OneTouch® UltraMini™ / UltraEasy™ Blood Glucose Meter RS-232 Communication Protocol

Software Developer

Please note that this protocol is not intended to be a substitute for a complete data management software product. We are providing this protocol to you with the understanding that you are very familiar with computers and software development, and will be able to use the information appropriately. This protocol has been reviewed, but it is expected that you will formally test and validate the use of this information with your software product. LifeScan will not be liable for any damages whatsoever.

Note that although this protocol refers to two products (UltraMini™ and UltraEasy™), the products are, in terms of their data communications, functionally identical. Hence where the text refers to "the meter", the text is applicable to both meters.

NO WARRANTY, EXPRESS OR IMPLIED, IS PROVIDED TO YOU WITH RESPECT TO THE ONETOUCH ULTRAMINI METER RS-232 COMMUNICATION SPECIFICATION CONTAINED HEREIN (THE “SPECIFICATION”) AND LIFESCAN SPECIFICALLY DISCLAIMS, WITHOUT LIMITATION, ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. THIS SPECIFICATION IS PROVIDED TO YOU STRICTLY ON AN “AS IS” BASIS. THE ENTIRE RISK OF THE QUALITY, APPLICABILITY AND PERFORMANCE OF THE SPECIFICATION AND ANY SOFTWARE DEVELOPED BY OR FOR YOU IS WITH YOU. NO LIFESCAN DEALER, AGENT OR EMPLOYEE, OR ANY OTHER PARTY, IS AUTHORIZED TO MAKE WARRANTIES OR CONDITIONS ON LIFESCAN'S BEHALF. TO THE MAXIMUM EXTENT PERMITTED BY LAW, UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY, CONTRACT, PATENT INFRINGEMENT, COPYRIGHT INFRINGEMENT OR OTHERWISE, WILL LIFESCAN BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, RELIANCE OR CONSEQUENTIAL DAMAGES, EVEN IF LIFESCAN HAS BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES AND REGARDLESS OF WHETHER ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE.

OneTouch® UltraMini™ Meter RS-232 Communication Protocol

The following information may be used when attempting to **upload** the OneTouch® UltraMini™ Meter memory to a computer with the OneTouch® Interface Cable.

EQUIPMENT NEEDED

Meter: OneTouch® UltraMini™ / UltraEasy™

Cable: OneTouch® Interface Cable (25-pin, 9-pin or USB)

Computer: IBM® compatible personal computer

Adapter: An adaptor may be required depending on the computer and version of the OneTouch® Interface Cable. For Example: IBM® compatible personal computer: A 25-pin to 9-pin adapter if serial/com port is a 9-pin and the interface cable is a 25-pin cable.

Cable: Connect OneTouch® Interface cable to an available serial or USB port on the computer. Insert the OneTouch® Interface cable stereo plug into the data port that is located at the bottom of the meter.

Software: A communications software package, such as HyperTerminal.
Select port settings in communications software:

Baud Rate = 9600 bps	Data Bits = 8
Stop Bits = 1	Parity = none
Flow Control = None	Com Port = port # utilized

Time-out Information:

- The inter-character timeout period for the Link Layer Protocol is 10msec and the inter packet timeout period is 100mSec.
- Link Layer Timeout – this shall be 0.5 seconds.

