ZX09-Series
ZX20/ZX21-Series

ZANDER AACHEN

Operating Manual

FPGA-based High-Speed Industrial PLC

Zander GmbH & Co. KG
Am Gut Wolf 15
52070 Aachen, Germany
info@zander-aachen.de
www.zander-aachen.de

Item no.: E61-436-00
Version: P01

This document is the English translation.

Technical Changes reserved,
all information is subject to change.

1. About this document 7
1.1 Validity 7
1.2 Target group 7
1.3 Explanation of symbols 7
2. General safety instructions 8
3. About this device 9
3.1 Product information 9
3.2 Delivery content and supplies 10
3.3 Intended usage 11
3.4 Exclusion of liability and warranty 11
3.5 Function 11
3.6 Device diagram 12
3.6.1 ZX09 12
3.6.2 ZX09A 12
3.6.3 ZX09D / ZX09E 13
3.6.4 ZX20T 13
3.6.5 ZX21TP/TC 14
3.6.6 ZX21TPA/TCA 14
4. Assembly 15
4.1 Assembly on the mounting rail 15
4.2 Disassembly off the mounting rail 16
5. Electrical connection 17
5.1 General rules for the electrical connection 17
5.2 Connection terminals 18
5.2.1 ZX09 / ZX09A / ZX09D / ZX09E 18
5.2.2 ZX20T, ZX21TP/TC, ZX21TPA/TCA 20
5.3 Failure protection 22
5.4 Electrical connection of the device 22
5.5 Inputs 23
5.5.1 Digital Inputs 24
5.5.2 Connection of the digital Inputs 25
5.5.3 Analog Inputs (only ZX09A/D/E and ZX21TPA/TCA) 28

5.5.4 Connection of the analog inputs 29
5.5.5 The SSI-Interface (only ZX09A, ZX09D and ZX09E) 30
5.5.6 Connection of an Encoder to the SSI-Interface 31
5.5.7 TTL differential signal inputs (only ZX09A/D/E and ZX21TPA/TCA) 32
5.5.8 TTL differential signal outputs (only ZX09A/D/E and ZX21TPA/TCA) 32
5.5.9 Connection of the In-/Outputs for the TTL differential signals 32
5.6. Digital outputs 33
5.6.1 Connection of the digital outputs 34
6. Programming 36
7. Networking 37
7.1 Networking via Ethernet 37
7.1.1 Modbus/TCP 37
7.1.2 ZanderNet 38
7.2 The RS 485-Interface (only ZX09A/D/E and ZX21TPA/TCA) 41
7.2.1 Modbus RTU 41
7.2.2 Modbus RTU wiring 42
7.2.3 ZanderLink 43
7.2.4 ZanderLink wiring 44
7.3 Networking via Profinet (only ZX21TP/TPA) 45
7.4 Networking via EtherCAT (only ZX21TC/TCA) 46
8. Commissioning 47
8.1 What happens at the start? 47
8.2 Initial commissioning 48
8.2.1 Step 1: Program creation in EX_PRESS 5 48
8.2.2 Step 2: Connecting the power supply 48
8.2.3 Step 3: Programming the PLC 48
8.2.4 Step 4: Connecting the inputs 48
8.2.5 Step 5: Connecting the encoder (ZX09A/D/E, ZX21TPA/TCA) 49
8.2.6 Step 6: Controlling the function of the PLC via the LEDs 49
8.2.7 Step 7: Connection to a network 49
8.2.8 Step 8: Connecting the outputs 50
8.2.9 Step 9: Connecting the power supply of the digital outputs 50

9. Diagnosis 50
10. Maintenance, repair and replacement 54
11. Dimensional drawing 55
12. Technical data 56
12.1 ZX09 / ZX09A 56
12.2 ZX09D / ZX09E 58
12.3 ZX20T, ZX21TP/TC 60
12.4 ZX21TPA/TCA 61
12.5 FPGA logic capacities 63
13. Example 1 64
13.1 The process 64
13.2 Step 1: Program creation in EX_PRESS 5 65
13.3 Step 2: Connecting the power supply 69
13.4 Step 3: Programming the PLC 69
13.5 Step 4: Connecting the inputs 69
13.6 Step 6: Controlling the function of the PLC via the LEDs 79
13.7 Step 8: Connecting the digital outputs 70
13.8 Step 9: Connecting the power supply of the digital outputs 71
14. Example 2 72
14.1 The process 72
14.2 Step 1: Program creation in EX_PRESS 5 73
14.3 Step 2: Connecting the power supply 76
14.4 Step 3: Programming the PLC 76
14.5 Step 4: Connecting the inputs 76
14.6 Step 6: Controlling the function of the PLC via the LEDs 77
14.7 Step 8: Connecting the digital outputs 77
14.8 Step 9: Connecting the power supply of the digital outputs 78
15. Example 3 79
15.1 The Application 79
15.2 Step 1: Program creation in EX_PRESS 5 80
15.3 Step 2: Connecting the power supply 83

15.4 Step 3: Programming the PLC 83
15.5 Step 4: Connecting the inputs 83
15.5 Step 7: Connecting the Ethernet interface.....84
16. Service 85

1. About this Document

1.1 Validity

This document ist valid for the following programmable logic controllers:
ZX09 (Item No.: 589200), ZX09A (Item No.: 589201), ZX09D (Item No.: 589204), ZX09E (Item No.: 589205), ZX20T (Item No.: 589000), ZX21TP (Item No.: 589052), ZX21TC (Item No.: 589053), ZX21TPA (Item No.: 589040), ZX21TCA (Item No.: 589041).

1.2 Target group

This document is intended for electricians, installation-, commissioning– and service personnel with appropriate knowledge in handling automation technology components.

1.3 Explanation of symbols

Symbol / Illustration	Meaning
	Document in printed form
	Document is available to download at www.zander-aachen.de
	Document on USB memory stick
	Safety instruction **Warning:** Possible dangers **Caution:** Possibility of damage to life and property
	Important information
HINT	Tip / useful information

2. General safety instructions



- Only authorised and qualified personnel may install the device and put it into operation, i.e. personnel which
  - is well-acquainted with correct technical handling of the electrical machine equipment,
  - is familiar with the applicable regulations concerning occupational safety and accident prevention,
  - has read and understood the operating manual and, if applicable, the programming manual
- In case of incorrect connection or inappropriate use, the safe functioning of the device during the machine operation will no longer be guaranteed. This can result in fatal injuries or serious material loss.
- Please observe the national guidelines, which are valid for the commissioning of the device.
- The electrical connection of the device must take place only when the voltage supply is switched off. This also applies to the connected sensors and actuators.
- The wiring of the device must be done in conformance with the instructions given in this operating manual.
- The person, who installs the device and puts it into operation, must be adequately protected against electrostatic charge (earthing strap or ESD shoes along with ESD flooring).
- Opening of the device and any manipulations to the device are not allowed; and shall result in loss of warranty and warranty claims.
- Please observe all the relevant safety guidelines and standards.
- Non-observance of the safety guidelines may lead to death, severe injuries and serious material damage.
- Please read the operating manual before use and preserve it carefully.
- Please ensure that the operating manual is always available at the time of assembly, commissioning and maintenance work.

3. About this device

3.1 Product information

The ZX09– and ZX20/21-series are high-speed industrial programmable logic controllers (PLCs). They can be used as stand-alone PLCs or in a network as decentralized PLCs for parallel processing of signals in real time. This is made possible by the heart of every ZX-control: A Field Programmable Gate Array (FPGA). This enables the fastest possible and equally fast, reproducible processes without cycle time in absolute parallelism.

The following table lists the type and number of inputs, outputs and communication interfaces for all device types:

		ZX09	ZX09A	ZX09D	ZX09E	ZX20T	ZX21TC	ZX21TP	ZX21TCA	ZX21TPA
Digital inputs		10	10	2	2	20	20	20	8	8
Digital outputs		4	4	4	4	16	16	16	4	4
Analog inputs 0..10V/4..20mA switchable		-	1	2	2	-	-	-	2	2
Analog inputs 0..10V		-	-	4	-	-	-	-	2	2
Analog inputs 4..20mA		-	-	-	4	-	-	-	-	-
Analog outputs 0..10V		-	-	-	-	-	-	-	4	4
RS-485 Ports: programmable alternatives	SSI	-	2	2	2	-	-	-	1	1
	Modbus RTU	-	2	2	2	-	-	-	-	-
	ZanderLink	-	2	2	2	-	-	-	1	1
	TTL differential inputs/outputs	-	4	4	4	-	-	-	2	2
Ethernet (Modbus/TCP, ZanderNet)		1	1	1	1	1	1	1	1	1
PROFINET		-	-	-	-	-	-	2 (Switch)	-	2 (Switch)
EtherCAT		-	-	-	-	-	2 (In/Out)	-	2 (In/Out)	-

The programming is carried out via the programming software EX_PRESS 5 (Item No.: 589092) in structured text (ST) after IEC 61131-3.

3.2 Delivery content and supplies

Delivery content

- One of the following PLCs: ZX09 (Item No.: 589200), ZX09A (Item No.: 589201), ZX09D (Item No.: 589204), ZX09E (Item No.: 589205), ZX20T (Item No.: 589000), ZX21TP (Item No.: 589052), ZX21TC (Item No.: 589053), ZX21TPA (Item No. 589040) or ZX21TCA (Item No. 589041)
- Pluggable screw terminals (plugged into the device when delivered):
  - ZX09: 1 x 10-pin and 2 x 4-pin
  - ZX09A: 2 x 10-pin and 2 x 4-pin
  - ZX09D/E: 1 x 10-pin and 5 x 4-pin
  - ZX20T, ZX21TP/TC: 4 x 10-pin
  - ZX21TPA/TCA: 2 x 8-pin, 5 x 4-pin
- This operating manual



Note!
For the program creation and programming of the High-Speed-PLC you need the software package EX_PRESS 5 in the latest version, which is separately available.

Required accessories

Designation	Execution	Item No.:
EX_PRESS 5	USB memory stick	589092

Optional accessories

Designation	Item No.:
Set of pluggable terminals for ZX09 with lateral cable outlets	589270
Set of pluggable terminals for ZX09A with lateral cable outlets	589271
Set of pluggable terminals for ZX09D/E with lateral cable outlets	589273
Set of pluggable terminals for ZX20T with lateral cable outlets	589070

3.3 Intended usage

The ZX09- und ZX20/21-Series have been designed for the use in machines and plants for industrial use. They must be installed in switch cabinets or appropriate housings with minimum degree of protection IP54. The approved operating parameters for use must be complied with (see chapter 12 “Technical data”).
The inputs and outputs have to be connected accordingly.
The controls can be integrated as slaves into an existing Modbus TCP/IP network or build an existing network with the Multimaster protocol “ZanderNet“ via the Ethernet-interface.

The controller ZX21TP/ZX21TPA can be integrated as slave (PROFINET IO-Device) into a PROFINET-network. The controller ZX21TC/ZX21TPA can be addressed as slave in an EtherCat-network.

3.4 Exclusion of liability and warranty

If the conditions above are not complied for the intended use, the safety instructions are not followed, or maintenance work is not performed as required, it will lead to a disclaimer or loss of warranty.

	**Note!** The user is responsible for the integration of the device into the overall system. For this, the correct electrical installation and the correct programming of the ZX09, ZX09A, ZX09D, ZX09E, ZX20T, ZX21TP/TC or ZX21TPA/TCA must be verified.

3.5 Function

The function of the device depends upon the loaded user program. The logic stored in this program determines how the input signals are evaluated and the outputs switch.

3.6 Device Diagram

3.6.1 ZX09

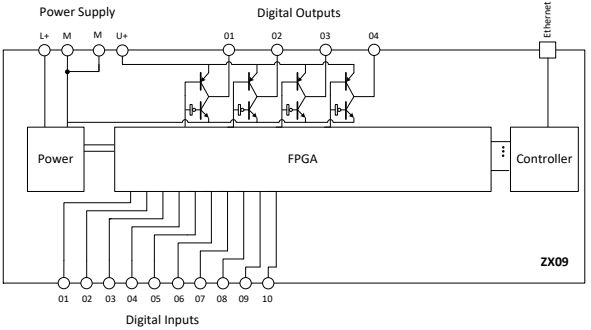


Fig. 1: Device diagram ZX09

3.6.2 ZX09A

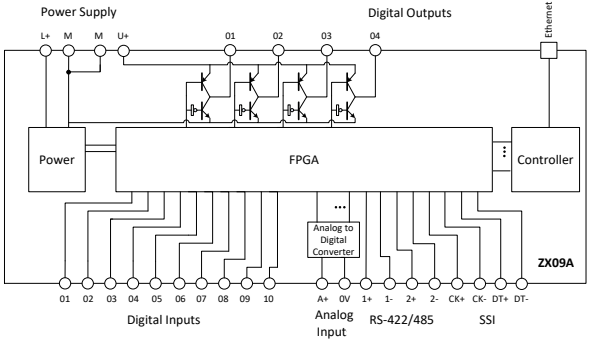


Fig. 2: Device diagram ZX09A

3.6.3 ZX09D / ZX09E

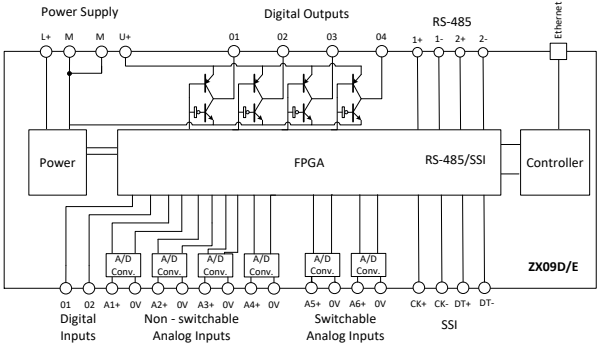


Fig. 3: Device diagram ZX09D / ZX09E

3.6.4 ZX20T

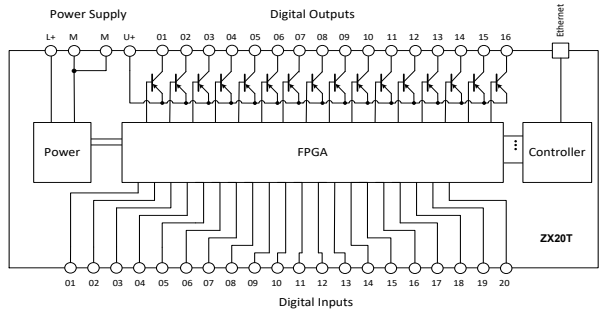


Fig. 4: Device diagram ZX20T

3.6.5 ZX21TP / ZX21TC

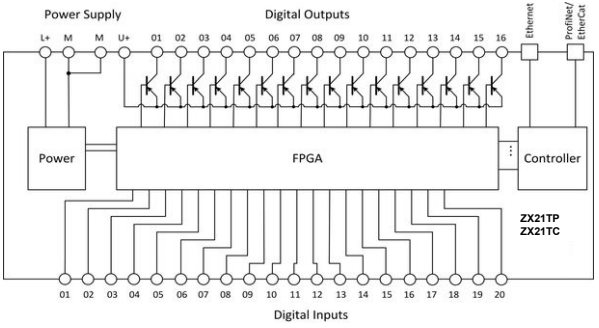


Fig. 5: Device diagram ZX21TP / ZX21TC

3.6.6 ZX21TPA / ZX21TCA

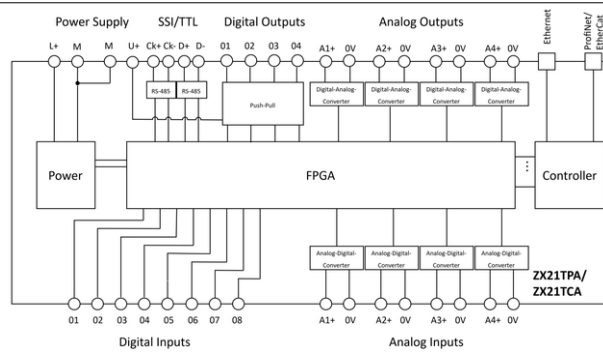


Fig. 5a: Device diagram ZX21TPA / ZX21TCA

4. Assembly

Please note:
The device must be installed in a switch cabinet or suitable housing with a minimum degree of protection of IP54

- Mount the controller on a 35 mm mounting rail according to EN 60715
- Maintain a mounting distance of minimum 5 mm from the adjacent devices
- Ensure that there is sufficient heat dissipation in the switch cabinet.
- Keep the PLC away from devices or components, which carry high voltage or cause strong electrical interference.

