KL4132

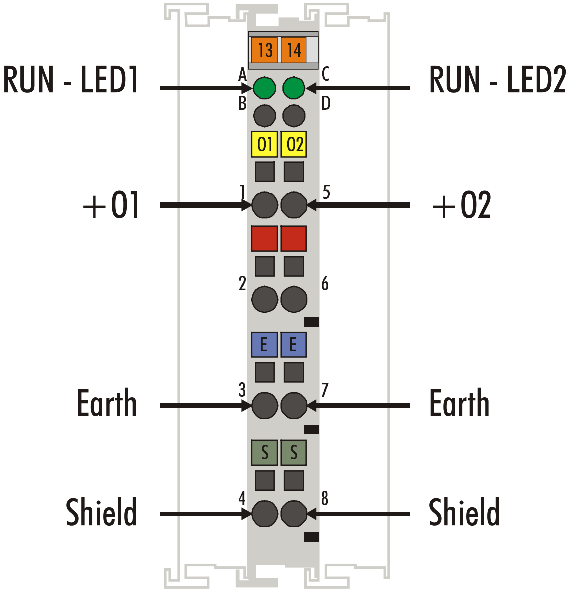
2-Channel Analog Output Terminal -10...10 V
Configuration Instructions

Version 2.0

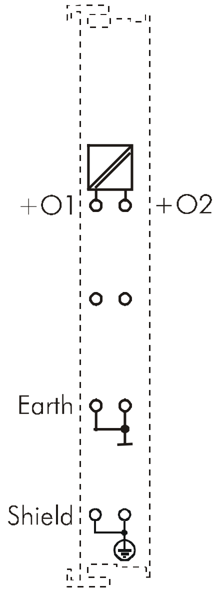
Eiserstraße 5 / D-33415 Verl / Phone 05246/963-0 / Fax 05246/963-149

Contents

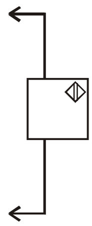
Technical data



Top View



Contact Assembly



Connection

Technical data	KL4132
Number of outputs	2
Power supply	via the K-Bus
Signal voltage	-10 V ... +10 V
Load	> 5 kΩ
Accuracy	< ± 0.1% (of the upper range value)
Resolution	16 bits
Conversion time	app. 1.5 ms
Electrical isolation	500 Vrms (K-Bus / signal voltage)
Current consumption from K-Bus	75 mA
Bit width in the process image	O: 2 x 16 bits data (2 x 8 bits control/status)
Configuration	no address or configuration setting
Weight approx..	85 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 50082 (ESD, Burst) / EN 50081
Installation position	any
Type of protection	IP20

Description of functions

The analog output terminal KL4132 generates output signals within the range from –10 V up to +10 V. The output voltage is output by the terminal with a 16-bit resolution. The output voltage is electrically isolated from the terminal bus level.

Input format of process data

In the default setting, process data is entered in twos complement (integer -1 corresponds to 0xFFFF).

Process data item (Hex/decimal)	Output value
0x8001 (-32767)	-10 V
0xC001 (-16383)	-5 V
0x0000 (0)	0 V
0x3FFF (16383)	5 V
0x7FFF (32767)	10 V

LED Display

The two RUN LEDs indicate the operating state of the affiliated terminal channel.

RUN LED:
On – normal operation
Off – watchdog-timer overflow has occurred. The green LEDs go off if no process data is transferred by the bus coupler for 100 ms. The output outputs a voltage that is configurable by the user (see feature register).

Process data

The process data arriving from the bus coupler is output to the process as follows.

X = Process data of the PLC
B_a,A_a = Manufacturer scaling (R17,R28)
B_h,A_h = Manufacturer scaling (R19,R20)
B_w,A_w = User scaling (R33,R34)
Y_dac = Output value to the DA converter

Neither user nor manufacturer scaling active:
Y_0 = B_a + A_a*X (1.0)
Y_dac = X

Manufacturer scaling active:
Y_1 = B_h + A_h * Y_0 (1.1)
Y_dac=Y_1

User scaling active:
Y_2 = B_w + A_w * Y_0 (1.2)
Y_dac=Y_2

Manufacturer and user scaling active:
Y_1 = B_h + A_h * Y_0 (1.3)
Y_dac = B_w + A_w * Y_1 (1.4)

The straight-line equations are activated via R32.

