User's Manual

Bluetooth® Low Energy Protocol Stack

API Reference Manual: HOGP

Renesas MCU

Target Device

RL78/G1D

Renesas Electronics
www.renesas.com

Rev.1.04 Apr 2015

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(2012.4)

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. Purpose and Target Readers

This manual describes the API (Application Program Interface) of the HID over GATT profile (HOGP) of the Bluetooth Low Energy protocol stack (BLE software), which is used to develop Bluetooth applications that incorporate the Renesas Bluetooth low energy microcontroller RL78/G1D. It is intended for users designing application systems incorporating this software. A basic knowledge of microcontrollers and Bluetooth low energy is necessary in order to use this manual.

Related documents

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Document Name	Document No.
Bluetooth Low Energy Protocol Stack	
User's Manual	R01UW0095E
API Reference Manual: Basics	R01UW0088E
API Reference Manual: FMP	R01UW0089E
API Reference Manual: PXP	R01UW0090E
API Reference Manual: HTP	R01UW0091E
API Reference Manual: BLP	R01UW0092E
API Reference Manual: HOGP	This manual
API Reference Manual: ScPP	R01UW0094E
API Reference Manual: HRP	R01UW0097E
API Reference Manual: CSCP	R01UW0098E
API Reference Manual: CPP	R01UW0099E
API Reference Manual: GLP	R01UW0103E
API Reference Manual: TIP	R01UW0106E
API Reference Manual: RSCP	R01UW0107E
API Reference Manual: ANP	R01UW0108E
API Reference Manual: PASP	R01UW0109E
API Reference Manual: LNP	R01UW0113E
Application Note: Sample Program	R01AN1375E
Application Note: rBLE Command Specification	R01AN1376E

List of Abbreviations and Acronyms

Abbreviation	Full Form	Remark
ANP	Alert Notification Profile	
ANS	Alert Notification Service	
API	Application Programming Interface	
ATT	Attribute Protocol	
BAS	Battery Service	
BB	Base Band	
BD_ADDR	Bluetooth Device Address	
BLE	Bluetooth low energy	
BLP	Blood Pressure Profile	
BLS	Blood Pressure Service	
CPP	Cycling Power Profile	
CPS	Cycling Power Service	
CSCP	Cycling Speed and Cadence Profile	
CSCS	Cycling Speed and Cadence Service	
CSRK	Connection Signature Resolving Key	
CTS	Current Time Service	
DIS	Device Information Service	
EDIV	Encrypted Diversifier	
FMP	Find Me Profile	
GAP	Generic Access Profile	
GATT	Generic Attribute Profile	
GLP	Glucose Profile	
GLS	Glucose Service	
HCI	Host Controller Interface	
HID	Human Interface Device	
HIDS	HID Service	
HOGP	HID over GATT Profile	
HRP	Heart Rate Profile	
HRS	Heart Rate Service	
HTP	Health Thermometer Profile	
HTS	Health Thermometer Service	
IAS	Immediate Alert Service	
IRK	Identity Resolving Key	
L2CAP	Logical Link Control and Adaptation Protocol	
LE	Low Energy	

Abbreviation	Full Form	Remark
LL	Link Layer	
LLS	Link Loss Service	
LNP	Location and Navigation Profile	
LNS	Location and Navigation Service	
LTK	Long Term Key	
MCU	Micro Controller Unit	
MITM	Man-in-the-middle	
MTU	Maximum Transmission Unit	
NDCS	Next DST Change Service	
OOB	Out of Band	
OS	Operating System	
PASP	Phone Alert Status Profile	
PASS	Phone Alert Status Service	
PXP	Proximity Profile	
RF	Radio Frequency	
RSCP	Running Speed and Cadence Profile	
RSCS	Running Speed and Cadence Service	
RSSI	Received Signal Strength Indication	
RTUS	Reference Time Update Service	
ScPP	Scan Parameters Profile	
ScPS	Scan Parameters Service	
SM	Security Manager	
SMP	Security Manager Protocol	
STK	Short Term Key	
TIP	Time Profile	
TK	Temporary Key	
TPS	Tx Power Service	
UART	Universal Asynchronous Receiver Transmitter	
UUID	Universal Unique Identifier	

Abbreviation	Full Form	Remark
APP	Application	
CSI	Clocked Serial Interface	
IIC	Inter-Integrated Circuit	
RSCIP	Renesas Serial Communication Interface Protocol	
VS	Vendor Specific	

Contents

1. Overview

This manual describes the API (Application Program Interface) of the HID over GATT profile (HOGP) of the Bluetooth Low Energy protocol stack (BLE software), which is used to develop Bluetooth applications that incorporate Renesas Bluetooth low energy microcontroller RL78/G1D.

For details about the organization and features of BLE software, see the Bluetooth Low Energy Protocol Stack User's Manual.

2. Common Definitions

This section describes the definitions common to the API of each profile.

2.1 Service Definitions

This section describes the common definitions of services used by the API of multiple profiles.

- Declaration of enumerated type for alert level

```
enum RBLE_SVC_ALT_LVL_enum {
   RBLE_SVC_ALERT_NONE = 0x00,            No alert
   RBLE_SVC_ALERT_MILD,                   Mild alert
   RBLE_SVC_ALERT_HIGH                    High alert
};
```

- Declaration of enumerated type for PnP ID characteristic vendor ID field

```
enum RBLE_SVC_PNP_VENDOR_ID_enum {
   RBLE_SVC_SIG_ASSIGNED_ID = 0x01,       Vendor ID assigned by Bluetooth SIG
   RBLE_SVC_USB_ASSIGNED_ID               Vendor ID assigned by USB Implementer's
                                          Forum
};
```

- Declaration of enumerated type for Name Space field of Characteristic Presentation Format descriptor

```
enum RBLE_SVC_PRESEN_NAMESPASE_enum {
   RBLE_SVC_NAMESPACE_SIG = 0x01,         Defined by Bluetooth SIG
};
```

- Declaration of enumerated type for security level of Service

```
enum RBLE_SVC_SEC_LVL_enum {
   RBLE_SVC_SEC_NONE = 0x01,              No security
   RBLE_SVC_SEC_UNAUTH = 0x02,            Require unauthenticated pairing
   RBLE_SVC_SEC_AUTH = 0x04,              Require authenticated pairing
   RBLE_SVC_SEC_AUTZ = 0x08,              Require authorization
   RBLE_SVC_SEC_ENC = 0x10                Require encryption
};
```

- Declaration of enumerated type for connection types

```
enum RBLE_PRF_CON_enum {
    RBLE_PRF_CON_DISCOVERY = 0x00,        Configuration connection performed
                                          when connecting for the first time
    RBLE_PRF_CON_NORMAL                   Normal connection performed when
                                          connecting for the second and
                                          subsequent times
};
```

- Declaration of enumerated type for client configuration characteristic value

```
enum RBLE_PRF_CLIENT_CONFIG_enum {
    RBLE_PRF_STOP_NTFIND = 0x00,            Stop notification or indication of
                                            characteristic value.
    RBLE_PRF_START_NTF,                     Start notification of
                                            characteristic value.
    RBLE_PRF_START_IND                      Start indication of
                                            characteristic value.
};
```

- Declaration of enumerated type for server configuration characteristic value

```
enum RBLE_PRF_SERVER_CONFIG_enum {
    RBLE_PRF_STOP_BRD = 0x00,               Stop broadcast of characteristic value.
    RBLE_PRF_START_BRD                      Start broadcast of characteristic value.
};
```

2.2 Status Definitions

This section describes the status definitions used by the API of each profile.

- Declaration of enumerated type for rBLE status

```
enum RBLE_STATUS_enum {
    RBLE_OK = 0x00,                          Normal operation
    RBLE_PRF_ERR_INVALID_PARAM = 0x90,       Invalid parameter specified for
                                             setting or acquiring a characteristic
                                             value
    RBLE_PRF_ERR_INEXISTENT_HDL,             Invalid handle specified for setting
                                             or acquiring a characteristic value
    RBLE_PRF_ERR_STOP_DISC_CHAR_MISSING,     The characteristic value is missing.
    RBLE_PRF_ERR_MULTIPLE_IAS,               Multiple IASs exist.
    RBLE_PRF_ERR_INCORRECT_PROP,             Incorrect property
    RBLE_PRF_ERR_MULTIPLE_CHAR,              Multiple characteristic values exist.
    RBLE_PRF_ERR_NOT_WRITABLE,               Writing is not permitted.
    RBLE_PRF_ERR_NOT_READABLE,               Reading is not permitted.
    RBLE_PRF_ERR_REQ_DISALLOWED,             Requesting is not permitted.
    RBLE_PRF_ERR_NTF_DISABLED,               Notification is disabled.
    RBLE_PRF_ERR_IND_DISABLED,               Indication is disabled.
    RBLE_PRF_ERR_ATT_NOT_SUPPORTED,          The characteristic value is not
                                             supported.
};
```

Note: Statuses other than the above are described in *API Reference Manual: Basics*.

3. HID over GATT Profile

This section describes the API of the HID over GATT profile. The HID over GATT profile adapts the USB HID specification for operating over a Bluetooth low energy wireless link to allow data communications between an HID Device and HID Host.

3.1 Definitions

This section describes the definitions used by the API of the HID over GATT profile.

- Declaration of maximum number of HID service instances

```
#define RBLE_HIDS_INST_MAX                0x02
```

- Declaration of maximum number of Battery service instances

```
#define RBLE_BAS_INST_MAX                 0x02
```

- Declaration of maximum HID report size

```
#define RBLE_HIDS_REPORT_MAX              0x20
```

- Declaration of enumerated type for HOGP event types

```
enum RBLE_HGP_EVENT_TYPE_enum {
    RBLE_HGP_EVENT_HDEVICE_ENABLE_COMP = 0x01,      HID Device enable completion event
                                                    (Parameter: hdevice_enable)
    RBLE_HGP_EVENT_HDEVICE_DISABLE_COMP,            HID Device disable completion
                                                    event
                                                    (Parameter: hdevice_disable)
    RBLE_HGP_EVENT_HDEVICE_ERROR_IND,               HID Device error indication event
                                                    (Parameter: error_ind)
    RBLE_HGP_EVENT_HDEVICE_CFG_INDNTF_IND,          Configured value change indication
                                                    event
                                                    (Parameter: hghd_cfg_indntf_ind)
    RBLE_HGP_EVENT_HDEVICE_REPORT_IND,              Report value setting indication
                                                    event
                                                    (Parameter: report_chg_ind)
    RBLE_HGP_EVENT_HDEVICE_PROTOCOL_MODE_CHG_EVT,
                                                    Protocol mode change notification
                                                    event
                                                    (Parameter: protocol_mode_chg_evt)
    RBLE_HGP_EVENT_HDEVICE_REPORT_EVT,              Report value notification event
                                                    (Parameter: report_chg_evt)
    RBLE_HGP_EVENT_HDEVICE_HID_CP_CHG_EVT,          Control Point change notification
                                                    event
                                                    (Parameter: hid_cp_chg_evt)
    RBLE_HGP_EVENT_HDEVICE_REPORT_COMP,             Report value send completion event
                                                    (Parameter: send_report)
```

RBLE_HGP_EVENT_HDEVICE_SEND_BATTERY_LEVEL_COMP,	Battery Level send completion event (Parameter: send_battery_level)
RBLE_HGP_EVENT_HDEVICE_COMMAND_DISALLOWED_IND,	Command disallowed indication event (Parameter: cmd_disallowed_ind)
RBLE_HGP_EVENT_BHOST_ENABLE_COMP = 0x81,	Boot Host enable completion event (Parameter: bhost_enable)
RBLE_HGP_EVENT_BHOST_DISABLE_COMP,	Boot Host disable completion event (Parameter: bhost_disable)
RBLE_HGP_EVENT_BHOST_ERROR_IND,	Boot Host error indication event (Parameter: error_ind)
RBLE_HGP_EVENT_BHOST_READ_CHAR_RESPONSE,	Characteristic read request response event (Parameter: rd_char_resp)
RBLE_HGP_EVENT_BHOST_WRITE_CHAR_RESPONSE,	Characteristic write request response event (Parameter: wr_char_resp)
RBLE_HGP_EVENT_BHOST_REPORT_NTF,	Report value notification event (Parameter: report_ntf)
RBLE_HGP_EVENT_BHOST_COMMAND_DISALLOWED_IND,	Command disallowed indication event (Parameter: cmd_disallowed_ind)
RBLE_HGP_EVENT_RHOST_ENABLE_COMP = 0xC1,	Report Host enable completion event (Parameter: rhost_enable)
RBLE_HGP_EVENT_RHOST_DISABLE_COMP,	Report Host disable completion event (Parameter: rhost_disable)
RBLE_HGP_EVENT_RHOST_ERROR_IND,	Report Host error indication event (Parameter: error_ind)
RBLE_HGP_EVENT_RHOST_READ_CHAR_RESPONSE,	Characteristic read request response event (Parameter: rd_char_resp)
RBLE_HGP_EVENT_RHOST_READ_LONG_CHAR_RESPONSE,	Long characteristic read request response event (Parameter: rd_long_char_resp)
RBLE_HGP_EVENT_RHOST_WRITE_CHAR_RESPONSE,	Characteristic write request response event (Parameter: wr_char_resp)
RBLE_HGP_EVENT_RHOST_REPORT_NTF,	Report value notification event (Parameter: report_ntf)
RBLE_HGP_EVENT_RHOST_BATTERY_LEVEL_NTF,	Battery Level notification event (Parameter: battery_level_ntf)

```
    RBLE_HGP_EVENT_RHOST_COMMAND_DISALLOWED_IND
                                                    Command disallowed indication
                                                    event
                                                    (Parameter: cmd_disallowed_ind)
};
```

- Declaration of data type for HOGP event types

```
typedef uint8_t RBLE_HGP_EVENT_TYPE;
```

- Declaration of data type for HOGP HID Device event callback function

```
typedef void ( *RBLE_HGHD_EVENT_HANDLER )( RBLE_HGHD_EVENT *event );
```

