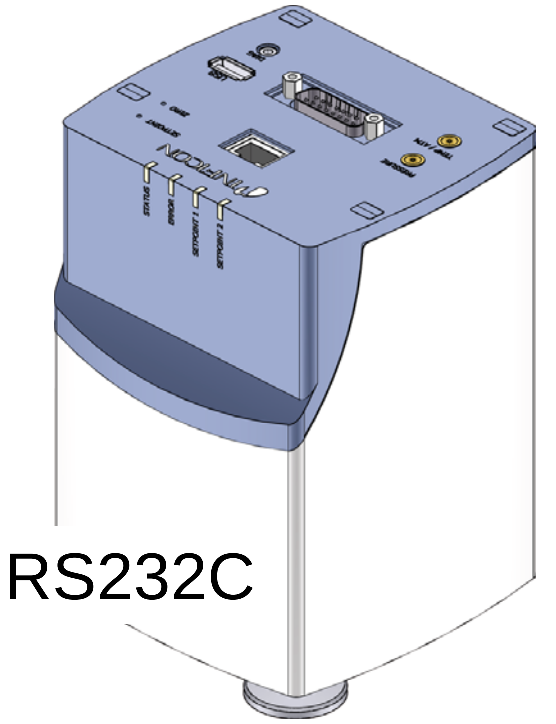
Communication Protocol

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**
>
> 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

Flange

1	DN 16 ISO-KF
3	DN 16 CF-R
E	8VCR female

Measurement range (F.S.)

3	0.1
6	1
9	10
C	100
F	1000

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)

$$p = \frac{\text{<pressure\_value>} \times a}{b} \times \text{F.S.R.\_Mantissa} \times 10^{(\text{F.S.R.\_Exp})}$$

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
F.S.R._Mantissa	F.S.R. factor according to "Sensor type" variable, which has to be read separately (→ "Read command")
F.S.R._Exp	F.S.R. 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

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 (byte No. 8)

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

Byte No.	Function	Value	Comment
0	Data string length	7	Constant value
1	Page No.	4	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.	2	CDGsci with output signal 10.00 V
2	Status	16	Pressure unit = Torr
3	Error	0	No Error
	Measurement		Calculation of pressure value: Conversion formula → 🗎 5
4	High byte	125	
5	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} ≙ 00 $A9_{hex}$ High order byte is ignored ⇒ Check sum = $A9_{hex}$ ≙ 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	$0 + 2 + 0 = 2_{dec} \triangleq 00\ 02_{hex}$ High bytes are ignored ⇒ Checksum = $02_{hex} \triangleq 2_{dec}$

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)
CPU2Unit	uint8	R		0x00	0x01	0x00	Pressure unit "mbar"
		W		0x10		0x01	Pressure unit "Torr"
						0x02	Pressure unit "Pa"
Filter	uint8	R		0x00	0x02	0x00	Filter dynamic
		W		0x10		0x01	Filter time fast
						0x02	Filter time slow
SP1 Level Low	sint16	R	H-Byte	0x00	0x04	<value>	Lower setpoint threshold SP1 [1)]
		W	H-Byte	0x10			
			L-Byte		0x05	<value>	
SP2 Level Low	sint16	R	H-Byte	0x00	0x06	<value>	Lower setpoint threshold SP2 [1)]
		W		0x10			
			L-Byte		0x07	<value>	
SP1 Level High	sint16	R	H-Byte	0x00	0x08	<value>	Upper setpoint threshold SP1 (hysteresis) [1)]
		W		0x10			
			L-Byte		0x09	<value>	
SP2 Level High	sint16	R	H-Byte	0x00	0x0A	<value>	Upper setpoint threshold SP2 (hysteresis) [1)]
		W		0x10			
			L-Byte		0x0B	<value>	SP1, percentage of ATM (only for 1000Torr)
PerOfAtmSP1	uint8	R		0x00	0xFC	<value>	SP2, percentage of ATM (only for 1000Torr)
		W		0x10			
PerOfAtmSp2	uint8	R		0x00	0xFD	<value>	
		W		0x10			
FirmwareRevisionCPU1	uint8	R		0x00	0x10	<value>	(<value> / 20) = Software Version CPU1 e.g. 20 = V1.0
Calib date	uint32	R	MSB	0x00	0x11	<value>	Date: YY,MM,DD,HH,MM
		R	Byte 2	0x00	0x12	<value>	e.g. 0410291109 =
		R	Byte 1	0x00	0x13	<value>	2004-10-29 at 11:09
		R	LSB	0x00	0x14	<value>	
ZeroAdjustValue	sint16	R	H-Byte	0x00	0x15	<value>	Zero Offset Adjust Level [2)]
		W		0x10			
			L-Byte		0x16	<value>	
DCOutputOffset	sint16	R	H-Byte	0x00	0x17	<value>	Customer DC Output Offset [2)]
		W		0x10			
			L-Byte		0x18	<value>	(Base pressure offset)
SerialNumber	uint8	R	Byte 0	0x00	0x19	<value>	Production number as ASCII string (barcode)
	uint8	R	Byte 1	0x00	0x1A	<value>	(Max. 16 byte)
	uint8	R	…	0x00	…	...	
	uint8	R	Byte 15	0x00	0x28	<value>	(Last digit: null terminator)
SWDateYear	uint16	R	H-Byte	0x00	0xD4	<value>	Software version date Year in Hex e.g. 0x2007 = 2007
			L-Byte		0xD5	<value>	

