User's Manual

RL78/G1D Module Firmware

User's Manual

Renesas Bluetooth low energy Module

Renesas Electronics
www.renesas.com

Rev.1.02 Jan 2022

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, 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.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

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

(Rev.4.0-1 November 2017)

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit 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.

   ¾ 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.

   ¾ 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.

   ¾ 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.

   ¾ 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.

   ¾ The characteristics of Microprocessing unit or Microcontroller unit products 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

Purpose and Target Readers

This manual describe the structure and function of the firmware (RL78/G1D Module Firmware) written to the module to be used for the development of application products that use the Renesas RL78/G1D Module (RY7011). It is intended for users designing application systems using this module. 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.
RL78/G1D Module	
Firmware User's Manual	This manual
User's Manual : Hardware	R02UH0004E

Document Name	Document No.
RL78/G1D Device	
User's Manual : Hardware	R01UH0515E
User's Manual : Software	R01US0015E

Document Name	Document No.
Bluetooth Low Energy Protocol Stack	
Quick Start Guide	R01AN2767E
User's Manual	R01UW0095E
API Reference Manual : Basics	R01UW0088E
API Reference Manual : FMP (Obsolete)	R01UW0089E
API Reference Manual : PXP (Obsolete)	R01UW0090E
API Reference Manual : HTP (Obsolete)	R01UW0091E
API Reference Manual : BLP (Obsolete)	R01UW0092E
API Reference Manual : HRP (Obsolete)	R01UW0097E
API Reference Manual : GLP (Obsolete)	R01UW0103E
API Reference Manual : TIP (Obsolete)	R01UW0106E
API Reference Manual : RSCP (Obsolete)	R01UW0107E
API Reference Manual : ANP (Obsolete)	R01UW0108E
API Reference Manual : PASP (Obsolete)	R01UW0109E
Application Note : Sample Program	R01AN1375E

Application Note : rBLE Command Specification	R01AN1376E
Application Note : GUI Tool	R01AN2469E
Application Note : BLE Virtual UART Application	R01AN3130E

List of Abbreviations and Acronyms

Abbreviation	Full Form	Remark
ANP	Alert Notification Profile	
ANS	Alert Notification Service	
API	Application Programming Interface	
BLE	Bluetooth low energy	
BLP	Blood Pressure Profile	
BLS	Blood Pressure Service	
FMP	Find Me Profile	
GAP	Generic Access Profile	
GATT	Generic Attribute Profile	
GLP	Glucose Profile	
GLS	Glucose Service	
HCI	Host Controller Interface	
HRP	Heart Rate Profile	
HRS	Heart Rate Service	
HTP	Health Thermometer Profile	
HTS	Health Thermometer Service	
L2CAP	Logical Link Control and Adaptation Protocol	
LE	Low Energy	
LL	Link Layer	
MCU	Micro Controller Unit	
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	
SMP	Security Manager Protocol	
UART	Universal Asynchronous Receiver Transmitter	

Abbreviation	Full Form	Remark
RSCIP	Renesas Serial Communication Interface Protocol	
RWKE	Renesas Wireless Kernel Extension	

These commodities, technology or software, must be exported in accordance with the export administration regulations of the exporting country. Diversion contrary to the law of that country is prohibited.

Contents

1. Overview

This manual describe the structure and function of the firmware (RL78/G1D Module Firmware) written to the module to be used for the development of application products that use the Renesas RL78G1D Module (RY7011). The firmware is developed based on the Bluetooth Low Energy protocol stack (BLE software). For details about the BLE software, also refer to the "Bluetooth Low Energy Protocol Stack User's Manual"(R01UW0095) and the related documents.

2. Applicability

The descriptions in this manual apply to RL78/G1D Module Firmware Version 1.01 or later.

3. Restrictions

This section describes the restrictions that apply to Firmware.

4. Precautions

(1) When carrying out FW Update, it isn't possible to return to usual BLE communication until an update is completed.

(2) A shipping check flag is written in block 254 of a cord flash memory, so please don't rewrite.

(3) Public Bluetooth Device Address is written in block 255 of a cord flash memory, so please don't rewrite.

(4) When connected to the extension board for Bluetooth Low Energy evaluation (R0K3ZBBBDBN00BR) of the RL78/G1D evaluation board (RTK0EN0001D01001BZ) for evaluation of the RL78/G1D Module evaluation board (RM-110-RFB-2), VBUS detection of the extension board for Bluetooth Low Energy evaluation is assigned to INTP3 of the RL78/G1D Module evaluation board, so please open the external expansion pin No.18 and No.19.

5. Firmware Overview

The firmware based on Bluetooth Low Energy protocol stack of Modem configuration based upon Bluetooth version 4.1 has been written to a module. The firmware supports master and slave. The profiles adopted from the Bluetooth SIG and the general purpose communication custom profiles are installed. It is possible to data transmission with other Bluetooth Low Energy devices by connecting Host MCU which controls a module and a PC in UART. It is also possible to update the firmware with the exception of the protocol stack by FW Update profile is a custom profile.

- Supported Profiles

- Adopted Bluetooth Profiles by Bluetooth SIG

  Due to the deprecation and withdrawal plan of the profile version by Bluetooth SIG, each profile has been obsoleted because product registration using the profile supported by the Bluetooth Low Energy protocol stack is no longer possible.

  For product registration, refer to "Bluetooth LE microcomputer/module Bluetooth qualification acquisition application note" (R01AN3177).

  - ✧ Proximity (Obsolete)
  - ✧ Find Me (Obsolete)
  - ✧ Heart Rate (Obsolete)
  - ✧ Time (Obsolete)
  - ✧ Alert Notification (Obsolete)
  - ✧ Running Speed and Cadence (Obsolete)
  - ✧ Health Thermometer (Obsolete)
  - ✧ Blood Pressure (Obsolete)
  - ✧ Glucose Profile (Obsolete)
  - ✧ Phone Alert Status (Obsolete)

- Custom Profiles
  - ✧ General Purpose Communication
  - ✧ FW Update

- Module Connection Diagram

Connection with the Host MCU to control the module using the UART 2 wire branch connection method. In addition to the UART 2 wire connection as shown in the figure below, to communicate and connect with the WAKEUP pin of module branches the TxD line of Host MCU. Refer to "Bluetooth Low Energy Protocol Stack User's Manual chapter 5.4.3"(R01UW0095) about details for UART 2 wire branch connection method.

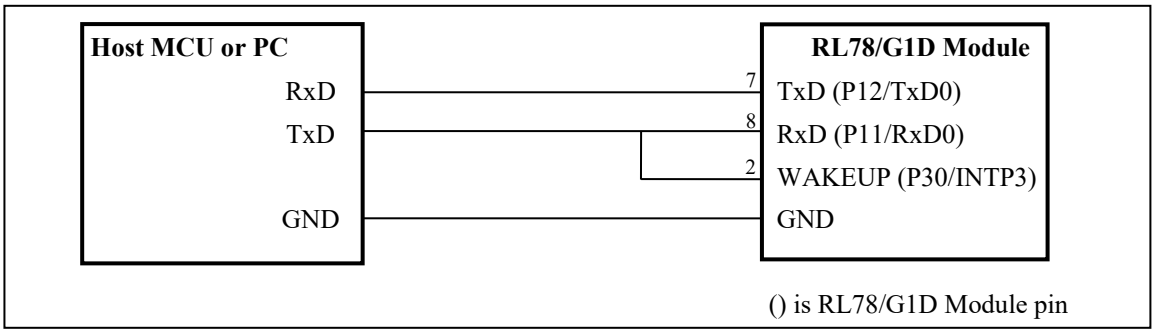


Figure 5-1 Module Connection Diagram

6. Firmware Specification

6.1 Specification

The following shows the on-board firmware specification.

- BLE Software version : V1.11

- Firmware Settings
  - ✧ BLE Software Configuration : Modem Configuration
  - ✧ Number of Simultaneous Connections : 6
  - ✧ Operating Frequency : 8 MHz
  - ✧ On-chip low-speed oscillator in the RF : use
  - ✧ DC-DC Converter : use
  - ✧ UART Communication Method : 2 wire with branch connection
  - ✧ Baud Rate : 4800, 9600, 19200, 38400, 57600, 115200, 250000 (default: 115200 bps)

- UART Settings
  - ✧ Data Length : 8 bits
  - ✧ Parity : none
  - ✧ Stop Bit : 1 bit
  - ✧ Flow Control : none

6.2 Pin Functions

Refer to "RL78/G1D Module (RY7011) User's Manual: Hardware chapter 2"(R02UH0004) about the setting of pin function in the firmware.

6.3 Public Bluetooth Device Address

A Public Bluetooth Device Address is written in block 255 of a cord flash memory. When rewriting a program of a module, please be careful so as not to erase block 255. Also refer to "8. Code flash memory rewriting" in this document.

6.4 Shipping Check Flag

A shipping check flag is written in block 254 of a cord flash memory. When rewriting a program of a module, please be careful so as not to erase block 254. Also refer to "8. Code flash memory rewriting" in this document.

6.5 Configuration

The following is a block diagram of the module and the host MCU software.

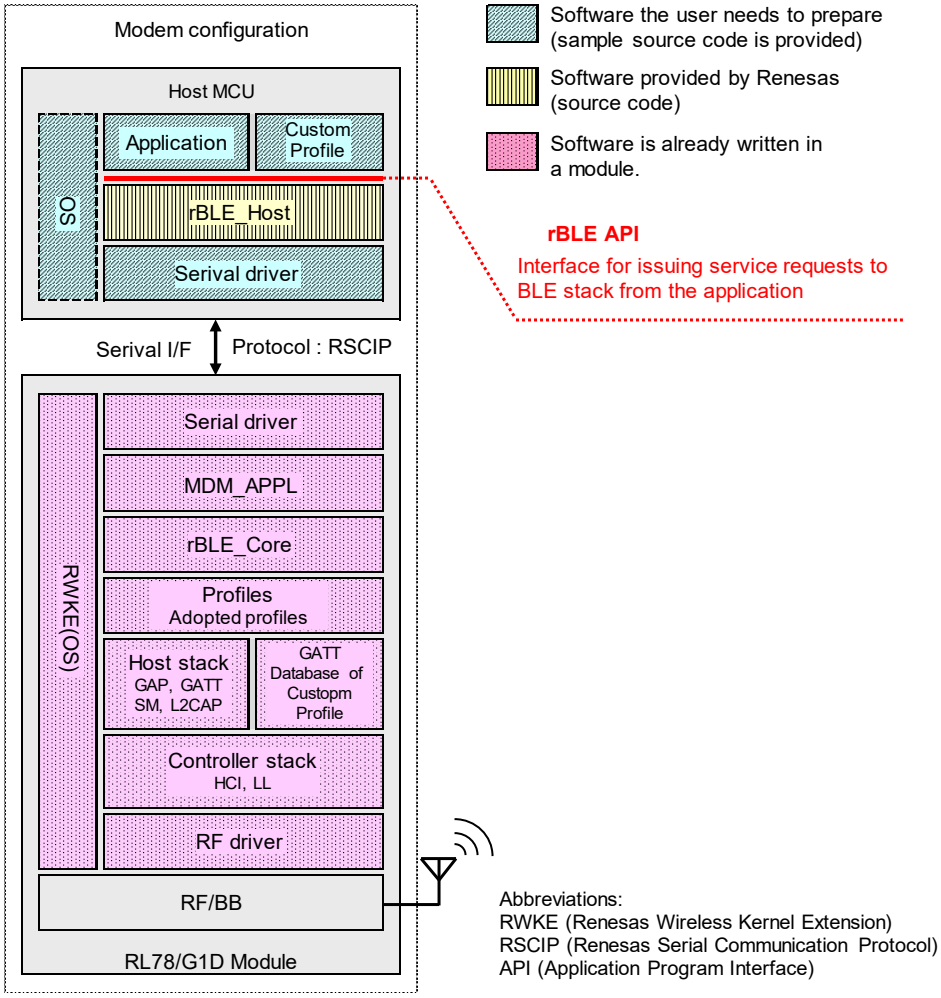


Figure 6-1 BLE Software Configuration

6.5.1 rBLE

The request from Host MCU to perform the BLE communication and a response from a module are performed in rBLE. The rBLE of Modem configuration is configured by 3 layers of the rBLE_Host, the MDM_APPL and the rBLE_Core. The rBLE_Host is located in the Host MCU side, the MDM_APPL and the rBLE_Core are located in the module side. It also provides a rBLE_API for accessing the BLE stack through the rBLE from the application as a user API. Application running on the Host MCU accesses the rBLE_Host through the rBLE_API, it can request the BLE services to the module and receive the response. Refer to "API Reference Manual : Basics"(R01UW0088) about rBLE_API specification and API Reference Manuals of each profile.

The communication command of between the Host MCU and the Module is used the rBLE command. The rBLE command is the communication command used by the modem configuration. The requests and responses that are executed through the rBLE_API, rBLE_Host converts to rBLE command. Refer to "rBLE Command Specification"(R01AN1376) about rBLE commands.

6.5.2 rBLE Command

The rBLE command is a communication command to be used in the configuration that the application is in a different MCU(Modem configuration). A command packets instruct a operation to the module from the Host MCU. A event packets notify a information to the Host MCU from the module. The Host MCU and the Module use the Renesas Serial Communication Interface Protocol (RSCIP) at a communication protocol. The RSCIP extends the SLIP(Serial Line Internet Protocol) which is prescribed in RFC1055. Error that occurred in the serial communication is retransmitted by the recovery function, to ensure the reliability of communication. Refer to "rBLE Command Specification"(R01AN1376) about the rBLE Command and the RSCIP.

6.5.3 Serial Driver

The UART 2-wire with branch connection is used at the Host the MCU and module. The 2-wire branch connection method to connect to the WAKEUP pin of the module branches the TxD line from the Host MCU. When sends a data from the Host MCU, it is capable of full-duplex communication by performing handshake to confirm the module is ready for reception. Refer to "Bluetooth Low Energy Protocol Stack User's Manual chapter 5.4.3"(R01UW0095) and "Sample Program chapter 6.1"(R01AN1375) about the UART 2-wire with branch connection.

6.5.4 BLE Stack

The profile layer, the host stack, the controller stack and RF driver that has been written to the module is the BLE stack. There is the adopted bluetooth profiles on the module side. There is a data base on the module side for a custom profile and the profile is on the Host MCU side. Refer to "Bluetooth Low Energy Protocol Stack User's Manual chapter 5"(R01UW0095) about BLE stack.

6.6 UART 2-wire with Branch Connection

TxD of the Host MCU diverges, is connected with the WAKEUP pin of the Module so that the Host MCU and the Module may make the Module get up when the Host MCU in addition to TxD that is the data signal line of a UART as shown in the following and RxD transmits data in this connected method, and it communicates.

When transmitting from the Host MCU, it is necessary to do handshaking though the full duplex transmission is possible. This is operation necessary to confirm the Module completes the preparation for the reception. Moreover, please observe by the time-out to do a reliable communication at handshaking, and execute handshaking again when you generate the time-out.

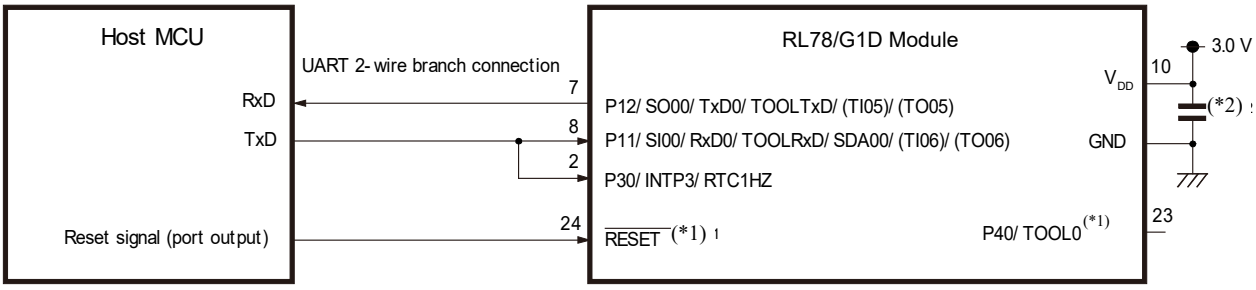


Figure 6-2 UART 2-wire with Branch Connection

Note *1. $\overline{\text{RESET}}$ and P40/TOOL0 pins are pulled-up/pulled-down with a resistor in accordance with the system requirement (see RL78/G1D User's Manual: Hardware).

*2. Insert bypass capacitor of several μF between the VDD and GND pins to suit the characteristics the power supply and wiring patterns.

Table 6-1 UART 2-wire with Branch Connection pin function

BLE MCU Pin Name	Direction	Function
TxD0	BLE MCU-> APP MCU	Serial Output Data Signal
RxD0	APP MCU-> BLE MCU	Serial Input Data Signal
INTP3(WAKEUP) - Low Active	APP MCU-> BLE MCU	External Trigger Input Signal for Wakeup APP MCU is set at an active level at the transmission request. ACK byte (0x88) reception or data reception from BLE MCU is waited for, and it returns it to an inactive level.

