KUHNKE

Kuhnke Electronics
Instruction Manual

Modules of Controllers
KUAX 680I, KUAX 680C, KDT 680CT

E 326 GB 7 May 1997 / 39.785

This manual is primarily intended for use by design, project, and development engineers. 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 and modifications.

Reproduction even of extracts only with the editor's express and written prior consent.

Table of contents

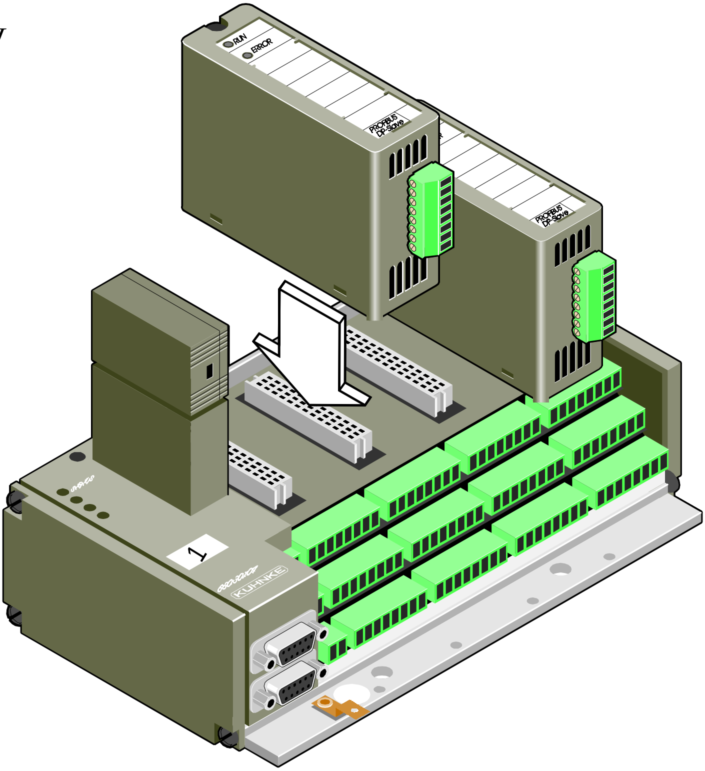
Appendix

Sales & Service

1. Introduction

KUAX 680I, KUAX 680C and KDT 680 CT are efficient, modularly constructed minicontrollers. They are equipped with modules which communicate with the user program in the controller either directly or via transfer addresses (SLx...). KUAX 680C and KDT 680CT also have built-in inputs and outputs which are not, however, described in this instruction manual. There are connectors for three-conductor connections so that proximity switches or other similar devices can be supplied via the same cable as the signal line. Additional terminal strips are thus made redundant. Simply plug the modules into the appropriate slots from above:

Fig.: KUAX 680I



1.1. Manual breakdown

- Table of contents

- Chapter 1
  Introduction

- Chapter 2
  Some comments on the reliability of the described products and their safe use

- Chapter 3
  Basic information about application and design of the modules as well as about slots and plug connectors

- Chapter 4
  Digital input and output modules
- Chapter 5
  Analogue input and output modules
- Chapter 6
  Counter modules and SSI module
- Chapter 7
  Communication modules V.24 (RS 232), TTY (20 mA) and RS 485, without processor
- Chapter 8
  Stepper motor modules without processor
- Chapter 9
  Stepper motor modules with processor

- Appendix A
  Power supply and load characteristics of the controllers
- Appendix B
  Module ordering information (summary)
- Appendix C
  References to general literature and to Kuhnke instruction manuals

- Index

☞ This manual only describes the modules. Please refer to the relevant instruction manuals to find out more about the actual controllers:

Controller	*Instruction manual*
KUAX 680I	E 380 GB
KUAX 680C	E 399 GB
KDT 680CT	E 414 GB

1.2. Further modules

Some modules are not described in this manual but, for subject matter reasons, have their own instruction manuals:

- PROFIBUS module 680.440.05
  Instruction manual: E 509 GB

- Positioning module 680.454.06
  Instruction manual: E 416 GB

- Servo counter module 680.454.05
  Instruction manual: E 416 GB

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 is written for 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,
statistic assessment 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 material/equipment.

2.4.1. To be observed during project planning and installation

☞

- 24V DC power supply:
  Generate as electrically safely separated low voltage. Suitable devices are, for example, split transformers constructed to correspond to European standard EN 60742 (corresponds to VDE 0551)
- 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 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 the planning and installation of electronic controllers 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 section 8 (Admissible deviations during working on parts) in particular, when measuring or checking a controller in a power-up condition.
- Repairs must only be made by specially trained Kuhnke staff (usually in the main factory in Malente). Warranty expires in every other case.
- Spare parts:
  Only use parts approved of by Kuhnke. Only genuine Kuhnke modules must be used 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 recognisably!) detracted from their proper functioning.
- Always deposit batteries and accumulators as hazardous waste.

2.5. Electromagnetic compatibility

2.5.1. Definition

Electromagnetic compatibility is the ability of a device to function satisfactorily in its electromagnetic environment without itself causing any electromagnetic interference that would be intolerable to other devices in this environment.

Of all known phenomena of electromagnetic noise, only a certain range occurs at the location of a given device. This noise depends on the exact location. It is determined in the relevant product standards.

The international standard regulating construction and degree of noise resistance of programmable logic controllers is IEC 1131-2 which, in Europe, has been the basis for European standard EN 61131-2.

2.5.2. Resistance to interference

Electrostatic discharge, ESD
in accordance with IEC 801-2, 3rd degree of sharpness

Fast transient interference, Burst
in accordance with IEC 801-4, 3rd degree of sharpness

Irradiation resistance of the device, HF
in accordance with IEC 801-3, 3rd degree of sharpness

Immunity to damped oscillations
in accordance with IEC 255-4 (1 MHz, 1 kV)

2.5.3. Interference emission

Interfering emission of electromagnetic fields, HF
in accordance with EN 55011, limiting value class A, group 1

☞ *If the controller is designed for use in residential districts, then high-frequency emissions must comply with limiting value class B as described in EN 55011.*
Fitting the controller into an earthed metal cabinet and equipping the supply cables with filters are appropriate means for keeping the corresponding limiting values.

2.5.4. General notes on installation

As component parts of machines, facilities and systems, electronic control systems must comply with valid rules and regulations, depending on the relevant field of application.

General requirements concerning the electrical equipment of machines and aiming at the safety of these machines are contained in Part 1 of European standard EN 60204 (corresponds to VDE 0113).

☞ *For safe installation of our control system please observe the following notes:*

2.5.5. Protection against external electrical influences

Connect the control system to the protective earth conductor to eliminate electromagnetic interference. Ensure practical wiring and laying of cables.

2.5.6. Cable routing and wiring

Separate laying of power supply circuits, never together with control current loops:

DC voltage	60 V ... 400 V
AC voltage	25 V ... 400 V

Joint laying of control current loops is allowed:

data signals, shielded
analogue signals, shielded

digital I/O lines, unshielded
DC voltages < 60 V, unshielded
AC voltages < 25 V, unshielded

2.5.7. Location of installation

Make sure that there are no impediments due to temperatures, dirt, impact, vibrations and electromagnetic interference.

Temperature
Consider heat sources such as general heating of rooms, sunlight, heat accumulation in assembly rooms or control cabinets.

Dirt
Use suitable casings to avoid possible negative influences due to humidity, corrosive gas, liquid or conducting dust.

Impact and vibration
Consider possible influences caused by motors, compressors, transfer routes, presses, ramming machines and vehicles.

Electromagnetic interference
Consider electromagnetic interference from various sources near the location of installation: motors, switching devices, switching thyristors, radio-controlled devices, welding equipment, arcing, switched-mode power supplies, converters / inverters.

2.5.8. Particular sources of interference

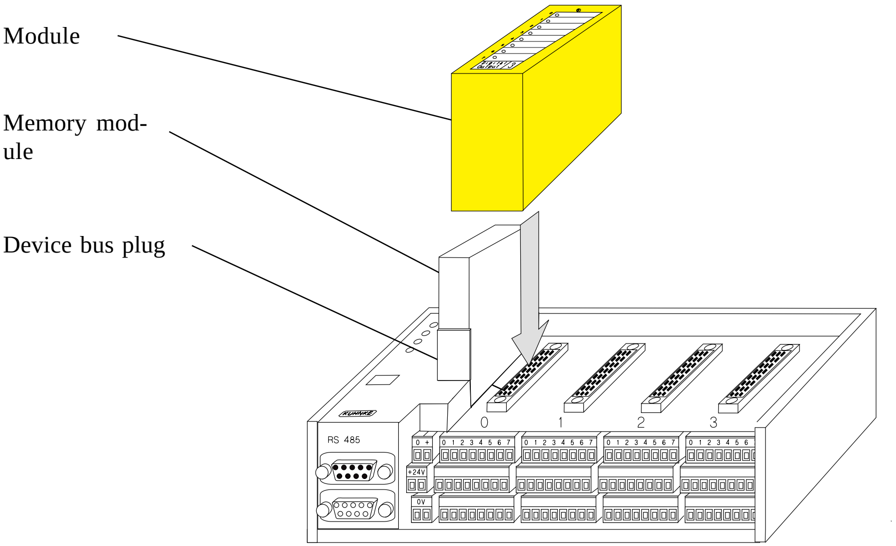
Inductive actuators
Switching off inductances (such as from relays, contactors, solenoids or switching magnets) produces overvoltages. It is necessary to reduce these extra voltages to a minimum.
Reducing elements my be diodes, Z-diodes, varistors or RC elements. To provide suitably designed reducing elements, we recommend asking the manufacturer or supplier of the corresponding actuators for the relevant information.

3. Modules and slots

KUAX 680I and 680C can be equipped with different types of modules. The number of modules depends on the size of the device:

- Devices with 4 slots: 4 modules (slots 0...3)
- Devices with 8 slots: 8 modules (slots 0...7) (KUAX 680I only)

Modules are plugged into the top of the device. They are connected to the controller via the device bus plug (the figure below shows KUAX 680I):



The following has to be observed when placing the modules:

⚠

- *Only plug modules in/out in an idle state. Danger of destruction!*
- *Sequence modules from left to right (see also "3.3. Slots").*
- *The modules must be screwed into the device. They could otherwise become loose and then cause dangerous conditions.*

3.1. Design

The modules are enclosed in a plastic casing. For cooling the electronics, there are ventilation slots on the narrow edges. The plug for the connection to the bus connector is on the bottom side.

Mounting

There is a screw sunk into the front side which is used to attach the module to the device frame.

⚠ *In KUAX 680C, KDT 680CT, and recent models of KUAX 680I (part numbers 680.423.xx) the modules are additionally supported by plastic rests. Modules made before calendar week 27/95 do not fit into these devices because they have no bores to place the plastic rests in.*

Labels

A large area on the front side is saved for a label. The label can be used to enter the symbolic designations of the input or output signals (cf. symbol table under KUBES) or the function of the module.
There is a sufficient number of labels. They are delivered together with the KUAX 680I as perforated A4 tear-off blanks. The line and column spacings are set in a way that makes an inscription possible also for dot-matrix printers with a condensed print.

Status indicators

On the lefthand side of the labels there are left as many holes as the module provides status indicators.
These holes fit across LEDs which are built into the actual module casing to indicate, for example, signal states of inputs or outputs.

☞ *By definition, light emitting diodes (also referred to as LEDs) are "Class 1 light emitting diodes (in acc. with EN 60825_1)".*

Channel numbers

On input and output modules, the corresponding channel numbers (0...3/7) are printed left to the LEDs. They correspond to the inscription on the signal strip underneath the module. The relation between terminal, LED and inscription is thus documented.

3.2. Screw-type locking connectors

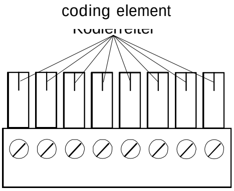
Screw-type locking connectors (supplied by Phoenix) are used to connect inputs and outputs to the device:
Connector type MINI-COMBICON, 3.81 mm matrix, connecting diameter 0.14...1,5 mm² , max. load 8 A

The green screw-type locking connectors sit very firmly in their position to avoid them becoming loose due to vibrations. Should you find it difficult to pull them off with your fingers, simply use a flat object such as a screwdriver with a wide blade as a lever.

Never pull the wires to disconnect the plug. The wires might otherwise slip out of the terminals or rip off even.

3.2.1. Coding

You can code the MINI-COMBICON connectors so that you do not get them confused when putting them on (by conntecting digital inputs to an SSI module, for example).
Push one or several coding profiles into the groove(s) provided on the socket part of the connector for this purpose. Use a side cutter, for example, to cut the corresponding coding element off the plug part.



There are some connectors in the basic device which are coded in the factory. Please refer to the corresponding illustrations to learn where such codings exist and what they look like.

3.3. Slots

KUAX 680I can be equipped with slots for either 4 or 8 modules. KUAX 680C and KDT 680CT always have 4 slots (or none).

3.3.1. Function slots

Individual slots carry additional leads. They make the application of certain modules possible for using services provided by the processor (counting, stepper motor control, analogue/digital conversion...):

Device with 8 slots (KUAX 680I only)							
Device with 4 slots							
0	1	2	3	4	5	6	7
	event counter x 2 *3) PWM x 2 *1)	analogue input x 2 *2) PWM x 2 *1)	analogue input x 4 *2)	analogue input x 4 *2)			

☞ *These specifications do not limit the application of any other module. If a function slot is not needed as such, it can also be used for a slot-independent module such as a digital input or output module.*

~~*1) PWM stands~~ for "pulse-width modulated output" for stepper motor control.

*2) This only concerns those analogue inputs with a resolution of 10bit that use the A/D converter on the CPU.

3.4. Service modules

Modules with more complex functions need software support to be able to carry out these functions. In order to avoid the necessity for the user to write these often very complicated programs himself, so-called service modules are embedded in the program to relieve the user.
As from release 4.00, these service modules are delivered with KUBES as individual files under the name SERV_xx.BIN. The KUBES installation program places these files in the same subdirectory as the program files (e.g. C:\KUBESEXE).

Configuration

When configuring KUAX 680I or 680C, you enter the modules into a list in the same order in which they will be plugged in later. KUBES can use this information to embed the necessary service modules in the user program and to create the reference to the modules.

3.4.1. Transfer address ranges

Service modules use so-called transfer address ranges (max. 32 byte) for each module for data exchange with the user program. These are directly assigned to the module slots (see table on next page). The service module assigned to the slot during configuration determines the assignment of the transfer addresses.

3.4.2. Interrupt modules

Under certain conditions, some modules trigger an interrupt in the CPU. Like this the service module, and sometimes also the user program, can react particularly quickly to an event. An interrupt caused by a module calls up an interrupt module (see table on next page). This module can contain user-defined instructions to be carried out in case of an interrupt.

Assignment of transfer addresses and interrupt modules

Slot	Transfer address range	Interrupt module
0	SLA00.00...01.15 SLB00.00...01.15	1 2
1	SLC00.00...01.15 SLD00.00...01.15	3 4
2	SLE00.00...01.15 SLF00.00...01.15	5 6
3	SLG00.00...01.15 SLH00.00...01.15	7 8
4	SLI00.00...01.15 SLJ00.00...01.15	9 10
5	SLK00.00...01.15 SLL00.00...01.15	11 12
6	SLM00.00...01.15 SLN00.00...01.15	13 14
7	SLO00.00...01.15 SLP00.00...01.15	15 16

3.5. Power supply of inputs and outputs

All inputs and outputs as well as the relay for polarity safeguarding in the digital output modules are centrally supplied via the corresponding terminals of the basic device (see the relevant instruction manuals of the individual controllers).

3.6. Differences between KUAX 680C/KDT 680CT and KUAX 680I

All examples given in this instruction manual refer to the use of the relevant modules in KUAX 680I.
When working with KUAX 680C or KDT 680CT you will be obliged to take note of some differences which are caused by the I/Os available to the basic device.

Limited applicability

- You cannot use the counter module for event counters, order no. 680.454.03; the reason being that in its standard configuration, the device is already equipped with two event counters (internal inputs).
- Stepper motor modules and internal analogue outputs share the same system resources, i.e. the PWM outputs of the processor:

PWM	analogue output	stepper motor module
1	AO00.00	680.444.01 and .02
2	AO00.01	680.444.02

Thus, if you are using a two-channel stepper motor module (680.440.02) neither of the two internal analogue outputs can be used. If you are working with a single-channel stepper motor module (680.440.01), you still have the option of using internal analogue output AO00.01 at least.

Addressing

Please note that input and output groups are occupied already by the internal I/Os. While plugged-in modules are numbered in groups from left to right just like in KUAX 680I, they start with different group numbers:

Modules	first group
digital inputs	I02
analogue inputs	AI01
digital outputs	O02
analogue outputs	AO01

☞ *For further information please refer to the device manuals.*

4. Digital inputs and outputs

Digital inputs and outputs are used for leading digital signals (1 or 0, High or Low, On or Off etc.) from the machine or system into the controller (inputs) or vice versa (outputs).

These include:

Digital input signals from
- switches
- key-switches
- sensors
- etc.

Digital output signals for switching
- relays
- contactors
- magnetic valves
- etc.

There are only 2 possible states of digital signals:
- logical 0 (Low, Off...)
- logical 1 (High, On...)

These two signal states form the basis for most of the functions of control engineering.

There are several digital input and output modules available for adjusting the controller to the different types of signals.

4.1. Plugging digital input and output modules

As from monitor version 4.10 (KUBES version 4.01), KUAX 680I can also be equipped with 16pin modules. KUAX 680C as from its first delivered machines.

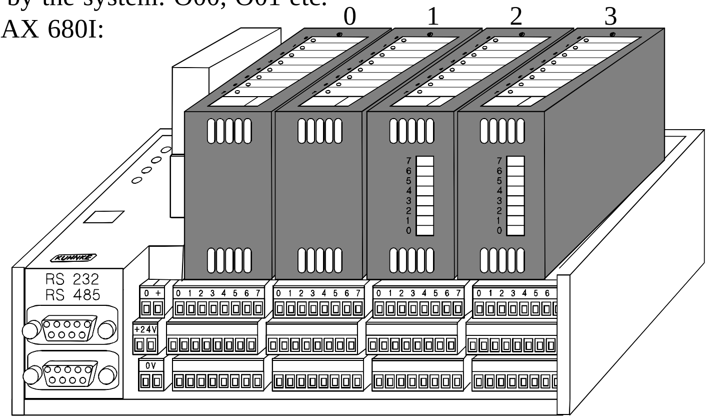
All earlier versions of KUAX 680I only allow application of 8pin modules.

Digital input and output modules can be plugged into any slot as long as this slot is not needed for another module (see "3.3.1. Function slots"). Certain regulations must be regarded, however:

4.1.1. Addressing (input/output groups and channels)

Upon switching the controller on, inputs and outputs of all plugged-in modules are automatically numbered from left to right in groups of 8 channels (.00... .07) each. Thus, the first input group is assigned the group number I00, the second one is assigned I01 etc. The output groups are treated in the same way by the system: O00, O01 etc.

Example for KUAX 680I:



Legend:

0	input module, 8pin	I00
1	output module, 8pin	O00
2	input module, 16pin	I01, I02
3	input/output module, 8/8pin	I03, O01

Groups

Inputs:	I00...max. I15 (on 16pin modules)
Ouputs:	O00...max. O07

⚠ *In KUAX 680C, the first two input groups, I00 and I01, and the first two output groups, O00 and O01, are allocated already to the internal inputs and outputs. The modules for inputs and outputs thus start with numbers I02 and O02.*

Channels

Each group consists of 8 channels, i.e. inputs or outputs:

Inputs:	Ixx.00...Ixx.07
Outputs:	Oxx.00...Oxx.07

4.1.2. Reserved slots

Reserved slots for a module type of which there are modules plugged in already (e.g. input modules) must always be located after the last plugged-in module of the same type. Otherwise, the module numbering (addressing) of the following modules of the same type will change.

```
Example:   I   I   _   O
                  |
                  Reserved slot for an input module (I02.)
```

Reserved slots for later plugging of function modules or analogue input modules may also be located between modules of one type.

```
Example:    I   _    I   O
            |   |    |   |
            |   |    |   O00.
            |   |    I01.
            |   Reserved slot for a counter module
           I00.
```

4.2. Digital input modules

Digital input modules are used for reading the signal states of switches, key switches, relay contacts etc. When working with proximity switches and semi-conductor sensors please make sure to keep them running within the switching thresholds given below.

Switching thresholds and filters

The input line is used to adjust the connected signals to the system voltage.

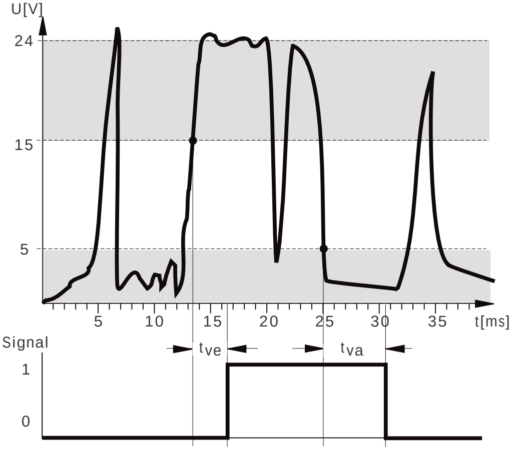
Defined signals:

Logical 0:	≤ 5 V
Logical 1:	≥ 15 V
(Hysteresis:	1...4 V)

Signal delay of inputs with normal (5ms) delay time:

In order to avoid voltage surges (noise pulses) being recognized as valid signals, thus executing wrong switching operations, they are filtered out.

Signal identification is thus delayed by a nominal period of 5 ms:



<u>Raising delay *)</u>:

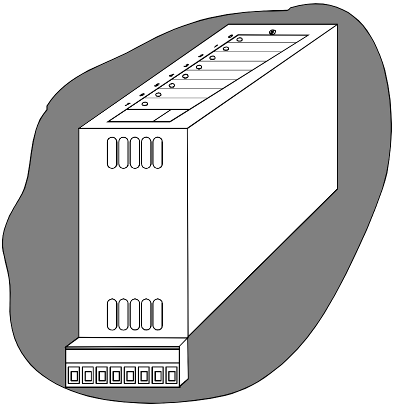
t_{ve} = 3.0 ... 7.0 ms

<u>Falling delay *)</u>:

t_{va} = 4.0 ... 7.0 ms

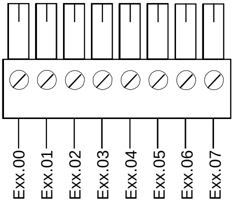
*) The input signals are read between individual program cycles and then written into the process image. To correctly determine the availability of the signals for the user program you would have to add the program cycle time to the delay time.

4.2.1. Input module, 24 V DC, 8 inputs



Signal line connection

Input voltage is 24 V DC. These signals are applied via plug-screw terminals on the signal strip underneath the module.



*)

*) read 'E' = 'I' (for Input)

4.2.1.1. Technical specifications

Application:	KUAX 680I, 680C, 680S, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs:	8
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	LEDs
colour	green
tapping point	in the input circuit
signal states	1: LED on
	0: LED off
Addressing:	Ixx.00...Ixx.07
Input voltage:	24 V DC -20%/+25% (incl. residual ripple)
Surge immunity	≤ 60 V DC (≤ 30 min.)
Signal recognition	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Power consumption / input:	max. 10 mA
Weight:	c. 77 g
Part number:	680.451.01

4.2.2. Input module, 24 V DC, 8 inputs, with real-time clock

These are the same inputs as described in chapter “4.2.1. Input module, 24 V DC, 8 inputs” (see there for picture and connections).
Added is the function of the real-time clock which is realised on this module.

4.2.2.1. Real-time clock

A clock module is integrated in the module which is supplied by an accu in case of a voltage cut.
A KUBES module is available for setting and evaluating time and date.

Name: CLOCK

Call from within the program:

```
JPK   CLOCK
      <actual time>             <status indication>
      <set>
```

The module should be called up in every PLC cycle. The program within the module is organized so as to updating the input parameter “actual time” every second.
If there is a set command (first byte in data field “set” = 255), the preset values are taken over immediately.
Output byte “status indication” reports any occurring errors.

4.2.2.2. Parameters of KUBES module “CLOCK”

The user has to reserve two data fields for the clock parameters and one byte for status indication. It is helpful to enter these operands into the symbol table (KUBES) at the beginning of project planning.

⚠ *The data fields must consist of subsequent operand ranges. When programming the KUBES module, the first address of each data field is entered.*

We recommend to take over the symbols as shown in the 3 following tables:

Input parameter “Actual time”

Data field, 21 byte: current time and date (read only):

Byte			Significance
No	Example	Symbol	
Actual value (decimal) for evaluation in the program:			
1	SBM00.00	SECOND	second 0...59
2	SBM00.01	MINUTE	minute 0...59
3	SBM00.02	HOUR	hour 0...23
4	SBM00.03	WEEKDAY	day of the week 0 (Sunday) ... 6 (Saturday)
5	SBM00.04	DAY	day 1...31
6	SBM00.05	MONTH	month 1...12
7	SBM00.06	YEAR	year 0...99
Actual value (ASCII) for transfer to text displays:			
8	SBM00.07	ASC_SEC	second lowbyte
9	SBM00.08		second highbyte
10	SBM00.09	ASC_MIN	minute lowbyte
11	SBM00.10		minute highbyte
12	SBM00.11	ASC_HOUR	hour lowbyte
13	SBM00.12		hour highbyte
14	SBM00.13	ASC_WDAY	weekday lowbyte
15	SBM00.14		weekday highbyte
16	SBM00.15	ASC_DAY	day lowbyte
17	SBM01.00		day highbyte
18	SBM01.01	ASC_MON	month lowbyte
19	SBM01.02		month highbyte
20	SBM01.03	ASC_YEAR	year lowbyte
21	SBM01.04		year highbyte

Input parameter “Set”

Data field, 7 byte: clock flag, slot, preset time and preset date:

Byte			Significance
No	Example	Symbol	
1	SBM01.05	CL_FLAG	clock flag (255: write values in bytes 3...7 into the clock)
2	SBM01.06	CL SLOT	slot number of the module (0...7)
3	SBM01.07	SET SEC	preset value second: 0...59
4	SBM01.08	SET MIN	preset value minute: 0...59
5	SBM01.09	SET HOUR	preset value hour: 0...23
6	SBM01.10	SET_WDAY	preset value weekday: 0 (Sunday) ... 6 (Saturday)
7	SBM01.11	SET_DAY	preset value day: 1...31
8	SBM01.12	SET_MON	preset value month: 1...12
9	SBM01.13	SET YEAR	preset value year: 0...99

Output parameter “Status indication”

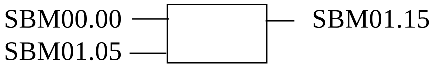
1 byte: error message or OK, resp.

