12. Interface

12.1 Host Interface

The **Miditron**® *Junior II* appliance has a serial interface connection to the customer's laboratory EDP system, hereinafter called the „Host". These specifications are concerned with the exchange of data, its activation, protocol formats and timing.
These specifications are based on the host connection of the **Miditron**® *Junior II* appliance
The **Miditron**® *Junior II* provides the following data: Date and time of measurement, the findings obtained from the urine test strip and sequence number of the measurement.
The **Miditron**® *Junior II* appliance does not handle patient identification.
Data transfer is thus restricted to upload functions.
Uploading is possible only after a measurement or at the end of a measurement series.
The customer initiates transfer by pressing the reprint/send key. The appliance setup must be set to "Host/PC On" to this end. All the findings which have not previously been transmitted are then transmitted.
The Host itself cannot request the measurement results.

12. Interface

12.2 Character definitions, representation conventions

Table for Character definitions:

No.	Abb.	Meaning	Representation
1	CR	ENTER	$0D_{16}$
2	LF	Line-Feed	$0A_{16}$
3	chr	ASCII-character to DIN 66003 Table 1 (International Reference version)	20_{16} to $7D_{16}$
4	txt	Letter characters or spaces	20_{16}, 41_{16} to $7D_{16}$
5	num	Numbers/ punctuation marks	30_{16} to 39_{16} and ",", ";", ".", ":", "-", "_", "/", "\", "(", ")", "+", "=", "|", "*".
6	STX	Start of protocol	02_{16}
7	ETX	End of protocol	03_{16}
8	SPE	"Specific Sample" : Data	$3B_{16}$
9	SPE-E	"Specific Sample" : E = Result	$3B_{16}$ 45_{16}
10	SPE-D	"Specific Sample" : D = color + turbidity	$3B_{16}$ 44_{16}
11	SPE-A	"Specific Sample" : A = Pat. ID. Download	$3B_{16}$ 41_{16}
12	SPM	"Specific Multiple"	$3C_{16}$
13	ANY	Any Inquiry	$3E_{16}$
14	MOR	More	$3E_{16}$
15	REP	Repeat	$3F_{16}$
16	END	End	$3A_{16}$
17	SP	Space	20_{16}

Table: tab-12e

Table for Representation conventions:

1	' xy '	**All the characters between the inverted commas are transmitted as ASCII characters, spaces included**
2	20xSP	20 spaces are transmitted consecutively.
3	3xnum	3 number characters are transmitted consecutively.
4	25x-	The character "-" is transmitted 25 times consecutively.
5	10xtxt	10 text characters are transmitted.
6	10xchr	10 characters are transmitted according to character definition 3 (see above).

Table: tab-11e

The numbers in this convention agreement are intended as examples only.

12. Interface

12.3 Protocols

No protocol is longer than 255 bytes. If a data field exists, it will consist of a function code, one space (or „Spare“, to use Hitachi's nomenclature, space = 20_{16}) and the data in question. The length of the block is neither coded nor transmitted. Protocol length is clearly defined in these specifications and so can be checked easily. The frame code indicates the purpose and task of the block. Much of this is purely „useful information“ and must be interpreted as a command or request.

Each act of communication takes the form of a cycle: The **Miditron**® *Junior II* transmits and awaits a reply. The **Miditron**® *Junior II* analyses communication status. It measures the Host's response times, checks block storage and the frame code of the replies. Erroneous protocols are not accepted. (Erroneous here means that at least one of the before mentioned tests does not produce the result expected; the cause may be a software error in the Host or a disturbance in the line). The Host must also check: Is the frame code permissible and block storage okay?
The handshake is represented as an interaction diagram and as a status transition diagram.
Every protocol transferred is transferred as a block. The blocks from Host and **Miditron**® *Junior II* are put together as follows:

12. Interface

Table for Protokcol structure:

Pat Id length of 10 characters

	Header				Data field					Trailer			
Transmitter	SC	FK	FC	SP	Pat-ID	Seq.-No.	Date	Time	Data	EC	CS	CR	Σ Bytes
Host	STX	SPE	A	SP	10 chr +SP					ETX	CS1,2	CR	19
Miditron	STX	SPE	D	SP	10 chr +SP	5 num. +SP	8 num. +SP	5 num. +SP	38 char	ETX	CS1,2	CR	78
Miditron	STX	SPE	E	SP	10 chr +SP	5 num. +SP	8 num. +SP	5 num. +SP	196 char	ETX	CS1,2	CR	236
Miditron	STX	SPM								ETX	CS1,2	CR	6
Miditron	STX	ANY								ETX	CS1,2	CR	6
HOST	STX	MOR								ETX	CS1,2	CR	6
Miditron/ HOST	STX	REP								ETX	CS1,2	CR	6
Miditron/ HOST	STX	END								ETX	CS1,2	CR	6

Table: tab-10e

Pat Id length of 13 characters

	Header				Data field					Trailer			
Transmitter	SC	FK	FC	SP	Pat-ID	Seq.-No.	Date	Time	Data	EC	CS	CR	Σ Bytes
Host	STX	SPE	A	SP	13 chr +SP					ETX	CS1,2	CR	22
Miditron	STX	SPE	D	SP	13 chr +SP	5 num. +SP	8 num. +SP	5 num. +SP	38 char	ETX	CS1,2	CR	81
Miditron	STX	SPE	E	SP	13 chr +SP	5 num. +SP	8 num. +SP	5 num. +SP	196 char	ETX	CS1,2	CR	239
Miditron	STX	SPM								ETX	CS1,2	CR	6
Miditron	STX	ANY								ETX	CS1,2	CR	6
HOST	STX	MOR								ETX	CS1,2	CR	6
Miditron/ HOST	STX	REP								ETX	CS1,2	CR	6
Miditron/ HOST	STX	END								ETX	CS1,2	CR	6

Table: tab-33e

12. Interface

TThe „protocol header“ or just „header“ means the start character, the frame code, function and the spare which follows (columns 1, 2, 3 and 4). The „protocol trailer“ or just „trailer“ consists of the end character, the test bytes and return (columns 10, 11 and 12).

The frame codes have the following meanings:

END	=	$3A_{16}$ =	‘:’				
SPE	=	$3B_{16}$ =	‘;’	MOR	=	$3E_{16}$	= ‘>’
SPM	=	$3C_{16}$ =	‘<‘	REP	=	$3F_{16}$	= ‘?’

12. Interface

12.4 Upload timing and handshake

The protocols are identified by abbreviated codes on the arrows symbolizing the direction of transfer. Those protocols which appear more than once are executed more than once.

Miditron® *Junior II* (Master)		**HOST (Slave)**
Start communication after pressing reprint/send key: Host is asked whether it is ready to receive	/SPM/ ——————>	
	<——————/MOR/ -	Reply
		If disturbance or not ready to recieve, no reply.
If no reply (15 s time out):	/SPM/ ——————>	as above
If disturbance:	/REP/ ——————>	Repeat above command; no command if disturbance

Repeat /SPM/, /REP/ max.4 times then no more communication.

If received /MOR/	/SPE/-E/ ——————>	
	<——————/MOR/	If received
	<——————/REP/	If disturbance
If /MOR/ received, transmit color + turbidity	/SPE/-D/ ——————>	as above
If /REP/ received, transmit previous finding	/SPE/-E/ ——————>	as above
If no reply (15 sec. time out):	/SPM/ ——————>	as above
If disturbance:	/REP/ ——————>	as above

Repeat /SPM/,/REP/ max. 4 times, then no more communication.

