KUHNKE

Kuhnke Elektronics
Instruction Manual
Profi Control KUAX 681M
Positioning Controller for PROFIBUS

E 311 GB 18 March 1996 / 66.622

This manual is primarily intended for the use of the designing engineer, the project planning engineer, and the developing engineer. It does not give any information about delivery possibilities. Data is only given to describe the product and must not be regarded as guaranteed properties in the legal sense. Any claims for damages against us – on whatever legal grounds – are excluded except in instances of deliberate intent or gross negligence on our part.
We reserve the rights for errors, omissions or modifications.

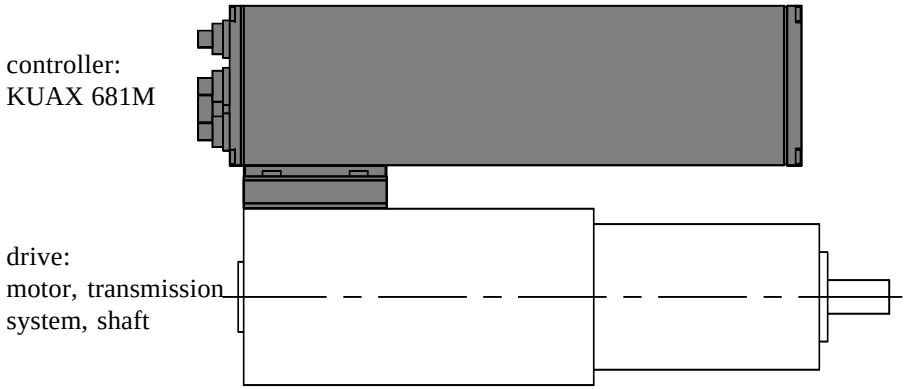
Table of contents

1. Introduction

The KUAX 681M is a decentralized positioning controller which you connect directly to the motor to be controlled (for motor types see appendix "A.2 Motors"). The connections to the motor are made via plug-type contacts.

The controller is designed for use on the PROFIBUS where it communicates either with higher-ranking masters such as KUAX 644, KUAX 680I, KUAX 657P or with other controllers.
All positioning information and control commands are transferred by the master via PROFIBUS to the KUAX 681M. The KUAX 681M uses the same medium to return its responses (positions, direction of motion etc.) to the master.

The construction of the KUAX 681M is such that the device is directly connected to the drive to be controlled:



1.1. What are typical applications of the KUAX 681M?

The control system KUAX 681M was designed for the operation of brushless DC drives. It is particularly well adjusted to the Dunker-made motor types BG 63 and BG 83.

Working principle

The motors are controlled via a four-quadrant regulator with block commutation. The rotor position is normally registered by Hall sensors. Distance measurement can be made either by the Hall sensors or by an incremental encoder.
Due to variable ramp parameters, the positioning profile can be adjusted to individual tasks.
Regulation of speed and position is taken care of by the software. These regulators can be set.

Modes of operation

In its basic function, the system is designed for the operation of servo drives. However, it can also be used for general positioning tasks which demand only little dynamics.
The system does not provide the functionality for maximum load operation (S1) or interrupted periodic operation with variable loads and speeds and no breaks (S7 to S9).

This information (modes of operation in accordance with VDE 0530 Part 1) only concerns the control electronics of the KUAX 681M. We do not provide any information about the actual motors as this information can be obtained from the manufacturer (Dunker).

1.2. Transition from individual controllers to network systems

Programmable logic controllers (PLCs) play a major role for industrial automation. This is due to three main reasons:

- their universal applicability;
- their easy and comprehendible programming;
- the extensive amount of tools for testing and start-up.

As problem-orientated micro-computers and in conjunction with their ever-increasing capacities, PLCs have adopted more and more elements of process computing systems. A strong tendency towards hierarchically structured process control systems has since become apparent which are characterized by task separation. Each sub-system is responsible for the jobs it is best suited for.

Task separation leads to decentralization

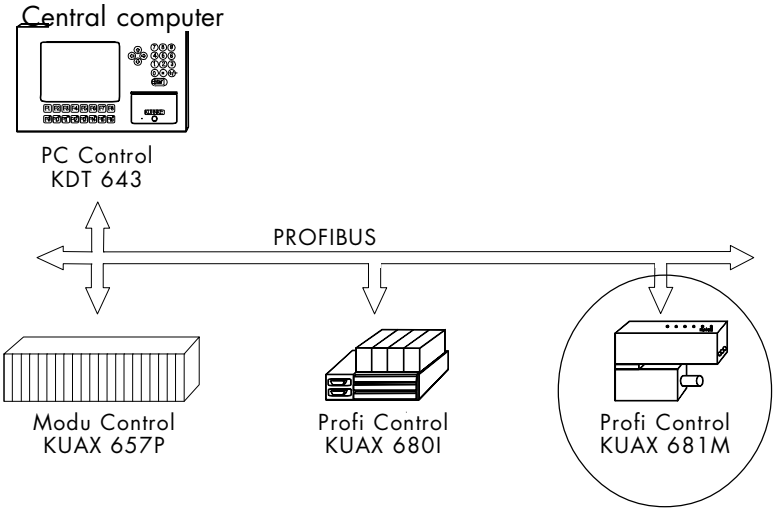
Including further components such as positioning controllers, valve islands, sensors and actuators produces networking systems at the field level.
High-capacity interfaces and transmission media are of the greatest importance because they carry the communication not only between individual PLCs but also between PLCs and other controllers or devices and PCs.

Advantages of decentralization:

- during the project planning phase
  machines of various manufacturers can be combined
- during installation
  less material (cables, connecting elements...) and less space (cable conduits, distributors, switching cabinet) is needed while mounting and assembly times are reduced
- during commissioning
  reduced machine setup times, modular construction facilitates pre-tests using indivdual stations
- during operation
  even if there is failure in one part of the system, the other parts can continue to operate and failure analysis is much facilitated by PROFIBUS

1.3. KUAX 681M on the PROFIBUS

The Profi Control KUAX 681M (order number 681.000.00 or 681.001.00 resp.) is equipped with a PROFIBUS interface. The device meets the classification of server profile class 2. While adhering to the master's positioning information and control commands, it is thus suitable for independently positioning a drive. It is also capable of transmitting current system information such as position, direction etc. to the master.



2. Safety and Reliability

2.1. Target group

This instruction manual contains all information necessary for the use of the described product (control device, software, etc.) according to instructions. It addresses the **personnel of the construction, project planning, service and commissioning departments**. For proper understanding and error-free application of technical descriptions, instructions for use and particularly of notes of danger and warning, **extensive knowledge of automation technology** is compulsory.

2.2. Reliability

Reliability of Kuhnke controllers is brought to the highest possible standards by extensive and cost-effective means in their design and manufacture.

These include:

- selecting high-quality components,
- quality arrangements with our sub-suppliers,
- measures for the prevention of static charge during the handling of MOS circuits,
- Worst Case dimensioning of all circuits,
- inspections during various stages of fabrication,
- computer aided tests of all assembly groups and their coefficiency in the circuit,
- stress-test in raised ambient temperatures during 72 hours realtime,
- statistic analysis of the quality of fabrication and of all returned goods for immediate taking of corrective action.

Despite these measures, the occurrence of errors in electronic control units - even if most highly improbable - must be taken into consideration.

2.3. Notes

Please pay particular attention to the additional notes which we have marked by symbols in this instruction manual:

2.3.1. Danger

This symbol warns you of dangers which may cause death, (grievous) bodily harm or material damage if the described precautions are not taken.

2.3.2. Dangers caused by high contact voltage

This symbol warns you of dangers of death or (grievous) bodily harm which may be caused by high contact voltage if the described precautions are not taken.

2.3.3 Important information / cross reference

This symbol draws your attention to important additional information concerning the use of the described product. It may also indicate a cross reference to information to be found elsewhere.

2.4. Safety

Our product normally becomes part of larger systems or installations. The following notes are intended to help integrating the product into its environment without dangers for man or machine/equipment.

2.4.1. To be observed during project planning and installation

- 24V DC power supply:
  · provide sufficient separation of low voltage,
  · apply power packs in accordance with IEC 364-4-41 or CENELEC HD 384.4.41 (VDE 0100, Part 410) respectively.
- In case of power breakdowns or power fades: the program has to be structured in such a way as to create a defined state at restart that excludes dangerous states.
- Emergency switch-off installations or other emergency installations have to be realized in accordance with EN 60204/ IEC 204 (VDE 0113). They must be effective at any time.
- Safety and precautions regulations for qualified applications have to be observed.
- Please pay particular attention to the notes of warning (→ 2.3. Notes) which, at relevant places, will make you aware of possible sources of dangerous mistakes or failures.
- The relevent standards and VDE regulations are to be observed in every case.
- Control elements have to be installed in such a way as to exclude unintended operation.
- Control cables have to be layed in such a way as to exclude interference (inductive or capacitive) which could influence the operation of the controller.

To achieve a high degree of conceptual safety in planning and installing an electronic controller it is essential to follow the instructions given in the manual exactly because wrong handling could lead to rendering measures against dangerous failures ineffective or to creating additional dangers.

2.4.2. To be observed during maintenance and servicing

- Precaution regulation VBG 4.0 must be observed, and §8 (Admissible deviations during working on parts) in particular, when measuring or checking a controller in a power-up condition.
- Repairs must only be executed by the trained Kuhnke personnel (usually in the main factory in Malente).
  Warranty expires in any other case.
- Spare parts:
  Only use parts approved of by Kuhnke. Use only genuine Kuhnke modules in modular controllers.
- Modules must only be connected to or disconnected from the controller with no voltage supplied. Otherwise they may be destroyed or (possibly not immediately recognizably!) detracted from their proper functioning.
- Always deposit batteries and accumulators as hazardous waste.

2.4.3. Measures for the prevention of electrostatic charge

Electrostatic charge is dangerous for components and assembly groups. It is a peculiarity of electrostatics to not destroy the sensitive components but to damage them in a not immediately conceivable way. It is because of this that devices stop functioning after some time of service.
The ESD measures (ESD = electrostatic discharge) executed in the factory are only guaranteed to be effective if they are also regarded by the user (service).

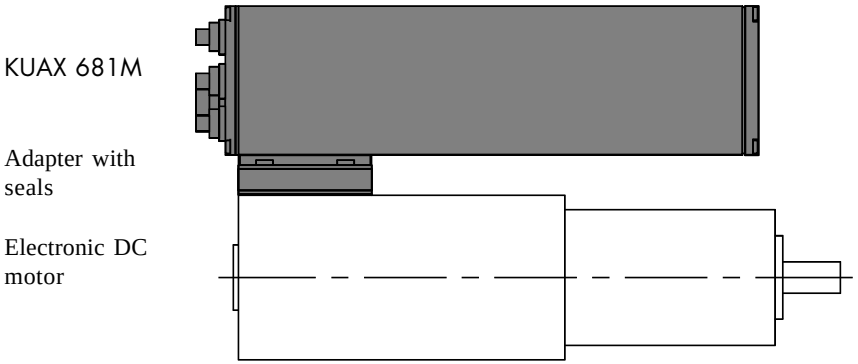
Please note:

- Only store parts in their factory-packing or in an antistatic packing of similar quality.
- Assembly groups must only be touched by persons who are grounded via a wrist bracelet and/or a discharging mat and shoe-grounding strips (⚠ observe protection of people!).
- Only ship assembly groups in their factory-packing or in an antistatic packing of similar quality.

Reference to the information brochure of 3M Deutschland GmbH, Neuss): "Wissenswertes über die Elektrostatik in der Mikroelektronik" (Interesting Facts about Electrostatics in Micro-Electronics)

3. Hardware

The KUAX 681M is a compact device which is mounted directly on the motor. All motor connections are made by a plug-type connector via the adapter.



Device types for use with the KUAX 681M

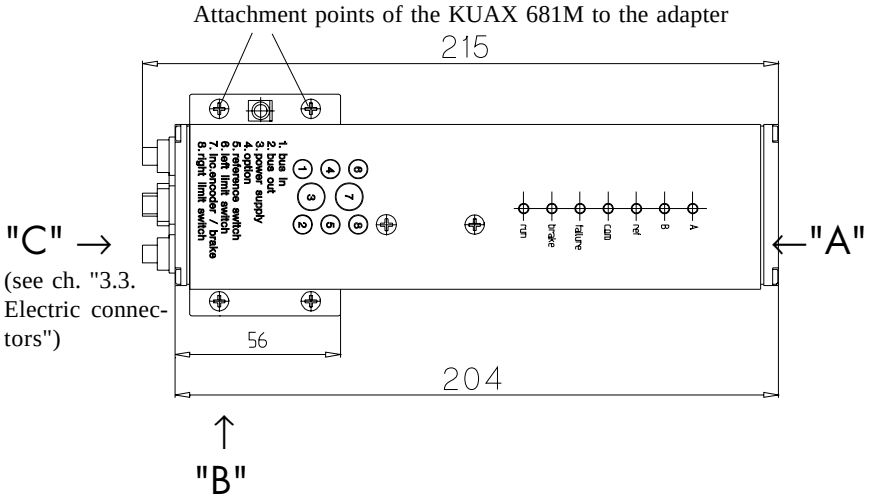
There are basically 2 different motors with various capacities that can be controlled by the KUAX 681M:

Positioning controller	Field bus	Order number	Motor voltage rat.	Adaptable drives*)
Profi Control KUAX 681M	PROFIBUS	681.000.00	24 V DC	BG 63 x 55
		681.001.00	40 V DC	BG 63 x 55 BG 83 x 90

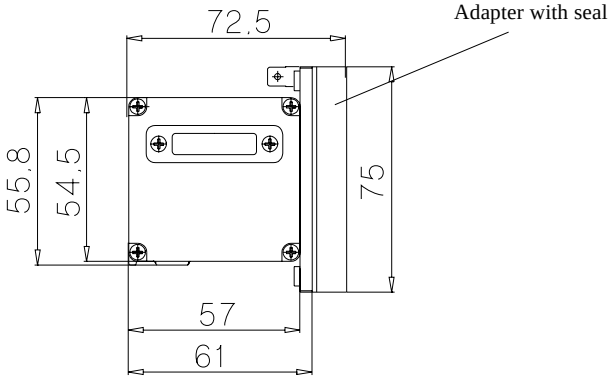
*) Electronic DC motors make Dunker (see appendix "B. Order information")

3.1. Dimensions and mounting

3.1.1. Top view



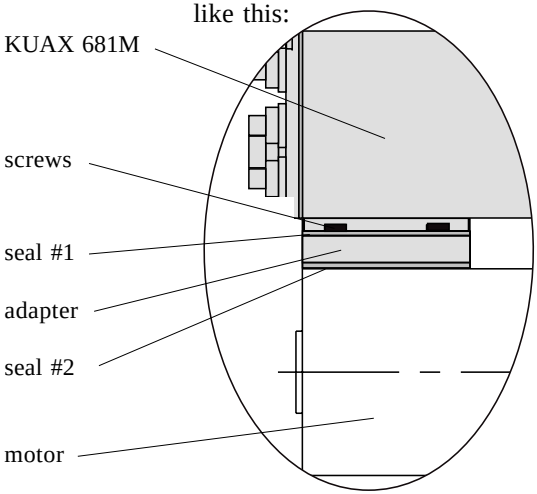
3.1.2. Side view "A"



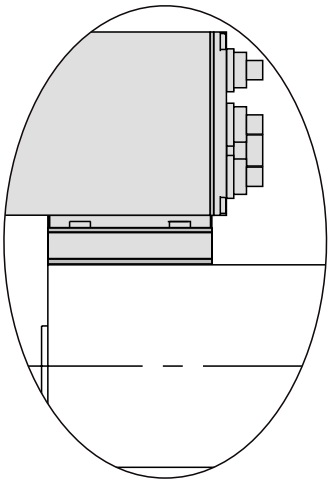
3.1.3. Mounting the KUAX 681M, adapter, motor

View "B" (see ch. "3.1.1. Top view")

Depending on the space available, the KUAX 681M can be mounted on the motor rotated by 180°:
like this:



or like this:



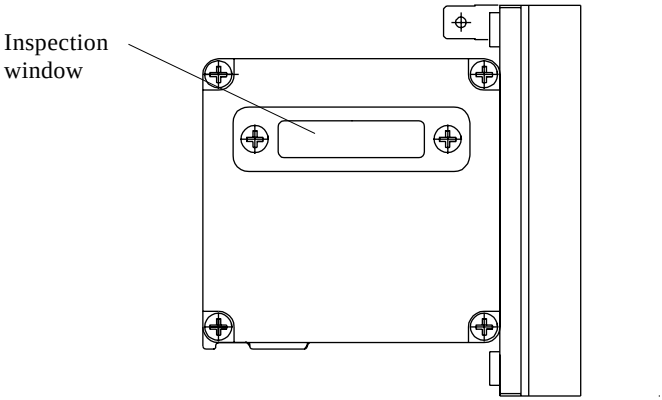
Proceed as follows

Screw adapter and seal #2 to the motor.
Connect the 8pin connector of the KUAX 681M to its counterpart on the motor.
Screw KUAX 681M and seal #1 to the adapter.

