KL-5001

Technical Documentation
SSI Sensor Interface

Please keep for further use !

Edition date/Rev. date:	03.02.1998
Document no./Rev. no.:	TRS - V - BA - GB – 0097 - 00
Software version:	1.0
File name:	TRS-V-BA-GB-0097.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

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

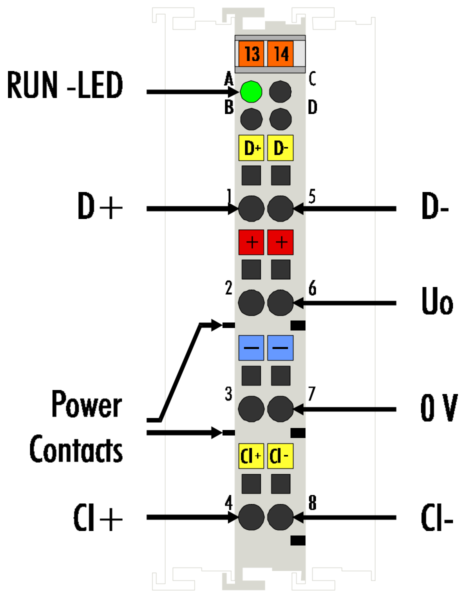
Note:
The cover of this document shows the current revision status and the corresponding date. Since each individual page has its own revision status and date in the footer, there may be different revision statuses within the document.

Document created: 03.02.1998

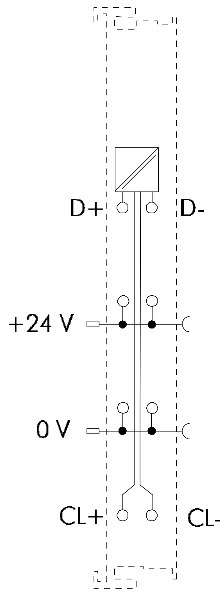
Revision	Date

Table of contents

SSI Sensor Interface
KL5001



Top View



KL 5001

Technical data	KL5001
Sensor connection	binary input: D+, D-; binary output: Cl+, Cl-
Power supply	24 V DC via power contacts
Current consumption	typically 85 mA without sensor
Sensor supply	24 V DC -15% / +20%
Data transfer rate	variable up to 1 MHz, 625 kHz default
Serial input	24 bit width (variable)
Data direction	read
Signal output	difference signal (RS422)
Signal input	difference signal (RS422)
Electrical isolation	Signal input via optocoupler, 500 V (T-Bus / field voltage)
Bit width in the process image	32 bits: 24 bits data, 8 bits control/status
Weight approx..	80 g
Operating temperature	0°C ... +55°C
Storage temperature	-25°C ... +85°C
Relative humidity	95%, no condensation
Vibration/shock resistance	conforms to IEC 68-2-6 / IEC 68-2-27
EMC resistance Burst / ESD	conforms to EN 61000-4-4 / EN 61000-4-2, limit EN 50082-2
Installation position	any
Type of protection	IP20

Description of functions

The terminal KL5001 is an SSI interface for the direct connection of an SSI sensor. The sensor is powered via the SSI interface. To read out the sensor, the terminal outputs a clock burst and provides the incoming data stream to the controller in the process image. Different operating modes, transmission frequencies, bit widths and code conversions can be set. The individual configuration is stored permanently in a register set.

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 transmitted by the bus coupler for 100 ms.

Process data
Alternative output format

The SSI interface is supplied with a data width of 24 bits and Gray binary number conversion activated in the alternative output format. The baud rate to the SSI senosr is set to 250 kHz. The process data is output in the input data bytes D0 - D3. Mapping of the terminal in the alternatie format is described in further detail in the chapter on terminal cofniguration.

