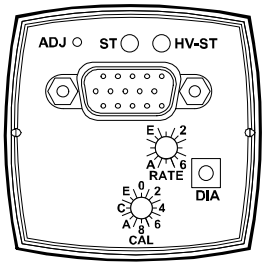
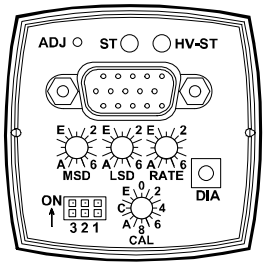
RS232C, RS485C

Serial Interface for Cold Cathode and Cold Cathode Pirani Gauges

MAG500, MAG504
MPG500, MPG504



RS232C



RS485C

For safety information, specifications and operation instructions of the vacuum gauges refer to Operating Manual tina83d1 (German) or tina83e1 (English). Both are included in scope of delivery of the gauge.

General Information

The serial interface allows the communication of the digital INFICON Cold Cathode Gauges (MAG500, MAG504) and the INFICON Cold Cathode Pirani Gauges (MPG500, MPG504) with a PC or another appropriate controller.

The RS232C or RS485C interface integrated in the 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 serial interface and a fieldbus interface 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 or RS485C and the fieldbus or diagnostic port.

Contents

1 Technical Data

For further technical data refer to Operating Manual tina83d1 (German) or tina83e1 (English).

Status signal (digital output)

Supply voltage	≤30 V (dc)
Current rating	100 mA (sink)
High voltage is ON	0 V
High voltage is OFF	open

High voltage cut-in, low active (digital input)

Input voltage	≤30 V (dc)
High voltage ON	<2.5 V (dc
High voltage OFF	>4.0 V (dc)

Data format

- Binary for the INFICON protocol
- ASCII for Edwards nAIM compatible protocol
- ASCII for MKS 972B compatible protocol
- 8 data bits
- 1 stop bit
- no parity bit
- no handshake

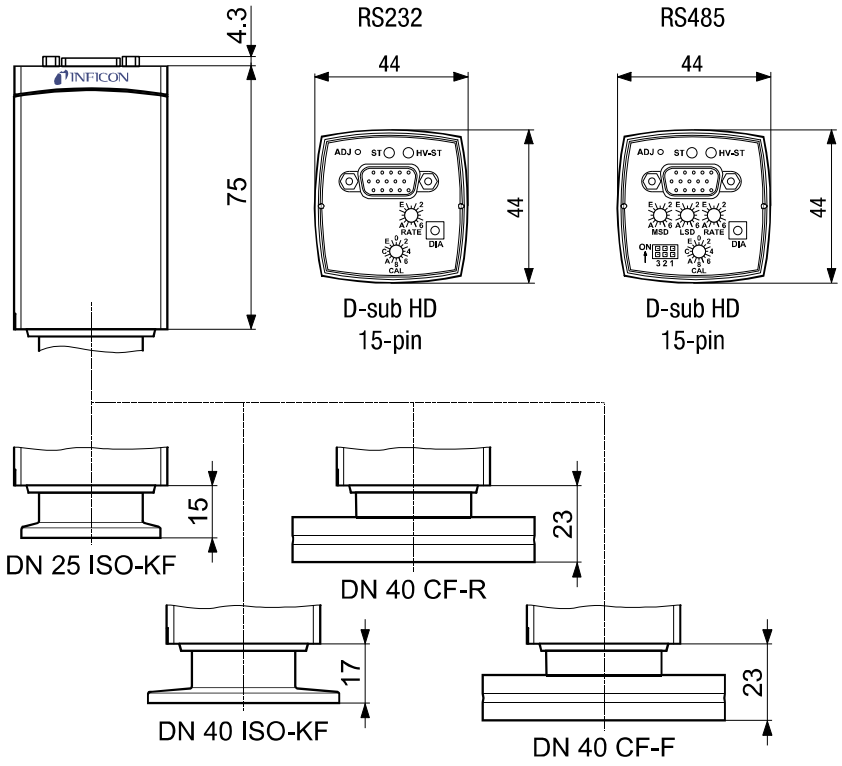
Baud rate & address

INFICON protocol	Baud rate			Address	
Connector – interface protocol	Default	PID	Setting via	Default	Setting via
DIA (diagnostic port) – RS232C (for factory setting only)	57600	180	Protocol	–	–
D-sub HD, 15-pin – RS232C (gauges with PN "3Mxx-xxx-x**45**x")	57600	190 (read only)	Switch <RATE>	–	–
D-sub HD, 15-pin – RS485C (gauges with PN "3Mxx-xxx-x**43**x")	57600	190 (read only)	Switch <RATE>	00_{hex}	Switches <MSD> <LSD>

Edwards nAIM protocol	Baud rate		Address	
Connector – interface protocol	Default	Command	Default	Command
D-sub HD, 15-pin – RS485C (gauges with PN "3Mxx-xxx-x**4TK**")	4 = 9600	!C780	00	!S750

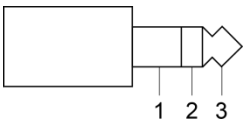
MKS 972B protocol	Baud rate		Address	
Connector – interface protocol	Default	Command	Default	Command
D-sub HD, 15-pin – RS485C (gauges with PN "3Mxx-xxx-x**4IP**")	9600	BR!	253	AD!

