



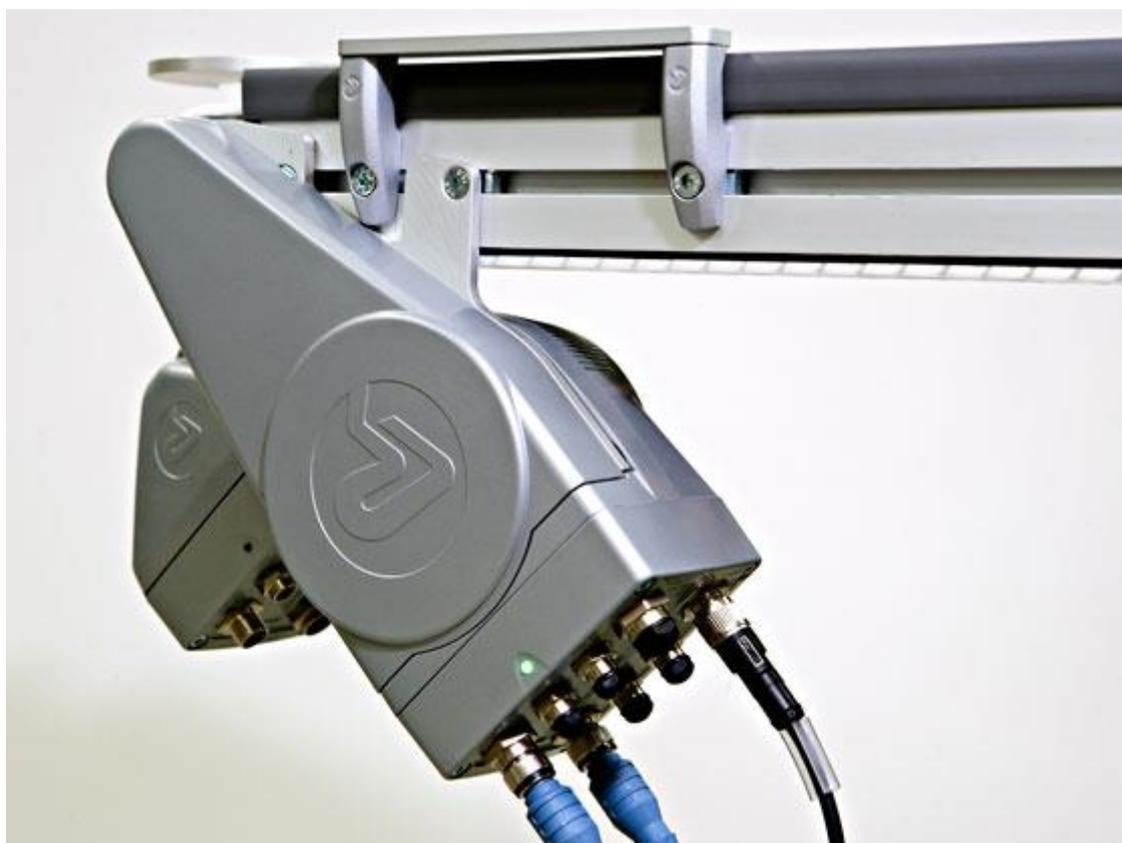
**FLEXLINK**<sup>®</sup>

a coesia company

5113269-02

# X45e ELECTRICAL SYSTEM

USER DOCUMENTATION



## Table of content

OVERVIEW.....	3
HARDWARE .....	5
Integrated motor.....	5
Electrical installation .....	9
SOFTWARE .....	14
Motor software.....	14
Sensor functionality .....	22
Line control .....	24
External communication .....	25
PROFINET .....	28
Ethernet/IP .....	32
APPENDIX.....	35
Speed parameter settings .....	35
Cycle time (Function units).....	36

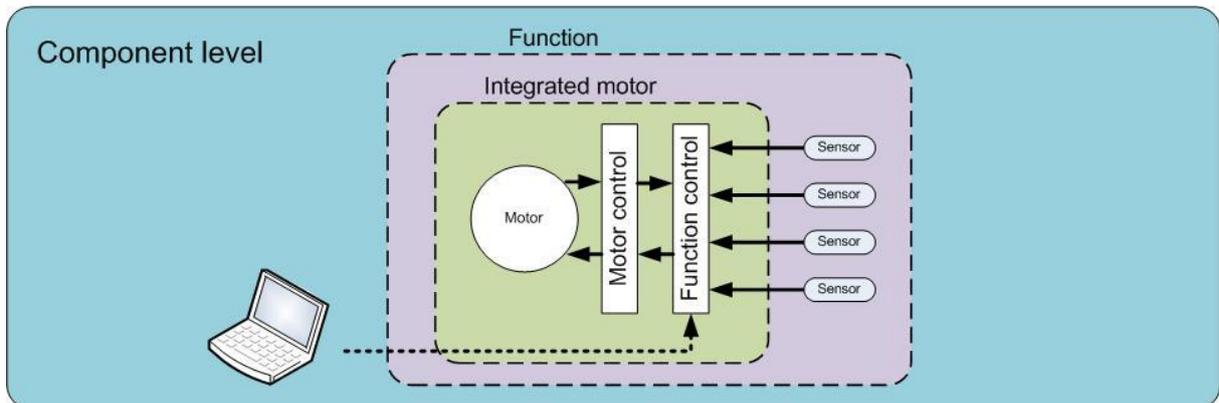
## Overview

The electrical and control part of the platform X45e are structured in an object oriented way. All motors in the platform have an integrated distributed control system and the local sensors are connected directly to the motor unit. It is a big advantage both regarding software developing, electrical design as well as electrical installation.

The X45e units was originally designed with a CAN interface giving the fieldbus options of either DeviceNet or CANopen. This product has been replaced by a newer generation of X45e units based on Ethernet. This document is only covering the new version.

See the **User documentation, Electrical system description, 5113269** for more details about the original version.

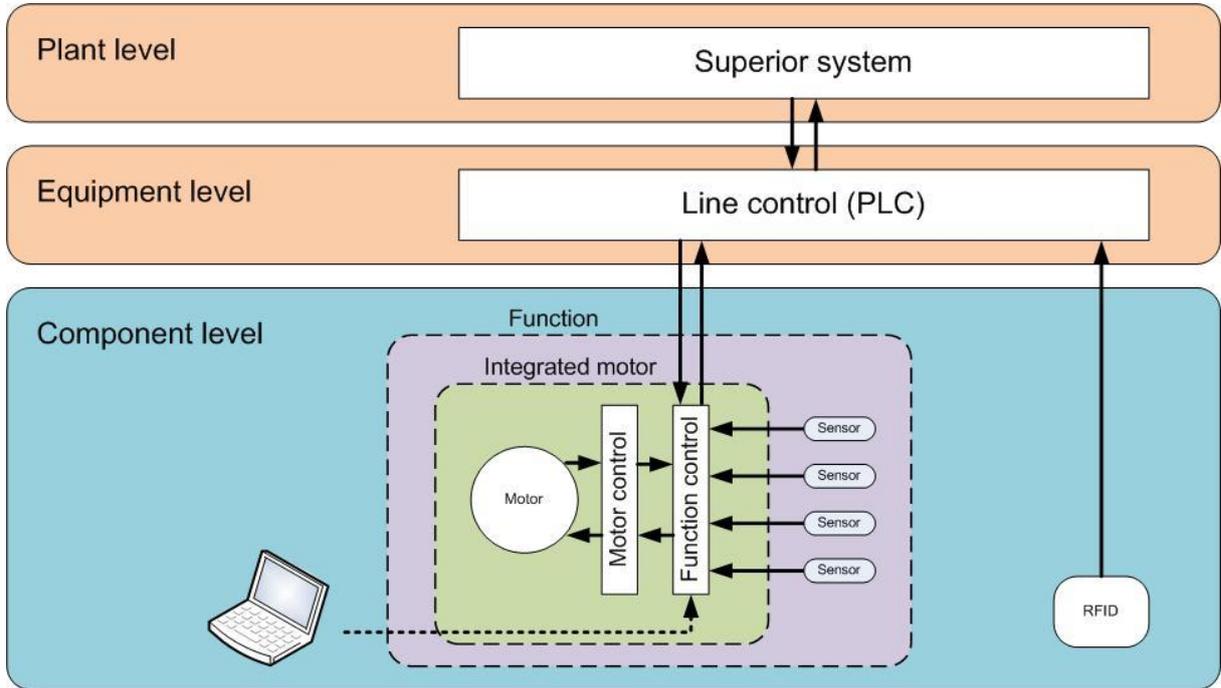
The control hierarchy in the platform can be implemented in two ways. The easiest solution is a stand-alone control of the motors with no control system in the level above. All input needed for the motors to execute their task are gathered via the digital inputs. Then also digital outputs exists for feedback signals.