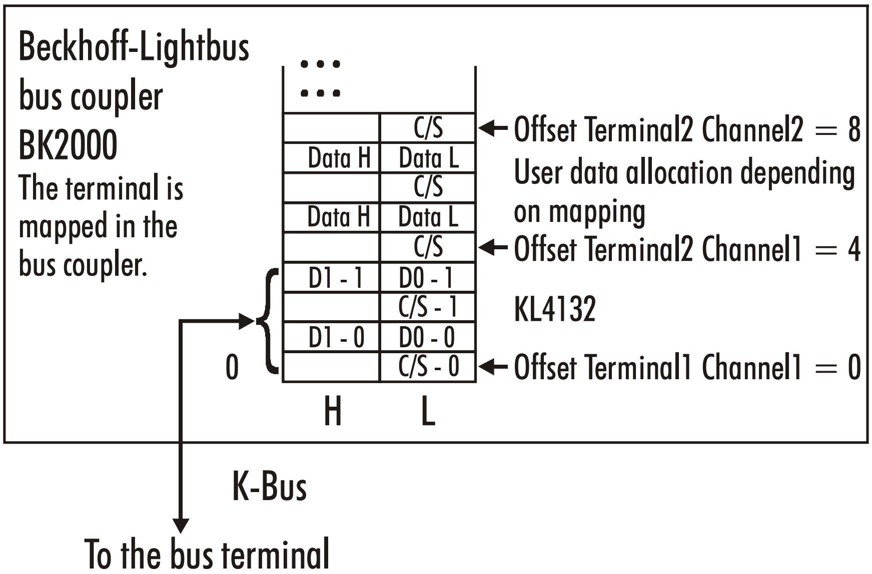
Terminal configuration

The terminal can be configured and parametrized by way of the internal register structure.

Each terminal channel is mapped in the bus coupler. The terminal's data is mapped differently in the bus coupler's memory 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.

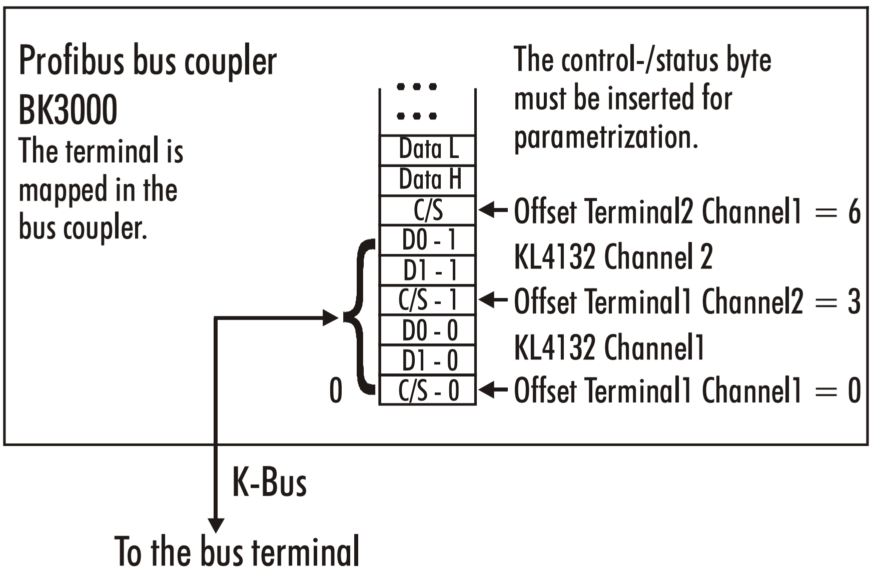
Beckhoff Lightbus Coupler BK2000

In the case of the Beckhoff-Lightbus coupler BK2000, the control /status byte is always mapped besides the data bytes. It is always in the low byte at the offset address of the terminal channel.



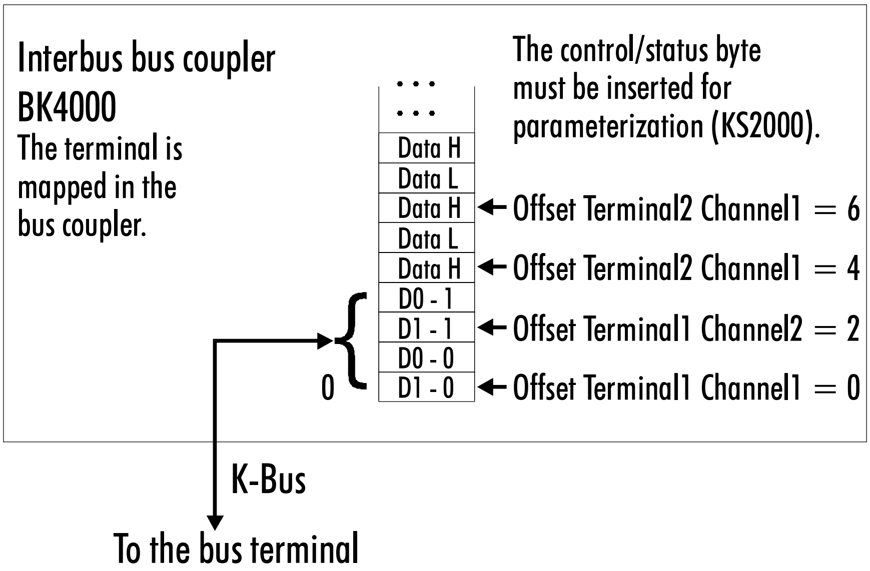
Profibus Coupler BK3000

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 KL4132 occupies 4 bytes of output data (2 bytes of user data per channel).