- Declaration of data type for HOGP Boot Host event callback function

```
typedef void ( *RBLE_HGBH_EVENT_HANDLER )( RBLE_HGBH_EVENT *event );
```

- Declaration of data type for HOGP Report Host event callback function

```
typedef void ( *RBLE_HGRH_EVENT_HANDLER )( RBLE_HGRH_EVENT *event );
```

- Declaration of enumerated type for HOGP device types

```
enum RBLE_HGHD_DEVICE_TYPE_enum{
    RBLE_HGHD_HID_DEVICE = 0x01,                    HID Device
    RBLE_HGHD_BOOT_KEYBOARD,                        Boot Keyboard
    RBLE_HGHD_BOOT_MOUSE                            Boot Mouse
};
```

- Declaration of enumerated type for HOGP characteristic value writing codes

```
enum RBLE_HGHD_WR_CHAR_CODE_enum {
    RBLE_HGHD_REPORT_INPUT_CODE = 0x01,             Report (Input) characteristic
    RBLE_HGHD_KB_REPORT_CODE,                       Boot Keyboard Input Report
                                                    characteristic
    RBLE_HGHD_MO_REPORT_CODE,                       Boot Mouse Input Report
                                                    characteristic
    RBLE_HGHD_BATTERY_LEVEL_CODE                    Battery Level characteristic
};
```

- Declaration of enumerated type for HOGP protocol mode

```
enum RBLE_HGHD_PROTOCOL_MODE_enum {
    RBLE_HGHD_PROTOCOL_MODE_BOOT = 0x00,            Boot protocol mode
    RBLE_HGHD_PROTOCOL_MODE_REPORT                  Report protocol mode
};
```

- Declaration of enumerated type for HOGP report types

```
enum RBLE_HGHD_REPORT_REFERENCE_enum {
    RBLE_HGHD_INPUT_REPORT = 0x01,                  Input report type
    RBLE_HGHD_OUTPUT_REPORT,                        Output report type
    RBLE_HGHD_FEATURE_REPORT                        Feature report type
```

```
};
```

- Declaration of enumerated type for HOGP HID information characteristic flag fields

```
enum RBLE_HGHD_HID_INFORMATION_enum {
    RBLE_HGHD_FLAG_REMOTE_WAKE = 0x01,          Remote wakeup signal sendable
                                                flag
    RBLE_HGHD_FLAG_NORMALLY_CONNECTABLE         Normally connectable flag
};
```

- Declaration of enumerated type for HOGP HID Control Point types

```
enum RBLE_HGHD_CONTROL_POINT_enum {
    RBLE_HGHD_CTRL_POINT_SUSPEND = 0x00,            Suspend
    RBLE_HGHD_POINT_EXIT_SUSPEND                    Exit suspend
};
```

- Declaration of enumerated type for HOGP Boot Host HID/Device Information/Battery service characteristic codes

```
enum RBLE_HGBH_RD_CHAR_CODE_enum {
    RBLE_HGBH_RD_HIDS_PM = 0x00,        Protocol Mode characteristic
    RBLE_HGBH_RD_HIDS_KI,               Boot Keyboard Input Report characteristic
    RBLE_HGBH_RD_HIDS_KI_CFG,           Boot Keyboard Input Report
                                        Configuration descriptor
    RBLE_HGBH_RD_HIDS_KO,               Boot Keyboard Output Report characteristic
    RBLE_HGBH_RD_HIDS_MI,               Boot Mouse Input Report characteristic
    RBLE_HGBH_RD_HIDS_MI_CFG,           Boot Mouse Input Report
                                        Configuration descriptor
    RBLE_HGBH_RD_DIS_PNPID,             PnP ID characteristic
    RBLE_HGBH_RD_BAS_BL                 Battery Level characteristic
};
```

- Declaration of enumerated type for HOGP Report Host HID/Device Information/Battery service characteristic codes

```
enum RBLE_HGRH_RD_CHAR_CODE_enum {
    RBLE_HGRH_RD_HIDS_PM = 0x00,        Protocol Mode characteristic
    RBLE_HGRH_RD_HIDS_RI,               Report (Input) characteristic
    RBLE_HGRH_RD_HIDS_RI_CFG,           Report (Input) configuration descriptor
    RBLE_HGRH_RD_HIDS_RI_REF,           Report (Input) reference descriptor
    RBLE_HGRH_RD_HIDS_RO,               Report (Output) characteristic
    RBLE_HGRH_RD_HIDS_RO_REF,           Report (Output) reference descriptor
    RBLE_HGRH_RD_HIDS_RF,               Report (Feature) characteristic
    RBLE_HGRH_RD_HIDS_RF_REF,           Report (Feature) reference descriptor
    RBLE_HGRH_RD_HIDS_RM,               Report Map characteristic
    RBLE_HGRH_RD_HIDS_ER_REF,           Report Map external reference descriptor
    RBLE_HGRH_RD_HIDS_HI,               HID Information characteristic
    RBLE_HGRH_RD_DIS_PNPID,             PnP ID characteristic
    RBLE_HGRH_RD_BAS_BL,                Battery Level
    RBLE_HGRH_RD_BAS_BL_CFG,            Battery Level configuration descriptor
    RBLE_HGRH_RD_BAS_BL_REF,            Battery Level reference descriptor
```

```
};
```

- HID Device characteristic information structures

```
typedef struct RBLE_HGP_DEVICE_PARAM_t{
    uint8_t     hids_inst_num;              Number of HID service instances
    uint8_t     bas_inst_num;               Number of Battery service instances
    uint16_t    report_input_ntf_en[RBLE_HIDS_INST_MAX];
                                            Report (Input) notification configuration
                                            value
    uint16_t    kb_report_ntf_en[RBLE_HIDS_INST_MAX];
                                            Boot Keyboard Input Report notification
                                            configuration value
    uint16_t    mo_report_ntf_en[RBLE_HIDS_INST_MAX];
                                            Boot Mouse Input Report notification
                                            configuration value
    uint8_t     protocol_mode_val[RBLE_HIDS_INST_MAX];
                                            Protocol Mode characteristic value
#if ((RBLE_HIDS_INST_MAX % 2) != 0)
    uint8_t     reserved;  Reserved
#endif
    uint16_t    battery_level_ntf_en[RBLE_BAS_INST_MAX];
                                            Battery Level notification configuration
                                            value
}RBLE_HGP_DEVICE_PARAM;
```

- Report structure

```
typedef struct RBLE_HGP_REPORT_DESC_t{
    uint8_t     device_type;                        Device type
    uint8_t     report_type;                        Report type
    uint8_t     value[RBLE_HIDS_REPORT_MAX];        Report value
#if ((RBLE_HIDS_REPORT_MAX % 2) != 0)
    uint8_t     reserved;                           Reserved
#endif
    uint16_t    value_size;                         Report size
}RBLE_HGP_REPORT_DESC;
```

- HID service content structures

```
typedef struct RBLE_HIDS_CONTENT_t{
    uint16_t     shdl;                  HID service start handle
    uint16_t     ehdl;                  HID service end handle
    uint16_t     protocol_md_char_hdl;  Protocol Mode characteristic handle
    uint16_t     protocol_md_val_hdl;   Protocol Mode characteristic value
                                        handle
    uint8_t      protocol_md_prop;      Protocol Mode characteristic property
    uint8_t      reserved;              Reserved
    uint16_t     report_input_char_hdl; Report (Input) characteristic handle
    uint16_t     report_input_val_hdl;  Report (Input) characteristic value
```

		handle
uint16_t	report_input_cfg_hdl;	Report (Input) characteristic configuration descriptor handle
uint16_t	input_rep_ref_hdl;	Report (Input) reference descriptor handle
uint8_t	report_input_prop;	Report (Input) property
uint8_t	reserved1;	Reserved
uint16_t	report_output_char_hdl;	Report (Output) characteristic handle
uint16_t	report_output_val_hdl;	Report (Output) characteristic value handle
uint16_t	output_rep_ref_hdl;	Report (Output) reference descriptor handle
uint8_t	report_output_prop;	Report (Output) property
uint8_t	reserved2;	Reserved
uint16_t	report_feature_char_hdl;	Report (Feature) characteristic handle
uint16_t	report_feature_val_hdl;	Report (Feature) characteristic value handle
uint16_t	feature_rep_ref_hdl;	Report (Feature) reference descriptor handle
uint8_t	report_feature_prop;	Report (Feature) property
uint8_t	reserved3;	Reserved
uint16_t	report_map_char_hdl;	Report Map characteristic handle
uint16_t	report_map_val_hdl;	Report Map characteristic value handle
uint16_t	external_rep_ref_hdl;	Report Map external reference descriptor handle
uint8_t	report_map_prop;	Report Map property
uint8_t	reserved4;	Reserved
uint16_t	bootkb_input_char_hdl;	Boot Keyboard Input Report characteristic handle
uint16_t	bootkb_input_val_hdl;	Boot Keyboard Input Report characteristic value handle
uint16_t	bootkb_input_cfg_hdl;	Boot Keyboard Input Report characteristic configuration descriptor handle
uint8_t	bootkb_input_prop;	Boot Keyboard Input Report property
uint8_t	reserved5;	Reserved
uint16_t	bootkb_output_char_hdl;	Boot Keyboard Output Report characteristic handle
uint16_t	bootkb_output_val_hdl;	Boot Keyboard Output Report characteristic value handle
uint8_t	bootkb_output_prop;	Boot Keyboard Output Report property
uint8_t	reserved6;	Reserved
uint16_t	bootmo_input_char_hdl;	Boot Mouse Input Report characteristic handle
uint16_t	bootmo_input_val_hdl;	Boot Mouse Input Report characteristic value handle
uint16_t	bootmo_input_cfg_hdl;	Boot Mouse Input Report characteristic configuration descriptor handle
uint8_t	bootmo_input_prop;	Boot Mouse Input Report property
uint8_t	reserved7;	Reserved

```
    uint16_t     hid_info_char_hdl;          HID Information characteristic handle
    uint16_t     hid_info_val_hdl;           HID Information characteristic value
                                             handle
    uint8_t      hid_info_prop;              HID Information property
    uint8_t      reserved8;                  Reserved
    uint16_t     hid_cp_char_hdl;            HID Control Point characteristic handle
    uint16_t     hid_cp_val_hdl;             HID Control Point characteristic value
                                             handle
    uint8_t      hid_cp_prop;                HID Control Point property
    uint8_t      reserved9;                  Reserved
    uint16_t     include_svc_hdl;            Included service handle
    uint16_t     include_svc_uuid;           Included service UUID
    uint16_t     incl_shdl;                  Included service start handle
    uint16_t     incl_ehdl;                  Included service end handle
}RBLE_HIDS_CONTENT;
```

- Device Information service content structures

```
typedef struct RBLE_DIS11_CONTENT_t{
    uint16_t     shdl;                       Device Information service start handle
    uint16_t     ehdl;                       Device Information service end handle
    uint16_t     pnp_id_char_hdl;            PnP ID characteristic handle
    uint16_t     pnp_id_val_hdl;             PnP ID characteristic value handle
    uint8_t      pnp_id_prop;                PnP ID property
    uint8_t      reserved;                   Reserved
}RBLE_DIS11_CONTENT;
```

- Battery service content structures

```
typedef struct RBLE_BAS_CONTENT_t{
    uint16_t     shdl;                       Battery service start handle
    uint16_t     ehdl;                       Battery service end handle
    uint16_t     battery_lvl_char_hdl;       Battery Level characteristic handle
    uint16_t     battery_lvl_val_hdl;        Battery Level characteristic value
                                             handle
    uint16_t     battery_lvl_cfg_hdl;        Battery Level characteristic
                                             configuration descriptor handle
    uint16_t     battery_lvl_rep_ref_hdl;    Battery Level reference descriptor handle
    uint8_t      battery_lvl_prop;           Battery Level property
    uint8_t      reserved;                   Reserved
}RBLE_BAS_CONTENT;
```

- HID Device event parameter structures

```
typedef struct RBLE_HGHD_EVENT_t {
    RBLE_HGP_EVENT_TYPE              type;              HOGP event type
    uint8_t                          reserved;          Reserved
    union Event_Hghd_Parameter_u {
        Generic event
        RBLE_STATUS                  status;            Status
```

HID Device enable completion event

```
struct RBLE_HGP_HDevice_Enable_t{
   uint16_t                  conhdl;          Connection handle
   RBLE_STATUS               status;          Status
   uint8_t                   reserved;        Reserved
}hdevice_enable;
```

HID Device disable completion event

```
struct RBLE_HGP_HDevice_Disable_t{
   uint16_t                  conhdl;          Connection handle
   RBLE_STATUS               status;          Status
   uint8_t                   reserved;        Reserved
   RBLE_HGP_DEVICE_PARAM     device_info;     HID Device characteristic
                                              information
}hdevice_disable;
```

HID Device error indication event

```
struct RBLE_HGP_HDevice_Error_Ind_t{
   uint16_t                  conhdl;          Connection handle
   RBLE_STATUS               status;          Status
   uint8_t                   reserved;        Reserved
}error_ind;
```

Configured value change indication event

```
struct RBLE_HGP_HDevice_Cfg_Indntf_Ind_t{
   uint16_t                  conhdl;          Connection handle
   uint8_t                   inst_idx;        Instance index
   uint8_t                   char_code;       Characteristic code
   uint16_t                  cfg_val;         Configuration value
}hghd_cfg_indntf_ind;
```

Report value setting indication event

```
struct RBLE_HGP_HDevice_Report_Ind_t{
   uint16_t                  conhdl;          Connection handle
   uint8_t                   inst_idx;        Instance index
   uint8_t                   reserved;        Reserved
   RBLE_HGP_REPORT_DESC      report;          Report
}report_chg_ind;
```

