RW BLE Blood Pressure Profile (BLP) Interface Specification

Interface Specification
RW-BLE-PRF-BLP-IS
Version 8.00
2015-07-29

Revision History

Version	Date	Revision Description	Author
0.1	May 2nd 2012	Initial Release	FBE
0.2	June 19th 2012	Update according review comments	FBE
0.3	August 13th 2012	Update according to database redesign	FBE
0.4	October 23th 2012	Corrections	FBE
0.5	December 4th 2012	Client Multi-Instance API	LT
7.0	November 28th 2014	Update for BLE 4.1	CM
8.0	July 29th 2015	Update for BLE 4.2	CM

Table of Contents

1 Overview

1.1 Document Overview

This document describes the non-standard interface of the RW BLE Blood Pressure Profile implementation. Along this document, the interface messages will be referred to as API messages for the profile block(s).

Their description will include their utility and reason for implementation for a better understanding of the user and the developer that may one day need to interface them from a higher application.

1.2 Protocol Overview

The Bluetooth Low Energy Blood Pressure profile enables the user to receive blood pressure measurements from a blood pressure sensor device and also configure it for different use cases. Within the profile, two roles can be supported: **Collector** and **Sensor**. The Collector must support the GAP Central Role and the Sensor, the GAP Peripheral role. The profile requires a connection to be established between the two devices for its functionality.

The functionality of a profile requires the presence of certain services and attributes on one of the two devices, which the other device can manipulate. In this case, the Blood Pressure device must have one instance of the Blood Pressure Service(BPS) and one instance of Device Information Service(DIS) in its attribute database. The Blood Pressure Profile Collector (BLPC) will discover these services and their characteristics, and it may then configure them to cause the Blood Pressure Profile Sensor (BLPS) device to take measurements and indicate/notify them to the Collector.

The various documents edited by the Bluetooth SIG Medical Working group present different use cases for this profile, their GATT, GAP and security, mandatory and optional requirements. The BLP profile and BPS, DIS services specifications have been adopted by the Bluetooth SIG on October 25th 2011 (**[1]**, **[2]**, **[3]**). Their related Test Specifications have been released at the same time and are referenced in **[4]**, **[5]**, **[6]**.

The profile is implemented in the RW-BLE software stack as two tasks, one for each role. Each task has an API decided after the study of the profile specifications and test specifications, and it is considered to be minimalistic and designed for a future application which would combine the profile functionality with the device connectivity and security procedures.