Picture: Autonomous control

The other way to implement the control is to connect the motors via a network. The motors are executing their task in the distributed control but commands based on information on system level can be sent from the line controller through the Ethernet line.

If RFID are to be used, there can be a separate network for this purpose.



Picture: Line control

## Hardware

### *Integrated motor*

The motors in the X45e platform are integrated motors with distributed controllers and sensor connectors. It exists in two variants. The drive motor is a permanent magnetic motor and the function motor is a stepper motor.

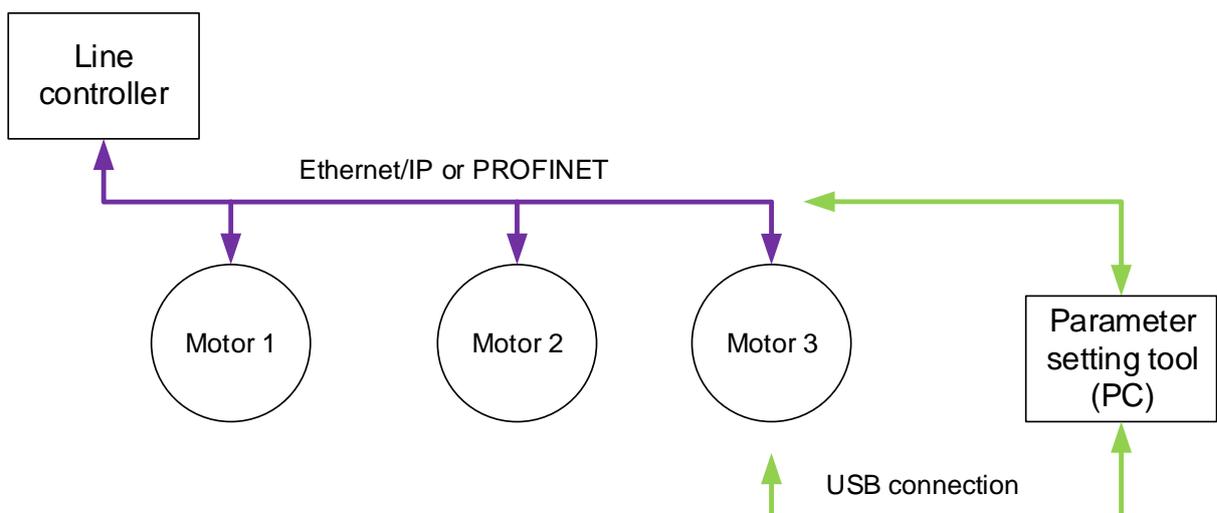


*Picture: Drive motor and function motor*

Both motors share the same integrated control box. This contains of two electric circuit boards. One (ECO) is dedicated for the motor control and the other (FEG) is the application handling the application level. This control box is also equipped with connectors mounted directly to the circuit board.

### Communication alternatives

In order to communicate with the motor from the Parameter Setting Tool software there are two alternatives:



#### Ethernet/IP or PROFINET

All motors can be set to communicate up to a line controller via Ethernet. The communication options are Ethernet/IP or PROFINET. Via the Parameter Setting Tool software it is possible to connect to this network for reading and writing parameters to all motors.

#### USB connection

The motor units have also an internal CAN network. This is used for communication between the two processors inside the motor. If Parameter Setting Tool software is used via USB it is also possible to log communication between processors. A standard mini-USB cable is required which is not included in delivery of the motor.

## Motor interface

The control box on the motors has eight connectors of M12 type.



*Picture: Connector description*

**ETH1** – External bus connector 1, M12 D coded  
Connect this if external Ethernet network is used (PROFINET or Ethernet/IP)

**ETH2** – External bus connector 2, M12 D coded  
Connect this if external Ethernet network is used (PROFINET or Ethernet/IP)

**X 4** – IO connector 4, M12 A coded  
DI4 + DI8 + DO4  
See section below for pin configuration

**X 3** – IO connector 3, M12 A coded  
DI3 + DI7 + DO3  
See section below for pin configuration

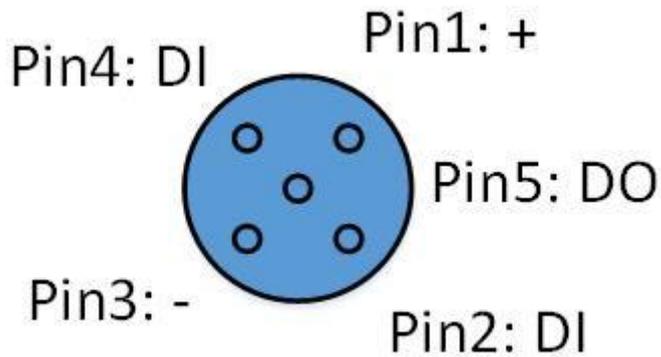
**X 2** – IO connector 2, M12 A coded  
DI2 + DI6 + DO2  
See section below for pin configuration

**X 1** – IO connector 1, M12 A coded  
DI1 + DI5 + DO1  
See section below for pin configuration

**PWR** – 24VDC Power in, M12 A coded  
Connect power cable for the unit.  
See separate description in the chapter regarding the electrical installation

**USB** – connection to unit with mini-USB:  
This connector is used for interaction via the PST software.  
See the **User documentation, Parameter Setting Tool, 5113270** for more details regarding the software. A standard mini-USB cable is required which is not included in delivery of the motor.

The connectors X1-X4 are multi-I/O connectors. The connector type is M12 A-coded with five pins.



*Picture: I/O Connector, pin description*

The primary input signal is always on pin 4. When a normal sensor cable is attached to connector X1-X4 the inputs DI1-DI4 is automatically used. The secondary inputs are located on pin 2. In order to use this input a splitter or T-connector has to be added to the X-connector. Every I/O-connector has also one digital output that can be used. This is connected to pin 5.

## Electrical installation

The electrical cabinet for this system have to be designed according to some rules. The main function with the cabinet is to supply power to the motors. Additional features can be line control and safety functions. Only the X45e specific features are described in this document.

## Power Supply

The motor units are supplied with power through a M12 A-coded contact (male contact on the motor).

Pin	Signal name	Direction	Function
1.	+24_POWER	In	+24 VDC Supply for motor drives, (Safety)
2.	GND	Out	Ground
3.	GND	Out	Ground
4.	+24_LOGIC	In	+24 VDC (continuous)
5	NC	--	Not connected

*Picture: Pin configuration power supply contact*

As can be seen in the table above both LOGIC (continuous power to the electronics) and POWER (drive power interrupted by emergency stop) has to be distributed. This can be achieved by two separate flat cables (AS-Interface, auxiliary power type) and a connection cable that gathers those two flat cables into one drop cable with the appropriate M12 contact. The components below can be used for this purpose.