4.1 Assembly on the support rail

The black underside of the device is placed flat on the mounting rail (see Fig. 6). Then press the unit firmly onto the mounting rail with your palms (see Fig. 7) until both orange locking slides, which are located at the bottom, in the middle, on the long outer sides and on the underside of the control unit, are engaged.

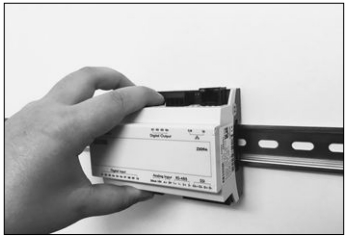


Fig. 6: Place device on top hat rail

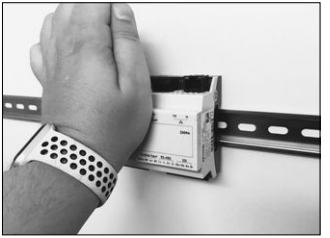


Fig. 7: Press device onto top-hat rail

4.2 Disassembly off the mounting rail

The two orange locking slides (top and bottom) are pulled out one after the other with a screwdriver (blade width max. 3.5 mm) (see Fig. 8). The device is then released from the top-hat rail and can be removed.



Fig. 8. Pull the catch to the outside

5. Electrical connection

5.1 General rules for the electrical connection

Warning

- The operating voltage (DC 24V) must correspond to the specifications given in chapter 12 "Technical data".
- The voltages at the digital inputs must correspond to the specifications given in chapter 12 "Technical Data".
- The voltages and currents at the analog input must correspond to the specification given in chapter 12 "Technical Data".
- The voltage applied to the voltage input of the digital outputs ("U+") must correspond to the specifications given in chapter 12 "Technical Data".
- Do not connect external voltage to the digital outputs.
- The supply voltage must be switched off during wiring. The supply voltages of all sensors and actuators must also be switched off.

Non-observance of these instructions can result in the destruction of electronic components within the device! In this case, warranty or guarantee claims are excluded.

Please also note:

- All electrical connections must be isolated from the mains either by safety transformers (SELV/PELV) according to IEC 61558-2-6 with limitation of the output voltage in the event of a fault or by equivalent isolation measures.
- All outputs must have sufficient protective circuits for inductive loads. For this purpose, the outputs must be protected with freewheeling diodes or varistors.
- Do not lay the signal and communication lines of the digital inputs in the same cable tray as AC supply voltage lines or lines with high interference levels. Keep a minimum distance of 20 cm to such cable trays.
- It is recommended to use shielded network cables of category Cat. 7 for the network connection.
- For connection to the SSI or RS 485 interface (ZX09 series only) a shielded, twisted pair cable is required (see Fig. 15).
- Tighten the screws of the connection terminals to a maximum of 0.8 Nm.

5.2 Connection terminals

5.2.1 ZX09 / ZX09A / ZX09D / ZX09E

Terminal	Description
L+	Positive supply voltage U_B
M (2 equivalent terminals)	Connection 0 V of the supply voltage and the voltage for the digital outputs and 0 V potential for digital and analog inputs. M and 0V are connected internally
U+	Connection for the positive voltage switched by the digital outputs (can be different from operating voltage L+)
Digital Input 01..02 01..10	Digital inputs
Digital Output 01..04	Digital outputs
Analog Input A+ / 0V A_1+ / 0V ... A_6+ / 0V	Analog current or voltage inputs
	Ethernet interface
RS-485	Serial interface RS-485/422 or second SSI interface (only ZX09A, ZX09D, ZX09E)
SSI	SSI interface or second RS-485 interface (only ZX09A, ZX09D, ZX09E)

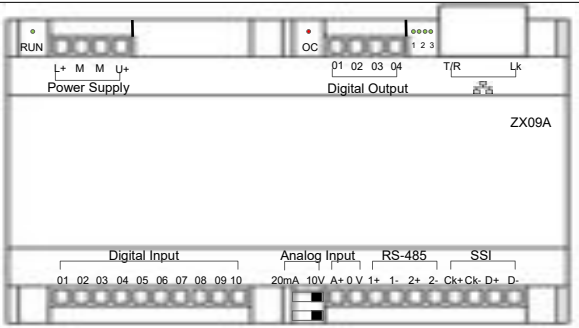


Fig. 9: Connection terminals ZX09A

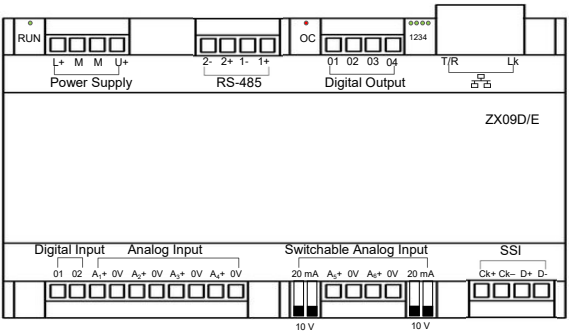


Fig. 10: Connection terminals ZX09D / ZX09E

5.2.2 ZX20T, ZX21TP/TC, ZX21TPA/TCA

Terminal	Description
L+	Positive supply voltage U_B
M (2 equivalent terminals)	0 V - Connection of the supply voltage and the voltage for the digital outputs and 0 V potential for digital inputs. M and 0 V are internally connected with each other
U+	Positive voltage for the dig. outputs (can be different to operating voltage L+)
Digital Input 01, 02, ...	Digital inputs
Digital Output 01, 02, ...	Digital outputs
	Ethernet interface
Analog Input A1+/0V, A2+/0V, ...	Analog inputs (only ZX21TPA/TCA)
Analog Output A1+/0V, A2+/0V, ...	Analog outputs (only ZX21TPA/TCA)
SSI/TTL	SSI interface, alternatively also usable as 2 TTL differential signal inputs or outputs or as ZanderLink interface (only ZX21TPA/TCA)
PROFINET IO-Device	PROFINET interface (only ZX21TP, ZX21TPA)
EtherCAT Slave	EtherCAT interface (only ZX21TC, ZX21TCA)

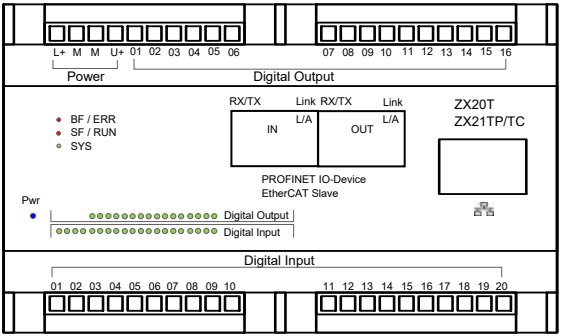


Fig. 11: Connection terminals ZX20T, ZX21TP/TC

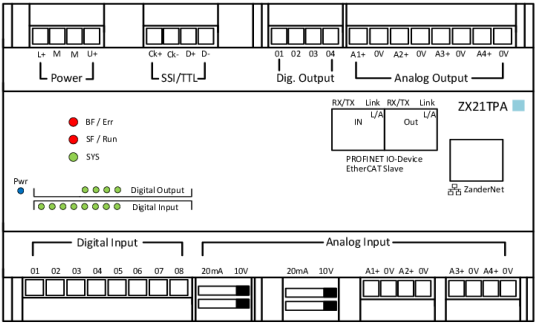


Fig. 11a: Connection terminals ZX21TPA/TCA

5.3 Failure protection

The operating voltage L+ / M is reverse polarity protected.
All digital outputs are short-circuit proof.

Warning:
The switching capacity and the switching capacity of the power supply unit specified in chapter 12 "Technical data" must be observed.
information for the digital outputs must be observed. In case of overload the output drivers can be destroyed.

5.4 Electrical connection of the device

The electrical connection of the device is shown in Figs. 12 and 13. The power supply is connected to the instrument via terminals L+ and M on the upper left terminal block. The max. permissible voltage on the mentioned terminal is approx. 27.5 V (24 V + 15%, see chapter 12 "Technical data"). For wiring, the positive voltage must be connected to terminal L+ and the corresponding 0 V potential to M. The PE potential of the voltage source must be grounded according to the requirements of the system. The controller itself must not be additionally earthed.

Warning:
Do **not** connect terminal M to the PE connection of the voltage source, as this can lead to massive EMC interference.

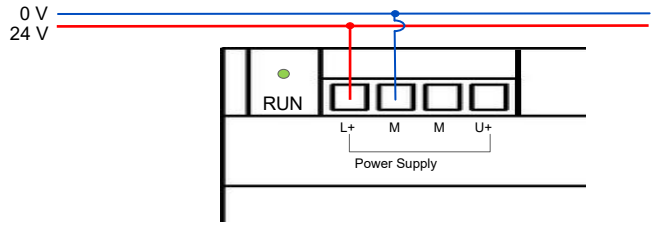


Fig. 12: Connection of the power supply of the ZX09/A/D/E

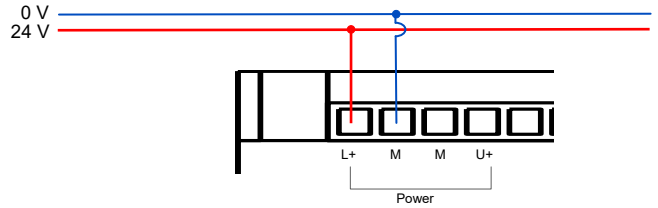


Fig. 13: Connection of the power supply of the ZX20T, ZX21TP/TC

5.5 Inputs

All device variants have digital inputs, and the ZX09A, ZX09D, ZX09E, ZX21TPA and ZX21TCA variants also have analog inputs. These can be connected to the desired sensor system via the lower terminal blocks "Digital Input" or "Analog Input".

If an input signal is applied to one of the inputs of a ZX20/21 series controller, this is indicated by a green LED on the PLC (see Figs. 14A and 14B). The LEDs "01", "02", "03", ... , "20" are continuously assigned to the digital inputs "01", "02", "03", ... , "20".



Note:

In the "Structured Text", input variables for digital inputs can be assigned to the above-mentioned connection terminals via the keyword "AT" using the designations "In_01", "In_02", ... , "In_10". Detailed information on this can be found in the programming manual in Chapter 2.2.5.1 "VAR_INPUT / VAR_OUTPUT", which is included as a PDF file in the software package "EX_PRESS 5" (Art. No. 589092) on the USB stick included in the package.

The ZX09 series does not have any LEDs to indicate the input states.

5.5.1 Digital Inputs

The ZX09 and ZX09A controllers provide ten digital inputs and the ZX09D/E controllers two. These are accessible via the connection terminals "Digital Input 01..10" or "Digital Input 01..02".

The ZX20T, ZX21TP and ZX21TC have 20 digital inputs. These are accessible via the lower terminal blocks "Digital Input 01..20".

The ZX21TPA and ZX21TCA have 8 digital inputs. These are accessible via the lower terminal blocks “Digital Input 01..08”.

The digital inputs are not galvanically isolated from the supply voltage, i.e. their 0 V reference potential is identical to the 0 V potential of the supply voltage (terminal M). For generating the input signals, a different voltage source than the operating voltage source can be used. If this is the case, the 0 V potential of this voltage source must be connected to the 0 V potential of the operating voltage with the lowest possible resistance. A SELV/PELV voltage source according to IEC 61558-2-6 with limitation of the output voltage in case of a fault must be used.

5.5.2 Connection of the digital Inputs

The connection of the digital inputs for the ZX09 series is shown in Figs. 15A and 15B using the ZX09 / ZX09A controller as an example. The connection of the digital inputs of the ZX09D/E controllers is identical, the only difference is the number of inputs.

The connection of the digital inputs of the ZX20/21 series controllers is shown in Figures 14A and 14B.

Note:
Please observe the permissible voltage range for the digital inputs (18 - 30 V).

Sensors which generate a digital signal can be connected to the terminals "Digital Input 01..10" or "Digital Input 01..02", here represented by simple switches. These can be supplied either by the operating voltage source or a separate voltage source. For the input voltage, a voltage range of 18 to 30 V applies (see chapter 12 "Technical data"). If a separate voltage source is used, its 0 V output must be connected to the 0 V output of the operating voltage. The signal cable of the sensors can be connected directly (see Fig. 14A or 15A) or, if fast input signals have to be processed, shielded (see Fig. 14B or 15B) to the input terminals. If a screened cable is used, the cable screen must be connected to a screen rail. The cable shield should be connected at one end as close as possible to the terminal.
It is possible to suppress inputs by software by providing them with a "debounce time" in the user program via the ".TDB" command. This is helpful when using electromechanical switching elements or input signals with very steep clock edges which cause overshooting at the input.

Note:

Detailed information on programming the debounce time can be found in the programming manual under chapter 2.3.1.1 "VAR_INPUT", which is included as a PDF file in the software package "EX_PRESS 5" (Art. No. 589092) on the USB stick enclosed there.