If /MOR/ received:	/SPE/-E ——————>	
	<——————/MOR/	If received
	<——————/REP/	If disturbance

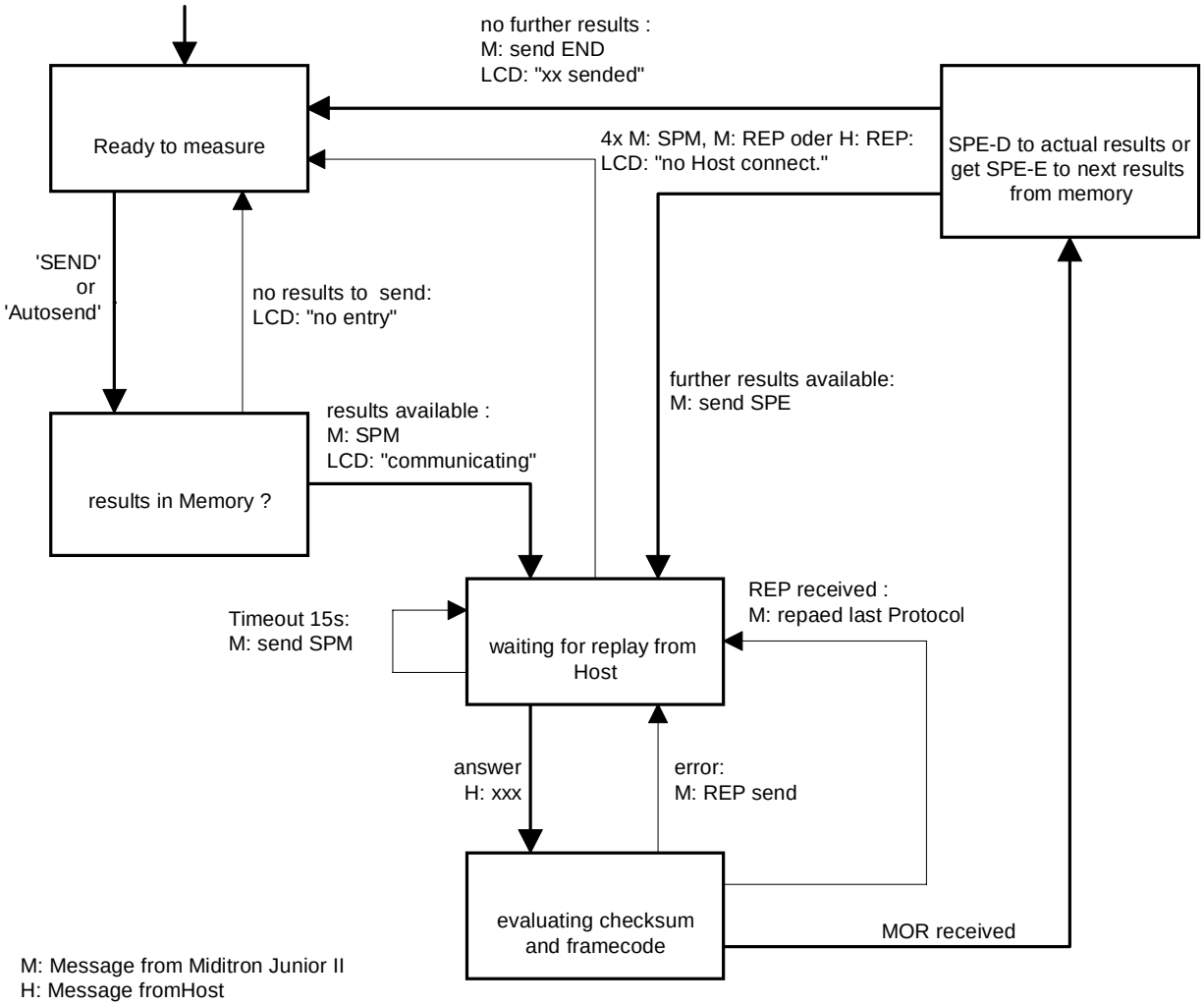
etc. until

End of communications:	/END/ ——————>	(No handshake for /END/)

„/SPE/-D/ and SPE-E“ is explained in more detail in Section 12.6 All data sets have the frame code /SPE/. „Data“ contains findings which have not yet been sent after a correctly received /MOR/ or, in the case of /REP/, a repeat of the last findings to be sent. If the transmission of /SPM/ or /REP/ by the **Miditron®** *Junior II* is necessary between two /SPE/ data sets, the last findings will be repeated after the next /MOR/.

