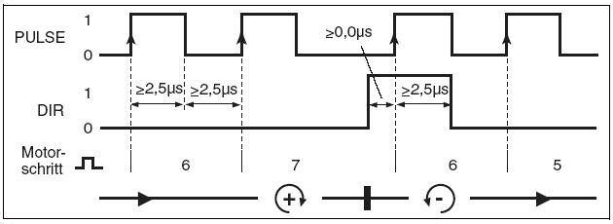
Twido and LEXIUM05/CPD17 via Pulse-Direction-Interface

Control of Speed and Position without using a Field bus



This document describes how to build a small position module using a TWIDO PLC and LEXIUM05/CPD17 drive including the wiring and a program example.

Designed 07.06.2005 by

Patrick Friedmann
Berger Lahr GmbH
Customer Support Center - Technical Support
Gewerbestraße 9
D-77749 Hohberg-Niederschopfheim.
Tel.: +49 7808 / 943-235
Fax: +49 7808 / 946-58 235
E-Mail: mailto:patrick.friedmann@berger-lahr.com

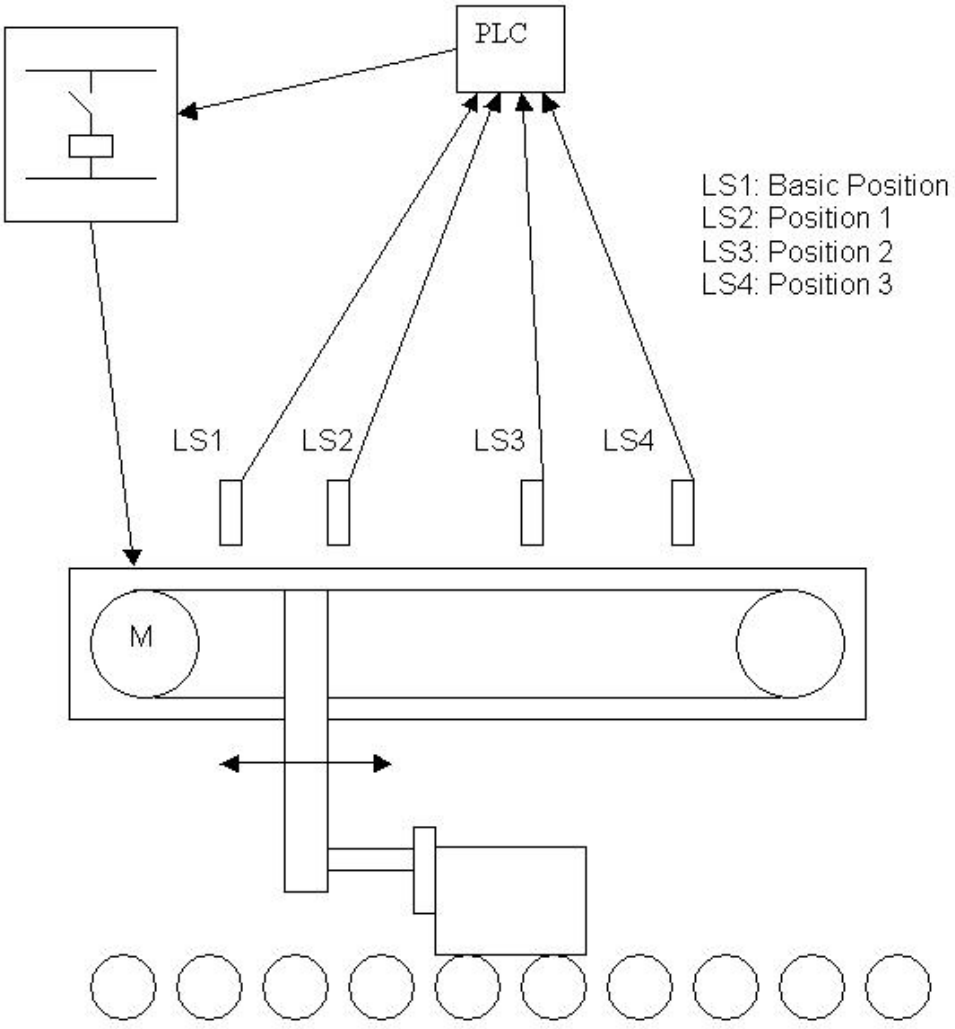
a company of Schneider Electric

BERGER LAHR

Typical Application: Pusher on a conveyor

A pusher moves material on a conveyor. The positions are defined with limit switches. The PLC stops the movement when the Pusher reaches the limit switches.

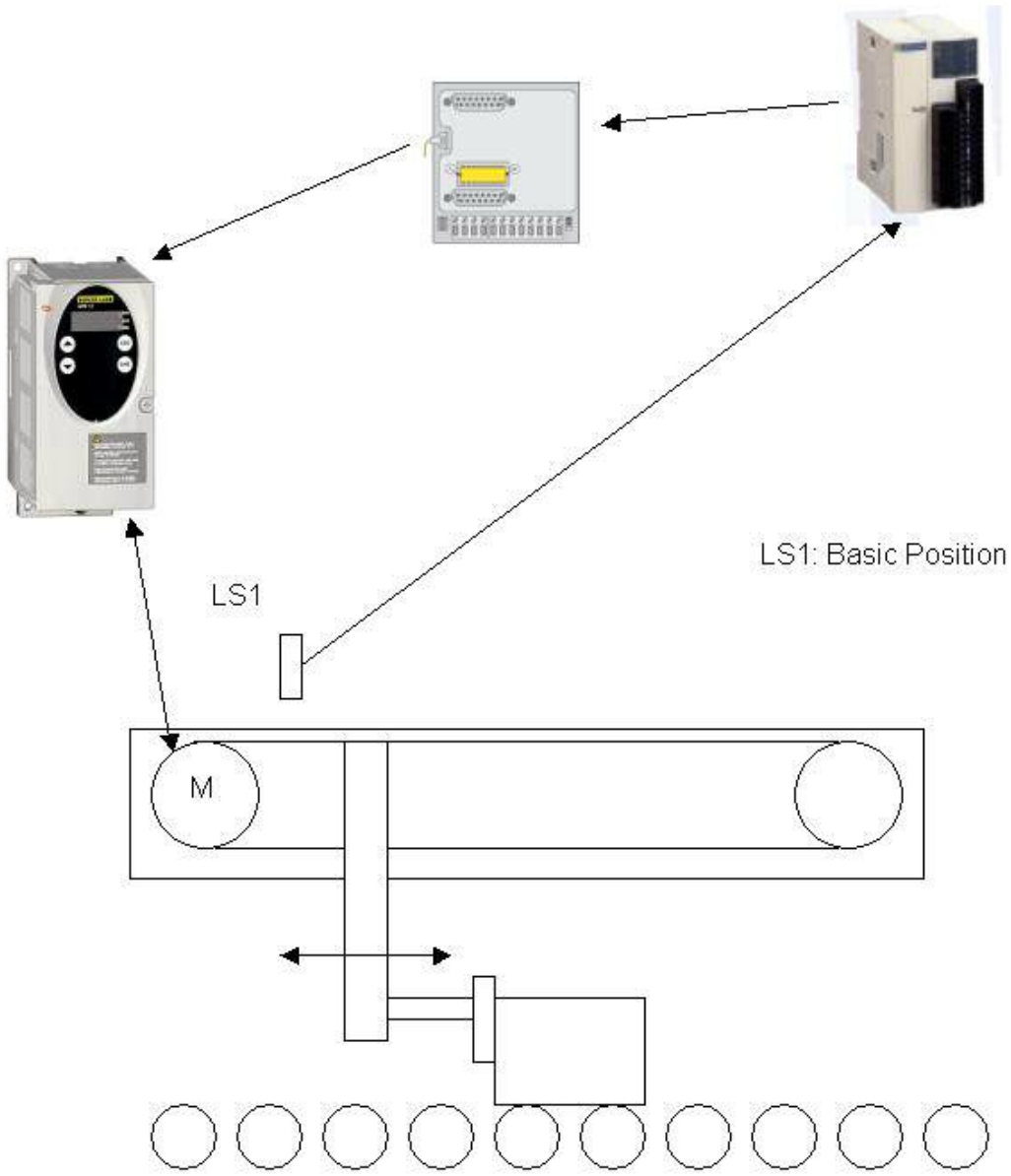
- Positions can only be adjusted by moving the limit switches
- Speed control with using a frequency inverter: Speed is only changeable between fixed values or continuous with additional hardware (analogue PLC outputs)
- Speed control without using a frequency inverter is not possible (exception: pole-changing motor)
- Complex hardware installations



