KL-6021

Technical Documentation
Serial Interface
RS485 (RS422)

Please keep for further use !

Edition date/Rev. date:	15.05.1998
Document no./Rev. no.:	TRS - V - BA - GB - 0117 - 00
Software version:	1.0
File name:	TRS-V-BA-GB-0117.DOC
Author:	KOH

TRSystemtechnik GmbH
Eglishalde 6
D-78647 Trossingen
Germany
Tel. +49 - (0) 7425 / 228-0
Fax +49 - (0) 7425 / 228-34

Imprint

TRSystemtechnik GmbH
D-78647 Trossingen
Eglishalde 6
Tel.: (++49) 07425/228-0
Fax: (++49) 07425/228-34

© Copyright 1998 TRSystemtechnik

Guarantee

In our ongoing efforts to improve our products, TRSystemtechnik reserve the right to alter the information contained in this document without prior notice.

Printing

This manual was edited using text formatting software on a DOS personal computer. The text was printed in *Arial*.

Fonts

Italics and **bold** type are used for the title of a document or to emphasize text passages.

Passages written in `Courier` show text which is visible on the display as well as software menu selections.

"< >" refers to keys on your computer keyboard (e.g. <RETURN>).

Note

Text following the "NOTE" symbol describes important features of the respective product.

Copyright Information ©

MS-DOS is a registered trademark of Microsoft Corporation.

Revision History

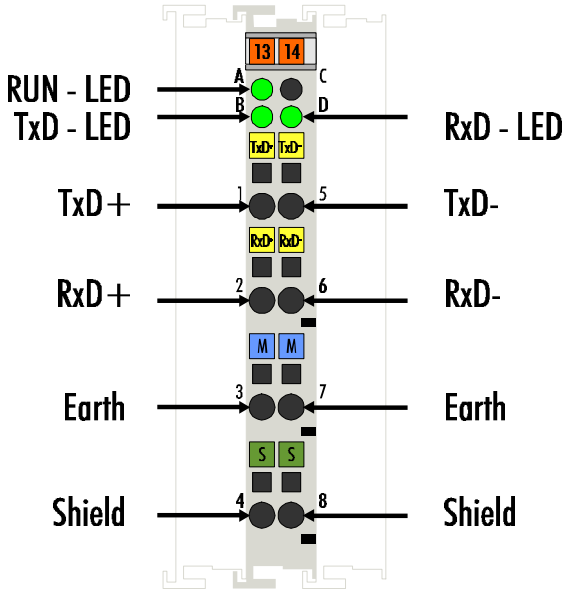
Note:
The cover of this document shows the current revision status and the corresponding date. Since each individual page has its own revision status and date in the footer, there may be different revision statuses within the document.

Document created: 15.05.1998

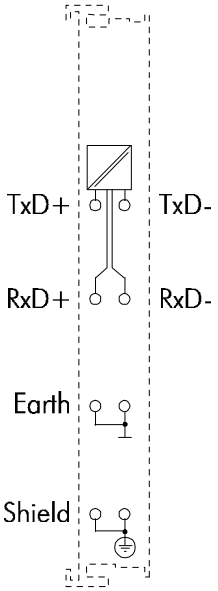
Revision	Date

Table of contents

Serial Interface RS485 (RS422) KL6021



Top View



Contact Assembly

Technical data	KL6021
Transmission channels	TxD and RxD, full duplex (half duplex)
Data transfer rate	9600 Baud (8N1) preset, (max. 19200 Baud)
Bit transfer	with difference signal
Line impedance	120 Ω
Transmission link	approximately 1000m twisted pair
Power supply	via the K-bus
K-bus current consumption	65 mA type
Electrical isolation	500 V rms (K-bus / signal voltage)
Data buffer	128 bytes receive buffer, 16 bytes send buffer
Bit width in the process image	I/O:3 x 8 bits user data, 1 x 8 bits status (up to 5 x 8 bits user data possible)
Configuration	no address setting, configuration setting via the bus coupler
Weight approximately	60 g
Operating temperature	0°C ... +55°C
Storage temperature	-25°C ... +85°C
Relative humidity	95%, no condensation
Vibrations/shock resistance	In accordance with IEC 68-2-6 / IEC 68-2-27
EMC resistance Burst / ESD	In accordance EN 61000-4-4 / EN 61000-4-2 Limits in accordance with EN 50082-2
Installation position	any
Degree of protection	IP20

Description of functions

The serial interface terminal KL6021 enables the connection of devices featuring an RS485 interface. Regardless of the higher-level bus system, data can be exchanged with the controller in full / half duplex mode. The receive buffer is 128 bytes large, while the send buffer embraces 16 bytes. Data transfer between the terminal and controller is handled via a handshake in the status and control byte. The terminal's works setting is 9600 baud, 8 data bits, 1 stop bit, no parity, full duplex mode.