(continued)

Parameter name	Data type	Access Right	Byte Type	Byte No. 1	Byte No. 2	Byte No. 3	Comment
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	0xD8	<value>	Part number as ASCII string
	uint8	R	Byte 1	0x00	0xD9	<value>	(Max 20 byte)
	uint8	R	…	0x00	…	<value>	e.g. 378-000
	uint8	R	Byte 19	0x00	0xE8	<value>	(Last digit: null terminator)
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	0	F.S.R. = E-3
						1	F.S.R. = E-2
						2	F.S.R. = E-1
						3	F.S.R. = E 0
						4	F.S.R. = E+1
						5	F.S.R. = E+2
						6	F.S.R. = E+3
						7	F.S.R. = E+4
SensFSR (Mantissa)	uint8	R		0x00	0x39	0	Mantissa = 1.0
						1	Mantissa = 1.1
						2	Mantissa = 2.0
						3	Mantissa = 2.5
						4	Mantissa = 5.0
FirmwareRevisionCPU2	uint8	R	Byte 0	0x00	0x10	<value>	Software version Cube application program
	uint8	R	Byte 1	0x00		<value>	
	uint8	R	…	x		<value>	e.g. 1.6.0.14
	uint8	R	Byte 19	x		<value>	(Last digit: null terminator)
ImageRevisionCPU2	uint8	R	Byte 0	0xE0	0x24	<value>	Image version of WinCE operating system
	uint8	R	Byte 1	0xE0		<value>	
	uint8	R	…	0xE0		<value>	e.g. WinCE60
	uint8	R	Byte 19	0xE0		<value>	(Last digit: null terminator)

(continued)