12. Interface

Status Transition Diagram Host Communications (Upload)



12. Interface

12.5 Download timing and handshake

Miditron® *Junior II* (Master)		**HOST (Slave)**
Start communication after pressing "receive list": Miditron is asked whether Host is ready to send	/ANY/ ——————>	
	<——————/SPA-A/-	Reply Pad ID
		If disturbance or not ready to receive, no reply.
If no reply (15 s time out):	/ANY/ ——————>	as above
If disturbance:	/REP/ ——————>	Repeat above command; no command if disturbance

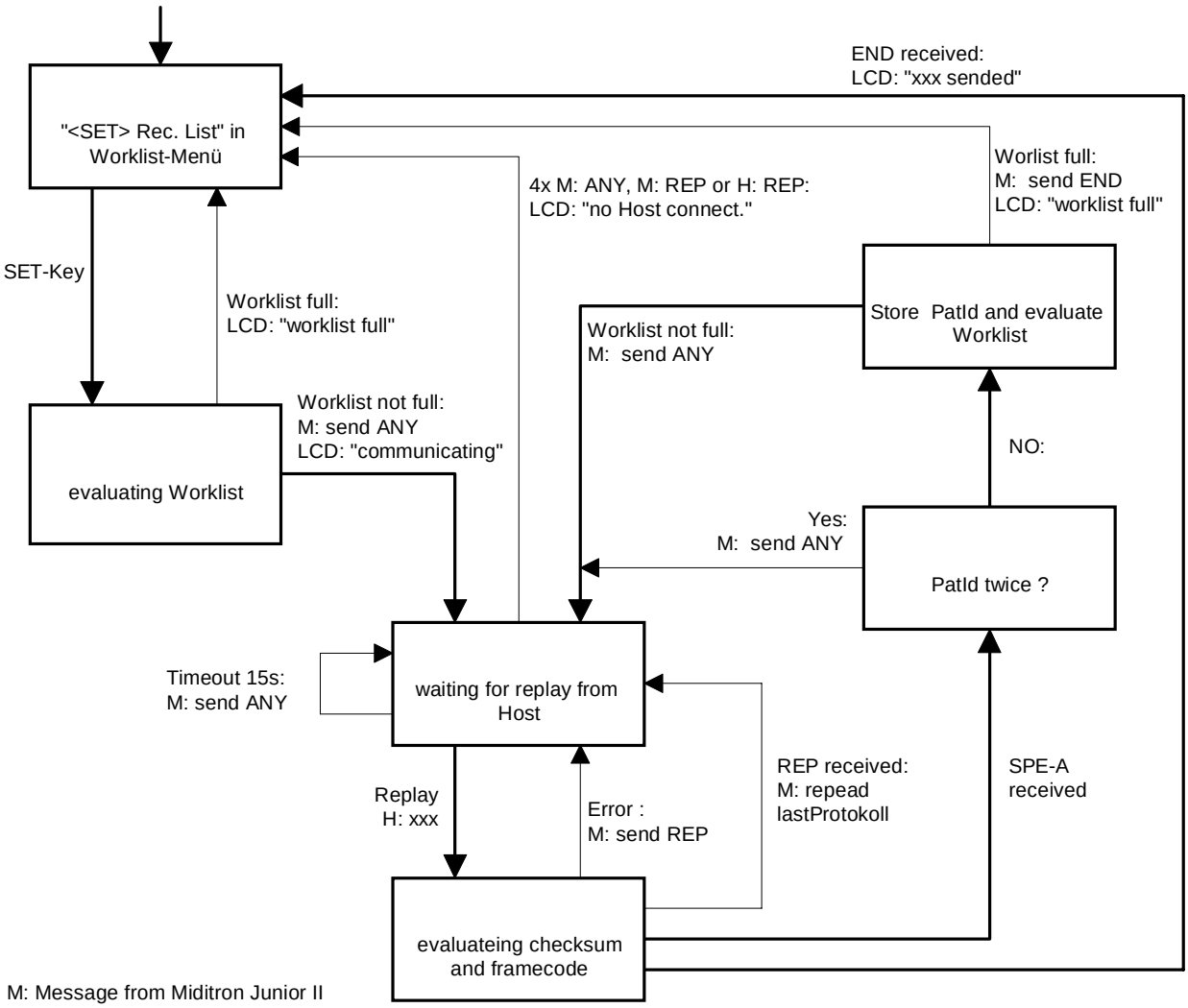
Repeat /ANY/, /REP/ max.4 times then no more communication.

