Dräger medical
A Dräger and Siemens Company

Dräger RS 232 MEDIBUS

Protocol Definition

Instructions for Use

Revision Level 4.03

Contents

For Your Safety and that of Your Patients

For correct and effective use of the apparatus and to avoid hazards it is essential to read the following recommendations and to act accordingly[1)]:

Strictly follow the Instructions for Use

Any use of the apparatus requires full understanding and strict observation of these instructions. The apparatus is only to be used for purposes specified here.

Liability for proper function or damage

The liability for the proper function of the software protocol is irrevocably transferred to the owner or operator if the software protocol is used in a manner not conforming to its intended use.

Dräger cannot be held responsible for damage caused by non-compliance with the recommendations given above. The warranty and liability provisions of the terms of sale and delivery of Dräger are likewise not modified by the recommendations given above.

Dräger Medical AG & Co. KGaA

1)
Insofar as reference is made to laws, regulations or standards, these are based on the legal system of the Federal Republic of Germany.

Intended Use

MEDIBUS is a software protocol intended to be used by two medical devices for exchanging data and control functions via their RS 232 interfaces.

Any data transmitted via the MEDIBUS interface are intended only for information purposes and should not be used to derive therapeutical decisions.

MEDIBUS consists of two independent software protocols one for the transmission of "slow" and one for the transmission of "fast" data.

"Slow" data:
Generated or updated in intervals of the magnitude of seconds. This part is called MEDIBUS.

"Fast" data:
Intended for the transmission of e. g. realtime curves. This part of MEDIBUS is called the "Realtime-Extension".

This manual contains a general description of the protocol including formats of commands and responses.
For device dependent descriptions of supported commands and data sets, port hardware and configurations for Dräger devices please refer to
MEDIBUS for Dräger Anaesthesia Devices (90 28 257)
MEDIBUS for Dräger Pediatric Devices (90 28 127)
MEDIBUS for Dräger Intensive Care Devices (90 28 329)

MEDIBUS
Introduction

The MEDIBUS protocol distinguishes two basic types of messages:

– commands

– responses.

A command is transmitted by one device to request data from the other device or to control its function.

A response is transmitted by one device upon receipt of a command from the other device. Responses may contain embedded commands.

Initializing Communication

(refer to MEDIBUS-life-cyle-diagram)

– To initialize communication or to restart communication after a time-out, a device must send the "Initialize Communications Command" **ICC**. Refer to section "Control commands" for the format of commands.

– A device considers communication initialized after having received either a response to a transmitted ICC or an ICC from the other device. Refer to section "Responding to commands" for the format of responses.

– Commands embedded in a response to an ICC are disregarded.

Terminating Communication

– To stop the communication the "STOP" command has to be sent.

– The command echo has to be checked to make sure the "Stop Communication" command has been received correctly. Further commands from the linked device may be ignored until the communication has been reinitialized.

Time-Out

– Any pause in the data flow exceeding 3 seconds leads to a time-out, terminating the communication link. To resume communication after a time-out, the device must re-initialize communication by transmitting an ICC command (cf. section "Initializing communication"). Whenever a device receives an ICC command, it must send a response to it (cf. section "Responses").

– After receiving a command the device has to send a complete response within 10 seconds.

– If there is no need for sending commands or responses the "NOP"–command has to be sent in 2-second-intervals to keep the communication alive.

Allowable Characters

– Printable ASCII characters.

– Control characters defined in this Instructions for Use.

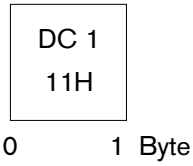
Software Handshaking

– Some control characters can be sent at any time to control the flow of data. They do not require responses.

Suspending data transmission

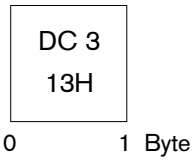
– If a receiving device wants the transmitting device to suspend transmission, the ASCII **"DC1"** character (11H) must be sent.

– Upon receipt of this character, the sending device will suspend any transmission immediately until it receives the ASCII "DC3" character.



Resuming data transmission

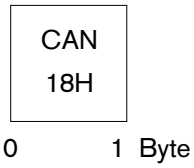
– To request the transmitting device to resume data transmission, the ASCII **"DC3"** character (13H) must be sent within 3 seconds.
Else communication will be reinizialized with an ICC-command.



Aborting data transmission

– To request the other device to abort sending a response or a command, the ASCII **"CAN"** character (18H) must be sent.

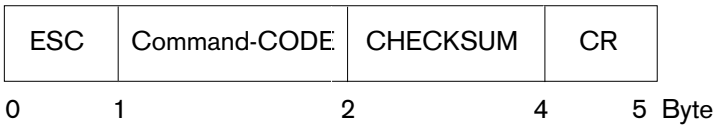
– Upon receipt of this character, any transmission in progress will be immediately aborted.
Communication may be restarted by repetition of the last sent command immediately.

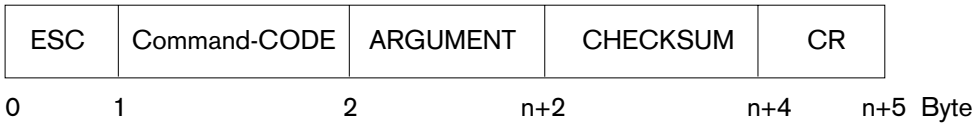


Structure of Commands

- A command is a string of ASCII characters transmitted by one device to request data from the other device or to control its function.
- **A command may be embedded in the response to another command, but a new command must not be transmitted until the response to the previous command has been received.**
- If, however, the response to a command has not been received in full within 10 seconds since the transmission of the last command byte, the command may be repeated or a new command may be transmitted.

Commands have one of the following formats:





ESC	ASCII "escape" character (1BH)
Command-Code	Single byte code specifying the command.
ARGUMENT	The argument string is of variable length n, but n must not exceed 251(0FBH) bytes. The string consists of printable ASCII characters, either text or ASCII HEX numbers. The format of the different arguments are specified in the following sections.
CHECKSUM	Least significant 8-bit sum of all preceding bytes beginning with "ESC" in ASCII HEX format (see section "ASCII Hex format").
CR	ASCII "carriage return" character (0DH)

Note:
To avoid communication breakdown, commands with arguments must not be sent to devices with a MEDIBUS-version less than 3.00.

Command Codes

Control Commands

Control commands are used to initialize, control and stop communication.

Command	Code
No Operation (NOP)	30H
Initialize Communication (ICC)	51H
Stop Communication (STOP)	55H

Data Request Commands

Data request commands are used to request data.

Command	Code
Request current measured Data (Codepage 1)	24H
Request current low Alarm Limits (Codepage 1)	25H
Request current high Alarm Limits (Codepage 1)	26H
Request current Alarms (Codepage 1)	27H
Request current Date and Time	28H
Request current Device Setting	29H
Request current Text Messages	2AH
Request current measured Data (Codepage 2)	2BH
Request current low Alarm Limits (Codepage 2)	2CH
Request current high Alarm Limits (Codepage 2)	2DH
Request current Alarms (Codepage 2)	2EH
Request Device Identification	52H

Miscellaneous Commands

Command	Code
Time changed	49H
Configure Data Response Command	4AH

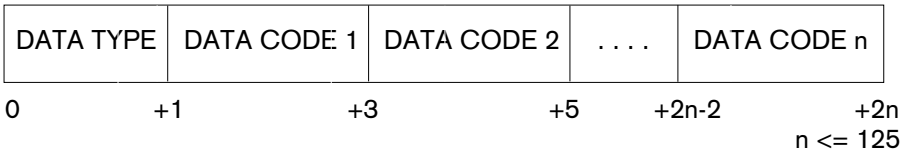
Time Changed Command

The "Time changed" command is sent if during runtime the time or date of the device has been changed. The receiving device can now ask for current date and time. Because of the "Time changed" command there is no need for sending periodically the "Request Current Date and Time" command.

