User's Manual

Bluetooth® Low Energy Protocol Stack

API Reference Manual: HTP

Renesas MCU

Target Device

RL78/G1D

Renesas Electronics
www.renesas.com

Rev.1.04 Apr 2015

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 Health Thermometer profile (HTP) 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	This manual
	API Reference Manual: BLP	R01UW0092E
	API Reference Manual: HOGP	R01UW0093E
	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 Health Thermometer profile (HTP) 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. Health Thermometer Profile

This section describes the API of the Health Thermometer profile. The Health Thermometer profile is used to enable a data collection device to obtain data from a thermometer sensor.

3.1 Definitions

This section describes the definitions used by the API of the Health Thermometer profile.

- Declaration of enumerated type for HTP event types

```
enum RBLE_HTP_EVENT_TYPE_enum {
    RBLE_HTP_EVENT_THERMOMETER_ENABLE_COMP = 0x01,
                                             Thermometer enable completion event
                                             (Parameter: thermometer_enable)
    RBLE_HTP_EVENT_THERMOMETER_DISABLE_COMP,     Thermometer disable completion event
                                             (Parameter: thermometer_disable)
    RBLE_HTP_EVENT_THERMOMETER_ERROR_IND,        Thermometer error indication event
                                             (Parameter: error_ind)
    RBLE_HTP_EVENT_THERMOMETER_SEND_TEMP_COMP,
                                             Temperature send completion event
                                             (Parameter: send_temp)
    RBLE_HTP_EVENT_THERMOMETER_REQ_MEASUREMENT_PERIOD_IND_COMP,
                                             Measurement period indication
                                             completion notification event
                                             (Parameter: send_meas_period)
    RBLE_HTP_EVENT_THERMOMETER_MEAS_INTV_CHG_IND,
                                             Measurement interval change
                                             indication event
                                             (Parameter: meas_intv_chg_ind)
    RBLE_HTP_EVENT_THERMOMETER_CFG_INDNTF_IND,
                                             Characteristic configuration change
                                             indication event
                                             (Parameter: htpt_cfg_indntf_ind)
    RBLE_HTP_EVENT_THERMOMETER_COMMAND_DISALLOWED_IND,
                                             Command disallowed indication event
                                             (Parameter: cmd_disallowed_ind)
    RBLE_HTP_EVENT_COLLECTOR_ENABLE_COMP = 0x81,
                                             Collector enable completion event
                                             (Parameter: collector_enable)
    RBLE_HTP_EVENT_COLLECTOR_DISABLE_COMP,       Collector disable completion event
                                             (Parameter: collector_disable)
    RBLE_HTP_EVENT_COLLECTOR_ERROR_IND,          Collector error indication event
                                             (Parameter: error_ind)
    RBLE_HTP_EVENT_COLLECTOR_TEMP_IND,           Measured temperature indication
                                             event
                                             (Parameter: temp_ind)
    RBLE_HTP_EVENT_COLLECTOR_MEAS_INTV_IND,      Measurement interval indication
                                             event
                                             (Parameter: meas_intv_ind)
    RBLE_HTP_EVENT_COLLECTOR_READ_CHAR_RESPONSE,
                                             Characteristic value read request
                                             response event
                                             (Parameter: rd_char_resp)
    RBLE_HTP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE,
                                             Characteristic value write request
                                             response event
                                             (Parameter: wr_char_resp)
    RBLE_HTP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND
                                             Command disallowed indication event
                                             (Parameter: cmd_disallowed_ind)
};
```

- Declaration of data type for HTP event types

```
typedef uint8_t RBLE_HTP_EVENT_TYPE;
```

- Declaration of data type for HTP Thermometer event callback function

```
typedef void ( *RBLE_HTPT_EVENT_HANDLER )( RBLE_HTPT_EVENT *event );
```

- Declaration of data type for HTP Collector event callback function

```
typedef void ( *RBLE_HTPC_EVENT_HANDLER )( RBLE_HTPC_EVENT *event );
```

- Declaration of enumerated type for temperature measurement flag field values

```
enum RBLE_HTPT_FLAG_enum {
    RBLE_HTPT_FLAG_CELSIUS = 0x00,                     Celsius
    RBLE_HTPT_FLAG_FAHRENHEIT = 0x01,                  Fahrenheit
    RBLE_HTPT_FLAG_TIME = 0x02,                        Time
    RBLE_HTPT_FLAG_TYPE = 0x04,                        Type
};
```

