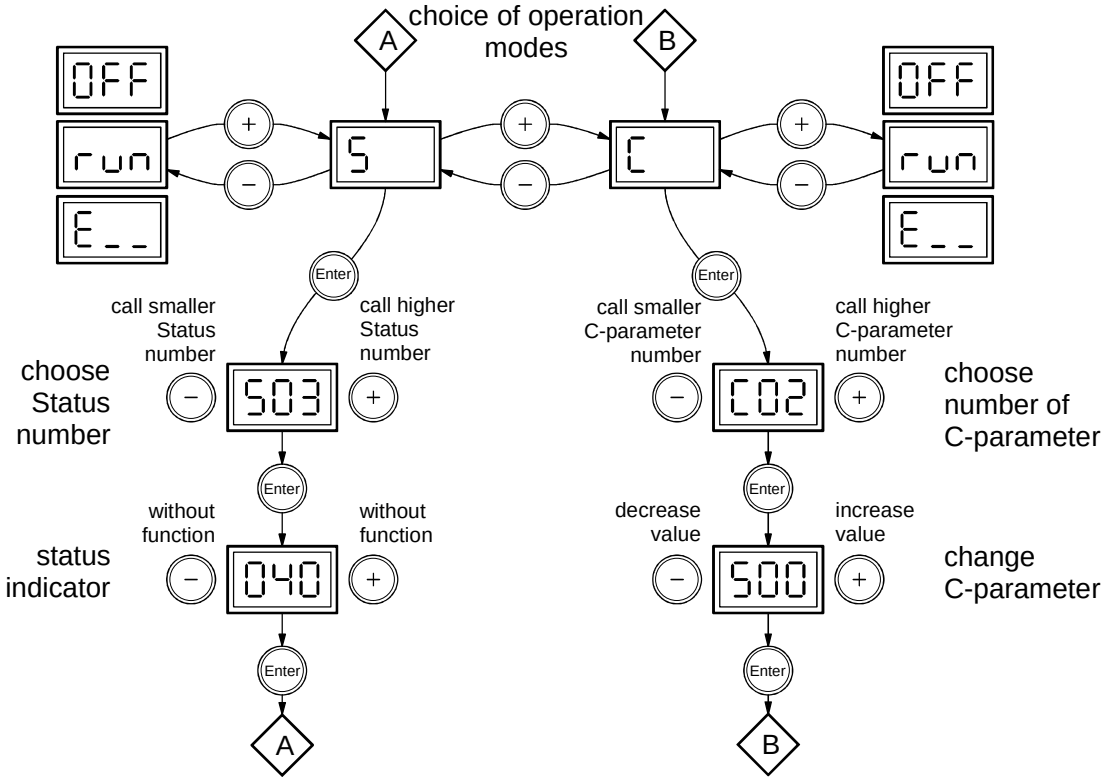
4.2.2 Baud rate

Up to 1M baud possible.
The baud rate is set with parameter P195 (unit: Bit/s). The following baud rates are possible:

20 kbit/s
50 kbit/s
100 kbit/s
125 kbit/s
250 kbit/s
500 kbit/s
800 kbit/s
1000 kbit/s

4.3 Bus setting via front panel

The bus protocol (COMPAX parameters P194, P195 and P196) can be set via the COMPAX front panel. Procedure:



Meaning:

C parameters	Meaning	Range	COMPAX parameters	Active on
C01	**Node ID (Address)**	1 ... 127	P194	**Power on**
C02	**Baud rate** (symbolic setting in accordance with the following list)	0...999	P195	**Power on**
C03	**Bus protocol**	0 ... 255	P196	**Power on**
C04 - C11	**reserved**			

Please refer to the operating instructions of the bus option used for the relevant range of values for the bus and the precise setting options.

The C - parameters are transferred with Power off / Power on!

Baud rate - setting via C02

Display value	20	50	100	125	250	500	800	999
Baud rate [Baud]	20 000	50 000	100 000	125 000	250 000	500 000	800 000	1 000 000

4.4 Data types

Physically, the types consist of one or more octets (Bytes). One byte consists of 8 bits (Bit 0 to 7). Bit 0 is the LSB (Least Significant Bit). A byte can also be depicted hexadecimally (0x00 ... 0xff).
If a data type consists of n byte, the following applies:

Data byte 1 (Byte in address x) = highest value byte
Data byte n (Byte in address x+n-1) = lowest value byte

The data coding in this chapter and the value ranges for the respective data types apply, unless otherwise explicitly stated in the data description of a COMPAX communication object.

4.4.1 Integer

Integer values are signed quantities.

Type	Value range	Length
Integer8	-128 ... +127	1 Byte
Integer16	-32 768 ... +32 767	2 Byte
Integer32	-2 147 483 647 ... +2 147 483 647 -2 147 483 648 (0x80 00 00 00) ⇨ Overflow	4 Byte

Coding	Two's complement

In objects with the data type Integer32 it is possible that its value lies outside the value range.
If this is the case, the corresponding object has the value -2 147 483 648 (0x80 00 00 00) ⇨ Overflow.
E.G. this is possible with the objects LAGE_IST or GRENZEN, as the actual travel area
of COMPAX M/S lies between +/- 4 000 000.

4.4.2 Unsigned

Unsigned values are unsigned quantities.

Type	Value range	Length
Unsigned8	0...255	1 Byte
Unsigned16	0...65 535	2 Byte
Unsigned32	0...4 294 967 295	4 Byte

Coding	binary

5 Object types

Each object type is determined by a **C**ommunication **O**bject **I**dentifier (COB-ID). This is made up of a 4 bit wide function code which represents the object type, and the 7 bit wide device address (node ID).
Communication **O**bject **I**dentifier:

Function code				**Device address (node ID): 1...127**						

The value of the COB-ID specifies the priority of the object type.
The smaller the value of the COB-ID, the higher the priority of the object type.

The following table shows the preset COB-IDs:

Communication Objects:	Function code	COB - Identifier (dec)	COB - Identifier (hex)	Defined in Index ...	Meaning	
Brodcast Objects						
NMT	0000_b	0	*0h*	-	Network management and identifier distribution	
SYNC	0001_b	128	*80h*	1005_h	SYNC is COMPAX not implemented.	
TIME	0010_b	256	*100h*	1012_h	TIME is COMPAX not implemented.	
Point to point objects						
EMCY	0001_b	129-255	81_h-FF_h	1014_h	Error message	
Boot-up	0001_b	129-255	81_h-FF_h	1014_h	Communication readiness	
T-PDO1	0011_b	385-511	181_h-$1FF_h$	1800_h	Assignment via index 1A00h	**Transmit** process data object
T-PDO2	0101_b	641-767	281_h-$2FF_h$	1801_h	Assignment via index 1A01h	(COMPAX→) max. 8 bytes
R-PDO1	0100_b	513-639	201_h-$27F_h$	1400_h	Assignment via index 1600h	**Receive** process data object
R-PDO2	0110_b	769-895	301_h-$37F_h$	1401_h	Assignment via index 1601h	(→ COMPAX) max. 8 bytes
T-SDO1	1011_b	1409-1535	581_h-$5FF_h$	1200_h	Transmit service data object1	
T-SDO2	-*	-*	-	*1201h*	Transmit service data object 2	
R-SDO1	1100_b	1537-1663	601_h-$67F_h$	1200_h	Receive service data object1	
R-SDO2	-*	-*	-	*1201h*	Receive service data object 2	
Node guard	1110_b	1793-1919	701_h-$77F_h$	$100E_h$	Checking bus subscriber.	

The COB-ID for an object is calculated as follows: **COB-ID = (function code * 128) + device address**
The default distribution of the COB-IDs can be changed via SDO after the bus is started.

Application of the communication object types

Transfer of real-time data (faster transmission, as higher priority)

T-PDO Transmit process data object: COMPAX answer (see from page 23).

R-PDO Receive process data object: Transmit to COMPAX (see from page).18

With the default setting, the transmission of the PDOs takes place at every change of its content.

One-off transfer e.g. of parameters of program lines

T-SDO Transmit service data object: COMPAX answer (see from page).15

R-SDO Receive service data object: Transmit to COMPAX (see from page).15

*SDO2 is not configured in the basic state but can be configured via the object with index 1201_h.

6 Object list

6.1 Overview of communication-specific objects

Index	Description
General objects	
1000	Device type
1001	Error register
1004	Number of PDOs
1005	COB-ID SYNC
1006	Communication cycle period
1007	Synchronization window length
1008	Device name
1009	Hardware version
100A	Software version
100B	Node number
C100	Monitoring period
D100	Lifetime factor
E100	COB-ID for node monitoring
F100	Number of SDOs
1014	COB-ID EMCY
Setting of service data objects	
1200	Server SDO1 parameter
1201	Server SDO2 parameter
Setting of process data objects (reading of COMPAX)	
1400	Communication parameters for reception PDO1 (COMPAX→)
1401	Communication parameters for reception PDO2 (COMPAX→)
1600	Mapping parameters for reception PDO1 (COMPAX→)
1601	Mapping parameters for reception PDO2 (COMPAX→)
Setting of process data objects (transmission to COMPAX)	
1800	Communication parameters for transmission PDO1 (→COMPAX)
1801	Communication parameters for transmission PDO2 (→COMPAX)
1A00	Mapping parameters for transmission PDO1 (→COMPAX)
1A01	Mapping parameters for transmission PDO2 (→COMPAX)

6.2 General objects

6.2.1 Device type

The device type specifies the type of device.
The low value 16 bits contain the device profile number. Additional information is stored in the 16 higher value bits .

Object Description

Index	0x1000				
Standard	0x00020192	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	constant	**PDO mapping**	no

6.2.2 Error register

The error register contains 8 bits, each one for a special error class.
If an error occurs at the device, only bit 0 is set.
COMPAX does not support bits 1 ... 7.

Object Description

Index	0x1001				
Standard	0x00	**Length**	1	**Min**	0x00
Object code	Variable			**Max**	0xFF
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	no

Meaning of the bits

Bit	Meaning
0	generic error
1	Power
2	Voltage
3	Temperature
4	Communication error (overrun, error state)
5	Specific device profile
6	reserved
7	manufacturer-specific

6.2.3 Number of PDOs

This object contains the number of supported PDOs.
Sub-index 0 describes the total number of PDOs.
Sub-index 1 specifies the number of synchronous and sub-index 2 the number of asynchronous PDOs.
Each of these entries is made up of two 16 bit numbers.
The higher-value 16 bits contain the number of the receiving PDOs and the lower value 16 bits the number of transmission PDOs.

Object Description

Index	0x**1004**				
Standard		**Length**	4	**Min**	0x00000000
Object code	Array	**Elements**	3	**Max**	0x01FF01FF
Data type	Unsigned32	**Access rights**	read	**PDO mapping**	no

Sub-index	Assignment	value
0	Number of all PDOs	0x00020002
1	Number of synchronous PDOs	0x00000000
2	Number of asynchronous PDOs	0x00020002

Data Description

MSB

Number of receiving PDOs	Number of transmission PDOs
0002	0002

6.2.4 COB-ID SYNC

This object contains the COB-ID of the SYNC message. If bit 31 is set, the device receives a SYNC. When bit 30 is set, it transmits the latter.
COMPAX does not support this function.

Object Description

Index	0x1005				
Standard	0x80000080	**Length**	4	**Min**	0x00000001
Object code	Variable	**Elements**		**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

6.2.5 Communication cycle period

The communication cycle period is the period between two SYNC messages in µs. It is 0 when it is not in use.
COMPAX does not support this function.

Object Description

Index	0x1006				
Standard	0x00000000	**Length**	4	**Min**	0x00000000
Object code	Variable	**Elements**		**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	µs

6.2.6 Synchronization window length

The synchronization window length is the period between the arrival of the SYNC message and the last synchronous PDO which was thereby triggered in µs.
It is 0 when it is not in use.
COMPAX does not support this function.

Object Description

Index	0x1007				
Standard	0x00000000	**Length**	4	**Min**	0x00000000
Object code	Variable	**Elements**		**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	µs

6.2.7 Device name

This object contains the device name.

Object Description

Index	0x1008				
Standard	COMPAX	**Length**		**Min**	
Object code		**Elements**		**Max**	
Data type	Visible string	**Access rights**	constant	**PDO mapping**	no

6.2.8 Hardware version

This object contains the device hardware version.

Object Description

Index	0x1009				
Standard	CPX 3.0	**Length**		**Min**	
Object code		**Elements**		**Max**	
Data type	Visible string	**Access rights**	constant	**PDO mapping**	no

6.2.9 Software version

This object contains the device software version.

Object Description

Index	0x100A				
Standard	1.00	**Length**		**Min**	
Object code		**Elements**		**Max**	
Data type	Visible string	**Access rights**	constant	**PDO mapping**	no

6.2.10 Node number (address, node ID)

The node number (address, node ID) is the module number of the device. It may only be assigned once per network.

Object Description

Index	0x100B				
Standard	0x00000020	**Length**	4	**Min**	0x00000001
Object code	Variable	**Elements**		**Max**	0x0000007F
Data type	Unsigned32	**Access rights**	read	**PDO mapping**	no

6.3 Node guarding (node guarding)

Node guarding is used for monitoring bus communication. A check is made whether a specific bus subscriber is still active. The master transmits an object after expiry of the monitoring period and receives a response from the addressed subscriber (slave). Master and slave are checked. If the master fails to receive an answer several times (in accordance with the specifications in the lifetime factor), an error is generated. If the slave (COMPAX) fails to receive a request several times (in accordance with the specifications in the lifetime factor), an error (E73) is also generated.

6.3.1 COB-ID for node guarding (node guarding)

This object contains the COB-ID for node guarding.

