User's Manual

Bluetooth® Low Energy Protocol Stack

API Reference Manual: HRP

Renesas MCU

Target Device

RL78/G1D

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.1.02 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 Heart Rate profile (HRP) 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	R01UW0093E
API Reference Manual: ScPP	R01UW0094E
API Reference Manual: HRP	This manual
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	

All trademarks and registered trademarks are the property of their respective owners.
Bluetooth is a registered trademark of Bluetooth SIG, Inc. U.S.A.
EEPROM is a trademark of Renesas Electronics Corporation.
Windows, Windows NT and Windows XP are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT is a trademark of International Business Machines Corporation.

Contents

1. Overview 1

2. Common Definitions 2

2.1 Service Definitions 2

2.2 Status Definitions 4

3. Heart Rate Profile 5

3.1 Definitions 5

3.2 Functions 12

3.2.1 RBLE_HRP_Sensor_Enable 13

3.2.2 RBLE_HRP_Sensor_Disable 13

3.2.3 RBLE_HRP_Sensor_Send_Measurements 14

3.2.4 RBLE_HRP_Collector_Enable 15

3.2.5 RBLE_HRP_Collector_Disable 16

3.2.6 RBLE_HRP_Collector_Read_Char 17

3.2.7 RBLE_HRP_Collector_Write_Char 17

3.2.8 RBLE_HRP_Collector_Write_Control_Point 18

3.3 Events 19

3.3.1 RBLE_HRP_EVENT_SENSOR_ENABLE_COMP 20

3.3.2 RBLE_HRP_EVENT_SENSOR_DISABLE_COMP 20

3.3.3 RBLE_HRP_EVENT_SENSOR_ERROR_IND 20

3.3.4 RBLE_HRP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP 20

3.3.5 RBLE_HRP_EVENT_SENSOR_CHG_CP_IND 21

3.3.6 RBLE_HRP_EVENT_SENSOR_CFG_NTF_IND 21

3.3.7 RBLE_HRP_EVENT_SENSOR_COMMAND_DISALLOWED_IND 21

3.3.8 RBLE_HRP_EVENT_COLLECTOR_ENABLE_COMP 22

3.3.9 RBLE_HRP_EVENT_COLLECTOR_DISABLE_COMP 23

3.3.10 RBLE_HRP_EVENT_COLLECTOR_ERROR_IND 23

3.3.11 RBLE_HRP_EVENT_COLLECTOR_MEASUREMENTS_NTF 23

3.3.12 RBLE_HRP_EVENT_COLLECTOR_READ_CHAR_RESPONSE 24

3.3.13 RBLE_HRP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE 24

3.3.14 RBLE_HRP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND 24

3.4 Message Sequence Chart 25

4. Notes 26

Appendix A How to Read Definition Tables.....27

Appendix B Referenced Documents 29

Appendix C Terminology 30

1. Overview

This manual describes the API (Application Program Interface) of the Heart Rate profile (HRP) 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. Heart Rate Profile

This section describes the API of the Heart Rate profile. The Heart Rate profile is used to enable a data collection device to obtain data from a Heart Rate Sensor.

3.1 Definitions

This section describes the definitions used by the API of the Heart Rate profile.

- Declaration of the number of r-r interval the maximum storage

#define RBLE_HRP_RR_INTERVAL_MAX 0x09

- Declaration of enumerated type for HRP event types

```
enum RBLE_HRP_EVENT_TYPE_enum {
    RBLE_HRP_EVENT_SENSOR_ENABLE_COMP = 0x01,     Sensor enable completion event
                                                  (Parameter: Sensor_enable)
    RBLE_HRP_EVENT_SENSOR_DISABLE_COMP,           Sensor disable completion event
                                                  (Parameter: Sensor_disable)
    RBLE_HRP_EVENT_SENSOR_ERROR_IND,              Sensor error indication event
                                                  (Parameter: error_ind)
    RBLE_HRP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP,
                                                  Measurements send completion event
                                                  (Parameter: send_mesurements)
    RBLE_HRP_EVENT_SENSOR_CHG_CP_IND,             Control Point change indication event
                                                  (Parameter: meas_intv_chg_ind)
    RBLE_HRP_EVENT_SENSOR_CFG_NTF_IND,            Characteristic configuration change
                                                  indication event
                                                  (Parameter: hrps_cfg_ntf_ind)
    RBLE_HRP_EVENT_SENSOR_COMMAND_DISALLOWED_IND,
                                                  Command disallowed indication event
                                                  (Parameter: cmd_disallowed_ind)
    RBLE_HRP_EVENT_COLLECTOR_ENABLE_COMP = 0x81,
                                                  Collector enable completion event
                                                  (Parameter: collector_enable)
    RBLE_HRP_EVENT_COLLECTOR_DISABLE_COMP,        Collector disable completion event
                                                  (Parameter: collector_disable)
    RBLE_HRP_EVENT_COLLECTOR_ERROR_IND,           Collector error indication event
                                                  (Parameter: error_ind)
    RBLE_HRP_EVENT_COLLECTOR_MEASUREMENTS_NTF,    Measurement notification event
                                                  (Parameter: measurements_ntf)
    RBLE_HRP_EVENT_COLLECTOR_READ_CHAR_RESPONSE,
                                                  Characteristic value read request
                                                  response event
                                                  (Parameter: rd_char_resp)
    RBLE_HRP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE,
                                                  Characteristic value write request
                                                  response event
                                                  (Parameter: wr_char_resp)
```