Dimensions [mm]



2 Electrical Connection

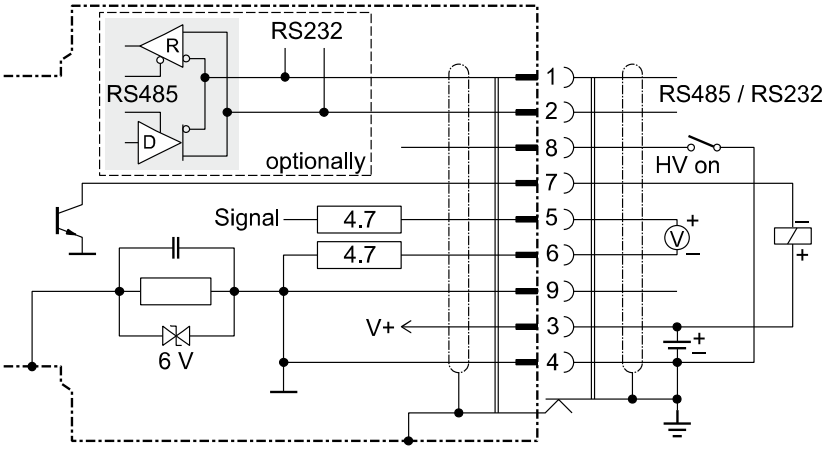
Diagnostic Port (RS232C)



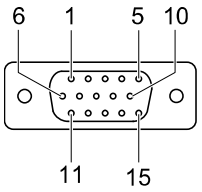
Pin 1	Supply common
Pin 2	RS232C, RxD
Pin 3	RS232C, TxD

D-sub HD, 15-pin (RS232C / RS485C)

If no sensor cable is available, make one according to the following diagram:



Pin 1	RS485C A–	or RS232C TX
Pin 2	RS485C B+	or RS232C RX
Pin 3	Supply 14.5 … 30 V	
Pin 4	Supply Common GND	
Pin 5	Signal output	
Pin 6	Signal common	
Pin 7	Status (Gauge on = low)	
Pin 8 [1)]	High voltage HV on/off (low active)	
Pin 9	reserved (internal use)	
Pin 10 … 15	reserved	



D-sub HD, 15-pin
female
soldering side

1) MAG only. HV is switched on via RS interface (default).

3 Protocol

3.1 INFICON Protocol

The serial interface is used in Master-Slave mode. Without a corresponding request from the Master, the device does not transmit any data.

Instructions to the gauge are transmitted via binary protocol.

RS232C address and transmission rate

It is not necessary to set an address of any RS232C interface. For the protocol 0 is always used. The transmission rate can be set as parameter (→ 🗎 10).

RS485C address <MSD>, <LSD>

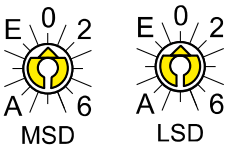
The node address is an unambiguous device address in the RS485C network. The gauge can only communicate with the network if the node address setting is correct. Valid address range: 0 … 255 in decimal form.

The node address setting is made on the gauge in hexadecimal form (00 … FF_{hex}) by means of two rotary switches.

<MSD>: upper half byte setting

<LSD>: lower half byte setting

Factory setting: Node address = 00_{hex}:



When the gauge is put into operation, the firmware queries its node address.

If the node address is modified during operation it is taken over immediately.

RS485C bias + termination

The bias + termination is set with the dip switch.

Position	Function
1 on	Signal "RS-485 B+" 470 Ω pull up to +3.3 V
1 off	Signal "RS-485 B+" no bias
2 on	120 Ω termination
2 off	no termination
3 on	Signal "RS-485 A-" 470 Ω pull down to GND
3 off	Signal "RS-485 A-" no bias

RS232C / RS485C transmission rate <RATE>

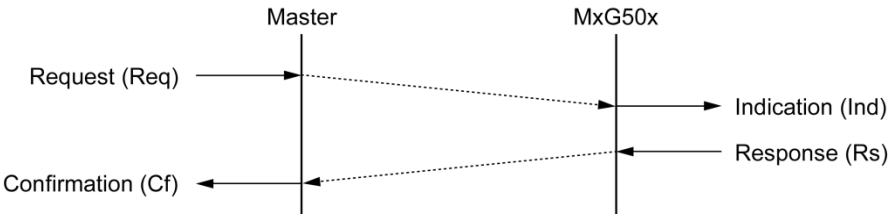
The transmission rate is set with the rotary switch <RATE>. It is only for the serial communication on the HD-sub15 connector. The rate for the DIAG connector is not selectabe with this switch.

