User's Manual

Bluetooth® Low Energy Protocol Stack

API Reference Manual: GLP

Renesas MCU

Target Device

RL78/G1D

Renesas Electronics
www.renesas.com

Rev.1.00 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 Glucose profile (GLP) 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	This manual
API Reference Manual: TIP	R01UW0106E
API Reference Manual: RSCP	R01UW0107E
API Reference Manual: ANP	R01UW0108E
API Reference Manual: PASP	R01UW0109E
API Reference Manual: LNP	R01UW0113E
Application Note: Sample Program	R01AN1375E
Application Note: rBLE Command Specification	R01AN1376E

List of Abbreviations and Acronyms

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

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

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

Contents

1. Overview

This manual describes the API (Application Program Interface) of the Glucose profile (GLP) 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. Glucose Profile

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

3.1 Definitions

This section describes the definitions used by the API of the Glucose profile.

- Declaration of enumerated type for GLP event types

```
enum RBLE_GLP_EVENT_TYPE_enum {
    RBLE_GLP_EVENT_SENSOR_ENABLE_COMP = 0x01,
                                                Sensor enable completion event
                                                (Parameter: sensor_enable)
    RBLE_GLP_EVENT_SENSOR_DISABLE_COMP,         Sensor disable completion event
                                                (Parameter: sensor_disable)
    RBLE_GLP_EVENT_SENSOR_ERROR_IND,            Sensor error indication event
                                                (Parameter: error_ind)
    RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP,
                                                Measurements send completion event
                                                (Parameter: send_measurements)
    RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_CONTEXT_COMP,
                                                Measurements context send
                                                completion event
                                                (Parameter: send_measurements_context)
    RBLE_GLP_EVENT_SENSOR_SEND_RA_CP_COMP,      RA control point send completion event
                                                (Parameter: send_ra_cp)
    RBLE_GLP_EVENT_SENSOR_CHG_CP_IND,           RA control point change indication event
                                                (Parameter: glps_chg_ra_cp_ind)
    RBLE_GLP_EVENT_SENSOR_CFG_INDNTF_IND,       Characteristic configuration change
                                                indication event
                                                (Parameter: glps_cfg_indntf_ind)
    RBLE_GLP_EVENT_SENSOR_COMMAND_DISALLOWED_IND,
                                                Command disallowed indication event
                                                (Parameter: cmd_disallowed_ind)
    RBLE_GLP_EVENT_COLLECTOR_ENABLE_COMP = 0x81, Collector enable completion event
                                                (Parameter: collector_enable)
    RBLE_GLP_EVENT_COLLECTOR_DISABLE_COMP,      Collector disable completion event
                                                (Parameter: collector_disable)
    RBLE_GLP_EVENT_COLLECTOR_ERROR_IND,         Collector error indication event
                                                (Parameter: error_ind)
    RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_NTF,  Measurements indication event
                                                (Parameter: measurements_ntf)
    RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_CONTEXT_NTF,
                                                Measurement context notification event
                                                (Parameter: measurements_context_ntf)
```

```
    RBLE_GLP_EVENT_COLLECTOR_RA_CP_IND,          RA control point indication event
                                                 (Parameter: ra_cp_ind)
    RBLE_GLP_EVENT_COLLECTOR_READ_CHAR_RESPONSE,
                                                 Characteristic value read request
                                                 response event
                                                 (Parameter: rd_char_resp)
    RBLE_GLP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE,
                                                 Characteristic value write request
                                                 response event
                                                 (Parameter: wr_char_resp)
    RBLE_GLP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND
                                                 Command disallowed indication event
                                                 (Parameter: cmd_disallowed_ind)
};
```

- Declaration of data type for GLP event types

```
typedef uint8_t RBLE_GLP_EVENT_TYPE;
```

- Declaration of data type for GLP Sensor event callback function

```
typedef void ( *RBLE_GLPS_EVENT_HANDLER )( RBLE_GLPS_EVENT *event );
```

- Declaration of data type for GLP Collector event callback function

```
typedef void ( *RBLE_GLPC_EVENT_HANDLER )( RBLE_GLPC_EVENT *event );
```

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

```
enum RBLE_GLPC_RD_CHAR_CODE_enum {
    RBLE_GLPC_RD_MEASUREMENT_CFG = 0x00,          Measurements notification
    RBLE_GLPC_RD_MEASUREMENT_CONTEXT_CFG,         Measurements context notification
    RBLE_GLPC_RD_RA_CP_CFG,                       RA control point indication
    RBLE_GLPC_RD_FEATURE,                         Glucose feature
    RBLE_GLPC_RD_DIS_MANUF,                       Sensor manufacturer name
    RBLE_GLPC_RD_DIS_MODEL,                       Sensor model number
    RBLE_GLPC_RD_DIS_SERNB,                       Sensor serial number
    RBLE_GLPC_RD_DIS_HWREV,                       Sensor hardware revision
    RBLE_GLPC_RD_DIS_FWREV,                       Sensor firmware revision
    RBLE_GLPC_RD_DIS_SWREV,                       Sensor software revision
    RBLE_GLPC_RD_DIS_SYSID,                       Sensor system ID
    RBLE_GLPC_RD_DIS_IEEE,                        Sensor IEEE certification
                                                  information
};
```

- Declaration of enumerated type for glucose service characteristic value settings

```
enum RBLE_GLPC_WR_CHAR_CODE_enum {
    RBLE_GLPC_WR_MEASUREMENT_CODE = 0x01,         Glucose measurement
                                                  notification setting
    RBLE_GLPC_WR_MEASUREMENT_CONTEXT_CODE,        Glucose measurement context
                                                  notification setting
```

```
    RBLE_GLPC_WR_RA_CONTROL_POINT_CODE,              RA control point
                                                     indication setting
};
```

- Declaration of enumerated type for glucose service characteristic code

```
enum RBLE_GLPC_WR_CHAR_CODE_enum {
    RBLE_GLPC_WR_MEASUREMENT_CODE = 0x01,            Glucose measurement characteristic
    RBLE_GLPC_WR_MEASUREMENT_CONTEXT_CODE,           Glucose measurement context
                                                     characteristic
    RBLE_GLPC_WR_RA_CONTROL_POINT_CODE               RA control point characteristic
};
```

- Declaration of enumerated type for glucose type setting

```
enum RBLE_GLPC_TYPE_enum {
    RBLE_GLP_TYPE_CAPILLARY_WHOLE_BLOOD = 0x01,      Capillary Whole blood
    RBLE_GLP_TYPE_CAPILLARY_PLASMA,                  Capillary Plasma
    RBLE_GLP_TYPE_VENOUS_WHOLE_BLOOD,                Venous Whole blood
    RBLE_GLP_TYPE_VENOUS_PLASMA,                     Venous Plasma
    RBLE_GLP_TYPE_ARTERIAL_WHOLE_BLOOD,              Arterial Whole blood
    RBLE_GLP_TYPE_ARTERIAL_PLASMA,                   Arterial Plasma
    RBLE_GLP_TYPE_UNDETERMINED_WHOLE_BLOOD,          Undetermined Whole blood
    RBLE_GLP_TYPE_UNDETERMINED_PLASMA,               Undetermined Plasma
    RBLE_GLP_TYPE_ISF,                               Interstitial Fluid (ISF)
    RBLE_GLP_TYPE_CONTROL_SOLUTION                   Control Solution
};
```

- Declaration of enumerated type for glucose sample location setting

```
enum RBLE_GLP_SAMPLELOC_enum {
    RBLE_GLP_SAMPLELOC_FINGER = 0x01,                Finger
    RBLE_GLP_SAMPLELOC_AST,                          Alternate Site Test (AST)
    RBLE_GLP_SAMPLELOC_EARLOBE,                      Earlobe
    RBLE_GLP_SAMPLELOC_CONTROL_SOLUTION,             Control solution
    RBLE_GLP_SAMPLELOC_NOT_AVALABLE                  Sample location value not available
};
```

- Declaration of enumerated type for glucose sensor status annunciation setting

```
enum RBLE_GLP_SENSORSTATUS_enum {
    RBLE_GLP_SENSORSTATUS_DEVICE_BATTERY_LOW = 0x0001,
                                                     Device battery low at time
                                                     of measurement
    RBLE_GLP_SENSORSTATUS_MALFUNCTION_FAULTING = 0x0002,

                                                     Sensor malfunction or faulting
                                                     at time of measurement
    RBLE_GLP_SENSORSTATUS_INSUFFICIENT = 0x0004,     Sample size for blood or control
                                                     solution insufficient at time
                                                     of measurement
    RBLE_GLP_SENSORSTATUS_STRIP_INSERTION_ERROR = 0x0008,
```

```
                                                    Strip insertion error
    RBLE_GLP_SENSORSTATUS_STRIP_TYPE_INCORRECT = 0x0010,
                                                    Strip type incorrect for device
    RBLE_GLP_SENSORSTATUS_RESULT_HIGH     = 0x0020,    Sensor result higher than
                                                    the device can process
    RBLE_GLP_SENSORSTATUS_RESULT_LOW = 0x0040,         Sensor result lower than
                                                    the device can process
    RBLE_GLP_SENSORSTATUS_TEMPERATURE_HIGH = 0x0080,   Sensor temperature too high
                                                    for valid test/result at time
                                                    of measurement
    RBLE_GLP_SENSORSTATUS_TEMPERATURE_LOW = 0x0100,    Sensor temperature too low
                                                    for valid test/result at time
                                                    of measurement
    RBLE_GLP_SENSORSTATUS_READ_INTERRUPTED = 0x0200,   Sensor read interrupted because
                                                    strip was pulled too soon at time
                                                    of measurement.
    RBLE_GLP_SENSORSTATUS_GENERAL_DEVICE_FAULT = 0x0400,
                                                    General device fault has occurred
                                                    in the sensor.
    RBLE_GLP_SENSORSTATUS_TIME_FAULT = 0x0800          Time fault has occurred in
                                                    the sensor and time
                                                    may be inaccurate.
};
```