Protocol Mode change notification event

```
struct RBLE_HGP_HDevice_Protocol_Mode_Chg_Evt_t{
   uint16_t                  conhdl;             Connection handle
   uint8_t                   inst_idx;           Instance index
   uint8_t                   protocol_mode_val;  Protocol Mode value
}protocol_mode_chg_evt;
```

```
        Report value notification event
        struct RBLE_HGP_HDevice_Report_Evt_t{
           uint16_t                  conhdl;              Connection handle
           uint8_t                   inst_idx;            Instance index
           uint8_t                   reserved;            Reserved
           RBLE_HGP_REPORT_DESC      report;              Report
        }report_chg_evt;

        Control Point change notification event
        struct RBLE_HGP_HDevice_Hid_Cp_Chg_Evt_t{
           uint16_t                  conhdl;              Connection handle
           uint8_t                   inst_idx;            Instance index
           uint8_t                   control_point_val;   Control Point value
        }hid_cp_chg_evt;

        Report value send completion event
        struct RBLE_HGP_HDevice_Send_Report_t{
           uint16_t                  conhdl;              Connection handle
           RBLE_STATUS               status;              Status
           uint8_t                   reserved;            Reserved
        }send_report;

        Battery Level send completion event
        struct RBLE_HGP_HDevice_Send_Battery_Level_t{
           uint16_t                  conhdl;              Connection handle
           RBLE_STATUS               status;              Status
           uint8_t                   reserved;            Reserved
        }send_battery_level;

        Command disallowed indication event
        struct RBLE_HGP_HDevice_Command_Disallowed_Ind_t{
           RBLE_STATUS               status;              Connection handle
           uint8_t                   reserved;            Reserved
           uint16_t                  opcode;              Opcode
        }cmd_disallowed_ind;
    }param;
}RBLE_HGHD_EVENT;
```

- Boot Host event parameter structures

```
typedef struct RBLE_HGBH_EVENT_t {
    RBLE_HGP_EVENT_TYPE              type;              Event type
    uint8_t                          reserved;          Reserved
    union Event_Hgbh_Parameter_u {
        Generic event
        RBLE_STATUS                  status;            Status

        Boot Host enable completion event
        struct RBLE_HGP_BHost_Enable_t{
           uint16_t                  conhdl;            Connection handle
           RBLE_STATUS               status;            Status
           uint8_t                   hids_inst_num;     Number of HID service
                                                        instances
           uint8_t                   bas_inst_num;      Number of Battery service
                                                        instances
           uint8_t                   reserved;          Reserved
           RBLE_HIDS_CONTENT         hids[RBLE_HIDS_INST_MAX];
                                                        HID service content
           RBLE_DIS11_CONTENT        dis;               Device Information service
                                                        content
           RBLE_BAS_CONTENT          bas[RBLE_BAS_INST_MAX];
                                                        Battery service content
        }bhost_enable;

        Boot Host disable completion event
        struct RBLE_HGP_BHost_Disable_t{
           uint16_t                  conhdl;            Connection handle
           RBLE_STATUS               status;            Status
           uint8_t                   reserved;          Reserved
        }bhost_disable;

        Boot Host error indication event
        struct RBLE_HGP_BHost_Error_Ind_t{
           uint16_t                  conhdl;            Connection handle
           RBLE_STATUS               status;            Status
           uint8_t                   reserved;          Reserved
        }error_ind;

        Report value notification event
        struct RBLE_HGP_BHost_Report_Ntf_t{
           uint16_t                  conhdl;            Connection handle
           uint8_t                   inst_idx;          Instance index
           uint8_t                   reserved;          Reserved
           RBLE_HGP_REPORT_DESC      report;            Report
        }report_ntf;
```

```
        Characteristic read request response event
        struct RBLE_HGP_BHost_Read_Char_Response_t{
           uint16_t                  conhdl;          Connection handle
           uint8_t                   att_code;        Characteristic acquisition
                                                      result
           uint8_t                   reserved;        Reserved
           RBLE_ATT_INFO_DATA data;                   Acquired characteristic data
        }rd_char_resp;

        Characteristic write request response event
        struct RBLE_HGP_BHost_Write_Char_Response_t{
           uint16_t                  conhdl;          Connection handle
           uint8_t                   att_code;        Characteristic write result
           uint8_t                   reserved;        Reserved
        }wr_char_resp;

        Command disallowed indication event
        struct RBLE_HGP_BHost_Command_Disallowed_Ind_t{
           RBLE_STATUS               status;          Status
           uint8_t                   reserved;        Reserved
           uint16_t                  opcode;          Opcode
        }cmd_disallowed_ind;
    }param;
}RBLE_HGBH_EVENT;
```

- Report Host event parameter structures

```
typedef struct RBLE_HGRH_EVENT_t {
    RBLE_HGP_EVENT_TYPE              type;              Event type
    uint8_t                          reserved;          Reserved
    union Event_Hgrh_Parameter_u {
        Generic event
        RBLE_STATUS                  status;            Status

        Report Host enable completion event
        struct RBLE_HGP_RHost_Enable_t{
           RBLE_STATUS               status;            Status
           uint8_t                   reserved;          Reserved
           uint16_t                  conhdl;            Connection handle
           uint8_t                   hids_inst_num;     Number of HID service
                                                        instances
           uint8_t                   bas_inst_num;      Number of Battery service
                                                        instances
           RBLE_HIDS_CONTENT         hids[RBLE_HIDS_INST_MAX];
                                                        HID service content
           RBLE_DIS11_CONTENT        dis;               Device Information service
                                                        content
           RBLE_BAS_CONTENT          bas[RBLE_BAS_INST_MAX];
                                                        Battery service content
        }rhost_enable;

        Report Host disable completion event
        struct RBLE_HGP_RHost_Disable_t{
           RBLE_STATUS               status;            Status
           uint8_t                   reserved;          Reserved
           uint16_t                  conhdl;            Connection handle
        }rhost_disable;

        Report Host error indication event
        struct RBLE_HGP_RHost_Error_Ind_t{
           RBLE_STATUS               status;            Status
           uint8_t                   reserved;          Reserved
           uint16_t                  conhdl;            Connection handle
        }error_ind;

        Report value notification event
        struct RBLE_HGP_RHost_Report_Ntf_t{
           uint16_t                  conhdl;            Connection handle
           uint8_t                   inst_idx;          Instance index
           uint8_t                   reserved;          Reserved
           RBLE_HGP_REPORT_DESC      report;            Report
        }report_ntf;
```

Battery Level notification event

```
struct RBLE_HGP_RHost_Battery_Level_Ntf_t{
   uint16_t                  conhdl;          Connection handle
   uint8_t                   inst_idx;        Instance index
   uint8_t                   battery_level;   Battery Level
}battery_level_ntf;
```

Characteristic read request response event

```
struct RBLE_HGP_RHost_Read_Char_Response_t{
   uint16_t                  conhdl;          Connection handle
   uint8_t                   att_code;        Characteristic acquisition
                                              result
   uint8_t                   reserved;        Reserved
   RBLE_ATT_INFO_DATA        data;            Acquired characteristic data
}rd_char_resp;
```

Long characteristic read request response event

```
struct RBLE_HGP_RHost_Read_Long_Char_Response_t{
   uint16_t                  conhdl;          Connection handle
   uint8_t                   att_code;        Characteristic acquisition
                                              result
   uint8_t                   reserved;        Reserved
   RBLE_ATT_INFO_LDATA       data;            Acquired characteristic data
}rd_long_char_resp;
```

Characteristic write request response event

```
struct RBLE_HGP_RHost_Write_Char_Response_t{
   uint16_t                  conhdl;          Connection handle
   uint8_t                   att_code;        Characteristic write result
   uint8_t                   reserved;        Reserved
}wr_char_resp;
```

Command disallowed indication event

```
struct RBLE_HGP_RHost_Command_Disallowed_Ind_t{
   RBLE_STATUS               status;          Status
   uint8_t                   reserved;        Reserved
   uint16_t                  opcode;          Opcode
}cmd_disallowed_ind;
  }param;
}RBLE_HGRH_EVENT;
```

3.2 Functions

The following table shows the API functions defined for the HOGP of rBLE and the following sections describe the API functions in detail.

Table 3-1 API Functions Used by the HOGP

RBLE_HGP_HDevice_Enable	Enables the HID Device.
RBLE_HGP_HDevice_Disable	Disables the HID Device.
RBLE_HGP_HDevice_Send_Report	Sends the Report.
RBLE_HGP_HDevice_Send_Battery_Level	Sends the Battery Level.
RBLE_HGP_BHost_Enable	Enables the Boot Host.
RBLE_HGP_BHost_Disable	Disables the Boot Host.
RBLE_HGP_BHost_Read_Char	Reads the characteristic value.
RBLE_HGP_BHost_Read_By_UUID_Char	Reads the characteristic value specified by UUID.
RBLE_HGP_BHost_Write_Char	Writes the characteristic value.
RBLE_HGP_BHost_Set_Report	Sets the Report value.
RBLE_HGP_BHost_Write_Protocol_Mode	Sends the Protocol Mode.
RBLE_HGP_BHost_Data_Output	Sends the Report value.
RBLE_HGP_RHost_Enable	Enables the Report Host.
RBLE_HGP_RHost_Disable	Disables the Report Host.
RBLE_HGP_RHost_Read_Char	Reads the characteristic value.
RBLE_HGP_RHost_Read_By_UUID_Char	Reads the characteristic value specified by UUID.
RBLE_HGP_RHost_Read_Long_Char	Reads the long characteristic value.
RBLE_HGP_RHost_Write_Char	Writes the characteristic value.
RBLE_HGP_RHost_Set_Report	Sets the Report value.
RBLE_HGP_RHost_Write_Protocol_Mode	Sends the Protocol Mode.
RBLE_HGP_RHost_Data_Output	Sends the Report value.
RBLE_HGP_RHost_Write_Control_Point	Sends the Control Point.

3.2.1 RBLE_HGP_HDevice_Enable

RBLE_STATUS RBLE_HGP_HDevice_Enable(uint16_t conhdl, uint8_t sec_lvl, uint8_t con_type, RBLE_HGP_DEVICE_PARAM *param, RBLE_HGHD_EVENT_HANDLER call_back)

This function enables the HOGP HID Device role.

If the Report (Input) value, Boot Keyboard Input Report value, Boot Mouse Input Report value, and Battery Level notification settings, and the Protocol Mode setting have not been specified from the HID Host, set the notification setting parameter to 0 to configure the connection. If these settings have been specified from the HID Host, perform a normal connection in accordance with the value stored in the HID Device.

The result is reported by using the HID Device role enable completion event RBLE_HGP_EVENT_HDEVICE_ENABLE_COMP.

(Note)

The application is responsible to prepare the memory block pointed by param and keep it until RBLE_HGP_HDevice_Disable has been completed.

Parameters:

conhdl	Connection handle		
sec_lvl	Security level		
con_type	RBLE_PRF_CON_DISCOVERY	Configuration connection	
	RBLE_PRF_CON_NORMAL	Normal connection	
*param	*hids_inst_num*	Number of HID service instances	
	bas_inst_num	Number of Battery service instances	
	report_input_ntf_en [RBLE_HIDS_INST_MAX]	RBLE_PRF_STOP_NTFIND	Stop notification/ indication of Report (Input).
		RBLE_PRF_START_NTF	Start notification of Report (Input).
	kb_report_ntf_en [RBLE_HIDS_INST_MAX]	RBLE_PRF_STOP_NTFIND	Stop notification/ indication of Boot Keyboard Input Report.
		RBLE_PRF_START_NTF	Start notification of Boot Keyboard Input Report.
	mo_report_ntf_en [RBLE_HIDS_INST_MAX]	RBLE_PRF_STOP_NTFIND	Stop notification/ indication of Boot Mouse Input Report.
		RBLE_PRF_START_NTF	Start notification of Boot Mouse Input Report.
	protocol_mode_val [RBLE_HIDS_INST_MAX]	Protocol Mode set value	
	battery_level_ntf_en [RBLE_BAS_INST_MAX]	RBLE_PRF_STOP_NTFIND	Stop notification/ indication of the Battery Level.
		RBLE_PRF_START_NTF	Start notification of the Battery Level.
call_back	Specify the callback function that reports the HID Device role event.		

Return:

RBLE_OK	Success
RBLE_ERR	Error occurred in HID Device role enable processing
RBLE_PARAM_ERR	Invalid parameter
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.2 RBLE_HGP_HDevice_Disable

RBLE_STATUS RBLE_HGP_HDevice_Disable(uint16_t conhdl)		
This function disables the HOGP HID Device role. The result is reported by using the HID Device role disable completion event RBLE_HGP_EVENT_HDEVICE_DISABLE_COMP.		
Parameters:		
conhdl	Connection handle	
Return:		
RBLE_OK	Success	
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.3 RBLE_HGP_HDevice_Send_Report

RBLE_STATUS RBLE_HGP_HDevice_Send_Report(uint16_t conhdl, uint8_t inst_idx, RBLE_HGP_REPORT_DESC *report)			
This function sends the Report. Specify the service instance number in inst_idx and specify the Report (Input) value, Boot Keyboard Input Report value, or Boot Mouse Input Report value according to the report you want to send, in device_type. An error occurs if anything other than RBLE_HGHD_INPUT_REPORT is specified for report_type. The result is reported by using the Report value send completion event RBLE_HGP_EVENT_HDEVICE_REPORT_COMP. (Note) 1. The application is responsible to prepare the memory block pointed by report and keep it until the report value has been sent. 2. The upper limit of value_size is as follows. When the device_type is equal to RBLE_HGHD_HID_DEVICE, it is the MTU size. When the device_type is equal to RBLE_HGHD_BOOT_KEYBOARD or RBLE_HGHD_BOOT_MOUSE, it is 8 bytes.			
Parameters:			
conhdl	Connection handle		
inst_idx	Instance index		
**report*	*device_type*	RBLE_HGHD_HID_DEVICE	Sends Report (Input).
		RBLE_HGHD_BOOT_KEYBOARD	Sends Boot Keyboard Input Report.
		RBLE_HGHD_BOOT_MOUSE	Sends Boot Mouse Input Report.
	report_type	RBLE_HGHD_INPUT_REPORT	Specifies that the report is Input Report Type.
	value[RBLE_HIDS_REPORT_MAX]	Report value	
	value_size	Report size	
Return:			
RBLE_OK	Success		
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.		