Position	Baud
1	9600
2	19200
3	38400
4	57600
0	57600 (default)
5 … F	reserved

Factory setting: Transmission rate = 00_{hex}:

If the transmission rate is modified during operation, a reset occurs and the new transmission rate is taken over.

Data Link Layer



The Data Link Layer uses the following terminology:

request (Req)

A request is an instruction (to read/or write) transmitted by the Master.

indication (Ind)

An indication means that the Slave (gauge) recognizes a request from the Master.

response (Rs)

A response is an answer from the Slave to the Master.

confirmation (Cf)

A confirmation is an acknowledgement of the response by the Master.

Protocol Frame

Every protocol layer of the communication protocol is represented in a protocol frame. The maximum length per frame is 64 byte. The data field of the data link layer consists of

- a command (Cmd)
- a parameter identifier (PID)
- a data field (Data).

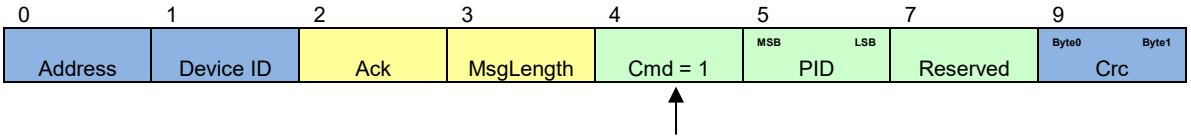
The command determines whether the data transmitted are read or write requests.

Command (Cmd)

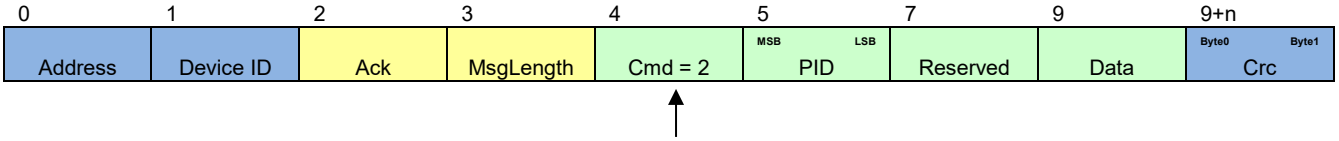
There are four different command types:

Cmd 1 Read request from the Master
Cmd 2 Read response from the gauge to a read request
Cmd 3 Write request
Cmd 4 Write response from the gauge to a write request

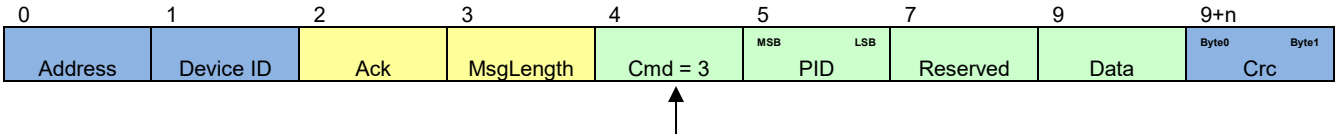
- Cmd 1: Read request from the Master



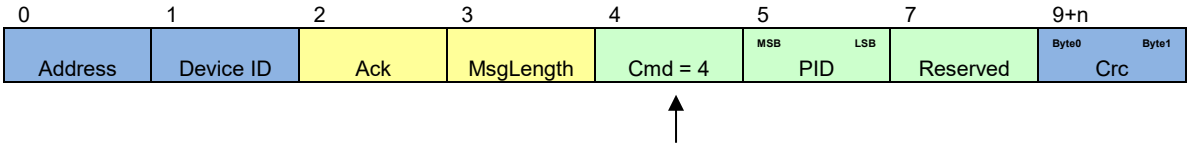
- Cmd 2: Response from the gauge to a read request



- Cmd 3: Write request from a Master



- Cmd 4: Response from the gauge to a write request



Parameter Identifier (PID)

The parameter identifier (PID) addresses a defined parameter of the device.

Data field (Data)

The data field contains the data of a request. In the case of a write request it contains the data transmitted from the Master to the gauge, in the case of a read request it contains the data to be transmitted from the gauge to the Master. Data are transmitted in Big Endian, i.e. the most significant byte is transmitted first (e.g. 0x46 0xD2 corresponds to 18130 decimal).