Solution with TWIDO, USIC and LEXIUM05/CPD 17

LEXIUM05/CPD17 is a small drive with a high performance. It can be used as an “electronic gear” via the pulse/direction interface. One pulse at the interface moves the motor one increment.
TWIDO is a small PLC with fast transistor outputs. Its configuration tool provides a standard function block that generates a sequence of pulses.
USIC converts the PLC output signals to a pulse/direction signal.

- Positions can be adjusted in the PLC program (optional with HMI)
- Speed control on the drive, set values from the PLC (optional with HMI)
- **No field bus needed!**



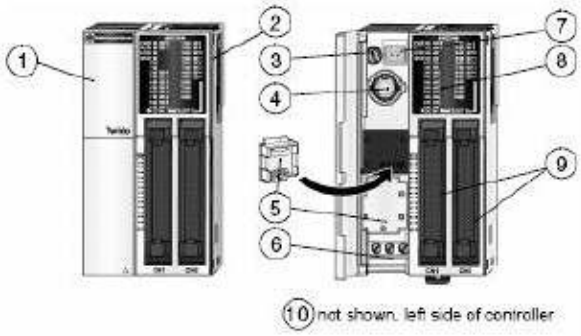
a company of Schneider Electric

BERGER LAHR



Components

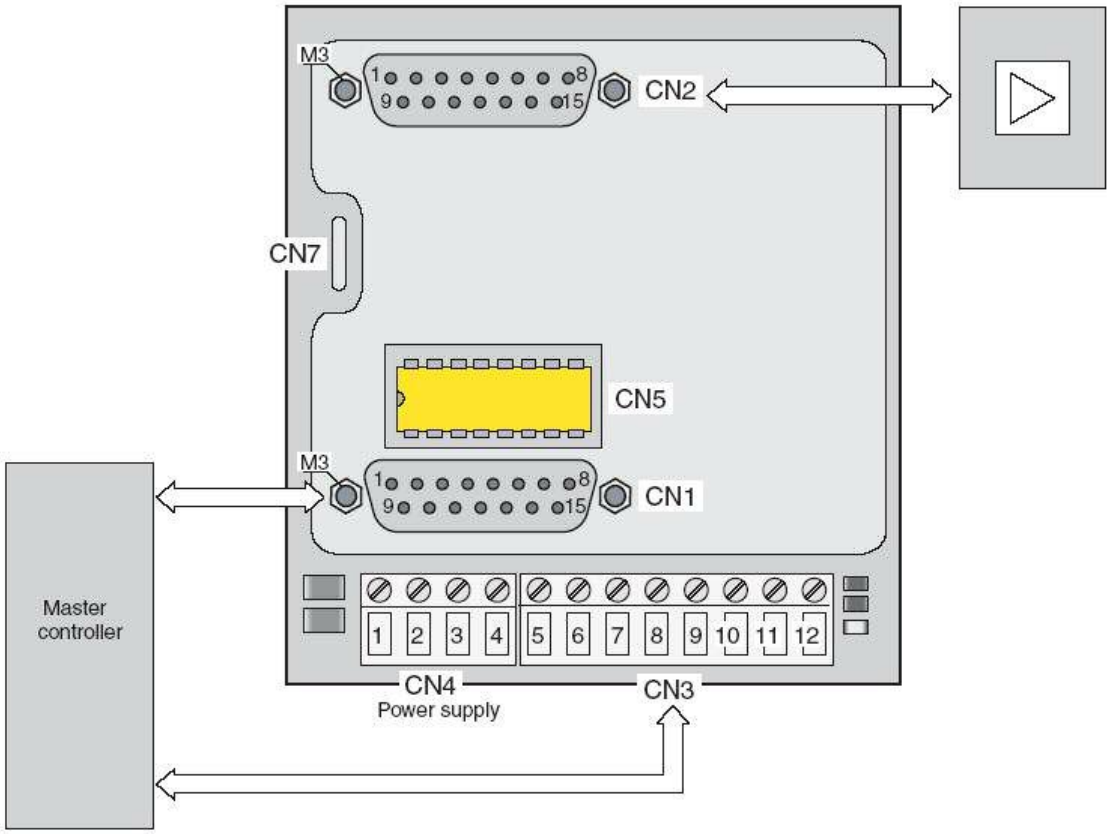
TWIDO Modular TWDLMDA20RT



Label	Description
1	Hinged lid
2	Expansion connector
3	Analog potentiometer
4	Serial port 1
5	Cartridge covers
6	24 VDC power supply terminals
7	Analog voltage input connector
8	LEDs
9	I/O terminals
10	Communication connector

16 Digital Inputs Sink/Source
8 Digital Outputs (6 Relays and 2 Transistor source)
Expandable with 7 additional modules

USIC (Universal Signal Interface Converter) GEA3EC001



The USIC (Universal Signal Interface Converter) is an interface adapter, which is used as a universal adapter for a pulse/direction interface to a PLC.