Configure Data Response Command

The "Configure Data Response" command is used to limit the number of data responded by a data source on a data request command. On receipt of any of these data request commands the receiver has to send all actual valid data. In cases where only a few of the possible data are used, the requesting device may configure the responding device to send only these used data by sending a "Configure Data Response" command.

The codes of useful data are given in the argument as follows:



DATA TYPE:
One byte identifying the data type to configure.

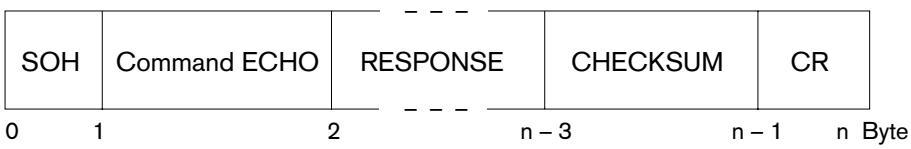
This may be:
24H for current Data, low Alarm Limits and high Alarm Limits (codepage 1)
27H for current Alarms (codepage 1)
29H for current Device Settings
2AH for current Textmessages
2BH for current Data, low Alarm Limits and high Alarm Limits (codepage 2)
2EH for current Alarms (codepage 2)

DATA CODE:
Two byte ASCII HEX data code. See appendices for code numbers.

The configuration stays valid until receipt of a new "Configure Data Response" command. After re-initialization of communication (ICC) and if the "Configure Data Response" is send without data codes, the configuration is set to its default state, where internal programmed configuration is used.
An example is given in Appendix.

Structure of Responses

Upon receipt of a command, a device must respond to it within 10 seconds. A command may be embedded within the response. The following format has to be used:



The response is of variable length, but must not exceed 3845 bytes.

SOH	ASCII "Start of Header" character (01H)
Command ECHO	Echo of the command code being responded to.
RESPONSE	Data as requested by the command, see sections "MEDIBUS Specification"
CHECKSUM	Least significant 8-bit sum of all preceeding bytes beginning with "SOH" in ASCII HEX format (see section "ASCII Hex format").
CR	ASCII "carriage return" character (0DH)

Responding to Control Commands and unknown Commands

A response to a control command or unknown command acknowledges receipt of the command, but contains no data:

SOH	ASCII "Start of Header" character (01H).
Command ECHO	Echo of control command being responded to.
CHECKSUM	Least significant 8-bit sum of all preceeding bytes beginning with "SOH" in ASCII HEX format (see section "ASCII Hex format").
CR	ASCII "carriage return" character (0DH).

– Refer to section "Control commands" for currently defined control commands.

Responding to corrupt Commands

If the received command is **corrupt** (bad checksum), the command echo field must consist of an ASCII "**NAK**" character (15H) and there must be no response field:

SOH	NAK	CHECKSUM		CR
01H	15H	31H	36H	0DH
0	1	2	3	4 5 Byte

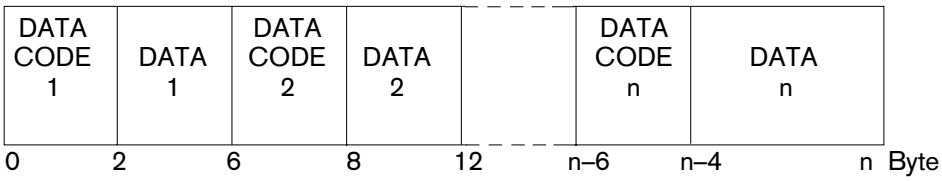
Responding to Data Request Commands

Responses to data request commands contain the current values of a device.

In case a value isn't available at a certain time (for example caused by temporary measurement problems) nothing must be sent for this value. Vice versa this means, that a value is invalid if it is not included in a data request response.

Current Measured Data and Alarm Limit Response

This response must be sent in reply to the "Request current measured Data (codepage 1)" command (24H), "Request current low Alarm Limits (codepage 1)" command (25H), "Request current high Alarm Limits (codepage 1)" command (26H), "Request current measured Data (codepage 2)" command (2BH), "Request current low Alarm Limits (codepage 2)" command (2CH) or "Request current high Alarm Limits (codepage 2)" command (2DH). It contains the current values of all measured parameters or alarm limits available on the responding device. The response field has the following format:

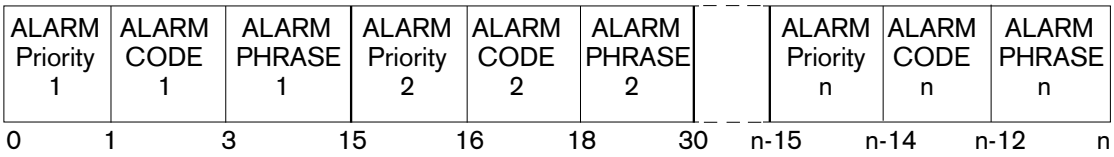


DATA CODE — Two byte ASCII HEX number identifying the parameter or alarm limit.

DATA — Four byte ASCII field containing the current value of the parameter or alarm limit.
See appendix for data formats. Surplus character positions and leading zeros must be replaced by an ASCII "Space" (20H).

Alarm Status Response

This response must be sent in reply to the "Request current Alarms (codepage 1)" command (27H) or "Request current Alarms (codepage 2)" command (2EH). It contains the alarm priority, alarm code, and alarm message for all currently active alarms on the responding device.
The response field has the following format:



ALARM Priority — One byte field specifying the alarm priority (number in the range of 1 to 31), 31 being the highest priority. The priority is encoded by adding 30H. The priorities, therefore, lie in the range from ASCII "1" (31H) to ASCII character "O" (4FH).
Alarm priorities see Appendix.

ALARM CODE — Two byte ASCII HEX number identifying the alarm.

ALARM PHRASE — Twelve byte ASCII character string describing the alarm.

Time & Date Update Response

This response is sent in reply to the "Request Current Date & Time" command (28H). It contains the current date and time from the responding device. The response field has the following format:

TIME (HH:MM:SS)	DATE (DD-MMM-YY)

0 +8 +17

TIME: Eight byte field containing ASCII numeric characters representing the current time in hours (HH), minutes (MM), and seconds (SS). Leading zeroes shall not be suppressed.*

DATE: Nine byte field containing ASCII alpha-numeric characters representing the current day (DD), month (MMM), and year (YY). The first three letters for each month are sent in ASCII. Leading zeroes shall not be suppressed.*

NOTE:
The month representing letters (MMM) has to be sent in German language.

German month representation in time & Date update responses

In a Time & Date update response the current month is represented by three ASCII letters.
These letters are:

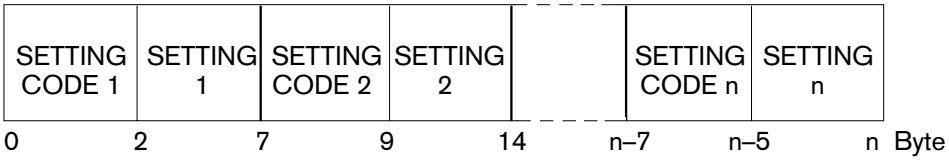
MONTH	REPRESENTIVE
January	JAN
February	FEB
March	MAR
April	APR
May	MAI
June	JUN
July	JUL
August	AUG
September	SEP
October	OKT
November	NOV
December	DEZ

* The PM 8040 substitudes leading zeros in the hours, day and year field with ASCII spaces (20H), e.g. ' 8:06:05 4-MAR 2'.

Device Setting Responses

This response must be sent in reply to the "Request Current Device Settings" command (29H). It contains the current values of all device settings applicable with the responding device.