(1) Transmit Operation (Host MCU)

The handshaking procedure when the Host MCU transmits the RSCIP packet to the Module is following T3.

T1: The Host MCU transmits REQ byte (0xC0) for the transmission request.
T2: The Host MCU detects ACK byte (0x88) from BLE MCU or the RSCIP packet by one byte.
T3: The Host MCU transmits the RSCIP packet.

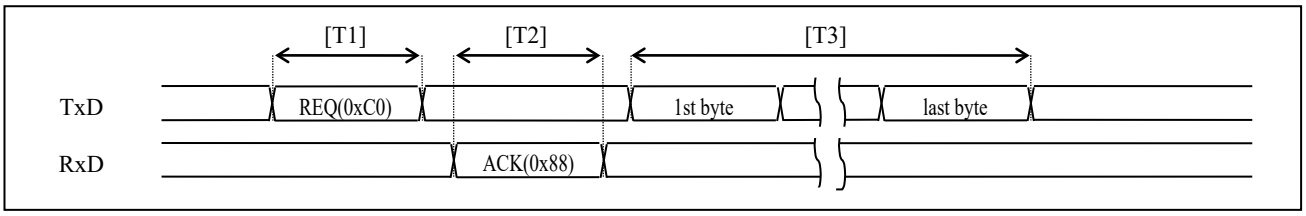


Figure 6-3 Transmit timing chart (Host MCU)

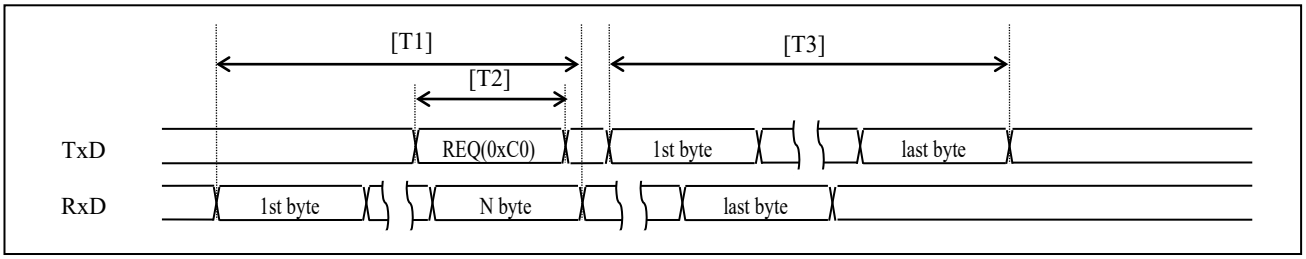


Figure 6-4 Transmit timing chart (Host MCU) (While the Module transmitting)

After the transmission request, the serial communications driver begins the time-out watch. When the time-out is generated, the serial communications driver is T1 that transmits the REQ byte for the re-transmission demand. The recommended value at the timeout period is assumed to be 5msec.

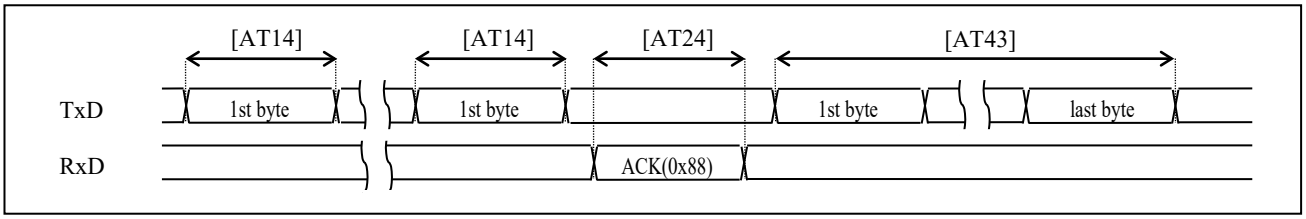


Figure 6-5 Transmit timing chart (Host MCU) (Timeout occurs)

The transmission sequence including the rBLE_Host and a serial communications driver's function calls is shown.

[When beginning to transmit]: The serial communications driver is T1 according to the call of the rBLE_Host of the transmission function that begins the transmission operation of the RSCIP packet, and transmits the REQ byte for the transmission request.

[When the transmission ends]: The serial communications driver notifies the rBLE_Host the transmission completion by calling the transmission completion notification function when the RSCIP packet transmission T1- T3 is completed.

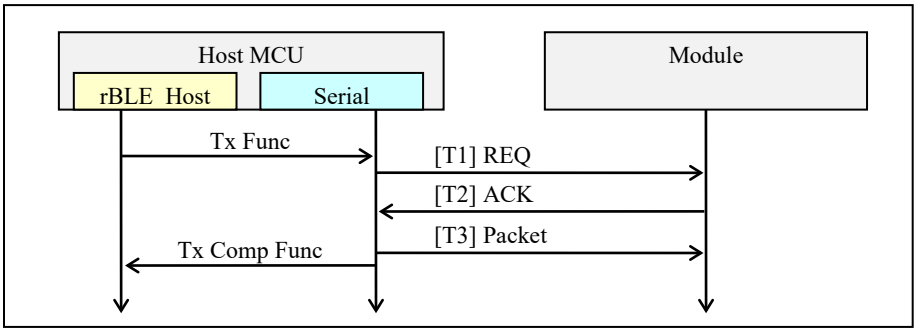


Figure 6-6 Transmit Sequence (Host MCU)

(2) Receive Operation (Host MCU)

The reception sequence including the rBLE_Host and a serial communications driver's function calls is shown. When one RSCIP packet is received, the rBLE_Host calls the reception function two or more times because the RSCIP packet is variable-length.

[When beginning to receive it] The RBLE_Host calls the reception function. As a result, a serial communications driver begins a reception operation of the RSCIP packet, and waits for a data reception.

[When the reception ends the packet on the way] A serial communications driver notifies the rBLE_Host the reception completion by calling the reception completion notification function after the reception ends. The RBLE_Host calls the reception function again, and a serial communications driver restarts the reception.

[When the reception of the entire packet ends] A serial communications driver notifies the rBLE_Host the reception completion by calling the reception completion notification function after the reception ends. The RBLE_Host calls the reception function again, and waits for the following the RSCIP packet reception.

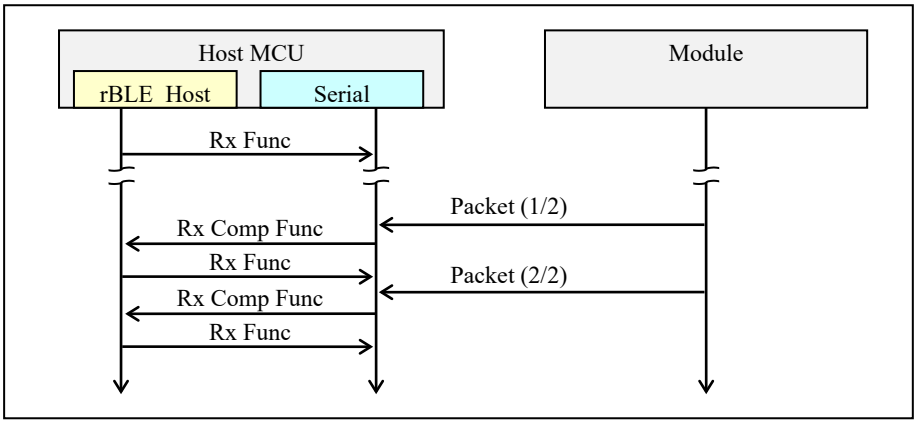


Figure 6-7 Receive Sequence (Host MCU)

6.7 rBLE Command

The firmware adds a vendor specific command of the following using the rBLE_API of BLE software to the vendor specific API. Refer to "9.5 FW Update", "9.6.1 Read Firmware Version", "9.6.2 Set Baud Rate" and "9.6.3 Software Reset" about how to use the each commands in this document.

- Vendor specific commands
  - RBLE_VS_Set_Params API Command
    - Set Baud Rate
    - Software Reset
    - Enter FW Update mode
  - RBLE_VS_Flash_Access API Command
    - Read Firmware Version

6.7.1 RBLE_VS_Set_Params API Command

Using 0x80 or later param_id in RBLE_VS_Set_Params, "Set Baud Rate", "Enter FW Update mode" and "Software Reset" adds the functionality. The API specification below are shown below.

[Note] Please note the following.

- After entering to the FW Update mode, it does not return to normal mode until the Update is complete.

RBLE_STATUS RBLE_VS_Set_Params (uint8_t param_id, uint8_t param_len, uint8_t *param_data)

This function sets the parameters in BLE MCU.

The result is reported by using the parameter setting completion event RBLE_VS_EVENT_SET_PARAMS_COMP

・The "Set baud rate" save a baud rate number to dataflash. When start the module, it read the baud rate number from the dataflash and initialize the serial driver. After performing the "Set baud rate", it execute the reset. Baud rate will take effect after reset.

・When it execute the "Enter FW Update mode", migrate to FW Update mode after one second.

・The "Software Reset" execute a internal reset by illegal instruction. When it execute the "Software Reset", internal reset occurs after one second.

Parameters:

param_id

Setting parameter ID

Setting parameter ID	Number	Description
RBLE_VS_PARAM_UART_BAUD_ID	0x80	Set baud rate
RBLE_VS_PARAM_FW_UPDATE	0xD9	Enter FW Update mode
RBLE_VS_PARAM_SOFT_RESET	0xFF	Software Reset

param_len

Length of setting parameter

Setting parameter ID	Parameter length
RBLE_VS_PARAM_UART_BAUD_ID	1
RBLE_VS_PARAM_FW_UPDATE	does not use
RBLE_VS_PARAM_SOFT_RESET	does not use

**param_data*

Pointer to the parameter data(the least significant byte first, left justified)

Setting parameter ID	Baud rate number
RBLE_VS_PARAM_UART_BAUD_ID	0: 4800 bps 1: 9600 bps 2: 19200 bps 3: 38400 bps 4: 57600 bps 5: 115200 bps 6: 250000 bps
RBLE_VS_PARAM_FW_UPDATE	does not use
RBLE_VS_PARAM_SOFT_RESET	does not use

Return:

RBLE_OK	Success
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.
RBLE_UNSUPPORTED	Unsupported
RBLE_PARAM_ERR	Invalid parameter

When it execute the RBLE_VS_Set_Params, RBLE_VS_EVENT_SET_PARAMS_COMP event occurs and setting results will returned.

RBLE_VS_EVENT_SET_PARAMS_COMP		
This event reports completion of setting up a parameter.		
Parameters:		
	status	Result of setting up a parameter

6.7.2 RBLE_VS_Flash_Access API Command

The "Read firmware version" is being added to RBLE_VS_Flash_Access. The API specification below are shown below.

RBLE_STATUS RBLE_VS_Flash_Access (RBLE_VS_FLASH_ACCESS_PARAM *param)		
This function writes the data to Data Flash or reads data from Data Flash. The result is reported by using the Data Flash access command completion event RBLE_VS_EVENT_FLASH_ACCESS_COMP. * Before calling this function, starts the access to Data Flash by using the RBLE_VS_Flash_Management. In addition, maintain buffer that is specified in the parameter until the data writing or reading is completed.		
Parameters:		
cmd	Data Flash access command RBLE_VS_FLASH_CMD_WRITE: Writes the data RBLE_VS_FLASH_CMD_READ: Reads the data	
id	Data ID (0x01 – 0xFF)	
	Setting parameter ID	Number / Description
	RBLE_VS_PARAM_EEL_ID_MODFWVER	4 / Firmware version
size	Data size (1 – 255 bytes)	
	Setting parameter ID	Data size
	RBLE_VS_PARAM_EEL_ID_MODFWVER	2
**addr*	Pointer to writing or reading buffer	
Return:		
RBLE_OK	Success	
RBLE_STATUS_ERROR	Not executable because the rBLE mode is other than RBLE_MODE_ACTIVE.	

When it execute the RBLE_VS_Flash_Access, RBLE_VS_EVENT_FLASH_ACCESS_COMP event occurs and executing results of data flash access command will returned. Pointer of firmware version storage location will be returned to "*addr" of parameter.

RBLE_VS_EVENT_FLASH_ACCESS_COMP	
This event reports completion of executing the Data Flash access command.	
Parameters:	
status	Result of executing the Data Flash access command
cmd	Execution command
id	Data ID
size	Data size
**addr*	Pointer to data buffer. Firmware version is stored. Format is below. [0] : low version [1] : high version ex) V1.0 : [1]=0x01, [0]=0x00

7. Profiles

7.1 Supported Profiles

Due to the deprecation and withdrawal plan of the profile version by Bluetooth SIG, each profile has been obsoleted because product registration using the profile supported by the Bluetooth Low Energy protocol stack is no longer possible.

For product registration, refer to "Bluetooth LE microcomputer/module Bluetooth qualification acquisition application note" (R01AN3177).

Table 7-1 The profiles adopted by the Bluetooth SIG

Profile	Role
Proximity Profile (Obsolete)	Monitor
	Reporter
Find Me Profile (Obsolete)	Locator
	Target
Heat Rate Profile (Obsolete)	Collector
	Sensor
Time Profile (Obsolete)	Client
	Server
Alert Notification Profile (Obsolete)	Client
	Server
Running Speed and Cadence Profile (Obsolete)	Collector
	Sensor
Health Thermometer Profile (Obsolete)	Collector
	Thermometer
Blood Pressure Profile (Obsolete)	Collector
	Sensor
Glucose Profile (Obsolete)	Collector
	Sensor
Phone Alert Status Profile (Obsolete)	Client
	Server

Firmware supports the following custom profiles. Refer to "9.4 General Purpose Communication" and "9.5 FW Update" in this document.

Table 7-2 Custom profiles

Profile	Role
General Purpose Communication	Initiator
	Acceptor
FW Update	Sender
	Receiver

7.2 Services

The following shows the firmware supported services and UUID.

Table 7-3 UUID of adopted Bluetooth services

Service	UUID (HEX)
Generic Access Service	1800
Immediate Alert Service	1802
Link Loss Service	1803
Tx Power Service	1804
Current Time Service	1805
Reference Time Update Service	1806
Next DST Change Service	1807
Glucose Service	1808
Health Thermometer Service	1809
Device Information Service	180A
Heart Rate Service	180D
Phone Alert Status Service	180E
Blood Pressure Service	1810
Alert Notification Service	1811
Running Speed and Cadence Service	1814

Table 7-4 UUID of custom services

Service	UUID (HEX)
General Purpose Communication	D68C0001-A21B-11E5-8CB8-0002A5D5C51B
FW Update	01010000-0000-0000-0000-000000000080

7.3 GATT database

The GATT data base integrated in a module is shown below.

