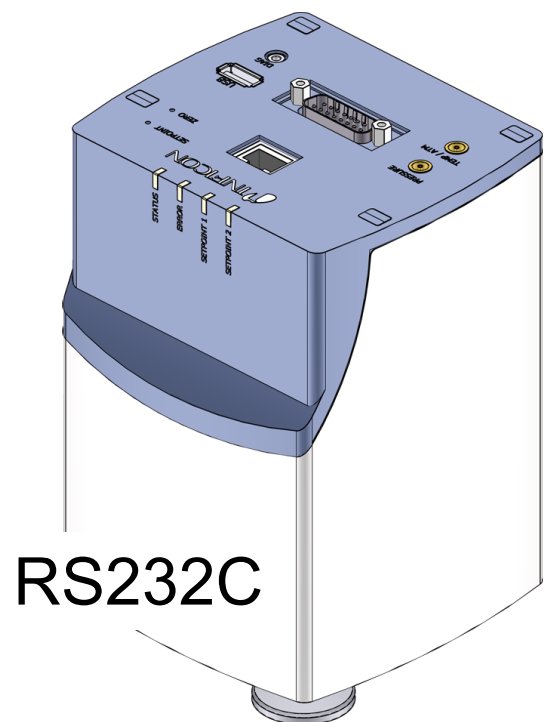


RS232C

Serial Interface

Cube CDGsci



General Information

The RS232C Serial Interface for Cube permits the communication between this digital INFICON Capacitance Diaphragm Gauge and

- an appropriate controller or
- a computer.

The RS232C Serial Interface integrated in the Capacitance Diaphragm Gauge allows to digitally transmit measurement values and information on the gauge status as well as to make parameter settings.

Caution

Data transmission errors

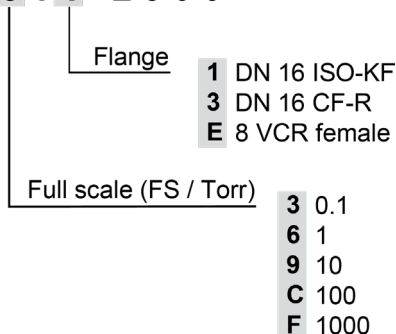
Any attempt to simultaneously operate the gauge via the RS232C Serial Interface and a fieldbus interface (DeviceNet or Profibus) or the diagnostic port may result in incorrect data and data transmission errors.

Therefore, it is inadmissible to simultaneously operate the gauge via the RS232C Serial Interface and DeviceNet, Profibus, or the diagnostic port.

Validity

This document applies to products with part numbers

3 C S 1 - 3 1 1 - 2 3 0 0



The part number (PN) can be taken from the product nameplate.

If not indicated otherwise in the legends, the illustrations in this document correspond to CDGsci gauges with the DN 16 ISO-KF vacuum connection.

We reserve the right to make technical changes without prior notice.

Functional Principle

The RS232C Serial Interface is used in duplex operation. The gauge continuously (approximately every 100 ms) transmits a nine byte send string without request.

Instructions to the gauge are transmitted via five-byte receipt strings.

The RS232C Serial Interface is only available on the DIAG port.

Data format

- binary
- 8 data bits
- 1 stop bit
- no parity bit
- no handshake

Transmission rate

- 9600 Baud

DIAG connector assignment

- TxD inner pin
- RxD middle ring
- GND outer ring

Response time

100 ms for pressure values
200 ... 1000 ms for other information or write commands

1 Interface Protocol

1.1 Send String

The complete send string (frame) is nine bytes (byte 0 ... 8) long. Bytes 1 ... 7 form the data string.

Structure of send string

Byte No.	Function	Value	Comment
0	Data string length	7	Constant value
1	Page No.	4	Constant value for CDGsci ¹⁾
2	Status		→ "Status byte"
3	Error		→ "Error byte"
4	Measured value high byte	<value>	→ "Calculation of pressure value"
5	Measured value mid byte	<value>	→ "Calculation of pressure value"
6	Read command	<value>	Read value
7	Measured value low byte ¹⁾	<value>	→ "Calculation of pressure value"
8	Checksum		→ "Synchronization"

¹⁾ The "Sensor Type" that is accessible via Byte 7 in all other CDGs can now be deduced from address 56 (Exponent) and 57 (Mantissa).