Please refer to appendix "B.2 Accessories" for a description of the mounting material consisting of adapter, seals and screws.

3.2. Coding switch

On the right side of the device (see ch. "3.1.1. Top view", view "A"), there is an 8pin coding switch located behind an inspection window. Unscrewing the inspection window gives you access to the switch for setting:



Use this switch to set the station address of the KUAX 681M on the field bus and the transfer rate.

3.2.1. Coding instructions

dip switch								field bus station address
1	2	3	4	5	6			
off	off	off	off	off	off			illegal setting
on	off	off	off	off	off			1
off	on	off	off	off	off			2
on	on	off	off	off	off			3
off	off	on	off	off	off			4
on	off	on	off	off	off			5
off	on	on	off	off	off			6
on	on	on	off	off	off			7
etc. through:								
off	on	on	on	on	on			62
on	on	on	on	on	on			63
1	2	4	8	16	32	←		significant. of switches 1-6 (binary coded)

PROFIBUS						7	8	baudrate [kBaud]	max. line length [m] *)
						off	off	500	200 (400)
						on	off	19.2	1200 (2400)
						off	on	9.6	1200 (2400)
						on	on	illegal setting	

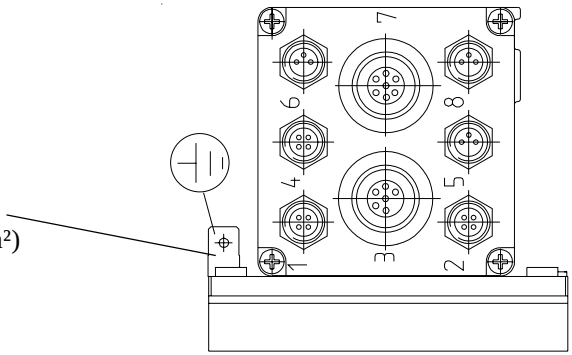
- The setting of the coding switch is only read when switching the supply voltage on. Changes during operation will have no effect.
- For reasons of space economy, not all of the station address settings have been included in the table. Switches 1...6 are binary coded. You can thus easily determine the missing station addresses going by significances.

*) The lines lengths given are valid for wire diameters of ≥ 0.22 mm². For diameters ≥ 0.5 mm², the values in brackets apply.

3.3. Electric connectors

The electric connectors are all located on one side of the KUAX 681M:

Side view "C" (see ch. "3.1.1. Top view")



Legend:

1 4pin male connector
- field bus in (interface RS 485)

2 4pin female connector
- field bus out (interface RS 485)

3 6pin male connector
- power supply for KUAX 681M and motor

4 4pin female connector
- 1 IRQ input
- 1 optional input (e.g. key switch)

5 3pin female connector
- reference switch

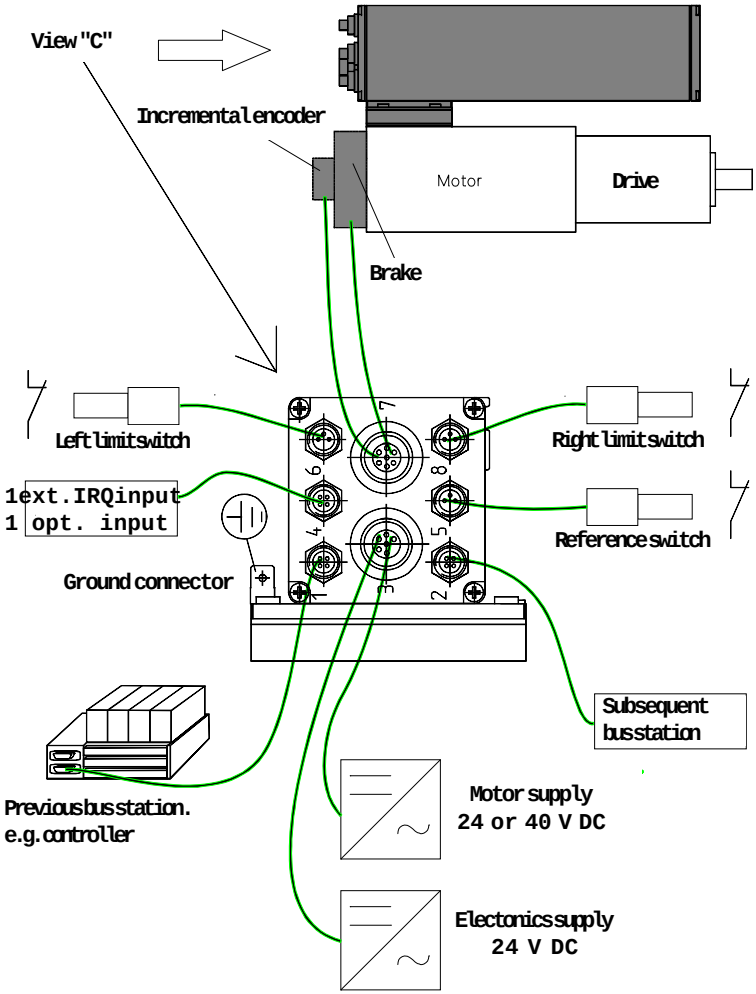
6 3pin female connector
- left limit switch

7 7pin female connector
-incremental encoder and brake

8 3pin female connector
- right limit switch

Refer to appendix "B.2. Accessories" for order information of the connecting material (male and female cable connectors etc.).

3.3.1. Wiring scheme



3.3.2. Bus connection

The KUAX 681M is operated via a field bus. Using a suitable controller (such as KUAX 657, KUAX 680I ...) which is also connected to the PROFIBUS you can transfer positioning information and control commands and also request actual values (current position and direction etc.).
The interface is an RS 485 with separated potential.

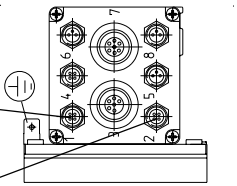
3.3.2.1. Plug-type connectors

To make the bus connection, there are two 4pin plug-type connectors located on the left side of the device (no.s 1 and 2, see illustration). Connector #1 is male, connector #2 is female.

The hardware of both connectors is switched in parallel so that they have the same function.

Like this it is easy to integrate the device into a PROFIBUS network:

- from the previous bus station (bus in)



- to the subsequent bus station (bus out)

A complicated connection via a T-plug and a spur line to the device is thus not required.

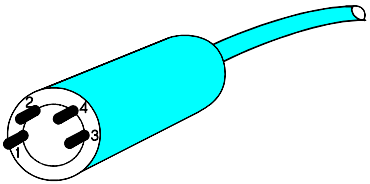
Even when you switch the device off, the bus connection to the previous and subsequent bus station remains active as long as you leave the connectors plugged in. It is possible to connect unplugged connectors to each other to maintain the bus connection between the other stations.

Pin wiring of the bus cable

Pin	Signal	Explanation
1	RxD/TxD-N	Receive/Transmit Data-N
2	RxD/TxD-P	Receive/Transmit Data-P
3	Shield	Shield or Protective Ground
4	Rt active	Activate Bus Termination *)

*) only if terminating station: bridge 2 and 4

Contact arrangement in the cable connectors



Cable plug (bus in, bus out)

3.3.2.2. Bus cable

Topology:

Line that is terminated at either end of the bus by a characteristic impedance resistor (see ch. "3.4.3. Bus termination"). Spur lines (if used) must be no longer than 0.3 m.

Medium:

see ch. "B.2.3. Bus cable"

Total line length:

The permissible line length is inversely proportional to the set baud rate (see table "3.2. Coding switch").

The data lines (pins 1 and 2) must be connected 1:1. Crossover connections are not permissible. The shielding must be connected to both sides.

3.3.2.3. Bustermination

Both ends of the entire field bus cable must be connected to a line termination. This ensures a defined idle potential on the line at times where there is no communication.

Built-in bus termination

The KUAX 681M is already equipped with bus termination resistors when it comes to you.
The pull-up and pull-down resistors are high-impedance resistors (33 kW) and can therefore stay permanently connected without problems.
The actual bus termination is internally connected to line RxD/TxD-N and, on the other side, connected to pin 4.

To activate the built-in bus termination

If the KUAX 681M is a terminating bus station, i.e. if it is connected to only one other station, then you activate the bus termination by implementing a jumper between pins 2 and 4:

Jumper wire:

Pin	Signal	Explanation
1	RxD/TxD-N	Receive/Transmit Data-N
2	RxD/TxD-P	Receive/Transmit Data-P
3	Shield	Shield or Protective Ground
4	Rt active	Activate Bus Termination *)

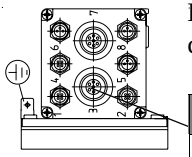
*) only if terminating station: bridge 2 and 4

Active bus termination

Extended tests and practical experience have shown that under extreme operating conditions, the built-in bus termination is not always sufficiently safe. We will renegotiate the concept. We therefore recommend using the active bus termination instead of the built-in bus termination for the time being.

The active bus termination is separately supplied with voltage (230 V AC). It thus remains active even when the station is switched off.
Order number 680.180.10

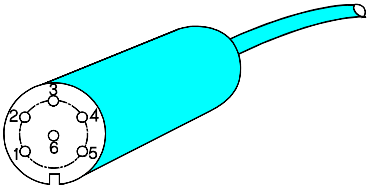
3.3.3. Power supply of KUAX 681M and motor



Both controller and motor are supplied with power via 6pin connector (#3). The voltages are fed in separately:

Pin	Supply voltage	
1	+ 24 or 40 V DC	motor supply depending on type of motor used: 24 or 40 V DC. Connect double due to load on the connector!
2		
3	0 V	
4		
5	0 V	supply of control electronics KUAX 681M: 24 V DC
6	+ 24 V DC	

Pin arrangement in the female connector



Recommended cables

Unshielded cable
LIYY 6 * 0.5 mm²
2 leads for electronics supply, 4 leads for motor supply

Shielded cable
LIYCY 4 * 0.75 mm²
2 leads for electronics supply, 2 leads for motor supply

If you are using LIYCY (4 x 0.75 mm²) you have to bridge connector pins (in the male connector) 1 and 2 as well as 3 and 4 to relieve the pins.

Power supply of the KUAX 681M control electronics

Nominal voltage:	24 Volt DC, ± 20%
Max. residual ripple:	< 5%
Nominal current:	approx. 150 mA, with brake output: approx. 650 mA
Recommended cable:	min. 2 * 0.5 mm ²

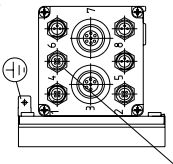
Motor power supply (intermediate circuit voltage)

	Type: 24 V / BG 63	*Type: 40 V / BG 83*
Nominal voltage:	24 - 28 Volt DC	40 - 50 Volt DC
Max. residual ripple:	< 5%	< 5%
Max. voltage:	30 V DC	70 V DC
Min. voltage:	20 V DC	30 V DC
Nominal current:	4.5 A (eff)	7.5 A (eff)
Max. impulse current:	2 * I_N, 500 ms	2 * I_N, 500 ms

Observe the following instructions to guarantee trouble-free motor operation:

A simple unstabilised DC power supply is required to provide the intermediate circuit voltage. Provide 1000 µF min. (buffer capacitor in the intermediate circuit) per 1 A working current. Never separate the intermediate circuit during motor operation. Install emergency shut-off devices on the primary side. Lay power supply lines separately or use shielded cables. Supply control electronics and motor from separate power packs. Due to motor switching operations and return currents from the motor, the supply voltage of the motor fluctuates very much which would not be tolerated by the electronics side.

3.3.4. Connector for external IRQ requester and optional input

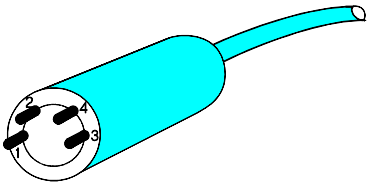


Connector #4 is a 4pin female connector for the connection of 2 inputs. One input is used as IRQ requester; the other one can be assigned to user requirements. However, this must be implemented as an option in the KUAX 681M software and is then considered a customer version.

Pin No.	Pin Colour *1)	Signal	Description
1	brown	+ 24 V DC	supply of the switch
2	white	input	switch signal (optional)
3	blue	0 V	supply of the switch *2)
4	black	input	signal of the external IRQ requester

*1) colour of cable in prefabricated connectors with cable
*2) required for use with an electronic proximity switch

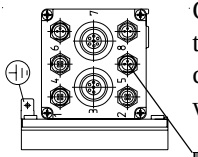
Pin arrangement in the male cable connector



Cable type

Use normal control cables designed for 24 V to connect the inputs.

3.3.5. Connector for the reference switch

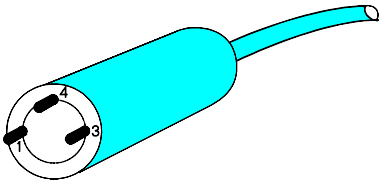


Connector #5 is a 3pin female connector for the connection of the reference switch. This switch must be a normally closed contact (n.c. switch or closed circuit contact). It is supplied with 24 V DC from the KUAX 681M:

Pin No.	Pin Colour *1)	Signal	Description
1	brown	+ 24 V DC	reference switch supply
3	blue	0 V	reference switch supply *2)
4	black	Ref	signal of reference switch

*1) colour of cable in prefabricated connectors with cable
*2) required for use with an electronic proximity switch

Pin arrangement in the male cable connector



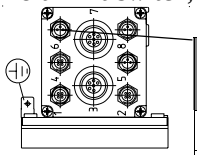
Cable type

Use normal control cables designed for 24 V to connect the inputs.