Interbus Coupler BK4000

By default, the Interbus coupler BK4000 maps the KL4132 with 4 bytes of output data (2 bytes of user data per channel). Parametrization via the field bus is not possible.The KS2000 software is needed for configuration if it is intended to use 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 under the heading of "Configuration of Masters".

Note

The annex contains an overview of the possible mapping configurations depending on the adjustable parameters.

Parametrization with the KS2000 software

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

Register description

The complex terminals can be adjusted to different operating modes or functionalities. The " general description of register " describes the contents of the registers, which are identical for all complex terminals.

The terminal-specific registers are explained in the section following to it.

The access to the internal registers of the terminal is described in the section " register communication ".

General register description

Complex terminals that possess a processor are capable of bidirectionally ex-changing 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, encoder interfaces, SSI interfaces, PWM terminals and all other parametrizable terminals.

Internally, all intelligent terminals possess a data structure that is identical in terms of it's 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	16-30
User parameters	31-47
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: Software 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. If the MSB is set, then the control/status byte is not necessarily needed for the function of the terminal and, with appropriate configuration of the coupler, is not transferred to the control system.

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. The user area is write protected over a Codeword.

Note

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/O's.

R33 - R47
Registers that depend on the terminal type

Extended application area

R47 - R63
These registers have not yet been implemented.

Terminal-specific register description

Process variables

R0 - R4: no function

R5: Raw DAC value Y_dac
The 16-bit value that is transferred to the DAC is referred to as the raw DAC value. This is calculated from the process data via manufacturer and user scaling.

R6 - R7: no function

Manufacturer parameters

R17: Offset – Hardware
Hardware offset adjustment of the terminal is realized via this register.

R18: Gain-Hardware *2^-16, 0xFFFF corresponds 1
Hardware gain adjustment of the terminal is realized via this register.

R19: Manufacturer offset B_h
This register contains the offset of the manufacturer's straight-line equation (1.1). The straight-line equation is activated via R32.

R20: Manufacturer scaling A_h
16 bit unsigned integer * 2^-8 [0x0100]
This register contains the scaling factor of the manufacturer's straight-line equation (1.1). The straight-line equation is activated via R32.
A 1 corresponds to the register value 0x0100

Application parameters

R32: Feature Register:
[0x0006]
The feature register determines the operating modes of the terminal.

Feature Bit No.		Mode description
Bit 0	1	User scaling (1.2) active [0]
Bit 1	1	Manufacturer scaling (1.1) active [1]
Bit 2	0	Watchdog timer active [0] The watchdog timer is on by default. In the event of a watchdog overflow, either the manufacturer or the user activation value is applied to output of the terminal.
Bit 3 - 7	-	not used, don't change
Bit 8	0/1	0: manufacturer activation value [0] 1: user activation value
Bits 9 - 15	-	not used, don't change

R33: User offset B_w
16 bit signed Integer [0x0000]
This register contains the offset of the user straight-line equation (4.1.). Thestraight-line equation is activated via R32.

R34: User scaling A_w
16 bit signed Integer * 2^-8 [0x0100]
This register contains the scaling factor of the user straight-line equation (4.1) The straight-line equation is activated via R32.

R35: User activation value
16 bit signed Integer [0x0000]
If the user activation value in R32 is activated, this value is applied to the output of the terminal after a system reset or a watchdog timer overlow (terminal has not received any process date for 100ms).

Eiserstraße 5 / D-33415 Verl / Phone 05246/963-0 / Fax 05246/963-149

Control/Status byte

CONTROL byte
in process data transfer
Gain and offset adjustment

The control byte is transferred from the controller to the terminal. It can be used in the register mode (REG = 1) or in process data transfer (REG = 0). The gain and offset of the terminal can be adjusted with the control byte (process data transfer).The code word must be entered in R31 to enable adjustment of the terminal. The terminal's gain and offset can then be adjusted.

The parameters are not permanently stored until the code word is reset!

Control byte:

Bit7 = 0

Bit6 = 1 Terminal adjustment function is active
Bit4 = 1 gain adjustment
Bit2 = 0 slow clock = 1000ms
1 fast clock = 50ms
Bit1 = 1 up
Bit0 = 1 down
Bit3 = 1 offset adjustment
Bit2 = 0 slow clock = 1000ms
1 fast clock = 50ms
Bit1 = 1 up
Bit0 = 1 down

STATUS byte
in process data transfer

The status byte is transferred from the terminal to the controller. In the case of the KL4132, the status byte has no function in process data transfer.

