User's Manual

Bluetooth® Low Energy Protocol Stack

API Reference Manual: LNP

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 Alert Notification profile (LNP) 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	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	This manual
Application Note: Sample Program	R01AN1375E
Application Note: rBLE Command Specification	R01AN1376E

List of Abbreviations and Acronyms

Abbreviation	Full Form	Remark
LNP	Alert Notification Profile	
LNS	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

Bluetooth Low Energy Protocol Stack
API Reference Manual: LNP

R01UW0113EJ0102
Rev.1.02
Apr 17, 2015

1. Overview

This manual describes the API (Application Program Interface) of the Location and Navigation profile (LNP) 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. Location and Navigation Profile

This section describes the API of the Location and Navigation profile. The Location and Navigation profile is used to enable a data collection device to information such as speed, location, and/or navigation data from a sensor device.

3.1 Definitions

This section describes the definitions used by the API of the Location and Navigation profile.

- Declaration of Max text size

```
#define RBLE_LNP_SENSOR_ROUTE_NAME_MAX    17    Max size of the route name string
```

- Declaration of enumerated type for LNP event types

```
enum RBLE_LNP_EVENT_TYPE_enum {
    RBLE_LNP_EVENT_SENSOR_ENABLE_COMP = 0x01,    Sensor enable completion event
                                                 (Parameter: sensor_enable)
    RBLE_LNP_EVENT_SENSOR_DISABLE_COMP,          Sensor disable completion event
                                                 (Parameter: sensor_disable)
    RBLE_LNP_EVENT_SENSOR_ERROR_IND,             Sensor error indication event
                                                 (Parameter: error_ind)
    RBLE_LNP_EVENT_SENSOR_SEND_LOCATION_SPEED_COMP,  Location and Speed send completion
                                                     event
                                                 (Parameter: send_location_speed)
    RBLE_LNP_EVENT_SENSOR_SET_POSITON_QUALITY_COMP, Position Quality set completion event
                                                 (Parameter: set_position_quality)
    RBLE_LNP_EVENT_SENSOR_SEND_LN_CP_COMP,       LN control point send completion event
                                                 (Parameter: send_ln_cp)
    RBLE_LNP_EVENT_SENSOR_SEND_NAVIGATION_COMP,  Navigation data send completion event
                                                 (Parameter: send_navigation)
    RBLE_LNP_EVENT_SENSOR_SEND_BATTERY_LEVEL_COMP,   Battery level send completion event
                                                 (Parameter: send_battery_level)
    RBLE_LNP_EVENT_SENSOR_CHG_LN_CP_IND,         LN control point change indication event
                                                 (Parameter: chg_ln_cp_ind)
    RBLE_LNP_EVENT_SENSOR_CFG_INDNTF_IND,        Characteristic configuration change
                                                 indication event
                                                 (Parameter: cfg_indntf_ind)
    RBLE_LNP_EVENT_SENSOR_COMMAND_DISALLOWED_IND, Command disallowed indication event
                                                 (Parameter: cmd_disallowed_ind)
    RBLE_LNP_EVENT_COLLECTOR_ENABLE_COMP = 0x81, Collector enable completion event
                                                 (Parameter: collector_enable)
    RBLE_LNP_EVENT_COLLECTOR_DISABLE_COMP,       Collector disable completion event
                                                 (Parameter: collector_disable)
    RBLE_LNP_EVENT_COLLECTOR_ERROR_IND,          Collector error indication event
                                                 (Parameter: error_ind)
    RBLE_LNP_EVENT_COLLECTOR_LOCATION_SPEED_NTF, Location and Speed notification event
                                                 (Parameter: location_speed_ntf)
    RBLE_LNP_EVENT_COLLECTOR_LN_CP_IND,          LN control point indication event
                                                 (Parameter: ln_cp_ind)
    RBLE_LNP_EVENT_COLLECTOR_NAVIGATION_NTF,     Navigation data notification event
                                                 (Parameter: navigation_ntf)
    RBLE_LNP_EVENT_COLLECTOR_BATTERY_LEVEL_NTF,  Battery level notification event
                                                 (Parameter: battery_level_ntf)
    RBLE_LNP_EVENT_COLLECTOR_READ_CHAR_RESPONSE,     Characteristic value read request
                                                     response event
                                                     (Parameter: rd_char_resp)
    RBLE_LNP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE,    Characteristic value write request
                                                     response event
                                                     (Parameter: wr_char_resp)
    RBLE_LNP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND  Command disallowed indication event
                                                     (Parameter: cmd_disallowed_ind)
};
```

- Declaration of data type for LNP event types

```
typedef uint8_t RBLE_LNP_EVENT_TYPE;
```

- Declaration of data type for LNP Sensor event callback function

```
typedef void ( *RBLE_LNPS_EVENT_HANDLER )( RBLE_LNPS_EVENT *event );
```

- Declaration of data type for LNP Collector event callback function

```
typedef void ( *RBLE_LNPC_EVENT_HANDLER )( RBLE_LNPC_EVENT *event );
```

- Declaration of enumerated type for reading LNS / DIS / BAS characteristic codes

```
enum RBLE_LNPC_RD_CHAR_CODE_enum {
    RBLE_LNPC_RD_LNS_LN_FEATURE      = 0x00,     LN feature
    RBLE_LNPC_RD_LNS_LOCATION_SPEED_CFG,         Location and Speed notification
    RBLE_LNPC_RD_LNS_POSITION_QUALITY,           Position Quality
    RBLE_LNPC_RD_LNS_LN_CP_CFG,                  LN control point indication
    RBLE_LNPC_RD_LNS_NAVIGATION_CFG,             Navigation notification
    RBLE_LNPC_RD_DIS_MANUF,                      Sensor manufacturer name
    RBLE_LNPC_RD_DIS_MODEL,                      Sensor model number
    RBLE_LNPC_RD_DIS_SERNB,                      Sensor serial number
    RBLE_LNPC_RD_DIS_HWREV,                      Sensor hardware revision
    RBLE_LNPC_RD_DIS_FWREV,                      Sensor firmware revision
    RBLE_LNPC_RD_DIS_SWREV,                      Sensor software revision
    RBLE_LNPC_RD_DIS_SYSID,                      Sensor system ID
    RBLE_LNPC_RD_DIS_IEEE,                       Sensor IEEE certification information
    RBLE_LNPC_RD_BAS_BL,                         Sensor battery level information
    RBLE_LNPC_RD_BAS_BL_CFG                      Sensor battery level notification
};
```

- Declaration of enumerated type for setting LNS / BAS characteristic codes

```
enum RBLE_LNPS_WR_CHAR_CODE_enum {
    RBLE_LNP_LOCATION_SPEED_CODE = 0x00,        Location and Speed notification setting
    RBLE_LNP_LN_CONTROL_POINT_CODE,             LN control point indication setting
    RBLE_LNP_NAVIGATION_CODE,                   Navigation characteristic notification
                                                setting
    RBLE_LNP_BATTERY_LEVEL_CODE                 Battery level characteristic
                                                notification setting
};
```

- Declaration of enumerated type for LN control point characteristic operation code setting

```
enum RBLE_LNP_LNCP_OP_CODE_enum {
    RBLE_LNP_OP_SET_CUMULATIVE_CODE = 0x01,     Set Cumulative Value
    RBLE_LNP_OP_MASK_LS_CONTENTS_CODE,          Mask Location and Speed Characteristic
                                                Content
    RBLE_LNP_OP_NAVIGATION_CONTROL_CODE,        Navigation Control
    RBLE_LNP_OP_REQ_NUM_OF_ROUTE_CODE,          Request Number of Routes
    RBLE_LNP_OP_REQ_NAME_OF_ROUTE_CODE,         Request Name of Route
    RBLE_LNP_OP_SELECT_ROUTE_CODE,              Select Route
    RBLE_LNP_OP_SET_FIX_RATE_CODE,              Set Fix Rate
    RBLE_LNP_OP_SET_ELEVATION_CODE,             Set Elevation
    RBLE_LNP_OP_RESPONSE_CODE = 0x20            Response code
};
```

- Declaration of enumerated type for LN control point characteristic response setting

```
enum RBLE_LNP_LNCP_RES_CODE_enum {
    RBLE_LNP_RES_SUCCESS_CODE = 0x01,           Success
    RBLE_LNP_RES_NOT_SUPPORTED_CODE,            Op Code not supported
    RBLE_LNP_RES_INVALID_PARAM_CODE,            Invalid parameter
    RBLE_LNP_RES_OP_FAILED_CODE                 Operation failed
};
```