LED Display

The Run LED indicates the operating state of the terminal.
On – normal operation

Off - watchdog timer overflow has occurred. The green LED goes off if no process data is transferred from the bus coupler for 100 ms.

The TxD and RxD LEDs indicate the states of the signal lines.

Process data
Alternative output format

In the alternative output format, 4 or 5 bytes (3 bytes of data and 1 byte or 2 bytes of control / status byte) are mapped in the bus coupler. When delivered, the KL6021 is set to the alternative format. Mapping of the terminal in the alternative format is described in further detail in the chapter entitled "Terminal Configuration".

Standard output format

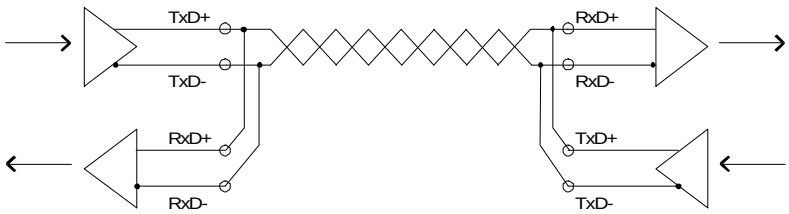
By default, in the standard output format 4 bytes (3 bytes of user data and 1 control / status byte) are mapped in the bus coupler. Up to 5 bytes of user data can be transferred by redefining the parameters of the KL6021.

Reference

The annex contains an over view of possible mapping configurations depending on the parameters that can be set.

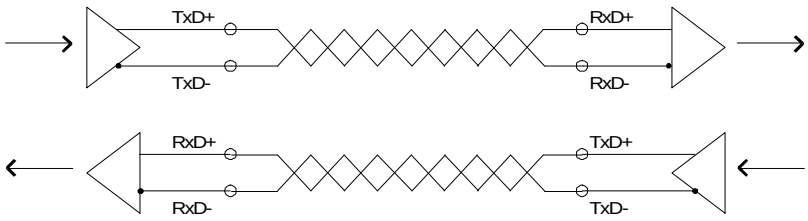
Connection for RS485 transfer

In the RS485 mode, the data is exchanged by means of half duplex transfer. A bus structure can be created in this mode of operation.



Connection in the case of RS422 transfer

In the RS422 mode, data is transferred in full duplex mode. Only peer-to-peer connections can be established.

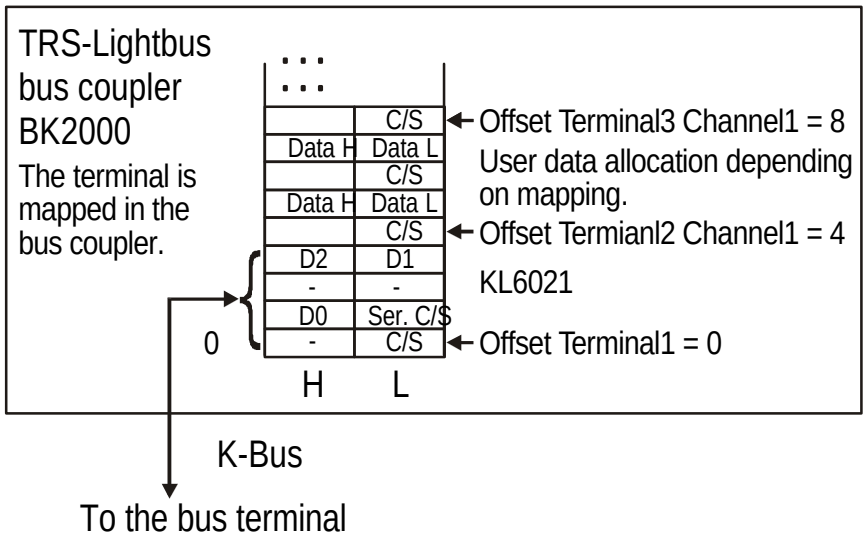


Terminal configuration

The terminal can be configured and parametrized via the internal register structure.
Each terminal channel is mapped in the bus coupler. The data of the terminal is mapped differently in the memory of the bus coupler depending on the type of the bus coupler and on the set mapping configuration (eg Motorola / Intel format, word alignment.
For parametrization of a terminal, the control / status byte must also be mapped.