Object Description

Index	0x100E				
Standard	0x00000700 + node ID	**Length**	4	**Min**	0x00000001
Object code	Variable	**Elements**		**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

6.3.2 Guard period (node guarding)

The node guarding period (node guarding) is the interval in ms at which the device is interrogated by the master.
This object defines the lifetime of the device together with the lifetime factor.
If the period lifetime factor times guard period has expired, communication with the device is deemed to have been interrupted.

Object Description

Index	0x100C				
Standard	0x0000	**Length**	2	**Min**	0x0000
Object code	Variable	**Elements**		**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	ms

6.3.3 Lifetime factor (node guarding)

The lifetime factor together with the guard period defines the lifetime of the device.
When the period lifetime factor times guard period has expired, communication with the device is deemed to have been interrupted and error E73 is generated.

Object Description

Index	0x100D				
Standard	0x00	**Length**	1	**Min**	0x00
Object code	Variable	**Elements**		**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

6.4 Emergency

6.4.1 COB-ID EMCY

This object defines the COB-ID of the emergency message.

Object Description

Index	0x1014				
Standard	0x00000080 + node ID	**Length**	4	**Min**	0x00000001
Object code	Variable	**Elements**		**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

6.4.2 Emergency message

COMPAX can generate the following pop-up messages via the emergency object if **P196 Bit 0 = 0**:
☞ an error has occurred.
☞ the programmed set point has been reached.
☞ the programmed comparator point has been reached.
The pop-up messages can be enabled individually via P193 (activated).

Pop-up message	Valency
automatic error message	1
automatic "position reached" message	2
automatic comparator switch points report	4

The required settings can be obtained by inputting the sum of the significants in P193

Setting the message via P193:

P193="1": autom. error message
P193="2": autom. "position reached" message
P193="4": autom. comparator switch points report

The emergency telegram is constructed as follows:

Profile error code: The importance accords with the error code described on page 37.

Error no. of the COMPAX error (see COMPAX User Guide)

Byte	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Contents	**Emergency Error Code**		**Error Register**	**Manufacturer-specific message**				
Error	Profile error code:		0x01	Error No.	0x45	0x00	0x00	0x00
Set point reached	0x00	0xFF	0x00	0x00	0x50	0x00	0x00	0x00
Comparator point reached	0x00	0xFF	0x00	Comp. No.	0x43	0x00	0x00	0x00

0x01: error

6.5 Boot-up object

This object shows the communication readiness of the device and is sent automatically following Power on.

The boot-up object consists of the COB-ID of the EMCY object and 0 data bytes (P196 bit 1 = 0).
Optionally, P196 can be used to set Bit 1 = 1, so that the boot-up object has the same format as the EMCY object; but all data bytes have value 0 in this case.

In addition, it is possible with setting P196 bit 2 = 1 that COMPAX transmits the boot-up object as soon as it has received the START message.

6.6 Service data objects

6.6.1 Number of SDOs

This object contains the number of supported SDOs. Each of these entries is made up of two 16 bit numbers. The higher-value 16 bits contain the number of the client SDOs and the lower value 16 bits the number of server PDOs.

Object Description

Index	0x100F				
Standard	0x00000002	**Length**	4	**Min**	0x00010000
Object code	Variable	**Elements**		**Max**	0x00800080
Data type	Unsigned32	**Access rights**	read	**PDO mapping**	no

Data Description

MSB

Number of client SDOs	Number of server SDOs
0000	0002

6.6.2 Server SDO1 parameters

This object describes the SDO1 parameters for which the device is server.

Object Description

Index	0x1200				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	3	**Max**	-
Data type	-	**Access rights**	read	**PDO mapping**	no

Number of entries

Index: subindex	0x1200:000				
Standard	0x02	**Length**	1	**Min**	0x02
Object code	-	**Elements**	-	**Max**	0x03
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	no

COB-ID Client -> Server: R-SDO1 (receiving data) (→ COMPAX)

Index: subindex	0x1200:001				
Standard	0x00000600 + node ID	**Length**	4	**Min**	0x00000001
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read	**PDO mapping**	no

COB-ID Server -> Client: T-SDO1 (transmission data) (COMPAX →)

Index: subindex	0x1200:002				
Standard	0x00000580 + node ID	**Length**	4	**Min**	0x00000001
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read	**PDO mapping**	no

6.6.3 Server SDO2 parameters

This object describes the SDO2 parameters for which the device is server.
SDO2 is switched off by the standard value 0x8... .

Object Description

Index	0x1201				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	3	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x1201:000				
Standard	0x02	**Length**	1	**Min**	0x02
Object code	-	**Elements**	-	**Max**	0x03
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	no

COB-ID Client -> Server: R-SDO1 (receiving data) (→ COMPAX)

Index: subindex	0x1201:001				
Standard	0x800006E0	**Length**	4	**Min**	0x00000001
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

COB-ID Server -> Client: T-SDO1 (transmission data) (COMPAX →)

Index: subindex	0x1201:002				
Standard	0x800006E0	**Length**	4	**Min**	0x00000001
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

6.7 Transmission of process data to COMPAX

6.7.1 Communication parameters for receiving PDO1 (R-PDO1)

This object describes the communication parameters of the first receiving PDO.
Sub-index 0 contains the highest implementable sub-index number. Sub-index 1 describes the COB-ID. When bit 31 of this entry is set, the PDO is deactivated.
The transmission mode is defined by sub-index 2 (default=FE: Asynchronous transmission of the PDO, triggered by a change in the value to be transmitted). A blocking period with a resolution of 100 µs can be parameterized on sub-index 3.
The fourth sub-index specifies the priority class of the PDO.

Object Description

Index	0x1400				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	4	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x1400:000				
Standard	0x03	**Length**	1	**Min**	0x02
Object code	-	**Elements**	-	**Max**	0x04
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	no

COB-ID (R-PDO1)

Index: subindex	0x1400:001				
Standard	0x00000200 + node ID	**Length**	4	**Min**	0x00000001
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Transmission mode

(default=FE: Asynchronous transmission of the PDO, triggered by a change of the value to be transmitted)

Index: subindex	0x1400:002				
Standard	0xFE	**Length**	1	**Min**	0x00
Object code	-	**Elements**	-	**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

Blocking period in 100µs (1=100µs)

(Minimum time between two PDO transmissions)

Index: subindex	0x1400:003				
Standard	0x0000	**Length**	2	**Min**	0x0000
Object code	-	**Elements**	-	**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	no

6.7.2 Communication parameters for receiving PDO2 (R-PDO2)

This object describes the communication parameters of the second receiving PDO.
Sub-index 0 contains the highest implementable sub-index number. Sub-index 1 describes the COB-ID. When bit 31 of this entry is set, the PDO is deactivated.
The transmission mode is defined by sub-index 2. A blocking period with a resolution of 100 µs can be parameterized on sub-index 3.
The fourth sub-index specifies the priority class of the PDO.

Object Description

Index	0x1401				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	4	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x1401:000				
Standard	0x03	**Length**	1	**Min**	0x00
Object code	-	**Elements**	-	**Max**	0xFF
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	no

COB-ID (R-PDO2)

Index: subindex	0x1401:001				
Standard	0x00000300 + node ID	**Length**	4	**Min**	0x00000001
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Transmission mode

(default=FE: Asynchronous transmission of the PDO, triggered by a change in the value to be transmitted).

Index: subindex	0x1401:002				
Standard	0xFE	**Length**	1	**Min**	0x00
Object code	-	**Elements**	-	**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

Blocking period in 100µs (1=100µs)

(Minimum time between two PDO transmissions)

Index: subindex	0x1401:003				
Standard	0x0000	**Length**	2	**Min**	0x0000
Object code	-	**Elements**	-	**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	no

6.7.3 Mapping-Parameter receiving PDO1 (R-PDO1)

This entry contains information about the data to be received on this PDO (mapping).
Sub-index 0 contains the number of mapped objects.
A maximum of 5 objects with a total length of 8 bytes can be mapped.
All other indices contain the index, sub-index and the length of the individual objects.
After Power on, the object CONTROL WORD is automatically entered.
P139 and P140 can be used to define two other objects which are entered after Power on.

Object Description

Index	0x1600				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	5	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x1600:000				
Standard	0x01	**Length**	1	**Min**	0x00
Object code	-	**Elements**	-	**Max**	0x05
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry of the first object (links, MSB) (R-PDO1) (0x6040: Index of control word)

Index: subindex	0x1600:001				
Standard	0x60400010	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry (R-PDO1)

Index: subindex	0x1600:002				
Standard	Value from P139	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry (R-PDO1)

Index: subindex	0x1600:003				
Standard	Value from P140	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

The structure of each mapping entry is:

Index	Sub-index	Length
xxx	xx	xx

Example

6040	00	10
Index of control word	Subindex = 00	Length 2 bytes

COMPAX objects which can be used as receiving PDOs (R-PDOs)

Object	Access	Index	Sub-index	COMPAX parameters P139-P142	Length	PDO	See page
Control byte	rd/wr	0x2000	0	2097152	1	R	28
COMPAX control word	rd/wr	0x2002	0	2097664	2	R	29
Control commands	wr	0x2004	0	2098176	1	R	32
Read/write parameters in binary format	rd/wr	0x2006	1...250	2098689-2098938	4	R	33
Set point speed and position default	wr	0x2023	0	2106112	6	R	45
Traverse speed	rd/wr	0x2030	0	2109440	2	R	46
Reduce traverse speed	rd/wr	0x2031	0	2109696	1	R	46
Run program record N	wr	0x2052	0	2118144	1	R	62
Start program from record N	wr	0x2053	0	2118400	1	R	62
Enter or read COMPAX variables	rd/wr	0x2057	1...40	2119425-2119464	4	R	64
Control word	rd/wr	0x6040	0	6307840	2	R	32
Position set point default	rd/wr	0x607A	0	6322688	4	R	41
Logic state of the 16 digit. Outputs	rd/wr	0x6300	0	6488064	2	T/R	52

⇨ When assigning a PDO observe the respective length of the objects!

⇨ Processing of the received process output data (R-PDOs) can be influenced as described below.

6.7.4 R-PDO control

This object can be used to influence the internal processing of the received process output data (R-PDOs).
A bit is allocated to each communication object which can be mapped to a R-PDO .
If the corresponding bit = 0, the received data for the object are used immediately; i.e. the resulting command is executed immediately.
If the corresponding bit = 1, a check is first made whether there has been a data change for the corresponding object, and only then are the data used or the resulting command generated.
Bit 15 = 1 achieves that PA data for the LAGE_ZIEL are used, but the running move is only carried out with the flank reversal (0->1) of the New Set Point Value bit (control word bit 4).

Object Description

Index	0x200F				
Symbol	PAD_CONTROL	**Length**	2	**Min**	0x8000
Object code	Variable			**Max**	0x7FFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	yes

Data Description

Bit	Assignment	Bit	Assignment
15	Position set point default only with New Set Point bit	7	Control word
14	-	6	Program start beginning at record N
13	-	5	Execute program record N
12	Set point speed and position default	4	Reduce traverse speed
11	Enter or read COMPAX variables	3	Traverse speed
10	Read/write parameters in binary format	2	Control commands
9	Logic state of the 16 digit. outputs	1	COMPAX control word
8	Target position default	0	Control byte

6.7.5 Mapping parameter for receiving PDO2 (R-PDO2)

This entry contains information about the data to be received on this PDO (mapping).
Sub-index 0 contains the number of mapped objects.
A maximum of 5 objects with a total length of 8 bytes can be mapped.
All other indices contain the index, sub-index and the length of the individual objects.
After Power on, no object is entered automatically as the default value 0 is entered under number of entries.
P141 and P142 can be used to define two objects which are mapped after Power on.

Object Description

Index	0x1601				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	5	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x1601:000				
Standard	0x00	**Length**	1	**Min**	0x00
Object code	-	**Elements**	-	**Max**	0x05
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry of the first object (links, MSB) (R-PDO2)

Index: subindex	0x1601:001				
Standard	Value from P141	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry (R-PDO2)

Index: subindex	0x1601:002				
Standard	Value from P142	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

The structure of each mapping entry is:

Index	Sub-index	Length
xxx	xx	xx

6.8 Reading process data from COMPAX

6.8.1 Communication parameters for transmission PDO1 (T-PDO1)

This object describes the communication parameters of the first transmission PDO.
Sub-index 0 contains the highest implementable sub-index number. Sub-index 1 describes the COB-ID. When bit 31 of this entry is set, the PDO is deactivated.
The transmission mode is defined by sub-index 2. A blocking period with a resolution of 100 µs can be parameterized on sub-index 3.
The fourth sub-index specifies the priority class of the PDO.

Object Description

Index	0x1800				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	4	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x1800:000				
Standard	0x03	**Length**	1	**Min**	0x02
Object code	-	**Elements**	-	**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

COB-ID (T-PDO1)

Index: subindex	0x1800:001				
Standard	0x00000180 + node ID	**Length**	4	**Min**	0x00000001
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Transmission mode

(default=FE: Asynchronous transmission of the PDO, triggered by a change in the value to be transmitted).

Index: subindex	0x1800:002				
Standard	0xFE	**Length**	1	**Min**	0x00
Object code	-	**Elements**	-	**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

Blocking period in 100µs (1=100µs)

(Minimum time between two PDO transmissions)

Index: subindex	0x1800:003				
Standard	0x0000	**Length**	2	**Min**	0x0000
Object code	-	**Elements**	-	**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	no

6.8.2 Communication parameters for transmission PDO2 (T-PDO2)

This object describes the communication parameters of the second transmission PDO.
Sub-index 0 contains the highest implementable sub-index number. Sub-index 1 describes the COB-ID. When bit 31 of this entry is set, the PDO is deactivated.
The transmission mode is defined by sub-index 2. A blocking period with a resolution of 100 µs can be parameterized on sub-index 3.
The fourth sub-index specifies the priority class of the PDO.