- Declaration of enumerated type for Navigation control

```
enum RBLE_LNP_CNTL_NAVIGATION_enum {
    RBLE_LNP_CNTL_NAVI_STOP = 0x00,             Stop Navigation
    RBLE_LNP_CNTL_NAVI_START,                   Start Navigation to the first waypoint
    RBLE_LNP_CNTL_NAVI_PAUSE,                   Pause Navigation keeping the next
                                                waypoint
    RBLE_LNP_CNTL_NAVI_RESUME,                  Continue Navigation from the point where
                                                navigation was paused
    RBLE_LNP_CNTL_NAVI_SKIP,                    Skip waypoint
    RBLE_LNP_CNTL_NAVI_SET_NEAREST              Select nearest waypoint on a route
};
```

- 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;
```

- Location and Navigation Sensor characteristic information structures

```
typedef struct RBLE_LNP_SENSOR_PARAM_t {
    uint16_t       location_speed_ntf_en;        Location and Speed notification
                                                 configuration value
    uint16_t       ln_cp_ind_en;                 LN Control point indication configuration
                                                 value
    uint16_t       navigation_ntf_en;            Navigation data notification
                                                 configuration value
    uint16_t       battery_level_ntf_en;         Battery level notification configuration
                                                 value
} RBLE_LNP_SENSOR_PARAM;
```

- Location and Speed information structures

```
typedef struct RBLE_LNP_LOCATION_SPEED_INFO_t {
    uint16_t       flags;                         Data field flags
    uint16_t       instant_speed;                 Instantaneous Speed (unit: 1/100[m/s])
    uint32_t       total_distance;                Total Distance (unit: 1/10[m])
    int32_t        latitude;                      Latitude (unit: 1/10^7[degree])
    int32_t        longitude;                     Longitude (unit: 1/10^7[degree])
    int32_t        elevation;                     Elevation (unit: 1/100[m])
    uint16_t       heading;                       Heading (unit: 1/100[degree])
    uint8_t        rolling_time;                  Rolling Time (unit: [second])
    uint8_t        reserved;                      Reserved
    RBLE_DATE_TIME utc_time;                      UTC time at the sensor when the position
                                                  was received
} RBLE_LNP_LOCATION_SPEED_INFO;
```

- Position Quality information structures

```
typedef struct RBLE_LNP_POSITION_QUALITY_INFO_t {
    uint16_t       flags;                         Data field flags
    uint8_t        beacon_solution_num;           Number of beacons used to calculate the
                                                  current position
    uint8_t        beacon_view_num;               Number of beacons from which the Server is
                                                  receiving data
    uint16_t       first_fix_time;                The time used to receive data and
                                                  calculate the initial position
                                                  (unit: 1/10[second])
    uint32_t      ehpe;                         Estimated Horizontal Position Error
                                                (unit: 1/100[m])
    uint32_t      evpe;                         Estimated Vertical Position Error
                                                (unit: 1/100[m])
    uint8_t       hdop;                         Horizontal Dilution of Precision
    uint8_t       vdop;                         Vertical Dilution of Precision
} RBLE_LNP_POSITION_QUALITY_INFO;
```

- LN Control Point structure

```
typedef struct RBLE_LNP_WR_CONTROL_POINT_INFO_t {
    uint8_t       OpCode;                       Op Code
    uint8_t       reserved;                     Reserved
    uint32_t      cumulative_value;             Total Distance
    uint16_t      content_mask;                 Mask Location and Speed Characteristic
    uint16_t      route_num;                    Number of Routes
    uint8_t       control_val;                  Control Code
    uint8_t       fix_rate;                     Fix Rate (unit: 1[second])
    int32_t       elevation;                    Elevation (unit: 1/100[m])
} RBLE_LNP_WR_CONTROL_POINT_INFO;
```

- LN Control Point indication structure

```
typedef struct RBLE_LNP_IND_CONTROL_POINT_INFO_t {
    uint8_t       OpCode;                       Op Code
    uint8_t       request_op_code;              Request Op Code
    uint8_t       response_val;                 Response value
    uint8_t       reserved;                     Reserved
    uint16_t      route_num;                    Number of Routes
    uint8_t       name_size;                    Length of route name string
    uint8_t       route_name[RBLE_LNP_SENSORE_ROUTE_NAME_MAX];
                                                Route name
} RBLE_LNP_IND_CONTROL_POINT_INFO;
```

- Navigation information structures

```
typedef struct RBLE_LNP_NAVIGATION_INFO_t {
    uint16_t      flags;                        Data field flags
    uint16_t      bearing;                      Direction where the user should be heading
                                                to reach the next waypoint or final
                                                destination (unit: 1/100[degree])
    uint16_t      heading;                      Direction where the user is heading to
                                                (unit: 1/100[degree])
    uint32_t      remain_dis;                   Remaining Distance (unit: 1/10[m])
    int32_t       remain_v_dis;                 Remaining Vertical Distance
                                                (unit: 1/100[m])
    RBLE_DATE_TIME estimate_time;               Estimated Time of Arrival
} RBLE_LNP_NAVIGATION_INFO;
```

- Location and Navigation service content structures

```
typedef struct RBLE_LNS_CONTENT_t{
    uint16_t        shdl;                        Location and Navigation service start handle
    uint16_t        ehdl;                        Location and Navigation service end handle
    uint16_t        ln_feature_char_hdl;             LN Feature characteristic handle
    uint16_t        ln_feature_val_hdl;              LN Feature characteristic value
                                                     handle
    uint8_t         ln_feature_prop;                 LN Feature characteristic property
    uint8_t         reserved1;                       Reserved
    uint16_t        location_speed_char_hdl;         Location and Speed characteristic
                                                     handle
    uint16_t        location_speed_val_hdl;          Location and Speed characteristic
                                                     value handle
    uint16_t        location_speed_cfg_hdl;          Location and Speed client
                                                     characteristic configuration
                                                     descriptor handle
    uint8_t         location_speed_prop;             Location and Speed characteristic
                                                     property
    uint8_t         reserved2;                       Reserved
    uint16_t        position_quality_char_hdl;       Position Quality characteristic
                                                     handle
    uint16_t        position_quality_val_hdl;        Position Quality characteristic
                                                     value handle
    uint8_t         position_quality_prop;           Position Quality characteristic
                                                     property
    uint8_t         reserved3;                       Reserved
    uint16_t        ln_cp_char_hdl;                  LN control point characteristic
                                                     handle
    uint16_t        ln_cp_val_hdl;                   LN control point characteristic
                                                     value handle
    uint16_t        ln_cp_cfg_hdl;                   LN control point client
                                                     characteristic    configuration
                                                     descriptor handle
    uint8_t         ln_cp_prop;                      LN control point characteristic
                                                     property
    uint8_t         reserved4;                       Reserved
    uint16_t        navigation_char_hdl;             Navigation characteristic handle
    uint16_t        navigation_val_hdl;              Navigation characteristic value
                                                     handle
    uint16_t        navigation_cfg_hdl;              Navigation client characteristic
                                                     configuration descriptor handle
    uint8_t         navigation_prop;                 Navigation characteristic
                                                     property
    uint8_t         reserved5;                       Reserved
} RBLE_LNS_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;
```

- Battery service content structures

```
typedef struct RBLE_BATS_CONTENT_t{
    uint16_t        shdl;                         Battery service start handle
    uint16_t        ehdl;                         Battery service end handle
    uint16_t        battery_lvl_char_hdl;         Battery level characteristic handle
    uint16_t        battery_lvl_val_hdl;          Battery level characteristic value
                                                  handle
    uint16_t        battery_lvl_cfg_hdl;          Battery level characteristic
                                                  configuration descriptor handle
    uint8_t         battery_lvl_prop;             Battery level property
    uint8_t         reserved;                     Reserved
}RBLE_BATS_CONTENT;
```

- LNP Sensor event parameter structures

```
typedef struct RBLE_LNPS_EVENT_t {
    RBLE_LNP_EVENT_TYPE             type;                 LNP event type
    uint8_t                         reserved;             Reserved
    union Event_Lns_Parameter_u {
        Generic event
        RBLE_STATUS                 status;               Status

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

        Sensor disable completion event
        struct RBLE_LNP_Sensor_Disable_t{
            uint16_t          conhdl;                     Connection handle
            RBLE_LNP_SENSOR_PARAM param;                  Location and Navigation
                                                          service information
        }server_disable;

        Sensor error indication event
        struct RBLE_LNP_Sensor_Error_Ind_t{
            uint16_t                conhdl;               Connection handle
            RBLE_STATUS             status;               Status
        }error_ind;
```

Sensor Location and Speed send completion event

```
struct RBLE_LNP_Sensor_Send_Location_Speed_t{
    uint16_t                conhdl;            Connection handle
    RBLE_STATUS             status;            Status
}send_location_speed;
```

Sensor Position Quality set completion event

```
struct RBLE_LNP_Sensor_Set_Position_Quality_t{
    RBLE_STATUS             status;            Status
}set_position_quality;
```

Sensor LN control point send completion event