3.2.4 RBLE_HGP_HDevice_Send_Battery_Level

RBLE_STATUS RBLE_HGP_HDevice_Send_Battery_Level(uint16_t conhdl, uint8_t inst_idx, uint8_t battery_level)		
This function sends the Battery Level. Specify the service instance number in inst_idx. The result is reported by using the Battery Level send completion event RBLE_HGP_EVENT_HDEVICE_SEND_BATTERY_LEVEL_COMP.		
Parameters:		
conhdl	Connection handle	
inst_idx	Instance index	
battery_level	Battery Level	
Return:		
RBLE_OK	Success	
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.5 RBLE_HGP_BHost_Enable

RBLE_STATUS RBLE_HGP_BHost_Enable(uint16_t conhdl, uint8_t con_type, uint8_t hids_inst_num, uint8_t bas_inst_num, RBLE_HIDS_CONTENT *hids, RBLE_DIS11_CONTENT *dis, RBLE_BAS_CONTENT *bas, RBLE_HGBH_EVENT_HANDLER call_back);		
This function enables the HOGP Boot Host role and starts access to the service exposed by the HID Device. The result is reported together with the service handle information by using the Boot Host enable completion event RBLE_HGP_EVENT_BHOST_ENABLE_COMP. When starting access to the service exposed by an HID Device for the first time, configure the connection (by using RBLE_PRF_CON_DISCOVERY) to allow the BLE software to start discovering the service for the HID Device. Save the service handle information that was reported by the Boot Host enable completion event. If the service handle information about the discovered service is saved and is used when a known HID Device is connected normally (by using RBLE_PRF_CON_NORMAL) for a second or subsequent time, detecting the service is skipped, which enables a high-speed access to the service. While the HOGP Boot Host role is enabled, the service exposed by only one HID Device is accessible. To connect more than one HID Devices at the same time and access the services exposed by each HID Device, repeat enable (by using RBLE_HGP_BHost_Enable) and disable (by using RBLE_HGP_BHost_Disable) of the HOGP Boot Host role in order to switch access to them. At that time, perform normal connection by using the connection handle (which was obtained when connecting to each HID Device) and the handle information (which was saved when starting access to the service for the first time) as parameters. (Note) The application is responsible to prepare the memory blocks pointed by hids and bas and keep them until RBLE_HGP_BHost_Disable has been completed.		
Parameters:		
conhdl	Connection handle	
con_type	RBLE_PRF_CON_DISCOVERY	Configuration connection performed when connecting for the first time
	RBLE_PRF_CON_NORMAL	Normal connection performed when connecting for the second and subsequent times
hids_inst_num	Number of HID service instances	
bas_inst_num	Number of Battery service instances	
**hids*	*shdl*	HID service start handle
	ehdl	HID service end handle
	protocol_md_char_hdl	Protocol Mode characteristic handle

		protocol_md_val_hdl	Protocol Mode characteristic value handle
		protocol_md_prop	Protocol Mode property
		report_input_char_hdl	Report (Input) characteristic handle
		report_input_val_hdl	Report (Input) characteristic value handle
		report_input_cfg_hdl	Report (Input) characteristic configuration descriptor handle
		input_rep_ref_hdl	Report (Input) characteristic reference descriptor handle
		report_input_prop	Report (Input) property
		report_output_char_hdl	Report (Output) characteristic handle
		report_output_val_hdl	Report (Output) characteristic value handle
		output_rep_ref_hdl	Report (Output) characteristic reference descriptor handle
		report_output_prop	Report (Output) property
		report_feature_char_hdl	Report (Feature) characteristic handle
		report_feature_val_hdl	Report (Feature) characteristic value handle
		feature_rep_ref_hdl	Report (Feature) characteristic reference descriptor handle
		report_feature_prop	Report (Feature) property
		report_map_char_hdl	Report Map characteristic handle
		report_map_val_hdl	Report Map characteristic value handle
		external_rep_ref_hdl	Report map characteristic external reference descriptor handle
		report_map_prop	Report Map property
		bootkb_input_char_hdl	Boot Keyboard Input Report characteristic handle
		bootkb_input_val_hdl	Boot Keyboard Input Report characteristic value handle
		bootkb_input_cfg_hdl	Boot Keyboard Input Report characteristic configuration descriptor handle
		bootkb_input_prop	Boot Keyboard Input Report property
		bootkb_output_char_hdl	Boot Keyboard Output Report characteristic handle
		bootkb_output_val_hdl	Boot Keyboard Output Report characteristic value handle
		bootkb_output_prop	Boot Keyboard Output Report property
		bootmo_input_char_hdl	Boot Mouse Input Report characteristic handle
		bootmo_input_val_hdl	Boot Mouse Input Report characteristic value handle
		bootmo_input_cfg_hdl	Boot Mouse Input Report characteristic configuration descriptor handle
		bootmo_input_prop	Boot Mouse Input Report property
		hid_info_char_hdl	HID Information characteristic handle
		hid_info_val_hdl	HID Information characteristic value handle
		hid_info_prop	HID Information property
		hid_cp_char_hdl	HID Control Point characteristic handle
		hid_cp_val_hdl	HID Control Point characteristic value handle
		hid_cp_prop	HID Control Point property
		include_svc_hdl	Included service handle

		include_svc_uuid	UUID of Included service
		incl_shdl	Included service start handle
		incl_ehdl	Included service end handle
	**dis*	*shdl*	Device Information service start handle
		ehdl	Device Information service end handle
		pnp_id_char_hdl	PnP ID characteristic handle
		pnp_id_val_hdl	PnP ID characteristic value handle
		pnp_id_prop	PnP ID property
	**bas*	*shdl*	Battery service start handle
		ehdl	Battery service end handle
		battery_lvl_char_hdl	Battery Level characteristic handle
		battery_lvl_val_hdl	Battery Level characteristic value handle
		battery_lvl_cfg_hdl	Battery Level characteristic configuration descriptor handle
		battery_lvl_rep_ref_hdl	Battery Level characteristic reference descriptor handle
		battery_lvl_prop	Battery Level property
	call_back	Specify the callback function that reports the Boot Host role event.	

Return:

RBLE_OK	Success
RBLE_ERR	Error occurred in HOGP Boot Host role enable processing
RBLE_PARAM_ERR	Invalid parameter
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.6 RBLE_HGP_BHost_Disable

RBLE_STATUS RBLE_HGP_BHost_Disable (uint16_t conhdl)

This function disables the HOGP Boot Host role and terminates the access to the service exposed by the HID Device.

The result is reported by using the Boot Host role disable completion event RBLE_HGP_EVENT_BHOST_DISABLE_COMP.

Parameters:

conhdl	Connection handle

Return:

RBLE_OK	Success
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.7 RBLE_HGP_BHost_Read_Char

RBLE_STATUS RBLE_HGP_BHost_Read_Char(uint16_t conhdl, uint8_t inst_idx, uint8_t char_code)			
This function reads the characteristic value of the HID service and the Device Information service. The result is reported by using the characteristic read request response event RBLE_HGP_EVENT_BHOST_READ_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	inst_idx	Instance index	
	char_code	RBLE_HGBH_RD_HIDS_PM	Protocol mode
		RBLE_HGBH_RD_HIDS_KI	Boot Keyboard Input Report value
		RBLE_HGBH_RD_HIDS_KI_CFG	Boot Keyboard Input Report value notification setting
		RBLE_HGBH_RD_HIDS_KO	Boot Keyboard Output Report value
		RBLE_HGBH_RD_HIDS_MI	Boot Mouse Input Report value
		RBLE_HGBH_RD_HIDS_MI_CFG	Boot Mouse Input Report value notification setting
		RBLE_HGBH_RD_DIS_PNPID	PnP ID of HID Device
		RBLE_HGBH_RD_BAS_BL	Battery Level
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.8 RBLE_HGP_BHost_Read_By_UUID_Char

RBLE_STATUS RBLE_HGP_BHost_Read_By_UUID_Char (uint16_t conhdl, uint8_t inst_idx, uint8_t char_code)			
This function reads the characteristic value of the HID service, Device Information, and Battery service specified by the UUID characteristic. The result is reported by using the characteristic read request response event RBLE_HGP_EVENT_BHOST_READ_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	inst_idx	Instance index	
	char_code	RBLE_HGBH_RD_HIDS_PM	Protocol Mode value
		RBLE_HGBH_RD_HIDS_KI	Boot Keyboard Input Report value
		RBLE_HGBH_RD_HIDS_KO	Boot Keyboard Output Report value
		RBLE_HGBH_RD_HIDS_MI	Boot Mouse Input Report value
		RBLE_HGBH_RD_DIS_PNPID	PnP ID of HID Device
		RBLE_HGBH_RD_BAS_BL	Battery Level
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.9 RBLE_HGP_BHost_Write_Char

RBLE_STATUS RBLE_HGP_BHost_Write_Char(uint16_t conhdl, uint8_t inst_idx, uint8_t char_code, uint16_t cfg_val)			
This function writes each characteristic configuration descriptor of the HID service. The result is reported by using the characteristic write request response event RBLE_HGP_EVENT_BHOST_WRITE_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	inst_idx	Instance index	
	char_code	RBLE_HGHD_KB_REPORT_CODE	Boot Keyboard Input Report characteristic
		RBLE_HGHD_MO_REPORT_CODE	Boot Mouse Input Report characteristic
	cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification/indication of the characteristic value.
		RBLE_PRF_START_NTF	Start notification of the characteristic value.
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.10 RBLE_HGP_BHost_Set_Report

RBLE_STATUS RBLE_HGP_BHost_Set_Report(uint16_t conhdl, uint8_t inst_idx, RBLE_HGP_REPORT_DESC *report)				
This function sets the Report value. The specifiable Report values are the Boot Keyboard Input Report value, Boot Keyboard Output value, and Boot Mouse Input Report value. Select the Report value by using a combination of device_type and Report_type. The result is reported by using the characteristic write request response event RBLE_HGP_EVENT_BHOST_WRITE_CHAR_RESPONSE. (Note) 1. The application is responsible to prepare the memory block pointed by report and keep it until the report value has been sent. 2. The upper limit of value_size is 8 bytes.				
Parameters:				
	conhdl	Connection handle		
	inst_idx	Instance index		
	**report*	*device_type*	RBLE_HGHD_BOOT_KEYBOARD	Sets the Boot Keyboard Input Report value or Boot Keyboard Output Report value.
			RBLE_HGHD_BOOT_MOUSE	Sets the Boot Mouse Input Report value.
		report_type	RBLE_HGHD_INPUT_REPORT	Specifies that the report is Input Report Type.
			RBLE_HGHD_OUTPUT_REPORT	Specifies that the report is Output Report Type.
		value[RBLE_HIDS_REPORT_MAX]	Report value	

		value_size	Report size
Return:			
	RBLE_OK		Success
	RBLE_STATUS_ERROR		Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.11 RBLE_HGP_BHost_Write_Protocol_Mode

RBLE_STATUS RBLE_HGP_BHost_Write_Protocol_Mode(uint16_t conhdl, uint8_t inst_idx, uint8_t protocol_mode_val)		
This function sends the Protocol Mode. This function is used by the Boot Host to notify the HID Device that the sender is the Boot Host.		
Parameters:		
conhdl	Connection handle	
inst_idx	Instance index	
protocol_mode_val	RBLE_HGHD_PROTOCOL_MODE_BOOT	Boot Protocol mode
Return:		
RBLE_OK	Success	
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.12 RBLE_HGP_BHost_Data_Output

RBLE_STATUS RBLE_HGP_BHost_Data_Output(uint16_t conhdl, uint8_t inst_idx, RBLE_HGP_REPORT_DESC *report)			
This function sends the Boot Keyboard Output Report value. (Note) 1. The application is responsible to prepare the memory block pointed by report and keep it until the report value has been sent. 2. The upper limit of value_size is 8 bytes.			
Parameters:			
conhdl	Connection handle		
inst_idx	Instance index		
**report*	*device_type*	RBLE_HGHD_BOOT_KEYBOARD	Sends Boot Keyboard Output Report value.
	report_type	RBLE_HGHD_OUTPUT_REPORT	Specifies that the report is Output Report Type.
	value[RBLE_HIDS_REPORT_MAX]	Report value	
	value_size	Report size	
Return:			
RBLE_OK		Success	
RBLE_STATUS_ERROR		Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.13 RBLE_HGP_RHost_Enable

RBLE_STATUS RBLE_HGP_RHost_Enable (uint16_t conhdl, uint8_t con_type, uint8_t hids_inst_num, uint8_t bas_inst_num, RBLE_HIDS_CONTENT *hids, RBLE_DIS11_CONTENT *dis, RBLE_BAS_CONTENT *bas, RBLE_HGRH_EVENT_HANDLER call_back);

This function enables the HOGP Report Host role and starts access to the service exposed by the HID Device. The result is reported together with the service handle information by using the Report Host enable completion event RBLE_HGP_EVENT_RHOST_ENABLE_COMP.
When starting access to the service exposed by an HID Device for the first time, configure the connection (by using RBLE_PRF_CON_DISCOVERY) to allow the BLE software to start discovering the service for the HID Device. Save the service handle information that was reported by the Report Host enable completion event.
If the service handle information about the discovered service is saved and is used when a known HID Device is connected normally (by using RBLE_PRF_CON_NORMAL) for a second or subsequent time, detecting the service is skipped, which enables a high-speed connection.
While the HOGP Report Host role is enabled, the service exposed by only one HID Device is accessible. To connect to more than one HID Devices at the same time and access the services exposed by each HID Device, repeat enable (by using RBLE_HGP_RHost_Enable) and disable (by using RBLE_HGP_RHost_Disable) of the HOGP Report Host role in order to switch access to them. At that time, perform normal connection by using the connection handle (which was obtained when connecting to each HID Device) and the handle information (which was saved when starting access to the service for the first time) as parameters.
(Note)
The application is responsible to prepare the memory blocks pointed by hids and bas and keep them until RBLE_HGP_RHost_Disable has been completed.