Status byte
(byte No. 2)

Bit 0	Definition
0	Continuous output of measured value
1	Individual measured value (polling) ¹⁾

¹⁾ → § 8, "DataTxMode". Send a read command of any parameter to the gauge for requesting a send string.

Bit 2	Bit 1	Definition
1	0	Manual setpoint setting
1	1	Zero adjust active

Bit 3	Definition
0 ⇔ 1	Toggle bit, changes with every string received correctly

Bit 5	Bit 4	Definition
0	0	Current pressure unit mbar
0	1	Current pressure unit Torr
1	0	Current pressure unit Pa

Bit 6	Definition
0	Standard measurement mode
1	Reserved for internal use

Bit 7	Definition
0	Heating
1	Sensor temperature attained

Error byte
(byte No. 3)

Bit No.	Definition
0	RS232 synchronization error
1	Incorrect command, e.g. inadmissible address (syntax error)
2	Inadmissible read command
3	SP1 status
4	SP2 status
5	Not used
6	Not used
7	Extended error set (→ Read command "Extended Error L-Byte and H-Byte")

No bit set → value = 0x00 = no error set

Error handling

Errors are only recorded in the error bytes as long as they persist, except for RS232 interface errors. Errors are not acknowledged.

RS232 errors are signaled by the "toggle bit", i.e. when an RS232 error occurs, the "toggle bit" is not inverted. For checking the status of the "toggle bit", a read operation is required, which also allows to read the error byte for detailed error analysis.

If an "extended error" is set, it has to be read as variable by means of the "read command" (→ table "Variables for bytes No. 2 and 3"). After the read operation, the variable is automatically erased.

Calculation of pressure value

The pressure is calculated from bytes 4, 5 and 7 of the send string (decimal presentation).

Conversion formula
(byte No. 4, 5 and 7)

$$p = \frac{\langle \text{pressure_value} \rangle \times a}{b} \times \text{FS_Mantissa} \times 10^{(\text{FS_Exp})}$$

The exponent (Exp) and Mantissa are stored in address 56 and 57, respectively.

Parameter	Description												
p	Pressure value in selected pressure unit (→ Parameter a)												
⟨pressure_value⟩	Pressure measurement data, composed of "low and high byte" (16 bit value) and converted into decimal format												
FS_Mantissa	FS factor according to "Sensor type" variable, which has to be read separately (→ "Read command")												
FS_Exp	FS exponent according to "Sensor type" variable, which has to be read separately (→ "Read command")												
a	Conversion factor for pressure units other than "Torr". Torr: a = 1.00 mbar: a = 1.3332 Pa: a = 133.32												
b	Factor for resolution												
	<table border="1"> <thead> <tr> <th>Page No. ¹⁾</th> <th>b</th> <th>Output signal</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>8'192'000</td> <td>10.20 V</td> </tr> <tr> <td>3</td> <td>8'192'000</td> <td>10.20 V</td> </tr> <tr> <td>4</td> <td>8'388'352</td> <td>10.00 V</td> </tr> </tbody> </table>	Page No. ¹⁾	b	Output signal	2	8'192'000	10.20 V	3	8'192'000	10.20 V	4	8'388'352	10.00 V
Page No. ¹⁾	b	Output signal											
2	8'192'000	10.20 V											
3	8'192'000	10.20 V											
4	8'388'352	10.00 V											

¹⁾ → 4, tabel "Structure of send string", byte no. 1

Read command (byte No. 6)

All variables in a receipt string that are addressed for reading are output on this byte. For variable types >1 byte, each byte (e.g. low, high, or further bytes) has to be addressed and read individually.

Read Command L-Byte → Read Data L-Byte
Read Command H-Byte → Read Data H-Byte

- After a write operation, the value of the addressed variable is output.
- After a reset (Power on) the software version is output on byte 6.