```
struct RBLE_LNP_Sensor_Send_LN_Control_Point_t{
    uint16_t                conhdl;            Connection handle
    RBLE_STATUS             status;            Status
}send_ln_cp;
```

Sensor Navigation data send completion event

```
struct RBLE_LNP_Sensor_Send_Navigation_t{
    uint16_t                conhdl;            Connection handle
    RBLE_STATUS             status;            Status
}send_navigation;
```

Sensor battery level send completion event

```
struct RBLE_LNP_Sensor_Send_Battery_Level_t{
    uint16_t                conhdl;            Connection handle
    RBLE_STATUS             status;            Status
}send_battery_level;
```

Sensor LN control point change indication event

```
struct RBLE_LNP_Sensor_Chg_Ln_Cp_Ind_t{
    uint16_t                      conhdl;      Connection handle
    RBLE_LNP_WR_CONTROL_POINT_INFO   cp_info;  LN control point information
}chg_ln_cp_ind;
```

Sensor configuration characteristic value indication event

```
struct RBLE_LNP_Sensor_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
}cfg_indntf_ind;
```

Sensor command disallowed indication event

```
struct RBLE_LNP_Sensor_Command_Disallowed_Ind_t{
```

```
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                opcode;             Opcode
        }cmd_disallowed_ind;
    } param;
} RBLE_LNPS_EVENT;
```

- LNP Collector event parameter structures

```
typedef struct RBLE_LNPC_EVENT_t {
    RBLE_LNP_EVENT_TYPE             type;               LNP event type
    uint8_t                         reserved;           Reserved
    union Event_Lnc_Parameter_u {
        Generic event
        RBLE_STATUS                 status;             Status

        Collector enable completion event
        struct RBLE_LNP_Collector_Enable_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                conhdl;             Connection handle
            RBLE_LNS_CONTENT        lns;                Location and Navigation service
                                                        content
            RBLE_DIS_CONTENT        dis;                Device information service content
            RBLE_BATS_CONTENT       bas;                Battery service content
        }collector_enable;

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

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

        Collector Location and Speed notification event
        struct RBLE_LNP_Collector_Location_Speed_Ntf_t{
            uint16_t                conhdl;             Connection handle
            RBLE_LNP_LOCATION_SPEED_INFO location_speed_info;
                                                        Location and Speed information
        }location_speed_ntf;
        Collector LN control point indication event
        struct RBLE_LNP_Collector_LN_CP_Ind_t{
            uint16_t                conhdl;             Connection handle
            RBLE_LNP_IND_CONTROL_POINT_INFO ind_cp_info;
                                                        LN control point information
        }ln_cp_ind;

        Collector Navigation data notification event
        struct RBLE_LNP_Collector_Navigation_Ntf_t{
            uint16_t                 conhdl;            Connection handle
            RBLE_LNP_NAVIGATION_INFO navigation_info;   Navigation information
        }navigation_ntf;

        Collector battery level notification event
        struct RBLE_LNP_Collector_Battery_Level_Ntf_t{
            uint16_t                conhdl;             Connection handle
            uint8_t                 battery_level;      Battery level
        }battery_level_ntf;