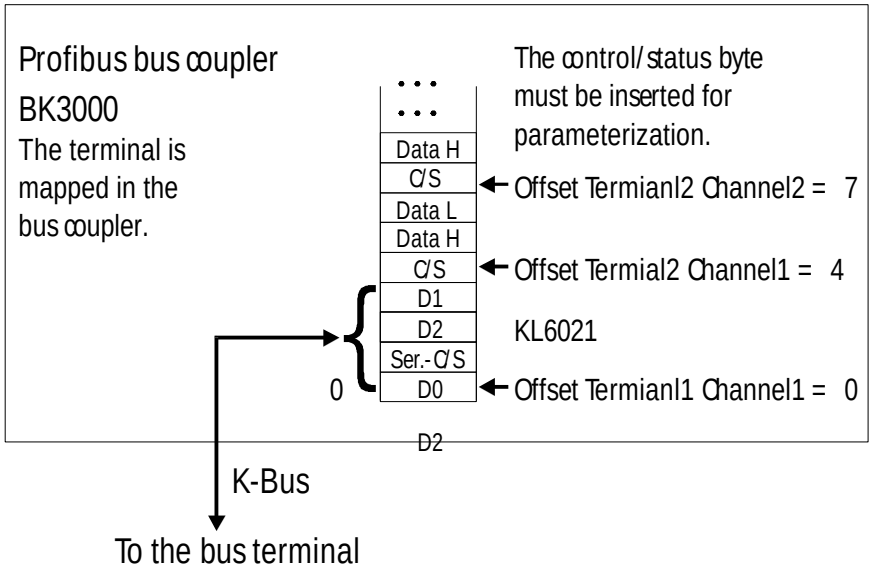
TRS Lightbus coupler BK2000

When using the TRS Lightbus coupler BK2000, the control / status byte is always mapped in additon to the data bytes. It is always in the low byte at the offset address of the terminal channel. In the case of the KL6021 the C/S byte is only used in the register mode. The serial C/S byte is used for the protocol.



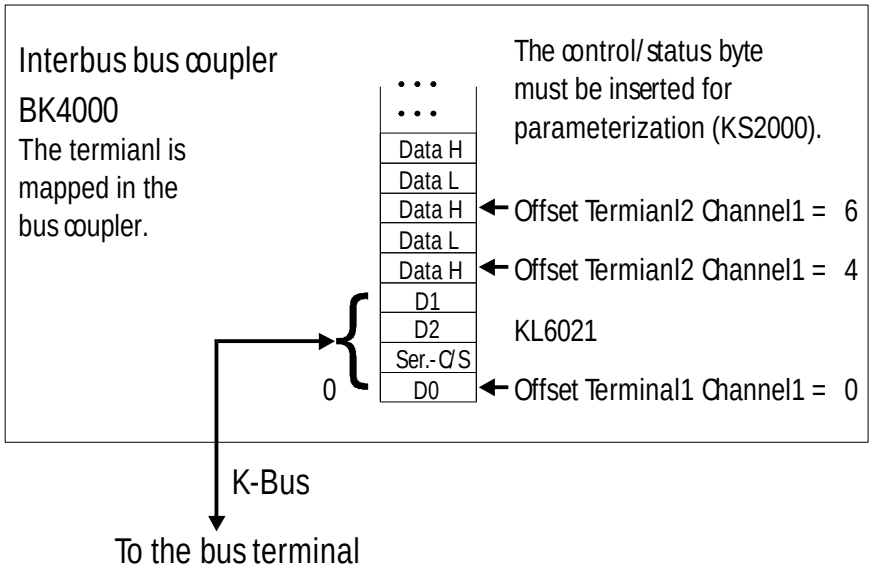
Profibus coupler BK3000

When using the Profibus coupler BK3000, how the KL6021 is to map itself in the bus coupler is set in the master configuration software. When delivered, the KL6021 is set to the alternative format. Please pay attention to the registers 34 and 35 if you wish to set the standard format and a different user data length. The figure shows the mapping for 4 bytes of input data and 4 bytes of output data.



