

KL-6051

Technical Documentation Data transfer terminal

Please keep for further use !

Edition date/Rev. date: 28.07.1998
Document no./Rev. no.: TRS - V - BA - GB - 0118 - 00
Software version: 1.0
File name: TRS-V-BA-GB-0118.DOC
Author: KOH

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

Imprint

TRSystemtechnik GmbH

D-78647 Trossingen
Eglisshalde 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

i

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:

28.07.1998

Revision	Date

Table of contents

1 **5**

Data transfer terminal with serial RS422 terminal KL-6051 **5**

 Technical data 5

 Description of functions 6

 Terminal configuration 6

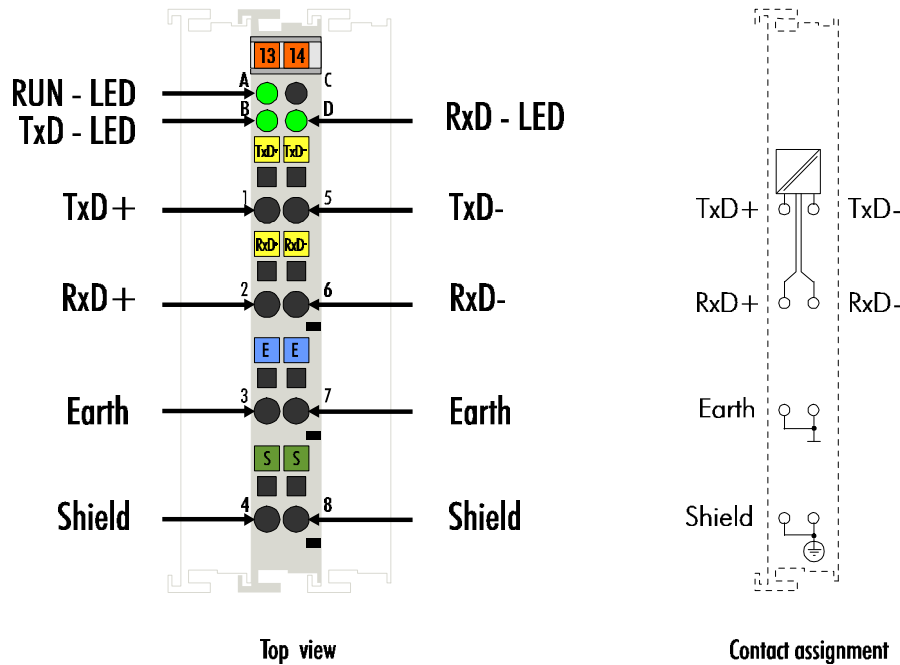
 Register communication KL-6051 8

 Data transfer 13

Annex **14**

1

Data transfer terminal with serial RS422 terminal KL-6051



Technical data	KL-6051
Data transfer channels	TxD and RxD, full duplex
Data transfer rate	62500 Baud (8N1)
Bit transfer	via 2 twisted wire pairs with difference signals
Line impedance	120 Ω
Data transfer link	app. 1000 m twisted pair
Power supply	via the K-bus
Current consumpt. of K-bus	65 mA typ.
Electrical isolation	500 Vrms (K-bus/data line)
Data buffer	128 bytes receive buffer, 16 bytes send buffer
Bit width in the process image	I/O: 4 x 8 bits user information data (up to 5 x 8 bits possible), (1 x 8 bits control/status can be optionally inserted)
Configuration	no address or configuration setting via the bus coupler
Function check.	data exchange is monitored by watchdog
Data exchange time	< 5 ms
Weight app.	60 g
Operating temperature	0°C ... +55°C
Storage temperature	-25°C ... +85°C
Relative humidity	95%, no condensation
Vibration/shock resistance	in accordance with IEC 68-2-6 / IEC 68-2-27
EMC resistance, burst/ESD	in accordance with EN 61000-4-4 / EN 61000-4-, limits in accordance with EN 50082-2-4
Installation position	any
Type of protection	IP20

Description of functions

The KL-6051 serial interface terminal enables an exchange of data between different field bus systems. Regardless of the higher-level field bus system, data can be exchanged in full duplex mode. Up to 40 inputs and 40 outputs are transmitted between the field bus systems and, in addition, the status byte contains information about the quality and the status of data transfer. The terminal is supplied with a 200 ms RCV timeout, i.e. the inputs of the higher-level controller are set to zero if the terminal does not receive any valid data via the serial interface from the second station within 200 ms. In the default setting, 32 bits are bidirectionally available for data exchange.