        Collector characteristic value read request response event
        struct RBLE_LNP_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_LNP_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_LNP_Collector_Command_Disallowed_Ind_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                opcode;             Opcode
        }cmd_disallowed_ind;
    } param;
} RBLE_LNPC_EVENT;
```

3.2 Functions

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

Table 3-1 API Functions Used by the LNP

RBLE_LNP_Sensor_Enable	Enables the Sensor role.
RBLE_LNP_Sensor_Disable	Disables the Sensor role.
RBLE_LNP_Sensor_Send_Location_Speed	Sends the Location and Speed information.
RBLE_LNP_Sensor_Set_Position_Quality	Sets the Position Quality information.
RBLE_LNP_Sensor_Send_LN_Control_Point	Sends the LN control point information.
RBLE_LNP_Sensor_Send_Navigation	Sends the Navigation data.
RBLE_LNP_Sensor_Send_Battery_Level	Sends the battery level.
RBLE_LNP_Collector_Enable	Enables the Collector role.
RBLE_LNP_Collector_Disable	Disables the Collector role.
RBLE_LNP_Collector_Read_Char	Reads the characteristic value.
RBLE_LNP_Collector_Write_CP_Control_Point	Sets the LN control point.
RBLE_LNP_Collector_Write_Char	Writes the characteristic value.

3.2.1 RBLE_LNP_Sensor_Enable

RBLE_STATUS RBLE_LNP_Sensor_Enable(uint16_t conhdl, uint8_t sec_lvl, uint8_t con_type,
    RBLE_LNP_SENSOR_PARAM *param, RBLE_LNPS_EVENT_HANDLER call_back)

This function enables the LNP Sensor role.
If the notification settings of the Location/Speed, Navigation, Battery level and indication settings of the LN control point are configured from the Collector, set the notification/indication 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/indication setting parameter.
The result is reported by using the Sensor role enable completion event RBLE_LNP_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*	*location_speed_ntf_en*	RBLE_PRF_STOP_NTFIND	Stop notification of Location and Speed information.
		RBLE_PRF_START_NTF	Start notification of Location and Speed information.
	ln_cp_ind_en	RBLE_PRF_STOP_NTFIND	Stop indication of LN control point.
		RBLE_PRF_START_IND	Start indication of LN control point.
	navigation_ntf_en	RBLE_PRF_STOP_NTFIND	Stop notification of Navigation information.
		RBLE_PRF_START_NTF	Start notification of Navigation information.
	battery_level_ntf_en	RBLE_PRF_STOP_NTFIND	Stop notification of battery level information.
		RBLE_PRF_START_NTF	Start notification of battery level information.
call_back	Specify the callback function that reports the LNP 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_LNP_Sensor_Disable

RBLE_STATUS RBLE_LNP_Sensor_Disable(uint16_t conhdl)		
This function disables the LNP Sensor role. The result is reported by using the Sensor role disable completion event RBLE_LNP_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_LNP_Sensor_Send_Location_Speed

RBLE_STATUS RBLE_LNP_Sensor_Send_Location_Speed (uint16_t conhdl, RBLE_LNP_LOCATION_SPEED_INFO * location_speed_info)				
This function sends the Location and Speed information from the sensor. The result is reported by using the Sensor role Location and Speed send completion event RBLE_LNP_EVENT_SENSOR_SEND_LOCATION_SPEED_COMP.				
Parameters:				
	conhdl	Connection handle		
	**location_speed_info*	*flags*	Flag that defines whether there is a data field in the characteristic value or not	
		instant_speed	Instantaneous Speed (unit: 1/100[m/s])	
		total_distance	Total Distance (unit: 1/10[m], range: 0 to 1677721.5m)	
		latitude	Latitude (unit: 1/10^7[degree], WGS-84 format)	
		longitude	Longitude (unit: 1/10^7[degree], WGS-84 format)	
		elevation	Elevation (unit: 1/100[m], range: -83886.08m to 83886.07m)	
		heading	Heading (unit: 1/100[degree], WGS-84 format)	
		rolling_time	Rolling Time (unit: [second])	
		utc_time	UTC time at the sensor when the position was received	
			year	Year
			month	Month
			day	Day
			hour	Hour
			min	Minute
			sec	Second
Return:				
	RBLE_OK	Success		
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.		

3.2.4 RBLE_LNP_Sensor_Set_Position_Quality

RBLE_STATUS RBLE_LNP_Sensor_Set_Position_Quality (uint16_t conhdl,
RBLE_LNP_POSITION_QUALITY_INFO *quality_info)

This function sets the Position Quality information of the sensor.
The result is reported by using the Sensor role Position Quality set completion event RBLE_LNP_EVENT_SENSOR_SET_POSITION_QUALITY_COMP.

Parameters:

conhdl	Connection handle	
**quality_info*	*flags*	Flag that defines whether there is a data field in the characteristic value or not
	beacon_solution_num	Number of beacons used to calculate the current position
	beacon_view_num	Number of beacons from which the Server is receiving data
	first_fix_time	The time used to receive data and calculate the initial position (unit: 1/10[second])
	ehpe	Estimated Horizontal Position Error (unit: 1/100[m])
	evpe	Estimated Vertical Position Error (unit: 1/100[m])
	hdop	Horizontal Dilution of Precision
	vdop	Vertical Dilution of Precision

Return:

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

3.2.5 RBLE_LNP_Sensor_Send_LN_Control_Point

RBLE_STATUS RBLE_LNP_Sensor_Send_LN_Control_Point (uint16_t conhdl,
RBLE_LNP_IND_CONTROL_POINT_INFO *ind_cp_info)

This function sends the LN control point information from the sensor.
If the operation is written to the LN control point from Collector, set RBLE_LNP_OP_RESPONSE_CODE to OpCode.
Set the operation code that is sent from the Collector to the request_op_code, set the status for the operation to the response_value.
Depending on the operation that is requested by the Collector, set the additional parameters required.
The result is reported by using the Sensor role LN control point send completion event RBLE_LNP_EVENT_SENSOR_SEND_LN_CP_COMP.

Parameters:

conhdl	Connection handle		
**ind_cp_info*	*OpCode*	RBLE_LNP_OP_RESPONSE_CODE	Response code
	request_op_code	RBLE_LNP_OP_SET_CUMULATIVE_CODE	Set cumulative value
		RBLE_LNP_OP_MASK_LS_CONTENTS_CODE	Mask Location and Speed Characteristic Content
		RBLE_LNP_OP_NAVIGATION_CONTROL_CODE	Navigation control
		RBLE_LNP_OP_REQ_NUM_OF_ROUTE_CODE	Request number of routes (set the *route_num*)
		RBLE_LNP_OP_REQ_NAME_OF_ROUTE_CODE	Request name of route (set the *name_size*, and *route_name*)
		RBLE_LNP_OP_SELECT_ROUTE_CODE	Select route
		RBLE_LNP_OP_SET_FIX_RATE_CODE	Set Fix Rate
		RBLE_LNP_OP_SET_ELEVATION_CODE	Set Elevation
	reqponse_value	RBLE_LNP_RES_SUCCESS_CODE	*Success*
		RBLE_LNP_RES_NOT_SUPPORTED_CODE	*Op Code not supported*
		RBLE_LNP_RES_INVALID_PARAM_CODE	*Invalid parameter*
		RBLE_LNP_RES_OP_FAILED_CODE	*Operation failed*
	route_num	Number of routes	
	name_size	Length of route name string	
	route_name[RBLE_LNP_SENSORE_ROUTE_NAME_MAX]	Root name of the specified route (UTF-8 format)	

Return:

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

3.2.6 RBLE_LNP_Sensor_Send_Navigation

<table>
<tr><td colspan="5">RBLE_STATUS RBLE_LNP_Sensor_Send_Navigation (uint16_t conhdl,
RBLE_LNP_NAVIGATION_INFO * navigation_info)</td></tr>
<tr><td colspan="5">This function sends the Navigation information from the sensor.