Parameter name	Data type	Access Right	Byte Type	Byte No. 1	Byte No. 2	Byte No. 3	Comment
SystemDateTime	uint8	R W	Byte 0	0xE2 0xF2	0x00	<value>	Set Date and Time.
	uint8	R W	Byte 1	0xE2 0xF2		<value>	Format: dd.mm.yyyy hh:mm:ss e.g. 21/10/2013 10:41:00
	uint8	R W	…	0xE2 0xF2		<value>	
	uint8	R W	Byte 19	0xE2 0xF2		<value>	(Last digit: null terminator)
EthernetLAN	uint8	R W		0xE2 0xF2	0x14	0	Switch Ethernet LAN off
						1	Switch Ethernet LAN on
WirelessLAN	uint8	R W		0xE2 0xF2	0x28	0	Switch Wireless LAN off
						1	Switch Wireless LAN on
WLANSettings	uint8	R	Byte 0	0xE2	0x51	<value>	TCP-IP Address of current wifi connection
	uint8	R	Byte 1	0xE2		<value>	Format: IP-Adress|Network-Mask
	uint8	R	…	0xE2		<value>	Gauge resets after successful IP change
	uint8	R	Byte 19	0xE2		<value>	(Last digit: null terminator)
LANSettings	uint8	R W	Byte 0	0xE2 0xF2	0x65	<value>	TCP-IP Address of current LAN connection
	uint8	R W	Byte 1	0xE2 0xF2		<value>	Format: IP-Adress|Network-Mask
	uint8	R W	…	0xE2 0xF2		<value>	Gauge resets after successful IP change
	uint8	R W	Byte 19	0xE2 0xF2		<value>	(Last digit: null terminator)
CustomAnalogOut	uint8	R W		0xE2 0xF2	0x89	0	Switch Zoom for analog out off
						1	Switch Zoom for analog out on
AnalogoutPLow	real32	R W	Byte 0	0xE2 0xF2	0x81	<value>	Low pressure threshold for zoom function
		R W	Byte 1	0xE2 0xF2		<value>	Will set this pressure value to 0.0V
		R W	Byte 2	0xE2 0xF2		<value>	
		R W	Byte 3	0xE2 0xF2		<value>	
AnalogoutPHigh	real32	R W	Byte 0	0xE2 0xF2	0x85	<value>	IEEE Standard for Floating-Point Arithmetic
		R W	Byte 1	0xE2 0xF2		<value>	Will set this pressure value to 10.0V
		R W	Byte 2	0xE2 0xF2		<value>	
		R W	Byte 3	0xE2 0xF2		<value>	
Pressure	real32	R	Byte 0	0xE2	0x8B	<value>	Pressure value in pressure units
		R	Byte 1			<value>	IEEE Standard for Floating-Point Arithmetic
		R	Byte 2			<value>	
		R	Byte 3			<value>	

(continued)

(concluded)

Parameter name	Data type	Access Right	Byte Type	Byte No. 1	Byte No. 2	Byte No. 3	Comment
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
		R	L-Byte	0xE2		<value>	Binäre 16 bit Darstellung
MACAddress	uint8	R	Byte 0	0xE2	0x9B	<value>	MAC address of Ethernet port
	uint8	R	Byte 1	0xE2		<value>	
	uint8	R	…	0xE2		<value>	
	uint8	R	Byte 19	0xE2		<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

$$\text{Setpoint\_level xy} = \frac{\text{<data\_value>}\times a}{b} \times \text{F.S.R.\_Mantisse}\times 10^{(\text{F.S.R.\_Exp})}$$

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 = F.S.R. – 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).

$$\text{Zero\_Adjust\_Value} = \frac{\text{<data\_value>}\times a}{b} \times \text{F.S.R.\_Mantisse}\times 10^{(\text{F.S.R.\_Exp})}$$

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 (F.S.R. - DC output level).

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

$$\text{DC Output Level} = \frac{<\text{data\_value}>\times a}{b} \times \text{F.S.R.\_Mantisse}\times 10^{(\text{F.S.R.\_Exp})}$$

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

$$\text{Software version} = \frac{<\text{data\_value}>}{20} \quad \text{e.g. } <\text{data\_value}> = 20 \triangleq \text{V1.0}$$

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

Parameter

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td><data_value></td><td>Zero offset measurement data, consisting of "low and high byte" (16 bit value), data value in decimal format.</td></tr>
<tr><td>a</td><td>Conversion factor for pressure units other than "Torr"
Torr: a = 1.00
mbar: a = 1.3332
Pa: a = 133.32</td></tr>
<tr><td>b</td><td>Factor for resolution
<table>
<tr><th>Page No. [1]</th><th>b</th><th>Output signal</th></tr>
<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>
</table></td></tr>
<tr><td>F.S.R._Mantisse</td><td>F.S.R. factor according to the "Sensor type" variable, which has to be read separately (→ "Read command").</td></tr>
<tr><td>F.S.R._Exp</td><td>F.S.R. exponent according to "Sensor type" variable, which has to be read separately (→ "Read command").</td></tr>
</table>

[1] → 🗎 4, tabel "Structure of send string", byte no. 1

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

www.inficon.com

Original: English   tira90e1   (2014-01)
tira90e1