The response field has the following format:



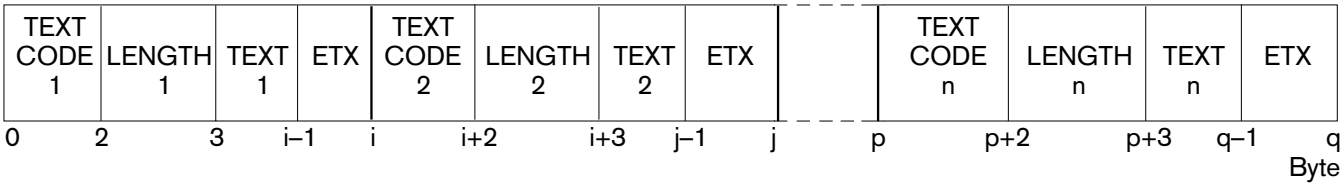
SETTING CODE — Two Byte ASCII HEX number identifying the parameter.

SETTING — Five byte ASCII field containing the current value of the specified parameter. Refer to appendix for the specific formats of the parameters. Surplus character positions and leading zeros must be filled up with ASCII "SPACE" (20H).

Text Message Response

This response must be sent in reply to the "Request Text Messages" command (2AH). It contains all the text messages the requested device currently holds for user information, along with the text code, text length and an end-of-text marker.

The response field has the following format:



TEXT CODE — Two byte ASCII HEX number identifying the text messages.

LENGTH — One byte field specifying the text length, a number in the range from 1 to 32. A text must not be longer than 32 characters. The length is encoded to ASCII format by adding 30H to the decimal length value. Thus, the text length ranges from ASCII "1" (31H) to ASCII "P" (50H).

TEXT — ASCII character string. Refer to appendix for text messages.

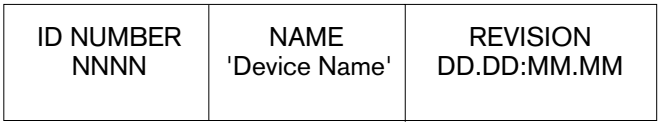
ETX — End-of-text marker (ASCII-Code 03H).

Note:
The length of the response field is limited to 3840 bytes. Due to different lengths of text messages, the maximum number of text messages in a response field depends on the length of the individual text messages.

Device Identification Response

This response must be sent in reply to the "Request Device Identification" command (52H). It contains the identification number, name and release number of the responding device and the MEDIBUS release number.

The response field has the following format:



$0 < n \leq 32$

ID NUMBER	Four byte field containing the ASCII device identification number NNNN.
NAME	ASCII character string delimited by apostrophes (ASCII Code 27H). Therefore, the device name itself must not contain apostrophes. The length of the device name may range from 1 to 32 characters.
REVISION:	Eleven byte field containing ASCII characters representing the device revision level (DD.DD) and the MEDIBUS revision level (MM.MM).

All identification numbers will be defined by Dräger, Lübeck.

MEDIBUS Realtime-Extension Introduction

The Realtime-Extension of MEDIBUS is designed to run independently and without any interference with the slow MEDIBUS concurrently on the same port.

Since realtime data are usually created and transmitted at fixed time intervals, it may happen that slow and fast records are mixed. This means that transmission of a slow command or response may be interrupted by a record of fast data at any time.

Time-Out

Realtime bytes (Sync-Byte, Sync-commands, Realtime values) do not affect the 3 sec time-out described in the "Time-out" chapter of the "slow" MEDIBUS-Protocol.

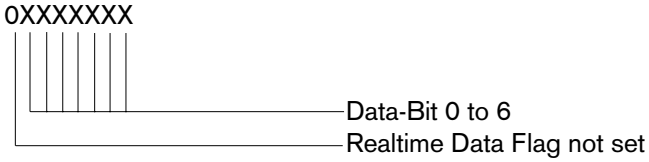
Allowable Characters for Realtime Transmission

Slow communication uses ASCII codes up to 7FH, meaning the most significant bit (bit 7) is always 0. Realtime ("fast") data is distinguished from slow data in that the most significant bit (realtime data flag) is set. All received data < 80H must be ignored within Realtime-transmission.

There are four types of data-bytes which can be distinguished unequivocal:

Command Response Bytes

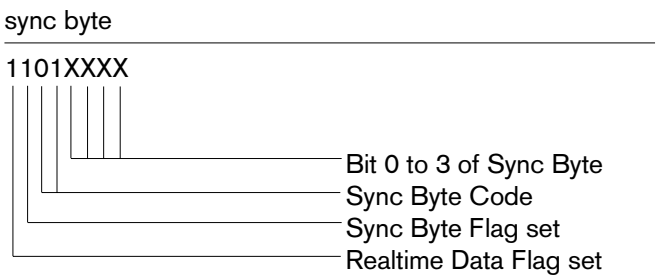
Slow communication's Control-Commands and the responses they require are transmitted in the following form:



Sync Byte

A Sync Byte is always the first byte in a realtime-data-record (see page 21).

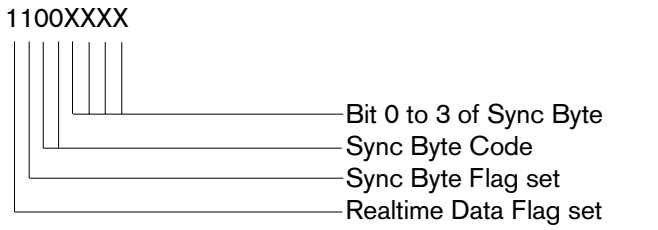
A Sync Byte is transmitted in the following format:



Sync-Command Bytes

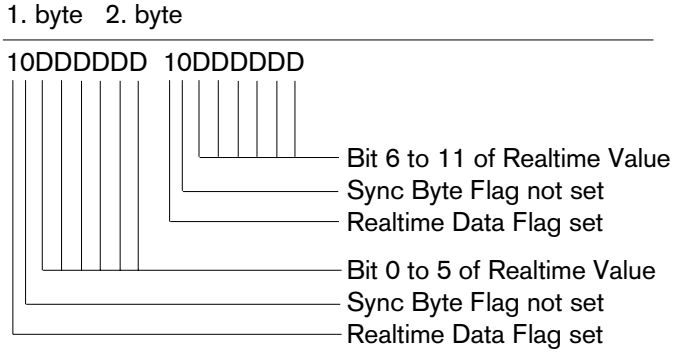
Sync-Commands are used for information which have to be transmitted without delay (see section Sync-Commands).

The form of a sync-command-byte and its argument is as follows:



Realtime-Value-Bytes

A realtime value is transmitted at a resolution of 12 bits. For transmission the value is diveded into two data bytes:



MEDIBUS Realtime-Extension Commands

In addition to the afore matined commands of MEDIBUS (see page 8) the Realtime-Extension uses the following additional commands.

Command Codes

Command	Code
Request Realtime Configuration	53H
Configure Realtime Transmission	54H
Realtime Configuration changed	56H

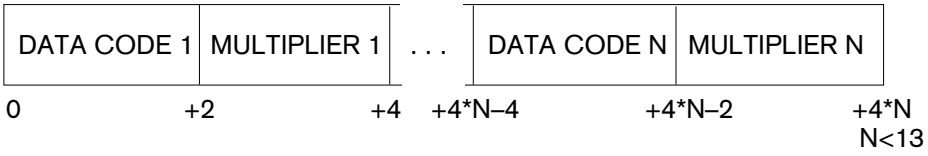
Request Realtime Configuration Command

This command is sent in order to request the supported realtime data of the other device.

Configure Realtime Transmission Command

Ahead of this command the "Request Realtime Configuration" Command must be sent to request the supported realtime data.