The result is reported by using the Sensor role Navigation data send completion event RBLE_LNP_EVENT_SENSOR_SEND_NAVIGATION_COMP.</td></tr>
<tr><td colspan="5">Parameters:</td></tr>
<tr><td rowspan="13"></td><td>*conhdl*</td><td colspan="3">Connection handle</td></tr>
<tr><td rowspan="12">*navigation_info</td><td>*flags*</td><td colspan="2">Flag that defines whether there is a data field in the characteristic value or not</td></tr>
<tr><td>*bearing*</td><td colspan="2">Direction where the user should be heading to reach the next waypoint or final destination (unit: 1/100[degree], WSG-84 format)</td></tr>
<tr><td>*heading*</td><td colspan="2">Direction where the user is heading to (unit: 1/100[degree], WSG-84 format)</td></tr>
<tr><td>*remain_dis*</td><td colspan="2">Remaining distance (unit: 1/10[m], range: 0 to 1677721.5m)</td></tr>
<tr><td>*remain_v_dis*</td><td colspan="2">Remaining vertical distance (unit: 1/100[m], range: -83886.08m to 83886.07m)</td></tr>
<tr><td rowspan="7">*estimate_time*</td><td colspan="2">Estimated time of arrival</td></tr>
<tr><td>*year*</td><td>Year</td></tr>
<tr><td>*month*</td><td>Month</td></tr>
<tr><td>*day*</td><td>Day</td></tr>
<tr><td>*hour*</td><td>Hour</td></tr>
<tr><td>*min*</td><td>Minute</td></tr>
<tr><td>*sec*</td><td>Second</td></tr>
<tr><td colspan="5">Return:</td></tr>
<tr><td rowspan="2"></td><td colspan="2">*RBLE_OK*</td><td colspan="2">Success</td></tr>
<tr><td colspan="2">*RBLE_STATUS_ERROR*</td><td colspan="2">Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.</td></tr>
</table>

3.2.7 RBLE_LNP_Sensor_Send_Battery_Level

<table>
<tr><td colspan="3">RBLE_STATUS RBLE_LNP_Sensor_Send_Battery_Level(uint16_t conhdl, uint8_t battery_level)</td></tr>
<tr><td colspan="3">This function updates the battery level of the sensor and then sends the battery level to collector if the battery level characteristic is enabled.
The result is reported by using the Sensor role battery level send completion event RBLE_LNP_EVENT_SENSOR_SEND_BATTERY_LEVEL_COMP.</td></tr>
<tr><td colspan="3">Parameters:</td></tr>
<tr><td rowspan="2"></td><td>*conhdl*</td><td>Connection handle</td></tr>
<tr><td>*battery_level*</td><td>Battery level</td></tr>
<tr><td colspan="3">Return:</td></tr>
<tr><td rowspan="2"></td><td>*RBLE_OK*</td><td>Success</td></tr>
<tr><td>*RBLE_STATUS_ERROR*</td><td>Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.</td></tr>
</table>

3.2.8 RBLE_LNP_Collector_Enable

RBLE_STATUS RBLE_LNP_Collector_Enable(uint16_t conhdl, uint8_t con_type,
    RBLE_LNS_CONTENT *lns, RBLE_DIS_CONTENT *dis, RBLE_BATS_CONTENT *bas,
    RBLE_LNPC_EVENT_HANDLER call_back)

This function enables the LNP Collector role and starts access to the service exposed by the LNP Sensor. The result is reported by using the Collector role enable completion event RBLE_LNP_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
**lns*	*shdl*	Location and Navigation service start handle
	ehdl	Location and Navigation service end handle
	ln_feature_char_hdl	LN Feature characteristic handle
	ln_feature_val_hdl	LN Feature characteristic value handle
	ln_feature_prop	LN Feature characteristic property
	location_speed_char_hdl	Location and Speed characteristic handle
	location_speed_val_hdl	Location and Speed characteristic value handle
	location_speed_cfg_hdl	Location and Speed client characteristic configuration descriptor handle
	location_speed_prop	Location and Speed characteristic property
	position_quality_char_hdl	Position Quality characteristic handle
	position_quality_val_hdl	Position Quality characteristic value handle
	position_quality_prop	Position Quality characteristic property
	ln_cp_char_hdl	LN control point characteristic handle
	ln_cp_val_hdl	LN control point characteristic value handle
	ln_cp_cfg_hdl	LN control point client characteristic configuration descriptor handle
	ln_cp_prop	LN control point characteristic property
	navigation_char_hdl	Navigation characteristic handle
	navigation_val_hdl	Navigation characteristic value handle
	navigation_cfg_hdl	Navigation client characteristic configuration descriptor handle
	navigation_prop	Navigation 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
	**bas*	*shdl*	Battery service start handle
		ehdl	Battery service end handle
		battery_lvl_char_hdl	Battery Level characteristic handle
		battery_lvl_val_hdl	Battery Level characteristic value handle
		battery_lvl_cfg_hdl	Battery Level characteristic configuration descriptor handle
		battery_lvl_prop	Battery Level 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.	

3.2.9 RBLE_LNP_Collector_Disable

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

3.2.10 RBLE_LNP_Collector_Read_Char

RBLE_STATUS RBLE_LNP_Collector_Read_Char (uint16_t conhdl, uint8_t char_code)			
This function reads the characteristic value of the Location and Navigation service, the device information service and the battery service. The result is reported by using the characteristic value read request response event RBLE_LNP_EVENT_COLLECTOR_READ_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	char_code	RBLE_LNPC_RD_LNS_LN_FEATURE	LN feature
		RBLE_LNPC_RD_LNS_LOCATION_SPEED_CFG	Location and Speed notification
		RBLE_LNPC_RD_LNS_POSITION_QUALITY	Position Quality
		RBLE_LNPC_RD_LNS_LN_CP_CFG	LN control point indication
		RBLE_LNPC_RD_LNS_NAVIGATION_CFG	Navigation notification
		RBLE_LNPC_RD_DIS_MANUF	Sensor manufacturer name
		RBLE_LNPC_RD_DIS_MODEL	Sensor model number
		RBLE_LNPC_RD_DIS_SERNB	Sensor serial number
		RBLE_LNPC_RD_DIS_HWREV	Sensor hardware revision
		RBLE_LNPC_RD_DIS_FWREV	Sensor firmware revision
		RBLE_LNPC_RD_DIS_SWREV	Sensor software revision
		RBLE_LNPC_RD_DIS_SYSID	Sensor system ID
		RBLE_LNPC_RD_DIS_IEEE	Sensor IEEE certification information
		RBLE_LNPC_RD_BAS_BL	Battery level
		RBLE_LNPC_RD_BAS_BL_CFG	Battery level notification
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.11 RBLE_LNP_Collector_Write_LN_Control_Point

RBLE_STATUS RBLE_LNP_Collector_Write_LN_Control_Point(uint16_t conhdl,
RBLE_LNP_WR_CONTROL_POINT_INFO * wr_cp_info)

This function sets the LN control point characteristic information of the Location and Navigation service.
If OpCode requires parameters, set the additional parameters to be set according to the operation.

All the notifications are enabled by default mask configuration of Location and Speed (RBLE_LNP_OP_MASK_LS_CONTENTS_CODE), and each time a connection is established, it shall use its default configuration. Therefore, each time a connection is established, turn off a particular field of the Location and Speed notification when that particular notification is not required.

In order to receive the notification of Navigation, it is required to start navigation by using Navigation Control (RBLE_LNP_OP_NAVIGATION_CONTROL_CODE).

The result is reported by using the characteristic value write request response event RBLE_LNP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE.

Parameters:

conhdl	Connection handle		
*wr_cp_info	LN control point setting value		
	OpCode	RBLE_LNP_OP_SET_CUMULATIVE_CODE	Set cumulative value (set the *cumulative_value*)
		RBLE_LNP_OP_MASK_LS_CONTENTS_CODE	Mask Location and Speed Characteristic Content (set the *content_mask*)
		RBLE_LNP_OP_NAVIGATION_CONTROL_CODE	Navigation Control (set the *control_val*)