- Declaration of enumerated type for glucose carbohydrate ID setting

```
enum RBLE_GLP_CARBOHYDRATEID_enum {
    RBLE_GLP_CARBOHYDRATEID_BREAKFAST = 0x01,          Breakfast
    RBLE_GLP_CARBOHYDRATEID_LUNCH,                     Lunch
    RBLE_GLP_CARBOHYDRATEID_DINNER,                    Dinner
    RBLE_GLP_CARBOHYDRATEID_SNACK,                     Snack
    RBLE_GLP_CARBOHYDRATEID_DRINK,                     Drink
    RBLE_GLP_CARBOHYDRATEID_SUPPER,                    Supper
    RBLE_GLP_CARBOHYDRATEID_BRUNCH                     Brunch
};
```

- Declaration of enumerated type for glucose meal setting

```
enum RBLE_GLP_MEAL_enum {
    RBLE_GLP_MEAL_PREPRANDIAL = 0x01,                  Preprandial (before meal)
    RBLE_GLP_MEAL_POSTPRANDIAL,                        Postprandial (after meal)
    RBLE_GLP_MEAL_FASTING,                             Fasting
    RBLE_GLP_MEAL_CASUAL,                              Casual (snacks, drinks, etc.)
    RBLE_GLP_MEAL_BEDTIME                              Bedtime
};
```

- Declaration of enumerated type for glucose tester setting

```
enum RBLE_GLP_TESTER_enum {
    RBLE_GLP_TESTER_SELF = 0x01,                       Self
    RBLE_GLP_TESTER_HEALTH_CARE_PRO,                   Health care professional
    RBLE_GLP_TESTER_LAB_TEST,                          Lab test
```

```
    RBLE_GLP_TESTER_NOT_AVAILABLE                    Tester value not available
};
```

- Declaration of enumerated type for glucose health setting

```
enum RBLE_GLP_HEALTH_enum {
    RBLE_GLP_HEALTH_MINOR_ISSUE = 0x01,              Minor health issues
    RBLE_GLP_HEALTH_MAJOR_ISSUE,                     Major health issues
    RBLE_GLP_HEALTH_DURING_MENSES,                   During menses
    RBLE_GLP_HEALTH_UNDER_STRESS,                    Under stress
    RBLE_GLP_HEALTH_NO_ISSUE,                        No health issues
    RBLE_GLP_HEALTH_NOT_AVAILABLE                    Health value not available
};
```

- Declaration of enumerated type for glucose medication ID setting

```
enum RBLE_GLP_MEDICATIONID_enum {
    RBLE_GLP_MEDICATIONID_RAPID_INSULIN = 0x01,      Rapid acting insulin
    RBLE_GLP_MEDICATIONID_SHORT_INSULIN,             Short acting insulin
    RBLE_GLP_MEDICATIONID_INTERMEDIATE_INSULIN,      Intermediate acting insulin
    RBLE_GLP_MEDICATIONID_LONG_INSULIN,              Long acting insulin
    RBLE_GLP_MEDICATIONID_PREMIXED_INSULIN           Pre-mixed insulin
};
```

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

```
enum RBLE_GLP_OPCODE_enum {
    RBLE_GLP_OPCODE_REPORT_RECORDS = 0x01,           Report stored records
    RBLE_GLP_OPCODE_DELETE_RECORDS,                  Delete stored records
    RBLE_GLP_OPCODE_ABORT_OPERATION,                 Abort operation
    RBLE_GLP_OPCODE_REPORT_RECORDS_NUMBER,           Report number of stored records
    RBLE_GLP_OPCODE_NUMBER_RECORDS_RESPONSE,         Number of stored records response
    RBLE_GLP_OPCODE_RESPONSE_CODE                    Response code
};
```

- Declaration of enumerated type for RA control point characteristic operator setting

```
enum RBLE_GLP_OPERATOR_enum {
    RBLE_GLP_OPERATOR_NULL = 0x00,                   Null
    RBLE_GLP_OPERATOR_ALL_RECORDS,                   All records
    RBLE_GLP_OPERATOR_LESS,                          Less than or equal to
    RBLE_GLP_OPERATOR_GREATER,                       Greater than or equal to
    RBLE_GLP_OPERATOR_WITHIN_RANGE,                  Within range of (inclusive)
    RBLE_GLP_OPERATOR_FIRST_RECORD,                  First record (i.e. oldest record)
    RBLE_GLP_OPERATOR_LAST_RECORD,                   Last record (i.e. most recent
record)
};
```

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

```
enum RBLE_GLP_OPERAND_RESPONSE_CODE_enum {
```

```
    RBLE_GLP_OPERAND_SUCCESS = 0x01,                Success
    RBLE_GLP_OPERAND_OPCODE_NOT_SUPPORTED,          Op Code not supported
    RBLE_GLP_OPERAND_INVALID_OPERATOR,              Invalid operator
    RBLE_GLP_OPERAND_OPERATOR_NOT_SUPPORTED,        Operator not supported
    RBLE_GLP_OPERAND_INVALID_OPERAND,               Invalid operand
    RBLE_GLP_OPERAND_NO_RECORD,                     No records found
    RBLE_GLP_OPERAND_ABORT_UNSUCCESSFUL,            Abort unsuccessful
    RBLE_GLP_OPERAND_NOT_COMPLETED,                 Procedure not completed
    RBLE_GLP_OPERAND_NOT_SUPPORTED                  Operand not supported
};
```

- Glucose service characteristic information structures

```
typedef struct RBLE_GLP_SENSOR_PARAM_t {
    uint16_t      glp_meas_ntf_en;          Glucose measurement notification
                                            configuration value
    uint16_t      glp_meas_context_ntf_en;  Glucose measurement context
                                            notification configuration value
    uint16_t      ra_cp_ind_en;             RA control point indication
                                            configuration value
}RBLE_GLP_SENSOR_PARAM;
```

- Date and time information structures

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

- Glucose measurement information structures

```
typedef struct RBLE_GLP_MEASUREMENTS_INFO_t {
    uint8_t          flags;                 Data field flag
    uint8_t          reserved;              Reserved
    uint16_t         seq_num;               Sequence number
    RBLE_DATE_TIME stamp;                   Time stamp
    int16_t          time_offset;           Time offset
    int16_t          concentration;         Glucose concentration
                                            (kg/L or mol/L)
    uint8_t          type;                  Type
    uint8_t          sample_location;       Sample location
    uint16_t         sensor_status_annun;   Sensor status annunciation
} RBLE_GLP_MEASUREMENTS_INFO;
```

- Glucose measurement context information structure

```
typedef struct RBLE_GLP_MEASUREMENTS_CONTEXT_INFO_t {
    uint8_t          flags;                          Data field flag
    uint8_t          reserved1;                      Reserved
    uint16_t         seq_num;                        Sequence number
    uint8_t          ex_flags;                       Extended flags
    uint8_t          carbohydrate_id;                Carbohydrate ID
    int16_t          carbohydrate_kg;                Carbohydrate – units of kilograms
    uint8_t          meal;                           Meal
    uint8_t          tester;                         Tester
    uint8_t          health;                         Health
    uint8_t          reserved2;                      Reserved
    uint16_t         exercise_duration;              Exercise duration
    uint8_t          exercise_intensity;             Exercise intensity
    uint8_t          medication_id;                  Medication ID
    int16_t          medication;                     Medication (kg or Litter)
    int16_t          HbA1c;                          HbA1c
} RBLE_GLP_MEASUREMENTS_CONTEXT_INFO;
```

- RA control point setting structure

```
typedef struct RBLE_GLP_RA_CONTROL_POINT_INFO_t{
    uint8_t          OpCode;                         Op Code
    uint8_t          racp_operator;                  Operator
    uint8_t          operand_value;                  Operand
    uint8_t          reserved;                       Reserved
    uint16_t         min_sequence_num;               Minimum sequence number
                                                     to specify a range of records
    uint16_t         max_sequence_num;               Maximum sequence number
                                                     to specify a range of records
    RBLE_DATE_TIME   min_stamp;                      Minimum time stamp
                                                     to specify a range of records
    RBLE_DATE_TIME   max_stamp;                      Minimum time stamp
                                                     to specify a range of records
} RBLE_GLP_RA_CONTROL_POINT_INFO;
```