Table 7-5 GATT database

Attribute Handle	Attribute Type	Attribute Value
0x0001	Primary Service Declaration	0x1800(Generic Access Service)
0x0002	Characteristic Declaration	Properties = 0x0A(RD, WR)
0x0003	0x2A00 (Device Name)	
0x0004	Characteristic Declaration	Properties = 0x0A(RD, WR)
0x0005	0x2A01 (Appearance)	
0x0006	Characteristic Declaration	Properties = 0x02(RD)
0x0007	0x2A04 (Peripheral Preferred Connection Parameters)	
0x000C	FW Update Refer to "Table 7-6 FW Update database"	
0x000D		
0x000E		
0x000F		
0x0010		
0x0011	Primary Service Declaration FW Update	0x1803(Link Loss Service)
0x0012	Characteristic Declaration	Properties = 0x0A(RD, WR)
0x0013	0x2A06 (Alert Level)	
0x0014	Primary Service Declaration	0x1804(Tx Power Service)
0x0015	Characteristic Declaration	Properties = 0x02(RD)
0x0016	0x2A07 (Tx Power Level)	
0x0017	Primary Service Declaration	0x1802(Immediate Alert Service)
0x0018	Characteristic Declaration	Properties = 0x04(WR_NO_RESP)
0x0019	0x2A06 (Alert Level)	
0x001A	Primary Service Declaration	0x1809(Health Thermometer Service)
0x001B	Characteristic Declaration	Properties = 0x20(IND)
0x001C	0x2A1C (Temperature Measurement)	
0x001D	0x2902 (Client Characteristic Configuration)	
0x001E	Characteristic Declaration	Properties = 0x02(RD)
0x001F	0x2A1D (Temperature Type)	
0x0020	Characteristic Declaration	Properties = 0x10(NTF)
0x0021	0x2A1E (Intermediate Temperature)	
0x0022	0x2902 (Client Characteristic Configuration)	
0x0023	Characteristic Declaration	Properties = 0x2A(RD, WR, IND)
0x0024	0x2A21 (Measurement Interval)	
0x0025	0x2902 (Client Characteristic Configuration)	
0x0026	0x2906 (Valid Range)	
0x0027	Primary Service Declaration	0x1810(Blood Pressure Service)
0x0028	Characteristic Declaration	Properties = 0x20(IND)
0x0029	0x2A35 (Blood Pressure Measurement)	
0x002A	0x2902 (Client Characteristic Configuration)	
0x002B	Characteristic Declaration	Properties = 0x10(NTF)
0x002C	0x2A36 (Intermediate Cuff Pressure)	
0x002D	0x2902 (Client Characteristic Configuration)	
0x002E	Characteristic Declaration	Properties = 0x02(RD)
0x002F	0x2A49 (Blood Pressure Feature)	
0x0030	Primary Service Declaration	0x180A(Device Information Service)
0x0031	Characteristic Declaration	Properties = 0x02(RD)
0x0032	0x2A23 (System ID)	
0x0033	Characteristic Declaration	Properties = 0x02(RD)

0x0034	0x2A24 (Model Number String)	
0x0035	Characteristic Declaration	Properties = 0x02(RD)
0x0036	0x2A25 (Serial Number String)	
0x0037	Characteristic Declaration	Properties = 0x02(RD)
0x0038	0x2A26 (Firmware Revision String)	
0x0039	Characteristic Declaration	Properties = 0x02(RD)
0x003A	0x2A27 (Hardware Revision String)	
0x003B	Characteristic Declaration	Properties = 0x02(RD)
0x003C	0x2A28 (Software Revision String)	
0x003D	Characteristic Declaration	Properties = 0x02(RD)
0x003E	0x2A29 (Manufacturer Name String)	
0x003F	Characteristic Declaration	Properties = 0x02(RD)
0x0040	0x2A2A (IEEE 11073-20601 Regulatory Certification Data List)	
0x0041	Primary Service Declaration	0x180D(Heart Rate Service)
0x0042	Characteristic Declaration	Properties = 0x10(NTF)
0x0043	0x2A37 (Heart Rate Measurement)	
0x0044	0x2902 (Client Characteristic Configuration)	
0x0045	Characteristic Declaration	Properties = 0x02(RD)
0x0046	0x2A38 (Body Sensor Location)	
0x0047	Characteristic Declaration	Properties = 0x08(WR)
0x0048	0x2A39 (Heart Rate Control Point)	
0x0049	Primary Service Declaration	0x1808(Glucose Service)
0x004A	Characteristic Declaration	Properties = 0x10(NTF)
0x004B	0x2A18 (Glucose Measurement)	
0x004C	0x2902 (Client Characteristic Configuration)	
0x004D	Characteristic Declaration	Properties = 0x10(NTF)
0x004E	0x2A34 (Glucose Measurement Context)	
0x004F	0x2902 (Client Characteristic Configuration)	
0x0050	Characteristic Declaration	Properties = 0x02(RD)
0x0051	0x2A51 (Glucose Feature)	
0x0052	Characteristic Declaration	Properties = 0x28(WR, IND)
0x0053	0x2A52 (Record Access Control Point)	
0x0054	0x2902 (Client Characteristic Configuration)	
0x0055	Primary Service Declaration	0x1805(Current Time Service)
0x0056	Characteristic Declaration	Properties = 0x12(RD, NTF)
0x0057	0x2A2B (Current Time)	
0x0058	0x2902 (Client Characteristic Configuration)	
0x0059	Characteristic Declaration	Properties = 0x02(RD)
0x005A	0x2A0F (Local Time Information)	
0x005B	Characteristic Declaration	Properties = 0x02(RD)
0x005C	0x2A14 (Reference Time Information)	
0x005D	Primary Service Declaration	0x1807(Next DST Change Service)
0x005E	Characteristic Declaration	Properties = 0x02(RD)
0x005F	0x2A11 (Time with DST)	
0x0060	Primary Service Declaration	0x1806(Reference Time Update Service)
0x0061	Characteristic Declaration	Properties = 0x04(WR_NO_RESP)
0x0062	0x2A16 (Time Update Control Point)	
0x0063	Characteristic Declaration	Properties = 0x02(RD)
0x0064	0x2A17 (Time Update State)	
0x0065	Primary Service Declaration	0x1811(Alert Notification Service)
0x0066	Characteristic Declaration	Properties = 0x02(RD)
0x0067	0x2A47 (Supported New Alert Category)	
0x0068	Characteristic Declaration	Properties = 0x10(NTF)
0x0069	0x2A46 (New Alert)	
0x006A	0x2902 (Client Characteristic Configuration)	

0x006B	Characteristic Declaration	Properties = 0x02(RD)
0x006C	0x2A48 (Supported Unread Alert Category)	
0x006D	Characteristic Declaration	Properties = 0x10(NTF)
0x006E	0x2A45 (Unread Alert Status)	
0x006F	0x2902 (Client Characteristic Configuration)	
0x0070	Characteristic Declaration	Properties = 0x08(WR)
0x0071	0x2A44 (Alert Notification Control Point)	
0x0072	Primary Service Declaration	0x180E(Phone Alert Status Service)
0x0073	Characteristic Declaration	Properties = 0x12(RD, NTF)
0x0074	0x2A3F (Alert Status)	
0x0075	0x2902 (Client Characteristic Configuration)	
0x0076	Characteristic Declaration	Properties = 0x12(RD, NTF)
0x0077	0x2A41 (Ringer Setting)	
0x0078	0x2902 (Client Characteristic Configuration)	
0x0079	Characteristic Declaration	Properties = 0x04(WR_NO_RESP)
0x007A	0x2A40 (Ringer Control Point)	
0x007B	Primary Service Declaration	0x1814(Running Speed and Cadence Service)
0x007C	Characteristic Declaration	Properties = 0x10(NTF)
0x007D	0x2A53 (RSC Measurement)	
0x007E	0x2902 (Client Characteristic Configuration)	
0x007F	Characteristic Declaration	Properties = 0x02(RD)
0x0080	0x2A54 (RSC Feature)	
0x0081	Characteristic Declaration	Properties = 0x02(RD)
0x0082	0x2A5D (Sensor Location)	
0x0083	Characteristic Declaration	Properties = 0x28(WR, IND)
0x0084	0x2A55 (SC Control Point)	
0x0085	0x2902 (Client Characteristic Configuration)	
0x0086	General purpose communication Refer to "Table 7-7 General purpose communication database"	
0x0087		
0x0088		
0x0089		
0x008A		
0x008B		

(1) Adopted profiles database specification

About adopted profile database specification, refer to following of "Bluetooth Low Energy protocol stack user's manual" (R01UW0095). And about profile specification of Bluetooth SIG, refer to profile specification of Bluetooth SIG. ("9.7 Referenced Documents")

- 7.2 Generic Access Profile
- 7.5 Find Me Profile
- 7.6 Proximity Profile
- 7.7 Health Thermometer Profile
- 7.8 Blood Pressure Profile
- 7.11 Heat Rate Profile
- 7.14 Glucose Profile
- 7.15 Time Profile
- 7.16 Running Speed and Cadence Profile
- 7.17 Alert Notification Profile
- 7.18 Phone Alert Status Profile

(2) Custom profile database specification

The GATT database specification of the custom profile integrated in a module is shown below.

Table 7-6 FW Update database

Attribute Handle	Attribute Type	Attribute Value
0x000C	Primary Service Declaration (0x2800)	UUID: 01010000-0000-0000-0000-000000000080
0x000D	Characteristic Declaration (0x2803)	Property: Write(0x08) Type: Characteristic Declaration UUID: 02010000-0000-0000-0000-000000000080
0x000E	Value	Control Data Cmd: 0 — Data transmission start Params: Current Block Num / Size Cmd: 1 — Data transmission completion Params: none Cmd: 2 — Data write confirmation Params: none Cmd: 3 — Data transmission completion(all data) Params: none
0x000F	Characteristic Declaration (0x2803)	Property: Write Without Response(0x04) Type: Characteristic Declaration UUID: 03010000-0000-0000-0000-000000000080
0x0010	Value	Update Data Update Data: 1-19 byte Check Sum: 1 byte

Table 7-7 General purpose communication database

Attribute Handle	Attribute Type	Attribute Value
0x0086	Primary Service Declaration (0x2800)	UUID: 0xD68C0001-A21B-11E5-8CB8-0002A5D5C51B
0x0087	Characteristic Declaration (0x2803)	Property: Indicate(0x20) Type: Characteristic Declaration UUID: 0xD68C0002-A21B-11E5-8CB8-0002A5D5C51B
0x0088	Indication Value	By setting characters to this characteristic and send Indication, the characters are sent from the server to the client. Max 20 characters.
0x0089	Client Characteristic Configuration Descriptor (0x2902)	Used for Indication enable / disable of the server from the client. 0x0000: Indications disabled 0x0002: Indications enabled
0x008A	Characteristic Declaration (0x2803)	Property: Write(0x08) Type: Characteristic Declaration UUID: 0xD68C0003-A21B-11E5-8CB8-0002A5D5C51B
0x008B	Write Value	By writing characters to this characteristic with “Write Request”, the characters are sent from the client to the server. Max 20 characters.

8. Code flash memory rewriting

A shipping check flag is written in block 254 of a cord flash memory. And Public Bluetooth Device Address is written in block 255. When rewriting a program of a module, please be careful so as not to erase block 254 and block 255. The following shows the setting of Renesas Flash Programmer (RFP).

(1) Renesas Flash Programmer V3

Select the [Operation Setting] tabbed page. Then select the "Erase Selected Blocks" in the [Erase Option].

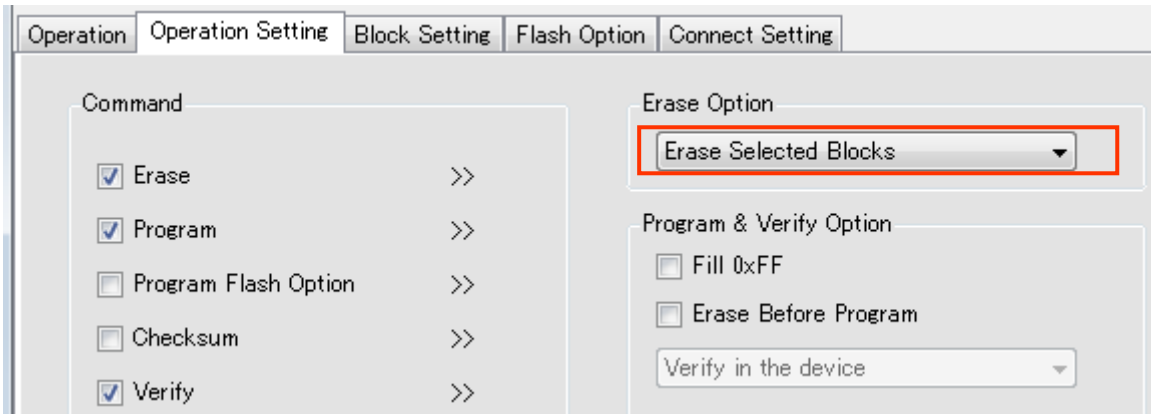


Figure 8-1 Setting of RFP V3 (1)

Select the [Block Setting] tabbed page. Then deselect the [Block 254] checkbox and the [Block 255] checkbox of the [Code Flash 1].

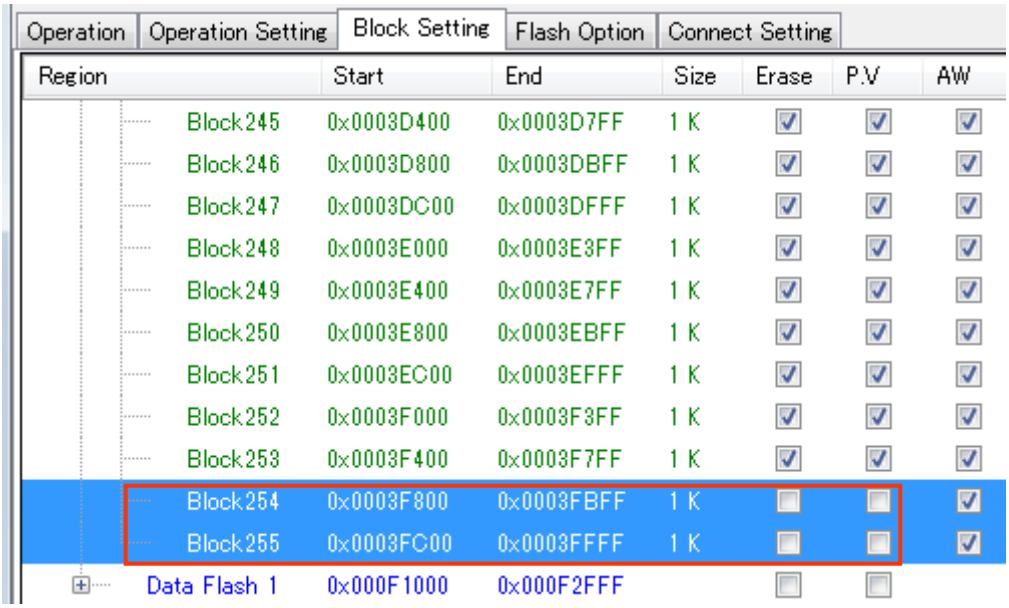


Figure 8-2 Setting of RFP V3 (2)

(2) Renesas Flash Programmer V2

When [Microcontroller] -> [Set Project] is selected from the menu bar, the [Project Settings] dialog is will appear. Select the [Other Settings] tabbed. Then select "Block(Code Flash)" for the [Target] - [Operation mode] and select "253" for the [End Block number of Code Flash].

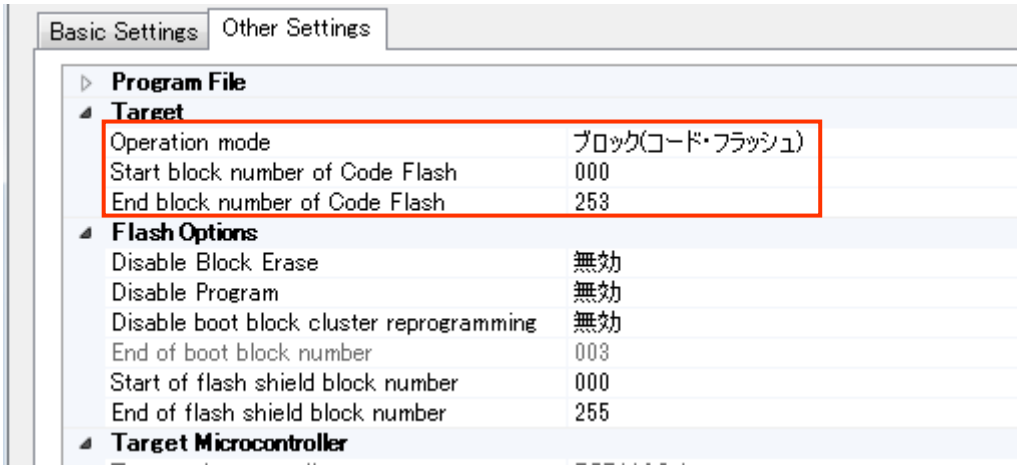


Figure 8-3 Setting of RFP V2

9. Appendix

It's explained below about the symbol used at this chapter.

- "␣" : It represents a space (blank) to separate the command input or the command-line options.

  Example: rBLE_sample.exe␣COM11␣115200

9.1 PC Tools

This section describes PC tools. Two tools are being prepared in order to control a module from a PC instead of Host MCU. Also refer to following application note.

- GUI Tool : GUI Tool Application Note (R01AN2469)
  Download a GUI tool from a web site of Renesas
  (https://www.renesas.com/document/scd/bluetooth-low-energy-protocol-stack-gui-tool)

- BLE_sample.exe : Sample Program Application Note (R01AN1375)
  Use the executable file that is included in the archive file.

9.1.1 GUI Tool

GUI tool is a GUI application for evaluating the API of GAP, SM, VS, GATT and five profiles (FMP, PXP, ANP, HRP and TIP) which is provided by BLE software.

This section describes how to configure the GUI tool when you start to communicate with the module. Refer to the "GUI tool" application notes(R01AN2469) about installation and detailed information on how to use the GUI tools.

[Note] The GUI tool does not support the General Purpose Communication and the FW Update.

When the GUI Tool is executed, the following serial setting dialog box will appear. Selection of the COM port, selection of the baud rate, select the UART 2-wire with Branch Connection, then press the OK button. The main dialog box and the log dialog box will appear.

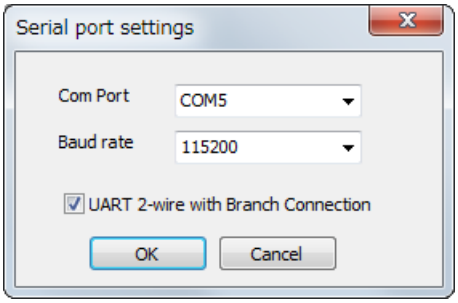


Figure 9-1 Serial Setting Dialog

If the main dialog and the log dialog shown in the figure below does not display anything, communication with the module has not been performed correctly. Please review the settings in the Serial port settings dialog.