3.3.6. Connectors for the limit switches

Connectors #6 and #8 are 3pin female connectors for the connection of the limit switches. These switches must be normally closed contacts (n.c. switch or closed circuit contact). They are supplied with 24 V DC from the KUAX 681M:

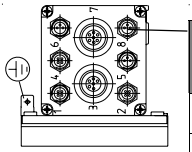
Left limit switch, female connector #6



Pin No.	Pin Colour *1)	Signal	Description
1	brown	+ 24 V DC	limit switch supply
3	blue	0 V	limit switch supply *2)
4	black	input	signal of limit switch

*1) colour of cable in prefabricated connectors with cable
*2) required for use with an electronic proximity switch

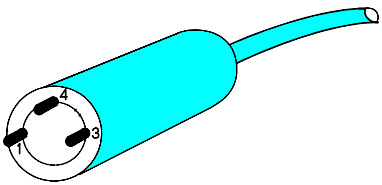
Right limit switch, female connector #8



Pin No.	Pin Colour *1)	Signal	Description
1	brown	+ 24 V DC	limit switch supply
3	blue	0 V	limit switch supply *2)
4	black	input	signal of limit switch

*1) colour of cable in prefabricated connectors with cable
*2) required for use with an electronic proximity switch

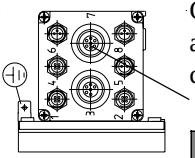
Pin arrangement in the male cable connector



Cable type

Use normal control cables designed for 24 V to connect the inputs.

3.3.7. Connector for incremental encoder and brake

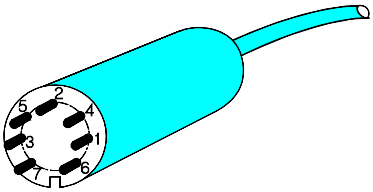


Connector #7 is 7pin female connector. If you are working with an incremental encoder and a brake you connect them to this connector.

Pin	Signal	Device	Description
1	A	incremental encoder	channel A
2	B		channel B
3	Ref imp. *)		encoder ref. impulse
4	+ 5 V DC		encoder supply
5	0 V		
6	0 V	brake	brake on
7	+ 24 V DC		

*) this signal is also called index signal (channel I) in Dunker-made motors

Pin arrangement in the male cable connector



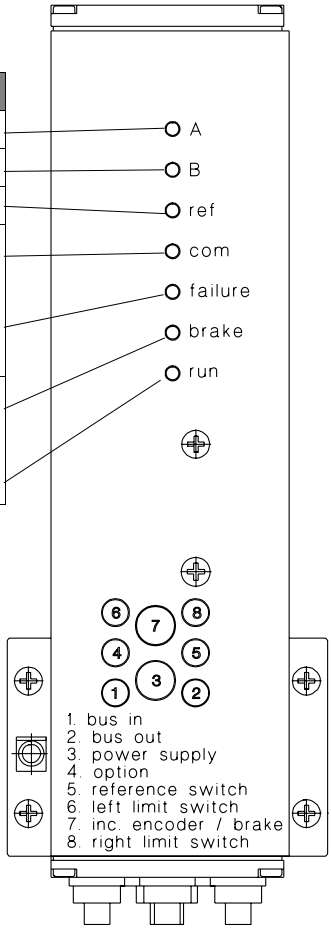
Cable type

The cable connector for incremental encoder and brake is delivered with the motor when it comes to you from the manufacturer. You have to order the cable separately, however.

3.4. Status indicators

On its top side, the KUAX 681M has a row of light emitting diodes:

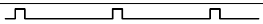
Colour	Function
yellow	incremental encoder: input A
yellow	incremental encoder: input B
green	reference switch: input ref
yellow	communication lights up while the KUAX 681M is transmitting
red	system failure: flashes in the event of a failure (see "3.4.1. System failure messages")
red	brake working current brake: lights up when addressed quiescent current br.: lights up when not addr.
green	program run



3.4.1. System failure messages

System errors and failures are internally coded from 1 through max. 255 and incidated via the "failure" LED. This LED indicates the number of the failure that occurred last by flashing in a certain rhythm.

Examples:

Number	Flash rhythm
1	 ....
2	 ....
3	 ....
4	 ....
etc.	

KUAX 681M failure codes

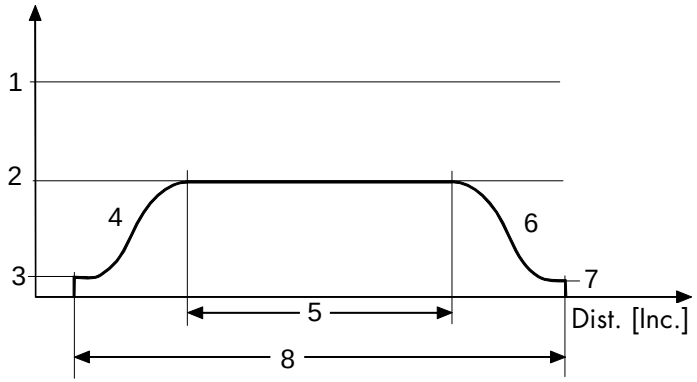
Number	Failure type
1	short circuit in motor control system
2	undervoltage/overvoltage of KUAX 681M supply
3	watchdog
4	field bus error or wrong station address set
5	
6	
7	
8	parameter checksum error
9	

The failures are also reported to the master controller via the field bus (refer to ch. "5.1. KUAX 681M on the PROFIBUS").

4. Description of functions

4.1. Positioning parameters

The figure below shows the parameters that are relevant for a positioning step and illustrates their importance:



Legend

Pos.	*Parameter*	*Unit*	*in Data Block*
1	max. nominal speed	[1/min]	parameters
2	nominal speed	[1/min]	commands
3	starting speed	[1/min]	parameters
4	starting ramp	[Inc.]	parameters
5	distance travelled at nominal speed		
6	stopping ramp	[Inc.]	parameters
7	deceleration speed	[1/min]	parameters
8	positioning step	[Inc.]	commands

The table indicates into which data block you input the parameters.

The operands given in the following text are examples. The programmer can define the range himself (refer to ch. "5.1. KUAX 681M on the PROFIBUS").

4.2. Ramp functions

The system allows you to adjust the ramps in a very general manner. You can set different parameters for starting and stopping ramps as well as for movements left and right.

4.2.1 Ramp forms

There are three different ramp forms available. The setting is made via the "parameters" data block (BM03.11).

Setting	Ramp form	Data
0	sinus A (alternative)	calculated after the start command
1	sinus²	read from table
2	linear	read from table

4.2.2 Ramp parameters

Input the following parameters into the "parameters" data block:

right starting ramp (BM01.04)
left starting ramp (BM01.00)
right starting speed (BM01.10)
left starting speed (BM01.08)
right stopping ramp (BM01.06)
left stopping ramp (BM01.02)
right deceleration speed (BM01.14)
left deceleration speed (BM01.10)

4.3. Positioning step

For a positioning step you define the destination position in increments. Positioning steps are made up of starting ramp, distance travelled at nominal speed and stopping ramp. It is calculated at the start command including all ramp functions. Positioning steps can be defined to be absolute, i.e. with the destination position always in relation to zero, or relative, i.e. relating to the previous destination position.

Set parameters and start commands in data block "commands":

right nominal speed (BM04.06)	[1/min]
left nominal speed (BM04.04)	[1/min]
destination position (BM04.00)	[Inc.]

positioning mode, command byte (BM04.15), bit 6
start command, command byte (BM04.15), bit 4

The nominal speed must be no smaller than the starting or deceleration speeds. Otherwise the system will output an error message to data block "status" (BM06.10).

4.3.1. Direction of the positioning run

The phase shift between inputs A and B of the incremental encoder decides whether the count level stored in the positioning module is incremented (increased, counting up, forward run) or decremented (decreased, counting down, backward run). The travelling direction is derived from this and thus defined as follows:

If the counter is	⇒ *the system travels*
- incremented	⇒ right
- decremented	⇒ left

4.4. Changing the motor speed

You can change the motor speed while the system is moving during both positioning steps and manual operation. There are two modes for changing the speed:

4.4.1. Immediate change of speed

In data block "commands"
input the desired nominal speed into byte "right nominal speed" or "left nominal speed";
set bit 2 of the system control byte to "1".

4.4.2. Changing the speed at a defined position

In data block "commands"
input the desired nominal speed into byte "right nominal speed" or "left nominal speed";
input the desired counter value into byte "change speed at position" (BM04.08);
set bit 3 of the system control byte to "1".

4.5. Manual positioning mode

Manual positioning means moving the axis without defining a destination position.
Manual positioning mode is activated by the command byte (BM04.15) in data block "commands". Set bit 2 (left) or bit 3 (right) to "1". The manual positioning mode remains active while the corresponding bit is set to "1". When the bit is cleared, the positioning run is stopped using the set stopping ramp.

In "absolute" positioning mode the run is limited by the software limit switches.

Data block "commands":
- right nominal speed (BM04.06) [1/min]
- left nominal speed (BM04.04) [1/min]
- positioning mode, command byte (BM04.15), bit 6
- start command, command byte (BM04.15), bit 4

The nominal speed must be no smaller than the starting or deceleration speeds. Otherwise the system will output an error message to data block "status" (BM06.10).

4.6. Stop functions

There are two possible ways of stopping the motor before it reaches the destination position.

4.6.1. Emergency stop

Emergency stops are triggered by the system control byte (BM04.14) of data block "commands". Set bit 0 to "1" to stop the motor immediately without ramp. When the motor has stopped, the motor control is switched off. The motor release is cancelled. The motor is current-free.
To restart operation, the motor must be released again. To do so set bit 7 of byte "system settings" (BM03.13) in data block "parameters".

Failure and error messages in data block "status“:
positioning error (BM06.11) , bit 7
motor failure (BM06.12), bit 0

4.6.2. Stop with ramp

The stop command is activated by setting bit 1 of the system control byte (BM04.14) in data block "commands" to "1". The motor stops using the stopping ramp set. There are no error messages.

4.7. Reference run

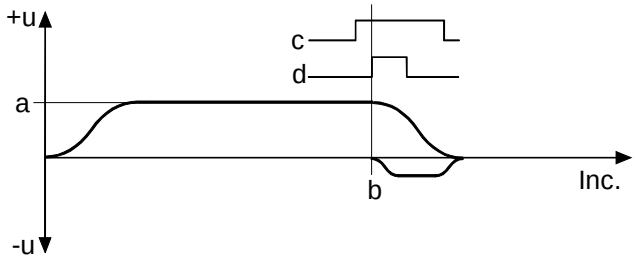
The system provides two modes for reference runs. The mode is selected by the setting of bit 0 of byte "counter setting" (BM03.14) in data block "parameters".

4.7.1. Setting the zero point by a reference switch only (e.g. initiator)

Only the reference switch (connector #5, 24 V n.c. contact) is analysed. The negative edge of the pulse zeroises the actual value counter.

4.7.2. Setting the zero point by the reference switch and the ref. impulse of the incremental encoder

The ref. switch (connector #5, 24 V n.c. contact) and the reference impulse of the incremental encoder are analysed. The positive edge of the reference impulse and the active reference switch together zeroise the actual value counter.



Process description:
run right with reference travelling speed (a)
stopping ramp activ. at pos. (c), ref. switch (e.g. initiator)
change of direction after stop
positioning run continued in left direction
run stopped immediately at positive edge of encoder reference impulse (d = machine zero point)
position regulator (b) maintains position

4.7.3. Relative reference point

After zeroisation, the drive automatically runs to a defined absolute counter value. This value is stored in data block "parameters" (BM00.04).

4.7.4. Starting a reference run

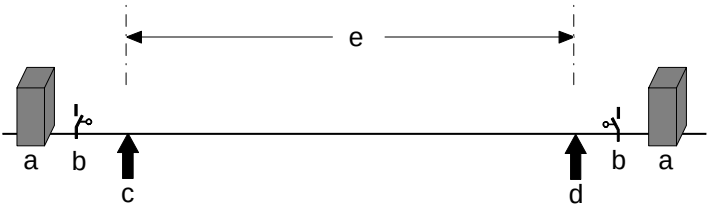
Travelling speed of the reference run and start command including definition of direction are set in
data block "commands":

right nominal speed (BM04.00) [1/min]
left nominal speed (BM04.00) [1/min]
start by the command byte (BM04.15)

The nominal reference speed must be no smaller than the starting or deceleration speeds. Otherwise the system will output an error message to data block "status".

4.8. Limit switches

In most cases of application, the working range of the axes will be limited by mechanical stops. These are protected against being passed over by limit switches (hardware). The hardware is switched off in case of a failure occurring.
You can also limit the working range by setting software limit switches without switching the hardware off. Error messages (in data block "status") ensure appropriate reactions by the positioning program.



a mechanical limitation stops
b mechanical limit switches
c left software limit switch
d right software limit switch
e axis working range

4.8.1. Hardware limit switches (conn. #6 and #8, n.c. contacts)

If a limit switch at either the left or the right end of the working range is activated, the motor stops immediately without ramp like in case of an emergency stop. When the motor has stopped, the motor control is switched off. The motor release is cancelled. The motor is current-free.
To restart operation, the motor must be released again. To do so set bit 7 of byte "system settings" (BM03.13) in data block "parameters".

Failure and error messages in data block "status“:
positioning error (BM06.11) , bit 7
motor failure (BM06.12), bit 0

4.8.2. Software limit switches

Function with absolute positioning steps
If the requested positioning step is greater than the set software limit switch, then the positioning step will not be carried out. The system will output an error message.

"Actual value counter" parameter in data block "parameters“:
- right software limit switch, positive (BM00.12) [Inc.]
- left software limit switch, negative (BM00.08) [Inc.]

Error messages in data block "status“:
- parameter error (BM06.10)

Function with relative positioning steps
The software limit switches are inactive.

Function during manual positioning mode
The motor stops with ramp at the counting value of the software limit switch. The stopping ramp will be activated before the software limit switch is reached.
Going into reverse frees the system from the software limit switch again.

Error message in data block "status“:
- positioning error (BM06.11)

4.9. Actual value counter

The system counts the increments of the path travelled in both right and left direction (depending on sign). Depending on the setting used, the system counts either the increments of the connected incremental encoder (default: 500 increments per revolution [Inc./rev] or the signals of the Hall sensor of the motor. There is always a fourfold analysis of the signals. This results for example in the following different resolutions:

Encoder resolution	*Fourfold analysis*
incremental encoder 500 Inc./rev	2000 Inc./rev
incremental encoder 1000 Inc./rev	4000 Inc./rev
Hall sensors of motor 2 Inc./rev	8 Inc./rev

The counter is set and released via data block "parameters" (BM03.14). The setting is accepted by setting bit 1 of byte "special settings" (BM03.15) to "1".
For safety reasons, the new settings will only become active when data block "commands" is being transmitted while bit 7 of the command byte (BM04.15) is set to "1" at the same time.

4.9.1. Counter preset value

It is possible to overwrite the contents of the actual value counter. To do so, simply input the desired value into byte "counter preset value" (BM00.00) of data block "parameters".
The new value is accepted by setting bit 7 of byte "special settings" (BM03.15) to "1".

For safety reasons, the new settings will only become active when data block "commands" is being transmitted while bit 7 of the command byte (BM04.15) is set to "1" at the same time.

4.10. Speed regulator

The speed regulator is constructed as a PID controller. You can set the following parameters:

P part	proportional factor, typical factor:	3 - 4 (e.g. 35/10)
Ta	sampling frequency ,	typical: 2 ms
Tn	adjustment time,	typical: 10 ms
Tv	suspension time,	typical: 1 ms

P part
To be able to also use decimal point numbers as proportional factors, dividend and divisor must be set in a separate byte each. The divisor must not be zero.

I part and D part
The time input in Ta does not constitute the real scanning frequency but, in conjunction with Tn and Tv, the I part and the D part of the PID controller. These parts are set by the relation Ta/Tn and Tv/Ta respectively.
The actual scanning frequency is set by the PID controller. It changes permanently during operation because the smallest possible value is chosen at any given time. This value in turn depends on the speed.