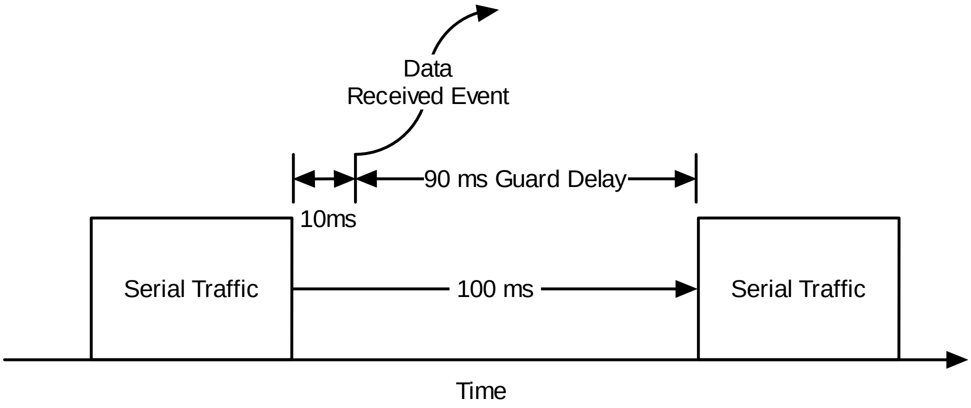
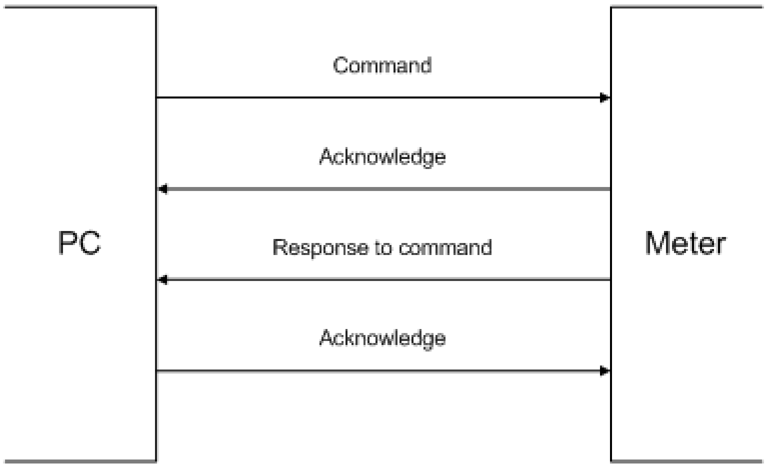


Figure 1 Inter-character timeout timing

Initiating Communications:

Initiate the terminal screen of your communications software package. Leave the meter powered **OFF**.

Communicating with the meter follows the following process flow:



Data Types:

Decimal, hexadecimal or binary numbers are used in this document.

- Decimal numbers are used to represent counts.
- Hexadecimal numbers are used to represent values like commands IDs and addresses.
- Binary numbers are used to describe bit patterns or bit settings within binary fields.
- Within a byte bits are numbered from 0 to 7, with 0 being the least significant bit

Commands:

The meter supports the following commands -

- Read Software Version String and Software Creation Date
- Read Serial Number
- Delete All Glucose Records
- Read Glucose Record
- Read Current Unit Settings
- Read Date Format
- Read/Write RTC

For each of these commands and their associated responses from the meter, the command frame layout is:

Start of message indicator	Length Byte	Control Byte	Data portion	End of message indicator	Check characters
STX	Size of packet	Link control information	Application data	ETX	CRC_{low} CRC_{high}

Elements of the command frame:

Element	Contents
Start of message indicator	The character STX (0x02)
Length Byte	A byte containing the number of bytes which make up the complete frame, from the STX character to the CRC_{high} character inclusive.
Link Control Byte	A byte containing sequencing number information, ack/nack and disconnection indications. Described below in the section **Link Control Byte**
Data portion	Up to 34 bytes of application specific data.
End of message indicator	The character ETX (0x03)
Check characters	These characters contain the CCITT-CRC16 of the frame. The CRC-16 is calculated on all the fields except the check characters. This is detailed in Appendix – CRC Calculation.

Link Control Byte:

The link control byte has the following format.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Unused	Unused	Unused	More	Disconnect	Acknowledge	E	S

Bit 4: More
When set this indicates that the sending party has a subsequent data frame to send. The more indication will be passed to the application. It is the application's responsibility to interpret the actions to be taken on the receipt of a more indication. This bit is only valid when the frame contains application data. Its setting is meaningless if the Disconnect bit is set.

Bit 3: Disconnect
When set this indicates that the sending party is requesting to terminate the data link. The receiving party must respond to this request by issuing a disconnect response. A full description of the disconnection procedure is provided in the next section.

Bit 2: Acknowledge
This indicates that the message packet is a link level acknowledgement packet that contains no application data. This packet is used to confirm the correct reception of a data message.

NOTE - This bit must be set to 0 when transmitting a message which contains application data, or when the disconnect bit is not set.

Bit 1: E
In order to maintain correct packet ordering and identify retries, each communicating party maintains an "Expected Receive" (E) sequence number, and a "Send" (S) sequence number. Bit 1 is used to hold the "Expected Receive" number of the sending party. On establishment of the data link this field will be set to 0. When a correctly framed message packet with a "Send" sequence number of 0 is received, this value will be set to 1. When a subsequent message packet with a "Send" sequence number of 1 is received this value will be set to 0. These alternations of the expected receive bit proceeds for the duration of the data connection.

NOTE - The value of this field is only changed on successful reception of a message packet containing application data. For this to happen, the packet must be correctly framed and check summed, with the S number in the packet being equal to the "Expected Receive" number held by the receiving party. This field is not altered by the reception of link level control packets such as acknowledgement and disconnect. It may also used when determining whether a packet containing application data is providing an implicit acknowledgement to a previously sent packet.