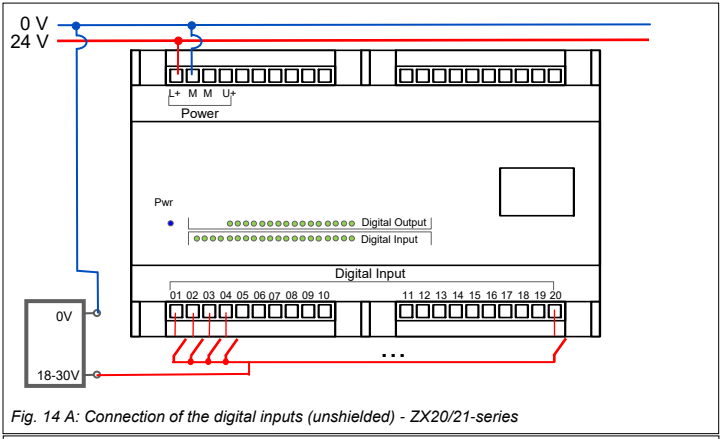


Fig. 14 A: Connection of the digital inputs (unshielded) - ZX20/21-series

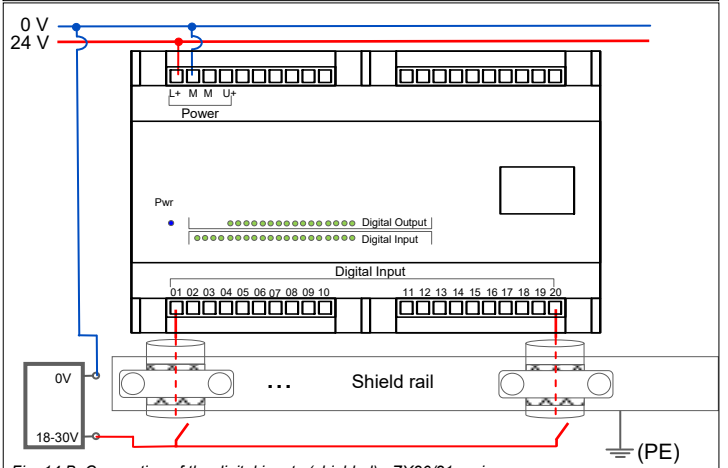


Fig. 14 B: Connection of the digital inputs (shielded) - ZX20/21-series

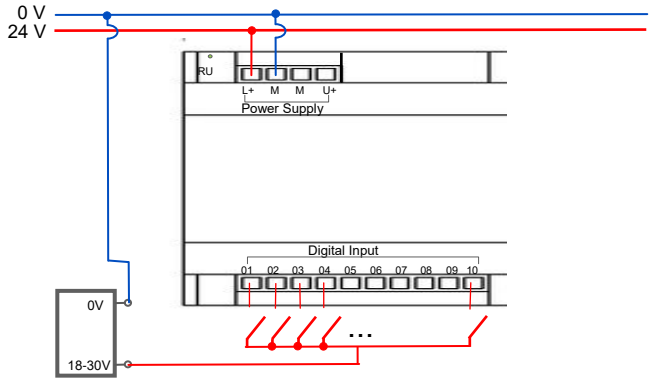


Fig. 15 A: Connection of the digital inputs (unshielded) - ZX09-series

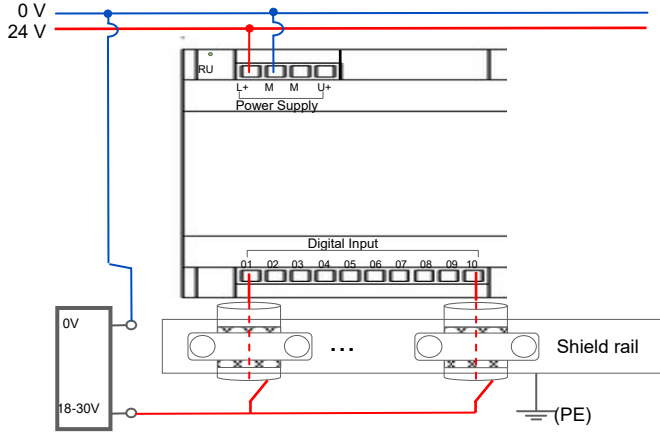


Fig. 15 B: Connection of the digital inputs (shielded) - ZX09-series

5.5.3 Analog inputs (only ZX09A/D/E and ZX21TPA/TCA)

The following device variants provide a different number of analog inputs:

Device version	Number (total)	Description
ZX09A	1	1 switchable input: 0..10 V or 4..20 mA
ZX09D	6	4 fixed inputs: 0..10 V 2 switchable inputs: 0..10 V or 4..20 mA
ZX09E	6	4 fixed inputs: 0..20 mA 2 switchable inputs: 0..10 V or 4..20 mA
ZX21TPA and ZX21TCA	4	2 fixed inputs: 0..10 V 2 switchable inputs: 0..10 V or 4..20 mA



Note:

In the "Structured Text" the input variable for the analog input can be declared under "VAR_ADC". Detailed information can be found in the programming manual under Chapter 2.2.6.3 "VAR_ADC", which is included as a PDF file in the software package EX_PRESS 5 (Art. No. 589092) on the USB stick included there.

The analog inputs are accessible via the terminals A1+ and 0V ... A6+ and 0V. Via the two switch positions 20mA and 10V on the controller (see Fig. 17), the switchable analog inputs can be configured as voltage or current inputs. For configuration, both switches must be moved to the corresponding position (position 10V for voltage measurement, position 20mA for current measurement). If the corresponding switches are in different positions when operating an analog input, the input may be destroyed. The analog inputs enable the detection of a current value from 0mA to 20mA or a voltage value from 0V to 10V by means of the corresponding switch position. The incoming current or voltage signal is connected to terminal A+ and the respective 0V potential to 0V.

5.5.4 Connection of the analog inputs

Fig. 16 shows an example of how to connect the analog input to a ZX09A controller for a voltage measurement. The signal generated by the sensor is connected to terminals A+ and 0V via a shielded cable. The supply line of the analog input must be made with a shielded cable. Connect the cable screen on one side as close as possible to the terminals of the control system with low impedance to a screen rail which is at PE potential (see fig.16). The cables must be laid as far away as possible from cables subject to interference. The connection of the analog inputs on the

device variants ZX09D, ZX09E, ZX21TPA and ZX21TCA is carried out according to these instructions. For ZX09D/E, terminals A1+ to A4+ are fixed inputs (ZX09D: voltage inputs; ZX09E: current inputs) and terminals A5+ and A6+ are inputs configurable as current or voltage inputs. For ZX21TPA/TCA, A1+ and A2+ are the inputs configurable as current or voltage inputs, and A3+ and A4+ are fixed voltage inputs.

Warning:
When using the analog input as current input, the applied current must not exceed 38 mA and no voltage source must be applied. Otherwise the analog input will be destroyed. For further information, please observe the permissible operating range for the analog inputs and the other specifications in chapter 12 "Technical Data".

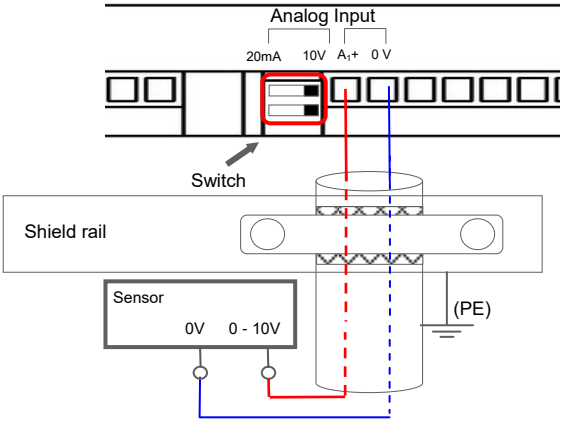


Fig. 16: Connection of a analog input (ZX09A)

5.5.5 The SSI-Interface (only ZX09A/D/E and ZX21TPA/TCA)

The controllers ZX09A/D/E have up to two SSI interfaces, the variants ZX21TPA/TCA provide one SSI interface. These are identified on the unit by the SSI or SSI/TTL and RS-485 terminals. An absolute encoder can be connected via this interface. Both "Single-Turn" and "Multi-Turn" versions are possible as well as the data formats "binary code" and "gray code". Both the SSI interface and the RS-485 interface can be configured as SSI interface via the predefined function blocks "enc_ssi_mt" or "enc_ssi_st" in EX_PRESS 5.



Note:
The SSI interface is assigned in "EX_PRESS 5" (Art. No. 589092) via the corresponding function block (see user manual for the program development system EX_PRESS 5, Chapter 4.4 "Function blocks in devices.lib")

Overview | Drehgeber: rot_enc_T1.s16 *

```
27 VAR
28     Encoder: ENC_SSI_MT;
29     num_rev_int: INT;
30     num_inside: DINT;
31 END_VAR;
32
33 (* external/network based *)
34 VAR_GLOBAL
35     num_of_rev_g: INT;   (* number of revolutions    *)
36     inside_rev_g: INT;   (* value inside revolution  *)
37 END_VAR;
38
39 (* ********************************** *)
40 (* *** SSI Function Block          *** *)
41 (* ********************************** *)
42
43     Encoder ( VAL_IN_REVOL := num_inside,
44               NUM_REVOLUTN := num_rev_int, NUM_OF_IF := 2,
45               NUM_OF_MAXBIT := 24, MID_BIT_POS := 12,
46               NUM_OF_DATABIT := 24, BIT_RESOLUTION := 12,
47               FREQ_VALUE := 800,
48               BINARY_NGRAY := 0 );
49
50     num_of_rev_g := num_rev_int;
51     inside_rev_g := num_inside;
```

Fig. 17: Program example with function block for multiturn encoder

5.5.6 Connection of an Encoder to the SSI-Interface

Fig. 18 shows the connection of an encoder to the SSI interface. The cables of the SSI encoder are connected to the control system via a shielded cable twisted in pairs. Connect the cable shield at one end as close as possible to the terminals of the control with a low-resistance shielding rail at PE potential (see Fig. 18). The data signal is connected to terminals D+ and D- (or 2+ and 2- if the RS 485 connection has been reconfigured accordingly via the function block in EX_PRESS 5). The positive signal is connected to D+ (2+) and the negative signal to D- (2-). The clock signals of the encoder are connected to the terminals Ck+ and Ck-, or 1+ and 1-.

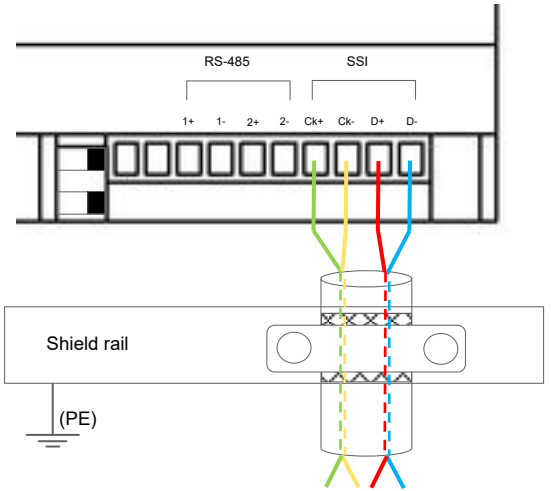


Fig. 18: Connection of an encoder

5.5.7 TTL differential signal inputs (only ZX09A/D/E, ZX21TPA/TCA)

The SSI or RS-485 interfaces can also be used to connect sensors with TTL differential outputs, e.g. corresponding incremental encoders with rectangular TTL outputs. Up to four TTL differential inputs are available via the "SSI" and "RS485" interfaces (see also section 5.5.9, Connecting the inputs/outputs for TTL differential signals).
In EX_PRESS 5, the signals are read into the control system by software via the following function blocks and can then be integrated into the logic of the user program:

- RS485_In_11 for the cable pair Ck+ and Ck– of the SSI-port
- RS485_In_12 for the cable pair D+ and D– of the SSI-port
- RS485_In_21 for the cable pair 1+ and 1– of the RS-485-port
- RS485_In_22 for the cable pair 2+ and 2– of the RS-485-port.

5.5.8 TTL differential signal outputs (ZX09A/D/E, ZX21TPA/TCA)

The SSI or RS-485 interfaces can also be used as differential outputs to connect actuators with TTL differential inputs. Up to four TTL differential inputs are available via the "SSI" and "RS485" interfaces (see also section 5.5.9, Connecting the inputs/outputs for TTL differential signals).
The TTL differential outputs are set up in EX_PRESS 5 by software using the following function blocks:

- RS485_Out_11 for the cable pair Ck+ and Ck– of the SSI-port
- RS485_Out_12 for the cable pair D+ and D– of the SSI-port
- RS485_Out_21 for the cable pair1+ and 1– of the RS-485-port
- RS485_Out_22 for the cable pair 2+ and 2– of the RS-485-port

5.5.9 Connection of the In-/Outputs for the TTL differential signals

The wiring of the inputs for TTL differential signals is made via terminals 1+ and 1- or 2+ and 2- of the RS-485 interface or terminals Ck+ and Ck- or D+ and D- of the SSI interface. The connection is made via shielded cables stranded in pairs. Connect the cable shield at one end as close as possible to the terminals of the controller with a low-resistance shielding rail at PE potential. The type of wiring is identical with the connection of an SSI encoder (see Fig. 18).



Note:
The acquisition of the TTL differential inputs or output to the TTL differential outputs is carried out in "EX_PRESS 5" (Art. No. 589092) via the corresponding function blocks (see user manual for the program development system EX_PRESS 5, Chapter 4.6 "Function blocks in user.lib")

5.6 Digital Outputs

The ZX09 series controllers have 4 digital outputs and the ZX20/21 series controllers have 16 digital outputs. These are accessible on the ZX20/21 series via the two upper terminal blocks (see Fig. 20) and on the ZX09 series via the upper right terminal block (see Fig. 19).
The outputs are not electrically isolated from the supply voltage, i.e. their 0 V reference potential is identical to the 0 V potential of the supply voltage (terminal "M").



Note:

In the "Structured Text", output variables for digital outputs may be assigned to the above mentioned terminals by means of the keyword "AT", namely by the designations "Out_01", "Out_02", ... , "Out_4" res. "Out_16". Detailed information can be found in chapter 2.2.5.1 of the programming manual, which is available as a PDF file in the software package "EX_PRESS 5" (Art-no. 589092).

If the programmed logic of the user program switches one of the outputs, this is indicated by four green LEDs to the right of the terminals (see Fig. 19). Here LED 1 corresponds to output 1, LED 2 to output 2, etc. If one of the outputs is activated, it switches through the voltage applied to U+, otherwise the output is open (ZX20/ZX21) or is actively pulled to 0 V (ZX09 family).

All digital outputs are permanently short-circuit proof. This is only valid for real short circuits, i.e. a low impedance connection with 0 V, not for overload.

Warning:
The switching capacity information of the digital outputs specified in chapter 12 "Technical data" must be observed.
In case of overload the output drivers can be destroyed.

In the event of a short circuit or if there is no voltage at "U+", the "OC" LED to the left of the terminals lights up on the ZX09 series controllers. When controlling inductive loads, a protective circuit in the form of a free-wheeling diode or a varistor must be provided.
It is possible to connect the supply voltage of the outputs without connecting the operating voltage. In this case the outputs are not switched.