4.10.1. Speed regulation range (Dunker-made motors)

The speed regulation range varies for the different motor types and depends on the encoder used:

Type	no incremental encoder	incremental encoder 500 Inc/rev
BG 63	500 - 3300 1/min	50 - 3300 1/min
BG 83	500 - 2750 1/min	50 - 2750 1/min

4.11. Position regulator

The task of the position regulator is to keep the system at the destination position. The device recognizes deviations from that position and immediately readjusts it. You can set the sensitivity of the regulator by influencing the proportional part (KP) in data block "parameters" (BM02.11). Use adjustment time TN (BM02.12) to improve regulator performance. The adjustment time contains the integral part of the regulator. Typical values are listed in the table below (see ch. "4.14. Motor parameters").

4.11.1. Position regulator ON/OFF

It is possible ot switch the position regulator off. To do so, input bit 6 of byte "system settings" (BM03.13) in data block "parameters".
The new setting is accepted by setting bit 7 of byte "special settings" (BM03.15) to "1".

For safety reasons, the new settings will only become active when data block "commands" is being transmitted while bit 7 of the command byte (BM04.15) is set to "1" at the same time.

4.11.1.1. Position control range

Depending on the requirements of the current application, it is possible to set a position control range. Position control only becomes active if the current axis value is outside the control range set.
The set value (e.g. 6 Inc.) is valid for both the positive and the negative counting direction. Write the value into BM02.14 with a max. size of 2 byte.

4.12. Motor enable

You must expressedly enable the motor each time you initialise the motor.
The same procedure becomes necessary when the motor has been disabled after error or failure message.

To enable the motor:
set bit 7 of byte "system settings" (BM03.13) in data block "parameters" to "1";
set bit 0 of byte "special settings" (BM03.15) in data block "parameters" to "1";
set bit 7 of the "command byte" (BM04.15) in data block "commands" to "1".

To re-enable the motor after a failure, transmit data block "parameters" to the KUAX 681M first and then data block "commands" (see also "motor failure byte" (BM06.12) in data block "status").

4.13. Motor with brake

The system is equipped with a separate connector for the operation of motors with integrated brake. This connector is in plug #7, pins 5 and 6 (see ch. "3.3.7. Connector for incremental encoder and brake"). It provides the following operational characteristics:

output voltage: 24 Volt DC
output current: max. 500 mA

The output can be set to function as quiescent current brake or as working current brake. The output status is indicated by the red "brake" LED. The LED always lights up when there is current on the output.

The settings required for operation are made in byte BM03.13 in data block "parameters" (see ch. "5.2.1 Data block "parameters" (index 43)"):

Enable/disable mechanical brake function
settings: bit 2 = 1, disable: bit 2 = 0

Quiescent/working current brake
settings: bit 3 = 1, quiescent current brake: bit 3 = 0

If the function has been set to "mechanical brake", then the position regulator is switched off automatically when arriving at the destination position.

4.14. Motor parameters

The table below lists the typical motor parameters of Dunker-made motors types BG 83 and BG 63. There are two different operating modes: operation with incremental encoder (500 Inc./rev) or operation with Hall sensors.
The table is meant to be an aid and as such does not claim to offer the correct information for every practical application. It may become necessary to adjust the ramp functions and the parameters for speed control and position regulation to the given situation.

Parameter	*Type*	*Byte*	*Operand*	*500 Inc./rev*	*Hall Sensors*	*Described where?*
act. value counter:						
preset value	4	BM00.00	0	0		ch. 4.9.1.
point of reference	4	BM00.04	0	0		ch. 4.7.
left software limit switch	4	BM00.08	$80 00 00 00			*) ch. 4.8.2.
right software limit switch	4	BM00.12	$7F FF FF FF			*) ch. 4.8.2
left starting ramp	2	BM01.00	16000	64		ch. 4.2.
left stopping ramp	2	BM01.02	16000	84		ch. 4.2.
right starting ramp	2	BM01.04	16000	64		ch. 4.2.
right stopping ramp	2	BM01.06	16000	84		ch. 4.2.
left starting speed	2	BM01.08	50	80		ch. 4.1.
left deceleration speed	2	BM01.10	50	80		ch. 4.1.
right starting speed	2	BM01.12	50	80		ch. 4.1.
right deceleration speed	2	BM01.14	50	80		ch. 4.1.
encoder resolution	2	BM02.00	500	2		
speed regulator P part, dividend	1	BM02.02	35	15		ch. 4.10.
speed regulator P part, divisor	1	BM02.03	10	10		ch. 4.10.

*) $ indicates hexadecimal input

Parameter	*Type*	*Byte*	*Operand*	*500 Inc./rev.*	*Hall Sensors*	*Described where?*
speed regulator scan. frequ. TA	2	BM02.04	2	2		ch. 4.10.
speed regulator adjustm. time TN	2	BM02.06	6	6		ch. 4.10.
speed regulator suspen. time TV	2	BM02.08	1	1		ch. 4.10.
unused	1	BM02.10				
position regulator P part, KP	1	BM02.11	25	10		ch. 4.11.
unused	1	BM02.12				
position regulator adjustm. time TN	1	BM02.13	10	55		ch. 4.11.
pos. regul. range	2	BM02.14	0	0		ch. 4.11.
nominal motor current (I max)	2	BM03.00	7000	4500/7000		see mot. char.s
unused	2	BM03.02				
nominal motor speed (V max)	2	BM03.04	2750	3300/2750		see mot. char.s
bus timeout	2	BM03.06	1	150		see descr. of bus
unidirectional gear positioning	2	BM03.08	0	0		
unused	1	BM03.10				
ramp form	1	BM03.11	1	1		ch. 4.2.1.
motor type (make Dunker)	1	BM03.12	1	1		
system settings	1	BM03.13	%10000001	%10000001		*)
counter settings	1	BM03.14	%10000001	%10000011		*)
special settings	1	BM03.15	%00001111	%00001111		*)

*) % indicates binary input, see also data block "parameters"

4.15. Error and failure messages

The system differentiates between three types of failures and errors. They are indicated in the status block:

Failure	*Byte (Example)*
parameter error	BM06.10
positioning error	BM06.11
motor failure	BM06.12

Parameter errors

are recognized if wrong or illogical parameters (e.g. no preset speed value) do not allow the motor to start (see ch. 5.2.4.3.).

Positioning errors

may occur in positioning operation (e.g. limit switch reached). In this case, the positioning step will be stopped or not started in the first place (see ch. 5.2.4.4.).

Motor failures

are recognized if the motor is switched off due to wrong wiring or overload (see ch. 5.2.4.5.).

In case of motor failures will automatically cancel the motor enable status. When you have removed the problem and before you restart the motor you have to re-enable the motor (see ch. 4.12.).

Disabling failure and error messages

It is possible to acknowleges error and failure messages via the user program in the controller (by setting bit 5 of byte BM04.14 in data block "commands" to "1").
However, this does not remove the error or failure message from the memory of the KUAX 681M. Here the message will only be reset when the problem has been removed and the motor restarted without problems.
New failures are indicated as described before.

5. PROFIBUS

The PROFIBUS is a field bus. Its name is derived from an acronym of the term "PROcess FIeld BUS".
It was developed to network controllers (such as KUAX 680I, KUAX 657P etc.) and to provide a connection to the field level, i.e. to sensors and actuators (e.g. using input/output devices such as KUAX 680S). Furthermore, it is the interface system to the control level where one or sometimes several central computers are implemented to control the overall process.

Open communication

The principle of open communication is designed to ensure that devices of different manufacturers can network in one bus system (multi-vendor system). This field bus has been standardised in DIN 19 245.

Bus topology

The PROFIBUS is constructed as a line. However, it is also possible to set up a tree structure. The number of stations on any one line is limited to 32. Where that is not enough, you can set up another line which is connected to the first line by a bidirectional line amplifier (repeater).
Line amplifiers also count as bus stations so that the number of "real" stations on a line is reduced to 31 (or 30 if 2 repeaters are used).
The maximum number of bus stations can be increased to 122 by using up to 3 line repeaters.

Station address

Every station on the PROFIBUS is assigned its own station address by which it can be addressed by all other stations.
PROFIBUS supports station addresses between 0 and 126.

For notes on further reading please refer to appendix "C.1. Further reading".

5.1. KUAX 681M am PROFIBUS

The Profi Control KUAX 681M is a decentralized positioning controller. It is equipped with a PROFIBUS interface for communication with other controllers or with computers. The interface runs the PROFIBUS FMS protocol.

Device profile

So-called profile classes have been defined to classify PROFIBUS devices. These profile classes consist of definitions of the behaviour of the device during communication via PROFIBUS and in relation to the application.
As a PROFIBUS station, the KUAX 681M is a typical slave. Due to the services that the machine can provide it was classified as a device of profile class Sensor/Actuator Profile Class 2. The device uses the PROFIBUS FMS protocol for communication with the master.

Station address

The station address of the KUAX 681M is set by means of a coding switch (see ch. "3.2. Coding switch"). With this switch you can set addresses in the range of 1...63.

Master

The KUAX 681M receives its positioning commands from a controller. This could be a micro-controller such as KUAX 680I, a modular controller such as KUAX 657P, a PC-PLC KUAX 644 or every other control device that has been defined as a PROFIBUS controller.

Communication via block transfer

Communication between KUAX 681M and master is realized by a so-called block transfer (see ch. "5.3. KUBES modules"). This type of communication does not rely on cyclic processes but on communication by command.

5.2. Data blocks

Four data blocks are reserved for communication with the master. They are divided according to their practical functions:

Function	Length [Byte]	Block (Index) KUAX 681M	Direction	Block (Index) Master
parameters	64	43	←	optional (e.g. 163)
commands	16	41	←	optional (e.g. 161)
unassigned	16	42	←	
status	32	44	→	optional (e.g. 164)

The direction of data transfer is set:

Parameters
are defined by the master and sent to the positioning controller

Commands
are transferred from the master to the positioning controller

Status
status messages of the positioning controller are transferred to the master

Please refer to the following tables to learn the specific assignments of the individual data blocks.

For Kuhnke controllers, data blocks are defined under VEBES by their starting address and length. While the user can choose optional starting addresses, there are set lengths for the individual blocks (see table above). The addresses listed in the tables below must only be seen as examples.

5.2.1. Data block "parameters" (index 43)

The data block has a length of 64 byte:

Function	Unit	Byte	Address (Example	Notes
Counter setting and software limit switches:				
counter preset value	Inc.	4	BM00.00	accept by "take over setting" (see ch. 5.2.1.5....)
relative point of ref.	Inc.	4	BM00.04	values are immediately accepted after modification
left SW limit switch	Inc.	4	BM00.08	
right SW limit switch	Inc.	4	BM00.12	
Ramp parameters:				
left starting ramp	Inc.	2	BM01.00	values are accepted immediately after modification
left stopping ramp	Inc.	2	BM01.02	
right starting ramp	Inc.	2	BM01.04	
right stopping ramp	Inc.	2	BM01.06	
left starting speed	1/min	2	BM01.08	
left deceleration speed	1/min	2	BM01.10	
right starting speed	1/min	2	BM01.12	
right deceleration speed	1/min	2	BM01.14	
Speed regulation and position control:				
encoder resolution	Inc.	2	BM02.00	for operation without encoder: input 2 Inc. (Hall sensor)
amplification factor	Factor	1	BM02.02	speed regulator (see ch. "5.2.1.1....")
amplification factor	Div.	1	BM02.03	
Ta scanning frequency	ms	2	BM02.04	
Tn adjustment time	ms	2	BM02.06	
Tv suspension time	ms	2	BM02.08	
unused		1	BM02.10	
Kp position controller	Factor	1	BM02.11	P part
unused		1	BM02.12	
Tn position controller	ms	1	BM02.13	I part
position control range	Inc.	2	BM02.14	value valid for both directions

continued on next page...

Data block "parameters", continued from previous page:

Function	Unit	Byte	Address (Example	Notes
Current settings and special functions:				
nominal current (In)	mA	2	BM03.00	see motor rating plate
unused		2	BM03.02	
nominal speed Vmax	mA	2	BM03.04	max. speed (see motor rating plate)
time out PROFIBUS	ms	2	BM03.06	bus monitoring typ. 1 ms
unidirectional gear posit.	Inc.	2	BM03.08	
unused		1	BM03.10	
ramp form		1	BM03.11	0=sin A, 1=sin², 2=linear
motor type		1	BM03.12	0=BG 63, 1=BG 83, 2=BG 43
system settings		1	BM03.13	see ch. "5.2.1.3...."
counter settings		1	BM03.14	see ch. "5.2.1.4...."
special settings		1	BM03.15	see ch. "5.2.1.5...."

5.2.1.1. Speed regulator (for example: BM02.02...08)

The time input in Ta does not constitute the real scanning frequency but, in conjunction with Tn and Tv, the I part and the D part of the regulator. These parts are set by the relation Ta/Tn and Tv/Ta respectively.
The actual scanning frequency is set by the regulator. It changes permanently during operation because the smallest possible value is chosen at any given time. This value in turn depends on the speed.
The values set for Tn and Tv are always converted in the relation of Ta/<real scanning frequency>.

5.2.1.2. Motor type

The parameters for the motor set in this address are set by function "accept default parameters" (see ch. "5.2.1.5. Special settings (for example: BM03.15), bit 4"). They can be overwritten again afterwards.

5.2.1.3. System settings (for example BM03.13)

Bit	Status	Description
0	1	without limit switch
	0	with limit switch
1	1	with creeping
	0	without creeping
2	1	with mechanical brake, position regulator off
	0	without mechanical brake, position regulator on
3	1	working current brake enable
	0	quiescent current brake enable
4		unused
5	1	left unidirectional gear positioning
	0	right unidirectional gear positioning
6	1	position regulator off
7	1	motor enable (set to "0" in case of failure)

5.2.1.4. Counter settings (for example BM03.14)

Bit	Status	Description
0	1	reference impulse of switch
	0	reference impulse of encoder
1	1	counting pulse of Hall sensors
	0	counting pulse of encoder
2		unused
3		unused
4		unused
5		unused
6		unused
7	1	counter enable

5.2.1.5. Special settings (for example BM03.15)

Bit	Status	Description
0	1	take over system settings from byte (for example BM03.13)
1	1	take over counter settings from byte (for example BM03.14)
2	1	take over motor type from byte (e.g. BM03.12)
3	1	take over ramp form from byte (e.g. BM03.11)
4	1	set default parameters for the motor type set in (e.g. BM03.12)
5		unused
6		unused
7	1	take over counter preset value from (e.g. BM00.00)

Use this byte to activate the values and settings of the assigned addresses by setting the corresponding bit to "1" (if a bit is set to "0", the assigned settings will be ignored).

For safety reasons, the new settings will only become active when data block "commands 1" is being transmitted while bit 7 of the command byte (BM04.15) is set (=1) at the same time (see ch. "5.2.2.1....")..

After processing, the bits of byte "accept special settings" are automatically set to "0".

5.2.2. Data block "commands" (index 41)

Data block "commands" has a length of 16 byte:

Function	Unit	Byte	Address (Example	Notes
Start commands and destination position:				
destination position	Inc.	4	BM04.00	absolute or relative current destination position
left nominal speed	1/min	2	BM04.04	up to set max. speed only (for example BM03.04)
right nominal speed	1/min	2	BM04.06	up to set max. speed only
change of speed at position	Inc.	4	BM04.08	change speed when reaching the position set in this byte in conjunction with bit 3 of the system control byte (see ch. "5.2.2.2....")
unused		2	BM04.12	
system control byte		1	BM04.14	see ch. "5.2.2.2...."
command byte		1	BM04.15	see ch. "5.2.2.1...."

5.2.2.1. Command byte (for example BM04.15)

Bit	Status	Description
0	1	left reference run
1	1	right reference run
2	1	left manual positioning run
3	1	right manual positioning run
4	1	positioning step
5	1	handwheel operation
6	1 0	relative run mode absolute run mode
7	1	take over special settings (see ch. "5.2.1.5....")