The "Configure Realtime-Transmission" Command is sent to request realtime-data in the order and interval as given in the command's argument. The argument has the following format:



DATA CODE: Two Byte ASCII HEX number specifying the requested realtime-data (refer to Appendix 1 for data-codes).

MULTIPLIER: Two Byte ASCII HEX number defining the multiples of the sample interval of the respective parameter as reported within the response of the "Request Realtime-Configuration"-command before. (A value of 1 causes the device to send each sampled data. If the value is set to 2, every second sample is transmitted, and so on.)

Note:
Up to 12 realtime-data may be configured by MEDIBUS. But due to technical restrictions some devices allow less realtime-data to be configured.

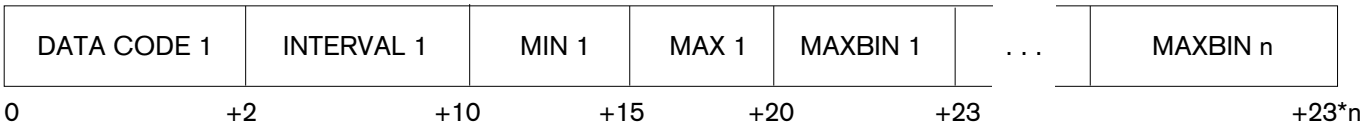
Realtime Configuration Changed Command

This command is sent, if during runtime the configuration has been changed. E.g., the selection of an anaesthesia gas has been changed. The receiving device can now request the current realtime configuration.

MEDIBUS Realtime-Extension Responses

Realtime Configuration Response

This response is sent on reply to the receipt of a Request Realtime-Configuration Command.
The response contains the codes of realtime data, that can be sent by the requested device, the related sample interval, minimal and maximal values:



DATA CODE: Two Byte ASCII HEX number which uniquely identifies the realtime data (see Appendix 1 for data codes).

INTERVAL: 8 Byte ASCII value specifying the sample interval of the realtime data in µs.

MIN: 5 Byte ASCII minimal decimal value of the realtime data.
NOTE:
MIN must be transmitted as a binary 0. Thus all binary data are signless (000_{hex} = 0_{dec} to FFF_{hex} = 4095_{dec}).

MAX: 5 Byte ASCII maximal decimal value of the realtime data.

MAXBIN: 3 Byte ASCII HEX representation of the binary value which is equal to the maximal realtime data given in MAX.

NOTE:
In MIN and MAX the first character may be a minus. MIN and MAX may contain a decimal point at any position except at the first or last character position. Leading zeros must be replaced by spaces.

Due to the definitions above the receiver can calculate the value $v(x_{bin})$ of a received binary realtime data x_{bin} by the following equation:

$$v(x_{bin}) = MIN + x_{bin} * (MAX - MIN)/MAXBIN$$

If the data source has an internal binary representation of the realtime data where the binary value x_{MIN} of MIN is not 0 then it can calculate the binary value to transmit x_{bin} from any internal value $x_{internal}$ as follows:

$$x_{bin} = x_{internal} - x_{MIN}$$

MEDIBUS Realtime-Extension
Realtime-Data Records

Structure of Realtime-Data Records

A realtime data record has the following format:

Sync Byte	1st Sync-Comm. Code 1	1st Sync-Comm. Argum. 1	. . .	1st Sync-Comm. Code n	1st Sync-Comm. Argum. n	1st value lower Bits	1st value higher Bits	2nd value lower Bits	2nd value higher Bits		nth Value lower Bits	nth Value higher Bits

One Sync byte is transmitted first followed by 0 up to 16 Sync-Commands and up to 12 realtime data.

Sync Byte

The sync byte status bits, if set, hold the following information:

Bit no.	Message
0	Value of 1st Realtime Curve is following
1	Value of 2nd Realtime Curve is following
2	Value of 3rd Realtime Curve is following
3	Value of 4th Realtime Curve is following

A status bit of the Sync Byte is set for each following realtime value. The bit-positions are related to the datastream-number in the same order as specified within the "Configure Realtime-Tranmission" command by the data-requesting device before. A missing value is replaced by the value of the next curve. In this case the total number of transmitted data bytes is reduced by 2 or multiple of 2, respectively.

Example (binary):

11010101 10110000 10100011 10001011 10111100

Sync Byte value of 1st curve 8F0H value of 3rd curve F0BH

2nd value missing

If more than 4 realtime data have been configured concurrently, their appearance is indicated using the Sync-Commands "Transmitted Datastreams". These commands are transmitted once each time the order or number of the transmitted datastreams 5 to 12 **changes** (e.g. if a data is newly available or missing).

The Sync Byte is sent only heading a data record, never alone.

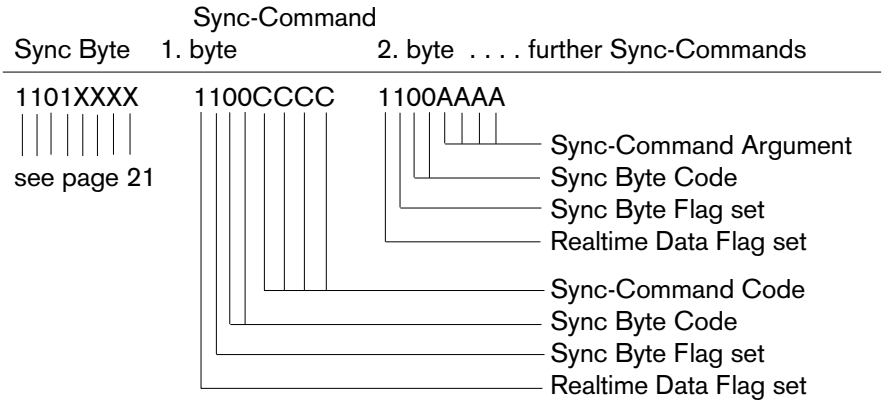
Sync-Commands

Sync-Commands are used for information which have to be sent very fast (e.g. enable/disable datastreams) or which belong to the realtime data directly following the Sync-Commands.

A Sync-Command is not responded to!

Sync-Commands are embedded within a realtime-data record (see chapter 5). Up to 16 Sync-Commands may optionally follow the Sync Byte.

A Sync-Command consists of two Sync-Command-Bytes:



The first Sync-Command byte contains the Sync Command Code in bits 0 to 3. The second byte holds the argument in bits 0 to 3 also.

The following Sync-Commands are defined currently:

Command Code	Argument	Meaning
1100 0000	11000000	End of Sync-Command Sequence
1100 0001	1100 xxxx	Datastream enable/disable (data 1-4) (*)
1100 0010	1100 xxxx	Datastream enable/disable (data 5-8) (*)
1100 0011	1100 xxxx	Datastream enable/disable (data 9-12) (*)
1100 0100	1100 xxxx	Transmitted Datastreams (data 5-8) (*) (**)
1100 0101	1100 xxxx	Transmitted Datastreams (data 9-12) (*) (**)
1100 0110	1100 xxxx ||||	Synchronisation Commands:
	0000	– Start of Ventilator Inspiratory Cycle
	0001	– Start of Ventilator Expiratory Cycle
1100 1111	11000000	Corrupt Data-Record received

(*):
The bits 0 to 3 within the argument are related to the datastreams in the same order as the Sync-Byte's status-bits to the datastream 1 to 4 (see 5.1). A set bit (1) enables, a resetted bit (0) disables the attached datastream.

(**):
"Transmitted Datastreams" commands must be sent only if the order of transmitted datastreams has changed. The datastreams will be sent in a fixed order, until a new "Transmitted Datastream" command announces a new order.

Framing of sync-command sequences:

Sync-Commands are always headed by a Sync Byte as described in section 5.1.

