boso
BOSCH + SOHN
GERMANY

corscience
cardiovascular innovations

Communication Protocol

Boso Medicus Prestige + BT

These instructions were drawn up with the utmost care. If you find any inaccuracies in the information presented here, we would appreciate it if you let us know so that we can eliminate any discrepancies as quickly as possible.

The information and figures in this user manual are subject to change due to further optical or technical developments.

All trademarks named and shown in the text are trademarks of the respective registrant and are acknowledged as copyright protected.

Reprinting, translation and duplication in any form – even in excerpt form – require the written permission of the manufacturer. This handbook is subject to change by Corscience GmbH & Co. KG.

CS60283C – 12.2011

corscience
cardiovascular innovations

Corscience GmbH & Co. KG
Henkestraße 91
91052 Erlangen
Germany

Phone: +49 9131 977986-0
Fax: +49 9131 977986-59
Email: info@corscience.de
Website: www.corscience.de

TABLE OF CONTENTS

1. INTRODUCTION

In the project COR1043 the serial Corscience protocol is applied for the communication between the blood pressure meter and a receiver-unit (e.g. PC or a mobile phone).

The blood pressure meter is a standard blood pressure device for ambulatory use. It is additionally equipped with a Bluetooth module for wireless data transmission. The Bluetooth module can hold up to 9 measurement values until the oldest one gets overwritten. The measurement values can be retrieved via different modes.

The protocol is designed for simple and memory-saving implementation in a microcontroller. The overhead was kept as low as possible. The principle of this protocol is basically modelled on the PPP (point-to-point protocol), which is often used to establish modem connections. Escape sequences are used to filter out reserved bytes (start flag, end flag, escape flag) from the data stream (so-called octet stuffing). The octet stuffing in the transmitter occurs after checksum calculation. For the checksum formation in the transmitter, the start flag and end flag are not included. In the receiver, inverse octet stuffing is applied first, before the data stream is saved or further processed. In the receiver, the checksum algorithm is run over the entire packet (without the Start flag and End flag). The result must then be 0.

In case a 16 Bit structure is used at the microcontroller all words are transmitted LSB-First.

1.1 Modes of Transmission

The blood pressure meter can be used in different wireless networks with different kinds of Bluetooth transmission.

For the wireless transmission the Bluetooth-RFCOMM interface is applied. It is possible to configure the blood pressure meter to operate in 3 different modes.

- “Active Data” Mode
  After a measurement the blood pressure meter can connect itself automatically to the receiver unit
- “Passive Data” Mode
  The blood pressure meter is connectable from outside to transmit the new value.
- “Service” Mode
  It is also possible to connect a PC to the blood pressure meter to configure the device.

The bluetooth device name of the blood pressure meter is "boso medicus BT". The service name is "SPPBP BT".

1.1.1 „Active Data“ Mode

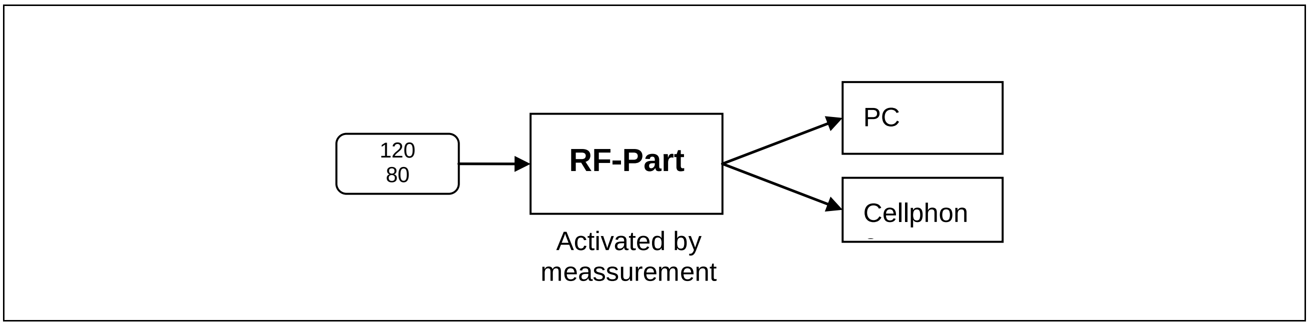

Figure 1: Diagram of Active Data Mode

After a new measurement is done, the bt connection will be established by the blood pressure (bp) device actively. The bp devive works as BT master. Then the bp-result will be transmitted to the remote device (PC or cellphone). After data transmission the bp device closes the BT connection. Any commands from the application on the remote device, exept 0x0200 ACK and 0x00XX commands, will be ignored by the bp device.

It is possible to choose the Bluetooth dial-up-network (DUN) for sendig the measured value. Then it will be send to a remote device (e.g. cellphone) using standard AT-commands. The data is written in a text mode SMS. The format of the SMS is described in appendix . The remote device receives the SMS via Bluetooth and automatically sends it to a configurable phone number via GSM-network.