5.6.1 Connection of the digital outputs

The connection of the digital outputs of the ZX09 series is shown in fig. 19, the connection of the ZX20/21 series in fig. 20. Different actuators (e.g. relays) can be connected to the terminals of the outputs. Their 0 V potential is connected to the 0 V potential of the voltage source for the digital outputs (terminal "M").
A voltage source must be connected to U+ and M so that a voltage is present when the outputs are switched. This can be supplied via the operating voltage source or a separate voltage source (10 V to 30 V, see chapter 12 "Technical Data"). If a separate voltage source is used, an additional connection of the 0 V output of the voltage source to the 0 V output of the operating voltage source is not necessary, since the terminals "M" are connected to each other internally in the device.

Note:
Please observe the permissible voltage range for the digital outputs in chapter 12 "Technical Data".

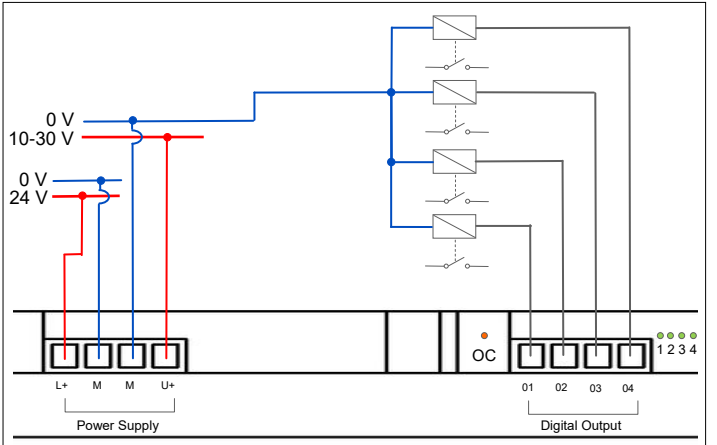


Fig. 19. Connection of the digital outputs (here e.g. relays) - ZX09-series

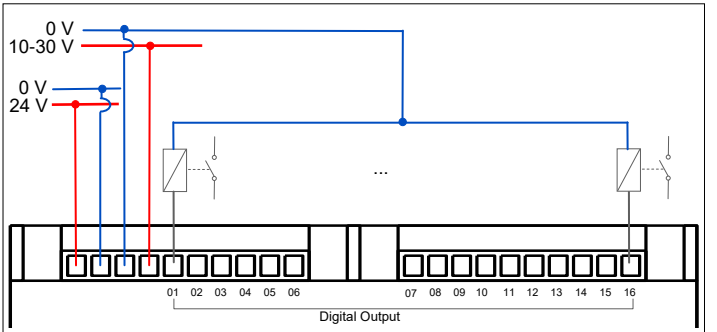


Fig. 20. Connection of the outputs (here e.g. relays) - ZX20/21-series

6. Programming

For programming, the separately available programming system "EX_PRESS 5" is required. The program is created in the PLC programming language "Structured Text" according to IEC 61131-3.
The finished program is downloaded to the controller via the Ethernet interface, which is marked on the device by the "" symbol. The connection to the PC can be a point-to-point connection or, alternatively, via a network. Several controllers of the same type can be located in this network. In both cases, the PC must be connected via the Ethernet interface.

Warning:
Do not switch off the supply voltage of the controller during the program download. If the supply voltage has been disconnected, it is possible that the controller can no longer be addressed. In this case, send the controller to the address in Chapter 15 "Service".

Note:
A programming manual, in which the programming system "EX_PRESS 5" (Art-no. 589092) is described in detail, can be found as a PDF document on the USB stick, which is included in the program package.

In addition to the nameplate on the side, each controller has a label with the corresponding MAC address (see Fig. 21). This label is located below the Ethernet interface on all ZX controllers. This MAC address is used to uniquely identify the controller on the network and can therefore be used for specific application programs. Programming via the Internet (IP-based) is not possible.

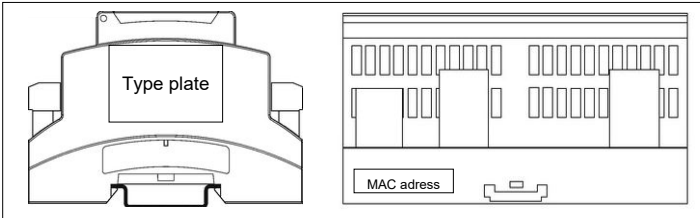


Fig. 21: Type plate and MAC adress

Important information:
In the case of the ZX21TC and ZX21TP controllers, downloading of the completed program into the controller is only possible if data are not simultaneously transferred via the PROFINET or EtherCAT interface.
With all other networking types, downloading is also possible while communication is running; network communication is only interrupted during the download process and then automatically resumed.

7. Networking

7.1 Networking via Ethernet

In addition to the possibility of loading a user program onto the controller, the Ethernet interface also allows Zander controllers to be integrated into a network.

7.1.1 Modbus/TCP

All controllers of the ZX09 or ZX20/21 series can be integrated as slaves (servers) in a Modbus/TCP network. A Modbus/TCP master (client), e.g. Zander MVisio HMI, can use the following Modbus function codes to access the network variables declared in the EX_PRESS 5 user program under "VAR_GLOBAL" and "VAR_EXTERNAL":

- Function code 3, Read Holding Registers
- Function code 6, Write Single Register
- Function code 16, Write Multiple Registers

The IP addresses of the Zander controllers are set in the user program using the reserved keywords "IP_ADR = ... ;" and "IP_MASK = ... ;" in the configuration section immediately after the keyword "PROGRAM".

The variables declared under "VAR_EXTERNAL" may be accessed reading and writing starting with Modbus address offset 1, the variables declared under "VAR_GLOBAL" may only be accessed reading starting with Modbus address offset 800.

Note:
Detailed information about configuration and programming of the Modbus/TCP communication can be found in the programming manual in chapter 9.2, which can be found as a PDF file in the software package "EX_PRESS 5" (Version 4.20 or higher, Art-no. 589092) on the USB stick enclosed there.

An exemplary network architecture of a Modbus/TCP system is shown in Fig. 22.

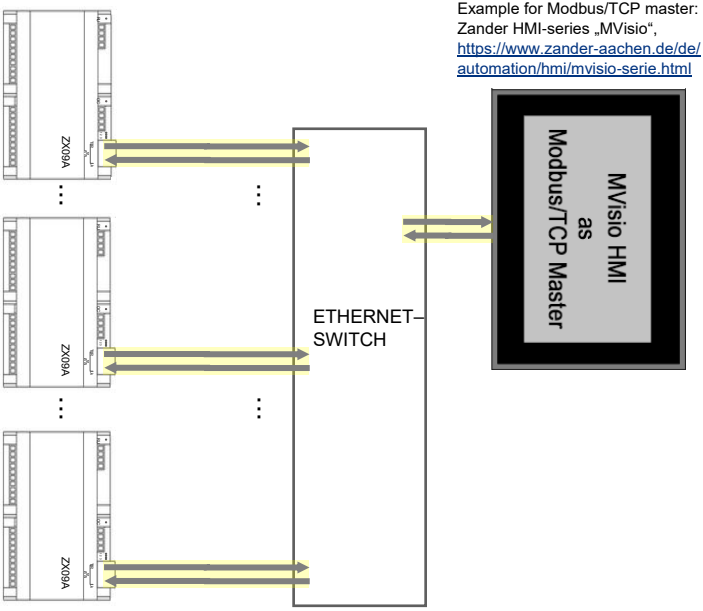


Fig. 22. Networking via Modbus TCP/IP (e.g. with ZX09A)

7.1.2 ZanderNet

The ZanderNet network protocol offers a further option for networking the controllers with each other via the Ethernet interface. Here, only Zander controllers of the ZX series communicate with each other (even different types from the ZX09 and ZX20/ ZX21 series mixed together) and form a distributed control system together. In this network, all controllers are equivalent, which corresponds to a multimaster operation.
If only two controllers should be connected with each other, this can be realized directly from controller to controller via an Ethernet cable. No further devices are necessary for this.
If more than two controllers should be connected with each other or controllers should be integrated into an already existing network, this can be realized by connecting them to an Ethernet switch. There is no upper limit for the number of controllers.
Fig. 23 A shows the networking of two controllers (exemplary with the ZX09A controllers), Fig. 23 B shows the networking via a switch (exemplary with the ZX09A controllers). There may be other devices in the same network as the Zander controllers, as their data traffic is ignored.



Note:

Detailed information on programming the communication via ZanderNet can be found in the programming manual in chapter 2.2.2.1, which can be found as a PDF file in the software package "EX_PRESS 5" (Art-no. 589092) on the included USB stick.

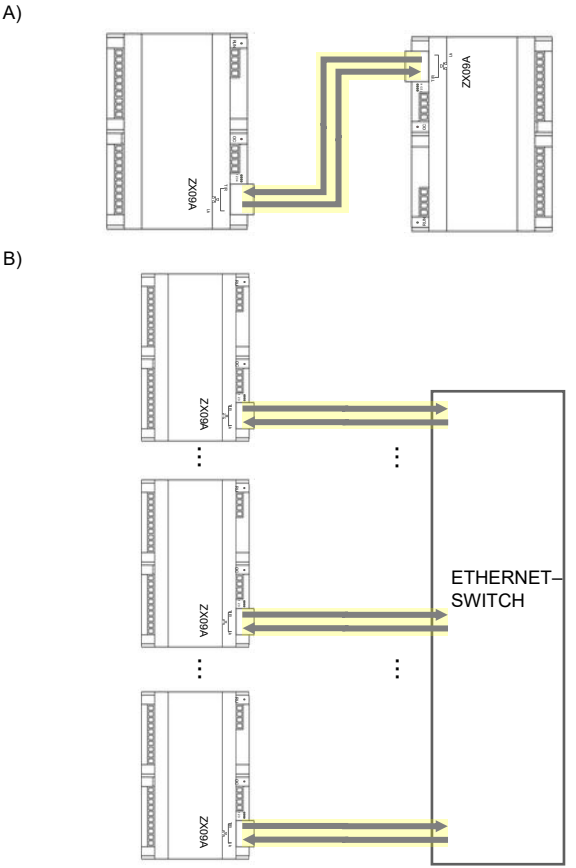


Fig. 23. Networking via ZanderNet (e.g. with ZX09A)

7.2 The RS 485-Interface (only ZX09A/D/E and ZX21TPA/TCA)

The controllers ZX09A, ZX09D, ZX09E, ZX21TPA and ZX21TCA have a RS-485/RS-422 interface. This is indicated on the device by the "RS-485" terminal. The interface marked "SSI" or “SSI/TTL” can also be used as a RS-485/RS-422 interface as an alternative for communication with an SSI compatible device (e.g. encoder).

7.2.1 Modbus RTU

The controllers ZX09A, ZX09D and ZX09E are able to communicate via their RS-485 interface with a superior controller using the Modbus RTU protocol. The master controller or an HMI is the master (client), the Zander controllers are slaves (server).
The Modbus RTU master (e.g. Zander MVisio HMI) uses the following Modbus function codes to access the network variables declared in the EX_PRESS 5 user program under "VAR_GLOBAL" and "VAR_EXTERNAL":

- Function code 3, Read Holding Registers
- Function code 6, Write Single Register
- Function code 16, Write Multiple Registers

The interface parameters of the Zander controllers are set in the user program using the reserved keywords "BITRATE = ... ;" for the data transmission rate, "SERIAL_FORMAT = ... ;" for the data format and "MB_ADR = ... ;" for the Modbus RTU device address in the configuration section immediately after the keyword "PROGRAM".

The variables declared under "VAR_EXTERNAL" may be accessed reading and writing from Modbus address offset 1, the variables declared under "VAR_GLOBAL" may only be accessed reading from Modbus address offset 800.



Note:

Detailed information on configuration and programming of the Modbus RTU communication can be found in the programming manual in chapter 9.2, which is available as a PDF file in the software package "EX_PRESS 5" (version 4.20 or higher, art. no. 589092) on the USB stick included in the package.

7.2.2 Modbus RTU wiring

The connection of one or more Zander controllers to a Modbus RTU Master is made via the connections 2+ and 2- of the RS-485 interface using a shielded twisted-pair two-wire cable (see Fig. 24). The shield must be connected to a shielding rail at one end with low impedance as close as possible to the device.
The topology of an RS-485 bus connection with several Modbus RTU participants is shown in Fig. 25. A terminating resistor of 120 Ohm must be used at the last participant in the bus.

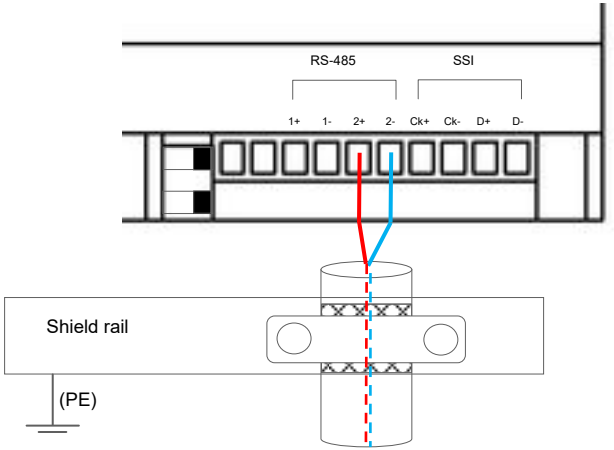


Fig. 24: RS-485 wiring (e.g. on ZX09A)

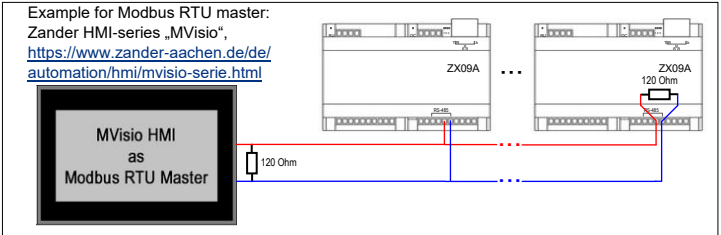


Fig. 25: Bus topology Modbus RTU (e.g. on ZX09A)

7.2.3 ZanderLink

Another way of using the RS-485 interface (or alternatively the SSI interface or SSI/TTL interface of ZX21TPA/TCA) is a point-to-point connection of two ZX controllers using "ZanderLink". One of the controllers is master, the other slave. The differences between master and slave are the initiation of the communication, which originates from the master, and the clock generation of the synchronous transmission, which is also done by the master and transmitted to the slave.
The master sends its packet, the slave responds with minimal delay, then again the master, and so on.

ZanderLink is designed for the exchange of relatively little user data, but this data is transferred very quickly. For example, the cyclic exchange of 32 bit user data (master to slave + slave to master) is possible within a cycle time of 3.75 µs.