- Declaration of enumerated type for health thermometer service/device information service characteristic codes

```
enum RBLE_HTPC_RD_CHAR_CODE_enum {
    RBLE_HTPC_RD_HTS_TM_CFG = 0x00,                  Temperature measurement
                                                     indication
    RBLE_HTPC_RD_HTS_TT,                             Temperature type
    RBLE_HTPC_RD_HTS_IT_CFG,                         Intermediate temperature
                                                     information notification
    RBLE_HTPC_RD_HTS_MI,                             Measurement interval
    RBLE_HTPC_RD_HTS_MI_CFG,                         Measurement interval indication
    RBLE_HTPC_RD_HTS_VR,                             Measurement interval valid range
    RBLE_HTPC_RD_DIS_MANUF,                          Thermometer manufacturer name
    RBLE_HTPC_RD_DIS_MODEL,                          Thermometer model number
    RBLE_HTPC_RD_DIS_SERNB,                          Thermometer serial number
    RBLE_HTPC_RD_DIS_HWREV,                          Thermometer hardware revision
    RBLE_HTPC_RD_DIS_FWREV,                            Thermometer firmware revision
    RBLE_HTPC_RD_DIS_SWREV,                            Thermometer software revision
    RBLE_HTPC_RD_DIS_SYSID,                            Thermometer system ID
    RBLE_HTPC_RD_DIS_IEEE,                             Thermometer IEEE certification
                                                       information
};
```

- Declaration of enumerated type for health thermometer service characteristic value settings

```
enum RBLE_HTPC_WR_CHAR_CODE_enum {
    RBLE_HTPC_TEMP_MEAS_CODE = 0x01,                   Temperature measurement
                                                       indication setting
    RBLE_HTPC_INTERM_TEMP_CODE,                        Intermediate temperature
                                                       information notification setting
    RBLE_HTPC_MEAS_INTV_CODE,                          Measurement interval indication
                                                       setting
};
```

- Health thermometer service characteristic information structures

```
typedef struct RBLE_HTP_THERM_PARAM_t {
    uint16_t      temp_meas_ind_en;                    Temperature measurement
                                                       indication configuration value
    uint16_t      interm_temp_ntf_en;                  Intermediate temperature
                                                       notification configuration value
    uint16_t      meas_intv_ind_en;                    Measurement interval indication
                                                       configuration value
    uint16_t      meas_intv;                           Measurement interval
}RBLE_HTP_THERM_PARAM;
```

- Date and time information structures

```
typedef struct RBLE_DATE_TIME_t{
    uint16_t      year;                                Year
    uint8_t       month;                               Month
    uint8_t       day;                                 Day
    uint8_t       hour;                                Hour
    uint8_t       min;                                 Minute
    uint8_t       sec;                                 Second
    uint8_t       reserved;                            Reserved
}RBLE_DATE_TIME;
```

- Temperature information structures

```
typedef struct RBLE_HTP_TEMP_INFO_t{
    uint8_t         flag_stable_meas;                  Measurement-in-progress flag
    uint8_t         flags;                             Data field flag
    int32_t         temp_val;                          Measured value
    RBLE_DATE_TIME stamp;                              Time stamp
    uint8_t         type;                              Type
    uint8_t         reserved;                          Reserved
}RBLE_HTP_TEMP_INFO;
```

- Health thermometer service content structures

```
typedef struct RBLE_HTS_CONTENT_t{
    uint16_t        shdl;                       Health thermometer service start handle
    uint16_t        ehdl;                       Health thermometer service end handle
    uint16_t        temp_meas_char_hdl;         Temperature measurement characteristic
                                                handle
    uint16_t        temp_meas_val_hdl;          Temperature measurement characteristic
                                                value handle
    uint16_t        temp_meas_cfg_hdl;          Temperature measurement client
                                                characteristic configuration
                                                descriptor handle
    uint8_t         temp_meas_prop;             Temperature measurement characteristic
                                                property
    uint8_t         reserved;                   Reserved
    uint16_t        temp_type_char_hdl;         Temperature type characteristic handle
    uint16_t        temp_type_val_hdl;          Temperature type characteristic value
                                                handle
    uint8_t         temp_type_prop;             Temperature type characteristic
                                                property
    uint8_t         reserved2;                  Reserved
    uint16_t        interm_temp_char_hdl;       Intermediate temperature
                                                characteristic handle
    uint16_t        interm_temp_val_hdl;        Intermediate temperature
                                                characteristic value handle
    uint16_t        interm_temp_cfg_hdl;        Intermediate temperature client
                                                characteristic configuration
                                                descriptor handle
    uint8_t         interm_temp_prop;           Intermediate temperature
                                                characteristic property
    uint8_t         reserved3;                  Reserved
    uint16_t        meas_intv_char_hdl;         Measurement interval characteristic
                                                handle
    uint16_t        meas_intv_val_hdl;          Measurement interval characteristic
                                                value handle
    uint16_t        meas_intv_cfg_hdl;          Measurement interval client
                                                characteristic configuration
                                                descriptor handle
    uint16_t        valid_range_hdl;            Valid range descriptor handle
    uint8_t         meas_intv_prop;             Measurement interval characteristic
                                                property
    uint8_t         reserved4;                  Reserved
}RBLE_HTS_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;
```

