RW BLE Heart Rate Profile (HRP) Interface Specification

Interface Specification
RW-BLE-PRF-HRP-IS
Version 8.0
2015-07-29

RivieraWaves
http://www.rivierawaves.com/

Revision History

Version	Date	Revision Description	Author
0.1	August 11th 2012	Initial Release	FBE
0.2	August 13th 2012	Corrections	FBE
0.3	October 23th 2012	Corrections	FBE
0.4	December 3rd 2012	Client Multi-Instances API	LT
7.0	December 1st 2014	Update to BLE 4.1	CM
7.1	January 6th 2015	Update of HRPS enable message	FBE
8.0	July 29th 2015	Update to BLE 4.2	CM

Table of Contents

1 Overview

1.1 Document Overview

This document describes the non-standard interface of the RW BLE Heart Rate 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 Heart Rate profile enables the user to receive Heart Rate measurements from a Heart Rate 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 Heart Rate device must have one instance of the Heart Rate Service(HRS) and one instance of Device Information Service(DIS) in its attribute database. The Heart Rate Profile Collector (HRPC) will discover these services and their characteristics, and it may then configure them to cause the Heart Rate Profile Sensor (HRPS) 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 HRP profile and HRS, 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

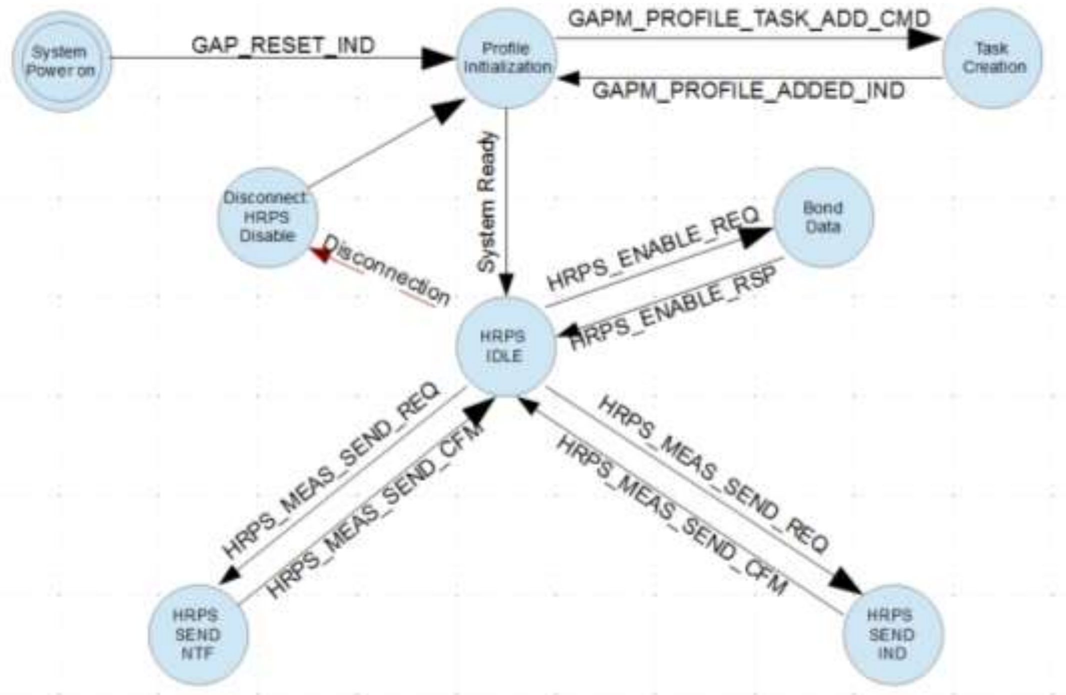
Basically, if a device needs only be Heart Rate Profile Sensor, the firmware should be compiled with this role only, and inversely for the Collector role. The role enables the part of the DB, which, important to know, will be hidden by the Heart Rate Sensor until its role is enabled post-connection establishment.

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 Heart Rate Profile Sensor

This role is meant to be activated on the device that acts as Heart Rate 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 HRS characteristics. Please refer to “hrps_task.h” for implementation of this API.

This task only has two states, IDLE and CONNECTED.