Object Description

Index	0x1801				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	4	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x1801:000				
Standard	0x03	**Length**	1	**Min**	0x02
Object code	-	**Elements**	-	**Max**	0x04
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

COB-ID (T-PDO2)

Index: subindex	0x1801:001				
Standard	0x00000280 + node ID	**Length**	4	**Min**	0x00000001
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Transmission mode

(default=FE: Asynchronous transmission of the PDO, triggered by a change in the value to be transmitted).

Index: subindex	0x1801:002				
Standard	0xFE	**Length**	1	**Min**	0x00
Object code	-	**Elements**	-	**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

Blocking period in 100µs (1=100µs)

(Minimum time between two PDO transmissions)

Index: subindex	0x1801:003				
Standard	0x0000	**Length**	2	**Min**	0x0000
Object code	-	**Elements**	-	**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	no

6.8.3 Mapping parameters for transmission PDO1 (T-PDO1)

This entry contains information about the data to be transmitted on this PDO (mapping).
Sub-index 0 contains the number of mapped objects.
A maximum of 5 objects with a total length of 8 bytes can be mapped.
All other indices contain the index, sub-index and the length of the individual objects.
After Power on, the object STATUS WORD is automatically entered.
P135 and P136 can be used to define two other objects which are mapped after Power on.

Object Description

Index	0x1A00				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	5	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x1A00:000				
Standard	0x01	**Length**	1	**Min**	0x00
Object code	-	**Elements**	-	**Max**	0x05
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry of the first object (links, MSB) (T-PDO1) (6041: Index of status word)

Index: subindex	0x1A00:001				
Standard	0x60410010	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry (T-PDO1)

Index: subindex	0x1A00:002				
Standard	^	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry (T-PDO1)

Index: subindex	0x1A00:003				
Standard	Value from P136	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

The structure of each mapping entry is:

Index	Sub-index	Length
xxx	xx	xx

COMPAX objects which can be used as transmission PDOs

Object	Access	Index	Sub-index	COMPAX parameters P135-P138	Length	PDO	See page
Status byte	rd	0x2001	0	2097408	1	T	28
COMPAX status word	rd	0x2003	0	2097920	2	T	30
Current motor torque	rd	0x2010	0	2101248	2	T	39
Contour error	rd	0x2021	0	2105600	2	T	44
Current axis process speed	rd	0x2035	0	2110720	2	T	49
Status word	rd	0x6041	0	6308096	2	T	31
Position actual value	rd	0x6064	0	6317056	4	T	41
Logic state of the 16 digit. inputs	rd	0x6100	0	6356992	2	T	52
Logic state of the 16 digit. Outputs	rd/wr	0x6300	0	6488064	2	T/R	52

When assigning a PDO observe the respective length of the objects!

6.8.4 Mapping parameters for transmission PDO2 (T-PDO2)

This entry contains information about the data to be transmitted on this PDO (mapping).
Sub-index 0 contains the number of mapped objects.
A maximum of 5 objects with a total length of 8 bytes can be mapped.
All other indices contain the index, sub-index and the length of the individual objects.
P137 and P138 can be used to define two objects which are mapped after Power on.

Object Description

Index	0x1A01				
Standard		**Length**	-	**Min**	-
Object code	Array	**Elements**	5	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x1A01:000				
Standard	0x00	**Length**	1	**Min**	0x00
Object code	-	**Elements**	-	**Max**	0x05
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry of the first object (links, MSB) (T-PDO2)

Index: subindex	0x1A01:001				
Standard	Value from P137	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

PDO mapping entry (T-PDO2)

Index: subindex	0x1A01:002				
Standard	Value from P138	**Length**	4	**Min**	0x00000000
Object code	-	**Elements**	-	**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

The structure of each mapping entry is:

Index	Sub-index	Length
xxx	xx	xx

6.9 Control

6.9.1 Selection code operating mode

Selection function to determine the operating mode.

Object Description

Index	0x6060				
Symbol	OPERATING MODE	**Length**	1	**Min**	0x80
Object code	Simple var.			**Max**	0x7F
Data type	Integer8	**Access rights**	Read/write all	**PDO mapping**	no

Data Description

Selection code	Operation mode	Selection code	Operation mode
-2	Reset mode	1	Target position input
-1	Continuous mode	3	Speed input 2

6.9.2 Operating mode display

This parameter returns the actual operating mode.
The meaning of the displayed value corresponds to the operation mode selection code.

Object Description

Index	0x6061				
Symbol	BETR_ART_AZ	**Length**	1	**Min**	0x80
Object code	Simple var.			**Max**	0x7F
Data type	Integer8	**Access rights**	Read all	**PDO mapping**	no

6.9.3 Control byte

Allows program start from record 1 - 15.
The record pointer is set to the corresponding program record.
The program can be started, stopped, and continued.
⇨ On record selection = "0000" and "Start", machine zero is set.
An error acknowledgment is possible.

Object Description

Index	0x2000				
Symbol	STEUERBYTE	**Length**	1	**Min**	0
Object code	Variable			**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	yes

Data Description

Bit	Assignment	Meaning
7	Quit (not when P190=22)	Error acknowledge with a positive edge
6	Stop with ramp P10 without clearing	with a positive edge
5	Continue ("1") / new start ("0") with Start	Continue: Continue program New start: Prog. start at selected record
4	Start / Stop	↑ Start after the response defined in Bit 5 ↓ Stop
3	Record select (2^3)	**Note!** With the record selection = "0000" and "New start", the machine zero point is approached
2	Record select (2^2)	
1	Record select (2^3)	
0	Record select (2^0)	

⇨ The "Quit" command is not accepted with P190=22 (Device Profile DSP402) (error confirmation in accordance with the status diagram on page 66).

6.9.4 Status byte

The status byte shows information about the status of the device as well as messages.

Object Description

Index	0x2001				
Symbol	STATUSBYTE	**Length**	1	**Min**	0
Object code	Variable			**Max**	0xFF
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	yes

Data Description

Bit	Assignment
7	Machine zero point approached
6	Idle after stop
5	Programmed nominal position reached
4	CPX_ZSW Bit 7
3	CPX_ZSW Bit 8
2	Ready for start (see below)
1	Warning
0	Fault

Meaning of "Ready for Start"
"Ready for start" is used for program control and is set:
- ☞ when the program is halted by a WAIT START instruction and is waiting for the START signal.
- ☞ after an interruption with STOP or BREAK and these signals no longer occur.
- ☞ after a rectified error.
- ☞ after power on.
- ☞ at the program end with the command END.

"Ready for Start" has no meaning when commands are input directly.

6.9.5 COMPAX control word

Activation of device control commands and setting/resetting of virtual inputs E17...E32
Control of the COMPAX M/S via the COMPAX control word is only possible when the corresponding bits have been set in P221.
The virtual inputs E17...E32 can be queried in the data memory.

Object Description

Index	0x2002				
Symbol	CPX_STW	**Length**	2	**Min**	0
Object code	Variable			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	yes

Data Description

Bit	Assignment	Bit	Assignment
15 ... 8	E32...E25	7...0	E24...E17

Bit	Function without shift	Function with shift	Enable
15 ... 8	none	none	P222/Bit 7 ... 0
7...6	none	none	P221/Bit 7 ... 6
5	STOP	BREAK	P221/Bit 5
4	START	none	P221/Bit 4
3	QUIT (not when P190=22)	Teach real zero	P221/Bit 3
2	Hand-	Approach real zero (RZ)	P221/Bit 2
1	Hand+	Approach machine zero (MZ)	P221/Bit 1
0	SHIFT		P221/Bit 0

The "Quit" command is not accepted with P190=22 (Device Profile DSP 402) (error confirmation in accordance with the status diagram on page 66).402

The partial switching of the input functions to the control word restricts functions through the multiple functioning of E1: Example: If a function is placed with the control word with E1 (e. g. teach real zero), then other functions with E1 (as well as the "QUIT" function) are blocked via the inputs.

Therefore: If you need all the input functions, the function must be completely reassigned, either to the inputs (P221 = 0) or to the control word (P221 = 63).

COMPAX I/O functions via the COMPAX control word

The direct switching of the I/O functions through removal of a function and simultaneous setting of another function is not recognized by COMPAX; **Exception:** STOP and BREAK (these are always recognized immediately).
Therefore proceed as follows:

- Remove current functions
- Then set new functions.

Example: Switch from Hand+ to Hand-

- Reset Hand+
- Reset Hand-

6.9.6 CPX_ZSW

The CPX status word shows information about the status of the device as well as messages.
From COMPAX Software Version 3.64 or higher the status information S16 and S17 can be placed on CPX_ZSW with parameter 203 bit 0 = 1.

Object Description

Index	2003				
Symbol	CPX_ZSW	**Length**	2	**Min**	0x0000
Object code	Variable			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	Read all	**PDO mapping**	yes

Data description (P203 Bit 0 = 0)

Data byte [Bit]	Assignment	Data byte [Bit]	Assignment
1 [7]	Status A16	2 [7]	Motor stalled
1 [6]	Status A15	2 [6]	Lag error
1 [5]	Status A14	2 [5]	Idle after stop
1 [4]	Status A13	2 [4]	Target position reached
1 [3]	Status A12	2 [3]	Ready for start
1 [2]	Status A11	2 [2]	MN was reached
1 [1]	Status A10	2 [1]	No warning
1 [0]	Status A9	2 [0]	No fault

Data description (P203 Bit 0 = 1)

Data byte [Bit]	Assignment	Data byte [Bit]		Assignment
1 [7]	---	**2 [7]**	**2 [6]**	**OUTPUT A0 = x**
		0	0	after OUTPUT A0 = 0
1 [6]	RUN ("0"= off or turned off in the event of an error)	0	1	after OUTPUT A0 = 1
		1	0	after OUTPUT A0 = 2
1 [5]	reserved	2 [5]		Idle after stop
1 [4]	Stop over input E6	2 [4]		Target position reached
1 [3]	Program memory running	2 [3]		Ready for start
1 [2]	Command active	2 [2]		MN was reached
1 [1]	Service password active	2 [1]		No warning
1 [0]	Password 302 active	2 [0]		No fault

6.9.7 Control word

Some bits in the control word influence the status equipment of the device control, enabling functions and determining the operating status of the devices.
The manufacturer-specific bits (data byte 1 bit 2...6) serve to activate device control commands which are only, however, effective if the corresponding release was activated with P221 (see COMPAX control word).
The SPM bit (Data byte 1 bit 7) is used to acknowledge an existing pop-up message.

Object Description

Index	0x6040				
Symbol	STEUERWORT	**Length**	2	**Min**	0x8000
Object code	Variable			**Max**	0x7FFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	yes

Data Description

Bit	Assignment	Bit	Assignment
15	SPM	7	Reset fault
14	Approach machine zero (MZ)	6	Target position relative or absolute
13	START	5	
12	Hand-	4	New set point
11	Hand+	3	Enable operation
10	reserved	2	Rapid stop
9	reserved	1	Disable voltage
8	STOP	0	Power on

6.9.8 Status word

Information about the status of the device as well as messages are displayed in the status word,
if SPM (status word) equals SPM (control word).
The status word displays pop-up messages, when SPM (status word) is not equal to SPM (control word).

Object Description

Index	0x6041				
Symbol	STATUSWORT	**Length**	2	**Min**	0
Object code	Variable			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read	**PDO mapping**	yes

Data Description

Bit	Assignment	Bit	Assignment
15	SPM	7	Warning
14	Idle after stop	6	Switch on disabled
13	MN was reached	5	Rapid stop
12	Actual value acknowledgment	4	Voltage disabled
11	Limit value	3	Fault
10	Set point reached	2	Operation enabled
9	Remote	1	Switched on
8	Ready for start	0	Ready to start

6.9.9 Pop-up message processing

COMPAX can generate the following pop-up message via the status word (if **P196Bit 0 = 1**):
☞ an error has occurred.
☞ the programmed set point has been reached.
☞ the programmed comparator point has been reached.
The pop-up messages can be switched on (activated) individually via P193 (see page 68).

Pop-up message	Valency
automatic error message	1
automatic "position reached" message	2
automatic comparator switch points report	4

Control commands

⇨ The required settings can be obtained by inputting the sum of the significants in P193

If the pop-up message processing is active and there is a pop-up message, COMPAX interrupts the normal status display in the status word, toggles the pop-up message flag "SPM" and displays the actual pop-up message in the status word.
The Master accepts the pop-up message and acknowledges it by toggling the "SPM" in the control word.
⇨ The status word displays a pop-up message, when SPM (status word) is not equal to SPM (control word).

Status word with pop-up messages

Bit	Assignment	Data byte [Bit]	Assignment
15	SPM	7	
14		6	
13	Pop-up message identifier	5	Error No.
12		4	or
11	1 = error	3	Comparator No.
10	2 = set point reached	2	
9	3 = Comparator point reached	1	
8		0	

6.9.10 Control commands

Compax commands not requiring any additional values.
The required commands are activated by reading in the relevant command number.

Object Description

Index	0x2004				
Symbol	CONTROL	**Length**	1	**Min**	1
Object code	Variable			**Max**	19
Data type	Unsigned8	**Access rights**	write	**PDO mapping**	yes

Data Description

Command No.	Function	Command No.	Function
1	Go to machine home	11	Drive dead with closed brake
2	Program start	12	Drive dead with opened brake
3	Stop program/positioning	13	Program jump via external inputs
4	Break off program/positioning	14	Deactivate password protection (GOTO 302)
5	Acknowledge error	15	Activate password protection(GOTO 270)
6	Read current position as real null	16	Deactivate password protection (GOTO 620)
7	Declare valid	17	Declare curve valid
8	Declare configuration valid	18	Not-Stop with clear
9	Traverse speed from external encoder	19	Not-Stop without clear
10	Drive under torque with opened brake	20-24	Not assigned

6.9.11 COMPAX commands

All COMPAX commands which exist for the RS232 interface can be transmitted with this object in plain language (as ASCII string in capital letters).