Medium Access Layer

The medium access layer contains the following data fields:

- RS485C address (for RS232C the address is 0)
- Device ID (Master 0, MPG50x 4, MAG50x 20)
- Header with Ack and Message Length (all APDU data are counted: Cmd, PID, Reserved, Data)
- 16 bit CRC (example → 🗎 12)

Communication Error

If a communication error occurs during transmission the PID is set to 0xFFFF and a data byte with the error information is added.

Byte 0, errors

Error Code	Description
1	Access error
2	Value higher max. or lower min.
3	Parameter not found
4	Length error
6	Memory access error
7	Memory access timeout

3.1.1 Protocol Description

The following parameters can be read via a serial interface.

Output pressure

PID	Name	Description	Factory setting	Min.	Max.	Type
221	Pressure	Output pressure in integer format (mbar)				LogFixs32en26
222	Pressure	Pressure in vacuum chamber in Real format. Pressure unit depends on the PID 224 setting.				Real32
224	Data Unit	Changes the pressure unit of all Real32 pressures. 0 mbar 1 Torr 2 Pascal 3 micron 4 Counts	0	0	4	Uint8

Error

PID	Name	Description	Factory setting	Min.	Max.	Type
228	Device Exception	0 No error 1 Timeout EEPROM memory access 2 EEPROM CRC error 4 EEPROM error 8 Pirani filament rupture 2048 CCIG short circuit	0			Uint32

General information

PID	Name	Description	Factory setting	Min.	Max.	Type
103	Reset	0 Write The sensor carries out a reset. 1 Write The sensor resets all parameters to their factory settings.		0	1	Uint8
104	Run Hours	Operating hours [0.25 h]				Uint32
207	Serial Number	Serial number			4294967295	Uint32
208	Product Name	Product name	MAG500			String
209	Manufacturers Name	Name of manufacturer	INFICON AG			String
210	Manufacturers Model Number	Model number				String
218	Software Version	Software version				String
180	RS232 Baud Rate Diagnostic Port	9600, 19200, 38400, 57600	57600	9600	57600	Uint32
190	RS232 Baud Rate D-sub HD, 15-pin Connector (read only)					

Sensor details

PID	Name	Description	Factory setting	Min.	Max.	Type
223	Active Instance Number	Current sensor 1 CCIG sensor active 2 Pirani sensor active 3 Mixed signal range Pirani and CCIG sensor active				Uint8
33000	Pirani Full Scale*	Pirani fullscale (mbar)	1000	1e-5	2047	LogFixs32en26
33001	Pirani Overrange Value*	Pirani overrange status output limit (mbar)	1000	100	1500	LogFixs32en26
255	Pirani Safe State*	Pressure reading in mbar that is output in the event of an error in Pirani mode 0 0 mbar 1 Pirani Full Scale value (PID 33000) is output 2 The last valid value is retained. 3 The Pirani Safe State value is output.	0	0	3	Uint8
256	Pirani Safe State Value*	Pressure reading in mbar that is output in the event of an error if the Pirani Safe State is set to 3.	1e-11	1e-11	1000	LogFixs32en26
418	Pirani Adjust Flag*	Executes a manual Pirani sensor adjustment. Set the value to "1" in order for the adjustment to be executed.	0	0	1	Uint8
504	CCIG Safe State	Pressure reading in mbar that is output in the event of an error in CCIG mode. 0 0 mbar 1 CCIG value is output 2 The last valid value is retained. 3 The CCIG Safe State value is output.	0	0	3	Uint8
505	CCIG Safe State Value	Pressure reading in mbar that is output in the event of an error if the CCIG Safe State is set to 3.	1e-11	1e-11	1e-1	LogFixs32en26

(continued)

(Table "Sensor details" concluded)

PID	Name	Description	Factory setting	Min.	Max.	Type
503	CCIG Full Scale	CCIG full scale	1E-2	1e-11	1e-1	LogFixs32en26
506	CCIG Overrange Value	If the pressure exceeds this value the measurement is invalid.	1E-2	1e-11	5e-2	LogFixs32en26
507	CCIG Underrange Value	If the pressure is below this value the measurement is invalid.	5E-9	1e-11	1e-1	LogFixs32en26
529	CCIG Switch on / off**	Switch on/off the MAG50x with RS485 [0 = off, 1 = on]	0	0	1	Uint8
533	CCIG Ignition Status	Status of the CCIG: [0 = CCIG off, 1 = CCIG on, but not ignited yet, 3 = CCIG on and ignited]	0	0	3	Uint8

* Parameter not available for MAG50x

** Parameter not available for MPG50x

3.1.2 Data Format

LogFixs32enXX

To calculate the pressure value from a (signed) LogFixs32enXX value, first convert the hexadecimal value, which is read by the interface, into a decimal number. Then divide this decimal number by the factor of 2^{xx}. The result is the pressure value in mbar.

