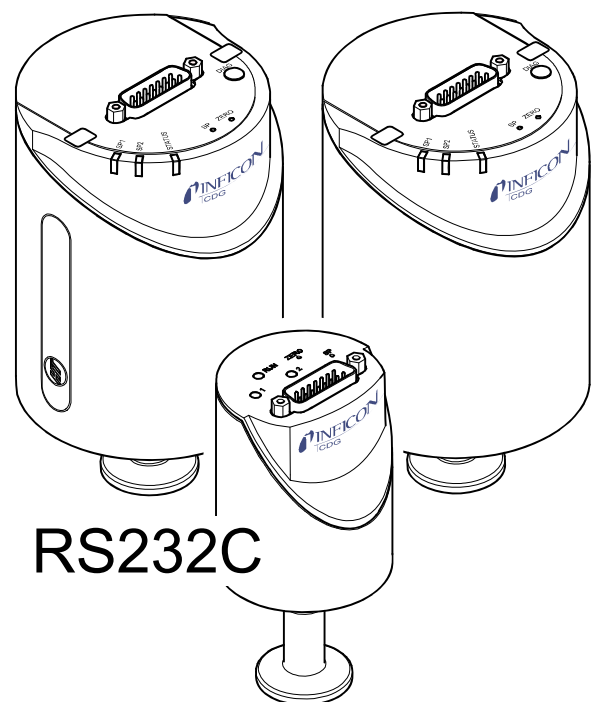


RS232C

Serial Interface

CDG025D, CDG045D,
CDG100D, CDG160D,
CDG200D




General Information


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

- an INFICON Vacuum Gauge Controller (VGC series) or
- another appropriate controller.

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.

Functional Principle

The RS232C Serial Interface is used in duplex operation. The gauge continuously (approximately every 20 ms) transmits a nine byte send string without request. Instructions to the gauge are transmitted via five-byte receipt strings.

Data format

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

Transmission rate

- 9600 Baud

Pin assignment

- TxD Pin 13
- RxD Pin 14
- GND Pin 5
(sensor cable connector)

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 3 2	Constant value for CDG025D ¹⁾ Constant value for CDGxxxD ²⁾ Constant value for CDG025D ³⁾
2	Status		→ "Status byte"
3	Error		→ "Error byte"
4	Measured value high byte	<value>	→ "Calculation of pressure value"
5	Measured value low byte	<value>	→ "Calculation of pressure value"
6	Read command	<value>	Read value
7	Sensor type		→ "Sensor type"
8	Checksum		→ "Synchronization"


¹⁾ CDG025D with output signal 10.00 V (analog)

²⁾ CDG045D, CDG100D, CDG160D, CDG200D with output signal 10.24 V (analog)

³⁾ CDG025D with output signal 10.24 V (analog)

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⁾
1	Sensor temperature attained ¹⁾

¹⁾ for CDG045D, CDG100D, CDG160D, CDG200D only

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 and 5 of the send string (decimal presentation).

Conversion formula
(byte No. 4 and 5)

$$p = \frac{\langle \text{pressure_value} \rangle \times a}{b} \times \text{F.S.R.}_{\text{Mantissa}} \times 10^{(\text{F.S.R.}_{\text{Exp}})}$$

Parameter	Description												
p	Pressure value in selected pressure unit (→ Parameter a)												
$\langle \text{pressure_value} \rangle$	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												
	<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>32000</td> <td>10.24 V</td> </tr> <tr> <td>3</td> <td>32000</td> <td>10.24 V</td> </tr> <tr> <td>4</td> <td>32767</td> <td>10.00 V</td> </tr> </tbody> </table>	Page No. ¹⁾	b	Output signal	2	32000	10.24 V	3	32000	10.24 V	4	32767	10.00 V
Page No. ¹⁾	b	Output signal											
2	32000	10.24 V											
3	32000	10.24 V											
4	32767	10.00 V											

¹⁾ → 3, 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.