Byte		Value	Significace
Example	Symbol		
SBM01.15	STATUS	0	okay message, no error
		1	undefined clock state (no load on the accu, identification cleared). Remedy: set the clock again and recharge accu (approx. 72 h on the net)
		2	set values outside the valid range
		3	input parameters: the range of a data field is exceeded
		4	illegal slot number
		5	-
		6	-
		7	-
		8	-

4.2.2.3. Programming suggestion for the real-time clock

```
          L       n           ;module slot (0...7)
          =       SBM01.06

          JPK     CLOCK
```

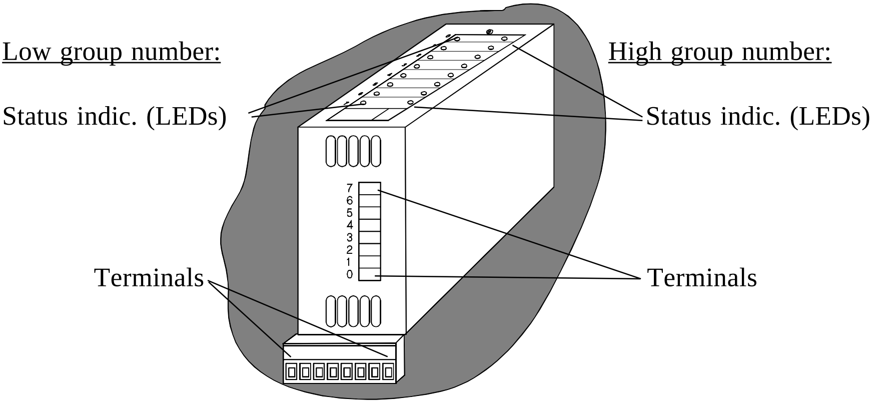


```
          L       SBM01.15
          CMP     0
          JP<>    ERROR       ;for error evaluation
DISPLAY.
          .                   ;display actual value
          .
SET       L       Mxx.xx
          =       PPxx.xx
          L       PPxx.xx
          JPCN    ENDCLK      ;to end of clock program
          L       255
          =       SBM01.05
PREVAL .
          .                   ;write preset value into oper.
          .                   ;SET_SEC...SET_YEAR
          JP  ENDCLK
ERROR   .
          .                   ;error evaluation
          .
ENDCLK.
          .                   ;normal program
```

4.2.2.4. Technical specifications

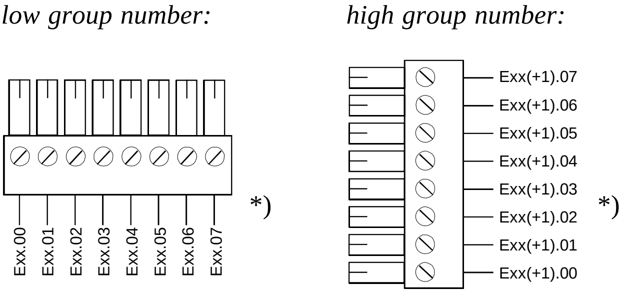
Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs:	8
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	LEDs
colour	green
tapping point	in the input circuit
signal states	1: LED on 0: LED off
Addressing:	Ixx.00...Ixx.07
Input voltage:	24 V DC -20%/+25% (incl. residual ripple)
Surge immunity	≤ 60 V DC (≤ 30 min.)
Signal recognition	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Power consumption / input:	max. 10 mA
Real-time clock:	second, minute, hour weekday, day, month, year
Accumulator (buffering of clock)	
buffer time:	~ 3 months
charging time:	~ 72 h
Weight:	c. 85 g
Part number:	680.451.02

4.2.3. Input module, 24 V DC, 16 inputs



Signal line connection

The input voltage is 24 V DC. These signals are applied via clamp-screw terminals. The terminals are located on the signal strip underneath the module for the low group and on the module itself for the high group (see diagram).



*) read 'E' = 'I' (for Input)

4.2.3. Technical specifications

Application:	KUAX 680I (as from version 4.10), KUAX 680S (part no. 680.301.04/08) KUAX 680C, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs:	16
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	LEDs
colour	green
tapping point	in the input circuit
signal states	1: LED on 0: LED off
Addressing:	Ixx.00...Ixx.07 Ixx(+1).00...Ixx(+1).07
Input voltage:	24 V DC -20%/+25% (incl. residual ripple)
Surge immunity	≤ 60 V DC (≤ 30 min.)
Signal recognition	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Power consumption / input:	max. 10 mA
Weight:	c. 104 g
Part number:	680.451.03

4.2.4. Input module, 24 V DC, 8 inputs, 1 ms

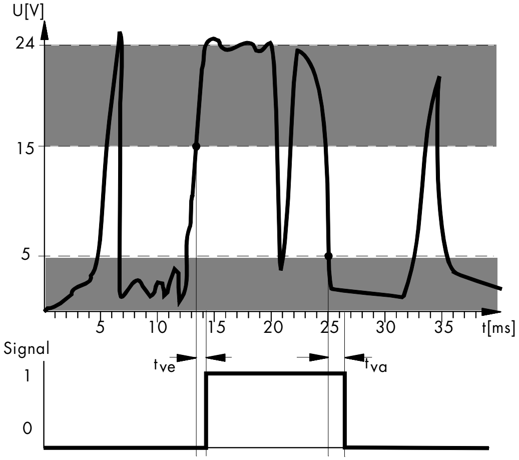
These are basically the same inputs as described earlier in chapter "4.2.1. Input module, 24 V DC, 8 inputs" (see there for diagram and connectors).
The difference between the two modules is in the much reduced signal delay time of the module described in this chapter. This shorter signal delay makes it possible to register very much increased signal sequences (e.g. fast counting impulses).

⚠ *Please take into consideration that the shorter signal delay time may lead to situations where noise pulses are not filtered out and are therefore not registered by the CPU. We therefore strongly recommend the use of shielded wires.*

Signal delay:

In order to avoid voltage surges (noise pulses) being recognized as valid signals, thus executing wrong switching operations, they are filtered out.

Signal identification is thus delayed by nominally 1 ms:



Raising delay*):

t_{ve} = 0.3 ... 1.0ms

Falling delay*):

t_{va} = 0.4 ... 1.4ms

*) The input signals are read between the individual program cycles and then written into the process image. To correctly determine the availability of the signals for the user program you would have to add the program cycle time to the delay time.

4.2.4.1. Technical specifications

Application:	KUAX 680I, 680C, 680S, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs:	8
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	LEDs
colour	green
tapping point	in the input circuit
signal states	1: LED on
	0: LED off
Addressing:	Ixx.00...Ixx.07
Input voltage:	24 V DC -20%/+25% (incl. residual ripple)
Surge immunity	≤ 60 V DC (≤ 30 min.)
Signal recognition	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Max. voltage:	28.8 V DC
Power consumption / input:	c. 10 mA
Weight:	c. 77 g
Part number:	680.451.04

4.2.5. Input module, 24 V DC, 16 inputs, 1 ms

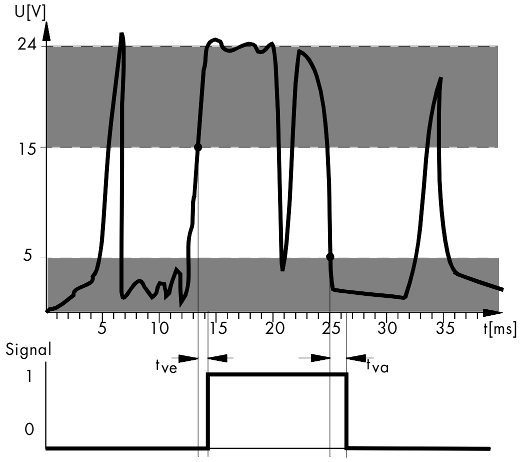
These are basically the same inputs as described earlier in chapter "4.2.3. Input module, 24 V DC, 16 inputs" (see there for diagram and connectors).
The difference between the two modules is in the much reduced signal delay time of the module described in this chapter. This shorter signal delay makes it possible to register very much increased signal sequences (e.g. fast counting impulses).

⚠ *Please take into consideration that the shorter signal delay time may lead to situations where noise pulses are not filtered out and are therefore not registered by the CPU. We therefore strongly recommend the use of shielded wires.*

Signal delay:

In order to avoid voltage surges (noise pulses) being recognized as valid signals, thus executing wrong switching operations, they are filtered out.

Signal identification is thus delayed by nominally 1 ms:



Raising delay*):

t_{ve} = 0.3 ... 1.0ms

Falling delay*):

t_{va} = 0.4 ... 1.4ms

*) The input signals are read between the individual program cycles and then written into the process image. To correctly determine the availability of the signals for the user program you would have to add the program cycle time to the delay time.

4.2.5.1. Technical specifications

Application:	KUAX 680I (as from version 4.10), KUAX 680S (part no. 680.301.04/08) KUAX 680C, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs:	16
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	LEDs
colour	green
tapping point	in the input circuit
signal states	1: LED on 0: LED off
Addressing:	Ixx.00...Ixx.07 Ixx(+1).00...Ixx(+1).07
Input voltage:	24 V DC -20%/+25% (incl. residual ripple)
Surge immunity	≤ 60 V DC (≤ 30 min.)
Signal recognition	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Power consumption / input:	max. 10 mA
Weight:	c. 104 g
Part number:	680.451.07

4.2.6. Input module, 24 V DC, 16 inputs, interrupt capability

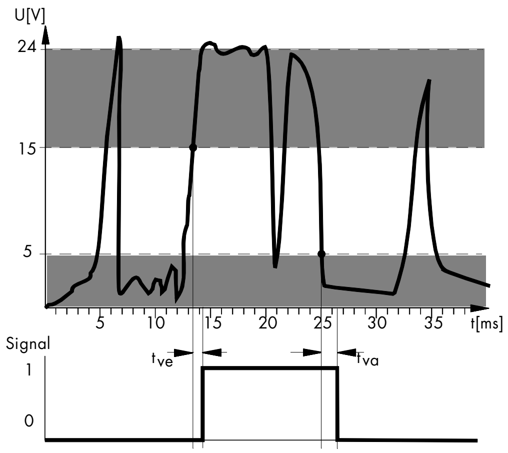
These are basically the same inputs as described earlier in chapter "4.2.3. Input module, 24 V DC, 16 inputs" (see there for diagram and connectors).
The difference between the two modules is in the extended functionality of the module described in this chapter concerning the registeration of fast input signals.

⚠ *Please take into consideration that the shorter signal delay time may lead to situations where noise pulses are not filtered out and are therefore not registered by the CPU. We therefore strongly recommend the use of shielded wires.*

4.2.6.1. Signal delay

In order to avoid voltage surges (noise pulses) being recognized as valid signals, thus executing wrong switching operations, they are filtered out. Signal identification is thus delayed:

Inputs Ixx.00...03 (lower group, terminals underneath the module)



The first 4 outputs have interrupt capability and have a particularly short signal delay time of nominally 0.3 ms.

Raising delay*):

t_{ve} = 0.09 ... 0.36ms

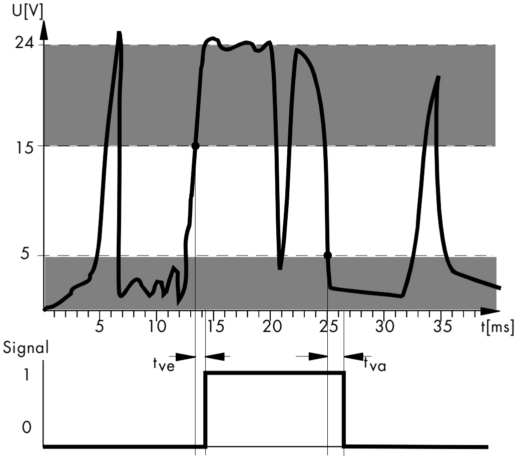
Falling delay*):

t_{va} = 0.12 ... 0.39ms

*) The input signals are read between the individual program cycles and then written into the process image. To correctly determine the availability of the signals for the user program you would have to add the program cycle time to the delay time.

Inputs Ixx.04...07 (lower group, terminal underneath the module)

The next 4 outputs have a short signal delay time of nominally 1ms.



Raising delay*):

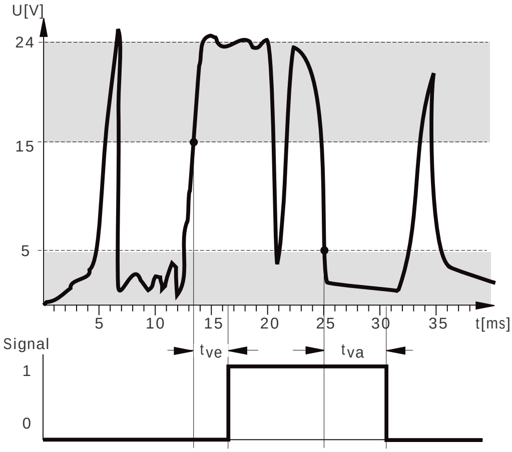
t_{ve} = 0.3 ... 1.0ms

Falling delay*):

t_{va} = 0.4 ... 1.4ms

Inputs Ixx(+1).04...07 (higher group, terminal on the module)

These 8 outputs have the normal signal delay time of nominally 5 ms.



Raising delay*):

t_{ve} = 3.0 ... 7.0ms

Falling delay*):

t_{va} = 4.0 ... 7.0ms

*) The input signals are read between the individual program cycles and then written into the process image. To correctly determine the availability of the signals for the user program you would have to add the program cycle time to the delay time.

4.2.6.2. Interrupt inputs

Inputs Ixx.00...03 (lower group, terminal underneath the module) support interrupt functions and have a particularly short signal delay time of nominally 0.3 ms.
For enabling and registering interrupts, each module is automatically assigned a transfer address range (see "3.4. Service modules") and an interrupt module for interrupt evaluation.

Transfer address ranges and interrupt modules

Slot	Transfer address range	Interrupt module
0	SLA00.00...01.15	1
1	SLC00.00...01.15	3
2	SLE00.00...01.15	5
3	SLG00.00...01.15	7
4	SLI00.00...01.15	9
5	SLK00.00...01.15	11
6	SLM00.00...01.15	13
7	SLO00.00...01.15	15

When enabled, each of the 4 inputs can trigger an interrupt in the CPU by raising or falling signal edges. In this case, the CPU immediately interrupts the normal program run and calls up the interrupt module that is assigned to this module slot. In this interrupt module you can get the information about the signal that triggered the interrupt and also start the necessary measures. We recommend keeping the program in the interrupt module as short as possible to avoid putting great loads on the cycle time, especially if interrupts are used frequently.

☞ *The user program can also scan the status of the interrupt inputs via the normal input addresses, Ixx.00...03. This means working with the signal delay that is extended by the cycle time, however.*

The functions of the transfer addresses are listed in the table on the next page:

Functions of the transfer addresses:

Address	Symbol	Function		
SLx00.00	INT_LH_0	interrupt triggered by: positive edge (Low->High)	input 00	CPU writes 255 after interrupt triggered via this channel; user program analyses information in interrupt module
SLx00.01	INT_LH_1		input 01	
SLx00.02	INT_LH_2		input 02	
SLx00.03	INT_LH_3		input 03	
SLx00.04	INT_HL_0	interrupt triggered by: negative edge (High->Low)	input 00	
SLx00.05	INT_HL_1		input 01	
SLx00.06	INT_HL_2		input 02	
SLx00.07	INT_HL_3		input 03	
SLx01.00	ENI_LH_0	enable interrupt trigger: positive edge (Low->High)	input 00	User program writes '255' to enable interrupt source, or '0' to disable interrupt source. Settings are transferred to CPU by 255 in SLx01.14
SLx01.01	ENI_LH_1		input 01	
SLx01.02	ENI_LH_2		input 02	
SLx01.03	ENI_LH_3		input 03	
SLx01.04	ENI_HL_0	enable interrupt trigger: negative edge (High->Low)	input 00	
SLx01.05	ENI_HL_1		input 01	
SLx01.06	ENI_HL_2		input 02	
SLx01.07	ENI_HL_3		input 03	
SLx01.08	ENI_MOD	User program writes 255 to enable input module for processing interrupt (0 for disabling)		
SLx01.14	SET_ENI	User program writes 255 to transfer new settings SLx01.00...14 to CPU; CPU acknowledges by writing 0		
SLx01.15	INT_VERS	Software date (version) of service module		

Example: Enabling the first input for interrupt H->L and L->H:

```
                L       M00.00     ;initialisation marker
                JPC     OK
                L       255
                =       ENI_MOD    ;general interrupt enable
                =       ENI_LH_0   ;enable interrupt Low->High
                =       ENI_HL_0   ;enable interrupt High->Low
                =       SET_ENI    ;transfer new settings
        OK      =1      M00.00     ;initialisation marker
```

Example: interrupt analysis

```
        LH_0    L       INT_LH_0   ;interrupt trigger Low->High?
                JPCN    HL_0
                ...         ;analysis program interrupt Low->High
                CLR     INT_LH_0   ;reset interrupt trigger
        HL_0    L       INT_HL_0   ;interrupt trigger High->Low?
                JPCN    CONTINUE
                ...         ;analysis program interrupt High->Low
                CLR     INT_HL_0   ;reset interrupt trigger
        CONTINUE ...
```

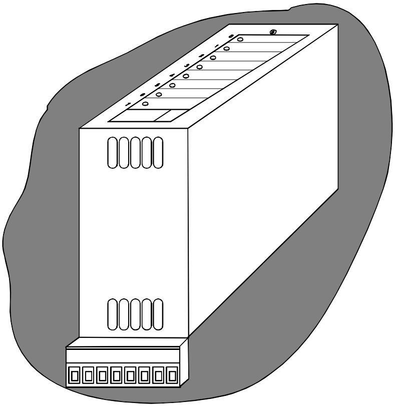
Interrupt frequency

Every interrupt lengthens the cycle time. If an overall program cycle is longer than 70 ms, the monitor will trigger a watchdog and interrupt the program run. Thus the interrupt frequency must not exceed a maximum of 2.5 kHz for short periods of time (approx. 10 ms). Taken over longer periods of time, the maximum interrupt frequency must be no higher than 1 kHz. If you are using the interrupt module as a counter you must make sure not to exceed the interrupt frequency of 1 kHz.

4.2.6.3. Technical specifications

Application:	KUAX 680I (as from version 4.17), KUAX 680C, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs:	16
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	LEDs
colour	green
tapping point	in the input circuit
signal states	1: LED on 0: LED off
Addressing:	Ixx.00...Ixx.07 Ixx(+1).00...Ixx(+1).07
Ixx.00...03	filter 0.3 ms with interr. cap.,≤2.5 kHz
Ixx.04...07	filter 1 ms
Ixx(+1).00...07	filter 5 ms
Input voltage:	24 V DC -20%/+25% (incl. residual ripple)
Surge immunity	≤ 60 V DC (≤ 30 min.)
Signal recognition	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Power consumption / input:	max. 10 mA
Weight:	c. 108 g
Part number:	680.451.06

4.3. Digital output module, 24 V DC, 0.5 A, 8 outputs

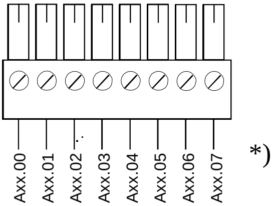


Function

Output modules provide the connection to external actuators (relays, contactors, magnets, valves...).
Resistive and inductive loads can be connected. Freewheeling diodes have been added to suppress inductive disable surges.
The switching state of the outputs is indicated by LEDs.

Signal line connection

The controller generates output signals of a voltage of 24 V DC. These are picked up at the plug-screw terminals on the signal strip. Each module is assigned one block of 8 of these terminals. The block is placed directly underneath the module slot (see diagram above).



*) read 'A' = 'O' (for Output)

Polarity safeguard

If the connection is correct, a relay in the module switches the supply of the outputs. A destruction of the module in case of reversed polarity is thus avoided.

Increased performance by parallel connection

The load-carrying ability of the individual outputs is limited (see Technical specifications). It is permissible, however, to connect 2 outputs in parallel, thus redoubling the performance.

⚠ *You must only connect outputs on the same module in parallel.*

Protection against short circuit and overload

The outputs are protected against destruction by overload or short circuit:
- the load current is limited to approx. 1.0...1.2 A
- a temperature monitor switches the output off after 0.1 to1 s and reports a short circuit to the CPU

in KUAX 680I and 680C:
- the CPU outputs a short circuit message,
- reports short circuit by flash pulse (1) on "failure" LED,
- activates interrupt module no. 18,
- see also instruction manual KUAX 680I, E 308 GB, Appendix D.1.

in KUAX 680S:
- the CPU disables all outputs,
- reports a short circuit by flash pulse (1) on "status" LED,
- sends event notification (1) to the master via PROFIBUS

after removing the error:
- reset device (by switching it off and on)

Restart

- Search source of error
- Put controller into idle condition
- Remove error
- Supply controller with voltage

Backfeeding of outputs

Backfeeding means that an output is externally supplied with voltage. This is no problem normally.
However, under certain circumstances it can destroy the free-wheeling diode of the relevant output:
if

- the controller is on
- and outputs are switched on and under a load
- and if the module supply (upper terminal) is off

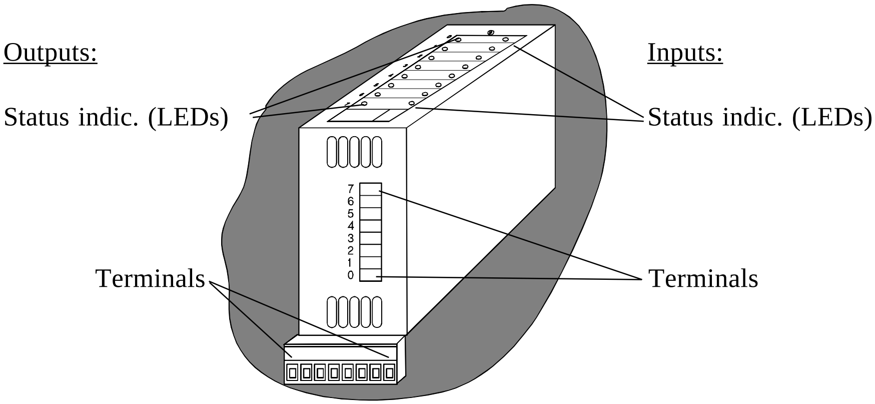
then

- the backfed output will use the built-in free-wheeling diode for supply
- and the load the current may destroy the free-wheeling diode.

4.3.1 Technical specifications

Application:	KUAX 680I, 680C, 680S, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Outputs:	8
type	semiconductor
Indicators	LEDs
colour	red
tapping point	in the load current circuit
signal states	1: LED on 0: LED off
Addressing:	Oxx.00...Oxx.07
Output voltage:	24 V DC -20%/+25% (incl. residual ripple)
Output current:	max. 0.5 A
Short circuit protection:	yes
Weight:	c. 84 g
Part number:	680.452.02 (older version 680.452.01)

4.4. Digital input/output module, 24 V DC, 8/8



Function

Inputs

The inputs are the same as described in chapter "4.2.1. Input module, 24 V DC, 8 inputs".

Outputs

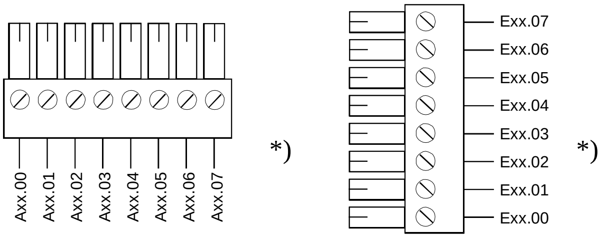
The outputs are the same as described in chapter "4.3. Digital output module, 24 V DC, 0.5 A, 8 outputs".

Signal line connection

The input signals are connected to the module via clamp-screw terminals.

The outputs signals are connected via the signal strip underneath the module (see diagram above).

Outputs: *Inputs:*

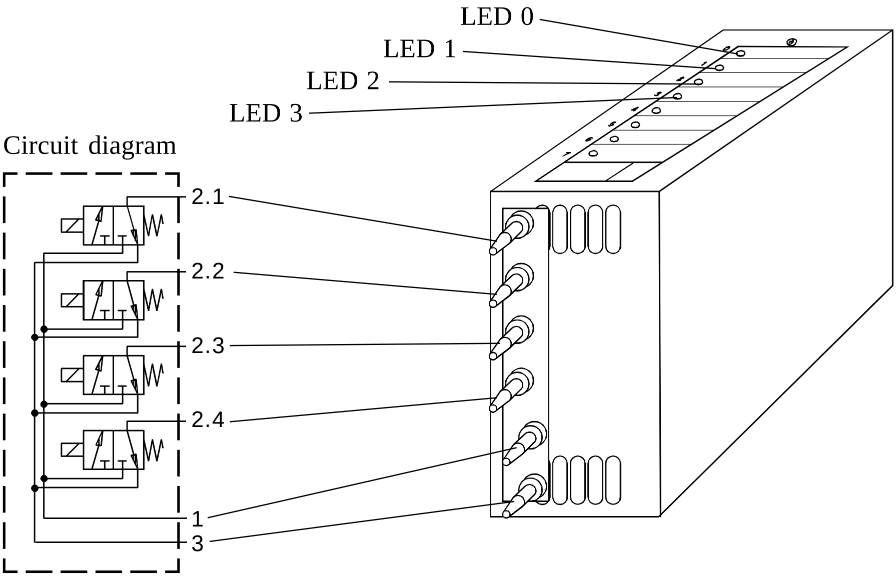


*) read 'A' = 'O' (for output) *) read 'E' = 'I' (for input)

4.4.1. Technical specifications

Application:	KUAX 680I (as from version 4.10), KUAX 680S (part no. 680.301.04/08) KUAX 680C, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs:	16
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	LEDs
colour	green
tapping point	in the input circuit
signal states	1: LED on 0: LED off
Addressing:	Ixx.00...Ixx.07
Input voltage:	24 V DC -20%/+25% (incl. residual ripple)
Surge immunity	≤ 60 V DC (≤ 30 min.)
Signal recognition	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Max. voltage:	28.8 V DC
Power consumption / input:	max. 10 mA
Outputs:	8
type	semiconductor
Indicators	LEDs
colour	red
tapping point	in the load current circuit
signal states	1: LED on 0: LED off
Addressing:	Oxx.00...Oxx.07
Output voltage:	24 V DC -20%/+25% (incl. residual ripple)
Output current:	max. 0.5 A
Short circuit protection:	yes
Weight:	c. 117 g
Order number:	680.450.02 (older version 680.450.01)

4.5. Pneumatic output module, 4 outputs 3/2 way



Connector	Function	Address	LED no.
2.1	pneum. output	Oxx.00	LED 0
2.2	pneum. output	Oxx.01	LED 1
2.3	pneum. output	Oxx.02	LED 2
2.4	pneum. output	Oxx.03	LED 3
1	air in	-	-
3	air out	-	-

Outputs

These are pneumatic outputs of 3/2 way directional control valves. The valves are addressed by the controller like normal digital outputs. The connection is made via the sleeve bits using 5 x 1 PE tube.

⚠ *Outgoing air should always be allowed to escape from the switching cabinet. Please note that unused outputs also give off air so that they must either be sealed or their air out led out of the cabinet..*

4.5.1. Technical specifications

Application:	KUAX 680I, 680C, 680S, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Outputs:	4
type	pneumatic
Indicators	LEDs
colour	red
signal states	1: LED on 0: LED off
Addressing:	Oxx.00...Oxx.03
Valves:	3/2 way directional control
Switching position:	normally closed (NC)
Nominal width (NW):	0.7 mm
Connector:	socket for 5 x 1 PE tube
Kv value:	0.21 l/min please ref. to catalogue P 411 GB
Max. pressure:	$P_{max} \le 7$ bar
Medium:	filtered (5 µm), oiled or filtered unoiled pressurized air. (Other gases such as helium, argon, or CO_2 can also be used. Please feel free to contact us.) ☞ If you are operating a component with oiled air once you have to make sure to continue operating it with oiled air as the oil will remove the initial lubrication.
Supply:	shared AIR IN connector(1) shared AIR OUT connector (3)
Switching diagram:	see left side
Weight:	c. 198 g
Order number:	680.453.01

5. Analogue inputs and outputs

Analogue inputs and outputs are used to lead analogue (i.e. permanently changeable) signals from the machine or the system into the controller (inputs) or vice versa (outputs).

These include:

Analogue signals such as
- temperature values
- liquid levels
- rotational speeds
- etc.

There is a great number of analogue modules available for the different applications. These are described on the following pages.

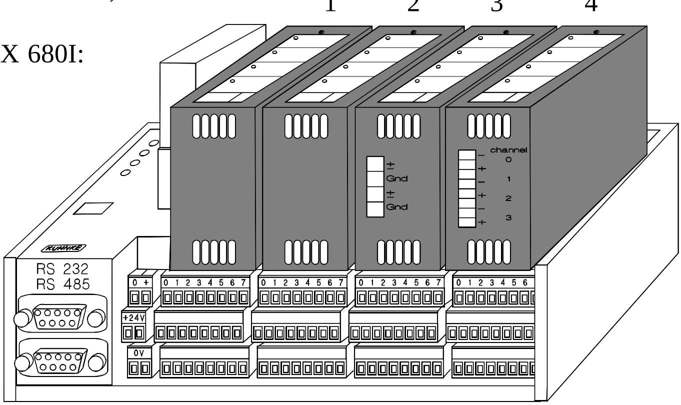
5.1. Plugging analogue input and output modules

When switching the controller on, all plugged-in modules are numbered automatically from left to right. Thus the first analogue input module is assigned the group number AI00, the second one becomes AI01 etc. (analogue outputs: AO00.00...). Analogue input modules with a resolution of 10 bit must only be plugged into certain slots (see “3.3.1. Function slots”) as they use the analogue/digital converter on the processor.
All other analogue modules can be plugged into any slot.