		RBLE_LNP_OP_REQ_NUM_OF_ROUTE_CODE	Request Number of Routes
		RBLE_LNP_OP_REQ_NAME_OF_ROUTE_CODE	Request Name of Route (set the *route_num*)
		RBLE_LNP_OP_SELECT_ROUTE_CODE	Select Route (set the *route_num*)
		RBLE_LNP_OP_SET_FIX_RATE_CODE	Set Fix Rate (set the *fix_rate*)
		RBLE_LNP_OP_SET_ELEVATION_CODE	Set Elevation (set the *elevation*)
	cumulative_value	Total Distance (unit: 1/10[m], range: 0 to 1677721.5m)	
	content_mask	Mask settings of Location and Speed (0: Leave, 1: Turn off) bit[0] : Instantaneous Speed bit[1] : Total Distance bit[2] : Location bit[3] : Elevation bit[4] : Heading bit[5] : Rolling Time bit[6] : UTC Time bit[7 - 15] : Reserved	
	route_num	Number of routes	
	control_val	Navigation Control parameter	
		RBLE_LNP_CNTL_NAVI_STOP	Stop Navigation
		RBLE_LNP_CNTL_NAVI_START	Start Navigation to the first waypoint

RBLE_STATUS RBLE_LNP_Collector_Write_LN_Control_Point(uint16_t conhdl, RBLE_LNP_WR_CONTROL_POINT_INFO * wr_cp_info)				
			RBLE_LNP_CNTL_NAVI_PAUSE	Pause Navigation keeping the next waypoint
			RBLE_LNP_CNTL_NAVI_RESUME	Continue Navigation from the point where navigation was paused
			RBLE_LNP_CNTL_NAVI_SKIP	Skip waypoint
			RBLE_LNP_CNTL_NAVI_SET_NEAREST	Select nearest waypoint on a route
		fix_rate	Fix Rate (unit: 1[second])	
		elevation	Elevation (unit: 1/100[m], range: -83886.08m to 83886.07m)	
Return:				
	RBLE_OK	Success		
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.		

3.2.12 RBLE_LNP_Collector_Write_Char

RBLE_STATUS RBLE_LNP_Collector_Write_Char(uint16_t conhdl, uint8_t char_code, uint16_t cfg_val)			
This function writes each client characteristic configuration descriptor of the Location and Navigation service or the Battery service. The result is reported by using the characteristic value write request response event RBLE_LNP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	char_code	RBLE_LNP_LOCATION_SPEED_CODE	Location and Speed client characteristic configuration descriptor
		RBLE_LNP_LN_CONTROL_POINT_CODE	LN control point client characteristic configuration descriptor
		RBLE_LNP_NAVIGATION_CODE	Navigation client characteristic configuration descriptor
		RBLE_LNP_BATTERY_LEVEL_CODE	Battery level client characteristic configuration descriptor
	cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification/ 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.	

3.3 Events

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

Table 3-2 Events Defined for the LNP

RBLE_LNP_EVENT_SENSOR_ENABLE_COMP	Sensor role enable completion event
RBLE_LNP_EVENT_SENSOR_DISABLE_COMP	Sensor role disable completion event
RBLE_LNP_EVENT_SENSOR_ERROR_IND	Sensor role error indication event
RBLE_LNP_EVENT_SENSOR_SEND_LOCATION_SPEED_COMP	Location and Speed send completion event
RBLE_LNP_EVENT_SENSOR_SET_POSITON_QUALITY_COMP	Position Quality set completion event
RBLE_LNP_EVENT_SENSOR_SEND_LN_CP_COMP	LN control point information send completion event
RBLE_LNP_EVENT_SENSOR_SEND_NAVIGATION_COMP	Navigation data send completion event
RBLE_LNP_EVENT_SENSOR_SEND_BATTERY_LEVEL_COMP	Battery level information send completion event
RBLE_LNP_EVENT_SENSOR_CHG_LN_CP_IND	LN control point change indication event
RBLE_LNP_EVENT_SENSOR_CFG_INDNTF_IND	Characteristic configuration change indication event
RBLE_LNP_EVENT_SENSOR_COMMAND_DISALLOWED_IND	Sensor role command disallowed indication event
RBLE_LNP_EVENT_COLLECTOR_ENABLE_COMP	Collector role enable completion event
RBLE_LNP_EVENT_COLLECTOR_DISABLE_COMP	Collector role disable completion event
RBLE_LNP_EVENT_COLLECTOR_ERROR_IND	Collector role error indication event
RBLE_LNP_EVENT_COLLECTOR_LOCATION_SPEED_NTF	Location and Speed notification event
RBLE_LNP_EVENT_COLLECTOR_LN_CP_IND	LN control point indication event
RBLE_LNP_EVENT_COLLECTOR_NAVIGATION_NTF	Navigation data notification event
RBLE_LNP_EVENT_COLLECTOR_BATTERY_LEVEL_NTF	Battery level notification event
RBLE_LNP_EVENT_COLLECTOR_READ_CHAR_RESPONSE	Characteristic value read request response event
RBLE_LNP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE	Characteristic value write request response event
RBLE_LNP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND	Collector role command disallowed indication event

3.3.1 RBLE_LNP_EVENT_SENSOR_ENABLE_COMP

RBLE_LNP_EVENT_SENSOR_ENABLE_COMP		
This event reports the result of enabling the Sensor role (RBLE_LNP_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_LNP_EVENT_SENSOR_DISABLE_COMP

RBLE_LNP_EVENT_SENSOR_DISABLE_COMP				
This event reports the result of disabling the Sensor role (RBLE_LNP_Sensor_Disable).				
Parameters:				
	conhdl	Connection handle		
	param	*location_speed_ntf_en*	RBLE_PRF_STOP_NTFIND	Stop notification of Location and Speed information.
			RBLE_PRF_START_NTF	Start notification of Location and Speed information.
		ln_cp_ind_en	RBLE_PRF_STOP_NTFIND	Stop indication of LN control point.
			RBLE_PRF_START_IND	Start indication of LN control point.
		navigation_ntf_en	RBLE_PRF_STOP_NTFIND	Stop notification of Navigation data.
			RBLE_PRF_START_NTF	Start notification of Navigation data.
		battery_level_ntf_en	RBLE_PRF_STOP_NTFIND	Stop notification of battery level information.
			RBLE_PRF_START_NTF	Start notification of battery level information.

3.3.3 RBLE_LNP_EVENT_SENSOR_ERROR_IND

RBLE_LNP_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_LNP_EVENT_SENSOR_SEND_LOCATION_SPEED_COMP

RBLE_LNP_EVENT_SENSOR_SEND_LOCATION_SPEED_COMP		
This event reports completion of sending the Location and Speed (RBLE_LNP_Sensor_Send_Location_Speed).		
Parameters:		
	conhdl	Connection handle
	status	Location and Speed 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_LNP_EVENT_SENSOR_SET_POSITION_QUALITY_COMP

RBLE_LNP_EVENT_SENSOR_SET_POSITION_QUALITY_COMP		
This event reports completion of setting the Position Quality (RBLE_LNP_Sensor_Set_Position_Quality).		
Parameters:		
	status	Position Quality set 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_LNP_EVENT_SENSOR_SEND_LN_CP_COMP

RBLE_LNP_EVENT_SENSOR_SEND_LN_CP_COMP		
This event reports completion of sending the LN control point information (RBLE_LNP_Sensor_Send_LN_Control_Point).		
Parameters:		
	conhdl	Connection handle
	status	LN control point information 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*.)

3.3.7 RBLE_LNP_EVENT_SENSOR_SEND_NAVIGATION_COMP

RBLE_LNP_EVENT_SENSOR_SEND_NAVIGATION_COMP		
This event reports completion of sending the Navigation data (RBLE_LNP_Sensor_Send_Navigation).		
Parameters:		
	conhdl	Connection handle
	status	Navigation data 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*.)

3.3.8 RBLE_LNP_EVENT_SENSOR_SEND_BATTERY_LEVEL_COMP

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

3.3.9 RBLE_LNP_EVENT_SENSOR_CHG_LN_CP_IND

RBLE_LNP_EVENT_SENSOR_CHG_LN_CP_IND				
This event indicates that the value of the LN control point characteristic of the Location and Navigation service has been changed by the Collector. If additional parameter is present, the parameter corresponding to the OpCode is set.				
Parameters:				
	conhdl	Connection handle		
	wr_cp_info	LN control point setting value		
		OpCode	RBLE_LNP_OP_SET_CUMULATIVE_CODE	Set cumulative value (set the *cumulative_value*)