Within a data record the end of a sequence of Sync-Commands is indicated by the first Data-Value Byte.

If the Sync-Commands are transmitted without any realtime-data, the Sync Byte (see 5.1) must be set to 1101 0000 (no data) and the last Sync-Command Byte must be followed by the Sync-Command "End of Sync-Command Sequence" (11000000 11000000).

MEDIBUS-Life-Cycle

Communications Start-up

After starting the device it must send an ICC-command. This command may be repeated each 3 seconds if no character was received or each 10 seconds if characters have been received but no complete response within this time.

On receipt of an ICC-response a "Request Device-Identification"-command may be sent if required.
After having received and processed a correct Device-Identification the device is now in an active protocol state. It is ready to answer Data Request Commands or to send any command on its own.

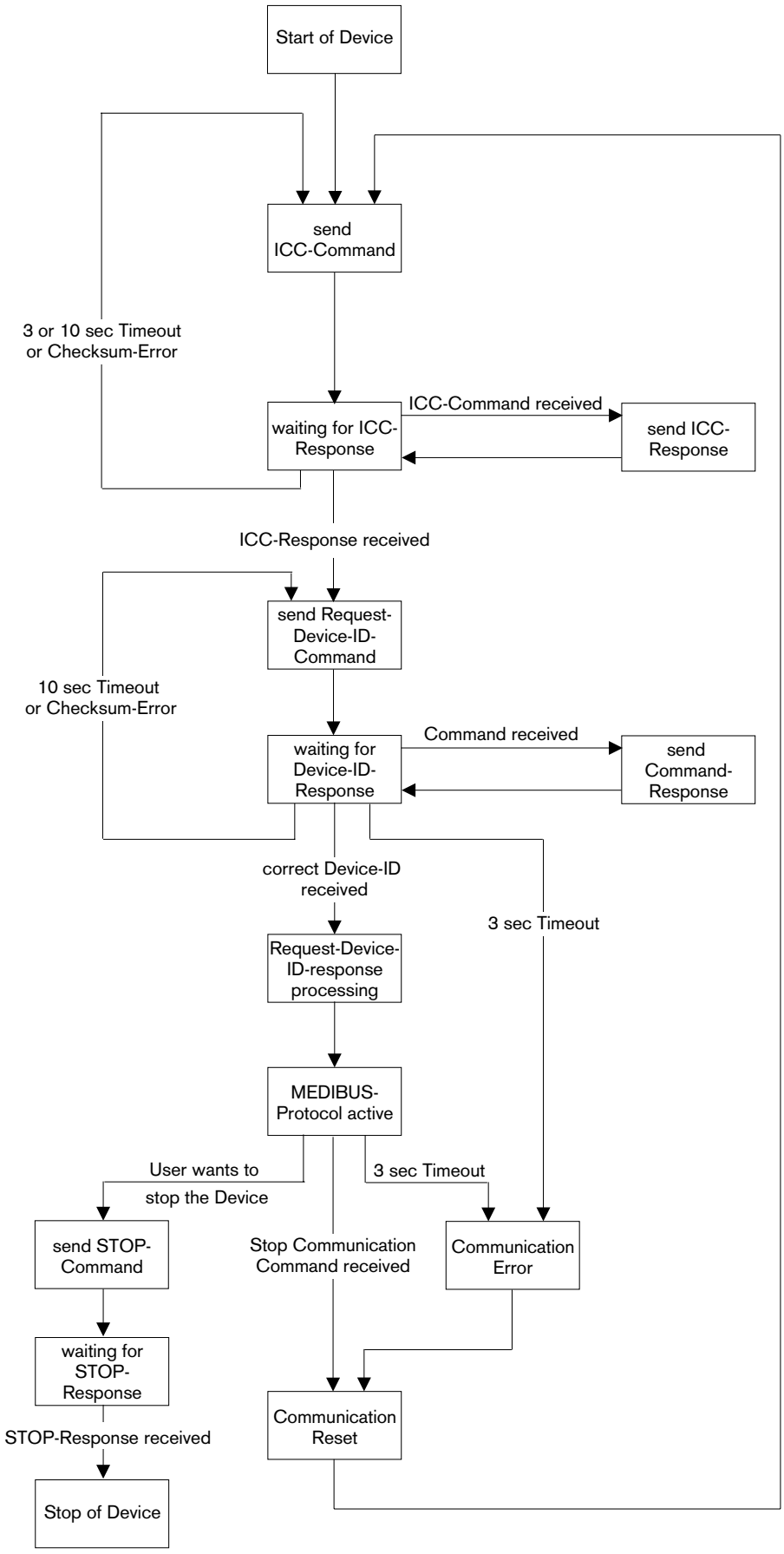
Communications Time-Out

If no character is received within 3 seconds the communication is assumed to be broken. In case of a 3 seconds time-out, communication must be re-initialized by sending an ICC-command.
If communication is in an idle state (no command or response pending) the NOP-command may be sent by any device every 2 seconds to keep communication alive and to avoid a 3 seconds time-out.

Terminating Communication

To terminate communication regular the STOP-command must be sent. After receipt of the STOP-response the device may completely stop the communication.

The STOP-command must be sent to avoid an error-state and error-communication on the linked device.


Start of Device
send ICC-Command
3 or 10 sec Timeout or Checksum-Error
waiting for ICC-Response
ICC-Command received
send ICC-Response
ICC-Response received
send Request-Device-ID-Command
10 sec Timeout or Checksum-Error
waiting for Device-ID-Response
Command received
send Command-Response
correct Device-ID received
3 sec Timeout
Request-Device-ID-response processing
MEDIBUS-Protocol active
User wants to stop the Device
3 sec Timeout
send STOP-Command
Stop Communication Command received
Communication Error
waiting for STOP-Response
Communication Reset
STOP-Response received
Stop of Device

MEDIBUS Realtime-Extension – Life-Cycle

Communications Start-Up

For the general startup sequence please refer to MEDIBUS-life Cycle.

The first step is to request for the supported realtime-data using the "Request Realtime-Data Configuration" Command.

Data-transmission must be configured by the "Configure Realtime-Transmission" Command, with the argument describing the requested realtime data.

All or single realtime-datastreams can be enabled using the "Enable/Disable Datastream" Commands, depending on how many different realtime data have been configured and which shall be enabled.

The enable/disable state of the datastreams can be changed by sending a new "Enable/Disable Datastream" Command with new arguments.

After receipt of an "Enable/Disable Datastream" Command the device starts transmission of the defined realtime data at the defined rate. Transmission is continued until receipt of new "Enable/Disable Datastream" Command or a "Configure Realtime-Transmission" Command.

Since there is no feedback about the correct receipt of a Sync-Command it is recommended to repeat a "Datastream enable/disable"-Sync-Command in an interval of some seconds. This will avoid mis-interpretation of datastreams for a long time if a previous Sync-Command has got lost or been changed during transmission.

Transmission is performed in the format described above.

