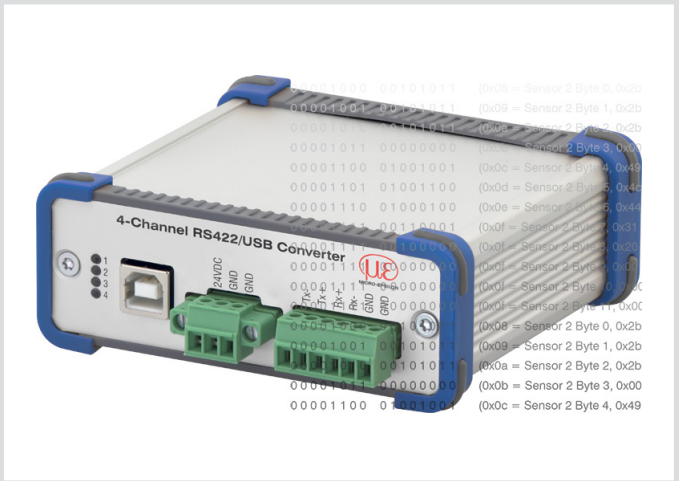
MICRO-EPSILON



Operating Instructions

IF2004/USB

4-Channel RS422/USB Converter
Data Format, Register Description

The following sensors/systems can be connected to the 4-channel RS422/USB converter:

- ILD 1302/1402/1420/1700/2200/2300 series sensors
- optoCONTROL ODC 2500/2520/2600 series sensors
- confocalDT IFD 2451/2471 series systems
- colorCONTROL ACS7000 series systems

MICRO-EPSILON
MESSTECHNIK
GmbH & Co. KG
Königbacher Strasse 15

94496 Ortenburg/Germany

Phone +49 (0) 8542/168-0
Fax +49 (0) 8542/168-90
email info@micro-epsilon.de
www.micro-epsilon.de

Contents

1. Symbols Used

The following symbols are used in these instructions.

i	Indicates a tip for users.
`Measurement`	Indicates hardware or a software button/menu.

2. Data Format

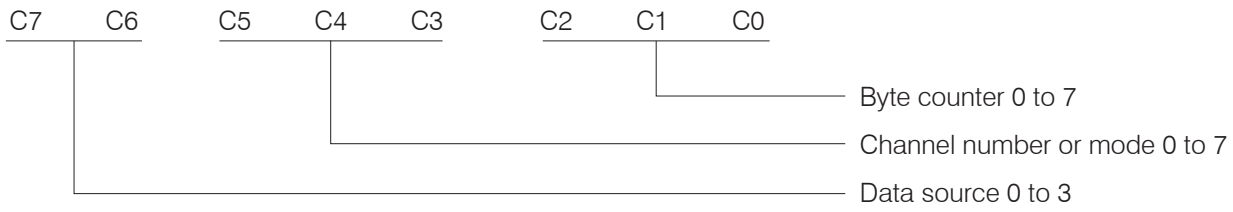
2.1 General Structure

	Tuple 1								Tuple 0							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	C7	C6	C5	C4	C3	C2	C1	C0	X7	X6	X5	X4	X3	X2	X1	X0
	Code bits								Data, address or mask bits							

Fig. 1 Data format

Tuple 0 Bits 0 to 7 contain the data, address or mask
Tuple 1 Bits 0 to 7 contain the coding

2.2 Code Bits



C7	C6	Data source
0	0	FIFO
0	1	IF2004 USB (control register)

Fig. 2 Data format of data sources

C5	C4	C3	Data source	
			FIFO	IF2004 USB
0	0	0	Sensor channel 1	Register write command
0	0	1	Sensor channel 2	Register read request
0	1	0	Sensor channel 3	Register change
0	1	1	Sensor channel 4	Status output
1	0	0	Inputs (Trigger 1 to 4 sensor RxD 1 to 4)	Reserved
1	0	1	Reserved	Reserved
1	1	0	Reserved	Reserved
1	1	1	Reserved	Reserved

Fig. 3 Data format of channel number or mode

C2	C1	C0	Data source	
			Sensor	IF2004 USB
X	X	X	Byte counter 000 to 111	Byte counter 000 to 111

Fig. 4 Data format of byte counter

i If data blocks containing more than 8 bytes are received, the byte counter stops at 7.