Vice versa, calculating decimal to LogFixs32enXX, logarithmize the pressure value [mbar] with base 10 and then multiply the result by the factor of 2^{xx}. Then convert this result into a hexadecimal value.

Examples for LogFixs32en26 calculations:

- Binary → mbar (the pressure value, which is read by the interface, is e.g. 0xEE 0xCB 0xBE 0xCB):
  Convert the hexadecimal pressure value 0xEE 0xCB 0xBE 0xCB into a (signed) decimal number (corresponds to -288637237 decimal). Then divide this decimal number by the factor of 2^{26} (-288637237 / 2^{26} = -4.30103) and then potentiate the result with the base 10 ($10^{-4.30103}$ = 5e-5). The result is the pressure value in mbar (5e-5 mbar).
- mbar → binary (e.g. set the hysteresis of the High Trip Point setpoint to e.g. 15 mbar):
  Logarithmize the hysteresis 15 mbar with base 10 ($\log_{10}(15)$ = 1.176) and then multiply the result by the factor of 2^{26} (1.176 × 2^{26} = 78926148). Then convert this value into a hexadecimal value (78926148 decimal corresponds to 0x04 0xB4 0x51 0x44 hexadecimal).

Real32

Use, for example, the following link for the conversion of a value in Real32 format into a binary 32 IEEE 754 format:

- https://www.h-schmidt.net/FloatConverter
  (e.g. 0x44 0x6B 0xBA 0x4D corresponds to 942.9 mbar)

3.1.3 Examples

The following examples show how read and write requests are made.

Read request (reading a pressure)

The required parameter has PID 221.

From Master to MxG50x:

0	1	2	3	4	5	7	9	9+n
Address	Device ID	Ack	MsgLength	Cmd	MSB PID LSB	Reserved	Data	Byte0 Crc Byte1
0x00	0x00	0x00	0x05	0x01	0x00DD	0x0000	-	0xAB21

From MxG50x to Master:

0	1	2	3	4	5	7	9	9+n
Address	Device ID	Ack	MsgLength	Cmd	MSB PID LSB	Reserved	Data	Byte0 Crc Byte1
0x00	0x04	0x01	0x09	0x02	0x00DD	0x0000	0x375A05BF	0xD9BB

Write request (setting a unit)

The required parameter has PID 224. To set the unit to Torr, 1 must be written.

From Master to MxG50x:

0	1	2	3	4	5	7	9	9+n
Address	Device ID	Ack	MsgLength	Cmd	MSB PID LSB	Reserved	Data	Byte0 Crc Byte1
0x00	0x00	0x00	0x06	0x03	0x00E0	0x0000	0x01	0x346D

From MxG50x to Master:

0	1	2	3	4	5	7	9	9+n
Address	Device ID	Ack	MsgLength	Cmd	MSB PID LSB	Reserved	Data	Byte0 Crc Byte1
0x00	0x04	0x01	0x05	0x04	0x00E0	0x0000	-	0x94EA

3.1.4 Calculating CRC

The data packages are secured with CRC16-MCRF4XX in the byte order Crc[7…0], Crc[15…8].

CRC polynominal 0x1021

CRC initial value 0xFFFF

For settings and troubleshooting the CRC value can be calculated with the following online calculators, for example:

- http://www.sunshine2k.de/coding/javascript/crc/crc_js.html
- http://crccalc.com.