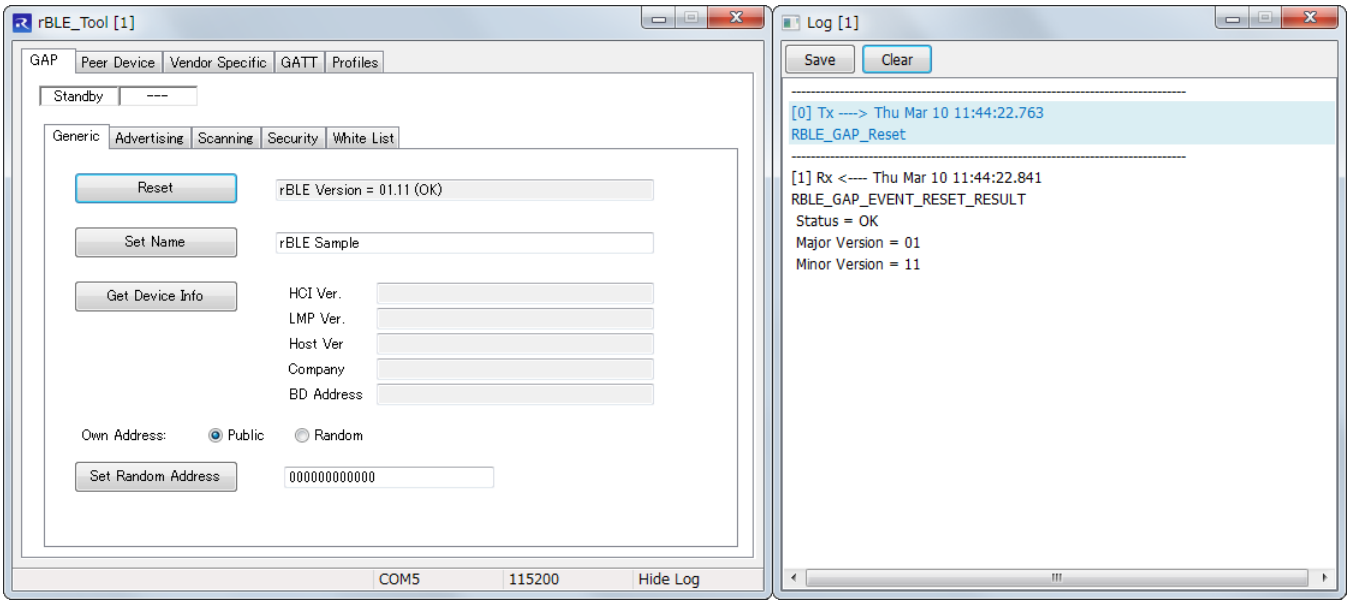


Figure 9-2 Main Dialog/Log Dialog

9.1.2 rBLE_sample

rBLE_sample is a command prompt application for evaluating the API of GAP, SM, VS, GATT and ten profiles (FMP, PXP, ANP, HRP, TIP, HTP, BLP, GLP, PASP, RSCP) which is provided by BLE software. In addition, it can check the operation of the General Purpose Communication.

This section describes how to execute the rBLE_sample and how to connection. About API, refer to the "API Reference Manual: Basics"(R01UW0088) and API Reference Manuals of each profile. About how to use the rBLE_sample, refer to the "Sample Program" application note(R01AN1375E).

9.1.2.1 Options

rBLE_sample will start by executing the EXE file "rBLE_Sample.exe". Execution of "rBLE_Sample.exe" require command line option. When starting, create a batch file. Describes below the options that are required to start.

Command Line Option:

rBLE_sample.exe␣[COM port number]␣[Baud Rate]␣[BD Address]␣[UART 2-wire Branch Connection]

Example:

rBLE_sample.exe␣COM5␣115200␣00:1B:DC:04:7A:34␣–div2wire

Table 9-1 Option of rBLE_sample.exe

Option	Description
COM Port Number	Specify the COM port number in the computer. (e.g., COM1, COM2, …)
Baud Rate	Select from the following baud rate. 4800, 9600, 19200, 38400, 57600, 115200, 250000 (default: 115200)
BD Address of remote device (Public address)	Set the BD address (Bluetooth device address) of the remote device to be connected to. With this address, it is not required to obtain the BD address of remote device using device search, and connection procedure can be started immediately. Use public address as BD address.
UART 2-wire Branch Connection	UART 2-wire with Branch Connection : -div2wire UART 2 wire : none

9.2 BD Address Confirmation

This section describes how to do the communication using the two modules. One is the master device, other one is slave device. Each device uses following setting as an example. Batch file to be used in the running of rBLE_sample requires two of the master device and the slave device. The BD address that has been described as an example, it uses the BD address that is written to the module. The COM port number of the batch files change to the com port number of each device which they connected to a PC. Please prepare an evaluation board equipped with the module.

- Baudrate
  - Baud Rate : 115200 bps

- Batch file Setting for rBLE_sample
  - run_master.bat : rBLE_sample.exe␣COM11␣115200␣22:22:22:22:22:22␣–div2wire
  - run_slave.bat : rBLE_sample.exe␣COM12␣115200␣11:11:11:11:11:11␣–div2wire

Indicates the point to type a command by the arrow icon.

9.2.1 Configuration Diagram

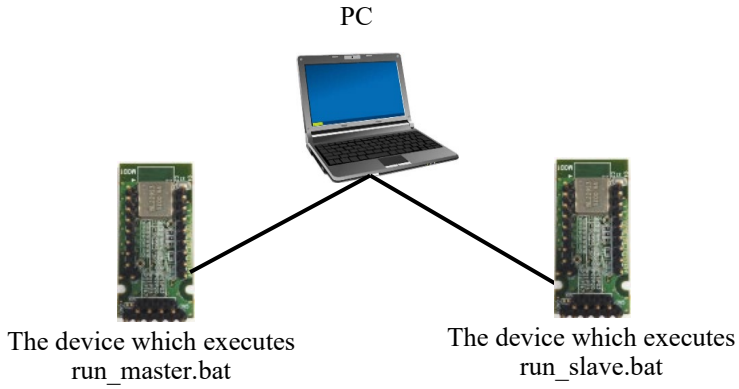
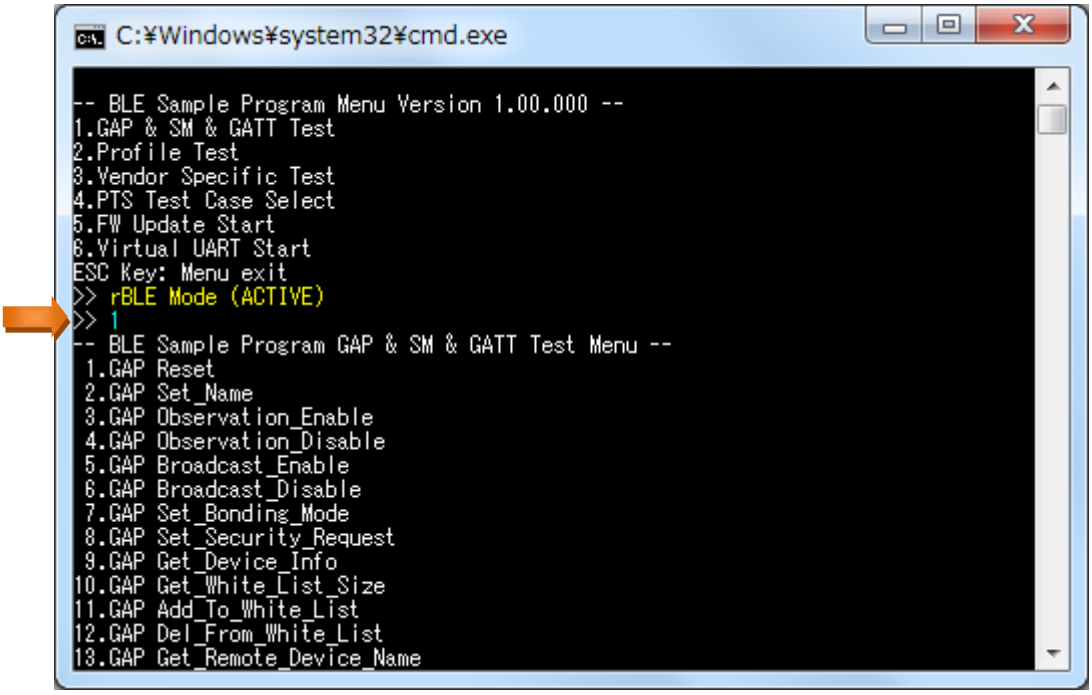


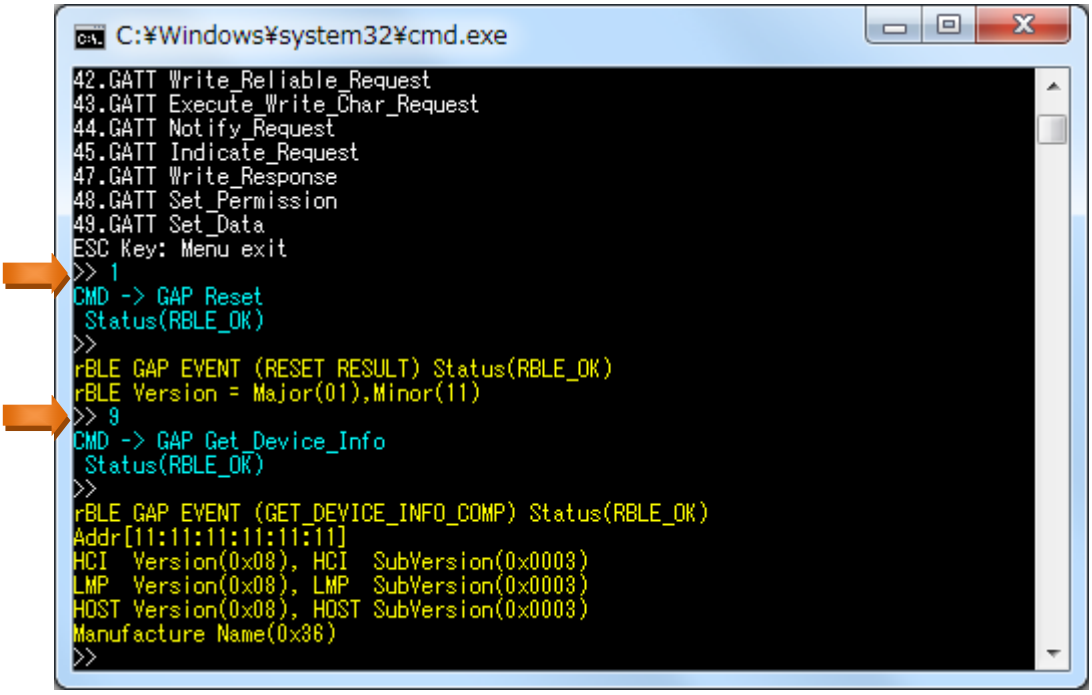
Figure 9-3 BD Address Confirmation Configuration Diagram

9.2.2 Get Device Information

Execute the rBLE_sample by using the run_master.bat. Top menu is displayed. Choose the "1.GAP & SM & GATT Test" from Top menu. Type "1" and press enter key.



Choose the "1.GAP Reset" menu. Type "1" and press enter key. The GAP Reset is executed. Choose the "9.GAP Get_Device_Info" menu. Type "9" and press enter key. The BD address is displayed.



Write to the run_slave.bat the BD address that was confirmed by the run_master.bat. In the similar way, write to the run_master.bat the BD address that was confirmed by the run_slave.bat.

9.3 Master-Slave Communication

This section describes how to do the communication using the two modules. One is the master device, other one is slave device. Each device uses following setting as an example. Batch file to be used in the running of rBLE_sample requires two of the master device and the slave device. The BD address that has been described as an example, it uses the BD address that is written to the module. The COM port number of the batch files change to the com port number of each device which they connected to a PC. Please prepare an evaluation board equipped with the module.

- Device Setting
  - Master BD Address : 11:11:11:11:11:11
  - Slave BD Address : 22:22:22:22:22:22
  - Baud Rate : 115200 bps

- Batch file Setting for rBLE_sample
  - run_master.bat : rBLE_sample.exe␣COM11␣115200␣22:22:22:22:22:22␣–div2wire
  - run_slave.bat : rBLE_sample.exe␣COM12␣115200␣11:11:11:11:11:11␣–div2wire

In the description of the method of operation, the device to be operated is indicated by the following icon.

Indicates the point to type a command by the arrow icon.

9.3.1 Configuration Diagram

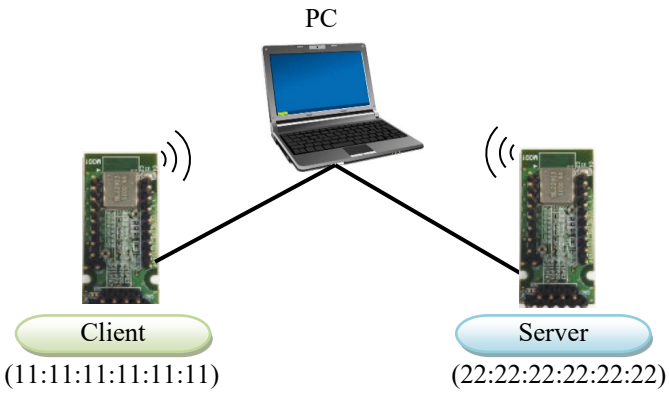
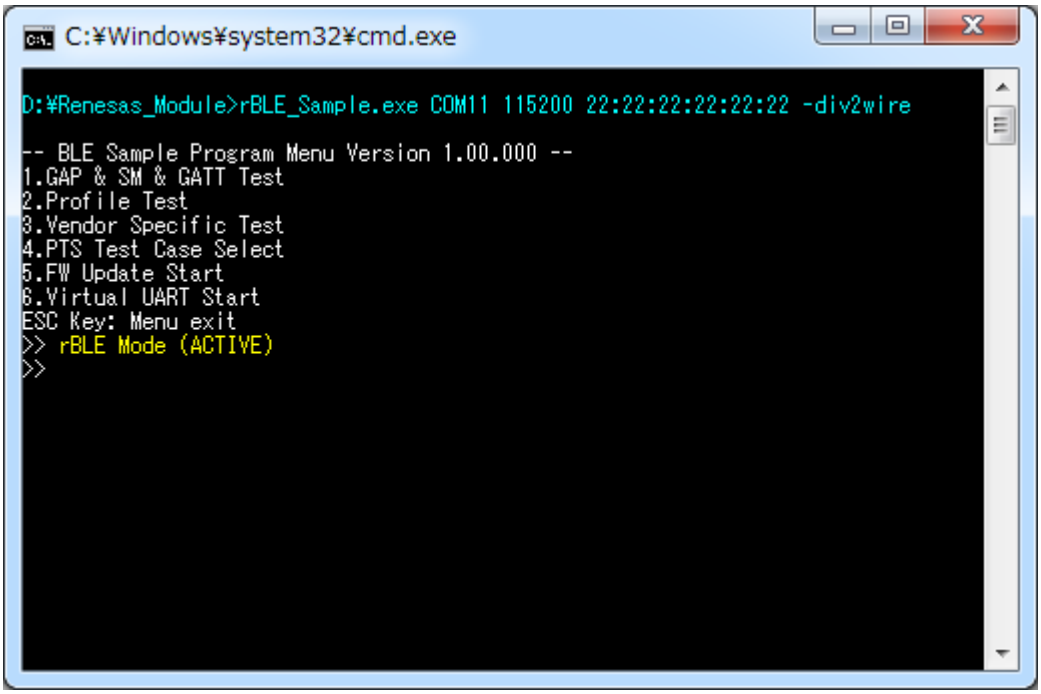