Sensor type
(byte No. 7)

Bit No. 0 ... 3	Description	Comment
0x0	F.S.R. = 10^{-3}	→ Variable "Sensor_pressure_range" F.S.R._Exponent = (<value> -3)
0x1	F.S.R. = 10^{-2}	
0x2	F.S.R. = 10^{-1}	
0x3	F.S.R. = $10^0 = 1$	
0x4	F.S.R. = $10^1 = 10$	
0x5	F.S.R. = $10^2 = 100$	
0x6	F.S.R. = $10^3 = 1000$	
0x7	F.S.R. = $10^4 = 10000$	

Bit No. 4 ... 7	Description	Comment
0x0	F.S.R._Mantissa = 1.0	→ Variable "Sensor_FSR"
0x1	F.S.R._Mantissa = 1.1	
0x2	F.S.R._Mantissa = 2.0	
0x3	F.S.R._Mantissa = 2.5	
0x4	F.S.R._Mantissa = 5.0	

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 CDG025D ¹⁾
		3	Constant value for CDGxxxD ²⁾
		2	Constant value for CDG025D ³⁾
8	Checksum of bytes No. 1 ... 7	0 ... 255	Low byte of checksum ⁴⁾

¹⁾ CDG025D with output signal 10.00 V (analog)

²⁾ CDG045D, CDG100D, CDG160D, CDG200D with output signal 10.24 V (analog)

³⁾ CDG025D with output signal 10.24 V (analog)

⁴⁾ 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	6	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	CDG025D with output signal 10.24 V
2	Status	16	Pressure unit = Torr
3	Error	0	No Error
4	Measurement High byte	125	Calculation of pressure value: Conversion formula $\rightarrow \cdot 4$
5	Low byte	0	
6	Read command	20	Software version = $20 / 20 = 1.0$
7	Sensor type	6	F.S.R. = 10^{+3}
8	Check sum	169	$2 + 16 + 0 + 125 + 0 + 20 + 6 = 169_{dec} \hat{=} 00A9_{hex}$ High order byte is ignored \Rightarrow 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		\rightarrow "Service command"
2	Data		\rightarrow "Address byte"
3	Data		\rightarrow "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)

Description	Data	Comment
Read command	0x00	Read command for variable according to address in byte No. 2
Write command	0x10	Write command for variable according to address in byte No. 2
Special services	0x40	Direct command (write command) without data information, e.g. Reset, Zero adjust

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

Das Beispiel basiert auf dem Empfangsstring:

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 ⇒ Checksum = $02_{hex} \hat{=} 2_{dec}$

Variables for bytes No. 2 and 3
(read/write command)

Parameter name	Data type	Description	Byte No. 2	Byte No. 3	Comment
DataTxMode	uint8_T / RW		0	0 ¹⁾	Continued output of measured value
				1	Individual output of measured value (polling)
Unit	uint8_T / RW		1	0	Pressure unit "mbar"
				1 ¹⁾	Pressure unit "Torr"
Filter	uint8_T / RW		2	0 ¹⁾	Filter dynamic
				1	Filter time fast
				2	Filter time slow
SP1 Level Low	sint16_T / RW	H-Byte	4	<value>	Lower setpoint threshold SP1 ²⁾
		L-Byte	5	<value>	
SP2 Level Low	sint16_T / RW	H-Byte	6	<value>	Lower setpoint threshold SP2 ²⁾
		L-Byte	7	<value>	
SP1 Level High	sint16_T / RW	H-Byte	8	<value>	Upper setpoint threshold SP1 (hysteresis) ²⁾
		L-Byte	9	<value>	
SP2 Level High	sint16_T / RW	H-Byte	10	<value>	Upper setpoint threshold SP2 (hysteresis) ²⁾
		L-Byte	11	<value>	
Software version	uint8_T / Ro		16	<value>	(<value> / 20) = Software Version e.g. 20 = V1.0
Calib date	uint32_T / Ro	MSB	17	<value>	Date: YY,MM,DD,HH,MM e.g. 0410291109 = 2004-10-29 at 11:09
		Byte 2	18	<value>	
		Byte 1	19	<value>	
		LSB	20	<value>	
Zero_Adjust_Value	sint16_T / RW	H-Byte	21	<value>	Zero Offset Adjust Level ³⁾
		L-Byte	22	<value>	
DC Output Offset	sint16_T / RW	H-Byte	23	<value>	Customer DC Output Offset ³⁾ (Base pressure offset)
		L-Byte	24	<value>	
Production No.	uint8_T / Ro	Byte 0	25	<value>	Production number as ASCII string (barcode) (Max. 16 byte) (Last digit: null terminator)
	uint8_T / Ro	Byte 1	26	<value>	
	uint8_T / Ro	
	uint8_T / Ro	Byte 15	40	<value>	
Software date Year	uint16_T / Ro	H-Byte	212	<value>	Software version date Year in Hex e.g. 0x2007 = 2007
		L-Byte	213	<value>	
Software date M/D	uint16_T / Ro	H-Byte	214	<value>	Software version date Month in Hex e.g. 0x03 = March
		L-Byte	215	<value>	Software version date Day in Hex e.g. 0x19 = 19
Part No.	uint8_T / Ro	Byte 0	218	<value>	Part number as ASCII string (Max 20 byte) e.g. 378-000 (Last digit: null terminator)
	uint8_T / Ro	Byte 1	219	<value>	
	uint8_T / Ro	<value>	
	uint8_T / Ro	Byte 19	237	<value>	
Remaining_Zero	sint16_T / Ro	H-Byte	72	<value>	Max. remaining offset value
		L-Byte	73	<value>	