3.1.5 CRC Example Code

For the code integration of a CRC calculation, an example code of implementation in C# is given below.

```
using System;

class Program
{
    static void Main()
    {
    // the following test array is a valid protocol frame with crc16 at the end
    byte[] arr = new byte[] { 0x00, 0x00, 0x00, 0x05, 0x01, 0x00, 0xDD, 0x00, 0x00, 0xAB, 0x21};
    UInt16 crc;
    Boolean b;
    // Calculate the crc of the test array arr, of course without crc (therefore length minus 2)
    crc = Crc16.Create(arr, (Byte)(arr.Length - 2));
    // Check, if the test array has a correct crc at the end (it is correct, therefore the returnvalue is
true)
    b = Crc16.Check(arr, (Byte)arr.Length);
    }
}

public class Crc16
{
    // initial value for crc16
    public static UInt16 inital = 0xFFFF;

    // function to create a crc16
    public static UInt16 Create(Byte[] buffer, Byte length)
    {
        UInt16 crc16 = new UInt16();
        UInt16 i = 0;

        // Initial Value for CRC calculation
        crc16 = Crc16.inital;

        while (i < length)
        {
            crc16 = (UInt16)((Crc16.crc16Tab[(crc16 ^ buffer[i++]) & (Byte)0xFF]) ^ (crc16 >> 8));
        }
        return crc16;
    }

    // function to check a buffer with a crc16 at the end
    public static Boolean Check(Byte[] buffer, Byte length)
    {
        UInt16 crc16 = Crc16.inital;
        UInt16 i = 0;
        // calculate crc for the buffer without crc
        while (i < length)
        {
            crc16 = (UInt16)((Crc16.crc16Tab[(crc16 ^ buffer[i++]) & (Byte)0xFF]) ^ (crc16 >> 8));
        }
        if (crc16 == 0)
        {
            return true;
        }
        return false;
    }
    // crc array
    public static UInt16[] crc16Tab =
    {
        0x0000, 0x1189, 0x2312, 0x329B, 0x4624, 0x57AD, 0x6536, 0x74BF,
        0x8C48, 0x9DC1, 0xAF5A, 0xBED3, 0xCA6C, 0xDBE5, 0xE97E, 0xF8F7,
        0x1081, 0x0108, 0x3393, 0x221A, 0x56A5, 0x472C, 0x75B7, 0x643E,
        0x9CC9, 0x8D40, 0xBFDB, 0xAE52, 0xDAED, 0xCB64, 0xF9FF, 0xE876,
        0x2102, 0x308B, 0x0210, 0x1399, 0x6726, 0x76AF, 0x4434, 0x55BD,
        0xAD4A, 0xBCC3, 0x8E58, 0x9FD1, 0xEB6E, 0xFAE7, 0xC87C, 0xD9F5,
        0x3183, 0x200A, 0x1291, 0x0318, 0x77A7, 0x662E, 0x54B5, 0x453C,
        0xBDCB, 0xAC42, 0x9ED9, 0x8F50, 0xFBEF, 0xEA66, 0xD8FD, 0xC974,
        0x4204, 0x538D, 0x6116, 0x709F, 0x0420, 0x15A9, 0x2732, 0x36BB,
        0xCE4C, 0xDFC5, 0xED5E, 0xFCD7, 0x8868, 0x99E1, 0xAB7A, 0xBAF3,
        0x5285, 0x430C, 0x7197, 0x601E, 0x14A1, 0x0528, 0x37B3, 0x263A,
        0xDECD, 0xCF44, 0xFDDF, 0xEC56, 0x98E9, 0x8960, 0xBBFB, 0xAA72,
      0x6306, 0x728F, 0x4014, 0x519D, 0x2522, 0x34AB, 0x0630, 0x17B9,
      0xEF4E, 0xFEC7, 0xCC5C, 0xDDD5, 0xA96A, 0xB8E3, 0x8A78, 0x9BF1,
      0x7387, 0x620E, 0x5095, 0x411C, 0x35A3, 0x242A, 0x16B1, 0x0738,
      0xFFCF, 0xEE46, 0xDCDD, 0xCD54, 0xB9EB, 0xA862, 0x9AF9, 0x8B70,
      0x8408, 0x9581, 0xA71A, 0xB693, 0xC22C, 0xD3A5, 0xE13E, 0xF0B7,
      0x0840, 0x19C9, 0x2B52, 0x3ADB, 0x4E64, 0x5FED, 0x6D76, 0x7CFF,
      0x9489, 0x8500, 0xB79B, 0xA612, 0xD2AD, 0xC324, 0xF1BF, 0xE036,
```

```
        0x18C1, 0x0948, 0x3BD3, 0x2A5A, 0x5EE5, 0x4F6C, 0x7DF7, 0x6C7E,
        0xA50A, 0xB483, 0x8618, 0x9791, 0xE32E, 0xF2A7, 0xC03C, 0xD1B5,
        0x2942, 0x38CB, 0x0A50, 0x1BD9, 0x6F66, 0x7EEF, 0x4C74, 0x5DFD,
        0xB58B, 0xA402, 0x9699, 0x8710, 0xF3AF, 0xE226, 0xD0BD, 0xC134,
        0x39C3, 0x284A, 0x1AD1, 0x0B58, 0x7FE7, 0x6E6E, 0x5CF5, 0x4D7C,
        0xC60C, 0xD785, 0xE51E, 0xF497, 0x8028, 0x91A1, 0xA33A, 0xB2B3,
        0x4A44, 0x5BCD, 0x6956, 0x78DF, 0x0C60, 0x1DE9, 0x2F72, 0x3EFB,
        0xD68D, 0xC704, 0xF59F, 0xE416, 0x90A9, 0x8120, 0xB3BB, 0xA232,
        0x5AC5, 0x4B4C, 0x79D7, 0x685E, 0x1CE1, 0x0D68, 0x3FF3, 0x2E7A,
        0xE70E, 0xF687, 0xC41C, 0xD595, 0xA12A, 0xB0A3, 0x8238, 0x93B1,
        0x6B46, 0x7ACF, 0x4854, 0x59DD, 0x2D62, 0x3CEB, 0x0E70, 0x1FF9,
        0xF78F, 0xE606, 0xD49D, 0xC514, 0xB1AB, 0xA022, 0x92B9, 0x8330,
        0x7BC7, 0x6A4E, 0x58D5, 0x495C, 0x3DE3, 0x2C6A, 0x1EF1, 0x0F78
    };
}
```

3.2 Edwards nAIM Protocol

General

The Edwards nAIM interface is used in Master-Slave mode. Without a corresponding request from the Master, the device does not transmit any data.