2.3 Examples

All examples with `WORD` access.

2.3.1 Sensor Access

Sensor values are received on channel 1, two values with three bytes each `0x42592b` + `0xc0690e`:

0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	(0x00 = Sensor 1 Byte 0, 0x2b = value 1 LSB)
0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	1	(0x01 = Sensor 1 Byte 1, 0x59 = value 1 NSB)
0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	(0x02 = Sensor 1 Byte 2, 0x42 = value 1 MSB)
0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	(0x03 = Sensor 1 Byte 3, 0x0e = value 2 LSB)
0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	1	(0x04 = Sensor 1 Byte 4, 0x69 = value 2 NSB)
0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	(0x05 = Sensor 1 Byte 5, 0xc0 = value 2 MSB)

Sending a command `+++\0ILD1 0x20 0x00 0x00 0x00` to the sensor on channel 2:

0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	1	(0x08 = Sensor 2 Byte 0, 0x2b = dat. (+))
0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1	(0x09 = Sensor 2 Byte 1, 0x2b = dat. (+))
0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	1	(0x0a = Sensor 2 Byte 2, 0x2b = dat. (+))
0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	(0x0b = Sensor 2 Byte 3, 0x00 = dat. (\0))
0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	(0x0c = Sensor 2 Byte 4, 0x49 = dat. (I))
0	0	0	0	1	1	0	1	0	1	0	0	1	1	0	0	(0x0d = Sensor 2 Byte 5, 0x4c = dat. (L))
0	0	0	0	1	1	1	0	0	1	0	0	0	1	0	0	(0x0e = Sensor 2 Byte 6, 0x44 = dat. (D))
0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1	(0x0f = Sensor 2 Byte 7, 0x31 = dat. (1))
0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	(0x0f = Sensor 2 Byte 8, 0x20 = dat. (0x20))
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	(0x0f = Sensor 2 Byte 9, 0x00 = dat. (0x00))
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	(0x0f = Sensor 2 Byte 10, 0x00 = dat. (0x00))
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	(0x0f = Sensor 2 Byte 11, 0x00 = dat. (0x00)

2.3.2 Register Write Access

When writing to a register, four tuples, consisting of two address tuples and two data tuples, are sent (byte counter 0 to 3). After receiving the tuple with the counter 3, the register is set.

Writing to register 0x0020 with value `0x1234`:

0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	(0x40 = UIF WR Byte 0, 0x20 = addr. LSB)
0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	(0x41 = UIF WR Byte 1, 0x00 = addr. MSB)
0	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	(0x42 = UIF WR Byte 2, 0x34 = dat. LSB)
0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	0	(0x43 = UIF WR Byte 3, 0x12 = dat. MSB)

2.3.3 Register Read Access

If there is a read request for a register, two tuples (address) are sent, and subsequently two tuples (data) are received.
Reading the register 0x0005:

0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	(0x48 = UIF RD Byte 0, 0x05 = addr. LSB)
0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	(0x49 = UIF RD Byte 1, 0x00 = addr. MSB)

Reply = address + data, e.g., `0xA062`:

0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	(0x48 = UIF RD Byte 0, 0x05 = addr. LSB)
0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	(0x49 = UIF RD Byte 1, 0x00 = addr. MSB)
0	1	0	0	1	0	1	0	0	1	1	0	0	0	1	0	(0x4A = UIF RD Byte 2, 0x62 = dat. LSB)
0	1	0	0	1	0	1	1	1	0	1	0	0	0	0	0	(0x4B = UIF RD Byte 3, 0xA0 = dat. MSB)

2.3.4 Register Updating

If a register is changed, 6 tuples (two address tuples, two data tuples and two bit mask tuples) are sent.
Updating register 0x0012, bits 0 to 3 with `0x000A`:

0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	(0x50 = UIF UD Byte 0, 0x12 = addr. LSB)
0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	(0x51 = UIF UD Byte 1, 0x00 = addr. MSB)
0	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	(0x52 = UIF UD Byte 2, 0x0A = dat. LSB)
0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	(0x53 = UIF UD Byte 3, 0x00 = dat. MSB)
0	1	0	1	0	1	0	0	0	0	0	0	1	1	1	1	(0x54 = UIF UD Byte 4, 0x0F = mask LSB)
0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	(0x55 = UIF UD Byte 5, 0x00 = mask MSB)