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

- Declaration of data type for HRP event types

```
typedef uint8_t RBLE_HRP_EVENT_TYPE;
```

- Declaration of data type for HRP Sensor event callback function

```
typedef void ( *RBLE_HRPS_EVENT_HANDLER )( RBLE_HRPS_EVENT *event );
```

- Declaration of data type for HRP Collector event callback function

```
typedef void ( *RBLE_HRPC_EVENT_HANDLER )( RBLE_HRPC_EVENT *event );
```

- Declaration of enumerated type for heart rate service/device information service characteristic codes

```
enum RBLE_HRPC_RD_CHAR_CODE_enum {
    RBLE_HRPC_RD_HRS_HM_CFG = 0x00,             Heart Rate measurement
                                                notification
    RBLE_HRPC_RD_HRS_BSL,                       Body sensor location
    RBLE_HRPC_RD_DIS_MANUF,                     Sensor manufacturer name
    RBLE_HRPC_RD_DIS_MODEL,                     Sensor model number
    RBLE_HRPC_RD_DIS_SERNB,                     Sensor serial number
    RBLE_HRPC_RD_DIS_HWREV,                     Sensor hardware revision
    RBLE_HRPC_RD_DIS_FWREV,                     Sensor firmware revision
    RBLE_HRPC_RD_DIS_SWREV,                     Sensor software revision
    RBLE_HRPC_RD_DIS_SYSID,                     Sensor system ID
    RBLE_HRPC_RD_DIS_IEEE,                      Sensor IEEE certification
                                                information
};
```

- Declaration of enumerated type for heart rate service characteristic value settings

```
enum RBLE_HRPC_WR_CP_CHAR_CODE_enum {
    RBLE_HRPC_HRTRATE_CTRL_POINT_RESET = 0x01,  Reset Energy
};
```

- Heart rate service characteristic information structures

```
typedef struct RBLE_HRP_SENSOR_PARAM_t {
    uint16_t      hrtrate_meas_ntf_en;          Notification configuration value
                                                measurements
}RBLE_HRP_SENSOR_PARAM;
```

- Temperature information structures

```
typedef struct RBLE_HRP_MEASUREMENTS_INFO_t{
    uint8_t         flags;                      Data field flag
    uint8_t         rr_interval_num;            Number of r-r interval
    uint16_t        heart_rate_measure;         Heart Rate measurement value
```

```
    uint16_t         energy_expended;                Energy Expended
    uint16_t         rr_interval[RBLE_HRP_RR_INTERVAL_MAX];
                                                     r-r interval value
}RBLE_HRP_MEASUREMENTS_INFO;
```

- Heart rate service content structures

```
typedef struct RBLE_HRS_CONTENT_t{
    uint16_t        shdl;                       Heart rate service start handle
    uint16_t        ehdl;                       Heart rate service end handle
    uint16_t        hrtrate_meas_char_hdl;      Heart rate measurement characteristic
                                                handle
    uint16_t        hrtrate_meas_val_hdl;       Heart rate measurement characteristic
                                                value handle
    uint16_t        hrtrate_meas_cfg_hdl;       Heart rate measurement client
                                                characteristic configuration
                                                descriptor handle
    uint8_t         hrtrate_meas_prop;          Heart rate measurement characteristic
                                                property
    uint8_t         reserved;                   Reserved
    uint16_t        body_sensor_loc_char_hdl;   Body sensor location characteristic handle
    uint16_t        body_sensor_loc_val_hdl;    Body sensor location characteristic value
                                                handle
    uint8_t         body_sensor_loc_prop;       Body sensor location characteristic
                                                property
    uint8_t         reserved2;                  Reserved
    uint16_t        hrtrate_cp_char_hdl;        Heart rate control point
                                                characteristic handle
    uint16_t        hrtrate_cp_val_hdl;         Heart rate control point
                                                characteristic value handle
    uint8_t         hrtrate_cp_prop;            Heart rate control point
                                                characteristic property
    uint8_t         reserved3;                  Reserved
}RBLE_HRS_CONTENT;
```