Standard output format

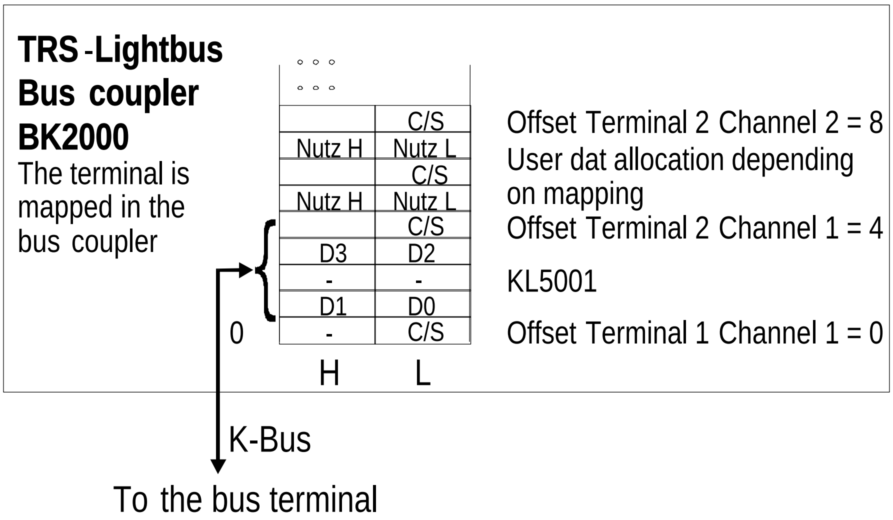
In the standard output format, 4 bytes of input data are mapped in the bus coupler by default.

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 als be mapped.

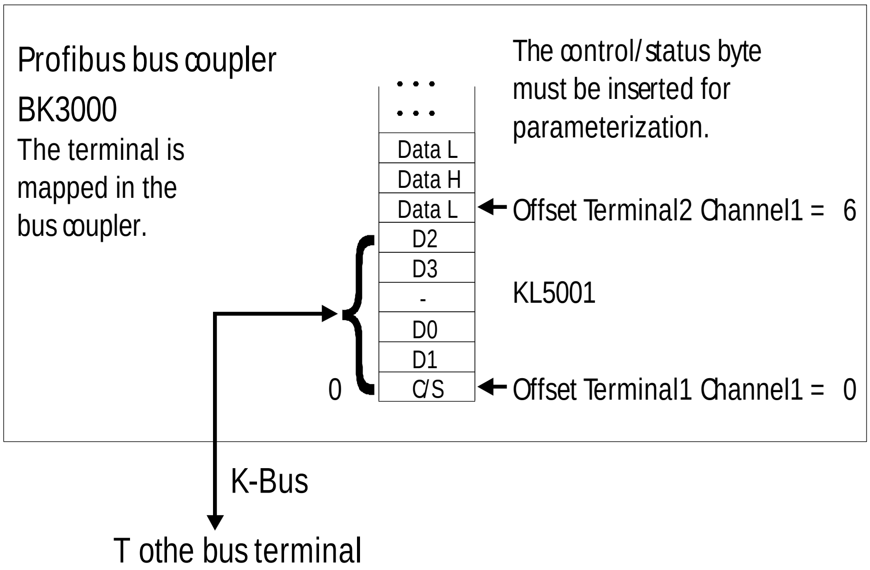
TRS Lightbus coupler BK2000

In the case of the TRS Lightbus coupler BK2000, the control /status byte is also always (ie in the case of all analog terminals) mapped in addition to the data bytes.It is always in the low byte at the offset address of the terminal channel.