2.3.5 Register Status Output

Status output is enabled automatically if an error flag has been set. Error flags, see Fig. 23: Status register, bits 8 to 12 (example for FIFO overflow).
Output = address (FIX 0x001A) + status, e.g., `0x1000`:

0	1	0	1	1	0	0	0	0	0	0	1	1	0	1	0	(0x58 = UIF status byte 0, 0x1A = addr. LSB)
0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	(0x59 = UIF status byte 1, 0x00 = addr. MSB)
0	1	0	1	1	0	1	0	0	1	1	0	0	0	1	0	(0x5A = UIF status byte 2, 0x00 = Status bit 0-7)
0	1	0	1	1	0	1	1	1	0	1	0	0	0	0	0	(0x5B = UIF status byte 3, 0x10 = Status bit 0-7)

3. Register, Address Assignment

Local Address Assignment

Base addr. +	Write access	Read access
00h	Sensor 1 baud rate	Sensor 1 baud rate
02h	Sensor 2 baud rate	Sensor 2 baud rate
04h	Sensor 3 baud rate	Sensor 3 baud rate
06h	Sensor 4 baud rate	Sensor 4 baud rate
08h	Timer 1 frequency	Timer 1 frequency
0Ah	Timer 1 pulse width	Timer 1 pulse width
0Ch	Timer 2 frequency	Timer 2 frequency
0Eh	Timer 2 pulse width	Timer 2 pulse width
10h	Timer 1 clock splitter Timer 2 clock splitter FIFO enable register	Timer 1 clock splitter Timer 2 clock splitter FIFO enable register
12h	Trigger outputs mode (sensor) Parity enable register	Trigger outputs mode (sensor) Parity enable register
14h	LED mode TxD outputs mode (sensor)	LED mode TxD outputs mode (sensor)
16h	Trigger outputs mode (ext.) Trigger inputs inverting (ext.)	Trigger outputs mode (ext.) Inverting of trigger inputs
18h	Release code for register write access and flash memory	Release code for register write access and flash memory
1Ah	Reset register	Status register
1Ch	No function	FPGA and hardware versions
1Eh	No function	Reserved for testing purposes
20h	RS422 baud rate	RS422 baud rate
22h	RS422 mode Option field	RS422 mode option field

Fig. 5 Local address assignment

4. Register Description

4.1 Baud Rate

Base addr.	Sensor channel	Value	Access
+ 00h	Sensor channel 1	5 to 65,535	Read and write access
+ 02h	Sensor channel 2	5 to 65,535	Read and write access
+ 04h	Sensor channel 3	5 to 65,535	Read and write access
+ 06h	Sensor channel 4	5 to 65,535	Read and write access
+ 20h	RS422 interface	5 to 65,535	Read and write access

Fig. 6 Base addresses for sensor and RS422 baud rates

Formula: Value = (48 MHz / baud rate) – 1

Example:
desired baud rate = 691.2 kBaud
Value = (48 MHZ / 691,200) - 1 = 68.44

The input value must be a whole number, i.e., the result must still be rounded:
Value = 68

4.2 Timer

Base addr.	Timer	Value	Access
+ 08h	Timer 1 frequency	0 to 65,535	Read and write access
+ 0Ah	Timer 1 pulse width	0 to 65,535	Read and write access
+ 0Ch	Timer 2 frequency	0 to 65,535	Read and write access
+ 0Eh	Timer 2 pulse width	0 to 65,535	Read and write access

Fig. 7 Base addresses for timer

Formula	$Value(F) = (F_{Clock} / F_{OUT}) - 1$
	$Value(PW) = (PW_{OUT} / T_{Clock})$

Example:
Desired frequency $F_{OUT} = 10$ kHz

Desired pulse width $PW_{OUT} = 25\ \mu s$

Clock splitter = 0 > F_{Clock} = 24 MHZ (T_{Clock} = 41.67 ns)

Value F_{OUT} = (24 MHZ / 10 kHz) – 1 = 2399

Value PW = (25 μs / 41.67 ns) = 600

i The input values must be whole numbers.