Checksum and synchronization

The recipient (master) is synchronized by checking three bytes:

Byte No.	Function	Value	Comment
0	Data string length	7	Constant value
1	Page No.	7	Constant value for CDGsci
8	Checksum of bytes No. 1 ... 7	0 ... 255	Low byte of checksum ¹⁾

¹⁾ Possible high bytes are ignored

Example

The example is based on the following output string:

Byte No.	0	1	2	3	4	5	6	7	8
Value	7	2	16	0	125	0	20	0	69

The instrument or controller (receiver) interprets this string as follows:

Byte No.	Function	Value	Comment
0	Length of datastring	7	(Set value)
1	Page No.	7	CDGsci
2	Status	16	Pressure unit = Torr
3	Error	0	No Error
4	Measurement High byte	125	Calculation of pressure value: Conversion formula → 5
5	Measurement Mid byte	0	
6	Read command	20	Software version = 20 / 20 = 1.0
7	Measurement Low byte	0	Calculation of pressure value: Conversion formula → 5
8	Check sum	169	$2 + 16 + 0 + 125 + 0 + 20 + 6 = 169_{dec} \hat{=} 00A9_{hex}$ High order byte is ignored ⇒ Check sum = $A9_{hex} \hat{=} 169_{dec}$

1.2 Receipt String

Commands to the gauge are transmitted in receipt strings (frames) consisting of five bytes (without <CR>). The data string is formed by bytes 1 ... 3.

Structure of receipt string

Byte No.	Designation	Value	
0	Data string length	3	(constant value)
1	Data		→ "Service command"
2	Data		→ "Address byte"
3	Data		→ "Data byte"
4	Checksum of bytes No. 1 ... 3	<value>	Low byte of checksum ¹⁾

¹⁾ Possible high bytes are ignored.

- The operation selected in byte No. 1 is addressed in byte No. 2.
- Variables are transmitted in byte No. 3. Variables >1 byte have to be transmitted in several receipt strings (splitting).

Service command
(byte No. 1)

For values for byte 1 see parameter table in section 2.

Address byte
(byte No. 2)

Enter the address of the variable to be read/written (→ table "Variables for bytes No. 2 and 3").

Data byte
(byte No. 3)

When a variable is written (receipt string) the content of byte No. 3 is written to the variable addressed in byte No. 2 (→ table "Variables for bytes No. 2 and 3").

When a variable is read (send string), the value of the variable addressed in byte No. 2 is output in byte No. 6 of the send string. The content of byte No. 3 is not relevant for read operations.

Checksum
(byte No. 4)

The checksum is calculated from the sum of byte No. 1 to 3.

Example

The example is based on receive string:

Byte No.	0	1	2	3	4
Value	3	0	2	0	2

The instrument or controller (receiver) interprets this string as follows:

Byte No.	Designation	Value	Comment
0	Data string length	3	(constant value)
1	Service command	0	Read command
2	Address byte	2	Filter
3	Data byte	0	
4	Checksum	2	$0 + 2 + 0 = 2_{dec} \hat{=} 00\ 02_{hex}$ High bytes are ignored \Rightarrow Checksum = $02_{hex} \hat{=} 2_{dec}$

2 Parameter Table

All values in this table are quoted in HEX, thus the prefix '0x'.

Parameter name	Data type	Access Right	Byte Type	Byte No. 1	Byte No. 2	Byte No. 3	Comment	
DataTxMode	uint8	R		0x00	0x00	0x00	Continued output of measured value	
		W		0x10		0x01	Individual output of measured value (polling)	
Unit	uint8	R		0x00	0x01	<value>	Device unit of low level signal processor	
FilterSettings	uint8					<value>	Adaptive filter settings	
		R		0x00	0x02	0x00	dynamic	
						0x01	time fast	
						0x02	time slow	
		W		0x10	0x02	0x00	dynamic	
						0x01	time fast	
0x02	time slow							
SP1 Level Low	sint16	R	H-Byte	0x00	0x04	<value>	Lower setpoint threshold SP1 ¹⁾	
			L-Byte			0x05		<value>
		W	H-Byte	0x10	0x04	<value>		
			L-Byte			0x05		<value>
SP2 Level Low	sint16	R	H-Byte	0x00	0x06	<value>	Lower setpoint threshold SP2 ¹⁾	
			L-Byte			0x07		<value>
		W	H-Byte	0x10	0x06	<value>		
			L-Byte			0x07		<value>
SP1 Level High	sint16	R	H-Byte	0x00	0x08	<value>	Upper setpoint threshold SP1 (hysteresis) ¹⁾	
			L-Byte			0x09		<value>
		W	H-Byte	0x10	0x09	<value>		
			L-Byte			0x09		<value>
SP2 Level High	sint16	R	H-Byte	0x00	0x0A	<value>	Upper setpoint threshold SP2 (hysteresis) ¹⁾	
			L-Byte			0x0B		<value>
		W	H-Byte	0x10	0x0A	<value>		
			L-Byte			0x0B		<value>
PerOfAtmSP1	uint8	R		0x00	0xFC	<value>	SP1, percentage of ATM (only for 1000 Torr)	
		W		0x10		<value>		
PerOfAtmSp2	uint8	R		0x00	0xFD	<value>	SP2, percentage of ATM (only for 1000 Torr)	
		W		0x10		<value>		
FirmwareRevisionCPU1	uint8	R		0x00	0x10	<value>	(<value> / 20) = Software Version CPU1 e.g. 20 = V1.0	
Calib date	uint32	R	Byte 3	0x00	0x11	<value>	Date: YY,MM,DD,HH,MM e.g. 0410291109 = 2004-10-29 at 11:09	
		R	Byte 2			0x12		<value>
		R	Byte 1			0x13		<value>
		R	Byte 0			0x14		<value>
ZeroAdjustValue	sint16	R	H-Byte	0x00	0x15	<value>	Zero Offset Adjust Level ²⁾	
		W		0x10				
DCOutputOffset	sint16	R	H-Byte	0x00	0x17	<value>	Customer DC Output Offset ²⁾ (Base pressure offset)	
			W		0x10			
			L-Byte		0x18	<value>		