1.3 Firmware Implementation Overview

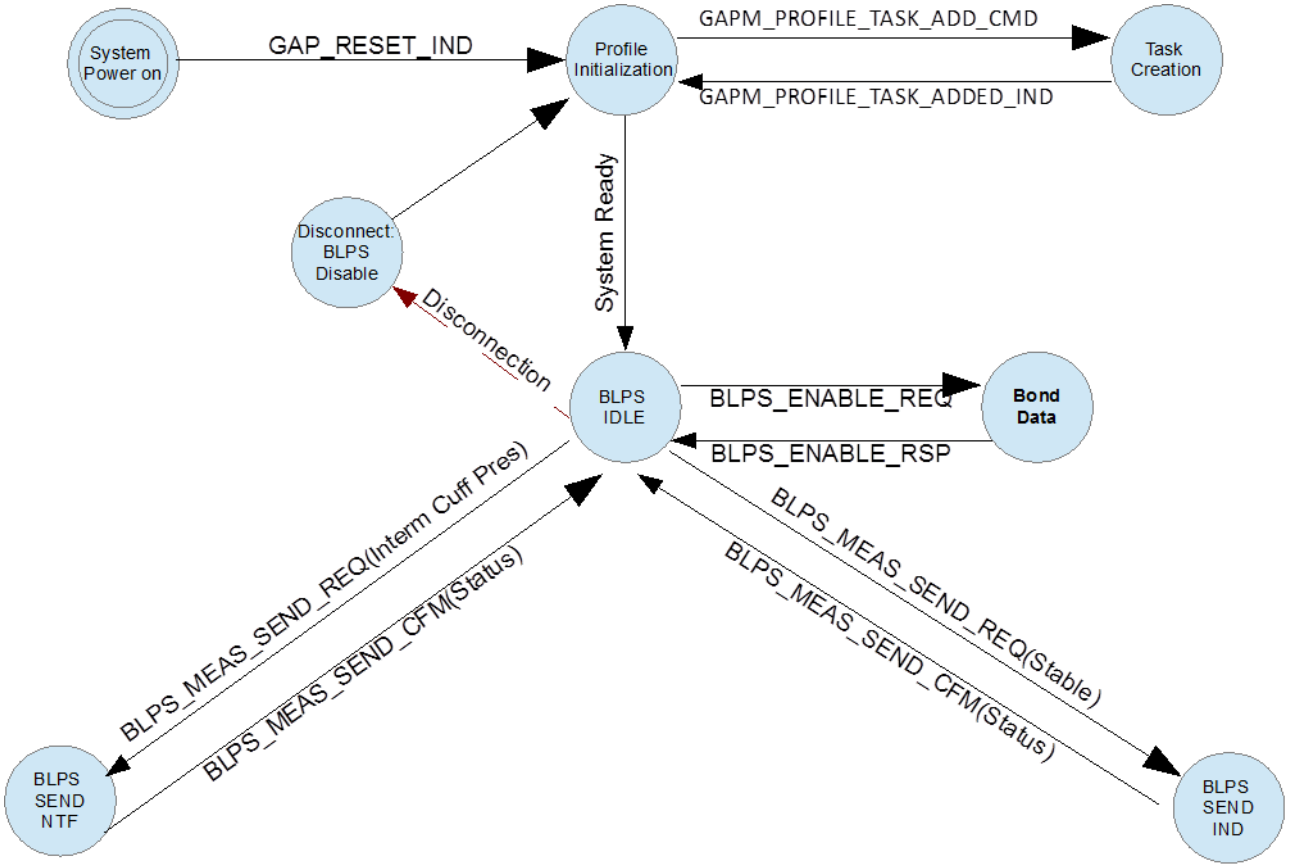
Basically, if a device needs only be Blood Pressure Profile Sensor, the firmware should be compiled with this role only, and inversely for the Collector role. The role enables the part of the DB, which, important to know, will be hidden by the Blood Pressure Sensor until its role is enabled post-connection establishment.

The Applications which will control the roles on end-products are responsible with creating the connection between the devices, using suggested advertising intervals and data, connection intervals, security levels, etc. The Profile implementation allows modulating the behavior depending on the final needs. Profile role enabling should be immediate after connection creation in order to allow correct profile behavior with the peer device.

2 Blood Pressure Profile Sensor

This role is meant to be activated on the device that acts as blood pressure sensor and sends measurement values to the Collector. It implies it is a GAP Peripheral. The FW task for this role will act following the configuration set by the Collector in the BPS characteristics. Please refer to "blps_task.h" for implementation of this API.

This task only has two states, IDLE and CONNECTED.



Blood Pressure Profile Sensor State Machine

2.1 Initialization/Database creation

During the initialization phase of the Blood Pressure Sensor, the memory for this task must be allocated using the message GAPM_PROFILE_TASK_ADD_CMD provided by the GAPM interface. Apart from the security level, the following parameters should be filled:

Parameters:

Type	Parameters	Description
uint8_t	features	Blood Pressure features used to create database: - BLPS_INTM_CUFF_PRESS_SUP
uint8_t	prfl_cfg	Status of the ntf/ind configuration

Response: GAPM_PROFILE_TASK_ADDED_IND

Description: This message shall be send after system power-on (or after GAP Reset) in order to create blood pressure

profile task. This database will be visible from a peer device bun not usable until BLPS_ENABLE_REQ message is sent within a BLE connection.