- RA control point response structure

```
typedef struct RBLE_GLP_RA_CONTROL_POINT_IND_INFO_t {
     uint8_t         OpCode;                         Op Code
     uint8_t         racp_operator;                  Operator
     uint16_t        num_of_records;                 Number of records
     uint8_t         request_op_code;                Request Op Code
     uint8_t         response_code_value;            Response code value
} RBLE_GLP_RA_CONTROL_POINT_IND_INFO;
```

- Glucose service content structures

```
typedef struct RBLE_GLS_CONTENT_t{
    uint16_t       shdl;                    Glucose service start handle
```

```
    uint16_t        ehdl;                       Glucose service end handle
    uint16_t        glucose_meas_char_hdl;      Glucose measurement characteristic
                                                handle
    uint16_t        glucose_meas_val_hdl;       Glucose measurement characteristic
                                                value handle
    uint16_t        glucose_meas_cfg_hdl;       Glucose measurement client characteristic
                                                configuration descriptor handle
    uint8_t         glucose_meas_prop;          Glucose measurement characteristic property
    uint8_t         reserved;                   Reserved
    uint16_t        glucose_meas_context_char_hdl;
                                                Glucose measurement context
                                                characteristic handle
    uint16_t        glucose_meas_context_val_hdl;
                                                Glucose measurement context
                                                characteristic value handle
    uint16_t        glucose_meas_context_cfg_hdl;
                                                Glucose measurement context client
                                                characteristic configuration
                                                descriptor handle
    uint8_t         glucose_meas_context_prop;
                                                Glucose measurement context
                                                characteristic property
    uint8_t         reserved2;                  Reserved
    uint16_t        glucose_feature_char_hdl;   Glucose feature characteristic handle
    uint16_t        glucose_feature_val_hdl;    Glucose feature characteristic value
                                                handle
    uint8_t         glucose_feature_prop;       Glucose feature characteristic property
    uint8_t         reserved3;                  Reserved
    uint16_t        glucose_ra_cp_char_hdl;     RA control point characteristic handle
    uint16_t        glucose_ra_cp_val_hdl;      RA control point characteristic value
                                                handle
    uint16_t        glucose_ra_cp_cfg_hdl;      RA control point client characteristic
                                                configuration descriptor handle
    uint8_t         glucose_ra_cp_prop;         RA control point characteristic property
    uint8_t         reserved4;                  Reserved
}RBLE_GLS_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;
```

- GLP Sensor event parameter structures

```
typedef struct RBLE_GLPS_EVENT_t {
    RBLE_GLP_EVENT_TYPE            type;              GLP event type
    uint8_t                        reserved;          Reserved
    union Event_Gls_Parameter_u {
        Generic event
        RBLE_STATUS                status;            Status

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

        Sensor disable completion event
        struct RBLE_GLP_Sensor_Disable_t{
            uint16_t               conhdl;            Connection handle
            RBLE_GLP_SENSOR_PARAM sensor_info;        Glucose service information
        }sensor_disable;

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

        Sensor measurements send completion event
        struct RBLE_GLP_Sensor_Send_Measurements_t{
            uint16_t               conhdl;            Connection handle
            RBLE_STATUS            status;            Status
        }send_measurements;

        Sensor measurement context send completion event
        struct RBLE_GLP_Sensor_Send_Measurements_Context_t{
            uint16_t               conhdl;            Connection handle
            RBLE_STATUS            status;            Status
        }send_measurements_context;

        Sensor RA control point send completion event
        struct RBLE_GLP_Sensor_Send_RA_Control_Point_t{
            uint16_t               conhdl;            Connection handle
            RBLE_STATUS            status;            Status
        }send_ra_cp;

        Sensor RA control point change indication event
```

```
        struct RBLE_GLP_Sensor_Chg_Ra_Cp_Ind_t{
            uint16_t                conhdl;             Connection handle
            RBLE_GLP_RA_CONTROL_POINT_INFO ra_cp_info;  RA control point information
    }glps_chg_ra_cp_ind;

        Sensor configuration characteristic value indication event
        struct RBLE_GLP_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
        }glps_cfg_indntf_ind;

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

- GLP Collector event parameter structures

```
typedef struct RBLE_GLPC_EVENT_t {
    RBLE_GLP_EVENT_TYPE             type;               GLP event type
    uint8_t                         reserved;           Reserved
    union Event_Glc_Parameter_u {
        Generic event
        RBLE_STATUS                 status;             Status

        Collector enable completion event
        struct RBLE_GLP_Collector_Enable_t{
            RBLE_STATUS             status;             Status
            uint8_t                 reserved;           Reserved
            uint16_t                conhdl;             Connection handle
            RBLE_GLS_CONTENT        gls;                Glucose service content
            RBLE_DIS_CONTENT        dis;                Device information service
                                                        content
        }collector_enable;

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

Collector error indication event

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

Collector measurement information indication event

```
struct RBLE_GLP_Collector_Measurements_Ntf_t{
    uint16_t                conhdl;             Connection handle
    RBLE_GLP_MEASUREMENTS_INFO  measure_info;   Sensor measurement information
}measurements_ntf;
```

Collector measurement context information indication event

```
struct RBLE_GLP_Collector_Measurements_Context_Ntf_t{
    uint16_t                conhdl;             Connection handle
    RBLE_GLP_MEASUREMENTS_CONTEXT_INFO  measure_context_info;
                                                Sensor measurement context
                                                information
}measurements_context_ntf;
```

Collector RA control point information indication event

```
struct RBLE_GLP_Collector_RA_CP_Ind_t{
    uint16_t                conhdl;             Connection handle
    RBLE_GLP_RA_CONTROL_POINT_IND_INFO ra_cp_ind_info;
                                                RA control point information
}ra_cp_ind;
```

Collector characteristic value read request response event

```
struct RBLE_GLP_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_GLP_Collector_Write_Char_Response_t{
    uint16_t                conhdl;             Connection handle
    uint8_t                 att_code;           Status
}wr_char_resp;
```

Collector command disallowed indication event

```
struct RBLE_GLP_Collector_Command_Disallowed_Ind_t{
    RBLE_STATUS             status;             Status
    uint8_t                 reserved;           Reserved
```

```
            uint16_t                opcode;             Opcode
        }cmd_disallowed_ind;
    } param;
} RBLE_GLPC_EVENT;
```

3.2 Functions

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

Table 3-1 API Functions Used by the GLP

RBLE_GLP_Sensor_Enable	Enables the Sensor role.
RBLE_GLP_Sensor_Disable	Disables the Sensor role.
RBLE_GLP_Sensor_Send_Measurements	Sends measurements information.
RBLE_GLP_Sensor_Send_Measurements_Context	Sends measurement context information.
RBLE_GLP_Sensor_Send_RA_Control_Point	Sends RA control point information.
RBLE_GLP_Collector_Enable	Enables the Collector role.
RBLE_GLP_Collector_Disable	Disables the Collector role.
RBLE_GLP_Collector_Read_Char	Reads the characteristic value.
RBLE_GLP_Collector_Write_Char	Writes the characteristic value.
RBLE_GLP_Collector_Write_RA_Control_Point	Sets RA control point.

3.2.1 RBLE_GLP_Sensor_Enable

RBLE_STATUS RBLE_GLP_Sensor_Enable(uint16_t conhdl, uint8_t sec_lvl, uint8_t con_type, RBLE_GLP_SENSOR_PARAM *param, RBLE_GLPS_EVENT_HANDLER call_back)

This function enables the GLP Sensor role.
If the measurement result notification, the measurement context information notification or the RA control point indication has been specified from the Collector, set the indication/notification setting parameter to 0 to configure the connection. If this setting or information has been specified from the Sensor, perform a normal connection in accordance with the indication/notification setting parameter.
The result is reported by using the Sensor role enable completion event RBLE_GLP_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	*glp_meas_ntf_en*	RBLE_PRF_STOP_NTFIND		Stop notification of measurement information.
		RBLE_PRF_START_NTF		Start notification of measurement information.
	glp_meas_context_ntf_en	RBLE_PRF_STOP_NTFIND		Stop notification of measurement context information.
		RBLE_PRF_START_NTF		Start notification of measurement context information.
	ra_cp_ind_en	RBLE_PRF_STOP_NTFIND		Stop indication of RA control point.
		RBLE_PRF_START_IND		Start indication of RA control point.
call_back	Specify the callback function that reports the GLP 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_GLP_Sensor_Disable

RBLE_STATUS RBLE_GLP_Sensor_Disable(uint16_t conhdl)		
This function disables the GLP Sensor role. The result is reported by using the Sensor role disable completion event RBLE_GLP_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_GLP_Sensor_Send_Measurements