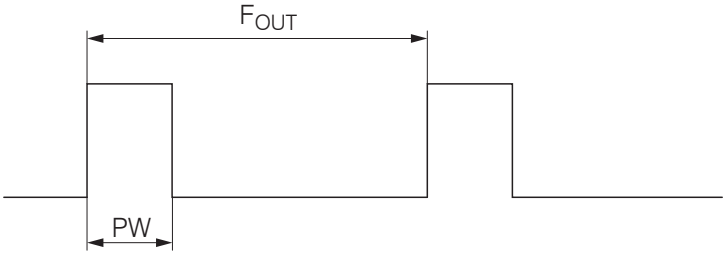


Fig. 8 Timer frequency and pulse width

i To turn off the timer, the frequency must be programmed to be 0. If pulse width > 0 is programmed when the timer is turned off, output is permanently set to High.
However, if the pulse width is also programmed to be 0, output is permanently set to Low.

4.3 Clock Splitter and FIFO Enable Register

Base addr. + 10h (read and write access)

4.3.1 Overview of Functions

Bit	Function	Reference
0 to 3	Timer 1 clock splitter	Table 15: Timer clock splitter
4 to 7	Timer 2 clock splitter	Table 15: Timer clock splitter
8 to 15	FIFO enable register	Table 16: FIFO enable register

Fig. 9 Overview of functions for clock splitter and FIFO enable register

4.3.2 Clock Splitter

Bit 3	Bit 2	Bit 1	Bit 0	Clock frequency timer 1	
Bit 7	Bit 6	Bit 5	Bit 4	Clock frequency timer 2	
0	0	0	0	24 MHz	24 MHz
0	0	0	1	24 MHz / 2	12 MHz
0	0	1	0	24 MHz / 4	6 MHZ
0	0	1	1	24 MHz / 8	3 MHz
0	1	0	0	24 MHz / 16	1.5 MHz
0	1	0	1	24 MHz / 32	750.0 kHz
0	1	1	0	24 MHz / 64	375.0 kHz
0	1	1	1	24 MHz / 128	187.5 kHz
1	0	0	0	24 MHz / 256	93.75 kHz
1	0	0	1	24 MHz / 512	46.88 kHz
1	0	1	0	24 MHz / 1024	23.44 kHz
1	0	1	1	24 MHz / 2048	11.72 kHz
1	1	0	0	24 MHz / 4096	5.859 kHz
1	1	0	1	24 MHz / 8192	2.930 kHz
1	1	1	0	24 MHz / 16,384	1.465 kHz
1	1	1	1	24 MHz / 32,768	732.4 Hz

Fig. 10 Timer clock splitter

4.3.3 FIFO enable register:

Bit	Function
8	0 = FIFO for sensor channel 1 disabled 1 = FIFO for sensor channel 1 enabled
9	0 = FIFO for sensor channel 2 disabled 1 = FIFO for sensor channel 2 enabled
10	0 = FIFO for sensor channel 3 disabled 1 = FIFO for sensor channel 3 enabled
11	0 = FIFO for sensor channel 4 disabled 1 = FIFO for sensor channel 4 enabled
12	0 = FIFO for trigger IN and RxD input disabled 1 = FIFO for trigger IN and RxD input enabled
13	0 = FIFO is disabled for sensor 1 and 2 if trigger IN 1 is active 1 = IN 1 does not affect FIFO
14	0 = FIFO is disabled for sensor 3 and 4 if trigger IN 2 is active 1 = IN 2 does not affect FIFO
15	0 = FIFO is disabled for trigger IN and RxD input if trigger IN 3 is active 1 = IN 3 does not affect FIFO

Fig. 11 FIFO enable register

4.4 Trigger Output Mode (Sensor) and Parity Enable Register

Base addr. + 12h (read and write access)

4.4.1 Overview of Functions

Bit	Function	Reference
0 to 2	Trigger output 1 mode	Fig. 14: Trigger outputs mode (sensor)
3 to 5	Trigger output 2 mode	
6 to 8	Trigger output 3 mode	
9 to 11	Trigger output 4 mode	
12 to 15	Parity enable register	Fig. 13: Parity enable register