- HTP Thermometer event parameter structures

```
typedef struct RBLE_HTPT_EVENT_t {
    RBLE_HTP_EVENT_TYPE             type;               HTP event type
    uint8_t                         reserved;           Reserved
    union Event_Htt_Parameter_u {
        Generic event
        RBLE_STATUS                 status;             Status

        Thermometer enable completion event
        struct RBLE_HTP_Thermometer_Enable_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                conhdl;             Connection handle
        }thermometer_enable;

        Thermometer disable completion event
        struct RBLE_HTP_Thermometer_Disable_t{
            uint16_t                conhdl;             Connection handle
            RBLE_HTP_THERM_PARAM therm_info;            Health thermometer service
                                                        information
        }thermometer_disable;

        Thermometer error indication event
        struct RBLE_HTP_Thermometer_Error_Ind_t{
            uint16_t                conhdl;             Connection handle
            RBLE_STATUS             status;             Status
        }error_ind;

        Temperature measured value send completion event
        struct RBLE_HTP_Thermometer_Send_Temp_t{
            uint16_t                conhdl;             Connection handle
            RBLE_STATUS             status;             Status
        }send_temp;

        Thermometer measurement period indication completion notification event
        struct RBLE_HTP_Thermometer_Req_Measurement_Period_Ind_t{
            uint16_t                conhdl;             Connection handle
            RBLE_STATUS             status;             Status
        }send_meas_period;

        Thermometer measurement interval change indication event
        struct RBLE_HTP_Thermometer_Meas_Intv_Chg_Ind_t{
            uint16_t                conhdl;             Connection handle
            uint16_t                intv;               Measurement interval
        }meas_intv_chg_ind;
        Thermometer configuration characteristic value indication event
        struct RBLE_HTP_Thermometer_Cfg_Indntf_Ind_t{
            uint16_t                conhdl;             Connection handle
            uint8_t                 char_code;          Characteristic value code
            uint8_t                 reserved;           Reserved
            uint16_t                cfg_val;            Configuration characteristic
                                                        value
        }htpt_cfg_indntf_ind;

        Thermometer command disallowed indication event
        struct RBLE_HTP_Thermometer_Command_Disallowed_Ind_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                opcode;             Opcode
        }cmd_disallowed_ind;
    } param;
} RBLE_HTPT_EVENT;
```

- HTP Collector event parameter structures

```
typedef struct RBLE_HTPC_EVENT_t {
    RBLE_HTP_EVENT_TYPE             type;               HTP event type
    uint8_t                         reserved;           Reserved
    union Event_Htc_Parameter_u {
        Generic event
        RBLE_STATUS                 status;             Status

        Collector enable completion event
        struct RBLE_HTP_Collector_Enable_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                conhdl;             Connection handle
            RBLE_HTS_CONTENT        hts;                Health thermometer service
                                                        content
            RBLE_DIS_CONTENT        dis;                Device information service
                                                        content
        }collector_enable;

        Collector disable completion event
        struct RBLE_HTP_Collector_Disable_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                conhdl;             Connection handle
        }collector_disable;
        Collector error indication event
        struct RBLE_HTP_Collector_Error_Ind_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                conhdl;             Connection handle
        }error_ind;

        Collector temperature measurement information indication event
        struct RBLE_HTP_Collector_Temp_Ind_t{
            uint16_t                conhdl;             Connection handle
            RBLE_HTP_TEMP_INFO      temp_info;          Temperature measurement
                                                        information
        }temp_ind;

        Collector measurement interval indication event
        struct RBLE_HTP_Collector_Meas_Intv_Ind_t{
            uint16_t                conhdl;             Connection handle
            uint16_t                intv;               Measurement interval
        }meas_intv_ind;

        Collector characteristic value read request response event
        struct RBLE_HTP_Collector_Read_Char_Response_t{
            uint16_t                conhdl;             Connection handle
            uint8_t                 att_code;           Status
            RBLE_ATT_INFO_DATA      data;               Acquired characteristic data
        }rd_char_resp;

        Collector characteristic value write request response event
        struct RBLE_HTP_Collector_Write_Char_Response_t{
            uint16_t                conhdl;             Connection handle
            uint8_t                 att_code;           Status
        }wr_char_resp;

        Collector command disallowed indication event
        struct RBLE_HTP_Collector_Command_Disallowed_Ind_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                opcode;             Opcode
        }cmd_disallowed_ind;
    } param;
} RBLE_HTPC_EVENT;
```

3.2 Functions

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

Table 3-1 API Functions Used by the HTP

RBLE_HTP_Thermometer_Enable	Enables the Thermometer role.
RBLE_HTP_Thermometer_Disable	Disables the Thermometer role.
RBLE_HTP_Thermometer_Send_Temp	Sends temperature measurement information.
RBLE_HTP_Thermometer_Req_Measurement_Period_Ind	Sends the measurement period.
RBLE_HTP_Collector_Enable	Enables the Collector role.
RBLE_HTP_Collector_Disable	Disables the Collector role.
RBLE_HTP_Collector_Read_Char	Reads the characteristic value.
RBLE_HTP_Collector_Write_Char	Writes the characteristic value.
RBLE_HTP_Collector_Set_Measurement_Period	Sets the measurement period.