5.1.1. Addressing (anal. input/output groups and channels)

Upon switching on KUAX 680I, inputs and outputs of all plugged-in modules are automatically numbered from left to right in groups of a maximum of 4 channels (.00... .03) each. Thus, the first analogue input group is assigned group number AI00, the second one becomes AI01 etc. (for exceptions see ch. "3.6. Differences between KUAX 680C/KDT 680CT...") The analogue output groups are treated in the same way by the system: AO00, AO01 etc.

Example for KUAX 680I:



Legend:

1	analogue output module, 4pin	AO00
2	analogue output module, 4pin	AO01
3	analogue input/output module, 2/2pin	AI00, AO02
4	analogue input module, 4pin	AI01

Groups

Inputs:	AI00...max. AI07
Outputs:	AO00...max. AO07

In KUAX 680C and KDT 680CT, the first input group, AI00, and the first output group, AO00, are allocated to the internal inputs and outputs. The modules for inputs and outputs thus start with number AI01 and AO01 resp.

Channels

Each group consists of a maximum of 4 channels, i.e. inputs or outputs:

Inputs:	AIxx.00...AIxx.03
Outputs:	AOxx.00...AOxx.03

5.1.2. Reserved slots

Reserved slots for a module type of which there are modules plugged in already (e.g. input modules) must always be located after the last plugged-in module of the same type. Otherwise, the module numbering (addressing) of the following modules of the same type will change.

Example: AI AI _ AO

Reserved slot (e.g. for an
analogue input module)

Reserved slots for later plugging of function modules or analogue input modules may also be located between modules of one type.

Example: AI _ AI AO

AO00.

AI01.

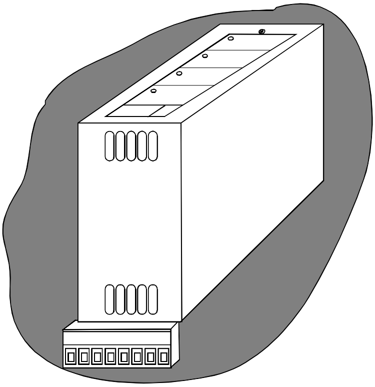
Reserved slot (e.g. for a counter module)

AI00.

5.2. Analogue input modules

Analogue inputs are used to lead analogue (i.e. permanently changeable) signals from the machine or the system into the controller. These can be temperature values, liquid levels, rotational speeds etc.

5.2.1. Analogue input module, 0...10 V, 10 bit, 4 channels



5.2.1.1. Slots

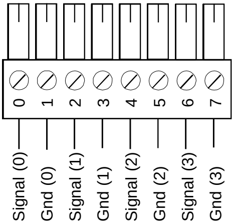
This module uses the internal analogue/digital converter of the processor on the CPU board. For this purpose, certain slots carry additional leads. The module can only be operated on these slots (see ch. “3.3. Slots”):

Slot no.	Useable channels
2	0, 1 *)
3	0, 1, 2, 3
4	0, 1, 2, 3

*) Channels 2 and 3 cannot be used if this module is applied to slot 2. In this case, the PWM outputs - also on slot 1 - are switched off internally.

5.2.1.2. Connectors

The connectors of the signal lines are located on the 8-pin plug-screw connector underneath the module (see diagram on previous page):



⚠ *Both leads, signal and Gnd$_n$, must definitely be connected for each channel (0...3). The Gnd$_n$-connections are not identical with the device-Gnd.*

Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

5.2.1.3. Representation of the analogue value

The read analogue value is digitalized and then written into a 16bit address as a two's complement representation. This address contains the value in bits 5...14. Bits 0...4 and 15 (sign bit) always have the status 0:

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	0	------------ value read -----------										0	0	0	0	0

In the user program, the value is read in a double-byte operation.

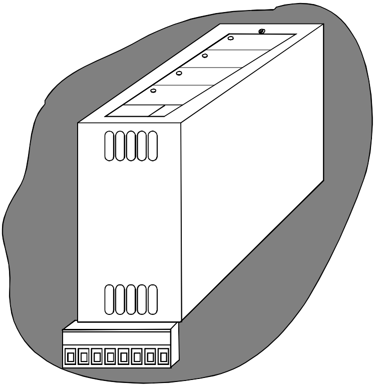
Example:

```
LD          AI00.00
CMPD>=      4V          ;input range 0...10.00V
=           M00.01
```

5.2.1.4. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs (channels):	4
Potential separation	no
Addressing:	AIxx.00...AIxx.03
Measuring range:	0...10 V
Resolution:	10 bit, ~ 0.01 V / digit
Precision (relative to final value)	
max. error (at 25 °C)	± 0.6 %
norm. error (at 25 °C)	± 0.3 %
temperature coefficient	0.01 % /K
linearity error	± 0.1 %
Conversion time:	10 ms
Input voltage protection:	60 V
Protection against noise pulses:	by filters and buffers
Weight	c. 77 g
Part number:	680.441.01

5.2.2. Analogue input module, 0(4)...20 mA, 10bit, 4 channels



5.2.2.1. Slots

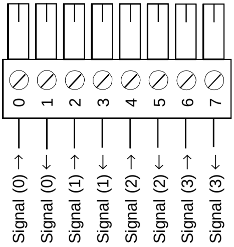
This module uses the analogue/digital converter of the processor on the CPU. For this purpose, certain slots carry additional leads. The module can only be operated on these slots (see “3.3. Slots”).

Slot no.	Useable channels
2	0, 1 *)
3	0, 1, 2, 3
4	0, 1, 2, 3

*) Channels 2 and 3 cannot be used if this module is applied to slot 2. In this case, the PWM outputs - also on slot 1 - are switched off internally.

5.2.2.2. Connections

The connections of the signal lines are on the 8-pin plug-screw connector underneath the module (see diagram on previous page):



Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

5.2.2.3. Representation of the analogue value

The read analogue value is digitalised and then written into a 16bit address as a two's complement representation. This address contains the value in bits 5...14. Bits 0...4 and 15 (sign bit) always have the status 0:

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	0	------------ value read -----------										0	0	0	0	0

In the user program, the value is read in a double-byte operation.

Example:

```
LD          AI00.00
CMPD<=      8mA         ;input range 0(4)...20.00mA
=           M00.01
```

☞ *Depending on the setting of the corresponding transfer address (see next page) the input value is defined as a 0...20mA or a 4...20mA value.*

5.2.2.4. Transfer address ranges

Each module is assigned a transfer address range of 16 byte (see also "3.4. Service modules"). This range is used by the analogue input module for data exchange with the CPU.

The address range depends on the slot:

Slot	Transfer address range
2	SLE00.00...01.15
3	SLG00.00...01.15
4	SLI00.00...01.15

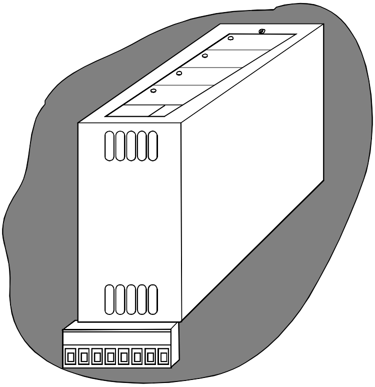
Assignment of transfer addresses

Address	Symbol	Significance			
		Value	Signal	Chan.	
SLx00.00	I_SIG_0	=0 <>0	0...20 mA 4...20 mA	0	user preselects the type of signal (default = 0)
SLx00.01	I_SIG_1			1	
SLx00.02	I_SIG_2			2	
SLx00.03	I_SIG_3			3	
SLx01.00	FAIL_0	=255	wire failure	0	error message from the service module, only with 4...20 mA signals
SLx01.01	FAIL_1	=255	wire failure	1	
SLx01.02	FAIL_2	=255	wire failure	2	
SLx01.03	FAIL_3	=255	wire failure	3	

5.2.2.5. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs (channels):	4
Potential separation	no
Addressing:	AIxx.00...AIxx.03
Measuring range:	0...20 mA or 4...20 mA
Resolution:	10 bit, ~ 0.02 mA / digit for 0...20 mA ~ 0.016 mA / digit for 4...20 mA
Precision (relative to final value)	
max. error (at 25 °C)	± 0.8 %
norm. error (at 25 °C)	± 0.4 %
temperature coefficient	0.02 % /K
linearity error	± 0.1 %
Conversion time:	10 ms
Input resistance	~ 40 Ω
Input voltage protection:	60 V
Protection against noise pulses:	by filters and buffers
Indicated failures	wire failure
Weight	c. 79 g
Part number:	680.441.02

5.2.3. Analogue input module, PT100, 0...300 °C, 10 bit, 4 channels, two-wire connection



5.2.3.1. Slots

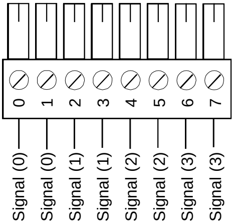
This module uses the analogue/digital converter of the processor on the CPU. For this purpose, certain slots carry additional leads. The module can only be operated on these slots (see “3.3. Slots”):

Slot no.	Useable channels
2	0, 1 *)
3	0, 1, 2, 3
4	0, 1, 2, 3

*) Channels 2 and 3 cannot be used if this module is applied to slot 2. In this case, the PWM outputs - also on slot 1 - are switched off internally.

5.2.3.2. Connectors

The connectors of the signal lines are located on the 8-pin plug-screw connector underneath the module (see diagram on previous page):



☞ *Unassigned inputs put an unnecessary load on the power supply. Unused channels should therefore be terminated by 100 Ω or should be short-circuited.*

Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

5.2.3.3. Transfer address ranges

Each module is assigned a transfer address range of 16 byte (see also "3.4. Service modules"). This range is used by the analogue input module for data exchange with the CPU.

The address range depends on the slot:

Slot	Transfer address range
2	SLE00.00...01.15
3	SLG00.00...01.15
4	SLI00.00...01.15

Assingment of transfer addresses

Address	Symbol	Significance	
SLx00.00	TMP_CH0	Lowbyte	temperature value channel 0, 0...300.0 °C
SLx00.01		Highbyte	
SLx00.02	TMP_CH1	Lowbyte	temperature value channel 1, 0...300.0 °C
SLx00.03		Highbyte	
SLx00.04	TMP_CH2	Lowbyte	temperature value channel 2, 0...300.0 °C
SLx00.05		Highbyte	
SLx00.06	TMP_CH3	Lowbyte	temperature value channel 3, 0...300.0 °C
SLx00.07		Highbyte	
SLx00.08	FAIL_0	255 = wire failure or overtemperature channel 0	
SLx00.09	FAIL_1	255 = wire failure or overtemperature channel 1	
SLx00.10	FAIL_2	255 = wire failure or overtemperature channel 2	
SLx00.11	FAIL_3	255 = wire failure or overtemperature channel 3	
SLx00.12	FAIL_MOD	255 = wire failure or overtemperature in module	

5.2.3.4. Evaluation in the user program

There are two ways of evaluating the input information in the user program:

As temperature value in steps of 1/10 °C (0...3000 ≙ 0...300.0 °C)

The temperature is represented as 12bit value (in 2 bytes) in the transfer addresses (see table on previous page).
In the user program, the temperature value is read in a double-byte operation.

Example:

```
LD          SLE00.00   ;temperature value channel 0
CMPD>=      1500       ;greater/equal 150 °C ?
=           M00.01     ;set marker if so
```

As analogue value in a 10bit resolution

The resolution corresponds to the Kuhnke standard for analogue inputs. The temperature is represented in the 16 bit analogue input addresses (AI...). The addressing corresponds to the sequence of the plugged-in analogue input modules (see "5.1. Plugging analogue input and output modules").
In this address the value is in bits 5...14. Bits 0...4 and 15 (sign bit) always have the status 0:

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	0	------------ value read -----------										0	0	0	0	0

In the user program, the value is read in a double-byte operation. Scales 0...10.00V or 0...20.00mA can be used as reference.

Example:

```
LD          AI00.00    ;analogue value channel 0
CMPD>=      5.00V       ;greater/equal 150 °C?
=           M00.01      ;set marker if so
```

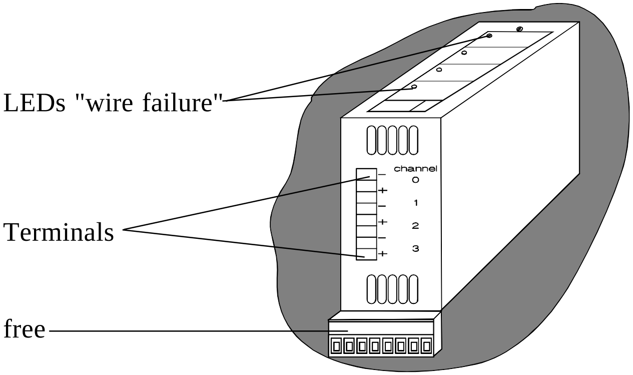
5.2.3.5. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs (channels):	4
Potential separation	no
Addressing:	
temperature value:	SLx... (0...3000 1/10 °C)
analogue value	AIxx.00...AIxx.03
Measuring range:	0...317 °C
Resolution:	10 bit, ~ 0.31 °C / digit
Precision (relative to final value)	
max. error (at 25 °C)	± 1.0 %
- up until prod. date Sept. 1996	± 1.5 %
norm. error (at 25 °C)	± 0.3 %
- up until prod. date Sept. 1996	± 1.0 %
temperature coefficient	0.03 % /K
linearity error	± 0.2 %
Conversion time:	10 ms
Input voltage protection:	60 V
Protection against noise pulses:	by filters and buffers
Indicated failures	wire failure, overtemperature
Weight	c. 78 g

Part number:	680.441.04

5.2.4. Analogue input module, thermocouple NiCrNi (type K), 0...1200 °C, 10 bit, 4 channels

This module serves registering temperature values by thermocouples. The inputs are difference inputs which are linearised on the module. Wire failures are recognized and indicated on the module by 4 red LEDs.



5.2.4.1. Slots

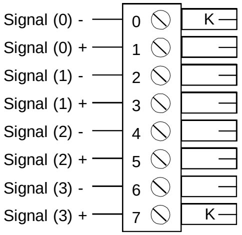
This module uses the analogue/digital converter of the processor on the CPU. For this purpose, certain slots carry additional leads. The module can only be operated on these slots (see “3.3. Slots”).

Slot no.	Useable channels
2	0, 1 *)
3	0, 1, 2, 3
4	0, 1, 2, 3

*) Channels 2 and 3 cannot be used if this module is applied to slot 2. In this case, the PWM outputs - also on slot 1 - are switched off internally.

5.2.4.2. Connections

The connections of the signal lines are on the side of the module. These are coded plug-screw connectors which can only be plugged into this location.



K = remove coding element

The signal strip underneath the module remains unoccupied.

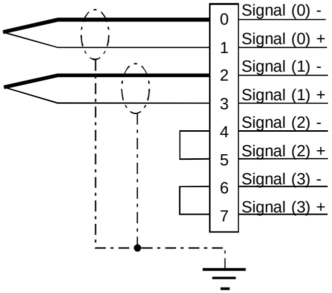
☞ *Open inputs put unnecessary load on the power supply. Unused channels should therefore be short-circuited.*

Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

Example for connection:

Channels 0 and 1 are connected, channels 2 and 3 are free



5.2.4.3. Transfer address ranges

Each module is assigned a transfer address range of 16 byte (see also "3.4. Service modules"). This range is used by the analogue input module for data exchange with the CPU.

The address range depends on the slot:

Slot	Transfer address range
2	SLE00.00...01.15
3	SLG00.00...01.15
4	SLI00.00...01.15

Assignment of transfer addresses

Address	Symbol	Significance	
SLx00.00	TMP_CH0	Lowbyte	temperature value channel 0, 0...1200 °C
SLx00.01		Highbyte	
SLx00.02	TMP_CH1	Lowbyte	temperature value channel 1, 0...1200 °C
SLx00.03		Highbyte	
SLx00.04	TMP_CH2	Lowbyte	temperature value channel 2, 0...1200 °C
SLx00.05		Highbyte	
SLx00.06	TMP_CH3	Lowbyte	temperature value channel 3, 0...1200 °C
SLx00.07		Highbyte	
SLx00.08	FAIL_0	255 = wire failure channel 0	
SLx00.09	FAIL_1	255 = wire failure channel 1	
SLx00.10	FAIL_2	255 = wire failure channel 2	
SLx00.11	FAIL_3	255 = wire failure channel 3	
SLx00.12	FAIL_MOD	255 = wire failure module	

5.2.4.4. Evaluation in the user program

There are two ways of evaluating the input information in the user program:

As temperature value in steps of 1 °C (0...1200 ≙ 0...1200 °C)

The temperature is represented as 12bit value (in 2 bytes) in the transfer addresses (see table on previous page).
In the user program, the temperature value is read in a double-byte operation.

Example:

```
LD          SLE00.00    ;temperature value channel 0
CMPD>       600         ;greater 600 °C ?
=           M00.01      ;set marker if yes
```

As analogue value in a 10bit resolution

The resolution corresponds to the Kuhnke standard for analogue inputs. The temperature is represented in the 16 bit analogue input addresses (AI...). The addressing corresponds to the sequence of the plugged-in analogue input modules (see "5.1. Plugging analogue input and output modules").
In this address the value is in bits 5...14. Bits 0...4 and 15 (sign bit) always have the status 0:

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	0	------------ value read -----------										0	0	0	0	0

In the user program, the value is read in a double-byte operation. Scales 0...10.00V or 0...20.00mA can be used as reference.

Example:

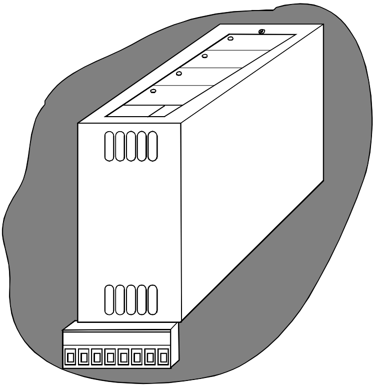
```
LD          AI00.00     ;analogue value channel 0
CMPD>       5.00V       ;greater 600 °C?
=           M00.01      ;set marker if so
```

5.2.4.5. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs (channels):	4
Potential separation	no
Addressing:	SLx... (0...1200 °C)
Analog value	AIxx.00...AIxx.03
Measuring range:	0...1200 °C (linearised on the module)
Resolution:	10 bit, ~ 1.17 °C / digit
Precision (relative to final value)	
max. error (at 25 °C)	± 1.0 %
norm. error (at 25 °C)	± 0.6 %
temperature coefficient	0.02 % /K
linearity error	± 0.2 %
Conversion time:	10 ms
Input voltage protection:	60 V
Protection against noise pulses:	by filters and buffers
Wire failure monitoring	failures indicated by LEDs
Weight	c. 93 g
Part number:	680.441.07

5.2.5. Analogue input module, potentiometer, 10bit, 4 channels

Analogue set point modules are input modules for the connection of potentiometers which are used for presetting values for: temperatures, liquid levels, rotation speeds, times etc.



5.2.5.1. Slots

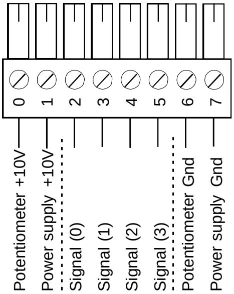
This module uses the analogue/digital converter of the processor on the CPU. For this purpose, certain slots carry additional leads. The module can only be operated on these slots (see “3.3. Slots”).

Slot no.	Useable channels
2	0, 1 *)
3	0, 1, 2, 3
4	0, 1, 2, 3

*) Channels 2 and 3 cannot be used if this module is applied to slot 2. In this case, the PWM outputs - also on slot 1 - are switched off internally.

5.2.5.2. Connectors

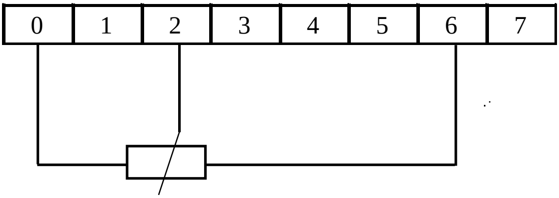
The connections for the potis are on the 8-pin plug-screw connector underneath the module. The supply (10V) is generated in the power pack of the control (system voltage):



Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable my be commendable).

Example for the connections of channel 0:



5.2.5.3. Representation of the analogue value

The read analogue value is digitalised and then written into a 16bit address as a two’s complement representation. This address contains the value in bits 5...14. Bits 0...4 and 15 (sign bit) always have the status 0:

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	0	------------ value read -----------										0	0	0	0	0

In the user program, the value is read in a double-byte operation.

Example:

```
LD          AI00.00
CMPD<= 5V          ( ≙ 50%)
=           M00.01
```

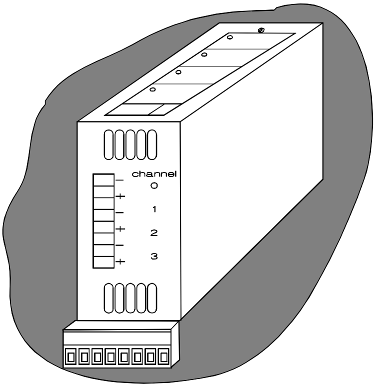
5.2.5.4. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Inputs (channels):	4 potentiometer inputs
Potential separation	no
Addressing:	AIxx.00...AIxx.03
Potentiometer scale:	4.7 kΩ ... 1 kΩ
Load resistance:	4.7 kΩ min.
Total load / module	8.5 mA max.
Measuring range:	0...10 V
Resolution:	10 bit, ~ 0.01 V / digit
Precision (relative to final value)	
max. error (at 25 °C)	± 0.6 %
norm. error (at 25 °C)	± 0.3 %
temperature coefficient	0.02 % /K
linearity error	± 0.1 %
Conversion time:	10 ms
Input voltage protection:	60 V
Protection against noise pulses:	by filters and buffers
Weight	c. 78 g
Part number:	680.441.05

5.3. Analogue output modules

Analogue outputs are lead from the controller into the machine or system. They may include: temperature values, liquid levels, speed settings etc.

5.3.1. Analogue output module, 0...10 V, 8 bit, 4 channels

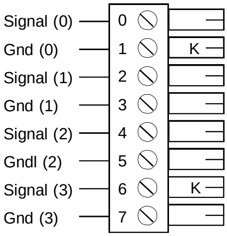


5.3.1.1. Slots

This module has its own digital-to-analogue converter. It can therefore be plugged into any slot of the controller.

5.3.1.2. Connectors

The connectors of the signal lines are on the 8-pin plug-screw connector underneath the module (see diagram on previous page):



K = remove coding element

⚠ *Both wires, signal and Gnd$_n$, must be connected for each channel. The Gnd$_n$ connections are not identical with the device's Gnd.*

Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

5.3.1.3. Repesentation of the analogue value

The read analogue value is digitalized and written into a 16 bit address as a two's complement representation. In the address, the value is in bits 7...14. Bits 0...6 are not evaluated, bit 15 (sign bit) always has the status 0:

Address bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status:	0	-------output value-------								x	x	x	x	x	x	x

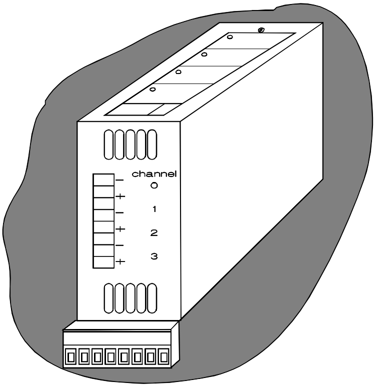
In the user program, the value is entered in a double-byte operation.

```
Example:          LD     5.5V         ;input range 0...10.00V
                  =D     AO00.00
```

5.3.1.4. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Outputs (channels):	4, short-circuit prot., single-ended
Short-circuit current	15 mA max.
Potential separation	no
Addressing:	AOxx.00...AOxx.03
Range:	0...10 V
Burden	2 kΩ min.
Resolution:	8 bit, ~ 0.04 V / digit
Transient building-up period	norm. 0.05 ms
Precision (relative to final value)	
max. error (at 25 °C)	± 1.0 %
norm. error (at 25 °C)	± 0.5 %
temperature coefficient	0.02 % /K
linearity error	± 0.4 %
Weight	c. 85 g
Part number:	680.442.01

5.3.2. Analogue output module, 0(4)...20 mA, 8bit, 4 channels

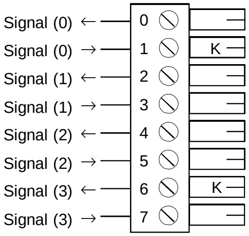


5.3.2.1. Slots

This module has its own digital-to-analogue converter. It can therefore be plugged into any slot of the controller.

5.3.2.2. Connectors

The connectors of the signal lines are on the 8-pin plug-screw connector underneath the module (see diagram on previous page):



K = remove coding element

Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

5.3.2.3. Repesentation of the analogue value

The read analogue value is digitalised and written into a 16 bit address as a two's complement representation. In the address, the value is in bits 7...14. Bits 0...6 are not evaluated, bit 15 (sign bit) always has the status 0:

Address bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status:	0	-------output value-------								x	x	x	x	x	x	x

In the user program, the value is entered in a double-byte operation.

Example:

```
LD      10 mA           ;input range 0(4)...20 mA
=D      AO00.00
```

☞ *Depending on the setting of the corresponding transfer address (see below), the input value is defined as 0...20 or 4...20mA value.*

5.3.2.4. Transfer address ranges

Each module is assigned a transfer address range of 32 byte (see also "3.4. Service modules"). The first group of this range is used by the analogue output module to determine the range of the output signal.
The address range depends on the slot:

Slot	Transfer address range
0	SLA00.00...01.15
1	SLC00.00...01.15
2	SLE00.00...01.15
3	SLG00.00...01.15
4	SLI00.00...01.15
5	SLK00.00...01.15
6	SLM00.00...01.15
7	SLO00.00...01.15

Assignment of transfer addresses

Address	Symbol	Significance			
		Value	Signal	Chan.	
SLx00.08	O_SIG_0	=0 <>0	0...20 mA 4...20 mA	0	User preselects the type of signal (default = 0)
SLx00.09	O_SIG_1			1	
SLx00.10	O_SIG_2			2	
SLx00.11	O_SIG_3			3	

5.3.2.5. Technical specifications

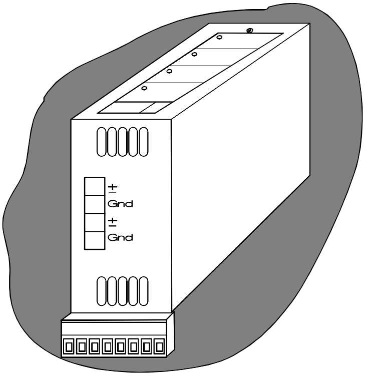
Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Outputs (channels):	4
Potential separation	no
Addressing:	AOxx.00...AOxx.03
Range:	0(4)...20 mA
Resolution:	8 bit, ~ 0.08 mA / digit for 0...20 mA ~ 0.06 mA / digit for 4...20 mA
Transient building-up period	norm. 0.015 ms
Precision (relative to final value)	
max. error (at 25 °C)	± 1.0 %
norm. error (at 25 °C)	± 0.5 %
temperature coefficient	0.02 % /K
linearity error	± 0.4 %
No-load voltage	15 V max.
Burden:	600 Ω max.
Weight	c. 86 g
Part number:	680.442.02

5.4. Analogue input/output modules

These modules have analogue inputs and analogue outputs. They are mainly used in cases where only a small amount of channels is required.

5.4.1. Analogue I/O module, 2 I 0...10 V, 2 O 0...±10V

Inputs:	0...10 V, 10 bit, 2 channels
Outputs:	-10...0...+10 V, 12 bit, 2 channels



5.4.1.1. Slots

Analogue inputs

This module uses the analogue-to-digital converter of the processor on the CPU for the analogue inputs. For this purpose, certain slots carry additional leads. If the analogue inputs are used, then the module can only be operated on these slots (see "3.3. Slots”):

Slot no.	Useable channels
2	0, 1
3	0, 1
4	0, 1

As there are only 2 analogue inputs the module should preferably be plugged into slot 2 as long as this is not occupied already.