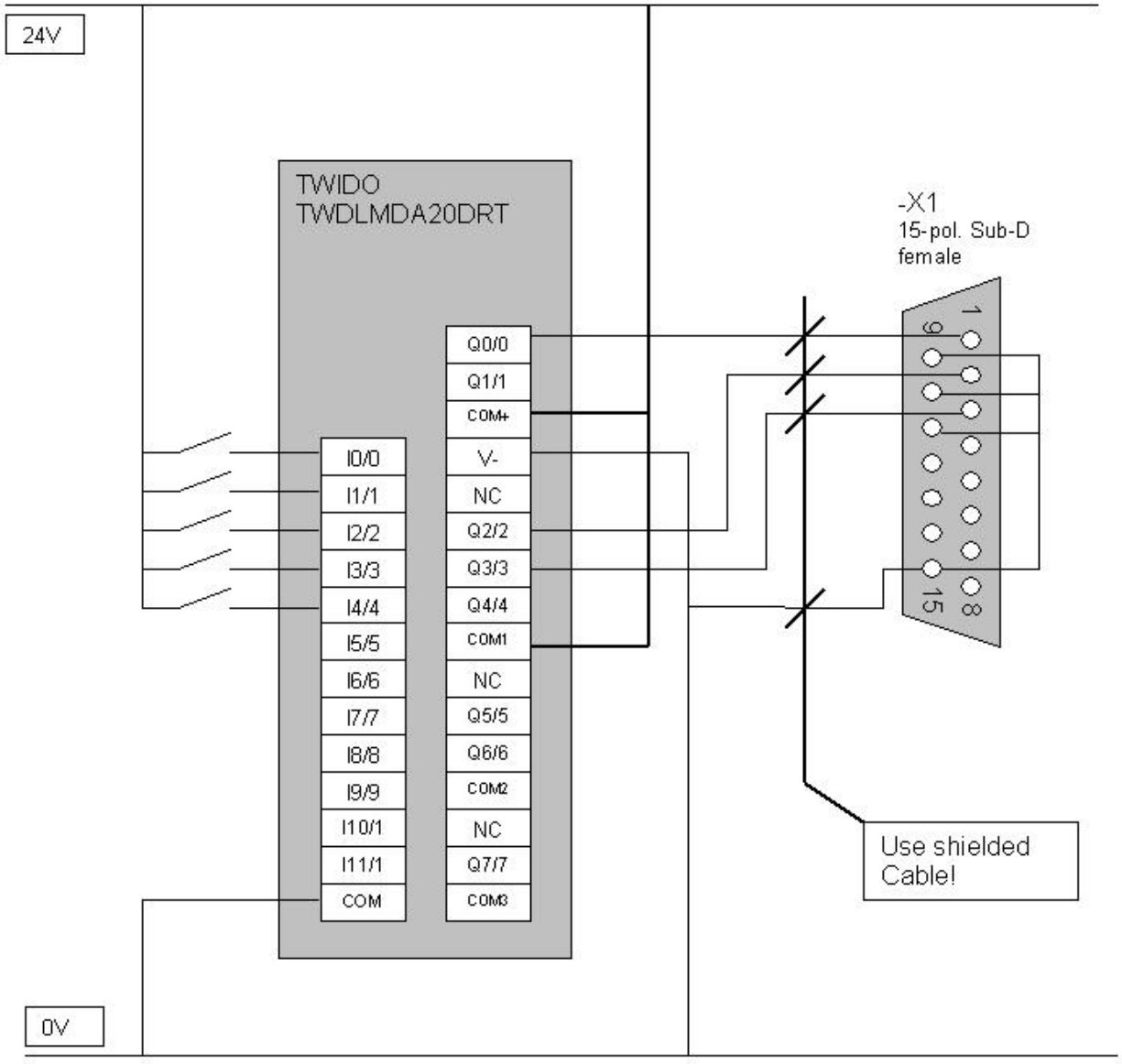
LEXIUM05/CPD17 Drive

A universal drive with:

- Auto tuning on board, manual optimisation of the control loop with PS 2
- +/- 10V analogue signal interface
- Pulse/direction and A/B signal interface
- Field bus interface CANopen DS402 and ModBus RTU,
  optional (in the future): Profibus DP
- Safety stop integrated in the drive
- 4 sizes
- Standard motors & drives combination are defined