3.2.1 RBLE_HTP_Thermometer_Enable

RBLE_STATUS RBLE_HTP_Thermometer_Enable(uint16_t conhdl, uint8_t sec_lvl, uint8_t con_type,
    RBLE_HTP_THERM_PARAM *param, RBLE_HTPT_EVENT_HANDLER call_back)

This function enables the HTP Thermometer role.
If the measurement result indication and intermediate temperature information notification setting, or the measurement interval information, has been specified from the Collector, set the indication/notification setting parameter to 0 to configure the connection. If this setting or information has been specified from the Thermometer, perform a normal connection in accordance with the indication/notification setting parameter.
The result is reported by using the Thermometer role enable completion event RBLE_HTP_EVENT_THERMOMETER_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*	*temp_meas_ind_en*	RBLE_PRF_STOP_NTFIND	Stop notification/ indication of temperature information.
		RBLE_PRF_START_IND	Start indication of temperature information.
	interm_temp_ntf_en	RBLE_PRF_STOP_NTFIND	Stop notification/ indication of temperature information.
		RBLE_PRF_START_NTF	Start notification of temperature information.
	meas_intv_ind_en	RBLE_PRF_STOP_NTFIND	Stop notification/ indication of measurement interval information.
		RBLE_PRF_START_IND	Start indication of measurement interval information.
	meas_intv	Measurement interval	
call_back	Specify the callback function that reports the HTP event.		

Return:

RBLE_OK	Success
RBLE_ERR	Error occurred in Thermometer 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_HTP_Thermometer_Disable

RBLE_STATUS RBLE_HTP_Thermometer_Disable(uint16_t conhdl)		
This function disables the HTP Thermometer role. The result is reported by using the Thermometer role disable completion event RBLE_HTP_EVENT_THERMOMETER_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_HTP_Thermometer_Send_Temp

RBLE_STATUS RBLE_HTP_Thermometer_Send_Temp(uint16_t conhdl, RBLE_HTP_TEMP_INFO *temp_info)				
This function sends the measured value data from the thermometer. To send intermediate temperature information from the thermometer, set flag_stable_meas to 0 and save the temperature information in temp_val before executing this function. To send the measurement result after the thermometer has completed measuring the temperature, set flag_stable_meas to 1 and save the temperature information in temp_val before executing this function. The result is reported by using the Thermometer role measured value send completion event RBLE_HTP_EVENT_THERMOMETER_SEND_TEMP_COMP.				
Parameters:				
	conhdl	Connection handle		
	**temp_info*	*flag_stable_meas*		Flag indicating that measurement is in progress (0) or that measurement is complete (1)
		flags		Flag that defines whether there is a data field in the characteristic value or not
		temp_val		Temperature information
		stamp	*year*	Year
			month	Month
			day	Day
			hour	Hour
			min	Minute
			sec	Second
		type		Temperature type
Return:				
	RBLE_OK	Success		
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.		

3.2.4 RBLE_HTP_Thermometer_Req_Measurement_Period_Ind

RBLE_STATUS RBLE_HTP_Thermometer_Req_Measurement_Period_Ind(uint16_t conhdl)		
This function sends the measurement period value. The result is reported by using the measurement period indication completion event RBLE_HTP_EVENT_THERMOMETER_REQ_MEASUREMENT_PERIOD_IND_COMP.		
Parameters:		
	conhdl	Connection handle
Return:		
	RBLE_OK	Success
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.5 RBLE_HTP_Collector_Enable

```
RBLE_STATUS RBLE_HTP_Collector_Enable(uint16_t conhdl, uint8_t con_type,
        RBLE_HTS_CONTENT *hts, RBLE_DIS_CONTENT *dis, RBLE_HTPC_EVENT_HANDLER call_back)
```

This function enables the HTP Collector role and starts access to the service exposed by the HTP Thermometer. The result is reported by using the Collector role enable completion event RBLE_HTP_EVENT_COLLECTOR_ENABLE_COMP.
When starting access to the service exposed by a Thermometer 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 Thermometer. If the handle information about the discovered service is saved and is used when the Thermometer 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 Thermometer is accessible. To connect to more than one Thermometer at the same time and access the services exposed by each Thermometer, 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 Thermometer) 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
**hts*	*shdl*	Health thermometer service start handle
	ehdl	Health thermometer service end handle
	temp_meas_char_hdl	Temperature measurement characteristic handle
	temp_meas_val_hdl	Temperature measurement characteristic value handle
	temp_meas_cfg_hdl	Temperature measurement client characteristic configuration descriptor handle
	temp_meas_prop	Temperature measurement characteristic property
	temp_type_char_hdl	Temperature type characteristic handle
	temp_type_val_hdl	Temperature type characteristic value handle
	temp_type_prop	Temperature type characteristic property
	interm_temp_char_hdl	Intermediate temperature characteristic handle
	interm_temp_val_hdl	Intermediate temperature characteristic value handle
	interm_temp_cfg_hdl	Intermediate temperature client characteristic configuration descriptor handle
	interm_temp_prop	Intermediate temperature characteristic property
	meas_intv_char_hdl	Measurement interval characteristic handle
	meas_intv_val_hdl	Measurement interval characteristic value handle
	meas_intv_cfg_hdl	Measurement interval client characteristic configuration descriptor handle
	valid_range_hdl	Valid range descriptor handle
	meas_intv_prop	Measurement interval 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.	