Features

- Coupling of two field bus systems
- Exchange of up to 40 bits bidirectionally
- Status byte for data channel status message
- Safeguarded data transfer by longitudinal parity, vertical parity, log
- Transmission medium: RS422 full duplex
- Maximum transmission distance: 1000 m
- Simple software interface for control by the emulation of up to 40 bits
- Parallel I/O
- Data transfer time exchange time < 5 ms

Terminal configuration

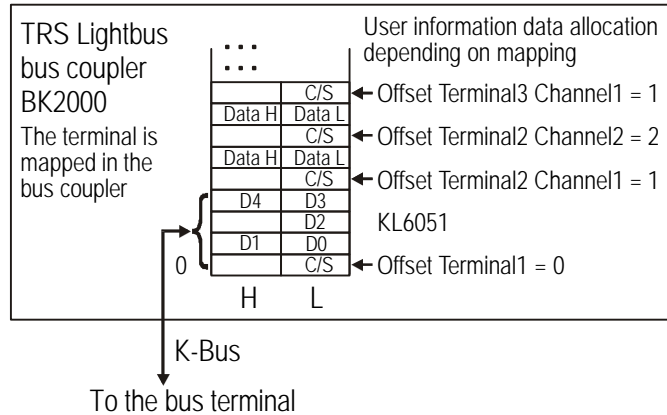
The terminal can be configured and parametrized via the internal register structure.

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.

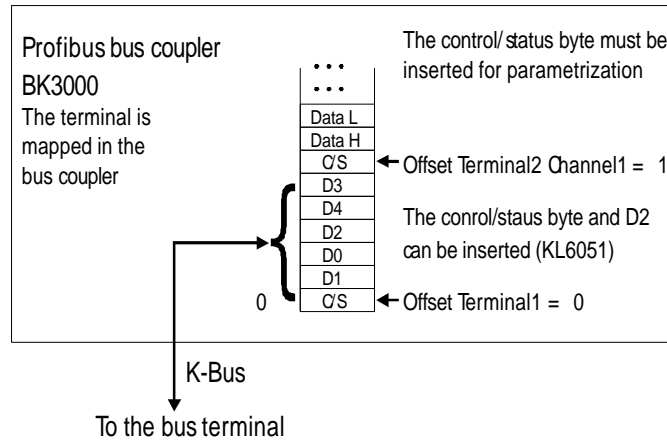
TRS Lightbus coupler BK2000

When using the TRS Lightbus coupler BK2000, the control / status byte is always mapped in addition to the data bytes. It is always in the low byte at the offset address of the terminal channel. In the case of the KL-6051 there will be 6 bytes data (5 bytes user data and 1 byte control/status) exchanged with the control system.



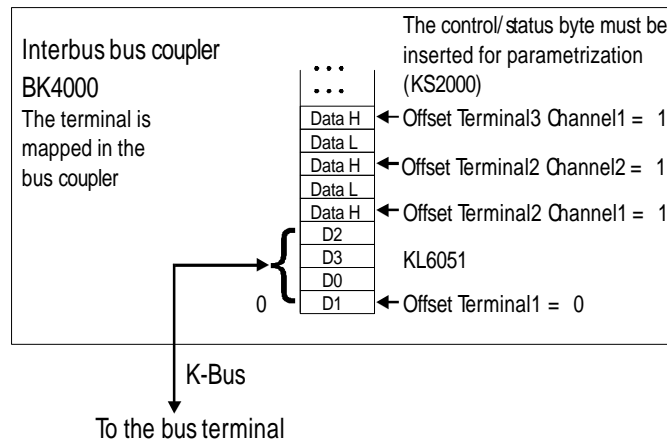
Profibus coupler BK3000

When using the Profibus coupler BK3000, how the KL-6051 is to map itself in the bus coupler is set in the master configuration software. The figure shows the mapping for 6 bytes of input data and 6 bytes of output data.



Interbus coupler BK4000

By default, the Interbus coupler BK4000 maps the KL-6051 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 KL-6051