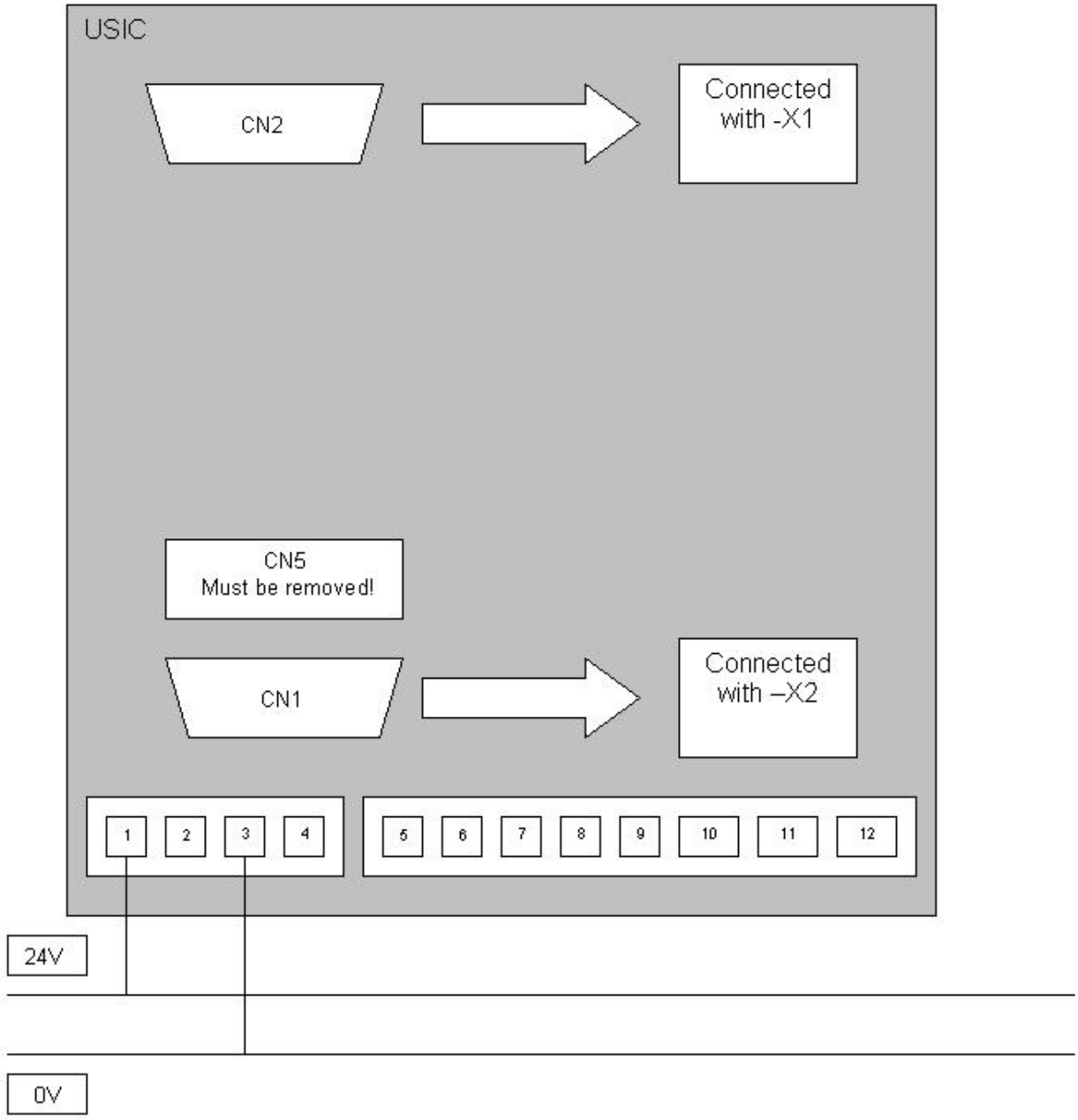
Wiring

TWIDO

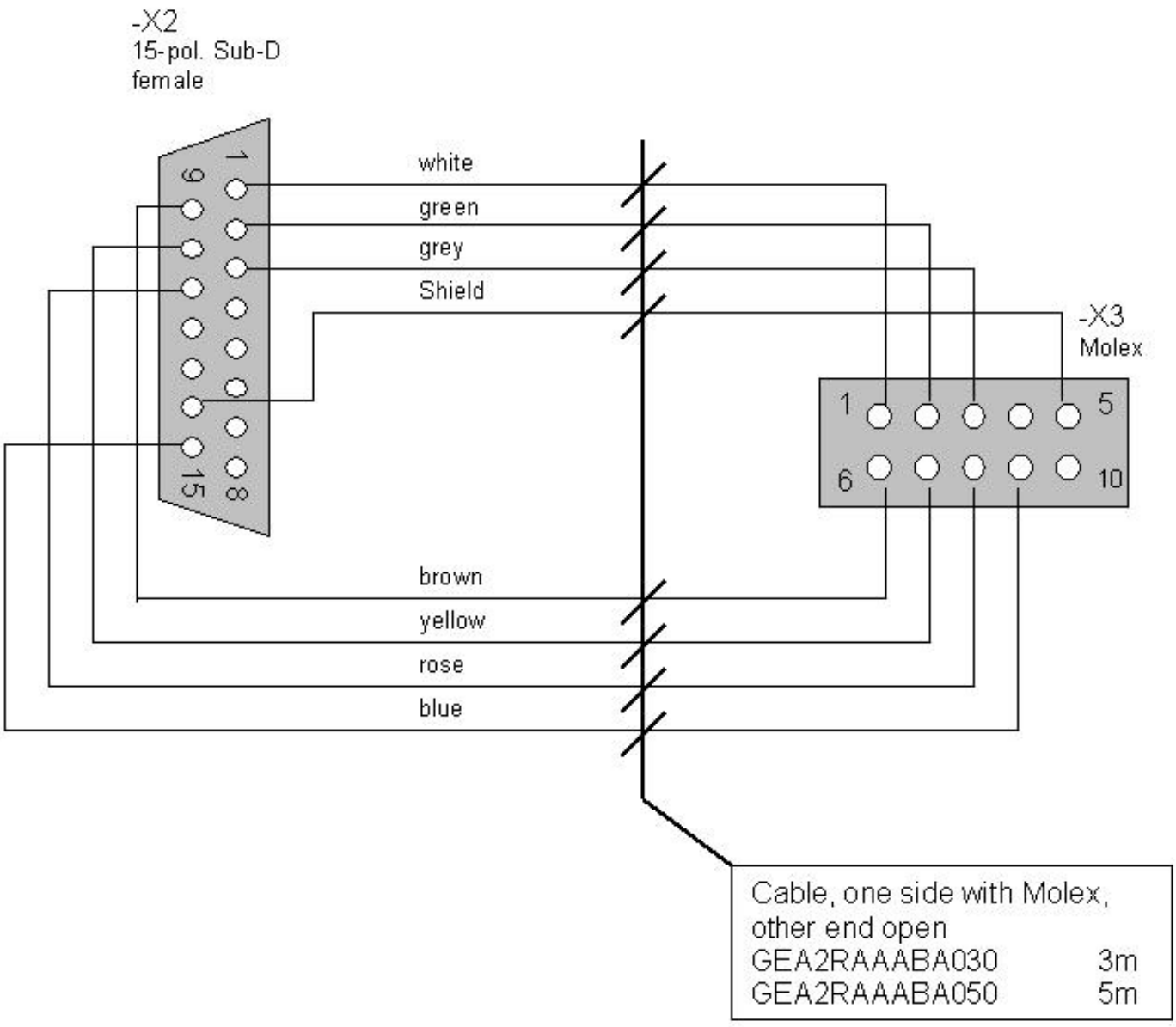


I0	Run CPU	Q0	Pulses
I1	Step mode/ continuous mode	Q2	Direction
I2	Start	Q3	Enable
I3	Enable		
I4	Direction		