Re-configuration of data-transmission

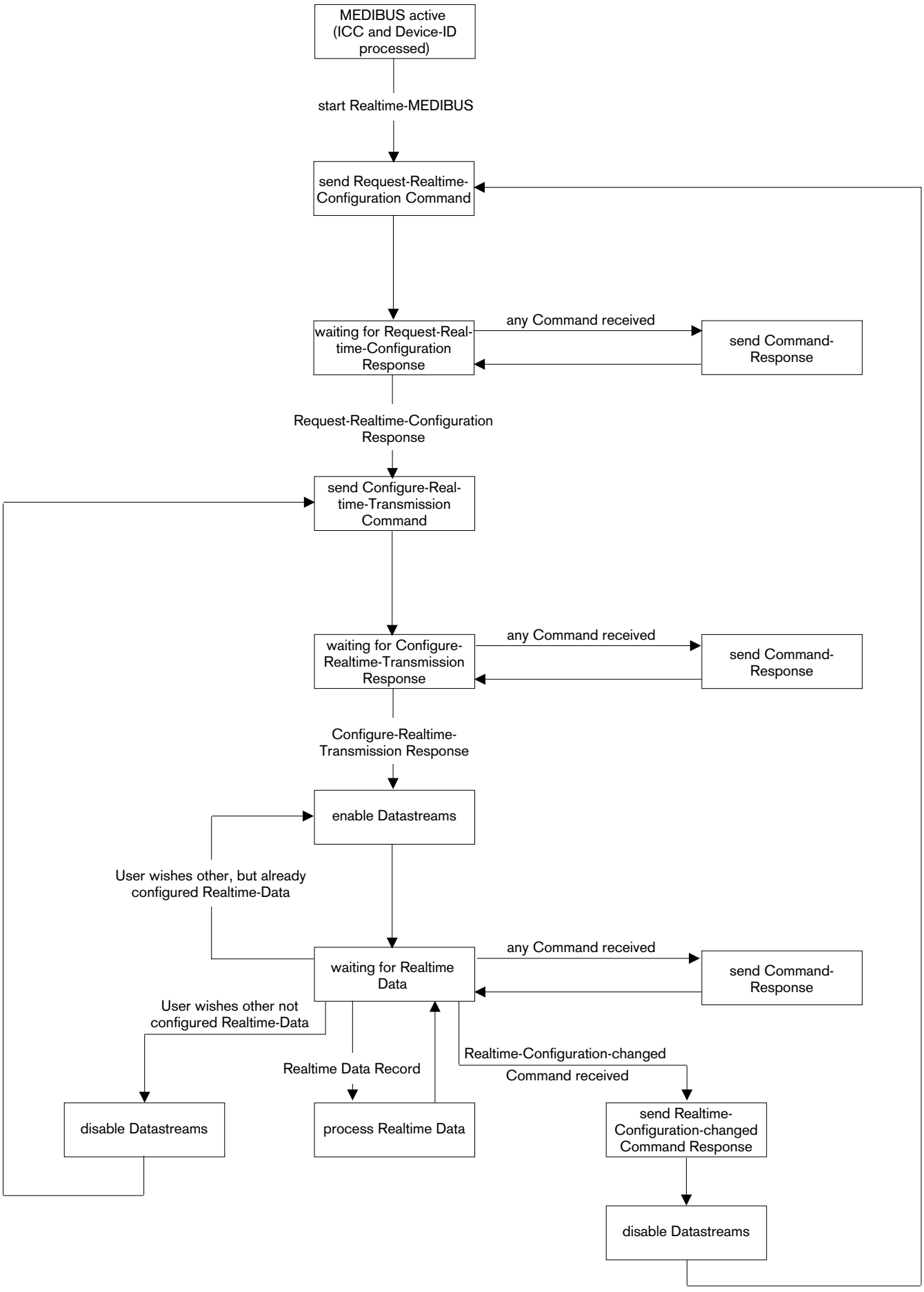
Re-configuration by receiver:

After realtime-data transmission has been configured and enabled once, using the "Configure Realtime-Transmission" command a new configuration may be sent at any time. The realtime data source will change the configuration after receipt of the configuration command. To keep the moment of change under control the realtime data receiver is recommended to disable all or only the changed datastreams by using the "Enable/Disable Datastream" commands before sending the new configuration and to enable the datastreams after receipt of the response to its configuration command.

Re-configuration by sender:

If the realtime-data source changes its realtime configuration (e.g. due to a change of resolution of any trace) it has to stop all transmissions of realtime-data to avoid misinterpretations of data by the receiving device.

As a second step the source has to send the "Realtime Configuration changed" command. Afterwards the receiving device may repeat the realtime-data configuration sequence with a "Request Realtime-Data Configuration" command.


MEDIBUS active (ICC and Device-ID processed)
start Realtime-MEDIBUS
send Request-Realtime-Configuration Command
waiting for Request-Real-time-Configuration Response
any Command received
send Command-Response
Request-Realtime-Configuration Response
send Configure-Real-time-Transmission Command
waiting for Configure-Realtime-Transmission Response
any Command received
send Command-Response
Configure-Realtime-Transmission Response
enable Datastreams
User wishes other, but already configured Realtime-Data
waiting for Realtime Data
any Command received
send Command-Response
User wishes other not configured Realtime-Data
Realtime Data Record
Realtime-Configuration-changed Command received
disable Datastreams
process Realtime Data
send Realtime-Configuration-changed Command Response
disable Datastreams

ASCII HEX Format

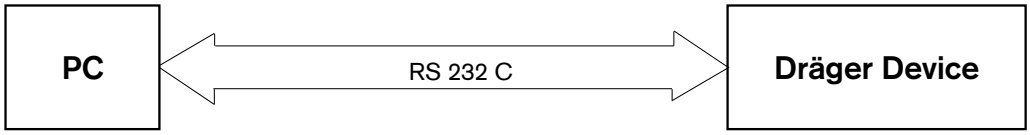
Alarm codes, data codes and checksums are transmitted in ASCII HEX format.

This format uses two bytes to represent a one-byte hexadecimal value in ASCII characters. The HEX numbers A – F are represented in upper case letters only.
Leading zeroes are not suppressed.

Example:

Decimal	=> Binary	=> HEX(H)	=> ASCII HEX Representation	
58	=> 0011 1010	=> 3AH	=> ASCII "3" (33H)	ASCII "A" (41H)
			0	1 2 Byte

MEDIBUS example 1



Dräger Device issues the **ICC** command:

ESC	ICC	CHECKSUM		CR
1BH	51H	36H	43H	0DH

PC responds:

SOH	ICC-Echo	CHECKSUM		CR
01H	51H	35H	32H	0DH

PC issues the device **ID** command:

ESC	Request Device Identity	CHECKSUM		CR
1BH	52H	36H	44H	0DH

Dräger Device responds:

SOH	Command Echo	ID Number	Device Name	Device Revision	MEDIBUS Revision	CHECKSUM		CR
01H	52H	8888	'Draeger Device '	03.00	03.00	38H	43H	0DH

Dräger Device issues the device **ID** command:

ESC	Request Device Identity	CHECKSUM		CR
1BH	52H	36H	44H	0DH

PC responds either with a complete Device Identification:

SOH	ID Echo	ID Number	Device Name	Device Revision	MEDIBUS Revision	CHECKSUM		CR
01H	52H	0161	'My PC'	01.00 : 03.00		46H	43H	0DH

or with an empty Device Identification:

SOH	ID Echo	CHECKSUM		CR
01H	52H	35H	33H	0DH

During a period of idleness: Dräger Device issues **NOP commands every 2 seconds.**

ESC	NOP	CHECKSUM		CR
1BH	30H	34H	36H	0DH

MEDIBUS example 2

Data update command and response

To request data from a device, the following command is sent:

ESC	Command CODE	CHECKSUM		CR
1BH	24H	33H	46H	0DH
0	1	2	4	5 Byte

If the receiving device has no data available, the response will be:

SOH	Command ECHO	CHECKSUM		CR
01H	24H	32H	35H	0DH
0	1	2	4	5 Byte

If the receiving device has »O2 SAT 98 %« and »OXI PULSE 70/min« to send, the response will be:

SOH	Echo	Data Code		O2 SAT ASCII Data 98 %				Data Code		PULSE ASCII Data 70				Checksum		CR
-	-	'E'	'B'	' '	'9'	'8'	' '	'E'	'1'	' '	'7'	'0'	' '	'7'	'A'	-
01H	24H	45H	42H	20H	39H	38H	20H	45H	31H	20H	37H	30H	20H	37H	41H	0DH
0	1	2		4				8		10				14		16 17

MEDIBUS example 3

Configure Data response command

If the receiving device of example 2 is only interested in O2Sat, it has to send the configure Data Response Command.