Parameters:

conhdl	Connection handle	
con_type	RBLE_PRF_CON_DISCOVERY	Configuration connection performed when connecting for the first time
	RBLE_PRF_CON_NORMAL	Normal connection performed when connecting for the second and subsequent times
hids_inst_num	Number of HID service instances	
bas_inst_num	Number of Battery service instances	
*hids	*shdl*	HID service start handle
	ehdl	HID service end handle
	protocol_md_char_hdl	Protocol Mode characteristic handle
	protocol_md_val_hdl	Protocol Mode characteristic value handle
	protocol_md_prop	Protocol Mode property
	report_input_char_hdl	Report (Input) characteristic handle
	report_input_val_hdl	Report (Input) characteristic value handle
	report_input_cfg_hdl	Report (Input) characteristic configuration descriptor handle
	input_rep_ref_hdl	Report (Input) characteristic reference descriptor handle
	report_input_prop	Report (Input) property
	report_output_char_hdl	Report (Output) characteristic handle
	report_output_val_hdl	Report (Output) characteristic value handle
	output_rep_ref_hdl	Report (Output) characteristic reference descriptor handle
	report_output_prop	Report (Output) property
	report_feature_char_hdl	Report (Feature) characteristic handle
	report_feature_val_hdl	Report (Feature) characteristic value handle

		feature_rep_ref_hdl	Report (Feature) characteristic reference descriptor handle
		report_feature_prop	Report (Feature) property
		report_map_char_hdl	Report Map characteristic handle
		report_map_val_hdl	Report Map characteristic value handle
		external_rep_ref_hdl	Report Map characteristic external reference descriptor handle
		report_map_prop	Report Map property
		bootkb_input_char_hdl	Boot Keyboard Input Report characteristic handle
		bootkb_input_val_hdl	Boot Keyboard Input Report characteristic value handle
		bootkb_input_cfg_hdl	Boot Keyboard Input Report characteristic configuration descriptor handle
		bootkb_input_prop	Boot Keyboard Input Report property
		bootkb_output_char_hdl	Boot Keyboard Output Report characteristic handle
		bootkb_output_val_hdl	Boot Keyboard Output Report characteristic value handle
		bootkb_output_prop	Boot Keyboard Output Report property
		bootmo_input_char_hdl	Boot Mouse Input Report characteristic handle
		bootmo_input_val_hdl	Boot Mouse Input Report characteristic value handle
		bootmo_input_cfg_hdl	Boot Mouse Input Report characteristic configuration descriptor handle
		bootmo_input_prop	Boot Mouse Input Report property
		hid_info_char_hdl	HID Information characteristic handle
		hid_info_val_hdl	HID Information characteristic value handle
		hid_info_prop	HID Information property
		hid_cp_char_hdl	HID Control Point characteristic handle
		hid_cp_val_hdl	HID Control Point characteristic value handle
		hid_cp_prop	HID Control Point property
		include_svc_hdl	Included service handle
		include_svc_uuid	UUID of Included service
		incl_shdl	Included service start handle
		incl_ehdl	Included service end handle
	**dis*	*shdl*	Device Information service start handle
		ehdl	Device Information service end handle
		pnp_id_char_hdl	PnP ID characteristic handle
		pnp_id_val_hdl	PnP ID characteristic value handle
		pnp_id_prop	PnP ID property
	**bas*	*shdl*	Battery service start handle
		ehdl	Battery service end handle
		battery_lvl_char_hdl	Battery Level characteristic handle
		battery_lvl_val_hdl	Battery Level characteristic value handle
		battery_lvl_cfg_hdl	Battery Level characteristic configuration descriptor handle

		battery_lvl_rep_ref_hdl	Battery Level characteristic reference descriptor handle
		battery_lvl_prop	Battery Level property
	call_back	Specify the callback function that reports the Report Host role event.	
Return:			
	RBLE_OK	Success	
	RBLE_ERR	Error occurred in Report Host role enable processing	
	RBLE_PARAM_ERR	Invalid parameter	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.14 RBLE_HGP_RHost_Disable

RBLE_STATUS RBLE_HGP_RHost_Disable (uint16_t conhdl)		
This function disables the HOGP Report Host role and terminates access to the service exposed by the HID Device. The result is reported by using the Report Host role disable completion event RBLE_HGP_EVENT_RHOST_DISABLE_COMP.		
Parameters:		
	conhdl	Connection handle
Return:		
	RBLE_OK	Success
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.15 RBLE_HGP_RHost_Read_Char

RBLE_STATUS RBLE_HGP_RHost_Read_Char(uint16_t conhdl, uint8_t inst_idx, uint8_t char_code)		
This function reads the characteristic value of the HID service and the Device Information service. The result is reported by using the characteristic read request response event RBLE_HGP_EVENT_RHOST_READ_CHAR_RESPONSE.		
Parameters:		
conhdl	Connection handle	
inst_idx	Instance index	
char_code	RBLE_HGRH_RD_HIDS_PM	Protocol Mode
	RBLE_HGRH_RD_HIDS_RI	Report (Input) value
	RBLE_HGRH_RD_HIDS_RI_CFG	Report (Input) value notification setting
	RBLE_HGRH_RD_HIDS_RI_REF	Report (Input) reference descriptor
	RBLE_HGRH_RD_HIDS_RO	Report (Output) value
	RBLE_HGRH_RD_HIDS_RO_REF	Report (Output) reference descriptor
	RBLE_HGRH_RD_HIDS_RF	Report (Feature) value
	RBLE_HGRH_RD_HIDS_RF_REF	Report (Feature) reference descriptor
	RBLE_HGRH_RD_HIDS_RM	Report Map
	RBLE_HGRH_RD_HIDS_ER_REF	Report Map external reference descriptor
	RBLE_HGRH_RD_HIDS_HI	HID information
	RBLE_HGRH_RD_DIS_PNPID	PnP ID of HID Device
	RBLE_HGRH_RD_BAS_BL	Battery Level

RBLE_STATUS RBLE_HGP_RHost_Read_Char(uint16_t conhdl, uint8_t inst_idx, uint8_t char_code)			
		RBLE_HGRH_RD_BAS_BL_CFG	Battery Level notification setting
		RBLE_HGRH_RD_BAS_BL_REF	Battery Level reference descriptor
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.16 RBLE_HGP_RHost_Read_By_UUID_Char

RBLE_STATUS RBLE_HGP_RHost_Read_By_UUID_Char (uint16_t conhdl, uint8_t inst_idx, uint8_t char_code)			
This function reads the characteristic value of the Device Information service and the Battery service specified by the UUID characteristic. The result is reported by using the characteristic read request response event RBLE_HGP_EVENT_RHOST_READ_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	inst_idx	Instance index	
	char_code	RBLE_HGRH_RD_DIS_PNPID	PnP ID of HID Device
		RBLE_HGRH_RD_BAS_BL	Battery Level
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.17 RBLE_HGP_RHost_Read_Long_Char

RBLE_STATUS RBLE_HGP_RHost_Read_Long_Char (uint16_t conhdl, uint8_t inst_idx, uint8_t char_code)			
This function reads the characteristic values of the HID service. Use this function to acquire a characteristic value of more than 24 octets. The result is reported by using the long characteristic read request response event RBLE_HGP_EVENT_RHOST_READ_LONG_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	inst_idx	Instance index	
	char_code	RBLE_HGRH_RD_HIDS_RI	Report (Input) value
		RBLE_HGRH_RD_HIDS_RO	Report (Output) value
		RBLE_HGRH_RD_HIDS_RF	Report (Feature) value
		RBLE_HGRH_RD_HIDS_RM	Report map
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.18 RBLE_HGP_RHost_Write_Char

RBLE_STATUS RBLE_HGP_RHost_Write_Char(uint16_t conhdl, uint8_t inst_idx, uint8_t char_code, uint16_t cfg_val)			
This function writes each characteristic configuration descriptor of the HID service and the Battery service. The result is reported by using the characteristic write request response event RBLE_HGP_EVENT_RHOST_WRITE_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	inst_idx	Instance index	
	char_code	RBLE_HGHD_REPORT_INPUT_CODE	Report (Input) characteristic
		RBLE_HGHD_BATTERY_LEVEL_CODE	Battery Level characteristic
	cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification/indication of the characteristic value.
		RBLE_PRF_START_NTF	Start notification of the characteristic value.
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.19 RBLE_HGP_RHost_Set_Report

RBLE_STATUS RBLE_HGP_RHost_Set_Report(uint16_t conhdl, uint8_t inst_idx, RBLE_HGP_REPORT_DESC *report)				
This function sets the Report value. The specifiable Report values are the Report (Input) value, the Report (Output) value, and the Report (Feature) value. Select the Report value by using a combination of device_type and Report_type. The result is reported by using the characteristic write request response event RBLE_HGP_EVENT_RHOST_WRITE_CHAR_RESPONSE. (Note) 1. The application is responsible to prepare the memory block pointed by report and keep it until the report value has been sent. 2. The upper limit of value_size is the MTU size.				
Parameters:				
	conhdl	Connection handle		
	inst_idx	Instance index		
	**report*	*device_type*	RBLE_HGHD_HID_DEVICE	Sets the Report value.
		report_type	RBLE_HGHD_INPUT_REPORT	Specifies that the report is Input Report Type.
			RBLE_HGHD_OUTPUT_REPORT	Specifies that the report is Output Report Type.
			RBLE_HGHD_FEATURE_REPORT	Specifies that the report is Feature Report Type.
		value[RBLE_HIDS_REPORT_MAX]	Report value	
		value_size	Report size	
Return:				

RBLE_OK	Success
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.20 RBLE_HGP_RHost_Write_Protocol_Mode

RBLE_STATUS RBLE_HGP_RHost_Write_Protocol_Mode(uint16_t conhdl, uint8_t inst_idx, uint8_t protocol_mode_val)

This function sends the Protocol Mode.
This function is used by the Report Host to notify the HID Device that the sender is the Report Host.

Parameters:

conhdl	Connection handle	
inst_idx	Instance index	
protocol_mode_val	RBLE_HGHD_PROTOCOL_MODE_REPORT	Report protocol mode

Return:

RBLE_OK	Success
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.21 RBLE_HGP_RHost_Data_Output

RBLE_STATUS RBLE_HGP_RHost_Data_Output(uint16_t conhdl, uint8_t inst_idx, RBLE_HGP_REPORT_DESC *report)

This function sends the Report (Output) value.
(Note)
1. The application is responsible to prepare the memory block pointed by report and keep it until the report value has been sent.
2. The upper limit of value_size is the MTU size.

Parameters:

conhdl	Connection handle		
inst_idx	Instance index		
**report*	*device_type*	RBLE_HGHD_HID_DEVICE	Sends the Report (Output) value.
	report_type	RBLE_HGHD_OUTPUT_REPORT	Specifies that the report is Output Report Type.
	value[RBLE_HIDS_REPORT_MAX]	Report value	
	value_size	Report size	

Return:

RBLE_OK	Success
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.22 RBLE_HGP_RHost_Write_Control_Point

RBLE_STATUS RBLE_HGP_RHost_Write_Control_Point (uint16_t conhdl, uint8_t inst_idx, uint8_t control_point_val)			
This function sends a Suspend status notification to the HID Device by using the HID command defined by the Bluetooth HID Profile.			
Parameters:			
	conhdl	Connection handle	
	inst_idx	Instance index	
	control_point_val	RBLE_HGHD_CTRL_POINT_SUSPEND	The Report Host is in the suspend state.
		RBLE_HGHD_POINT_EXIT_SUSPEND	The Report Host has exited the suspend state.
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.3 Events

Table 3-2 shows the events defined for the HOGP of rBLE and the following sections describe the events in detail. HOGP events are named according to the following rules:

- Events related to the HID Device: RBLE_HGP_EVENT_HDEVICE_*XXXXX*
- Events related to the Boot Host: RBLE_HGP_EVENT_ BHOST_*XXXX*X
- Events related to the Report Host: RBLE_HGP_EVENT_ RHOST_*XXXXX*

Table 3-2 Events Defined for the HOGP

RBLE_HGP_EVENT_HDEVICE_ENABLE_COMP	HID Device enable completion event
RBLE_HGP_EVENT_HDEVICE_DISABLE_COMP	HID Device disable completion event
RBLE_HGP_EVENT_HDEVICE_ERROR_IND	HID Device error indication event
RBLE_HGP_EVENT_HDEVICE_CFG_INDNTF_IND	Configured value change indication event
RBLE_HGP_EVENT_HDEVICE_REPORT_IND	Report value change indication event
RBLE_HGP_EVENT_HDEVICE_PROTOCOL_MODE_CHG_EVT	Protocol Mode change notification event
RBLE_HGP_EVENT_HDEVICE_REPORT_EVT	Report value notification event
RBLE_HGP_EVENT_HDEVICE_HID_CP_CHG_EVT	Control Point change notification event
RBLE_HGP_EVENT_HDEVICE_REPORT_COMP	Report value send completion event
RBLE_HGP_EVENT_HDEVICE_SEND_BATTERY_LEVEL_COMP	Battery Level send completion event
RBLE_HGP_EVENT_HDEVICE_COMMAND_DISALLOWED_IND	Command disallowed indication event
RBLE_HGP_EVENT_BHOST_ENABLE_COMP	Boot Host enable completion event
RBLE_HGP_EVENT_BHOST_DISABLE_COMP	Boot Host disable completion event
RBLE_HGP_EVENT_BHOST_ERROR_IND	Boot Host error indication event
RBLE_HGP_EVENT_BHOST_READ_CHAR_RESPONSE	Characteristic read request response event
RBLE_HGP_EVENT_BHOST_WRITE_CHAR_RESPONSE	Characteristic write request response event
RBLE_HGP_EVENT_BHOST_REPORT_NTF	Report value notification event
RBLE_HGP_EVENT_BHOST_COMMAND_DISALLOWED_IND	Command disallowed indication event
RBLE_HGP_EVENT_RHOST_ENABLE_COMP	Report Host enable completion event
RBLE_HGP_EVENT_RHOST_DISABLE_COMP	Report Host disable completion event
RBLE_HGP_EVENT_RHOST_ERROR_IND	Report Host error indication event
RBLE_HGP_EVENT_RHOST_READ_CHAR_RESPONSE	Characteristic read request response event
RBLE_HGP_EVENT_RHOST_READ_LONG_CHAR_RESPONSE	Long characteristic read request response event
RBLE_HGP_EVENT_RHOST_WRITE_CHAR_RESPONSE	Characteristic write request response event
RBLE_HGP_EVENT_RHOST_REPORT_NTF	Report value notification event
RBLE_HGP_EVENT_RHOST_BATTERY_LEVEL_NTF	Battery Level notification event
RBLE_HGP_EVENT_RHOST_COMMAND_DISALLOWED_IND	Command disallowed indication event