USIC

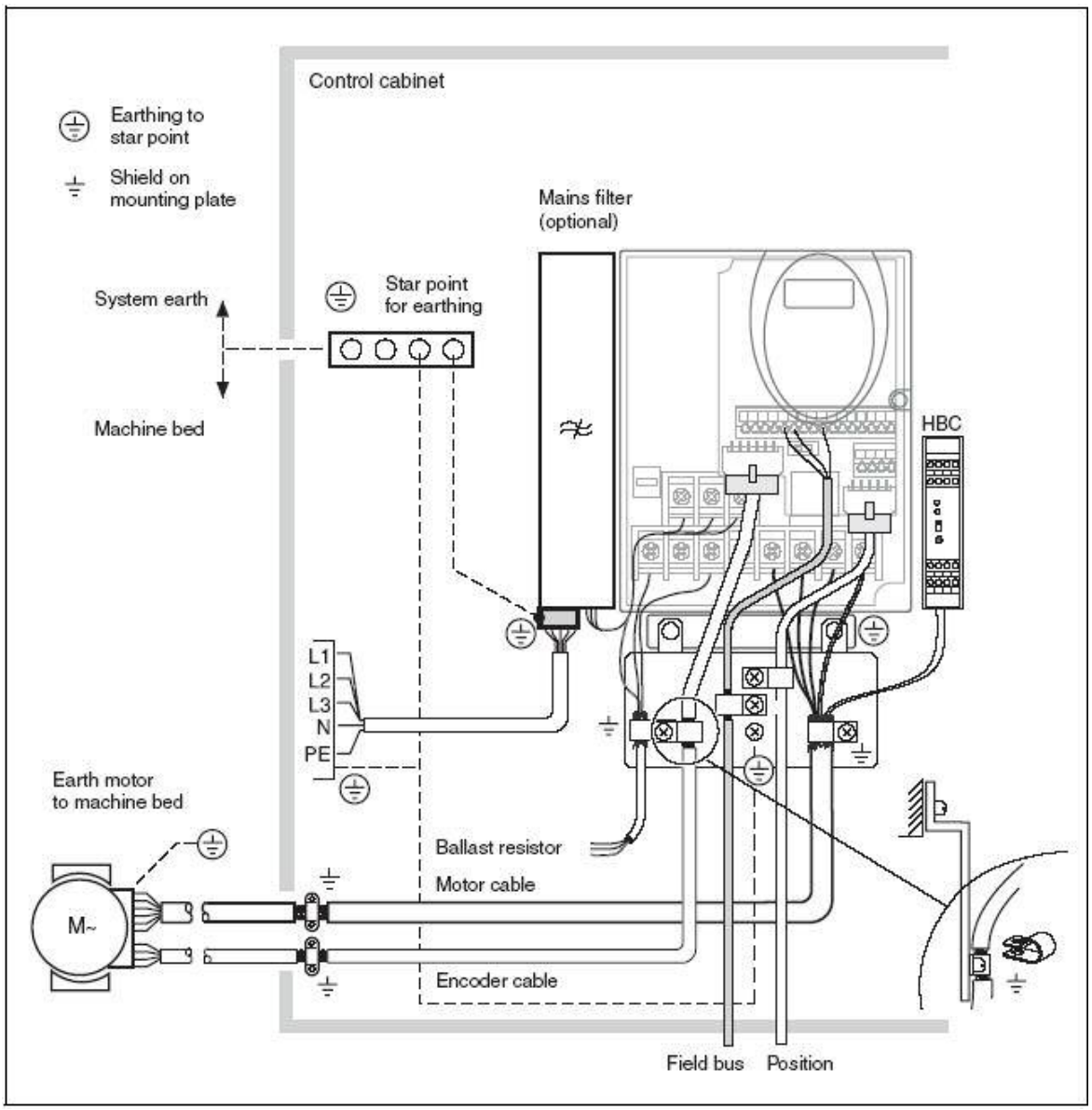

USIC
CN2
Connected with -X1
CN5
Must be removed!
CN1
Connected with –X2
1
2
3
4
5
6
7
8
9
10
11
12
24V
0V

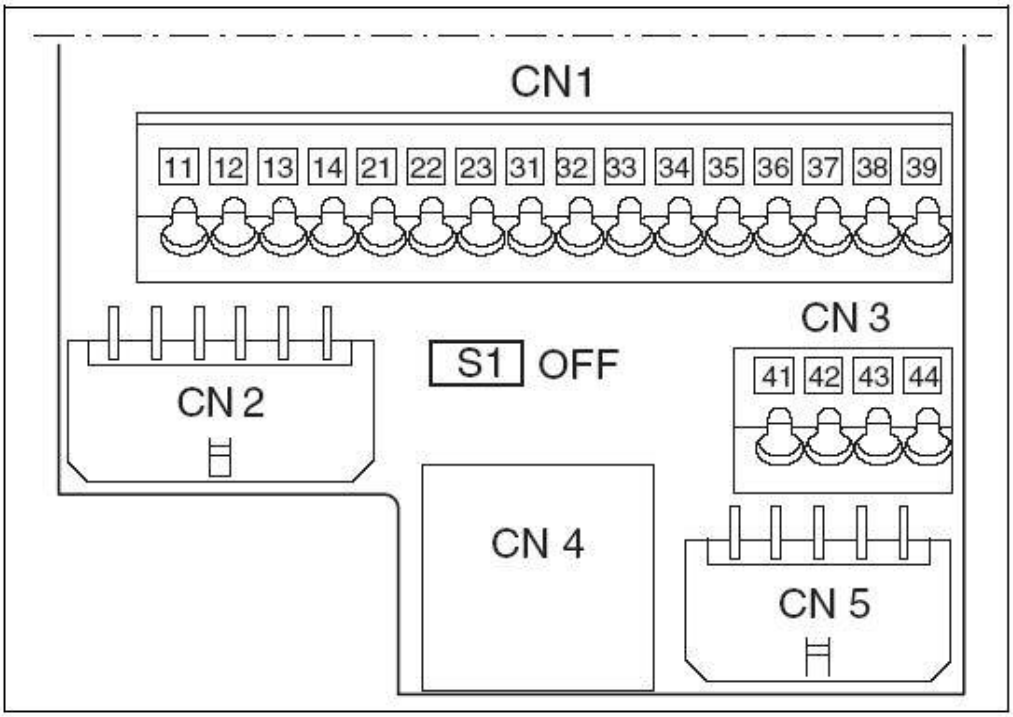
Connection cable between USIC and LEXIUM05/CPD17



LEXIUM05/CPD17

Overview





CN1	Pin 11-14	Analogue Inputs	here not used
CN1	Pin 21-23	Can Open	here not used
CN1	Pin 31	Output “No Fault”	optional
	Pin 32	Output for HBC holding break controller	optional
	Pin 33	not used	
	Pin 34	Input Reset	optional
	Pin 35	Input Enable	here not used
	Pin 36	Input /Halt (Function Stop)	must be connected
	Pin 37	Input /Safe_Disable_A (Emergency Stop)	must be connected
	Pin 38	Input /Safe_Disable_B (Emergency Stop)	must be connected
	Pin 39	24V for Pin 37+38 if not used	optional

CN2 Connection of motor sensor

CN3	Pin 41+42	0VDC	must be connected
	Pin 43+44	24VDC	must be connected

CN4 Connection to PC or peripheral control terminal

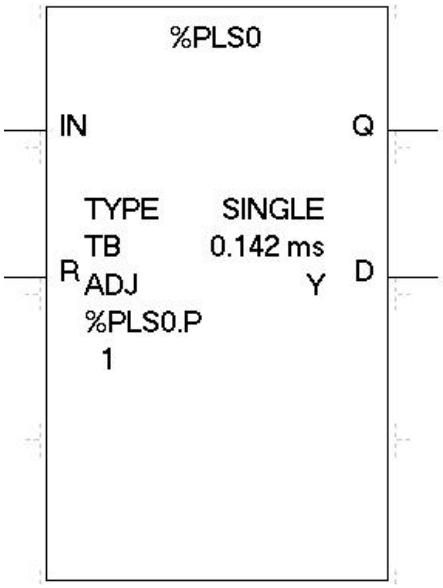
CN5 pulse/direction interface connected with –X3

Connection of motor phases and supply voltages depends on the size of the drive. Use the manuals!

Implementation

Twido

Twidosoft, the configuration software tool, provides two impulse generator function blocks. They allow generating a number of pulses with several frequencies.



Function	Object	Description
Time base	T_B	0,142ms, 0,57ms, 10ms or 1s
Pre-adjustable period	%PLS.P	Basic setting
Number of pulses	%PLS.N %PLS.ND	defines the number of pulses. If %PLS=0 an unlimited number of pulses is generated
Adjustable	Yes/No	If Yes the parameter %PLS.P is adjustable from the program
Start	IN	Starts the generation of pulses
Reset	R	Resets the outputs and the number of the generated pulses
Pulses active	Q	Pulse generation is active
Finished	D	Pulse generation is ready

The function block %PLS0 uses the output Q0.0 as a default setting. The relation between T_{on}/T_{off} is always 50%.