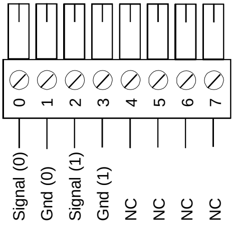
Analogue outputs

The digital-to-analogue converter can convert 12bit digital values (including signs).
If only the analogue outputs of the module are to be used, then the module can also be plugged into any other slot.

5.4.1.2. Connectors

Analogue inputs

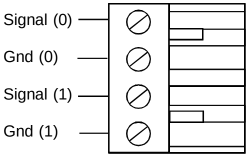
The connections of the signal lines for the analogue inputs are on the 8-pin plug-screw connector underneath the module:



⚠ *Both leads, signal and Gnd$_n$, must definitely be connected for each channel. The Gnd$_n$-connections are not identical with the device-Gnd.*

Analogue outputs

The connections of the signal lines for the analogue outputs are on the side of the module. A 4-pin plug-screw connector (5.08 mm grid) is used, which is delivered together with the module.



Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commmendable).

5.4.1.3. Representation of the analogue value

Analogue inputs

The read analogue value is digitalized and then written into a 16bit address as a two’s complement representation. In this address, the value is in bits 5...14. Bits 0...4 and 15 (sign bit) always have the status 0:

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	0	------------ value read -----------										0	0	0	0	0

In the user program, the value is read in a double-byte operation.

Example:

```
LD       AI00.00
CMPD<    5.5V          ;input range 0...10.00V
=        M00.01
```

Analogue outputs

The analogue value to be output must be written into a 16bit address as a two’s complement representation by the user program. In this address, the value is in bits 4...14 and the sign in bit 15. Bits 0...3 are not evaluated.

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	±	------------ output value -------------											x	x	x	x

In the user program, the value is written in a double-byte operation.

Example:

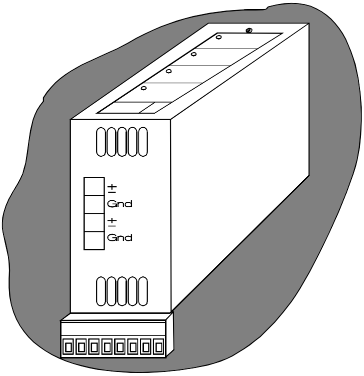
```
LD     5.5V          ;input range 0...±10.00V
=D     AO00.00
```

5.4.1.4. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation	0...55 °C
relative humidity	50...95 %
Inputs	2
Potential separation	no
Addressing:	AIxx.00...AIxx.01
Measuring range:	0...10 V
Resolution:	10bit, ~ 0.01 V / digit
Precision (relative to final value)	
max. error (at 25 °C)	± 0.6 %
norm. error (at 25 °C)	± 0.3 %
temperature coefficient	0.01 % /K
linearity error	± 0.1 %
Conversion time:	10 ms
Input voltage protection:	60 V
Protection against noise pulses:	by filters and buffers
Outputs	2, short-circuit protected
Potential separation:	no
Addressing:	AOxx.00...AOxx.01
Range:	-10...0...+10 V
Max. output current:	5 mA max, 10 nF
Resolution:	11bit + sign, ~ 0.005 V / digit
Transient building-up period:	norm. 0.07 ms
Burden:	2 kΩ min.
Short circuit current:	15 mA max.
Precision (relative to final value)	
max. error (at 25 °C)	± 0.4 %
norm. error (at 25 °C)	± 0.1 %
temperature coefficient	0.01 % /K
linearity error	± 0.1 %
Sign:	1bit
Weight:	c. 98 g
Part number:	680.441.03

5.4.2. Analogue I/O module, 2 I 0...20 mA, 2 O 0...±10V

Inputs:	0(4)...20 mA, 10 bit, 2 channels
Outputs:	-10...0...+10 V, 12 bit, 2 channels



5.4.2.1. Slots

Analogue inputs

This module uses the analogue-to-digital converter on the processor of the CPU for the analogue inputs. For this purpose, certain slots carry additional leads. If the analogue inputs are used, then the module can only be operated on these slots (see 3.3. Slots”).

Slot no.	Useable channels
2	0, 1
3	0, 1
4	0, 1

As there are only 2 analogue inputs the module should preferably be plugged into slot 2 as long as this is not occupied already.

Analogue outputs

The digital-to-analogue converter can convert 12bit digital values (including sign).

If only the analogue outputs of a module are to be used, then the module can also be plugged into any other slot.

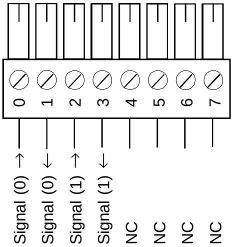
Exception for KUAX 680I with 2 RS-485 connectors:

Do not plug into slot 0 as this might lead to collisions with the V.24 leads.

5.4.2.2. Connectors

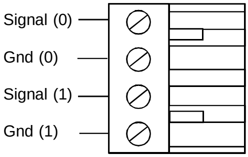
Analogue inputs

The connections of the signal lines for the analogue inputs are on the 8-pin plug-screw connector underneath the module:



Analogue outputs

The connections of the signal lines for the analogue outputs are on the side of the module. A 4-pin plug-screw connector (5.08 mm grid) is used, which is delivered together with the module.



Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

5.4.2.3. Representation of the analogue value

Analogue inputs

The read analogue value is digitalized and then written into a 16bit address as a two’s complement representation. In this address, the value is in bits 5...14. Bits 0...4 and 15 (sign bit) always have the status 0:

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	0	------------ value read -----------										0	0	0	0	0

In the user program, the value is read in a double-byte operation.

Example:

```
LD          AI00.00
CMPD>=      8mA          ;input range 0...20.00mA
=           M00.01
```

☞ *Depending on the setting of the corresponding transfer address (see next page), the input value is defined as 0...20 mA or 4...20 mA value.*

Analogue outputs

The analogue value to be output must be written into a 16bit address as a two’s complement representation by the user program. In this address, the value is in bits 4...14 and the sign in bit 15. Bits 0...3 are not evaluated.

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	±	------------ output value -------------											x	x	x	x

In the user program, the value is written in a double-byte operation.

Example:

```
LD     5.5V          ;input range -10.00V...+10.00V
=D     AO00.00
```

5.4.2.4. Transfer address ranges of the analogue inputs

Each module is assigned a transfer address range of 32 byte (see also "3.4. Service modules"). In this range, the analogue input/output module uses the first group to determine the range of the input signal. Furthermore, the service module reports any occurring wire failure to this range.
The address range depends on the slot:

Slot	Transfer address range
0	SLA00.00...01.15
1	SLC00.00...01.15
2	SLE00.00...01.15
3	SLG00.00...01.15
4	SLI00.00...01.15
5	SLK00.00...01.15
6	SLM00.00...01.15
7	SLO00.00...01.15

Assignment of the transfer addresses

Address	Symbol	Significance			
		Value	Signal	Chan.	
SLx00.00	I_SIG_0	=0 <>0	0...20 mA 4...20 mA	0	user preselects the type of signal (default = 0)
SLx00.01	I_SIG_1			1	
SLx00.02	I_SIG_2			2	
SLx00.03	I_SIG_3			3	
SLx01.00	FAIL_0	=255	wire failure	0	error message from the service module, only with 4...20 mA signals
SLx01.01	FAIL_1	=255	wire failure	1	
SLx01.02	FAIL_2	=255	wire failure	2	
SLx01.03	FAIL_3	=255	wire failure	3	

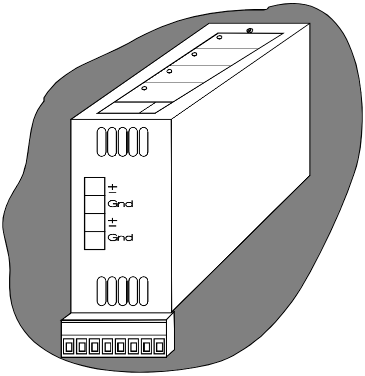
5.4.2.5. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation	0...55 °C
relative humidity	50...95 %
Inputs	2
Potential separation	no
Addressing:	AIxx.00...AIxx.01
Measuring range:	0(4)...20 mA
Internal resistance:	~ 40 Ω
Resolution:	10bit, ~ 0.02 mA / digit for 0...20 mA ~ 0.016 mA / digit for 4...20 mA
Precision (relative to final value)	
max. error (at 25 °C)	± 0.8 %
norm. error (at 25 °C)	± 0.4 %
temperature coefficient	0.02 % /K
linearity error	± 0.1 %
Conversion time:	10 ms
Input resistance:	~ 40 Ω
Protection against noise pulses:	by filters and buffers
Indicated failures:	wire failure
Outputs	2, short-circuit protected
Potential separation	no
Addressing:	AOyy.00...AOyy.01
Range:	-10...0...+10 V
Output current:	5 mA max, 10 nF
Resolution:	11bit, ~ 0.005 V / digit
Transient building-up period	norm. 0.07 ms
Burden:	2 kΩ min.
Short circuit current:	15 mA max.

Precision (relative to final value)	
max. error (at 25 °C)	± 0.4 %
norm. error (at 25 °C)	± 0.1 %
temperature coefficient	0.01 % /K
linearity error	± 0.1 %
Sign:	1bit
Weight:	c. 98 g
Part number:	680.441.06

5.4.3. Analogue I/O module, 2 I 0...10 V, 2 O 0...20 mA

Inputs:	0...10 V, 10 bit, 2 channels
Outputs:	0(4)...20 mA, 12 bit, 2 channels



5.4.3.1. Slots

Analogue inputs

This module uses the analogue-to-digital converter of the processor on the CPU for the analogue inputs. For this purpose, certain slots carry additional leads. If the analogue inputs are used, then the module can only be operated on these slots (see "3.3. Slots”):

Slot no.	Useable channels
2	0, 1
3	0, 1
4	0, 1

As there are only 2 analogue inputs the module should preferably be plugged into slot 2 as long as this is not occupied already.

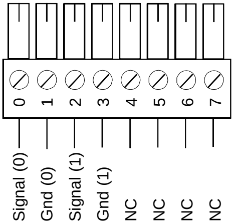
Analogue outputs

The digital-to-analogue converter can convert 12bit digital values (including signs).
If only the analogue outputs of the module are to be used, then the module can also be plugged into any other slot.

5.4.3.2. Connectors

Analogue inputs

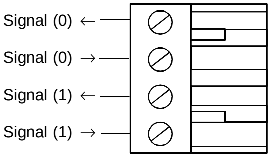
The connections of the signal lines for the analogue inputs are on the 8-pin plug-screw connector underneath the module:



⚠ *Both leads, signal and Gnd$_n$, must definitely be connected for each channel. The Gnd$_n$-connections are not identical with the device-Gnd.*

Analogue outputs

The connections of the signal lines for the analogue outputs are on the side of the module. A 4-pin plug-screw connector (5.08 mm grid) is used, which is delivered together with the module.



Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commmendable).

5.4.3.3. Representation of the analogue value

Analogue inputs

The read analogue value is digitalized and then written into a 16bit address as a two’s complement representation. In this address, the value is in bits 5...14. Bits 0...4 and 15 (sign bit) always have the status 0:

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	0	------------ value read -----------										0	0	0	0	0

In the user program, the value is read in a double-byte operation.

Example:

```
LD      AI00.00
CMPD<   5.5V        ;input range 0...10.00V
=       M00.01
```

Analogue outputs

The analogue value to be output must be written into a 16bit address as a two’s complement representation by the user program. In this address, the value is in bits 4...14 and the sign in bit 15. Bits 0...3 are not evaluated.

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	±	------------ output value -------------											x	x	x	x

In the user program, the value is written in a double-byte operation.

Example:

```
LD      11mA        ;input range 0...±10.00V
=D      AO00.00
```

☞ *Depending on the setting of the corresponding transfer address (see next page), the input value is defined as 0...20 mA or 4...20 mA value.*

5.4.3.4. Transfer address ranges of the analogue inputs

Each module is assigned a transfer address range of 32 byte (see also "3.4. Service modules"). In this range, the analogue input/output module uses the first group to determine the range of the input signal. Furthermore, the service module reports any occurring wire failure to this range.
The address range depends on the slot:

Slot	Transfer address range
0	SLA00.00...01.15
1	SLC00.00...01.15
2	SLE00.00...01.15
3	SLG00.00...01.15
4	SLI00.00...01.15
5	SLK00.00...01.15
6	SLM00.00...01.15
7	SLO00.00...01.15

Assignment of transfer addresses

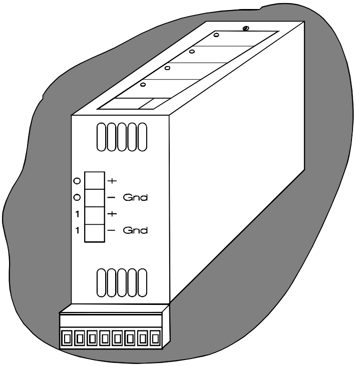
Address	Symbol	Significance			
		Value	Signal	Chan.	
SLx00.08	O_SIG_0	=0 <>0	0...20 mA 4...20 mA	0	User preselects the type of signal (default = 0)
SLx00.09	O_SIG_1			1	
SLx00.10	O_SIG_2			2	
SLx00.11	O_SIG_3			3	

5.4.3.5. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation	0...55 °C
relative humidity	50...95 %
Inputs	2
Potential separation	no
Addressing:	AIxx.00...AIxx.01
Measuring range:	0...10 V
Resolution:	10bit, ~ 0.01 V / digit
Precision (relative to final value)	
max. error (at 25 °C)	± 0.6 %
norm. error (at 25 °C)	± 0.3 %
temperature coefficient	0.01 % /K
linearity error	± 0.1 %
Conversion time:	10 ms
Input voltage protection:	60V
Protection against noise pulses:	by filters and buffers
Outputs	2, short-circuit protected
Potential separation	no
Addressing:	AOyy.00...AOyy.01
Range:	0(4)...20 mA
Max. output current:	5 mA
Resolution:	11bit, ~ 0.01 mA / digit for 0...20 mA ~ 0.008 mA / digit for 4...20 mA
Transient building-up period:	norm. 0.02 ms
Burden:	600 Ω max.
No-load voltage:	15 V max.
Precision (relative to final value)	
max. error (at 25 °C)	± 0.6 %
norm. error (at 25 °C)	± 0.2 %
temperature coefficient	0.02 % /K
linearity error	± 0.1 %
Weight:	c. 90 g
Part number:	680.441.08

5.4.4. Anal. I/O module, 2 I 0...20 mA, 2 O 0...20 mA

Inputs:	0(4)...20 mA, 10 bit, 2 channels
Outputs:	0(4)...20 mA, 12 bit, 2 channels



5.4.4.1. Slots

Analogue inputs

This module uses the analogue-to-digital converter on the processor of the CPU for the analogue inputs. For this purpose, certain slots carry additional leads. If the analogue inputs are used, then the module can only be operated on these slots (see "3.3. Slots”).

Slot no.	Useable channels
2	0, 1
3	0, 1
4	0, 1

As there are only 2 analogue inputs the module should preferably be plugged into slot 2 as long as this is not occupied already.

Analogue outputs

The digital-to-analogue converter can convert 12bit digital values (including sign).

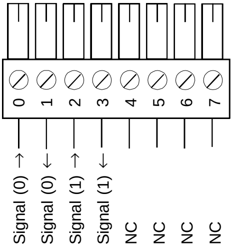
If only the analogue outputs of a module are to be used, then the module can also be plugged into any other slot.

⚠ *Exception for devices with 2 RS-485 connectors:*
Do not plug into slot 0 as this might lead to collisions with the V.24 leads.

5.4.4.2. Connectors

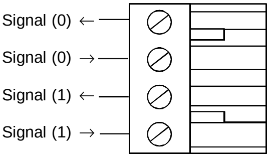
Analogue inputs

The connections of the signal lines for the analogue inputs are on the 8-pin plug-screw connector underneath the module:



Analogue outputs

The connections of the signal lines for the analogue outputs are on the side of the module. A 4-pin plug-screw connector (5.08 mm grid) is used, which is delivered together with the module.



Shielding

Shielded cables have to be used for connecting the analogue signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

5.4.4.3. Representation of the analogue value

Analogue inputs

The read analogue value is digitalised and then written into a 16bit address as a two’s complement representation. In this address, the value is in bits 5...14. Bits 0...4 and 15 (sign bit) always have the status 0:

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	0	------------ value read -----------										0	0	0	0	0

In the user program, the value is read in a double-byte operation.

Example:

```
LD          AI00.00
CMPD>=      8mA          ;input range 0...20.00mA
=           M00.01
```

Analogue outputs

The analogue value to be output must be written into a 16bit address as a two’s complement representation by the user program. In this address, the value is in bits 4...14 and the sign in bit 15. Bits 0...3 are not evaluated.

Address bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	±	------------ output value -------------											x	x	x	x

In the user program, the value is written in a double-byte operation.

Example:

```
LD     11mA        ;input range 0...20mA
=D     AO00.00
```

☞ *Depending on the setting of the corresponding transfer address (see next page), the input value is defined as 0...20 mA or 4...20 mA value.*

5.4.4.4. Transfer address ranges

Each module is assigned a transfer address range of 32 byte (see also "3.4. Service modules"). In this range, the analogue input/output module uses the first group to determine the range of the input signal. Furthermore, the service module reports any occurring wire failure to this range.
The address range depends on the slot:

Slot	Transfer address range
0	SLA00.00...01.15
1	SLC00.00...01.15
2	SLE00.00...01.15
3	SLG00.00...01.15
4	SLI00.00...01.15
5	SLK00.00...01.15
6	SLM00.00...01.15
7	SLO00.00...01.15

Assignment of transfer addresses

Address	Symbol	Significance			
		Value	Signal	Chan.	
SLx00.00	I_SIG_0	=0 <>0	0...20 mA 4...20 mA	0	User preselects the type of signal of the inputs (default = 0)
SLx00.01	I_SIG_1			1	
SLx00.02	I_SIG_2			2	
SLx00.03	I_SIG_3			3	
SLx00.08	O_SIG_0	=0 <>0	0...20 mA 4...20 mA	0	User preselects the type of signal of the outputs (default = 0)
SLx00.09	O_SIG_1			1	
SLx00.10	O_SIG_2			2	
SLx00.11	O_SIG_3			3	
SLx01.00	FAIL_0	=255	Wire failure	0	Error message from the service module, only for inputs with 4...20 mA signal
SLx01.01	FAIL_1	=255	Wire failure	1	
SLx01.02	FAIL_2	=255	Wire failure	2	
SLx01.03	FAIL_3	=255	Wire failure	3	

5.4.4.5. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature:	-25...+70 °C
ambient temp. during operation	0...55 °C
relative humidity	50...95 %
Inputs	2
Potential separation	no
Addressing:	AIxx.00...AIxx.01
Measuring range:	0(4)...20 mA
Internal resistance:	~ 40 Ω
Resolution:	10bit, ~ 0.02 mA / digit
Precision (relative to final value)	
max. error (at 25 °C)	± 0.8 %
norm. error (at 25 °C)	± 0.4 %
temperature coefficient	0.01 % /K
linearity error	± 0.1 %
Conversion time:	10 ms
Input resistance:	~ 40 Ω
Protection against noise pulses:	by filters and buffers
Indicated failures:	wire failure
Outputs	2, short-circuit protected
Potential separation	no
Addressing:	AOyy.00...AOyy.01
Range:	0(4)...20 mA
Resolution:	11bit, ~ 0.01 mA / digit for 0...20 mA ~ 0.008 mA / digit for 4...20 mA
Transient building-up period:	norm. 0.02 ms
Burden:	600 Ω max.
No-load voltage:	15 V max.
Precision (relative to final value)	
max. error (at 25 °C)	± 0.6 %
norm. error (at 25 °C)	± 0.2 %
temperature coefficient	0.02 % /K
linearity error	± 0.1 %
Sign:	1bit
Weight:	c. 90 g
Part number:	680.441.09

6. Counter modules

Counter modules are used when the defined funtions of the software counters in the program are not sufficient or when the requested counting frequency cannot be reached because of the cycle time.

There are various counter modules available.

- multi-function counter modules with 1 or 2 channels and 24V inputs, counting frequency up to 25 kHz

- multi-function counter modules with 2 channels and RS422 interface, counting frequency up to 100 kHz

- event counter modules with 2 channels and 24V inputs, counting frequency up to 25 kHz

6.1. Counter module, 1 or 2 multi-function counters, 24bit

There are modules with one or two counters available each of which have their own bidirectional counter. The input wiring is set so as to allowing registering frequencies of up to 25 kHz.

6.1.1. Functions

Bidirectional counters register signals from incremental encoders. The count level can be preset and can be compared to a preset value. Use the software to set the counter to one of two possible operating modes:

A-B-Ref counter

The counting direction is recognized by two counting inputs (tracks A and B) which are set in a 90° opposition to each other. The reference point is determined by a reference input (ref).

Event counter

Furthermore, the module can also be used for event counting (connection of a simple pulse generator). In this case, input A is used as pulse input and input B determines the counting direction (0 = up, 1 = down). The reference input (ref) has no function.

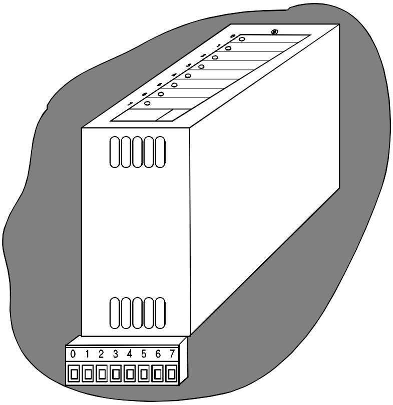
6.1.2. Slots

The module can be plugged into any slot of the controller.

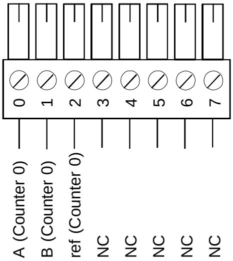
6.1.3. Counter modules with 24V inputs

6.1.3.1. Connectors

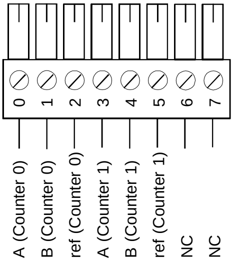
The connectors of the signal lines are located on the 8-pin plug-screw connector underneath the module:



Module with 1 counter (680.454.01):



Module with 2 counters (680.454.02):



Generator supply

The generator should be supplied via the same supply that also supplies the controller with 24 V DC. If the device is supplied externally, a potential equalization between the 0V potentials must be installed in any case.

⚠ *The module may recognize reference pulses in case of a voltage drop. This might reset the count level.*

Shielding

Shielded cables have to be used for connecting the input signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

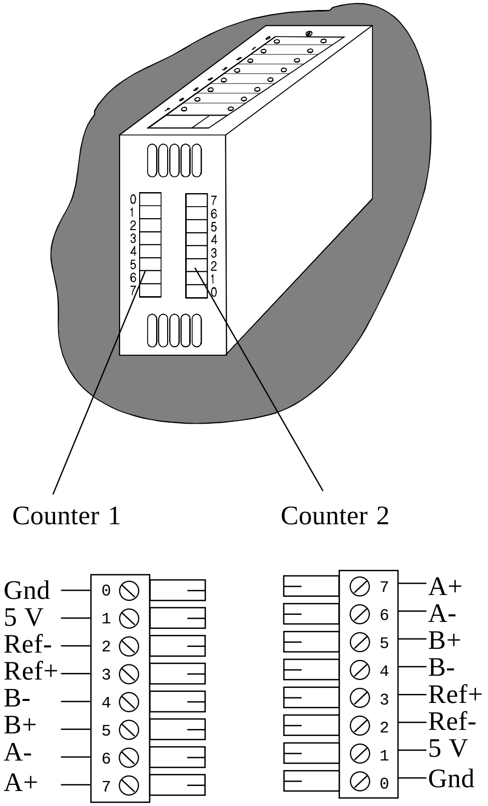
6.1.3.2. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Number of counters / module:	1 or 2
Counting depth:	24bit
Functions:	- A-B-Ref counter - event counter
Inputs:	A, B and ref
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	LEDs
colour	A, B: yellow ref: red
tapping point	in the input circuit
signal states	1: LED on 0: LED off
Input voltage:	24 V DC +25 % -20 % (incl. residual ripple)
Surge immunity	≤ 60 V DC (≤ 30 min.)
Signal identification	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Power consumption / input:	max. 10 mA
Clock pulse frequency:	max. 25 kHz
Weight	
module with 1 counter	78 g
module with 2 counters	102 g
Part numbers	
module with 1 counter:	680.454.01
module with 2 counters:	680.454.02

6.1.4. Counter module with RS422 interface

6.1.4.1. Connectors

The connectors for the signal lines are located at the bottom narrow module edge:



Enclosure to instruction manual E 326 GB, Page 6-6:
Correction of connector description for module 680.454.08
Kuhnke GmbH, 20 October1997

Generator supply

The generator should be supplied with 5V via the module. If the device is supplied externally, a potential equalisation between the 0V potentials must be installed in any case.

The module may recognize reference pulses in case of a voltage drop. This might reset the count level.

Shielding

Shielded cables have to be used for connecting the input signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

6.1.4.2. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Number of counters / module:	1 or 2
Counting depth:	24bit
Functions:	- A-B-Ref counter - event counter
<u>Inputs</u>:	A, B and ref
Type	RS422
Potential separation	no
Indicators	LEDs
colour	A, B: yellow ref: red
tapping point	in the input circuit
signal states	1: LED on 0: LED off
Input voltage:	RS422 level
Clock pulse frequency:	max. 100 kHz
Weight:	c. 112 g
Part number	680.454.08

6.1.5. Programming

A service module (see "3.4. Service modules") realises the counting functions. The mode of operation of the counter module is determined by programming transfer addresses and interrupt modules which are assigned to the slot of the module.

6.1.5.1. Transfer address ranges and interrupt modules

The assignment of the transfer address ranges and of the interrupt modules to the slots is shown in the table in chapter "3.4. Service modules".
Counter 0 is served via the first transfer address range and the first interrupt module. Counter 1 (only exists on modules with 2 counters) uses the second transfer address range and the second interrupt module.

Example:

Counter module with 2 counters in slot 3

Counter	Transfer address range	Interrupt module
0	SLG00.00...01.15	7
1	SLH00.00...01.15	8

Assignment of transfer addresses

Address	Symbol	Comment	
SLx00.00	AV_LB	actual value	lowbyte
SLx00.01	AV_MB		middlebyte
SLx00.02	AV_HB		highbyte
SLx00.03			
SLx00.04	PV_LB	preset value	lowbyte
SLx00.05	PV_MB		middlebyte
SLx00.06	PV_HB		highbyte
SLx00.07			
SLx00.08	RV_LB	reference value	lowbyte
SLx00.09	RV_MB		middlebyte
SLx00.10	RV_HB		highbyte
SLx00.11			
SLx00.12			
SLx00.13			
SLx00.14			
SLx00.15			
SLx01.00	CTRL_IRQ	counter control	<>0: release IRQ (interr. "reference value reached")
SLx01.01	CTRL_REF		<>0: clear counter when "reference input activated"
SLx01.02	CTRL_RV		<>0: clear counter when "reference value reached"
SLx01.03	CTRL_CNT		counter control (<>0: counter on, =0: counter off)
SLx01.04	EVT_ENC		function selection (<>0: event, =0: A-B-Ref counter)
SLx01.05	CTRL_1_4		function selection (<>0: simple count, =0: 4fold coun
SLx01.06			
SLx01.07			
SLx01.08	DISP_REF	displays	reference signal "ref" is activated
SLx01.09	DISP_PV		reference value reached (reset by setting SET_RV)
SLx01.10			
SLx01.11	SET_RV	control flags	<>0: transfer reference value (SLx00.08...10) to count
SLx01.12	SET_PV		set counter to preset value (SLx00.04...06)
SLx01.13	ACT_CTRL		activate commands of counter control (SLx01.00...05)
SLx01.14			
SLx01.15	VERSION		software version

6.1.5.2. Control flags

The addresses SLx01.11 (SET_RV), SLx01.12 (SET_PV) and SLx01.13 (ACT_CTRL) work as control flags which request the counter (when the signal changes from "0" to "<>0") to take over the corresponding settings (see below).
After the counter has taken over the settings it resets the flags (0). Pulses are therefore sufficient for controlling the counter.