Object Description

Index	0x2005				
Symbol	COMMAND	**Length**	20	**Min**	
Object code	Variable			**Max**	
Data type	Visible string	**Access rights**	read/write	**PDO mapping**	no

Data Description

Coding	ASCII	**Value range**	0x20 ... 0x7f

Data byte	Assignment	Data byte	Assignment
1	1. character of the command string	20	20. character of the command string

6.9.12 COMPAX parameters

Entering or reading COMPAX parameters in binary format.
The corresponding parameter is selected using the Sub-index (Sub-index = parameter No.).

Object Description

Index	0x2006				
Symbol	Px	**Length**	4	**Min**	0x80000000
Object code	Array	**Elements**	250	**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	read/write	**PDO mapping**	yes

Data Description

Parameter	Resolution	Parameter	Resolution
001 .. 005	1 ⇔ 0.001	035 .. 036	1 ⇔ 0.000001
006 .. 010	1	037 .. 049	1 ⇔ 0.001
011 .. 016	1 ⇔ 0.001	050 .. 072	1
017 .. 020	1	073 .. 099	1 ⇔ 0.001
021 .. 022	1 ⇔ 0.000001	100 .. 186	1
023 .. 029	1	187 .. 196	1 ⇔ 0.001
030 .. 034	1 ⇔ 0.001	197 .. 250	1

6.10 Diagnostics

6.10.1 Status S6

Temperature of the power output stage.

Object Description

Index	0x2011				
Symbol	S6	**Length**	2	**Min**	0
Object code	Variable			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read	**PDO mapping**	no

Data Description

Unit	degrees Celsius	**Resolution**	1 ⇔ 0.1°C

6.10.2 Status S9

Number of axis motion cycles.

Object Description

Index	0x2013				
Symbol	S9	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read	**PDO mapping**	no

6.10.3 Status S10

COMPAX operating hours

Object Description

Index	0x2014				
Symbol	S10	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read	**PDO mapping**	no

Data Description

Unit	Hours	**Resolution**	1 ⇔ 0.1 h

6.10.4 Status S11

Loop counter for a running REPEAT loop.

Object Description

Index	0x2015				
Symbol	S11	**Length**	2	**Min**	0
Object code	Variable			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read	**PDO mapping**	no

6.10.5 Status S16_S17

Status bits 1 and status bits 2.

Object Description

Index	0x2016				
Symbol	S16_S17	**Length**	1	**Min**	0x00
Object code	Array	**Elements**	2	**Max**	0xFF
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	no

Data Description

Bit	Assignment
7 ... 0	Status bits

Sub-index	Assignment	Sub-index	Assignment
1	Status bits1	2	Status bits 2

The meaning of the status bits can be found in the COMPAX-M/S user guide!

6.10.6 Status S23_S26

Status of the drive, the switch, the limits and the output stage.

Object Description

Index	0x2017				
Symbol	S23_S26	**Length**	1	**Min**	0x00
Object code	Array	**Elements**	4	**Max**	0xFF
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	no

Data Description

Bit	Assignment
7 ... 0	Status bits

Sub-index	Assignment	Sub-index	Assignment
1	Status drive	3	Status limits
2	Status switch	4	Status final step

The meaning of the status bits can be found in the COMPAX-M/S user guide!

6.10.7 Status S30

Error message.
This object contains the error number of the current error and the last occurring error.
If the error number of the current error = 0, there is no error.

Object Description

Index	0x2018				
Symbol	S30	**Length**	1	**Min**	0
Object code	Array	**Elements**	2	**Max**	0xFF
Data type	Unsigned8	**Access rights**	Read all	**PDO mapping**	no

Data Description

Sub-index	Assignment	Sub-index	Assignment
1	Error number of current error	2	Error number of last error

6.10.8 Error

Fault code. If COMPAX is in the "Fault" status, the fault code is given a value not equal to 0.
If COMPAX is not in the "Fault" status, this object then gives the value 0.

Object Description

Index	0x603f				
Symbol	STOERUNG	**Length**	2	**Min**	0x8000
Object code	Variable			**Max**	0x7FFF
Data type	Unsigned16	**Access rights**	read	**PDO mapping**	no

Data Description

Bit	Coding	Bit	Coding
15 ... 12	Main groups	7...0	Details
11...8	Subgroups		

Fault codes:

Code	COMPAX error
0000h	E00 Stop/no error
6320h	E01 Not configured
6320h	E02 Limits
1000h	E03
1000h	E04
7300h	E05 MN-Ini not found
1000h	E06
8200h	E07 Processor error
8200h	E08 Synchronous stop is enabled
8200h	E09 Drive not working
8611h	E10 Contour error too large
8600h	E11 Prog. position not reached
7320h	E12 Slip error
7300h	E13 Ini 1 not damped
7300h	E14 Ini 2 not bed.
7320h	E15 Error in the 2. position measuring system
6300h	E16 Record number does not exist
6300h	E17 Record number too large
6300h	E18 Record 250 is assigned
6300h	E19 No room in memory
8612h	E20 Target position behind pos. limit stop
8612h	E21 Target position behind neg. limit stop
8500h	E22 MN is not approached
6200h	E23 Command is not allowed
8400h	E24 Speed is invalid
8600h	E25 Position is invalid
6300h	E26 END command missing for REPEAT
6320h	E27 Parameter cannot be written to
1000h	E28
6320h	E29 Motor values missing
5500h	E30 Hardware fault
6320h	E31 Parameter error
6320h	E32 Parameter error
6300h	E33 Data memory error
6300h	E34 Data memory error
5500h	E35 Hardware fault
5500h	E36 Hardware fault
5111h	E37 Auxiliary voltage +15V missing
5120h	E38 Voltage in the intermediate circuit too high
4210h	E39 Temperature too high
1000h	E40

Code	COMPAX error
5410h	E41 Output stage signals error
7303h	E42 Resolver error
2300h	E43 Output loaded
5111h	E44 Pos. auxiliary voltage outside tolerance
5111h	E45 neg. auxiliary voltage outside tolerance
5112h	E46 24V too large
5112h	E47 24V too small
4310h	E48 Thermostatic switch motor indicates error
7121h	E49 Motor/drive indicates disabling
8612h	E50 Limit switch 1 activated
8612h	E51 Limit switch 2 activated
7200h	E52 Error in emergency stop controlling
7120h	E53 Motor loaded
8400h	E54 Speed too high
8000h	E55 External emergency stop
8000h	E56 Emergency stop directly in COMPAX M
5120h	E57 Voltage in the intermediate circuit too low
4200h	E58 Temperature getting too high
1000h	E59
7200h	E60 Slip warning
1000h	E61
1000h	E62
1000h	E63
1000h	E64
7305h	E65 Encoder module not enabled
1000h	E66
1000h	E67
1000h	E68
1000h	E69
1000h	E70
1000h	E71...E255

6.11 Torque, power, voltage

6.11.1 Maximum permissible motor torque

Max. torque value.
This value is the maximum permissible torque for the motor.

Object Description

Index	0x6072				
Symbol	MOMENT_MAX	**Length**	2	**Min**	0x0000
Object code	Variable			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	Per-thousands of motor rated torque	**Resolution**	1 ⇔ 1 per-thousand

6.11.2 Motor rated torque

Rated torque motor.
This value can be found on the rating plate of the motor.

Object Description

Index	0x6076				
Symbol	NENNMOMENT	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	mNm	**Resolution**	1 ⇔ 1 mNm

6.11.3 Torque actual value

Torque actual value.
The torque actual value corresponds to the current torque in the drive motor.

Object Description

Index	0x6077				
Symbol	MOMENT_IST	**Length**	2	**Min**	0x8000
Object code	Variable			**Max**	0x7FFF
Data type	Integer16	**Access rights**	read	**PDO mapping**	no

Data Description

Unit	Per-thousands of motor rated torque	**Resolution**	1 ⇔ 1 per-thousand

6.11.4 Status S5

Current motor torque.
Value in % of the rated torque.

Object Description

Index	0x2010				
Symbol	S5	**Length**	2	**Min**	0x8000
Object code	Variable			**Max**	0x7FFF
Data type	Integer16	**Access rights**	read	**PDO mapping**	yes

Data Description

Unit	%	**Resolution**	1 ⇔ $^{1}/_{64}$%; (6400 ⇔ 100%)

6.11.5 Motor nominal current

Motor nominal current.
This value can be found on the rating plate of the motor.

Object Description

Index	0x6075				
Symbol	NENNSTROM	**Length**	4	**Min**	0
Object code	Variable			**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	Ampere	**Resolution**	1 ⇔ 1 mA

6.11.6 Intermediate circuit voltage

Intermediate circuit voltage
This parameter describes the current intermediate circuit voltage in the drive controller.

Object Description

Index	0x6079				
Symbol	ZWK_SPG	**Length**	2	**Min**	0
Object code	Variable			**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read	**PDO mapping**	no

Data Description

Unit	Volt	**Resolution**	1 ⇔ 1 mV

6.11.7 Status S7_S8

Control voltage and power or intermediate circuit voltage.

Object Description

Index	0x2012				
Symbol	S7_S8	**Length**	2	**Min**	0x0000
Object code	Array	**Elements**	2	**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read	**PDO mapping**	no

Data Description

Unit	Volt	**Resolution**	1 ⇔ 0.1 V

Sub-index	Assignment	Sub-index	Assignment
1	Control voltage	2	intermediate circuit voltage

6.12 Positioning

6.12.1 Position target

Position set point default
Absolute positioning. Reference point is the real null (RN) or relative positioning:
adjustable via the control word bit 6 (0 = absolute; 1 = relative).
Positioning is done with the set acceleration time (deceleration time) and the set speed. If these values were not set, then valid are **substitute values:**
Velocity: Parameter P002; acceleration time: Parameter P006

Object Description

Index	0x607a				
Symbol	LAGE_ZIEL	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	read/write	**PDO mapping**	yes

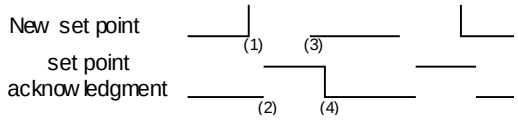
Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

The "Position target" object can be laid on a R-PDO. A handshake can be used to assign a new set point value (see also with R-PDO control; bit 15; page 21). This is done using the following bits:

- Control word byte 2 bit 4 "new set point" and
- Status word byte 1 bit 4 "acknowledge record point"

Function:



Transition	Meaning	Conditions
1	New target value	Target val.- acknowledgment ="0" actual value can be transferred
2	Set point	Set point acknowledgment ="1" Set point recognized
3	acknowledgment	New set point ="0"
4	New set point	Target val.- acknowledgment ="0" New target value can be transferred
	Set point acknowledgment	

The automatic assignment of the position target of the RPDO can be switched off with bit 15 from the R-PDO control object (see page 21).

6.12.2 Position actual value

Position actual value.
Current drive position.

Object Description

Index	0x6064				
Symbol	LAGE_IST	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	read	**PDO mapping**	yes

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.12.3 Status S2

Target position.
End position of the current or last positioning cycle implemented.

Object Description

Index	0x2020				
Symbol	S2	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	Read all	**PDO mapping**	no

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.12.4 Status S12

Position of the absolute value sensor (Option A1).

Object Description

Index	0x2022				
Symbol	S12	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	read	**PDO mapping**	no

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.12.5 Real zero

Reference measurement offset.
Difference between real zero point and machine zero point.
After the reference run, all positioning processes refer to the real null point.

Object Description

Index	0x607c				
Symbol	REALNULL	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.12.6 Limits

Position limit value min-max.
The position limit values are software end limits and correspond to the absolute position limits, within which the set values and actual values (in absolute form) must be moved. Each new target position is checked with these limits. They always refer to the machine zero point; therefore they must be corrected using the reference measurement offset.

Object Description

Index	0x607d				
Symbol	GRENZEN	**Length**	4	**Min**	0x80000000
Object code	Array	**Elements**	2	**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

Sub-index	Assignment	Sub-index	Assignment
1	Negative limit stop	2	Positive limit stop

Observe the number range of the integer 32 format (see page 7 under "Integer ").
Limits outside -2 147 483 - +2 147 483 units cannot be depicted.

6.12.7 Positioning window

The positioning window is situated symetrically around the position target.
Once the position actual value lies within this window, the Bit "Position reached" is set in the status word.

Object Description

Index	6067				
Symbol	POS_FENSTER	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.12.8 Lag error window

The lag error window is situated symetrically around the actually specified position set point value.
If the current position indicator actual value lies outside this window, a contour error occurs.

Object Description

Index	0x6065				
Symbol	SCHLEPP_FEN	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0xFFFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.12.9 Status S3

Contour error.
Difference between set and actual position in a positioning cycle.

Object Description

Index	0x2021				
Symbol	S3	**Length**	2	**Min**	0x8000
Object code	Variable			**Max**	0x7FFF
Data type	Integer16	**Access rights**	read	**PDO mapping**	yes

Data Description

Unit	mm (or inch)	**Resolution**	$1 \Leftrightarrow {}^{1}/_{256}$ mm (or inch)

6.12.10 Polarity

The set point and actual values are multiplied by 1 or -1 depending on polarity. This allows the user to reverse the direction of orientation

Object Description

Index	0x607e				
Symbol	POLARITÄT	**Length**	1	**Min**	0x00
Object code	Variable			**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

Data Description

Bit 7	Assignment	Bit 7	Assignment
= 1	Reversal of direction (motor left)	= 0	default (motor right)

The other bits are irrelevant for COMPAX.