Please note that the Blood Pressure profile requires the presence of three DIS characteristics : *Manufacturer Name String, Model Number, System Identifier*. It is application's responsibility to add an instance of the DIS into the database by using the same API message (please see the RW BLE Device Information Service Interface Specification document [9]).

The security level is very important because it allows the Application to modulate the protection of the attributes related to the profile, in this case, the DIS and BPS. The implementation only allows modulation of the read/write/notify/indicate permissions (!not properties) of the characteristic values. (if unauthenticated is requested, then a Read/Write to a characteristic won't be allowed if the link between the devices is not Unauthenticated level of security (Mode 1 Level 2).

2.2 BLPS_ENABLE_REQ

Source: TASK_APP

Destination: TASK_BLPS

Required State: IDLE

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection index for which the profile blood pressure sensor role is enabled.
uint16_t	bp_meas_ind_en	Value stored for Blood Pressure indication Client Configuration Char.
uint16_t	interm_cp_ntf_en	Value stored for intermediate cuff pressure notification Client Configuration Char.
uint16_t	bp_feature	Specific blood pressure feature description. (See Blood Pressure Service Spec [2])

Response: BLPS_ENABLE_RSP

Description: This API message is used for restoring the bond data in Blood Pressure Sensor role of the Blood Pressure profile.

2.3 BLPS_ENABLE_RSP

Source: TASK_BLPS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Status of the operation

Description: This API message informs the application about the result of the enabling operation.

2.4 BLPS_MEAS_SEND_REQ

Source: TASK_APP

Destination: TASK_BLPS

Required State: CONNECTED

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection index for which the profile blood pressure sensor role is enabled.
uint16_t	flag_interm_cp	Own code for differentiating between Blood Pressure Measurement, and Intermediate Cuff Pressure Measurement characteristics
struct bps_bp_meas	meas_val	Blood Pressure Measurement Structure (see Blood Pressure Measurement Structure (struct bps_bp_meas))

Response: BLPS_MEAS_SEND_RSP

Description: This message is used by the application (which handles the blood pressure device driver and measurements) to send a blood pressure measurement through the blood pressure sensor role. The flag_interm_cp determines if measurement is an intermediate cuff pressure that will be notified or blood pressure full value that will be indicated.

Flag parameter determines what is present in the blood pressure structure in the PDU of the indication/notification.

Upon reception of this request, BLPS task will check if the necessary action (indication/notification) is possible with the current configuration set by the Collector in the BTS attributes:

- If no, an error status is sent to the application.
- If action is possible, blood pressure value is packed into a correct format in appropriate attribute value. Notification/indication request is sent to GATT to generate the required PDU for the peer.

If it's an indication, the confirmation will come to the application directly from GATT.

If it's a notification, a confirmation does not come from peer, but through BLPS_MEAS_SEND_CFM right after sending GATT message to notify the Intermediate Cuff Pressure Characteristic.

2.5 BLPS_MEAS_SEND_RSP

Source: TASK_BLPS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection conidx for which the profile blood pressure sensor role is enabled.
uint8_t	status	Status of Blood pressure notification : Error code (uses profiles error codes.)

Description: This message is used by BLPS to send to the application, a confirmation, or error status of a notification request being sent to GATT for the Intermediate Cuff Pressure Char. This message should one day take over the GATT confirmation received for an indication, but so far the GATT confirmations are not identified per handle. The

importance of a confirmation of a specific indication is that the application can erase the blood pressure measurement values that it sent the peer successfully. For intermediate cuff pressure the confirmation is useless, they would not be stored.

This has been implemented this way for the sake of symmetry from Application point of view. (See application error codes Error Codes).

2.6 BLPS_CFG_INDNTF_IND

Source: TASK_BLPS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection conidx for which the profile Blood Pressure sensor role is enabled.
uint8_t	char_code	Own code for differentiating between Blood Pressure Measurement, and Intermediate Cuff Pressure Measurement characteristics: - BPS_BP_MEAS_CODE: Blood Pressure Measurement - BPS_INTERM_CP_CODE: Intermediate Cuff Pressure Measurement
uint16_t	cfg_val	Stop/notify value to configured by the peer device in order to send or not notifications

Description: This message is used by BLPS to inform application that peer device has changed notification configuration.