6.1.5.3. Setting the counter to the preset value / to 0

Control flag SLx01.12 (SET_PV) effects a taking over of the preset value in SLx00.04...06(PV_xB) into the counting level. Preset value = 0 means: clear counter.

6.1.5.4. Setting the reference value

Control flag SLx01.11 (SET_RV) effects a taking over of the reference value in SLx00.08...10(RV_xB) into the counter reference register. At the same time, the counter clears the display SLx01.09 (DISP_PV).

☞ *The reference value can also be used to scan the counting level for "0".*

6.1.5.5. Counter control

Control flag SLx01.13 (ACT_CTRL) effects an activation of the counter control in SLx01.00...05:

SLx01.00 (CTRL_IRQ) allows or disallows a interrupt request. An interrupt calls the corresponding interrupt module (see next page).

SLx01.01 (CTRL_REF) activates or deactivates the reference input. If it is activated, the counter level is set to 0 by the "ref" input.

SLx01.02 (CTRL_RV) activates or deactivates the clearing of the counter level upon reaching the reference value. This can be used to create variable ring counters.

SLx01.03 (CTRL_CNT) switches the counter on or off without clearing the counter level..

SLx01.04 (EVT_ENC) switches between counter functions (event counter with 1 input or A-B-Ref counter with 3 inputs).

SLx01.05 (CTRL_1_4) switches between the counting functions of the A-B-Ref counter: simple count (rising edge of A) or fourfold count (both edges of A and B).

6.1.5.6. Reference value

Upon reaching the reference value in SLx00.08...10 (RV_xB) the following happens:

- the display SLx01.09 (DISP_PV) is set to 255 by the counter. SLx01.09 remains set until a new reference value is transferred to the counter (see above).
- If SLx01.00 (CTRL_IRQ) is <> 0 and the value has been taken over by the counter already, an interrupt is triggered (see next page). SLx01.09 (DISP_PV) is then automatically reset in the corresponding interrupt module.

6.1.5.7. Interrupt

If an interrupt is generated by reaching the reference value in SLx00.08...10 (RV_xB), the corresponding interrupt module is called automatically (see "6.1.5.1. Transfer address ranges and interrupt modules").
The following activities are then carried out automatically:

Actions before entering the interrupt module:

- The actual counter value in SLx00.00...02 (AV_xB) is updated.

Actions within the interrupt module:

- The user program evaluates the current counter level.

⚠ *Count pulses are not recognized by the counter as long as the interrupt module is activated. They are lost.*

The user program in the interrupt module should therefore be as short as possible.

Actions immediately before leaving the interrupt module:

- Display SLx01.09 (DISP_PV) is set =0.
- The current reference value in SLx00.08...10 (RV_xB) is taken over independent of the control flag SLx01.04 (EVT_ENC).
- The preset value in SLx00.04...06 (PV_xB) is taken over, if control flag SLx01.12 (SET_PV) is <>0.

⚠ *A new preset value should only be taken over if the counter is not counting (if the axle stands still). Counting pulses could otherwise get lost.*

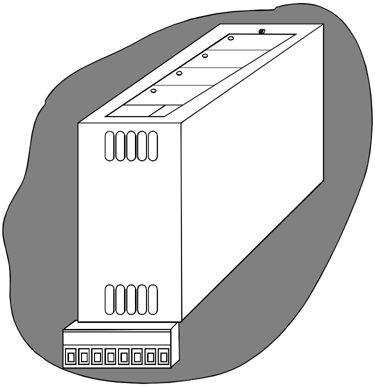
- The current settings for counter control are taken over, if control flag SLx01.13 (ACT_CTRL) is <>0.

Like this the interrupt module can be used for programming new counter settings which will then become valid immediately.

6.2. Counter module, 2 event counters, 16bit

Function

Event counter for connecting a simple pulse generator. The counting direction can be preset by the program.
The input wiring is set in a way that allows registering frequencies of up to 25 kHz.



6.2.1. Slots

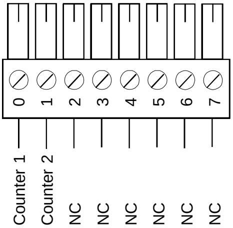
The module uses the counters on the processor of the CPU. For this purpose, a specified slot carries additional leads. The module can only be operated on this slot (see "3.3. Slots"):

Slot no.	Useable channels
1	0, 1

⚠ *You may not use this module on a KUAX 680C. This device is already equipped with two event counters that are also controlled by the processor.*

6.2.2. Connectors

The connections of the signal lines for the counter inputs (generator) are on the 8-pin plug-screw connector underneath the module (see diagram on previous page):



Connector	*Function*
Counter 1	pulse input for counter 1
Counter 2	pulse input for counter 2
NC	unused

Generator supply

The generator supply (24 V DC) can be tapped from the two supply strips underneath the signal strip (see Instruction manual KUAX 680I, E 308GB, "System supply and feeding of the supply strips"). If an external supply is used, a potential equalization between the 0V potentials must be installed in any case.

Shielding

Shielded cables have to be used for connecting the analog signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

6.2.3. Technical specifications

Application:	KUAX 680I, *not KUAX 680C or KDT 680CT*
slot	1
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Number of counters / module:	2
Counting depth:	16 bit
Function:	event counter
Inputs:	2
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	LEDs
colour	green
tapping point	in the input circuit
signal states	1: LED on 0: LED off
Input voltage:	24 V DC +25 % -20 % (incl. residual ripple)
Surge immunity	≤ 60 V DC (≤ 30 min.)
Signal identification	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Power consumption / input:	max. 10 mA
Clock pulse frequency:	max. 25 kHz
Weight:	c. 75 g
Part number:	680.454.03

6.2.4. Programming

A service module (see "3.4. Service modules") realises the counter functions. The required mode of operation is determined by programming transfer addresses and interrupt modules which are assigned to the slot of the module.

6.2.4.1. Transfer address ranges

This module can only be operated in slot 1. The first transfer address range is for counter 0, the second one for counter 1:

Slot	Transfer address range	Counter no.
1	SLC00.00...01.15	1
	SLD00.00...01.15	2

This counter module cannot release interrupts. It can therefore neither call interrupt modules.

Assignment of transfer addresses

Address	Symbol	Comment	
SLx00.00	AV_LB	actual value	lowbyte
SLx00.01	AV_HB		highbyte
SLx00.02			
SLx00.03			
SLx00.04	PV_LB	preset value	lowbyte
SLx00.05	PV_HB		highbyte
SLx00.06			
SLx00.07			
SLx00.08			
SLx00.09			
SLx00.10			
SLx00.11			
SLx00.12			
SLx00.13			
SLx00.14			
SLx00.15			
SLx01.00	RUN	counter control	counter enable/disable (<>0: ON, 0: OFF)
SLx01.01	UPDWN		counting direction (<>0: count up, =0: count down)
SLx01.02	MODE		counting mode (<>0: count positive and negative edges, =0: only count positive edges)
SLx01.03	CLR		<>0: clear count (actual value)
SLx01.04	SET_PV		<>0: set counter to preset value
SLx01.05			
SLx01.06			
SLx01.07			
SLx01.08			
SLx01.09			
SLx01.10			
SLx01.11			
SLx01.12			
SLx01.13			
SLx01.14			
SLx01.15			

6.2.4.2. Setting the counter to the preset value / to 0

Control flag SLx01.04 (PV_LB) is used for setting the counter:

SLx01.04 =0: no function
<>0: take over preset value in SLx00.04...05 (PV_xB)

6.2.4.3. Switching the counter on / off

Control flag SLx01.00 (RUN) is used for switching the counter on or off:

SLx01.00 =0: counter off
<>0: counter on

6.2.4.4. Selecting the counting direction

Control flag SLx01.01 (UPDWN) determines the counting direction:

SLx01.01 =0: count down
<>0: count up

6.2.4.5. Selecting the counting mode

Control flag SLx01.02 (MODE) determines the counting mode:

SLx01.01 =0: count only positive edges
<>0: count positive and negative edges

6.2.4.6. Clearing the count

Control flag SLx01.03 (CLR) clears the count:

SLx01.01 =0: no function
<>0: clear count

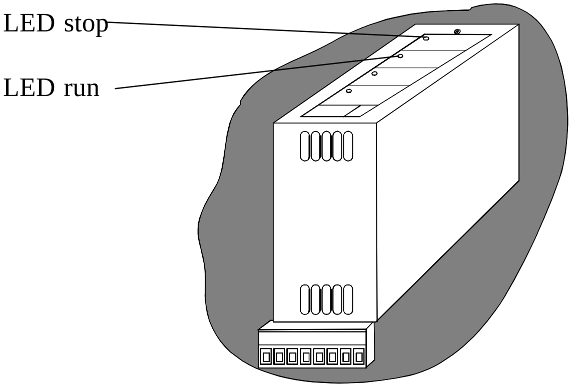
6.2.4.7. Evaluating the count

The evaluation of the count (actual value) in SLx00.00...01 (AV_xB) is done in the user program. The counter module itself does not provide any evaluation signals.

6.3. SSI module, 24 bit, for 2 absolute value devices

Function

SSI module for the connection of two absolute value devices, also referred to as absolute angle encoders. You can set the clock pulse frequency for the encoder via the program. Physically speaking, SSI interfaces are in accordance with the specifications set for RS 422 interfaces.



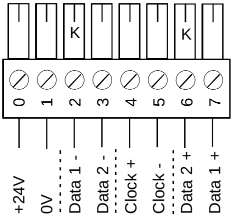
6.3.1. Slots

The module can be plugged into any slot of the controller. The number of modules per controller is limited due to the power consumption of each module (see appendix A):

Controller	*Max. number of SSI modules*
KUAX 680I	
- 680.420.xx	1
- 680.423.xx	3
KUAX 680C	3
KDT 680CT	3

6.3.2. Connectors

The connectors of the signal lines for the absolute angle devices are on the 8-pin plug-screw connector underneath the module (see illustration on previous page):



K = coding element

Connector	*Function*
+24V	power supply of the absolute value devices
0V	"
Data 1 +/-	data lines encoder #1
Data 2 +/-	data lines encoder #2
Clock +/-	clock pulse for both encoders

Generator supply

The supply (24 V DC) for the absolute value devices is tapped from terminals 0 and 1. The voltage is taken from the supply of the outputs and must therefore be connected there (see "A.1. Power supply").

Coding

Please code the screw-type locking connector as suggested (see "K" in the illustration above). Refer to chapter "3.2.1. Coding" to learn how this can be done.

This coding prevents the possibility of confusing connectors that might cause destruction of components because 24 V DC are supplied via this connector.

Shielding

Shielded cables have to be used for connecting the analog signals. The shielding is connected to the aluminium baseprofile of the controller by M3 screws (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

6.3.3. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
slot	all slots
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Number of counters / module:	2
Counting depth:	24 bit
Function:	registration of position values from absolute value devices
Inputs:	2
Type (acc. to IEC 1131)	1
Potential separation	no
Indicators	2 LEDs
	colour function
	green OK
	red failure
Input voltage:	24 V DC -20 %/+25 % (incl. residual ripple)
Signal identification	acc. to specification for RS 422
Clock pulse frequency:	user-definable from 95 to 420 kHz
Weight:	c. 123 g
Part number:	680.454.04

6.3.4. Programming

A service module (see "3.4. Service modules") realizes the counter functions. The required mode of operation is determined by programming transfer addresses and interrupt modules which are assigned to the slot of the module.

6.3.4.1. Transfer address ranges

Each module is assigned a transfer address range of 32 byte (see also "3.4. Service modules"). The first group of these is used by teh analog output module to set the range of the output signal.
The address range depends on the slot used:

Slot	Transfer address range
0	SLA00.00...01.15
1	SLC00.00...01.15
2	SLE00.00...01.15
3	SLG00.00...01.15
4	SLI00.00...01.15
5	SLK00.00...01.15
6	SLM00.00...01.15
7	SLO00.00...01.15

This counter module cannot release interrupts. It can therefore neither call up interrupt modules.

Assignment of transfer addresses

Address	Symbol	Comment	
SLx00.00	SSI1_LB	actual value encoder #1	low byte
SLx00.01	SSI1_MB		middle byte
SLx00.02	SSI1_HB		high byte
SLx00.03		unused	
SLx00.04	SSI2_LB	actual value encoder #2	low byte
SLx00.05	SSI2_MB		middle byte
SLx00.06	SSI2_HB		high byte
SLx00.07		unused	
SLx00.08	ENC_SOL1	resolution of encoder #1 24 bit max.(default = 24 bit), as from version 3.	
SLx00.09	ENC_SOL2	resolution of encoder #2 24 bit max.(default = 24 bit), as from version 3.	
SLx00.11	SSI_CODE	0 = Gray code (default), 255 = binary code	
SLx00.12	SSI_FRQ	settings of the clock pulse frequency in [kHz]: 2 = 420 3 = 310 4 = 250 5 = 210 6 = 185 -> default at system startup 7 = 155 8 = 140 9 = 125 10 = 115 11 = 105 12 = 95	
SLx00.13	SSI_SET	255 = import settings from SLx00.11 and .12	
SLx00.14		unused	
SLx00.15	SSI_FAIL	error and failure messages (bit set = fault): bit 0: encoder #1 not connected bit 1: endocer #2 not connected bit 2: unable to read data bit 3: failure of SSI module	
SLx01.00 to SLx01.14		unused	
SLx01.15	SSI_VERS	software version (as from version 2)	

6.3.4.2. User program

The user program directly accesses the transfer addresses.

Actual values

The actual encoder values are written into three addresses each. The user program reads these data directly out of the transfer addresses.

☞ *Read the contents of address SLx00.15 before reading in the actual values. The actual values are only valid if the value in address SLx00.15 is 0 (no fault).*

Code

You can use the module with absolute value devices working with gray code or binary code. The code is set via address SLx00.12. At startup, the system defaults to Gray code.

Clock pulse frequency

The default clock pulse frequency is 185 kHz. If you need to work with a different frequency, then input the corresponding value (2...12) into address SLx00.12.

⚠ *Higher clock pulse frequencies make the system more responsive to noise.*

To change the system defaults

Proceed as follows to change the default code or clock pulse frequency settings:

- write the desired value into SLx00.11 (to change the code) or SLx00.12 (to change the clock pulse frequency)
- input a value <>0 into SLx00.13
- the module accepts the new settings and writes 0 into SLx00.13

⚠ *If default settings are changed, updating of actual values will be skipped for one cycle.*

Error and failure messages

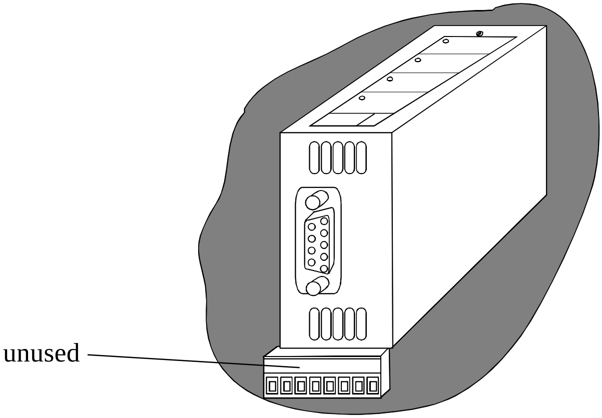
In the case of errors or failures, the system inputs corresponding coded messages into SLx00.15 from where they can be imported into the user program for analysis.

7. Communication modules

The controllers are equipped with at least one serial interface when they come to you. Additional interfaces are sometimes needed, however. In these cases you apply communication modules.

7.1. V.24 (RS 232) module

This module has its own V.24 interface (RS 232) and is therefore independent of the ACIA in the processor on the CPU board. It can be applied to every KUAX 680I or 680C. It serves data communication only and cannot be used for programming the controller.



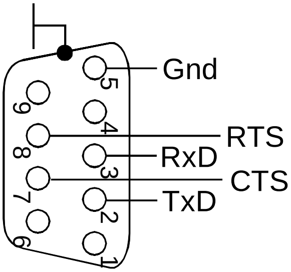
7.1.1. Slots

The module can be plugged into any slot.

7.1.2. Connector

There is a female 9pin Sub-D connector on the lower narrow side of the module.

Pin assignment:



Inter-connections

- cross over the connections of data lines TxD and RxD of both communication partners;
- connect Gnd to the Gnd connector of the communication partner;
- bridge the CTS and RTS connectors if you are using communication without handshake;
- if you are using a handshake, then cross over the CTS and RTS connectors when connecting them to the CTS and RTS connectors of the communication partner.

The signal strip underneath the module remains unused.

Shielding

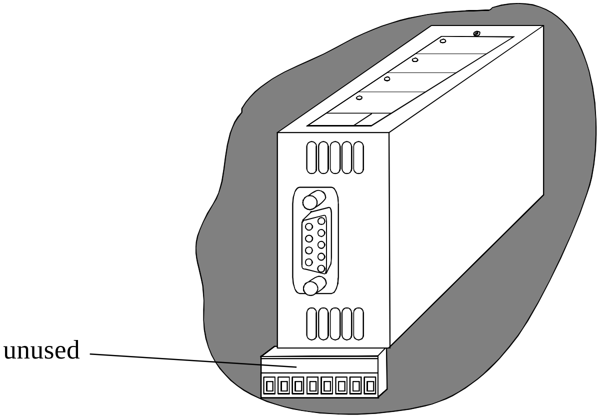
A shielded cable has to be used for connecting the V.24. The shielding is connected to the cable plug which is connected to the module in the controller (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

7.1.3. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
slots:	all slots
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Type:	V.24 interface (RS232)
Function:	data communication, point-to-point
Transfer rate:	600...19200 baud (max. per controller) adjustable via software function *If more than 1 module is used at the same time the max. baud rate per module is 19200 devided by the number of modules.*
Number of interfaces:	1 per module
Connector:	female 9pin Sub-D connector
Hardware handshake:	CTS and RTS
Weight:	c. 89 g
Part number:	680.440.01

7.2. TTY module (20 mA)

This module has its own TTY interface (20 mA current loop) and is therefore independent of the ACIA in the processor on the CPU board. The TTY is a passive interface, i.e. the power supply must be provided externally, e.g. by the communications partner.
The module can be applied to every KUAX 680I or 680C. It serves data communication only and cannot be used for programming the controller.



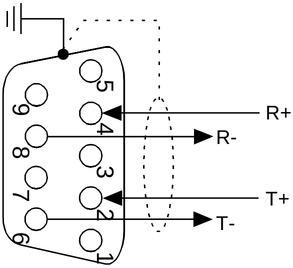
7.2.1. Slots

The module can be plugged into every slot.

7.2.2. Connector

There is a female 9pin Sub-D connector in the lower narrow side of the module.

Pin assignment:



Connection	*Function*
R+, R-	Receive channel
T+, T-	Transmit channel

The signal strip underneath the module remains unused.

Shielding

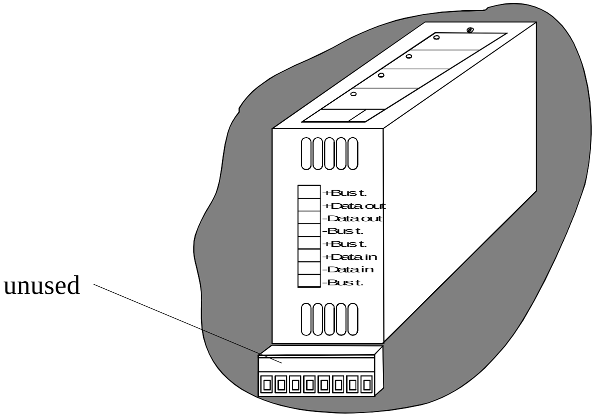
We recommend using shielded cables if cables of greater length have to be applied. The shielding is connected to the casing of the cable plug which is connected to the module in the controller (under certain circumstances an additional grounding of the other end of the cable may be commendable).

7.2.3. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
slots:	all slots
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Type:	passive TTY interface (20 mA current loop)
Function:	data communication
Transfer rate:	600...9600 baud adjustable via software function
Number of interfaces:	1 per module
Connector:	female 9pin Sub-D connector
Weight:	c. 86 g
Part number:	680.440.02

7.3. RS 485 module

This module has its own RS 485 interface and is therefore independent of the ACIA in the processor on the CPU board. The module can be applied to every KUAX 680I or 680C. It serves data communication only and cannot be used for programming the controller.

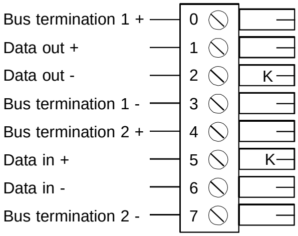


7.3.1. Slots

The module can be plugged into every slot.

7.3.2. Connector

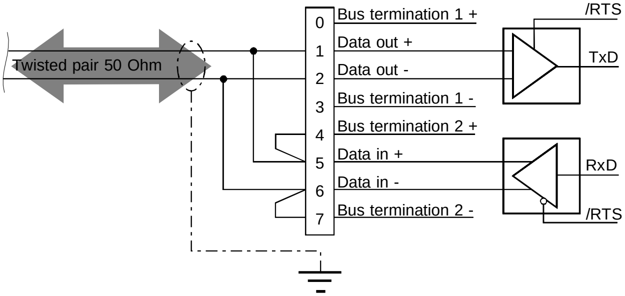
There is an 8pin screw-type locking connector in the lower narrow side of the module.



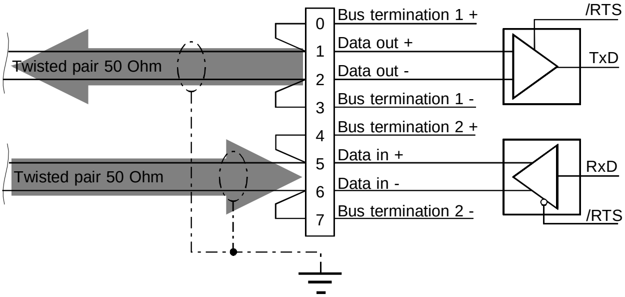
K = remove coding element

Typical connections:

Half duplex, with bus termination:



Full duplex, with bus termination:



Typical connections

- Point-to-point: the module is physically connected to only one communication partner.
- Bus: the module is physically connected to two communication partners. The data lines of the two partners are connected in parallel to the module.

Bus termination

The module is equipped with bus termination resistors in the factory. All you have to do is to activate them at the terminating partner station by jumper wires on the connector (see illustration, jumpers 4-5, 6-7 and 0-1, 2-3). The module is defined as a terminating partner station if it is physically connected to only one other communication partner.
As soon as the module is physically connected to two communication partners in a bus system, it is no longer a terminating partner station and the connector must be without jumpers.

Shielding

A shielded cable has to be used for connecting the RS 485. The shielding is connected to the casing of the cable plug which is connected to the module in the controller (under certain circumstances, the additional grounding of the other end of the cable may be commendable).

7.3.3. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
slots:	all slots
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation:	0...55 °C
relative humidity	50...95 %
Type:	RS 485 interface without potential separation
Function:	data communication - point-to-point connection or - bus connection
Transfer rate:	600...19200 baud adjustable via software
Number of interfaces:	1 per module
Connector:	8pin screw-type locking connector
Weight:	c. 89 g
Part number:	680.440.03

7.4. Programming the V.24, TTY and RS 485 modules

The modules are suitable for simple and more complex tasks. Their programming is accordingly regulated. There are different way of programming the modules.
The three modules only differ in their hardware. The software is the same. All information given on the following pages is therefore always applicable to all three modules.

7.4.1. Communication programs SE 680I, KUSI680, RS485

We recommend using more complex communication programs for more complex tasks (sending/receiving larger amounts of data, interrupt-controlled communication...):

- SE_680I	for the V.24 and TTY modules
- KUSI680	for the V24 and TTY modules, protocol 3964(R)
- RS 485	for the RS 485 module

These programs can be easily embedded into the user program in the shape of KUBES modules. They provide a wide range of functions.
The programs are described in a separate instruction manual:

☞

Instruction manual
SE_680I, KUSI680, RS 485 - Software for communication modules 680
E 334 D

Ordering specification .... SE_680I, KUSI680, RS 485
Data communication programs
Part number 680.505.01

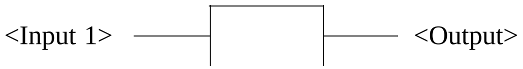
You can skip the information on the following pages if you are working with these programs.

7.4.2. KUBES modules V24XE, V24XS, V24XSTRG

The three KUBES modules delivered with KUBES will suffice for simple communication tasks:

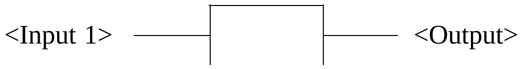
KUBES module V24XE

Function: Receives individual characters or strings



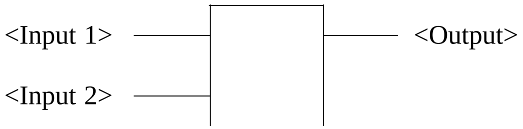
KUBES module V24XS

Function: Sends out individual characters



KUBES module V24XSTRG

Function: Sends out strings



The parameters are explained on the next page. The bytes listed in the tables can also contain different addresses (see "Example") as long as they form consistent blocks.

7.4.2.1. Parameters of the KUBES modules

<Input 1>

The first input parameter is the same for all three KUBES modules. It consists of a coherent, user-defined data field of byte operands:

Byte			Explanation
No.	Example	Symbol	
1	BM00.00	BAUD	user sets the baudrate (transfer rate) (see table "To set the baudrate" on next page)
2	BM00.01		
3	BM00.02	FORMAT	data format (see table "To set the data format" on page after next)
4	BM00.03	INITV24	<>0: take over BAUD and FORMAT
5	BM00.04	SLOT	module slot (0...3/7)
6	BM00.05	RES_1	unused
7	BM00.06	REC_CHR	received character
8	BM00.07	REC_RUN	<>0: character has been received; user program clears after evaluation
9	BM00.08	SND_CHR	character to be sent
10	BM00.09	SND_RUN	user program sets <>0: sending confirmed; V.24 clears when done

<Output>

The output parameter is used for status messages:

Byte			Explanation
No.	Example	Symbol	
1	BM00.10	STATUS	status message KUBES module: =0: ok, =4: slot address too big

<Input 2>

This input parameter only exists in KUBES module V24XSTRG. Its length is the number of characters to be transferred + 1:

Byte			Explanation
No.	Example	Symbol	
1	BM01.00	NMB_CHR	number of characters to be sent; input by the user
2	BM01.01	S_CHAR1	first character to be sent
3	BM01.02		second character to be sent
4	BM01.03		third character to be sent
5	BM01.04		etc.