			RBLE_LNP_OP_MASK_LS_CONTENTS_CODE	Mask Location and Speed Characteristic Content (set the *content_mask*)
			RBLE_LNP_OP_NAVIGATION_CONTROL_CODE	Navigation Control (set the *control_val*)
			RBLE_LNP_OP_REQ_NUM_OF_ROUTE_CODE	Request Number of Routes
			RBLE_LNP_OP_REQ_NAME_OF_ROUTE_CODE	Request Name of Route (set the *route_num*)
			RBLE_LNP_OP_SELECT_ROUTE_CODE	Select Route (set the *route_num*)
			RBLE_LNP_OP_SET_FIX_RATE_CODE	Set Fix Rate (set the *fix_rate*)
			RBLE_LNP_OP_SET_ELEVATION_CODE	Set Elevation (set the *elevation*)
		cumulative_value	Total Distance (unit: 1/10[m], range: 0 to 1677721.5m)	
		content_mask	Mask settings of Location and Speed (0: Leave, 1: Turn off) bit[0] : Instantaneous Speed bit[1] : Total Distance bit[2] : Location bit[3] : Elevation bit[4] : Heading bit[5] : Rolling Time bit[6] : UTC Time bit[7 - 15] : Reserved	
		route_num	Number of routes	
		control_val	Navigation Control parameter	
			RBLE_LNP_CNTL_NAVI_STOP	Stop Navigation
			RBLE_LNP_CNTL_NAVI_START	Start Navigation to the first waypoint
			RBLE_LNP_CNTL_NAVI_PAUSE	Pause Navigation keeping the next waypoint
			RBLE_LNP_CNTL_NAVI_RESUME	Continue Navigation from the point where navigation was paused
			RBLE_LNP_CNTL_NAVI_SKIP	Skip waypoint
			RBLE_LNP_CNTL_NAVI_SET_NEAREST	Select nearest waypoint on a route

RBLE_LNP_EVENT_SENSOR_CHG_LN_CP_IND			
		fix_rate	Fix Rate (unit: 1[second])
		elevation	Elevation (unit: 1/100[m], range: -83886.08m to 83886.07m)

3.3.10 RBLE_LNP_EVENT_SENSOR_CFG_INDNTF_IND

RBLE_LNP_EVENT_SENSOR_CFG_INDNTF_IND			
This event indicates that the value of the client characteristic configuration descriptor of the Location and Navigation service or the battery service has been set by the Collector.			
Parameters:			
	conhdl	Connection handle	
	char_code	RBLE_LNP_LOCATION_SPEED_CODE	Location and Speed client characteristic configuration descriptor
		RBLE_LNP_LN_CONTROL_POINT_CODE	LN control point client characteristic configuration descriptor
		RBLE_LNP_NAVIGATION_CODE	Navigation client characteristic configuration descriptor
		RBLE_LNP_BATTERY_LEVEL_CODE	Battery level client characteristic configuration descriptor
	cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification/ indication
		RBLE_PRF_START_NTF	Start notification
		RBLE_PRF_START_IND	Start indication

3.3.11 RBLE_LNP_EVENT_SENSOR_COMMAND_DISALLOWED_IND

RBLE_LNP_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_LNP_SENSOR_ENABLE	Sensor role enable command
		RBLE_CMD_LNP_SENSOR_DISABLE	Sensor role disable command
		RBLE_CMD_LNP_SENSOR_SEND_LOCATION_SPEED	Location and Speed send command
		RBLE_CMD_LNP_SENSOR_SEND_LN_CONTROL_POINT	LN control point send command
		RBLE_CMD_LNP_SENSOR_SEND_NAVIGATION	Navigation send command
		RBLE_CMD_LNP_SENSOR_SEND_BATTERY_LEVEL	Battery level send command

3.3.12 RBLE_LNP_EVENT_COLLECTOR_ENABLE_COMP

RBLE_LNP_EVENT_COLLECTOR_ENABLE_COMP		
This event reports the result of enabling the Collector role (RBLE_LNP_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	
lns	*shdl*	Location and Navigation service start handle
	ehdl	Location and Navigation service end handle
	ln_feature_char_hdl	LN Feature characteristic handle
	ln_feature_val_hdl	LN Feature characteristic value handle
	ln_feature_prop	LN Feature characteristic property
	location_speed_char_hdl	Location and Speed characteristic handle
	location_speed_val_hdl	Location and Speed characteristic value handle
	location_speed_cfg_hdl	Location and Speed client characteristic configuration descriptor handle
	location_speed_prop	Location and Speed characteristic property
	position_quality_char_hdl	Position Quality characteristic handle
	position_quality_val_hdl	Position Quality characteristic value handle
	position_quality_prop	Position Quality characteristic property
	ln_cp_char_hdl	LN control point characteristic handle
	ln_cp_val_hdl	LN control point characteristic value handle
	ln_cp_cfg_hdl	LN control point client characteristic configuration descriptor handle
	ln_cp_prop	LN control point characteristic property
	navigation_char_hdl	Navigation characteristic handle
	navigation_val_hdl	Navigation characteristic value handle
	navigation_cfg_hdl	Navigation client characteristic configuration descriptor handle
	navigation_prop	Navigation 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
	bas	*shdl*	Battery service start handle
		ehdl	Battery service end handle
		battery_lvl_char_hdl	Battery Level characteristic handle
		battery_lvl_val_hdl	Battery Level characteristic value handle
		battery_lvl_cfg_hdl	Battery Level characteristic configuration descriptor handle
		battery_lvl_prop	Battery Level property

3.3.13 RBLE_LNP_EVENT_COLLECTOR_DISABLE_COMP

RBLE_LNP_EVENT_COLLECTOR_DISABLE_COMP		
This event reports the result of disabling the Collector role (RBLE_LNP_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

3.3.14 RBLE_LNP_EVENT_COLLECTOR_ERROR_IND

RBLE_LNP_EVENT_COLLECTOR_ERROR_IND		
This event indicates an error code unique to the LNP 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

3.3.15 RBLE_LNP_EVENT_COLLECTOR_LOCATION_SPEED_NTF

RBLE_LNP_EVENT_COLLECTOR_LOCATION_SPEED_NTF				
This event indicates the Location and Speed information sent from the Sensor.				
Parameters:				
	conhdl	Connection handle		
	location_speed_info	*flags*	Flag that defines whether there is a data field in the characteristic value or not	
		instant_speed	Instantaneous Speed (unit: 1/100[m/s])	
		total_distance	Total Distance (unit: 1/10[m], range: 0 to 1677721.5m)	
		latitude	Latitude (unit: 1/10^7[degree], WGS-84 format)	
		longitude	Longitude (unit: 1/10^7[degree], WGS-84 format)	
		elevation	Elevation (unit: 1/100[m], range: -83886.08m to 83886.07m)	
		heading	Heading (unit: 1/100[degree], WGS-84 format)	
		rolling_time	Rolling Time (unit: [second])	
		utc_time	UTC time at the sensor when the position was received	
			year	Year
			month	Month
			day	Day
			hour	Hour
			min	Minute
			sec	Second

3.3.16 RBLE_LNP_EVENT_COLLECTOR_LN_CP_IND

RBLE_LNP_EVENT_COLLECTOR_LN_CP_IND
This event indicates the response of LN control point from the Sensor. Check the *request_op_code* and the *response_value* whether the operation were sent by the LN control point characteristic setting (RBLE_LNP_Collector_Write_LN_Control_Point). If the *response_value* is RBLE_LNP_RES_SUCCESS_CODE, the value of the member corresponding to *request_opcode* is available.
Parameters:

conhdl	Connection handle		
ind_cp_info	*OpCode*	RBLE_LNP_OP_RESPONSE_CODE	Response code
	request_op_code	RBLE_LNP_OP_SET_CUMULATIVE_CODE	Set cumulative value
		RBLE_LNP_OP_MASK_LS_CONTENTS_CODE	Mask Location and Speed Characteristic Content
		RBLE_LNP_OP_NAVIGATION_CONTROL_CODE	Navigation control
		RBLE_LNP_OP_REQ_NUM_OF_ROUTE_CODE	Request number of routes (set the *route_num*)
		RBLE_LNP_OP_REQ_NAME_OF_ROUTE_CODE	Request name of route (set the *name_size*, and *route_name*)
		RBLE_LNP_OP_SELECT_ROUTE_CODE	Select route
		RBLE_LNP_OP_SET_FIX_RATE_CODE	Set Fix Rate
		RBLE_LNP_OP_SET_ELEVATION_CODE	Set Elevation
	reqponse_value	RBLE_LNP_RES_SUCCESS_CODE	*Success*
		RBLE_LNP_RES_NOT_SUPPORTED_CODE	*Op Code not supported*
		RBLE_LNP_RES_INVALID_PARAM_CODE	*Invalid parameter*