(continued)

Parameter name	Data type	Access Right	Byte Type	Byte No. 1	Byte No. 2	Byte No. 3	Comment	
SerialNumber	uint32	R	Byte 0	0x00	0x32	<value>	Production number as ASCII string (barcode)	
			Byte 1			0x33		<value>
			Byte 2			0x34		<value>
			Byte 3			0x35		<value>
SWDateYear	uint16	R	H-Byte	0x00	0xD4	<value>	Software version date Year in Hex e.g. 0x2007 = 2007	
			L-Byte			0xD5		<value>
SWDateMonthDay	uint16	R	H-Byte	0x00	0xD6	<value>	Software version date Month in Hex e.g. 0x03 = March	
			L-Byte			0xD7	<value>	Software version date Day in Hex e.g. 0x19 = 19
Part No.	uint8	R	Byte 0	0x00	0xDA	<value>	Part number as ASCII string (Max 20 byte) e.g. 378-000 (Last digit: null terminator)	
			Byte 1			0xDB		<value>
				<value>
			Byte 19			0xED		<value>
RemainingZero	sint16	R	H-Byte	0x00	0x48	<value>	Max. remaining offset value	
			L-Byte			0x49		<value>
ExtendedError H-Byte	uint8	R	H-Byte	0x00	0x36	Bit 0	PT1000 fault	
						Bit 1	Heater block overtemp.	
						Bit 2	Electronic overtemp.	
						Bit 3	Zero adjust error	
						Bit 4	Reserve	
						Bit 5	Reserve	
						Bit 6	Reserve	
						Bit 7	Reserve	
ExtendedError L-Byte	uint8	R	L-Byte	0x00	0x37	Bit 0	Atm. pressure out of range	
						Bit 1	Temperature out of range	
						Bit 2	Reserve	
						Bit 3	Reserve	
						Bit 4	Cal. mode wrong	
						Bit 5	Pressure underflow	
						Bit 6	Pressure overflow	
						Bit 7	Zero adjust warning	
SensPressRange (Exponent)	uint8	R		0x00	0x38	<value>	Full scale of Cube (Exponent): <ul style="list-style-type: none"> • 0 = E-3 • 1 = E-2 • 2 = E-1 • 3 = E 0 • 4 = E+1 • 5 = E+2 • 6 = E+3 • 7 = E+4 	
SensFSR (Mantissa)	uint8	R		0x00	0x39	<value>	Full scale of Cube (Mantissa): <ul style="list-style-type: none"> • 0 = 1.0 • 1 = 1.1 • 2 = 2.0 • 3 = 2.5 • 4 = 5.0 	

(continued)