In contrast to the other types of networking, ZanderLink does not use the network variables "VAR_EXTERNAL" and "VAR_GLOBAL", but the programming of the data transmission is done via the function block "ZLINK".



Note:

Detailed information Programming the ZanderLink communication can be found in the programming manual in chapter 9.3, which can be found as a PDF file in the software package "EX_PRESS 5" (version 4.20 or higher, art. no. 589092) on the USB stick included there.

7.2.4 ZanderLink wiring

ZanderLink uses the complete RS-485 interface for the connection per subscriber, i.e. all four terminals, since it is a synchronous serial transmission (one channel for a clock signal, the second channel for the data bits).
The connection of two Zander controllers via ZanderLink is made via the terminals 1+ and 1- as well as 2+ and 2- of the RS-485 interface using a shielded twisted-pair four-wire cable (see Fig. 26a). The shield must be connected to a shielding rail as close as possible to the devices with low impedance. A terminating resistor of 120 Ohm must be connected to terminals 1+ and 1- or 2+ and 2-.
Alternatively, the SSI or SSI/TTL port of the controllers can be used for ZanderLink. Here the terminals Ck+ and Ck- as well as D+ and D- of both participants are to be connected to each other by means of a shielded twisted-pair four-wire cable (see fig. 26b). The SSI ports are already equipped with 120 Ohm terminating resistors inside the device, so that no external resistors have to be connected here.
The SSI port and the RS-485 port can also be used in parallel, so that, for example, a master can communicate with two slaves or one slave can simultaneously be the master for another slave.

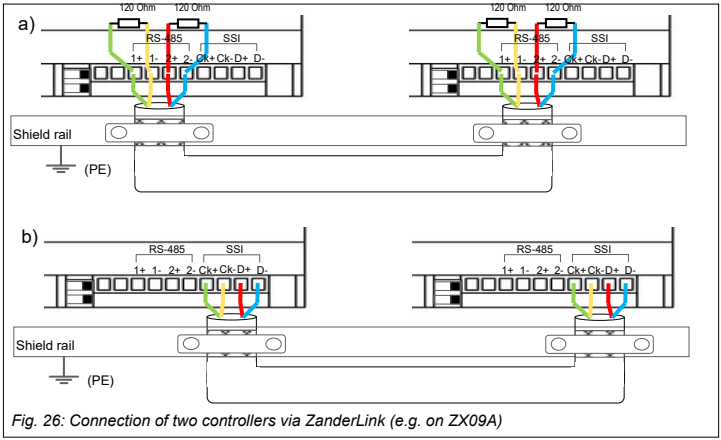


Fig. 26: Connection of two controllers via ZanderLink (e.g. on ZX09A)

7.3 Networking via Profinet (only ZX21TP/ZX21TPA)

The ZX21TP controller can be integrated into a PROFINET network as a "PROFINET IO Device". There it serves as subsystem and exchanges data with a "PROFINET IO Controller".
One of the two RJ45 sockets of the "PROFINET IO Device" interface is connected to the "PROFINET IO Controller" by means of an Ethernet cable. The two RJ45 sockets are equivalent and have an integrated switch. Thus it is possible to connect another "PROFINET IO Device".
The GSDML file must be included in the "PROFINET IO Controller" configuration software. In the software, the input and output data which are exchanged with the "PROFINET IO Controller" must be configured on slot 1 (output data) or slot 5 (input data). The input and output data can each comprise up to 64 bytes. PROFINET RT is supported as protocol standard (not PROFINET IRT). For further information please refer to the documentation of your "PROFINET IO Controller" configuration software. To check your configuration, use the LEDs on the device front (see chapter 9 "Diagnostics").



Note:
The GSDML file can be found in XML format on the USB stick that comes with the program package.

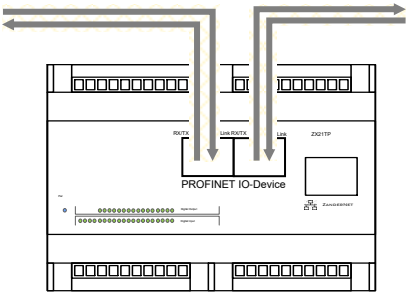


Fig. 27: Networking via PROFINET

7.4 Networking via EtherCAT (only ZX21TC/ZX21TCA)

The ZX21TC controller can be integrated into an EtherCAT network as an "EtherCAT slave". There it serves as a subsystem and exchanges data with an "EtherCAT Master".
The right RJ45 socket ("IN") is connected to the "EtherCAT Master" or the "OUT" of a previous "EtherCAT Slave". If a further "EtherCAT Slave" follows in the chain, it has to be connected to the left RJ45 socket ("OUT").
The EDS file must be included in the "EtherCAT Master" configuration software. The input and output data that are exchanged with the master can each contain up to 64 bytes. The "Distributed Clocks" function is not supported. For further information please refer to the documentation of your "EtherCAT Master" configuration software.
To check your configuration, use the LEDs on the front of the device (see chapter 9 "Diagnostics")



Note:
You will find the EDS file in XML format on the USB stick that is included in the program package.

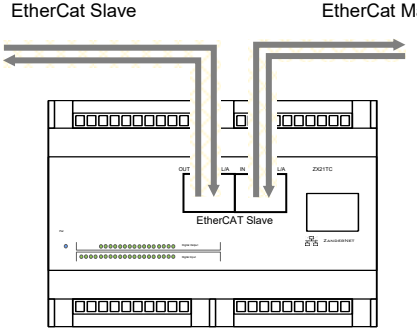


Fig. 28: Networking via EtherCAT

8. Commissioning

Before the final commissioning of the controller, a qualified specialist must verify that the previous steps "Electrical connection" and "Programming" and, if necessary, "Networking" have been carried out correctly. It is recommended to execute the program sequence first with the voltage for the actuators switched off and to verify the correct function at the LEDs for the outputs (see chapter 9 "Diagnosis").



Warning:
A faulty electrical connection or incorrect programming can lead to unpredictable operating conditions and result in fatal or serious injuries and high material damage.

The controllers of the ZX09 series, as well as the ZX20/21 series are not suitable for fulfilling functional safety tasks. Therefore, if dangerous movements or dangerous states are possible in the overall system, a risk assessment must be carried out in accordance with the Machinery Directive. If additional measures are to be taken with regard to functional safety, approved safety components such as emergency stop devices or safety door monitors must be provided for this purpose, which enable the safe shutdown of the corresponding drives independently of one of the control systems mentioned. Suitable components for this purpose can be obtained from ZANDER.

8.1 What happens at the start?

At the first start the device initializes without function. Only the RUN LED lights up green (ZX09 series) or the Pwr LED blue (ZX20/21 series).
In order for the device to fulfil the desired function, the device must be put into operation correctly. To do this, follow the steps in chapter 8.2 "Initial commissioning procedure".

8.2 Initial commissioning

8.2.1. Step 1: Program creation in EX_PRESS 5

For an exact description of the program creation, please follow the instructions and information from the programming manual and chapter 7 "Program creation" of the installation manual EX_PRESS 5.
The program is created in the programming language "Structured Text".

8.2.2 Step 2: Connecting the power supply

The power supply must be connected so that the PLC can be programmed. Proceed as described in Chapter 5.4 "Electrical connection of the instrument". Only switch on the supply voltage after wiring. The switched on PLC is indicated by the green RUN LED (ZX09 series) or the blue Pwr LED (ZX20/21 series).



Attention:
The maximum permissible operating voltage is 24V + 15% = 27.5 V (see chapter 12, Technical Data).

8.2.3 Step 3: Programming the PLC

For program transfer, the PLC must be connected to an existing network to which the PC is connected or directly to the PC via the Ethernet interface. To do this, connect an Ethernet cable to the PLC. Connect the other end to your computer or an existing network. The correct network connection is indicated by the LEDs on the Ethernet socket. Afterwards, the PLC can be programmed with the created program. Please refer to chapter 1.6.2 "Calling the programming tool" in the programming manual.

8.2.4 Step 4: Connecting the inputs

When programming is complete, the inputs can be connected. Proceed as described in section 5.5.2 "Connecting the digital inputs" and section 5.5.4 "Connecting the analog inputs" (ZX09A/D/E and ZX21TPA/TCA only). If no input signals are provided, this step can be skipped.

8.2.5 Step 5: Connecting the encoder (ZX09A/D/E, ZX21TPA/TCA)

Now connect the encoder as described in section 5.5.6 "Connecting a device to the SSI interface".
If the connection of an encoder is not intended, this step can be skipped.

8.2.6 Step 6: Controlling the function of the PLC via the LEDs

Before connecting the network and the outputs, the operation of the PLC should be checked via the LEDs. If the program does not correspond to the desired function, a new programming with connected digital inputs can be carried out. It is therefore not necessary to repeat steps 1-5. You should only continue when it has been ensured that the desired function is correctly fulfilled.

8.2.7 Step 7: Connection to a network

If it is intended to use the controller within a data transmission network (Modbus/TCP or ZanderNet), connect the controller to the network using an Ethernet cable. Proceed as described in chapter 7.1 "Networking via Ethernet".
If data communication should take place via the RS-485 interface (Modbus RTU (only ZX09A/D/E) or ZanderLink (only ZX09A/D/E or ZX21TPA/TCA)), carry out the wiring as described in chapter 7.2 "The RS-485 interface".
If the device should be used within a PROFINET network (only ZX21TP, ZX21TPA), proceed as described in chapter 7.3 "Networking via PROFINET".
If the device should be used within an EtherCAT network (only ZX21TC, ZX21TCA), proceed as described in section 7.4 "Networking via EtherCAT".
If integration into a network is not intended, this step can be skipped.

	**Warning:** If it is intended that the controller should send data to a controller that has already been commissioned, it is recommended that the controller which has already been installed is switched off for the period of initial commissioning, as otherwise data may already be received.

8.2.8 Step 8: Connecting the outputs

In the next step the digital outputs can now be connected. For this, proceed as described in chapter 5.6.1 "Connecting the digital outputs". If an output is programmed, the output LEDs 1..4 light up according to the programming. No signal is emitted at the digital outputs as long as U+ is not connected.
If an output of signals via the digital outputs is not planned, this step can be skipped.

8.2.9 Step 9: Connecting the power supply of the digital outputs

In the last step the power supply of the digital outputs is connected. Proceed as described in chapter 5.6.1 "Connection of the digital outputs". Only switch on the supply voltage after connecting it to the terminals.

9. Diagnosis

There are various LEDs on the controllers, which allow easy diagnosis and are helpful in the event of any necessary troubleshooting. A green LED labelled RUN (ZX09 series, see Fig. 28) or a blue LED labelled Pwr (ZX20/21 series, see Fig. 29) is used to indicate the presence of the operating voltage in addition to the power supply connection.
The green "RUN" LED on the ZX09 series outputs flashing codes in the event of an error. For this purpose, a number of pulses corresponding to the error code is output repeatedly:

Error	Error code
Error at the digital outputs (red LED "OC" is lit additionally)	7
Error in checksum (retain variables)	5

The states of the digital outputs are indicated by the LEDs 1..4 (ZX09 series) next to the digital outputs or by the LEDs Digital Output 01..16 (ZX20/21 series) on the top of the controller (see Fig.28 or 29). In addition, the ZX09 series has a red LED labelled OC next to the outputs (see Fig. 28). This indicates an "Over Current", i.e. a short circuit at one of the digital outputs, or the absence of a power source at U+ / M.

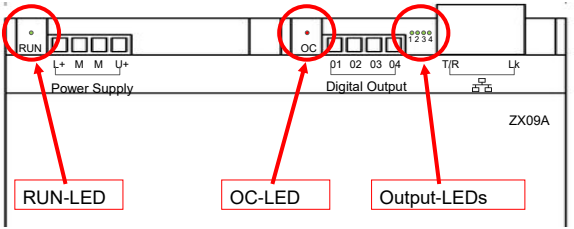


Fig. 28: Diagnosis LEDs (ZX09A, exemplary)

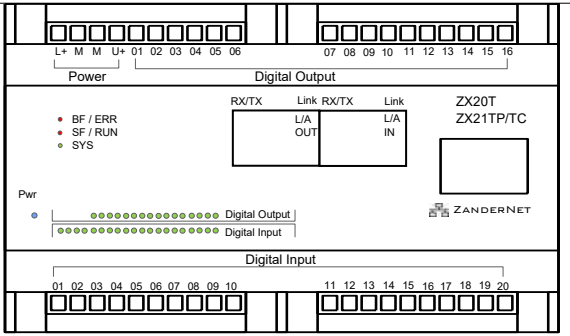


Fig. 29: Diagnosis LEDs (ZX20/21-series)

The following tables list the possible states of the LEDs and their meaning for the control variants with PROFINET interface (ZX21TP/TPA) or EtherCAT interface (ZX21TC/TCA).

ZX21TP/ZX21TPA:

LED	Color	Status	Meaning
SYS	Green	On	Bus module ready for operation
	Green/ Yellow	Flashing Green/Yellow	Bus module moves up
	Yellow	On	
	-	Off	Device is switched off or bus module is defective
BF	Red	On	No configuration; no/bad connection
	Red	Flashing cyclically with 2 Hz	No data exchange
	-	Off	No error
SF	Red	On	Watchdog time-out; system error
	Red	Flashing cyclically with 2 Hz	DCP signal service is triggered via the bus
	-	Off	No error
Link Ch0 Ch1	Green	On	There is a connection to the Ethernet
	-	Off	There is no connection to the Ethernet
RX/TX Ch0 Ch1	Yellow	Flashing	Data is sent/received

ZX21TC/ZX21TCA

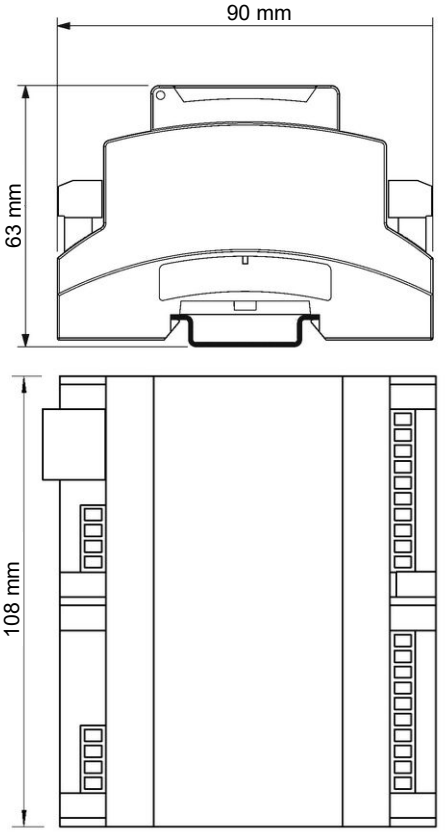
LED	Color	Status	Meaning
SYS	Green	On	Bus module ready for operation
	Green/ Yellow	Flashing Green/Yellow	Bus module moves up
	Yellow	On	
	-	Off	Device is switched off or bus module is defective
ERR	Red	Flashing	Invalid configuration
	Red	Single flash	Local error: The slave device application has changed the EtherCAT state independently
	Red	Double flash	Process data watchdog timeout
	-	Off	No error
RUN	Green	Flashing	The device is in pre-operational state
	Green	Single flash	The device is in Safe Operational State
	Green	On	The device is in operational state
	-	Off	The device is in INIT state
L/A IN L/A OUT	Green	On	There is a connection
	Green	Flashing	Data is sent/received
	Green	Off	There is no connection
L/A IN L/A OUT	Yellow	Off	This LED is not used

10. Maintenance, repair and replacement

Assuming correct installation, maintenance work is not required. Repairs to the device may only be carried out by the manufacturer. Opening the device by the user will result in the loss of any warranty or guarantee claims.
For a possibly necessary replacement, it must be ensured that the new control unit is provided with the identical user program before it is put into operation again.