If received /SPE-A/	/ANY/ ——————>	
	<——————/SPE-A/	If received next Pad ID
	<——————/REP/	If disturbance
If /REP/ received, transmit previous finding	<——————/END/	When all Pad Id´s transmitted
Working list is full	/END/ ——————>	No replay
If /REP/ received	/ANY/ ——————>	as above
No replay (15 sec. time out):	/ANY/ ——————>	as above
disturbance:	/REP/ ——————>	as above

Send /ANY/ after disturbance or /REP/ max. 4 times: stop communication

12. Interface

Status Transition Diagram Host Communications (Download)



M: Message from Miditron Junior II
H: Message from Host
bold lines : normal Communication without error

12. Interface

12.6 Protocol structure

12.6.1 Protocol "/REP/": Repeat request

Transmitter	HOST/**Miditron**® *Junior II*	
Byte No.:	Meaning	Comments
1	STX	Start character
2	REP	Frame code; „repeat“;
3	ETX	End code
4	PB1	Test byte 1
5	PB2	Test byte 2
6	CR	Return

12.6.2 Protocol "/SPM/": Start Communication

Transmitter	**Miditron**® *Junior II*	
Byte No.:	Meaning	Comments
1	STX	Start character
2	REP	Frame code
3	ETX	End code
4	PB1	Test byte 1
5	PB2	Test byte 2
6	CR	Return

12.6.3 Protocol "/MOR/": Receipt confirmed/ Request for next set

Transmitter	Host	
Byte No.:	Meaning	Comments
1	STX	Start character
2	REP	Frame code
3	ETX	End code
4	PB1	Test byte 1
5	PB2	Test byte 2
6	CR	Return

12.6.4 Protocol "/END/": End of communication

Transmitter	**Miditron**® *Junior II*	
Byte No.:	Meaning	Comments
1	STX	Start character
2	REP	Frame code
3	ETX	End code
4	PB1	Test byte 1
5	PB2	Test byte 2
6	CR	Return

12.6.5 Protocol "/SPE-D/ + Data": Data protocol color + turbidity

Transmitter: **Miditron®** *Junior II*

Byte Nr. for 10	Byte Nr. for 13	meaning Pat-Id-length	comment
1	1	STX	Start character
2	2	SPE	Frame character; $3B_{16}$; ';'
3	3	'D'	Block D; 'D' = 44_{16}
4	4	SP	Space
5 .. 14	5 .. 17	Pat-Id	length 10 or 13, as chosen in Setup
15	18	SP	Space
16 .. 20	19 .. 23	Seq. Nr	sequenz -Number of result
21	24	SP	Space
22 .. 29	25 .. 32	date	Datum of result
30	33	SP	Space
31 .. 35	34 .. 38	time	time of result
36	39	SP	Space
37 .. 54	40 .. 57	color	color left handed orientated
55	58	SP	Space
56 .. 73	59 .. 76	turbidity	turbidity left handed orientated
74	77	SP	Space
75	78	ETX	End character
76	79	CS1	Checksum 1
77	80	CS2	Checksum 2
78	81	CR	Return