3.2.6 RBLE_HTP_Collector_Disable

RBLE_STATUS RBLE_HTP_Collector_Disable(uint16_t conhdl)		
This function disables the HTP Collector role and terminates the access to the service exposed by HTP Thermometer. The result is reported by using the Collector role disable completion event RBLE_HTP_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.

3.2.7 RBLE_HTP_Collector_Read_Char

RBLE_STATUS RBLE_HTP_Collector_Read_Char (uint16_t conhdl, uint8_t char_code)			
This function reads the characteristic value of the health thermometer service and the device information service. The result is reported by using the characteristic value read request response event RBLE_HTP_EVENT_COLLECTOR_READ_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	char_code	RBLE_HTPC_RD_HTS_TM_CFG	Temperature measurement indication
		RBLE_HTPC_RD_HTS_TT	Temperature type
		RBLE_HTPC_RD_HTS_IT_CFG	Intermediate temperature information notification
		RBLE_HTPC_RD_HTS_MI	Measurement interval
		RBLE_HTPC_RD_HTS_MI_CFG	Measurement interval indication
		RBLE_HTPC_RD_HTS_VR	Measurement interval valid range
		RBLE_HTPC_RD_DIS_MANUF	Thermometer manufacturer name
		RBLE_HTPC_RD_DIS_MODEL	Thermometer model number
		RBLE_HTPC_RD_DIS_SERNB	Thermometer serial number
		RBLE_HTPC_RD_DIS_HWREV	Thermometer hardware revision
		RBLE_HTPC_RD_DIS_FWREV	Thermometer firmware revision
		RBLE_HTPC_RD_DIS_SWREV	Thermometer software revision
		RBLE_HTPC_RD_DIS_SYSID	Thermometer system ID
		RBLE_HTPC_RD_DIS_IEEE	Thermometer IEEE certification information
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.8 RBLE_HTP_Collector_Write_Char

RBLE_STATUS RBLE_HTP_Collector_Write_Char(uint16_t conhdl, uint8_t char_code, uint16_t cfg_val)		
This function writes each client characteristic configuration descriptor of the health thermometer service. The result is reported by using the characteristic value write request response event RBLE_HTP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE.		
Parameters:		
conhdl	Connection handle	
char_code	RBLE_HTPC_TEMP_MEAS_CODE	Temperature measurement indication setting
	RBLE_HTPC_INTERM_TEMP_CODE	Intermediate temperature information notification setting
	RBLE_HTPC_MEAS_INTV_CODE	Measurement interval indication setting
cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification or indication
	RBLE_PRF_START_NTF	Start notification
	RBLE_PRF_START_IND	Start indication
Return:		
RBLE_OK	Success	
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.9 RBLE_HTP_Collector_Set_Measurement_Period

RBLE_STATUS RBLE_HTP_Collector_Set_Measurement_Period(uint16_t conhdl, uint16_t intv)	
This function sets the measurement period characteristic value of the health thermometer service. The value set must be within the measurement period valid range.	
Parameters:	
conhdl	Connection handle
intv	Measurement interval
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 HTP of rBLE and the following sections describe the events in detail.

Table 3-2 Events Defined for the HTP

RBLE_HTP_EVENT_THERMOMETER_ENABLE_COMP	Thermometer role enable completion event
RBLE_HTP_EVENT_THERMOMETER_DISABLE_COMP	Thermometer role disable completion event
RBLE_HTP_EVENT_THERMOMETER_ERROR_IND	Thermometer role error indication event
RBLE_HTP_EVENT_THERMOMETER_SEND_TEMP_COMP	Temperature measurement information send completion event
RBLE_HTP_EVENT_THERMOMETER_REQ_MEASUREMENT_PERIOD_IND_COMP	Measurement period indication completion notification event
RBLE_HTP_EVENT_THERMOMETER_MEAS_INTV_CHG_IND	Measurement interval change indication event
RBLE_HTP_EVENT_THERMOMETER_CFG_INDNTF_IND	Characteristic value indication event
RBLE_HTP_EVENT_THERMOMETER_COMMAND_DISALLOWED_IND	Thermometer role command disallowed indication event
RBLE_HTP_EVENT_COLLECTOR_ENABLE_COMP	Collector role enable completion event
RBLE_HTP_EVENT_COLLECTOR_DISABLE_COMP	Collector role disable completion event
RBLE_HTP_EVENT_COLLECTOR_ERROR_IND	Collector role error indication event
RBLE_HTP_EVENT_COLLECTOR_TEMP_IND	Temperature measurement information indication event
RBLE_HTP_EVENT_COLLECTOR_MEAS_INTV_IND	Measurement interval indication event
RBLE_HTP_EVENT_COLLECTOR_READ_CHAR_RESPONSE	Characteristic value read request response event
RBLE_HTP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE	Characteristic value write request response event
RBLE_HTP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND	Collector role command disallowed indication event