6.12.11 Reference movement selection

Selection function which describes the method for the reference movement.

Object Description

Index	0x6098				
Symbol	REF_MODE	**Length**	1	**Min**	0x80
Object code	Simple var.			**Max**	0x7F
Data type	Integer8	**Access rights**	read/write	**PDO mapping**	no

Data Description

Selection code	Method	Selection code	Method
8	MN-Ini. approach in direction 1	12	MN-Ini. approach in direction 2

The COMPAX parameter P213 is influenced by this object.
The direction in which the machine zero is approached is also influenced by the COMPAX parameters P212 (machine zero mode) and P3 (speed of machine zero travel).

6.12.12 Traverse rate and set point position default

Traverse speed in % of the nominal speed (nominal rpm * travel per motor revolution).
and position set point default
Absolute positioning. Reference point is the real null (RN) or relative positioning:
adjustable via the control word bit 6 (0 = absolute; 1 = relative).

Object Description

Index	0x2023				
Symbol	POSRXSPEEDY	**Length**	6	**Min**	
Object code	Variable			**Max**	
Data type	Octet String	**Access rights**	write	**PDO mapping**	yes

Data Description

Data byte	Assignment	Data byte	Assignment
1	Highest value byte of the speed	2	Lowest value byte of the speed

Coding	binary	**Unit**	%
Value range	1 ... 38400	**Resolution**	$1 \Leftrightarrow {}^{1}/_{64}\%$; (6400 ⇔ 100%)

Data byte	Assignment	Data byte	Assignment
3	Highest value byte of the position	6	Lowest value byte of the position

Coding	Complement to two	**Unit**	mm (or inch)
Value range	-2 147 483 648 ... +2 147 483 647	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.13 Speed

6.13.1 Traverse speed

Traverse speed in % of the nominal speed (nominal rpm * travel per motor revolution).
The value is valid until a new value is programmed.
The set speed can be reduced by using the OVERRIDE object.
A speed change during the positioning cycle is possible by using the POSR0SPEED object.

Object Description

Index	0x2030				
Symbol	SPEED	**Length**	2	**Min**	0x8000
Object code	Variable			**Max**	0x7FFF
Data type	Integer16	**Access rights**	read/write	**PDO mapping**	yes

Data Description

Unit	%	**Resolution**	$1 \Leftrightarrow {}^{1}/_{64}\%$; (6400 ⇔ 100%)

6.13.2 Override

Reduce traverse speed.
Software emulation of an external potentiometer on the override input (X11.6).

Object Description

Index	0x2031				
Symbol	OVERRIDE	**Length**	1	**Min**	0x00
Object code	Variable			**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	yes

Data Description

Unit	%	**Resolution**	$1 \Leftrightarrow {}^{1}/_{255}\%$; (255 ⇔ 100%)

6.13.3 POSR 0 SPEED

Changing traverse speed during a positioning cycle.

Object Description

Index	0x2032				
Symbol	POSR0SPEED	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0x000927C0
Data type	Unsigned32	**Access rights**	write	**PDO mapping**	no

Data Description

Unit	%	**Resolution**	1 ⇔ 0.001 %

6.13.4 Speed step profiling

Each speed step profiling can have a maximum of 8 speed steps. The position value is given as a relative measurement. It is referenced to the positioning start point.

Object Description

Index	0x2033				
Symbol	POSRXSPEEDY	**Length**	8	**Min**	
Object code	Variable			**Max**	
Data type	Octet String	**Access rights**	write	**PDO mapping**	no

Data Description

Data byte	Assignment	Data byte	Assignment
1	Highest value byte of the position	4	Lowest value byte of the position

Coding	Complement to two	**Unit**	mm (or inch)
Value range	-2 147 483 648 ... +2 147 483 647	**Resolution**	1 ⇔ 0.001 mm (or inch)

Data byte	Assignment	Data byte	Assignment
5	Highest value byte of the speed	8	Lowest value byte of the speed

Coding	binary	**Unit**	%
Value range	1 ... 600 000	**Resolution**	1 ⇔ 0.001 %

6.13.5 Speed step profiling with acceleration

Each speed step profiling can have a maximum of 8 speed steps. The position value is given as a relative measurement. It is referenced to the positioning start point.

Object Description

Index	0x2034				
Symbol	PRXSDYALZ	**Length**	10	**Min**	
Object code	Variable			**Max**	
Data type	Octet String	**Access rights**	write	**PDO mapping**	no

Data Description

Data byte	Assignment	Data byte	Assignment
1	Highest value byte of the position	4	Lowest value byte of the position

Coding	Complement to two	**Unit**	mm (or inch)
Value range	-2 147 483 648 ... +2 147 483 647	**Resolution**	1 ⇔ 0.001 mm (or Inch)

Data byte	Assignment	Data byte	Assignment
5	Highest value byte of the speed	8	Lowest value byte of the speed

Coding	binary	**Unit**	%
Value range	1 ... 600 000	**Resolution**	1 ⇔ 0.001 %

Data byte	Assignment	Data byte	Assignment
9	MSB of the ramp time	10	LSB of the ramp time

Coding	binary	**Unit**	ms
Value range	0 ... 65 000	**Resolution**	1 ⇔ 1 ms

6.13.6 Maximum motor speed

The maximum speed is specified for both directions of rotation together with a resolution of 1 rpm.
These are for the protection of the motor and can be found in the motor data sheet.

Object Description

Index	0x6080				
Symbol	DREHZAHLMAX	**Length**	2	**Min**	0x0000
Object code	Variable			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	Rpm	**Resolution**	1 ⇔ 1 rpm

6.13.7 Maximum speed

Maximum speed amount
The maximum speed applies to both directions of rotation.

Object Description

Index	0x607F				
Symbol	GESCHW_MAX	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	%	**Resolution**	1 ⇔ 0.001 %

6.13.8 Traverse speed.

Traverse speed.
Given in % of the nominal speed (nominal rpm * travel per motor revolution).
The value is valid until a new value is programmed.
The set speed can be reduced by using the OVERRIDE object.
A speed change during the positioning cycle is possible by using the POSR0SPEED object.

Object Description

Index	0x6081				
Symbol	VERF_GESCHW	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	%	**Resolution**	1 ⇔ 0.001 %

6.13.9 Status S4

Current axis process speed.
Value in % of the nominal speed (nominal rpm * travel per motor revolution).

Object Description

Index	0x2035				
Symbol	S4	**Length**	2	**Min**	0x8000
Object code	Variable			**Max**	0x7FFF
Data type	Integer16	**Access rights**	read	**PDO mapping**	yes

Data Description

Unit	%	**Resolution**	1 ⇔ $^{1}/_{64}$%; (6400 ⇔ 100%)

6.13.10 Traverse speed actual value

Traverse speed actual value
Value in % of the nominal speed (nominal rpm * travel per motor revolution).

Object Description

Index	0x606C				
Symbol	GESCHW_IST	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	read	**PDO mapping**	no

Data Description

Unit	%	**Resolution**	1 ⇔ 0.001 %

6.13.11 Reference run speed

Speed set point value for approaching the machine zero point.
Given in % of the nominal speed (nominal rpm * travel per motor revolution).

Object Description

Index	0x6099				
Symbol	GESCHW_REF	**Length**	4	**Min**	0x80000000
Object code	Array		2	**Max**	0x7FFFFFFF
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Unit	%	**Resolution**	1 ⇔ 0.001 %

Sub-index	Assignment	Sub-index	Assignment
1	Enter speed during initiator search	2	Enter speed during zero impulse search

⇨ With COMPAX both values are always the same!

6.14 Acceleration

1.1.1 Ramp form

Selection function which describes acceleration process.

Object Description

Index	0x6086				
Symbol	RAMPENFORM	**Length**	2	**Min**	0x8000
Object code	Variable			**Max**	0x7FFF
Data type	Integer16	**Access rights**	read/write	**PDO mapping**	no

Data Description

Selection code	Ramp shape	Selection code	Ramp shape
-1	square	2	without jerk
0	linear		

6.14.2 Acceleration

Time specification for the acceleration process.
Also the time setting for the deceleration process as long as the object ACCEL-NEG or RAMPE-NEG has not been written to.

The time specification applies to nominal speed (100%). $t_a = \frac{SPEED}{100\%} \bullet RAMPE\_POS$

Object Description

Index	0x6083				
Symbol	RAMPE_POS	**Length**	4	**Min**	0x00000000
Object code	Simple var.			**Max**	0x0000FDE8
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Value range	0 ... 65 000		
Unit	ms	**Resolution**	1 ⇔ 1 ms

6.14.3 Time specification for the braking process

The time specification applies to nominal speed (100%).

$$t_a = \frac{SPEED}{100\%} \bullet RAMPE\_NEG$$

Object Description

Index	0x6084				
Symbol	RAMPE_NEG	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0x0000FDE8
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Value range	0 ... 65 000		
Unit	ms	**Resolution**	1 ⇔ 1 ms

6.14.4 Quick stop

Time specification for the braking process if the quick stop command is given (bit 2 in the control word), a limit switch is activated or after an emergency stop.

The time specification applies to nominal speed (100%).

$$t_a = \frac{SPEED}{100\%} \bullet RAMPE\_NOTS$$

Object Description

Index	0x6085				
Symbol	RAMPE_NOTS	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0x0000FDE8
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Value range	0 ... 65 000		
Unit	ms	**Resolution**	1 ⇔ 1 ms

6.14.5 Reference movement acceleration

Acceleration time for approaching the machine zero point.

The time specification applies to nominal speed (100%).

$$t_a = \frac{SPEED}{100\%} \bullet RAMPE\_REF$$

Object Description

Index	0x609A				
Symbol	RAMPE_REF	**Length**	4	**Min**	0x00000000
Object code	Variable			**Max**	0x0000FDE8
Data type	Unsigned32	**Access rights**	read/write	**PDO mapping**	no

Data Description

Value range	0 ... 65 000		
Unit	ms	**Resolution**	1 ⇔ 1 ms

6.15 Inputs/outputs

1.1.1 INPUT_WORD

Logic state of the 16 digital inputs.
Some inputs are assigned fixed control functions.

Input	Assignment	Input	Assignment
1	SHIFT	1 & 3	Find real null (RN)
2	Hand+	1 & 4	Teach real zero
3	Hand-	1 & 5	reserved
4	Quit	1 & 6	Break
5	START	9...13	Freely assignable in standard model
6	STOP	14	Activate label reference
7...8	Freely assignable in standard model	15	Faster start
1 & 2	Find machine zero (MN)	16	Label input

Object Description

Index	0x6100				
Symbol	INPUT_WORD	**Length**	-	**Min**	-
Object code	Array	**Elements**	1	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x6100:000				
Standard	0x01	**Length**	1	**Min**	0x01
Object code	-	**Elements**	-	**Max**	0x01
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	no

INPUT_WORD

Index	0x6100:001				
Symbol	INPUT_WORD	**Length**	2	**Min**	0x0000
Object code	-	**Elements**	-	**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read	**PDO mapping**	yes

Data Description

Bit	Assignment	Bit	Assignment
15	Status input 16	0	Status input 1

6.15.2 INPUT_MASK

Masking inputs.
Standard configuration is for functions assigned to the COMPAX inputs.
If the corresponding bit of the mask is set, access is given to this COMPAX input function via the control word or COMPAX control word, while at the same time the corresponding COMPAX input loses this function and is freely available.
INPUT_MASK directly reads/describes the parameters P221 (LSB) and P222 (MSB).

Object Description

Index	0x2040				
Symbol	INPUT_MASK	**Length**	2	**Min**	0x0000
Object code	Variable			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read all/write	**PDO mapping**	no

Data Description

Bit	Assignment	Parameter
15 ... 8	no function (="0")	P222/Bit 7 ... 0
7...6	no function (="0")	P221/Bit 7 ... 6
5...0	Mask input 6...1	P221/Bit 5 ... 0

6.15.3 OUTPUT_WORD

Logic state of the 16 digital outputs.
Some outputs are assigned a fixed status information.

Output	Assignment	Output	Assignment
1	No fault	5	Programmed nominal position reached
2	No warning	6	Idle after stop
3	Machine zero has been approached	7...15	Freely assignable in standard model
4	Ready for start	16	Label present after max. feed length

This object allows the outputs to be set or reset to default.
Each output which is to be influenced via this object must be explicitly released for this purpose by means of P223,P224. The output thereby loses any status information which was assigned to it.

Object Description

Index	0x6300				
Symbol	OUTPUT_WORD	**Length**	-	**Min**	-
Object code	Array	**Elements**	1	**Max**	-
Data type	-	**Access rights**	-	**PDO mapping**	no

Number of entries

Index: subindex	0x6300:000				
Standard	0x01	**Length**	1	**Min**	0x01
Object code	-	**Elements**	-	**Max**	0x01
Data type	Unsigned8	**Access rights**	read	**PDO mapping**	no

OUTPUT_WORD

Index	0x6300:001				
Symbol	OUTPUT_WORD	**Length**	2	**Min**	0x0000
Object code	-			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	yes

Data Description

Bit	Assignment	Bit	Assignment
15	Status output 16	0	Status output 1

6.15.4 OUTPUT_MASK

Mask outputs.
Each output which is to be influenced via the object OUTPUT_WORD must be explicitly released (masked) for this purpose.
The output thereby loses any status information which was assigned to it.
OUTPUT-MASK directly reads/describes the parameters P223 (LSB) and P224 (MSB).