Instructions to the gauge are transmitted via an ASCII protocol which has an addressing option.

If the node address (command S750) is set between 01 and 98 a request has the following form:

#aa:bbCcccc [d]

aa	address of the Slave
bb	address of the Master
C	kind of command: "?" for a read request, "!" for a write request
cccc	command id
d	optional arguments / parameters (especially used for write requests)

Each request has to be terminated with a carriage return.

Example: #05:01?V752

Is a read request for the gauge pressure and the status information, send to gauge 05 from Master 01.

The response will be: #01:05=V752 2.94E-04;8022

Which means: send to Master 01 from gauge 05, the command is repeated and the pressure and status bits are added.

Addresses 00 and 99 have a special function:

00	all gauges execute the request but don't send a response back (usefully e.g. to set at all gauges the pressure unit to hPa). This avoids bus collisions.
99	all gauges execute the request and send back a response (usefully if the address of the gauge is unknown but if more than one gauge is connected to the bus this mode leads to bus collisions).

If the node address (command S750) is set to 00, the non-addressed mode is used. Only the command id is needed, e.g. ?V752 for a valid read request.

The response on a valid read command starts with an equal sign, repeats the request command and adds the answer.

The response on a valid write command starts with a multiplication sign, repeats the request command and adds zero to acknowledge.

The response on a wrong command starts with a multiplication sign, repeats the request command (if possible) and adds an error number (please see table with the error value).

Command list

Command	Command Id	Arguments	Read / Write	Description
Gauge type	?S0 or ?S751		Read	Delivers: gauge identity; software-version; gauge (numerical) name Example: =S0 MAG500_RS485;V012100;0001
Gauge address	?S750		Read	Delivers the gauge address (two digits) Example: =S750 00
	!S750	nn	Write (lockable)	Set the gauge address with two digits 00 = non addressed mode 01…98 = addressed mode Example: !S750 02 sets the gauge into addressed mode, the gauge has now the address 2
Gauge name	!S751	nnnn	Write (lockable)	Set a numerical name for the gauge (four digits), range: 0000…9999, value is used by the gauge type command Example: !S751 9876
Pressure and status	?V752		Read	Delivers the measured pressure (in data unit) and the status bits (see table “Gauge status bits” for a detailed description) Example: =V752 5.66E-04;0022
Strike control	!C752	n	Write	Switch on and off the cold cathode (0 = off, 1= on) Example: !C752 1
Command lock	!S753	n	Write	After setting it to the value 1 all commands marked with "Write (lockable)" can not be executed anymore. Value 0 deactivates this function again. Example: !C753 1
Data unit	!S755	n	Write (lockable)	Set the pressure unit 1 = mbar / hPa 2 = Pascal (factory setting) 3 = Torr Example: !S755 2
Factory reset	!S757	1	Write (lockable)	Reset the data unit back to the factory settings
Internal temperature	?V759		Read	Delivers the internal gauge temperature in degree Example: =V759 26.8
Run hours	?V769		Read	Delivers the gauge run hours; cold cathode operation hours; cold cathode exposure value[1)] Example: =V769 0000064;0000062;8.32E-02
	!C769	1234	Write (lockable)	Reset all counters (run hours, cold cathode operation hours and exposure value).
Cold cathode exposure	?S769		Read	Delivers the cold cathode exposure threshold in data unit per hours Example: =S769 5.0E+05
	!S769	n.nE±nn	Write (lockable)	Set the cold cathode exposure threshold (value range: 1E-7…5E+5, in data unit per hours) Example: !S769 1.0E-04
Baud rate	!C780	n	Write (lockable)	Set the baud rate 1 = 38400 2 = 19200 4 = 9200 (factory setting) Example: !C780 4
Serial number	?S790		Read	Delivers the serial number of the gauge Example =S790 123456789

[1)] Cold cathode exposure value is the sum up of the mean pressure value (including data unit). The value and the status bit 15 is calculated and refreshed hourly.