3.3.1 RBLE_HTP_EVENT_THERMOMETER_ENABLE_COMP

RBLE_HTP_EVENT_THERMOMETER_ENABLE_COMP		
This event reports the result of enabling the Thermometer role (RBLE_HTP_Thermometer_Role_Enable).		
Parameters:		
	status	Result of enabling the Thermometer 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_HTP_EVENT_THERMOMETER_DISABLE_COMP

RBLE_HTP_EVENT_THERMOMETER_DISABLE_COMP				
This event reports the result of disabling the Thermometer role (RBLE_HTP_Thermometer_Role_Disable).				
Parameters:				
	conhdl	Connection handle		
	therm_info	*temp_meas_ind_en*	RBLE_PRF_STOP_NTFIND	Stop notification/indication of the measurement result.
			RBLE_PRF_START_IND	Start indication of the measurement result.
		interm_temp_ntf_en	RBLE_PRF_STOP_NTFIND	Stop notification/indication of intermediate temperature information.
			RBLE_PRF_START_NTF	Start notification of intermediate temperature information.
		meas_intv_ind_en	RBLE_PRF_STOP_NTFIND	Stop notification/indication of measurement interval.
			RBLE_PRF_START_IND	Start indication of measurement interval.
		meas_intv	Measurement interval	

3.3.3 RBLE_HTP_EVENT_THERMOMETER_ERROR_IND

RBLE_HTP_EVENT_THERMOMETER_ERROR_IND		
This event indicates an error code unique to the Thermometer 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_HTP_EVENT_THERMOMETER_SEND_TEMP_COMP

RBLE_HTP_EVENT_THERMOMETER_SEND_TEMP_COMP		
This event reports completion of sending the measured value (RBLE_HTP_Thermometer_Send_Temp).		
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_HTP_EVENT_THERMOMETER_REQ_MEASUREMENT_PERIOD_IND_COMP

RBLE_HTP_EVENT_THERMOMETER_REQ_MEASUREMENT_PERIOD_IND_COMP		
This event reports completion of indicating the measurement period (RBLE_HTP_Thermometer_Req_Measurement_Period_Ind).		
Parameters:		
	conhdl	Connection handle
	status	Measurement period indication completion result (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)

3.3.6 RBLE_HTP_EVENT_THERMOMETER_MEAS_INTV_CHG_IND

RBLE_HTP_EVENT_THERMOMETER_MEAS_INTV_CHG_IND		
This event indicates that the value of the measurement interval characteristic of the health thermometer service has been changed by the Collector.		
Parameters:		
	conhdl	Connection handle
	intv	Health thermometer service measurement interval characteristic value

3.3.7 RBLE_HTP_EVENT_THERMOMETER_CFG_INDNTF_IND

RBLE_HTP_EVENT_THERMOMETER_CFG_INDNTF_IND			
This event indicates that the value of the client characteristic configuration descriptor of the health thermometer service has been set by the Collector.			
Parameters:			
	conhdl	Connection handle	
	char_code	RBLE_HTPC_TEMP_MEAS_CODE	Temperature measurement indication setting
		RBLE_HTPC_INTERM_TEMP_CODE	Intermediate temperature information notification setting
		RBLE_HTPC_MEAS_INTV_CODE	Measurement interval indication setting
	cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification or indication.
		RBLE_PRF_START_NTF	Start notification.
		RBLE_PRF_START_IND	Start indication.

3.3.8 RBLE_HTP_EVENT_THERMOMETER_COMMAND_DISALLOWED_IND

RBLE_HTP_EVENT_THERMOMETER_COMMAND_DISALLOWED_IND		
This event indicates the error that occurs when a command executed by the Thermometer 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_HTP_THERMOMETER_ENABLE	Thermometer role enable command
	RBLE_CMD_HTP_THERMOMETER_DISABLE	Thermometer role disable command
	RBLE_CMD_HTP_THERMOMETER_SEND_TEMP	Temperature data send command
	RBLE_CMD_HTP_THERMOMETER_REQ_MEASURE MENT_PERIOD_IND	Measurement period set command