Interbus coupler BK4000

By default, the Interbus coupler BK4000 maps the KL6021 with 4 bytes of input data and 4 bytes of output data. Parametrization via the field bus is not possible. The KS2000 software is needed to redefine the terminal's parameters.



Other bus couplers and further information

You will find further information of the mapping configuration of bus couplers in the annex of the respective bus coupler manual under the heading of "Configuration of Masters".

Parametrization with the KS2000 software

Parametrization operations can be carried out independently of the field bus system using the TRS KS2000 configuration software via the serial configuration interface in the bus coupler.

Register communication KL6021

General register description

Complex terminals that possess a processor are capable of bidirectionally exchanging data with the higher-level control system. Below, these terminals are referred to as intelligent bus terminals. They include the analog inputs, (0-10V, -10-10V, 0-20mA, 4-20mA), the analog outputs, (0-10V, -10-10V, 0-20mA, 4-20mA), serial interface terminals (RS485, RS232, TTY, data transfer terminals), counter terminals, the encoder interface, the SSI interface, the PWM terminal and all other parametrizable terminals.

Internally, all intelligent terminals possess a data structure that is identical in terms of ist essential characteristics. This data area is organized in words and embraces 64 memory locations. The essential data and parameters of the terminal can be read and adjusted by way of the structure. Function calls with corresponding parameters are also possible. Each logical channel of an intelligent terminal has such a structure (therefore, 4-channel analog terminals have 4 register sets.

This structure is broken down into the following areas:
(You will find a list of all registers at the end of this documentation.

Area	Address
Process variables	0-7
Type registers	8-15
Manufacturer parameters	16-31
User parameters	32-47
Extended user area	48-63

Process variables

R0 - R7: Registers in the terminal's internal RAM:

The process variables can be used in additional to the actual process image and their functions are specific to the terminal.

R0 - R5: These registers have a function that depends on the terminal type.

R6: Diagnostic register

The diagnostic register may contain additional diagnostic information. In the case of serial interface terminals, for example, parity errors that have occurred during data transfer are indicated.

R7: Command register

High-Byte_Write = function parameter
Low-Byte _Write = function number
High-Byte _Read = function result
Low-Byte_ Read = function number

Type registers

R8 - R15 Registers in the terminal's internal ROM der Klemme

The type and system parameters are programmed permanently by the manufacturer and can only be read by the user but cannot be modified.

R8: Terminal type:

The terminal type in register R8 is needed to identify the terminal.

R9: Softwar version X.y

The software version can be read as an ASCII character string.

R10: Data length

R10 contains the number of multiplexed shift registers and their length in bits.
The bus coupler sees this structure.

R11: Signal channels

In comparison with R10, the number of logically existing channels is located here. For example, one physically existing shift register may consist of several signal channels.

R12: Minimum data length

The respective byte contains the minimum data length of a channel to be transferred. The status byte is omitted if the MSB is set.

R13: Data type register

Data type register	
0x00	Terminal without valid data type
0x01	Byte array
0x02	1 byte n bytes structure
0x03	Word array
0x04	1 byte n words structure
0x05	Double word array
0x06	1 byte n double words structure
0x07	1 byte 1 double word structure
0x08	1 byte 1 double word structure
0x11	Byte-array with a variable logical channel length
0x12	1 byte n bytes structure with a variable logical channel length (eg 60xx)
0x13	Word-array with a variable logical channel length
0x14	1 byte n words structure with a variable logical channel length
0x15	Double word array with a variable logical channel length
0x16	1 byte n double words structure with a variable logical channel length

R14: not used

R15: Alignment bits (RAM)

The analog terminal is set to a byte limit in the terminal bus with the alignment bits.