Object Description

Index	0x2041				
Symbol	OUTPUT_MASK	**Length**	2	**Min**	0x0000
Object code	Variable			**Max**	0xFFFF
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	no

Data Description

Bit	Assignment	Parameter
15 ... 8	Mask output 16...9	P224/Bit 7 ... 0
7...0	Mask output 8...1	P223/Bit 7 ... 0

6.15.5 OUTPUT

Set or reset a digital output.
The corresponding output is selected using the Sub-index (Sub-index = output no.).
Some outputs have a fixed status information assigned (see OUTPUT-WORD).

Object Description

Index	0x2042				
Symbol	OUTPUT	**Length**	1	**Min**	0x00
Object code	Array	**Elements**	16	**Max**	0xFF
Data type	Boolean	**Access rights**	write	**PDO mapping**	no

Data Description

Data byte	Function	Data byte	Function
= 0xff (TRUE)	Output [Sub-index] = 1	= 0x00 (FALSE)	Output [Sub-index] = 0

6.15.6 Comparator function (active high)

Setting a not assigned output within the positioning process (active high).
The position value is given as a relative measurement. It is referenced to the positioning start point.
A maximum of 8 comparators can be set for a positioning process.
The corresponding output is selected using the Sub-index (Sub-index = output no.).

Object Description

Index	0x2043				
Symbol	POSROUTPUTP	**Length**	4	**Min**	0x80000000
Object code	Array	**Elements**	16	**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	write	**PDO mapping**	no

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.15.7 Comparator function (active low)

Resetting a not assigned output within the positioning process.
The position value is given as a relative measurement. It is referenced to the positioning start point.
A maximum of 4 comparators can be set for a positioning process.
The corresponding output is selected using the Sub-index (Sub-index = output no.).

Object Description

Index	0x2044				
Symbol	POSROUTPUTN	**Length**	4	**Min**	0x80000000
Object code	Array	**Elements**	16	**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	write	**PDO mapping**	yes

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.16 Programming

6.16.1 Reading and writing the program memory.

Reading and writing the program memory with command records in binary format.
The record number is determined with the Sub-index (Sub-index = record No.)

Object Description

Index	0x2050				
Symbol	Nx	**Length**	20	**Min**	
Object code	Array	**Elements**	250	**Max**	
Data type	Octet String	**Access rights**	read/write	**PDO mapping**	no

Data Description

Data byte	Assignment
1	Record length (number of relevant record data)
2	1st character of the record content
...	
20	last character of the record contents

COMPAX Command Codes

Definition of the command code: (A command code consists of 1 byte).

Sorted according to command code

Code	Command
0x20	Empty instruction (No Operation)
0x01	**VALIDP / C / F**
0x41	**POSA** Value / **POSA HOME**
0x42	**GOSUB**
0x43	**SETC** n
0x45	**END**
0x47	**GOTO**
0x49	**IF E**x=y ... / **IF ERROR ... / IF STOP ...**
0x4A	**IF <**Operand1**> <**Comparison Operator**> <**Operand2**> ...**
0x4B	**LOOP** n
0x4C	**ACCEL** Value
0x4D	**SETM** Value
0x4F	**OUTPUT A**x=y
0x50	**P**n**=. . .**
0x51	**SETS** Value
0x52	**POSR** Value / **POSR CAM**
0x53	**SPEED** Value / **SPEED SYNC**
0x54	**REPEAT** Value
0x55	**RETURN**
0x56	**V**n**=. . .**
0x57	**WAIT** Value / **WAIT START**
0x61	**POSA** Parameter
0x6B	**LOOP** Parameter
0x6C	**ACCEL** Parameter
0x6D	**SETM** Parameter
0x71	**SETS** Parameter
0x72	**POSR** Parameter
0x73	**SPEED** Parameter
0x74	**REPEAT** Parameter
0x77	**WAIT** Parameter
0xC1	**POSA** Variable
0xCB	**LOOP** Variable
0xCC	**ACCEL** Variable
0xCD	**SETM** Variable
0xD1	**SETS** Variable
0xD2	**POSR** Variable
0xD3	**SPEED** Variable
0x4D	**REPEAT** Variable
0xD7	**WAIT** Variable

Sorted by command

Code	Command
0x6C	**ACCEL** Parameter
0xCC	**ACCEL** Variable
0x4C	**ACCEL** Value
0x45	**END**
0x42	**GOSUB**
0x47	**GOTO**
0x4A	**IF <**Operand1**> <**Comparison Operator**> <**Operand2**> ...**
0x49	**IF E**x=y ... / **IF ERROR ... / IF STOP ...**
0x20	Empty instruction (No Operation)
0x4B	**LOOP** n
0x6B	**LOOP** Parameter
0xCB	**LOOP** Variable
0x4F	**OUTPUT A**x=y
0x50	**P**n**=. . .**
0x61	**POSA** Parameter
0xC1	**POSA** Variable
0x41	**POSA** Value / **POSA HOME**
0x72	**POSR** Parameter
0xD2	**POSR** Variable
0x52	**POSR** Value / **POSR CAM**
0x74	**REPEAT** Parameter
0x4D	**REPEAT** Variable
0x54	**REPEAT** Value
0x55	**RETURN**
0x43	**SETC** n
0x6D	**SETM** Parameter
0xCD	**SETM** Variable
0x4D	**SETM** Value
0x71	**SETS** Parameter
0xD1	**SETS** Variable
0x51	**SETS** Value
0x73	**SPEED** Parameter
0xD3	**SPEED** Variable
0x53	**SPEED** Value / **SPEED SYNC**
0x01	**VALIDP / C / F**
0x56	**V**n**=. . .**
0x77	**WAIT** Parameter
0xD7	**WAIT** Variable
0x57	**WAIT** Value / **WAIT START**

Reading and writing the program memory.

Definition of Operands

An operand consists of 7 bytes; 1 byte for the type indicator and 6 data bytes.

Operand	Type	D1	D2	D3	D4	D5	D6
Parameter	0x50	No.H	No.L	0x00	0x00	0x00	0x00
Status	0x53	No.H	No.L	0x00	0x00	0x00	0x00
Variable	0x56	No.H	No.L	0x00	0x00	0x00	0x00
Constants	0x20	NL	NM	NH	VL	VM	VH

Definition of Comparison Operators

A comparison operator consists of 1 byte.

Comparison operator	Symbols	Code
Equal	=	0x3D
Less than	<	0x3C
Greater than	>	0x3E
Equal to/less than	<=	0xBC
Equal to/greater than	>=	0xBE
Does not equal	<>	0xBB

Definition of Arithmetical Operators

An arithmetic operator consists of 1 byte.

Arithmetic Operator	Symbols	Code
Addition	+	0xB2
Subtraction	-	0x2D
Multiplication	*	0x2A
Division	/	0x2F
Whole number division	\	0x5C
Modulo calculation	%	0x25

Use of this coding results in the following record memory - command code table. All of the commands are listed individually here!

COMPAX natural language memory command code table

Command	Code							
ACCEL Parameter	0x6C	No.H	Nr.L					
ACCEL Variable	0xCC	No.H	Nr.L					
ACCEL Value	0x4C	MSB	LSB					
END	0x45	0x00						
GOSUB EXT	0x42	0x00	0x00					
GOSUB Value	0x42	MSB	LSB					
GOTO EXT	0x47	0x00	0x00					
GOTO Value	0x47	MSB	LSB					
IF ERROR GOSUB n	0x49	0x00	0xFF	0x31	0x42	n MSB	n LSB	
IF ERROR GOTO n	0x49	0x00	0xFF	0x31	0x47	n MSB	n LSB	
IF Ex=y **GOSUB** n	0x49	x MSB	x LSB	y	0x42	n MSB	n LSB	
IF Ex=yy **GOSUB** n	0x49	x MSB	x LSB	y1	y2	0x42	n MSB	n LSB
IF Ex= . . .	...	...	...	...	...	...	...	...
IF Ex=yyyyyyyy **GOSUB** n	0x49	x MSB	x LSB	y1	y2	y3	y4	y5
	y6	y7	y8	0x42	n MSB	n LSB		
IF Ex=y **GOTO** n	0x49	x MSB	x LSB	y	0x47	n MSB	n LSB	
IF Ex=yy **GOTO** n	0x49	x MSB	x LSB	y1	y2	0x47	n MSB	n LSB
IF Ex= . . .	...	...	...	...	...	...	...	...
IF Ex=yyyyyyyy **GOTO** n	0x49	x MSB	x LSB	y1	y2	y3	y4	y5
	y6	y7	y8	0x47	n MSB	n LSB		
IF <Operand1**> <**Comparison operator**> <**Operand2**> GOTO** n	0x4A	O1Type	O1D1	O1D2	O1D3	O1D4	O1D5	O1D6
	Vglop	O2Type	O2D1	O2D2	O2D3	O2D4	O2D5	O2D6
	0x47	n MSB	n LSB					
IF <Operand1**> <** Comparison operator **> <**Operand2**> GOSUB** n	0x4A	O1Type	O1D1	O1D2	O1D3	O1D4	O1D5	O1D6
	Vglop	O2Type	O2D1	O2D2	O2D3	O2D4	O2D5	O2D6
	0x42	n MSB	n LSB					
IF STOP GOSUB n	0x49	0x00	0xFE	0x31	0x42	n MSB	n LSB	
IF STOP GOTO n	0x49	0x00	0xFE	0x31	0x47	n MSB	n LSB	
LOOP n	0x4B	n MSB	n LSB					
LOOP Parameter	0x6B	No.H	Nr.L					
LOOP Variable	0xCB	No.H	Nr.L					
OUTPUT Ax=y	0x4F	x MSB	x LSB	y				
OUTPUT Ax=yy	0x4F	x MSB	x LSB	y1	y2			
OUTPUT Ax=. . .	...	...	...	...	...	...		
OUTPUT Ax=yyyyyyyy	0x4F	x MSB	x LSB	y1	y2	y3	y4	y5
	y6	y7	y8					
OUTPUT A0=y	0x4F	0x00	0x00	y				
POSA HOME	0x41	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	
POSA Parameter	0x61	No.H	Nr.L	0x00	0x00	0x00	0x00	
POSA Variable	0x1C	No.H	Nr.L	0x00	0x00	0x00	0x00	
POSA Value t	0x41	NL	NM	NH	VL	VM	VH	
POSR CAM	0x52	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	
POSR Parameter	0x72	No.H	Nr.L	0x00	0x00	0x00	0x00	
POSR Variable	0x2D	No.H	Nr.L	0x00	0x00	0x00	0x00	
POSR Value	0x52	NL	NM	NH	VL	VM	VH	
REPEAT Parameter	0x74	No.H	Nr.L					
REPEAT Variable	0x4D	No.H	Nr.L					
REPEAT Value t	0x54	MSB	LSB					
RETURN	0x55	0x00						
SETC n	0x43	n MSB	n LSB					

Reading and writing the program memory.

Command	Code							
SETM Value	0x4D	NL	NM	NH	VL	VM	VH	
SETM Parameter	0x6D	No.H	Nr.L	0x00	0x00	0x00	0x00	
SETM Variable	0xCD	No.H	Nr.L	0x00	0x00	0x00	0x00	
SETS Value	0x51	NL	NM	NH	VL	VM	VH	
SETS Parameter	0x71	No.H	Nr.L	0x00	0x00	0x00	0x00	
SETS Variable	0x1D	No.H	Nr.L	0x00	0x00	0x00	0x00	
SPEED Parameter	0x73	No.H	Nr.L	0x00	0x00	0x00	0x00	
SPEED Variable	0x3D	No.H	Nr.L	0x00	0x00	0x00	0x00	
SPEED Value	0x53	NL	NM	NH	VL	VM	VH	
SPEED SYNC	0x53	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	
VALIDP	0x01	0x56	0x50					
VALIDC	0x01	0x56	0x43					
VALIDF	0x01	0x56	0x46					
WAIT Parameter	0x77	No.H	Nr.L					
WAIT Variable	0x7D	No.H	Nr.L					
WAIT Value	0x57	MSB	LSB					
WAIT START	0x57	0x00	0x00					
POSA Value **WAIT** Value	0x41	NL	NM	NH	VL	VM	VH	0x57
	MSB	LSB						
POSA . . . **WAIT** . . .	...	...	...	...	...	...	...	
(Combinations examples.: ... Value ... .V12, ...P40 ... .V10)	...	...	...	...	...	...	...	
POSA Variable **WAIT** Variable	0x1C	No.H	Nr.L	0x00	0x00	0x00	0x00	0x7D
	No.H	Nr.L						
POSR Value **OUTPUT A**x=y	0x52	NL	NM	NH	VL	VM	VH	0x4F
	x MSB	x LSB	y					
POSR Parameter **OUTPUT A**x=y	0x72	No.H	Nr.L	0x00	0x00	0x00	0x00	0x4F
	x MSB	x LSB	y					
POSR Variable **OUTPUT A**x=y	0x2D	No.H	Nr.L	0x00	0x00	0x00	0x00	0x4F
	x MSB	x LSB	y					
POSR Value t **SPEED** Value	0x52	NL	NM	NH	VL	VM	VH	0x53
	NL	NM	NH	VL	VM	VH		
POSR . . . **SPEED** . . .	...	...	...	...	...	...	...	
(Combinations examples.: ... Value ... .V12, ...P40 ... .V10)	...	...	...	...	...	...		
POSR Variable **SPEED** Variable	0x2D	No.H	Nr.L	0x00	0x00	0x00	0x00	0x3D
	No.H	Nr.L	0x00	0x00	0x00	0x00		
POSR Value **SPEED** Value **ACCEL** Value	0x52	NL	NM	NH	VL	VM	VH	0x53
	NL	NM	NH	VL	VM	VH	0x4C	MSB
	LSB							
POSR ... **SPEED** ... **ACCEL** ...	...	...	...	...	...	...	...	
(Combinations examples.: ... Value ... .V12 ... .V13, ...P40 ... .V10 ... .P41)	...	...	...	...	...	...	...	
	...							
POSR Variable **SPEED** Variable **ACCEL** Variable	0x2D	No.H	Nr.L	0x00	0x00	0x00	0x00	0x3D
	No.H	Nr.L	0x00	0x00	0x00	0x00	0xCC	No.H
	Nr.L							
POSR Value **WAIT** Value	0x52	NL	NM	NH	VL	VM	VH	0x57
	MSB	LSB						
POSR . . . **WAIT** . . .	...	...	...	...	...	...	...	...
(Combinations examples.: ... Value ... .V12, ...P40 ... .V10)	...	...						
POSR Variable **WAIT** Variable	0x2D	No.H	Nr.L	0x00	0x00	0x00	0x00	0x7D
	No.H	Nr.L						