		RBLE_LNP_RES_OP_FAILED_CODE	*Operation failed*
	route_num	Number of routes	
	name_size	Length of route name string	
	route_name[RBLE_LNP_SENSORE_ROUTE_NAME_MAX]	Root name of the specified route (UTF-8 format)	

3.3.17 RBLE_LNP_EVENT_COLLECTOR_NAVIGATION_NTF

RBLE_LNP_EVENT_COLLECTOR_NAVIGATION_NTF				
This event indicates the Navigation information sent from the Sensor.				
Parameters:				
	conhdl	Connection handle		
	navigation_info	*flags*	Flag that defines whether there is a data field in the characteristic value or not	
		bearing	Direction where the user should be heading to reach the next waypoint or final destination (unit: 1/100[degree], WSG-84 format)	
		heading	Direction where the user is heading to (unit: 1/100[degree], WSG-84 format)	
		remain_dis	Remaining distance (unit: 1/10[m], range: 0 to 1677721.5m)	
		remain_v_dis	Remaining vertical distance (unit: 1/100[m], range: -83886.08m to 83886.07m)	
		estimate_time	Estimated time of arrival	
			year	Year
			month	Month
			day	Day
			hour	Hour
			min	Minute
			sec	Second

3.3.18 RBLE_LNP_EVENT_COLLECTOR_BATTERY_LEVEL_NTF

RBLE_LNP_EVENT_COLLECTOR_BATTERY_LEVEL_NTF		
This event indicates the battery level sent from the Sensor.		
Parameters:		
	conhdl	Connection handle
	battery_level	Battery level

3.3.19 RBLE_LNP_EVENT_COLLECTOR_READ_CHAR_RESPONSE

RBLE_LNP_EVENT_COLLECTOR_READ_CHAR_RESPONSE

This event reports the response to the characteristic value read request (RBLE_LNP_Collector_Read_Char).
Read out the read data in accordance with the contents of the request.
When Position Quality is read(RBLE_LNPC_RD_LNS_POSITION_QUAITY), structure and data size change with Flags.

The structure of the data is as follows (It describes the structure of when all the fields present.):

・RBLE_LNPC_RD_LNS_LN_FEATURE

LSB	Octet0	Octet1	Octet2	Octet3	Octet4	Octet5	MSB
	LN Feature bit[7-0]	*LN Feature bit[15-8]*	*LN Feature bit[23-16]-*	*LN Feature bit[31-24]-*	-	-	

・RBLE_LNPC_RD_LNS_LOCATION_SPEED_CFG
・RBLE_LNPC_RD_LNS_LN_CP_CFG
・RBLE_LNPC_RD_LNS_NAVIGATION_CFG
・RBLE_LNPC_RD_BAS_BL_CFG

LSB	Octet0	Octet1	Octet2	Octet3	Octet4	Octet5	MSB
	client configuration (lower)	*client configuration (upper)*	-	-	-	-	

・RBLE_LNPC_RD_LNS_POSITION_QUALITY

LSB	Octet0	Octet1	Octet2	Octet3	Octet4	Octet5	MSB
	Flags (lower)	*Flags (upper)*	*Number of Beacons in Solution*	*Number of Beacons in View*	*Time to First Fix (lower)*	*Time to First Fix (upper)*	
	Octet6	Octet7	Octet8	Octet9	Octet10	Octet11	
	EHPE bit[7-0]	*EHPE bit[15-8]*	*EHPE bit[23-16]*	*EHPE bit[31-24]*	*EVPE bit[7-0]*	*EVPE bit[15-8]*	
	Octet12	Octet13	Octet14	Octet15	Octet16	Octet17	
	EVPE bit[23-16]	*EVPE bit[31-24]*	*HDOP*	*VDOP*	-	-	

Parameters:

conhdl	Connection handle	
att_code	0x00	Characteristic value successfully acquired
	Other than 0x00	Error occurred when acquiring characteristic value (See *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for ATT error code*.)
data	*each_len*	Length of each result
	len	Data length
	data[RBLE_ATTM_MAX_VALUE]	Read characteristic data

3.3.20 RBLE_LNP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE

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

3.3.21 RBLE_LNP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND

RBLE_LNP_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_LNP_COLLECTOR_ENABLE	Collector role enable command
		RBLE_CMD_LNP_COLLECTOR_DISABLE	Collector role disable command
		RBLE_CMD_LNP_COLLECTOR_READ_CHAR	Characteristic read command
		RBLE_CMD_LNP_COLLECTOR_WRITE_CONTROL_POINT	LN control point write command
		RBLE_CMD_LNP_COLLECTOR_WRITE_CHAR	Characteristic write command

3.4 Message Sequence Chart

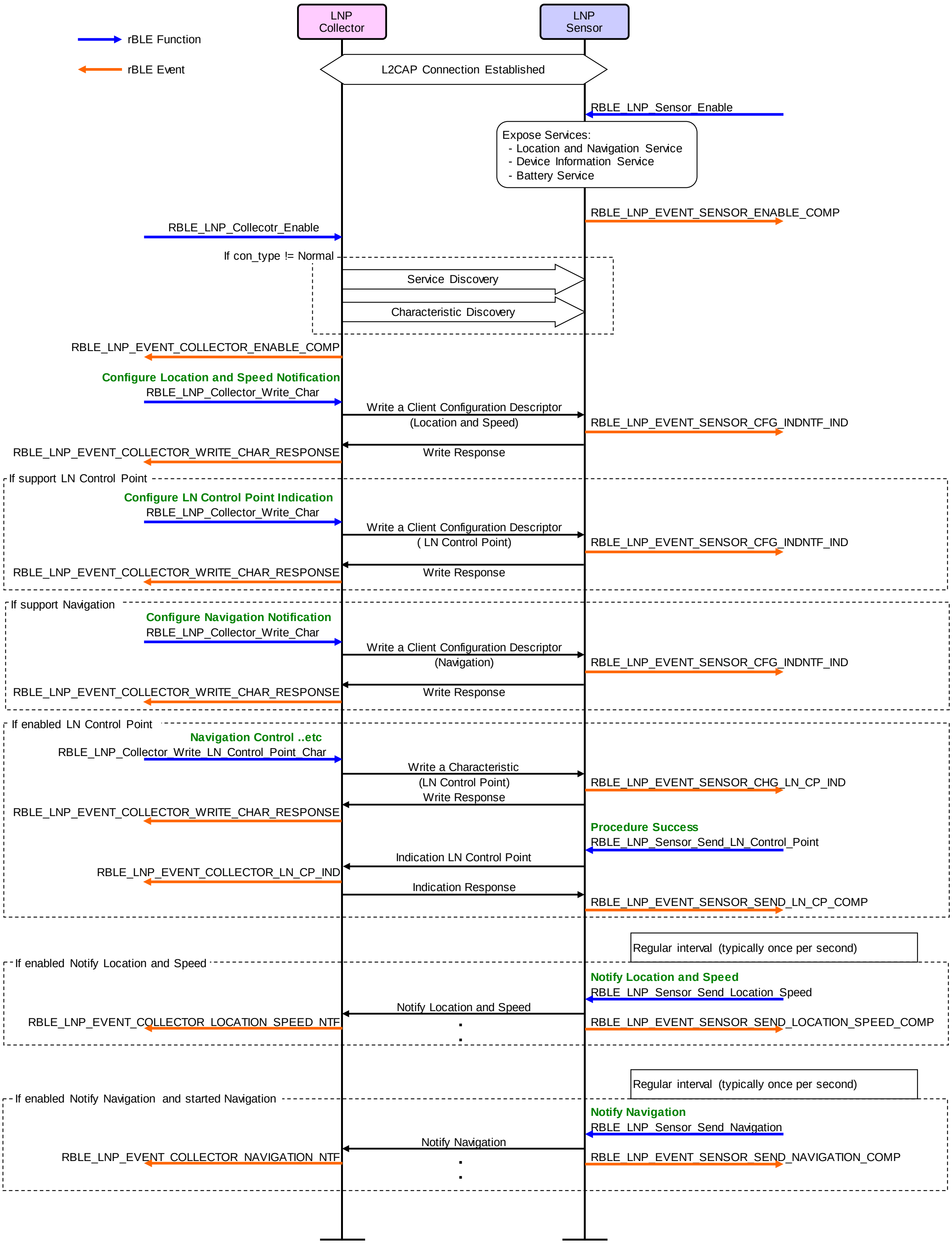

LNP Collector
LNP Sensor
rBLE Function
rBLE Event
L2CAP Connection Established
RBLE_LNP_Sensor_Enable
Expose Services:
- Location and Navigation Service
- Device Information Service
- Battery Service
RBLE_LNP_EVENT_SENSOR_ENABLE_COMP
RBLE_LNP_Collecotr_Enable
If con_type != Normal
Service Discovery
Characteristic Discovery
RBLE_LNP_EVENT_COLLECTOR_ENABLE_COMP
Configure Location and Speed Notification
RBLE_LNP_Collector_Write_Char
Write a Client Configuration Descriptor
(Location and Speed)
RBLE_LNP_EVENT_SENSOR_CFG_INDNTF_IND
RBLE_LNP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE
Write Response
If support LN Control Point
Configure LN Control Point Indication
RBLE_LNP_Collector_Write_Char
Write a Client Configuration Descriptor
(LN Control Point)
RBLE_LNP_EVENT_SENSOR_CFG_INDNTF_IND
RBLE_LNP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE
Write Response
If support Navigation
Configure Navigation Notification
RBLE_LNP_Collector_Write_Char
Write a Client Configuration Descriptor
(Navigation)
RBLE_LNP_EVENT_SENSOR_CFG_INDNTF_IND
RBLE_LNP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE
Write Response
If enabled LN Control Point
Navigation Control ..etc
RBLE_LNP_Collector_Write_LN_Control_Point_Char
Write a Characteristic
(LN Control Point)
RBLE_LNP_EVENT_SENSOR_CHG_LN_CP_IND
Write Response
RBLE_LNP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE
Procedure Success
RBLE_LNP_Sensor_Send_LN_Control_Point
Indication LN Control Point
RBLE_LNP_EVENT_COLLECTOR_LN_CP_IND
Indication Response
RBLE_LNP_EVENT_SENSOR_SEND_LN_CP_COMP
Regular interval (typically once per second)
If enabled Notify Location and Speed
Notify Location and Speed
RBLE_LNP_Sensor_Send_Location_Speed
Notify Location and Speed
RBLE_LNP_EVENT_COLLECTOR_LOCATION_SPEED_NTF
RBLE_LNP_EVENT_SENSOR_SEND_LOCATION_SPEED_COMP
Regular interval (typically once per second)
If enabled Notify Navigation and started Navigation
Notify Navigation
RBLE_LNP_Sensor_Send_Navigation
Notify Navigation
RBLE_LNP_EVENT_COLLECTOR_NAVIGATION_NTF
RBLE_LNP_EVENT_SENSOR_SEND_NAVIGATION_COMP

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

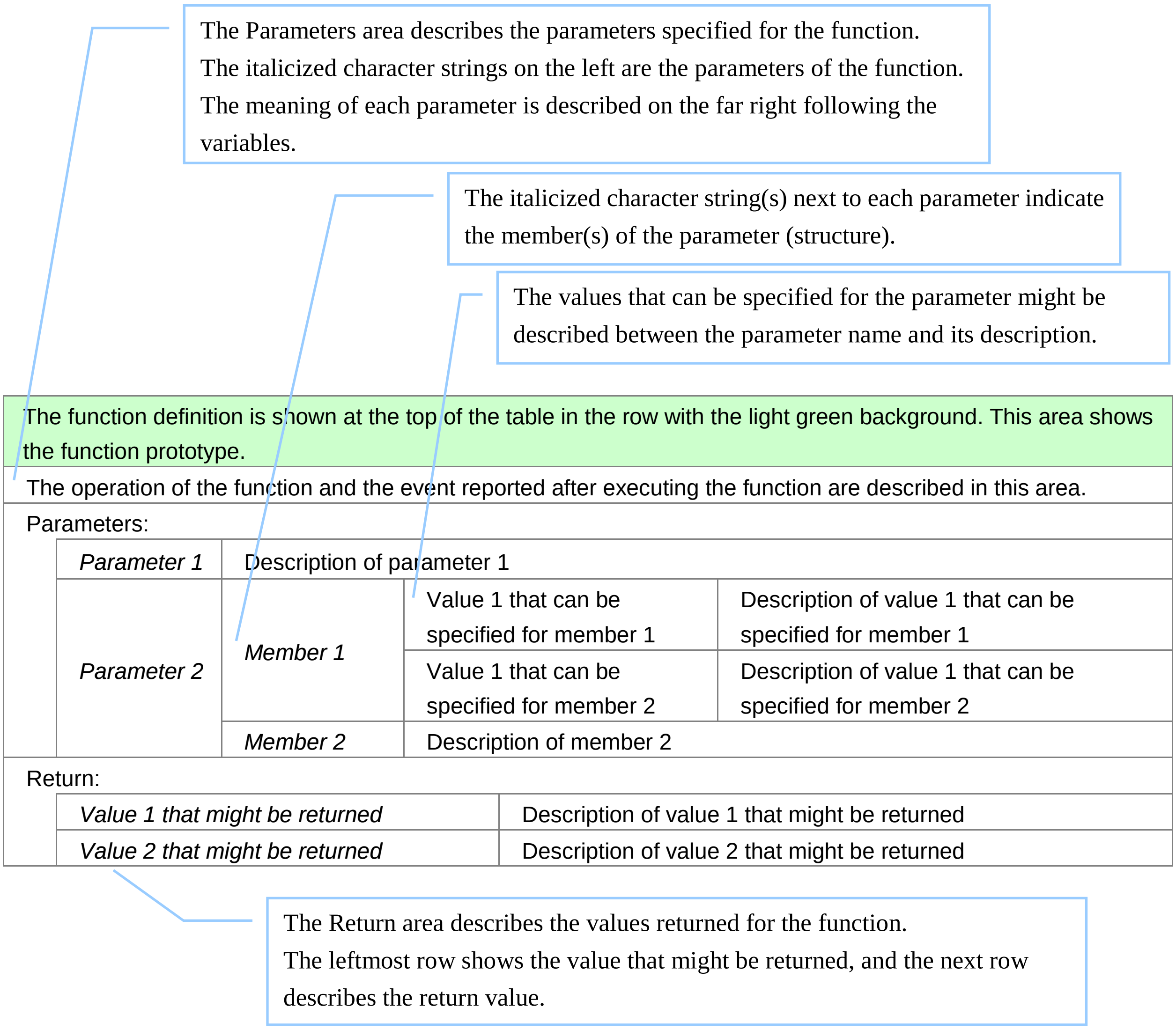
4. Notes

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. Time Profile Specification v1.0, Bluetooth SIG
26. Current Time Service Specification v1.0, Bluetooth SIG
27. Next DST Change Service Specification v1.0, Bluetooth SIG
28. Reference Time Update Service Specification v1.0, Bluetooth SIG
29. Alert Notification Service Specification v1.0, Bluetooth SIG
30. Alert Notification Profile Specification v1.0, Bluetooth SIG
31. Location and Navigation Service Specification v1.0, Bluetooth SIG
32. Location and Navigation Profile Specification v1.0, Bluetooth SIG
33. Bluetooth SIG Assigned Numbers https://www.bluetooth.org/Technical/AssignedNumbers/home.htm
34. Services & Characteristics UUID http://developer.bluetooth.org/gatt/Pages/default.aspx
35. 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.
Sensor 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: LNP

Rev.	Date	Description	
		Page	Summary
0.10	Sep 6, 2013	---	Provisional Edition issued
1.00	Nov 29, 2013	---	First Edition issued
		40	3.4 Message Sequence Chart is changed
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: LNP

Publication Date: Rev.1.02 Apr 17, 2015

Published by: Renesas Electronics Corporation

Renesas Electronics Corporation

SALES OFFICES

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

Renesas Electronics Corporation