To set the baudrate

Input the transfer rate into address BAUD (example: BM00.00...01). The corresponding values are show in the following table:

Baudrate	Value to be set	Module		
		V.24	RS 485	TTY
300	384	yes	yes	yes
600	192			
1200	96			
1800	64			
2000	58			
2400	48			
3600	32			
4800	24			
7200	16			
9600	12			
19200	6			no

Example 9600 Baud:

in the initialization module

```
BM00.00      WORD        12
```

or in the program module

```
LD      12
=D      BM00.00
```

⚠ *The maximum baudrate for the TTY module is 9600 baud.*

To set the data format

Input the data format into address FORMAT (example: BM00.02). The corresponding values are shown in the following table:

Bit								Function
7	6	5	4	3	2	1	0	
0	0	0				0	0	5 bits
						0	1	6 bits
						1	0	7 bits
						1	1	8 bits
					0	x	x	1 stop bit
					1	0	0	1.5 stop bits
					1	0	1	2 stop bits
					1	1	0	"
					1	1	1	"
				1	x	x	x	parity
				0				no parity
			1	1				even parity
			0					odd parity

Example 8 bits, 1 stop bit, no parity:

in the initialization module

```
        BM00.02      BYTE      %00000011
```

or in the program module

```
        L        %0000011
        =        BM00.02
```

7.4.2.2. Example program

The example is structured as follows:

- The ORG(anization) module contains the program decision for whether you want to send individual characters or strings. Depending on the result, the program branches to the corresponding program modules INDIV.PRO or STRINGS.PRO.
- The V.24 interface is initialized in the initialization module INI_V24.INI which is called up once at the start of the program by the ORG module.
- The text string to be sent is defined in the initialization module INI_STRG.

```
========  Kubes  =====================================================
                        Project structure
Project : V24_EXT
                                     created :   Jul 20 1993 15:12
User    : Gerd Hildebrandt           altered :   Jul 21 1993 14:33
Comment : Communication via the V.24 module

======================================================================
ORG.ORG/1
|
*——>INDIV.PRO/1
|       |
|       *——>V24XE.KUN/7
|       |
|       *——>V24XS.KUN/13
|
*——>STRINGS.PRO/2
|       |
|       *——>V24XE.KUN/7
|       |
|       *——>V24XSTRG.KUN/14
|       |
|       *——>INI_STRG.INI/2
|
*——>INI_V24.INI/1
                                              Page : 1
```

```
========  Kubes  ======================================================
                        Symbol table
Project : V24_EXT
                                     created :   Jul 20 1993 15:12
User    : Gerd Hildebrandt           altered :   Jul 21 1993 14:33
Comment : Communication via the V.24 module

======================================================================
Address:    Symbol:     Comment:                            Supplement:
I00.00      SEND        send data
I00.07      IC_STRG     send individual character or string

Address:    Symbol:     Commment:                           Supplement:
BM00.00     BAUD        Baudrate
BM00.01     BM00_01     Baudrate
BM00.02     FORMAT      data format
BM00.03     INITV24     take over data format
BM00.04     SLOT        module slot (0...7)
BM00.05     RES_1
BM00.06     REC_CHR     received character
BM00.07     REC_RUN     received: V.24 message
BM00.08     SND_CHR     character to be sent
BM00.09     SND_RUN     send: message to V.24
BM00.10     STATUS      status of KUBES module

Address:    Symbol:     Comment:                            Supplement:
BM01.00     NMB_CHR
BM01.01     S_CHR1
Address:    Symbol:     Comment:                            Supplement:
PP00.00     P_SEND      pulse data send
                                                        Page : 1
```

```
========  Kubes  =====================================================
                       Organization module IL
Project:  V24_EXT
Module  : ORG            No.: 1     created :   Jul 20 1993 15:12
User    : Gerd Hildebrandt          altered :   Jul 21 1993 14:28

======================================================================
  1: ; initialize interface
  2: ; ——————————————
  3:
  4: V24INIT  L       M00.00
  5:          JPC     START
  6:          JPINIT  INI_V24              1
  7:          =1      M00.00
  8:
  9: START    NOP
 10:
 11: ; individual characters or strings: select
 12:
 13:          L       IC_STRG    I00.07 ;(send  indiv.char./str.)
 14: IC       JPCP    INDIV       1     ;individual  characters
 15:
 16: STR      LN      IC_STRG I00.07    ;(send  indiv.char./str.)
 17:          JPCP    STRINGS      2    ;strings
 18:
 19:
 20:

                                                 Page : 1
```

```
========  Kubes  =====================================================
                          Program module IL
Project:  V24_EXT
Module  : INDIV       No.: 1     created :   Jul 21 1993 09:32
User    : Gerd Hildebrandt       altered :   Jul 21 1993 14:28
Comment:  INDIVIDUAL

======================================================================
; receive individual characters
; ———————————

; KUBES module
          JPK  V24XE    , _____
               BAUD     -|_____|- STATUS

; evaluate received data
          L       REC_RUN     BM00.07 ;(received: V.24 message)
          JPCN NO_REC
          L       REC_CHR     BM00.06 ; (received character)

; ...
; the received character can be evaluated here
; ...

          CLR  REC_RUN    BM00.07 ;(received: V.24 message)
NO_REC    NOP



; send individual character
; ——————————

; enter character to be sent
          L       ‘K’
          =       SND_CHR     BM00.08 ;(character to be sent)

; send: confirm
          L       SEND        I00.00  ;(send data)
          =       P_SEND      PP00.00 ;(pulse data send)
          L       P_SEND      PP00.00 ;(pulse data send)
          AN      SND_RUN     BM00.09 ;(send: conf. to V.24)
          JPCN    NO_SND
          L       255
          =       SND_RUN     BM00.09 ;(send: conf. to V.24)
NO_SND    NOP

; KUBES module
          JPK     V24XS    ,   _____
                    BAUD     -|_____|- STATUS

```

```
========  Kubes  ================================================
                          Program module IL
Project:  V24_EXT
Module  : STRINGS      No.: 2        created :   Jul 21 1993 09:32
User    : Gerd Hildebrandt           altered :   Jul 21 1993 14:28
Comment:  STRINGS

=========================================================================
  1: ; receive individual characters
  2: ; ———————————
  3:
  4: ; KUBES module
  5:          JPK   V24XE    ,    _____
  5:                  BAUD     -|_____|- STATUS
  6:
  7: ; evaluate received data
  8:          L     REC_RUN       BM00.07 ;(received:V.24 message)
  9:          JPCN  NO_REC
 10:          L     REC_CHR       BM00.06 ;(received character)
 11:
 12: ; ...
 13: ; the received character can be evaluated here
 14: ; ...
 15:
 16:          CLR   REC_RUN       BM00.07 ;(received: V.24 message)
 17: NO_REC   NOP
 18:
 19:
 20:
 21: ; send string
 22: ; ——————
 23:
 24: ; enter character to be sent
 25:          JPINIT  INI_STRG          2
 26:
 27: ; send: confirm
 28:          L       SEND         I00.00 ;(send data)
 29:          =       P_SEND      PP00.00 ;(pulse data send)
 30:          L       P_SEND      PP00.00 ;(pulse data send)
 31:          AN      SND_RUN     BM00.09 ;(send: conf. to V.24)
 32:          JPCN    NO_SND
 33:          L       255
 34:          =       SND_RUN     BM00.09 ;(send: conf. to V.24)
 35: NO_SND   NOP
 36:
 37: ; KUBES module
 38:          JPK     V24XSTRG ,   _____
 38:                    BAUD     -|     |- STATUS,
 38:                    NMB_CHR  -|_____|-
 39:
```

```
========  Kubes  ====================================================
                        Init. module IL
Project:  V24_EXT
Module  : INI_STRG  No.: 2       created :    Jul 21 1993 11:36
User    : Gerd Hildebrandt       altered :    Jul 21 1993 11:36
Comment:  INI_STRG

=====================================================================
  1: NMB_CHR   BM01.00 TEXT    “Kuhnke” ;string to be sent incl. no
  2:
  3:
```

```
========  Kubes  ===================================================
                        Init. module IL
Project:  V24_EXT
Module  : INI_V24         No.: 1     created :   Jul 21 1993 09:35
User    : Gerd Hildebrandt           altered :   Jul 21 1993 13:09
Comment:  INI_V24

=====================================================================
 1: BAUD          BM00.00 WORD    12            ;(Baudrate) 9600 baud
 2: FORMAT        BM00.02 BYTE    %0000_0011    ;(data format) 8,1,N
 3: SLOT          BM00.04 BYTE    3             ;(module slot (0...7))
 4:
```

7.5. PROFIBUS modules

PROFIBUS modules can be used in conjunction with KUAX 680I, KUAX 680C and KDT 680CT. They use the PROFIBUS-DP protocol that is standardised by DIN 19245, Part 3. These modules are equipped with their own processor to minimise the load on the CPU. The actual communication program is located in the EPROM inside the module.
Use KUBES modules in your user program to define the data that is to be exchanged between communication partners.

The "PROFIBUS-DP slave module" provides slave functions and has part number 680.440.05.

The PROFIBUS-DP master module had not been completed at the time this manual was printed.

Due to their extensive descriptions, PROFIBUS modules have their own instruction manual:

☞

Instruction Manual
PROFIBUS Modules
for the 680 System
E 509 GB

8. Stepper motor modules without processor

Stepper motor modules are used for controlling stepper motors. In KUAX 680I as from monitor version 4.16.

⚠ *When working with KUAX 680C you must make sure that the module uses the same processor ressources as the internal analog outputs (see ch. "3.6. Differences...").*

One stepper motor module with 1 or 2 channels can be applied per controller, i.e. up to 2 stepper motors can be controlled (except for modules equipped with a processor, see ch. 9).

Program

The modules have no processor and no data memory for program-controlled runs of their own.
The software for the stepper motor is in the service module which is embedded into the program during configuration of the controller (under KUBES). The service module also defines the transfer addresses function which regulates the communication between user program and stepper motor software.
The start and stop frequencies, the ramp slopes and the maximum travelling frequency are all defined by the user via the software. The maximum frequency is 10 kHz.

Inputs and outputs

The signal level for the inputs and outputs is 24 V DC.
The hardware limit switches directly affect the module without going through the PLC program. They are used as limit switches and also as reference switches during reference runs.
The outputs are also directly controlled by the module.

Power packs

Additional external power packs are needed for driving the motors. These are not part of the module but adapted to the motor. They are normally delivered by the supplier of the stepper motor.

8.1. Plugging stepper motor modules

Stepper motor modules use the 2 pulse-width modulated outputs (PWM) of the processor on the CPU board of the controller.
These PWM outputs are installed in parallel on 2 slots (see "3.3.1. Function slots") so that a stepper motor module can also be applied to only one of these outputs:

Slot no.	Useable PWM channels
1	0, 1
2	0, 1

⚠ *Running a stepper motor module from slot 1 excludes the existence of an analog input module with a resolution of 10 bit in slot 2.*

A 10bit input module in slot 2 switches the PWM outputs of the processor off.
This is not valid in the opposite case as no such input module can be run from slot 1.

8.2. Service modules

Service modules are delivered together with KUBES (as from version 4.00). After installing KUBES on the PC they are in the sub-directory that contains the program files (KUBESEXE). They are transferred into the user program of the controller together with the project.
The service module for the stepper motor module contains the stepper motor software that makes the functions of the module available. The service module also defines the assignment of the transfer addresses.

8.2.1. Transfer address ranges

Each slot is assigned two transfer address ranges of 32 byte each. The stepper motor module uses these for data communication with the CPU.
On the occurrence of an interrupt (by the reference switches), the stepper motor module triggers an interrupt module. The used slot also determines which interrupt module is triggered:

Slot	SM channel*)	Transfer address range	Interrupt module
1	0	SLC00.00...01.15	3
	1	SLD00.00...01.15	4
2	0	SLE00.00...01.15	5
	1	SLF00.00...01.15	6

*) This is always channel 0 with stepping motor modules with 1 channel

8.2.2. Assignment of transfer addresses

Assigns a function to the tranfer addresses. The address selection is decided by the slot into which the module is plugged and by the channel that is used:

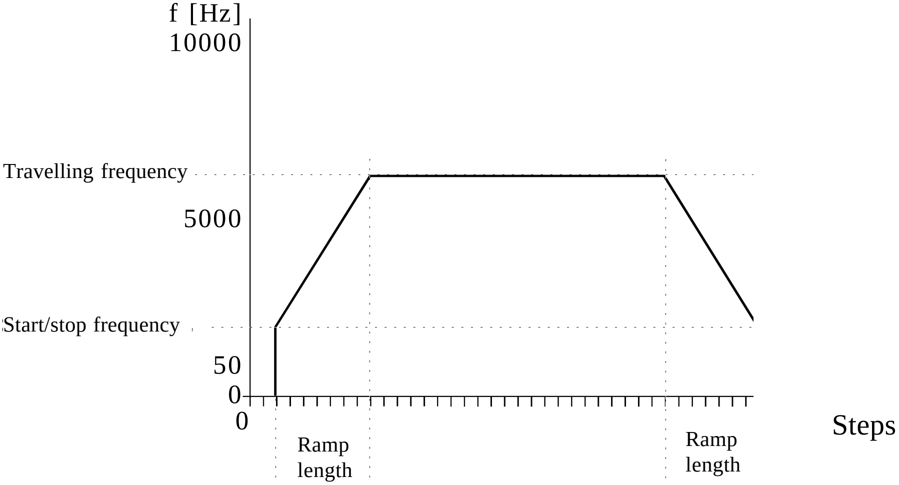
Address	Symbol	Significance		
SLx00.00	DP_LW	destination position: destination (in steps) of a program run	low word	low byte
SLx00.01				high byte
SLx00.02	DP_HW		high word	low byte
SLx00.03				high byte
SLx00.04	RP_LW	actual postion: current position (in steps)	low word	low byte
SLx00.05				high byte
SLx00.06	RP_HW		high word	low byte
SLx00.07				high byte
SLx00.08	SLSL_LW	software limit switch left: max. position left, stated in steps	low word	low byte
SLx00.09				high byte
SLx00.10	SLSL_HW		high word	low byte
SLx00.11				high byte
SLx00.12	SLSR_LW	software limit switch right: max. position right, stated in steps	low word	low byte
SLx00.13				high byte
SLx00.14	SLSR_HW		high word	low byte
SLx00.15				high byte
SLx01.00	SV_LW	preset value step counter: is used for setting a defined actual poisition, taken over by bit 5 in SLx01.07	low word	low byte
SLx01.01				high byte
SLx01.02	SV_HW		high word	low byte
SLx01.03				high byte
SLx01.04	ERR	error code		
SLx01.05		unused		
SLx01.06		unused		
SLx01.07	CONTROL	control functions (see "8.3.2.8. ....")		
SLx01.08	RAMP	start and stop ramp (stated in steps, min: 2)		low byte
SLx01.09				high byte
SLx01.10	ST_STO_F	start/stop frequency: min. 50 Hz, max. 5 kHz		low byte
SLx01.11				high byte
SLx01.12	MOV_FRQ	travelling frequency: min. start/stop frequency, max. 10 kHz		low byte
SLx01.13				high byte
SLx01.14	MODE	mode of operation (see "8.3.2.7....")		
SLx01.15	SFT_VER	stepper motor software version (service module)		

8.3. Software

During controller configuration with the program KUBES, a service module is automatically embedded into the program when the stepper motor module is selected.
This module contains the stepper motor software as well as the assignment list of the transfer addresses described in chapter 8.2.
The user program uses this transfer address range for data communication with the stepper motor software and with the module.
Here the run parameters and the commands are transferred and the feedback messages are read.

8.3.1. Representation of a run diagram

The run diagram shows the example of a typical run:



The terminology used in this diagram is explained on the following pages.

8.3.2. Terminology and functions

Step

A step is the path length a stepper motor covers with each clock pulse.

8.3.2.1. Start/stop ramp SLx01.08 (RAMP)

The ramp length indicates the number of steps between start and travelling frequency and between travelling frequency and stop. The minimum ramp length is 2 steps. If you input a value smaller 2, the software will automatically input the value 2.
The software automatically calculates the ramp slope in dependence of the ramp length and the travelling frequency.
In order to reduce the load on the processor during positioning, a new run command will only be calculated and executed after finishing the last one.

8.3.2.2. Start/stop frequency SLx01.10 (ST_STO_F)

The start/stop frequency can be set in a range of 50 Hz to 5 kHz. It is valid for start and stop.
If the travelling frequency of a positioning run is smaller than the start/stop frequency, then the run will be executed without ramps. The travelling frequency will be increased to the value of the start/stop frequency.

8.3.2.3. Travelling frequency SLx01.12 (MOV_FRQ)

The travelling – or moving – frequency determines the motor speed. Its minimum value can be as small as the set start/stop frequency; its maximum value is 10 kHz.
Each positioning operation may have a different frequency.
The set travelling frequency remains valid for all subsequent runs, including manual runs and reference runs, until it is changed.

8.3.2.4. Actual position SLx00.04 (RP_LW)

An internal step counter counts the steps of the run in dependence of the running direction. The count level is equal to the actual position.
This counter has a size of 31 bit plus sign.
The counter is also influenced by

- reference run:
  sets the count level to 0 as soon as the reference point is reached
- Preset value step counter SLx01.00 (SV_LW):
  the software copies the preset value into the counter as soon as 255 is written into address SLx01.04 (SV_TO_RP)

8.3.2.5. Destination position SLx00.00 (DP_LW)

The destination position indicates the preset end of the run in steps. The position is always indicated as an absolute value, i.e. in relation to zero. The preset value can carry a positive or a negative sign.

8.3.2.6. Preset value step counter SLx01.00 (SV_LW)

The step counter preset allows you to change the step counter (actual position) to any desired value at any time. The software copies the preset value into the counter as soon as bit 5 in byte "Control functions" SLx01.07 (CONTROL) is set.

8.3.2.7. Mode of positioning operations SLx01.14 (MODE)

Input a certain value into this byte for setting the desired mode of positioning operation:

Mode of operation		Start	Run	Stop
Value	Name			
2	manual run left	"Start" command *1)	in negative direction, considering ramp function and step frequency	"Stop" command *2)
3	manual run right		in positive direction, considering ramp function and step frequency	
4	reference run left		in negative direction, considering ramp function and step frequency	end switch left
5	reference run right		in positive direction, considering ramp function and step frequency	end switch right
8	positioning run		in target direction, considering ramp function and step frequency	target reached

*1) The start command is triggered by bit 0 in control byte SLx01.07.

*2) The stop command is triggered by bit 1 in control byte SLx01.07.

*3) If the hardware and software limit switches are enabled (see control byte Sl they stop the run as soon as they are triggered.

Stepper motor enable:

The motor must receive an enable signal before starting the run or it will not start. If the release signal is switched off during a run, the motor will stop at once.
The enable signal is given by setting bit 4 in byte "control functions" SLx01.07 (CONTROL).

Manual run:

A manual run is stopped automatically (with ramps) when a software limit switch is reached. An error message will be put out. The manual run can still be continued in the opposite direction. The error message extinguishes when the software limit switch is left.

Reference run:

In this case the hardware limit switches serve as reference switches. The software limit switches remain inactive.

Positioning run:

If the destination position is outside the range of the set software limit switches, then the motor will not start and an error message will be put out (see "8.3.2.9. Error messages", bit 5). This error message extinguishes automatically as soon as a valid run command is entered.
The run is stopped as soon as the destination position is reached. The software then clears the start bit in control byte SLx01.07 (CONTROL). This operation can be analyzed as feedback message by the user program.

Hardware limit switch

Upon reaching a limit switch the stepper motor software triggers an Emergency Off. The motor stops without ramp. An error message is put out.

Run without hardware or software limit switches:

For operation without limit switches, the hardware and software limit switches can be deactivated via the software. This is done by clearing the corresponding bits in control byte SLx01.07 (CONTROL).

8.3.2.8. Control functions SLx01.07 (CONTROL)

The command is enabled by setting the corresponding bit to 1. The stepper motor software acknowledges by 0:

Bit	Function
0	Start: Starts a run in the selected mode of operation and considering the set ramp functions. The mode of operation must be set in Mode byte SLx01.14 (MODE) before. As a feedback message, the software clears the bit at the end of the run.
1	Stop: Stops the current run under consideration of the ramp function set.
2	Emergency Off: Immediately stops the current run without ramp function.
3	unused
4	Stepper motor enable: Enables the release signal, output "enable".
5	Copy preset value SLx01.00 (SV_LW) into Actual position SLx00.04 (RP_LW) (4 byte).
6	Enable hardware limit switch: Enables the function of the connected hardware limit switches. Reaching these limit switches immediately stops the run without ramp function. Note: during a reference run, the hardware limit switches work as reference switches. This is active during every reference run, independent of the status of the enable signal.
7	Software limit switch enable: Enables the function of the set software limit switches. Reaching these limit switches stops the run under consideration of the ramp function. Early recognition guarantees the stop at an exaxt position despite the ramp.

To set control bits

Using an OR operation to set individual bits in a byte by a binary value where only the desired bit takes on status "1". All bits stating "0" remain unchanged.

Example "Start run":

```
        L      SLx01.07      ;load control byte
        O      %00000001     ;set bit 0
        =      SLx01.07      ;write into control byte
```

To reset control bits

Using an AND operation to reset individual bits in a byte by a binary value where only the desired bit takes on status "0". All bits stating "1" remain unchanged.

Example "Disable hardware limit switch":

```
        L      SLx01.07      ;load control byte
        A      %110111111    ;reset bit 6
        =      SLx01.07      ;write into control byte
```

To scan control bits

Using an AND operation to register the status of an individual bit in a byte where only the desired bit is set to "1". The result is evaluated by a jump command.

Example "Check whether run complete":

```
        L      SLx01.07      ;load control byte
        A      %00000001     ;mask bit 0
        JP<>   R_OVER        ;jump if bit still set
;here the evaluation for "run over" is done
        .
        .
        .
R_OVER NOP                   ;return to normal prog. run
```

8.3.2.9. Error messages

The following errors may occur:

Error		Function
Bit	Number	
0	1	hardware limit switch left reached
1	2	hardware limit switch right reached
2	3	software limit switch left reached
3	4	software limit switch right reached
4	5	no "ready" message from motor
5	6	illegal run command
6	7	short-circuit or overload on output
7		unused

Error byte SLx01.04 (ERR)

Any occurring errors can be detected by the user program via this byte. Each bit (see table above) stands for an error message. The message is valid if the bit is set to 1.

Example for evaluation:

To facilitate message evaluation we recommend to map the contents of the byte in 8 bit markers:

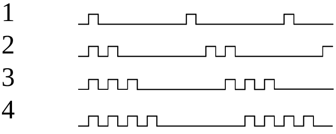
```
L      SLx01.04  ;load error byte
C8T1   M00.00    ; copy to M00.00...07
```

Bit 0 (error1) is now in M00.00, bit 1 in M00.01 etc.

LED "ERR"

This red light emitting diode is placed on top of the module. In case of any occurring errors the LED flashes in a rhythm that reflects the error number (see table above):

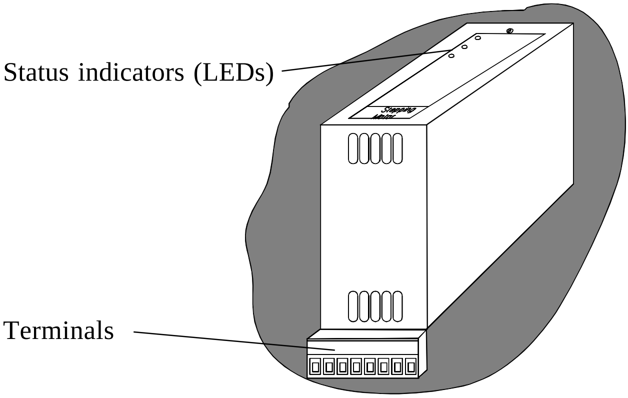
No.   Rhythm of flashes



etc.

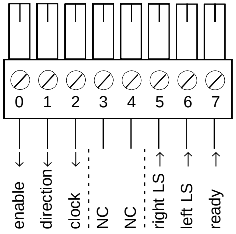
The LED flashes in fast impulses (250/250ms). Then there is an interval of 1s and the flashing starts again.

8.4. Stepper motor module, 1 channel



8.4.1. Connection of the signal lines

The input and output signals are connected to the module by screw-type locking terminals on the signal strip underneath the module:

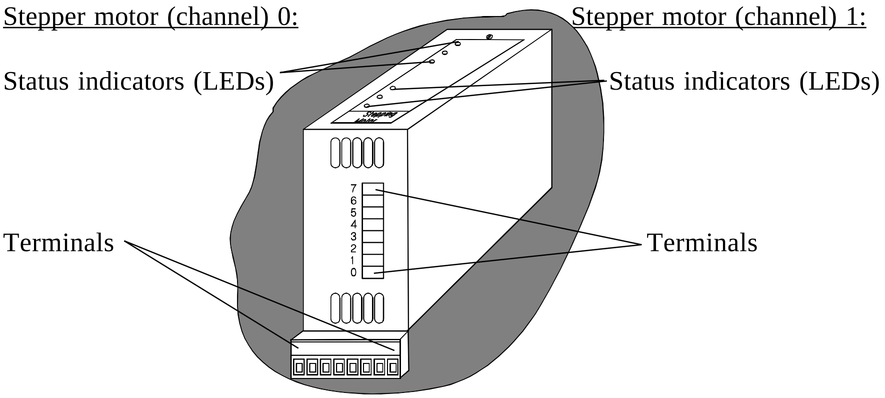


Connector	*Function*
enable	motor enable (to the motor)
direction	direction of rotation (to the motor)
clock	clock pulse (to the motor)
NC	unused
right LS	right limit switch (n.c. switch)
left LS	left limit switch (n.c. switch)
ready	motor ready message (n.o. switch)

8.4.2. Technical specifications

Application:	KUAX 680I (as from ver. 4.16), KUAX 680C, KDT 680CT	
slot	1 or 2	
Admissible ambient conditions		
storage temperature	-25...+70 °C	
ambient temp. during operation	0...55 °C	
relative humidity	50...95 %	
Number of stepper motors (channels)	1	
Travelling frequency	10 kHz max.	
Counting depth	31 bit plus sign bit	
Inputs:	3	
Type (in acc. with IEC 1131)	1, terminals 5, 6, 7	
Potential separation	no	
Input voltage:	24 V DC -20 % / +25 % (incl. residual ripple)	
Signal identification		
logical 0:	≤ 5 V DC	
logical 1:	≥ 15 V DC	
Max. voltage:	28.8 V DC	
Power consumption / input:	10 mA max.	
Outputs:	3, terminals 1, 2, 3	
Output voltage:	24 V DC -20 % / +25 % (incl. residual ripple)	
Output current:	80 mA max.	
Status indicators:	3 LEDs	
Function	*Designation*	*Colour*
reference point reached	REF	green
error message	ERR	red
motor ready message	STAT	green
Weight:	c. 82 g	
Part number:	680.444.01	

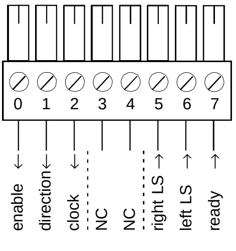
8.5. Stepper motor module, 2 channels



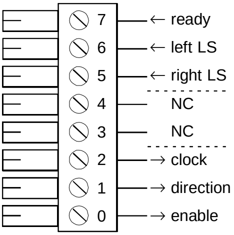
8.5.1. Connection of the signal lines

The input and output signals are connected to the module by screw-type locking terminals underneath (channel 0) the module and on (channel 1) the module.