The following Edwards commands are not supported by INFICON and deliver a command error (e.g. *S752 2) back:

!S752	Acknowledge gauge errors
!S754 0;	High setpoint threshold
?S754 0	
!S754 1;	Low setpoint threshold
?S741 1	
!S756	Gas type
!C781	Gauge address auto enumerator

Gauge status bits

Bit	Flag	Description
0	Gauge error	Set if any failure situation occurred in the gauge
1	Cold cathode on	Set if high voltage supply of the cold cathode sensor is switched on
2	Not used	
3	Command lock	Set if command lock is active
4	Pressure unit	1 = mbar / hPa, 0 = Pascal, 1 = Torr
5		0 = mbar / hPa, 1 = Pascal, 1 = Torr
6	EEPROM error	Set if any EEPROM error occurred in the gauge
7	Not used	
8	Cold cathode striking	Set if cold cathode is on but didn't have struck yet
9…14	Not used	
15	Cold cathode exposure	Set if cold cathode exposure threshold is exceeded

If an error occurred during executing of a request the following error values are returned:

Error value	Description
0	No error (used as acknowledge at a write request)
1	No access rights for this command (e.g. try to write to a read only command)
2	The requested command is unknown or not yet supported by INFICON
3	Too less parameters are handed over at a write request
4	The value of the argument is out of range.
5	Write access to a command but the command is locked
6	Not used
7	An EEPROM error occurred during storing.
8	Not used
9	Reserved for future use

Examples for error response

Wrong request (read access to the pressure unit which is a write only command): ?S755

Error response of the gauge: *S755 1

3.3 MKS 972B Protocol

General

The MKS 972B interface is used in Master-Slave mode. Without a corresponding request from the Master, the device does not transmit any data.

Instructions to the gauge are transmitted via an ASCII protocol where each gauge has an address.

The regular address range goes from 001 to 253. Addresses 254 and 255 have a special function:

- At address 254 all gauges in the network execute the request and send back an answer. It is useful if the address of the gauge is unknown. If more than one gauge is connected to the bus this mode leads to bus collisions.
- At address 255 all gauges in the network execute the request but don't send back an answer. It is useful e.g. to switch on all gauges on the bus.

All requests start with the at-sign (@) and are terminated with the sequence: ";FF".

Read request

A read request has the form:

@<ADR><CMD>?;FF

ADR	three address digits
CMD	two or three command digits

Example: @001PR5?;FF requests the pressure value from the gauge with address 001

The gauge will answer with e.g. @001ACK2.56E-8;FF

Write request

Write requests have the form:

@<ADR><CMD>!<ARG>;FF

ADR	three digits address
CMD	two or three digits command
ARG	arguments (the count of digits depends on the kind of command)

Example: @001BR!115200;FF requests to change the baud rate of the gauge with address 001 to 115200 bauds

The gauge will answer with: @001ACK115200;FF

The ACK (for acknowledge) signalizes the request was successful executed by the gauge. The argument of the request is repeated also.

In case of an error a NAK (for no acknowledge) and an error code is sent back by the gauge (please see table with the error value below)

Command list:

Command	Command Id	Arguments	Read / Write	Description
Baud rate	BR		Read	Delivers: the currently used baud rate Example: @001BR?;FF -> @001ACK57600;FF
		9600 or 19200 or 38400 or 57600 or 115200	Write	Set a new baud rate Example: @001BR!115200;FF
Gauge address	AD		Read	Delivers: the currently used address of the gauge, often used if the address is unknown. For this situation the request: @254AD?;FF can be useful. Example: @001ACK001;FF
		001…253	Write	Set a new address Example @001AD!002;FF
Pressure	PR1 or PR5		Read	Delivers: the combined pressure value in a three digits format Example: @001PR1?;FF -> @001ACK2.56E-8;FF
Cold cathode switch on behavior	ENC		Read	Delivers: the switch on behavior of the cold cathode: ON = Pirani sensor controls the cold cathode OFF = cold cathode is controlled by the serial interface (command FP) Example: @001ENC?;FF -> @001ACKOFF;FF
		ON / OFF	Write	Set the switch on behavior of the cold cathode (see above) Example: @001ENC!ON;FF
Manual cold cathode control [1)]	FP		Read	Delivers: the current state of the cold cathode high voltage source: ON = high voltage source is switch on OFF = high voltage source is switch off Example: @001FP?;FF -> @001ACKOFF;FF
		ON / OFF	Write	Switch the high voltage supply of the cold cathode on/off Example: @001FP!OFF;FF

1) This command works only if the cold cathode switch on behavior (ENC) is set to OFF.

All other MKS 972B commands are not supported by INFICON yet. The request answer will be: @<ADR>NAK160;FF.

If an error occurred during executing of a request (the answer contains a NAK) the following error values are returned:

Error value	Description
160	The request command contains error(s) or isn't supported by INFICON yet
169	Argument of a command is missing or wrong
172	Value of the argument is out of range
175	Wrong operation for the requested command (e.g. write to a read-only command)
195	Manual cold cathode control isn't allowed because the switch on behavior (ENC) is set to Pirani control

Notes

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

www.inficon.com

Original: English
tira83e1-e