Bit 0: S
This field holds the “Send” sequence number field from the sending party as described for Bit 1 above. This number is incremented when the link level receives an acknowledgement message from the receiving party. The initial value of this field is 0, when an acknowledgement is received it will be set to 1, a subsequent successful message transmission will result in the send sequence number being set to 0. These alternations of the send number proceeds for the duration of the data connection.

NOTE - The value of this field is only changed on the successful acknowledgement of the previously sent packet. For this to happen, the received packet must be correctly framed and check summed, with the E number in the packet being different from the “Send” sequence number of the receiving party

Link Level acknowledgement:

A link level acknowledgement packet is sent in response to a valid message containing application data. The example below shows the format of an acknowledgement message that would be sent after reception of a message which had a send sequence number of 0, and contained application data. Note that the Acknowledge bit is set and the expected receive bit in the control byte is set to 1.

Start of message indicator	Length	Control Byte	End of message indicator	Check-characters
STX	0x06	00000110	ETX	CRC$_{low}$ CRC$_{high}$

It is possible for link layer acknowledgements to be lost, in which case out of sequence packets can be received. In this case, the data packet should be acknowledged by sending the link layer ACK, however the data message should not be passed to the application. Figure 3 illustrates this sequence.

Link Level Timeout:

The link level timeout is used by the sender to recover from transmission failures of application data frames and disconnect messages. There is no failure recovery implemented for acknowledgement messages.

The sender of the frame sets its transmission counter to 1 and starts the timer when the last byte of the frame is sent. If the timer expires before a response is received the sender interprets this as an error and increments its transmission counter. If the transmission counter exceeds 3 then the link layer discards the message and reports to the application that it failed to send. It is up to the application to take the appropriate recovery action.

If the sender receives a link level acknowledgement packet OR a message packet whose control byte acknowledges the message has been received, then the sender informs its local application of the successful transmission of the message.

Link Level Disconnection:

In case of serious line error, or when the dialogue is completed, either party can issue a link-level disconnect request. Disconnect requests have no attached data. After sending the disconnect request, it waits for the receiving party to issue a disconnect response. The format is as follows:

Start of message indicator	Length	Control Byte	End of message indicator	Check-characters
STX	0x06	000010ES	ETX	CRC$_{low}$ CRC$_{high}$

After sending the disconnect, the originator waits for the other party to acknowledge by issuing a link level disconnect response of the following format.

Start of message indicator	Length	Control Byte	End of message indicator	Check-characters
STX	0x06	000011ES	ETX	CRC$_{low}$ CRC$_{high}$

Where

E : Indicates the sending parties expected receive sequence number.
S : Indicates the sending parties current send number.

The link level terminates the data connection as soon as it receives a disconnect response.

Commands:

To start a session it is recommended that an initial disconnect command is sent to get the meter into a known state. This is achieved by issuing the command listed in 0 above.

An example of the data exchange is shown below:

Command from PC: Disconnect

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x08	0x03	0xC2	0x62

Reply from Meter: Disconnected and Acknowledged

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x0C	0x03	0x06	0xAE

Command: Read Software Version String and Software Creation Date:

This command allows the PC to read the meters software version string and that version's assigned creation date.

Below is an example communication between the PC and Meter (following a disconnect/acknowledge between the PC and Meter):

Command Message from PC: Read Software Version String and Software Creation Date

STX	Len	Link	CM1	CM2	CM3	ETX	CRC low	CRC high
0x02	0x09	0x00	0x05	0x0D	0x02	0x03	0xDA	0x71

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x06	0x03	0xCD	0x41

Reply Message 2 from Meter: S/W Version String and Creation Date: “P02.00.0025/05/07”

STX	Len	Link	RM1	RM2	RM3	‘P’	‘0’	‘2’	‘.’	‘0’	‘0’	‘.’
0x02	0x1A	0x02	0x05	0x06	0x11	0x50	0x30	0x32	0x2E	0x30	0x30	0x2E

‘0’	‘0’	‘2’	‘5’	‘/’	‘0’	‘5’	‘/’	‘0’	‘7’	ETX	CRC	CRC
											low	high
0x30	0x30	0x32	0x35	0x2F	0x30	0x35	0x2F	0x30	0x37	0x03	0xAB	0x25

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x07	0x03	0xFC	0x72

Command: Read Serial Number

This command allows the PC to read the meters serial number.