Channel 0:



Channel 1:



Connector	*Function*
enable	motor enable (to the motor)
direction	direction of rotation (to the motor)
clock	clock pulse (to the motor)
NC	unused
right LS	right limit switch (n.c. switch)
left LS	left limit switch (n.c. switch)
ready	motor ready message (n.o. switch)

8.5.2. Technical specifications

Application:	KUAX 680I (as from ver. 4.16), KUAX 680C, KDT 680CT	
slot	1 or 2	
Admissible ambient conditions		
storage temperature	-25...+70 °C	
ambient temp. during operation	0...55 °C	
relative humidity	50...95 %	
Number of stepper motors (channels)	2	
Travelling frequency	10 kHz max.	
Counting depth	31 bit plus sign bit	
<u>Inputs</u>:	3 per channel, terminals 5, 6, 7	
Type (in acc. with IEC 1131)	1	
Potential separation	no	
Input voltage:	24 V DC -20 % / +25 % (incl. residual ripple)	
Signal identification		
logical 0:	≤ 5 V DC	
logical 1:	≥ 15 V DC	
Max. voltage:	28.8 V DC	
Power consumption / input:	10 mA max.	
<u>Outputs</u>:	3 per channel, terminals 1, 2, 3	
Output voltage:	24 V DC -20 % / +25 % (incl. residual ripple)	
Output current:	80 mA max.	
Short-circuit protection:	yes	
Status indicators:	3 LEDs	
Function	*Designation*	*Colour*
reference point reached	REF	green
error message	ERR	red
motor ready message	STAT	green
Weight:	c 116 g	
Part number:	680.444.02	

8.6. Example program

```
 ======== KUBES ======================================================
                           Project  structure
Project :   SM1_680I
                                  created : Aug 30 1993 11:32
User    :   Lillge                modified: Jun 10 1994 12:36
Comment :   Test program for SM module 680I
  ====================================================================
ORG.ORG/1
|
*———>PARAME.INI/1
|
*———>COMMANDS.PRO/1
        |
        *———>AUTOPRO.PRO/2
```

```
 ======== KUBES ======================================================
                           Organisationsbaustein   IL
Project: SM1_680I
Module : ORG       No.: 1      created : Aug 30 1993 11:33
User   : Lillge                modified: Aug 30 1993 15:18
  ====================================================================
  1:          L       INI_OK      M00.00 ; (SM init. OK)
  2:          JPC     RUN
  3:          JPINIT  PARAME          1
  4:          JP      END
  5: RUN      JPP     COMMANDS        1
  6: END      NOP
  7:
```

```
======== KUBES ====================================================
                      Program module  IL
Project: SM1_680I
Module : AUTOPRO   No.: 2      created : Aug 31 1993 14:31
User   : Lillge                modified: Sep 03 1993 08:07
Comment: Step sequence for posi program
==================================================================
STEP_0   L       SBM15.00
         CMP     0
         JP<>    STEP_1
         LD      65000
         =D      X_PRE_LL     SLC00.00;(pres.pos.x-ax.SM)
         LD      $0000
         =D      X_PRE_HL     SLC00.02;(pres.pos.x-ax.SM)
         INC     SBM15.00
         JP      END

STEP_1   L       SBM15.00
         CMP     1
         JP<>    STEP_2
         LD      $01D0
         =D      X_PRE_LL     SLC00.00;(pres.pos.x-ax.SM)
         LD      $0001
         =D      X_PRE_HL     SLC00.02;(pres.pos.x-ax.SM)
         INC     SBM15.00
         JP      END

STEP_2   L       SBM15.00
         CMP     2
         JP<>    STEP_3
         LD      $09A0
         =D      X_PRE_LL     SLC00.00;(pres.pos.x-ax.SM)
         LD      $0001
         =D      X_PRE_HL     SLC00.02;(pres.pos.x-ax.SM)
         INC     SBM15.00
         JP      END

STEP_3   L       SBM15.00
         CMP     3
         JP<>    STEP_4
         LD      8000
         =D      X_PRE_LL     SLC00.00;(pres.pos.x-ax.SM)
         LD      $0000
         =D      X_PRE_HL     SLC00.02;(pres.pos.x-ax.SM)
         INC     SBM15.00
         JP      END

```

```
STEP_4   L       SBM15.00
         CMP     4
         JP<>    STEP_5
         LD      9000
         =D      X_PRE_LL    SLC00.00;(pres.pos.x-ax.SM)
         LD      $0000
         =D      X_PRE_HL    SLC00.02;(pres.pos.x-ax.SM)
         INC     SBM15.00
         JP      END

STEP_5   L       SBM15.00
         CMP     5
         JP<>    STEP_6
         LD      10000
         =D      X_PRE_LL    SLC00.00;(pres.pos.x-ax.SM)
         LD      $0000
         =D      X_PRE_HL    SLC00.02;(pres.pos.x-ax.SM)
         INC     SBM15.00
         JP      END

STEP_6   L       SBM15.00
         CMP       6
         JP<>    DEL
         LD      2000
         =D      X_PRE_LL    SLC00.00;(pres.pos.x-ax.SM)
         LD      $0000
         =D      X_PRE_HL    SLC00.02;(pres.pos.x-ax.SM)
         INC     SBM15.00
         JP      END

DEL      L       SBM15.00
         CMP       7
         JP<>    END
         CLR     SBM15.00

END      NOP

```

```
======== KUBES =====================================================
                        Program module  IL
Projext: SM1_680I
Module : COMMANDS    No.: 1      created : Aug 30 1993 15:09
User   : Lillge                  modified: Nov 09 1993 17:54
Comment: Execute SM functions
====================================================================
; —— Trigger reference run ——
          L       X_REF       I00.00 ; (start X ref. run)
          =       PP00.00
          L       PP00.00
          JPCN    REF_END
          LD      500
          =D      X_RAMP_L    SLC01.08;(ramp x-ax.in steps

          LD      2500               ; Hz
          =D      X_POSV_L    SLC01.12;(posit.speed  x-ax.)
          L       4                     ;Ref.run,4=lft,5=rgt
          =       X_MODE      SLC01.14;(x-ax.oper.mode)

          L       X_COMMAN    SLC01.07;(cmnd byte x-axis)
          O       %0000_0001
          =       X_COMMAN    SLC01.07;(cmnd byte x-axis)

REF_END   NOP

; —— Start positioning step –
          L       X_START     I00.02 ; (start x-axis)
          =       PP00.02
          L       PP00.02
          JPCN    STA_END
          LD      800
          =D      X_RAMP_L    SLC01.08;(ramp x-ax.in steps

          LD      10500
          =D      X_PRE_LL    SLC00.00   ;(pres.pos.x-ax.SM)
          LD      $0000
          =D      X_PRE_HL    SLC00.02   ;(pres.pos.x-ax.SM)

          LD      10000                  ; Hz
          =D      X_POSV_L    SLC01.12   ;(pos.speed  x-ax.)
          L       8                      ; Posi step
          =       X_MODE      SLC01.14   ;(oper.mode  x-axis)

          L       X_COMMAN    SLC01.07   ;(cmnd byte x-axis)
          O       %0000_0001
          =       X_COMMAN    SLC01.07   ;(cmnd byte x-axis)
          CLR     SBM15.00

STA_END   NOP
```

```
; —— Start posi program ————
         L       X_REFSCH        I00.01  ;(X-ref.init.,simul.
         JPCN    SA_END
         L       X_COMMAN     SLC01.07   ;(cmnd byte x-axis)
         A       %0000_0001
         JP<>    SA_END
         JPP     AUTOPRO           2

         LD      800
         =D      X_RAMP_L     SLC01.08   ;(ramp x-ax.in steps

         LD      10000                   ; Hz
         =D      X_POSV_L     SLC01.12   ;(pos.x-ax.in  steps)
            L          8                 ; Posi step
         =       X_MODE       SLC01.14   ;(oper.mode  x-axis)

         L       X_COMMAN     SLC01.07   ;(cmnd byte x-axis)
         O       %0000_0001
         =       X_COMMAN     SLC01.07   ;(cmnd byte x-axis)

SA_END   NOP

; —— Trigger stop (with ramps) —
         L       X_STOP          I00.03  ;(stop x-axis)
         =       PP00.03
         L       PP00.03
         JPCN    STP_END

         L       X_COMMAN     SLC01.07   ;(cmnd byte x-axis)
         O       %0000_0010
         =       X_COMMAN     SLC01.07   ;(cmnd byte x-axis)
STP_END  NOP

; —— Trigger emergency off (without ramps)–
         L       X_NSTOP         I00.04  ;(emerg.off  x-axis)
         =       PP00.04
         L       PP00.04
         JPCN    NST_END

         L       X_COMMAN     SLC01.07   ;(cmnd byte x-axis)
         O       %0000_0100
         =       X_COMMAN     SLC01.07   ;(cmnd byte x-axis)
NST_END  NOP

```

```
; —— Positive manual run ———
          L       X_H_NEG          I00.05 ;(neg.man.run  x-ax.)
          =       PP00.05
          L       PP00.05
          JPCN    HRI_STP
          LD      50
          =D      X_RAMP_L    SLC01.08    ;(ramp x-ax.in steps
          LD      1500                    ; Hz
          =D      X_POSV_L    SLC01.12    ;(pos.speed  x-axis)
          L       3
          =       X_MODE      SLC01.14    ;(oper.mode  x-axis)

          L       X_COMMAN    SLC01.07    ;(cmnd byte  x-axis)
          O       %0000_0001
          =       X_COMMAN    SLC01.07    ;(cmnd byte  x-axis)
          JP      HRI_END

HRI_STP   LN      X_H_NEG          I00.05 ;(neg.man.run  x-ax.)
          =       PP00.06
          L       PP00.06
          JPCN    HRI_END

          L       X_COMMAN    SLC01.07    ;(cmnd byte  x-axis)
          O       %0000_0010
          =       X_COMMAN    SLC01.07    ;(cmnd byte  x-axis)
HRI_END   NOP

; —— Negative manual run ———
          L       X_H_POS          I00.06 ;(pos.man.run  x-ax.)
          =       PP00.07
          L       PP00.07
          JPCN    HLE_STP
          LD      50
          =D      X_RAMP_L    SLC01.08    ;(ramp x-ax.in steps
          LD      1500                    ; Hz
          =D      X_POSV_L    SLC01.12    ;(pos.speed  x-ax.)
          L       2
          =       X_MODE      SLC01.14    ;(oper.mode  x-axis)
          L       X_COMMAN    SLC01.07    ;(cmnd byte  x-axis)
          O       %0000_0001
          =       X_COMMAN    SLC01.07    ;(cmnd byte  x-axis)

```

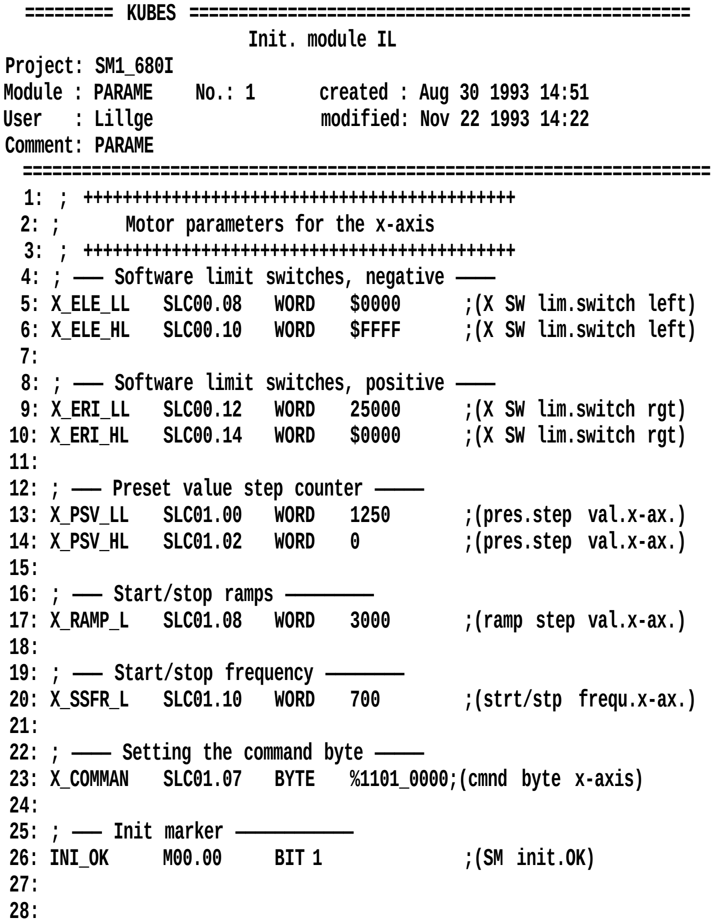
```
HLE_STP  LN       X_H_POS               I00.06 ;(pos.man.run  x-ax.)
         =        PP00.08
         L        PP00.08
         JPCN     HLE_END

         L        X_COMMAN    SLC01.07    ;(cmnd byte x-axis)
         O        %0000_0010
         =        X_COMMAN    SLC01.07    ;(cmnd byte x-axis)
HLE_END  NOP
; ——— New value for step counter ——
         L        X_PSV_TO        I00.07  ;(import PV x-ax.)
         =        PP00.09
         L        PP00.09
         JPCN     TOV_END
         LD       64000
         =D       X_PSV_LL    SLC01.00    ;(pres.val.stepx-ax)
         LD       0
         =D       X_PSV_HL    SLC01.02    ;(pres.val.stepx-ax)

         L        X_COMMAN    SLC01.07    ;(cmnd byte x-axis)
         O        %0010_0000
         =        X_COMMAN    SLC01.07    ;(cmnd byte x-axis)
TOV_END  NOP
; ——— Status message ———————

;        Step counter
         LD       X_ACT_LL    SLC00.04    ;(act.pos.x-axis  SM)
         LD       X_ACT_HL    SLC00.06    ;(act.pos.x-axis  SM)
;        Error messages
         L        X_ERROR     SLC01.04    ;(error code x-axis)
;        Software identification
         L        X_SWIDNT    SLC01.15    ;(SW ident. x-ax.)

```



```
 ========= KUBES ===================================================
                         Init. module IL
Project: SM1_680I
Module : PARAME    No.: 1      created : Aug 30 1993 14:51
User   : Lillge                modified: Nov 22 1993 14:22
Comment: PARAME
 =====================================================================
 ; ++++++++++++++++++++++++++++++++++++++++++++
 ;       Motor parameters for the x-axis
 ; ++++++++++++++++++++++++++++++++++++++++++++
 ; ——— Software limit switches, negative ———
 X_ELE_LL   SLC00.08   WORD   $0000      ;(X SW lim.switch left)
 X_ELE_HL   SLC00.10   WORD   $FFFF      ;(X SW lim.switch left)

 ; ——— Software limit switches, positive ———
 X_ERI_LL   SLC00.12   WORD   25000      ;(X SW lim.switch rgt)
 X_ERI_HL   SLC00.14   WORD   $0000      ;(X SW lim.switch rgt)

 ; ——— Preset value step counter —————
 X_PSV_LL   SLC01.00   WORD   1250       ;(pres.step val.x-ax.)
 X_PSV_HL   SLC01.02   WORD   0          ;(pres.step val.x-ax.)

 ; ——— Start/stop ramps —————————
 X_RAMP_L   SLC01.08   WORD   3000       ;(ramp step val.x-ax.)

 ; ——— Start/stop frequency ————————
 X_SSFR_L   SLC01.10   WORD   700        ;(strt/stp frequ.x-ax.)

 ; ———— Setting the command byte —————
 X_COMMAN   SLC01.07   BYTE   %1101_0000;(cmnd byte x-axis)

 ; ——— Init marker ————————————
 INI_OK     M00.00     BIT 1             ;(SM init.OK)


```

9. Stepper motor module with processor

Stepper motor modules are used for controlling stepper motors. The module described in this chapter has its own processor thus reducing the load on the CPU capacity. The restrictions and limitations of the stepper motor modules without processor described in chapter 8 (amount, slots, processor resources in KUAX 680C) do not apply to this module.

Program

The actual program for the stepper motor is stored in the EPROM (of the module) and is processed by the module processor. All that has to be included in the user program is defining positioning jobs, giving start and stop commands and analysing the status and error messages created by the module. A service module defines the transfer address functions controlling the communication processes between user program and stepper motor software.
Start and stop frequency, ramp slopes and max. positioning frequencies are optional parameters to be set via the software. Max. frequency is 15 kHz.

Inputs and outputs

The signal level for the inputs and outputs is 24 V DC.
The hardware limit switches directly affect the module without going through the PLC program. They are used as limit switches and also as reference switches during reference runs. The outputs are also directly controlled by the module.

Power packs

Additional external power packs are needed for driving the motors. These are not part of the module but adapted to the motor. They are normally delivered by the supplier of the stepper motor.

9.1. Plugging stepper motor modules

As opposed to the stepper motor modules without processor (chapter 8), this module can be plugged into any controller slot. The only limitation to the number of plugged-in modules is the availability of free slots and the total power consumption of all plugged-in modules.

9.2. Service modules

Service module "SERV_85.BIN" for the stepper motor module becomes part of the KUBES software package (not yet included in KUBES versions up to 4.30; you can, however, order it separately or download it from our mailbox, tel. +49-4523-402-310). The service module defines the transfer addresses and processes communication processes to and from the module.

9.2.1. Transfer address ranges

Each slot is assigned two transfer address ranges of 32 byte each. The stepper motor module uses these for data communication with the CPU. The module does not release interrupts.

Slot	Transfer Address Range	SM Channel
0	SLA00.00...01.15 SLB00.00...01.15	0 1
1	SLC00.00...01.15 SLD00.00...01.15	0 1
2	SLE00.00...01.15 SLF00.00...01.15	0 1
3	SLG00.00...01.15 SLH00.00...01.15	0 1
4	SLI00.00...01.15 SLJ00.00...01.15	0 1
5	SLK00.00...01.15 SLL00.00...01.15	0 1
6	SLM00.00...01.15 SLN00.00...01.15	0 1
7	SLO00.00...01.15 SLP00.00...01.15	0 1

9.2.2. Assignment of transfer addresses

See table on previous page for assignment of address ranges.

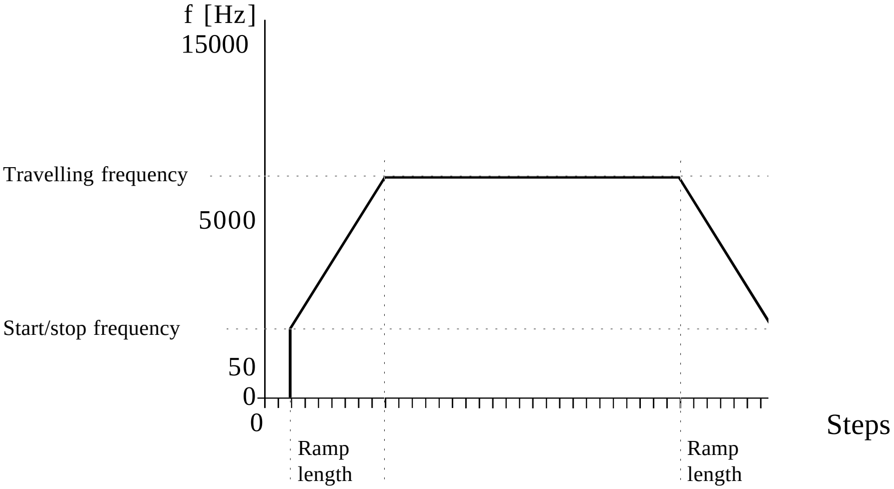
Address	Symbol	Significance		
SLx00.00	DP_LW	destination position:	low word	low byte
SLx00.01				high byte
SLx00.02	DP_HW	destination (in steps) of a program run	high word	low byte
SLx00.03				high byte
SLx00.04	RP_LW	actual postion:	low word	low byte
SLx00.05				high byte
SLx00.06	RP_HW	current position (in steps)	high word	low byte
SLx00.07				high byte
SLx00.08	SLSL_LW	software limit switch left:	low word	low byte
SLx00.09				high byte
SLx00.10	SLSL_HW	max. position left, stated in steps	high word	low byte
SLx00.11				high byte
SLx00.12	SLSR_LW	software limit switch right:	low word	low byte
SLx00.13				high byte
SLx00.14	SLSR_HW	max. position right, stated in steps	high word	low byte
SLx00.15				high byte
SLx01.00	SV_LW	preset value step counter:	low word	low byte
SLx01.01		used for setting a defined actual poisition, taken over by bit 5 in SLx01.07		high byte
SLx01.02	SV_HW		high word	low byte
SLx01.03				high byte
SLx01.04	ERR	error code (see ch. "9.3.2.10...")		
SLx01.05	DEST_OK	start: master writes $00, dest. reached: slave writes $FF		
SLx01.06	BUF_FUL	job buffer overflow in module: slave writes $FF		
SLx01.07	CONTROL	control functions (see "9.3.2.9. ....")		
SLx01.08	RAMP	start and stop ramp (stated in steps, min: 2)		low byte
SLx01.09				high byte
SLx01.10	ST_ST_F	start/stop frequency: min. 50 Hz, max. 5 kHz		low byte
SLx01.11				high byte
SLx01.12	MOV_FRQ	travelling frequency: min. start/stop frequency, max. 15 kHz		low byte
SLx01.13				high byte
SLx01.14	MODE	mode of operation (see "9.3.2.7....")		
SLx01.15	SW_V_SB SW_V_MO	service module version (address ranges A, C, E, ...O) module software version (address ranges B, D, F, ...P)		

9.3. Software

During controller configuration with the program KUBES, a service module is automatically embedded in the program when the stepper motor module is selected.
This module contains the stepper motor software as well as the assignment list of the transfer addresses described in chapter 9.2.
The user program uses this transfer address range for data communication with the stepper motor module software.
Here the run parameters and the commands are transferred and the feedback messages are read.

9.3.1. Representation of a run diagram

The run diagram shows the example of a typical run:



The terminology used in this diagram is explained on the following pages.

9.3.2. Terminology and functions

Step

A step is the path length a stepper motor covers with each clock pulse.

9.3.2.1. Start/stop ramp SLx01.08 (RAMP)

The ramp length indicates the number of steps between start and travelling frequency and between travelling frequency and stop. The minimum ramp length is 2 steps. If you input a value smaller 2, the software will automatically input the value 2. The software automatically calculates the ramp slope in dependence of the ramp length and the travelling frequency.

9.3.2.2. Start/stop frequency SLx01.10 (ST_STO_F)

The start/stop frequency can be set in a range of 50 Hz to 5 kHz. It is valid for start and stop.
If the travelling frequency of a positioning run is smaller than the start/stop frequency, then the run will be executed without ramps. The travelling frequency will be increased to the value of the start/stop frequency.

9.3.2.3. Travelling frequency SLx01.12 (MOV_FRQ)

The travelling – or moving – frequency determines the motor speed. Its minimum value can be as small as the set start/stop frequency; its maximum value is 15 kHz.
Each positioning operation may have a different frequency.
The set travelling frequency remains valid for all subsequent runs, including manual runs and reference runs, until it is changed.

9.3.2.4. Actual position SLx00.04 (RP_LW)

An internal step counter counts the steps of the run in dependence of the running direction. The count level is equal to the actual position.
This counter has a size of 31 bit plus sign bit.
The counter is also influenced by

- reference runs:
  set the count level to 0 as soon as the reference point is reached
- preset value step counter SLx01.00 (SV_LW):
  the software overwrites the actual value with the preset value as soon as bit 5 has been set in address SLx01.07 (CONTROL)

9.3.2.5. Destination position SLx00.00 (DP_LW)

The destination position indicates the preset end of the run in steps. The position is always indicated as an absolute value, i.e. in relation to zero. The preset value can carry a positive or a negative sign.

9.3.2.6. Preset value step counter SLx01.00 (SV_LW)

The step counter preset allows you to change the step counter (actual position) to any desired value at any time. The software copies the preset value into the counter as soon as bit 5 in byte "Control functions" SLx01.07 (CONTROL) is set.

9.3.2.7. Mode of positioning operations SLx01.14 (MODE)

Input a certain value into this byte for setting the desired mode of positioning operation:

Mode of operation		Start	Run	Stop
Value	Name			
2	manual run left	"Start" command *1)	in negative direction, considering ramp function and step frequency	"Stop" command *2)
3	manual run right		in positive direction, considering ramp function and step frequency	
4	reference run left		in negative direction, considering ramp function and step frequency	end switch left
5	reference run right		in positive direction, considering ramp function and step frequency	end switch right
8	positioning run		in target direction, considering ramp function and step frequency	target reached
24	positioning run 1000 Hz		in target direction, without ramp function, with max. 1000 Hz	target reached

*1) The start command is triggered by bit 0 in control byte SLx01.07.

*2) The stop command is triggered by bit 1 in control byte SLx01.07.

*3) If the hardware and software limit switches are enabled (see control byte Sl they stop the run as soon as they are triggered.

Stepper motor enable:

The motor must receive an enable signal before starting the run or it will not start. If the enable signal is switched off during a run, the motor will stop at once.
The enable signal is given by setting bit 4 in byte "control functions" SLx01.07 (CONTROL).

Manual run:

A manual run is stopped automatically (with ramps) when a software limit switch is reached. An error message will be put out. Manual runs can still be continued in the opposite direction. The error message extinguishes when the software limit switch is left.

Reference run:

In this case the hardware limit switches serve as reference switches. The software limit switches remain inactive.

Positioning run:

If the destination position is outside the range of the set software limit switches, the motor will not start and an error message will be put out (see ch. "9.3.2.9. Error messages", bit 5). This error message extinguishes automatically as soon as a valid run command is entered.
If the run command was given the module by the service module, bit 0 in Sx01.07 (CONTROL) will be cleared.
The run is stopped as soon as the destination position is reached. Byte SLx01.05 (DEST_OK) is set to $FF. This operation can be analysed as feedback message by the user program.
As long as there is no job buffer overflow, i.e. as long as byte SLx01.06 (BUF_FUL) is set to $00, you can give the next run command already.

Hardware limit switches

Upon reaching a limit switch the stepper motor software triggers an Emergency Stop. The motor stops without ramp. An error message is put out.

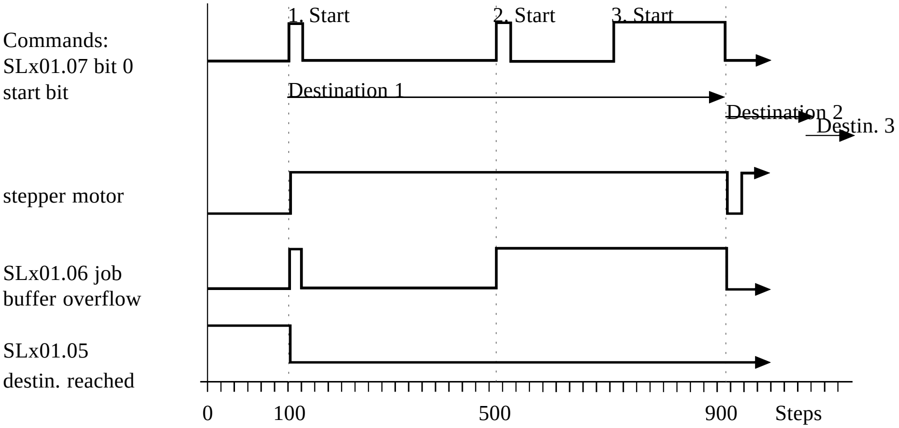
Run without hardware or software limit switches:

For operation without limit switches, the hardware and software limit switches can be deactivated via the software. This is done by clearing the corresponding bits in control byte SLx01.07 (CONTROL).

9.3.2.8. Timing diagrams of run jobs

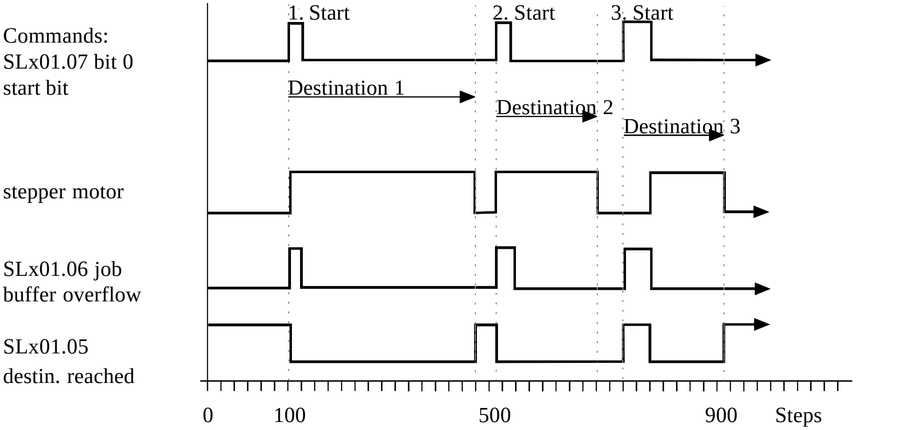
Exampl #1

Restarting several run jobs before reaching the destination position. Flag "job buffer overflow" shows that it is possible to give a new run command.



Example #2

Restarting a run job after the destination has been reached. "Destination position reached" (SLx01.05) shows that the destination position has been reached. "Destination position reached" must be reset by the user program.



9.3.2.9. Control functions SLx01.07 (CONTROL)

The command is enabled by setting the corresponding bit to 1. The stepper motor software acknowledges by writing 0:

Bit	Function
0	Job transfer to the module: Transfers a run job of the selected mode of operation and considering the set ramp functions. The mode of operation must have been set previously in mode byte SLx01.04 (MODE). The bit is cleared after the run job has been transferred to the module.
1	Stop: Stops the current run under consideration of the ramp function set.
2	Emergency Stop: Immediately stops the current run without ramp function.
3	Transfer parameters: Transfers the parameters for start/stop ramp, start/stop frequency and travelling frequency to the module. The bit is cleared by the service module after the run job has been transferred to the module.
4	Stepper motor enable: Enables the release signal, output "enable".
5	Copy preset value SLx01.00 (SV_LW) into actual position SLx00.04 (RP_LW) (4 byte). After the operation, the service module resets the bit.
6	Enable hardware limit switch: Enables the function of the connected hardware limit switches. Reaching these limit switches immediately stops the run without ramp function. Note: during reference runs, the hardware limit switches work as reference switches. This function is active during every reference run, independent of the status of the enable signal.
7	Software limit switch enable: Transfers the software limit switch values to the module and enables the function of the set software limit switches. Reaching these limit switches stops the run under consideration of the ramp function. Early recognition guarantees the stop at an exact position despite the ramp.