RBLE_STATUS RBLE_GLP_Sensor_Send_Measurements(uint16_t conhdl, RBLE_GLP_MEASUREMENTS_INFO *measurements_info)				
This function sends the measured value data from the sensor. The result is reported by using the Sensor measurements send completion event RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP. When sending the measured value data continuously, send the next measured value data after completion event RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP is reported.				
Parameters:				
conhdl	Connection handle			
**measurements_info*	*flags*		Flag that defines whether there is a data field in the characteristic value or not	
	seq_num		Sequence number	
	stamp	Time stamp		
		year	Year	
		month	Month	
		day	Day	
		hour	Hour	
		min	Minute	
		sec	Second	
	time_offset		Time offset	
	concentration		Glucose Concentration (kg/L or mol/L)	
	type		Type	
			RBLE_GLP_TYPE_CAPILLARY_WHOLE_BLOOD	Capillary Whole blood
			RBLE_GLP_TYPE_CAPILLARY_PLASMA	Capillary Plasma
			RBLE_GLP_TYPE_VENOUS_WHOLE_BLOOD	Venous Whole blood
			RBLE_GLP_TYPE_VENOUS_PLASMA	Venous Plasma
			RBLE_GLP_TYPE_ARTERIAL_WHOLE_BLOOD	Arterial Whole blood
			RBLE_GLP_TYPE_ARTERIAL_PLASMA	Arterial Plasma
			RBLE_GLP_TYPE_UNDETERMINED_WHOLE_BLOOD	Undetermined Whole blood
			RBLE_GLP_TYPE_UNDETERMINED_PLASMA	Undetermined Plasma
			RBLE_GLP_TYPE_ISF	Interstitial Fluid (ISF)
			RBLE_GLP_TYPE_CONTROL_SOLUTION	Control Solution
	sample_location		Sample location	
			RBLE_GLP_SAMPLELOC_FINGER	Finger

RBLE_STATUS RBLE_GLP_Sensor_Send_Measurements(uint16_t conhdl, RBLE_GLP_MEASUREMENTS_INFO *measurements_info)				
			RBLE_GLP_SAMPLELOC_AST	Alternate Site Test (AST)
			RBLE_GLP_SAMPLELOC_EARLOBE	Earlobe
			RBLE_GLP_SAMPLELOC_CONTROL_SOLUTION	Control solution
			RBLE_GLP_SAMPLELOC_NOT_AVALABLE	Sample Location value not available
		Sensor_status_annun	Sensor status annunciation	
			RBLE_GLP_SENSORSTATUS_DEVICE_BATTERY_LOW	Device battery low at time of measurement
			RBLE_GLP_SENSORSTATUS_MALFUNCTION_FAULTING	Sensor malfunction or faulting at time of measurement
			RBLE_GLP_SENSORSTATUS_INSUFFICIENT	Sample size for blood or control solution insufficient at time of measurement
			RBLE_GLP_SENSORSTATUS_STRIP_INSERTION_ERROR	Strip insertion error
			RBLE_GLP_SENSORSTATUS_STRIP_TYPE_INCORRECT	Strip type incorrect for device
			RBLE_GLP_SENSORSTATUS_RESULT_HIGH	Sensor result higher than the device can process
			RBLE_GLP_SENSORSTATUS_RESULT_LOW	Sensor result lower than the device can process
			RBLE_GLP_SENSORSTATUS_TEMPERATURE_HIGH	Sensor temperature too high for valid test/result at time of measurement
			RBLE_GLP_SENSORSTATUS_TEMPERATURE_LOW	Sensor temperature too low for valid test/result at time of measurement
			RBLE_GLP_SENSORSTATUS_READ_INTERRUPTED	Sensor read interrupted because strip was pulled too soon at time of measurement
			RBLE_GLP_SENSORSTATUS_GENERAL_DEVICE_FAULT	General device fault has occurred in the sensor
			RBLE_GLP_SENSORSTATUS_TIME_FAULT	Time fault has occurred in the sensor and time may be inaccurate
Return:				
	RBLE_OK	Success		

RBLE_STATUS RBLE_GLP_Sensor_Send_Measurements(uint16_t conhdl, RBLE_GLP_MEASUREMENTS_INFO *measurements_info)		
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.

3.2.4 RBLE_GLP_Sensor_Send_Measurements_Context

RBLE_STATUS RBLE_GLP_Sensor_Send_Measurements_Context(uint16_t conhdl, RBLE_GLP_MEASUREMENTS_INFO *measurements_context_info)			
This function sends the measurement context data from the sensor. The result is reported by using the Sensor measurements send completion event RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_CONTEXT_COMP. When sending the measurement context data continuously, send the next measurement context data after completion event RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_CONTEXT_COMP is reported.			
Parameters:			
conhdl	Connection handle		
**measurements_context_info*	*flags*	Flag that defines whether there is a data field in the characteristic value or not	
	seq_num	Sequence number	
	ex_flags	Extended flags	
	carbohydrate_id	Carbohydrate ID	
		RBLE_GLP_CARBOHYDRATEID_BREAKFAST	Breakfast
		RBLE_GLP_CARBOHYDRATEID_LUNCH	Lunch
		RBLE_GLP_CARBOHYDRATEID_DINNER	Dinner
		RBLE_GLP_CARBOHYDRATEID_SNACK	Snack
		RBLE_GLP_CARBOHYDRATEID_DRINK	Drink
		RBLE_GLP_CARBOHYDRATEID_SUPPER	Supper
		RBLE_GLP_CARBOHYDRATEID_BRUNCH	Brunch
	carbohydrate_kg	Carbohydrate – units of kilograms	
	meal	Meal	
		RBLE_GLP_MEAL_PREPRANDIAL	Preprandial (before meal)
		RBLE_GLP_MEAL_POSTPRANDIAL	Postprandial (after meal)
		RBLE_GLP_MEAL_FASTING	Fasting
		RBLE_GLP_MEAL_CASUAL	Casual (snacks, drinks, etc.)
		RBLE_GLP_MEAL_BEDTIME	Bedtime
	tester	Tester	
		RBLE_GLP_TESTER_SELF	Self
		RBLE_GLP_TESTER_HEALTH_CARE_PRO	Health Care Professional
		RBLE_GLP_TESTER_LAB_TEST	Lab test

RBLE_STATUS RBLE_GLP_Sensor_Send_Measurements_Context(uint16_t conhdl, RBLE_GLP_MEASUREMENTS_INFO *measurements_context_info)				
			RBLE_GLP_TESTER_NOT_AVAILABLE	Tester value not available
		health	Health	
			RBLE_GLP_HEALTH_MINOR_ISSUE	Minor health issues
			RBLE_GLP_HEALTH_MAJOR_ISSUE	Major health issues
			RBLE_GLP_HEALTH_DURING_MENSES	During menses
			RBLE_GLP_HEALTH_UNDER_STRESS	Under stress
			RBLE_GLP_HEALTH_NO_ISSUE	No health issues
			RBLE_GLP_HEALTH_NOT_AVAILABLE	Health value not available
		exercise_duration	Exercise Duration	
		exercise_intensity	Exercise Intensity	
		medication_id	Medication ID	
			RBLE_GLP_MEDICATIONID_RAPID_INSULIN	Rapid acting insulin
			RBLE_GLP_MEDICATIONID_SHORT_INSULIN	Short acting insulin
			RBLE_GLP_MEDICATIONID_INTERMEDIATE_INSULIN	Intermediate acting insulin
			RBLE_GLP_MEDICATIONID_LONG_INSULIN	Long acting insulin
			RBLE_GLP_MEDICATIONID_PREMIXED_INSULIN	Pre-mixed insulin
		medication	Medication (kg or litter)	
		HbA1c	HbA1c	

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

3.2.5 RBLE_GLP_Sensor_Send_RA_Control_Point

RBLE_STATUS RBLE_GLP_Sensor_Send_RA_Control_Point(uint16_t conhdl,
RBLE_GLP_RA_CONTROL_POINT_IND_INFO *ra_cp_info)

This function sends RA control point information from the sensor. After executing an operation code that is written to the operation of RA control points from the Collector, respond by using this API.
If RBLE_GLP_OPCODE_REPORT_RECORDS has been written to RA control point from the Collector, respond by setting RBLE_GLP_OPCODE_RESPONSE_CODE to OpCode, after sending the record to the Collector by calling RBLE_GLP_Sensor_Send_Measurements and RBLE_GLP_Sensor_Send_Measurements_Context.
If RBLE_GLP_OPCODE_DELETE_RECORDS has been written to RA control point from the Collector, respond by setting RBLE_GLP_OPCODE_RESPONSE_CODE to OpCode, after deleting records.
If RBLE_GLP_OPCODE_ABORT_OPERATION has been written to RA control point from the Collector, respond by setting RBLE_GLP_OPCODE_RESPONSE_CODE to OpCode, after aborting the operation being performed.
If RBLE_GLP_OPCODE_REPORT_RECORDS_NUMBER has been written to RA control point from the Collector, respond by setting RBLE_GLP_OPCODE_NUMBER_RECORDS_RESPONSE to OpCode and setting a number of records to num_of_records.
Also, set the operation code received from the collector into the request_op_code, and set the execution result of the operation into the response_code_value.
The result is reported by using the Sensor role measurements send completion event RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_CONTEXT_COMP.