11. Dimensional drawing

Fig. 30: Dimer



12. Technical data

12.1 ZX09 / ZX09A

Electrical data	
Operating voltage U_B (terminals „L+“ and „M“)	DC 24 V
Voltage tolerance	± 15%
Inrush current (with power-on) for <500 ms	approx. 200 mA
Current consumption at 24 V	approx. 80 mA
Power consumption at 24 V	approx. 1.9 W
Internal reverse polarity protection	Yes
Internal short circuit protection	Yes, electronic (self resetting)
Digital inputs	
Quantity	10
Input voltage	18 - 30 V
Input current at 24 V	6.5 mA
Input resistance	3,7 kΩ
Galvanic separation	No
Low-level	< 5 V
High-level	> 18 V
Pulse suppression	Configurable 1µs - 1073 s (see programming manual)
Max. Switch-on delay	1 µs
Max. Input frequency	500 kHz
Reaction time of digital inputs to outputs	< 3 µs
Digital outputs	
Quantity	4
Structure	Push-/Pull driver
Switching capacity per output	DC 10 - 30 V; 500 mA
Galvanic separation	No
Short circuit proof	Yes (see chapter 5.6 "Outputs")
Switching time	1 µs
Output voltage at “1“ (max. load)	„U+“ - 0.5 V
Output voltage at „0“	< 0.2 V
Max. switching frequency with clock ratio 1:1	50 kHz

Analog Input (ZX09A)	
Quantity	1
Type	0..10 V or 0..20 mA
Resolution	12 Bit
Accuracy	Range 1 (0-10 mA): ±0.1% of full scale (20mA) Range 2 (10-20 mA): ±0.2% of full scale (20mA) Range 3 (0-10V): ±0.1% of full scale (10V)
Offset	≤ ±2 mV or ≤ ±0.0004 mA
Linearity	Current measurement: ≤ 6 LSB Voltage measurement: ≤ 4 LSB
Reaction time analog input to FPGA chip	< 70 µs
Input resistance Switch position „10 V“	1000 MΩ
Input resistance Switch position „20 mA“	165 Ω
Low pass filter cut-off frequency (TP 2nd order)	110 kHz
Dielectric strength switch position „10 V“	35 V
Current resistance Switch position „20 mA“	38 mA
Processing time	
Internal processing time FPGA	< 20 ns
Interfaces	
RS 485/422 (only ZX09A)	1
SSI (only ZX09A)	1 (2), 20 Mbit/s, Multi-Turn, Single-Turn
Ethernet	1, 10Base –T or 100Base-T
Environmental data	
Ambient temperature	0 °C to 50 °C
Storage temperature	-20 °C to 80 °C
Moisture stress	93% r.m. at +40 °C, dewless
Vibrations according to EN 61131-2	Frequency: 5 - 8.4 Hz, 3.5 mm amplitude Frequency: 8.4 - 150 Hz, 1.0 g acc.
Shock according to EN 61131-2	Acceleration: 15 g
Mechanical data	
Protection class	IP20
Assembly	Mounting rail TH25 after EN 60715: 2018-07
Cable cross section	0.25 - 2.5 mm^2
Dimensions (B x H x T)	108 x 90 x 62 mm
Weight	210 g
Housing material	Polycarbonate
Connection type	Pluggable screw terminal blocks

12.2 ZX09D / ZX09E

Electrical data	
Operating voltage U_B (terminals „L+“ and „M“)	DC 24 V
Voltage tolerance	± 15%
Inrush current (with power-on) for <500 ms	approx. 200 mA
Current consumption at 24 V	approx. 80 mA
Power consumption at 24 V	approx. 1.9 W
Internal reverse polarity protection	Yes
Internal short circuit protection	Yes, electronic (self resetting)
Digital inputs	
Quantity	2
Input voltage	18 - 30 V
Input current at 24 V	6.5 mA
Input resistance	3.7 kΩ
Galvanic separation	No
Low-level	< 5 V
High-level	> 18 V
Pulse suppression	Configurable 1µs - 1073 s (see programming manual)
Max. switch-on delay	1 µs
Max. input frequency	500 kHz
Reaction time of digital inputs to outputs	< 3 µs
Digital outputs	
Quantity	4
Structure	Push-/Pull driver
Switching capacity per output	DC 10 - 30 V; 500 mA
Galvanic separation	No
Short circuit proof	Yes (see chapter 5.6 "Outputs")
Switching time	1 µs
Output voltage at “1“ (max. load)	„U+“ - 0.5 V
Output voltage at „0“	< 0.2 V
Max. switching frequency with clock ratio 1:1	50 kHz
Analog inputs	
Quantity	6
Type ZX09D:	4x 0..10 V (non-adjustable) 2x 0..10V or 0..20 mA (switchable)
Type ZX09E:	4x 0..20mA (non-adjustable) 2x 0..10V or 0..20 mA (switchable)

Resolution	12 Bit
Accuracy	Range 1 (0-10 mA): ±0.1% of Final measurement value (20mA) Range 2 (10-20 mA): ±0.2% of Final measurement value (20mA) Range 3 (0-10V): ±0.1% of Measurement end value (10V)
Offset	≤ ±2 mV or. ≤ ±0,0004 mA
Linearity	Current measurement: ≤ 6 LSB Voltage measurement: ≤ 4 LSB
Reaction time analog input to FPGA chip	< 10 µs (also parallel)
Input resistance Switch position „10 V“	1000 MΩ
Input resistance Switch position „20 mA“	165 Ω
Low pass filter cut-off frequency (TP 2nd order)	110 kHz
Dielectric strength switch position „10 V“	35 V
Current resistance Switch position „20 mA“	38 mA
Processing time	
Internal processing time FPGA	< 20 ns
Interfaces	
RS-485 / SSI	1 / 1 (2) single-turn, multi-turn
Ethernet	1, 10Base -T or 100Base-T
Environmental data	
Ambient temperature	0 °C to 50 °C
Storage temperature	-20 °C to 80 °C
Moisture stress	93% r.m. at +40 °C, dewless
Vibrations according to EN 61131-2	Frequency: 5 - 8.4 Hz, 3.5 mm amplitude Frequency: 8.4 - 150 Hz, 1.0 g acc.
Shock according to EN 61131-2	Acceleration: 15 g
Mechanical Data	
Protection class	IP20
Assembly	Mounting rail TH25 according to EN 60715: 2018-07
Cable cross section	0.25 - 2.5 mm^2
Dimensions (B x H x T)	108 x 90 x 62 mm
Weight	210 g
Housing material	Polycarbonate
Connection type	Pluggable screw terminal blocks

12.3 ZX20T, ZX21TP/TC

Electrical Data		
Operating voltage U_B (Terminal „L+“ und „M“)		DC 24 V (± 15%)
Current input at U_B	for ZX20T	approx. 60 mA
	for ZX21TP/TC	approx. 160 mA
(all inputs activated / without load)		
Power consumption at U_B	for ZX20T, ZX21T	approx. 1.5 W
	for ZX21TP/TC	approx. 3.9 W
(all inputs activated / without load)		
Digital inputs		
Quantity		20
Input voltage		18 - 30 V
Input current		7 mA
Galvanic separation		No
Low-level		< 5 V
High-level		> 18 V
Pulse suppression		Configurable (see programming manual)
Max. Switch-on delay		1 µs
Max. Input frequency		500 kHz
Digital outputs		
Quantity		16
Structure		PNP-transistors
Switching capacity per output		DC 10 - 30 V; 500 mA
Galvanic separation		No
Short circuit proof		Yes (see chapter 5.6 „outputs“)
Switching time		7 µs *)
Output voltage at “1“ (max. load)		„U+“ - 0.5 V
Processing time		
Internal processing time FPGA		< 20 ns
Interfaces		
Ethernet		Ethernet interface, 10Base-T or 100Base-T
PROFINET (ZX21TP)		PROFINET IO Device Interface
EtherCAT (ZX21TC)		EtherCAT Slave Interface

*) The specified switching times for the digital outputs only apply to a load current of at least 250 mA. With smaller load currents, the switching time is longer; if necessary, the load current can be increased by parallel connection of an additional resistor.

Environmental data	
Ambient temperature	0 °C to 50 °C
Storage temperature	-20 °C to 80 °C
Moisture stress	93% r.m. at +40 °C, dewless
Vibrations according to EN 61131-2	Frequency: 5 - 8.4 Hz, 3.5 mm amplitude
	Frequency: 8.4 - 150 Hz, 1.0 g acc.
Shock according to EN 61131-2	Acceleration: 15 g
Mechanical data	
Protection class	IP20
Assembly	Mounting rail after EN 60715TH35
Cable cross section	0.25 - 2.5 mm^2
Dimensions (B x H x T)	108 x 90 x 62 mm
Weight	210 g
Housing material	Polycarbonate
Connection type	Pluggable screw terminal blocks

12.4 ZX21TPA/TCA

Electrical Data	
Operating voltage U_B (Terminal „L+“ und „M“)	DC 24 V (± 15%)
Current input at U_B (all inputs activated / without load)	approx. 160 mA
Power consumption at U_B for ZX21TP/TC (all inputs activated / without load)	approx. 3.9 W

Digital inputs	
Quantity	8
Input voltage	18 - 30 V
Input current	7 mA
Galvanic separation	No
Low-level	< 5 V
High-level	> 18 V
Pulse suppression	Configurable (see programming manual)
Max. Switch-on delay	1 µs
Max. Input frequency	500 kHz

Digital outputs	
Quantity	4
Structure	Push-Pull driver
Switching capacity per output	DC 10 - 30 V; 500 mA
Galvanic separation	No
Short circuit proof	Yes (see chapter 5.6 „outputs“)
Switching time	1 µs
Output voltage at “1“ (max. load)	„U+“ - 0.5 V
Processing time	
Internal processing time FPGA	< 20 ns
Interfaces	
Ethernet	Ethernet interface, 10Base-T or 100Base-T
PROFINET (ZX21TPA)	PROFINET IO Device Interface
EtherCAT (ZX21TCA)	EtherCAT Slave Interface

Analog Inputs	
Quantity	4
Type	0..10 V or 0..20 mA
Resolution	12 Bit
Accuracy	Range 1 (0-10 mA): ±0.1% of full scale (20mA) Range 2 (10-20 mA): ±0.2% of full scale (20mA) Range 3 (0-10V): ±0.1% of full scale (10V)
Offset	≤ ±2 mV or ≤ ±0.0004 mA
Linearity	Current measurement: ≤ 6 LSB Voltage measurement: ≤ 4 LSB
Reaction time analog input to FPGA chip	< 15 µs
Input resistance Switch position „10 V“	1000 MΩ
Input resistance Switch position „20 mA“	165 Ω
Low pass filter cut-off frequency (TP 2nd order)	110 kHz
Dielectric strength switch position „10 V“	35 V
Current resistance Switch position „20 mA“	38 mA

Analog Outputs	
Quantity	4
Type	0..10 V
Resolution	16 Bit
Accuracy	Range 1 (0-10 V): ±0.1% of full scale (10 V)
Reaction time FPGA chip to analog output	< 10 µs
Power per analog output	> 20 mA at 10 V

Environmental data	
Ambient temperature	0 °C to 50 °C
Storage temperature	-20 °C to 80 °C
Moisture stress	93% r.m. at +40 °C, dewless
Vibrations according to EN 61131-2	Frequency: 5 - 8.4 Hz, 3.5 mm amplitude
	Frequency: 8.4 - 150 Hz, 1.0 g acc.
Shock according to EN 61131-2	Acceleration: 15 g
Mechanical data	
Protection class	IP20
Assembly	Mounting rail after EN 60715TH35
Cable cross section	0.25 - 2.5 mm^2
Dimensions (B x H x T)	108 x 90 x 62 mm
Weight	210 g
Housing material	Polycarbonate
Connection type	Pluggable screw terminal blocks

12.5 FPGA logic capacities

Device family	Logic cells	Registers
ZX09-series	9112	18224
ZX20-series	1792	3584
ZX21-series	3584	7168

13. Example 1

In this example, the steps of initial commissioning (see section 8.2) are carried out with a ZX09A for a specific example. Starting with the development of a program in step 1, the individual steps are carried out until the system is fully operational after completion of step 9.

13.1 The process

In this example, the control system of a bottling plant is shown. Fig. 31 shows the process which is to be automated. Empty containers should be transported to a filling plant via a controlled conveyor belt. With the aid of a light barrier it should be detected whether a new container has entered the plant. If this is the case, the supplying conveyor belt should stop and therefore not transport a new container to the plant. In the plant, the empty container should be transported under the filling hopper via a further conveyor belt. A proximity switch should detect whether the container is in the correct position. If this is the case, the filling process should be started. The filling process should be continuously monitored by a pressure sensor so that it will be filled optimally. As soon as the process is completed, the conveyor belt should transport the now filled container out of the plant. A second light barrier should detect whether the filled container has left the plant. If this is the case, the first conveyor belt can be started again to transport a new container to the plant. In addition to this, the system should have an abort switch with which the current filling process can be stopped and the error state can be exited.
Should an unknown condition or an error occur, the complete system should stop and indicate an error condition. The discharging conveyor belt is not controlled.