3.3.1 RBLE_HGP_EVENT_HDEVICE_ENABLE_COMP

RBLE_HGP_EVENT_HDEVICE_ENABLE_COMP		
This event reports the result of enabling the HOGP HID Device role (RBLE_HGP_HDevice_Enable).		
Parameters:		
	status	Result of enabling the HID Device (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)
	conhdl	Connection handle

3.3.2 RBLE_HGP_EVENT_HDEVICE_DISABLE_COMP

RBLE_HGP_EVENT_HDEVICE_DISABLE_COMP			
This event reports the result of disabling the HOGP HID Device role (RBLE_HGP_HDevice_Disable).			
Parameters:			
conhdl	Connection handle		
status	Result of enabling the HID Device (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)		
**device_info*	*hids_inst_num*	Number of HID service instances	
	bas_inst_num	Number of Battery service instances	
	report_input_ntf_en [RBLE_HIDS_INST_MAX]	RBLE_PRF_STOP_NTFIND	Stop notification/indication of Report (Input).
		RBLE_PRF_START_NTF	Start notification of Report (Input).
	kb_report_ntf_en [RBLE_HIDS_INST_MAX]	RBLE_PRF_STOP_NTFIND	Stop notification/indication of Boot Keyboard Input Report.
		RBLE_PRF_START_NTF	Start notification of Boot Keyboard Input Report.
	mo_report_ntf_en [RBLE_HIDS_INST_MAX]	RBLE_PRF_STOP_NTFIND	Stop notification/indication of Boot Mouse Input Report.
		RBLE_PRF_START_NTF	Start notification of Boot Mouse Input Report.
	protocol_mode_val [RBLE_HIDS_INST_MAX]	Protocol Mode set value	
	battery_level_ntf_en [RBLE_BAS_INST_MAX]	RBLE_PRF_STOP_NTFIND	Stop notification/indication of the Battery Level.
		RBLE_PRF_START_NTF	Start notification of the Battery Level.

3.3.3 RBLE_HGP_EVENT_HDEVICE_ERROR_IND

RBLE_HGP_EVENT_HDEVICE_ERROR_IND		
This event indicates an error code unique to the HOGP HID Device role.		
Parameters:		
	conhdl	Connection handle
	status	Error code (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)

3.3.4 RBLE_HGP_EVENT_HDEVICE_CFG_INDNTF_IND

RBLE_HGP_EVENT_HDEVICE_CFG_INDNTF_IND		
This event indicates that the value of the client characteristic configuration descriptor of the HID Device service has been set.		
Parameters:		
conhdl	Connection handle	
inst_idx	Instance index	
char_code	RBLE_HGHD_REPORT_INPUT_CODE	Report (Input) characteristic
	RBLE_HGHD_KB_REPORT_CODE	Boot Keyboard Input Report characteristic
	RBLE_HGHD_MO_REPORT_CODE	Boot Mouse Input Report characteristic
	RBLE_HGHD_BATTERY_LEVEL_CODE	Battery Level characteristic
cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification/indication.
	RBLE_PRF_START_NTF	Start notification.

3.3.5 RBLE_HGP_EVENT_HDEVICE_REPORT_IND

RBLE_HGP_EVENT_HDEVICE_REPORT_IND				
This event indicates that the Report value has been set from the Boot Host or Report Host. The specifiable Report values are the Report (Input) value, Report (Output) value, Report (Feature) value, Boot Keyboard Input Report value, Boot Keyboard Output value, and Boot Mouse Input Report value. Determine which value has been set by checking the combination of device_type and Report_type.				
Parameters:				
conhdl	Connection handle			
inst_idx	Instance index			
report	*device_type*	RBLE_HGHD_HID_DEVICE	Indicates that the value is the Report (Input) value, Report (Output) value, or Report (Feature) value.	
		RBLE_HGHD_BOOT_KEYBOARD	Indicates that the value is the Boot Keyboard Input Report value or the Boot Keyboard Output Report value.	
		RBLE_HGHD_BOOT_MOUSE	Indicates that the value is the Boot Mouse Input Report value.	
	report_type	RBLE_HGHD_INPUT_REPORT	Indicates that the report is Input Report Type.	
		RBLE_HGHD_OUTPUT_REPORT	Indicates that the report is Output Report Type.	
		RBLE_HGHD_FEATURE_REPORT	Indicates that the report is Feature Report Type.	
	value[RBLE_HIDS_REPORT_MAX]	Report value		
	value_size	Report size		

3.3.6 RBLE_HGP_EVENT_HDEVICE_PROTOCOL_MODE_CHG_EVT

RBLE_HGP_EVENT_HDEVICE_PROTOCOL_MODE_CHG_EVT			
This event reports that the Protocol Mode has been received from the Boot Host or Report Host.			
Parameters:			
conhdl	Connection handle		
inst_idx	Instance index		
protocol_mode_val	RBLE_HGHD_PROTOCOL_MODE_BOOT	Boot Protocol mode	
	RBLE_HGHD_PROTOCOL_MODE_REPORT	Report Protocol mode	

3.3.7 RBLE_HGP_EVENT_HDEVICE_REPORT_EVT

RBLE_HGP_EVENT_HDEVICE_REPORT_EVT				
This event reports that the Report value has been received from the Boot Host or Report Host. The received Report value is either the Report (Output) value or the Boot Keyboard Output Report value. Determine which value has been received by checking the combination of device_type and Report_type.				
Parameters:				
conhdl	Connection handle			
inst_idx	Instance index			
report	*device_type*	RBLE_HGHD_HID_DEVICE	Indicates that the value is the Report (Output) value.	
		RBLE_HGHD_BOOT_KEYBOARD	Indicates that the value is the Boot Keyboard Output Report value.	
	report_type	RBLE_HGHD_OUTPUT_REPORT	Indicates that the report is Output Report Type.	
	value[RBLE_HIDS_REPORT_MAX]		Report value	
	value_size		Report size	

3.3.8 RBLE_HGP_EVENT_HDEVICE_HID_CP_CHG_EVT

RBLE_HGP_EVENT_HDEVICE_HID_CP_CHG_EVT			
This event reports that a Suspend notification has been received from the Report Host by using the HID command defined by the Bluetooth HID Profile.			
Parameters:			
conhdl	Connection handle		
inst_idx	Instance index		
control_point_val	RBLE_HGHD_CTRL_POINT_SUSPEND	The Report Host is in the suspend state.	
	RBLE_HGHD_POINT_EXIT_SUSPEND	The Report Host has exited the suspend state.	

3.3.9 RBLE_HGP_EVENT_HDEVICE_REPORT_COMP

RBLE_HGP_EVENT_HDEVICE_REPORT_COMP	
This event reports completion of Report sending.	
Parameters:	
conhdl	Connection handle
status	Report send completion result (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)

3.3.10 RBLE_HGP_EVENT_HDEVICE_SEND_BATTERY_LEVEL_COMP

RBLE_HGP_EVENT_HDEVICE_SEND_BATTERY_LEVEL_COMP		
This event reports completion of sending the Battery Level to the Report Host.		
Parameters:		
conhdl	Connection handle	
status	Battery Level send completion result (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)	

3.3.11 RBLE_HGP_EVENT_HDEVICE_COMMAND_DISALLOWED_IND

RBLE_HGP_EVENT_HDEVICE_COMMAND_DISALLOWED_IND		
This event indicates the error that occurs when a command executed by the BLE software is in a state in which the HID Device role cannot execute a command, and so that command cannot be accepted.		
Parameters:		
status	Result of command execution (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)	
opcode	RBLE_CMD_HGP_HDEVICE_ENABLE	HID Device enable event
	RBLE_CMD_HGP_HDEVICE_DISABLE	HID Device disable event
	RBLE_CMD_HGP_HDEVICE_SEND_REPORT	Report send command
	RBLE_CMD_HGP_HDEVICE_SEND_BATTERY_LEVEL	Battery Level send command

3.3.12 RBLE_HGP_EVENT_BHOST_ENABLE_COMP

RBLE_HGP_EVENT_BHOST_ENABLE_COMP		
This event reports the result of enabling the HOGP Boot Host (RBLE_HGP_BHost_Enable). Save the obtained handle information about the discovered service, to enable a high-speed access to the service without service detection when restarting access to the service exposed by the HID Device.		
Parameters:		
conhdl	Connection handle	
status	Result of enabling the Boot Host (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)	
hids_inst_num	Number of HID service instances	
bas_inst_num	Number of Battery service instances	
hids[RBLE_HIDS_INST_MAX]	*shdl*	HID service start handle
	ehdl	HID service end handle
	protocol_md_char_hdl	Protocol Mode characteristic handle
	protocol_md_val_hdl	Protocol Mode characteristic value handle
	protocol_md_prop	Protocol Mode property
	report_input_char_hdl	Report (Input) characteristic handle
	report_input_val_hdl	Report (Input) characteristic value handle
	report_input_cfg_hdl	Report (Input) characteristic configuration descriptor handle
	input_rep_ref_hdl	Report (Input) characteristic reference descriptor handle
	report_input_prop	Report (Input) property
	report_output_char_hdl	Report (Output) characteristic handle
	report_output_val_hdl	Report (Output) characteristic value handle
	output_rep_ref_hdl	Report (Output) characteristic reference descriptor handle
	report_output_prop	Report (Output) property
	report_feature_char_hdl	Report (Feature) characteristic handle
	report_feature_val_hdl	Report (Feature) characteristic value handle
	feature_rep_ref_hdl	Report (Feature) characteristic reference descriptor handle
	report_feature_prop	Report (Feature) property
	report_map_char_hdl	Report Map characteristic handle
	report_map_val_hdl	Report Map characteristic value handle
	external_rep_ref_hdl	Report Map characteristic external reference descriptor handle
	report_map_prop	Report Map property
	bootkb_input_char_hdl	Boot Keyboard Input Report characteristic handle
	bootkb_input_val_hdl	Boot Keyboard Input Report characteristic value handle
	bootkb_input_cfg_hdl	Boot Keyboard Input Report characteristic configuration descriptor handle
	bootkb_input_prop	Boot Keyboard Input Report property

		bootkb_output_char_hdl	Boot Keyboard Output Report characteristic handle
		bootkb_output_val_hdl	Boot Keyboard Output Report characteristic value handle
		bootkb_output_prop	Boot Keyboard Output Report property
		bootmo_input_char_hdl	Boot Mouse Input Report characteristic handle
		bootmo_input_val_hdl	Boot Mouse Input Report characteristic value handle
		bootmo_input_cfg_hdl	Boot Mouse Input Report characteristic configuration descriptor handle
		bootmo_input_prop	Boot Mouse Input Report property
		hid_info_char_hdl	HID Information characteristic handle
		hid_info_val_hdl	HID Information characteristic value handle
		hid_info_prop	HID Information property
		hid_cp_char_hdl	HID Control Point characteristic handle
		hid_cp_val_hdl	HID Control Point characteristic value handle
		hid_cp_prop	HID Control Point property
		include_svc_hdl	Included service handle
		include_svc_uuid	UUID of Included service
		incl_shdl	Included service start handle
		incl_ehdl	Included service end handle
	dis	*shdl*	Device Information service start handle
		ehdl	Device Information service end handle
		pnp_id_char_hdl	PnP ID characteristic handle
		pnp_id_val_hdl	PnP ID characteristic value handle
		pnp_id_prop	PnP ID property
	bas[RBLE_BAS_INST_MAX]	*shdl*	Battery service start handle
		ehdl	Battery service end handle
		battery_lvl_char_hdl	Battery Level characteristic handle
		battery_lvl_val_hdl	Battery Level characteristic value handle
		battery_lvl_cfg_hdl	Battery Level characteristic configuration descriptor handle
		battery_lvl_rep_ref_hdl	Battery Level characteristic reference descriptor handle
		battery_lvl_prop	Battery Level property

3.3.13 RBLE_HGP_EVENT_BHOST_DISABLE_COMP

RBLE_HGP_EVENT_BHOST_DISABLE_COMP		
This event reports the result of disabling the HOGP Boot Host (RBLE_HGP_BHost_Disable).		
Parameters:		
	conhdl	Connection handle
	status	Result of disabling the Boot Host (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)

3.3.14 RBLE_HGP_EVENT_BHOST_ERROR_IND

RBLE_HGP_EVENT_BHOST_ERROR_IND		
This event indicates an error code unique to the HOGP Boot Host role. This event is generated when the BLE software cannot continue processing for reasons such as that an invalid parameter has been specified in a request from the application.		
Parameters:		
conhdl	Connection handle	
status	Error code (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)	

3.3.15 RBLE_HGP_EVENT_BHOST_READ_CHAR_RESPONSE

RBLE_HGP_EVENT_BHOST_READ_CHAR_RESPONSE		
This event reports the response to the characteristic value read request (RBLE_HGP_BHost_Read_Char or RBLE_HGP_BHost_Read_By_UUID_Char). Read out the acquired data in accordance with the contents of the request.		
Parameters:		
conhdl	Connection handle	
att_code	0x00	Characteristic value successfully acquired
	Other than 0x00	Error occurred when acquiring characteristic value See *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for ATT error code*.)
data	*each_len*	Length of each result
	len	Data length
	data[RBLE_ATTM_MAX_VALUE]	Read characteristic data

3.3.16 RBLE_HGP_EVENT_BHOST_WRITE_CHAR_RESPONSE

RBLE_HGP_EVENT_BHOST_WRITE_CHAR_RESPONSE		
This event reports the response to the characteristic value write request (RBLE_HGP_BHost_Write_Char).		
Parameters:		
conhdl	Connection handle	
att_code	0x00	Characteristic value successfully written
	Other than 0x00	Error occurred when writing characteristic value See *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for ATT error code*.)