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

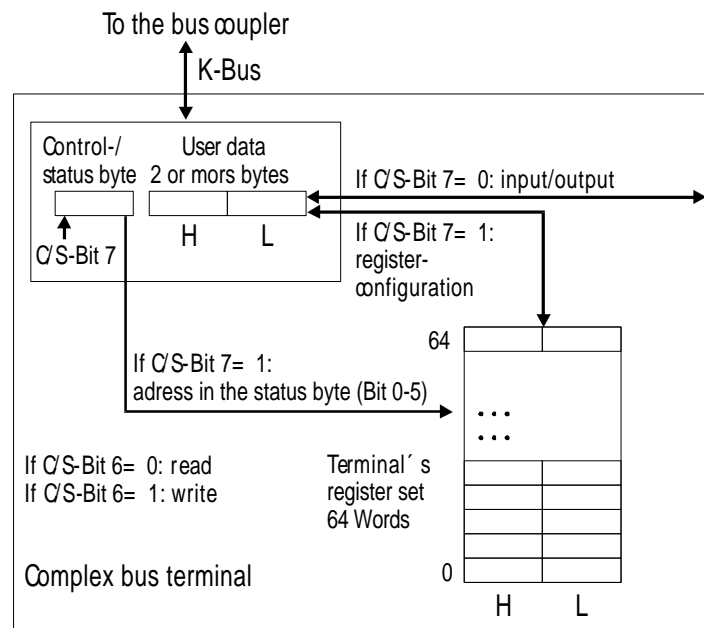
Bits 0 to 5: address

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 KI3022 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

R1-R5: no function

R6: diagnostic register [R/W]

HB HB Status of station 2 (partner terminal)
LB LB Status of station 1

R7: no function

User parameters

R32 Feature register
[0x0007]

Feature bit no.		
bit 0	1	RCV-timeout enable (R334) [1]
bit 1	1	TRS-timeout enable (R35) [1]
bit 2	0/1	0: terminal bus communication via interrupt 1: terminal bus: poll state [1]
bit 3	0/1	0: data is transferred word-consistently 1: data is transferred completely consistently
bit 15-4	-	not used

R33: Baud rate

[0x0003] (62.5kHz)

HB = DC

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

*R34: RCV timeout
[0x00014] (200ms)*

HB, LB = unsigned integer, 1 digit corresponds to 10ms

When the RCV timeout enable bit is set in R32, this value is valid. If the terminal does not receive any valid data via the serial interface for X-ms, the controller's inputs are set to the value ZERO.

*R35: TRS timeout
[0x0014] (200ms)*

HB, LB = unsigned integer, 1 digit corresponds to 10ms

When the TRS timeout enable bit is set in R32, this value is valid. If the terminal does not receive any valid data via the serial interface for X-ms, no data is sent via the serial interface. Accordingly, the RCV timeout of the 2nd terminal would take effect.

Data transfer

8 data bits, 1 start bit, 1 stop bit, even parity

Data containing errors is not transferred to the controller. The processed data is still valid when the CHK, OVR or PAR date is set. These bits merely represent the quality of data transfer.

*Status byte in
the process data mode*

MSB

REG=0			RCVT1	RCVT2	CHK	OVR	PAR
-------	--	--	-------	-------	-----	-----	-----

PAR: parity error or invalid data frame

OVR: buffer overflow

CHK: invalid checksum

RCVT2: the partner terminal has an RCD timeout

RCVT1: the terminal is not receiving any data from the partner. The terminal has set the controller inputs to zero.

For use without status information, the data transfer link can be monitored from the other respective side of the link by means of a bit set by the controller. Data transfer of the KL-6051 is checked by a watchdog. Thus, failures of the field buses or of data transfer between the KLK6051 units are easily recognizable.

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 (e.g. ComProfibus). The following tables provide information on how the KL-6051 maps itself in the bus coupler depending on the set parameters.

Mapping in the bus coupler The KL is mapped in the bus coupler depending on the set parameters. The terminal occupies memory space in the process image of the inputs and outputs.

Default: CAN CAL, CANopen,

DeviceNet

	I/O Offset	High Byte	Low Byte
Complete evaluation = 0	3		
MOTOROLA format = 0	2		
Word alignment = X	1	D3	D2
	0	D1	D0

Default: Profibus, Interbus

	I/O Offset	High Byte	Low Byte
Complete evaluation = 0	3		
MOTOROLA format = 1	2		
Word alignment = X	1	D2	D3
	0	D0	D1

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

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

Default: Lightbus

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

	I/O Offset	High Byte	Low Byte
Complete evaluation = 1	3	D3	D4
MOTOROLA format = 1	2		D2
Word alignment = 1	1	D0	D1
	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).
D0 – D4: Data bytes 0 - 4

Table of the register set
of the KL-6051

Address	Designation	Default	R/W	Storage Medium
R0	Not used	0x0000	R	
R1	Not used	0x0000	R	
R2	Not used	0x0000	R	
R3	Not used	0x0000	R	
R4	Not used	0x0000	R	
R5	Not used	0x0000	R	
R6	Diagnostic register	0x0000	R	RAM
R7	Command register not used	0x0000	R	
R8	Terminal type	6051	R	ROM
R9	Software version number	0x????	R	ROM
R10	Multiplexed 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	Not used	0x0000	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	Feature register	0x0007	R/W	SEEROM
R33	Baud rate	0x0003	R/W	SEEROM
R34	RCV timeout	0x0014	R/W	SEEROM
R35	TRS timeout	0x0014	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