ESC	Command Code	Datatype	Data Code		CHECKSUM		CR
			'E'	'B'	'1'	'0'	
1BH	4AH	24H	45H	42H	31H	30H	0DH

The Dräger Device responds:

SOH	Command ECHO	CHECKSUM		CR
01H	4AH	34H	42H	0DH

After sending again the request current data command

ESC	Command CODE	CHECKSUM		CR
1BH	24H	33H	46H	0DH
0	1	2	4	5 Byte

The receiving device sends

SOH	Echo	Data Code		O2 SAT ASCII Data 98 %				Checksum		CR
-	-	'E'	'B'	' '	'9'	'8'	' '	'7'	'A'	
01H	24H	45H	42H	20H	39H	38H	20H			0DH
0	1	2		4						

MEDIBUS example 4

Request current date and time.

After Start-up a PC requests current date and time from Dräger Device:

ESC	Request current Date and Time	CHECKSUM		CR
1BH	28H	34H	33H	0D

The Dräger Device responds as follows:

SOH	Echo	Time								Date									Checksum		CR
-	-	'1'	'1'	':'	'1'	'1'	':'	'1'	'9'	'1'	'1'	'–'	'N'	'0'	'V'	'–'	'9'	'3'	'E'	'6'	
01H	28H	31H	31H	3AH	31H	31H	3AH	31H	39H	31H	31H	2DH	4EH	4FH	56H	2DH	39H	33H	45H	36H	0DH

After a while the user changes the time and date setting on the Dräger Device, which therefore sends the "Time changed" command:

ESC	Time changed	CHECKSUM		CR
1BH	49H	36H	34H	0DH

The PC echos

SOH	Echo	CHECKSUM		CR
01H	49H	34H	41H	0DH

After reception of the "Time changed" command the PC is informed about a new time and date setting and may now, before, while or after transmitting the command-echo send again the "Request current date and time" command to get the new setting.

MEDIBUS example 5

Realtime-Extension

After general start-up procedure a PC requests a Dräger Device for its Realtime Configuration:

ESC	Request Realtime Configuration	CHECKSUM		CR
1BH	53H	36H	45H	0D

The Dräger Device offers Airway Pressure (Code 00H) in a range of –10 to 100 mbar, where 100 mbar is represented as $370_{hex} = 880_{dec}$. And CO_2 (Code 06H) in a range of –20 to 100 mmHg and 100 mmHg represented as $460_{hex} = 1120_{dec}$. Both values will be transmitted at a minimal interval of 16 ms.

SOH	Echo
01H	53H

Code		Interval								Minimal Value					Maximal Value					MaxBin		
'0'	'0'	' '	' '	' '	'1'	'6'	'0'	'0'	'0'	'–'	' '	' '	'1'	'0'	' '	' '	'1'	'0'	'0'	'3'	'7'	'0'
30H	30H	20H	20H	20H	31H	36H	30H	30H	30H	2DH	20H	20H	31H	30H	20H	20H	31H	30H	30H	33H	37H	30H

Code		Interval								Minimal Value					Maximal Value					MaxBin		
'0'	'6'	' '	' '	' '	'1'	'6'	'0'	'0'	'0'	'–'	' '	' '	'2'	'0'	' '	' '	'1'	'0'	'0'	'4'	'6'	'0'
30H	36H	20H	20H	20H	31H	36H	30H	30H	30H	2DH	20H	20H	32H	30H	20H	20H	31H	30H	30H	34H	36H	30H

Checksum		CR
'3'	'B'	
33H	45H	0DH

PC now requests each second sample of Airway Pressure and each third sample of CO_2, respectively:

ESC	Configure Realtime Transmission Command Code	Code		Multipl		Code		Multipl		Checksum		CR
		'0'	'0'	'0'	'2'	'0'	'6'	'0'	'3'	'F'	'A'	
1BH	54H	30H	30H	30H	32H	30H	36H	30H	33H	46H	41H	0DH

Dräger Device sends a command acception response without transmitting any realtime data yet:

SOH	Echo	Checksum		CR
01H	54H	35H	35H	0DH

PC enables both realtime traces:

Sync Byte	Sync Command enable Traces 1 + 2		End of Sync Command Sequence	
D0H	C1H	C3H	C0H	C0H

Dräger Device starts periodical transmission of Realtime Data:

time t = 0: One realtime value for each trace is sent:

Sync Byte	Realtime Value Airway Pressure		Realtime Value CO_2	
D3H	91H	81H	8DH	83H

Sync Byte =
D3H = 1101 0011_{bin} indicates realtime values 0 and 1 are following because bits 0 (LSB) and 1 are set.

Airway Pressure =
91H 81H = 1001 0001_{bin} 1000 0001_{bin}
=> Regarding the format of Realtime-Value-Bytes (see page 17)
the twelve bit realtime value is 0000 0101 0001_{bin} = 81_{dec}
=> the according Airway Pressure value $v(x_{bin})$ in mbar is (regard page 20):
$v(x_{bin})$ = –10 mbar + 81 * (100 mbar – (–10 mbar)) / 880 = 0.125 mbar

CO_2 =
8DH 83H = 1000 1101_{bin} 1000 0011_{bin}
=> twelve bit binary value 0000 1100 1101_{bin} = 205_{dec}
=> CO_2 value $v(x_{bin})$ in mmHg:
$v(x_{bin})$ = –20 mmHg + 205 * (100 mmHg – (–20 mmHg))/1120 = 1.96 mmHg

time t = 0 + 1 * 16 ms: No Realtime values are sent.

time t = 0 + 2 * 16 ms: Only a value for Airway Pressure is sent:

Sync Byte	Realtime Value Airway Pressure	
D1H	94H	81H

Sync Byte D1H = 1101 0001_{bin} has only bit 0 set, indicating only the first configured trace Airway Pressure is following the Sync Byte.
The second trace CO_2 is missing.

time t = 0 + 3 * 16 ms: Only a value for CO_2 is sent:

Sync Byte	Realtime Value CO_2	
D2H	8EH	83H

Sync Byte D2H = 1101 0010 $_{bin}$ has only bit 1 set, indicating only the second configured trace CO_2 is following the Sync Byte. The first trace Airway Pressure is missing.

At each start of ventilator inspiratory cycle Dräger Device may inform PC about that event so PC may synchronize its Airway Pressure displaying. In this case Dräger Device will send the "Start of ventilator inspiratory cycle" Sync-Command with the respective realtime value for Airway Pressure:

Sync Byte	Start of Insp. Cycle		Realtime Value Airway Pressure		Realtime Value CO_2	
D3H	C6H	C0H	90H	81H	8AH	83H

Frequently Asked Questions

The following are frequently asked questions showing some typical problems which occur when, for instance, a PC is being programmed to communicate with a Dräger medical device. The respective answers should help the understanding of and provide advice for the programming process.

Note: Within the MEDIBUS protocol definition Dräger medical devices may show varying behavior. For this reason, the following general answers may not apply in each and every case for each and every device.

Do all Dräger medical devices send the ICC command on their own?

Yes. As long as an ICC response and an ICC command are not received, Dräger medical devices send the ICC command approx. every 3 seconds.

Does my PC have to send an ICC command itself, or will it suffice to respond to an ICC command sent by a Dräger medical device?

Responding to the ICC command sent by the Dräger medical device will suffice.

Does my PC have to respond to the ICC command from the Dräger medical device?

Yes. Every command must be responded to.

