RW BLE Battery Service Interface Specification

Interface Specification

RW-BLE-BAS-IS

Version 9.02

2019-09-05

Revision History

Version	Date	Revision Description	Author
1.0	2012-10-30	Initial release	LT
2.0	2012-12-03	Client Multi-Instance API	LT
7.00	2014-11-27	Updated for BLE 4.1	FBE
8.00	2015-07-29	Updated for BLE 4.2	CM
9.00	2017-03-09	Updated for BLE 5	LT
9.01	2018-09-03	3.3.2.1 BASC_READ_INFO_REQ Correction in Info parameter description	KYAP
9.02	2019-09-05	3.3.2.5 BASC_BATT_LEVEL_NTF_CFG_RSP Added a missing parameter description (bas_nb)	ZL

Table of Contents

1 Overview

1.1 Document Overview

This document describes the non-standard interface of the RivieraWaves (RW) Bluetooth Low Energy (BLE) Battery Service (BAS) 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 Bluetooth Low Energy Battery Service Overview

The BLE BAS enables a device to expose its current battery level to a peer device.

This service has been implemented as a profile. Within this profile, two roles can be supported: Server role (BASS) and Client role (BASC). The Client role must support the GAP Central Role and the Server role, the GAP Peripheral role. The profile requires a connection to be established between the two devices for its functionality.

The various documents edited by the Bluetooth SIG present different use cases for this profile, their GATT, GAP and security, mandatory and optional requirements. The Battery Service (BAS) specification has been adopted by the Bluetooth SIG on December 27th 2011 ([1]). Its related Test Specification has been released at the same time and is referenced in [2].

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.

The Battery Service structure as defined in the Battery Service specification is exposed in the table below:

Characteristic Name	Requirements	Properties	Security	Descriptors
Battery Level	Mandatory	Read (Mandatory) Notify (Optional)	None	Client Characteristic Configuration *Requirement: If notify is supported* *Permissions: Read (Mandatory), Write (Mandatory)*
				Characteristic Presentation Format *Requirement: If multiple service instances* *Permissions: Read (Mandatory)*

The Battery Level Characteristic value is a percentage from 0% (battery is fully discharged) to 100% (battery is fully charged).

2   BAS Server Role API

2.1   Environment

A device may have more than one instance of the Battery Service. As we currently have a static implementation, the maximal number of BAS instances that can be managed within the server task has been set to 2. This value can be modified in the bass.h file.

When several battery service instances are required by application, the BAS tasks allocate those services on a contiguous handle range into the attribute database.

Battery service task is a mono-instantiated task; it means that connection index is **not** present into task index.

2.2   Initialization / Database Creation

During the initialization phase of the device, to use the Battery Service task, the BASS task has to be allocated and corresponding attribute database initialized, using GAPM API. Application has to send GAPM_PROFILE_TASK_ADD_CMD [4] with specific device required security level and following parameters.

Parameters:

Type	Parameters	Description
uint8_t	bas_nb	Number of BAS instances to add in the database.
uint8_t	features[BASS_NB_BAS_INSTANCES_MAX]	Configuration for each BAS to add.
struct prf_char_pres_fmt	batt_level_pres_format [BASS_NB_BAS_INSTANCES_MAX]	Battery Level Characteristic Characteristic Presentation Format Descriptor value (see Table 4. 1).

The features parameter shall be used to specify if the Battery Level Characteristic supports sending of notifications. Available values are BAS_BATT_LVL_NTF_NOT_SUP (0x00 – Notifications are not supported) and BAS_BATT_LVL_NTF_SUP (0x01 – Notifications supported). If the value of features is BAS_BATT_LVL_NTF_SUP, a Client Characteristic Descriptor is added for the Characteristic; else, it is considered that notifications are not supported.

If bas_nb value is upper than 1, a Characteristic Presentation Format Descriptor will be added for each Battery Level Characteristic added within the BAS.

2.3   API Messages

2.3.1   Database Configuration Messages

These API messages have been defined to allow the application to add one or more instances of the BAS in the database. These messages shall be used during the initialization phase of the device, before any connection establishment with a peer device.

2.3.2 Connection Messages

2.3.2.1 BASS_ENABLE_REQ

Source: TASK_APP

Destination: TASK_BASS

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection Index.
uint8_t	ntf_cfg	Notification Configuration
uint8_t	old_batt_lvl[BASS_NB_BAS_INSTANCES_MAX]	For bonded devices, battery level value saved after the last disconnection.

Response: BASS_ENABLE_RSP

Description: This API message can be used after the connection with a peer device has been established in order to restore known device bond data.

The ntf_cfg parameter is a bit field containing notification configuration for each battery service instances.

Upon reconnection with a bonded device (Normal Connection), if for this peer device sending of notifications was enabled (batt_level_ntf_cfg) and if the Battery Level Characteristic value has changed while the device has been disconnected (old_batt_lvl != current_batt_lvl), the new Battery Level Characteristic value is notified.

2.3.2.2 BASS_ENABLE_RSP

Source: TASK_BASS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection Index.
uint8_t	status	Status code (see [3])

Description: Inform application if restoring bond data for peer device succeed or not.