- Device information service content structures

```
typedef struct RBLE_DIS_CONTENT_t {
    uint16_t        shdl;                           Device information service start
                                                    handle
    uint16_t        ehdl;                           Device information service end
                                                    handle
    uint16_t        sys_id_char_hdl;                System ID characteristic handle
    uint16_t        sys_id_val_hdl;                 System ID characteristic value
                                                    handle
    uint8_t         sys_id_prop;                    System ID characteristic property
    uint8_t         reserved;                       Reserved
    uint16_t        model_nb_char_hdl;              Model number characteristic handle
    uint16_t        model_nb_val_hdl;               Model number characteristic value
                                                    handle
    uint8_t         model_nb_prop;                  Model number characteristic property
```

```
    uint8_t         reserved2;                  Reserved
    uint16_t        serial_nb_char_hdl;         Serial number characteristic handle
    uint16_t        serial_nb_val_hdl;          Serial number characteristic value
                                                handle
    uint8_t         serial_nb_prop;             Serial number characteristic property
    uint8_t         reserved3;                  Reserved
    uint16_t        fw_rev_char_hdl;            Firmware revision characteristic
                                                handle
    uint16_t        fw_rev_val_hdl;             Firmware revision characteristic
                                                value handle
    uint8_t         fw_rev_prop;                Firmware revision characteristic
                                                property
    uint8_t         reserved4;                  Reserved
    uint16_t        hw_rev_char_hdl;            Hardware revision characteristic
                                                handle
    uint16_t        hw_rev_val_hdl;             Hardware revision characteristic
                                                value handle
    uint8_t         hw_rev_prop;                Hardware revision characteristic
                                                property
    uint8_t         reserved5;                  Reserved
    uint16_t        sw_rev_char_hdl;            Software revision characteristic
                                                handle
    uint16_t        sw_rev_val_hdl;             Software revision characteristic
                                                value handle
    uint8_t         sw_rev_prop;                Software revision characteristic
                                                property
    uint8_t         reserved6;                  Reserved
    uint16_t        manuf_name_char_hdl;        Manufacturer name characteristic
                                                handle
    uint16_t        manuf_name_val_hdl;         Manufacturer name characteristic
                                                value handle
    uint8_t         manuf_name_prop;            Manufacturer name characteristic
                                                property
    uint8_t         reserved7;                  Reserved
    uint16_t        ieee_certif_char_hdl;       IEEE certification characteristic
                                                handle
    uint16_t        ieee_certif_val_hdl;        IEEE certification characteristic
                                                value handle
    uint8_t         ieee_certif_prop;           IEEE certification characteristic
                                                property
    uint8_t         reserved8;                  Reserved
}RBLE_DIS_CONTENT;
```

- HRP Sensor event parameter structures

```
typedef struct RBLE_HRPS_EVENT_t {
    RBLE_HRP_EVENT_TYPE             type;               HRP event type
    uint8_t                         reserved;           Reserved
    union Event_Hrs_Parameter_u {
        Generic event
        RBLE_STATUS                 status;             Status

        Sensor enable completion event
        struct RBLE_HRP_Sensor_Enable_t{
            uint16_t                conhdl;             Connection handle
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
        }Sensor_enable;

        Sensor disable completion event
        struct RBLE_HRP_Sensor_Disable_t{
            uint16_t                conhdl;             Connection handle
            RBLE_HRP_SENSOR_PARAM   sensor_info;        Heart rate service
                                                        information
        }Sensor_disable;

        Sensor error indication event
        struct RBLE_HRP_Sensor_Error_Ind_t{
            uint16_t                conhdl;             Connection handle
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
        }error_ind;

        Sensor measured value send completion event
        struct RBLE_HRP_Sensor_Send_Measurements_t{
            uint16_t                conhdl;             Connection handle
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
        }send_measurements;

        Sensor control point change indication event
        struct RBLE_HRP_Sensor_Chg_Cp_Ind_t{
            uint16_t                conhdl;             Connection handle
            uint8_t                 cp_val;             Control point value
            uint8_t                 reserved;           Reserved
        }hrps_chg_cp_ind;
```

```
        Sensor configuration characteristic value notification event
        struct RBLE_HRP_Sensor_Cfg_Ntf_Ind_t{
            uint16_t                conhdl;             Connection handle
            uint16_t                cfg_val;            Configuration characteristic
                                                        value
        }hrps_cfg_ntf_ind;

        Sensor command disallowed indication event
        struct RBLE_HRP_Sensor_Command_Disallowed_Ind_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                opcode;             Opcode
        }cmd_disallowed_ind;
    } param;
} RBLE_HRPS_EVENT;
```

- HRP Collector event parameter structures

```
typedef struct RBLE_HRPC_EVENT_t {
    RBLE_HRP_EVENT_TYPE             type;               HRP event type
    uint8_t                         reserved;           Reserved
    union Event_Htc_Parameter_u {
        Generic event
        RBLE_STATUS                 status;             Status

        Collector enable completion event
        struct RBLE_HRP_Collector_Enable_t{
            uint16_t                conhdl;             Connection handle
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            RBLE_HRS_CONTENT        hrs;                Heart rate service
                                                        content
            RBLE_DIS_CONTENT        dis;                Device information service
                                                        content
        }collector_enable;

        Collector disable completion event
        struct RBLE_HRP_Collector_Disable_t{
            uint16_t                conhdl;             Connection handle
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
        }collector_disable;
```