In the most cases it will be helpful to use the smallest Time base.

$T_B = 0{,}142ms$

Then %PLS.P is adjustable between 1 and 255. That leads to the minimum and maximum frequencies:

$$f_{min} = \frac{1}{255 \cdot 0{,}142 \cdot 10^{-3} s} = 27{,}6Hz$$

$$f_{max} = \frac{1}{1 \cdot 0{,}142 \cdot 10^{-3} s} = 7042{,}3Hz = \frac{1}{0{,}142 \cdot 10^{-3} s} \cdot \frac{60s}{\min} = 422.535\,\min^{-1} = 422.535\frac{Inc}{\min}$$

The resulting speed on the motor depends also on the adjusted gear factor GF and the internal resolution IR of the drive. The internal resolution is the number of increments for one revolution of the motor. For the LEXIUM05/CPD17 the internal resolution is:

$IR = 131072Inc$

The resulting speeds are:

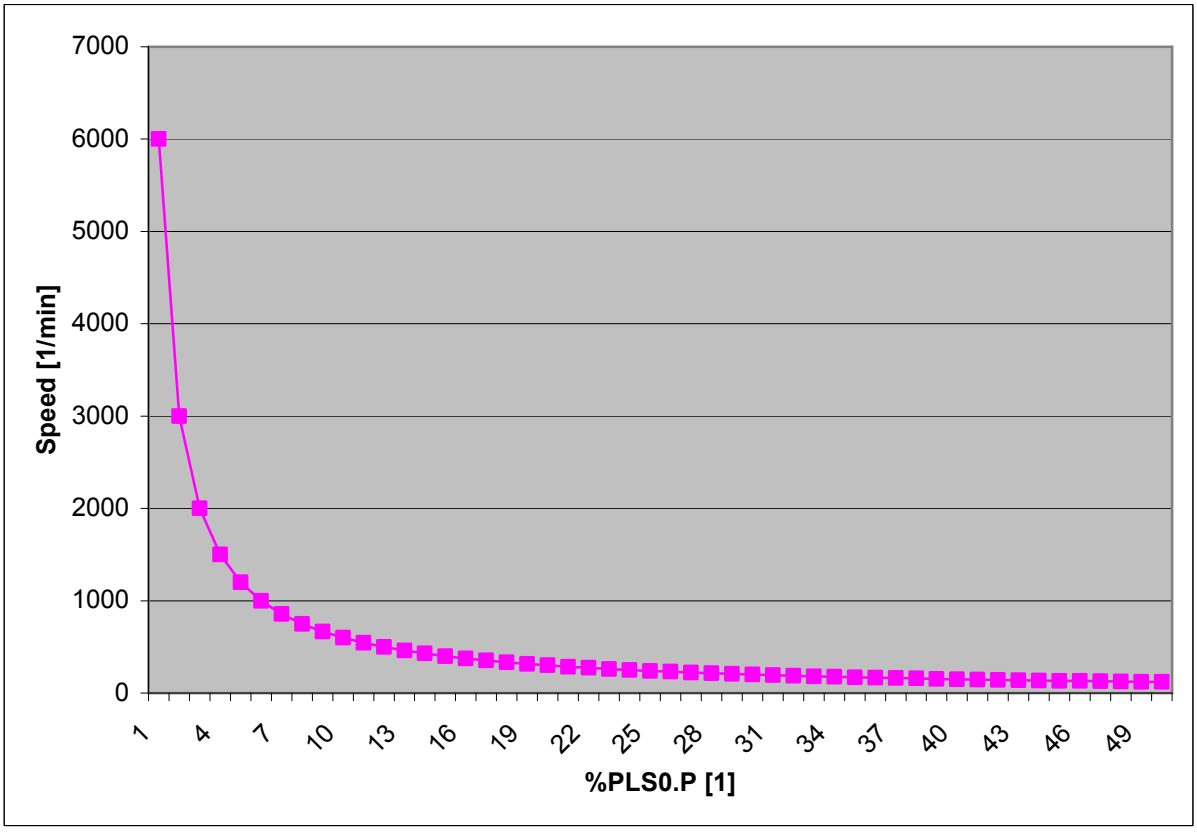
$$n_{max} = \frac{f\,\max \cdot GF}{IR} = \frac{422535\frac{Inc}{\min} \cdot GF}{131072Inc}$$

$$n = \frac{n_{max}}{\%PLS0.P}$$

Now an example. The maximum speed should be 6000min^{-1}. Then the gear factor is

$$GF = \frac{n_{\max} \cdot IR}{f\ \max} = \frac{6000\frac{1}{\min} \cdot 131072 Inc}{422535\frac{Inc}{\min}\cdot} = 1861{,}24$$

and the speed is changeable in the steps as shown in the following diagram.