- Phoenix contacts: VS-ASI-J-Y-B-M12FS or VS-ASI-J-Y-B-PUR-1,0-M12Fs SCO
- Bihl Wiedemann: BW1974

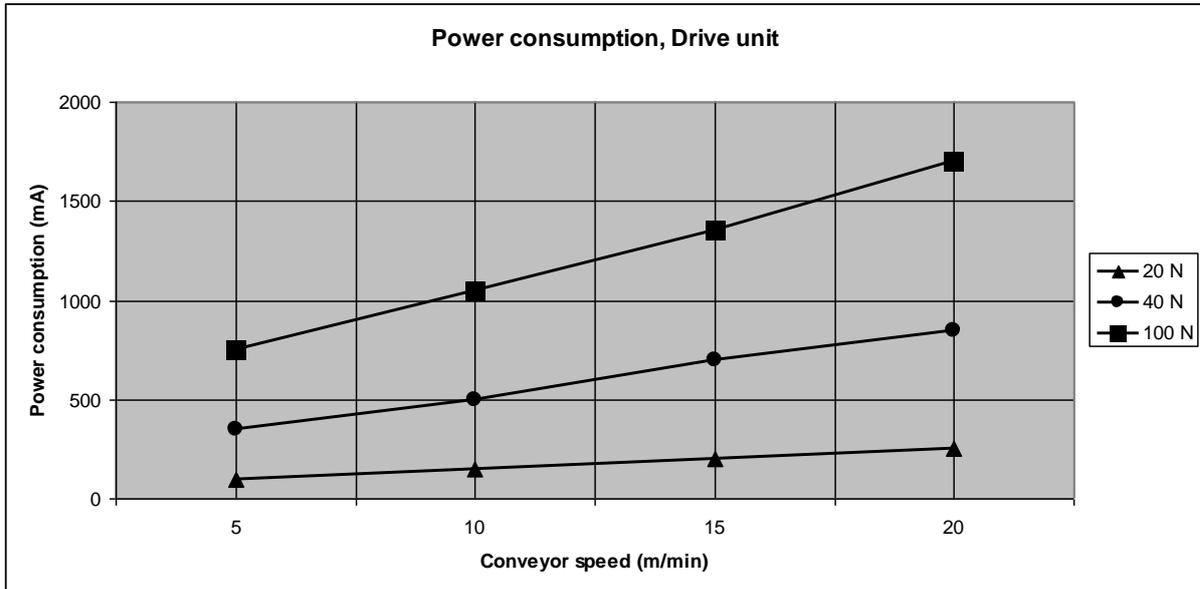
The AS-I flat cables are rated maximum 8A. Many vendors can also supply a number of useful additional components in this AS-I concept (T-connectors, splitters, ...).  
A main 24 VDC power supply is needed in every X45e system. The consumers of power can be divided in the following categories:

- Drive power supply
- Electronics power supply
- External bus power supply

### Drive power supply:

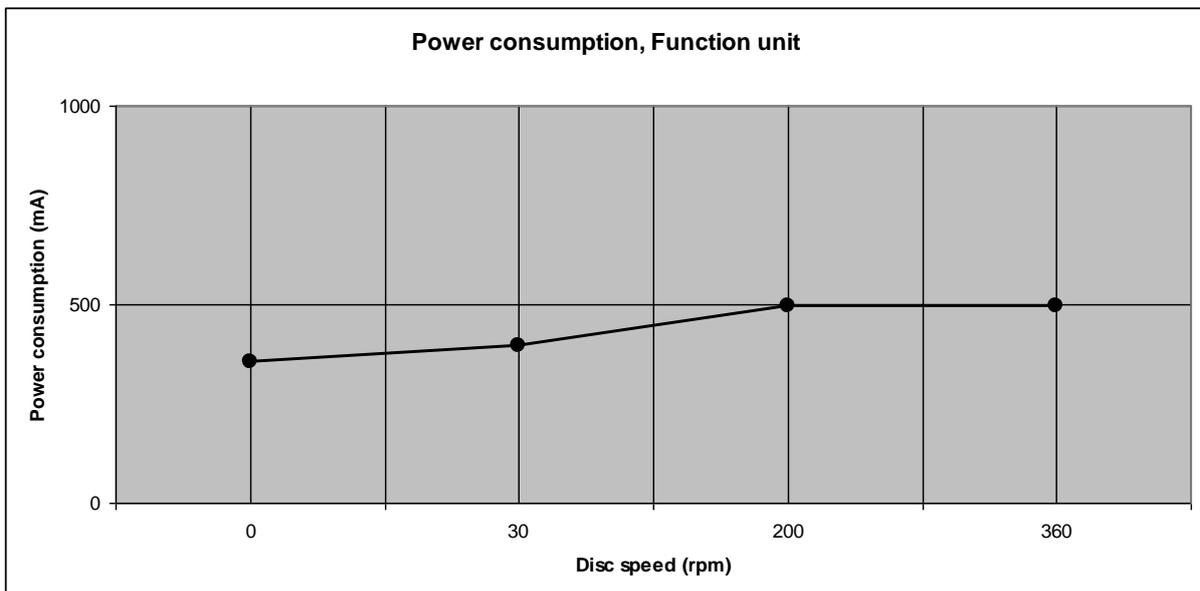
The power consumption for a drive unit (conveyor motor) depends mainly on two parameters, conveyor speed (x axis) and chain pull (the different curves).  
(Use FlexLink Calculation Tool to calculate actual force)

Maximum continuous current for a drive unit: 1.7 A (20 m/min and 100 N)



Picture: Power consumption, Drive unit

The power consumption for a function unit depends only on rotation speed of the rotation disc. This relation is not linear due to the fact that there has to be a stand still torque. Maximum continuous current for a function motor: 0.5 A



Picture: Power consumption, Function unit

Electronics power supply:

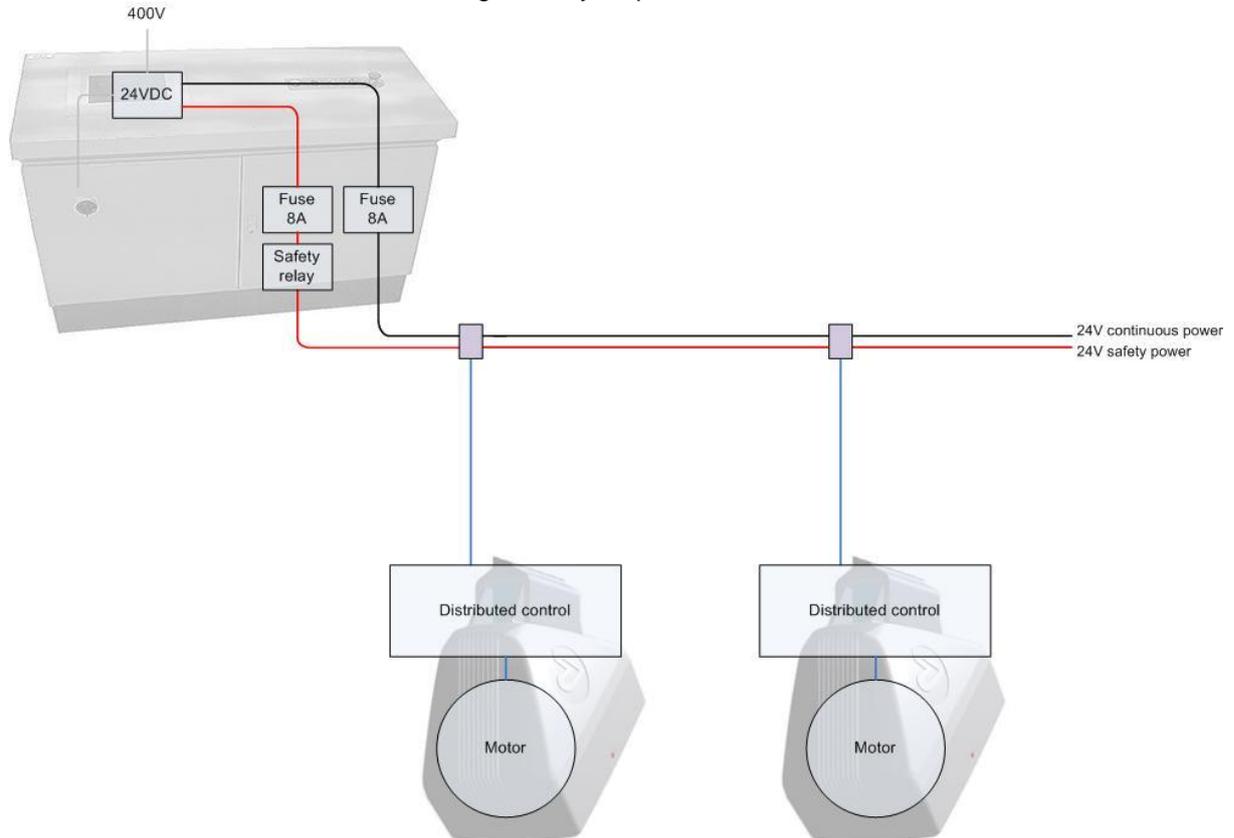
Each motor unit has continuous power consumption of approximately 130 mA for feeding all electronic components.