1.1.2 „Passive Data“ Mode

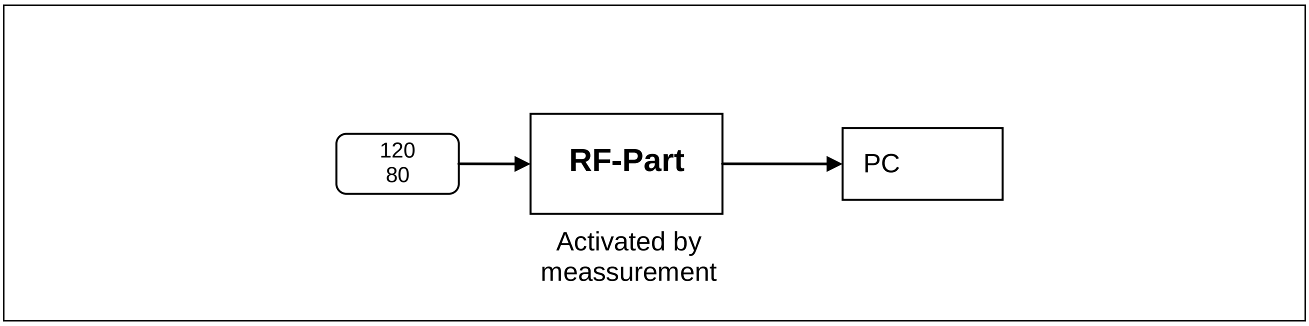

Figure 2: Diagram of Passive Data Mode

This mode is the default mode of the bp device. After pressing the START button, the bp device is connectable for other BT devices in range. The bp device keeps connectable for 5 minutes. During this time a remote device (e.g. PC) can connect to the bp device (BT slave).

The blood pressure meter offers a serial port profile (SPP) service.

When a BT connection to this service is established the remote device only can request for the mesurement result(s) of the bp device using the command 0x0800 with payload 0x0706 (command “Transmit blood pressure data”). All other commands, exept 0x0200 ACK and 0x00XX commands, will be ignored.

There is a timeout of 30 minutes after the last received command, this will close the connection, if it is not done by the remote device.

1.1.3 “Service” Mode

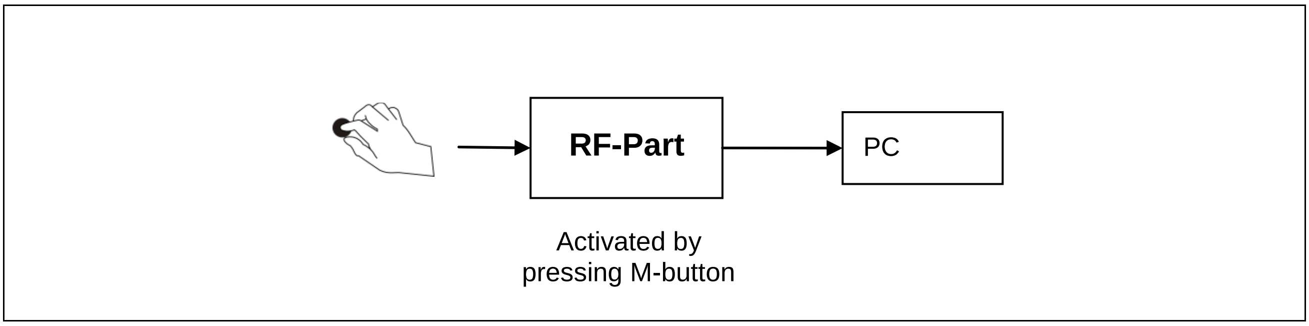

Figure 3: Diagram of Service Mode

To start the device in “Service” mode, press the “M” button for approx. 10 seconds after a measurement is done. You can also start and stop the device by pressing “START” twice and then pressing the “M” button for approx. 10 seconds. The “medicus prestige” will acknowledge with one long beep.
For safety reasons it is not possible to enter the service mode without pressing “START” before.

In this mode the bp device is connectable for other BT devices in range. The bp device keeps connectable for 5 minutes. During this time a remote device (e.g. PC) can connect to the bp device (BT slave). The blood pressure meter offers a serial port profile (SPP) service.

When a BT connection to this service is established the Corscience protocol is used for communication and configuration. The commands that can be used are described in chapter 5 “implemented commands”.

There is a timeout of 30 minutes after the last received command, this will close the connection, if it is not done by the remote device.

To leave the service mode please use the shutdown command (0x00FF). All changes made in service mode are active then. The device will leave the service mode automatically after 5 minutes without a connection or 30 minutes after the last received command. It will not leave the service mode by pressing the “START” button. We recommend to leave the service mode by shutdown command.

1.2 Bluetooth Pairing

When the blood pressure meter is switched on the first time, it is in the Bluetooth “passive data” mode. As long as no remote device connects to the bp device, it keeps visible for other BT devices in range, If a device opens the BT connection and authenticates correctly the bp device is not longer visible for other BT devices in range. The bp device can be set visible by pressing the “M”-button longer than 15 seconds and start the bp device again. The blood pressure meter will signal the 15 seconds through 3 long beep tones.