12.6.6 Protocol "/SPE-E/ + Data": Data protocol results

Transmitter: **Miditron®** *Junior II*

Byte Nr. for 10	Byte Nr. for 13	meaning Pat-Id-length	comment
1	1	STX	Start character
2	2	SPE	Frame character; $3B_{16}$; ';'
3	3	'E'	Block D; 'E' = 45_{16}
4	4	SP	Space
5 .. 14	5 .. 17	Pat-Id	length 10 or 13, as chosen in Setup
15	18	SP	Space
16 .. 20	19 .. 23	Seq. Nr	Sequenz-Number of result
21	24	SP	Space
22 .. 29	25 .. 32	Datum	Datum of result
30	33	SP	Space
31 .. 35	34 .. 38	time	time of result
36	39	SP	Space
37 .. 232	40 .. 235	Data	results
233	236	ETX	End character
234	237	CS1	Check sum 1
235	238	CS2	Check sum 2
236	239	CR	Return

12.6.7 Protocol "/SPE-A/ + Pat-Id.": Data protocol Pat-Id.

Transmitter:	HOST		
Byte Nr. for 10	Byte Nr. for 13	meaning Pat-Id-length	explanation
1	1	STX	Start character
2	2	SPE	Frame character; $3B_{16}$; ';'
3	3	'A'	Block D; 'D' = 41_{16}
4	4	SP	Space
5 .. 14	5 .. 17	Pat-Id	length 10 or 13, as chosen in Setup
15	18	SP	Space
16	19	ETX	End character
17	20	CS1	Check sum 1
18	21	CS2	Check sum 2
19	22	CR	Return

12.7 Format of results-data :

Byte Nr. of 10-Id	Byte Nr. of 13-Id	field-length	explanation
37... 49	40... 52	13	SG + 5xBef + 6xSP
50... 60	53... 63	11	PH + 3xBef + 6xSP
61... 80	64... 83	20	LEU + 11xres + 5xArb + SP
81... 92	84... 95	12	NIT + 03xres + 5xArb + SP
93...112	96...115	20	PRO + 11xres + 5xArb + SP
113...132	116...135	20	GLU + 11xres + 5xArb + SP
133...152	136...155	20	KET + 11xres + 5xArb + SP
153...172	156...175	20	UBG + 11xres + 5xArb + SP
173...192	176...195	20	BIL + 11xres + 5xArb + SP
193...212	196...215	20	ERY + 11xres + 5xArb + SP
213...232	216...235	20	NAG + 17xSP

11x = number of bits
res = result in Con. or SI units
Arb = Arbitrary units
SP = Space

12.7.1 Structure of results-data Programm-1 (International) :

example : setting "**conventional"** or "**conv & Arb**" ; Programm-1

Byte Nr. for 10-Id	Byte Nr. for 13-Id	field-length	example of results
37... 49	40... 52	13	SG|1.030|▪▪▪▪▪▪|
50... 60	53... 63	11	PH|▪▪7|▪▪▪▪▪▪|
61... 80	64... 83	20	LEU|▪▪▪▪▪▪▪▪neg|▪▪▪▪▪|▪|
81... 92	84... 95	12	NIT|neg|▪▪▪▪▪▪|▪|
93...112	96...115	20	PRO|▪▪▪75▪mg/dl|▪▪▪++|▪|