Manufacturer parameters

R16 - R30 is the area of the "Manufacturer parameters" (SEEROM)

The manufacturer parameters are specific to each terminal type. They are programmed by the manufacturer but can also be modified from the control system. The manufacturer parameters are stored permanently in a serial EEPROM and are therefore not destroyed by power failures.
These registers can only be modified after setting a code word in R31.

User parameters

R31 - R47 "Application parameters" area (SEEROM)

The application parameters are specific to each terminal type. They can be modified by the programmer. The application parameters are stored permanently in a serial EEPROM in the terminal and cannot be destroyed by power failures. From software version 2.A, the user area is write-protected by way of a code word.

R31: Code word-register in the RAM

The code word 0x1235 must be entered here to enable modification of parameters in the user area. Write-protection is set if a different value is entered in this register. When write protection is inactive, the code word is returned during reading of the register. The register contains the value zero when write protection is active.

R34: Feature-register

This register defines the operating modes of the terminal. For example, a user-specific scaling can be activated for the analog I/O's.

R33 - R47

Registers that depend on the terminal type

Extended application area

R47 - R63

These registers have not yet been implemented.

Register access via process data transfer

Bit 7=1: register mode

When bit 7 of the control byte is set, the first two bytes of the user data are not used for process data transfer, but are written into or read out of the terminal's register.

Bit 6=0: read
Bit 6=1: write

In bit 6 of the control byte, you define whether a register is to be read or written. When bit 6 is not set, a register is read without modification. The value can be taken from the input process image.
When bit 6 is set, the user data is written into a register. The operation is concluded as soon as the status byte in the input process image has assumed the same value as the control byte in the output process image.

Bits0 to 5: address

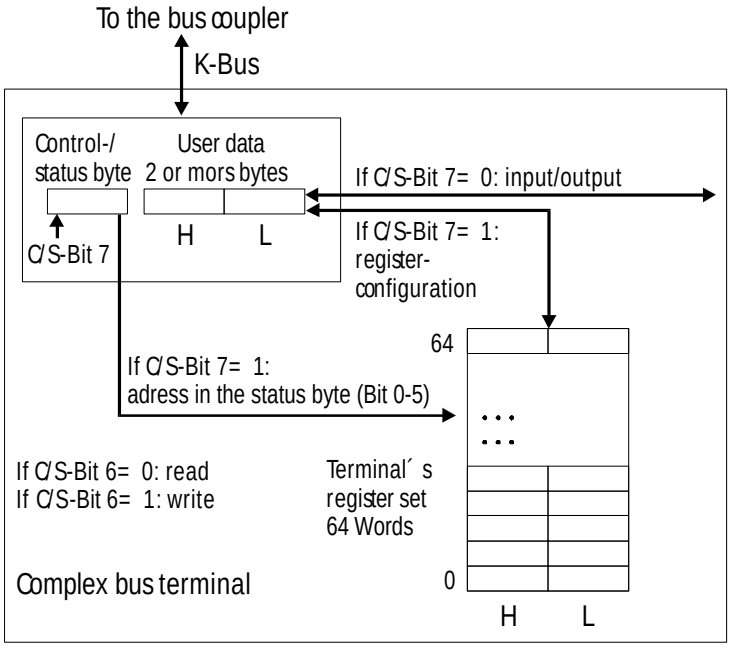
The address of the register to be addressed is entered in bits 0 to 5 of the control byte.

Control byte in the register mode

MSB

REG=1	W/NR	A5	A4	A3	A2	A1	A0

REG = 0 : Process data transfer
REG = 1 : Access to register structure
W/NR = 0 : Read register
W/NR = 1 : Write register
A5..A0 = Register address
A total of 64 registers can be addressed with the addresses A5....A0.



The control or status byte occupies the lowest address of a logical channel. The corresponding register values are located in the following 2 data bytes (the BK2000 is an exception to the rule: here, an unused data byte is inserted after the control or status byte, thus setting the register value to a word limit).

Example

Reading register 8 in the BK2000 with a Kl3022 and the end terminal.

If the following bytes are transferred from the controller to the terminal,

Byte0	Byte1	Byte2	Byte3
0x88	0xXX	0xXX	0xXX

the terminal returns the following type designation (0xBCE corresponds to the unsigned integer 3022)

Byte0	Byte1	Byte2	Byte3
0x88	0x00	0xCE	0x0B

A further example

Writing register 31 in the BK2000 with an intelligent terminal and end the end terminal.