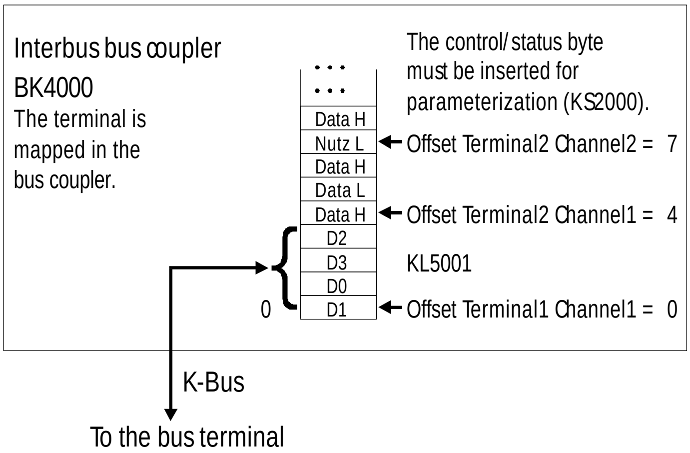
Profibus coupler BK3000

In the case of the Profibus coupler BK3000, for which terminal channels the control/status byte is also to be inserted must be defined in the master configuration. If the control/status byte is not evaluated, the KL5001 occupies 4 bytes of input data. The figure shows the mapping with control/status byte.



Interbus coupler BK4000

By default, the Interbus coupler BK4000 maps the Kl5001 with 4 bytes of input data. Parametrization via the field bus is not possible. The KS2000 software is required for configuration if use is to be made of the control/status byte.



Other bus couplers and further information

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

Reference

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

Parametrization with the KS2000 software

Independently of the field bus system, parameters can be set via the serial configuration interface in the bus coupler using the TRS KS2000 configuration software.

Register Communication KL5001

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 this 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-register	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

Register type

R8-R15 Registers in the terminal`s internal ROM

From the producer the type- and system parameters are programmed fixed. That means that this parameters are not changeable. The user can only read the parameter.

R8: Terminal type:

The terminal type in register R8 is used for the identification.

R9: Software version X.y

The software version can be read as an ASCII string.

R10: Data length

R10 contains the number of the multiplexed shifting registers and the length in bit.
The bus coupler recognized this structure.

R11: Signal channels

In comparison to R10 register R11 contains the number of the logical available channels. So for example it is possible that a physical available shifting register can be consisted of sevaral signal channels.

R12: Minimum data length

The respective byte contains the minimum transmitting data length of a channel. Is the MSB is set the status byte is cancelled.

R13: Data type registers

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 word 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

Data type register	
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 in the terminal 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.

R32: Feature register

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

R33 - R47

Registers that depend on the terminal type.

Extended application area

R47-R63

These registers have not yet been implemented.

Register access via proces 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 set.

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.

Bits 0 to 5: address

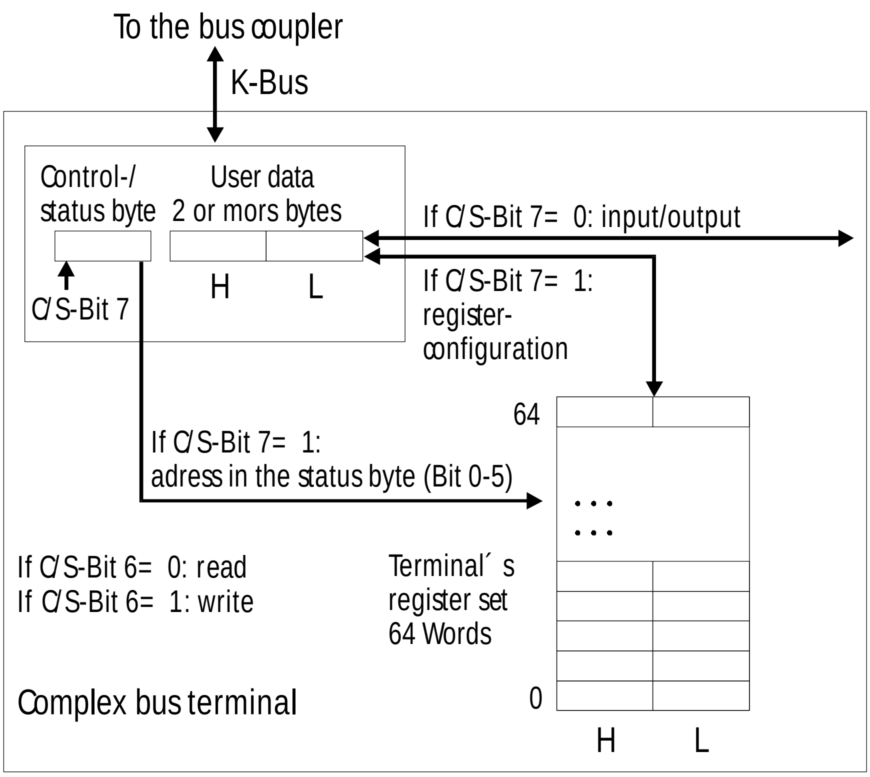
The address of the register to be addressed is entered in bits 0 to 5 of control byte.