3 Blood Pressure Profile Collector

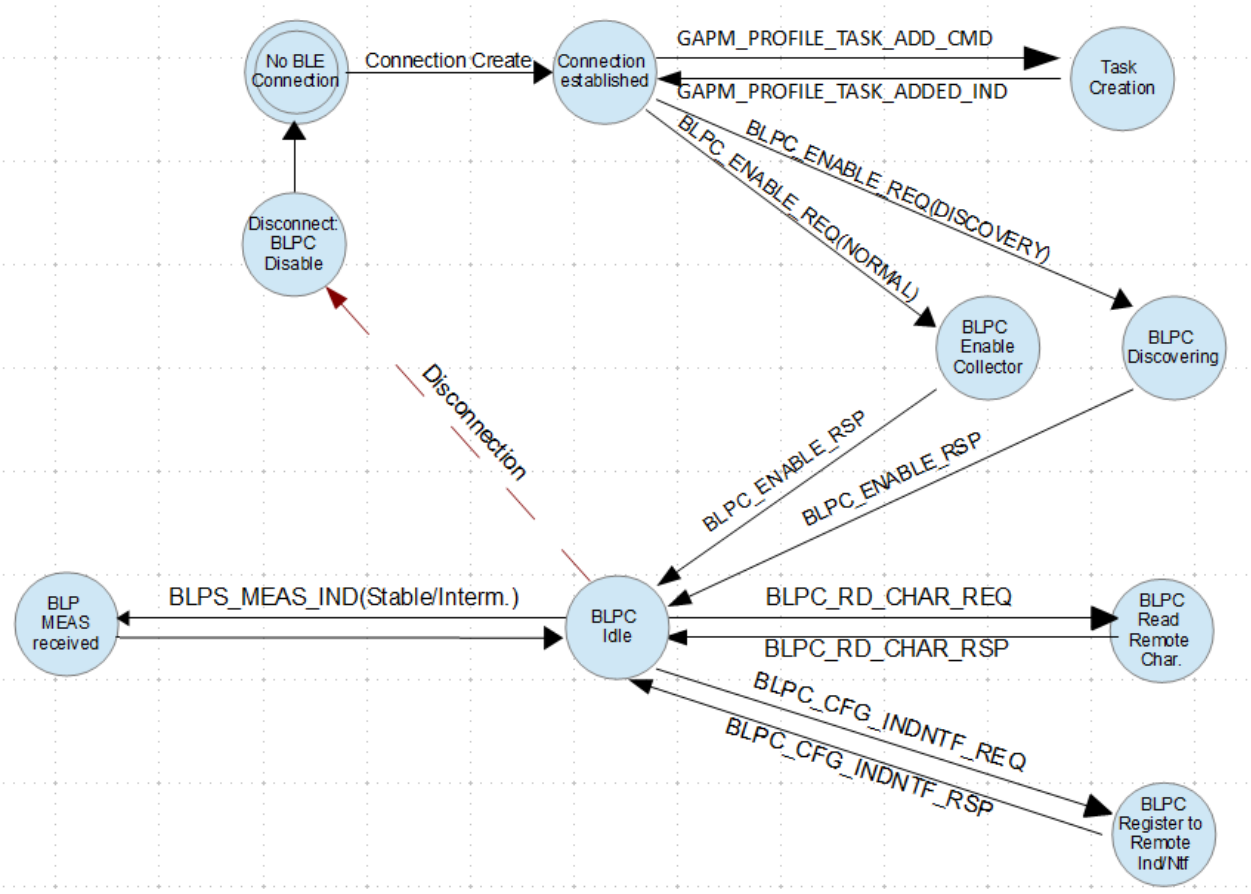
This role is meant to be activated on the device that will collect the blood pressure measurements from the Blood Pressure Sensor. It implies it is a GAP Central. The FW task for this role will discover the BPS and DIS present on the peer Server, after establishing connection, and will allow configuration of the BPS attributes if so required. Please refer to "blpc_task.h" for implementation of this API.