## Safety function

The motors are prepared for the ability to cut the drive power but keep the logic power to the units. If the system will be designed with this feature some safety components has to be added in the main cabinet.

The power line with safe power has to be cut by a safety relay or a corresponding unit. This power supplies all the motors with drive power.

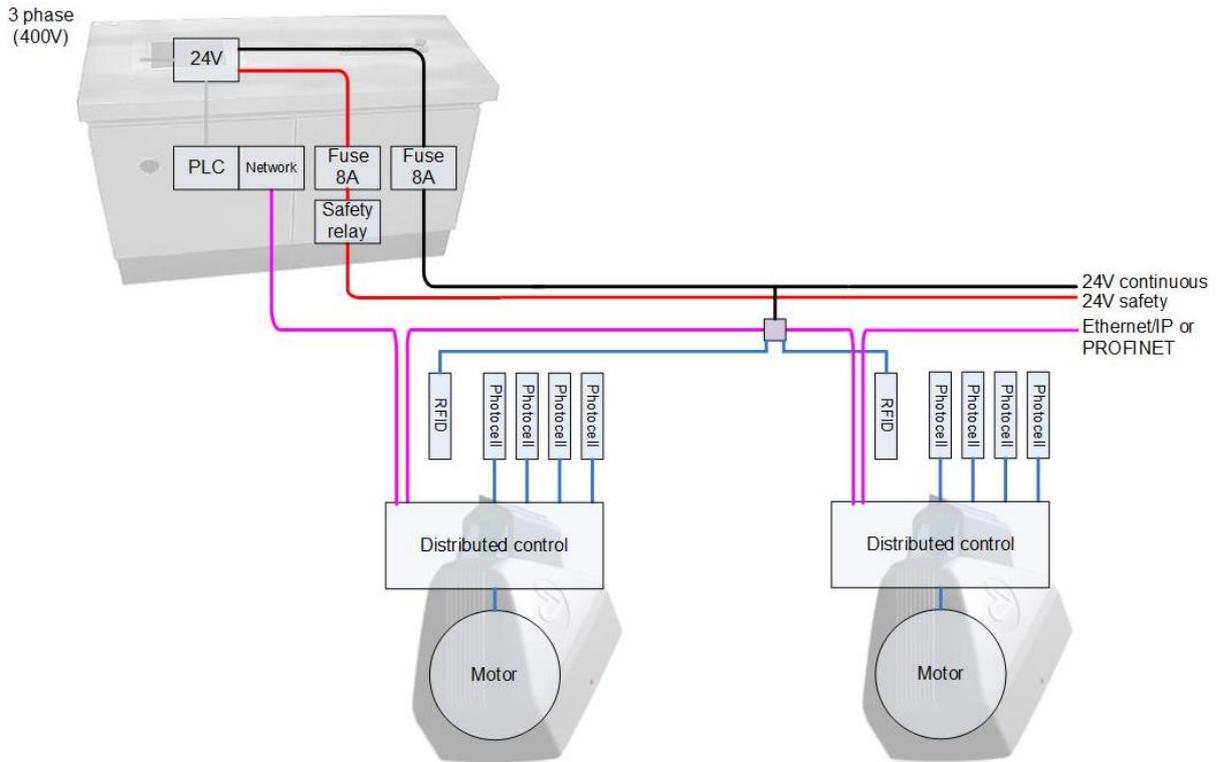
The continuous power line supplies all electrical components inside the motors. This makes the motors able to communicate even during a safety stop.



During a loss of only safety power (drive power to the motors) the motors get an internal alarm, Low UDC. This can be read out via the monitoring function in the Parameter Setting Tool. The motor units are internally keeping their state even though they have no drive power. When the safety power is back the Low UDC alarms are automatically reset and the unit continues with its task.

## Cabling

One big advantage with the X45e platform is the process of installation and cabling. The example below shows a setup with line controller and RFID.



Picture: Overview of electrical installation, cabling

## External network (ETH)

The motor nodes have two standard M12 D-coded contacts for external bus connections. There are no difference between those two connectors. They are both connected to an internal switch. This network can be either Ethernet/IP or PROFINET. The installation method is described separately for each protocol on the next pages.

## Sensors

All sensors that are needed for each function are already connected before delivery. Each motor unit has four M12 connectors for digital input and outputs. If some of them are not connected by the delivered function there is a possibility to connect them with other sensors and read these signals over the external bus. The sensors exist in two variants only differed by the sensing distance and one variant with reflector.

Sensors, SICK

GTB2S-P5331 – 30 mm  
WTB2S-2P3160 – 60 mm  
GL2S- F5311 – reflector

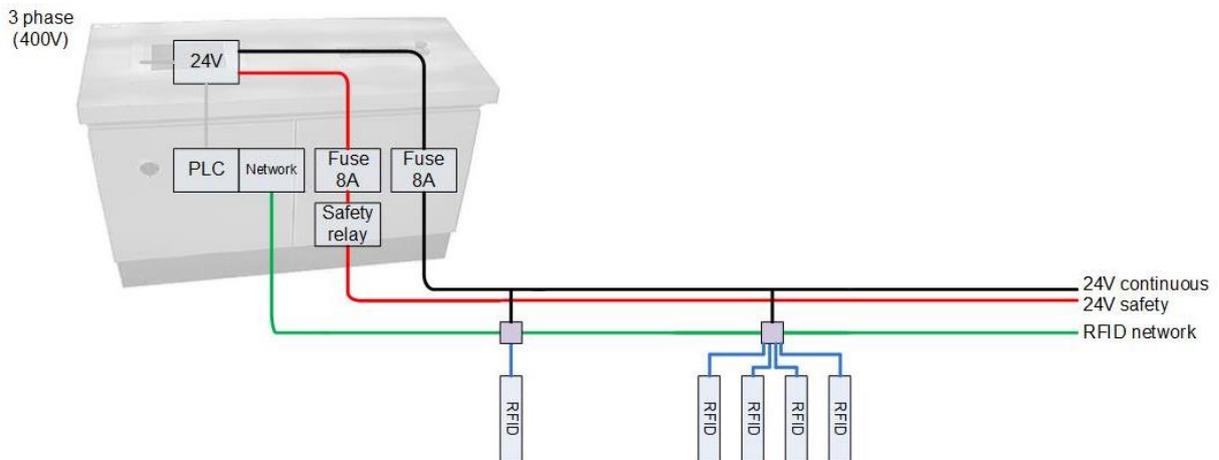
## RFID

The RFID readers/writer used in the X45e platform are IDENT from Pepperl+Fuchs. These units communicate with a line controller via a field bus unit. This field bus unit can be ordered with many different standard protocols e.g. Profibus, Profinet, DeviceNet and Ethernet IP. The field bus exists in different variants which can connect to 1, 2 or 4 reader/writer units.

Recommended RFID components: (Pepperl+Fuchs)

IQH1-F61-V1: Read/write head  
IC-KP-B12-V45: Field bus node (4 channels, Ethernet IP/Profinet IO)  
IC-KP2-2HB17-2V1D: Field bus node (2 channels, Ethernet IP/Profinet IO)  
IQC33-30: RFID Tag, 16 kBit  
IQC21-30P: RFID Tag, 1024 Bit  
V1-G-5M-PUR-ABG-V1-W: Cable Read/write head – Field bus node

See FlexLink Product catalogue!



Picture: RFID installation

## Software

### **Motor software**

Each motor has two microprocessors for achieving the desired functionality of the motor. The first one is the drive controller (ECO). This controller sends the actual control signals to the motors and receives feedback. The second microprocessor (FEG) runs the application program which completes the function. This controller receives signals from the sensors and communicates both down to the drive controller and up the line controller, if used. This controller has a set of parameter that can be changed in order to achieve a certain function.

How to set these parameters from a PC can be read in the user documentation for the Parameter setting tool (51113270 in the Technical library at flexlink.com).