Command	Code							
SPEED Value **WAIT** Value	0x53	NL	NM	NH	VL	VM	VH	0x57
	MSB	LSB						
SPEED . . . **WAIT** . . .	...	...	...	...	...	...	...	...
(Combinations examples.: ... Value ... .V12, ...P40 ... .V10)	...	...						
SPEED Variable **WAIT** Variable	0x3D	No.H	Nr.L	0x00	0x00	0x00	0x00	0x7D
	No.H	Nr.L						
WAIT POSA Value	0x57	0x00	0x00	0x41	NL	NM	NH	VL
	VM	VH						
WAIT POSR Value	0x57	0x00	0x00	0x52	NL	NM	NH	VL
	VM	VH						
Pn**=<**Operand1**> [<**Arithmetic Operator**> <**Operand2**>]**	0x50	n MSB	n LSB	O1Type	O1D1	O1D2	O1D3	O1D4
	O1D5	O1D6	AriOp	O2Type	O2D1	O2D2	O2D3	O2D4
	O2D5	O2D6						
Vn**=<**Operand1**> [<**Arithmetic Operator**> <**Operand2**>]**	0x56	n MSB	n LSB	O1Type	O1D1	O1D2	O1D3	O1D4
	O1D5	O1D6	AriOp	O2Type	O2D1	O2D2	O2D3	O2D4
	O2D5	O2D6						

Key:

No.H:	High byte of the parameter/variable number
Nr.L:	Low byte of the parameter/variable number
MSB:	High byte of an integer value
LSB:	Low byte of an integer value
NL:	Low byte of the fractional digit of a value in DSP number format
NM:	Mid byte of the fractional digit of a value in DSP number format
NH:	High byte of the fractional digit of a value in DSP number format
VL:	Low Byte of the integral digit of a value in DSP number format
VM:	Mid Byte of the integral digit of a value in DSP number format
VH:	High byte of the integral digit of a value in DSP number format
O1Type:	Type indicator of the 1. operand
O1D1...O1D6:	Data for the 1st operand
O2Type:	Type indicator of the 2. operand
O2D1...O2D6:	Data for the 2nd operand
Vglop:	Comparison operator
AriOp:	Arithmetic Operator
y (y1, y2, ...)	y=0x30 for "1" and y=0x31 for "0"

6.16.2 Goto

Set and read record pointer.

Object Description

Index	0x2051				
Symbol	GOTO	**Length**	1	**Min**	0x01
Object code	Variable			**Max**	0xFF
Data type	Unsigned8	**Access rights**	read/write	**PDO mapping**	no

Data Description

Data format	binary	**Unit**	Record number
Value range	1 ... 250	**Resolution**	1

6.16.3 Execute program data record N

Only this record is processed. The record pointer remains at this program record.

Object Description

Index	0x2052				
Symbol	START_N	**Length**	1	**Min**	0x01
Object code	Variable			**Max**	0xFF
Data type	Unsigned8	**Access rights**	write	**PDO mapping**	yes

Data Description

Data format	binary	**Unit**	Record number
Value range	1 ... 250	**Resolution**	1

6.16.4 Program start from record N.

The data record indicator is set on the corresponding program data record and then the program is started.

Object Description

Index	0x2053				
Symbol	START_N_GO	**Length**	1	**Min**	0x01
Object code	Variable			**Max**	0xFF
Data type	Unsigned8	**Access rights**	write	**PDO mapping**	yes

Data Description

Data format	binary	**Unit**	Record number
Value range	1 ... 250	**Resolution**	1

6.16.5 Assign current position in record N

The command "POSA *current position*" is saved in data record N.

Object Description

Index	0x2054				
Symbol	TEACH_N	**Length**	1	**Min**	0x01
Object code	Variable			**Max**	0xFF
Data type	Unsigned8	**Access rights**	write	**PDO mapping**	no

Data Description

Data format	binary	**Unit**	Record number
Value range	1 ... 250	**Resolution**	1

6.16.6 Enter or read COMPAX variables

Enter or read COMPAX variables.
The corresponding variable is selected using the Sub-index (Sub-index = variable No.).
Sub-index = 40 addresses variable 0 of the COMPAX, with which all variables can be set to the same value.

Object Description

Index	0x2057				
Symbol	VX	**Length**	4	**Min**	0x80000000
Object code	Array	**Elements**	40	**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	read/write	**PDO mapping**	yes

6.17 COMPAX XX50 commands

6.17.1 WAITPOSA

Synchronization with automatic reverse travel (clocked command; COMPAX XX50).
Starting from the rest position of the drive, a complete synchronization move is carried out.
The value for this object is the processing interval (length of material when cutting).

Object Description

Index	0x2025				
Symbol	WAITPOSA	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	write all	**PDO mapping**	no

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.17.2 WAITPOSR

Synchronization without automatic reverse travel (clocked command; COMPAX XX50).
Starting from the rest position of the drive, a complete synchronization move is carried out.
The value for this object is the processing interval (length of material when cutting).

Object Description

Index	0x2026				
Symbol	WAITPOSR	**Length**	4	**Min**	0x80000000
Object code	Variable			**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	write all	**PDO mapping**	no

Data Description

Unit	mm (or inch)	**Resolution**	1 ⇔ 0.001 mm (or inch)

6.18 COMPAX XX70 commands

6.18.1 Cam commands

The corresponding command is selected using the sub-index (sub-index = CAM command).

Object Description

Index	0x2024				
Symbol	CAM_CMD	**Length**	4	**Min**	0x80000000
Object code	Array	**Elements**	6	**Max**	0x7FFFFFFF
Data type	Integer32	**Access rights**	write all	**PDO mapping**	no

Data Description

Sub-index	Command	Resolution
1	SETC	1
2	SETM	1 ⇔ 0.001
3	SETS	1 ⇔ 0.001
4	POSR CAM	-
5	LOOP	1
6	VF	-

6.18.2 Setting and reading cam memory indicator

Object Description

Index	0x2055				
Symbol	CAM_MEM_P	**Length**	2	**Min**	0x0001
Object code	Variable			**Max**	0x1554
Data type	Unsigned16	**Access rights**	read/write	**PDO mapping**	no

Data Description

Data format	binary	**Unit**	Curve memory number
Value range	1 ... 5460	**Resolution**	1

6.18.3 Read and write cam memory

The cam memory number is defined by the actual value of the cam memory indicator (CAM_MEM_P).
The curve memory pointer is automatically incremented after this object is accessed.

Object Description

Index	0x2056				
Symbol	CAM_MEM	**Length**	3	**Min**	
Object code	Variable			**Max**	
Data type	Octet String	**Access rights**	read/write	**PDO mapping**	no

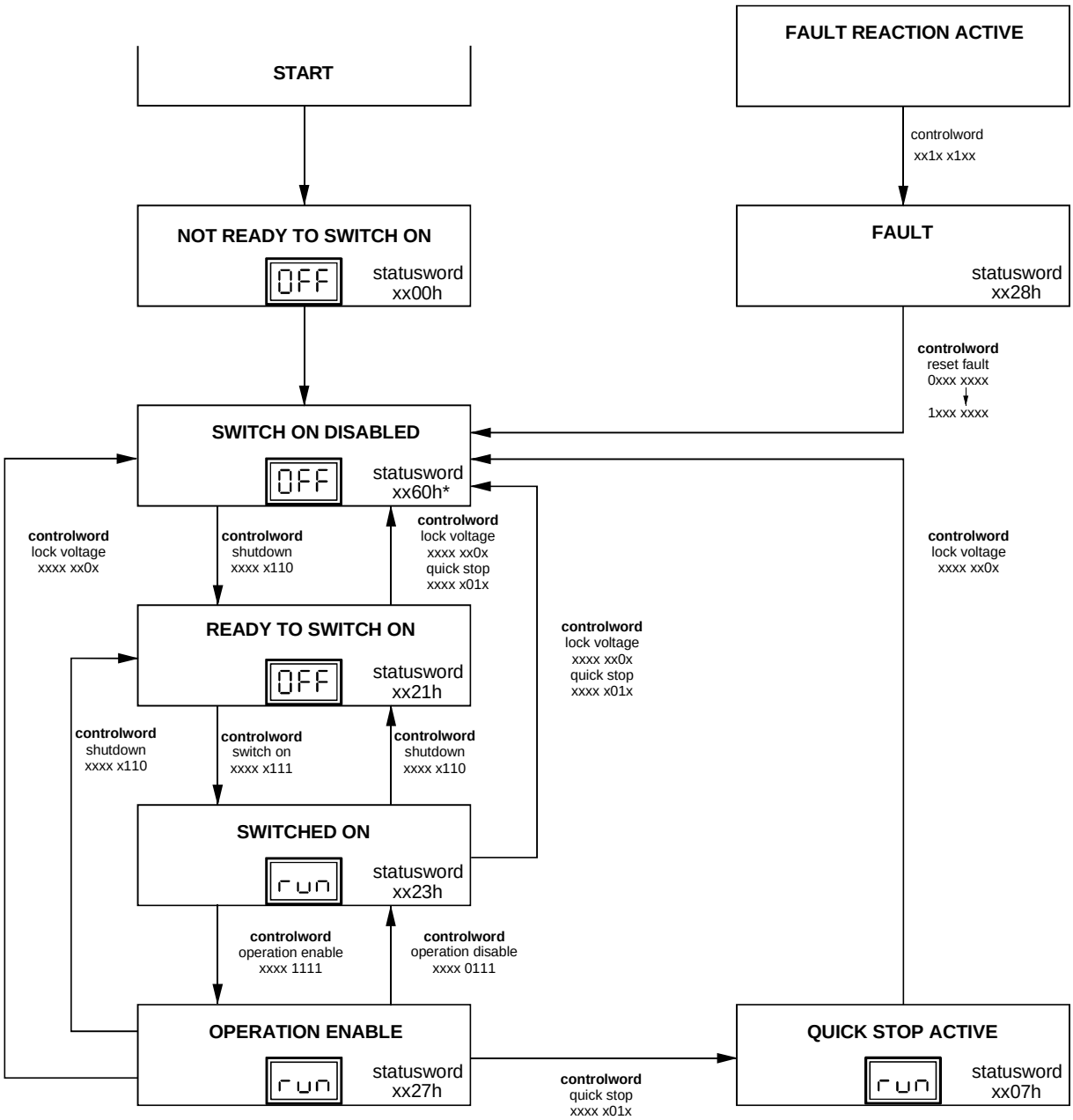
Data Description

Data byte	1	2	3
Meaning	Record memory contents		
Assignment	MSB	...	LSB

7 Device Profile DSP 402

From COMPAX program version V3.01 onwards, the operating mode 'Device Profile DSP 402' is available in all COMPAX variations. It can be activated **via P190=22**. It should be observed here that P190 must already be set to this value at 'Power-On' in order for the switch on procedure defined in the DSP402 profile is run completely (this is different from the normal behavior of the COMPAX after 'Power-On').
In accordance with DSP 402 specification there are a series of states which are assumed by the device in sequence. The corresponding status equipment is integrated in the COMPAX. These states are described in the documentation for 'DSP402' using diagrams and tables. The following description of the device states is therefore only intended as a supplement to the DSP402 documentation:

Conditions diagram



Note! P189=0 produces "xx00h" here: P189=15 will give you xx60h.

When operating mode 'Device Profile DSP 402' is activated, the COMPAX command OUTPUT A0=... must not be used.