What kind of identification should my PC send?

For most Dräger medical devices, it is enough to send an empty response. Some devices (e.g. Cicero B and Evita 4) do, however, expect a complete response with specific content. For this reason, it is recommended that a complete response containing the following information is sent:

ID Number:	0161
Name of Device:	any
Device Version:	any
MEDIBUS Version:	the version number for the MEDIBUS protocol definition according to which the PC program was developed

Compare with MEDIBUS Example 1 in this protocol definition.

Does my PC also have to send a device identification request?

Only if your PC program requires this information, otherwise: No

Why isn't the data configuration command processed; in other words: Why does my PC still receive all data when subsequent data request commands are given?

The data configuration command is often sent too early. All Dräger medical devices send a device identification request after the ICC sequence. Your PC needs to respond to this request first (see above) and then allow the Dräger medical device approx. 200ms to process this response. Only after this will the Dräger medical device be ready to process the data configuration command. All previous data configuration commands shall be ignored by the Dräger medical device.

My PC sets up the connection to the Dräger medical device and communicates for some time with it. Why does the Dräger medical device send another ICC command after some time?

In many cases, the command sent by the Dräger medical device is answered either incorrectly or not at all.
A command incorrectly responded to will lead to the command being repeated in a short span of time. A command that is not responded to will be repeated after approx. 10 seconds. After a maximum of 3 repetitions, the Dräger medical device assumes that the connection is faulty and tries to re-establish communication using an ICC command.
Thus, if communication only stands for a few seconds, then your PC is probably sending an incorrect response (e.g. a response with an incorrect format or check sum).
If communication is stable for approx. 30 to 40 seconds before the Dräger medical device sends another ICC command, then your PC probably isn't responding to the command from the Dräger medical device at all. (Embedded commands often go unnoticed. In many cases, for example, the device identification request from the Dräger medical device is embedded in the ICC response.)

Will my PC also be asked for data?

If your PC sends a device identification response with the information given above, then: No.

Which commands shall be sent to my PC from the Dräger medical device?

There is, for every device, a table of “Transmitted Commands” in the instructions for use "MEDIBUS for Dräger Anesthesia Devices", "MEDIBUS for Dräger Pediatric Devices" and "MEDIBUS for Dräger Intensive Care Devices".

Note: A flexible PC program design that allows every command received from the Dräger medical device to be responded to at any given time has often proven very valuable. In such cases, it should make no difference whether or not the command is known and, as expected, comes at a specific time, or if the command comes at an unexpected time or is even completely unknown.
Communicating with the Dräger medical device should pose no problem if every command is responded to with at least an empty response (for exceptions see “device identification response”). Commands which are not responded to, however, always lead to an interruption in communication.

When does my PC has to send or respond to an NOP command?

An NOP command should be sent when other commands (e.g. data request commands) should **not** be sent to the other device for more than 2 seconds, when communication should continue to stand and, for this reason, an exceeding of the 3-second time-out has to be prevented.
The Dräger medical device shall send an NOP command if it has received neither command nor response from the PC for approx. 2 seconds, and if it itself did not send a previous command and is, therefore, not waiting for a response. Thus, it will suffice if the PC sends an NOP response to the Dräger medical device within approx. 1 second. It is, however, better and provides better protection against communication errors caused by exceeding time-outs if the PC sends an NOP command itself in such cases.

How often can my PC send a command?

As soon as the response to the preceding command has been received, the next command can be sent.
If the PC sends a command before the Dräger medical device could respond in full to the preceding command, the Dräger medical device shall ignore the previous command and begin processing and responding to the new command.

How long does it take the response to come from the Dräger medical device?

This depends on the Dräger medical device, the command, the length of the response and the Baud rate. For this reason, this question cannot be answered completely. Usually, far less than 500ms pass between the completed reception of a command and the transfer of the first character of the response.

What happens after a 3-second time-out?

If the Dräger medical device detects a 3-second time-out (no signal from the PC for 3 seconds or more), communication shall be re-initialized. Insofar as the Dräger medical device has a corresponding possibility, a communication error will be optically and acoustically displayed until the re-initialization has been successfully completed.

What happens after a 10-second time-out?

When a 10-second time-out occurs (no complete response to a command takes place within 10 seconds), the last command sent by the Dräger medical device shall be repeated; up to three times. After that, communication is re-initialized and the communication error, as was the case for the 3-second time-out, displayed.

Does my PC have to monitor 3-second and 10-second time-outs? Or what happens if these time-outs cannot be monitored or cannot be monitored accurately by my PC?

The 3-second time-out allows the PC to detect whether or not communication with the Dräger medical device stands or has been interrupted. Communication can be re-initialized with an ICC command if there was an interruption and the user would see the communication error displayed.
The 10-second time-out allows the PC to monitor whether or not the last command sent to the Dräger medical device was completely and correctly received. In the case of an error, the last command can be repeated or a new command can be sent.
If the PC does not have these kinds of requirements, the time-outs on the PC either do not need to be monitored or only monitored less strictly. (Dräger medical devices always monitor these time-outs.)

Is all information and are all configurations sent to the Dräger medical device before the re-initialization lost during a communication re-initialization (ICC), e.g. after a 3-second or a 10-second time-out, and does everything have to be re-sent for this reason?

Yes. All information in the Dräger medical device sent previously will be deleted with every communication (re-)initialization.

Where can commands be embedded in responses?

Commands can be embedded in responses at any position.

Where can real-time data records be embedded in responses and commands?

Real-time data records can be embedded in responses and commands at any position.

Will the ESC for commands / the SOH for responses be calculated into the check sum?

Yes. See the MEDIBUS examples in this instruction for use.

What influence does a command which is embedded in a response have on the check sum calculation?

None. Check sums from commands and responses are always calculated separately. Thus, a command embedded in a response does not influence the check sum of the response or vice versa.

Do real-time data records embedded in commands or responses influence the check sum?

Not at all. Real-time data records are ignored by the check sum calculation.

What does the Stop command do?

Sending a Stop command to the Dräger medical device prevents the Dräger medical device from constantly displaying communication errors after the communication has been ended.

Logbook of Changes

Version	Changes	backward compatible
2.00	First released verision	——
3.00	Protocol extended by – "Device Settings" – "Text Messages" – Realtime transmission	yes
4.00	Protocol extended by second codepage for current measured Data, Alarm Limits and Alarms	yes
4.01	New chapter "Frequently Asked Questions". No changes of protocol	yes
4.02	Correction of chapter "Time & Date Update Response"	no
4.03	New Realtime Sync-Command "Start of Ventilator Expiratory Cycle"	yes

Table of used ASCII-Codes

Code	Explanation	Hexadecimal code
SOH	Start of Header Start of Respondings	01H
ETX	End of Text	03H
CR	Carriage Return, End of Commands or Respondings	0DH
DC1	Device Control 1 Transmitting Device suspends Transmission immediately	11H
DC3	Device Control 3 Transmitting Device resume Transmission	13H
NAK	Negative Acknowledge Character for corrupt Commands	15H
CAN	Any Transmission in Progress will be immediately abborted.	18H
ESC	Escape Start of Commands	1BH
Space	Blank Key Surplus Character Positions (Data and Settings) must be filled up with this Character. Character to fill up leading Zeroes.	20H
20H – 7FH	Range used for the "slow" MEDIBUS-Protocol	
80H – FFH	Range used for the MEDIBUS Realtime-Extension	

Dräger Medical AG & Co. KGaA
Germany
Moislinger Allee 53 – 55
D-23542 Lübeck
(4 51) 8 82 - 0
26 80 70
FAX (4 51) 8 82-20 80
http://www.draeger.com

90 28 258 - GA 6494.380 en

7th edition - September 2003
Subject to alteration