2.3.3   Communication Messages

2.3.3.1   BASS_BATT_LEVEL_UPD_REQ

Source: TASK_APP

Destination: TASK_BASS

Parameters:

Type	Parameters	Description
uint8_t	bas_instance	Battery Service Instance.
uint8_t	batt_level	Battery Level Characteristic value.

Response: BASS_BATT_LEVEL_UPD_RSP

Description: This API message is sent by the application to update the Battery Level Characteristic value for one of the BAS instance. This value will be stored in the database so that it can be read by the peer device.

All connected devices that are configured to receive notification will be inform about the new battery level.

Note: Current battery level for a device could be considered as bond data in order to inform peer device about an update of the battery level.

2.3.3.2   BASS_BATT_LEVEL_UPD_RSP

Source: TASK_BASS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Status code (see [3])

Description: This API message is sent to the application to inform it if a notification has been sent to the peer device.

2.3.3.3   BASS_BATT_LEVEL_NTF_CFG_IND

Source: TASK_BASS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	conidx	Connection Index.
uint8_t	ntf_cfg	New value of the notification configuration bit field.

Description: This API message is sent to the application when the notification configuration has been modified for one of the Battery Level Characteristics.

3   BAS Client Role API

3.1   Environment

The BLE BAS Client role has been designed to allow a collector to be informed about the current battery level of a device either upon the reception of notifications or upon a read request.

As we currently have a static implementation, the maximal number of BAS instances that can be managed in this role task has been set to 2. This value can be modified in the basc.h file.

Within the BASC task, three state are defined: FREE, IDLE, BUSY.

**Important Note**: The TASK_BASC 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. To communicate with the peer device, the corresponding connection index has to be used to calculate the BASC task instance.

The term TASK_BASC_IDX will be used in the rest of the document to refer to any instance of the Battery Service Client Role Task. The term TASK_BASC will refer to the first instance of this task.

3.2   Initialization

During the initialization phase of the device, to use the Battery Client task, the BASC task has to be allocated using GAPM API. Application has to send GAPM_PROFILE_TASK_ADD_CMD [4].

3.3   API Messages

3.3.1   Connection Messages

3.3.1.1   BASC_ENABLE_REQ

Source: TASK_APP

Destination: TASK_BASC_IDX

Required state: IDLE

Parameters:

Type	Parameters	Description
uint8_t	con_type	Connection type
uint8_t	bas_nb	Number of instances of the BAS that have been found during the last discovery.
struct bas_content	bas[BASC_NB_BAS_INSTANCES_MAX]	Information about the BAS instances that have been found during the last discovery (see Table 4. 2).

Response: BASC_ENABLE_RSP

Description: This API message is used for enabling the Client role of the BAS. This Application message contains connection type and the previously saved discovered BAS details on peer.

The connection type may be PRF_CON_DISCOVERY (0x00) for discovery/initial configuration or PRF_CON_NORMAL (0x01) for a normal connection with a bonded device. Application shall save those information to reuse them for other connections. During normal connection, previously discovered device information can be reused.

If it is a discovery /configuration type of connection, it is useless to fill the BAS parameters (bas_nb and bas) are useless. Otherwise they will contain pertinent data which will be kept in the Client environment while enabled.

For a normal connection, the response to this request is sent right away after saving the BAS content in the environment and registering BASC in GATT to receive the notifications for the known attribute handles in BAS that would be notified (Battery Level Characteristic). For a discovery connection, discovery of the peer BAS 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.

3.3.1.2   BASC_ENABLE_RSP

Source: TASK_BASC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Discovery status (see [3])
uint8_t	bas_nb	Number of instances of the BAS that have been found during the discovery.
struct bas_content	bas[BASC_NB_BAS_INSTANCES_MAX]	Information about the BAS instances that have been found during the discovery (see Table 4. 2).

Description: This API message is used by the Client role task to either send the discovery results of BAS on the peer device and confirm enabling of the Client role, or simply confirm enabling of Client role if it is a normal connection and the attribute details are already known.

3.3.2   Communication Messages

3.3.2.1   BASC_READ_INFO_REQ

Source: TASK_APP

Destination: TASK_BASC_IDX

Required state: IDLE/BUSY

Parameters:

Type	Parameters	Description
uint8_t	info	Characteristic info (see Table 3.1)
uint8_t	bas_nb	Battery Service instance.

Response: BASC_READ_INFO_RSP

Description: This API message shall be used to read the value of a characteristic or a descriptor in the peer device database.

The info parameter shall be one of these values:

Flag	Value	Description
BASC_BATT_LVL_VAL	0x00	Read the Battery Level Characteristic value.
BASC_NTF_CFG	0x01	Read the Battery Level Characteristic Client Characteristic Configuration Descriptor.
BASC_BATT_LVL_PRES_FORMAT	0x02	Read the Battery Level Characteristic Presentation Format Descriptor.