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 this 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 interger 3022).

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

A further example

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

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

Terminal-specific register description
Process variables
Application parameters

R32: feature register: [0x0007]

Feature bit No.		Description of the operating mode
Bit 0	0/1	0: binary output 1: Gray binary number conversion [1] The numbers are output in Gray code
Bit 1	0/1	0: standard output format 1: alternative output format [1]
Bit 2	0/1	0: freewheeling 1: synchronous mode [1] The data is loaded in synchronism with the read cycle of the terminal bus.
Bit 3	0/1	0: multiturn evaluation of the sensor [0] 1: single turn evaluation of the sensor
Bit 4	1	Disable Frame Error [0] After the last valid bit, no check is made as to whether the data line is supplying a zero signal.
Bit 5 - 15	-	Not used

R33: baud rate
[0x0002]
The baud rate for reading the SSI sensor is selected via this register.

High byte = not used

Low byte	Baud rate
1	1 MHz
2	250 kHz [2]
3	125 kHz
4	100 kHz
5	83 kHz
6	71 kHz
7	62.5 kHz

R34: data length
[0x18]
The data length that appears in the process image can be set by this register.
The permissible value range is: 0-32 bits

HB = not used
Low byte = 0...32 bits in hexadecimal notation

STATUS byte during process data exchange

The status byte is transferred from the terminal to the controller. The status byte contains various status bits of the SSI sensor interface terminal KL5001.

MSB

REG=0	ERROR	0	0	0	0	FRAME_E	SSI_IN_E

Bit	Function
ERROR	A general error has occurred. This bit is set if a FRAME or SSI_IN error has occurred.
FRAME_E	An invalid data frame has occurred, i.e. the data frame is not terminated with zero (possibly a wire breakage on clock lines).
SSI_IN_E	The SSI input of the terminal has low level when no data transfer is taking place. (SSI has no power supply or wire breakage at the SSI data inputs D+ or D- or data lines swapped.)

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 configuration software KS2000 or using master configuration software (eg ComProfibus). The following tables provide information on how KL5001,maps itself in the bus coupler depending on the set parameters.

Standard Format

The KL5001 is mapped in the bus coupler depending on the set parameters. If the terminal is evaluated completely, the terminal occupies memory space in the process image of the input and outputs.

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

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

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

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

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

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

Alternatives Format

In the alternative format the KL5001 is mapped with 4 or 6 data bytes. If the terminal is evaluated completely, the terminal occupies memory space in the process image of the input and outputs.

Default: CAN CAL, CANopen

DevicNet

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

Default: Interbus, Profibus

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

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

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

Default: Lightbus

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

Setting		I/O Offset	High Byte	Low Byte
Complete evaluation	= 1	3	D2	D3
MOTOROLA format	= 1	2	-	-
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 formal 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 – D3: Data bytes

Table of the register set
of the KL5001

Address	Description	Default value	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	variable	R	RAM
R7	Command register - not used	0x0000	R	
R8	Terminal type	5001	R	ROM
R9	Software version number	0x????	R	ROM
R10	Multiplex shift register	0x0218	R	ROM
R11	Signal channels	0x0128	R	ROM
R12	Minimum data length	0x00A8	R	ROM
R13	Data structure	0x0000	R	ROM
R14	not used	0x0000	R	
R15	Alignment register	variable	R/W	RAM
R16	Hardware version number	0x????	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	0x0002	R/W	SEEROM
R34	Data length	0x0018	R/W	SEEROM
R35	not used	0x0000	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