Figure 9-4 Master - Slave Communication Configuration Diagram

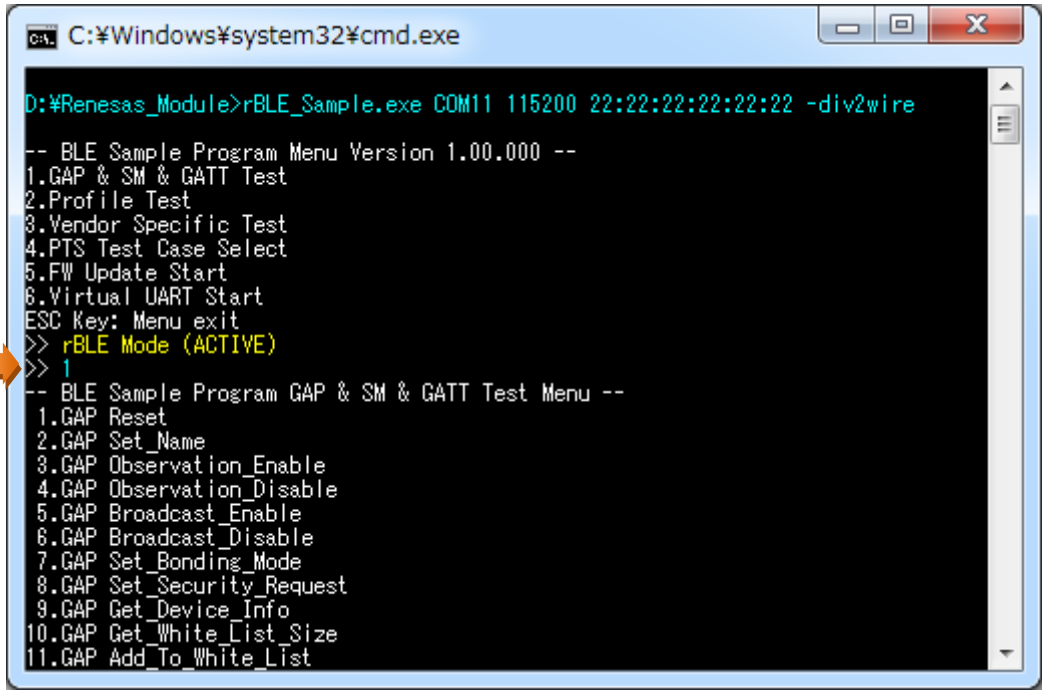
9.3.2 Run and GAP Reset

Master Slave

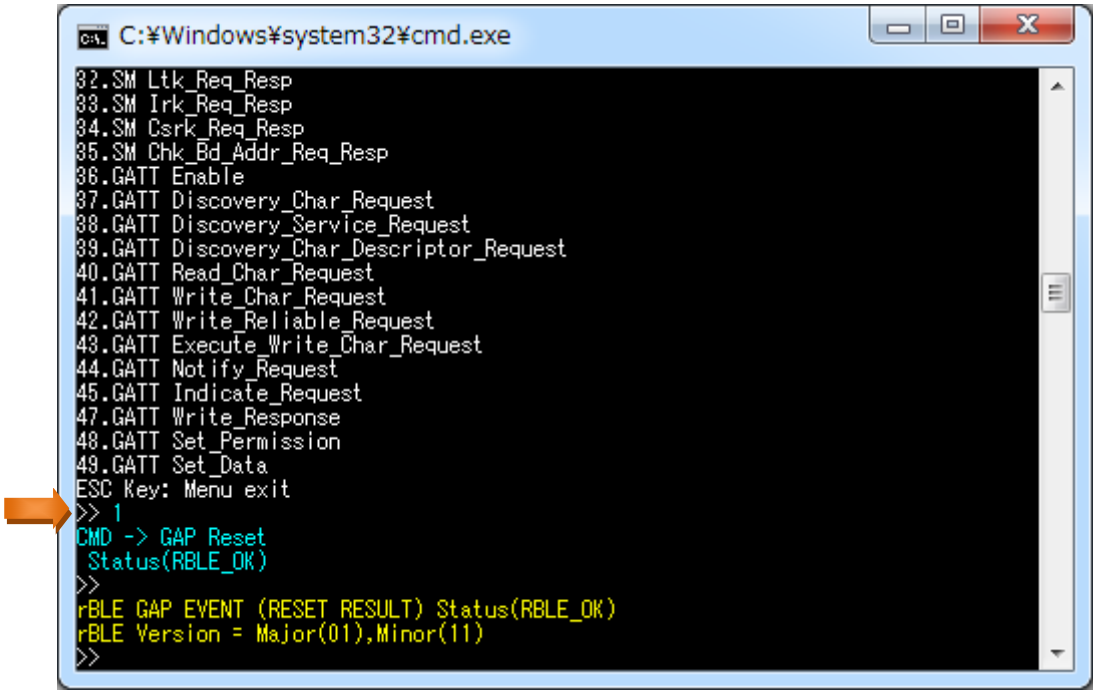
Execute the rBLE_sample by using the run_master.bat and the run_slave.bat. Top menu is displayed.



Choose the "1.GAP & SM & GATT Test" from top menu. Type "1" and press enter key.



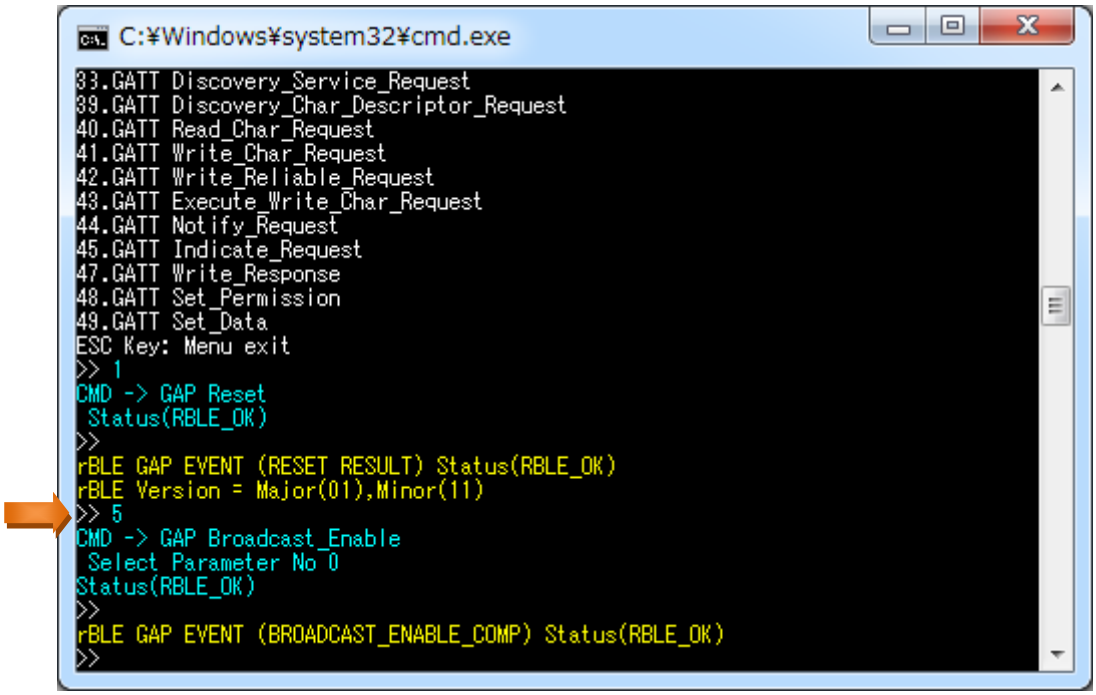
Choose the "1.GAP Reset" menu. Type "1" and press enter key. The GAP Reset is executed.



9.3.3 Advertise

Slave

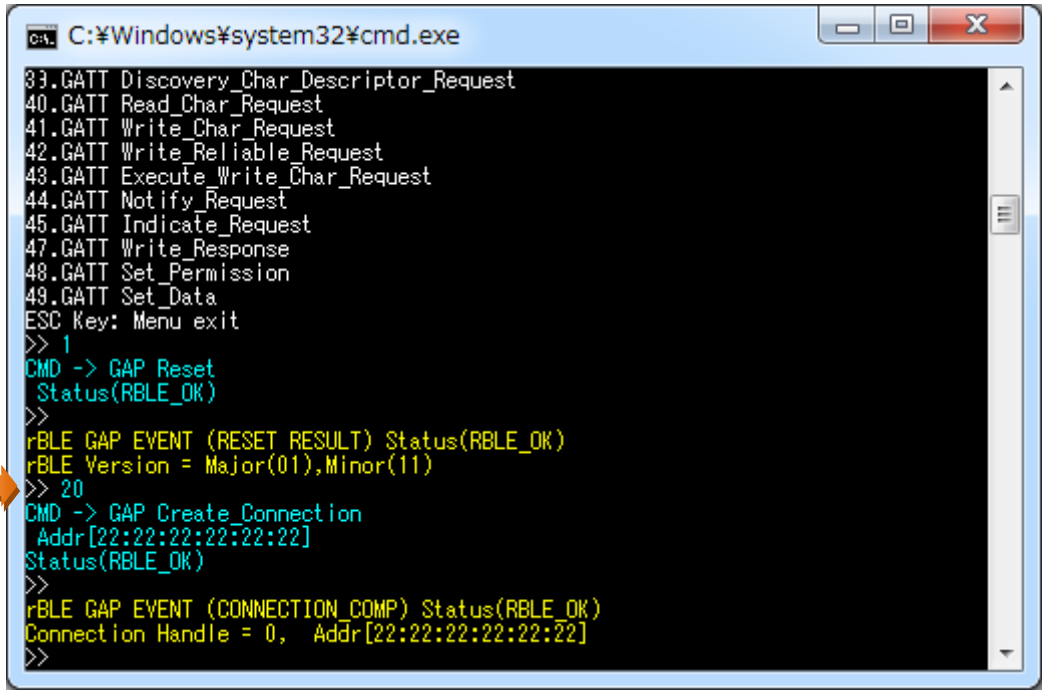
The slave device execute the advertising to wait for the connection from Master device. Choose the "5.GAP Broadcast_Enable" menu. Type "5" and press enter key.



9.3.4 Connect

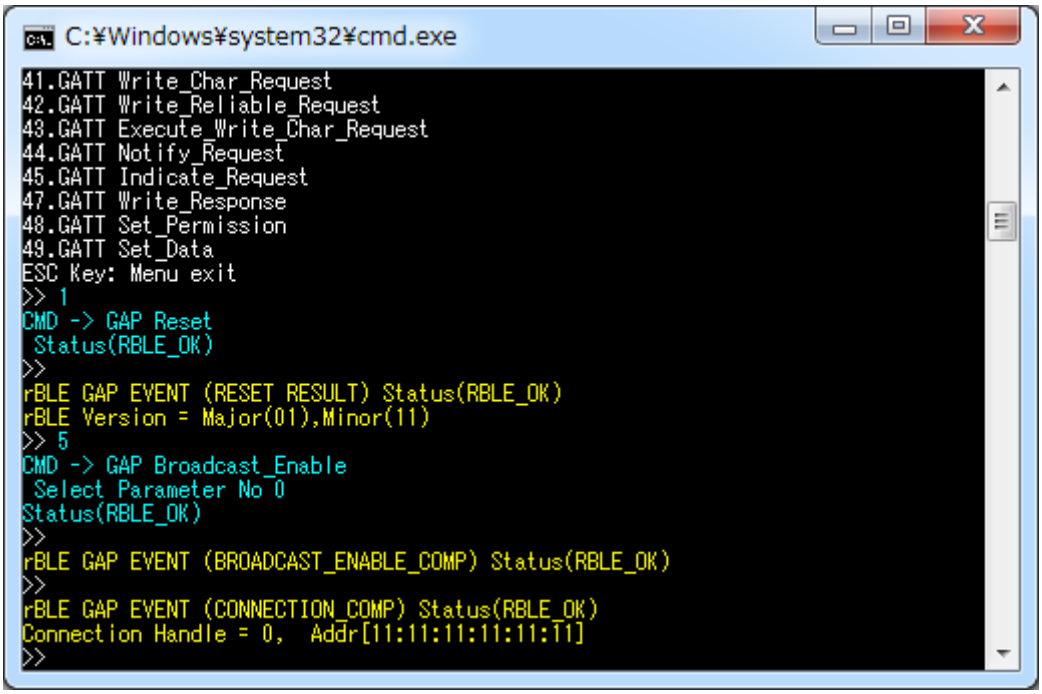
Master

In order to connect the master and the slave, it execute the create connection from the master. Choose "20.GAP Create_Connection" menu. Type "20" and press enter key. When connection with the slave was completed, the connection handle and slave device address are displayed.



Slave

The connection handle and master device address are displayed by slave.



9.4 General Purpose Communication

This section describes how to do the general purpose communication using the two modules. One is the local device, other one is remote device. The general purpose communication can be transmitted and received string in both directions between the connected devices.

AT command is used for control, such as connection or disconnection of BLE. The device which has initiated a connection is client. The device which has been connected is server. Typed characters to rBLE_sample of a client is sent to a server. Then they are displayed on rBLE_sample of a server. Typed characters to rBLE_sample of a server is sent to a client. Then they are displayed on rBLE_sample of a client. This demonstration will not be able to return to the MENU selection from the general purpose communication. Please reset the module to return to the MENU selection.

For each devices setting, refer to "9.3 Master-Slave Communication" of this document. In addition, the general purpose communication is compatible with the BLE Virtual UART. Also see "BLE Virtual UART Application"(R01AN3130).

In the description of the method of operation, the device to be operated is indicated by the following icon.

Client Server

Indicates the point to type a command by the arrow icon.

9.4.1 Configuration Diagram

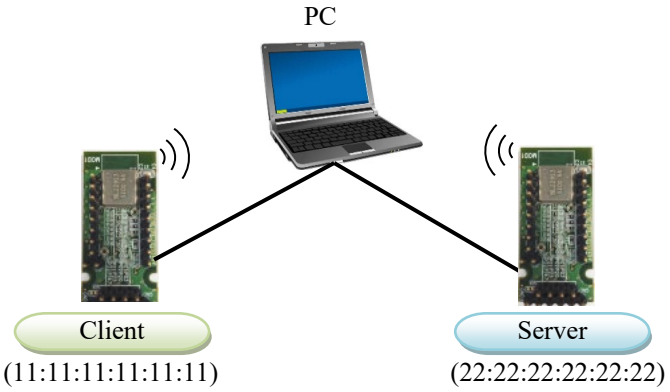


Figure 9-5 General Purpose Communication Configuration Diagram

9.4.2 Run and GAP Reset

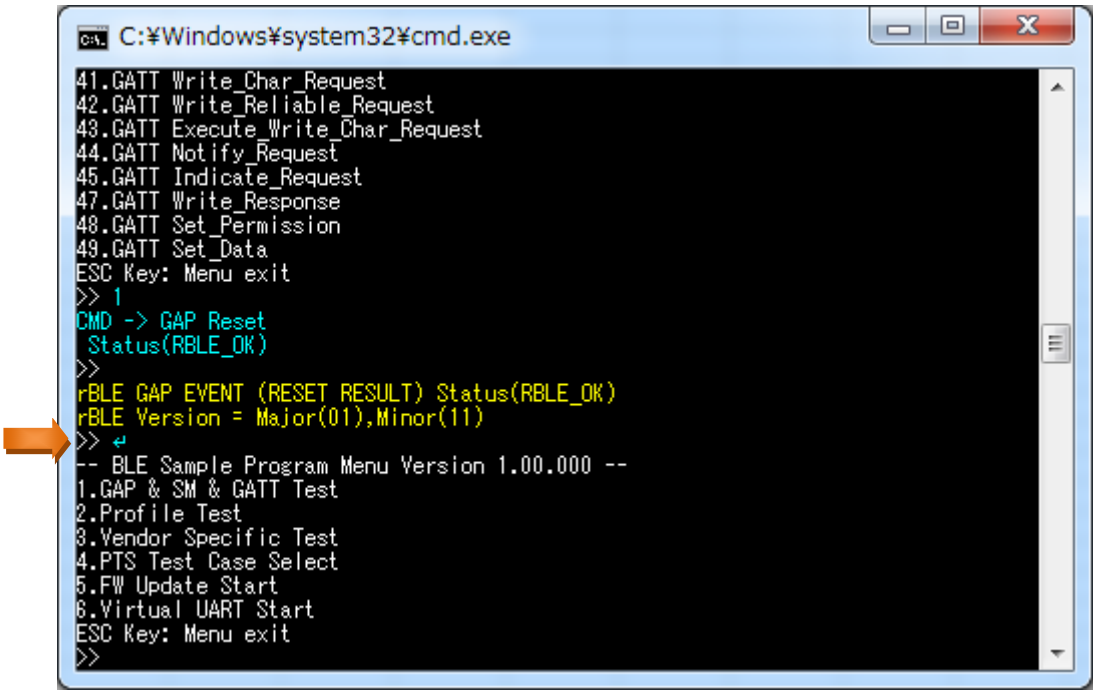
Client Server

Execute the rBLE_sample by using the run_master.bat and the run_slave.bat. For method, refer to "9.3.2 Run and GAP Reset" of this document. And execute the GAP reset.