### **Motor parameters**

#### Motor type

1 = Conveyor motor (PM)

2 = Function motor (Stepper motor)

This parameter is read only and is pre-set from the production.

#### Function mode

Only applicable to a Function motor

1 = Diverter

2 = Merger – Combined Diverter/Merger

3 = Transfer – Stop – Locating unit

4 = Position mode

5 = Speed mode

#### Calibrate

Execute internal calibration sequence for a Function motor if Motor Type = 2

Internal FEG Calibrate sequence:

1. ECO standing still.

2. FEG changes the mode to: Calibrate mode (PDO2)

3. ECO saves the position and leaves Calibrate mode.

4. Position value in PDO1 is now updated to 0.

5. FEG Request 0 speed. (PDO1)

Thumb of rule for homing position:

- *The home position is always the forward direction of the conveyor the function unit is mounted on.*

#### Node ID

Corresponds to the last part of the IP address.

Used as node ID on the internal bus.

#### Speed setpoint 2

The speed of a Conveyor motor or a Function motor

Conveyor motor

Second speed for special usage like reverse direction

If the motor is a Right version positive speed values runs the conveyor in forward direction.

If the motor is a Left version negative speed values runs the conveyor in forward direction.

Speed value interval:  
Right version: 80 – 400 rpm  
Left version: - 80 – - 400 rpm

Function motor  
Speed used during turn  
Interval: 0 - 360 rpm

#### Speed setpoint 1

The speed of a Conveyor motor or a Function motor

Conveyor motor  
Standard speed  
If the motor is a Right version positive speed values runs the conveyor in forward direction.  
If the motor is a Left version negative speed values runs the conveyor in forward direction.

Speed value interval:  
Right version: 80 – 400 rpm  
Left version: - 80 – - 400 rpm

Function motor  
Speed used pre turn and post turn  
Interval: 0 - 360 rpm

#### Torque

The torque of Conveyor motor  
Interval: 0 – 100 %

#### Receive angle 1

The angle position 0 – 360° for receiving pucks from conveyor 1, counted clockwise relative the home position.

#### Receive angle 2

The angle position 0 – 360° for receiving pucks from conveyor 2, counted clockwise relative the home position.

#### Receive angle 3

The angle position 0 – 360° for receiving pucks from conveyor 3, counted clockwise relative the home position.

#### Release angle 1

The angle position 0 – 360° for releasing pucks to conveyor 1, counted clockwise relative the home position.

#### Release angle 2

The angle position 0 – 360° for releasing pucks to conveyor 2, counted clockwise relative the home position.

#### Wait angle 1

When the diverter has released a puck on conveyor 1 it can be set to go to a waiting position. The next receive command is completing the motion to the receive angle. This angle position is also counted clockwise relative the home position.

### Wait angle 2

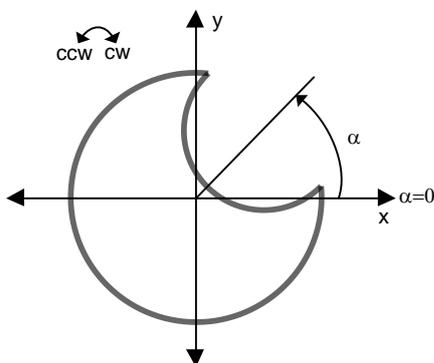
When the diverter has released a puck on conveyor 2 it can be set to go to a waiting position. The next receive command is completing the motion to the receive angle. This angle position is also counted clockwise relative the home position.

### Acceleration ramp

The speed increasing ramp of a Function or a Conveyor motor  
Increasing interval: 1 – 300

### Deceleration ramp

The speed decreasing ramp of Function or a Conveyor motor  
Decreasing interval: 1 – 300



### Enable drive

Enable the drive in Function or a Conveyor motor

0 = Disable

1 = Enable

Only valid if *A-bus enable drive and reset* is set to 0.

### Reset errors

Reset all errors by sending 1. If the value re-appears in PDO2Tx[Errors] (from ECO) the error is still valid

### Mode (debug mode)

Activates test modes in FEG. This parameter value is automatically reset after reset.

0 = No test

1 = Flashing with LEDs

2 = Show digital input levels on LEDs

3 = Start calibration mode

128 = Command to reboot the FEG processor.

### Function Parameter 1, 2, 3, 4

See Appendix for detailed description

### Requested angle

This value is not set by the operator. It is updated by the homing function (Calibration).

### Pre turn delay

Timer value used for delays (in motor state Pre turn)

During turn delay

Timer value used for delays (in motor state During turn)

Post turn delay

Timer value used for delays (in motor state Post turn)

Interlock delay (on)

Timer value used for delays (max queue)

Interlock delay (off)

Timer value used for delays (max queue)

Automatic reset errors

1 = FEG reset errors automatically when they appear

0 = Leave errors to be cleared by Line controller

Interlock mode

0 = No interlock

1 = Normally open

The interlock function is blocking the unit from running when the input (DI4) is active.

This could be interpreted as a max queue mode.

The two interlock timers are activated.

2 = Normally closed

The interlock function is blocking the unit from running when the input (DI4) is inactive.

This could be interpreted as a sensor enable mode.

The two interlock timers are activated.

3 = NC Advanced

This alternative is a more complex feature.

Enable = DI4 or (DI5 and DI6 and DI7)

A-bus enable drive and reset

0 = Disable drive control and clear error through cyclic messages

This means the bits enable and reset in the cyclic communication are inactivated. These two signals have to be sent by acyclic communication. Typically used for stand-alone control.

1 = Enable drive control and clear error through cyclic communication

This means the bits enable and reset in the cyclic communication are activated. These two signals are sent over Ethernet.

FEG Program version

This value is the revision of the embedded motor software.

This value is read only and can't be set by the operator.

Network protocol (A-Bus / ETH)

This value is setting the network type on the external bus (A-bus).

0 = No network

1 = CANopen (not applicable for this hardware)

2 = DeviceNet (not applicable for this hardware)

3 = Ethernet/IP

4 = PROFINET

Operating motor power

Momentary power consumption. Accuracy 1/100 W  
Calculated from voltage and current parameters  
Only valid for a conveyor motor.

Operating years

Accumulated time when the enable signal is on (years)  
This value is stored in the permanent memory of the motor unit.

Operating days

Accumulated time when the enable signal is on (days)  
This value is stored in the permanent memory of the motor unit.

Operating hours

Accumulated time when the enable signal is on (hours)  
This value is stored in the permanent memory of the motor unit.

Operating minutes

Accumulated time when the enable signal is on (minutes)  
This value is lost when no power is supplied

Operating seconds

Accumulated time when the enable signal is on (seconds)  
This value is lost when no power is supplied

Operating cycles (High word)

Accumulated number of cycles executed by a function motor. This value has to be interpreted together with the next parameter (the low word).  
This value is stored every operating hour.

Operating cycles (Low word)

Accumulated number of cycles executed by a function motor. This value has to be interpreted together with the next parameter (the high word).  
This value is stored every operating hour.

Positioner mode

Operation mode of a Function motor that runs at positioner.  
1 = angular positioner (0 – 360°)  
2 = distance positioner (mm)

Positioner ratio

Ratio of mm per revolution if positioner runs as distance positioner

Function group

Depending on the function the motor is used for (drive, divert, merge ...) it is member of a group.  
For detailed description see documentation of function variants.

Function variant

The function variant is a combination of function group, network protocol, A-bus enable drive and reset, control and variant.  
For detailed description see documentation of function variants.

ECO program version

This value is the revision of the embedded motor software.  
This value is read only and can't be set by the operator.

#### Temperature

This parameter is showing the temperature inside the motor housing. There is an analogue sensor on the PCB board.

#### Debug enable

Enables debug signals. Only for development purpose.

#### Limit switch wait

Timer value used for delays at drive motor using limit switch.

#### Pos Req absolute closest

Command position request. Moves to an absolute position the closest direction

#### Pos request absolute CCW

Command position request. Moves to an absolute position counter clockwise

#### Pos request absolute CW

Command position request. Moves to an absolute position clockwise

#### Pos request relative

Command position request. Moves to an relative position

#### Init to default

When this parameter is set to one (1) all parameters are set back to factory default

#### ECO current angle CCW

Actual position on the motor axis, counted counter clockwise from the home position

#### ECO current angle CW

Actual position on the motor axis, counted clockwise from the home position

#### IP handling automatic

0 = static IP addressing

1= dynamic IP addressing

#### IP address

Address of this motor in network. 4-byte value with little endian byte order (e.g. 192.168.43.225 -> 3777734848).