3.3.17 RBLE_HGP_EVENT_BHOST_REPORT_NTF

RBLE_HGP_EVENT_BHOST_REPORT_NTF				
This event reports the Report value received from the HID Device. The received Report value is either the Boot Keyboard Input Report value or the Boot Mouse Input Report value. Determine which value has been received by checking the combination of device_type and Report_type.				
Parameters:				
	conhdl	Connection handle		
	inst_idx	Instance index		
	report	*device_type*	RBLE_HGHD_BOOT_KEYBOARD	Indicates that the value is the Boot Keyboard Input Report value.
			RBLE_HGHD_BOOT_MOUSE	Indicates that the value is the Boot Mouse Input Report value.
		report_type	RBLE_HGHD_INPUT_REPORT	Indicates that the report is Input Report Type.
		value[RBLE_HIDS_REPORT_MAX]	Report value	
		value_size	Report size	

3.3.18 RBLE_HGP_EVENT_BHOST_COMMAND_DISALLOWED_IND

RBLE_HGP_EVENT_BHOST_COMMAND_DISALLOWED_IND			
This event indicates the error that occurs when a command executed by the BLE software is in a state in which the Boot Host role cannot execute a command, and so that command cannot be accepted.			
Parameters:			
	status	Result of command execution (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)	
	opcode	RBLE_CMD_HGP_BHOST_ENABLE	Boot Host enable command
		RBLE_CMD_HGP_BHOST_DISABLE	Boot Host disable command
		RBLE_CMD_HGP_BHOST_READ_CHAR	Characteristic read command
		RBLE_CMD_HGP_BHOST_READ_CHAR_BY_UUID	UUID-specified characteristic read command
		RBLE_CMD_HGP_BHOST_WRITE_CHAR	Characteristic write command
		RBLE_CMD_HGP_BHOST_SET_REPORT	Report value setting command
		RBLE_CMD_HGP_BHOST_SET_PROTOCOL_MODE	Protocol Mode send command
		RBLE_CMD_HGP_BHOST_DATA_OUTPUT	Report value send command

3.3.19 RBLE_HGP_EVENT_RHOST_ENABLE_COMP

RBLE_HGP_EVENT_RHOST_ENABLE_COMP		
This event reports the result of enabling the HOGP Report Host (RBLE_HGP_RHost_Enable). Save the obtained handle information about the discovered service, to enable a high-speed access to the service without service detection when restarting access to the service exposed by the HID Device.		
Parameters:		
conhdl	Connection handle	
status	Result of enabling the Report Host (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)	
hids_inst_num	Number of HID service instances	
bas_inst_num	Number of Battery service instances	
hids[RBLE_HIDS_INST_MAX]	*shdl*	HID service start handle
	ehdl	HID service end handle
	protocol_md_char_hdl	Protocol Mode characteristic handle
	protocol_md_val_hdl	Protocol Mode characteristic value handle
	protocol_md_prop	Protocol Mode property
	report_input_char_hdl	Report (Input) characteristic handle
	report_input_val_hdl	Report (Input) characteristic value handle
	report_input_cfg_hdl	Report (Input) characteristic configuration descriptor handle
	input_rep_ref_hdl	Report (Input) characteristic reference descriptor handle
	report_input_prop	Report (Input) property
	report_output_char_hdl	Report (Output) characteristic handle
	report_output_val_hdl	Report (Output) characteristic value handle
	output_rep_ref_hdl	Report (Output) characteristic reference descriptor handle
	report_output_prop	Report (Output) property
	report_feature_char_hdl	Report (Feature) characteristic handle
	report_feature_val_hdl	Report (Feature) characteristic value handle
	feature_rep_ref_hdl	Report (Feature) characteristic reference descriptor handle
	report_feature_prop	Report (Feature) property
	report_map_char_hdl	Report Map characteristic handle
	report_map_val_hdl	Report Map characteristic value handle
	external_rep_ref_hdl	Report Map characteristic external reference descriptor handle
	report_map_prop	Report Map property
	bootkb_input_char_hdl	Boot Keyboard Input Report characteristic handle
	bootkb_input_val_hdl	Boot Keyboard Input Report characteristic value handle
	bootkb_input_cfg_hdl	Boot Keyboard Input Report characteristic configuration descriptor handle
	bootkb_input_prop	Boot Keyboard Input Report property

		bootkb_output_char_hdl	Boot Keyboard Output Report characteristic handle
		bootkb_output_val_hdl	Boot Keyboard Output Report characteristic value handle
		bootkb_output_prop	Boot Keyboard Output Report property
		bootmo_input_char_hdl	Boot Mouse Input Report characteristic handle
		bootmo_input_val_hdl	Boot Mouse Input Report characteristic value handle
		bootmo_input_cfg_hdl	Boot Mouse Input Report characteristic configuration descriptor handle
		bootmo_input_prop	Boot Mouse Input Report property
		hid_info_char_hdl	HID Information characteristic handle
		hid_info_val_hdl	HID Information characteristic value handle
		hid_info_prop	HID Information property
		hid_cp_char_hdl	HID Control Point characteristic handle
		hid_cp_val_hdl	HID Control Point characteristic value handle
		hid_cp_prop	HID Control Point property
		include_svc_hdl	Included service handle
		include_svc_uuid	UUID of Included service
		incl_shdl	Included service start handle
		incl_ehdl	Included service end handle
	dis	*shdl*	Device Information service start handle
		ehdl	Device Information service end handle
		pnp_id_char_hdl	PnP ID characteristic handle
		pnp_id_val_hdl	PnP ID characteristic value handle
		pnp_id_prop	PnP ID property
	bas[RBLE_BAS_INST_MAX]	*shdl*	Battery service start handle
		ehdl	Battery service end handle
		battery_lvl_char_hdl	Battery Level characteristic handle
		battery_lvl_val_hdl	Battery Level characteristic value handle
		battery_lvl_cfg_hdl	Battery Level characteristic configuration descriptor handle
		battery_lvl_rep_ref_hdl	Battery Level characteristic reference descriptor handle
		battery_lvl_prop	Battery Level property

3.3.20 RBLE_HGP_EVENT_RHOST_DISABLE_COMP

RBLE_HGP_EVENT_RHOST_DISABLE_COMP		
This event reports the result of disabling the HOGP Report Host (RBLE_HGP_RHost_Disable).		
Parameters:		
	conhdl	Connection handle
	status	Result of disabling the Report Host (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)

3.3.21 RBLE_HGP_EVENT_RHOST_ERROR_IND

RBLE_HGP_EVENT_RHOST_ERROR_IND		
This event indicates an error code unique to the HOGP Report Host role. This event is generated when the BLE software cannot continue processing for reasons such as that an invalid parameter has been specified in a request from the application.		
Parameters:		
conhdl	Connection handle	
status	Error code (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)	

3.3.22 RBLE_HGP_EVENT_RHOST_READ_CHAR_RESPONSE

RBLE_HGP_EVENT_RHOST_READ_CHAR_RESPONSE			
This event reports the response to the characteristic value read request (RBLE_HGP_RHost_Read_Char or RBLE_HGP_RHost_Read_By_UUID_Char). Read out the acquired data in accordance with the contents of the request.			
Parameters:			
conhdl	Connection handle		
att_code	0x00	Characteristic value successfully acquired	
	Other than 0x00	Error occurred when acquiring characteristic value See *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for ATT error code*.)	
data	*each_len*		Length of each result
	len		Data length
	data[RBLE_ATTM_MAX_VALUE]		Read characteristic data

3.3.23 RBLE_HGP_EVENT_RHOST_READ_LONG_CHAR_RESPONSE

RBLE_HGP_EVENT_RHOST_READ_LONG_CHAR_RESPONSE			
This event reports the response to the long characteristic value read request (RBLE_HGP_RHost_Read_Long_Char). Read out the acquired data in accordance with the contents of the request.			
Parameters:			
conhdl	Connection handle		
att_code	0x00	Characteristic value successfully acquired	
	Other than 0x00	Error occurred when acquiring characteristic value See *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for ATT error code*.)	
data	*val_len*		Data length
	attr_hdl		characteristic handle
	data[RBLE_ATTM_MAX_VALUE]		Read characteristic data

3.3.24 RBLE_HGP_EVENT_RHOST_WRITE_CHAR_RESPONSE

RBLE_HGP_EVENT_RHOST_WRITE_CHAR_RESPONSE			
This event reports the response to the characteristic value write request (RBLE_HGP_RHost_Write_Char).			
Parameters:			
conhdl	Connection handle		
att_code	0x00	Characteristic value successfully written	
	Other than 0x00	Error occurred when writing characteristic value See *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for ATT error code*.)	

3.3.25 RBLE_HGP_EVENT_RHOST_REPORT_NTF

RBLE_HGP_EVENT_RHOST_REPORT_NTF			
This event reports the Report (Input) value received from the HID Device.			
Parameters:			
conhdl	Connection handle		
inst_idx	Instance index		
report	*device_type*	RBLE_HGHD_HID_DEVICE	Indicates that the value is the Report (Input) value.
	report_type	RBLE_HGHD_INPUT_REPORT	Indicates that the report is Input Report Type.
	value[RBLE_HIDS_REPORT_MAX]	Report value	
	value_size	Report size	

3.3.26 RBLE_HGP_EVENT_RHOST_BATTERY_LEVEL_NTF

RBLE_HGP_EVENT_RHOST_BATTERY_LEVEL_NTF	
This event reports the Battery Level received from the HID Device.	
Parameters:	
conhdl	Connection handle
inst_idx	Instance index
battery_level	Battery Level

3.3.27 RBLE_HGP_EVENT_RHOST_COMMAND_DISALLOWED_IND

RBLE_HGP_EVENT_RHOST_COMMAND_DISALLOWED_IND			
This event indicates the error that occurs when a command executed by the BLE software is in a state in which the Report Host role cannot execute a command, and so that command cannot be accepted.			
Parameters:			
	status	Result of command execution (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)	
	opcode	RBLE_CMD_HGP_RHOST_ENABLE	Report Host enable command
		RBLE_CMD_HGP_RHOST_DISABLE	Report Host disable command
		RBLE_CMD_HGP_RHOST_READ_CHAR	Characteristic read command
		RBLE_CMD_HGP_RHOST_READ_CHAR_BY_UUID	UUID-specified characteristic read command
		RBLE_CMD_HGP_RHOST_READ_LONG_CHAR	Long characteristic read command
		RBLE_CMD_HGP_RHOST_WRITE_CHAR	Characteristic write command
		RBLE_CMD_HGP_RHOST_SET_REPORT	Report value setting command
		RBLE_CMD_HGP_RHOST_SET_PROTOCOL_MODE	Protocol Mode setting command
		RBLE_CMD_HGP_RHOST_DATA_OUTPUT	Report value send command
		RBLE_CMD_HGP_RHOST_SET_CONTROL_POINT	Control Point setting command

3.4 Message Sequence Chart

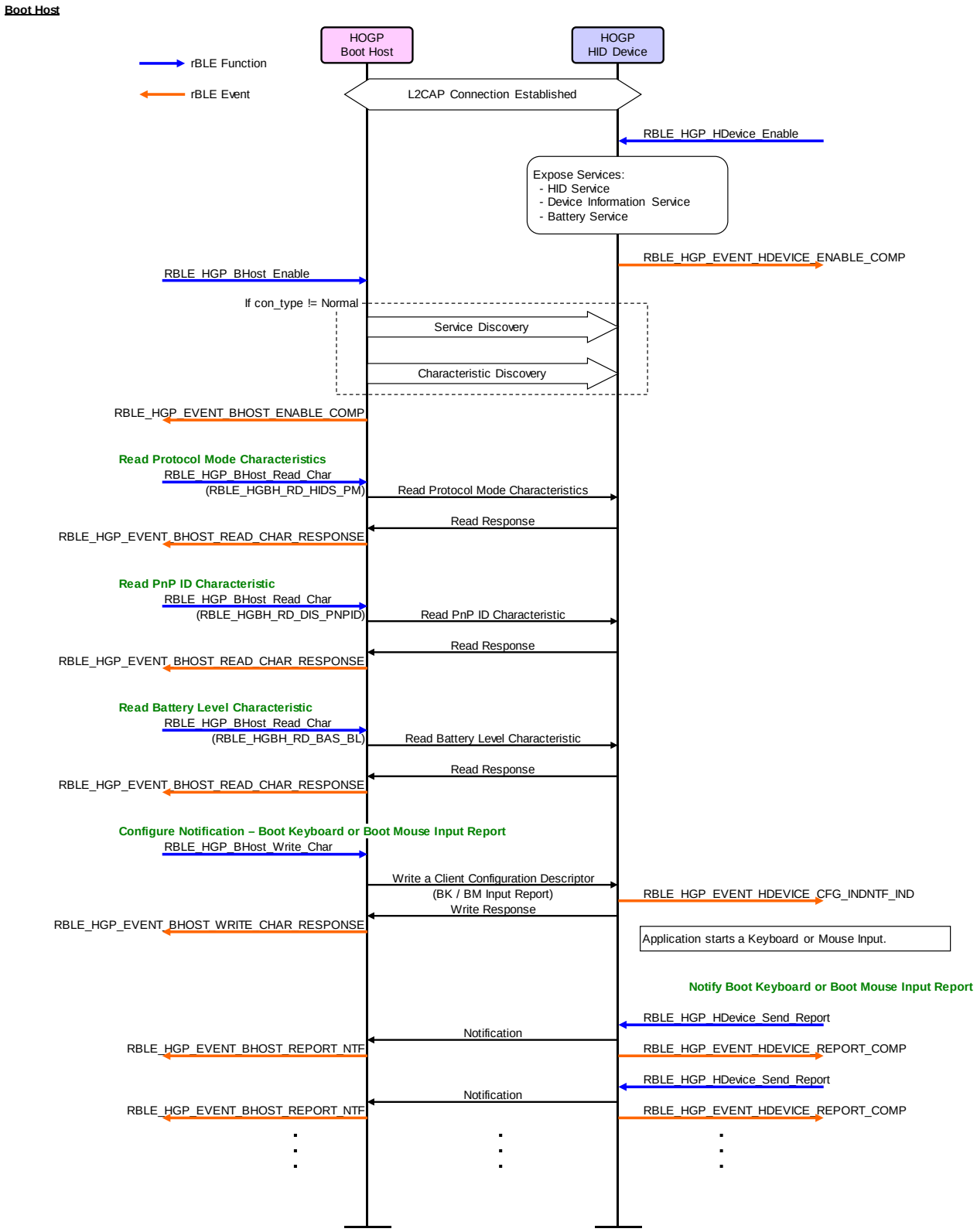

Figure 3-1 Example of Use Case in Which HOGP is Implemented by Using rBLE API (Boot Host and HID Device)