113...132	116...135	20	GLU|▪1000▪mg/dl|▪++++|▪|
133...152	136...155	20	KET|▪▪▪▪5▪mg/dl|▪▪▪▪+|▪|
153...172	156...175	20	UBG|▪▪▪▪4▪mg/dl|▪▪▪++|▪|
173...192	176...195	20	BIL|▪▪▪▪3▪mg/dl|▪▪▪++|▪|
193...212	196...215	20	ERY|▪▪▪▪▪▪25/ul|▪▪▪++|▪|
213...232	216...235	20	NAG|▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪|

example : setting "**SI"** or "**SI & Arb**" ; Programm-1

Byte Nr. for 10-Id	Byte Nr. for 13-Id	field-length	example of results
37... 49	40... 52	13	SG|1.030|▪▪▪▪▪▪|
50... 60	53... 63	11	PH|▪▪7|▪▪▪▪▪▪|
61... 80	64... 83	20	LEU|▪▪▪▪▪▪500/ul|▪▪+++|▪|
81... 92	84... 95	12	NIT|pos|▪▪pos|▪|
93...112	96...115	20	PRO|▪▪▪0.75▪g/l|▪▪▪++|▪|
113...132	116...135	20	GLU|▪▪56▪mmol/l|▪++++|▪|
133...152	136...155	20	KET|▪0.5▪mmol/l|▪▪▪▪+|▪|
153...172	156...175	20	UBG|▪▪68▪umol/l|▪▪▪++|▪|
173...192	176...195	20	BIL|▪▪50▪umol/l|▪▪▪++|▪|
193...212	196...215	20	ERY|▪▪▪▪▪▪25/ul|▪▪▪++|▪|
213...232	216...235	20	NAG|▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪|

example : setting "**arbitrary**" ; Programm-1

Byte Nr. for 10-Id	Byte Nr. for 13-Id	field-length	example of results
37... 49	40... 52	13	SG|1.030|▪▪▪▪▪▪|
50... 60	53... 63	11	PH|▪▪7|▪▪▪▪▪▪|
61... 80	64... 83	20	LEU|▪▪▪▪▪▪▪▪▪▪|▪▪+++|▪|
81... 92	84... 95	12	NIT|▪▪▪|▪▪pos|▪|
93...112	96...115	20	PRO|▪▪▪▪▪▪▪▪▪▪|▪▪▪++|▪|
113...132	116...135	20	GLU|▪▪▪▪▪▪▪▪▪▪|▪++++|▪|
133...152	136...155	20	KET|▪▪▪▪▪▪▪▪▪▪|▪▪▪▪+|▪|
153...172	156...175	20	UBG|▪▪▪▪▪▪▪▪▪▪|▪▪▪++|▪|
173...192	176...195	20	BIL|▪▪▪▪▪▪▪▪▪▪|▪▪▪++|▪|
193...212	196...215	20	ERY|▪▪▪▪▪▪▪▪▪▪|▪▪▪++|▪|
213...232	216...235	20	NAG|▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪|

12. Interface

12.8 Procedures for checking test bytes

12.8.1 European language variations of Miditron® software: LRC test bytes

The LRC test bytes are a Longitudinal Redundancy Check - a kind of longitudinal parity test of the bits contained in the data protocol. The procedure is quite simple: Byte for byte, the protocol is linked bit by bit to XOR. The resulting byte is then split into two bytes (to avoid the occurrence of control characters) and attached to the protocol.

LRC-Byte = Byte1 XOR Byte2 XOR Byte3 ..... XOR Byte last

LRC1-Byte = high-Nibble (shifted by 4 bits) of the LRC-Byte OR 30_{16}
LRC2-Byte = low-Nibble of LRC-Byte OR 30_{16}

Example:

		Bit-No. 8	7	6	5	4	3	2	1
	1	0	1	1	0	0	0	1	0
	2	0	0	0	0	0	0	1	1
Byte-No.	3	0	0	1	1	0	1	1	0
	4	0	0	1	1	1	0	1	0
	5	0	0	1	1	1	1	1	0
LRC-Byte		0	1	0	1	0	0	1	1

high-Nibble		0	1	0	1				
low-Nibble		0	0	1	1				
LRC 1 - Byte =		0	0	1	1	0	0	0	0 /* 30_{16} */
	OR	0	0	0	0	0	1	0	1 /* high-Nibble */
		0	0	1	1	0	1	0	1
LRC 2 - Byte =		0	0	1	1	0	0	0	0 /* 30_{16} */
	OR	0	0	0	0	0	0	1	1 /* low-Nibble */
		0	0	1	1	0	0	1	1

All bytes beginning with STX (inclusive) up to ETX (inclusive) are taken into account in the formation of the LRC. Some laboratory computers have a manufacturer-specific transmit/receive driver implemented which cuts off the STX in protocols and does not allow it to get into the overriding user software. In this case, the user must first switch off block testing in the Host.

urther examples of ASCII representation:
MOR-Protocol: ☺>♥3? REP-Protocol: ☺?♥3> SPM-Protocol: ☺<♥3=

12.8.2 American/Canadian language version of Miditron® *Junior II* software:

Check total

The check total is reached simply by adding together the bytes to be transferred, the individual bytes being interpreted as positive, whole numbers. STX, BTX and CR are not included in the addition. The result of the addition is taken as modulo 256. The resulting number can be represented by a single byte. The two half-bytes of this number are represented as hexadecimal figures ("0".."9", "A".."F").Initial zeros are included.

$$CS = (Byte_2 + Byte_3 + ... + Byte_{Length-4}) \text{ modulo } 256$$

CS1-Byte $= (CS / 16) + 30_{16}$ for $(CS / 16) \leq 9$
$= (CS / 16) + 37_{16}$ for $(CS / 16) \geq 10$

CS2-Byte $= (CS \text{ modulo } 16) + 30_{16}$ for $(CS \text{ modulo } 16) \leq 9$
$= (CS \text{ modulo } 16) + 37_{16}$ for $(CS \text{ modulo } 16) \geq 10$

Example: MOR-Protocol

STX,'>',ETX,'3E',CR
$02\ 3E\ 03\ 33\ 45\ 0D_{16}$ with $CS = 62 = 3E_{16}$

Further examples of ASCII-representation:
MOR-Protocol: ☺>♥3E REP-Protocol: ☺?♥3F SPM-Protocol: ☺<♥3C

12.8.3 Automatic adaption to the test procedure used by the host

Miditron® *Junior II* is in a position to adapt itself automatically to the test procedure used by the host. If an error is discovered when checking the test bytes of the receive protocol, the check will be continued using the algorithms of the alternative procedure. If, using the alternative test procedure, the test bytes are recognized as correct, this procedure will be used for all protocols in future. This new setting is retained even after the appliance has been switched off. If the alternative procedure does not recognize any correct test bytes either, an REP protocol is transmitted and there is no change of test procedure.

If, upon startup (i.e. the first time **Miditron**® *Junior II* is connected up to the host) , two different procedures are set, the host must reply to the SPM from **Miditron**® *Junior II* with an REP or MOR and the relevant test bytes.
Example:

Miditron® *Junior II*		**Host**	**Remarks:**
	/SPM/ ——————>		(with LRC)
	<—————— /MOR/		(with check total)
	/SPE/+data ———>		(with check total)
	Hencefort, check-total permanently set as the test procedure in **Miditron**® *Junior II*		
	<—————— /MOR/		(with LRC)
or	<—————— /REP/		(witht LRC)
	/SPE/+data ———>		(with LRC)
	Hencefort, LRC permanently set as the test procedure in **Miditron**® *Junior II*		
	<—————— /MOR/		(with check total)
or	<—————— /REP/		(with check total)
	/SPE/+data ———>		(with check total)
	Henceforth, check-total permanently set as the test procedure in **Miditron**® *Junior II*		

etc.