If the following bytes (user code word are transferred from the controller to the terminal,

Byte0	Byte1	Byte2	Byte3
0xDF	0xXX	0x12	0x35

the user code word is set and the terminal returns the register address with the bit 7 for register access and the acknowledgement.

Byte0	Byte1	Byte2	Byte3
0x9F	0x00	0x00	0x00

Terminal-specific register description
Process variables

R0: Number of data bytes in the send FIFO
The number of data items in the send FIFO is in the low byte. The high byte is not used.

R1: Number of data bytes in the receive FIFO
The low byte contains the number of data in the receive FIFO. The high byte is not used.

R2 - R5: no function

R6: Diagnostic register
High byte: not used
Low byte: status of the receive channel (bits 0 – 7)

Bit No.		Meaning
Bit 0	1	The receive buffer has overflown and arriving data is lost.
Bit 1	1	Parity error has occurred.
Bit 2	1	Framing error has occurred.
Bit 3	1	Over run error has occurred.
Bit 4	1	Buffer is full
Bits 5 - 15	-	not used

Manufacturer parameters

R18: Buffer size
[0x0080]
Register R18 defines the number of data items in the receive FIFO as from which the BUF_F bit is set in the status byte.

Low byte: BUF_F is set in the status if this value is reached.
High byte: not used

User parameters

R32: Baud rate:
[0x0006]

Bit No.		Baud rate
Bit 2 Bit 1 Bit 0		
	0 1 1	1200 Baud
	1 0 0	2400 Baud
	1 0 1	4800 Baud
	1 1 0	9600 Baud [1 1 0]
	1 1 1	19200 Baud
Bits 3 - 15	-	not used

The baud rate can also be set in accordance with the following equation:

Baud rate = 4 MHz/(16*(HB+1))

At the same time, 0xFF must be written into the low byte and the high byte (HB) specifies the operator.

R33: Data frame
[0x0003]
The data frame is set in this register.

Bit No.		Meaning
Bit 2 Bit 1 Bit 0		
	0 0 1	7 data bits, even-parity
	0 1 0	7 data bits, odd-parity
	0 1 1	8 data bits, no parity [0 1 1]
	1 0 0	8 data bits, even-parity
	1 0 1	8 data bits, odd-parity
Bit 3	0/1	0: 1 stop bit [0] 1: 2 stop bits
Bits 4- 15	-	not used

R34: Feature register:
[0x0003]

Feature Bit No.		Mode description
Bit 0	0/1	0: half duplex: receipt of the data sent by the teminal is suppressed. 1: full duplex: data sent is monitored in RS485. [1]
Bit 1	0/1	0: standard output format 1: alternative output format [1]
Bit 2	1	The terminal copies the status byte into the shift register of the K bus one cycle later than the more significant data bytes, thus reducing the data transfer rate to the controller.[0]
Bit 3	1	The terminal supports the XON/XOFF protocol when sending data, i.e. the terminal sends the data transferred from the controller until it receives the XOFF (DC3==0x13) signal from the partner. Sending is then suppressed until the XON (DC1==0x11) signal is received. [0]
Bit 4	1	The terminal supports the XON/XOFF protocol when receiving data. The terminal sends the XOFF control character when the terminal's buffer contains 118 characters. XON is sent if XOFF has been sent beforehand and the buffer's contents have fallen below the buffer limit. of 18 bytes. [0]
Bit 5	0/1	0: The terminal is used in a bus structure in conformity with the RS485 standard. [0] 1: The terminal is used as a point-to-point connection (RS422), and so the controller no longer switches the data line to high impedence.
Bits6 - 15	-	not used

R35: Number of data bytes mapped in the bus coupler
[0x0003]
Low byte: number of data bytes in the bus coupler and transferred to the controller. Between 1 and 5 data bytes can be transferred. If more than 3 bytes of user data are to be transferred, the new number of bytes must be entered in this register.

High byte = not used

Data transfer

Control byte in process data transfer

The control byte is transferred from the terminal to the controller. It can be used in the register mode (REG = 1) or in the process data transfer (REG = 0) (see remark in the annex). The control and status byte in process data transfer is used to handle data transfer (handshake)

MSB

REG=0	OL2	OL1	OL0	0	IR	RA	TR

Status byte in process data mode

The status byte is transferred from the terminal to the controller. It contains the data needed for the handshake.