5.2.2.2. System control byte (for example BM04.14)

Bit	Status	Description
0	1	emergency stop
1	1	stop
2	1	immediate speed change
3	1	speed change to preset value (value is in e.g. BM04.08 and following bytes)
4		unused
5	1	disable failure and error messages (see ch. "4.15....")
6	1	acknowledge interrupt (not active at present
7		unused

5.2.3. Free data block (index 42)

No function assigned at present.

5.2.4. Data block "status" (index 44)

Status and error messages are transferred to data block "status". This data block has a length of 32 byte:

Function	Unit	Byte	Address (Example	Notes
Motor control signals, status and error messages:				
current counter value	Inc.	4	BM06.00	updated permanently
current speed	1/min	2	BM06.04	updated permanently
current motor current	mA	2	BM06.06	updated permanently
task		1	BM06.08	see ch. "5.2.4.1...."
status		1	BM06.09	see ch. "5.2.4.2...."
parameter error		1	BM06.10	see ch. "5.2.4.3....."
positioning error		1	BM06.11	see ch. "5.2.4.4...."
motor failure		1	BM06.12	see ch. "5.2.4.5...."
command echo		1	BM06.13	commands of the master are read back during execution and indicated here
motor control echo		1	BM06.14	
special function echo		1	BM06.15	
Position, speed, hardware and software identifications:				
preset position (read)	Inc.	4	BM07.00	preset position of the current run task; also after manual positioning run
preset speed	1/min	2	BM07.04	calculated current preset speed for the current run task
actual position IRQ	Inc.	4	BM07.06	actual position after requesting an interrupt via IRQ input (see ch. "3.3.4...."); the value is stored until enable activated by bit 7 of the system control byte (see ch. "5.2.2.2....")
unused		1	BM07.10	for testing
SYS_ERR		1	BM07.11	system error message (see ch. "3.4.1....")
hardware identification		1	BM07.12	always "13" at present
software identification		3	BM07.13	date of software release

5.2.4.1. Task byte (for example BM06.08)

Bit	Status	Description
0	1	reference run
1	1	manual positioning run
2	1	positioning run
3	1 0	left travelling direction right travelling direction
4	1	initialization of run command
5	1	starting ramp active
6	1	stopping ramp active
7	1	short run or handwheel operation

We recommend permanently analysing the contents of data block "status" in the user program. This helps making operation as safe as possible.
The current motor status messages must be analysed for controlling the run program.

5.2.4.2. Status byte (for example BM06.09)

Bit	Status	Description
0	1	READY, system is ready to operate
1	1	BUSY, motor is moving
2	1	destination position reached
3	1	position regulator is on
4	1	mechanical brake is on
5	1	new interrupt value
6	1	counter enabled
7	1	watchdog

5.2.4.3. Parameter error byte (for example BM06.10)

Bit	Status	Description
0	1	no preset speed set
1	1	preset speed < starting or deceleration speed
2	1	preset speed > max. motor speed
3	1	destination position outside software limit switches
4	1	no ramp form set
5	1	no motor type set, no encoder resolution set
6	1	wrong regulator parameter set
7	1	RAM checksum error (all parameters are lost)

5.2.4.4. Positioning error byte (for example BM06.11)

Bit	Status	Description
0	1	left limit switch
1	1	right limit switch
2	1	software limit switch reached (manual mode)
3	1	undervoltage of power supply
4	1	deviation range of position regulator exceeded
5	1	absolute positioning mode no longer possible because axis in relative positioning mode has left counting range
6	1	counting range of absolute positioning mode left
7	1	emergency stop: activated either by the user via emergency stop function or by the system in combination with another error message

5.2.4.5. Motor failure byte (for example BM06.12)

Bit	Status	Description
0	1	no motor enable signal
1	1	wrong direction of motor rotation
2		unused
3	1	set current value range exceeded
4	1	no motor rotation (motor jammed)
5	1	no motor current
6	1	short-circuit of motor current
7	1	short-circuit of motor brake

5.3. KUBES modules (to define data blocks for the master)

KUBES modules are programs which take over komplex tasks in Kuhnke controllers (KUAX 680I, 644 or 657P).

You can define up to 10 data blocks per master using VEBES, the network configuration and operating software: blocks no. 160...169 (corresp. to OD indices 160...69).
These data blocks are operand ranges whose starting address (e.g. BM00.00) and length (e.g. 16) is set by the user.
During block transfer, the data contained in such a block (own block no.) is transferred to the block of another station (partner block no.), i.e. KUAX 681M in this case. You must make sure to use the set lengths of the data blocks.

All block transfer routines are taken care of by means of three KUBES modules:

- PB_SEND
KUBES module for sending a data block to the communication partner.
- PB_REC
KUBES module for receiving a data block from the communication partner.
- PB_STAT
KUBES module for requesting a status report about a task.

Use and application of these KUBES modules in the control device is described in detail elsewhere:

Instruction Manual
PROFIBUS
E 365 GB

Please refer to the appendix, chapter "D. Example of a PROFIBUS program", to find the printout of an example program.

5.4. Communication relations

Each PROFIBUS station defines logical channels (service access points, or SAPs) for communication with other stations. Initially, these channels are independent of their future use. All of these logical channels are entered into a list, the so-called communication reference list (CRL).
The KUAX 681M automatically creates the CRL when you switch on the supply voltage.

5.4.1. Master-Slave acyclic, no Event Notification

Both read and write are valid services. However, no error message is sent to the master:

Local LSAP	Type	ATTR	RSAP	RADR	FMS-Features-Sup Requ.	FMS-Features-Sup Resp.	SCC	RCC	SAC	RAC	ACI	CCI
4	MSAC	D	ALL	ALL	00 30 00	00 00 00	1	0	0	0	0	0

Maximum length of PDU	high	low
Send.:	0	128
Rec.:	0	241

5.4.2. Master-Slave acyclic, with Event Notification

Both read and write are valid services. The event notification is also the vehicle for sending error messages to the master:

Local LSAP	Type	ATTR	RSAP	RADR	FMS-Features-Sup Requ.	FMS-Features-Sup Resp.	SCC	RCC	SAC	RAC	ACI	CCI
5	MSAC_SI	D	ALL	ALL	00 30 00	400090	1	0	0	1	0	0

Maximum length of PDU	high	low
Send.:	0	128
Rec.:	241	241

We recommend using communication reference (CR) 3 to establish the connection to the KUAX 681M to ensure that an event notification is sent if an error occurs during polling.

5.5. Object dictionary (OD)

In PROFIBUS, two devices communicate via so-called objects. These objects can differ depending on the device type and the configuration used.
All available objects are listed in an object dictionary including their relevant attributes such as data type, access rights, access code etc.
In the KUAX 681M, the data blocks described in chapter 5.2. represent the objects. The corresponding information is listed in the object dictionary of the KUAX 681M. Every other bus station that belongs to a client profile class (master) can use the GetOD service to read these object dictionary.

Objects of the KUAX 681M:

OD Index	Data Type	Length [Bytes]	Object Code	Access	Function
40	Unsigned16	2	Simp_Var	R&W	Min station delay time
41	Octet_String	16	Simp_Var	R&W	commands 1 (write only)
42	Octet_String	16	Simp_Var	R&W	commands 2 (write only)
43	Octet_String	64	Simp_Var	R&W	parameters (write only)
44	Octet_String	32	Simp_Var	read only	status (read only)
50	Octet_String	2	Simp_Var	event	Event Notification. The KUAX 681M uses this to report any occurring errors: Byte 1: error code (see ch. "3.4.1. System error...") Byte 2: own station address

Communication with the KUAX 681M is made by block transfer via indices 41 to 44. Refer to chapter "5.2. Data blocks" to learn how the individual data blocks are structured.

5.6. Valid services

The KUAX 681M supports the following PROFIBUS services:

Service	Description
Get_OD	get object dictionary
Read	read objects
Write	write into objects
Identify	read device information
Event Notification	message about event
Initiate	initiate connection
Abort	abort connection
Reject	reject illegal PCU (FMS, not user)

A. Technical specifications

A.1. KUAX 681M

Function ..... decentralized positioning controller, controlled via field bus
Application ..... servo drives (see chapter 1.1.)
Field bus protocol ..... PROFIBUS-FMS
Inputs
- external incremental encoder ..... 500 Inc./rev, channels A, B and Ref. imp., 5 V TTL
- Hall sensor built-in in the motor ..... 2 channels, 5 V
- limit switches ..... 2 (left - right)
- reference switch (Ref. enable) 1
- optional ..... 2, user-definable function

Outputs .....
- transistor half bridges for motor control ..... 4 (24 or 40 V / 8A)
- power supply for ext. incr. encoder ... 5 V
- quiescent or holding current brake .... max. 500 mA

Software features
- motor commutation ..... exists
- counting depth ..... 31 bit plus sign bit
- position regulator ..... adjustable
- unidirectional gear positioning ..... exists
- manual positioning function ..... exists
- ramp lengths ..... user-definable
- positioning step ..... absolute or relative
- anti-blocking protection ..... exists
- software limit switches ..... exist
- emergency stop function ..... exists
- diagnosis function ..... exists
- motor current monitoring ..... exists

Power supply
- KUAX 681M ..... 24 V DC ± 20%
- motor ..... 24 or 40 V DC (max. 300 W)

Degree of protection ..... IP 54 (with cable connect. plugged in)
Admissible ambient temperature ..... 0 ... 55 °C

A.2. Motors

	Electronic DC Motors				
Type	Unit	Note	BG 63 x 55		BG 83 x 90
Nominal voltage(s)	V		24	40 V	40 V
Nominal speed	min^{-1}	2)	3300	3650	2750
Nominal torque	Ncm	2)	18	18	62
Nominal current	A	2)	≤ 4.5	≤ 2.9	≤ 7.2
Demagnetiz. current	A	1)	32	19.8	39
Open circuit speed	min^{-1}	1)	3950	4100	3450
Open circuit current	A	1)	≤ 0.66	≤ 0.44	≤ 0.68
Starting torque	Ncm	1)	≥ 110	≥ 140	≥ 410
Nom. power output	W	2)	62.2	68.8	179
Efficiency	%	2)	65	67.5	66.5
Moment of inertia	gcm^2		600	600	2610
Weight	kg		1.95	1.95	5.15

All values listed in the table refer to
1) J_R = 20 °C or
2) DJ_w = 110 K
and operation with positioning controller KUAX 681M

Excerpt from:
Catalogue "Dunker-made Motors"
by kind permission of the Standard Elektrik Lorenz AG, Components Division, Motors and Drives Product Range

B. Order information

B.1. Control devices KUAX 681M

The Profi Control KUAX 681M ships in two different device types:

For Motor BG 63 x 55:
24 V DC..... 681.000.00
40 V DC..... 681.001.00
For Motor BG 83 x 90:
40 V DC..... 681.001.00

Various additional accessories are required for fitting and connecting the device (→)

B.2. Accessories

B.2.1. Mounting material for KUAX 681M - motor

You need the following parts to attach the control device to the motor (see ch. "3.1.3. Mounting..."):

1 adapter (aluminium)
4 screws M4 adapter - motor
4 screws M4 KUAX 681M - adapter
8 tooth lock washer
2 seals

Mounting material for KUAX 681M - motor BG 63 x 55 ..... 681.180.02

mounting material for KUAX 681M - motor BG 83 x 90 ..... 681.180.01

B.2.2. Plug-type connectors

The electrical connections are made through plug-type connectors (see ch. "3.3. Electrical connectors").
The male and female cable connectors required for the connections must be ordered separately. The degree of protection for all plug-type connectors is IP 65.
The plug-type connectors come to you without cables. It is possible to order pre-assembled connectors for connection 4, 5, 6 and 8. The Lumberg company supplies these with various cable lengths.

The IP 65 degree of protection of the controller KUAX 681M can only be guaranteed if alle plug-type connectors have been properly plugged in.
Cover unused connector locations with appropriate cover caps.

Plug-type connectors without cables

For connectors #1 (bus in), #2 (bus out) and #4 "option":
- 4pin male cable connector 681.180.08
  alternative *1): miniat. round plug-type connector, series 07, type RSMC 4

For connectors #5 (reference switch), #6 and #8 (left and right limit switch):
- 3pin male cable connector 681.180.09
  alternative *1): miniat. round plug-type connector, series 07, type RSMC 3

For connector #3 (power supply):
- 6pin female cable connector 681.180.05
  alternative *2): round plug-type conn., series C 091 D, C 091 31 D 006 1002

For connector #7 (inc. encoder / brake):
- 7pin male cable connector 681.180.07
  alternative *2): round plug-type conn., series C 091 D, C 091 31 H 107 1002

Pre-assembled connectors with cable *1)

Cable	Sheathing mat. PVC	Sheathing mat. PUR
3pin, for connectors 5, 6 and 8		
2 m cable	RSMV 3 - 62 / 2.0	RSMV 3 - 61 / 2.0
5 m cable	RSMV 3 - 62 / 5.0	RSMV 3 - 61 / 5.0
7.5 m cable	RSMV 3 - 62 / 7.5	RSMV 3 - 61 / 7.5
10 m cable	RSMV 3 - 62 / 10.0	RSMV 3 - 61 / 10.0
4pin, for connector 4		
2 m cable	RSMV 4 - 07 / 2.0	RSMV 4 - 21/ 2.0
5 m cable	RSMV 4 - 07 / 5.0	RSMV 4 - 21 / 5.0
7.5 m cable	RSMV 4 - 07 / 7.5	RSMV 4 - 21 / 7.5
10 m cable	RSMV 4 - 07 / 10.0	RSMV 4 - 21 / 10.0

Spare cover caps (IP 65) for unused connectors

1 piece, fits connectors 1, 2, 4, 5, 6, 8 ..... 681.180.11
1 piece, fits connector 7 ..... 681.180.12

Sources of supply:
*1) Karl Lumberg GmbH & Co., Postfach Box 1360, W-5885 Schalksmühle
*2) Amphenol-Tuchel Electronics GmbH, August-Hausser-Str. 10, W-7100 Heilbronn

B.2.3. Bus cable

Cables with particular physical properties are compulsory for making the bus connection.