Heart Rate Profile Sensor State Machine

2.1 Initialization/Database creation

During the initialization phase of the Heart Rate 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	Heart rate features used to create database: - HRPS_BODY_SENSOR_LOC_CHAR_SUP: Body Sensor Location Feature Supported - HRPS_ENGY_EXP_FEAT_SUP: Energy Expanded Feature Supported
uint8_t	body_sensor_loc	Exact location of the sensor

Response: GAPM_PROFILE_TASK_ADDED_IND

Description: This message shall be send after system power-on (or after GAP Reset) in order to create heart rate profile database. This database will be visible from a peer device but not usable until HRPS_ENABLE_REQ message is sent within a BLE connection.

Please note that the Heart Rate profile requires the presence of one DIS characteristic: *Manufacturer Name String*. It is application's responsibility to add an instance of the DIS into the database by using the same GAPM message (please see the RW BLE Device Information Service Interface Specification document[9]).

2.2 HRPS_ENABLE_REQ

Source: TASK_APP

Destination: TASK_HRPS

Required state: IDLE

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection index for which the profile Heart Rate sensor role is enabled.
uint16_t	hr_meas_ntf_en	Heart Rate Notification configuration

Response: HRPS_ENABLE_RSP

Description: This API message is used for restoring the Heart Rate Sensor bond data for a given connection index. Before sending this message, a BLE connection shall exist with peer device.

Application must also set the Heart Rate features that will set in HRS attribute database in order to be available (in read mode) for peer device collector.

2.3 HRPS_ENABLE_RSP

Source: TASK_HRPS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection index for which the profile Heart Rate sensor role is enabled.
uint8_t	status	Status of the enabling operation

Description: This API message is used by the Heart Rate sensor role to inform the Application of a correct enable.

2.4 HRPS_MEAS_SEND_REQ

Source: TASK_APP

Destination: TASK_HRPS

Required state: CONNECTED

Parameters:

Type	Parameters	Description
struct hrs_hr_meas	meas_val	Heart Rate Measurement Structure (see Heart Rate Measurement Structure (struct hrs_hr_meas))

Response: HRPS_MEAS_SEND_RSP

Description: This message is used by the application (which handles the Heart Rate device driver and measurements) to send a Heart Rate measurement through the Heart Rate sensor role to every connected device.

Upon reception of this request, HRPS task will check if the necessary action (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, Heart Rate value is packed into a correct format in appropriate attribute value. Notification request is sent to GATT to generate the required PDU for the peer.

Notification confirmation does not come from peer, but through HRPS_MEAS_SEND_RSP right after sending GATT message.

2.5 HRPS_MEAS_SEND_RSP

Source: TASK_HRPS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Status of Heart Rate notification : Status error code: (see Error Codes)

Description: This message is used by HRPS to send to the application, a confirmation, or error status of a notification request being sent to GATT.

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

2.6 HRPS_CFG_INDNTF_IND

Source: TASK_HRPS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection index for which the profile Heart Rate sensor role is enabled.
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 HRPS to inform application that peer device has changed notification configuration.

2.7 HRPS_ENERGY_EXP_RESET_IND

Source: TASK_HRPS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint16_t	conidx	Connection index for which the profile Heart Rate sensor role is enabled.

Description: This message is used by HRPS to inform application that Energy Expanded value shall be reset