Below is an example communication between the PC and Meter (following a disconnect/acknowledge between the PC and Meter):

Command Message from PC: Read Serial Number

STX	Len	Link	CM1	CM2	CM3	CM4	CM5	CM6	CM7	CM8	CM9	CM10
0x02	0x12	0x00	0x05	0x0B	0x02	0x00	0x00	0x00	0x00	0x84	0x6A	0xE8

CM11	CM12	ETX	CRC	CRC
			low	high
0x73	0x00	0x03	0x9B	0xEA

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x06	0x03	0xCD	0x41

Reply Message 2 from Meter: Serial Number = “C176SA0O0”

STX	Len	Link	CM1	CM2	‘C’	‘1’	‘7’	‘6’	‘S’	‘A’	‘0’	‘O’
0x02	0x11	0x02	0x05	0x06	0x43	0x31	0x37	0x36	0x53	0x41	0x30	0x4F

‘0’	ETX	CRC	CRC
		low	high
0x30	0x03	0x49	0x43

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x07	0x03	0xFC	0x72

Command: Delete All Glucose Records

This command deletes all glucose records in the meter.

Below is an example communication between the PC and Meter (following a disconnect/acknowledge between the PC and Meter):

Command Message from PC: Delete All Glucose Records

STX	Len	Link	CM1	CM2	ETX	CRC	CRC
						low	high
0x02	0x08	0x00	0x05	0x1A	0x03	0x56	0xB0

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x06	0x03	0xCD	0x41

Reply Message 2 from Meter: Command Executed

STX	Len	Link	RM1	RM2	ETX	CRC	CRC
						low	high
0x02	0x08	0x02	0x05	0x06	0x03	0x20	0x1B

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x07	0x03	0xFC	0x72

Command: Read Glucose Record

This command allows the PC to read a specified record. Records are indexed from 0 to 499. Record 0 is the most recent glucose record recorded.

To use this facility to read one or more records successfully, the PC must first establish how many records are present in the meter. This is achieved by requesting to read record 501 which is an invalid request but the meter will reply with the number of records that are available.

Below is an example communication between the PC and Meter (following a disconnect/acknowledge between the PC and Meter) where the meter contains 3 records with the values and time stamps:

1. 76 mg/dL 16:05 20 June 2025
2. 89 mg/dL 10:50 26 April 2012
3. 79 mg/dL 16:30 25 Dec 2007

First the PC requests the number of records available by requesting an invalid record (501):

Command Message from PC: Read Glucose Record 501

STX	Len	Link	CM1	CM2	501	ETX	CRC	CRC
							low	high
0x02	0x0A	0x00	0x05	0x1F	0xF5, 0x01	0x03	0x38	0xAA

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x06	0x03	0xCD	0x41

Reply Message 2 from Meter: Invalid Record + Number of Records = 3

STX	Len	Link	RM1	RM2	Number Of Records	ETX	CRC	CRC
							low	high
0x02	0x0A	0x02	0x05	0x0F	0x03, 0x00	0x03	0x1C	0x58

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x07	0x03	0xFC	0x72

The PC has now established there are 3 records available on the meter. It now requests to read all 3 records:

READ RECORD 1

Command Message from PC: Read Glucose Record 1 (offset = 000)

STX	Len	Link	CM1	CM2	Record 1	ETX	CRC	CRC
							low	high
0x02	0x0A	0x03	0x05	0x1F	0x00, 0x00	0x03	0x4B	0x5F

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x05	0x03	0x9E	0x14

Reply Message 2 from Meter: Record glucose value + date stamp

STX	Len	Link	RM1	RM2	DT1	DT2	DT3	DT4	GR1	GR2	GR3	GR4
0x02	0x10	0x01	0x05	0x06	0xAC	0x86	0x55	0x68	0x4C	0x00	0x00	0x00

ETX	CRC	CRC
	Low	high
0x03	0x86	0x0B

This result can be interpreted as follows:

- date and time in hex is 685586AC (from DT4 to DT1) = 16:05 20 June 2025
- glucose value in hex is 4C(from GR4 to GR1) = 76 in decimal.