This task has 4 possible states: **FREE, IDLE**, **DISCOVERING** and **BUSY**

Important Note: During the initialization phase of the Blood Pressure Collector, the memory for this task must be allocated using the message GAPM_PROFILE_TASK_ADD_CMD provided by the GAPM interface.
The TASK_BLPC task is multi-instantiated, one instance is created for each connection for which the profile will be enabled and each of these instances will have a different task ID. Thus, it is very important for the application to keep the source task ID of the BLPC_ENABLE_CFM message to be able to communicate with the peer device linked to this task ID once it has been enabled.

The term TASK_BLPC_IDX will be used in the rest of the document to refer to any instance of the Blood Pressure Profile Client Role Task. The term TASK_BLPC will refer to the first instance of this task.

A few proprietary error codes are defined for this role: (see Error Codes).



Blood Pressure Profile Collector State Machine

3.1 BLPC_ENABLE_REQ

Source: TASK_APP

Destination: TASK_BLPC

Required State: All

Parameters:

Type	Parameters	Description
uint8_t	con_type	Connection type: 1st discovery(configuration)(0) or normal connection.(1)
struct bps_content	bps	Existing handle values BPS (see Blood Pressure Content Structure (struct bps_content))

Table 1: Blood Pressure Content Structure (struct bps_content)

Type	Parameters	Description
struct prf_svc	svc	service info (see Service Handle Structure (struct prf_svc))
struct prf_char_inf	chars[0]	Blood Pressure Measurement characteristic (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[1]	Intermediate Cuff pressure characteristic (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[2]	Blood Pressure Feature characteristic (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_desc_inf	descs[0]	Blood Pressure Measurement client configuration descriptor (see Characteristic Descriptor Info Structure (struct prf_char_inf))
struct prf_char_desc_inf	descs[1]	Intermediate Cuff pressure client configuration descriptor (see Characteristic Descriptor Info Structure (struct prf_char_inf))

Response: BLPC_ENABLE_RSP

Description: This API message is used for enabling the Collector role of the Blood Pressure profile. This Application message contains BLE connection handle, the connection type and the previously saved discovered BPS and DIS details on peer.

The connection type may be 0 = Connection for discovery/initial configuration or 1 = Normal connection. This parameter is used by Application to discover peer device services once at first connection. Application shall save those information to reuse them for other connections. During normal connection, previously discovered device information can be reused.

This is usefull since most use cases allow Bood Pressure Sensor to disconnect the link once all measurements have been sent to Collector.

If it is a discovery /configuration type of connection, the BPS and DIS parameters are useless, they will be filled with 0's.

Otherwise they will contain pertinent data which will be kept in the Collector environment while enabled. It allows for the Application to not be aware of attribute details.

For a normal connection, the response to this request is sent right away after saving the BPS and DIS content in the environment and registering BLPC in GATT to receive the indications and notifications for the known attribute handles in BPS that would be notified/indicated. For a discovery connection, discovery of the peer BPS and DIS is started and

the response will be sent at the end of the discovery with the discovered attribute details.

The Task for this profile role will go from IDLE state to CONNECTED state for a normal connection, and to DISCOVERING state for a discovery/configuration type of connection.

3.2 BLPC_ENABLE_RSP

Source: TASK_BLPC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Enable status: discovery error code if anything goes wrong during a configuration type connection. (see Error Codes)
struct bps_content	bps	Existing handle values BPS (see Blood Pressure Content Structure (struct bps_content))

Description: This API message is used by the Collector in order to either send the discovery results of BPS on the Blood Pressure or to simply confirm enabling of Collector role if it is a normal connection and the attribute details are already known.