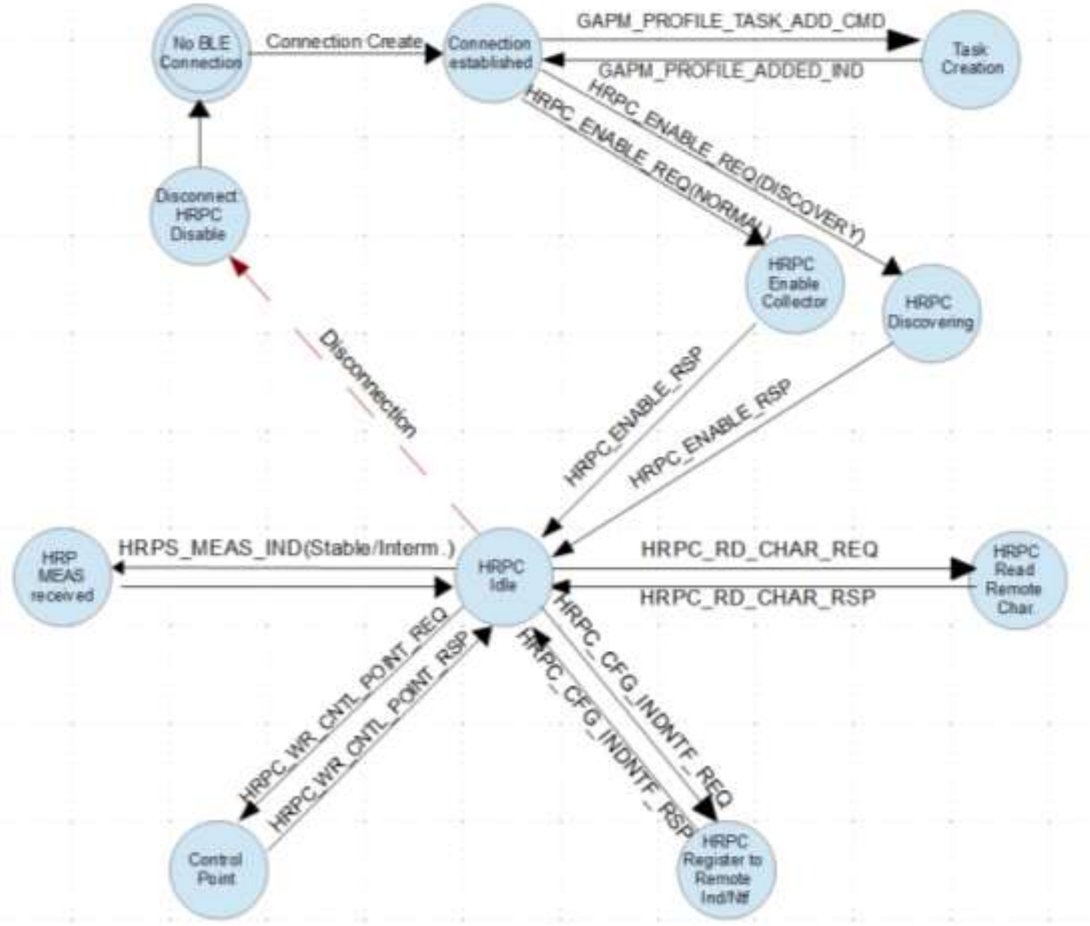
3 Heart Rate Profile Collector

This role is meant to be activated on the device that will collect the Heart Rate measurements from the Heart Rate Sensor. It implies it is a GAP Central. The FW task for this role will discover the HRS present on the peer Server, after establishing connection, and will allow configuration of the HRS attributes if so required. Please refer to “hrpc_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 Heart Rate Collector, the memory for this task must be allocated using the message GAPM_PROFILE_TASK_ADD_CMD provided by the GAPM interface.
The TASK_HRPC 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 HRPC_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_HRPC_IDX will be used in the rest of the document to refer to any instance of the Heart Rate Profile Client Role Task. The term TASK_HRPC will refer to the first instance of this task.

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



Heart Rate Profile Collector State Machine

3.1 HRPC_ENABLE_REQ

Source: TASK_APP

Destination: TASK_HRPC

Required state: All

Parameters:

Type	Parameters	Description
uint8_t	con_type	Connection type: 1st discovery(configuration)(0) or normal connection.(1)
struct hrs_content	hrs	Existing handle values HRS (see Heart Rate Content Structure (struct hrs_content))

Table 1: **Heart Rate Content Structure (struct hrs_content)**

Type	Parameters	Description
struct prf_svc	svc	service info (see Service Handle Structure (struct prf_svc))
struct prf_char_inf	chars[0]	Heart Rate 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]	Heart Rate Feature characteristic (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_desc_inf	descs[0]	Heart Rate 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: HRPC_ENABLE_RSP

Description: This API message is used for enabling the Collector role of the Heart Rate profile. This Application message contains the connection type and the previously saved discovered HRS 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 this information to use them for other connections. During normal connection, previously discovered device information can be reutilized.

This is useful since most use cases allow Heart Rate Sensor to disconnect the link once all measurements have been sent to Collector.