We suggest using the following:
Type ..... Alfaskop S 41
Product number 141980
Make ..... Dätwyler System- und Netzwerk GmbH
Gottfried von Cramm Str. 1
D-85375 Neufahrn

or:
Type ..... UNITRONIC®-BUS LD
Product number 2170203
Cores ..... 1 x 2 x 0.22 mm²
Make ..... LAPP GmbH Kabelwerk
Schulte-Delitzsch-Str. 25
D-70565 Stuttgart

C. Further reading and trademarks

C.1. Further reading

Instruction Manual E 365 GB
PROFIBUS
for Kuhnke Controllers
Kuhnke GmbH

Instruction Manual E 315 GB
VEBES
PROFIBUS Network Operating Software
Kuhnke GmbH

Motors Catalogue
Dunker-made Motors
Standard Elektrik Lorenz AG, Components Division, Motors and Drives Product Range

DIN 19 245 Part 1
PROFIBUS
Transmission Techniques, Protocol for Bus Access and Transmission, Service Interface to the Application Layer, Management
Beuth Verlag, Berlin

DIN 19 245 Part 2
PROFIBUS
Communication Model, Services for Applications, Protocol, Syntax, Coding, Interface to the Data Link Layer, Management
Beuth Verlag Berlin

PROFIBUS
The Fieldbus for Industrial Automation
Klaus Bender (Ed.)
Prentice Hall International (UK) Ltd, Hemel Hempstead
ISBN 13-012691-8 (hbk)

C.2. Trademarks

IBM
is a registered trademarks of the International Business Machines Corporation

MS-DOS
is a registered trademark of the Microsoft Corporation

EPSON
is a registered trademark of the Epson Corporation

D. Example of a PROFIBUS program

```
======== Kubes ================================== KUAX 680I =======
                        Project structure
Project : PLC680I               Network  : PB_BSP_8
                                 created : Mar 11 1996 09:20
Use     : Paul Posi              edite   : Mar 11 1996 14:53
Comment  : Example program 680I<->681M, one axis

=====================================================================
ORG.ORG/1
|
*---->MOT_INIT.PRO/1
|       |
|        *---->M1_PARA.INI/1
|
*---->MOT_ENAB.PRO/2
|
*---->MO_AUT.PRO/3
|
*---->MOT_REF.PRO/4
|
*--->MOT_STOP.PRO/5
|
*---->MO_START.PRO/6
|
*---->MOT_MAN.PRO/7
|
*---->BLCTRA_M.PRO/8
|       |
|        *---->STATUS.PRO/9
|       |
|        *---->PB_REC.KUN/3
|       |
|        *---->PB_SEND.KUN/1
|       |
|        *---->PB_STAT.KUN/2
```

```
======== Kubes ================================== KUAX 680I =======
                        Symbol table
Project : PLC680I               Network   : PB_BSP_8
                                 created  : Mar 11 1996 09:20
User    : Paul Posi              edited   : Mar 11 1996 14:53
Comment : Example program 680I<–>681M, one axis

=====================================================================
Address:    Symbol:     Comment:                    Supplement:
BM00.00    | CNTSET_0 | counter preset value, LL            |
BM00.01    | CNTSET_1 | counter preset value, LH            |
BM00.02    | CNTSET_2 | counter preset value, HL            |
BM00.03    | CNTSET_3 | counter preset value, HH            |
BM00.04    | REL_RF_0 | relative point of reference, LL     |
BM00.05    | REL_RF_1 | relative point of reference, LH     |
BM00.06    | REL_RF_2 | relative point of refernce, HL      |
BM00.07    | REL_RF_3 | relative point of reference, HH     |
BM00.08    | LIM_L_0  | left limit switch, LL               |
BM00.09    | LIM_L_1  | left limit switch, LH               |
BM00.10    | LIM_L_2  | left limit switch, HL               |
BM00.11    | LIM_L_3  | left limit switch, HH               |
BM00.12    | LIM_R_0  | right limit switch, LL              |
BM00.13    | LIM_R_1  | right limit switch, LH              |
BM00.14    | LIM_R_2  | right limit switch, HL              |
BM00.15    | LIM_R_3  | right limit switch, HH              |
Address:    Symbol:     Comment:                    Supplement:
BM01.00    | ST_RPL   | starting ramp left                  |
BM01.01    | BM03_01  | starting ramp left                  |
BM01.02    | STP_RPL  | stopping ramp left                  |
BM01.03    | BM03_03  | stopping ramp left                  |
BM01.04    | ST_RPR   | starting ramp right                 |
BM01.05    | BM03_05  | starting ramp right                 |
BM01.06    | STP_RPR  | stopping ramp right                 |
BM01.07    | BM03_07  | stopping ramp right                 |
BM01.08    | V_STRT_L | starting speed left                 |
BM01.09    | BM03_09  | starting speed left                 |
BM01.10    | V_STOP_L | deceleration speed left             |
BM01.11    | BM03_11  | deceleration speed left             |
BM01.12    | V_STRT_R | starting speed right                |
```

```
BM01.13    | BM03_13  | starting speed right                |
BM01.14    | V_STOP_R | deceleration speed right            |
BM01.15    | BM03_15  | deceleration speed right            |
Address:    Symbol:     Comment:                    Supplement:
BM02.00    | ENC_RES  | encoder resolution                  |
BM02.01    | BM04_01  | encoder resolution                  |
BM02.02    | P_DIV    | P part of speed regulator, factor   |
BM02.03    | P_DIVS   | P part, divisor                     |
BM02.04    | TA       | scanning frequency speed regulator  |
BM02.05    | TA_HB    | scanning frequency speed regulator  |
BM02.06    | TN       | adjustment time speed regulator     |
BM02.07    | TN_HB    | adjustment time speed regulator     |
BM02.08    | TV       | suspension time speed regulator     |
BM02.09    | TV_HB    | suspension time speed regulator     |
BM02.10    | BM04_10  | unused                              |
BM02.11    | KPP      | KP position regulator               |
BM02.12    | BM04_12  | unused                              |
BM02.13    | TNP      | I part position regulator           |
BM02.14    | POS_RNG  | position regulation range           |
BM02.15    | PS_RNG_H | position regulation range           |
Address:    Symbol:     Comment:                    Supplement:
BM03.00    | I_NOM    | nominal motor current, mA           |
BM03.01    | BM05_01  | nominal motor current, mA           |
BM03.02    | I_GO     | motor startup current, mA           |
BM03.03    | BM05_03  | motor startup current, mA           |
BM03.04    | V_MAX    | maximum motor speed, rpm            |
BM03.05    | V_MAX_HB | maximum motor speed, rpm            |
BM03.06    | T_OUT    | timeout (Net_i only)                |
BM03.07    | BM05_07  | timeout (Net_i only)                |
BM03.08    | UGP      | unidirectional gear position., inc. |
BM03.09    | UGP_HB   | unidirectional gear position., inc. |
BM03.10    | BM05_10  | unused                              |
BM03.11    | SET_RAMP | ramp form                           |
BM03.12    | SET_TYP  | motor type                          |
BM03.13    | SYS_INI  | system settings, bit selection      |
BM03.14    | CNT_INI  | counter settings, bit selection     |
BM03.15    | SPE_FCT  | special function, bit selection     |
```

Address:	Symbol:	Comment:	Supplement:
BM04.00	PPOS_LL	preset position, LL	
BM04.01	PPOS_LH	preset position, LH	
BM04.02	PPOS_HL	preset position, HL	
BM04.03	PPOS_HH	preset position, HH	
BM04.04	VPOSB_L	preset motor speed, backward LB	
BM04.05	VPOSB_H	preset motor speed, backward HB	
BM04.06	VPOSF_L	preset motor speed, forward LB	
BM04.07	VPOSF_H	preset motor speed, forward, HB	
BM04.08	V_CHNG	change speed at position LL	
BM04.09	BM00_09	change speed at position LH	
BM04.10	BM00_10	change speed at position HL	
BM04.11	BM00_11	change speed at position HH	
BM04.12	BM00_12	unused	
BM04.13	BM00_13	unused	
BM04.14	SYS_BYTE	system control byte, bit analysis	
BM04.15	COM_BYTE	command byte, bit analysis	
Address:	Symbol:	Comment:	Supplement:
BM05.00	BM05_00	block is free at present	
Address:	Symbol:	Comment:	Supplement:
BM06.00	CNT_LL	position counter LL	
BM06.01	CNT_LH	position counter LH	
BM06.02	CNT_HL	position counter HL	
BM06.03	CNT_HH	position counter HH	
BM06.04	V_ACT	actual speed LB	
BM06.05	V_ACT_HB	actual speed HB	
BM06.06	I_ACT	actual motor current	
BM06.07	I_ACT_HB	actual motor current	
BM06.08	TASK	program step in position monitor	
BM06.09	STATUS_P	status of positioning	
BM06.10	PAR_ERR	parameter error	
BM06.11	POS_ERR	positioning error	
BM06.12	MOT_FAIL	motor failure	
BM06.13	COMECHO	command echo	
BM06.14	CNTECHO	motor control echo	
BM06.15	SPFECHO	special function echo	
Address:	Symbol:	Comment:	Supplement:
BM07.00	P_POS	preset position (read)	
BM07.01	BM06_01	LH byte	
BM07.02	BM06_02	HL byte	

```
BM07.03    | BM06_03  | HH byte                              |
BM07.04    | P_SPEED  | preset speed (read) 11               |
BM07.05    | BM06_07  | L_byte                               |
BM07.06    | IRQ_POS  | position at IRQ input 11 byte        |
BM07.07    | BM06_09  | LH byte                              |
BM07.08    | BM06_10  | HL byte                              |
BM07.09    | BM06_11  | HH byte                              |
BM07.10    | BM06_12  | unused                               |
BM07.11    | BM06_13  | unused                               |
BM07.12    | HARDW    | hardware identification              |
BM07.13    | SOFTW    | software identification              |
BM07.14    | BM06_14  | software identification              |
BM07.15    | BM06_15  | software identification              |
Address:    Symbol:     Comment:                     Supplement:
BM08.00    | I_LOK_S  | block number of sender, LB           |
BM08.01    | I_LOK    | block number of sender, HB           |
BM08.02    | I_REMO_S | block number of receiver, LB         |
BM08.03    | I_REMO   | block number of receiver, HB         |
BM08.04    | STA_SEND | station address PB_SEND              |
BM08.05    | TSK_NO_S | task number of sender                |
BM08.06    | STAT_T   | status of transmission               |
BM08.07    | STA_REC  | station address PB_REC               |
BM08.08    | I_LOK_R  | receiver block number, LB            |
BM08.09    | I_LOK_R_ | receiver block number, HB            |
BM08.10    | I_REMO_R | block number of receiver             |
BM08.11    | I_REM    | block number receiver, HB            |
BM08.12    | TSK_NO_R | task number of receiver              |
BM08.13    | STAT_R   | status of receive                    |
BM08.15    | SEND_CNT | step counter, sending                |
Address:    Symbol:     Comment:                     Supplement:
BM09.00    | MOT_NO   | motor number                         |
Address:    Symbol:     Comment:                     Supplement:
I00.00     | MO_REF   | start reference run                  |
I00.01     | MO_STOP  | stop motor                           |
I00.02     | MO_START | start motor                          |
I00.03     | MO_ENAB  | motor enable                         |
I00.04     | ST_AUTO  | start automatic program              |
```