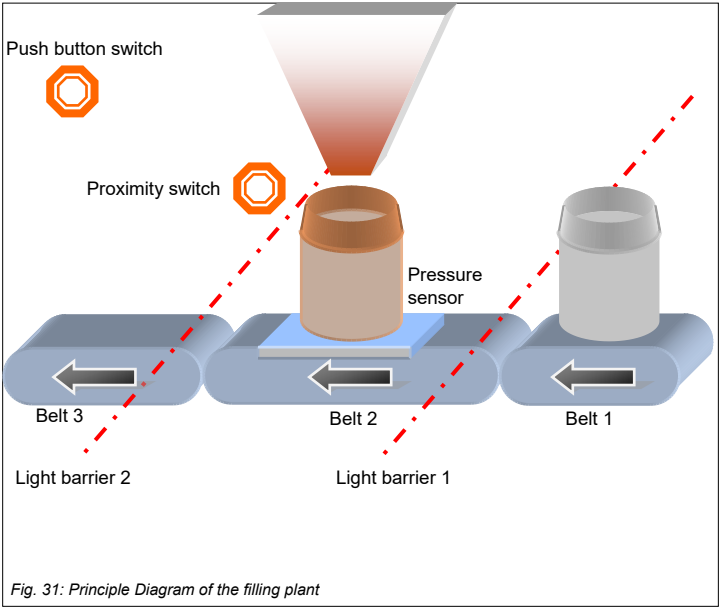


Fig. 31: Principle Diagram of the filling plant

13.2 Program creation in EX_PRESS 5

In the following the program corresponding to the application is shown.
In the first part of the program, marked by "DECLARATION", all necessary variables are declared and partly also initialized.
Under "VAR_INPUT" four digital input variables for the four sensors are declared for this purpose. The variables "Light barrier1" and "Light barrier2" correspond to the respective light barriers, "proximity switch" to the proximity switch and "push button switch" to the abort switch. These are assigned to terminals 1, 3, 5 and 7 via the "AT" command specific input terminals.
The output signals are declared under "VAR_OUTPUT". The application requires 3 control signals and one error status signal. The control signals are the control of the

filling process (hopper), the control of the conveyor belts 1 and 2 (belt 1 / belt 2) and the error signal.
Under "VAR_ADC" the analog input "pressure sensor" is declared. This is the pressure sensor, which should continuously record the filling status.
Under "VAR" the declaration of internal variables takes place. These are a status bit "filled" for the filling status, an error bit "error" and an initial bit "initial". These variables are all initialized with "0".
Finally, the internal clock of the controller is declared under "VAR_TIMER".
The actual program is written in the logic part, marked by "LOGIC PART". Via various "IF...ELSE" queries the control of the plant is implemented. This depends on the described process sequence of the plant.
In the last part, marked by "TIMERS", the timer is initialized and the individual variables are clocked.

```
PROGRAM Plant

PLC_NAME = "Filling process";
PLC_TYPE = "ZX09A";

(*--------------------DECLARATION--------------------*)
VAR_INPUT
        Lightbarrier1           AT In_01;
        Lightbarrier2           AT In_03;
        Pushbuttonswitch        AT In_05;
        Proxswitch              AT In_07;
END_VAR;

VAR_OUTPUT
        Funnel          AT Out_01;
        Belt1           AT Out_02;
        Belt2           AT Out_03;
        Mistake         AT Out_04;

END_VAR;

VAR_ADC
        Pressure sensor;
END_VAR;

VAR
        filled:         BIT:=0;
        Error:          BIT:=0;
        initial:        BIT:=0;

END_VAR;
VAR_TIMER
timer1;
```

```
END_VAR;

(*The input signals of the light barrier are debounced here. Therefore the container can be detected
by the sensor. The 1000ms correspond to the time the container needs to be transported past the
sensor.*)

Lightbarrier1.TDB           :=1000ms;
Lightbarrier2.TDB           :=1000ms;


(*--------------------LOGIC PART--------------------*)

(* in the first IF statement the initial state of the installation is defined. Here, volume 1 should be in
operation so that a new container can be delivered. Initial is then also set to 1 so that this condition
no longer applies.*)

IF initial=0 THEN
        Belt1:=1;
        Initial         :=1;
(*If the signal is applied to light barrier 1, belt 1 is stopped so that no further container is conveyed
into the system.*)

ELSIF Lightbarrier1=1 THEN
        Belt1:=0;
(*As soon as light barrier 2 is activated, belt 1 is reactivated so that a new container is conveyed to
the plant.*)

ELSIF Lightbarrier2=1 THEN;
        Belt1:=1;

(*The following two conditions contain two specific error states which must be defined. These are:
Case 1: Filling is started via "hopper" without a container being at the required position.
Case 2: New containers are transported to the system via belt 1, although a container is at the filling
position.
In both cases an error is set and the corresponding outputs are set.*)

ELSIF Proxswitch=0 AND Funnel=1 THEN
        Belt2:=1;
        Error           :=1;
        Funnel          :=0;

ELSIF Belt1=1 AND Proxswitch=1 THEN
        Error:=1;
        Belt1:=0;
        Belt2:=1;

(*This condition allows continuous replenishment within the plant. As long as there is no container at
the filling position and the push-button has not been operated, the belt will convey 2.*)


ELSIF Proxswitch=0 AND Pushbuttonswitch=0 AND Funnel=0 THEN
        Belt2:=1;
```

```
(*If the proximity switch is activated while belt 2 is running and the push button is not operated, belt 2
is stopped.*)

ELSIF Proxswitch=1 AND Belt2=1 AND Psuhbuttonswitch=0 THEN
        Belt2:=0;

(*If there is a proximity switch at the filling position, the filling process is carried out. This maintains
itself until the container has been filled. *)

ELSIF Belt2=0 AND Proxswitch=1 AND filled=0 AND Pushbuttonswitch=0 THEN

        IF Pressuresensor<3000 THEN
            Funnel      :=1;
        ELSE
            Funnel      :=0;
            filled      :=1;
        END_IF;

(* If the container is filled and is still at the filling position, conveyor 2 is started again. The full con-
tainer is then transported away.*)

ELSIF Belt2=0 AND Proxswitch=1 AND filled=1 AND Pushbuttonswitch=0 THEN
        Belt2           :=1;
        filled          :=0;

(If the push button is pressed while a container is at the filling position, the process is aborted and
the container is conveyed out of the system.*)

ELSIF Pushbuttonswitch=1 AND Proxswitch=1 THEN
        Funnel          :=0;
        Belt2           :=1;

(If the push button is pressed at any other time, even during an active error, it cancels the error state
and restarts bands 1 and 2.*)

ELSIF Pushbuttonswitch=1 THEN
        Error           :=0;
        Belt1:=1;
        Belt2:=1;

ELSIF Mistake=1 AND Pushbuttonswitch=1 THEN
        Error           :=0;
ELSE
        Error           :=1;
END_IF;

(* If there is an error condition, all processes are terminated.*)
IF Error=1 THEN
        Mistake         :=1;
        Band1           :=0;
        Band2           :=0;
        Funnel          :=0;
ELSE
```

```
                    Mistake          :=0;
END_IF;

(*****************************************************************************************************************)

(*--------------------TIMERS--------------------*)

timer1.POL       :=HIGH;
timer1.MODE      :=CONT;
timer1.ENABLE:=1;
timer1.RESET     :=0;
Timer1           :=50 us;

Funnel.CLK       :=timer1;
Belt1.CLK        :=timer1;
Belt2.CLK        :=timer1;
filled.CLK       :=timer1;
Mistake.CLK      :=timer1;
Error.CLK        :=timer1;
initial.CLK      :=timer1;

END_PROGRAM;
```

13.3 Step 2: Connecting the power supply

As described in Chapter 5.4 "Electrical connection of the device", the power supply is connected to the controller.

13.4 Step 3: Programming the PLC

For programming, the controller is connected to a computer via the Ethernet interface. The PLC is programmed using the EX_PRESS 5 software. After successfully compiling and fitting the program code, it can be transferred to the control (see chapter 1.6.2 "Calling the programming tool" in the EX_PRESS 5 manual).

13.5 Step 4: Connecting the inputs

Fig.32 shows the connection of the digital and analog inputs. Special attention must be paid to wiring with the correct terminal. In this example, terminal 1 is the light barrier 1, terminal 3 is the light barrier 2, terminal 5 is the push button switch and terminal 7 is the proximity switch. No shielded cable is used for this purpose. The connection is made according to chapter 5.5.2 "Connection of the digital inputs".

In this example the pressure sensor supplies a voltage signal of 0 V - 10 V. For this reason, both switches next to the analog terminals on the controller are set to 10 V (see Chapter 5.5.3 "Analog Inputs (only ZX09A/D/E and ZX21TPA/TCA)").

Step 5 is omitted, since the application does not provide an encoder.

13.6 Step 6: Controlling the function of the PLC via the LEDs

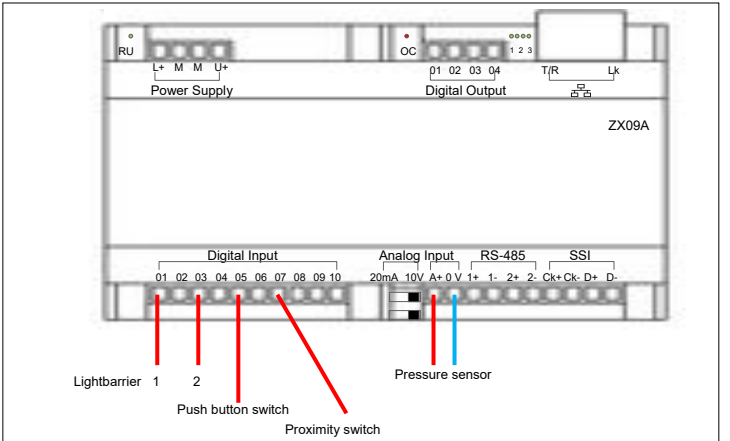


Fig. 32: Connection of the inputs

After successful programming and connection of the inputs, the function is controlled via the LEDs next to the digital outputs. Step 5 is skipped because no encoder is connected.

Step 7 is omitted because no data communication is provided in the application.

13.7 Step 8: Connecting the outputs

Fig. 33 shows the connection of the digital outputs. As with the connection of the inputs, make sure that the cabling is connected to the correct terminals. The control of the filling process must therefore be connected to terminal 1, the control of volume 1 to terminal 2, the control of volume 2 to terminal 3 and the error status display to terminal 4. The connection is made as described in chapter 5.6.1 "Connection of the digital outputs".

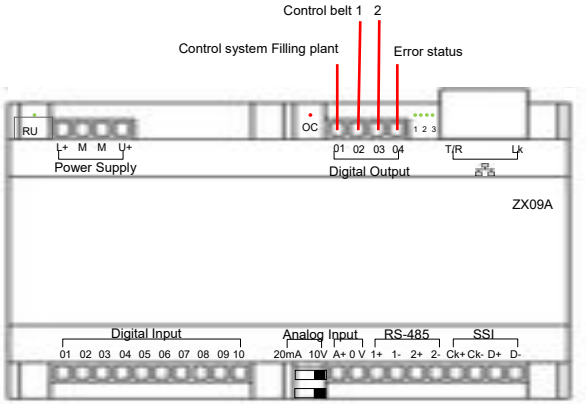


Fig. 33: Connection of the outputs

13.8 Step 9: Connecting the power supply of the digital outputs

Finally the power supply of the digital outputs is connected. The voltage supply of the outputs is connected as described in chapter 5.6.1 "Connection of the digital outputs".

14. Example 2

In this example, the steps of initial commissioning (see section 8.2) are carried out for a specific example. Starting with the development of a program in step 1, the individual steps are carried out until the system is fully operational after completing step 9.

14.1 The process

In this example the control of a good/bad part sorting shall be shown. Figure 34 shows the process to be automated. Component parts are delivered via a conveyor belt. At position 1 these parts are to be checked for their maximum height. For this purpose a light barrier is to be used to check whether the individual components exceed the specified maximum size. If this is the case, the defective products are to be sorted out at position 2 via a trap door. To ensure that only the faulty component is sorted out, a light barrier at position 2 should monitor the passing of the component. If the maximum height is not exceeded, the components should be transported further to position 3. Here a colour sensor is to be used to check the component for the correct colour. Similar to position 2, the faulty component should be sorted out at position 4. If the component meets the specified conditions, it should be conveyed to another machine at position 5.

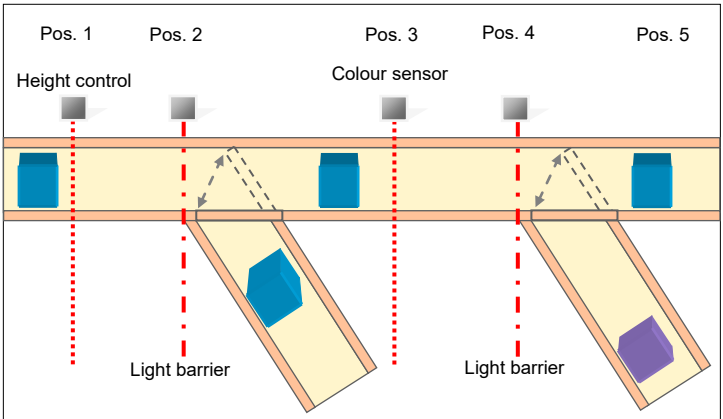


Fig. 34. Principle Diagram of good/bad part sorting

14.2 Step 1: Program creation in EX_PRESS 5

In the following the program for application is shown. In the first part of the program, marked by "DECLARATION", all necessary variables are declared.
Under "VAR_INPUT" 4 digital input variables for the four sensors are declared for this purpose. Here the variable "High" corresponds to the sensor at position 1, "Light1" to the light barrier at position 2, "Color" to the sensor at position 3 and "Light2" to the light barrier at position 4. These are assigned to specific input terminals, namely 1, 2, 3 and 4, via the command "AT".
The output signals are declared under "VAR_OUTPUT". This is the control of trap door 1 and 2, "Door1Open" and "Door2Open". If these are active, the components are sorted out at the respective position.
Under "VAR" the variables within the program are declared. These are two status variables "Fail1" and "Fail2", which activate the Controlling trap doors. Under "VAR_TIMER" all required timers of the program are declared. A total of 4 timers are required for these, two for each sorting step.
The actual program is written in the logic part of the program, marked by "LOGIC-PART". For this, the signals of the light barriers are first debounced with 200 ms. This ensures that only passing components activate the light barrier.