Parameters:

conhdl	Connection handle			
ra_cp_info	*OpCode*	RBLE_GLP_OPCODE_NUMBER_RECORDS_RESPONSE	Number of stored records response
		RBLE_GLP_OPCODE_RESPONSE_CODE	Response Code
	racp_operator	RBLE_GLP_OPERATOR_NULL	Null
	num_of_records	Number of records	
	request_op_code	RBLE_GLP_OPCODE_REPORT_RECORDS	Report stored records
		RBLE_GLP_OPCODE_DELETE_RECORDS	Delete stored records
		RBLE_GLP_OPCODE_ABORT_OPERATION	Abort operation
		RBLE_GLP_OPCODE_REPORT_RECORDS_NUMBER	Report number of stored records
	response_code_value	RBLE_GLP_OPERAND_SUCCESS	Success
		RBLE_GLP_OPERAND_OPCODE_NOT_SUPPORTED	Op Code not supported
		RBLE_GLP_OPERAND_INVALID_OPERATOR	Invalid Operator
		RBLE_GLP_OPERAND_OPERATOR_NOT_SUPPORTED	Operator not supported
		RBLE_GLP_OPERAND_INVALID_OPERAND	Invalid Operand

RBLE_STATUS RBLE_GLP_Sensor_Send_RA_Control_Point(uint16_t conhdl, RBLE_GLP_RA_CONTROL_POINT_IND_INFO *ra_cp_info)				
			RBLE_GLP_OPERAND_ NO_RECORD	No records found
			RBLE_GLP_OPERAND_ ABORT_UNSUCCESSFU L	Abort unsuccessful
			RBLE_GLP_OPERAND_ NOT_COMPLETED	Procedure not completed
			RBLE_GLP_OPERAND_ NOT_SUPPORTED	Operand not supported
Return:				
	RBLE_OK	Success		
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.		

3.2.6 RBLE_GLP_Collector_Enable

RBLE_STATUS RBLE_GLP_Collector_Enable(uint16_t conhdl, uint8_t con_type, RBLE_GLS_CONTENT *gls, RBLE_DIS_CONTENT *dis, RBLE_GLPC_EVENT_HANDLER call_back)

This function enables the GLP Collector role and starts access to the service exposed by the GLP Sensor. The result is reported by using the Collector role enable completion event RBLE_GLP_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
*gls	*shdl*	Glucose service start handle
	ehdl	Glucose service end handle
	glucose_meas_char_hdl	Glucose measurement characteristic handle
	glucose_meas_val_hdl	Glucose measurement characteristic value handle
	glucose_meas_cfg_hdl	Glucose measurement client characteristic configuration descriptor handle
	glucose_meas_prop	Glucose measurement characteristic property
	glucose_meas_context_char_hdl	Glucose measurement context characteristic handle
	glucose_meas_context_val_hdl	Glucose measurement context characteristic value handle
	glucose_meas_context_cfg_hdl	Glucose measurement context client characteristic configuration descriptor handle
	glucose_meas_context_prop	Glucose measurement context characteristic property
	glucose_feature_char_hdl	Glucose feature characteristic handle
	glucose_feature_val_hdl	Glucose feature characteristic value handle
	glucose_feature_prop	Glucose feature characteristic property
	ra_cp_char_hdl	RA control point characteristic handle
	ra_cp_val_hdl	RA control point characteristic value handle
	ra_cp_cfg_hdl	RA control point client characteristic configuration descriptor handle
	ra_cp_prop	RA control point characteristic property
*dis	*shdl*	Device information service start handle
	ehdl	Device information service end handle
	sys_id_char_hdl	System ID characteristic handle
	sys_id_val_hdl	System ID characteristic value handle
	sys_id_prop	System ID characteristic property

		model_nb_char_hdl	Model number characteristic handle
		model_nb_val_hdl	Model number characteristic value handle
		model_nb_prop	Model number characteristic property
		serial_nb_char_hdl	Serial number characteristic handle
		serial_nb_val_hdl	Serial number characteristic value handle
		serial_nb_prop	Serial number characteristic property
		fw_rev_char_hdl	Firmware revision characteristic handle
		fw_rev_val_hdl	Firmware revision characteristic value handle
		fw_rev_prop	Firmware revision characteristic property
		hw_rev_char_hdl	Hardware revision characteristic handle
		hw_rev_val_hdl	Hardware revision characteristic value handle
		hw_rev_prop	Hardware revision characteristic property
		sw_rev_char_hdl	Software revision characteristic handle
		sw_rev_val_hdl	Software revision characteristic value handle
		sw_rev_prop	Software revision characteristic property
		manuf_name_char_hdl	Manufacturer name characteristic handle
		manuf_name_val_hdl	Manufacturer name characteristic value handle
		manuf_name_prop	Manufacturer name characteristic property
		ieee_certif_char_hdl	IEEE certification characteristic handle
		ieee_certif_val_hdl	IEEE certification characteristic value handle
		ieee_certif_prop	IEEE certification characteristic property
	call_back	Callback	

Return:

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

3.2.7 RBLE_GLP_Collector_Disable

RBLE_STATUS RBLE_GLP_Collector_Disable(uint16_t conhdl)

This function disables the GLP Collector role and terminates the access to the service exposed by GLP Sensor. The result is reported by using the Collector role disable completion event RBLE_GLP_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.

3.2.8 RBLE_GLP_Collector_Read_Char

RBLE_STATUS RBLE_GLP_Collector_Read_Char (uint16_t conhdl, uint8_t char_code)			
This function reads the characteristic value of the glucose service and the device information service. The result is reported by using the characteristic value read request response event RBLE_GLP_EVENT_COLLECTOR_READ_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	char_code	RBLE_GLPC_RD_MEASUREMENT_CFG	Measurement s notification
		RBLE_GLPC_RD_MEASUREMENT_CONTEXT_CFG	Measurement s context notification
		RBLE_GLPC_RD_RA_CP_CFG	RA control point indication
		RBLE_GLPC_RD_FEATURE	Glucose feature
		RBLE_GLPC_RD_DIS_MANUF	Sensor manufacturer name
		RBLE_GLPC_RD_DIS_MODEL	Sensor model number
		RBLE_GLPC_RD_DIS_SERNB	Sensor serial number
		RBLE_GLPC_RD_DIS_HWREV	Sensor hardware revision
		RBLE_GLPC_RD_DIS_FWREV	Sensor firmware revision
		RBLE_GLPC_RD_DIS_SWREV	Sensor software revision
		RBLE_GLPC_RD_DIS_SYSID	Sensor system ID
		RBLE_GLPC_RD_DIS_IEEE	Sensor IEEE certification information
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.9 RBLE_GLP_Collector_Write_Char

RBLE_STATUS RBLE_GLP_Collector_Write_Char(uint16_t conhdl, uint8_t char_code, uint16_t cfg_val)			
This function writes each client characteristic configuration descriptor of the glucose service. The result is reported by using the characteristic value write request response event RBLE_GLP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE.			
Parameters:			
	conhdl	Connection handle	
	char_code	RBLE_GLPC_WR_MEASURMENT_CODE	Glucose measurement client characteristic configuration descriptor
		RBLE_GLPC_WR_MEASURMENT_CONTEXT_CODE	Glucose measurement context client characteristic configuration descriptor
		RBLE_GLPC_WR_RA_CONTROL_POINT_CODE	RA control point client characteristic configuration descriptor
	cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification or indication.
		RBLE_PRF_START_NTF	Start notification.
		RBLE_PRF_START_IND	Start indication.
Return:			
	RBLE_OK	Success	
	RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

3.2.10 RBLE_GLP_Collector_Write_RA_Control_Point

RBLE_STATUS RBLE_GLP_Collector_Write_RA_Control_Point(uint16_t conhdl,
RBLE_GLP_RA_CONTROL_POINT_INFO *ra_cp_info)

This function sets the RA control point characteristic value of the glucose service.
Specify the request operations the OpCode.

The racp_operator is specified as follows.
If the OpCode is RBLE_GLP_OPCODE_REPORT_RECORDS, RBLE_GLP_OPCODE_DELETE_RECORDS or RBLE_GLP_OPCODE_REPORT_RECORDS_NUMBER, It can be specified RBLE_GLP_OPERATOR_LAST_RECORD, RBLE_GLP_OPERATOR_LESS, RBLE_GLP_OPERATOR_GREATER, RBLE_GLP_OPERATOR_WITHIN_RANGE, RBLE_GLP_OPERATOR_FIRST_RECORD or RBLE_GLP_OPERATOR_ALL_RECORDS.
If the OpCode is RBLE_GLP_OPCODE_ABORT_OPERATION, set RBLE_GLP_OPERATOR_NULL.