3.3.9 RBLE_HTP_EVENT_COLLECTOR_ENABLE_COMP

RBLE_HTP_EVENT_COLLECTOR_ENABLE_COMP		
This event reports the result of enabling the Collector role (RBLE_HTP_Collector_Role_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	
hts	*shdl*	Health thermometer service start handle
	ehdl	Health thermometer service end handle
	temp_meas_char_hdl	Temperature measurement characteristic handle
	temp_meas_val_hdl	Temperature measurement characteristic value handle
	temp_meas_cfg_hdl	Temperature measurement client characteristic configuration descriptor handle
	temp_meas_prop	Temperature measurement characteristic property
	temp_type_char_hdl	Temperature type characteristic handle
	temp_type_val_hdl	Temperature type characteristic value handle
	temp_type_prop	Temperature type characteristic property
	interm_temp_char_hdl	Intermediate temperature characteristic handle
	interm_temp_val_hdl	Intermediate temperature characteristic value handle
	interm_temp_cfg_hdl	Intermediate temperature client characteristic configuration descriptor handle
	interm_temp_prop	Intermediate temperature characteristic property
	meas_intv_char_hdl	Measurement interval characteristic handle
	meas_intv_val_hdl	Measurement interval characteristic value handle
	meas_intv_cfg_hdl	Measurement interval client characteristic configuration descriptor handle
	valid_range_hdl	Valid range descriptor handle
	meas_intv_prop	Measurement interval 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

3.3.10 RBLE_HTP_EVENT_COLLECTOR_DISABLE_COMP

RBLE_HTP_EVENT_COLLECTOR_DISABLE_COMP		
This event reports the result of disabling the Collector role (RBLE_HTP_Collector_Role_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

3.3.11 RBLE_HTP_EVENT_COLLECTOR_ERROR_IND

RBLE_HTP_EVENT_COLLECTOR_ERROR_IND		
This event indicates an error code unique to the HTP 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

3.3.12 RBLE_HTP_EVENT_COLLECTOR_TEMP_IND

RBLE_HTP_EVENT_COLLECTOR_TEMP_IND			
This event indicates the measured value sent from the Thermometer. When sending intermediate temperature information, the temperature information is reported with flag_stable_meas set to 0 and then saved in temp_val. When sending the temperature information after measurement is complete, the temperature information is reported with flag_stable_meas is set to 1 and then saved in temp_val.			
Parameters:			
temp_info	*flag_stable_meas*		Flag indicating that measurement is in progress (0) or that measurement is complete (1)
	flags		Flag that defines whether there is a data field in the characteristic value or not
	temp_val		Temperature information
	stamp	*year*	Year
		month	Month
		day	Day
		hour	Hour
		min	Minute
		sec	Second
	type		Temperature type
conhdl	Connection handle		

3.3.13 RBLE_HTP_EVENT_COLLECTOR_MEAS_INTV_IND

RBLE_HTP_EVENT_COLLECTOR_MEAS_INTV_IND	
This event indicates the measurement interval sent from the Thermometer.	
Parameters:	
conhdl	Connection handle
intv	Measurement interval

3.3.14 RBLE_HTP_EVENT_COLLECTOR_READ_CHAR_RESPONSE

RBLE_HTP_EVENT_COLLECTOR_READ_CHAR_RESPONSE		
This event reports the response to the characteristic value read request (RBLE_HTP_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

3.3.15 RBLE_HTP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE

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

3.3.16 RBLE_HTP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND

RBLE_HTP_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_HTP_COLLECTOR_ENABLE	Collector role enable command
	RBLE_CMD_HTP_COLLECTOR_DISABLE	Collector role disable command
	RBLE_CMD_HTP_COLLECTOR_READ_CHAR	Characteristic read command
	RBLE_CMD_HTP_COLLECTOR_WRITE_CHAR	Characteristic write command
	RBLE_CMD_HTP_COLLECTOR_SET_MEASUREMENT_PERIOD	Measurement period setup command