MSB

REG=0	IL2	IL1	IL0	BUF_F	IA	RR	TA

TR/TA:TRANSMIT-REQUEST/ TRANSMIT-ACCEPTED bits

The handshake for sending the data is realized by way of this bit. A change of state on the part of TR results in loading of the number of data items defined via OL0-OL2 (up to 5) into the send FIFO. The terminal signals execution of this command via TA.

Example

Output Control byte	Input status byte	Comment
00000000	0XXXX0X0	Start of data transfer
00100001 Data bytes: in D0 and D1	0XXXX0X0	Controller requests sending of 2-data from the terminal
....	....	
00100001 Data bytes: in D0 and D1,	0XXXX0X1	Terminal has loaded 2-data into the send FIFO and thecommand has been executed.
01010000 Data bytes in D0 to D4	0XXXX0X1 Data bytes:DC	Controller requests sending of 5-data (D0-D4) from the terminal
....	....	
01010000 Data bytes: in D0 und D1	0XXXX0X0	Terminal has loaded 5-data into the send FIFO and the command has been executed

RA/RR:REICEIVE-ACCEPTED/RECEIVE-REQUEST

By way of a status change of RR, the terminal informs the controller that the number of data items indicated in IL0-IL1 is located in D0-D4. Transfer of the data is acknowledged in the control byte with RA, and only then is new data transferred from the terminal to the controller.

Example

Output control byte	Input status byte	Comment
00000000	0XXXX00X	Start of data transfer
0XXX000X	0011X01X	Terminal requests acceptance of 3-data from D0-D2 by the controller.
....	....	
0XXX001X	0011X01X	Controller has accepted data
0XXX001X	0101X00X	Terminal requests acceptance of 5-data from D0-D4 by the controller
....	....	
0XXX001X	0101X00X	Controller has accepted data

IR/IA: INIT-REQUEST/INIT-ACCEPTED

The terminal performs initialization if IR is high. The send and receive functions are disabled, the FIFO flags are reset and the interface is initialized with the values of the responsible registers (R32-R35,R18). The terminal acknowledges execution of initialization with IA.

Example

Output control byte	Input status byte	Comment
0XXXXXXX	0XXXXXXX	Start of data transfer
00000100	0XXXXXXX	Initialization is requested by the controller.
....	....	
00000100	00000100	Terminal has completed initialization
00000000	00000100	Controller requests data exchange
....	....	
00000000	00000000	Terminal is ready

BUF_F: BUFFER-FULL_Flag

The receive FIFO is full. Data that is now received is lost.

Error handling

If a parity, framing or overrun error occurs, the data item concerned is lost, and it is not loaded into the terminal's receive FIFO.

Incoming data is ignored if the buffer is full.

The corresponding diagnostic bits are set in R6 if an error occurs.

Annex

As already described in the chapter on terminal configuration, each bus terminal is mapped in the bus coupler. In the standard case, this mapping is done with the default setting in the bus coupler / bus terminal. This default setting can be modified with the TRS KS2000 configuration software or using master configuration software (eg ComProfibus). The following tables provide information on how the KL6021, maps itself in the bus coupler depending on the set parameters.

Standard format

In the standard format, by default the KL6021 is mapped with 4 bytes (adjustable: 2 to 6 bytes via R35) of input and output data.
Remark: in the standard format, the CT/ST byte is used for register and process data communication.

		I/O Offset	High Byte	Low Byte
Complete evaluation	= X	3		
MOTOROLA format	= X	2	D4(opt.)	D3(opt.)
Word alignment	= X	1	D2(opt.)	D1(opt.)
		0	D0	CT/ST

Alternative format

In the alternative format, the KL6021 is mapped with 4/6 bytes of input data and 4/6 bytes of output data. When delivered the KL6021 is set to the alternative format.
Remark: in the alternative format, the CT/ST byte is used only for register communication and the serial CT/ST byte is used only for the data handshake.