Collector error indication event

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

Collector heart rate measurement information notification event

```
struct RBLE_HRP_Collector_Measurements_Ntf_t{
    uint16_t                conhdl;             Connection handle
    RBLE_HRP_MEASUREMENTS_INFO  measurements_info;
                                                Heart rate measurement
                                                information
}measurements_ntf;
```

Collector characteristic value read request response event

```
struct RBLE_HRP_Collector_Read_Char_Response_t{
    uint16_t                conhdl;             Connection handle
    uint8_t                 att_code;           Status
    uint8_t                 reserved;           Reserved
    RBLE_ATT_INFO_DATA      data;               Acquired characteristic data
}rd_char_resp;
```

Collector characteristic value write request response event

```
struct RBLE_HRP_Collector_Write_Char_Response_t{
    uint16_t                conhdl;             Connection handle
    uint8_t                 att_code;           Status
    uint8_t                 reserved;           Reserved
}wr_char_resp;
```

Collector command disallowed indication event

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

3.2 Functions

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

Table 3-1 API Functions Used by the HRP

RBLE_HRP_Sensor_Enable	Enables the Sensor role.
RBLE_HRP_Sensor_Disable	Disables the Sensor role.
RBLE_HRP_Sensor_Send_Measurements	Sends heart rate measurement information.
RBLE_HRP_Collector_Enable	Enables the Collector role.
RBLE_HRP_Collector_Disable	Disables the Collector role.
RBLE_HRP_Collector_Read_Char	Reads the characteristic value.
RBLE_HRP_Collector_Write_Char	Writes the characteristic value.
RBLE_HRP_Collector_Write_Control_Point	Sets the heart rate control point.

3.2.1 RBLE_HRP_Sensor_Enable

RBLE_STATUS RBLE_HRP_Sensor_Enable(uint16_t conhdl, uint8_t sec_lvl, uint8_t con_type, RBLE_HRP_SENSOR_PARAM *param, RBLE_HRPS_EVENT_HANDLER call_back)			
This function enables the HRP Sensor role. If the measurement result notification setting has been specified from the Collector, set the notification setting parameter to 0 to configure the connection. If this setting has been specified from the Sensor, perform a normal connection in accordance with the notification setting parameter. The result is reported by using the Sensor role enable completion event RBLE_HRP_EVENT_SENSOR_ENABLE_COMP.			
Parameters:			
conhdl	Connection handle		
sec_lvl	Security level		
con_type	RBLE_PRF_CON_DISCOVERY	Configuration connection	
	RBLE_PRF_CON_NORMAL	Normal connection	
**param*	*hrtrate_meas_ntf_en*	RBLE_PRF_STOP_NTFIND	Stop notification of heart rate information.
		RBLE_PRF_START_NTF	Start notification of heart rate information.
call_back	Specify the callback function that reports the HRP event.		
Return:			
RBLE_OK	Success		
RBLE_ERR	Error occurred in Sensor 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_HRP_Sensor_Disable

RBLE_STATUS RBLE_HRP_Sensor_Disable(uint16_t conhdl)	
This function disables the HRP Sensor role. The result is reported by using the Sensor role disable completion event RBLE_HRP_EVENT_SENSOR_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_HRP_Sensor_Send_Measurements

RBLE_STATUS RBLE_HRP_Sensor_Send_Measurements (uint16_t conhdl, RBLE_HRP_MEASUREMENTS_INFO *measurements_info)			
This function sends the measured value data from the heart rate Sensor. The result is reported by using the Sensor role measured value send completion event RBLE_HRP_EVENT_SENSOR_SEND_TEMP_COMP.			
Parameters:			
	conhdl	Connection handle	
	**measurements_ info*	*flags*	Flag that defines whether there is a data field in the characteristic value or not
		rr_interval_num	Number of r-r interval
		heart rate measure	Heart rate measurement value
		energy expended	Energy expended
		rr_interval [RBLE_HRP_RR_ INTERVAL_MAX]	r-r interval value
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.4 RBLE_HRP_Collector_Enable

RBLE_STATUS RBLE_HRP_Collector_Enable(uint16_t conhdl, uint8_t con_type,
RBLE_HRS_CONTENT *hrs, RBLE_DIS_CONTENT *dis, RBLE_HRPC_EVENT_HANDLER call_back)

This function enables the HRP Collector role and starts access to the service exposed by the HRP Sensor. The result is reported by using the Collector role enable completion event RBLE_HRP_EVENT_COLLECTOR_ENABLE_COMP.
When starting access to the service exposed by a Sensor to be connected for the first time, set 0 to the parameters of the service to configure the connection and to discover the service for the Sensor. If the handle information about the discovered service is saved and is used when the Sensor is connected normally for a second or subsequent time, detecting the service is skipped, which enables a high-speed access to the service.
While the Collector role is enabled, the service exposed by only one Sensor is accessible. To connect to more than one Sensor at the same time and access the services exposed by each Sensor, repeat enable/disable of the Collector 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 Sensor) and the handle information (which was saved when starting access to the service for the first time) as parameters.