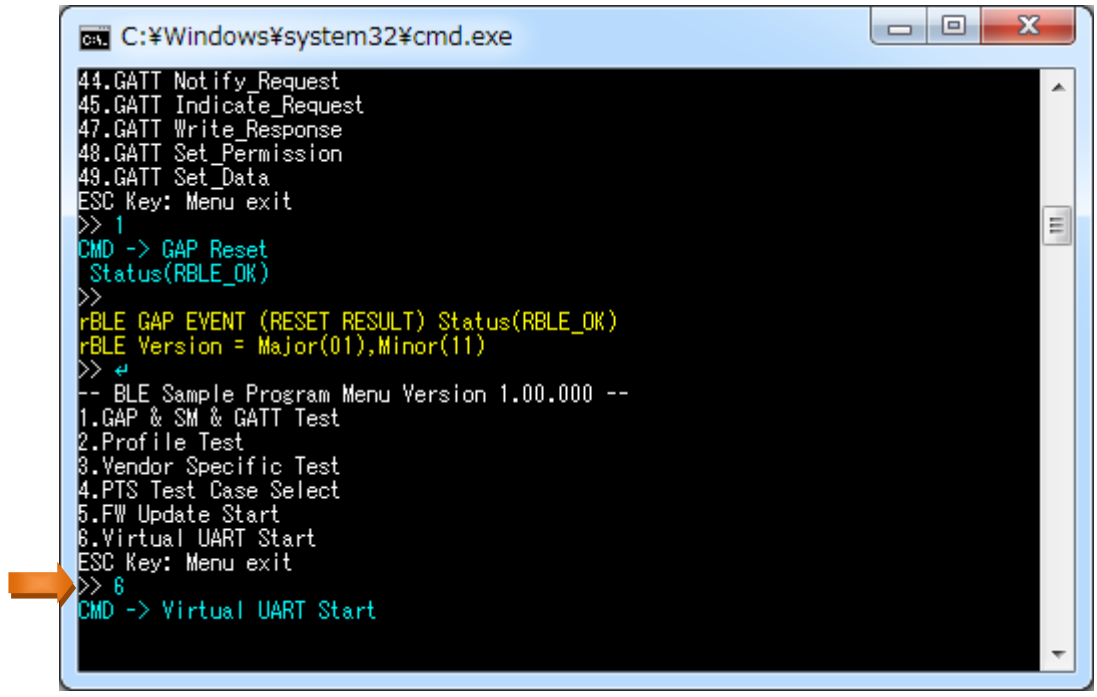
9.4.3 General Purpose Communication Mode

Client Server

Press ESC key for return to the top menu at the client and the server.



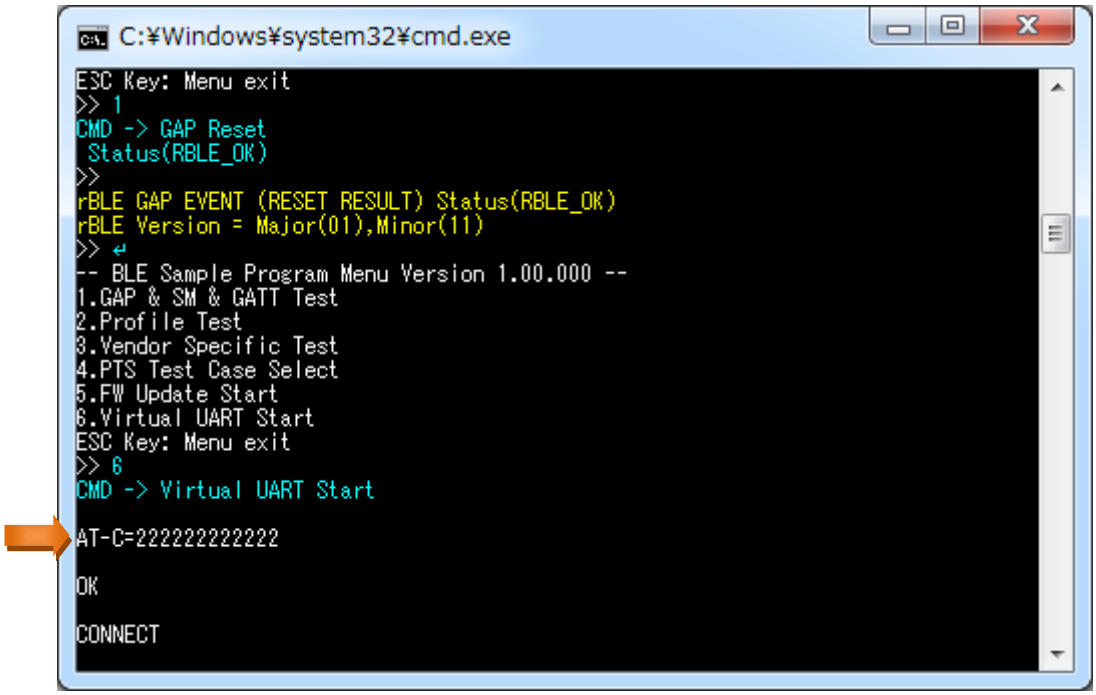
Choose the "6.Virtual UART Start" menu at the client and the server. Type "6" and press enter key. It will enter to the general purpose communication mode.



9.4.4 Connect

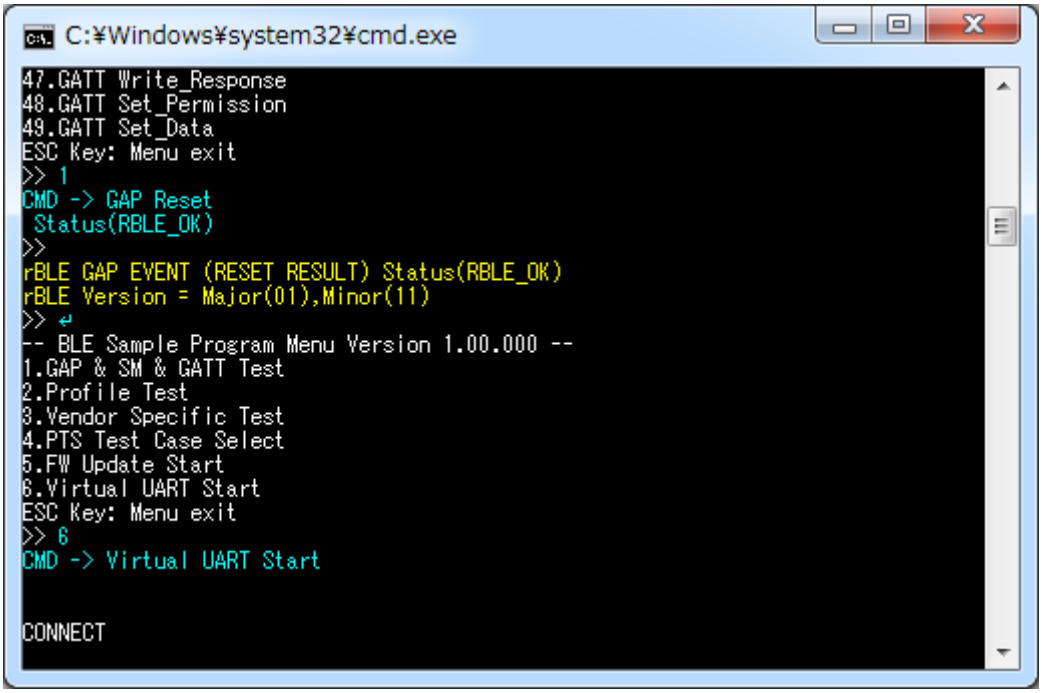
Client

In order to connect the client and the server, it execute the AT-C command from the client. Type "AT-C=222222222222" and press enter key. When connection with the server was completed, "CONNECT" is displayed.



Server

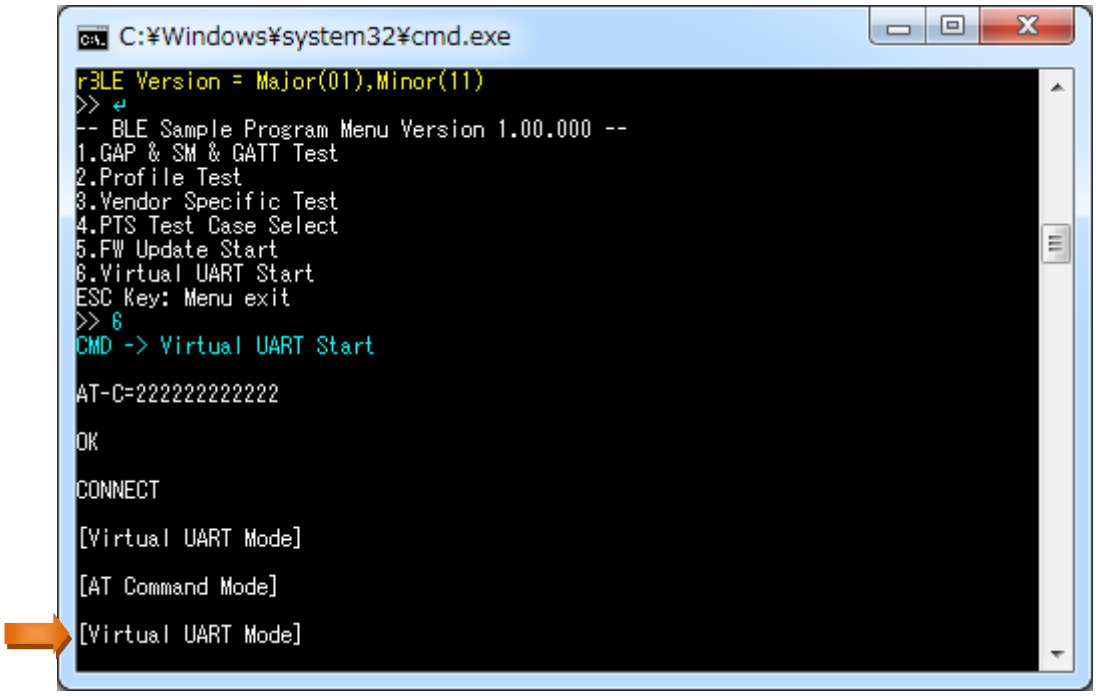
When connection with the client was completed, "CONNECT" is displayed.



9.4.5 Communication

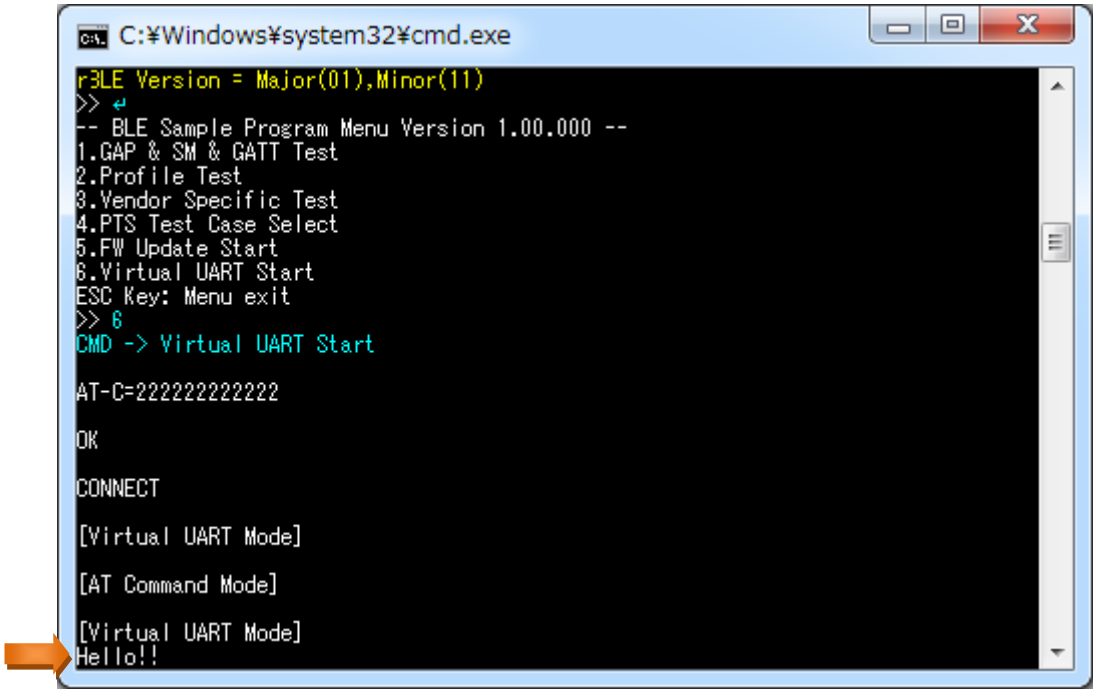
Client Server

The general purpose communication have AT command mode and Virtual UART Press ESC key and the two modes will toggle. Put the client and the server in the Virtual UART mode by pressing ESC key.



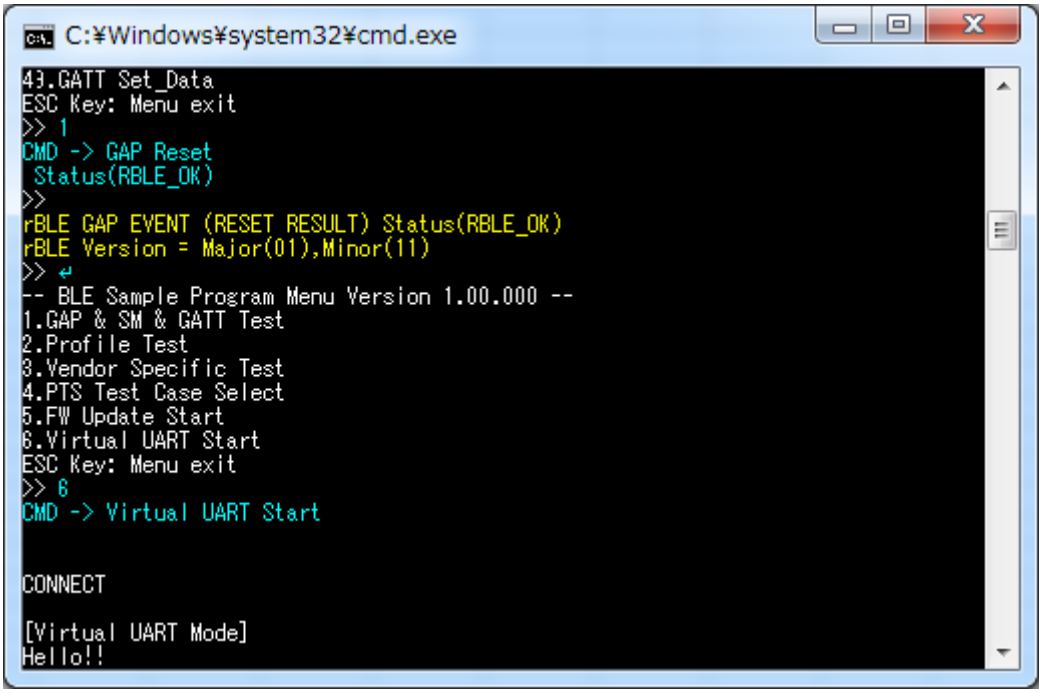
Client

Type character string at the client.



Server

The character string input to the client is displayed to the server. Similarly, it can send character string to the client from the server.



9.5 FW Update

This section describes how to do the FW Update using the two modules. One is the sender device, other one is receiver device. FW Update sends data for update from the sender device and can change the profile of the receiver device. Also refer to "Bluetooth Low Energy Protocol Stack User's Manual chapter 11"(R01UW0095) about the FW Update.

For each devices setting, refer to "9.3 Master-Slave Communication" of this document. In the description of the method of operation, the device to be operated is indicated by the following icon.

Sender Receiver

Indicates the point to type a command by the arrow icon.