The operand_value, min_sequence_num, max_sequence_num, min_stamp and max_stamp are specified as follows.
If the racp_operator is RBLE_GLP_OPERATOR_ALL_RECORDS, RBLE_GLP_OPERATOR_FIRST_RECORD or RBLE_GLP_OPERATOR_LAST_RECORD, the operand_value, min_sequence_num, max_sequence_num, min_stamp and max_stamp are not used.
If the racp_operator is RBLE_GLP_OPERATOR_LESS and the operand_value is 0x01, specify the maximum sequence number of records to retrieve to the max_sequence_num.
If the racp_operator is RBLE_GLP_OPERATOR_LESS and the operand_value is 0x02, specify the maximum time stamp of records to retrieve to the max_stamp.
If the racp_operator is RBLE_GLP_OPERATOR_GREATER and the operand_value is 0x01, specify the minimum sequence number of records to retrieve to the min_sequence_num.
If the racp_operator is RBLE_GLP_OPERATOR_GREATER and the operand_value is 0x02, specify the minimum time stamp of records to retrieve to the min_stamp.
If the racp_operator is RBLE_GLP_OPERATOR_WITHIN_RANGE and the operand_value is 0x01, specify the maximum sequence number of records to retrieve to the max_sequence_num, and specify the minimum sequence number of records to retrieve to the min_sequence_num.
If the racp_operator is RBLE_GLP_OPERATOR_WITHIN_RANGE and the operand_value is 0x02, specify the maximum time stamp of records to retrieve to the max_stamp, and specify the minimum time stamp of records to retrieve to the min_stamp.

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

Parameters:

conhdl	Connection handle		
ra_cp_info	*RA control point setting value*		
	OpCode	RBLE_GLP_OPCODE_REPORT_RECORDS	Report stored records
		RBLE_GLP_OPCODE_DELETE_RECORDS	Delete stored records
		RBLE_GLP_OPCODE_ABORT_OPERATION	Abort operation
		RBLE_GLP_OPCODE_REPORT_RECORDS_NUMBER	Report number of stored records
	racp_operator	Operator	
		RBLE_GLP_OPERATOR_NULL	Null

RBLE_STATUS RBLE_GLP_Collector_Write_RA_Control_Point(uint16_t conhdl, RBLE_GLP_RA_CONTROL_POINT_INFO *ra_cp_info)				
			RBLE_GLP_OPERATOR_ALL_RECORDS	All records
			RBLE_GLP_OPERATOR_LESS	Less than or equal to
			RBLE_GLP_OPERATOR_GREATER	Greater than or equal to
			RBLE_GLP_OPERATOR_WITHIN_RANGE	Within range of (inclusive)
			RBLE_GLP_OPERATOR_FIRST_RECORD	First record(i.e. oldest record)
			RBLE_GLP_OPERATOR_LAST_RECORD	Last record (i.e. most recent record)
		operand_value	Operand value	
		min_sequence_num	Minimum sequence number to specify a range of records	
		max_sequence_num	Maximum sequence number to specify a range of records	
		min_stamp	Minimum time stamp to specify a range of records	
			year	Year
			month	Month
			day	Day
			hour	Hour
			min	Minute
			sec	Second
		max_stamp	Minimum time stamp to specify a range of records	
			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.

3.3 Events

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

Table 3-2 Events Defined for the GLP

RBLE_GLP_EVENT_SENSOR_ENABLE_COMP	Sensor role enable completion event
RBLE_GLP_EVENT_SENSOR_DISABLE_COMP	Sensor role disable completion event
RBLE_GLP_EVENT_SENSOR_ERROR_IND	Sensor role error indication event
RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP	Glucose measurements send completion event
RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_CONTEXT_COMP	Glucose measurement context send completion event
RBLE_GLP_EVENT_SENSOR_SEND_RA_CP_COMP	RA control point send completion event
RBLE_GLP_EVENT_SENSOR_CHG_RA_CP_IND	RA control point change indication event
RBLE_GLP_EVENT_SENSOR_CFG_INDNTF_IND	Characteristic value indication event
RBLE_GLP_EVENT_SENSOR_COMMAND_DISALLOWED_IND	Sensor role command disallowed indication event
RBLE_GLP_EVENT_COLLECTOR_ENABLE_COMP	Collector role enable completion event
RBLE_GLP_EVENT_COLLECTOR_DISABLE_COMP	Collector role disable completion event
RBLE_GLP_EVENT_COLLECTOR_ERROR_IND	Collector role error indication event
RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_NTF	Glucose measurement notification event
RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_CONTEXT_NTF	Glucose measurement context notification event
RBLE_GLP_EVENT_COLLECTOR_RA_CP_IND	RA control point indication event
RBLE_GLP_EVENT_COLLECTOR_READ_CHAR_RESPONSE	Characteristic value read request response event
RBLE_GLP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE	Characteristic value write request response event
RBLE_GLP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND	Collector role command disallowed indication event

3.3.1 RBLE_GLP_EVENT_SENSOR_ENABLE_COMP

RBLE_GLP_EVENT_SENSOR_ENABLE_COMP		
This event reports the result of enabling the Sensor role (RBLE_GLP_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_GLP_EVENT_SENSOR_DISABLE_COMP

RBLE_GLP_EVENT_SENSOR_DISABLE_COMP				
This event reports the result of disabling the Sensor role (RBLE_GLP_Sensor_Disable).				
Parameters:				
	conhdl	Connection handle		
	sensor_info	*glp_meas_ntf_en*	RBLE_PRF_STOP_NTFIND	Stop notification of measurement information.
			RBLE_PRF_START_NTF	Start notification of measurement information.
		glp_meas_context_ntf_en	RBLE_PRF_STOP_NTFIND	Stop notification of measurement context information.
			RBLE_PRF_START_NTF	Start notification of measurement context information.
		ra_cp_ind_en	RBLE_PRF_STOP_NTFIND	Stop indication of RA control point.
			RBLE_PRF_START_IND	Start indication of RA control point.

3.3.3 RBLE_GLP_EVENT_SENSOR_ERROR_IND

RBLE_GLP_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_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_COMP

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

3.3.5 RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_CONTEXT_COMP

RBLE_GLP_EVENT_SENSOR_SEND_MEASUREMENTS_CONTEXT_COMP		
This event reports completion of sending the measurement context information (RBLE_GLP_Sensor_Send_Measurements_Context).		
Parameters:		
	conhdl	Connection handle
	status	Measured value send completion result (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)

3.3.6 RBLE_GLP_EVENT_SENSOR_SEND_RA_CP_COMP

RBLE_GLP_EVENT_SENSOR_SEND_RA_CP_COMP		
This event reports completion of sending the RA control point (RBLE_GLP_Sensor_Send_RA_Control_Point).		
Parameters:		
	conhdl	Connection handle
	status	Measured value send completion result (See *2.2* and *Bluetooth Low Energy Protocol Stack API Reference Manual: Basics, 3.2, Declaration of enumerated type for rBLE status*.)

3.3.7 RBLE_GLP_EVENT_SENSOR_CHG_RA_CP_IND

RBLE_GLP_EVENT_SENSOR_CHG_RA_CP_IND				
This event indicates that the value of the RA control point characteristic of the glucose service has been changed by the Collector. After executing an operation code that is written to the operation of the RA control point from the Collector, respond by calling RBLE_GLP_Sensor_Send_RA_Control_Point. If the OpCode is RBLE_GLP_OPCODE_REPORT_RECORDS, call the RBLE_GLP_Sensor_Send_RA_Control_Point after sending the record to the Collector by calling RBLE_GLP_Sensor_Send_Measurements and RBLE_GLP_Sensor_Send_Measurements_Context. If the OpCode is RBLE_GLP_OPCODE_DELETE_RECORDS, call the RBLE_GLP_Sensor_Send_RA_Control_Point after deleting records. If the OpCode is RBLE_GLP_OPCODE_ABORT_OPERATION, call the RBLE_GLP_Sensor_Send_RA_Control_Point after aborting the operation being performed. If the OpCode is RBLE_GLP_OPCODE_REPORT_RECORDS_NUMBER, respond a number of records by calling the RBLE_GLP_Sensor_Send_RA_Control_Point.				
Parameters:				
conhdl	Connection handle			
	OpCode	RBLE_GLP_OPCODE_REPORT_RECORDS	Report stored records	
		RBLE_GLP_OPCODE_DELETE_RECORDS	Delete stored records	
		RBLE_GLP_OPCODE_ABORT_OPERATION	Abort operation	
		RBLE_GLP_OPCODE_REPORT_RECORDS_NUMBER	Report number of stored records	
	racp_operator	Operator		
		RBLE_GLP_OPERATOR_NULL	Null	
		RBLE_GLP_OPERATOR_ALL_RECORDS	All records	
		RBLE_GLP_OPERATOR_LESS	Less than or equal to	
		RBLE_GLP_OPERATOR_GREATER	Greater than or equal to	
		RBLE_GLP_OPERATOR_WITHIN_RANGE	Within range of (inclusive)	
		RBLE_GLP_OPERATOR_FIRST_RECORD	First record(i.e. oldest record)	
		RBLE_GLP_OPERATOR_LAST_RECORD	Last record (i.e. most recent record)	
	operand_value	Operand value		
	min_sequence_num	Minimum sequence number to specify a range of records		
	max_sequence_num	Maximum sequence number to specify a range of records		
	min_stamp	Minimum time stamp to specify a range of records		
		year	Year	
		month	Month	
		day	Day	

RBLE_GLP_EVENT_SENSOR_CHG_RA_CP_IND				
			hour	Hour
			min	Minute
			sec	Second
		max_stamp	Minimum time stamp to specify a range of records	
			year	Year
			month	Month
			day	Day
			hour	Hour
			min	Minute
			sec	Second

3.3.8 RBLE_GLP_EVENT_SENSOR_CFG_INDNTF_IND

RBLE_GLP_EVENT_SENSOR_CFG_INDNTF_IND			
This event indicates that the value of the client characteristic configuration descriptor of the glucose service has been set by the Collector.			
Parameters:			
	conhdl	Connection handle	
	char_code	RBLE_GLPC_WR_MEASURMENT_CODE	Glucose measurement characteristic
		RBLE_GLPC_WR_MEASURMENT_CONTEXT_CODE	Glucose measurement context characteristic
		RBLE_GLPC_WR_RA_CONTROL_POINT_CODE	RA control point characteristic
	cfg_val	RBLE_PRF_STOP_NTFIND	Stop notification or indication.
		RBLE_PRF_START_NTF	Start notification.
		RBLE_PRF_START_IND	Start indication.