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x04	0x03	0xAF	0x27

READ RECORD 2

Command Message from PC: Read Glucose Record 2 (offset = 001)

STX	Len	Link	CM1	CM2	Record 2	ETX	CRC	CRC

							low	high
0x02	0x0A	0x00	0x05	0x1F	0x01, 0x00	0x03	0x9B	0xA6

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x06	0x03	0xCD	0x41

Reply Message 2 from Meter: Record glucose value + date stamp

STX	Len	Link	RM1	RM2	DT1	DT2	DT3	DT4	GR1	GR2	GR3	GR4
0x02	0x10	0x02	0x05	0x06	0x58	0x28	0x99	0x4F	0x59	0x00	0x00	0x00

ETX	CRC low	CRC High
0x03	0x5D	0x60

This result can be interpreted as follows:

- date and time in hex is 4F992858 (from DT4 to DT1) = 10:50 26 April 2012
- glucose value in hex is 59 (from GR4 to GR1) = 89 in decimal.

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x07	0x03	0xFC	0x72

READ RECORD 3

Command Message from PC: Read Glucose Record 3 (offset = 002)

STX	Len	Link	CM1	CM2	Record 3	ETX	CRC low	CRC high
0x02	0x0A	0x03	0x05	0x1F	0x02, 0x00	0x03	0x2B	0x31

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x05	0x03	0x9E	0x14

Reply Message 2 from Meter: Record glucose value + date stamp

STX	Len	Link	RM1	RM2	DT1	DT2	DT3	DT4	GR1	GR2	GR3	GR4
0x02	0x10	0x01	0x05	0x06	0x08	0x30	0x71	0x47	0x4F	0x00	0x00	0x00

ETX	CRC low	CRC high
0x03	0x58	0x05

This result can be interpreted as follows:

- date and time in hex is 47713008 (from DT4 to DT1) = 16:30 25/December/2007
- glucose value in hex is 4F (from GR4 to GR1) = 79 in decimal.

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x04	0x03	0xAF	0x27

Command: Read Current Unit Settings

This command allows the PC to read the current unit settings (mg/dL or mmoL).

Below is an example communication between the PC and Meter (following a disconnect/acknowledge between the PC and Meter):

Command Message from PC: Read Current Unit Settings

STX	Len	Link	CM1	CM2	CM3	CM4	PM1	PM2	PM3	PM4	ETX
0x02	0x0E	0x00	0x05	0x09	0x02	0x09	0x00	0x00	0x00	0x00	0x03

CRC low	CRC high
0xCE	0xE7

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x06	0x03	0xCD	0x41

Reply Message 2 from Meter: Current Unit Settings

STX	Len	Link	RM1	RM2	PM1	PM2	PM3	PM4	ETX	CRC low	CRC high
0x02	0x0C	0x02	0x05	0x06	0x00	0x00	0x00	0x00	0x03	0x20	0xC1

This result is interpreted as follows: PM1 is 0 => unit setting is mg/dL.
If PM1 = 1, then unit setting is mmoL.

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x07	0x03	0xFC	0x72

Command: Read Date Format

This command allows the PC to read the date format (EU or US).

Below is an example communication between the PC and Meter (following a disconnect/acknowledge between the PC and Meter):

Command Message from PC: Read Date Format

STX	Len	Link	CM1	CM2	CM3	CM4	PM1	PM2	PM3	PM4	ETX
0x02	0x0E	0x00	0x05	0x08	0x02	0x00	0x00	0x00	0x00	0x00	0x03

CRC low	CRC high
0xFF	0xE8

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x06	0x03	0xCD	0x41

Reply Message 2 from Meter: Date Format

STX	Len	Link	RM1	RM2	PM1	PM2	PM3	PM4	ETX	CRC	CRC
										low	high
0x02	0x0C	0x02	0x05	0x06	0x01	0x00	0x00	0x00	0x03	0x71	0x6B

This result is interpreted as follows: PM1 is 1 which indicates EU date format.
If PM1 = 0, then US date format.

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x07	0x03	0xFC	0x72

Command: Read/Write RTC

This command allows the PC to read and write the RTC of the meter.

Below is an example communication between the PC and Meter (following a disconnect/acknowledge between the PC and Meter) where the PC reads the RTC and then writes a new RTC value to the meter:

Command Message from PC: Read RTC

STX	Len	Link	CM1	CM2	CM3	DT1	DT2	DT3	DT4	ETX	CRC	CRC
											low	high
0x02	0x0D	0x00	0x05	0x20	0x02	0x00	0x00	0x00	0x00	0x03	0xEC	0x61

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x06	0x03	0xCD	0x41

Reply Message 2 from Meter: RTC Current Setting

STX	Len	Link	RM1	RM2	DT1	DT2	DT3	DT4	ETX	CRC	CRC
										low	high
0x02	0x0C	0x02	0x05	0x06	0x83	0XA4	0xFF	0x41	0x03	0x3B	0xDC