9.5.1 Configuration Diagram

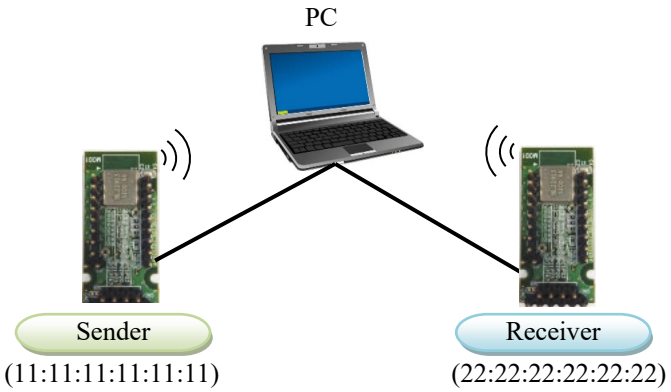


Figure 9-6 FW Update Communication Configuration Diagram

9.5.2 Run and GAP Reset

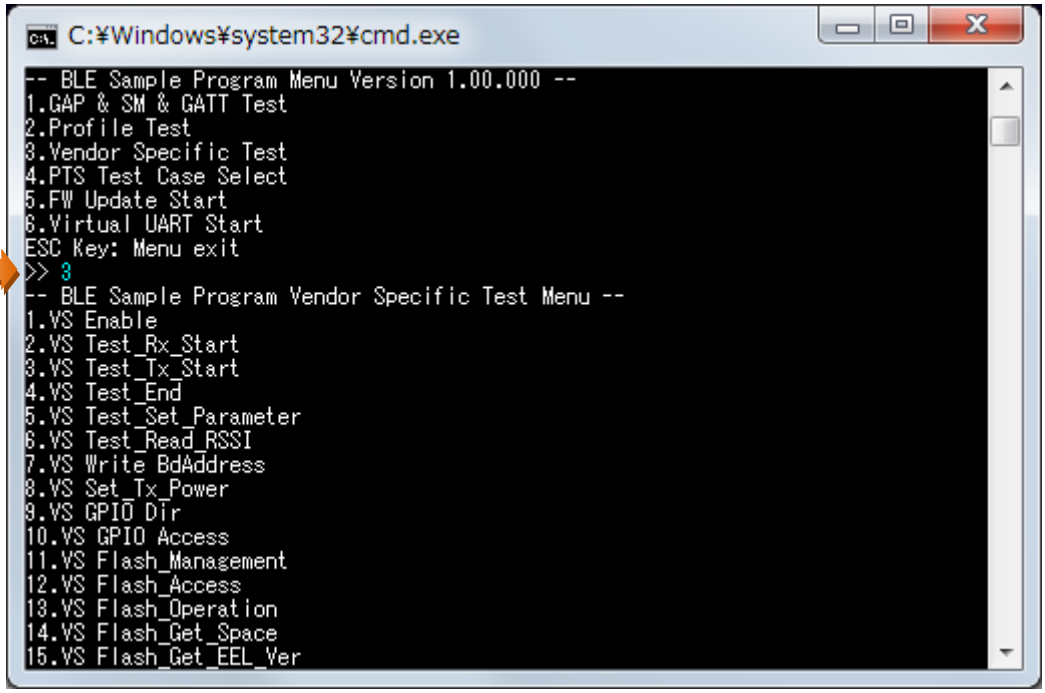
Sender Receiver

Execute the rBLE_sample by using the run_master.bat and the run_slave.bat. For method, refer to "9.3.2 Run and GAP Reset" of this document. And execute the GAP reset.

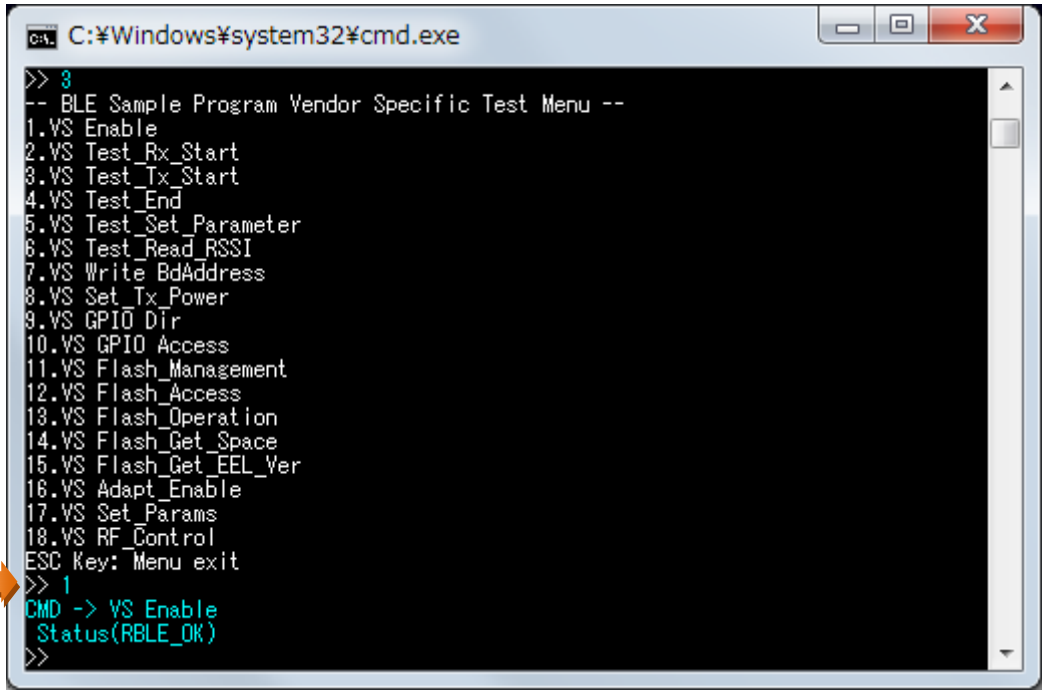
9.5.3 Update Data Reception Preparations - Run Update

Receiver

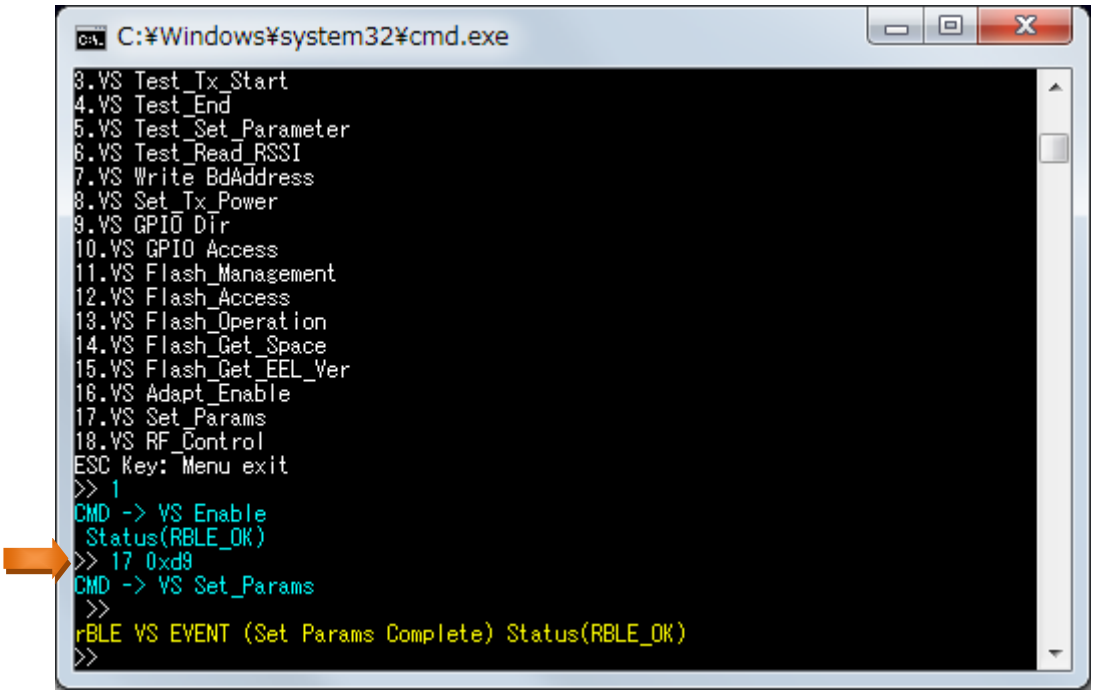
The receiver listens for update data sent from the sender. Press ESC key to return to the top menu. Choose the "3.Vendor Specific Test" menu. Type "3" and press enter key.



Enable the vendor specific function by "1.VS Enable". Type "1" and press enter key.



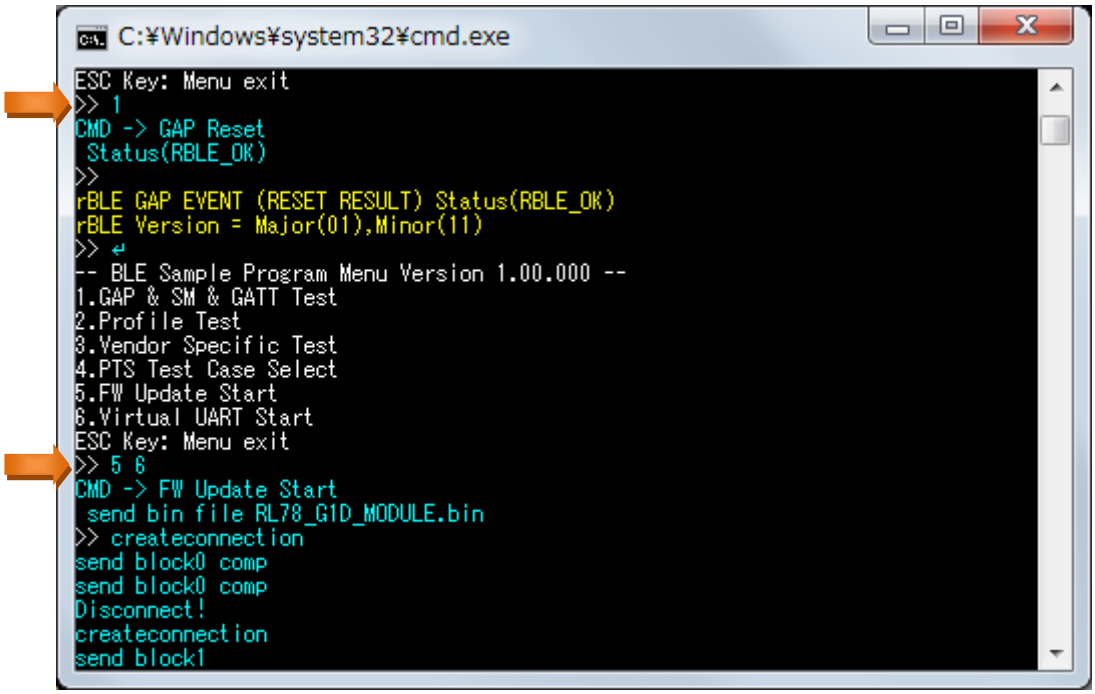
Enter the FW Update mode by "17. VS Set_Params". Type "17␣0xd9" and press enter key. After data transmission completion for update from the sender, execute reset.



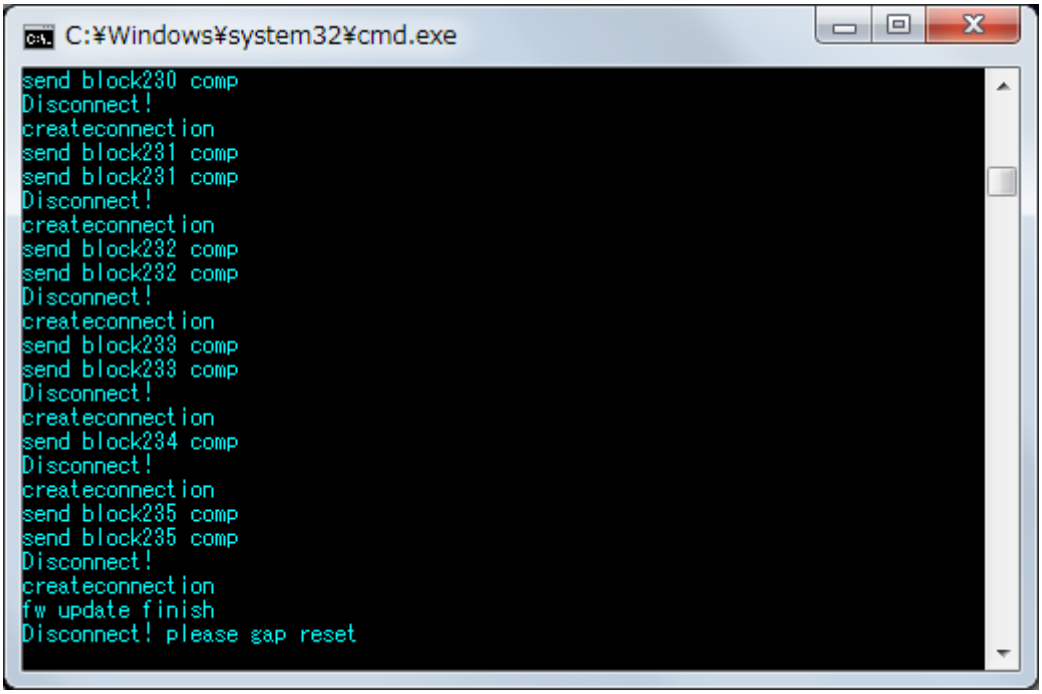
9.5.4 Data Transmission for Update

Sender

Send the data for update from the sender. Press ESC key to return to the top menu. Choose the "5.FW Update Start" menu. Type "5␣6" and press enter key. The transmission of data for Update is started.



When the transmission of data for Update is completed, a message of "fw update finish" is displayed. After displaying the menu by pressing enter key, and run the GAP Reset.



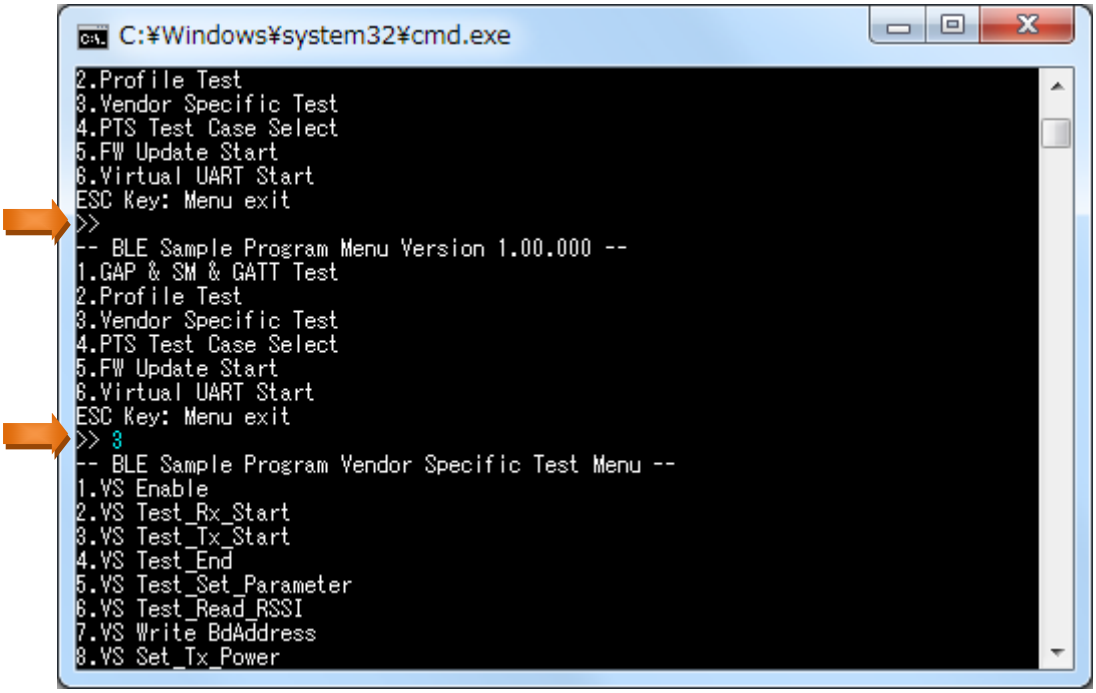
9.6 rBLE Commands

An original command using the rBLE API of the BLE software is added to the firmware as vendor specific. This section describes how to use the command by using the rBLE_sample. Other rBLE_sample commands refer to the "Sample Program Application Note. Chapter 5"(R01AN1375).

- Original Command
- Read firmware version
- Set baud rate
- Software reset

Indicates the point to type a command by the arrow icon.

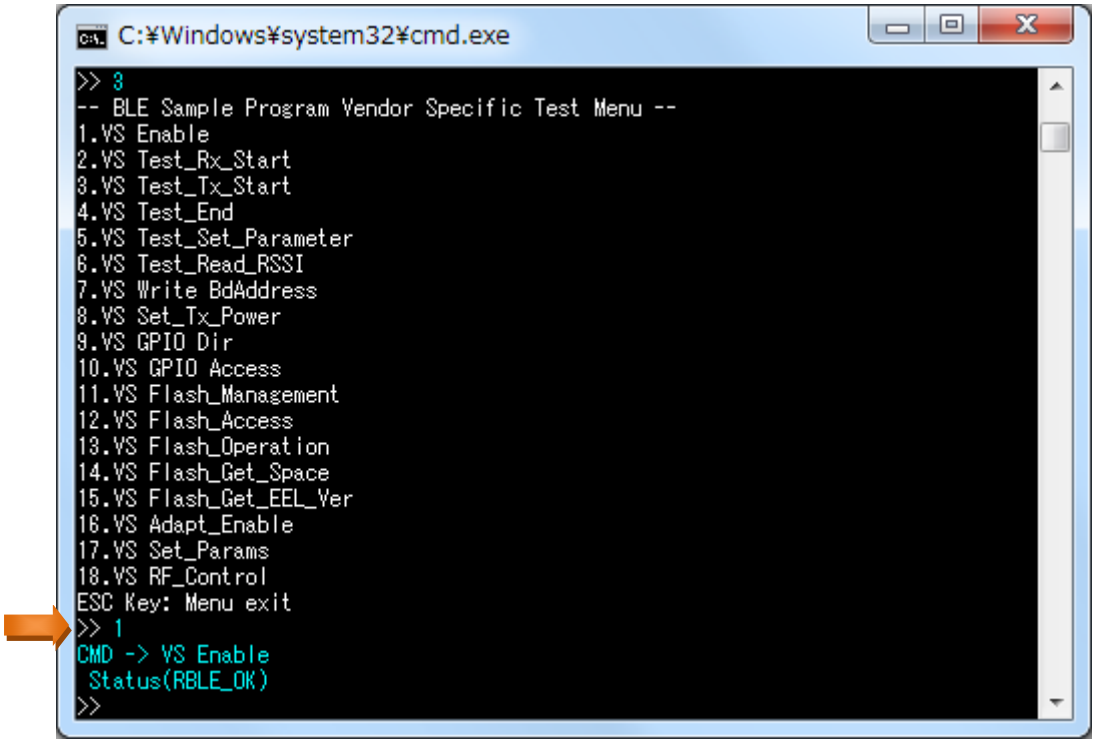
Execute the rBLE_sample as ready to run the command. And execute the GAP reset. About execution method refer to the "9.3.2 Run and GAP Reset". Press ESC key after executing the GAP reset. Next, choose the "Vendor Specific Test" menu. Type "3" and press enter key.