To set control bits

Using an OR operation to set individual bits in a byte by a binary value where only the desired bit takes on status "1". All bits stating "0" remain unchanged.

Example "Transfer run job":

```
          L      SLx01.07      ;load control byte
          O      %00000001     ;set bit 0
          =      SLx01.07      ;write into control byte
```

To reset control bits

Using an AND operation to reset individual bits in a byte by a binary value where only the desired bit takes on status "0". All bits stating "1" remain unchanged.

Example "Disable hardware limit switch":

```
          L      SLx01.07      ;load control byte
          A      %110111111    ;reset bit 6
          =      SLx01.07      ;write into control byte
```

To scan control bits

Using an AND operation to register the status of an individual bit in a byte where only the desired bit is set to "1". The result is evaluated by a jump command.

Example "Check whether preset value accepted":

```
          L      SLx01.07      ;load control byte
          A      %00100000     ;mask bit 6
          JP=    R_OVER        ;jump if bit reset
;here the evaluation for "preset value accepted" is done
          .
          .
          .
R_OVER    NOP                  ;return to normal prog. run
```

9.3.2.10. Error messages

The following errors may occur:

Error		Function
Bit	Number	
0	1	hardware limit switch left reached
1	2	hardware limit switch right reached
2	3	software limit switch left reached
3	4	software limit switch right reached
4	5	no "ready" message from motor
5	6	illegal run command
6	7	short-circuit or overload at output **display for both channels at channel 0**
7	8	module failure (watchdog)

Error byte SLx01.04 (ERR)

The user program can use this byte to find any occurring errors. Each bit (see table above) stands for an error message. The message is valid if the bit is set to 1.

Example for analysis:

To facilitate message analysis we recommend to map the contents of the byte in 8 bit markers:

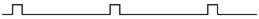
```
L       SLx01.04   ;load error byte
C8T1    M00.00     ; copy to M00.00...07
```

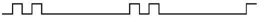
Bit 0 (error1) is now in M00.00, bit 1 in M00.01 etc.

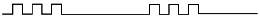
LED "ERR"

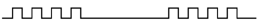
This red light emitting diode is located on top of the module. In case of any occurring errors the LED flashes in a rhythm that reflects the error number (see table above):

No. Rhythm of flashes

1 

2 

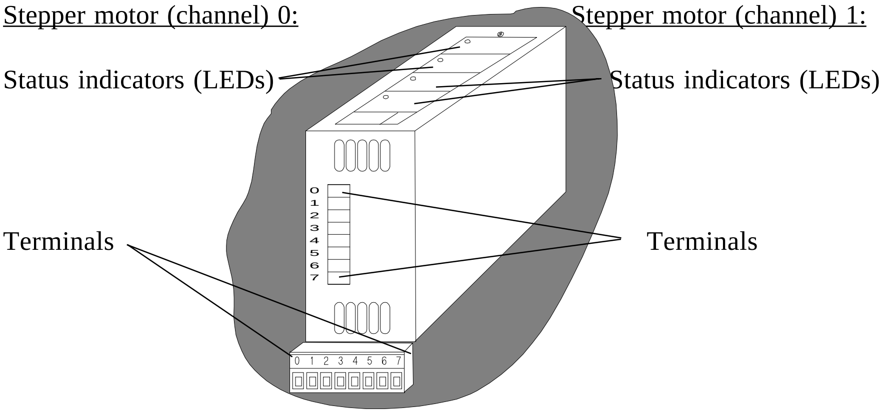
3 

4 

etc.

The LED flashes in fast impulses (250/250ms). Then there is an interval of 1s and the flashing starts again.

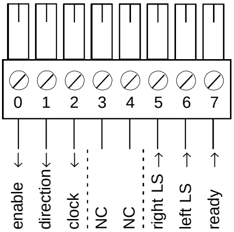
9.4. Stepper motor module, 2 channels

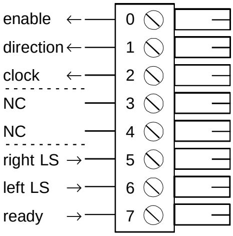


9.4.1. Connection of the signal lines

The input and output signals are connected to the module by screw-type locking terminals underneath (channel 0) the module and on (channel 1) the module.

Channel 0: Channel 1:





Connector	*Function*
enable	motor enable (to the motor)
direction	direction of rotation (to the motor)
clock	clock pulse (to the motor)
NC	unused
right LS	right limit switch (n.c. switch)
left LS	left limit switch (n.c. switch)
ready	motor ready message (n.o. switch)

9.5.2. Technical specifications

Application:	KUAX 680I, 680C, KDT 680CT
slot	optional
Admissible ambient conditions	
storage temperature	-25...+70 °C
ambient temp. during operation	0...55 °C
relative humidity	50...95 %
Number of stepper motors (channels)	2
Travelling frequency	15 kHz max.
Counting depth	31 bit plus sign bit
Inputs:	3 per channel, terminals 5, 6, 7
Type (in acc. with IEC 1131)	1
Potential separation	no
Input voltage:	24 V DC -20 % / +25 % (incl. 5 % residual ripple)
Signal identification	
logical 0:	≤ 5 V DC
logical 1:	≥ 15 V DC
Max. voltage:	28.8 V DC
Power consumption / input:	10 mA max.
Outputs:	3 per channel, terminals 1, 2, 3
Output voltage:	24 V DC -20 % / +25 % (incl. residual ripple)
Output current:	40 mA max.
Short-circuit protection:	yes
Status indicators:	6 LEDs, class 1 light emitting diodes (in acc. with EN 60825-1)

Function	*Designation*	*Colour*
reference point reached	REF	green
error message	ERR	red
motor ready message	STAT	green

Power consumption of the module	
5 V system voltage	250 mA max.
24 V power supply (output)	250 mA max. (depending on load on the outputs)
Weight:	c 128 g
Part number:	680.444.03

9.5. Example program

On the following pages you will find a program documentation for channel 0, called x-axis.

```
========  KUBES  =====================================================
                         Project structure
Project : SM680BE               Network  :
                                      created : Aug 30 1993 11:32
User    : Paul Posi                   changed : Jun 27 1996 10:06
Comment : SM module with up for KUAX 680.  680.454.06

=======================================================================
ORG.ORG/1
|
*——>PARAME.INI/1
|
*——>COMMNDX1.PRO/6
|       |
|       *——>AUTOPRO.PRO/2
|
*——>COMMNDX2.PRO/5
```

```
========  KUBES  =====================================================
                         Organisation module IL
Project     : SM680BE               Network  :
Module      : ORG       No.: 1      created : Aug 30 1993 11:33
User        : Paul Posi             changed : Jun 27 1996 09:59
Comment     : Initialisation and main program

=======================================================================
 ; ***************************************************
 ;         Org module with:
 ;         1. SM module initialisation
 ;         2. jumper strip to the main programs
 ;            only for the x-axis.
 ; ***************************************************

         L       INIT_OK         M00.00 ; (initialisation OK)
         JPC     RUN
         JPINIT  PARAME              1
         JP      END

RUN      JPP     COMMNDX1            6
         JPP     COMMNDX2            5
END      NOP

; ──── Status message ───────

; ── Step counter
         LD      RP_LWX       SLA00.04 ; (X-axis SM act. pos.)
         LD      RP_HWX       SLA00.06 ; (X-axis SM act. pos.)

; ── Error messages
         L       ERRX         SLA01.04 ; (error code X-axis)

; ── Software version
         L       SW_V_X       SLA01.15 ; (software version X-axis)


```

```
========  KUBES  ====================================================
                        Program module  IL
Project     : SM680BE              Network  :
Module      : AUTOPRO   No.: 2       created : Aug 31 1993 14:31
User        : Paul Posi              changed : Jun 27 1996 09:58
Comment     : Step chain for the positioning program

======================================================================
; **************************************************
;        Destination positions for the auto program
;        7 set destinations
; **************************************************

STEP_0   L       X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
         CMP       0
         JP<>    STEP_1
         LD      1000
         =D      DP_LWX        SLA00.00 ; (SM set pos. X-axis)
         LD      $0000
         =D      DP_HWX        SLA00.02 ; (SM set pos. X-axis)
         INC     X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
         JP      END

STEP_1   L       X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
         CMP       1
         JP<>    STEP_2
         LD      2000
         =D      DP_LWX        SLA00.00 ; (SM set pos. X-axis)
         LD      0
         =D      DP_HWX        SLA00.02 ; (SM set pos. X-axis)
         INC     X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
         JP      END

STEP_2   L       X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
         CMP       2
         JP<>    STEP_3
         LD      3000
         =D      DP_LWX        SLA00.00 ; (SM set pos. X-axis)
         LD      0
         =D      DP_HWX        SLA00.02 ; (SM set pos. X-axis)
         INC     X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
         JP      END

```

```
STEP_3    L       X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
          CMP       3
          JP<>    STEP_4
          LD      4000
          =D      DP_LWX        SLA00.00 ; (SM set pos. X-axis)
          LD      $0000
          =D      DP_HWX        SLA00.02 ; (SM set pos. X-axis)
          INC     X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
          JP      END

STEP_4    L       X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
          CMP       4
          JP<>    STEP_5
          LD      5000
          =D      DP_LWX        SLA00.00 ; (SM set pos. X-axis)
          LD      $0000
          =D      DP_HWX        SLA00.02 ; (SM set pos. X-axis)
          INC     X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
          JP      END

STEP_5    L       X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
          CMP       5
          JP<>    STEP_6
          LD      6000
          =D      DP_LWX        SLA00.00 ; (SM set pos. X-axis)
          LD      $0000
          =D      DP_HWX        SLA00.02 ; (SM set pos. X-axis)
          INC     X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
          JP      END

STEP_6    L       X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
          CMP       6
          JP<>    END
          LD      7000
          =D      DP_LWX        SLA00.00 ; (SM set pos. X-axis)
          LD      $0000
          =D      DP_HWX        SLA00.02 ; (SM set pos. X-axis)
          CLR     X_STEP_C      SBM15.00 ; (step cntr x-auto prg.)
END      NOP

```

```
========  KUBES  =====================================================
                     Program module  IL
Project : SM680BE                Network  :
Module  : COMMNDX2     No.: 5    created : Apr 22 1996 09:26
User    : Paul Posi              changed : Jun 27 1996 09:58
Comment : Execute SM functions for X-axis, part 2

======================================================================
; ********************************************
;        Commands for the X-axis, part 2
;        - positive manual run
;        - negative manual run
; *********************************************

; ––– Positive manual run, start –––
          L       X_H_NEG         I00.05 ; (X-axis neg. man. run)
          =       PP00_05        PP00.05 ; (progr. pulse)
          L       PP00_05        PP00.05 ; (progr. pulse)
          JPCN    HRE_STP

          LD      50
          =D      RAMPX         SLA01.08 ; (ramp value X-axis)

          LD      2500                   ; Hz
          =D      MOV_FRQX      SLA01.12 ; (posi speed X-axis)

          L       3
          =       MODEX         SLA01.14 ; (oper. mode X-axis)

          L       CONTROLX      SLA01.07 ; (command X-axis)
          O       %1000_1001
          =       CONTROLX      SLA01.07 ; (command X-axis)
          JP      HRE_END

; ––– Positive manual run, stop –––
HRE_STP   LN      X_H_NEG         I00.05 ; (X-axis neg. man. run)
          =       PP00_06        PP00.06 ; (progr. pulse)
          L       PP00_06        PP00.06 ; (progr. pulse)
          JPCN    HRE_END

          L       CONTROLX      SLA01.07 ; (command X-axis)
          O       %1000_0010
          =       CONTROLX      SLA01.07 ; (command X-axis)
HRE_END   NOP

```

```
; —— Negative manual run, start ——
         L       X_H_POS        I00.06 ; (pos. man. run X-axis)
         =       PP00_07       PP00.07 ; (progr. pulse)
         L       PP00_07       PP00.07 ; (progr. pulse)
         JPCN    HLI_STP
         LD      50
         =D      RAMPX        SLA01.08 ; (ramp value X-axis)

         LD      3000                  ; Hz
         =D      MOV_FRQX     SLA01.12 ; (posi speed X-axis)

         L       2
         =       MODEX        SLA01.14 ; (oper. mode X-axis)

         L       CONTROLX     SLA01.07 ; (command X-axis)
         O       %1000_1001
         =       CONTROLX     SLA01.07 ; (command X-axis)

; —— Negative manual run, start ——
HLI_STP  LN      X_H_POS        I00.06 ; (pos. man. run X-axis)
         =       PP00_08       PP00.08 ; (progr. pulse)
         L       PP00_08       PP00.08 ; (progr. pulse)
         JPCN    HLI_END

         L       CONTROLX     SLA01.07 ; (command X-axis)
         O       %1000_0010
         =       CONTROLX     SLA01.07 ; (command X-axis)
HLI_END  NOP
```

```
========  KUBES  =====================================================
                        Program module  IL
Project : SM680BE               Network  :
Module  : COMMNDX1    No.: 6     created : Aug 30 1993 15:09
User    : Paul Posi              changed : Jun 27 1996 10:01
Comment : Execute SM functions for X-axis, part 1

======================================================================
 ; *********************************************
 ;        Commands for the X-axis, part 1
 ;        - reference run
 ;        - positioning step
 ;        - automatic mode
 ;        - stop command
 ;        - Emergency Stop command
 ;        - status messages
 ;        - new step counter value
 ; *********************************************

 ; —— Start reference run ——
           L       X_REF            I00.00 ; (start ref. run X-axis)
           =       PP00_00         PP00.00 ; (progr. pulse)
           L       PP00_00         PP00.00 ; (progr. pulse)
           JPCN    REF_END

           LD      700                     ; steps
           =D      ST_ST_FX       SLA01.10 ; (start/stop fr. X-axis)
           LD      1000
           =D      RAMPX          SLA01.08 ; (ramp value X-axis)
           LD      2000                    ; Hz
           =D      MOV_FRQX       SLA01.12 ; (posi speed X-axis)
           L       4                       ; Ref.run,4=le, 5=ri
           =       MODEX          SLA01.14 ; (oper. mode X-axis)

           L       CONTROLX       SLA01.07 ; (command X-axis)
           O       %0000_1001
           A       %0111_1111              ; SE analysis
           =       CONTROLX       SLA01.07 ; (command X-axis)
           L       0
           =       DEST_OKX       SLA01.05 ; (destin. X-axis reached)
           CLR     X_STEP_C       SBM15.00 ; (step cntr X-autopgrg)

 REF_END   NOP

```

```
; —— Start positioning step –
         L       X_START           I00.02 ; (start X-axis)
         A       DEST_OKX        SLA01.05 ; (destin. X-axis reached)
         =       PP00_01          PP00.01 ; (progr. pulse)
         L       PP00_01          PP00.01 ; (progr. pulse)
         JPCN    STA_END

         LD      900
         =D      ST_ST_FX        SLA01.10 ; (start/stop fr. X-axis)
         LD      1000
         =D      RAMPX           SLA01.08 ; (ramp value X-axis)
         LD      8000                     ; Hz
         =D      MOV_FRQX        SLA01.12 ; (posi speed X-axis)
         LD      15000
         =D      DP_LWX          SLA00.00 ; (set SM pos. X-axis)
         LD      $0000
         =D      DP_HWX          SLA00.02 ; (set SM pos. X-axis)

         L       8                        ; positioning step
         =       MODEX           SLA01.14 ; (oper. mode X-axis)
         L       CONTROLX        SLA01.07 ; (command X-axis)
         O       %1000_1001
         =       CONTROLX        SLA01.07 ; (command X-axis)
         CLR     DEST_OKX        SLA01.05 ; (destin. X-axis reached)

STA_END  NOP

; —— Start auto-posi program ———
;        Seven destination positions set
;        Attention:  note special mode 18
;                    Refer to Instruction Manual
; ——————————————————————

         L       X_AUTO            I00.01 ; (start autoprog. X-axis)
         AN      BUF_FULX        SLA01.06 ; (buffer overflow X-axis)
         A       DEST_OKX        SLA01.05 ; (X-axis destin. reached)
         JPCN    A_END

         LD      900
         =D      ST_ST_FX        SLA01.10 ; (start/stop fr. X-axis)
         LD      2000
         =D      RAMPX           SLA01.08 ; (ramp value X-axis)
         LD      7000                     ; Hz
         =D      MOV_FRQX        SLA01.12 ; (posi speed X-axis)

         JPP     AUTOPRO                2
         L       8                        ; positioning step
         =       MODEX           SLA01.14 ; (oper. mode X-axis)

```

```
          L       CONTROLX      SLA01.07 ; (command X-axis)
          O       %1000_1001
          =       CONTROLX      SLA01.07 ; (command X-axis)
          CLR     DEST_OKX      SLA01.05 ; (X-axis destin. reached)
A_END     NOP

; ——— Trigger stop (with ramps) ——
          L       X_STOP           I00.03 ; (stop X-axis)
          =       PP00_03         PP00.03 ; (progr. pulse)112: NST_END  NOP
          L       PP00_03         PP00.03 ; (progr. pulse)
          JPCN    STP_END

          L       CONTROLX      SLA01.07 ; (command X-axis)
          O       %0000_0010
          =       CONTROLX      SLA01.07 ; (command X-axis)
STP_END   NOP

; ——— Trigger Emergency Stop (without ramps)–
          L       X_NSTOP          I00.04 ; (Emerg. Stop X-axis)
          =       PP00_04         PP00.04 ; (progr. pulse)
          L       PP00_04         PP00.04 ; (progr. pulse)
          JPCN    NST_END

          L       CONTROLX      SLA01.07 ; (command X-axis)
          O       %0000_0100

          =       CONTROLX      SLA01.07 ; (command X-axis)113:
; ———— New step counter value ——
          L       X_VW_UE          I00.07 ; (preset X-axis)
          =       PP00_09         PP00.09 ; (progr. pulse)
          L       PP00_09         PP00.09 ; (progr. pulse)
          JPCN    UEN_END

          LD      2800
          =D      SV_LWX        SLA01.00 ; (preset step X-axis)
          LD      0
          =D      SV_HWX        SLA01.02 ; (preset step X-axis)

          L       CONTROLX      SLA01.07 ; (command X-axis)
          O       %0010_0000
          =       CONTROLX      SLA01.07 ; (command X-axis)
UEN_END   NOP


```

```
========  KUBES  ====================================================
                      Init. module  IL
Project : SM680BE              Network  :
Module  : PARAME      No.: 1     created : Aug 30 1993 14:51
User    : Paul Posi              changed : Jun 27 1996 10:00
Comment : X- and Y-axis parameters

=====================================================================
; +++++++++++++++++++++++++++++++++++++++++++
;      Motor parameters for X-axis
; +++++++++++++++++++++++++++++++++++++++++++
; ——— Negative software limit switch ———
SLSL_LWX   SLA00.08    WORD    $FF00    ; (X softw. limit sw. left)
SLSL_HWX   SLA00.10    WORD    $FFFF    ; (X softw. limit sw. left)

; ——— Positive software limit switch ———
SLSR_LWX   SLA00.12    WORD    $FFFF    ; (X softw. limit sw. right)
SLSR_HWX   SLA00.14    WORD    $0000    ; (X softw. limit sw. right)

; ——— Preset step counter value ————
SV_LWX     SLA01.00    WORD    0        ; (preset step cntr X-axis)
SV_HWX     SLA01.02    WORD    0        ; (preset step cntr X-axis)

; ——— Start and stop ramps —————————
RAMPX      SLA01.08    WORD    3000     ; (ramp value X-axis)

; ——— Start and stop frequency ————————
ST_ST_FX   SLA01.10    WORD    700      ; (start/stop fr. X-axis)

; ——— Preset destination position reached ————
DEST_OKX   SLA01.05    BYTE    255      ; (destin. reached X-axis

; ———— Set command byte ————
CONTROLX   SLA01.07    BYTE    %0101_1000 ; (command X-axis)

; Bit 0 = start
; Bit 1 = stop
; Bit 2 = Emergency Stop
; Bit 3 = accept parameters
; Bit 4 = enable
; Bit 5 = accept preset value
; Bit 6 = hardware limit switch enable
; Bit 7 = software limit switch enable
```

A. Power consumption of the modules

The controller is supplied with voltage at two points.

A.1. Power supply

A.1.1. 24 V DC system supply

The externally provided 24 V for the system supply are transformed in the device:

- System voltage: 5 V DC
  supplies the controller and the modules
  Maximum load by the modules:

• KUAX 680I (part no. 680.420...422.xx)	350 mA
• KUAX 680I (as from part no.680.423...nnn.xx)	1000mA
• KUAX 680C	1200mA
• KDT 680CT	1200mA

- Supply of the analogue outputs: 15 V DC
  only supplies the analogue outputs

A.1.2. Supply of digital outputs and inputs

The digital outputs and inputs are separately supplied with 24 V DC. Each digital output module has a relay as a means of protection against reversed polarity. This relay is supplied with the same voltage.

Voltage is fed to the device via screw-type locking connectors. Please note that no more current must go through these connectors than is indicated in the table below:

- KUAX 680I, 4 slots	8 A max.
- KUAX 680I, 8 slots	2 connectors of 8 A max.
- KUAX 680C	12 A max.
- KDT 680CT	12 A max.

A.2. Load on the power supply by the modules

Each module puts additional load on the voltage supply/supplies:

Modules		Power [mA] source:		
Type	Part no.	Syst. supply 5 V	 24 V	24 V supply of outputs
Digital input and output modules				
8 inputs	680.451.01/04	0.1	0.05	-
ditto with real-time clock	680.451.02	25	8	
16 inputs	680.451.03/07 680.451.06	0.3 60	0.1 20	
8 outputs, 24 V	680.452.01	0.1	0.05	base load: 80 plus max. 500 per O
8 inputs/ 8 outputs	680.450.01	0.2	0.1	
4 pneumatic outputs	680.453.01	20	10	600
Analogue input and output modules				
inputs 0...10 V	680.441.01	6	15	-
inputs 0...20 mA	680.441.02	6	15	
inputs PT100, 0...300 °C	680.441.04	12	30	
inputs thermo. 1200 °C	680.441.07	18	20	
potentiometer inputs	680.441.05	2.5	25	
outputs 0...10 V	680.442.01	3.0	55*1)	
outputs 0(4)...20 mA	680.442.02	3.0	95	
I/O 0...10 V / 0...±10 V	680.441.03	2	60	
I/O 0(4)...20 mA / 0...±10 V	680.441.06	2	60	
I/O 0...10 V / 0(4)...20 mA	680.441.08	10	75	
I/O 0(4)...20 mA / 0(4)...20 mA	680.441.09	10	75	
Counter modules				
1 multi-function counter, 24 bit	680.454.01	46	12	-
2 multi-function counters, 24 bit	680.454.02	46	12	
2 event counter, 16 bit	680.454.03	15	3.5	
2 SSI generators, 24 bit	680.454.04	250	depending on gen. used	
Communication modules				
V.24 module	680.440.01	35	12	-
TTY module	680.440.02	35	12	
RS 485 module	680.440.03	280	-	
Positioning modules				
1 stepper motor	680.444.01	10	2.5	max. 80 per output
2 stepper motors	680.444.02	10	2.5	
2 stepper motors (module with processor)	680.444.03	250	2.5	max. 40 per output

*1) Base load: 15 mA plus 10 mA max. per output

The load by the modules on the 5 V system voltage must not exceed a certain value (see appendix "A.1.1. 24 V DC system supply").

B. Order specifications

Digital input modules

Input module, 24 V DC, 8 inputs	680.451.01
Input module, 24 V DC, 8 inputs, 1 ms	680.451.04
Input module, 24 V DC, 8 inputs, with real-time clock	680.451.02
Input module, 24 V DC, 16 inputs	680.451.03
Input module, 24 V DC, 16 inputs, with interrupt capability	680.451.06
Input module, 24 V DC, 16 inputs, 1 ms	680.451.07

Digital output modules

Output module, 24 V DC, 0.5 A, 8 outputs	680.452.01
Pneumatic output module, 4 outputs 3/2-way	680.453.01

Digital input / output modules

Input / output module, 24 V DC, 8 inputs, 8 outputs	680.450.01

Analogue input modules

Analogue input module, 0...10 V, 10 bit, 4 channels	680.441.01
Analogue input module, 0(4)...20 mA, 10 bit, 4 channels	680.441.02
Analogue input module, PT100, 0...300 °C, 10 bit, 4 channels	680.441.04
Analogue input module, thermocouple Ni-Cr-Ni (type K) 0...1200 °C, 10 bit, 4 channels	680.441.07
Analogue input module, potentiometer, 10 bit, 4 channels	680.441.05

Analogue output modules

Analogue output module, 0...10 V, 8 bit, 4 channels	680.442.01
Analogue output module, 0(4)...20 mA, 8 bit, 4 channels	680.442.02

Analogue input / output modules

Analogue input/output module, 2 I 0...10 V, 2 O 0...± 10 V	680.441.03
Analogue input/output module, 2 I 0...20 mA, 2 O 0...±10 V	680.441.06
Analogue input/output module, 2 I 0...10 V, 2 O 0...20 mA	680.441.09
Analogue input/output module, 2 I 0...20 mA, 2 O 0...20 mA	680.441.09

Order specifications

Counter modules

Counter module, 1 multi-function counter, 24 V, 24 bit	680.454.01
Counter module, 2 multi-function counters, 24 V, 24 bit	680.454.02
Counter module, 2 multi-function counters, RS 422, 24 bit	680.454.08
Counter module, 2 event counters, 24 V, 16 bit	680.454.03
SSI module, 2 generator connectors, 24 bit	680.454.04

Communication modules

V.24 module	680.440.01
TTY module	680.440.02
RS 485 module	680.440.03
SE_680I and RS485 communication programs	680.505.01
PROFIBUS-DP slave module	680.440.05

Stepper motor modules

without processor, 1 channel	680.444.01
without processor, 2 channels	680.444.02
with processor, 2 channels	680.444.03

Positioning modules

Complete positioning module, 1 channel	680.454.06
Servo counter module, 2 channels, 24 V, 24 bit	680.454.05

Accessories

8pin screw-type locking connnectors for digital I/Os (12 pcs.)	680.180.02
Dummy casing for covering unused slots	680.180.09
Simulator box (4 x 8pin) for digital inputs	680.155.50
Labels for modules and devices (1 A4 sheet)	680.180.04

C. Literature list

Instruction manual E 308 GB
KUAX 680I Profi Control
Controller with module slots and PROFIBUS connector
Kuhnke GmbH, Malente

Instruction manual E 399 GB
KUAX 680C Compact Control
Controller with built-in I/Os and module slots
Kuhnke GmbH, Malente

Instruction manual E 414 GB
KDT 680CT Control Terminal
Terminal and controller in one
Kuhnke GmbH, Malente

Instruction manual E 509 GB
PROFIBUS Modules for the 680 System
Kuhnke GmbH, Malente

Instruction manual E 334 GB
SE_680I and RS485
Communication programs for V.24, TTL or RS485 modules
Kuhnke GmbH, Malente

Instruction manual E 327 GB
KUBES 4
Kuhnke User Software
User interface for programming, testing and documenting user programs in Kuhnke controllers
Kuhnke GmbH, Malente

Modules of the 680 Series, Weights 15/8/96

Part number	Weigth
680.440.01	89 g
680.440.02	86 g
680.440.03	89 g
680.440.04	127 g
680.440.05	127 g
680.441.01	77 g
680.441.02	79 g
680.441.03	98 g
680.441.04	78 g
680.441.05	78 g
680.441.06	98 g
680.441.07	93 g
680.441.08	90 g
680.441.09	90 g
680.441.89	90 g
680.442.01	85 g
680.442.02	86 g
680.444.01	82 g
680.444.02	116 g
680.444.03	128 g
680.450.01	117 g
680.451.01	77 g
680.451.02	85 g
680.451.03	104 g
680.451.04	77 g
680.451.06	108 g
680.452.01	84 g
680.453.01	198 g
680.453.03	76 g
680.454.01	78 g
680.454.02	102 g
680.454.03	75 g
680.454.04	123 g
680.454.05	106 g
680.454.06	110 g
680.454.08	112 g

Index