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
**hrs*	*shdl*	Heart rate service start handle
	ehdl	Heart rate service end handle
	hrtrate_meas_char_hdl	Heart rate measurement characteristic handle
	hrtrate_meas_val_hdl	Heart rate measurement characteristic value handle
	hrtrate_meas_cfg_hdl	Heart rate measurement client characteristic configuration descriptor handle
	hrtrate_meas_prop	Heart rate measurement characteristic property
	body_sensor_loc_char_hdl	Body sensor location characteristic handle
	body_sensor_loc_val_hdl	Body sensor location characteristic value handle
	body_sensor_loc_prop	Body sensor location characteristic property
	hrtrate_cp_char_hdl	Heart rate control point characteristic handle
	hrtrate_cp_val_hdl	Heart rate control point characteristic value handle
	hrtrate_cp_prop	Heart rate control point characteristic property
**dis*	*shdl*	Device information service start handle
	ehdl	Device information service end handle
	sys_id_char_hdl	System ID characteristic handle
	sys_id_val_hdl	System ID characteristic value handle
	sys_id_prop	System ID characteristic property
	model_nb_char_hdl	Model number characteristic handle
	model_nb_val_hdl	Model number characteristic value handle
	model_nb_prop	Model number characteristic property
	serial_nb_char_hdl	Serial number characteristic handle
	serial_nb_val_hdl	Serial number characteristic value handle
	serial_nb_prop	Serial number characteristic property
	fw_rev_char_hdl	Firmware revision characteristic handle
	fw_rev_val_hdl	Firmware revision characteristic value handle

		fw_rev_prop	Firmware revision characteristic property
		hw_rev_char_hdl	Hardware revision characteristic handle
		hw_rev_val_hdl	Hardware revision characteristic value handle
		hw_rev_prop	Hardware revision characteristic property
		sw_rev_char_hdl	Software revision characteristic handle
		sw_rev_val_hdl	Software revision characteristic value handle
		sw_rev_prop	Software revision characteristic property
		manuf_name_char_hdl	Manufacturer name characteristic handle
		manuf_name_val_hdl	Manufacturer name characteristic value handle
		manuf_name_prop	Manufacturer name characteristic property
		ieee_certif_char_hdl	IEEE certification characteristic handle
		ieee_certif_val_hdl	IEEE certification characteristic value handle
		ieee_certif_prop	IEEE certification characteristic property
	call_back	Callback	
Return:			
	RBLE_OK	Success	
	RBLE_ERR	Error occurred in initialization processing	
	RBLE_PARAM_ERR	Invalid parameter	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.5 RBLE_HRP_Collector_Disable

RBLE_STATUS RBLE_HRP_Collector_Disable(uint16_t conhdl)		
This function disables the HRP Collector role and terminates the access to the service exposed by HRP Sensor. The result is reported by using the Collector role disable completion event RBLE_HRP_EVENT_COLLECTOR_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.

3.2.6 RBLE_HRP_Collector_Read_Char

RBLE_STATUS RBLE_HRP_Collector_Read_Char (uint16_t conhdl, uint8_t char_code)		
This function reads the characteristic value of the heart rate service and the device information service. The result is reported by using the characteristic value read request response event RBLE_HRP_EVENT_COLLECTOR_READ_CHAR_RESPONSE.		
Parameters:		
conhdl	Connection handle	
char_code	RBLE_HRPC_RD_HRS_HM_CFG	Heart rate measurement notification
	RBLE_HRPC_RD_HRS_BSL	Body sensor location
	RBLE_HRPC_RD_DIS_MANUF	Sensor manufacturer name
	RBLE_HRPC_RD_DIS_MODEL	Sensor model number
	RBLE_HRPC_RD_DIS_SERNB	Sensor serial number
	RBLE_HRPC_RD_DIS_HWREV	Sensor hardware revision
	RBLE_HRPC_RD_DIS_FWREV	Sensor firmware revision
	RBLE_HRPC_RD_DIS_SWREV	Sensor software revision
	RBLE_HRPC_RD_DIS_SYSID	Sensor system ID
	RBLE_HRPC_RD_DIS_IEEE	Sensor IEEE certification information
Return:		
RBLE_OK	Success	
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.7 RBLE_HRP_Collector_Write_Char

RBLE_STATUS RBLE_HRP_Collector_Write_Char(uint16_t conhdl, uint16_t cfg_val)		
This function writes a client characteristic configuration descriptor of the heart rate service. The result is reported by using the characteristic value write request response event RBLE_HRP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE.		
Parameters:		
conhdl	Connection handle	
cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification.
	RBLE_PRF_START_NTF	Start notification.
Return:		
RBLE_OK	Success	
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.8 RBLE_HRP_Collector_Write_Control_Point

RBLE_STATUS RBLE_HRP_Collector_Write_Control_Point (uint16_t conhdl, uint8_t cp_val)		
This function sets the heart rate control point characteristic value of the heart rate service. The result is reported by using the characteristic value write request response event RBLE_HRP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE.		
Parameters:		
	conhdl	Connection handle
	cp_val	Heart rate control point value
Return:		
	RBLE_OK	Success
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.3 Events

The following table shows the events defined for the HRP of rBLE and the following sections describe the events in detail.