Fig. 12 Overview of functions, trigger outputs mode, inverting of trigger inputs

4.4.2 Parity Enable Register:

Bit	Function
12	0 = Parity bit for sensor channel 1 disabled 1 = Parity bit for sensor channel 1 enabled (only even parity)
13	0 = Parity bit for sensor channel 2 disabled 1 = Parity bit for sensor channel 2 enabled (only even parity)
14	0 = Parity bit for sensor channel 3 disabled 1 = Parity bit for sensor channel 3 enabled (only even parity)
15	0 = Parity bit for sensor channel 4 disabled 1 = Parity bit for sensor channel 4 enabled (only even parity)

Fig. 13 Parity enable register

4.4.3 Trigger Output Mode (Sensor):

Bit	Function			
0 to 2	Bit 2	Bit 1	Bit 0	Function
	0	0	0	Sensor trigger 1 switches with ext. trigger IN 1
	0	0	1	Sensor trigger 1 switches with ext. trigger IN 2
	0	1	0	Sensor trigger 1 switches with ext. trigger IN 3
	0	1	1	Sensor trigger 1 switches with ext. trigger IN 4
	1	0	0	Sensor trigger 1 switches with timer 1
	1	0	1	Sensor trigger 1 switches with timer 2
	1	1	0	Sensor trigger 1 permanently LO
	1	1	1	Sensor trigger 1 permanently HI
3 to 5	Bit 5	Bit 4	Bit 3	Function
	0	0	0	Sensor trigger 2 switches with ext. trigger IN 1
	0	0	1	Sensor trigger 2 switches with ext. trigger IN 2
	0	1	0	Sensor trigger 2 switches with ext. trigger IN 3
	0	1	1	Sensor trigger 2 switches with ext. trigger IN 4
	1	0	0	Sensor trigger 2 switches with timer 1
	1	0	1	Sensor trigger 2 switches with timer 2
	1	1	0	Sensor trigger 2 permanently LO
	1	1	1	Sensor trigger 2 permanently HI
6 to 8	Bit 8	Bit 7	Bit 6	Function
	0	0	0	Sensor trigger 3 switches with ext. trigger IN 1
	0	0	1	Sensor trigger 3 switches with ext. trigger IN 2
	0	1	0	Sensor trigger 3 switches with ext. trigger IN 3
	0	1	1	Sensor trigger 3 switches with ext. trigger IN 4
	1	0	0	Sensor trigger 3 switches with timer 1
	1	0	1	Sensor trigger 3 switches with timer 2
	1	1	0	Sensor trigger 3 permanently LO
	1	1	1	Sensor trigger 3 permanently HI
9 to 11	Bit 11	Bit 10	Bit 9	Function
	0	0	0	Sensor trigger 4 switches with ext. trigger IN 1
	0	0	1	Sensor trigger 4 switches with ext. trigger IN 2
	0	1	0	Sensor trigger 4 switches with ext. trigger IN 3
	0	1	1	Sensor trigger 4 switches with ext. trigger IN 4
	1	0	0	Sensor trigger 4 switches with timer 1
	1	0	1	Sensor trigger 4 switches with timer 2
	1	1	0	Sensor trigger 4 permanently LO
	1	1	1	Sensor trigger 4 permanently HI

Fig. 14 Trigger output mode (sensor)

4.5 LED Mode and TxD Outputs Mode

Base addr. + 14h (read and write access)

4.5.1 Overview of Functions

Bit	Function	Reference
0 and 1	LED 1 mode	Fig. 16: Overview of functions LED mode
2 and 3	LED 2 mode	
4 and 5	LED 3 mode	
6 and 7	LED 4 mode	
8 and 9	TxD output 1 mode	Fig. 17: TxD outputs mode
10 and 11	TxD output 2 mode	
12 and 13	TxD output 3 mode	
14 and 15	TxD output 4 mode	