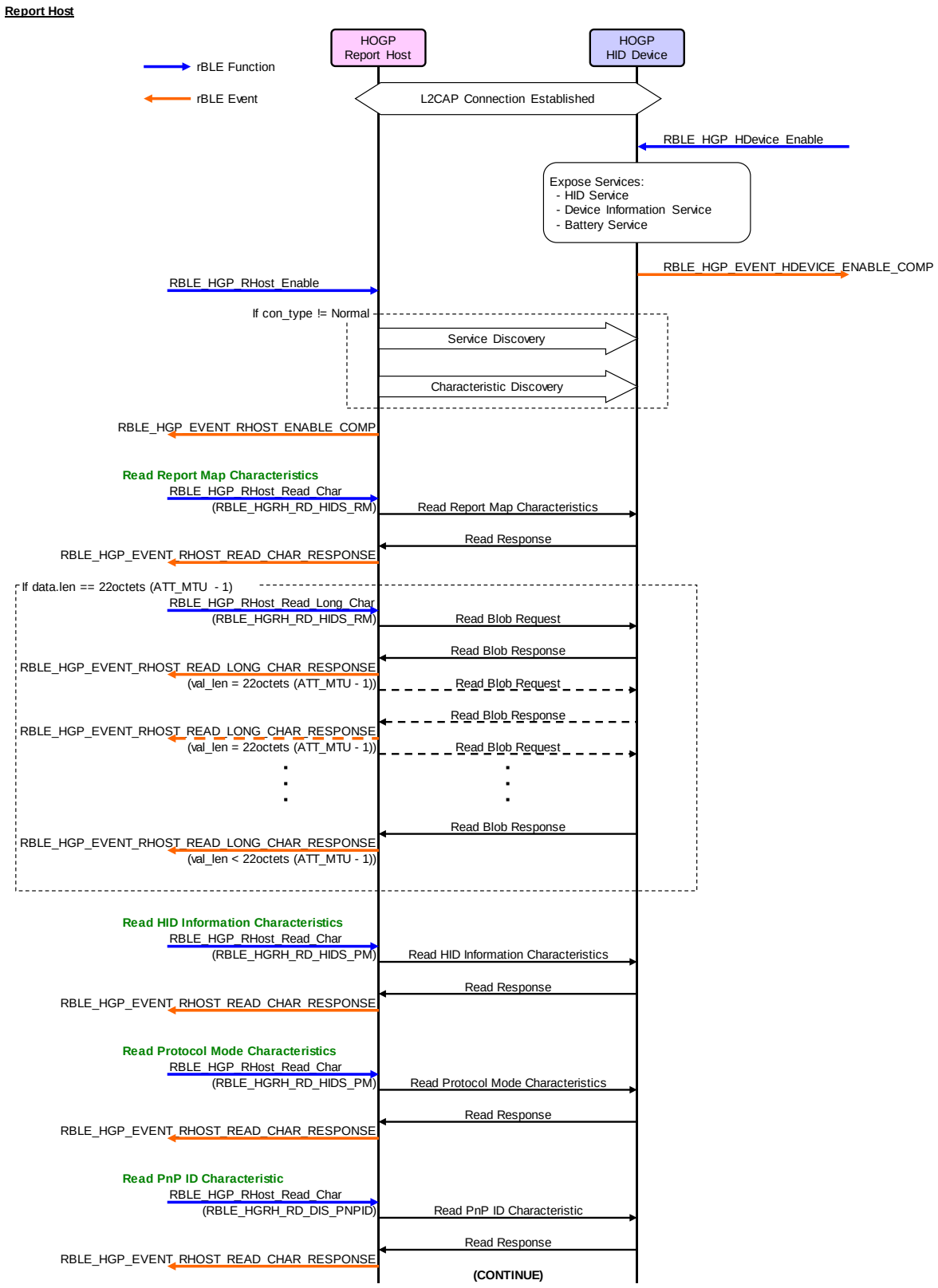

Figure 3-2 Example of Use Case in Which HOGP is Implemented by Using rBLE API (Report Host and HID Device)

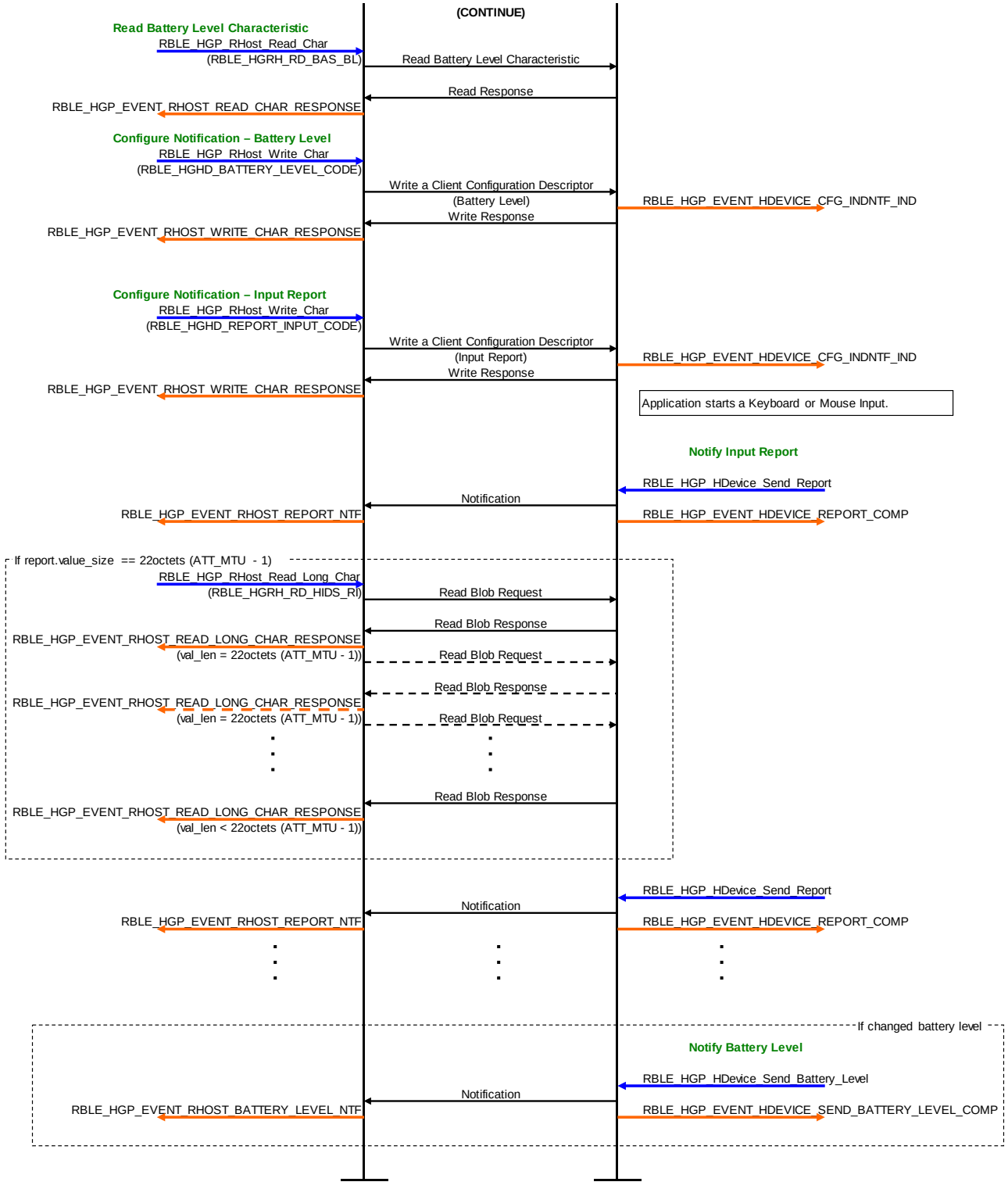

Figure 3-3 Example of Use Case in Which HOGP is Implemented by Using rBLE API (Report Host and HID Device)

4. Notes

Appendix A How to Read Definition Tables

This section shows how to read the tables that describes the rBLE API functions and events shown in this document.

A.1 How to Read Function Definition Tables

The following contents are included in the function definition tables:

The Parameters area describes the parameters specified for the function.
The italicized character strings on the left are the parameters of the function.
The meaning of each parameter is described on the far right following the variables.

The italicized character string(s) next to each parameter indicate the member(s) of the parameter (structure).

The values that can be specified for the parameter might be described between the parameter name and its description.

The function definition is shown at the top of the table in the row with the light green background. This area shows the function prototype.				
The operation of the function and the event reported after executing the function are described in this area.				
Parameters:				
	Parameter 1	Description of parameter 1		
	Parameter 2	*Member 1*	Value 1 that can be specified for member 1	Description of value 1 that can be specified for member 1
			Value 1 that can be specified for member 2	Description of value 1 that can be specified for member 2
		Member 2	Description of member 2	
Return:				
	Value 1 that might be returned	Description of value 1 that might be returned		
	Value 2 that might be returned	Description of value 2 that might be returned		

The Return area describes the values returned for the function.
The leftmost row shows the value that might be returned, and the next row describes the return value.

A.2 How to Read Event Definition Tables

The following contents are included in the event definition tables:

The Parameters area describes the parameters specified for the event.
The italicized character strings on the left show the parameters of the event parameter structure. The meaning of each parameter is described on the far right.

The italicized character string(s) next to each parameter indicate the member(s) of the parameter (structure).

The event definition is shown at the top of the table in the row with the orange background. This area shows the event type.			
The information reported by the event is described in this area.			
Parameters:			
	Parameter 1	Description of parameter 1	
	Parameter 2	*Member 1*	Description of member 1
		Member 2	Description of member 2
		Member 3	Description of member 3
	Parameter 3	Value 1 that can be specified for parameter 3	Description of value 1 that can be specified for parameter 3
		Value 2 that can be specified for parameter 3	Description of value 2 that can be specified for parameter 3

The values that can be specified for the parameter might be shown between the parameter name and its description.

Appendix B Referenced Documents

1. Bluetooth Core Specification v4.0, Bluetooth SIG
2. Find Me Profile Specification v1.0, Bluetooth SIG
3. Immediate Alert Service Specification v1.0, Bluetooth SIG
4. Proximity Profile Specification v1.0, Bluetooth SIG
5. Link Loss Service Specification v1.0, Bluetooth SIG
6. Tx Power Service Specification v1.0, Bluetooth SIG
7. Health Thermometer Profile Specification v1.0, Bluetooth SIG
8. Health Thermometer Service Specification v1.0, Bluetooth SIG
9. Device Information Service Specification v1.1, Bluetooth SIG
10. Blood Pressure Profile Specification v1.0, Bluetooth SIG
11. Blood Pressure Service Specification v1.0, Bluetooth SIG
12. HID over GATT Profile Specification v1.0, Bluetooth SIG
13. HID Service Specification v1.0, Bluetooth SIG
14. Battery Service Specification v1.0, Bluetooth SIG
15. Scan Parameters Profile Specification v1.0, Bluetooth SIG
16. Scan Parameters Service Specification v1.0, Bluetooth SIG
17. Bluetooth SIG Assigned Numbers https://www.bluetooth.org/Technical/AssignedNumbers/home.htm
18. Services & Characteristics UUID http://developer.bluetooth.org/gatt/Pages/default.aspx
19. Personal Health Devices Transcoding White Paper v1.2, Bluetooth SIG

Appendix C Terminology

Term	Description
Service	A service is provided from a GATT server to a GATT client. The GATT server exposes some characteristics as the interface. The service prescribes how to access the exposed characteristics.
Profile	A profile enables implementation of a use case by using one or more services. The services used are defined in the specifications of each profile.
Characteristic	A characteristic is a value used to identify services. The characteristics to be exposed and their formats are defined by each service.
Role	Each device takes the role prescribed by the profile or service in order to implement the specified use case.
Client Characteristic Configuration Descriptor	A descriptor is used to control notifications or indications of characteristic values that include the client characteristic configuration descriptor sent from the GATT server.
Connection Handle	The handle determined by the controller stack and is used to identify connection with a remote device. The valid handle range is between 0x0000 and 0x0EFF.

REVISION HISTORY	Bluetooth Low Energy Protocol Stack API Reference Manual: HOGP

Rev.	Date	Description	
		Page	Summary
1.00	Feb 15, 2013	---	First Edition issued
1.01	Mar 27, 2013	---	The description about the high-speed access to the service for a second or subsequent time is added.
1.02	Jun 28, 2013	---	Message sequence chart is added. Bookmark is added.
1.03	Sep 19, 2014	2 5 ---	The common definitions of profile are added. Definitions of client configuration characteristic value and connection type are deleted. Parameter description is changed to use the common definitions of profile.
1.04	Apr 17, 2015	2	The service definitions are updated.

Bluetooth Low Energy Protocol Stack
API Reference Manual: HOGP

Publication Date: Rev.1.04 Apr 17, 2015

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0

Bluetooth Low Energy Protocol Stack

Renesas Electronics Corporation

R01UW0093EJ0104