3.3 BLPC_RD_CHAR_REQ

Source: TASK_APP

Destination: TASK_BLPC_IDX

Required State: CONNECTED

Parameters:

Type	Parameters	Description
uint8_t	char_code	Code for which characteristic to read.

Response: BLPC_RD_CHAR_RSP

Description: This API message is used by the application to send a GATT_READ_CHAR_REQ with the parameters deduced from the char_code. The definitions for the different mapping codes for characteristics that are possibly readable are in blpc.h (for BPS) and in svc.h (for DIS). Upon reception of this message, BLPC checks whether the parameters are correct, then if the handle for the characteristic is valid (not 0x0000) and the request is sent to GATT. When the peer has responded to GATT, and the response is routed to BLPC, the BLPC_RD_CHAR_RSP message will be generically built and the Application must be able to interpret it based on the read request it made. And error status is also possible either for the Read procedure or for the application request, in the second case, the BLPC_RD_CHAR_RSP message is sent to Application with the error code.

No parsing intelligence of the received response is added in this API handler, so all the work of interpretation must be added in the Application depending of its request and use of the response.

3.4 BLPC_RD_CHAR_RSP

Source: TASK_BLPC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
struct att_info_data	data	Structure containing the read value • uint8_t: each result length • uint8_t: data length • uint8_t[0x18]: data

Description: This API message is used by the Collector role to inform the Application of a received read response. The status and the data from the read response are passed directly to Application, which must interpret them based on the request it made.

3.5 BLPC_CFG_INDNTF_REQ

Source: TASK_APP

Destination: TASK_BLPC_IDX

Required State: CONNECTED

Parameters:

Type	Parameters	Description
uint8_t	char_code	Code for which characteristic to configure in ntf/ind
uint16_t	cfg_val	Configuration value.

Response: BLPC_WR_CHAR_RSP

Description: This API message is used by the application to send a GATT_WRITE_CHAR_REQ with the parameters deduced from the char_code and cfg_val. The definitions for the different codes for characteristics that can be configured to indicate/notify are in blpc.h. Upon reception of this message, BLPC checks whether the parameters are correct, then if the handle for the characteristic is valid (not 0x0000) and the request is sent to GATT. When the peer has responded to GATT, and the response is routed to BLPC, the BLPC_WR_CHAR_RSP message will be generically built and sent to Application. An error status is also possible either for the Write procedure or for the application request, in the second case, the BLPC_WR_CHAR_RSP message is sent to Application.

3.6 BLPC_CFG_INDNTF_RSP

Source: TASK_BLPC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Error code: (see Error Codes)

Description: This API message is used by the Collector role to inform the Application of a received write response. The status and the data from the write response are passed directly to Application, which must interpret them based on the request it made.

3.7 BLPC_BP_MEAS_IND

Source: TASK_BLPC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint16_t	flag_interm_cp	Flag indicating if it is a intermediary cuff pressure measurement (0) or stable blood pressure measurement (1).
struct bps_bp_meas	meas_val	Blood Pressure Measurement Structure (see Blood Pressure Measurement Structure (struct bps_bp_meas))

Description: This API message is used by the Collector role to inform the Application of a received blood pressure value, either by notification flag_interm_cp = intermediate) or indication (flag_interm_cp = stable). The application will do what it needs to do with the received measurement. No confirmation of reception is needed because the GATT sends it directly to the peer.

4 Miscellaneous

4.1 Error Codes

See RW BLE Host Error Code Interface Specification **[7]**

4.2 Types

Table 2: Blood Pressure Measurement Structure (struct bps_bp_meas)

Type	Parameters	Description
uint8_t	flags	Information available in measurement
uint8_t	user_id	User Identifier
SFLOAT	systolic	Systolic measurement value (mmHg/kPa)
SFLOAT	diastolic	Diastolic measurement value (mmHg/kPa)
SFLOAT	mean_arterial_pressure	Mean arterial pressure measurement value (mmHg/kPa)
SFLOAT	pulse_rate	Measured pulse rate
uint16_t	meas_status	Measurement Status information
struct prf_date_time	time_stamp	Time Stamp (see Time Stamp Structure (struct prf_date_time))