Parameter name	Data type	Access Right	Byte Type	Byte No. 1	Byte No. 2	Byte No. 3	Comment	
FirmwareRevisionCPU2	uint8	R	Byte 0	0xE0	0x10	<value>	Software version Cube application program e.g. 1.6.0.14 (Last digit: null terminator)	
			Byte 1			<value>		
			...			<value>		
			Byte 19			<value>		
ImageRevisionCPU2	uint8	R	Byte 0	0xE0	0x24	<value>	Image version of Linux operating system (Last digit: null terminator)	
			Byte 1			<value>		
			...			<value>		
			Byte 19			<value>		
SystemDateTime	uint8	R	Byte 0	0xE2	0x00	<value>	Set Date and Time. Format: dd.mm.yyyy hh:mm:ss e.g. 21/10/2013 10:41:00 (Last digit: null terminator)	
		W		0xF2				
		R	Byte 1	0xE2		<value>		
		W		0xF2				
		R	...	0xE2		<value>		
		W		0xF2				
		R	Byte 19	0xE2		<value>		
		W		0xF2				
EthernetLAN	uint8	R		0xE2	0xB9	<value>	Ethernet LAN status: • 0=off • 1=on	
		W		0xF2		0x00		Switch Ethernet LAN off
						0x01		Switch Ethernet LAN on
WirelessLAN	uint8	R		0xE2	0x28	<value>	Wireless LAN status: • 0=off • 1=on	
		W		0xF2		0x00		Switch Wireless LAN off
						0x01		Switch Wireless LAN on
WLANSettings	uint8	R	Byte 0	0xE2	0x51	<value>	TCP-IP Address of current wifi connection Format: IP-Adress Network-Mask Gauge resets after successful IP change (Last digit: null terminator)	
			Byte 1			<value>		
			...			<value>		
			Byte 19			<value>		
LANSettings	uint8	R	Byte 0	0xE2	0x65	<value>	TCP-IP Address of current LAN connection Format: IP-Adress Network-Mask Gauge resets after successful IP change (Last digit: null terminator)	
		W		0xF2				
		R	Byte 1	0xE2		<value>		
		W		0xF2				
		R	...	0xE2		<value>		
		W		0xF2				
		R	Byte 19	0xE2		<value>		
		W		0xF2				

(continued)

(concluded)

Parameter name	Data type	Access Right	Byte Type	Byte No. 1	Byte No. 2	Byte No. 3	Comment	
CustomAnalogOut	uint8	R		0xE2	0x89	<value>	Zoom for analog out • 0=off • 1=on	
		W		0xF2		0x00	Switch Zoom for analog out off	
			0x01	Switch Zoom for analog out on				
AnalogoutPLow	real32	R	Byte 0	0xE2	0x81	<value>	Low pressure threshold for zoom function	
		W		0xF2				
		R	Byte 1	0xE2		<value>	Will set this pressure value to 0.0V	
		W		0xF2				
		R	Byte 2	0xE2		<value>		
		W		0xF2				
		R	Byte 3	0xE2		<value>		
W		0xF2						
AnalogoutPHigh	real32	R	Byte 0	0xE2	0x85	<value>	IEEE Standard for Floating-Point Arithmetic	
		W		0xF2				
		R	Byte 1	0xE2		<value>	Will set this pressure value to 10.0V	
		W		0xF2				
		R	Byte 2	0xE2		<value>		
		W		0xF2				
		R	Byte 3	0xE2		<value>		
W		0xF2						
Pressure	real32	R	Byte 0	0xE2	0x8B	<value>	Pressure value in pressure units IEEE Standard for Floating-Point Arithmetic	
			Byte 1			0x8E		<value>
			Byte 2			0x8F		<value>
			Byte 3			0x91		<value>
CPU2Unit	uint8	R			0x8A	0	Cube pressure unit, 0 = mbar	
		W				1	Cube pressure unit, 1 = Torr	
			2	Cube pressure unit, 2 = Pa				
ATMValue	uint16	R	H-Byte	0xE2	0x92	<value>	ATM value of Cube	
			L-Byte			<value>	Binary 16 bit representation	
MACAddress	uint8	R	Byte 0	0xE2	0x9B	<value>	MAC address of Ethernet port	
			Byte 1			<value>		
			...			<value>		
			Byte 19			<value>		

RW = Read / Write

R = Read only

1) Conversion → Section 1.1, byte 6 "Read command"

2) Conversion → Section 1.1, bytes 4 and 5 "Pressure unit"

Variables for bytes No. 2 and 3 (special services)

Parameter name	Data type	Byte No. 1	Byte No. 2	Byte No. 3	Comment
Reset	uint8 / W	0x40	0x00	0x00	Power reset: Starts continuous pressure output
ResetFactory	uint8 / W	0x40	0x01	0x00	Factory reset: Sets factory configuration
ZeroAdjust	uint8 / W	0x40	0x02	–	Starts zero offset adjustment

W = Write

Description of variables

Setpoint_level xy

$$\text{Setpoint_level xy} = \frac{\langle \text{data_value} \rangle \times a}{b} \times \text{FS_Mantisse} \times 10^{(\text{FS_Exp})}$$

Parameter ¹⁾	Description
Setpoint_level xy	Setpoint threshold in the selected pressure unit.