Fig. 15 Overview of functions LED mode and TxD outputs mode

4.5.2 LED Mode

Bit	Function		
0 and 1	Bit 1	Bit 0	Function
	0	0	LED 1 lights up with USB ready
	0	1	LED 1 lights up with trigger input 1
	1	0	LED 1 lights up with receiver input 1 (RxD)
	1	1	LED 1 lights up with transmitter output 1 (TxD)
2 and 3	Bit 3	Bit 2	Function
	0	0	LED 2 lights up with external power +24 Volt
	0	1	LED 2 lights up with trigger input 2
	1	0	LED 2 lights up with receiver input 2 (RxD)
	1	1	LED 2 lights up with transmitter output 2 (TxD)
4 and 5	Bit 5	Bit 4	Function
	0	0	LED 3 lights up when data in FIFO
	0	1	LED 3 lights up with trigger input 3
	1	0	LED 3 lights up with receiver input 3 (RxD)
	1	1	LED 3 lights up with transmitter output 3 (TxD)
6 and 7	Bit 7	Bit 6	Function
	0	0	LED 4 lights up with data transmitter (TxD 1-4)
	0	1	LED 4 lights up with trigger input 4
	1	0	LED 4 lights up with receiver input 4 (RxD)
	1	1	LED 4 lights up with transmitter output 4 (TxD)

Fig. 16 Overview of functions LED mode

4.5.3 TxD Outputs Mode:

Bit	Function		
8 and 9	Bit 9	Bit 8	Function
	0	0	TxD 1 switches with transmitter sensor channel 1
	0	1	TxD 1 permanently LO
	1	0	TxD 1 permanently HI
	1	1	TxD 1 switches with transmitter sensor channel 1-4
10 and 11	Bit 11	Bit 10	Function
	0	0	TxD 2 switches with transmitter sensor channel 2
	0	1	TxD 2 permanently LO
	1	0	TxD 2 permanently HI
	1	1	TxD 2 switches with transmitter sensor channel 1-4
12 and 13	Bit 13	Bit 12	Function
	0	0	TxD 3 switches with transmitter sensor channel 3
	0	1	TxD 3 permanently LO
	1	0	TxD 3 permanently HI
	1	1	TxD 3 switches with transmitter sensor channel 1-4
14 and 15	Bit 15	Bit 14	Function
	0	0	TxD 4 switches with transmitter sensor channel 4
	0	1	TxD 4 permanently LO
	1	0	TxD 4 permanently HI
	1	1	TxD 4 switches with transmitter sensor channel 1-4

Fig. 17 TxD outputs mode

4.6 Trigger Outputs Mode, Inverting of External Trigger Inputs

Base addr. + 16h (read and write access)

4.6.1 Overview of Functions

Bit	Function	Reference
0 to 2	Mode ext. trigger output 1	Fig. 19: Mode ext. trigger outputs
3	Reserved	
4 to 6	Mode ext. trigger output 2	Fig. 19: Mode ext. trigger outputs
7	Reserved	
8 and 9	FIFO latch source (for trigger and sensor inputs)	Fig. 20: FIFO latch source (for trigger and sensor inputs)
10 and 11	Reserved	
12 to 15	Inverting of ext. trigger input 1-4	Fig. 21: Inverting of trigger inputs

Fig. 18 Overview of functions ext. trigger outputs mode and inverting of external trigger inputs

4.6.2 Trigger Outputs Mode

Bit	Function			
0 to 2	Bit 2	Bit 1	Bit 0	Function
	0	0	0	Ext. trigger OUT 1 switches with sensor channel 1
	0	0	1	Ext. trigger OUT 1 switches with sensor channel 2
	0	1	0	Ext. trigger OUT 1 switches with sensor channel 3
	0	1	1	Ext. trigger OUT 1 switches with sensor channel 4
	1	0	0	Ext. trigger OUT 1 switches with timer 1
	1	0	1	Ext. trigger OUT 1 switches with timer 2
	1	1	0	Ext. Trigger OUT 1 permanently LO
	1	1	1	Ext. Trigger OUT 1 permanently HI
4 to 6	Bit 6	Bit 5	Bit 4	Function
	0	0	0	Ext. trigger OUT 2 switches with sensor channel 1
	0	0	1	Ext. trigger OUT 2 switches with sensor channel 2
	0	1	0	Ext. trigger OUT 2 switches with sensor channel 3
	0	1	1	Ext. trigger OUT 2 switches with sensor channel 4
	1	0	0	Ext. trigger OUT 2 switches with timer 1
	1	0	1	Ext. trigger OUT 2 switches with timer 2
	1	1	0	Ext. Trigger OUT 2 permanently LO
	1	1	1	Ext. Trigger OUT 2 permanently HI

Fig. 19 Ext. trigger outputs mode

4.6.3 FIFO Latch Source

For triggers and sensor inputs

Bit	Function		
8 and 9	Bit 9	Bit 8	Function
	0	0	Sensor channel 1
	0	1	Sensor channel 3
	1	0	Timer 1
	1	1	Timer 2

Fig. 20 FIFO latch source (for triggers and sensor inputs)

i Bits 8 and 9 can be used to select a latch source whose trigger event writes the external trigger inputs (IN 1 to IN 4) and the RxD inputs (sensor 1 to 4) to FIFO.