Table 3: Time Stamp Structure (struct prf_date_time)

Type	Parameters	Description
uint16_t	year	Year
uint8_t	month	Month (1-12)
uint8_t	day	Day (1-31)
uint8_t	hour	Hour (0-24)
uint8_t	min	Minutes (0-60)
uint8_t	sec	Seconds (0-60)

Table 4: Service Handle Structure (struct prf_svc)

Type	Parameters	Description
uint16_t	shdl	Start handle
uint16_t	ehdl	End handle

Table 5: Characteristic Info Structure (struct prf_char_inf)

Type	Parameters	Description
uint16_t	char_hdl	Characteristic handle
uint16_t	val_hdl	Value handle
uint8_t	prop	Characteristic properties

Table 6: Characteristic Descriptor Info Structure (struct prf_char_inf)

Type	Parameters	Description
uint16_t	desc_hdl	Descriptor handle

Table 7: Device Information Service Content Structure (struct dis_content)

Type	Parameters	Description
struct prf_svc	svc	service info (see Service Handle Structure (struct prf_svc))
struct prf_char_inf	chars[0]	Manufacturer Name String (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[1]	Model Number String (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[2]	Serial Number String (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[3]	Hardware Revision String (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[4]	Firmware Revision String (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[5]	Software Revision String (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[6]	System ID (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[7]	IEEE 11073-20601 Regulatory Certification Data List (see Characteristic Info Structure (struct prf_char_inf))

5 Abbreviations

Abbreviation	Original Terminology
API	Application Programming Interface
BLE	Bluetooth Low Energy
DIS	Device Information Service
BLPC	Blood Pressure Profile Collector
BLPS	Blood Pressure Profile Sensor
BLP	Blood Pressure Profile
BPS	Blood Pressure Service
GAP	Generic Access Profile
GATT	Generic Attribute Profile
RW	RivieraWaves

References

[1]	**Title**	Blood Pressure Profile		
	Reference	BLP_SPEC_V10r00		
	Version	V10r00	**Date**	October 25th 2011
	Source	Bluetooth SIG – Medical Working Group		

[2]	**Title**	Blood Pressure Service		
	Reference	BPS_V10r00		
	Version	V10r00	**Date**	October 25th 2011
	Source	Bluetooth SIG – Medical Working Group		

[3]	**Title**	Device Information Service		
	Reference	DIS_SPEC_V10		
	Version	V10r00	**Date**	May 24th 2011
	Source	Bluetooth SIG – Medical Working Group		

[4]	**Title**	Blood Pressure Profile (BLP) 1.0		
	Reference	BLP.TS.1.0.0		
	Version	1.0.0	**Date**	October 25th 2011
	Source	Bluetooth SIG		

[5]	**Title**	Blood Pressure Service (BPS) 1.0		
	Reference	BPS.TS.1.0.0		
	Version	1.0.0	**Date**	October 25th 2011
	Source	Bluetooth SIG		

[6]	**Title**	Device Information Service (DIS) 1.0		
	Reference	DIS.TS.1.0.0		
	Version	1.0.0	**Date**	May 24th 2011
	Source	Bluetooth SIG		

[7]	**Title**	RW BLE Host Error Code Interface Specification		
	Reference	RW-BLE-HOST-ERR-CODE-IS		
	Version	0.1	**Date**	August 13th 2012
	Source	RivieraWaves SAS		

[8]	**Title**	ATTDB Interface Specification		
	Reference	RW-BLE-ATTDB-IS		
	Version	0.4	**Date**	August 13th 2012
	Source	RivieraWaves SAS		

[9]	**Title**	Device Information Service Interface Specification		
	Reference	RW-BLE-DIS-IS		
	Version	0.1	**Date**	August 14th 2012
	Source	RivieraWaves SAS		