3.4 Message Sequence Chart

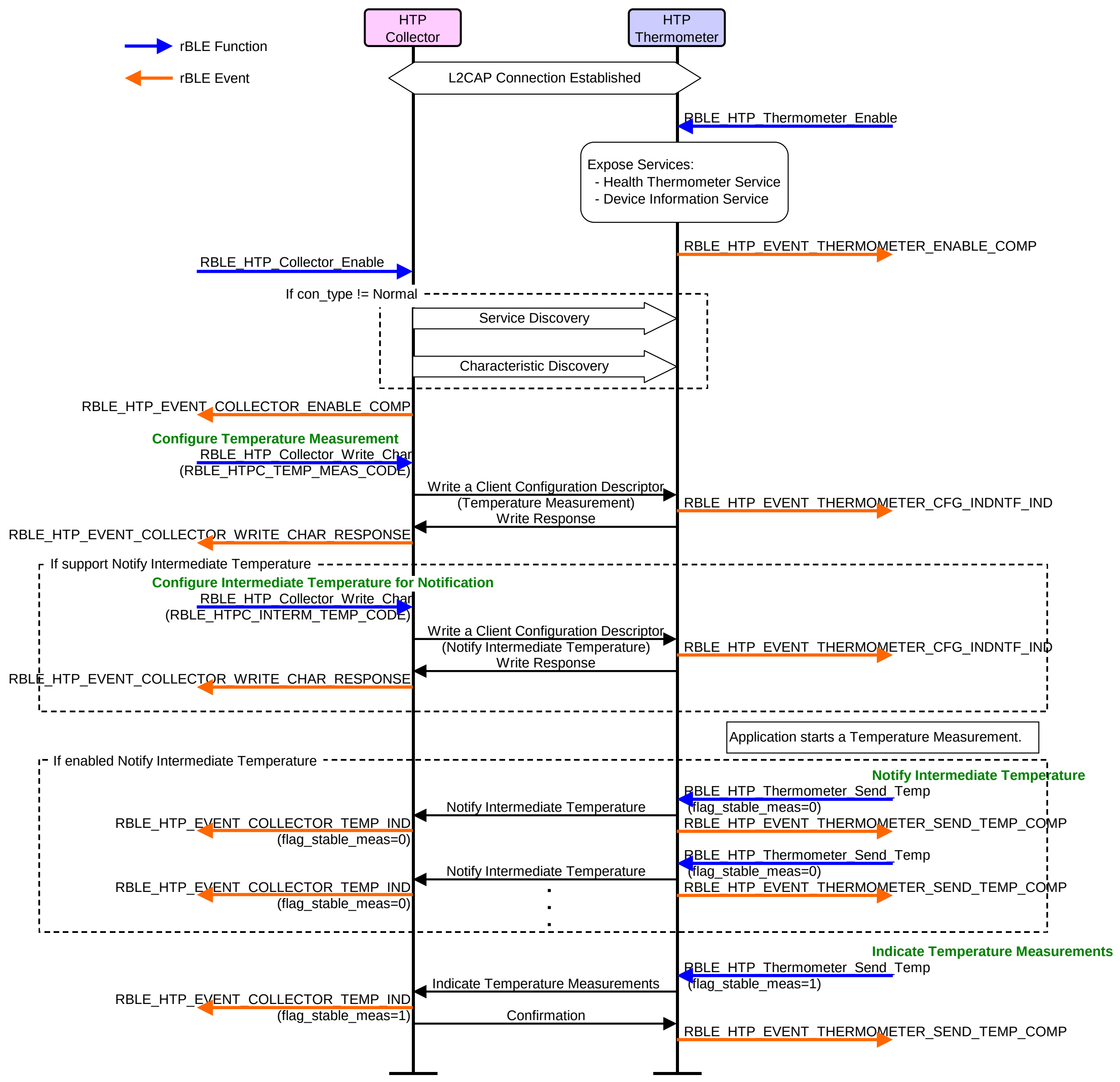
HTP Collector
HTP Thermometer
rBLE Function
rBLE Event
L2CAP Connection Established
RBLE_HTP_Thermometer_Enable
Expose Services:
- Health Thermometer Service
- Device Information Service
RBLE_HTP_EVENT_THERMOMETER_ENABLE_COMP
RBLE_HTP_Collector_Enable
If con_type != Normal
Service Discovery
Characteristic Discovery
RBLE_HTP_EVENT_COLLECTOR_ENABLE_COMP
Configure Temperature Measurement
RBLE_HTP_Collector_Write_Char
(RBLE_HTPC_TEMP_MEAS_CODE)
Write a Client Configuration Descriptor
(Temperature Measurement)
RBLE_HTP_EVENT_THERMOMETER_CFG_INDNTF_IND
Write Response
RBLE_HTP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE
If support Notify Intermediate Temperature
Configure Intermediate Temperature for Notification
RBLE_HTP_Collector_Write_Char
(RBLE_HTPC_INTERM_TEMP_CODE)
Write a Client Configuration Descriptor
(Notify Intermediate Temperature)
RBLE_HTP_EVENT_THERMOMETER_CFG_INDNTF_IND
Write Response
RBLE_HTP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE
Application starts a Temperature Measurement.
If enabled Notify Intermediate Temperature
Notify Intermediate Temperature
RBLE_HTP_Thermometer_Send_Temp
(flag_stable_meas=0)
Notify Intermediate Temperature
RBLE_HTP_EVENT_COLLECTOR_TEMP_IND
(flag_stable_meas=0)
RBLE_HTP_EVENT_THERMOMETER_SEND_TEMP_COMP
RBLE_HTP_Thermometer_Send_Temp
(flag_stable_meas=0)
Notify Intermediate Temperature
RBLE_HTP_EVENT_COLLECTOR_TEMP_IND
(flag_stable_meas=0)
RBLE_HTP_EVENT_THERMOMETER_SEND_TEMP_COMP
Indicate Temperature Measurements
RBLE_HTP_Thermometer_Send_Temp
(flag_stable_meas=1)
Indicate Temperature Measurements
RBLE_HTP_EVENT_COLLECTOR_TEMP_IND
(flag_stable_meas=1)
Confirmation
RBLE_HTP_EVENT_THERMOMETER_SEND_TEMP_COMP

Figure 3-1 Example of Use Case In Which HTP Is Implemented 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. 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: HTP

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	---	Bookmark is added.
1.03	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.
1.04	Apr 17, 2015	2	The service definitions are updated.

Bluetooth Low Energy Protocol Stack
API Reference Manual: HTP

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

Bluetooth Low Energy Protocol Stack

Renesas Electronics Corporation

R01UW0091EJ0104