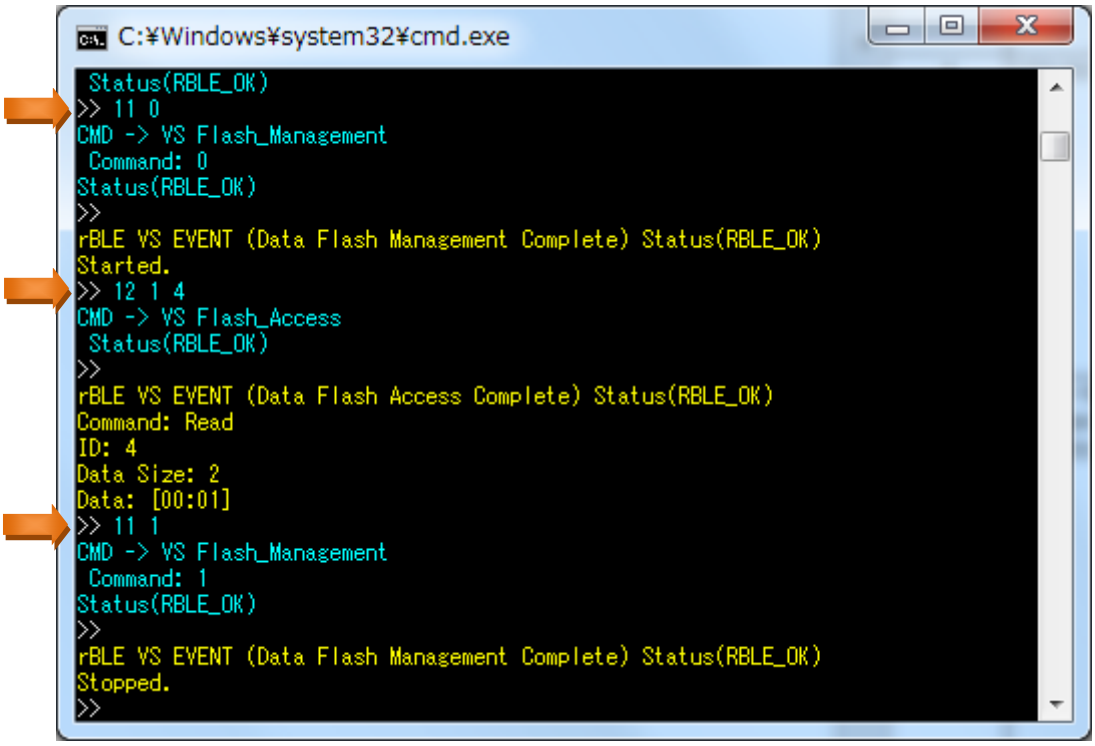
9.6.1 Read Firmware Version

9.6.1.1 Operation Procedure

Enables the vendor specific function. Type "1" and press enter key.



Put the firmware in access start by data flash access management function(VS Flash_Management). Type "11␣0" and press enter key. Then, read the firmware version by data flash access(VS Flash_Access). Type "12␣1␣4" and press enter key. Last, put the firmware in access stop by Data Flash access management function(VS Flash_Management). Type "11␣1" and press enter key.



When put the firmware in data flash access, the MCU is not enter the STOP mode. After the access to the Data Flash finished, certainly put the firmware in access stop.

the format shown below is displayed firmware version format on command prompt.

Data: [Low Byte : High Byte]

Example)
Data: [00:01] = V1.00

9.6.1.2 Command Flow

The figure shown below is command flow of read firmware version. Left side is menu name of rBLE_sample menu. Right side is command to input.

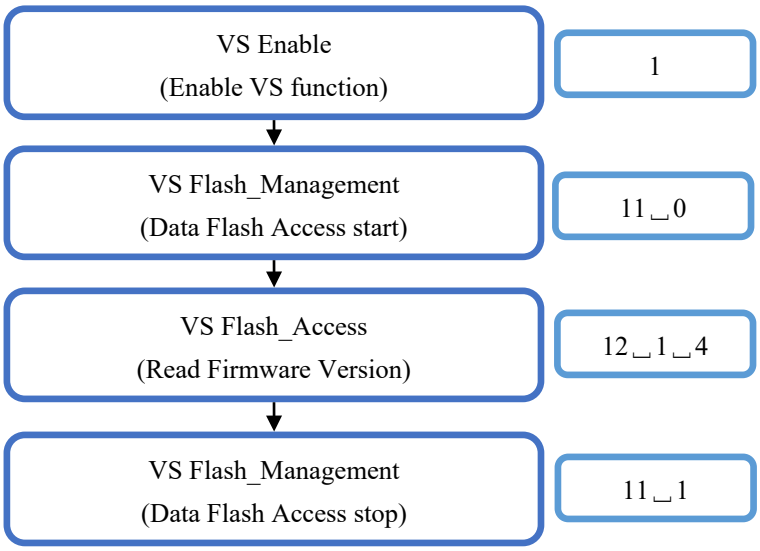
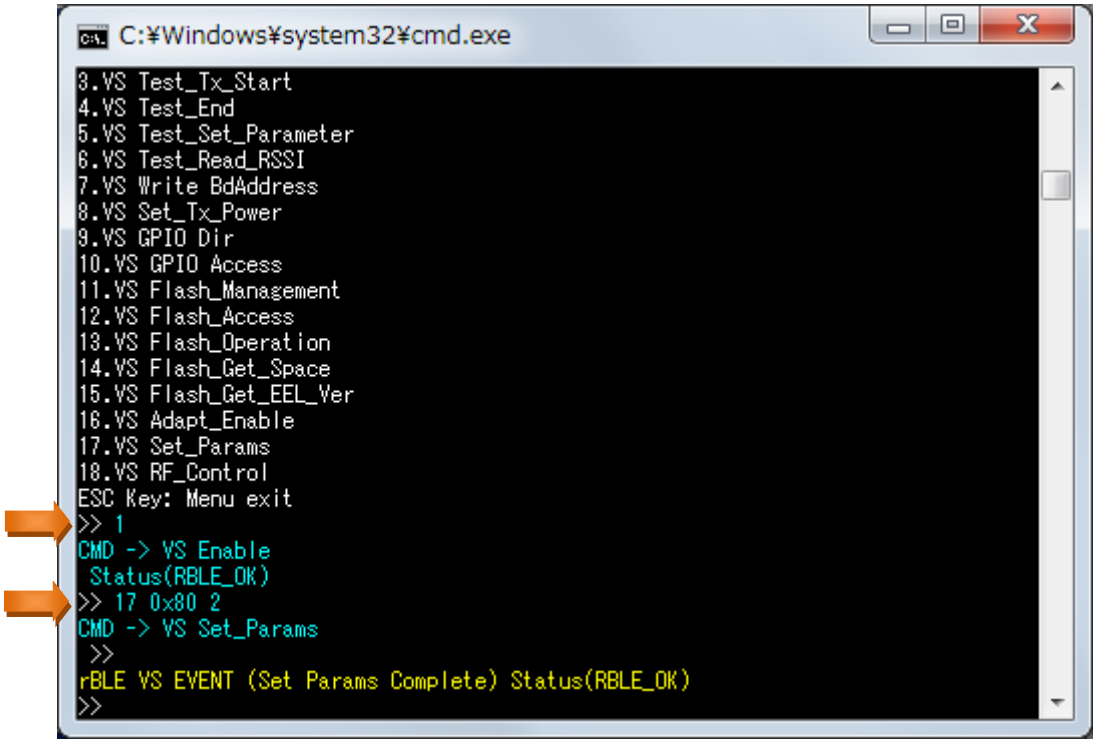


Figure 9-7 Command flow of read firmware version

9.6.2 Set Baud Rate

9.6.2.1 Operation Procedure

Enable the vendor specific function. Type "1" and press enter key. If it has already enabled, it can be omitted. Set the baud rate to the module by set parameter(VS Set_Params). Type "17␣0x80␣2" and press enter key. 2nd parameter is the baud rate setting, 3rd parameter is the baud rate number of the command. For the baud rate number, refer to "Table 9-2 Baud rate number".



In order to reset the module, execute the hardware reset or software reset. For execute the software reset, refer to "9.6.3 Software Reset" of this document. Press ESC key twice to exit the rBLE_sample. Modify the batch file to 19200 bps. Execute the batch file to start the rBLE_sample.

9.6.2.2 Command Flow

The figure shown below is command flow of set baud rate. Left side is menu name of rBLE_sample menu. Right side is command to input.

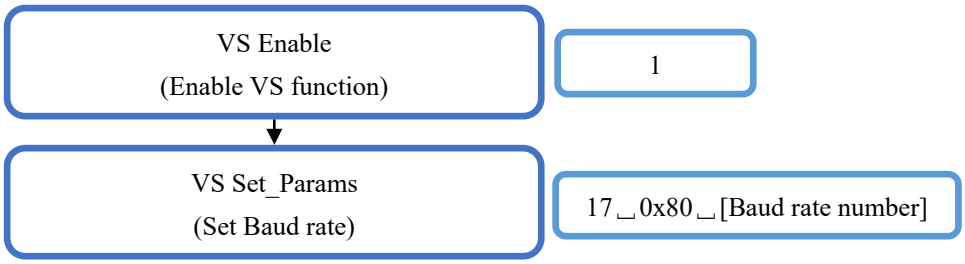


Figure 9-8 Command flow of set baud rate

9.6.2.3 Baud Rate Number

The following shows the correspondence table of the baud rate and baud rate number that can be set.

Table 9-2 Baud rate number

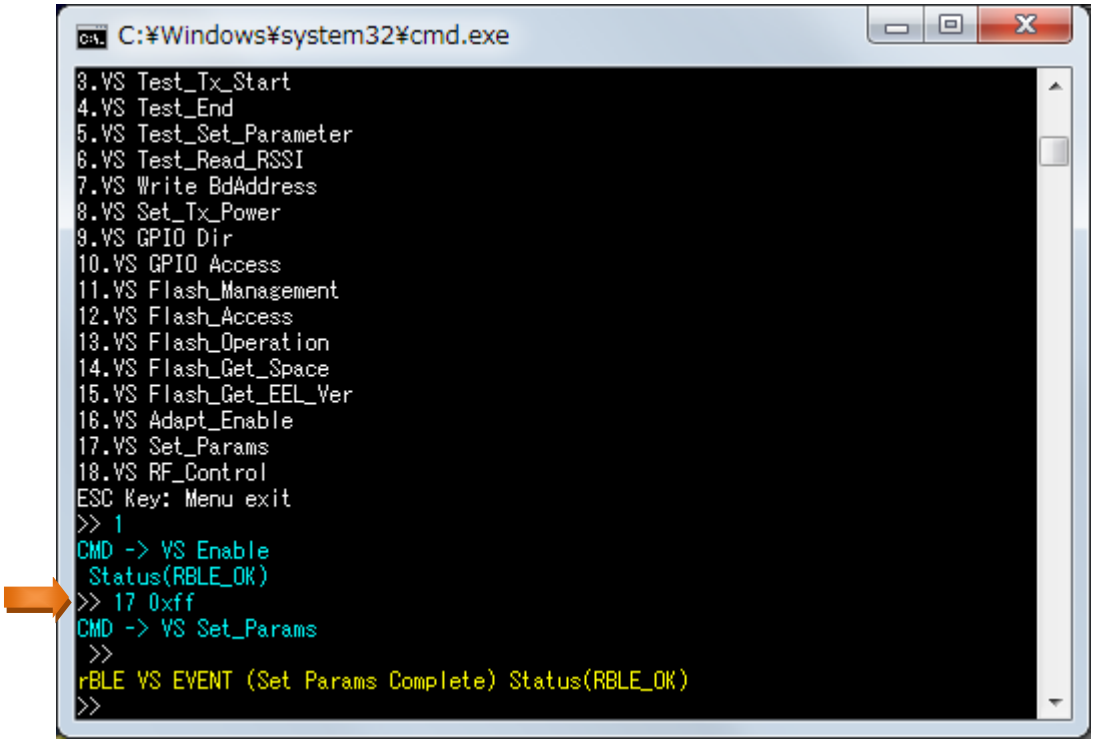
Baud rate (bps)	Baud rate number
4800	0
9600	1
19200	2
38400	3
57600	4
115200	5
250000	6

9.6.3 Software Reset

9.6.3.1 Operation Procedure

Enable the vender specific function. Type "1" and press enter key. If it has already enabled, it can be omitted. Execute the software reset by set parameter(VS Set_Params). Type "17␣0xff" and press enter key. The module will be reset after 1 second.

After execute the software reset, restart the rBLE_sample.



9.6.3.2 Command Flow

The figure shown below is command flow of the software reset. Left side is menu name of rBLE_sample menu. Right side is command to input.

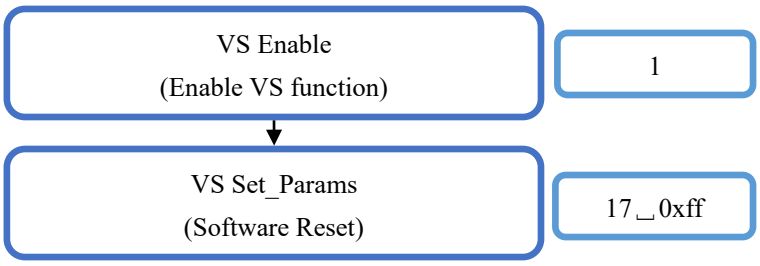


Figure 9-9 Command flow of set baud rate

9.7 Referenced Documents

1. Bluetooth Core Specification v4.2, 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. Battery Service Specification v1.0, Bluetooth SIG
13. Heart Rate Profile Specification v1.0, Bluetooth SIG
14. Heart Rate Service Specification v1.0, Bluetooth SIG
15. Glucose Profile Specification v1.0, Bluetooth SIG
16. Glucose Service Specification v1.0, Bluetooth SIG
17. Time Profile Specification v1.0, Bluetooth SIG
18. Current Time Service Specification v1.0, Bluetooth SIG
19. Next DST Change Service Specification v1.0, Bluetooth SIG
20. Reference Time Update Service Specification v1.0, Bluetooth SIG
21. Alert Notification Service Specification v1.0, Bluetooth SIG
22. Alert Notification Profile Specification v1.0, Bluetooth SIG
23. Phone Alert Status Service Specification v1.0, Bluetooth SIG
24. Phone Alert Status Profile Specification v1.0, Bluetooth SIG
25. Bluetooth SIG Assigned Numbers https://www.bluetooth.com/specifications/assigned-numbers
26. Personal Health Devices Transcoding White Paper v1.4, Bluetooth SIG

9.8 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.
Universally Unique Identifier (UUID)	This is an identifier for uniquely identifying an item. In the BLE standard, a 16-bit UUID is defined for identifying services and their characteristics.
Bluetooth Device Address (BD Address)	This is a 48-bit address for identifying a Bluetooth device. The BLE standard defines both public and random addresses, and at least one or the other must be supported.
Public Address	This is an address that includes an allocated 24-bit OUI (Organizationally Unique Identifier) registered with the IEEE.

9.9 Revision History

Rev	Date	Description	
		Page	Summary
1.00	June 09, 2016	-	First Edition issued.
1.01	Sep 30, 2016	4 9 19 28	Added (4) precautions. Changed Figure 6-2 UART 2-wire with Branch Connection. Added 7.3 GATT database. Added 9.2 BD Address Confirmation.
1.02	Mar 30, 2018	4	Changed Evaluation board name.

RL78/G1D Module Firmware
User's Manual

Publication Date : Rev.1.02 Jan 31, 2022

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.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

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: +44-1628-651-700

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, 200333 P. R. China
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, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 5.1

RL78/G1D Module Firmware

Renesas Electronics Corporation

R01UW0160EJ0102