This result is interpreted as follows:
- date and time in hex is 41FFA483 (from DT4 to DT1) = 12:34:56 on 01/Feb/2005

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC	CRC
				low	high
0x02	0x06	0x07	0x03	0xFC	0x72

Command Message from PC: Write RTC = 12:34:56 29 Feb 2008

STX	Len	Link	CM1	CM2	CM3	DT1	DT2	DT3	DT4	ETX	CRC	CRC
											low	high
0x02	0x0D	0x03	0x05	0x20	0x01	0xE0	0xED	0xC7	0x47	0x03	0x14	0x33

This command is interpreted as follows:
- date and time in hex is 47C7EDE0 (from DT4 to DT1) = 12:34:56 29/Feb/2008

Reply Message 1 from Meter: Acknowledge

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x05	0x03	0x9E	0x14

Reply Message 2 from Meter: RTC Current Setting

STX	Len	Link	RM1	RM2	DT1	DT2	DT3	DT4	ETX	CRC low	CRC high
0x02	0x0C	0x01	0x05	0x06	0xE0	0xED	0xC7	0x47	0x03	0x09	0xB8

Reply from PC: Acknowledge

STX	Len	Link	ETX	CRC low	CRC high
0x02	0x06	0x04	0x03	0xAF	0x27

Voice Module Serial Output

The meter also supports serial communications output intended for use by voice modules. The serial format used is as specified above in the section **Software**

The output delivered to the serial port by the meter directly following a completed test is illustrated in the following output strings:

Example: glucose value 21.4 mmoL

The output string delivered for this value is shown below in ASCII and hex:

0	,	“	2	1	.	4	“	,	1	,	“	“
0x30	0x2C	0x22	0x32	0x31	0x2E	0x34	0x22	0x2C	0x31	0x2C	0x22	0x22

,	“		“
0x2C	0x22	0x20	0x22

The tenth character (‘1’) indicates the value delivered is in mmol/L.
If this is set to ‘2’, then the value delivered is in mg/dL

Example: glucose value 72 mg/dl

The output string delivered for this value is shown below in ASCII and hex:

0	,	“	7	2	“	,	2	,	“	“
0x30	0x2C	0x22	0x37	0x32	0x22	0x2C	0x32	0x2C	0x22	0x22

,	“		“
0x2C	0x22	0x20	0x22

Example: glucose value HI (glucose value > HIGLX)

The output string delivered for this value is shown below in ASCII and hex:

0	,	“	H	I	“	,	1	,	“	“
0x30	0x2C	0x22	0x48	0x49	0x22	0x2C	0x31	0x2C	0x22	0x22

,	“		“
0x2C	0x22	0x20	0x22

Example: glucose value LO (glucose value < LOGLX)

The output string delivered for this value is shown below in ASCII and hex:

0	,	“	L	O	“	,	2	,	“	“
0x30	0x2C	0x22	0x4C	0x4F	0x22	0x2C	0x32	0x2C	0x22	0x22

,	“		“
0x2C	0x22	0x20	0x22

1. APPENDIX

Calculating the CRC:

The CCITT-CRC16 is employed: $X^{16} + X^{12} + X^5 + X^1$ where the algorithm seed is 0xffff and input is the string to be transmitted from the first character up to but not including the CRC bytes.

The following C function will return the CRC when: initial_crc set to 0xffff, the buffer pointer set to the start of the string to be transmitted and the length set to the number of bytes in the string not including the two bytes for the CRC.

```
unsigned short crc_calculate_crc (unsigned short initial_crc, const unsigned char *buffer, unsigned short length)
{
    unsigned short index = 0;
    unsigned short crc = initial_crc;

    if (buffer != NULL)
    {
        for (index = 0; index < length; index++)
        {
            crc = (unsigned short)((unsigned char)(crc >> 8) | (unsigned short)(crc << 8));
            crc ^= buffer [index];
            crc ^= (unsigned char)(crc & 0xff) >> 4;
            crc ^= (unsigned short)((unsigned short)(crc << 8) << 4);
            crc ^= (unsigned short)((unsigned short)((crc & 0xff) << 4) << 1);
        }
    }

    return (crc);
}
```

A test case of this is shown below:

Given the array:

```
unsigned char test_crc[4] = {02,06,06,03};
```

The function call:

```
unsigned short crc = crc_calculate_crc( 0xffff, test_crc, 4 );
```

yields the resultant CRC of 0x41CD.