If the Blood pressure meter is set to “active data” mode, it will first unpair itself. After the first measurement in this mode, it searches the surrounding for Bluetooth devices. The found devices are ordered by priority of their device names. If they belong to a predefined group of devices they are prefered when the blood pressure meter tries to get the authorisation to connect to one of the devices. The devices of the prefered group can be programmed.

If no predefined device is found the measured value is stored and the blood pressure meter stays unpaired. If a new blood pressure value will be measured the pairing will start again.

Once a pairing has been sucessfull the device is remembered. For the next communication only this device is searched and a connection will only be established to this paired device.

The blood pressure meter could be unpaired by pressing the “M”-button longer than 15 seconds. If an unpair is done the blood pressure meter will signal this through 3 long beep tones.

The predefined BT Pin is 1609.

2. PACKET STRUCTURE

Each packet is structured as shown in the table below:

start flag	packet-number	command	payload	CRC	end flag
0xFC	1 byte	2 bytes	x bytes	2 bytes	0xFD

2.1 Description of Individual Fields

The individual fields of the above overview are described in more detail below.

2.1.1 Start Flag

The start flag, which has a length of one byte and the fixed value **0xFC** is the unmistakable start character of a packet. This character may not appear again in the rest of the packet to avoid confusion. To prevent this from happening, octet stuffing (see 4.) is used.

2.1.2 Packet Number

The packet number field has a length of one byte and can therefore sweep through numbers from 0 to 255. The packets are consecutively numbered, separately from both sides. Once 255 has been reached, the next packet restarts with 0. The numbers are given separately from both sides and need not be identical. The packets are numbered so that missing packets can be detected, received packets can be confirmed, undesired packets can be rejected and defective packets can be resent.

2.1.3 Command

The **command** field is two bytes long and contains the command (LSB first).

2.1.4 Payload

The actual data to be transmitted. This field can contain other fields, depending on the command. The fields have a defined function within the command.

2.1.5 Checksum

The checksum has a length of two bytes. The parameters of the checksum are:

CRC16 (CCITT), polynomial 0x1021, start value 0xFFFF, LSB first.

The checksum is calculated in the **transmitter** over the entire packet with the exception of the **start flag**, e**nd flag** and **checksum**. The checksum must be calculated in the transmitter before octet stuffing.

In the **receiver**, the checksum is calculated over the entire packet with the exception of the **start-flag** and the **end flag**. For a valid packet, the result must be **0**. The checksum test must be applied after octet stuffing at the sender and before octet destuffing at the receiver.

2.1.6 End Flag

The ***end flag*** has a length of one byte and has the fixed value of **0xFD**, which is the unmistakable end character of a packet. To avoid confusion, this character may not appear again in the rest of the packet. To prevent this from happening, so-called octet stuffing is applied.

3. OCTET STUFFING

There are three special characters which may never appear in the packet:

- Start flag 0xFC
- End flag 0xFD
- Escape flag 0xFE

These three characters must be filtered out of the data stream before sending the packet and must be treated specially. If one of these characters appears in the data stream, an escape flag is sent first, and then the original byte is linked with 0x20 EXOR.

Characters within the packet:	Are changed to the following when sent:
0xFC	0xFE 0xDC
0xFD	0xFE 0xDD
0xFE	0xFE 0xDE

When an escape sequence is received by the receiver, this byte isn’t saved, but the following byte is linked with 0x20 EXOR and saved.

The receiver decodes the escape sequence again:

Received characters:	Decoded:
0xFE 0xDC	0xFC
0xFE 0xDD	0xFD
0xFE 0xDE	0xFE

4. IMPLEMENTED COMMANDS

Command	**Name of command**	**Direction** **bp - remote** **device - device**	**Description**
0x0000	close connection	←	The connection is closed
0x0001	ping	←	Communication test
0x00FF	shutdown	←	Shutdown the bp device
0x0100	protocol	→	Protocol version is given. Furthermore, the buffer sizes are transmitted. The sender gives the maximum length of a packet (in bytes), which the sender can receive. Additionally the amount of packets, which – in case of a NAK-answer – can be buffered, is.
0x0150	firmware version	→	Software version of the implemented software (article number)
0x0151	MAC address	→	Returns the Bluetooth device address of the device
0x0200	acknowledge	→ ←	Single packet with distinct packet number is acknowledged as correct
0x0300	not acknowledge	→ ←	Single packet with distinct packet number is not acknowledged (checksum error). This is a demand for retransmission.
0x0400	reject	→ ←	Single packet with distinct packet number is not acknowledged and rejected (e.g. undefined command or parameter)
0x0500	identification	→	Producer-ID, Device-ID and serial number are transmitted
0x0501	software version	→	Software version of the implemented software
0x0600	self test	→	Results of the self test
0x0601	config BP	←	Configuration of the blood pressure device
0x0602	config new remote device	←	Configure a new remote device
0x0603	config Bluetooth pin	←	Configure a new Bluetooth security pin
0x0631	buzzer sequence	←	Starts a buzzer sequence of 4 short beep tones
0x0632	config buzzer	←	Command to enable and disable the buzzer signals
0x0706	transmit blood pressure data	→	The measured values are transmitted in this command
0x0800	request	←	With this command another command can be requested

The following paragraphs describe the commands in detail. Also the content of the payload bytes are defined.