3.3.9 RBLE_GLP_EVENT_SENSOR_COMMAND_DISALLOWED_IND

RBLE_GLP_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_GLP_SENSOR_ENABLE	Sensor role enable command
		RBLE_CMD_GLP_SENSOR_DISABLE	Sensor role disable command
		RBLE_CMD_GLP_SENSOR_SEND_MEASUREMENTS	Glucose measurement send command
		RBLE_CMD_GLP_SENSOR_SEND_MEASUREMENTS_CONTEXT	Glucose measurement context send command
		RBLE_CMD_GLP_SENSOR_SEND_RA_CONTROL_POINT	RA control point send command

3.3.10 RBLE_GLP_EVENT_COLLECTOR_ENABLE_COMP

RBLE_GLP_EVENT_COLLECTOR_ENABLE_COMP		
This event reports the result of enabling the Collector role (RBLE_GLP_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	
gls	*shdl*	Glucose service start handle
	ehdl	Glucose service end handle
	glucose_meas_char_hdl	Glucose measurement characteristic handle
	glucose_meas_val_hdl	Glucose measurement characteristic value handle
	glucose_meas_cfg_hdl	Glucose measurement client characteristic configuration descriptor handle
	glucose_meas_prop	Glucose measurement characteristic property
	glucose_meas_context_char_hdl	Glucose measurement context characteristic handle
	glucose_meas_context_val_hdl	Glucose measurement context characteristic value handle
	glucose_meas_context_cfg_hdl	Glucose measurement context client characteristic configuration descriptor handle
	glucose_meas_context_prop	Glucose measurement context characteristic property
	glucose_feature_char_hdl	Glucose feature characteristic handle
	glucose_feature_val_hdl	Glucose feature characteristic value handle
	glucose_feature_prop	Glucose feature characteristic property
	ra_cp_char_hdl	RA control point characteristic handle
	ra_cp_val_hdl	RA control point characteristic value handle
	ra_cp_cfg_hdl	RA control point client characteristic configuration descriptor handle
	ra_cp_prop	RA control point characteristic property
dis	*shdl*	Device information service start handle
	ehdl	Device information service end handle
	sys_id_char_hdl	System ID characteristic handle
	sys_id_val_hdl	System ID characteristic value handle
	sys_id_prop	System ID characteristic property
	model_nb_char_hdl	Model number characteristic handle
	model_nb_val_hdl	Model number characteristic value handle
	model_nb_prop	Model number characteristic property
	serial_nb_char_hdl	Serial number characteristic handle
	serial_nb_val_hdl	Serial number characteristic value handle
	serial_nb_prop	Serial number characteristic property
	fw_rev_char_hdl	Firmware revision characteristic handle
	fw_rev_val_hdl	Firmware revision characteristic value handle
	fw_rev_prop	Firmware revision characteristic property

		hw_rev_char_hdl	Hardware revision characteristic handle
		hw_rev_val_hdl	Hardware revision characteristic value handle
		hw_rev_prop	Hardware revision characteristic property
		sw_rev_char_hdl	Software revision characteristic handle
		sw_rev_val_hdl	Software revision characteristic value handle
		sw_rev_prop	Software revision characteristic property
		manuf_name_char_hdl	Manufacturer name characteristic handle
		manuf_name_val_hdl	Manufacturer name characteristic value handle
		manuf_name_prop	Manufacturer name characteristic property
		ieee_certif_char_hdl	IEEE certification characteristic handle
		ieee_certif_val_hdl	IEEE certification characteristic value handle
		ieee_certif_prop	IEEE certification characteristic property

3.3.11 RBLE_GLP_EVENT_COLLECTOR_DISABLE_COMP

RBLE_GLP_EVENT_COLLECTOR_DISABLE_COMP	
This event reports the result of disabling the Collector role (RBLE_GLP_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

3.3.12 RBLE_GLP_EVENT_COLLECTOR_ERROR_IND

RBLE_GLP_EVENT_COLLECTOR_ERROR_IND	
This event indicates an error code unique to the GLP 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

3.3.13 RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_NTF

RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_NTF				
This event indicates the measured value sent from the Sensor.				
Parameters:				
conhdl	Connection handle			
measure_info	*flags*		Flag that defines whether there is a data field in the characteristic value or not	
	seq_num		Sequence number	
	stamp	Time stamp		
		year	Year	
		month	Month	
		day	Day	
		hour	Hour	
		min	Minute	
		sec	Second	
	time_offset		Time offset	
	concentration		Glucose Concentration (kg/L or mol/L)	
	type		Type	
			RBLE_GLP_TYPE_CAPILLARY_WHOLE_BLOOD	Capillary Whole blood
			RBLE_GLP_TYPE_CAPILLARY_PLASMA	Capillary Plasma
			RBLE_GLP_TYPE_VENOUS_WHOLE_BLOOD	Venous Whole blood
			RBLE_GLP_TYPE_VENOUS_PLASMA	Venous Plasma
			RBLE_GLP_TYPE_ARTERIAL_WHOLE_BLOOD	Arterial Whole blood
			RBLE_GLP_TYPE_ARTERIAL_PLASMA	Arterial Plasma
			RBLE_GLP_TYPE_UNDETERMINED_WHOLE_BLOOD	Undetermined Whole blood
			RBLE_GLP_TYPE_UNDETERMINED_PLASMA	Undetermined Plasma
			RBLE_GLP_TYPE_ISF	Interstitial Fluid (ISF)
			RBLE_GLP_TYPE_CONTROL_SOLUTION	Control Solution
	sample_location		Sample location	
			RBLE_GLP_SAMPLELOC_FINGER	Finger
			RBLE_GLP_SAMPLELOC_AST	Alternate Site Test (AST)
			RBLE_GLP_SAMPLELOC_EARLOBE	Earlobe

RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_NTF				
			RBLE_GLP_SAMPLELOC_CONTROL_SOLUTION	Control solution
			RBLE_GLP_SAMPLELOC_NOT_AVALABLE	Sample Location value not available
		sensor_status_annun	Sensor status annunciation	
			RBLE_GLP_SENSORSTATUS_DEVICE_BATTERY_LOW	Device battery low at time of measurement
			RBLE_GLP_SENSORSTATUS_MALFUNCTION_FAULTING	Sensor malfunction or faulting at time of measurement
			RBLE_GLP_SENSORSTATUS_INSUFFICIENT	Sample size for blood or control solution insufficient at time of measurement
			RBLE_GLP_SENSORSTATUS_STRIP_INSERTION_ERROR	Strip insertion error
			RBLE_GLP_SENSORSTATUS_STRIP_TYPE_INCORRECT	Strip type incorrect for device
			RBLE_GLP_SENSORSTATUS_RESULT_HIGH	Sensor result higher than the device can process
			RBLE_GLP_SENSORSTATUS_RESULT_LOW	Sensor result lower than the device can process
			RBLE_GLP_SENSORSTATUS_TEMPERATURE_HIGH	Sensor temperature too high for valid test/result at time of measurement
			RBLE_GLP_SENSORSTATUS_TEMPERATURE_LOW	Sensor temperature too low for valid test/result at time of measurement
			RBLE_GLP_SENSORSTATUS_READ_INTERRUPTED	Sensor read interrupted because strip was pulled too soon at time of measurement
			RBLE_GLP_SENSORSTATUS_GENERAL_DEVICE_FAULT	General device fault has occurred in the sensor
			RBLE_GLP_SENSORSTATUS_TIME_FAULT	Time fault has occurred in the sensor and time may be inaccurate