4.6.4 Inverting of Trigger Inputs:

Bit	Function
12	0 = Trigger input 1 normal (HI active) 1 = Trigger input 1 inverted (LO active)
13	0 = Trigger input 2 normal (HI active) 1 = Trigger input 2 inverted (LO active)
14	0 = Trigger input 3 normal (HI active) 1 = Trigger input 3 inverted (LO active)
15	0 = Trigger input 4 normal (HI active) 1 = Trigger input 4 inverted (LO active)

Fig. 21 Inverting of trigger inputs

4.7 Release Code For Register Write Access and Flash Memory

Base addr. + 18h (read and write access)

Release code for register write access

To be able to write to the registers, a release code must first be written to base address `18h`.

Release code for register write access: `0xD5EA`

If address `18h` is not equal to the release code, all write access (register and sensor) is disabled.

Code for flash memory read and write access

The codes below can be used to save the FPGA registers in the flash memory and read them back.

Enable read-only: `0x3B10`

Disable read-only: `0x3B13`

Write FPGA registers to the flash memory: `0x3B14`

Load FPGA registers from the flash memory: `0x3B18`

Before FPGA registers can be written to the flash memory, read-only mode `0x3B13` must be disabled. After the FPGA registers have been written to the flash memory, read-only mode `0x3B10` should be re-enabled for safety reasons. To read the flash memory back into the FPGA registers, code `0x3B18` is sufficient, read-only mode does not need to be disabled.

4.8 Reset Register

Base addr. + 1Ah (only write access)

Bit	Function
0	Delete FIFO
1	Delete parity error and FIFO overflow flag
2	Reserved
3	Send RX buffer immediately
4 to 15	Reserved

Fig. 22 Reset register

Writing 1 to bit 0 deletes FIFO.
Writing 1 to bit 1 deletes the error flags (status register, bits 8 to 12).
The bits only need to be set, resetting them is not necessary.
If an error occurs, the status register, see Fig. 23, is automatically output to the USB interface:

Example for FIFO overflow: 0x58 = UIF status byte 0, 0x1A = addr. LSB
0x59 = UIF status byte 1, 0x00 = addr. MSB
0x5A = UIF status byte 2, 0x00 = status bit 0 to 7
0x5B = UIF status byte 3, 0x10 = status bit 8 to 15

The error flags are maintained until they are reset by setting the reset register bit 1.

4.9 Status Register

Base addr. + 1Ah, only read access

Bit	Function
0	1 = external trigger IN 1 active
1	1 = external trigger IN 2 active
2	1 = external trigger IN 3 active
3	1 = external trigger IN 4 active
4	1 = RxD input 1 active
5	1 = RxD input 2 active
6	1 = RxD input 3 active
7	1 = RxD input 4 active
8	1 = Parity error 1 active
9	1 = Parity error 2 active
10	1 = Parity error 3 active
11	1 = Parity error 4 active
12	1 = FIFO overflow
13	1 = EEPROM access active
14	1 = RS422 transmitter access active
15	Reserved

Fig. 23 Status register

4.10 FPGA and Hardware Versions

Base addr. + 1Ch, only read access

Overview of functions

Bit	Function
0 to 7	FPGA version
8 to 15	Hardware version

Fig. 24 FPGA and hardware versions

4.11 RS422 Mode

Base addr. + 22h, read and write access

4.11.1 Overview of Functions

Bit	Function	Reference
0 and 1	RS422 data output	Fig. 26: RS422 data output
2	Reserved	
3	Comparator mode (min or max)	Fig. 27: Comparator mode
4 to 7	Sensor selection for comparator mode	Fig. 28: Sensor selection for comparator mode
8 to 10	Selecting the trigger source for comparator value output to the RS422 interface	Fig. 29: Trigger source for comparator value output
11	Reserved	
12 to 15	Option field	Freely available