4.1 0x00XX commands

The 00XX commands can be sent every time.

0x0000 close connection (no payload)

Start	Packet-Nr.	Command 2 byte		Checksum 2 byte		End
FC	XX	00	00	XX	XX	FD

This command causes a Bluetooth disconnect.

0x0001 ping (no payload)

Start	Packet-Nr.	Command 2 byte		Checksum 2 byte		End
FC	XX	01	00	XX	XX	FD

The command ping is answered with an Acknowledge (see 4.3) and is used to test the connection. It can be sent every time especially before identification.

0x00FF shutdown (no payload)

Start	Packet-Nr.	Command 2 byte		Checksum 2 byte		End
FC	XX	FF	00	XX	XX	FD

If a shutdown is sent to the blood pressure meter the Bluetooth module will shut down and will not restart until the O/I button of the blood pressure meter is pressed again.

4.2 0x01XX commands

0x0100 protocol (Payload 4 byte)

Start	Packet-Nr.	Command 2 byte		Protocol version	Max.length*) 2 byte		Max. packets**)	Checksum 2 byte		End
FC	XX	00	01	02	00	28	01	XX	XX	FD

*) max. length of a packet which can be received in bytes (here 40byte = hex 28)

**) max. no. of packets, which can be buffered

The max. no. of packets, which can be buffered, is set to 0x01 (in this project COR1043). Therefore, each packet must be acknowledged (see 4.3), before a new packet can be sent (**Stop-and-Wait technique**).

0x0150 firmware version (Payload 14 byte)

Start	**Packet-Nr.**	**Command 2 byte**		**Firmware 7 byte**							**Derivative 3 byte**			**Ver-sion**	**Development version 3 byte**			**Check-sum 2 byte**		**end**
FC	XX	50	01	43	53	31	30	31	31	37	2D	30	34	XX	XX	XX	XX	XX	XX	FD

The firmware version is saved as a string. It includes the name of the firmware, the derivative, the version and the development version. If there is not derivative or the software is not a development version any longer, the parts are left free with empty space. If the development version is not transmitted the software is a released version.

The payload is written in Ascii.

Example:	
Firmware:	CS10117
Derivative:	-04
Version:	A
Dev. Version:	E01
“CS10117-04AE01”	FC XX 50 01 43 53 31 30 31 31 37 2D 30 34 41 45 30 31 XX XX FD This means this is the first development version of release A. Release A is not released yet.
“CS10117-04B“	FC XX 50 01 43 53 31 30 31 31 37 2D 30 34 42 20 20 20 XX XX FD

0x0151 MAC address (Payload 8 byte)

Start	**Packet-Nr.**	**Command 2 byte**		**MAC Adress**								**Checksum 2 byte**		**End**
				lap 4 byte				uap 2 byte		nap 2 byte				
FC	XX	51	01	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	FD

Command to read and transmit the MAC address of the own device. In this project (COR1043) the Bluetooth device address is transmitted.

For example, the following notation results for the Bluetooth address

00:a0:96:2d:dd:92

```
uint32_t bd_addr.lap = 0x2DDD92;      /* lower address part – Bit 23…00 */
uint16_t bd_addr.uap = 0x96;          /* upper address part – Bit 31…24 */
uint16_t bd_addr.nap = 0x00A0;        /* non significant – Bit 47…32 */
```

→ FC XX 51 00 2D DD 92 00 96 00 A0 XX XX FD

4.3 02XX commands

0x0200 acknowledge (ACK) (Payload 1 byte)

Start	Packet-Nr.	Command 2 byte		Packet number of packet, which should be acknowledged	Checksum 2 byte		End
FC	XX	00	02	XX	XX	XX	FD

All commands, which had been processed successfully, are acknowledged, unless they require a distinct answer. Received data is acknowledged with an ACK.

In this project the stop- and wait technique is used. Therefore, in maximum one ACK can be owing.

4.4 03XX commands

0x0300 not acknowledge (NAK) (Payload 1 byte)

Start	Packet-Nr.	Command 2 byte		Packet number of packet, which should be acknowledged	Checksum 2 byte		End
FC	XX	00	03	XX	XX	XX	FD

With this command, a packet is rejected based on checksum errors. This command contains the packet number of the last received packet. In the following example, the packet no. 4 was corrupt and therefore packet no. 4 is not acknowledged.

4.5 04XX commands

0x0400 reject (REJ) (Payload 1 byte)

Start	Packet-Nr.	Command 2 byte		Packet number of the packet, which is rejected	Checksum 2 byte		End
FC	XX	00	04	XX	XX	XX	FD

A packet / command, which can not be processed, is rejected.