Table 3-2 Events Defined for the HRP

RBLE_HRP_EVENT_SENSOR_ENABLE_COMP	Sensor role enable completion event
RBLE_HRP_EVENT_SENSOR_DISABLE_COMP	Sensor role disable completion event
RBLE_HRP_EVENT_SENSOR_ERROR_IND	Sensor role error indication event
RBLE_HRP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP	Heart rate measurement information send completion event
RBLE_HRP_EVENT_SENSOR_CHG_CP_IND	Heart rate control point change indication event
RBLE_HRP_EVENT_SENSOR_CFG_NTF_IND	Characteristic value indication event
RBLE_HRP_EVENT_SENSOR_COMMAND_DISALLOWED_IND	Sensor role command disallowed indication event
RBLE_HRP_EVENT_COLLECTOR_ENABLE_COMP	Collector role enable completion event
RBLE_HRP_EVENT_COLLECTOR_DISABLE_COMP	Collector role disable completion event
RBLE_HRP_EVENT_COLLECTOR_ERROR_IND	Collector role error indication event
RBLE_HRP_EVENT_COLLECTOR_MEASUREMENTS_NTF	Heart rate measurement information notification event
RBLE_HRP_EVENT_COLLECTOR_READ_CHAR_RESPONSE	Characteristic value read request response event
RBLE_HRP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE	Characteristic value write request response event
RBLE_HRP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND	Collector role command disallowed indication event

3.3.1 RBLE_HRP_EVENT_SENSOR_ENABLE_COMP

RBLE_HRP_EVENT_SENSOR_ENABLE_COMP	
This event reports the result of enabling the Sensor role (RBLE_HRP_Sensor_Enable).	
Parameters:	
status	Result of enabling the Sensor role (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_HRP_EVENT_SENSOR_DISABLE_COMP

RBLE_HRP_EVENT_SENSOR_DISABLE_COMP			
This event reports the result of disabling the Sensor role (RBLE_HRP_Sensor_Disable).			
Parameters:			
conhdl	Connection handle		
sensor_info	*hrtrate_meas_ntf_en*	RBLE_PRF_STOP_NTFIND	Stop notification of heart rate measurement information.
		RBLE_PRF_START_NTF	Start notification of heart rate measurement information.

3.3.3 RBLE_HRP_EVENT_SENSOR_ERROR_IND

RBLE_HRP_EVENT_SENSOR_ERROR_IND	
This event indicates an error code unique to the Sensor 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_HRP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP

RBLE_HRP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP	
This event reports completion of sending the measured value (RBLE_HRP_Sensor_Send_Measurements).	
Parameters:	
conhdl	Connection handle
status	Measured value 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*.)

3.3.5 RBLE_HRP_EVENT_SENSOR_CHG_CP_IND

RBLE_HRP_EVENT_SENSOR_CHG_CP_IND	
This event indicates that the value of the heart rate control point characteristic of the heart rate service has been changed by the Collector.	
Parameters:	
conhdl	Connection handle
cp_val	Heart rate control point characteristic value

3.3.6 RBLE_HRP_EVENT_SENSOR_CFG_NTF_IND

RBLE_HRP_EVENT_SENSOR_CFG_NTF_IND		
This event indicates that the value of the client characteristic configuration descriptor of the heart rate service has been set by the Collector.		
Parameters:		
conhdl	Connection handle	
cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification.
	RBLE_PRF_START_NTF	Start notification.

3.3.7 RBLE_HRP_EVENT_SENSOR_COMMAND_DISALLOWED_IND

RBLE_HRP_EVENT_SENSOR_COMMAND_DISALLOWED_IND		
This event indicates the error that occurs when a command executed by the Sensor role 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_HRP_SENSOR_ENABLE	Sensor role enable command
	RBLE_CMD_HRP_SENSOR_DISABLE	Sensor role disable command
	RBLE_CMD_HRP_SENSOR_SEND_MEASUREMENTS	Heart rate measurement data send command