(continued)

(completed)

Parameter name	Data type	Description	Byte No. 2	Byte No. 3	Comment
Extended error H-Byte	uint8_T / Ro	H-Byte	54	Bit 0	PT1000 fault ⁴⁾
				Bit 1	Heater block overtemp. ⁴⁾
				Bit 2	Electronic overtemp. ⁴⁾
				Bit 3	Zero adjust error
				Bit 4	Reserve
				Bit 5	Reserve
				Bit 6	Reserve
				Bit 7	Reserve
Extended error L-Byte	uint8_T / Ro	L-Byte	55	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
Pressure range (Exponent)	uint8_T / Ro		56	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
Pressure range (Mantissa)	uint8_T / Ro		57	0	Mantissa = 1.0
				1	Mantissa = 1.1
				2	Mantissa = 2.0
				3	Mantissa = 2.5
				4	Mantissa = 5.0
Gauge config	uint8_T / Ro		58	0	= Analog out 0 ... 10.24 V
				1	= Analog out 1 ... 9 V
CDG type	uint8_T / Ro		59	0	= CDG025D
				1	= CDG045D
				2	= CDG100D
				3	= CDG160D
				4	= CDG200D

RW = Read / Write

Ro = Read only

¹⁾ Factory setting

²⁾ Conversion → Section 1.1, byte 6 "Read command"

³⁾ Conversion → Section 1.1, bytes 4 and 5 "Pressure unit"

⁴⁾ CDG045D, CDG100D, CDG160D, CDG200D only

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

Parameter name	Data type	Description	Byte No. 2	Byte No. 3	Comment
RESET	uint8_T / W		0	0	Power reset: Starts continuous pressure output
RESET Factory	uint8_T / W		1	0	Factory reset: Sets factory configuration
Zero_adjust	uint8_T / W		2	–	Starts zero offset adjustment

W = Write

Description of variables

Setpoint_level xy

$$\text{Setpoint_level xy} = \frac{\langle \text{data_value} \rangle \times a}{b} \times \text{F.S.R.}_{\text{Mantisse}} \times 10^{(\text{F.S.R.}_{\text{Exp}})}$$

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

¹⁾ Further parameter → 11, 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 (→ 4).

$$\text{Zero_Adjust_Value} = \frac{\langle \text{data_value} \rangle \times a}{b} \times \text{F.S.R.}_{\text{Mantisse}} \times 10^{(\text{F.S.R.}_{\text{Exp}})}$$

Parameter ¹⁾	Description
Zero_Adjust_Value	Zero pressure offset in the selected pressure unit (→ 11, table "Parameter").

¹⁾ Further parameter → 11, 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.

$$\text{DC Output Level} = \frac{\langle \text{data_value} \rangle \times a}{b} \times \text{F.S.R.}_{\text{Mantisse}} \times 10^{(\text{F.S.R.}_{\text{Exp}})}$$

Parameter ¹⁾	Description
DC Output Level	DC-Output-Signal in the selected pressure unit (→ 11, table "Parameter").

¹⁾ Further parameter → 11, 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>32000</td> <td>10.24 V</td> </tr> <tr> <td>3</td> <td>32000</td> <td>10.24 V</td> </tr> <tr> <td>4</td> <td>32767</td> <td>10.00 V</td> </tr> </tbody> </table>	Page No. ¹⁾	b	Output signal	2	32000	10.24 V	3	32000	10.24 V	4	32767	10.00 V
Page No. ¹⁾	b	Output signal											
2	32000	10.24 V											
3	32000	10.24 V											
4	32767	10.00 V											
F.S.R._Mantisse	F.S.R. factor according to the "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").												

¹⁾ → 3, tabel "Structure of send string", byte no. 1

Original: German tira49d1-a (2010-02)



tira49e1-a



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

www.inficon.com