The reason for a Reject can be:

- wrong number of arguments for the received command
- unknown command received
- command received at a point in the communication process when the command can not be processed

4.6 05XX commands

0x0500 identification (Payload 22 byte)

Start	Packet-Nr.	Command 2 byte		*)	**)	Serial number 20 bytes																				Checksum 2 byte		End
FC	XX	00	05	01	29	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	XX	XX	FD

*) producer-ID, here 0x01 = Corscience

**) device-ID (type of the device), here 0x29 = Blood pressure meter

The serial number consists only of characters and is written in Ascii.

The unused bytes of the serial number are filled with 0x00 and are added after the serial number.

0x0501 software version (Payload 2 byte)

Start	Packet-Nr.	Command 2 byte		Software Version 2 byte		Checksum 2 byte		End
FC	XX	01	05	XX	XX	XX	XX	FD

SVN-Build of software version. This is for internal uses.

4.7 06XX commands

0x0600 self test (Payload 4 byte)

Start	Packet-Nr.	Command 2 byte		Selftest 4 byte				Checksum 2 byte		End
FC	XX	00	06	XX	00	00	00	XX	XX	FD

Selftest byte 1	Selftest byte 2	Selftest byte 3	Selftest byte 4
1 byte	1 byte	1 byte	1 byte

Selftest byte 1							
Bluetooth Pin	Protocol to use	Remote CoD	Device to connect to	Remote service name	Remote device name	Phone Number	Serial Number
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Selftestbyte2, Selftestbyte3 and Selftestbyte4 are reserved for future use and always 0x00.

Serial number:	the serial number which is stored in flash is not valid
Phone number:	the phone number stored in flash is not valid
Remote device name:	the remote device name stored in flash is not valid
Remote service name:	the remote service name stored in flash is not valid
Device to connect to:	the device to connect to variable in flash is invalid
Remote CoD:	the remote class of device stored in flash is invalid
Protocol to use:	the protocol type for the configuarated device is invalid

Bluetooth pin:	the Bluetooth pin stored in the flash is invalid

Example:		
0x01	0b00000001	“Serial Number”
0x20	0b00100000	“Remote CoD”
0x44	0b01000100	“Remote device name” and “Protocol to use”

0x0601 config BP (Payload 22 byte)

Start	**Packet-Nr.**	**Command 2 byte**		**Reserved 1 byte**	**Device 1 byte**	**Phone Number 20 byte**													**Checksum 2 byte**		**End**
FC	XX	01	06	XX	XX	XX	XX	XX	XX	XX	XX	…	XX	XX	XX	XX	XX	XX	XX	XX	FD

The mode in which the Blood pressure meter is used is set with this command. The settings ares stored in the flash. If the command includes non valid settings or could not be stored in the flash, a reject is send as answer.

Device (The kind of device to which the blood pressure meter will connect):	
BL analog	0x00 (predefined CoD: 00 00 02 00, Corscience Protocol)
Mobile phone	0x01 (predefined CoD: 00 00 02 04, AT commands)
IBM Hub	0x02 (predefined CoD: 00 00 01 04, Corscience Protocol)
Bodyphone	0x03 (predefined CoD: 00 00 02 04, Corscience Protocol)
Configurable device	0x04

PhoneNumber (The phone number to which the SMS will be sent if necessary):	**The phone number must be transmitted in ASCII characters in the international format and it must begin with a 0x2B (‘+’). If not all 20 digits are used the rest must be filled with 0x00.**
Example Phone Number:	0x2B 0x34 0x39 0x39 0x31 0x33 0x31 0x39 0x37 0x39 0x38 0x36 0x32 0x32 0x00 0x00 0x00 0x00 0x00 The number of the example above is +49 9131 977986 22.

If this command is received correctly the actual pairing is deleted (confirmed with an acoustic signal, i.e. 3 long beeps) to guarantee that the correct communication protocol is used to communicate with the connected remote device.

0x0602 config New Remote Device (Payload 29 byte)

Start	**Pack-et-Nr.**	**Command 2 byte**		**CoD 4 byte**				**Reserved**				**Remote device name 10 byte**					**Remote service name 10 byte**					**Pro-tocol**	**Checksum 2 byte**		**End**
FC	XX	02	06	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	…	XX	XX	XX	…	XX	XX	XX	XX	XX	XX	FD

Remote device name: 0x4E 0x61 0x6D 0x65 0x00 0x00 0x00 0x00 0x00 0x00

Remote service name: 0x53 0x45 0x52 0x56 0x49 0x43 0x45 0x00 0x00 0x00

This command is used to specify the “Configurable Device” setting of the “config BP” command. There are several possibilities for CoD. Find more information in the world wide web by searching “The General- and Device-Specific Inquiry Access Codes (DIACs)“.