3.3.8 RBLE_HRP_EVENT_COLLECTOR_ENABLE_COMP

RBLE_HRP_EVENT_COLLECTOR_ENABLE_COMP		
This event reports the result of enabling the Collector role (RBLE_HRP_Collector_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.		
Parameters:		
status	Result of enabling the Collector role (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	
hrs	*shdl*	Heart rate service start handle
	ehdl	Heart rate service end handle
	hrtrate_meas_char_hdl	Heart rate measurement characteristic handle
	hrtrate_meas_val_hdl	Heart rate measurement characteristic value handle
	hrtrate_meas_cfg_hdl	Heart rate measurement client characteristic configuration descriptor handle
	hrtrate_meas_prop	Heart rate measurement characteristic property
	body_sensor_loc_char_hdl	Body sensor location characteristic handle
	body_sensor_loc_val_hdl	Body sensor location characteristic value handle
	body_sensor_loc_prop	Body sensor location characteristic property
	hrtrate_cp_char_hdl	Heart rate control point characteristic handle
	hrtrate_cp_val_hdl	Heart rate control point characteristic value handle
	hrtrate_cp_prop	Heart rate control point characteristic property
dis	*shdl*	Device information service start handle
	ehdl	Device information service end handle
	sys_id_char_hdl	System ID characteristic handle
	sys_id_val_hdl	System ID characteristic value handle
	sys_id_prop	System ID characteristic property
	model_nb_char_hdl	Model number characteristic handle
	model_nb_val_hdl	Model number characteristic value handle
	model_nb_prop	Model number characteristic property
	serial_nb_char_hdl	Serial number characteristic handle
	serial_nb_val_hdl	Serial number characteristic value handle
	serial_nb_prop	Serial number characteristic property
	fw_rev_nb_char_hdl	Firmware revision characteristic handle
	fw_rev_nb_val_hdl	Firmware revision characteristic value handle
	fw_rev_nb_prop	Firmware revision characteristic property
	hw_rev_nb_char_hdl	Hardware revision characteristic handle
	hw_rev_nb_val_hdl	Hardware revision characteristic value handle
	hw_rev_nb_prop	Hardware revision characteristic property
	sw_rev_nb_char_hdl	Software revision characteristic handle
	sw_rev_nb_val_hdl	Software revision characteristic value handle
	sw_rev_nb_prop	Software revision characteristic property
	manuf_name_char_hdl	Manufacturer name characteristic handle
	manuf_name_val_hdl	Manufacturer name characteristic value handle

		manuf_name_prop	Manufacturer name characteristic property
		ieee_certif_char_hdl	IEEE certification characteristic handle
		ieee_certif_val_hdl	IEEE certification characteristic value handle
		ieee_certif_prop	IEEE certification characteristic property

3.3.9 RBLE_HRP_EVENT_COLLECTOR_DISABLE_COMP

RBLE_HRP_EVENT_COLLECTOR_DISABLE_COMP		
This event reports the result of disabling the Collector role (RBLE_HRP_Collector_Disable).		
Parameters:		
	status	Result of disabling the Collector role (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

3.3.10 RBLE_HRP_EVENT_COLLECTOR_ERROR_IND

RBLE_HRP_EVENT_COLLECTOR_ERROR_IND		
This event indicates an error code unique to the HRP Collector role.		
Parameters:		
	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*.)
	conhdl	Connection handle

3.3.11 RBLE_HRP_EVENT_COLLECTOR_MEASUREMENTS_NTF

RBLE_HRP_EVENT_COLLECTOR_MEASUREMENTS_NTF			
This event indicates the measured value sent from the Sensor.			
Parameters:			
	measurements_info	*flags*	Flag that defines whether there is a data field in the characteristic value or not
		rr_interval_num	Number of r-r interval
		heart_rate_mueasure	Heart rate measurement value
		energy_expended	Energy expended
		rr_interval[RBLE_HRP_RR_INTERVAL_MAX]	*r-r interval value*
	conhdl	Connection handle	

3.3.12 RBLE_HRP_EVENT_COLLECTOR_READ_CHAR_RESPONSE

RBLE_HRP_EVENT_COLLECTOR_READ_CHAR_RESPONSE			
This event reports the response to the characteristic value read request (RBLE_HRP_Collector_Read_Char). Read out the read 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	
data	*each_len*		Length of each result
	len		Data length
	data[RBLE_ATTM_MAX_VALUE]		Read characteristic data

3.3.13 RBLE_HRP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE

RBLE_HRP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE		
This event reports the response to the characteristic value write request (RBLE_HRP_Collector_Write_Char).		
Parameters:		
conhdl	Connection handle	
att_code	0x00	Characteristic value successfully written
	Other than 0x00	Error occurred when writing characteristic value

3.3.14 RBLE_HRP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND

RBLE_HRP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND		
This event indicates the error that occurs when a command executed by the Collector role 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_HRP_COLLECTOR_ENABLE	Collector role enable command
	RBLE_CMD_HRP_COLLECTOR_DISABLE	Collector role disable command
	RBLE_CMD_HRP_COLLECTOR_READ_CHAR	Characteristic read command
	RBLE_CMD_HRP_COLLECTOR_WRITE_CHAR	Characteristic write command
	RBLE_CMD_HRP_COLLECTOR_WRITE_CONTROL_POINT	Heart rate control point setup command