#### IP subnet mask

Subnet mask of network (e.g. 255.255.255.0). 4-byte value with little endian byte order (e.g. 192.168.43.225 -> 3777734848).

#### IP Gateway

Gateway to use (e.g. motor IP address). 4-byte value with little endian byte order (e.g. 192.168.43.225 -> 3777734848).

#### IP DNS server

DNS server to use (e.g. 0.0.0.0). 4-byte value with little endian byte order (e.g. 192.168.43.225 -> 3777734848).

#### Operating motor current

Momentary motor current value, read from ECO, accuracy 1 mA

Operating motor voltage

Momentary motor voltage value, read from ECO, accuracy 1/100 V

FEG minor version

This value is the minor (second) part of revision of the embedded motor software.  
This value is read only and can't be set by the operator.

FEG patch version

This value is the patch (third) part of revision of the embedded motor software.  
This value is read only and can't be set by the operator.

RabbitMQ mode

Working mode of RabbitMQ (0 = off; 1 = listen only; 2 = control). Requires reboot after change to take effect.

RabbitMQ server

Address of RabbitMQ server to talk to. Requires reboot after change to take effect.

RabbitMQ port

Port at RabbitMQ server to talk to. Requires reboot after change to take effect.

RabbitMQ exchange

Exchange at RabbitMQ server to talk to. Requires reboot after change to take effect.

RabbitMQ device name

Human friendly name of device used in key of RabbitMQ messages. Requires reboot after change to take effect.

RabbitMQ runtime send frequency

Send frequency of runtime messages in seconds.

RabbitMQ power debounce

Debounce for sending messages at changed power of drive unit in percent.

## Motor state

The function motors have an internal sequence in order to complete its cycle.

	State	
0	Rescue	
1	Init	Movement after power up
2	Pre-turn	Movement to receive the puck
3	During turn	Movement to release the puck
4	Post-turn	Movement after releasing the puck
5	Calibration	Movement during calibration
6	Idle	Wait for new order
7	Fault	

*Table: Motor state*

## Motor alarms

The motor unit has seven different faults that can be monitored via the PST (Tool/Monitor) or be read out via the cyclic parameters. Also the LED indicator shows a red light when there is an active alarm.

### Over current

Current exceeds 4.9 A.

### High UDC

Voltage exceeds 29 VDC.

### Low UDC

Voltage is below 19 VDC.

This alarm is not lit the LED to red.

Motor attempts a self-reset when the Low UDC disappears.

### PCB over temp

The internal temperature inside the housing is over 100° C.

### Locked rotor

The position control loop is unable to reach its required speed setpoint within 2 sec.

### Internal fault

Not used

### Output overload

Digital outputs are taking too much current.

## **Sensor functionality**

Both drive motors and function motors have eight digital inputs that can be used in many ways. When no network protocol is selected on the A-bus the control of the motors is only depending on the condition of the first four digital inputs (DI1-4).

### **Drive motors**

#### Interlock

The drive motor is by default configured to always start when powered up. If a start or stop signal is wanted this can be achieved by the interlock function on digital input 4 (DI4). This works also if a network protocol is set.

#### Second speed setpoint

All motors have two speed parameters, speed setpoint 1 and speed setpoint 2. The motor is by default running on speed setpoint 1. For a drive motor without a network protocol there are two way to get the motor to switch between the speed setpoints. The first solution is to simply let one digital input 3 (DI3) control this. The other solution is to have a limit switch functionality. Then both digital inputs DI3 and DI4 are used to set each setpoint.

#### Adjusting speed setpoints

The two speed setpoint parameters can be adjusted via the Parameter Setting Tool software using a PC and a communication cable. There is also a possibility to adjust this using the digital input signals. The digital input 1 (DI1) is increasing the speed setpoint parameter. It increases the values with 1 rpm per second the first two seconds. After this it increases the value with 10 rpm per second. The digital input 2 (DI2) is decreasing the speed setpoint parameter with the same timer functions. This fine tuning procedure is adjusting the speed setpoint that currently is chosen. This means if DI3 is inactive it affects speed setpoint 1 and if DI3 is active it affects speed setpoint 2. This fine tuning is only working if the unit is set with no network protocol.

### **Function motors**

#### Interlock

The function motor is not set to a network protocol; it is by default configured to always start when powered up. If a start or stop signal is wanted this can be achieved by the interlock function on digital input 4 (DI4). This works also if a network protocol is set.

The following description is covering all digital input connected to different kind of functions. The In position is triggered on the event that a puck has entered the gap in the guide disc and the function is ready to start its movement with the puck. The puck entering sensor is triggered by the first puck in the queue waiting to be moved by the guide disc. These sensors are replaced by signals from the PLC if RFID is used.

#### Diverter

DI1: Puck entering  
DI2: Diverter selection (except variant 1/1)  
DI3: In position  
DI4: Interlock

#### Merger

DI1: Puck entering on conveyor 1 (only without network communication)  
DI2: Puck entering on conveyor 2 (only without network communication)  
DI3: In position, conveyor 1  
DI4: In position, conveyor 2

#### Combined diverter/merger

DI1: Puck entering on conveyor 1 (only without network communication)

DI2: Puck entering on conveyor 2 (on parallel version, only without network communication)

DI2: Puck entering on conveyor 3 (on 90° version, only without network communication)

DI3: In position, conveyor 1 (only without network communication)

DI4: In position, conveyor 2 (on parallel version, only without network communication)

DI4: In position, conveyor 3 (on 90° version, only without network communication)

#### Transfer

DI1: In position, conveyor 1

DI2:

DI3:

DI4: Interlock

#### Locating unit

DI1: In position, conveyor 1

DI2: Release from locating (only variant sensor)

DI3:

DI4: Interlock

#### Stop unit

DI1: In position, conveyor 1

DI2:

DI3:

DI4: Interlock

#### Speed mode

All sensor are used exactly like a drive motor in stand-alone mode (read chapter above).

## ***Line control***

The main purpose with the line control system is to send commands to the motors and receive status signals. If RFID are used the status of each RFID has to be read in order to send correct commands. Other tasks could be emergency/safety stop and HMI functionalities and communication with superior systems.

Some system can be designed without a line controller. These motor has to be parameterized to stand alone mode and will not communicate with anything except via digital inputs. The start/stop is following the supplied power.

## **Route handling**

All the logic regarding reading and write RFID has to be dealt with in the line controller. This identification results in a route handling that can be achieved by the line controller sends some commands to the motors via the external network.

## **Alarm handling**

The line controller can read out all alarms from the motors.  
See chapter *External communication / Cyclic data / Data3: Error* for more info.  
It is also possible to send a reset command to the motor units.

## **Safety functions**

Because of the layout with two different power buses (logic and drive), there are possible to communicate with the nodes even under an emergency stop. This safety function has to be dealt with in the line controller.

## **Line start/stop**

The overall start and stop of the line is of course something that has to be controlled from the line controller.

## External communication

The external communication is the data transfer between the motors and the line controller. It uses two M12 connectors on each motor in order to interlink the units (daisy chain). These connectors are marked ETH1 and ETH2 on the motor label. The communication can be divided into two categories. First there is cyclic data. This data is sent frequent and contains all basic data needed for external control. The other kind of communication is acyclic data. This is not sent frequently but on request. These two data transfer types have different names depending on which protocol is used.

### Cyclic data

This data is time critical low volume data (less than 8 byte).

Any parameter from the acyclic messaging can be sent in the cyclic message.

The Parameter ID is specified in Data 3 and the actual value should be stored in Data 4 to 7.

This ID is always the instance number in the Ethernet/IP RR Data table even for PROFINET.