Samples for CoD:	**The class of device of the new remote device (need for filtered inquiry) e.g.:**	
	00 00 01 00	All computers
	00 00 01 04	Desktop
	00 00 01 0C	Laptop
	00 00 01 10	Handheld PC PDA
	00 00 01 14	Palm sized PC PDA
	00 00 01 18	Wearable
	00 00 02 00	All phones
	00 00 02 04	Cellular
	00 00 02 10	Wired Modem
	00 00 02 14	ISDN Access
	00 00 05 00	All Periph. Device
	00 00 05 10	Sensing device
	00 00 05 0C	Remote control
Remote device name:	**The name of the remote device which should be used to connect to.**	
	If the name of the found and of the configured device does not fit a connection could not be established. If the name is shorter than 10 digits the rest of the 10 characters must be filled with 0x00. If the name is longer than 10 digits only the first 10 chacters are checked. The name must consist at least of one character.	
Remote service name:	**the name of the remote service which should be used to connect to.**	
	If the name of the found and of the configured service does not fit a connection could not be established. If the name is shorter than 10 digits the rest of the 10 characters must be filled with 0x00. If the service name is longer than 10 digits only the first 10 chacters are checked. The name must consist at least of one character.	
Protocol:	**This configures the protocol which should be used if a connction is established to the configured device.**	
	Corscience protocol “Active Data” Mode	0x00
	AT commands	0x01
	Corscience protocol “Passive Data” Mode	0x02

CAUTION: When sending this command, the command “config BP” (0x0601) must also be sent to ensure proper function.

0x0603 config bluetooth pin (Payload 11 byte)

Start	Packet-Nr.	Command 2 byte		Pin length	Bluetooth Pin 10 byte										Checksum 2 byte		End
FC	XX	03	06	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	FD

This command is used to change the Bluetooth pin which is used for pairing two Bluetooth devices.

The pin could have a maximum length of ten bytes and is coded in Ascii. Only characters should be used for the pin. The unused part of the Bluetooth Pin field in this command should be filled with 0x00. The Pin must have at least a length of one byte.

Example:

Pin = 1234	FC XX 03 06 04 31 32 33 34 00 00 00 00 00 00 XX XX FD

The predefined bluetooth pin is 1609.

0x0631 buzzer sequence (Payload 0 byte)

Start	Packet-Nr.	Command 2 byte		Checksum 2 byte		End
FC	XX	31	06	XX	XX	FD

Activates the buzzer. (This helps to identify which BT devices are connected.)

0x0632 buzzer state (Payload 1 byte)

Start	Packet-Nr.	Command 2 byte		Buzzer state	Checksum 2 byte		End
FC	XX	32	06	XX	XX	XX	FD

Explanation	
0x00	Buzzer off (valid only to the next reset)
0x01	Buzzer on (default)

Is acknowledged by the bt device with an ACK (see 4.3).This command should not be used, as the blood pressure meter cannot signalise any communication error to the user. The only possibility is that the receiving unit handles all errors sufficiently and guaranties the safety of the wearer. It has to be kept in mind that this must also be true, when the communication is interrupted.

4.8 07XX commands

The transmission of measured data of the blood pressure meter must not be interrupted by other requests from the receiver.

0x0706 transmit blood pressure data (Payload 11 byte)

Start	Packet-Nr.	Command 2 byte		Measured value											Check-sum 2 byte		End
				Time stamp 6 byte						Blood pressure value 5 byte							
FC	XX	06	07	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	FD

Time stamp: the time when the blood pressure value was taken						
Name:	year	month	day	hour	minute	second
Size:	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte
Values:	0-255	1-12	1-31	0-23	0-59	0-59

Blood pressure value: the measured blood pressure value			
IHB (Insufficient Heart Beat) value	systolic	diastolic	pulse
1 byte	2 byte	1 byte	1 byte

Note: In the previous protocol version (1) the pressure unit (0x00 mmHg, 0x01 kPa) was transmitted instead of the IHB value.

Example	
timestamp:	30.08.2009 - 16:24:40 o’clock
IHB value:	0 (0 = no arrhythmia detected; 1 = arrhythmia detected)
systolic pressure	133
diastolic pressure	74
pulse	68 bpm
FC XX 06 07 09 08 1E 10 18 28 00 00 85 4A 44 XX XX FD	

0x07FA send no data (Payload 0 byte)

Start	Packet-Nr.	Command 2 byte		Checksum 2 byte		End
FC	XX	FA	07	XX	XX	FD

This command is sent if data is requested and no more blood pressure values are available for transmission.

4.9 08XX commands

0x0800 request (Payload 2 byte)

Start	Paket-Nr.	Command 2 byte		requested command		CHK-SUM 2 byte		END
FC	XX	00	08	XX	XX	XX	XX	FD

This command is used to request other commands. The blood pressure meter will answer this request.

To get the last measured values in “Passive Data”-Mode send: FC XX 00 08 06 07 XX XX FD