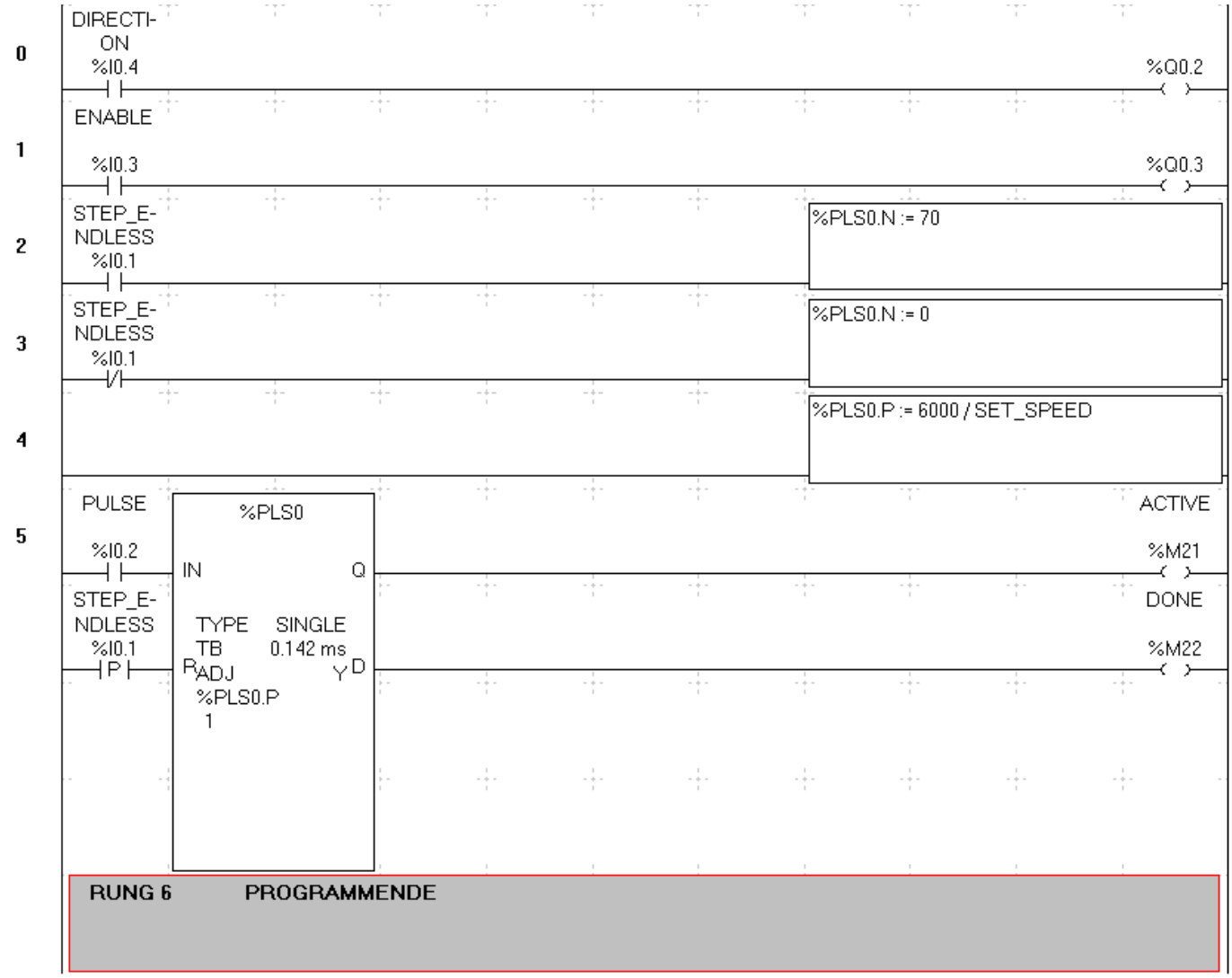


The stroke per revolution of the motor corresponds to

$$U = \frac{IR}{GF} = \frac{131072 Inc}{1861{,}24} = 70{,}4 Inc$$

That means if the TWIDO generates 70 Pulses, the motor makes about one resolution.

The software is quite easy. Here is an example:



Input 0.1 changes between step mode and continuous mode.
If %I0.1 is true and a positive flag on %I0.2 TWIDO generates 70 pulses and the motor makes about one revolution.
If %I0.1 is false and a positive flag on %I0.2 TWIDO generates an endless number of pulses and the motor is turning continuously.

Input 0.2 a positive flag on %I0.2 starts the generation of pulses.

Input 0.3 enables the drive

Input 0.4 switches the direction of the motor

%MW10 SET SPEED is the set value of speed. Here it is changeable in the animation table.

The commissioning of the drive is also very simple. With the first time of switch on the power supply of the drive with the connected motor, the drives reads automatically the motor data from the hyperface sensor (motor sensor).

Then it is necessary to do the “First set up” via the integrated HMI.

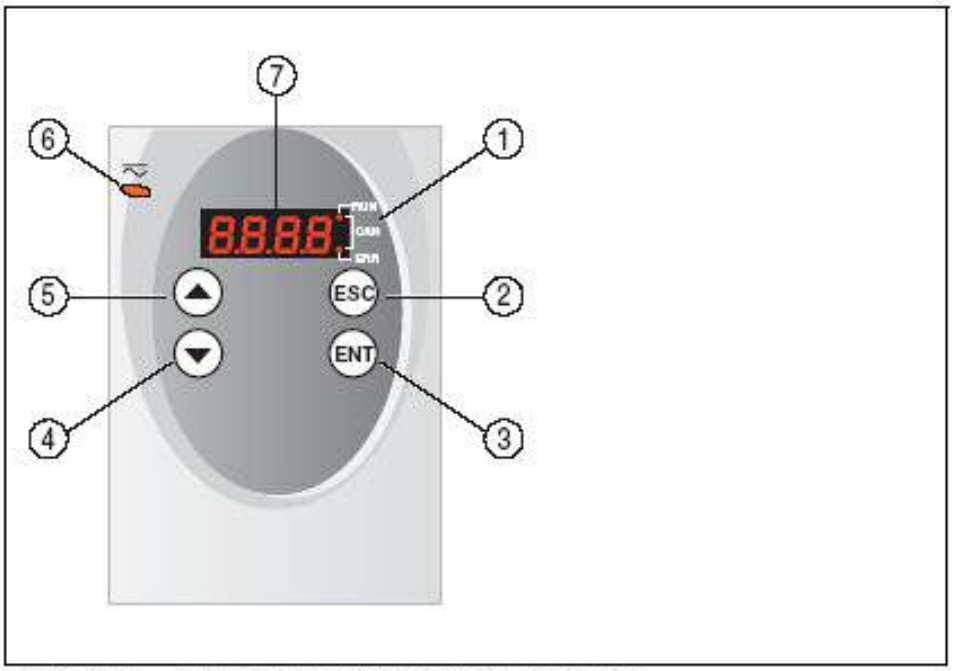
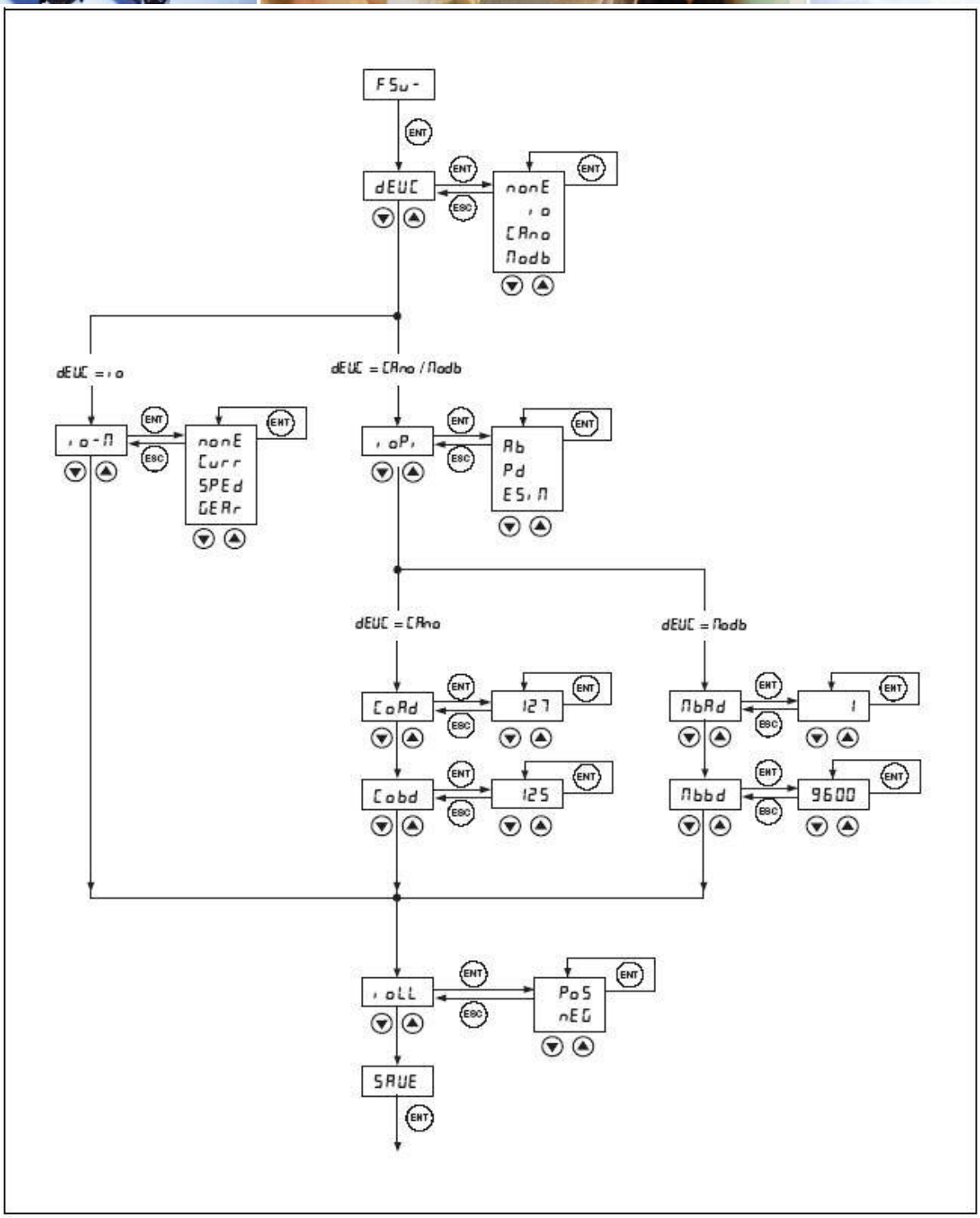