#### Line controller -> Motor

Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7
Control Byte 0	Control Byte 1	Spare	Parameter ID	Parameter value, Byte 0	Parameter value, Byte 1	Parameter value, Byte 2	Parameter value, Byte 3

#### Data 0: Control Byte 0

Bit 7: not used

Bit 6: Reset alarm

Bit 5: Enable

Bit 4: Reversed speed. Only for conveyor motors

Bit 3: Digital Output 4

Bit 2: Digital Output 3

Bit 1: Digital Output 2

Bit 0: Digital Output 1

#### Data 1: Control Byte 1

Bit 7: not used

Bit 6: Control bit 1 = Move to Wait angle 2

Bit 5: Control bit 0 = Move to Wait angle 1

Bit 4: Control bit 4 = Receive puck from conveyor 3

Bit 3: Control bit 3 = Receive puck from conveyor 2

Bit 2: Control bit 2 = Receive puck from conveyor 1

Bit 1: Control bit 1 = Release the puck to conveyor 2

Bit 0: Control bit 0 = Release the puck to conveyor 1

#### Data 2: Reserved for future use

**Data 3: Parameter ID** (instance number from the Ethernet/IP table, even for PN)

**Data 4: Parameter value, Byte 0**

**Data 5: Parameter value, Byte 1**

**Data 6: Parameter value, Byte 2**

**Data 7: Parameter value, Byte 3**

### Motor -> Line controller

Data0	Data1	Data2	Data3	Data4
Status	Confirmation Byte 1	Confirmation Byte 2	Error	Spare

#### Data 0: Status

Bit 7: Digital input 8  
 Bit 6: Digital input 7  
 Bit 5: Digital input 6  
 Bit 4: Digital input 5  
 Bit 3: Digital input 4  
 Bit 2: Digital input 3  
 Bit 1: Digital input 2  
 Bit 0: Digital input 1

#### Data 1: Confirmation Byte

Bit 7: Clear error (acknowledge)  
 Bit 6: Enable drive (ETH)  
 Bit 5: Reverse direction (M1)  
 Bit 4: In position  
 Bit 3: State wait  
 Bit 2: State bit 3  
 Bit 1: State bit 2  
 Bit 0: State bit 1

State:	Function unit	Drive unit
0	Rescue	Rescue
1	Init	Running
2	Pre turn	Stopped
3	During turn	Fault
4	Post turn	
5	Calibrate	
6	Idle	
7	Fault	

Table: State bits

#### Data 2: Confirmation Byte

Bit 7: Not used  
 Bit 6: Wait angle 2  
 Bit 5: Wait angle 1  
 Bit 4: Receive puck from conveyor 3  
 Bit 3: Receive puck from conveyor 2  
 Bit 2: Receive puck from conveyor 1  
 Bit 1: Release the puck to conveyor 2  
 Bit 0: Release the puck to conveyor 1

#### Data 3: Error

Bit 7: Not used  
 Bit 6: Error bit 6 – Output overload  
 Bit 5: Error bit 5 – internal fault  
 Bit 4: Error bit 4 – Locked rotor  
 Bit 3: Error bit 3 – PCB over temp  
 Bit 2: Error bit 2 – Low UDC  
 Bit 1: Error bit 1 – High UDC  
 Bit 0: Error bit 0 – Over current

#### **Data 4: Spare**

Bit 7: Not used  
Bit 6: Not used  
Bit 5: Not used  
Bit 4: Not used  
Bit 3 = Not used  
Bit 2 = Not used  
Bit 1 = Not used  
Bit 0 = Not used

#### **Hand shake communication**

When using the control and confirmation bits it is recommended use the following guidelines.

These requirements have to be fulfilled before a new command is sent to the unit:

Step = 6 (Idle): Divider/Combiner

Step = 3 (During turn): Locating unit

No old commands are active (either receive or release)

Unique evaluation for the application (RFID, sensor, ...)

Then new commands can be sent to the unit. If both receive and release commands are used both can be sent simultaneously. These commands are individually reset by the confirmation bits from the motor.

### **Acyclic data**

Acyclic data are for changing parameters that are not sent frequently over the communication bus. A list of these parameters can be found under Software/Motor software/Motor parameters.

### **Network trouble shooting**

The Parameter Setting Tool software has also some diagnostic features that can be helpful. More info on this issue can be found in the separate user documentation regarding this software (5113270 in the Technical library at [www.flexlink.com](http://www.flexlink.com)).

## ***PROFINET***

### **GSDML file**

A GSDML can be downloaded from the X45e software download homepage at [www.flexlink.com](http://www.flexlink.com).

### **Process Data Exchange**

See chapter of Cyclic data in order to see detailed information about the communication

## Record Request Data

To read and write parameters use always:

- API: 0
- Slot: 1
- Sub slot: 1

GSDML file contains only index 1 to 5. This parameters is intended for PLC use. Other parameters should be used for configuration with PST or similar.

Index	Size	Offset (byte)	Min	Max	Scale	Unit	Parameter
1	S16	0	-400	400	1	rpm	Speed set point 1
1	S16	2	-400	400	1	rpm	Speed set point 2
1	U16	4	0	100	1	%	Torque
1	U16	6	1	300	1	cHz/s	Acceleration
1	U16	8	1	300	1	cHz/s	Deceleration
2	U16	0	-360	360	1	Deg	Receive angle 1
2	U16	2	-360	360	1	Deg	Receive angle 2
2	U16	4	-360	360	1	Deg	Receive angle 3
2	U16	6	-360	360	1	Deg	Release angle 1
2	U16	8	-360	360	1	Deg	Release angle 2
2	U16	10	-360	360	1	Deg	Wait angle 1
2	U16	12	-360	360	1	Deg	Wait angle 2
3	U16	0	0	65000	1	ms	Pre turn delay
3	U16	2	0	65000	1	ms	During turn delay
3	U16	4	0	65000	1	ms	Post turn delay
3	U16	6	0	65000	1	ms	Interlock delay on
3	U16	8	0	65000	1	ms	Interlock delay off
3	U16	10	0	65000	1	ms	Limit Switch wait
4	S16	0	-360	360	1	Deg	Position request absolute closest
4	S16	2	-360	360	1	Deg	Position request absolute CCW
4	S16	4	-360	360	1	Deg	Position request absolute CW
4	S16	6	-360	360	1	Deg	Position request relative
5	U8	0	0	255	1	y	Operating Years
5	U16	1	0	365	1	d	Operating Days
5	U8	3	0	23	1	h	Operating Hours
5	U8	4	0	59	1	m	Operating Minutes
5	U8	5	0	59	1	s	Operating Seconds
5	U16	6	0	65535	1		Operating Cycles high
5	U16	8	0	65535	1		Operating Cycles low
5	U16	10	0	65535	1	°C	Operating Temperature
5	U16	12	0	360	1	Deg	Operating Current angle CCW
5	U16	14	0	360	1	Deg	Operating Current angle CW
5	U16	16	0	65535	0.01	W	Operating Motor power
5	U16	18	0	65535	1	mA	Operating Motor current
5	U16	20	0	65535	0.01	V	Operating Motor voltage
10	U8	0	0	255	1	y	Operating Years
11	U16	0	0	365	1	d	Operating Days
12	U8	0	0	23	1	h	Operating Hours
13	U8	0	0	59	1	m	Operating Minutes
14	U8	0	0	59	1	s	Operating Seconds
15	U16	0	0	65535	1		Operating Cycles high
16	U16	0	0	65535	1		Operating Cycles low
17	S16	0	0	65535	1	°C	Operating Temperature
18	U16	0	0	360	1	Deg	Operating Current angle CCW
19	U16	0	0	360	1	Deg	Operating Current angle CW