5. APPENDIX

5.1 Example for communication via Corscience protocol

Note: all commands are transmitted LSB first

Example 1: Service Mode

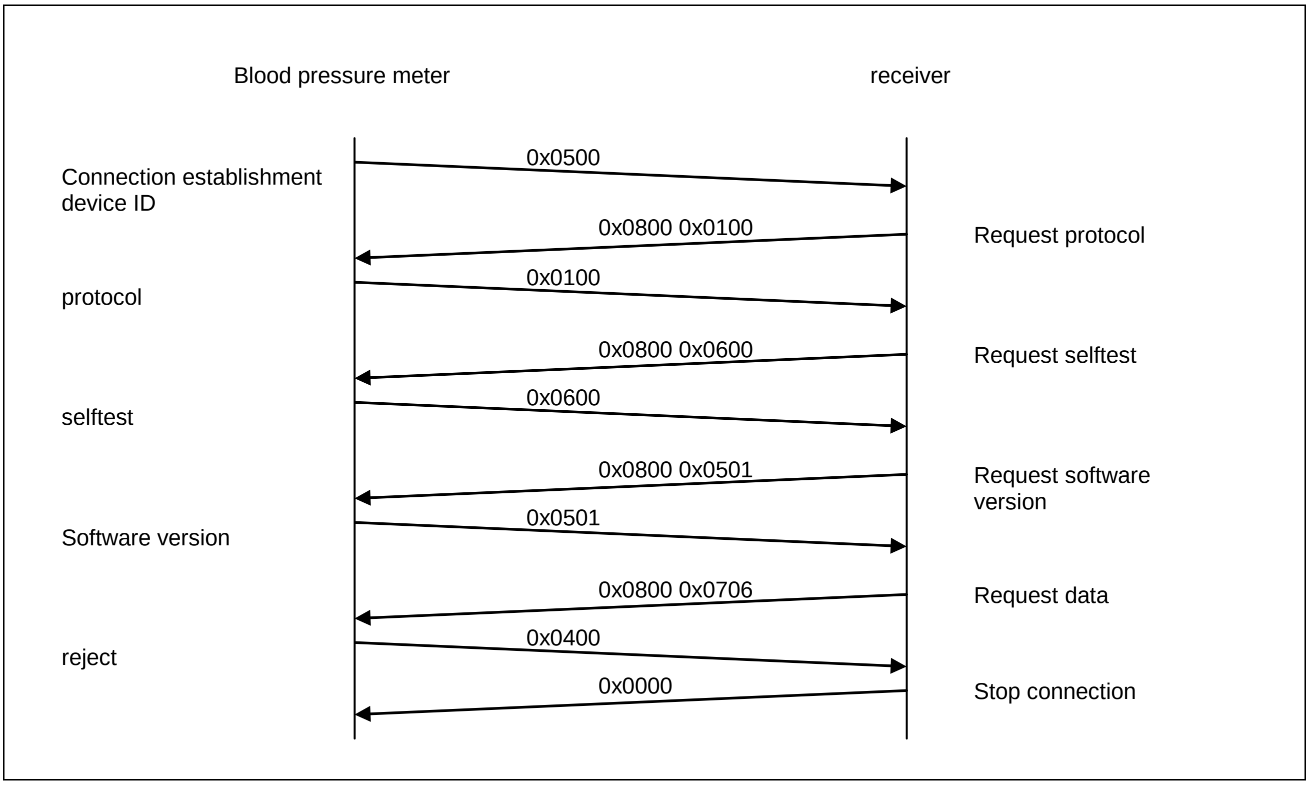

Figure 4: Communication protocol service mode

Example 2: “Passive Data” Mode

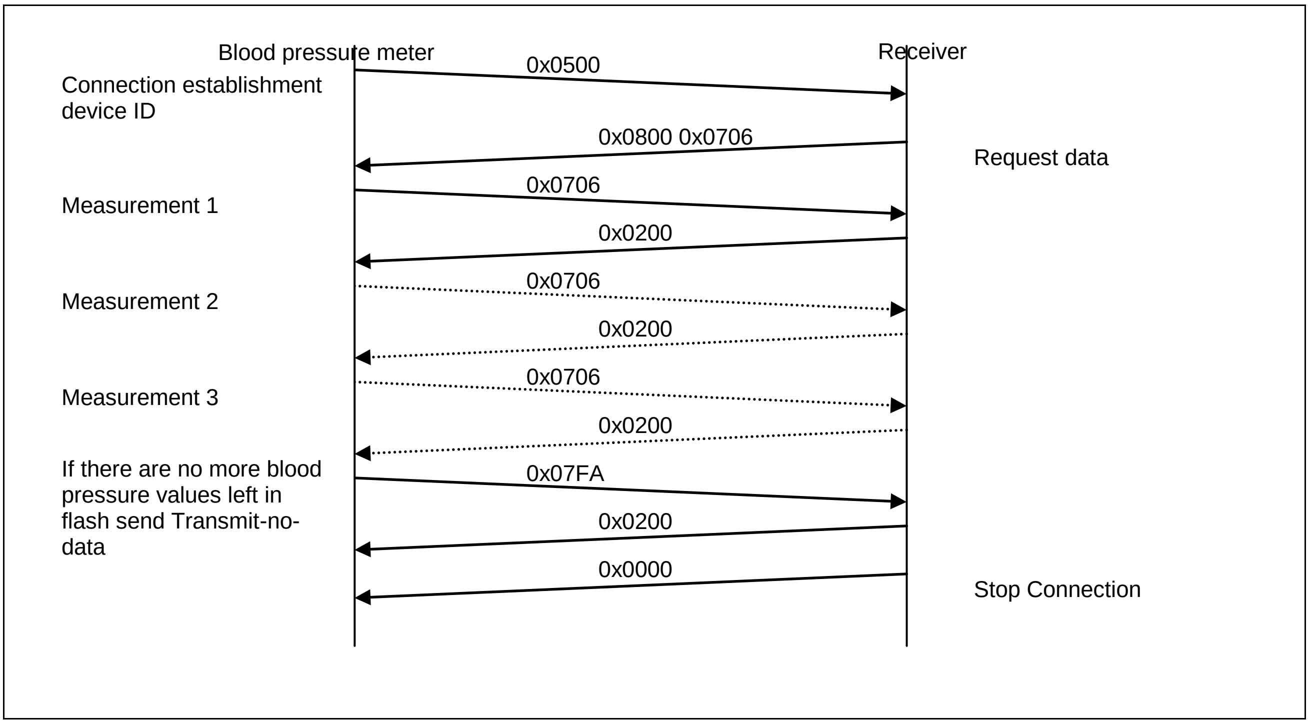

Figure 5: Communication protocol passive data mode

Example 3: “Active Data” Mode

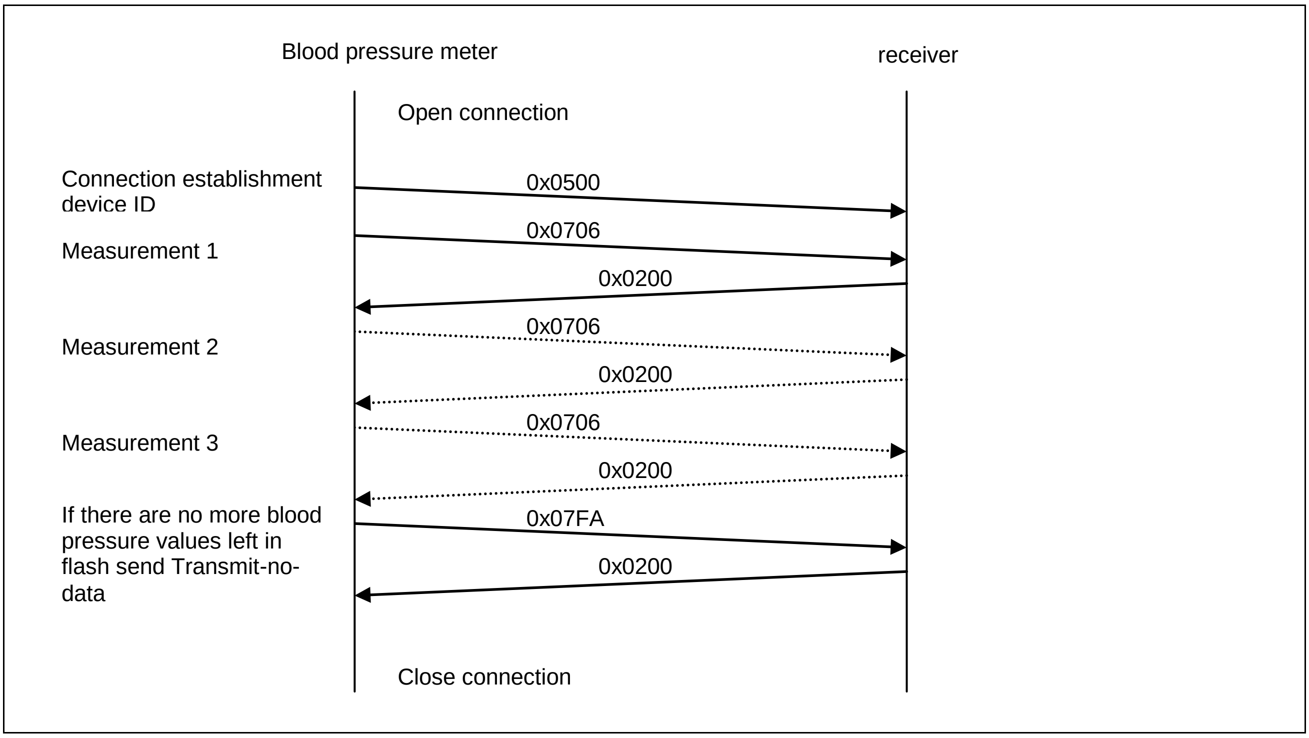

Figure 6: Communication protocol passive data mode

5.2 Protocol for the SMS

The SMS which is sent by the blood pressure meter via the Bluetooth dial up network profile to a mobile phone and then via GSM has the following structure:

Example:	**cor_336800447L;01.01.2000;00:00:00;mmHg;SYS:128;DIA:084;PULS:074;IHB:000;**	
Explanation:	cor_	
	336800447L	(Serial number: „33“, „System ID“; „System serial number“)
	01.01.2000	(Date: DD.MM.YYYY)
	00:00:00	(Time: HH:MM:SS)
	mmHg	(unit: mmHg oder _kPa)
	SYS:128	(systolic bood pressure value: 128)
	DIA:084	(diastolic blood pressure value: 84)
	PULS:074	(pulse: 74 bpm)
	IHB:000	(IHB value: 000 = no arrhythmia detected, 001 = arrhythmia detected)