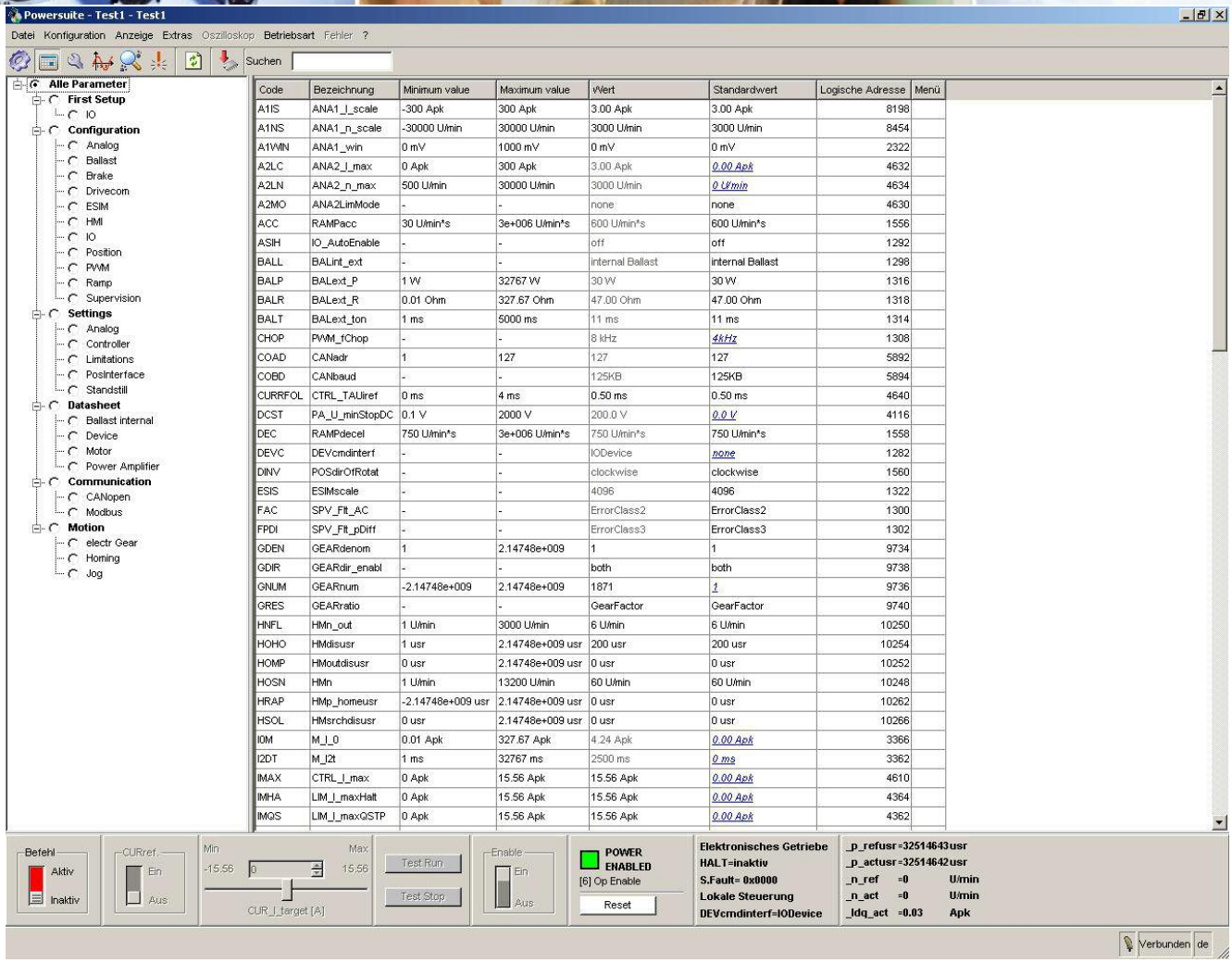
Figure 7.2 HMI and decentralised control terminal

(1) LEDs for CANopen
(2) ESC:
- exit a menu or parameter
- return from the displayed to the last saved value
(3) ENT:
- call a menu or parameter
- save the displayed value to EEPROM
(4) Down arrow:
- switch to next menu or parameter
- reduce the displayed value
(5) Up arrow:
- switch to previous menu or parameter
- increase the displayed value
(6) Red LED on: DC bus under voltage
(7) Status display
(8) Quick Stop (Software Stop)
(9) Fault Reset (Continue)
(10) No function

2 LEDs show the status of the CANopen state machine as per the CANopen standard DR 303-3.



The device control mode (dEUC) is IO-mode, and the IO-mode is electronic gear (GEAr). With the parameter IO Logical level it is possible to adjust a positive or negative logic. After the saving of the parameters the drive must be switched off and on again.
Only two parameters more must be adjusted, the configuration of the pulse/direction interface and the gear factor. That could be done via the HMI or with Powersuite, a comfortable programming tool.



Designed 07.06.2005 by

Patrick Friedmann

Berger Lahr GmbH

Customer Support Center - Technical Support

Gewerbestraße 9

D-77749 Hohberg-Niederschopfheim.

Tel.: +49 7808 / 943-235

Fax: +49 7808 / 946-58 235

E-Mail: mailto:patrick.friedmann@berger-lahr.com

a company of Schneider Electric

BERGER LAHR