20	U16	0	0	65535	0.01	W	Operating Motor power
21	U16	0	0	65535	1	mA	Operating Motor current
22	U16	0	0	65535	0.01	V	Operating Motor voltage
23	U8	0	1	2			Motor type
24	U8	0	1	6			Function mode
25	U16	0	1	65535			Calibrate
26	U8	0	0	1			Enable drive
27	U8	0	0	1			Reset errors
28	U8	0	0	3, 128			Debug mode
29	U8	0	1	20			Function parameter 1
30	U8	0	1	20			Function parameter 2
31	U8	0	1	20			Function parameter 3
32	U8	0	1	20			Function parameter 4
33	S16	0	-360	360	1	Deg	Request angle
34	U8	0	0	1			Automatic reset errors
35	U8	0	0	1			Interlock mode
36	U8	0	0	1			Enable drive reset
37	U16	0					FEG program major version
38	U8	0	0	4			Network protocol
39	U8	0	1	2			Positioner mode
40	U16	0	1	65535	1	mm/r	Positioner ratio
41	U16	0	1	65535			Function group
42	U16	0	1	65535			Function variant
43	U16	0	1	65535			ECO program version
44	U8	0	0	1			Debug enable
45	U8	0	0	1			Initialize to default
46	U8	0	0	1			IP handling automatic
47	U32	0					IP address
48	U32	0					IP subnet mask
49	U32	0					IP gateway
50	U32	0					IP DNS server
51	U16	0					FEG program minor version
52	U16	0					FEG program patch version
53	S16	0	-400	400	1	rpm	Speed set point 1
54	S16	0	-400	400	1	rpm	Speed set point 2
55	U16	0	0	100	1	%	Torque
56	U16	0	1	300	1	cHz/s	Acceleration
57	U16	0	1	300	1	cHz/s	Deceleration
58	U16	0	-360	360	1	Deg	Receive angle 1
59	U16	0	-360	360	1	Deg	Receive angle 2
60	U16	0	-360	360	1	Deg	Receive angle 3
61	U16	0	-360	360	1	Deg	Release angle 1
62	U16	0	-360	360	1	Deg	Release angle 2
63	U16	0	-360	360	1	Deg	Wait angle 1
64	U16	0	-360	360	1	Deg	Wait angle 2
65	U16	0	0	65000	1	ms	Pre turn delay
66	U16	0	0	65000	1	ms	During turn delay
67	U16	0	0	65000	1	ms	Post turn delay
68	U16	0	0	65000	1	ms	Interlock delay on
69	U16	0	0	65000	1	ms	Interlock delay off
70	U16	0	0	65000	1	ms	Limit Switch wait
71	S16	0	-360	360	1	Deg	Position request absolute closest
72	S16	0	-360	360	1	Deg	Position request absolute CCW
73	S16	0	-360	360	1	Deg	Position request absolute CW
74	S16	0	-360	360	1	Deg	Position request relative
75	U8	0	1	127			Node ID
76	U8	0	0	2			RabbitMQ mode
77	Char[32]	0					RabbitMQ server
78	U16	0					RabbitMQ port

79	Char[32]	0					RabbitMQ exchange
80	Char[32]	0					RabbitMQ device name
81	U16	0	0	65000	1	s	RabbitMQ runtime send frequency
82	U8	0	0	100	1	%	RabbitMQ power debounce

*Table: PROFINET, Record Request Data table*

## ***Ethernet/IP***

### **EDS file**

An Ethernet/IP EDS file can be downloaded from the X45e software download homepage at [www.flexlink.com](http://www.flexlink.com).

### **I/O messages**

See chapter of Cyclic data in order to see detailed information about the communication

## RR Data

Class	Instance	Size	R/W	Min	Max	Scale	Unit	Name in PST
\$F	0	U8	R	1	255			
\$F	1	U8	WR	1	2			Motor type
\$F	3	U8	WR	1	6			Function mode
\$F	5	BOOL	WR	1	65535			Calibrate
\$F	6	U8	WR	1	127			NodeID
\$F	7	S16	WR	-400	400	1	rpm	Speed set point 2
\$F	9	S16	WR	-400	400	1	rpm	Speed set point 1
\$F	10	U16	WR	0	100	1	%	Torque
\$F	11	S16	WR	-360	360	1	Deg	Receive angle 1
\$F	12	S16	WR	-360	360	1	Deg	Receive angle 2
\$F	13	S16	WR	-360	360	1	Deg	Release angle 1
\$F	14	S16	WR	-360	360	1	Deg	Release angle 2
\$F	30	S16	WR	-360	360	1	Deg	Receive angle 3
\$F	15	S16	WR	-360	360	1	Deg	Wait angle 1
\$F	16	S16	WR	-360	360	1	Deg	Wait angle 2
\$F	17	U16	WR	1	300	1	cHz/s	Acceleration ramp
\$F	18	U16	WR	1	300	1	cHz/s	Deceleration ramp
\$F	19	BOOL	WR	0	1			Enable drive
\$F	20	BOOL	WR	0	1			Reset errors
\$F	22	U8	WR	0	3,128			Mode
\$F	23	U8	WR	1	20			Function parameter 1
\$F	24	U8	WR	1	20			Function parameter 2
\$F	25	U8	WR	1	20			Function parameter 3
\$F	26	U8	WR	1	20			Function parameter 4
\$F	27	S16	WR	-360	360	1	Deg	Request angle
\$F	28	U16	WR	0	65000	1	ms	During turn delay
\$F	29	U16	WR	0	65000	1	ms	Post turn delay
\$F	33	U16	WR	0	65000	1	ms	Interlock delay on
\$F	34	U16	WR	0	65000	1	ms	Interlock delay off
\$F	36	U8	WR	0	1			Init to default
\$F	47	U16	WR	0	65000	1	ms	Pre turn delay
\$F	31	U8	WR	0	1			Automatic reset errors
\$F	32	U8	WR	0	1			Interlock mode
\$F	35	U8	WR	0	1			A-Bus enable and reset
\$F	38	U16	R					FEG Program version
\$F	37	U8	WR	0	4			A-Bus network protocol
\$F	39	U16	R	0	65535	0.01	W	Motor power consumption
\$F	44	U8	WR	0	255	1	y	Operating years
\$F	43	U16	WR	0	365	1	d	Operating days
\$F	42	U8	WR	0	23	1	h	Operating hours
\$F	41	U8	WR	0	59	1	m	Operating minutes
\$F	40	U8	WR	0	59	1	s	Operating seconds
\$F	45	U16	WR	0	65535	1		Op. cycles (High word)
\$F	46	U16	WR	0	65535	1		Op. cycles (Low word)
\$F	48	U8	WR	1	2			Positioner Mode
\$F	49	U16	WR	1	65535	1	mm/r	Positioner Ratio
\$F	50	U16	WR	1	65535			Function group
\$F	51	U16	WR	1	65535			Function variant
\$F	52	U16	R	1	65535			ECO Program version
\$F	53	U16	R	1	65535	1	°C	Temperature
\$F	54	U8	RW	0	1			Enable debug signals
\$F	56	S16	RW	0	360	1	Deg	Position abs closest
\$F	57	S16	RW	0	360	1	Deg	Position abs CCW
\$F	58	S16	RW	0	360	1	Deg	Position abs CW
\$F	59	S16	RW	-360	360	1	Deg	Position relative
\$F	60	U16	R	0	360	1	Deg	Position actual, CCW

\$F	61	U16	R	0	360	1	Deg	Position actual, CW
\$F	62	U8	RW	0	1			IP handling automatic
\$F	63	U32	RW					IP address
\$F	64	U32	RW					IP subnet mask
\$F	65	U32	RW					IP gateway
\$F	66	U32	RW					IP DNS server
\$F	67	U16	R			1	mA	Operating motor current
\$F	68	U16	R			0.01	V	Operating motor voltage
\$F	69	U16	R					FEG minor version
\$F	70	U16	R					FEG patch version
\$F	71	U8	RW	0	2			RabbitMQ mode
\$F	72	Char[32]	RW					RabbitMQ server
\$F	73	U16	RW					RabbitMQ port
\$F	74	Char[32]	RW					RabbitMQ exchange
\$F	75	Char[32]	RW					RabbitMQ device name
\$F	76	U16	RW	0	65000	1	s	RabbitMQ runtime send frequency
\$F	77	U8	RW	0	100	1	%	RabbitMQ power debounce

Table: Ethernet/IP RR Data table

## Appendix

### *Speed parameter settings*

Conveyor velocity (m/min)	Speed parameter Conveyor motor (rpm)	Speed parameter Function motor (rpm)	Release puck delay Time B (ms)
5	92	27	350
6	110	32	
7	122	37	
8	138	42	200
9	153	47	
10	168	52	150
11		57	
12	200	62	
13		67	
14		72	
15	250	77	75
16		82	
17		87	
18		92	
19		97	
20	326	102	

### Cycle time (Function units)