Table 3.1 – Peer battery info that can be read

3.3.2.2 BASC_READ_INFO_RSP

Source: TASK_APP

Destination: TASK_BASC_IDX

Parameters:

Type	Parameters	Description
uint8_t	status	Status code (see [3])
uint8_t	info	Characteristic info (see Table 3.1)
uint8_t	bas_nb	BAS instance.
union basc_data	data	Information data (see Table 3.2)

Description: Message received when peer device info has been read

Type	Parameters	Description
uint16_t	ntf_cfg	Notification Configuration Value (if info = BASC_NTF_CFG)
uint8_t	batt_level	Battery Level (if info = BASC_BATT_LVL_VAL)
struct prf_char_pres_fmt	char_pres_format	Characteristic Presentation Format (see Table 4. 1). (if info = BASC_BATT_LVL_PRES_FORMAT)

Table 3.2 – Read information Data

3.3.2.3 BASC_BATT_LEVEL_IND

Source: TASK_BASC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	batt_level	Battery Level value.
uint8_t	bas_nb	BAS Instance.

Description: This API message is sent to the application when a Battery Level Characteristic value has been received either upon reception of a notification, or upon reception of the read response.

3.3.2.4 BASC_BATT_LEVEL_NTF_CFG_REQ

Source: TASK_APP

Destination: TASK_BASC_IDX

Required state: IDLE/BUSY

Parameters:

Type	Parameters	Description
uint16_t	ntf_cfg	Notification Configuration Value
uint8_t	bas_nb	Battery Service instance.

Response: BASC_BATT_LEVEL_NTF_CFG_RSP

Description: This API message shall be used to set the notification configuration for specific battery service instance.

3.3.2.5   BASC_BATT_LEVEL_NTF_CFG_RSP

Source: TASK_BASC_IDX

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Write request status code (see [3])
uint8_t	bas_nb	Battery Service instance.

Description: This API message is sent to the application when notification response has been received from the peer device after sending of a write request.

4 Miscellaneous

Type	Parameters	Description
uint16_t	unit	Unit (The unit is a UUID – defined in attm.h). The Battery Level value is a percentage, thus this value should be set to ATT_UNIT_PERCENTAGE (0x27AD).
uint16_t	description	The description is an enumerated value from the organization identified by the namespace field.
uint8_t	format	Format. This value shall be set to 4 (unsigned 8-bit integer).
uint8_t	exponent	Exponent.
uint8_t	namespace	The namespace field is used to identify the organization that is responsible for defining the enumerations for the description field. Currently, the only authorized value is 1 (Bluetooth SIG Assigned Numbers).

Table 4. 1 – Characteristic Presentation Format Descriptor value structure (struct prf_char_pres_fmt)

Type	Parameters	Description
struct prf_svc	svc	Start handle and End handle of the BAS Service (see Table 4. 3).
struct prf_char_inf	chars[BAS_CHAR_MAX]	Informations about the found Characteristics (see Table 4. 4). *BAS_CHAR_MAX = 1* *chars[BAS_CHAR_BATT_LEVEL] = Battery Level Characteristic.*
struct prf_char_desc_inf	descs[BAS_DESC_MAX]	Informations about the found Descriptors (see Table Table 4. 5). *BAS_DESC_MAX = 2* *descs[BAS_DESC_BATT_LEVEL_PRES_FORMAT] = Characteristic Presentation Format Descriptor* *descs[BAS_DESC_BATT_LEVEL_CFG] = Client Characteristic Configuration Descriptor*

Table 4. 2 – Battery Service Content structure (struct bas_content)

Type	Parameters	Description
uint16_t	shdl	Start handle of the Service.
uint16_t	ehdl	End handle of the Service.

Table 4. 3 – Service description structure (struct prf_svc)

Type	Parameters	Description
uint16_t	char_hdl	Characteristic declaration attribute handle.
uint16_t	val_hdl	Characteristic value attribute handle.
uint8_t	prop	Properties
uint8_t	char_ehdl_off	Number of attribute within the Characteristic.

Table 4. 4 – Characteristic description structure (struct prf_char_inf)

Type	Parameters	Description
uint16_t	desc_hdl	Descriptor attribute handle

Table 4. 5 – Descriptor description structure (struct prf_char_desc_inf)

5 Abbreviations

Abbreviation	Original Terminology
API	Application Programming Interface
BAS	Battery Service
BLE	Bluetooth Low Energy
GAP	Generic Access Profile
GATT	Generic Attribute Profile
RW	RivieraWaves

6 References

[1]	Title	BATTERY SERVICE SPECIFICATION		
	Reference	BAS_SPEC_V10		
	Version	V10r00	Date	2011-12-27
	Source	Bluetooth SIG		

[2]	Title	BATTERY TEST SPECIFICATION		
	Reference	BAS.TS.1.0.3		
	Version	1.0.3	Date	2016-12-13
	Source	Bluetooth SIG		

[3]	Title	RW BLE Host Error Code Interface Specification		
	Reference	RW-BLE-HOST-ERR-CODE-IS		
	Version	9.1	Date	2018-10-24
	Source	RivieraWaves SAS		

[4]	Title	GAP Interface Specification		
	Reference	RW-BLE-GAP-IS		
	Version	9.15	Date	2019-04-02
	Source	RivieraWaves SAS		