Register communication KL4132

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 supplied an acknowledgement (see examples).

Bits 0 to 5: address

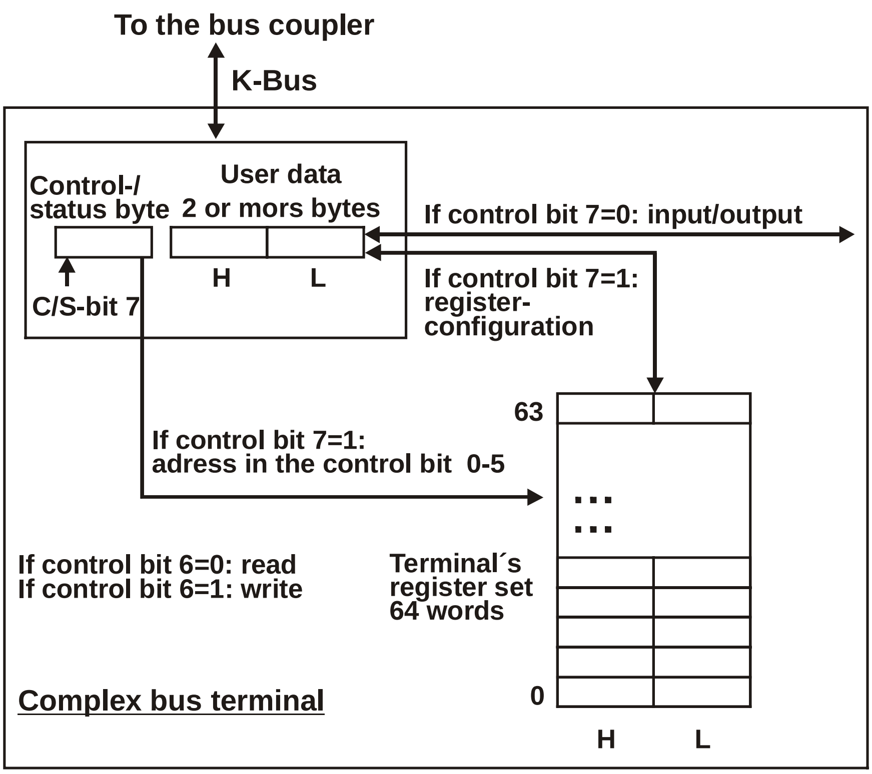
The address of the register to be addressed is entered in bits 0 to 5 of the control byte.

Control byte in the register mode

MSB

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

REG = 0 : Process data transfer
REG = 1 : Access to register structure
W/R = 0 : Read register
W/R = 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 Control	Byte1 Not used	Byte2 Data OUT, high byte	Byte3 Data OUT, low byte
0x88	0xXX	0xXX	0xXX

the terminal returns the following type designation (0x0BCE corresponds to the unsigned integer 3022).

Byte0 Status	Byte1 Not used	Byte2 Data IN, high byte	Byte3 Data IN, low byte
0x88	0x00	0x0B	0xCE

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 Control	Byte1 Not used	Byte2 Data OUT, high byte	Byte3 Data OUT, low byte
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 Status	Byte1 Not used	Byte2 Data IN, high byte	Byte3 Data IN, low byte
0x9F	0x00	0x00	0x00

Annex

As already described in the chapter 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 Beckhoff KS2000 configuration software or using master configuration software (e.g. ComProfibus or TwinCAT System Manager). The following tables provide information on how the KL4132 maps itself in the bus coupler depending on the set parameters.

Mapping in the bus coupler

Mapping in the bus coupler

The KL4132 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 inputs and outputs.

Default: CANCAL, CANopen, RS232, RS485, ControlNet, DeviceNet

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

Default: Interbus, Profibus

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

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

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

Default: Lightbus, Bus Terminal Controller (BCxxxx)

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

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

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 - 0 : D0 = Data-Low-Byte, 0 = channel 1
D1 – 1 : D1 = Data-High-Byte, 1 = channel 2

Table of the register

Register set

Address	Description	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	Raw DAC value	variable	R	RAM
R6	Diagnostic register	0x0000	R	
R7	Command register	0x0000	R	
R8	Terminal type	4132	R	ROM
R9	Software version number	0x????	R	ROM
R10	Multiplex shift register	0x0218	R	ROM
R11	Signal channels	0x0218	R	ROM
R12	Minimum data length	0x9800	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	Hardware offset adjustment	specific	R/W	SEEROM
R18	Hardware-gain adjustment	specific	R/W	SEEROM
R19	Manufacturer scaling: offset	0x0000	R/W	SEEROM
R20	Manufacturer scaling: gain	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	0x0000	R/W	SEEROM
R33	User offset	0x0000	R/W	SEEROM
R34	User gain	0x0100	R/W	SEEROM
R35	User activation value	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