Status	Description
NOT READY FOR SWITCHING ON:	◆ Self-test is running ◆ Initialisation is running ◆ Drive function is disabled ◆ All functions are disabled
SWITCHING ON DISABLED:	◆ Software/hardware initialization is terminated ◆ Communication via all interfaces is enabled ◆ Modifications to parameters, variables, records is possible ◆ Drive function is disabled (motor dead) ◆ Display shows 'OFF'
READY TO SWITCH ON:	◆ Communication through all interfaces is enabled ◆ Modifications to parameters, variables, records is possible ◆ Drive function is disabled (motor dead) ◆ Display shows 'OFF'
SWITCHED ON:	◆ Communication through all interfaces is enabled ◆ Modifications to parameters, variables, records is possible ◆ Drive function is disabled (but motor has current) ◆ Display shows 'run'
OPERATION ENABLE:	◆ Communication through all interfaces is enabled ◆ Modifications to parameters, variables, records is possible ◆ Drive function is enabled (motor has current) ◆ Display shows 'run'
QUICK STOP ACTIVE:	◆ Communication through all interfaces is enabled ◆ Modifications to parameters, variables, records is possible ◆ Drive implements the STOP function (motor has current) ◆ Display shows 'run'
FAULT:	◆ Communication through all interfaces is enabled ◆ Modifications to parameters, variables, records is possible ◆ Drive function is disabled ◆ Display shows 'Exx'
ERROR REACTION:	◆ Error handling only, no static device status

8 CANopen COMPAX parameter

No.	Meaning		Minimum value	Default value	Maximum value	When valid
P190	Sets "Device profile DSP 402" operating mode	="0": "DSP402 " operating mode inactive ="22": "DSP402 " operating mode active				Power on
P191	Bus time-out	="0": no response, except error message E73, during a time-out ="1": stop with E73 and shut down during activation of holding brake				VP
P193	Pop-up messages	="1": autom. Error message ="2": autom. "position reached" - message ="4": autom. comparator switch points report				immedia-tely
P194	Address of unit		0	99	99	VP
P195	Baud rate. Possible settings (in baud): 20 000 ● 50 000 ● 100 000 ● 125 000 ● 250 000 ● 500 000 ● 800 000 ● 1 000 000		0	9600	1 000 000	Power on
P196	Protocol	Bit 0 = 0 pop-up messages are reported with EMCY Bit 0 = 1 Pop-up messages are displayed in the status word Bit 1 = 0 boot-up object with 0 data byte Bit 1 = 1 boot-up object structure as EMCY; data=0 Bit 2 = 0 boot-up object only transmitted after Power on Bit 2 = 1 boot-up object also transmitted after a START message				Power on
P135	Index and subindex of 2nd object on the TPDO1 after power on*.		0	0	16777215	Power on
P136	Index and subindex of 3rd object on the TPDO1 after power on*.		0	0	16777215	Power on
P137	Index and subindex of 1rd object on the TPDO2 after power on*.		0	0	16777215	Power on
P138	Index and subindex of 2rd object on the TPDO2 after power on*.		0	0	16777215	Power on
P139	Index and subindex of 2nd object on the RPDO1 after power on*.		0	0	16777215	Power on
P140	Index and subindex of 3nd object on the RPDO1 after power on*.		0	0	16777215	Power on
P141	Index and subindex of 1nd object on the RPDO2 after power on*.		0	0	16777215	Power on
P142	Index and subindex of 2nd object on the RPDO2 after power on*.		0	0	16777215	Power on
P203	Assigning status S16 and S17 to CPX_ZSW		Bit 0 ="0" CPX_ZSW (standard assignment) Bit 0 ="1" S16, S17 to CPX_ZSW			immedia-tely
P221	Standard functions of the digital inputs accessible from the CONTROLWORD object. Physical inputs freely available. Is written/read by object INPUT_MASK (LSB).		0	0	255	immedia-tely
P222	Standard functions of the digital inputs E9...E16 accessible from the CONTROLWORD object. Physical inputs freely available. Is written/read by object INPUT_MASK (MSB).		0	0	255	immedia-tely
P223	Outputs A1...A8 are accessible from object OUTPUT_WORD. Is written/read by object OUTPUT_MASK (LSB).		0	0	255	immedia-tely
P224	Outputs A9...A16 are accessible from object OUTPUT_WORD. Is written/read by object OUTPUT_MASK (MSB).		0	0	255	immedia-tely

9 COMPAX error message of the CAN-Bus

No.	Cause	Remedy / Causes	Acknowledge with	No power to drive
E73	Node monitoring error The error response is influenced with P191.	No longer connection with master Monitoring time and/or lifetime factor not set correctly	[1]	no[2]

[1] Start_Remote_Node indication.
COMPAX automatically goes into pre-operational state due to the emergence of a node monitoring error.
[2] Depends on P191.

* value to be entered=index*256+subindex

10 Overview: manufacturer-specific objects

10.1 Objects sorted by index

Command	Service	Index	Subind.	Byte	PDO	See page
Control byte	rd/wr	0x2000	0	1	R	32
Status byte	rd	0x2001	0	1	T	28
COMPAX control word	rd/wr	0x2002	0	2	R	29
COMPAX status word	rd	0x2003	0	2	T	30
Control commands	wr	0x2004	0	1	R	32
Command input in ASCII format	rd/wr	0x2005	0	20	-	33
Read/write parameters in binary format	rd/wr	0x2006	1...250	4	R	33
Settings for processing RPDO	rd/wr	0x200F	0	2	-	21
Current motor torque	rd	0x2010	0	2	T	39
Temperature of the power output stage	rd	0x2011	0	2	-	34
Control voltage and intermediate circuit voltage	rd	0x2012	1...2	2	-	40
Number of axis motion cycles	rd	0x2013	0	4	-	34
COMPAX operating hours	rd	0x2014	0	4	-	34
Loop counter for a running REPEAT loop	rd	0x2015	0	2	-	35
Status bits 1 and status bits 2.	rd	0x2016	1...2	1	-	40
Diagnosis values	rd	0x2017	1...4	1	-	35
Error message	rd	0x2018	1...2	1	-	36
Target position	rd	0x2020	0	4	-	35
Contour error	rd	0x2021	0	2	T	44
Position of the absolute value sensor	rd	0x2022	0	4	-	42
Set point speed and position default	wr	0x2023	0	6	R	45
Special commands for COMPAX XX70	wr	0x2024	1...6	4	-	45
Synchronization with automatic reverse travel	wr	0x2025	0	4	-	64
Synchronization without automatic reverse travel	wr	0x2026	0	4	-	64
Traverse speed	rd/wr	0x2030	0	2	R	46
Reduce traverse speed	rd/wr	0x2031	0	1	R	46
Change traverse speed	wr	0x2032	0	3	-	46
Speed step profile	wr	0x2033	0	8	-	47
Speed step profile with ACCEL	wr	0x2034	0	10	-	47
Current axis process speed	rd	0x2035	0	2	T	49
Mask inputs	rd/wr	0x2040	0	2	-	52
Mask outputs	rd/wr	0x2041	0	2	-	54
One digit. a digital output	wr	0x2042	1...16	1	-	52
Comparator function (active high)	wr	0x2043	1...16	4	-	55
Comparator function (active low)	wr	0x2044	1...16	4	-	55
Read/write program memory (binary)	rd/wr	0x2050	1...250	20	-	56
Set and read pointer	rd/wr	0x2051	0	1	-	62
Run program record N	wr	0x2052	0	1	R	62
Start program from record N	wr	0x2053	0	1	R	62
Transfer current position in record N	wr	0x2054	0	1	-	63
Set and read curve memory pointer	rd/wr	0x2055	0	2	-	63
Read and write curve memory	rd/wr	0x2056	0	3	-	63
Enter or read COMPAX variables	rd/wr	0x2057	1...40	4	R	64
Fault code	rd	0x603F	0	2	-	37
Control word	rd/wr	0x6040	0	2	R	32
Status word	rd	0x6041	0	2	T	31
Operation mode selection code	rd/wr	0x6060	0	2	-	27
Operation mode display	rd	0x6061	0	2	-	27
Position actual value	rd	0x6064	0	4	T	41

Command	Service	Index	Subind.	Byte	PDO	See page
Contour error window	rd/wr	0x6065	0	4	-	43
Positioning window	rd/wr	0x6067	0	4	-	43
Traverse speed actual value	rd	0x606C	0	4	-	49
Torque max. value	rd/wr	0x6072	0	2	-	38
Motor nominal current	rd/wr	0x6075	0	2	-	39
Rated torque motor	rd/wr	0x6076	0	2	-	38
Torque actual value	rd	0x6077	0	2	-	38
Intermediate circuit voltage	rd	0x6079	0	2	-	39
Target position default	rd/wr	0x607A	0	4	R	41
Reference measurement offset	rd/wr	0x607C	0	4	-	42
Position limit value min-max	rd/wr	0x607D	1...2	4	-	43
Polarities	rd/wr	0x607E	0	1	-	44
Max speed value	rd/wr	0x607F	0	4	-	48
Max motor speed value	rd/wr	0x6080	0	2	-	48
Traverse speed	rd/wr	0x6081	0	4	-	48
Acceleration	rd/wr	0x6083	0	4	-	50
Lag	rd/wr	0x6084	0	4	-	51
Rapid stop	rd/wr	0x6085	0	4	-	51
Ramp form speed	rd/wr	0x6086	0	2	-	50
Reference run selection code	rd/wr	0x6098	0	2	-	44
Reference run speed	rd/wr	0x6099	1...2	4	-	49
Accelerate reference run	rd/wr	0x609A	0	4	-	51
Logic state of the 16 digit. inputs	rd	0x6100	0	2	T	52
Logic state of the 16 digit. Outputs	rd/wr	0x6300	0	2	T/R	52

10.2 Objects sorted by command

Command	Service	Index	Subind.	Byte	PDO	See page
Accelerate reference run	rd/wr	0x609A	0	4	-	51
Acceleration	rd/wr	0x6083	0	4	-	50
Change traverse speed	wr	0x2032	0	3	-	46
Command input in ASCII format	rd/wr	0x2005	0	20	-	33
Comparator function (active high)	wr	0x2043	1...16	4	-	55
Comparator function (active low)	wr	0x2044	1...16	4	-	55
COMPAX control word	rd/wr	0x2002	0	2	R	29
COMPAX operating hours	rd	0x2014	0	4	-	34
COMPAX status word	rd	0x2003	0	2	T	30
Contour error	rd	0x2021	0	2	T	44
Contour error window	rd/wr	0x6065	0	4	-	43
Control byte	rd/wr	0x2000	0	1	R	32
Control commands	wr	0x2004	0	1	R	32
Control voltage and intermediate circuit voltage	rd	0x2012	1...2	2	-	40
Control word	rd/wr	0x6040	0	2	R	32
Current axis process speed	rd	0x2035	0	2	T	49
Current motor torque	rd	0x2010	0	2	T	39
Diagnosis values	rd	0x2017	1...4	1	-	35
Enter or read COMPAX variables	rd/wr	0x2057	1...40	4	R	64
Error message	rd	0x2018	1...2	1	-	36
Fault code	rd	0x603F	0	2	-	37
Intermediate circuit voltage	rd	0x6079	0	2	-	39
Lag	rd/wr	0x6084	0	4	-	51
Logic state of the 16 digit. inputs	rd	0x6100	0	2	T	52
Logic state of the 16 digit. Outputs	rd/wr	0x6300	0	2	T/R	52

Command	Service	Index	Subind.	Byte	PDO	See page
Loop counter for a running REPEAT loop	rd	0x2015	0	2	-	35
Mask inputs	rd/wr	0x2040	0	2	-	52
Mask outputs	rd/wr	0x2041	0	2	-	54
Max motor speed value	rd/wr	0x6080	0	2	-	48
Max speed value	rd/wr	0x607F	0	4	-	48
Motor nominal current	rd/wr	0x6075	0	2	-	39
Number of axis motion cycles	rd	0x2013	0	4	-	34
One digit. a digital output	wr	0x2042	1...16	1	-	52
Operation mode display	rd	0x6061	0	2	-	27
Operation mode selection code	rd/wr	0x6060	0	2	-	27
Polarities	rd/wr	0x607E	0	1	-	44
Position actual value	rd	0x6064	0	4	T	41
Position limit value min-max	rd/wr	0x607D	1...2	4	-	43
Position of the absolute value sensor	rd	0x2022	0	4	-	42
Positioning window	rd/wr	0x6067	0	4	-	43
Ramp form speed	rd/wr	0x6086	0	2	-	50
Rapid stop	rd/wr	0x6085	0	4	-	51
Rated torque motor	rd/wr	0x6076	0	2	-	38
Read and write curve memory	rd/wr	0x2056	0	3	-	63
Read/write parameters in binary format	rd/wr	0x2006	1...250	4	R	33
Read/write program memory (binary)	rd/wr	0x2050	1...250	20	-	56
Reduce traverse speed	rd/wr	0x2031	0	1	R	46
Reference measurement offset	rd/wr	0x607C	0	4	-	42
Reference run selection code	rd/wr	0x6098	0	2	-	44
Reference run speed	rd/wr	0x6099	1...2	4	-	49
Run program record N	wr	0x2052	0	1	R	62
Set and read curve memory pointer	rd/wr	0x2055	0	2	-	63
Set and read pointer	rd/wr	0x2051	0	1	-	62
Set point speed and position default	wr	0x2023	0	6	R	45
Settings for processing RPDO	rd/wr	0x200F	0	2	-	21
Special commands for COMPAX XX70	wr	0x2024	1...6	4	-	45
Speed step profile	wr	0x2033	0	8	-	47
Speed step profile with ACCEL	wr	0x2034	0	10	-	47
Start program from record N	wr	0x2053	0	1	R	62
Status bits 1 and status bits 2.	rd	0x2016	1...2	1	-	40
Status byte	rd	0x2001	0	1	T	28
Status word	rd	0x6041	0	2	T	31
Synchronization with automatic reverse travel	wr	0x2025	0	4	-	64
Synchronization without automatic reverse travel	wr	0x2026	0	4	-	64
Target position	rd	0x2020	0	4	-	35
Target position default	rd/wr	0x607A	0	4	R	41
Temperature of the power output stage	rd	0x2011	0	2	-	34
Torque actual value	rd	0x6077	0	2	-	38
Torque max. value	rd/wr	0x6072	0	2	-	38
Transfer current position in record N	wr	0x2054	0	1	-	63
Traverse speed	rd/wr	0x2030	0	2	R	46
Traverse speed	rd/wr	0x6081	0	4	-	48
Traverse speed actual value	rd	0x606C	0	4	-	49

The PDO column indicates whether the corresponding object can be imaged on a process data object. The information in the PDO column means the following:

- T Object can be imaged on a transmit PDO.
- T Object can be imaged on a receive PDO.
- T/R Object can be imaged on a receive and a transmit PDO.
- - imaging on a PDO is not possible.

11 Index