Address:	Symbol:	Comment:	Supplement:
M00.00	R_MARKER	controls block transfer, read	
M00.01	W_MARKER	initiates block transfer, write	
M00.02	HM_REF	HM for reference run	
M00.03	HM_MAN_R	HM manual operation, right	
M00.04	HM_MAN_L	HM manual operation, left	
M00.05	HM_ASTAR	HM automatic start	
M00.06	HM_START	HM start motor	
M00.07	HM_STOP	HM stop motor	
M00.08	HM_MSTPL	HM manual operation left, stop	
M00.09	HM_MSTPR	HM manual operation right, stop	
M00.10	MINI_1	transfer block no. 163	
M00.11	MINI_2	transfer block no. 161	
M00.12	INI_OK	initialization of motor ok	
M00.14	DEST_POS	motor destination position reached	
Address:	Symbol:	Comment:	Supplement:
M01.00	M_READY	motor ready (BM06.09, C8T1)	
M01.01	M_BUSY	motor busy (BM06.09, C8T1)	
M01.02	M_DEST	motor at dest. pos. (BM06.09, C8T1)	
M01.03	M_POSREG	position regulator on?	
M01.04	M_BRAKE	brake active?	
M01.05	M_IRQ_NW	new IRQ!	
M01.06	M_CNT_OK	counter is enabled	
M01.07	M_W_DOG	watchdog?	
Address:	Symbol:	Comment:	Supplement:
PP00.00	PP00_00	programmable pulse	
PP00.01	PP00_01	programmable pulse	
PP00.02	PP00_02	programmable pulse	
PP00.03	PP00_03	programmable pulse	
PP00.04	PP00_04	programmable pulse	
PP00.05	PP00_05	programmable pulse	
PP00.06	PP00_06	programmable pulse	
PP00.07	PP00_07	programmable pulse	
PP00.08	PP00_08	programmable pulse	
PP00.09	PP00_09	programmable pulse	
PP00.10	PP00_10	programmable pulse	

```
======== Kubes ================================== KUAX 680I =======
                        Organization module IL
Project : PLC680I                Network   : PB_BSP_8
Module  : ORG        No.: 1       created  : Mar 11 1996 09:20
User    : Paul Posi               edited   : Mar 11 1996 14:53
Comment : Main program

=====================================================================
 1: ; ————————
 2: ;        Transfer motor parameters
 3: ; ————————
 4: MO_INIT  L        INI_OK          M00.12 ; (initialization of motor ok)
 5:          JPC      MAIN
 6:
 7:          L        2
 8:          =        MOT_NO         BM09.00 ; (motor number)
 9:          JPP      MOT_INIT            1
10:
11: ; – Main program ——
12: MAIN     JPP      MOT_ENAB            2
13:          JPP      MO_AUT              3
14:          JPP      MOT_REF             4
15:          JPP      MOT_STOP            5
16:          JPP      MO_START            6
17:          JPP      MOT_MAN             7
18:          JPP      BLCTRA_M            8
19: ; – Transfer of motor number –
20:
21: END      NOP
22:
23:
```

```
======== Kubes ================================== KUAX 680I =======
                        Program module IL
Project : PLC680I                Network   : PB_BSP_8
Module  : BLCTRA_M  No.: 8        created  : Mar 11 1996 09:20
User    : Paul Posi               edited   : Mar 11 1996 14:53
Comment : Initiate block transfer

=====================================================================
   1: ; ***************************************************
  2: ;        (This module controls the block transfer.
  3: ;        In the example, the motor data are permanently
  4: ;        collected so that actual position and status messages
  5: ;        of the axis are available. Reading is only briefly
  6: ;        interrupted by sending command block 161 (PLC)
  7: ;        to 41 (motor).
  8: ;        Communication is designed to be cyclic but
  9: ;        on command rather.
 10: ;        Write marker M00.01 initiates the sending operation.
 11: ;        Where necessary it is set with memory function
 12: ;        in the user program and reset here after block
 13: ;        transfer has been completed successfully.
  14: ; ***************************************************
 15: ; ----------
 16: ;        Only valid for monitor versions smaller 4.20!!
 17: ;        The next three program lines must be excluded when
 18: ;        working with monitor versions as from 4.20.
 19: ; ----------
 20: ;        l       psa00.02              ; is station #2 active?
 21: ;        CMP     0
 22: ;        JP<>    END                   ; error treatment
 23:
 24: ; ----------
 25: ;        Read and write
 26: ;        step chain control for block.
 27: ; ----------
 28:          L       SEND_CNT       BM08.15 ; (step counter, sending)
 29:          CMP     0
 30:          JP=     SEND_GO               ; start send job
 31:          CMP     1
 32:          JP=     SEND_STA              ; send job status ok?
```

```
 33:           CMP      2
 34:           JP=      REC_GO                  ; start reading job
 35:           CMP      3
 36:           JP=      REC_STAT                ; reading job status ok?
 37:           JP       END
 38:
 39: ; ————————————
 40: ;         Transfer block
 41: ; ————————————
 42: SEND_GO   L        W_MARKER         M00.01 ; (initiates block transfer,
write)
 43:           JPCN     REC_GO
 44:
 45:           JPK      PB_SEND   ,   _____
 45:                      STA_SEND -|     |- TSK_NO_S,
 45:                      I_LOK_S  -|     |- STAT_T,
 45:                      I_REMO_S -|_____|-
 46:
 47:           L        STAT_T          BM08.06 ; (status of transmission)
 48:           CMP      41
 49:           JP=      END
 50:           INC      SEND_CNT        BM08.15 ; (step counter, sending)
 51:           JP       END
 52:
 53: SEND_STA JPK      PB_STAT   ,   _____
 53:                      TSK_NO_S -|_____|- STAT_T
 54:
 55: ; – Analysis of task status –
 56:           L        STAT_T          BM08.06 ; (status of transmission)
 57:           CMP      40
 58:           JP=      END
 59:           CMP      7                       ; task status ok
 60:           JP=      SEND_OK
 61:           CLR      SEND_CNT        BM08.15 ; (step counter, sending)
 62:           JP       END
 63:
 64: SEND_OK   NOP
 65:           =0       W_MARKER         M00.01 ; (initiates block transfer,
write)
 66:           CLR      SEND_CNT        BM08.15 ; (step counter, sending)
 67:
```

```
 68: ; ————————
 69: ;         Read block
 70: ; ————————
 71: REC_GO    L       INI_OK          M00.12 ; (initialization of motor ok)
 72:           JPCN    REC_NOK
 73:           L       2
 74:           =       SEND_CNT        BM08.15 ; (step counter, sending)
 75:
 76:           L       MOT_NO          BM09.00 ; (motor number)
 77:           =       STA_REC         BM08.07 ; (station address PB_REC)
 78:           LD      164                     ; status block in the PLC
 79:           =D      I_LOK_R         BM08.08 ; (receiver block number, LB)
 80:           LD      44                      ; status block in the motor
 81:           =D      I_REMO_R        BM08.10 ; (block number of receiver)
 82:
 83:           JPK     PB_REC   ,   _____
 83:                     STA_REC  -|     |- TSK_NO_R,
 83:                     I_LOK_R  -|     |- STAT_R,
 83:                     I_REMO_R -|_____|-
 84:           L       STAT_R          BM08.13 ; (status of re ceive)
 85:           CMP     41
 86:           JP=     END
 87:           INC     SEND_CNT        BM08.15 ; (step counter, sending)
 88:           JP      END
 89:
 90: ; — Request status ——
 91: REC_STAT JPK      PB_STAT  ,   _____
 91:                     TSK_NO_R -|_____|- STAT_R
 92:
 93:           L       STAT_R          BM08.13 ; (status of receive)
 94:           CMP     40
 95:           JP=     END
 96:           CMP     14                      ; get task data ok
 97:           JP=     REC_OK
 98: REC_NOK   CLR     SEND_CNT        BM08.15 ; (step counter, sending)
 99:           JP      END
100:
101: REC_OK    JPP     STATUS               9
102:           CLR     SEND_CNT        BM08.15 ; (step counter, sending)
103:
104: END       NOP
```

```
======== Kubes ================================== KUAX 680I =======
                        Program module IL
Project : PLC680I                Network   : PB_BSP_8
Module  : MOT_ENAB  No.: 2        created  : Mar 11 1996 09:20
User    : Paul Posi               edited   : Mar 11 1996 14:53
Comment : Motor enable after interruptions

=====================================================================
   1: ; ***************************************************
  2: ;         Manual motor enable.
  3: ;         After interruptions such as active limit switch
  4: ;         the motor enable signal must be repeated
  5: ;         for restart.
  6: ;         - in this case: re-initialise motor
  7: ;         - restart by I00.03
   8: ; ***************************************************
  9:           L       MO_ENAB          I00.03 ; (motor enable)
 10:           =       PP00_07         PP00.07 ; (programmable pulse)
 11:           L       PP00_07         PP00.07 ; (programmable pulse)
 12:           JPCN    ME_END
 13:           =0      MINI_1           M00.10 ; (transfer block no. 163)
 14:           =0      MINI_2           M00.11 ; (transfer block no. 161)
 15:           =0      INI_OK           M00.12 ; (initialization of motor ok)
 16: ME_END    NOP
```

```
======== Kubes ================================== KUAX 680I =======
                        Program module IL
Project : PLC680I                Network   : PB_BSP_8
Module  : MOT_MAN   No.: 7        created  : Mar 11 1996 09:20
User    : Paul Posi               edited   : Mar 11 1996 14:53
Comment : M2_MAN

=====================================================================
   1: ; ****************************************************
  2: ;         Manual positioning runs left and right
  3: ;         while the software limit switches are active
   4: ; ****************************************************
  5:
  6: ; ———————————
  7: ;           Start manual positioning run left
  8: ; ———————————
  9:          L       HM_MAN_L        M00.04 ; (HM manual op eration, left)
 10:          JPC     LE_STOP                ; if prog. pulse is not received
 11:
 12:          L       I00.05
 13:          =       PP00_05        PP00.05 ; (programmable pulse)
 14:          L       PP00_05        PP00.05 ; (programmable pulse)
 15:          JPCN    LE_STOP
 16:          =       HM_MAN_L        M00.04 ; (HM manual operation, left)
 17:
 18: ; — Block transfer active ?? —
 19: L_ST     L       W_MARKER        M00.01 ; (initiates block transfer,
write)
 20:          JPC     LE_STOP
 21: ; — Set V_Man —
 22:          LD      500                    ; rev./min, speed of man. pos. run
 23:          =D      VPOSB_L        BM04.04 ; (preset motor speed, backward
LB)
 24:          =D      VPOSF_L        BM04.06 ; (preset motor speed, forward LB)
 25:
 26: ; — Set control bytes —
 27:          L       %0000_0000
 28:          =       SYS_BYTE       BM04.14 ; (system control byte, bit analy-
sis)
 29:          L       %0000_1000
 30:          =       COM_BYTE       BM04.15 ; (command byte, bit analysis)
```

```
 31:
 32: ; — Send data via the bus —
 33:          L       MOT_NO          BM09.00 ; (motor number)
 34:          =       STA_SEND        BM08.04 ; (station address PB_SEND)
 35:          LD      161                     ; command block in the PLC
 36:          =D      I_LOK_S         BM08.00 ; (block number of sender, LB)
 37:          LD      41                      ; command block in the motor
 38:          =D      I_REMO_S        BM08.02 ; (block number of receiver, LB)
 39:          =1      W_MARKER         M00.01 ; (initiates block transfer,
write)
 40:          =0      HM_MAN_L         M00.04 ; (HM manual operation, left)
 41:
 42: ; ———————————
 43: ;         Stop manual positioning run left
 44: ; ———————————
 45: LE_STOP  L       HM_MSTPL         M00.08 ; (HM manual operation left, stop)
 46:          JPC     W_2
 47:          LN      I00.05
 48:          =       PP01.05
 49:          L       PP01.05
 50:          JPCN    LE_END
 51:
 52:          =1      HM_MSTPL         M00.08 ; (HM manual operation left, stop)
 53:
 54: W_2      L       W_MARKER         M00.01 ; (initiates block transfer,
write)
 55:          JPC     LE_END
 56:
 57: ; — Set control bytes ——
 58:          L       %0000_0000
 59:          =       SYS_BYTE        BM04.14 ; (system control byte, bit analy-
sis)
 60:          L       %0000_0000
 61:          =       COM_BYTE        BM04.15 ; (command byte, bit analysis)
 62:
 63: ; — Send data via the bus ——
 64:          L       MOT_NO          BM09.00 ; (motor number)
 65:          =       STA_SEND        BM08.04 ; (station address PB_SEND)
 66:          LD      161                     ; command block in the PLC
 67:          =D      I_LOK_S         BM08.00 ; (block number of sender, LB)
 68:          LD      41                      ; command block in the motor
```

```
 69:           =D      I_REMO_S       BM08.02 ; (block number of receiver, LB)
 70:           =1      W_MARKER        M00.01 ; (initiates block transfer,
write)
 71:           =0      HM_MAN_L        M00.04 ; (HM manual operation, left)
 72:           =0      HM_MSTPL        M00.08 ; (HM manual operation left, stop)
 73: LE_END    NOP
 74: ;                  * * *
 75:
 76:
 77: ; ————————
 78: ;         Start manual positioning run right
 79: ; ————————
 80:           L       HM_MAN_R        M00.03 ; (HM manual operation, right)
 81:           JPC     R_ST
 82:           L       I00.06
 83:           =       PP00_06        PP00.06 ; (programmable pulse)
 84:           L       PP00_06        PP00.06 ; (programmable pulse)
 85:           JPCN    RI_STOP
 86:           =1      HM_MAN_R        M00.03 ; (HM manual operation, right)
 87:
 88: R_ST      L       W_MARKER        M00.01 ; (initiates block transfer,
write)
 89:           JPC     RI_STOP
 90:
 91:           LD      500                    ; rev/min., speed of man. pos. run
 92:           =D      VPOSB_L        BM04.04 ; (preset motor speed, backward
LB)
 93:           =D      VPOSF_L        BM04.06 ; (preset motor speed, forward LB)
 94:
 95: ; — Set control bytes ——
 96:           L       %0000_0000
 97:           =       SYS_BYTE       BM04.14 ; (system control byte, bit analy-
sis)
 98:           L       %0100_0100
 99:           =       COM_BYTE       BM04.15 ; (command byte, bit analysis)
100:
101: ; — Send data via the bus ——
102:           L       MOT_NO         BM09.00 ; (motor number)
103:           =       STA_SEND       BM08.04 ; (station address PB_SEND)
104:           LD      161                    ; command block in the motor
105:           =D      I_LOK_S        BM08.00 ; (block number of sender, LB)
```

```
106:          LD      41                    ; command block in the PLC
107:          =D      I_REMO_S       BM08.02 ; (block number of receiver, LB)
108:          =1      W_MARKER        M00.01 ; (initiates block transfer,
write)
109:          =0      HM_MAN_R        M00.03 ; (HM manual operation, right)
110:
111: ; ———————————————
112: ; Stop manual positioning run right
113: ; ———————————————
114: RI_STOP  L       HM_MSTPR        M00.09 ; (HM manual operation right,
stop)
115:          JPC     W_3
116:          LN      I00.06
117:          =       PP01.06
118:          L       PP01.06
119:          JPCN    RI_END
120:          =1      HM_MSTPR        M00.09 ; (HM manual operation right,
stop)
121:
122: W_3      L       W_MARKER        M00.01 ; (initiates block transfer,
write)
123:          JPC     RI_END
124:
125: ; —— Set control bytes ———
126:          L       %0000_0000
127:          =       SYS_BYTE       BM04.14 ; (system control byte, bit analy-
sis)
128:          L       %0000_0000
129:          =       COM_BYTE       BM04.15 ; (command byte, bit analysis)
130:
131: ; —— Send data via the bus ——
132:          L       MOT_NO         BM09.00 ; (motor number)
133:          =       STA_SEND       BM08.04 ; (station address PB_SEND)
134:          LD      161                    ; command block in the PLC
135:          =D      I_LOK_S        BM08.00 ; (block number of sender, LB)
136:          LD      41                     ; command block in the motor
137:          =D      I_REMO_S       BM08.02 ; (block number of receiver, LB)
138:          =1      W_MARKER        M00.01 ; (initiates block transfer,
write)
139:          =0      HM_MSTPR        M00.09 ; (HM manual operation right,
stop)
140: RI_END   NOP
```

```
======== Kubes ================================== KUAX 680I =======
                        Program module IL
Project : PLC680I                Network   : PB_BSP_8
Module  : MOT_INIT  No.: 1        created  : Mar 11 1996 09:20
User    : Paul Posi               edited   : Mar 11 1996 14:53
Comment : Transfer motor parameters for motor #2

=====================================================================
   1: ; ***************************************************
  2: ;         Initialise motor.
  3: ;         Block -parameters- (64 byte) incl the INI data
  4: ;         is transferred once to motor #2.
   6: ; ***************************************************
  7:
  7: TRANS    L        MINI_1          M00.10 ; (transfer block no. 163)
  8:          JPC      L_1
  9:          L        W_MARKER        M00.01 ; (initiates block transfer,
write)
 10:          JPC      END
 11:
 12: ; — Parameters from INIT module –
 13:          JPINIT   M1_PARA             1
 14:
 15: ; — Send data via the bus ——
 16:          L        MOT_NO         BM09.00 ; (motor number)
 17:          =        STA_SEND       BM08.04 ; (station address PB_SEND)
 18:          LD       163                    ; command block in the PLC
 19:          =D       I_LOK_S        BM08.00 ; (block number of sender, LB)
 20:          LD       43                     ; command block in the motor
 21:          =D       I_REMO_S       BM08.02 ; (block number of receiver, LB)
 22:          =1       W_MARKER        M00.01 ; (initiates block transfer,
write)
 23:          =1       MINI_1          M00.10 ; (transfer block no. 163)
 24:          JP       END
 25:
```

```
 26:  ;  *****************************************************
 27: ;         Block -commands- (16 byte) incl. the motor
 28: ;         parameters is transferred twice to motor #2.
 29: ;         You must then set ramp forms, motor type, system
 30: ;         settings, counter settings, special settings for
 31: ;         the selected motor.
 32: ;         Bit 7 -accept special settings- of the command
 33: ;         byte (BM04.15) must be set.
 34:  ;  *****************************************************
 35:
 36: L_1      L        W_MARKER        M00.01 ; (initiates block transfer,
write)
 37:          JPC      END                     ; send task not yet completed
 38:
 39:          L        MINI_2          M00.11 ; (transfer block no. 161)