For this purpose the signals of the light barriers are first debounced with 200 ms. This ensures that only passing components activate the light barrier.
The status bit of the respective sorting is now set via an IF query. If the component is too high and passes the light barrier, Fail1 is set to 1, otherwise Fail1 is set to 0. A similar condition is set for the second sorting. If the color is wrong and the light barrier is passed, Fail2 is set to 1, otherwise Fail2 is set to 0.
Finally, the output signals are assigned. For this, the outputs are switched over a defined period of time.
In the last part of the program, marked by "TIMERS", the different timers are defined. Via "timerDoor1" the first trap door is controlled. As soon as Fail1 is activated, the trap door is opened for a period of 1200 ms and then closed again.
timerDoor2" is defined in the same way. "timerHigh" and "timerColor" are timers that maintain a faulty sorting signal until the corresponding component arrives at the trap door.

```
PROGRAM Sorting
PLC_NAME = "Good/Bad";
PLC_TYPE = "ZX20T";

(************************DECLARATION*******************)

VAR_INPUT
                High AT In_01;
                Light1          AT In_02;
                Colour          AT In_03;
                Light2          AT In_04;
END_VAR;

VAR_OUTPUT
                Door1Open       AT Out_01;
                Door2Open       AT Out_02;

END_VAR;

VAR
                Fail1;
                Fail2;
END_VAR;

VAR_TIMER
                timerHigh;
                timerColor;
```

```
                timerDoor1;
                timerDoor2;
END_VAR;

(*************************LOGIC-PART*******************)
(*Debouncing of the input signals of the light barriers*)
Light1.TDB:=200ms;
Light2.TDB:=200ms;

(*If 1. the light barrier 2 is activated and 2. the component is detected as too high, Fail1 is set*)

IF Light1=1 AND timerHigh=1 THEN
                Fail1:=1;
ELSE
                Fail1:=0;
END_IF;

(*If 1. the light barrier 2 is activated and 2. the component had the wrong colour, Fail2 is set*)

IF Light2=1 AND timerColor=1 THEN
                Fail2:=1;
ELSE
                Fail2:=0;
END_IF;


Door1Open:=timerDoor1;
Door2Open:=timerDoor2;

(*************************TIMERS*******************)

(*As soon as Fail1 is active, "timerDoor1" is restarted and delivers a high signal for 1200 ms*)

timerDoor1.ENABLE:=1;
timerDoor1.RESET:=Fail1;
timerDoor1.POL:=HIGH;
timerDoor1.MODE:=SINGLE_SHOT;
timerDoor1:=1200ms;

(*As soon as Fail2 is active, "timerDoor2" is restarted and delivers a high signal for 1200 ms*)

timerDoor2.ENABLE:=1;
timerDoor2.RESET:=Fail2;
timerDoor2.POL:=HIGH;
timerDoor2.MODE:=SINGLE_SHOT;
timerDoor2:= 1200ms;

(*If the component is detected as too high, "timerHigh" is started and holds the value until the light
barrier 1* is reached)

timerHigh.ENABLE:=1;
timerHigh.RESET:=High;
```

```
timerHigh.POL:=HIGH;
timerHigh.MODE:=SINGLE_SHOT_SE;
timerHigh:=1s;

(*If the component is recognized as too wrongly colored, "timerColor" is started and keeps the value
until the light barrier 2* is reached)

timerColor.ENABLE:=1;
timerColor.RESET:= Color;
timerColor.POL:=HIGH;
timerColor.MODE:=SINGLE_SHOT_SE;
timerColor:=1s;


END_PROGRAM;
```

14.3 Step 2: Connecting the power supply

As described in section 5.4 "Electrical connection of the device", the power supply is connected to the controller.

14.4 Step 3: Programming the PLC

For programming, the controller is connected to a computer via the Ethernet interface. The PLC is programmed using the EX_PRESS 5 software. After successfully compiling and fitting the program code, it can be transferred to the control (see chapter 1.6.2 "Calling the programming tool" in the EX_PRESS 5 manual).

14.5 Step 4: Connecting the inputs

Fig. 35 shows the connection of the digital inputs. Special attention must be paid to the wiring with the correct terminals. In this example the height sensor is connected to terminal 1, the first light barrier to terminal 2, the colour sensor to terminal 3 and the second light barrier to terminal 4. No shielded cable is used. The connection is carried out according to chapter 5.5.1 "Connection of the digital inputs".

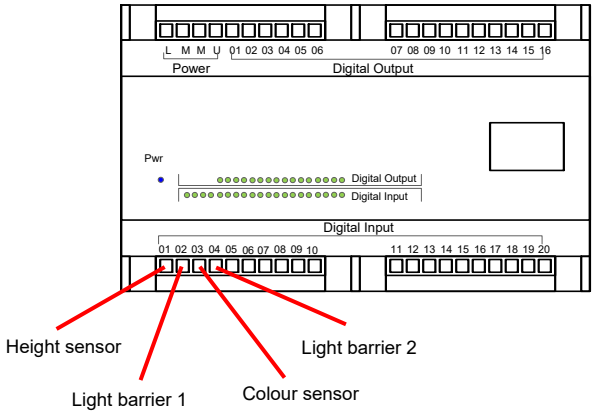


Fig. 35: Connection of the inputs

Step 5 is omitted, since the application does not provide an encoder.

14.6 Step 6: Controlling the function of the PLC via the LEDs

After successful programming and connection of the inputs, the function is checked via the LEDs on the controller. Step 6 is skipped because no network is connected.

Step 7 is omitted because no data communication is provided in the application.

14.7 Step 8: Connecting the digital outputs

Fig. 36 shows the connection of the digital outputs. As with the connection of the inputs, make sure that the cabling is connected to the correct terminals.
Thus, the control from the first trapdoor must be connected to terminal 1 and the control from the second trapdoor to terminal 2. The connection is made as described in chapter 5.6.1 "Connecting the digital outputs".

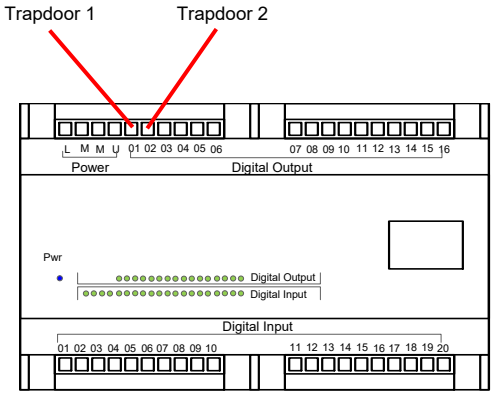


Fig. 36: Connection of the outputs

14.8 Step 9: Connecting the power supply of the digital outputs

Finally the power supply of the digital outputs is connected. The voltage supply of the outputs is connected as described in chapter 5.6.1 "Connection of the digital outputs".

15. Example 3

In this example, the steps of initial commissioning (see section 8.2) are carried out for a specific example. Starting with the development of a program in step 1, the individual steps are carried out until the system is fully operational after completing step 9.

15.1 The application

In this example, the transmission of measured sensor values for display on an HMI display device is shown. Fig. 37 shows the setup provided for this purpose. A ZX09A and an HMI MVisio are connected to each other via an Ethernet switch. In addition, a PC is connected to this switch for programming the devices. The analog input of the ZX09A is connected to the measured value output (0-10V) of a distance sensor.The analog signal will be continuously sampled in the ZX09A and queried for display by the HMI via a Modbus TCP connection. In this system, the HMI forms the Modbus master and the ZX09A the Modbus slave.

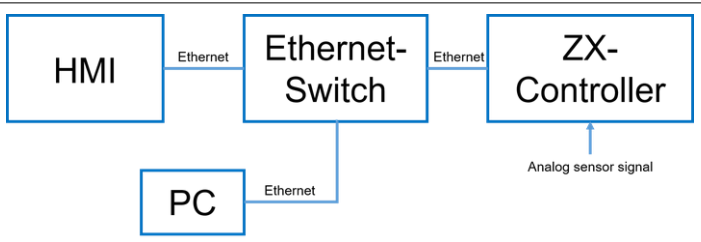


Fig. 37: Principle diagram of the connection between ZX controller and HMI

15.2 Step 1: Program creation in EX_PRESS 5

In the following the program for this application is shown. In the first part of the program, marked by "CONNECTION", the Modbus TCP connection is configured. For this purpose it is defined under NETWORK_TYPE that the device shall be a Modbus slave. Then the IP address, the subnet mask and the standard gateway are assigned and activated. With the entry DHCP_ACTIVE=FALSE the address assignment via a DHCP server is deactivated. More information about the configuration of the Modbus connection under Chap. 7.1.1 "Modbus/TCP".

Subsequently, all necessary variables are declared in the program, marked by "DECLARATION". Under "VAR_ADC" the input variable for the analog distance sensor is declared. Under "VAR_GLOBAL" the variables are declared that are sent to the master via the Modbus connection. These are five variables "MValX", which contain the actual measured values and the variable "PNo", which contains a packet number. The program-internal variables are declared under "VAR". These are the intermediate variables "AnaX", which transfer the analog measured values to the network variables "MValX". In addition, a "counter" is declared, which continuously assigns the analog values to the corresponding variables. The variables "varbit" and "resetbit" activate and deactivate the value transfer clock-controlled. Under "VAR_TIMER" the required timer of the program is declared.

In the logic part of the program, marked by "LOGIC-PART" the actual program is written. At first the reset is assigned to the variable varbit by the variable resetbit. Then the value transfer is done clock-controlled in the IF-condition.
In the case statement, the current analog measured value is passed to the five variables "AnaX". The counter variable is increased continuously by 1 during this time. If the counter variable reaches the value 5, the measured values stored in the intermediate variables "AnaX" are transferred to the network variables. In addition, the packet number "PNo" is incremented by 1 in this step and the reset is activated by equating the resetbit variable with the varbit variable. Thus, a packet with five analog measurement values and an associated packet number is now made available to the Modbus network. In the ELSE condition of the value transfer, the reset variable is then reset and, when a packet number of 30 is reached, this is also reset.

In the last part of the program, marked by "CLOCK", the used timer "timer1" is defined. Via this the transfer of the analog measuring value is activated. As soon as the timer is activated, it is HIGH for maximum 0.5ms, until the value transfer is finished and it becomes LOW again by the RESET. Finally the clock of timer1 is assigned to the used variables.

```
PROGRAM Measured_value_transfer
PLC_NAME = "Measured_value_transfer";
PLC_TYPE = "ZX09A";

(************************CONNECTION*******************)
(*Configuration of the Modbus TCP Connection*)
NETWORK_TYPE = MODTCP_SLV;
IP_ADR = 192.168.15.42;
IP_MASK = 255.255.255.0;
IP_GATE = 192.168.15.21;
IP_ACTIVE = TRUE;
DHCP_ACTIVE = FALSE;
VAR_COMM = FALSE;
VAR_COMM_CYCLE = 200;


(************************DECLARATION*******************)
VAR_ADC
                ANALOG_VALUE;
END_VAR;

VAR_GLOBAL
                MVal1: WORD;
                MVal2: WORD;
                MVal3: WORD;
                MVal4: WORD;
                MVal5: WORD;
                PNo: WORD;
END_VAR;

VAR
                Ana1: INT;
                Ana2: INT;
                Ana3: INT;
                Ana4: INT;
                Ana5: INT;
                counter: INT;
                varbit: BIT;
                resetbit: BIT;
END_VAR;

VAR_TIMER
                timer1;
END_VAR;
```

```
(*************************LOGIC-PART*********************)
varbit := 1;
varbit.RE := resetbit;

IF( varbit = 1 ) THEN
                CASE counter OF
                    0: Ana1 := ANALOG_VALUE;
                    1: Ana2 := ANALOG_VALUE;
                    2: Ana3 := ANALOG_VALUE;
                    3: Ana4 := ANALOG_VALUE;
                    4: Ana5 := ANALOG_VALUE;
                END_CASE;
                counter := counter+1;
                IF(counter = 5) THEN
                    counter := 0;
                    MVal1 := Ana1;
                    MVal2 := Ana2;
                    MVal3 := Ana3;
                    MVal4 := Ana4;
                    MVal5 := Ana5;
                    PNo := PNo+1;
                    resetbit := varbit;
                END_IF;
ELSE
                resetbit := FALSE;
                IF (PNo = 30) THEN
                    PNo := 0;
                END_IF;
END_IF;

(*************************CLOCK*********************)
timer1.MODE := CONT;
timer1 := 1 ms;
timer1.RESET := 0;
timer1.ENABLE := 1;
timer1.POL := LOW;
Ana1.CLK := timer1;
Ana2.CLK := timer1;
Ana3.CLK := timer1;
Ana4.CLK := timer1;
Ana5.CLK := timer1;
MVal1.CLK := timer1;
MVal2.CLK := timer1;
MVal3.CLK := timer1;
MVal4.CLK := timer1;
MVal5.CLK := timer1;
PNo.CLK := timer1;
counter.CLK := timer1;
varbit.CLK := timer1;
resetbit.CLK := timer1;

END_PROGRAM;
```

15.3 Step 2: Connecting the power supply

As described in section 5.4 "Electrical connection of the device", the power supply is connected to the controller.

15.4 Step 3: Programming the PLC

For programming, the controller is connected to a computer via the Ethernet interface. The PLC is programmed using the EX_PRESS 5 software. After successfully compiling and fitting the program code, it can be transferred to the control (see chapter 1.6.2 "Calling the programming tool" in the EX_PRESS 5 manual).

15.5 Step 4: Connecting the inputs

Fig. 38 shows the connection of the analog input. Particular attention must be paid to the position of the DIP switches. The connection is carried out according to chapter 5.5.4 "Connection of the analog inputs".

Step 5 is omitted because no encoder is provided in the application.

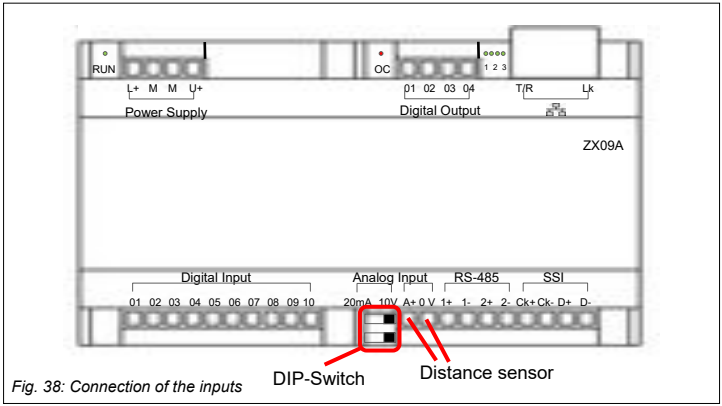


Fig. 38: Connection of the inputs

Step 6 is omitted, since no control via the digital outputs is provided.

15.6 Step 7: Connecting the Ethernet interface

Fig. 39 shows the connection of the Ethernet connection. The connection is carried out according to chapter 7.1.1 "Modbus/TCP".

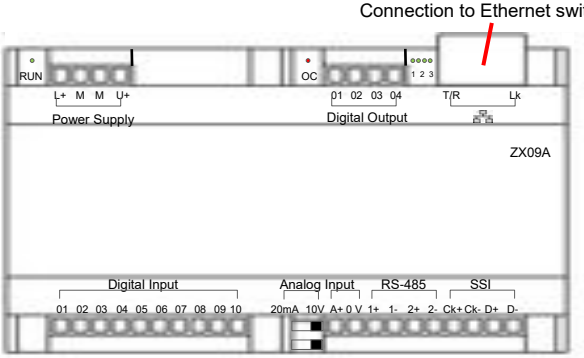


Fig. 39: Connecting the Ethernet connection

Steps 8 and 9 are omitted, since no control via the digital outputs is intended.

16. Service

Contact us in case of service:
H. Zander GmbH & Co. KG
Am Gut Wolf 15
52070 Aachen
Germany

Service phone:
Within Germany:
(0241) 910501-0
From abroad:
+49 241 910501-0
E-Mail:
info@zander-aachen.de
Internet:
www.zander-aachen.de