¹⁾ Further parameter → [13](#), table "Parameter"

- **Minimum value** lower switching threshold = 0 ; negative values are not admissible.
- **Maximum value** lower switching threshold = FS – 1 % hysteresis.

Zero_Adjust_Value

Zero_Adjust_Value contains the zero pressure offset value required for zeroing (writable and readable).

- Automatic zero_adjust function via key or command (→ table "Variables for bytes No. 2 and 3 (special services)").
- Base-Pressure-Adjust for adjusting a defined zero offset, e.g. if the required final pressure as indicated in the operating manual is not reached.

The Zero_Adjust_Value consists of the high and low byte and has to be converted with the "Pressure value" formula (→ [5](#)).

$$\text{Zero_Adjust_Value} = \frac{\langle \text{data_value} \rangle \times a}{b} \times \text{FS_Mantisse} \times 10^{(\text{FS_Exp})}$$

Parameter ¹⁾	Description
Zero_Adjust_Value	Zero pressure offset in the selected pressure unit (→ 13 , table "Parameter").

¹⁾ Further parameter → [13](#), table "Parameter"

Remaining_Zero

Maximal remaining offset value. The Zero_Adjust can only be executed within this value.

DC Output Level

The "DC Output Level" variable is used for assigning a defined offset level to the analog output signal, e.g. for setting a certain zero offset signal level.



A "DC Output Level" >0 reduces the output range of the measurement range 0 ... 10 V by the selected offset value (FS - DC output level).

The "DC Output Level" parameter (16-Bit) consists of the high and low byte.

$$\text{DC Output Level} = \frac{\langle \text{data_value} \rangle \times a}{b} \times \text{FS_Mantisse} \times 10^{(\text{FS_Exp})}$$

Parameter ¹⁾	Description
DC Output Level	DC-Output-Signal in the selected pressure unit (→ 13, table "Parameter").

¹⁾ Further parameter → 13, table "Parameter"

Software version

$$\text{Software version} = \frac{\langle \text{data_value} \rangle}{20} \quad \text{e.g. } \langle \text{data_value} \rangle = 20 \triangleq \text{V1.0}$$

Parameter	Description
<data_value>	1 byte value (8 bit), data value in decimal format.

Parameter

Parameter	Description												
<data_value>	Zero offset measurement data, consisting of "low and high byte" (16 bit value), data value in decimal format.												
a	Conversion factor for pressure units other than "Torr" Torr: a = 1.00 mbar: a = 1.3332 Pa: a = 133.32												
b	Factor for resolution												
	<table border="1"> <thead> <tr> <th>Page No. ¹⁾</th> <th>b</th> <th>Output signal</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>8'192'000</td> <td>10.20 V</td> </tr> <tr> <td>3</td> <td>8'192'000</td> <td>10.20 V</td> </tr> <tr> <td>4</td> <td>8'388'352</td> <td>10.00 V</td> </tr> </tbody> </table>	Page No. ¹⁾	b	Output signal	2	8'192'000	10.20 V	3	8'192'000	10.20 V	4	8'388'352	10.00 V
Page No. ¹⁾	b	Output signal											
2	8'192'000	10.20 V											
3	8'192'000	10.20 V											
4	8'388'352	10.00 V											
FS_Mantisse	FS factor according to the "Sensor type" variable, which has to be read separately (→ "Read command").												
FS_Exp	FS exponent according to "Sensor type" variable, which has to be read separately (→ "Read command").												

¹⁾ → 4, tabel "Structure of send string", byte no. 1

Original: English



tira50e1-a



LI-9496 Balzers
Liechtenstein
Tel +423 / 388 3111
Fax +423 / 388 3700
reachus@inficon.com

www.inficon.com