3.3.14 RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_CONTEXT_NTF

RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_CONTEXT_NTF			
This event indicates the measurement context value sent from the Sensor.			
Parameters:			
conhdl	Connection handle		
measure_context_info	*flags*	Flag that defines whether there is a data field in the characteristic value or not	
	seq_num	Sequence number	
	ex_flags	Extended flags	
	carbohydrate_id	Carbohydrate ID	
		RBLE_GLP_CARBOHYDRATEID_BREAKFAST	Breakfast
		RBLE_GLP_CARBOHYDRATEID_LUNCH	Lunch
		RBLE_GLP_CARBOHYDRATEID_DINNER	Dinner
		RBLE_GLP_CARBOHYDRATEID_SNACK	Snack
		RBLE_GLP_CARBOHYDRATEID_DRINK	Drink
		RBLE_GLP_CARBOHYDRATEID_SUPPER	Supper
		RBLE_GLP_CARBOHYDRATEID_BRUNCH	Brunch
	carbohydrate_kg	Carbohydrate – units of kilograms	
	meal	Meal	
		RBLE_GLP_MEAL_PREPRANDIAL	Preprandial (before meal)
		RBLE_GLP_MEAL_POSTPRANDIAL	Postprandial (after meal)
		RBLE_GLP_MEAL_FASTING	Fasting
		RBLE_GLP_MEAL_CASUAL	Casual (snacks, drinks, etc.)
		RBLE_GLP_MEAL_BEDTIME	Bedtime
	tester	Tester	
		RBLE_GLP_TESTER_SELF	Self
		RBLE_GLP_TESTER_HEALTH_CARE_PRO	Health Care Professional
		RBLE_GLP_TESTER_LAB_TEST	Lab test
		RBLE_GLP_TESTER_NOT_AVAILABLE	Tester value not available
	health	Health	

RBLE_GLP_EVENT_COLLECTOR_MEASUREMENTS_CONTEXT_NTF				
			RBLE_GLP_HEALTH_MINOR_ISSUE	Minor health issues
			RBLE_GLP_HEALTH_MAJOR_ISSUE	Major health issues
			RBLE_GLP_HEALTH_DURING_MENSES	During menses
			RBLE_GLP_HEALTH_UNDER_STRESS	Under stress
			RBLE_GLP_HEALTH_NO_ISSUE	No health issues
			RBLE_GLP_HEALTH_NOT_AVAILABLE	Health value not available
		exercise_duration	Exercise Duration	
		exercise_intensity	Exercise Intensity	
		medication_id	Medication ID	
			RBLE_GLP_MEDICATIONID_RAPID_INSULIN	Rapid acting insulin
			RBLE_GLP_MEDICATIONID_SHORT_INSULIN	Short acting insulin
			RBLE_GLP_MEDICATIONID_INTERMEDIATE_INSULIN	Intermediate acting insulin
			RBLE_GLP_MEDICATIONID_LONG_INSULIN	Long acting insulin
			RBLE_GLP_MEDICATIONID_PREMIXED_INSULIN	Pre-mixed insulin
		medication	Medication (kg or litter)	
		HbA1c	HbA1c	

3.3.15 RBLE_GLP_EVENT_COLLECTOR_RA_CP_IND

RBLE_GLP_EVENT_COLLECTOR_RA_CP_IND			
This event indicates the response of RA control point operation sent from the Sensor. Check that the value of the request_op_code is the same as the operation code sent by the RA control point setting API (RBLE_GLP_Collector_Write_RA_Control_Point). If the OpCode is RBLE_GLP_OPCODE_NUMBER_RECORDS_RESPONSE, retrieve the number of records from num_of_records. If the OpCode is RBLE_GLP_OPCODE_RESPONSE_CODE, confirm the execution result of operation in response_code_value.			
Parameters:			
conhdl	Connection handle		
ra_cp_info	*OpCode*	RBLE_GLP_OPCODE_NUMBER_RECORDS_RESPONSE	Number of stored records response
		RBLE_GLP_OPCODE_RESPONSE_CODE	Response Code
	racp_operator	RBLE_GLP_OPERATOR_NULL	Null
	num_of_records	Number of records	
	request_op_code	RBLE_GLP_OPCODE_REPORT_RECORDS	Report stored records
		RBLE_GLP_OPCODE_DELETE_RECORDS	Delete stored records
		RBLE_GLP_OPCODE_ABORT_OPERATION	Abort operation
		RBLE_GLP_OPCODE_REPORT_RECORDS_NUMBER	Report number of stored records
	response_code_value	RBLE_GLP_OPERAND_SUCCESS	Success
		RBLE_GLP_OPERAND_OPCODE_NOT_SUPPORTED	Op Code not supported
		RBLE_GLP_OPERAND_INVALID_OPERATOR	Invalid Operator
		RBLE_GLP_OPERAND_OPERATOR_NOT_SUPPORTED	Operator not supported
		RBLE_GLP_OPERAND_INVALID_OPERAND	Invalid Operand
		RBLE_GLP_OPERAND_NO_RECORD	No records found
		RBLE_GLP_OPERAND_ABORT_UNSUCCESSFUL	Abort unsuccessful
		RBLE_GLP_OPERAND_NOT_COMPLETED	Procedure not completed
		RBLE_GLP_OPERAND_NOT_SUPPORTED	Operand not supported

3.3.16 RBLE_GLP_EVENT_COLLECTOR_READ_CHAR_RESPONSE

RBLE_GLP_EVENT_COLLECTOR_READ_CHAR_RESPONSE			
This event reports the response to the characteristic value read request (RBLE_GLP_Collector_Read_Char). Read out the read data in accordance with the contents of the request.			
Parameters:			
conhdl	Connection handle		
att_code	0x00	Characteristic value successfully acquired	
	Other than 0x00	Error occurred when acquiring characteristic value	
data	*each_len*		Length of each result
	len		Data length
	data[RBLE_ATTM_MAX_VALUE]		Read characteristic data

3.3.17 RBLE_GLP_EVENT_COLLECTOR_WRITE_CHAR_RESPONSE

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

3.3.18 RBLE_GLP_EVENT_COLLECTOR_COMMAND_DISALLOWED_IND

RBLE_GLP_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_GLP_COLLECTOR_ENABLE	Collector role enable command
	RBLE_CMD_GLP_COLLECTOR_DISABLE	Collector role disable command
	RBLE_CMD_GLP_COLLECTOR_READ_CHAR	Characteristic read command
	RBLE_CMD_GLP_COLLECTOR_WRITE_CHAR	Characteristic write command
	RBLE_CMD_GLP_COLLECTOR_WRITE_RA_CONTROL_POINT	RA control point setup command

3.4 Message Sequence Chart

T.B.D.

4. Notes

Appendix A How to Read Definition Tables

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

A.1 How to Read Function Definition Tables

The following contents are included in the function definition tables:

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

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

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

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

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

A.2 How to Read Event Definition Tables

The following contents are included in the event definition tables:

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

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

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

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

Appendix B Referenced Documents

1. Bluetooth Core Specification v4.0, Bluetooth SIG
2. Find Me Profile Specification v1.0, Bluetooth SIG
3. Immediate Alert Service Specification v1.0, Bluetooth SIG
4. Proximity Profile Specification v1.0, Bluetooth SIG
5. Link Loss Service Specification v1.0, Bluetooth SIG
6. Tx Power Service Specification v1.0, Bluetooth SIG
7. Health Thermometer Profile Specification v1.0, Bluetooth SIG
8. Health Thermometer Service Specification v1.0, Bluetooth SIG
9. Device Information Service Specification v1.1, Bluetooth SIG
10. Blood Pressure Profile Specification v1.0, Bluetooth SIG
11. Blood Pressure Service Specification v1.0, Bluetooth SIG
12. HID over GATT Profile Specification v1.0, Bluetooth SIG
13. HID Service Specification v1.0, Bluetooth SIG
14. Battery Service Specification v1.0, Bluetooth SIG
15. Scan Parameters Profile Specification v1.0, Bluetooth SIG
16. Scan Parameters Service Specification v1.0, Bluetooth SIG
17. Heart Rate Profile Specification v1.0, Bluetooth SIG
18. Heart Rate Service Specification v1.0, Bluetooth SIG
19. Cycling Speed and Cadence Profile Specification v1.0, Bluetooth SIG
20. Cycling Speed and Cadence Service Specification v1.0, Bluetooth SIG
21. Cycling Power Profile Specification v0.9, Bluetooth SIG
22. Cycling Power Service Specification v0.9, Bluetooth SIG
23. Glucose Profile Specification v1.0, Bluetooth SIG
24. Glucose Service Specification v1.0, Bluetooth SIG
25. Bluetooth SIG Assigned Numbers https://www.bluetooth.org/Technical/AssignedNumbers/home.htm
26. Services & Characteristics UUID http://developer.bluetooth.org/gatt/Pages/default.aspx
27. Personal Health Devices Transcoding White Paper v1.2, Bluetooth SIG

Appendix C Terminology

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

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

Rev.	Date	Description	
		Page	Summary
0.12	Jan 30, 2015	---	Provisional Edition issued
1.00	Apr 17, 2015	2	The service definitions are updated.

Bluetooth Low Energy Protocol Stack
API Reference Manual: GLP

Publication Date: Rev.1.00 Apr 17, 2015

Published by: Renesas Electronics Corporation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Colophon 4.0

Bluetooth Low Energy Protocol Stack

Renesas Electronics Corporation

R01UW0103EJ0100