If it is a discovery /configuration type of connection, the HRS 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 HRS content in the environment and registering HRPC in GATT to receive the indications and notifications for the known attribute handles in HRS that would be notified/indicated. For a discovery connection, discovery of the peer HRS 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 HRPC_ENABLE_RSP

Source: TASK_HRPC_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 hrs_content	hrs	Existing handle values HRS (see Heart Rate Content Structure (struct hrs_content))

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

3.3 HRPC_RD_CHAR_REQ

Source: TASK_APP

Destination: TASK_HRPC_IDX

Parameters:

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

Response: HRPC_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 hrpc.h (for HRS). Upon reception of this message, HRPC 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 HRPC, the HRPC_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 HRPC_RD_CHAR_RSP message is sent to Application.

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 HRPC_RD_CHAR_RSP

Source: TASK_HRPC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
struct att_info_data	data	Structure containing the read value • uint16_t: request handle • uint16_t: data length • uint8_t: status • 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 HRPC_WR_CNTL_POINT_REQ

Source: TASK_APP

Destination: TASK_HRPC_IDX

Required state: IDLE

Parameters:

Type	Parameters	Description
uint8_t	value	Reset (1)

Response: HRPC_WR_CNTL_POINT_RSP

Description: This API message is used by the application to write control point attribute in order to reset Energy Expanded value.

3.6 HRPC_WR_CNTL_POINT_RSP

Source: TASK_HRPC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Status of the operation

Response: HRPC_WR_CHAR_RSP

Description: This API message informs the application about the status of the operation performed by the control point.

3.7 HRPC_CFG_INDNTF_REQ

Source: TASK_APP

Destination: TASK_HRPC_IDX

Required state: CONNECTED

Parameters:

Type	Parameters	Description
uint16_t	cfg_val	Configuration value.

Response: HRPC_CFG_INDNTF_RSP

Description: This API message is used by the application to send a GATT_WRITE_CHAR_REQ to the peer device in order to write the provided value in the measurement characteristic. Upon reception of this message, HRPC checks whether the parameter is 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 HRPC, the HRPC_CFG_INDNTF_RSP message will be generically built and sent to Application.

3.8 HRPC_CFG_INDNTF_RSP

Source: TASK_HRPC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	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.

3.9 HRPC_HR_MEAS_IND

Source: TASK_HRPC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
struct hrs_bp_meas	meas_val	Heart Rate Measurement Structure (see Heart Rate Measurement Structure (struct hrs_hr_meas))

Description: This API message is used by the Collector role to inform the Application of a received Heart Rate value by notification. 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: **Heart Rate Measurement Structure (struct hrs_hr_meas)**

Type	Parameters	Description
uint8_t	flags	Measurement Flags
uint8_t	nb_rr_interval	RR-Interval numbers (max 4)
uint16_t [4]	rr_intervals	RR-Intervals
uint16_t	heart_rate	Heart Rate Measurement Value
uint16_t	energy_expended	Energy Expended

Table 3: **Service Handle Structure (struct prf_svc)**

Type	Parameters	Description
uint16_t	shdl	Start handle
uint16_t	ehdl	End handle

Table 4: **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 5: **Characteristic Descriptor Info Structure (struct prf_char_inf)**

Type	Parameters	Description
uint16_t	desc_hdl	Descriptor handle

5 Abbreviations

Abbreviation	Original Terminology
API	Application Programming Interface
BLE	Bluetooth Low Energy
DIS	Device Information Service
HRPC	Heart Rate Profile Collector
HRPS	Heart Rate Profile Sensor
HRP	Heart Rate Profile
HRS	Heart Rate Service
GAP	Generic Access Profile
GATT	Generic Attribute Profile
RW	RivieraWaves

References

[1]	**Title**	Heart Rate Profile		
	Reference	HRP_SPEC_V10r00		
	Version	V10r00	**Date**	July 12th 2011
	Source	Bluetooth SIG – Medical Working Group		

[2]	**Title**	Heart Rate Service		
	Reference	HRS_V10r00		
	Version	V10r00	**Date**	July 12th 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**	Heart Rate Profile (HRP) 1.0		
	Reference	HRP.TS.1.0.0		
	Version	1.0.0	**Date**	July 12th 2011
	Source	Bluetooth SIG		

[5]	**Title**	Heart Rate Service (HRS) 1.0		
	Reference	HRS.TS.1.0.0		
	Version	1.0.0	**Date**	July 12th 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**	DIS Interface Specification		
	Reference	RW-BLE-DIS-IS		
	Version	0.1	**Date**	August 14th 2012
	Source	RivieraWaves SAS		