 40:          JPC      END
 41:
 42: ; — Send data via the bus ——
 43:          L        %0000_0000
 44:          =        SYS_BYTE        BM04.14 ; (system control byte, bit analy-
sis)
 45:          L        %1000_0000
 46:          =        COM_BYTE        BM04.15 ; (command byte, bit analysis)
 47:
 48:          L        MOT_NO          BM09.00 ; (motor number)
 49:          =        STA_SEND        BM08.04 ; (station address PB_SEND)
 50:          LD       161                     ; command block in the PLC
 51:          =D       I_LOK_S         BM08.00 ; (block number of sender, LB)
 52:          LD       41                      ; command block in the motor
 53:          =D       I_REMO_S        BM08.02 ; (block number of receiver, LB)
 54:          =1       W_MARKER         M00.01 ; (initiates block transfer,
write)
 55:          =1       MINI_2           M00.11 ; (transfer block no. 161)
 56:          =1       INI_OK           M00.12 ; (initialization of motor ok)
 57: END      NOP
 58:
 59:
```

```
======== Kubes ================================== KUAX 680I =======
                        Program module IL
Project : PLC680I                Network   : PB_BSP_8
Module  : MOT_REF   No.: 4        created  : Mar 11 1996 09:20
User    : Paul Posi               edited   : Mar 11 1996 14:53
Comment : Initiate reference run

=====================================================================
   1: ; ***************************************************
  2: ;         Motor reference run left
  3: ;         Start by input I00.00
   4: ; ***************************************************
  5:
  6:          L       HM_REF          M00.02 ; (HM for reference run)
  7:          JPC     REF_GO                 ; if prog. pulse is not received
  8:
  9:
 10:          L       MO_REF          I00.00 ; (start reference run)
 11:          =       PP00_00        PP00.00 ; (programmable pulse)
 12:          L       PP00_00        PP00.00 ; (programmable pulse)
 13:          JPCN    REF_END
 14:          =1      HM_REF          M00.02 ; (HM for reference run)
 15:
 16: ; — Block transfer still active ?? ——
 17: REF_GO   L       W_MARKER        M00.01 ; (initiates block transfer,
write)
 18:          JPC     REF_END
 19:
 20: ; — Reference speed as rpm ——
 21:          LD      750                    ; rev/min
 22:          =D      VPOSB_L        BM04.04 ; (preset motor speed, backward
LB)
 23:          =D      VPOSF_L        BM04.06 ; (preset motor speed, forward LB)
 24:
 25: ; — Set system control byte ——
 26:          L       %0000_0000
 27:          =       SYS_BYTE       BM04.14 ; (system control byte, bit analy-
sis)
 28:
 29: ; — Set control byte —
 30:          L       %0000_0001             ; reference run left
```

```
 31:           =       COM_BYTE       BM04.15 ; (command byte, bit analysis)
 32:
 33: ; — Starting command via the bus ——
 34:           L       MOT_NO         BM09.00 ; (motor number)
 35:           =       STA_SEND       BM08.04 ; (station address PB_SEND)
 36:           LD      161                    ; command block in the PLC
 37:           =D      I_LOK_S        BM08.00 ; (block number of sender, LB)
 38:           LD      41                     ; command block in the motor
 39:           =D      I_REMO_S       BM08.02 ; (block number of receiver, LB)
 40:           =1      W_MARKER        M00.01 ; (initiates block transfer,
write)
 41:           =0      HM_REF          M00.02 ; (HM for reference run)
 42: REF_END   NOP
 43:
 44:
 45:
```

```
======= Kubes ================================== KUAX 680I =======
                        Program module IL
Project : PLC680I               Network   : PB_BSP_8
Module  : MOT_STOP  No.: 5       created  : Mar 11 1996 09:20
User    : Paul Posi              edited   : Mar 11 1996 14:53
Comment : Motor stop

=====================================================================
   1: ; ***************************************************
  2: ;        Stop motor with ramp
  3: ;        - stop by I00.01
   4: ; ***************************************************
  5:          L       HM_STOP         M00.07 ; (HM stop motor)
  6:          JPC     STP_GO                 ; if prog. pulse is not received
  7:
  8:
  9:          L       MO_STOP         I00.01 ; (stop motor)
 10:          =       PP00_01        PP00.01 ; (programmable pulse)
 11:          L       PP00_01        PP00.01 ; (programmable pulse)
 12:          JPCN    STP_END
 13:          =1      HM_STOP         M00.07 ; (HM stop motor)
 14:          =0      HM_ASTAR        M00.05 ; (HM automatic start)
 15:
 16: ; — Block transfer still active ?? ——
 17: STP_GO   L       W_MARKER        M00.01 ; (initiates block transfer,
write)
 18:          JPC     STP_END
 19:
 20: ; — Set stop bit of system control byte ——
 21:          L       %0000_0010             ; stop with ramp
 22:          =       SYS_BYTE       BM04.14 ; (system control byte, bit analy-
sis)
 23: ; — Set command byte to zero ——
 24:          L       %0000_0000
 25:          =       COM_BYTE       BM04.15 ; (command byte, bit analysis)
 26:
```

```
 27: ; — Stop motor ——
 28:           L       MOT_NO          BM09.00 ; (motor number)
 29:           =       STA_SEND        BM08.04 ; (station address PB_SEND)
 30:           LD      161                     ; command block in the PLC
 31:           =D      I_LOK_S         BM08.00 ; (block number of sender, LB)
 32:           LD      41                      ; command block in the motor
 33:           =D      I_REMO_S        BM08.02 ; (block number of receiver, LB)
 34:           =1      W_MARKER         M00.01 ; (initiates block transfer,
write)
 35:           =0      HM_STOP          M00.07 ; (HM stop motor)
 36: STP_END   NOP
 37:
 38:
 39:
```

```
======== Kubes ================================== KUAX 680I =======
                        Program module IL
Project : PLC680I               Network   : PB_BSP_8
Module  : MO_AUT    No.: 3       created  : Mar 11 1996 09:20
User    : Paul Posi              edited   : Mar 11 1996 14:53
Comment : Automatic start of motor #2

=====================================================================
   1: ; ***************************************************
  2: ;         Initiate relative positioning step.
  3: ;         - automatic start by input I00.04
  4: ;         - COM_BYTE = %0101_0000
  5: ;         - stop by input I00.01
   6: ; ***************************************************
  7:
  8: ; —— Analyse motor status, destination position reached? -
  9:          L       M_DEST          M01.02 ; (motor at dest. pos. (BM06.09,
C8T1))
 10:          =       PP00_10        PP00.10 ; (programmable pulse)
 11:          L       PP00_10        PP00.10 ; (programmable pulse)
 12:          S       DEST_POS        M00.14 ; (motor destination position
reached)
 13:
 14: ; —— Automatic start ——
 15: AUTO_ST  L       HM_ASTAR        M00.05 ; (HM automatic start)
 16:          A       DEST_POS        M00.14 ; (motor destination position
reached)
 17:          JPC     AUTO_GO                ; if prog. pulse is not received
 18:
 19:          L       ST_AUTO         I00.04 ; (start automatic program)
 20:          =       PP00_04        PP00.04 ; (programmable pulse)
 21:          L       PP00_04        PP00.04 ; (programmable pulse)
 22:          JPCN    END                    ; if not start conditions exist
 23:          =1      HM_ASTAR        M00.05 ; (HM automatic start)
 24:
 25:
 26: ; —— Block transfer still active ?? ——
 27: AUTO_GO  L       W_MARKER        M00.01 ; (initiates block transfer,
write)
 28:          JPC     END
 29:
```

```
 30: ; — Set preset speed —
 31:          LD      2500                    ; rev/min
 32:          =D      VPOSB_L         BM04.04 ; (preset motor speed, backward
LB)
 33:          =D      VPOSF_L         BM04.06 ; (preset motor speed, forward LB)
 34:
 35: ; — Destination position (increments) —
 36:          LD      20000
 37:          =D      PPOS_LL         BM04.00 ; (preset position, LL)
 38:          LD      0
 39:          =D      PPOS_HL         BM04.02 ; (preset position, HL)
 40:
 41: ; — Set control bytes ——
 42:          L       %0000_0000
 43:          =       SYS_BYTE        BM04.14 ; (system control byte, bit analy-
sis)
 44:          L       %0101_0000
 45:          =       COM_BYTE        BM04.15 ; (command byte, bit analysis)
 46:
 47: ; — Initiate block transfer —
 48:          L       MOT_NO          BM09.00 ; (motor number)
 49:          =       STA_SEND        BM08.04 ; (station address PB_SEND)
 50:          LD      161                     ; command block in the PLC
 51:          =D      I_LOK_S         BM08.00 ; (block number of sender, LB)
 52:          LD      41                      ; command block in the motor
 53:          =D      I_REMO_S        BM08.02 ; (block number of receiver, LB)
 54:          =1      W_MARKER         M00.01 ; (initiates block transfer,
write)
 55: ; — Reset destination marker to zero —
 56:          =0      DEST_POS         M00.14 ; (motor destination position
reached)
 57:
 58: END      NOP
```

```
======= Kubes ================================== KUAX 680I =======
                        Program module IL
Project : PLC680I               Network   : PB_BSP_8
Module  : MO_START  No.: 6       created  : Mar 11 1996 09:20
User    : Paul Posi              edited   : Mar 11 1996 14:53
Comment : M2_START

=====================================================================
   1: ; ****************************************************
  2: ;         Initiate absolute positioning step.
  3: ;         - start by input I00.02
  4: ;         - COM_BYTE = %0001_0000
   5: ; ****************************************************
  6:           L       HM_START        M00.06 ; (HM start motor)
  7:           JPC     ST_GO                  ; if prog. pulse is not received
  8:
  9: ; — Start by input ——
 10:           L       MO_START        I00.02 ; (start motor)
 11:           =       PP00_02        PP00.02 ; (programmable pulse)
 12:           L       PP00_02        PP00.02 ; (programmable pulse)
 13:           JPCN    ST_END
 14:           =1      HM_START        M00.06 ; (HM start motor)
 15:
 16: ; — Block transfer still active ?? –
 17: ST_GO     L       W_MARKER        M00.01 ; (initiates block transfer,
write)
 18:           JPC     ST_END                 ; if other send task is active
 19:
 20: ; — Motor speed ———
 21:           LD      1500                  ; rev/min
 22:           =D      VPOSB_L        BM04.04 ; (preset motor speed, backward
LB)
 23:           =D      VPOSF_L        BM04.06 ; (preset motor speed, forward LB)
 24:
 25: ; — Destination position, increments ——
 26:           LD      40000
 27:           =D      PPOS_LL        BM04.00 ; (preset position, LL)
 28:           LD      0
 29:           =D      PPOS_HL        BM04.02 ; (preset position, HL)
 30:
```

```
 31: ; — Set system control byte ——
 32:          L       %0000_0000
 33:          =       SYS_BYTE       BM04.14 ; (system control byte, bit analy-
sis)
 34:
 35: ; — Initiate relative positioning step –
 36:          L       %0001_0000
 37:          =       COM_BYTE       BM04.15 ; (command byte, bit analysis)
 38:
 39: ; — Start command via the bus ——
 40:          L       MOT_NO         BM09.00 ; (motor number)
 41:          =       STA_SEND       BM08.04 ; (station address PB_SEND)
 42:          LD      161                    ; command block in the PLC
 43:          =D      I_LOK_S        BM08.00 ; (block number of sender, LB)
 44:          LD      41                     ; command block in the motor
 45:          =D      I_REMO_S       BM08.02 ; (block number of receiver, LB)
 46:          =1      W_MARKER        M00.01 ; (initiates block transfer,
write)
 47:          =0      HM_START        M00.06 ; (HM start motor)
 48: ST_END   NOP
 49:
```

```
======== Kubes ================================== KUAX 680I =======
                        Program module IL
Project : PLC680I                Network   : PB_BSP_8
Module  : STATUS    No.: 9        created  : Mar 11 1996 09:20
User    : Paul Posi               edited   : Mar 11 1996 14:53
Comment : Sort status messages

=====================================================================
   1: ; ******************************************************
  2: ;         Motor status messages (block 44)
   3: ; ******************************************************
  4:
  5: ; — Here request for motor #2 –
  6:           L       STA_REC         BM08.07 ; (station address PB_REC)
  7:          CMP     2
  8:           JP<>    MOT_3
  9:
 10: ; — Actual counter value (motor position) —
 11:           LD      CNT_LL          BM06.00 ; (position counter LL)
 12:           LD      CNT_HL          BM06.02 ; (position counter HL)
 13:
 14: ; — Motor status ——
 15:           L       STATUS_P        BM06.09 ; (status of positioning)
 16:           C8T1    M_READY          M01.00 ; (motor ready (BM06.09, C8T1))
 17:
 18: ; — Error and failure messages —
 19:           L       PAR_ERR         BM06.10 ; (parameter error)
 20:           L       POS_ERR         BM06.11 ; (positioning error)
 21:           L       MOT_FAIL        BM06.12 ; (motor failure)
 22:
 23: ; — Software identification —
 24:           L       SOFTW           BM07.13 ; (software identification)
 25:           L       BM06_14         BM07.14 ; (software identification)
 26:           L       BM06_15         BM07.15 ; (software identification)
 27:
 28: MOT_3     NOP
```

```
======== Kubes ================================== KUAX 680I =======
                    Initialization module IL
Project : PLC680I                Network   : PB_BSP_8
Module  : M1_PARA   No.: 1        created  : Mar 11 1996 09:20
User    : Paul Posi               edited   : Mar 11 1996 14:53
Comment : Motor parameters for motor #2

=============================================================================
   1: ; ****************************************************
  2: ;           Motor parameters for Dunker-made motor typeBG83
  3: ;           Motor #2 with incremental encoder, res. 500
   4: ; ****************************************************
  5:
  6: ; — Parameter block no. 43 —
  7: ;          Counter preset
  8: CNTSET_0    BM00.00      WORD     0         ; (counter preset value, LL)
  9: CNTSET_2    BM00.02      WORD     0         ; (counter preset value, HL)
 10:
 11: ;          Relative point of reference
 12: REL_RF_0    BM00.04      WORD     $0000     ; (relative point of reference,
LL)
 13: REL_RF_2    BM00.06      WORD     $0000     ; (relative point of refernce, HL)
 14:
 15: ;          Software limit switches
 16: LIM_L_0     BM00.08      WORD     $FFFF     ; (left limit switch, LL)
 17: LIM_L_2     BM00.10      WORD     $FF00     ; (left limit switch, HL)
 18:
 19: ;          Software limit switches
 20: LIM_R_0     BM00.12      WORD     $FFFF     ; (right limit switch, LL)
 21: LIM_R_2     BM00.14      WORD     $00FF     ; (right limit switch, HL)
 22:
 23: ;          Length of ramps
 24: ST_RPL      BM01.00      WORD     16000     ; (starting ramp left)
 25: STP_RPL     BM01.02      WORD     16000     ; (stopping ramp left)
 26: ST_RPR      BM01.04      WORD     16000     ; (starting ramp right)
 27: STP_RPR     BM01.06      WORD     16000     ; (stopping ramp right)
 28:
 29: ;          Starting and creeper speeds
 30: V_STRT_L    BM01.08      WORD     50        ; (starting speed left)
 31: V_STOP_L    BM01.10      WORD     50        ; (deceleration speed left)
 32: V_STRT_R    BM01.12      WORD     50        ; (starting speed right)
```

```
 33: V_STOP_R   BM01.14     WORD    50       ; (deceleration speed right)
 34:
 35: ;          Simple encoder resolution
 36: ENC_RES    BM02.00     WORD    500      ; (encoder resolution)
 37:
 38: ;          Amplification factor dividend speed regulator
 39: P_DIV      BM02.02     BYTE    35       ; (P part of speed regulator, fac-
tor)
 40:
 41: ;          Amplification factor divisor speed regulator
 42: P_DIVS     BM02.03     BYTE    10       ; (P part, divisor)
 43:
 44: ;          Scanning frequency speed regulator
 45: TA         BM02.04     WORD    2        ; (scanning frequency speed regu-
lator)
 46:
 47: ;          Adjustment time speed regulator
 48: TN         BM02.06     WORD    6        ; (adjustment time speed regula-
tor)
 49:
 50: ;          Suspension time speed regulator
 51: TV         BM02.08     WORD    1        ; (suspension time speed regula-
tor)
 52:
 53: ;         unused
 54: BM04_10    BM02.10     BYTE    0        ; (unused)
 55:
 56: ;          Amplification factor position regulator
 57: KPP        BM02.11     BYTE    25       ; (KP position regulator)
 58:
 59: ;         unused
 60: BM04_12    BM02.12     BYTE    0        ; (unused)
 61:
 62: ;          Adjustment time position regulator
 63: TNP        BM02.13     BYTE    20       ; (I part position regulator)
 64:
 65: ;          Position regulator range
 66: POS_RNG    BM02.14     WORD    0        ; (position regulation range)
 67:
 68: ;          Nominal current in mA
 69: I_NOM      BM03.00     WORD    7500     ; (nominal motor current, mA)
```

```
 70:
 71: ;          Startup current in mA
 72: I_GO       BM03.02      WORD     20000     ; (motor startup current, mA)
 73:
 74: ;          Nominal motor speed
 75: V_MAX      BM03.04      WORD     2750      ; (maximum motor speed, rpm)
 76:
 77: ;          Net-i timeout (polling channel)
 78: T_OUT      BM03.06      WORD     0         ; (timeout (Net_i only))
 79:
 80: ;          Unidirectional gear positioning in increments
 81: UGP        BM03.08      WORD     0         ; (unidirectional gear position.,
inc.)
 82:
 83: ;          unused
 84: BM05_10    BM03.10      BYTE     0         ; (unused)
 85:
 86: ;          Ramp form
 87: SET_RAMP   BM03.11      BYTE     1         ; (ramp form)
 88:
 89: ;          Motor type
 90: SET_TYP    BM03.12      BYTE     1         ; (motor type) 1 = BG 83
 91:
 92: ;          System settings
 93: SYS_INI    BM03.13      BYTE     %1000_0001 ; (system settings, bit selec-
tion)
 94:
 95: ;          Counter settings
 96: CNT_INI    BM03.14      BYTE     %1000_0001 ; (counter settings, bit selec-
tion)
 97:
 98: ;          Special function settings
 99: SPE_FCT    BM03.15      BYTE     %0000_1111 ; (special function, bit selec-
tion)
100:
101: ; — Preset destination marker ———
102: DEST_POS   M00.14       BIT      1         ; (motor destination position
reached)
103:
104:
105:
```

Index