Fig. 25 Overview of functions RS422 mode

4.11.2 RS422 Data Output:

Bit	Function		
0 and 1	Bit 1	Bit 0	Function
	0	0	RS422 interface parallel to sensor 1 [1]
	0	1	Output comparator value (3-byte mode) [2]
	1	0	Coded data output (only register) [3]
	1	1	Coded data output (FIFO and register) [4]

Fig. 26 RS422 data output

1) Sensor 1 can be connected either using the sub-D plug on the back or the 6-pin terminal connector on the front. Data lines RxD or TxD of the plugs on the front or back are switched in parallel.

2) 3-byte mode corresponds to the data format of sensor ILD2300 and is only output to the RS422 interface. In this mode, register values and FIFO data are only output to the USB interface. If an error occurs at a selected sensor (sensor outside measuring range or sensor does not output measurements), error messages are generated:

0x3FFF0 No sensor selected for comparator mode (bits 4 to 7 = 0)
0x3FFF1 Error at sensor 1
0x3FFF2 Error at sensor 2
0x3FFF4 Error at sensor 3
0x3FFF8 Error at sensor 4

i If sensor 1 or sensor 3 is selected as trigger source (bits 8 to 10) and the data line for the selected sensor is interrupted, an error message cannot be generated. We therefore recommend that an internal timer should be used as the trigger source.

3) If coded data output (register only) has been selected, the register values can be written to and read from the USB and the RS422 interfaces.

4) If coded data output (FIFO and register) has been selected, the register values can be written to and read from the USB and the RS422 interfaces. FIFO data are no longer output at the USB interface, but only at the RS422 interface.

4.11.3 Comparator Mode

Bit	Function
3	0 = Minimum value (smallest value of up to 4 sensors) 1 = Maximum value (largest value of up to 4 sensors)

Fig. 27 Comparator mode

4.11.4 Sensor Selection for Comparator Mode

Bit	Function
4	0 = Sensor 1 for comparator mode disabled 1 = Sensor 1 for comparator mode enabled
5	0 = Sensor 2 for comparator mode disabled 1 = Sensor 2 for comparator mode enabled
6	0 = Sensor 3 for comparator mode disabled 1 = Sensor 3 for comparator mode enabled
7	0 = Sensor 4 for comparator mode disabled 1 = Sensor 4 for comparator mode enabled

Fig. 28 Sensor selection for comparator mode

4.11.5 Selecting the Trigger Source for Comparator Value Output to the RS422 Interface

Bit	Function			
8 to 10	Bit 10	Bit 9	Bit 8	Trigger source for comparator value output
	0	0	0	Timer 1
	0	0	1	Timer 2
	0	1	0	Sensor 1
	0	1	1	Sensor 3
	1	0	0	External trigger IN 1
	1	0	1	External trigger IN 2
	1	1	0	External trigger IN 3
	1	1	1	External trigger IN 4

Fig. 29 Trigger source for comparator value output

5. Liability for Material Defects

All components of the device have been checked and tested for functionality at the factory. However, if defects occur despite our careful quality control, MICRO-EPSILON or your dealer must be notified immediately.

The liability for material defects is 12 months from delivery. Within this period, defective parts, except for wearing parts, will be repaired or replaced free of charge, if the device is returned to MICRO-EPSILON with shipping costs prepaid. Any damage that is caused by improper handling, the use of force or by repairs or modifications by third parties is not covered by the liability for material defects. Repairs are carried out exclusively by MICRO-EPSILON.

Further claims can not be made. Claims arising from the purchase contract remain unaffected. In particular, MICRO-EPSILON shall not be liable for any consequential, special, indirect or incidental damage. In the interest of further development, MICRO-EPSILON reserves the right to make design changes without notification.
For translations into other languages, the German version shall prevail.

MICRO-EPSILON MESSTECHNIK GmbH & Co. KG
Königbacher Str. 15 · 94496 Ortenburg/Germany
Tel. +49 (0) 8542/168-0 · Fax +49 (0) 8542/168-90
info@micro-epsilon.de · www.micro-epsilon.de

X9751407-A011079MSC