		I/O Offset	High Byte	Low Byte
Complete evaluation	= 0	3		
MOTOROLA format	= 0	2		
Word alignment	= 0	1	D2	D1
		0	D0	Ser-CT/ST

		I/O Offset	High Byte	Low Byte
Complete evaluation	= 0	3		
MOTOROLA format	= 1	2		
Word alignment	= 0	1	D1	D2
		0	Ser-CT/ST	D0

		I/O Offset	High Byte	Low Byte
Complete evaluation	= 1	3		
MOTOROLA format	= 0	2	D2	D1
Word alignment	= 0	1	--	D0
		0	Ser-CT/ST	CT/ST

		I/O Offset	High Byte	Low Byte
Complete evaluation	= 1	3		
MOTOROLA format	= 1	2	D1	D2
Word alignment	= 0	1	--	Ser-CT/ST
		0	D0	CT/ST

		I/O Offset	High Byte	Low Byte
Complete evaluation	= 1	3	D2	D1
MOTOROLA format	= 0	2		--
Word alignment	= 1	1	D0	Ser-CT/ST
		0		CT/ST

		I/O Offset	High Byte	Low Byte
Complete evaluation	= 1	3	D1	D2
MOTOROLA format	= 1	2		--
Word alignment	= 1	1	Ser-CT/ST	D0
		0		CT/ST

Legend

Complete evaluation: the terminal is mapped with control / status byte.
Motorola format: the Motorola or Intel format can be set.
Word alignment: the terminal is at a word limit in the bus coupler.
CT: Control byte (appears in the PI of the outputs).
ST: Status byte (appears in the PI of the inputs).
Ser.-CT: control byte for the handshake (appears in the PI of the outputs)
Ser.-ST: status byte for the handshake (appears in the PI of the inputs)
D0 – D4: data bytes 0 – 4

Table of the register set of the KL6021

Address	Description	Default value	R/W	Storage medium
R0	Number of data bytes in the send buffer	variable	R	RAM
R1	Number of data bytes in the receive buffer	variable	R	RAM
R2	not used	0x0000	R	
R3	not used	0x0000	R	
R4	not used	0x0000	R	
R5	not used	0x0000	R	
R6	Diagnostic register	variable	R	RAM
R7	Command register - not used	0x0000	R	
R8	Terminal type	6021	R	ROM
R9	Software version number	0x????	R	ROM
R10	Multiplex shift register	0x0218	R	ROM
R11	Signal channels	0x0130	R	ROM
R12	Minimum data length	0x3030	R	ROM
R13	Data structure	0x0000	R	ROM
R14	not used	0x0000	R	
R15	Alignment register	variable	R/W	RAM
R16	Hardware version number	specific	R/W	SEEROM
R17	not used	0x0000	R/W	SEEROM
R18	Buffer full indication	0x0080	R/W	SEEROM
R19	not used	0x0000	R/W	SEEROM
R20	not used	0x0000	R/W	SEEROM
R21	not used	0x0000	R/W	SEEROM
R22	not used	0x0000	R/W	SEEROM
R23	not used	0x0000	R/W	SEEROM
R24	not used	0x0000	R/W	SEEROM
R25	not used	0x0000	R/W	SEEROM
R26	not used	0x0000	R/W	SEEROM
R27	not used	0x0000	R/W	SEEROM
R28	not used	0x0000	R/W	SEEROM
R29	not used	0x0000	R/W	SEEROM
R30	not used	0x0000	R/W	SEEROM
R31	Code word register	variable	R/W	RAM
R32	Baud rate	0x0006	R/W	SEEROM
R33	Data frame	0x0003	R/W	SEEROM
R34	Feature register	0x0002	R/W	SEEROM
R35	Number of data bytes to the bus coupler	0x0003	R/W	SEEROM
R36	not used	0x0000	R/W	SEEROM
R37	not used	0x0000	R/W	SEEROM
R38	not used	0x0000	R/W	SEEROM
R39	not used	0x0000	R/W	SEEROM
R40	not used	0x0000	R/W	SEEROM
R41	not used	0x0000	R/W	SEEROM
R42	not used	0x0000	R/W	SEEROM
R43	not used	0x0000	R/W	SEEROM
R44	not used	0x0000	R/W	SEEROM
R45	not used	0x0000	R/W	SEEROM
R46	not used	0x0000	R/W	SEEROM
R47	not used	0x0000	R/W	SEEROM