3.4 Message Sequence Chart

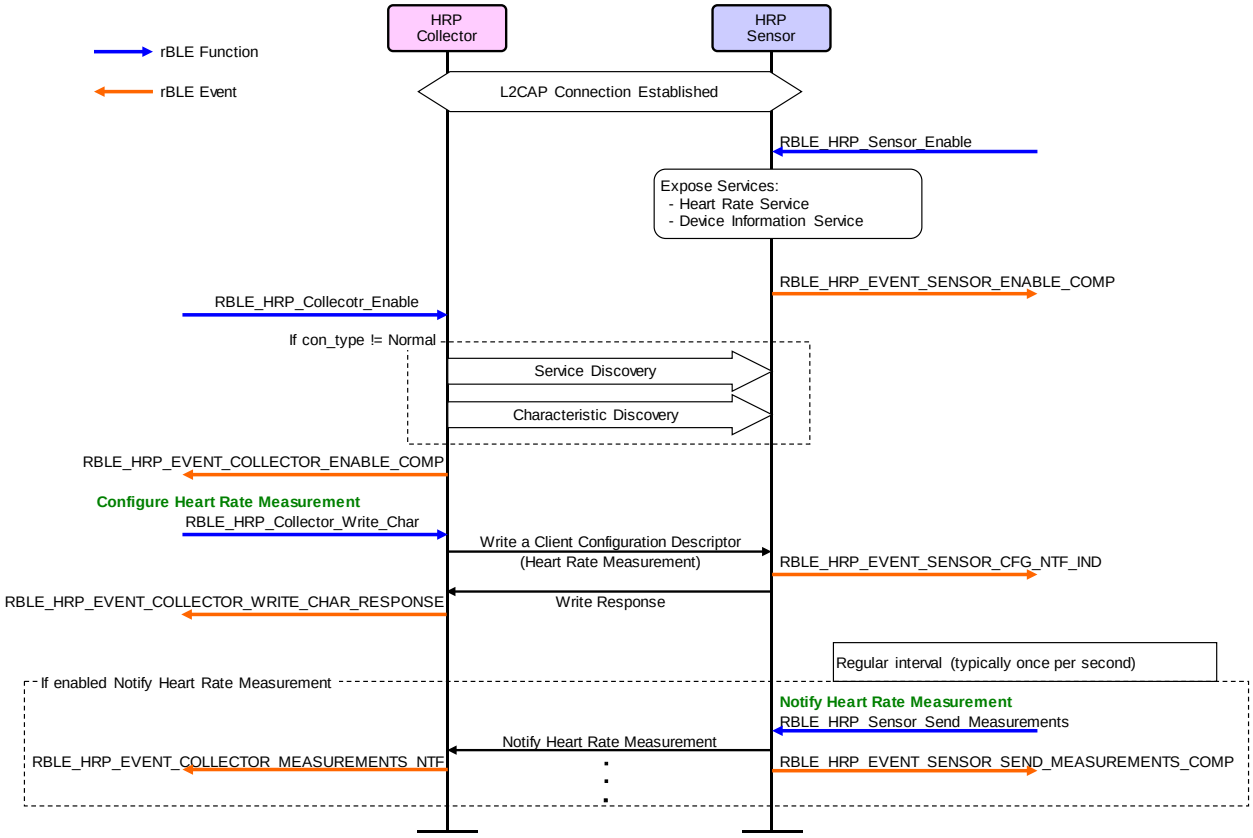

図 3-1 example of use case realization of HRP by using rBLE API

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. Heart Rate Profile Specification v1.0, Bluetooth SIG
18. Heart Rate Service Specification v1.0, Bluetooth SIG
19. Cycling Speed and Cadence Profile Specification v1.0, Bluetooth SIG
20. Cycling Speed and Cadence Service Specification v1.0, Bluetooth SIG
21. Cycling Power Profile Specification v0.9, Bluetooth SIG
22. Cycling Power Service Specification v0.9, Bluetooth SIG
23. Glucose Profile Specification v1.0, Bluetooth SIG
24. Glucose Service Specification v1.0, Bluetooth SIG
25. Bluetooth SIG Assigned Numbers https://www.bluetooth.org/Technical/AssignedNumbers/home.htm
26. Services & Characteristics UUID http://developer.bluetooth.org/gatt/Pages/default.aspx
27. 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.
Server Characteristic Configuration Descriptor	A descriptor is used to control broadcast of characteristic values that include the server 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: HRP

Rev.	Date	Description	
		Page	Summary
0.12	Apr 5, 2013	---	Provisional Edition issued
0.13	Apr 12, 2013	---	Bookmark is added.
1.00	Nov 29, 2013	---	First Edition issued
		25	3.4.Message Sequence Chart is added.
1.01	Sep 19, 2014	2	The common definitions of profile are added.
		5	Definitions of client configuration characteristic value and connection type are deleted.
		---	Parameter description is changed to use the common definitions of profile.
1.02	Apr 17, 2015	2	The service definitions are updated.

Bluetooth Low Energy Protocol Stack
API Reference Manual: HRP

Publication Date: Rev.1.02 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

© 2015 Renesas Electronics Corporation. All rights reserved.

Colophon 4.0

Bluetooth Low Energy Protocol Stack

Renesas Electronics Corporation

R01UW0097EJ0102